# Optimal control using nonlinear programming

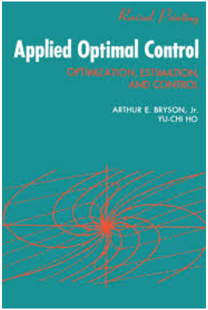
*From smooth to non-smooth optimal control*

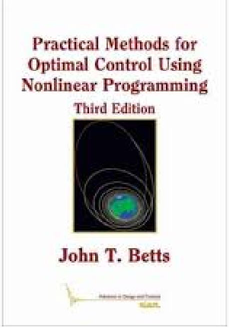
François Pacaud
frapac.github.io

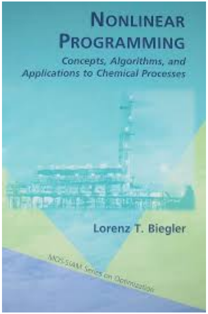
CAS, Mines Paris - PSL

**Séminaire McTAO**
June 17 2026

# What do I mean by optimal control?




# Outline

Smooth optimal control

Nonsmooth optimal control

# Introducing the notations

Notations:

- $x \in \mathbb{R}^{n_x}$: state (dependent variables); $u \in \mathbb{R}^{n_u}$: control (degrees of freedom);
- $g(\cdot)$: smooth dynamics

## Discretize-then-optimize

The continuous dynamics

$$x(0) = x_{init} \ , \quad \dot{x}(t) = g(x(t), u(t)) \quad \forall t > 0$$

is discretized as

$$x_0 = x_{init} \ , \quad g_k(x_{k+1}, x_k, u_k, \Delta) = 0 \quad \forall k$$

with $x_k := x(\Delta k)$, $\Delta$ given timestep

The function $g_k$ encodes the discretization scheme, e.g.,

- *Implicit Euler*: $g_k(x_{k+1}, x_k, u_k, \Delta) = x_{k+1} - x_k - \Delta\, g(x_{k+1}, u_k)$
- *Trapezoidal rule*: $g_k(x_{k+1}, x_k, u_k, \Delta) = x_{k+1} - x_k - \frac{1}{2}\Delta \left(g(x_k, u_k) + g(x_{k+1}, u_k)\right)$

## Questions

- How to use formulate optimal control problems as nonlinear programs?
- Why is nonsmooth optimal control useful?

# Writing down the optimal control problem in discrete time

In the previous slide, we have explicited the dynamics. Introduce now, for all $k$:

- Operational constraints on state and control: $h_k(x_k, u_k) \geq 0$
- Cost functional $\ell_k(x_k, u_k)$

We consider a **finite** horizon $N$

## Optimal control problem

Starting from $x_{init}$, we solve:

$$\begin{aligned} \min_{x,u(,\Delta)} \quad & \sum_{k=0}^{N-1} \ell_k(x_k, u_k) + \ell_N(x_N) \\ \text{s.t. } & x_0 = x_{init} \\ & g_k(x_{k+1}, x_k, u_k, \Delta) = 0 \qquad \forall k = 0, \cdots, N-1 \\ & h_k(x_k, u_k) \geq 0 \qquad \forall k = 0, \cdots, N-1 \end{aligned}$$

- **Fixed-time optimal control:** $\Delta$ is given parameter
- **Free-time optimal control:** $\Delta$ is a degree-of-freedom

Let $x := (x_0, \cdots, x_N)$ and $u := (u_0, \cdots, u_{N-1})$.

# What do I mean by nonlinear programming?

We abstract the previous optimal control problem as a nonlinear program

Let $f : \mathbb{R}^n \to \mathbb{R}$, $g : \mathbb{R}^n \to \mathbb{R}^{m_E}$, $h : \mathbb{R}^n \to \mathbb{R}^{m_I}$.

