AGIMUS

INNOVATIVE ROBOTICS FOR AGILE PRODUCTION

AGIMUS WINTER SCHOOL 2023

OPTIMIZATION AND OPTIMAL CONTROL II

Wilson Jallet

LAAS-CNRS, Gepetto team / Inria, Willow team

Overview

1. Constrained optimization

    A kind refresher

    ProxQP – AL methods applied to QPs

    Augmented Lagrangians for general NLPs

2. Constrained trajectory optimization

    Problem definition

    Augmented Lagrangian trajectory optimization with ProxDDP

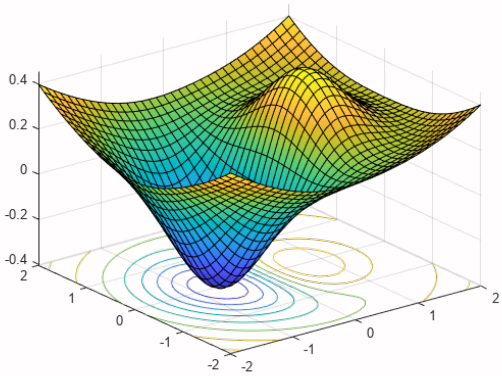
Foreword

The goal of this presentation is to (re)familiarize yourself with concepts from **CONSTRAINED OPTIMIZATION** and its difficulties.

We will talk of **NONLINEAR PROGRAMS** (NLPs) in general and apply the concepts of proximal methods to tackle them, first in the quadratic programming and later for optimal control.

Constrained optimization

A kind refresher



Unconstrained optimization

Unconstrained optimization: only needs an objective function $c : \mathbb{R}^{n_x} \to \mathbb{R}$. The problem is simply:

$$\underset{x \in \mathbb{R}^{n_x}}{\text{minimize}}\, c(x). \tag{1}$$

A point $x^\star \in \mathbb{R}^{n_x}$ is a **LOCAL MINIMIZER** if

$$\text{for all } x' \text{ in a neighborhood of } x^\star,\ f(x^\star) \leq f(x')$$

and a *strict* local min. if $f(x^\star) < f(x')$ for $x' \neq x^\star$.

Remark

$c(x) \in \mathbb{R} \Rightarrow$ there are no implicit constraints (as introduced in Adrien's talk)

Recall – Global minima and convexity

A point $x^\star$ is a **GLOBAL MINIMUM** if for *all* $x' \in \mathbb{R}^n$, $f(x^\star) \leq f(x')$.

When does local imply global?

Recall – Global minima and convexity

A point $x^\star$ is a **GLOBAL MINIMUM** if for *all* $x' \in \mathbb{R}^n$, $f(x^\star) \leq f(x')$.

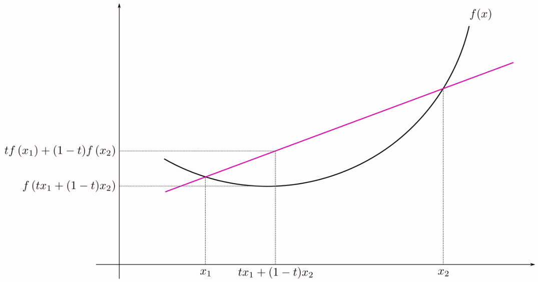
When does local imply global? When the function f is **CONVEX**:

Definition (Convexity)

f is called *convex* when for any x, y and $t \in [0, 1]$,

$$f(tx + (1-t)y) \leq tf(x) + (1-t)f(y).$$

Strictly convex when for $x \neq y$ and $t \in (0, 1)$, the inequality is strict.



Recall – Global minima and convexity

A point $x^\star$ is a **GLOBAL MINIMUM** if for *all* $x' \in \mathbb{R}^n$, $f(x^\star) \leq f(x')$.

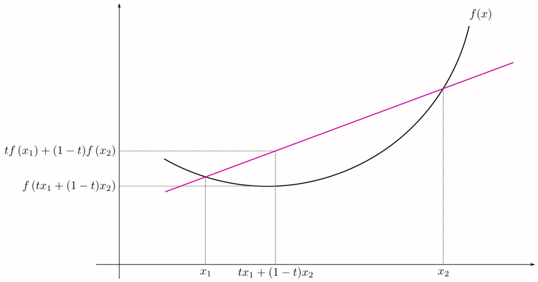
When does local imply global? When the function f is **CONVEX**:

Definition (Convexity)

f is called *convex* when for any x, y and $t \in [0, 1]$,

$$f(tx + (1-t)y) \leq tf(x) + (1-t)f(y).$$

Strictly convex when for $x \neq y$ and $t \in (0, 1)$, the inequality is strict.



Alternative characterization: if f has second derivatives, when $\nabla^2 f \succeq 0$ ($\succ 0$ for *strict* convexity).

