A path planning approach to noise-minimal rotoracft landing trajectories

Robert A. Morris
Ames Research Center
robert.a.morris@nasa.gov

Matthew Johnson
Institute for
Human and Machine Cognition
mjohnson@ihmc.us

K. Brent Venable
Tulane University
Florida Instittue for
Human and Machine Cognition
kvenabl@tulane.edu

James Lindsey
Monterey Technologies
james.e.lindsay@nasa.gov

Abstract

NASA and the international aviation community are investing in the development of a commercial transportation infrastructure that includes the increased use of rotorcraft, specifically helicopters and civil tilt rotors. However, there is significant concern over the impact of noise on the communities surrounding the transportation facilities. In this paper we address the rotorcraft noise problem by exploiting a powerful search technique coming from artificial intelligence coupled with simulation to design low-noise flight profiles which can be tested in through field tests. In particular, this paper investigates the use of simulation based on predictive physical models, combined with the A* path planning algorithm to design for low-noise approach trajectories for rotorcraft. Novel features of our approach include the use a discrete search space with a resolution that can be varied, and the coupling of search with a robust simulator to evaluate candidates.

Introduction

The problem of designing low noise flight profiles can be viewed as a trajectory optimization problem (TOP) (LaValle 2006). Informally, a TOP consists of a set of *states*, a vector of *control decisions*, a start and goal state, a cost function, and a set of constraints. A state represents locations (i.e. points in a 3D space), velocity and heading. A control decision is a vector representing change in velocity, altitude, heading, and in turn radius. The TOP can be stated informally as follows: *given a set of states and control actions, find a sequence of actions (trajectory) that minimizes a cost function subject to a set of dynamic constraints, and constraints on start- or end-states.*

The objectives of this paper are to formulate a 3D trajectory noise optimization problem, and to investigate the effectiveness of a path planning solution method for the problem using discrete heuristic search. The problem formulation is based on a model that is comprised of a tunable discretization of the state and control-space; a constraint-based dynamics model of the rotorcraft; a restriction of the solution space to a class of standard approach patterns; a set of constraints related to pilot procedure and passenger safety and comfort; and a noise cost function that aggregates noise intensities predicted by a ground noise simulator.

The Study of Noise

Helicopter noise sources include the main rotor, the tail rotor, the engine(s), and the drive systems. The most noticeable acoustical property of helicopters is referred to as BVI (Blade Vortex Interaction) noise. This impulsive noise occurs during high-speed forward flight as a result of blade thickness and compressible flow on the advancing blade. This causes the blades airloads to fluctuate rapidly and results in impulsive noise with shock waves that can propagate forward. At lower airspeeds, and typically during a descent, BVI can occur when a blade intersects its own vortex system or that of another blade (Fly 2009).

One of the most common noise measures is the *Sound Exposure Level (SEL)*. *SEL* summarizes the variable energy level of an event with arbitrary duration by mapping it to an event of one second duration with the same overall energy and a constant energy level. *SEL* provides a comprehensive way to describe noise events for use in modeling and comparing noise environments. The average SEL value over the plane is called the SEL average (SEL_{av}). One equation for SEL_{av} (Goplan et al. 2003) is

$$SEL_{av} = 10log_{10}\Sigma_n(\Sigma_i 10^{SPL_{dB,i,n}/10}\Delta t_{i,n}/T_0)\Delta A_n/A_0 \quad [1]$$

Here, n ranges over the locations on the ground plane and i refers to path elements. $SPL_{db,i,n}$ refers to the *Sound Pressure Level* in decibals for a location and a path element, and the Δs are elemental ground- or trajectory elements. A_0 refers to the area of the ground plane and T_0 is a reference interval of one second.

A procedure called *Q-SAM* (Quasi-Static Acoustic Mapping) can be used to compute the BVI noise for a rotorcraft in slowly maneuvering flight. (Sim, Schmitz, and Gopalan 2002). Using Q-SAM one can specify a flight path (in terms of acceleration and flight path angle) time history and compute the effective SEL at any location along the path. The result is usually displayed as a contour plot (Figure 1) over a ground plane. Each color corresponds to a dB level (redder and lighter colors noisier).

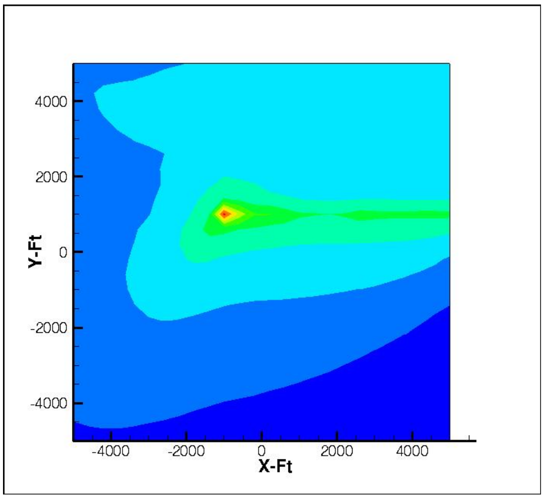

Figure 1: A Noise Contour Plot.

