Optimization Method & Optimal Guidance

Haichao Hong
AE8120

Contents

Sequential Convex Optimization Methods with Control Parameterization

Sequential Convex Optimization Methods

1. Single Shooting, forward Euler

2. Collocation, forward Euler

3. Collocation, trapezoidal rule

Sequential Convex Optimization Methods

Defining the total coefficient vector as

$$\boldsymbol{c} = \left[\left(\boldsymbol{c}^{(1)} \right)^{\mathrm{T}}, \left(\boldsymbol{c}^{(2)} \right)^{\mathrm{T}}, \ldots, \left(\boldsymbol{c}^{(m)} \right)^{\mathrm{T}} \right]^{\mathrm{T}},$$

and the basis matrix as

$$\boldsymbol{S}_N (t) = \begin{bmatrix} \boldsymbol{s}_N^{(1)} (t) & 0 & \ldots & 0 \\ 0 & \boldsymbol{s}_N^{(2)} (t) & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & \boldsymbol{s}_N^{(m)} (t) \end{bmatrix},$$

The parameterization can be written in a more compact manner as

$$\boldsymbol{u} (t) = \boldsymbol{S}_N (t) \, \boldsymbol{c} \, .$$

1. Single Shooting, forward Euler

$$d\boldsymbol{x}_{k+1} = \sum_{j=1}^{k} \boldsymbol{B}_{k,j} d\boldsymbol{u}_j \, .$$

Here, it can be noticed that $\boldsymbol{u}_k$ is a linear function of $\boldsymbol{c}$, so the increment of the control at any step k, $d\boldsymbol{u}_k$, can be expressed as the increment of the coefficients

$$d\boldsymbol{u}_k = \Big[\boldsymbol{S}_N\Big]_k d\boldsymbol{c} \, .$$

Noticing that $\boldsymbol{c}$ is time-invariant,

$$
\begin{aligned}
d\boldsymbol{x}_{k+1} &= \sum_{j=1}^{k} \boldsymbol{B}_{k,j} \left[\boldsymbol{S}_N\right]_j d\boldsymbol{c} \\
&= \left(\sum_{j=1}^{k} \boldsymbol{D}_{k,j}\right) d\boldsymbol{c}
\end{aligned}
$$

Now the state increments are functions of the coefficient correction!

The initial and terminal control constraints are realized as

$$
\begin{aligned}
d\boldsymbol{u}_1 &= \left[\boldsymbol{S}_N\right]_1 d\boldsymbol{c} = \boldsymbol{u}_i - \boldsymbol{u}_1^p, \\
d\boldsymbol{u}_{N_t-1} &= \left[\boldsymbol{S}_N\right]_{N_t-1} d\boldsymbol{c} = \boldsymbol{u}_f - \boldsymbol{u}_{N_t-1}^p.
\end{aligned}
$$

The zero initial and terminal constraints on the control derivatives are considered as

$$
\begin{aligned}
d\dot{\boldsymbol{u}}_1 &= \left[\frac{\mathrm{d}\boldsymbol{S}_N}{\mathrm{d}t}\right]_1 d\boldsymbol{c} = 0 - \dot{\boldsymbol{u}}_1^p, \\
d\dot{\boldsymbol{u}}_{N_t-1} &= \left[\frac{\mathrm{d}\boldsymbol{S}_N}{\mathrm{d}t}\right]_{N_t-1} d\boldsymbol{c} = 0 - \dot{\boldsymbol{u}}_{N_t-1}^p.
\end{aligned}
$$

The control path constraints are incorporated as

$$\boldsymbol{u}_{min} - \boldsymbol{u}_k^p \le \left[\boldsymbol{S}_N\right]_k d\boldsymbol{c} \le \boldsymbol{u}_{max} - \boldsymbol{u}_k^p \,.$$

The derivative constraints are achieved by considering

$$\dot{\boldsymbol{u}}_{min} - \dot{\boldsymbol{u}}_k^p \le \left[\frac{\mathrm{d}\boldsymbol{S}_N}{\mathrm{d}t}\right]_k d\boldsymbol{c} \le \dot{\boldsymbol{u}}_{max} - \dot{\boldsymbol{u}}_k^p \,.$$

A second-order system in the time-domain can be written as

$$u_{cmd}(t) = \frac{1}{(\omega_n(\boldsymbol{p}))^2}\ddot{u}(t) + \frac{2\zeta(\boldsymbol{p})}{\omega_n(\boldsymbol{p})}\dot{u}(t) + u(t)$$

where ζ and ω_n are the damping ratio and the natural frequency, respectively. $\boldsymbol{p}$ is the model-dependent parameters, e.g., the aerodynamic hinge moment. One may therefore be able to reversely compute the ideal actuator command, $u_{cmd}(t)$ based on the desired control variables and their derivatives.

2. Collocation, forward Euler

$$
\begin{aligned}
d\boldsymbol{x}_{k+1}^{(j+1)} + \boldsymbol{x}_{k+1}^{(j)} &= h\left[(\boldsymbol{F_x})_k^{(j)} d\boldsymbol{x}_k^{(j+1)} + (\boldsymbol{F_u})_k^{(j)} d\boldsymbol{u}_k^{(j+1)} + \boldsymbol{f}_k^{(j)}\right] + \left[\boldsymbol{x}_k^{(j)} + d\boldsymbol{x}_k^{(j+1)}\right] \\
&= \left[h(\boldsymbol{F_x})_k^{(j)} + \boldsymbol{I}_n\right] d\boldsymbol{x}_k^{(j+1)} + h(\boldsymbol{F_u})_k^{(j)} d\boldsymbol{u}_k^{(j+1)} + h\boldsymbol{f}_k^{(j)} + \boldsymbol{x}_k^{(j)}
\end{aligned}
$$

Taking advantage of the linear properties of the parameterization, the following equation can be obtained:

$$
\begin{aligned}
h(\boldsymbol{F_u})_k^{(j)} d\boldsymbol{u}_k^{(j+1)} &= h\left(\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{u}}\right)_k^{(j)} \left(\frac{\partial \boldsymbol{u}}{\partial \boldsymbol{c}}\right)_k d\boldsymbol{c}^{(j+1)} \\
&= \left[h\left(\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{u}}\right)_k^{(j)} (\boldsymbol{S}_N)_k\right] d\boldsymbol{c}^{(j+1)} = \left(\tilde{\boldsymbol{F}}_{\boldsymbol{u}}\right)_k^{(j)} d\boldsymbol{c}^{(j+1)}
\end{aligned}
$$

