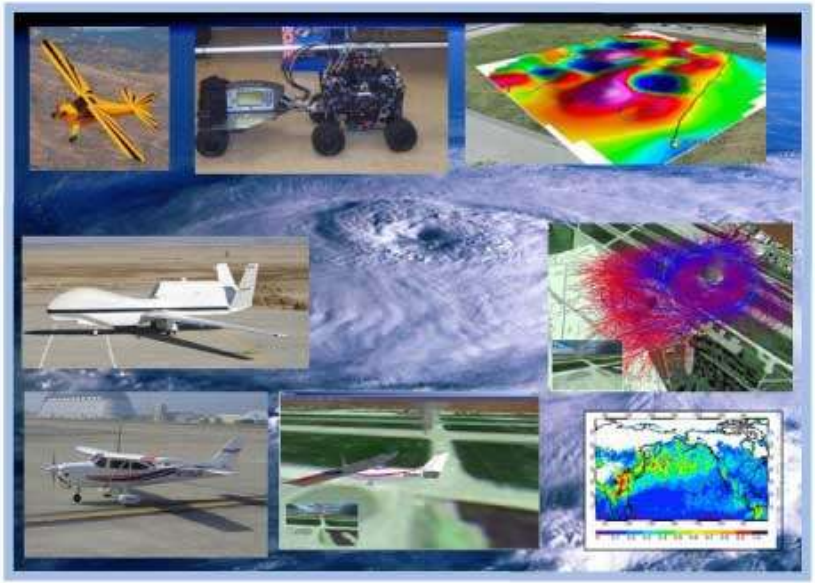
Western Earth Surface Processes

Seminar Series

Intelligent Payload-Directed Vehicle Autonomy for Earth Science Missions



Corey Ippolito
Researcher, Payload Directed Flight Task Lead
NASA Ames Research Center, Moffet Field

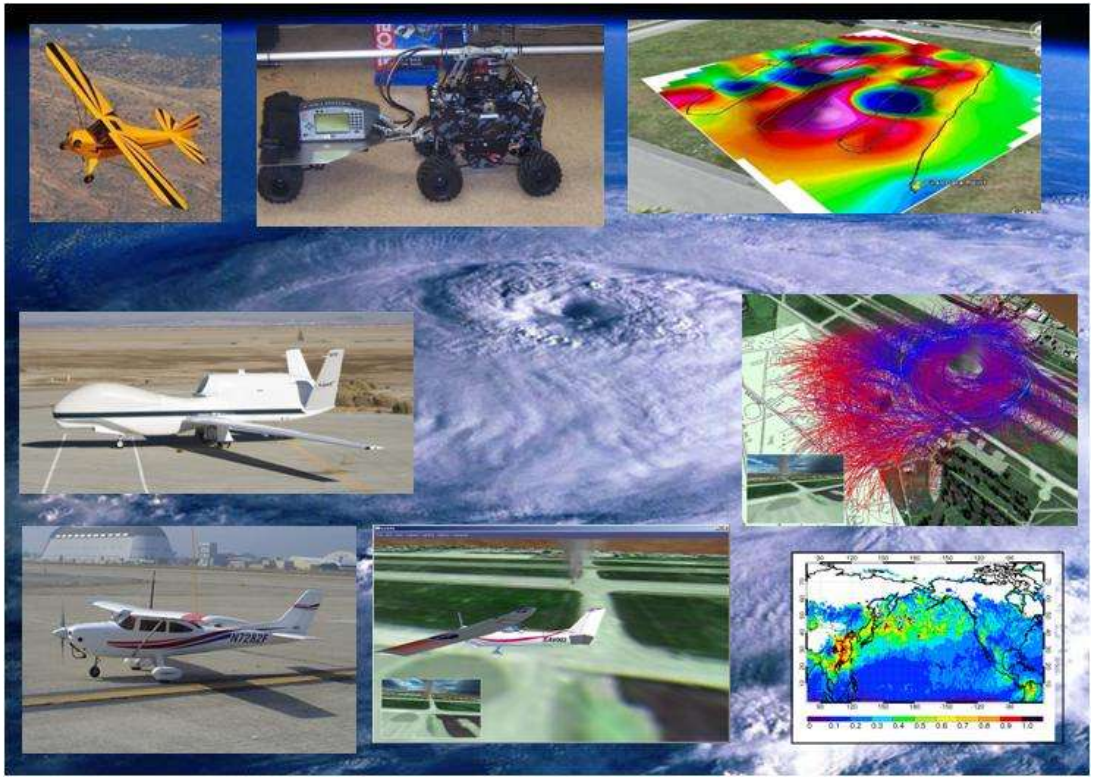
Thursday, March 26, 2:00 p.m.
U. S. Geological Survey, Menlo Park
Building 15, Room 3245

Intelligent Payload Directed Vehicle Autonomy for Earth Science Applications

Corey Ippolito
Intelligent System Division, Code TI
NASA Ames Research Center
Moffett Field, CA 94035
corey.a.ippolito@nasa.gov

Yoo-Hsiu Yeh
Ritchie Lee
Carnegie Mellon University
Silicon Valley Campus
Moffett Field, CA 94035

Tony Adami
Ohio University



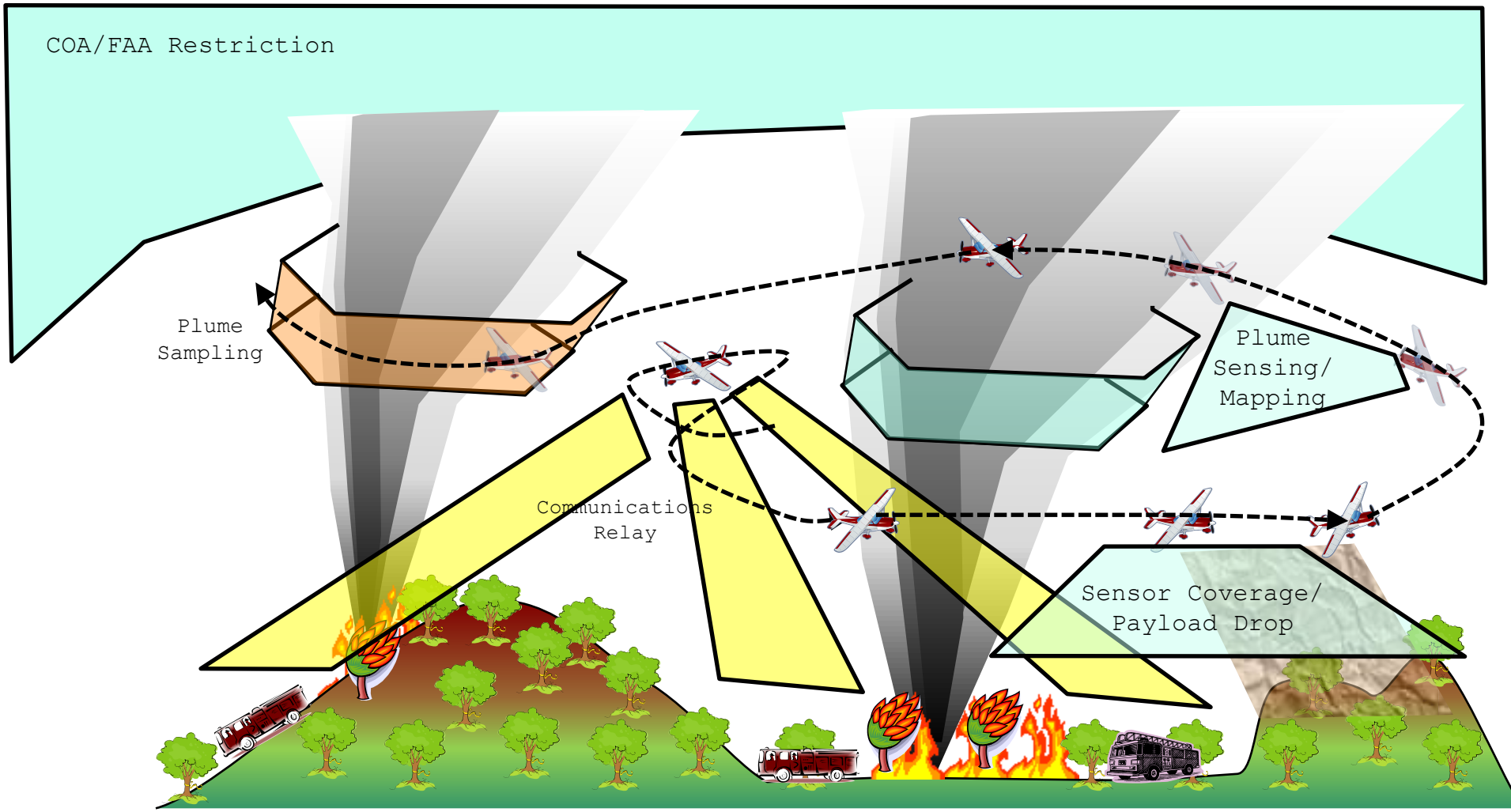
Introduction and SFW Needs

- **Payload Directed Flight is a NASA ARMD Subsonic Fixed Wing (SFW) research task**
- **Investigate closing control loops around payload sensors**
- **Maximize objectives related to the data return from sensor payload suites**
- **Automation towards greater efficiency, performance, and enabling capabilities.**
- **All levels of control and decision making**
  - Mission planning and scheduling
  - Strategic and tactical planning
  - Guidance and control

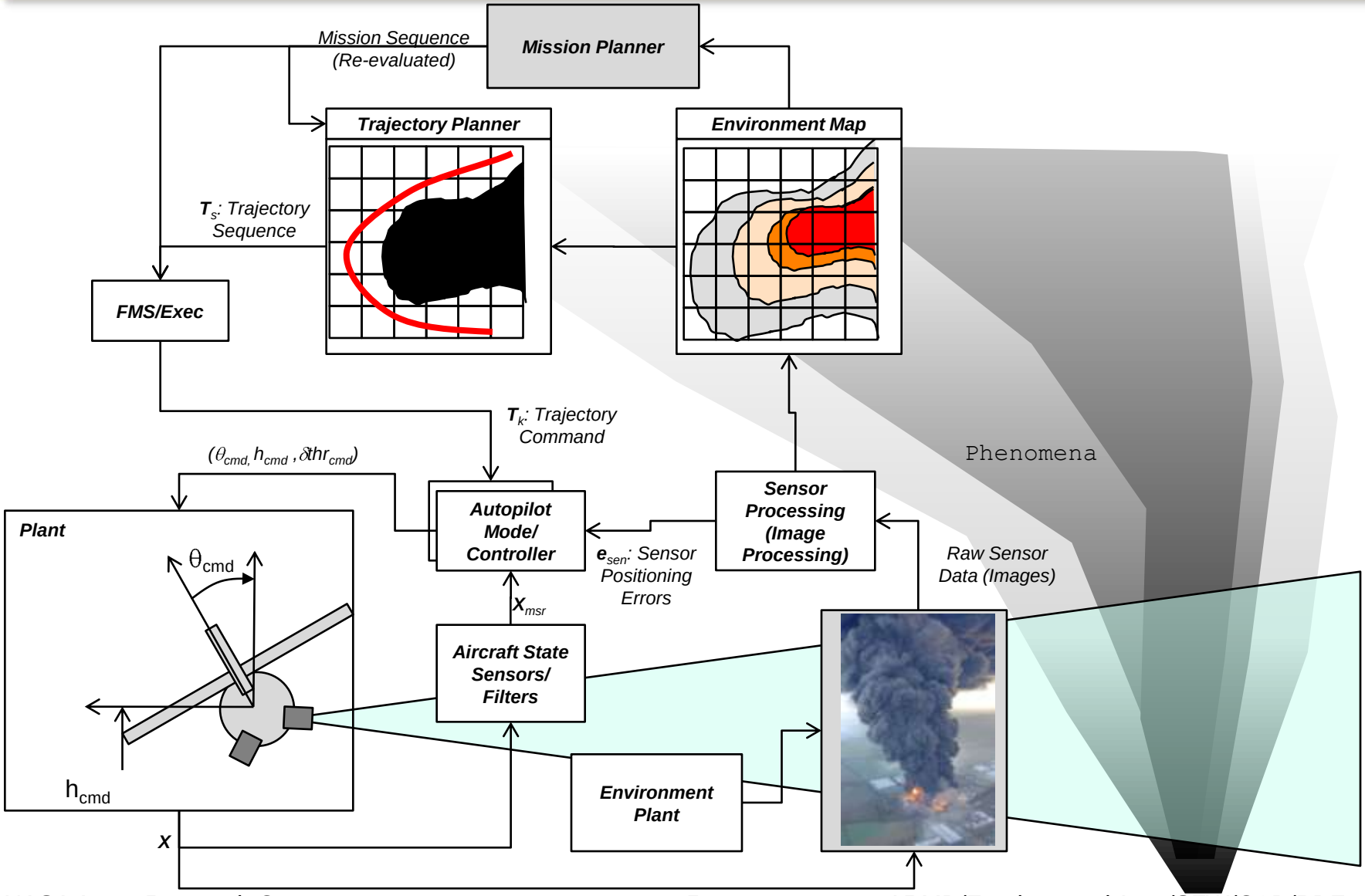
NASA Project Hierarchy

- **Aeronautics Research Mission Directorate (ARMD)**
- **Fundamental Aeronautics Program (FAP)**
- **Subsonic Fixed-Wing (SFW)**
- **Controls and Dynamics (C&D)**
- **Foundational Research**
- **Payload Directed Flight (PDF)**

Representative Mission: Low Altitude Wildfire Monitoring Mission

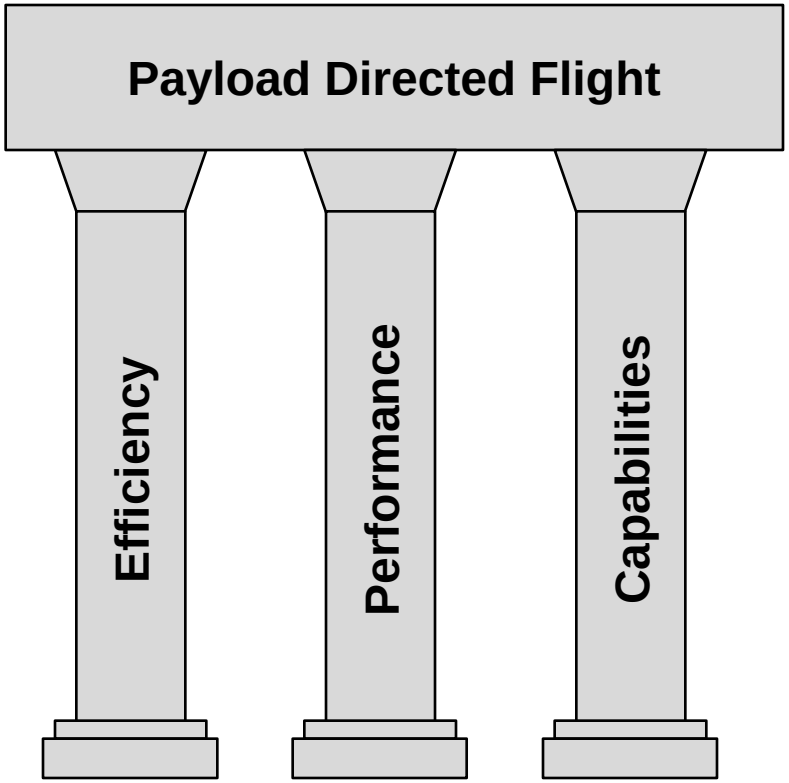


Sensing and Mapping Localized Dynamic Environmental Phenomena



Goals of Payload Directed Flight

- **<u>Increased Efficiency</u> :** By sensing objectives and controlling to these objectives directly, we can guide the vehicle system to increase operational efficiency in the performance of mission tasks (e.g., optimizing vortex formation flying reduction in induced drag.)
- **<u>Increased Performance</u> :** By sensing objectives and controlling to these objectives directly, we can increase performance metrics (e.g., increased data return from payload sensors).
- **<u>Enabling New Capabilities</u> :** Forward looking aspect of research to enable autonomy in the performance of tasks that are not currently possible with state of the art technology.



