SAND2020-4107PE

Predictive Guidance and Control for Robotic Obstacle Avoidance and Hypersonic Trajectory Generation

Isaac Kassim

Department of Mechanical and Aerospace Engineering

New Mexico State University

Thesis Committee

Assistant Professor Hyeongjun Park (Chair)

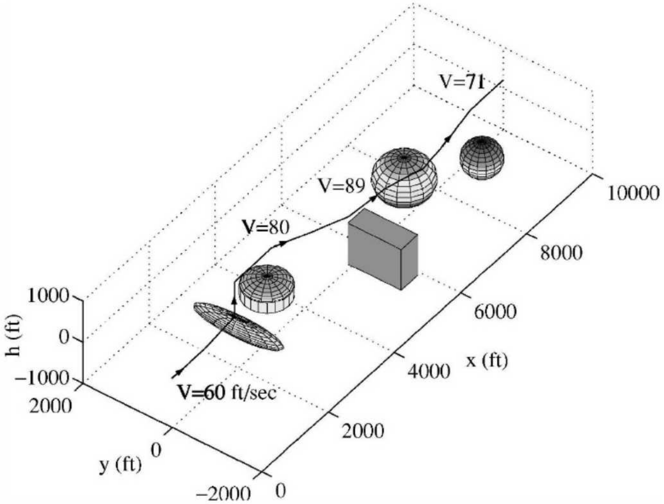
Assistant Professor Liang Sun

Professor James McAteer (Dean's Representative)

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

Overview

https://www.popularmechanics.com/technology/robots/a27542297/meet-the-new-flying-robots-that-nasa-is-sending-into-space/



[Yang et al. 2004]

https://www.nbcnews.com/mach/science/hypersonic-airliner-would-take-you-los-angeles-tokyo-under-two-ncna1045986

Content Outline

- Introduction
  - Motivation and Background
  - NMPC Framework
  - Research Contributions

- NMPC based robotic obstacle avoidance
  - Astrobee free-flyer system architecture and purpose
  - Obstacle avoidance case studies and results

- NMPC based hypersonic trajectory generation
  - HFV system architectures
  - 2D and 3D model formulations
  - Trajectory generation case studies and results

- Conclusions and future work

Introduction

Model Predictive Control Background

- Model Predictive Control (MPC)
  - Effective feedback control for constrained control problems with multi-variable systems
  - Various applications [Camacho 2007; Qin & Badgwell 2003]
    - Petrochemical, Power Management, Industrial Process Control
- Linear MPC
  - Widely applied & mature technology [Mayne et al. 2000; Allgöwer et al. 2004]
- Nonlinear Model Predictive Control (NMPC)
  - Used for full fidelity nonlinear system dynamics and constraints integration into MPC framework
  - Main challenge: computational complexity for real-time implementation

Literature Review

- Methodologies to simplify real-time computations
  - Model order reduction/ simplification techniques [Choroszucha 2015; Hovland et al. 2007; Dufour et al. 2003]
    - Focusing on dominant dynamics / linear approximation
    - Degraded closed-loop performance with significant nonlinearities
  - Explicit MPC [Bemporad et al. 2002; Johansen 2004; Tøndel 2003]
    - Explicit solutions are obtained offline and stored locally
    - Large local memory may be required
    - If problem conditions change, the solution can not adapt
  - Numerical methods and algorithms [Diehl et al. 2009; Cannon 2004]
    - Real-time optimization solvers
    - Advantageous for
      - Nonlinear systems
      - Systems with changing parameters

Interior Point Optimization

- Interior Point (IP) Optimization
  - Effective method for solving nonlinear constrained optimization problems [Wang & Boyd 2008; Ding et al. 2016; Klintberg & Gros 2017]
    - Directly integrates equality and inequality constraints into objective function optimization
  - Widely available in solvers in commercial and open source spaces
    - MATLAB
    - COIN-OR IPOPT (Interior Point OPTimizer) [Wächter & Bigeler 2005]
  - IP optimization will be used in all NMPC implementations discussed in this work

Nonlinear MPC Framework

- Constrained optimal control problem over a receding horizon

$$
\begin{aligned}
&\min_{u(\cdot)} \; J(x(\cdot),u(\cdot)),\\
&\text{where } J(x(\cdot),u(\cdot)) = \Phi(x(t+N)) + \sum_{k=t}^{t+N-1} L(x(k),u(k)),\\
&\text{subject to}\\
&x(k+1) = f(x(k),u(k)), \quad f : \mathrm{R}^{n+m} \to \mathrm{R}^{n},\\
&x(t) = x_t, \quad x_t \in \mathrm{R}^{n},\\
&C(x(k),u(k)) \le 0, \quad C : \mathrm{R}^{n+m} \to \mathrm{R}^{l},\; k = t,\cdots,t+N-1,\\
&\bar{C}(x(k))) \le 0, \quad \bar{C} : \mathrm{R}^{n} \to \mathrm{R}^{q},\; k = t,\cdots,t+N.
\end{aligned}
\qquad (1)
$$

  - Minimize the cost function over a prediction horizon
  - Obtain an optimal input sequence given an initial state
  - Implement the first element of the input sequence

Research Contributions

NMPC controller based on IP optimization for real-time robotics and flight vehicle control

Robotic obstacle avoidance

- **Astrobee Free-Flyer**
  - Flight in 3D constrained by hardware and environment
- **Case Studies**
  - Static obstacle avoidance cases w/ nonlinear keep-out zones

Hypersonic trajectory generation

- **Hypersonic Flight Vehicles**
  - 2D and 3D models constrained by hardware and mission requirements
  - Highly nonlinear system dynamics w/ multi-variable dependencies
- **Case Studies**
  - Trajectory generation considering No-Fly-Zone avoidance in tandem with waypoint following

Robotic Obstacle Avoidance

This research was funded by the New Mexico Space Grant Consortium.
New Mexico Space Grant Consortium (NMSGC) is a statewide collaboration with industry, government and academic partners. New Mexico State University (NMSU) is the lead Land Grant and Space Grant University in the state of New Mexico. NMSGC is a member of the congressionally funded National Space Grant College and Fellowship Program that has been administered by the National Aeronautics and Space Administration (NASA) since 1989.

Introduction to Free Flyers

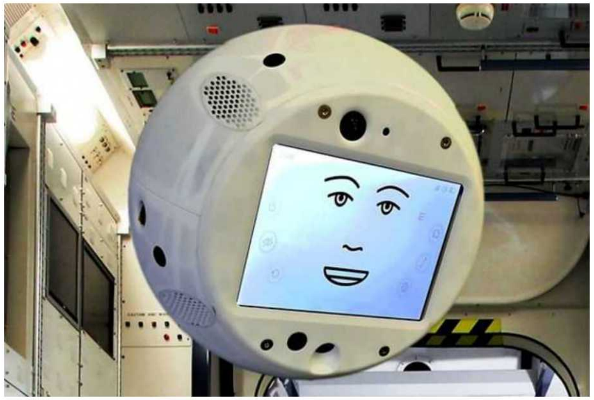
Free Flyers

- Robots flying with 6-DOF on ISS
- Manual or Semi-Autonomous Control

Uses

- Repetitive/simple environmental monitoring
- Remote agent to assist astronauts
- Guest science

https://www.electronicsweekly.com/news/research-news/inertia-steers-int-ball-drone-international-space-station-2017-07/



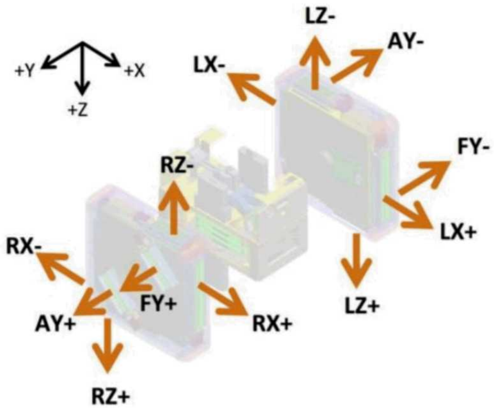
https://www.bbc.com/news/technology-44655675

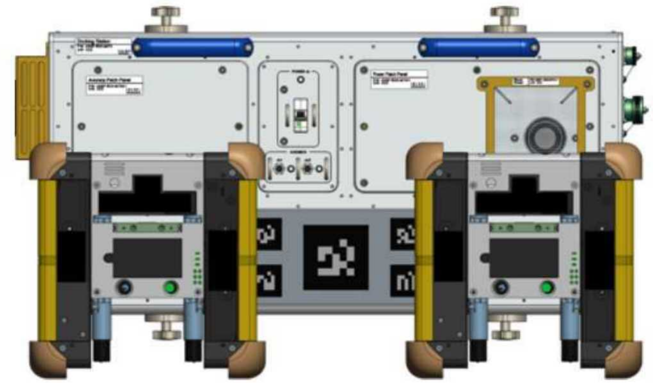
https://www.nasa.gov/astrobee/

Astrobee System Architecture

Physical Hardware

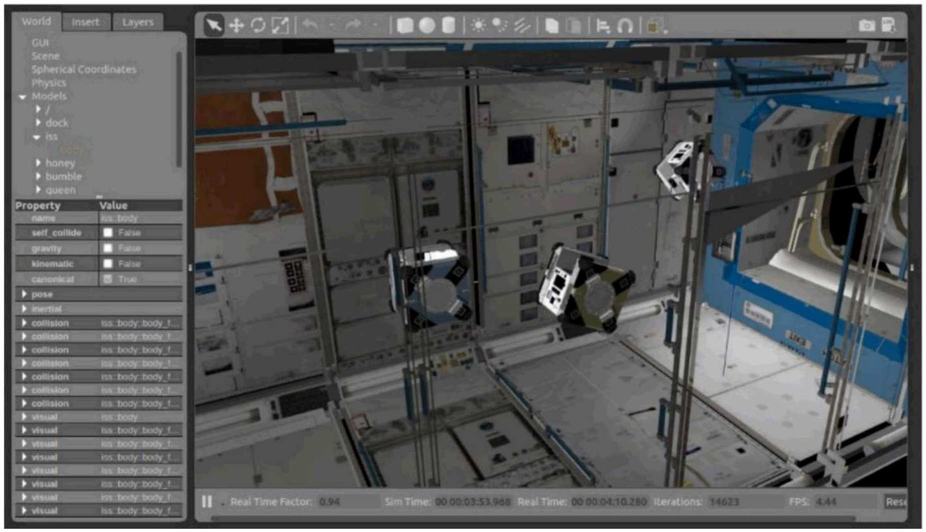
- Electric impeller propulsion
- Modular structure





Electronic Hardware

- ARM processor and Android OS
- Multiple cameras/sensors



Software

- Programming with ROS
- Hardware simulation with Simulink
- Operation visualization with Gazebo

Research Objectives (1/2)

Motivation

1. Astronaut and ISS structural safety requires collision avoidance during Free-Flyer maneuvering to guarantee

2. Minimize maneuvering time while maximizing battery life

→ Goal: Autonomous real-time obstacle avoidance

Research Objectives (2/2)

Approach

1. Create G&C algorithm based on NMPC framework

2. Simulate realistic scenarios with the G&C algorithm (avoidance of static obstacles wrt. ISS spatial limitations)

3. Verify and validate the G&C algorithm with the Gazebo/ROS simulator

NMPC Framework Design

NMPC problem formulation

Minimize cost function J

where
$$J(x(\cdot),u(\cdot)) = (x(t+N)-x_f)^T P(x(t+N)-x_f) + \sum_{k=t}^{t+N-1} (x(k)-x_f)^T Q(x(k)-x_f) + u(k)^T R u(k),$$

Terminal Cost Weight (Final Positing & Velocity)

Incremental Cost Weight for Position & Velocity

Incremental Cost Weight for Control Input (Battery Use)

Constraints

1. Astrobee's equations of motion
2. Propulsion system constraints (maximum thrust)
3. Maximum safe velocity
4. ISS corridor area
5. Collision avoidance from obstacle keep out zones

Constraints (1/2)

1. Astrobee's equations of motion (translational motion only)

$$\ddot{x} = \frac{F_x}{m},\ \ddot{y} = \frac{F_y}{m},\ \ddot{z} = \frac{F_z}{m}$$

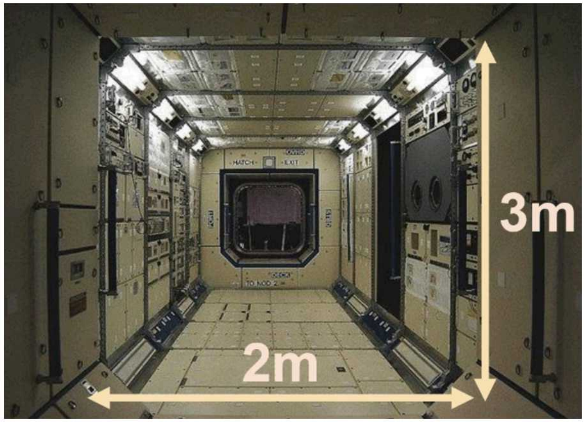
2. Propulsion system constraints (maximum thrust)

$$-0.6 \le F_i \le 0.6\ \ [N]$$

3. Maximum velocity $-0.5 \le V_i \le 0.5\ \ [m/s]$

4. 2 m x 3 m x 2 m area

$$0 \le y \le 2\ [m]$$
$$0 \le z \le 3\ [m]$$



Constraints (2/2)

5. Collision avoidance constraints with keep-out zone

Astrobee: Sphere approximation

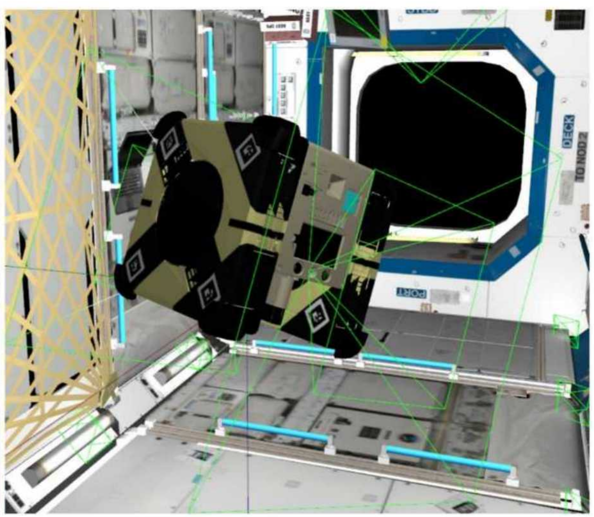
$$(x - x_o)^2 + (y - y_o)^2 + (z - z_o)^2 \geq r^2$$

