OPTIMIZATION FOR REAL-TIME CONTROL WITH LIMITED RESOURCES

Themelis, Andreas
Kyushu University

https://hdl.handle.net/2324/6790350

出版情報：2021-08-22
バージョン：
権利関係：

OPTIMIZATION FOR REAL-TIME CONTROL

WITH LIMITED RESOURCES

Andreas Themelis
Kyushu University
`andreas.themelis@ees.kyushu-u.ac.jp`

Real time optimization for control: basics and recent progress

August 22, 2021 @ E-COSM 2021

Optimization: what and why

Optimization: what and why

Technology of devising effective *decisions* or *predictions*

Formal definition

Choose **variables** from within an allowed set that minimize a cost

Applications

- **Engineering design:** plants, building, circuits
- **Resource allocation:** logistics, finance, communications
- **Machine learning:** (un)supervised learning, regression
- **Signal processing:** compressed sensing, signal estimation
- **Control:** autonomous systems, path planning, tracking, navigation

Optimization: what and why

Examples — Recommender system

- Which items you might like based on rating database $\bar{X}_{i,j} \quad (i,j) \in \Omega$
- Ratings depend on few *“latent features”*

$$\underset{X \in \mathbb{R}^{m \times n}}{\text{minimize}} \; \|(X - \bar{X})_{\Omega}\|^2 + \lambda \operatorname{rank} X$$

$$X = \begin{array}{cccccccccccccccccc} & & & & & & & & & & & \text{user } j & & & & & & \\ \left(\begin{matrix} ? \\ ? \\ ? \\ ? \\ ? \end{matrix}\right. & ? & ? & ? & ? & ? & ? & ? & ? & ? & & & & & & & & \end{array}$$

$$X = \left(\begin{array}{cccccccccccccccc} ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? \\ ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & 4 & ? & ? & ? & ? & ? \\ ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? \\ ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? \\ ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? \end{array}\right)$$

Optimization: what and why

Examples — Back/foreground extraction

- Separate fore/background from N many $m \times n$ video frames

- $S = [\text{frame } 1, \ \ldots \ , \text{frame } N] \in \mathbb{R}^{mn \times N}$
- Foreground X: moving objects ⇒ sparse
- Background Y: constant over time ⇒ low rank

$$\underset{X,Y}{\text{minimize}} \, \|X + Y - S\|^2 + \lambda \, \mathrm{nnz}(X) \quad \text{subject to} \, \mathrm{rank}\, Y \leq k$$

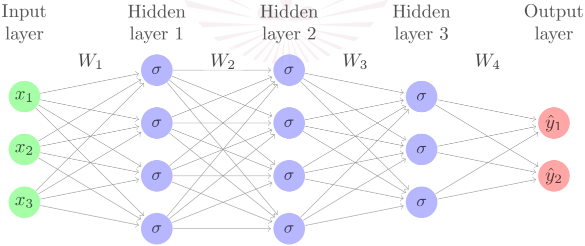
Optimization: what and why

Examples — Deep NNs

- **“Learn”** weights W_j defining a classifying function
- by **“training”**

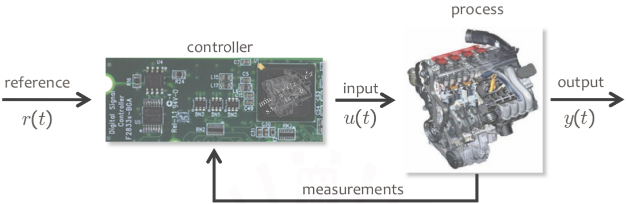
$$\underset{W_1 \ldots W_4}{\text{minimize}} \ \frac{1}{N} \sum_{i=1}^{N} \ell(y^i, \hat{y}^i) \quad \text{where} \quad \hat{y}^i = W_4\sigma\Big(W_3\sigma\Big(W_2\sigma\Big(W_1\sigma(x^i)\Big)\Big)\Big)$$

- on a training set (x^i, y^i)



Optimization: what and why

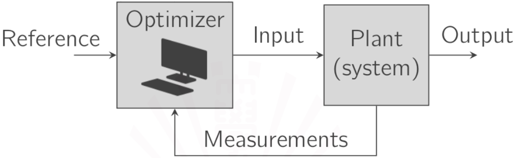
Optimization in control



A **controller** is a decision-making mechanism
who decides & actuates **inputs** to steer the plant/process/system

Optimization: what and why

Optimization in control



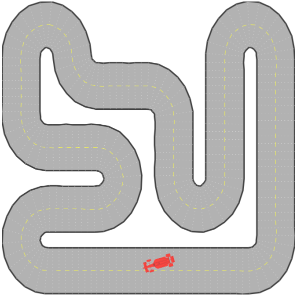
Conceptual example

Goal

- Fastest trajectory

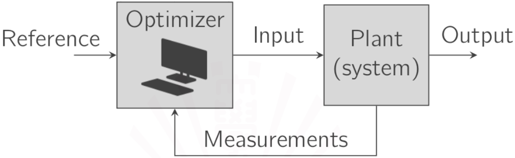
Constraints

- Stay on road
- Avoid other vehicles
- Physical limits (speed,. . .)



Optimization: what and why

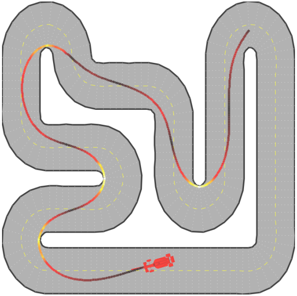
Optimization in control



Conceptual example

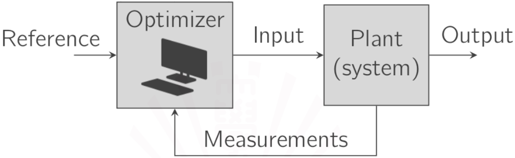
Problem

minimize	circuit time
considering	car dynamics road conditions
while	avoiding other cars staying on road



Optimization: what and why

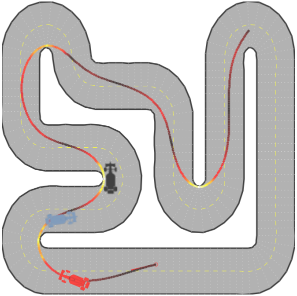
Optimization in control



Conceptual example

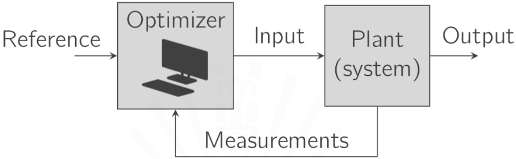
Input actuation

- Apply optimal input
- Check car/environment again!



Optimization: what and why

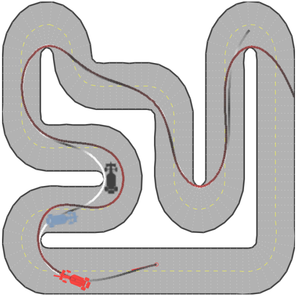
Optimization in control



Conceptual example

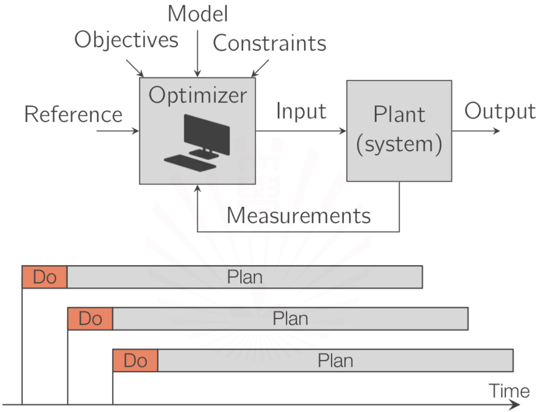
Feedback

- **Discard** the long-term prediction
- **Restart** based on new info/measurements