Sequential Convex Optimization Methods

$$
\begin{aligned}
d\boldsymbol{x}_{k+1}^{(j+1)} + \boldsymbol{x}_{k+1}^{(j)} &= h\left[(\boldsymbol{F}_{\boldsymbol{x}})_k^{(j)} d\boldsymbol{x}_k^{(j+1)} + (\boldsymbol{F}_{\boldsymbol{u}})_k^{(j)} d\boldsymbol{u}_k^{(j+1)} + \boldsymbol{f}_k^{(j)}\right] + \left[\boldsymbol{x}_k^{(j)} + d\boldsymbol{x}_k^{(j+1)}\right] \\
&= \left[h(\boldsymbol{F}_{\boldsymbol{x}})_k^{(j)} + \boldsymbol{I}_n\right] d\boldsymbol{x}_k^{(j+1)} + h(\boldsymbol{F}_{\boldsymbol{u}})_k^{(j)} d\boldsymbol{u}_k^{(j+1)} + h\boldsymbol{f}_k^{(j)} + \boldsymbol{x}_k^{(j)}
\end{aligned}
$$

Taking advantage of the linear properties of the parameterization, the following equation can be obtained:

$$
\begin{aligned}
h(\boldsymbol{F}_{\boldsymbol{u}})_k^{(j)} d\boldsymbol{u}_k^{(j+1)} &= h\left(\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{u}}\right)_k^{(j)} \left(\frac{\partial \boldsymbol{u}}{\partial \boldsymbol{c}}\right)_k d\boldsymbol{c}^{(j+1)} \\
&= \left[h\left(\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{u}}\right)_k^{(j)} (\boldsymbol{S}_N)_k\right] d\boldsymbol{c}^{(j+1)} = \left(\tilde{\boldsymbol{F}}_{\boldsymbol{u}}\right)_k^{(j)} d\boldsymbol{c}^{(j+1)}
\end{aligned}
$$

Sequential Convex Optimization Methods

$$d\boldsymbol{x}_{k+1}^{(j+1)} - \left(\tilde{\boldsymbol{F}}_{\boldsymbol{x}}\right)_k^{(j)} d\boldsymbol{x}_k^{(j+1)} - \left(\tilde{\boldsymbol{F}}_{\boldsymbol{u}}\right)_k^{(j)} d\boldsymbol{u}_k^{(j+1)} = \boldsymbol{e}_{k+1}^{(j)}$$

$$\Updownarrow$$

$$d\boldsymbol{x}_{k+1}^{(j+1)} - \left(\tilde{\boldsymbol{F}}_{\boldsymbol{x}}\right)_k^{(j)} d\boldsymbol{x}_k^{(j+1)} - \left(\tilde{\boldsymbol{F}}_{\boldsymbol{u}}\right)_k^{(j)} d\boldsymbol{c}^{(j+1)} = \boldsymbol{e}_{k+1}^{(j)}$$

Sequential Convex Optimization Methods

$$d\boldsymbol{x}_{k+1}^{(j+1)} - \left(\tilde{\boldsymbol{F}}_{\boldsymbol{x}}\right)_k^{(j)} d\boldsymbol{x}_k^{(j+1)} - \left(\tilde{\boldsymbol{F}}_{\boldsymbol{u}}\right)_k^{(j)} d\boldsymbol{u}_k^{(j+1)} = \boldsymbol{e}_{k+1}^{(j)}$$

$$\boldsymbol{D}^{(j)} d\boldsymbol{x}_a^{(j+1)} = \boldsymbol{e}_a^{(j)}$$

where

$$\boldsymbol{D}^{(j)} = \begin{bmatrix} \boldsymbol{I}_n & 0 & \dots & 0 & 0 & -\left(\tilde{\boldsymbol{F}}_{\boldsymbol{u}}\right)_1^{(j)} & 0 & \dots & 0 \\ -\left(\tilde{\boldsymbol{F}}_{\boldsymbol{x}}\right)_2^{(j)} & \boldsymbol{I}_n & \dots & 0 & 0 & 0 & -\left(\tilde{\boldsymbol{F}}_{\boldsymbol{u}}\right)_2^{(j)} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & -\left(\tilde{\boldsymbol{F}}_{\boldsymbol{x}}\right)_{N_t-1}^{(j)} & \boldsymbol{I}_n & 0 & 0 & \dots & -\left(\tilde{\boldsymbol{F}}_{\boldsymbol{u}}\right)_{N_t-1}^{(j)} \end{bmatrix}$$

Sequential Convex Optimization Methods

$$\boldsymbol{D}^{(j)} d\boldsymbol{x}_a^{(j+1)} = \boldsymbol{e}_a^{(j)} \,,$$

where

$$\boldsymbol{D}^{(j)} = \begin{bmatrix} \boldsymbol{I}_n & 0 & \dots & 0 & 0 & -\left(\tilde{\boldsymbol{F}}_{\boldsymbol{u}}\right)_1^{(j)} \\ -\left(\tilde{\boldsymbol{F}}_{\boldsymbol{x}}\right)_2^{(j)} & \boldsymbol{I}_n & \dots & 0 & 0 & -\left(\tilde{\boldsymbol{F}}_{\boldsymbol{u}}\right)_2^{(j)} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & -\left(\tilde{\boldsymbol{F}}_{\boldsymbol{x}}\right)_{N_t-1}^{(j)} & \boldsymbol{I}_n & -\left(\tilde{\boldsymbol{F}}_{\boldsymbol{u}}\right)_{N_t-1}^{(j)} \end{bmatrix},$$

