Proceedings of the 2003 IEEE/RSJ
Intl. Conference on Intelligent Robots and Systems
Las Vegas, Nevada · October 2003

Design and Verification of Controllers for Airships

Jongwoo Kim Jim Keller Vijay Kumar

General Robotics, Automation, Sensing, and Perception (GRASP) Laboratory
University of Pennsylvania, 3330 Walnut Street, Philadelphia, PA 19104
E-mail: {jwk, jfkeller, kumar}@grasp.cis.upenn.edu

Abstract— **Robotic airships have several beneficial properties such as low operation cost, low noise, and low speed flight capability. We present in this paper the design and verification of a feedback control algorithm for waypoint to waypoint navigation of an outdoor blimp. A Rapidly exploring Random Tree (RRT) is used for the validation of the blimp system control law. We describe an implementation of an algorithm that systematically searches the set of all disturbances to validate viability of the control law in the presence of winds. Experimental results with a simulator show that the RRT method can be effective in verifying controller design under unpredictable but bounded disturbances.**

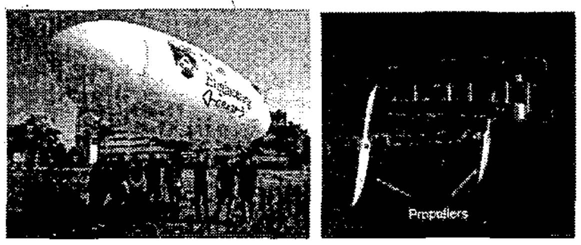

Fig. 1. The nine meter GRASP blimp equipped with electric motors, onboard computer, and sensors and the gondola with the two propellers.

I. INTRODUCTION

Airships occupy a small niche in aviation, mostly used for aerial coverage of sporting events. However, the helium blimp is emerging as a vehicle for research since it affords the user a relatively safe and inexpensive approach to deploying equipment aloft where fast airspeeds are not required or desired. This ease and flexibility of use for scientific applications has led the research community to reconsider this vehicle. The development of autonomous airships has been an active research area recently. Some results include visual servo control [11], [1] and INS/GPS based flight [4]. The GRASP blimp shown in Figure 1 is a nine meter airship equipped with electric motors, on-board computer, and sensors, including GPS, IMU, and several cameras. An on-board laptop communicates with a ground station via a wireless ethernet link. Our eventual goal is to be able to use the blimp for aerial imagery that can be used to localize UGVs and for air-ground coordination.

This paper addresses the development of a feedback control law for waypoint to waypoint inertial navigation. While a theoretical study of stability and convergence is possible [2], the performance of the control law is hard to predict under the existence of disturbances. Thus we are also interested in a methodology to test and validate the performance of the controller using simulation techniques. To illustrate the later point, consider the results of a dynamic simulation shown in Figure 2. The figure shows the blimp trajectories under wind disturbances. It is desired to guide the blimp from the first waypoint at $[0,\ 0,\ -10]^T$ to the destination at $[100\ -100\ -10]^T$ with a target speed of $1m/sec$. Note that in this paper z-axis in the inertial frame is downward. Therefore, the xy-plane is from beneath blimp. It is possible to design a feedback law (as shown later in Section III) that performs well in the absence of wind disturbances (see the "no wind" trajectory in the figure). One might think the worst case wind condition is the broadside wind condition where, for example, the wind blows from starboard to port. Intuitively, it can be thought of as the worst condition for the given case since the blimp has no control actuation in this direction and the winds apply forces to the blimp against the desired track. See Figure 2. However, experiments with simulation show there exist worse trajectories where the blimp is blown far away from the target next waypoint, an example of which is shown in Figure 2. This result illustrates it is hard to obtain the worst case by intuition. Thus, we are interested in developing a suitable paradigm and algorithms for the performance evaluation of feedback controllers. Because disturbances such as those caused by wind gusts can be random, it is natural to think of randomized algorithms for simulation that can be designed to find worst case wind conditions. Our approach uses the Rapidly-exploring Random Tree (RRT) algorithm, a randomized algorithm that has been found to be successful in a broad class of motion planning problems [8], [9]. It is well suited to the problem of quickly searching high-dimensional spaces that have both algebraic and differential constraints. The key idea is to bias the exploration toward unexplored portions of the space by sampling points in the state space, and incrementally pulling the search tree toward them. At the same time, it is possible to bias the search toward