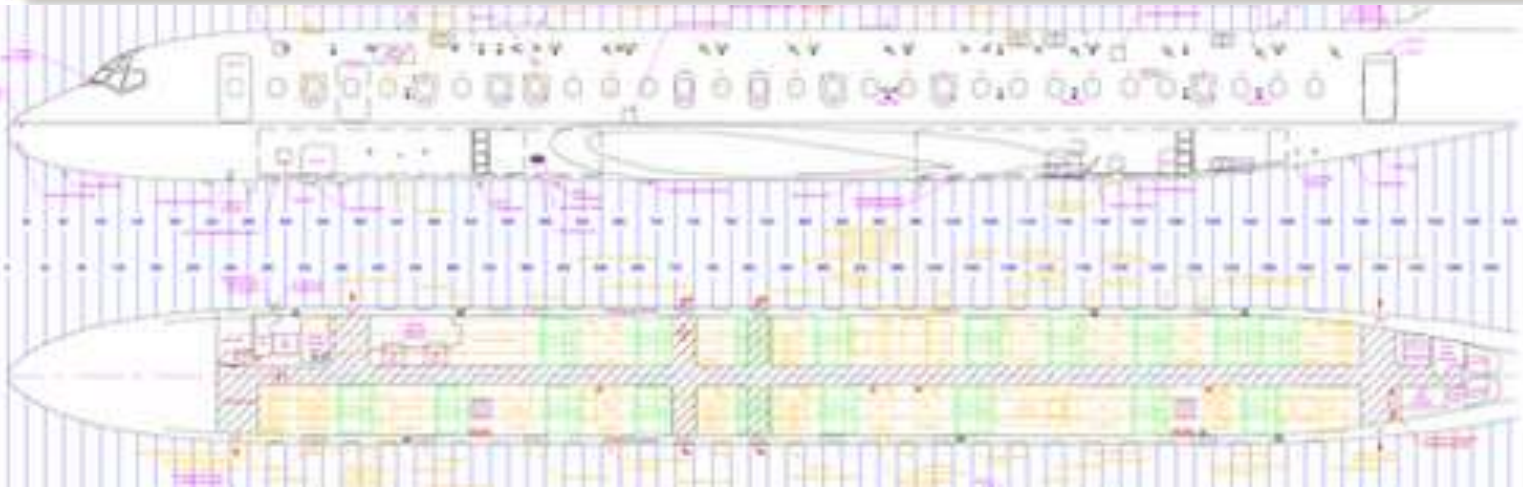
Sub-Sonic Fixed Wing Applications

- **Fuel Efficient Trajectories and Operations, Reduced Emissions**
- **Autonomy for Automated Air Cargo Transport**
- **Formation Flight Tactical and Strategic Planning**
- **Links to Autonomy, Trajectory Management, Planning, and Analysis in Other ARMD Programs (JPDO)**

Needs from Airborne Earth Science

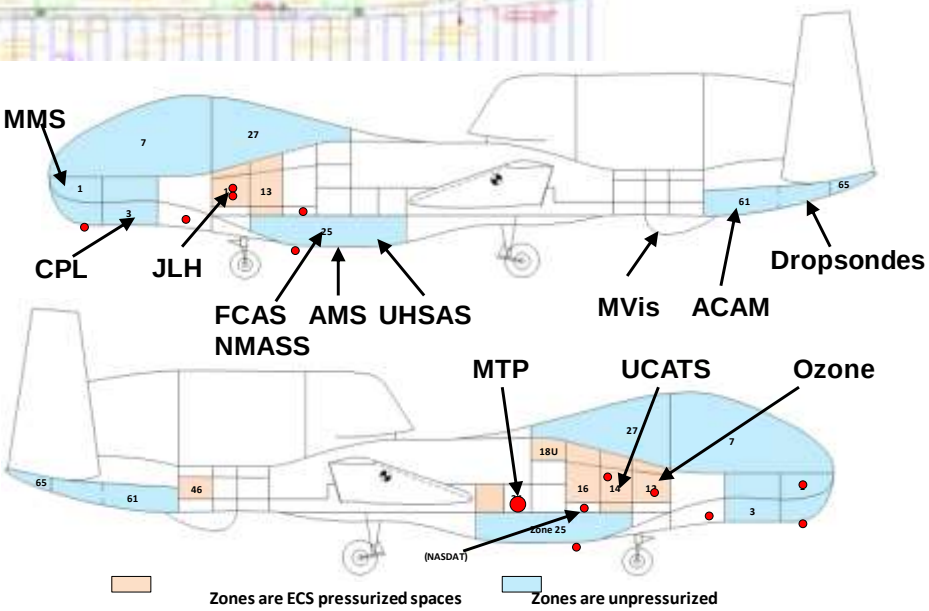
- **How can ARMD's Subsonic Fixed Wing research into Payload Directed Flight apply to the goals of Earth Science?**
- **NASA Airborne Earth Science program is in the forefront of the sensor payload integration into fixed-wing platforms.**
  - "Payload Directed Flight" directly identified as a key requirement in several future-looking SMD/SSP requirement documents.
  - Active missions being conducted allow for real-world integration, testing, and evaluation.
  - Compelling and immediate applications for PDF. Many other possible application areas are still in concept or development phase, and require significant technical challenges to be overcome.
- **Benefits for Earth Science**
  1. Maximize scientific return, and increase scientific output from missions utilizing intelligent flight planning and control.
  2. Increase operational efficiency of airborne missions.
  3. Increase technology integration of tools for prediction, simulation, and control.
  4. Get critical feedback from scientists needed for ARMD/SFW's Payload Directed Flight task.
  5. Next-Gen technology for airborne earth science missions.

Sensors and Payloads



ARCTAS 2008 DC-8 Configuration

(http://www.espo.nasa.gov/arctas/airborne_inst.php)

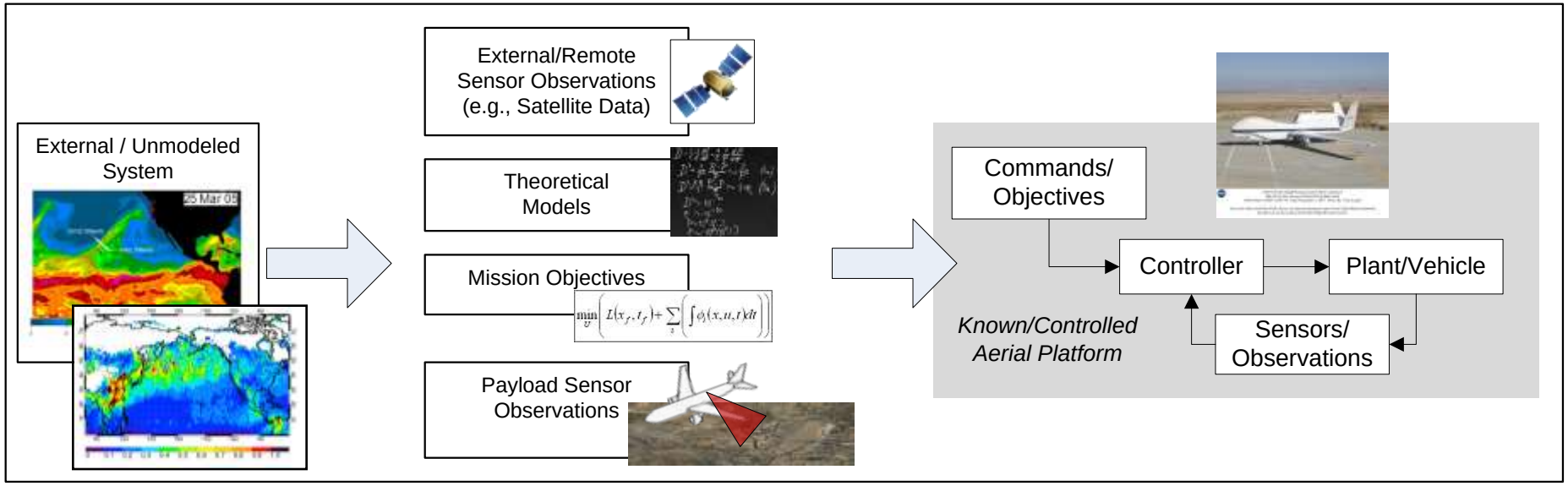


GloPac 2009 Global Hawk Configuration

(Courtesy of Paul Newman (NASA GSFC) and David Fahey (NOAA))

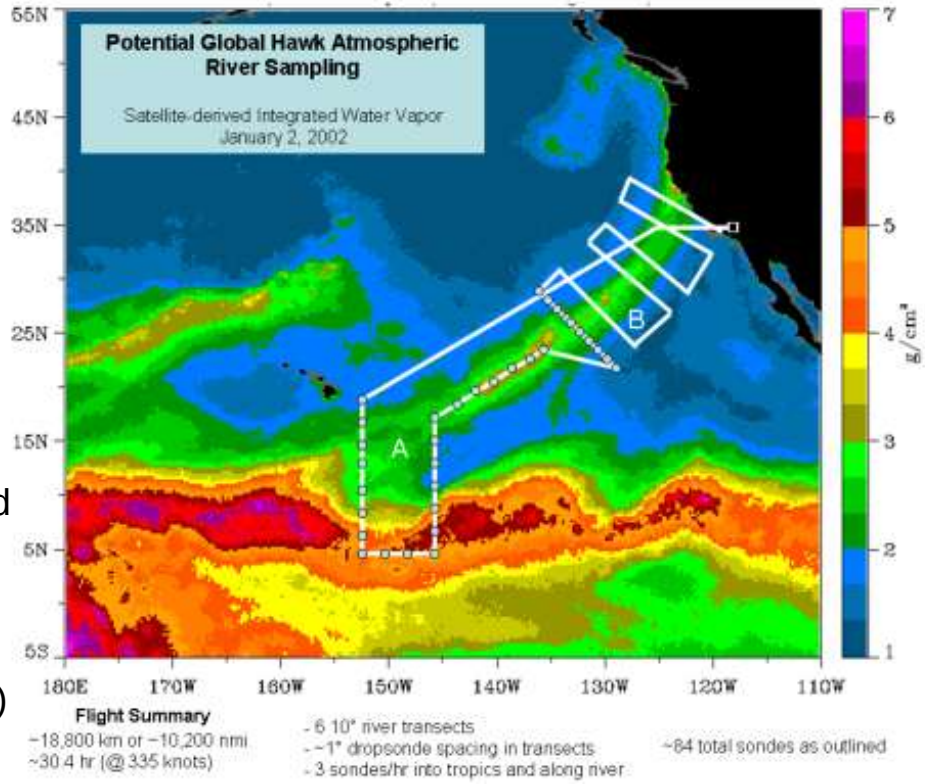
Payload Directed Flight

- **Application Space**
  - We seek performance driven architectures that optimize interaction between the vehicle system as it interacts with and a larger external system, where sensor data feedback gives partial observation of this partially observable system.



Earth Science Representative Missions: Atmospheric Rivers, Aerosol & Pollution Transport

- **Objectives from GloPac 2009 and ARCTAS**
- **Atmospheric River Objectives: Take measurements of streamers of moist air from the central tropical Pacific that move onto the West Coast of the United States**
- **Model and Data Products (from which we build initial trajectories)**
  - Satellite Images (MODIS, GEOS-4, SSMI)
  - Simulation and Numerical Predicative Models
  - Note: Can accommodate lack of input data, build model during flight.
- **Sensors**
  - Deployment of dropsondes (2005, first day deployed 44 dropsondes in two parallel curtains)
  - AMS water vapor, cloud properties sensor, etc.
- **Data Product**
  - Flight plan and dropsonde deployment locations

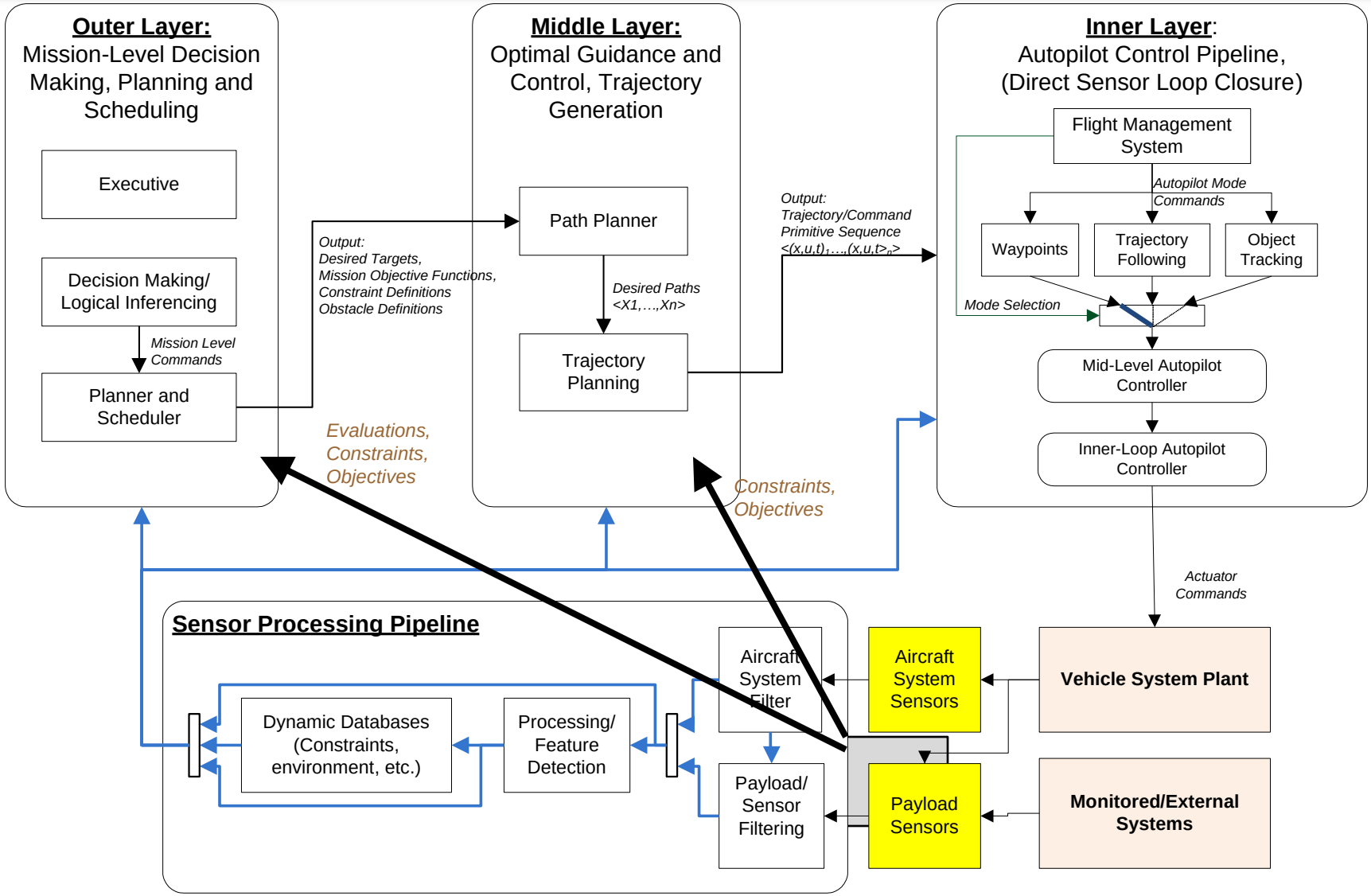


(NOAA SSMI Satellite Image)

