

# A Trajectory Generation Approach for Payload Directed Flight

Corey A Ippolito<sup>1</sup>

*NASA Ames Research Center, Moffett Field, CA 94035*

Yoo-Hsiu Yeh<sup>2</sup>

*Carnegie Mellon University Silicon Valley Campus, Moffett Field, CA, 94035*

Presently, flight systems designed to perform payload-centric maneuvers require pre-constructed procedures and special hand-tuned guidance modes. To enable intelligent maneuvering via strong coupling between the goals of payload-directed flight and the autopilot functions, there exists a need to rethink traditional autopilot design and function. Research into payload directed flight examines sensor and payload-centric autopilot modes, architectures, and algorithms that provide layers of intelligent guidance, navigation and control for flight vehicles to achieve mission goals related to the payload sensors, taking into account various constraints such as the performance limitations of the aircraft, target tracking and estimation, obstacle avoidance, and constraint satisfaction. Payload directed flight requires a methodology for accurate trajectory planning that lets the system anticipate expected return from a suite of onboard sensors. This paper presents an extension to the existing techniques used in the literature to quickly and accurately plan flight trajectories that predict and optimize the expected return of onboard payload sensors.

## 1. Introduction

Presently, flight systems designed to perform payload-centric maneuvers require pre-constructed procedures and special hand-tuned guidance modes. To enable intelligent maneuvering via strong coupling between the goals of payload-directed flight and the autopilot functions, there exists a need to rethink traditional autopilot design and function. Research into payload directed flight (PDF) examines sensor and payload-centric autopilot modes, architectures, and algorithms that provide layers of intelligent guidance, navigation and control for flight vehicles to achieve mission goals related to the payload sensors, taking into account various constraints such as the performance limitations of the aircraft, target tracking and estimation, obstacle avoidance, and constraint satisfaction.

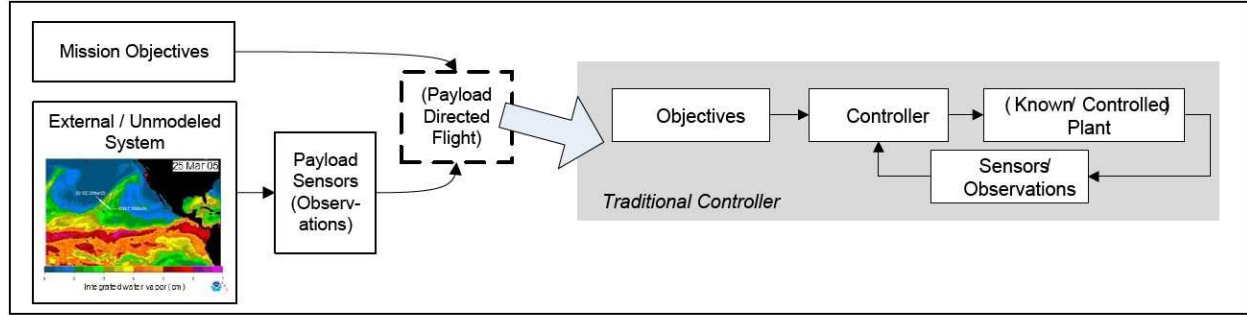
A central problem to address in payload directed flight is to control a known and controllable plant interacting with an external system based on payload and sensor data feedback that gives partial observation and understanding of the external system, to satisfy mission objectives and constraints on the combined system. This research focuses on trajectory generation and flight control under varying constraints in a highly dynamic environment, autonomous feature detection and estimation, and modelless autopilot design concepts for multi-objective system control. Application of this research is targeted towards increasing capabilities, performance, and efficiency in the execution of missions that require payload-directed and target-directed maneuvering.

This central problem is shown conceptually in the block diagram in Figure 1, where a controllable system is coupled with an external system which may be unmodeled or poorly modeled for various reasons. These reasons may include complexity, uncertainty, lack of observability from sensor to state, the size of the external system's state may overwhelm computational and modeling resources, or lack of available data to generate a model. A suite of sensors provide some set of observations into the system, and a set of mission objectives are defined concerning the combined system. The PDF research objectives seek methods, tools, and techniques for designing controllers around these blocks to ensure the combined system meets mission objective under varying constraints.

---

<sup>1</sup> Research Scientist, Intelligent Systems Division, NASA Ames Research Center, AIAA Member

<sup>2</sup> Project Engineer, Intelligent Systems Division, NASA Ames Research Center, AIAA Member



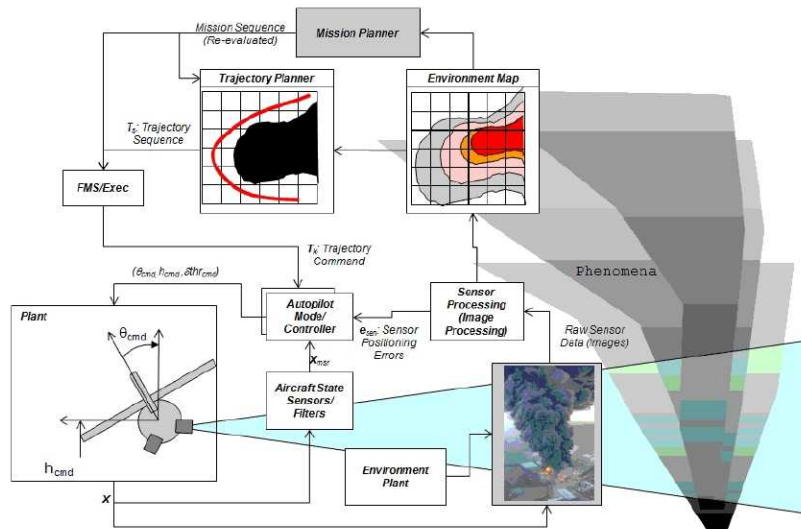
**Figure 1. Payload Directed Flight Problem**

PDF mission concepts require simultaneous payload data loop closure at multiple levels in the control system hierarchy, from high-level cognitive deliberative decision making down to low-level reactive continuous-time control. The PDF architecture defines three distinct layers of loop closure: the outer layer, which comprises of the mission planner and scheduler, the middle layer, which comprises optimal trajectory calculations, and the inner layer, where filtered sensor data cuts directly into the autopilot system, providing for instance attitude command queues. The PDF architecture middle layer problem has two main constituents: computing optimal trajectories, and incorporating knowledge representation of the observed system. Optimal trajectory calculations play a key role in the concepts of Payload Directed Flight, particularly in the middle layer PDF architecture, that concerns planning behaviors on the time-scale of minutes.

#### A. Payload Missions Requiring Middle Layer Planning

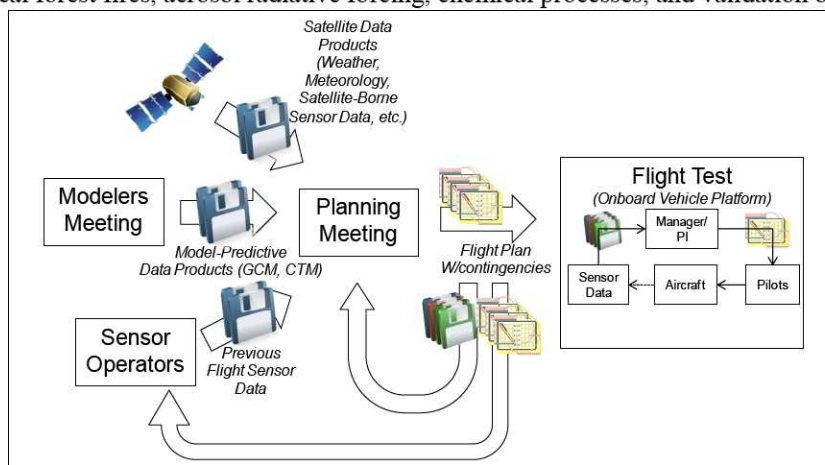
The requirements for loop closure at the middle PDF layer require the onboard systems be able to compute trajectories under complex constraints and objectives. For instance, automated formation flying aircraft need to be able to represent complex phenomena - other aircraft, wing tip vortices, weather patterns - in calculating trajectories that will allow an aircraft to join and follow a formation, while the lead aircraft needs to be able to compute trajectories that provide the optimal paths for the convoy. A large number of possible applications for payload directed flight are outlined in Ref. 1.

A more complex example is shown in Figure 2. Consider an aircraft whose mission is to locate, identify, and monitor a hazardous smoke plume, utilizing onboard body fixed imagers. At the lowest level of control, direct control of the actuators can be utilized to center the image in the view frame. At the middle layer, sensor data is used to update an online environment model, and trajectories are planned to field the sensors at locations that maximize expected data return while navigating safely around the hazardous phenomena. At a higher level, mission objectives can be planned, scheduled, and prioritized based on what is being observed from the sensors.



**Figure 2. Mid Level Loop Closure in Hazardous Sensing Missions**

The airborne earth science community regularly field missions that require flight planning to investigate large scale highly complex phenomena, fielding onboard payload sensor suites from theoretical predictive models. Currently, human intuition and judgment is required to interpret the data and develop flight plans to maximize the expected data return. The 2008 Arctic Research of the Composition of the Troposphere from Aircraft and Satellites (ARCTAS) mission featured three aircraft – a DC-8, P-3, and B-200 – fielding highly sophisticated scientific payloads to investigate a variety of scientific themes, including long-range transport of pollution to the arctic, emissions from boreal forest fires, aerosol radiative forcing, chemical processes, and validation of satellite sensors.



**Figure 3. ARCTAS Flight Planning and Execution Model**

The authors observed the planning model in Figure 3 during the 2008 ARCTAS deployment fielded out of Fairbanks, Alaska, which was based on earlier INTEX-B<sup>2,3</sup> flight planning process. Participants in this mission included project managers and principle investigators who oversaw operation of the mission, platform operators and flight crew who operated the airborne platforms, instrument operators and experimentalists who operated sensor equipment and interpreted data in real-time to provide to the managers during flight, and theoretical investigators and modelers who produced data products from chemical transport models and general circulation models (CTMs and GCMs) for chemical and meteorological predictions. The flight plans also required aircraft underflights of satellites for ground truthing and validation. Satellite and model science teams were involved at all stages of pre-mission flight planning, flight execution, and post-mission data interpretation. Flight planning relied heavily on CTM/GCM simulation, analysis of satellite observations, meteorological and chemical forecasts from several modeling teams, analyses of near real-time satellite data and satellite validation needs, inputs from the aircraft science teams, and reviews of progress towards meeting mission objectives. The flight planning process required planners to estimate sensor return expected from the flight paths based on the various data products. A PDF middle layer planner could integrate into this model as a first step for validation of these algorithms, providing suggested plans to the project managers and principle investigator teams both in the pre-planning phase and during flight testing as shown in Figure 4.



