Lecture 15:

Inverse Kinematics

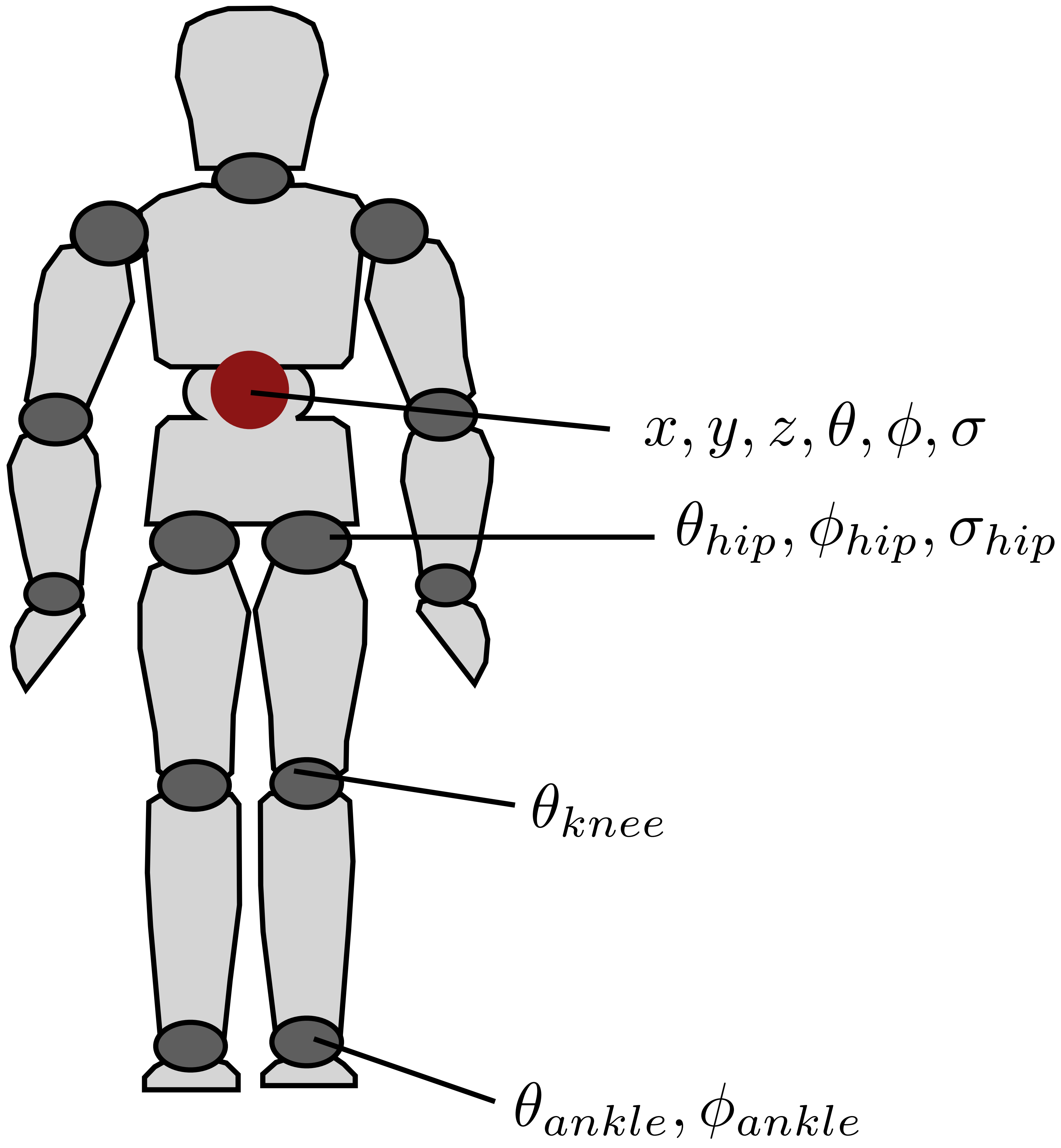
FUNDAMENTALS OF COMPUTER GRAPHICS

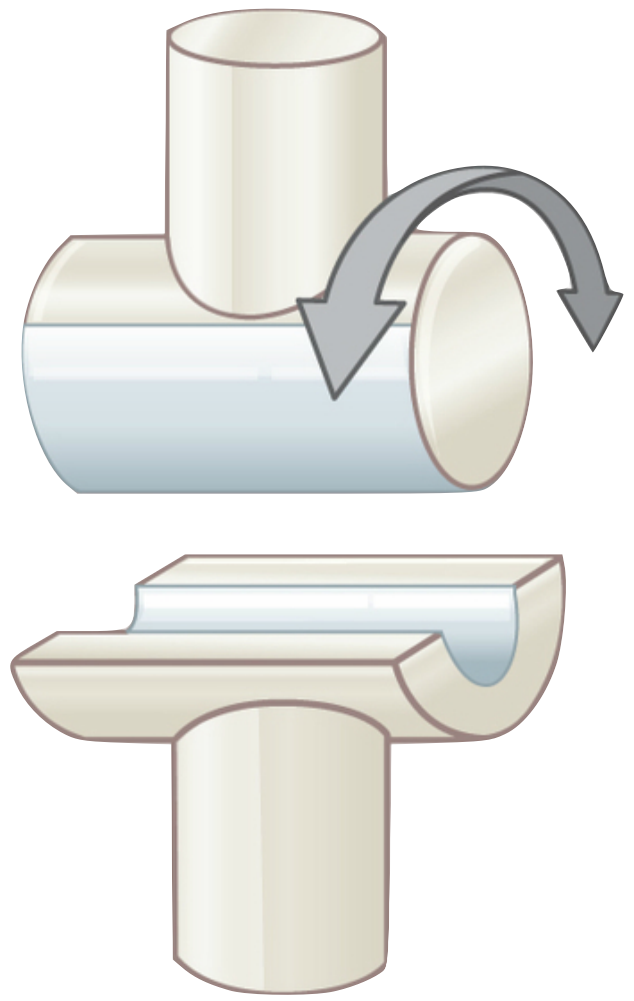
Animation & Simulation

Stanford CS248B, Fall 2022

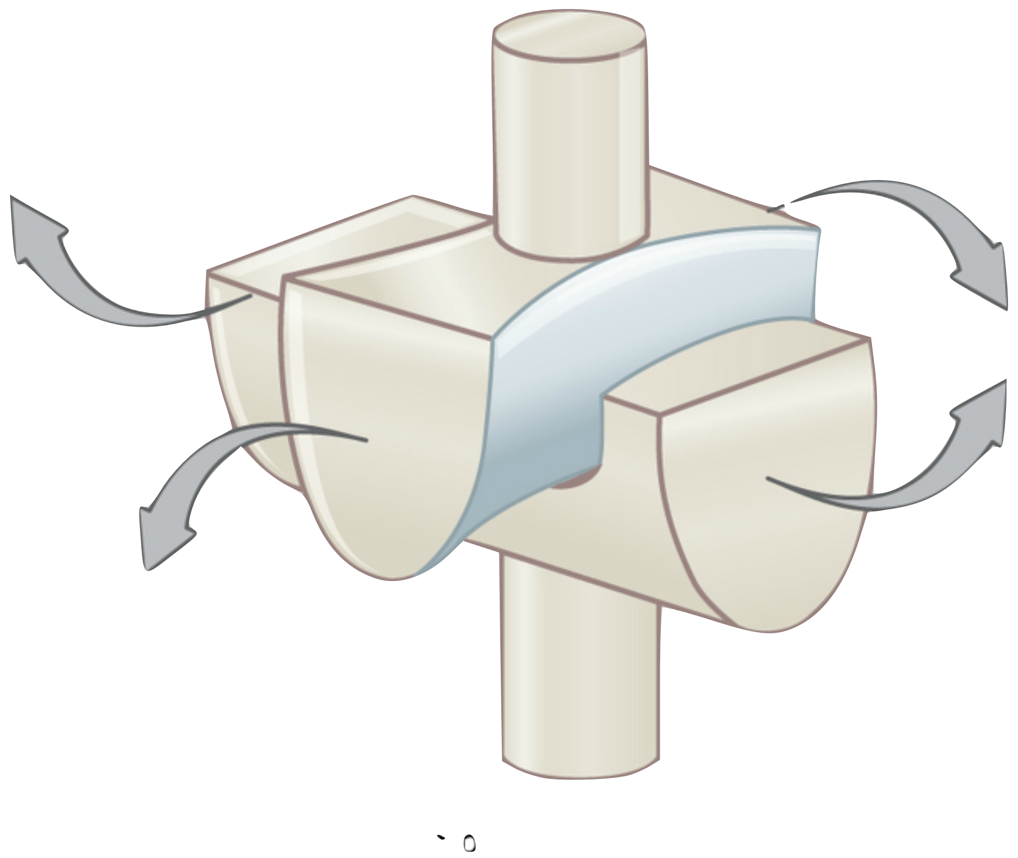
Define a configuration (pose)

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

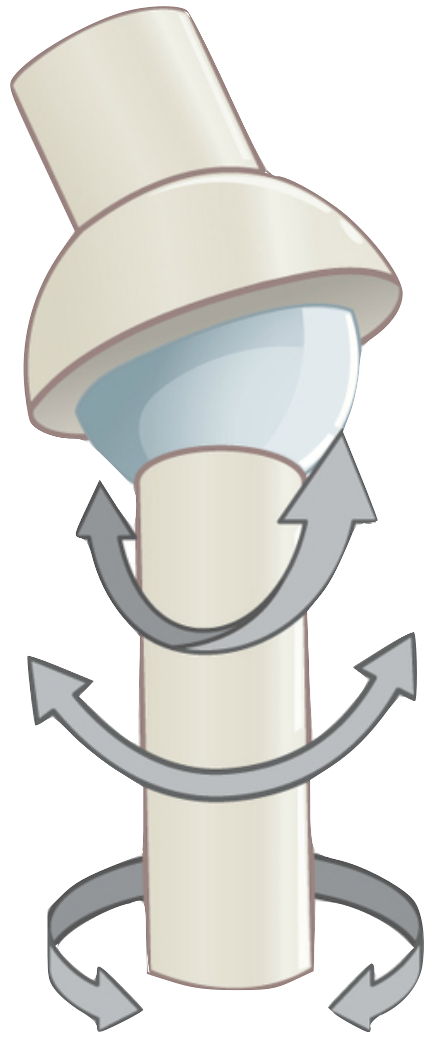




1-DOF joint



2-DOF joint



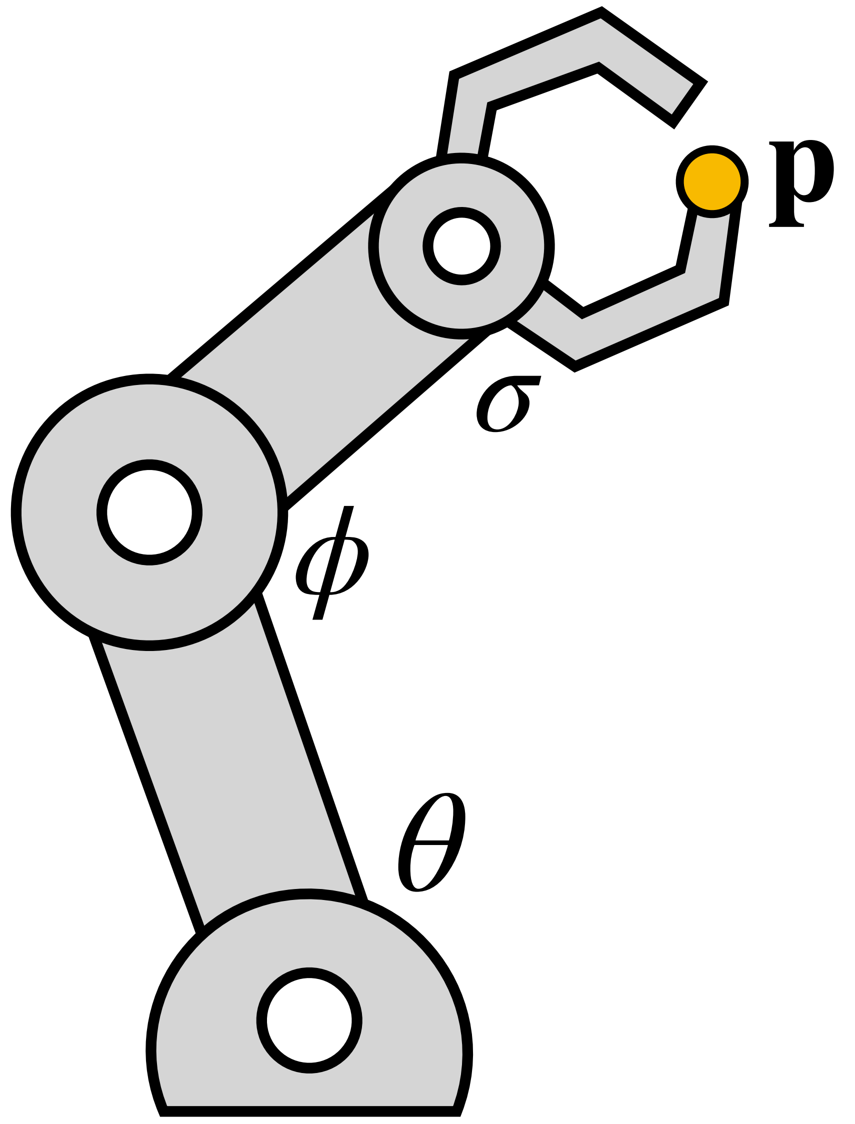
3-DOF joint

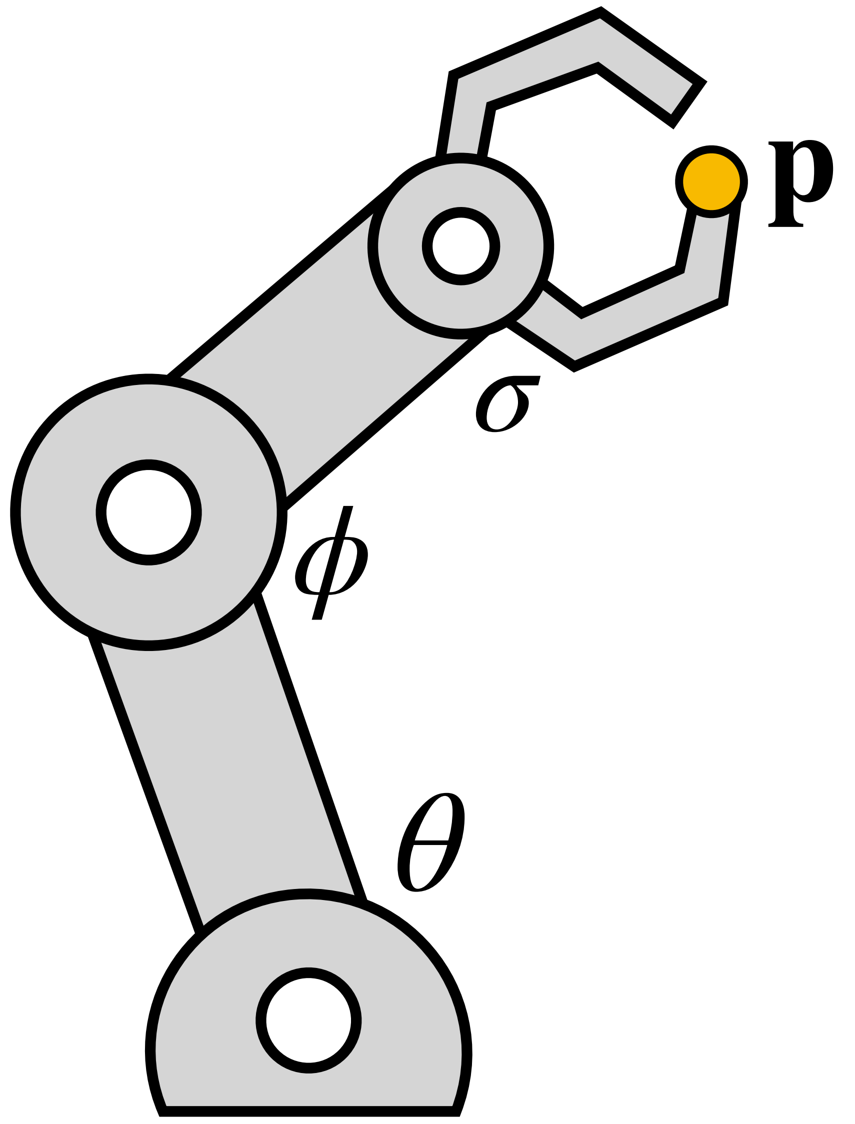
Kinematics

- **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?





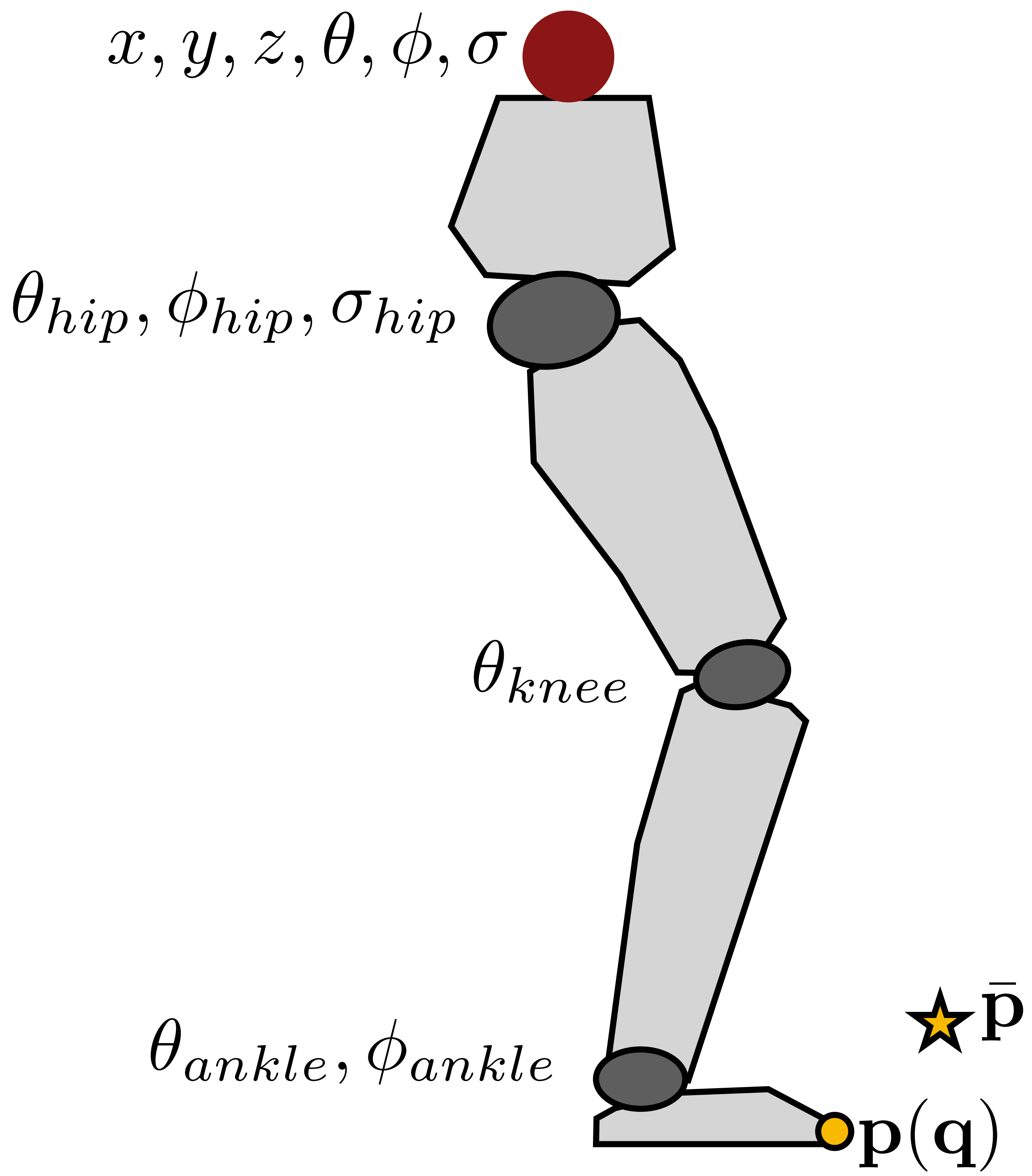
$$[\theta, \phi, \sigma] = f(\mathbf{p})$$

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

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

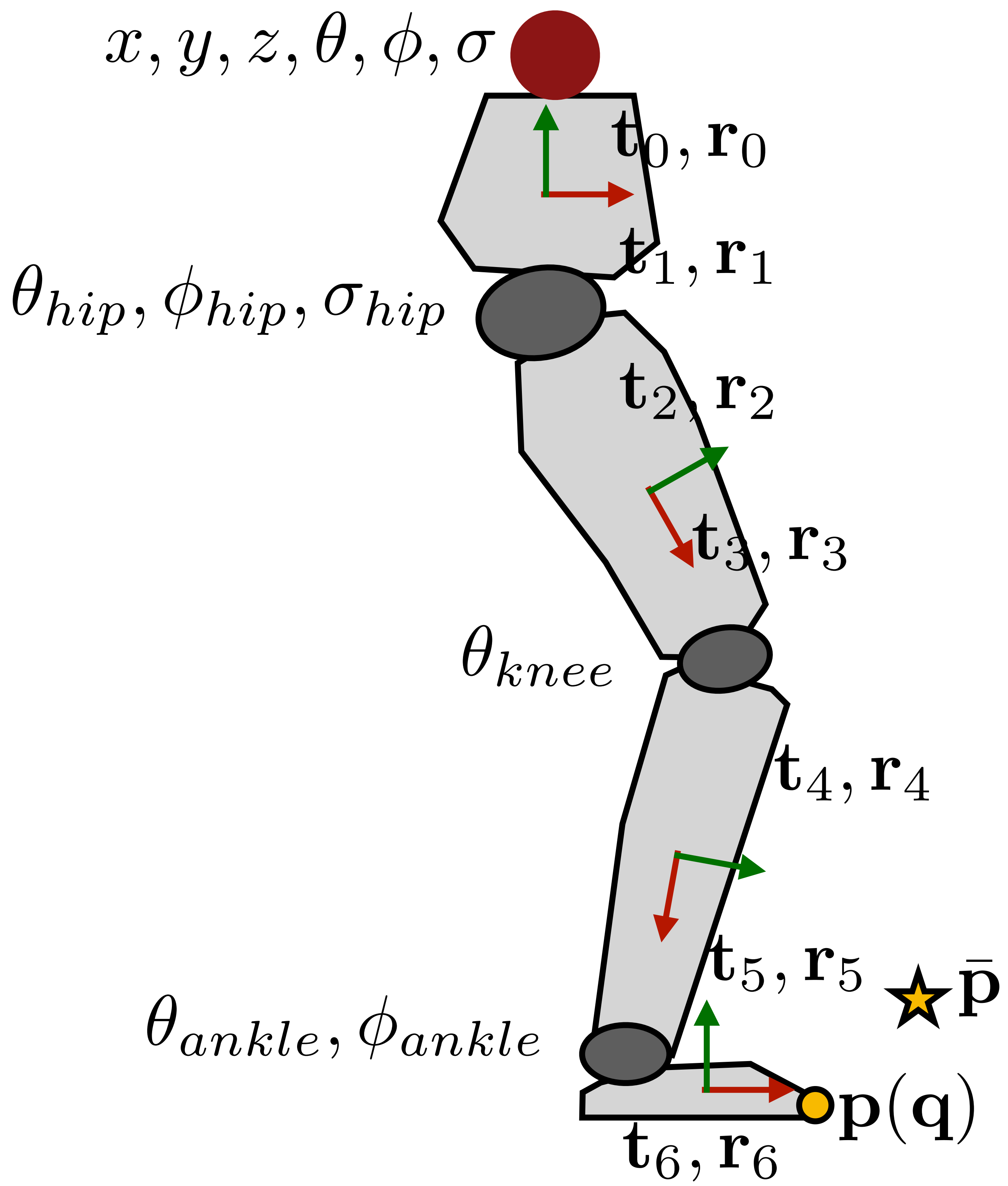


What is the pose such that the toe reaches point $\bar{\mathbf{p}}$?

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

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

IK constraints



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

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

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

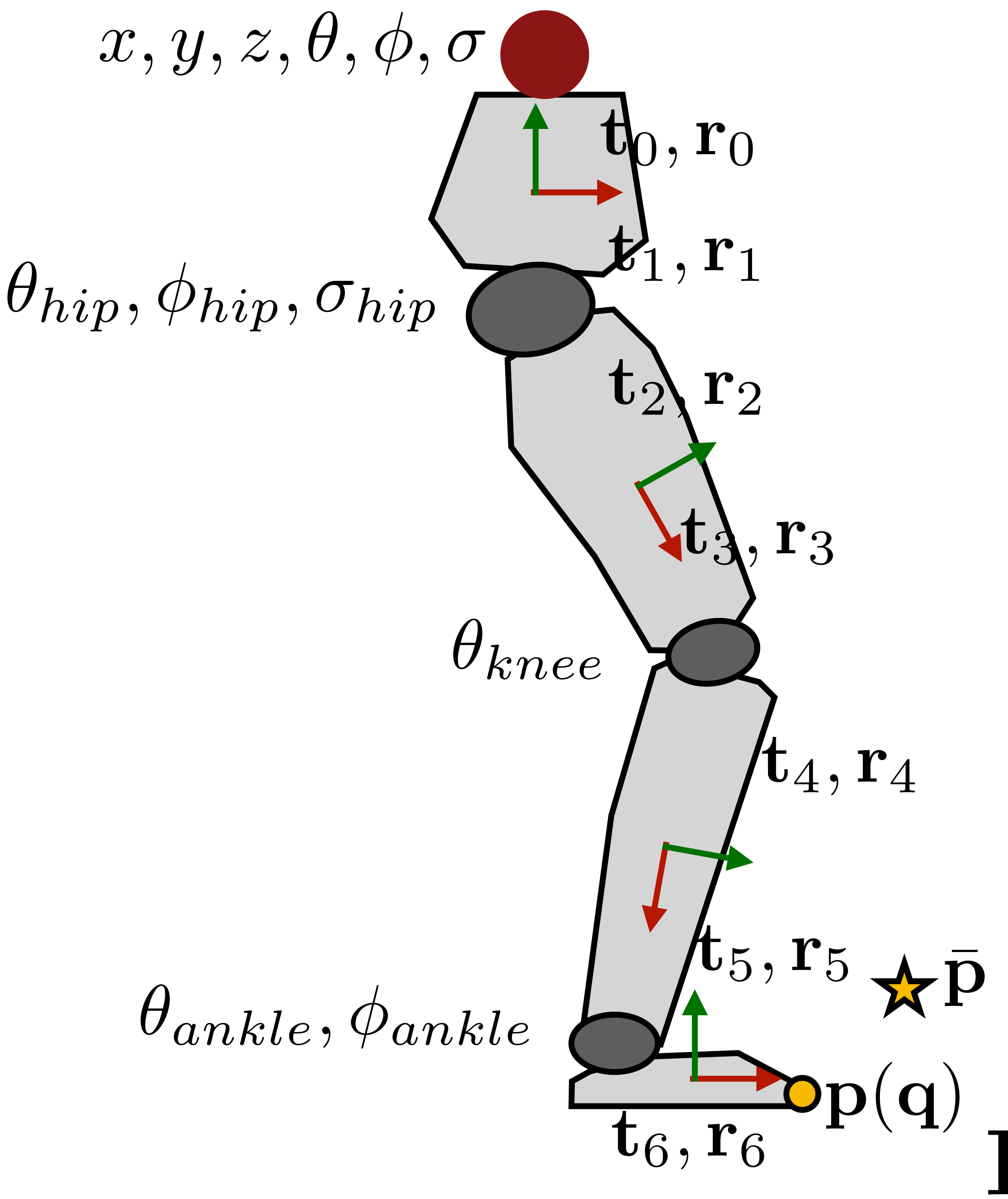
the toe in local frame

$$\mathbf{p}(\mathbf{q}) = \mathbf{T}(x, y, z)\mathbf{R}(\theta, \phi, \sigma)\mathbf{F}(\mathbf{t}_0, \mathbf{r}_0)\mathbf{F}(\mathbf{t}_1, \mathbf{r}_1)\,\mathbf{R}(\theta_{hip}, \phi_{hip}, \sigma_{hip})\,\mathbf{F}(\mathbf{t}_2, \mathbf{r}_2)\mathbf{F}(\mathbf{t}_3, \mathbf{r}_3)$$

$$\mathbf{R}(\theta_{knee})\mathbf{F}(\mathbf{t}_4, \mathbf{r}_4)\,\mathbf{F}(\mathbf{t}_5, \mathbf{r}_5)\mathbf{R}(\theta_{ankle}, \phi_{ankel})\,\mathbf{F}(\mathbf{t}_6, \mathbf{r}_6)\mathbf{p}_0$$