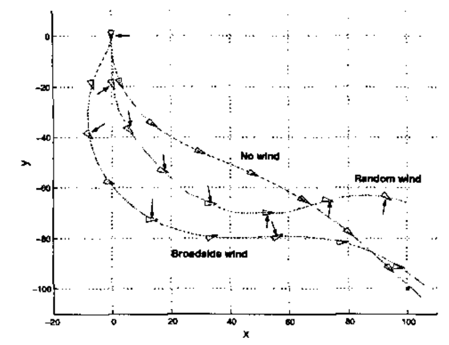

Fig. 2. Comparison of broadside wind and random wind disturbance (arrows mean wind vectors acting on the hull of the blimp). The max. velocity of the wind is $0.5m/sec$.

unsafe sets allowing us to explore the worst case inputs (or disturbances) via simulation. Our interest lies in the implementation of RRTs for the performance evaluation and validation of the feedback control law by searching the set of disturbances systematically. The method allows us to consider the second order dynamics of the airship as well as constraints imposed by the underactuated system. We design a suitable metric that reflects these constraints for searching the configuration space. In addition, we explore a range of sampling strategies for the disturbance space and the configuration space.

The organization of the paper is as follows. We next present the dynamic model of the blimp in Section II. In Section III a feedback control algorithm for waypoint navigation is presented. Section IV describes the performance evaluation of the control law using the randomized method. We end the paper in Section V with discussion of the results and the advantages and the limitations of our approach.

II. DYNAMIC MODELING

In this section, we review the dynamic modeling of the blimp. Our approach is similar to that found in [5], [6], except that we will explicitly account for the inertial effects due to the added mass in the system. The airship is assumed to be a rigid body (ignoring its elasticity) that is symmetric about the $x_b - z_b$ plane. The buoyancy of the hull and aerodynamics of the control surfaces act as stabilizing forces on the blimp. The majority of the equipment for the power supplies, sensing, control, and communication is mounted on the gondola. The gondola is attached under the hull, which locates the center of mass under the center of buoyancy. This provides a stabilizing restoring torque about the roll and pitch axes.

As shown in Figure 3, we have chosen the center of mass, C, as the origin of our body-fixed reference frame. B is the center buoyancy, and $\mathbf{a}$ is the position vector of B in the body fixed frame. It is convenient to write the dynamic equations of motion in the body-fixed frame, whose axes are aligned with the principal axes.

$$M\dot{V} = F_d + F_g + F_a + F_s + F_p \tag{1}$$

where V is the 6×1 Cartesian velocity (twist) vector in the body-fixed frame, M is the 6×6 mass and inertia matrix with added-mass effects, F_d, F_g, F_a, F_s, and F_p are 6×1 force vectors (wrenches). F_d consists of Coriolis and centrifugal terms, F_g includes gravitational and buoyancy-induced forces and moments, F_a includes aerodynamic forces and moments arising from the flow of air around the hull of the blimp, F_s is the vector of aerodynamic forces and moments resulting from the flow of air over the empennage control surfaces (rudder and elevator), and F_p consists of the propulsion forces and moments generated by propeller thrusts. Let the components of the matrices and vectors above be defined as follows in the body-fixed frame. The twist V is given by:

$$V = [v^T \; \omega^T]^T = [v_x \; v_y \; v_z \; \omega_x \; \omega_y \; \omega_z]^T.$$