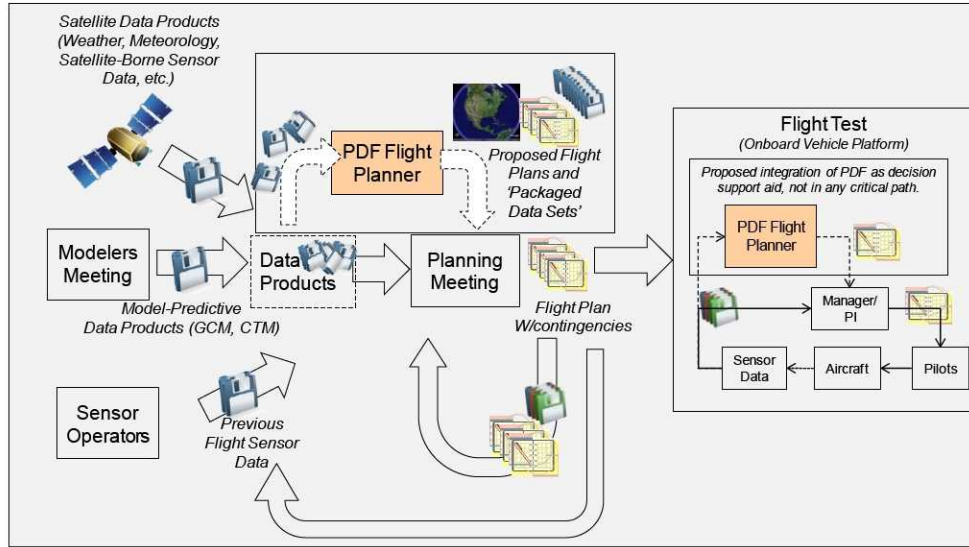


Figure 4. Conceptual Integration for PDF Technology

## B. Related Approaches

There are many examples in the UAV literature of closing the loop around payload sensors. In Ref. 4 and 5, vision systems are used to extract road centerlines. Here the road identification process is achieved through image processing that exploits the linear or locally linear nature of roadways. Once the road center line is extracted, the tracking problem is reduced to a simple path following problem. The traditional approach to this problem is to formulate an inner loop controller using linear control theory and design an outer loop controller that uses the desired path to specify the desired bank angle or lateral acceleration. Nonlinear control laws have been proposed for solving this problem<sup>4</sup> (similar control results are also found in Refs. 6-9). Additionally, similar research tracking rivers and shorelines is presented in Refs. 10 and 11. An approach for tracking a ground vehicle is presented in Ref. 12. In Ref. 13, a UAV serves as a communication repeater in a larger communication network. To achieve this, the UAV orbits a radio or communication ground source maintaining a fixed signal-to-noise ratio (SNR) with that source. The problem is analogous to the contour following problem with in-situ sensors. Presented in Ref. 13 is a traditional PID based control law formulated around the SNR error between time steps. This formulation, as demonstrated through simulation, will cause the UAV to spiral towards the specified contour (specified SNR value) and remain there once it is acquired. However, implicit in the formulation of this control law is the assumption that the SNR field is monotonic. In many of the cases already presented, this assumption was not applicable and necessitated the development of behavioral approaches.

The re-tasking problem for UAS's has also been explored in the literature. In Ref. 14 a list of targets to service is treated as a Traveling Salesman Problem. The problem of incorporating vehicle dynamics is achieved by solving the Traveling Salesman Problem using heuristics from traditional combinatorial optimization and then alternating the paths between targets as linear paths and minimum Dubins paths<sup>16</sup>. Several competing approaches using Dubins minimum paths are also available in the literature<sup>14-18</sup>.

In Ref. 19, the notion of planning based on sensor swath was explored. In this work a UAS has a downward facing camera with a field of view. The problem is to determine the optimal tour through the targets such that all targets are observed. Unlike the TSP problem where the UAS passes through all the targets, this is a case where it is only necessary for the targets to pass through the sensor's field of view. To solve this problem the UAS was modeled as a Dubins car with a discrete set of inputs  $u$  operating over a finite  $\Delta t$ . The operational space for the UAS was then explored using this model and the *learning A\* algorithm* operating with an admissible heuristic. A similar problem as this was also considered in Ref. 20. Here, however, the operational space is explored using a probabilistic planning approach based on the Rapidly-Exploring Random Root Tree algorithm<sup>21,22</sup>.

Some of the research in the vein of payload directed flight for fixed-winged vehicles is focused on searching for targets with an unknown location. One example in Ref. 23 is probabilistic in nature and involves selecting a search space, discretizing that search space as a grid of cell locations, applying a probability that a target is in a given cell, and identifying the optimal path (in a probabilistic sense) for identifying targets. The research in this field is

currently focused on optimal searching with multiple UAVs and optimizing the target identification or mapping ability over many vehicles<sup>24</sup>.

The approach developed in this paper will utilize a numerical technique from the path planning literature to solve problems posed in an optimal control formulation. The algorithm and approach described below extends previous algorithmic techniques to include highly accurate vehicle trajectory generation under a general set of dynamic, state, and input constraints. Our approach differs from the state of the art by utilizing continuous-time feedback control systems designed specifically to solve the trajectory generation problem utilizing the most accurate vehicle models available, or to the level of fidelity desired. In essence, this approach favorably trades run-time complexity for offline control system design. As a consequence of this approach, input control actuation is a product of the trajectory generation process and only dynamically feasible trajectories will be generated, rather than searching over discretized subset of inputs and generating highly simplified, unverified trajectories.

### C. Payload Directed Flight Architecture

Consider the overall PDF system architecture, shown in the figure below.

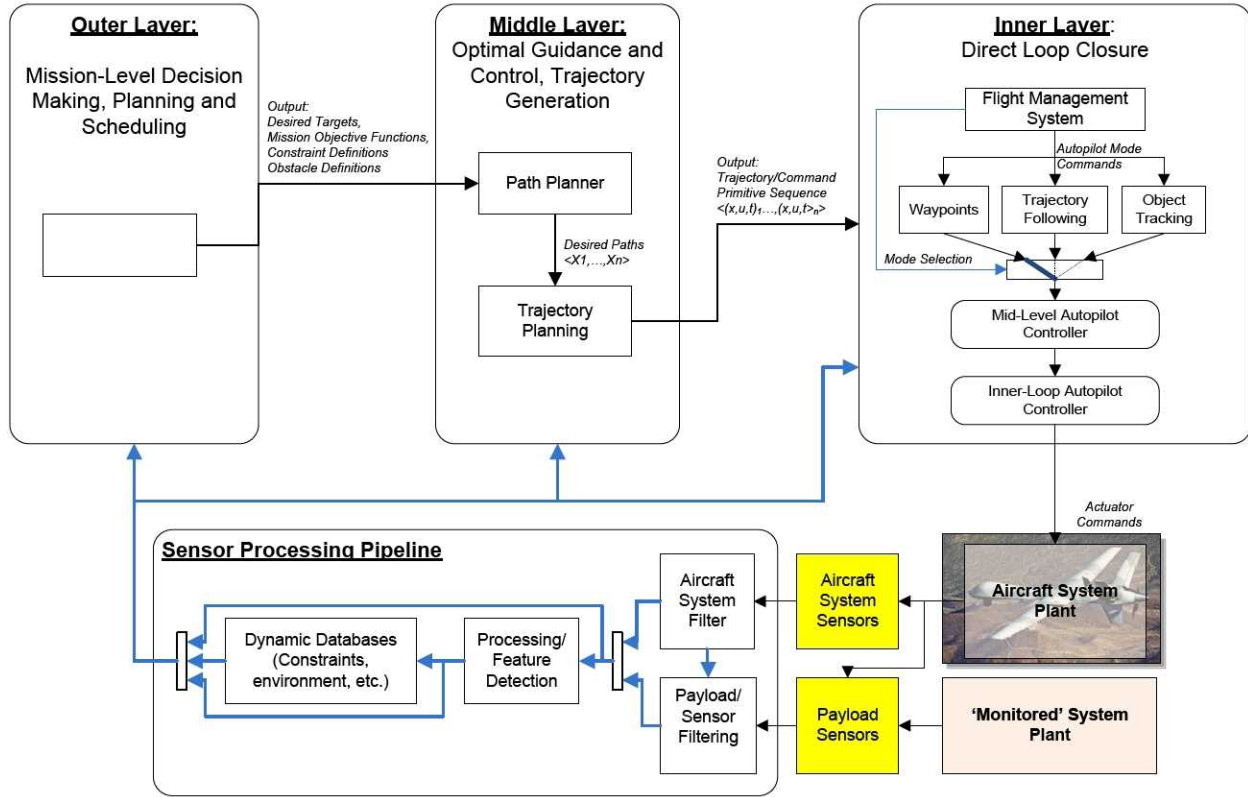


Figure 5. PDF-Wide System Architecture

The trajectory planning algorithms presented in this paper are focused on the mid-level loops, ignoring the mission planning and scheduling blocks (which are the focus of the outer layer), and abstracting the lower-level autopilot controller in the inner layer. The PDF middle layer architecture is shown in the figure below, grouped into three conceptual components. The knowledge system is responsible for maintaining knowledge and belief world, using sensor inputs, state information, apriori beliefs, environment models, etc. The trajectory planning and optimization system is responsible for plotting trajectories from one state to intermediate goal states given by the higher level mission planner. The trajectory sequence then is sent to the autopilot system. These three systems are cascaded to form the mid-level closed-loop system in the PDF architecture.



