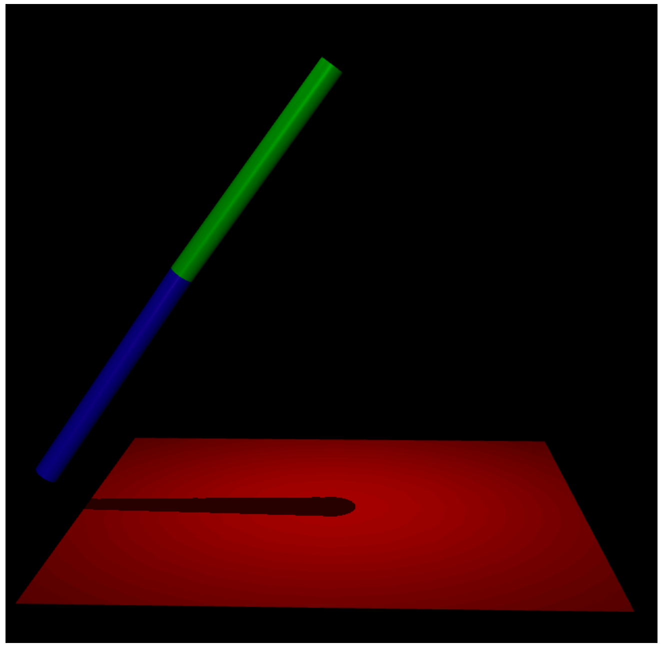
MuJoCo: finite state machine/trajectory tracking

Using template_writeData2.zip to get started

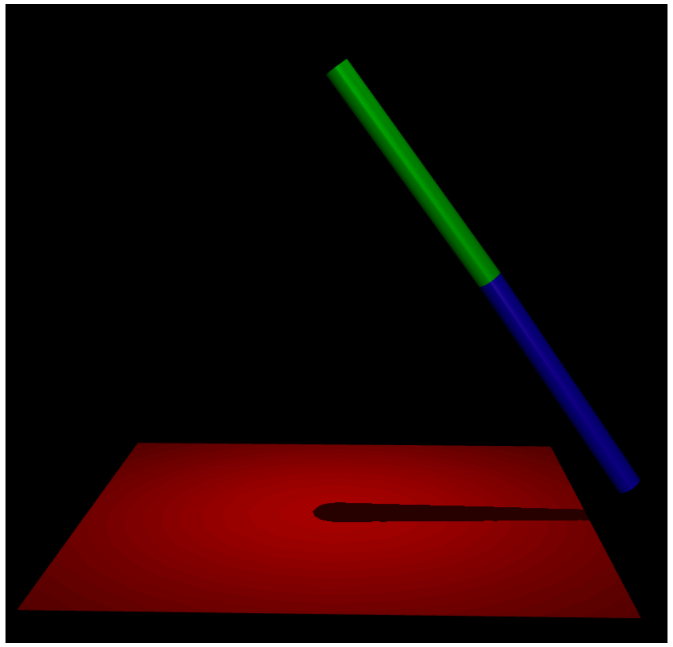
1. From tiny.cc/mujoco download template_writeData2.zip and unzip in myproject
2. Rename folder template to dbpendulum_fsm
3. Make these three changes
    1. main.c — line 28, change template_writeData2/ to dbpendulum_fsm/
    2. makefile — change ROOT = template_writeData to ROOT = dbpendulum_fsm also UNCOMMENT (del #) appropriate to your OS
    3. run_unix / run_win.bat change <template_writeData2> to <dbpendulum_fsm>
4. In the *shell, navigate to dbpendulum_fsm and type ./run_unix (unix) or run_win (windows); *shell = terminal for mac/linux / x64 for win

MuJoCo: Problem Statement (1)