Sequential Convex Optimization Methods

$$
\begin{aligned}
d\boldsymbol{x}_{k+1} = {} & \frac{d\tau}{2}\left[\frac{\partial \boldsymbol{x}'}{\partial \boldsymbol{x}}\right]_k^p d\boldsymbol{x}_k + \underline{\frac{d\tau}{2}\left[\frac{\partial \boldsymbol{x}'}{\partial \boldsymbol{u}}\right]_k^p d\boldsymbol{u}_k} + \frac{d\tau}{2}\left[\frac{\partial \boldsymbol{x}'}{\partial t_f}\right]_k^p dt_f \\
& + \frac{d\tau}{2}\left[\frac{\partial \boldsymbol{x}'}{\partial \boldsymbol{x}}\right]_{k+1}^p d\boldsymbol{x}_{k+1} + \underline{\frac{d\tau}{2}\left[\frac{\partial \boldsymbol{x}'}{\partial \boldsymbol{u}}\right]_{k+1}^p d\boldsymbol{u}_{k+1}} + \frac{d\tau}{2}\left[\frac{\partial \boldsymbol{x}'}{\partial t_f}\right]_{k+1}^p dt_f \\
& + d\boldsymbol{x}_k + \boldsymbol{e}_{k+1}
\end{aligned}
$$

$$
\begin{aligned}
d\boldsymbol{x}_{k+1} = {} & \frac{d\tau}{2}\left[\frac{\partial \boldsymbol{x}'}{\partial \boldsymbol{x}}\right]_k^p d\boldsymbol{x}_k + \underline{\frac{d\tau}{2}\left[\frac{\partial \boldsymbol{x}'}{\partial \boldsymbol{u}}\right]_k^p \left[\left(\boldsymbol{S}_N\right)_k d\boldsymbol{c}\right]} + \frac{d\tau}{2}\left[\frac{\partial \boldsymbol{x}'}{\partial t_f}\right]_k^p dt_f \\
& + \frac{d\tau}{2}\left[\frac{\partial \boldsymbol{x}'}{\partial \boldsymbol{x}}\right]_{k+1}^p d\boldsymbol{x}_{k+1} + \underline{\frac{d\tau}{2}\left[\frac{\partial \boldsymbol{x}'}{\partial \boldsymbol{u}}\right]_{k+1}^p \left[\left(\boldsymbol{S}_N\right)_{k+1} d\boldsymbol{c}\right]} + \frac{d\tau}{2}\left[\frac{\partial \boldsymbol{x}'}{\partial t_f}\right]_{k+1}^p dt_f \\
& + d\boldsymbol{x}_k + \boldsymbol{e}_{k+1}
\end{aligned}
$$

3. Collocation, trapzoidal

$$
\begin{bmatrix}
-\left[\boldsymbol{F}_{\boldsymbol{x}}^{+}\right]_1 & \left[\boldsymbol{F}_{\boldsymbol{x}}^{-}\right]_2 & 0 & \cdots & 0 \\
0 & -\left[\boldsymbol{F}_{\boldsymbol{x}}^{+}\right]_2 & \left[\boldsymbol{F}_{\boldsymbol{x}}^{-}\right]_3 & \cdots & 0 \\
\vdots & \vdots & \ddots & \ddots & \vdots \\
0 & 0 & 0 & -\left[\boldsymbol{F}_{\boldsymbol{x}}^{+}\right]_{N-1} & \left[\boldsymbol{F}_{\boldsymbol{x}}^{-}\right]_N
\end{bmatrix}
\begin{bmatrix}
d\boldsymbol{x}_1 \\ d\boldsymbol{x}_2 \\ \vdots \\ d\boldsymbol{x}_N
\end{bmatrix}
$$

$$
-\begin{bmatrix}
\left[\boldsymbol{F}_{\boldsymbol{u}}\right]_1 & \left[\boldsymbol{F}_{\boldsymbol{u}}\right]_2 & 0 & \cdots & 0 \\
0 & \left[\boldsymbol{F}_{\boldsymbol{u}}\right]_2 & \left[\boldsymbol{F}_{\boldsymbol{u}}\right]_3 & \cdots & 0 \\
\vdots & \vdots & \ddots & \ddots & \vdots \\
0 & 0 & 0 & \left[\boldsymbol{F}_{\boldsymbol{u}}\right]_{N-1} & \left[\boldsymbol{F}_{\boldsymbol{u}}\right]_N
\end{bmatrix}
\begin{bmatrix}
d\boldsymbol{u}_1 \\ d\boldsymbol{u}_2 \\ \vdots \\ d\boldsymbol{u}_N
\end{bmatrix}
-\begin{bmatrix}
\left[\boldsymbol{F}_{t_f}\right]_1 \\ \left[\boldsymbol{F}_{t_f}\right]_2 \\ \vdots \\ \left[\boldsymbol{F}_{t_f}\right]_{N-1}
\end{bmatrix} dt_f =
\begin{bmatrix}
\boldsymbol{e}_2 \\ \boldsymbol{e}_3 \\ \vdots \\ \boldsymbol{e}_N
\end{bmatrix}
$$

Sequential Convex Optimization Methods

$$\boldsymbol{M}\boldsymbol{z}_o = \boldsymbol{e}$$

where

$$\boldsymbol{M} = \begin{bmatrix} -\left[\boldsymbol{F}_{\boldsymbol{x}}^{+}\right]_1 & \left[\boldsymbol{F}_{\boldsymbol{x}}^{-}\right]_2 & 0 & \cdots & 0 & -\left[\boldsymbol{F}_{\boldsymbol{c}}\right]_1 & -\left[\boldsymbol{F}_{t_f}\right]_1 \\ 0 & -\left[\boldsymbol{F}_{\boldsymbol{x}}^{+}\right]_2 & \left[\boldsymbol{F}_{\boldsymbol{x}}^{-}\right]_3 & \cdots & 0 & -\left[\boldsymbol{F}_{\boldsymbol{c}}\right]_2 & -\left[\boldsymbol{F}_{t_f}\right]_2 \\ \vdots & \vdots & \ddots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & -\left[\boldsymbol{F}_{\boldsymbol{x}}^{+}\right]_{N-1} & \left[\boldsymbol{F}_{\boldsymbol{x}}^{-}\right]_N & -\left[\boldsymbol{F}_{\boldsymbol{c}}\right]_{N-1} & -\left[\boldsymbol{F}_{t_f}\right]_{N-1} \end{bmatrix}$$

