Control with QP and SOCP

Contact-aware Control, Lecture 8

by Sergei Savin

Fall 2020

Content

- Quadratic programming
  - Special case with an analytic solution
  - Example
  - Example - solution using quadprog
  - Example - solution using CVX
  - Application
  - What it can't do
- Second-order cone programming
- Friction cone and SOCP
  - Formulation
  - Application
- Read more

Quadratic programming

There are special cases of optimization problems that can be solved numerically very efficiently. One of those is *quadratic programming*.

General form of a quadratic program (QP) is given below:

$$\begin{aligned} \underset{\mathbf{x}}{\text{minimize}} \quad & \mathbf{x}^\top \mathbf{H} \mathbf{x} + \mathbf{f}^\top \mathbf{x}, \\ \text{subject to} \quad & \begin{cases} \mathbf{A}\mathbf{x} \leq \mathbf{b}, \\ \mathbf{C}\mathbf{x} = \mathbf{d}. \end{cases} \end{aligned} \tag{1}$$

where $\mathbf{H}$ is positive-definite and $\mathbf{A}\mathbf{x} \leq \mathbf{b}$ describe a *convex region*.

If there are no inequality constraints:

$$\begin{aligned} \underset{\mathbf{x}}{\text{minimize}} \quad & \mathbf{x}^\top \mathbf{H} \mathbf{x} + \mathbf{c}^\top \mathbf{x}, \\ \text{subject to} \quad & \mathbf{A}\mathbf{x} = \mathbf{b}. \end{aligned} \tag{2}$$

a quadratic program can be solved analytically.

We have the following problem: find such $\mathbf{x}$ that minimizes $\mathbf{x}^\top \mathbf{M} \mathbf{x}$, while $\mathbf{C}\mathbf{x} = \mathbf{y}$. In other words:

$$\begin{aligned} \underset{\mathbf{x}}{\text{minimize}} \quad & \mathbf{x}^\top \mathbf{M} \mathbf{x}, \\ \text{subject to} \quad & \mathbf{C}\mathbf{x} = \mathbf{y}. \end{aligned} \tag{3}$$

More concrete:

$$\begin{aligned} \underset{\mathbf{x}}{\text{minimize}} \quad & \begin{bmatrix} x_1 & x_2 & x_3 \end{bmatrix} \begin{bmatrix} 1 & 0 & 1 \\ 0 & 5 & 0 \\ 1 & 0 & 3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}, \\ \text{subject to} \quad & \begin{bmatrix} 1 & 7 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = 1. \end{aligned} \tag{4}$$

We can use a dedicated solver for this class of problems - `quadprog` provided by MATLAB. Here is the solution:

```
M = [1 0 1; 0 5 0; 1 0 3];
C = [1 7 2];
y = 1;

x = quadprog(M,[],[],[],C,y)
```

Average solution time is **0.56** ms

Alternatively we can invoke one of the most powerful convex optimization tools with a user-friendly coding style - CVX:

```
M = [1 0 1; 0 5 0; 1 0 3];
C = [1 7 2];
y = 1;

cvx_begin
variables x(3);
minimize( x' * M * x );
subject to
    C*x == y;
cvx_end
```

Official web page of the CVX package

Remember CTC control from the lecture on error dynamics:

$$\begin{cases} \mathbf{v} = \mathbf{H}(\ddot{\mathbf{q}}^* + \mathbf{K}_d \dot{\mathbf{e}} + \mathbf{K}_p \mathbf{e}) + \mathbf{c} \\ \mathbf{T}\mathbf{u} + \mathbf{F}^\top \lambda = \mathbf{v} \end{cases} \tag{5}$$

Assume that $\lambda = [\lambda_1^\top, \ ...\lambda_m^\top]^\top \in \mathbb{R}^{3m}$, $\lambda_i \in \mathbb{R}^3$ and represents a m individual unilateral contact with normal unit-vectors to the contact surface $\mathbf{n}_i$, and no friction. Then we can write the control law as a solution to the quadratic program:

$$\begin{aligned} \underset{\mathbf{u},\mathbf{v},\lambda_i}{\text{minimize}} \quad & \mathbf{u}^\top \mathbf{u}, \\ \text{subject to} \quad & \begin{cases} \mathbf{H}(\ddot{\mathbf{q}}^* + \mathbf{K}_d \dot{\mathbf{e}} + \mathbf{K}_p \mathbf{e}) + \mathbf{c} = \mathbf{v} \\ \mathbf{T}\mathbf{u} + \sum\limits_{i=1}^{m} \mathbf{F}_i^\top \lambda_i = \mathbf{v} \\ -\mathbf{n}_i^\top \lambda_i \leq 0 \end{cases} \end{aligned} \tag{6}$$

Remember friction cone constraints:

$$||[\tau_1 \ \tau_2]^\top \mathbf{f}|| \leq \mu \mathbf{n}^\top \mathbf{f} \tag{7}$$

These can't be directly represented in a quadratic program.