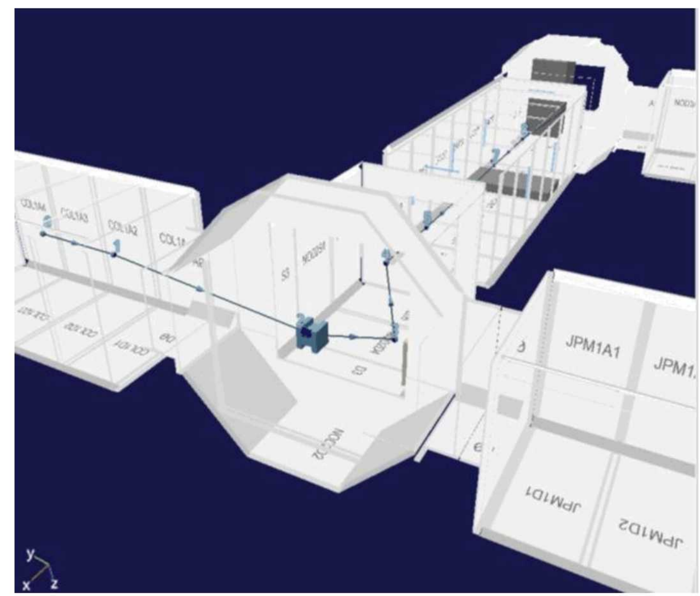
Astronaut: Cylinder approximation

$$(x - x_o)^2 + (y - y_o)^2 \geq r^2$$

Case Studies

- Cases for obstacle avoidance demonstration
  - Case 1: Static Astrobee avoidance
  - Case 2: Static astronaut avoidance





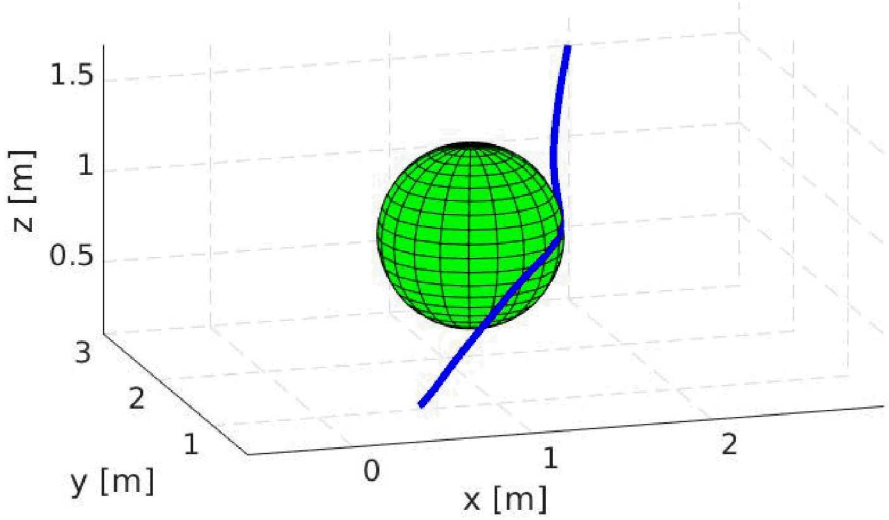
[Fluckiger et al. 2019

Case 1 – Static Astrobee (1/2)

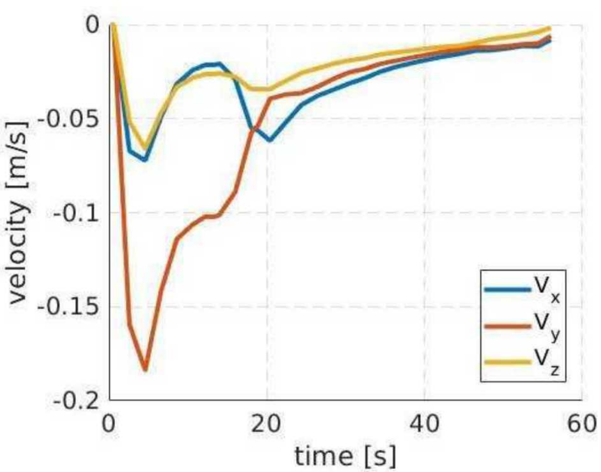
Case 1 problem set-up

Parameter	Value
Initial Position	(2, 3, 1.5) m
Desired Position	(0.3, 0.3, 0.3) m
Other Astrobee's Position	(1, 1.5, 0.8) m
Collision Avoidance Keep-out Radius	0.5 m
Sample Time	2 s
Prediction Horizon	40

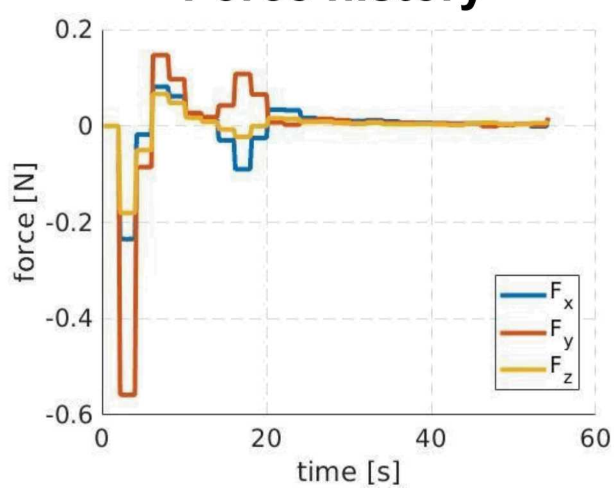
Trajectory results



Velocity history

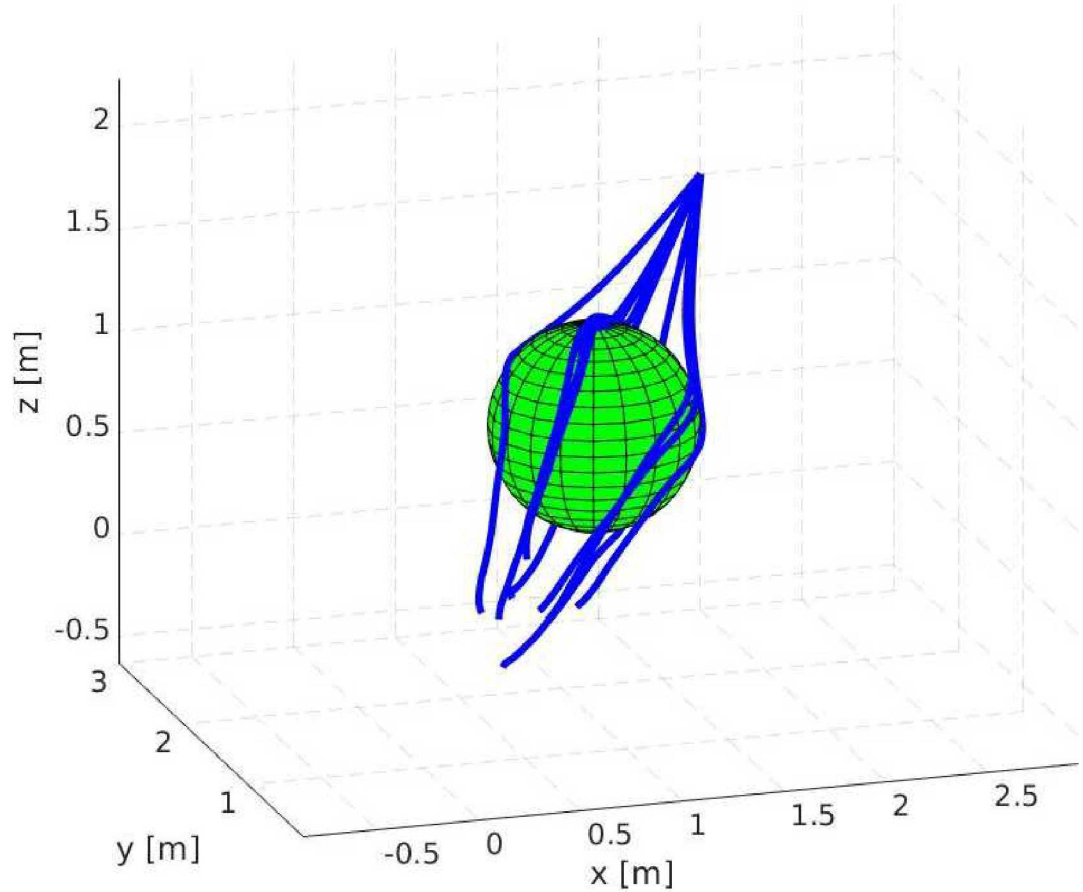


Force history



Case 1 - Static Astrobee (2/2)

Multiple trajectories for different position goals

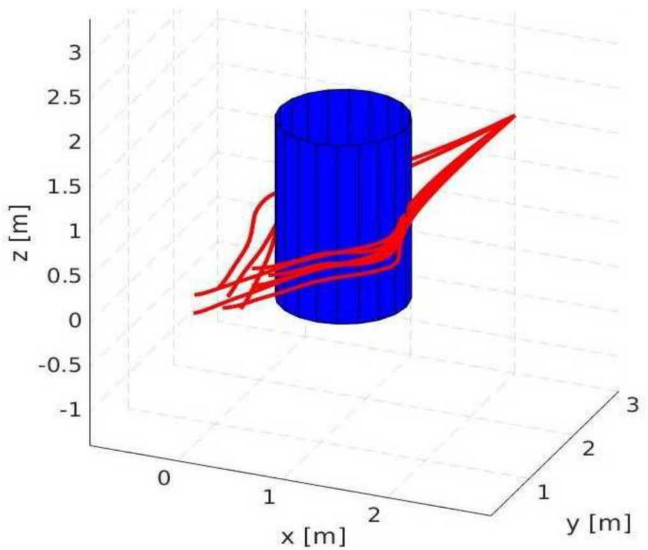


Case 2 – Static Astronaut

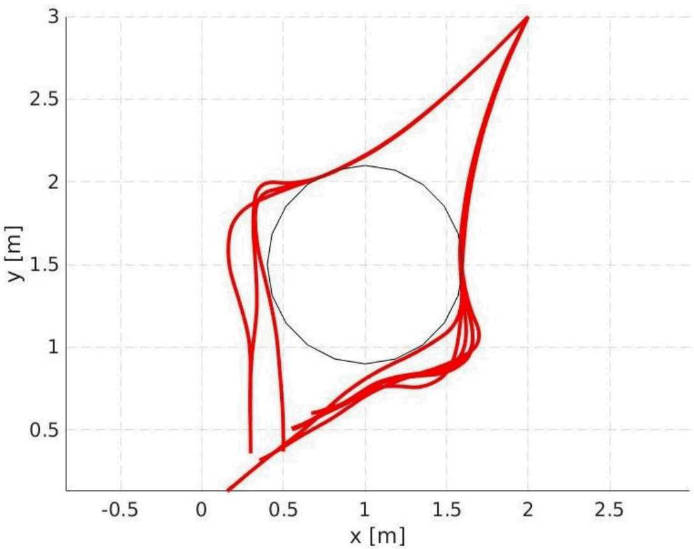
Case 2 problem set-up

Parameter	Value
Initial Position	(2, 3, 1.5) m
Desired Position	(0.3, 0.3, 0.3) m
Other Astrobee's Position	(1, 1.5, 0.8) m
Collision Avoidance Keep-out Radius	0.5 m
Sample Time	2 s
Prediction Horizon	40