Optimization: what and why

Optimization in control — MPC

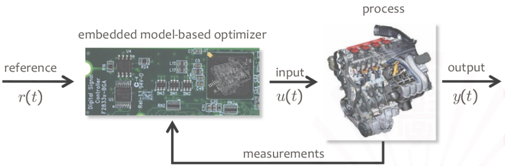


Feedback granted by **receding horizon** strategy

Optimization: what and why

Optimization in control — MPC in a nutshell

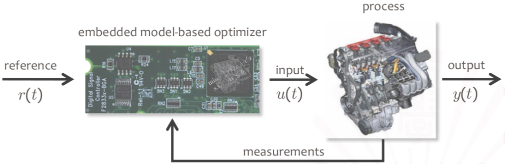
Use a dynamical model of the process to predict its evolution and choose control actions

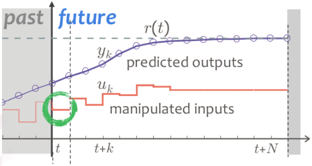


Optimization: what and why

Optimization in control — MPC in a nutshell

Use a dynamical model of the process to predict its evolution and choose control actions by **recursively solving** finite discrete-time optimal control problems in a **receding horizon** fashion





Optimal control problem

$$\underset{u,x}{\text{minimize}} \sum_{t=0}^{N-1} \ell_t(x_t,u_t)+\ell_N(x_N)$$

$$\text{s.t.}\, x_{t+1}=F_t(x_t,u_t)$$

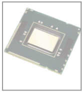
$$u_t \in \mathcal{U}_t$$

$$x_t \in \mathcal{X}_t$$

Optimization: what and why

Challenges — Fast sampling rates

New optimization problems must be solved within **sampling time**



Computer control — ns

μs — Power systems

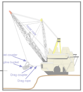
Traction control — ms

Seconds — Buildings

Refineries — Minutes

Hours — Nurse rostering

Train scheduling — Days



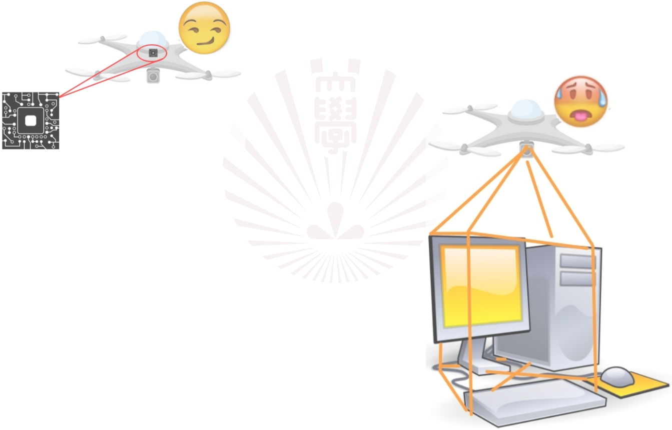
Weeks — Production planning

High-computing power required

Optimization: what and why

Challenges — Embeddability

Some applications **can't afford** high computing

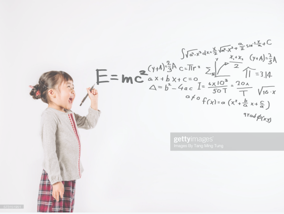


Algorithms must be **embeddable** on low-power chipsets

Optimization: what and why

Challenges — Nonsmoothness

Constraints ⇒ nonsmooth problem



[On paper]

no big deal for modern optimization

🤦

[In practice]

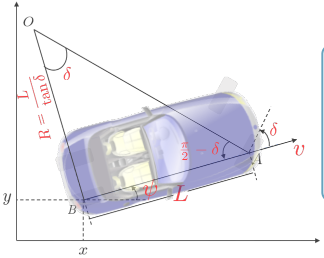
fast algorithms (IP, SQP...) **too heavy** for low-power chipsets

😟

Optimization: what and why

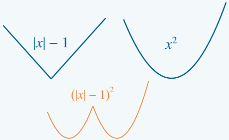
Challenges — Nonconvexity

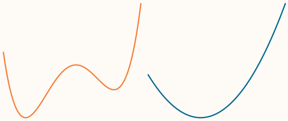
Nonlinear dynamics ⇒ nonconvex problem



✓ convex ∘ **linear** = convex

✗ convex ∘ **nonlinear** = **?**





- Local VS global minima
  ...let's live with it
- Much theory **not applicable**
  - ✗ Duality
  - ✗ Monotone operators
  - ✗ Fejér monotonicity
  - ...

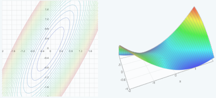
Optimization: what and why

Challenges — Summary

ms-**fast**

< sampling time

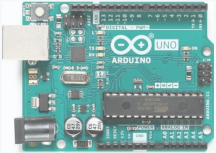
Reliable

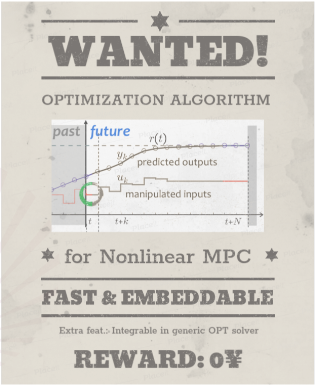


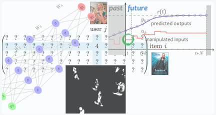
vs
- ill-scaling
- nonsmoothness
- nonconvexity

Embeddable

simple operations only







& more(?)

✓ warm-startable
✓ multi-purpose

Problem setting & toolbox

Problem setting & toolbox

Functions, variables, constraints

$$\underset{u,\,x}{\text{minimize}} \sum_{t=0}^{N-1} \ell_t(x_t, u_t) + \ell_N(x_N) \qquad \text{subject to} \begin{cases} x_{t+1} = F_t(x_t, u_t) \\ u_t \in \mathcal{U}_t \\ x_t \in \mathcal{X}_t \end{cases}$$

Requirements

- ℓ_t, F_t smooth (e.g. C^2) — can be relaxed
- input constraints $\mathcal{U}_t$ *easy* to project onto (boxes, balls…) — no need for convex constraints, but typical in practice

Problem setting & toolbox

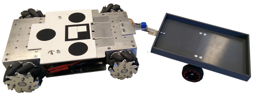
Functions, variables, constraints

$$\underset{u,\, x}{\text{minimize}} \sum_{t=0}^{N-1} \ell_t(x_t, u_t) + \ell_N(x_N) \qquad \text{subject to} \begin{cases} x_{t+1} = F_t(x_t, u_t) \\ u_t \in \mathcal{U}_t \\ x_t \in \mathcal{X}_t \end{cases}$$

Requirements

- ℓ_t, F_t smooth (e.g. C^2) — can be relaxed
- input constraints $\mathcal{U}_t$ *easy* to project onto (boxes, balls…) — no need for convex constraints, but typical in practice

---



Example: holonomic land vehicle with trailer

- steer to reference pos./orient. $(p_\star, \vartheta_\star)$
- avoid obstacles

Dynamics

$$\begin{cases} \dot{p}_x = u_x + L\dot{\vartheta} \sin\vartheta \\ \dot{p}_y = u_y - L\dot{\vartheta} \cos\vartheta \\ \dot{\vartheta} = \frac{1}{L}(u_y \cos\vartheta - u_x \sin\vartheta) \end{cases}$$

(input) velocity u
(state) trailer pos. p & head. angle ϑ
(const) trailer arm length L

MPC problem

Given current pos. p_0 & head. angle θ_0,

$$\underset{\substack{u_{t-1}, (p_t, \vartheta_t) \\ t=1\ldots N}}{\text{minimize}} \sum_{t=1}^{N} \frac{C_x}{2} \|p_t - p_\star\|^2 + \frac{C_x}{2} \|\vartheta_t - \vartheta_\star\|^2 + \frac{C_u}{2} \|u_{t-1}\|^2$$

$$\text{subject to} \begin{cases} (p_{t+1}, \theta_{t+1}) = F_t(p_t, \vartheta_t, u_t) \\ u_{\min} \leq u_t \leq u_{\max} \\ p_t \notin \text{obstacles area} \end{cases}$$

F_t discretized dynamics (e.g. RK4)

Problem setting & toolbox

Functions, variables, constraints

$$\operatorname*{minimize}_{u,\,x} \sum_{t=0}^{N-1} \ell_t(x_t, u_t) + \ell_N(x_N) \qquad \text{subject to} \begin{cases} x_{t+1} = F_t(x_t, u_t) \\ u_t \in \mathcal{U}_t \\ x_t \in \mathcal{X}_t \end{cases}$$

