Adaptive Trajectory Planning for Flight Management Systems

Igor Alonso-Portillo **Ella M. Atkins**

Aerospace Engineering Department
University of Maryland
College Park, MD 20742
{alonsoip, atkins}@glue.umd.edu

Abstract

Current Flight Management Systems (FMS) can autonomously fly an aircraft from takeoff through landing but may not provide robust operation to anomalous events. We present an adaptive trajectory planner capable of dynamically adjusting its world model and re-computing feasible flight trajectories in response to changes in aircraft performance characteristics. To demonstrate our approach, we consider the class of situations in which an emergency landing at a nearby airport is desired (or required) for safety considerations. Our system incorporates a constraint-based search engine to select and prioritize emergency landing sites, then it synthesizes a waypoint-based trajectory to the best airport based on post-anomaly flight dynamics. We present an engine failure/fuel starvation case study and illustrate the utility of our approach during a simulated thrusting power failure for a B-747 over the Bay Area.

Nomenclature

C_L – Lift Coefficient
C_D – Drag Coefficient
C_{D0} - Zero-Lift Drag Coefficient
D - Drag
E – Total energy
g - Gravity
h – Altitude
K – Induced drag parameter
m - Mass
Q – Fuel flow
T- Thrust
V- Airspeed
V_{wind} – Wind speed
γ - Flight path angle
φ - Bank angle
ψ - Heading

Introduction

Current Flight Management Systems (FMS) are capable of autonomously controlling an aircraft from takeoff through landing during nominal flight operations. Although these systems provide very high levels of autonomy, they lack robustness to a multitude of failure modes. Therefore, FMS require that the pilot remain "in the loop" for response to in-flight emergencies as well as Air Traffic Control (ATC) directives.

This paper describes an *adaptive trajectory planner* capable of computing new flight paths that take into account flight plan goals as well as system failures that affect aircraft performance. We propose feedback of changing flight dynamics from the lower level control systems to the high level path-planning module. This information can be crucial when there are variations in the flight envelope of the aircraft that invalidate the presumed model. Based on dynamic parameter feedback, our path planner adapts its performance model. Then, it either verifies that current trajectories are still safe or else generates a new trajectory that allows continued autonomous operation during post-failure flight. This trajectory regeneration will include not only an evaluation of the flight dynamics but also the search for a trajectory to an optimal landing site within the maximal reachable area.

Flight Management Systems

Flight Management Systems represent one of the best examples of long-term, robust autonomy in the aerospace field. First introduced two decades ago, today they have become standard equipment on commercial transport vehicles. FMS automation has produced significant improvements, which include reduction in the pilot workload, increase in flight safety and the introduction of large databases of routes, and trajectory optimization tools that have improved the economy of operation considerably (Sherry 1998). Significant future enhancements to cockpit automation are imperative to maintain safe flight conditions as air traffic volume increases and the tolerance for systems failures decreases.

Flight management requires the accomplishment of tasks both on the ground and in the air. Before leaving the gate, a flight plan must be created and approved by ATC. This plan is composed of a set of waypoints and associated arrival times at these 3-D locations.

During flight, under nominal conditions, the aircraft follows the flight plan using the FMS guidance, navigation and control (GN&C) system. Performance is monitored and optimized, and problems (e.g. fuel usage, systems failure, etc.) are reported to the pilot. Currently, the pilot provides the interface with ATC, translating course/flight plan changes dictated by ATC into FMS trajectory adjustments and reporting systems status when required (e.g. during emergency situations) to ATC.

Below we describe in more detail the basic functions of FMS divided into subgroups (Lidén 1994):

Flight Planning: Flight planning involves the specification of a sequence of airways, flight levels, waypoints, time and altitude constraints at waypoints, weather information, etc. The flight plan can be entered manually or a aircraft-specific route may be chosen instead.
Performance Optimization: The objective of this function is the computation of vertical and speed trajectory profiles that are optimal in terms of time, fuel and/or total operational cost subject to ATC constraints and within the operational flight envelope of the aircraft.
Performance Prediction: The goal of this module is to predict future trajectory information, including distance to destination, arrival time, aircraft state at future waypoints, etc. This system operates as a faster-than-real-time flight dynamics propagator, enabling early detection of divergence from the original flight plan, implications of ATC directives, and potential traffic conflicts.
Guidance: Guidance is the conversion of the planned trajectory into pitch, roll, yaw, speed, and thrust commands at each time point.
Navigation: Navigation is the process of computing aircraft state based on sensor data. State includes position (altitude, latitude, longitude), attitude, and velocity.
Electronic Flight Instrument System Display (EFIS): Data generated in the Flight Management Computer is displayed on the EFIS (A), as shown in Figure 1 below (Shaw 2001). This includes the Mode Control Panel (B), the Primary Flight Displays (C) and the Navigation Displays (D) that provide reference trajectory and dynamic navigation data. The EFIS is utilized for reference and data input.
Control and Display Unit (CDU): The CDU is the primary FMS/pilot interface, as shown in Figure 1 (E). The display of data and the fields for data entry are organized in pages presented on the CDU screen. Inputs are typically entered from a keypad.
Air-Ground Data Link: This system provides two-way data communication between the cockpit and ATC. Data includes flight plans, weather, radar information, etc.