The velocity dependent inertial forces, F_d, are given by:

$$F_d = \begin{pmatrix} -m_z v_z \omega_y + m_y \omega_z v_y \\ -m_x v_x \omega_z + m_z \omega_x v_z \\ -m_y v_y \omega_x + m_x \omega_y v_x \\ -(J_z - J_y)\omega_z \omega_y + J_{zx}\omega_x \omega_y - (m_z - m_y) v_y v_z \\ -(J_x - J_z)\omega_x \omega_z + J_{zx}(\omega_z^2 - \omega_x^2) - (m_x - m_z) v_z v_x \\ -(J_y - J_x)\omega_y \omega_x - J_{zx}\omega_z \omega_y - (m_y - m_x) v_x v_y \end{pmatrix} \tag{2}$$

where m_x, m_y, m_z, J_x, J_y, J_z, and J_{zx} are the mass and inertia with added-mass effects. Let coordinate of the center of the buoyancy in the body frame, $\mathbf{a}$, be given by

$$\mathbf{a} = [a_x \; a_y \; a_z]^T,$$

and denote by $\mathbf{k}$ the unit vector along the gravitional vector with components

$$\mathbf{k} = [k_x \; k_y \; k_z]^T,$$

in the body-fixed frame.

$$F_g = \begin{pmatrix} k_x(\mathrm{mg} - B) \\ k_y(\mathrm{mg} - B) \\ k_z(\mathrm{mg} - B) \\ a_z k_y B \\ (-a_z k_x + a_x k_z)B \\ -k_y a_x B \end{pmatrix} \tag{3}$$

where mg is weight of the airship (m is the airship mass without the added mass) and B is the buoyancy force. Note that $a_z < 0$ which is important for the stability of the blimp.

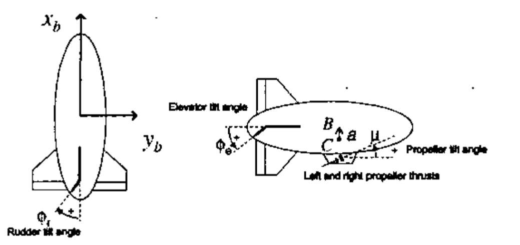

Fig. 3. The body-fixed frame attached to the blimp, and control inputs of the blimp.

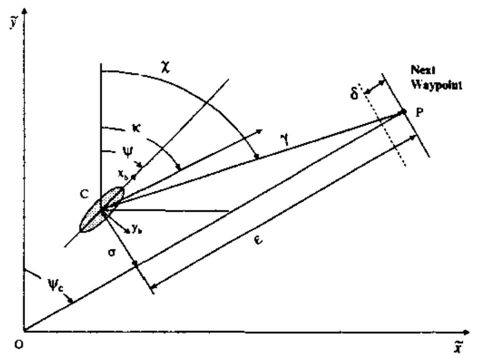

Fig. 4. Parameter definitions. ψ_c is course, χ is bearing to next waypoint, γ is distance to next waypoint, ε is along track error, σ is cross track error, ψ is heading, κ is track angle, δ is waypoint capture proximity, O is last waypoint, and P is next waypoint

The development of equations for the components M, F_a, and F_s may be found in [6].
Finally, let $\mathbf{d} = [d_x\ d_y\ d_z]^T$ denote the position of the stardboard thruster. The position of the port thruster is laterally symmetric to that of the starboard one. The input from the propellers is given by:

$$F_p = \begin{pmatrix} (T_s+T_p)cos(\mu) \\ 0 \\ -(T_s+T_p)sin(\mu) \\ (T_p-T_s)sin(\mu)d_y \\ (T_s+T_p)(d_z cos(\mu)+d_x sin(\mu)) \\ (T_p-T_s)cos(\mu)d_y \end{pmatrix} \tag{4}$$

where T_s is the thrust from the starboard (right) propeller, T_p is the thrust from the port (left) propeller, μ is the propeller tilt angle. The control inputs of the blimp are left thrust, right thrust, thrust tilt angle, elevator tilt angle, and rudder tilt angle. Figure 3 shows axis of the body-fixed frame and the control inputs of the blimp including sign conventions.