Isometric trajectory



Top-down trajectory

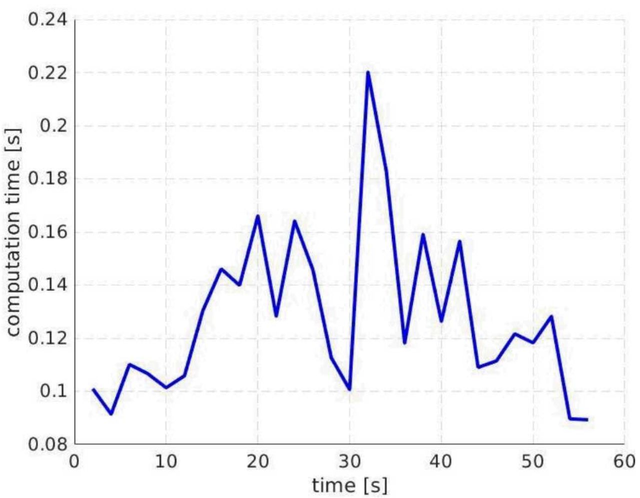


Real-time Implementability

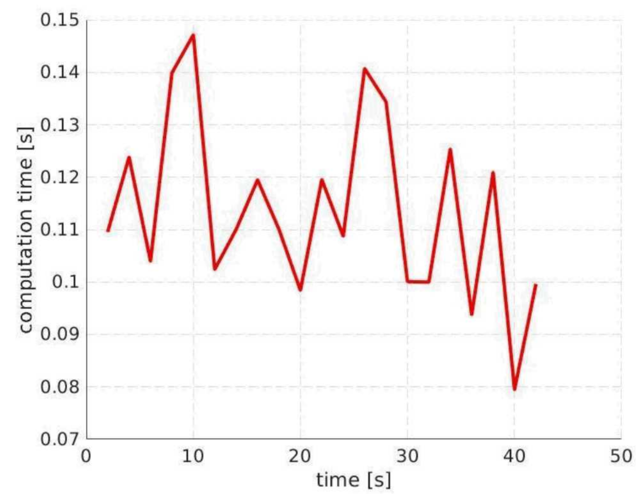
IPOPT Computation Time Comparison

	Case 1	Case 2	Sample Time
Avg / Max Comp. Time [s]	0.13 / 0.22	0.11 / 0.15	2

Case 1 Computation Time History



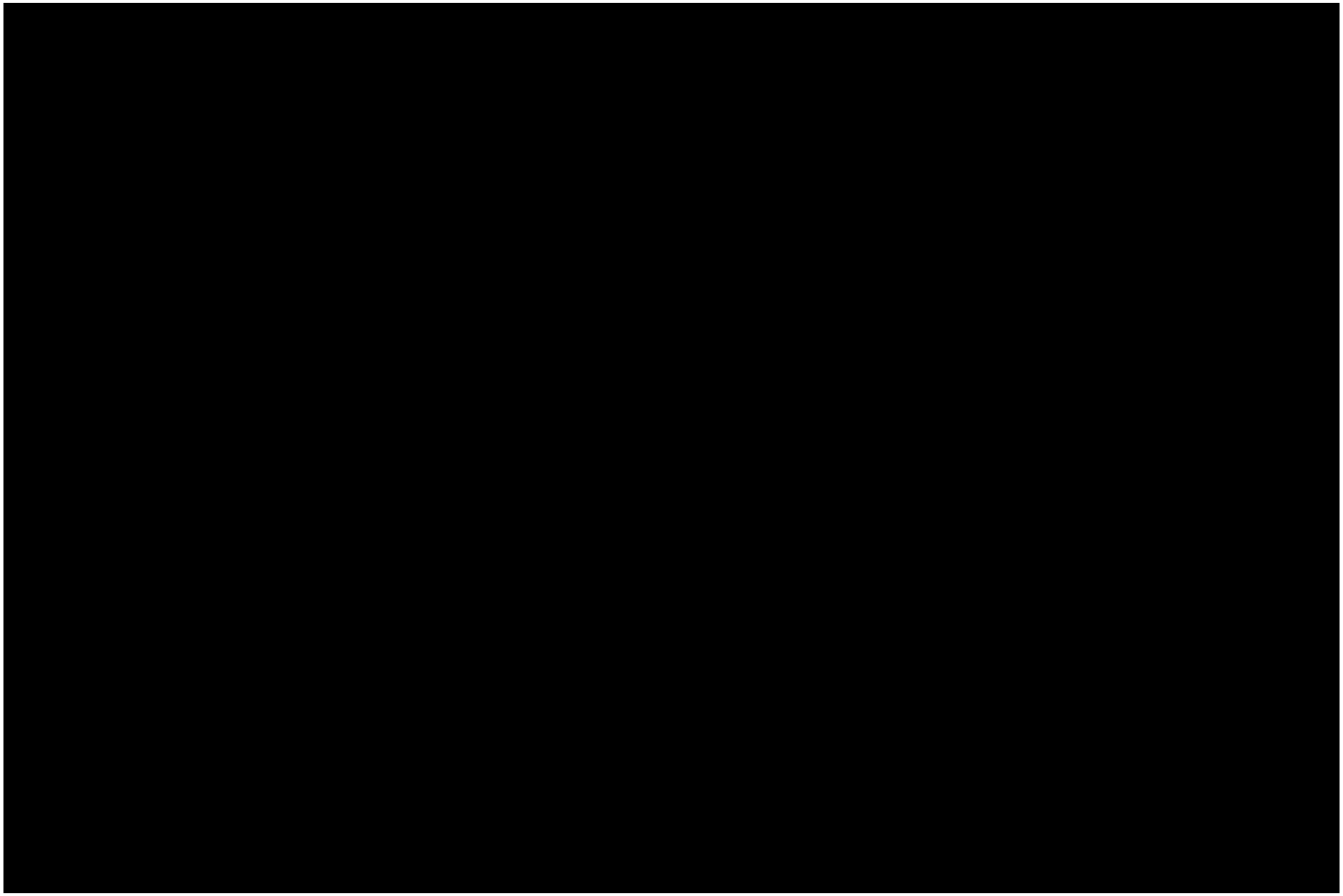
Case 2 Computation Time History



– Simulations on a laptop with Intel(R) CPU @ 1.60 GHz

ROS / Gazebo Implementation (1/2)

Kinematic Model Visualization



ROS / Gazebo Implementation (2/2)

Dynamic Model Visualization

Robotic Obstacle Avoidance - Summary

Completed Work

- Static obstacle avoidance with NMPC controller
- Integration obstacle avoidance results into ROS/Gazebo visualization framework

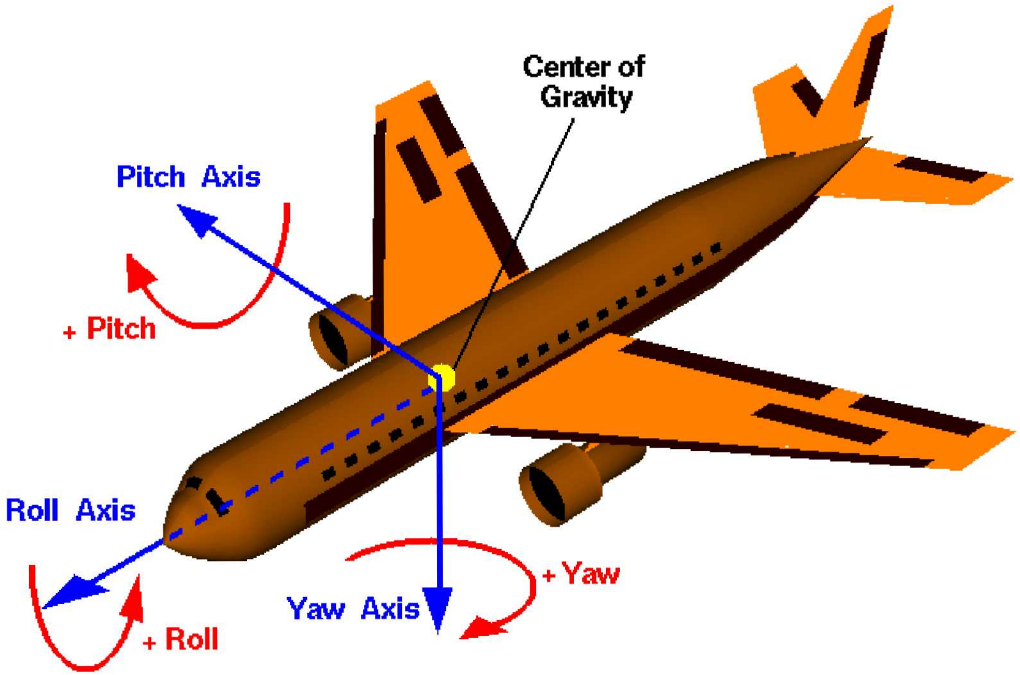
Future work

- Moving obstacle / Multiple Obstacle cases both offline and implemented into ROS/Gazebo
- Validating controller design using NASA Ames test facilities

Hypersonic Trajectory Generation

This research was funded by the Sandia National Laboratories Laboratory-Directed Research and Development Program. Sandia National Laboratories is a multi-mission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525

Aircraft Principal Axes



NASA Glenn Research Center
https://www.grc.nasa.gov/WWW/K-12/airplane/rotations.html

Hypersonic Flight Vehicle Overview

- Hypersonic Glide Vehicles (HGVs) are reentry vehicles with dynamic flight capabilities but no on-board propulsion
- Hypersonic Cruise Vehicles (HCVs) hypersonic flight platforms with on-board propulsion
  - Fully atmosphere based booster platforms or suborbital reentry vehicles

Aircraft vehicle dynamics models

Model	States	Controls	EOM	Constraints
2 DOF navigation+ 1 DOF point mass	x, y, ε	$\dot{\varepsilon}$	$\dot{x} = V_o \sin(\varepsilon)$ $\dot{y} = V_o \cos(\varepsilon)$ $\varepsilon = \int \dot{\varepsilon}\, dt$	$\dot{\varepsilon} \le \frac{V_o}{R_{\min}}$
3 DOF navigation + 1 DOF point mass	x, y, z, ε	$\dot{\varepsilon}, \gamma$	$\dot{x} = V_o \sin(\varepsilon)\cos(\gamma)$ $\dot{y} = V_o \cos(\varepsilon)\cos(\gamma)$ $\dot{z} = -V_o \sin(\gamma)$	$\gamma \le \gamma_{\max}$
3 DOF navigation + 2 DOF point mass	$x, y, z, \gamma, \varepsilon$	V, C_L, ϕ	$\dot{\varepsilon} = \tfrac{1}{2}\rho \frac{m}{S}^{-1} V C_L \sin(\phi)$ $\dot{\gamma} = \tfrac{1}{2}\rho \frac{m}{S}^{-1} V C_L \cos(\phi) - \frac{g\cos(\gamma)}{V}$	$V \le V_{\max}$ $C_L < C_{L_{\max}}$ $\phi \le \phi_{\max}$
3 DOF navigation + 3 DOF point mass	$x, y, z, \gamma, \varepsilon, V$	T, C_L, ϕ	$\dot{V} = \frac{T}{m} - \frac{g}{m}\sin(\gamma) - \tfrac{1}{2}\rho \frac{m}{S}^{-1} V^2 C_D(C_L)$	$T \le T_{\max}$
Many more models!	....	....	....	....

+dynamics
+variables
+complexity
+details

Stanford University, Roberto A. Bunge, AA241X. April 2015

HCV 2D Model Formulation

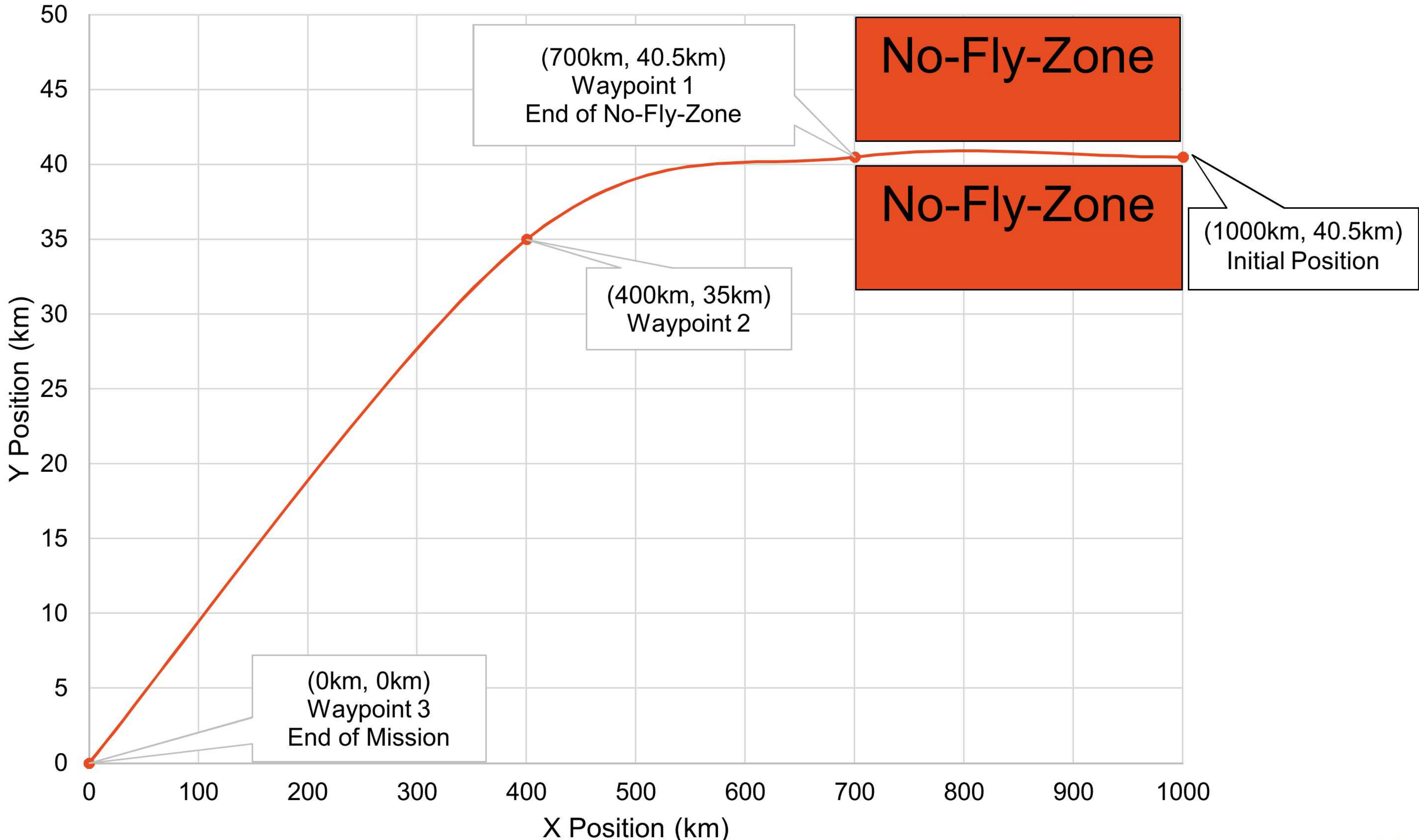
2D HCV Problem Statement and Mission Structure

Mission Structure

1. Hypersonic release with constant thrust to maintain velocity & altitude
2. Hypersonic cruise through No-Fly-Zone corridor (Corridor is 300km in length and 1km in width)
3. Waypoint 1 at the end of hypersonic corridor
4. Turn to waypoint 2
5. Mission end at waypoint 3

- The HCV (Hypersonic Cruise Vehicle) governed by 2D eq. of motion
- This model used as baseline for future higher fidelity models

2D Mission Visualization



2D Model Assumptions & Control Input

- Assumptions
  - Altitude is maintained as constant
  - Deceleration is zero (decouples velocity change from bank angle control input)
- Control input
  - Control u is a function of bank angle normalized by tangent of the maximum bank angle
  - Thrust is constant and maintains acceleration at 0

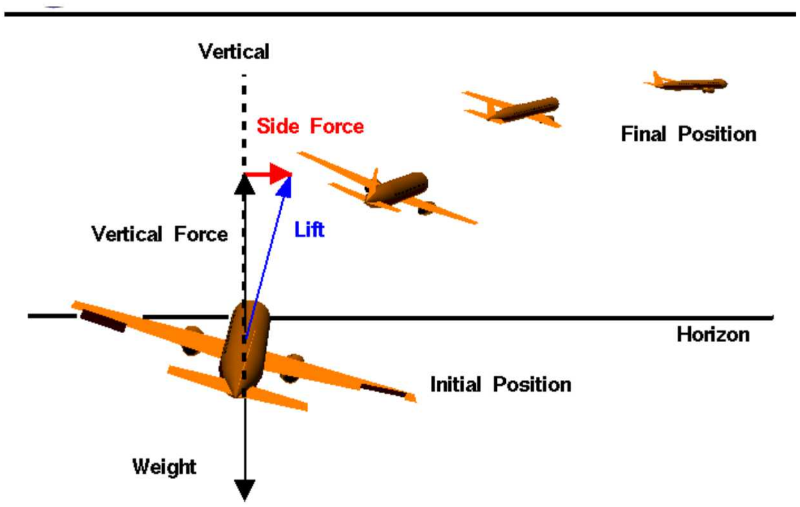
$$u = \frac{\tan(\sigma)}{\tan(\sigma_{\max})}$$

$$\sigma_{\max} = \pm 20°$$

$$-1 \leq u(k) \leq 1$$

[Jorris 2007]

Banking Turn



https://www.grc.nasa.gov/www/k-12/airplane/turns.html

2D Equations of Motion & State Space Model

Equations of Motion

$$\dot{x} = V cos(\theta)$$

$$\dot{y} = V sin(\theta)$$

$$\dot{\theta} = \frac{\tan(\sigma_{max})}{V} u$$

$$\dot{V} = 0$$

Symbol	Parameter
x	x position
y	y position
θ	Heading angle
V	Velocity
a	Acceleration
σ	Bank angle
σ_{max}	Max. bank angle