Agent based Flight Management System

For this discussion, we segment the FMS into two hierarchical agents: planning and reactive. The reactive agent is designed to follow the flight trajectories specified by the planning agent, as shown in Figure 2. This agent representation is equivalent to the subgroup explanation given above. It must be noted however that several tasks have been omitted, namely those related to the pilot/FMS interfaces since this research is not focused on human/machine interactions.

In current FMS, both planning and reactive agents utilize a fixed dynamic model, and there is no feedback from control to guidance to flight planner regarding variations in this model. Complementary work in systems identification will detect these variations and can feed back new parameters to both reactive and flight planning agents. The feedback of this model to the adaptive trajectory generator (shown in Figure 2) is the key for its robustness to failures that affect performance.

Optimal flight trajectory synthesis is performed by the FMS flight planner prior to take-off. The term *trajectory* is used to describe a sequence of 4-dimensional aircraft states *(x,y,z,t)*, where *(x,y,z)* is the 3-D spatial coordinate vector and *t* is time. Flight path generation software must always verify that the trajectory lies inside the safe flight envelope of the aircraft, satisfies constraints imposed by ATC, and produces an optimal tradeoff between time, fuel and total operational cost as was outlined previously.

Figure 1: Partial View of the Boeing 777 Cockpit – Eurocontrol eCockpit Simulator.

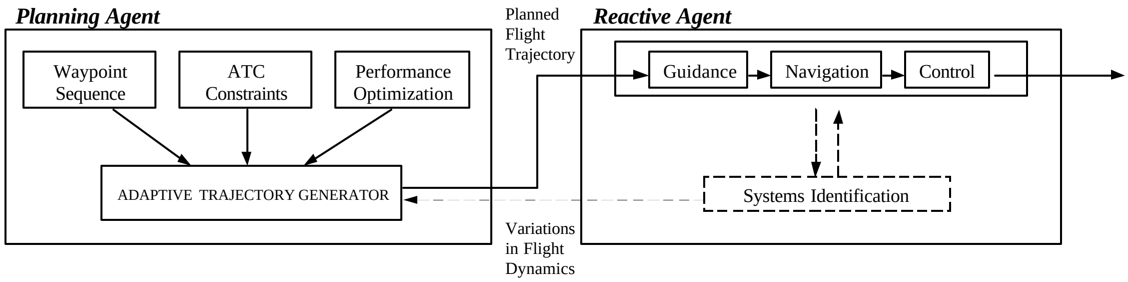

Figure 2: Agent based FMS Model.

Aircraft Flight Dynamics

The flight dynamics of an aircraft are accurately represented using a full 6 degree-of-freedom model (Nelson 1998) to characterize the forces and moments acting on the airplane. These consist of aerodynamic forces, thrust, and gravitational forces. However, in practice, to minimize complexity for online computation, the model utilized by FMS for the generation of optimum trajectories is a simplified point mass performance model.

The point mass model balances the primary forces acting on the aircraft, namely lift (L), drag (D), thrust (T) and weight (mg), as shown in Figures 3, 4 and 5. Although this is a crude approximation compared to a full 6-degree-of freedom rigid (or flexible) body representation, this simplification is necessary to reduce optimization complexity and to enable faster-than-real-time calculations for the performance prediction previously discussed. The mathematical representation of this point mass model (BADA 1998) is presented in Table 1, and assumes the following:

- Flat, non-rotating Earth.
- Standard Atmosphere.
- Fully coordinated flight. There are no side forces and side-slip angle is always zero.
- Aircraft is a point thus dynamics of its movements around its center of gravity can be ignored.

