SIPS Honours Thesis Seminar – Progress as of 02 October 2018

Lossless Convexification for Quadrotors

Mathew Gardiner

Supervisor

Assoc. Prof. KC Wong
The University of Sydney

Mentors

Theodore Tzanetos
Gene Merewether
Jet Propulsion Laboratory
California Institute of Technology

THE UNIVERSITY OF SYDNEY

Jet Propulsion Laboratory
California Institute of Technology

Overview

- TORQ system
- Background
  - Convex optimization
  - Differential flatness
  - Existing quadrotor trajectory optimization
- Lossless convexification
- Implementation
- Results

This research was carried out at the Jet Propulsion Laboratory, California Institute of Technology, and was sponsored by the University of Sydney' Industry Placement Scholarship and the National Aeronautics and Space Administration.

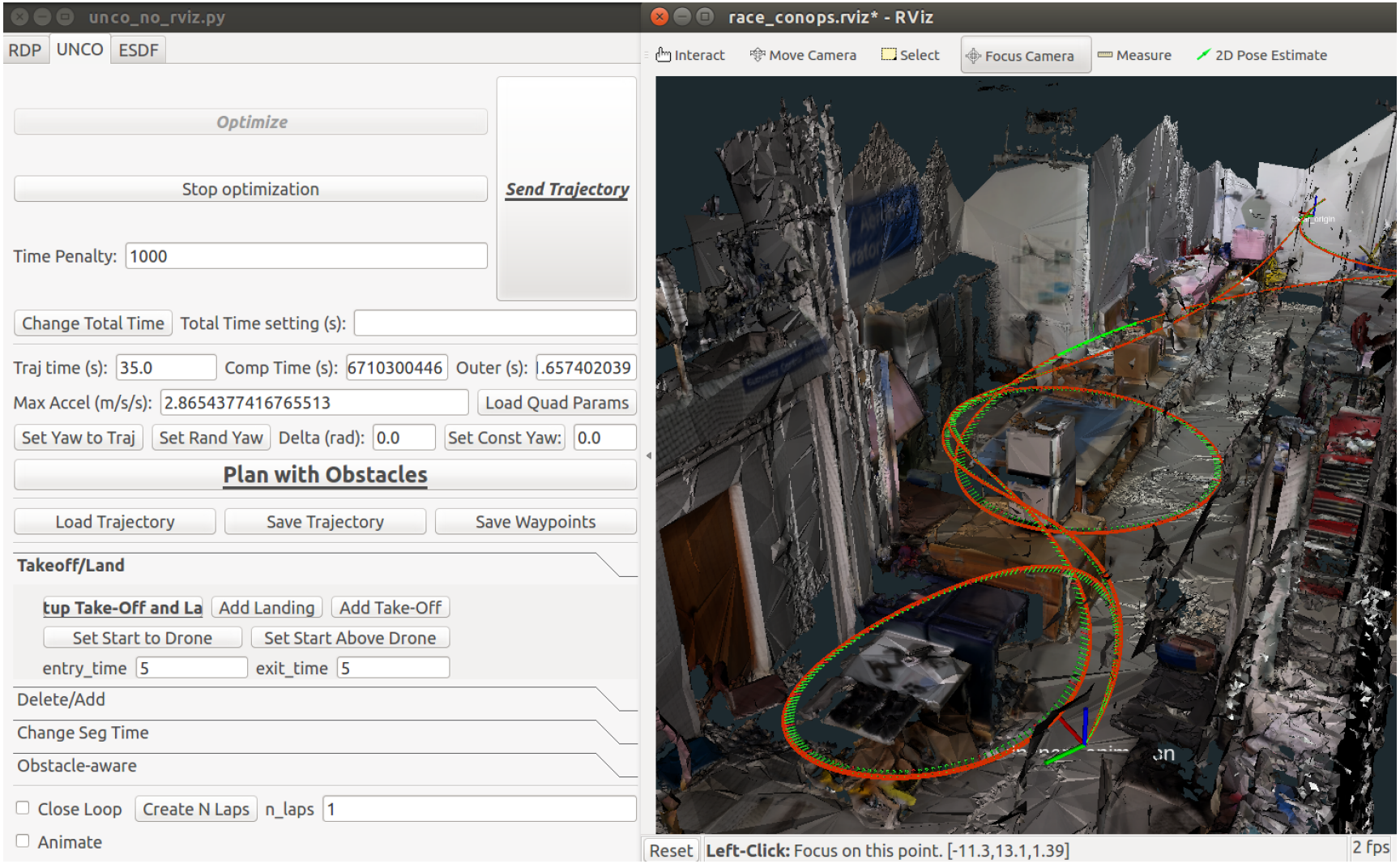
TORQ System

Quadrotor

Rigter, 2018 [1]

TORQ System

Ground Control Station

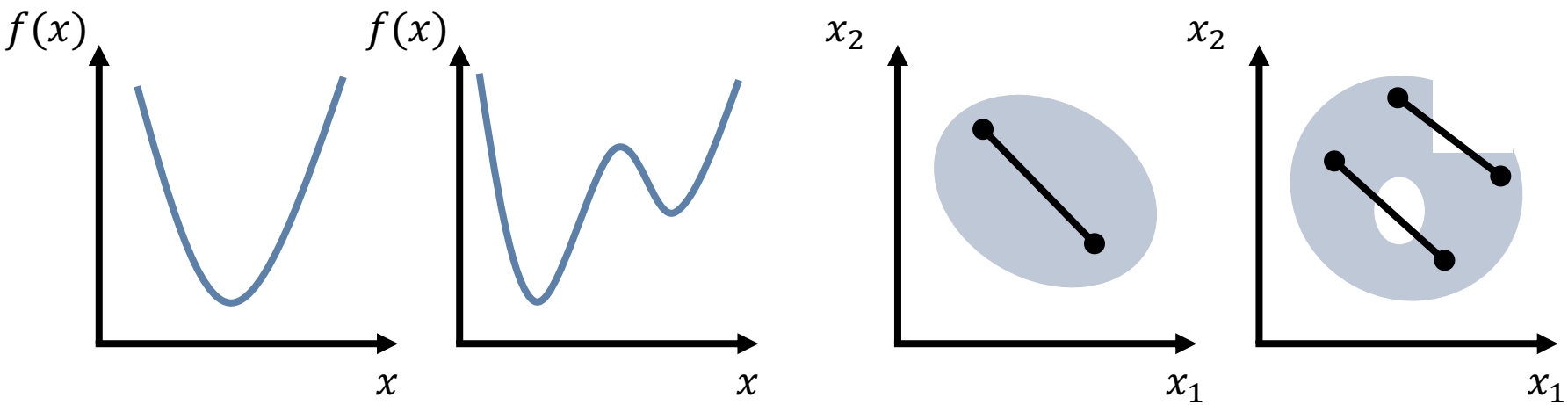


Convex optimisation

$$\begin{aligned} \underset{x}{\text{minimise}} \quad & f(x) \\ \text{such that} \quad & g_i(x) \leq b_i \ , \qquad i = 1, \dots, N \\ & Ax = b \end{aligned}$$

$f(x),\ g_i(x)$ are convex functions

$g_i(x) \leq b_i$ forms a convex set



Convex Non-convex Convex Non-convex

Differential flatness

Quadrotor transform

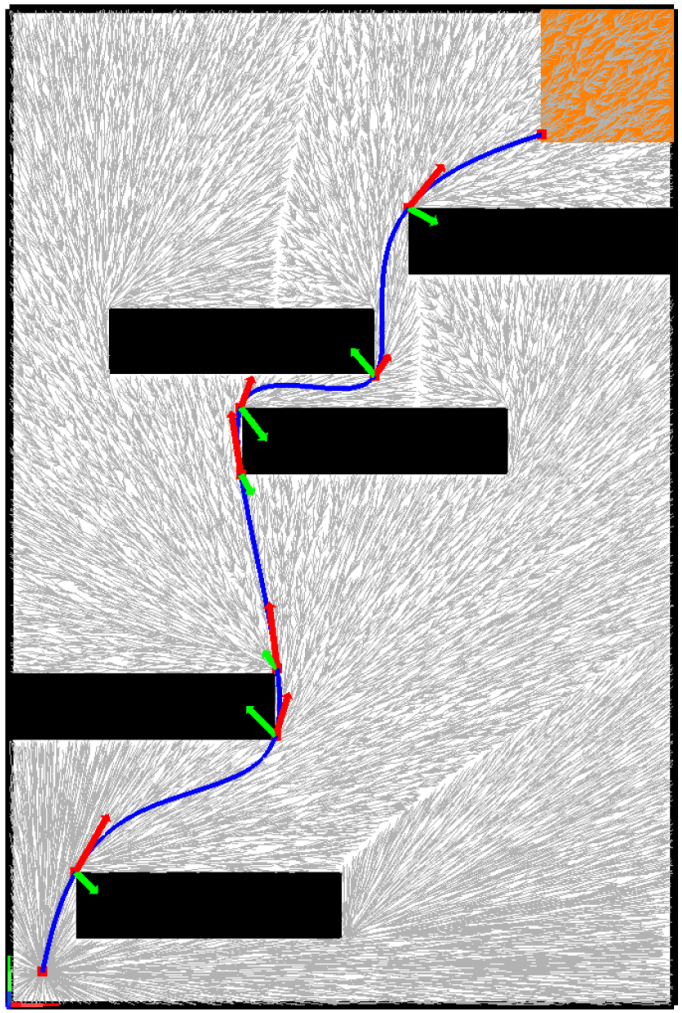
Position	$\boldsymbol{r}, \dot{\boldsymbol{r}}, \ddot{\boldsymbol{r}}, \dddot{\boldsymbol{r}}, \boldsymbol{r}^{(4)}$	$\Leftrightarrow$	$\boldsymbol{r}, \dot{\boldsymbol{r}}, {}^{W}R_B, \dot{\boldsymbol{\omega}}$	State vector
Yaw	$\psi, \dot{\psi}, \ddot{\psi}$		$T, \boldsymbol{\tau}$	Control inputs
	Flat outputs		**Trajectory and controls**	

- A sufficiently smooth path in space and yaw can be transformed to a set of dynamically feasible states and control inputs
- Transforms exist that include rotor drag
- Singularities
  1. Zero thrust – set minimum thrust limit
  2. $\pm 90°$ pitch – run alternate coordinates and switch near singularity

Existing planning algorithms

UNConstrained Optimiser (UNCO)

- Minimum snap trajectories
- Piecewise polynomial segments through predefined waypoints
- **Inner loop:** Unconstrained quadratic programming problem
- **Outer loop:** Segment time gradient descent
- Fast, smooth and robust
- No consideration of thrust/RPM constraints
- No treatment of obstacles in optimisation

