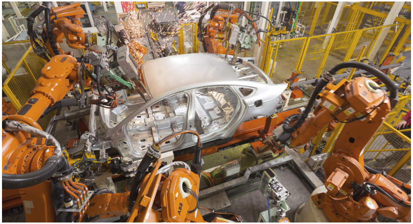
A stabilizing NMPC scheme for time-optimal point-to-point motions

Robin Verschueren, Joachim Ferreau, Alessandro Zanarini,
Mehmet Merçangöz and Moritz Diehl

Systems Control and Optimization laboratory

Time-optimal control



Time-optimal control formulation

Go from A to B in the shortest time

Time-optimal control formulation

Go from A to B in the shortest time

$$
\begin{aligned}
\underset{x(\cdot),\,u(\cdot)}{\text{minimize}} \quad & \int_{t=0}^{T} 1 \,\mathrm{d}t \\
\text{subject to} \quad & \frac{\mathrm{d}x(t)}{\mathrm{d}t} = f(x(t), u(t)), \quad && \forall t \in [0, T], \\
& x(0) = \mathrm{A}, \\
& x(T) = \mathrm{B}, \\
& c(x(t), u(t)) \leq 0, && \forall t \in [0, T].
\end{aligned}
$$

Time-optimal control formulation

Go from A to B in the shortest time

$$
\begin{aligned}
\underset{x(\cdot),\, u(\cdot)}{\text{minimize}} \quad & \int_{t=0}^{T} 1 \,\mathrm{d}t \\
\text{subject to} \quad & \frac{\mathrm{d}x(t)}{\mathrm{d}t} = f(x(t), u(t)), \quad \forall t \in [0, T], \\
& x(0) = \mathrm{A}, \\
& x(T) = \mathrm{B}, \\
& c(x(t), u(t)) \leq 0, \qquad\quad \forall t \in [0, T].
\end{aligned}
$$

Standard trick for time-optimal problems

Introduce a ‘pseudo-time’ $\tau := t/T$.

Standard trick for time-optimal problems

Introduce a ‘pseudo-time’ $\tau := t/T$.

$$
\begin{aligned}
\underset{x(\cdot),\,u(\cdot)}{\text{minimize}} \quad & \int_{\tau=0}^{1} T \,\mathrm{d}t \\
\text{subject to} \quad & \frac{\mathrm{d}x(\tau)}{\mathrm{d}\tau} = f(x(\tau), u(\tau)) \cdot T, \quad && \forall \tau \in [0,1], \\
& x(0) = \mathrm{A}, \\
& x(1) = \mathrm{B}, \\
& c(x(\tau), u(\tau)) \le 0, && \forall \tau \in [0,1].
\end{aligned}
$$

Discretize...

Now perform a standard multiple shooting discretization.

$$
\begin{aligned}
\underset{\substack{x_0,\dots,x_N,\\ u_0,\dots,u_{N-1},\\ T_0,\dots,T_{N-1}}}{\text{minimize}} \quad & T = \sum_{k=0}^{N-1} \frac{T_k}{N} && && \text{(1a)}\\
\text{subject to} \quad & x_{k+1} = f_T(x_k, u_k, T_k), && k = 0,\dots,N-1, && \text{(1b)}\\
& x_0 = \mathrm{A}, && && \text{(1c)}\\
& x_N = \mathrm{B}, && && \text{(1d)}\\
& c(x_k, u_k) \le 0, && k = 0,\dots,N-1, && \text{(1e)}\\
& T_k = T_{k+1}, && k = 0,\dots,N-2, && \text{(1f)}\\
& 0 \le T_k, && k = 0,\dots,N-1, && \text{(1g)}
\end{aligned}
$$

Discretize...

Now perform a standard multiple shooting discretization.

$$
\begin{aligned}
\underset{\substack{x_0,\ldots,x_N,\\ u_0,\ldots,u_{N-1},\\ T_0,\ldots,T_{N-1}}}{\text{minimize}} \quad & T = \sum_{k=0}^{N-1} \frac{T_k}{N} && && \text{(1a)}\\
\text{subject to} \quad & x_{k+1} = f_T(x_k, u_k, T_k), && k = 0, \ldots, N-1, && \text{(1b)}\\
& x_0 = \mathrm{A}, && && \text{(1c)}\\
& x_N = \mathrm{B}, && && \text{(1d)}\\
& c(x_k, u_k) \leq 0, && k = 0, \ldots, N-1, && \text{(1e)}\\
& T_k = T_{k+1}, && k = 0, \ldots, N-2, && \text{(1f)}\\
& 0 \leq T_k, && k = 0, \ldots, N-1, && \text{(1g)}
\end{aligned}
$$

With T as a control, sparsity pattern is preserved!

