Autonomous Flight and Its Challenges

Irene Gregory, Ph.D. S.T.
Andrew Patterson, Ph.D.
NASA Langley Research Center

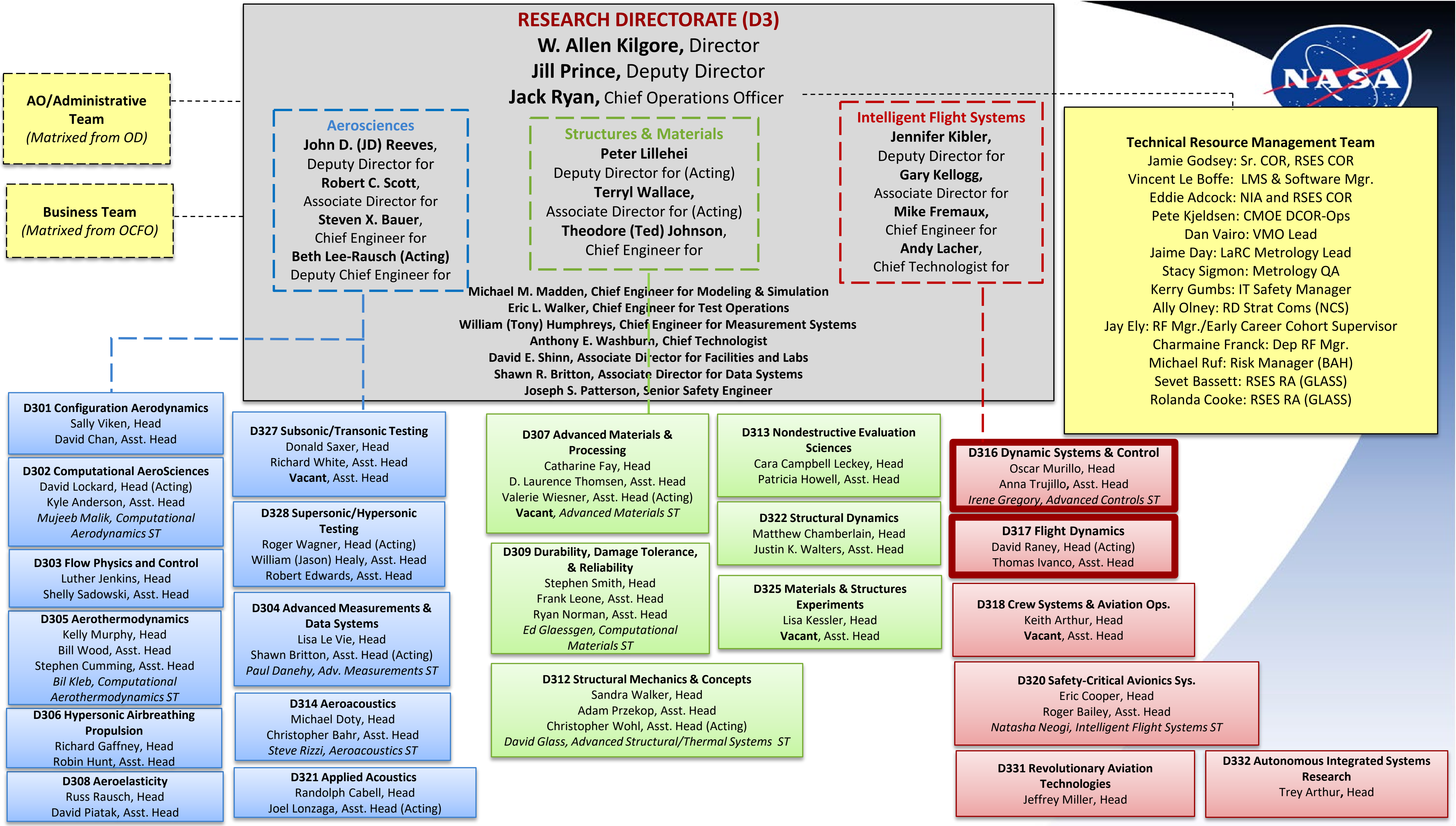
18 April 2024
Seminar
Penn State Aerospace Engineering Department

Abstract

- Aviation is undergoing a revolution and a paradigm change. New technologies are moving aviation towards on-demand transportation. To fully realize the promise of “anyone, anytime, anywhere” transportation, autonomy must play a key role. Our research team is focused on the intersection of new vehicle eVTOL configurations, popularly known as “air taxis”, and autonomous flight in complex urban environment. It has been widely recognized that dealing with contingencies, especially in safe, scalable and flexible way, is the most difficult challenge for autonomy. The presentation is intended to outline what we consider fundamental challenges and describe our current approaches. Moreover, we are working on establishing wide ranging collaborations to address these fundamental autonomy challenges in a relevant environment with real-world assumptions and constraints. Hence, we would like to take this opportunity to discuss open challenge problems with this research community.

Overview

1. Who We Are and What We Do
2. Why Autonomous Flight
3. Challenges in Autonomy with Selected Examples
4. Approach to Developing Autonomy Algorithms
5. Example Algorithm and Results
6. Simulation Environment



RESEARCH DIRECTORATE (D3)

W. Allen Kilgore, Director
Jill Prince, Deputy Director
Jack Ryan, Chief Operations Officer

AO/Administrative Team
(Matrixed from OD)

Business Team
(Matrixed from OCFO)

Aerosciences
John D. (JD) Reeves, Deputy Director for
Robert C. Scott, Associate Director for
Steven X. Bauer, Chief Engineer for
Beth Lee-Rausch (Acting) Deputy Chief Engineer for

Structures & Materials
Peter Lillehei Deputy Director for (Acting)
Terryl Wallace, Associate Director for (Acting)
Theodore (Ted) Johnson, Chief Engineer for

Intelligent Flight Systems
Jennifer Kibler, Deputy Director for
Gary Kellogg, Associate Director for
Mike Fremaux, Chief Engineer for
Andy Lacher, Chief Technologist for

Michael M. Madden, Chief Engineer for Modeling & Simulation
Eric L. Walker, Chief Engineer for Test Operations
William (Tony) Humphreys, Chief Engineer for Measurement Systems
Anthony E. Washburn, Chief Technologist
David E. Shinn, Associate Director for Facilities and Labs
Shawn R. Britton, Associate Director for Data Systems
Joseph S. Patterson, Senior Safety Engineer

Technical Resource Management Team
Jamie Godsey: Sr. COR, RSES COR
Vincent Le Boffe: LMS & Software Mgr.
Eddie Adcock: NIA and RSES COR
Pete Kjeldsen: CMOE DCOR-Ops
Dan Vairo: VMO Lead
Jaime Day: LaRC Metrology Lead
Stacy Sigmon: Metrology QA
Kerry Gumbs: IT Safety Manager
Ally Olney: RD Strat Coms (NCS)
Jay Ely: RF Mgr./Early Career Cohort Supervisor
Charmaine Franck: Dep RF Mgr.
Michael Ruf: Risk Manager (BAH)
Sevet Bassett: RSES RA (GLASS)
Rolanda Cooke: RSES RA (GLASS)

D301 Configuration Aerodynamics
Sally Viken, Head
David Chan, Asst. Head

D302 Computational AeroSciences
David Lockard, Head (Acting)
Kyle Anderson, Asst. Head
Mujeeb Malik, Computational Aerodynamics ST

D303 Flow Physics and Control
Luther Jenkins, Head
Shelly Sadowski, Asst. Head

D305 Aerothermodynamics
Kelly Murphy, Head
Bill Wood, Asst. Head
Stephen Cumming, Asst. Head
Bil Kleb, Computational Aerothermodynamics ST

D306 Hypersonic Airbreathing Propulsion
Richard Gaffney, Head
Robin Hunt, Asst. Head

D308 Aeroelasticity
Russ Rausch, Head
David Piatak, Asst. Head

D327 Subsonic/Transonic Testing
Donald Saxer, Head
Richard White, Asst. Head
Vacant, Asst. Head

D328 Supersonic/Hypersonic Testing
Roger Wagner, Head (Acting)
William (Jason) Healy, Asst. Head
Robert Edwards, Asst. Head

D304 Advanced Measurements & Data Systems
Lisa Le Vie, Head
Shawn Britton, Asst. Head (Acting)
Paul Danehy, Adv. Measurements ST

D314 Aeroacoustics
Michael Doty, Head
Christopher Bahr, Asst. Head
Steve Rizzi, Aeroacoustics ST

D321 Applied Acoustics
Randolph Cabell, Head
Joel Lonzaga, Asst. Head (Acting)

D307 Advanced Materials & Processing
Catharine Fay, Head
D. Laurence Thomsen, Asst. Head
Valerie Wiesner, Asst. Head (Acting)
Vacant*, Advanced Materials ST*

D309 Durability, Damage Tolerance, & Reliability
Stephen Smith, Head
Frank Leone, Asst. Head
Ryan Norman, Asst. Head
Ed Glaessgen, Computational Materials ST

D312 Structural Mechanics & Concepts
Sandra Walker, Head
Adam Przekop, Asst. Head
Christopher Wohl, Asst. Head (Acting)
David Glass, Advanced Structural/Thermal Systems ST

D313 Nondestructive Evaluation Sciences
Cara Campbell Leckey, Head
Patricia Howell, Asst. Head

D322 Structural Dynamics
Matthew Chamberlain, Head
Justin K. Walters, Asst. Head

D325 Materials & Structures Experiments
Lisa Kessler, Head
Vacant, Asst. Head

D316 Dynamic Systems & Control
Oscar Murillo, Head
Anna Trujillo**,** Asst. Head
Irene Gregory, Advanced Controls ST

D317 Flight Dynamics
David Raney, Head (Acting)
Thomas Ivanco, Asst. Head

D318 Crew Systems & Aviation Ops.
Keith Arthur, Head
Vacant, Asst. Head

D320 Safety-Critical Avionics Sys.
Eric Cooper, Head
Roger Bailey, Asst. Head
Natasha Neogi, Intelligent Flight Systems ST

D331 Revolutionary Aviation Technologies
Jeffrey Miller, Head

D332 Autonomous Integrated Systems Research
Trey Arthur, Head

Who We Are and What We Do

Who we are:

- Small research team with expertise in flight dynamics, general control [flight control, path/trajectory planning], machine learning, data science and simulation development

What we do

- Research that is trying to push the boundaries of autonomous flight with focus on urban air mobility and autonomous cargo

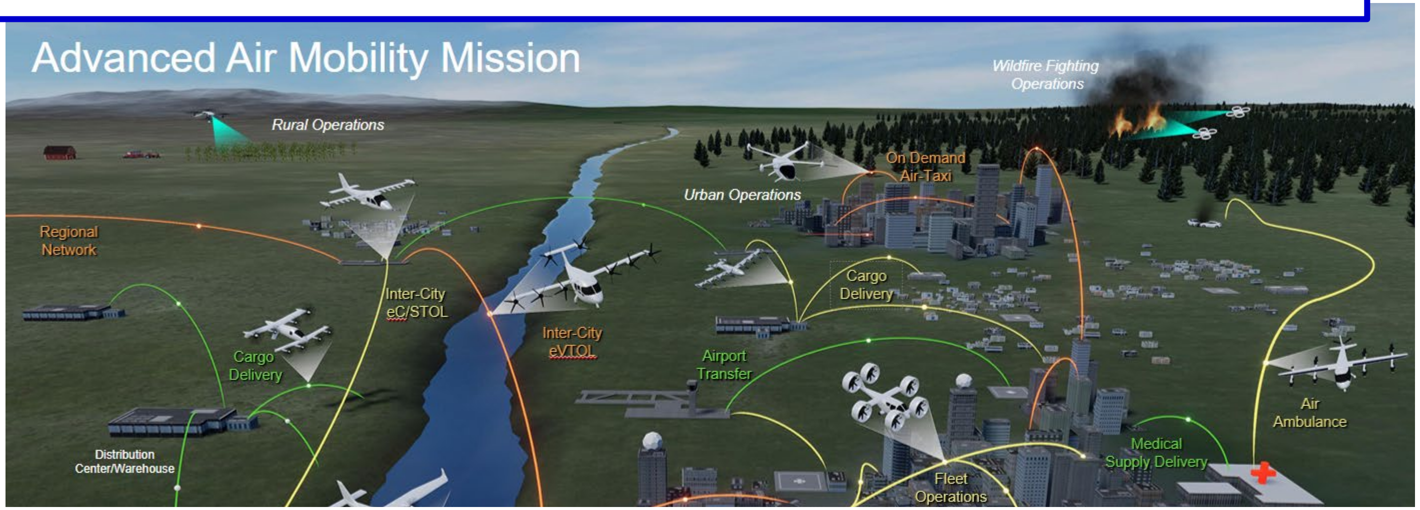
Why we do it

Where are we heading in Aeronautics? – Advanced Air Mobility

Safe, sustainable, affordable, and accessible aviation for transformational local and intraregional missions

Paradigm shift in Aviation

- Anyone, Anywhere, Anytime concept
- On demand air transportation
  - Urban suburban, rural, inter-city
  - Moves people and cargo
- Largely enabled by electrification and automation



Benefits of Autonomy

- System wide **performance** improvements, maximizes capability for fleet/vehicle operations over human operator
- Enhance aviation **sustainability**
- Maintains/enhances **safety** as density of heterogeneous fleet and operations increases

Autonomy is REQUIRED to enable paradigm shift

Autonomy must be implemented in a safe, efficient, scalable, certifiable way

(Autonomous) Intelligent Contingency Management

We have Nominal Autonomous Operations Nailed!!

Off-Nominal Autonomous Operations Major Research Challenge!!

Human (ICM):

- Aviate

- Navigate

- Communicate

Autonomous (ICM):

- **Robust & Adaptive Control**
- **Safety (Certificates) & Learning Control**
- **Failure ID & Flt Envelope Estimation**
- **Collision Avoidance & Pattern Entry**
- **Perception & Environment…**
- **ML training & off training guarantees**

- Planners: Long, Short, Contingency (Spectrum)
- Perception & Environment
- Recognize Contingency /Failures and Replan..
- Multi-path and Optimizations…

- **Algorithm to algorithm (asynchronous, timescales)**
- **Aircraft system to system**
- **External aircraft (datalink, voice?)**
- **System of system communications**
- **Sift out Faulty Information…**

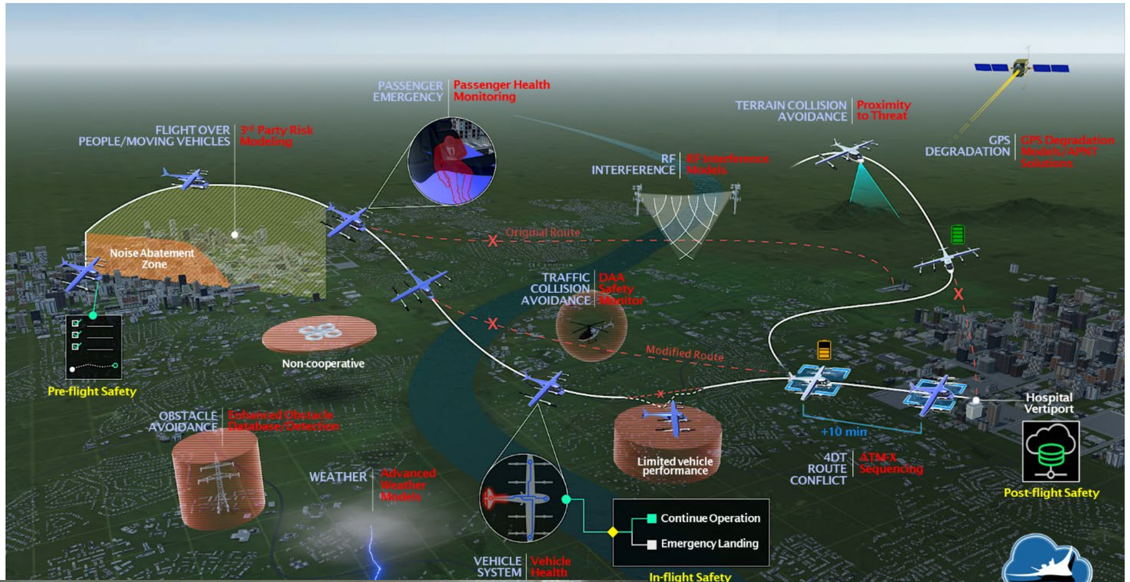
Autonomous Flight - Motivation for Intelligent Contingency Management

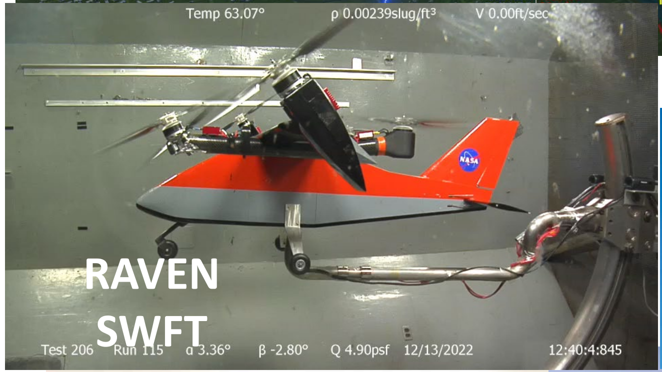
Emerging aviation characteristics:

- **Complexity** of the environment
- Unconventional vehicle configurations with **multi-modal dynamics**
- **Highly nonlinear flight dynamics**

Challenges:

- **Off-nominal events – both common and unforeseen**
- **New technology** – more likely to experience performance degradation/failures
- **Narrow performance margins** – cannot afford conservatism
- **Accurate trajectory following** under system uncertainty and atmospheric disturbances
- **Safe control** while learning elements are engaged





RAVEN SWFT

Wisk

Archer Aviation

Joby Aviation

Why NASA?

- NASA has been a valued partner in **accelerating** maturation and adoption of advanced **technologies** for U.S. industry (industry view)
- NASA serves as a bridge between academia's fundamental research and industry's limited time horizon
- NASA's role should be to **accelerate** and **enable** autonomy in aerospace industry
- **NASA** and key partners can **collectively** consider the **most difficult mission challenges**, **ambitious operations** and help mature required technologies to **enable** US industrial competitiveness and leadership
- NASA can **accelerate U.S. development and deployment** by
  - Providing the data and confidence regulators need to certify and approve
  - Tackling systems approach, integrated individual algorithms/methods
  - Openly sharing knowledge and unifying community around remaining challenges
  - Taking **higher technical risk** and focus beyond immediate time horizon

Overview

1. Who We Are and What We Do
2. Why Autonomous Flight
3. Challenges in Autonomy with Selected Examples
4. Approach to Developing Autonomy Algorithms
5. Example Algorithm and Results
6. Simulation Environment

Challenges in Autonomy with Selected Examples

Perception

runway detection, image segmentation, traffic identification, relative localization, traffic identification, weather monitoring and prediction

Planning

Standard maneuvers, strategic planning, tactical replanning, contingency aware plans

Decision Making

Symbolic reasoning, counterfactual prediction, identifying change points mission goals

Challenge 1

Algorithms may solve individual problems, *but autonomy is a system level problem*.

Each algorithm and system needs to be evaluated with respect to all subcomponents and the overall system.

Challenges within components

Perception

- Do we need to recertify when perception model is trained on new information?
- Can we guarantee safety for all vision inputs?
- Can existing requirements, or their underlying rationale, be directly satisfied with electronic perception?

Planning

- What objectives do we really want to satisfy?
- How do we encode robustness of our plan to uncertainties in perception and possible changes in decisions?
- How can our plan span the full range of dynamic complexity and all timescales of interest?

Decision Making

- How do we change our goals?
- Autonomously create and compare decisions over a set of possible conditions? How do we decide which conditions are worth more computation?

Challenge 2

How do we map *operational considerations* into the *mathematical frameworks* where our solutions can be developed?

Our Approach

Main Idea: Layers of algorithms, mathematical guarantees when possible and exhaustive testing when it is not.

