Numerical Optimal Control with DAEs
Lecture 12: Optimal Control with DAEs

Sébastien Gros

AWESCO PhD course

Objectives of the lecture

- Basic Optimal Control Problems with DAEs
- Transcription of DAE-based OCPs into NLPs
- A first view at LICQ issues in Optimal Control with DAEs
- (Constraints drift in Optimal Control with DAEs)

Outline

Preliminary remarks

Preliminary remarks

Semi-explicit DAE-constrained OCP

$$\begin{aligned}
\min_{\mathbf{x}(.),\mathbf{z}(.),\mathbf{u}(.)} \quad & \phi\left(\mathbf{x}\left(.\right),\mathbf{z}\left(.\right),\mathbf{u}\left(.\right)\right) \\
\text{s.t.} \quad & \dot{\mathbf{x}}\left(t\right) = \mathbf{F}\left(\mathbf{z}\left(t\right),\mathbf{x}\left(t\right),\mathbf{u}\left(t\right)\right) \\
& 0 = \mathbf{G}\left(\mathbf{z}\left(t\right),\mathbf{x}\left(t\right),\mathbf{u}\left(t\right)\right) \\
& \mathbf{h}\left(\mathbf{z}\left(t\right),\mathbf{x}\left(t\right),\mathbf{u}\left(t\right)\right) \leq 0 \\
& \mathbf{x}(0) - \bar{\mathbf{x}}_0 = 0
\end{aligned}$$

Preliminary remarks

Semi-explicit DAE-constrained OCP

$$\begin{aligned}
\min_{\mathbf{x}(.),\mathbf{z}(.),\mathbf{u}(.)} \quad & \phi\left(\mathbf{x}\left(.\right),\mathbf{z}\left(.\right),\mathbf{u}\left(.\right)\right) \\
\text{s.t.} \quad & \dot{\mathbf{x}}\left(t\right) = \mathbf{F}\left(\mathbf{z}\left(t\right),\mathbf{x}\left(t\right),\mathbf{u}\left(t\right)\right) \\
& 0 = \mathbf{G}\left(\mathbf{z}\left(t\right),\mathbf{x}\left(t\right),\mathbf{u}\left(t\right)\right) \\
& \mathbf{h}\left(\mathbf{z}\left(t\right),\mathbf{x}\left(t\right),\mathbf{u}\left(t\right)\right) \leq 0 \\
& \mathbf{x}(0) - \bar{\mathbf{x}}_0 = 0
\end{aligned}$$

Fully implicit DAE-constrained OCP

$$\begin{aligned}
\min_{\mathbf{x}(.),\mathbf{z}(.),\mathbf{u}(.)} \quad & \phi\left(\mathbf{x}\left(.\right),\mathbf{z}\left(.\right),\mathbf{u}\left(.\right)\right) \\
\text{s.t.} \quad & \mathbf{F}\left(\dot{\mathbf{x}}\left(t\right),\mathbf{z}\left(t\right),\mathbf{x}\left(t\right),\mathbf{u}\left(t\right)\right) = 0 \\
& \mathbf{h}\left(\mathbf{z}\left(t\right),\mathbf{x}\left(t\right),\mathbf{u}\left(t\right)\right) \leq 0 \\
& \mathbf{x}(0) - \bar{\mathbf{x}}_0 = 0
\end{aligned}$$

Preliminary remarks

Semi-explicit DAE-constrained OCP

$$\begin{aligned} \min_{\mathbf{x}(.),\mathbf{z}(.),\mathbf{u}(.)} \quad & \phi\left(\mathbf{x}\left(.\right),\mathbf{z}\left(.\right),\mathbf{u}\left(.\right)\right) \\ \text{s.t.} \quad & \dot{\mathbf{x}}\left(t\right) = \mathbf{F}\left(\mathbf{z}\left(t\right),\mathbf{x}\left(t\right),\mathbf{u}\left(t\right)\right) \\ & 0 = \mathbf{G}\left(\mathbf{z}\left(t\right),\mathbf{x}\left(t\right),\mathbf{u}\left(t\right)\right) \\ & \mathbf{h}\left(\mathbf{z}\left(t\right),\mathbf{x}\left(t\right),\mathbf{u}\left(t\right)\right) \leq 0 \\ & \mathbf{x}(0) - \bar{\mathbf{x}}_0 = 0 \end{aligned}$$

Fully implicit DAE-constrained OCP

$$\begin{aligned} \min_{\mathbf{x}(.),\mathbf{z}(.),\mathbf{u}(.)} \quad & \phi\left(\mathbf{x}\left(.\right),\mathbf{z}\left(.\right),\mathbf{u}\left(.\right)\right) \\ \text{s.t.} \quad & \mathbf{F}\left(\dot{\mathbf{x}}\left(t\right),\mathbf{z}\left(t\right),\mathbf{x}\left(t\right),\mathbf{u}\left(t\right)\right) = 0 \\ & \mathbf{h}\left(\mathbf{z}\left(t\right),\mathbf{x}\left(t\right),\mathbf{u}\left(t\right)\right) \leq 0 \\ & \mathbf{x}(0) - \bar{\mathbf{x}}_0 = 0 \end{aligned}$$

- OCPs based on **index-1 DAEs** are the most common, we will focus on this case

Preliminary remarks

Semi-explicit DAE-constrained OCP

$$\begin{aligned} \min_{\mathbf{x}(.),\mathbf{z}(.),\mathbf{u}(.)} \quad & \phi\left(\mathbf{x}\left(.\right),\mathbf{z}\left(.\right),\mathbf{u}\left(.\right)\right) \\ \text{s.t.} \quad & \dot{\mathbf{x}}\left(t\right) = \mathbf{F}\left(\mathbf{z}\left(t\right),\mathbf{x}\left(t\right),\mathbf{u}\left(t\right)\right) \\ & 0 = \mathbf{G}\left(\mathbf{z}\left(t\right),\mathbf{x}\left(t\right),\mathbf{u}\left(t\right)\right) \\ & \mathbf{h}\left(\mathbf{z}\left(t\right),\mathbf{x}\left(t\right),\mathbf{u}\left(t\right)\right) \leq 0 \\ & \mathbf{x}(0) - \bar{\mathbf{x}}_0 = 0 \end{aligned}$$

Fully implicit DAE-constrained OCP

$$\begin{aligned} \min_{\mathbf{x}(.),\mathbf{z}(.),\mathbf{u}(.)} \quad & \phi\left(\mathbf{x}\left(.\right),\mathbf{z}\left(.\right),\mathbf{u}\left(.\right)\right) \\ \text{s.t.} \quad & \mathbf{F}\left(\dot{\mathbf{x}}\left(t\right),\mathbf{z}\left(t\right),\mathbf{x}\left(t\right),\mathbf{u}\left(t\right)\right) = 0 \\ & \mathbf{h}\left(\mathbf{z}\left(t\right),\mathbf{x}\left(t\right),\mathbf{u}\left(t\right)\right) \leq 0 \\ & \mathbf{x}(0) - \bar{\mathbf{x}}_0 = 0 \end{aligned}$$

- OCPs based on **index-1 DAEs** are the most common, we will focus on this case
- For now we will focus on OCPs with **assigned initial conditions**, i.e. $\mathbf{x}\left(0\right)$ has to take a specific value $\bar{\mathbf{x}}_0$

Preliminary remarks

Semi-explicit DAE-constrained OCP

$$
\begin{aligned}
\min_{\mathbf{x}(.),\mathbf{z}(.),\mathbf{u}(.)} \quad & \phi\left(\mathbf{x}\left(.\right),\mathbf{z}\left(.\right),\mathbf{u}\left(.\right)\right) \\
\text{s.t.} \quad & \dot{\mathbf{x}}\left(t\right) = \mathbf{F}\left(\mathbf{z}\left(t\right),\mathbf{x}\left(t\right),\mathbf{u}\left(t\right)\right) \\
& 0 = \mathbf{G}\left(\mathbf{z}\left(t\right),\mathbf{x}\left(t\right),\mathbf{u}\left(t\right)\right) \\
& \mathbf{h}\left(\mathbf{z}\left(t\right),\mathbf{x}\left(t\right),\mathbf{u}\left(t\right)\right) \leq 0 \\
& \mathbf{x}(0) - \bar{\mathbf{x}}_0 = 0
\end{aligned}
$$

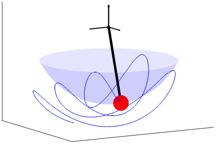
Fully implicit DAE-constrained OCP

$$
\begin{aligned}
\min_{\mathbf{x}(.),\mathbf{z}(.),\mathbf{u}(.)} \quad & \phi\left(\mathbf{x}\left(.\right),\mathbf{z}\left(.\right),\mathbf{u}\left(.\right)\right) \\
\text{s.t.} \quad & \mathbf{F}\left(\dot{\mathbf{x}}\left(t\right),\mathbf{z}\left(t\right),\mathbf{x}\left(t\right),\mathbf{u}\left(t\right)\right) = 0 \\
& \mathbf{h}\left(\mathbf{z}\left(t\right),\mathbf{x}\left(t\right),\mathbf{u}\left(t\right)\right) \leq 0 \\
& \mathbf{x}(0) - \bar{\mathbf{x}}_0 = 0
\end{aligned}
$$

- OCPs based on **index-1 DAEs** are the most common, we will focus on this case
- For now we will focus on OCPs with **assigned initial conditions**, i.e. $\mathbf{x}\left(0\right)$ has to take a specific value $\bar{\mathbf{x}}_0$
- The selected initial condition $\bar{\mathbf{x}}_0$ has to be **consistent**, i.e.

  $$\mathbf{C}\left(\bar{\mathbf{x}}_0\right) = 0$$

  where function $\mathbf{C}$ gathers the DAE consistency condition. Then the DAE is consistent throughout the trajectories...

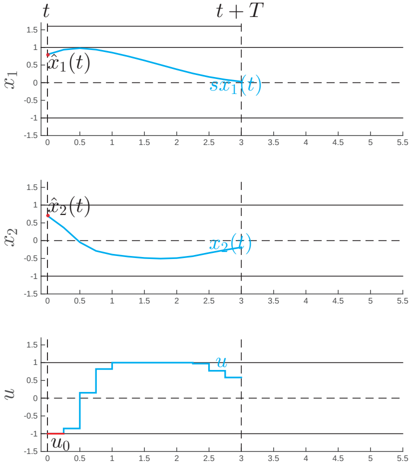


Online Optimal Control with Index-reduced DAEs

NMPC: OCP repeatedly solved online

$$
\begin{aligned}
\min_{\mathbf{x}(.),\mathbf{z}(.),\mathbf{u}(.)} \quad & T\left(\mathbf{x}\left(t_{\mathrm{f}}\right)\right) + \int_0^{t_{\mathrm{f}}} L\left(\mathbf{x},\mathbf{u}\right)\mathrm{d}\tau \\
\text{s.t.} \quad & \mathbf{F}\left(\dot{\mathbf{x}}\left(t\right),\mathbf{z}\left(t\right),\mathbf{x}\left(t\right),\mathbf{u}\left(t\right)\right) = 0 \\
& \mathbf{h}\left(\mathbf{z}\left(t\right),\mathbf{x}\left(t\right),\mathbf{u}\left(t\right)\right) \leq 0 \\
& \mathbf{x}(0) - \hat{\mathbf{x}}(t) = 0
\end{aligned}
$$

from the current state estimation $\hat{\mathbf{x}}(t)$.

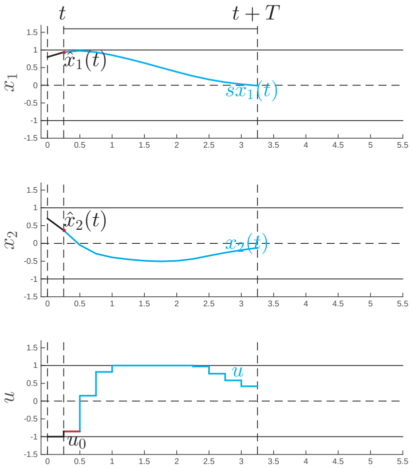


Online Optimal Control with Index-reduced DAEs

NMPC: OCP repeatedly solved online

$$
\begin{aligned}
\min_{\mathbf{x}(.),\mathbf{z}(.),\mathbf{u}(.)} \quad & T\left(\mathbf{x}\left(t_{\mathrm{f}}\right)\right) + \int_0^{t_{\mathrm{f}}} L\left(\mathbf{x},\mathbf{u}\right)\mathrm{d}\tau \\
\text{s.t.} \quad & \mathbf{F}\left(\dot{\mathbf{x}}\left(t\right),\mathbf{z}\left(t\right),\mathbf{x}\left(t\right),\mathbf{u}\left(t\right)\right) = 0 \\
& \mathbf{h}\left(\mathbf{z}\left(t\right),\mathbf{x}\left(t\right),\mathbf{u}\left(t\right)\right) \leq 0 \\
& \mathbf{x}(0) - \hat{\mathbf{x}}(t) = 0
\end{aligned}
$$

from the current state estimation $\hat{\mathbf{x}}(t)$.

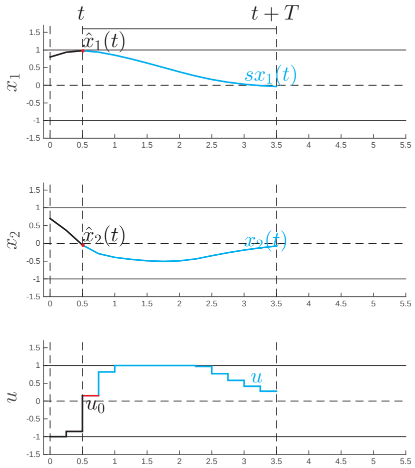


Online Optimal Control with Index-reduced DAEs

NMPC: OCP repeatedly solved online

$$\begin{aligned}
\min_{\mathbf{x}(.),\mathbf{z}(.),\mathbf{u}(.)} \quad & T\left(\mathbf{x}\left(t_\mathrm{f}\right)\right) + \int_0^{t_\mathrm{f}} L\left(\mathbf{x},\mathbf{u}\right)\mathrm{d}\tau \\
\text{s.t.} \quad & \mathbf{F}\left(\dot{\mathbf{x}}\left(t\right),\mathbf{z}\left(t\right),\mathbf{x}\left(t\right),\mathbf{u}\left(t\right)\right) = 0 \\
& \mathbf{h}\left(\mathbf{z}\left(t\right),\mathbf{x}\left(t\right),\mathbf{u}\left(t\right)\right) \leq 0 \\
& \mathbf{x}(0) - \hat{\mathbf{x}}(t) = 0
\end{aligned}$$

from the current state estimation $\hat{\mathbf{x}}(t)$.

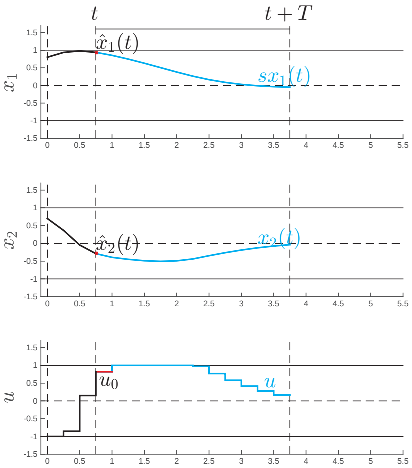


Online Optimal Control with Index-reduced DAEs

NMPC: OCP repeatedly solved online

$$
\begin{aligned}
\min_{\mathbf{x}(.),\mathbf{z}(.),\mathbf{u}(.)} \quad & T\left(\mathbf{x}\left(t_{\mathrm{f}}\right)\right) + \int_0^{t_{\mathrm{f}}} L\left(\mathbf{x},\mathbf{u}\right)\mathrm{d}\tau \\
\text{s.t.} \quad & \mathbf{F}\left(\dot{\mathbf{x}}\left(t\right),\mathbf{z}\left(t\right),\mathbf{x}\left(t\right),\mathbf{u}\left(t\right)\right) = 0 \\
& \mathbf{h}\left(\mathbf{z}\left(t\right),\mathbf{x}\left(t\right),\mathbf{u}\left(t\right)\right) \leq 0 \\
& \mathbf{x}(0) - \hat{\mathbf{x}}(t) = 0
\end{aligned}
$$

from the current state estimation $\hat{\mathbf{x}}(t)$.

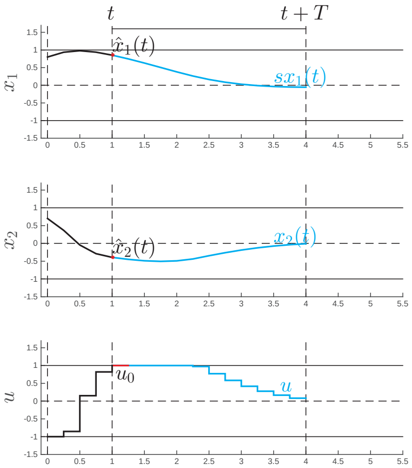


Online Optimal Control with Index-reduced DAEs

NMPC: OCP repeatedly solved online

$$
\begin{aligned}
\min_{\mathbf{x}(.),\mathbf{z}(.),\mathbf{u}(.)} \quad & T\left(\mathbf{x}\left(t_\mathrm{f}\right)\right) + \int_0^{t_\mathrm{f}} L\left(\mathbf{x},\mathbf{u}\right)\mathrm{d}\tau \\
\text{s.t.} \quad & \mathbf{F}\left(\dot{\mathbf{x}}\left(t\right), \mathbf{z}\left(t\right), \mathbf{x}\left(t\right), \mathbf{u}\left(t\right)\right) = 0 \\
& \mathbf{h}\left(\mathbf{z}\left(t\right), \mathbf{x}\left(t\right), \mathbf{u}\left(t\right)\right) \leq 0 \\
& \mathbf{x}(0) - \hat{\mathbf{x}}(t) = 0
\end{aligned}
$$

from the current state estimation $\hat{\mathbf{x}}(t)$.

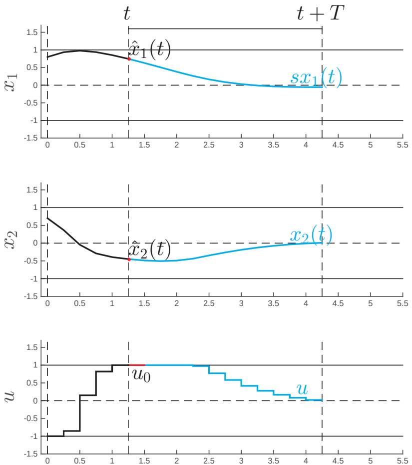


Online Optimal Control with Index-reduced DAEs

NMPC: OCP repeatedly solved online

$$
\begin{aligned}
\min_{\mathbf{x}(.),\mathbf{z}(.),\mathbf{u}(.)} \quad & T\left(\mathbf{x}\left(t_{\mathrm{f}}\right)\right) + \int_0^{t_{\mathrm{f}}} L\left(\mathbf{x},\mathbf{u}\right)\mathrm{d}\tau \\
\text{s.t.} \quad & \mathbf{F}\left(\dot{\mathbf{x}}\left(t\right),\mathbf{z}\left(t\right),\mathbf{x}\left(t\right),\mathbf{u}\left(t\right)\right) = 0 \\
& \mathbf{h}\left(\mathbf{z}\left(t\right),\mathbf{x}\left(t\right),\mathbf{u}\left(t\right)\right) \leq 0 \\
& \mathbf{x}(0) - \hat{\mathbf{x}}(t) = 0
\end{aligned}
$$

from the current state estimation $\hat{\mathbf{x}}(t)$.

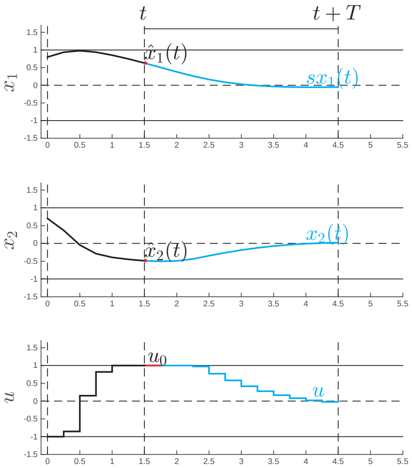


Online Optimal Control with Index-reduced DAEs

NMPC: OCP repeatedly solved online

$$
\begin{aligned}
\min_{\mathbf{x}(.),\mathbf{z}(.),\mathbf{u}(.)} \quad & T\left(\mathbf{x}\left(t_{\mathrm{f}}\right)\right) + \int_0^{t_{\mathrm{f}}} L\left(\mathbf{x},\mathbf{u}\right)\mathrm{d}\tau \\
\text{s.t.} \quad & \mathbf{F}\left(\dot{\mathbf{x}}\left(t\right),\mathbf{z}\left(t\right),\mathbf{x}\left(t\right),\mathbf{u}\left(t\right)\right) = 0 \\
& \mathbf{h}\left(\mathbf{z}\left(t\right),\mathbf{x}\left(t\right),\mathbf{u}\left(t\right)\right) \leq 0 \\
& \mathbf{x}(0) - \hat{\mathbf{x}}(t) = 0
\end{aligned}
$$

from the current state estimation $\hat{\mathbf{x}}(t)$.

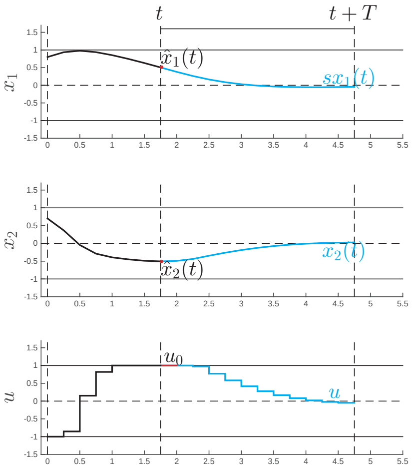


Online Optimal Control with Index-reduced DAEs

NMPC: OCP repeatedly solved online

$$\begin{aligned}
\min_{\mathbf{x}(.),\mathbf{z}(.),\mathbf{u}(.)} \quad & T\left(\mathbf{x}\left(t_{\mathrm{f}}\right)\right) + \int_0^{t_{\mathrm{f}}} L\left(\mathbf{x},\mathbf{u}\right)\mathrm{d}\tau \\
\text{s.t.} \quad & \mathbf{F}\left(\dot{\mathbf{x}}\left(t\right),\mathbf{z}\left(t\right),\mathbf{x}\left(t\right),\mathbf{u}\left(t\right)\right) = 0 \\
& \mathbf{h}\left(\mathbf{z}\left(t\right),\mathbf{x}\left(t\right),\mathbf{u}\left(t\right)\right) \leq 0 \\
& \mathbf{x}(0) - \hat{\mathbf{x}}(t) = 0
\end{aligned}$$

from the current state estimation $\hat{\mathbf{x}}(t)$.

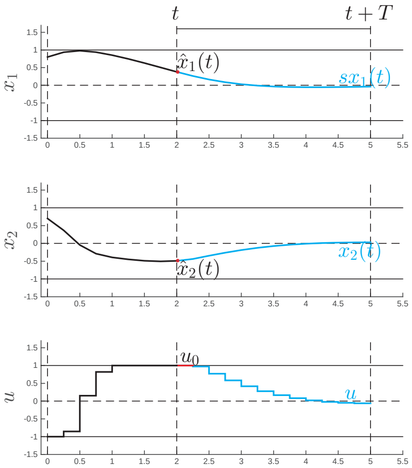


Online Optimal Control with Index-reduced DAEs

NMPC: OCP repeatedly solved online

$$
\begin{aligned}
\min_{\mathbf{x}(.),\mathbf{z}(.),\mathbf{u}(.)} \quad & T\left(\mathbf{x}\left(t_{\mathrm{f}}\right)\right) + \int_0^{t_{\mathrm{f}}} L\left(\mathbf{x},\mathbf{u}\right)\mathrm{d}\tau \\
\text{s.t.} \quad & \mathbf{F}\left(\dot{\mathbf{x}}\left(t\right),\mathbf{z}\left(t\right),\mathbf{x}\left(t\right),\mathbf{u}\left(t\right)\right) = 0 \\
& \mathbf{h}\left(\mathbf{z}\left(t\right),\mathbf{x}\left(t\right),\mathbf{u}\left(t\right)\right) \leq 0 \\
& \mathbf{x}(0) - \hat{\mathbf{x}}(t) = 0
\end{aligned}
$$

from the current state estimation $\hat{\mathbf{x}}(t)$.

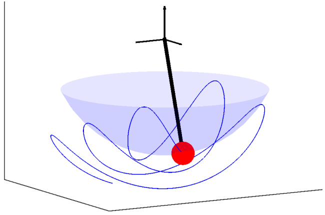


Online Optimal Control with Index-reduced DAEs

NMPC: OCP repeatedly solved online

$$\begin{aligned}
\min_{\mathbf{x}(.),\mathbf{z}(.),\mathbf{u}(.)} \quad & T\left(\mathbf{x}\left(t_\mathrm{f}\right)\right) + \int_0^{t_\mathrm{f}} L\left(\mathbf{x},\mathbf{u}\right)\mathrm{d}\tau \\
\text{s.t.} \quad & \mathbf{F}\left(\dot{\mathbf{x}}\left(t\right),\mathbf{z}\left(t\right),\mathbf{x}\left(t\right),\mathbf{u}\left(t\right)\right) = 0 \\
& \mathbf{h}\left(\mathbf{z}\left(t\right),\mathbf{x}\left(t\right),\mathbf{u}\left(t\right)\right) \leq 0 \\
& \mathbf{x}(0) - \hat{\mathbf{x}}(t) = 0
\end{aligned}$$

from the current state estimation $\hat{\mathbf{x}}(t)$.



How to impose the DAE consistency condition ?

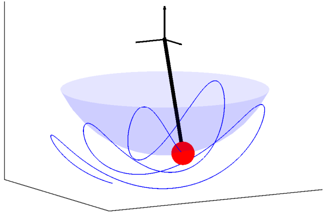
Online Optimal Control with Index-reduced DAEs

NMPC: OCP repeatedly solved online

$$\begin{aligned}\min_{\mathbf{x}(.),\mathbf{z}(.),\mathbf{u}(.)} \quad & T\left(\mathbf{x}\left(t_\mathrm{f}\right)\right) + \int_0^{t_\mathrm{f}} L\left(\mathbf{x},\mathbf{u}\right)\mathrm{d}\tau \\ \text{s.t.} \quad & \mathbf{F}\left(\dot{\mathbf{x}}\left(t\right),\mathbf{z}\left(t\right),\mathbf{x}\left(t\right),\mathbf{u}\left(t\right)\right) = 0 \\ & \mathbf{h}\left(\mathbf{z}\left(t\right),\mathbf{x}\left(t\right),\mathbf{u}\left(t\right)\right) \leq 0 \\ & \mathbf{x}(0) - \hat{\mathbf{x}}(t) = 0\end{aligned}$$

from the current state estimation $\hat{\mathbf{x}}(t)$.

How to impose the DAE consistency condition ? See previous slide: the initial conditions $\hat{\mathbf{x}}(t)$ assigned to the NMPC must be consistent... how ?

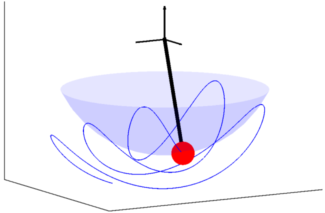


Online Optimal Control with Index-reduced DAEs

NMPC: OCP repeatedly solved online

$$\begin{aligned} \min_{\mathbf{x}(.),\mathbf{z}(.),\mathbf{u}(.)} \quad & T\left(\mathbf{x}\left(t_{\mathrm{f}}\right)\right) + \int_0^{t_{\mathrm{f}}} L\left(\mathbf{x},\mathbf{u}\right)\mathrm{d}\tau \\ \text{s.t.} \quad & \mathbf{F}\left(\dot{\mathbf{x}}\left(t\right),\mathbf{z}\left(t\right),\mathbf{x}\left(t\right),\mathbf{u}\left(t\right)\right) = 0 \\ & \mathbf{h}\left(\mathbf{z}\left(t\right),\mathbf{x}\left(t\right),\mathbf{u}\left(t\right)\right) \leq 0 \\ & \mathbf{x}(0) - \hat{\mathbf{x}}(t) = 0 \end{aligned}$$

from the current state estimation $\hat{\mathbf{x}}(t)$.



How to impose the DAE consistency condition ? See previous slide: the initial conditions $\hat{\mathbf{x}}(t)$ assigned to the NMPC must be consistent... how ?

When deploying **NMPC** with an underlying **index-reduced DAE model**, the consistency of the initial condition must be achieved in the **state-estimation algorithm** (Kalman filter, EKF, MHE, particle filter)

Online Optimal Control with Index-reduced DAEs

NMPC: OCP repeatedly solved online

$$
\begin{aligned}
\min_{\mathbf{x}(.),\mathbf{z}(.),\mathbf{u}(.)} \quad & T\left(\mathbf{x}\left(t_{\mathrm{f}}\right)\right) + \int_0^{t_{\mathrm{f}}} L\left(\mathbf{x},\mathbf{u}\right)\mathrm{d}\tau \\
\text{s.t.} \quad & \mathbf{F}\left(\dot{\mathbf{x}}\left(t\right),\mathbf{z}\left(t\right),\mathbf{x}\left(t\right),\mathbf{u}\left(t\right)\right) = 0 \\
& \mathbf{h}\left(\mathbf{z}\left(t\right),\mathbf{x}\left(t\right),\mathbf{u}\left(t\right)\right) \leq 0 \\
& \mathbf{x}(0) - \hat{\mathbf{x}}(t) = 0
\end{aligned}
$$

from the current state estimation $\hat{\mathbf{x}}(t)$.