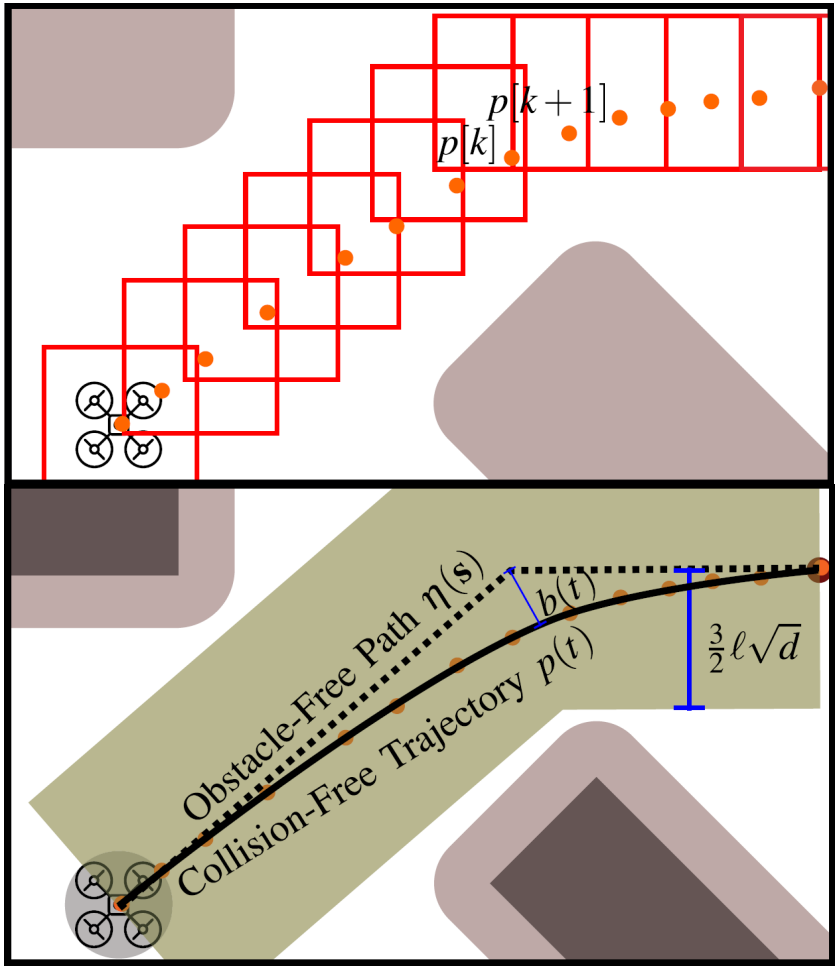


Bry et. al., 2015 [2]

Existing planning algorithms

Tube And Cube Optimisation (TACO)

- Similar formulation to UNCO
- Additional convex cube constraints between waypoints
- Convex optimisation problem
- “Slow and safe”
- No consideration of thrust/RPM constraints

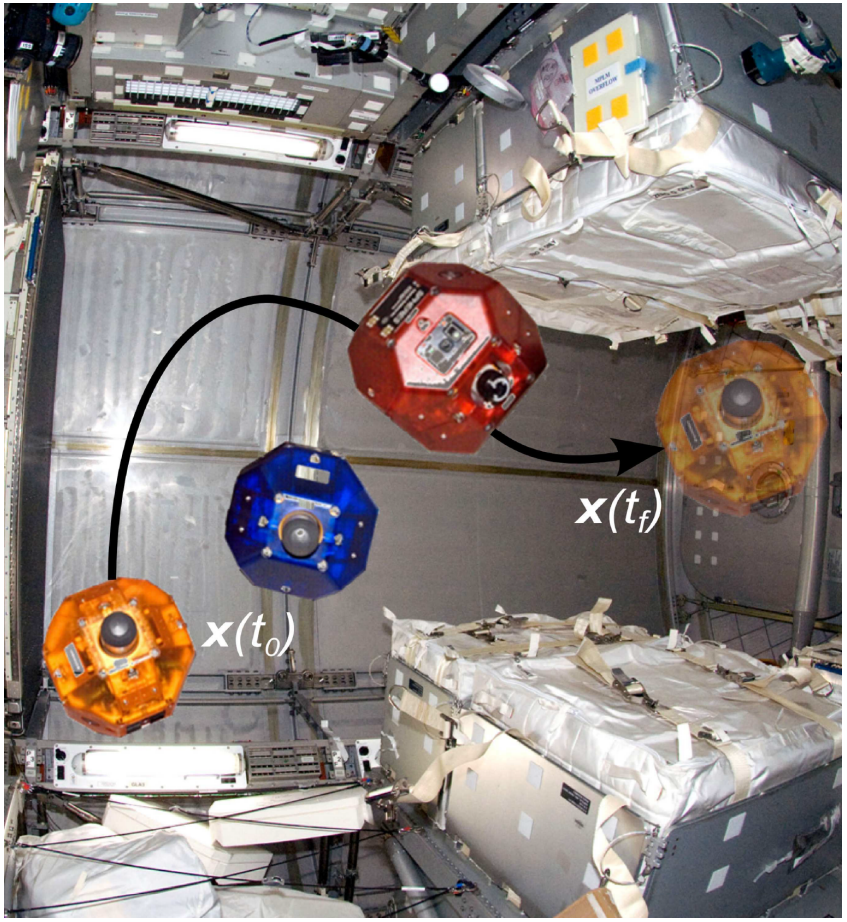


Campos-Macias et. al., 2017 [3]

Existing planning algorithms

Admissible Subspace TRajectory Optimiser (ASTRO)

- Optimisation of Legendre polynomials
- Optimisation in a subspace of coefficients that enforce boundary/continuity conditions
- Gradient descent approach
- Obstacle collisions treated as part of the cost function
- Ideal for cluttered environments



Morrell, 2018 [4]

Motivation

- Perform aggressive maneuvers with inverted flight
- Need consideration of min/max RPM limits in trajectory optimisation
- Minimum thrust constraint creates non-convex optimisation problem

Gavrilets et. al., 2004 [5]

Lossless convexification

Non-convex optimal control problem

$$\underset{t_f, T_c}{\text{minimise}} \quad \int_0^{t_f} \|T_c(t)\| \, \mathrm{d}t$$

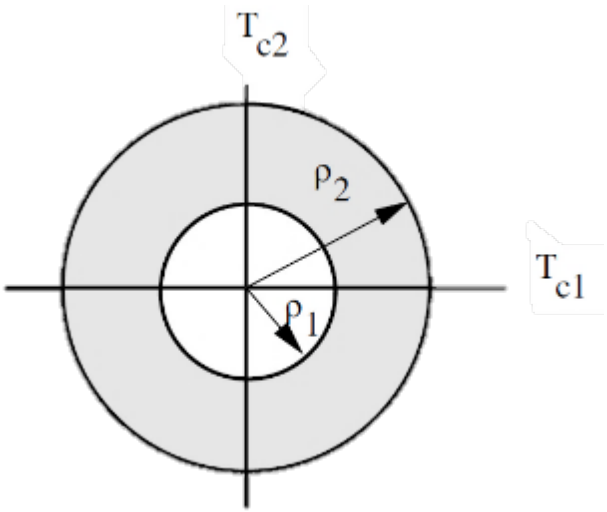
$$\text{such that} \quad 0 < \rho_1 \leq \|T_c(t)\| \leq \rho_2$$

dynamics are satisfied

boundary conditions are satisfied

other constraints are satisfied

- Designed for planetary landing
- Convex cost function
- Convex “other” constraints
- Non-convex thrust constraint



Açıkmeşe and Ploen, 2007 [6]

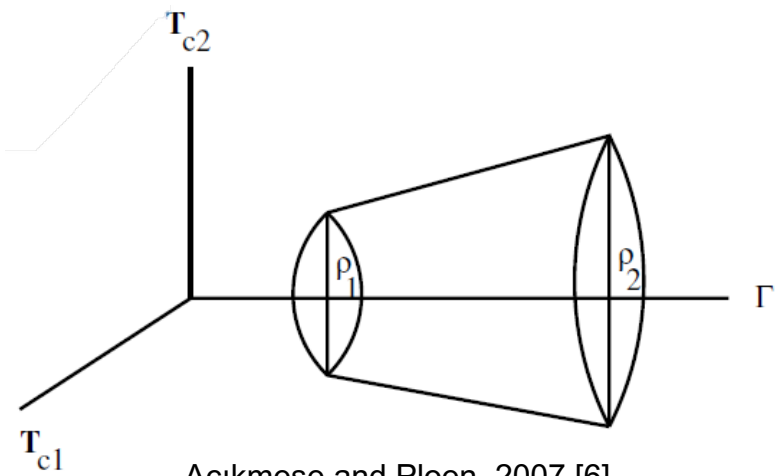
Lossless convexification

Convex optimal control problem

$$\underset{t_f, \Gamma, T_c}{\text{minimise}} \quad \int_0^{t_f} \Gamma(\mathrm{t})\, \mathrm{d}t$$

$$\begin{aligned}
\text{such that} \quad & 0 < \rho_1 \leq \Gamma(t) \leq \rho_2 \\
& \|T_c(t)\| \leq \Gamma(t) \\
& \text{dynamics are satisfied} \\
& \text{boundary conditions are satisfied} \\
& \text{other constraints are satisfied}
\end{aligned}$$

