Proceedings of the 2003 IEEE
International Conference on Robotics & Automation
Taipei, Taiwan, September 14-19, 2003

Motion Planning of Aerial Robot using Rapidly-exploring Random Trees with Dynamic Constraints

Jongwoo Kim James P. Ostrowski

General Robotics, Automation, Sensing, and Perception (GRASP) Laboratory
University of Pennsylvania, 3401 Walnut Street, Philadelphia, PA 19104-6228
E-mail: {jwk, jpo}@grasp.cis.upenn.edu

Abstract*— *We describe some recent work on randomized motion planning algorithms and consider the problem of motion planning for systems with both kinematic and dynamic constraints. Such a problem is often referred to as kinodynamic motion planning. A Rapidly-exploring Random Tree (RRT) is used for the motion planning of a blimp system, and some techniques for improving the performance of the planner are proposed. Based on a dynamic model of a blimp, dynamic constraints are introduced into the algorithm design and simulations are conducted for trajectory generation.

I. INTRODUCTION

The problem of finding a suitable trajectory and control inputs to drive a robot from an initial state to a goal state while satisfying physically-based dynamic constraints has been an active research area in many fields [7]. The feasible trajectory can be used as a test of performance for a robot design or as a reference trajectory for feedback control. The decoupled approach, in which one solves a basic path planning problem followed by finding a trajectory and controller that satisfies the dynamics, has been applied successfully to a variety of problems. However, decoupled approaches often fail to find the feasible solution due to the system dynamics or control input constraints. It is often the case that kinematic and dynamic constraints have to be taken into consideration simultaneously. This type of problem is known as kinodynamic planning [2]. While there has been a great deal of work on this problem, it is known that complete methods for such robots have overwhelming complexity. Therefore, it is reasonable to focus on randomized algorithms that can solve many challenging high-dimensional problems efficiently at the expense of completeness. There are several successfully applied randomized algorithms to the general problems of path planning in a high-dimensional configuration space, such as randomized potential field methods and probabilistic roadmap methods [8], [5]. However, they do not naturally extend to general problems that involve differential constraints.

A Rapidly-exploring Random Tree (RRT) is a randomized algorithm that is designed for a broad class of path planning problems [9], [10]. The advantage of RRTs is that they can be directly applied to nonholonomic and kinodynamic planning. We designed a motion planning algorithm based on the RRT method, including the effect of the dynamics of the blimp.

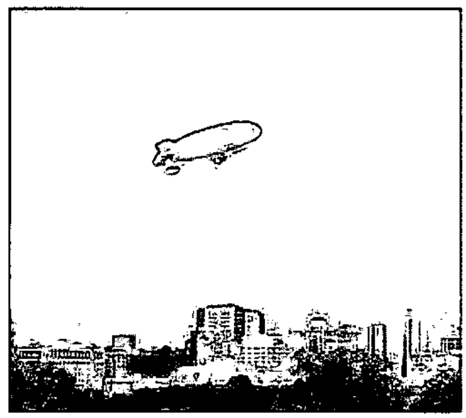

Fig. 1. Nine meter blimp equipped with on-board computer and sensors

Blimps have been used in various tasks such as surveillance, freight transport, advertising, and monitoring roles because of several beneficial properties such as low operation cost, low noise, and low speed stable flight capability [1]. The development of an autonomous airship has been an active research area. Some results include visual servo control [11] and INS/GPS based flight [1]. Since a blimp is an underactuated system that has thrust forces and angular inputs as control inputs, we need to analyze not only the kinematics but also the dynamics of the system. The system dynamics of the blimp are highly nonlinear. In addition, there exist various constraints on the control inputs. This leads us to consider randomized motion planning algorithms. We also propose heuristics taking into account dynamic constraints in designing the algorithm and provide some justification for their use.

The remainder of the paper is organized as follows:

Section 2 gives a formal problem formulation. In Section 3 the dynamics and modelling of the blimp are presented. Section 4 describes the motion planning algorithm used for the generation of the trajectory and nominal inputs. Section 5 presents some numerical simulation results. Finally, discussion of the results and concluding remarks are given in Section 6.

II. Problem Formulation

The class of problems we consider can be formulated as follows.