III. FEEDBACK CONTROL ALGORITHM

The blimp flight control system features closed loop guidance laws to provide it with an autonomous navigation capability. This capability is structured around the control requirements to move from one inertial waypoint to the next. In this regard, as currently configured, we note that our blimp is not able to sustain hovering flight. Closed loop control laws for the blimp at present are limited to proportional inertial navigation control laws. In this manner, the feedback compensation to correct for unstable blimp dynamics in the presence of angle of attack or sideslip is based on navigation errors. Specifically, closed loop commands are generated to keep the blimp at the desired speed, altitude and ground track commensurate with a sequence of waypoints in inertial space. The flight control computer, in our case a Pentium3 laptop, sequences through a stack of waypoints that constitute its flight plan. This plan can be updated in real-time from a base station for flexibility. Onboard sensing is comprised of a GPS receiver and an IMU. A complete listing of the specifications of our blimp may be found at the website www.cis.upenn.edu/marsteams.

The reference trajectory is generated using a flat earth approximation as specified by the World Geodetic System 1984 model. Desired speed, altitude and ground track to the next waypoint are computed based on the current last and next waypoints, as illustrated in figure Figure 4.

The closed loop guidance laws use proportional feedback to correct speed, altitude and ground track errors. The navigation plan specifies fixed values for desired altitude and speed for each leg of the plan. The directional control law uses the last and next waypoint to compute the desired course to be tracked. Independent errors are then generated based on the cross track error and track angle error. See Figure 4 for definitions of these errors. Cross track error is measured as the offset distance from the desired track and the current position. It is normal to the desired track. In this manner, if the vehicle has been blown off course, steering commands will be generated to bring it back to the desired ground track as opposed to simply maintaining the heading of the desired course, since that would result in a parallel ground track that would not intersect the destination waypoint. In addition, even if the cross track error is zero, if the vehicle track angle, which is defined as the direction of its velocity vector, does not match the desired course, steering commands will be generated to correct the track angle error. This dimension of the control law acts to steer the vehicle to prevent cross track errors. Note that present position of the blimp from GPS to flat earth coordinates ($[\bar{x},\ \bar{y},\ \bar{z}]$) is different from the coordinate in inertial frame ($[x,\ y,\ z]$) since we choose z-axis in the inertial frame as downward. The directional control law can be expressed as follows:

1) Determine target waypoint:
   Last Waypoint: $(\tilde{x}_1,\ \tilde{y}_1)$
   Next Waypoint: $(\tilde{x}_2,\ \tilde{y}_2)$
   $\psi_c = atan2(\tilde{x}_2 - \tilde{x}_1,\ \tilde{y}_2 - \tilde{y}_1)$
   $\chi = atan2(\tilde{x}_2 - \tilde{x},\ \tilde{y}_2 - \tilde{y})$
   $\gamma = norm([\tilde{x}_2 - \tilde{x},\ \tilde{y}_2 - \tilde{y}])$
   $\varepsilon = \gamma cos(\chi - \psi_c)$
   switch to next waypoint pair when $\varepsilon < 1m$
2) Compute $\omega_{z_{des}}$ (desired yaw rate):
   $\sigma = \gamma sin(\chi - \psi_c)$
   $\kappa = atan2(\dot{x},\ \dot{y})$
   $\omega_{z_{des}} = K_\sigma \sigma + K_{\psi_c}(\psi_c - \kappa)$
3) Compute state errors for all axes:
   Heading of the blimp from IMU: ψ
   Forward velocity error: $e_{v_x} = v_{x_{des}} - v_x$, where $v_x = \dot{x}sin(\psi) + \dot{y}cos(\psi)$
   Altitude error: $e_z = z_{des} - (-\tilde{z})$
   Yaw rate error: $e_{\omega_z} = \omega_{z_{des}} - \dot{\psi}$
4) Compute forces and moment $(T_x,\ T_z,\ M_z)$:
   Thrust in x: $T_x = K_x e_{v_x} + T_{x_0}$
   Thrust in z: $T_z = K_z e_z + T_{z_0}$
   Torque in z: $M_z = K_{\omega_z} e_{\omega_z} + M_{z_0}$
   where T_{x_0}, T_{z_0}, and M_{z_0} are trim values.

Errors with respect to desired values are then mixed into commands for the propellers and empennage control surfaces.