Simulate the swinging of a robotic leg



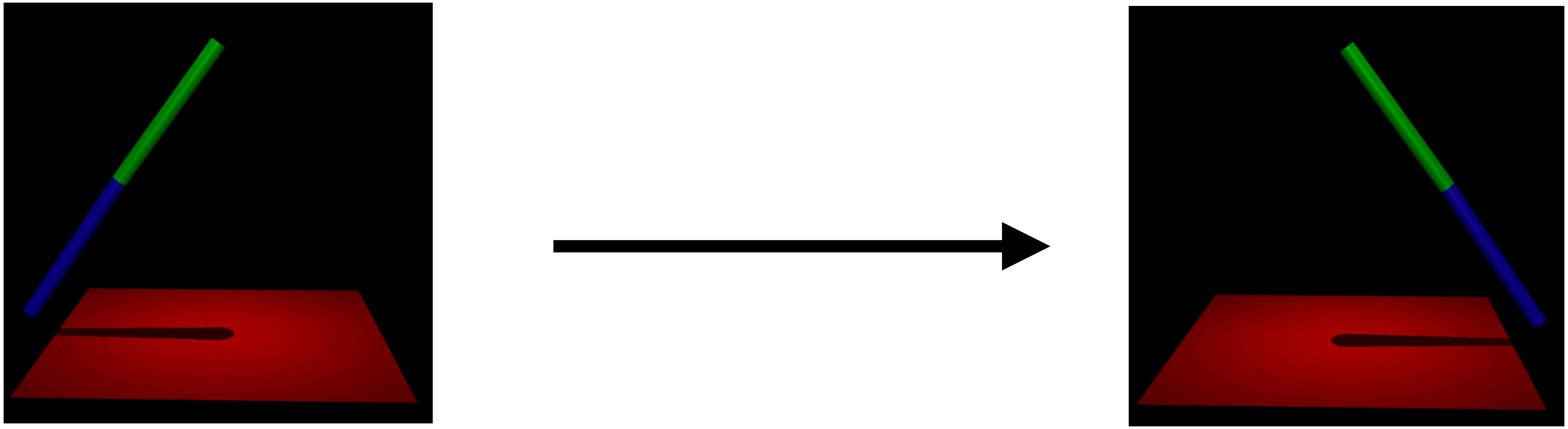
Start



End

MuJoCo: Problem Statement (2)

