Lecture 13
Trajectory Generation

Prof. Katie DC

March 21, 2023

Modern Robotics Ch 9

Administrivia

- HW7 due Friday
- Review lecture on Thursday
- Exam 2 from 3/27-3/29
  - Remember to sign-up on the CBTF website!
  - No class on Tuesday
  - No office hours
- Guest Lecture 2 on 3/30
  - Reflection due following Friday

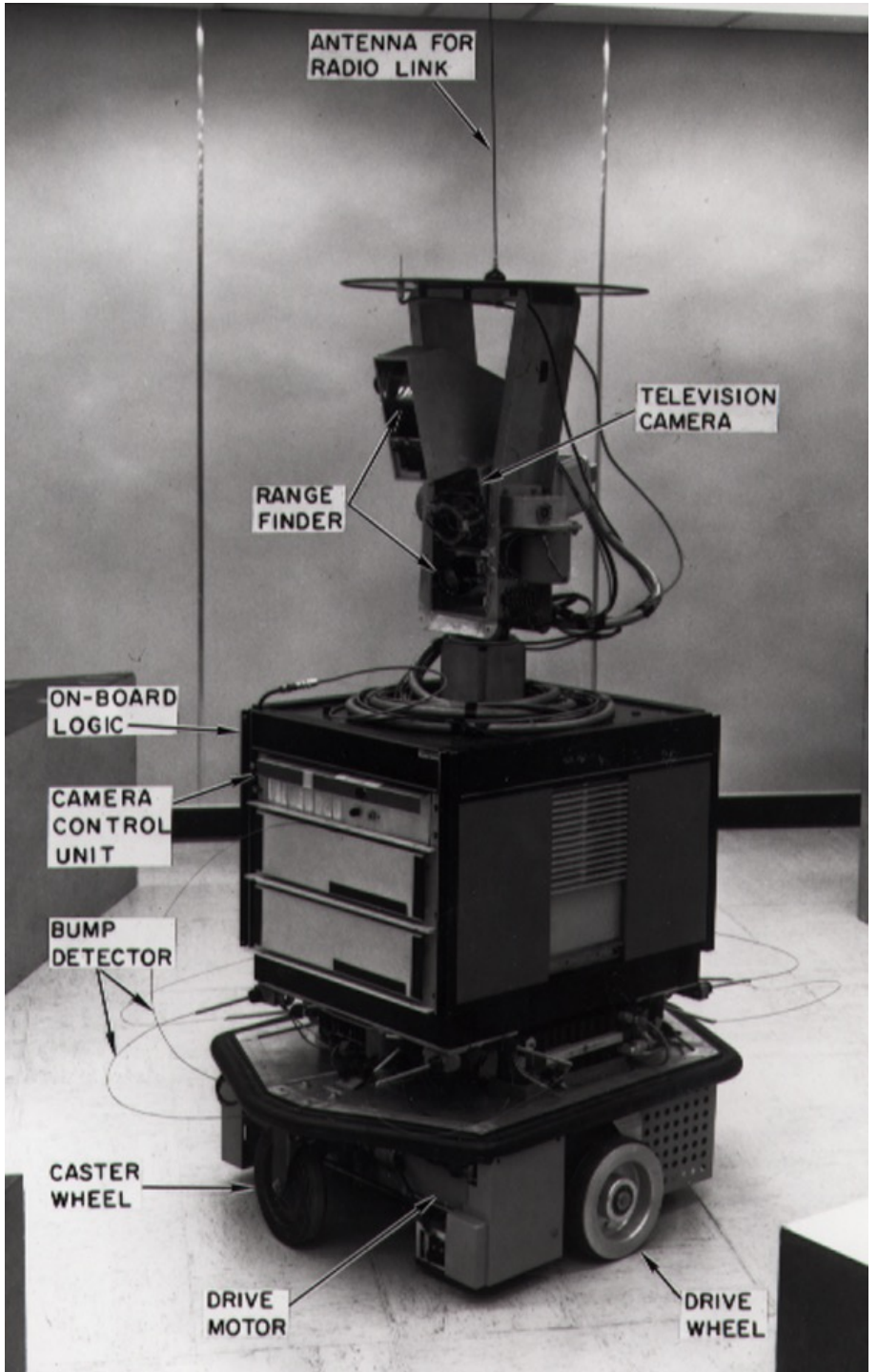
Historical Fun Fact

Meet Shakey the Robot:

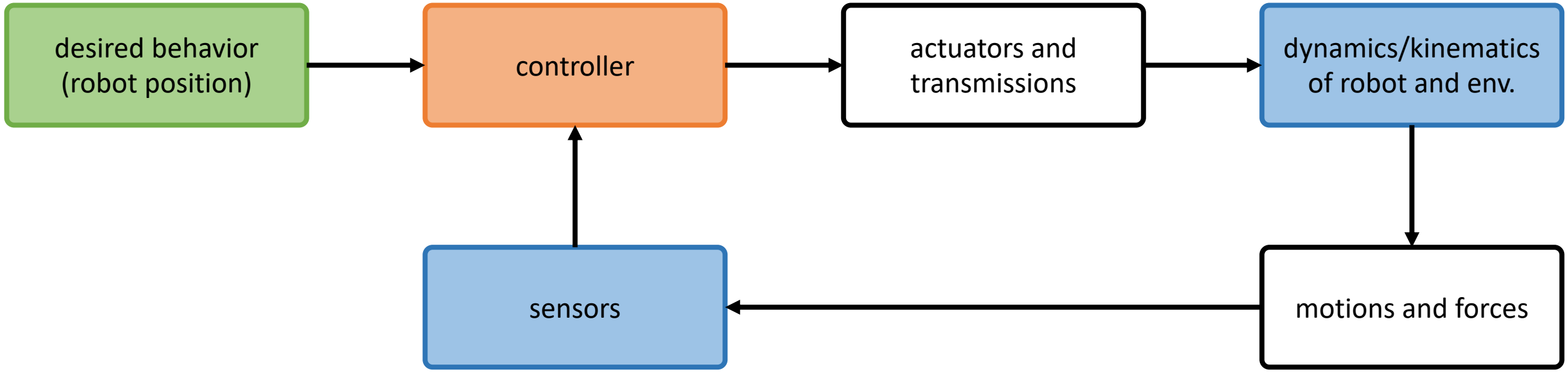
An Experiment in Robot Planning and Learning

Developed by Stanford Research Institute (SRI) (1966 – 1972)

1. An operator types the command "push the block off the platform" at a computer console.
2. Shakey looks around, identifies a platform with a block on it, and locates a ramp in order to reach the platform.
3. Shakey then pushes the ramp over to the platform, rolls up the ramp onto the platform, and pushes the block off the platform.
4. Mission accomplished.



Control Paradigm



Trajectories and Paths

- The specification of a robot state as a function of time is called a **trajectory**

- Using forward kinematic maps, we can obtain the position of each link given as joint angles
  - The trajectory of the end-effector is then $T_{sb}\big(\theta(t)\big)$
- A **path** is a set of points

Normalized Trajectories

- **Path** $\theta(s)$ maps a scalar path parameter $s \in [0,1]$ to a point in the robot's configuration space

- A **time-scaling** $s(t)$ is a monotonically increasing function:

Straight-Line Paths

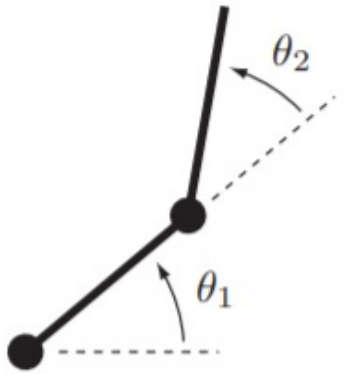
- Given θ_0 and θ_1, find straight-line path:

Straight-Line Paths

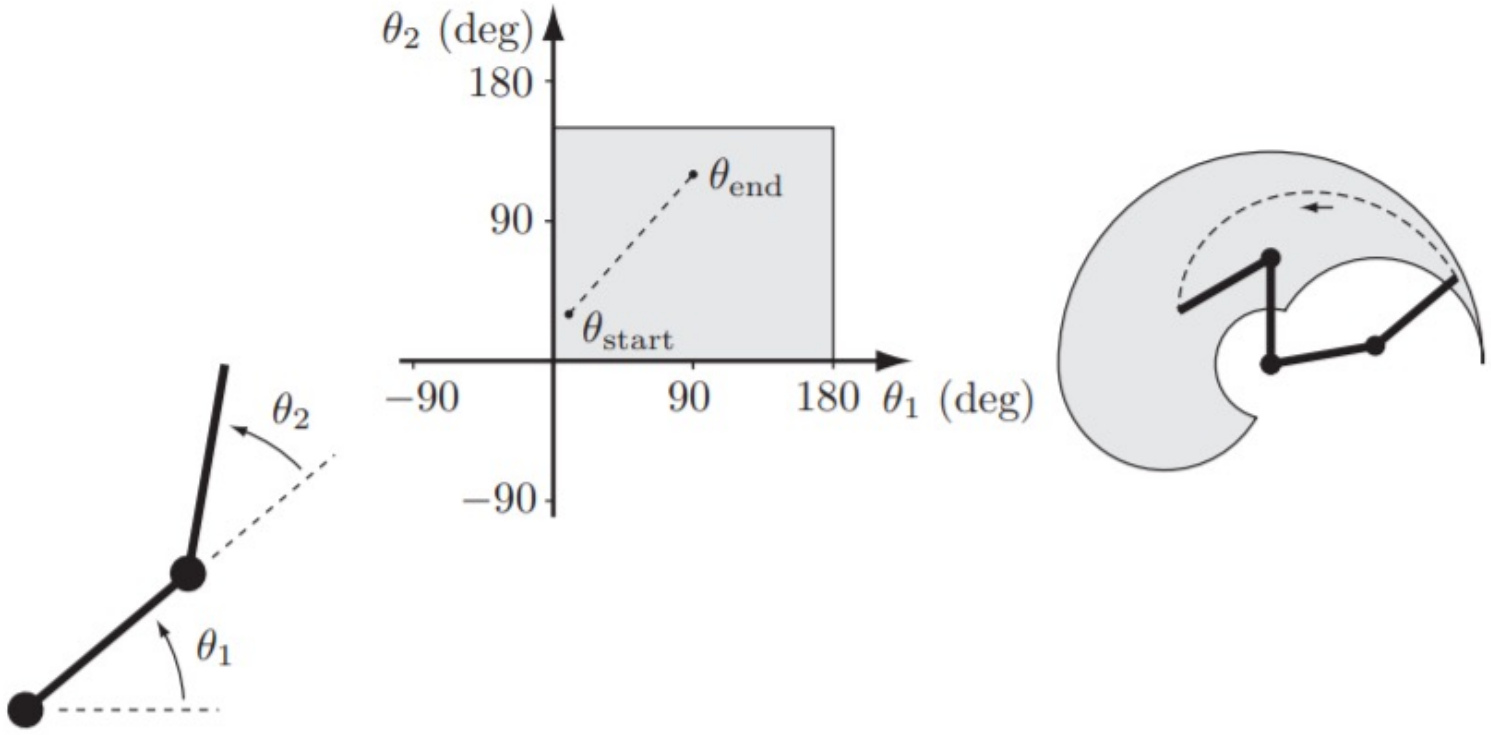
- Given θ_0 and θ_1, find straight-line path:

- Is this in the task or configuration space?
  - Straight lines in joint space do not lead to straight lines in end-effector/task space
- Straight line in task space:

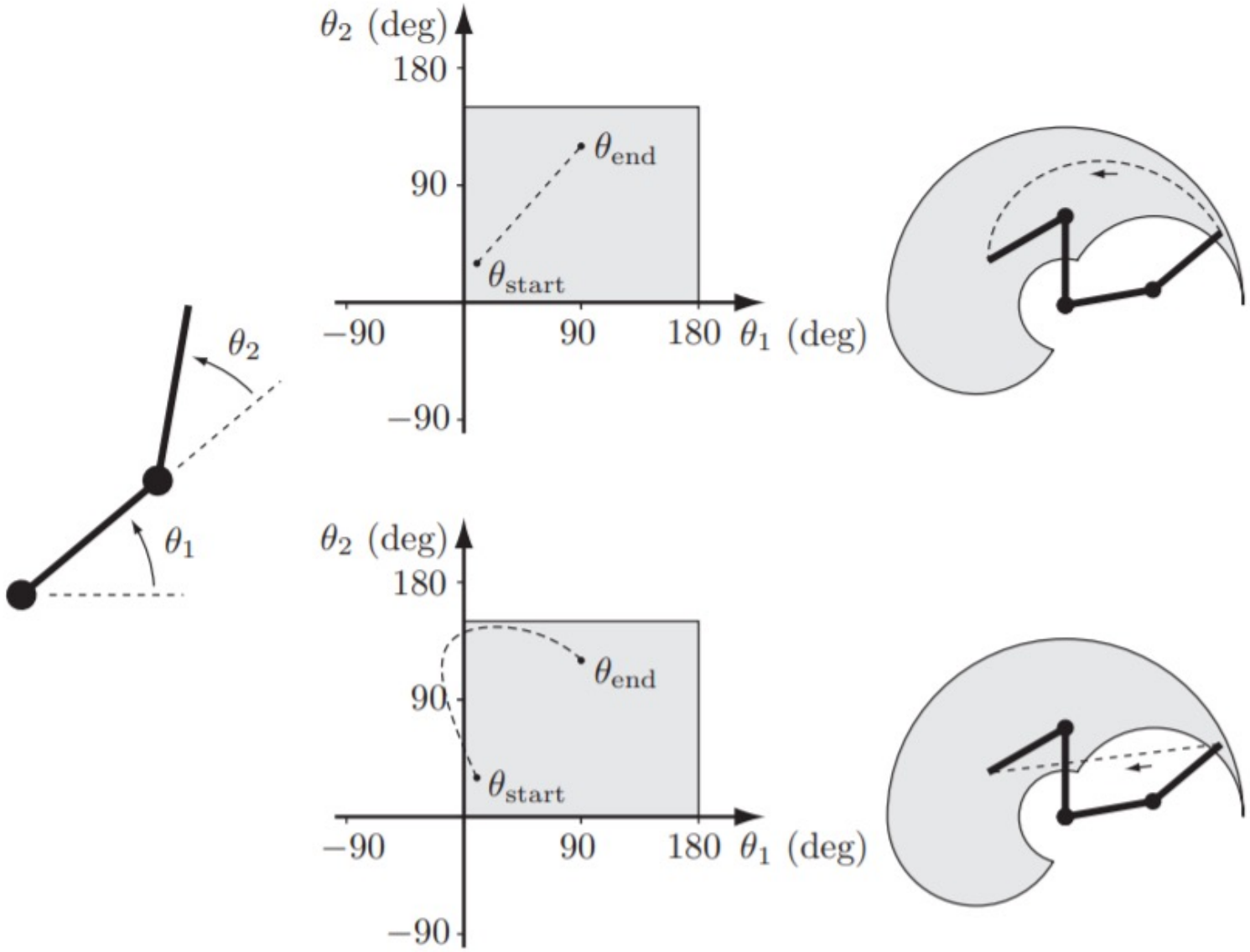
Straight-line Paths



Straight-line Paths



Straight-line Paths



Straight lines in SE(3)

In $\mathbb{R}^2$, straight lines are characterized by a constant velocity

Straight lines in SE(3)

- We can decouple rotation and translation:

Time-scaling of straight-line paths

- Time scaling ensures that the motion is smooth and constraints are met

Polynomial Time-Scaling (1)

Polynomial Time-Scaling (2)

Summary

- Defined **paths, time-scaling,** and **trajectories**
- Looked at how to find **straight-line paths** in various spaces
- We choose a **parametrization $\boldsymbol{s(t)}$**, and computed the resulting velocity and acceleration profiles of the trajectory
  - Using a third-order polynomial, we tuned their maximal values to meet requirements with one parameter T

Summary

- Defined **paths, time-scaling,** and **trajectories**
- Looked at how to find **straight-line paths** in various spaces
- We choose a **parametrization $s(t)$**, and computed the resulting velocity and acceleration profiles of the trajectory
  - Using a third-order polynomial, we tuned their maximal values to meet requirements with one parameter T
- We can follow the same procedure with different parametrizations for $s(t)$ (e.g., polynomials of order 5, trapezoidal functions, splines, etc.)
  - Having more parameters allows us to meet more constraints. For example, using a fifth order polynomial, we can ensure that $\ddot{\theta}(0) = \ddot{\theta}(T) = 0$, meaning no jerk at beginning and end of the motion
- **Next topics** are on different concepts of / approaches to planning when the path may not be given