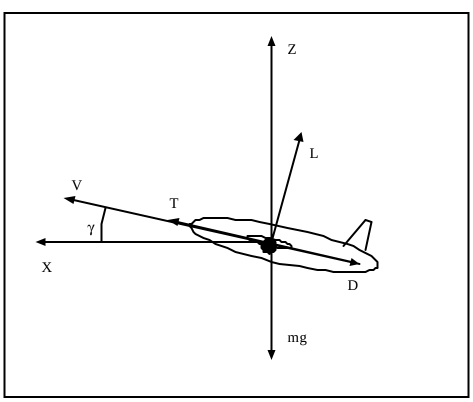

Figure 3: Side view of the Point Mass Model.

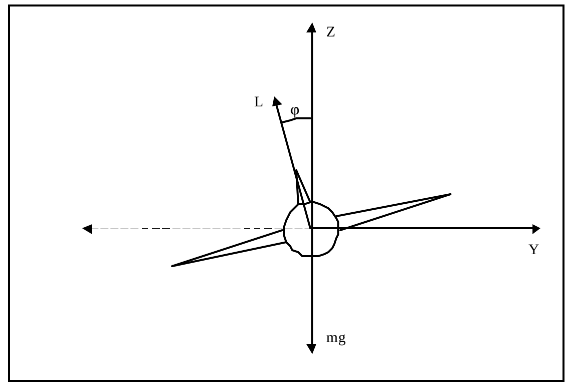

Figure 4: Front view of the Point Mass Model.

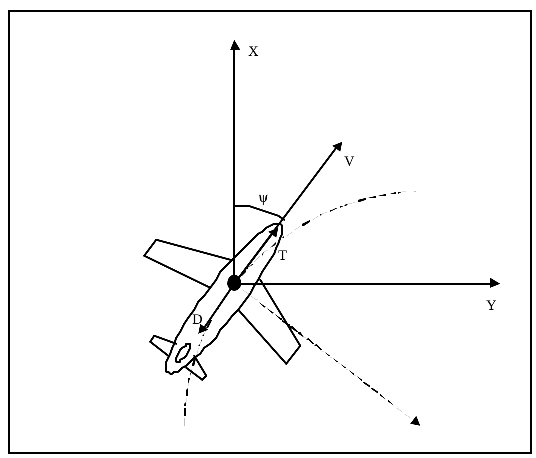

Figure 5: Top view of the Point Mass Model.

The equations presented in Table 1 can be easily derived with basic trigonometry from the force and velocity diagrams shown in Figures 3, 4 and 5. Forward propagation is minimally complex as they are all first order ordinary differential equations. In this point performance model, the longitudinal and lateral dynamics of the aircraft are decoupled which greatly simplifies the performance optimization computations.

$$\dot{V} = \frac{T - D}{m} \cdot V - g \cdot \sin\gamma$$

$$\dot{\psi} = L \cdot \frac{\sin\varphi}{m \cdot V \cdot \cos\gamma}$$

$$L = m \cdot g \cdot \frac{\cos\gamma}{\cos\varphi}$$

$$C_L = \frac{2 \cdot L}{\rho \cdot V^2 \cdot S}$$

$$C_D = C_{D0} + K \cdot C_L^2$$

$$D = C_D \cdot \frac{1}{2} \cdot \rho \cdot V^2 \cdot S$$

$$\dot{h} = V \cdot \sin\gamma + V_{wind}^{z}$$

$$\dot{x} = V \cdot \cos\gamma \cdot \sin\psi + V_{wind}^{x}$$

$$\dot{y} = V \cdot \cos\gamma \cdot \cos\psi + V_{wind}^{y}$$

$$\dot{m} = Q$$

Table 1: Point Mass Model Dynamic Equations.

The point mass equations define relationships between speed (V), flight path angle (γ), vertical rate ($\dot{h}$), etc. However, these variables must always be restricted to values that lie within the flight limits of the aircraft. For a given aircraft, these boundaries define the controllable and operational regimes for flight trajectories. The set of valid values are determined by aerodynamic, propulsion and structural characteristics and are also known as the "flight envelope" of the aircraft.

Adaptive Trajectory Generator for Robust Aircraft Failure Response

Numerous aircraft crashes may be attributed to structural or mechanical failures. These will cause the flight dynamics to adversely vary and therefore the aircraft performance to be modified. Failures can include control surface jams, engine failures, flap and slat deployment in flight, etc. These are primarily single-event failures. However, progressive faults such as ice accumulation on the aircraft need to be considered also as they will modify the performance increasingly over time if a corrective action is not taken.

Adaptation of the automatic control laws is evidently important, but robust reconfigurable control systems may not be sufficient alone to successfully handle all failure events. The flight envelope of the aircraft will be altered by the change in flight dynamics, and thus the pre-failure flight plan may no longer lie within the controllable and operational regime of the aircraft. As previously mentioned, current FMS do not feed this information back to the trajectory planner. This "missing link" makes the system lack robustness under failure conditions, instead relying on the human pilot to manually specify the appropriate emergency controls.