Requirements

- ℓ_t, F_t smooth (e.g. C^2) — can be relaxed
- input constraints $\mathcal{U}_t$ *easy* to project onto (boxes, balls. . .) — no need for convex constraints, but typical in practice

---

Multiple shooting

Treat x as variable

(keep dynamics as constraints)

☹ $3\times$ larger variable

☺ sparse formulation

☹ dynamics not respected

☺ easy state constraints

Problem setting & toolbox

Functions, variables, constraints

$$\underset{u,\,\cancel{x}}{\text{minimize}} \sum_{t=0}^{N-1} \ell_t(x_t(u), u_t) + \ell_N(x_N(u)) \quad \text{subject to} \begin{cases} u_t \in \mathcal{U}_t \\ x_t(u) \in \mathcal{X}_t \end{cases}$$

Requirements

- ℓ_t, F_t smooth (e.g. C^2) — can be relaxed
- input constraints $\mathcal{U}_t$ *easy* to project onto (boxes, balls. . .) — no need for convex constraints, but typical in practice

---

Multiple shooting

Treat x as variable

(keep dynamics as constraints)

- ☹ 3× larger variable
- ☺ sparse formulation
- ☹ dynamics not respected
- ☺ easy state constraints

Single shooting

Express x in terms of u

(keep only u as optim. variable)

- ☺ smaller variable
- ☹ densely nested nonlinearities
- ☺ dynamics inherently satisfied
- ☹ complicated state constraints

Problem setting & toolbox

Simpl*est* formulation

- **Single shooting** approach (keep only inputs u)

  For $t = 1, \ldots, N$, recursively express

$$\begin{aligned} x_t &= x_t(u_0, \ldots, u_{t-1}) \\ &= F_{t-1}(x_{t-1}(u_0, \ldots, u_{t-2}), u_{t-1}) \end{aligned}$$

- For now, **discard state constraints** (we'll fix this later) 😇

Problem setting & toolbox

Simplest formulation

- **Single shooting** approach (keep only inputs u)

  For $t = 1, \dots, N$, recursively express

$$\begin{aligned} x_t &= x_t(u_0, \dots, u_{t-1}) \\ &= F_{t-1}(x_{t-1}(u_0, \dots, u_{t-2}), u_{t-1}) \end{aligned}$$

- For now, **discard state constraints** (we'll fix this later) 😇

- Problem becomes

$$\underset{u}{\text{minimize}} \ \underbrace{f(u; x_0)}_{\text{smooth}} \quad \text{subject to} \ u \in \underbrace{\mathcal{U}_0 \times \cdots \times \mathcal{U}_{N-1}}_{\text{easy to project onto (e.g. } u_{\text{MIN}} \le u_t \le u_{\text{MAX}})}$$

Problem setting & toolbox

Simpl*est* formulation

$$\operatorname*{minimize}_{u} \ \underbrace{f(u;\, x_0)}_{\text{smooth}} \quad \text{subject to } \ u \in \underbrace{\mathcal{U}_0 \times \cdots \times \mathcal{U}_{N-1}}_{\text{easy to project onto (e.g. } u_{\text{MIN}} \le u_t \le u_{\text{MAX}})}$$

(Projected) gradient method (Cauchy, 1847)

$$\text{iterate} \quad u \leftarrow \underbrace{\Pi_{\mathcal{U}}}_{\text{projection on constraints}} \overbrace{(u - \gamma \nabla f(u))}^{\text{gradient descent step}}$$

Augustin-Louis Cauchy (1789-1857)

A. Cauchy, *Méthode générale pour la résolution des systemes d'équations simultanées,* Comp. Rend. Sci. Paris **25**:536–538, **1847**

Problem setting & toolbox

Simple*st* formulation

$$\operatorname*{minimize}_{u}\ \underbrace{f(u;\, x_0)}_{\text{smooth}} \quad \text{subject to}\ u \in \underbrace{\mathcal{U}_0 \times \cdots \times \mathcal{U}_{N-1}}_{\text{easy to project onto (e.g. } u_{\mathsf{MIN}} \le u_t \le u_{\mathsf{MAX}})}$$

(Projected) gradient method (Cauchy, 1847)

$$\text{iterate} \quad u \leftarrow \underbrace{\Pi_{\mathcal{U}}}_{\text{projection on constraints}} \overbrace{(u - \gamma \nabla f(u))}^{\text{gradient descent step}}$$

Augustin-Louis Cauchy (1789-1857)

Arguably the **simplest** possible method

- ☺ **Embeddable**
- ☺ **Minimal assumptions**

Unreliable for real-time applications

- ☹ **Slow**
- ☹ Sensitive to conditioning

A. Cauchy, *Méthode générale pour la résolution des systemes d'équations simultanées,* Comp. Rend. Sci. Paris **25**:536–538, **1847**

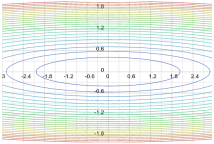
Problem setting & toolbox

Simplest formulation (but also slowest)

(Unconstrained) example

Fix $c \gg 1$

$$\underset{(x,y)\in\mathbb{R}^2}{\text{minimize}} \; \overbrace{\tfrac{1}{2}x^2 + \tfrac{c}{2}y^2}^{f(x,y)}$$



Gradient method

$$\begin{pmatrix} x^+ \\ y^+ \end{pmatrix} = \begin{pmatrix} x \\ y \end{pmatrix} - \gamma \overbrace{\begin{pmatrix} x \\ cy \end{pmatrix}}^{\nabla f(x,y)} = \begin{pmatrix} (1-\gamma)x \\ (1-c\gamma)y \end{pmatrix}$$

Starting from $(x^{(0)}, y^{(0)}) = (\bar{x}, \bar{y})$,

$$\begin{pmatrix} x^{(k)} \\ y^{(k)} \end{pmatrix} = \begin{pmatrix} (\,1-\gamma\,)^k\,\bar{x} \\ (1-c\gamma)^k\,\bar{y} \end{pmatrix}$$

- If $\gamma \notin (0, {}^2\!/_c)$ no convergence
- If $\gamma \in (0, {}^2\!/_c)$ $(x^{(k)}, y^{(k)}) \to (0,0)$ but **very slowly!**

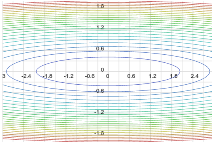
Problem setting & toolbox

Simple*st* formulation (but also slowe*st*)

(Unconstrained) example

Fix $c \gg 1$

$$\underset{(x,y)\in\mathbb{R}^2}{\text{minimize}}\ \overbrace{\tfrac{1}{2}x^2 + \tfrac{c}{2}y^2}^{f(x,y)}$$



Gradient method

$$\begin{pmatrix} x^+ \\ y^+ \end{pmatrix} = \begin{pmatrix} x \\ y \end{pmatrix} - \gamma \overbrace{\begin{pmatrix} x \\ cy \end{pmatrix}}^{\nabla f(x,y)} = \begin{pmatrix} (1-\gamma)x \\ (1-c\gamma)y \end{pmatrix}$$

Starting from $(x^{(0)}, y^{(0)}) = (\bar{x}, \bar{y})$,

$$\begin{pmatrix} x^{(k)} \\ y^{(k)} \end{pmatrix} = \begin{pmatrix} (\ 1-\gamma\ )^k\ \bar{x} \\ (1-c\gamma)^k\ \bar{y} \end{pmatrix}$$

- If $\gamma \notin (0, {}^{2}\!/_{c})$ no convergence
- If $\gamma \in (0, {}^{2}\!/_{c})$ $(x^{(k)}, y^{(k)}) \to (0,0)$ but **very slowly!**

- Take $\gamma = {}^{1}\!/_{c}$, $(\bar{x}, \bar{y}) = (1,1)$
- then $x^{(k)} = (1 - {}^{1}\!/_{c})^k$ and $y^{(k)} = 0$
- to be ε-close to the solution, $\frac{\log \varepsilon}{\log 1 - {}^{1}\!/_{c}} \approx c \log \frac{1}{\varepsilon}$ iterations needed
- $c = 3 \cdot 10^5$, **>1million** iterations needed to reach $\varepsilon = 0.1$ accuracy