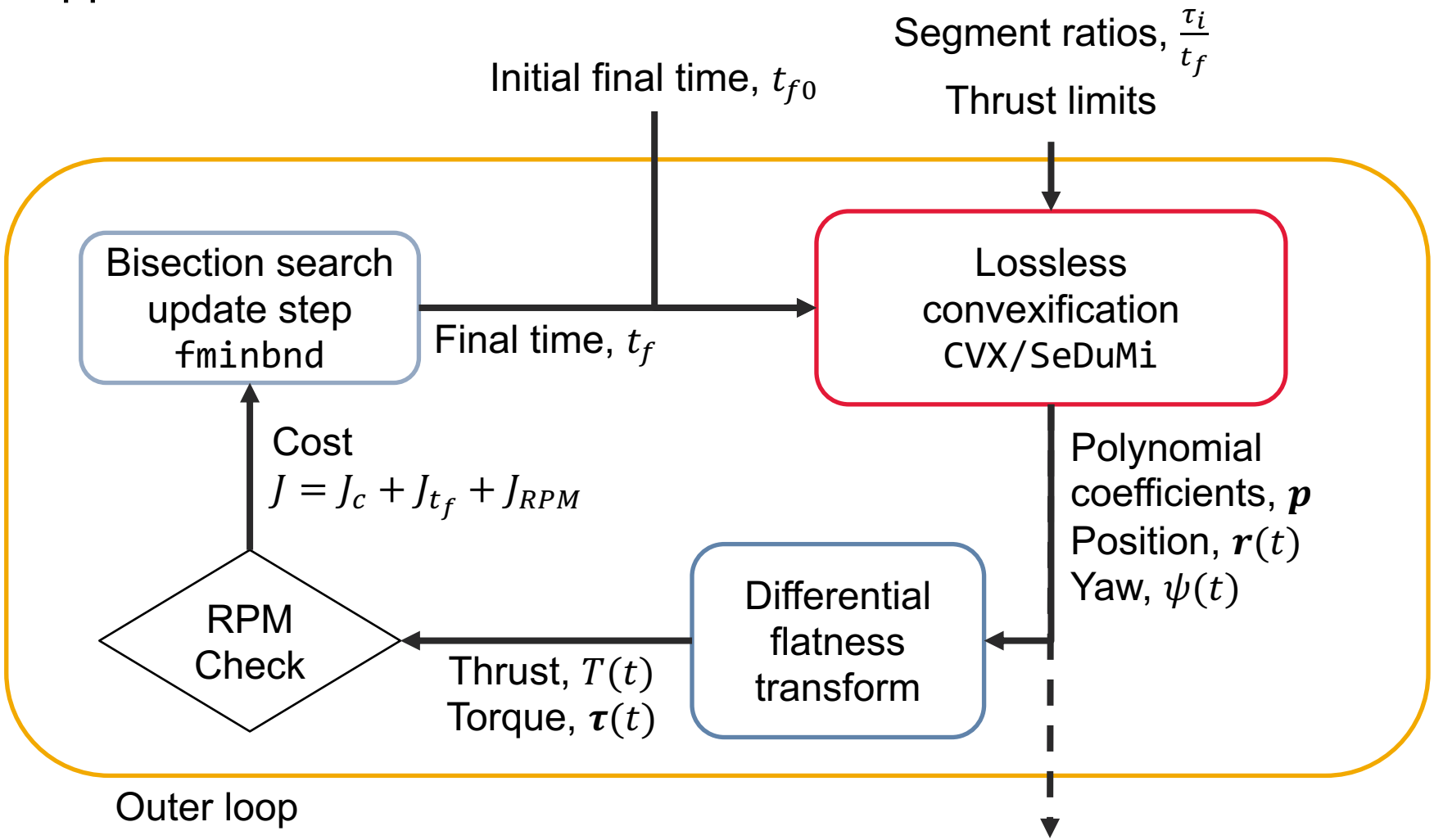
- Introduce slack variable $\Gamma(t)$
- Search space is now convex
- Can show $\|T_c^*(t)\| = \Gamma^*(t)$



Açıkmeşe and Ploen, 2007 [6]

Implementation

Approach



Implementation

Convex optimisation problem

$$\underset{\Gamma,\,\boldsymbol{x},\,\psi}{\text{minimise}} \qquad \int_0^{t_f} c_\Gamma \Gamma^2(\mathrm{t}) + c_s \left\| \boldsymbol{x}^{(4)}(t) \right\|^2 \mathrm{d}t$$

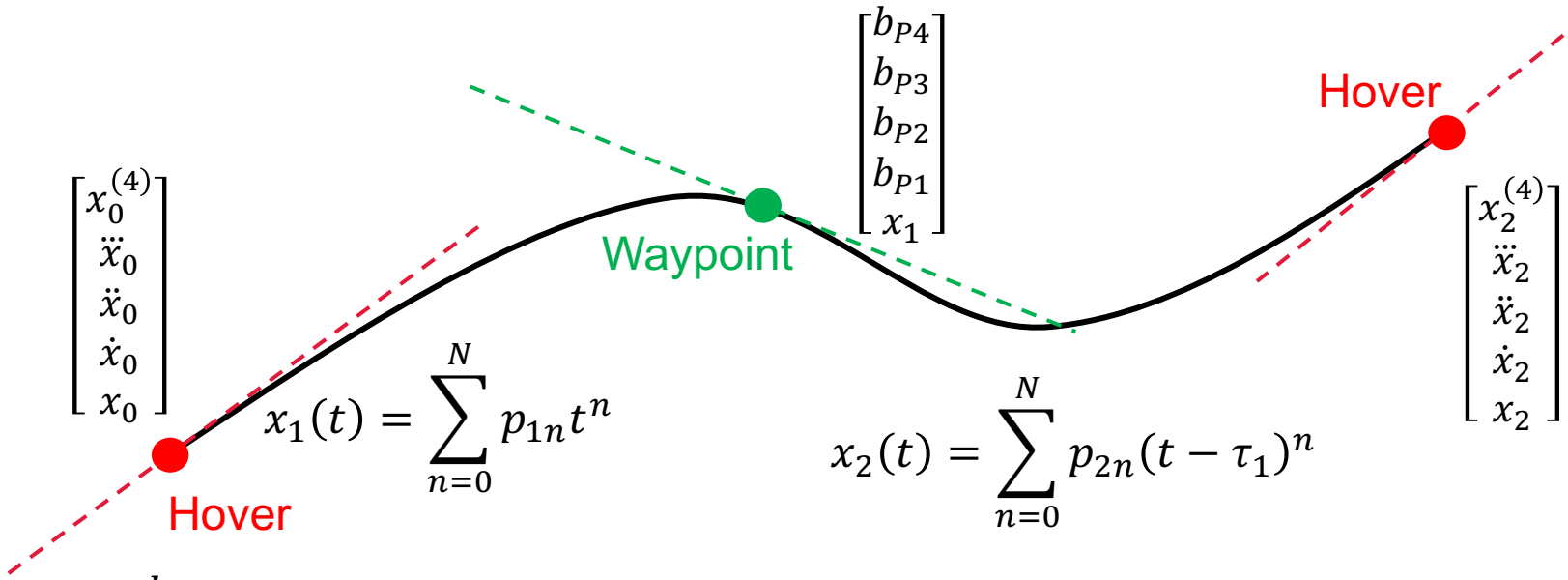
$$\text{such that} \qquad 0 < \left(\frac{\alpha T_{\min}}{m}\right)^2 \leq \Gamma(t) \leq \left(\frac{T_{\max}}{m}\right)^2$$

$$\ddot{x}^2 + \ddot{y}^2 + (\ddot{z} + g)^2 \leq \Gamma(t)$$

- Optimal trajectory: $\ddot{x}^2 + \ddot{y}^2 + (\ddot{z} + g)^2 = \Gamma(t)$
- Minimum thrust and snap trajectory
- Lossless convexification enforces thrust constraint
- Multiple segments with free or fixed boundary conditions
- Solved with CVX/SeDuMi

Implementation

Derivative optimisation – inner loop



$$\boldsymbol{b}_P = \begin{bmatrix} b_{P4} \\ b_{P3} \\ b_{P2} \\ b_{P1} \end{bmatrix} \qquad \boldsymbol{b}_F = \begin{bmatrix} \boldsymbol{x}_0 \\ x_1 \\ \boldsymbol{x}_2 \end{bmatrix} \qquad \boldsymbol{b} = C \begin{bmatrix} \boldsymbol{b}_F \\ \boldsymbol{b}_P \end{bmatrix} \qquad A\boldsymbol{p} = \begin{bmatrix} \boldsymbol{b} \\ 0 \end{bmatrix}$$

Free derivatives | Fixed derivatives | Permutation | Mapping/continuity

$$J = \min_{\boldsymbol{p}} \boldsymbol{p}^T (c_s Q_s + c_\Gamma Q_\Gamma) \boldsymbol{p} = \min_{\boldsymbol{b}_P} \begin{bmatrix} \boldsymbol{b}_F \\ \boldsymbol{b}_P \end{bmatrix}^T C A^{-T} (c_s Q_s + c_\Gamma Q_\Gamma) A^{-1} C^T \begin{bmatrix} \boldsymbol{b}_F \\ \boldsymbol{b}_P \end{bmatrix}$$

Implementation

Thrust mixing and saturation

Static:

$$\begin{bmatrix} T \\ \tau_x \\ \tau_y \\ \tau_z \end{bmatrix} = \begin{bmatrix} c_{T1} & c_{T2} & c_{T3} & c_{T4} \\ -c_{T1}l & c_{T2}l & c_{T3}l & -c_{T4}l \\ -c_{T1}l & c_{T2}l & -c_{T3}l & c_{T4}l \\ -c_q & -c_q & c_q & c_q \end{bmatrix} \begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{bmatrix}$$

Dynamic:

$$T_i = k_1\omega^2 + k_2\omega(v_z + v_i)$$

$$T_i = k_4 v_i \sqrt{(v_z + v_i)^2 + v_{xy}^2}$$

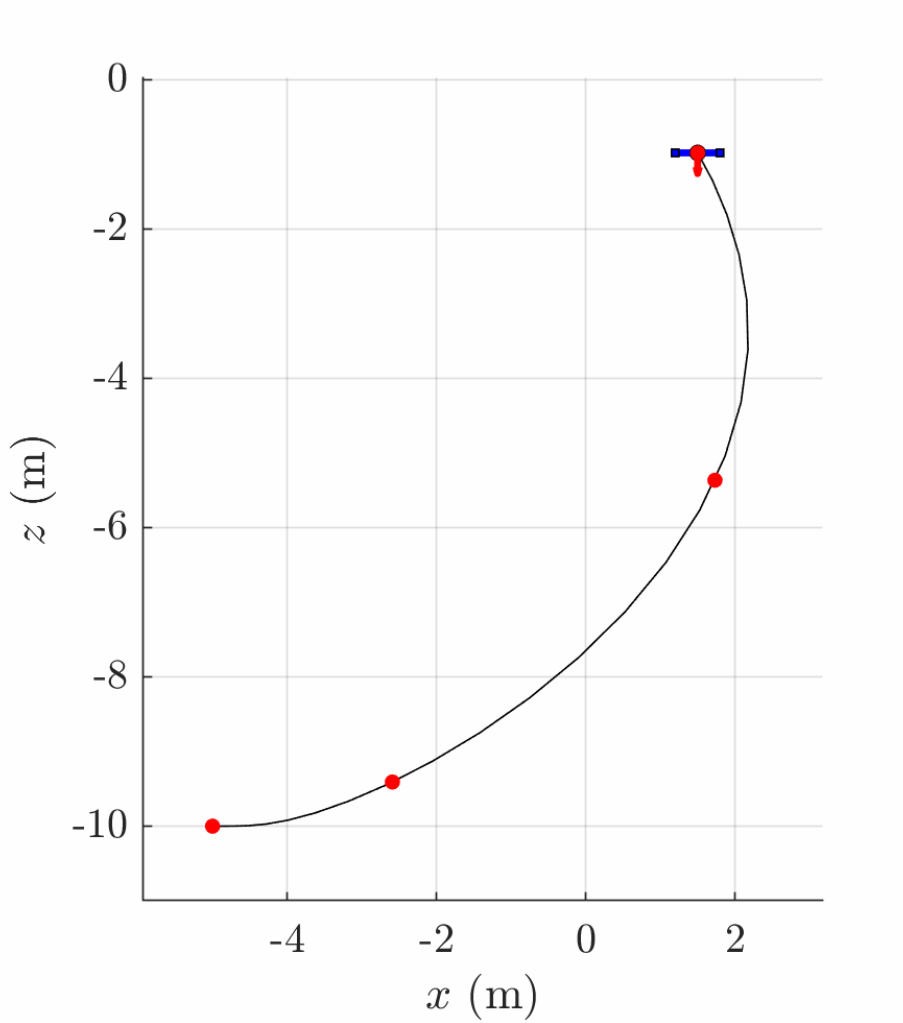
$$c_{Ti} = \frac{T_i}{\omega^2}$$

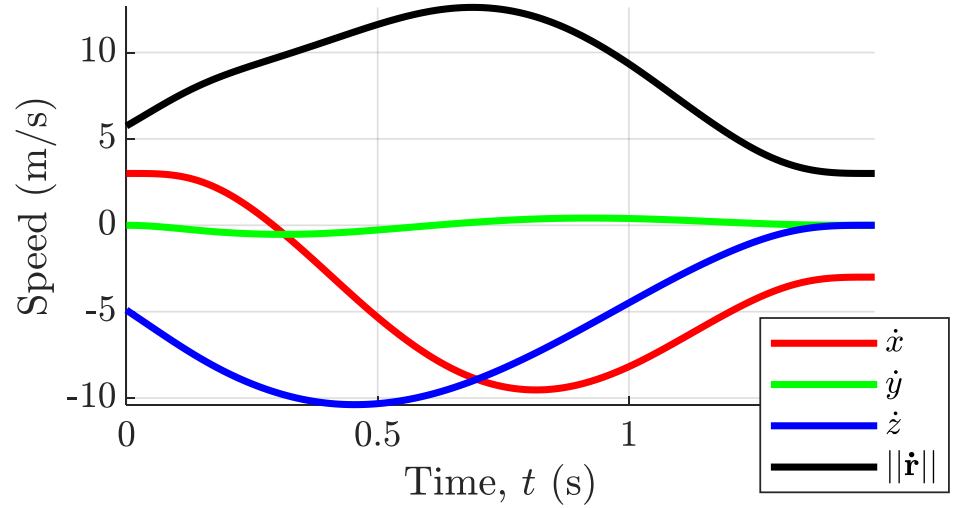
- Saturation sacrifices yaw control to meet RPM limits