$$\boldsymbol{z}_o = \left[(d\boldsymbol{x}_1)^{\mathsf{T}}, (d\boldsymbol{x}_2)^{\mathsf{T}}, \ldots, (d\boldsymbol{x}_N)^{\mathsf{T}}, (d\boldsymbol{c})^{\mathsf{T}}, dt_f\right]^{\mathsf{T}}$$

$$\boldsymbol{e} = \left[(\boldsymbol{e}_2)^{\mathsf{T}}, (\boldsymbol{e}_3)^{\mathsf{T}}, \ldots, (\boldsymbol{e}_N)^{\mathsf{T}}\right]^{\mathsf{T}}$$

$$\left[\boldsymbol{F}_{\boldsymbol{c}}\right]_k = \frac{d\tau}{2}\left(\left[\frac{\partial \boldsymbol{x}'}{\partial \boldsymbol{u}}\right]_k^p (\boldsymbol{S}_M)_k + \left[\frac{\partial \boldsymbol{x}'}{\partial \boldsymbol{u}}\right]_{k+1}^p (\boldsymbol{S}_M)_{k+1}\right)$$

Control the Time Behavior

We introduce a new variable that monotonically increases with respect to time:

$$\frac{\mathrm{d}p}{\mathrm{d}t} = v(t) > 0$$

$$p(0) = 0$$

$$p(T_f) = P$$

Next, we reformulate the control parameterizationto be a function of p:

$$u\left(p\right) = \boldsymbol{s}_N\left(p\right)\boldsymbol{c}_N$$

The time behavior of the control u is no longer only affected by $\boldsymbol{c}_N$ but preserves characteristics of $p\left(t\right)$ which are defined already in the designing process of $v\left(t\right)$.

This can be inferred by using the chain rule as

$$\frac{\mathrm{d}u}{\mathrm{d}t} = \frac{\mathrm{d}u}{\mathrm{d}p}\frac{\mathrm{d}p}{\mathrm{d}t}$$

Provided that T_f is fixed, the value of $\boldsymbol{s}_N$ at each discrete point is known a priori. That is to say, $\boldsymbol{s}_N$ does not change during the optimization algorithm execution. Therefore, it can be evaluated only once before executing the optimization program. For some optimization approaches including the one utilized in this paper, this feature saves some computational effort and streamlines the formulation.

We write the parameterization in a discrete time form for the control vector

$$\boldsymbol{u}_k = [\boldsymbol{S}_N]_k \, \bar{\boldsymbol{c}}_N, \;\; k = 1, 2, \ldots, N_t - 1$$

where

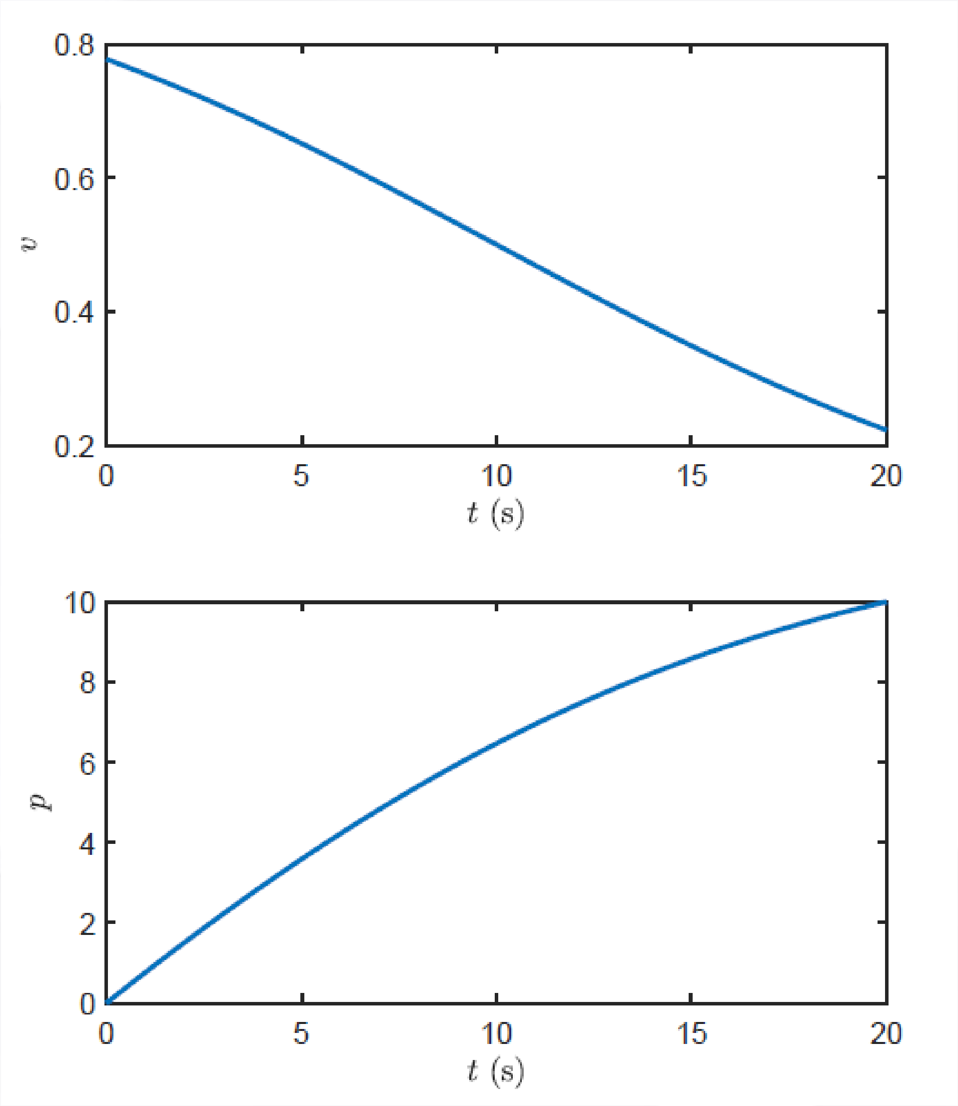
$$[\boldsymbol{S}_N]_k = \begin{bmatrix} \left[\boldsymbol{s}_N^{(1)}\right]_k & 0 & \ldots & 0 \\ 0 & \left[\boldsymbol{s}_N^{(2)}\right]_k & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & \left[\boldsymbol{s}_N^{(m)}\right]_k \end{bmatrix}$$