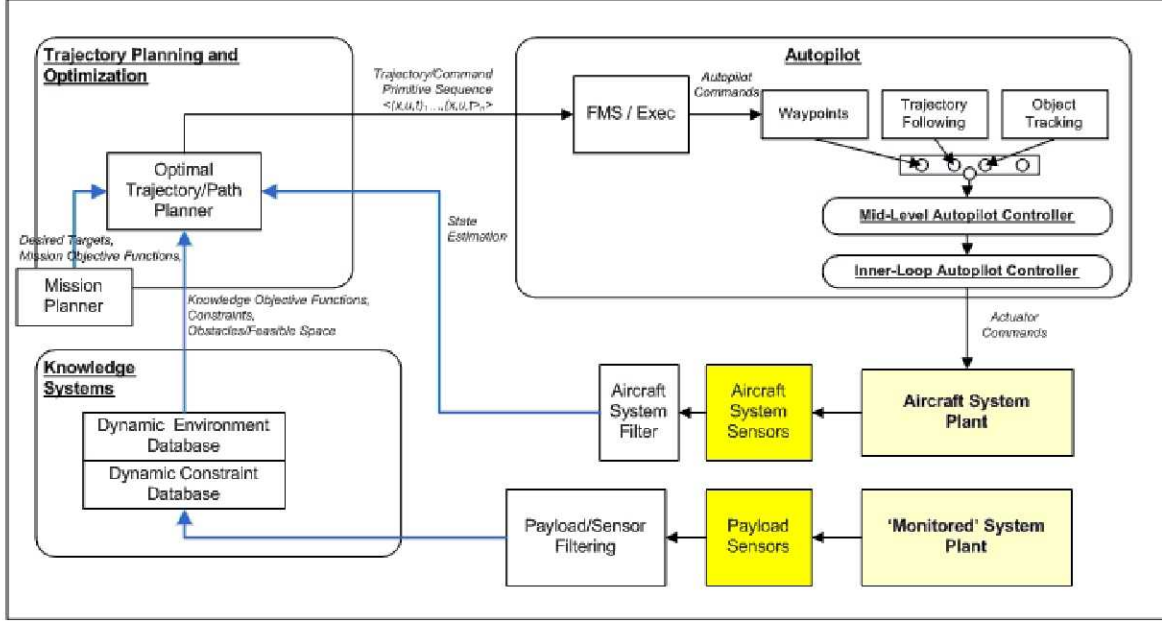


Figure 6. PDF Middle-Layer Architecture

## 2. Problem Formulation and Algorithm Outline

Consider a vehicle system  $P$  whose dynamics are governed by a series of non-linear differential equations of the following form, where  $x \in \mathbb{R}^n$ ,  $u \in \mathbb{R}^r$ ,  $y \in \mathbb{R}^p$ .

$$P: \begin{cases} \dot{x} = f(x, u, t) \\ y = h(x, u, t) \end{cases}$$

The model is augmented for vehicle or payload constraints, equality or inequality constraints in the set  $G = (G_e, G_i)$ , of the form

$$G_e(x, \dot{x}, u, t) = 0 \quad \text{and} \quad G_i(x, \dot{x}, u, t) \leq 0$$

Additionally, the system is subject to a set of boundary constraints  $B(x, t) \leq 0$  specified on the state variables of the system, and can be time dependent. Sensors and mission objectives can be modeled and input into this algorithm as integral cost functionals  $\phi_i(x, u, t)$ . A final state cost  $L(x_f, t_f)$  can also be added. The objective function can be specified as

$$J_i = L(x_f, t_f) + \sum_s \left( \int \phi_i(x, u, t) dt \right)$$

Given an initial state  $x_0$ , and a subset of the final state  $\tilde{x}_f$ , (where  $\tilde{x} \subseteq x$  is the goal space), find the continuous set of control inputs and the associated trajectories from  $x_0$  to  $\tilde{x}_f$  that minimize the objective function  $J$ , subject to constraints in  $G$  and  $B$ .

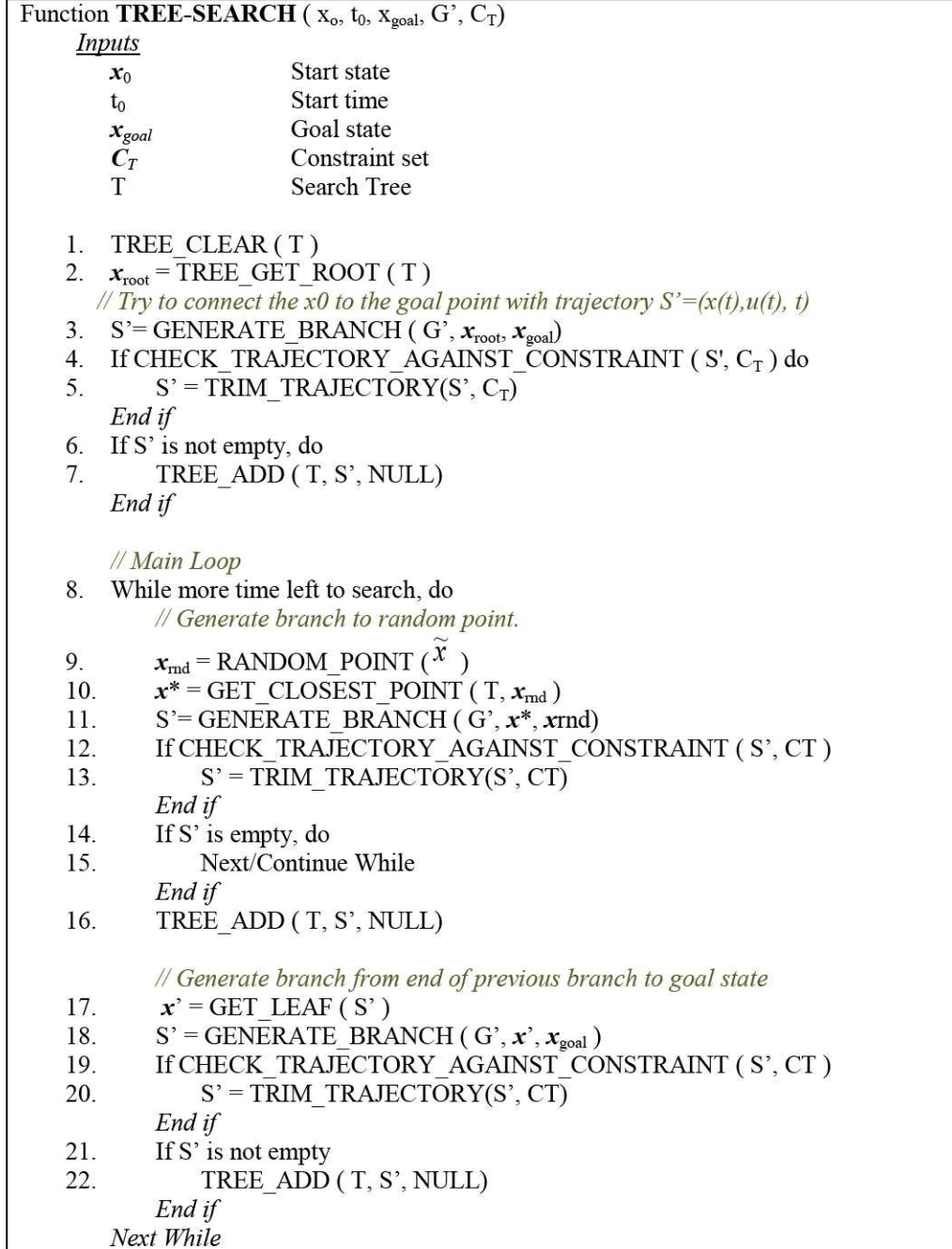
### A. General Approach

Our approach to solution of this problem is as follows.

1. Define the plant  $P$ , given by  $P: \begin{cases} \dot{x} = f(x, u, t) \\ y = h(x, u, t) \end{cases}$
2. Define the state space subsets. Define the goal space  $\tilde{x}$  as a subset of  $x$ , and define the search space  $\hat{x}$  as a subset of the state space that contains the goal space; i.e.,  $\tilde{x} \subseteq \hat{x} \subseteq x$ .
3. Define the augmented plant  $P'$  by incorporating the plant constraints  $G$  into the plant model.

4. Design a closed loop system  $G'$  – the branch trajectory controller system - that can control the adjointed plant  $P'$  from an initial state  $\mathbf{x}_0 \in \mathcal{X}$ , to a destination point in the goal space  $\tilde{\mathbf{x}}_f \in \tilde{\mathcal{X}}$ , utilizing the controller  $U$ .
5. Incorporate/implement  $G'$  in the online trajectory search algorithm outlined below.

The closed loop control system  $G'$  is used to compute trajectories that represent branches in a search tree, similar to Ref. 22. Our modified Tree Search Algorithm is as follows.



**Figure 7. TREE-SEARCH Algorithm**

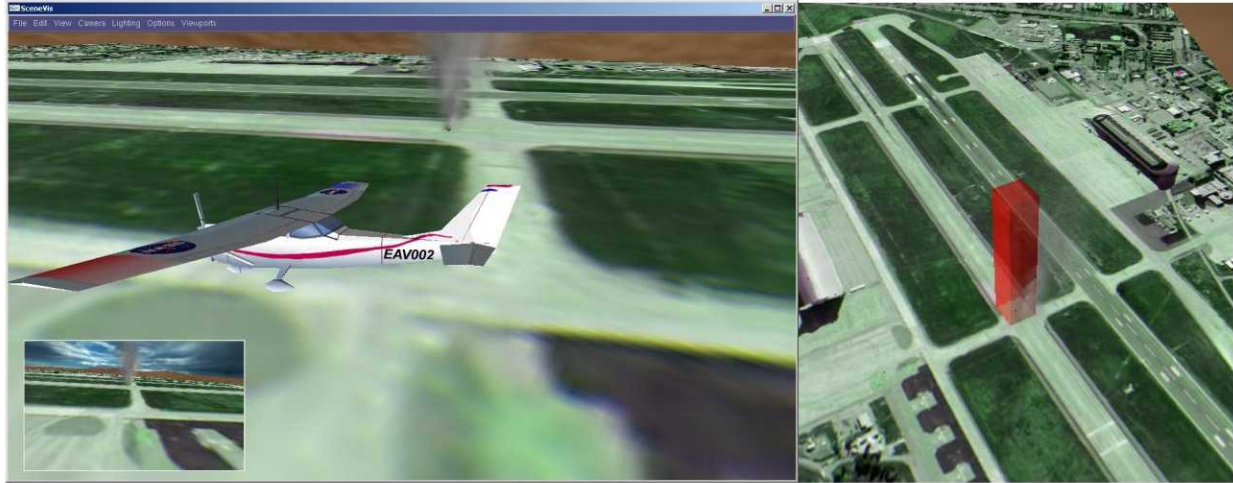
Algorithm **TRIM\_TRAJECTORY** ( $S', C_T$ )

23. Trim branch  $S'$  to remove points of collision and imminent collision with  $C_T$
24. Generate avoidance/recovery maneuver branch  $S^*$
25. If avoidance/recovery maneuver branch collides
26.     Discard  $S'$  and  $S^*$  branches.
- Else
27.     Add  $S^*$  to end of  $S'$
- End if

**Figure 8. TRIM\_TRAJECTORY Algorithm**

### B. Plume Pointing Problem Example

The trajectory generation system was implemented in simulation and tested as a navigation solution for the plume-pointing problem outlined in Figure 2. The aerial vehicle starts from a ground position with no concept of where the smoke plume is. The vehicle is constrained to an approved flight area. The vehicle system must observe the plume through body-fixed imagers, develop an internal model of the plume, and compute trajectories safely around the plume – within the approved flight area - that allows the vehicle to train the imagers on the plume in order to update its internal model. The simulation scenario was implemented in the Reflection Architecture, including a custom smoke particle generation rendering system, as shown in Figure 9.