**Nonlinear program**

$$\min_{w} f(w)$$
$$\text{subject to } g(w) = 0\ ,\ \ h(w) \geq 0$$

For optimal control: $w := (x, u)$

For multipliers $(y, z) \in \mathbb{R}^{m_E} \times \mathbb{R}^{m_I}$, define the Lagrangian :

$$L(w, y, z) = f(w) + y^\top g(w) + z^\top h(w)$$

Karush-Kuhn-Tucker (KKT) conditions (necessary but not sufficient):

$$\begin{aligned} &\nabla f(w) + \nabla g(w) y + \nabla h(w) z = 0 \\ &g(w) = 0 \\ &0 \leq z \perp -h(w) \geq 0 \end{aligned}$$

# Regularity conditions: first-order

Denote the active set $\mathcal{A}(w) = \{i \in [m_I] \mid h_i(w) = 0\}$ and the active Jacobian:

$$J(w) = \begin{bmatrix} \nabla g(w)^\top \\ \nabla h_{\mathcal{A}}(w)^\top \end{bmatrix}$$

## First-order constraint qualification

We say that the point $w$ is *qualified* is the local geometry of the feasible can be captured by a *linearized* model near $w$

*(in math language, the tangent cone is equal to the set of linearized feasible directions)*

- The *Linear Independence Constraint Qualification* (LICQ) holds if the active Jacobian $J(w)$ is full row-rank
- The *Mangasarian-Fromovitz Constraint Qualification* (MFCQ) holds if $(y, z) = (0, 0)$ is the unique solution to the linear system

$$\begin{cases} \nabla g(w)^\top y + \nabla h(w)^\top z = 0 \\ z_i \geq 0 & \forall i \in \mathcal{A}(w) \end{cases}$$

J. Nocedal, SJ. Wright. Numerical optimization.

## Regularity conditions: second-order

Let $(w, y, z)$ a primal-dual point satisfying KKT. We define the *critical cone*:

$$\begin{aligned}\mathcal{C}(w, y, z) = \{d \in \mathbb{R}^n \mid \nabla g_i(w)^\top d &= 0 \ , \ i \in [m_e] \ , \\ \nabla h_i(w)^\top d &= 0 \ , \ i \in \mathcal{A}(w) \text{ with } z_i > 0 \ , \\ \nabla h_i(w)^\top d &\leq 0 \ , \ i \in \mathcal{A}(w) \text{ with } z_i = 0\} \ .\end{aligned}$$

*Strict complementarity* (SC) holds if $z_i > 0$ for all $i \in \mathcal{A}(w)$.

### Second-order sufficient condition (SOSC)

We say that $(w, y, z)$ satisfies SOSC if

$$d^\top \nabla^2_{ww} \mathcal{L}(w, y, z) d > 0 \quad \forall d \in \mathcal{C}(w, y, z) \setminus \{0\}$$

Under SOSC, the problem is **locally convex** near $w$ and the solution is **isolated**

### Proposition

Suppose LICQ and SC hold. SOSC holds if and only if

$$Z^\top \nabla^2_{ww} L(w, y, z) Z \succ 0$$

with $Z$ a basis of the null-space of the active Jacobian $J(w)$.

# Are optimal control problems regular?

Well... it depends

### Problem with state constraints

If an operational constraint $h_k(x_k, u_k) \leq 0$ depends on $x_k$ and is **active**,
the LICQ fails (not enough degree of freedom).

### Problem with singular arcs

If the optimal control problem has a singular arc, the SOSC fails.

Most of the time, optimal control problems transcribe as difficult nonlinear programs.

# Primal-dual interior-point method

Rewrite the (nonsmooth) KKT system as a *smooth* nonlinear system, for **positive** $(s, z) > 0$:

$$F_\mu(x, s, \overbrace{y, z}^{\text{Dual variables}}) := \begin{bmatrix} \nabla f(x) + \nabla g(x)\, y + \nabla h(x)\, z \\ g(x) \\ h(x) + s \\ \underbrace{Sz - \mu e}_{\text{Homotopy, } S = \operatorname{diag}(s)} \end{bmatrix} = 0$$

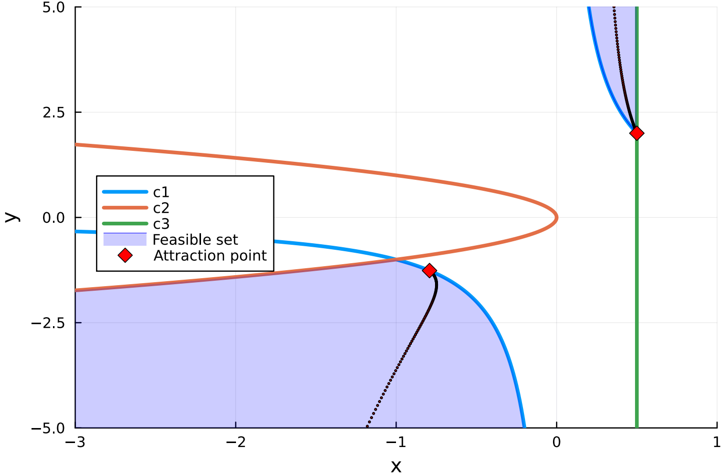
→ often referred to as the *central path equations*

## Primal-dual interior-point method

Solve $F_\mu(x, s, y, z) = 0$ using Newton method while driving $\mu \to 0$.