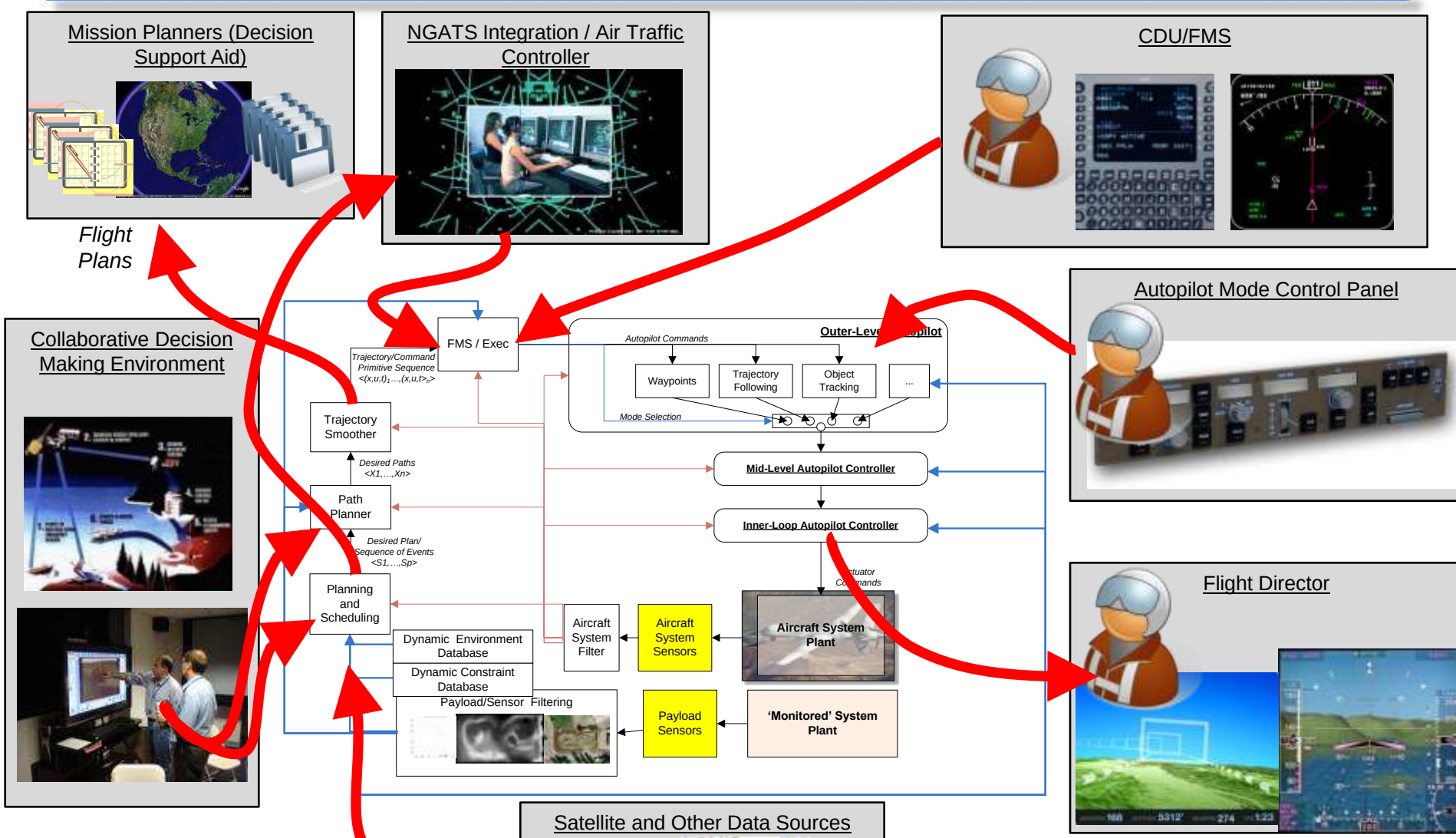
Earth Science Representative Missions

- **Hurricane Sensing**
  - Requires tactical/strategic planning and control that observe constraints, minimize fuel expenditure in highly uncertain/dynamic environment, plan for adverse weather, avoids weather bands, etc.

PDF Architecture

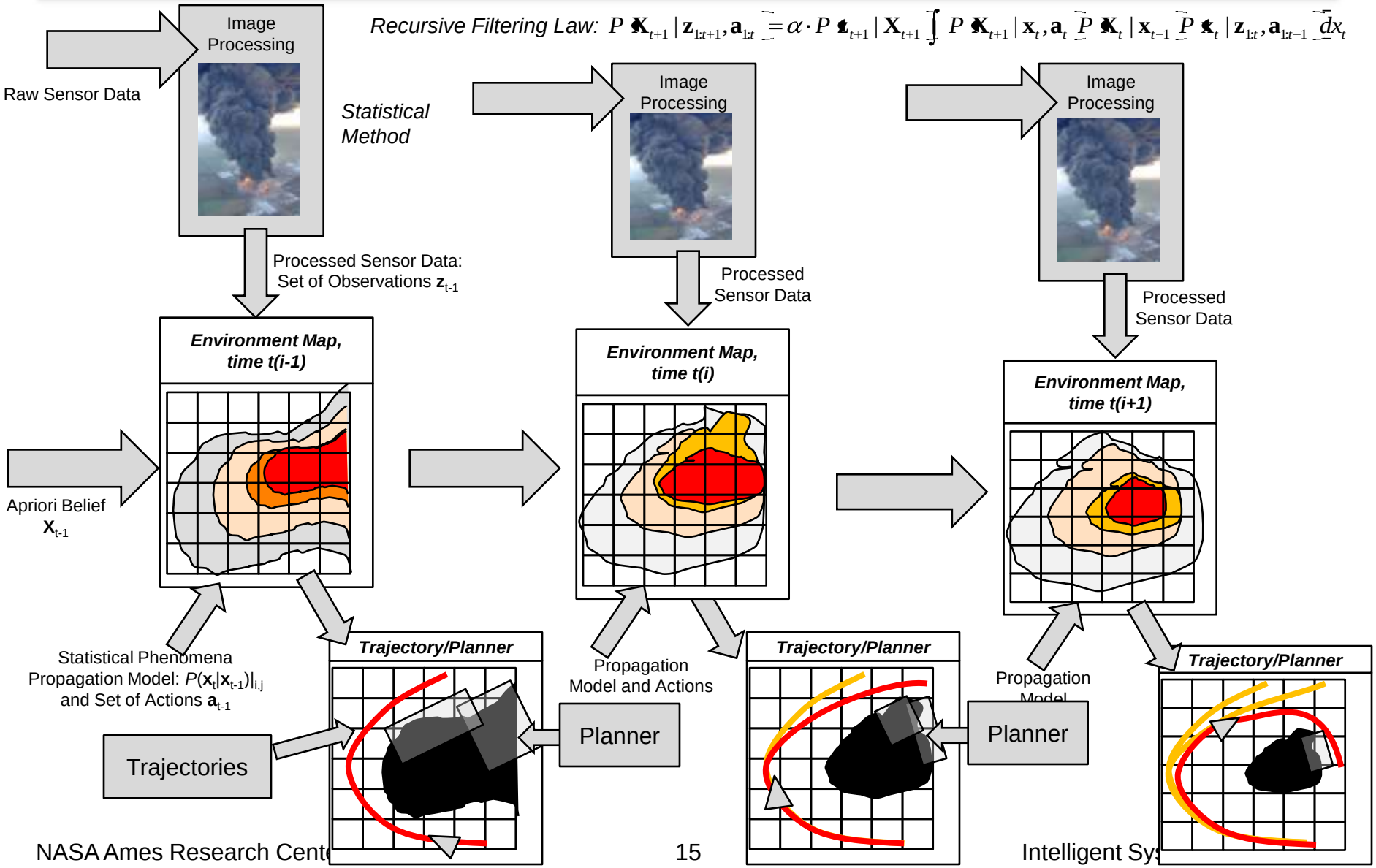


Variable Autonomy and Applications



Sensing and Mapping Phenomena: Payload Feedback through Statistical Environment Map

Recursive Filtering Law: $P(\mathbf{X}_{t+1} \mid \mathbf{z}_{1:t+1}, \mathbf{a}_{1:t}) = \alpha \cdot P(\mathbf{z}_{t+1} \mid \mathbf{X}_{t+1}) \int P(\mathbf{X}_{t+1} \mid \mathbf{x}_t, \mathbf{a}_t)\, P(\mathbf{X}_t \mid \mathbf{x}_{t-1})\, P(\mathbf{x}_t \mid \mathbf{z}_{1:t}, \mathbf{a}_{1:t-1})\, dx_t$



Middle Layer Planning Challenges

- **Tactical / Strategic Planning Requires Optimal Trajectory Calculations Approaches**
  - Optimal Control Problems

$$J = \Phi(\mathbf{x}(t_0), t_0, \mathbf{x}(t_f), t_f) + \int_{t_0}^{t_f} \mathcal{L}(\mathbf{x}(t), \mathbf{u}(t), t)\,\mathrm{d}t$$

  - HJB and Pontryagin Principles
  - Special Solutions for Linear Problems (Linear Quadratic)

$$J = \tfrac{1}{2}\mathbf{x}^{\mathrm{T}}(t_f)\mathbf{S}_f\mathbf{x}(t_f) + \tfrac{1}{2}\int_{t_0}^{t_f} (\mathbf{x}^{\mathrm{T}}(t)\mathbf{Q}(t)\mathbf{x}(t) + \mathbf{u}^{\mathrm{T}}(t)\mathbf{R}(t)\mathbf{u}(t))\,\mathrm{d}t$$

  - Numerical Approaches
    - Nonlinear Programs, Psuedospectral Methods, etc.
- **Limitations**
  - Solutions for limited class of problems
  - Dimensionality
  - Limited class and number of constraints
  - Human-interaction required
  - Maneuver primitives and combinatorial searches
  - Computational load and processing time

NAETGen: Numerical Analysis Engine Trajectory Generation System

- **Online Strategic and Tactical Planning Algorithm**
  - Start with a vehicle model $\dot{x} = f(x,u,t)$

    $y = h(x,u,t)$
  - Augment the model for vehicle or payload constraints, equality or inequality constraints in the set C={C_e,C_i}, of the form

    $C_e(x,\dot{x},u,t) = 0 \qquad C_i(x,\dot{x},u,t) \le 0$
  - Define trajectory generation controllers $G_{tg} = g_i$; *where* $g_i(y_{des},y,u,t)$
  - Define obstacle constraints of the form $C_o(x,t) \le 0$
  - Model sensors and objectives as a set Φ of integral cost functionals.

    $\Phi = \phi_i(x,u,t)$
  - Define final state cost of $L(x_f,t_f)$.
- **Problem**
  - Find control input u*(t) over time period (t_0,t_f) and associated trajectory x(t) leading from a given x_0 to a given x_{goal} that minimizes **J** subject to constraints **C**.

$$u^* = \arg\min_u J_i(u) \qquad J_i = L(x_f,t_f) + \sum_s \left(\int_{t_0}^{t_f} \phi_i(x,u,t)\,dt \right)$$

NAETGen: Numerical Analysis Engine Trajectory Generation System

- **Algorithmic Trajectory Generation**
  - High fidelity model used for synthesis of trajectory branches
  - Constant-time branch generation through online closed-loop control system
  - Handles High Fidelity Wind/Gust Models and Complex Dynamic Environments
  - Produces High Fidelity Predictive Trajectories Flight Plans
  - Large Number of Constraints, Multiple Objectives
- **Tree nodes defined as N=(x,t,j), where x=state, t=time, and j=cost.**
- **Inputs:**

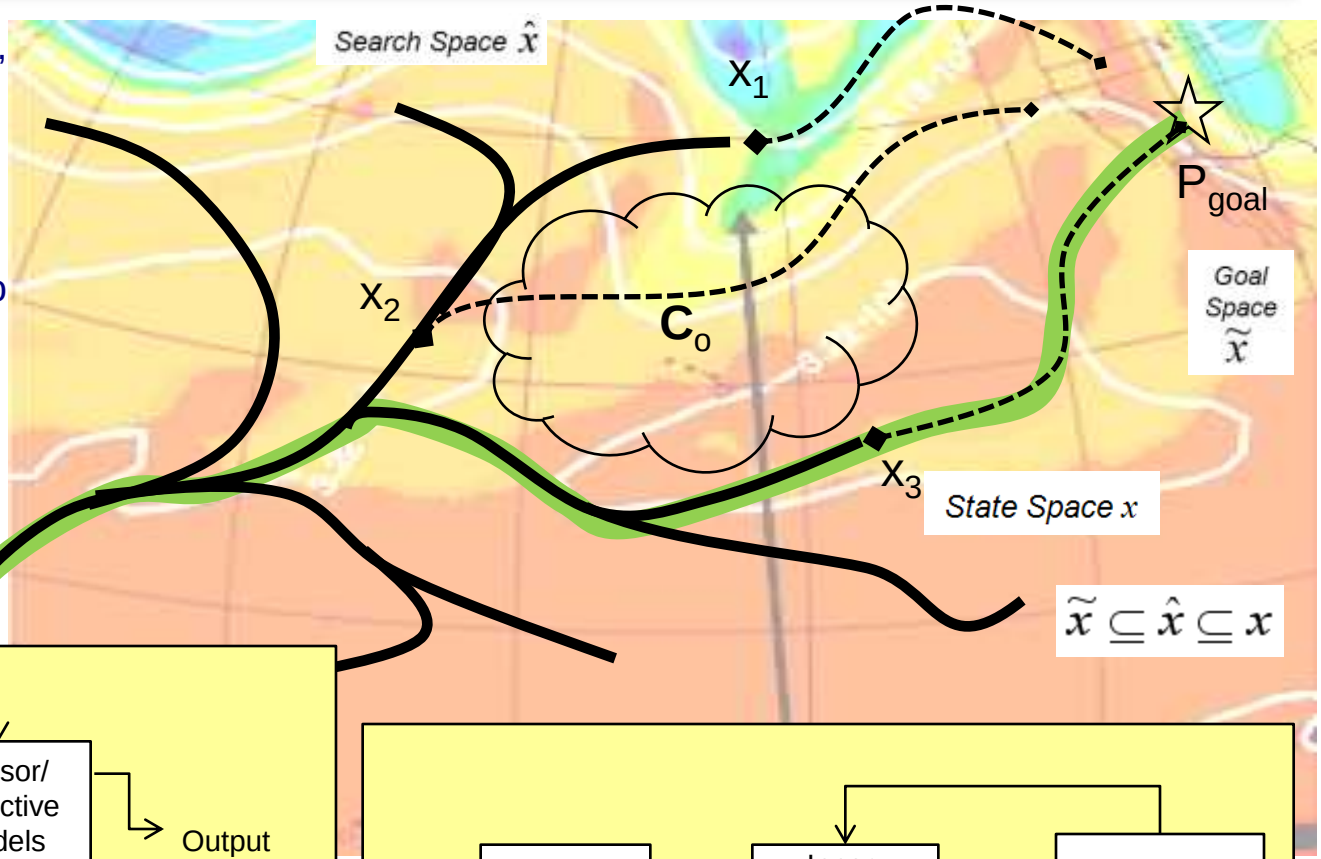
$\boldsymbol{x}_0$	Start state
t_0	Start time
$\boldsymbol{x}_{goal}$	Goal state
$\boldsymbol{C}_T$	Constraint set
T	Search Tree