**Figure 9. Simulation of a Smoke Plume Monitoring Experiment.**

*Body-fixed camera looking out the right side of the vehicle, under the wing. Box represents area to avoid.*

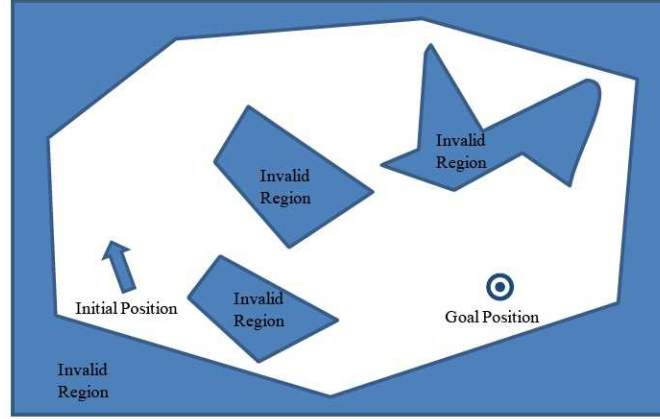
This phase of testing is only concerned with the trajectory generation system and trajectory following controller. To that extent, real-time sensor feedback from the camera sensors has not yet been integrated. In this experiment, the area of the plume that the vehicle wishes to navigate is fixed, and the areas to avoid are also fixed. The aircraft must compute trajectories that avoid the smoke plume, the ground, and stay within the approved airspace.

### C. Trajectory Generation Problem

The first part of the problem is to consider the constraints. Consider an aircraft system that must navigate from some initial state  $\mathbf{x}$  and reach a goal position  $\mathbf{P} \in \mathcal{R}^3$ . The aircraft must avoid obstacles while performing this maneuver, which are specified as a set of inequality trajectory constraints in  $C_T$ . Additionally, the aircraft dynamics are limited: the ailerons can only be deployed in a limited range, and the elevator rates are limited.

$$C_T = \{ \mathbf{obs}_1, C_{obs2}, \dots \}; \quad C_P = \left\{ \begin{array}{l} a_{\min} \leq \delta_{ail} \leq a_{\max} \\ \left| \frac{d\delta_{ele}}{dt} \right| \leq 0.5 \end{array} \right\}$$





**Figure 10. A Simple Trajectory Calculation Exercise**

The aircraft is required to make this maneuver while maximizing fuel efficiency. We choose not to have the goal position incorporated into the cost function  $\phi$  at the final time (terminal error), but rather this constraint will be handled by the search algorithm. We will use a quadratic form for the cost function, which was chosen for ease of implementation for later problems. Consider the following cost formulation, where the sensor pointing cost terms,  $L_{ds}$  and  $L_{dr}$ , are defined in section D.

$$J = \int_0^{t_f} \left( x^T A_J x + u^T B_J u + L_{ds} + L_{dr} \right) d\tau \quad ; \quad B_J = \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & 10 \end{bmatrix}$$

The trajectory problem is stated as follows. Find the continuous set of control inputs and the associated trajectories from  $x_0$  to  $\tilde{x}_f$  that minimize the objective function  $J$ , subject to constraints  $C$  and  $S$ .

### 1. Plant Definition

First, we must define the plant. This solution will use the dynamics of the Exploration Aerial Vehicle (EAV) at NASA Ames Research Center as the model. The EAV is a small unmanned aircraft designed for subscale testing of vehicle control systems. The linear lateral and longitudinal dynamics for the EAV at 25 m/s and 130m altitude were estimated through a series of flight tests. The system states and inputs are given by

$$x = \begin{bmatrix} x_{lat} & x_{lon} \end{bmatrix}^T \in \mathbb{R}^8 \quad ; \quad u = \begin{bmatrix} u_{lat} & u_{lon} \end{bmatrix}^T \in \mathbb{R}^4$$

The lateral dynamics for the aircraft were estimated to be

$$x_{lat} = \begin{bmatrix} \beta & \Delta p & \Delta r & \Delta \phi \end{bmatrix}^T \quad ; \quad u_{lat} = \begin{bmatrix} \delta_{ail} & \delta_{rdr} \end{bmatrix}^T$$

$$A_{lat} = \begin{bmatrix} -1.0176 & 0.0728 & -0.8887 & 0.2918 \\ -16.7704 & -5.2969 & 1.2992 & 0 \\ 13.7306 & -0.3138 & -1.3475 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \quad ; \quad B_{LAT} = \begin{bmatrix} 0.3404 & 0.5297 \\ 55.2736 & 4.6948 \\ -1.4038 & -9.7679 \\ 0 & 0 \end{bmatrix}$$

The longitudinal mode were estimated to be the following.

$$x_{lon} = \begin{bmatrix} v_T & \Delta \alpha & \Delta q & \Delta \theta \end{bmatrix}^T \quad ; \quad u_{lon} = \begin{bmatrix} \delta_{ele} & \delta_{thr} \end{bmatrix}^T$$

$$A_{lon} = \begin{bmatrix} -1.409 & -0.040515 & 0 & -0.096803 \\ -0.0281 & -4.0540 & 1.2896 & 0 \\ -0.0566 & -20.9832 & -1.6784 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad ; \quad B_{lon} = \begin{bmatrix} -0.0699 & 3.24 \\ -0.8405 & -0.03057 \\ -25.99 & -0.07424 \\ 0 & 0 \end{bmatrix}$$

In order to incorporate position into this system, we augment the system with the state vector  $\mathbf{p}=[x \ y \ z]^T$ , where  $\mathbf{p}$  is the position vector of the aircraft in a north/east/down coordinate system. We assume a local flat inertial non-rotating earth assumption, and augment the system with the kinematic relationship, where  $\mathbf{p}, \Phi \in \mathbb{R}^3$ , and  $\psi$  is the Euler yaw angle.

$$\mathbf{p} = \begin{bmatrix} x_N & y_E & z_D \end{bmatrix}^T ; \quad \frac{d}{dt} \mathbf{p} = \Phi \mathbf{v}_T$$

$$\frac{d}{dt} \psi = \begin{bmatrix} \frac{\sin \phi}{\cos \theta} & \frac{\cos \phi}{\cos \theta} \end{bmatrix} \begin{bmatrix} q \\ r \end{bmatrix}$$

Although this is written as a linear relationship, the  $\Phi$  is a nonlinear function of the Euler angles and wind axis angles, given by

$$\Phi(\phi, \theta, \psi, \alpha, \beta) = R_{ba2ned}(\phi, \theta, \psi) R_{wi2ba}(\alpha, \beta) \begin{bmatrix} 0 & 0 \end{bmatrix}^T$$

The transformations from body axis (BA) to wind axis (WI) to world axis (NED) is given by

$$R_{wi2ba}(\alpha, \beta) = R_y(\alpha) R_z(\beta)$$

$$R_{ba2ned}(\phi, \theta, \psi) = R_z(\psi) R_y(\theta) R_x(\phi)$$

## 2. Space Definitions

The dynamic/kinematic system is formulated with a state space vector  $\mathbf{x} \in \mathbb{R}^{12}$  and control input space  $\mathbf{u} \in \mathbb{R}^4$ , where the system model's state and input vectors were given by

$$\mathbf{x} = \begin{bmatrix} \psi & x_{lat} & x_{lon} \end{bmatrix}^T ; \quad \mathbf{u}_{lat} = \begin{bmatrix} \delta_{ail} & \delta_{rdr} & \delta_{ele} & \delta_{thr} \end{bmatrix}$$

Here,

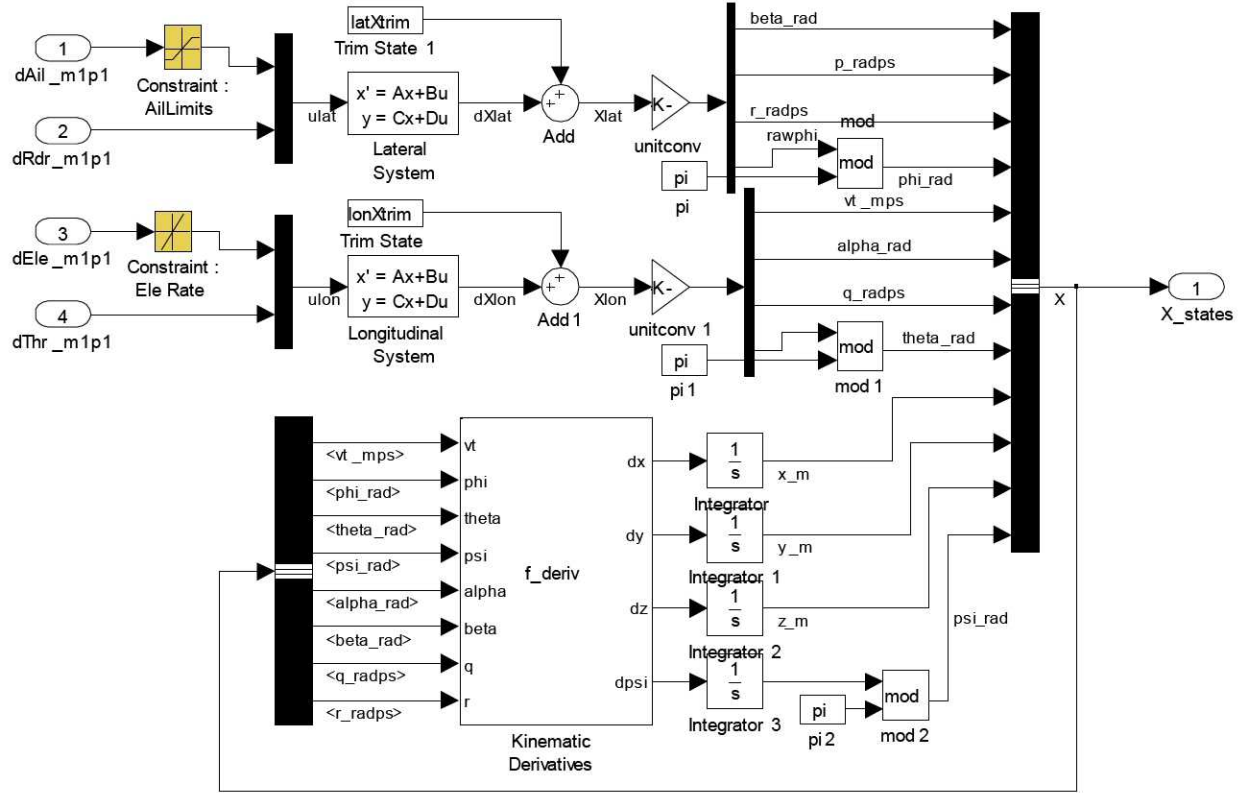
$$x_{lon} = \begin{bmatrix} v_T & \Delta \alpha & \Delta q & \Delta \theta \end{bmatrix}^T ; \quad x_{lat} = \begin{bmatrix} \beta & \Delta p & \Delta r & \Delta \phi \end{bmatrix}^T$$

This problem is formulated with a goal state given as the final desired Cartesian position, independent of all other states. The search space and goal space are defined as

$$\tilde{\mathbf{x}} = \hat{\mathbf{x}} = \begin{bmatrix} x_N & y_E & z_D \end{bmatrix}^T$$

## 3. Augmented Plant

Next we develop the augmented system  $P'$ . Our original plant  $P$  can be modeled in block diagram form as shown in Figure 11. Constraints  $C_P$  have been added to the input, and are shown with a filled background.