Finding local minima – Necessary conditions of optimality

Question: how do we know a point $x^\star \in \mathbb{R}^{n_x}$ is a (local) minimizer?

STATIONARITY CONDITIONS: if $x^\star$ is a *local optimum*, then $x^\star$ **is an optimum along any line**:

$$\text{for all } v \in \mathbb{R}^{n_x},\ \left.\frac{d}{dt}(c(x^\star + tv))\right|_{t=0} = \langle v, \nabla c(x^\star)\rangle = 0, \tag{2}$$

i.e. the *first-order condition*:

$$\boxed{\nabla c(x^\star) = 0.} \tag{3}$$

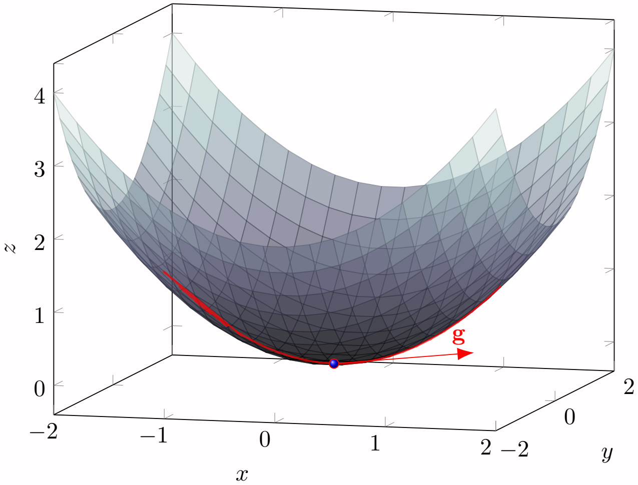


Figure 1: At the minimum, the tangent vectors to the graph **g** are flat – i.e. they are all of the form $(g_x, g_y, 0)$.

Constrained optimization

Consider the (smooth) *constrained* minimization problem

$$\min_{x \in \mathbb{R}^{n_x}} c(x) \tag{4a}$$

$$\text{s.t. } g(x) = 0 \tag{4b}$$

$$h(x) \leq 0. \tag{4c}$$

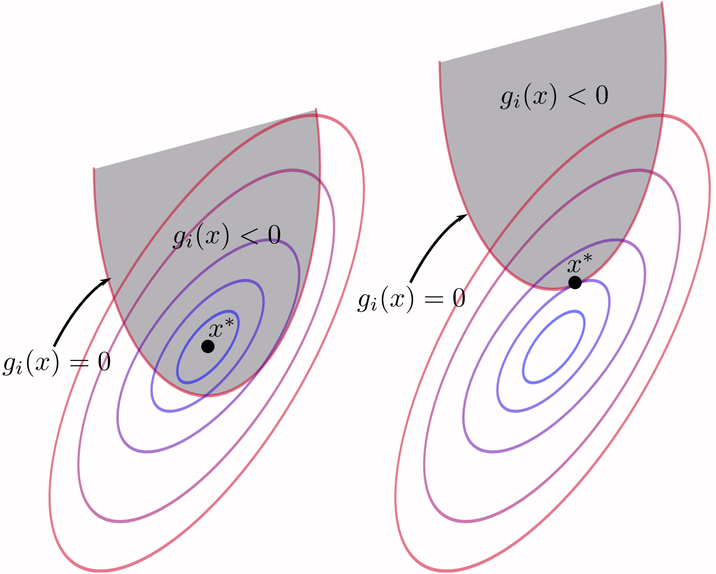


diagram source: Wikipedia

Necessary conditions

Given by the **KKT CONDITIONS**: a point $x^\star \in \mathbb{R}^{n_x}$ is a **LOCAL MINIMIZER** if there are **LAGRANGE MULTIPLIERS** $(y^\star, z^\star) \in \mathbb{R}^{n_g} \times \mathbb{R}_+^{n_h}$ satisfying

$$\nabla c(x^\star) + \partial_x g(x^\star)^\top y^\star + \partial_x h(x^\star)^\top z^\star = 0 \quad \text{(stationarity)} \tag{5a}$$

$$g(x^\star) = 0 \quad \text{(eq. constraint)} \tag{5b}$$

$$h(x^\star) \leq 0 \quad \text{(ineq. constraint)} \tag{5c}$$

$$h(x^\star) \odot z^\star = 0 \ (h_i z_i = 0) \quad \text{(complementarity)} \tag{5d}$$