# Central path: illustration on HS15

Plotted with CentralPath.jl




## Caveats

- Multiple solutions, non connex feasible set
- Non-unicity of the central path at a given $\mu$

# Augmented KKT system

The Newton algorithm translates to the solution of a sequence of linear systems

Denote the primal-dual variable by $v^k := (w^k, s^k, y^k, z^k)$. At iteration $k$:

1. Compute Newton step $d^k$ as solution of the linear system

$$\boxed{\nabla F_\mu(v^k)}\; d^k = -F_\mu(v^k)$$

2. Update $v^k$ as

$$v^{k+1} = v^k + \alpha^k d^k$$

with $\alpha^k$ computed using a line-search algorithm

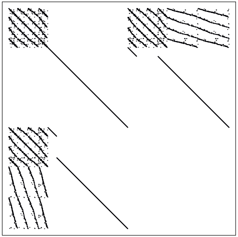


Figure: $\nabla F_\mu$

**Augmented KKT system**

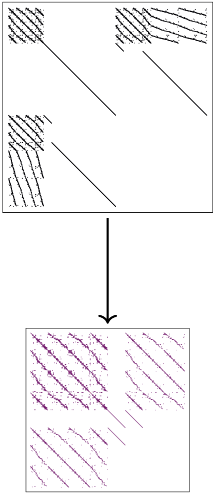
After (slight) reformulation, the Newton step writes as

$$\boxed{\begin{bmatrix} W & 0 & G^\top & H^\top \\ 0 & \Sigma_s & 0 & I \\ G & 0 & 0 & 0 \\ H & I & 0 & 0 \end{bmatrix}} \begin{bmatrix} d_w \\ d_s \\ d_y \\ d_z \end{bmatrix} = - \begin{bmatrix} r_1 \\ r_2 \\ r_3 \\ r_4 \end{bmatrix}$$

with $W = \nabla^2_{xx} L(\cdot)$, $\Sigma_s = S^{-1}\mathrm{diag}(z)$

# Condensed KKT system

We condense the linear system by removing the inequality blocks



## Condensed KKT system

The augmented KKT system condenses to

$$\begin{bmatrix} K & G^\top \\ G & 0 \end{bmatrix} \begin{bmatrix} d_w \\ d_y \end{bmatrix} = - \begin{bmatrix} r_1 + H^\top (\Sigma_s r_4 + r_2) \\ r_3 \end{bmatrix}$$

with the *condensed matrix* $K = W + H^\top \Sigma_s H$.

We recover $(d_s, d_z)$ as

$$d_s = -\Sigma_s^{-1}(r_3 + d_y) \, , \quad d_z = \Sigma_s \left( H \, d_x - r_4 \right) - r_2 \; .$$

# Condensed KKT systems in optimal control

## Proposition

For $K$ positive definite, the solution of the saddle-point linear system

$$\begin{bmatrix} K & G^\top \\ G & 0 \end{bmatrix} \begin{bmatrix} w \\ y \end{bmatrix} = \begin{bmatrix} -c \\ b \end{bmatrix}$$

is the primal-dual solution of the convex QP

$$\min_w \ \frac{1}{2} w^\top K w + c^\top w \quad \text{s.t.} \quad Gw = b$$

## Corollary

For problem with a dynamic structure, the condensed KKT system yields the LQR

$$\min_{d_x, d_u} \ \sum_{k=0}^{N-1} \begin{bmatrix} d_{x,k} \\ d_{u,k} \end{bmatrix}^\top \begin{bmatrix} K_{xx}^k & K_{xu}^k \\ K_{ux}^k & K_{uu}^k \end{bmatrix} \begin{bmatrix} d_{x,k} \\ d_{u,k} \end{bmatrix} + \begin{bmatrix} r_{x,k} \\ r_{u,k} \end{bmatrix}^\top \begin{bmatrix} d_{x,k} \\ d_{u,k} \end{bmatrix}$$

$$\text{s.t.} \quad d_{x,k+1} = G_{x,k} d_{x,k} + G_{u,k} d_{u,k}$$

- Efficient solution with (backward) Riccati recursions
- Require convexification in nonlinear programming

