

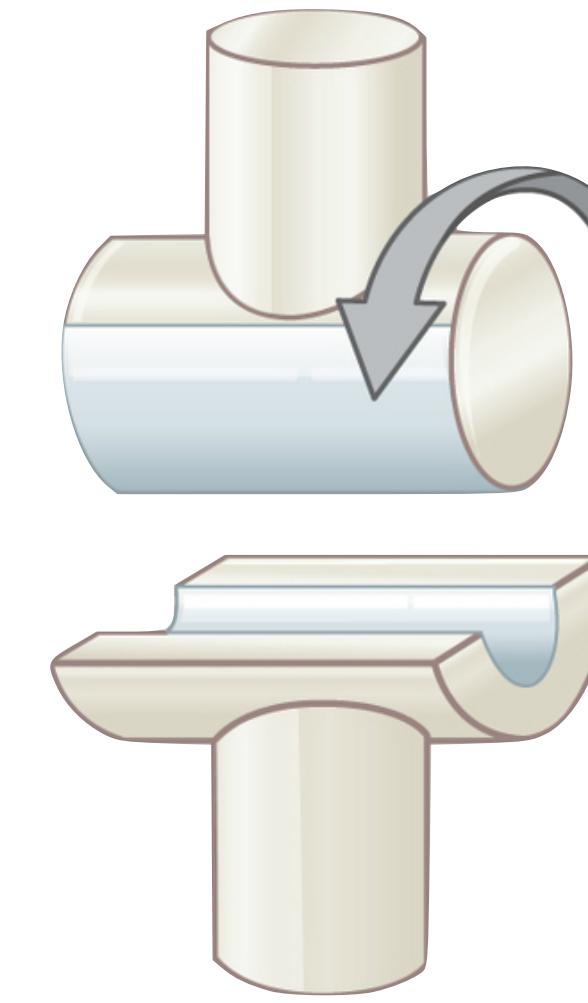
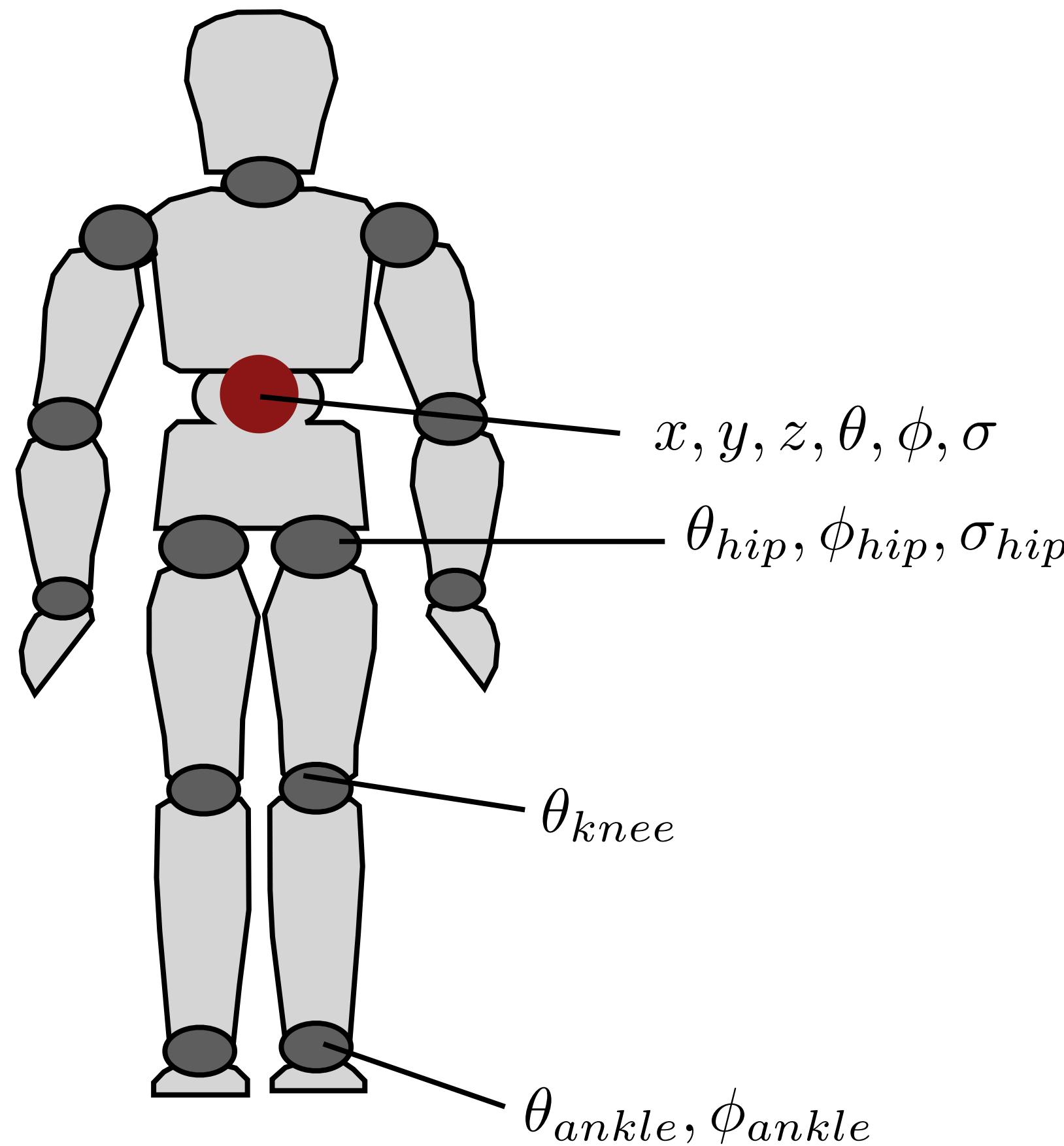
Lecture 15:

Inverse Kinematics

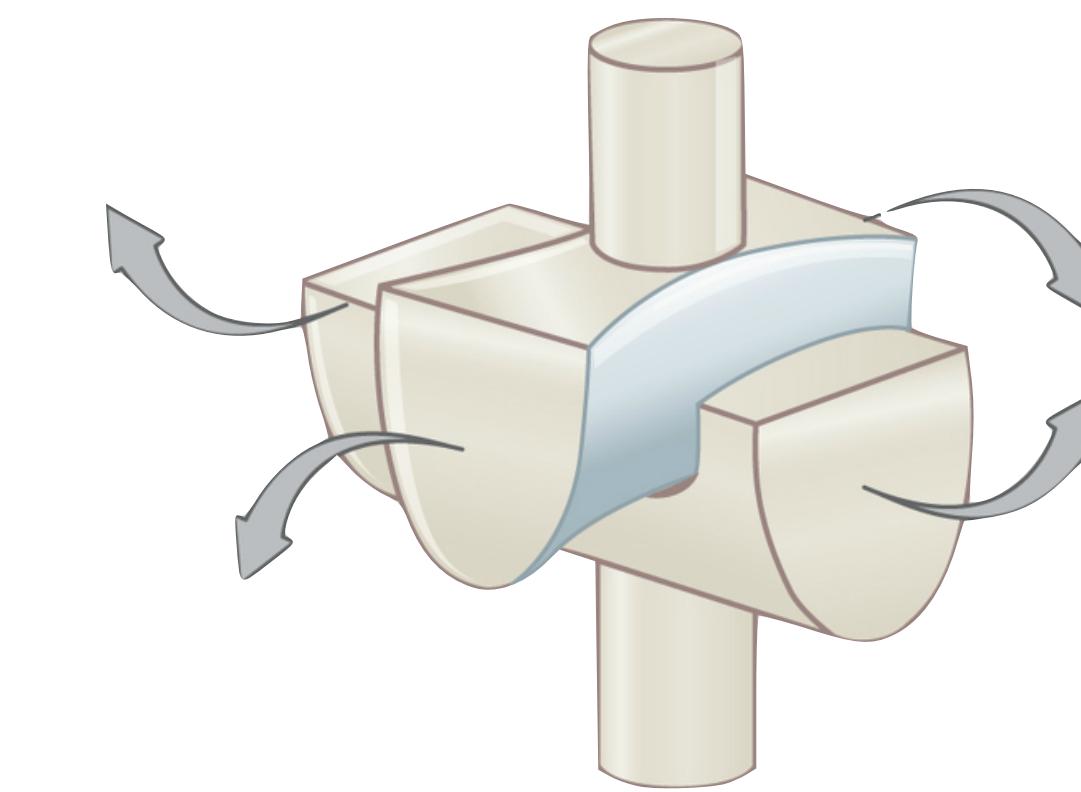
**FUNDAMENTALS OF COMPUTER GRAPHICS
Animation & Simulation
Stanford CS248B, Fall 2023**

Define a configuration (pose)

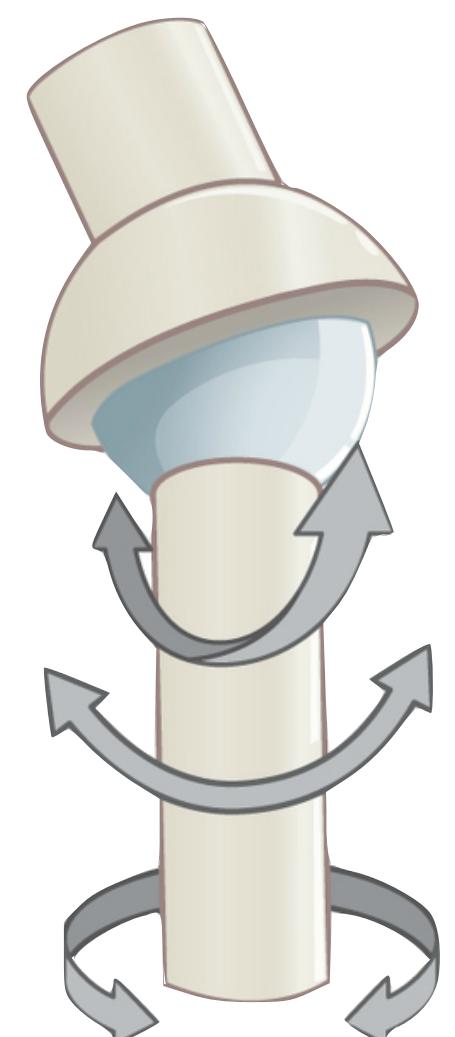
$$\mathbf{q} \leftarrow \{x, y, z, \theta, \phi, \sigma, \theta_{hip}, \phi_{hip}, \sigma_{hip}, \theta_{knee}, \theta_{ankle}, \phi_{ankle}, \dots\}$$



1-DOF joint



2-DOF joint



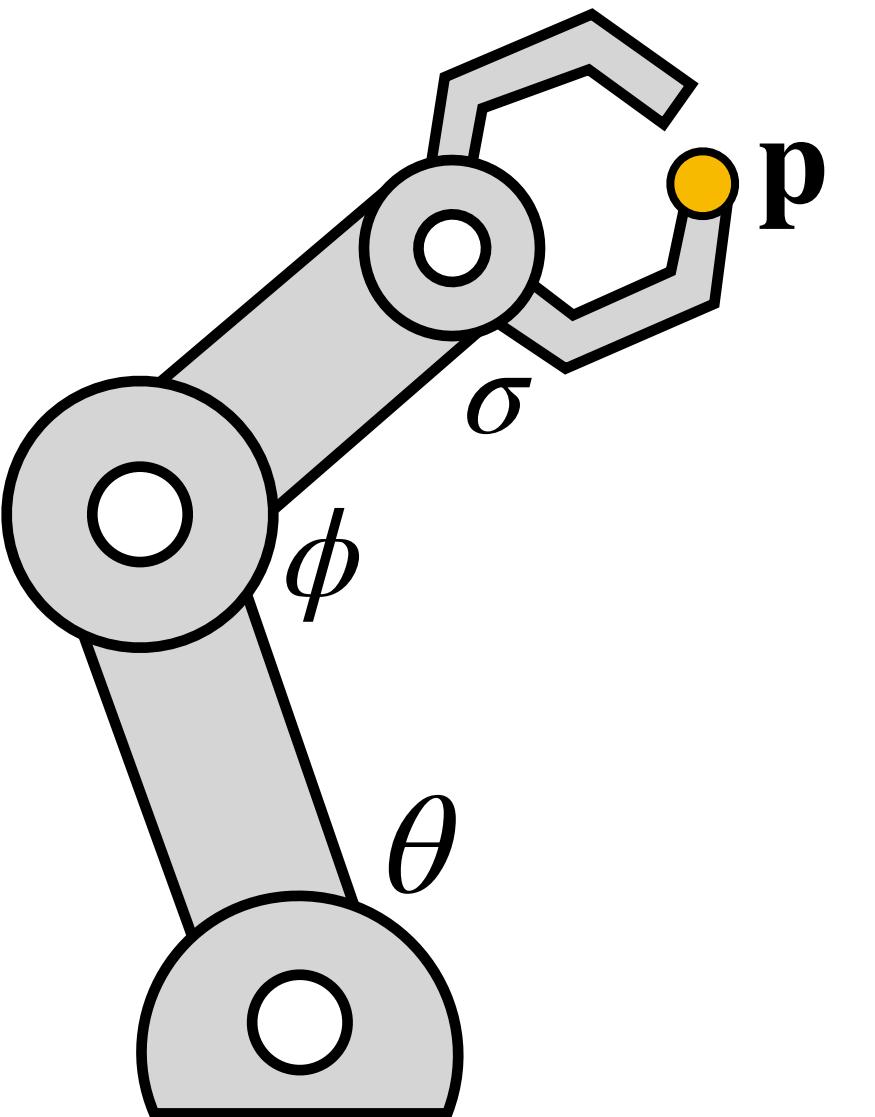
3-DOF joint

Kinematics

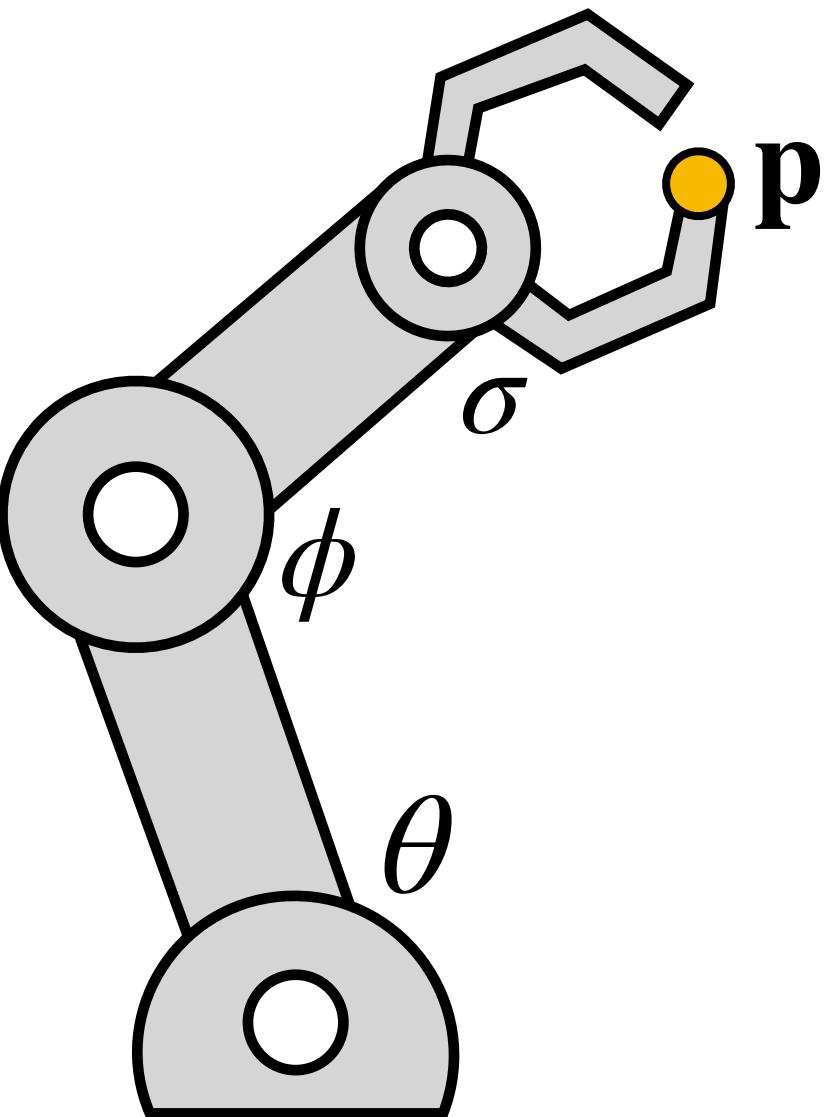
- **Forward kinematics**
 - Given a joint configuration, what is the 3D position of a point on the structure?
- **Inverse kinematics**
 - Given a target position for a point on the structure, what angles do the joints need to be to achieve that target point?