1) State Space : $X \subset R^n$
2) Boundary Values : $x_{init} \in X$ and either $x_{goal} \in X$ or $X_{goal} \subset X$
3) Control Inputs : A set $U \subset R^m$, which specifies the complete set of controls
4) Equations of Motion : $\dot{x} = f(x,u)$
5) Constraints on the Inputs : $f(u,\dot{u},\ldots) = 0$, $g(u,\dot{u},\ldots) \leq 0$

Motion planning is generally viewed as a search for the trajectory $\tau : [t_0,t_f] \rightarrow X$ from an initial state x_{init} to a goal state x_{goal} or a goal region X_{goal} and the associated set of inputs, $u : [t_0,t_f] \rightarrow U$. The trajectory is a time-parameterized continuous path that satisfies the kinodynamic constraints. There also exist constraints on inputs such as bounded magnitude of the inputs and bounded rate of change of the inputs. It might also be appropriate to select a path that optimizes some cost functions, such as the time to reach x_{goal}.

In this paper, we address the problem of motion planning to drive the blimp starting from an initial configuration (position and orientation) with zero velocity and finishing at a goal region. We consider here only goal regions based solely on the configuration variables(pose) of the blimp, without any restriction on the final velocity. This can be envisioned as a mission with take-off from the platform and finishing by entering a desired region.

III. Dynamic Modeling and a Simulator

A. Dynamics and modeling of the blimp

We review the dynamic modeling of the blimp. This modeling is similar to that found in [4], [6]. Note however that we have chosen the center of mass as our reference frame and included important inertial effects due to the added mass. The airship is assumed as a rigid body ignoring its elasticity and symmetric about the xz plane. The dynamic model is expressed in the body-fixed frame with the origin at the center of mass as:

$$M\dot{X} = F_d + F_g + F_a + F_s + F_p$$

where

$X : 6 \times 1$ velocity vector($[u\ v\ w\ p\ q\ r]^T$) in the body frame composed by linear $(u\ v\ w)$ and angular $(p\ q\ r)$ velocity

$M : 6 \times 6$ mass and inertia matrix with virtual terms

F_d : centrifugal and Coriolis terms($[f_1\ f_2\ f_3\ f_4\ f_5\ f_6]^T$)
$f_1 = -m_z wq + m_y rv$
$f_2 = -m_x ur + m_z pw$
$f_3 = -m_y vp + m_x qu$
$f_4 = -(J_z - J_y)rq + J_{zx}pq - (m_z - m_y)vw$
$f_5 = -(J_x - J_z)pr + J_{zx}(r^2 - p^2) - (m_x - m_z)wu$
$f_6 = -(J_y - J_x)qp - J_{zx}rq - (m_y - m_x)uv$

F_g : gravity and buoyancy forces and moments

$$F_g = \begin{pmatrix} \lambda_{31}(W-B) \\ \lambda_{32}(W-B) \\ \lambda_{33}(W-B) \\ a_z\lambda_{32}B \\ (-a_z\lambda_{31} + a_x\lambda_{33})B \\ -\lambda_{32}a_xB \end{pmatrix}$$

where W: airship weight, B: buoyancy force, $[a_x\ a_y\ a_z]^T$: coordinate of the center of the buoyancy in the body frame, $[\lambda_{31}\ \ \lambda_{32}\ \ \lambda_{33}]^T$: coordinate of the Z-axis of the inertial frame viewed in the body frame

F_a : aerodynamic forces and moments arising from the hull of the blimp

F_s : aerodynamic forces and moments arising from control surfaces(rudder and elevator)

F_p : propulsion forces and moments generated by engine thrusts

$$F_p = \begin{pmatrix} (T_s + T_p)cos(\mu) \\ 0 \\ -(T_s + T_p)sin(\mu) \\ (T_p - T_s)sin(\mu)dy \\ (T_s + T_p)(dzcos(\mu) + dxsin(\mu)) \\ (T_p - T_s)cos(\mu)dy \end{pmatrix}$$

where T_s: starboard thrust, T_p: port thrust, μ: engine tilt angle, $[dx\ dy\ dz]^T$: position of the thrusters $(dy > 0)$

The mass and inertia matrix includes both actual inertia as well as virtual inertia [4], [6]. Virtual inertia produces additional moment if the hull is flown at an incidence. The buoyancy of the hull and 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 stabilizing restoring torque about the roll and pitch axes. There also exist drift

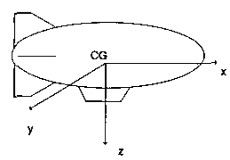

Fig. 2. Axis of the body fixed frame

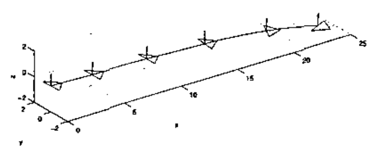

Fig. 3. Trajectory of the blimp generated by the simulator

terms such as drag, gravity, and buoyancy forces in the dynamics.

