CMPE 185 Autonomous Mobile Robots

Navigation and Control

Dr. Wencen Wu

Computer Engineering Department

Control

- Suppose we have a plan:
  - "Hey robot! Move north one meter, then east one meter, then north again for one meter."
- How do we execute this plan?
  - How do we go exactly one meter?
  - How do we go exactly north?

How do we control the robots?

Control Architectures

- Today, most robots control systems have a mixture of planning and behavior-based control strategies

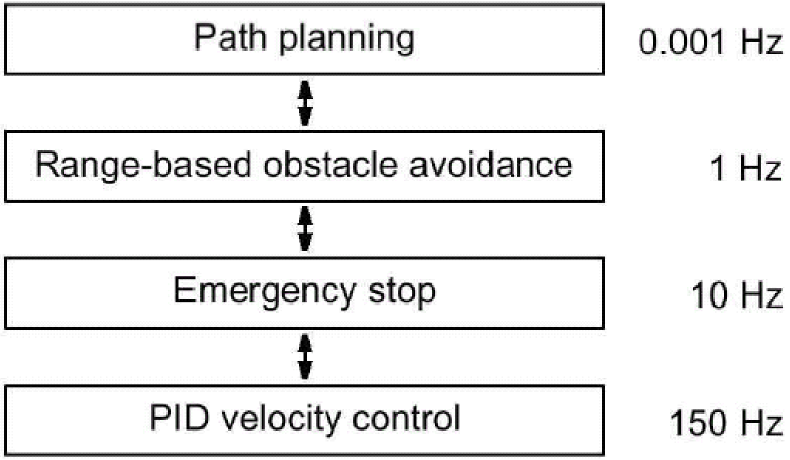
- To implement these strategies, a control architecture is used

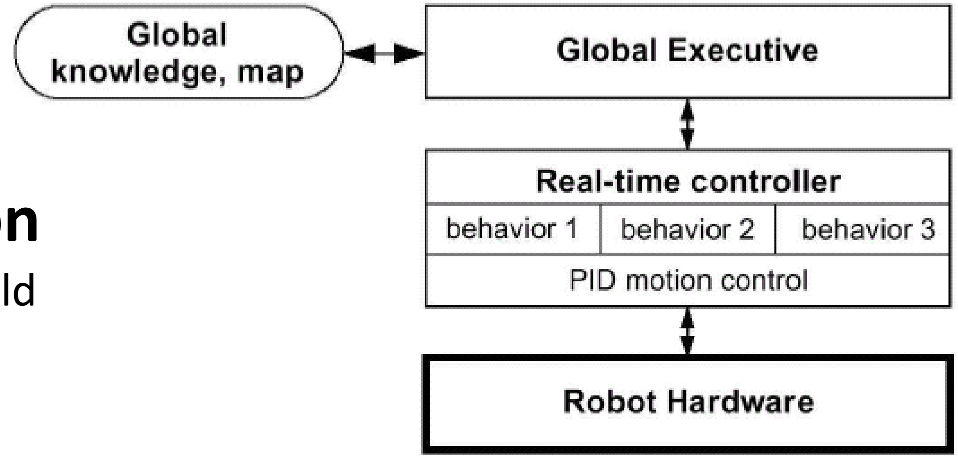
- Control architectures should consider:
  - **Code Modularity**
    - Allows programmers to interchange environment types sensors, path planners, propulsion, etc.