Quiz

Which one is solving Inverse Kinematics?



$$[\theta, \phi, \sigma] = f(\mathbf{p})$$

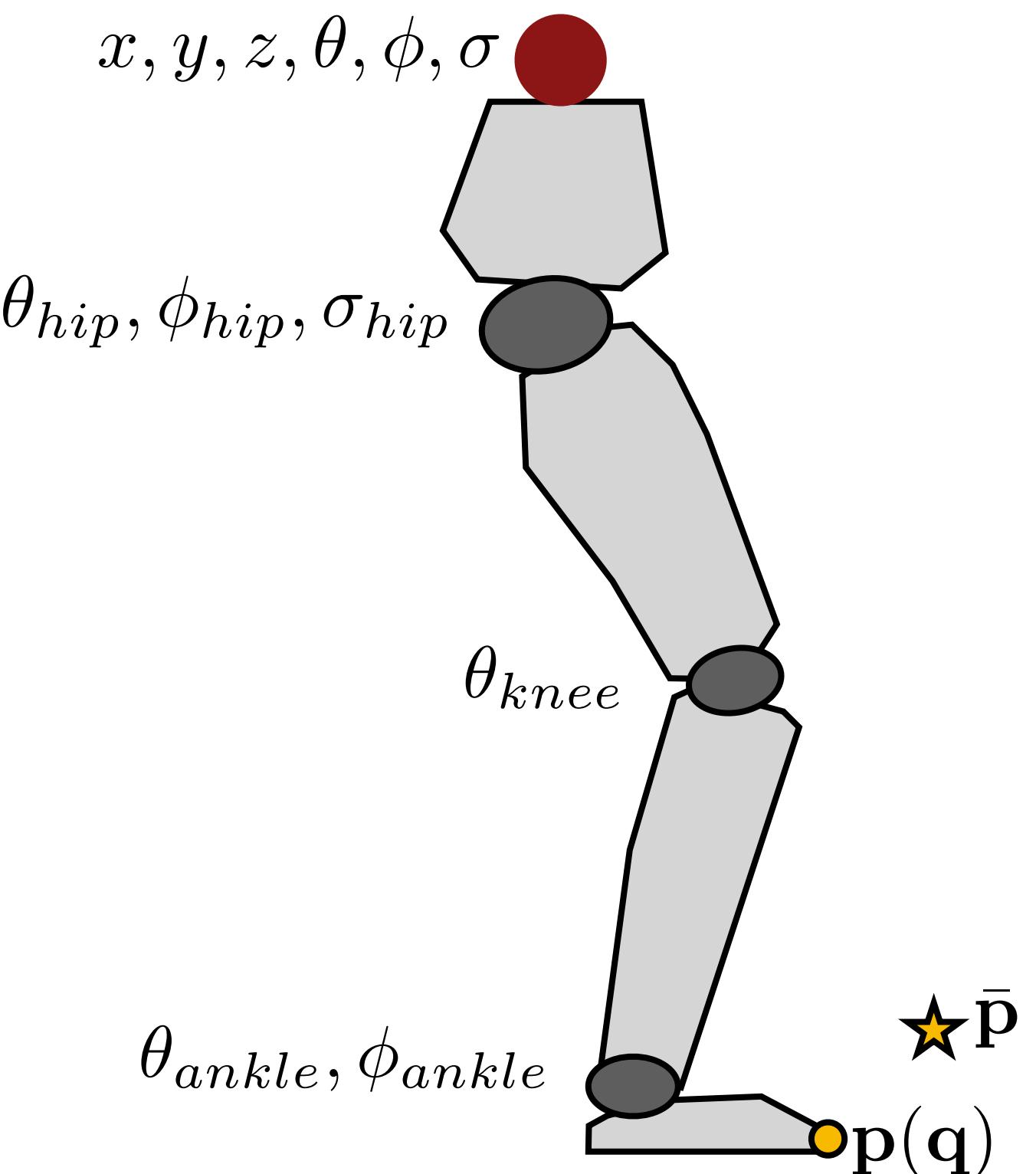


$$\mathbf{p} = f(\theta, \phi, \sigma)$$

Why inverse?

- The world is described in the Cartesian space but the movement is described in the pose space.
 - IK provides more intuitive control.
 - IK maintains environment constraints.

IK constraints



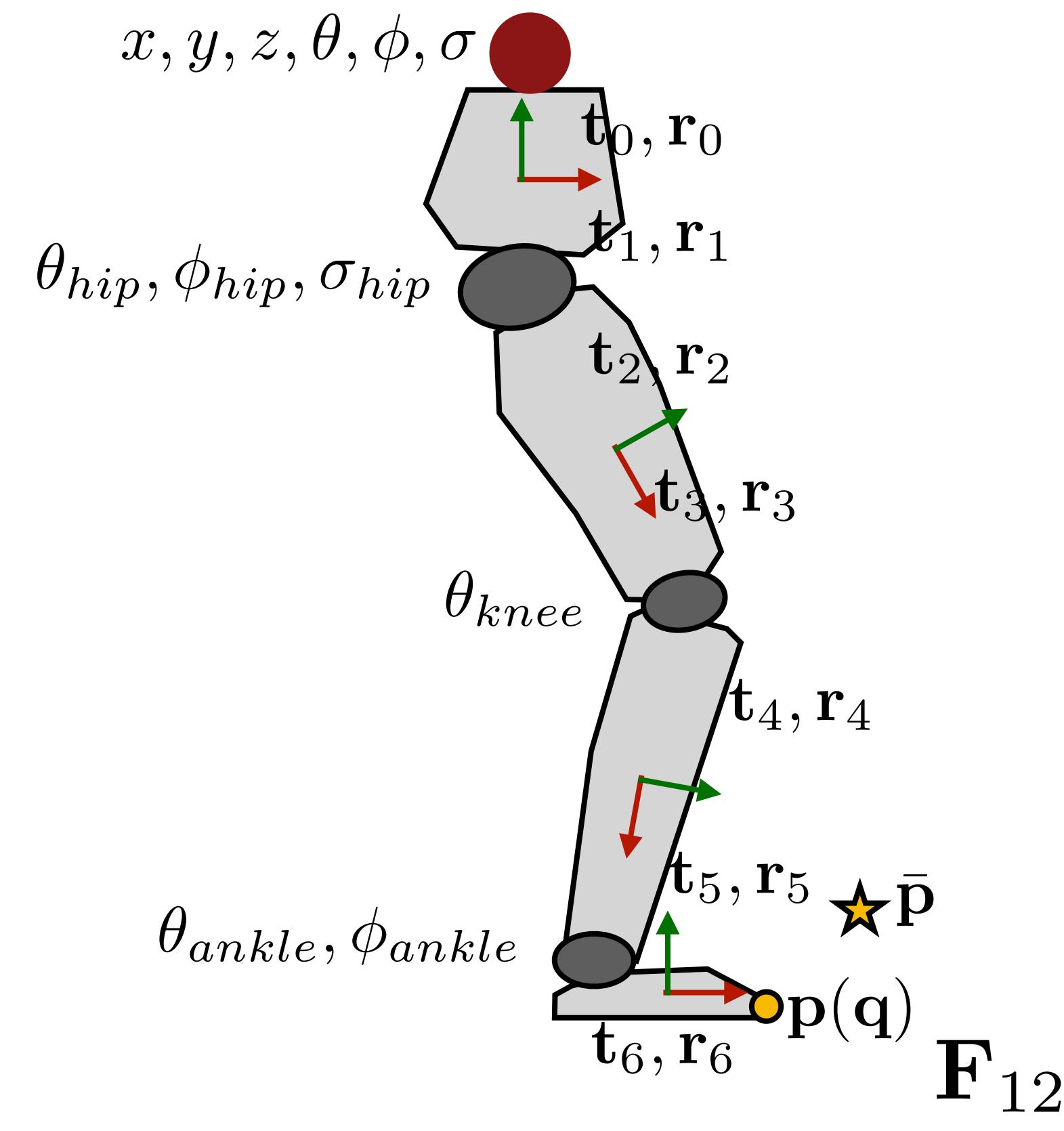
What is the pose such that the toe reaches point \bar{p} ?

$$C(q) = p(q) - \bar{p} = 0$$

This is an *inverse kinematic* question, but we need to evaluate toe position, $p(q)$, given a pose q , and *that* is a *forward kinematics* question.

What is the coordinates of the toe in the world frame?

IK constraints



What is the pose such that the toe reaches point \bar{p} ?

$$C(q) = p(q) - \bar{p} = 0$$

What is the coordinates of the toe in the world frame?

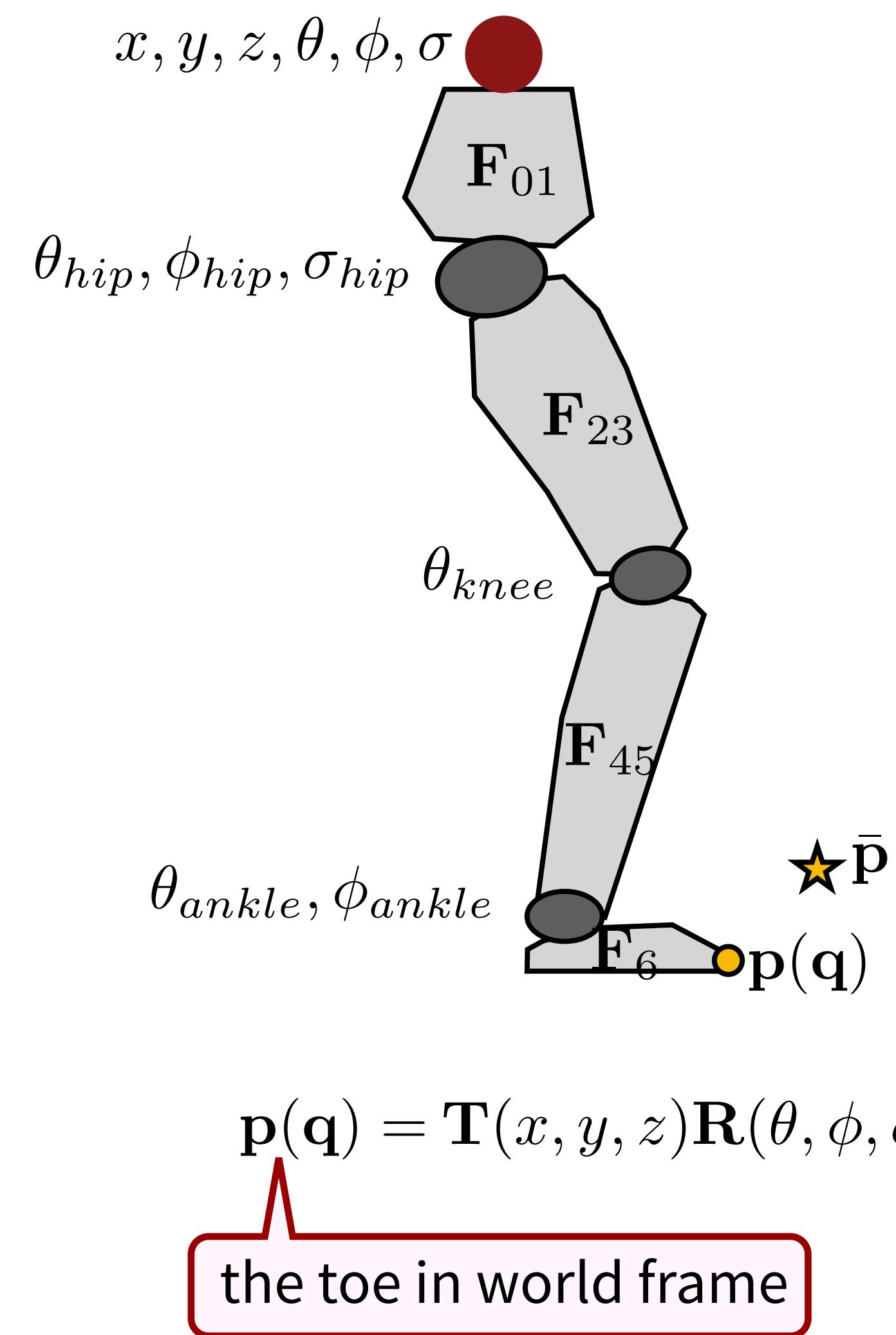
$$p(q) = T(x, y, z)R(\theta, \phi, \sigma)F(t_0, r_0)F(t_1, r_1)R(\theta_{hip}, \phi_{hip}, \sigma_{hip})F(t_2, r_2)F(t_3, r_3)$$

the toe in world frame

$$\begin{aligned} & R(\theta_{knee})F(t_4, r_4)F(t_5, r_5)R(\theta_{ankle}, \phi_{ankle})F(t_6, r_6)p_0 \\ & F_{23} \\ & F_{45} \\ & F_6 \end{aligned}$$

the toe in local frame

IK constraints



What is the pose such that the toe reaches point \bar{p} ?

$$C(q) = p(q) - \bar{p} = 0$$

Plugging the kinematic chain,
 $C(q) = 0$ is a nonlinear root
finding problem.

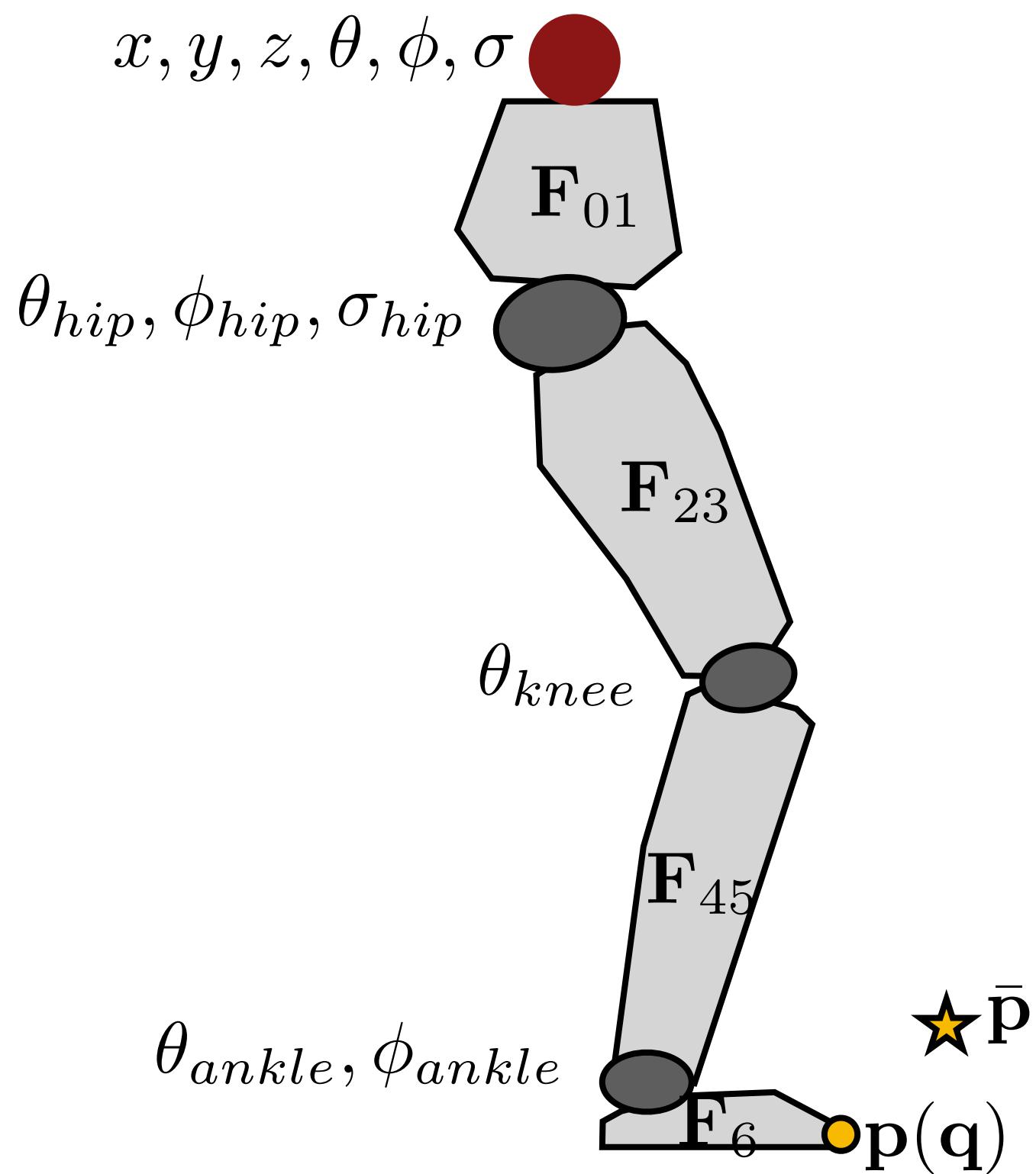
What is the coordinates of the toe in the world frame?