E.g. MHE provides $\hat{\mathbf{x}}(t)$ via:

$$
\begin{aligned}
\min_{\hat{\mathbf{x}}(.),\hat{\mathbf{z}}(.),\hat{\mathbf{u}}(.)} \quad & \int_{t-\hat{T}}^{t} \left\|\mathbf{y}\left(\hat{\mathbf{x}},\hat{\mathbf{z}},\hat{\mathbf{u}}\right) - \bar{\mathbf{y}}\right\|^2 \mathrm{d}\tau \\
\text{s.t.} \quad & \mathbf{F}\left(\dot{\hat{\mathbf{x}}},\hat{\mathbf{z}},\hat{\mathbf{x}},\hat{\mathbf{u}}\right) = 0 \\
& \mathbf{C}\left(\hat{\mathbf{x}}(t)\right) = 0
\end{aligned}
$$

How to impose the DAE consistency condition ? See previous slide: the initial conditions $\hat{\mathbf{x}}(t)$ assigned to the NMPC must be consistent... how ?

When deploying **NMPC** with an underlying **index-reduced DAE model**, the consistency of the initial condition must be achieved in the **state-estimation algorithm** (Kalman filter, EKF, MHE, particle filter)

Online Optimal Control with Index-reduced DAEs

NMPC: OCP repeatedly solved online

$$
\begin{aligned}
\min_{\mathbf{x}(.),\mathbf{z}(.),\mathbf{u}(.)} \quad & T\left(\mathbf{x}\left(t_\mathrm{f}\right)\right) + \int_0^{t_\mathrm{f}} L\left(\mathbf{x},\mathbf{u}\right)\mathrm{d}\tau \\
\text{s.t.} \quad & \mathbf{F}\left(\dot{\mathbf{x}}\left(t\right),\mathbf{z}\left(t\right),\mathbf{x}\left(t\right),\mathbf{u}\left(t\right)\right) = 0 \\
& \mathbf{h}\left(\mathbf{z}\left(t\right),\mathbf{x}\left(t\right),\mathbf{u}\left(t\right)\right) \leq 0 \\
& \mathbf{x}(0) - \hat{\mathbf{x}}(t) = 0
\end{aligned}
$$

from the current state estimation $\hat{\mathbf{x}}(t)$.

How to impose the DAE consistency condition ? See previous slide: the initial conditions $\hat{\mathbf{x}}(t)$ assigned to the NMPC must be consistent... how ?

When deploying **NMPC** with an underlying **index-reduced DAE model**, the consistency of the initial condition must be achieved in the **state-estimation algorithm** (Kalman filter, EKF, MHE, particle filter)

E.g. MHE provides $\hat{\mathbf{x}}(t)$ via:

$$
\begin{aligned}
\min_{\hat{\mathbf{x}}(.),\hat{\mathbf{z}}(.),\hat{\mathbf{u}}(.)} \quad & \int_{t-\hat{T}}^{t} \left\|\mathbf{y}\left(\hat{\mathbf{x}},\hat{\mathbf{z}},\hat{\mathbf{u}}\right) - \bar{\mathbf{y}}\right\|^2 \mathrm{d}\tau \\
\text{s.t.} \quad & \mathbf{F}\left(\dot{\hat{\mathbf{x}}},\hat{\mathbf{z}},\hat{\mathbf{x}},\hat{\mathbf{u}}\right) = 0 \\
& \mathbf{C}\left(\hat{\mathbf{x}}(t)\right) = 0
\end{aligned}
$$

- Constraint $\mathbf{C}\left(\hat{\mathbf{x}}(t)\right) = 0$ ensures a consistent state estimation

Online Optimal Control with Index-reduced DAEs

NMPC: OCP repeatedly solved online

$$\begin{aligned}\min_{\mathbf{x}(.),\mathbf{z}(.),\mathbf{u}(.)} \quad & T\left(\mathbf{x}\left(t_\mathrm{f}\right)\right) + \int_0^{t_\mathrm{f}} L\left(\mathbf{x},\mathbf{u}\right)\mathrm{d}\tau \\ \text{s.t.} \quad & \mathbf{F}\left(\dot{\mathbf{x}}\left(t\right),\mathbf{z}\left(t\right),\mathbf{x}\left(t\right),\mathbf{u}\left(t\right)\right) = 0 \\ & \mathbf{h}\left(\mathbf{z}\left(t\right),\mathbf{x}\left(t\right),\mathbf{u}\left(t\right)\right) \leq 0 \\ & \mathbf{x}(0) - \hat{\mathbf{x}}(t) = 0\end{aligned}$$

from the current state estimation $\hat{\mathbf{x}}(t)$.

How to impose the DAE consistency condition ? See previous slide: the initial conditions $\hat{\mathbf{x}}(t)$ assigned to the NMPC must be consistent... how ?

When deploying **NMPC** with an underlying **index-reduced DAE model**, the consistency of the initial condition must be achieved in the **state-estimation algorithm** (Kalman filter, EKF, MHE, particle filter)

E.g. MHE provides $\hat{\mathbf{x}}(t)$ via:

$$\begin{aligned}\min_{\hat{\mathbf{x}}(.),\hat{\mathbf{z}}(.),\hat{\mathbf{u}}(.)} \quad & \int_{t-\hat{T}}^{t} \left\|\mathbf{y}\left(\hat{\mathbf{x}},\hat{\mathbf{z}},\hat{\mathbf{u}}\right) - \bar{\mathbf{y}}\right\|^2 \mathrm{d}\tau \\ \text{s.t.} \quad & \mathbf{F}\left(\dot{\hat{\mathbf{x}}},\hat{\mathbf{z}},\hat{\mathbf{x}},\hat{\mathbf{u}}\right) = 0 \\ & \mathbf{C}\left(\hat{\mathbf{x}}(t)\right) = 0\end{aligned}$$

- Constraint $\mathbf{C}\left(\hat{\mathbf{x}}(t)\right) = 0$ ensures a consistent state estimation
- Note that consistency is imposed **at the end** of the estimation horizon so as to maximize its numecial accuracy (e.g. imposing the consistency at time $t - \hat{T}$ would let numerical errors accumulate in the integration).

Outline

Multiple-Shooting for DAE-constrained OCPs

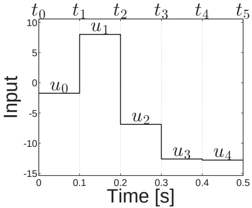
Integrator for index-1 DAE:

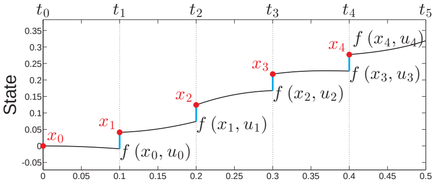
$$\mathbf{F}\left(\dot{\mathbf{x}}\left(t\right),\mathbf{z}\left(t\right),\mathbf{x}\left(t\right),\mathbf{u}\left(t\right)\right)=0$$

Provides the function:

$$\mathbf{f}\left(\mathbf{x}_k,\mathbf{u}_k\right)$$

delivering the integration of the DAE over a time interval $[t_k,\ t_{k+1}]$.





Integrations on the time intervals $[t_k,\ t_{k+1}]$

Multiple-Shooting for DAE-constrained OCPs

Integrator for index-1 DAE:

$$\mathbf{F}\left(\dot{\mathbf{x}}\left(t\right),\mathbf{z}\left(t\right),\mathbf{x}\left(t\right),\mathbf{u}\left(t\right)\right)=0$$

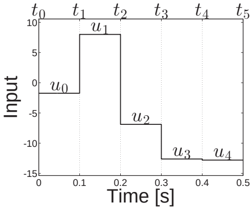
Provides the function:

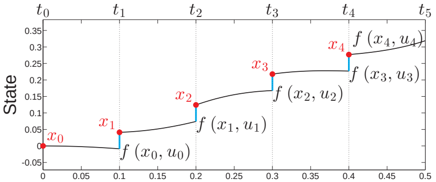
$$\mathbf{f}\left(\mathbf{x}_k,\mathbf{u}_k\right)$$

delivering the integration of the DAE over a time interval $[t_k,\ t_{k+1}]$.

E.g. semi-explicit DAE:

$$\begin{aligned}\dot{\mathbf{x}}\left(t\right)&=\mathbf{F}\left(\mathbf{z}\left(t\right),\mathbf{x}\left(t\right),\mathbf{u}\left(t\right)\right)\\0&=\mathbf{G}\left(\mathbf{z}\left(t\right),\mathbf{x}\left(t\right),\mathbf{u}\left(t\right)\right)\end{aligned}$$





Integrations on the time intervals $[t_k,\ t_{k+1}]$

Multiple-Shooting for DAE-constrained OCPs

Integrator for index-1 DAE:

$$\mathbf{F}\left(\dot{\mathbf{x}}\left(t\right), \mathbf{z}\left(t\right), \mathbf{x}\left(t\right), \mathbf{u}\left(t\right)\right) = 0$$

Provides the function:

$$\mathbf{f}\left(\mathbf{x}_k, \mathbf{u}_k\right)$$

delivering the integration of the DAE over a time interval $[t_k,\ t_{k+1}]$.

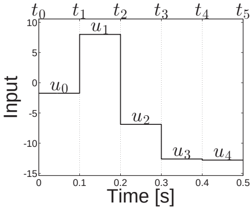
E.g. semi-explicit DAE:

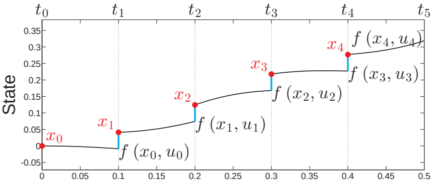
$$\begin{aligned}\dot{\mathbf{x}}\left(t\right) &= \mathbf{F}\left(\mathbf{z}\left(t\right), \mathbf{x}\left(t\right), \mathbf{u}\left(t\right)\right)\\ 0 &= \mathbf{G}\left(\mathbf{z}\left(t\right), \mathbf{x}\left(t\right), \mathbf{u}\left(t\right)\right)\end{aligned}$$

with one-step implicit Euler:

- Solve for $\mathbf{x}_+, \mathbf{z}_+$:

$$\begin{aligned}\mathbf{x}_+ &= \mathbf{x}_k + h\mathbf{F}\left(\mathbf{x}_+, \mathbf{z}_+, \mathbf{u}_k\right)\\ 0 &= \mathbf{G}\left(\mathbf{x}_+, \mathbf{z}_+, \mathbf{u}_k\right)\end{aligned}$$





Integrations on the time intervals $[t_k,\ t_{k+1}]$

Multiple-Shooting for DAE-constrained OCPs

Integrator for index-1 DAE:

$$\mathbf{F}\left(\dot{\mathbf{x}}\left(t\right),\mathbf{z}\left(t\right),\mathbf{x}\left(t\right),\mathbf{u}\left(t\right)\right)=0$$

Provides the function:

$$\mathbf{f}\left(\mathbf{x}_k,\mathbf{u}_k\right)$$

delivering the integration of the DAE over a time interval $[t_k,\ t_{k+1}]$.

E.g. semi-explicit DAE:

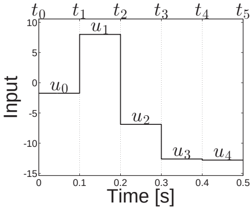
$$\begin{aligned}\dot{\mathbf{x}}\left(t\right)&=\mathbf{F}\left(\mathbf{z}\left(t\right),\mathbf{x}\left(t\right),\mathbf{u}\left(t\right)\right)\\ 0&=\mathbf{G}\left(\mathbf{z}\left(t\right),\mathbf{x}\left(t\right),\mathbf{u}\left(t\right)\right)\end{aligned}$$

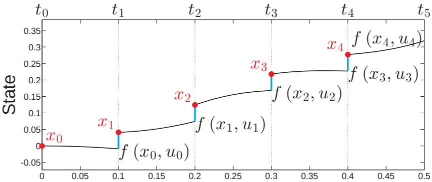
with one-step implicit Euler:

- Solve for $\mathbf{x}_+,\mathbf{z}_+$:

$$\begin{aligned}\mathbf{x}_+&=\mathbf{x}_k+h\mathbf{F}\left(\mathbf{x}_+,\mathbf{z}_+,\mathbf{u}_k\right)\\ 0&=\mathbf{G}\left(\mathbf{x}_+,\mathbf{z}_+,\mathbf{u}_k\right)\end{aligned}$$

- Return $\mathbf{f}\left(\mathbf{x}_k,\mathbf{u}_k\right)\equiv\mathbf{x}_+$





Integrations on the time intervals $[t_k,\ t_{k+1}]$

Multiple-Shooting for DAE-constrained OCPs

Integrator for index-1 DAE:

$$\mathbf{F}\left(\dot{\mathbf{x}}(t), \mathbf{z}(t), \mathbf{x}(t), \mathbf{u}(t)\right) = 0$$

Provides the function:

$$\mathbf{f}\left(\mathbf{x}_k, \mathbf{u}_k\right)$$

delivering the integration of the DAE over a time interval $[t_k,\ t_{k+1}]$.

Note that the integrator "eliminates" the algebraic variables $\mathbf{z}(.)$ by treating them "internally" !! We have some "hidden" complexity...

E.g. semi-explicit DAE:

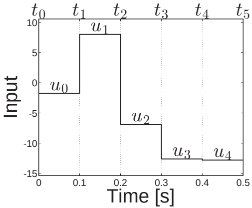
$$\begin{aligned} \dot{\mathbf{x}}(t) &= \mathbf{F}\left(\mathbf{z}(t), \mathbf{x}(t), \mathbf{u}(t)\right) \\ 0 &= \mathbf{G}\left(\mathbf{z}(t), \mathbf{x}(t), \mathbf{u}(t)\right) \end{aligned}$$

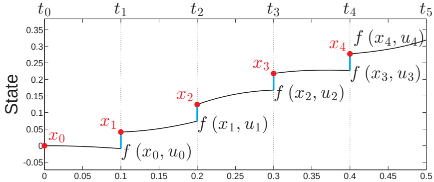
with one-step implicit Euler:

- Solve for $\mathbf{x}_+, \mathbf{z}_+$:

$$\begin{aligned} \mathbf{x}_+ &= \mathbf{x}_k + h\mathbf{F}\left(\mathbf{x}_+, \mathbf{z}_+, \mathbf{u}_k\right) \\ 0 &= \mathbf{G}\left(\mathbf{x}_+, \mathbf{z}_+, \mathbf{u}_k\right) \end{aligned}$$

- Return $\mathbf{f}\left(\mathbf{x}_k, \mathbf{u}_k\right) \equiv \mathbf{x}_+$



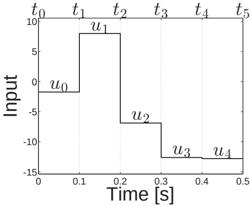


Integrations on the time intervals $[t_k,\ t_{k+1}]$

NLP from Multiple-Shooting

OCP:

$$\begin{aligned} \min \quad & \Phi\left(\mathbf{x}(.), \mathbf{u}(.)\right) \\ \text{s.t.} \quad & \mathbf{F}\left(\dot{\mathbf{x}}\left(t\right), \mathbf{z}\left(t\right), \mathbf{x}\left(t\right), \mathbf{u}\left(t\right)\right) = 0 \\ & \mathbf{h}\left(\mathbf{x}(t), \mathbf{u}(t), t\right) \leq 0 \\ & \mathbf{x}\left(t_0\right) = \bar{\mathbf{x}}_0 \end{aligned}$$

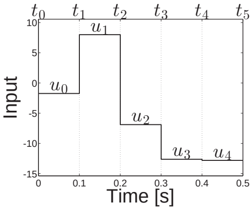


NLP from Multiple-Shooting

OCP:

$$\begin{aligned} \min \quad & \Phi\left(\mathbf{x}(.), \mathbf{u}(.)\right) \\ \text{s.t.} \quad & \mathbf{F}\left(\dot{\mathbf{x}}\left(t\right), \mathbf{z}\left(t\right), \mathbf{x}\left(t\right), \mathbf{u}\left(t\right)\right) = 0 \\ & \mathbf{h}\left(\mathbf{x}(t), \mathbf{u}(t), t\right) \leq 0 \\ & \mathbf{x}\left(t_0\right) = \bar{\mathbf{x}}_0 \end{aligned}$$

$\mathbf{f}\left(\mathbf{x}_k, \mathbf{u}_k\right)$ integrates the dynamics over the time interval $\left[t_k,\ t_{k+1}\right]$

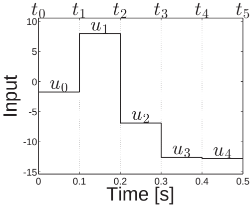


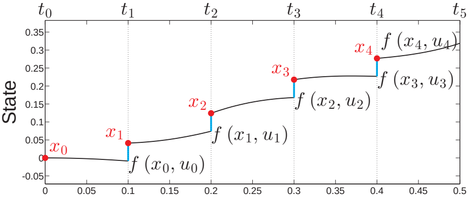
NLP from Multiple-Shooting

OCP:

$$\begin{aligned} \min \quad & \Phi\left(\mathbf{x}(.), \mathbf{u}(.)\right) \\ \text{s.t.} \quad & \mathbf{F}\left(\dot{\mathbf{x}}\left(t\right), \mathbf{z}\left(t\right), \mathbf{x}\left(t\right), \mathbf{u}\left(t\right)\right) = 0 \\ & \mathbf{h}\left(\mathbf{x}(t), \mathbf{u}(t), t\right) \leq 0 \\ & \mathbf{x}\left(t_0\right) = \bar{\mathbf{x}}_0 \end{aligned}$$

$\mathbf{f}\left(\mathbf{x}_k, \mathbf{u}_k\right)$ integrates the dynamics over the time interval $[t_k,\ t_{k+1}]$





NLP from Multiple-Shooting

OCP:

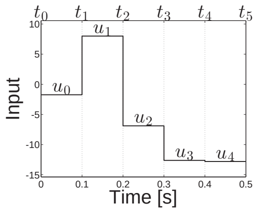
$$
\begin{aligned}
\min \quad & \Phi\left(\mathbf{x}(.), \mathbf{u}(.)\right) \\
\text{s.t.} \quad & \mathbf{F}\left(\dot{\mathbf{x}}\left(t\right), \mathbf{z}\left(t\right), \mathbf{x}\left(t\right), \mathbf{u}\left(t\right)\right) = 0 \\
& \mathbf{h}\left(\mathbf{x}(t), \mathbf{u}(t), t\right) \leq 0 \\
& \mathbf{x}\left(t_0\right) = \bar{\mathbf{x}}_0
\end{aligned}
$$

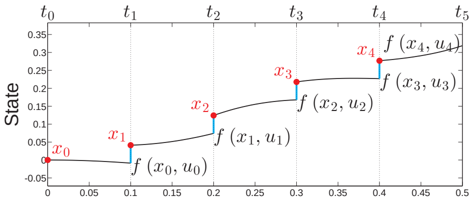
$\mathbf{f}\left(\mathbf{x}_k, \mathbf{u}_k\right)$ integrates the dynamics over the time interval $[t_k,\ t_{k+1}]$

NLP with $\mathbf{w} = \{\mathbf{x}_0, \mathbf{u}_0, ..., \mathbf{x}_{N-1}, \mathbf{u}_{N-1}, \mathbf{x}_N\}$

$$
\min_{\mathbf{w}} \quad \Phi\left(\mathbf{w}\right)
$$

s.t.





NLP from Multiple-Shooting

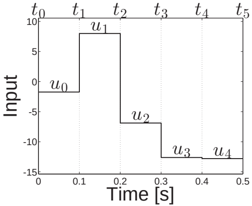
OCP:

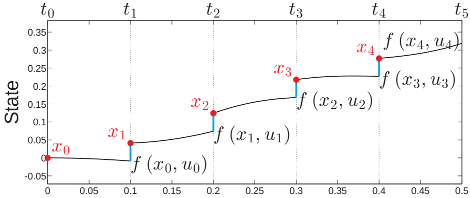
$$\begin{aligned} \min \quad & \Phi\left(\mathbf{x}(.), \mathbf{u}(.)\right) \\ \text{s.t.} \quad & \mathbf{F}\left(\dot{\mathbf{x}}\left(t\right), \mathbf{z}\left(t\right), \mathbf{x}\left(t\right), \mathbf{u}\left(t\right)\right) = 0 \\ & \mathbf{h}\left(\mathbf{x}(t), \mathbf{u}(t), t\right) \leq 0 \\ & \mathbf{x}\left(t_0\right) = \bar{\mathbf{x}}_0 \end{aligned}$$

$\mathbf{f}\left(\mathbf{x}_k, \mathbf{u}_k\right)$ integrates the dynamics over the time interval $[t_k,\ t_{k+1}]$

NLP with $\mathbf{w} = \{\mathbf{x}_0, \mathbf{u}_0, ..., \mathbf{x}_{N-1}, \mathbf{u}_{N-1}, \mathbf{x}_N\}$

$$\begin{aligned} \min_{\mathbf{w}} \quad & \Phi\left(\mathbf{w}\right) \\ \text{s.t.} \quad & \mathbf{g}\left(\mathbf{w}\right) = \begin{bmatrix} \bar{\mathbf{x}}_0 - \mathbf{x}_0 \\ \mathbf{f}\left(\mathbf{x}_0, \mathbf{u}_0\right) - \mathbf{x}_1 \\ \dots \\ \mathbf{f}\left(\mathbf{x}_N, \mathbf{u}_{N-1}\right) - \mathbf{x}_{N-1} \end{bmatrix} = 0 \end{aligned}$$





NLP from Multiple-Shooting

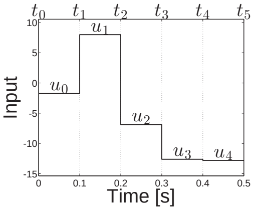
OCP:

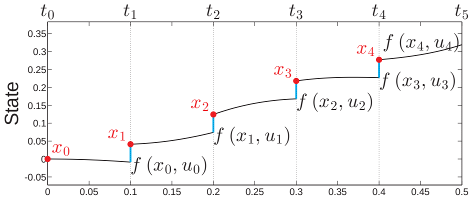
$$\begin{aligned} \min \quad & \Phi\left(\mathbf{x}(.), \mathbf{u}(.)\right) \\ \text{s.t.} \quad & \mathbf{F}\left(\dot{\mathbf{x}}\left(t\right), \mathbf{z}\left(t\right), \mathbf{x}\left(t\right), \mathbf{u}\left(t\right)\right) = 0 \\ & \mathbf{h}\left(\mathbf{x}(t), \mathbf{u}(t), t\right) \leq 0 \\ & \mathbf{x}\left(t_0\right) = \bar{\mathbf{x}}_0 \end{aligned}$$

$\mathbf{f}\left(\mathbf{x}_k, \mathbf{u}_k\right)$ integrates the dynamics over the time interval $[t_k,\ t_{k+1}]$

NLP with $\mathbf{w} = \{\mathbf{x}_0, \mathbf{u}_0, ..., \mathbf{x}_{N-1}, \mathbf{u}_{N-1}, \mathbf{x}_N\}$

$$\begin{aligned} \min_{\mathbf{w}} \quad & \Phi\left(\mathbf{w}\right) \\ \text{s.t.} \quad & \mathbf{g}\left(\mathbf{w}\right) = \begin{bmatrix} \bar{\mathbf{x}}_0 - \mathbf{x}_0 \\ \mathbf{f}\left(\mathbf{x}_0, \mathbf{u}_0\right) - \mathbf{x}_1 \\ \dots \\ \mathbf{f}\left(\mathbf{x}_N, \mathbf{u}_{N-1}\right) - \mathbf{x}_{N-1} \end{bmatrix} = 0 \\ & \mathbf{h}\left(\mathbf{w}\right) = \begin{bmatrix} \mathbf{h}\left(\mathbf{x}_0, \mathbf{u}_0\right) \\ \dots \\ \mathbf{h}\left(\mathbf{x}_{N-1}, \mathbf{u}_{N-1}\right) \\ \mathbf{h}\left(\mathbf{x}_N\right) \end{bmatrix} \leq 0 \end{aligned}$$





NLP from Multiple-Shooting

OCP:

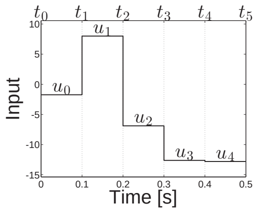
$$\begin{aligned} \min \quad & \Phi\left(\mathbf{x}(.), \mathbf{u}(.)\right) \\ \text{s.t.} \quad & \mathbf{F}\left(\dot{\mathbf{x}}(t), \mathbf{z}(t), \mathbf{x}(t), \mathbf{u}(t)\right) = 0 \\ & \mathbf{h}\left(\mathbf{x}(t), \mathbf{u}(t), t\right) \leq 0 \\ & \mathbf{x}(t_0) = \bar{\mathbf{x}}_0 \end{aligned}$$

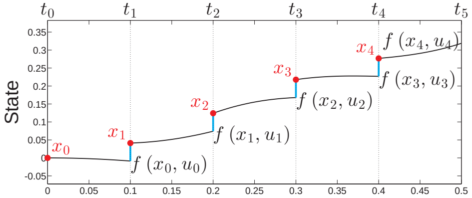
$\mathbf{f}(\mathbf{x}_k, \mathbf{u}_k)$ integrates the dynamics over the time interval $[t_k,\ t_{k+1}]$

Algebraic variables are hidden within the integrator... Is that the end of the story ?

NLP with $\mathbf{w} = \{\mathbf{x}_0, \mathbf{u}_0, ..., \mathbf{x}_{N-1}, \mathbf{u}_{N-1}, \mathbf{x}_N\}$

$$\begin{aligned} \min_{\mathbf{w}} \quad & \Phi(\mathbf{w}) \\ \text{s.t.} \quad & \mathbf{g}(\mathbf{w}) = \begin{bmatrix} \bar{\mathbf{x}}_0 - \mathbf{x}_0 \\ \mathbf{f}(\mathbf{x}_0, \mathbf{u}_0) - \mathbf{x}_1 \\ \dots \\ \mathbf{f}(\mathbf{x}_N, \mathbf{u}_{N-1}) - \mathbf{x}_{N-1} \end{bmatrix} = 0 \\ & \mathbf{h}(\mathbf{w}) = \begin{bmatrix} \mathbf{h}(\mathbf{x}_0, \mathbf{u}_0) \\ \dots \\ \mathbf{h}(\mathbf{x}_{N-1}, \mathbf{u}_{N-1}) \\ \mathbf{h}(\mathbf{x}_N) \end{bmatrix} \leq 0 \end{aligned}$$





NLP from Multiple-Shooting

OCP:

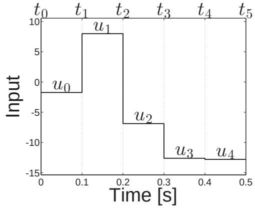
$$\begin{aligned} \min \quad & \Phi\left(\mathbf{x}(.), \mathbf{u}(.)\right) \\ \text{s.t.} \quad & \mathbf{F}\left(\dot{\mathbf{x}}\left(t\right), \mathbf{z}\left(t\right), \mathbf{x}\left(t\right), \mathbf{u}\left(t\right)\right) = 0 \\ & \mathbf{h}\left(\mathbf{x}(t), \mathbf{u}(t), t\right) \leq 0 \\ & \mathbf{x}\left(t_0\right) = \bar{\mathbf{x}}_0 \end{aligned}$$

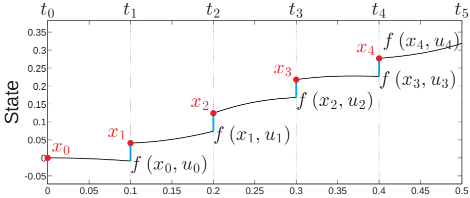
$\mathbf{f}\left(\mathbf{x}_k, \mathbf{u}_k\right)$ integrates the dynamics over the time interval $[t_k,\ t_{k+1}]$

Algebraic variables are hidden within the integrator... Is that the end of the story ? Not necessarily...

NLP with $\mathbf{w} = \left\{\mathbf{x}_0, \mathbf{u}_0, ..., \mathbf{x}_{N-1}, \mathbf{u}_{N-1}, \mathbf{x}_N\right\}$

$$\begin{aligned} \min_{\mathbf{w}} \quad & \Phi\left(\mathbf{w}\right) \\ \text{s.t.} \quad & \mathbf{g}\left(\mathbf{w}\right) = \begin{bmatrix} \bar{\mathbf{x}}_0 - \mathbf{x}_0 \\ \mathbf{f}\left(\mathbf{x}_0, \mathbf{u}_0\right) - \mathbf{x}_1 \\ \dots \\ \mathbf{f}\left(\mathbf{x}_N, \mathbf{u}_{N-1}\right) - \mathbf{x}_{N-1} \end{bmatrix} = 0 \\ & \mathbf{h}\left(\mathbf{w}\right) = \begin{bmatrix} \mathbf{h}\left(\mathbf{x}_0, \mathbf{u}_0\right) \\ \dots \\ \mathbf{h}\left(\mathbf{x}_{N-1}, \mathbf{u}_{N-1}\right) \\ \mathbf{h}\left(\mathbf{x}_N\right) \end{bmatrix} \leq 0 \end{aligned}$$



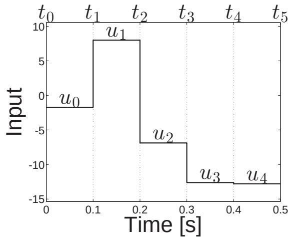


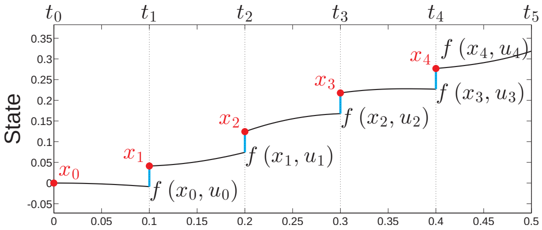
Algebraic variables in the cost & inequality constraints

OCP:

$$\begin{aligned} \min \quad & \Phi\left(\mathbf{x}(.), \mathbf{u}(.)\right) \\ \text{s.t.} \quad & \mathbf{F}\left(\dot{\mathbf{x}}\left(t\right), \mathbf{z}\left(t\right), \mathbf{x}\left(t\right), \mathbf{u}\left(t\right)\right) = 0 \\ & \mathbf{h}\left(\mathbf{x}(t), \mathbf{u}(t)\right) \leq 0 \\ & \mathbf{x}\left(t_0\right) = \bar{\mathbf{x}}_0 \end{aligned}$$

$\mathbf{f}\left(\mathbf{x}_k, \mathbf{u}_k\right)$ integrates the dynamics $\mathbf{F}$ over the time intervals $[t_k,\ t_{k+1}]$, provides the state $\mathbf{x}$ at t_{k+1}.