Sequential Convex Optimization Methods

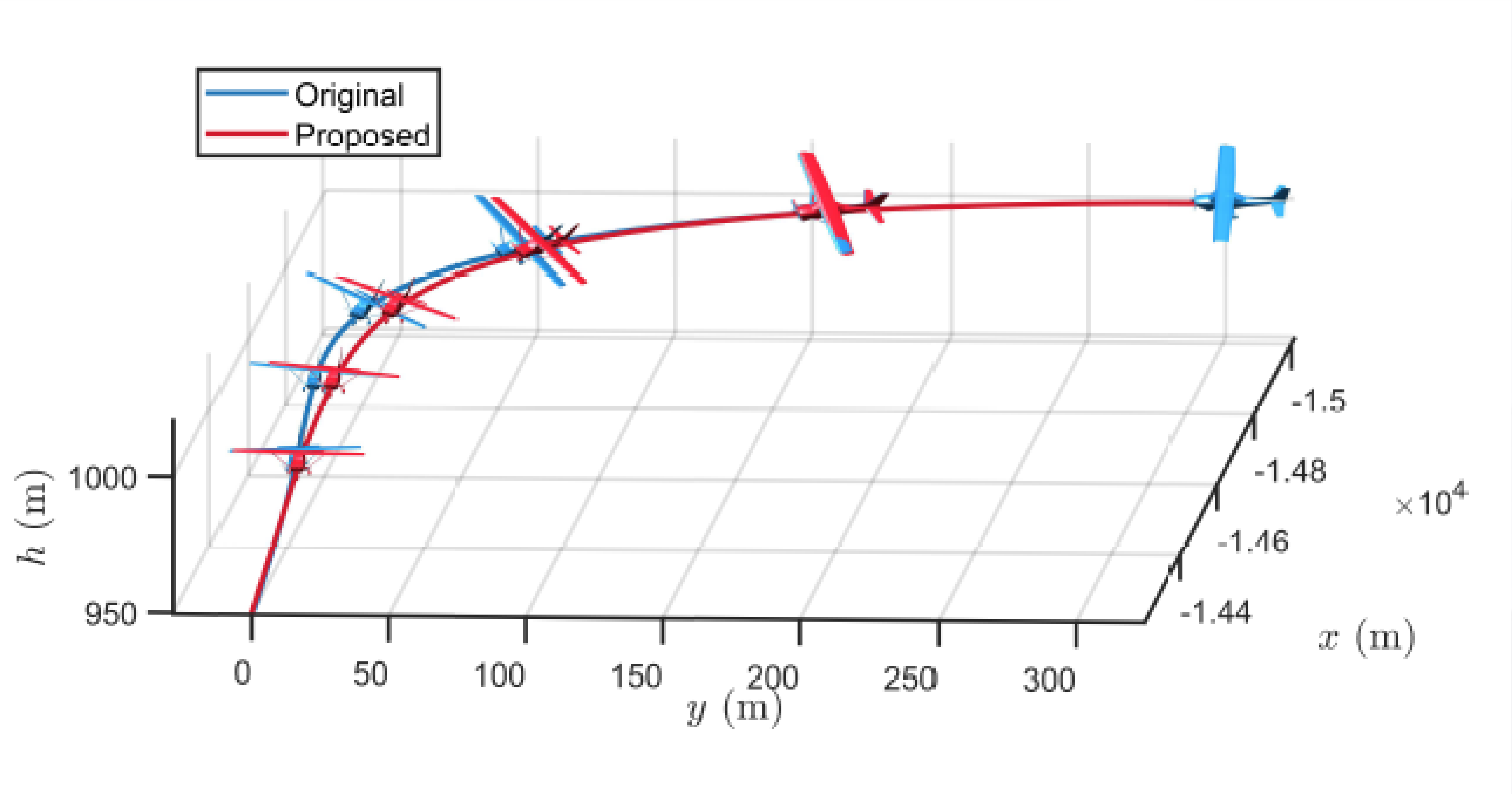
$$
\begin{aligned}
d\boldsymbol{x}_{k+1} &= \sum_{j=1}^{k} \boldsymbol{B}_{k,j} \left[\boldsymbol{S}_N\right]_j d\bar{\boldsymbol{c}}_N \\
&= \sum_{j=1}^{k} \boldsymbol{D}_{k,j} d\bar{\boldsymbol{c}}_N
\end{aligned}
$$