Problem setting & toolbox

Speeding up (textbook attempt I)

Let's keep things **unconstrained**…

$$\underset{u\in\mathbb{R}^n}{\text{minimize}}\, f(u) \qquad f \text{ (twice) smooth}$$

Isaac Newton (1642-1727)

Newton's method

- Suppose $\nabla^2 f(u) \succ 0$,

$$u^+ = u - \nabla^2 f(u)^{-1} \nabla f(u)$$

Ex. $f(x,y) = \frac{1}{2}x^2 + \frac{c}{2}y^2 \;\Rightarrow$ convergence in **1 iteration!**

I. Newton, *Philosophiae Naturalis Principia Mathematica,* London, **1687**

Problem setting & toolbox

Speeding up (textbook attempt I)

Let's keep things **unconstrained**…

$$\underset{u\in\mathbb{R}^n}{\text{minimize}}\, f(u) \qquad f \ \text{(twice) smooth}$$

Isaac Newton (1642-1727)

Newton's method

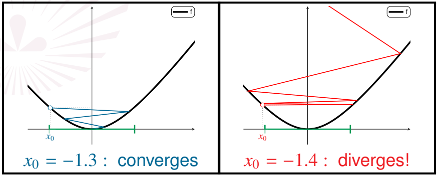
- Suppose $\nabla^2 f(u) \succ 0$,

$$u^+ = u - \nabla^2 f(u)^{-1} \nabla f(u)$$

But
applied to

$$f(x) = x \arctan x - \tfrac{1}{2}\log(1+x^2)$$

(convex, C^∞, Lipschitz differentiable)



$x_0 = -1.3$: converges

$x_0 = -1.4$: diverges!

I. Newton, *Philosophiae Naturalis Principia Mathematica*, London, **1687**

Problem setting & toolbox

Speeding up (textbook attempt I)

Let's keep things **unconstrained**…

$$\underset{u\in\mathbb{R}^n}{\text{minimize}}\, f(u) \qquad f \text{ (twice) smooth}$$

Isaac Newton (1642-1727)

Damped Newton's method

- Suppose $\nabla^2 f(u) \succ 0$,

$$u^+ = u - \tau \nabla^2 f(u)^{-1} \nabla f(u)$$

- Convergence only close to a solution
- In general, need τ small enough to guarantee $f(u^+) < f(u)$

I. Newton, *Philosophiae Naturalis Principia Mathematica,* London, **1687**

Problem setting & toolbox

Speeding up (textbook attempt I)

Let's keep things **unconstrained**…

$$\underset{u \in \mathbb{R}^n}{\text{minimize}}\, f(u) \qquad f \text{ (twice) smooth}$$

Isaac Newton (1642-1727)

Damped Newton's method

- Suppose $\nabla^2 f(u) \succ 0$,

$$u^+ = u - \tau \nabla^2 f(u)^{-1} \nabla f(u)$$

- Convergence only close to a solution
- In general, need τ small enough to guarantee $f(u^+) < f(u)$
- **Linesearch:**

$$\langle \nabla f(u), d \rangle < 0 \quad \Rightarrow \quad f(u + \tau d) = f(u) + \tau \langle \nabla f(u), d \rangle + o(\tau) < f(u)$$

for τ small enough!

I. Newton, *Philosophiae Naturalis Principia Mathematica*, London, **1687**

Problem setting & toolbox

Speeding up (textbook attempt II)

Let's **continue** keeping things **unconstrained**…

$$\underset{u \in \mathbb{R}^n}{\text{minimize}}\, f(u) \qquad f \ \text{(twice) smooth}$$

Quasi-Newton methods

- Computing $\nabla^2 f$ impractical
- Newton direction $-\nabla^2 f^{-1}(u) \nabla f(u)$ approximated with linear algebra

Problem setting & toolbox

Speeding up (textbook attempt II)

Let's **continue** keeping things **unconstrained**…

$$\underset{u \in \mathbb{R}^n}{\text{minimize}}\, f(u) \qquad f \text{ (twice) smooth}$$

Quasi-Newton methods

- Computing $\nabla^2 f$ impractical
- Newton direction $-\nabla^2 f^{-1}(u) \nabla f(u)$ approximated with linear algebra

Key idea:

$$\underbrace{\nabla f(u^{(k)}) - \nabla f(u^{(k-1)})}_{y^{(k)}} \approx \underbrace{\nabla^2 f(u^{(k)})}_{\approx B_{k+1}} \underbrace{(u^{(k)} - u^{(k-1)})}_{s^{(k)}}$$

  - Update estimate $B_k \mapsto B_{k+1}$ by enforcing the *"secant condition"*

$$B_{k+1} s^{(k)} = y^{(k)}$$

  - *"Limited memory"* variants (e.g. **L-BFGS**) need **vector-vector products only!**
  - Can ensure convergence with suitable **linesearch**

Problem setting & toolbox

Speeding up — Summary

Projected grandient method

☺ Cheap**(est)**
☺ Constraints ✓

☹ Slow

Newton method

☺ Fast **(very!)**

☹ Expensive
☹ Constraints ✗

Quasi-Newton methods

☺ Fast
☺ Cheap

☹ Constraints ✗

Problem setting & toolbox

Speeding up — Summary

Projected grandient method

☺ Cheap**(est)** ☺ Constraints ✓	☹ Slow

Newton method

☺ Fast **(very!)**	☹ Expensive ☹ Constraints ✗

Quasi-Newton methods

☺ Fast ☺ Cheap	☹ Constraints ✗

Linesearch methods

$$u^{+} = u + \tau d$$

all require

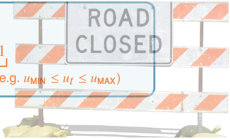
- $\langle \nabla f(u), d \rangle < 0$
- (in particular f **smooth**)

Novel speedup

Novel speedup

Fast update directions — for nonsmooth problems

$$\underset{u}{\text{minimize}}\ \underbrace{f(u;\,x_0)}_{\text{smooth}} \quad \text{subject to}\ u \in \underbrace{\mathcal{U}_0 \times \cdots \times \mathcal{U}_{N-1}}_{\text{easy to project onto (e.g. } u_{\mathsf{MIN}} \le u_t \le u_{\mathsf{MAX}})}$$



Optimality conditions

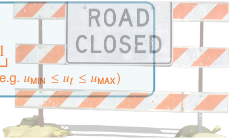
$$u \text{ local minimum} \quad \Rightarrow \quad u - \Pi_{\mathcal{U}}(u - \gamma \nabla f(u)) = 0$$

- If $\mathcal{U} = \mathbb{R}^n$ (unconstrained), reduces to $\nabla f(u) = 0$

Novel speedup

Fast update directions — for nonsmooth problems

$$\underset{u}{\text{minimize}}\ \underbrace{f(u;\, x_0)}_{\text{smooth}} \quad \text{subject to}\ u \in \underbrace{\mathcal{U}_0 \times \cdots \times \mathcal{U}_{N-1}}_{\text{easy to project onto (e.g. } u_{\mathsf{MIN}} \le u_t \le u_{\mathsf{MAX}})}$$



Optimality conditions

$$u \text{ local minimum} \quad \Rightarrow \quad \overbrace{u - \Pi_{\mathcal{U}}(u - \gamma \nabla f(u))}^{\mathcal{R}(u)} = 0$$

- **Idea** for “fast” directions

Quasi-Newton on $\mathcal{R}$

$$\underbrace{\mathcal{R}(u^{(k)}) - \mathcal{R}(u^{(k-1)})}_{y^{(k)}} = B_{k+1} \underbrace{(u^{(k)} - u^{(k-1)})}_{s^{(k)}}$$

Novel speedup

Fast update directions — for nonsmooth problems

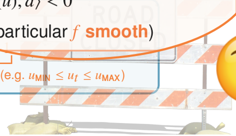
$$\underset{u}{\text{minimize}}\ \underbrace{f(u;\, x_0)}_{\text{smooth}} \quad \text{subject to}\ u \in \underbrace{\mathcal{U}_0 \times \cdots \times \mathcal{U}_N}_{\text{easy to project onto (e.g. } u_{\text{MIN}} \le u_t \le u_{\text{MAX}})}$$