Algebraic variables in the cost & inequality constraints

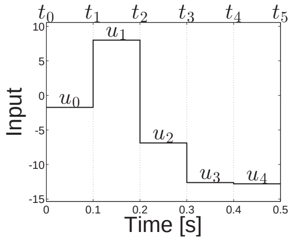
OCP:

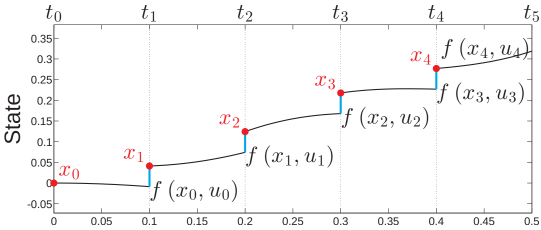
$$\begin{aligned} \min \quad & \Phi\left(\mathbf{x}(.), \mathbf{u}(.)\right) \\ \text{s.t.} \quad & \mathbf{F}\left(\dot{\mathbf{x}}\left(t\right), \mathbf{z}\left(t\right), \mathbf{x}\left(t\right), \mathbf{u}\left(t\right)\right) = 0 \\ & \mathbf{h}\left(\mathbf{x}(t), \mathbf{u}(t)\right) \leq 0 \\ & \mathbf{x}\left(t_0\right) = \bar{\mathbf{x}}_0 \end{aligned}$$

$\mathbf{f}\left(\mathbf{x}_k, \mathbf{u}_k\right)$ integrates the dynamics $\mathbf{F}$ over the time intervals $[t_k,\ t_{k+1}]$, provides the state $\mathbf{x}$ at t_{k+1}.

OCP:

$$\begin{aligned} \min \quad & \Phi\left(\mathbf{x}(.), \mathbf{z}(.), \mathbf{u}(.)\right) \\ \text{s.t.} \quad & \mathbf{F}\left(\dot{\mathbf{x}}\left(t\right), \mathbf{z}\left(t\right), \mathbf{x}\left(t\right), \mathbf{u}\left(t\right)\right) = 0 \\ & \mathbf{h}\left(\mathbf{x}(t), \mathbf{z}(t), \mathbf{u}(t)\right) \leq 0 \\ & \mathbf{x}\left(t_0\right) - \bar{\mathbf{x}}_0 = 0 \end{aligned}$$





Algebraic variables in the cost & inequality constraints

OCP:

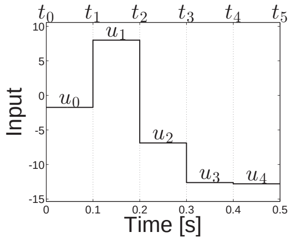
$$\begin{aligned} \min \quad & \Phi\left(\mathbf{x}(.), \mathbf{u}(.)\right) \\ \text{s.t.} \quad & \mathbf{F}\left(\dot{\mathbf{x}}(t), \mathbf{z}(t), \mathbf{x}(t), \mathbf{u}(t)\right) = 0 \\ & \mathbf{h}\left(\mathbf{x}(t), \mathbf{u}(t)\right) \leq 0 \\ & \mathbf{x}(t_0) = \bar{\mathbf{x}}_0 \end{aligned}$$

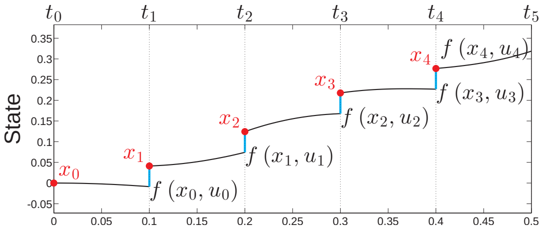
$\mathbf{f}\left(\mathbf{x}_k, \mathbf{u}_k\right)$ integrates the dynamics $\mathbf{F}$ over the time intervals $[t_k,\ t_{k+1}]$, provides the state $\mathbf{x}$ at t_{k+1}.

OCP:

$$\begin{aligned} \min \quad & \Phi\left(\mathbf{x}(.), \mathbf{z}(.), \mathbf{u}(.)\right) \\ \text{s.t.} \quad & \mathbf{F}\left(\dot{\mathbf{x}}(t), \mathbf{z}(t), \mathbf{x}(t), \mathbf{u}(t)\right) = 0 \\ & \mathbf{h}\left(\mathbf{x}(t), \mathbf{z}(t), \mathbf{u}(t)\right) \leq 0 \\ & \mathbf{x}(t_0) - \bar{\mathbf{x}}_0 = 0 \end{aligned}$$

Then the integrator needs to **report the algebraic variables** $\mathbf{z}(.)$ as well...





Algebraic variables in the cost & inequality constraints

OCP:

$$\begin{aligned} \min \quad & \Phi\left(\mathbf{x}(.), \mathbf{u}(.)\right) \\ \text{s.t.} \quad & \mathbf{F}\left(\dot{\mathbf{x}}\left(t\right), \mathbf{z}\left(t\right), \mathbf{x}\left(t\right), \mathbf{u}\left(t\right)\right) = 0 \\ & \mathbf{h}\left(\mathbf{x}(t), \mathbf{u}(t)\right) \leq 0 \\ & \mathbf{x}\left(t_0\right) = \bar{\mathbf{x}}_0 \end{aligned}$$

$\mathbf{f}\left(\mathbf{x}_k, \mathbf{u}_k\right)$ integrates the dynamics $\mathbf{F}$ over the time intervals $[t_k,\ t_{k+1}]$, provides the state $\mathbf{x}$ at t_{k+1}.

E.g. semi-explicit DAE ...:

$$\begin{aligned} \dot{\mathbf{x}}\left(t\right) &= \mathbf{F}\left(\mathbf{z}\left(t\right), \mathbf{x}\left(t\right), \mathbf{u}\left(t\right)\right) \\ 0 &= \mathbf{G}\left(\mathbf{z}\left(t\right), \mathbf{x}\left(t\right), \mathbf{u}\left(t\right)\right) \end{aligned}$$

OCP:

$$\begin{aligned} \min \quad & \Phi\left(\mathbf{x}(.), \mathbf{z}(.), \mathbf{u}(.)\right) \\ \text{s.t.} \quad & \mathbf{F}\left(\dot{\mathbf{x}}\left(t\right), \mathbf{z}\left(t\right), \mathbf{x}\left(t\right), \mathbf{u}\left(t\right)\right) = 0 \\ & \mathbf{h}\left(\mathbf{x}(t), \mathbf{z}(t), \mathbf{u}(t)\right) \leq 0 \\ & \mathbf{x}\left(t_0\right) - \bar{\mathbf{x}}_0 = 0 \end{aligned}$$

Then the integrator needs to **report the algebraic variables** $\mathbf{z}\left(.\right)$ as well...

Algebraic variables in the cost & inequality constraints

OCP:

$$\begin{aligned} \min \quad & \Phi\left(\mathbf{x}(.), \mathbf{u}(.)\right) \\ \text{s.t.} \quad & \mathbf{F}\left(\dot{\mathbf{x}}(t), \mathbf{z}(t), \mathbf{x}(t), \mathbf{u}(t)\right) = 0 \\ & \mathbf{h}\left(\mathbf{x}(t), \mathbf{u}(t)\right) \leq 0 \\ & \mathbf{x}(t_0) = \bar{\mathbf{x}}_0 \end{aligned}$$

$\mathbf{f}\left(\mathbf{x}_k, \mathbf{u}_k\right)$ integrates the dynamics $\mathbf{F}$ over the time intervals $[t_k,\ t_{k+1}]$, provides the state $\mathbf{x}$ at t_{k+1}.

OCP:

$$\begin{aligned} \min \quad & \Phi\left(\mathbf{x}(.), \mathbf{z}(.), \mathbf{u}(.)\right) \\ \text{s.t.} \quad & \mathbf{F}\left(\dot{\mathbf{x}}(t), \mathbf{z}(t), \mathbf{x}(t), \mathbf{u}(t)\right) = 0 \\ & \mathbf{h}\left(\mathbf{x}(t), \mathbf{z}(t), \mathbf{u}(t)\right) \leq 0 \\ & \mathbf{x}(t_0) - \bar{\mathbf{x}}_0 = 0 \end{aligned}$$

Then the integrator needs to **report the algebraic variables** $\mathbf{z}(.)$ as well...

E.g. semi-explicit DAE ...:

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \mathbf{F}\left(\mathbf{z}(t), \mathbf{x}(t), \mathbf{u}(t)\right) \\ 0 &= \mathbf{G}\left(\mathbf{z}(t), \mathbf{x}(t), \mathbf{u}(t)\right) \end{aligned}$$

... with one-step implicit Euler:

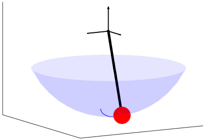
- Solve for $\mathbf{x}_+, \mathbf{z}_+$:

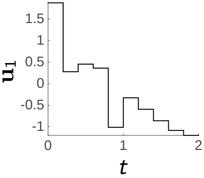
$$\begin{aligned} \mathbf{x}_+ &= \mathbf{x}_k + h\mathbf{F}\left(\mathbf{x}_+, \mathbf{z}_+, \mathbf{u}_k\right) \\ 0 &= \mathbf{G}\left(\mathbf{x}_+, \mathbf{z}_+, \mathbf{u}_k\right) \end{aligned}$$

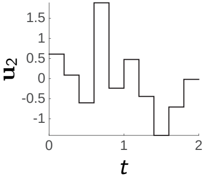
- Return $\mathbf{f}\left(\mathbf{x}_k, \mathbf{u}_k\right) \equiv \mathbf{x}_+, \quad \mathbf{z}_+$

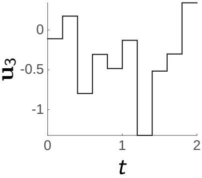
Algebraic variables & discrete inputs

3D pendulum with discretized inputs: (force on the mass)



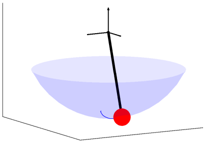


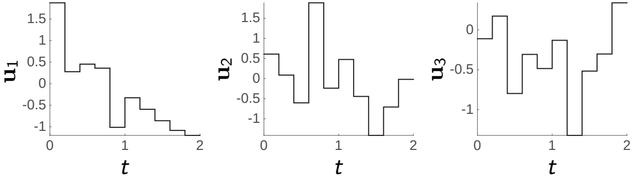




Algebraic variables & discrete inputs

3D pendulum with discretized inputs: (force on the mass)



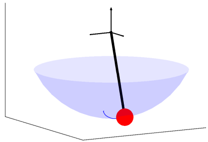


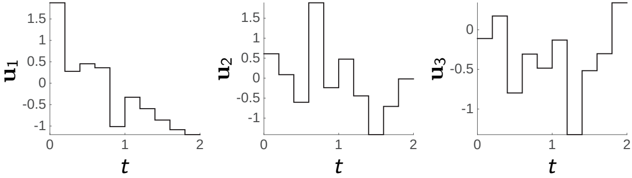
Index-1 DAE:

$$\begin{bmatrix} ml & \mathbf{p} \\ \mathbf{p}^\top & 0 \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{p}} \\ z \end{bmatrix} = \begin{bmatrix} \mathbf{u} - mg\mathbf{e}_3 \\ -\dot{\mathbf{p}}^\top \dot{\mathbf{p}} \end{bmatrix}$$

Algebraic variables & discrete inputs

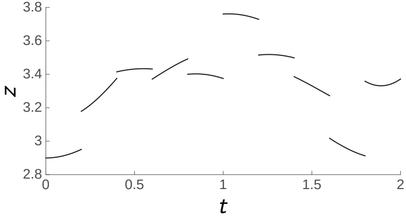
3D pendulum with discretized inputs: (force on the mass)





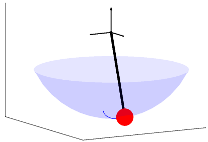
Index-1 DAE:

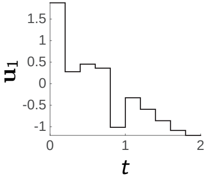
$$\begin{bmatrix} ml & \mathbf{p} \\ \mathbf{p}^\top & 0 \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{p}} \\ z \end{bmatrix} = \begin{bmatrix} \mathbf{u} - mg\mathbf{e}_3 \\ -\dot{\mathbf{p}}^\top \dot{\mathbf{p}} \end{bmatrix}$$

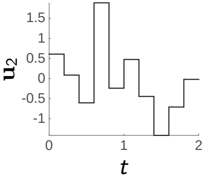


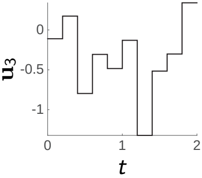
Algebraic variables & discrete inputs

3D pendulum with discretized inputs: (force on the mass)





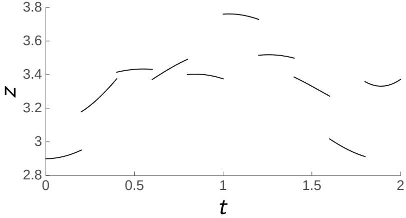




Index-1 DAE:

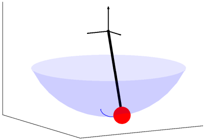
$$\begin{bmatrix} ml & \mathbf{p} \\ \mathbf{p}^\top & 0 \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{p}} \\ z \end{bmatrix} = \begin{bmatrix} \mathbf{u} - mg\mathbf{e}_3 \\ -\dot{\mathbf{p}}^\top \dot{\mathbf{p}} \end{bmatrix}$$

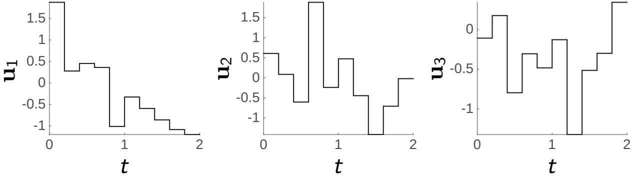
When using a discontinuous input parametrization, the algebraic variables **can** also be discontinuous !!



Algebraic variables & discrete inputs

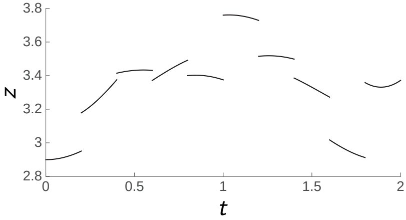
3D pendulum with discretized inputs: (force on the mass)





Index-1 DAE:

$$\begin{bmatrix} ml & \mathbf{p} \\ \mathbf{p}^\top & 0 \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{p}} \\ z \end{bmatrix} = \begin{bmatrix} \mathbf{u} - mg\mathbf{e}_3 \\ -\dot{\mathbf{p}}^\top \dot{\mathbf{p}} \end{bmatrix}$$



When using a discontinuous input parametrization, the algebraic variables **can** also be discontinuous !!

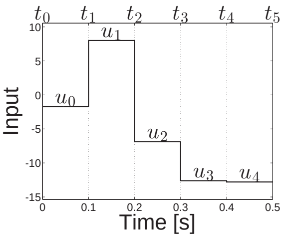
When ? Observe : $\dfrac{\partial}{\partial \mathbf{u}} \begin{bmatrix} \dot{\mathbf{x}} \\ \mathbf{z} \end{bmatrix} = - \begin{bmatrix} \frac{\partial \mathbf{F}}{\partial \dot{\mathbf{x}}} & \frac{\partial \mathbf{F}}{\partial \mathbf{z}} \end{bmatrix}^{-1} \dfrac{\partial \mathbf{F}}{\partial \mathbf{u}}$ $\quad \dfrac{\partial \mathbf{z}}{\partial \mathbf{u}} \neq 0 \quad \Rightarrow$ discontinuous z

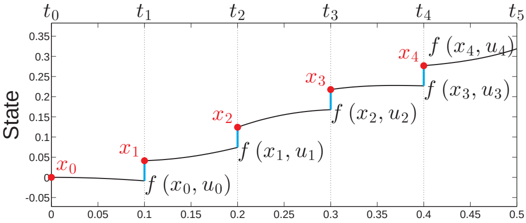
Algebraic variables in the cost & inequality constraints

OCP:

$$\begin{aligned}\min \quad & \Phi\left(\mathbf{x}(.), \mathbf{u}(.)\right) \\ \text{s.t.} \quad & \mathbf{F}\left(\dot{\mathbf{x}}\left(t\right), \mathbf{z}\left(t\right), \mathbf{x}\left(t\right), \mathbf{u}\left(t\right)\right) = 0 \\ & \mathbf{h}\left(\mathbf{x}(t), \mathbf{u}(t)\right) \leq 0 \\ & \mathbf{x}\left(t_0\right) = \bar{\mathbf{x}}_0\end{aligned}$$

$\mathbf{f}\left(\mathbf{x}_k, \mathbf{u}_k\right)$ integrates the dynamics $\mathbf{F}$ over the time intervals $[t_k, \; t_{k+1}]$, provides the state $\mathbf{x}$ at t_{k+1}.





Algebraic variables in the cost & inequality constraints

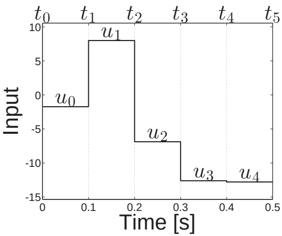
OCP:

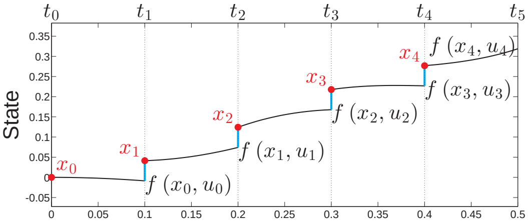
$$\begin{aligned} \min \quad & \Phi\left(\mathbf{x}(.), \mathbf{u}(.)\right) \\ \text{s.t.} \quad & \mathbf{F}\left(\dot{\mathbf{x}}(t), \mathbf{z}(t), \mathbf{x}(t), \mathbf{u}(t)\right) = 0 \\ & \mathbf{h}\left(\mathbf{x}(t), \mathbf{u}(t)\right) \leq 0 \\ & \mathbf{x}(t_0) = \bar{\mathbf{x}}_0 \end{aligned}$$

OCP:

$$\begin{aligned} \min \quad & \Phi\left(\mathbf{x}(.), \mathbf{z}(.), \mathbf{u}(.)\right) \\ \text{s.t.} \quad & \mathbf{F}\left(\dot{\mathbf{x}}(t), \mathbf{z}(t), \mathbf{x}(t), \mathbf{u}(t)\right) = 0 \\ & \mathbf{h}\left(\mathbf{x}(t), \mathbf{z}(t), \mathbf{u}(t)\right) \leq 0 \\ & \mathbf{x}(t_0) - \bar{\mathbf{x}}_0 = 0 \end{aligned}$$

$\mathbf{f}(\mathbf{x}_k, \mathbf{u}_k)$ integrates the dynamics $\mathbf{F}$ over the time intervals $[t_k, t_{k+1}]$, provides the state $\mathbf{x}$ at t_{k+1}.





Algebraic variables in the cost & inequality constraints

OCP:

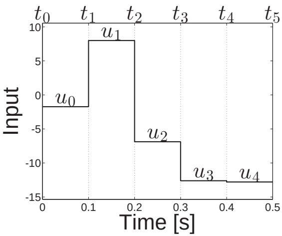
$$\begin{aligned} \min \quad & \Phi\left(\mathbf{x}(.), \mathbf{u}(.)\right) \\ \text{s.t.} \quad & \mathbf{F}\left(\dot{\mathbf{x}}(t), \mathbf{z}(t), \mathbf{x}(t), \mathbf{u}(t)\right) = 0 \\ & \mathbf{h}\left(\mathbf{x}(t), \mathbf{u}(t)\right) \leq 0 \\ & \mathbf{x}(t_0) = \bar{\mathbf{x}}_0 \end{aligned}$$

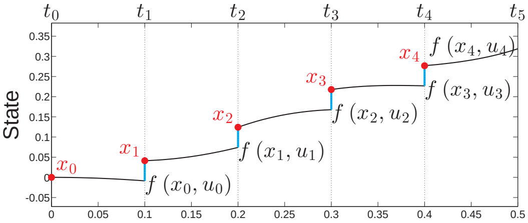
$\mathbf{f}(\mathbf{x}_k, \mathbf{u}_k)$ integrates the dynamics $\mathbf{F}$ over the time intervals $[t_k, t_{k+1}]$, provides the state $\mathbf{x}$ at t_{k+1}.

OCP:

$$\begin{aligned} \min \quad & \Phi\left(\mathbf{x}(.), \mathbf{z}(.), \mathbf{u}(.)\right) \\ \text{s.t.} \quad & \mathbf{F}\left(\dot{\mathbf{x}}(t), \mathbf{z}(t), \mathbf{x}(t), \mathbf{u}(t)\right) = 0 \\ & \mathbf{h}\left(\mathbf{x}(t), \mathbf{z}(t), \mathbf{u}(t)\right) \leq 0 \\ & \mathbf{x}(t_0) - \bar{\mathbf{x}}_0 = 0 \end{aligned}$$

Then the integrator needs to **report the algebraic variables** as well...



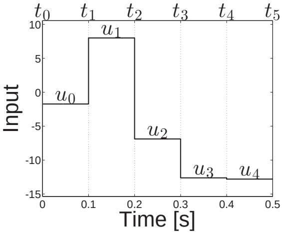


Algebraic variables in the cost & inequality constraints

OCP:

$$\begin{aligned} \min \quad & \Phi\left(\mathbf{x}(.), \mathbf{u}(.)\right) \\ \text{s.t.} \quad & \mathbf{F}\left(\dot{\mathbf{x}}\left(t\right), \mathbf{z}\left(t\right), \mathbf{x}\left(t\right), \mathbf{u}\left(t\right)\right) = 0 \\ & \mathbf{h}\left(\mathbf{x}(t), \mathbf{u}(t)\right) \leq 0 \\ & \mathbf{x}\left(t_0\right) = \bar{\mathbf{x}}_0 \end{aligned}$$

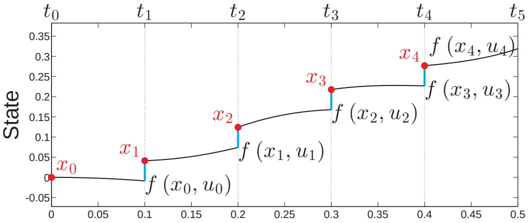
$\mathbf{f}\left(\mathbf{x}_k, \mathbf{u}_k\right)$ integrates the dynamics $\mathbf{F}$ over the time intervals $[t_k, \; t_{k+1}]$, provides the state $\mathbf{x}$ at t_{k+1}.



OCP:

$$\begin{aligned} \min \quad & \Phi\left(\mathbf{x}(.), \mathbf{z}(.), \mathbf{u}(.)\right) \\ \text{s.t.} \quad & \mathbf{F}\left(\dot{\mathbf{x}}\left(t\right), \mathbf{z}\left(t\right), \mathbf{x}\left(t\right), \mathbf{u}\left(t\right)\right) = 0 \\ & \mathbf{h}\left(\mathbf{x}(t), \mathbf{z}(t), \mathbf{u}(t)\right) \leq 0 \\ & \mathbf{x}\left(t_0\right) - \bar{\mathbf{x}}_0 = 0 \end{aligned}$$

Then the integrator needs to **report the algebraic variables** as well... but **where to impose the constraints** ? At the beginning or at the end of the shooting interval ? Ideally both...



Algebraic variables in the cost & inequality constraints

OCP:

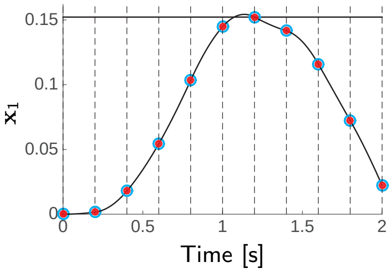
$$\begin{aligned} \min \quad & \Phi\left(\mathbf{x}(.), \mathbf{u}(.)\right) \\ \text{s.t.} \quad & \mathbf{F}\left(\dot{\mathbf{x}}\left(t\right), \mathbf{z}\left(t\right), \mathbf{x}\left(t\right), \mathbf{u}\left(t\right)\right) = 0 \\ & \mathbf{h}\left(\mathbf{x}(t), \mathbf{u}(t)\right) \leq 0 \\ & \mathbf{x}\left(t_0\right) = \bar{\mathbf{x}}_0 \end{aligned}$$

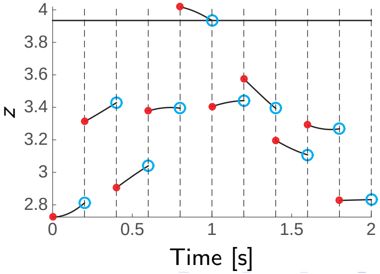
$\mathbf{f}\left(\mathbf{x}_k, \mathbf{u}_k\right)$ integrates the dynamics $\mathbf{F}$ over the time intervals $[t_k,\ t_{k+1}]$, provides the state $\mathbf{x}$ at t_{k+1}.

OCP:

$$\begin{aligned} \min \quad & \Phi\left(\mathbf{x}(.), \mathbf{z}(.), \mathbf{u}(.)\right) \\ \text{s.t.} \quad & \mathbf{F}\left(\dot{\mathbf{x}}\left(t\right), \mathbf{z}\left(t\right), \mathbf{x}\left(t\right), \mathbf{u}\left(t\right)\right) = 0 \\ & \mathbf{h}\left(\mathbf{x}(t), \mathbf{z}(t), \mathbf{u}(t)\right) \leq 0 \\ & \mathbf{x}\left(t_0\right) - \bar{\mathbf{x}}_0 = 0 \end{aligned}$$

Then the integrator needs to **report the algebraic variables** as well... but **where to impose the constraints** ? At the beginning or at the end of the shooting interval ? Ideally both...





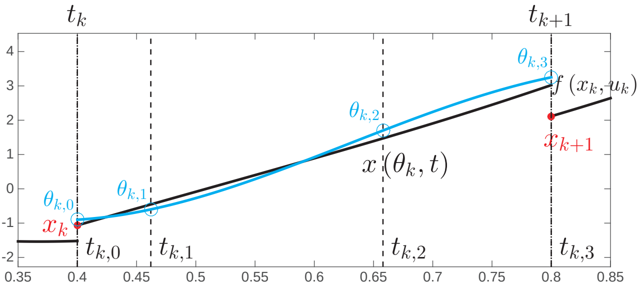
Outline

Direct Collocation - Reminder

On each interval $[t_k, t_{k+1}]$, approximate dynamics $\mathbf{F}\left(\dot{\mathbf{x}}, \mathbf{x}, \mathbf{u}\right) = 0$ using:

$$\mathbf{x}\left(\boldsymbol{\theta}_k, t\right) = \sum_{i=0}^{K} \underbrace{\boldsymbol{\theta}_{k,i}}_{\text{parameters}} \cdot \underbrace{P_{k,i}(t)}_{\text{polynomials}} \qquad \text{with} \qquad \mathbf{x}\left(\boldsymbol{\theta}_k, t_{k,i}\right) = \boldsymbol{\theta}_{k,i}$$

Note: $K + 1$ d.o.f. per state and per interval k.