Rotorcraft Noise Model Simulation Tool

The Rotorcraft Noise Model (RNM) (Conner, Burley, and Smith 2006) is a simulation program that predicts how the sound of a rotorcraft will propagate through the atmosphere and accumulate on the ground. The core of the RNM method is a database of vehicle source noises defined as *sound spheres*. Spheres are obtained through measured test data or through models. The spheres allow for a representation of the 3D noise directivity patterns associated with the operating rotorcraft. A sphere is associated with one noise source and one flight condition (flight path angle, nacelle angle (for tilt-rotors) and airspeed). Each sphere represents constant airspeed conditions for a given flight path angle. During simulation, RNM performs an ordering of the spheres based on similarity with the flight conditions along the input path. The sound source properties are extracted from the sphere database using a linear interpolation of both required speed and flight path angle (Page, Wilmer, and Plotkin 2007).

The input to RNM consists of a set of computational parameters, including identity of rotorcraft, and the dimensions and resolution of a grid that will display output noise; a specification of points of interest; and a specification of the flight trajectory, including position, velocity and orientation. The input data are interpolated for a user-specified time spacing using a trajectory model, producing a description of the ground track of the rotorcraft, as well as the operating state of the aircraft (velocity, flight path angle), as well as the vehicle orientation (heading and yaw). The flight track is passed into a propagation module which constructs the sound profile at each ground position as a function of time. The main propagation loop advances point by point through the flight path and, for each vehicle position, the individual noise sources are propagated independently. The propagation model expresses the sound level at a distance from the source as a sum of the sound at the source with other factors such as the atmospheric absorption, ground reflection and wind effects. RNM also integrates the Q-SAM method for modeling longitudinal acceleration and deceleration. Consequently, RNM is capable of calculating cumulative noise exposures such as A-weighted SEL (see (Page, Wilmer, and Plotkin 2007) for more details).

Related Work

Trajectory Optimization is a vast and diverse research field that had its modern origins in the exploration of space, specifically, in the design of spacecraft trajectories. Methods for solving Trajectory Optimization Problems are based either on methods of optimal control or on approximations to optimal control problems based on non-linear programming methods (NLP) (Betts 1998).

Recently, direct sample methods have evolved as the optimization algorithms of choice. These algorithms may require orders of magnitude increases in the number of functional samples but exhibit robustness to non-smoothness in the trajectory model. Examples include: genetic algorithms, stochastic sampling methods, and hill climbing algorithms. Other methods of trajectory design and optimization address the challenges of high-dimensional, non-linear systems by using randomized path-planning methods. To apply any of these methods, the state- or control-space must be discretized, and many different methods of discretization have emerged. Examples of recent trends include Rapidly expanding Random Trees (RRT) (Cheng and Lavalle 2002) and Probabilistic Road Maps (Kavraki, Kolountzakis, and Latombe 1998),(Pettersson and Doherty 2004). Finally, of relevance to our work are heuristic grid-based path planners such as A* and D*, and the problem of finding smooth paths in grids (Ferguson and Stentz 2005).

Much of the previous work on ground noise minimization for Rotorcraft uses sampling-based approaches for optimization. The work described in (Padula et al. 2009) uses a genetic algorithm to search a space of in 2D (X-Z, no turns) trajectories. It employs a set of trajectory design variables based on pre-compiled low-noise strategies employed in practice by pilots, in order to reduce the dimensionality of the search space.

Another discretization of the search space employs cell decomposition (in 2D space) (Atkins and Xue 2004) and k-ary tree structuring (for 3D modeling) (Xue and Atkins 2006a). These approaches allow for the handling of obstacles that may be due to rotorcraft landing environments that intersect with fixed-wing landing corridors. The result of cell decomposition is a connectivity graph which is then searched by using a uniform cost search (i.e., A* with h =0).

The TOP model that we use in this paper is inspired by the work in (Morris et al. 2012). In particular, we use the same parameters and constraints but we modify the cost function in order to incorporate the information on human annoyance level. We also adopt a completely different solving approach, since in (Morris et al. 2012) an incomplete local search approach was presented, while here we also consider a complete path planning technique.

Problem Formulation

In this section we transform the TOP into a discrete optimal planning problem following the work presented in (Morris et al. 2012). This involves defining a discretization of the world, the state of the rotorcraft, and the set of actions available to the pilot; defining a set of constraints on the discrete space; and defining a cost function for evaluating and ordering trajectories.