  - **Localization**
    - Embed specific navigation functions within modules to allow different levels of control (e.g., from task planning to wheel velocity control)

Control Architectures – Decomposition

- Decomposition allows us to modularize our control system based on different axes:
  - **Temporal Decomposition**
    - Facilitates varying degrees of real-time processes

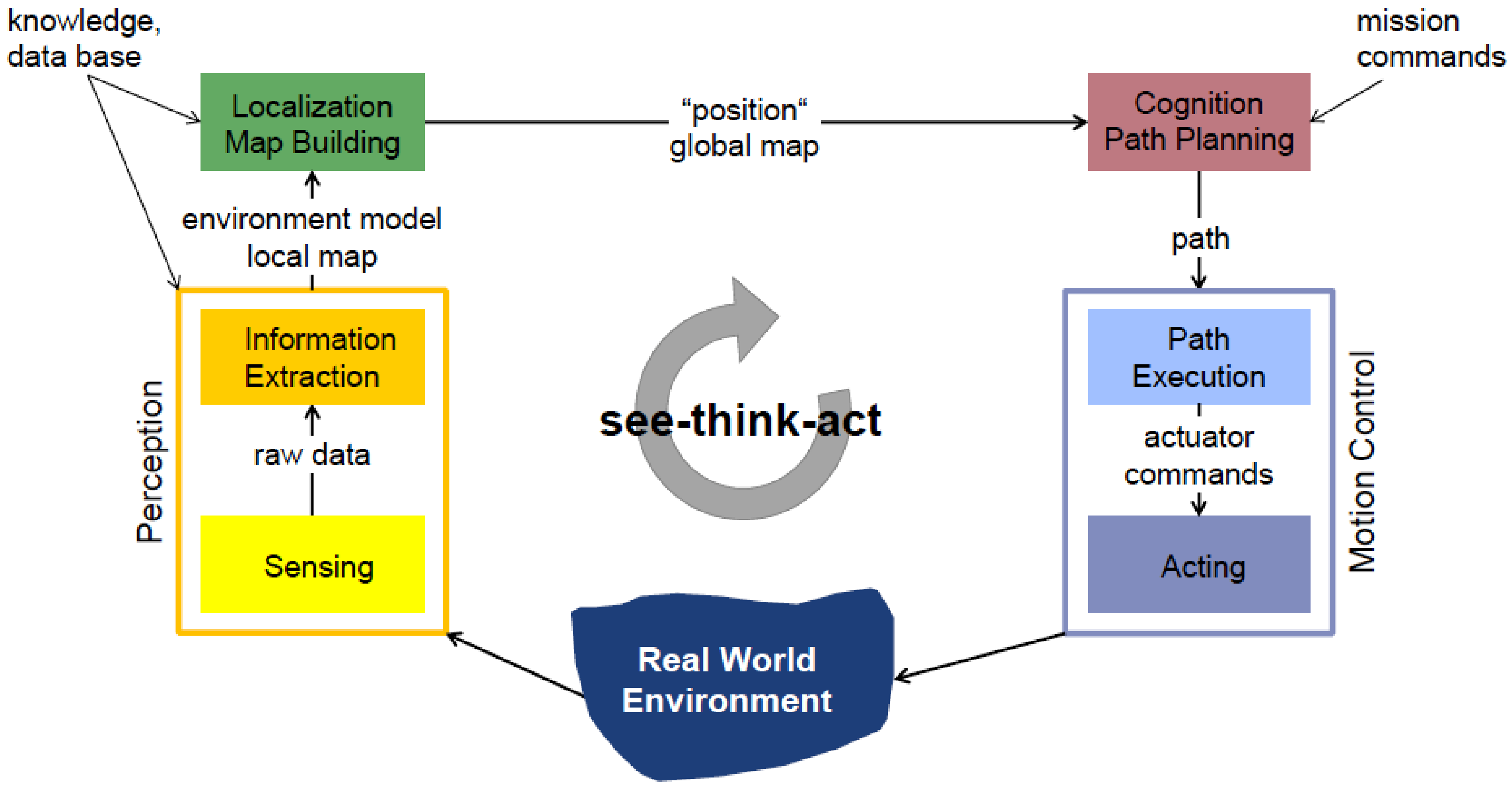




  - **Control Decomposition**
    - Defines how modules should interact: serial or parallel?