1) Identify subproblems and deploy appropriate framework, e.g., Control Theory, Optimal Control, Machine Learning, Game Theory
2) Deploy and verify solutions to subproblems in framework's assumed environment
3) Identify differences in subproblem assumptions
4) Fill gaps with rules of thumb and heuristics based on current operational rules
5) Deploy in high complexity simulation

Challenge 1
Each algorithm and system needs to be evaluated with respect to all subcomponents and the overall system.

Challenge 2
How do we map *operational considerations* into the *mathematical frameworks* where our solutions can be developed?

This mirrors traditional control design pipelines where one would design for a linear system, implement using well known heuristics such as gain scheduling/ feedback gains selection rules , and verify performance in a nonlinear simulation and flight tests.

Limitations and Discussion

1. **Regulation:** Only discussing algorithm performance, another layer of complexity is added when discussing regulation. Mapping *operational considerations* to *requirements* is its own challenge. (See vs Detect and avoid)
2. **Performance Metrics:** Accurate comparison of systems requires careful and honest bookkeeping of limitations during design and testing. Capturing what doesn't work at this stage is still valuable.
3. **Complexity:** Our approach, and others like it, relies very heavily on extensive testing in complex simulations or real environments. This complexity makes concise discussion difficult. We need new models and abstractions!

Challenge 1
Each algorithm and system needs to be evaluated with respect to all subcomponents and the overall system.

Challenge 2
How do we map *operational considerations* into the *mathematical frameworks* where our solutions can be developed?

Example Algorithm

Specific Case:

Terminal Area Operations w/ eVTOL vehicle

Subproblem Discussion

Traffic prediction, pattern entry, *replanning based on new information*: nominal (adjusting spacing), *off-nominal (collision detected).*

Subproblem Frameworks

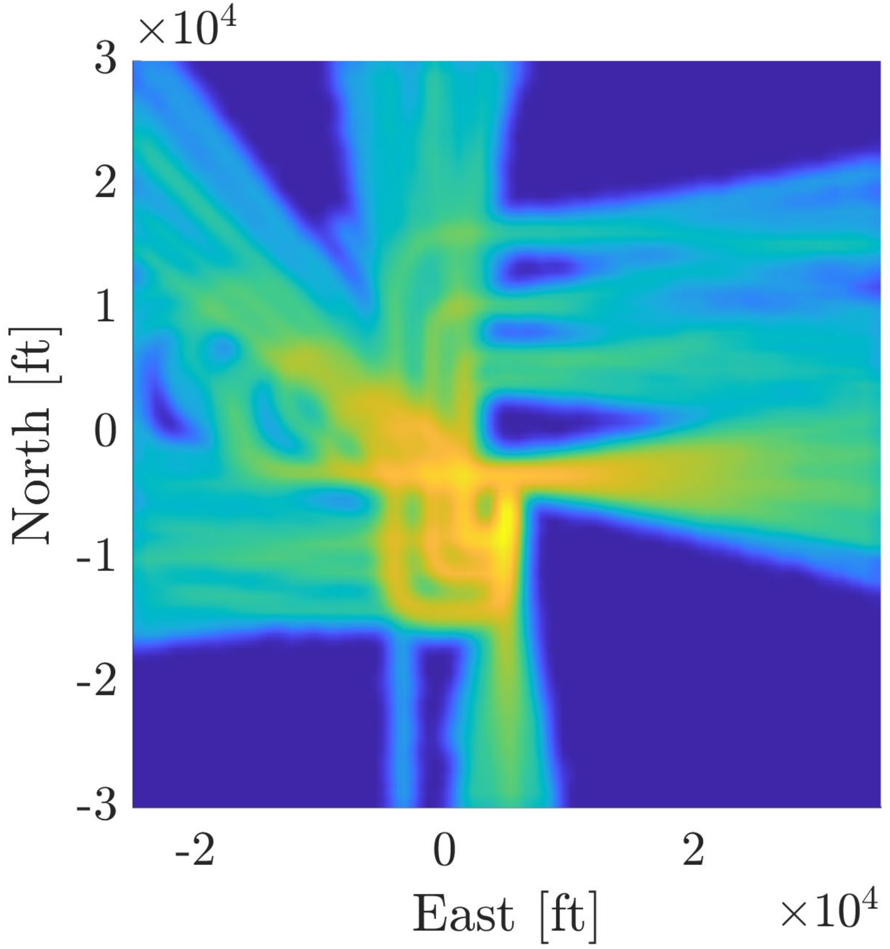
Sensor fusion, nonparametric estimation and prediction, rules-based trajectory prediction, *differentially flat trajectory generation*, global optimization, *local optimization*

Where are the gaps?

There are many!

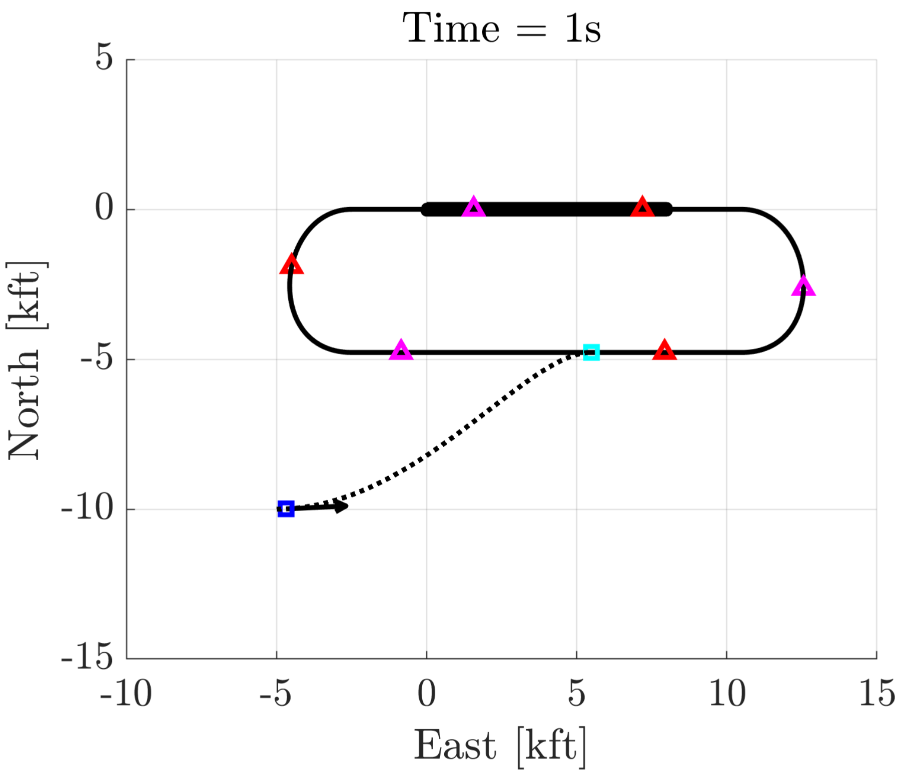
High complexity testing

Algorithms developed with simple models, tested in GUAM.

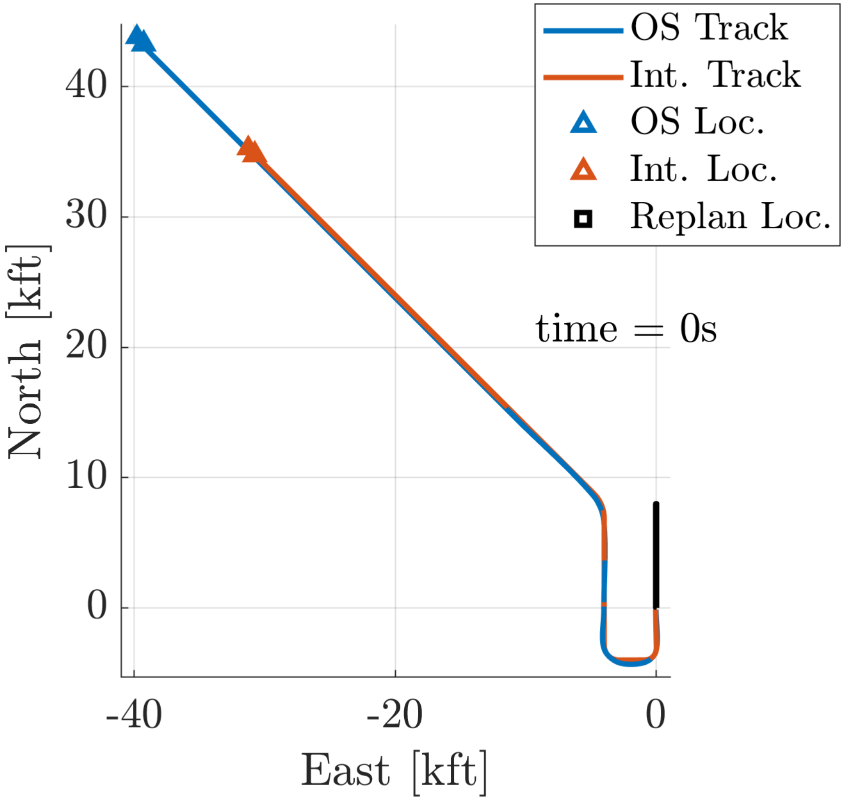


Heatmap of simulated fixed-wing traffic approaching an airport.

Specific Case: Background Algorithms



Existing solution to enter into pattern with appropriate spacing for perfect traffic knowledge.



Add optimization to reduce speed if spacing at runway threshold is not sufficient.

Specific Case: Collision Avoidance Contingencies

Given our initial solution and a simple replanning algorithm to maintain spacing, how do we handle a specific contingency, a pop-up obstacle, possibly unobserved traffic.

Problem: In full scale sim, optimization is costly and low order models miss some of the feasibility concerns.

Solution: System that uses the best parts of several algorithms to span dynamic complexity and time scales.

COBRA-DDP proposed

- Bernstein Polynomials
- Optimal Reciprocal Collision Avoidance
- Differential Dynamic Programming

(CO)mbined (B)ernstein Polynomial, Optimal (R)eciprocal Collision (A)voidance DDP

Trajectory Re-planner Requirements:

- “Real-time” dynamically feasible trajectories for UAM class (transitioning) vehicles with separation assurances
- Dynamic planning for large number of (stationary & moving) cooperative/uncooperative obstacles

Combine to get best of each algorithm!