Model parameters

We employ a standard approach to discrete motion planning by introducing a grid of points in 3D space. This discretization is virtually identical to parameters specified in RNM input files discussed above. We define a size and resolution of the grid in terms of the lower left and upper right values of (X, Y), and the distance (in feet) between grid points, respectively. Altitude (Z) will have the range $[0, zmax]$, where $zmax$ is defined by the starting point of the problem (in these experiments, between 1000 and 1500 feet). Typically, but not necessarily, the grid will define the space for which we're interested in measuring ground noise.

We define a state as a 5-tuple of values from the variables $S = (X, Y, Z, V, H)$ consisting of location in 3-space, velocity and heading. Similarly, an action (control) vector is a 5-tuple of values from the variables $U = (\Delta X, \Delta Y, \Delta Z, \Delta V, \Delta H)$ defined as changes to position, velocity and heading. Given a state s, let $s(X)$ ($s(Y)$, etc.) be the value of s for state variable X, etc. A path (trajectory) is a sequence $s_0, u_0, s_1, u_1, \ldots, s_t, u_t, s_t$ of state and action pairs, based on a transition function $F : S \times U \rightarrow S$, which captures the dynamical and other constraints governing rotorcraft control. A feasible path based on F is one in which for all $i = 1 \ldots t-1$, $s_{k+1} = F(s_k, u_k)$.

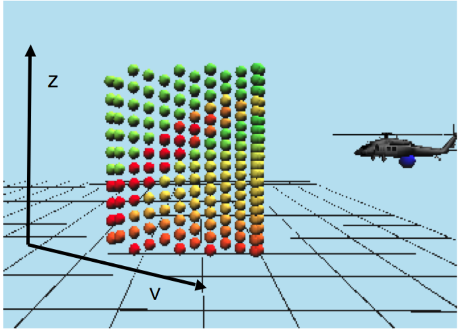

Figure 2: Visualizing search resolution. The rotorcraft is at the state marked by the blue dot. The feasible states of the next state of the rotorcraft are marked by the colored balls. The colors reflect the predicted noise produced by that action. Note that this control space is not to be confused by the state space, which is 4D space of X,Y,Z and V.

Constraint Model

The constraint model defined here arises from considerations of safety and passenger comfort, defined by pilot preference and standard procedure, as well as rotorcraft dynamics. For the experiments here, which focus on approach and landing, we allow only deceleration and only descent (although this constraint is by no means imposed by the approach we take, and can easily be removed). There are three kinds of constraints:

- on initial and final state, s_0, s_t;
- boundary constraints on v and z; and
- constraints on flight path angle and rate of deceleration.

Based on standard procedure, we limit the initial state $s_0(V)$ to range from 100 to 135kts and for $s(Z)$ to range from 1000 to 1500 feet. The final state s_t is constrained in the obvious way so that $S(V) = s(Z) = 0$. Second, there are a minimal velocity and altitude (v_{mini}, z_{mini}) that a rotorcraft must have when starting the final part of the approach (that is at the so-called landing decision point).

The third class of constraints state that any part of a trajectory will be be characterized by an angle of descent $\gamma \in [0^o, 12^o]$ and a deceleration $a \in [0g, 0.1g]$ (or $a \in [40ft/sec^2, 201ft/sec^2]$). Such restrictions induce constraints on the change of velocity and altitude as follows. Given a pair of nodes N_i, N_j and a path between them of distance $dist_{ij}$ we have:

- the deceleration constraint (dec):

  $$\Delta v_i \in \{\delta_v \| \exists a \in [0, 0.1], \delta_v = \sqrt{v_i^2 + 2a \times dist_{ij}} - v_i\},$$

  where a is expressed in gs.
- the angle-of-decent constraint (aod):

  $$\Delta z_i \in \{\delta_z \| \exists \gamma \in [0^o, 12^o], tan(\gamma) = \frac{\delta_z}{dist_{ij}}\}$$

  .

A trajectory is said to be *flyable* if it satisfies all three kinds of constraints just defined for the model.

The constraint model induces a transition function F and a definition of the set of feasible paths based on F. To apply F in a discrete search setting, a further discretization of the control space U is required. The result of the discretization is called the *search resolution* and is visualized in Figure 2. The search resolution defines a two-dimensional space of discrete altitudes and velocities that the rotorcraft can obtain at each state of a trajectory, given the upper and lower bounds defined by the boundary constraints. Each ball in the figure thus represents a pair $\langle v, z \rangle$ of feasible velocity and altitude values in a transition from some current state. This space of *feasible transitions* is used during the search. Notice that each state (ball) is colored to indicate the predicted impact of choosing the state on the overall noise profile (redder means noisier). In this way, as we'll discuss later in more detail, the feasible transitions can be ordered to guide the search to the best solution.