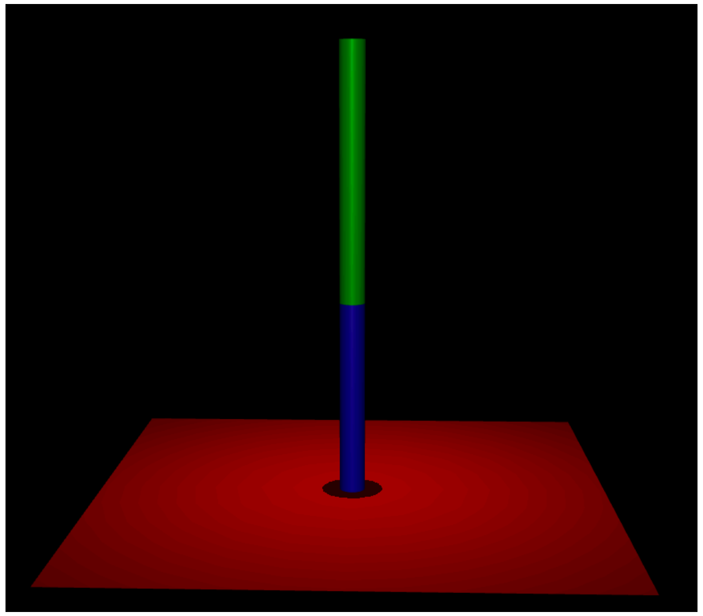
Simulate the swinging of a robotic leg



Start

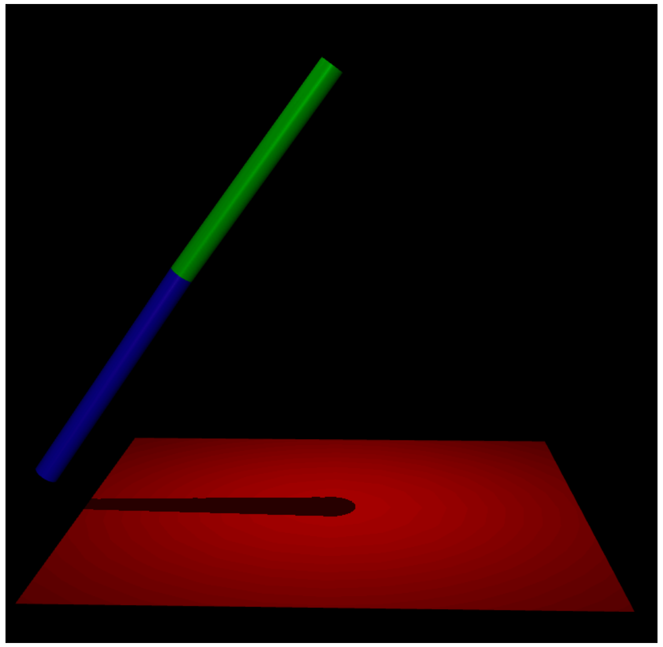
End

Issue: Foot collision

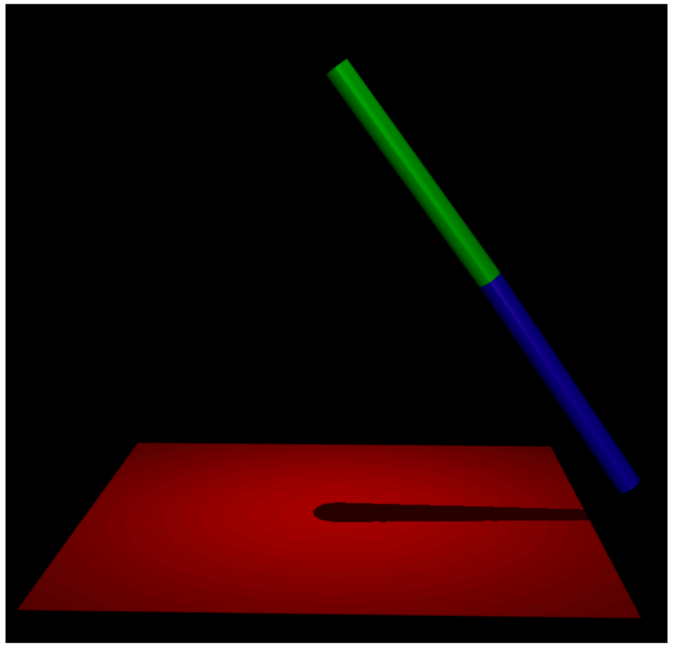


MuJoCo: Problem Solution (3)

