Preliminary Evaluation of Rapidly-Exploring Random Trees for Sling-Load Flight Guidance

Eric N. Johnson
Lockheed Martin Associate Professor of Avionics Integration
Georgia Institute of Technology
270 Ferst Drive
Atlanta, Georgia, USA 30332-0150

John G. Mooney
Graduate Research Assistant
Georgia Institute of Technology
270 Ferst Drive
Atlanta, Georgia, USA 30332-0150

ABSTRACT

A novel approach to providing guidance for helicopters with under-slung loads is presented. Rapidly-exploring Random Trees are adapted to plan trajectories for simplified helicopter-load models. The algorithm is presented with a task to place a load in a specific location at a moment when the load is motionless, mimicing the actions of helicopter-based Christmas tree harvesting. The solutions provided by the RRT method vary in aggressiveness and precision, and are not ready for live implementation, but show promise in future development of this approach.

NOTATION

P: the pivot point of the pendulum
M: the pendular mass
$\vec{r}$: position vectors, with subscript denoting point P or point M
$\vec{\eta}$: unit vector pointing from P to M
ρ: air density (slugs/ft^3)
f: equivalent flat plate area of the load (ft^2)
l_0: sling length (ft)
m: mass of load (slugs)
$\vec{g}$: gravitational acceleration in vector form
$\vec{a}_c$: centripetal acceleration
$\vec{a}_{AP}$: acceleration of the load due to acceleration of the pivot
$\vec{D}$: viscous drag
$\vec{F}_{CE}$: virtual force for constraint enforcement
k_{CE}, c_{CE}: spring and damping coefficients used for constraint enforcement
τ: unit vector in the direction of pivot acceleration
T, θ: magnitude and angle of pivot acceleration
$\vec{x}$: state vector
$\vec{u}$: vector of control inputs
J: the value of the cost function
α: relative cost weighting between position error and velocity error

INTRODUCTION

Automatic guidance and control are rapidly becoming an integral part of civil aviation operations. However, there are a number of tasks which, thus far, still require the skill, adaptability, and experience of a human pilot. One task in particular that requires an especially skilled pilot is helicopter sling-load operations.

Presented at the 2nd Asia-Australia Rotorcraft Forum and 4th International Basic Research Conference on Rotorcraft Technology, Tianjin, China, September 8–11, 2013.

Fig. 1. Helicopter loading a Christmas tree for transport.

Slung-loads, used in such operations as disaster relief, fighting forest fires, and Christmas tree harvesting (Figure 1), have highly nonlinear dynamics that are strongly coupled to the aircraft. Additionally, helicopters typically do not have any additional control mechanisms to handle the additional degrees of freedom represented by such a load. Motion planning thus is an important aspect of slung-load operations, as it needs to ensure a load is transported to its destination quickly and safely while avoiding excessive forces.

Review of Relevant Literature

The field of motion planning has been well-studied, including the unique aspects of the problem presented by aircraft. Planning Algorithms by Lavalle (Ref. 1) is possibly the most

complete resource available, serving as the textbook for more than a dozen robotics courses at universities across the US.

However, a helicopter carrying a slung-load presents a unique set of challenges to the motion planning problem. Depending upon the complexity of the model, it brings at least two additional degrees of freedom to the dynamic system, typically without any additional control mechanism. The problems inherent in slung load operation have been studied since the earliest days of helicopters (Ref. 2). There have been efforts to model slung-load dynamics (Refs. 3, 4), classify the system stability (Ref. 5), and develop methods to directly impart control forces upon the load via aerodynamic surfaces (Ref. 6), winches (Ref. 5), or even reaction wheels (Ref. 7). More recently, researchers have classified the dynamics using empirical system identification techniques (Refs. 8, 9).

One body of research of particular interest was developed by Bisgaard, la Cour-Harbo, and several other authors. They created an extremely detailed dynamic model in (Ref. 10). Their model includes load attitude, line collapse and line collision (or "snap"), and load aerodynamics including rotor wake effects, in addition to standard helicopter dynamics. They also developed a method for estimating the load state using vision (Ref. 11), stabilizing the load via input shaping (Refs. 12, 13), and planning trajectories (Ref. 14).