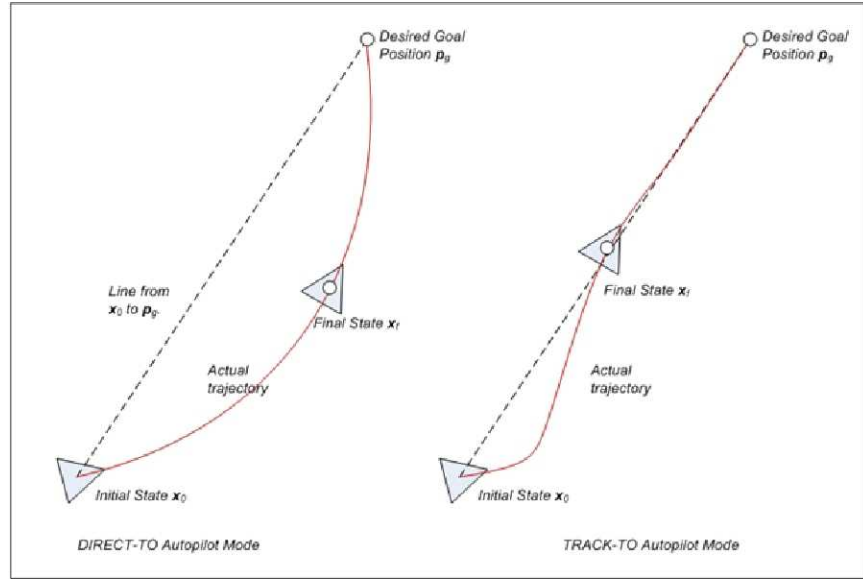
**Figure 11. Vehicle Plant Model**

#### 4. Branch Trajectory Controller

The next step in our approach is to design a closed-loop controller/plant system  $G'$  that can control the adjoined plant  $P'$  from an initial state  $x_0 \in X$ , to a destination point in the goal space  $\tilde{x}_f \in \tilde{X}$ .

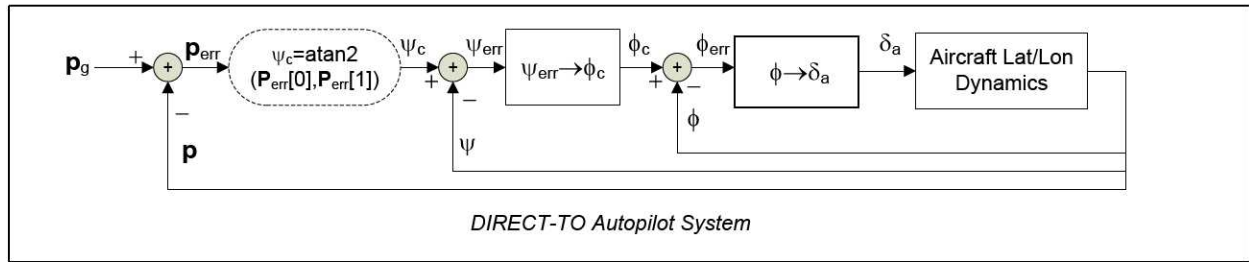
There are a few different autopilot system modes we can use to implement the trajectory computation algorithm. We will base this control problem based on the existing heading-command autopilot system designed for the EAV. This autopilot system has the option of implementing either a "track-to" or "direct-to" autopilot mode.





**Figure 12. Two Candidate Autopilot Control Modes for Waypoint Following**  
*Track-To and Direct-To from initial state to desired goal position*

The ‘direct-to’ autopilot mode will guide the aircraft directly to the waypoint based on heading command from the current aircraft state. This autopilot design is a little simpler, as a single arctan evaluation is needed to compute the heading error at any point in time. An autopilot block diagram is shown below. Here, roll command is used in favor of roll-rate command for increased margins and implementation on the real EAV system.



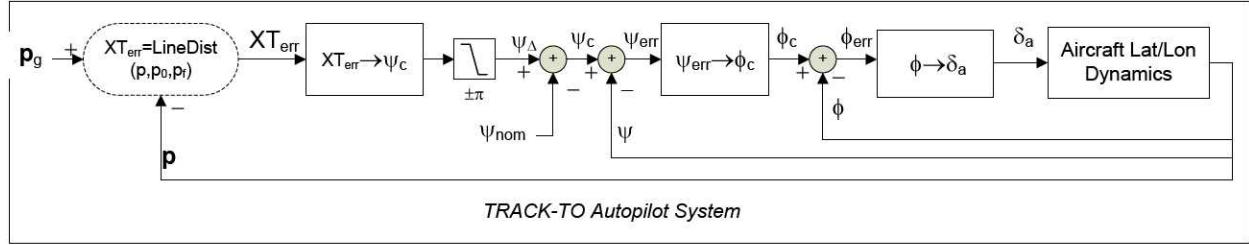
**Figure 13. Direct-To Autopilot Design**

Each block shown in the figure above is a controller block which can vary in form; here we consider the classic 3-parameter PID form for the transfer function.

$$Block_{A \rightarrow B} = \frac{B(s)}{A(s)} = K_p + \frac{K_i}{s} + K_d s$$

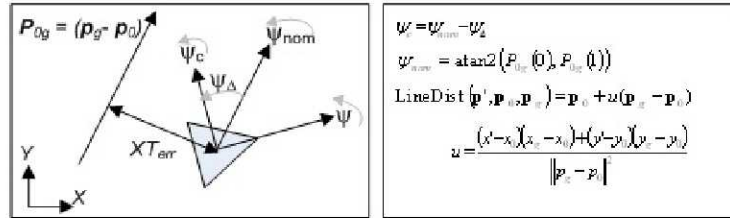
**Figure 14. Transfer function for Transform Blocks (Simple PID Block)**

The TRACK-TO autopilot system is similar in form to a DIRECT-TO autopilot system, except for the additional calculation of a cross-track error as an error signal to drive to zero. The block diagram is shown below.



**Figure 15. TRACK-TO Autopilot Design**

The crosstrack error ( $XT_{err}$ ) is used to calculate a delta heading angle ( $\psi_{\Delta}$ ) which is the deviation from the nominal heading angle ( $\psi_{nom}$ ). This relationship is shown graphically below. Also, the line distance calculation block is also shown.

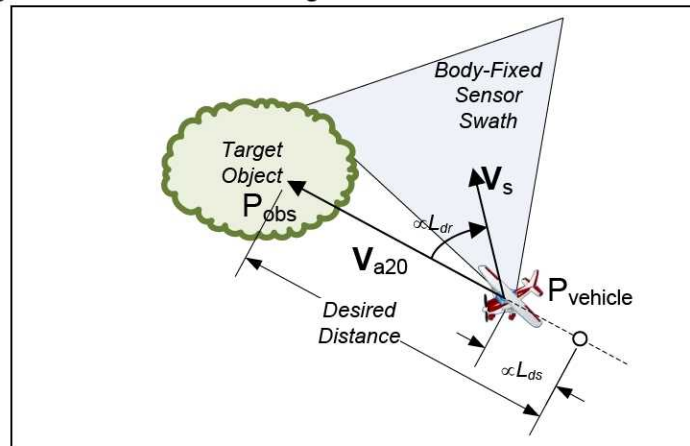


**Figure 16. Details of XTE Calculations**

Our exact vehicle system could be used for these computations, but effectively a heading control system will turn the complex 6-DOF vehicle system with controller into some sort of low-pass filter. The next step is to develop the autopilot and simulation test with this reduced model. The Matlab Simulink implementation of this controller is shown in Figure 19 below. The diagrams in Figure 20 shows the output of a test script, which generated a random path, and a set of random initial conditions for the aircraft. The resultant trajectories computed by the closed-loop system are shown.

#### D. Sensor Models

The implementation in Figure 19 contains a “Cost Model” block. This block contains the computations for compute the derivative of the cost function at any point in time and state of the vehicle system. Three different cost models are included in this test. The first cost model penalizes sensor pointing inaccuracy of the sensor towards a specified target, as well as pointing distance, as illustrated in Figure 17 below.



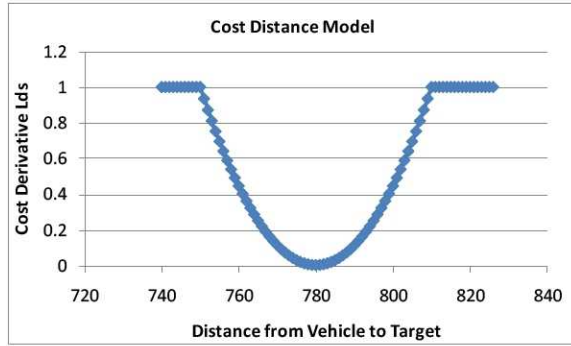
**Figure 17. Cost Models Used for Sensor Pointing Accuracy**

Let a sensor axis frame  $F_s$  be defined as a fixed frame relative to the aircraft, with the y-axis pointing in the direction of the sensor. Let  $R_{s2l}$  represent the 3x3 transformation matrix from  $F_s$  to  $F_l$ , where  $F_l$  is the local axis system. Let  $R_{l2w}$  represent the rotation matrix from  $F_l$  to  $F_w$ , where  $F_w$  is the world axis system (here we assume selection of an

appropriate world axis system, such as a world-fixed east/north/up axis system). Let the position of the vehicle be given by  $\mathbf{P}_{vehicle}$ , and the position of the observed point be specified by  $\mathbf{P}_{obs}$ . Then the directional cost incurred from point inaccuracy,  $L_{dr}$ , is computed by the following equation. The resulting costs for random trajectories – when only  $L_{dr}$  directional costs are taken into account - are shown in Figure 21.

$$L_{dr} = \frac{dC}{dt} \Big|_{Direction} = \frac{1 - \left( \frac{\mathbf{P}_{obs} - \mathbf{P}_{vehicle}}{\|\mathbf{P}_{obs} - \mathbf{P}_{vehicle}\|} \cdot \mathbf{R}_{s2l} \right) \cdot \begin{bmatrix} 1 & 0^T \end{bmatrix}}{2}$$

In addition to pointing accuracy, the sensors need to maintain a specific distance to the target. This is modeled as a curve that falls off with the square of the distance to the target position. Let  $R_{des}$  be the desired distance from the sensor to the phenomena, and let  $R_{dev}$  be the deviation distance, then the resulting equation is and graph is shown in Figure 18 below. The resulting costs for random trajectories – when only  $L_{ds}$  distance costs are taken into account - are shown in Figure 21.



$$L_{ds} = \frac{dC}{dt} \Big|_{Distance} = MIN \left( 1, \frac{\|\mathbf{P}_{obs} - \mathbf{P}_{vehicle}\| - R_{des}}{R_{dev}^2} \right)$$

