Lecture 25 - Whole Body Control

Last time:

- Task-space control using Pseudo-inverses
- Briefly: Extensions to robots with contacts

Today:

- Formulation with whole-body QPs
- Embedding optimal template behavior
- Solving QPs efficiently

What about in contact?

J_c = Contact Jacobian J_t = Task Jacobian

$$H\ddot{q} + C\dot{q} + \tau_g = S^T\tau_j + J_c^T f_c \qquad J_c\dot{q} = 0$$

"Projected" Equations:

$$H\ddot{q} + N_c^T C\dot{q} + J_c^T \Lambda \dot{J}_c \dot{q} + N_c^T \tau_g = N_c^T S^T \tau_j \quad (*)$$

Same as before

$(*)\ J_t H^{-1}(\cdot) =$

Shorthand: $J_{t|c} = J_t N_c$

$$\Lambda_{t|c} = \left(J_{t|c} H^{-1} J_{t|c}^T\right)^{-1}$$

$$\bar{J}_{t|c} = H^{-1} J_{t|c}^T \Lambda_{t|c}$$

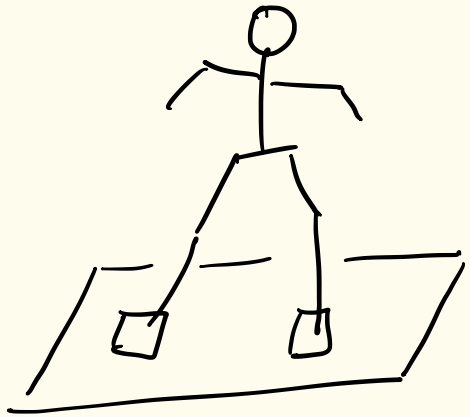
$$\mu_{t|c} = \bar{J}_{t|c}^T C + \Lambda_{t|c} J_t H^{-1} J_c^T \Lambda_c \dot{J}_c - \Lambda_{t|c}\dot{J}_t$$

$$p_{t|c} = \bar{J}_{t|c}^T \tau_g$$

Constraint-consistent task space dynamics

$$\Lambda_{t|c}\dot{V}_t + \mu_{t|c}\dot{q} + p_{t|c} = \bar{J}_{t|c}^T S^T \tau$$

Same form as before!

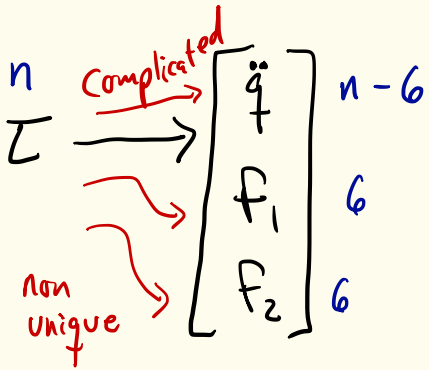


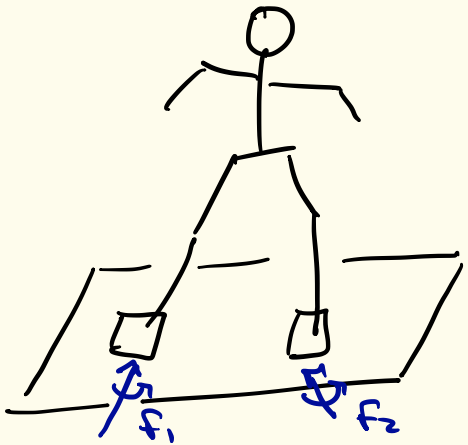
Limitations and Benefits of Projecting away the Constraints

$$H\ddot{q} + N_c^T C\dot{q} + J_c^T \Lambda \dot{J}_c \dot{q} + N_c^T \tau_g = N_c^T S^T \tau_j$$

(Implicitly $\tau \Rightarrow \begin{bmatrix} f_1 \\ f_2 \end{bmatrix}$ under feet)

- When moving conservatively we might not need to worry about contact force limits
- Forces are ignored $\Leftarrow$ if they stay w/in bounds this may simplify control
- However these equations aren't valid if required contact forces exceed limits...
- Consider n joints, so $\ddot{q} \in \mathbb{R}^{n+6}$ but only n-6 DoFs