The goal of this research is to develop an adaptive trajectory planner module that can be "plugged into" an FMS. We place emphasis on maximizing the usage of current FMS capabilities, dynamic models, trajectory generation algorithms, etc. The proposed module is meant to improve existing system robustness rather than replace the proven and mature technology implemented within the FMS.

As shown in Figure 2, the adaptive trajectory generator processes changes in the aircraft dynamic model fed back from the systems identification module (Hamel and Jategaonkar 1995). These changes are represented in the form of numerical dynamic coefficients in an analogous format to that utilized currently by FMS. In the simplified point mass model equations previously presented these would correspond to C_L, C_D, C_{D0}, and K. These coefficients are specified for different flight conditions (take-off, climb, cruise, descent, landing) currently. The systems identification module will provide new post-failure values for these parameters.

The high-level goal of the adaptive trajectory planner is to verify that the existing waypoint-based flight plan is valid and to specify a new path otherwise. Ideally, any flight path changes will strictly involve small perturbations to the trajectory to maintain optimality. Such changes may be fed into the reactive agent but then can be ignored in the high-level flight plan. In other cases (e.g., partial power loss), dynamic model changes will require modifications to the trajectory that also affect waypoint arrival times, but still the overall flight plan remains valid.

In the most extreme cases, the adaptive trajectory generator is unable to find any valid trajectory that achieves the flight plan waypoint goals. In these situations, either the pilot or an *adaptive flight planner* must develop a new set of waypoints that can be followed with a feasible trajectory. A multitude of AI planning architectures can build waypoint sequences given a "reachable" waypoint transition map, available landing sites, and reward values associated with each landing site. Of course, real-time response is crucial for dangerous failures, thus the adaptive flight planner/pilot must be capable of a timely reaction.

Case Study: Loss of Thrusting Power

In order to perform a preliminary study of the elements, strategies, and algorithms necessary for the adaptive trajectory generator, a case study of a specific failure was performed. This first failure was chosen to be a complete aircraft engine failure or equivalently fuel starvation. As discussed previously, trajectory regeneration requires system identification to feed back an accurate model of the post-failure flight dynamics. In the case of engine failure, this model variation is trivial since all equations and parameters are same except that the thrust is zero. For the same reason, the flight and maneuverability envelope remains constant except for the propulsion information. Thus in this example, we concentrate on the high-level replanning agent assuming perfect information feedback.

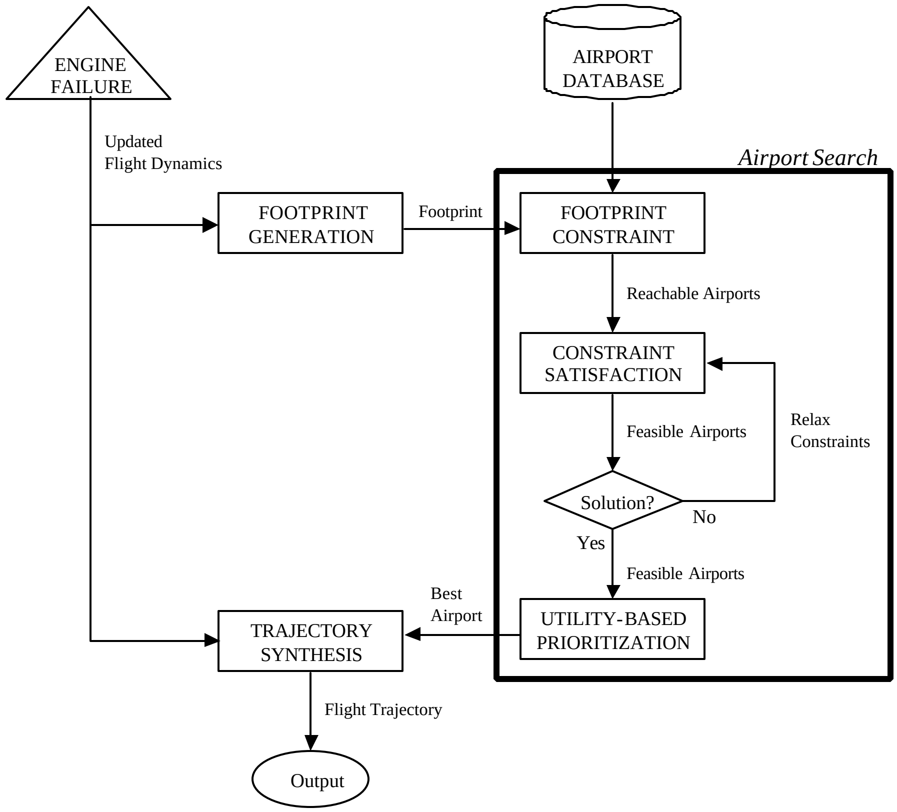

Figure 6. Fault Recovery Algorithm for Engine Failure.