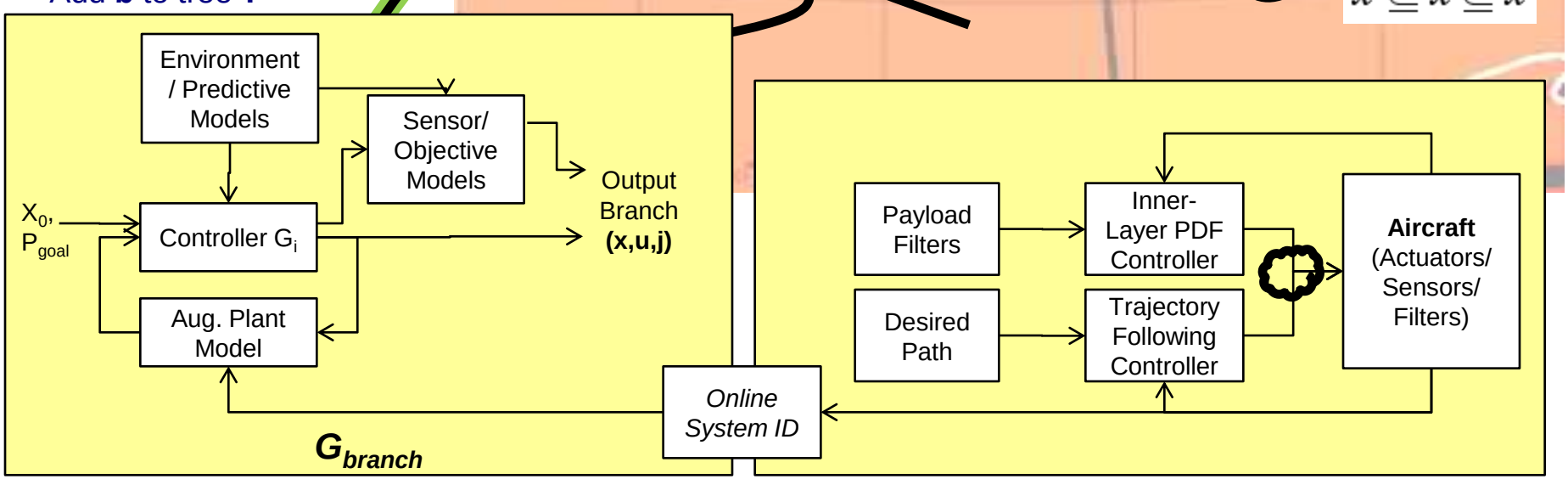
- **Algorithm:**
  - Initialize**T** = $\{(x_0,t_0,0)\}$
  - GenerateBranch (**T**, $\mathbf{x}_{goal}$, $\mathbf{C}_T$)
  - While Time Remains
    - $\mathbf{x}_{rnd}$ = *GenerateRandomSearchSpaceVertex*()
    - **b** = GenerateBranch (T, $\mathbf{x}_{rnd}$, $\mathbf{C}_T$)
    - GenerateBranch (**b**, $\mathbf{x}_{goal}$, $\mathbf{C}_T$)
  - Select branch from x_0 to the lowest cost point near goal node.

NAETGen Branch Generation Routine

GenerateBranch(**P**goal, Tree **T**, Constraints $\mathbf{C}_o$)

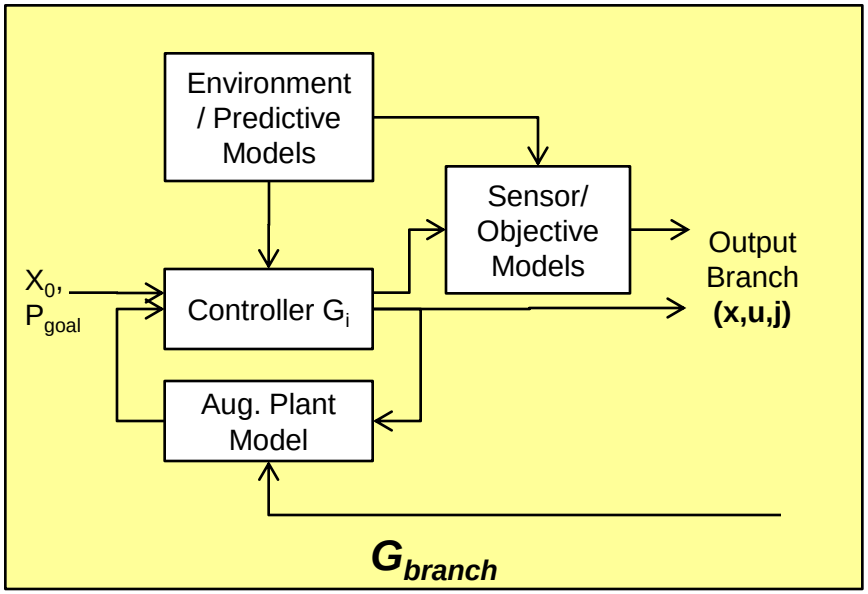
- Generate candidate set **x** of closest tree nodes through distance/cost heuristics
- For each $\mathbf{x}_i \in \mathbf{x}$, Use $\mathbf{G}_{branch}$ to generate trajectory **b**=(**x**,**u**,**j**) based on augmented plant (constant time)
- Check against obstacle constraint set $\mathbf{C}_o$, trim trajectory as needed
- Add **b** to tree **T**





NAETGen Closed-Loop Trajectory Construction Model (CLTCM)

- **Branch selection based on vehicle's dynamic model wrapped by branch generation autopilot controller**
- **CLTCM allows for highest fidelity model of vehicle**
  - High-fidelity 6-DOF solutions
  - Augmented vehicle model incorporates limits to vehicle model, such as actuator limits, damaged system dynamics, etc.
  - Constraints such as rate limits can be included in vehicle model or in the controller
- **Branch generation controller designed to take random vector in the search space**
- **Environment model allows for highly complex environmental and atmospheric phenomena**
- **Environment database**
- **Integration of sensors and objective cost derivatives.**
- **Outputs**
  - High-fidelity states
  - Control inputs
  - Cost objective function

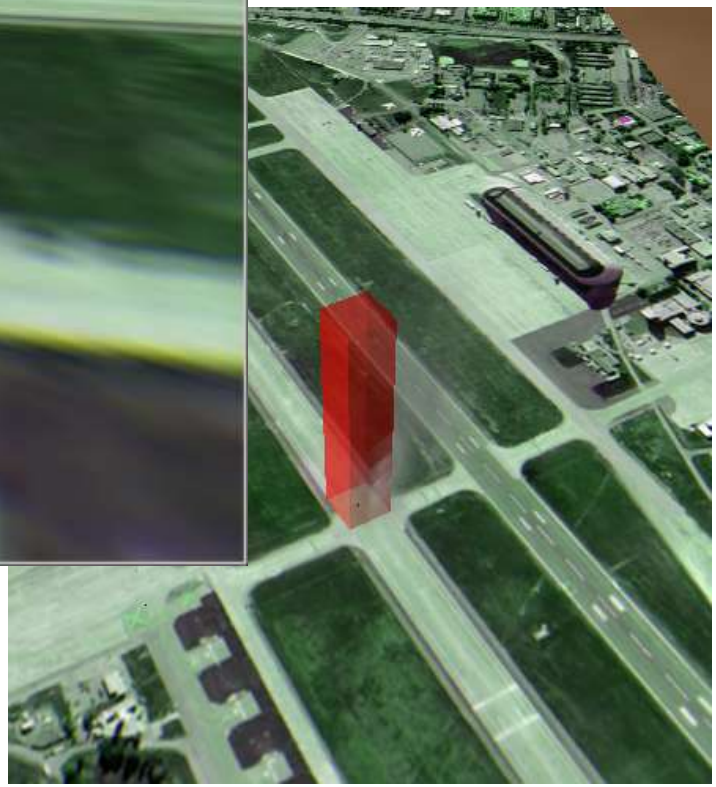


- **Current Research**
  - **Multiple controller modules, G_i, for different types of control modes. Selection based on heuristic approaches.**
  - **Multiple cost objectives**
  - **Confidence metrics and adaptive mission cost functions**
  - **TLC controller for trajectory control**

Integrated Simulation Testing
EAV Simulation System/Reflection Architecture

- **Requirement:**
  - Perform mission fully autonomously (include takeoff, but not mission termination)
  - Operate continuously for 30 minutes, 5 seconds to plan, planning 30 seconds ahead of the current time.
  - Maintain plume target location in view at all times (disregarding initial takeoff phase)
  - Do not violate airspace constraints, do not collide with the plume.
- **Testing performed at the Exploration Aerial Vehicles Laboratory, at NASA Ames Research Center**
- **Tests performed using hardware-in-the-loop simulation testing facility, based on Reflection Architecture**
- **Non-linear 6-DOF model of EAV platform used.**
- **Matlab Simulink model was converted into C++ code using the Real-Time Workshop Embedded Coder**
- **Created a GPU-based smoke plume particle rendering system.**
- **Tree search/generation algorithm implemented in C++**
- **No optimization performed on code.**
- **Tested on Intel Core 2 X9650, 3 GHZ, 3GB RAM**
- **Search time limited to 5 seconds**

Integrated Simulation Testing
EAV Simulation System/Reflection Architecture

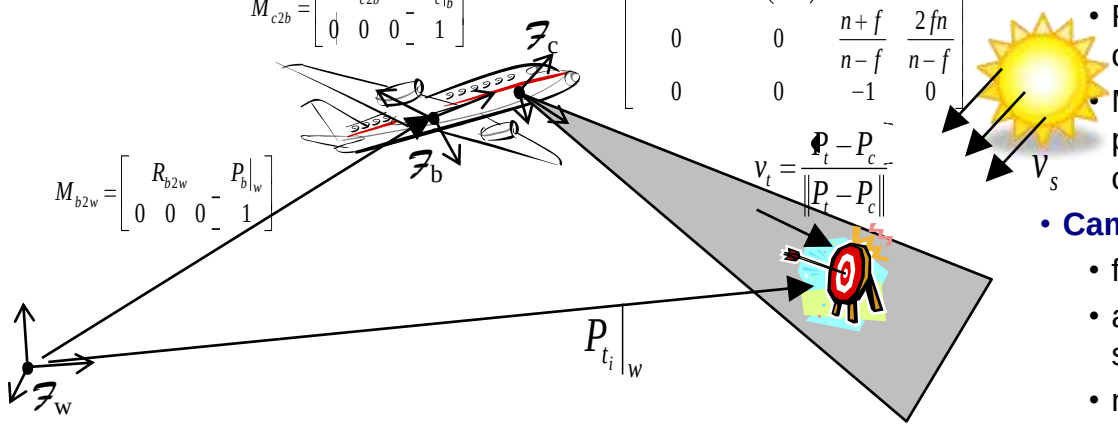


Sensor Modeling Example: Simple Body Fixed Camera System

$$M_{c2s} = \begin{bmatrix} \frac{\cot\left(\frac{fov}{2}\right)}{AR} & 0 & 0 & 0 \\ 0 & \cot\left(\frac{fov}{2}\right) & 0 & 0 \\ 0 & 0 & \frac{n+f}{n-f} & \frac{2fn}{n-f} \\ 0 & 0 & -1 & 0 \end{bmatrix}$$

$$M_{c2b} = \begin{bmatrix} R_{c2b} & P_c|_b \\ 0\ 0\ 0 & 1 \end{bmatrix}$$

$$M_{b2w} = \begin{bmatrix} R_{b2w} & P_b|_w \\ 0\ 0\ 0 & 1 \end{bmatrix}$$



- **Target t is arbitrarily shaped geometry, P_{ti} represents position of ith geometry vertex**
- **Frames and Transforms**
  - F_w, F_b, F_c are the world, aircraft body, and camera frames, respectively
  - M_{b2w}, M_{c2b}, M_{c2s} are the F_b to F_w, F_c to F_b, and perspective projection from F_c to F_s (screen coordinates), respectively
- **Camera Optical Properties**
  - fov : vertical field of view angle
  - ar : aspect ratio of camera (horizontal size/vertical size)
  - n,f: Near and far clip planes (for efficiency)
  - Let Pix(P) return 1 if the pixel is in view, otherwise 0
- **Desired Sun-Camera Angle**
  - v_t and v_s are the unit vector from camera to target, and sun vector respectively
  - Minimize magnitude of dot product for best sun angle (side-lit object)
- **Cost Functional**

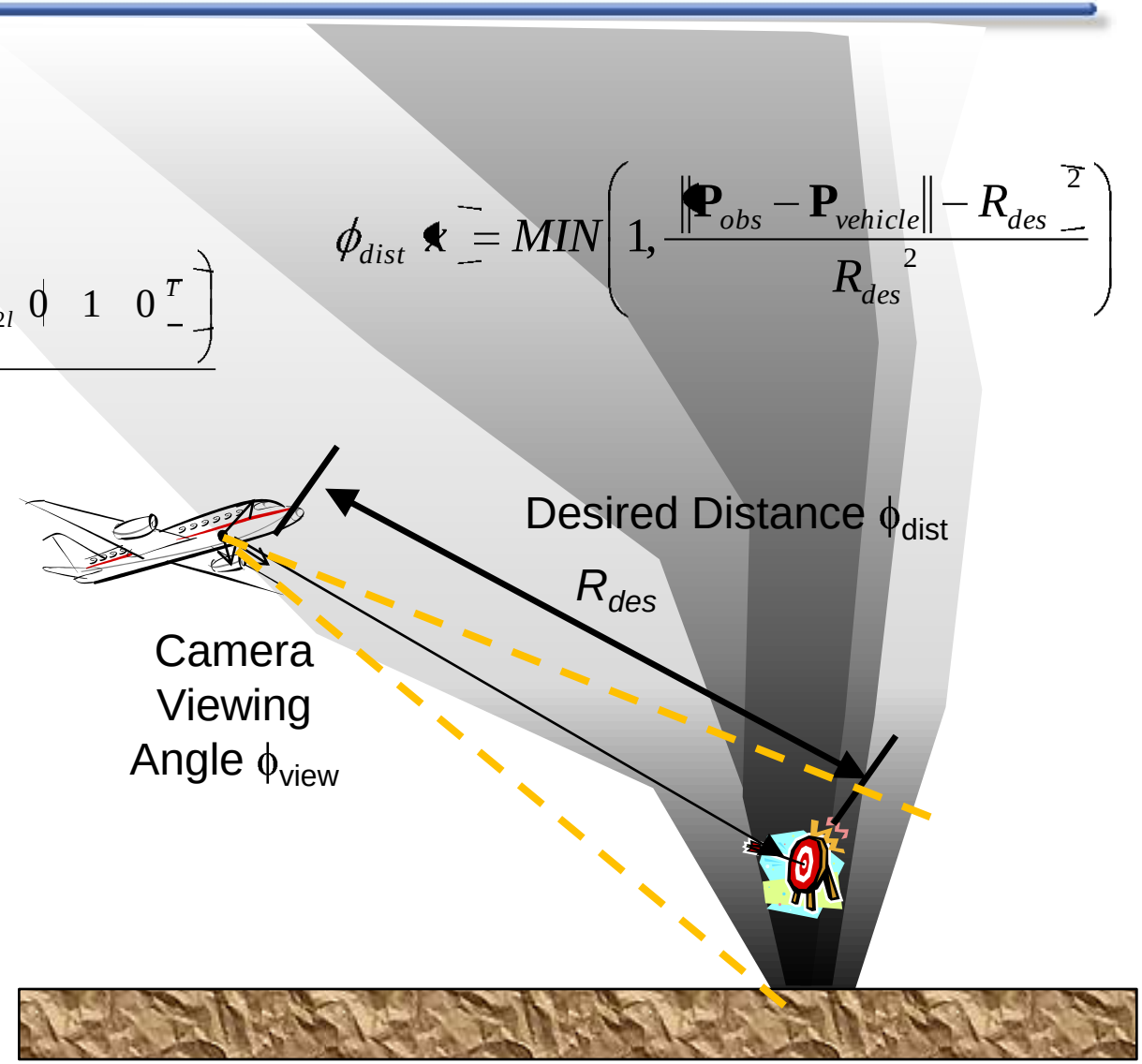
<u>Example Camera Sensor</u>
Point Grey DragonFly w/ BW38B1000
Interface: IEEE-1394 Firewire
Sensor: Sony 1/3" progressive scan CCD
Resolution: 640x480 or 1024x768
Frame Rates: 1024x768 at 15FPS
Signal to Noise: 60dB
Wieght/Size: 2”x2.5”
Power: <2W
Focal Length: 3.94mm
Angle of View: 74° H, 54° V, 100° D (AR=1.37)