See-think-act Model of Mobile Robots

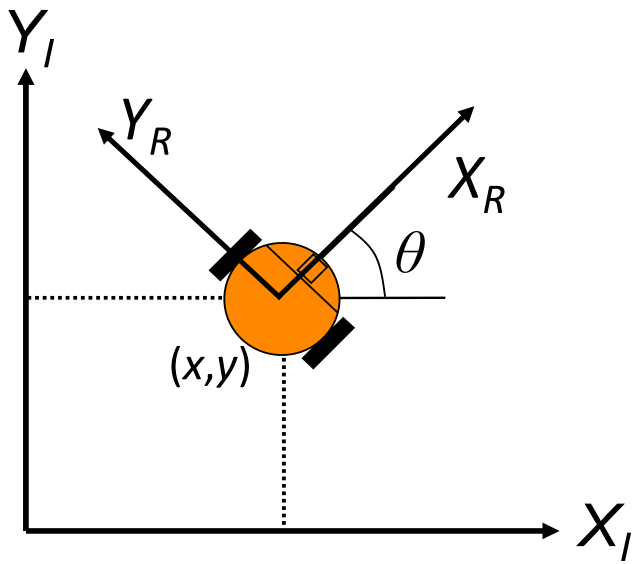
- An example of a control decomposition using a mixture of serial and parallel approaches



Recall: Mobile Robot Kinematics – Two Models

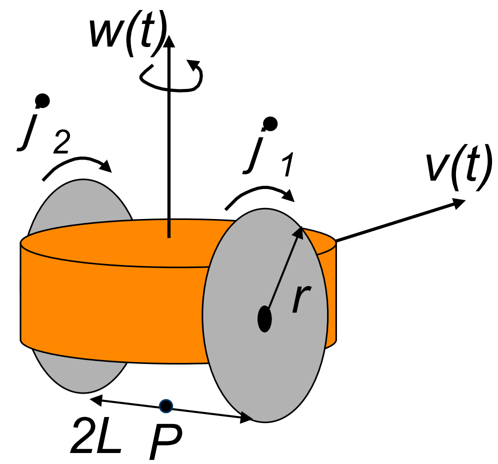
- Two models

How to design *v* and *w* so that the robot can follow a given trajectory?



$$\begin{cases} \dot{x} = v \cos\theta \\ \dot{y} = v \sin\theta \\ \dot{\theta} = w \end{cases}$$

Design for this model!



$$\begin{cases} \dot{x} = \frac{r}{2}(\dot{\varphi}_1 + \dot{\varphi}_2)\cos\theta \\ \dot{y} = \frac{r}{2}(\dot{\varphi}_1 + \dot{\varphi}_2)\sin\theta \\ \dot{\theta} = \frac{r}{2L}(\dot{\varphi}_2 - \dot{\varphi}_1) \end{cases}$$

Implement this model

The Basic Building Blocks

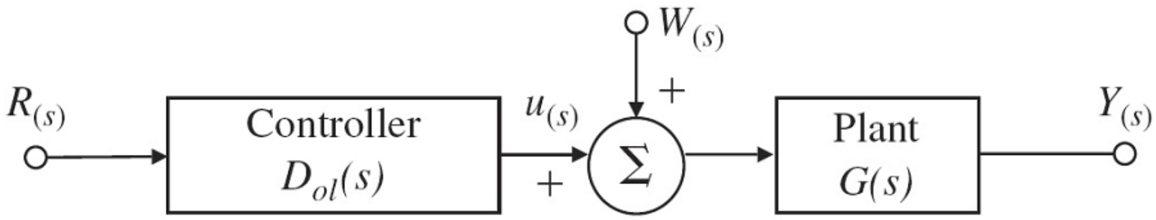
- ***State*** = Representation of what the system is currently doing
- ***Dynamics*** = Description of how the state changes
- ***Reference*** = What we want the system to do
- ***Output*** = Measurement of (some aspects of the) system
- ***Input*** = Control signal
- ***Feedback*** = Mapping from outputs to inputs

Control Theory = How to pick the input signal u?