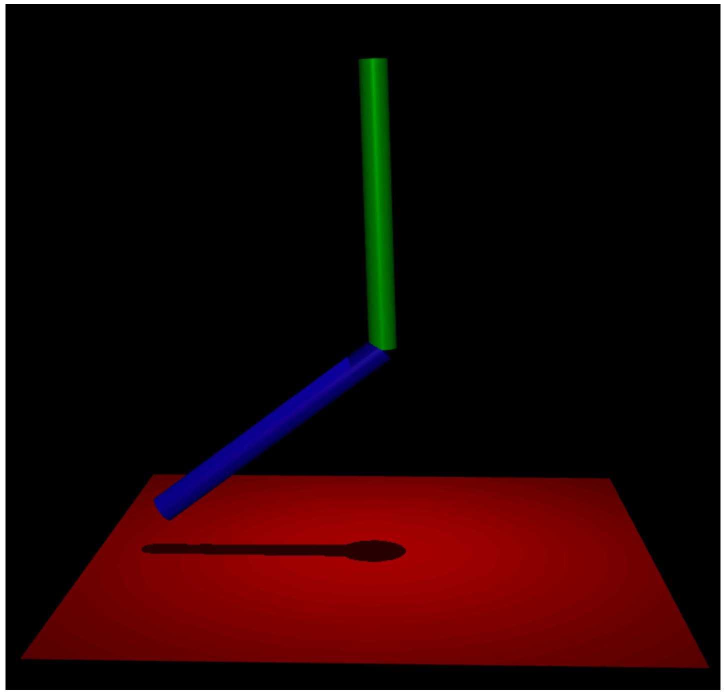
Simulate the swinging of a robotic leg



Start



End

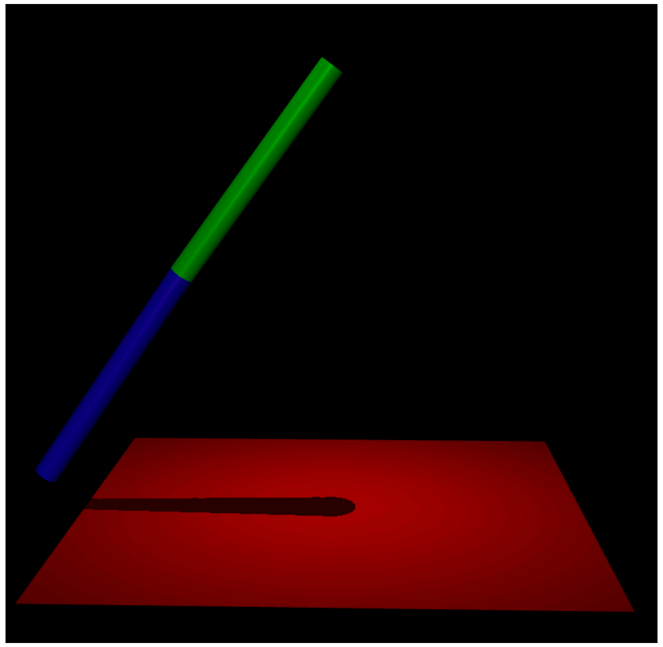


Intermediate

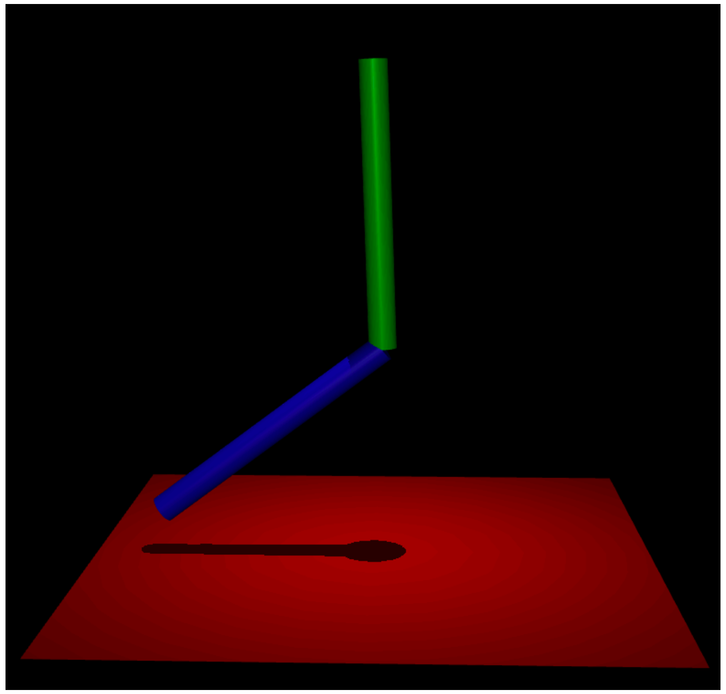
Bend the knees

MuJoCo: Problem Solution (3)