The general form of a Second-order cone program (SOCP) is:

$$\begin{aligned} \underset{\mathbf{x}}{\text{minimize}} \quad & \mathbf{f}^\top \mathbf{x}, \\ \text{subject to} \quad & \begin{cases} ||\mathbf{A}_i \mathbf{x} + \mathbf{b}_i||_2 \le \mathbf{c}_i^\top \mathbf{x} + d_i, \\ \mathbf{F}\mathbf{x} = g. \end{cases} \end{aligned} \tag{8}$$

QP are a subset of SOCP.

Friction cone and SOCP

We already saw that a no-friction contact can be represented as a QP. Next we will try to show that friction cone (a more general model) can be represented as a SOPC (a more general optimization problem type).

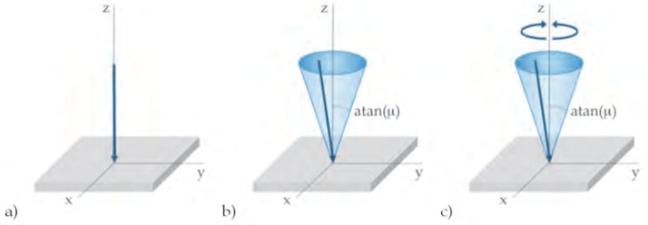

Fig. 2. Contact models commonly used in robotics: a) Point contact without friction; b) Point contact with friction; c) Soft-finger contact

Picture is from *Sancho-Bru, J.L., Pérez-González, A., Mora, M.C., Lón, B.E., Vergara, M., Iserte, J.L., Rodriguez-Cervantes, P.J. and Morales, A., 2011. Towards a realistic and self-contained biomechanical model of the hand. In Theoretical biomechanics. IntechOpen.*

Coming back to the the friction cone general view:

$$||[\tau_1 \ \tau_2]^\top \mathbf{f}|| \le \mu \mathbf{n}^\top \mathbf{f} \tag{9}$$

This is has the form of a conic constraint, which is admissible for SOCP:

$$||\mathbf{A}_i \mathbf{x} + \mathbf{b}_i||_2 \le \mathbf{c}_i^\top \mathbf{x} + d_i \tag{10}$$

Consider this problem again:

$$\begin{cases} \mathbf{v} = \mathbf{H}(\ddot{\mathbf{q}}^* + \mathbf{K}_d\dot{\mathbf{e}} + \mathbf{K}_p\mathbf{e}) + \mathbf{c} \\ \mathbf{T}\mathbf{u} + \mathbf{F}^\top\lambda = \mathbf{v} \end{cases} \tag{11}$$

But this time we require that:

$$||[\tau_{i,1} \ \tau_{i,2}]^\top \lambda_i|| \leq \mu \mathbf{n}_i^\top \lambda_i \tag{12}$$

Solution can take the form of a second-order cone program:

$$
\begin{aligned}
& \underset{\mathbf{u},\mathbf{v},\lambda_i}{\text{minimize}} \quad \mathbf{u}^\top \mathbf{u}, \\
& \text{subject to} \quad
\begin{cases}
\mathbf{H}(\ddot{\mathbf{q}}^* + \mathbf{K}_d \dot{\mathbf{e}} + \mathbf{K}_p \mathbf{e}) + \mathbf{c} = \mathbf{v} \\
\mathbf{T}\mathbf{u} + \sum_{i=1}^{m} \mathbf{F}_i^\top \lambda_i = \mathbf{v} \\
||[\tau_{i,1} \ \ \tau_{i,2}]^\top \lambda_i|| \leq \mu \mathbf{n}_i^\top \lambda_i
\end{cases}
\end{aligned}
\tag{13}
$$

Read more

You can read and watch more at:

- CVX user guide
- quadprog - MATLAB documentation
- Computational Intelligence 2020, Lecture 5 (Least Squares and Quadratic Programming)
- Computational Intelligence 2020, Lecture 6 (Domain, Convex Domains)
- Computational Intelligence 2020, Lecture 7 (Linear inequality representation of convex domains)
- Computational Intelligence 2020, Lecture 8 (QCQP, SOCP)

Lecture slides are available via Moodle.

You can help improve these slides at:
github.com/SergeiSa/Contact-Aware-Control-Slides-Fall-2020

Check Moodle for additional links, videos, textbook suggestions.