Lexicographic least-squares: fast resolution and applications in robotics

Dimitar Dimitrov

INRIA (BIPOP team)

05, November, 2014

A typical problem in robotics

- Simply concluding: "infeasible problem" is often not acceptable
- Ability to handle infeasibility in a meaningful way is critical
- Constraints of different nature, different importance

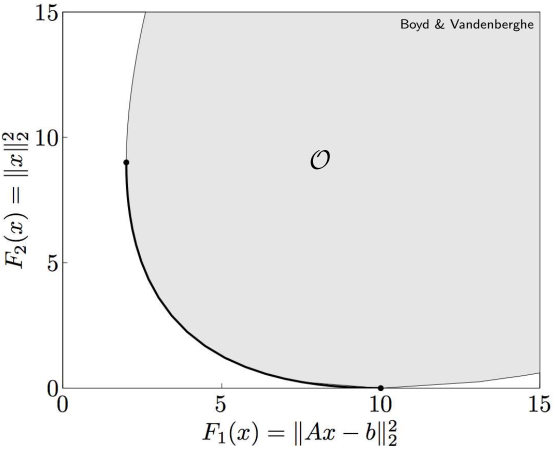
Some common goals/objectives

(1) $H\ddot{q} + h = \tau + J_c^T f$	dynamics
(2) $\lvert\tau\rvert \leq \tau_{max}$	joint torque limits
(3) $f \in \mathcal{C}$	friction cone
(4) $J_c\ddot{q} + \dot{J}_c\dot{q} = 0$	maintaining contact
(5) $J_d\ddot{q} + \dot{J}_d\dot{q} \geq \ddot{d}$	collision avoidance
(6) $\lvert\dot{q}\rvert \leq \dot{q}_{max}$	joint velocity limits
(7) $J_p\ddot{q} + \dot{J}_p\dot{q} = \ddot{p}$	a generic task

Multi-objective optimization

$$\underset{x}{\text{minimize}} \quad f_0 = (F_1(x), F_2(x))$$
$$\text{subject to} \quad x \in \mathcal{X}.$$

- $f_0 : \mathbb{R}^n \to \mathbb{R}^2$
- $\mathcal{O}$ - achievable objective values



Two popular solution strategies

Scalarization via weighting

$$\underset{x}{\text{minimize}} \quad w^T f_0(x)$$
$$\text{subject to} \quad x \in \mathcal{X}.$$

- choice of weights - nontrivial
- large weight - numerical problems

The lexicographic approach

$$\text{lex}\,\underset{x}{\text{minimize}} \quad f_0(x)$$
$$\text{subject to} \quad x \in \mathcal{X}.$$

- no finite tradeoffs among objectives
- strict hierarchical levels

Lexicographic order (example)

$$\underbrace{\begin{bmatrix}1\\1\\4\end{bmatrix}}_{f_0(x_1)} \prec \underbrace{\begin{bmatrix}1\\2\\3\end{bmatrix}}_{f_0(x_2)}$$

Lexicographic optimization

General lexicographic problem

$$\begin{aligned} \underset{x}{\text{lex minimize}} \quad & f_0(x) = (F_1(x), \ldots, F_p(x)) \\ \text{subject to} \quad & x \in \mathcal{X}. \end{aligned}$$

Lexicographic least-squares problem

$$\begin{aligned} \underset{x,\, v_1, \ldots, v_p}{\text{lex minimize}} \quad & f_0 = \left(\|v_1\|^2, \ldots, \|v_p\|^2\right) \\ \text{subject to} \quad & \begin{bmatrix} b_1^l \\ \vdots \\ b_p^l \end{bmatrix} \le \begin{bmatrix} A_1 \\ \vdots \\ A_p \end{bmatrix} x - \begin{bmatrix} v_1 \\ \vdots \\ v_p \end{bmatrix} \le \begin{bmatrix} b_1^u \\ \vdots \\ b_p^u \end{bmatrix} \end{aligned}$$

$$\mathcal{X} = \left\{(x, v) : b^l \le Ax - v \le b^u\right\}, \quad F_k = \|v_k\|^2$$

Lexicographic optimization - applications

Lexicographic optimization is not only an attractive theoretical formulation but a widely used tool in practice