In the case of total loss of engine thrust, achieving the original planned waypoint sequence is not possible, and the priority becomes the search for a landing site. This site will be chosen from a database of existing airports. For this research we utilize a database of all US airports provided by NASA (see ASAC website reference). This database includes airport locations, runway specifications, and a variety of other relevant information.

Our fault recovery algorithm focuses on a directed search for the best landing site and is presented in the block diagram shown in Figure 6. Once the failure occurs, it is detected and the flight dynamic model modified accordingly. Then a maximal reachable footprint is generated utilizing the performance optimization and prediction tools discussed in the FMS section. The geographical coordinates of this footprint are then superimposed onto the airport database, allowing selection of reachable airports. A constraint analysis is then performed to select minimally safe airports within those that are reachable, and as shown, constraints are relaxed as required to give at least one solution. Next, a utility function is applied based on airport characteristics to determine the "best" airport for an emergency landing. Finally a trajectory to that airport is synthesized utilizing the current FMS tools already introduced.

Footprint Generation

The output of the footprint generation is an approximate maximal reachable terrain area. The optimal footprint generation is a complicated process that is performed via the calculus of variations (Vinh 1981). This is a computationally complex process, which yields a highly non-linear solution. This method was not utilized for two main reasons. First, the nature of the computation is complex and in an emergency situation a fast response is mandatory. Second, the pilots or autopilots do not fly a highly non-linear profile in 3 dimensions. The flight path control systems installed in FMS include the following flight modes: height hold, speed hold, mach hold, heading hold, and vertical speed of flight path angle hold. These correspond to simple trajectories where waypoints are connected with straight lines. Thus it is advisable to generate a footprint in the same fashion.

The footprint is created by generating a finite set of "flattest glide" trajectories for a range of 0-360 degrees of desired final heading. This type of trajectory yields a maximum range descent and is only affected by the aerodynamic characteristics of the aircraft and the initial altitude (Hale 1984). An initial heading change is performed, after which the best glide flight is projected. Also, turning at the highest altitude, from a safety point of view, is desired as it allows for longer reaction time in case of error.

The computational complexity of this process depends on two factors: the algorithm used to integrate the equations of motion and the number of headings chosen to define the footprint. A smaller order integration strategy will provide a faster solution but a possibly inaccurate predicted trajectory. For this study, a 4th order Runge Kutta algorithm was utilized. The computational complexity is proportional to number of unique headings generated to define the footprint. An infinite set of heading values would provide a continuous boundary for the footprint but require too much time to compute. Instead a rectangular footprint is generated as a first approximation. Let the pre-failure heading be ψ. Then, to generate the footprint, turns to headings (ψ-90^o, ψ+0^o, ψ+90^o and ψ+180^o) are performed. A rectangular "pseudo-footprint" can be quickly generated utilizing this basic information. Having generated the footprint of maximal reachable area, all airports that lie within the footprint are identified from the database.

Constraint Satisfaction

Application of the footprint constraint yields a set of *reachable airports* based strictly on geographic location. However, other constraints must be met before the airport may be considered a *feasible* landing site. For example, a Boeing 747 cannot land on "short" airstrips, eliminating the numerous general aviation airports (e.g., Palo Alto) from consideration. Conversely, many general aviation aircraft (e.g., Cessna 152) can land on a very short runway but cannot be expected to land on a runway with a heavy crosswind.

The US airport database supplemented by current wind/weather conditions can provide the information necessary to select feasible landing sites. We have developed a basic set of constraints required for a safe landing. These are listed below in Tables 2 and 3, where Table 2 lists type-specific constraints and Table 3 lists other airport and weather-related constraints.

Constraint Description	**B-747**	**C-152**
Minimum runway length	8000 feet	2000 feet
Minimum runway width	150 feet	50 feet
Maximum crosswind	35 knots	15 knots

Table 2: Aircraft-specific Constraints.

Constraint Description	**Constraint Applicability**
Runway lighting	Night flight
Instrument approach	Bad visibility (IFR) conditions
Paved runway surface	Soft (wet) turf / heavy aircraft

Table 3: Airport and Weather Constraints.