the toe in world frame

IK constraints



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

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

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

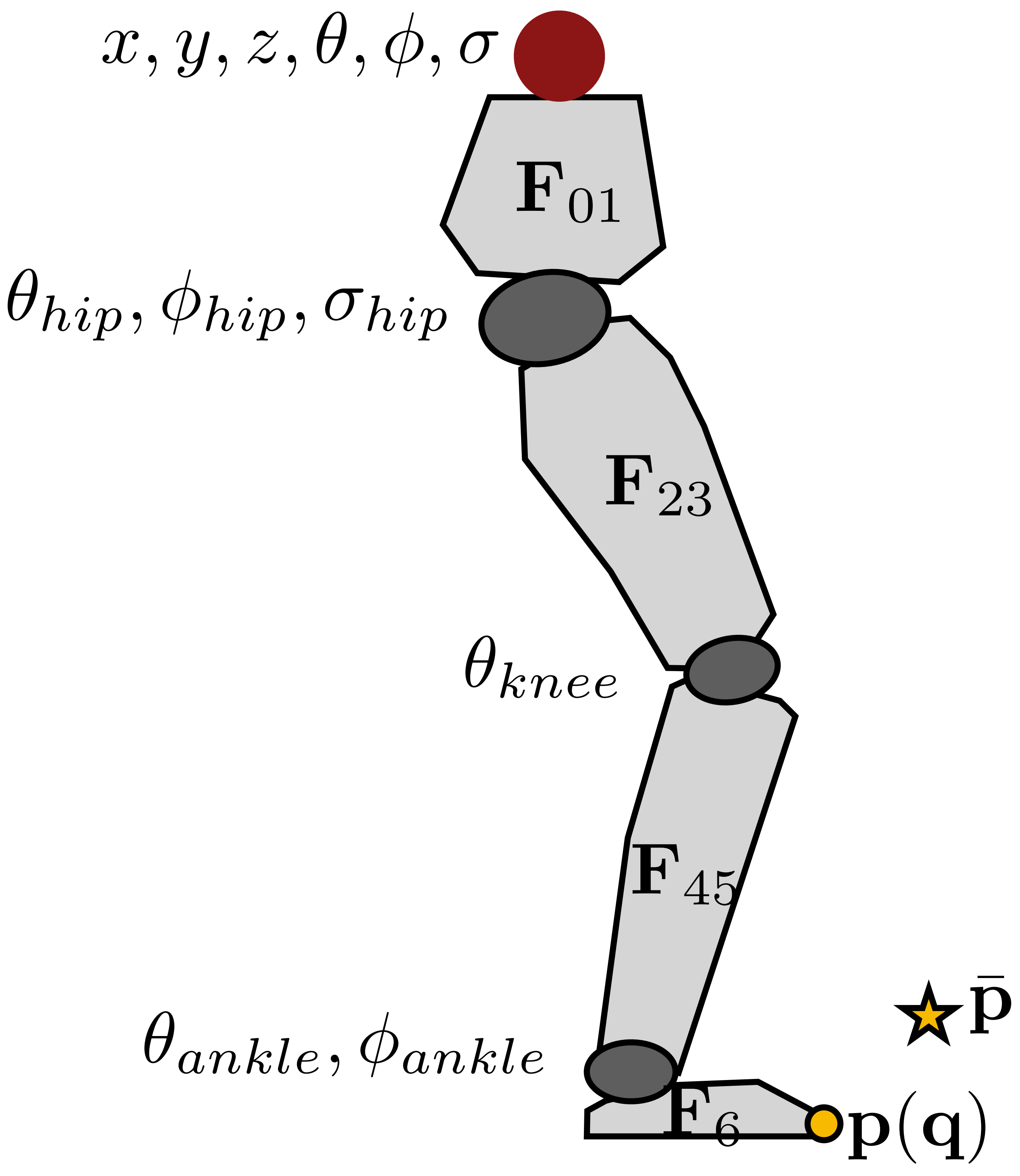
the toe in local frame

$$\mathbf{p}(\mathbf{q}) = \mathbf{T}(x, y, z)\mathbf{R}(\theta, \phi, \sigma)\overbrace{\cancel{\mathbf{F}(\mathbf{t}_0, \mathbf{r}_0)\mathbf{F}(\mathbf{t}_1, \mathbf{r}_1)}}^{\mathbf{F}_{12}}\mathbf{R}(\theta_{hip}, \phi_{hip}, \sigma_{hip})\overbrace{\cancel{\mathbf{F}(\mathbf{t}_2, \mathbf{r}_2)\mathbf{F}(\mathbf{t}_3, \mathbf{r}_3)}}^{\mathbf{F}_{23}}$$

$$\mathbf{R}(\theta_{knee})\underbrace{\cancel{\mathbf{F}(\mathbf{t}_4, \mathbf{r}_4)\mathbf{F}(\mathbf{t}_5, \mathbf{r}_5)}}_{\mathbf{F}_{45}}\mathbf{R}(\theta_{ankle}, \phi_{ankel})\underbrace{\cancel{\mathbf{F}(\mathbf{t}_6, \mathbf{r}_6)}}_{\mathbf{F}_6}\mathbf{p}_0$$

the toe in world frame

IK constraints



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

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

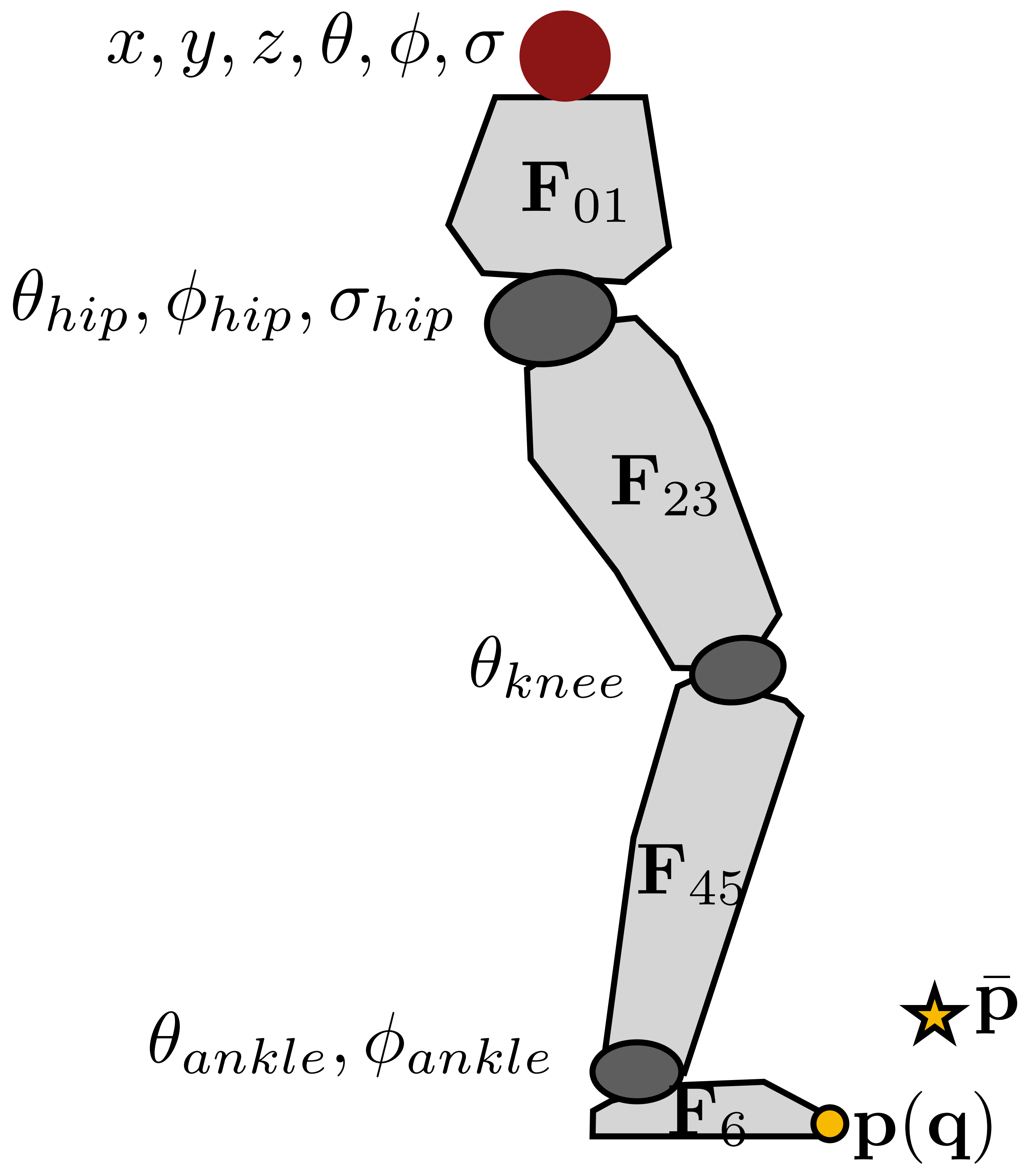
What is the coordinates of the toe in the world frame?

the toe in local frame

$$\mathbf{p}(\mathbf{q}) = \mathbf{T}(x, y, z)\mathbf{R}(\theta, \phi, \sigma)\mathbf{F}_{01}\mathbf{R}(\theta_{hip}, \phi_{hip}, \sigma_{hip})\mathbf{F}_{23}\mathbf{R}(\theta_{knee})\mathbf{F}_{45}\mathbf{R}(\theta_{ankle}, \phi_{ankel})\mathbf{F}_6\mathbf{p}_0$$

the toe in world frame

IK constraints



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

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

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

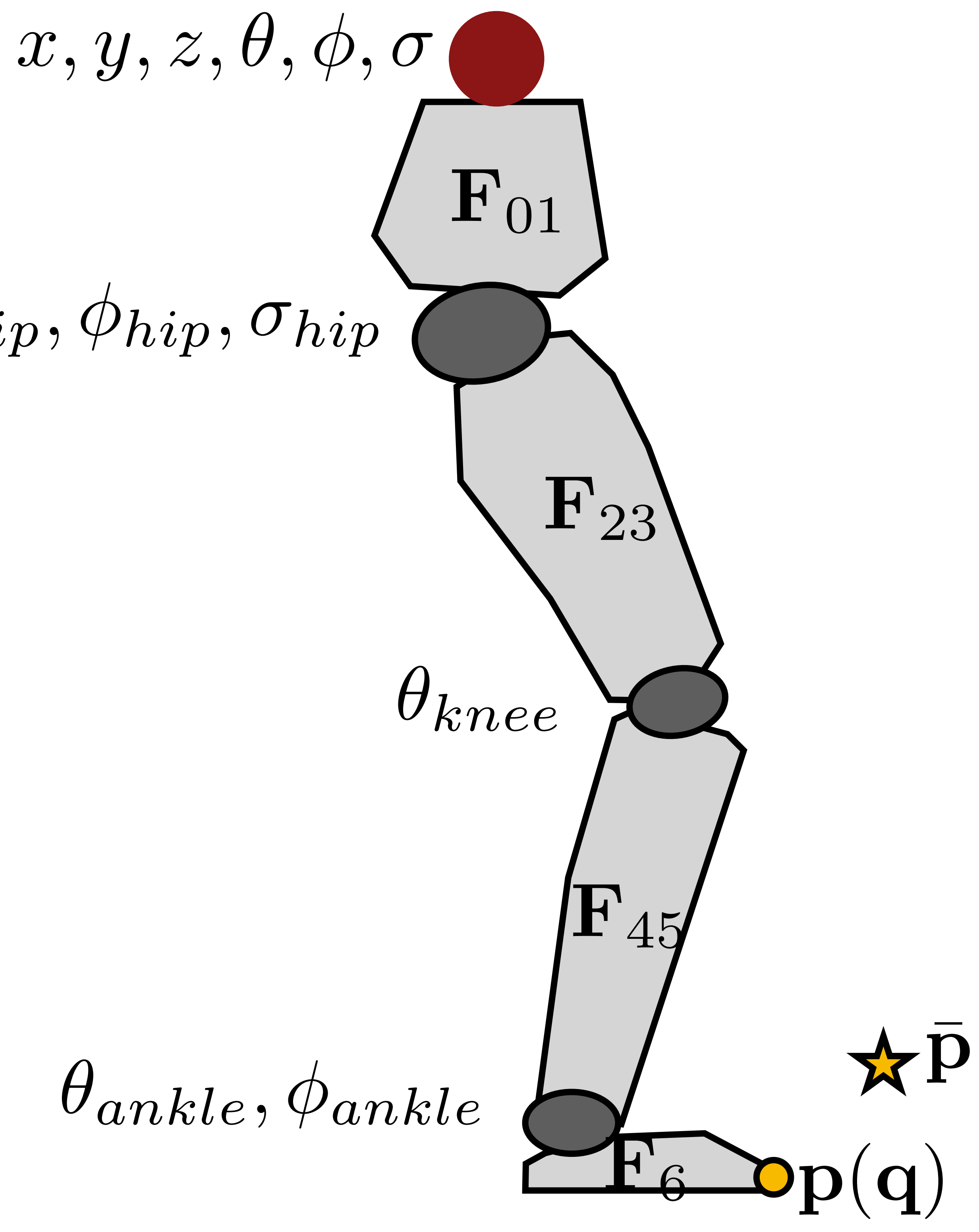
What is the coordinates of the toe in the world frame?

the toe in local frame

$$\mathbf{p}(\mathbf{q}) = \mathbf{T}(x, y, z)\mathbf{R}(\theta, \phi, \sigma)\mathbf{F}_{01}\mathbf{R}(\theta_{hip}, \phi_{hip}, \sigma_{hip})\mathbf{F}_{23}\mathbf{R}(\theta_{knee})\mathbf{F}_{45}\mathbf{R}(\theta_{ankle}, \phi_{ankel})\mathbf{F}_6\mathbf{p}_0$$

the toe in world frame

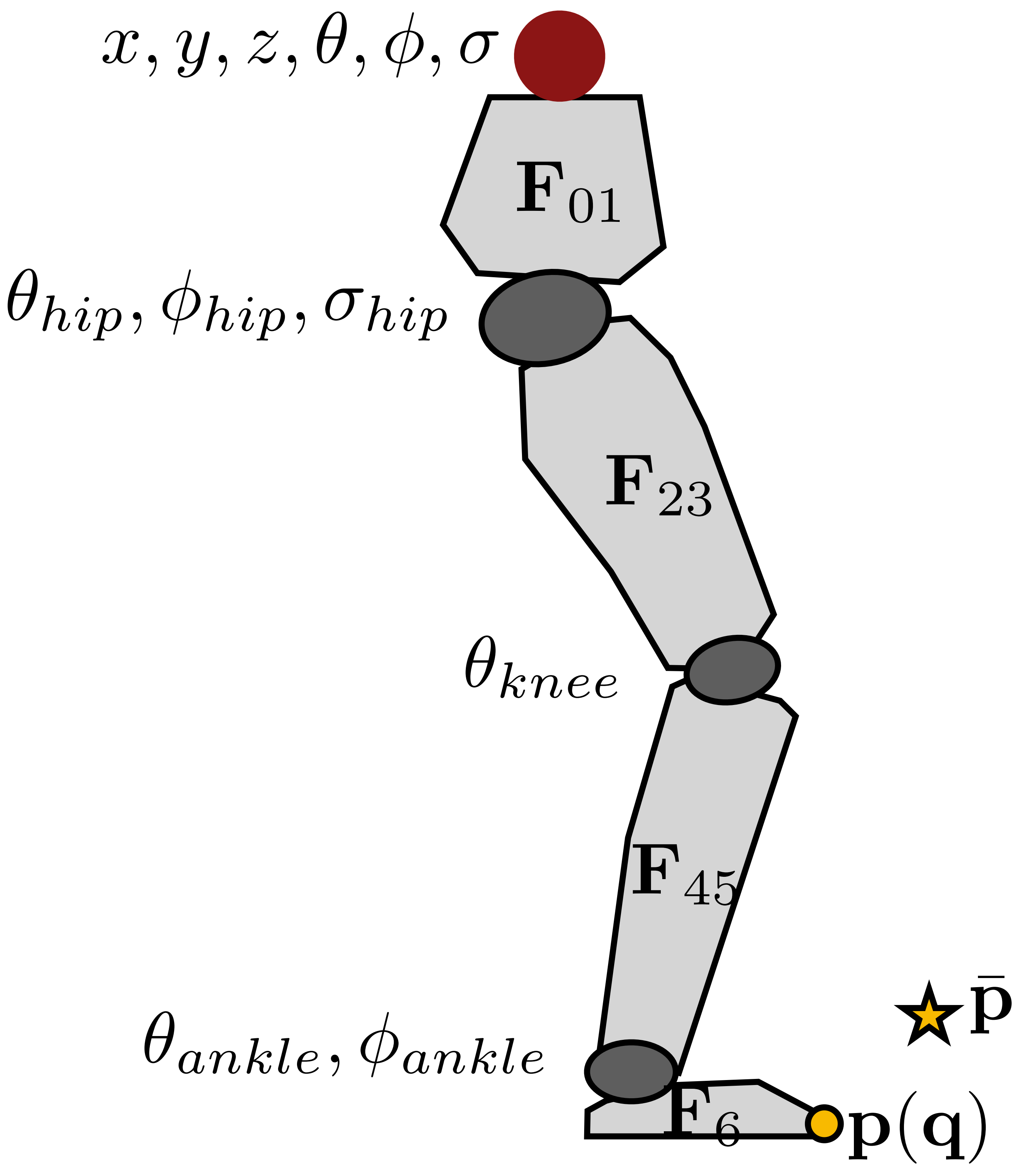
Solutions



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

- **Three possible outcomes to $\mathbf{C}(\mathbf{q}) = \mathbf{0}$:**
  - **Single solution**
  - **No solution**
  - **Multiple solutions**

Solutions

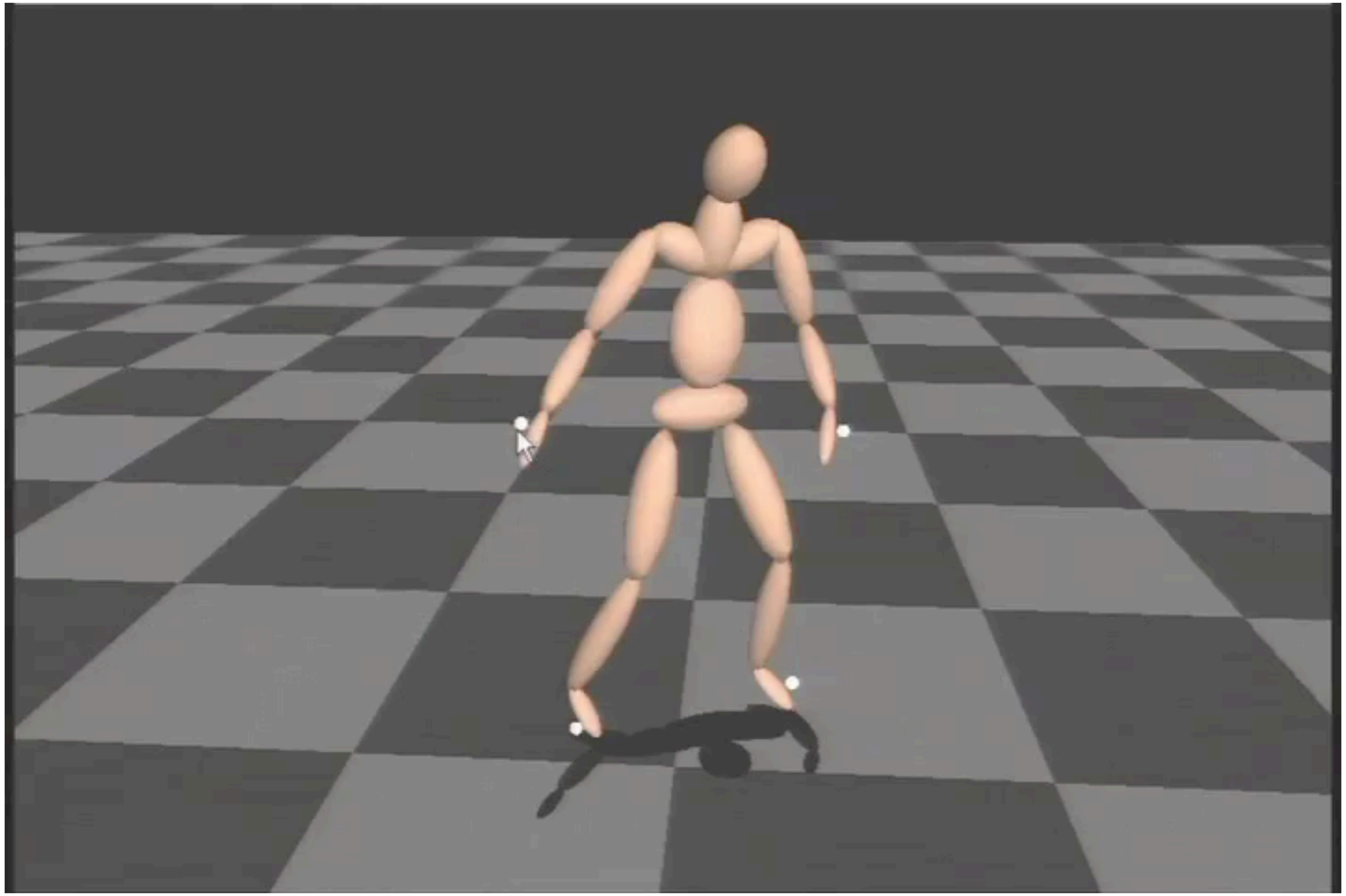


$$\mathbf{C}(\mathbf{q}) = \mathbf{p}(\mathbf{q}) - \bar{\mathbf{p}} = \mathbf{0}$$

- Three possible outcomes to $\mathbf{C}(\mathbf{q}) = \mathbf{0}$:
  - Single solution
  - No solution
  - Multiple solutions

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

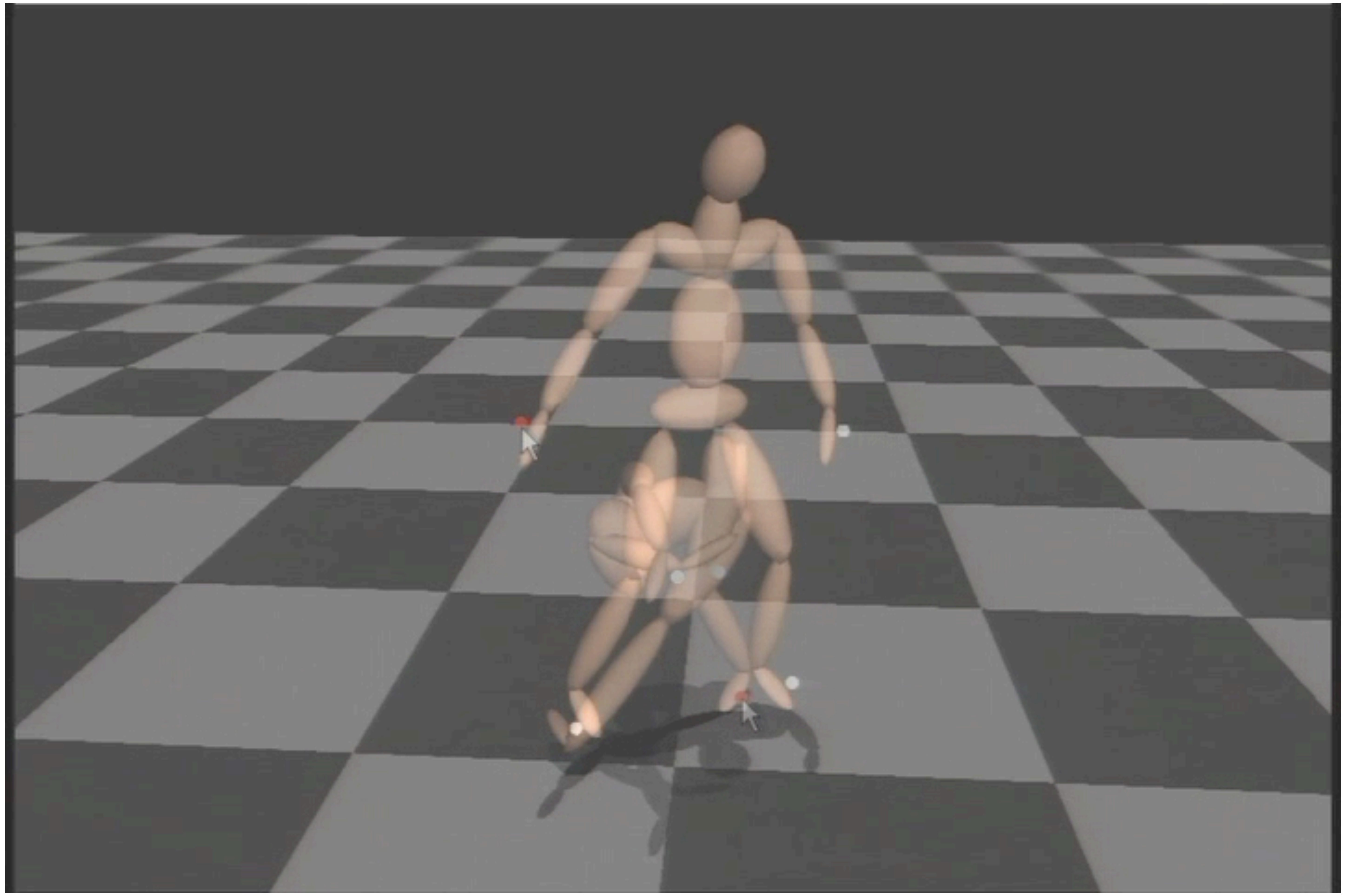
Underconstrained IK



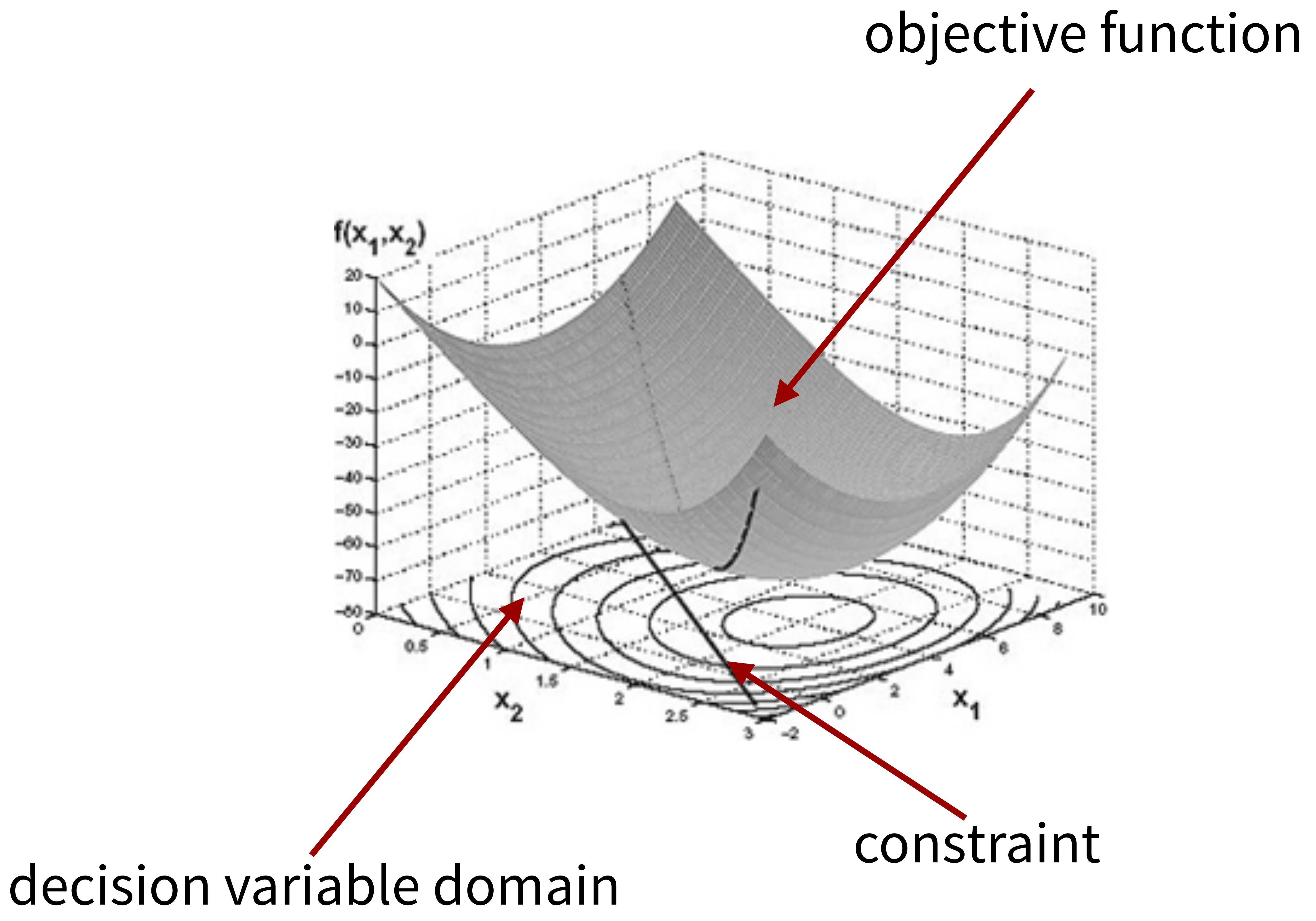
How to find optimal solution?