Piecewise Bernstein Polynomial Curves:

- Advantages: Fast and compact trajectory representation, smooth derivatives (position, velocity & acceleration)
- Disadvantages: one piece-wise segment can’t represent all curves exactly (e.g., circular arcs)

Optimal Reciprocal Collision Avoidance (ORCA):

- Advantages: fast computation for large number of cooperative/non-cooperative with separation assurances
- Disadvantages: no assurance of dynamic feasibility

Differential Dynamic Programming (DDP):

- Advantages: fast computation of dynamically feasible optimal trajectories
- Disadvantages: Degraded computation speed for incorporation of state constraints (e.g., obstacles)

Variable Time Horizon Planner Considerations

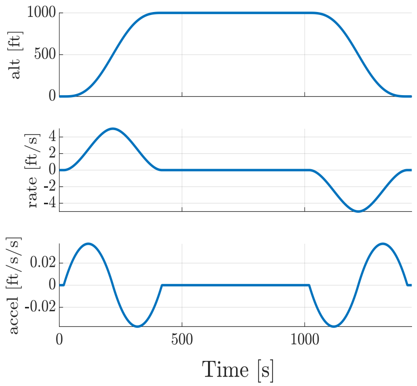
Planner challenges:

- Principled solutions/guarantees
- Accurate trajectory planning & replanning
- Epistemic uncertainty in model
- Multiple operational modes and flight regimes
- Transferability to different vehicles
- Replanning and collision avoidance for *VTOL vehicles* with highly nonlinear dynamics are slow and computationally costly

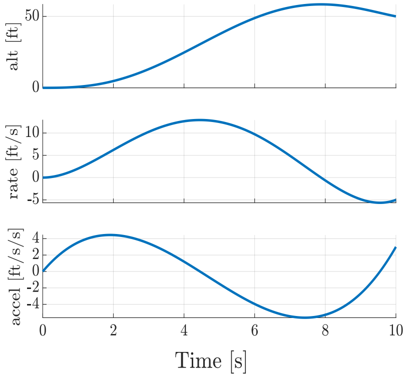
Trajectory Generation Dynamic Complexity Proposed Levels

(Less) < Dynamic Complexity -> (More)

Strategic
Pre-determined velocity and acceleration profiles



Tactical
Arbitrary derivative profiles



Full State Dynamic
Vehicle specific dynamic behavior

$$\dot{p} = \mathbf{v},$$
$$\dot{\eta} = S\omega,$$
$$\dot{\bar{\mathbf{v}}} = -\dot{\psi} e_3 \times \bar{\mathbf{v}} + g + m^{-1}\bar{F}\left(\bar{\mathbf{v}}, \omega, \bar{R}, u\right),$$
$$J\dot{\omega} = -\omega \times J\omega + \tau\left(\bar{\mathbf{v}}, \omega, \bar{R}, u\right),$$

- All of these levels must be **compatible** – something learned or constructed at one level can be transferred to the others since overall **algorithm spans complexity**
- Provides freedom to design at highest level and automatically generate something flyable, only need to interact with different levels of detail as needed.

Bernstein Polynomials

Basics

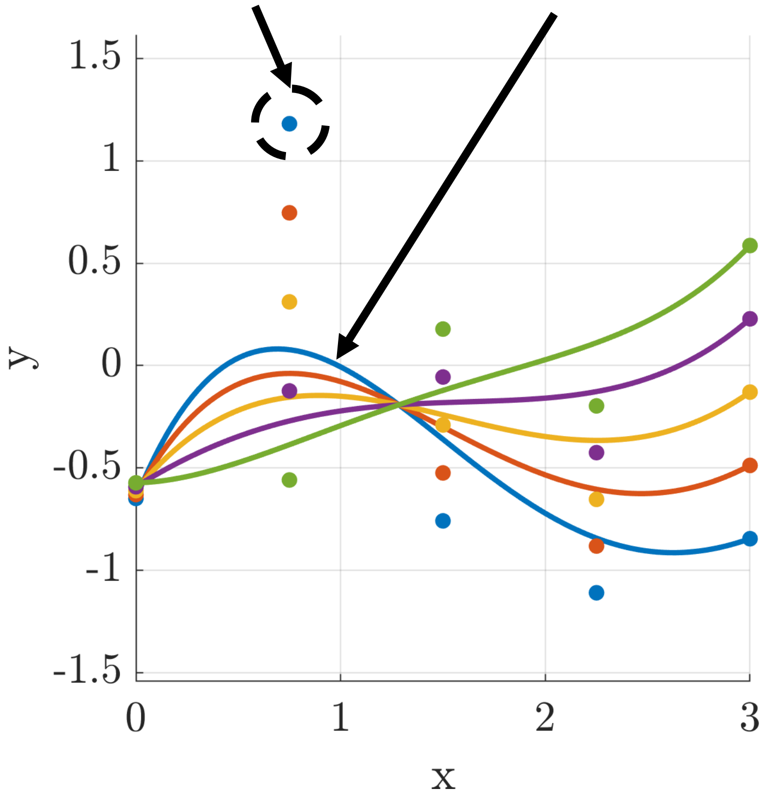
- Polynomial curves using Bernstein basis (rather than monomial basis)
- Polynomial coefficients become control points

Benefits of Bernstein Form

- Control points have physical interpretation
  - Curve connected to end points
  - Curve contained inside control points
- Fast collision and constraint checking algorithms
  - Dynamic feasibility checks
- Differentiation yields Bernstein curve

Control point

Curve Evaluation



Bernstein Polynomial control points and evaluations.

Dynamic Waypoints

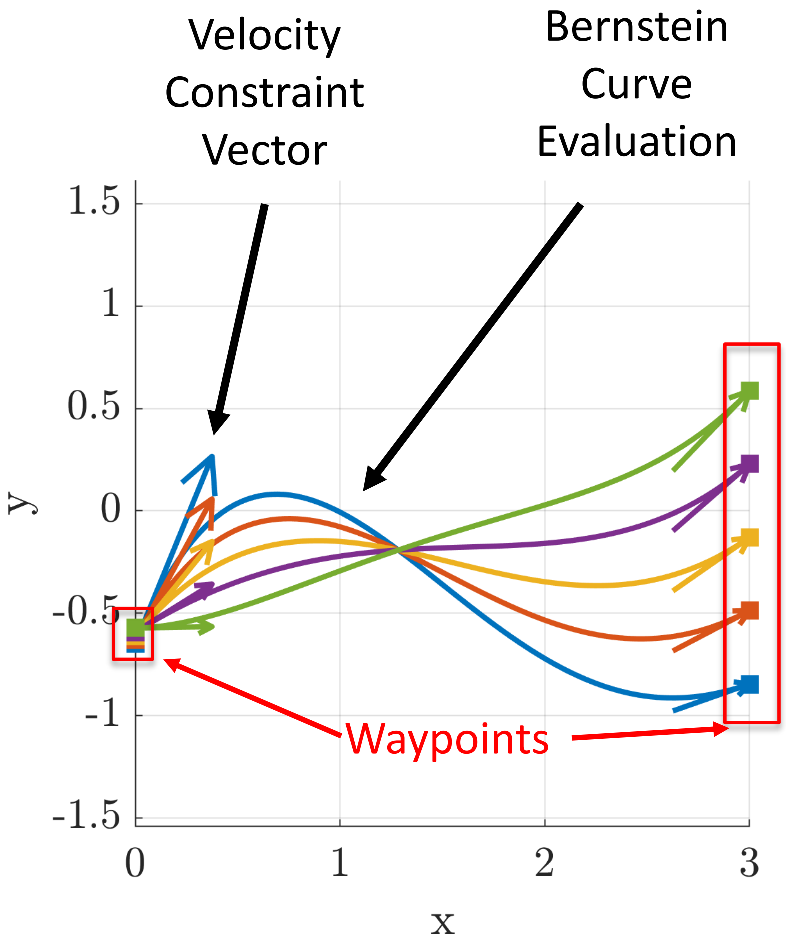
- Waypoints allow specification of spatial/dynamic initial/terminal constraints.
- *e.g. climb from 100 to 200 ft and accelerate from 0 fps to 4 fps.*

Conversion

- Generate Bernstein control points from waypoints (matrix multiplication)
- Bernstein polynomials are the back-end