the toe in local frame

$$p(q) = T(x, y, z)R(\theta, \phi, \sigma)F_{01}R(\theta_{hip}, \phi_{hip}, \sigma_{hip})F_{23}R(\theta_{knee})F_{45}R(\theta_{ankle}, \phi_{ankle})F_6p_0$$

the toe in world frame

Solutions

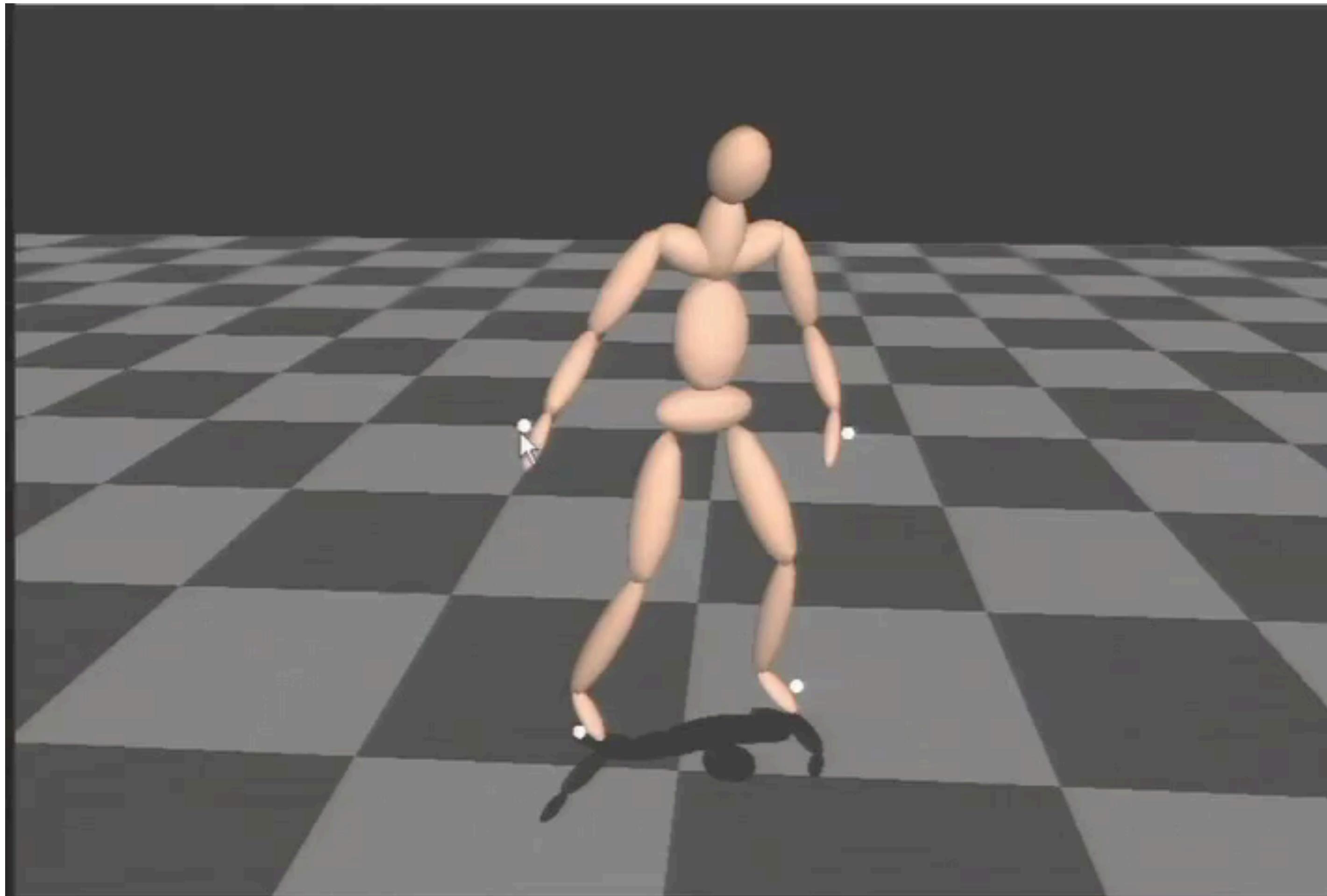


$$C(q) = p(q) - \bar{p} = 0$$

- Two challenges when $C(q)$ is nonlinear in q
 - Difficult to solve analytically
 - There could be
 - Single solution
 - No solution
 - Multiple solutions

For human pose problem, we usually have multiple solutions. Mostly bad!

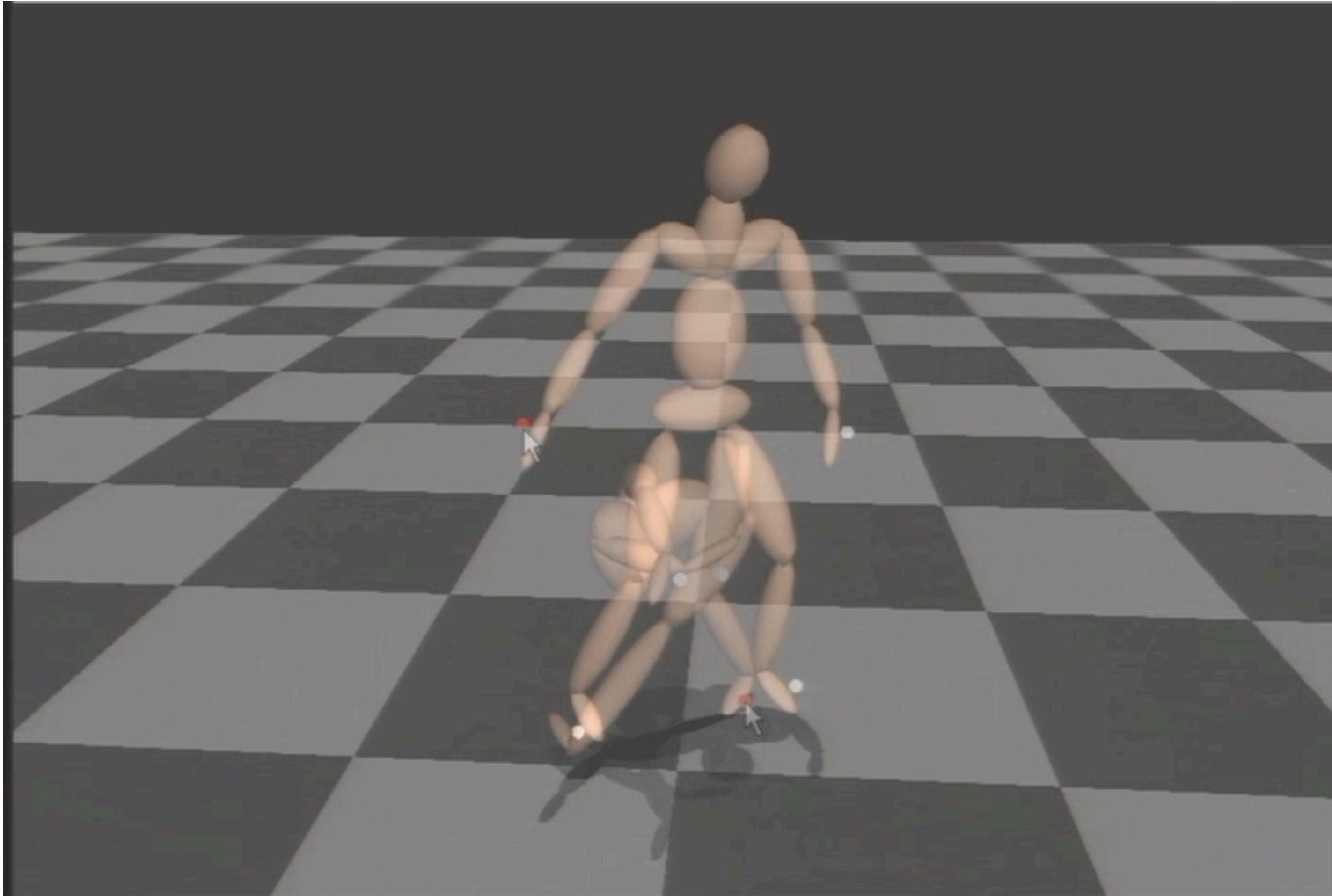
Under constrained IK



How to find optimal solution?

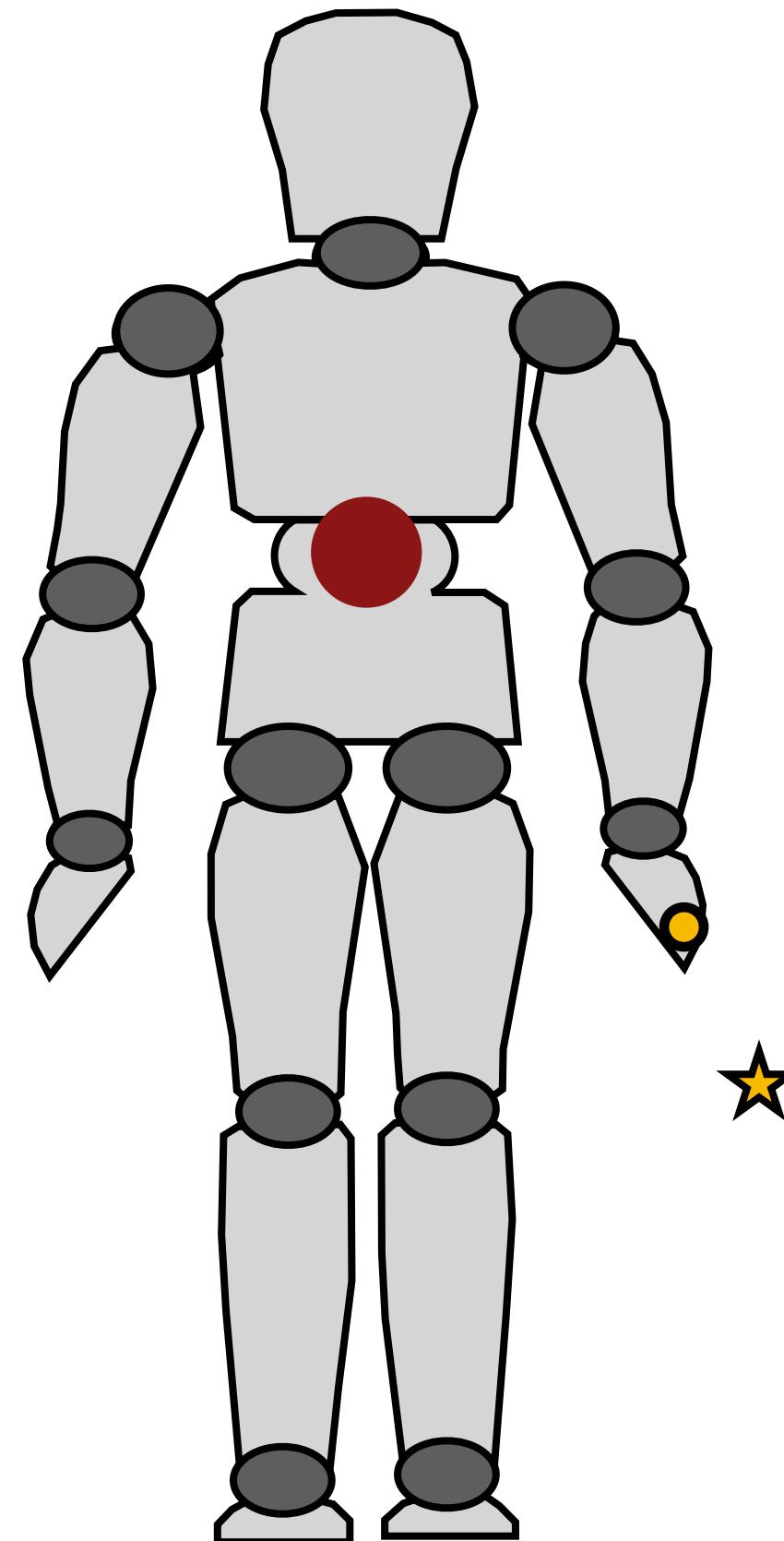
- Heuristics (move the outermost links first)
- Closest to the current configuration
- Energy minimization
- Natural looking motion (based on real-world data)

Humanlike IK



Formulate an optimization problem

Solve for joint configuration \mathbf{q}



$$\text{minimize } g(\mathbf{q})$$

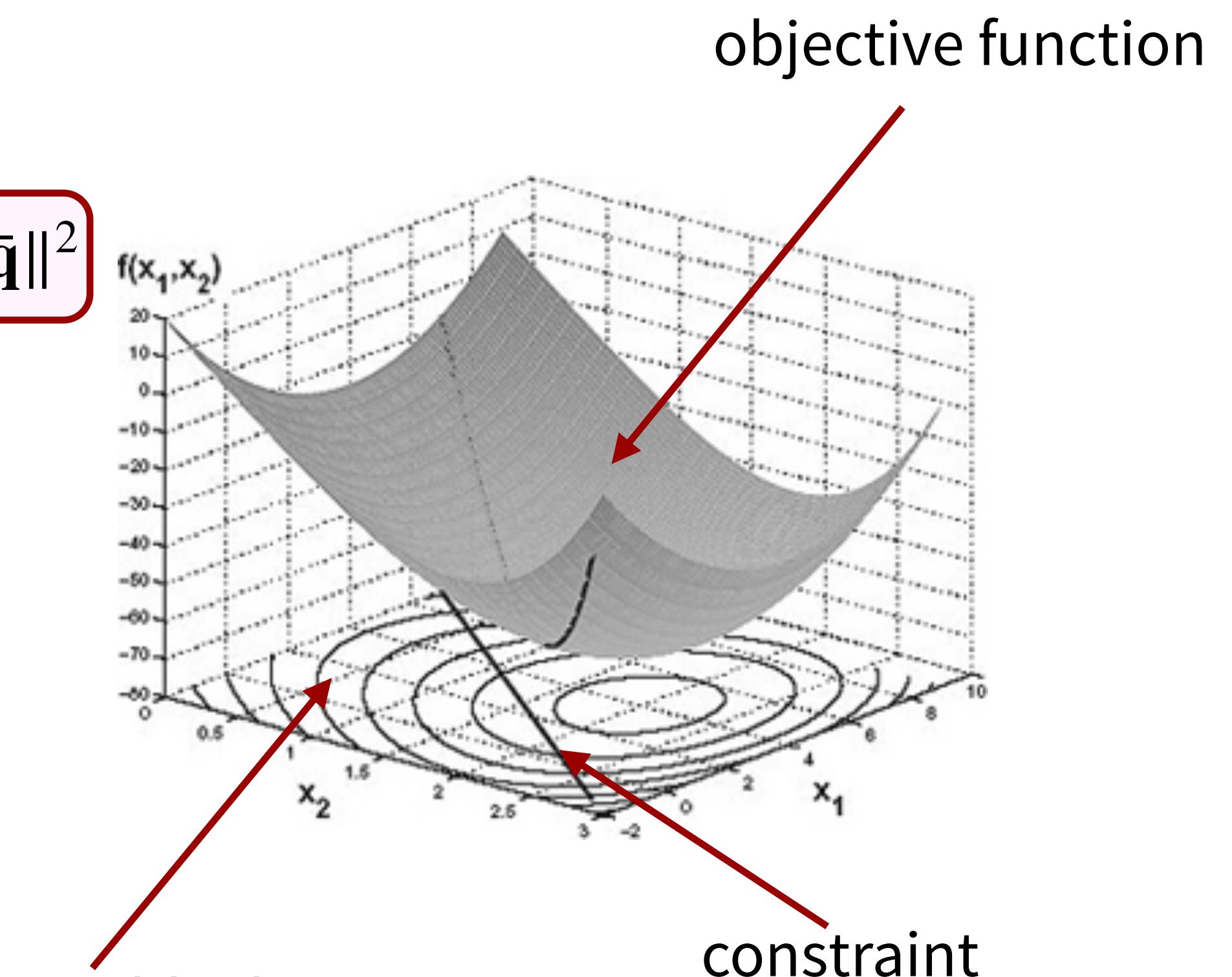
e.g. $g(\mathbf{q}) = \|\mathbf{q} - \bar{\mathbf{q}}\|^2$

$$\text{subject to } C(\mathbf{q}) = 0$$

$$C(\mathbf{q}) = p(\mathbf{q}) - \bar{p}$$

decision variable domain

Nonconvex optimization!