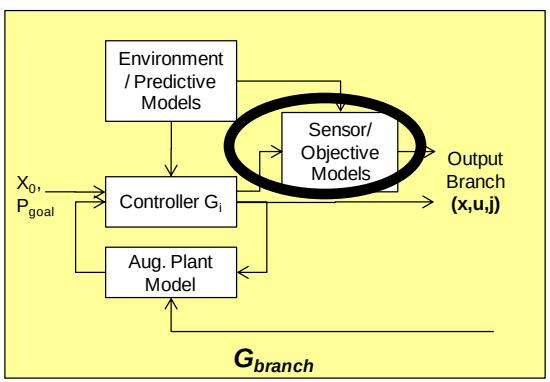
$$\phi_{cam}(x,u,t) = \underbrace{\frac{K_1}{n_{pix}} \sum_i Pix\left(M_{c2s} M_{c2b}^{-1} M_{b2w}^{-1} P_{t_i}|_w\right)}_{View} + \underbrace{K_2 |v_t \cdot v_s|}_{Sun\ Angle} + \underbrace{K_3 \left(\|P_t - P_c\| - d\right)}_{Distance}$$

Simplified Body Fixed Sensor Model

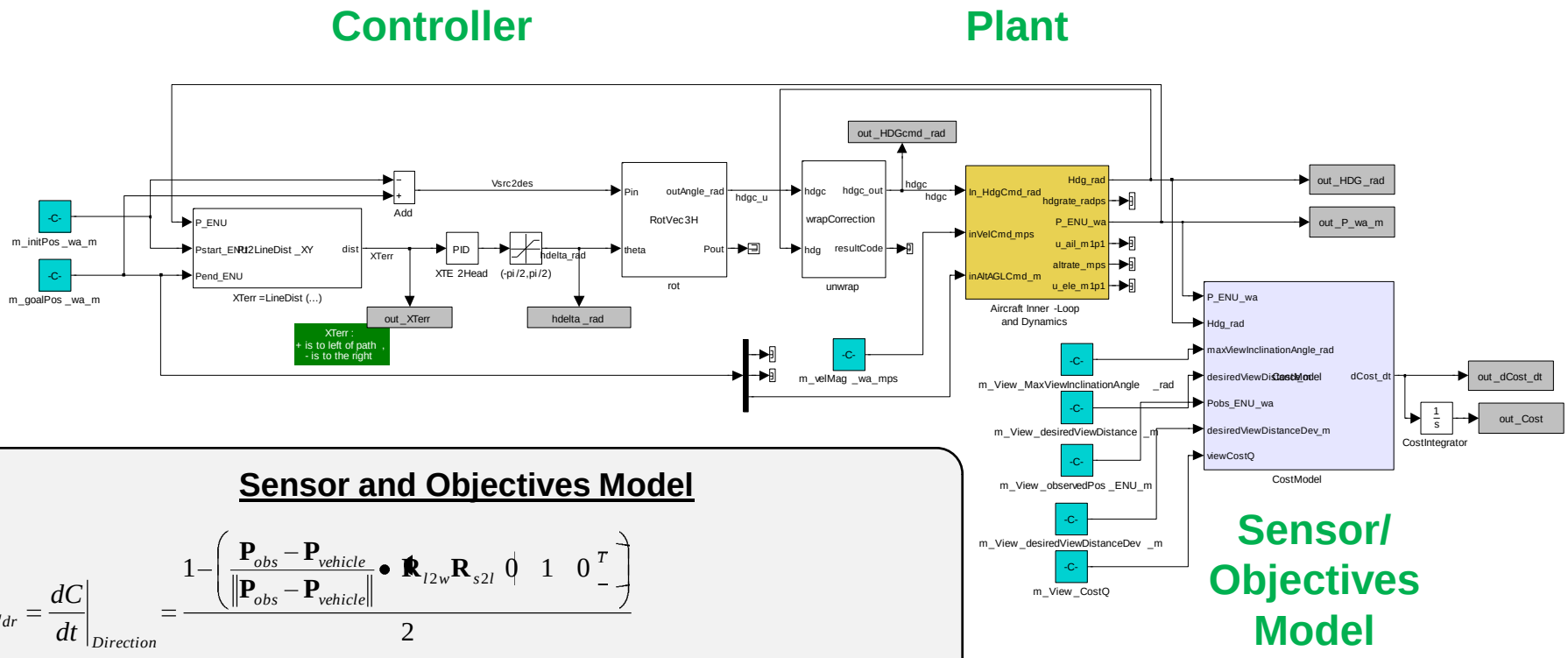
$$\phi_{view}(\mathbf{x}) = \frac{1 - \left(\frac{\mathbf{P}_{obs} - \mathbf{P}_{vehicle}}{\left\| \mathbf{P}_{obs} - \mathbf{P}_{vehicle} \right\|} \bullet \mathbf{R}_{l2w} \mathbf{R}_{s2l} \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}^T \right)}{2}$$

$$\phi_{dist}(\mathbf{x}) = MIN\left(1, \frac{\left(\left\| \mathbf{P}_{obs} - \mathbf{P}_{vehicle} \right\| - R_{des} \right)^2}{R_{des}^2}\right)$$





Simulation Testing: Simulink Trajectory Generation Model



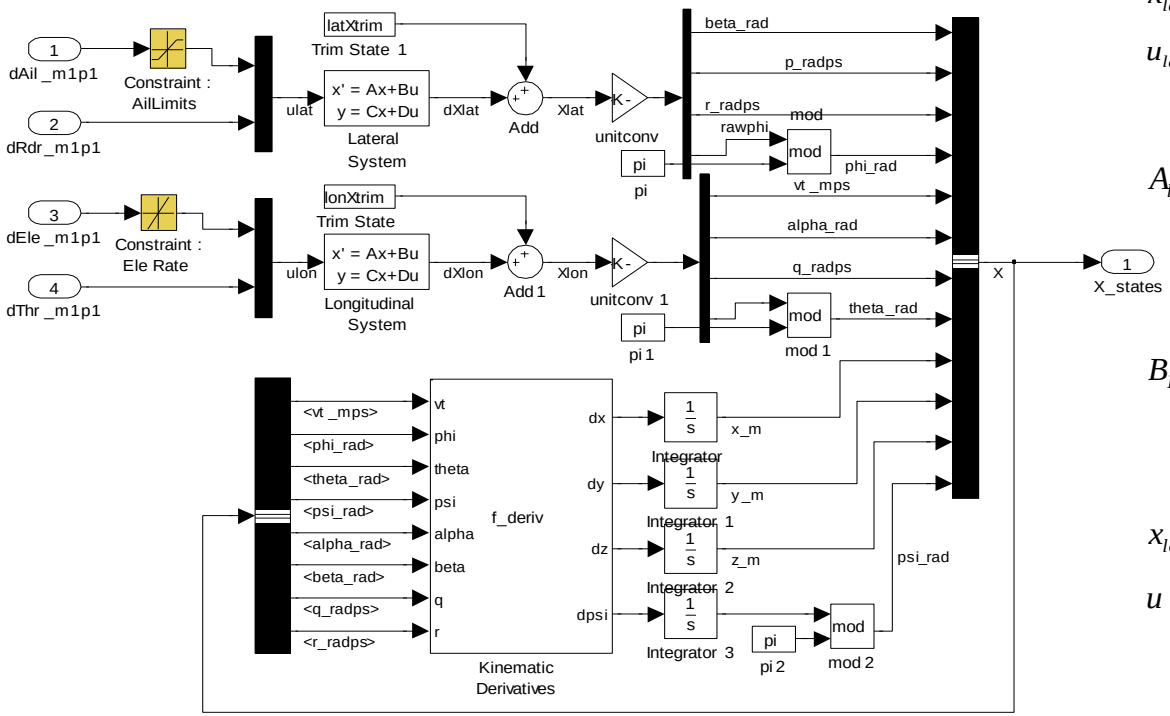
Sensor and Objectives Model

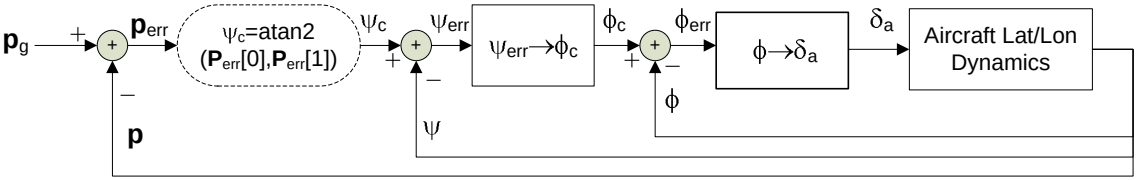
$$L_{dr} = \left.\frac{dC}{dt}\right|_{Direction} = \frac{1-\left(\frac{\mathbf{P}_{obs}-\mathbf{P}_{vehicle}}{\left\|\mathbf{P}_{obs}-\mathbf{P}_{vehicle}\right\|}\bullet\mathbf{R}_{l2w}\mathbf{R}_{s2l}\begin{bmatrix}0 & 1 & 0\end{bmatrix}^T\right)}{2}$$

$$\phi_{cam}(x,u,t) = \frac{K_1}{n_{pix}}\sum_i Pix\left(M_{c2s}M_{c2b}^{-1}M_{b2w}^{-1}\left(\mathbf{P}_{t_i}\right|_w\right) + K_2\left|v_t\cdot v_s\right| + K_3\left(\left\|\mathbf{P}_t-P_c\right\|-d\right)$$

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

Simulation Testing: Augmented Vehicle Plant Model and Waypoint Controller





DIRECT-TO Autopilot System

$$x_{lat} = \begin{bmatrix} \Delta\beta & \Delta p & \Delta r & \Delta\phi \end{bmatrix}^T$$

$$u_{lat} = \begin{bmatrix} \Delta\delta_{ail} & \Delta\delta_{rdr} \end{bmatrix}^T$$

$$A_{lat} = \begin{bmatrix} -1.0176 & 0.0728 & -0.8887 & 0.2918 \\ -16.7704 & -5.2969 & 1.2992 & 0 \\ 13.7306 & -0.3138 & -1.3475 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

$$B_{LAT} = \begin{bmatrix} 0.3404 & 0.5297 \\ 55.2736 & 4.6948 \\ -1.4038 & -9.7679 \\ 0 & 0 \end{bmatrix}$$

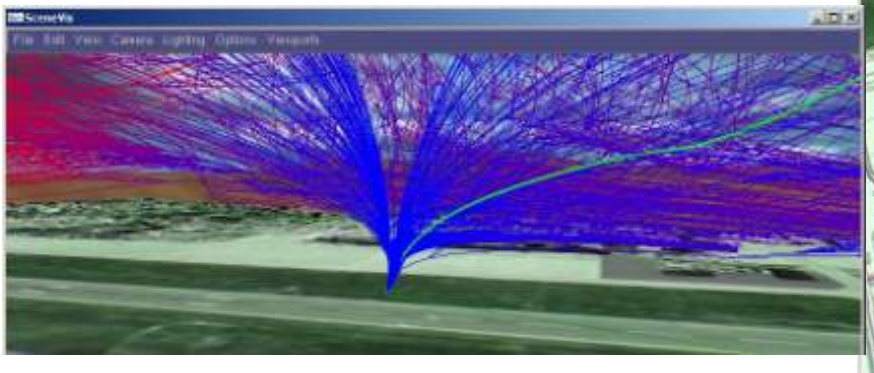
$$x_{lon} = \begin{bmatrix} \Delta v_T & \Delta\alpha & \Delta q & \Delta\theta \end{bmatrix}^T$$

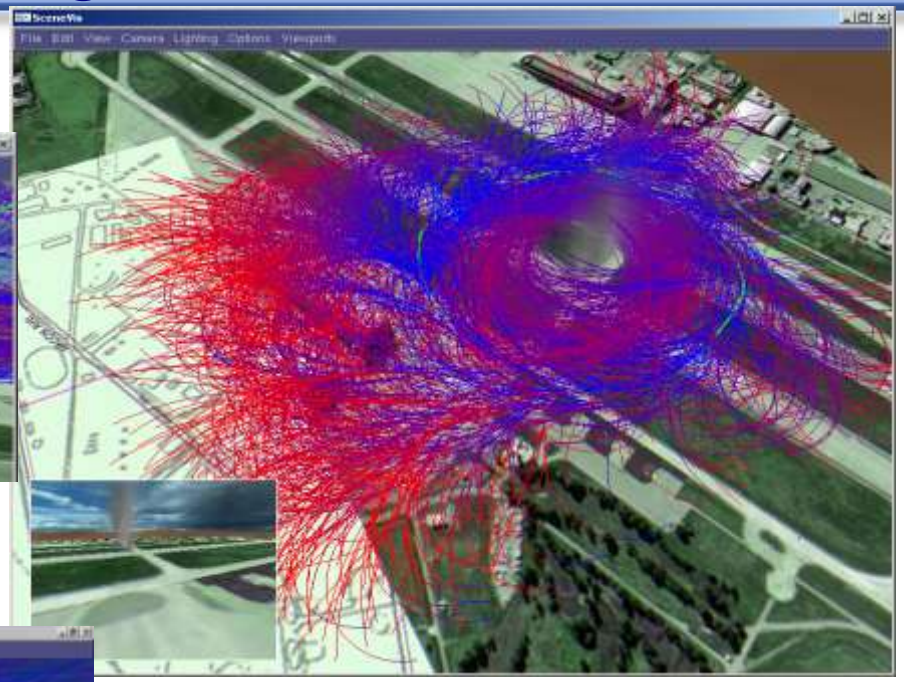
$$u = \begin{bmatrix} u_{lat} & u_{lon} \end{bmatrix}^T$$

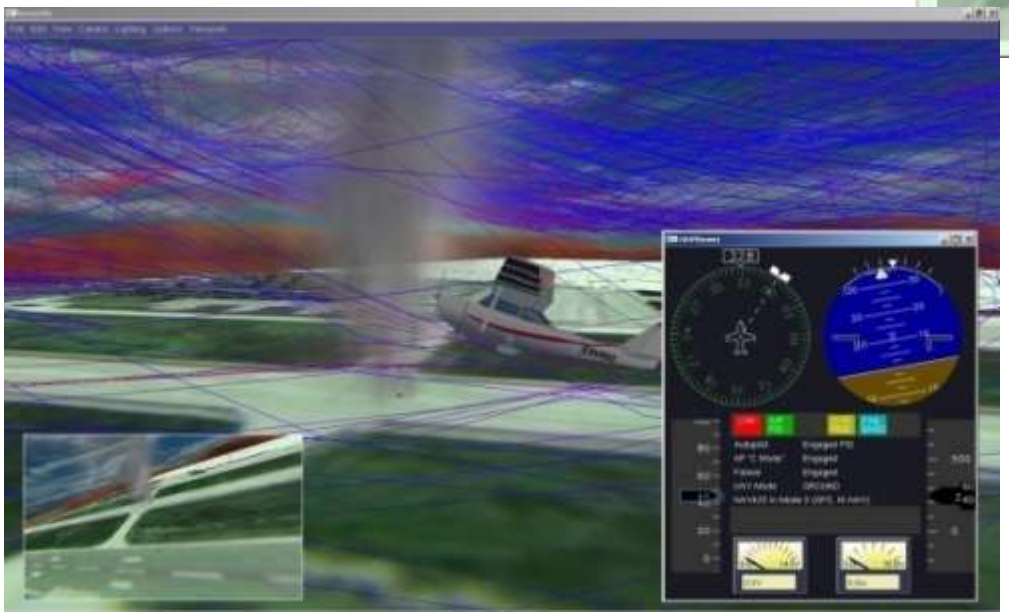
$$A_{lon} = \begin{bmatrix} -1.409 & -0.040515 & 0 & -0.096803 \\ -0.0281 & -4.0540 & 1.2896 & 0 \\ -0.0566 & -20.9832 & -1.6784 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