Summarizing, a *Trajectory Noise Optimization Problem* (TNOP) is a tuple $\langle S, D, s_0, s_f, aod, dec, v_{mini}, z_{mini} \rangle$, where S is a set of states, D a set of decisions, s_0, s_f are initial and final states, aod, dec are deceleration and descent angle constraints, and v_{mini} and z_{mini} are as just defined. A feasible solution to a TNOP is a flyable path $P = s_0, u_0, s_1, u_1 \ldots, s_k$.

As a final restriction on the size of this space for the work described in this paper, we limit the X, Y values to define a pattern that is considered 'standard' by pilots, involving a downwind leg and two turns. The standard approach we use we call a 'box' pattern, such as the one shown in Figure 3. Consequently, the search space is limited here to two actions: change in V and change in Z.

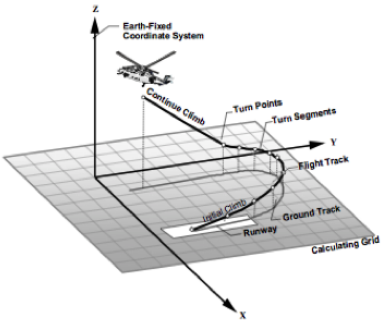

Figure 3: A standard approach pattern (reprinted from (Page, Wilmer, and Plotkin 2007)).

Cost Function

In (Morris et al. 2012) a scalar cost function that summarizes the data in a contour map such as the one in Figure 1 was presented. Here we redefine noise ranges so that they roughly correspond to different levels of annoyance (low, medium, high, very high). Then, for each trajectory and associated contour map, we count the number of dB values that fall into each range, and take the weighed sum as the noise score for that trajectory (lower scores better).

To assign reasonable weights we draw upon previous human factors research on noise tolerance (on Noise 1992), which plotted Day-Night Average Sound Level in dB against the percentages of human subjects that classified that level as annoying. From this we established a weight distribution for our cost function.

Formally, we define a *Binning Heuristic function* (BIN) as follows. Given in input a solution t, RNM computes the A-weighted SEL value for each of the grid points. Let us denote with $SEL(t, x, y)$ such a value for the grid point (x, y) given trajectory t. We define a sequence of decreasing ranges, $\langle r_1, r_2, \ldots, r_n \rangle$ partitioning the SEL values of the grid points. Given a trajectory t let us denote by $S_i(t) = \{(x, y) | SEL(t, x, y) \in r_i\}$. We define the following vector $b(t) = \langle b_1(t), b_2(t), \ldots, b_n(t) \rangle$ where $b_i(t) = |S_i(t)|$. The BIN-score of solution t is $BIN(t) = \Sigma_{i=1 \ldots n} w_i b_i(t)$ where w_i is the weight associated to the i-th bin, $w_i > w_{i+1}$ and $\Sigma_{i=1,\ldots,n} w_i = 1$. Thus a solution that assigns lower levels of noise to larger regions of the grid is to be preferred. Weights are used to penalize the presence of, even small, extremely noisy regions. The goal will be that of minimizing the BIN value.

Optimization Approach

We employ a two-phase architecture, consisting broadly of a search step phase and an evaluation phase. The output of the search step phase is a trajectory P, which becomes input to the evaluation phase. The output of the evaluation phase is a score for P. The evaluation phase components are the RNM simulator, which produces a contour map, and the cost function, which calculates the BIN score from the contour map, as defined above. For optimizers we choose complete search using A* and incomplete search using Stochastic Local Search (SLS). With A*, the search step phase produces an expansion of a partial plan; for SLS the same phase always produces a complete trajectory as output.

A* Heuristic for TNOP A* is complete and optimal provided the heuristic function h never overestimates the cost of achieving the goal through s. A common cost function design method for A* is to find a relaxation of the problem to be solved, and to define a heuristic that is a perfect predictor of the relaxed problem; the same heuristic is also admissible for the original problem. The approach used here is a heuristic in which the estimated cost for a node is the cost of a trajectory in which altitude is maintained until the goal, but airspeed is reduced as fast as possible to the minimum velocity ($vmin$). This *fly high and slow* heuristic has been empirically confirmed to be admissible.

Non-factorability of the BIN cost A* requires a way of aggregating a path-cost g as a solution is generated, where $g(x)$ is the cost from an initial state of the solution to x. Given two states, x, y, an edge distance $d(x, y)$ is defined; the aggregate cost is $g(y) = g(x) + d(x, y)$. Unfortunately, aggregating the cost for noise is problematic, because the cost is not a simple sum. It is rather, as indicated by [1], an average of the noise over time.