**Figure 18. Cost Distance Model and Equation**  
Shown for  $R_{des} = 780$ ,  $R_{dev} = 30$



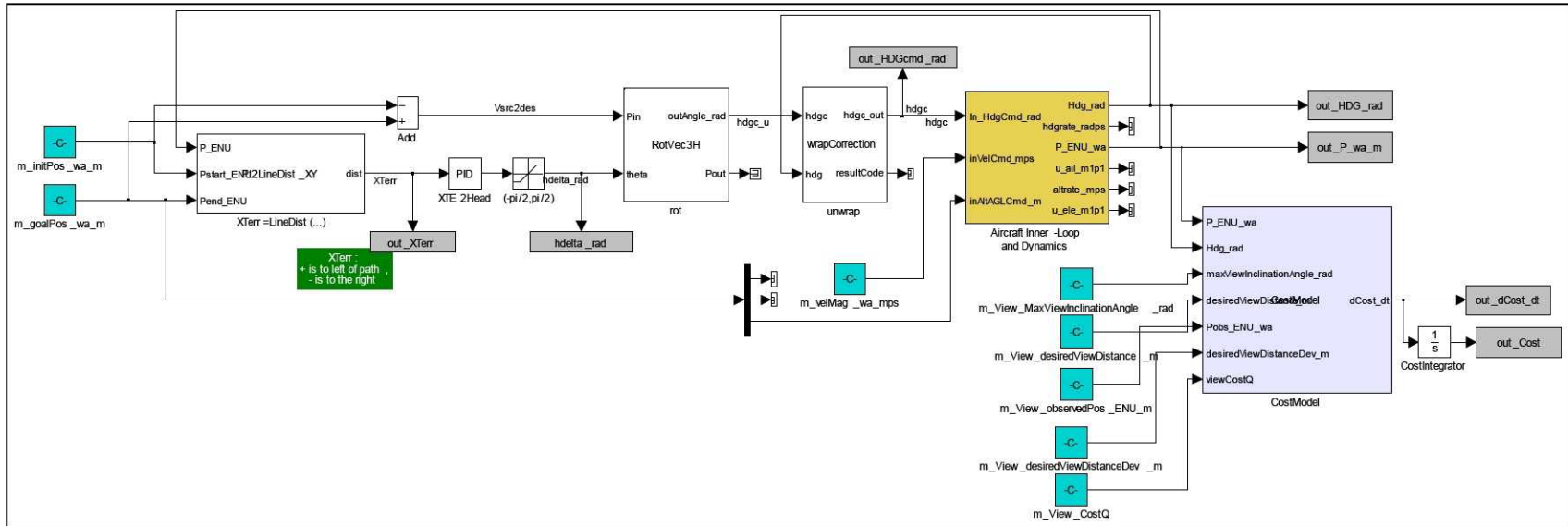
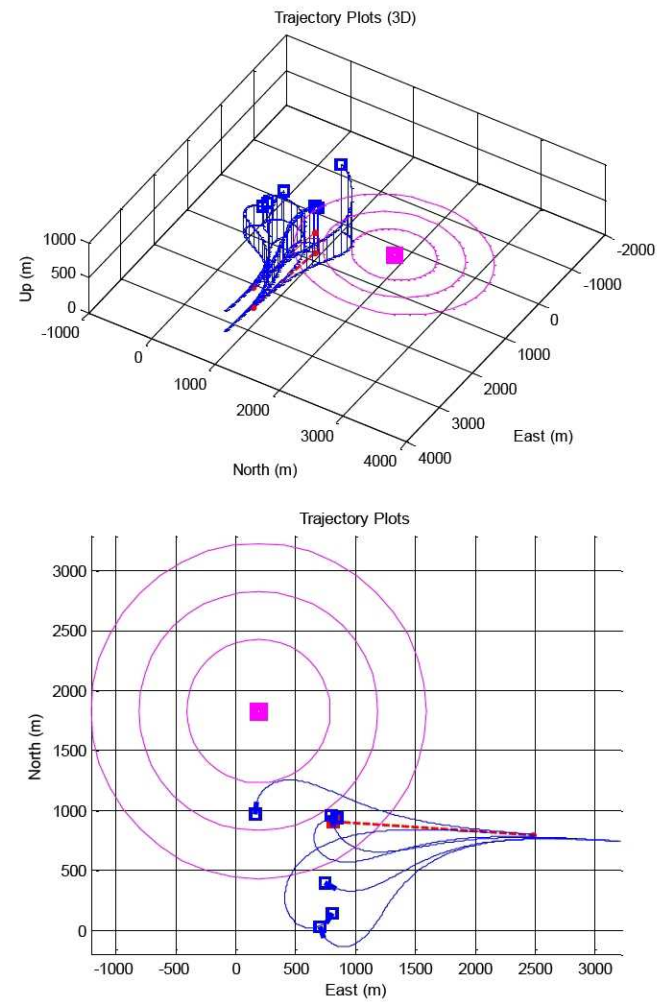
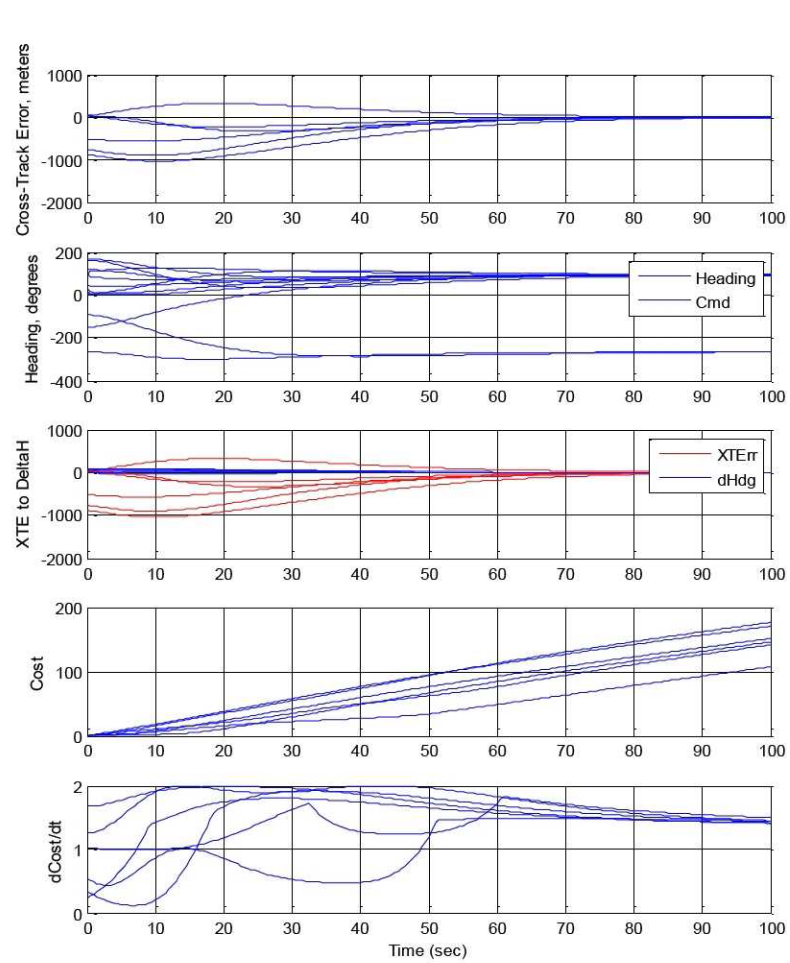
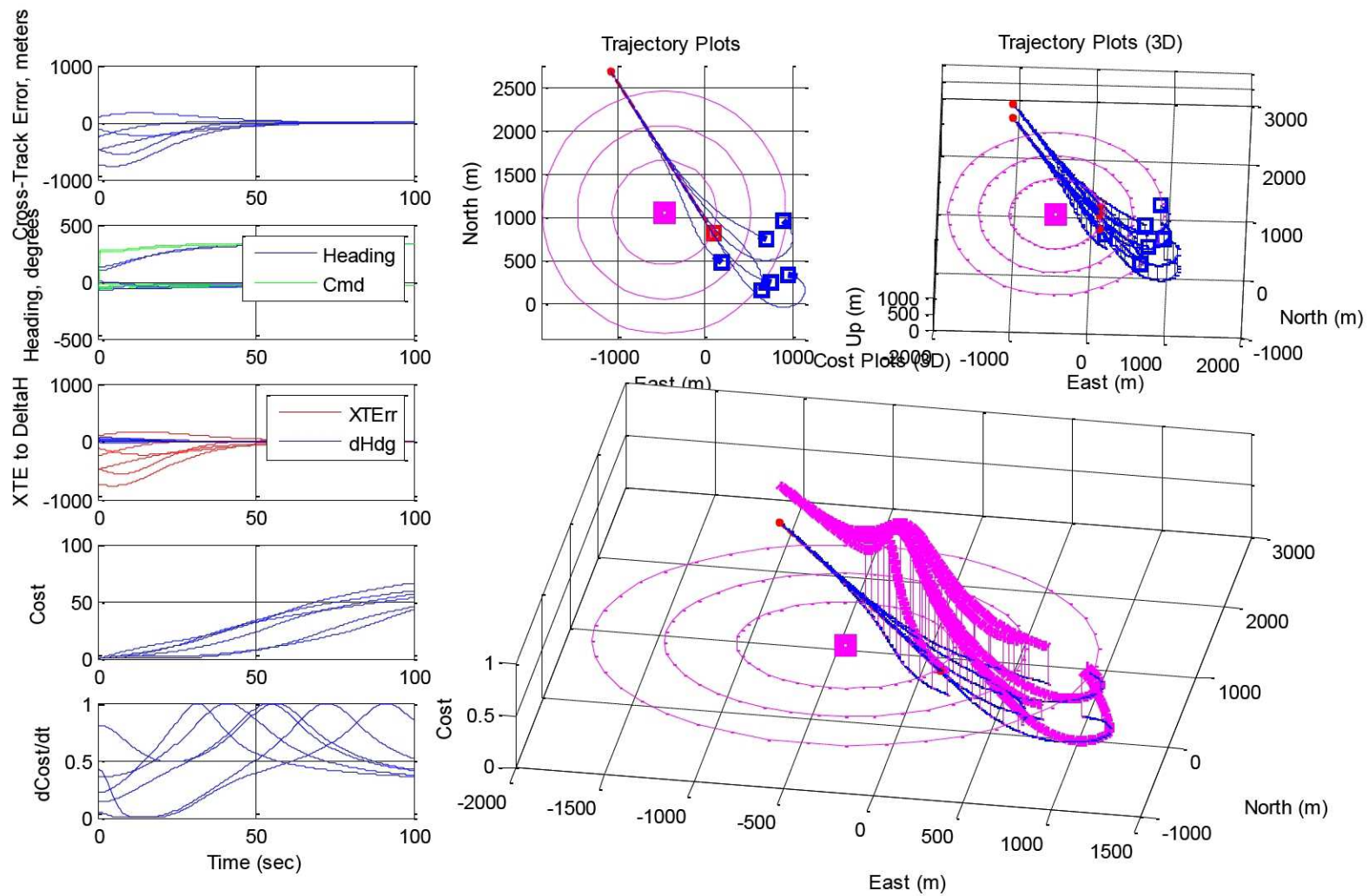