Necessary conditions

Given by the **KKT CONDITIONS**: a point $x^\star \in \mathbb{R}^{n_x}$ is a **LOCAL MINIMIZER** if there are **LAGRANGE MULTIPLIERS** $(y^\star, z^\star) \in \mathbb{R}^{n_g} \times \mathbb{R}_+^{n_h}$ satisfying

$$\nabla c(x^\star) + \partial_x g(x^\star)^\top y^\star + \partial_x h(x^\star)^\top z^\star = 0 \quad \text{(stationarity)} \tag{5a}$$

$$g(x^\star) = 0 \quad \text{(eq. constraint)} \tag{5b}$$

$$h(x^\star) \leq 0 \quad \text{(ineq. constraint)} \tag{5c}$$

$$h(x^\star) \odot z^\star = 0 \ (h_i z_i = 0) \quad \text{(complementarity)} \tag{5d}$$

Equation (5a) above is the gradient of the classical **LAGRANGIAN FUNCTION** (Rockafellar 1997)

$$\mathscr{L}(x, y, z) = c(x) + y^\top g(x) + z^\top h(x). \tag{6}$$

Necessary conditions

Given by the **KKT CONDITIONS**: a point $x^\star \in \mathbb{R}^{n_x}$ is a **LOCAL MINIMIZER** if there are **LAGRANGE MULTIPLIERS** $(y^\star, z^\star) \in \mathbb{R}^{n_g} \times \mathbb{R}_+^{n_h}$ satisfying

$$\nabla c(x^\star) + \partial_x g(x^\star)^\top y^\star + \partial_x h(x^\star)^\top z^\star = 0 \quad \text{(stationarity)} \tag{5a}$$

$$g(x^\star) = 0 \quad \text{(eq. constraint)} \tag{5b}$$

$$h(x^\star) \leq 0 \quad \text{(ineq. constraint)} \tag{5c}$$

$$h(x^\star) \odot z^\star = 0 \ (h_i z_i = 0) \quad \text{(complementarity)} \tag{5d}$$

Equation (5a) above is the gradient of the classical **LAGRANGIAN FUNCTION** (Rockafellar 1997)

$$\mathscr{L}(x, y, z) = c(x) + y^\top g(x) + z^\top h(x). \tag{6}$$

(5d) are called the **COMPLEMENTARITY CONDITIONS**. The set of i such that $z_i^\star > 0$ $(h_i(x^\star) = 0)$ is called the **ACTIVE SET OF CONSTRAINTS**.

Some things which are NLPs:

- quadratic programs (QPs), among which linear-quadratic (LQ) control problems
- contact problems (see Quentin's stuff)
- collision detection (talk to Louis)
- inverse kinematics
- others?...

Some things which are NLPs:

- quadratic programs (QPs), among which linear-quadratic (LQ) control problems
- contact problems (see Quentin's stuff)
- collision detection (talk to Louis)
- inverse kinematics
- others?...

Convex?

- if f, h is convex, and g is affine (sorry)

As Adrien pointed out, **general nonlinear programming is hard**.

Many methods exist:

- straight **sequential quadratic programming** (SQP), solving a cascade of inequality-QPs with linesearch/filter/trust-region strategies, see SNOPT (Gill *et al.* 2002)
- **interior-point methods**: add barrier for inequalities then move to equality-SQP, see IPOPT (Wächter and Biegler 2006)
- **augmented Lagrangian** methods, with second-order approaches e.g. LANCELOT (A. R. Conn *et al.* 2010)

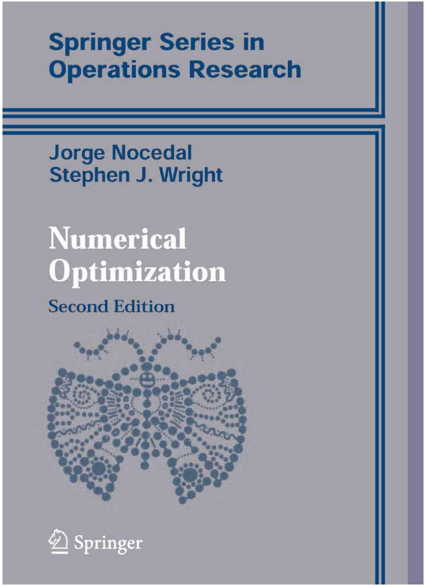


Figure 2: The holy book: *Numerical Optimization* (Nocedal and Wright 2006)

Constrained optimization

ProxQP – AL methods applied to QPs

Equality-constrained QPs (EQPs)

The problem. Let $Q \in \mathbf{S}_n^+(\mathbb{R})$, $q \in \mathbb{R}^n$, $A \in \mathbb{R}^{m \times n}$ and $b \in \mathbb{R}^m$. We consider the simple equality-constrained QP

$$\begin{aligned} \min_x \; & \frac{1}{2} x^\top Q x + q^\top x \\ \text{s.t. } & Ax + b = 0 \end{aligned} \tag{EQP}$$

Lagrangian:

$$\mathcal{L}(x, y) = \tfrac{1}{2} x^\top Q x + q^\top x + y^\top (Ax + b). \tag{7}$$