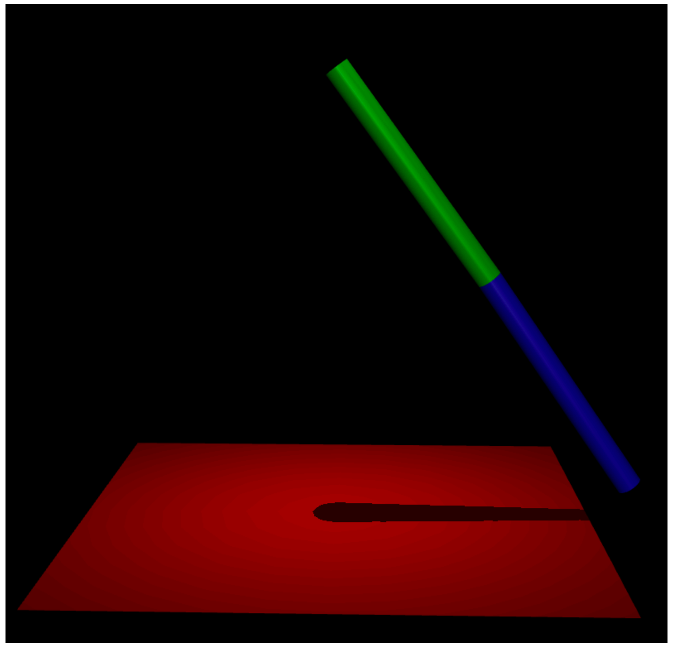
Simulate the swinging of a robotic leg



Start



Intermediate

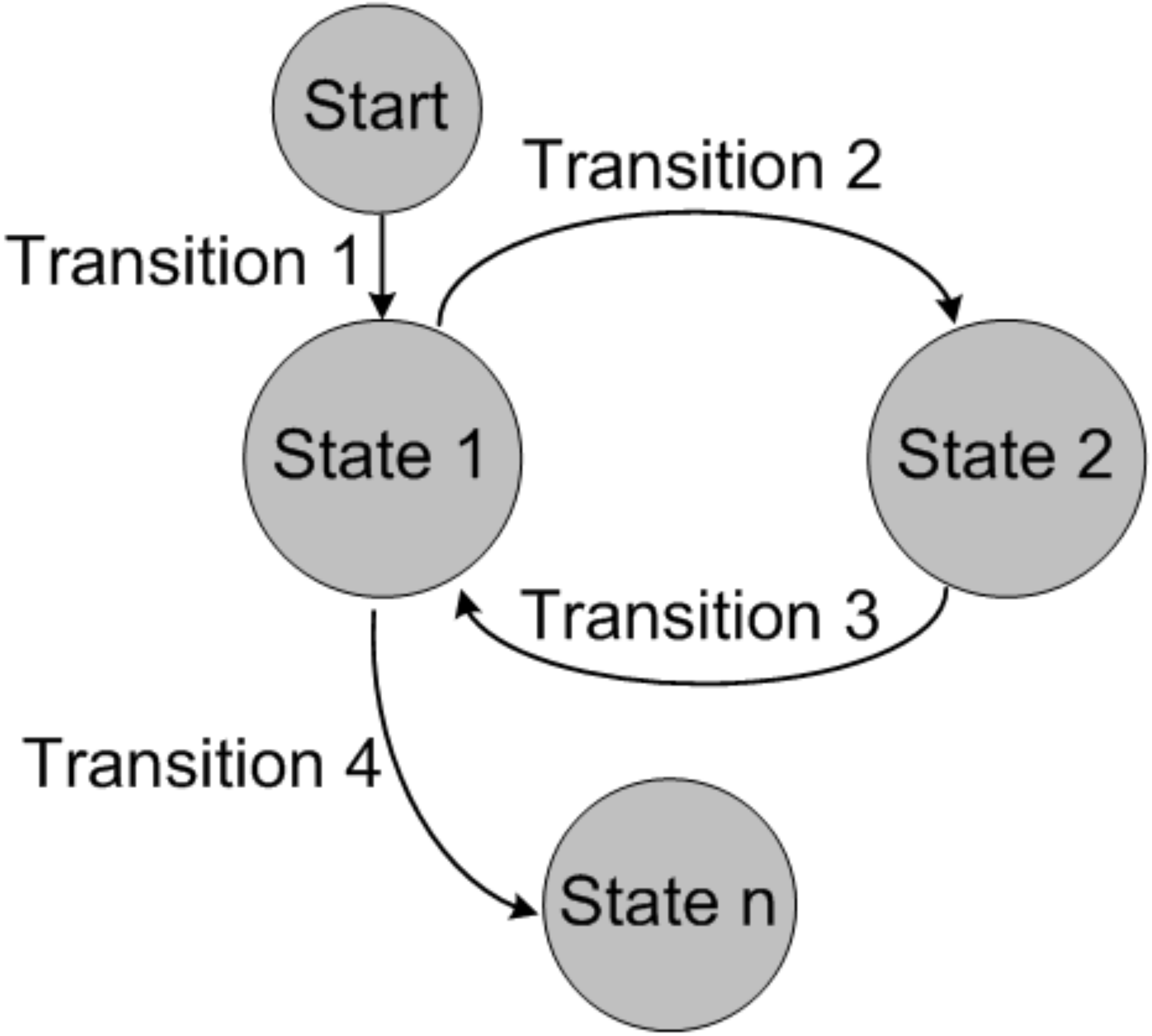


End

Let's use ./simulate to choose these angles

Bend the knees

MuJoCo: Finite State Machine (FSM)