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

Humanlike IK

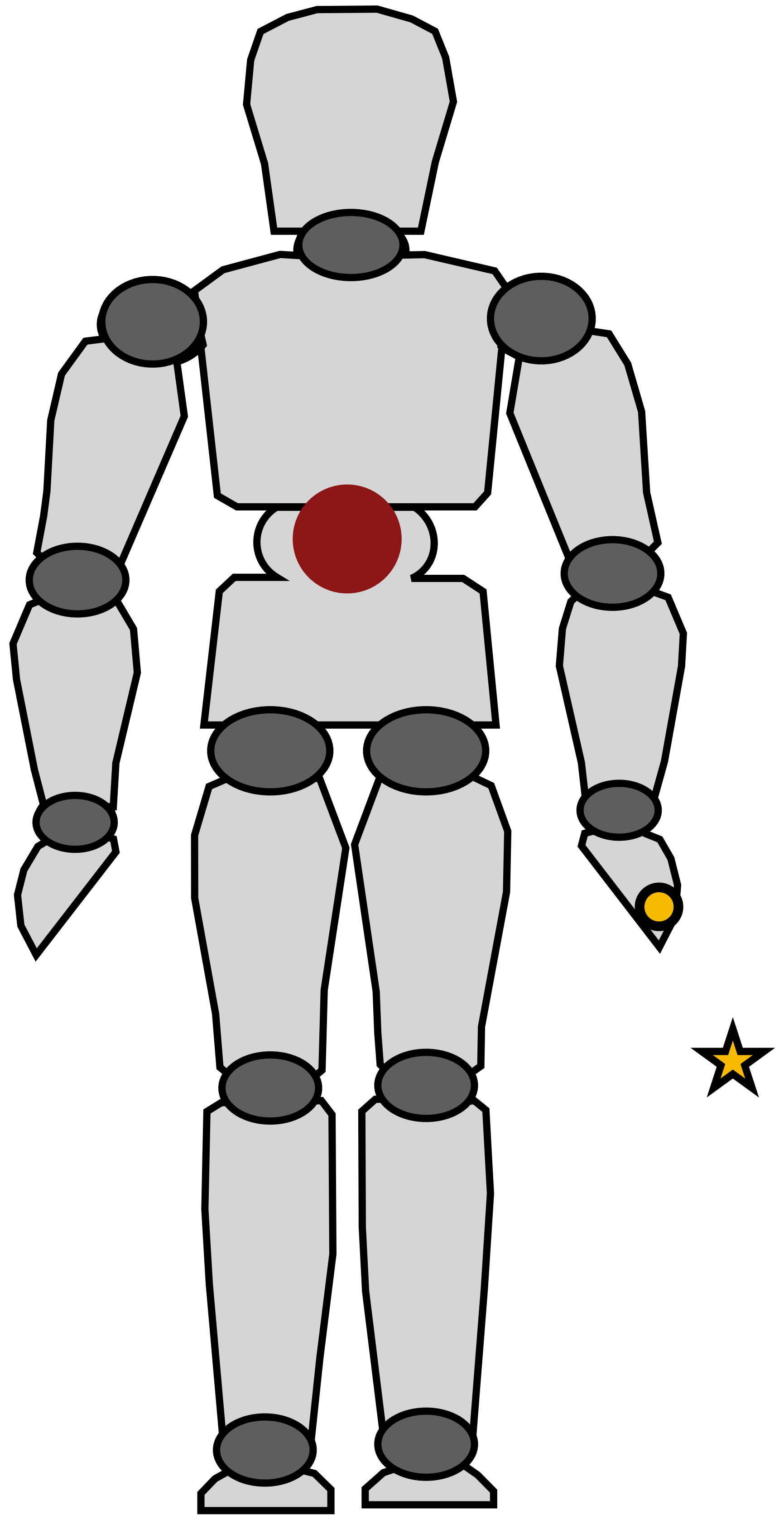


Formulate an optimization problem



Formulate an optimization problem

Solve for joint configuration $\mathbf{q}$

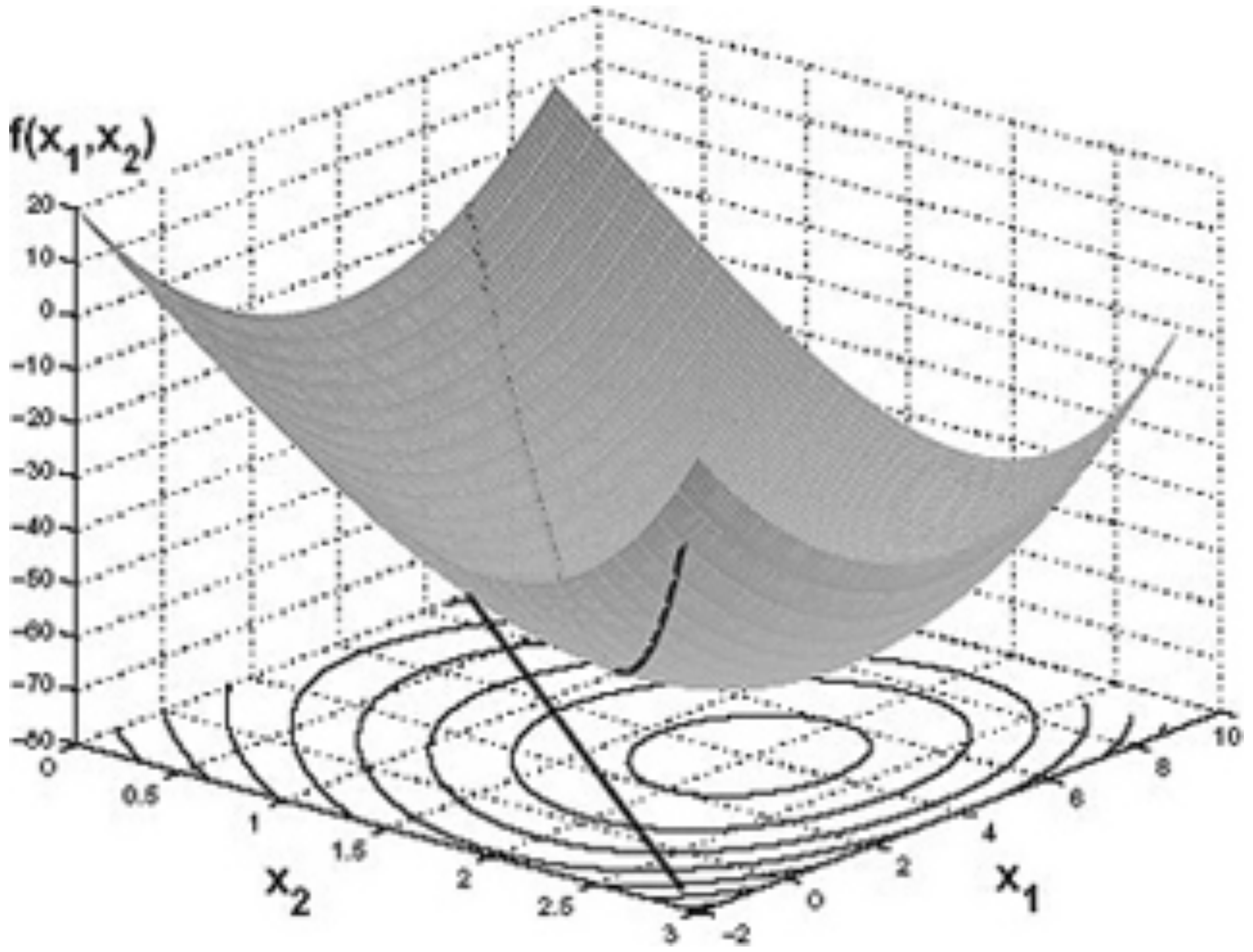


$$\text{minimize} \quad g(\mathbf{q})$$

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

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

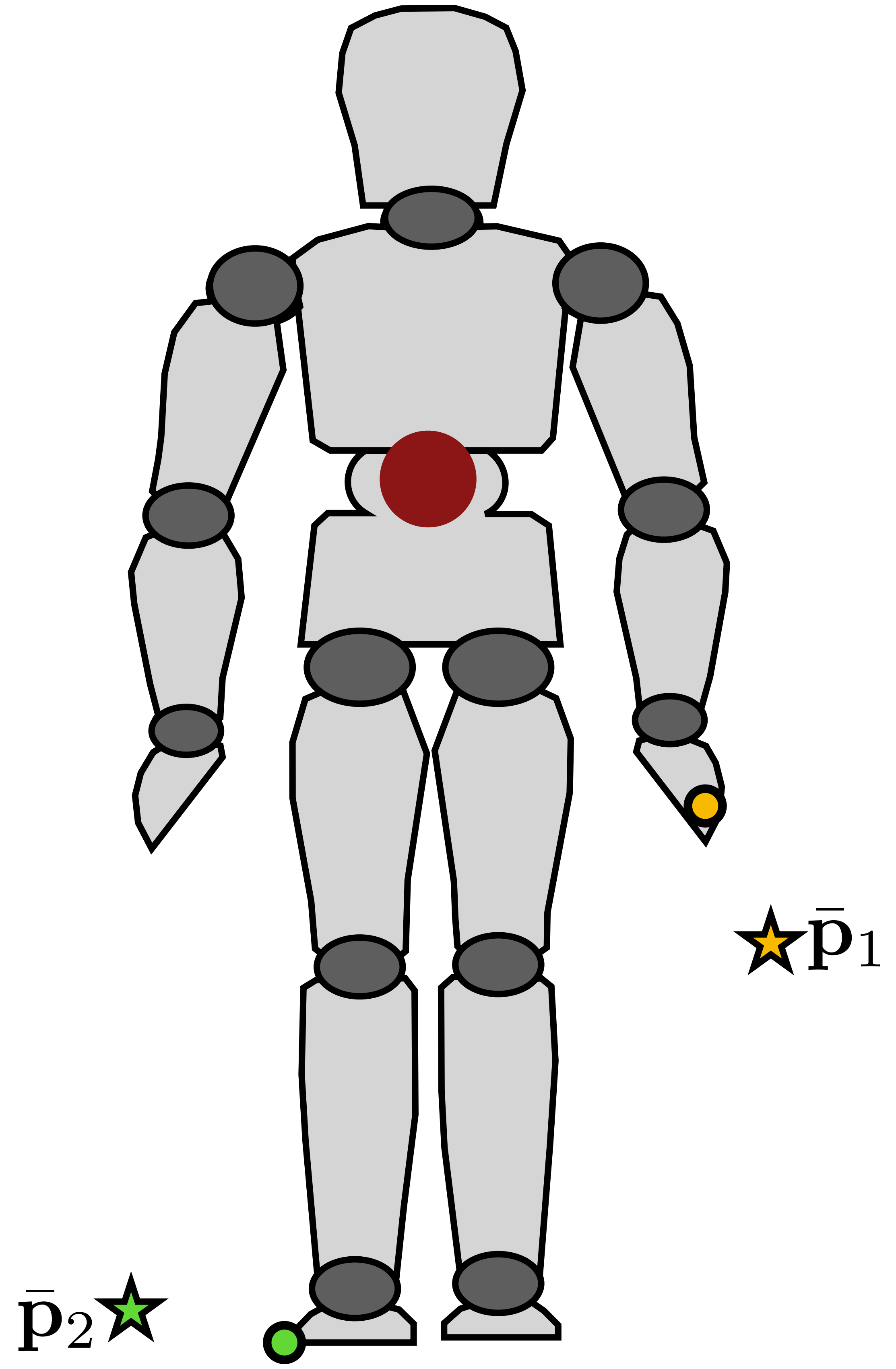
$$\mathbf{C}(\mathbf{q}) = \mathbf{p}(\mathbf{q}) - \bar{\mathbf{p}}$$



Nonconvex optimization!

Formulate an optimization problem

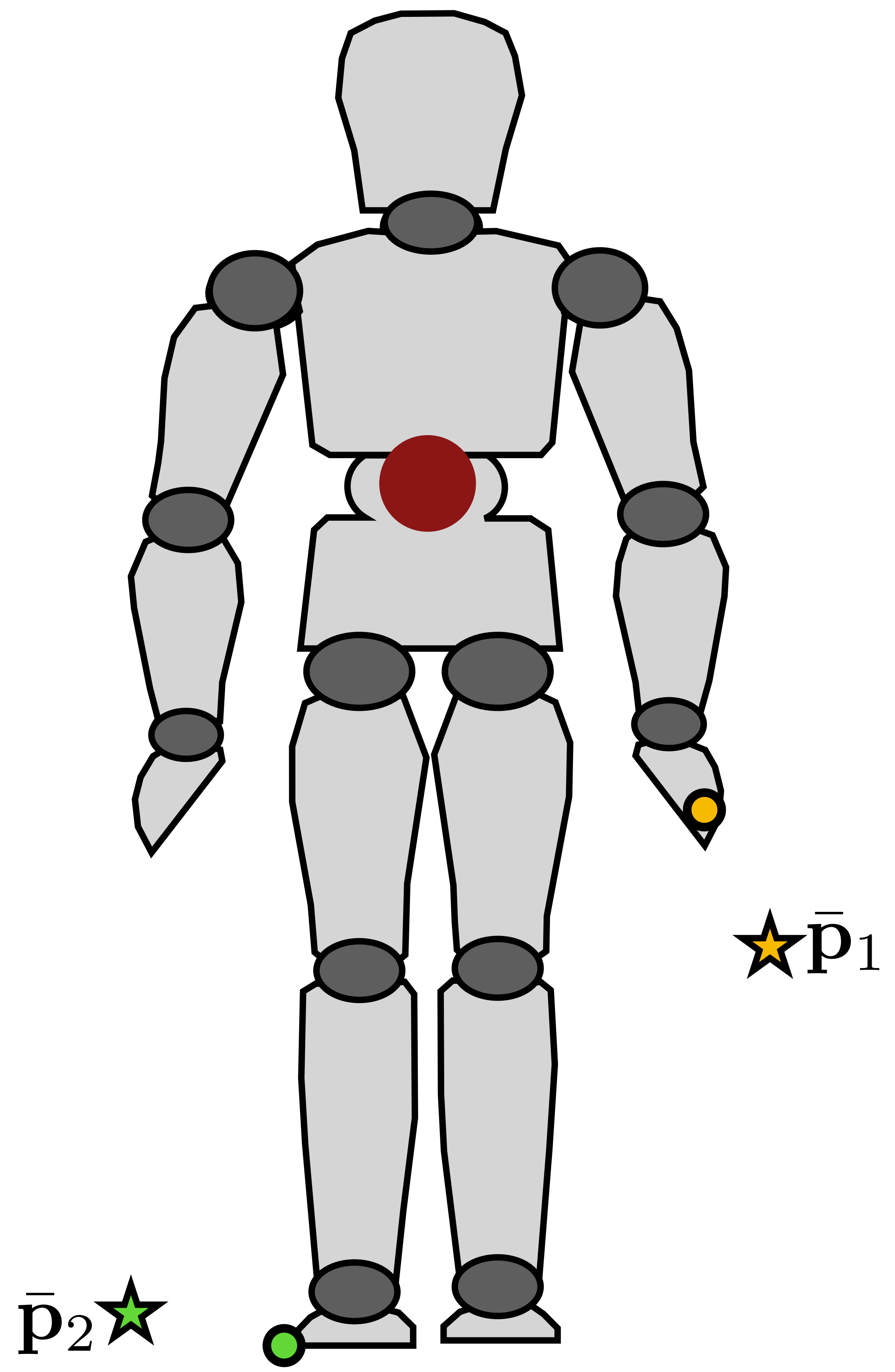
What if we have multiple constraints?



$$\begin{aligned} &\text{minimize} \quad g(\mathbf{q}) \\ &\text{subject to} \quad \mathbf{C}_1(\mathbf{q}) = \mathbf{0} \\ &\qquad\qquad\quad\;\; \mathbf{C}_2(\mathbf{q}) = \mathbf{0} \end{aligned}$$

Formulate an optimization problem

What if we have multiple constraints?

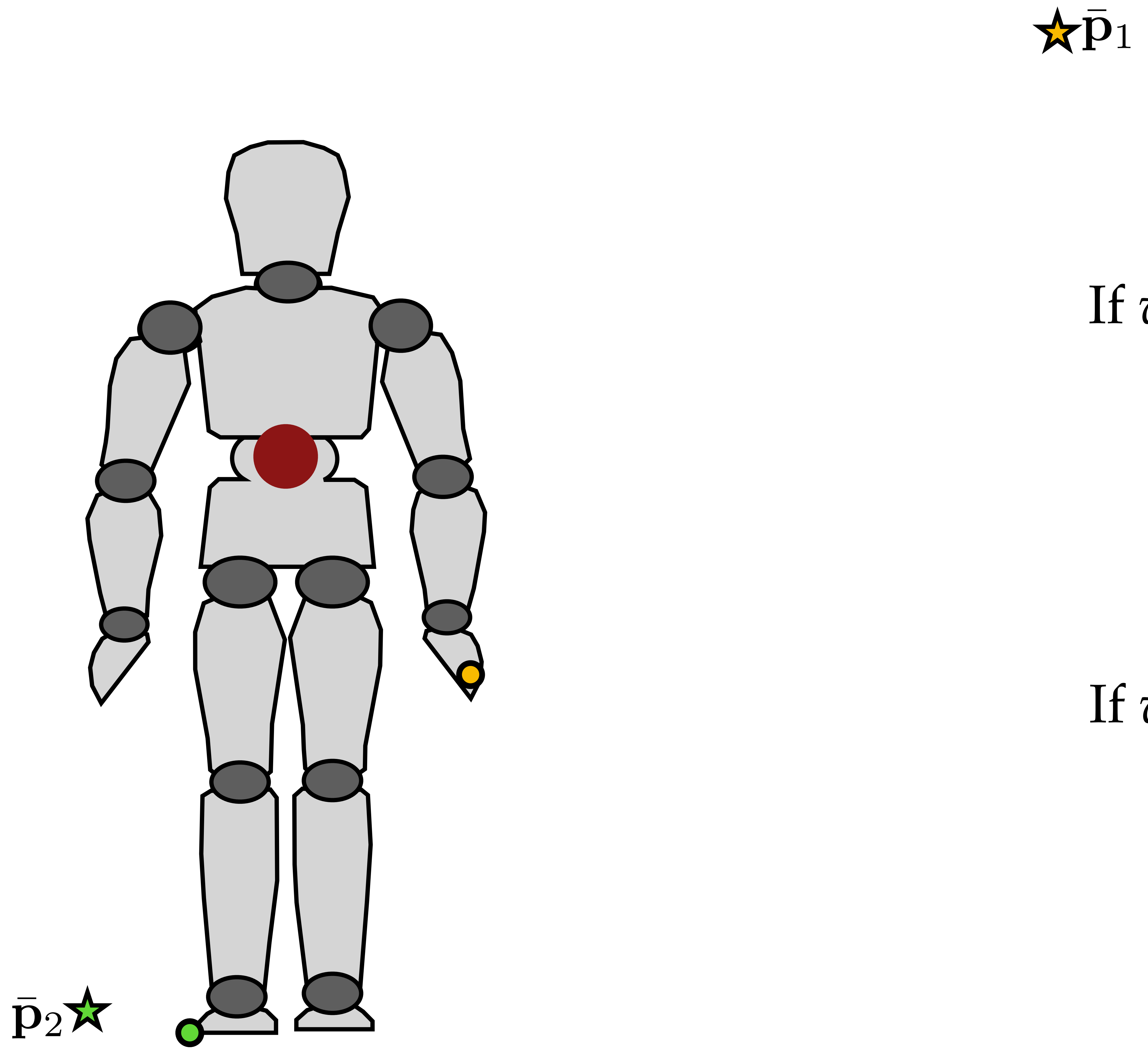


$$\text{minimize} \quad g(\mathbf{q})$$
$$\text{subject to} \quad \mathbf{C}_1(\mathbf{q}) = \mathbf{0}$$
$$\mathbf{C}_2(\mathbf{q}) = \mathbf{0}$$

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

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

Quiz



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

If $w_1 >> w_2$, 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

---

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

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

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

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

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

---

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

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

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

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

Gradient descent

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

To compute gradient in each optimization iteration,

we need to evaluate the constraint:

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

Gradient descent

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

To compute gradient in each optimization iteration,

we need to evaluate the constraint:

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

we need to compute the Jacobian matrix:

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

Gradient descent

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

To compute gradient in each optimization iteration,

we need to evaluate the constraint:

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

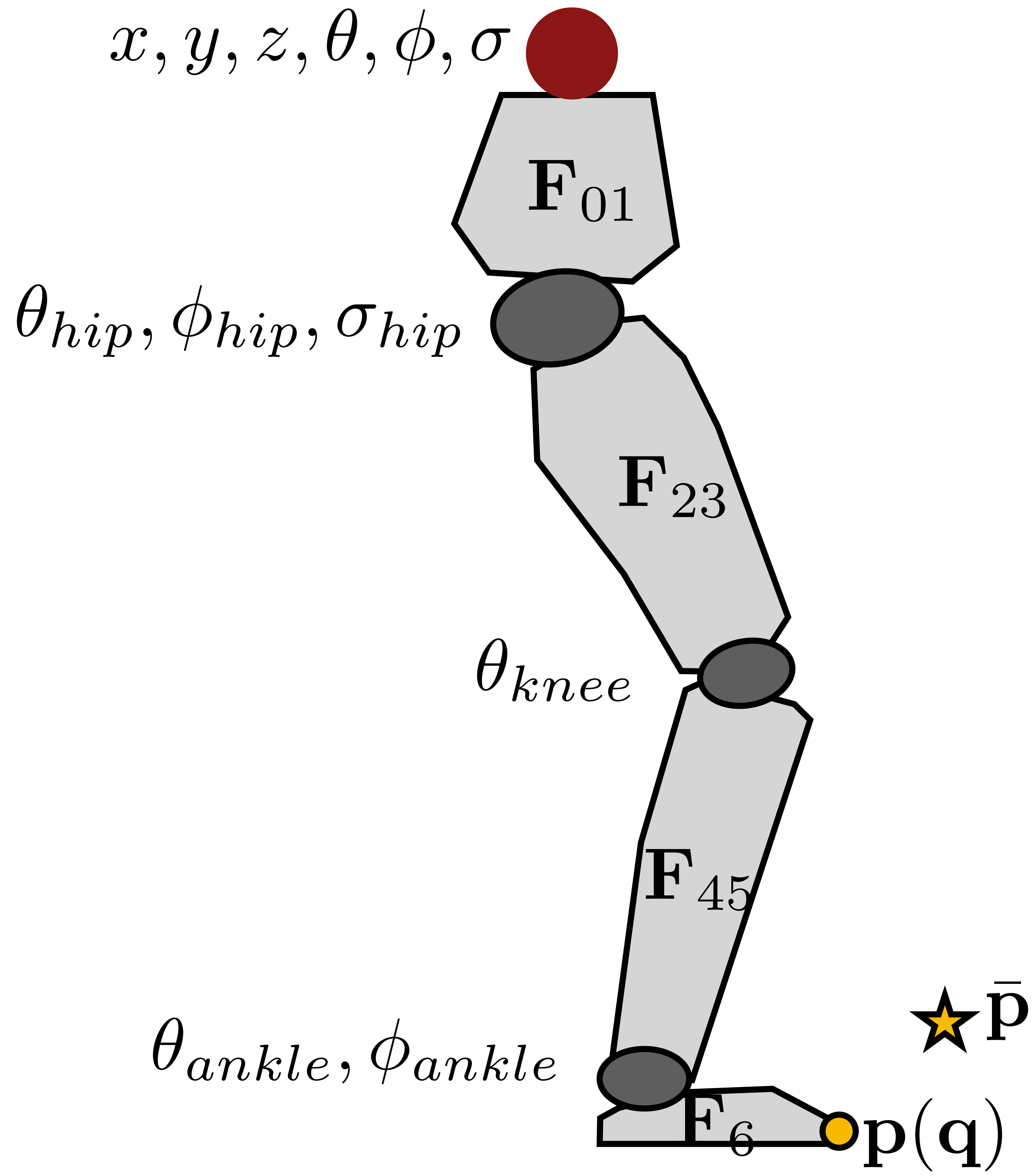
we need to compute the Jacobian matrix:

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

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



To compute gradient in each optimization iteration,

we need to evaluate the constraint:

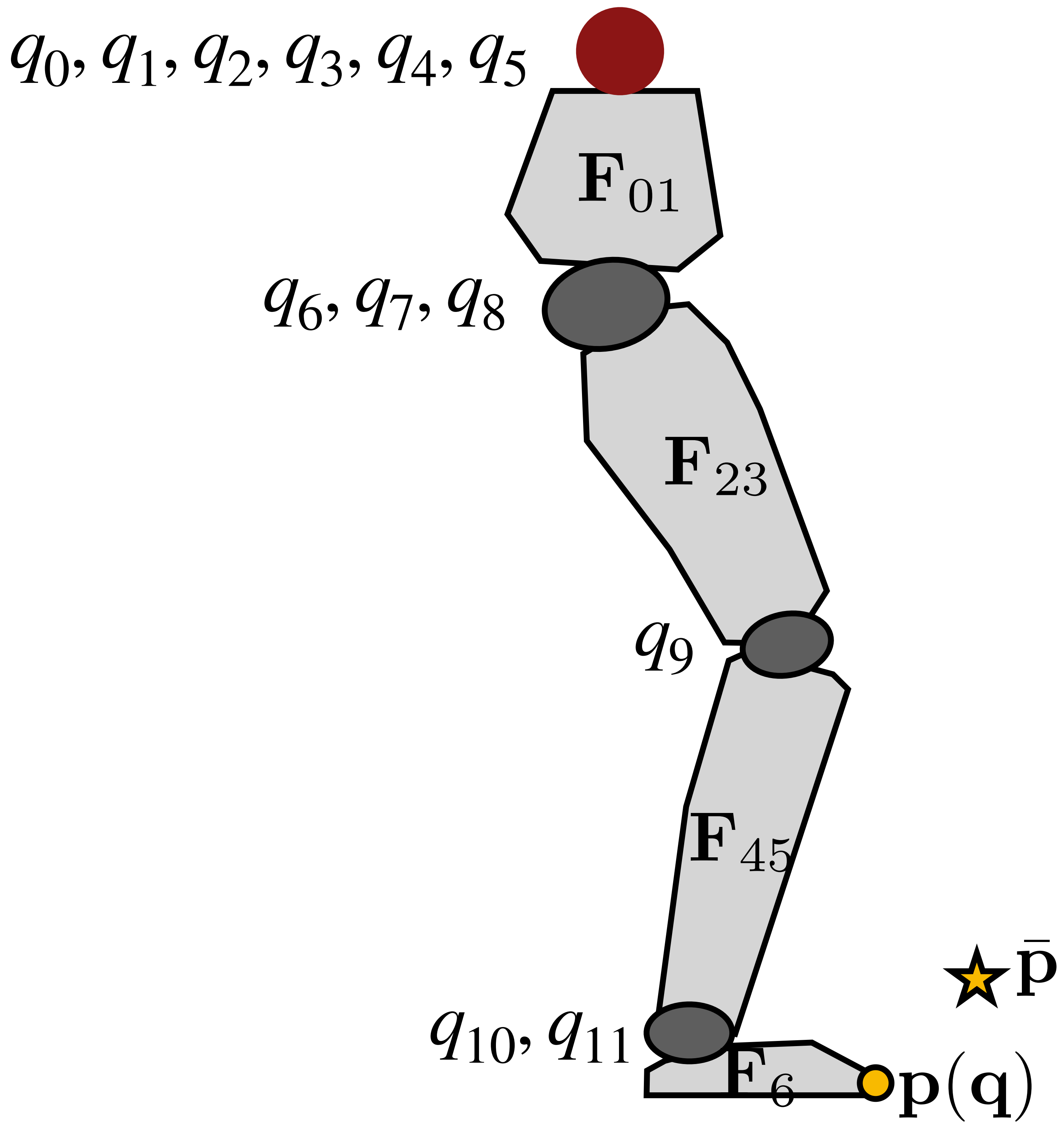
$\mathbf{C}(\mathbf{q})$

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: $\mathbf{q} \leftarrow \{q_0, q_1, q_2, q_3, q_4, q_5, q_6, q_7, q_8, q_9, q_{10}, q_{11}\}$



To compute gradient in each optimization iteration,

we need to evaluate the constraint:

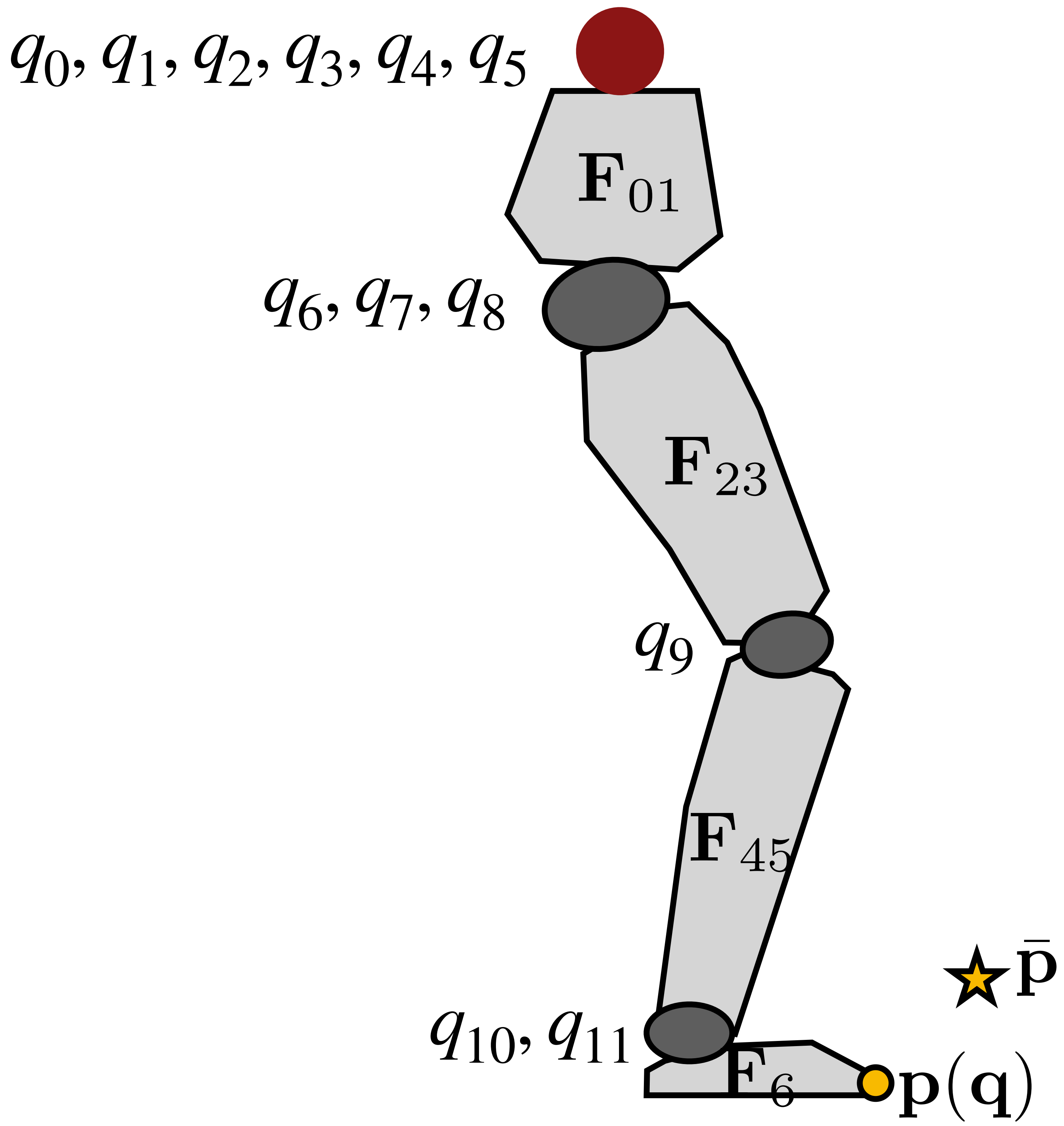
$\mathbf{C}(\mathbf{q})$

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: $\mathbf{q} \leftarrow \{q_0, q_1, q_2, q_3, q_4, q_5, q_6, q_7, q_8, q_9, q_{10}, q_{11}\}$



To compute gradient in each optimization iteration,

we need to evaluate the constraint:

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

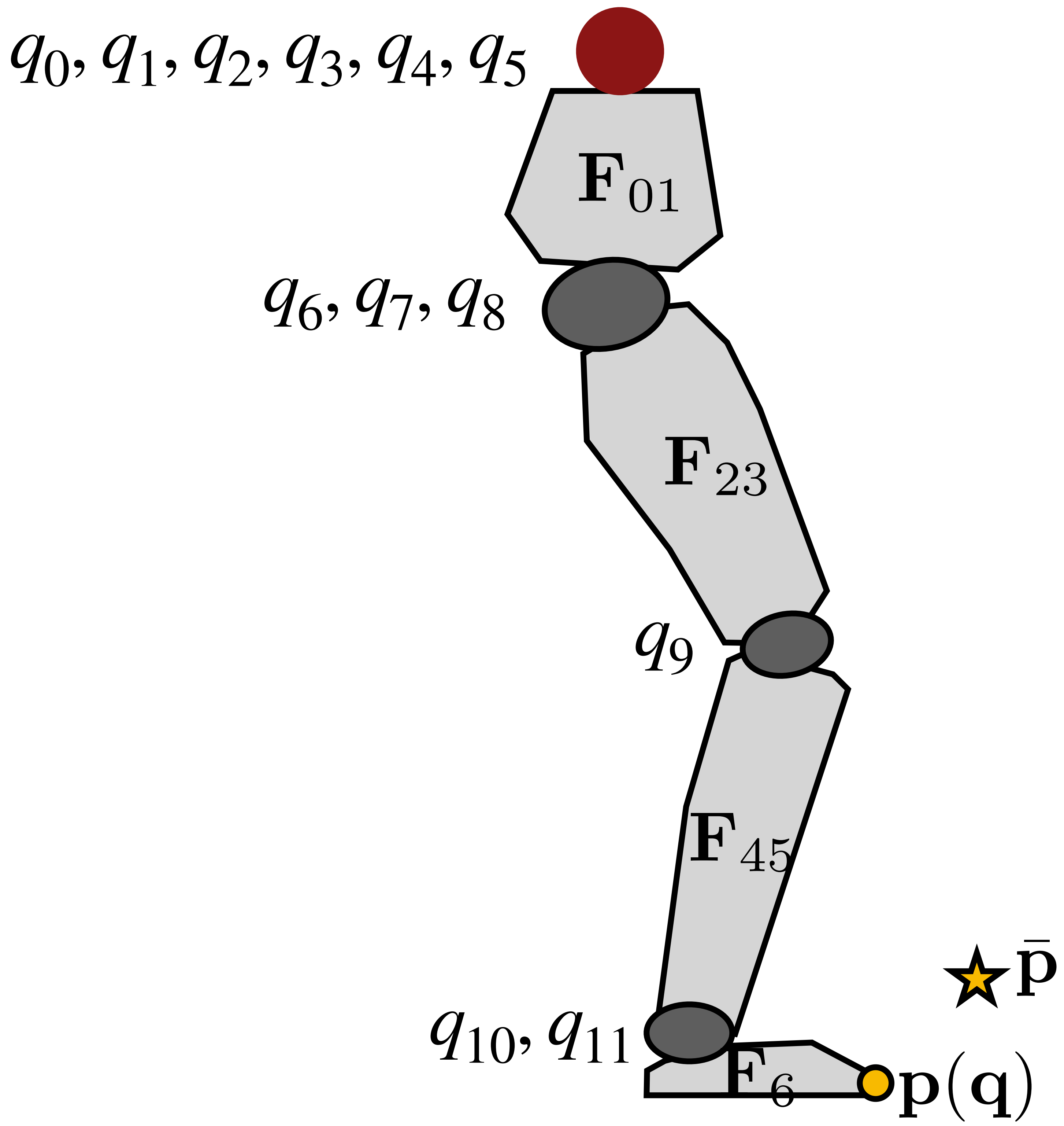
$$= \mathbf{T}(q_0, q_1, q_2)\mathbf{R}(q_3, q_4, q_5)\mathbf{F}_{01}\mathbf{R}(q_6)\mathbf{R}(q_7)\mathbf{R}(q_8)\mathbf{F}_{23}\mathbf{R}(q_9)\mathbf{F}_{45}\mathbf{R}(q_{10})\mathbf{R}(q_{11})\mathbf{F}_6\mathbf{p}_0 - \bar{\mathbf{p}}$$

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: $\mathbf{q} \leftarrow \{q_0, q_1, q_2, q_3, q_4, q_5, q_6, q_7, q_8, q_9, q_{10}, q_{11}\}$



To compute gradient in each optimization iteration,

we need to evaluate the constraint:

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

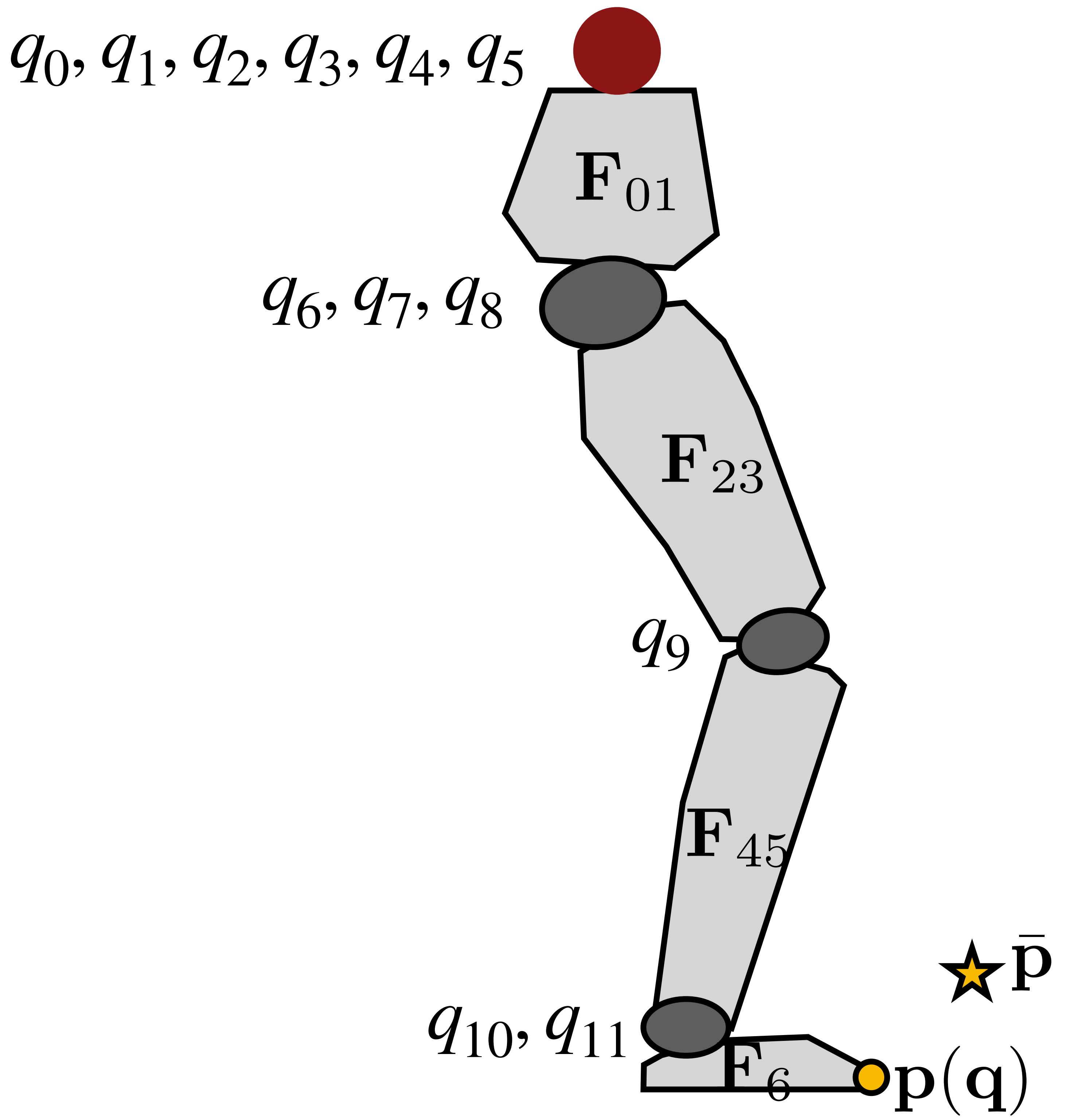
$$= \mathbf{T}(q_0, q_1, q_2)\mathbf{R}(q_3, q_4, q_5)\mathbf{F}_{01}\mathbf{R}(q_6)\mathbf{R}(q_7)\mathbf{R}(q_8)\mathbf{F}_{23}\mathbf{R}(q_9)\mathbf{F}_{45}\mathbf{R}(q_{10})\mathbf{R}(q_{11})\mathbf{F}_6\mathbf{p}_0 - \bar{\mathbf{p}}$$

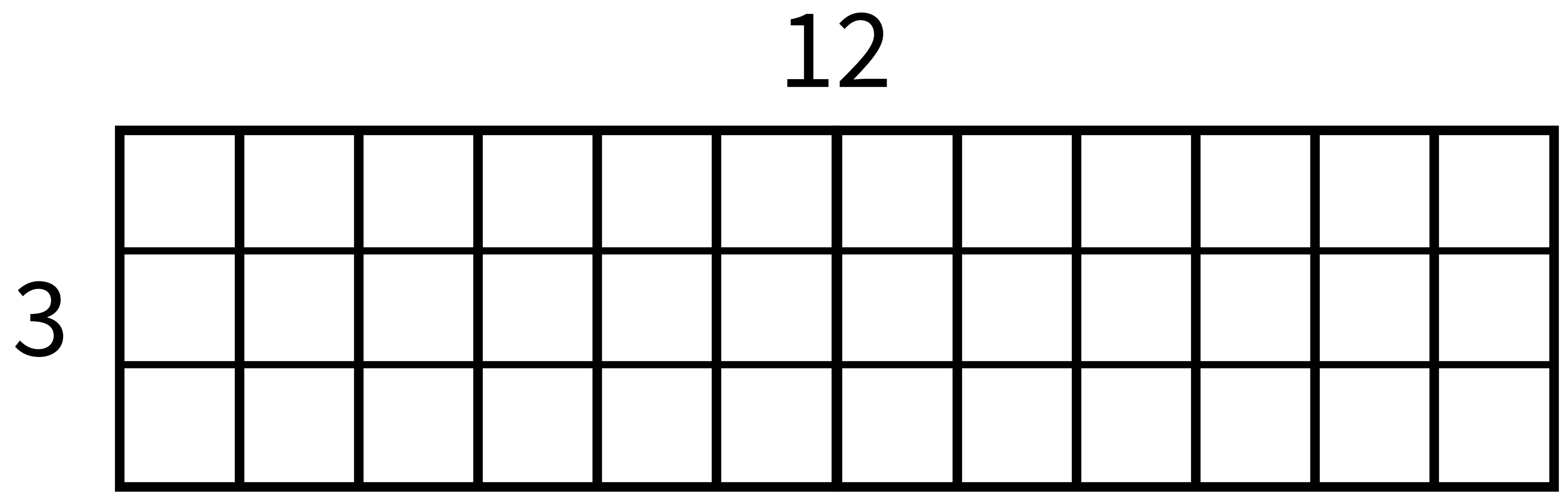
we need to compute the Jacobian matrix:

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

Quiz

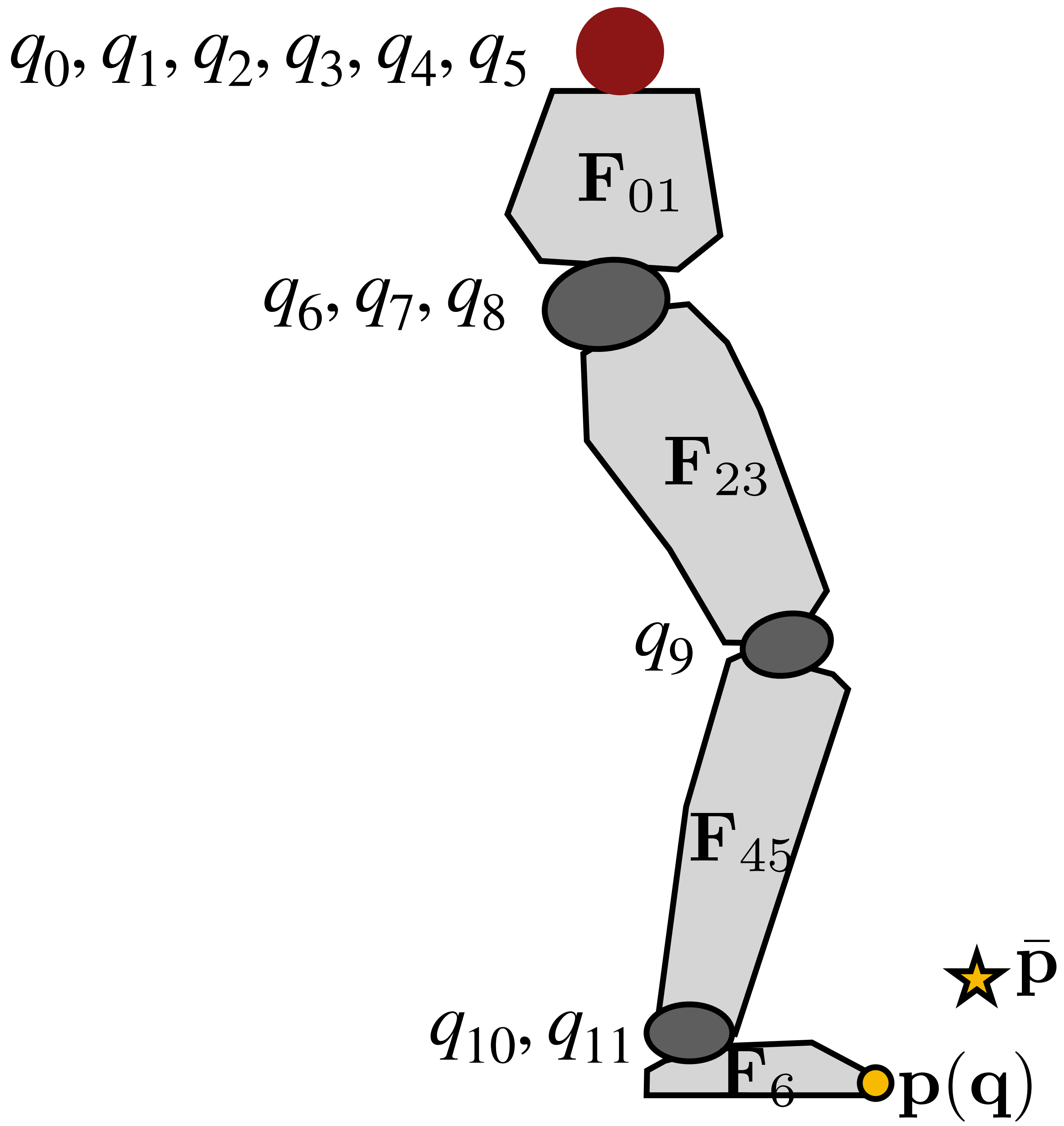
What's the size of Jacobian $\dfrac{\partial \mathbf{C}(\mathbf{q})}{\partial \mathbf{q}}$?





Jacobian matrix

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

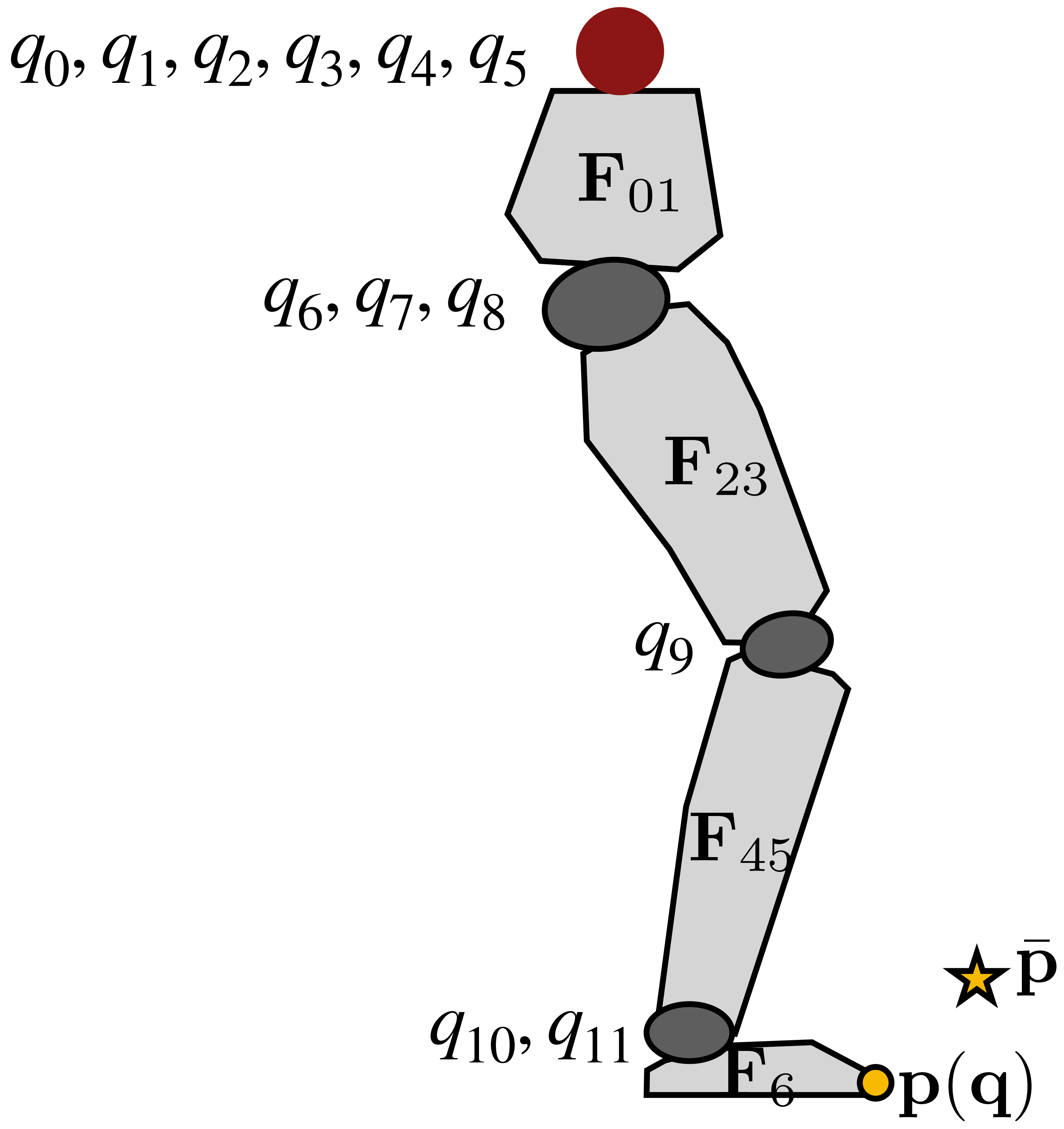


$$\frac{\partial \mathbf{C}(\mathbf{q})}{\partial \mathbf{q}} =$$

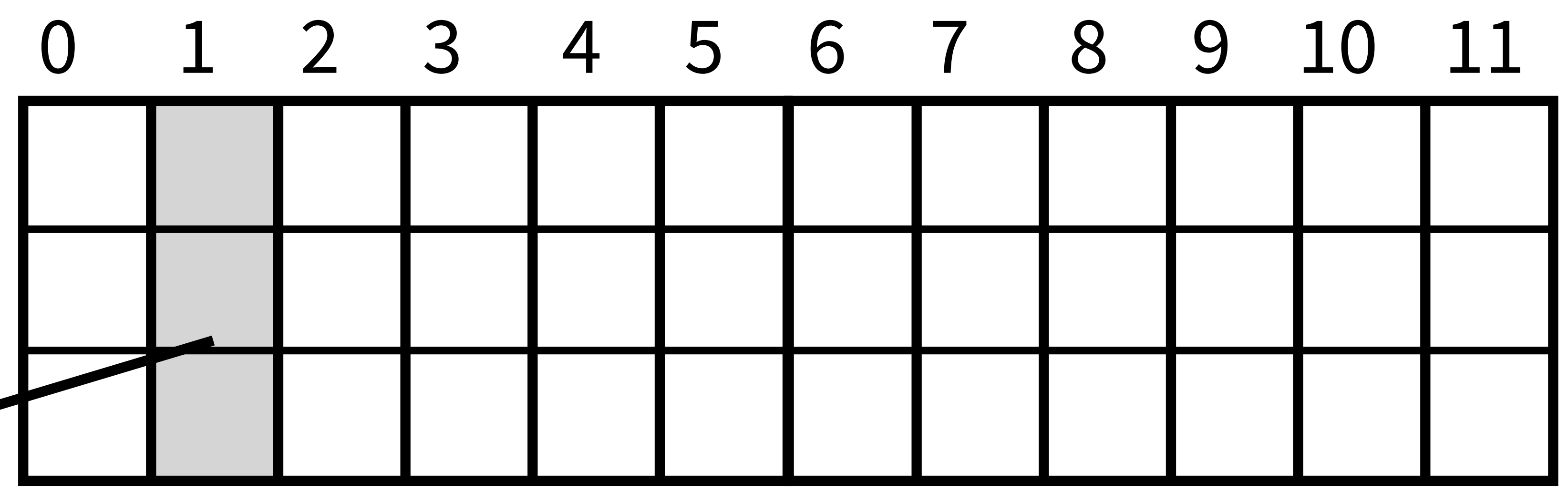
0	1	2	3	4	5	6	7	8	9	10	11

Jacobian matrix

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



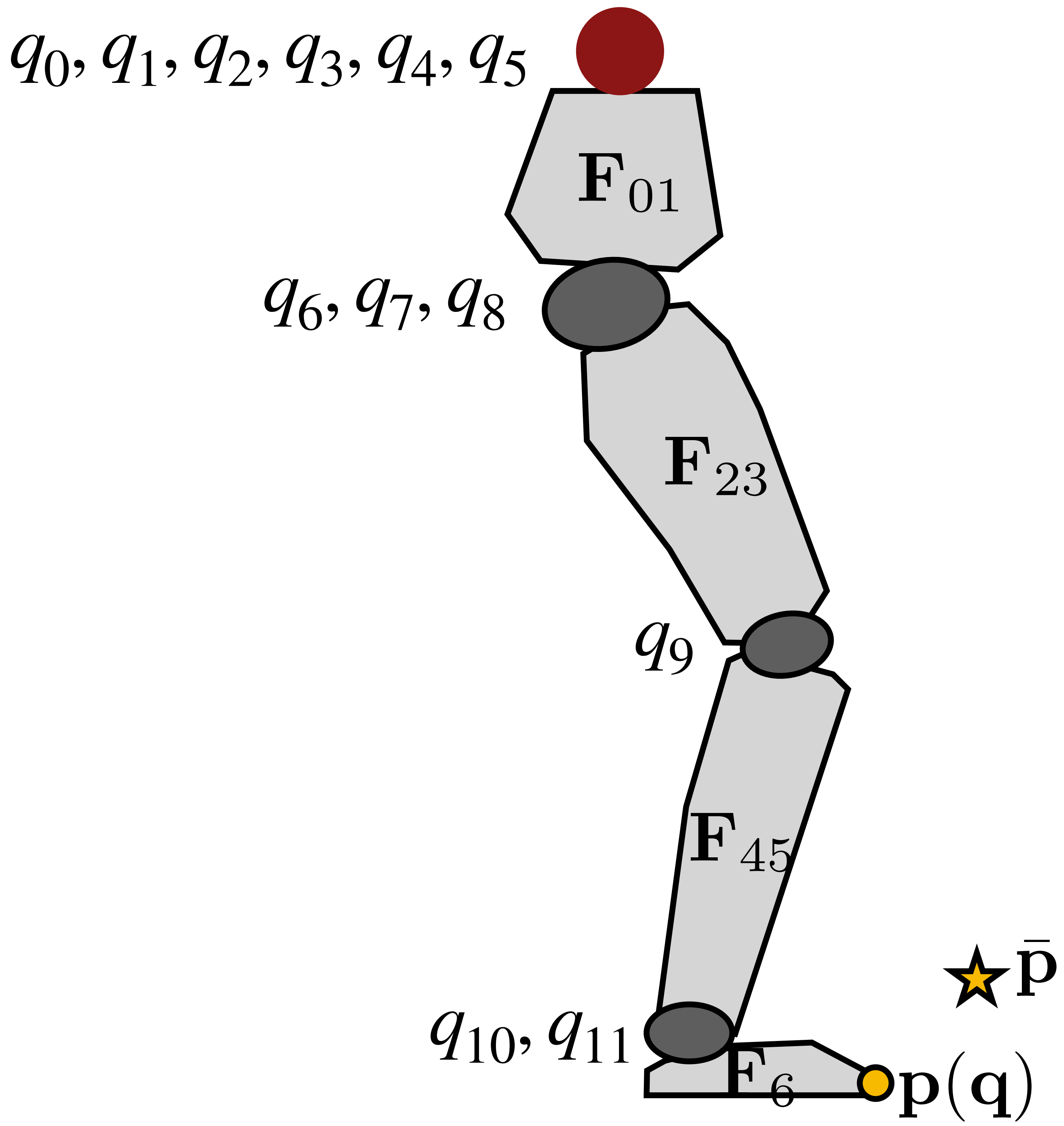
$$\frac{\partial \mathbf{C}(\mathbf{q})}{\partial \mathbf{q}} =$$



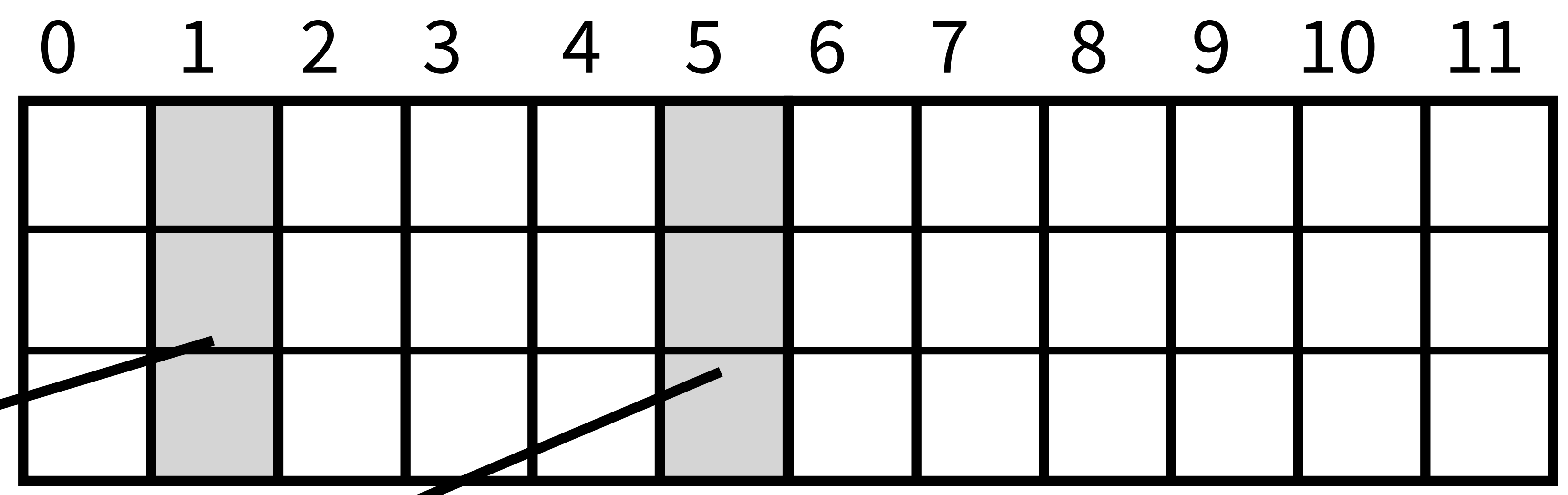
$$\frac{\partial \mathbf{C}(\mathbf{q})}{\partial q_1} = \frac{\partial \mathbf{T}(q_0, q_1, q_2)}{\partial q_1}\mathbf{R}(q_3, q_4, q_5)\mathbf{F}_{01}\mathbf{R}(q_6)\mathbf{R}(q_7)\mathbf{R}(q_8)\mathbf{F}_{23}\mathbf{R}(q_9)\mathbf{F}_{45}\mathbf{R}(q_{10})\mathbf{R}(q_{11})\mathbf{F}_6\mathbf{p}_0$$

Jacobian matrix

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



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

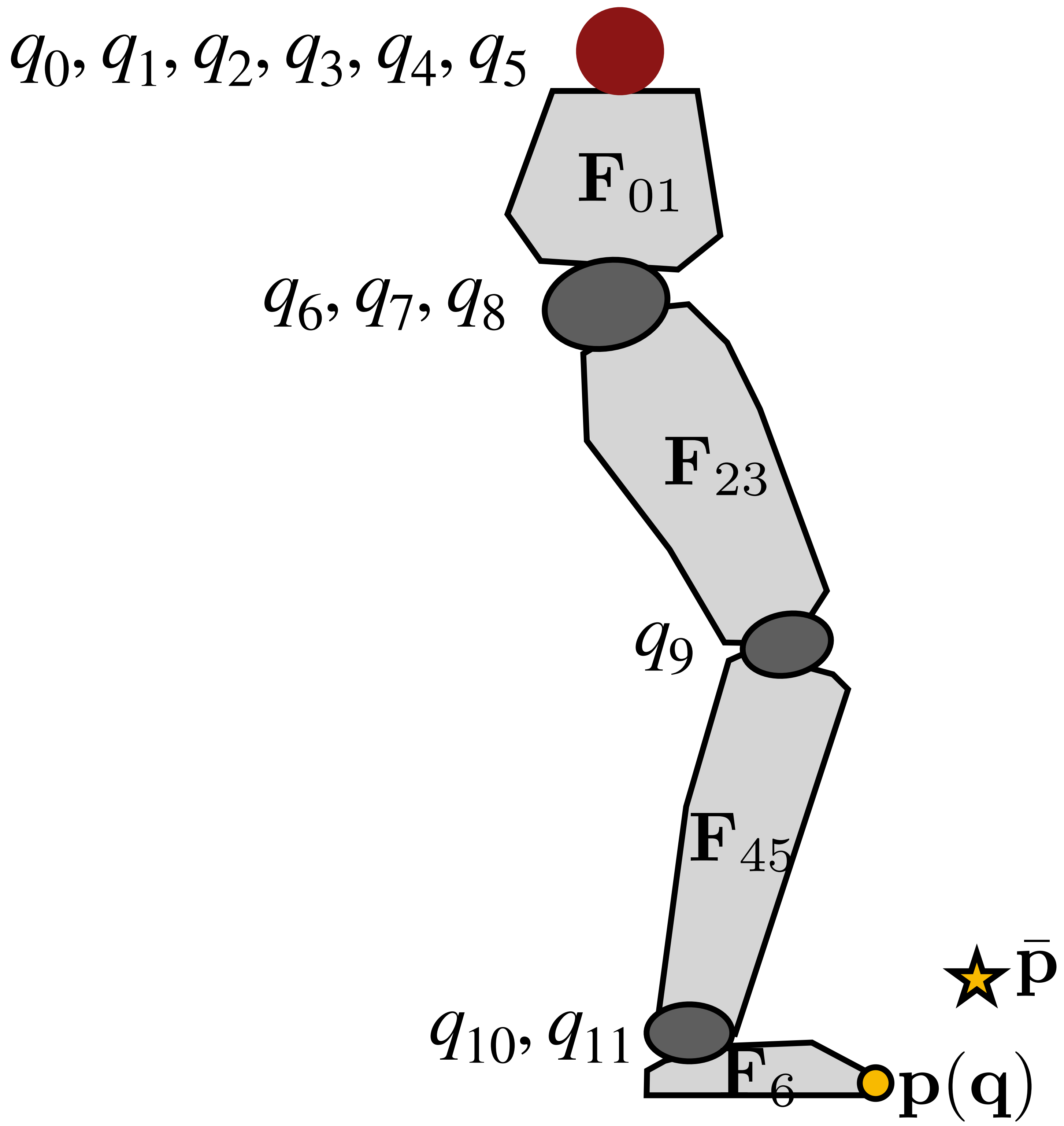


$$\frac{\partial \mathbf{C}(\mathbf{q})}{\partial q_1} = \frac{\partial \mathbf{T}(q_0, q_1, q_2)}{\partial q_1}\mathbf{R}(q_3, q_4, q_5)\mathbf{F}_{01}\mathbf{R}(q_6)\mathbf{R}(q_7)\mathbf{R}(q_8)\mathbf{F}_{23}\mathbf{R}(q_9)\mathbf{F}_{45}\mathbf{R}(q_{10})\mathbf{R}(q_{11})\mathbf{F}_6\mathbf{p}_0$$