5) Calculate control command:
   Case1 $(T_x \neq 0)$
      $\mu = -tan^{-1}(T_z/T_x)$
      $T_p = 0.5(T_x/cos(\mu) + M_z/(cos(\mu)d_y))$
      $T_s = 0.5(T_x/cos(\mu) - M_z/(cos(\mu)d_y))$
   Case2 $(T_x = 0,\ M_z = 0,\ T_z = 0)$
      $\mu = -\pi/2$
      $T_p = T_s = 0$
   Case3 $(T_x = 0,\ M_z = 0,\ T_z \neq 0)$
      $\mu = -tan^{-1}(T_z/T_x)$
      $T_p = T_s = -T_z/(2sin(\mu))$
   Case4 $(T_x = 0,\ M_z \neq 0)$
      $\mu = 0$
      $T_p = M_z/(2d_y),\ T_s = -T_p$
   Elevator tilt angle: $\phi_e = K_{elevator} e_z$
   Rudder tilt angle: $\phi_r = K_{rudder} e_{\omega_z}$
6) Scale inputs to make them between max. and min. maintaining the ratio of T_s and T_p

Control input is obtained via the feedback control law.

$$u = [T_s\ T_p\ \mu\ \phi_e\ \phi_r]^T \tag{5}$$

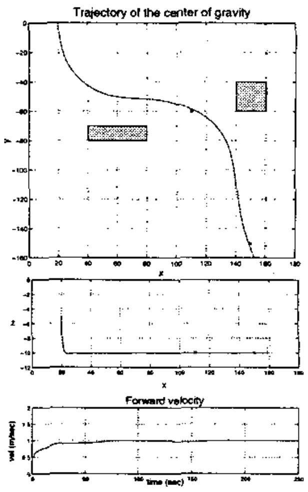

Fig. 5. Trajectory and forward velocity of the blimp generated by the feedback control law.

Figure 5 shows the trajectory and forward velocity of the blimp generated by the feedback control law. The starting position is $[20\ 0\ -5]^T$ with initial forward velocity $0.5m/sec$ and the goal position is $[150\ \ -150\ \ -10]^T$. The target forward velocity is $1m/sec$. An intermediate waypoint is picked to avoid the obstacles.

IV. PERFORMANCE ASSESSMENT AND VALIDATION

In this section, we consider the validation of control law through simulation experiments. While our main focus in this paper is on the controller presented in Section III, the general approach is applicable to other controllers, and indeed to other complex, nonlinear systems. We formulate the problem in a general setting before specializing the formulation to the study of the blimp controller.

A. Problem Formulation

Let x denote the state vector of the dynamic system,

$$x \in X \subset R^n,$$

and let the equations of motion be written as:

$$\dot{x} = f(x,u,w) \tag{6}$$

where $u \in U \subset R^m$ is the control input, and $w \in W \subset R^p$ is the disturbance vector. The controller provides a feedback law, $u = k(x)$, thus reducing Equation 6 to the form:

$$\dot{x} = f(x,k(x),w) \tag{7}$$

We consider a set of unsafe states:

$$\mathscr{S} = \{x | h(x) < 0\} \tag{8}$$

The main question we address in this paper is one of verification. Does there exist a disturbance input $w(x)$ that can drive the control system (7) to $\mathscr{S}$?

In the above setting w is a disturbance that is not controllable and u is an input vector that is determined by a suitably designed control law. One variation on this basic theme is a two-person game where u is chosen to drive the system to a goal set, while w is chosen to drive the system to an unsafe set.

The Rapidly-exploring Random Tree (RRT) algorithm lends itself to analyzing such problems in high-dimensional settings. The basic idea in the RRT method is as follows. Each node in the RRT is a state vector, and each edge in the RRT is associated with a potential path of the dynamic system. At each step, a random state (x_{rand}) is chosen in the state space. The state that is the closest to x_{rand} in the tree, denoted by x_{near} is selected. A (possibly random) set of inputs $(u_1, u_2, \ldots u_M)$, $u_i \in U$ are chosen. The system is simulated M times with each input being applied for Δt seconds starting from x_{near}. From the M potentially new states, the state that is closest to x_{rand} is selected as a new state. This new state is added to the tree as a new vertex. This process is continued until the RRT reaches the goal set or the unsafe set $\mathscr{S}$.