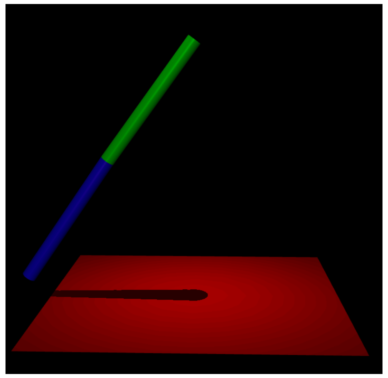


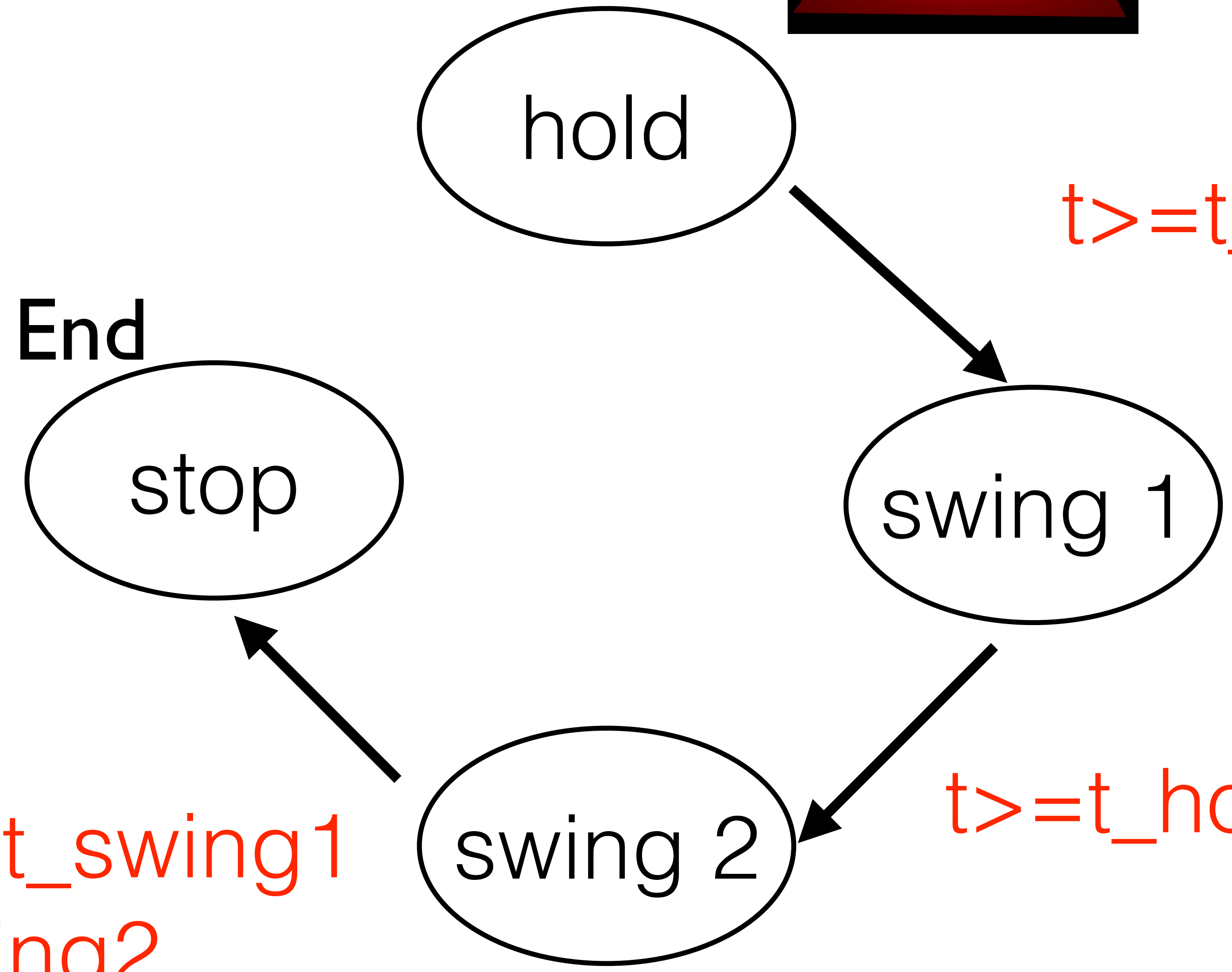
State: Do an action
e.g., move, grasp

Transition: Switching condition.
e.g., t > 4 sec,
reached a position

https://en.wikipedia.org/wiki/Finite-state_machine

MuJoCo: Problem solution (4)





Start

hold

t>=t_hold

swing 1

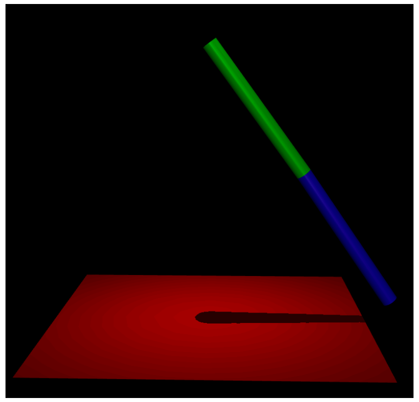
t>=t_hold+t_swing1

swing 2

t>=t_hold+t_swing1
+t_swing2

stop

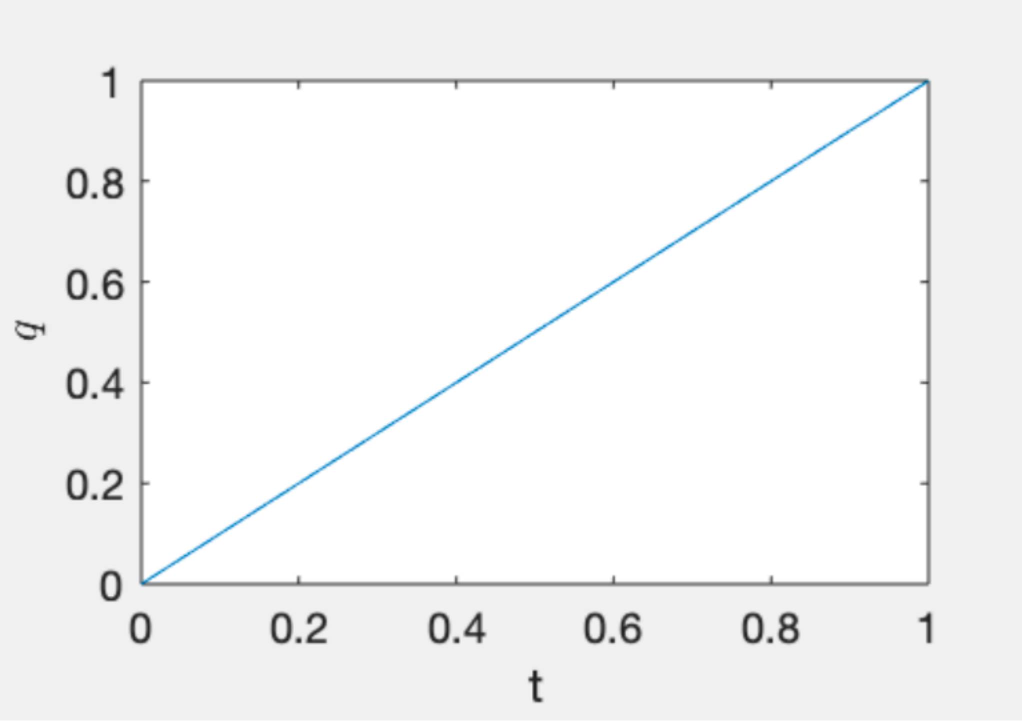
End



MuJoCo: Trajectory generation (1)