Equality-constrained QPs (EQPs)

The problem. Let $Q \in \mathbf{S}_n^+(\mathbb{R})$, $q \in \mathbb{R}^n$, $A \in \mathbb{R}^{m \times n}$ and $b \in \mathbb{R}^m$. We consider the simple equality-constrained QP

$$\begin{aligned} \min_x \ & \frac{1}{2} x^\top Q x + q^\top x \\ \text{s.t. } & Ax + b = 0 \end{aligned} \tag{EQP}$$

Lagrangian:

$$\mathcal{L}(x, y) = \tfrac{1}{2} x^\top Q x + q^\top x + y^\top (Ax + b). \tag{7}$$

KKT conditions. Very classically:

$$\begin{bmatrix} Q & A^\top \\ A & \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = - \begin{bmatrix} q \\ b \end{bmatrix}. \tag{8}$$

Unique solution iff matrix is invertible.

Unique solution $(x^\star, y^\star)$ iff KKT matrix $\begin{bmatrix} Q & A^\top \\ A & \end{bmatrix}$ is invertible.

Proposition (see Nocedal and Wright 2006, chap. 16)

The KKT matrix is nonsingular if:

- **LICQ** (*linear independence constraint qualification*) i.e. linear independence of rows of A
- if Z basis matrix $\ker(A)$ (i.e. Z full rank, $AZ = 0$), then $Z^\top Q Z \succ 0$.

Unique solution $(x^\star, y^\star)$ iff KKT matrix $\begin{bmatrix} Q & A^\top \\ A & \end{bmatrix}$ is invertible.

Proposition (see Nocedal and Wright 2006, chap. 16)

The KKT matrix is nonsingular if:

- **LICQ** (*linear independence constraint qualification*) i.e. linear independence of rows of A
- if Z basis matrix $\ker(A)$ (i.e. Z full rank, $AZ = 0$), then $Z^\top Q Z \succ 0$. (*Strict convexity* ($Q \succ 0$) is sufficient[1])

[1] Even required by some solvers e.g. QUADPROG (https://github.com/quadprog/quadprog) based on Goldfarb and Idnani 1983 Goldfarb and Idnani 1983

Unique solution $(x^\star, y^\star)$ iff KKT matrix $\begin{bmatrix} Q & A^\top \\ A & \end{bmatrix}$ is invertible.

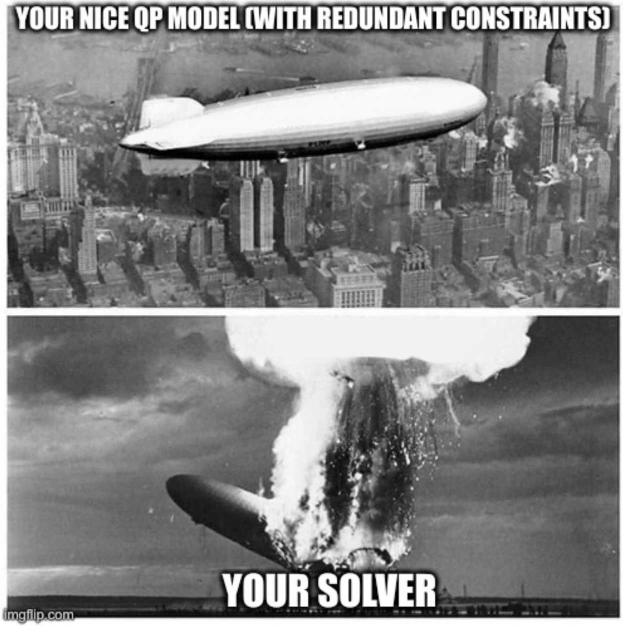
Proposition (see Nocedal and Wright 2006, chap. 16)

The KKT matrix is nonsingular if:

- **LICQ** (*linear independence constraint qualification*) i.e. linear independence of rows of A
- if Z basis matrix $\ker(A)$ (i.e. Z full rank, $AZ = 0$), then $Z^\top Q Z \succ 0$. (*Strict convexity* ($Q \succ 0$) is sufficient[1])

In practice: not very fun! (no redundant constraints)

[1]Even required by some solvers e.g. QUADPROG (`https://github.com/quadprog/quadprog`) based on Goldfarb and Idnani 1983 Goldfarb and Idnani 1983



AGIMUS
INNOVATIVE ROBOTICS FOR AGILE PRODUCTION

Method of multipliers for rank-deficient EQPs

Redundant constraints? Augmented Lagrangians (AL) to the rescue!

The primal way. Let $\mu > 0$. The AL associated with (EQP) is the quadratic

$$\mathcal{L}_\mu(x; y_e) \stackrel{\text{def}}{=} \frac{1}{2} x^\top Q x + q^\top x + y_e^\top (Ax + b) + \tfrac{1}{2\mu} \|Ax + b\|_2^2$$

Method of multipliers for rank-deficient EQPs

Redundant constraints? Augmented Lagrangians (AL) to the rescue!