The constraints from Tables 2 and 3 will provide the list of feasible airports. As discussed in the next section, if this list contains multiple candidates, we use a utility function to prioritize the selection. With either a small footprint (e.g., due to low-altitude engine failure) or when over-flying a remote area, the default constraint set may eliminate all airports. In this case, as shown in Figure 6, the constraints must be relaxed until at least one runway is identified. As an example, consider the relaxation of the runway length constraint. The values shown in Table 2 give a comfortable safety margin should the engine-out approach not be perfect. However, if reaching a runway that exceeds the Table 2 minima is not possible, minimum runway length could be reduced until at least one runway is identified. Landing on a short runway at least allows the aircraft to make ground contact under controlled conditions. This substantially increases the odds of surviving a forced landing.

Utility-based Prioritization

In situations where multiple airports are feasible landing sites, we wish to select the "best". To do this, we apply a utility function that prioritizes the airports, using the same airport and weather databases referenced above for constraint satisfaction. After the prioritization, we submit the highest-utility airport to the trajectory synthesizer, which develops a path from the current aircraft location (initial state) to the landing site (final state).

The airport database set contains over 100 data fields describing the airport facilities. As a start, we incorporate the same data used for constraint satisfaction, based on a numerical scale such that increased capabilities yield higher utility. For example, a 10,000 foot runway is safer thus has higher utility than an 8,000 foot runway, although both are adequate. As another example, we consider different categories of instrument approach equipment (rather than its simple existence). For example, an ILS (Instrument Landing System) approach with glide slope is the most accurate commercial equipment available today, and is categorized as Category I, II, or III based on equipment certification. The Cat. III ILS approach would score perfectly in the utility function (since it allows landing in zero-visibility conditions). Other approaches, including Cat. I or II ILS, GPS, VOR, and NDB, would score lower, proportional to the minimum decision height (descent altitude) specified on the instrument approach charts for that airport.

A simple utility function is shown in Equation (1), where the C_i represent user-specified coefficients, r_l is runway length, r_w is runway width, I is instrument approach availability (e.g., 1=ILS, 0.8=GPS), w_c is crosswind velocity, and S is surface type (e.g., 1=non-skid paved, 0.9=concrete, 0.8=asphalt, etc). A host of other data may be incorporated in the future, including obstacles on approach and adverse weather conditions such as thunderstorms and wind shear. Statistical data and domain expert input will be crucial to improve utility function quality.

$$U = C_1 r_l + C_2 r_w + C_3 I + C_4 w_c + C_5 S \qquad (1)$$

Results

The design of our adaptive trajectory planner and its performance in the handling of failures was tested in an FMS simulation environment. In order to do so, we built a Matlab-based footprint generator over an existing FMS trajectory planning tool. Designed at the Eurocontrol Experimental Center in Brétigny-Sur-Orge, France, our Matlab-based FMS model is derived from a Flight Management and Guidance Control System for an ATC simulation traffic generator (Hoffman 2000).

The results of the case study for engine failure or fuel starvation are presented. In this example, the failure occurs at the Latitude and Longitude of Palo Alto, California during cruise flight at an altitude of 30,000 feet.

Figure 7 shows the output of the footprint generation tool. As was described above, the footprint represents the approximate maximal terrain area that the aircraft can reach after the failure occurs. Figure 8 illustrates the real-time generation of the simplified rectangular pseudo footprint. As can be observed from the figure, the complexity of this generation is minimal since only four trajectories need to be forward propagated to obtain the rectangular area.

Figure 9 shows the airports within the expansive 200 km radius footprint region. As illustrated, there are a large number of airports within range, corresponding to a high airport-density area. Among these, performance constraints in both runway length and width were imposed resulting in a selection of feasible airports for an emergency landing. These are shown in Figure 10. Finally, the utility function evaluation was performed to determine the "best" landing site as discussed in the previous section. The results of this utility calculation are presented in Table 4. For this particular example the coefficients from Equation (1) are chosen such that all the individual utility characteristics are weighted equally. For illustrative purposes, the relative scores of the airports were normalized to produce a perfect score of 1.0 for the "best" landing site.

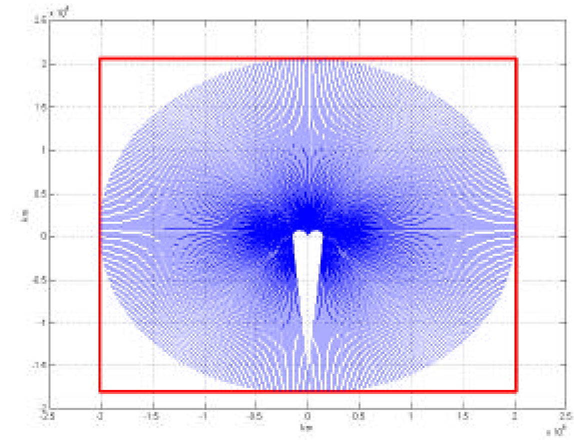

Figure 7: B-747 Footprint from 10,000m Altitude.