The key underlying idea comes the following observation. Since a vertex with a larger Voronoi region has a higher probability to be chosen as x_{near}, the sizes of largest Voronoi regions are reduced as the tree grows. Therefore, the graph explores the state space uniformly and quickly.

There are many variations on this basic idea. To improve the performance of the RRT, several techniques have been proposed such as biasing the sampling process and making the process less sensitive to the choice of the metric in state space [3], [8], [9]. In this paper, we introduce new modifications to this technique, which are discussed in the next subsection.

B. Validation of the blimp controller

In this paper, we consider the case where $x \in X \subset R^{12}$ is given by:

$$x = \begin{pmatrix} g \\ V \end{pmatrix} \tag{9}$$

where g is an element of the special Euclidean group, $SE(3)$, and V is an element of the Lie algebra $se(3)$. The equations of motion are given by Equation (1), while the control input vector u is obtained from Equations (4) and (5). We consider the control law to be unsafe if there exists a disturbance (wind condition) which could cause the blimp to enter an undesirable region, $\mathscr{S}$, while moving from the last waypoint to the next waypoint or be blown away from the next waypoint. For a given feedback control law, we try to find a wind condition which makes the control law unsafe.

a) Choice of metric: The problem of determining the vertex that is nearest to x_{rand} requires the definition of a metric. Thus the RRT technique requires a metric on the state space. To find a metric that yields good performance can be a very difficult task. The ideal metric is the optimal cost-to-go [3]. However, it is very difficult to find this metric in cases with differential constraints. We use our knowledge of the dynamics of the blimp and the underlying Lie group structure to design a suitable metric.

The shortest distance path given by Hamilton's principle can be obtained analytically for a rigid body with no external forces for which the moments of inertia are identical. As shown in [10], the trajectory between two points $g_1 = (R_1, d_1)$ and $g_2 = (R_2, d_2)$ in $SE(3)$, where R_i is the 3×3 rotation matrix and d_i is the 3×1 translation associated with g_i, is given by:

$$g(t) = (R_1 exp(\Omega t),\ d_1 + (d_2 - d_1)t),\ t \in [0, 1] \tag{10}$$

where Ω is the skew symmetric matrix:

$$\Omega = log(R_1^{-1} R_2).$$

The distance between g_1 and g_2, or the length of the trajectory in Equation (10) is given by:

$$\rho(g_1,\ g_2)\ =\ \|R_1^{-1}(d_2 - d_1)\| + w_a|\theta| \tag{11}$$

where w_a is a scalar weighting factor that incorporates the characteristic length of the system, and $|\theta|$ is the norm of the vector obtained from the components of Ω. Note that $\theta \in (-\pi,\ \pi]$ is the angle of rotation associated with the transformation $g_1^{-1} g_2$.

Unfortunately, there is no equivalent of Equation (11) for general rigid bodies, let alone for the more general case with underactuated dynamics in Equation (1). However, the above simple metric does give us some information about how close two points are in state space in terms of their proximity in configuration space with respect to the metric for unforced motion. We now modify the metric to more closely reflect the underactuated dynamics.

The distance between g_1 and g_2 for the blimp is obtained in the following manner. First consider the trajectory for rigid body motion from g_1 to g_2 in Δt seconds, obtained by rescaling Equation (10). One can calculate the velocity to be:

$$\omega = \frac{\Omega^\vee}{\Delta t}, v = \frac{(d_2 - d_1)}{\Delta t} \tag{12}$$

We define a quadratic form at each point in configuration space according to:

$$\Phi(\omega, v)\ \doteq\ \begin{bmatrix} v^T & \omega^T \end{bmatrix} G \begin{bmatrix} v \\ \omega \end{bmatrix} \tag{13}$$

where

$$G\ =\ diag(G_1,\ G_2,\ G_3,\ G_4,\ G_5,\ G_6). \tag{14}$$

The distance between g_1 and g_2 is now approximated by:

$$\rho(g_1,\ g_2) = \Phi(\omega,v)\Delta t \tag{15}$$

We choose G_i based on three observations. First, there is no direct control of the blimp translation along y_b. This suggests a large value of G_2 compared to G_1. Second, since large translations along z_b requires large propeller tilt angles which compromize the ability to yaw, the value of G_3 must also be greater than G_1. Finally, passive stabilizing torques about the roll and pitch axes restore the pitch and roll angles to zero. This suggests that the cost associated with the pitch and roll angles can be made small. Based on these reasons, we choose weighting factors as $G_1 < G_3 < G_2$ and $G_4 = G_5 = 0$.