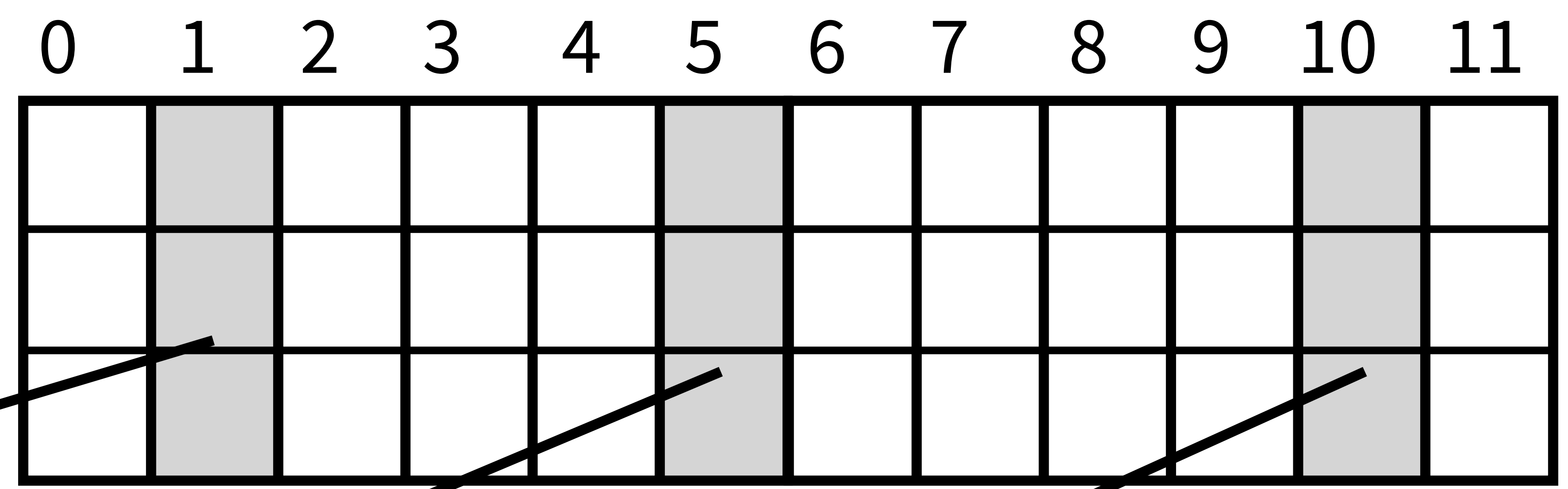
$$\frac{\partial \mathbf{C}(\mathbf{q})}{\partial q_5} = \mathbf{T}(q_0, q_1, q_2)\frac{\partial \mathbf{R}(q_3, q_4, q_5)}{\partial q_5}\mathbf{F}_{01}\mathbf{R}(q_6)\mathbf{R}(q_7)\mathbf{R}(q_8)\mathbf{F}_{23}\mathbf{R}(q_9)\mathbf{F}_{45}\mathbf{R}(q_{10})\mathbf{R}(q_{11})\mathbf{F}_6\mathbf{p}_0$$

Jacobian matrix

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



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



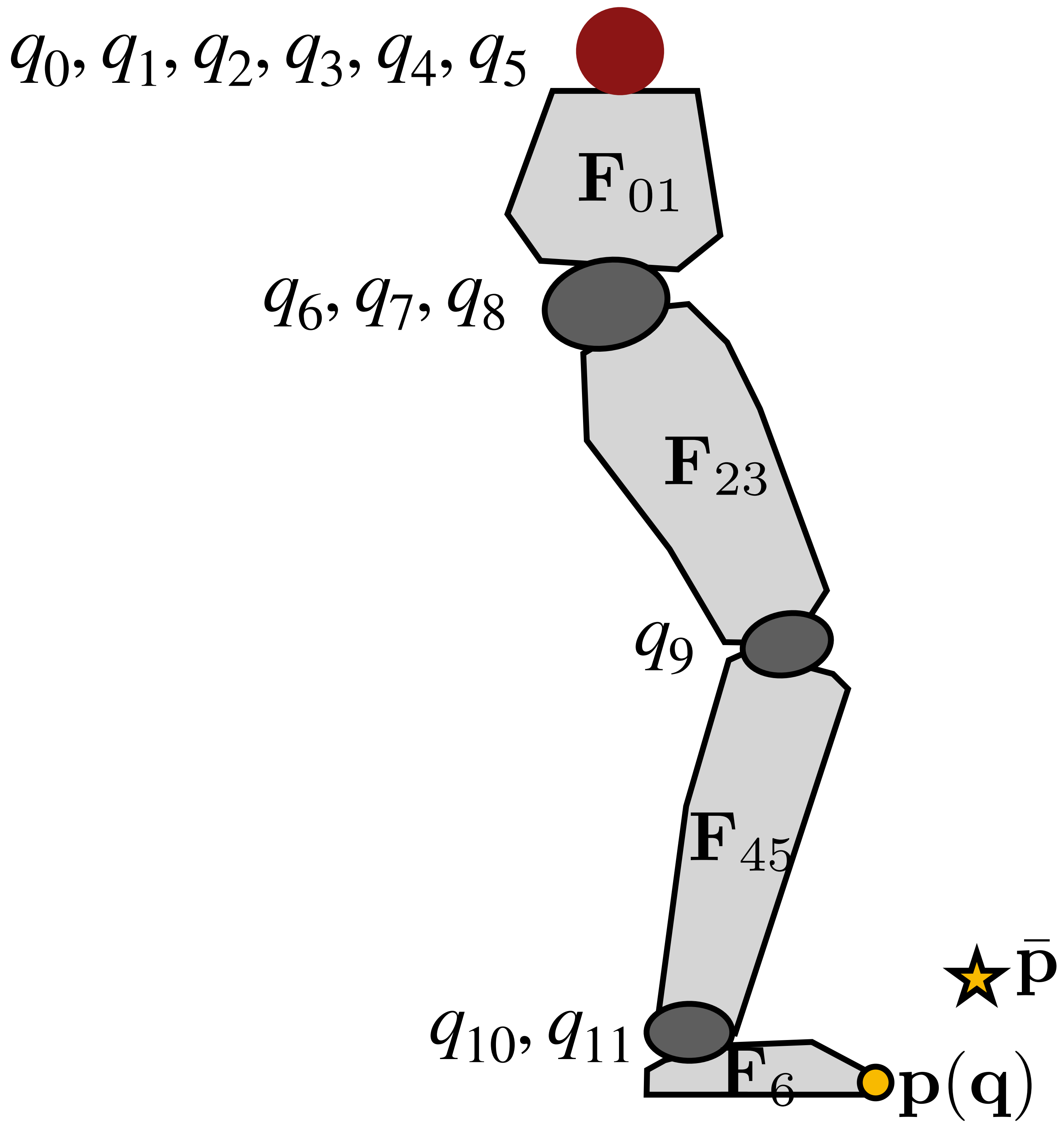
$$\frac{\partial \mathbf{C}(\mathbf{q})}{\partial q_1} = \frac{\partial \mathbf{T}(q_0, q_1, q_2)}{\partial q_1}\mathbf{R}(q_3, q_4, q_5)\mathbf{F}_{01}\mathbf{R}(q_6)\mathbf{R}(q_7)\mathbf{R}(q_8)\mathbf{F}_{23}\mathbf{R}(q_9)\mathbf{F}_{45}\mathbf{R}(q_{10})\mathbf{R}(q_{11})\mathbf{F}_6\mathbf{p}_0$$

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

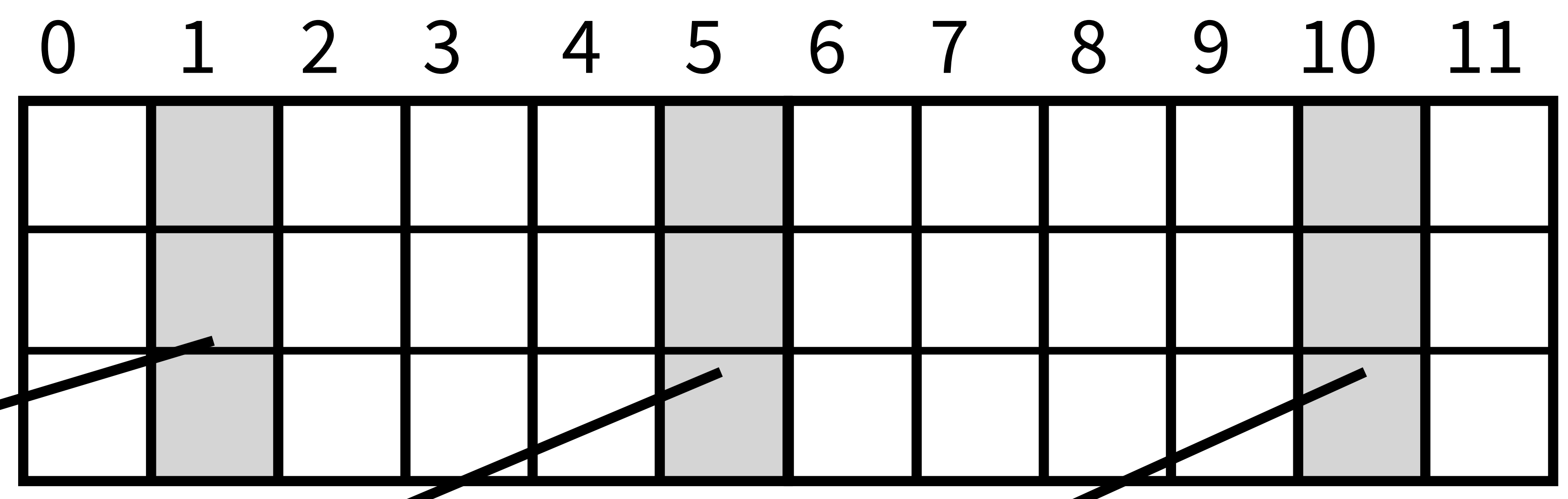
$$\frac{\partial \mathbf{C}(\mathbf{q})}{\partial q_{10}} = \mathbf{T}(q_0, q_1, q_2)\mathbf{R}(q_3, q_4, q_5)\mathbf{F}_{01}\mathbf{R}(q_6)\mathbf{R}(q_7)\mathbf{R}(q_8)\mathbf{F}_{23}\mathbf{R}(q_9)\mathbf{F}_{45}\frac{\partial \mathbf{R}(q_{10})}{\partial q_{10}}\mathbf{R}(q_{11})\mathbf{F}_6\mathbf{p}_0$$

Jacobian matrix

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



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



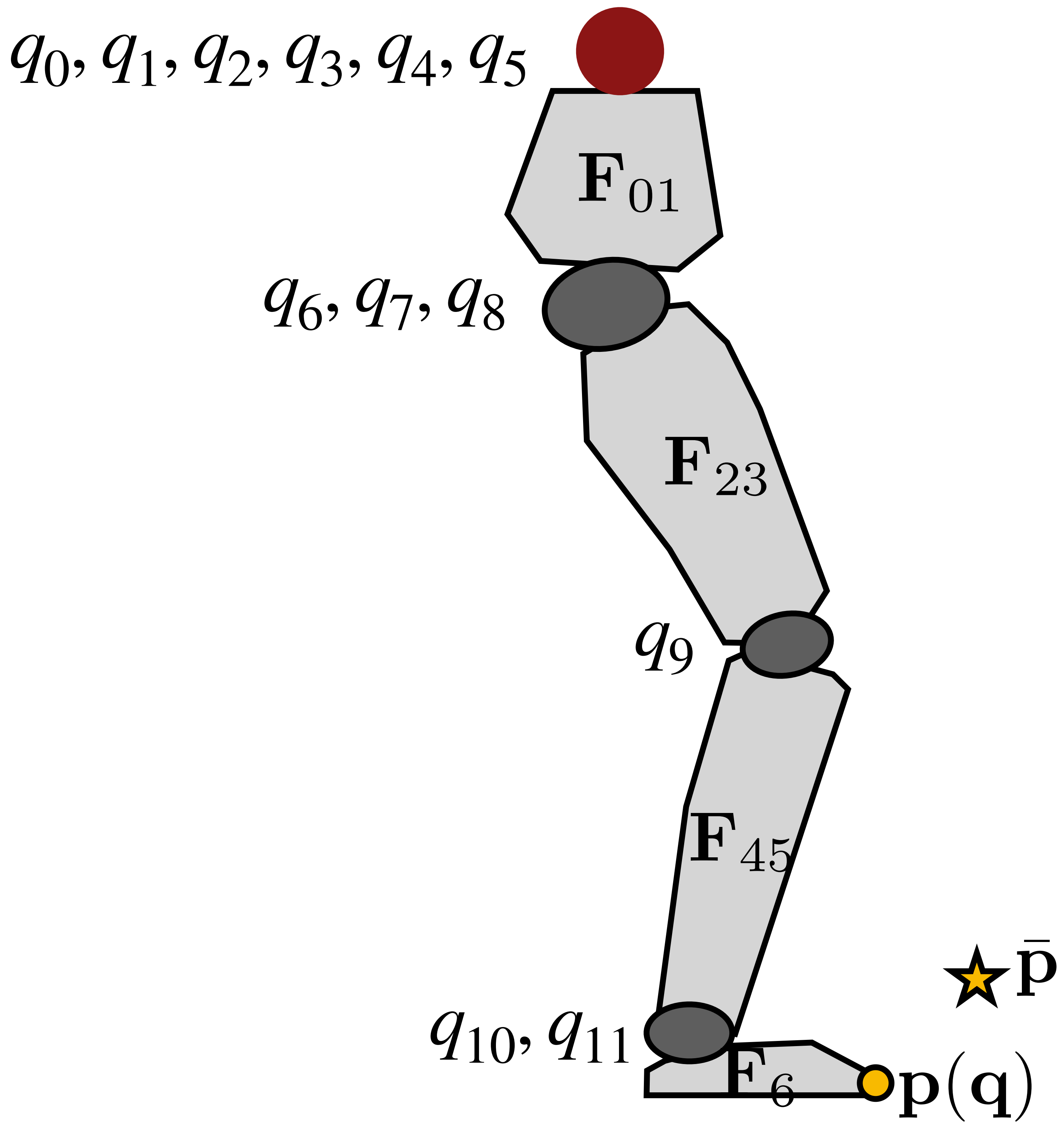
$$\frac{\partial \mathbf{C}(\mathbf{q})}{\partial q_1} = \boxed{\frac{\partial \mathbf{T}(q_0, q_1, q_2)}{\partial q_1}}\mathbf{R}(q_3, q_4, q_5)\mathbf{F}_{01}\mathbf{R}(q_6)\mathbf{R}(q_7)\mathbf{R}(q_8)\mathbf{F}_{23}\mathbf{R}(q_9)\mathbf{F}_{45}\mathbf{R}(q_{10})\mathbf{R}(q_{11})\mathbf{F}_6\mathbf{p}_0$$