Use

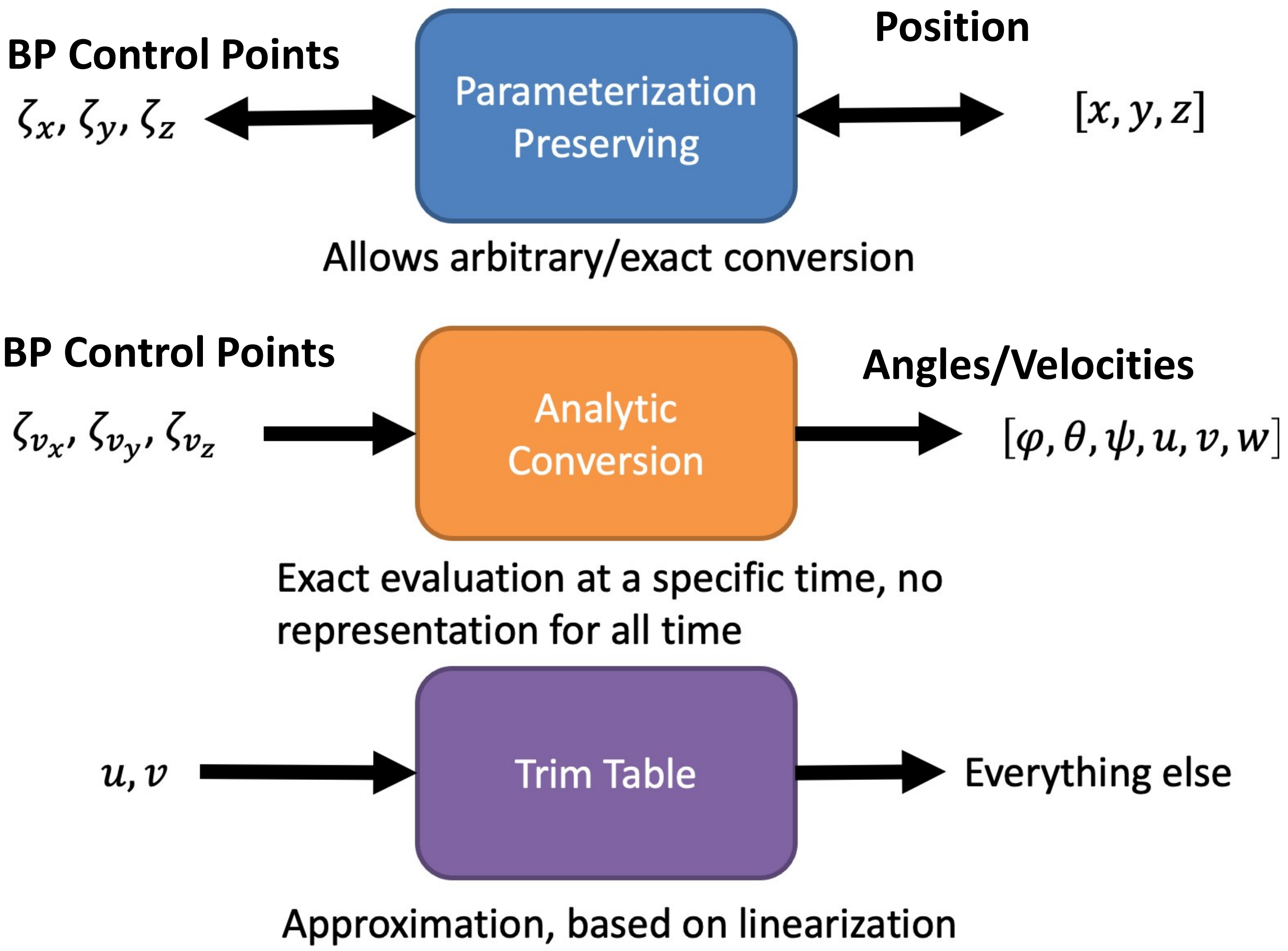
- ORCA waypoints are expressed as terminal constraints on time, position, velocity
- DDP samples Bernstein polynomial at arbitrary times



Waypoints connected by polynomial curve evaluations (Only position and velocity constraints shown)

Continuous Trajectory Generation

- BP waypoints are converted into states for DDP to optimize
- Linearized trim table of cruise are interpolated to complete trajectory states/controls
- These nonoptimal states and control serve as strong starting point for DDP optimization
- BP waypoints can be sampled efficiently at whatever rate DDP requires



BP Control Points

$\zeta_x, \zeta_y, \zeta_z$

Parameterization Preserving

Position

$[x, y, z]$

Allows arbitrary/exact conversion

BP Control Points

$\zeta_{v_x}, \zeta_{v_y}, \zeta_{v_z}$

Analytic Conversion

Angles/Velocities

$[\varphi, \theta, \psi, u, v, w]$

Exact evaluation at a specific time, no representation for all time

u, v

Trim Table

Everything else

Approximation, based on linearization

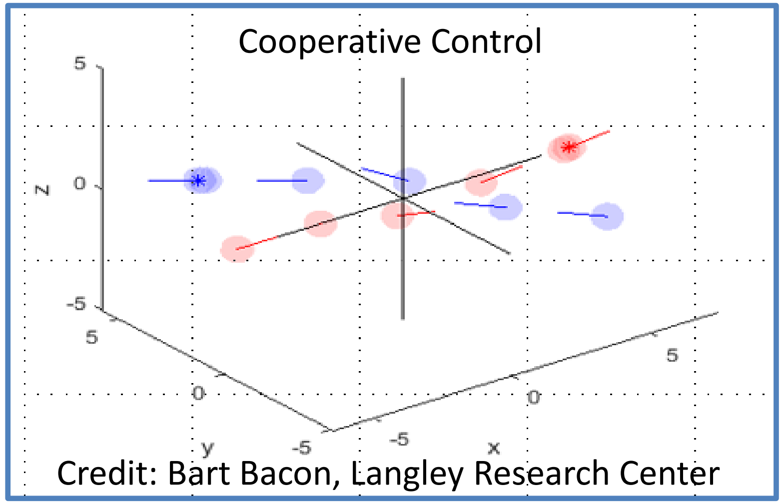
ORCA

Optimal Reciprocal Collision Avoidance Algorithm (robotics community focused):

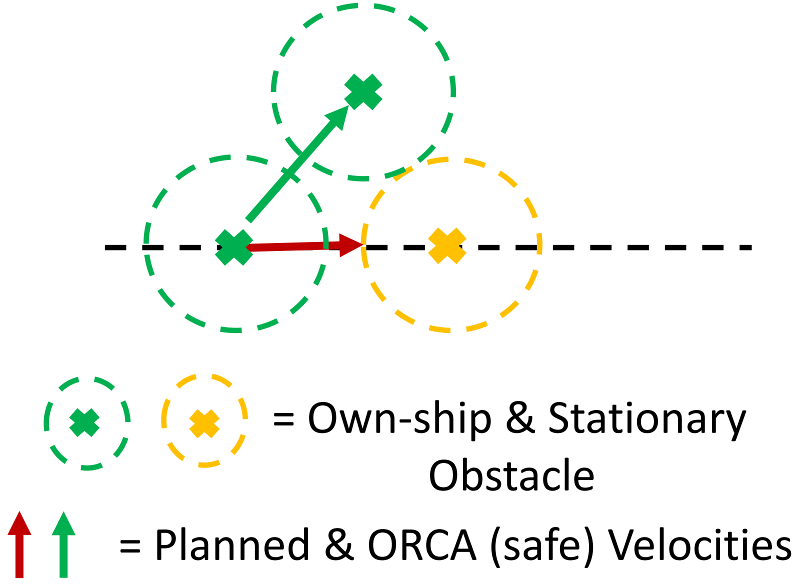
Collision Avoidance: robotics literature defines as autonomous robot navigation with fixed/moving obstacles (other intelligent vehicles)
Recurring cycle: sense/act, repeat

ORCA:

- Input: position and velocity knowledge (own-ship, obstacles/vehicles)
- Output: next own-ship velocity step (magnitude and direction)
- Point modeling (no vehicle dynamics) with safety sphere (keep-out radius)
- “Velocity object” representations, provide mathematical guarantees of collision free for look-ahead time
- Cooperative law: each vehicles applies ½ velocity correction
- Uncooperative law: own-ship takes 100% of velocity correction


Cooperative Control

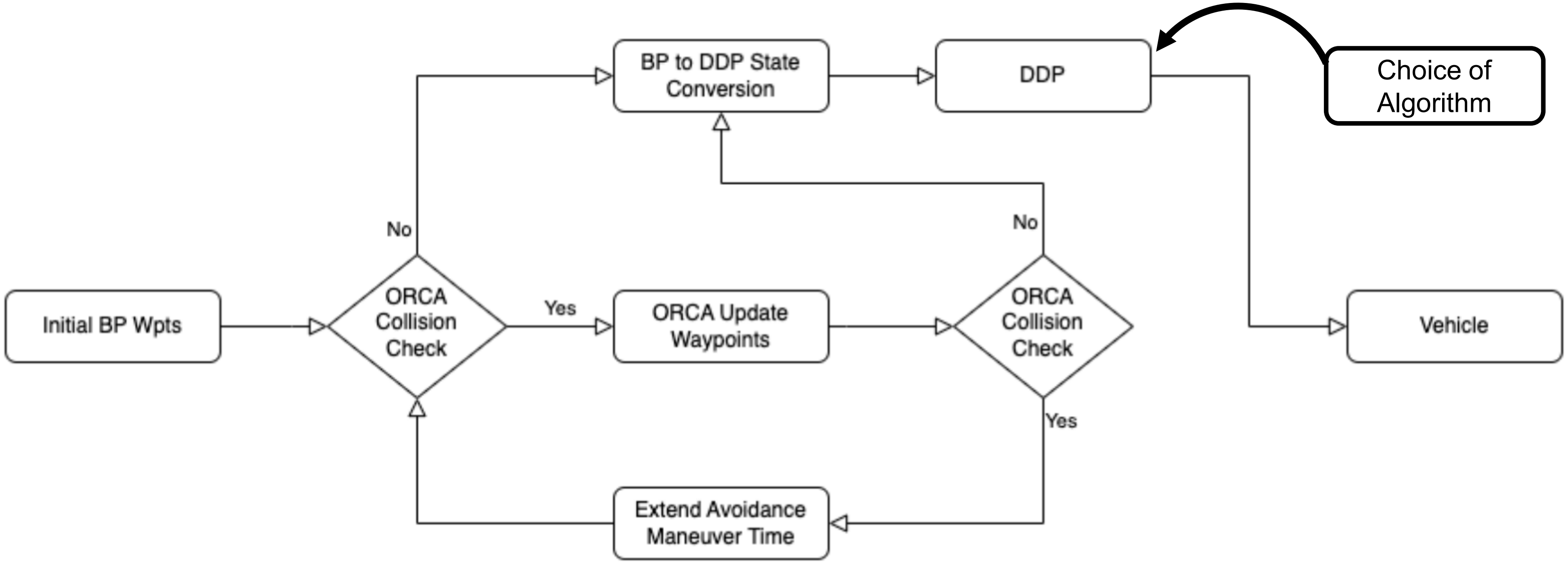
Credit: Bart Bacon, Langley Research Center


✖ ✖ = Own-ship & Stationary Obstacle

↑ ↑ = Planned & ORCA (safe) Velocities

Combination

- Leverages ORCA fast collision avoidance checks and preferential avoidance direction selection
- BP's serve as compact trajectory representation between ORCA and DDP that can be quickly evaluated at any time along the curve
- DDP provides short time horizon dynamically feasible optimal trajectories given simplified trajectory



Optimal Control Problem

Problem: Find optimal control (and states) to achieve a desired end state while minimizing cost

Discrete system nonlinear dynamics

$$\mathbf{x}_{t+1} = \mathbf{f}(\mathbf{x}_t, \mathbf{u}_t)$$

State Vector

$$\mathbf{x}_t \in \mathbb{R}^n$$

Control Vector

$$\mathbf{u}_t \in \mathbb{R}^m$$

Cost Function

$$\mathcal{J}(\mathbf{U}) = \sum_{t=1}^{T-1} \mathcal{L}(\mathbf{x}_t, \mathbf{u}_t) + \phi(\mathbf{x}_T)$$