Thus the process of computing the distance between two points x_1 and x_2 in state space involves the following steps. First, we consider their projections on configuration space g_1 and g_2 respectively and compute the idealized, unconstrained rigid body motion according to Equation (10). The metric G allows us to incorporate heuristics arising the dynamic constraints of the blimp without explicitly solving a minimization problem to determine the actual distance.

b) Applying the wind disturbance: Our goal is to try to find a wind condition which drives the blimp to an undesired region. We bound the wind disturbance by limiting the magnitude of the wind gust and the rate of change of wind velocity. In The total derivative of the wind velocity is expressed as

$$\frac{dw}{dt} = \frac{\partial w}{\partial t} + (v \cdot \nabla)w \tag{16}$$

where v is the velocity of the blimp. We assume w is not an explicit function of time. In other words, $w = w(x,\ y)$ and $\frac{\partial w}{\partial t} = 0$. Thus the rate of change of wind velocity depends on the tensor ∇w.

In addition to limiting the maximum wind velocity to w_{max}, we also restrict the norm of the tensor ∇w as well as the change of direction of the wind velocity. Thus, as the RRT grows, different wind conditions are used for each time step to find x_{new}.

c) Sampling strategies: The RRT algorithm requires sampling in two different spaces, the disturbance space and the configuration space. We use a uniform grid to sample the disturbance space, and a random sampling strategy for the configuration space. Because of the passive stability properties of roll and pitch, we only sample $g \in SE(2) \times R$, keeping the roll and pitch angles to zero. To improve the performance of RRT, we bias the sampling toward the unsafe set $\mathscr{S}$.

Motivated by the discussion in [7], we considered different sampling strategies in configuration space. For example, it is known that a Halton sequence yields better properties than a uniform random sequence in Euclidean space. However, we found no obvious differences and used the uniform random sampling strategy in all our experiments.

$\|w\|_{max}$	No. of vertices explored in RRT			
(m/sec)	$0 \sim 712$	$713 \sim 2935$	$2936 \sim 4235$	$4236 \sim 10000$
0.3	safe	safe	safe	safe
0.4	safe	safe	safe	unsafe
0.5	safe	safe	unsafe	unsafe
0.6	safe	unsafe	unsafe	unsafe

TABLE I

SAFETY FOR VARIOUS WIND CONDITIONS AND SIZES OF RRT

C. Simulation

In this application, the RRT analysis technique allows the designer to efficiently explore the safeness of the blimp closed loop flight control laws for navigation plans in the presence of obstacles. In a general sense, the technique provides a direct measure of how far an actual flight will deviate from a plan. For specific flight plans, the designer can readily ascertain what magnitude of wind is required to carry the blimp off course enough that flight may not be safe. In this regard, safe operational limits can be established through use of this techinique. In the event that a plan is observed to not be viable, this technique can also be used to allow the planner to iteratively re-select navigation waypoints to ensure performance in winds for flight to a designated point.

Figure 6 shows the trajectories of the blimp under different maximum magnitude of winds. The starting position is $[20\ \ 0\ \ -5]^T$ with initial forward velocity $0.5m/sec$ and the goal position is $[150\ \ -150\ \ -10]^T$. The target forward velocity is $1m/sec$. An intermediate waypoint at $[110\ \ -60\ \ -10]^T$ is picked to avoid the obstacles. For the wind, we assume the max. change rate of magnitude$= 0.05(1/s)$ and the max. change rate of direction$= 18^o/m$. In the case $\|w\| \leq 0.3m/sec$, the blimp does not reach the undesirable regions while it enters these regions under the condition of $\|w\| \leq 0.4m/sec$. The result shows the flight may not safe under the wind condition of $\|w\|_{max} \geq 0.4m/sec$ for blimp target speed at $1.0m/s$. Table I shows no. of vertices explored in the simulation until the blimp reaches the undesirable regions for each max. magnitude of wind.

