Path Planning: Differential Dynamic Programming and Model Predictive Path Integral Control on VTOL Aircraft

Matthew D. Houghton, Alex Oshin, Michael J. Acheson and Irene Gregory

NASA Langley Research Center

Evangelos A.Theodorou

Georgia Institute of Technology

AIAA SciTech Forum and Exposition 2022, San Diego, CA

Outline

- Motivation: Challenge of trajectory planning for urban air mobility (UAM) vehicles
- Trajectory Planning
  - Problem Statement
  - Differential Dynamic Programming (DDP) theory
  - Model Predictive Path Integral Control (MPPI) theory
- Experiments:
  - Sample Results with interesting observations
- Conclusions and Ongoing Work

Lift + Cruise

First application of planning algorithms, used extensively in robotics, to UAM class vehicles

Motivation: Challenge of Urban Air Mobility VTOL vehicles

- Future urban air mobility (UAM) necessitates high levels of automation (due to cost, airspace congestion, pilot experience, novel aircraft etc.)

- Trajectory replanning is required capability to enable UAM (obstacles, traffic, aircraft contingencies, weather, change of mission etc.)

- Trajectory replanning for UAM class vehicles extremely challenging
  - Highly complex nonlinear system: large number of propulsors, novel configurations, aero-propulsive interactions
  - Three phases of flight: hover, transition, and cruise
  - Dynamic complexity suggests large model uncertainty
  - Overactuated vehicles: multiple control solutions
  - Example aircraft (Lift + Cruise): 8 lifting rotors, 4 aero surfaces, 1 pusher propeller

Optimal Trajectory Problem

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

State Vector

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

Discrete system nonlinear dynamics

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

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

Value Function (cost-to-go)

Truncated Control Sequence

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

Bellman's Principle of Optimality: find overall optimal control as sequence minimization for each truncated control sequence backwards in time

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

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
      - We perform line search on α to heuristically find the best learning rate

Model Predictive Path Integral Control

Stochastic sampling algorithm to generate optimal trajectories

- Iterative algorithm: equivalent to Hamilton-Jacobi-Bellman equation
- Generalized (nonlinear) cost criteria (not just quadratic)
- Parallel processing (GPU) of sampled (noisy) controls and propagates all trajectories simultaneously
- Parallel computation enables real-time trajectory planning
- Control law update exponentially weighted average of sampled controls

Model Predictive Path Integral (cont)

Given

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

Nonlinear Dynamics

$$\mathbf{v}_t \sim \mathcal{N}(\mathbf{u}_t, \Sigma)$$
$$\Sigma \in \mathbb{R}^{m \times m}$$

Gaussian Controls Distribution

Cost Formulation

$$\mathcal{L}(\mathbf{x}_t, \mathbf{u}_t) := q(\mathbf{x}_t) + \frac{\lambda}{2}\mathbf{u}_t^{\top}\Sigma^{-1}\mathbf{u}_t$$