The primal way. Let $\mu > 0$. The AL associated with (EQP) is the quadratic

$$\begin{aligned} \mathcal{L}_\mu(x; y_e) &\stackrel{\text{def}}{=} \frac{1}{2} x^\top Q x + q^\top x + y_e^\top (Ax + b) + \tfrac{1}{2\mu} \|Ax + b\|_2^2 \\ &= -\underbrace{\min_y \left\{ -\mathcal{L}(x, y) + \tfrac{\mu}{2} \|y - y_e\|_2^2 \right\}}_{\textit{proximal!}} \end{aligned} \tag{9}$$

Method of multipliers for rank-deficient EQPs

Redundant constraints? Augmented Lagrangians (AL) to the rescue!

The primal way. Let $\mu > 0$. The AL associated with (EQP) is the quadratic

$$\begin{aligned} \mathcal{L}_\mu(x; y_e) &\overset{\text{def}}{=} \frac{1}{2}x^\top Qx + q^\top x + y_e^\top(Ax + b) + \tfrac{1}{2\mu}\|Ax + b\|_2^2 \\ &= -\underbrace{\min_y \left\{-\mathcal{L}(x, y) + \tfrac{\mu}{2}\|y - y_e\|_2^2\right\}}_{\textit{proximal!}} \end{aligned} \tag{9}$$

Method of multipliers. Minimum given by $\nabla_x \mathcal{L}_\mu(x^+; y_e) = 0$ i.e.

$$(Q + \tfrac{1}{\mu}A^\top A)x^+ = -[q + A^\top(y_e + \tfrac{1}{\mu}b)] \tag{10}$$

and dual step $y^+ = y_e + \frac{1}{\mu}(Ax^+ + b)$.

Set $x \leftarrow x^+$, $y_e \leftarrow y^+$, **rinse and repeat.**

Method of multipliers for rank-deficient EQPs

Redundant constraints? Augmented Lagrangians (AL) to the rescue!

The primal way. Let $\mu > 0$. The AL associated with (EQP) is the quadratic

$$\begin{aligned}\mathcal{L}_\mu(x; y_e) &\overset{\text{def}}{=} \frac{1}{2}x^\top Qx + q^\top x + y_e^\top(Ax + b) + \tfrac{1}{2\mu}\|Ax + b\|_2^2 \\ &= -\underbrace{\min_y \left\{-\mathcal{L}(x, y) + \tfrac{\mu}{2}\|y - y_e\|_2^2\right\}}_{\textit{proximal!}}\end{aligned} \tag{9}$$

Method of multipliers. Minimum given by $\nabla_x \mathcal{L}_\mu(x^+; y_e) = 0$ i.e.

$$(Q + \tfrac{1}{\mu}A^\top A)x^+ = -[q + A^\top(y_e + \tfrac{1}{\mu}b)] \tag{10}$$

and dual step $y^+ = y_e + \frac{1}{\mu}(Ax^+ + b)$.
Set $x \leftarrow x^+$, $y_e \leftarrow y^+$, **rinse and repeat.**

Caveat: *bad* numerical conditioning (matrix eigenvalues might span a large range of values e.g. 10^{-6} to 10^6)

Primal-dual/saddle-point view. Introduces a regularized KKT matrix:

$$\begin{bmatrix} Q & A^\top \\ A & -\mu I \end{bmatrix} \begin{bmatrix} x^+ \\ y^+ \end{bmatrix} = - \begin{bmatrix} q \\ b + \mu y_e \end{bmatrix} \tag{11}$$

Remark

- μ controls convergence speed $\rightarrow$ lower is faster (but less stable)
- clever heuristics for $\{\mu_k\}$ for good compromises e.g. BCL (A. Conn *et al.* 1991)

Further explored in the practical session!

A link through linear algebra with **Schur complements**:

$$Q + \frac{1}{\mu} A^\top A \xleftrightarrow{\text{Schur compl.}} \begin{bmatrix} Q & A^\top \\ A & -\mu I \end{bmatrix} \xleftrightarrow{\text{Schur compl.}} \mu I + A Q^{-1} A^\top \qquad (12)$$

2nd variant is similar to Goldfarb and Idnani 1983, also used in Carpentier *et al.* 2021 (RSS).