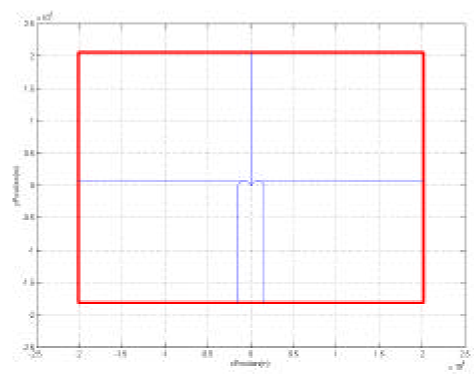

Figure 8: Pseudo-footprint Generation.

Airport	Identifier	Utility
Sacramento Mather	MHR/04R	1.000
San Francisco Int.	NUQ/10L	0.922
San Francisco Int.	SUU/10R	0.900
McClellan AFB	MCC/16	0.880
San Jose Int.	SJC/12R	0.860
Met. Oakland Int.	OAK/11	0.857
Castle	MER/13	0.847
Beale AFB	BAB/15	0.769
Travis AFB	SUU/3L	0.712
Travis AFB	SUU/3R	0.712

Table 4: Normalized utility values for feasible airports.

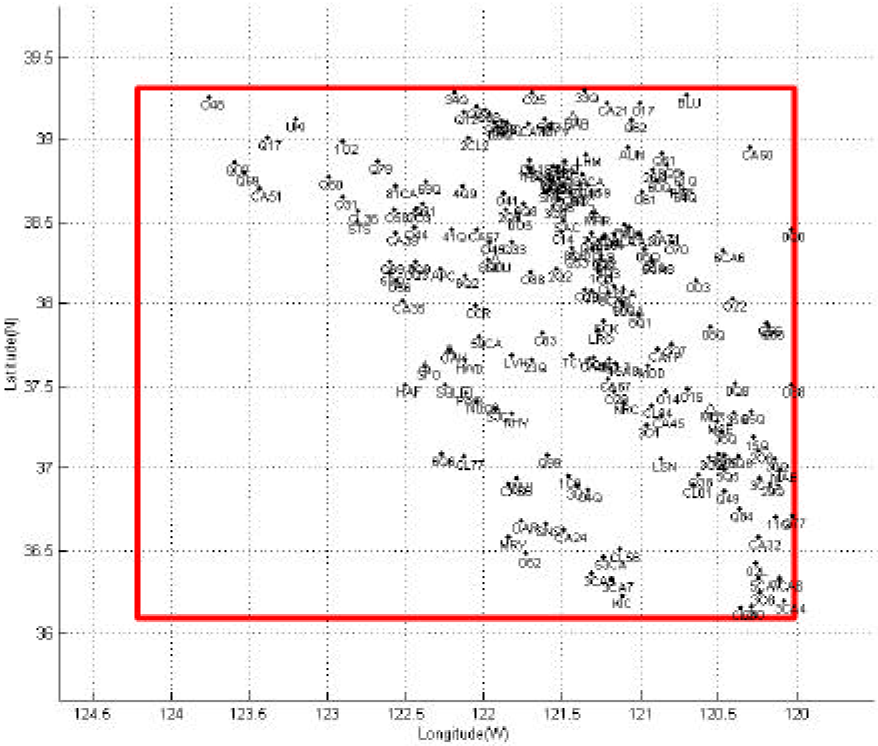

Figure 9. Reachable airports within rectangular pseudo footprint.