Running Cost

$$\mathcal{L}(\mathbf{x}_t, \mathbf{u}_t)$$

Terminal Cost

$$\phi(\mathbf{x}_T)$$

State Trajectory

$$\mathbf{X} := \{\mathbf{x}_1, \ldots, \mathbf{x}_T\}$$

Control Trajectory

$$\mathbf{U} := \{\mathbf{u}_1, \ldots \mathbf{u}_{T-1}\}$$

Finite Time

$$T \in \mathbb{N}^+$$

Differential Dynamic Programming

DDP: Given nominal trajectory, use linear (or quadratic) approx. of system nonlinear dynamics and quadratic approx. of cost to yield updates to optimal controls that quadratically converge

Cost Function

$$\mathcal{J}_i(\mathbf{x}_i, \mathbf{U}_i) := \sum_{t=i}^{T-1} \mathcal{L}(\mathbf{x}_t, \mathbf{u}_t) + \phi(\mathbf{x}_T)$$

Truncated Control Sequence

$$\mathbf{U}_i := \{\mathbf{u}_i, \ldots, \mathbf{u}_{T-1}\}$$

Bellman's Principle of Optimality: find overall optimal control as sequence minimization for each truncated control sequence backwards in time (cost-to-go)

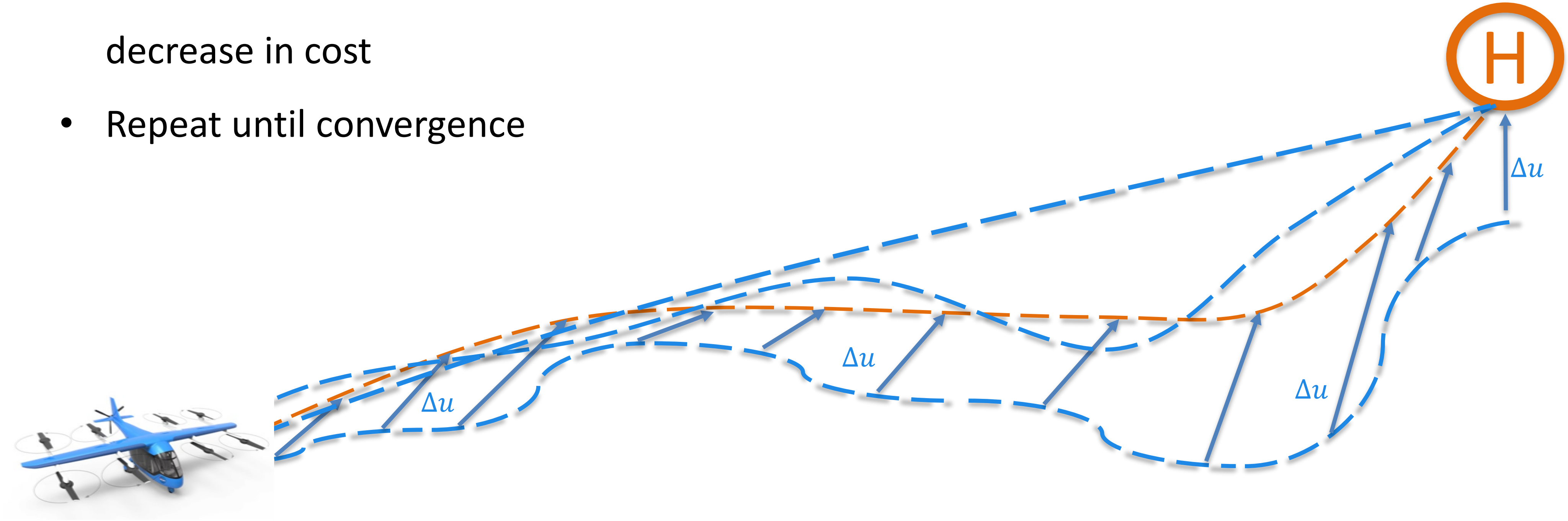
$$V(\mathbf{x}_i) = \min_{\mathbf{u}_i} \Big[\underbrace{\mathcal{L}(\mathbf{x}_i, \mathbf{u}_i) + V(\mathbf{x}_{i+1})}_{Q(\mathbf{x}_i, \mathbf{u}_i)} \Big]$$

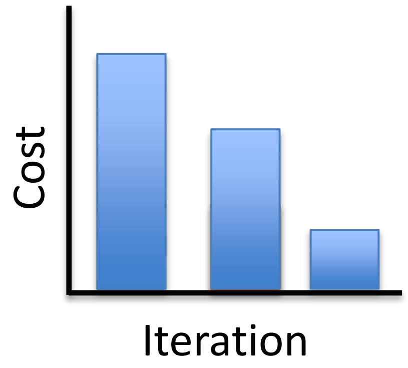
Differential Dynamic Programming

- Given: Initial state x_0, nominal control trajectory $\mathbf{u}_{0:T-1}$
- Repeat until convergence:
  - Forward pass:
    - Forward simulate dynamics from x_0 using $\mathbf{u}_{0:T-1}$ to get state trajectory $\mathbf{x}_{0:T}$
    - Compute derivatives of dynamics $f, f_x, f_u, f_{xx}, f_{xu}, f_{ux}, f_{uu}$ at each time t
    - Compute costs and derivatives $\ell, \ell_x, \ell_u, \ell_{xx}, \ell_{xu}, \ell_{ux}, \ell_{uu}$ at each time t
  - Backward pass:
    - Compute quadratic value function expansion $Q, Q_x, Q_u, Q_{xx}, Q_{xu}, Q_{ux}, Q_{uu}$ at each time t
    - Compute feedforward and feedback gains k and K at each time t
    - Update control $u_t \leftarrow u_t + \alpha k_t + K_t \delta x_t$
      - $\alpha \in [0, 1]$ is a learning rate that is tuned
      - Perform line search on α to heuristically find the best learning rate

Differential Dynamic Programming

- Apply nonlinear dynamics to initial trajectory, $\boldsymbol{x_0}, \boldsymbol{u}$
- Find controls that minimize expected cost using approx. of cost and dynamics, $\Delta\boldsymbol{u}$
- Apply updated controls to determine if new trajectory leads to a decrease in cost
- Repeat until convergence





Augmented Lagrangian

Augmented Lagrangian DDP (AL-DDP):
Adds state constraints to the original optimal control problem
Convert single constrained problem into series of unconstrained problems using penalty functions
Optimization for state constraints greatly **increases computational complexity**, requires an **inner and outer loop**

$$\min_{\mathbf{U}} \tilde{\mathcal{J}}(\mathbf{U}) = \min_{\mathbf{U}} \sum_{t=0}^{T-1} \tilde{\mathcal{L}}(\mathbf{x}_t, \mathbf{u}_t, \lambda_t, \mu_t) + \tilde{\phi}(\mathbf{x}_T, \lambda_T, \mu_T),$$

$$\text{subject to } \mathbf{x}_{t+1} = f(\mathbf{x}_t, \mathbf{u}_t), \ \forall t = 0, \ldots, T-1,$$