One big problem

When used in NMPC, difficult to prove stability

- because time might shrink and expand

One big problem

When used in NMPC, difficult to prove stability

- because time might shrink and expand

Solution?

- Other scheme $\rightarrow$ This talk

Let's take a step back

What is a time-optimal solution in discrete time?

$$
\begin{aligned}
\min_{\substack{N, x_0, \ldots, x_N, \\ u_0, \ldots, u_{N-1}}} \quad & N \\
\text{s.t.} \quad & x_{k+1} = f_{\mathrm{d}}(x_k, u_k), \; k = 0, \ldots, N-1, \\
& x_0 = \overline{x}, \\
& x_N = 0, \\
& c(x_k, u_k) \leq 0, \qquad k = 0, \ldots, N-1.
\end{aligned}
\tag{2}
$$

Definition

We define a time-optimal solution subject to discrete dynamical system $x_{k+1} = f_{\mathrm{d}}(x_k, u_k)$ as any solution to (2) that brings the system from $\overline{x}$ to the origin in $N^\star(\overline{x})$ steps, where $N^\star(\overline{x})$ is the solution to (2). Furthermore, let $\mathcal{X}_{N^\star}$ denote the set of states $\overline{x}$ such that the optimal value of (2) is smaller than or equal to $N^\star(\overline{x})$.

The crux

We want to get 'as fast as possible' from $\overline{x}_0$ to 0.

The crux

We want to get 'as fast as possible' from $\overline{x}_0$ to 0.

- Due to Definition 1, this is a feasibility problem.

$$\min_{\substack{x_0,\ldots,x_{N^\star},\\ u_0,\ldots,u_{N^\star-1}}} \quad 0 \tag{3a}$$

$$\text{s.t.} \quad x_{k+1} = f_{\mathrm{d}}(x_k, u_k),\ k = 0, \ldots, N^\star - 1, \tag{3b}$$

$$x_0 = \overline{x}, \tag{3c}$$

$$x_{N^\star} = 0, \tag{3d}$$

$$c(x_k, u_k) \leq 0, \qquad k = 0, \ldots, N^\star - 1 \tag{3e}$$

Did we shoot ourselves in the foot again?

Yes. But we can solve that!

Did we shoot ourselves in the foot again?

Yes. But we can solve that!

- We assume $N > N^\star$.

Did we shoot ourselves in the foot again?

Yes. But we can solve that!

- We assume $N > N^\star$.

$$\begin{aligned}
\min_{\substack{x_0,\dots,x_N,\\ u_0,\dots,u_{N-1}}} \quad & \sum_{k=0}^{N-1} \theta^k \|x_k\|_1 && \text{(4a)}\\
\text{s.t.} \quad & x_{k+1} = f_\mathrm{d}(x_k, u_k),\ k = 0, \dots, N-1, && \text{(4b)}\\
& x_0 = \overline{x}, && \text{(4c)}\\
& x_N = 0, && \text{(4d)}\\
& c(x_k, u_k) \leq 0, \qquad k = 0, \dots, N-1. && \text{(4e)}
\end{aligned}$$

Did we shoot ourselves in the foot again?

Yes. But we can solve that!

- We assume $N > N^\star$.

$$
\begin{aligned}
\min_{\substack{x_0,\dots,x_N,\\ u_0,\dots,u_{N-1}}} \quad & \sum_{k=0}^{N-1} \theta^k \|x_k\|_1 && \text{(4a)}\\
\text{s.t.} \quad & x_{k+1} = f_{\mathrm{d}}(x_k, u_k),\ k = 0,\dots,N-1, && \text{(4b)}\\
& x_0 = \overline{x}, && \text{(4c)}\\
& x_N = 0, && \text{(4d)}\\
& c(x_k, u_k) \leq 0, \qquad k = 0,\dots,N-1. && \text{(4e)}
\end{aligned}
$$

Exponential weighting is important here!