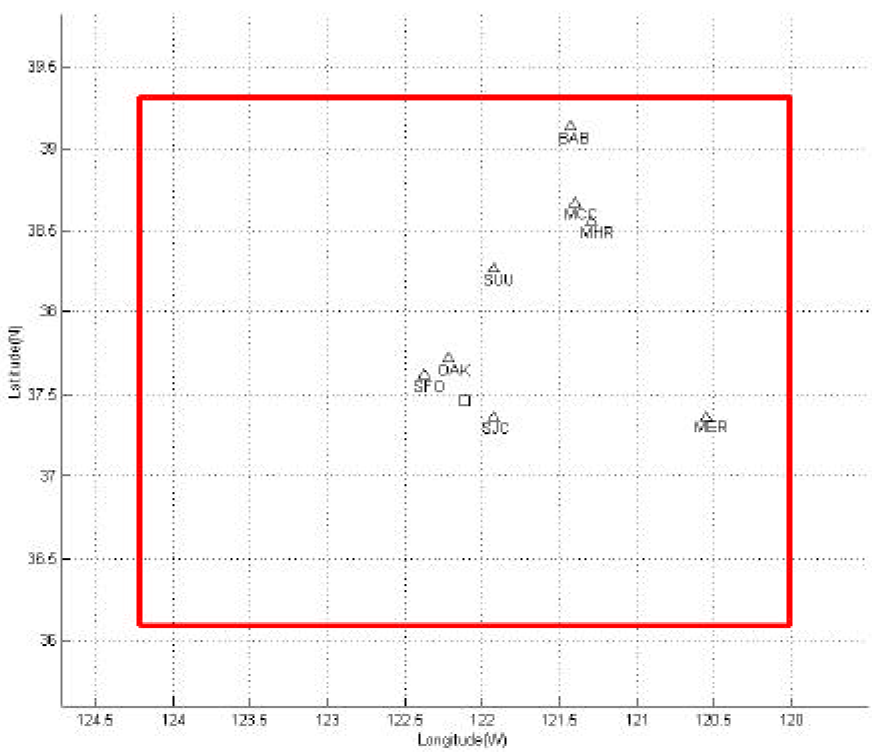

Figure 10. Feasible landing sites after performance constraint.

Summary and Future Work

An adaptive trajectory generation module for flight management systems (FMS) has been introduced. This module can enhance current FMS autonomy and provide robustness to different failure modes, a capability currently not available. The simulation results for an engine failure case study illustrate the utility of our fault recovery algorithm to intelligently select a feasible emergency landing site based on a dynamically-updated aircraft performance model.

Ongoing work is progressing toward the definition of a more complete utility function. We are also refining the strategy by which we generate detailed waypoints to autonomously guide the aircraft down to the emergency landing runway. To perform this task, the FMS must automatically define a pattern from any approach heading that aligns the aircraft properly with the chosen runway and accounts for expected wind/weather conditions.

The concept of adaptive trajectory generation, consisting of footprint computation, landing site selection, and trajectory synthesis, is general for any anomaly that requires flight path alteration. However, the algorithms internal to the landing site selection process require more work before we can cast them in a more general search/planning framework applicable to any anomalous situation. We are in the process of characterizing more dynamically-complex failures modes that affect aircraft performance, such as a bound control surface or airframe icing, into our algorithm.

One of the major challenges faced by the planning community is to adequately model and adapt to changing dynamic behavior in complex systems. Our approach combines features of a simple symbolic planner (e.g., airport selection) with the continuous and adaptive dynamic models required to accurately characterize system performance. We believe this hybrid strategy will ultimately provide a "bridge" between the symbolic planning and traditional control communities, and we look forward to continued progress along this path to robust autonomy.

References

Eurocontrol, "User Manual for the Base of Aircraft Data (BADA) revision 3.1," *EEC Note No. 25/98*, Eurocontrol Experimental Centre, Bretigny-sur-Orge, November 1998.

Hale, F.J., *Introduction to Aircraft Performance, Selection and Design*, Wiley, New York, 1984.

Hamel, P.G., and Jategaonkar, R.V., "The Evolution of Flight Vehicle System Identification," in *AGARD Structures and Materials Panel Specialists' Meeting on Advanced Aeroservoelastic Testing and Data Analysis*, Rotterdam, The Netherlands, 1995.

Hoffman, E., and Levrez, J.M., "Flight Management and Guidance Control System Model for an ATC Simulation Traffic Generator", *EEC Report No. 303*, Eurocontrol Experimental Centre, July 2000.

Liden, S., "The Evolution of Flight Management Systems," *IEEE Digital Avionics Systems Conference*, 1994, pp. 157-169.

Nelson, R.C., *Flight Stability and Automatic Control*, McGraw-Hill, New York, 1998.

Shaw, C., http://www.eurocontrol.fr/projects/freer/e-Cockpitversions/, Eurocontrol, Bretigny-sur-Orge, 2001.

Sherry, L., "Intelligent Process Control in Aviation," *PCAI Magazine*, May/June 1998.
Kostiuk, P., http://www.asac.lmi.org/html/dserver/airportdb.html, LMI, McLean, VA, 2001.

Vinh, N.X., *Optimal Trajectories in Atmospheric Flight*, Elsevier, New York, 1981.