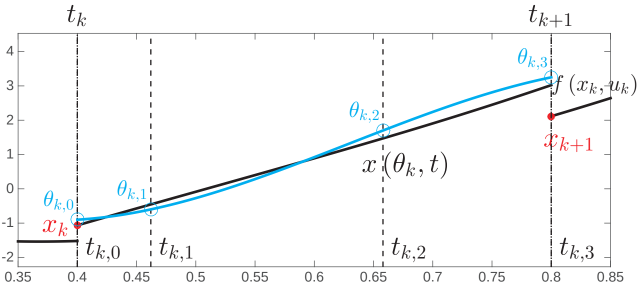
Direct Collocation - Reminder

On each interval $[t_k, t_{k+1}]$, approximate dynamics $\mathbf{F}(\dot{\mathbf{x}}, \mathbf{x}, \mathbf{u}) = 0$ using:

$$\mathbf{x}(\boldsymbol{\theta}_k, t) = \sum_{i=0}^{K} \underbrace{\boldsymbol{\theta}_{k,i}}_{\text{parameters}} \cdot \underbrace{P_{k,i}(t)}_{\text{polynomials}} \qquad \text{with} \qquad \mathbf{x}(\boldsymbol{\theta}_k, t_{k,i}) = \boldsymbol{\theta}_{k,i}$$

Note: $K+1$ d.o.f. per state and per interval k. Collocation uses the constraints:

Initial condition: $\mathbf{x}(\boldsymbol{\theta}_k, t_k) - \mathbf{x}_k = 0,$



Direct Collocation - Reminder

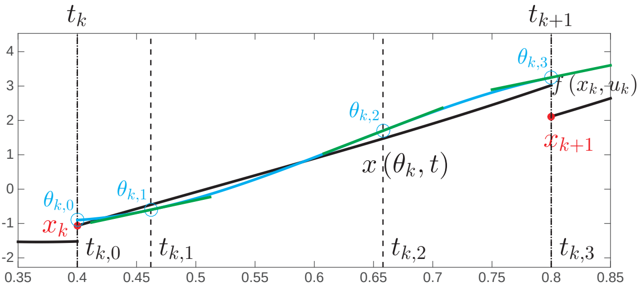
On each interval $[t_k, t_{k+1}]$, approximate dynamics $\mathbf{F}(\dot{\mathbf{x}}, \mathbf{x}, \mathbf{u}) = 0$ using:

$$\mathbf{x}(\boldsymbol{\theta}_k, t) = \sum_{i=0}^{K} \underbrace{\boldsymbol{\theta}_{k,i}}_{\text{parameters}} \cdot \underbrace{P_{k,i}(t)}_{\text{polynomials}} \qquad \text{with} \qquad \mathbf{x}(\boldsymbol{\theta}_k, t_{k,i}) = \boldsymbol{\theta}_{k,i}$$

Note: $K+1$ d.o.f. per state and per interval k. Collocation uses the constraints:

Initial condition: $\mathbf{x}(\boldsymbol{\theta}_k, t_k) - \mathbf{x}_k = 0,$

Dynamics: $\mathbf{F}\left(\frac{\partial}{\partial t}\mathbf{x}(\boldsymbol{\theta}_k, t_{k,i}), \mathbf{x}(\boldsymbol{\theta}_k, t_{k,i}), \mathbf{u}_k\right) = 0, \qquad i = 1, ..., K$



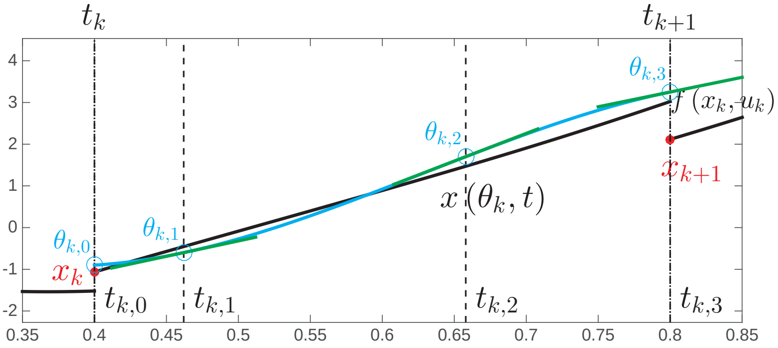
Direct Collocation- Reminder

Collocation uses the constraints:

$$\mathbf{x}\left(\boldsymbol{\theta}_k, t_k\right) = \mathbf{x}_k$$

$$\frac{\partial}{\partial t}\mathbf{x}\left(\boldsymbol{\theta}_k, t_{k,i}\right) = \mathbf{F}\left(\mathbf{x}\left(\boldsymbol{\theta}_k, t_{k,i}\right), \mathbf{u}_k\right),$$

with $i = 1, ..., K$.



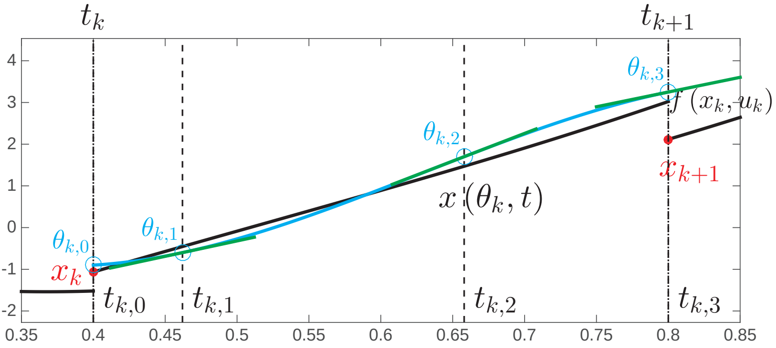
Direct Collocation- Reminder

Collocation uses the constraints:

$$\boldsymbol{\theta}_{k,0} = \mathbf{x}_k$$

$$\frac{\partial}{\partial t}\mathbf{x}\left(\boldsymbol{\theta}_k, t_{k,i}\right) = \mathbf{F}\left(\boldsymbol{\theta}_{k,i}, \mathbf{u}_k\right),$$

with $i = 1, ..., K$.



Direct Collocation- Reminder

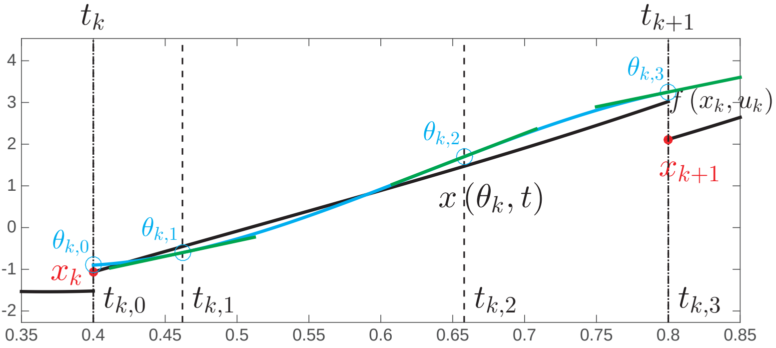
Collocation uses the constraints:

$$\boldsymbol{\theta}_{k,0} = \mathbf{x}_k$$

$$\frac{\partial}{\partial t}\mathbf{x}\left(\boldsymbol{\theta}_k, t_{k,i}\right) = \mathbf{F}\left(\boldsymbol{\theta}_{k,i}, \mathbf{u}_k\right),$$

with $i = 1, ..., K$. Note:

$$\frac{\partial}{\partial t}\mathbf{x}\left(\boldsymbol{\theta}_k, t\right) = \sum_{j=0}^{K} \boldsymbol{\theta}_{k,j}\dot{P}_{k,j}(t)$$



Direct Collocation- Reminder

Collocation uses the constraints:

$$\boldsymbol{\theta}_{k,0} = \mathbf{x}_k$$

$$\frac{\partial}{\partial t}\mathbf{x}\left(\boldsymbol{\theta}_k, t_{k,i}\right) = \mathbf{F}\left(\boldsymbol{\theta}_{k,i}, \mathbf{u}_k\right),$$

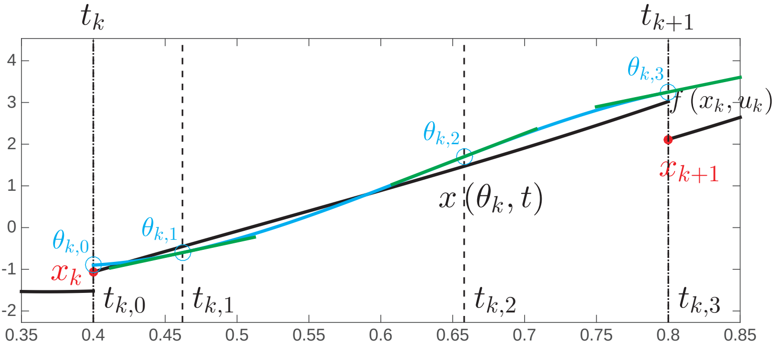
with $i = 1, ..., K$. Note:

$$\frac{\partial}{\partial t}\mathbf{x}\left(\boldsymbol{\theta}_k, t\right) = \sum_{j=0}^{K} \boldsymbol{\theta}_{k,j}\dot{P}_{k,j}(t)$$

Solve for $\boldsymbol{\theta}_{k,i}$ using Newton

$$\boldsymbol{\theta}_{k,0} = \mathbf{x}_k$$

$$\sum_{j=0}^{K} \boldsymbol{\theta}_{k,j}\dot{P}_{k,j}(t_{k,i}) = \mathbf{F}\left(\boldsymbol{\theta}_{k,i}, \mathbf{u}_k\right), \quad i = 1, ..., K$$



Direct Collocation- Reminder

Collocation uses the constraints:

$$\boldsymbol{\theta}_{k,0} = \mathbf{x}_k$$

$$\frac{\partial}{\partial t}\mathbf{x}\left(\boldsymbol{\theta}_k, t_{k,i}\right) = \mathbf{F}\left(\boldsymbol{\theta}_{k,i}, \mathbf{u}_k\right),$$

with $i = 1, ..., K$. Note:

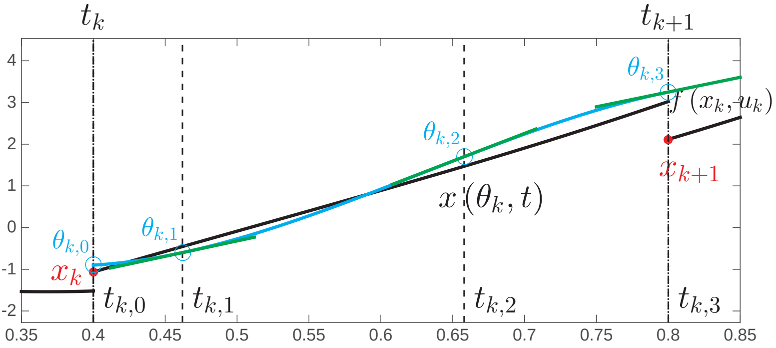
$$\frac{\partial}{\partial t}\mathbf{x}\left(\boldsymbol{\theta}_k, t\right) = \sum_{j=0}^{K} \boldsymbol{\theta}_{k,j} \dot{P}_{k,j}(t)$$

Shooting constraints

$$\underbrace{\mathbf{f}\left(\mathbf{x}_k, \mathbf{u}_k\right)}_{=\theta_{k,K}} - \underbrace{\mathbf{x}_{k+1}}_{=\theta_{k+1,0}} = 0$$

becomes:

$$\boldsymbol{\theta}_{k,K} - \boldsymbol{\theta}_{k+1,0} = 0$$



Direct Collocation - Reminder

On each interval $[t_k, t_{k+1}]$ with:

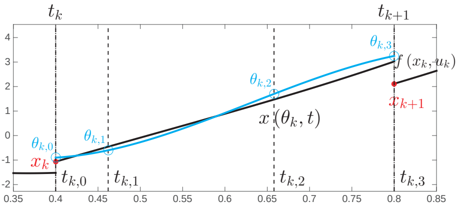
$$\dot{\mathbf{x}} = \mathbf{F}\left(\mathbf{x}, \mathbf{u}_k\right)$$

integration is approximated using:

$$\mathbf{x}\left(\boldsymbol{\theta}_k, t\right) = \sum_{i=0}^{K} \underbrace{\boldsymbol{\theta}_{k,i}}_{\text{parameters}} \cdot \underbrace{P_{k,i}(t)}_{\text{polynomials}}$$

Note:

- $\mathbf{x}\left(\boldsymbol{\theta}_{k,i}, t_{k,i}\right) = \boldsymbol{\theta}_{k,i}$
- $K+1$ degrees of freedom per state.



Direct Collocation - Reminder

On each interval $[t_k, t_{k+1}]$ with:

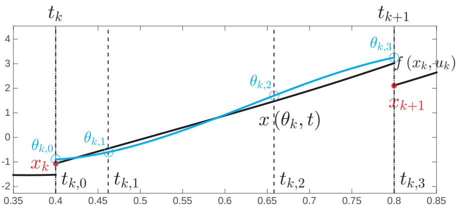
$$\dot{\mathbf{x}} = \mathbf{F}\left(\mathbf{x}, \mathbf{u}_k\right)$$

integration is approximated using:

$$\mathbf{x}\left(\boldsymbol{\theta}_k, t\right) = \sum_{i=0}^{K} \underbrace{\boldsymbol{\theta}_{k,i}}_{\text{parameters}} \cdot \underbrace{P_{k,i}(t)}_{\text{polynomials}}$$

Note:

- $\mathbf{x}\left(\boldsymbol{\theta}_{k,i}, t_{k,i}\right) = \boldsymbol{\theta}_{k,i}$
- $K+1$ degrees of freedom per state.



Integration constraints ($i = 1, ..., K$)

$$\frac{\partial}{\partial t}\mathbf{x}\left(\boldsymbol{\theta}_k, t_{k,i}\right) = \mathbf{F}\left(\mathbf{x}\left(\boldsymbol{\theta}_k, t_{k,i}\right), \mathbf{u}_k\right)$$

i.e.

$$\sum_{j=0}^{K} \boldsymbol{\theta}_{k,j} \dot{P}_{k,j}(t_{k,i}) = \mathbf{F}\left(\boldsymbol{\theta}_{k,i}, \mathbf{u}_k\right)$$

Direct Collocation - Reminder

On each interval $[t_k, t_{k+1}]$ with:

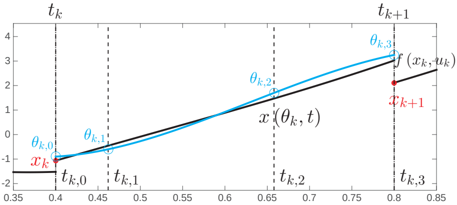
$$\dot{\mathbf{x}} = \mathbf{F}\left(\mathbf{x}, \mathbf{u}_k\right)$$

integration is approximated using:

$$\mathbf{x}\left(\boldsymbol{\theta}_k, t\right) = \sum_{i=0}^{K} \underbrace{\boldsymbol{\theta}_{k,i}}_{\text{parameters}} \cdot \underbrace{P_{k,i}(t)}_{\text{polynomials}}$$

Note:

- $\mathbf{x}\left(\boldsymbol{\theta}_{k,i}, t_{k,i}\right) = \boldsymbol{\theta}_{k,i}$
- $K+1$ degrees of freedom per state.



NLP with direct collocation

$$\min_{\mathbf{w}} \quad \Phi\left(\mathbf{w}\right)$$

$$\text{s.t.} \quad \mathbf{g}\left(\mathbf{w}\right) = \left[\quad \right]$$

Integration constraints $(i = 1, ..., K)$

$$\frac{\partial}{\partial t}\mathbf{x}\left(\boldsymbol{\theta}_k, t_{k,i}\right) = \mathbf{F}\left(\mathbf{x}\left(\boldsymbol{\theta}_k, t_{k,i}\right), \mathbf{u}_k\right)$$

i.e.

$$\sum_{j=0}^{K} \boldsymbol{\theta}_{k,j} \dot{P}_{k,j}(t_{k,i}) = \mathbf{F}\left(\boldsymbol{\theta}_{k,i}, \mathbf{u}_k\right)$$

Direct Collocation - Reminder

On each interval $[t_k, t_{k+1}]$ with:

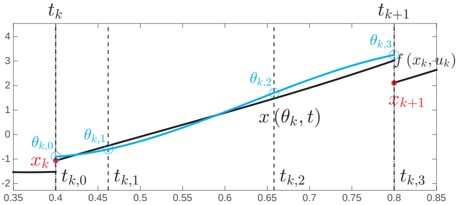
$$\dot{\mathbf{x}} = \mathbf{F}\left(\mathbf{x}, \mathbf{u}_k\right)$$

integration is approximated using:

$$\mathbf{x}\left(\boldsymbol{\theta}_k, t\right) = \sum_{i=0}^{K} \underbrace{\boldsymbol{\theta}_{k,i}}_{\text{parameters}} \cdot \underbrace{P_{k,i}(t)}_{\text{polynomials}}$$

Note:

- $\mathbf{x}\left(\boldsymbol{\theta}_{k,i}, t_{k,i}\right) = \boldsymbol{\theta}_{k,i}$
- $K+1$ degrees of freedom per state.



NLP with direct collocation

$$\min_{\mathbf{w}} \quad \Phi\left(\mathbf{w}\right)$$

$$\text{s.t.} \quad \mathbf{g}\left(\mathbf{w}\right) = \begin{bmatrix} \boldsymbol{\theta}_{0,0} - \bar{\mathbf{x}}_0 \\ \\ \\ \\ \end{bmatrix}$$

Initial conditions $\bar{\mathbf{x}}_0$

Integration constraints $(i = 1, \ldots, K)$

$$\frac{\partial}{\partial t}\mathbf{x}\left(\boldsymbol{\theta}_k, t_{k,i}\right) = \mathbf{F}\left(\mathbf{x}\left(\boldsymbol{\theta}_k, t_{k,i}\right), \mathbf{u}_k\right)$$

i.e.

$$\sum_{j=0}^{K} \boldsymbol{\theta}_{k,j} \dot{P}_{k,j}(t_{k,i}) = \mathbf{F}\left(\boldsymbol{\theta}_{k,i}, \mathbf{u}_k\right)$$

Direct Collocation - Reminder

On each interval $[t_k, t_{k+1}]$ with:

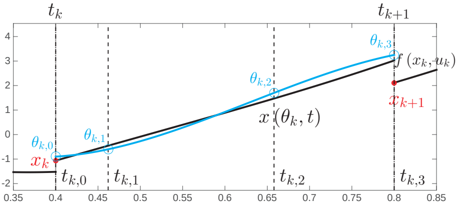
$$\dot{\mathbf{x}} = \mathbf{F}(\mathbf{x}, \mathbf{u}_k)$$

integration is approximated using:

$$\mathbf{x}(\boldsymbol{\theta}_k, t) = \sum_{i=0}^{K} \underbrace{\boldsymbol{\theta}_{k,i}}_{\text{parameters}} \cdot \underbrace{P_{k,i}(t)}_{\text{polynomials}}$$

Note:

- $\mathbf{x}(\boldsymbol{\theta}_{k,i}, t_{k,i}) = \boldsymbol{\theta}_{k,i}$
- $K+1$ degrees of freedom per state.



NLP with direct collocation

$$\min_{\mathbf{w}} \quad \Phi(\mathbf{w})$$

$$\text{s.t.} \quad \mathbf{g}(\mathbf{w}) = \begin{bmatrix} \boldsymbol{\theta}_{0,0} - \bar{\mathbf{x}}_0 \\ \boldsymbol{\theta}_{0,K} - \boldsymbol{\theta}_{1,0} \\ \\ \\ \\ \end{bmatrix}$$

Continuity constraints ($\equiv$ shooting gaps)

Integration constraints ($i = 1, \ldots, K$)

$$\frac{\partial}{\partial t}\mathbf{x}(\boldsymbol{\theta}_k, t_{k,i}) = \mathbf{F}(\mathbf{x}(\boldsymbol{\theta}_k, t_{k,i}), \mathbf{u}_k)$$

i.e.

$$\sum_{j=0}^{K} \boldsymbol{\theta}_{k,j} \dot{P}_{k,j}(t_{k,i}) = \mathbf{F}(\boldsymbol{\theta}_{k,i}, \mathbf{u}_k)$$

Direct Collocation - Reminder

On each interval $[t_k, t_{k+1}]$ with:

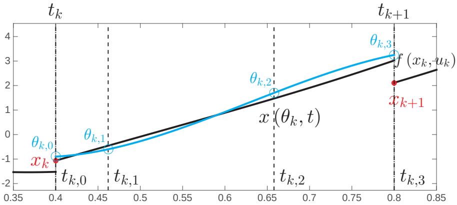
$$\dot{\mathbf{x}} = \mathbf{F}(\mathbf{x}, \mathbf{u}_k)$$

integration is approximated using:

$$\mathbf{x}(\boldsymbol{\theta}_k, t) = \sum_{i=0}^{K} \underbrace{\boldsymbol{\theta}_{k,i}}_{\text{parameters}} \cdot \underbrace{P_{k,i}(t)}_{\text{polynomials}}$$

Note:

- $\mathbf{x}(\boldsymbol{\theta}_{k,i}, t_{k,i}) = \boldsymbol{\theta}_{k,i}$
- $K+1$ degrees of freedom per state.



NLP with direct collocation

$$\min_{\mathbf{w}} \quad \Phi(\mathbf{w})$$

$$\text{s.t.} \quad \mathbf{g}(\mathbf{w}) = \begin{bmatrix} \boldsymbol{\theta}_{0,0} - \bar{\mathbf{x}}_0 \\ \boldsymbol{\theta}_{0,K} - \boldsymbol{\theta}_{1,0} \\ \mathbf{F}(\boldsymbol{\theta}_{0,i}, \mathbf{u}_0) - \sum_{j=0}^{K} \boldsymbol{\theta}_{0,j} \dot{P}_{0,j}(t_{0,i}) \\ \\ \\ \\ \end{bmatrix}$$

Integration constraints for $k = 0$

Integration constraints $(i = 1, \ldots, K)$

$$\frac{\partial}{\partial t}\mathbf{x}(\boldsymbol{\theta}_k, t_{k,i}) = \mathbf{F}(\mathbf{x}(\boldsymbol{\theta}_k, t_{k,i}), \mathbf{u}_k)$$

i.e.

$$\sum_{j=0}^{K} \boldsymbol{\theta}_{k,j} \dot{P}_{k,j}(t_{k,i}) = \mathbf{F}(\boldsymbol{\theta}_{k,i}, \mathbf{u}_k)$$

Direct Collocation - Reminder

On each interval $[t_k, t_{k+1}]$ with:

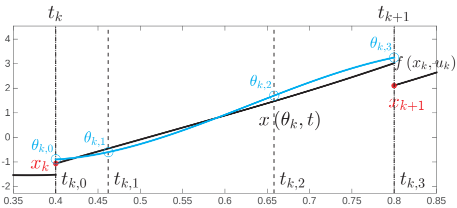
$$\dot{\mathbf{x}} = \mathbf{F}\left(\mathbf{x}, \mathbf{u}_k\right)$$

integration is approximated using:

$$\mathbf{x}\left(\boldsymbol{\theta}_k, t\right) = \sum_{i=0}^{K} \underbrace{\boldsymbol{\theta}_{k,i}}_{\text{parameters}} \cdot \underbrace{P_{k,i}(t)}_{\text{polynomials}}$$

Note:

- $\mathbf{x}\left(\boldsymbol{\theta}_{k,i}, t_{k,i}\right) = \boldsymbol{\theta}_{k,i}$
- $K+1$ degrees of freedom per state.



NLP with direct collocation

$$\min_{\mathbf{w}} \quad \Phi\left(\mathbf{w}\right)$$

$$\text{s.t.} \quad \mathbf{g}\left(\mathbf{w}\right) = \begin{bmatrix} \boldsymbol{\theta}_{0,0} - \bar{\mathbf{x}}_0 \\ \boldsymbol{\theta}_{0,K} - \boldsymbol{\theta}_{1,0} \\ \mathbf{F}\left(\boldsymbol{\theta}_{0,i}, \mathbf{u}_0\right) - \sum_{j=0}^{K} \boldsymbol{\theta}_{0,j} \dot{P}_{0,j}(t_{0,i}) \\ \dots \\ \boldsymbol{\theta}_{k,K} - \boldsymbol{\theta}_{k+1,0} \\ \mathbf{F}\left(\boldsymbol{\theta}_{k,i}, \mathbf{u}_k\right) - \sum_{j=0}^{K} \boldsymbol{\theta}_{k,j} \dot{P}_{k,j}(t_{k,i}) \\ \dots \end{bmatrix}$$

Remaining integration constraints $k = 1, ..., N-1$

Integration constraints $(i = 1, ..., K)$

$$\frac{\partial}{\partial t}\mathbf{x}\left(\boldsymbol{\theta}_k, t_{k,i}\right) = \mathbf{F}\left(\mathbf{x}\left(\boldsymbol{\theta}_k, t_{k,i}\right), \mathbf{u}_k\right)$$

i.e.

$$\sum_{j=0}^{K} \boldsymbol{\theta}_{k,j} \dot{P}_{k,j}(t_{k,i}) = \mathbf{F}\left(\boldsymbol{\theta}_{k,i}, \mathbf{u}_k\right)$$

Direct Collocation - Reminder

On each interval $[t_k, t_{k+1}]$ with:

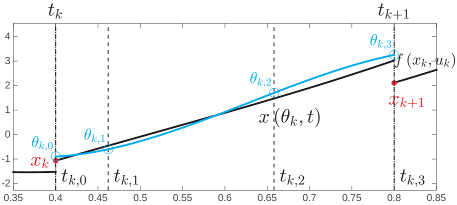
$$\dot{\mathbf{x}} = \mathbf{F}(\mathbf{x}, \mathbf{u}_k)$$

integration is approximated using:

$$\mathbf{x}(\boldsymbol{\theta}_k, t) = \sum_{i=0}^{K} \underbrace{\boldsymbol{\theta}_{k,i}}_{\text{parameters}} \cdot \underbrace{P_{k,i}(t)}_{\text{polynomials}}$$

Note:

- $\mathbf{x}(\boldsymbol{\theta}_{k,i}, t_{k,i}) = \boldsymbol{\theta}_{k,i}$
- $K+1$ degrees of freedom per state.



NLP with direct collocation

$$\min_{\mathbf{w}} \quad \Phi(\mathbf{w})$$

$$\text{s.t.} \quad \mathbf{g}(\mathbf{w}) = \begin{bmatrix} \boldsymbol{\theta}_{0,0} - \bar{\mathbf{x}}_0 \\ \boldsymbol{\theta}_{0,K} - \boldsymbol{\theta}_{1,0} \\ \mathbf{F}(\boldsymbol{\theta}_{0,i}, \mathbf{u}_0) - \sum_{j=0}^{K} \boldsymbol{\theta}_{0,j} \dot{P}_{0,j}(t_{0,i}) \\ \dots \\ \boldsymbol{\theta}_{k,K} - \boldsymbol{\theta}_{k+1,0} \\ \mathbf{F}(\boldsymbol{\theta}_{k,i}, \mathbf{u}_k) - \sum_{j=0}^{K} \boldsymbol{\theta}_{k,j} \dot{P}_{k,j}(t_{k,i}) \\ \dots \end{bmatrix}$$

Decision variables:

$$\mathbf{w} = \{\boldsymbol{\theta}_{0,1}, \dots, \boldsymbol{\theta}_{0,K}, \mathbf{u}_0, \dots, \boldsymbol{\theta}_{N-1,1}, \dots, \boldsymbol{\theta}_{N-1,K}, \mathbf{u}_{N-1}\}$$

Integration constraints $(i = 1, \dots, K)$

$$\frac{\partial}{\partial t} \mathbf{x}(\boldsymbol{\theta}_k, t_{k,i}) = \mathbf{F}(\mathbf{x}(\boldsymbol{\theta}_k, t_{k,i}), \mathbf{u}_k)$$

i.e.