JC. Dunn, D. Bertsekas. "Efficient dynamic programming implementations of Newton's method for unconstrained optimal control problems." 1989
R. Verschueren et al. "A sparsity preserving convexification procedure for indefinite quadratic programs arising in direct optimal control." 2017

# Remember that the IPM solution is always inaccurate!

IPM is NOT an active-set method

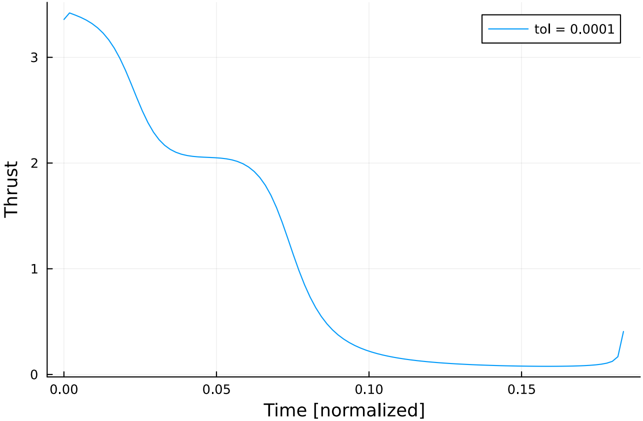



Figure: Optimal solution of the Goddard rocket problem with $nh = 100$

# Remember that the IPM solution is always inaccurate!

IPM is NOT an active-set method

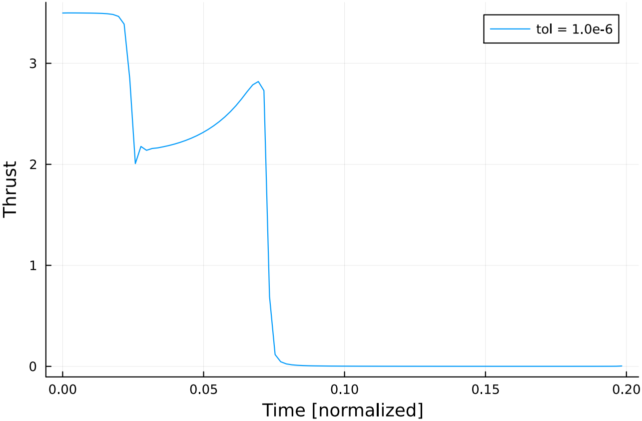



Figure: Optimal solution of the Goddard rocket problem with $nh = 100$

# Remember that the IPM solution is always inaccurate!

IPM is NOT an active-set method

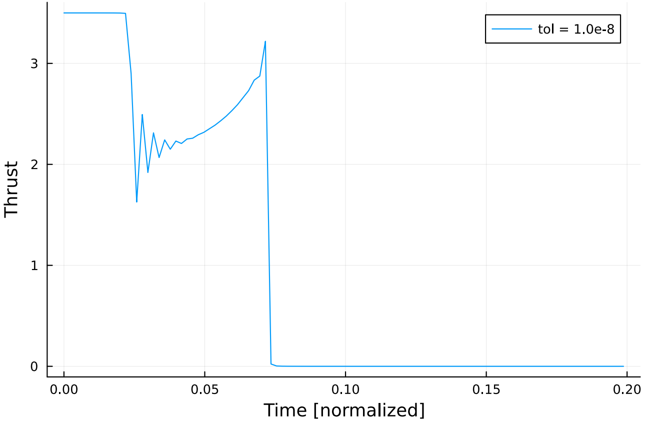



Figure: Optimal solution of the Goddard rocket problem with $nh = 100$

# Remember that the IPM solution is always inaccurate!

IPM is NOT an active-set method

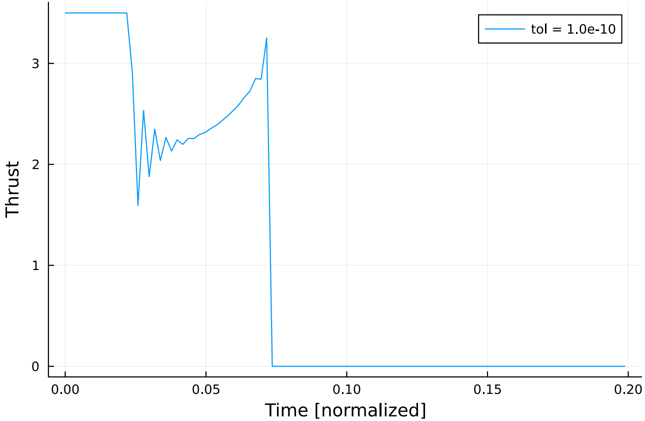



Figure: Optimal solution of the Goddard rocket problem with $nh = 100$

# Remember that the IPM solution is always inaccurate!

IPM is NOT an active-set method

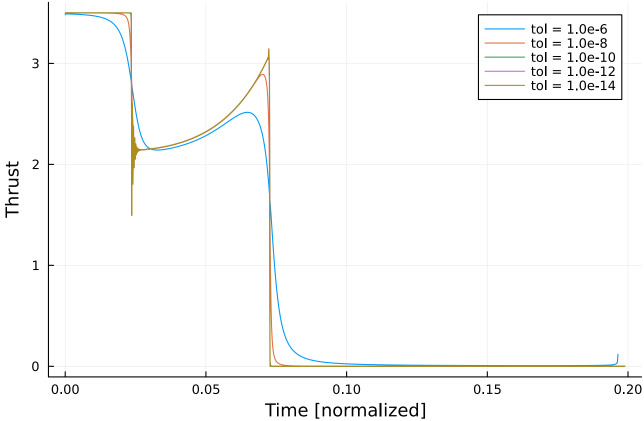



Figure: Optimal solution of the Goddard rocket problem with $nh = 1000$

# Outline

Smooth optimal control

Nonsmooth optimal control

# Nonsmooth systems

Following Armin Nurkanović's classification

Three types of nonsmoothness:

1. **Non-differentiable RHS** (e.g. activation function)

$$\dot{x} = 1 + |x|$$

2. **Discontinuous RHS** (e.g. friction)