Robotics: Science and Systems 2021
Held Virtually, July 12–16, 2021

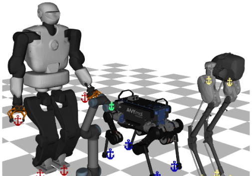
Proximal and Sparse Resolution of Constrained Dynamic Equations

Justin Carpentier
Inria, École normale supérieure
CNRS, PSL Research University
75005 Paris, France
Email: justin.carpentier@inria.fr

Rohan Budhiraja
Inria Paris
75012 Paris, France
Email: rohan.budhiraja@inria.fr

Nicolas Mansard
LAAS-CNRS, ANITI
University of Toulouse
31400 Toulouse, France
Email: nicolas.mansard@laas.fr

***Abstract*—Control of robots with kinematic constraints like loop-closure constraints or interactions with the environment requires solving the underlying constrained dynamics equations of motion. Several approaches have been proposed so far in the literature to solve these constrained optimization problems, for instance by either taking advantage in part of the sparsity of the kinematic tree or by considering an explicit formulation of the constraints in the problem resolution. Yet, not all the constraints allow an explicit formulation and in general, approaches of the state of the art suffer from singularity issues, especially in the context of redundant or singular constraints. In this paper, we propose a unified approach to solve forward dynamics equations involving constraints in an efficient, generic and robust manner. To this aim, we first (i) propose a proximal formulation of the constrained dynamics which converges to an optimal solution in the least-square sense even in the presence of singularities. Based on this proximal formulation, we introduce (ii) a sparse**



Inequality-constrained QPs

A (slightly?) harder problem:

$$\min_x \frac{1}{2} x^\top Q x + q^\top x \tag{13a}$$

$$\text{s.t. } Ax + b = 0 \tag{13b}$$

$$Cx + u \leq 0 \tag{13c}$$

Inequality-constrained QPs

A (slightly?) harder problem:

$$\min_x \frac{1}{2} x^\top Q x + q^\top x \quad (13a)$$

$$\text{s.t. } Ax + b = 0 \quad (13b)$$

$$Cx + u \leq 0 \quad (13c)$$

Actually **WAY HARDER**.

KKT CONDITIONS are like before, plus the complementarity:

$$Qx + q + A^\top y + C^\top z = 0 \quad (14a)$$

$$Ax + b = 0 \quad (14b)$$

$$Cx + u \leq 0 \quad (14c)$$

$$z \odot [Cx + u] = 0 \quad (14d)$$

AGIMUS
INNOVATIVE ROBOTICS FOR AGILE PRODUCTION

Inequality-constrained QPs

A (slightly?) harder problem:

$$\min_x \frac{1}{2} x^\top Q x + q^\top x \tag{13a}$$
$$\text{s.t. } Ax + b = 0 \tag{13b}$$
$$Cx + u \leq 0 \tag{13c}$$

KKT CONDITIONS are like before, plus the complementarity:

$$Qx + q + A^\top y + C^\top z = 0 \tag{14a}$$
$$Ax + b = 0 \tag{14b}$$
$$Cx + u \leq 0 \tag{14c}$$
$$z \odot [Cx + u] = 0 \tag{14d}$$

Actually **WAY HARDER**. Many methods employed for this:

- dual method (*strictly convex*) AKA Goldfarb and Idnani 1983 AKA quadprog
- solve EQP + ADMM (see the OSQP solver (Stellato *et al.* 2020))
- active-set search ~~sorcery~~ (solver: qpOASES (Ferreau *et al.* 2014))
- and AL! See QPALM (Hermans *et al.* 2019), QPDO (De Marchi 2022) and ours, **ProxQP** (Bambade *et al.* 2023)

AL for inequality-constrained problems

(Generalized) AL function. (see Rockafellar 1976)

$$\mathcal{L}_\mu(x; y_e, z_e) = \frac{1}{2}x^\top Qx + q^\top x + \underbrace{y_e^\top(Ax+b) + \frac{1}{2\mu}\|Ax+b\|_2^2}_{\text{equality penalty}} + \underbrace{\frac{1}{2\mu}\|[Cx+u+\mu z_e]_+\|_2^2 - \frac{\mu}{2}\|z_e\|_2^2}_{\text{inequality penalty}}. \quad (15)$$

AL for inequality-constrained problems

(Generalized) AL function. (see Rockafellar 1976)

$$\mathcal{L}_\mu(x; y_e, z_e) = \frac{1}{2}x^\top Q x + q^\top x + \underbrace{y_e^\top (Ax + b) + \frac{1}{2\mu}\|Ax + b\|_2^2}_{\text{equality penalty}} + \underbrace{\frac{1}{2\mu}\|[Cx + u + \mu z_e]_+\|_2^2 - \frac{\mu}{2}\|z_e\|_2^2}_{\text{inequality penalty}}. \tag{15}$$

Terrible news!