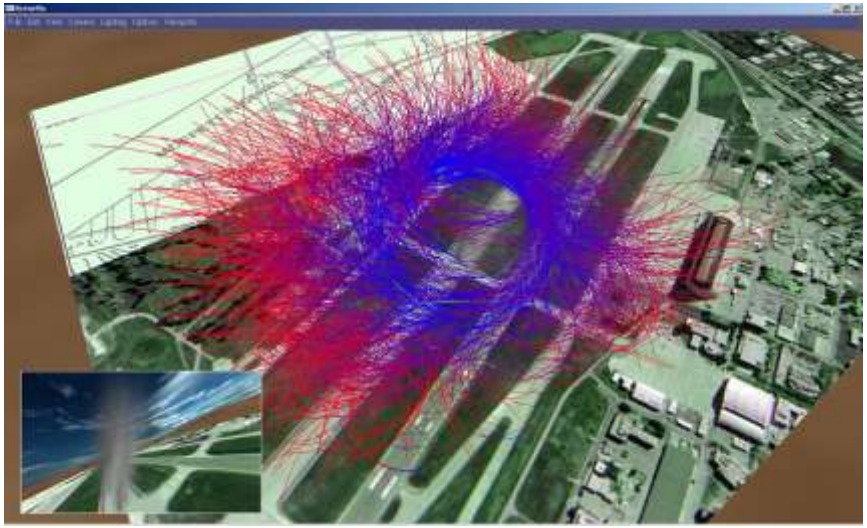
$$B_{lon} = \begin{bmatrix} -0.0699 & 3.24 \\ -0.8405 & -0.03057 \\ -25.99 & -0.07424 \\ 0 & 0 \end{bmatrix}$$

Simulation Testing: EAV Plume Monitoring Mission

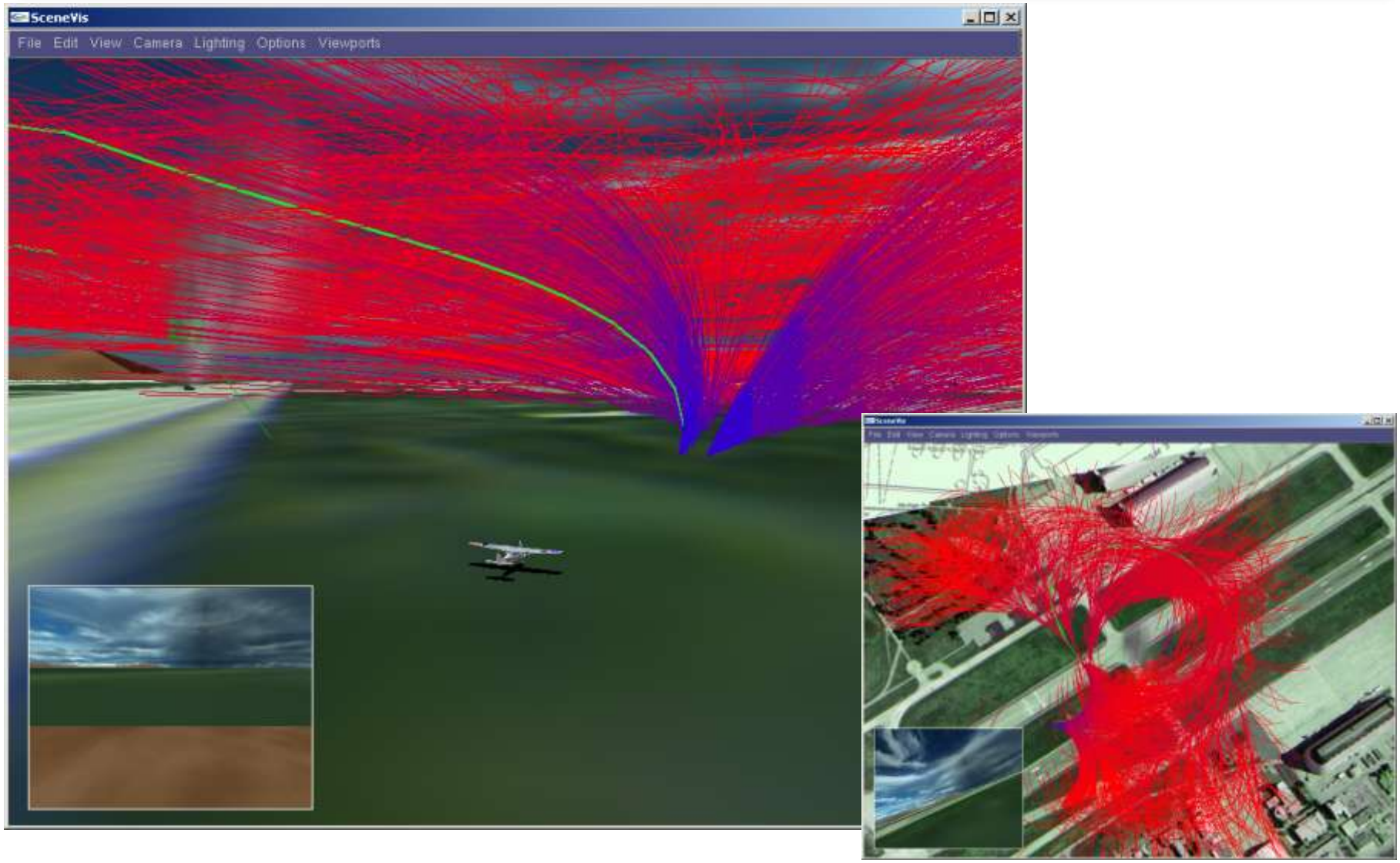




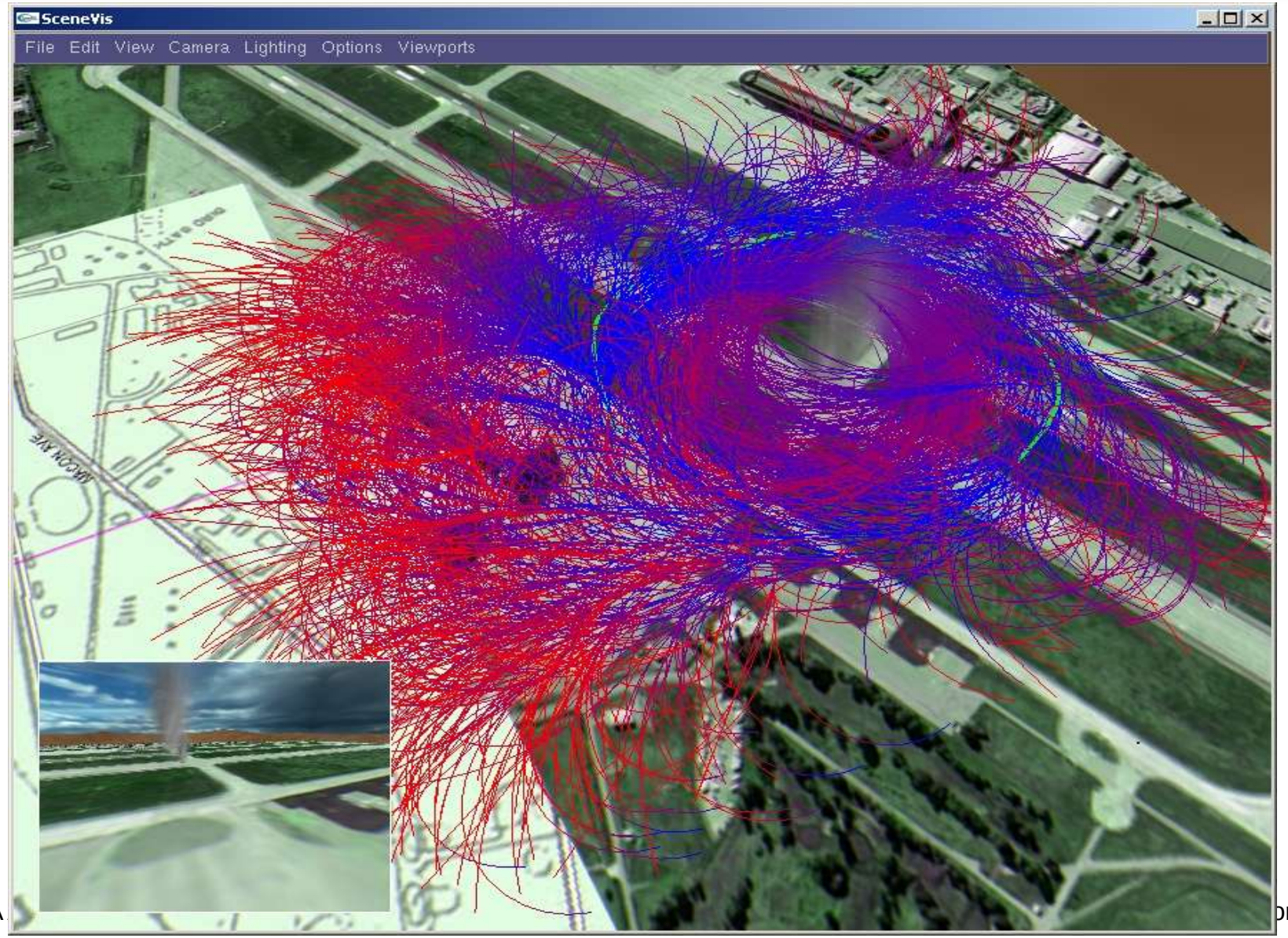




Integrated Simulation Testing
EAV Simulation System/Reflection Architecture



Integrated Simulation Testing
EAV Simulation System/Reflection Architecture

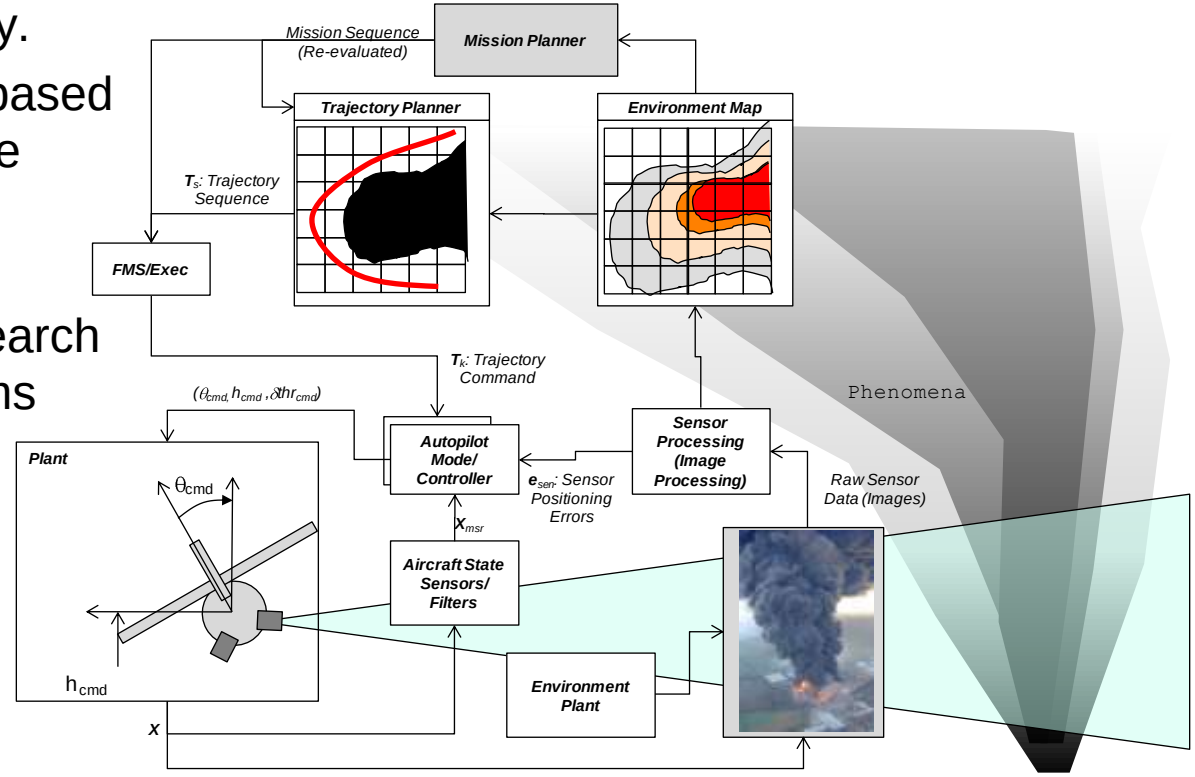


Simulation Results

- **Results**
  - Entire simulation - rendering, controller, simulation, etc – ran on a single machine (Intel Core 2 X9650, 3 GHZ, 3GB RAM).
  - Trajectory generation thread ran by itself on second processor.
  - In five second search window, search algorithm averaged 48,202 nodes added to the tree, with 3435 branches, taking 15,002,030,502 CPU clock cycles. No optimizations performed (algorithm is parallelizable for faster real-time operation).
  - All requirements satisfied, view of plume was never lost, constraints never violated.