$$\begin{cases} \dot{x} = v \\ \dot{v} = a\, t(\theta) + F\, n(\theta) \\ F \in -\mu N \operatorname{sgn}(n(\theta)^\top v) \end{cases}$$

3. **State-dependent jump** (e.g. rigid bodies with impacts and friction)

$$\begin{cases} \dot{q} = v \\ M(q)\dot{v} = f_v(q, v) + J_n(q)\lambda_n \\ 0 \leq \lambda_n \perp f_c(q) \geq 0 \end{cases}$$

**Important notice**

Nonsmooth systems are not hybrid automata!

# Example: piecewise smooth systems (PCS)

Type 2 (**discontinuous RHS**), piecewise smooth on disjoint open regions $R_i \subset \mathbb{R}^{n_x}$.

## Nonsmooth dynamics

For $n_f$ distinct dynamics $f_1, \cdots, f_{n_f}$,

$$\dot{x} = f_i(x, u) \quad \text{if} \quad x \in R_i$$

**Stewart's representation:** assume the set $R_i$ is described as

$$R_i = \{x \in \mathbb{R}^{n_x} \mid g_i(x) < \min_{j \neq i} \; g_j(x)\}$$

## Filippov's convexification

Model the PCS with the differential inclusion:

$$\dot{x} \in F_F(x, u) = \left\{ \sum_{i=1}^{n_f} f_i(x, u)\theta_i \mid \sum_{i=1}^{n_f} \theta_i = 1, \; \theta \geq 0 \right\}$$

such that $\theta_i = 0$ if $x \notin R_i \cup \partial R_i$

# Dynamic complementarity systems

Let $F(x,u) = (f_1(x,u), \cdots, f_{n_f}(x,u))$ and $g(x) = (g_1(x), \cdots, g_{n_f}(x))$

## Linear programming representation ala Stewart

$$\dot{x} = F(x,u)\,\theta \quad \text{with} \quad \theta \in \operatorname{argmin}\; g(x)^\top \tilde{\theta} \quad \text{s.t.}\; \tilde{\theta} \geq 0\,,\; e^\top \tilde{\theta} = 1$$

**Idea:** Replace LP by its stationarity condition!

## Dynamic complementarity system (DCS)

$$\begin{aligned}
&\dot{x} = F(x,u)\theta \\
&g(x) - ye - z = 0 \\
&e^\top \theta = 1 \\
&0 \leq \theta \perp z \geq 0
\end{aligned}$$