- *Not* quadratic anymore – just piecewise.
- **No closed-form minimum.**
- **Not even smooth!**

AL for inequality-constrained problems

(Generalized) AL function. (see Rockafellar 1976)

$$\mathcal{L}_\mu(x; y_e, z_e) = \frac{1}{2}x^\top Qx + q^\top x + \underbrace{y_e^\top(Ax+b) + \frac{1}{2\mu}\|Ax+b\|_2^2}_{\text{equality penalty}} + \underbrace{\frac{1}{2\mu}\|[Cx+u+\mu z_e]_+\|_2^2 - \frac{\mu}{2}\|z_e\|_2^2}_{\text{inequality penalty}}. \tag{15}$$

Terrible news!

- *Not* quadratic anymore – just piecewise.
- **No closed-form minimum.**
- **Not even smooth!**

Methods such as **ProxQP** and QPALM $\rightarrow$ **inexact minimization using semi-smooth Newton methods** (not covered in this session).

In general, these methods are **difficult to implement**, especially with **performance** in mind.

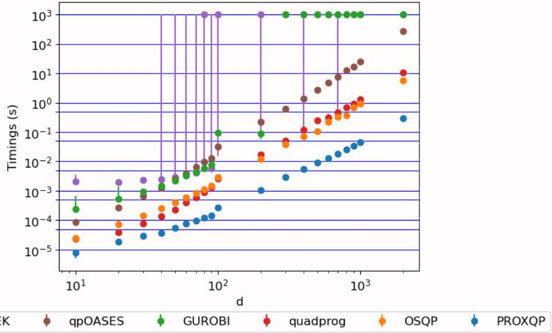
AGIMUS
INNOVATIVE ROBOTICS FOR AGILE PRODUCTION

In general, these methods are **difficult to implement**, especially with **performance** in mind.

Try our solver!

ProxSuite

THE ADVANCED PROXIMAL OPTIMIZATION TOOLBOX



`conda install -c conda-forge proxsuite`

AGIMUS

INNOVATIVE ROBOTICS FOR AGILE PRODUCTION

Constrained optimization

Augmented Lagrangians for general NLPs

Assume your initial problem is **not** a QP (i.e. nonquadratic $c(z)$, nonlinear constraints...).

AL method is still posed as the iteration:

1. minimize the AL function (**HOW?**)

$$\mathcal{L}_\mu(x; y_e, z_e) = c(x) + \frac{1}{2\mu}\|g(x) + \mu y_e\|^2 + \frac{1}{2\mu}\|[h(x) + \mu z_e]_+\|^2$$

2. update multipliers:

$$y^+ = y_e + \tfrac{1}{\mu} g(x^+), \; z^+ = [z_e + \tfrac{1}{\mu} h(x^+)]_+ \tag{16}$$

3. update μ maybe

Constrained trajectory optimization

Problem definition

Constrained trajectory optimization – continuous time

Our objective, in continuous time, is to solve trajectory optimization problem of the form

$$\min_{x,u} \int_0^T \ell(t, x(t), u(t))\,\mathrm{d}t + \ell_T(x(T)) \tag{17a}$$

$$\text{s.t. } \dot{x}(t) = f(t, x(t), u(t)) \tag{17b}$$

$$h(t, x(t), u(t)) \leq 0 \tag{17c}$$

$$h_T(x(T)) \leq 0. \tag{17d}$$

UR10 ballistics video

Quadrotor slalom video

Whole-body MPC on Solo

Constrained trajectory optimization – discrete time

We consider the following discrete-time OCP:

$$
\begin{aligned}
\min_{\boldsymbol{x},\boldsymbol{u}} J(\boldsymbol{x},\boldsymbol{u}) &= \sum_{t=0}^{N-1} \ell_t(x_t,u_t) + \ell_N(x_N) \\
\text{s.t. } x_{t+1} &= f_t(x_t,u_t),\ \ t \in [\![0, N-1]\!] && \longleftrightarrow \lambda_{t+1} \\
x_0 &= \bar{x}_0 && \longleftrightarrow \lambda_0 \\
h_t(x_t,u_t) &\leq 0 && \longleftrightarrow \nu_t \\
h_N(x_N) &\leq 0 && \longleftrightarrow \nu_N
\end{aligned}
\tag{18}
$$

Optimality conditions – the Bellman way

The Bellman principle of optimality The optimal trajectory satisfies the relationship between the cost-to-go functions

$$V_t(x_t) = \min_{u_t} \max_{\nu_t} \ell_t(x_t, u_t) + \nu_t^\top h_t(x_t, u_t) + V_{t+1}(x_{t+1}) \tag{19}$$

where $x_{t+1} = f_t(x_t, u_t)$, and boundary condition

$$V_N(x) = \max_{\nu_N} \ell_N(x) + \nu_N^\top h_N(x). \tag{20}$$

Optimality conditions – the KKT way

The problem Lagrangian is

$$\mathscr{L}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\lambda}, \boldsymbol{\nu}) = \sum_{t=0}^{N-1} \ell_t(x_t, u_t) + \lambda_{t+1}^\top (f_t(x_t, u_t) - x_{t+1}) + \nu_t^\top h_t(x_t, u_t) \\ + \ell_N(x_N) + \nu_N^\top h_N(x_N) + \lambda_0^\top (x_0 - \bar{x}_0). \tag{21}$$