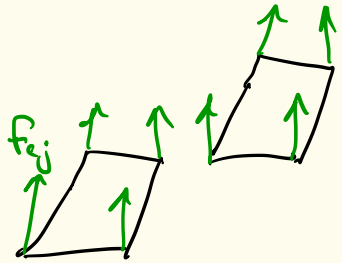




Alternate Strategy: Let optimization do all the work

$$\min_{\ddot{q},\tau,F_{c_j}} \left\| J_t\ddot{q} + \dot{J}_t\dot{q} - \dot{v}_t^c \right\|_2^2 + \sum_j \|F_{c_j} - F_{c_j}^o\|_2^2 + \|\tau\|_2^2$$

$$\text{s.t.} \begin{cases} H\ddot{q} + C\dot{q} + \tau_g = S^T\tau + \sum_j J_{c_j}^T F_{c_j} \\ J_c\ddot{q} + \dot{J}_c\dot{q} = 0 \\ F_{c_j} \in \text{cone} \\ \underline{\tau} \le \tau \\ \tau \le \bar{\tau} \end{cases} \quad \text{C1-C5}$$



$f_{c_j} \in$ 

$x^TQx + c^Tx$

$Ax < b$

- Depending on how you treat the friction cone this is a SOCP OR a QP

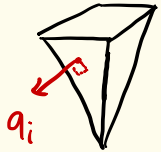
Friction Constraints with Friction Pyramids

A polyhedral convex can be represented in two forms:

- Face Form

$$\mathcal{C} = \left\{ x \;\middle|\; \begin{bmatrix} a_1^T \\ \vdots \\ a_m^T \end{bmatrix} x \leq 0 \right\}$$

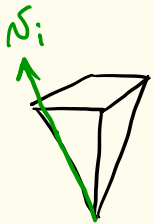
(each face is a hyperplane $a_i^T x = 0$)



- Span Form

$$\mathcal{C} = \left\{ \sum_i \lambda_i v_i \;\middle|\; \text{each } \lambda_i \geq 0 \right\}$$

each edge is a ray in direction v_i (sometimes called the generators of the cone)



- Cone double description (CDD) algorithms to convert between reps

Alternate Strategy: Let optimization do all the work

First:

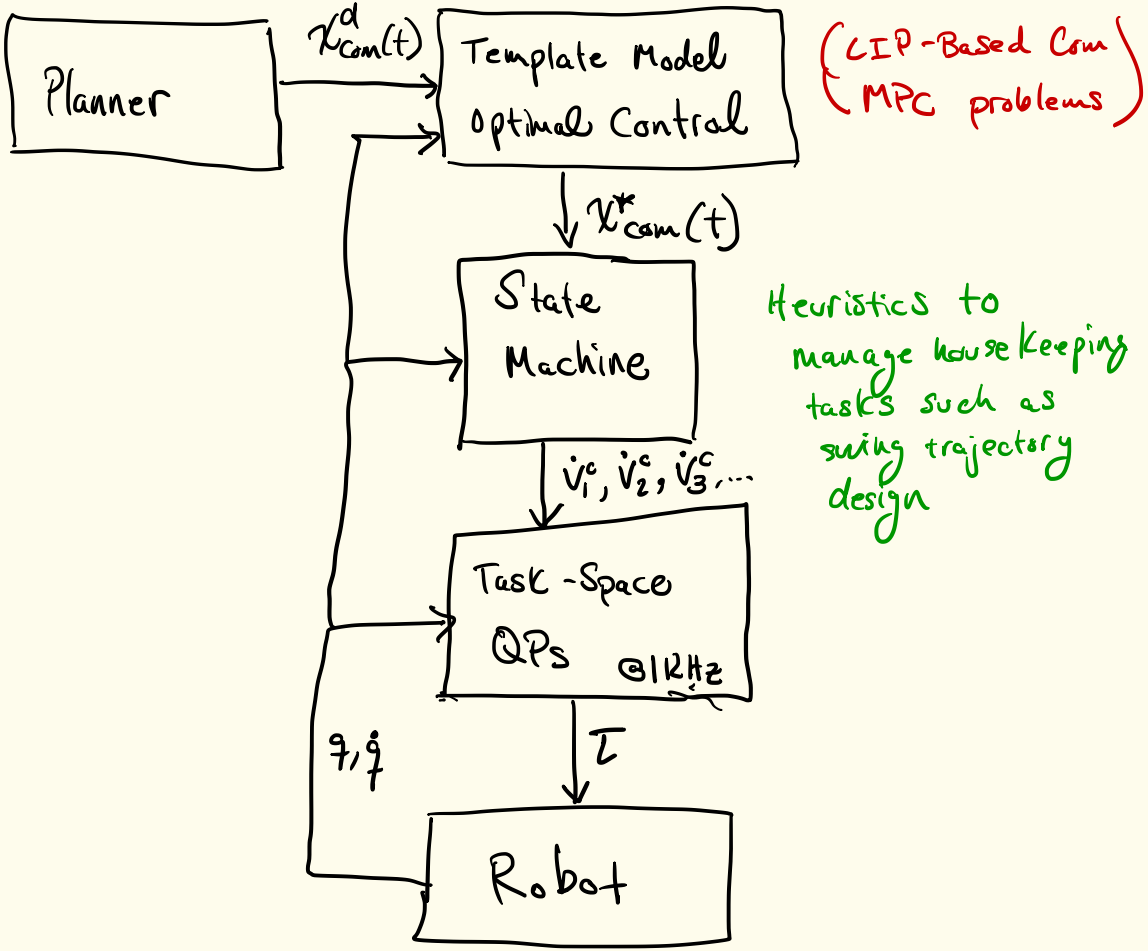
$$\min_{\ddot{q},\tau,f_{cj}} \| J_1\ddot{q} + \dot{J}_1\dot{q} - \dot{V}_1^c \| \quad \text{s.t. } C_1 - C_5 \quad \Rightarrow \ddot{q}_1^*$$

Next:

$$\min_{\ddot{q},\tau,f_{cj}} \| J_2\ddot{q} + \dot{J}_2\dot{q} - \dot{V}_2^c \| \quad \text{s.t. } C_1 - C_5$$

$$J_1\ddot{q} + \dot{J}_1\dot{q} = J_1\ddot{q}_1^* + \dot{J}_1\dot{q}$$

Common Framework

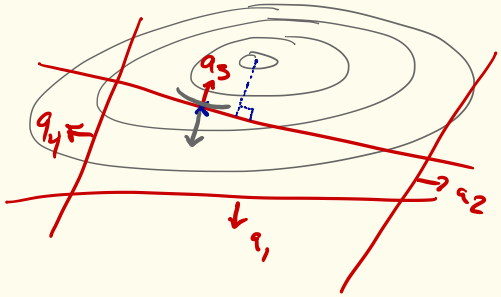


Solving QPs:

$\min \frac{1}{2} x^T Q x + c^T x$

s.t. $Ax \leq b$

$$A = \begin{bmatrix} a_1^T \\ \vdots \\ a_m^T \end{bmatrix}$$



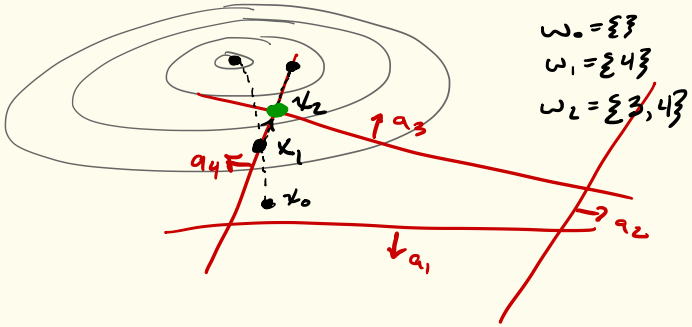
- We say a constraint $a_i^T x \leq b_i$ is active @ x^* if $a_i^T x^* = b_i$
- Denote the active set $\mathcal{A}(x^*) = \{ j \mid a_j^T x^* = b_j \}$
- Suppose x^* is an optimal solution and that the vectors $\{a_j\}_{j \in \mathcal{A}(x^*)}$ are linearly independent. Then $\exists!\ \lambda \in \mathbb{R}^m$ s.t.