Some motivation

- We know the following from Augmented Lagrangian theory:

l_1 Penalty

Consider

$$\begin{aligned} w^\star := \quad & \arg\min_{w} \quad \phi(w) \\ & \text{subject to} \;\; g(w) = 0 \quad | \quad \lambda \end{aligned}$$

Some motivation

- We know the following from Augmented Lagrangian theory:

l_1 Penalty

Consider

$$w^\star := \quad \arg\min_{w} \quad \phi(w) \qquad \text{subject to} \;\; g(w) = 0 \quad | \quad \lambda$$

Then $w^\star$ is a local minimizer of

$$\operatorname*{minimize}_{w} \quad \phi_1(w) := \phi(w) + \mu \, \|g(w)\|_1$$

where $\mu > \|\lambda^\star\|_\infty$.

Time-optimal OCP scheme

Let's apply that to our scheme.

$$
\begin{aligned}
\min_{\substack{x_0,\ldots,x_N,\\ u_0,\ldots,u_{N-1}}} \quad & 0 && \text{(5a)}\\
\text{s.t.} \quad & x_{k+1} = f_{\mathrm{d}}(x_k, u_k),\ k = 0, \ldots, N-1, && \text{(5b)}\\
& x_0 = \overline{x}, && \text{(5c)}\\
& x_N^{\star} = 0, && \text{(5d)}\\
& x_N = 0, && \text{(5e)}\\
& c(x_k, u_k) \leq 0, \qquad k = 0, \ldots, N-1. && \text{(5f)}
\end{aligned}
$$

Time-optimal OCP scheme

Remove $x_{N^\star} = 0$:

$$
\begin{aligned}
\min_{\substack{x_0,\ldots,x_N,\\ u_0,\ldots,u_{N-1}}} \quad & 0 && \text{(6a)}\\
\text{s.t.} \quad & x_{k+1} = f_\mathrm{d}(x_k, u_k),\ k = 0, \ldots, N-1, && \text{(6b)}\\
& x_0 = \overline{x}, && \text{(6c)}\\
& x_N = 0, && \text{(6d)}\\
& c(x_k, u_k) \leq 0, \qquad k = 0, \ldots, N-1. && \text{(6e)}
\end{aligned}
$$

Time-optimal OCP scheme

Add penalty:

$$
\begin{aligned}
\min_{\substack{x_0,\ldots,x_N,\\ u_0,\ldots,u_{N-1}}} \quad & \sum_{k=0}^{N-1} \theta^k \|x_k\|_1 && \text{(7a)}\\
\text{s.t.} \quad & x_{k+1} = f_{\mathrm{d}}(x_k, u_k),\ k = 0,\ldots,N-1, && \text{(7b)}\\
& x_0 = \overline{x}, && \text{(7c)}\\
& x_N = 0, && \text{(7d)}\\
& c(x_k, u_k) \leq 0, \qquad k = 0,\ldots,N-1. && \text{(7e)}
\end{aligned}
$$

Two proofs

We need to prove two things before this is useful:

- That our scheme induces time-optimal solutions (OCP)
  - Basically, that $\theta^{N^\star} > \|\lambda^\star\|_\infty$

Two proofs

We need to prove two things before this is useful:

- That our scheme induces time-optimal solutions (OCP)
  - Basically, that $\theta^{N^\star} > \|\lambda^\star\|_\infty$
- That it is stable (NMPC)
  - Standard Lyapunov-based proof.

Two proofs

We need to prove two things before this is useful:

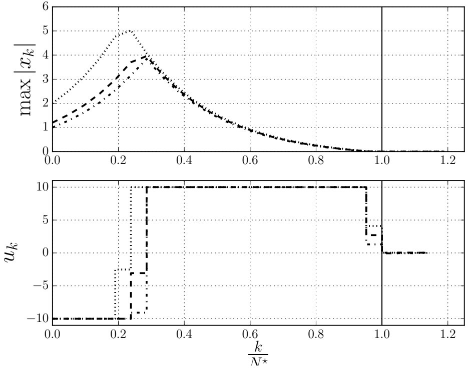
- That our scheme induces time-optimal solutions (OCP)
  - Basically, that $\theta^{N^\star} > \|\lambda^\star\|_\infty$
- That it is stable (NMPC)
  - Standard Lyapunov-based proof.