B. A blimp simulator

Based on the dynamic analysis, a blimp simulator has been implemented in Matlab and Simulink. This simulator can generate the configuration in $SE(3)$ and velocities of the blimp for given initial conditions and control inputs. Figure 3 shows a trajectory generated for given control inputs.

Euler angles are used to represent the rotational component of the state vector. Orthogonal rotation matrix $SO(3)$ can be obtained from Euler angles body-fixed 321. Over the acceptable operation range of the blimp, the Euler angles can be used without singularity problems. If orientation is composed of a rotation of ψ about the z-axis, followed by a rotation of θ about the y-axis, and a rotation of ϕ about the x-axis, the relationship between the Euler angles and rotation matrix is,

$$R = \begin{bmatrix} c_\psi c_\theta & -s_\psi c_\phi + c_\psi s_\theta s_\phi & s_\psi s_\phi + c_\psi s_\theta c_\phi \\ s_\psi c_\theta & c_\psi c_\phi + s_\psi s_\theta s_\phi & -c_\psi s_\phi + s_\psi s_\theta c_\phi \\ -s_\theta & s_\phi c_\theta & c_\phi c_\theta \end{bmatrix}$$

From the velocity with respect to the body-frame, the position and the Euler angles can be obtained. The relationship between linear velocities of the center of mass in the inertial frame and linear velocities in body-frame is,

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{pmatrix} = R \begin{pmatrix} u \\ v \\ w \end{pmatrix}$$

The relationship between time derivative of Euler angles and angluar velocities in body-frame is,

$$\begin{pmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix} = \frac{1}{c_\theta} \begin{pmatrix} c_\theta & s_\phi s_\theta & c_\phi s_\theta \\ 0 & c_\phi c_\theta & -s_\phi c_\theta \\ 0 & s_\phi & c_\phi \end{pmatrix} \begin{pmatrix} p \\ q \\ r \end{pmatrix}$$

IV. Motion Planning Algorithm

The Rapidly-exploring Random Tree (RRT) is an algorithm which is suited for 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, leading to quick and uniform exploration of even high-dimensional state spaces. RRT methods can be applied to a broad class of problems including kinodynamic motion planning. We present a short review of the RRT algorithm here. A detailed presentation can be found in [9], [10]. The underlying premise is to build a graph structure with nodes at explored positions (and velocities) and with edges describing the control inputs needed to move from node to node.

For each step, a random state (x_{rand}) is chosen in the state space. Then, x_{near} in the tree that is the closest to the x_{rand} in metric ρ is selected. Inputs $u \in U$(input set) are applied for Δt, making motions toward x_{rand} from x_{near}. Among the potential new states, the state that is as close as possible to x_{rand} is selected as a new state, x_{new}. The new state is added to the tree as a new vertex. This process is continued until x_{new} reaches x_{goal}.

Since a vertex with a larger Voronoi region has a higher probability to be chosen as x_{near} and it is pulled to the randomly chosen state as close as possible, the size of larger Voronoi regions is reduced as the tree grows. Therefore, the graph explores the state space uniformly and quickly.

To improve the performance of the RRT, several techniques have been proposed such as biased sampling and reducing metric sensitivity [3], [9], [10]. We review them and present heuristic methods applied to our simulation.