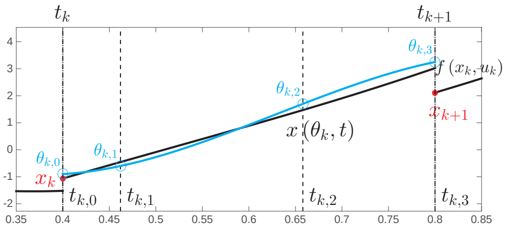
$$\sum_{j=0}^{K} \boldsymbol{\theta}_{k,j} \dot{P}_{k,j}(t_{k,i}) = \mathbf{F}(\boldsymbol{\theta}_{k,i}, \mathbf{u}_k)$$

Outline

Direct Collocation for DAE-constrained problems

On each interval $[t_k, t_{k+1}]$ with:

$$\mathbf{F}\left(\dot{\mathbf{x}}, \mathbf{x}, \mathbf{z}, \mathbf{u}_k\right) = 0$$



Direct Collocation for DAE-constrained problems

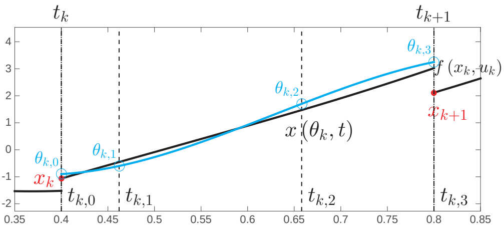
On each interval $[t_k, t_{k+1}]$ with:

$$\mathbf{F}\left(\dot{\mathbf{x}}, \mathbf{x}, \mathbf{z}, \mathbf{u}_k\right) = 0$$

Integration is approximated using:

$$\mathbf{x}\left(\boldsymbol{\theta}_k, t\right) = \sum_{i=0}^{K} \overbrace{\boldsymbol{\theta}_{k,i}}^{\text{parameters}} \cdot \overbrace{P_{k,i}(t)}^{\text{polynomials}}$$

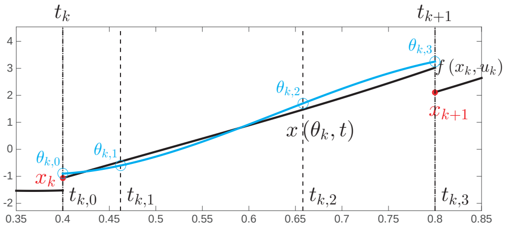
$$\mathbf{z}\left(\mathbf{z}_k, t\right) = \sum_{i=1}^{K} \underbrace{\mathbf{z}_{k,i}}_{\text{parameters}} \cdot \underbrace{P_{k,i}(t)}_{\text{polynomials}}$$



Direct Collocation for DAE-constrained problems

On each interval $[t_k, t_{k+1}]$ with:

$$\mathbf{F}\left(\dot{\mathbf{x}}, \mathbf{x}, \mathbf{z}, \mathbf{u}_k\right) = 0$$



Integration is approximated using:

$$\mathbf{x}\left(\boldsymbol{\theta}_k, t\right) = \sum_{i=0}^{K} \overbrace{\boldsymbol{\theta}_{k,i}}^{\text{parameters}} \cdot \overbrace{P_{k,i}(t)}^{\text{polynomials}}$$

$$\mathbf{z}\left(\mathbf{z}_k, t\right) = \sum_{i=1}^{K} \underbrace{\mathbf{z}_{k,i}}_{\text{parameters}} \cdot \underbrace{P_{k,i}(t)}_{\text{polynomials}}$$

Note:

- $\mathbf{x}\left(\boldsymbol{\theta}_k, t_{k,i}\right) = \boldsymbol{\theta}_{k,i}$
- $\mathbf{z}\left(\mathbf{z}_k, t_{k,i}\right) = \mathbf{z}_{k,i}$
- $K + 1$ d.o.f. per differential state
- K d.o.f. per algebraic state

Direct Collocation for DAE-constrained problems

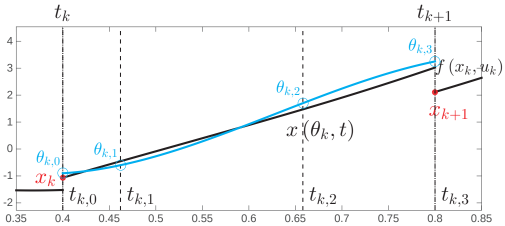
On each interval $[t_k, t_{k+1}]$ with:

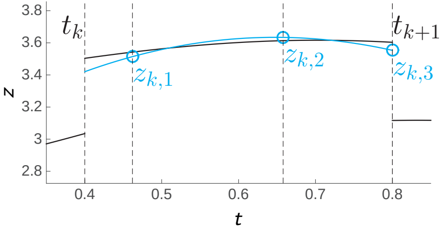
$$\mathbf{F}\left(\dot{\mathbf{x}}, \mathbf{x}, \mathbf{z}, \mathbf{u}_k\right) = 0$$

Integration is approximated using:

$$\mathbf{x}\left(\boldsymbol{\theta}_k, t\right) = \sum_{i=0}^{K} \overbrace{\boldsymbol{\theta}_{k,i}}^{\text{parameters}} \cdot \overbrace{P_{k,i}(t)}^{\text{polynomials}}$$

$$\mathbf{z}\left(\mathbf{z}_k, t\right) = \sum_{i=1}^{K} \underbrace{\mathbf{z}_{k,i}}_{\text{parameters}} \cdot \underbrace{P_{k,i}(t)}_{\text{polynomials}}$$





Note:

- $\mathbf{x}\left(\boldsymbol{\theta}_k, t_{k,i}\right) = \boldsymbol{\theta}_{k,i}$
- $\mathbf{z}\left(\mathbf{z}_k, t_{k,i}\right) = \mathbf{z}_{k,i}$
- $K + 1$ d.o.f. per differential state
- K d.o.f. per algebraic state

Direct Collocation for DAE-constrained problems

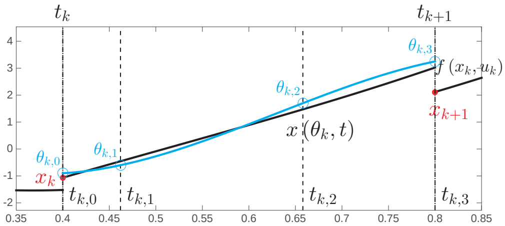
On each interval $[t_k, t_{k+1}]$ with:

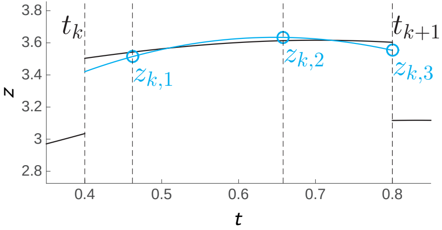
$$\mathbf{F}\left(\dot{\mathbf{x}}, \mathbf{x}, \mathbf{z}, \mathbf{u}_k\right) = 0$$

Integration is approximated using:

$$\mathbf{x}\left(\boldsymbol{\theta}_k, t\right) = \sum_{i=0}^{K} \overbrace{\boldsymbol{\theta}_{k,i}}^{\text{parameters}} \cdot \overbrace{P_{k,i}(t)}^{\text{polynomials}}$$

$$\mathbf{z}\left(\mathbf{z}_k, t\right) = \sum_{i=1}^{K} \underbrace{\mathbf{z}_{k,i}}_{\text{parameters}} \cdot \underbrace{P_{k,i}(t)}_{\text{polynomials}}$$





Note:

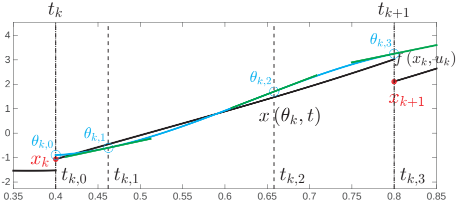
- $\mathbf{x}\left(\boldsymbol{\theta}_k, t_{k,i}\right) = \boldsymbol{\theta}_{k,i}$
- $\mathbf{z}\left(\mathbf{z}_k, t_{k,i}\right) = \mathbf{z}_{k,i}$
- $K + 1$ d.o.f. per differential state
- K d.o.f. per algebraic state

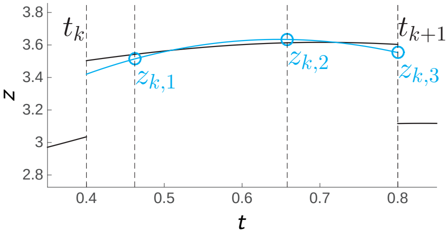
Why different d.o.f ? The differential states need an extra degree of freedom (hence $K+1$) for continuity (i.e. to close the shooting gaps). Algebraic states can be discontinuous and therefore need only K degrees of freedom !

Direct Collocation for DAE-constrained problems

Fully implicit DAE:

$$\mathbf{F}\left(\dot{\mathbf{x}}, \mathbf{x}, \mathbf{z}, \mathbf{u}_k\right) = 0$$





Direct Collocation for DAE-constrained problems

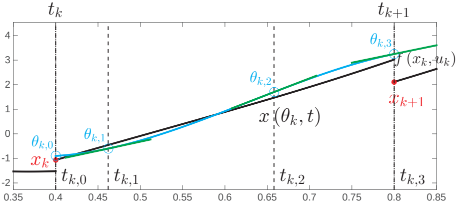
Fully implicit DAE:

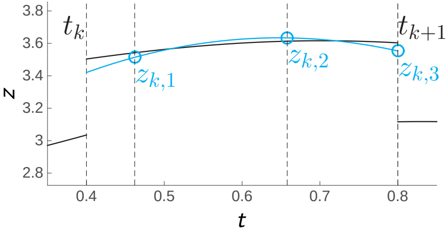
$$\mathbf{F}\left(\dot{\mathbf{x}}, \mathbf{x}, \mathbf{z}, \mathbf{u}_k\right) = 0$$

Interpolation:

$$\mathbf{x}\left(\boldsymbol{\theta}_k, t\right) = \sum_{i=0}^{K} \boldsymbol{\theta}_{k,i} P_{k,i}(t)$$

$$\mathbf{z}\left(\mathbf{z}_k, t\right) = \sum_{i=1}^{K} \mathbf{z}_{k,i} P_{k,i}(t)$$





Direct Collocation for DAE-constrained problems

Fully implicit DAE:

$$\mathbf{F}\left(\dot{\mathbf{x}}, \mathbf{x}, \mathbf{z}, \mathbf{u}_k\right) = 0$$

Interpolation:

$$\mathbf{x}\left(\boldsymbol{\theta}_k, t\right) = \sum_{i=0}^{K} \boldsymbol{\theta}_{k,i} P_{k,i}(t)$$

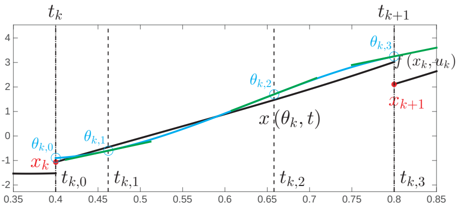
$$\mathbf{z}\left(\mathbf{z}_k, t\right) = \sum_{i=1}^{K} \mathbf{z}_{k,i} P_{k,i}(t)$$

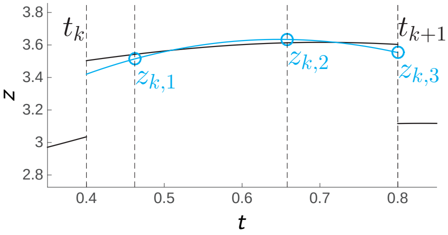
Collocation uses the constraints:

$$\mathbf{x}\left(\boldsymbol{\theta}_k, t_k\right) - \mathbf{x}\left(\boldsymbol{\theta}_{k+1}, t_k\right) = 0 \quad \textbf{continuity}$$

$$\mathbf{F}\left(\frac{\partial}{\partial t}\mathbf{x}\left(\boldsymbol{\theta}_k, t_{k,i}\right), \mathbf{x}\left(\boldsymbol{\theta}_k, t_{k,i}\right), \mathbf{z}_{k,i}, \mathbf{u}_k\right) = 0 \quad \textbf{dynamics}$$

with $k = 0, \ldots, N-1$, and $i = 1, \ldots, K$.





Direct Collocation for DAE-constrained problems

Fully implicit DAE:

$$\mathbf{F}\left(\dot{\mathbf{x}}, \mathbf{x}, \mathbf{z}, \mathbf{u}_k\right) = 0$$

Interpolation:

$$\mathbf{x}\left(\boldsymbol{\theta}_k, t\right) = \sum_{i=0}^{K} \boldsymbol{\theta}_{k,i} P_{k,i}(t)$$

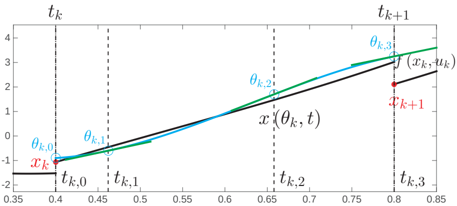
$$\mathbf{z}\left(\mathbf{z}_k, t\right) = \sum_{i=1}^{K} \mathbf{z}_{k,i} P_{k,i}(t)$$

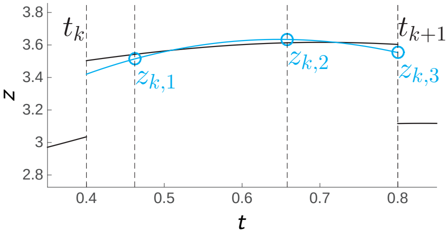
Collocation uses the constraints:

$$\boldsymbol{\theta}_{k,K} - \boldsymbol{\theta}_{k+1,0} = 0 \quad \textbf{continuity}$$

$$\mathbf{F}\left(\sum_{j=0}^{K} \boldsymbol{\theta}_{k,j} \dot{P}_{k,j}(t_{k,i}), \boldsymbol{\theta}_{k,i}, \mathbf{z}_{k,i}, \mathbf{u}_k\right) = 0 \quad \textbf{dynamics}$$

with $k = 0, ..., N-1$, and $i = 1, ..., K$.





Direct Collocation for DAE-constrained problems

Fully implicit DAE:

$$\mathbf{F}\left(\dot{\mathbf{x}}, \mathbf{x}, \mathbf{z}, \mathbf{u}_k\right) = 0$$

Interpolation:

$$\mathbf{x}\left(\boldsymbol{\theta}_k, t\right) = \sum_{i=0}^{K} \boldsymbol{\theta}_{k,i} P_{k,i}(t)$$

$$\mathbf{z}\left(\mathbf{z}_k, t\right) = \sum_{i=1}^{K} \mathbf{z}_{k,i} P_{k,i}(t)$$

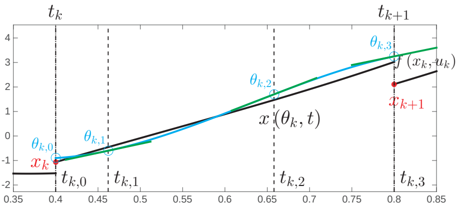
Collocation uses the constraints:

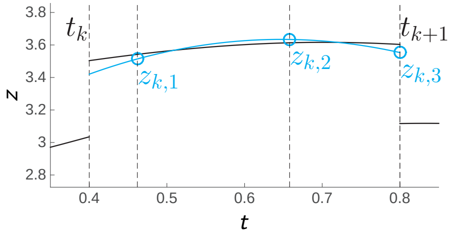
$$\boldsymbol{\theta}_{k,K} - \boldsymbol{\theta}_{k+1,0} = 0 \quad \textbf{continuity}$$

$$\mathbf{F}\left(\sum_{j=0}^{K} \boldsymbol{\theta}_{k,j} \dot{P}_{k,j}(t_{k,i}), \boldsymbol{\theta}_{k,i}, \mathbf{z}_{k,i}, \mathbf{u}_k\right) = 0 \quad \textbf{dynamics}$$

with $k = 0, ..., N-1$, and $i = 1, ..., K$.

Note: algebraic states appear only in the **dynamics** ($i = 1, ..., K$ hence K equations !!), hence only K are needed.

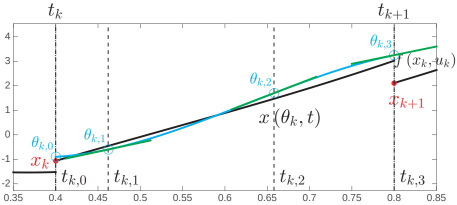


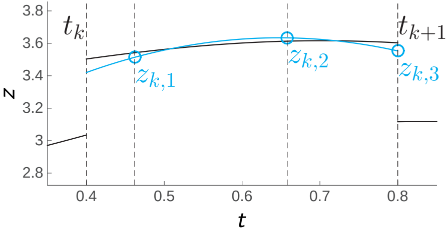


Direct Collocation for DAE-constrained problems

Semi-explicit DAE

$$\begin{aligned}\dot{\mathbf{x}} &= \mathbf{F}\left(\mathbf{x}, \mathbf{z}, \mathbf{u}_k\right)\\ 0 &= \mathbf{G}\left(\mathbf{x}, \mathbf{z}, \mathbf{u}_k\right)\end{aligned}$$





Direct Collocation for DAE-constrained problems

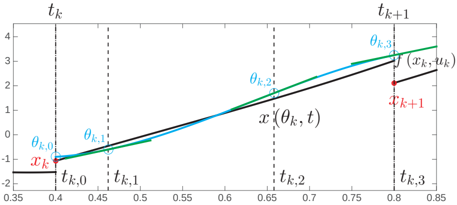
Semi-explicit DAE

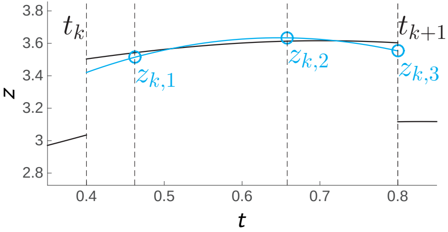
$$\dot{\mathbf{x}} = \mathbf{F}\left(\mathbf{x}, \mathbf{z}, \mathbf{u}_k\right)$$
$$0 = \mathbf{G}\left(\mathbf{x}, \mathbf{z}, \mathbf{u}_k\right)$$

Collocation uses the constraints:

$$0 = \mathbf{x}\left(\boldsymbol{\theta}_k, t_k\right) - \mathbf{x}\left(\boldsymbol{\theta}_{k+1}, t_k\right) \quad \textbf{continuity}$$
$$\frac{\partial}{\partial t}\mathbf{x}\left(\boldsymbol{\theta}_k, t_{k,i}\right) = \mathbf{F}\left(\mathbf{x}\left(\boldsymbol{\theta}_k, t_{k,i}\right), \mathbf{z}_{k,i}, \mathbf{u}_k\right) \quad \textbf{dynamics}$$
$$0 = \mathbf{G}\left(\mathbf{x}\left(\boldsymbol{\theta}_k, t_{k,i}\right), \mathbf{z}_{k,i}, \mathbf{u}_k\right) \quad \textbf{algebraic}$$

with $k = 0, ..., N-1$, and $i = 1, ..., K$.





Direct Collocation for DAE-constrained problems

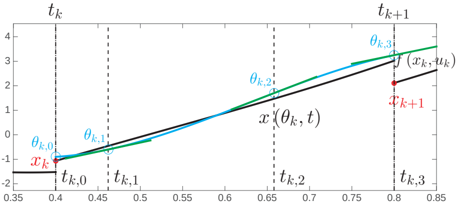
Semi-explicit DAE

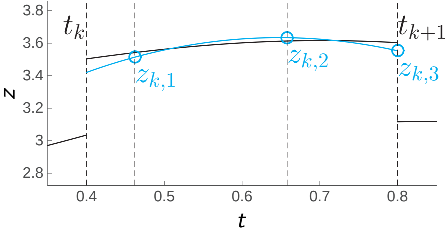
$$\dot{\mathbf{x}} = \mathbf{F}\left(\mathbf{x}, \mathbf{z}, \mathbf{u}_k\right)$$
$$0 = \mathbf{G}\left(\mathbf{x}, \mathbf{z}, \mathbf{u}_k\right)$$

Collocation uses the constraints:

$$0 = \boldsymbol{\theta}_{k,K} - \boldsymbol{\theta}_{k+1,0} \quad \textbf{continuity}$$
$$\frac{\partial}{\partial t}\mathbf{x}\left(\boldsymbol{\theta}_k, t_{k,i}\right) = \mathbf{F}\left(\boldsymbol{\theta}_{k,i}, \mathbf{z}_{k,i}, \mathbf{u}_k\right) \quad \textbf{dynamics}$$
$$0 = \mathbf{G}\left(\boldsymbol{\theta}_{k,i}, \mathbf{z}_{k,i}, \mathbf{u}_k\right) \quad \textbf{algebraic}$$

with $k = 0, ..., N-1$, and $i = 1, ..., K$.





Selection of the collocation points for DAEs

What collocation scheme to use for DAEs ?!?

Selection of the collocation points for DAEs

What collocation scheme to use for DAEs ?!?

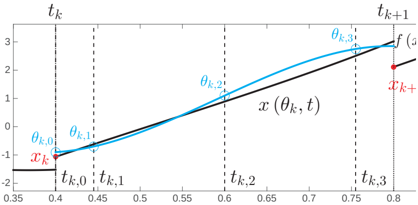
K	Legendre	Radau
1	0.5	1.0
2	0.211325	0.333333
	0.788675	1.000000
3	0.112702	0.155051
	0.500000	0.644949
	0.887298	1.000000
4	0.069432	0.088588
	0.330009	0.409467
	0.669991	0.787659
	0.930568	1.000000
5	0.046910	0.057104
	0.230765	0.276843
	0.500000	0.583590
	0.769235	0.860240
	0.953090	1.000000

c.f. Lecture "Direct Collocation"

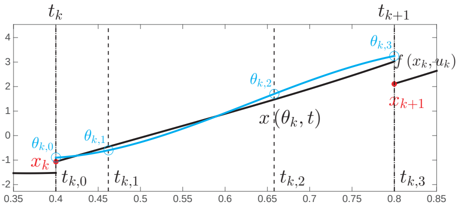
Selection of the collocation points for DAEs

What collocation scheme to use for DAEs ?!?

E.g. Legendre, $K = 3$



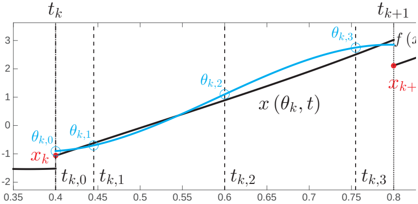
E.g. Radau, $K = 3$



Selection of the collocation points for DAEs

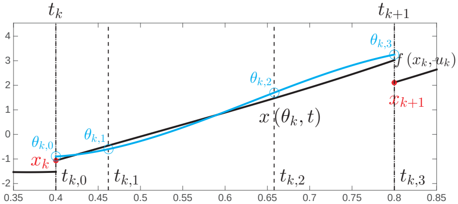
What collocation scheme to use for DAEs ?!?

E.g. Legendre, $K = 3$



- has a collocation point at t_k all others inside $[t_k,\ t_{k+1}]$

E.g. Radau, $K = 3$

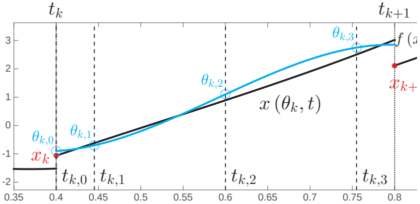


- has collocation points at t_k and t_{k+1}

Selection of the collocation points for DAEs

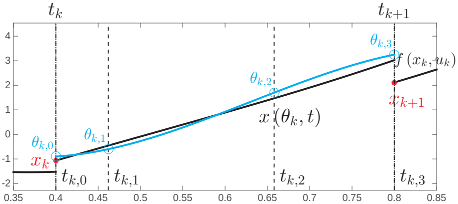
What collocation scheme to use for DAEs ?!?

E.g. Legendre, $K = 3$



- has a collocation point at t_k all others inside $[t_k,\ t_{k+1}]$
- integration order $2K = 6$

E.g. Radau, $K = 3$

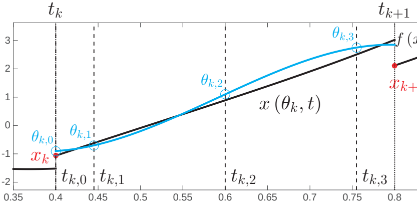


- has collocation points at t_k and t_{k+1}
- integration order $2K - 1 = 5$

Selection of the collocation points for DAEs

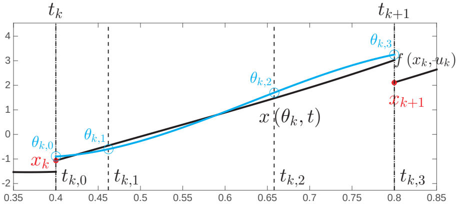
What collocation scheme to use for DAEs ?!?

E.g. Legendre, $K = 3$



- has a collocation point at t_k all others inside $[t_k,\ t_{k+1}]$
- integration order $2K = 6$
- has **A-stability** (stable for eigenvalues $\rightarrow -\infty$)

E.g. Radau, $K = 3$

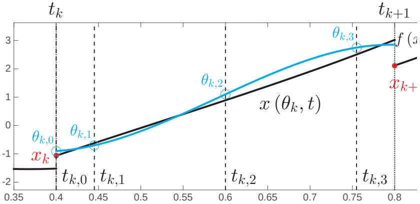


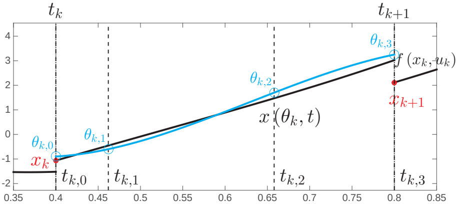
- has collocation points at t_k and t_{k+1}
- integration order $2K - 1 = 5$
- has **L-stability** (stable for eigenvalues at $-\infty$)

Selection of the collocation points for DAEs

What collocation scheme to use for DAEs ?!?

E.g. Legendre, $K = 3$	**E.g. Radau, $K = 3$**





Legendre:

- has a collocation point at t_k all others inside $[t_k,\ t_{k+1}]$
- integration order $2K = 6$
- has **A-stability** (stable for eigenvalues $\to -\infty$)
- best **suited for stiff ODEs**

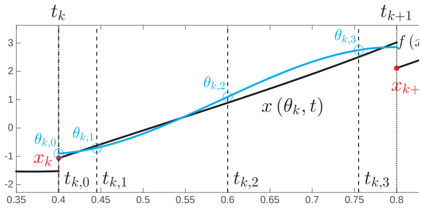
Radau:

- has collocation points at t_k and t_{k+1}
- integration order $2K - 1 = 5$
- has **L-stability** (stable for eigenvalues at $-\infty$)
- best **suited for DAEs**

Selection of the collocation points for DAEs

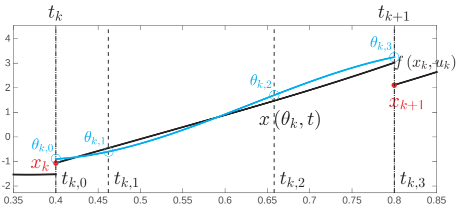
What collocation scheme to use for DAEs ?!?

E.g. Legendre, $K = 3$



- has a collocation point at t_k all others inside $[t_k,\ t_{k+1}]$
- integration order $2K = 6$
- has **A-stability** (stable for eigenvalues $\rightarrow -\infty$)
- best **suited for stiff ODEs**

E.g. Radau, $K = 3$



- has collocation points at t_k and t_{k+1}
- integration order $2K - 1 = 5$
- has **L-stability** (stable for eigenvalues at $-\infty$)
- best **suited for DAEs**

Careful: using a very high order collocation setup *can* deteriorate the conditioning of your KKT matrices and hinder the linear algebra underlying the NLP solver !!

NLP from Direct Collocation for DAE-constrained OCPs

Fully implicit DAE:

$$\mathbf{F}\left(\dot{\mathbf{x}}, \mathbf{x}, \mathbf{z}, \mathbf{u}_k\right) = 0$$

NLP from Direct Collocation for DAE-constrained OCPs

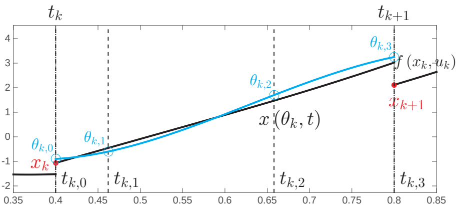
Fully implicit DAE:

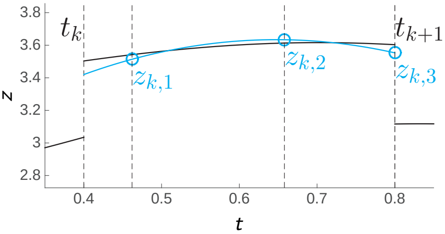
$$\mathbf{F}\left(\dot{\mathbf{x}}, \mathbf{x}, \mathbf{z}, \mathbf{u}_k\right) = 0$$

Interpolation:

$$\mathbf{x}\left(\boldsymbol{\theta}_k, t\right) = \sum_{i=0}^{K} \boldsymbol{\theta}_{k,i} P_{k,i}(t)$$

$$\mathbf{z}\left(\mathbf{z}_k, t\right) = \sum_{i=1}^{K} \mathbf{z}_{k,i} P_{k,i}(t)$$





NLP from Direct Collocation for DAE-constrained OCPs

Fully implicit DAE:

$$\mathbf{F}\left(\dot{\mathbf{x}}, \mathbf{x}, \mathbf{z}, \mathbf{u}_k\right) = 0$$

Interpolation:

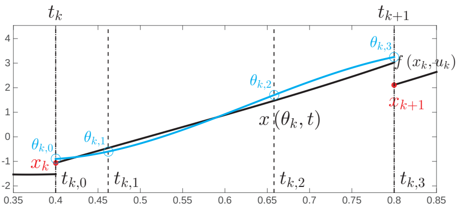
$$\mathbf{x}\left(\boldsymbol{\theta}_k, t\right) = \sum_{i=0}^{K} \boldsymbol{\theta}_{k,i} P_{k,i}(t)$$

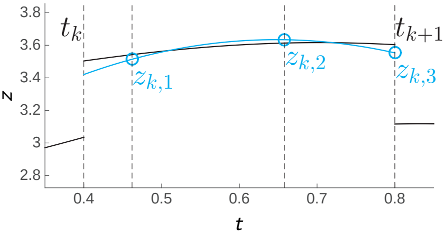
$$\mathbf{z}\left(\mathbf{z}_k, t\right) = \sum_{i=1}^{K} \mathbf{z}_{k,i} P_{k,i}(t)$$

NLP with direct collocation

$$\min_{\mathbf{w}} \quad \Phi\left(\mathbf{w}\right)$$

$$\text{s.t.} \quad \mathbf{g}\left(\mathbf{w}\right) = \left[\quad \right]$$





NLP from Direct Collocation for DAE-constrained OCPs

Fully implicit DAE:

$$\mathbf{F}\left(\dot{\mathbf{x}}, \mathbf{x}, \mathbf{z}, \mathbf{u}_k\right) = 0$$

Interpolation:

$$\mathbf{x}\left(\boldsymbol{\theta}_k, t\right) = \sum_{i=0}^{K} \boldsymbol{\theta}_{k,i} P_{k,i}(t)$$

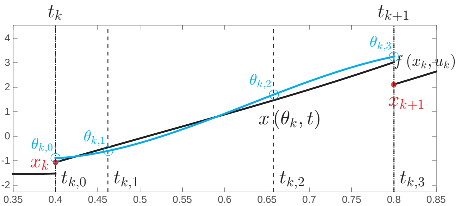
$$\mathbf{z}\left(\mathbf{z}_k, t\right) = \sum_{i=1}^{K} \mathbf{z}_{k,i} P_{k,i}(t)$$

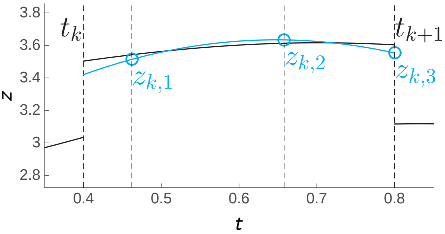
NLP with direct collocation

$$\min_{\mathbf{w}} \quad \Phi\left(\mathbf{w}\right)$$

$$\text{s.t.} \quad \mathbf{g}\left(\mathbf{w}\right) = \begin{bmatrix} \boldsymbol{\theta}_{0,0} - \bar{\mathbf{x}}_0 \\ \\ \\ \\ \\ \end{bmatrix}$$