Current Development

- **Future Plans**
  - Higher fidelity applications for NAETGen
  - Integration into NASA Airborne Science Missions as Decision Support Tool
  - Working to fill in the pieces for fully autonomous plume tracking problem.
  - Developing Linearized Trajectory Controller (J. Zhu, T. Adama, Ohio U) for better tracking of trajectory.
  - Developing online vision-based estimation filters for smoke plume (R. Lee, CMU)
  - Flight testing on X-SCAV UAV at NASA Ames Research Center (and other platforms including UGV)

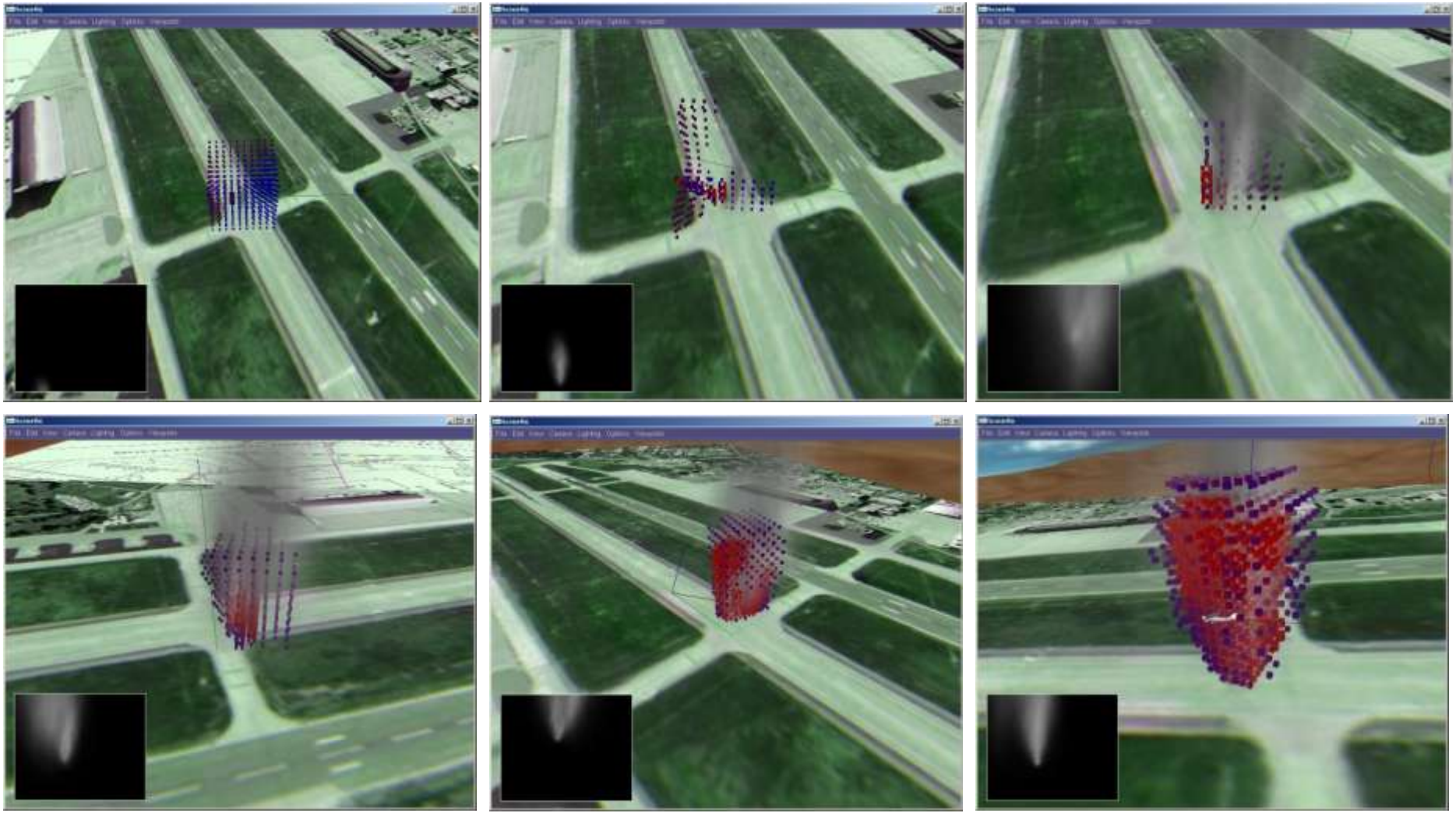


Knowledge Representation through Recursive Estimation

$$Q_{k+1} = Q_k - Q_k A_{k+1}^T (A_{k+1} Q_k A_{k+1}^T + \Sigma_{k+1})^{-1} A_{k+1} Q_k$$

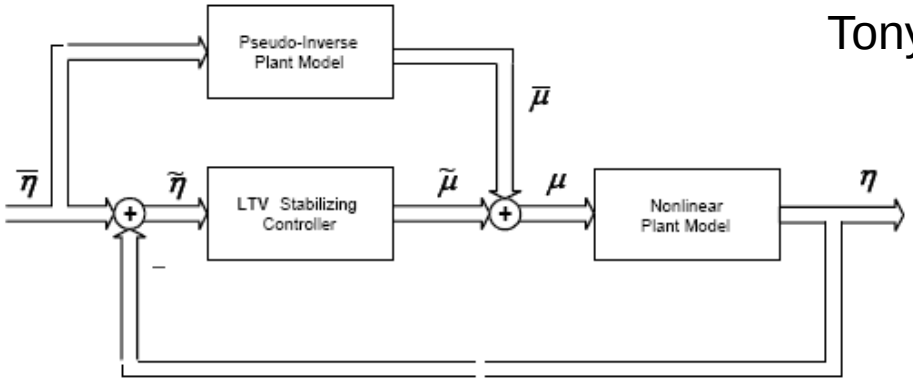
$$\hat{x}_{k+1} = \hat{x}_k + Q_k A_{k+1}^T \left(A_{k+1} Q_k A_{k+1}^T + \Sigma_{k+1}\right)^{-1} (y_{k+1} - A_{k+1} \hat{x}_k)$$

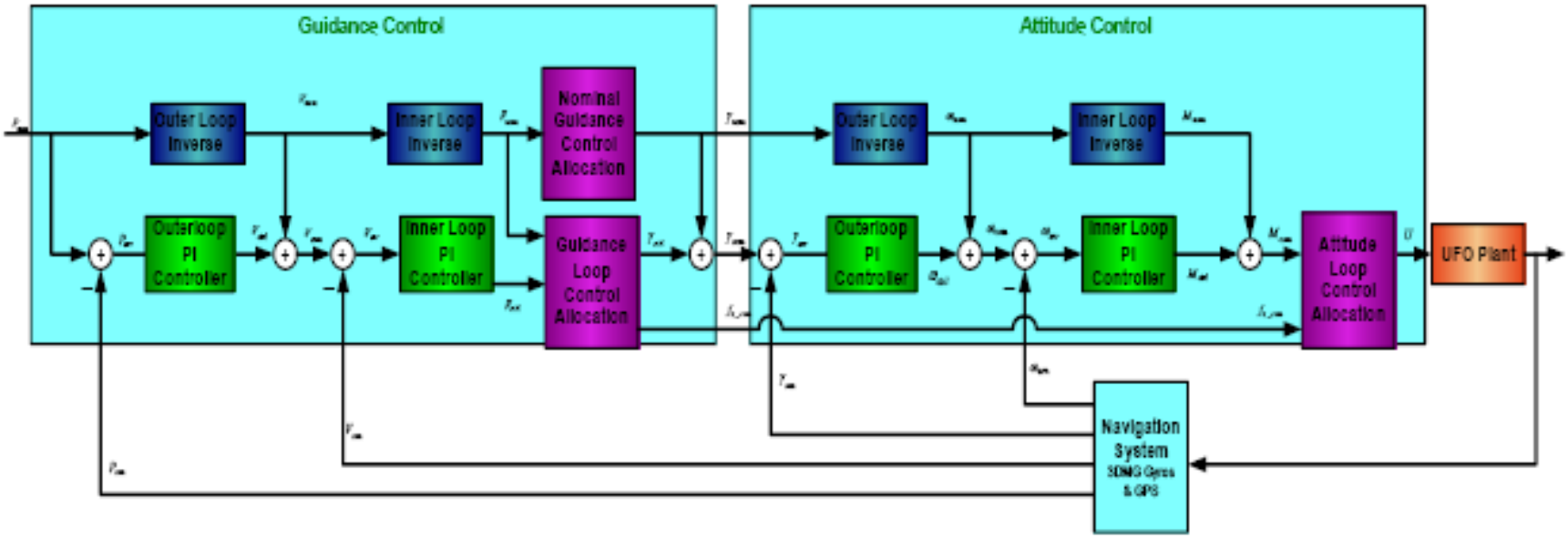
Ritchie Lee, CMU



Trajectory Linearized Controller

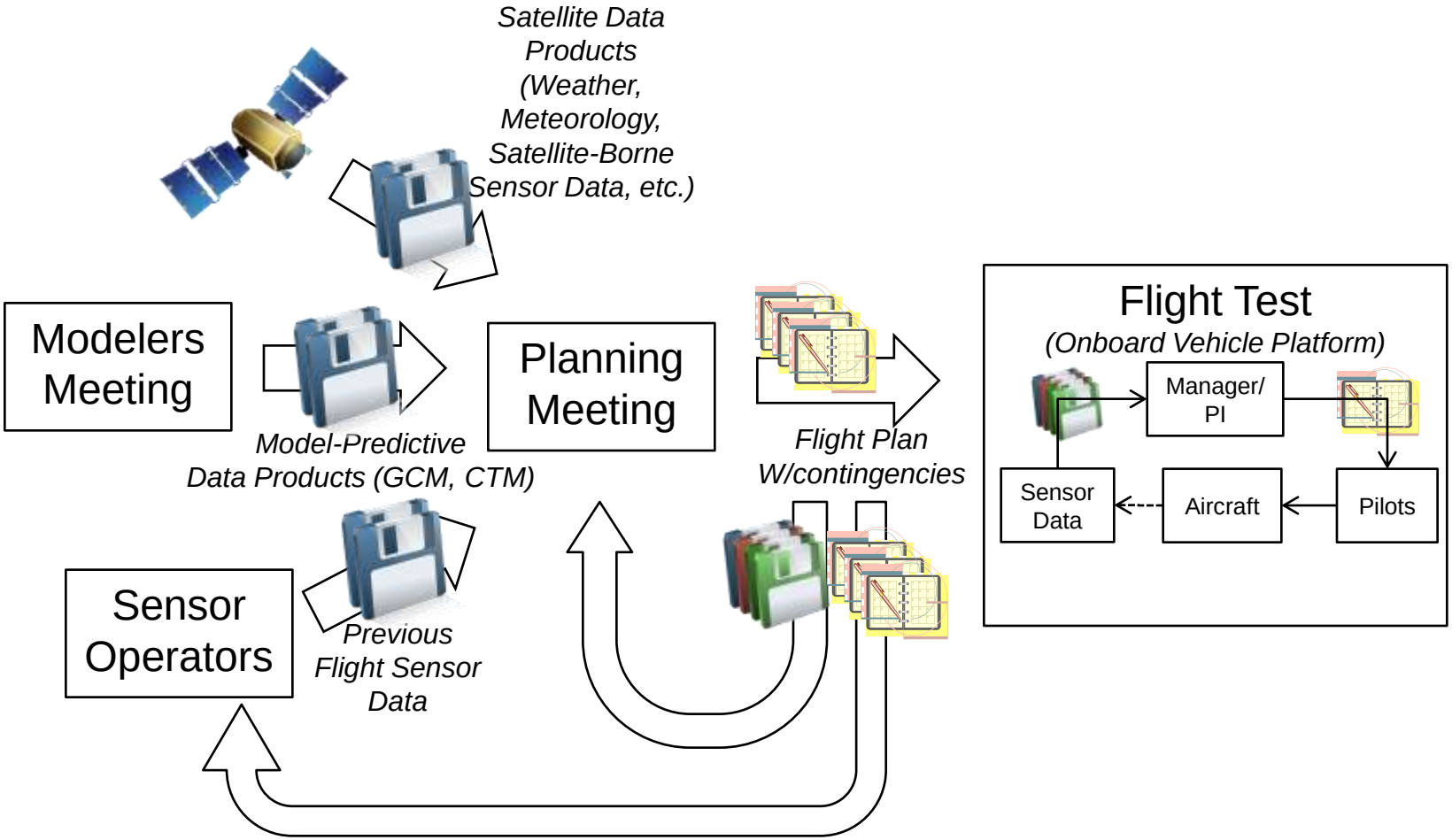
Tony Adami, Ohio University





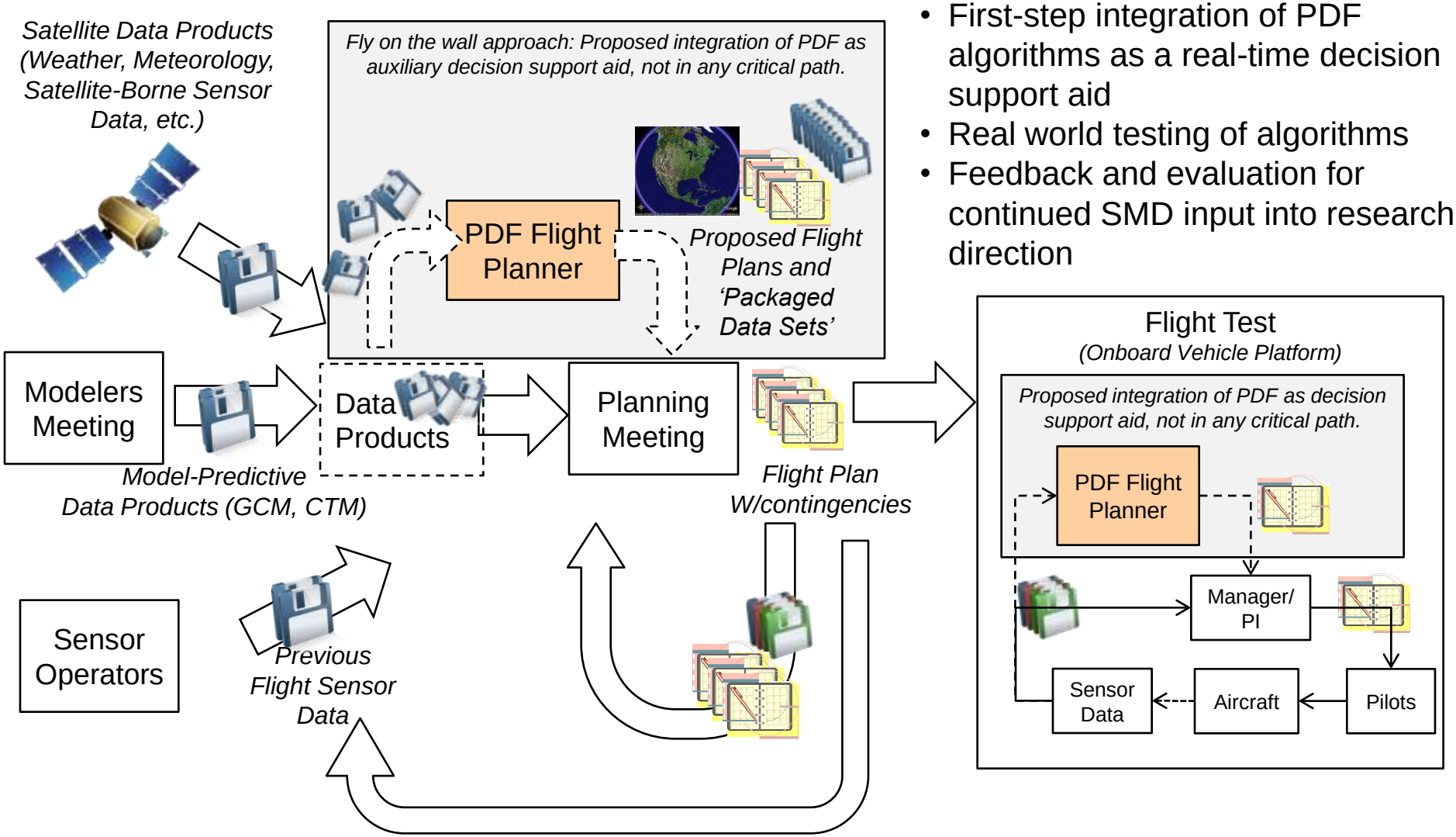
Analysis of Integration with NASA Airborne Science Missions