Figure 19. Simulink Model for Trajectory Control System

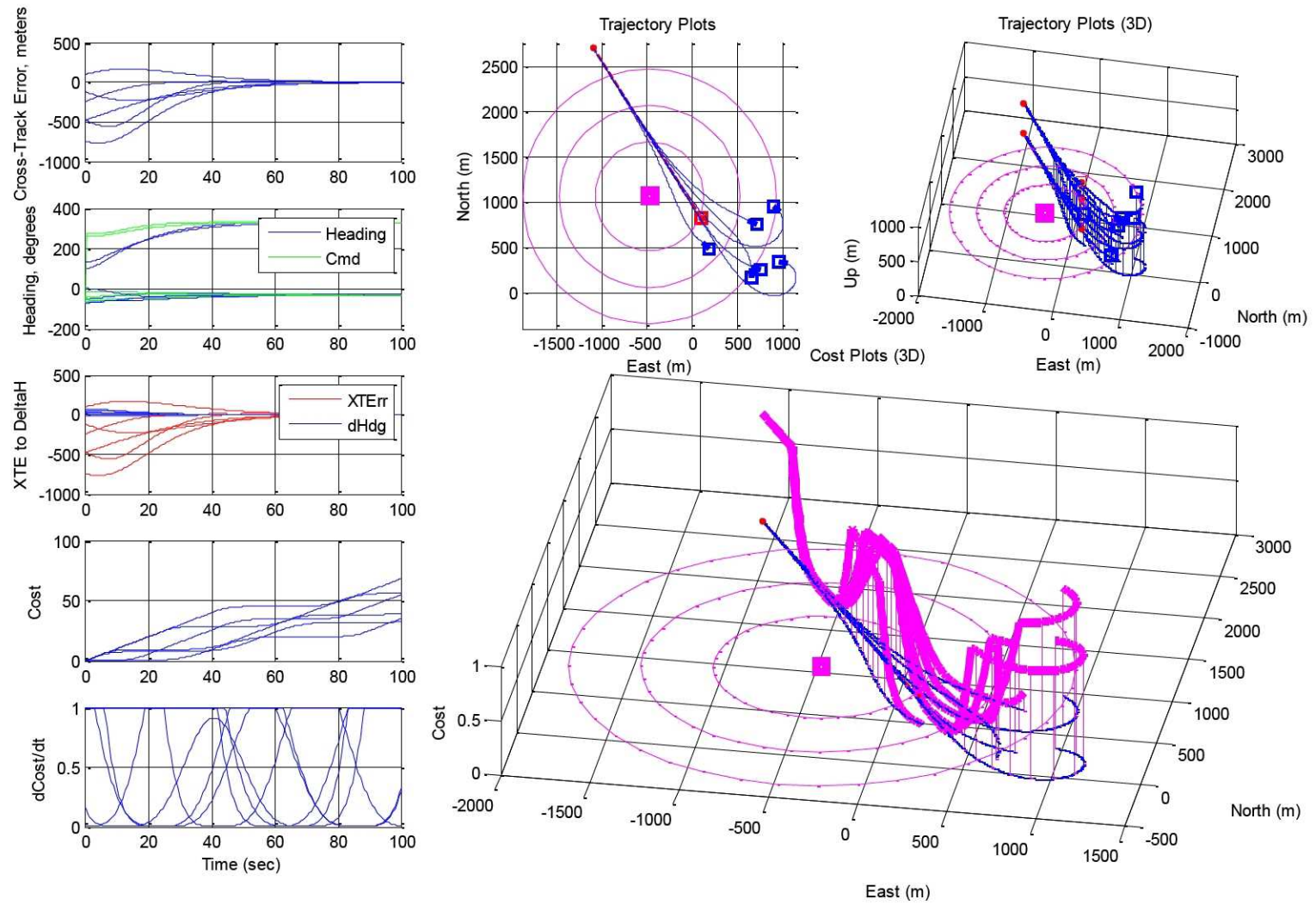


**Figure 20. Trajectory Response for Randomly Generated Points**



**Figure 21. Pointing Cost Derivative for Trajectories.**  
*Sensor is pointing to the right of the vehicle.*

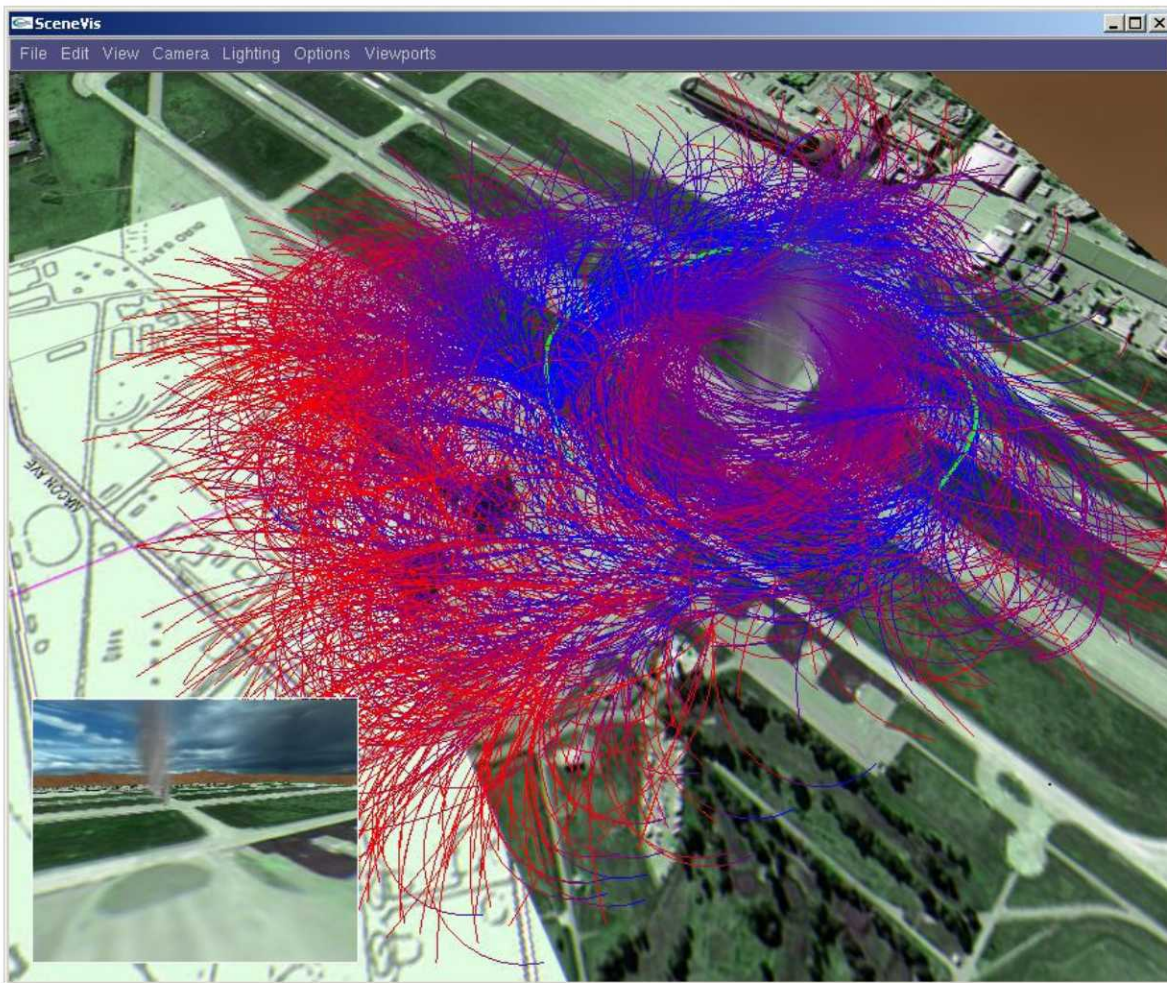




**Figure 22. Distance Cost Derivative for Vehicle Trajectories.**

### 3. Simulation Results

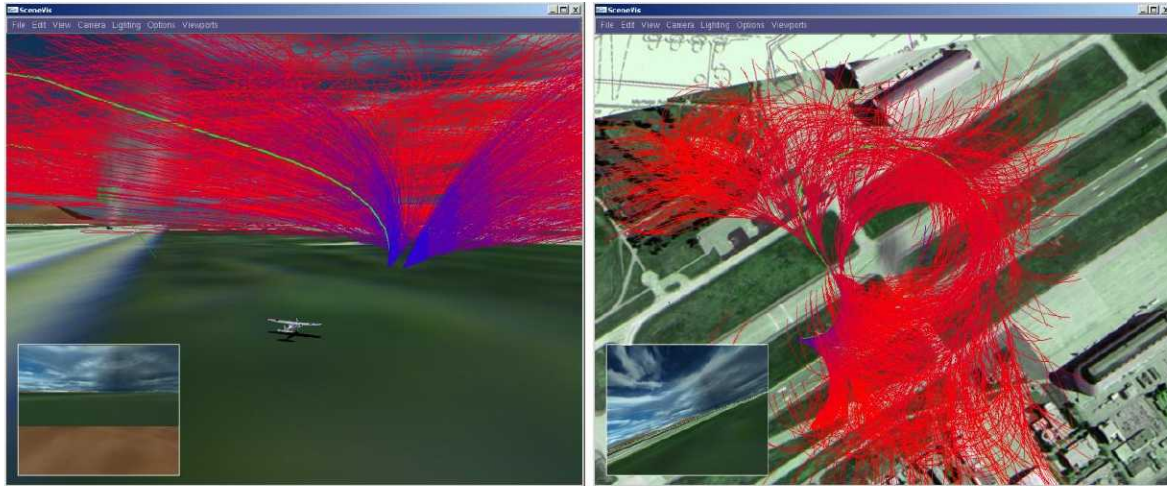
The controller was tested simulation using the Reflection Architecture, with the entire mission scenario (except for landing) being controlled by the trajectory planner. The Matlab Simulink model was converted into C++ code using the Real-Time Workshop Embedded Coder product from Mathworks. The tree-branching algorithm was allowed to plan for 5 seconds. The planning system looked ahead of the current time by 10 seconds, and repeated the planning algorithm every 20 seconds. The system was tested on several systems with the renderer, visualizations, and full UAV simulation being run on the same computer. The test was run on an Intel Core 2 X9650, 3 GHZ, 3GB RAM. The screenshots shown in Figure 23 shows a typical screenshot of the entire system running, with the visualization rendering 500 branches, and associated costs shown in color: red represents higher costs, blue represents lower costs. The trajectory planner immediately guides the aircraft to the optimal track, which is a right hand turn to allow sensor monitoring of the plume, at the distance  $R_{dev}$  from the target position. All trajectories found avoid the plume's bounding volume and guides the aircraft safely inside the approved fly zone. To reliability of the system in real-time, the flight management system can be disengaged and the aircraft can be hand piloted away from the plume and commanded to a hazardous orientation; as soon as the autopilot and planner is reengaged, the trajectory planner immediately computes the appropriate control inputs and trajectories to recover from the unusual orientation and guides the aircraft safely back to the optimal track. For a test lasting 30 minutes, the search algorithm averaged 48,202 nodes (states) added to the tree, with 3435 branches, taking 15,002,030,502 CPU clock cycles. The trajectory planner was running in its own thread and had full utilization of one of the processors.