Open-loop

- If I command the motors to “full power” for three seconds, the robot probably will go forward one meter

- Open-loop system with
  - Reference *R*
  - Control *U*
  - Disturbance *W*

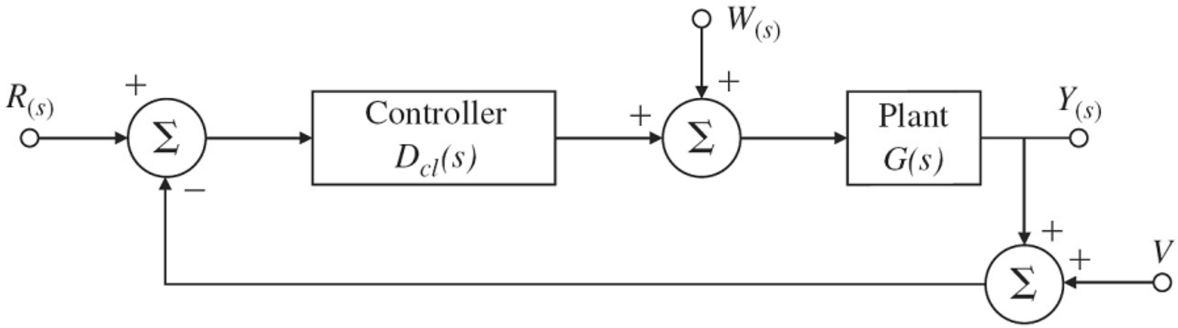


- Recall: Errors in odometry reading

Closed-loop

- Use real-time information about system performance to improve system performance

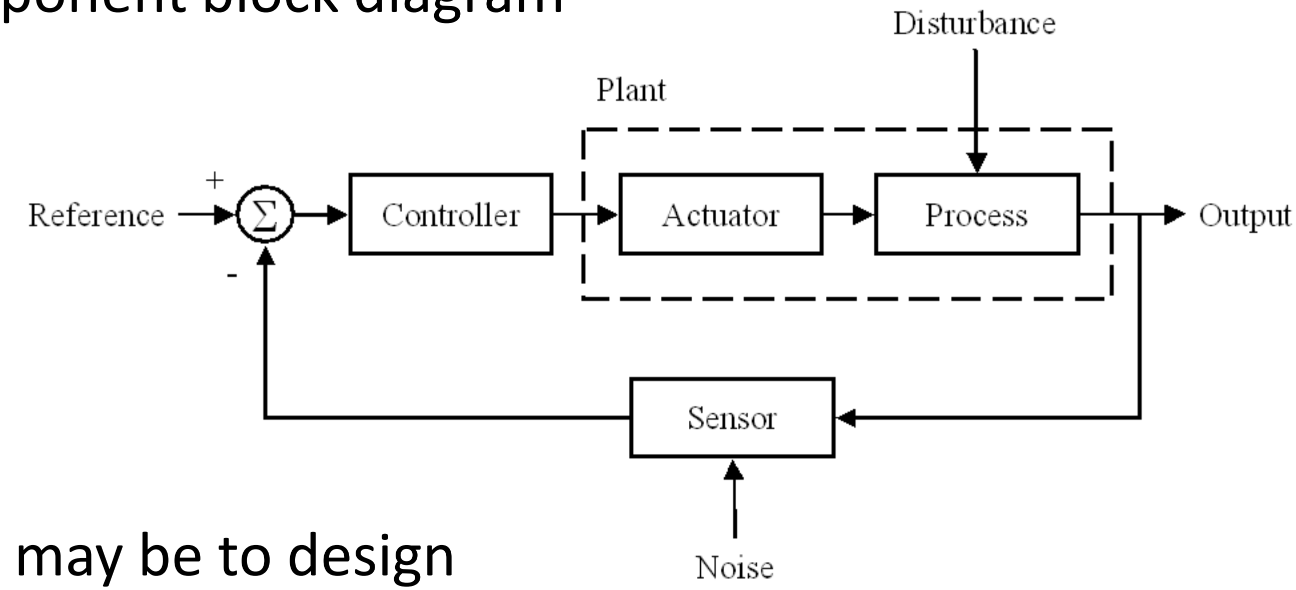
- Closed-loop system with
  - Reference *R*
  - Control *U*
  - Disturbance *W*
  - Sensor noise *V*