We can define the Hamiltonian

$$H_t(x, u, \lambda, \nu) = \ell_t(x, u) + \lambda^\top f_t(x, u) + \nu^\top h_t(x, u) \tag{22}$$

and terminal Lagrangian

$$\mathscr{L}_N(x, \nu) = \ell_N(x) + \nu^\top h_N(x). \tag{23}$$

Thus, the optimality conditions can be written as

$$\lambda_t = \nabla_x H_t(x_t, u_t, \lambda_{t+1}, \nu_t) \tag{24a}$$

$$0 = \nabla_u H_t(x_t, u_t, \lambda_{t+1}, \nu_t) \tag{24b}$$

$$0 = f_t(x_t, u_t) - x_{t+1} \tag{24c}$$

$$0 \leq h_t(x_t, u_t) \perp \nu_t \geq 0 \tag{24d}$$

$$0 \leq h_N(x_N) \perp \nu_N \geq 0 \tag{24e}$$

and boundary conditions

$$x_0 = \bar{x}_0 \tag{24f}$$

$$\lambda_N = \nabla_x \mathscr{L}_N(x_N, \nu_N). \tag{24g}$$

Can we SQP?

Yes. Start by defining

$$
\begin{aligned}
Q_t &= \nabla^2_{xx} H_t,\ S_t = \nabla^2_{xu} H_t,\ R_t = \nabla^2_{uu} H_t \\
q_t &= \nabla_x H_t,\ r_t = \nabla_u H_t \\
A_t &= \frac{\partial f_t}{\partial x},\ B_t = \frac{\partial f_t}{\partial u},\ s_t = f_t(x_t, u_t) \\
C_t &= \frac{\partial h_t}{\partial x},\ D_t = \frac{\partial h_t}{\partial u},\ d_t = h_t(x_t, u_t)
\end{aligned}
\tag{25}
$$

We can show that the SQP update $(\delta \boldsymbol{x}, \delta \boldsymbol{u}, \boldsymbol{\lambda}^+, \boldsymbol{\nu}^+)$ is obtained by solving the structured QP or *constrained LQR*

$$
\begin{aligned}
\min_{\delta \boldsymbol{x}, \delta \boldsymbol{u}} \sum_{t=0}^{N-1} \frac{1}{2} \begin{bmatrix} \delta x_t \\ \delta u_t \end{bmatrix}^\top \begin{bmatrix} Q_t & S_t \\ S_t^\top & R_t \end{bmatrix} \begin{bmatrix} \delta x_t \\ \delta u_t \end{bmatrix} + \ell_{t,x}^\top \delta x_t + \ell_{t,u}^\top \delta u_t \quad && \text{(26a)} \\
\text{s.t. } \delta x_{t+1} = A_t \delta x_t + B_t \delta u_t + \gamma_t \quad && \text{(26b)} \\
C_t \delta x_t + D_t \delta u_t + d_t \leq 0, \quad && \text{(26c)} \\
C_N \delta x_N + d_N \leq 0 \quad && \text{(26d)}
\end{aligned}
$$

This method is often called iLQR in the literature (Li and Todorov 2004; Giftthaler *et al.* 2018)

- not to be confused with the iLQR of Tassa *et al.* 2012.

(Software) Solutions

- ACADOS (Verschueren *et al.* 2022) implements an SQP-type algorithm, relying on the interior-point method HPIPM for the LQRs (Frison and Diehl 2020).
- CROCODDYL (Mastalli, Budhiraja, *et al.* 2020; Mastalli, Chhatoi, *et al.* 2023) has support for projection-based methods for equality constraints

[2] https://github.com/meco-group/fatrop
[3] https://github.com/machines-in-motion/mim_solvers

(Software) Solutions

- ACADOS (Verschueren *et al.* 2022) implements an SQP-type algorithm, relying on the interior-point method HPIPM for the LQRs (Frison and Diehl 2020).
- CROCODDYL (Mastalli, Budhiraja, *et al.* 2020; Mastalli, Chhatoi, *et al.* 2023) has support for projection-based methods for equality constraints
- FATROP[2] (Vanroye *et al.* 2023) implements an interior-point with an equality-LQR backend
- MIM-SOLVERS[3] (Jordana *et al.* 2023) implements a filter line-search SQP
- our library `aligator`, using proximal/augmented Lagrangian methods based on our prior work (J., Mansard, Carpentier ICRA'22, J., Bambade et *al.* IROS'22 + J., Bambade et *al.* T-RO journal submission)

[2] `https://github.com/meco-group/fatrop`
[3] `https://github.com/machines-in-motion/mim_solvers`