State Space Model

$$\mathbf{x} = \begin{bmatrix} x \\ y \\ \theta \\ V \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} \qquad \dot{\mathbf{x}} = \begin{bmatrix} x_4 cos(x_3) \\ x_4 sin(x_3) \\ \frac{0.363u}{x_4} \\ 0 \end{bmatrix}$$

$$u = \frac{\tan(\sigma)}{\tan(\sigma_{\max})}$$

$$\tan(\sigma_{max}) = \tan(20°) = 0.363$$

$$\dot{\boldsymbol{x}} = [\, \dot{x} \;\; \dot{y} \;\; \dot{\theta} \;\; \dot{V}]^T = \mathrm{f}(\boldsymbol{x}, \boldsymbol{u})$$

[Jorris 2007]

Optimization Problem (1/3)

- NMPC problem formulation

Minimize
$$J(x(\cdot),u(\cdot)) = (x(t+N)-x_f)^T P(x(t+N)-x_f) + \sum_{k=t}^{t+N-1} (x(k)-x_f)^T Q(x(k)-x_f) + u(k)^T R u(k),$$

subject to

$$x(k+1) = f(x(k),u(k)),$$
$$x(t) = x_i,$$
$$-1 \le u(k) \le 1,$$
$$40 \le x_2(k) \le 41, \qquad \text{if } 700 \le x_1(k) \le 1000$$
$$x(t+N) = x_f \qquad x_f \text{ is updated from waypoints}$$

Optimization Problem (2/3)

- Interior-point optimization algorithms require
  - Hessian of Hamiltonian
  - Jacobian of state constraints

$$H(k) = L(\mathbf{x}(k), \mathbf{u}(k)) + \sum_{l=1}^{n} \lambda_l g_l(\mathbf{x}(k), \mathbf{u}(k))$$

$$\frac{\partial^2 H}{\partial z_i z_j} = \frac{\partial^2 L(\mathbf{x}(k), \mathbf{u}(k))}{\partial z_i z_j} + \sum_{l=1}^{n} \lambda_l \frac{\partial^2 g_l((\mathbf{x}(k), \mathbf{u}(k))}{\partial z_i z_j}$$

where

$$\mathbf{z} = [\, x_1 , x_2 , x_3 , x_4 , u \,]^T$$
$$= [\, z_1 , z_2 , z_3 , z_4 , z_5]^T$$

Symbol	Parameter
H	Hamiltonian
L	Incremental Cost Function
g	Constraint state formulation
λ	Lagrangian multiplier
$\mathbf{x}$	State vector
$\mathbf{u}$	Control vector
$\mathbf{z}$	Total states & controls vector
n	Total number of state constraints

Optimization Problem (3/3)

State Constraint Formulation

$-1 \leq u(k) \leq 1$ ⇒ Handled by explicit control constraint implementation – no state formulation needed

$x(k+1) = f(x(k), u(k))$ ⇒ one constraint formulation for every state (total of 4)
g_1 , g_2 , g_3 , g_4

$40 \leq x_2(k) \leq 41$ ⇒ only considers 1 state (y-pos = x_2)
g_5

$$g = \begin{bmatrix} g_1 \\ g_2 \\ g_3 \\ g_4 \\ g_5 \end{bmatrix} = \begin{bmatrix} x_1(k+1) - \left(x_1(k) + Ts \cdot f\left(x(k), u(k)\right)\right) \\ x_2(k+1) - \left(x_2(k) + Ts \cdot f\left(x(k), u(k)\right)\right) \\ x_3(k+1) - \left(x_3(k) + Ts \cdot f\left(x(k), u(k)\right)\right) \\ x_4(k+1) - \left(x_4(k) + Ts \cdot f(x(k), u(k))\right) \\ x_2(k) \end{bmatrix} = \begin{bmatrix} x_6 - x_1 + Ts \cdot x_4 \cos(x_3) \\ x_7 - x_2 + Ts \cdot x_4 sin(x_3) \\ x_8 - x_3 + Ts \cdot 0.363 x_5 / x_4 \\ x_9 - x_4 \\ x_2 \end{bmatrix}$$

2D Model Software Implementations

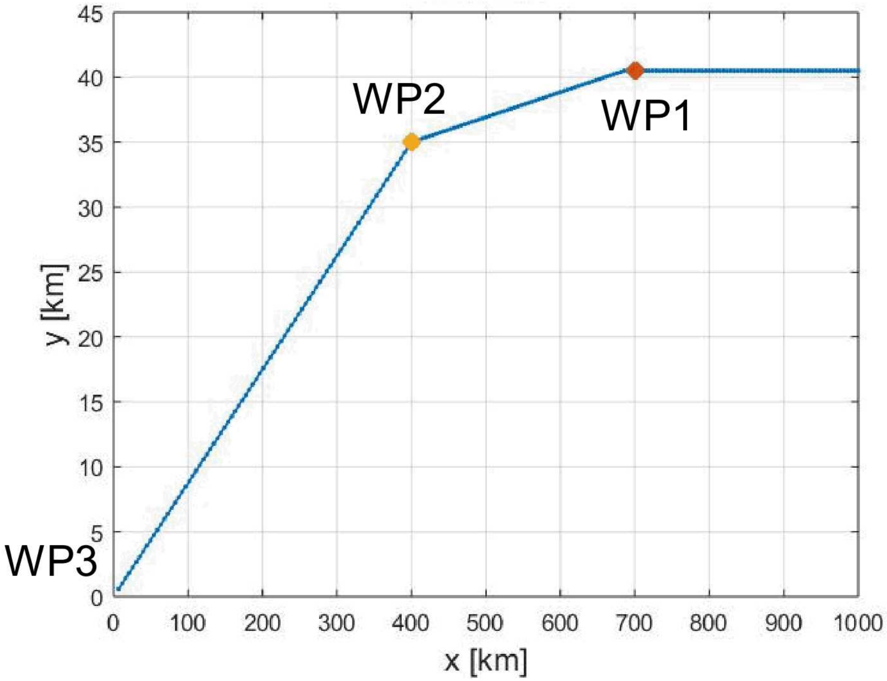
- MATLAB implementation using 'fmincon' function with interior-point optimization algorithm was used for baseline

- COIN-OR IPOPT implemented into a Simulink model in an Ubuntu Linux environment

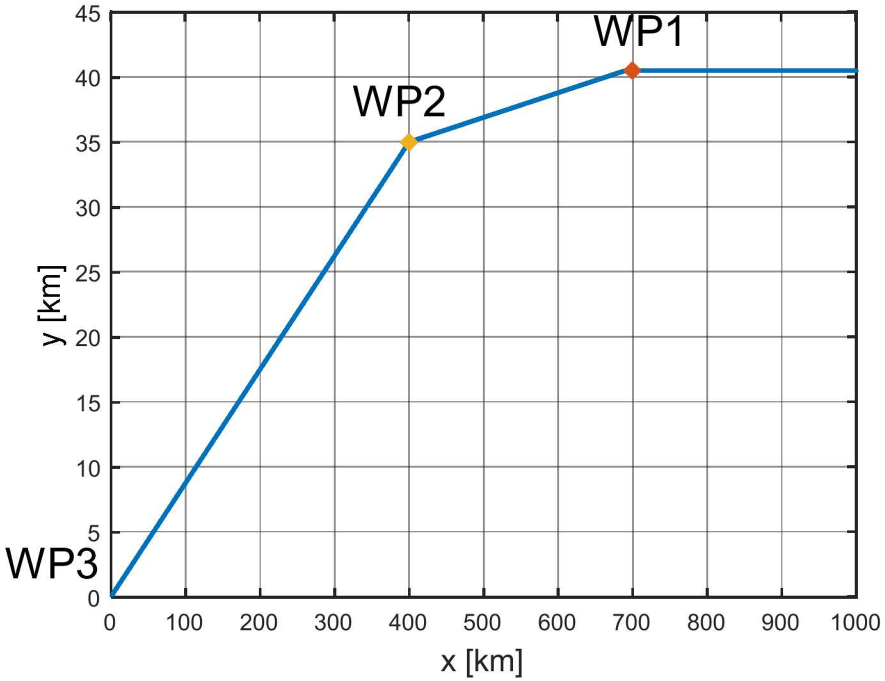
- Ad-hoc conditional logic handles transition between NFZ corridor and sequential waypoint state objectives
  - Initial waypoint: WP1
  - If distance from WP1 is within 10 km
    → Check x-position range, and if it is out of NFZ corridor, then switch to WP2
  - If distance from WP2 is within 10 km
    → Check x-position range and switch to WP3
  - If distance from WP3 is within 10 km
    → Terminate simulation

2D Model Simulation Results (1/3)

MATLAB fmincon Trajectory



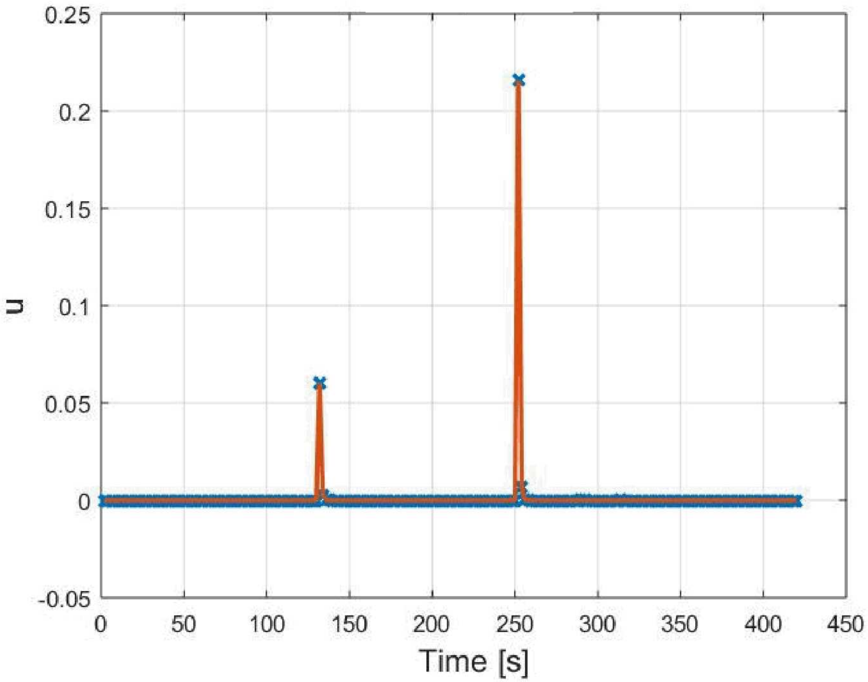
IPOPT/Simulink Trajectory



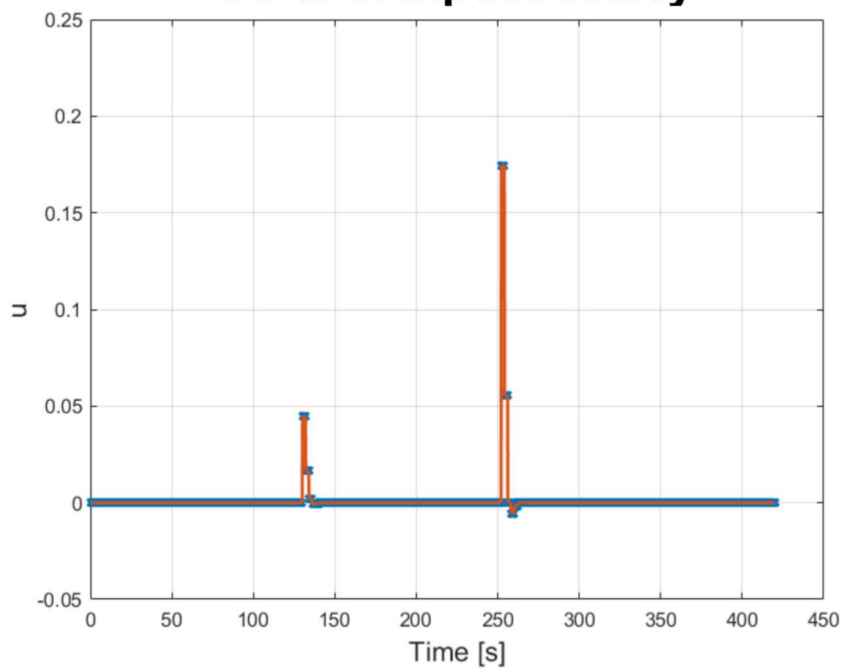
- Identical trajectory performance

2D Model Simulation Results (2/3)

MATLAB fmincon
Control Input History



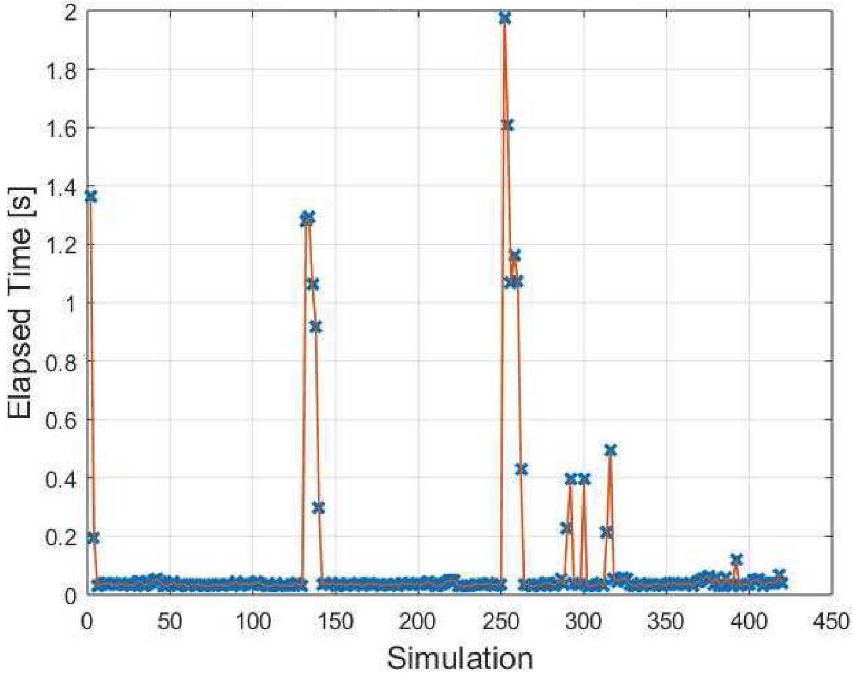
IPOPT/Simulink
Control Input History



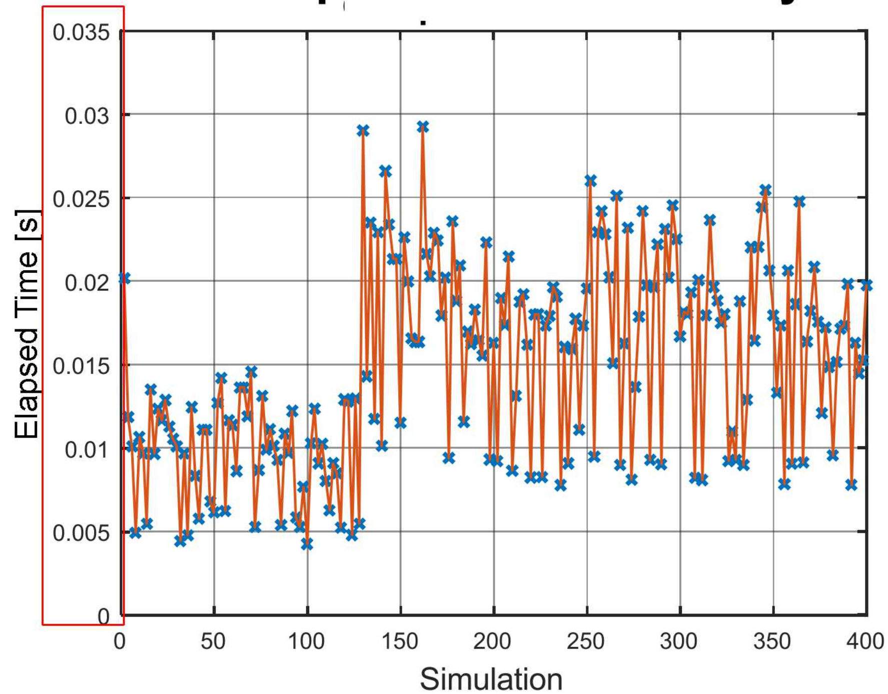
- Inputs are slightly different
- Simulink/IPOPT has smaller control inputs, but distributed in 2-3 sampling instants
- Overall similar results with slightly different algorithm setup