ARCTAS Planning and Operations Model



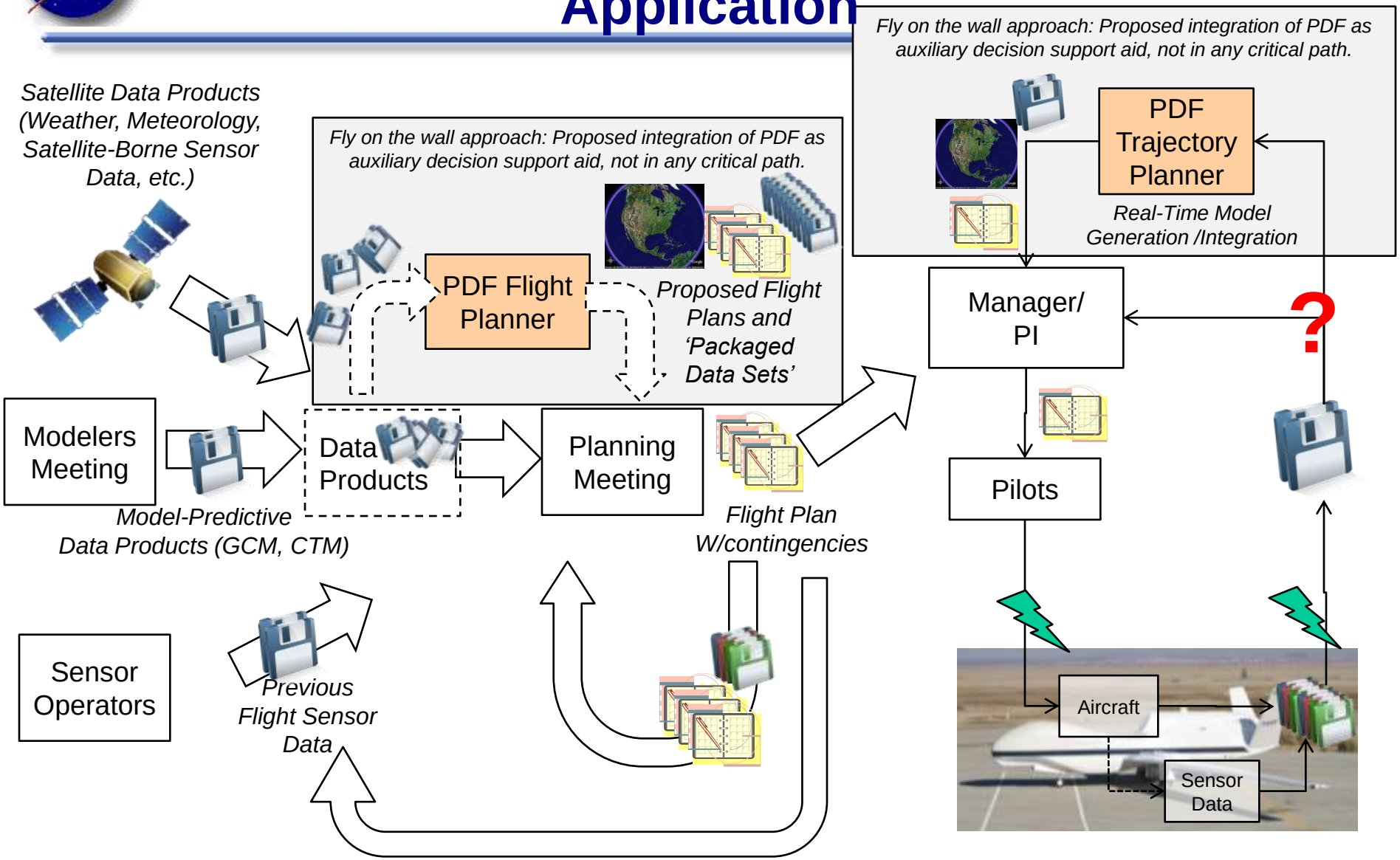
Observed Model from 2008 ARCTAS Mission (INTEX-B Heritage)

ACRTAS Style Mission: PDF Conceptual Integration

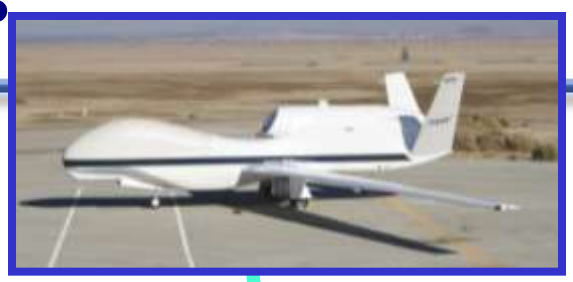


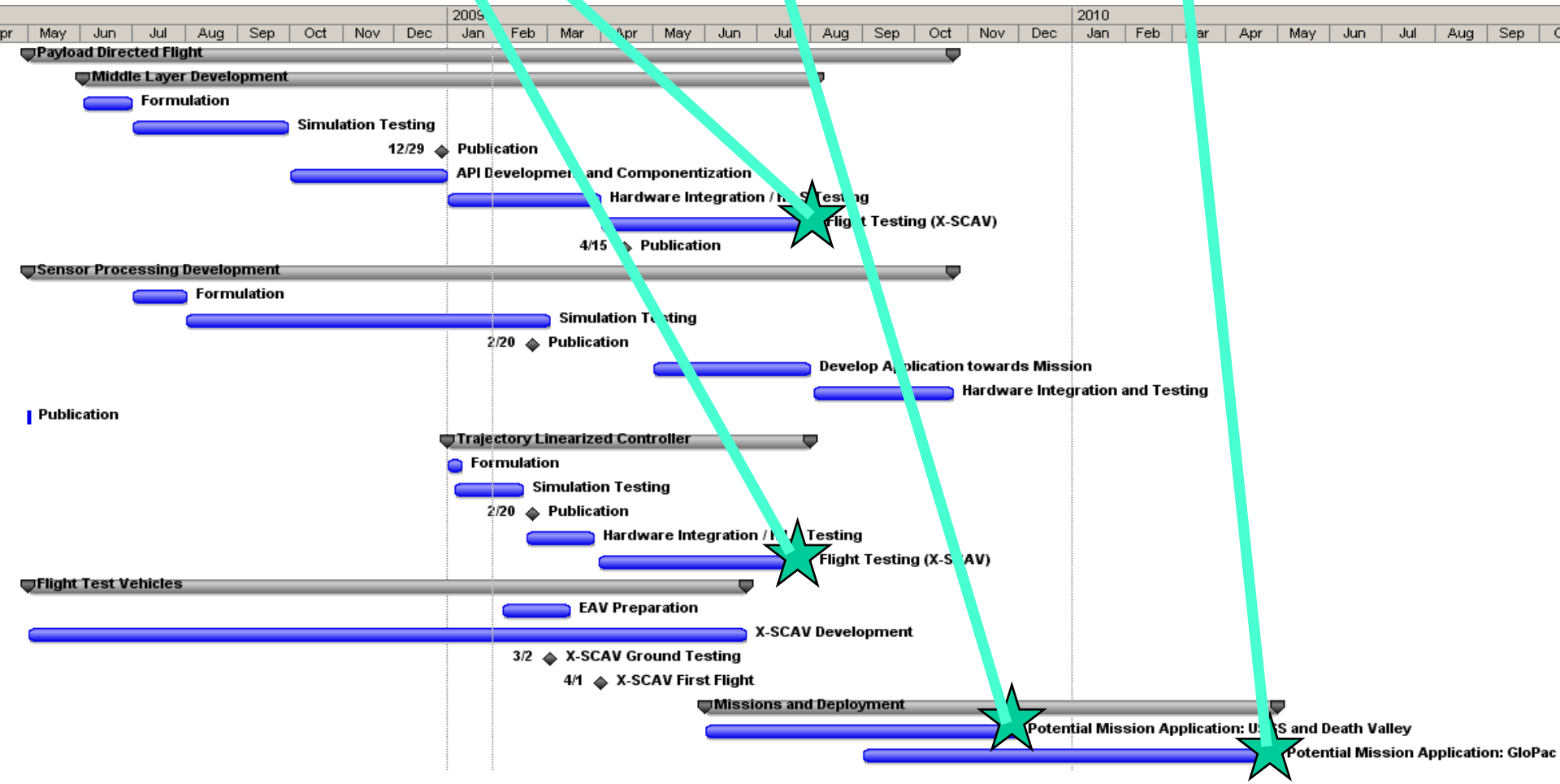
- First-step integration of PDF algorithms as a real-time decision support aid
- Real world testing of algorithms
- Feedback and evaluation for continued SMD input into research direction

Conceptual Integration in UAV Application

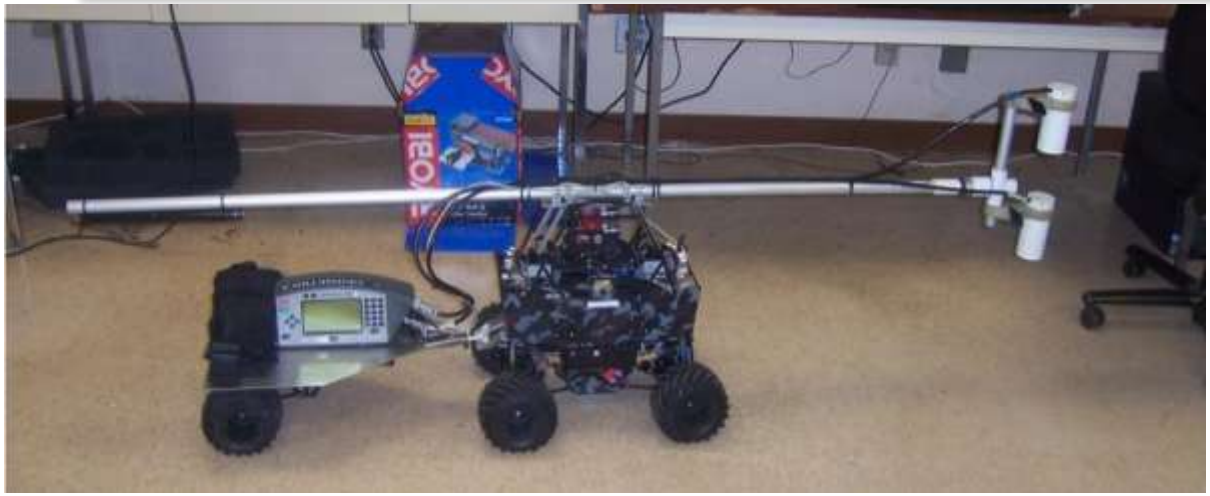


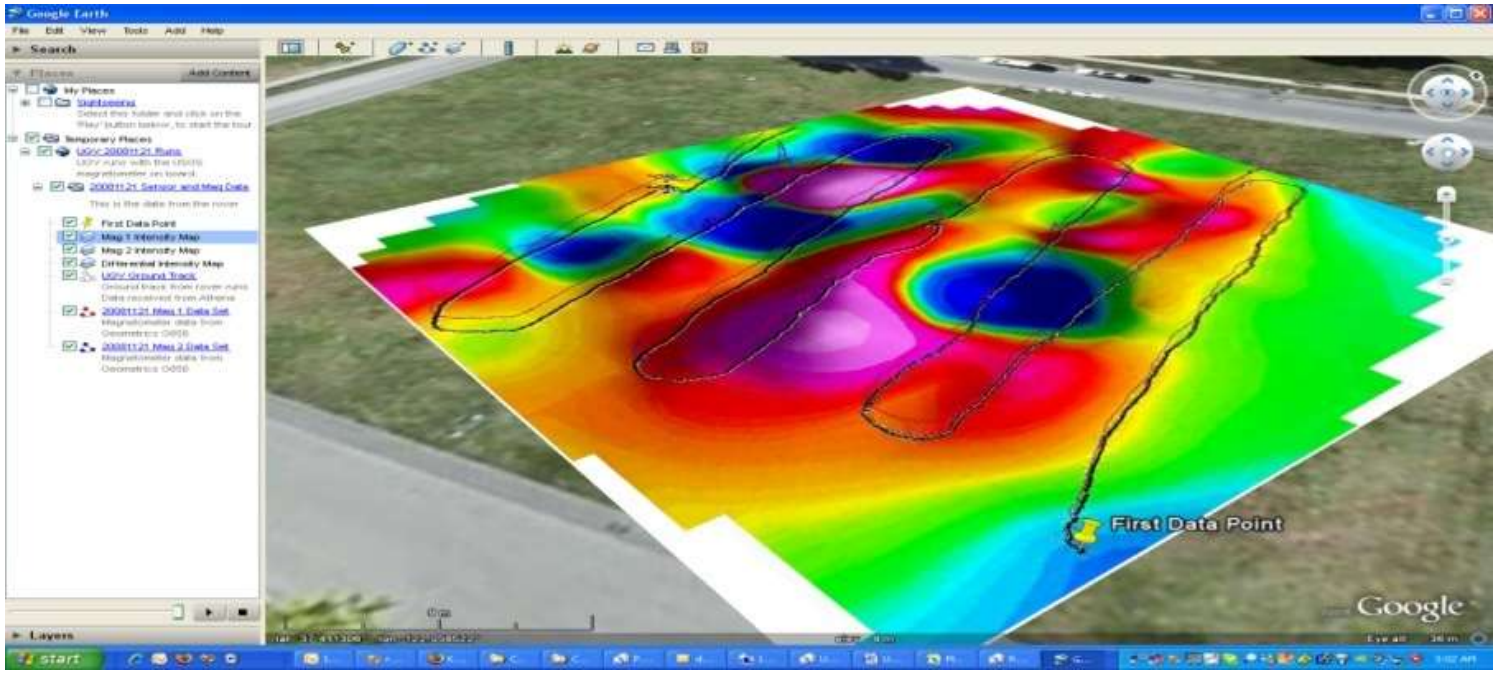
Project Plans





PDF Field Trial: USGS Autonomous Magnetometer Surveying Mission

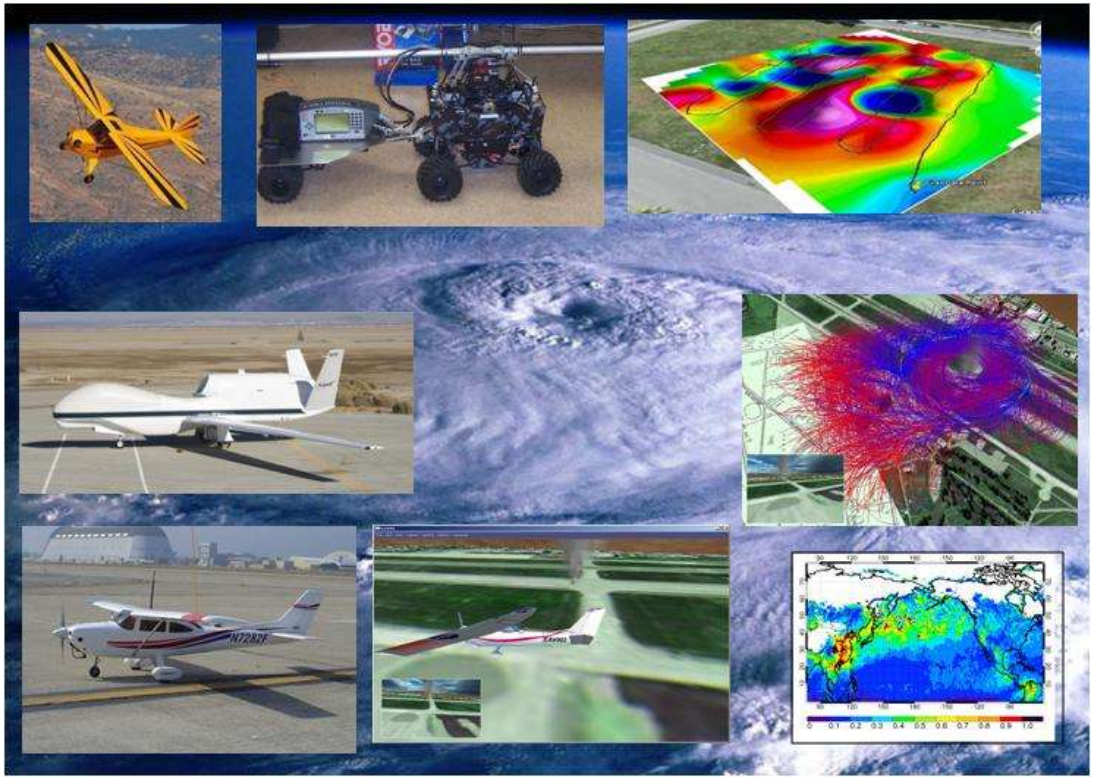




(End of Presentation)

Intelligent Payload Directed Vehicle Autonomy for Earth Science Applications

Corey Ippolito
Intelligent System Division, Code TI
NASA Ames Research Center
Moffett Field, CA 94035
corey.a.ippolito@nasa.gov

Yoo-Hsiu Yeh
Ritchie Lee
Carnegie Mellon University
Silicon Valley Campus
Moffett Field, CA 94035

Tony Adami
Ohio University