A natural candidate approach to motion planning for helicopters with slung loads is Rapidly-exploring Random Trees, or RRTs. RRTs have been shown to be able to efficiently search high-dimensional spaces with nonlinear metrics and constraints. Lavalle and Kuffner introduced RRTs in 1998 (Ref. 15) and subsequently proved and demonstrated a number of properties of RRTs (Ref. 16), as well as exploring several key variants (Refs. 17, 18). Their applications have included systems with many degrees of freedom as well as holonomic and nonholonomic constraints. Frazzoli and Karaman (Ref. 19) have extended this work by developing a version of RRTs that can approximate an optimal solution.

This paper will be organized as follows. First, the framework of the RRT used in this research will be described, including the dynamic model(s) for the helicopter-load system and the logic used to extend the tree and connect new nodes. Second, results from simulations of increasingly complex models will be presented. Finally, a discussion of the key insights developed in this research will be presented.

PROBLEM FORMULATION

Guidance Algorithm

While there are a wide variety of extensions to the basic RRT method, the research here will use one of the most basic. It is very similar to the basic kinodynamic planning presented in (Ref. 17). This method does not promise optimality; however, it does ensure that the planned path is dynamically feasible by construction. The tree is expanded by generating a random state from a set of uniform distributions (though occasionally the goal state is selected according to a user-defined bias). The nearest node in the existing tree is selected using a distance function. That node is the point from which the tree grows. The input space of the system is reduced to a series of basis functions selected and scaled randomly. The tree is grown from the nearest node using the selected control and simulating forward in time. Barring failed constraint or collision checks, the state at the end of the control sequence is added to the tree (Algorithm 1).

Algorithm 1 Generate RRT (adapted from (Ref. 15))

procedure GENERATE_RRT($x_{init}, x_{goal}, K, \Delta t, bias$)
  $T.init(x_{init})$;
  for $k = 1$ to K **do**
    $x_{rand} \leftarrow$ RANDOM_STATE($x_{goal}, bias$);
    $x_{near} \leftarrow$ NEAREST_NEIGHBOR(T, x_{rand});
    $u \leftarrow$ SELECT_INPUT();
    $x_{new} \leftarrow$ NEW_STATE($x_{near}, u, \Delta t$);
    $T.add\_node\_and\_edge(x_{new}, x_{near}, u)$;
    if $x_{new} = x_{goal}$ **then**
      break;
    end if
  end for
  $x_{nearest} \leftarrow$ NEAREST_NEIGHBOR(T, x_{goal});
  $U \leftarrow$ ASSEMBLE_CONTROL_SEQ($T, x_{nearest}$);
  return U
end procedure

Dynamic Models for Planning

For the purposes of this paper, the helicopter-load system will be modeled as a pendulum attached to, alternatively, a constant-altitude double integrator, a two-degree-of-freedom double integrator (Equations (1) - (10)), and a point mass with first-order pitch and thrust dynamics (Equations (11) - (13)). The equations of motion for these systems are formulated using multibody equations of motion, with a virtual spring-mass-damper in the linkage to enforce the constraint between the pivot and the mass. These models are drastic simplifications of real aircraft with slungloads, but are being examined to determine if they are "good enough" for planning purposes.

$$\vec{x} = \left\{ \vec{r}_P^T,\ \dot{\vec{r}}_P^T,\ \vec{r}_M^T,\ \dot{\vec{r}}_M^T \right\}^T \tag{1}$$

$$\dot{\vec{x}} = \left\{ \begin{array}{c} \dot{\vec{r}}_P \\ \ddot{\vec{r}}_P \\ \dot{\vec{r}}_M \\ \vec{a}_{EXT} + \vec{a}_C + \vec{a}_{AP} + \frac{\vec{F}_{CE}}{m} \end{array} \right\} \tag{2}$$

where

$$l = \|\vec{r}_M - \vec{r}_P\| \tag{3}$$

$$\vec{\eta} = \frac{(\vec{r}_M - \vec{r}_P)}{l} \tag{4}$$

$$\vec{a}_C = \frac{\left\| \dot{\vec{r}}_M - \left(\dot{\vec{r}}_M^T \vec{\eta}\right) \vec{\eta} \right\|^2}{l_0} \vec{\eta} \tag{5}$$

$$\vec{a}_{AP} = \left(\ddot{\vec{r}}_P^T \vec{\eta}\right) \vec{\eta} \tag{6}$$

$$\vec{D} = -\frac{\rho f}{2} \left\| \dot{\vec{r}}_M \right\| \dot{\vec{r}}_M \tag{7}$$