Note: $\boldsymbol{S}_N$ is different due to the time scaling

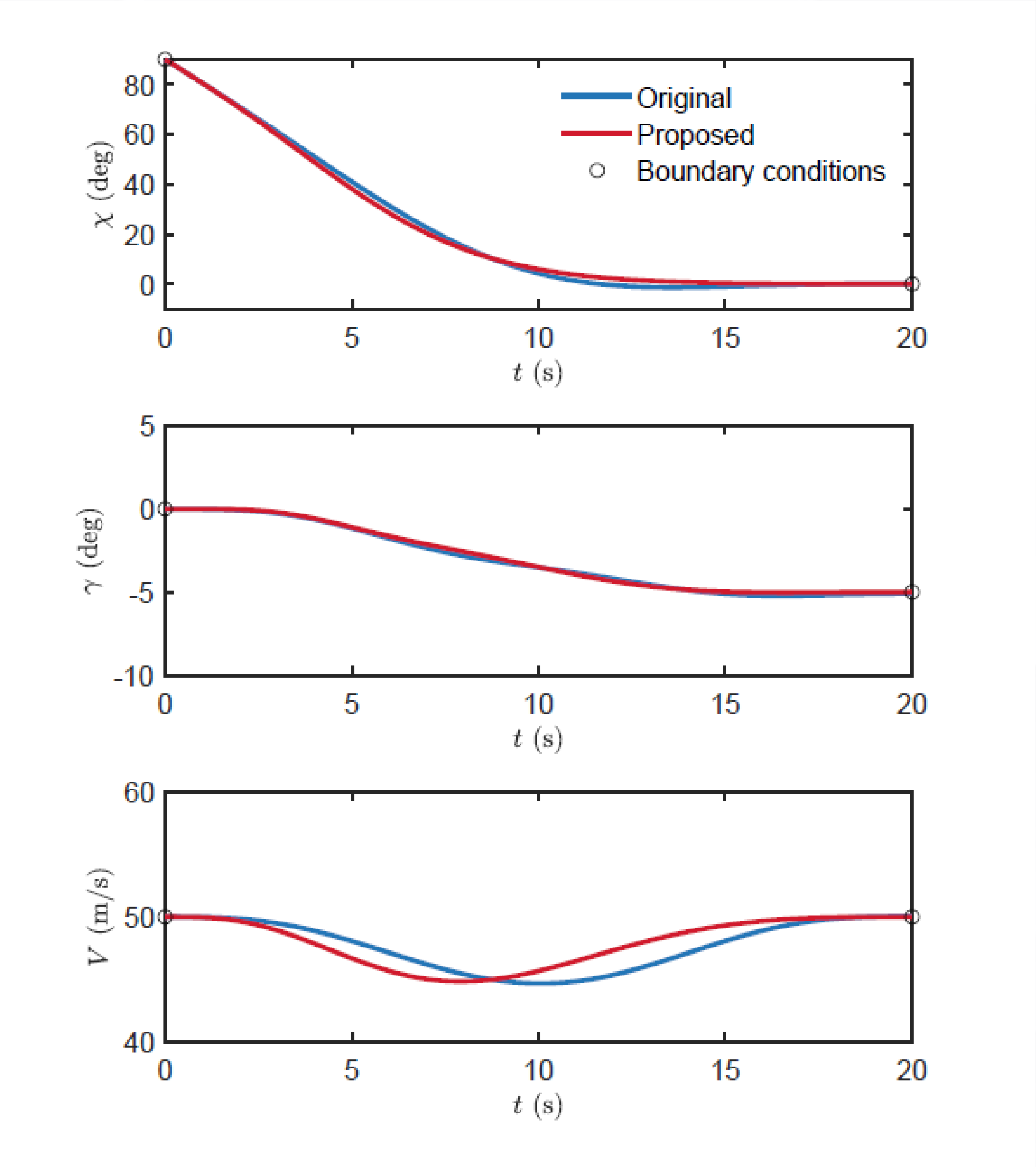
Sequential Convex Optimization Methods

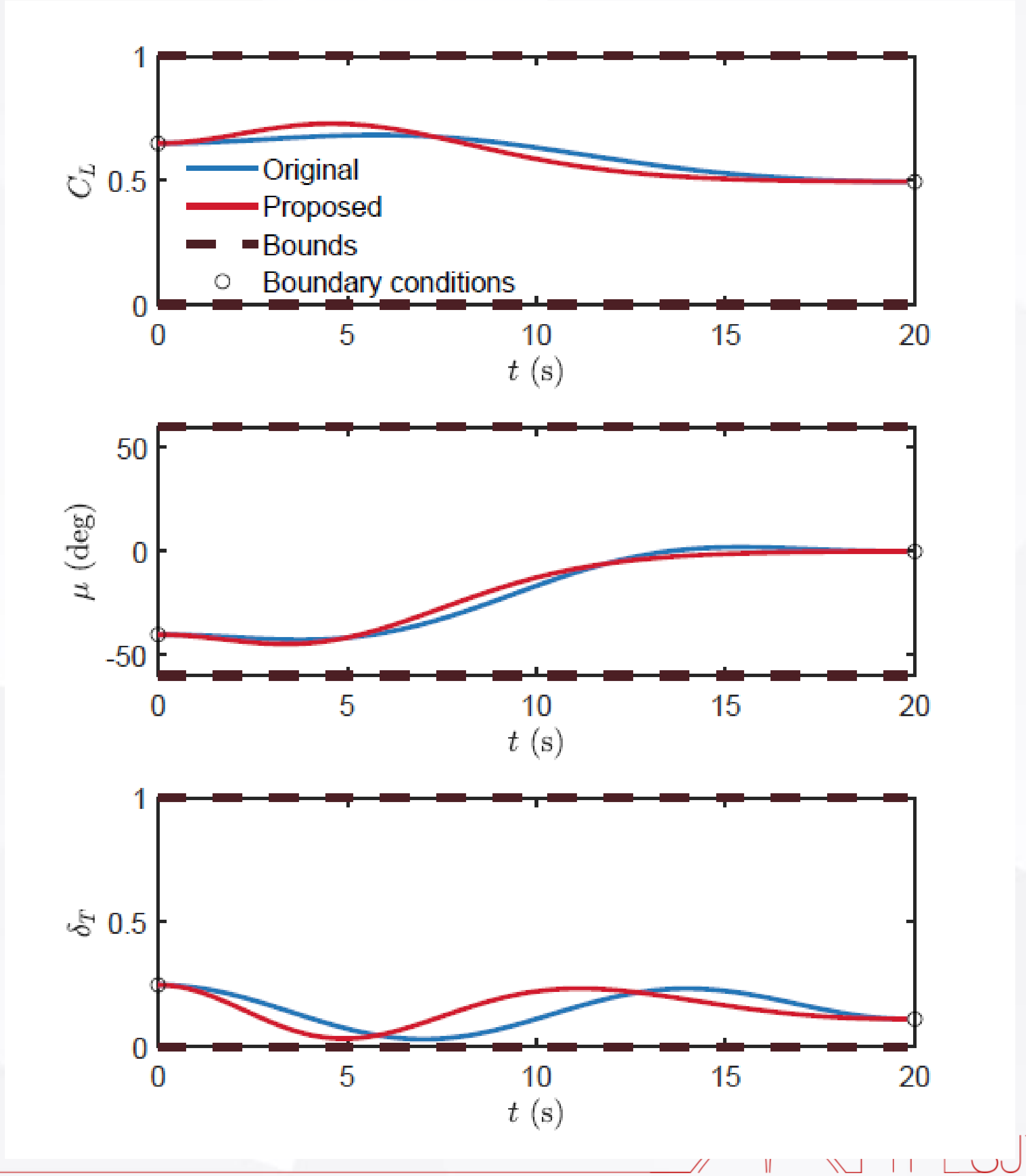


Sequential Convex Optimization Methods



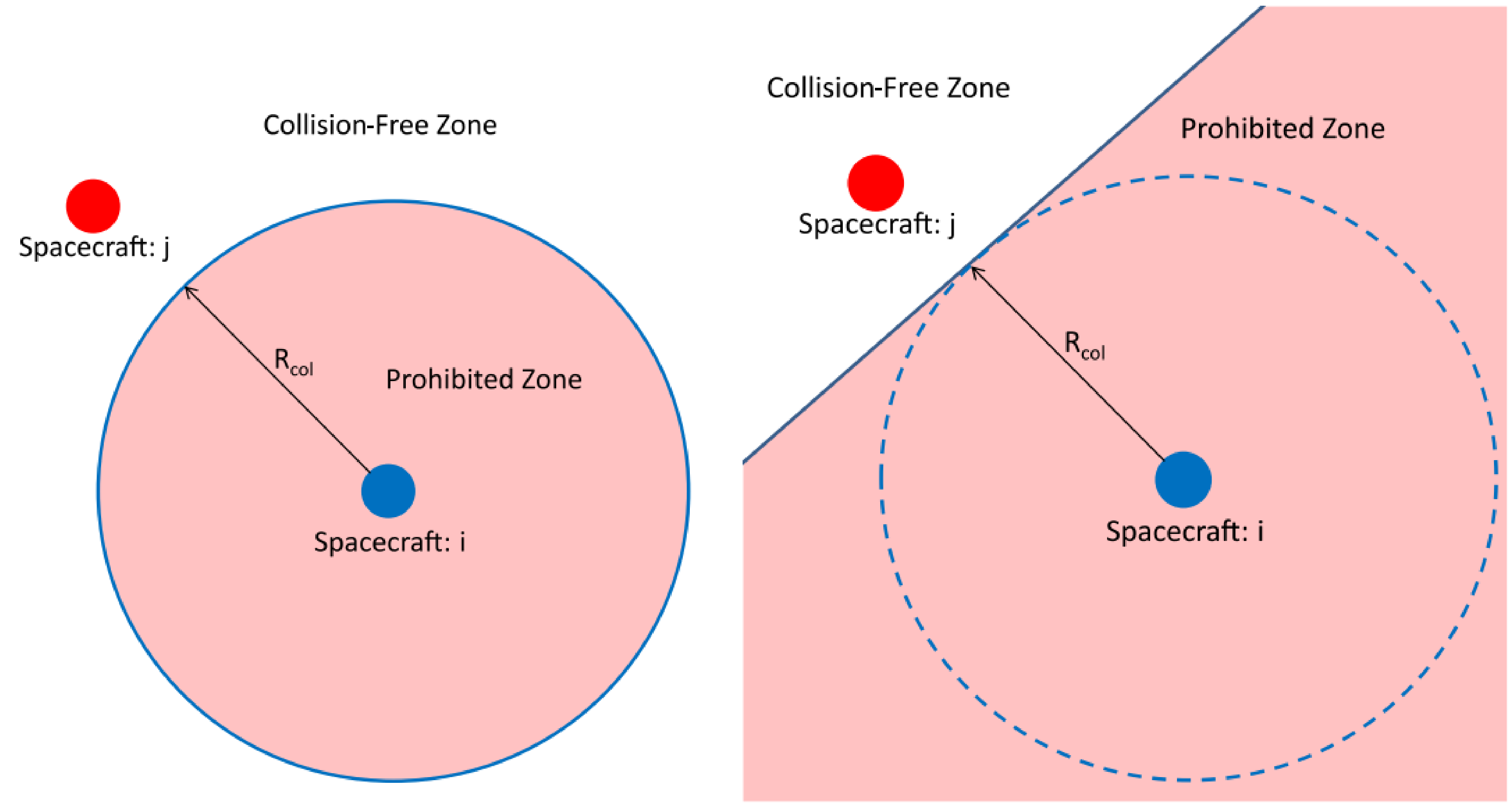
Sequential Convex Optimization Methods