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

Parallel Computation: Dynamics and Cost

$$\mathbf{V}^{(k)} := \{\mathbf{v}_1^{(k)}, \ldots, \mathbf{v}_{T-1}^{(k)}\}$$

$$\mathcal{J}^{(k)} = \mathcal{J}(\mathbf{V}^{(k)})$$

Normalize

$$\eta = \sum_{k=1}^{K} \exp(-\lambda^{-1}(J^{(k)} - \rho))$$

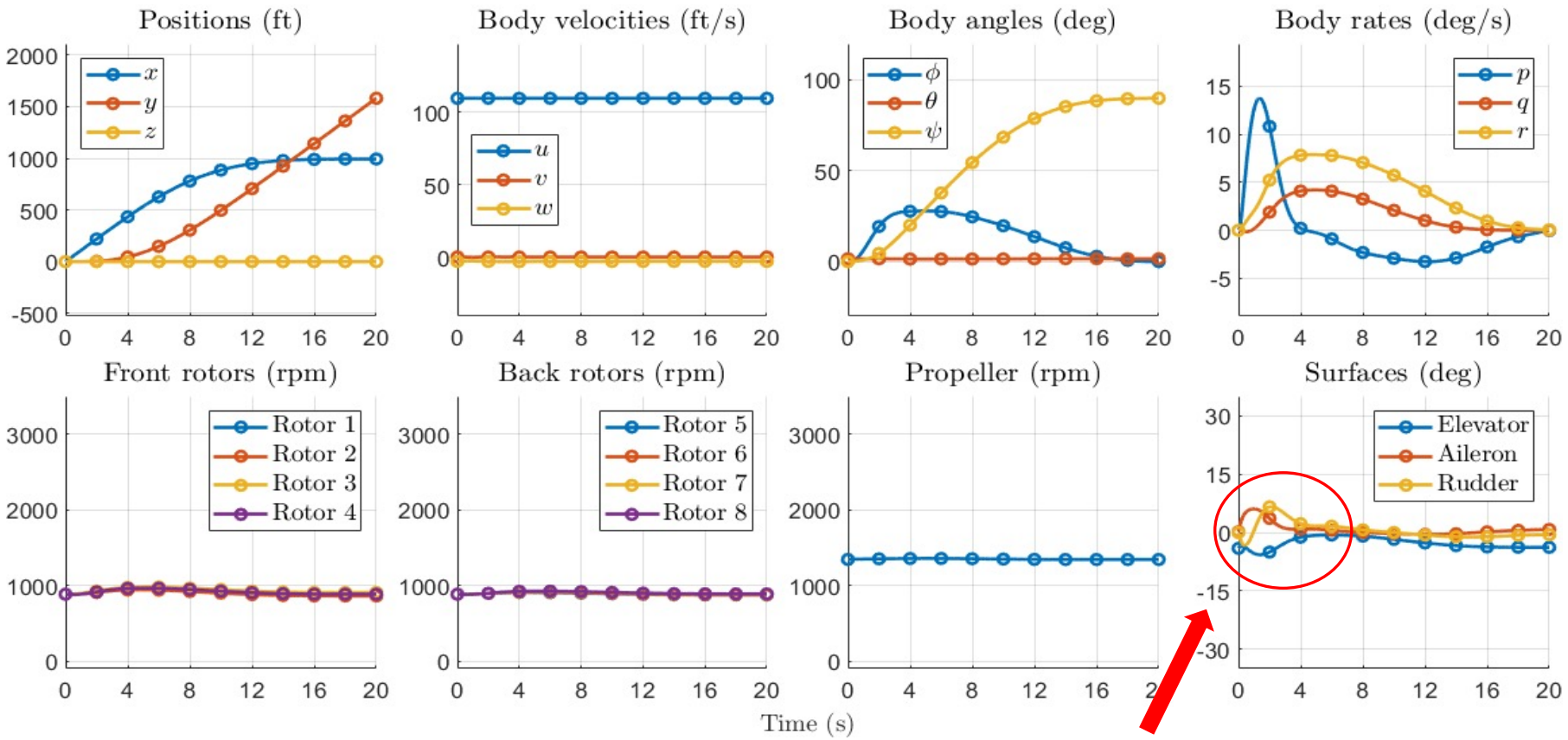
Compute Weights

$$w^{(k)} = \frac{1}{\eta}\exp(-\lambda^{-1}(J^{(k)} - \rho))$$

Compute Optimal Controls

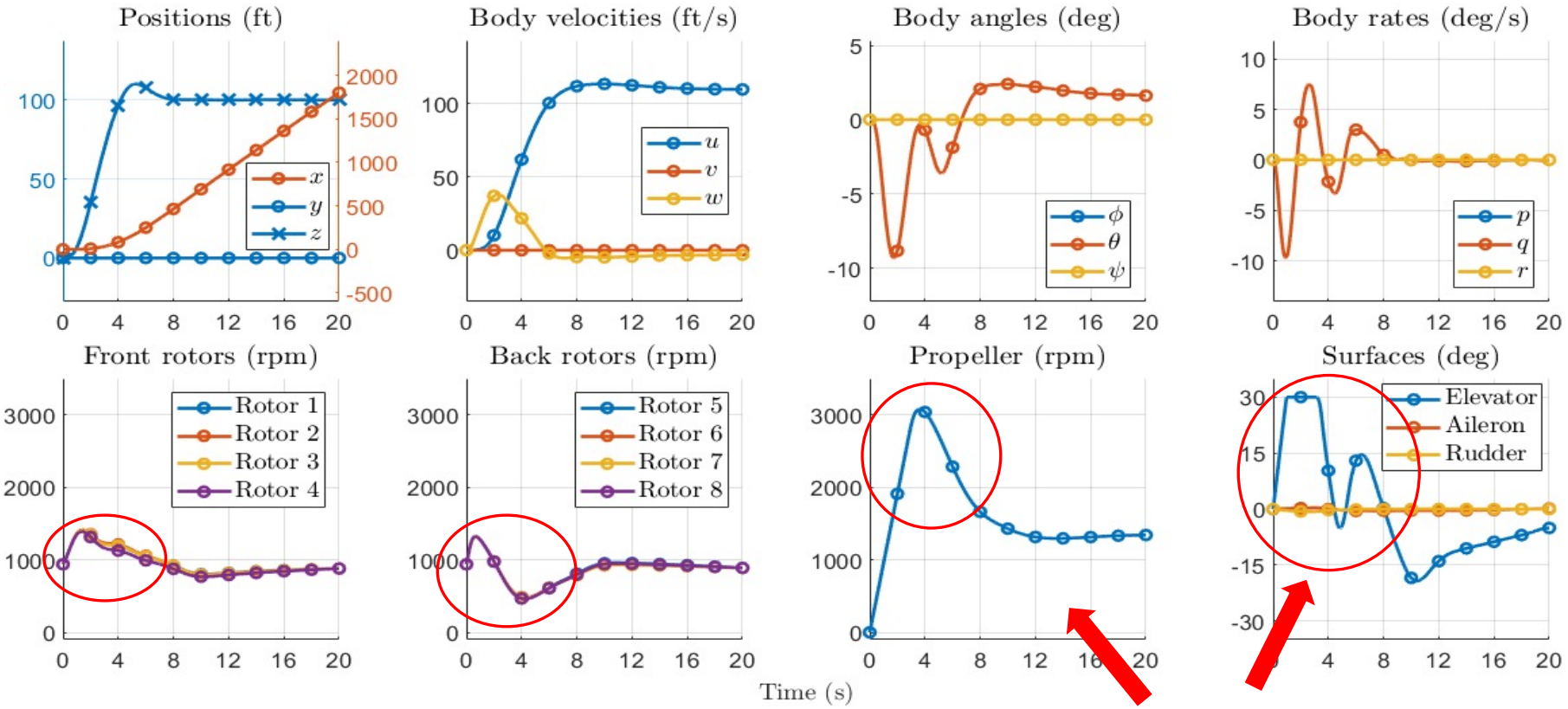
$$\mathbf{u}_t^* = \sum_{k=1}^{K} w^{(k)}\mathbf{v}_t^{(k)}$$