2D Model Simulation Results (3/3)

MATLAB fmincon Computation Time History



IPOPT/Simulink Computation Time History



	MATLAB fmincon	Simulink/IPOPT
Avg. Comp. Time	0.1179 sec	0.0148 sec (~15 ms)
Max. Comp. Time (worst case)	1.9733 sec	0.0293 sec (~29 ms)

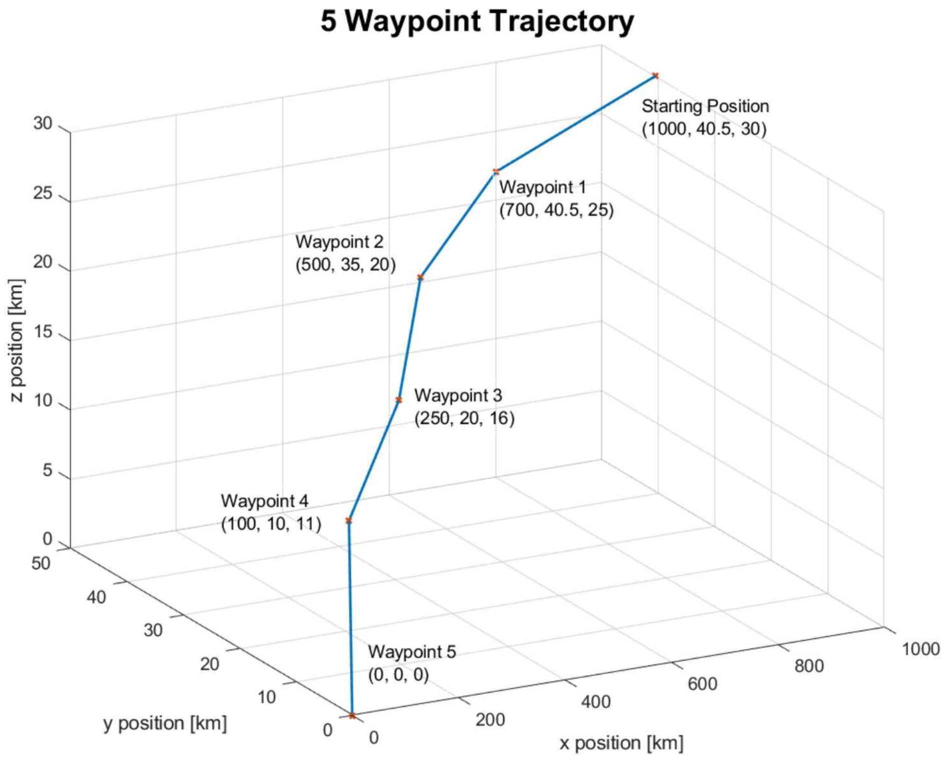
HGV 3D Model Formulations and Results

Problem Statement and Mission Structure

- HGV (Hypersonic Glide Vehicle) governed by 3D Eq. of motion
- No propulsion power (thrust will not be a control variable)

Mission Structure

1. Hypersonic release (no engine thrust)
2. Hypersonic glide through No-Fly-Zone corridor (Corridor is 300km in length and 1km in width)
3. Waypoint 1 at the end of NFZ corridor (altitude decreases)
4. Waypoint following sequence from 1 to 5 decreasing in altitude



3D Eq. of Motion and State Space Model

Hood Reformulation

$$\dot{x} = V\cos(\gamma)\cos(\theta)$$

$$\dot{y} = V\cos(\gamma)\sin(\theta)$$

$$\dot{h} = V\sin(\gamma)$$

$$\dot{V} = -\frac{D}{m} - g\sin(\gamma)$$

$$\dot{\gamma} = \frac{L\cos(\sigma) - S\sin(\sigma)}{mV} - \frac{g\cos(\gamma)}{V}$$

$$\dot{\theta} = \frac{L\sin(\sigma) + S\cos(\sigma)}{mV\cos(\gamma)}$$

Modified Formulation

$$\dot{x} = V\cos(\gamma)\cos(\theta)$$

$$\dot{y} = V\cos(\gamma)\sin(\theta)$$

$$\dot{h} = V\sin(\gamma)$$

$$\dot{V} = -\frac{\rho V^2 S C_L^*(1 + c_l^2)}{4E^* m} - g\sin(\gamma)$$

$$\dot{\gamma} = \frac{\rho c_l C_L^* S V\cos(\sigma)}{2m} - \frac{S\sin(\sigma)}{mV} - \frac{g\cos(\gamma)}{V}$$

$$\dot{\theta} = \frac{\rho c_l C_L^* S V\sin(\sigma)}{2m\cos(\gamma)} + \frac{S\cos(\sigma)}{mV\cos(\gamma)}$$

where

$$L = \frac{1}{2}\rho V^2 c_l C_L^* S$$

$$D = \frac{1}{2}\rho V^2 \frac{C_L^*(1 + c_l^2)}{2E^*} S$$

*Note: Multiple formulations of 3D model were evaluated. Flight path angle small angle & Three waypoints

$$\boldsymbol{x} = \begin{bmatrix} x \\ y \\ h \\ V \\ \gamma \\ \theta \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix}$$

$$\boldsymbol{u} = \begin{bmatrix} c_l \\ \sigma \end{bmatrix} = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

$$\dot{\boldsymbol{x}} = \begin{bmatrix} x_4\cos(x_5)\cos(x_6) \\ x_4\cos(x_5)\sin(x_6) \\ x_4\sin(x_5) \\ -\frac{\rho C_L^* S x_4^2(1 + u_1^2)}{4mE^*} - g\sin(x_5) \\ \frac{\rho C_L^* S x_4 u_1\cos(u_2)}{2m} - \frac{S\sin(u_2)}{m x_4} - \frac{g\cos(x_5)}{x_4} \\ \frac{\rho C_L^* S x_4 u_1\sin(u_2)}{2m\cos(x_5)} + \frac{S\cos(u_2)}{m x_4\cos(x_5)} \end{bmatrix}$$

*Note: Air density is recalculated at every sampling time
($\rho = \rho_{sl} e^{-\beta h}$)
$\rho_{sl} = density\ at\ sea\ level\ \left(1.225\ \frac{kg}{m^3}\right)$
$\beta = air\ density\ decay\ factor\ (-0.14 km^{-1})$
$h = altitude\ (km)$

[Jorris 2007]
[Hood 2019]

Aerodynamic Heating

- Formulation for nose heating was added to the model to guide future considerations for thermal or structural constraints

$$\dot{q} = \frac{k}{\sqrt{r_{nose}}}\left(\frac{\rho}{\rho_{sl}}\right)\left(\frac{V}{\sqrt{g_0 r_0}}\right)^3$$

$$where\ k = 17000\ \frac{BTU \cdot ft^{\frac{-3}{2}}}{s}$$

- Note: $\dot{q}$ is calculated in BTU/s and converted to Watts afterwards

[Jorris 2007]

Symbol	Parameter
k	Heating constant
r_{nose}	Radius of aircraft nose
ρ	Local air density
ρ_{sl}	Sea level air density
V	Velocity
g_0	Earth Gravity
r_0	Radius from earth center

Updated Optimization Problem

State constraint formulation

$x(k+1) = f(x(k), u(k))$ ⇒ one constraint formulation for every state (total of 6)

g_1 , g_2 , g_3 , g_4, g_5, g_6

$40 \leq x_2(k) \leq 41$ ⇒ only considers 1 state (y-pos = x_2)

g_7

$$g = \begin{bmatrix} g_1 \\ g_2 \\ g_3 \\ g_4 \\ g_5 \\ g_6 \\ g_7 \end{bmatrix} = \begin{bmatrix} x_9 - x_1 - Ts \cdot x_4\, cos(x_5)cos(x_5) \\ x_{10} - x_2 - Ts \cdot x_4\, cos(x_5)sin(x_5) \\ x_{11} - x_3 - Ts \cdot x_4\, sin(x_5) \\ x_{12} - x_4 - Ts \cdot (-\frac{\rho C_L^* S x_4^2 (1+u_1^2)}{4mE^*} - gsin(x_5)) \\ x_{13} - x_5 - Ts \cdot (\frac{\rho C_L^* S x_4\, u_1 cos(u_2)}{2m} - \frac{Ssin(u_2)}{mx_4} - \frac{gcos(x_5)}{x_4}) \\ x_{14} - x_6 - Ts \cdot (\frac{\rho C_L^* S x_4 u_1\, sin(u_2)}{2mcos(x_5)} + \frac{Scos(u_2)}{mx_4\, cos(x_5)}) \\ x_2 \end{bmatrix}$$

Simulation Parameters

MATLAB fmincon Implementation

Optimization Parameters

n_s = 6 - Number of State Variables
n_u = 2 - Number of Control Variables
n_a = 8 - State and Control Variables
n_c = 7 - Number of Constraints
T_s = 2 - Sampling Time [sec]
N = 20 - Prediction Horizon

Weighting matrices
Q = diag[0.005, 0.03, 0.03, 0, 0, 0]
R = [1,1]
$S = 2 \cdot Q$

For waypoint 4 to 5 $S = 3 \cdot Q$

Initial Conditions

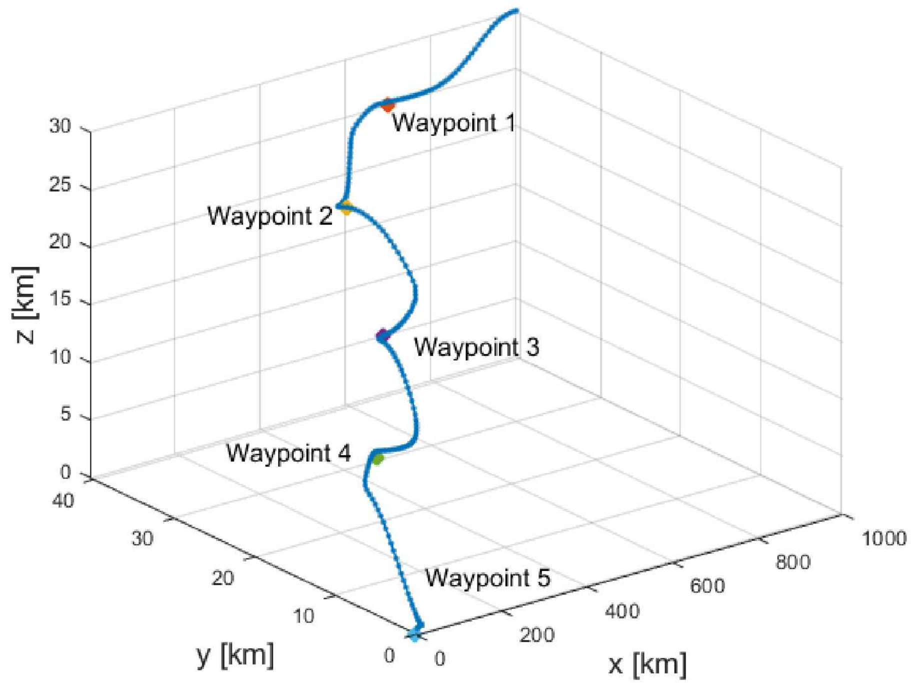
x_1 (x-pos) initial = 1000 km
x_2 (y-pos) initial = 40.5 km
x_3 (z-pos) initial = 30 km
x_4 (V) initial =4.08 km/s (Mach 12)
x_5 (γ) initial = 0 rad (0 deg)
x_6 (θ) initial = π rad (180 deg)

Waypoints (Target)

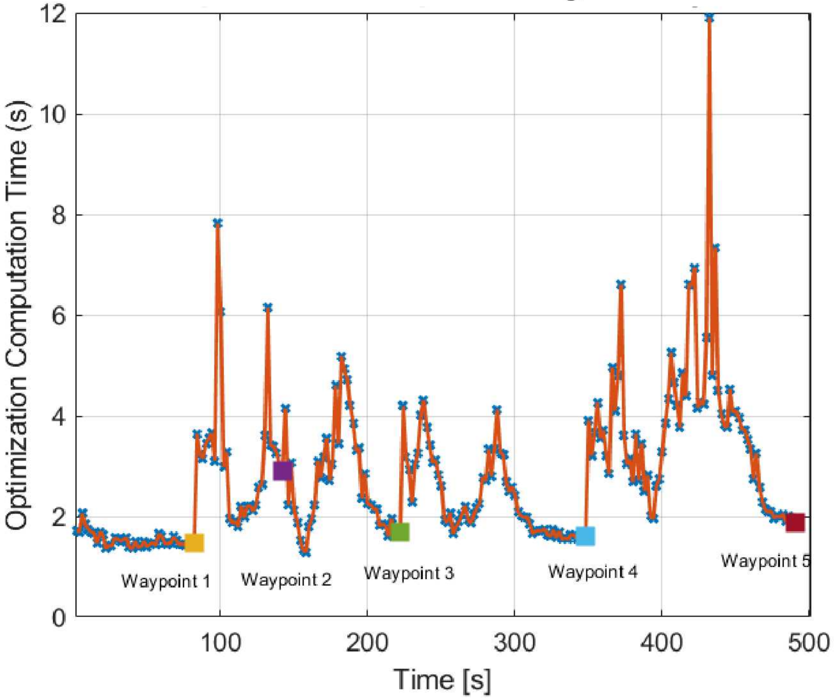
WP1: (700, 40.5, 23) [km]
WP2: (500, 35, 18) [km]
WP3: (300, 20, 10) [km]
WP4: (100, 10, 6) [km]
WP5: (0, 0, 0) [km]