$$\vec{a}_{EXT} = \left(\frac{\vec{D}}{m} + \vec{g}\right) - \left[\left(\frac{\vec{D}}{m} + \vec{g}\right)^T \vec{\eta}\right] \vec{\eta} \tag{8}$$

$$\vec{F}_{CE} = -k_{CE}(l - l_0)\vec{\eta} - c_{CE}\left[\left(\dot{\vec{r}}_M^T \vec{\eta}\right) - \left(\dot{\vec{r}}_P^T \vec{\eta}\right)\right] \vec{\eta} \tag{9}$$

$$\ddot{\vec{r}}_P = \vec{u} \tag{10}$$

The coefficients on the constraint enforcement are set so that the spring-mass-damper frequency is an order of magnitude faster the pendulum frequency, but the period is less than half of the numerical integration time step size.

This same set of equations applies to the pitching particle model, with T and θ added to the state vector, and the following modifications to equation (10):

$$\ddot{\vec{r}}_P = T\vec{\tau} \tag{11}$$

$$\vec{\tau} = -\begin{Bmatrix} \sin\theta \\ \cos\theta \end{Bmatrix} \tag{12}$$

$$\begin{Bmatrix} \dot{T} \\ \dot{\theta} \end{Bmatrix} = \vec{u} \tag{13}$$

Note that these models are generic enough to be adapted to three dimensions, with only changes to the definition of $\vec{\tau}$ by including a second orientation parameter to the state (in this case, probably heading) and a corresponding control input.

Task and Test Conditions

The guidance system will be presented with a task which mimics the Christmas tree harvesting task pictured in Figure 1. Specifically, the initial condition will have the aircraft and load in a stable hover 300 feet away from the goal configuration. The goal configuration will specify a load position corresponding to where the load could drop straight down into a hypothetical truck bed (Figure 2.) The load should have near-zero velocity at this point, enabling the aircraft to release the load for vertical drop.

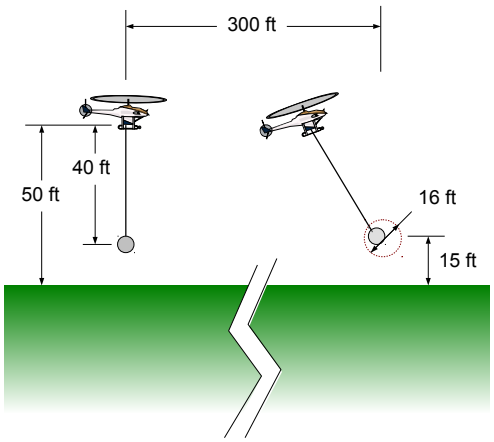

Fig. 2. Diagram of planning task.

One characteristic of RRTs is that the parameters of the planner, in particular the cost function and/or distance metric, must be carefully selected in order to reward the desired aircraft behavior. This cost function will generally vary based on the specific task. For the experiments described here, a cost function was designed to strongly penalize position error and ignore velocity error when the aircraft is far from the goal, while nearly ignoring position error near the center of the target and heavily penalizing load speed (Eq. 14, 15). This is accomplished by selecting the weighting parameter, α, based on the distance to goal. For the work done here, the constant C is selected to be 2.

$$\alpha = C^{\left(-\left\|\vec{r}_{M,goal} - \vec{r}_M\right\| / l_0\right)} \tag{14}$$

$$J = (1-\alpha)\left\|\vec{r}_{M,goal} - \vec{r}_M\right\|^2 + \alpha\left\|\dot{\vec{r}}_{M,goal} - \dot{\vec{r}}_M\right\|^2 \tag{15}$$

Metrics

The quality of a planned trajectory can measured a number of ways. Here, we will use planned time of flight and the final value of the cost function as a measure of overall optimality. Because of the stochastic and non-optimum seeking characteristics of the flavor of RRT used, performance will be presented in aggregate as distributions of results from a series of Monte Carlo experiments. Additionally, the final tree, selected trajectory, and "cost to go" history will be presented for both the shortest returned trajectory and a randomly-selected "typical" one. Average computation time will be presented as well; however, note that an implementation of this software for flight would be orders of magnitude faster, and so computation time is included for comparison between different models only.

Table 1. Dynamic System Paramters.