① $\nabla_x \mathcal{L} = 0$

② $\lambda \geq 0 \qquad Ax^* - b \leq 0$

③ $\lambda_i (a_i^T x^* - b_i) = 0$ (complementary slackness)

Solving QPs: Active Set Methods



(1) Consider an initial feasible guess x_0
Let W_0 a linearly independent subset of the active constraints @ x_0

(2) Solve an equality-constrained QP w/ W_i as the set of constraints

⇒ If already @ the optimal point for (2), check Lagrange multipiers

⇒ If all $\lambda_j \geq 0$, then stop, problem solved

⇒ If any $\lambda_j < 0$, pick one (K) set $W_{i+1} = W_i \setminus \{K\}$

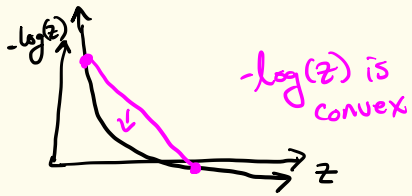
(3) Move as far as possible in the direction of the optima from (2)

⇒ Update $W_{i+1} = W_i \cup \{j\}$ where j is the index of a newly active constraint.

(4) Repeat back to (2)

Under mild assumptions, terminates in finite time.

Solving QPs: Interior Point Methods



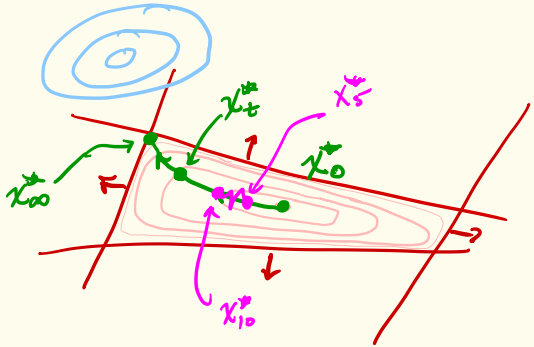
Consider the unconstrained optimization problem:

Barrier Subproblem:

$$x_t^* = \text{argmin } \tfrac{1}{2}x^T Q x + c^T x + \frac{1}{t}\sum_{j=1}^{m} -\log(b_i - a_i^T x) : f_t(x)$$

t: barrier parameter

barrier function



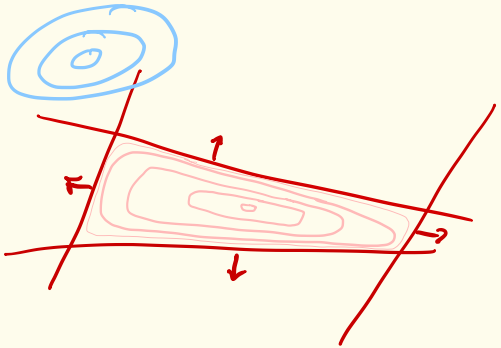
- The log barrier gives a penalty for being close to the edge of the constraints, and rewards being far away from them.
- x_t^* traces what we call the central path

- Although the problem is not a QP anymore it is convex and unconstrained.

  ⇒ Can be solved by finding a point where $\nabla_x f_t(x) = 0$

  can be found efficiently with Newton's method.

- IP methods solve barrier subproblems for increasing values of t.



Active Set Methods

- Polynomial runtime in x combinatorially in # constraints
  2^m possible working sets
- Work for linear constraints and can handle nonlinear constraints via linearizations
- Can be "warm started" with a good initial guess. Often faster than IPM w/ this trick.
- Moves along boundary of feasible region

Interior Point Methods

- Polynomial time in size of x and in the # constraints
- Penalty functions generalize to nonlinear constraints w/o linearization (e.g., SOC, or LMI)
- Difficult to warm start but highly predictable solve times.
- Stays strictly w/in the feasible region