Interesting statistics

Reported applications of multi-objective optimization involved

- lexicographic variant: 80%
- weighting variant: 18%
- other variants: 2%

Many algorithms are based on the use of lexicographic order

- some interior-point methods (for linear programs) solve

$$\operatorname*{lex\,minimize}_{x} \quad \left(\|C_1 x - y_1\|^2, \ldots, \|C_p x - y_p\|^2\right)$$

"useful models, but a major difficulty ... a bit involved constrained least-squares problem"

Example (pseudo-inverse)

Least-squares

$$\begin{aligned} \underset{x,\, v_1}{\text{minimize}} \quad & \|v_1\|^2 \\ \text{subject to} \quad & v_1 = Cx - y \end{aligned}$$

Least-norm

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & \|x\|^2 \\ \text{subject to} \quad & Cx = y \end{aligned}$$

Tikhonov regularization

$$\begin{aligned} \underset{x,\, v_1, v_2}{\text{minimize}} \quad & \|v_1\|^2 + \mu \|v_2\|^2 \\ \text{subject to} \quad & v_1 = Cx - y \\ & v_2 = x \end{aligned}$$

$x^\star = C^+ y \ (\mu \to 0)$

$$\begin{aligned} \underset{x,\, v_1, v_2}{\text{lex minimize}} \quad & (\|v_1\|^2, \|v_2\|^2) \\ \text{subject to} \quad & v_1 = Cx - y \\ & v_2 = x \end{aligned}$$

Solving a lexicographic problem

Isaac Asimov's famous three laws of robotics:

1. A robot may not injure a human being or, through inaction, allow a human being to come to harm.
2. A robot must obey the orders given to it by human beings, except where such orders would conflict with the 1st law.
3. A robot must protect its own existence as long as such protection does not conflict with the 1st or 2nd law.

These laws are stated in a way that clearly imposes a strict hierarchy of importance between different goals.

Classical approach

- minimize $F_1(x)$
- minimize $F_2(x)$, subject to $F_1(x) \leq F_1^\star$
- minimize $F_3(x)$, subject to $F_1(x) \leq F_1^\star$, $F_2(x) \leq F_2^\star$
- ...

Example (pseudo-inverse)

$$x^\star = C^+ y$$

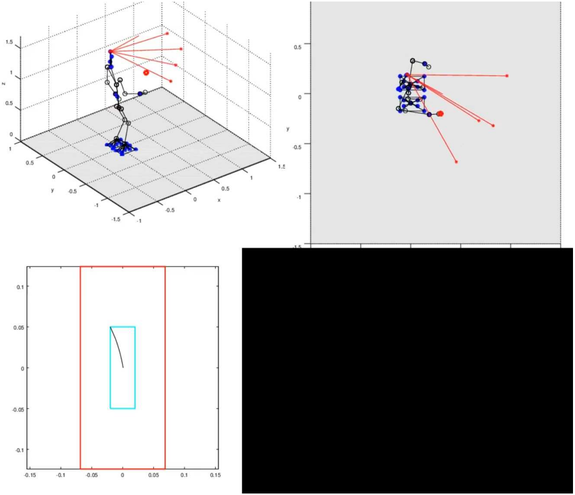
A sequence of two least-squares problems

- find
$$\mathcal{S} = \operatorname{arg}\min_x \ \|Cx - y\|^2$$
- solve
$$\begin{array}{ll} \underset{x}{\text{minimize}} & \|x\|^2 \\ \text{subject to} & x \in \mathcal{S} \end{array}$$

Example: stability margin (Movie)

$$(1) \succ (2) \succ (3) \succ (4) \succ (5)$$

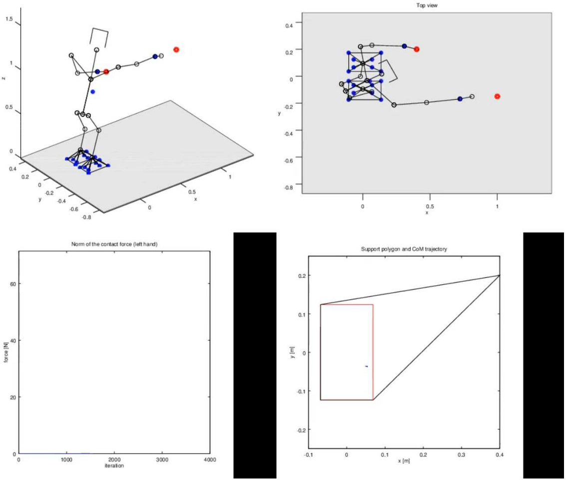
(1) dynamics
(2) joint angle/torque limits
(3) CoM stay in red box
(4) reaching task
(5) CoM stay in cyan box