Linesearch methods

$$u^+ = u + \tau d$$

all require

- $\langle \nabla f(u), d \rangle < 0$
- (in particular f **smooth**)



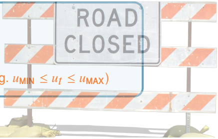
Can’t use **(classical)** linesearch

The **true** cost $u \mapsto \begin{cases} f(u) & \text{if } u \in \mathcal{U} \\ \infty & \text{if } u \notin \mathcal{U} \end{cases}$ is nonsmooth

Novel speedup

Globalization — A novel nonsmooth LS

$$\underset{u}{\text{minimize}}\ \underbrace{f(u;\, x_0)}_{\text{smooth}} \quad \text{subject to}\ u \in \underbrace{\mathcal{U}_0 \times \cdots \times \mathcal{U}_{N-1}}_{\text{easy to project onto (e.g. } u_{\mathrm{MIN}} \leq u_t \leq u_{\mathrm{MAX}})}$$



New tool: for $\gamma > 0$, define

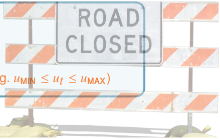
$$\varphi_\gamma(u) := \min_{w \in \mathcal{U}} \left\{ f(u) + \langle \nabla f(u), w - u \rangle + \tfrac{1}{2\gamma} \|w - u\|^2 \right\}$$

Remarks

Novel speedup

Globalization — A novel nonsmooth LS

$$\operatorname*{minimize}_{u} \underbrace{f(u; x_0)}_{\text{smooth}} \quad \text{subject to } u \in \underbrace{\mathcal{U}_0 \times \cdots \times \mathcal{U}_{N-1}}_{\text{easy to project onto (e.g. } u_{\mathsf{MIN}} \le u_t \le u_{\mathsf{MAX}})}$$



New tool: for $\gamma > 0$, define

$$\varphi_\gamma(u) := \min_{w \in \mathcal{U}} \left\{ f(u) + \underbrace{\langle \nabla f(u), w - u \rangle + \tfrac{1}{2\gamma} \|w - u\|^2}_{\frac{1}{2\gamma}\|w - u + \gamma \nabla f(u)\|^2 - \frac{\gamma}{2}\|\nabla f(u)\|^2} \right\}$$

Remarks

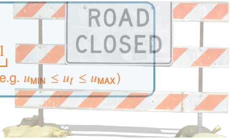
- The minimizer is $\bar{u} = \Pi_{\mathcal{U}}(u - \gamma \nabla f(u))$

$$\varphi_\gamma(u) = f(u) + \langle \nabla f(u), \bar{u} - u \rangle + \tfrac{1}{2\gamma} \|\bar{u} - u\|^2$$

Novel speedup

Globalization — A novel nonsmooth LS

$$\underset{u}{\text{minimize}}\ \underbrace{f(u;\, x_0)}_{\text{smooth}} \quad \text{subject to}\ u \in \underbrace{\mathcal{U}_0 \times \cdots \times \mathcal{U}_{N-1}}_{\text{easy to project onto (e.g. } u_{\text{MIN}} \le u_t \le u_{\text{MAX}})}$$



New tool: for $\gamma > 0$, define

$$\varphi_\gamma(u) := \min_{w \in \mathcal{U}} \left\{ f(u) + \langle \nabla f(u), w - u \rangle + \tfrac{1}{2\gamma} \|w - u\|^2 \right\}$$

Remarks

- The minimizer is $\bar{u} = \Pi_{\mathcal{U}}(u - \gamma \nabla f(u))$

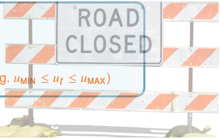
$$\varphi_\gamma(u) = f(u) + \langle \nabla f(u), \bar{u} - u \rangle + \tfrac{1}{2\gamma} \|\bar{u} - u\|^2$$

- $u \in \mathcal{U} \ \Rightarrow\ \varphi_\gamma(u) \le f(u)$

Novel speedup

Globalization — A novel nonsmooth LS

$$\operatorname*{minimize}_{u} \; \underbrace{f(u;\, x_0)}_{\text{smooth}} \quad \text{subject to } \; u \in \underbrace{\mathcal{U}_0 \times \cdots \times \mathcal{U}_{N-1}}_{\text{easy to project onto (e.g. } u_{\mathsf{MIN}} \leq u_t \leq u_{\mathsf{MAX}})}$$



New tool: for $\gamma > 0$, define

$$\varphi_\gamma(u) := \min_{w \in \mathcal{U}} \left\{ f(u) + \langle \nabla f(u), w - u \rangle + \tfrac{1}{2\gamma} \|w - u\|^2 \right\}$$

Remarks

- The minimizer is $\bar{u} = \Pi_{\mathcal{U}}(u - \gamma \nabla f(u))$

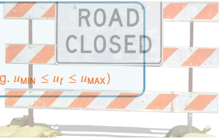
$$\varphi_\gamma(u) = f(u) + \langle \nabla f(u), \bar{u} - u \rangle + \tfrac{1}{2\gamma} \|\bar{u} - u\|^2$$

- $u \in \mathcal{U} \;\Rightarrow\; \varphi_\gamma(u) \leq f(u)$
- φ_γ is **continuous**

Novel speedup

Globalization — A novel nonsmooth LS

$$\underset{u}{\text{minimize}}\ \underbrace{f(u;\, x_0)}_{\text{smooth}} \quad \text{subject to}\ u \in \underbrace{\mathcal{U}_0 \times \cdots \times \mathcal{U}_{N-1}}_{\text{easy to project onto (e.g. } u_{\text{MIN}} \le u_t \le u_{\text{MAX}})}$$



New tool: for $\gamma > 0$, define

$$\varphi_\gamma(u) := \min_{w \in \mathcal{U}} \left\{ f(u) + \langle \nabla f(u), w - u \rangle + \tfrac{1}{2\gamma} \|w - u\|^2 \right\}$$

Remarks

- The minimizer is $\bar{u} = \Pi_{\mathcal{U}}(u - \gamma \nabla f(u))$

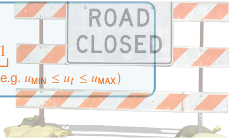
$$\varphi_\gamma(u) = f(u) + \langle \nabla f(u), \bar{u} - u \rangle + \tfrac{1}{2\gamma} \|\bar{u} - u\|^2$$

- $u \in \mathcal{U} \;\Rightarrow\; \varphi_\gamma(u) \le f(u)$
- φ_γ is **continuous**
- If $f(\bar{u}) \le f(u) + \langle \nabla f(u), \bar{u} - u \rangle + \frac{\alpha}{2\gamma} \|\bar{u} - u\|^2$, then

$$f(\bar{u}) \le \varphi_\gamma(u) - \tfrac{1-\alpha}{2\gamma} \|\bar{u} - u\|^2$$

Novel speedup

Globalization — A novel nonsmooth LS



$$\underset{u}{\text{minimize}}\ \underbrace{f(u;\, x_0)}_{\text{smooth}} \qquad \text{subject to}\ \ u \in \underbrace{\mathcal{U}_0 \times \cdots \times \mathcal{U}_{N-1}}_{\text{easy to project onto (e.g. } u_{\mathsf{MIN}} \le u_t \le u_{\mathsf{MAX}})}$$

New tool: for $\gamma > 0$, define

$$\varphi_\gamma(u) \coloneqq \min_{w\in\mathcal{U}} \left\{ f(u) + \langle \nabla f(u), w - u\rangle + \tfrac{1}{2\gamma}\|w-u\|^2 \right\}$$

Remarks

- The minimizer is $\bar{u} = \Pi_{\mathcal{U}}(u - \gamma\nabla f(u))$

$$\varphi_\gamma(u) = f(u) + \langle \nabla f(u), \bar{u} - u\rangle + \tfrac{1}{2\gamma}\|\bar{u}-u\|^2$$