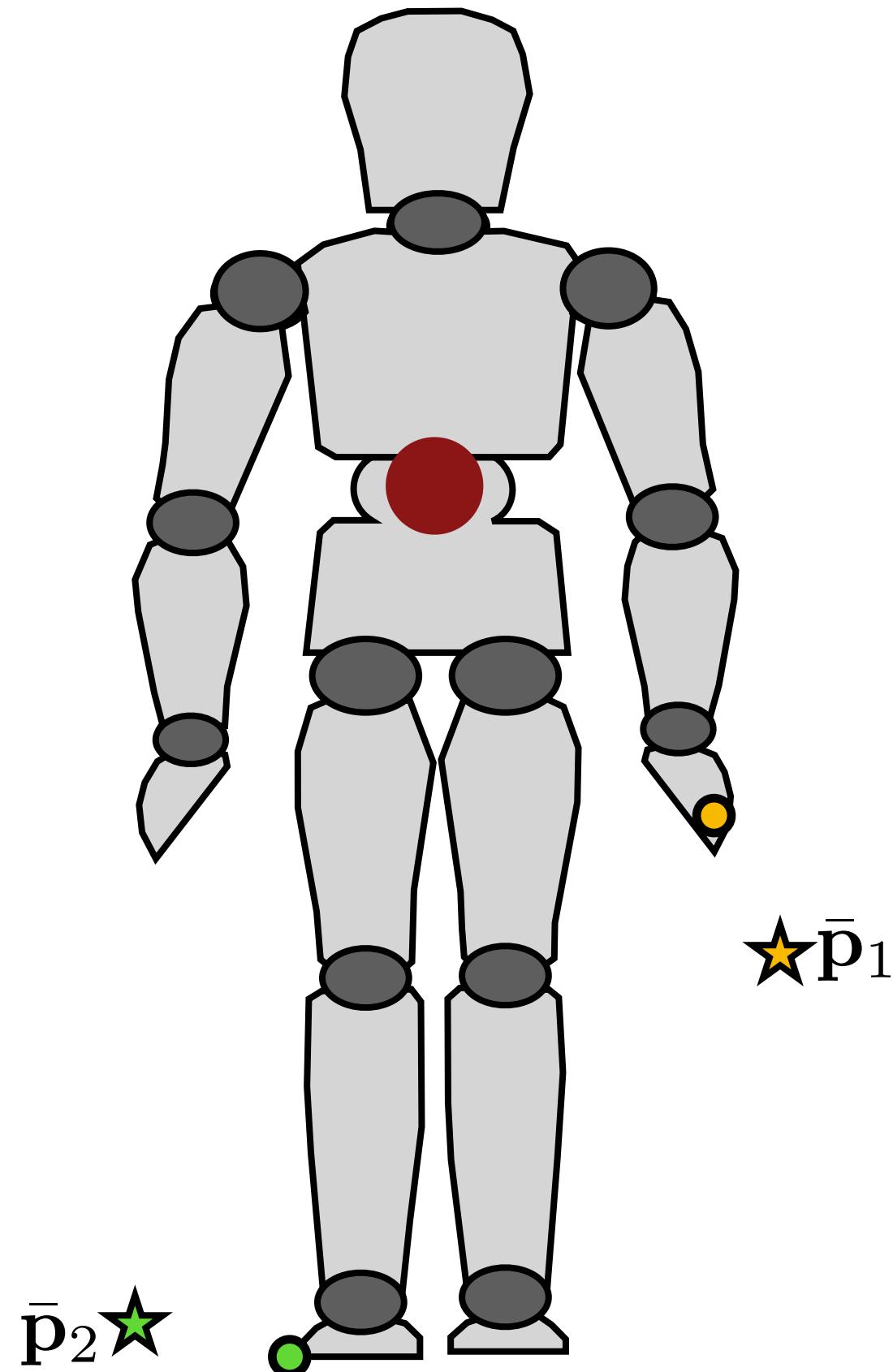


objective function

constraint

Formulate an optimization problem

What if we have multiple constraints?

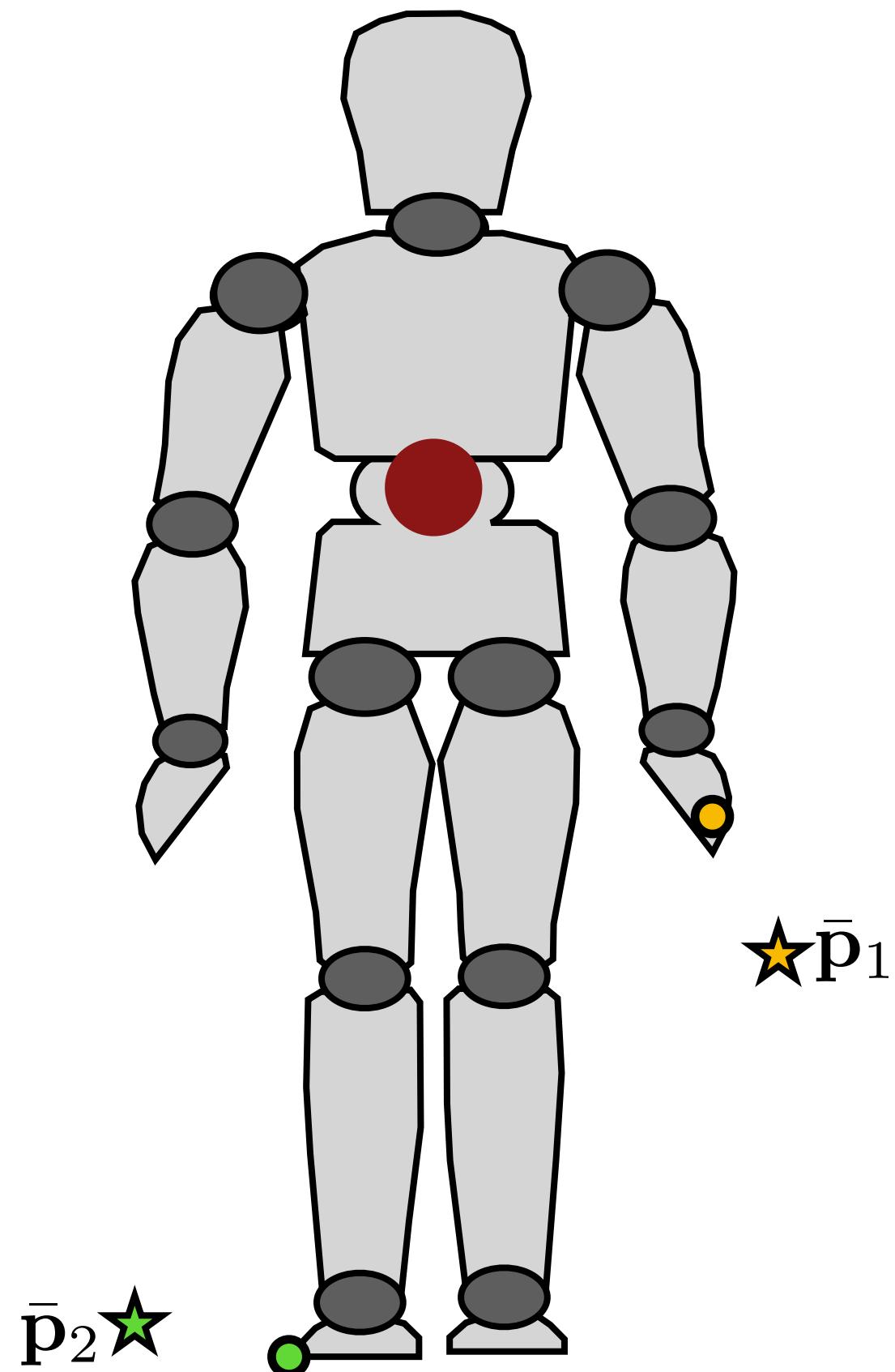


minimize $g(\mathbf{q})$

subject to $\mathbf{C}_1(\mathbf{q}) = \mathbf{0}$

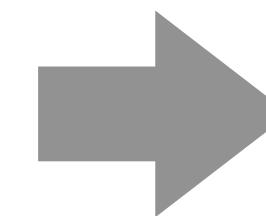
$\mathbf{C}_2(\mathbf{q}) = \mathbf{0}$

Convert to an unconstrained optimization



minimize $g(\mathbf{q})$

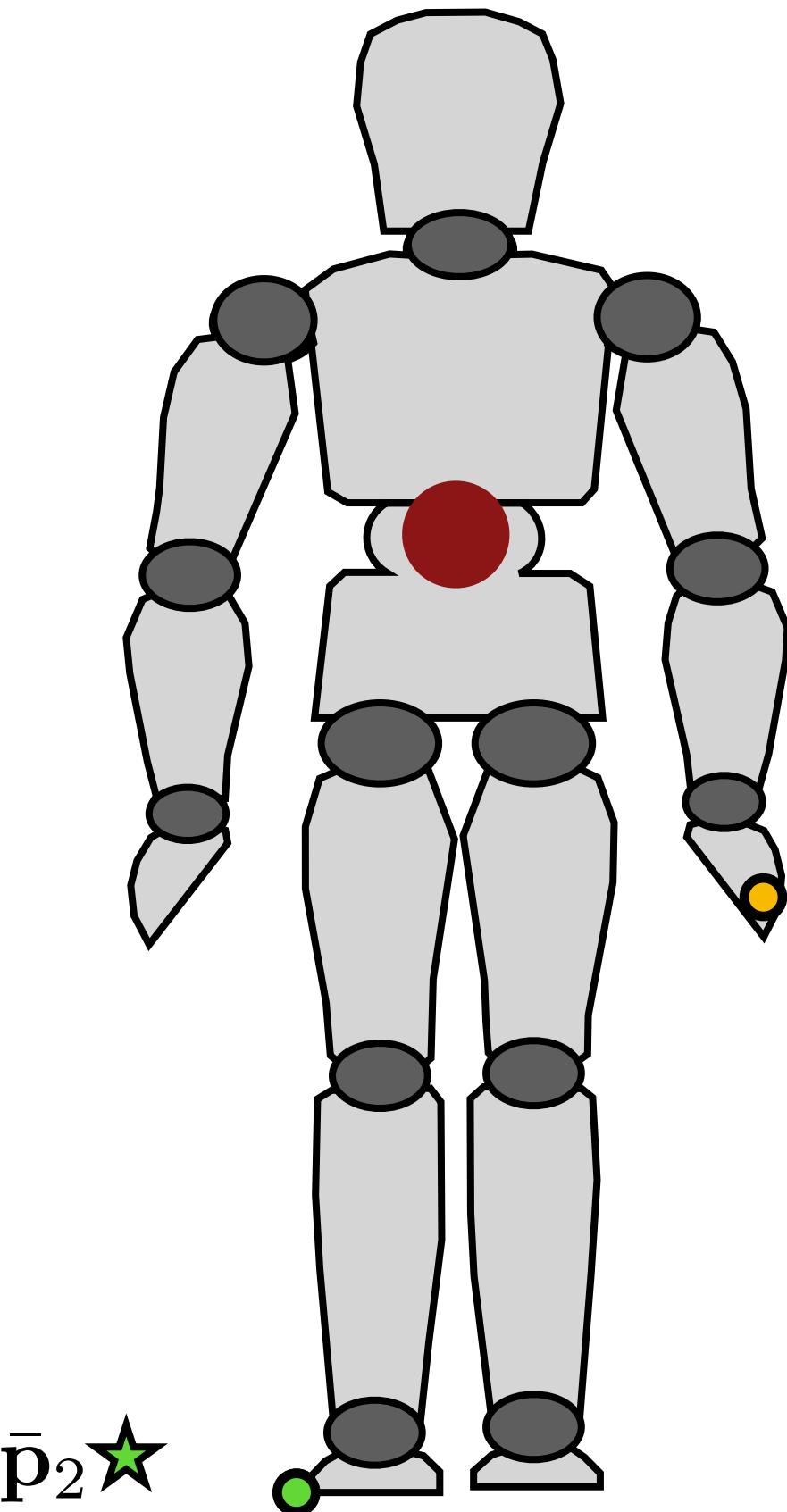
subject to $\mathbf{C}_1(\mathbf{q}) = \mathbf{0}$
 $\mathbf{C}_2(\mathbf{q}) = \mathbf{0}$



minimize $g(\mathbf{q}) + w_1\|\mathbf{C}_1(\mathbf{q})\|^2 + w_2\|\mathbf{C}_2(\mathbf{q})\|^2$

Treat each constraint as a separate metric and minimize weighted sum of all metrics; also called the penalty method

Quiz



★ \bar{p}_1

If $w_1 = w_2$, what will the solution pose look like?

If $w_1 \gg w_2$, what will the solution pose look like?

Gradient descent

How to minimize $g(\mathbf{q}) + w_1\|\mathbf{C}_1(\mathbf{q})\|^2 + w_2\|\mathbf{C}_2(\mathbf{q})\|^2$

while not converged

compute gradient of objective function at current \mathbf{q}

update \mathbf{q} by moving along the negative gradient direction by a small step

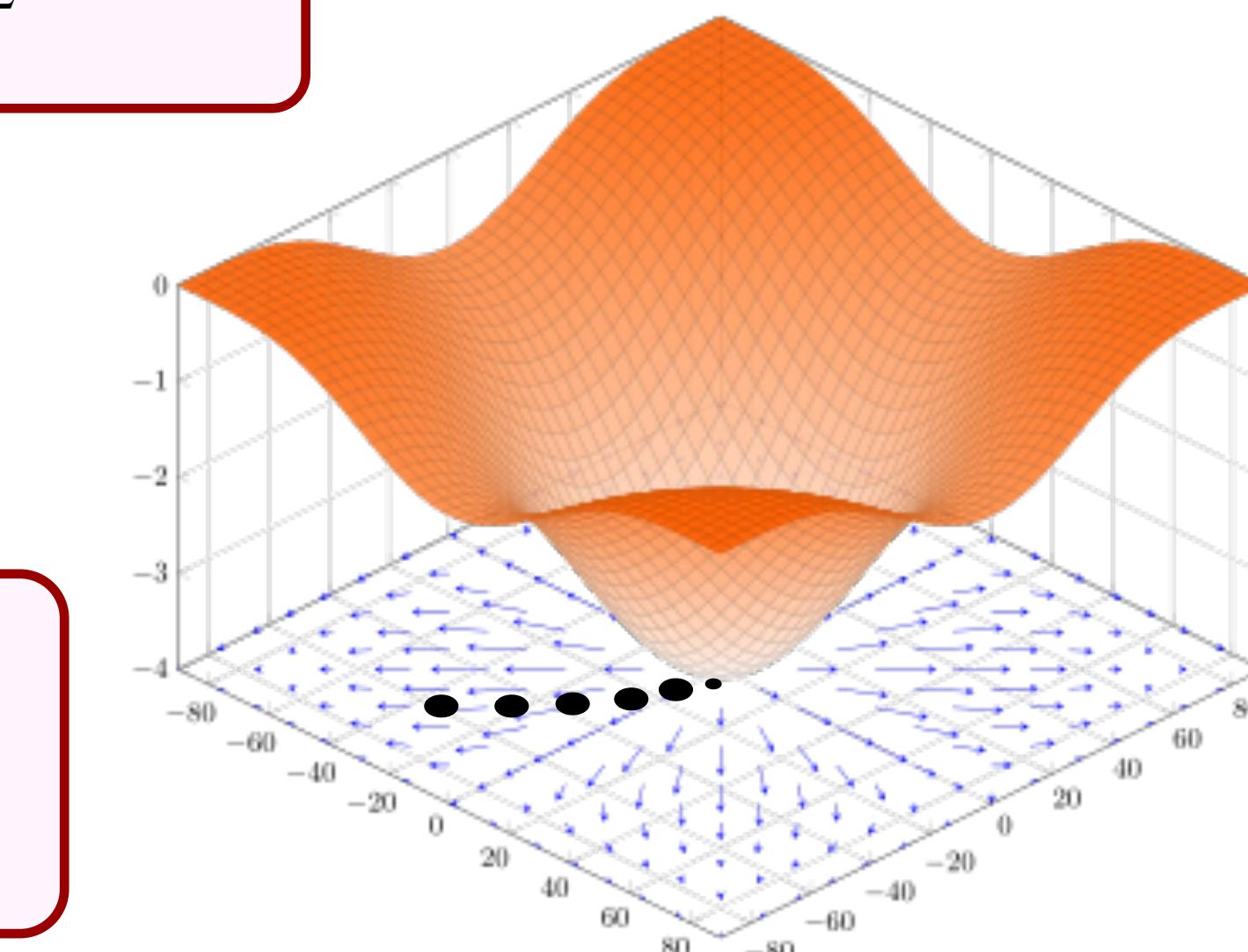
while $\|\mathbf{C}(\mathbf{q})\|^2 > \epsilon$

Take gradient of the objective function,
 $g(\mathbf{q}) + w_1\|\mathbf{C}_1(\mathbf{q})\|^2 + w_2\|\mathbf{C}_2(\mathbf{q})\|^2$

$$\mathbf{d} = \frac{\partial g(\mathbf{q})}{\partial \mathbf{q}} + 2 \sum_i w_i \left[\frac{\partial \mathbf{C}_i(\mathbf{q})}{\partial \mathbf{q}} \right]^T \mathbf{C}_i(\mathbf{q})$$

$$\mathbf{q} = \mathbf{q} - \alpha \mathbf{d}$$

step size α in each search iteration
can be determined arbitrarily, or
through a binary search



Gradient descent



To compute gradient in each optimization iteration,

we need to evaluate the constraint:

$$\mathbf{C}(\mathbf{q})$$

we need to compute the Jacobian matrix:

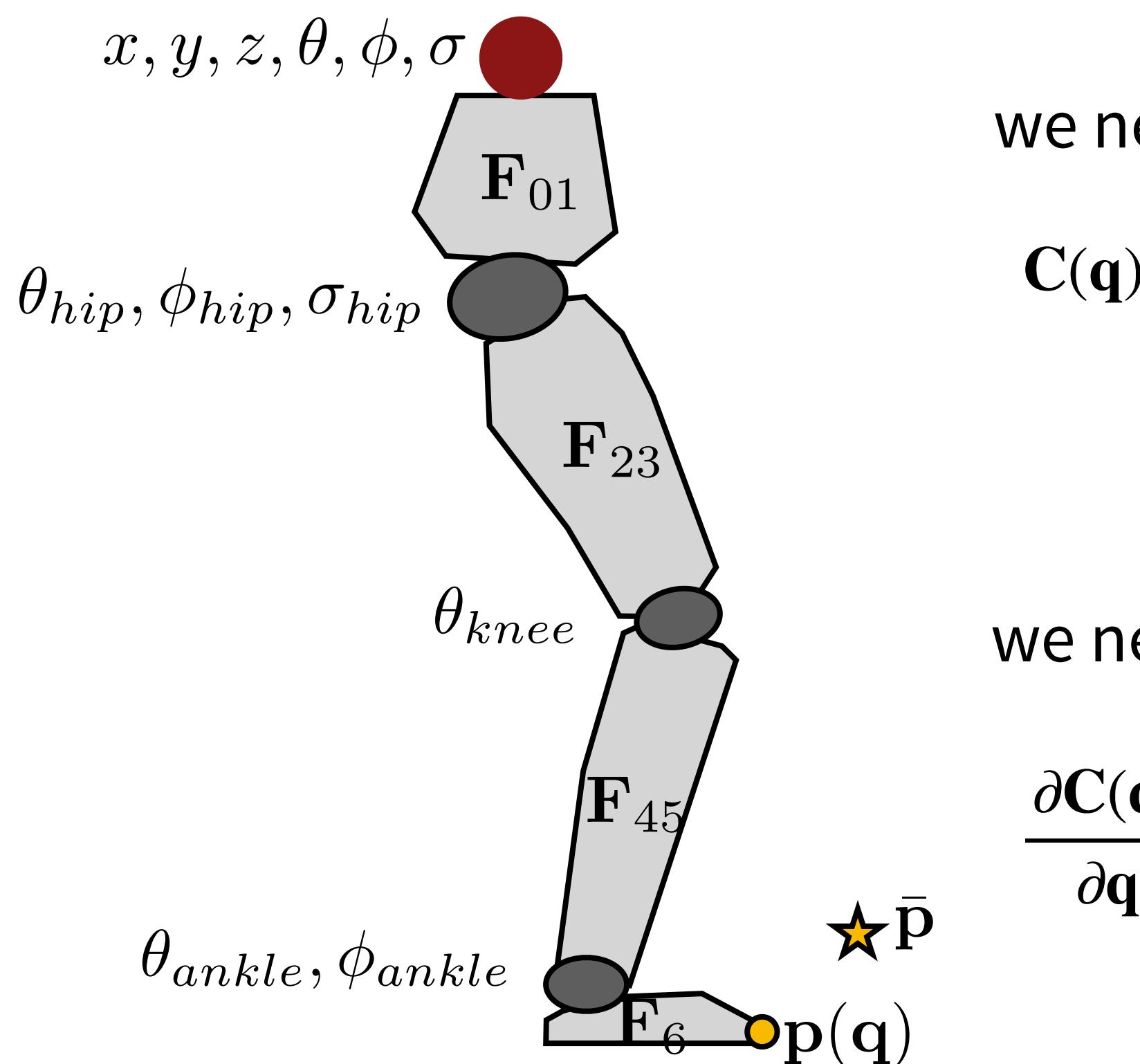
$$\mathbf{d} = \frac{\partial g(\mathbf{q})}{\partial \mathbf{q}} + 2 \sum_i w_i \left[\frac{\partial \mathbf{C}_i(\mathbf{q})}{\partial \mathbf{q}} \right]^T \mathbf{C}_i(\mathbf{q})$$

we also need to evaluate the gradient of function g , but let's ignore that for now.

The leg example

Degrees of freedom in the system: $\mathbf{q} \leftarrow \{x, y, z, \theta, \phi, \sigma, \theta_{hip}, \phi_{hip}, \sigma_{hip}, \theta_{knee}, \theta_{ankle}, \phi_{ankle}, \dots\}$

To compute gradient in each optimization iteration,



we need to evaluate the constraint:

$$\mathbf{C}(\mathbf{q})$$

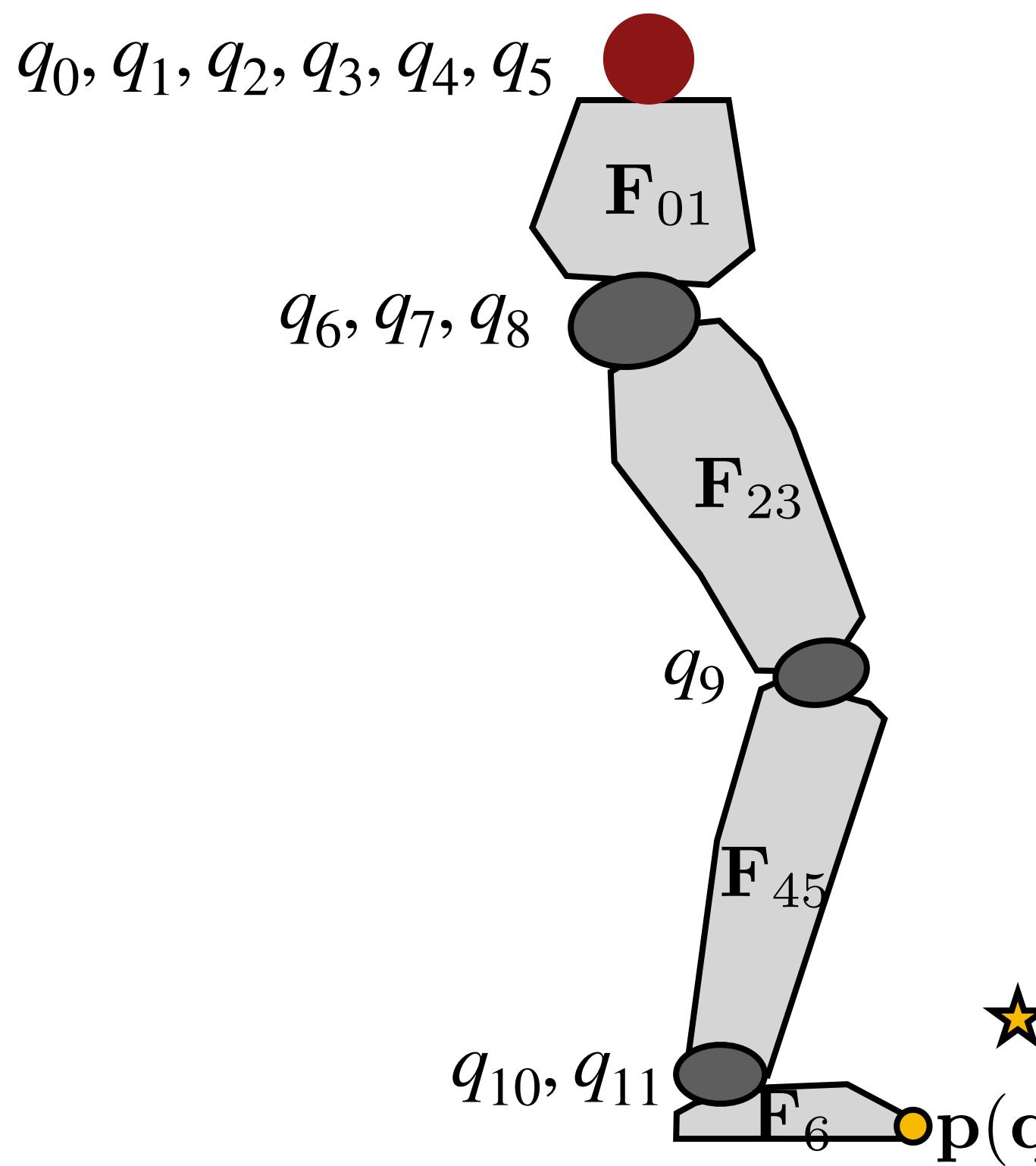
we need to compute the Jacobian matrix:

$$\frac{\partial \mathbf{C}(\mathbf{q})}{\partial \mathbf{q}}$$

The leg example

Degrees of freedom in the system: $\mathbf{q} \leftarrow \{q_0, q_1, q_2, q_3, q_4, q_5, q_6, q_7, q_8, q_9, q_{10}, q_{11}, F_{01}, F_{23}, F_{45}, F_6, p(\mathbf{q}), \phi_{hip}, \phi_{knee}, \phi_{ankle}, \dots\}$

To compute gradient in each optimization iteration,



we need to evaluate the constraint:

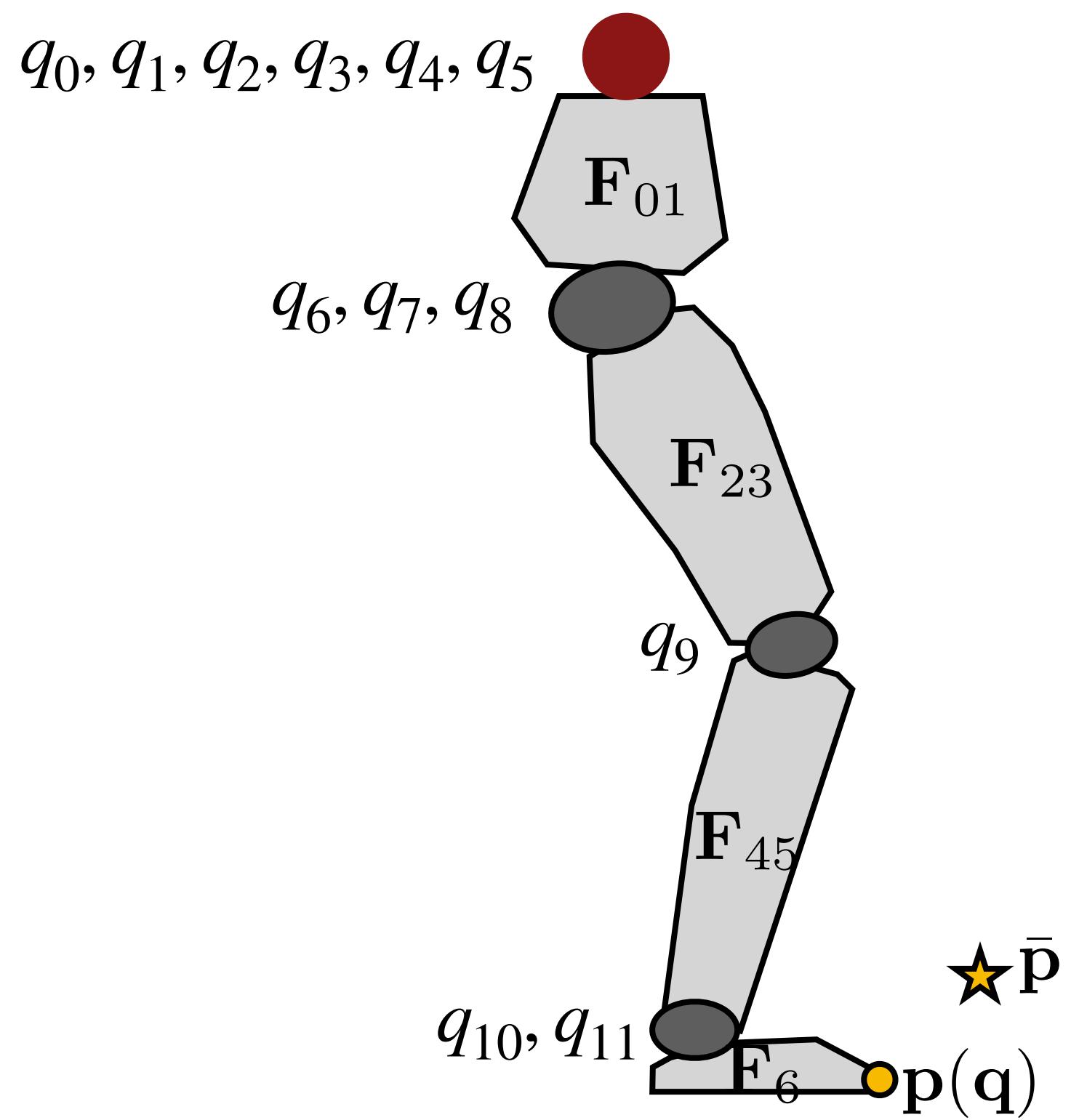
$$\mathbf{C}(\mathbf{q})$$

we need to compute the Jacobian matrix:

$$\frac{\partial \mathbf{C}(\mathbf{q})}{\partial \mathbf{q}}$$

Quiz

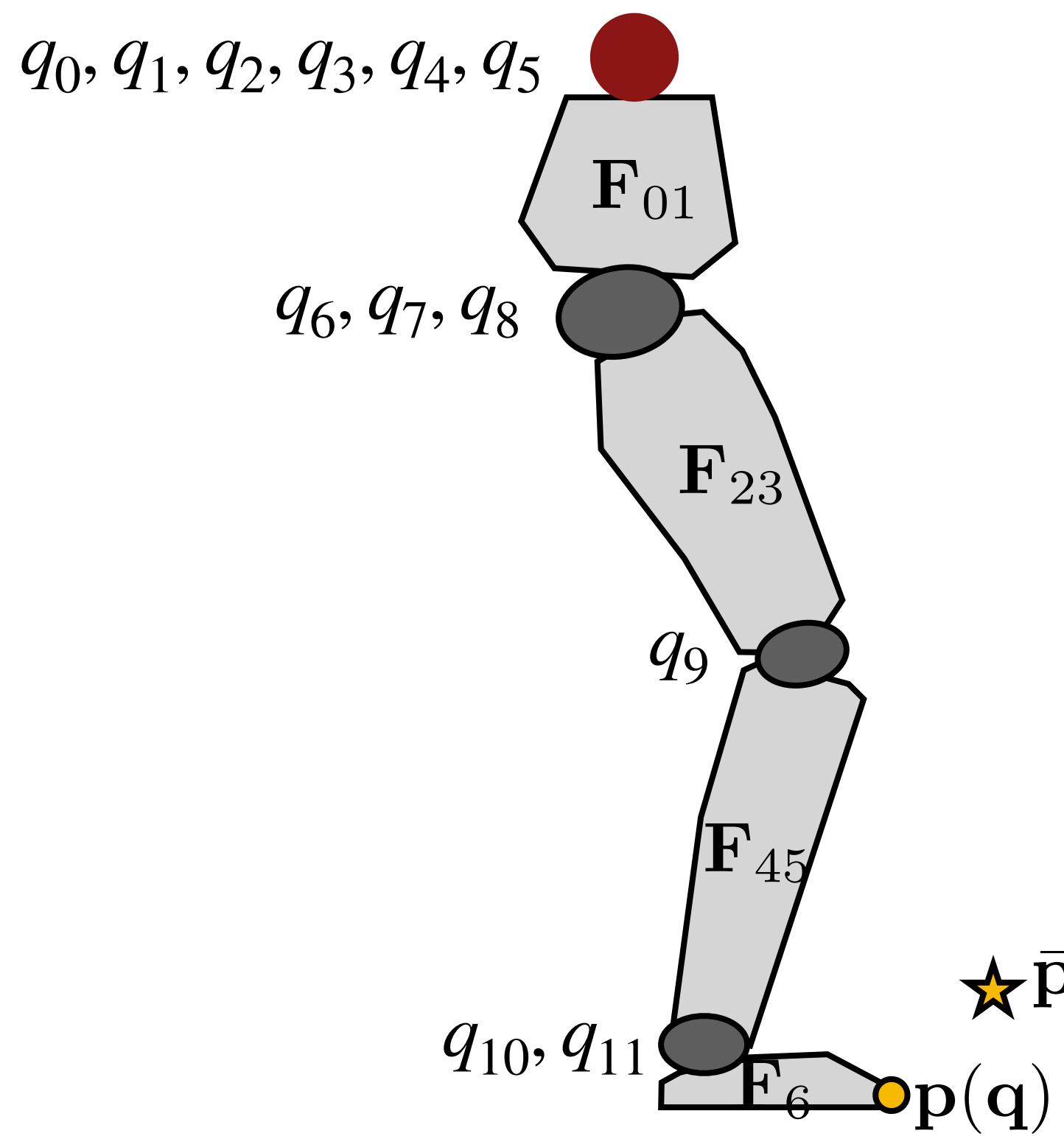
What's the size of Jacobian $\frac{\partial \mathbf{C}(\mathbf{q})}{\partial \mathbf{q}}$?