Initial conditions $\bar{\mathbf{x}}_0$





NLP from Direct Collocation for DAE-constrained OCPs

Fully implicit DAE:

$$\mathbf{F}\left(\dot{\mathbf{x}}, \mathbf{x}, \mathbf{z}, \mathbf{u}_k\right) = 0$$

Interpolation:

$$\mathbf{x}\left(\boldsymbol{\theta}_k, t\right) = \sum_{i=0}^{K} \boldsymbol{\theta}_{k,i} P_{k,i}(t)$$

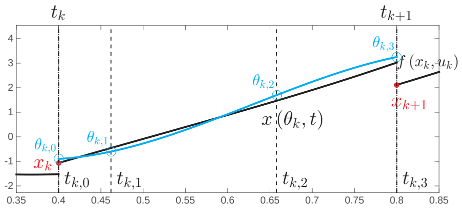
$$\mathbf{z}\left(\mathbf{z}_k, t\right) = \sum_{i=1}^{K} \mathbf{z}_{k,i} P_{k,i}(t)$$

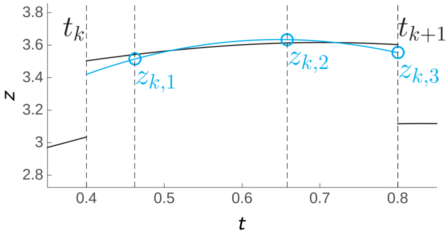
NLP with direct collocation

$$\min_{\mathbf{w}} \quad \Phi\left(\mathbf{w}\right)$$

$$\text{s.t.} \quad \mathbf{g}\left(\mathbf{w}\right) = \begin{bmatrix} \boldsymbol{\theta}_{0,0} - \bar{\mathbf{x}}_0 \\ \boldsymbol{\theta}_{0,K} - \boldsymbol{\theta}_{1,0} \\ \\ \\ \\ \end{bmatrix}$$

Continuity constraints ($\equiv$ shooting gaps)





NLP from Direct Collocation for DAE-constrained OCPs

Fully implicit DAE:

$$\mathbf{F}\left(\dot{\mathbf{x}}, \mathbf{x}, \mathbf{z}, \mathbf{u}_k\right) = 0$$

Interpolation:

$$\mathbf{x}\left(\boldsymbol{\theta}_k, t\right) = \sum_{i=0}^{K} \boldsymbol{\theta}_{k,i} P_{k,i}(t)$$

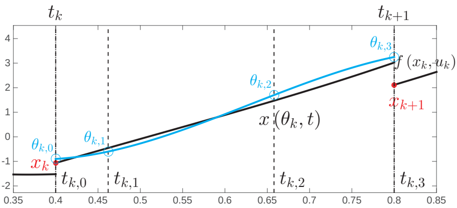
$$\mathbf{z}\left(\mathbf{z}_k, t\right) = \sum_{i=1}^{K} \mathbf{z}_{k,i} P_{k,i}(t)$$

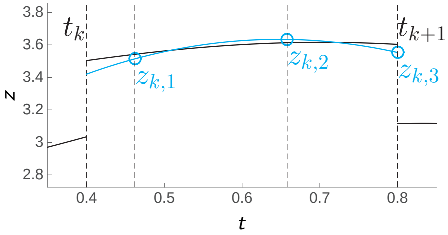
NLP with direct collocation

$$\min_{\mathbf{w}} \quad \Phi\left(\mathbf{w}\right)$$

$$\text{s.t.} \quad \mathbf{g}\left(\mathbf{w}\right) = \begin{bmatrix} \boldsymbol{\theta}_{0,0} - \bar{\mathbf{x}}_0 \\ \boldsymbol{\theta}_{0,K} - \boldsymbol{\theta}_{1,0} \\ \mathbf{F}\left(\frac{\partial}{\partial t}\mathbf{x}\left(\boldsymbol{\theta}_k, t_{k,0}\right), \boldsymbol{\theta}_{k,0}, \mathbf{z}_{k,0}, \mathbf{u}_k\right) \\ \\ \\ \end{bmatrix}$$

Integration constraints for $k = 0$





NLP from Direct Collocation for DAE-constrained OCPs

Fully implicit DAE:

$$\mathbf{F}\left(\dot{\mathbf{x}}, \mathbf{x}, \mathbf{z}, \mathbf{u}_k\right) = 0$$

Interpolation:

$$\mathbf{x}\left(\boldsymbol{\theta}_k, t\right) = \sum_{i=0}^{K} \boldsymbol{\theta}_{k,i} P_{k,i}(t)$$

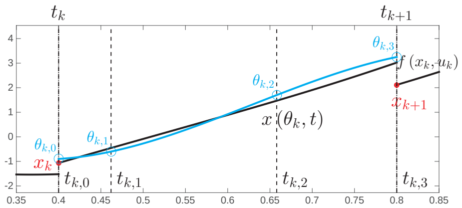
$$\mathbf{z}\left(\mathbf{z}_k, t\right) = \sum_{i=1}^{K} \mathbf{z}_{k,i} P_{k,i}(t)$$

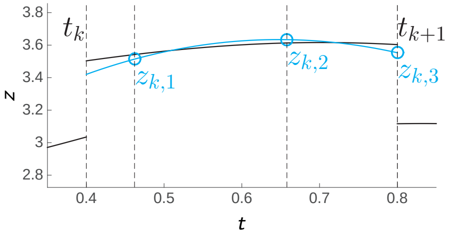
NLP with direct collocation

$$\min_{\mathbf{w}} \quad \Phi\left(\mathbf{w}\right)$$

$$\text{s.t.} \quad \mathbf{g}\left(\mathbf{w}\right) = \begin{bmatrix} \boldsymbol{\theta}_{0,0} - \bar{\mathbf{x}}_0 \\ \boldsymbol{\theta}_{0,K} - \boldsymbol{\theta}_{1,0} \\ \mathbf{F}\left(\frac{\partial}{\partial t}\mathbf{x}\left(\boldsymbol{\theta}_k, t_{k,0}\right), \boldsymbol{\theta}_{k,0}, \mathbf{z}_{k,0}, \mathbf{u}_k\right) \\ \dots \\ \boldsymbol{\theta}_{k,K} - \boldsymbol{\theta}_{k+1,0} \\ \mathbf{F}\left(\frac{\partial}{\partial t}\mathbf{x}\left(\boldsymbol{\theta}_k, t_{k,K}\right), \boldsymbol{\theta}_{k,i}, \mathbf{z}_{k,K}, \mathbf{u}_k\right) \\ \dots \end{bmatrix}$$

Remaining integration constraints $k = 1, ..., N-1$





NLP from Direct Collocation for DAE-constrained OCPs

Fully implicit DAE:

$$\mathbf{F}\left(\dot{\mathbf{x}}, \mathbf{x}, \mathbf{z}, \mathbf{u}_k\right) = 0$$

Interpolation:

$$\mathbf{x}\left(\boldsymbol{\theta}_k, t\right) = \sum_{i=0}^{K} \boldsymbol{\theta}_{k,i} P_{k,i}(t)$$

$$\mathbf{z}\left(\mathbf{z}_k, t\right) = \sum_{i=1}^{K} \mathbf{z}_{k,i} P_{k,i}(t)$$

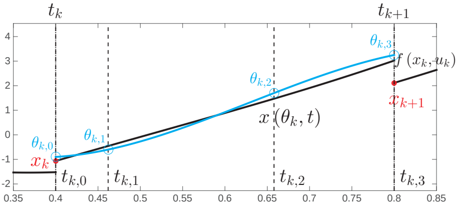
NLP with direct collocation

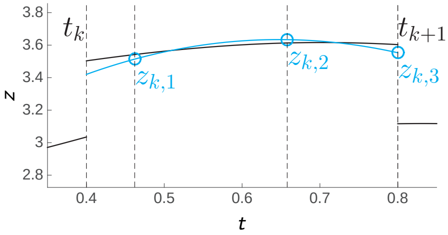
$$\min_{\mathbf{w}} \quad \Phi\left(\mathbf{w}\right)$$

$$\text{s.t.} \quad \mathbf{g}\left(\mathbf{w}\right) = \begin{bmatrix} \boldsymbol{\theta}_{0,0} - \bar{\mathbf{x}}_0 \\ \boldsymbol{\theta}_{0,K} - \boldsymbol{\theta}_{1,0} \\ \mathbf{F}\left(\frac{\partial}{\partial t}\mathbf{x}\left(\boldsymbol{\theta}_k, t_{k,0}\right), \boldsymbol{\theta}_{k,0}, \mathbf{z}_{k,0}, \mathbf{u}_k\right) \\ \dots \\ \boldsymbol{\theta}_{k,K} - \boldsymbol{\theta}_{k+1,0} \\ \mathbf{F}\left(\frac{\partial}{\partial t}\mathbf{x}\left(\boldsymbol{\theta}_k, t_{k,K}\right), \boldsymbol{\theta}_{k,i}, \mathbf{z}_{k,K}, \mathbf{u}_k\right) \\ \dots \end{bmatrix}$$

Decision variables ($k = 0, ..., N-1$)

$$\mathbf{w} = \left\{..., \boldsymbol{\theta}_{k,0}, \boldsymbol{\theta}_{k,1}, \mathbf{z}_{k,1}, ..., \boldsymbol{\theta}_{k,K}, \mathbf{z}_{k,K}, \mathbf{u}_k, ...\right\}$$





NLP from Direct Collocation for DAE-constrained OCPs

Fully implicit DAE:

$$\mathbf{F}\left(\dot{\mathbf{x}}, \mathbf{x}, \mathbf{z}, \mathbf{u}_k\right) = 0$$

Interpolation:

$$\mathbf{x}\left(\boldsymbol{\theta}_k, t\right) = \sum_{i=0}^{K} \boldsymbol{\theta}_{k,i} P_{k,i}(t)$$

$$\mathbf{z}\left(\mathbf{z}_k, t\right) = \sum_{i=1}^{K} \mathbf{z}_{k,i} P_{k,i}(t)$$

Note: for $\mathbf{z}$, the interpolation plays no role in the collocation equations !

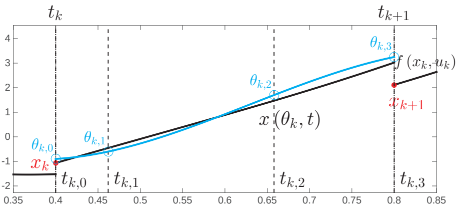
NLP with direct collocation

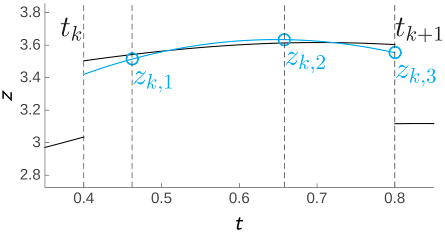
$$\min_{\mathbf{w}} \quad \Phi\left(\mathbf{w}\right)$$

$$\text{s.t.} \quad \mathbf{g}\left(\mathbf{w}\right) = \begin{bmatrix} \boldsymbol{\theta}_{0,0} - \bar{\mathbf{x}}_0 \\ \boldsymbol{\theta}_{0,K} - \boldsymbol{\theta}_{1,0} \\ \mathbf{F}\left(\frac{\partial}{\partial t}\mathbf{x}\left(\boldsymbol{\theta}_k, t_{k,0}\right), \boldsymbol{\theta}_{k,0}, \mathbf{z}_{k,0}, \mathbf{u}_k\right) \\ \ldots \\ \boldsymbol{\theta}_{k,K} - \boldsymbol{\theta}_{k+1,0} \\ \mathbf{F}\left(\frac{\partial}{\partial t}\mathbf{x}\left(\boldsymbol{\theta}_k, t_{k,K}\right), \boldsymbol{\theta}_{k,i}, \mathbf{z}_{k,K}, \mathbf{u}_k\right) \\ \ldots \end{bmatrix}$$

Decision variables ($k = 0, \ldots, N-1$)

$$\mathbf{w} = \left\{\ldots, \boldsymbol{\theta}_{k,0}, \boldsymbol{\theta}_{k,1}, \mathbf{z}_{k,1}, \ldots, \boldsymbol{\theta}_{k,K}, \mathbf{z}_{k,K}, \mathbf{u}_k, \ldots\right\}$$

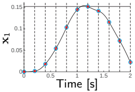


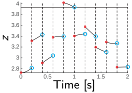


Direct Methods for DAE-based OCPs - Wrap up

Multiple-shooting

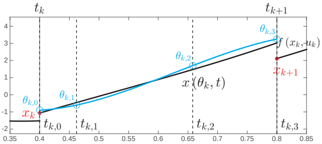
- Hides the algebraic variables $\mathbf{z}$ in the integrator
- If they are needed in the constraints/cost, the integrator needs to report them back to the NLP solver, with sensitivities.

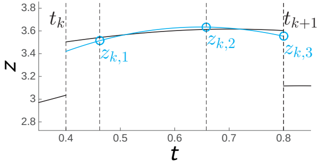




Direct Collocation:

- collocation equations are *almost* the same as for ODEs
- A discrete instance of the algebraic variables exists at every collocation time <u>but the first one</u> (associated to the continuity conditions)
- Use the Radau collocation times
- Carefule about very high orders in the collocation polynomial !





Outline

Reminder - LICQ condition

NLP:

$$\begin{aligned} \min_{\mathbf{w}} \quad & \Phi(\mathbf{w}) \\ \text{s.t.} \quad & \mathbf{g}(\mathbf{w}) = 0 \end{aligned}$$

has LICQ at its solution $\mathbf{w}^\star$ if:

$$\nabla \mathbf{g}(\mathbf{w}^\star)$$

is full column rank.

Reminder - LICQ condition

NLP:

$$\begin{aligned} \min_{\mathbf{w}} \quad & \Phi(\mathbf{w}) \\ \text{s.t.} \quad & \mathbf{g}(\mathbf{w}) = 0 \end{aligned}$$

has LICQ at its solution $\mathbf{w}^\star$ if:

$$\nabla \mathbf{g}(\mathbf{w}^\star)$$

is full column rank.

LICQ fails if

Reminder - LICQ condition

NLP:

$$\begin{aligned} \min_{\mathbf{w}} \quad & \Phi(\mathbf{w}) \\ \text{s.t.} \quad & \mathbf{g}(\mathbf{w}) = 0 \end{aligned}$$

has LICQ at its solution $\mathbf{w}^\star$ if:

$$\nabla \mathbf{g}(\mathbf{w}^\star)$$

is full column rank.

LICQ fails if

- For some linear combination

$$\sum_i v_i \cdot \nabla \mathbf{g}_i = 0 \quad \text{with some } v_i \neq 0$$

Reminder - LICQ condition

NLP:

$$\begin{aligned} \min_{\mathbf{w}} \quad & \Phi(\mathbf{w}) \\ \text{s.t.} \quad & \mathbf{g}(\mathbf{w}) = 0 \end{aligned}$$

has LICQ at its solution $\mathbf{w}^\star$ if:

$$\nabla \mathbf{g}(\mathbf{w}^\star)$$

is full column rank.

LICQ fails if

- For some linear combination

$$\sum_i v_i \cdot \nabla \mathbf{g}_i = 0 \quad \text{with some } v_i \neq 0$$

- For some vector $\mathbf{v} \neq 0$

$$\nabla \mathbf{g} \cdot \mathbf{v} = 0$$

Reminder - LICQ condition

NLP:

$$\begin{aligned} \min_{\mathbf{w}} \quad & \Phi(\mathbf{w}) \\ \text{s.t.} \quad & \mathbf{g}(\mathbf{w}) = 0 \end{aligned}$$

has LICQ at its solution $\mathbf{w}^\star$ if:

$$\nabla \mathbf{g}(\mathbf{w}^\star)$$

is full column rank.

LICQ fails if

- For some linear combination

$$\sum_i v_i \cdot \nabla \mathbf{g}_i = 0 \quad \text{with some } v_i \neq 0$$

- For some vector $\mathbf{v} \neq 0$

$$\nabla \mathbf{g} \cdot \mathbf{v} = 0$$

- For some vector $\mathbf{v} \neq 0$

$$\mathbf{v}^\top \frac{\partial \mathbf{g}}{\partial \mathbf{w}} = 0$$

Reminder - LICQ condition

NLP:

$$\begin{aligned} \min_{\mathbf{w}} \quad & \Phi(\mathbf{w}) \\ \text{s.t.} \quad & \mathbf{g}(\mathbf{w}) = 0 \end{aligned}$$

has LICQ at its solution $\mathbf{w}^\star$ if:

$$\nabla \mathbf{g}(\mathbf{w}^\star)$$

is full column rank.

Why is LICQ important ?

Newton step on the NLP:

$$\underbrace{\begin{bmatrix} \nabla^2 \mathcal{L} & \nabla \mathbf{g} \\ \nabla \mathbf{g}^\top & 0 \end{bmatrix}}_{\text{KKT}} \begin{bmatrix} \Delta \mathbf{w} \\ \boldsymbol{\lambda} \end{bmatrix} = - \begin{bmatrix} \nabla \Phi \\ \mathbf{g} \end{bmatrix}$$

KKT matrix becomes rank-deficient for $\nabla \mathbf{g}$ rank-deficient !!

LICQ fails if

- For some linear combination
$$\sum_i v_i \cdot \nabla \mathbf{g}_i = 0 \quad \text{with some } v_i \neq 0$$
- For some vector $\mathbf{v} \neq 0$
$$\nabla \mathbf{g} \cdot \mathbf{v} = 0$$
- For some vector $\mathbf{v} \neq 0$
$$\mathbf{v}^\top \frac{\partial \mathbf{g}}{\partial \mathbf{w}} = 0$$
- For some matrix $M \neq 0$
$$\nabla \mathbf{g} \cdot M = 0$$

Reminder - LICQ condition

NLP:

$$\begin{aligned} \min_{\mathbf{w}} \quad & \Phi(\mathbf{w}) \\ \text{s.t.} \quad & \mathbf{g}(\mathbf{w}) = 0 \end{aligned}$$

has LICQ at its solution $\mathbf{w}^\star$ if:

$$\nabla \mathbf{g}(\mathbf{w}^\star)$$

is full column rank.

Why is LICQ important ?

Newton step on the NLP:

$$\underbrace{\begin{bmatrix} \nabla^2 \mathcal{L} & \nabla \mathbf{g} \\ \nabla \mathbf{g}^\top & 0 \end{bmatrix}}_{\text{KKT}} \begin{bmatrix} \Delta \mathbf{w} \\ \boldsymbol{\lambda} \end{bmatrix} = -\begin{bmatrix} \nabla \Phi \\ \mathbf{g} \end{bmatrix}$$

KKT matrix becomes rank-deficient for $\nabla \mathbf{g}$ rank-deficient !!

LICQ fails if

- For some linear combination

$$\sum_i v_i \cdot \nabla \mathbf{g}_i = 0 \quad \text{with some } v_i \neq 0$$

- For some vector $\mathbf{v} \neq 0$

$$\nabla \mathbf{g} \cdot \mathbf{v} = 0$$

- For some vector $\mathbf{v} \neq 0$

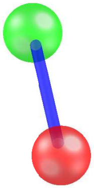
$$\mathbf{v}^\top \frac{\partial \mathbf{g}}{\partial \mathbf{w}} = 0$$

- For some matrix $M \neq 0$

$$\nabla \mathbf{g} \cdot M = 0$$

Some NLP solvers attempt "fixes" in your problem in case of LICQ deficiency. They often fail when the "fixing" is not trivial to do...

Example of Point-to-Point motion - Two linked masses

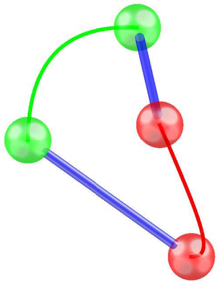


With generalized coordinates:

$$\mathbf{q} = \begin{bmatrix} \mathbf{p}_1 \\ \mathbf{p}_2 \end{bmatrix}$$

Dynamics preserve the distance $\|\mathbf{p}_2 - \mathbf{p}_1\|$

Example of Point-to-Point motion - Two linked masses



With generalized coordinates:

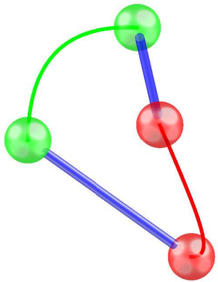
$$\mathbf{q} = \begin{bmatrix} \mathbf{p}_1 \\ \mathbf{p}_2 \end{bmatrix}$$

Dynamics preserve the distance $\|\mathbf{p}_2 - \mathbf{p}_1\|$

OCP

$$\begin{aligned} \min \quad & \Phi\left(\mathbf{x}(.), \mathbf{u}(.)\right) \\ \text{s.t.} \quad & \mathbf{F}\left(\dot{\mathbf{x}}\left(t\right), \mathbf{z}\left(t\right), \mathbf{x}\left(t\right), \mathbf{u}\left(t\right)\right) = 0 \\ & \mathbf{x}\left(t_0\right) = \bar{\mathbf{x}}_0, \quad \mathbf{x}\left(t_\mathrm{f}\right) = \bar{\mathbf{x}}_\mathrm{f} \end{aligned}$$

Example of Point-to-Point motion - Two linked masses



With generalized coordinates:

$$\mathbf{q} = \begin{bmatrix} \mathbf{p}_1 \\ \mathbf{p}_2 \end{bmatrix}$$

Dynamics preserve the distance $\|\mathbf{p}_2 - \mathbf{p}_1\|$

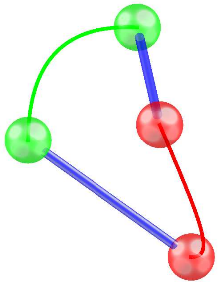
OCP

$$\begin{aligned} \text{min} \quad & \Phi\left(\mathbf{x}(.), \mathbf{u}(.)\right) \\ \text{s.t.} \quad & \mathbf{F}\left(\dot{\mathbf{x}}(t), \mathbf{z}(t), \mathbf{x}(t), \mathbf{u}(t)\right) = 0 \\ & \mathbf{x}(t_0) = \bar{\mathbf{x}}_0, \quad \mathbf{x}(t_f) = \bar{\mathbf{x}}_f \end{aligned}$$

LICQ problem

- Initial condition imposes the distance $\|\mathbf{p}_2 - \mathbf{p}_1\|$

Example of Point-to-Point motion - Two linked masses



With generalized coordinates:

$$\mathbf{q} = \begin{bmatrix} \mathbf{p}_1 \\ \mathbf{p}_2 \end{bmatrix}$$

Dynamics preserve the distance $\|\mathbf{p}_2 - \mathbf{p}_1\|$

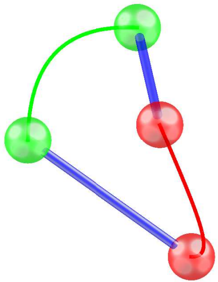
OCP

$$\begin{aligned} \min \quad & \Phi\left(\mathbf{x}(.), \mathbf{u}(.)\right) \\ \text{s.t.} \quad & \mathbf{F}\left(\dot{\mathbf{x}}(t), \mathbf{z}(t), \mathbf{x}(t), \mathbf{u}(t)\right) = 0 \\ & \mathbf{x}(t_0) = \bar{\mathbf{x}}_0, \quad \mathbf{x}(t_\mathrm{f}) = \bar{\mathbf{x}}_\mathrm{f} \end{aligned}$$

LICQ problem

- Initial condition imposes the distance $\|\mathbf{p}_2 - \mathbf{p}_1\|$
- Dynamics impose the distance $\|\mathbf{p}_2 - \mathbf{p}_1\|$ at final time

Example of Point-to-Point motion - Two linked masses



With generalized coordinates:

$$\mathbf{q} = \begin{bmatrix} \mathbf{p}_1 \\ \mathbf{p}_2 \end{bmatrix}$$

Dynamics preserve the distance $\|\mathbf{p}_2 - \mathbf{p}_1\|$

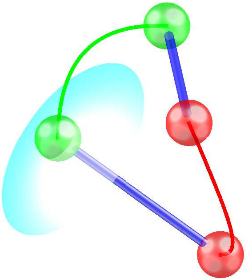
OCP

$$\begin{aligned} \min \quad & \Phi\left(\mathbf{x}(.), \mathbf{u}(.)\right) \\ \text{s.t.} \quad & \mathbf{F}\left(\dot{\mathbf{x}}(t), \mathbf{z}(t), \mathbf{x}(t), \mathbf{u}(t)\right) = 0 \\ & \mathbf{x}(t_0) = \bar{\mathbf{x}}_0, \quad \mathbf{x}(t_\mathrm{f}) = \bar{\mathbf{x}}_\mathrm{f} \end{aligned}$$

LICQ problem

- Initial condition imposes the distance $\|\mathbf{p}_2 - \mathbf{p}_1\|$
- Dynamics impose the distance $\|\mathbf{p}_2 - \mathbf{p}_1\|$ at final time
- Terminal condition clamps the two final positions...

Example of Point-to-Point motion - Two linked masses



With generalized coordinates:

$$\mathbf{q} = \begin{bmatrix} \mathbf{p}_1 \\ \mathbf{p}_2 \end{bmatrix}$$

Dynamics preserve the distance $\|\mathbf{p}_2 - \mathbf{p}_1\|$

OCP

$$\begin{aligned} \min \quad & \Phi\left(\mathbf{x}(.), \mathbf{u}(.)\right) \\ \text{s.t.} \quad & \mathbf{F}\left(\dot{\mathbf{x}}(t), \mathbf{z}(t), \mathbf{x}(t), \mathbf{u}(t)\right) = 0 \\ & \mathbf{x}(t_0) = \bar{\mathbf{x}}_0, \quad \mathbf{x}(t_\mathrm{f}) = \bar{\mathbf{x}}_\mathrm{f} \end{aligned}$$

LICQ problem

- Initial condition imposes the distance $\|\mathbf{p}_2 - \mathbf{p}_1\|$
- Dynamics impose the distance $\|\mathbf{p}_2 - \mathbf{p}_1\|$ at final time
- Terminal condition clamps the two final positions...

If the distance and mass 1 are fixed at final time, then mass 2 is **free only on a 2-dimensional** manifold. But the position of mass 2 at final time is imposed via **3 constraints** !! The problem is overconstrained...

Point-to-point motion with Index-reduced DAEs

OCP:

$$\begin{aligned} \text{min} \quad & \Phi\left(\mathbf{x}(.), \mathbf{u}(.)\right) \\ \text{s.t.} \quad & \mathbf{F}\left(\dot{\mathbf{x}}\left(t\right), \mathbf{z}\left(t\right), \mathbf{x}\left(t\right), \mathbf{u}\left(t\right)\right) = 0 \\ & \mathbf{x}\left(t_0\right) = \bar{\mathbf{x}}_0, \quad \mathbf{x}\left(t_\mathrm{f}\right) = \bar{\mathbf{x}}_\mathrm{f} \end{aligned}$$

$\mathbf{f}\left(\mathbf{x}_k, \mathbf{u}_k\right)$ integrates the dynamics $\mathbf{F}$ over the time interval $[t_k,\ t_{k+1}]$. Label $\mathbf{C}$ the consistency conditions. Note that $\mathbf{C}\left(\bar{\mathbf{x}}_0\right) = 0$ is preserved by $\mathbf{f}$.

Point-to-point motion with Index-reduced DAEs

OCP:

$$\begin{array}{ll} \min & \Phi\left(\mathbf{x}(.), \mathbf{u}(.)\right) \\ \text{s.t.} & \mathbf{F}\left(\dot{\mathbf{x}}\left(t\right), \mathbf{z}\left(t\right), \mathbf{x}\left(t\right), \mathbf{u}\left(t\right)\right) = 0 \\ & \mathbf{x}\left(t_0\right) = \bar{\mathbf{x}}_0, \quad \mathbf{x}\left(t_\mathrm{f}\right) = \bar{\mathbf{x}}_\mathrm{f} \end{array}$$

$\mathbf{f}\left(\mathbf{x}_k, \mathbf{u}_k\right)$ integrates the dynamics $\mathbf{F}$ over the time interval $[t_k,\ t_{k+1}]$. Label $\mathbf{C}$ the consistency conditions. Note that $\mathbf{C}\left(\bar{\mathbf{x}}_0\right) = 0$ is preserved by $\mathbf{f}$.

Proposition if

$$\mathbf{C}\left(\mathbf{w}\right) = 0, \quad \forall\, \mathbf{w} \quad \text{s.t.} \quad \mathbf{g}\left(\mathbf{w}\right) = 0$$

then $\nabla\mathbf{C} \in \mathrm{span}\left\{\nabla\mathbf{g}\right\}$

Proof: for any $\mathbf{d}$ *such that* $\nabla\mathbf{g}^\top\mathbf{d} = 0$, *equality:*

$$\nabla\mathbf{C}^\top\mathbf{d} = 0$$

holds.

Point-to-point motion with Index-reduced DAEs

OCP:

$$\begin{aligned} \min \quad & \Phi\left(\mathbf{x}(.), \mathbf{u}(.)\right) \\ \text{s.t.} \quad & \mathbf{F}\left(\dot{\mathbf{x}}(t), \mathbf{z}(t), \mathbf{x}(t), \mathbf{u}(t)\right) = 0 \\ & \mathbf{x}(t_0) = \bar{\mathbf{x}}_0, \quad \mathbf{x}(t_{\mathrm{f}}) = \bar{\mathbf{x}}_{\mathrm{f}} \end{aligned}$$

$\mathbf{f}\left(\mathbf{x}_k, \mathbf{u}_k\right)$ integrates the dynamics $\mathbf{F}$ over the time interval $[t_k,\ t_{k+1}]$. Label $\mathbf{C}$ the consistency conditions. Note that $\mathbf{C}\left(\bar{\mathbf{x}}_0\right) = 0$ is preserved by $\mathbf{f}$.