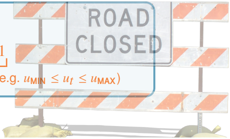
- $u \in \mathcal{U} \;\Rightarrow\; \varphi_\gamma(u) \le f(u)$
- φ_γ is **continuous**
- If $f(\bar{u}) \le f(u) + \langle \nabla f(u), \bar{u} - u\rangle + \frac{\alpha}{2\gamma}\|\bar{u}-u\|^2$, then

$$\varphi_\gamma(\bar{u}) \le f(\bar{u}) \le \varphi_\gamma(u) - \tfrac{1-\alpha}{2\gamma}\|\bar{u}-u\|^2$$

Novel speedup

Globalization — A novel nonsmooth LS

$$\underset{u}{\text{minimize}}\ \underbrace{f(u;\, x_0)}_{\text{smooth}} \quad \text{subject to}\ u \in \underbrace{\mathcal{U}_0 \times \cdots \times \mathcal{U}_{N-1}}_{\text{easy to project onto (e.g. } u_{\text{MIN}} \le u_t \le u_{\text{MAX}})}$$



Recap

$$\bar{u} = \Pi_{\mathcal{U}}(u - \gamma \nabla f(u))$$

$$u \text{ loc. min.} \quad \Rightarrow \quad u - \bar{u} = 0$$

$$\varphi_\gamma(u) = f(u) + \langle \nabla f(u), \bar{u} - u \rangle + \tfrac{1}{2\gamma} \|\bar{u} - u\|^2$$

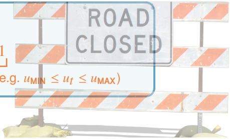
1. φ_γ continuous
2. $\alpha \in (0, 1)$, γ small enough

$$\varphi_\gamma(\bar{u}) \le \varphi_\gamma(u) - \tfrac{1-\alpha}{2\gamma} \|\bar{u} - u\|^2 \quad \forall u$$

Novel speedup

Globalization — A novel nonsmooth LS

$$\operatorname*{minimize}_{u} \ \underbrace{f(u;\, x_0)}_{\text{smooth}} \quad \text{subject to } \ u \in \underbrace{\mathcal{U}_0 \times \cdots \times \mathcal{U}_{N-1}}_{\text{easy to project onto (e.g. } u_{\text{MIN}} \le u_t \le u_{\text{MAX}})}$$



Recap

$$\bar{u} = \Pi_{\mathcal{U}}(u - \gamma \nabla f(u))$$

$$u \text{ loc. min.} \quad \Rightarrow \quad u - \bar{u} = 0$$

$$\varphi_\gamma(u) = f(u) + \langle \nabla f(u), \bar{u} - u \rangle + \tfrac{1}{2\gamma} \|\bar{u} - u\|^2$$

① φ_γ continuous

② $\alpha \in (0, 1)$, γ small enough

$$\varphi_\gamma(\bar{u}) \le \varphi_\gamma(u) - \tfrac{1-\alpha}{2\gamma} \|\bar{u} - u\|^2 \qquad \forall u$$

Novel linesearch

$$\cancel{u^+ = u + \tau d}$$

$$u^+ = (1 - \tau)\bar{u} + \tau(u + d)$$

reducing τ until

$$\varphi_\gamma(u^+) < \varphi_\gamma(u) - \ \ldots \qquad (\star)$$

① + ② ⇒ $(\star)$ passed for small enough τ

Embeddable *ms*-fast NMPC solvers

Embeddable *ms*-fast NMPC solvers

(still free states...)

$$\operatorname*{minimize}_{u} \overbrace{\sum_{t=0}^{N-1} \ell_t(x_t, u_t) + \ell_N(x_N)}^{f(u)=\ell(x(u),u)} \quad \text{subject to } u_t \in \mathcal{U}_t$$

```
Require $\alpha \in (0,1)$, $\gamma > 0$, $\sigma < \frac{1-\alpha}{2\gamma}$, initial $u = (u_0 \dots u_{N-1})$
Iterate   until $\|u - \bar{u}\| \leq \varepsilon_{\mathtt{tol}}$
    1. Compute $\nabla f(u)$
    2. $\bar{u} = \Pi_{\mathcal{U}}(u - \gamma \nabla f(u))$
    3. Choose a direction $d$
    4. $u^+ = (1-\tau)\bar{u} + \tau(u + d)$
       reducing $\tau$ until
       $\varphi_\gamma(u^+) \leq \varphi_\gamma(u) - \sigma\|u - \bar{u}\|^2$
```

Embeddable *ms*-fast NMPC solvers

(still free states...)

$$\underset{u}{\text{minimize}} \ \overbrace{\sum_{t=0}^{N-1} \ell_t(x_t, u_t) + \ell_N(x_N)}^{f(u) = \ell(x(u),u)} \quad \text{subject to}\ u_t \in \mathcal{U}_t$$

Require $\alpha \in (0, 1),\ \gamma > 0,\ \sigma < \frac{1-\alpha}{2\gamma},$ initial $u = (u_0 \ldots u_{N-1})$

Iterate until $\|u - \bar{u}\| \leq \varepsilon_{\texttt{tol}}$

1. Compute $\nabla f(u)$

2. $\bar{u} = \Pi_{\mathcal{U}}(u - \gamma \nabla f(u))$

3. Choose a direction d

4. $u^+ = (1 - \tau)\bar{u} + \tau(u + d)$

reducing τ until

$\varphi_\gamma(u^+) \leq \varphi_\gamma(u) - \sigma\|u - \bar{u}\|^2$

CasADi Automatic Differentiation tool

Optimized C code for (backward) AD

J. Andersson, *A general-purpose software framework for dynamic optimization.* KU Leuven, **2013**

Embeddable *ms*-fast NMPC solvers

(still free states...)

$$\operatorname*{minimize}_{u} \overbrace{\sum_{t=0}^{N-1} \ell_t(x_t, u_t) + \ell_N(x_N)}^{f(u)=\ell(x(u),u)} \quad \text{subject to } u_t \in \mathcal{U}_t$$

Require $\alpha \in (0,1),\ \gamma > 0,\ \sigma < \frac{1-\alpha}{2\gamma},$ initial $u = (u_0 \dots u_{N-1})$

Iterate until $\|u - \bar{u}\| \le \varepsilon_{\mathtt{tol}}$

```
1. Compute $\nabla f(u)$
2. $\bar{u} = \Pi_{\mathcal{U}}(u - \gamma\nabla f(u))$
3. Choose a direction $d$
4. $u^+ = (1-\tau)\bar{u} + \tau(u + d)$
   reducing $\tau$ until
   $\varphi_\gamma(u^+) \le \varphi_\gamma(u) - \sigma\|u - \bar{u}\|^2$
```

$\mathcal{U}_0 \times \cdots \times \mathcal{U}_{N-1}$ is separable

$$\bar{u}_t = \Pi_{\mathcal{U}_t}(u_t - \gamma\nabla_{u_t} f(u))$$

N projections in parallel

Embeddable *ms*-fast NMPC solvers

(still free states...)

$$\operatorname*{minimize}_{u} \overbrace{\sum_{t=0}^{N-1} \ell_t(x_t, u_t) + \ell_N(x_N)}^{f(u)=\ell(x(u),u)} \quad \text{subject to } u_t \in \mathcal{U}_t$$

Require $\alpha \in (0,1),\ \gamma > 0,\ \sigma < \frac{1-\alpha}{2\gamma}$, initial $u = (u_0 \dots u_{N-1})$

Iterate until $\|u - \bar{u}\| \leq \varepsilon_{\mathtt{tol}}$

1. Compute $\nabla f(u)$
2. $\bar{u} = \Pi_{\mathcal{U}}(u - \gamma \nabla f(u))$
3. Choose a direction d
4. $u^+ = (1-\tau)\bar{u} + \tau(u + d)$

   reducing τ until

   $\varphi_\gamma(u^+) \leq \varphi_\gamma(u) - \sigma\|u - \bar{u}\|^2$

Optimality conditions $\mathcal{R}(u) = 0$ where

$$\mathcal{R}(u) := u - \Pi_{\mathcal{U}}(u - \gamma \nabla f(u))$$

Idea: quasi-Newton method on $\mathcal{R}$

e.g., **L-BFGS** (only scalar products)