Example: reaching with optional support (Movie)

$$(1) \succ (2) \succ (3) \succ (4) \succ (5) \succ (6)$$

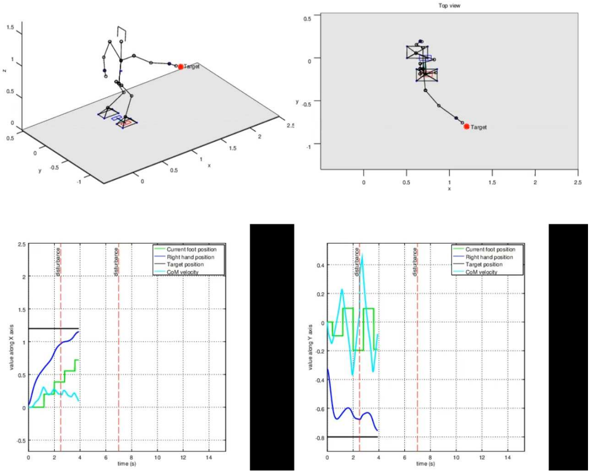
(1) dynamics
(2) joint angle/torque limits
(3) avoid to fall
(4) reaching task
(5) small CoM velocity
(6) avoid additional support



Example: task induced walking (Movie)

$$\cdots \succ (k) \succ (j) \succ \ldots$$

(.) ...
(k) avoid to fall
(j) reaching task
(.) ...



Solving a LexLSI problem

Sequential approach (classical)

Identify $v_k^\star$ by solving the problem

$$\begin{aligned} \underset{x,\, v_k}{\text{minimize}} \quad & \|v_k\|^2 \\ \text{subject to} \quad & b_k^l \leq A_k x - v_k \leq b_k^u \\ & b_j^l \leq A_j x - v_j^\star \leq b_j^u, \quad j = 1, \ldots, k-1 \end{aligned}$$

Simultaneous approach

We approach the problem as a whole

$$\begin{aligned} \underset{x,\, v_1, \ldots, v_p}{\text{lex minimize}} \quad & (\|v_1\|^2, \ldots, \|v_p\|^2) \\ \text{subject to} \quad & b^l \leq Ax - v \leq b^u \end{aligned}$$

by using a multi-objective primal active-set strategy

- much less iterations
- much easier to hot-start meaningfully

Primal active-set strategy

Fact

There exists a LexLSE problem such that $x^\star_{lse} = x^\star_{lsi}$

LexLSE problem

$$\begin{aligned} \underset{x,\, r_1,\ldots,r_p}{\text{lex minimize}} \quad & \left(\|r_1\|^2, \ldots, \|r_p\|^2\right) \\ \text{subject to} \quad & \begin{bmatrix} r_1 \\ \vdots \\ r_p \end{bmatrix} = \begin{bmatrix} C_1 \\ \vdots \\ C_p \end{bmatrix} x - \begin{bmatrix} y_1 \\ \vdots \\ y_p \end{bmatrix} \end{aligned}$$

- (C_k, y_k) - constraints assumed to be active at level k
- r_k - residual at level k (to be minimized)

Active-set schemes try to identify the “correct” LexLSE problem.

Activation of a constraint (nothing new here)

"Phase 1" is not required

Initial feasible pair (x, v) can be constructed in a trivial way, $e.g.$, assume $x = 0$, and set

$$v = \frac{1}{2}(b^l + b^u)$$

Given

- Feasible iterate (x, v)
- Solution $(x^\star, v^\star)$ of the current LexLSE problem

Form

- Step direction $(\Delta x, \Delta v) = (x^\star - x, v^\star - v)$

Take a step

- $(x, v) \leftarrow (x, v) + \alpha(\Delta x, \Delta v)$

α is the largest step in the direction $(\Delta x, \Delta v)$ for which feasibility is preserved.