Parameter	Value
Line Length	40 ft
Load Mass	20 lbm
Load Equiv. Flat Plate Area	1 ft^2
Air Density	0.002378 slug/ ft^3

Table 2. Planner Parameters.

Parameter	Value
Number of Nodes	1000
Terminal Condition	Cost < 0.001
Control Space (linear)	$[-X,0,X]$, $X \sim U(0,3)\ ft/s^2$
Control Space (angular)	$[-Y,0,Y]$, $Y \sim U(0,1.4)\ rad/s$
Branch Length	$T \sim U(0,3)\ s$
Integration Time Step	$0.05\ s$
Goal Bias	0.1
Sampling Domain	
Load Position	$\vec{R}_M \sim [U(-20,320), U(0,100)]^T\ ft$
Load Velocity	$\dot{\vec{R}}_M \sim [U(-50,50), U(-10,50)]^T\ ft$

SIMULATION RESULTS AND DISCUSSION

Model and Test Parameters

In order to prepare for porting of this planning algorithm for flight test, the parameters of the system and planner were selected to mimic the GTMax guidance, navigation, and control research testbed (Ref. 20). The parameters used in this work are included in Tables 1 and 2.

The simulations were carried out 100 times for each model in Matlab running under Ubuntu 12.04 LTS (32-bit) on an 8-core Intel Xeon 3.20 GHz CPU with 8 GB of memory.

Constant-Altitude Double Integrator

The constant altitude double integrator performed somewhat well, presumably due to the fact that the starting altitude of the load was very close to the target altitude. The average trajectory duration was about 44 seconds (Figure 3), an average speed of 6.8 ft/s, which is quite docile for this scale aircraft. This model, however, was the only one that could consistently put the load inside the target at low speed—typically within 6 ft of the target and less than 2 ft/s. The fastest trajectory was completed in just under 23 seconds, which is an average of 13 ft/s, a significantly faster clip. The trees of the fastest and "typical" solutions are a bit distorted due to the limited vertical separation of the branches (Figures 4 and 7). It is notable that while the RRT does not produce an optimal flight path, it did produce in nearly every case a trajectory in which the cost function monotonically decreased from start to finish. In other words, the trajectory solution will constantly drive the system toward the goal configuration (Figure 6).

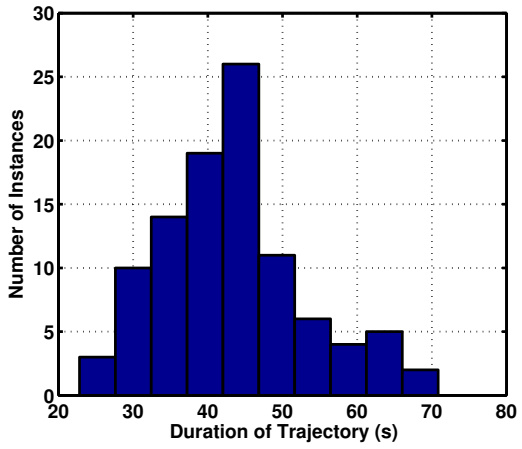

Fig. 3. Distribution of flight times for returned trajectory, constant altitude model.

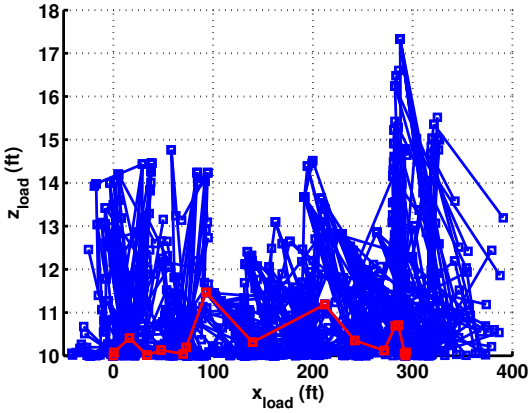

Fig. 4. Final RRT for best-time trajectory, constant altitude model. Tree in blue; final selected branch in red.

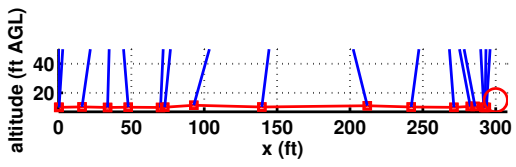

Fig. 5. System trajectory for best-time solution, constant altitude model. Load position and path in red; selected line configurations in blue.