A. Techniques for performance improvement

1) Bias: To improve the performance of the RRT method, bias techniques in choosing random states have been proposed [10]. For example, more chances to pick x_{goal} as a random state or a normal distribution centered at the goal state can improve the planner. However, too strong artificial bias could degrade RRTs' quick exploration property. In this paper, a normal distribution bias is

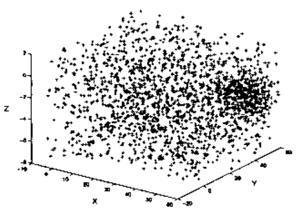

Fig. 4. Biasing for choosing x_{rand}

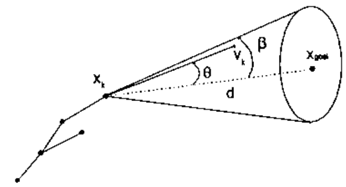

Fig. 5. The condition for subconnection

utilized. Figure 4 shows positions of x_{rand} biased around x_{goal}. The position of x_{goal} is $[50\ 50\ \ -3]^T$ in the figure. Euler angles can be biased in a similar way. Since states near the goal state have higher probability to be chosen as a random state, the vertices tend to grow toward the goal state.

2) Reducing metric sensitivity: 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 the metric in cases with differential constraints. Instead of trying to obtain the ideal metric, the reduction of metric sensitivity method was proposed [3]. Exploration information on which inputs already have been applied for each state is used to reduce the metric sensitivity. This idea is applied in our simulation. An input already applied for a state is not considered as a candidate again, and a state for which all possible inputs are exhausted is excluded from x_{near}.

B. Heuristics

1) Rate of change of control inputs: We need to consider the rate of change of the control inputs due to the actuator constraints. For example, control inputs cannot be turned on and off in no time. In the RRT algorithm, constraints on the rate of change of the control inputs can be directly considered in the planner. In our approach, initial inputs and final inputs for Δt are calculated based on rate of change of each control input. Then the control inputs vary linearly within the two values over the Δt to prevent discontinuity.

2) Subconnection: Performance of the RRT algorithm can be improved by a method we call subconnection. Figure 5 shows the condition for subconnection. The main idea is to choose one good state as a new initial state and conduct RRT searching again on a smaller set.

We note that for a dynamic system such as the blimp, the velocity plays an important role in determining whether two points are "close enough". If the Euclidean distance between x_k and x_{goal}, denoted by d, is smaller than some value α and the linear velocity vector in the inertial frame at the state x_k, denoted by v_k, lies within a cone made by x_k, x_{goal}, and β, (i.e. $\theta < \beta$) the state x_k is considered roughly connected to the goal state. β changes with d so that the cone narrows as d gets smaller. Then, a second RRT planning is executed once again with the roughly connected state and the goal state as the new initial state and the goal state in a smaller environment that includes the two states. In the second RRT, smaller Δt and more candidate inputs than in the first RRT can be used for better performance. Because the random states are selected in a smaller region, the second RRT is more efficient than the first one in connecting x_k to x_{goal}. The entire trajectory is simply the connection of two trajectories obtained in the first and the second RRT.

3) Metric design: The added mass induces a moment in case of flight in incidence. The pitching moment can be countered by both the buoyancy force and the fin effectively. However, yawing moments cannot be completely corrected by the fin and torque generated by thrusts in many cases, resulting in uncontrollable rotation. Therefore, yawing moments due to the added mass and the angle between the heading and the velocity of the center of mass are also considered in the metric. A simple metric based on a weighted Euclidean distance is utilized in determining x_{near}.

$$\rho(x_{near}) = d + w_a \|a\|,$$

where d is Euclidean distance, a is differences of Euler angles, and w_a is a weight factor. For a metric to determine x_{new}, the added mass effect is considered. Additionally, a similar idea to the condition for subconnection is applied so that the state 'flying toward' the goal region has smaller metric.

$$\rho(x_{new}) = d + w_a\|a\| + w_{p1}|b| + w_{p2}|\tau| + w_v|\theta|$$

where b is angle of incidence, τ is the yawing moment due to the added mass, θ is the angle defined in Figure 5, and w_a, w_{p1}, w_{p2}, w_v are weight factors.

V. Simulation Result

The blimp model discussed in this paper has five control inputs which are left and right engine force, engine tilt

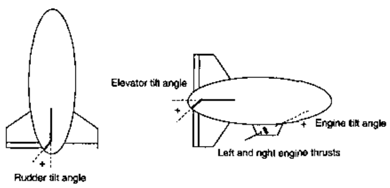

Fig. 6. Control inputs of the blimp

angle, elevator tilt angle, and rudder tilt angle as shown in Figure 6. The maximum and minimum engine thrust are 16N (8N each for left and right engines) and 0N respectively. The tail fins can be rotated from $-90°$ to $90°$ and the engines can be tilted from $-90°$ to $90°$. It is assumed that it takes 2 seconds for engine thrusts to change from their minimun values to their maximum values, and $30°/sec$ for max. engine and fin tilt speed. In the simulation, vertices placed beyond the environment or with the negative x-direction body velocity are discarded in the process of RRT generation to prevent unnecessary searching.