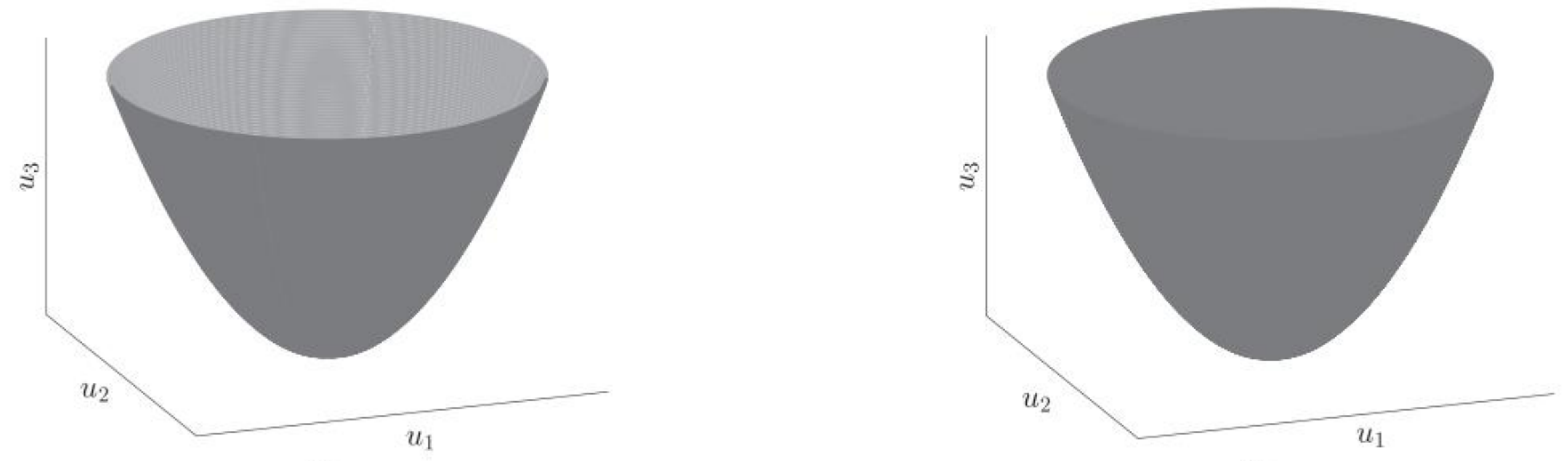
Convex Relaxation of Constraints

Inexact



a) Nonconvex prohibited zone **b) Convex approximation of prohibited zone**

Exact

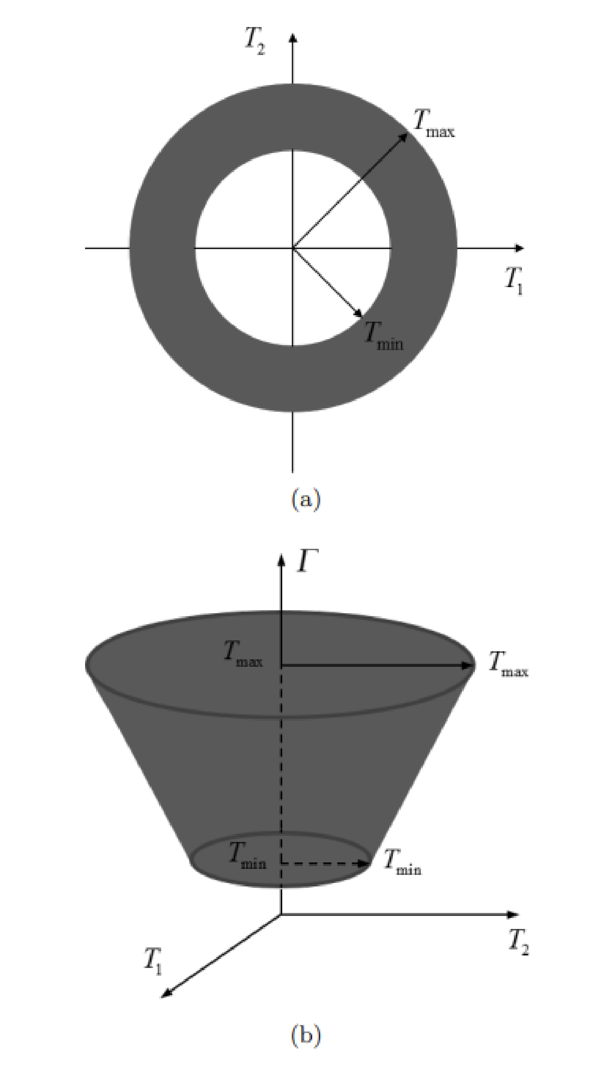


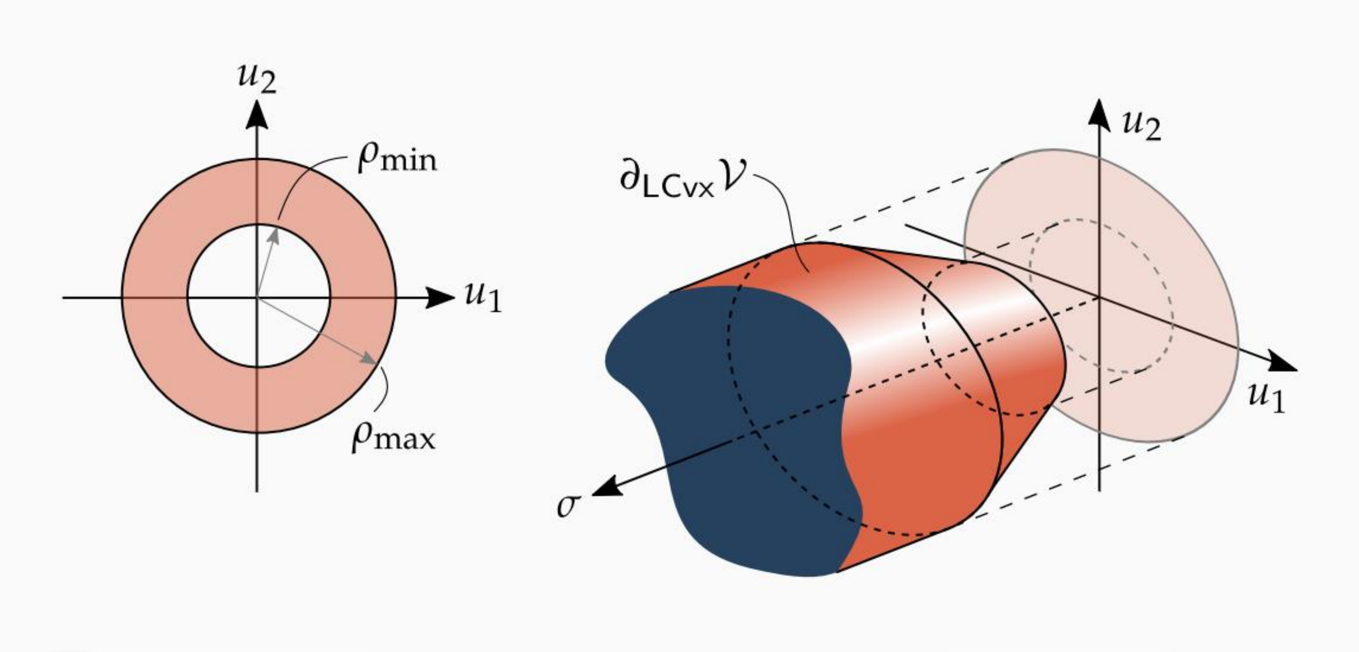
(a) (b)

$$\boldsymbol{u} \in \mathcal{U} := \left\{u_1^2 + u_2^2 = u_3,\, 0 \le u_3 \le \bar{u}_3\right\}$$

$$\boxed{\boldsymbol{u} \in \bar{\mathcal{U}} = \mathbf{conv}\,\mathcal{U} := \left\{u_1^2 + u_2^2 \le u_3,\, 0 \le u_3 \le \bar{u}_3\right\}}$$

Exact





$$T_{\min} \leqslant \|\boldsymbol{T}\| \leqslant T_{\max}$$

$$\|\boldsymbol{T}\| \leqslant \Gamma$$

$$T_{\min} \leqslant \Gamma \leqslant T_{\max}$$