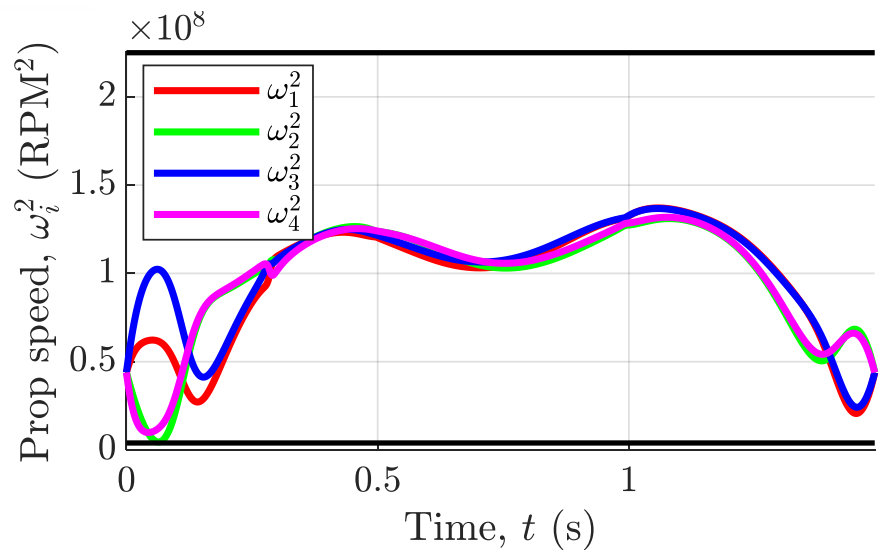
Results

Half flip – Min-snap formulation

$c_\Gamma = 1$, $c_s = 1$: Feasible



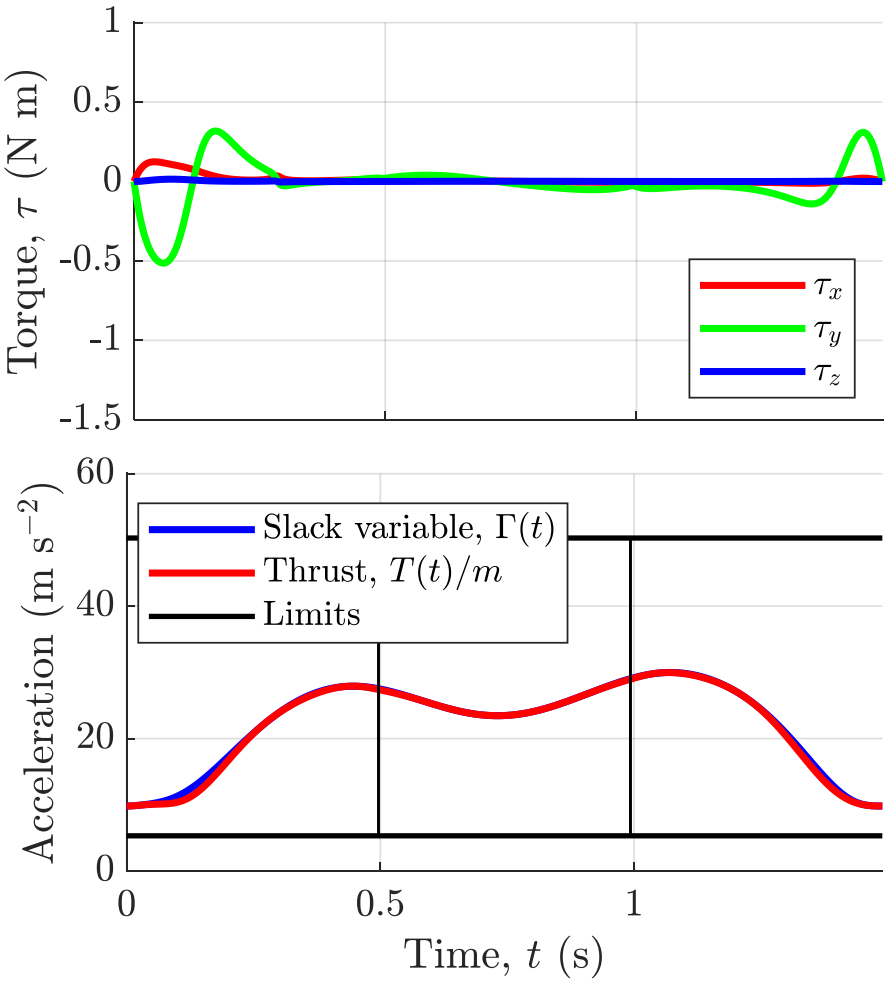




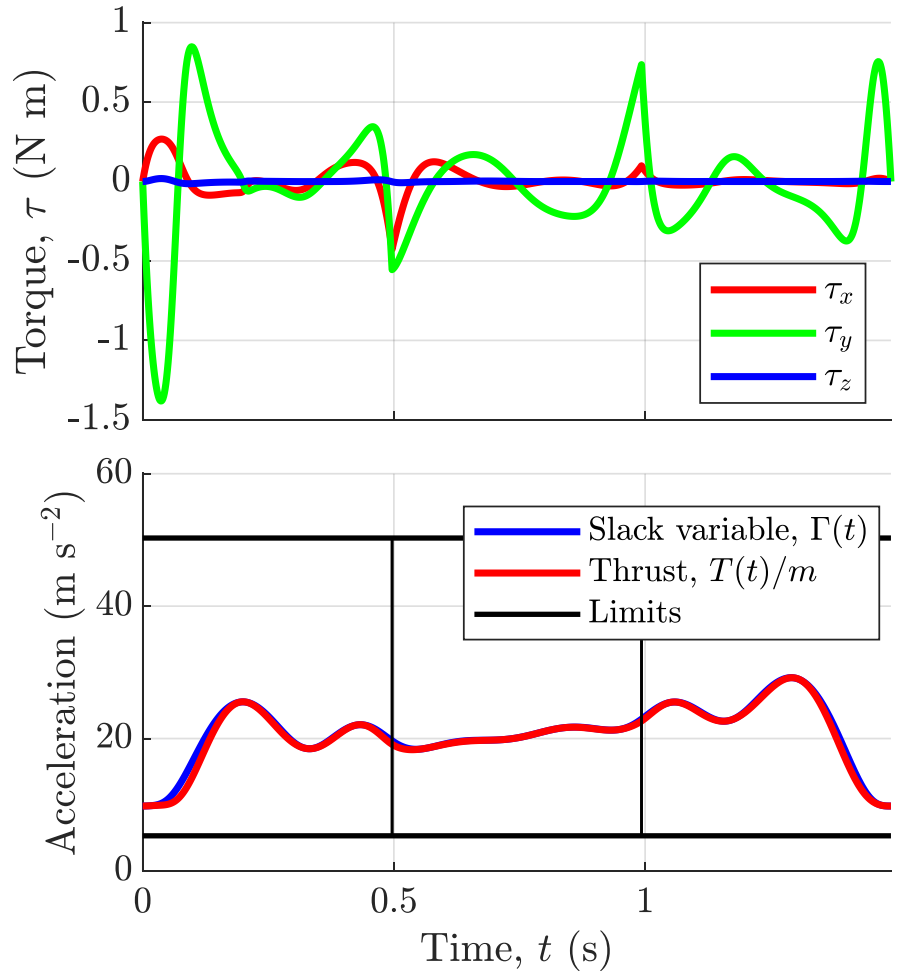
Results

Half flip – Min-snap formulation

$c_\Gamma = 1$, $c_s = 1$: Feasible



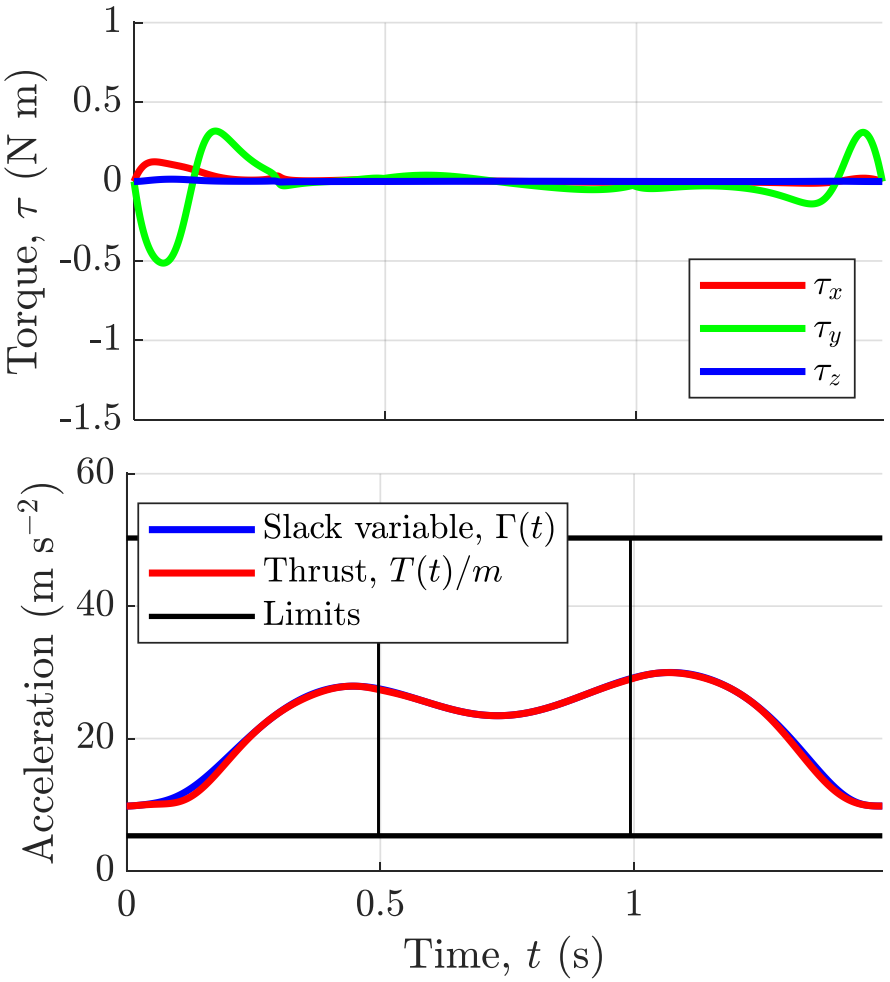
$c_\Gamma = 1$, $c_s = 0$: Infeasible



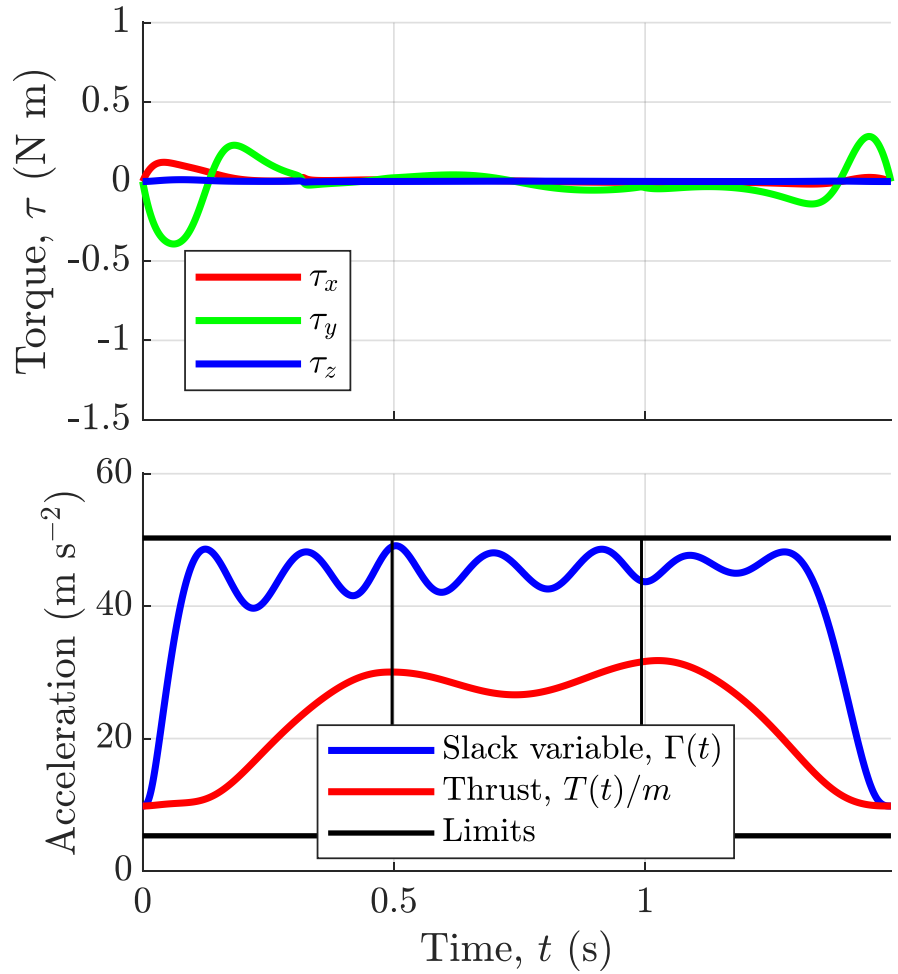
Results

Half flip – Min-snap formulation

$c_\Gamma = 1$, $c_s = 1$: Feasible

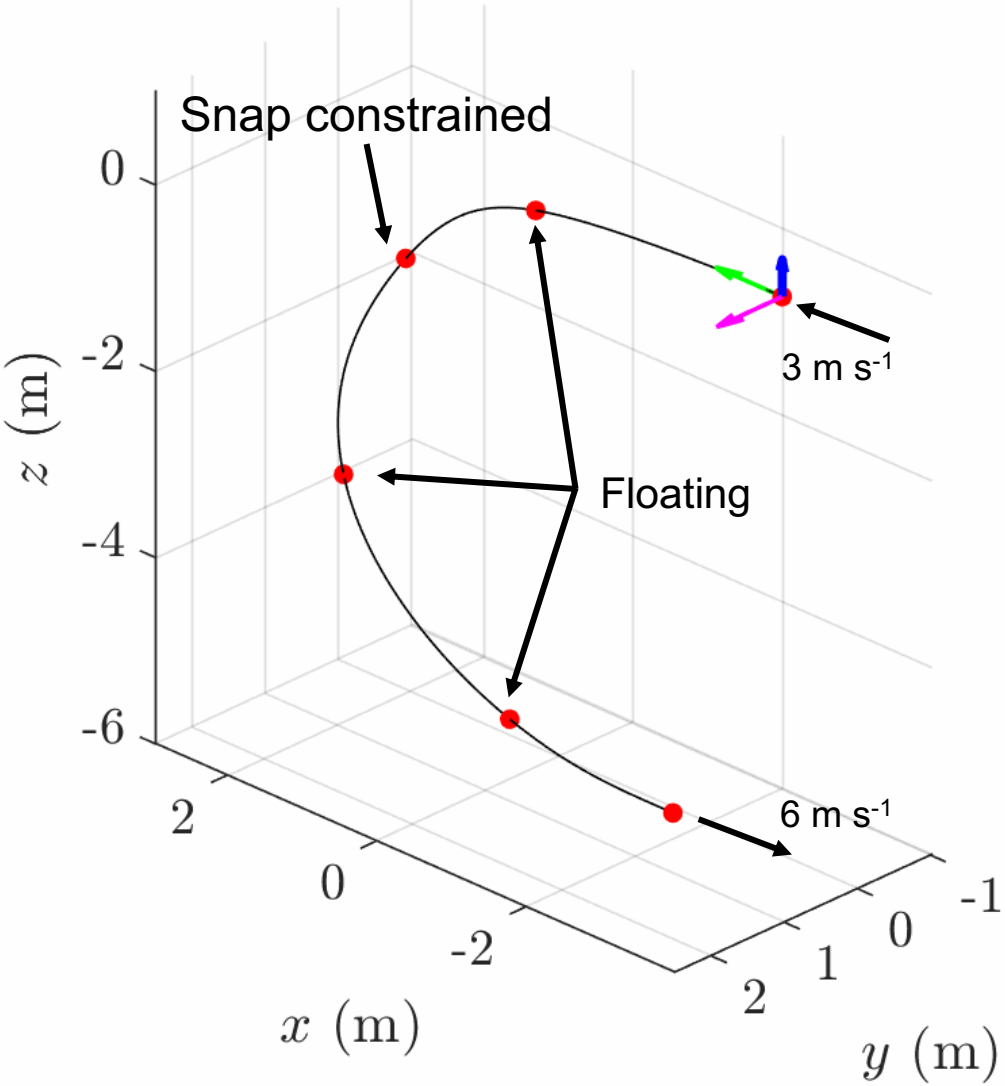


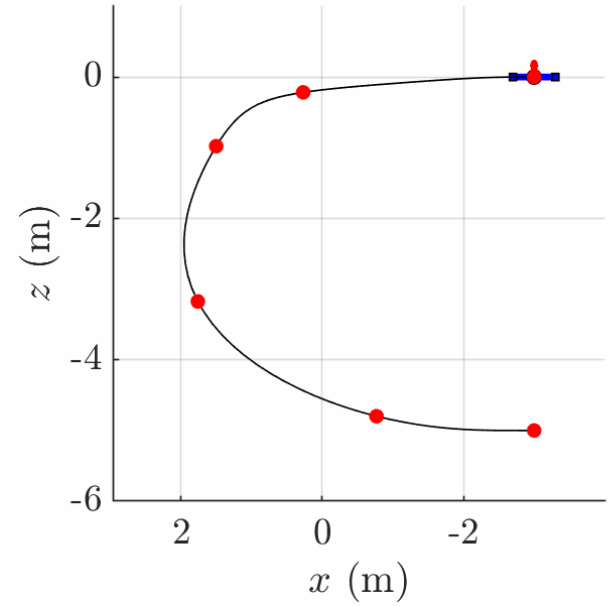
$c_\Gamma = 0$, $c_s = 1$: Feasible, not converged