Deactivation of a constraint

When $(\Delta x, \Delta v) = (0, 0)$, verify if $x^{\star}_{lse}$ is optimal for LexLSI

Lagrange multiplier associated with a constraint

Measures rate of change of objective, consequent upon changes in the constraint

- Single objective, m constraints $\rightarrow$ m Lagrange multipliers
- p objectives, m constraints $\rightarrow$ pm Lagrange multipliers

We need to evaluate the sensitivity of every objective to (small) variations of each constraint

Deactivation of a constraint

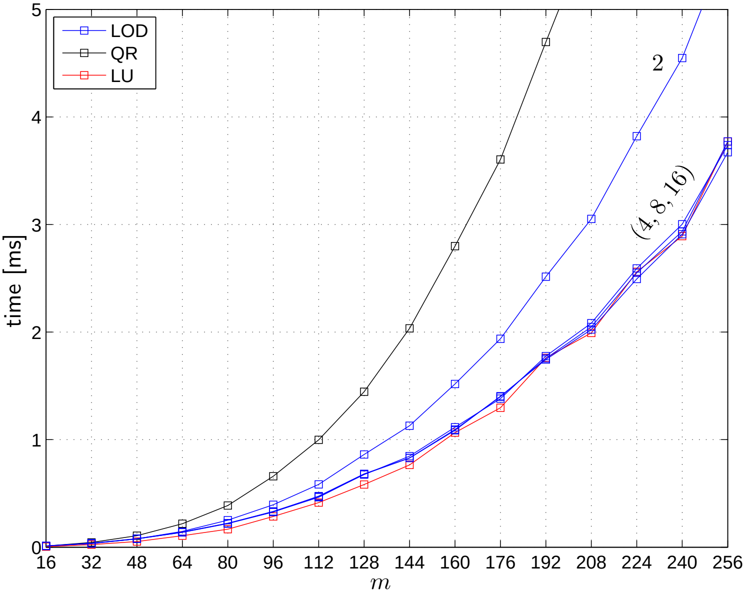
A matrix of Lagrange multipliers

$$\Lambda = \begin{bmatrix} \lambda_{11} & \lambda_{12} & \dots & \lambda_{1p} \\ & \lambda_{22} & \dots & \lambda_{2p} \\ & & \ddots & \vdots \\ & & & \lambda_{pp} \end{bmatrix} \in \mathbb{R}^{m \times p}.$$

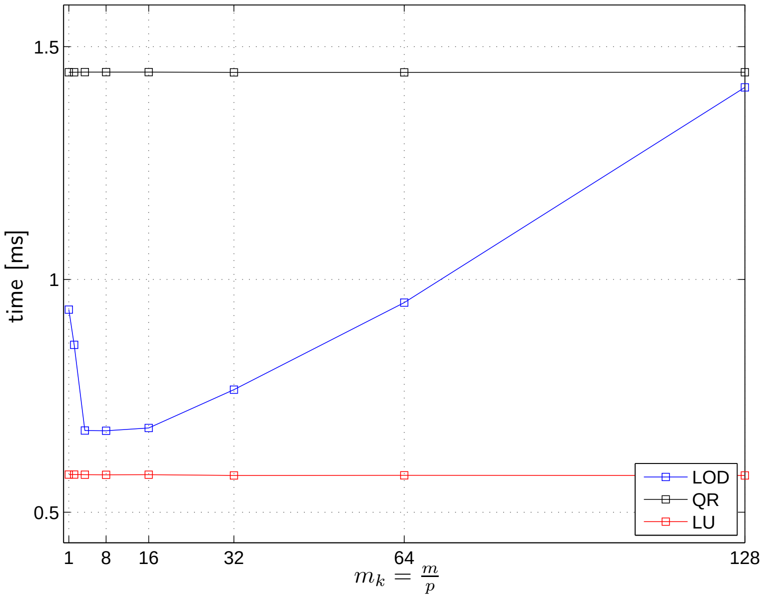
- $\lambda_{kk} = r_k$
- λ_{jk} sensitivity of objective k to (small) changes in the constraints associated with objective j
- Objective k is insensitive to constraints associated with objective j for $j \geq k$

- i-th constraint can be deactivated if i-th row of Λ has lexicographically wrong sign
- the algorithm terminates with a solution after finitely many iterations

LexLSE problem: vary problem size



LexLSE problem: vary number of hierarchical levels



QR vs. LU (intuitions)