Case Studies: DDP 90° Turn



DDP turn using typical aero surfaces / not propulsors

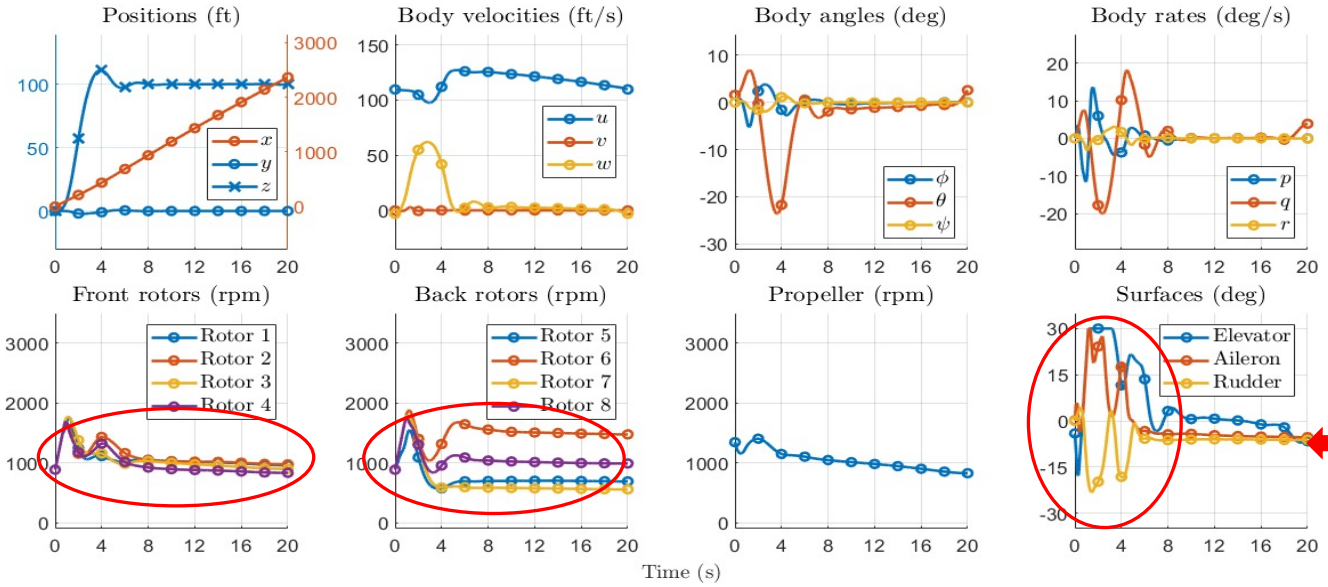
Case Studies: DDP Hover - Transition



Interesting use of differential rotors, pusher and elevator

Case Studies: DDP Transition Altitude Change

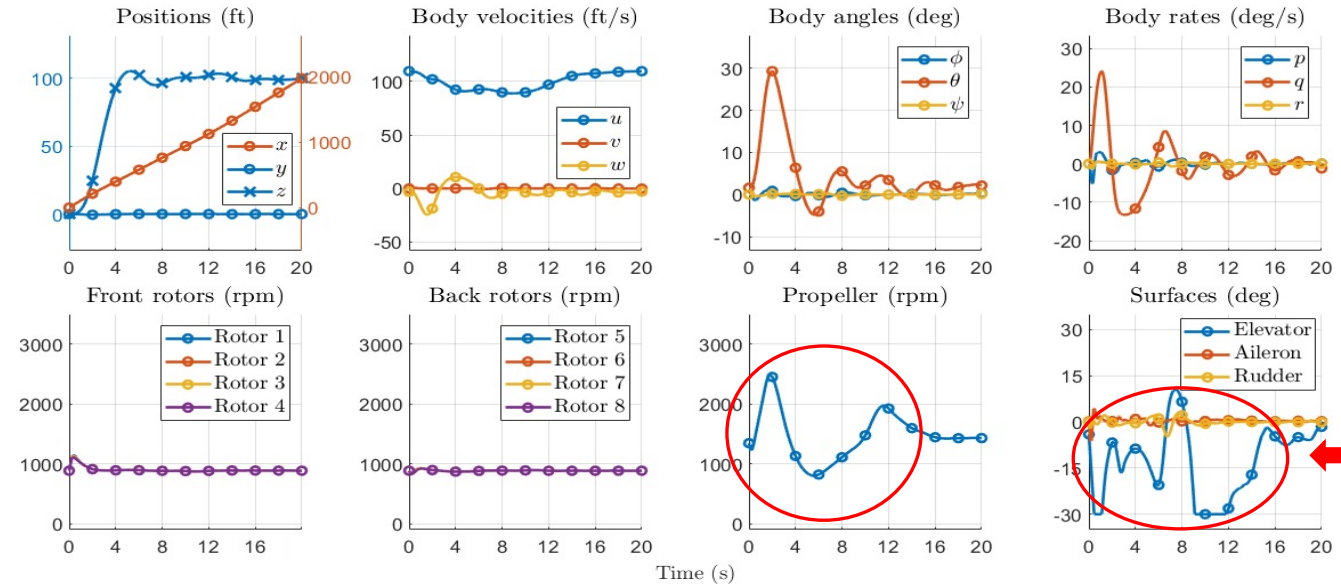
Using Rotors to Ascend



Effects of Overactuated Vehicle

Novel climb: differential rotor and all aero effectors

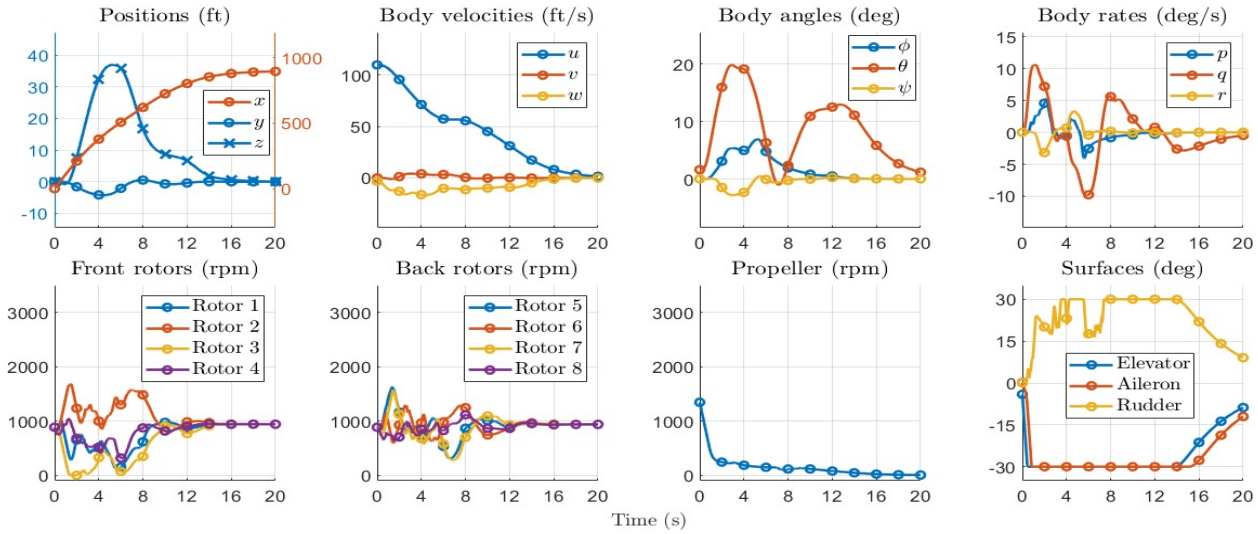
Without Rotors to Ascend



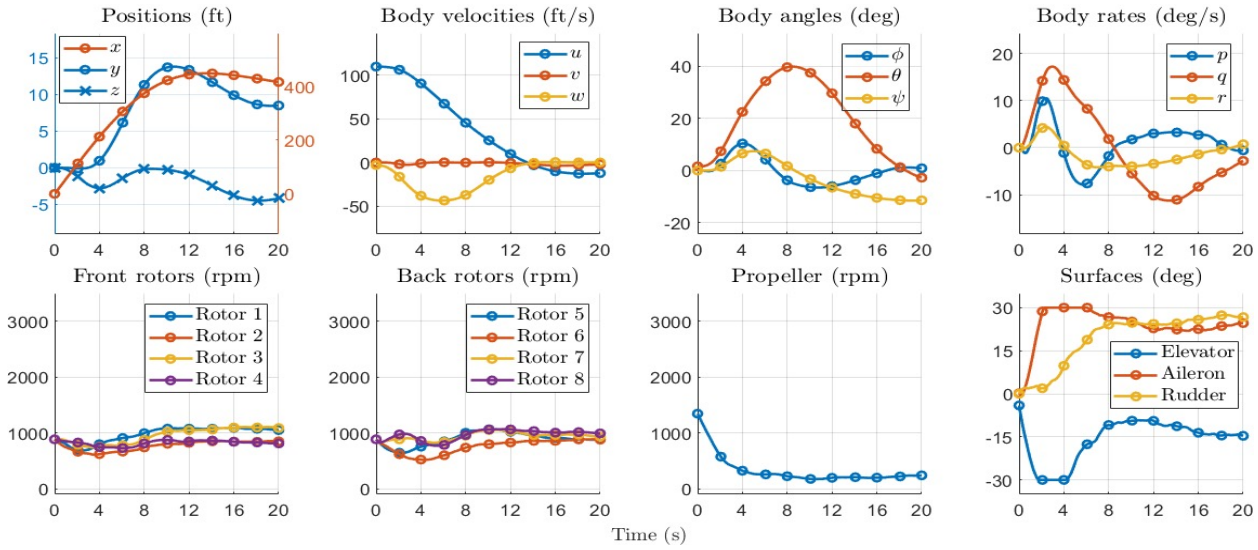