**Figure 23. Screenshot of Planner and Branches.**

*Branches are shown as red/blue lines, where red represents higher costs and blue represents lower costs. The green line represents the trajectory selected.*





**Figure 24. Takeoff Trajectories (Overhead shot shown on right).**

*Costs build up fast because of the distance to the plume, but the selected trajectory immediately takes the vehicle to the 'optimal' path for sensor pointing.*

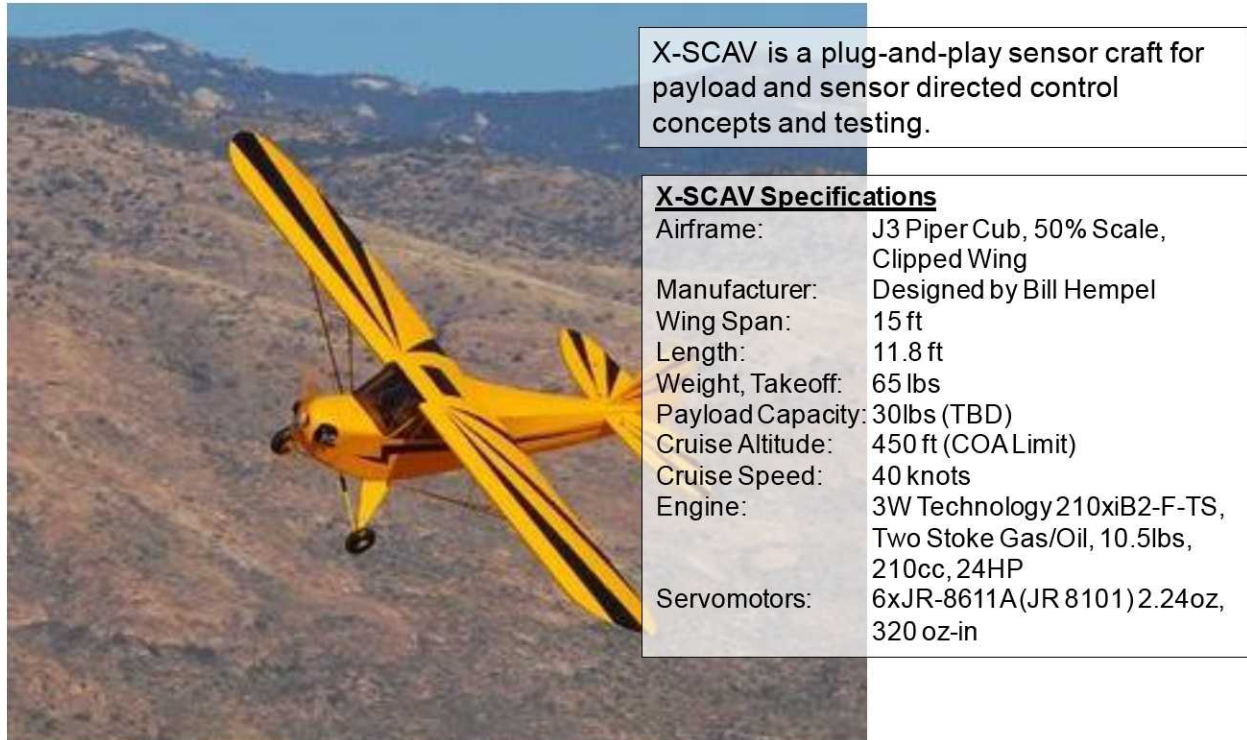
#### 4. Conclusion and Future Work

The results of the simulation were promising. The vehicle was able to successfully navigate the area and keep the smoke plume in view 100% of the time after the aircraft establishes its position in the 'optimal' track. Constraint satisfaction was maintained for 100% of the time after takeoff, and constraints (plume collision volume, ground volume, and approved airspace volume) were never violated.

As follow up work on this project, this simulation will be extended to complete the mission objectives completely autonomously.

1. Integrate the camera sensor processing module and environment knowledge database to close the loop fully around this scenario, where the plume position is not known, and the mission is executed fully autonomously.
2. Integrate a higher fidelity non-linear 6-DOF model of our UAV for trajectory planning.
3. Develop more accurate tracking with a linearized trajectory controller to follow the trajectories.
4. Extend the trajectory generation algorithm to include swappable cost modules for different sensors, and multiple sensors.
5. Extend the generation algorithm to select from several different closed-loop control systems, based on a heuristic.
6. Perform a point-sampling maneuver into the 'less-dense' areas of the plume, and a ground sensing maneuver over a specified target area.
7. Flight test these algorithms at NASA Ames Research Center/Moffett Airfield on the X-SCAV (eXperimental Sensor Controlled Aerial Vehicle) plug-and-play UAV (see Figure 25).





**Figure 25. X-SCAV UAV Platform Specifications**

## 5. References

- <sup>1</sup> C. Ippolito, M. Fladeland, Y Yeh. Applications of Payload Directed Flight. 2009 IEEE Aerospace Conference, March 7-14, 2009, Big Sky, Montana
- <sup>2</sup> C Yuhas, Atmospheric Composition (Troposphere) - Science Focus Work Group Products, INTEX Special Session of NASA Science Missions of the Future Workshop, July 13-15, 2004, New Hampshire
- <sup>3</sup> Dunlea, E. et al., Evolution of Asian aerosols during transpacific transport in INTEX-B, Atmos. Chem. Phys. Discuss., 8, 15375-15461, 2008.
- <sup>4</sup> E. Frew, T. McGee, Z. W. Kim, X. Xiao, S. Jackson, M. Morimoto, S. Rathinam, J. Padial, R. Sengupta, "Vision-Based Road-Following Using a Small Autonomous Aircraft," *Proc. IEEE Aerospace Conference*, 2004.
- <sup>5</sup> S. Rathinam, Z. Kim, A. Soghikian, R. Sengupta "Vision Based Following of Locally Linear Structures Using an Unmanned Aerial Vehicle", *Proc. 44th IEEE Conference on Decision and Control*, 2005.
- <sup>6</sup> S. Park, J. Deyst, and J. P. How, "A New Nonlinear Guidance Logic for Trajectory Tracking", *AIAA Guidance, Navigation, and Control Conference*, 2004.
- <sup>7</sup> H. Kim, D. H. Shim, S. Sastry, "Nonlinear Model Predictive Tracking Control for Rotorcraft-based Unmanned Aerial Vehicles," *Proc. of the American Control Confer.*, 2002.
- <sup>8</sup> R. Frezza, G. Picci, S. Soatto, "A Lagrangian Formulation of Nonholonomic Path Following," *The Confluence of Vision and Control*, Springer Verlag, 1998.
- <sup>9</sup> A. Matsuura, S. Suzuki, M. Kono, A. Sakaguchi, "Lateral Guidance Control of UAV using Feedback Error Learning," *AIAA Infotech@Aerospace*, Rohnert Park, CA 2007.
- <sup>10</sup> S. Rathinam, P. Almeida, Z. W. Kim, S. Jackson, A. Tinka, W. Grossman, R. Sengupta, "Autonomous Searching and Tracking of a River Using an UAV", *Proc. of the American Control Conference*, 2007.
- <sup>11</sup> P. Baker, A. Kahn, B. Kamgar-Parsi, J. Kellogg, "Optical Guidance for UAV Following of Shorelines," *AIAA Guidance, Navigation, and Control Conf.*, 2007.
- <sup>12</sup> J. Lee, et. al., "Strategies of Path-Planning for a UAV to Track a Ground Vehicle," *AINS Conf.*, 2003.
- <sup>13</sup> C. Dixon, E. Frew, B. Argrow, "Radio Leashing of an Unmanned Aircraft," *AIAA Infotech@Aerospace*, VA, Arlington, VA 2005.
- <sup>14</sup> K. Savla, E. Frazzoli, F. Bullo, "On the point-to-point and Traveling Salesperson Problems for Dubins' Vehicle," in *American Control Conference*, 2005.
- <sup>15</sup> X. Ma, D. A. Castanon, "Receding Horizon Planning for Dubins Traveling Salesman Problems," *Proc. IEEE Conf. on Decision and Control*, 2006.
- <sup>16</sup> L. E. Dubins, On Curves of Minimal Length with a Constraint on Average Curvature and with Prescribed initial and Terminal Positions and Tangents, *American Journal of Mathematics*, Vol 79, 1957.

- <sup>17</sup> T. Fraichard, A. Scheuer, "From Reeds and Shepp's to Continuous-Curvature Paths," *IEEE Transactions on Robotics*, Vol. 20, No. 6, December 2004.
- <sup>18</sup> A. Scheuer, Ch. Laugier, "Planning Sub-Optimal and Continuous-Curvature Paths for Car-Like Robots," *Proceedings of the 1998 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Victoria, B.C., Canada, October 1998.
- <sup>19</sup> J. K. Howlett, T. W. McLain, and M. A. Goodrich, "Learning Real-Time A\* Path Planner for Unmanned Air Vehicle Target Sensing," *Journal of Aerospace Computing, Information, and Communication*, Vol. 3, March 2006.
- <sup>20</sup> J. J. Kehoe, A. S. Watkins, and R. Lind, "Trajectory Generation for Effective Sensing," *AIAA Guidance, Navigation, and Control Conference and Exhibit*, Hilton Head, SC, 2007.
- <sup>21</sup> E. Frazzoli, M. Dahleh, and E. Ferron, "Real-time motion planning for agile autonomous vehicles," in Proceedings of the AIAA Guidance, Navigation, and Control Conference, (Denver, CO), August 2000. AIAA Paper No. AIAA-2000-4056.
- <sup>22</sup> S.M. LaValle and J.J. Kuffner. Randomized kinodynamic planning. In *Proc. IEEE Int'l Conf. on Robotics and Automation (ICRA'99)*, Detroit, MI, May 1999.
- <sup>23</sup> L. F. Bertuccelli, J. P. How, "Robust UAV Search for Environments with Imprecise Probability Maps," *Proc. IEEE Conf. on Decision and Control*, 2005.
- <sup>24</sup> Y. Jin, A. Minai, M. Polycarpou, "Cooperative Real-Time Search and Task Allocation in UAV Teams," *IEEE Conf. on Decision and Control*, 2003.