Embeddable *ms*-fast NMPC solvers

(still free states...)

$$\underset{u}{\text{minimize}} \; \overbrace{\sum_{t=0}^{N-1} \ell_t(x_t, u_t) + \ell_N(x_N)}^{f(u) = \ell(x(u),u)} \quad \text{subject to } u_t \in \mathcal{U}_t$$

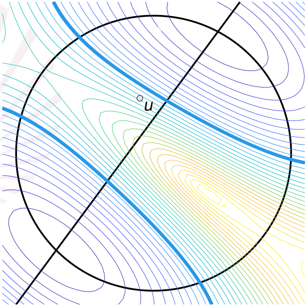
Require $\alpha \in (0,1)$, $\gamma > 0$, $\sigma < \frac{1-\alpha}{2\gamma}$, initial $u = (u_0 \dots u_{N-1})$

Iterate until $\|u - \bar{u}\| \leq \varepsilon_{\mathtt{tol}}$

1. Compute $\nabla f(u)$
2. $\bar{u} = \Pi_{\mathcal{U}}(u - \gamma \nabla f(u))$
3. Choose a direction d
4. $u^+ = (1-\tau)\bar{u} + \tau(u + d)$

   reducing τ until

   $\varphi_\gamma(u^+) \leq \varphi_\gamma(u) - \sigma\|u - \bar{u}\|^2$



Illustrative (toy) example

$f(u) = \frac{1}{2}\,\mathrm{dist}^2(u, l)$

l is a line intersecting a circumference $\mathcal{U}$

Embeddable *ms*-fast NMPC solvers

(still free states...)

$$\underset{u}{\text{minimize}} \; \overbrace{\sum_{t=0}^{N-1} \ell_t(x_t, u_t) + \ell_N(x_N)}^{f(u)=\ell(x(u),u)} \quad \text{subject to } u_t \in \mathcal{U}_t$$

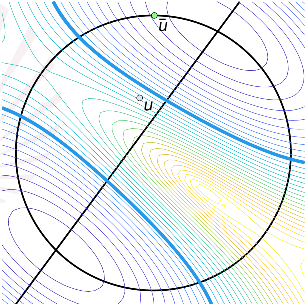
Require $\alpha \in (0,1)$, $\gamma > 0$, $\sigma < \frac{1-\alpha}{2\gamma}$, initial $u = (u_0 \dots u_{N-1})$

Iterate until $\|u - \bar{u}\| \leq \varepsilon_{\mathtt{tol}}$

1. Compute $\nabla f(u)$
2. $\bar{u} = \Pi_{\mathcal{U}}(u - \gamma \nabla f(u))$
3. Choose a direction d
4. $u^{+} = (1-\tau)\bar{u} + \tau(u + d)$

   reducing τ until

   $\varphi_\gamma(u^{+}) \leq \varphi_\gamma(u) - \sigma\|u - \bar{u}\|^2$



Illustrative (toy) example

$f(u) = \frac{1}{2}\,\mathrm{dist}^2(u, l)$

l is a line intersecting a circumference $\mathcal{U}$

Embeddable *ms*-fast NMPC solvers

(still free states...)

$$\underset{u}{\text{minimize}} \overbrace{\sum_{t=0}^{N-1} \ell_t(x_t, u_t) + \ell_N(x_N)}^{f(u)=\ell(x(u),u)} \quad \text{subject to } u_t \in \mathcal{U}_t$$

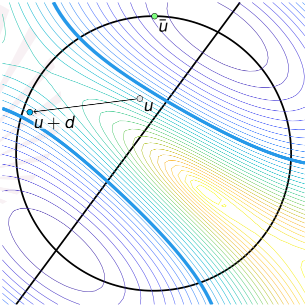
Require $\alpha \in (0,1)$, $\gamma > 0$, $\sigma < \frac{1-\alpha}{2\gamma}$, initial $u = (u_0 \dots u_{N-1})$

Iterate until $\|u - \bar{u}\| \leq \varepsilon_{\mathtt{tol}}$

1. Compute $\nabla f(u)$
2. $\bar{u} = \Pi_{\mathcal{U}}(u - \gamma \nabla f(u))$
3. Choose a direction d
4. $u^+ = (1-\tau)\bar{u} + \tau(u+d)$

   reducing τ until

   $\varphi_\gamma(u^+) \leq \varphi_\gamma(u) - \sigma\|u - \bar{u}\|^2$



Illustrative (toy) example

$f(u) = \frac{1}{2}\,\text{dist}^2(u, l)$

l is a line intersecting a circumference $\mathcal{U}$

Embeddable *ms*-fast NMPC solvers

(still free states...)

$$\underset{u}{\text{minimize}} \ \overbrace{\sum_{t=0}^{N-1} \ell_t(x_t, u_t) + \ell_N(x_N)}^{f(u)=\ell(x(u),u)} \quad \text{subject to } u_t \in \mathcal{U}_t$$

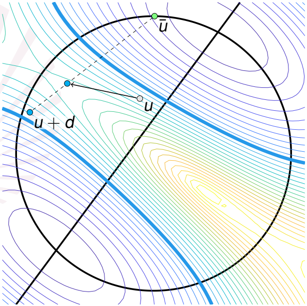
Require $\alpha \in (0,1)$, $\gamma > 0$, $\sigma < \frac{1-\alpha}{2\gamma}$, initial $u = (u_0 \ldots u_{N-1})$

Iterate until $\|u - \bar{u}\| \leq \varepsilon_{\texttt{tol}}$

1. Compute $\nabla f(u)$
2. $\bar{u} = \Pi_{\mathcal{U}}(u - \gamma \nabla f(u))$
3. Choose a direction d
4. $u^+ = (1-\tau)\bar{u} + \tau(u + d)$

   reducing τ until

   $\varphi_\gamma(u^+) \leq \varphi_\gamma(u) - \sigma\|u - \bar{u}\|^2$



Illustrative (toy) example

$f(u) = \frac{1}{2}\,\mathrm{dist}^2(u, l)$

l is a line intersecting a circumference $\mathcal{U}$

Embeddable *ms*-fast NMPC solvers

(still free states...)

$$\underset{u}{\text{minimize}} \; \overbrace{\sum_{t=0}^{N-1} \ell_t(x_t, u_t) + \ell_N(x_N)}^{f(u) = \ell(x(u),u)} \quad \text{subject to } u_t \in \mathcal{U}_t$$

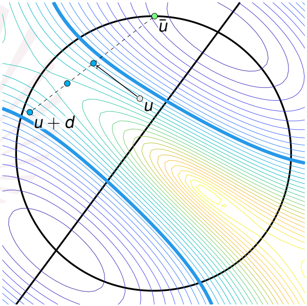
Require $\alpha \in (0, 1)$, $\gamma > 0$, $\sigma < \frac{1-\alpha}{2\gamma}$, initial $u = (u_0 \dots u_{N-1})$

Iterate until $\|u - \bar{u}\| \leq \varepsilon_{\mathtt{tol}}$

1. Compute $\nabla f(u)$
2. $\bar{u} = \Pi_{\mathcal{U}}(u - \gamma \nabla f(u))$
3. Choose a direction d
4. $u^+ = (1 - \tau)\bar{u} + \tau(u + d)$

   reducing τ until

   $\varphi_\gamma(u^+) \leq \varphi_\gamma(u) - \sigma\|u - \bar{u}\|^2$



Illustrative (toy) example

$f(u) = \frac{1}{2}\,\mathrm{dist}^2(u, l)$

l is a line intersecting a circumference $\mathcal{U}$

Embeddable *ms*-fast NMPC solvers

(still free states...)

$$\underset{u}{\text{minimize}} \; \overbrace{\sum_{t=0}^{N-1} \ell_t(x_t, u_t) + \ell_N(x_N)}^{f(u) = \ell(x(u),u)} \quad \text{subject to } u_t \in \mathcal{U}_t$$