- Types:
  - Bang Bang
  - PID

Feedback Control System Basic Ingredients

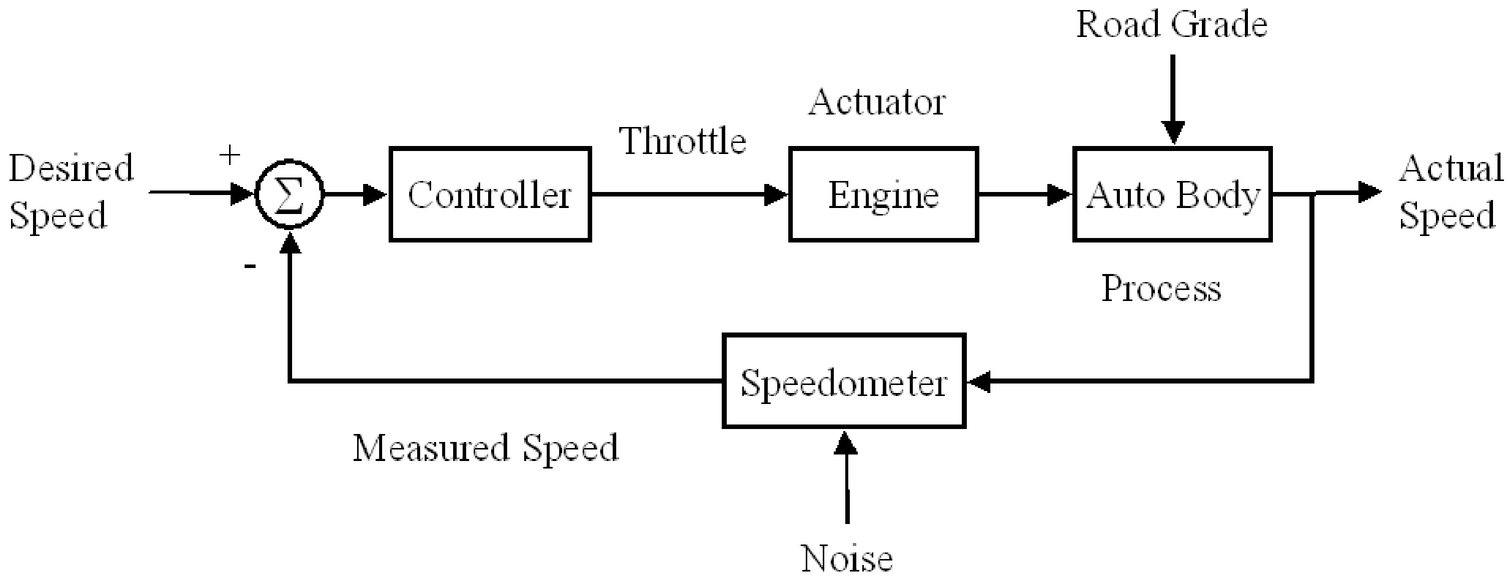
- Component block diagram



- Goal may be to design
  - **Regulating control**: maintain a fixed output
  - **Servo control**: follow a changing reference
- so that the system
  - is **stable** (e.g., bounded-input-bounded-output)
  - **rejects** disturbances
  - is **robust** to parameter changes