V. CONCLUSION

We designed a blimp control algorithm for straight and level flight using proportional inertial feedback and proposed the use of RRTs for verification and validation of the controller in the presence of disturbances due to winds. Simulation results were presented for a specific examples. The randomized method allows us to obtain worst case bounded uncertainties (direction-changing wind field)

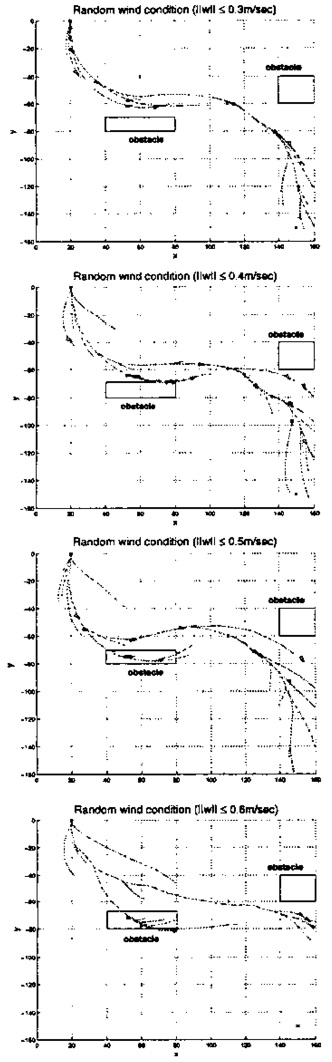

Fig. 6. RRTs of the blimp under different wind conditions:

which drives the airship to undesired regions. Such a worst case analysis is difficult to perform analytically or by intuition. Our on-going work addresses experiments in various wind conditions with feedback control laws for more complicated missions.

ACKNOWLEDGEMENTS

This work was in part supported by DARPA MARS NBCH1020012, ARO Grant DAAD19-02-01-0383 and AFOSR F49620-01-1-0382.

VI. REFERENCES

[1] J. Azinheira, P. Rives, J. Carvhalo, G. Silveira, E. Paiva, and S. Bueno. Visual servo control for the hovering of an outdoor robotic airship. *Proc. of IEEE Int. Conf. on Robotics and Automation*, 2002.

[2] Jonh H. Blakelock. *Autonomous control of aircraft and missiles.* John Wiley and Sons Ltd, 1991.

[3] P. Cheng and S. LaValle. Reducing metric sensitivity in randomized trajectory design. *IEEE/RSJ Int. Conf. on Intelligent Robotics and Systems*, 2001.

[4] A. Elfes, S. Siqueira Bueno, M. Bergerman, and J. G. Ramos. Jr. A semi-autonomous robotic airship for environmental monitoring missions. *IEEE International Conference on Robotics and Automation*, May 1998.

[5] Sergio B. Varella Gomes and Josue G. Ramos. Jr. Airship dynamics modeling for autonomous operation. *IEEE International Conference on Robotics and Automation*, May 1998.

[6] G. A. Khoury and J. D. Gillett. *Airship technology.* Cambridge University Press, 1999.

[7] S. M. LaValle and M. S. Branicky. On the relationship between classical grid search and probabilistic roadmaps. *Proc. Workshop on the Algorithmic Foundations of Robotics*, 2002.

[8] S. M. Lavella and J. Kuffner Jr. Randomized kinodynamic planning. *Proc. of IEEE Int. Conf. on Robotics and Automation*, 1999.

[9] S. M. Lavella and J. Kuffner Jr. Rapidly-exploring random trees: Progress and prospects. *2000 Workshop on the Algorithmic Foundations of Robotics*, 2000.

[10] M Zefran. *Continuous methods for motion planning.* PhD thesis, U. of Pennsylvania, 1996.

[11] H. Zhang and J. P. Ostrowski. Visual servoing with dynamics: control of an unmanned blimp. *Proc. of IEEE Int. Conf. on Robotics and Automation*, 1999.