Generate a trajectory q(t) and track the trajectory (PD control)

Linear Trajectory



$$q(t) = a_0 + a_1 t$$

a0 and a1 are constants

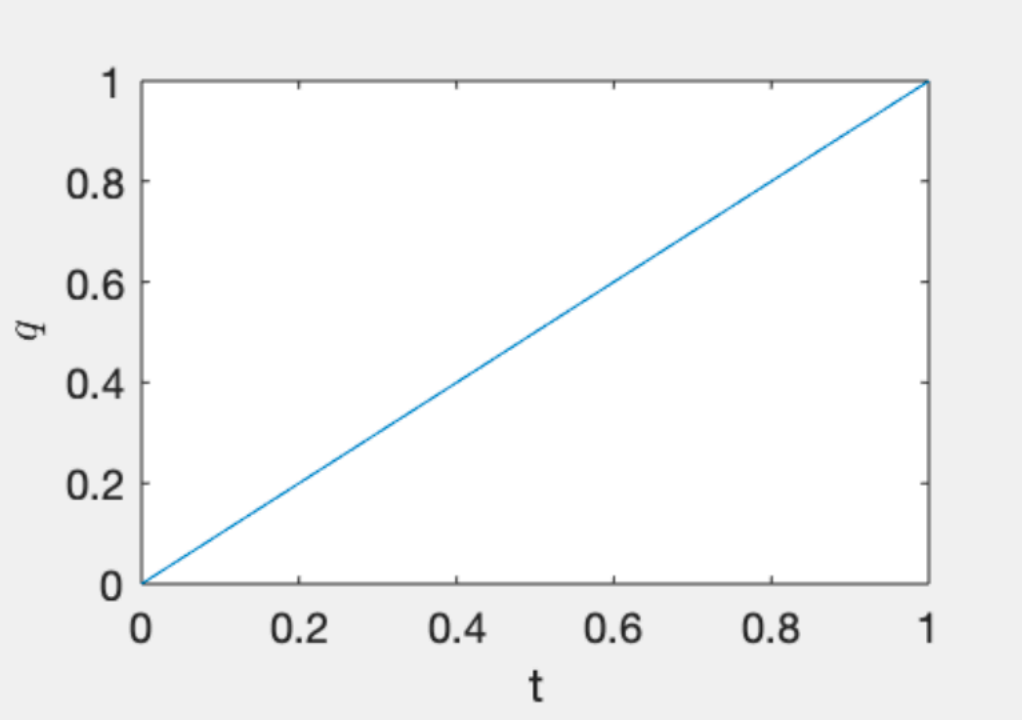
MuJoCo: Trajectory generation (2)

Generate a trajectory q(t) and track the trajectory (PD control)

Boundary conditions (2)

$$q_0 = a_0 + a_1 t_0$$

$$q_f = a_0 + a_1 t_f$$



Solver for a0 and a1

$$q(t) = \left(\frac{q_0 t_f - q_f t_0}{t_f - t_0} \right) + \left(\frac{q_f - q_0}{t_f - t_0} \right) t$$

NOTE:

$$\dot{q}(t) = \left(\frac{q_f - q_0}{t_f - t_0} \right) = \text{constant}$$

MuJoCo: Trajectory generation (3)

Generate a trajectory q(t) and track the trajectory (PD control)

Cubic Trajectory

$$q(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3$$

a0, a1, a2, a3 are constants

Boundary conditions (4)

$$q(t=0) = q_0, \quad q(t=t_f) = q_f, \quad \dot{q}(t=0) = 0, \quad \dot{q}(t=t_f) = 0$$