A simple example illustrates the aggregation problem for cost functions based on average. Let A and B be paths that are adjacent in the sense that they share a common state such that their join, AB is a feasible path. Let $BIN(A)$ be the BIN cost of trajectory A. Let A and B have durations $dur(A)$ and $dur(B)$. Assume at 1 second intervals noise from the source is propagated to the ground. Following equation (1) above, assume that the noise is computed at each data point by a simple averaging the noise measurements over the duration of the path. Thus, for path A, the noise level assigned is the sum $\Sigma_i n_i$ divided by the duration of A, similarly for B. Suppose we join the two paths into AB. Then to compute the noise level for each data point, we again sum over all the individual measurements and divide by the duration of AB, which by definition is $dur(A) + dur(B)$. In general, however, $BIN(AB) \neq BIN(A) + BIN(B)$, because the least common multiple of two (non-zero) integers is never equal to their sum.

As a consequence, when a path $P = S_1 \ldots S_k$ is expanded by adding a new state S_{k+1}, A* does not compute the cost $BIN(S_k) + BIN(S_k, S_{k+1})$. Instead, we compute the cost $BIN(S_1 S_2 \ldots S_{k+1})$, i.e., we recompute the noise cost from the start state to the new state. This improves the accuracy of the 'cost to come' g, but it does not completely solve the aggregation problem. In general, what is required is a 'factorability' property, in the sense that the contribution made by a segment of a path P should not change if P is expanded. But again this property clearly does not hold. For example, suppose a ground location ΔL is directly underneath the rotorcraft at the beginning of some trajectory, and the rotorcraft moves further and further away from ΔL as it proceeds along the trajectory. Then each noise measurement taken at ΔL will diminish over time, and so the overall time-averaged noise for ΔL will be less for the complete trajectory than for the initial segment. On the other hand, other locations at different places along the path will undergo different changes in how they contribute to the noise of the whole path. Consequently, in general, *one only knows for certain the contribution made by a specific location to the overall noise of a given path by evaluating the entire path.* In other words, noise is not factorable in space or time. Despite this result, in the experiments reported below we employ the cost function in which we assume factorability; in effect we assume that the noise cost contributed by an individual location ΔL for an arbitrary segment of any path is the same as its noise cost for the entire path. Although not true in general, experiments with the approximate cost function have assured us that the version of A* does produce a solution that is close to optimal, and enables the sort of cost aggregation required by the algorithm.

Stochastic Local Search

In order to better assess the performance of our implementation of A^* we have implemented a slightly modified version of the stochastic local search approach (SLS) described in (Morris et al. 2012).

The parameters to this algorithm are the objective function, and a value MAXTRIES that indicate how many steps are allowed before termination. SLS iteratively takes search steps using a neighborhood function.

The neighborhood function works as follows. Given the set of states $s_0, s_1, \ldots, s_t$ of the current path, let again $F(s_i)_p$ be the set of feasible states for s_i in p, as described above. This set is determined by the constraints on flight path angle and deceleration relating s_{i-1}, s_i, and s_{i+1} (with different conditions defined at the path boundaries). Let $Neighbors(p) = \bigcup_{1 \leq i \leq t} F(s_i)_p$. Thus each neighbor p' of p can be expressed as a triple $\langle i, v, z \rangle$, such that p' is the result of replacing s_i in p by a state s'_i with $s'_i(V) = v$ and $s'_i(Z) = z$.

A neighborhood function takes a feasible solution and makes small changes to it, resulting in a new feasible solution. There are two kinds of steps, greedy and noisy, which are decided by a third parameter, called greedy [1]. SLS keeps track of a current estimate of the best solution p^*, and its cost function score, as well as a current solution p, and its score. In an initialization step, an arbitrary path p_{init} is generated.

A greedy step is one that enumerates over all the neighbors of the current solution p until the first solution p' is found that improves on the score of current solution (this makes our version of SLS an instance of so-called hill-climbing search). If no such improvement exists, the current path is set to the last path examined. A noisy step, on the other hand, randomly chooses a neighbor of p and it becomes the current path. SLS also has a restart condition, which happens in the case that a current state has no feasible neighbors. SLS restarts by resetting the current path to p_{init}. The purpose of the greedy parameter, as described in detail in (Mengshoel 2008), is to better understand the difficulty of the problem space. For example, adding randomness can improve the performance of SLS on problem spaces with many 'traps' in local optima.

Experiments

The main purpose of the experiments reported here is to assess the performance of the solution method based on A* and to compare it against local search. The performance criteria are solution quality (BIN score) and run time. We are restricted here to the 'box pattern' described above. By making the comparison we hope to reveal how close SLS can get to the optimal solution when A* is able to generate a solution, and also what classes of TNOP problems (defined in terms of search or grid resolution) SLS can solve that A* cannot. Since A* is deterministic and SLS is not, we compare an average performance on SLS, but not on A*. For these experiments we average over 10 runs of SLS.

In these experiments, we placed a setting of 55 on the depth of SLS search, and set the SLS 'greedy' parameter to .5, which means that on average, SLS will make a greedy step half the time, and a random move half the time. In future reports, we will present data summarizing the effects of varying these parameters (search depth and degree of randomness) on SLS performance.