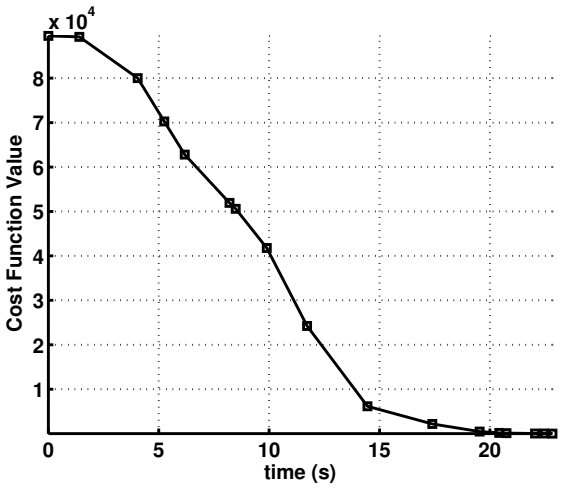

Fig. 6. "Cost-to-go" history for best-time trajectory, constant altitude model.

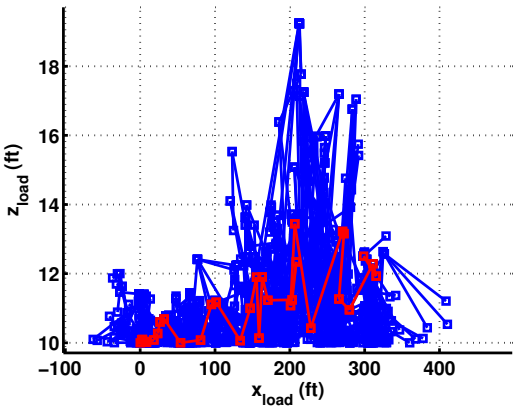

Fig. 7. Final RRT for a typical trajectory, constant altitude model. Tree in blue; final selected branch in red.

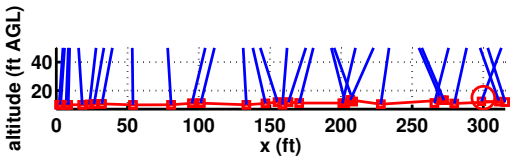

Fig. 8. System trajectory for a typical solution, constant altitude model. Load position and path in red; selected line configurations in blue.

Two-Degree-of-Freedom Double Integrator

The double integrator generated trajectories which were both more aggressive and less able to precisely drop the load in the target location. The average time of flight was about 36 seconds, a little more than 8 ft/s. However, the load was dropped further away, an average of 10 ft, and while moving faster, an average of 7 ft/s.

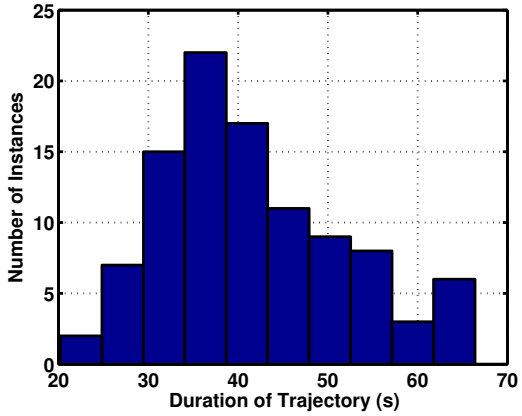

Fig. 9. Distribution of flight times for returned trajectory, 2DOF model.

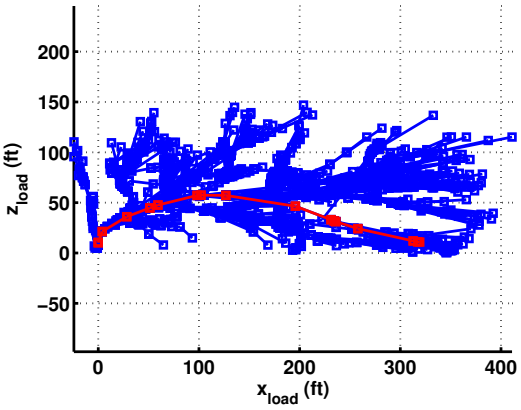

Fig. 10. Final RRT for best-time trajectory, 2DOF model. Tree in blue; final selected branch in red.