3D Model Simulation Results (1/2)

3D Model 5 Waypoint Trajectory



3D Model Computation Time History



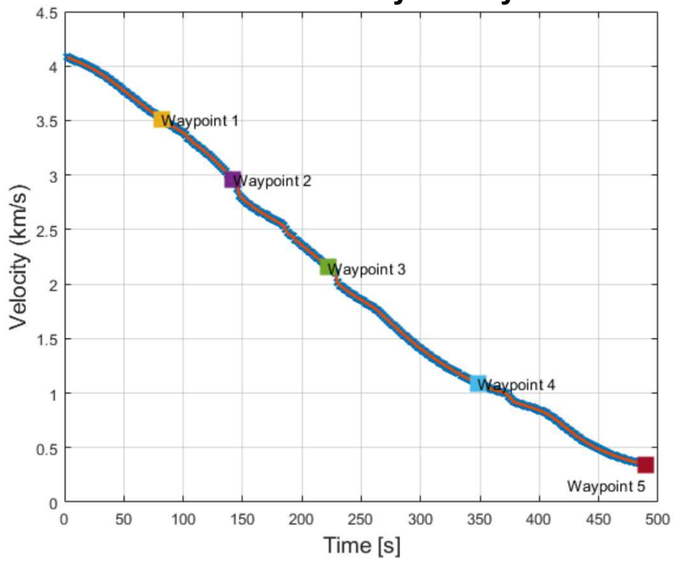
Avg. Time = 2.78 s

- Trajectory smoothness better but still room for improvement
- Aircraft takes discrete turns in y and z before completing x state goal

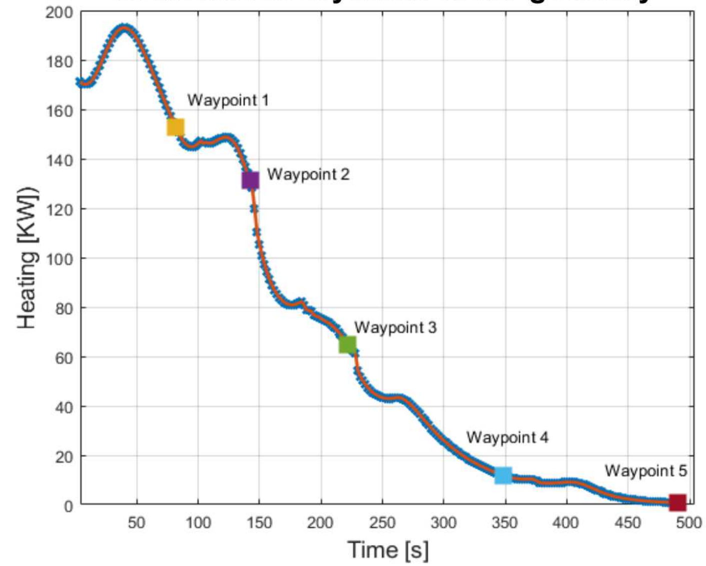
3D Model Simulation Results (2/2)

- Velocity is well maintained by the 3D no-small-angle model.
- Aerodynamic heating reaches a maximum of approx. 195 kW
- Control input in large bursts as the model maneuvers before and after waypoint checks

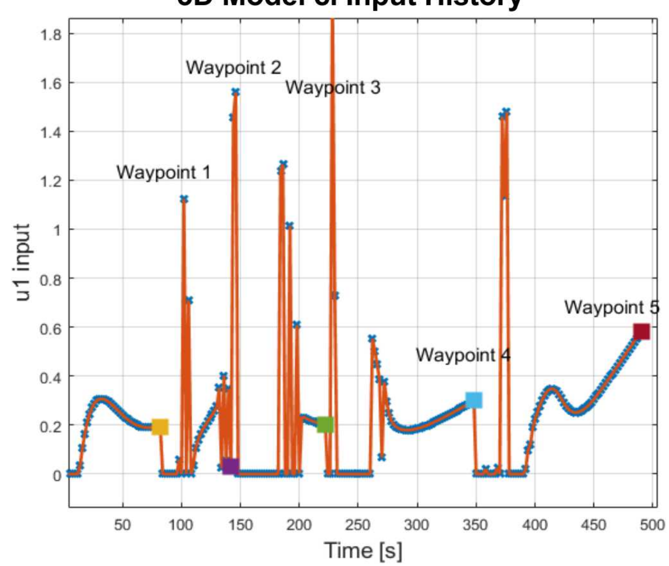
3D Model Velocity History



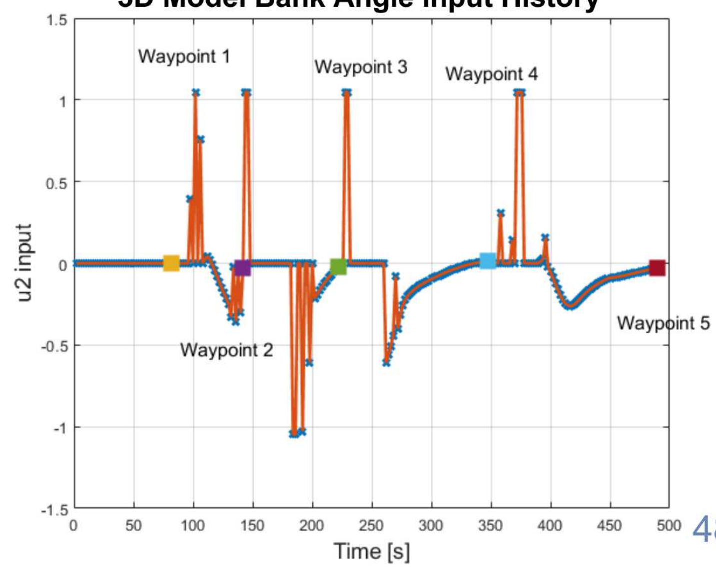
3D Model Aerodynamic Heating History



3D Model cl Input History



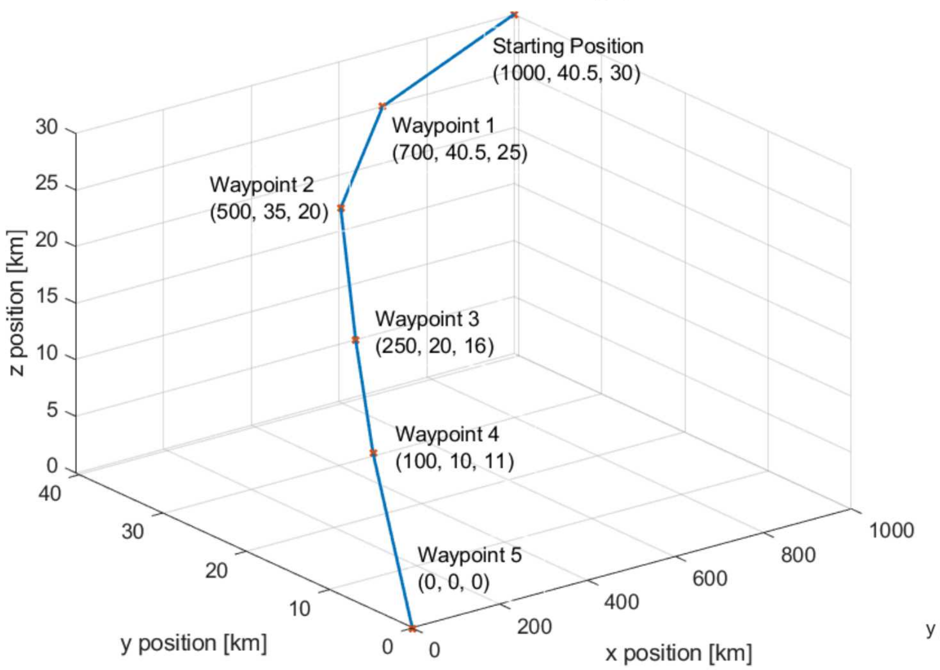
3D Model Bank Angle Input History



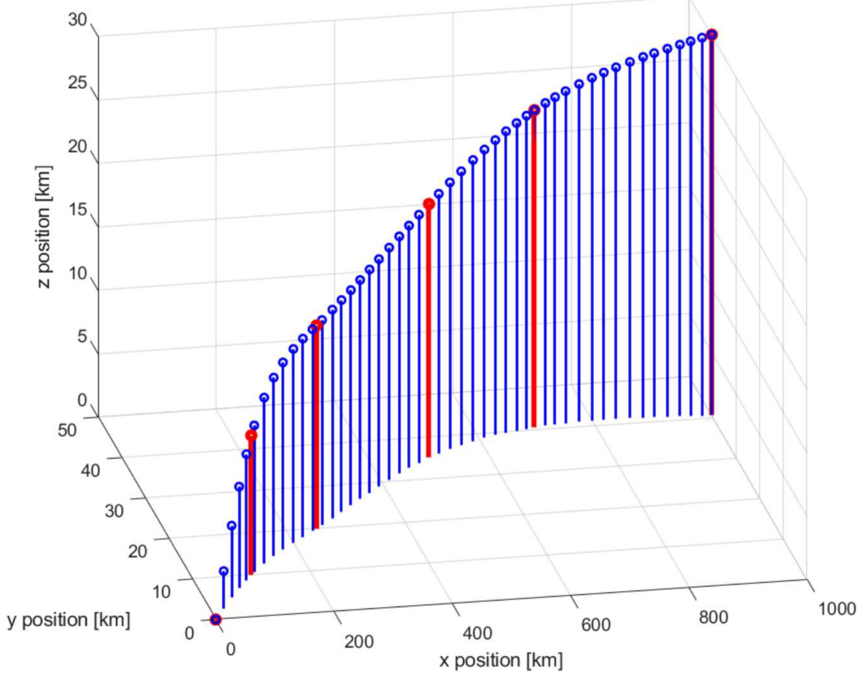
Waypoint Reformulation

- Waypoint refined to generate a smother trajectory
- 5 waypoints should be used to generate a smooth spline of several dozen waypoints to enforce strict waypoint following

Reformulated 5 Waypoints



50 Sub-Waypoint Point Spline



Parameters (Spline)

MATLAB fmincon Implementation

Optimization Parameters

n_s = 6 - Number of State Variables
n_u = 2 - Number of Control Variables
n_a = 8 - State and Control Variables
n_c = 7 - Number of Constraints
T_s = 2 - Sampling Time [sec]
N = 20 - Prediction Horizon

Weighting matrices
Q = diag[0.0002, 0.25, 0.25, 0, 0, 0]
R = [1,1]
$S = 2 \cdot Q$

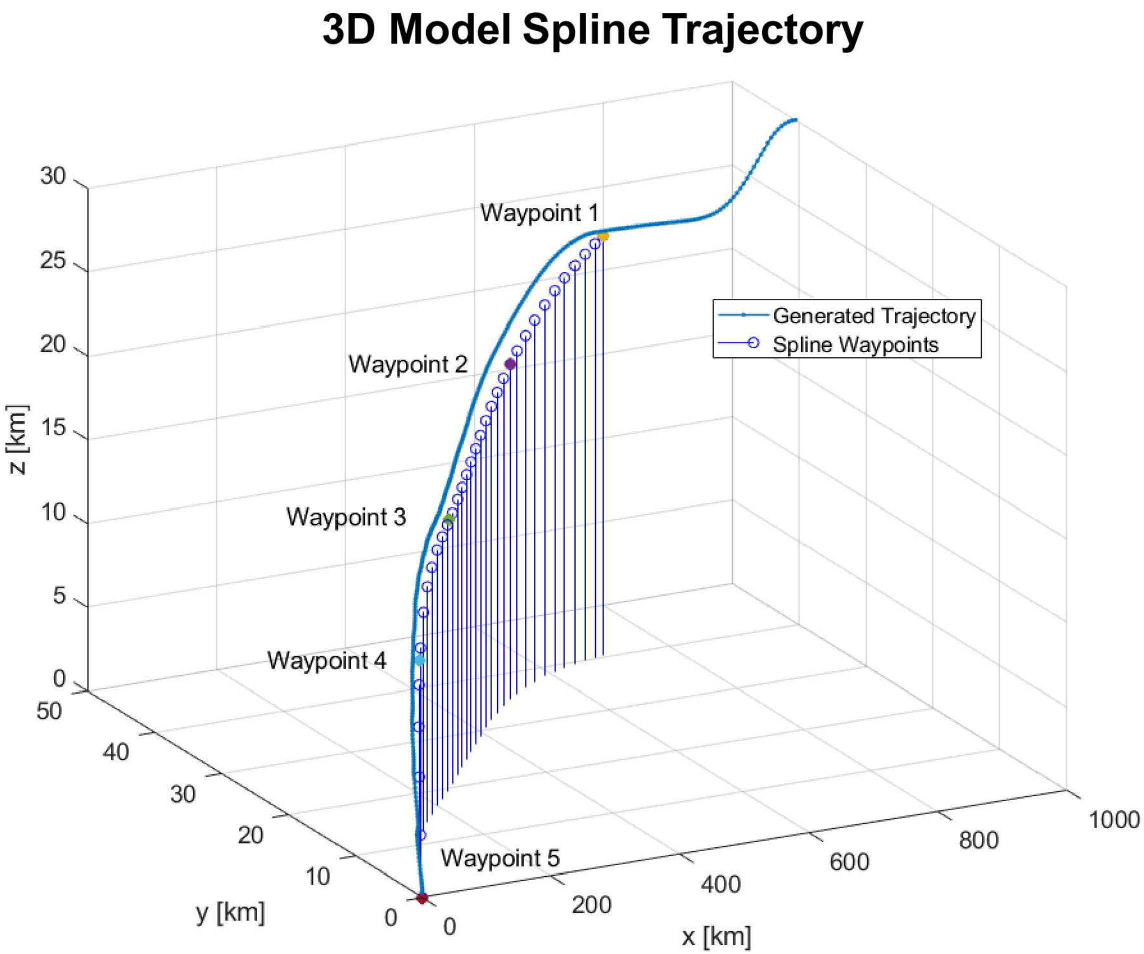
Initial Conditions

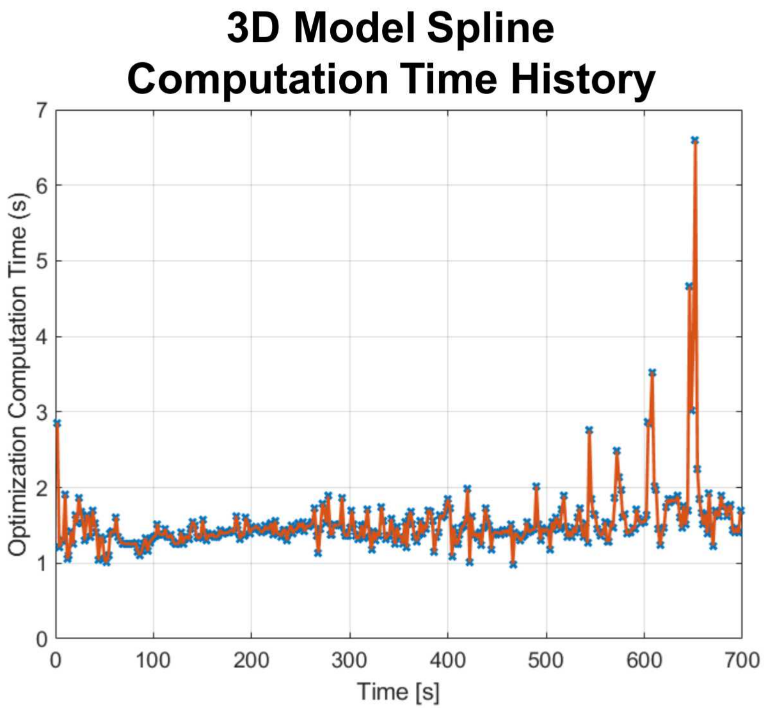
x_1 (x-pos) initial = 1000 km
x_2 (y-pos) initial = 40.5 km
x_3 (z-pos) initial = 30 km
x_4 (V) initial =2.72 km/s (Mach 8)
x_5 (γ) initial = 0 rad (0 deg)
x_6 (θ) initial = π rad (180 deg)