12
3

Jacobian matrix

$$\mathbf{C}(\mathbf{q}) = \mathbf{T}(q_0, q_1, q_2) \mathbf{R}(q_3, q_4, q_5) \mathbf{F}_{01} \mathbf{R}(q_6) \mathbf{R}(q_7) \mathbf{R}(q_8) \mathbf{F}_{23} \mathbf{R}(q_9) \mathbf{F}_{45} \mathbf{R}(q_{10}) \mathbf{R}(q_{11}) \mathbf{F}_6 \mathbf{p}_0 - \bar{\mathbf{p}}$$



$$\frac{\partial \mathbf{C}(\mathbf{q})}{\partial \mathbf{q}} =$$

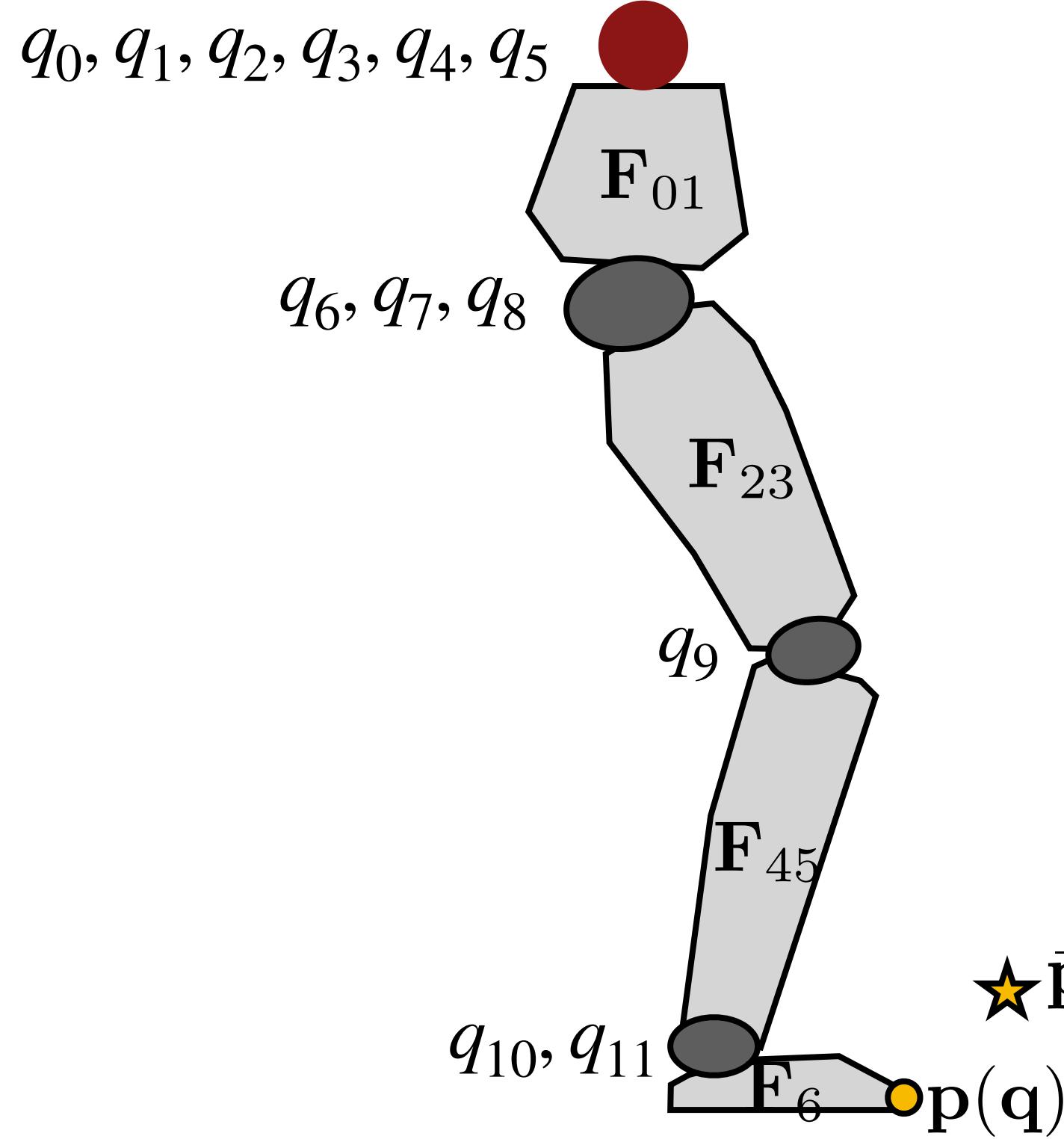
0	1	2	3	4	5	6	7	8	9	10	11

$$\frac{\partial \mathbf{C}(\mathbf{q})}{\partial q_1} = \frac{\partial \mathbf{T}(q_0, q_1, q_2)}{\partial q_1} \mathbf{R}(q_3, q_4, q_5) \mathbf{F}_{01} \mathbf{R}(q_6) \mathbf{R}(q_7) \mathbf{R}(q_8) \mathbf{F}_{23} \mathbf{R}(q_9) \mathbf{F}_{45} \mathbf{R}(q_{10}) \mathbf{R}(q_{11}) \mathbf{F}_6 \mathbf{p}_0$$

$$\frac{\partial \mathbf{C}(\mathbf{q})}{\partial q_5} = \mathbf{T}(q_0, q_1, q_2) \frac{\partial \mathbf{R}(q_3, q_4, q_5)}{\partial q_5} \mathbf{F}_{01} \mathbf{R}(q_6) \mathbf{R}(q_7) \mathbf{R}(q_8) \mathbf{F}_{23} \mathbf{R}(q_9) \mathbf{F}_{45} \mathbf{R}(q_{10}) \mathbf{R}(q_{11}) \mathbf{F}_6 \mathbf{p}_0$$

$$\frac{\partial \mathbf{C}(\mathbf{q})}{\partial q_{10}} = \mathbf{T}(q_0, q_1, q_2) \mathbf{R}(q_3, q_4, q_5) \mathbf{F}_{01} \mathbf{R}(q_6) \mathbf{R}(q_7) \mathbf{R}(q_8) \mathbf{F}_{23} \mathbf{R}(q_9) \mathbf{F}_{45} \frac{\partial \mathbf{R}(q_{10})}{\partial q_{10}} \mathbf{R}(q_{11}) \mathbf{F}_6 \mathbf{p}_0$$

Jacobian matrix



$$C(\mathbf{q}) = \mathbf{T}(q_0, q_1, q_2)\mathbf{R}(q_3, q_4, q_5)\mathbf{F}_{01}\mathbf{R}(q_6)\mathbf{R}(q_7)\mathbf{R}(q_8)\mathbf{F}_{23}\mathbf{R}(q_9)\mathbf{F}_{45}\mathbf{I}$$

$$\frac{\partial \mathbf{T}(q_0, q_1, q_2)}{\partial q_1} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

$$\frac{\partial \mathbf{R}(q_{10})}{\partial q_{10}} = \begin{pmatrix} -\sin q_{10} & -\cos q_{10} & 0 & 0 \\ \cos q_{10} & -\sin q_{10} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

(assuming q_{10} is rotating about z-axis in the ankle frame)

What about $\frac{\partial \mathbf{R}(q_3, q_4, q_5)}{\partial q_5}$?

recall affine transformations

$$\begin{bmatrix} x + t_x \\ y + t_y \\ z + t_z \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta & 0 & 0 \\ \sin \theta & \cos \theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

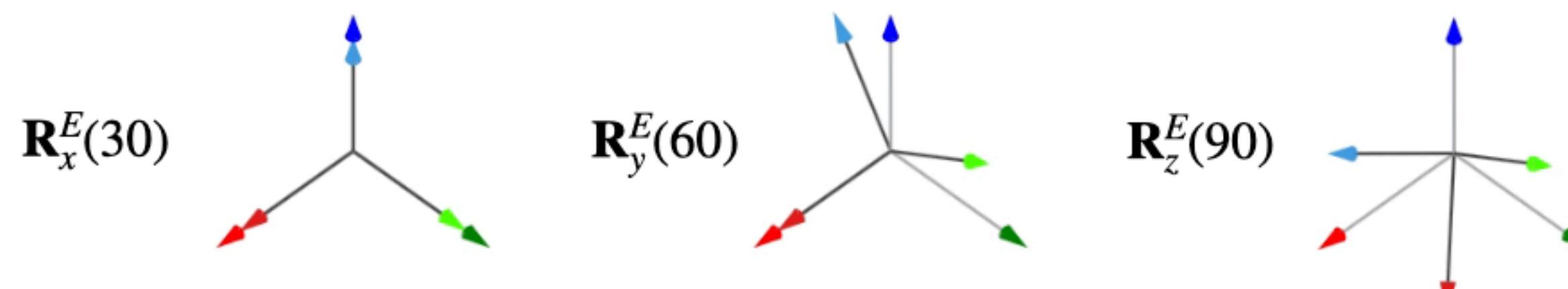
Quiz

- If a 3-dof joint, $\mathbf{R}(q_3, q_4, q_5)$ is represented by three Euler angles, what is $\frac{\partial \mathbf{R}(q_3, q_4, q_5)}{\partial q_5}$?

$$\mathbf{R}(q_3)\mathbf{R}(q_4)\frac{\partial \mathbf{R}(q_5)}{\partial q_5}$$

recall Euler Angle representation

Euler angle: axes move with object



Quiz

- If a 3-dof joint, $\mathbf{R}(q_x, q_y, q_z, q_w)$ is represented as q quaternion, what is

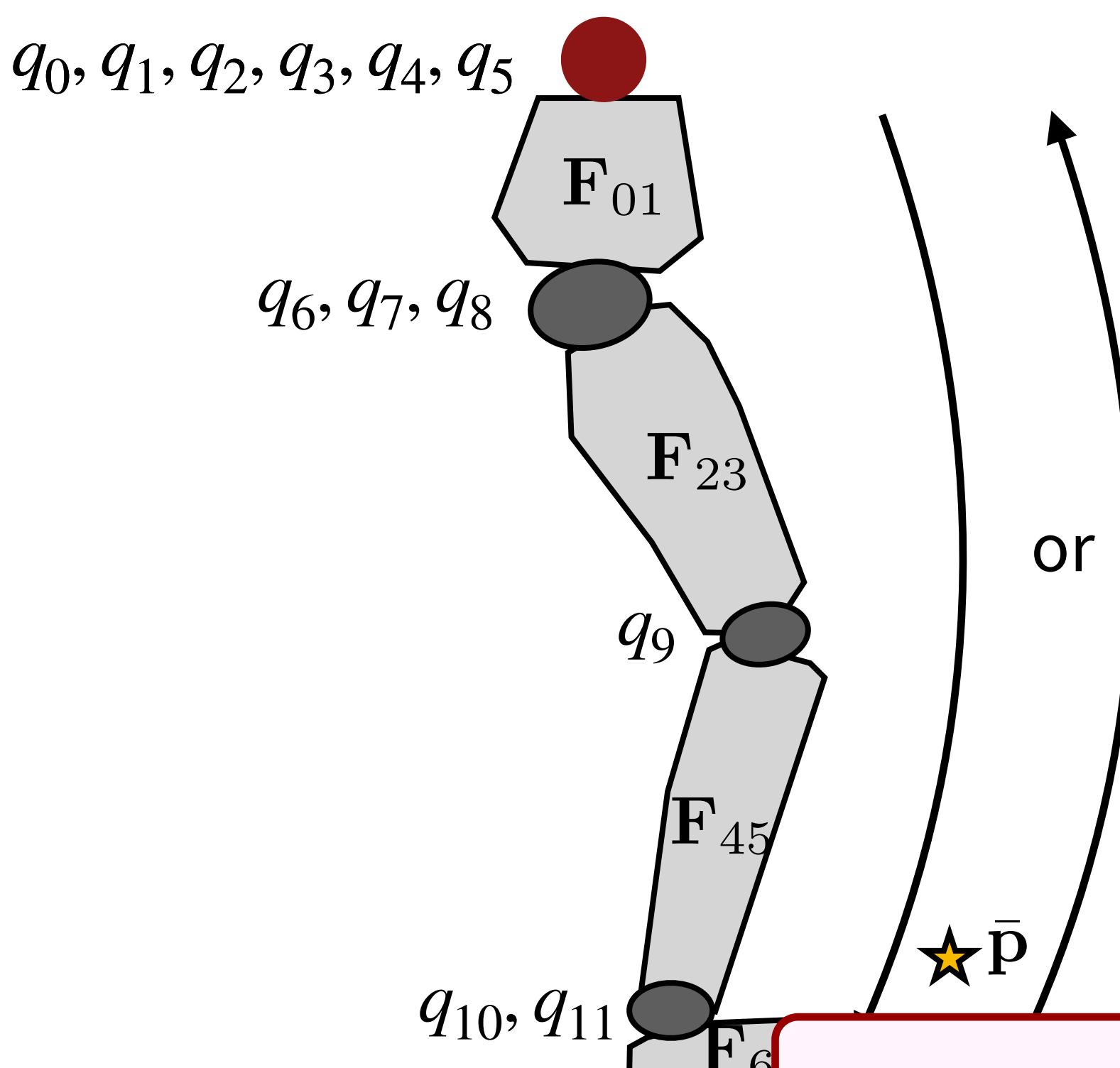
$$\frac{\partial \mathbf{R}(q_x, q_y, q_z, q_w)}{\partial q_y}?$$

recall quaternion in matrix form

$$\mathbf{R}(\mathbf{q}) = \begin{bmatrix} 1 - 2y^2 - 2z^2 & 2xy - 2wz & 2xz + 2wy & 0 \\ 2xy + 2wz & 1 - 2x^2 - 2z^2 & 2yz - 2wx & 0 \\ 2xz - 2wy & 2yz + 2wx & 1 - 2x^2 - 2y^2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Jacobian matrix

What's the most efficient order of dofs to compute Jacobian?



already computed in
graphics matrix stack

Down the kinematic chain?

$$\begin{aligned} \frac{\partial \mathbf{C}(\mathbf{q})}{\partial q_8} &= \mathbf{T}(q_0, q_1, q_2) \mathbf{R}(q_3, q_4, q_5) \mathbf{F}_{01} \mathbf{R}(q_6) \mathbf{R}(q_7) \frac{\partial \mathbf{R}(q_8)}{\partial q_8} \mathbf{F}_{23} \mathbf{R}(q_9) \mathbf{F}_{45} \mathbf{R}(q_{10}) \mathbf{R}(q_{11}) \mathbf{F}_6 \mathbf{p}_0 \\ &= \mathbf{M} \mathbf{R}(q_6) \mathbf{R}(q_7) \frac{\partial \mathbf{R}(q_8)}{\partial q_8} \mathbf{F}_{23} \mathbf{R}(q_9) \mathbf{F}_{45} \mathbf{R}(q_{10}) \mathbf{R}(q_{11}) \mathbf{I}_{3 \times 3} \end{aligned}$$

pass down from parent

Up the kinematic chain?

$$\begin{aligned} \frac{\partial \mathbf{C}(\mathbf{q})}{\partial q_8} &= \mathbf{T}(q_0, q_1, q_2) \mathbf{R}(q_3, q_4, q_5) \mathbf{F}_{01} \mathbf{R}(q_6) \mathbf{R}(q_7) \frac{\partial \mathbf{R}(q_8)}{\partial q_8} \mathbf{F}_{23} \mathbf{R}(q_9) \mathbf{F}_{45} \mathbf{R}(q_{10}) \mathbf{R}(q_{11}) \mathbf{F}_6 \mathbf{p}_0 \\ &= -\mathbf{T}(q_0, q_1, q_2) \mathbf{R}(q_3, q_4, q_5) \mathbf{F}_{01} \mathbf{R}(q_6) \mathbf{R}(q_7) \frac{\partial \mathbf{R}(q_8)}{\partial q_8} \mathbf{v} \end{aligned}$$

pass up from child



Better because parent transformation \mathbf{M} is already stored in the graphics matrix stack, and passing \mathbf{v} is cheaper than passing \mathbf{M} .