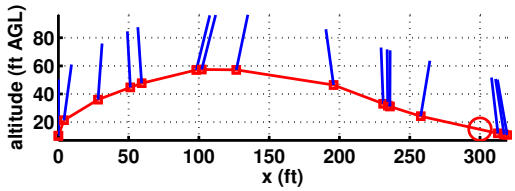

Fig. 11. System trajectory for best-time solution, 2DOF model. Load position and path in red; selected line configurations in blue.

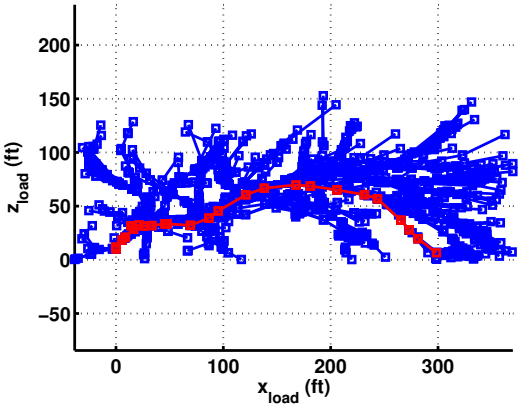

Fig. 12. Final RRT for a typical trajectory, 2DOF model. Tree in blue; final selected branch in red.

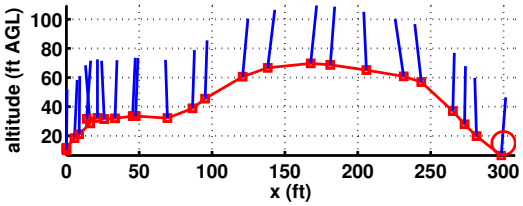

Fig. 13. System trajectory for a typical solution, 2DOF model. Load position and path in red; selected line configurations in blue.

Pitching Particle

The pitching particle model produced the fastest trajectories, but resulted in even less precision hitting the goal state. The average flight time was about 25 seconds, a speed of 12 ft/s, but the load missed the target by an average of 25 feet, and was moving at an average of 20 ft/s at the trajectory end.

Two things were notable about the performance of this model that were distinct from the other two models. First, the best-time trajectory looks amazingly similar to the trajectories flown by human pilots in Christmas-tree harvesting (Figure 16). Though the human pilots are not necessarily optimal, through training and repetition they have become very efficient and it is promising that the planner here is finding similar trajectories. Second, since this model is more representative of the actual dynamics of a helicopter, it gives some intuition about the difficulty of this task.

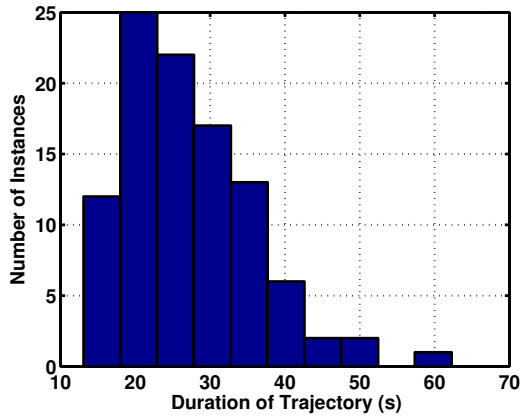

Fig. 14. Distribution of flight times for returned trajectory, pitching particle model.

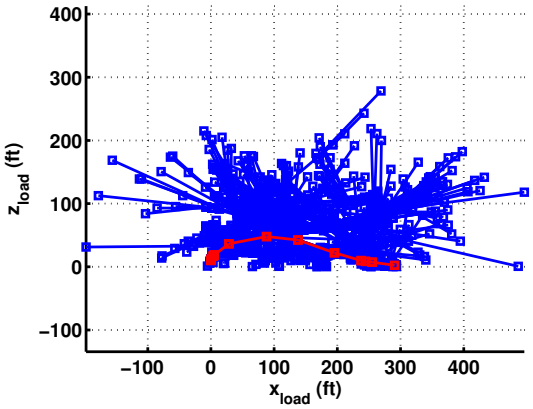

Fig. 15. Final RRT for best-time trajectory, pitching particle model. Tree in blue; final selected branch in red.