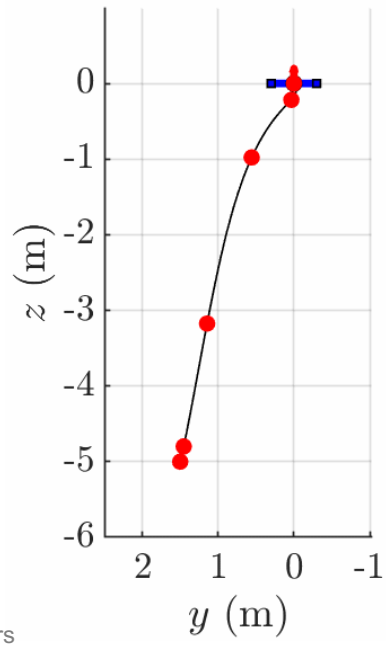


Results

Constrained Split-S

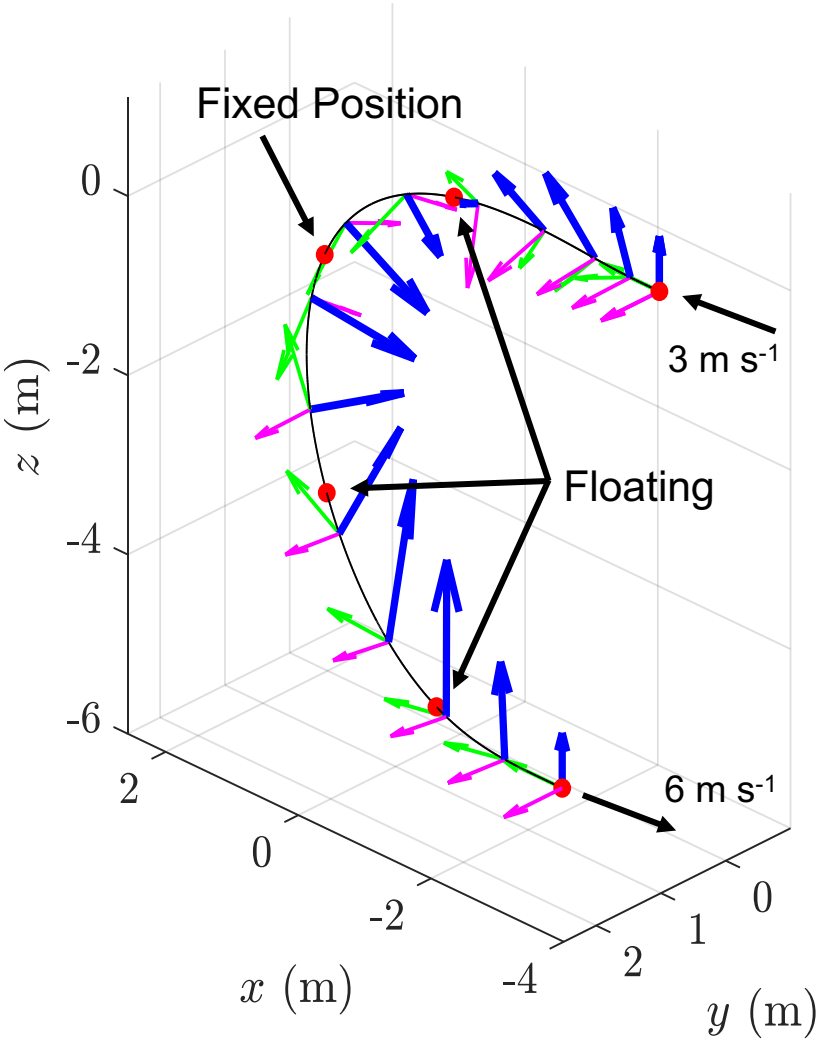


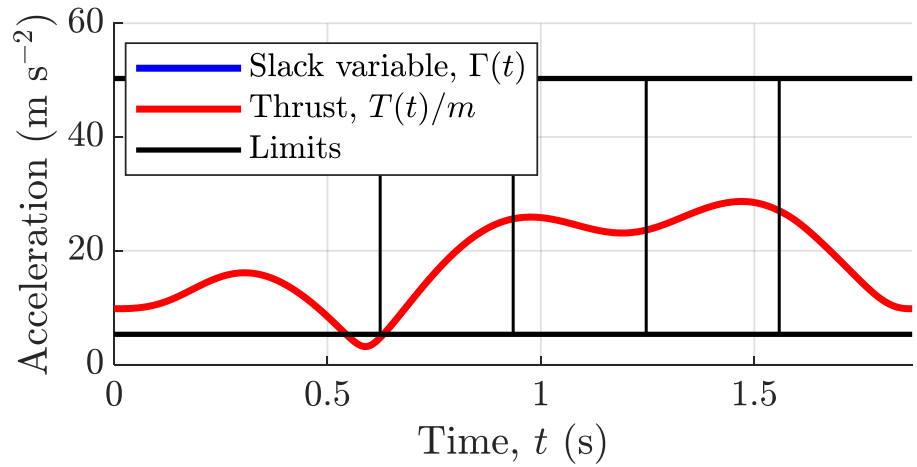


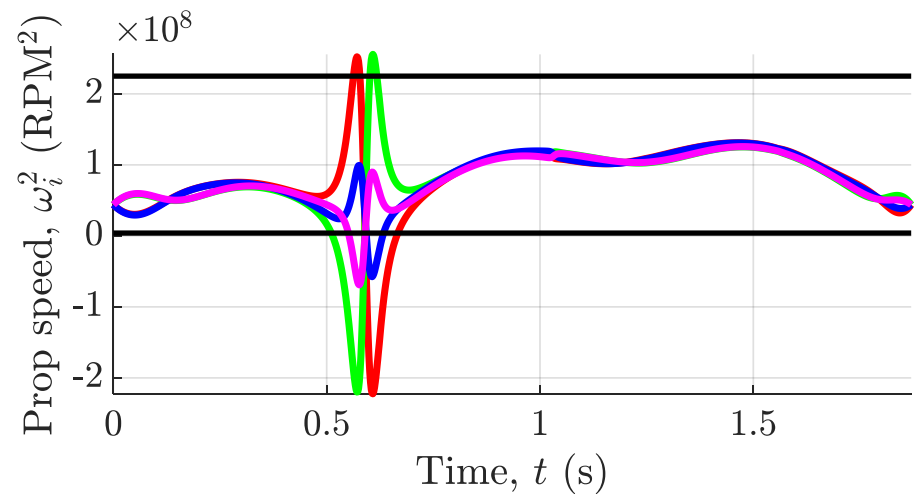


Results

Split-S - UNCO

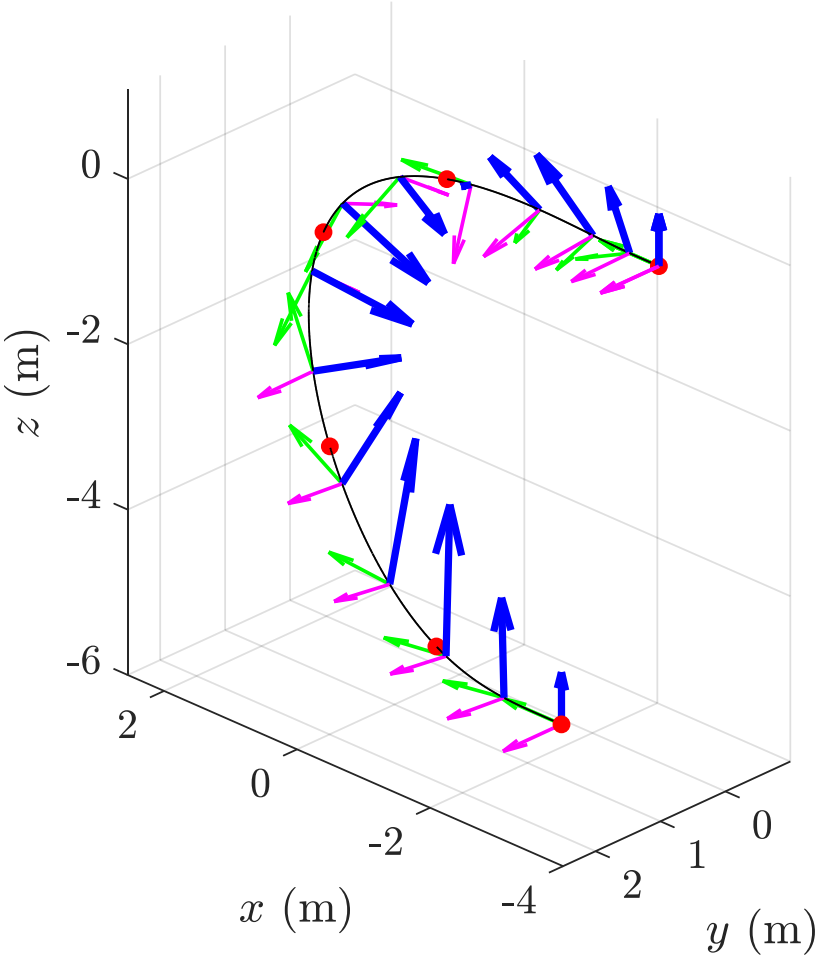


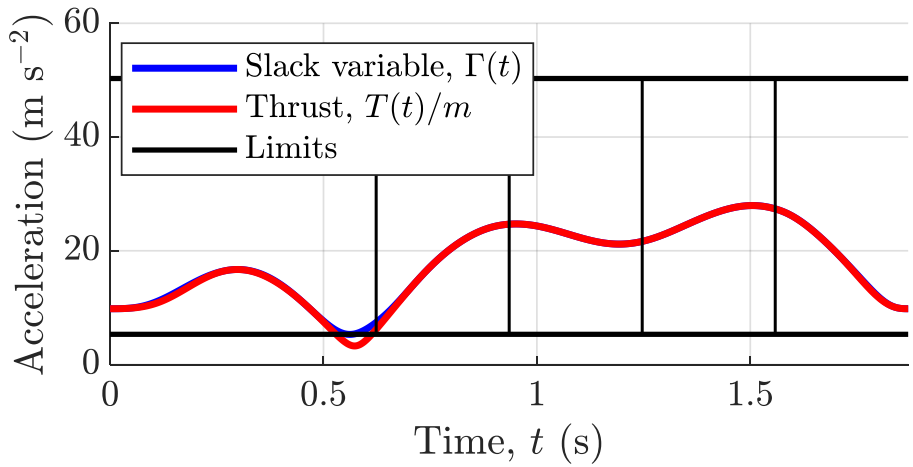


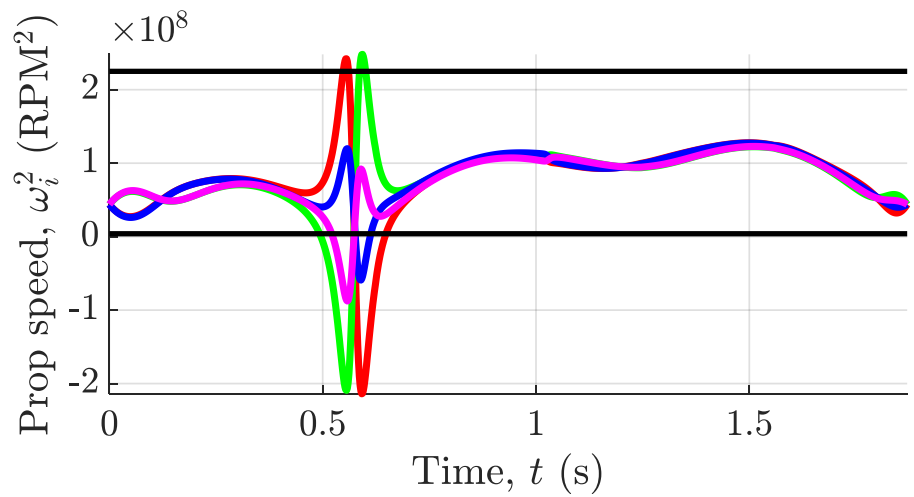


Results

Split-S – Lossless convexification – Low slack weighting

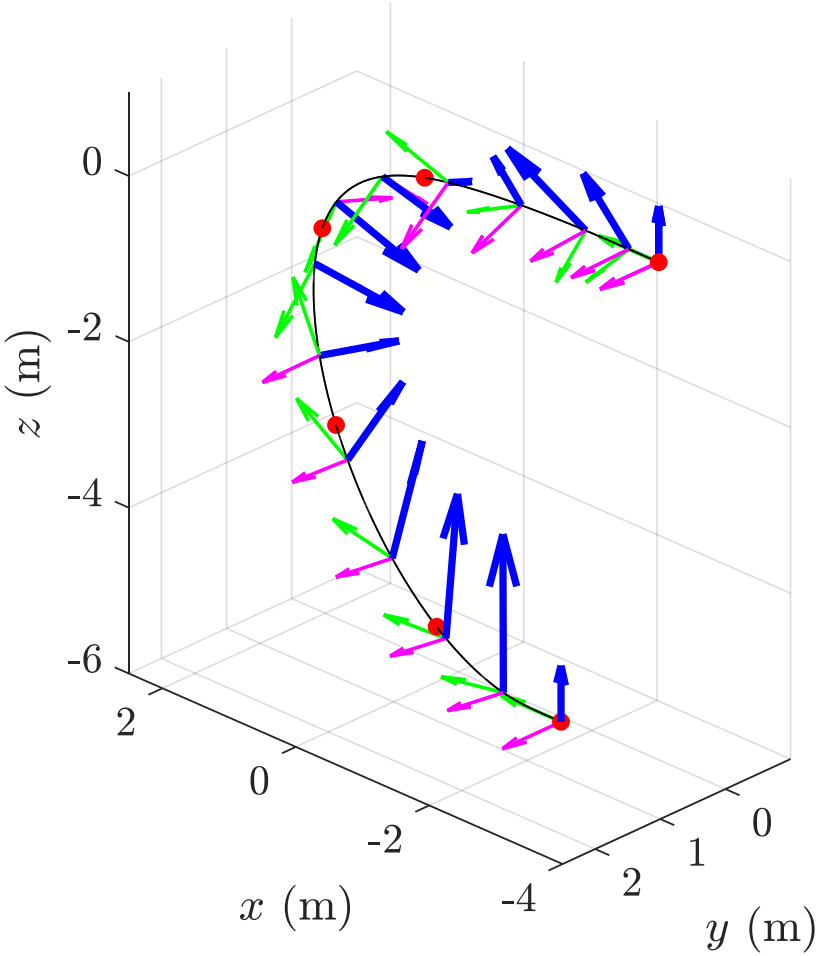


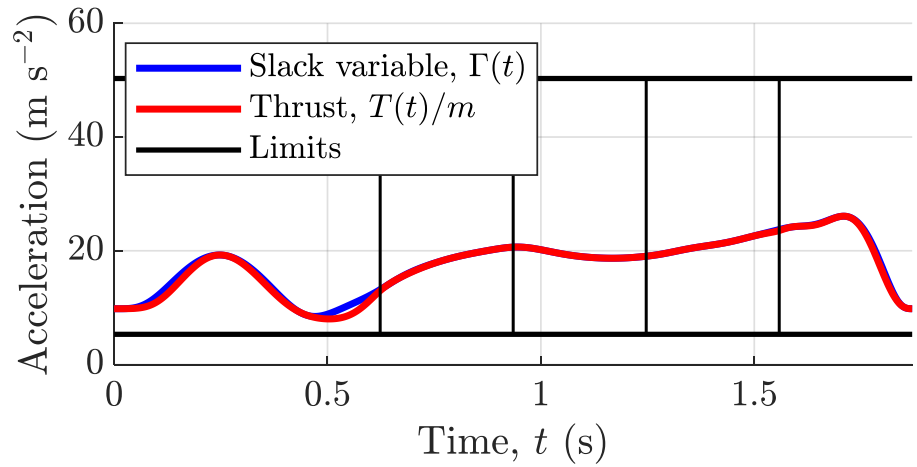


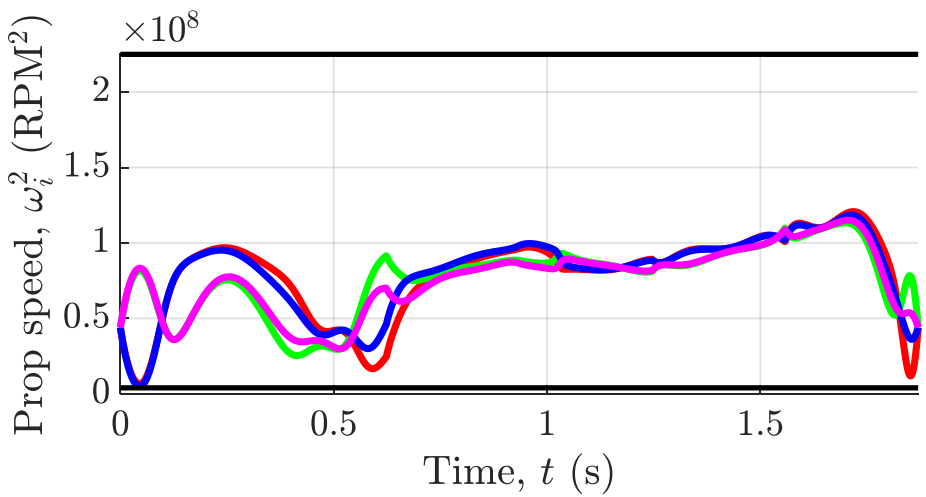


Results

Split-S – Lossless convexification – High slack weighting

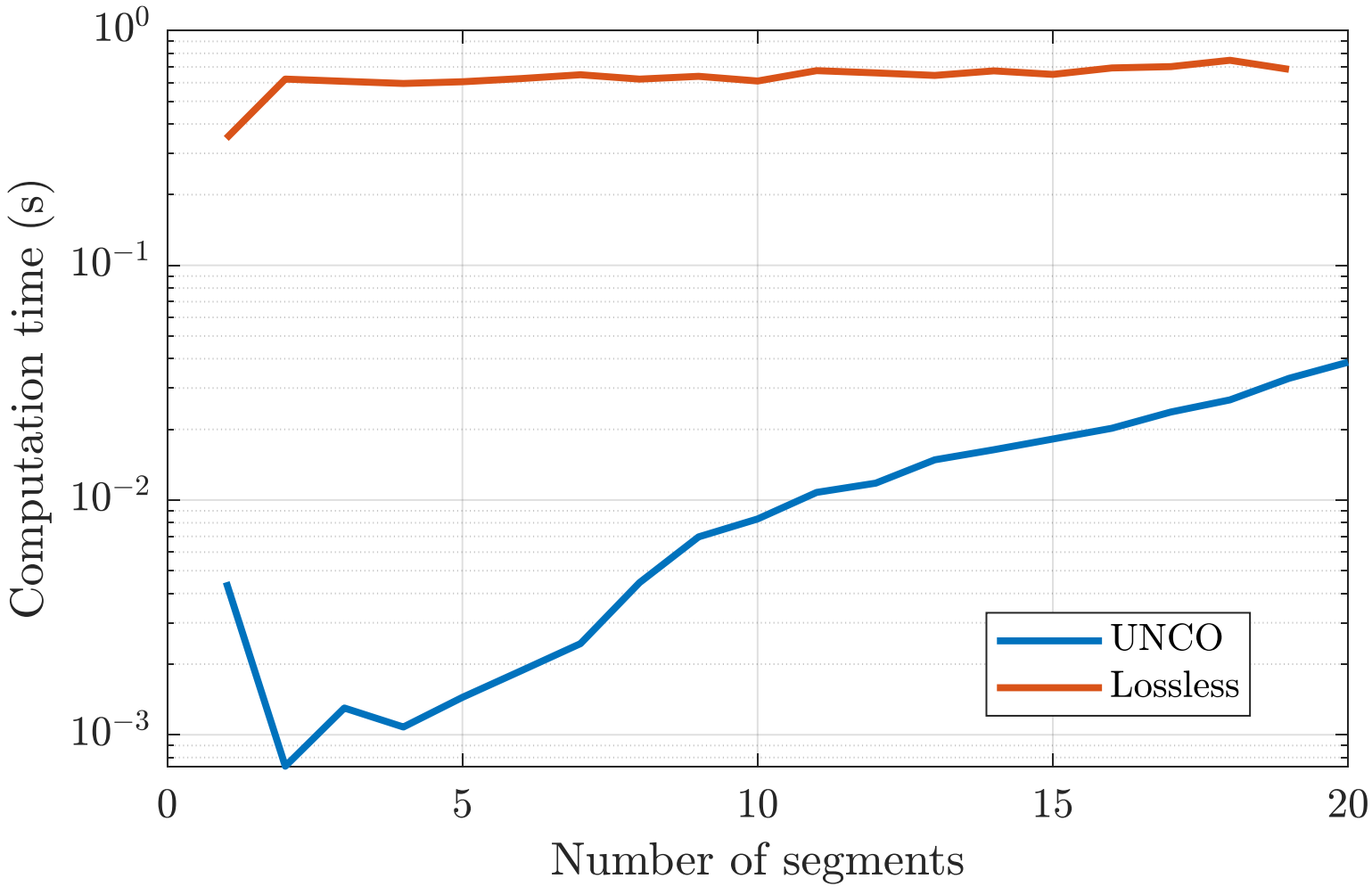






Results

Timing



Key outcomes

- Flexible Matlab implementation of unconstrained and lossless convexification polynomial trajectory optimisation routines
- Lossless convexification is promising as an efficient optimisation routine that accounts for input feasibility
- Combined minimum snap and minimum control input trajectories successful
- A Split-S appears to be “human optimal”

Next steps

- Hardware tests with upgraded flight stack
- Segment time optimisation (in progress)
- Combined algorithm
  1. Try solution with UNCO
  2. If non-feasible attempt lossless convexification
- Obstacle avoidance constraints
  - Iterative Regional Inflation by Semi-definite programming (IRIS)
  - TACO cubic constraints
  - Cylindrical regions
  - Mixed integer programming