$\rightarrow$ see CDC2017 paper

Some illustrations

Toy example with a model from Chen1998:

$$\dot{p} = q + u(\mu + (1-\mu)p),$$
$$\dot{q} = p + u(\mu - 4(1-\mu)q).$$



Hanging pendulum with varying length

Pendulum has a trolley (subscript ‘t’) and a cable (subscript ‘c’).

$$\dot{x}_t = v_t, \quad \dot{v}_t = a_t, \tag{8a}$$

$$\dot{x}_c = v_c, \quad \dot{v}_c = a_c, \tag{8b}$$

$$\dot{\varphi} = \omega, \tag{8c}$$

$$\dot{\omega} = \frac{-(2\omega v_c + a_t \cos(\varphi) + g \sin(\varphi))}{x_c}. \tag{8d}$$

Hanging pendulum with varying length

Optimal control result, compared with the 'scaling' method:

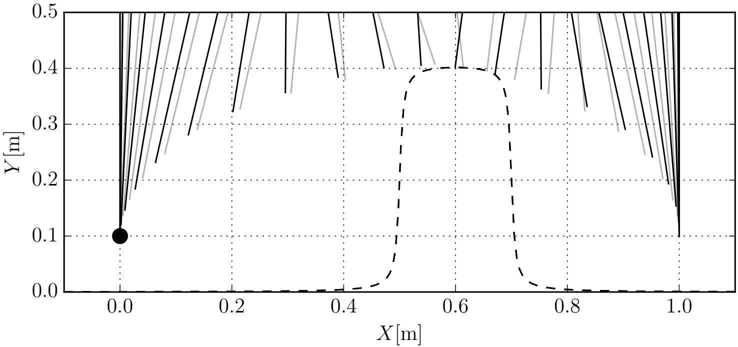


Figure: Black: exponential weighting, gray: time scaling

Hanging pendulum with varying length

Optimal control result, compared with the 'scaling' method:

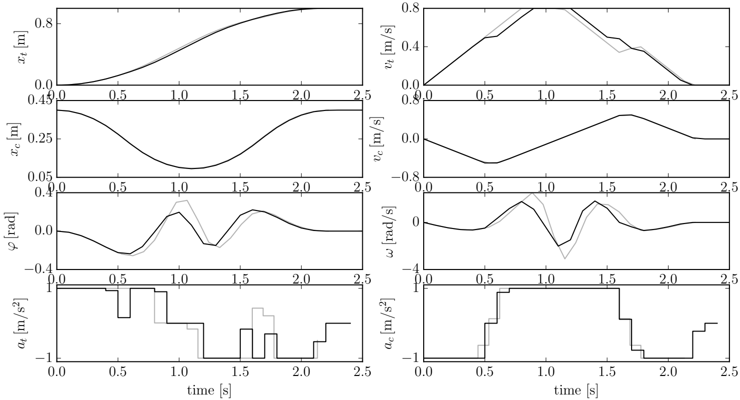


Figure: Black: exponential weighting, gray: time scaling

Hanging pendulum with varying length

NMPC with Hardware-In-the-Loop (ABB AC 800PEC, used for time- and safety-critical applications)

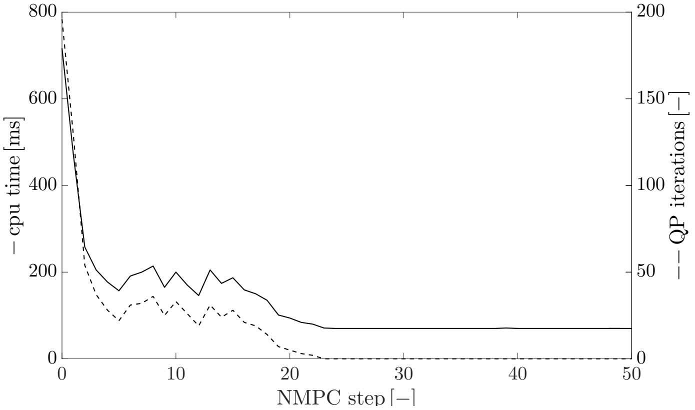


Figure: Upper: cpu time, lower: QP iterations

Conclusion

We found a new time-optimal control formulation which

- does not use time scaling,
- and so facilitates a stability proof,
- and is useful in practice.

That was it!
Questions?