Gradient descent

How to minimize $g(\mathbf{q}) + w_1 \|\mathbf{C}_1(\mathbf{q})\|^2 + w_2 \|\mathbf{C}_2(\mathbf{q})\|^2$

while not converged

compute gradient of objective function at current \mathbf{q}

update \mathbf{q} by moving along the negative gradient direction by a small step

while $\|\mathbf{C}(\mathbf{q})\|^2 > \epsilon$

$$\mathbf{d} = \frac{\partial g(\mathbf{q})}{\partial \mathbf{q}} + 2 \sum_i w_i \left[\frac{\partial \mathbf{C}_i(\mathbf{q})}{\partial \mathbf{q}} \right]^T \mathbf{C}_i(\mathbf{q})$$

$$\mathbf{q} = \mathbf{q} - \alpha \mathbf{d}$$

An easier way to approximate the gradient is called Finite Differencing

Gradient descent

$$\text{minimize} \quad g(\mathbf{q}) + w_1 \|\mathbf{C}_1(\mathbf{q})\|^2 + w_2 \|\mathbf{C}_2(\mathbf{q})\|^2$$

Gradient descent

minimize $f(\mathbf{q})$

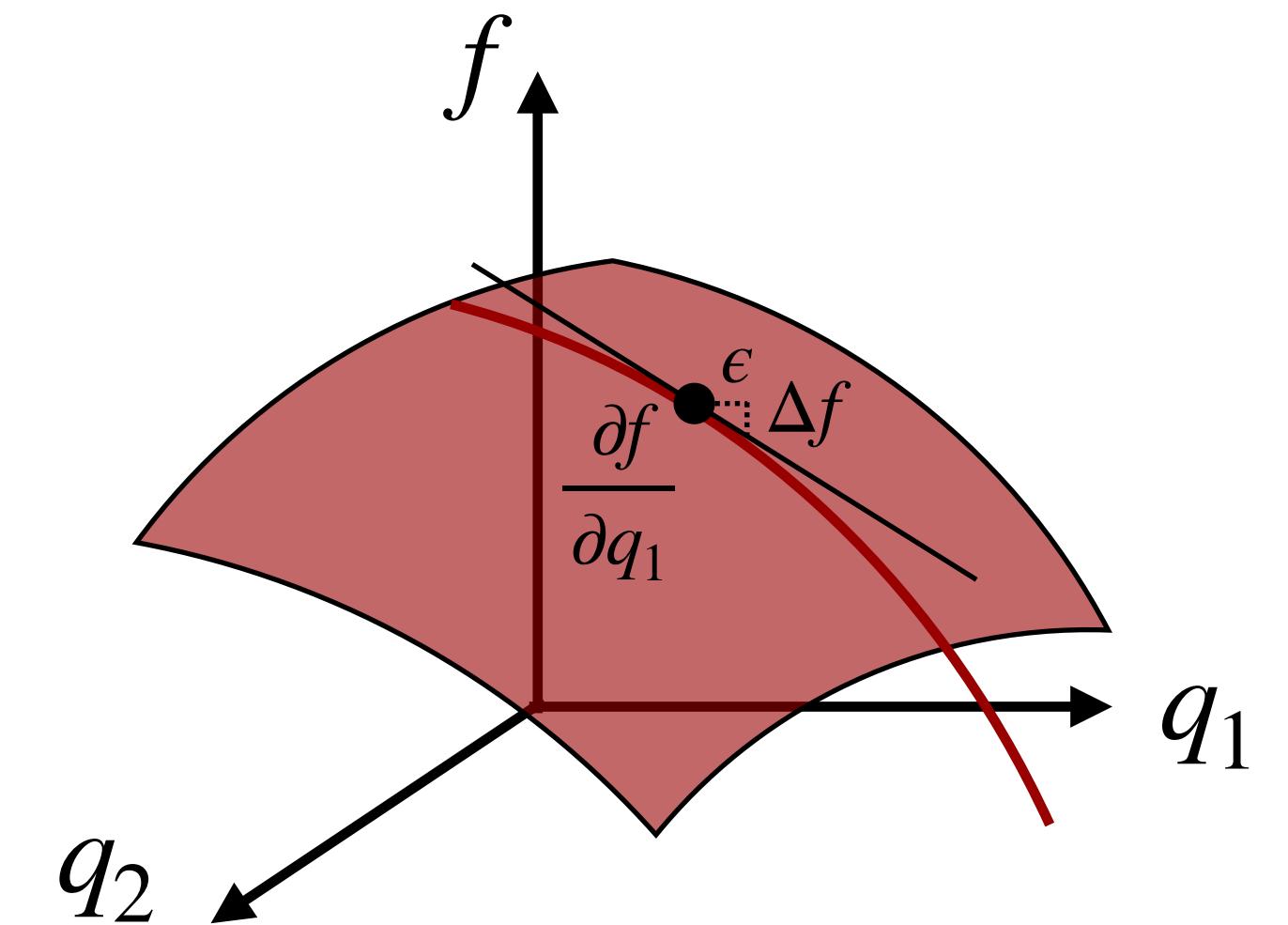
where $f(\mathbf{q}) \equiv g(\mathbf{q}) + w_1 \|\mathbf{C}_1(\mathbf{q})\|^2 + w_2 \|\mathbf{C}_2(\mathbf{q})\|^2$

$$\frac{\partial f(\mathbf{q})}{\partial \mathbf{q}} = \left[\frac{\partial f(\mathbf{q})}{\partial q_1}, \frac{\partial f(\mathbf{q})}{\partial q_2}, \frac{\partial f(\mathbf{q})}{\partial q_3}, \dots \right]$$

$$\frac{\partial f(\mathbf{q})}{\partial q_1} \approx \frac{\Delta f}{\Delta q_1} = \frac{f(\mathbf{q} + \begin{bmatrix} \epsilon \\ 0 \\ 0 \end{bmatrix}) - f(\mathbf{q})}{\epsilon}$$

$$\frac{\partial f(\mathbf{q})}{\partial q_2} \approx ?$$

$$\frac{\partial f(\mathbf{q})}{\partial q_3} \approx ?$$



Additional objective term

$$\text{minimize } g(\mathbf{q}) + \sum_i w_i \|\mathbf{C}_i(\mathbf{q})\|^2$$

$g(\mathbf{q})$ could be maintaining a desired pose

desired pose

$$g(\mathbf{q}) = \|\mathbf{q} - \bar{\mathbf{q}}\|^2$$

$g(\mathbf{q})$ could be minimizing the velocity of joint angles

previous pose

$$g(\mathbf{q}) = \|\mathbf{q} - \mathbf{q}'\|^2$$

$g(\mathbf{q})$ could be maintaining static balance

projected COM

center of supported area

$$g(\mathbf{q}) = \|f_{com}(\mathbf{q}) - f_{feet}(\mathbf{q})\|^2$$

Quiz

Compute the gradient of $g(\mathbf{g})$ defined as follows:

$g(\mathbf{q})$ could be maintaining a desired pose

$$g(\mathbf{q}) = \|\mathbf{q} - \bar{\mathbf{q}}\|^2$$

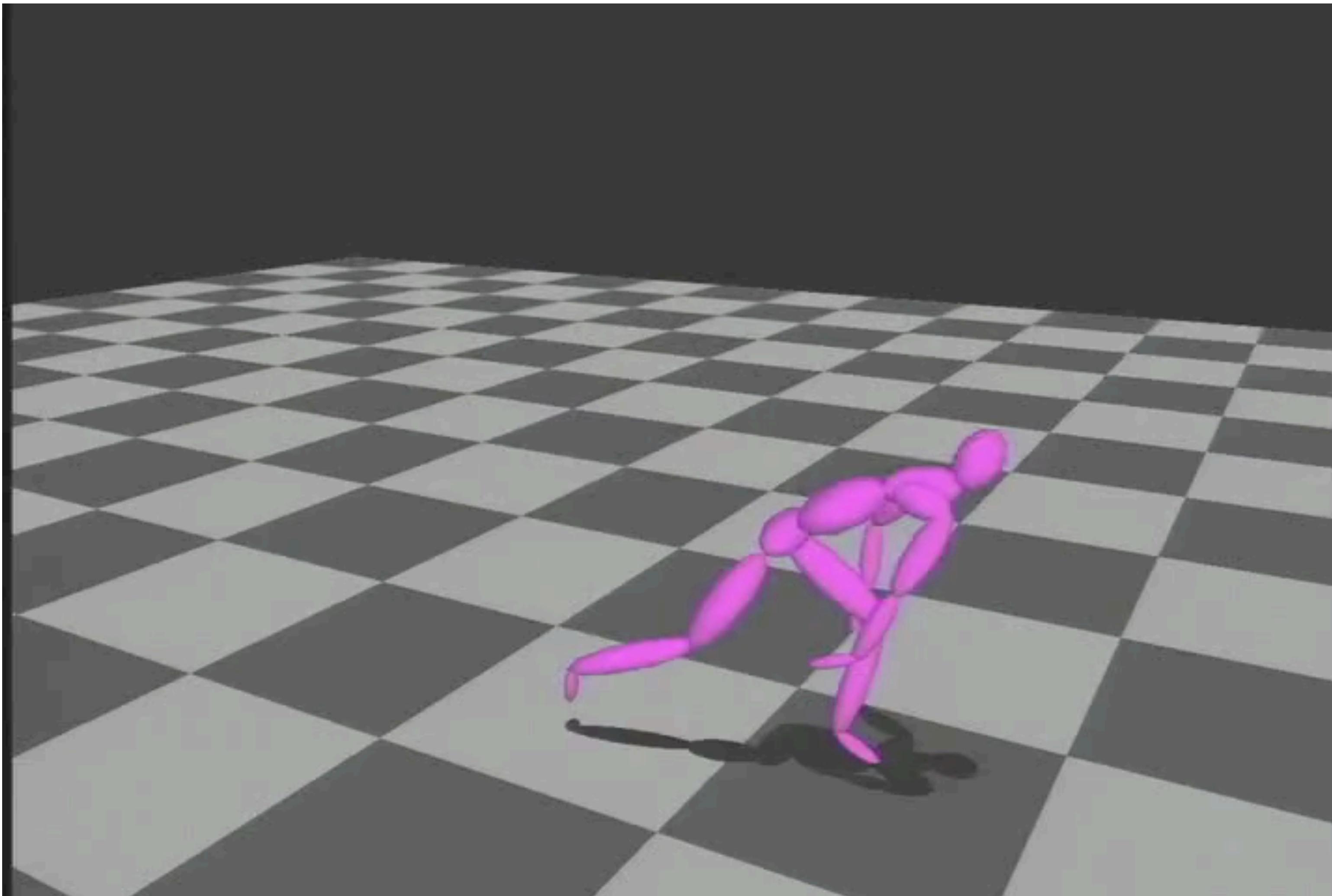
$g(\mathbf{q})$ could be minimizing the velocity of joint angles

$$g(\mathbf{q}) = \|\dot{\mathbf{q}} - \dot{\mathbf{q}}'\|^2$$

$g(\mathbf{q})$ could be maintaining static balance

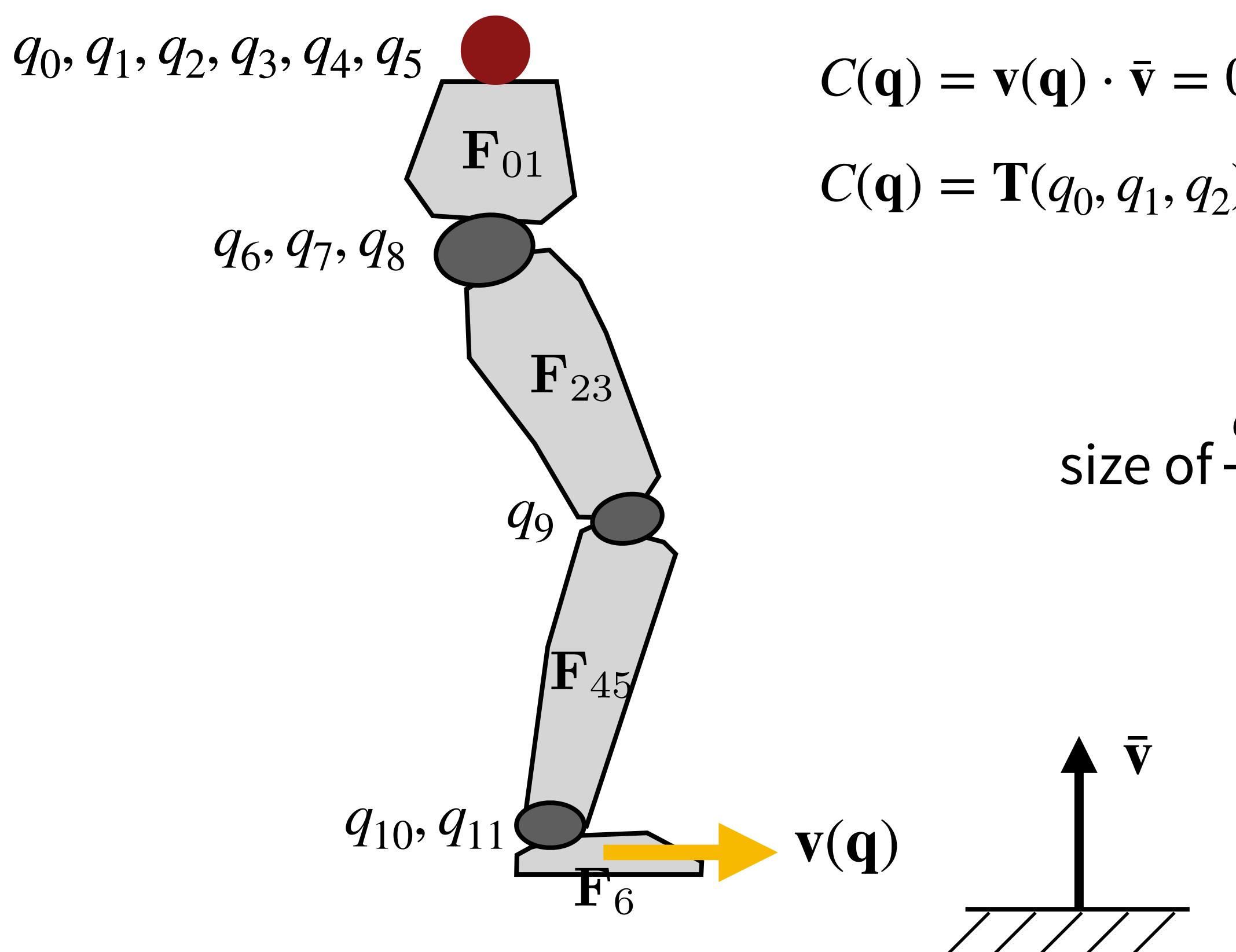
$$g(\mathbf{q}) = \|f_{com}(\mathbf{q}) - f_{feet}(\mathbf{q})\|^2$$

Realistic human poses



Orientation constraints

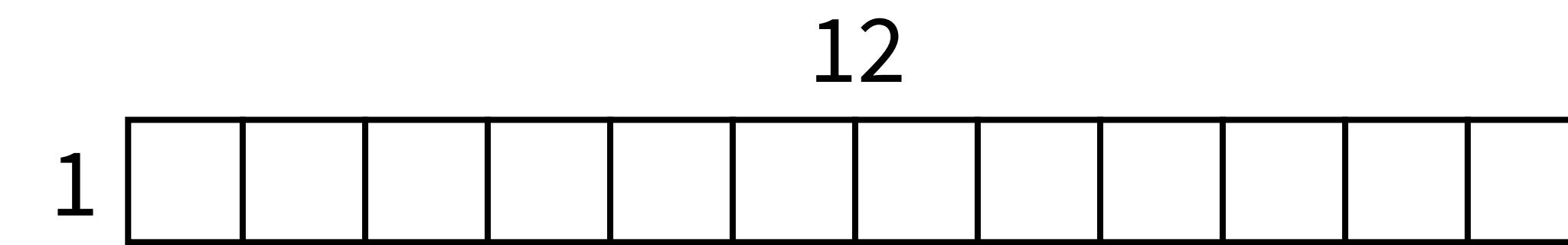
How to formulate a constraint that enforces the foot parallel to the ground?



$$C(\mathbf{q}) = \mathbf{v}(\mathbf{q}) \cdot \bar{\mathbf{v}} = 0$$

$$C(\mathbf{q}) = \mathbf{T}(q_0, q_1, q_2)\mathbf{R}(q_3, q_4, q_5)\mathbf{F}_{01}\mathbf{R}(q_6)\mathbf{R}(q_7)\mathbf{R}(q_8)\mathbf{F}_{23}\mathbf{R}(q_9)\mathbf{F}_{45}\mathbf{R}(q_{10})\mathbf{R}(q_{11})\mathbf{F}_6\mathbf{v}_0 \cdot \bar{\mathbf{v}} = 0$$

size of $\frac{\partial C}{\partial \mathbf{q}}$



\mathbf{v}_0 is a vector parallel to the bottom of the foot in the foot frame. Note the fourth element of \mathbf{v}_0 is zero. Why?

Alternative approach: Analytical IK

- Generate source code to solve a nonlinear system symbolically

$$f(\mathbf{q}) = \mathbf{p}$$

- Suitable for robotic manipulation problems in which

$$\dim(\mathbf{q}) \sim= \dim(\mathbf{p})$$

- Return multiple solutions,

IK: $\mathbf{p} \mapsto \mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3 \dots$

Alternative approach: Iterative IK

Jacobi inverse method:

$$\mathbf{J} = \frac{\partial \mathbf{C}}{\partial \mathbf{q}} \approx \frac{\Delta \mathbf{e}}{\Delta \mathbf{q}}$$

$$\mathbf{J}\Delta \mathbf{q} \approx \Delta \mathbf{e}$$

Moore-Penrose inverse matrix

$$\Delta \mathbf{q} = \mathbf{J}^+ \Delta \mathbf{e}$$

$$\Delta \mathbf{q} = (\mathbf{J}^T \mathbf{J})^{-1} \mathbf{J}^T \Delta \mathbf{e} \quad \text{or} \quad \Delta \mathbf{q} = \mathbf{J}^T (\mathbf{J} \mathbf{J}^T)^{-1} \Delta \mathbf{e}$$

$$\mathbf{q} = \mathbf{q} + \Delta \mathbf{q}$$

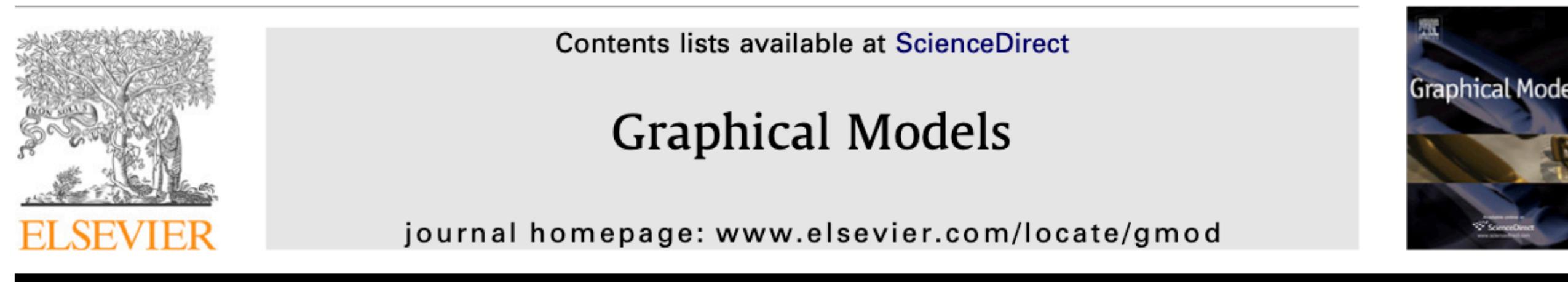
If \mathbf{J} is tall and thin, use this one.