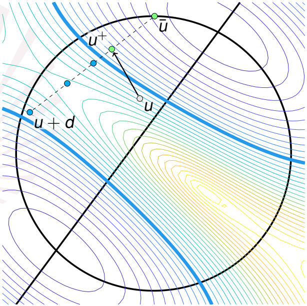
Require $\alpha \in (0,1)$, $\gamma > 0$, $\sigma < \frac{1-\alpha}{2\gamma}$, initial $u = (u_0 \dots u_{N-1})$

Iterate until $\|u - \bar{u}\| \leq \varepsilon_{\mathtt{tol}}$

1. Compute $\nabla f(u)$
2. $\bar{u} = \Pi_{\mathcal{U}}(u - \gamma \nabla f(u))$
3. Choose a direction d
4. $u^+ = (1 - \tau)\bar{u} + \tau(u + d)$

   reducing τ until

   $\varphi_\gamma(u^+) \leq \varphi_\gamma(u) - \sigma\|u - \bar{u}\|^2$



Illustrative (toy) example

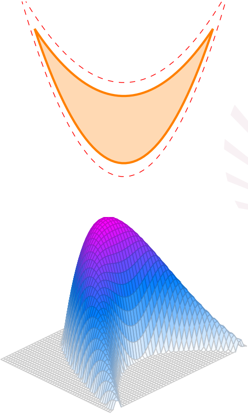
$f(u) = \frac{1}{2} \operatorname{dist}^2(u, l)$

l is a line intersecting a circumference $\mathcal{U}$

Embeddable *ms*-fast NMPC solvers

Handling state constraints

- novel obstacle avoidance constraints encoding
- uses a single equality constraint!



$\psi(x) = \prod_{i=1}^{m} \max\{h_i(x), 0\}$

$$x \notin \mathcal{O} := \{z \mid h_i(x) > 0,\ i = 1, \ldots, m\}$$

$$\Updownarrow$$

$$\exists i \quad \text{such that} \quad h_i(x) \leq 0$$

$$\Updownarrow$$

$$\exists i \quad \text{such that} \quad \max\{h_i(x), 0\} = 0$$

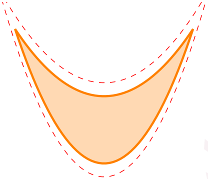
$$\Updownarrow$$

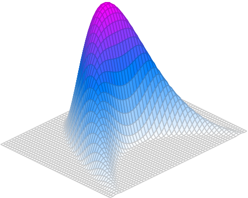
$$\psi(x) := \prod_{i=1}^{m} \max\{h_i(x), 0\} = 0$$

Embeddable *ms*-fast NMPC solvers

Handling state constraints

- novel obstacle avoidance constraints encoding
- uses a single equality constraint!





$\psi^2(x) = \prod_{i=1}^{m} \max\{h_i(x), 0\}^2$

$$x \notin \mathcal{O} := \{z \mid h_i(x) > 0,\ i = 1, \dots, m\}$$

$$\Updownarrow$$

$$\exists i \quad \text{such that} \quad h_i(x) \leq 0$$

$$\Updownarrow$$

$$\exists i \quad \text{such that} \quad \max\{h_i(x), 0\} = 0$$

$$\Updownarrow$$

$$\psi(x) := \prod_{i=1}^{m} \max\{h_i(x), 0\} = 0$$

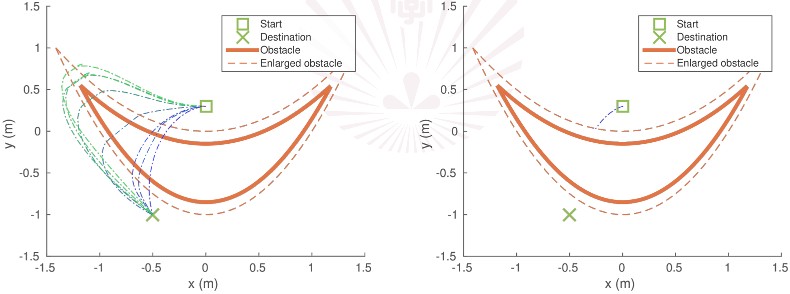
Quadratic penalty

Soften obstacle avoidance by adding $\mu\psi^2$ to cost function ($\mu > 0$)

Embeddable *ms*-fast NMPC solvers

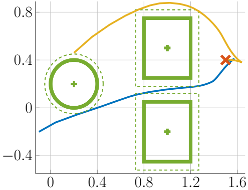
Handling state constraints

- **Quadratic penalty method:** gradually increase penalty μ
- Solve subproblems, **warm starting** with previous solution
- **Helps avoiding local minima** (getting stuck to obstacles)



Embeddable *ms*-fast NMPC solvers

Comparisons — Obstacle avoidance



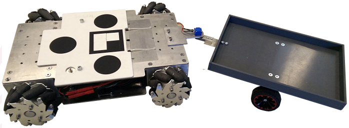
Goals

- steer vehicle to reference pos./orient. $(p_\star, \vartheta_\star)$
- avoid obstacles

Nonlinear system

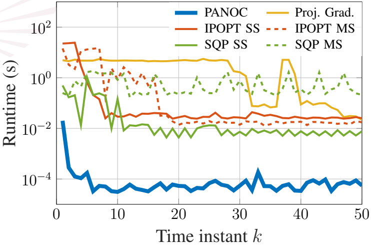
$$\begin{cases} \dot{p}_x = u_x + L\dot{\vartheta}\sin\vartheta \\ \dot{p}_y = u_y - L\dot{\vartheta}\cos\vartheta \\ \dot{\vartheta} = \frac{1}{L}(u_y\cos\vartheta - u_x\sin\vartheta) \end{cases}$$

u: velocity
p: position
ϑ: head. angle



Implementation

- discretized with RK4
- horizon $N = 50$
- 10Hz NMPC control rate
- $\|u\|_\infty \leq 0.8 m/s$
- soft-constrained enlarged obstacles with adaptive penalty



Conclusions

Conclusions

The end of the journey

We ended up explaining how **PANOC** algorithm works

- Efficient, **QP-free**, NMPC line-search algorithm
- Give it a try
  - Embeddable NMPC `C` code generator (`Matlab` & `Python` interfaces)
    `https://github.com/kul-optec/nmpc-codegen`
  - Standalone `Julia` version
    `https://github.com/kul-optec/PANOC.jl`
- More than NMPC:
  engine of **generic optimization solvers**
  - OpEn (embedded Optimization Engine)
    `https://alphaville.github.io/optimization-engine/`
  - ALM solver
    `https://github.com/tttapa/PANOC-ALM`

Lorenzo **Stella**
Amazon Berlin

Pantelis **Sopasakis**
QU Belfast

Panos **Patrinos**
KU Leuven

Conclusions

Take-home message

Fast

$\not\Downarrow$

high-computing

+ *new*

Old ^ is **gold**

More info on PANOC (shameful self-advertisement) 😅

- AT, L. Stella and P. Patrinos, *Forward-backward envelope for the sum of two nonconvex functions: Further properties and nonmonotone linesearch algorithms,* SIAM J Opt **28**(3):2274-2303, **2018**
- L. Stella, AT, P. Sopasakis and P. Patrinos, *A simple and efficient algorithm for nonlinear model predictive control,* In: *IEEE 56th CDC*, **2017**
- A. Sathya, P. Sopasakis, R. Van Parys, AT, G. Pipeleers and P. Patrinos, *Embedded nonlinear model predictive control for obstacle avoidance using PANOC,* In: *IEEE ECC*, Jun **2018**

Conclusions

Take-home message

Fast

$\not\Downarrow$

high-computing

Thank you!

+ *new*

Old ^ is **gold**

More info on PANOC (shameful self-advertisement) 😅

- AT, L. Stella and P. Patrinos, *Forward-backward envelope for the sum of two nonconvex functions: Further properties and nonmonotone linesearch algorithms,* SIAM J Opt **28**(3):2274-2303, **2018**
- L. Stella, AT, P. Sopasakis and P. Patrinos, *A simple and efficient algorithm for nonlinear model predictive control,* In: *IEEE 56th CDC*, **2017**
- A. Sathya, P. Sopasakis, R. Van Parys, AT, G. Pipeleers and P. Patrinos, *Embedded nonlinear model predictive control for obstacle avoidance using PANOC,* In: *IEEE ECC*, Jun **2018**