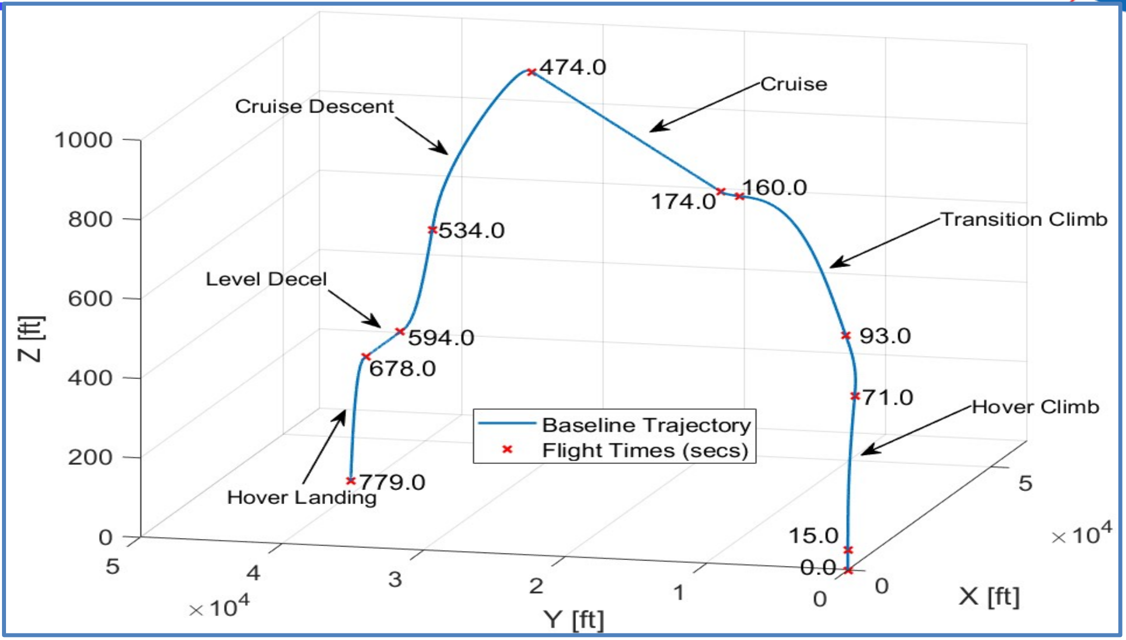
where

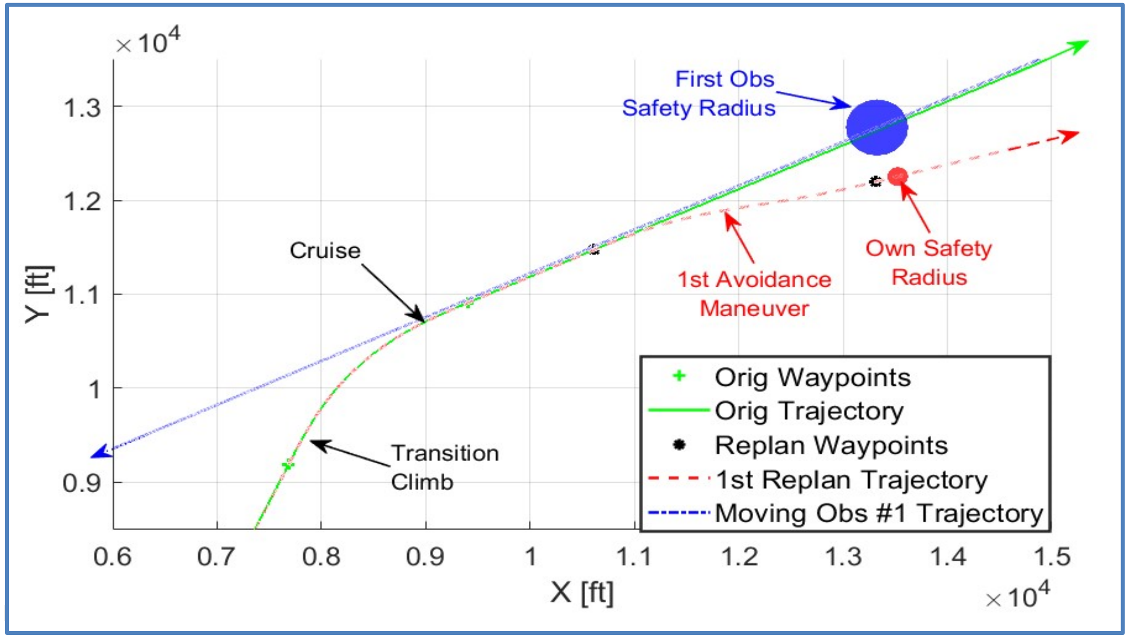
$$\tilde{\mathcal{L}}(\mathbf{x}_t, \mathbf{u}_t, \lambda_t, \mu_t) = \mathcal{L}(\mathbf{x}_t, \mathbf{u}_t) + \sum_{i=1}^{c} \mathcal{P}(g_{t,i}(\mathbf{x}), \lambda_{t,i}, \mu_t),$$

$$\tilde{\phi}(\mathbf{x}_T, \lambda_T, \mu_T) = \phi(\mathbf{x}_T) + \sum_{i=1}^{c} \mathcal{P}(g_{T,i}(\mathbf{x}), \lambda_{T,i}, \mu_T).$$

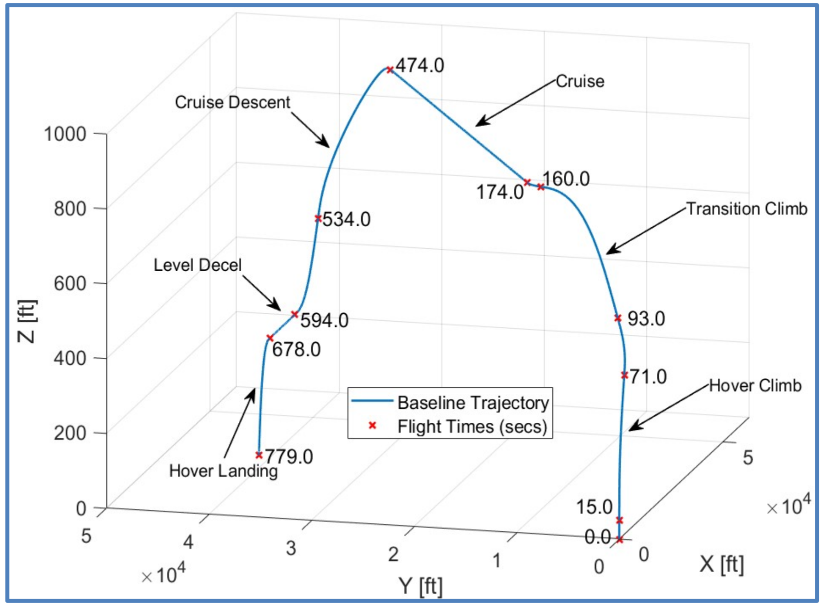
Experimental Result

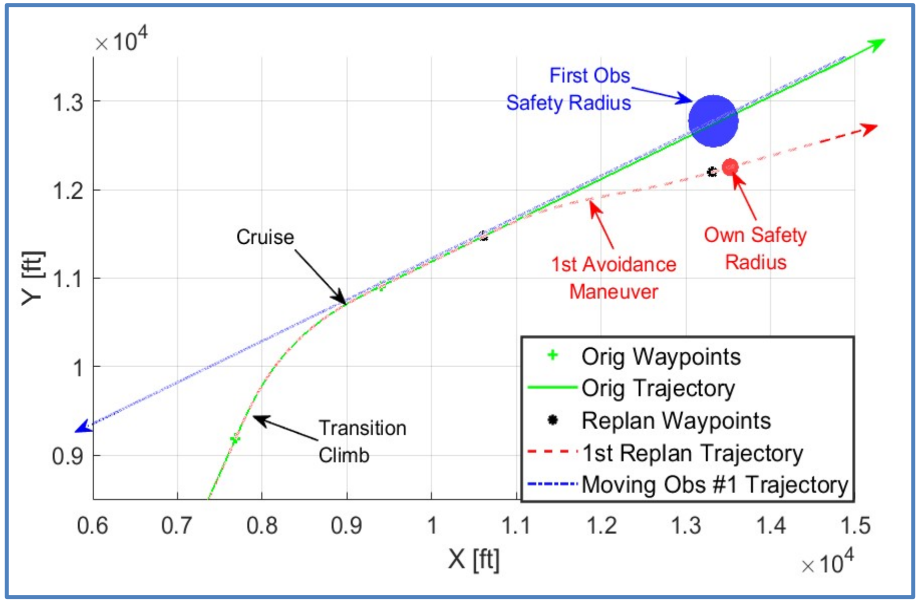
- Full flight tested
  - Vertical Takeoff/Landing
  - Transition
  - Cruise
- Replanning due to multiple moving obstacles
  - Avoidance of one obstacle requires planning to avoid the other
- Replanning descent portion due to large stationary obstacle
- BP-ORCA Time Horizon: 15 seconds
- DDP Time Horizon: 2 seconds

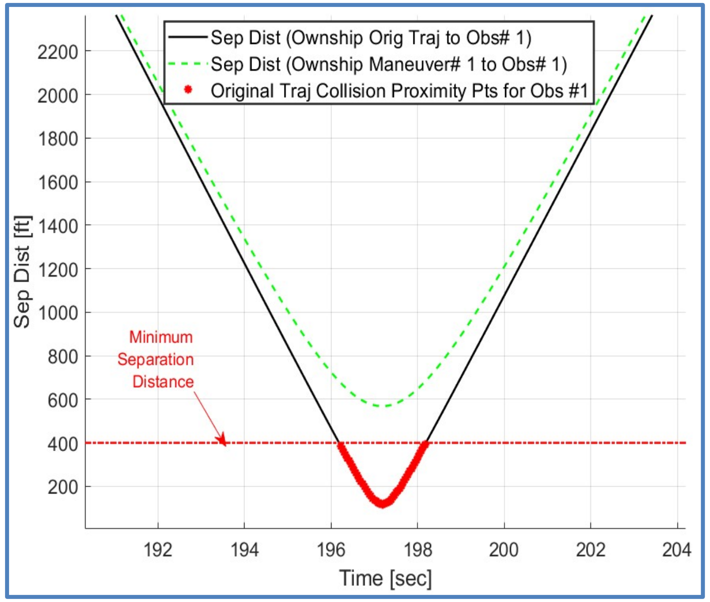


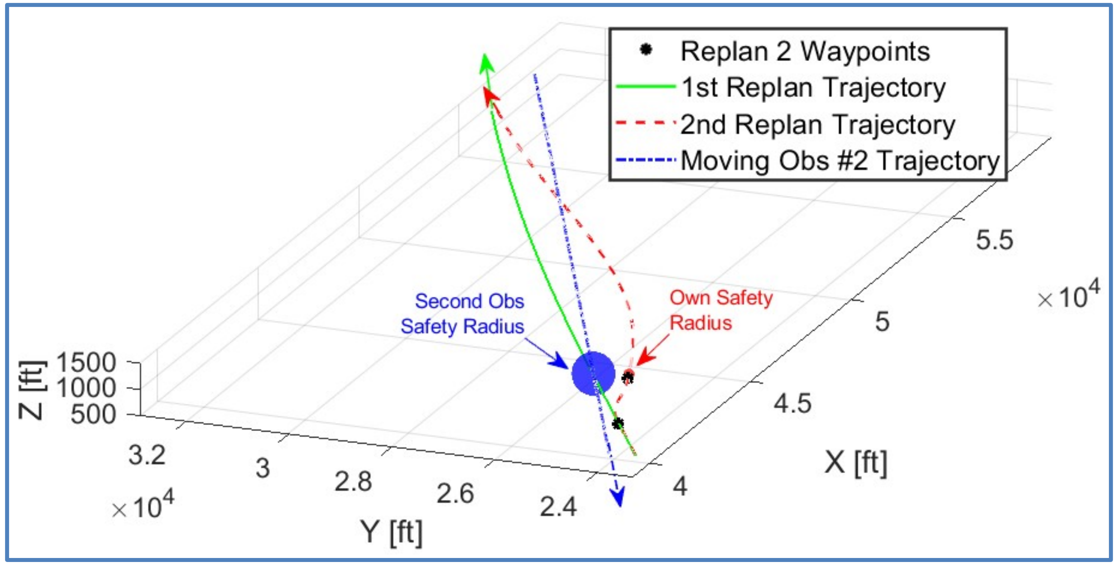


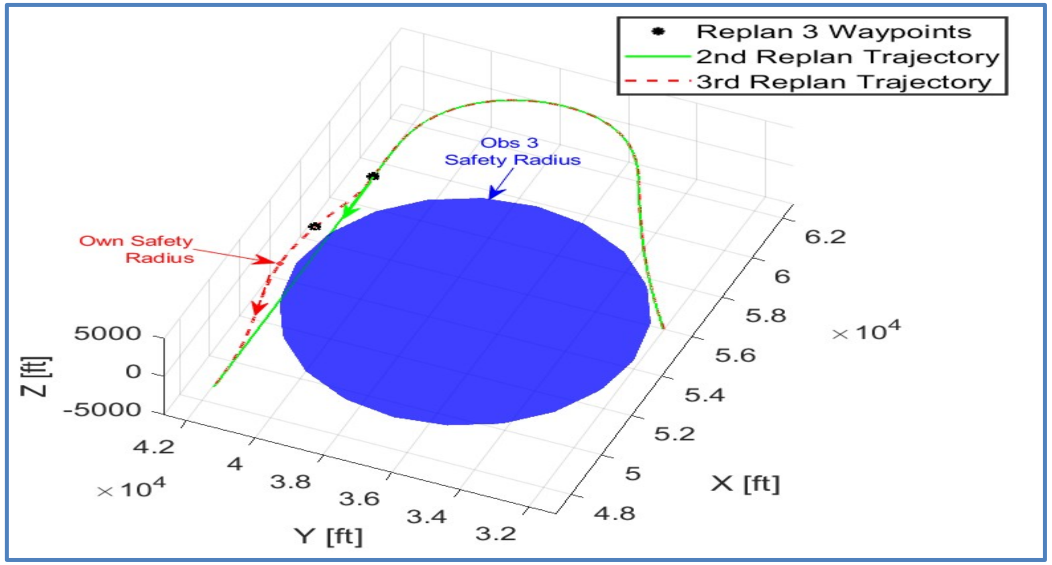
Experimental Result











Conclusion

Main Idea: Layers of algorithms, mathematical guarantees when possible and exhaustive testing when it is not.