As discussed earlier, we have partitioned the discretization of the problem into search resolution and grid resolution. In Figure 4 we show three plots, each for a different setting of search resolution (4X4, 5X5, 6X6). The X axis shows variable settings of grid resolution (700-1000). Symmetrically, Figure 5 shows three plots, each for a different setting of grid resolution (800, 900,1000). The X axis shows variable settings of search resolution (from 4X3=12,4X4,5X4,5X5, 6X5, 6X6, to 7X6=42). The left plot documents time to solution, with a cutoff point of 21,600K seconds (6 hours); the right plot documents BIN score (the score is arbitrarily set to 0 when an algorithm times out) . [2]

The results allow us to draw the following conclusions (some of which are preliminary and require more experi-

[1] We could have called it 'noisy' but that might cause some confusion given the problem domain.

[2] Keep on mind therefore that lower grid number means higher grid resolution, whereas *higher* search number means higher search resolution. The toughest problems are therefore with high grid resolution (lower number) and high search resolution (higher number).

ments to verify). First, A* times out on a highest setting of search resolution (7X7) for virtually all grid resolutions examined (an exception being seen on the bottom right panel of Figure 4 where the spike at 900 indicates that A* happened to find a solution at that grid resolution). This conclusion could be the result of a weak heuristic (not allowing for sufficiently tight bounds on best solutions), which in turn results in its incurring the overhead of running RNM for evaluation a prohibitive number of times.

Second, SLS seems to be competitive with A* on the quality of solutions it generates, often at a speed that is an order of magnitude improvement on A*. Indeed, the SLS BIN score is always within 10% of the optimal A* score, which seems to make it competitive in quality. A proper verification of this observation requires an examination of the actual optimal trajectories generated by A* and SLS (space limitations prohibits a display of these trajectories). Also, we are currently running tests that increases the depth if SLS search to see whether SLS can converge to global optimal, and at what depth this occurs.

Finally, these results indicate that running RNM to evaluate candidates quickly becomes prohibitive for large resolution problems. A current research goal is to find linear or low-order polynomial approximations to the function defined by RNM which will allow A* to bypass the invocation of RNM for each evaluation.

Summary and Future Work

This paper offers the following advances to the development of discrete search-based methods for trajectory optimization:

- A 3D state space and 6 DOF control model of the aircraft.
- Constraints one both state transitions and control transitions, related to safety and comfort.
- A cost function that aggregates data from a robust ground noise simulator taking into account human annoyance levels.
- Grid-based search space with parameters for enabling different resolutions.
- The implementation and experimental evaluation of a complete path planning method (A*).
- Head to head performance comparison of the complete method against an incomplete method (in particular, A* vs Local Search).

Our current research focus includes the investigation of the use of more sophisticated sampling-based motion planning methods, such as probabilistic road maps, to improve the quality of trajectories generated from a high-dimensional search space. We're also exploring more realistic environmental models based on 'land use' constraints; for example, residential areas, or hospitals along an approach pattern would be viewed as 'obstacles' in the problem.

References

Atkins, E., and Xue, M. 2004. Noise-sensitive final approach trajectory optimization for runway-independent aircraft. *Journal of Aerospace Computation, Information and Communication* 1:269–287.

Betts, J. T. 1998. Survey of numerical methods for trajectory optimization. *Journal of Guidance, Control and Dynamics* 21(2):193–207.

Cheng, P., and Lavalle, S. M. 2002. Resolution complete rapidly-exploring random trees. In *In Proc. IEEE Intl Conf. on Robotics and Automation*, 267–272.

Conner, D. A.; Burley, C. L.; and Smith, C. D. 2006. Flight acoustic testing and data acquisition for the rotor noise model (rnm). In *Proceedings of the 62nd Annual Forum of the American Helicopter Society*, 1–17.

Ferguson, D., and Stentz, A. T. 2005. The field d* algorithm for improved path planning and replanning in uniform and non-uniform cost environments. Technical Report CMU-RI-TR-05-19, Robotics Institute, Pittsburgh, PA.

2009. Fly neighborly guide. Technical report, Helicopter Association International.

Goplan, G.; Xue, M.; Atkins, E.; and Schmitz, F. H. 2003. Longitudinal-plane simultaneous non-interfering approach trajectory design for noise minimization. In *Proceedings of the 59th AHS International Forum and Technology Display*, 1–18.

Kavraki, L.; Kolountzakis, M. N.; and Latombe, J.-C. 1998. Analysis of probabilistic roadmaps for path planning.

LaValle, S. M. 2006. *Planning Algorithms*. Cambridge, U.K.: Cambridge University Press. Available at http://planning.cs.uiuc.edu/.