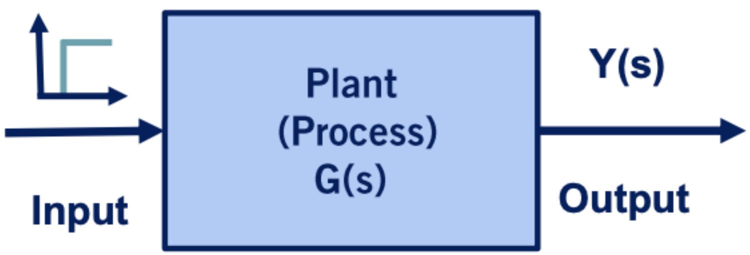
Control System: Example

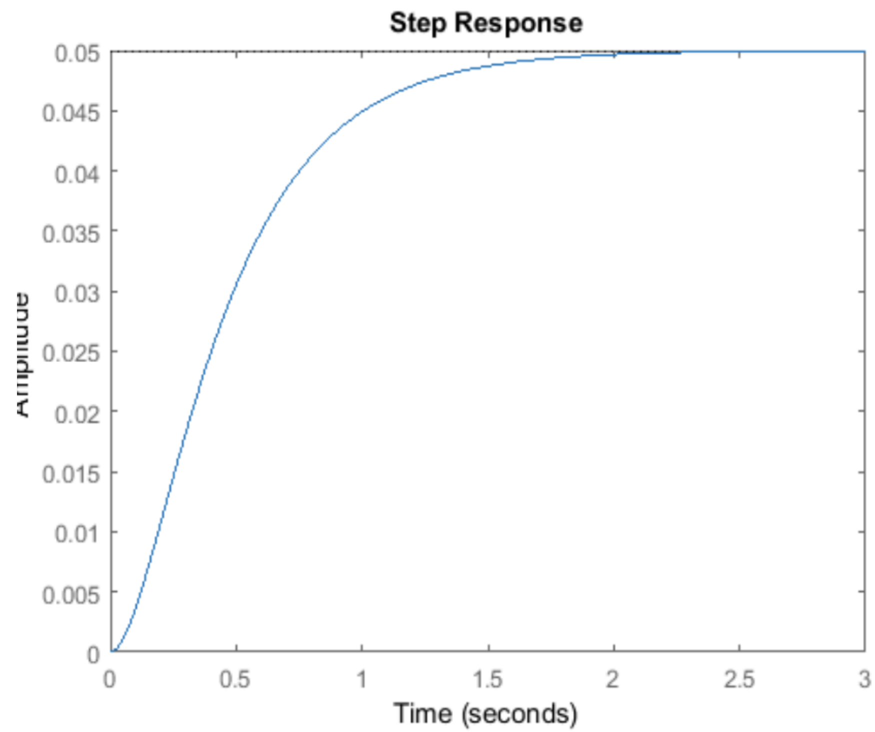
- Automobile cruise control



Open-Loop Step Response

- Let m = 1, b = 10, k = 20, F = 1





Time-Domain Specifications

- Rise time t_r*:* how fast the system reacts to a change in its input
- Setting time t_s *:* how fast the system's transient decays
- Overshoot M_p*:* How far the response grows beyond its final value during transients
- Peak time t_p *:* How far the response reaches the peak value

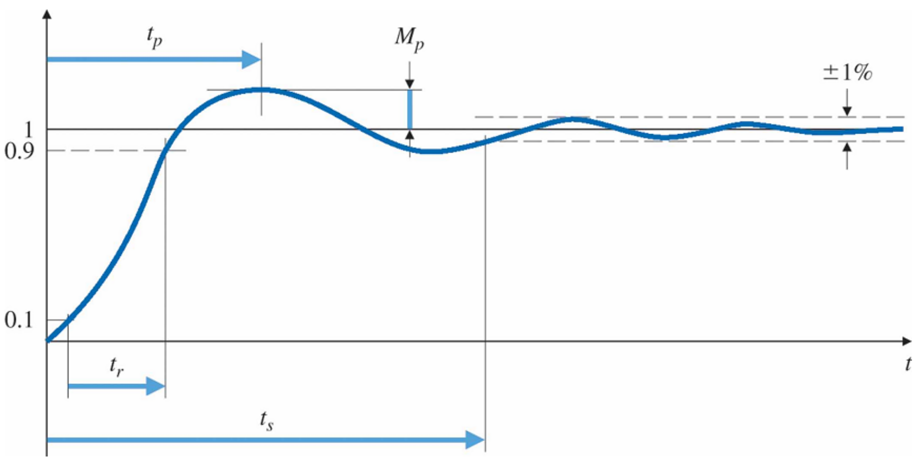


Figure : Definitions of time-domain specifications.