Waypoints (Target)

33 Sub-waypoints from end of No-Fly-Zone (700, 40.5, 25) to old Waypoint 5 (0,0,0)

3D Model Spline Results (1/2)



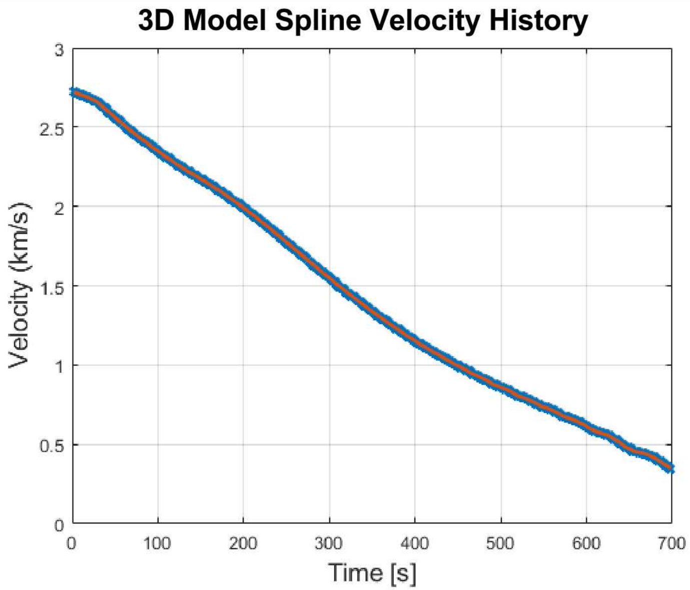


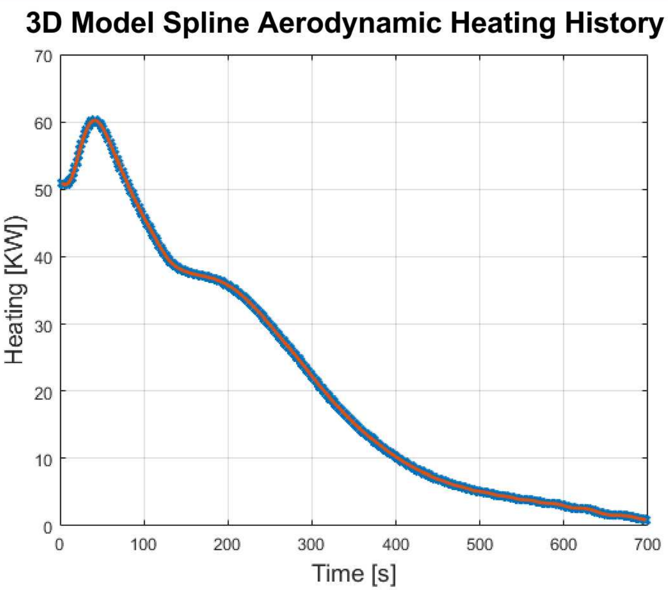
Avg. Time = 1.53 s

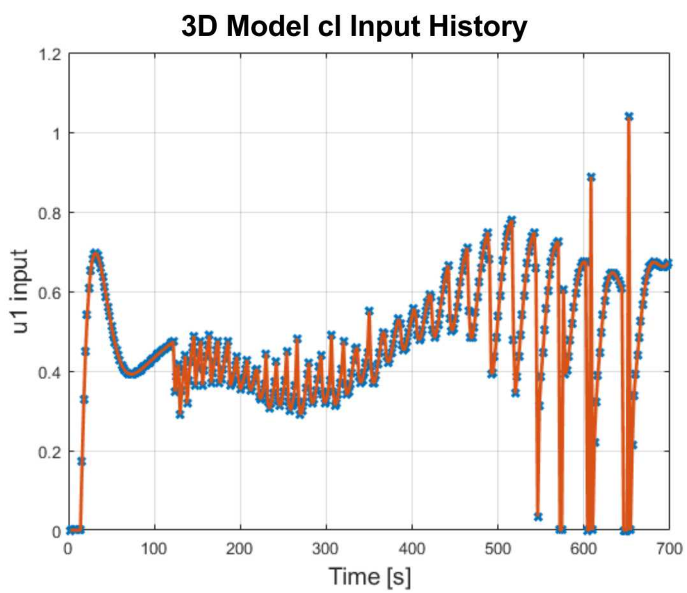
- Trajectory smoothness greatly improved
- Favorable computation time performance
- Computation time depends on complex interactions between simulation params

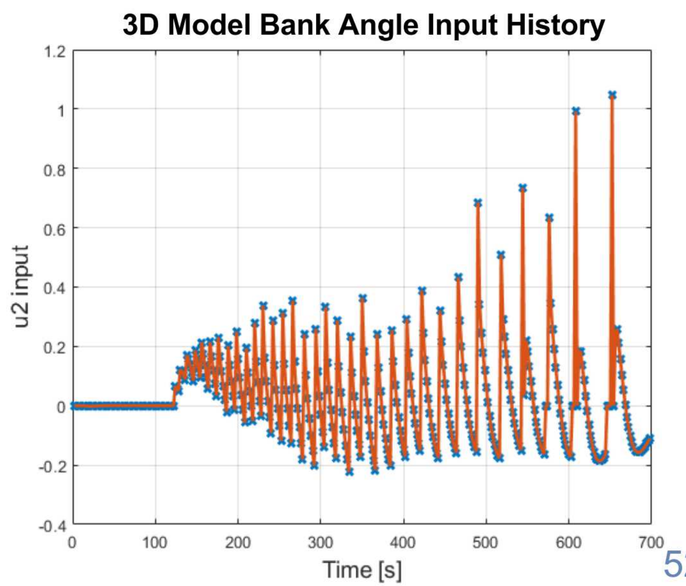
3D Model Spline Results (2/2)

- Velocity is well maintained despite lower initial velocity and constant cl control input.
- Aerodynamic heating reaches a maximum of approx. 60 kW
- Control input in constant fluctuations with no maximum constrained inputs