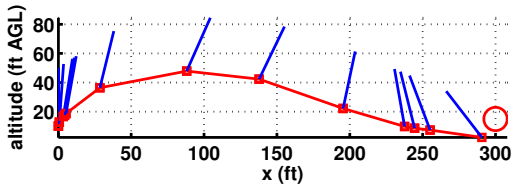

Fig. 16. System trajectory for best-time solution, pitching particle model. Load position and path in red; selected line configurations in blue.

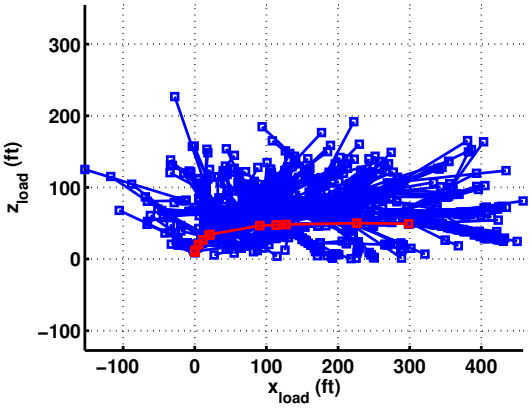

Fig. 17. Final RRT for a typical trajectory, pitching particle model. Tree in blue; final selected branch in red.

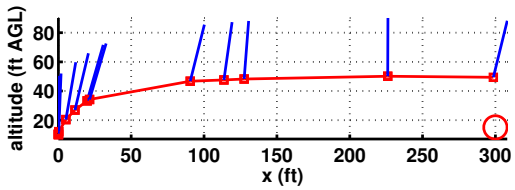

Fig. 18. System trajectory for a typical solution, pitching particle model. Load position and path in red; selected line configurations in blue.

Discussion of Results

The results shown above indicate that this method is not quite ready for implementation in a live system, but does show some promise. The primary problem is one that affects many systems that use RRTs, namely that there is significant excess motion. Additionally, the lack of precision with the more aggressive models would be problematic for practical applications. However, the approach still shows some promise.

First, while the RRT approach has not performed well for the relatively simple task shown here, it may yet perform well in more complex scenarios. RRTs have been shown in the past to handle obstacles, constraints, and dimensionality with little degradation of performance.

Second, the modular nature of RRTs mean that performance can be improved via a number of adjustments or modifications. The performance of the algorithm is highly sensitive to the selection of cost function, so a different choice could improve the results. The control input selected for each branch is uniformly random here, but could instead be selected to drive the system in a certain direction. Other flavors of RRT could mean a great deal of improvement; for example, RRT-Connect (Ref. 21) could provide more precision at the end of the trajectory, and RRT* (Ref. 19) could eliminate excess motion and produce optimal trajectories. Finally, the number of computations per tree scale at $O(N^2)$, where N is the number of nodes in the tree. Any ways to reduce computation could allow a deeper search of the solution space, with possibly a better result. Finally, reducing the size of the tree and re-running the algorithm multiple times may produce a better solution.

Computation time between the three models did not vary drastically—most trees took between 50 and 60 seconds to compute. The greatest computational cost was in nearestNeighbor lookups (see Algorithm 1). In the work shown here, this lookup was performed by exhaustive search of all the nodes in the tree. A faster approach could be the use of an index that groups nodes by location—such a system would only search a subset of near nodes to find the nearest. A more radical approach would be to employ a tree method that does not require nearest neighbor lookups; for example, kinematic trees (Ref. 22) or Monte Carlo tree search (Ref. 23).

CONCLUSIONS

The work so far has pointed to several conclusions about the use of RRTs for helicopter-slungload guidance. First, RRTs are likely a viable method for motion planning of the system, though work remains to fine-tune parameters and subelements of the method. In particular, the method seems to be sensitive to the cost function, and tends to produce excess motion and variable precision. Future research will investigate modifications to these elements, as well as different planning models or possibly entirely different tree-based methods.

REFERENCES

[1]LaValle, S. M., *Planning Algorithms*, Cambridge University Press, Cambridge, 2006.
doi: 10.1017/CBO9780511546877

[2]Glauert, H., "The Stability of a Body Towed by a Light Wire, R&M 1312," Technical report, Aeronautical Research Committee, Farnsborough, England, February 1930.

[3]Poli, C. and Cromack, D., "Dynamics of Slung Bodies Using a Single-Point Suspension System," *AIAA Journal of Aircraft*, 1972, pp. 80–87.