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

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

Jacobian matrix

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



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

recall affine transformations

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

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

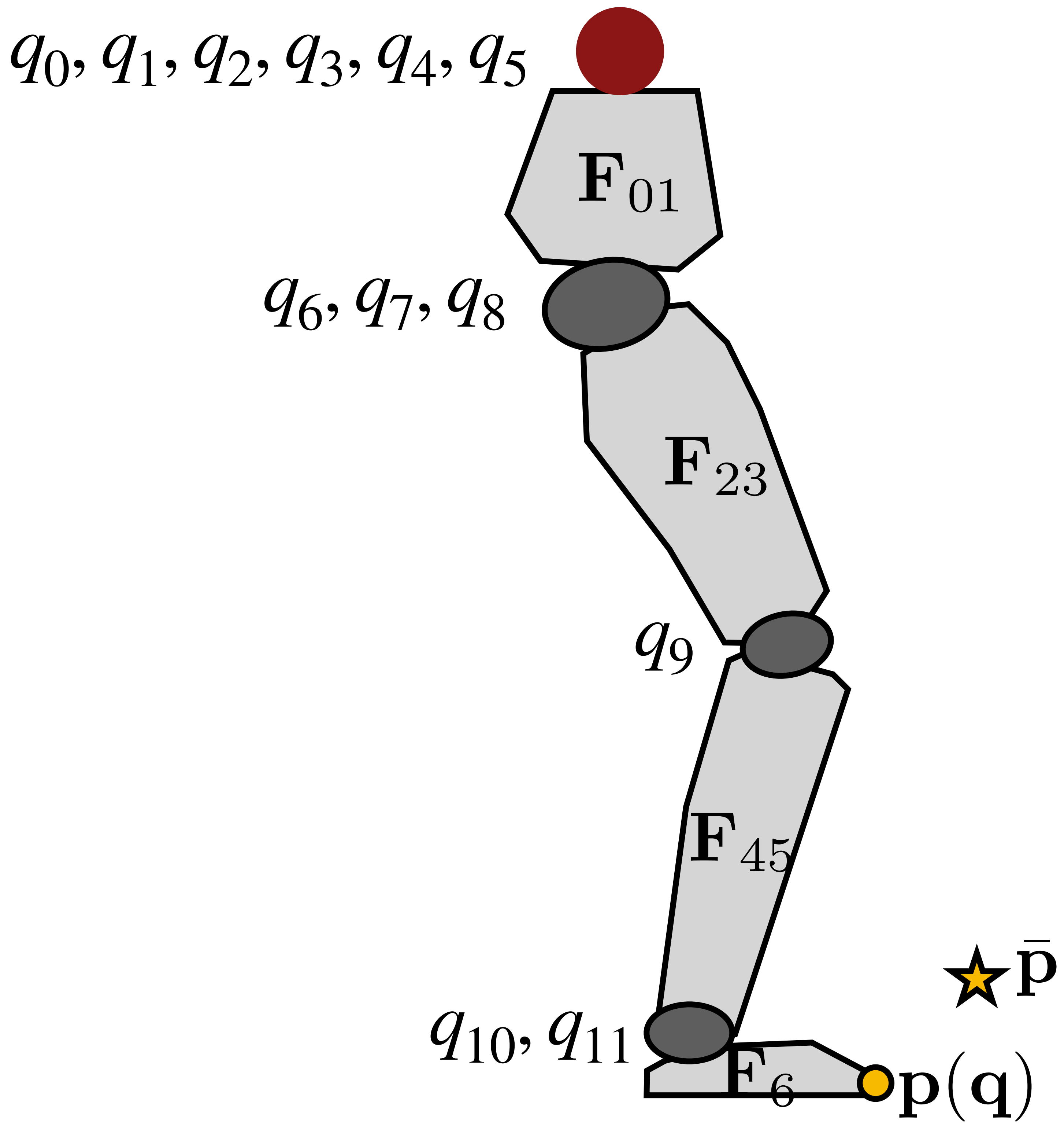
Jacobian matrix

recall affine transformations

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

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

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



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

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

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

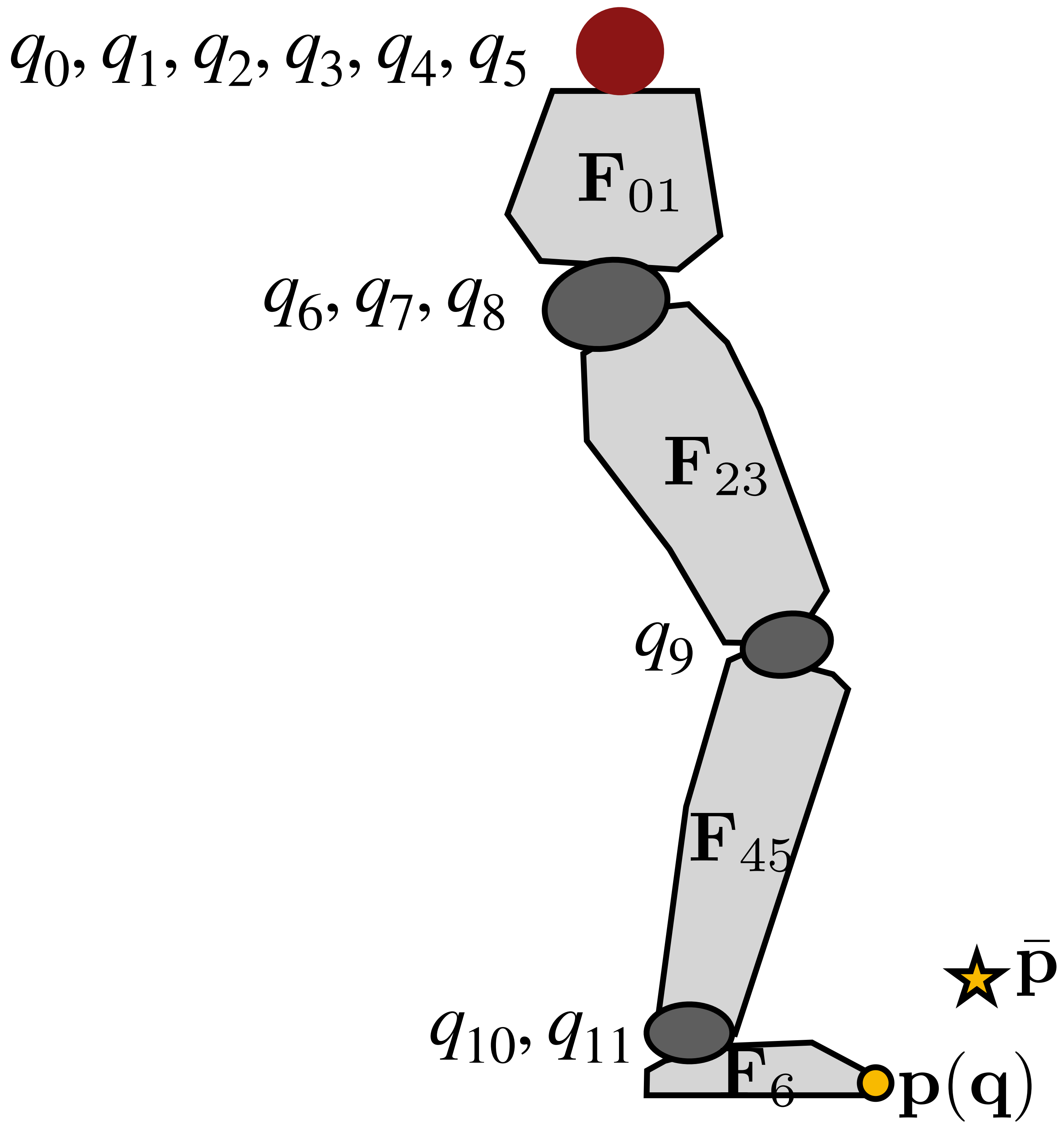
Jacobian matrix

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

recall affine transformations

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

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



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

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

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

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

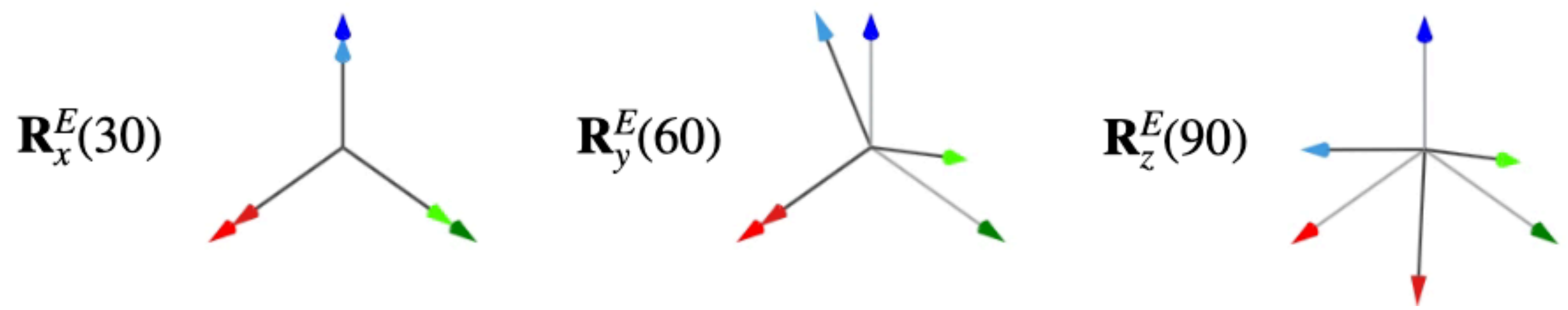
Quiz

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

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

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**

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

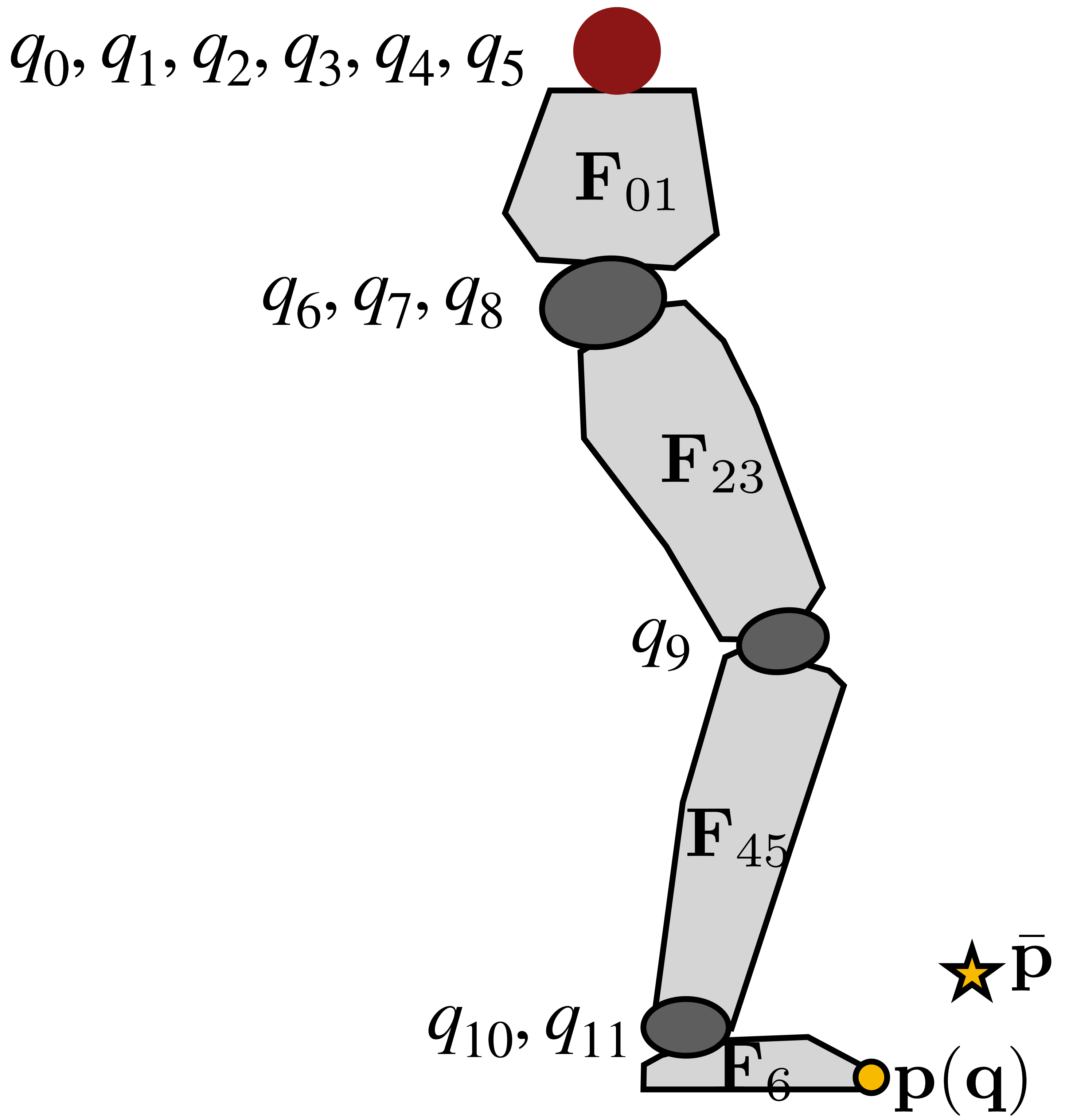
recall quaternion in matrix form

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

Jacobian matrix

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

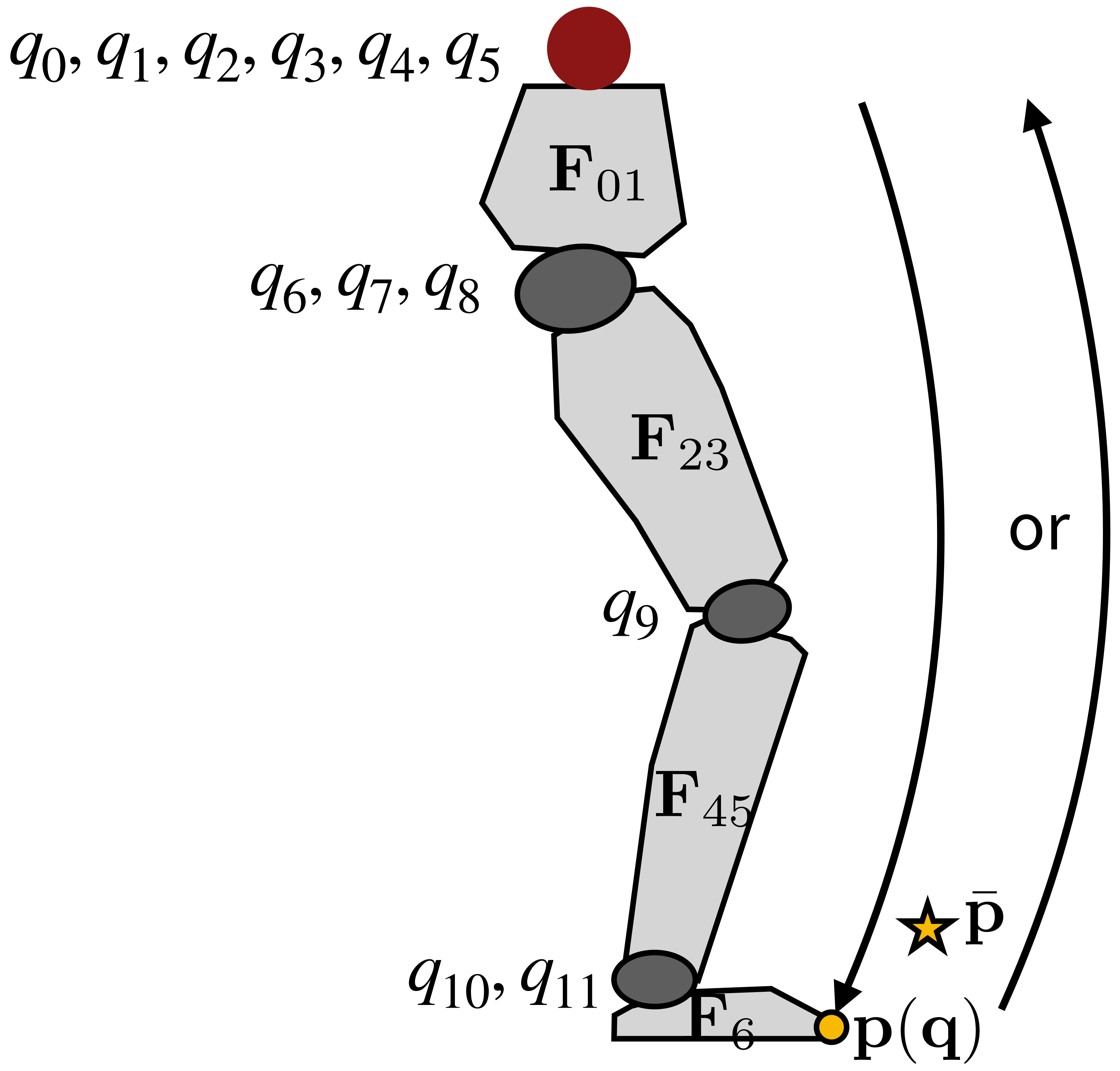
0	1	2	3	4	5	6	7	8	9	10	11



Jacobian matrix

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

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



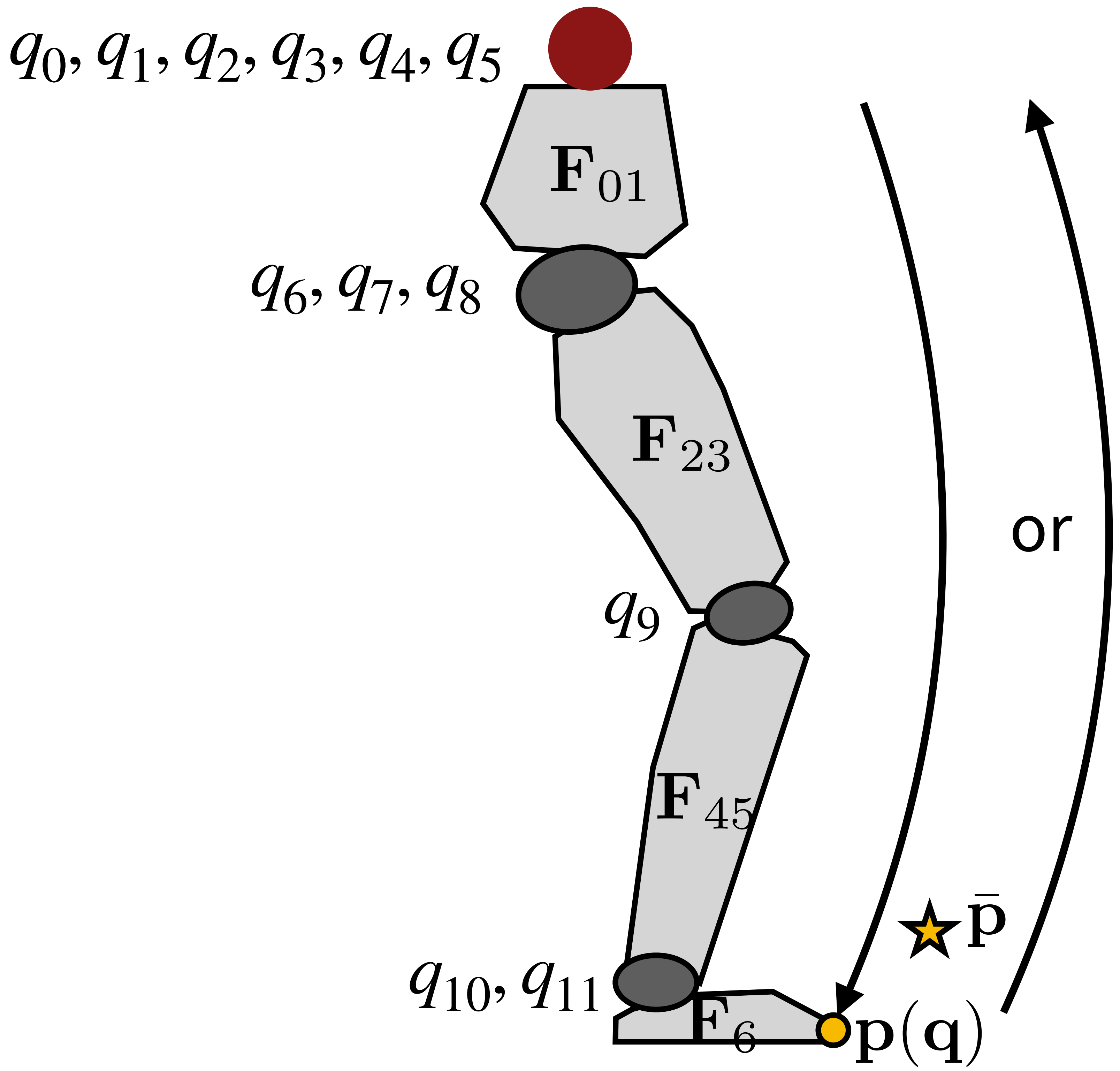
Down the kinematic chain? $q_0, q_1, q_2, q_3, q_4, q_5, q_6, q_7, q_8, q_9, q_{10}, q_{11}$

Up the kinematic chain? $q_{10}, q_{11}, q_9, q_6, q_7, q_8, q_0, q_1, q_2, q_3, q_4, q_5$

Jacobian matrix

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

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



Down the kinematic chain? $q_0, q_1, q_2, q_3, q_4, q_5, q_6, q_7, q_8, q_9, q_{10}, q_{11}$

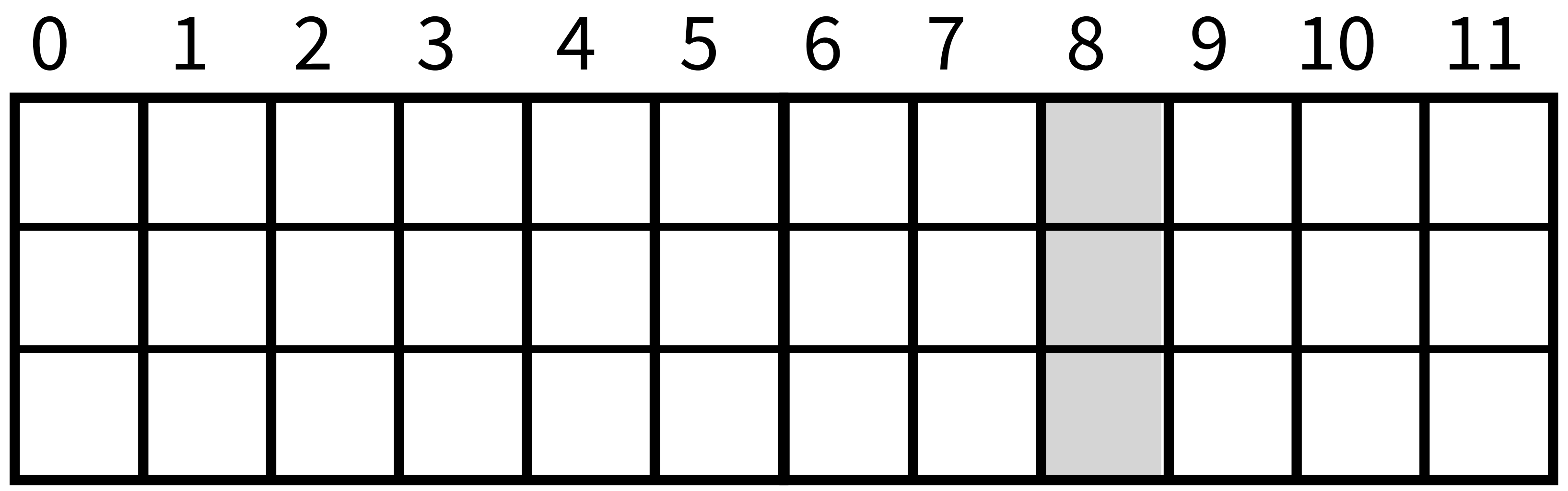
$$\frac{\partial \mathbf{C}(\mathbf{q})}{\partial q_8} = \mathbf{T}(q_0, q_1, q_2)\mathbf{R}(q_3, q_4, q_5)\mathbf{F}_{01}\mathbf{R}(q_6)\mathbf{R}(q_7)\frac{\partial \mathbf{R}(q_8)}{\partial q_8}\mathbf{F}_{23}\mathbf{R}(q_9)\mathbf{F}_{45}\mathbf{R}(q_{10})\mathbf{R}(q_{11})\mathbf{F}_6\mathbf{p}_0$$

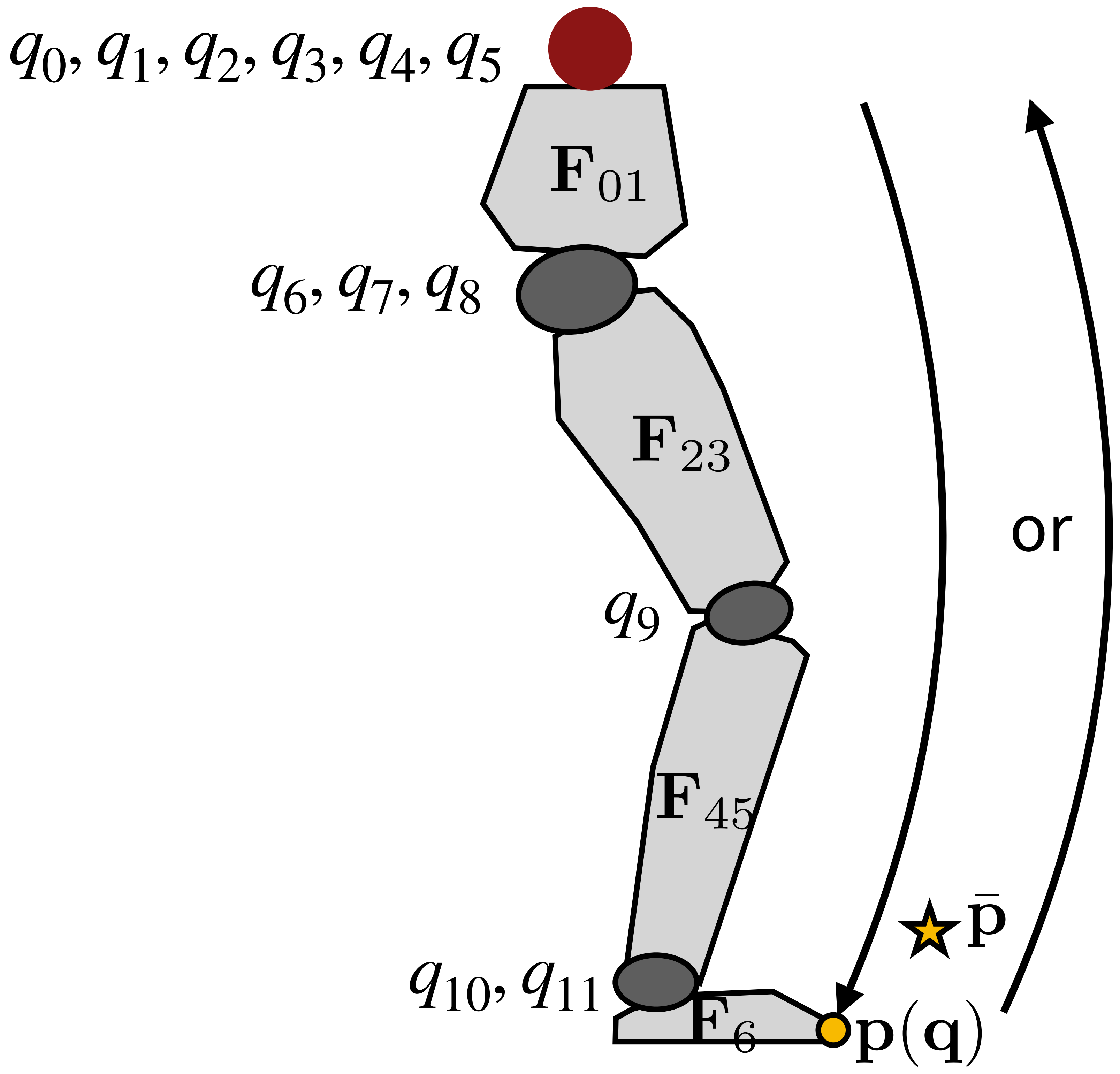
pass down from parent

Up the kinematic chain? $q_{10}, q_{11}, q_9, q_6, q_7, q_8, q_0, q_1, q_2, q_3, q_4, q_5$

Jacobian matrix

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





Down the kinematic chain? $q_0, q_1, q_2, q_3, q_4, q_5, q_6, q_7, q_8, q_9, q_{10}, q_{11}$

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

$$= \mathbf{M}\mathbf{R}(q_6)\mathbf{R}(q_7)\frac{\partial \mathbf{R}(q_8)}{\partial q_8}\mathbf{F}_{23}\mathbf{R}(q_9)\mathbf{F}_{45}\mathbf{R}(q_{10})\mathbf{R}(q_{11})$$

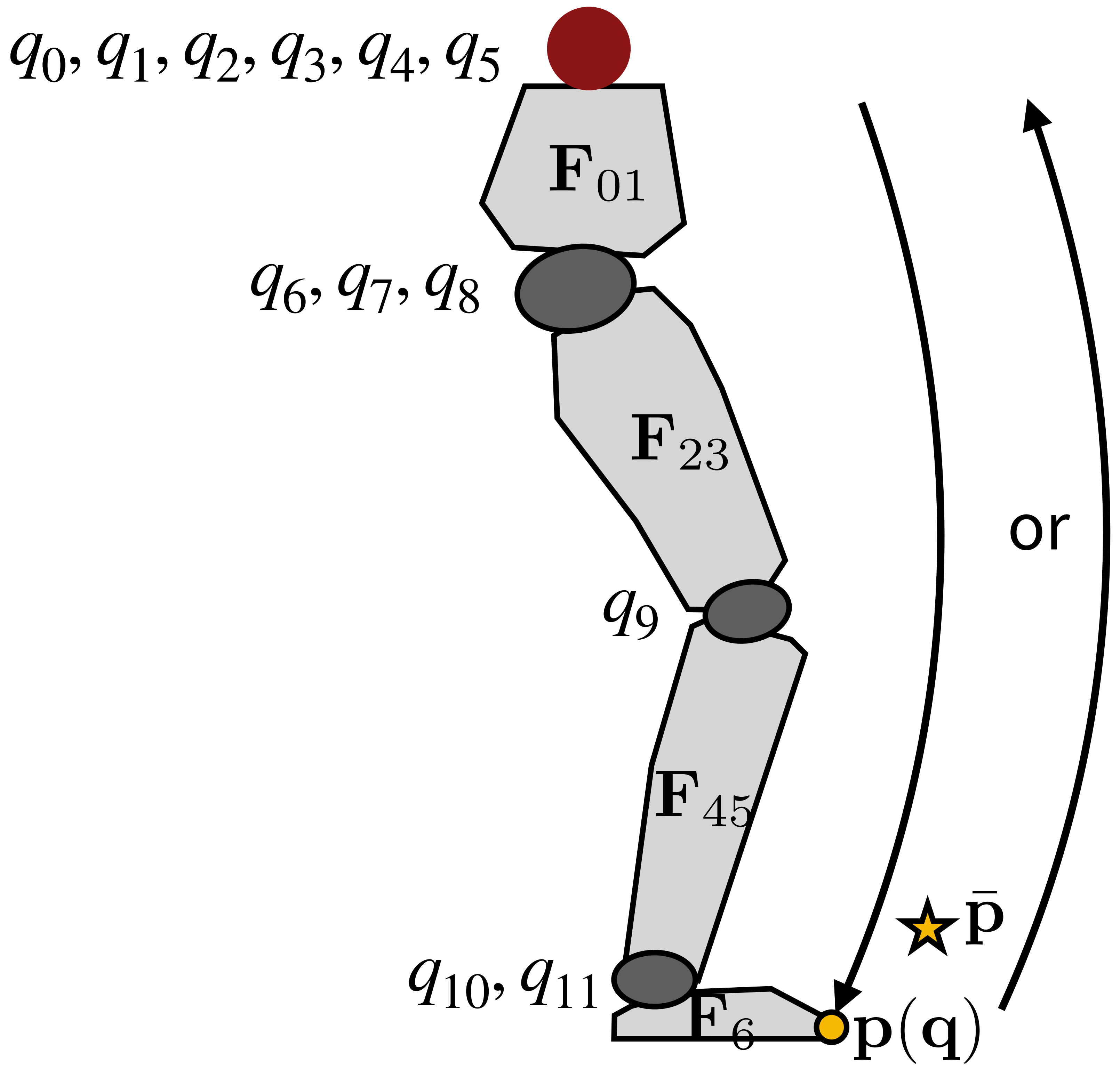
pass down from parent

Up the kinematic chain? $q_{10}, q_{11}, q_9, q_6, q_7, q_8, q_0, q_1, q_2, q_3, q_4, q_5$

Jacobian matrix

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

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



Down the kinematic chain? $q_0, q_1, q_2, q_3, q_4, q_5, q_6, q_7, q_8, q_9, q_{10}, q_{11}$

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

pass down from parent

$$= \mathbf{M}\mathbf{R}(q_6)\mathbf{R}(q_7)\frac{\partial \mathbf{R}(q_8)}{\partial q_8}\mathbf{F}_{23}\mathbf{R}(q_9)\mathbf{F}_{45}\mathbf{R}(q_{10})\mathbf{R}(q_{11})\mathbf{F}_6\mathbf{p}_0$$

Up the kinematic chain? $q_{10}, q_{11}, q_9, q_6, q_7, q_8, q_0, q_1$

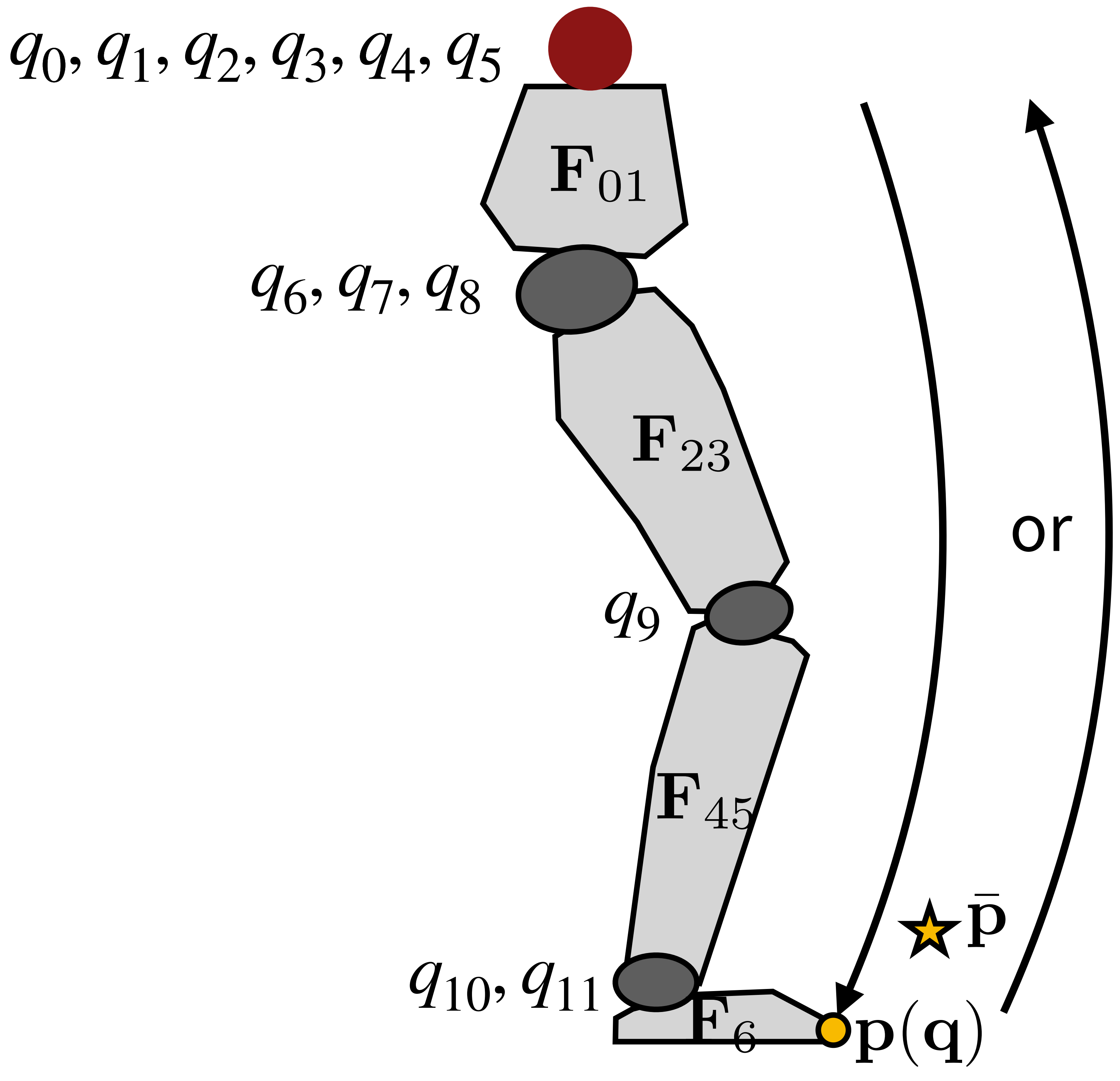
pass up from child

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

Jacobian matrix

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

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



Down the kinematic chain? $q_0, q_1, q_2, q_3, q_4, q_5, q_6, q_7, q_8, q_9, q_{10}, q_{11}$

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

$$= \mathbf{M}\mathbf{R}(q_6)\mathbf{R}(q_7)\frac{\partial \mathbf{R}(q_8)}{\partial q_8}\mathbf{F}_{23}\mathbf{R}(q_9)\mathbf{F}_{45}\mathbf{R}(q_{10})\mathbf{R}(q_{11})$$

pass down from parent

Up the kinematic chain? $q_{10}, q_{11}, q_9, q_6, q_7, q_8, q_0, q_1$

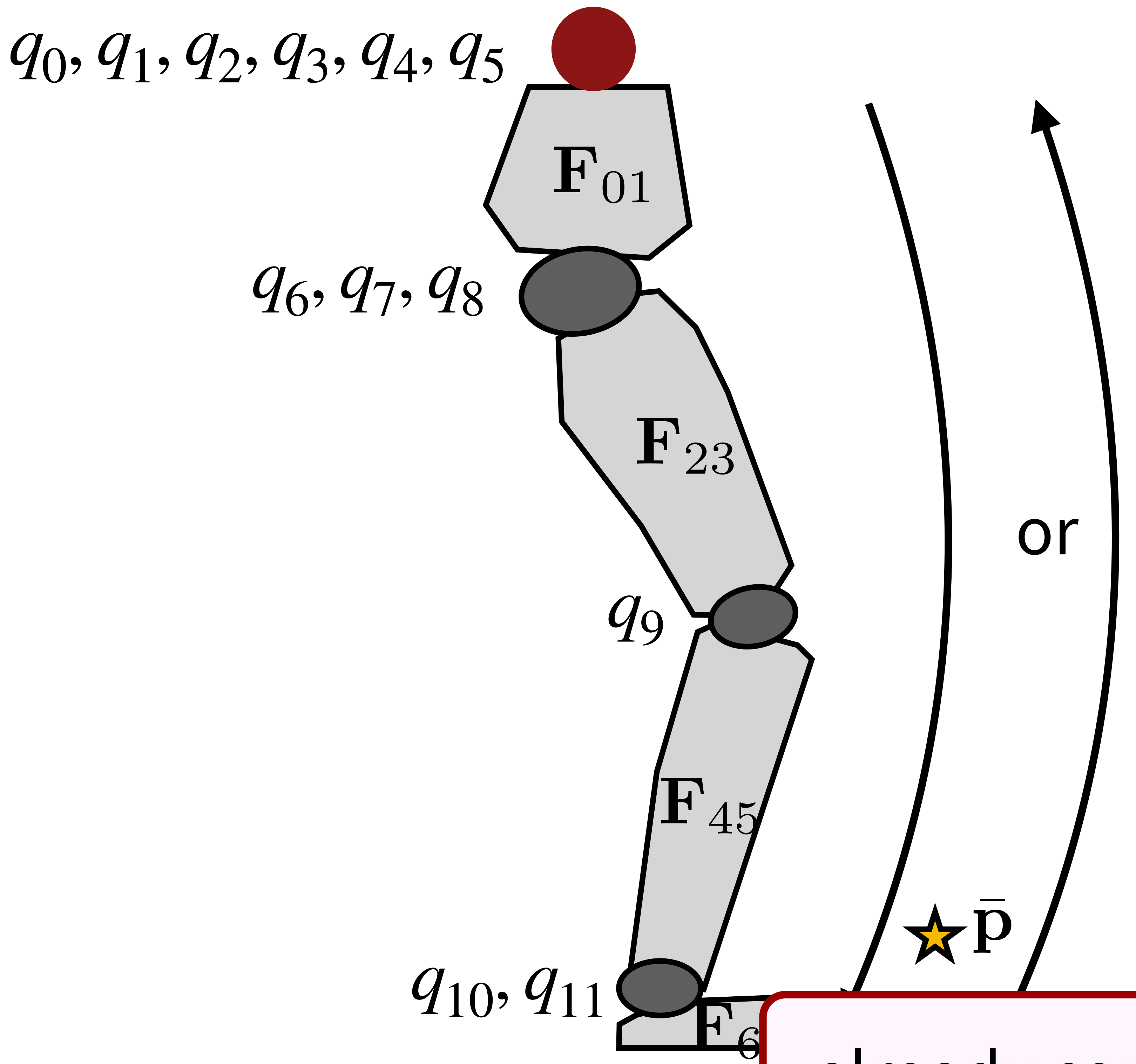
pass up from child

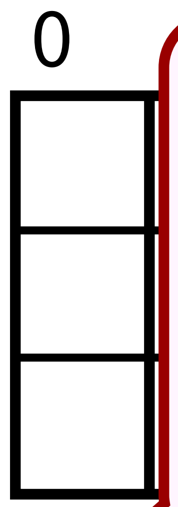
$$\frac{\partial \mathbf{C}(\mathbf{q})}{\partial q_8} = \mathbf{T}(q_0, q_1, q_2)\mathbf{R}(q_3, q_4, q_5)\mathbf{F}_{01}\mathbf{R}(q_6)\mathbf{R}(q_7)\frac{\partial \mathbf{R}(q_8)}{\partial q_8}\mathbf{F}_{23}\mathbf{R}(q_9)\mathbf{F}_{45}\mathbf{R}(q_{10})\mathbf{R}(q_{11})\mathbf{F}_6\mathbf{p}_0$$

$$= \mathbf{T}(q_0, q_1, q_2)\mathbf{R}(q_3, q_4, q_5)\mathbf{F}_{01}\mathbf{R}(q_6)\mathbf{R}(q_7)\frac{\partial \mathbf{R}(q_8)}{\partial q_8}\mathbf{v}$$

Jacobian matrix

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





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

Down the kinematic chain? $q_0, q_1, q_2, q_3, q_4, q_5, q_6, q_7, q_8, q_9, q_{10}, q_{11}$

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

pass down from parent

$$= \mathbf{M}\mathbf{R}(q_6)\mathbf{R}(q_7)\frac{\partial \mathbf{R}(q_8)}{\partial q_8}\mathbf{F}_{23}\mathbf{R}(q_9)\mathbf{F}_{45}\mathbf{R}(q_{10})\mathbf{R}(q_{11})\mathbf{F}_6\mathbf{p}_0$$

Up the kinematic chain? $q_{10}, q_{11}, q_9, q_6, q_7, q_8, q_0, q_1$

pass up from child

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

already computed in graphics matrix stack

$$= \mathbf{T}(q_0, q_1, q_2)\mathbf{R}(q_3, q_4, q_5)\mathbf{F}_{01}\mathbf{R}(q_6)\mathbf{R}(q_7)\frac{\partial \mathbf{R}(q_8)}{\partial q_8}\mathbf{v}$$