If \mathbf{J} is short and fat, use this one.

Logistics

- The due date of HW3 is extended to tonight 11:59pm
- Project 3 is due on 12/9 11:59PM
- HW4 will be released by tomorrow and due on 12/7 11:59pm
- About the final exam:
 - It will be a 3-hour in-person exam
 - You can bring in any printed materials
 - No internet, no computers, no electronic devices
 - It will be strictly within the materials covered by slides (Unlike homework, where some problems encourage you to think beyond the lecture materials)
 - We will have a review session on next Thursday during the lecture time

Alternative: Heuristic-based IK



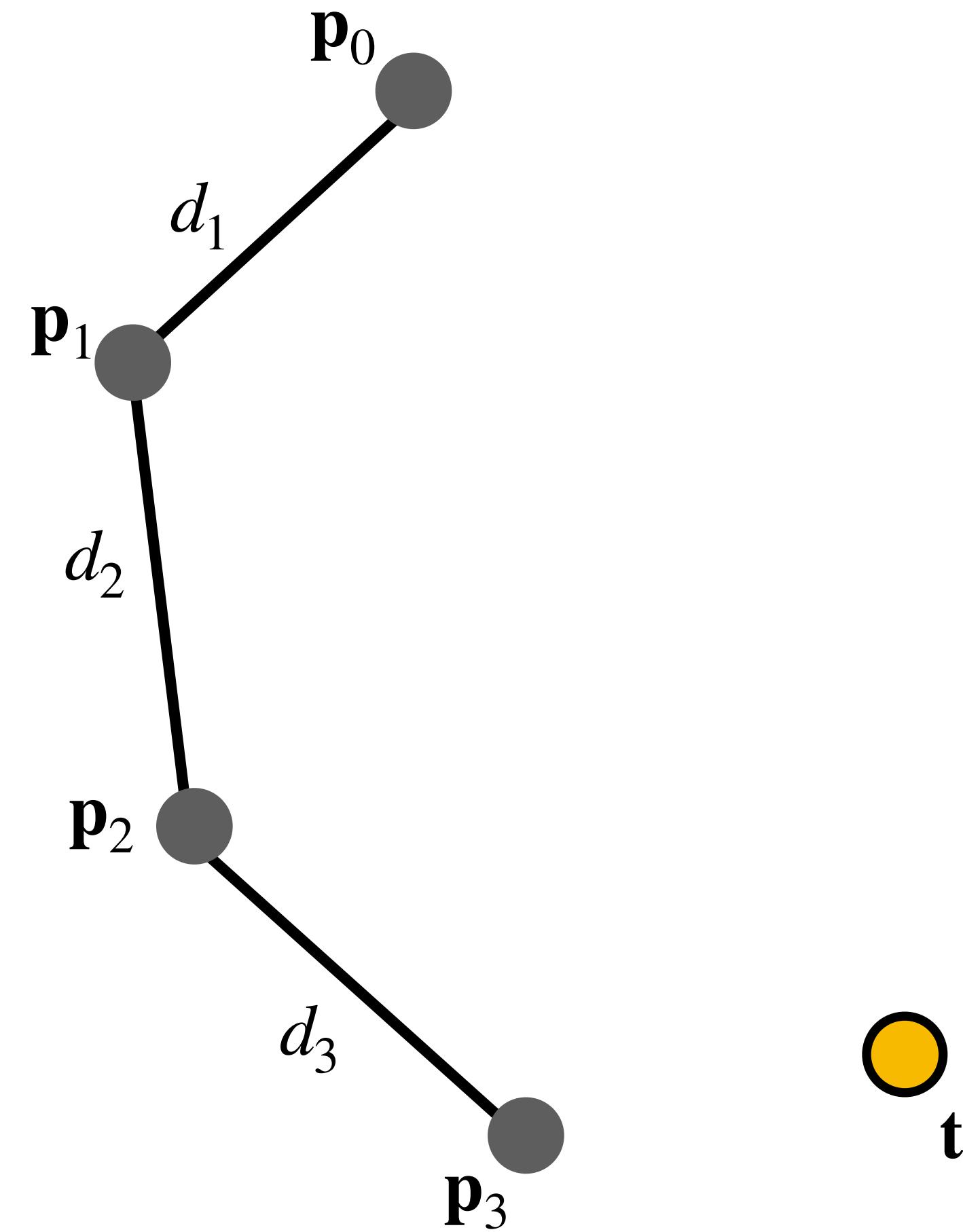
FABRIK: A fast, iterative solver for the Inverse Kinematics problem [☆]

Andreas Aristidou ^{*}, Joan Lasenby

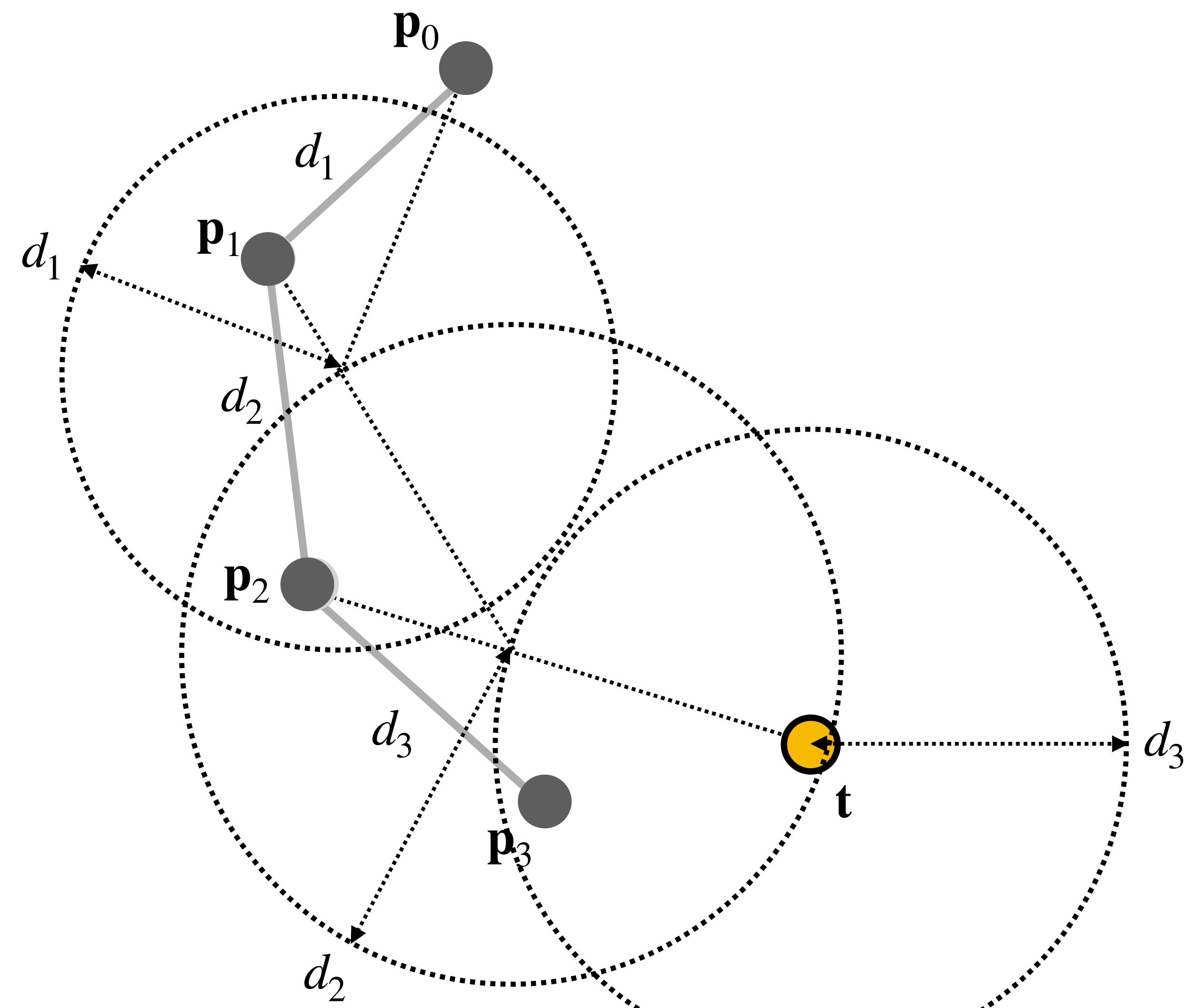
Department of Engineering, University of Cambridge, Cambridge CB2 1PZ, UK

- **Forward And Backward Reaching Inverse Kinematics (FABRIK).**
- **FABRIK finds each joint position via locating a point on a line, and converges in few iterations, has low computational cost and produces visually realistic poses.**
- **Constraints can be incorporated within FABRIK and multiple chains with multiple end effectors are also supported.**

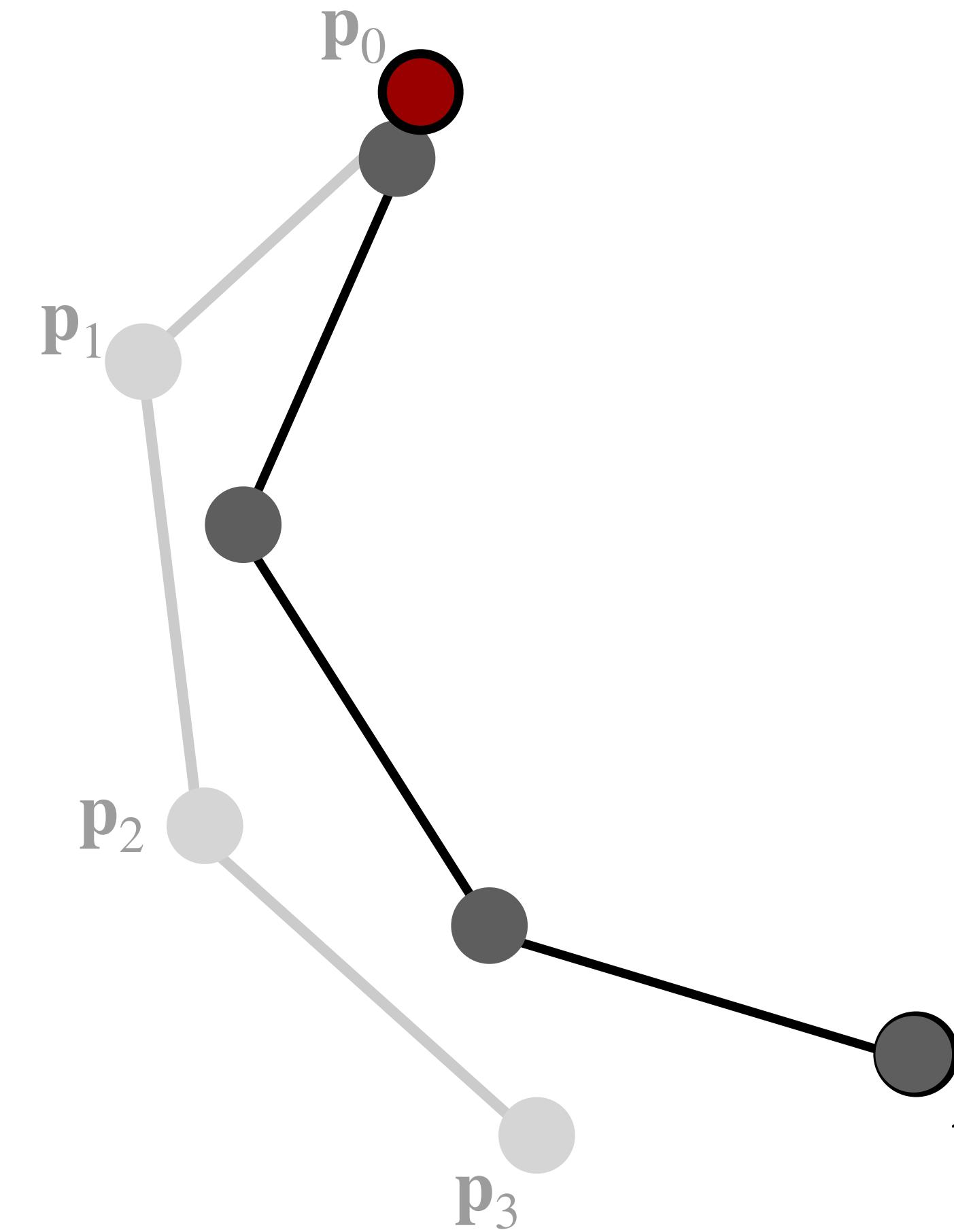
Basic algorithm



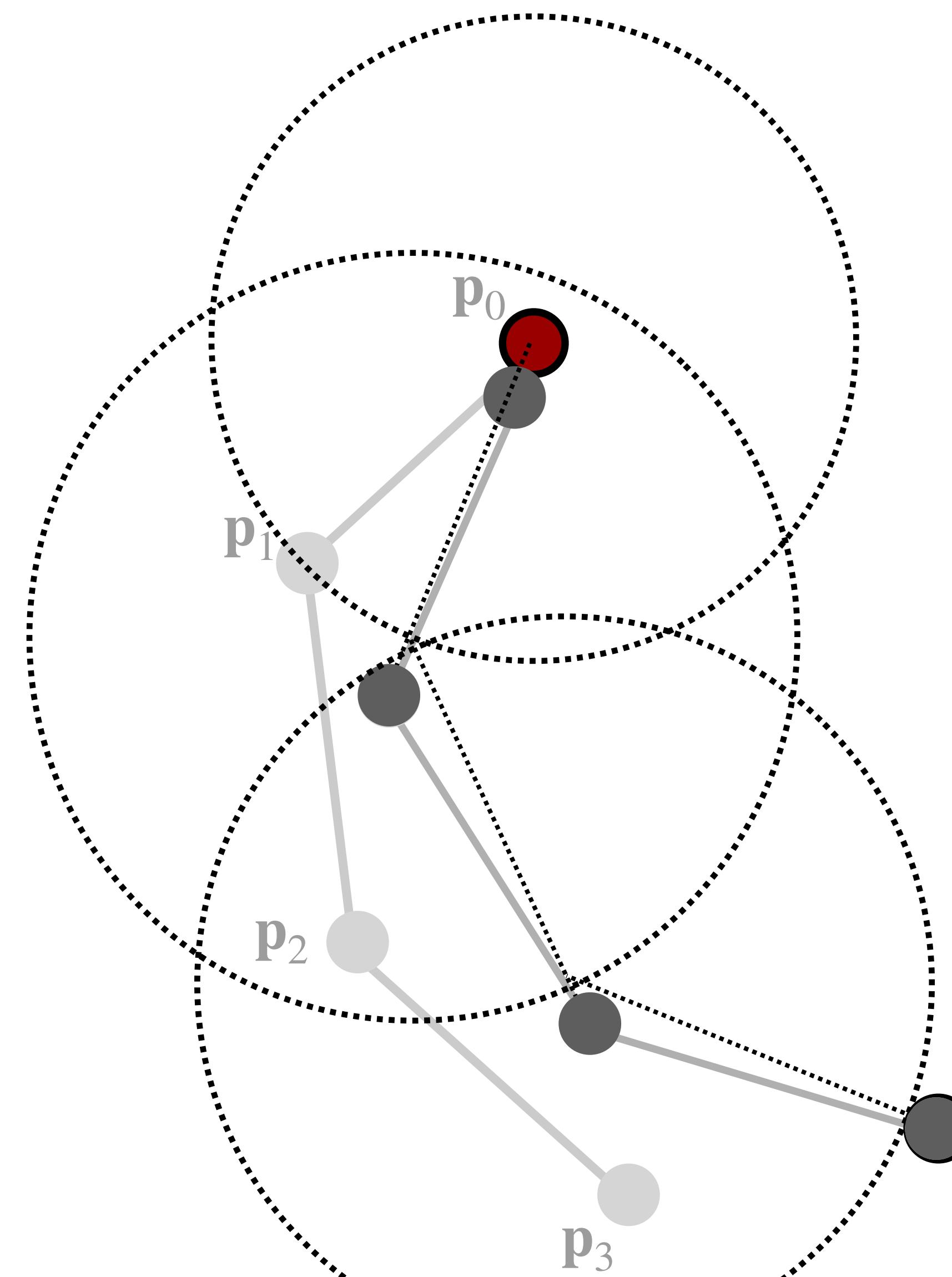
Basic algorithm



Basic algorithm



Basic algorithm



FABRIK Demos

- <http://andreasaristidou.com/FABRIK.html>

Additional reading

- **Style-based Inverse Kinematics**
 - <https://grail.cs.washington.edu/projects/styleik/>
- **Inverse Kinematics: a review of existing techniques and introduction of a new fast iterative solver**
 - <http://www.andreasaristidou.com/publications/papers/CUEDF-INFENG,%20TR-632.pdf>
- **Linesearch in gradient descent**
 - [https://optimization.mccormick.northwestern.edu/index.php/Line search methods](https://optimization.mccormick.northwestern.edu/index.php/Line_search_methods)
- **FABRIK: <http://andreasaristidou.com/publications/papers/FABRIK.pdf>**

Created by
@Freyaholmér
poor ltl unreal :-)



UNREAL
ENGINE