Solving for a's

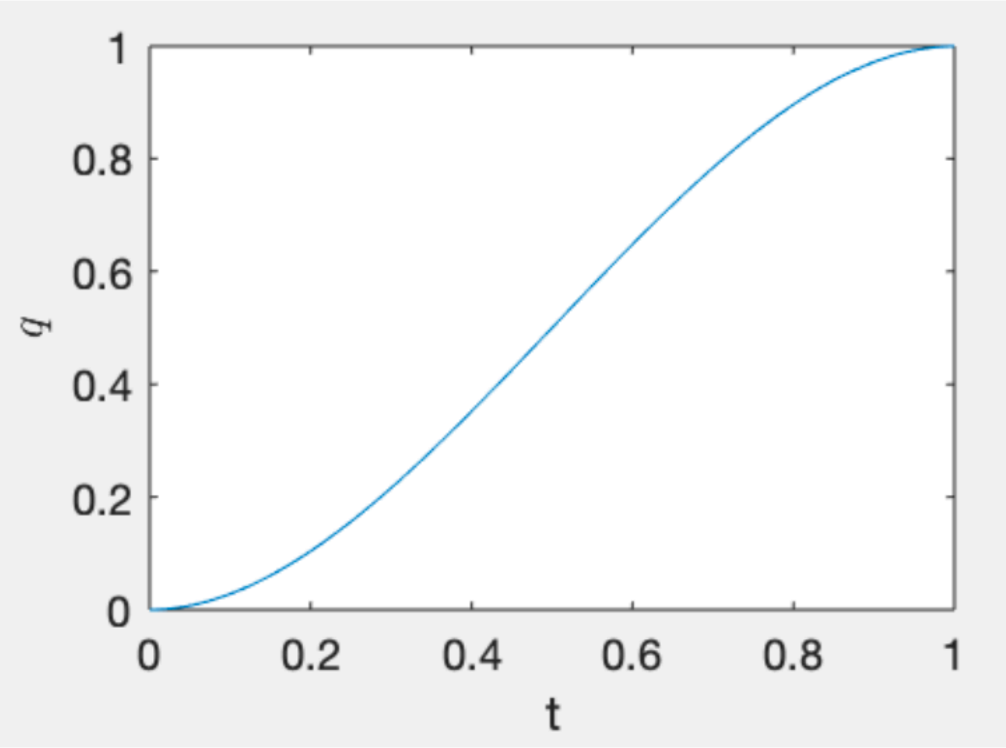
$$\begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} = \frac{1}{(t_f - t_0)^3} \begin{bmatrix} q_f t_0^2 (3t_f - t_0) + q_0 t_f^2 (t_f - 3t_0) \\ 6 t_0 t_f (q_0 - q_f) \\ 3(t_0 + t_f)(q_f - q_0) \\ 2(q_0 - q_f) \end{bmatrix}$$

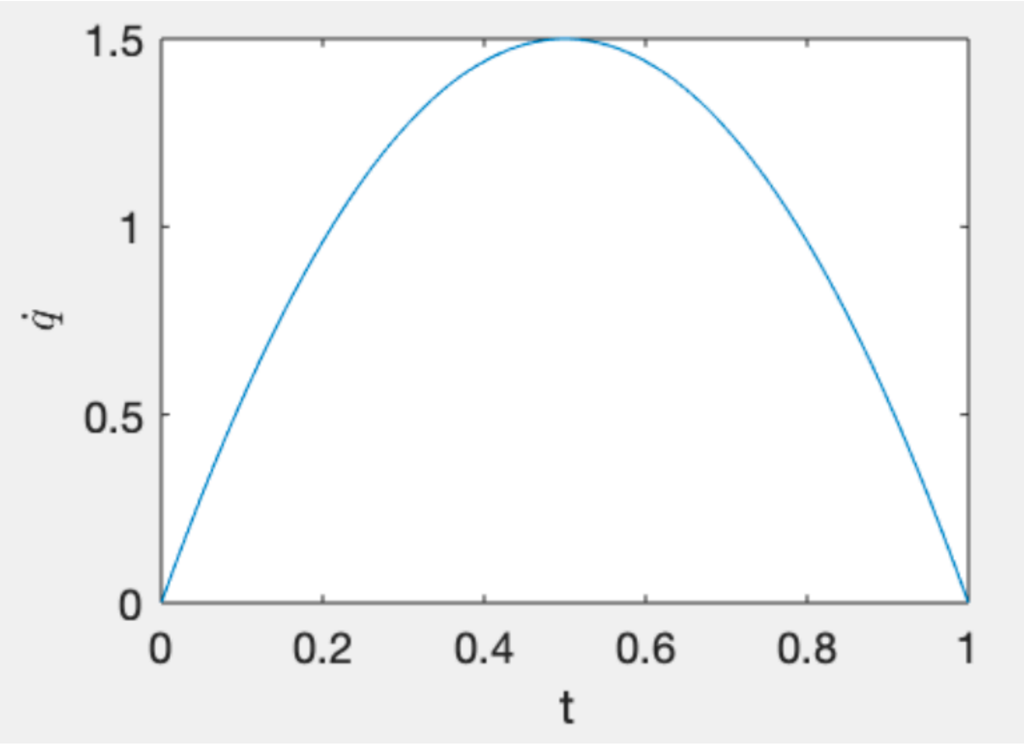
MuJoCo: Trajectory generation (4)

Generate a trajectory q(t) and track the trajectory (PD control)

Cubic Trajectory

$$\begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} = \frac{1}{(t_f - t_0)^3} \begin{bmatrix} q_f t_0^2 (3t_f - t_0) + q_0 t_f^2 (t_f - 3t_0) \\ 6 t_0 t_f (q_0 - q_f) \\ 3(t_0 + t_f)(q_f - q_0) \\ 2(q_0 - q_f) \end{bmatrix}$$


q
1
0.8
0.6
0.4
0.2
0
0
0.2
0.4
0.6
0.8
1
t


$\dot{q}$
t