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

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

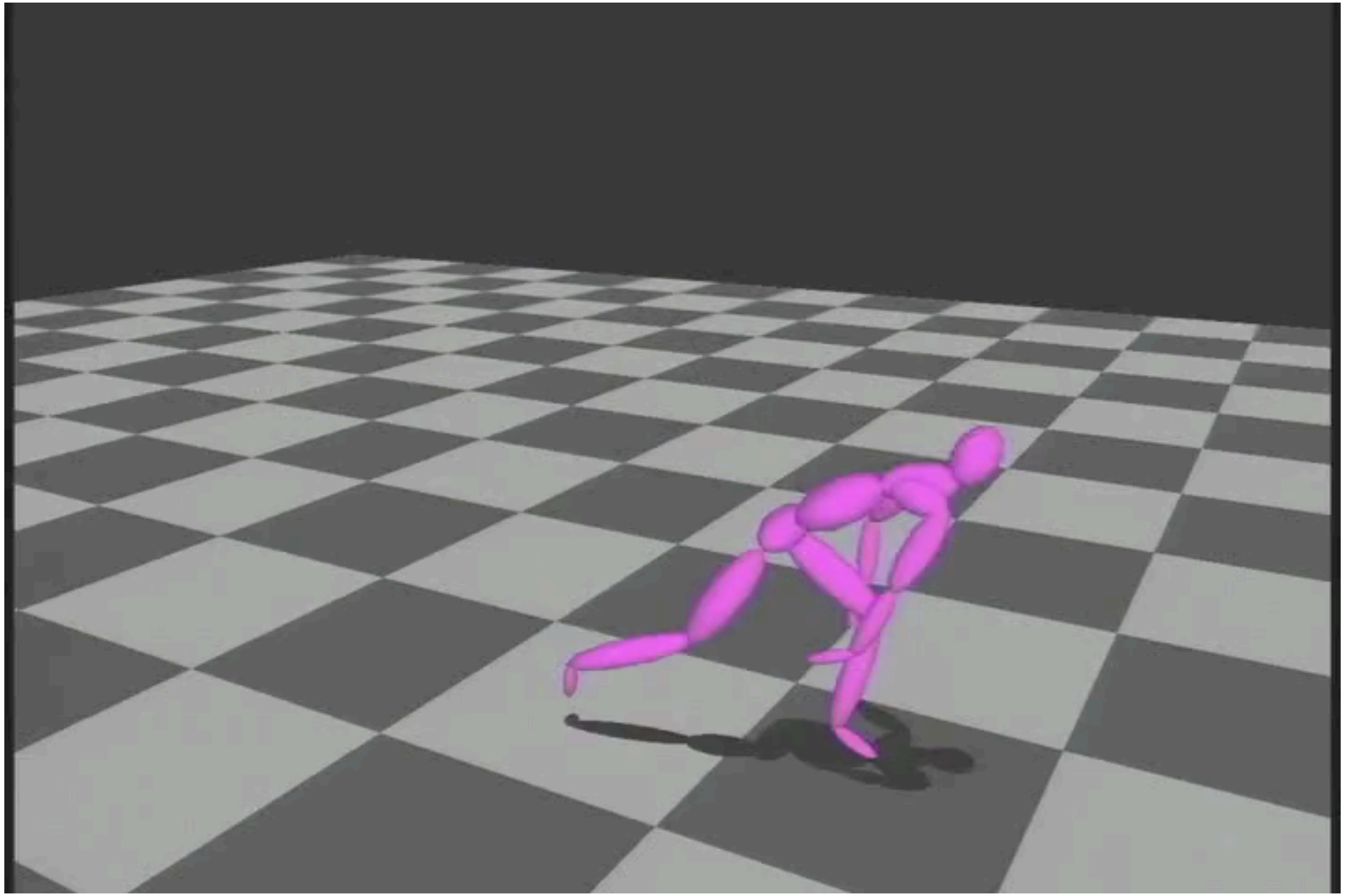
$g(\mathbf{q})$ could be minimizing the velocity of joint angles

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

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

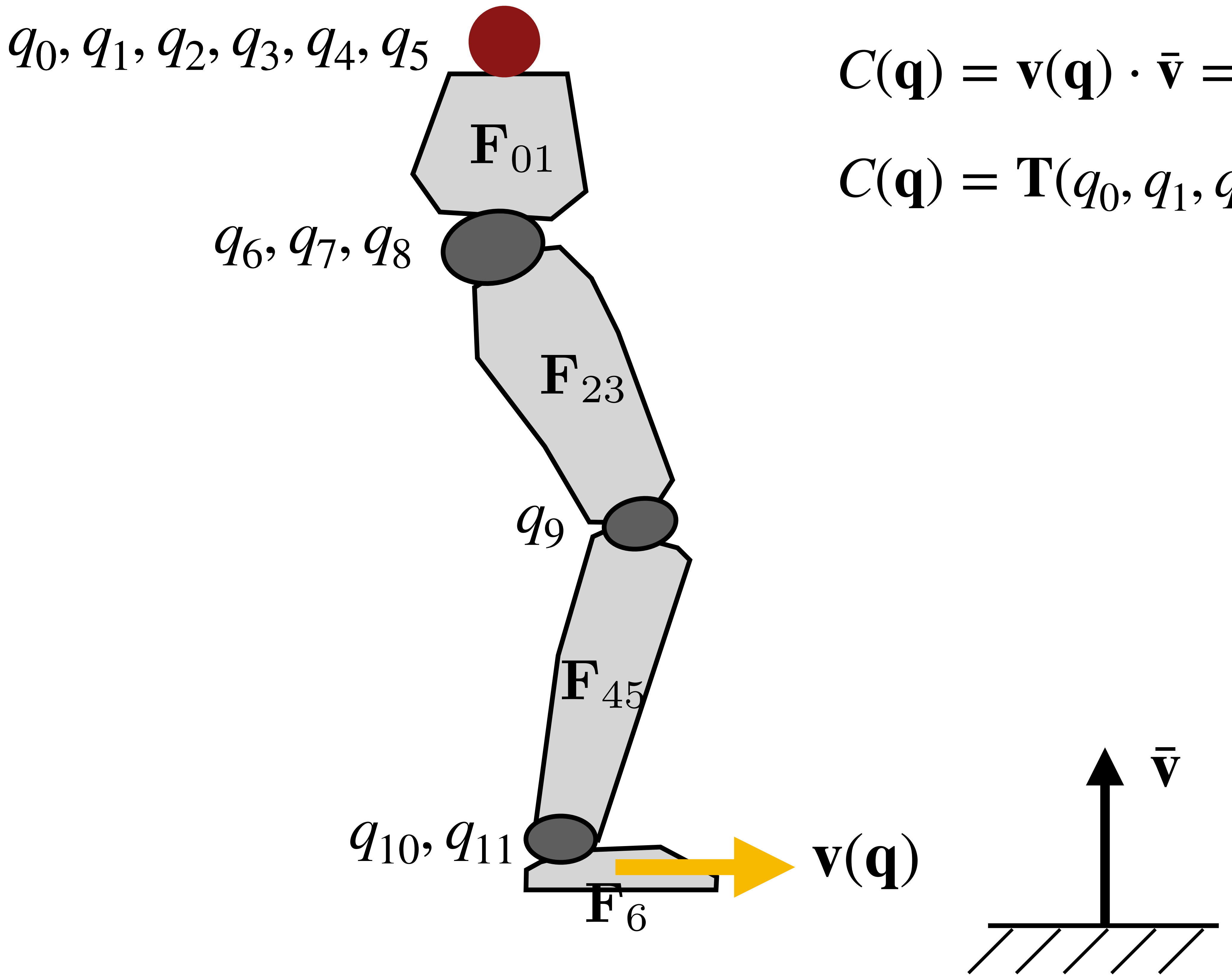
$$g(\mathbf{q}) = \|f_{com}(\mathbf{q}) - f_{feet}(\mathbf{q})\|^2$$

Realistic human poses



Orientation constraints

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

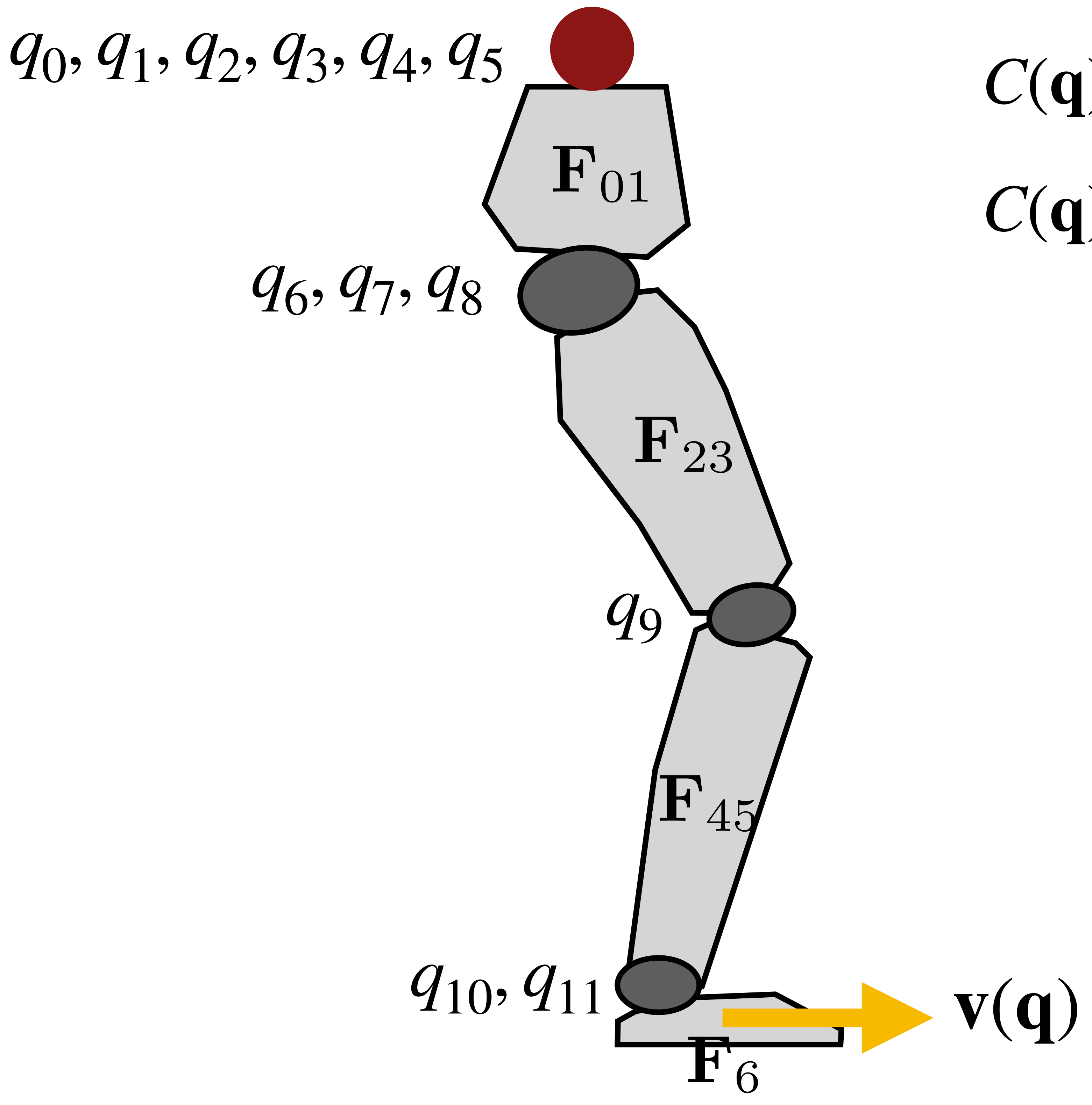


$$C(\mathbf{q}) = \mathbf{v}(\mathbf{q}) \cdot \bar{\mathbf{v}} = 0$$

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

Orientation constraints

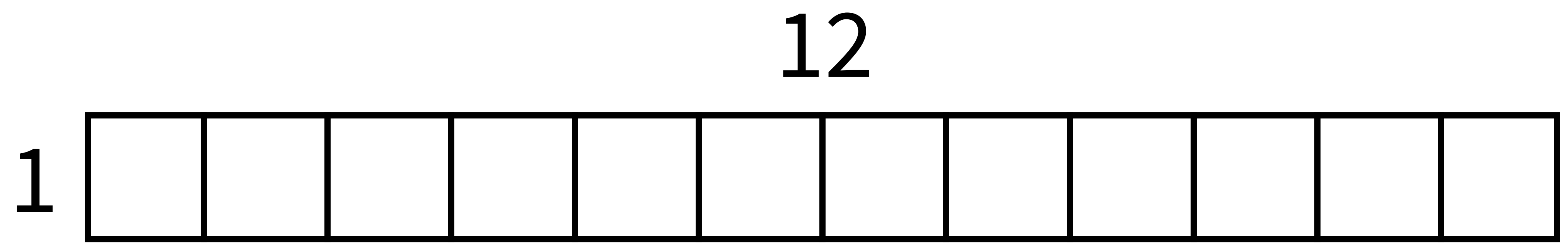
How to formulate a constraint that enforces the foot parallel to the ground?

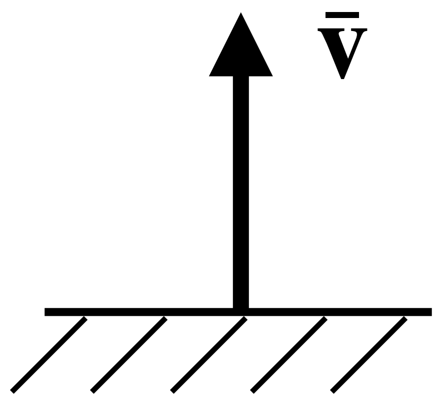


$$C(\mathbf{q}) = \mathbf{v}(\mathbf{q}) \cdot \bar{\mathbf{v}} = 0$$

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

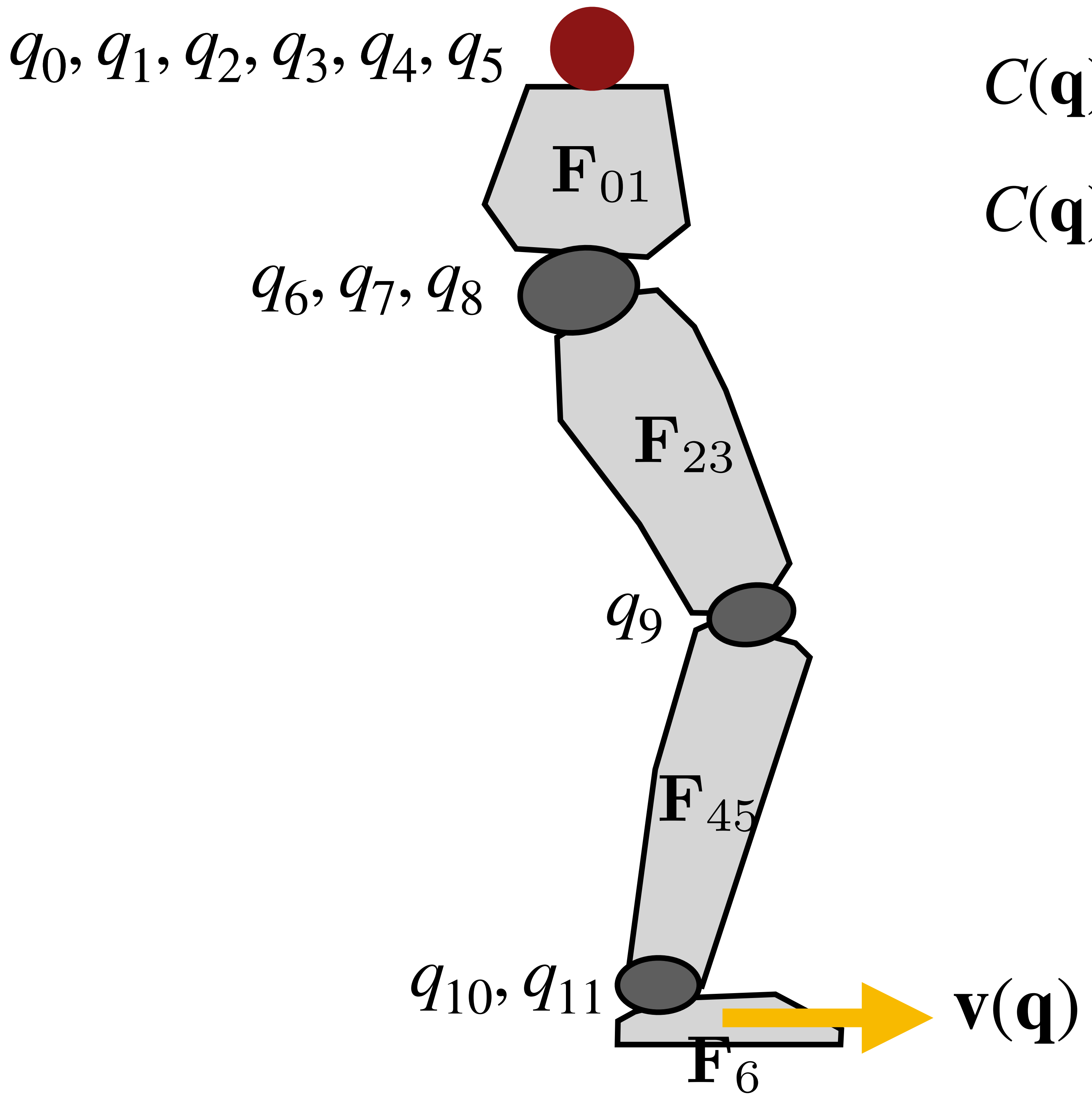
size of $\dfrac{\partial C}{\partial \mathbf{q}}$

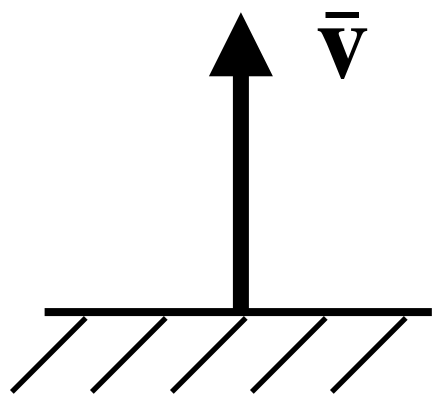




Orientation constraints

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

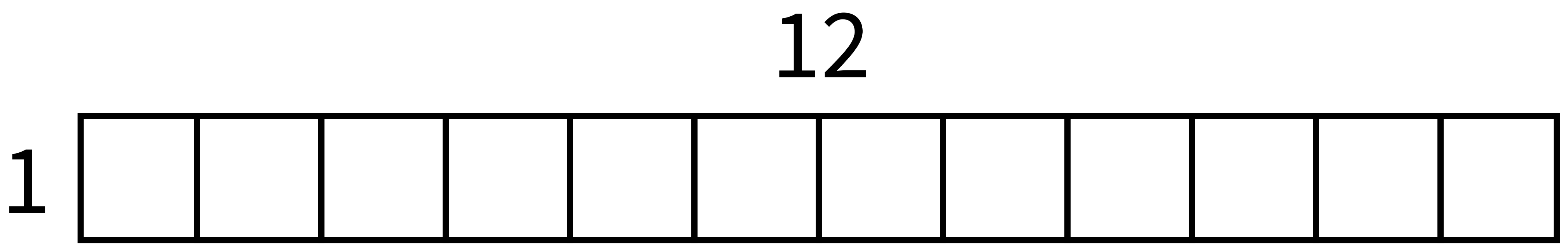




$$C(\mathbf{q}) = \mathbf{v}(\mathbf{q}) \cdot \bar{\mathbf{v}} = 0$$

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

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



$\mathbf{v}_0$ 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**

unknown pose

target

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

- **Suitable for robotic manipulation problems in which**

typically 7 for an arm

typically 6

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

- **Return multiple solutions,**

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

Alternative approach: Iterative IK

Jacobi inverse method:

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Alternative: Heuristic-based IK

Contents lists available at ScienceDirect

Graphical Models

journal homepage: www.elsevier.com/locate/gmod

ELSEVIER

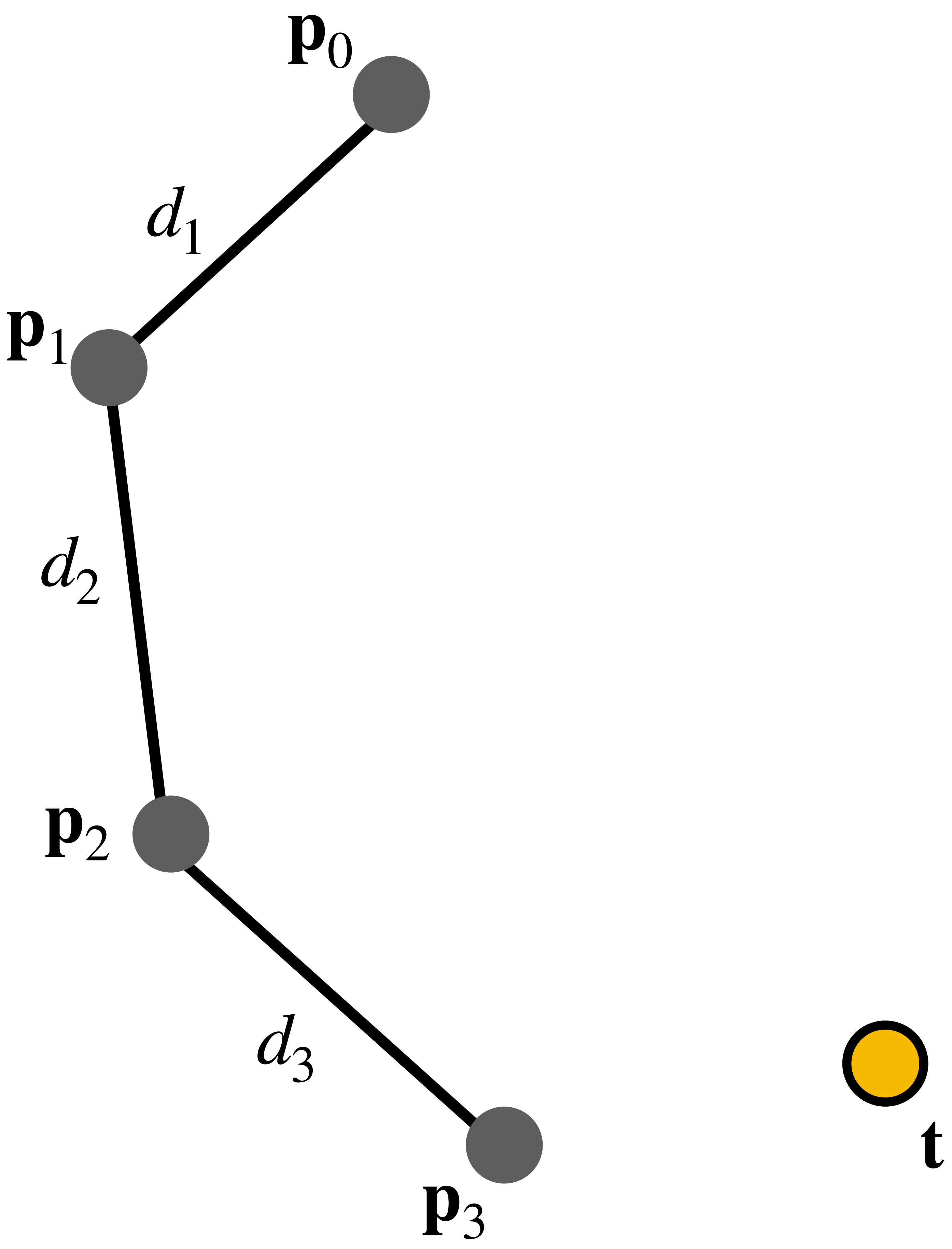
FABRIK: A fast, iterative solver for the Inverse Kinematics problem ☆

Andreas Aristidou *, Joan Lasenby

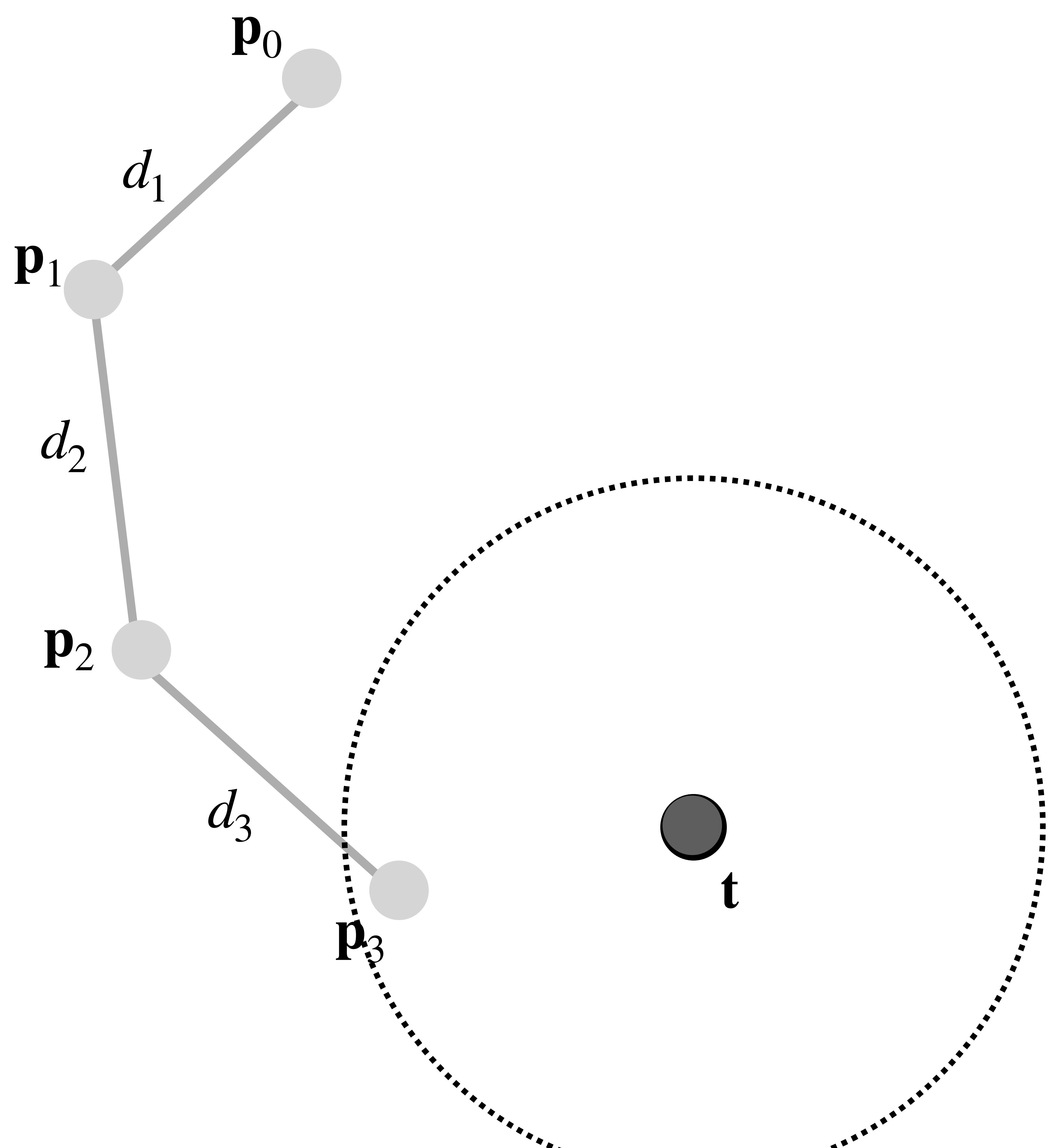
Department of Engineering, University of Cambridge, Cambridge CB2 1PZ, UK