Mengshoel, O. J. 2008. Understanding the role of noise in stochastic local search: Analysis and experiments. *Artificial Intelligence* 955–990.

Morris, R.; Venable, K. B.; Pegoraro, M.; and Lindsay, J. 2012. Local search for designing noise-minimal rotorcraft approach trajectories. In *IAAI*.

on Noise, F. I. C. 1992. 1992 federal interagency commitee on noise (ficon) report - federal agency review of selected airport noise analysis issues. Technical report.

Padula, S. L.; Burley, C. L.; Jr, D. D. B.; and Marcolini, M. A. 2009. Design of quiet rotorcraft approach trajectories. In *sometitle*, xxxx–xxxx.

Page, J.; Wilmer, C.; and Plotkin, K. 2007. Rotorcraft noise model technical reference and user manual (version 7). Technical Report WR 07-04, Wyle Laboratories for NASA Langley Research Center.

Pettersson, P. O., and Doherty, P. 2004. Probabilistic roadmap based path planning for an autonomous unmanned aerial vehicle.

Sim, B. W.-C.; Schmitz, F. H.; and Gopalan, G. 2002. Flight-path management/control methodology to reduce helicopter blade-vortex interaction noise. *Journal of Aircraft* 39(2):193–205.

Xue, M., and Atkins, E. M. 2006a. Noise-minimum runway-independent aircraft approach design for baltimore-washington international airport. *Journal of Aircraft, American Institute of Aeronautics and Astronautics (AIAA)* 43(1):39–51.

Xue, M., and Atkins, E. M. 2006b. Terminal area trajectory optimization using simulated annealing. In *44th AIAA Aerospace Sciences Meeting and Exhibit*. Reno, Nevada: AIAA.