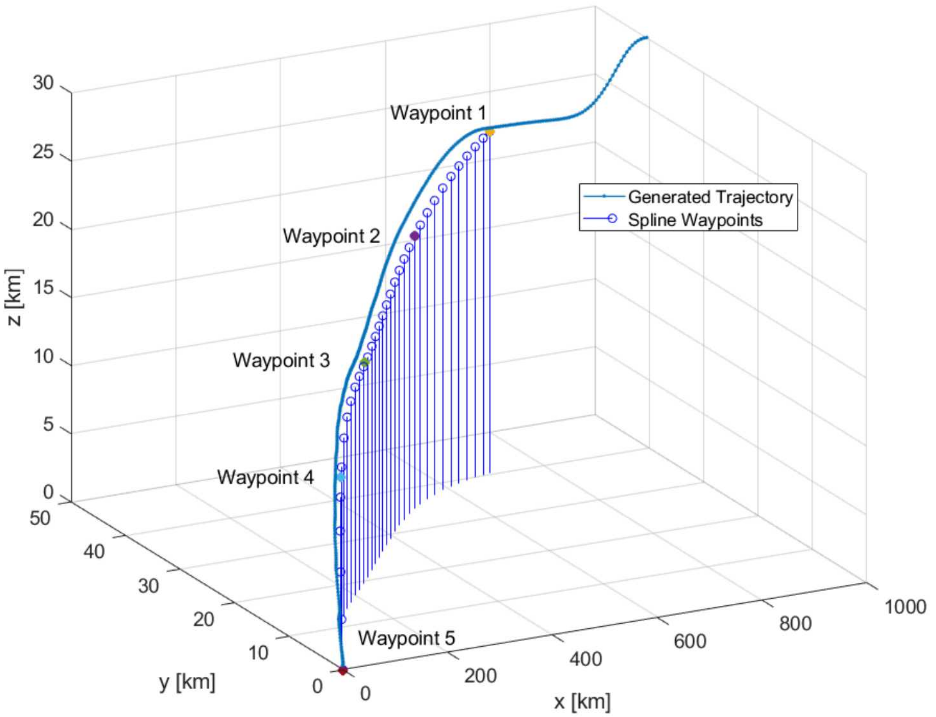




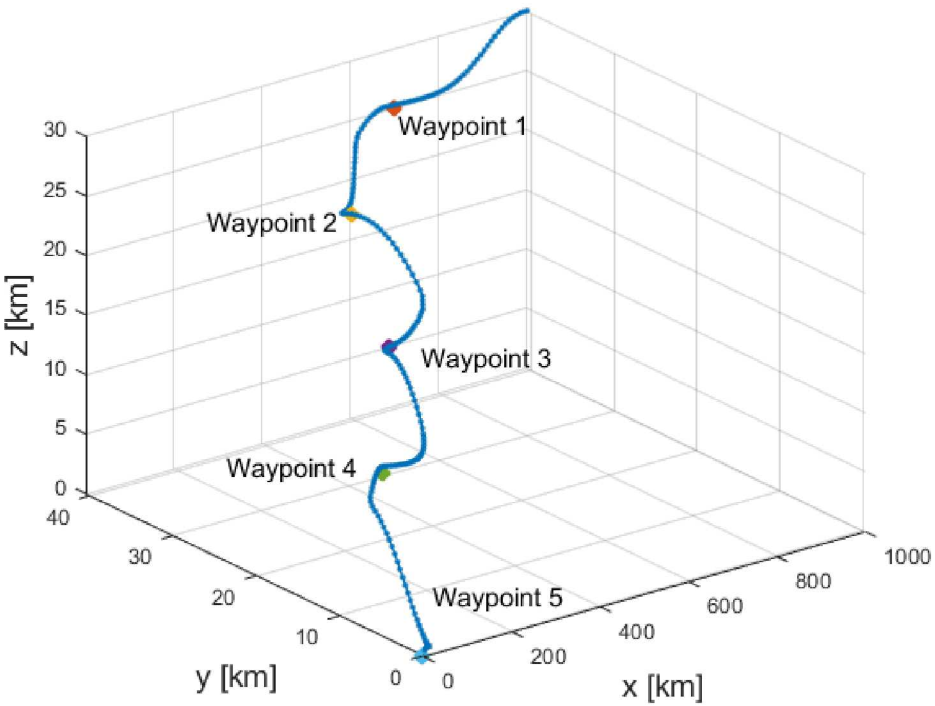


Trajectory Comparisons

3D Model Spline Trajectory

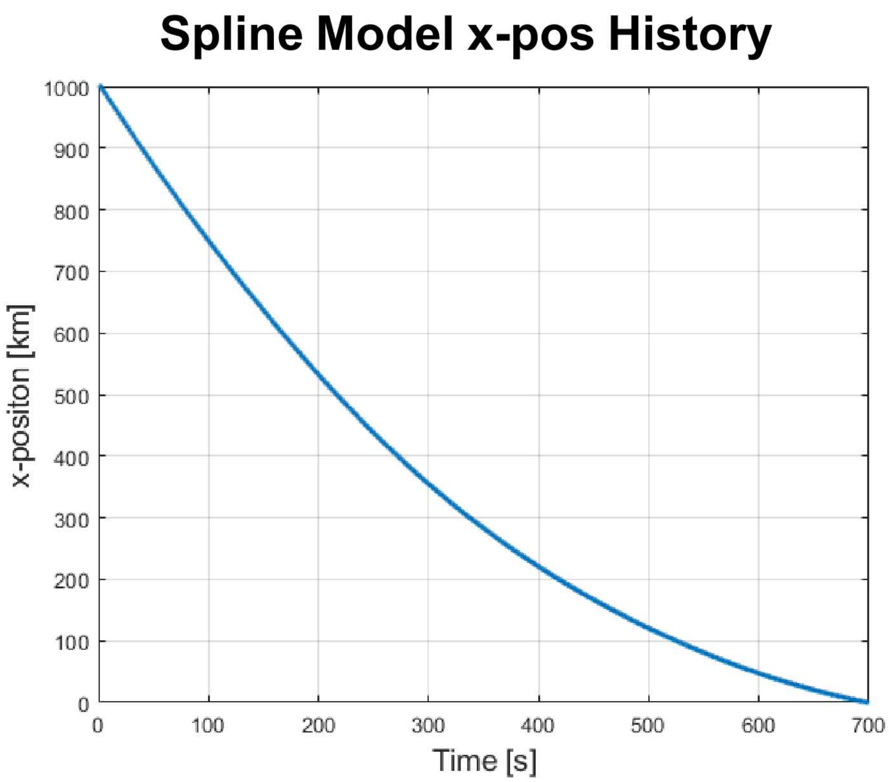


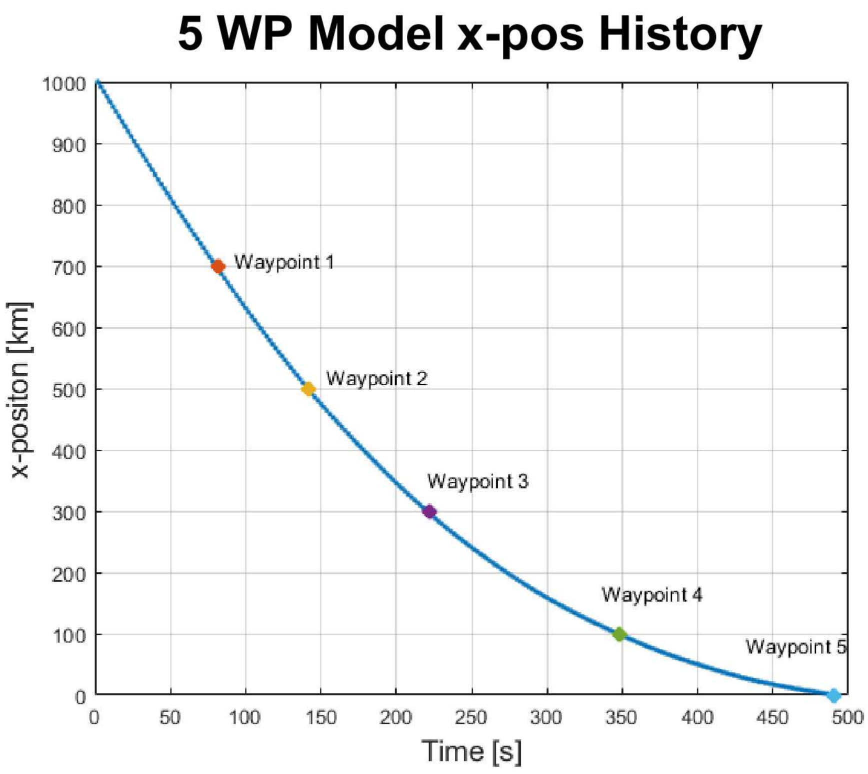
3D Model 5 Waypoint Trajectory



- NMPC framework for hypersonic flight works best with many short term state objectives rather than few long term objectives
- Spline model has high accuracy for final waypoint (Within 50m)

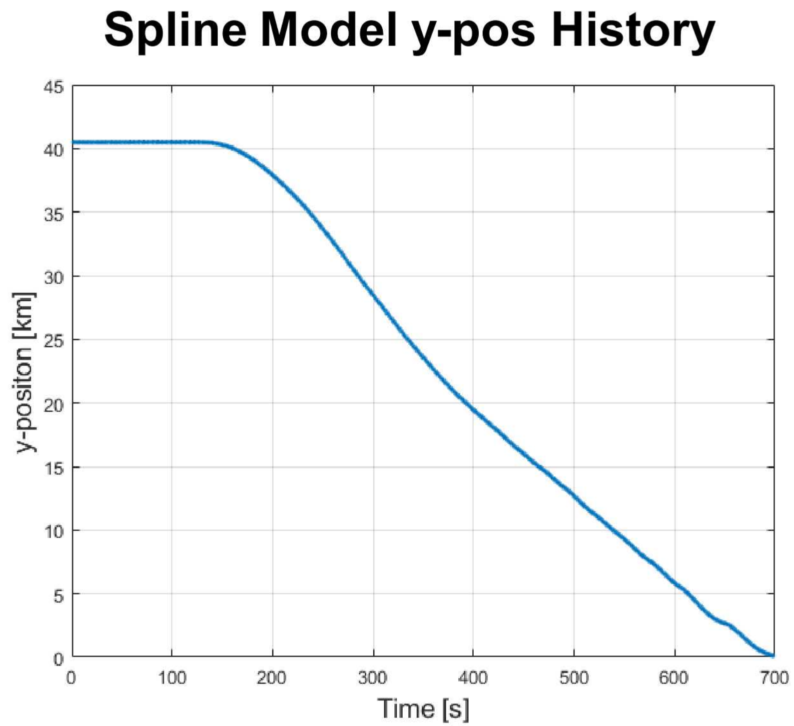
Discrete x-position Comparison

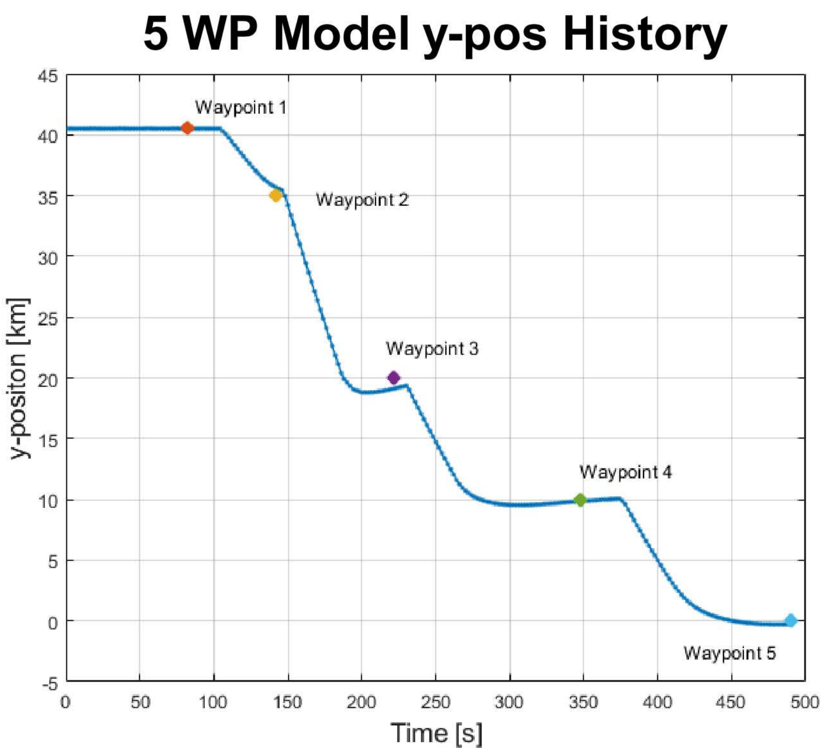




- Negligible difference in x-pos trajectory performance

Discrete y-position Comparison

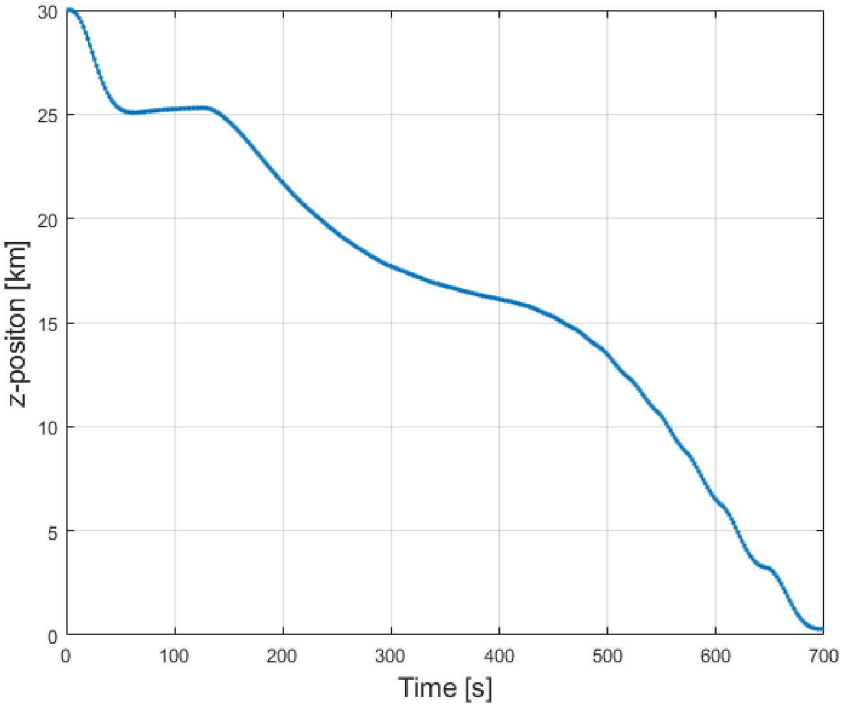




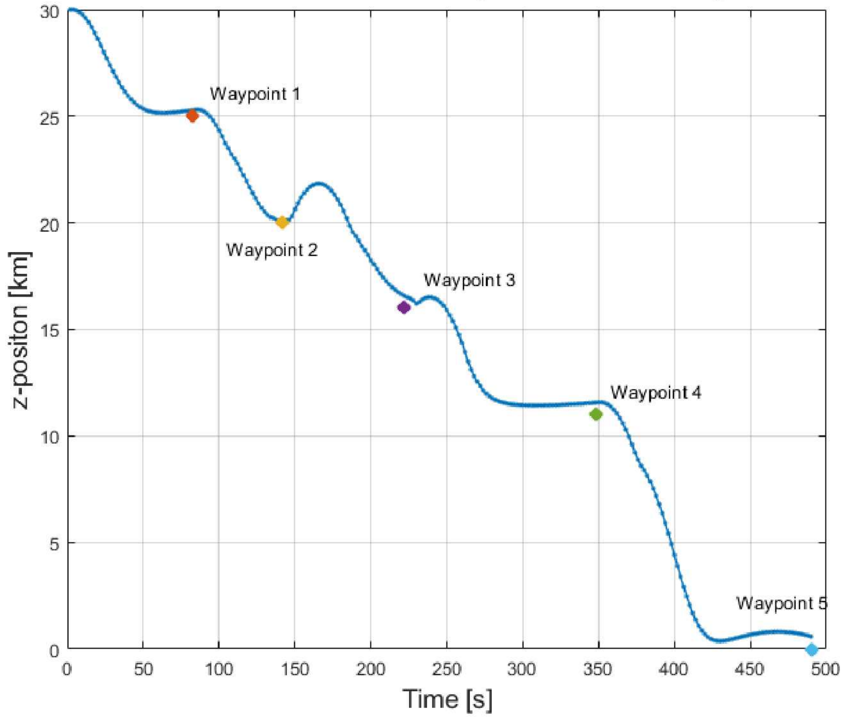
- Spline model is much improved. Less slope variation (i.e. smoother trajectory)
- Improvements in efficiency, reduction in structural loading

Discrete z-position Comparison

Spline Model z-pos History



5 WP Model z-pos History



- Spline model is much improved. Less slope variation (i.e. smoother trajectory)
- Spline model has negligible in z-pos
- Improvements in efficiency, reduction in structural loading

Conclusions and Future Work

Robotic Obstacle Avoidance

- Developed NMPC framework for static obstacle avoidance
- Integrated obstacle avoidance case study results into ROS/Gazebo Simulation
- **Future Work:**
  - Multi-obstacle cases & moving obstacle cases
  - Integration of algorithms with real world hardware

Hypersonic Trajectory Generation

- Developed NMPC framework for hypersonic flight vehicle trajectory generation
- Integrated multiple iterations of vehicle dynamics models to achieve best trajectory results
- **Future Work:**
  - 3D model IPOPT integration for real-time computational capability validation
  - 6DOF model with added thermo/structural constraints

Acknowledgements

Advisor
Dr. Hyeongjun Park

Committee
Dr. Hyeongjun Park (Chair)
Dr. Liang Sun
Dr. James McAteer (Dean's Representative)

New Mexico Space Grant Consortium
Dr. Paulo Oemig

NASA Ames Research Center
Dr. Brian Coltin
Jose Benavides

Sandia National Labs
Dr. Bethany Nicholson

References

E. F. Camacho and C. Bordons, "Model Predictive Control". Springer, 2007.

D. Mayne, J. Rawlings, C. Rao, and P. Scokaert. Constrained model predictive control: stability and optimality. Automatica, 36(6):789-814, 2000.

F. Allgöwer, R. Findeisen, and Z. Nagy. Nonlinear model predictive control: from theory to application. Journal-Chinese Institute of Chemical Engineers, 35(3):299-316. 2004.

P. Dufour, Y. Tour´e, D. Blanc, and P. Laurent, "On nonlinear distributed parameter model predictive control strategy: on-line calculation time reduction and application to an experimental drying process", Computers and Chemical Engineering, vol. 27, no. 11, pp. 1533–1542, 2003

M. Hovd, J Lee, and M. Morari. Truncated step response models for model predictive control. Journal of Process Control, 3(2):67-73, 1993.

A. Bemporad, M. Morari, V. Dua, and E. Pistikopoulos. The explicit linear quadratic regulator for constrained systems. Automatica, 38(1):3-20, 2002.

T. Johansen. Approximate explicit receding horizon control of constrained nonlinear systems. Automatica, 40(2):293-300, 2004.

P. Tondel. Constrained optimal control via multiparametric quadratic programming. PhD thesis, Norwegian University of Science and Technology, 2003.

M. Diehl, H. Ferreau, and N. Haverbeke. Efficient numerical methods for nonlinear MPC and moving horizon estimation. Nonlinear Model Predictive Control, pages 391-417, 2009.

Y. Wang and S. Boyd, "Fast Model Predictive Control Using Online Optimization," in *IEEE Transactions on Control Systems Technology*, vol. 18, no. 2, pp. 267-278, March 2010.

S. Hovland, K. Willcox, J. T. Gravdahl, Jan. (2007). MPC for Large-Scale Systems via Model Reduction and Multiparametric Quadratic Programming. Proceedings of the IEEE Conference on Decision and Control. 3418 - 3423.

R. B. Choroszucha, J. Sun and K. Butts, "Closed-loop model order reduction and MPC for diesel engine airpath control," *2015 American Control Conference (ACC)*, Chicago, IL, 2015, pp. 3279-3284

A. Wächter, L. Biegler. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Math. Program.* 106, 25–57 (2006).

J. Yoo, I. Park, V. To, J. Q. H. Lum and T. Smith, "Avionics and perching systems of free-flying robots for the International Space Station," 2015 IEEE International Symposium on Systems Engineering (ISSE), Rome, 2015, pp. 198-201.

L. Fluckiger, & B. Coltin. "Astrobee Robot Software: Enabling Mobile Autonomy on the ISS". (2019).

T. Jorris, "Common aero vehicle autonomous reentry trajectory optimiza-tion satisfying waypoint and no-fly zone constraints," Ph.D. dissertation, AirForce Institute of Technology, 2007

L. Hood, G. Bennet, and J. Parish, "Developing a 3-DOF compliant perchingarm for a free-flying robot on the International Space Station," 2017 IEEEInternational Conference on Advanced Intelligent Mechatronics (AIM), 2019.

Q & A

BE BOLD. Shape the Future.

College of Engineering