Each of the subcomponents for COBRA DDP was independently tested and developed but complex testing environments revealed each method’s limitations.

Individual algorithms may not be general enough to capture the full challenge of autonomy problems.

Together, they can attempt to address it. The parametric trajectory structure provides the “language” that allows each algorithm interoperate to without losing time scale or dynamic complexity information.

Challenge 1
Each algorithm and system needs to be evaluated with respect to all subcomponents and the overall system.

Challenge 2
How do we map *operational considerations* into the *mathematical frameworks* where our solutions can be developed?

Next Steps

1. Finding “languages” like parametric trajectories that allow the natural integration of different algorithms into a single system.
2. Finding metrics for higher-level/abstract planning as a first step towards encoding general operational considerations into a mathematical framework.
3. Creating Benchmark problems to identify and separate systems based on both subcomponent and system level performance.
4. Increasing simulation complexity to improve testing capabilities.

Challenge 1
Each algorithm and system needs to be evaluated with respect to all subcomponents and the overall system.

Challenge 2
How do we map *operational considerations* into the *mathematical frameworks* where our solutions can be developed?

Government, Industry and Academia Barriers to Collaboration:

- **Access to real-world (relevant) transition vehicles (hover/transition/cruise):**
  - Industry has best access to relevant vehicles, but major issues of proprietary/competition inhibit collaboration
  - Academia has easiest access to small flight vehicles (e.g., quadrotors) but limited access to transition class vehicles. Advantages include vast pool of graduate students and relative ease in flight operations
  - Government has the freedom to tackle the hard research problems (not driven by publishing or bringing product to market) but very limited access to vehicles and small research staffing
- **Sharing of algorithms and data:**
  - Large overhead in reproducing, modifying, and/or sharing algorithms from user-generated code/simulations (need for common simulation framework, data buses, etc.)
  - Proprietary, releasability, and data rights barriers
- **Research Staff Availability:**
  - Research field is enormous, and no-one has staffing to address host of issues: e.g., aero-propulsive modeling, robust control, perception, trajectory management, emergency response, weather and traffic avoidance

- **Intelligent Contingency Management Team's Response:**
  - Generalized Urban Air Mobility (GUAM) v1.0 Simulation: open-source, high-fidelity, 6-DOF non-linear transition vehicle simulation. Facilitates collaboration with no issues with proprietary data and fully publishable results
  - Challenge Problems: Formulate and release major challenge problems (e.g., AIAA SciTech) to researchers
    - E.g., effector failure detection/mitigation, collision avoidance, trajectory replanning, and pattern entry
    - GUAM framework enables direct code sharing collaboration, but at least facilitates direct comparison

No one group can tackle this problem alone!

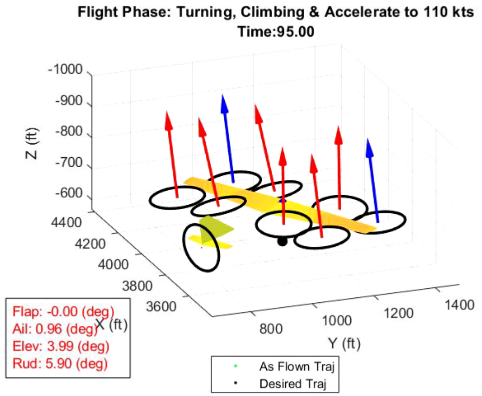
Why is Autonomy for Transition Vehicles so Hard?

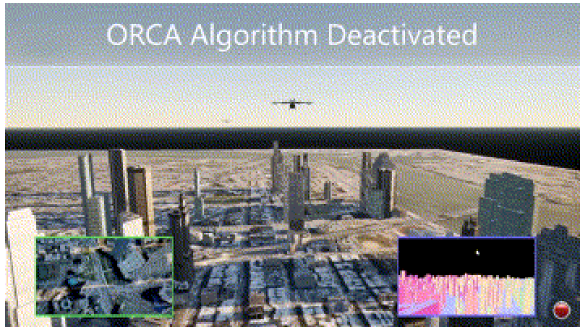
Vehicle challenges:

- Novel configurations **strain** capability for aero-propulsive modeling
- Electrical propulsion systems drive **lower** vehicle performance **margins**
- “**Transition**” vehicle operations: hover, transition and cruise
- Constant **low** altitude operations (most dangerous portion of flight)

Autonomy challenges:

- **Dynamic**, unstructured, **3-D** environment compared to autonomous cars
- Autonomous **perception** (all-weather)
- **Severity** of **mishaps** necessitates requirement for high safety assurance
- Low profit margin missions, **preclude** long and expensive vehicle development cycles





Generic Urban Air Mobility (GUAM) Simulation

- 6-DOF, non-linear, rigid body simulation framework for Urban Air Mobility vehicles
- Goal: to facilitate research exchange/comparison and foster collaboration between researchers of UAM class vehicles with focus on autonomous flight and operations
- Initial GUAM release includes a NASA Lift+Cruise vehicle configuration including:
  - Two aero-propulsive models: strip theory and polynomial models
  - Baseline unified (across hover/transition/cruise) iLQR controller
  - Effector models including failure methods
  - Matlab autocode capable and compatible with Python, Unreal Engine, ROS II, etc..
- Subsequent release is ML friendly with Cognitive architecture focus
- Our research focus is: Intelligent Contingency Management and we are releasing a series of **Challenge Problems** this year (e.g., collision avoidance, effector failure detection and recovery…)

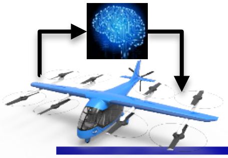
Approved for open-source release:

Available soon at

https://code.nasa.gov

https://nari.arc.nasa.gov/ttt-ram/community

https://github.com/nasa



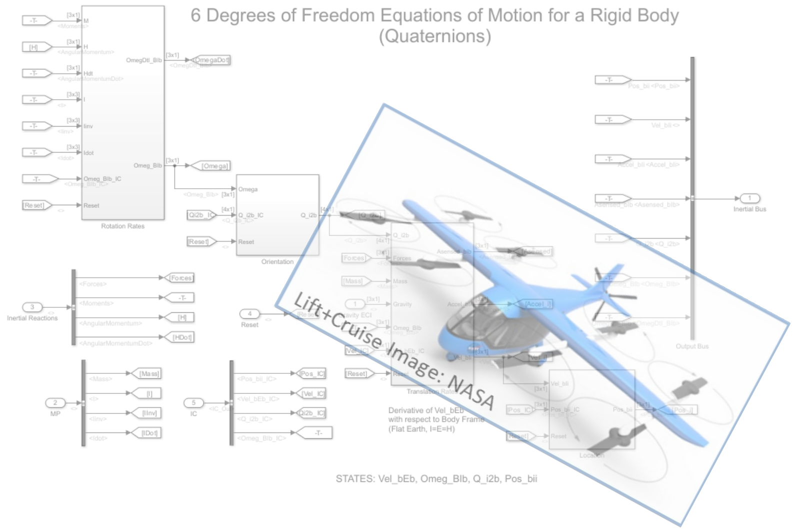
Generalized Urban Air Mobility (GUAM)

What is it?

- Open source, 6-DOF, non-linear, rigid body simulation framework for Urban Air Mobility (UAM) vehicles
- Modular simulation architecture facilities easy implementation of vehicle model, controller, sensor (perception), autonomy algorithms

What is in the initial release?

- Single NASA Lift+Cruise vehicle configuration
- Two aero-propulsive models: strip theory and polynomial models
- Baseline unified (across hover/transition/cruise) iLQR controller
- Aero/propulsor effector and failure models
- Matlab-autocode capable
- Compact, continuous trajectories using piece-wise Berstein polynomials



Why is NASA releasing it?

- Foster collaboration to address wide array of research challenges for UAM autonomy
- Provide common simulation architecture for collaboration and performance comparison within the autonomy UAM research community
- Address barriers of proprietary information, limited research staffs, and limited access to medium/high fidelity UAM models

Challenge Problems:

- Provide code & data that formulates large research challenges for the autonomous UAM community
- Examples: collision avoidance, effector failure (control and flight envelope estimation), traffic pattern entry, trajectory replanning, etc.

GUAM Community:

- GIT repo and wiki: GUAM git
- NASA Code Repo: Code
- Challenge Problems: GUAM Challenge Problems

What is in the next release?

- Flight visualization and large visualization data sets
- Artificial Intelligence/Machine Learning friendly
- Cognitive (decision making) architecture
- Python and Robot Operating System (ROS) capable

Perception

Aero-propulsive Modeling

Transition vehicle control

Cooperative & uncooperative collision avoidance

Flight envelope estimation

Control allocation

Dynamically feasible trajectory

Trajectory optimization

Cognitive architecture

Human machine interface

Safety guarantees