The initial state is $(0\ 0\ 0\ 0\ 0\ \pi/3)$ in postion and Euler angles with zero velocities. The goal state is $(50\ 50\ -3\ 0\ 0\ 0)$. The tolerance for the goal region is $(2\ 2\ 2\ \pi/6\ \pi/6\ \pi/6)$.

An RRT for a roughly connected trajectory is conducted first, followed by a second RRT refinement in the smaller environment. Figure 7 shows the trees and trajectory for the first RRT. After reaching the first state which satisfies the subconnection condition, the second RRT is conducted with the new x_{init} and the environment.

Figure 8 shows the trees for the second RRT and the final connected trajectory.

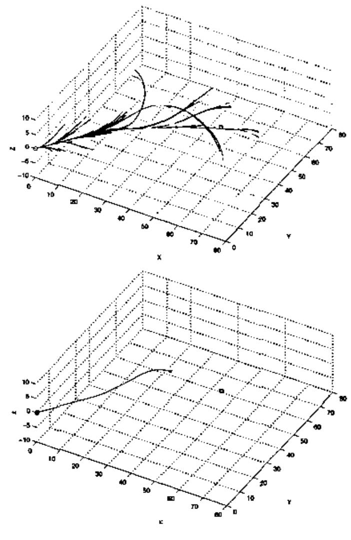

Fig. 7. 1st RRTs and trajectory for 1st RRT

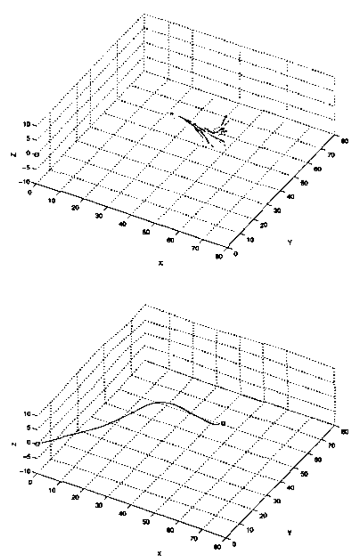

Fig. 8. 2nd RRTs and final trajectory

VI. Discussion and Conclusion

We implemented the Rapidly-exploring Random Trees algorithm and conducted simulations for a kinodynamic motion planning problem of a blimp system. We showed that the trajectory planning of the system with dynamic constraints and drift can be achieved with the randomized algorithm. Several techiniques are implemented to improve the performance of the planner such as the bias and the metric design considering the dynamics of the blimp. Conceptually, the RRT is a simple method. However, it is also a very efficient method in that it can be applied to a broad class of problems such as holonomic, nonholonomic, and kinodynamic motion planning.

In the future, we will develop adequate control strategies for various tasks and behaviors such as volume sweeping and trajectory following, and conduct a set of real experiment using the blimp with the on-board computer and sensors. Also, we note that the solution does not provide a feedback controller, so one must be added to provide more robustness.

Acknowledgements

We gratefully acknowledge discussions with Vijay Kumar on blimp models and motion planning/control algorithms and the support of DARPA/IPTO, grants AF F30602-01-2-0563, and NBCH1020012.

VII. REFERENCES

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

[2] J. Canny B. Donald, P. Xavier and J. Reif. Kinodynamic motion planning. *Journal of the Acm, 40(5): 1048-1066*, November 1993.

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

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

[5] L. E Kavraki and J. C. Latombe. Randomized preprocessing of configuration space for fast path planning. *Proc. IEEE Int. Conf. Robotics and Automation, 2138-2145*, 1994.

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

[7] J. J. Kuffner. Motion planning with dynamics. *Physiqual*, March 1998.

[8] J. C. Latombe L. Kavrakim P. Svestka and M. H. Overmar. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *Int. Transactions on Robotics and Automation: 12(4):566-580*, 1996.

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

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

[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.