- **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.**

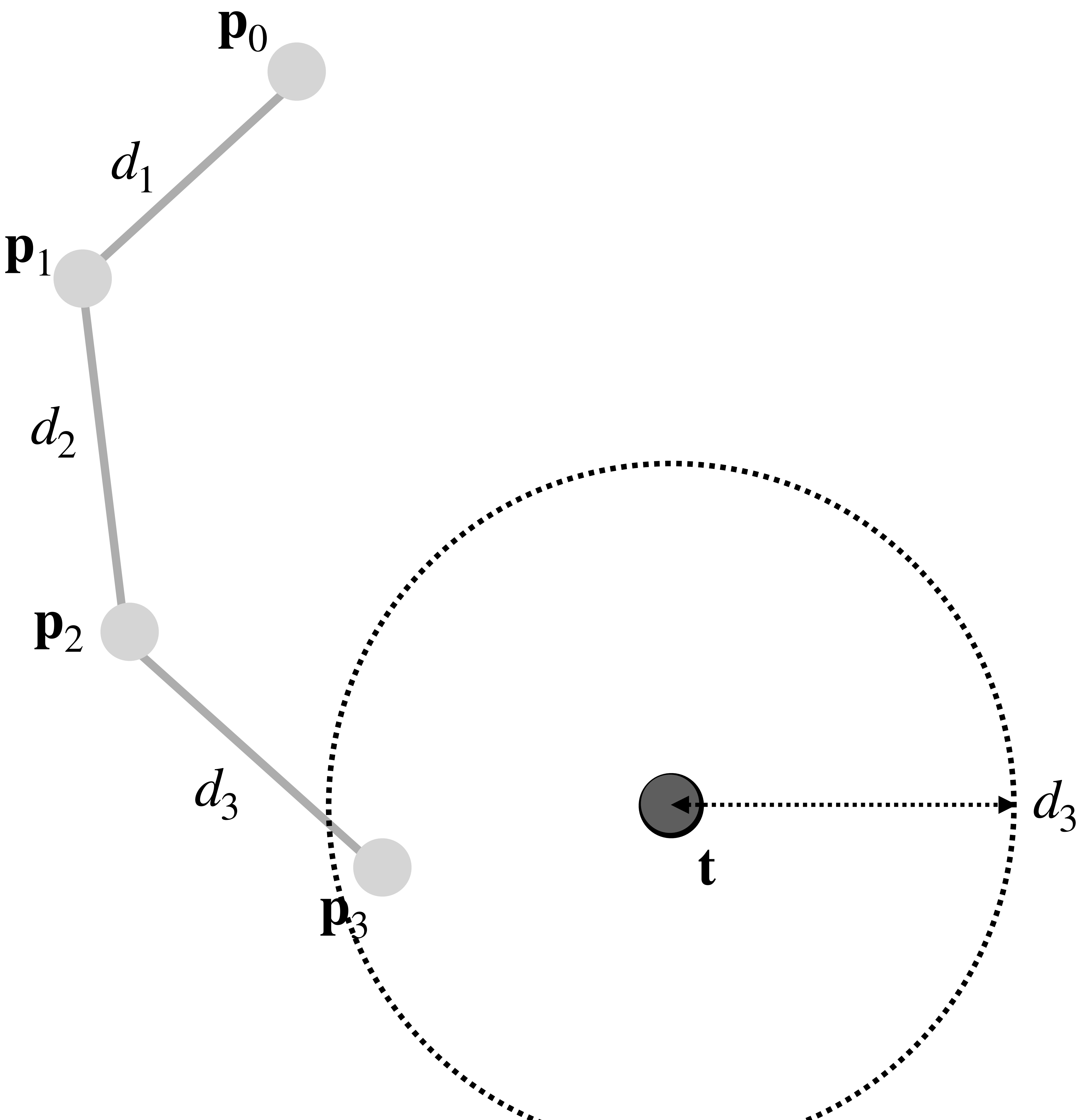
Basic algorithm



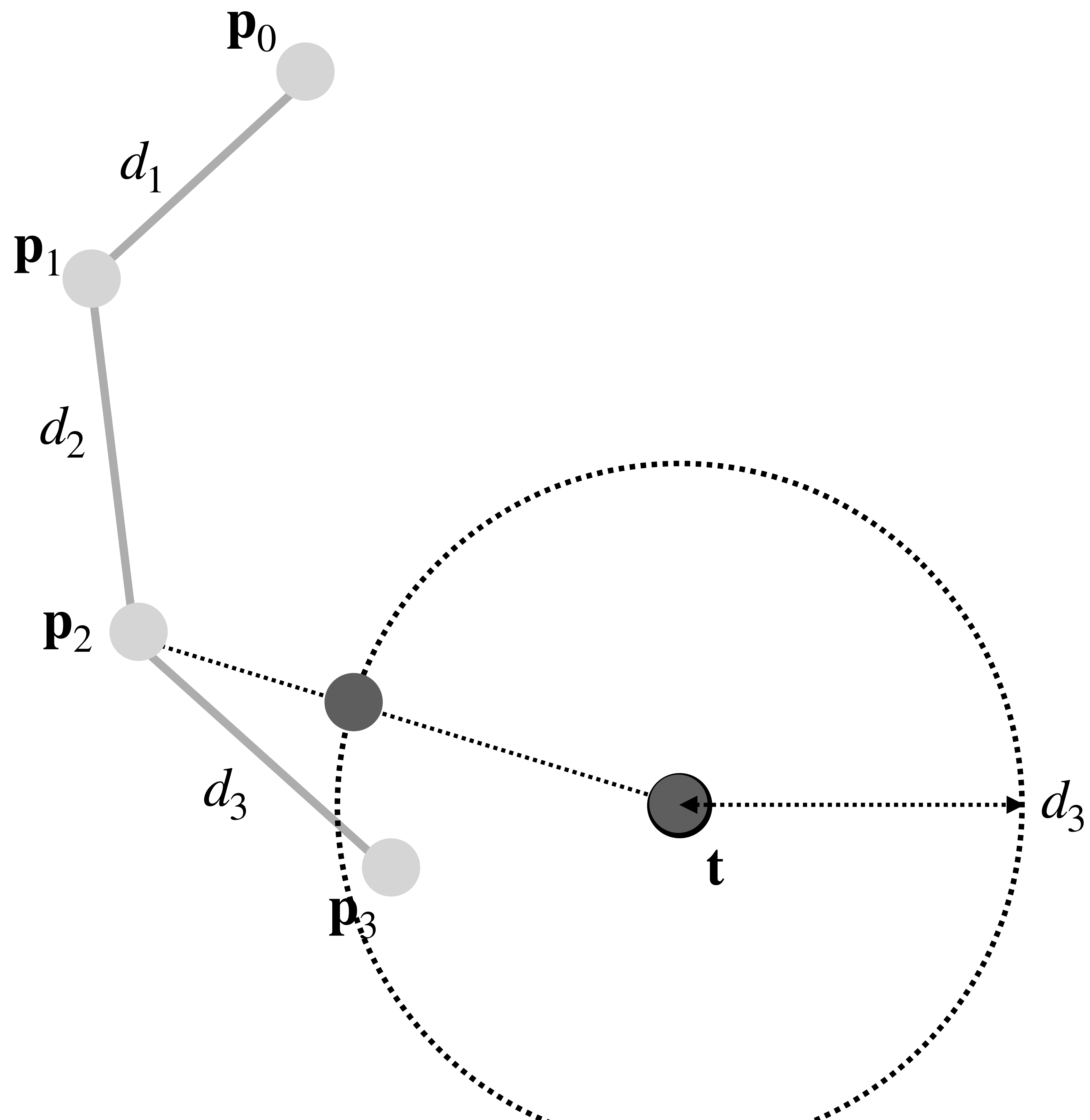
Basic algorithm



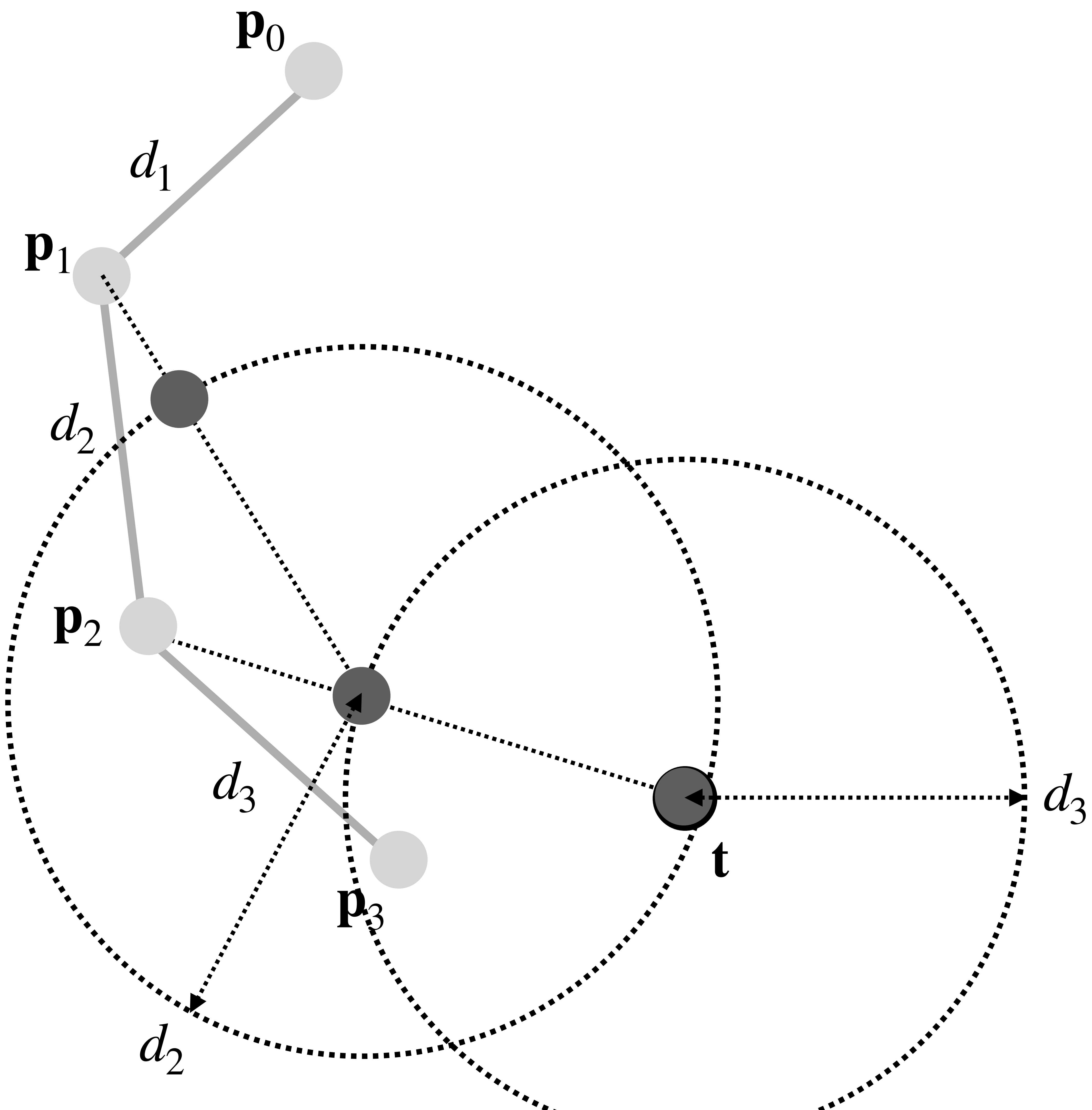
Basic algorithm



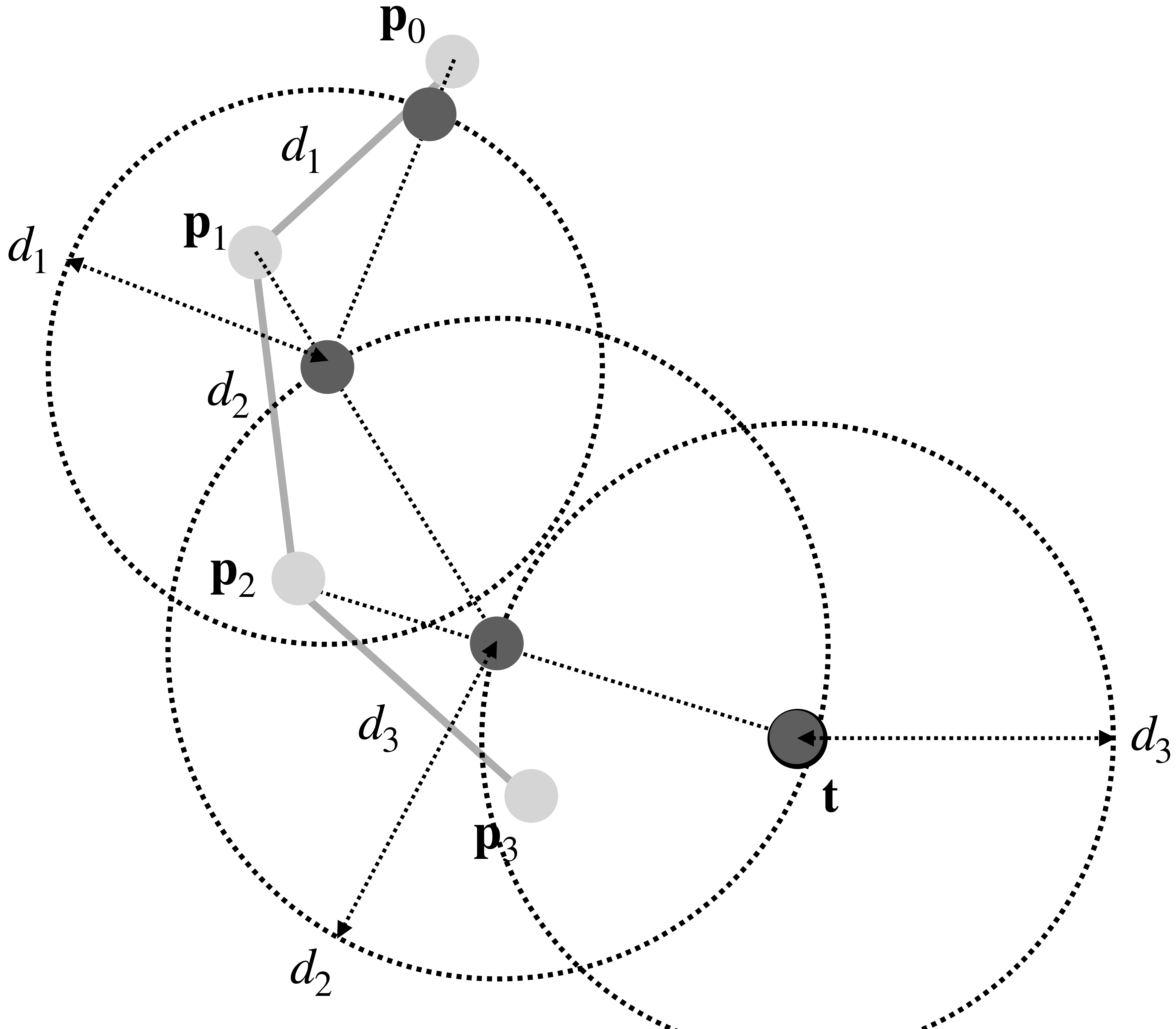
Basic algorithm



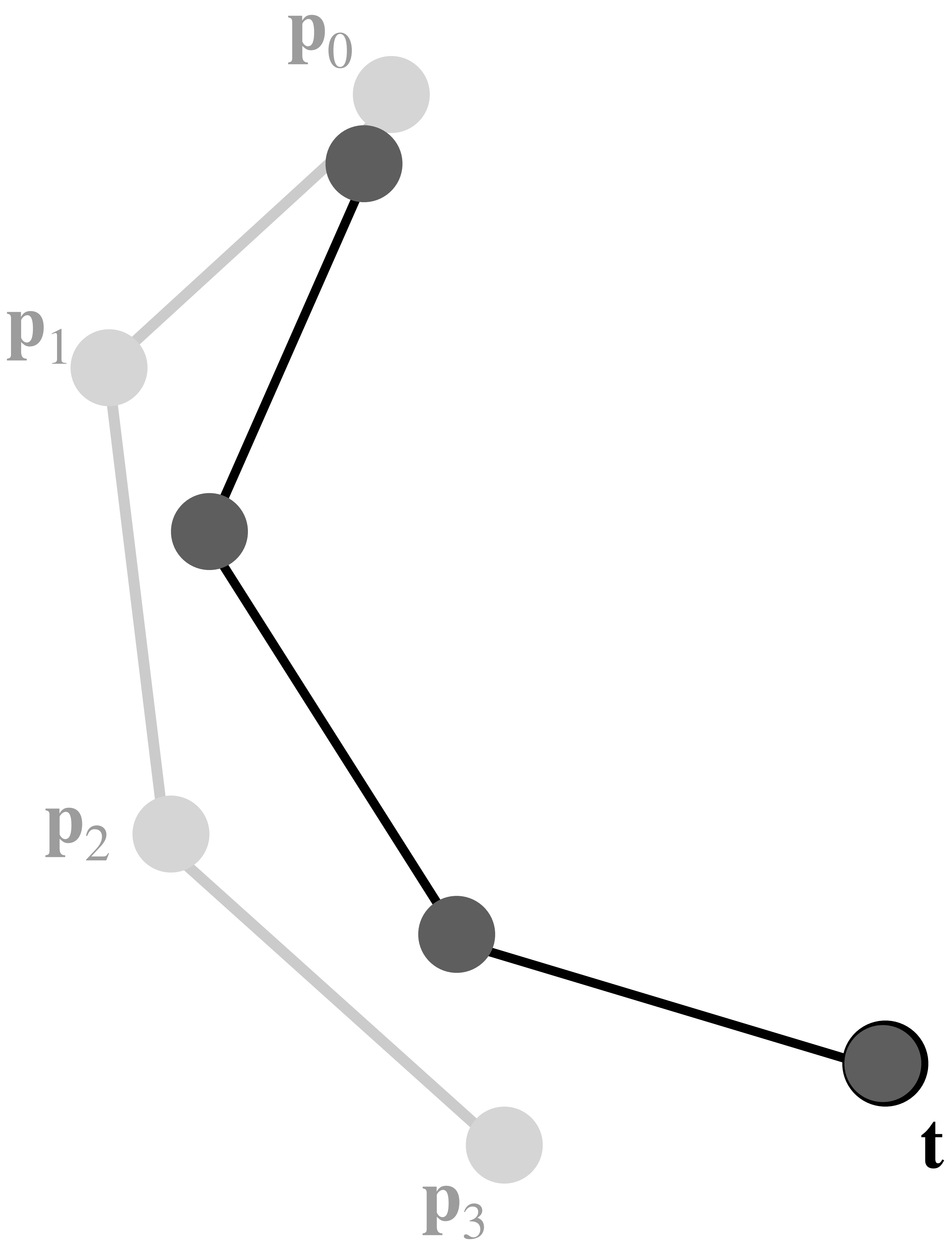
Basic algorithm



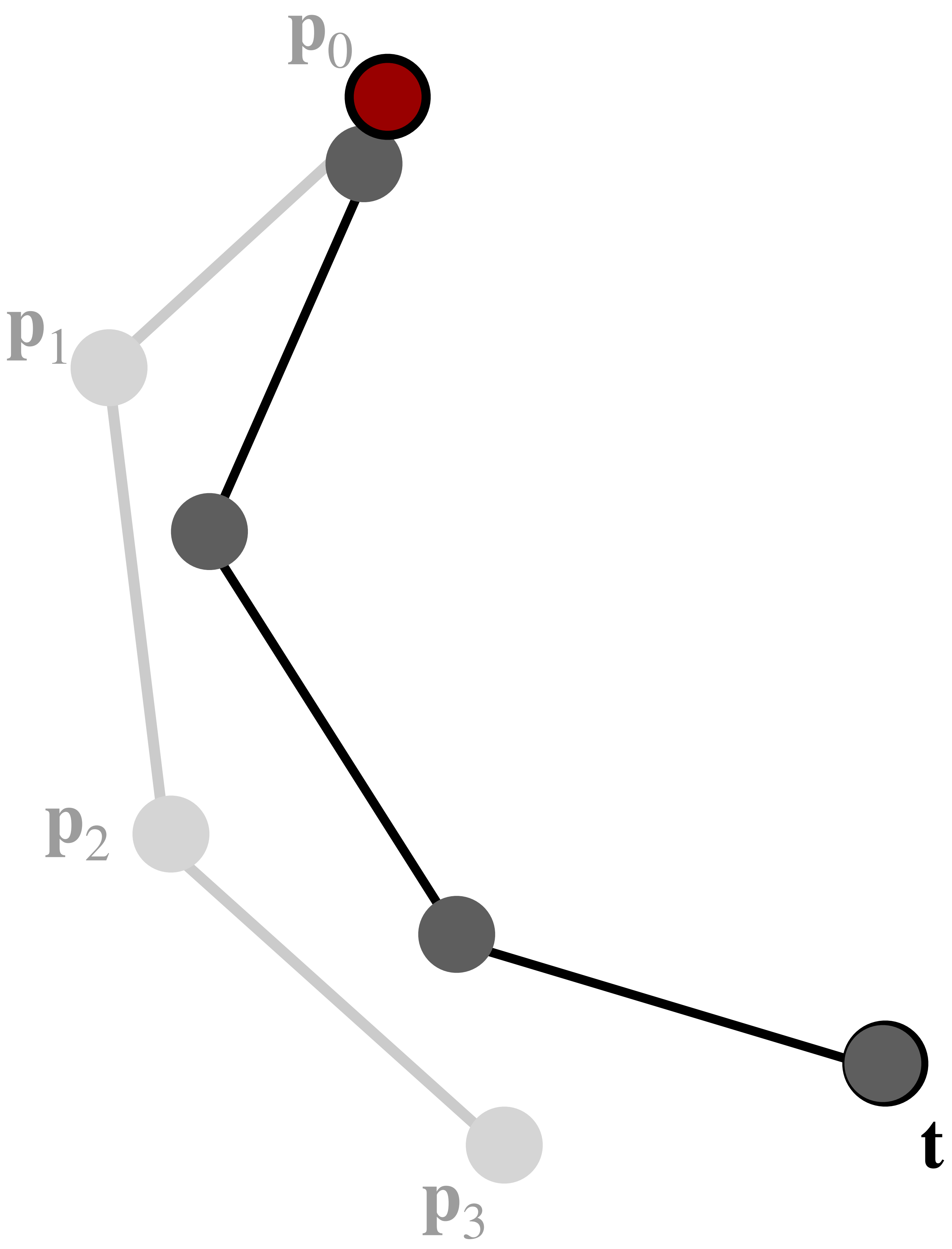
Basic algorithm



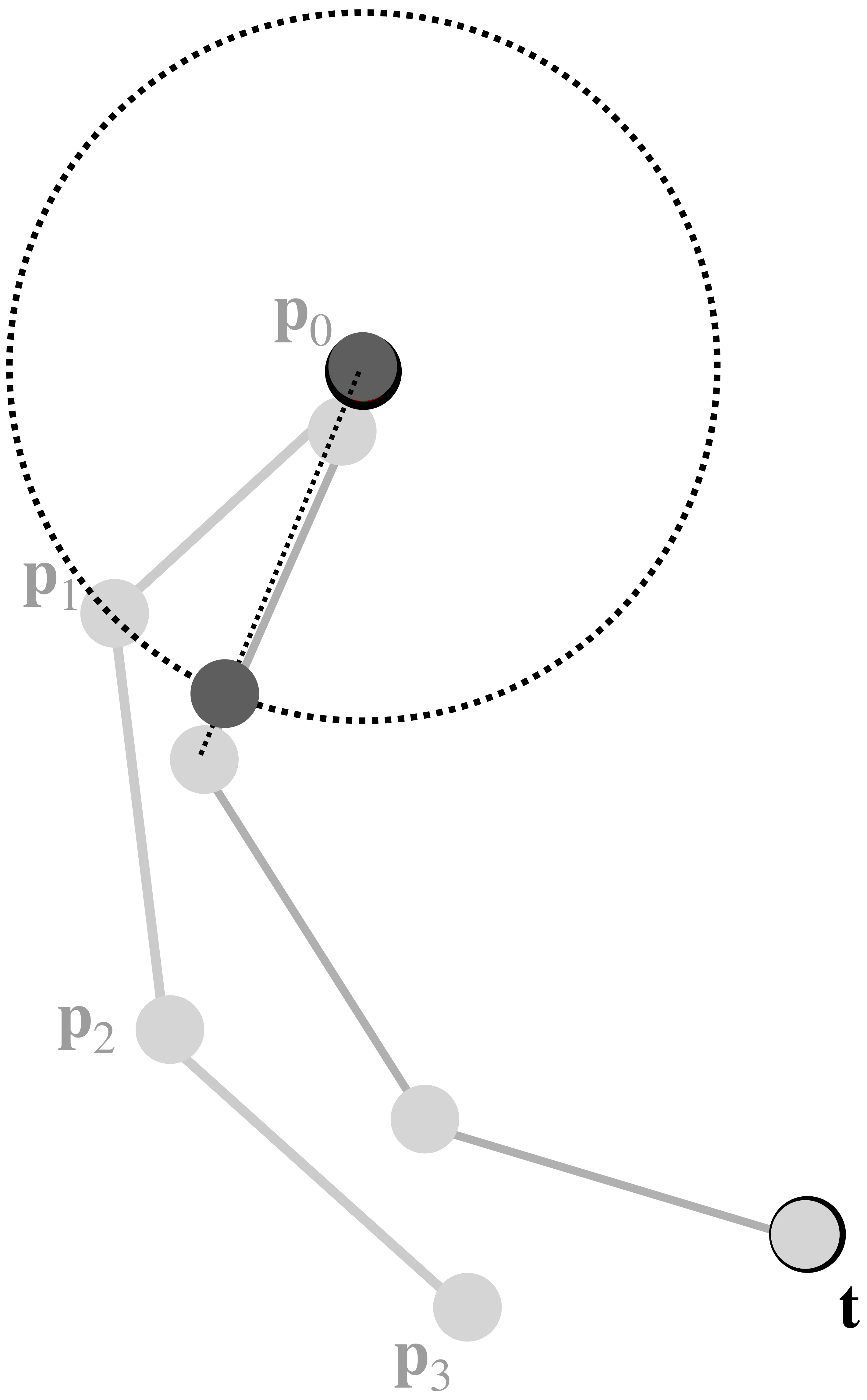
Basic algorithm



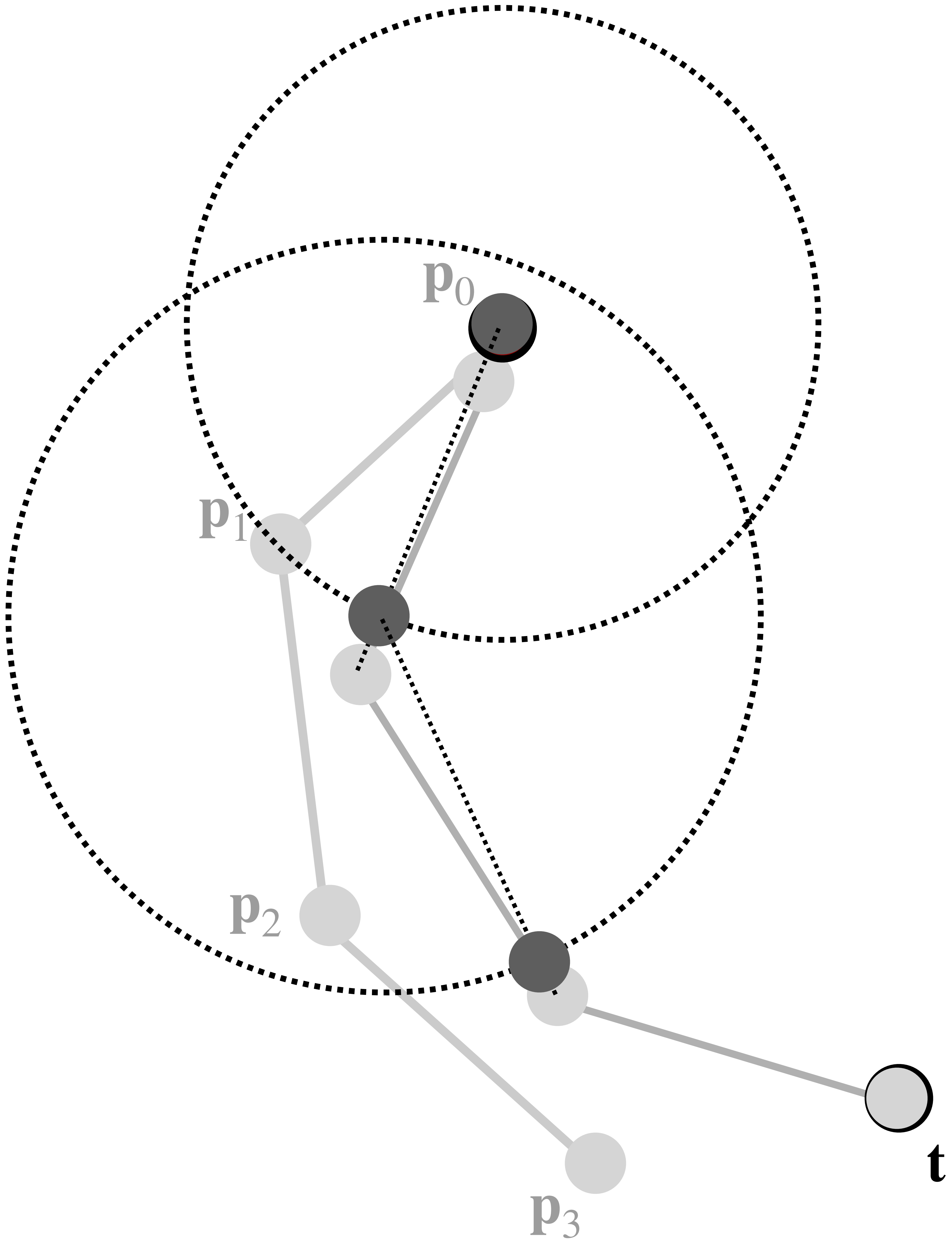
Basic algorithm



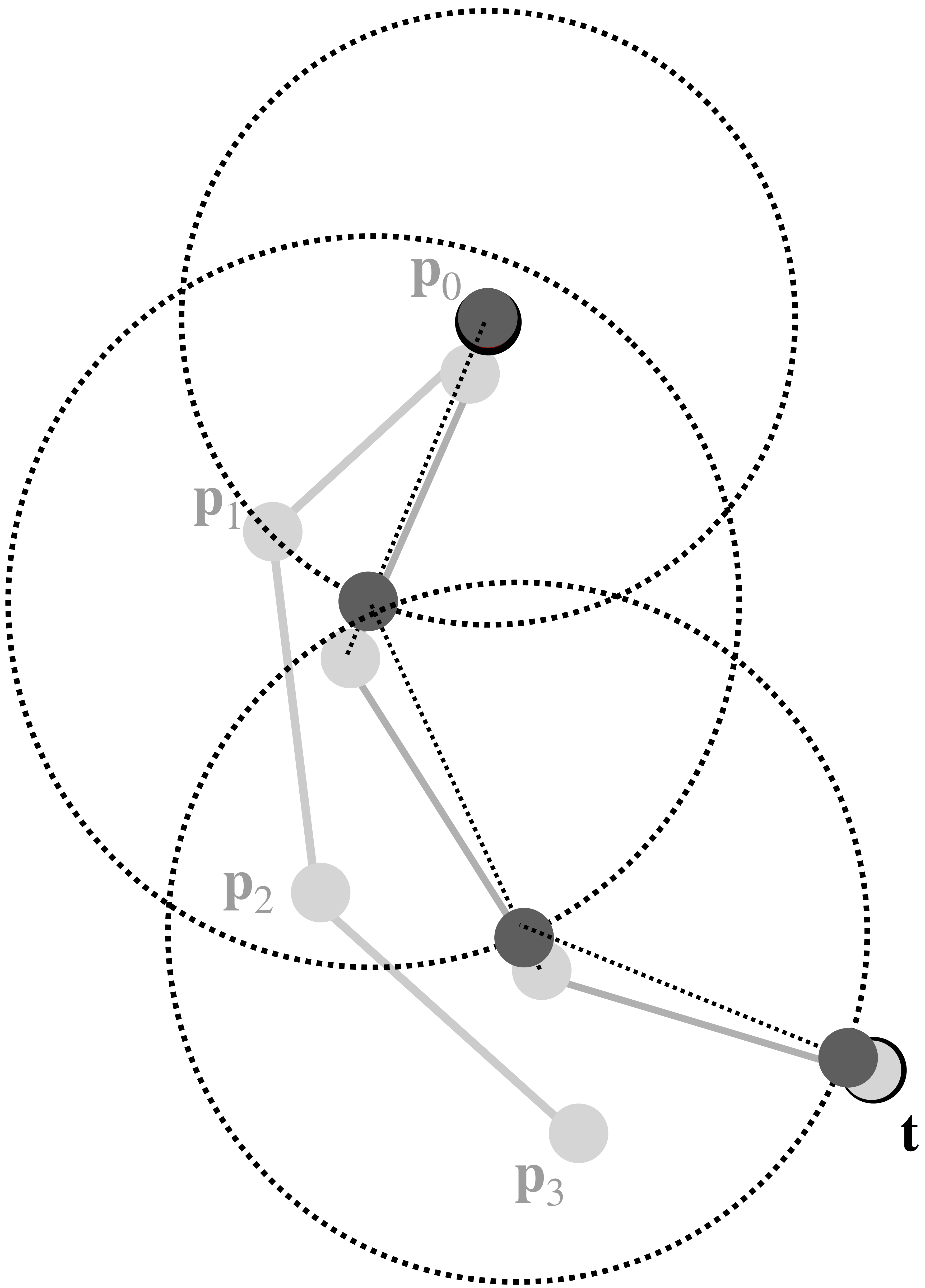
Basic algorithm



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
- **FABRIK:** http://andreasaristidou.com/publications/papers/FABRIK.pdf

Project 3: SUPER HOT IK

- Build an interactive IK solver to manipulate the pose of the character using a mouse interface.
- Create a game similar to SUPER HOT using your IK solver.
- Program in Python using Jupyter Notebook
- Use PyBullet (no pun intended) physics engine to animate the character and the bullet.