Corollary: matrix $\begin{bmatrix} \nabla \mathbf{g} & \nabla \mathbf{T} \end{bmatrix}$ is rank-deficient if

$$\begin{aligned} \mathbf{g}(\mathbf{w}) = 0 \quad & \Rightarrow \quad \mathbf{C}(\mathbf{w}) = 0 \quad \text{and} \\ \mathbf{T}(\mathbf{w}) = 0 \quad & \Rightarrow \quad \mathbf{C}(\mathbf{w}) = 0 \end{aligned}$$

Proof: observe that

$$\nabla \mathbf{C} = \nabla \mathbf{g} \boldsymbol{\alpha} = \nabla \mathbf{T} \boldsymbol{\beta}$$

then

$$\begin{bmatrix} \nabla \mathbf{g} & \nabla \mathbf{T} \end{bmatrix} \begin{bmatrix} \boldsymbol{\alpha} \\ -\boldsymbol{\beta} \end{bmatrix} = 0$$

Point-to-point motion with Index-reduced DAEs

OCP:

$$\begin{aligned} \min \quad & \Phi\left(\mathbf{x}(.), \mathbf{u}(.)\right) \\ \text{s.t.} \quad & \mathbf{F}\left(\dot{\mathbf{x}}\left(t\right), \mathbf{z}\left(t\right), \mathbf{x}\left(t\right), \mathbf{u}\left(t\right)\right) = 0 \\ & \mathbf{x}\left(t_0\right) = \bar{\mathbf{x}}_0, \quad \mathbf{x}\left(t_\mathrm{f}\right) = \bar{\mathbf{x}}_\mathrm{f} \end{aligned}$$

$\mathbf{f}\left(\mathbf{x}_k, \mathbf{u}_k\right)$ integrates the dynamics $\mathbf{F}$ over the time interval $[t_k, \ t_{k+1}]$. Label $\mathbf{C}$ the consistency conditions. Note that $\mathbf{C}\left(\bar{\mathbf{x}}_0\right) = 0$ is preserved by $\mathbf{f}$.

$$\begin{aligned} \min_{\mathbf{w}} \quad & \Phi\left(\mathbf{w}\right) \\ \text{s.t.} \quad & \mathbf{g}\left(\mathbf{w}\right) = \left[\begin{array}{c} \bar{\mathbf{x}}_0 - \mathbf{x}_0 \\ \mathbf{f}\left(\mathbf{x}_0, \mathbf{u}_0\right) - \mathbf{x}_1 \\ \dots \\ \mathbf{f}\left(\mathbf{x}_{N-1}, \mathbf{u}_{N-1}\right) - \mathbf{x}_N \end{array}\right] = 0 \\ & \mathbf{T}\left(\mathbf{w}\right) = \mathbf{x}_N - \bar{\mathbf{x}}_\mathrm{f} = 0 \end{aligned}$$

Corollary: matrix $\left[\begin{array}{cc} \nabla \mathbf{g} & \nabla \mathbf{T} \end{array}\right]$ is rank-deficient if

$$\begin{aligned} \mathbf{g}\left(\mathbf{w}\right) = 0 \quad & \Rightarrow \quad \mathbf{C}\left(\mathbf{w}\right) = 0 \quad \text{and} \\ \mathbf{T}\left(\mathbf{w}\right) = 0 \quad & \Rightarrow \quad \mathbf{C}\left(\mathbf{w}\right) = 0 \end{aligned}$$

Proof: observe that

$$\nabla \mathbf{C} = \nabla \mathbf{g} \boldsymbol{\alpha} = \nabla \mathbf{T} \boldsymbol{\beta}$$

then

$$\left[\begin{array}{cc} \nabla \mathbf{g} & \nabla \mathbf{T} \end{array}\right] \left[\begin{array}{c} \boldsymbol{\alpha} \\ -\boldsymbol{\beta} \end{array}\right] = 0$$

Point-to-point motion with Index-reduced DAEs

OCP:

$$\begin{aligned} \min \quad & \Phi\left(\mathbf{x}(.), \mathbf{u}(.)\right) \\ \text{s.t.} \quad & \mathbf{F}\left(\dot{\mathbf{x}}(t), \mathbf{z}(t), \mathbf{x}(t), \mathbf{u}(t)\right) = 0 \\ & \mathbf{x}(t_0) = \bar{\mathbf{x}}_0, \quad \mathbf{x}(t_\mathrm{f}) = \bar{\mathbf{x}}_\mathrm{f} \end{aligned}$$

$\mathbf{f}(\mathbf{x}_k, \mathbf{u}_k)$ integrates the dynamics $\mathbf{F}$ over the time interval $[t_k,\ t_{k+1}]$. Label $\mathbf{C}$ the consistency conditions. Note that $\mathbf{C}(\bar{\mathbf{x}}_0) = 0$ is preserved by $\mathbf{f}$.

Corollary: matrix $\left[\begin{array}{cc} \nabla\mathbf{g} & \nabla\mathbf{T} \end{array}\right]$ is rank-deficient if

$$\begin{aligned} \mathbf{g}(\mathbf{w}) = 0 \quad & \Rightarrow \quad \mathbf{C}(\mathbf{w}) = 0 \quad \text{and} \\ \mathbf{T}(\mathbf{w}) = 0 \quad & \Rightarrow \quad \mathbf{C}(\mathbf{w}) = 0 \end{aligned}$$

Proof: observe that

$$\nabla\mathbf{C} = \nabla\mathbf{g}\boldsymbol{\alpha} = \nabla\mathbf{T}\boldsymbol{\beta}$$

then

$$\left[\begin{array}{cc} \nabla\mathbf{g} & \nabla\mathbf{T} \end{array}\right] \left[\begin{array}{c} \boldsymbol{\alpha} \\ -\boldsymbol{\beta} \end{array}\right] = 0$$

$$\begin{aligned} \min_{\mathbf{w}} \quad & \Phi(\mathbf{w}) \\ \text{s.t.} \quad & \mathbf{g}(\mathbf{w}) = \left[\begin{array}{c} \bar{\mathbf{x}}_0 - \mathbf{x}_0 \\ \mathbf{f}(\mathbf{x}_0, \mathbf{u}_0) - \mathbf{x}_1 \\ \dots \\ \mathbf{f}(\mathbf{x}_{N-1}, \mathbf{u}_{N-1}) - \mathbf{x}_N \end{array}\right] = 0 \\ & \mathbf{T}(\mathbf{w}) = \mathbf{x}_N - \bar{\mathbf{x}}_\mathrm{f} = 0 \end{aligned}$$

- If $\bar{\mathbf{x}}_0$ is consistent, i.e. $\mathbf{C}(\bar{\mathbf{x}}_0) = 0$ then $\mathbf{C}(\mathbf{x}_N) = 0$ is enforced via satisfying the dynamics $\mathbf{g}(\mathbf{w}) = 0$

Point-to-point motion with Index-reduced DAEs

OCP:

$$\begin{aligned} \min \quad & \Phi\left(\mathbf{x}(.), \mathbf{u}(.)\right) \\ \text{s.t.} \quad & \mathbf{F}\left(\dot{\mathbf{x}}(t), \mathbf{z}(t), \mathbf{x}(t), \mathbf{u}(t)\right) = 0 \\ & \mathbf{x}(t_0) = \bar{\mathbf{x}}_0, \quad \mathbf{x}(t_f) = \bar{\mathbf{x}}_f \end{aligned}$$

$\mathbf{f}\left(\mathbf{x}_k, \mathbf{u}_k\right)$ integrates the dynamics $\mathbf{F}$ over the time interval $[t_k,\ t_{k+1}]$. Label $\mathbf{C}$ the consistency conditions. Note that $\mathbf{C}\left(\bar{\mathbf{x}}_0\right) = 0$ is preserved by $\mathbf{f}$.

Corollary: matrix $\left[\begin{array}{cc} \nabla \mathbf{g} & \nabla \mathbf{T} \end{array}\right]$ is rank-deficient if

$$\begin{aligned} \mathbf{g}(\mathbf{w}) = 0 \quad &\Rightarrow \quad \mathbf{C}(\mathbf{w}) = 0 \quad \text{and} \\ \mathbf{T}(\mathbf{w}) = 0 \quad &\Rightarrow \quad \mathbf{C}(\mathbf{w}) = 0 \end{aligned}$$

Proof: observe that

$$\nabla \mathbf{C} = \nabla \mathbf{g} \boldsymbol{\alpha} = \nabla \mathbf{T} \boldsymbol{\beta}$$

then

$$\left[\begin{array}{cc} \nabla \mathbf{g} & \nabla \mathbf{T} \end{array}\right] \left[\begin{array}{c} \boldsymbol{\alpha} \\ -\boldsymbol{\beta} \end{array}\right] = 0$$

$$\begin{aligned} \min_{\mathbf{w}} \quad & \Phi(\mathbf{w}) \\ \text{s.t.} \quad & \mathbf{g}(\mathbf{w}) = \left[\begin{array}{c} \bar{\mathbf{x}}_0 - \mathbf{x}_0 \\ \mathbf{f}\left(\mathbf{x}_0, \mathbf{u}_0\right) - \mathbf{x}_1 \\ \dots \\ \mathbf{f}\left(\mathbf{x}_{N-1}, \mathbf{u}_{N-1}\right) - \mathbf{x}_N \end{array}\right] = 0 \\ & \mathbf{T}(\mathbf{w}) = \mathbf{x}_N - \bar{\mathbf{x}}_f = 0 \end{aligned}$$

- If $\bar{\mathbf{x}}_0$ is consistent, i.e. $\mathbf{C}\left(\bar{\mathbf{x}}_0\right) = 0$ then $\mathbf{C}\left(\mathbf{x}_N\right) = 0$ is enforced via satisfying the dynamics $\mathbf{g}(\mathbf{w}) = 0$
- If $\bar{\mathbf{x}}_f$ is consistent, i.e. $\mathbf{C}\left(\bar{\mathbf{x}}_f\right) = 0$ then $\mathbf{C}\left(\mathbf{x}_N\right) = 0$ is enforced via satisfying the terminal constraints $\mathbf{T}(\mathbf{w}) = 0$

Point-to-point motion with Index-reduced DAEs

OCP:

$$\begin{aligned} \min \quad & \Phi\left(\mathbf{x}(.), \mathbf{u}(.)\right) \\ \text{s.t.} \quad & \mathbf{F}\left(\dot{\mathbf{x}}(t), \mathbf{z}(t), \mathbf{x}(t), \mathbf{u}(t)\right) = 0 \\ & \mathbf{x}(t_0) = \bar{\mathbf{x}}_0, \quad \mathbf{x}(t_f) = \bar{\mathbf{x}}_f \end{aligned}$$

$\mathbf{f}\left(\mathbf{x}_k, \mathbf{u}_k\right)$ integrates the dynamics $\mathbf{F}$ over the time interval $[t_k, \ t_{k+1}]$. Label $\mathbf{C}$ the consistency conditions. Note that $\mathbf{C}\left(\bar{\mathbf{x}}_0\right) = 0$ is preserved by $\mathbf{f}$.

Corollary: matrix $\left[\begin{array}{cc} \nabla \mathbf{g} & \nabla \mathbf{T} \end{array}\right]$ is rank-deficient if

$$\begin{aligned} \mathbf{g}(\mathbf{w}) = 0 \quad &\Rightarrow \quad \mathbf{C}(\mathbf{w}) = 0 \quad \text{and} \\ \mathbf{T}(\mathbf{w}) = 0 \quad &\Rightarrow \quad \mathbf{C}(\mathbf{w}) = 0 \end{aligned}$$

Proof: observe that

$$\nabla \mathbf{C} = \nabla \mathbf{g} \boldsymbol{\alpha} = \nabla \mathbf{T} \boldsymbol{\beta}$$

then

$$\left[\begin{array}{cc} \nabla \mathbf{g} & \nabla \mathbf{T} \end{array}\right] \left[\begin{array}{c} \boldsymbol{\alpha} \\ -\boldsymbol{\beta} \end{array}\right] = 0$$

$$\begin{aligned} \min_{\mathbf{w}} \quad & \Phi(\mathbf{w}) \\ \text{s.t.} \quad & \mathbf{g}(\mathbf{w}) = \left[\begin{array}{c} \bar{\mathbf{x}}_0 - \mathbf{x}_0 \\ \mathbf{f}\left(\mathbf{x}_0, \mathbf{u}_0\right) - \mathbf{x}_1 \\ \ldots \\ \mathbf{f}\left(\mathbf{x}_{N-1}, \mathbf{u}_{N-1}\right) - \mathbf{x}_N \end{array}\right] = 0 \\ & \mathbf{T}(\mathbf{w}) = \mathbf{x}_N - \bar{\mathbf{x}}_f = 0 \end{aligned}$$

- If $\bar{\mathbf{x}}_0$ is consistent, i.e. $\mathbf{C}\left(\bar{\mathbf{x}}_0\right) = 0$ then $\mathbf{C}\left(\mathbf{x}_N\right) = 0$ is enforced via satisfying the dynamics $\mathbf{g}(\mathbf{w}) = 0$
- If $\bar{\mathbf{x}}_f$ is consistent, i.e. $\mathbf{C}\left(\bar{\mathbf{x}}_f\right) = 0$ then $\mathbf{C}\left(\mathbf{x}_N\right) = 0$ is enforced via satisfying the terminal constraints $\mathbf{T}(\mathbf{w}) = 0$

Then $\left[\begin{array}{cc} \nabla \mathbf{g} & \nabla \mathbf{T} \end{array}\right]$ is rank-deficient !!
The NLP fails LICQ

Point-to-point motion with Index-reduced DAEs - Projection Method

OCP:

$$\begin{aligned} \min \quad & \Phi\left(\mathbf{x}(.), \mathbf{u}(.)\right) \\ \text{s.t.} \quad & \mathbf{F}\left(\dot{\mathbf{x}}\left(t\right), \mathbf{z}\left(t\right), \mathbf{x}\left(t\right), \mathbf{u}\left(t\right)\right) = 0 \\ & \mathbf{x}\left(t_0\right) = \bar{\mathbf{x}}_0, \quad \mathbf{x}\left(t_f\right) = \bar{\mathbf{x}}_f \end{aligned}$$

$\mathbf{f}\left(\mathbf{x}_k, \mathbf{u}_k\right)$ integrates the dynamics $\mathbf{F}$ over the time interval $[t_k,\ t_{k+1}]$. Label $\mathbf{C} \in \mathbb{R}^m$ the consistency conditions. Note that $\mathbf{C}\left(\bar{\mathbf{x}}_0\right) = 0$ is preserved by $\mathbf{f}$.

$$\begin{aligned} \min_{\mathbf{w}} \quad & \Phi\left(\mathbf{w}\right) \\ \text{s.t.} \quad & \mathbf{g}\left(\mathbf{w}\right) = \begin{bmatrix} \bar{\mathbf{x}}_0 - \mathbf{x}_0 \\ \mathbf{f}\left(\mathbf{x}_0, \mathbf{u}_0\right) - \mathbf{x}_1 \\ \ldots \\ \mathbf{f}\left(\mathbf{x}_{N-1}, \mathbf{u}_{N-1}\right) - \mathbf{x}_N \end{bmatrix} = 0 \\ & \mathbf{T}\left(\mathbf{w}\right) = \mathbf{x}_N - \bar{\mathbf{x}}_f = 0 \end{aligned}$$

Matrix $\begin{bmatrix} \nabla \mathbf{g} & \nabla \mathbf{T} \end{bmatrix}$ is rank-deficient !!
The NLP fails LICQ

Point-to-point motion with Index-reduced DAEs - Projection Method

OCP:

$$\begin{aligned} \min \quad & \Phi\left(\mathbf{x}(.), \mathbf{u}(.)\right) \\ \text{s.t.} \quad & \mathbf{F}\left(\dot{\mathbf{x}}(t), \mathbf{z}(t), \mathbf{x}(t), \mathbf{u}(t)\right) = 0 \\ & \mathbf{x}(t_0) = \bar{\mathbf{x}}_0, \quad \mathbf{x}(t_\mathrm{f}) = \bar{\mathbf{x}}_\mathrm{f} \end{aligned}$$

$\mathbf{f}\left(\mathbf{x}_k, \mathbf{u}_k\right)$ integrates the dynamics $\mathbf{F}$ over the time interval $[t_k,\ t_{k+1}]$. Label $\mathbf{C} \in \mathbb{R}^m$ the consistency conditions. Note that $\mathbf{C}\left(\bar{\mathbf{x}}_0\right) = 0$ is preserved by $\mathbf{f}$.

$$\begin{aligned} \min_{\mathbf{w}} \quad & \Phi\left(\mathbf{w}\right) \\ \text{s.t.} \quad & \mathbf{g}\left(\mathbf{w}\right) = \begin{bmatrix} \bar{\mathbf{x}}_0 - \mathbf{x}_0 \\ \mathbf{f}\left(\mathbf{x}_0, \mathbf{u}_0\right) - \mathbf{x}_1 \\ \dots \\ \mathbf{f}\left(\mathbf{x}_{N-1}, \mathbf{u}_{N-1}\right) - \mathbf{x}_N \end{bmatrix} = 0 \\ & \mathbf{T}\left(\mathbf{w}\right) = \mathbf{x}_N - \bar{\mathbf{x}}_\mathrm{f} = 0 \end{aligned}$$

Let matrix $Z \in \mathbb{R}^{n \times n-m}$ be a basis of the "left-hand" null-space of $\nabla\mathbf{C}\left(\bar{\mathbf{x}}_\mathrm{f}\right)$, i.e.

$$Z^\top \nabla\mathbf{C}\left(\bar{\mathbf{x}}_\mathrm{f}\right) = 0$$

Modify the NLP according to...

Matrix $\begin{bmatrix} \nabla\mathbf{g} & \nabla\mathbf{T} \end{bmatrix}$ is rank-deficient !!
The NLP fails LICQ

Point-to-point motion with Index-reduced DAEs - Projection Method

OCP:

$$\begin{aligned} \min \quad & \Phi\left(\mathbf{x}(.), \mathbf{u}(.)\right) \\ \text{s.t.} \quad & \mathbf{F}\left(\dot{\mathbf{x}}\left(t\right), \mathbf{z}\left(t\right), \mathbf{x}\left(t\right), \mathbf{u}\left(t\right)\right) = 0 \\ & \mathbf{x}\left(t_0\right) = \bar{\mathbf{x}}_0, \quad \mathbf{x}\left(t_\mathrm{f}\right) = \bar{\mathbf{x}}_\mathrm{f} \end{aligned}$$

$\mathbf{f}\left(\mathbf{x}_k, \mathbf{u}_k\right)$ integrates the dynamics $\mathbf{F}$ over the time interval $[t_k,\ t_{k+1}]$. Label $\mathbf{C} \in \mathbb{R}^m$ the consistency conditions. Note that $\mathbf{C}\left(\bar{\mathbf{x}}_0\right) = 0$ is preserved by $\mathbf{f}$.

Let matrix $Z \in \mathbb{R}^{n \times n-m}$ be a basis of the "left-hand" null-space of $\nabla\mathbf{C}\left(\bar{\mathbf{x}}_\mathrm{f}\right)$, i.e.

$$Z^\top \nabla\mathbf{C}\left(\bar{\mathbf{x}}_\mathrm{f}\right) = 0$$

Modify the NLP according to...

$$\begin{aligned} \min_{\mathbf{w}} \quad & \Phi\left(\mathbf{w}\right) \\ \text{s.t.} \quad & \mathbf{g}\left(\mathbf{w}\right) = \begin{bmatrix} \bar{\mathbf{x}}_0 - \mathbf{x}_0 \\ \mathbf{f}\left(\mathbf{x}_0, \mathbf{u}_0\right) - \mathbf{x}_1 \\ \dots \\ \mathbf{f}\left(\mathbf{x}_{N-1}, \mathbf{u}_{N-1}\right) - \mathbf{x}_N \end{bmatrix} = 0 \\ & \mathbf{T}\left(\mathbf{w}\right) = Z^\top\left(\mathbf{x}_N - \bar{\mathbf{x}}_\mathrm{f}\right) = 0 \end{aligned}$$

Point-to-point motion with Index-reduced DAEs - Projection Method

OCP:

$$\begin{aligned}\min \quad & \Phi\left(\mathbf{x}(.), \mathbf{u}(.)\right) \\ \text{s.t.} \quad & \mathbf{F}\left(\dot{\mathbf{x}}\left(t\right), \mathbf{z}\left(t\right), \mathbf{x}\left(t\right), \mathbf{u}\left(t\right)\right) = 0 \\ & \mathbf{x}\left(t_0\right) = \bar{\mathbf{x}}_0, \quad \mathbf{x}\left(t_\mathrm{f}\right) = \bar{\mathbf{x}}_\mathrm{f}\end{aligned}$$

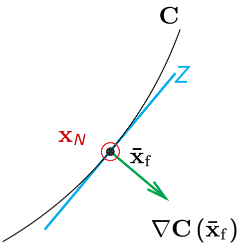
$\mathbf{f}\left(\mathbf{x}_k, \mathbf{u}_k\right)$ integrates the dynamics $\mathbf{F}$ over the time interval $[t_k,\ t_{k+1}]$. Label $\mathbf{C} \in \mathbb{R}^m$ the consistency conditions. Note that $\mathbf{C}\left(\bar{\mathbf{x}}_0\right) = 0$ is preserved by $\mathbf{f}$.

$$\begin{aligned}\min_{\mathbf{w}} \quad & \Phi\left(\mathbf{w}\right) \\ \text{s.t.} \quad & \mathbf{g}\left(\mathbf{w}\right) = \begin{bmatrix} \bar{\mathbf{x}}_0 - \mathbf{x}_0 \\ \mathbf{f}\left(\mathbf{x}_0, \mathbf{u}_0\right) - \mathbf{x}_1 \\ \dots \\ \mathbf{f}\left(\mathbf{x}_{N-1}, \mathbf{u}_{N-1}\right) - \mathbf{x}_N \end{bmatrix} = 0 \\ & \mathbf{T}\left(\mathbf{w}\right) = Z^\top\left(\mathbf{x}_N - \bar{\mathbf{x}}_\mathrm{f}\right) = 0\end{aligned}$$

Let matrix $Z \in \mathbb{R}^{n \times n-m}$ be a basis of the "left-hand" null-space of $\nabla\mathbf{C}\left(\bar{\mathbf{x}}_\mathrm{f}\right)$, i.e.

$$Z^\top \nabla\mathbf{C}\left(\bar{\mathbf{x}}_\mathrm{f}\right) = 0$$

Modify the NLP according to...



Point-to-point motion with Index-reduced DAEs - Projection Method

OCP:

$$\begin{aligned} \min \quad & \Phi\left(\mathbf{x}(.), \mathbf{u}(.)\right) \\ \text{s.t.} \quad & \mathbf{F}\left(\dot{\mathbf{x}}(t), \mathbf{z}(t), \mathbf{x}(t), \mathbf{u}(t)\right) = 0 \\ & \mathbf{x}(t_0) = \bar{\mathbf{x}}_0, \quad \mathbf{x}(t_f) = \bar{\mathbf{x}}_f \end{aligned}$$

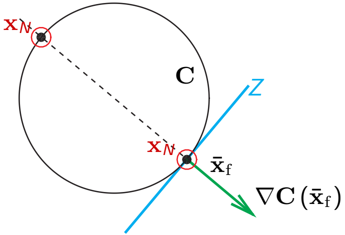
$\mathbf{f}(\mathbf{x}_k, \mathbf{u}_k)$ integrates the dynamics $\mathbf{F}$ over the time interval $[t_k,\ t_{k+1}]$. Label $\mathbf{C} \in \mathbb{R}^m$ the consistency conditions. Note that $\mathbf{C}(\bar{\mathbf{x}}_0) = 0$ is preserved by $\mathbf{f}$.

$$\begin{aligned} \min_{\mathbf{w}} \quad & \Phi(\mathbf{w}) \\ \text{s.t.} \quad & \mathbf{g}(\mathbf{w}) = \begin{bmatrix} \bar{\mathbf{x}}_0 - \mathbf{x}_0 \\ \mathbf{f}(\mathbf{x}_0, \mathbf{u}_0) - \mathbf{x}_1 \\ \dots \\ \mathbf{f}(\mathbf{x}_{N-1}, \mathbf{u}_{N-1}) - \mathbf{x}_N \end{bmatrix} = 0 \\ & \mathbf{T}(\mathbf{w}) = Z^\top(\mathbf{x}_N - \bar{\mathbf{x}}_f) = 0 \end{aligned}$$

Let matrix $Z \in \mathbb{R}^{n \times n-m}$ be a basis of the "left-hand" null-space of $\nabla\mathbf{C}(\bar{\mathbf{x}}_f)$, i.e.

$$Z^\top \nabla \mathbf{C}(\bar{\mathbf{x}}_f) = 0$$

Modify the NLP according to...



Point-to-point motion with Index-reduced DAEs - Projection Method

OCP:

$$\begin{aligned} \min \quad & \Phi\left(\mathbf{x}(.), \mathbf{u}(.)\right) \\ \text{s.t.} \quad & \mathbf{F}\left(\dot{\mathbf{x}}\left(t\right), \mathbf{z}\left(t\right), \mathbf{x}\left(t\right), \mathbf{u}\left(t\right)\right) = 0 \\ & \mathbf{x}\left(t_0\right) = \bar{\mathbf{x}}_0, \quad \mathbf{x}\left(t_\mathrm{f}\right) = \bar{\mathbf{x}}_\mathrm{f} \end{aligned}$$

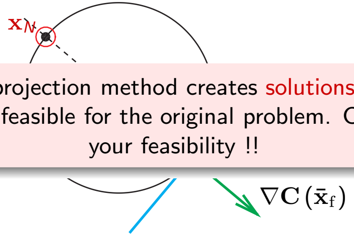
$\mathbf{f}\left(\mathbf{x}_k, \mathbf{u}_k\right)$ integrates the dynamics $\mathbf{F}$ over the time interval $[t_k,\ t_{k+1}]$. Label $\mathbf{C} \in \mathbb{R}^m$ the consistency conditions. Note that $\mathbf{C}\left(\bar{\mathbf{x}}_0\right) = 0$ is preserved by $\mathbf{f}$.

$$\begin{aligned} \min_{\mathbf{w}} \quad & \Phi\left(\mathbf{w}\right) \\ \text{s.t.} \quad & \mathbf{g}\left(\mathbf{w}\right) = \begin{bmatrix} \bar{\mathbf{x}}_0 - \mathbf{x}_0 \\ \mathbf{f}\left(\mathbf{x}_0, \mathbf{u}_0\right) - \mathbf{x}_1 \\ \dots \\ \mathbf{f}\left(\mathbf{x}_{N-1}, \mathbf{u}_{N-1}\right) - \mathbf{x}_N \end{bmatrix} = 0 \\ & \mathbf{T}\left(\mathbf{w}\right) = Z^\top\left(\mathbf{x}_N - \bar{\mathbf{x}}_\mathrm{f}\right) = 0 \end{aligned}$$

Let matrix $Z \in \mathbb{R}^{n \times n-m}$ be a basis of the "left-hand" null-space of $\nabla\mathbf{C}\left(\bar{\mathbf{x}}_\mathrm{f}\right)$, i.e.

$$Z^\top \nabla\mathbf{C}\left(\bar{\mathbf{x}}_\mathrm{f}\right) = 0$$

Modify the NLP according to...



The projection method creates solutions that are infeasible for the original problem. Check your feasibility !!

Outline

Constraints drift - Reminder

Index-1 DAE:

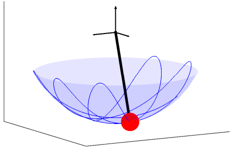
$$\begin{bmatrix} ml & \mathbf{p} \\ \mathbf{p}^\top & 0 \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{p}} \\ z \end{bmatrix} = \begin{bmatrix} \mathbf{u} - mg\mathbf{e}_3 \\ -\dot{\mathbf{p}}^\top \dot{\mathbf{p}} \end{bmatrix}$$

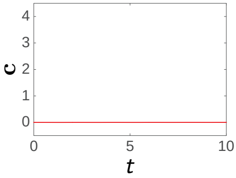
impose $\ddot{\mathbf{c}} = 0$ at all time.

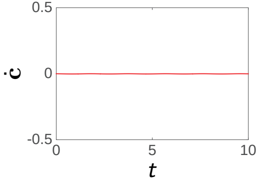
With the **consistency conditions**:

$$\mathbf{c} = \frac{1}{2}\left(\mathbf{p}^\top \mathbf{p} - L^2\right) = 0, \qquad \dot{\mathbf{c}} = \mathbf{p}^\top \dot{\mathbf{p}} = 0$$

imposed at t_0 result in $\dot{\mathbf{c}} = 0$ and $\mathbf{c} = 0$ holding at all time.







Constraints drift - Reminder

Index-1 DAE:

$$\begin{bmatrix} mI & \mathbf{p} \\ \mathbf{p}^\top & 0 \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{p}} \\ z \end{bmatrix} = \begin{bmatrix} \mathbf{u} - mg\mathbf{e}_3 \\ -\dot{\mathbf{p}}^\top \dot{\mathbf{p}} \end{bmatrix}$$

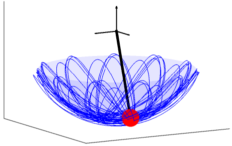
impose $\ddot{\mathbf{c}} = 0$ at all time.