Classic climb: elevator and pusher

Case Studies: Cruise to Hover: DDP & MPPI

DDP



MPPI

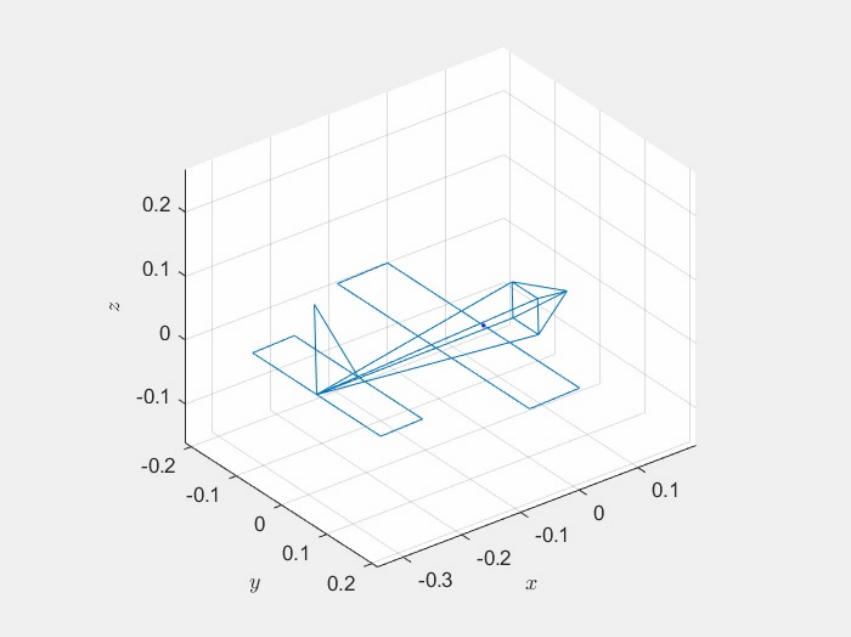


Differences in cost/weightings

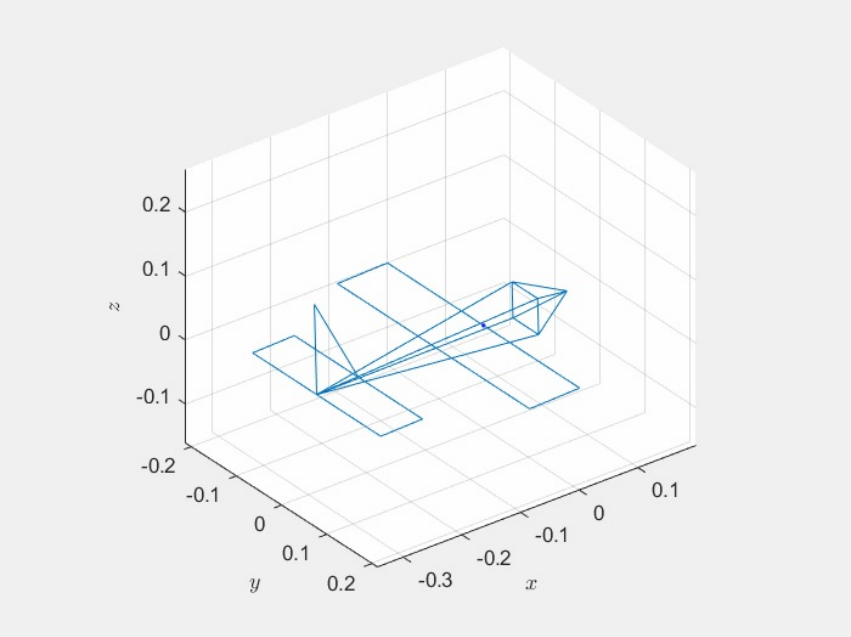
Different use of rotor/propeller & elevator combinations

Case Studies: Cruise to Hover: DDP & MPPI

DDP



MPPI



Conclusions and Ongoing Work

- **Conclusions**
  - First application of planning algorithms, used extensively in robotics, to UAM class vehicles
  - Both DDP and MPPI demonstrated capability to trajectory plan for UAM class vehicles across all flight phases
  - Balancing required between computation time and planning horizon

- **Ongoing / Future Work**
  - Determine minimum vehicle model required to plan (e.g., fewer states, split into longitudinal/lateral models etc..)
  - New parameterized DDP theory developed (paper pending)
  - Enhance research with state constraints and other improvements and additions to these algorithms → allows specification of different mission objectives in trajectory planning and replanning

Questions?

Contact Information:
matthew.d.houghton@nasa.gov