Jet Propulsion Laboratory
California Institute of Technology

jpl.nasa.gov

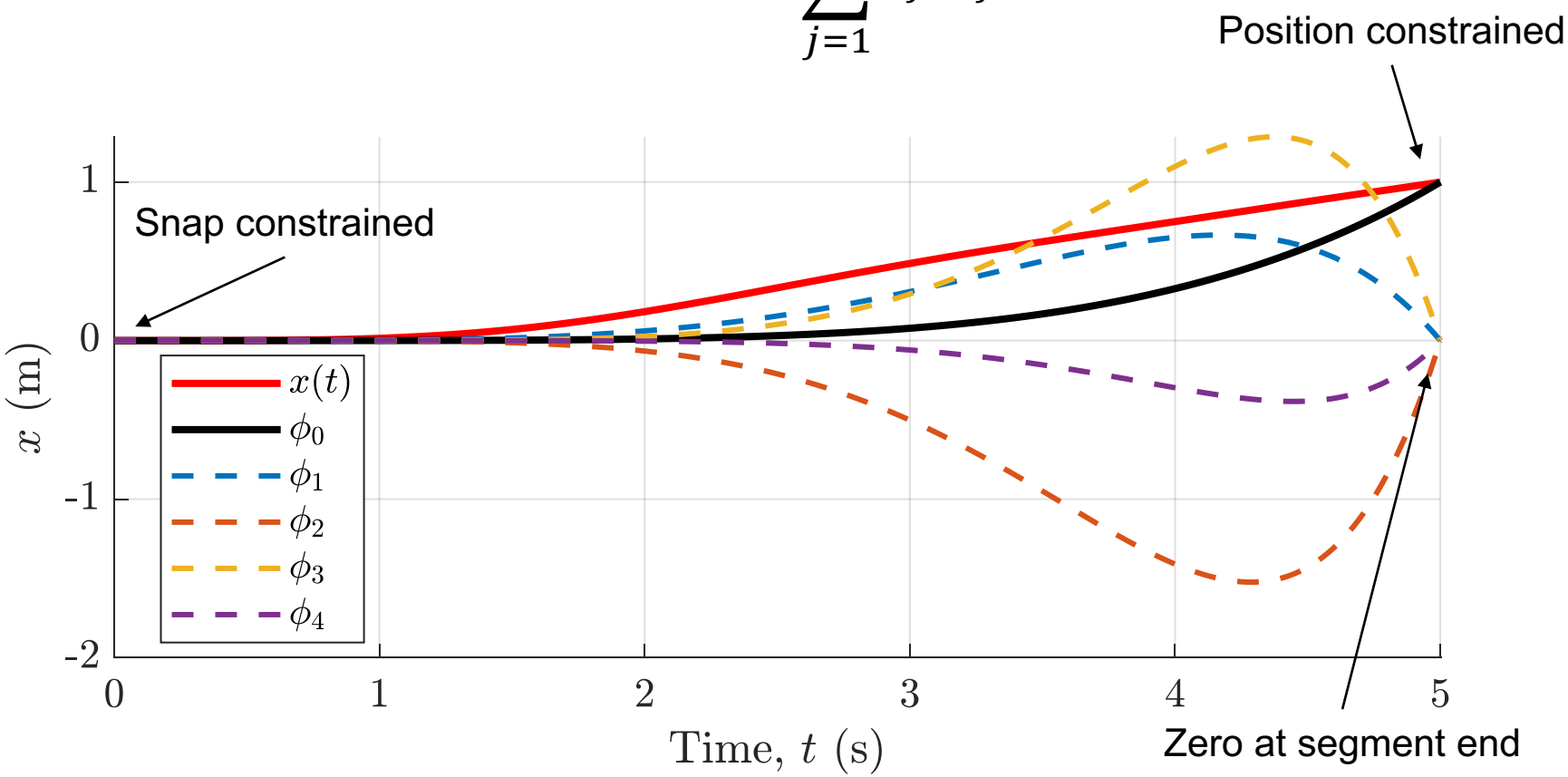
References

1. Marc Rigter. High performance quadrotor system identification and control. Undergraduate Honours Thesis, The University of Sydney, 1 2018.
2. Adam Bry, Charles Richter, Abraham Bachrach, and Nicholas Roy. Aggressive flight of fixed-wing and quadrotor aircraft in dense indoor environments. *The International Journal of Robotics Research*, 34(7):969--1002, mar 2015.
3. Leobardo Campos-Macias, David Gomez-Gutierrez, Rodrigo Aldana-Lopez, Rafael de la Guardia, and Jose I. Parra-Vilchis. A hybrid method for online trajectory planning of mobile robots in cluttered environments. *IEEE Robotics and Automation Letters*, 2(2):935--942, apr 2017.
4. Benjamin J. Morrell. *Enhancing 3D autonomous navigation through obstacle fields*. PhD thesis, The University of Sydney, 5 2018.
5. V. Gavrilets, Bernard Mettler, and E. Feron. Human-inspired control logic for automated maneuvering of miniature helicopter. *Journal of Guidance, Control, and Dynamics*, 27(5):752--759, sep 2004.
6. Behcet Acikmese and Scott R. Ploen. Convex programming approach to powered descent guidance for mars landing. *Journal of Guidance, Control, and Dynamics*, 30(5):1353--1366, sep 2007.
7. Benjamin Morrell, Marc Rigter, Gene Merewether, Robert Reid, Rohan Thakker, Theodore Tzanetos, Vinay Rajur, and Gregory Chamitoff. Differential flatness transformations for aggressive quadrotor flight. *TBD*, 1(1):0--0, TBD 2018.

Implementation

Polynomial form

$$\sigma_l(t) = \phi_{0l}(t) + \sum_{j=1}^{k} a_{jl}\phi_{jl}(t)$$



Implementation

Discretisation – inner loop

$$\underset{\boldsymbol{a}}{\text{minimise}} \quad \sum_{i=1}^{N} (A_\Gamma \boldsymbol{a})_i$$

$$\text{such that} \quad \left(\frac{\alpha T_{\min}}{m}\right)^2 \leq A_\Gamma \boldsymbol{a} + \boldsymbol{\Gamma_0} \leq \left(\frac{T_{\max}}{m}\right)^2$$

$$(A_{\ddot{x}} \boldsymbol{a} + \ddot{\boldsymbol{x}}_0)^2 + (A_{\ddot{y}} \boldsymbol{a} + \ddot{\boldsymbol{y}}_0)^2 + (A_{\ddot{z}} \boldsymbol{a} + \ddot{\boldsymbol{z}}_0)^2 \leq \Gamma(t)$$

$$A_P \boldsymbol{a} = \boldsymbol{b}_{P0}$$

$$\text{where} \quad \sigma_l(t_i) = \phi_{0l}(t_i) + \sum_{j=1}^{k} a_{jl} \phi_{jl}(t_i) = \boldsymbol{\sigma_{0i}}(t_i) + A_{\sigma i}(t_i) \boldsymbol{a}$$

Implementation

Polynomial forms

$$\sigma_l(t) = \phi_{0l}(t) + \sum_{j=1}^{k} a_{jl}\phi_{jl}(t)$$

$$\phi_{0l}(t) = \sum_{n=0}^{p_{l-1}} b_n (t-\xi_{l-1})^n + \sum_{n=0}^{p_l} b_{p_{l-1}+1+n} (t-\xi_{l-1})^{p_{l-1}+1} (t-\xi_l)^n$$

$$\phi_{jl}(t) = t^{j-1}(t-\xi_{l-1})^{p_{l-1}+1}(t-\xi_l)^{p_l} \text{ for } j = 1, \dots, k$$

$$k = K - (p_{l-1} + p_l)$$

$$\xi_l = \sum_{i}^{l} \tau_i$$

Implementation

Derivative optimisation – inner loop

$$\underset{\boldsymbol{b}_P}{\text{minimise}} \quad J = \begin{bmatrix} \boldsymbol{b}_F \\ \boldsymbol{b}_P \end{bmatrix}^T C A^{-T} (c_s Q_s + c_\Gamma Q_\Gamma) A^{-1} C^T \begin{bmatrix} \boldsymbol{b}_F \\ \boldsymbol{b}_P \end{bmatrix}$$

$$\text{such that} \quad \left(\frac{\alpha T_{\min}}{m}\right)^2 \leq A_\Gamma \boldsymbol{b}_P + \boldsymbol{\Gamma_0} \leq \left(\frac{T_{\max}}{m}\right)^2$$

$$(A_{\ddot{x}} \boldsymbol{b}_P + \ddot{\boldsymbol{x}}_0)^2 + (A_{\ddot{y}} \boldsymbol{b}_P + \ddot{\boldsymbol{y}}_0)^2 + (A_{\ddot{z}} \boldsymbol{b}_P + \ddot{\boldsymbol{z}}_0)^2 \leq A_\Gamma \boldsymbol{b}_P + \boldsymbol{\Gamma_0}$$

$$\text{where} \quad \sigma_l(t) = \sum_{n=0}^{N} p_{ln} \left(t - \sum_{0}^{l} \tau_i \right)^n$$

Polynomial form

$$A\boldsymbol{p} = \begin{bmatrix} C \begin{bmatrix} \boldsymbol{b}_F \\ \boldsymbol{b}_P \end{bmatrix} \\ 0 \end{bmatrix}$$

Derivative/coefficient mapping and continuity

Differential flatness

Overview

For a *differentially flat* system:

$$\dot{x} = f(x, u)$$

we can define a set of *flat outputs* $\sigma(t)$ such that:

$$x(t) = x\left(\sigma, \dot{\sigma}, \ddot{\sigma}, \dots, \sigma^{(p)}\right)$$
$$u(t) = u\left(\sigma, \dot{\sigma}, \ddot{\sigma}, \dots, \sigma^{(p)}\right)$$

- We can express states and controls as a function of the flat outputs and derivatives
- Any p times differentiable trajectory in the flat outputs can be transformed into a *dynamically feasible* trajectory and controls
- Allows planning in flat output space

Existing planning algorithms

Minimum snap polynomial trajectories

Algorithm	Approach	Obstacles	Use case
UNConstrained Optimiser (UNCO)	Direct matrix solution given segment times Outer loop gradient descent for time	No explicit treatment	Fast, smooth trajectories
Tube And Cube Optimisation (TACO)	Similar to UNCO	Convex cube constraints in fee space	Slow and safe
Admissible Subspace TRajectory Optimiser (ASTRO)	Optimisation of space of polynomial coefficients that enforce BCs/continuity	Incorporated in cost function	Cluttered environments

Implementation

6-axis method

Morrell et. al., 2018 [7]