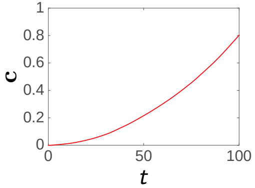
With the **consistency conditions**:

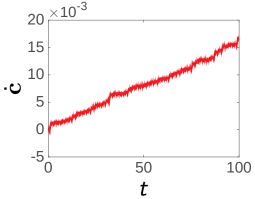
$$\mathbf{c} = \frac{1}{2}\left(\mathbf{p}^\top \mathbf{p} - L^2\right) = 0, \qquad \dot{\mathbf{c}} = \mathbf{p}^\top \dot{\mathbf{p}} = 0$$

imposed at t_0 result in $\dot{\mathbf{c}} = 0$ and $\mathbf{c} = 0$ holding at all time.

However, consistency $\mathbf{c} = 0$ and $\dot{\mathbf{c}} = 0$ are satisfied at all time only with no numerical error in the integration.







Constraints drift - Reminder

Index-1 DAE:

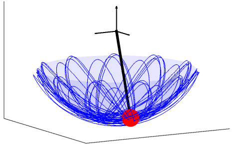
$$\begin{bmatrix} m\mathrm{I} & \mathbf{p} \\ \mathbf{p}^\top & 0 \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{p}} \\ z \end{bmatrix} = \begin{bmatrix} \mathbf{u} - mg\mathbf{e}_3 \\ -\dot{\mathbf{p}}^\top \dot{\mathbf{p}} \end{bmatrix}$$

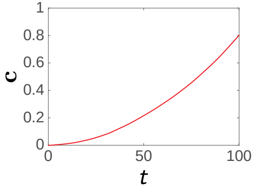
impose $\ddot{\mathbf{c}} = 0$ at all time.

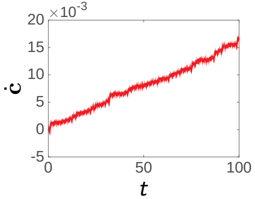
With the **consistency conditions**:

$$\mathbf{c} = \frac{1}{2}\left(\mathbf{p}^\top \mathbf{p} - L^2\right) = 0, \qquad \dot{\mathbf{c}} = \mathbf{p}^\top \dot{\mathbf{p}} = 0$$

imposed at t_0 result in $\dot{\mathbf{c}} = 0$ and $\mathbf{c} = 0$ holding at all time.







However, consistency $\mathbf{c} = 0$ <u>and</u> $\dot{\mathbf{c}} = 0$ are satisfied at all time only with no numerical error in the integration. <u>Always</u> check your consistency at the solution of your OCP when you work with index-reduced DAEs !!

How to handle drift (if needed) ? E.g. multiple-shooting

OCP:

$$\begin{aligned} \text{min} \quad & \Phi\left(\mathbf{x}(.), \mathbf{u}(.)\right) \\ \text{s.t.} \quad & \mathbf{F}\left(\dot{\mathbf{x}}\left(t\right), \mathbf{z}\left(t\right), \mathbf{x}\left(t\right), \mathbf{u}\left(t\right)\right) = 0 \\ & \mathbf{x}\left(t_0\right) = \bar{\mathbf{x}}_0 \end{aligned}$$

$\mathbf{f}\left(\mathbf{x}_k, \mathbf{u}_k\right)$ integrates the dynamics $\mathbf{F}$ over the time interval $[t_k,\ t_{k+1}]$. Label $\mathbf{C}$ the consistency conditions. Note that $\mathbf{C}\left(\bar{\mathbf{x}}_0\right) = 0$ is preserved by $\mathbf{f}$.

How to handle drift (if needed) ? E.g. multiple-shooting

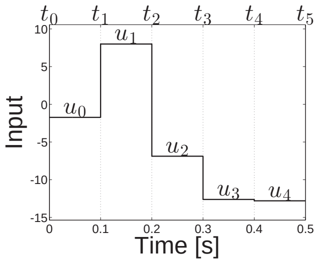
OCP:

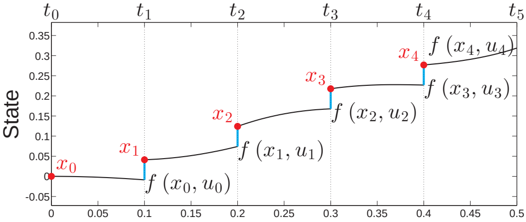
$$\begin{aligned} \min \quad & \Phi\left(\mathbf{x}(.), \mathbf{u}(.)\right) \\ \text{s.t.} \quad & \mathbf{F}\left(\dot{\mathbf{x}}(t), \mathbf{z}(t), \mathbf{x}(t), \mathbf{u}(t)\right) = 0 \\ & \mathbf{x}(t_0) = \bar{\mathbf{x}}_0 \end{aligned}$$

$\mathbf{f}\left(\mathbf{x}_k, \mathbf{u}_k\right)$ integrates the dynamics $\mathbf{F}$ over the time interval $[t_k,\ t_{k+1}]$. Label $\mathbf{C}$ the consistency conditions. Note that $\mathbf{C}\left(\bar{\mathbf{x}}_0\right) = 0$ is preserved by $\mathbf{f}$.

NLP with $\mathbf{w} = \{\mathbf{x}_0, \mathbf{u}_0, \ldots, \mathbf{x}_{N-1}, \mathbf{u}_{N-1}, \mathbf{x}_N\}$

$$\begin{aligned} \min_{\mathbf{w}} \quad & \Phi(\mathbf{w}) \\ \text{s.t.} \quad & \mathbf{g}(\mathbf{w}) = \begin{bmatrix} \bar{\mathbf{x}}_0 - \mathbf{x}_0 \\ \mathbf{f}\left(\mathbf{x}_0, \mathbf{u}_0\right) - \mathbf{x}_1 \\ \ldots \\ \mathbf{f}\left(\mathbf{x}_{N-1}, \mathbf{u}_{N-1}\right) - \mathbf{x}_N \end{bmatrix} = 0 \end{aligned}$$





How to handle drift (if needed) ? E.g. multiple-shooting

OCP:

$$\begin{aligned} \min \quad & \Phi(\mathbf{x}(.), \mathbf{u}(.)) \\ \text{s.t.} \quad & \mathbf{F}(\dot{\mathbf{x}}(t), \mathbf{z}(t), \mathbf{x}(t), \mathbf{u}(t)) = 0 \\ & \mathbf{x}(t_0) = \bar{\mathbf{x}}_0 \end{aligned}$$

$\mathbf{f}(\mathbf{x}_k, \mathbf{u}_k)$ integrates the dynamics $\mathbf{F}$ over the time interval $[t_k,\ t_{k+1}]$. Label $\mathbf{C}$ the consistency conditions. Note that $\mathbf{C}(\bar{\mathbf{x}}_0) = 0$ is preserved by $\mathbf{f}$.

NLP with $\mathbf{w} = \{\mathbf{x}_0, \mathbf{u}_0, ..., \mathbf{x}_{N-1}, \mathbf{u}_{N-1}, \mathbf{x}_N\}$

$$\begin{aligned} \min_{\mathbf{w}} \quad & \Phi(\mathbf{w}) \\ \text{s.t.} \quad & \mathbf{g}(\mathbf{w}) = \begin{bmatrix} \bar{\mathbf{x}}_0 - \mathbf{x}_0 \\ \mathbf{f}(\mathbf{x}_0, \mathbf{u}_0) - \mathbf{x}_1 \\ \dots \\ \mathbf{f}(\mathbf{x}_{N-1}, \mathbf{u}_{N-1}) - \mathbf{x}_N \end{bmatrix} = 0 \end{aligned}$$

How to handle drift (if needed) ? E.g. multiple-shooting

OCP:

$$\begin{aligned} \min \quad & \Phi\left(\mathbf{x}(.), \mathbf{u}(.)\right) \\ \text{s.t.} \quad & \mathbf{F}\left(\dot{\mathbf{x}}\left(t\right), \mathbf{z}\left(t\right), \mathbf{x}\left(t\right), \mathbf{u}\left(t\right)\right) = 0 \\ & \mathbf{x}\left(t_0\right) = \bar{\mathbf{x}}_0 \end{aligned}$$

$\mathbf{f}\left(\mathbf{x}_k, \mathbf{u}_k\right)$ integrates the dynamics $\mathbf{F}$ over the time interval $[t_k,\ t_{k+1}]$. Label $\mathbf{C}$ the consistency conditions. Note that $\mathbf{C}\left(\bar{\mathbf{x}}_0\right) = 0$ is preserved by $\mathbf{f}$.

We *would like* to **impose**:

$$\begin{aligned} \mathbf{f}\left(\mathbf{x}_k, \mathbf{u}_k\right) - \mathbf{x}_{k+1} &= 0 \\ \mathbf{C}\left(\mathbf{x}_{k+1}\right) &= 0 \end{aligned}$$

at every shooting node k, so as to control the drift. However, the problem would be over-constrained $\Rightarrow$ **LICQ deficiency** !!

NLP with $\mathbf{w} = \left\{\mathbf{x}_0, \mathbf{u}_0, ..., \mathbf{x}_{N-1}, \mathbf{u}_{N-1}, \mathbf{x}_N\right\}$

$$\begin{aligned} \min_{\mathbf{w}} \quad & \Phi\left(\mathbf{w}\right) \\ \text{s.t.} \quad & \mathbf{g}\left(\mathbf{w}\right) = \left[\begin{array}{c} \bar{\mathbf{x}}_0 - \mathbf{x}_0 \\ \mathbf{f}\left(\mathbf{x}_0, \mathbf{u}_0\right) - \mathbf{x}_1 \\ \dots \\ \mathbf{f}\left(\mathbf{x}_{N-1}, \mathbf{u}_{N-1}\right) - \mathbf{x}_N \end{array}\right] = 0 \end{aligned}$$

How to handle drift (if needed) ? E.g. multiple-shooting

OCP:

$$\begin{aligned} \min \quad & \Phi\left(\mathbf{x}(.), \mathbf{u}(.)\right) \\ \text{s.t.} \quad & \mathbf{F}\left(\dot{\mathbf{x}}\left(t\right), \mathbf{z}\left(t\right), \mathbf{x}\left(t\right), \mathbf{u}\left(t\right)\right) = 0 \\ & \mathbf{x}\left(t_0\right) = \bar{\mathbf{x}}_0 \end{aligned}$$

$\mathbf{f}\left(\mathbf{x}_k, \mathbf{u}_k\right)$ integrates the dynamics $\mathbf{F}$ over the time interval $[t_k,\ t_{k+1}]$. Label $\mathbf{C}$ the consistency conditions. Note that $\mathbf{C}\left(\bar{\mathbf{x}}_0\right) = 0$ is preserved by $\mathbf{f}$.

We *would like* to **impose**:

$$\begin{aligned} \mathbf{f}\left(\mathbf{x}_k, \mathbf{u}_k\right) - \mathbf{x}_{k+1} &= 0 \\ \mathbf{C}\left(\mathbf{x}_{k+1}\right) &= 0 \end{aligned}$$

at every shooting node k, so as to control the drift. However, the problem would be over-constrained $\Rightarrow$ **LICQ deficiency** !!

NLP with $\mathbf{w} = \left\{\mathbf{x}_0, \mathbf{u}_0, ..., \mathbf{x}_{N-1}, \mathbf{u}_{N-1}, \mathbf{x}_N\right\}$

$$\begin{aligned} \min_{\mathbf{w}} \quad & \Phi\left(\mathbf{w}\right) \\ \text{s.t.} \quad & \mathbf{g}\left(\mathbf{w}\right) = \left[\begin{array}{c} \bar{\mathbf{x}}_0 - \mathbf{x}_0 \\ \mathbf{f}\left(\mathbf{x}_0, \mathbf{u}_0\right) - \mathbf{x}_1 \\ \ldots \\ \mathbf{f}\left(\mathbf{x}_{N-1}, \mathbf{u}_{N-1}\right) - \mathbf{x}_N \end{array}\right] = 0 \end{aligned}$$

How to handle drift (if needed) ? E.g. multiple-shooting

OCP:

$$\begin{aligned} \min \quad & \Phi\left(\mathbf{x}(.), \mathbf{u}(.)\right) \\ \text{s.t.} \quad & \mathbf{F}\left(\dot{\mathbf{x}}(t), \mathbf{z}(t), \mathbf{x}(t), \mathbf{u}(t)\right) = 0 \\ & \mathbf{x}(t_0) = \bar{\mathbf{x}}_0 \end{aligned}$$

$\mathbf{f}\left(\mathbf{x}_k, \mathbf{u}_k\right)$ integrates the dynamics $\mathbf{F}$ over the time interval $[t_k,\ t_{k+1}]$. Label $\mathbf{C}$ the consistency conditions. Note that $\mathbf{C}\left(\bar{\mathbf{x}}_0\right) = 0$ is preserved by $\mathbf{f}$.

We *would like* to **impose**:

$$\begin{aligned} \mathbf{f}\left(\mathbf{x}_k, \mathbf{u}_k\right) - \mathbf{x}_{k+1} &= 0 \\ \mathbf{C}\left(\mathbf{x}_{k+1}\right) &= 0 \end{aligned}$$

at every shooting node k, so as to control the drift. However, the problem would be over-constrained $\Rightarrow$ **LICQ deficiency** !!

NLP with $\mathbf{w} = \{\mathbf{x}_0, \mathbf{u}_0, ..., \mathbf{x}_{N-1}, \mathbf{u}_{N-1}, \mathbf{x}_N\}$

$$\begin{aligned} \min_{\mathbf{w}} \quad & \Phi\left(\mathbf{w}\right) \\ \text{s.t.} \quad & \mathbf{g}\left(\mathbf{w}\right) = \begin{bmatrix} \bar{\mathbf{x}}_0 - \mathbf{x}_0 \\ \mathbf{f}\left(\mathbf{x}_0, \mathbf{u}_0\right) - \mathbf{x}_1 \\ \cdots \\ \mathbf{f}\left(\mathbf{x}_{N-1}, \mathbf{u}_{N-1}\right) - \mathbf{x}_N \end{bmatrix} = 0 \end{aligned}$$

Why LICQ deficiency ?? Consider one interval:

$$\mathbf{C}\left(\mathbf{f}\left(\mathbf{x}_k, \mathbf{u}_k\right)\right) = 0, \quad \forall \mathbf{u}_k$$

holds (mathematically)

How to handle drift (if needed) ? E.g. multiple-shooting

OCP:

$$\begin{aligned} \min \quad & \Phi\left(\mathbf{x}(.), \mathbf{u}(.)\right) \\ \text{s.t.} \quad & \mathbf{F}\left(\dot{\mathbf{x}}(t), \mathbf{z}(t), \mathbf{x}(t), \mathbf{u}(t)\right) = 0 \\ & \mathbf{x}(t_0) = \bar{\mathbf{x}}_0 \end{aligned}$$

$\mathbf{f}\left(\mathbf{x}_k, \mathbf{u}_k\right)$ integrates the dynamics $\mathbf{F}$ over the time interval $[t_k,\ t_{k+1}]$. Label $\mathbf{C}$ the consistency conditions. Note that $\mathbf{C}\left(\bar{\mathbf{x}}_0\right) = 0$ is preserved by $\mathbf{f}$.

We *would like* to **impose**:

$$\begin{aligned} \mathbf{f}\left(\mathbf{x}_k, \mathbf{u}_k\right) - \mathbf{x}_{k+1} &= 0 \\ \mathbf{C}\left(\mathbf{x}_{k+1}\right) &= 0 \end{aligned}$$

at every shooting node k, so as to control the drift. However, the problem would be over-constrained $\Rightarrow$ **LICQ deficiency** !!

NLP with $\mathbf{w} = \left\{\mathbf{x}_0, \mathbf{u}_0, ..., \mathbf{x}_{N-1}, \mathbf{u}_{N-1}, \mathbf{x}_N\right\}$

$$\begin{aligned} \min_{\mathbf{w}} \quad & \Phi\left(\mathbf{w}\right) \\ \text{s.t.} \quad & \mathbf{g}\left(\mathbf{w}\right) = \left[\begin{array}{c} \bar{\mathbf{x}}_0 - \mathbf{x}_0 \\ \mathbf{f}\left(\mathbf{x}_0, \mathbf{u}_0\right) - \mathbf{x}_1 \\ \cdots \\ \mathbf{f}\left(\mathbf{x}_{N-1}, \mathbf{u}_{N-1}\right) - \mathbf{x}_N \end{array}\right] = 0 \end{aligned}$$

Why LICQ deficiency ?? Consider one interval:

$$\mathbf{C}\left(\mathbf{f}\left(\mathbf{x}_k, \mathbf{u}_k\right)\right) = 0, \quad \forall \mathbf{u}_k$$

holds (mathematically), such that:

$$\nabla_{\mathbf{u}_k}\mathbf{f}\left(\mathbf{x}_k, \mathbf{u}_k\right)\nabla\mathbf{C}\left(\mathbf{x}_{k+1}\right) = 0$$

holds at the solution.

How to handle drift (if needed) ? E.g. multiple-shooting

OCP:

$$\begin{aligned} \min \quad & \Phi\left(\mathbf{x}(.), \mathbf{u}(.)\right) \\ \text{s.t.} \quad & \mathbf{F}\left(\dot{\mathbf{x}}(t), \mathbf{z}(t), \mathbf{x}(t), \mathbf{u}(t)\right) = 0 \\ & \mathbf{x}(t_0) = \bar{\mathbf{x}}_0 \end{aligned}$$

$\mathbf{f}\left(\mathbf{x}_k, \mathbf{u}_k\right)$ integrates the dynamics $\mathbf{F}$ over the time interval $[t_k,\ t_{k+1}]$. Label $\mathbf{C}$ the consistency conditions. Note that $\mathbf{C}\left(\bar{\mathbf{x}}_0\right) = 0$ is preserved by $\mathbf{f}$.

We *would like* to **impose**:

$$\begin{aligned} \mathbf{f}\left(\mathbf{x}_k, \mathbf{u}_k\right) - \mathbf{x}_{k+1} &= 0 \\ \mathbf{C}\left(\mathbf{x}_{k+1}\right) &= 0 \end{aligned}$$

at every shooting node k, so as to control the drift. However, the problem would be over-constrained $\Rightarrow$ **LICQ deficiency** !!

NLP with $\mathbf{w} = \left\{\mathbf{x}_0, \mathbf{u}_0, \ldots, \mathbf{x}_{N-1}, \mathbf{u}_{N-1}, \mathbf{x}_N\right\}$

$$\begin{aligned} \min_{\mathbf{w}} \quad & \Phi\left(\mathbf{w}\right) \\ \text{s.t.} \quad & \mathbf{g}\left(\mathbf{w}\right) = \left[\begin{array}{c} \bar{\mathbf{x}}_0 - \mathbf{x}_0 \\ \mathbf{f}\left(\mathbf{x}_0, \mathbf{u}_0\right) - \mathbf{x}_1 \\ \ldots \\ \mathbf{f}\left(\mathbf{x}_{N-1}, \mathbf{u}_{N-1}\right) - \mathbf{x}_N \end{array}\right] = 0 \end{aligned}$$

Then :

$$\nabla_{\mathbf{x}_{k+1}, \mathbf{u}_k} \left[\begin{array}{c} \mathbf{f}_k - \mathbf{x}_{k+1} \\ \mathbf{C}\left(\mathbf{x}_{k+1}\right) \end{array}\right] = \left[\begin{array}{cc} -I & \nabla \mathbf{C} \\ \nabla_{\mathbf{u}_k} \mathbf{f}_k & 0 \end{array}\right]$$

Result in:

$$\left[\begin{array}{cc} -I & \nabla \mathbf{C} \\ \nabla_{\mathbf{u}_k} \mathbf{f}_k & 0 \end{array}\right] \left[\begin{array}{c} \nabla \mathbf{C} \\ I \end{array}\right] = 0$$

i.e. LICQ fails !!

How to handle drift (if needed) ? E.g. multiple-shooting

OCP:

$$\begin{aligned} \min \quad & \Phi\left(\mathbf{x}(.), \mathbf{u}(.)\right) \\ \text{s.t.} \quad & \mathbf{F}\left(\dot{\mathbf{x}}\left(t\right), \mathbf{z}\left(t\right), \mathbf{x}\left(t\right), \mathbf{u}\left(t\right)\right) = 0 \\ & \mathbf{x}\left(t_0\right) = \bar{\mathbf{x}}_0 \end{aligned}$$

$\mathbf{f}\left(\mathbf{x}_k, \mathbf{u}_k\right)$ integrates the dynamics $\mathbf{F}$ over the time interval $[t_k,\ t_{k+1}]$. Label $\mathbf{C}$ the consistency conditions. Note that $\mathbf{C}\left(\bar{\mathbf{x}}_0\right) = 0$ is preserved by $\mathbf{f}$.

Idea: **project** the continuity conditions **in the null space** of the consistency conditions, i.e.:

$$\begin{aligned} Z_k^\top \left(\mathbf{f}\left(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}\right) - \mathbf{x}_k\right) = 0 \\ \mathbf{C}\left(\mathbf{x}_k\right) = 0 \end{aligned}$$

where Z_k is a basis of the "left-hand" null-space of $\nabla \mathbf{C}\left(\mathbf{x}_k\right)$:

$$Z_k^\top \nabla \mathbf{C}\left(\mathbf{x}_k\right) = 0$$

NLP with $\mathbf{w} = \left\{\mathbf{x}_0, \mathbf{u}_0, ..., \mathbf{x}_{N-1}, \mathbf{u}_{N-1}, \mathbf{x}_N\right\}$

$$\begin{aligned} \min_{\mathbf{w}} \quad & \Phi\left(\mathbf{w}\right) \\ \text{s.t.} \quad & \mathbf{g}\left(\mathbf{w}\right) = \begin{bmatrix} \bar{\mathbf{x}}_0 - \mathbf{x}_0 \\ \mathbf{f}\left(\mathbf{x}_0, \mathbf{u}_0\right) - \mathbf{x}_1 \\ \dots \\ \mathbf{f}\left(\mathbf{x}_{N-1}, \mathbf{u}_{N-1}\right) - \mathbf{x}_N \end{bmatrix} = 0 \end{aligned}$$

How to handle drift (if needed) ? E.g. multiple-shooting

OCP:

$$\begin{aligned} \min \quad & \Phi\left(\mathbf{x}(.), \mathbf{u}(.)\right) \\ \text{s.t.} \quad & \mathbf{F}\left(\dot{\mathbf{x}}\left(t\right), \mathbf{z}\left(t\right), \mathbf{x}\left(t\right), \mathbf{u}\left(t\right)\right) = 0 \\ & \mathbf{x}\left(t_0\right) = \bar{\mathbf{x}}_0 \end{aligned}$$

$\mathbf{f}\left(\mathbf{x}_k, \mathbf{u}_k\right)$ integrates the dynamics $\mathbf{F}$ over the time interval $[t_k,\ t_{k+1}]$. Label $\mathbf{C}$ the consistency conditions. Note that $\mathbf{C}\left(\bar{\mathbf{x}}_0\right) = 0$ is preserved by $\mathbf{f}$.

Idea: **project** the continuity conditions **in the null space** of the consistency conditions, i.e.:

$$\begin{aligned} Z_k^\top \left(\mathbf{f}\left(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}\right) - \mathbf{x}_k\right) = 0 \\ \mathbf{C}\left(\mathbf{x}_k\right) = 0 \end{aligned}$$

where Z_k is a basis of the "left-hand" null-space of $\nabla \mathbf{C}\left(\mathbf{x}_k\right)$:

$$Z_k^\top \nabla \mathbf{C}\left(\mathbf{x}_k\right) = 0$$

NLP with $\mathbf{w} = \left\{\mathbf{x}_0, \mathbf{u}_0, ..., \mathbf{x}_{N-1}, \mathbf{u}_{N-1}, \mathbf{x}_N\right\}$

$$\begin{aligned} \min_{\mathbf{w}} \quad & \Phi\left(\mathbf{w}\right) \\ \text{s.t.} \quad & \begin{bmatrix} \bar{\mathbf{x}}_0 - \mathbf{x}_0 \\ Z_1^\top \left(\mathbf{f}\left(\mathbf{x}_0, \mathbf{u}_0\right) - \mathbf{x}_1\right) \\ \mathbf{C}\left(\mathbf{x}_1\right) \\ \dots \\ Z_{N-1}^\top \left(\mathbf{f}\left(\mathbf{x}_{N-1}, \mathbf{u}_{N-1}\right) - \mathbf{x}_N\right) \\ \mathbf{C}\left(\mathbf{x}_{N-1}\right) \end{bmatrix} = 0 \end{aligned}$$

How to handle drift (if needed) ? E.g. multiple-shooting

OCP:

$$\begin{aligned} \min \quad & \Phi(\mathbf{x}(.), \mathbf{u}(.)) \\ \text{s.t.} \quad & \mathbf{F}(\dot{\mathbf{x}}(t), \mathbf{z}(t), \mathbf{x}(t), \mathbf{u}(t)) = 0 \\ & \mathbf{x}(t_0) = \bar{\mathbf{x}}_0 \end{aligned}$$

$\mathbf{f}(\mathbf{x}_k, \mathbf{u}_k)$ integrates the dynamics $\mathbf{F}$ over the time interval $[t_k,\ t_{k+1}]$. Label $\mathbf{C}$ the consistency conditions. Note that $\mathbf{C}(\bar{\mathbf{x}}_0) = 0$ is preserved by $\mathbf{f}$.

NLP with $\mathbf{w} = \{\mathbf{x}_0, \mathbf{u}_0, ..., \mathbf{x}_{N-1}, \mathbf{u}_{N-1}, \mathbf{x}_N\}$

$$\min_{\mathbf{w}} \quad \Phi(\mathbf{w})$$

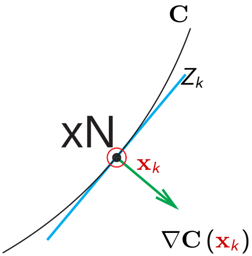
$$\text{s.t.} \quad \begin{bmatrix} \bar{\mathbf{x}}_0 - \mathbf{x}_0 \\ Z_1^\top(\mathbf{f}(\mathbf{x}_0, \mathbf{u}_0) - \mathbf{x}_1) \\ \mathbf{C}(\mathbf{x}_1) \\ \ldots \\ Z_{N-1}^\top(\mathbf{f}(\mathbf{x}_{N-1}, \mathbf{u}_{N-1}) - \mathbf{x}_N) \\ \mathbf{C}(\mathbf{x}_{N-1}) \end{bmatrix} = 0$$

Idea: **project** the continuity conditions **in the null space** of the consistency conditions, i.e.:

$$\begin{aligned} Z_k^\top(\mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}) - \mathbf{x}_k) &= 0 \\ \mathbf{C}(\mathbf{x}_k) &= 0 \end{aligned}$$

where Z_k is a basis of the "left-hand" null-space of $\nabla\mathbf{C}(\mathbf{x}_k)$:

$$Z_k^\top \nabla\mathbf{C}(\mathbf{x}_k) = 0$$



How to handle drift (if needed) ? E.g. multiple-shooting

OCP:

$$\begin{aligned} \min \quad & \Phi\left(\mathbf{x}(.), \mathbf{u}(.)\right) \\ \text{s.t.} \quad & \mathbf{F}\left(\dot{\mathbf{x}}(t), \mathbf{z}(t), \mathbf{x}(t), \mathbf{u}(t)\right) = 0 \\ & \mathbf{x}(t_0) = \bar{\mathbf{x}}_0 \end{aligned}$$

$\mathbf{f}\left(\mathbf{x}_k, \mathbf{u}_k\right)$ integrates the dynamics $\mathbf{F}$ over the time interval $[t_k,\ t_{k+1}]$. Label $\mathbf{C}$ the consistency conditions. Note that $\mathbf{C}\left(\bar{\mathbf{x}}_0\right) = 0$ is preserved by $\mathbf{f}$.

Idea: **project** the continuity conditions **in the null space** of the consistency conditions, i.e.:

$$\begin{aligned} Z_k^\top\left(\mathbf{f}\left(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}\right) - \mathbf{x}_k\right) &= 0 \\ \mathbf{C}\left(\mathbf{x}_k\right) &= 0 \end{aligned}$$

where Z_k is a basis of the "left-hand" null-space of $\nabla\mathbf{C}\left(\mathbf{x}_k\right)$:

$$Z_k^\top \nabla\mathbf{C}\left(\mathbf{x}_k\right) = 0$$

NLP with $\mathbf{w} = \left\{\mathbf{x}_0, \mathbf{u}_0, ..., \mathbf{x}_{N-1}, \mathbf{u}_{N-1}, \mathbf{x}_N\right\}$

$$\begin{aligned} \min_{\mathbf{w}} \quad & \Phi\left(\mathbf{w}\right) \\ \text{s.t.} \quad & \begin{bmatrix} \bar{\mathbf{x}}_0 - \mathbf{x}_0 \\ Z_1^\top\left(\mathbf{f}\left(\mathbf{x}_0, \mathbf{u}_0\right) - \mathbf{x}_1\right) \\ \mathbf{C}\left(\mathbf{x}_1\right) \\ \ldots \\ Z_{N-1}^\top\left(\mathbf{f}\left(\mathbf{x}_{N-1}, \mathbf{u}_{N-1}\right) - \mathbf{x}_N\right) \\ \mathbf{C}\left(\mathbf{x}_{N-1}\right) \end{bmatrix} = 0 \end{aligned}$$

Observe that $Z_k = Z_k\left(\mathbf{x}_k\right)$!! Can be difficult to deploy if the Z_k cannot be computed explicitly. Then they have to be introduced as decision variables in the NLP, and computed implicitly. That yields a very large and often tricky NLP (we will get back to this soon !)