[4]Feaster, L., Kirchhoff, R., and Poli, C., "Dynamics of a slung load," *Journal of Aircraft*, Vol. 14, (2), February 1977, pp. 115–121.
doi: 10.2514/3.44578

[5]Asseo, S. J. and Whitbeck, R. F., "Control Requirements for Sling-Load Stabilization in Heavy Lift Helicopters," *Journal of the American Helicopter Society*, Vol. 18, (3), 1973, pp. 23–31.

[6]Hutto, A. J., "Qualitative Report on Flight Test of a Two-Point External Load Suspension System," *AHS Preprint 473*, January 1970.

[7]Micale, E. C. and Poli, C., "Dynamics of Slung Bodies Utilizing a Rotating Wheel for Stability," *Journal of Aircraft*, Vol. 10, (12), 1973, pp. 760–763.

[8]Ivler, C. M., Powell, J. D., Tischler, M. B., and Fletcher, J. W., "Design and Flight Test of a Cable Angle / Rate Feedback Flight Control System for the RASCAL JUH-60 Helicopter," American Helicopter Society Annual Forum, 2012.

[9]Ivler, C. M. and Tischler, M. B., "Case Studies of System Identification Modeling for Flight Control Design," *Journal of the American Helicopter Society*, Vol. 58, (1), January 2013, pp. 1–16.
doi: 10.4050/JAHS.58.012003

[10]Bisgaard, M., Bendtsen, J. D., and Cour-harbo, A., "Modelling of Generic Slung Load System," Proceedings of AIAA Modeling and Simulation Technologies Conference, 2006.

[11]Bisgaard, M., Cour-Harbo, A., Johnson, E. N., and Bendtsen, J. D., "Vision aided state estimator for helicopter slung load system," *Sevententh IFAC Symposium on Automatic Control in Aerospace*, 2007.

[12]Bisgaard, M., Cour-harbo, A., and Bendtsen, J. D., "Input Shaping for Helicopter Slung Load Swing Reduction," Paper August, AIAA Guidance, Navigation and Control Conference and Exhibit, 2008.

[13]Ottander, J. A. and Johnson, E. N., "Precision Slung Cargo Delivery onto a Moving Platform," Proceedings of AIAA Modeling and Simulation Technologies Conference, 2010.

[14]Cour-harbo, A. and Bisgaard, M., "State-Control Trajectory Generation for Helicopter Slung Load System using Optimal Control," Paper August, AIAA Guidance, Navigation and Control Conference and Exhibit, 2009.

[15]LaValle, S. M., "Rapidly-exploring random trees: A new tool for path planning," Technical report, Computer Science Department, Iowa State University, 1998.

[16]LaValle, S. M. and Kuffner, J. J. J., "Rapidly-exploring random trees: Progress and prospects," Workshop on the Algorithmic Foundations of Robotics, 2000.

[17]LaValle, S. M. and Kuffner, J. J. J., "Randomized kinodynamic planning," *The International Journal of Robotics Research*, Vol. 20, (5), 2001, pp. 378–400.

[18]LaValle, S. M., "Resolution complete rapidly-exploring random trees," *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No.02CH37292)*, Vol. 1, (May), 2002, pp. 267–272.
doi: 10.1109/ROBOT.2002.1013372

[19]Karaman, S. and Frazzoli, E., "Optimal kinodynamic motion planning using incremental sampling-based methods," Proceedings of the IEEE Conference on Decision and Control, 2010.

[20]Johnson, E. N. and Schrage, D. P., "The Georgia Tech Unmanned Aerial Research Vehicle: GTMax," *Guidance, Navigation, and Control*, 2003.

[21]Kuffner Jr, J. J. and LaValle, S. M., "RRT-connect: An efficient approach to single-query path planning," Paper Icra, IEEE International Conference on Robotics and Automation, 2000.

[22]Langelaan, J. W., "Tree-based trajectory planning to exploit atmospheric energy," American Control Conference, June 2008.
doi: 10.1109/ACC.2008.4586839

[23]Browne, C., Powley, E., Whitehouse, D., Lucas, S., Cowling, P. I., Rohlfshagen, P., Tavener, S., Perez, D., Samothrakis, S., and Colton, S., "A Survey of Monte Carlo Tree Search Methods," *IEEE Transactions on Computational Intelligence and AI in Games*, Vol. 4, (1), 2012, pp. 1–49.