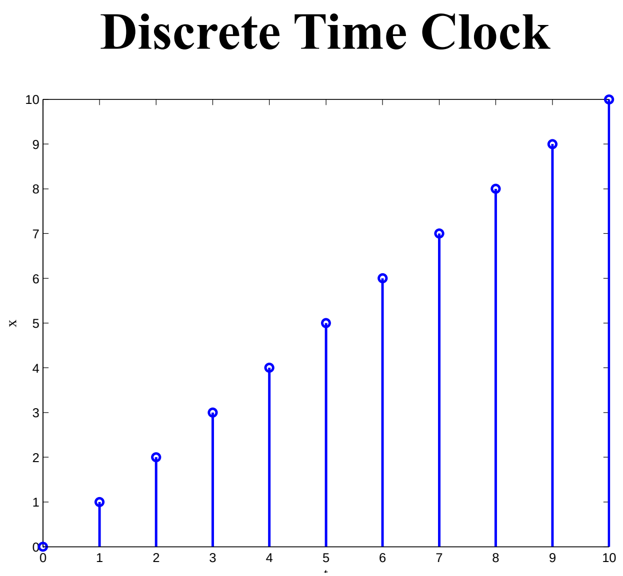
Dynamic Models

- Effective control strategies rely on predictive models
- **Discrete time:**

$$x_{k+1} = f(x_k, u_k)$$ ← Difference equation

Example: clock

$$x_{k+1} = x_k + 1$$



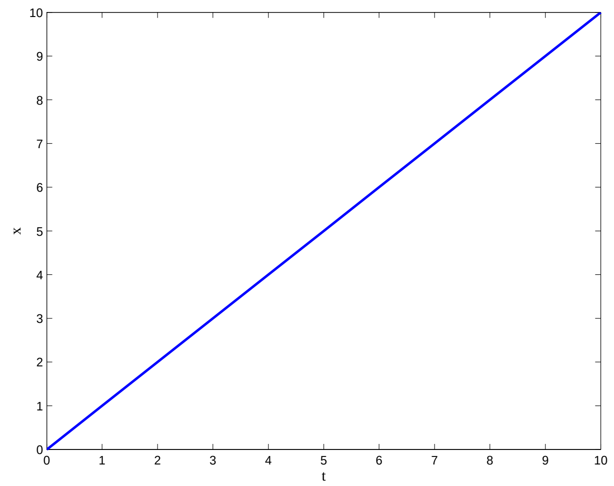
Dynamic Models

- Laws of Physics are all in continuous time
- Instead of “next” state, we need derivatives w.r.t. time
- **Continuous time:**

$$\frac{dx}{dt} = f(x,u) \sim \dot{x} = f(x,u)$$ ← Differential equation

Example: clock $\dot{x} = 1$

Continuous Time Clock



Dynamic Models

- Effective control strategies rely on predictive models
- For the unicycle model:

$$\begin{cases} \dot{x} = v \cos \theta \\ \dot{y} = v \sin \theta \\ \dot{\theta} = w \end{cases}$$

- In implementation, everything is discrete/sampled!
- From time step *k* to time step *k+ 1*, the position changes to

$$\begin{cases} x_{k+1} = x_k + v\Delta t \cos \theta_k \\ y_{k+1} = y_k + v\Delta t \sin \theta_k \\ \theta_{k+1} = \theta_k + w\Delta t \end{cases}$$

v, *w*: control input!

- Thank You!