**Important:** You can reformulate different nonsmooth systems as a DCS

## Observation

Here, we treat nonsmooth system with DCS formalism

# Mathematical program with complementarity constraints (MPCC)

## MPCC in standard form

Isolate the complementarity contributions with two variables $(w_1, w_2)$:

$$\begin{aligned} \min_{w} \ & f(w) \\ \text{subject to } & c(w) = 0 \\ & 0 \leq w_1 \perp w_2 \geq 0 \end{aligned}$$

with $w = (w_0, w_1, w_2) \in \mathbb{R}^{n-2p} \times \mathbb{R}^p \times \mathbb{R}^p$ an appropriate splitting

# Scholtes relaxation method

Observe that the MPCC is equivalent to the (degenerate) nonlinear program

$$
\begin{aligned}
\min_{w \in \mathbb{R}^n} \quad & f(w) \\
\text{subject to} \quad & c(w) = 0 \ , \\
& (w_1, w_2) \geq 0 \\
& w_{1i} w_{2i} \leq 0 \quad \forall i = 1, \cdots, p
\end{aligned}
$$

- Smooth, but it violates MFCQ at all feasible points!
- Multipliers are unbounded: failure in optimization solvers

## Scholtes relaxation (homotopy method)

For $\tau > 0$, solve

$$
\begin{aligned}
\min_{x \in \mathbb{R}^n} \quad & f(w) \\
\text{subject to} \quad & c(w) = 0 \ , \\
& (w_1, w_2) \geq 0 \\
& w_{1i} w_{2i} \leq \underset{\text{Homotopy}}{\boxed{\tau}} \quad \forall i = 1, \cdots, p
\end{aligned}
$$

- If MPCC-MFCQ holds, we recover a solution for MPCC as $\tau \to 0$

Scholtes. Convergence Properties of a Regularization Scheme for Mathematical Programs with Complementarity Constraints. 2001.

# CCOpt: a new solver supporting complementarity constraints

Joint work with Anton Pozharskiy and Armin Nurkanović!

## Core idea

Adapt $\tau$ proportionally to the barrier parameter $\mu$

Plus other refinements:

- Adaptive update for homotopy parameter $\tau$
- Targetted regularization of the KKT linear system
- Adapted lower bound relaxation in the final iterations

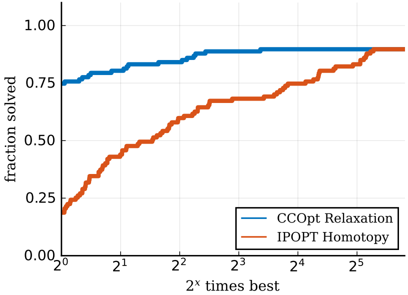



Figure: Performance profile on NOSBENCH benchmark



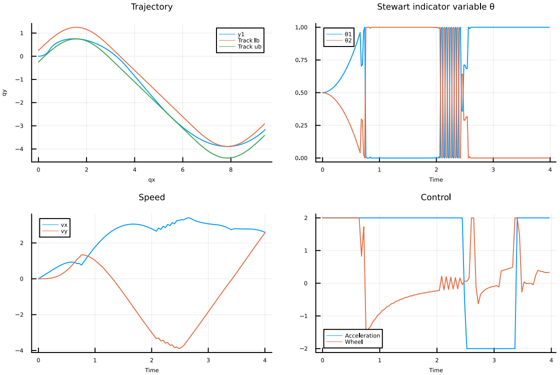
Pozharskiy A et al. COpt: an Open-Source Solver for Large-Scale Mathematical Programs with Complementarity Constraints.

# Results on the Schumacher problem

## Problem description

- Find minimum time to drive through a given circuit ;
- Non-smooth dynamics, with $(t(\theta), n(\theta))$ tangent and normal vectors:

$$\begin{cases} \dot{x} = v \\ \dot{v} = a\, t(\theta) + F\, n(\theta) \\ F \in -\mu N \,\mathrm{sgn}(n(\theta)^\top v) \end{cases}$$



## Still ongoing developments

### Research angle / optimization

How to improve the performance in CCOpt?

### Research angle / optimal control

How to handle properly state constraints and singular arcs
with a Dynamic Complementarity System (DCS)?

Don't hesitate to give it a try!
https://github.com/MadNLP/CCOpt.jl