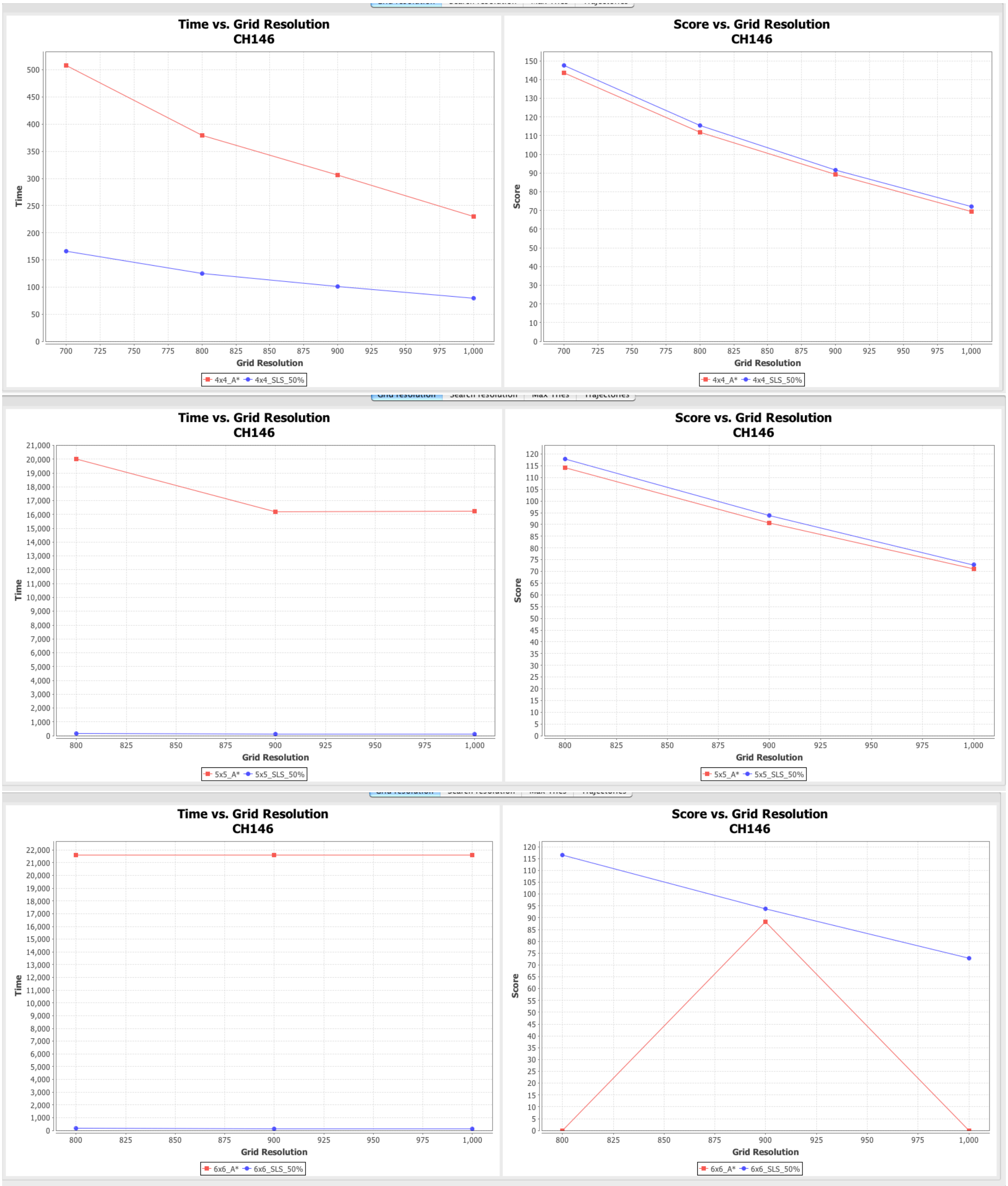

Figure 4: SLS vs A* varying grid resolution for three settings of search resolutions

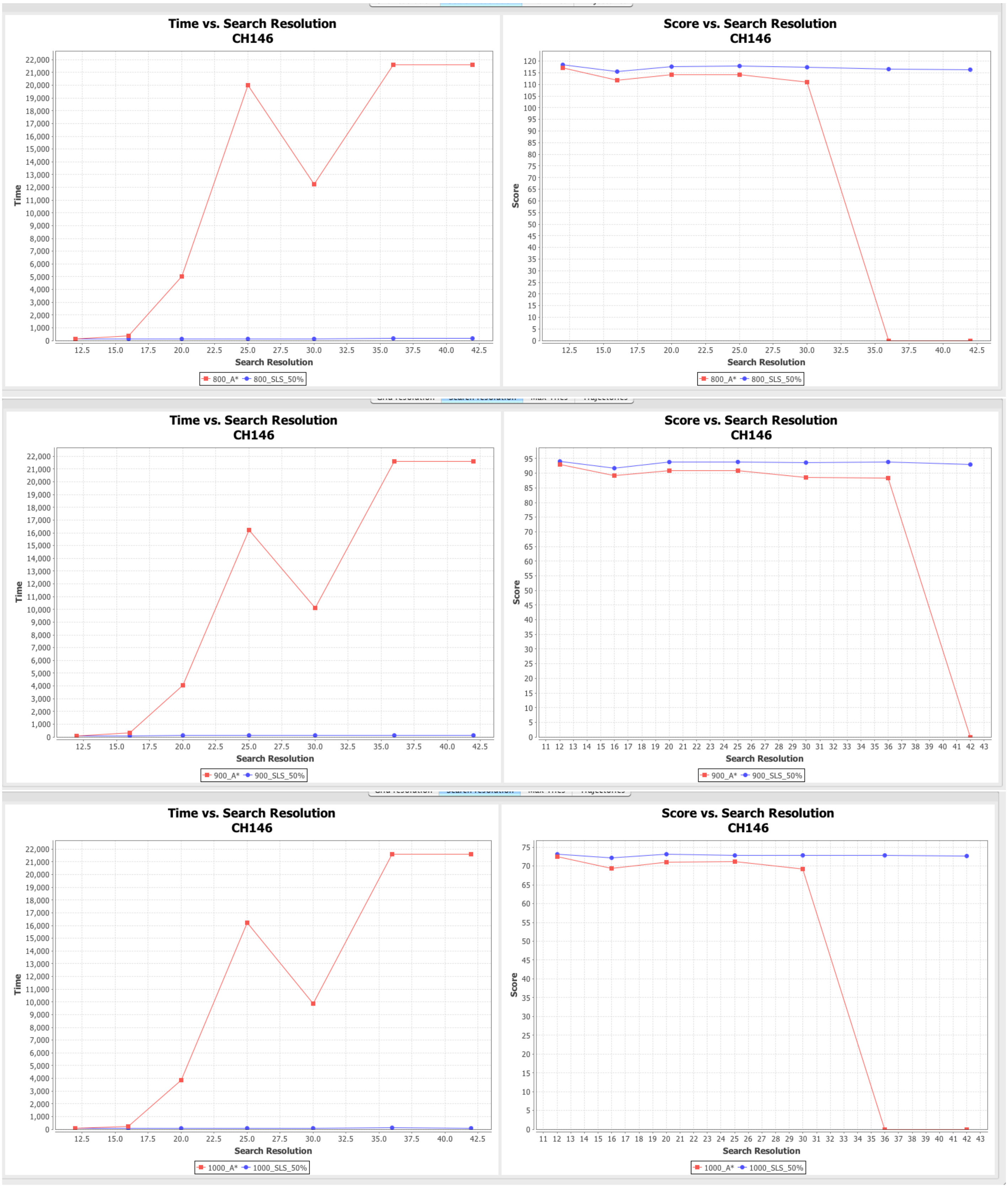

Figure 5: SLS vs A* varying search resolution for three settings of grid resolutions