Example (two conflicting equations)

$$\begin{bmatrix}1\\1\end{bmatrix} x = \begin{bmatrix}0\\1\end{bmatrix}$$

QR

Apply Givens rotation

$$R = \begin{bmatrix}\cos(\frac{\pi}{4}) & \sin(\frac{\pi}{4})\\ -\sin(\frac{\pi}{4}) & \cos(\frac{\pi}{4})\end{bmatrix}$$

$$\begin{bmatrix}\sqrt{2}\\0\end{bmatrix} x = \frac{1}{2}\begin{bmatrix}\sqrt{2}\\\sqrt{2}\end{bmatrix}$$

$$x^{QR} = \frac{1}{2}, \quad v^{QR} = \begin{bmatrix}0.5\\-0.5\end{bmatrix}$$

LU

Apply Gauss transformation

$$G = \begin{bmatrix}1 & 0\\ -1 & 1\end{bmatrix}$$

$$\begin{bmatrix}1\\0\end{bmatrix} x = \begin{bmatrix}0\\1\end{bmatrix}$$

$$x^{LU} = 0, \quad v^{LU} = \begin{bmatrix}0\\-1\end{bmatrix}$$

Weighted Givens rotation

Weighted problem

$$\underset{x}{\text{minimize}} \quad \left\| W \left(\begin{bmatrix} a_1 \\ a_2 \end{bmatrix} x - \begin{bmatrix} 0 \\ 1 \end{bmatrix} \right) \right\|^2, \quad W = \begin{bmatrix} w_1 & 0 \\ 0 & w_2 \end{bmatrix}$$

Can be solved by applying a “weighed Givens rotation” to

$$\begin{bmatrix} a_1 \\ a_2 \end{bmatrix} x = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

Weighed Givens rotation → Gauss transformation ($\frac{w_1}{w_2} \to \infty$)

$$\begin{bmatrix} 1 & \frac{w_2^2}{w_1^2} \frac{a_2}{a_1} \\ -\frac{a_2}{a_1} & 1 \end{bmatrix}$$

Application: time optimal control

Double-integrator (discrete-time)

$$s_{k+1} = \underbrace{\begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}}_{A} s_k + \underbrace{\begin{bmatrix} \frac{T^2}{2} \\ T \end{bmatrix}}_{B} u_k, \quad k = 0, \dots, N-1$$

Preview

$$\underbrace{\begin{bmatrix} s_1 \\ \vdots \\ s_N \end{bmatrix}}_{\bar{s}} = S s_0 + T \underbrace{\begin{bmatrix} u_0 \\ \vdots \\ u_{N-1} \end{bmatrix}}_{\bar{u}}$$

Time optimal control (a definition)

$$\bar{s} = S s_0 + T\bar{u}$$

Definition

Transfer s_0 to the origin using bounded control inputs ($\bar{u}$) in the smallest number of discrete-time steps (N)

Alternative point of view

Find $\bar{u}$ such that $\bar{s}$ has the most number of zeros

Bang-bang control

Find $\bar{u}$ such that $\bar{s}$ has the most number of zeros

Can be achieved by solving (in a MPC context)

$$\begin{aligned} \underset{\bar{u},\,\bar{s}}{\text{minimize}} \quad & \|\bar{s}\|_0 \\ \text{subject to} \quad & \bar{s} = S s_0 + T\bar{u} \\ & -\mathbb{1} \leq \bar{u} \leq \mathbb{1} \end{aligned}$$

Bang-bang control using ℓ_1 norm

Cardinality problem

$$\begin{aligned} \underset{\bar{u},\,\bar{s}}{\text{minimize}} \quad & \|\bar{s}\|_0 \\ \text{subject to} \quad & \bar{s} = S s_0 + T\bar{u} \\ & -\mathbb{1} \le \bar{u} \le \mathbb{1} \end{aligned}$$

ℓ_1 approximation

$$\begin{aligned} \underset{\bar{u},\,\bar{s}}{\text{minimize}} \quad & \|W\bar{s}\|_1 \\ \text{subject to} \quad & \bar{s} = S s_0 + T\bar{u} \\ & -\mathbb{1} \le \bar{u} \le \mathbb{1} \end{aligned}$$

Important observation

For some choices of W, the solution of the ℓ_1 problem results in time-optimal state evolution

Choice of W

Many different ways to parameterize W. We investigated the use of

$$W(\beta)\bar{s} = \sum_k (x_k + \beta \dot{x}_k),$$

where β is a non-negative scalar.

Problems

Is there a single β that transfers any s_0 to the origin in a time-optimal way?

Conclusion

- A single β is enough under some (possibly restrictive) conditions on N and T
- To find such a β solve multi-parametric programming problem (parameters: s_0, β)