

# Introduction

Stefano Carpin

Department of Computer Science and Engineering  
School of Engineering  
University of California, Merced  
<https://sites.ucmerced.edu/scarpin>  
<https://robotics.ucmerced.edu>



# Robot

- From the Czech word *robota* meaning *forced labor*.
- first appeared in 1920 in the play *R.U.R.* '*Rossum's Universal Robots*' by writer Karl Čapek



Source: <http://www.umich.edu/ engb415/literature/pontee/RUR/RURsmry.html>



# Autonomy / Autonomous Robot

**Autonomous**: from the Greek words *auto* (self) and *nomos* (law). Today is interpreted as self governed.



# Definitions of Robot

**Wikipedia** a machine – especially one programmable by a computer – capable of carrying out a complex series of actions automatically  
(<https://en.wikipedia.org/wiki/Robot>).

**Merriam-Webster** a machine that resembles a living creature in being capable of moving independently (as by walking or rolling on wheels) and performing complex actions (such as grasping and moving objects)  
(<https://www.merriam-webster.com/dictionary/robot>)

**RIA** A robot is defined as a reprogrammable, multifunctional manipulator designed to move material, parts, tools, or specialized devices through various programmed motions for the performance of a variety of tasks (RIA, 1979).



# Video Interlude...

# Robot Taxonomies

## Mobility

- aerial
- wheeled
- legged
- marine

## Applications

- industrial
- service
- medical
- field
- social



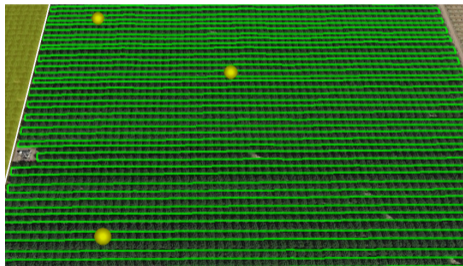
# Mobile Robots

- a robot not constrained to remain in a preassigned area
- not a formal or exhaustive definition
- may operate in a structured or unstructured environment



# Functionalities

- Planning
- Localization
- Navigation and Obstacle Avoidance



# Robots Feature Multiple Interacting Subsystems



# Open loop operation

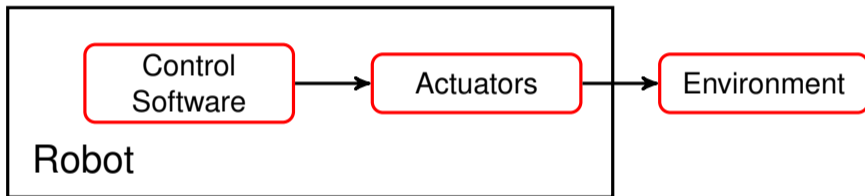


Figure: A robot operating in open loop.



# Robot Modeling: Agent View

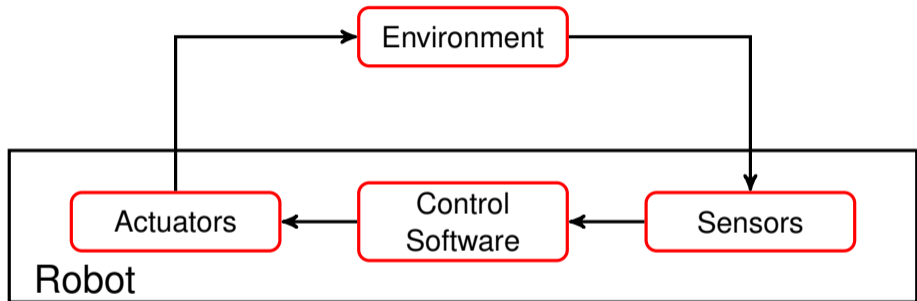


Figure: An agent view of a mobile robot.



# Robot Modeling: Dynamical System View

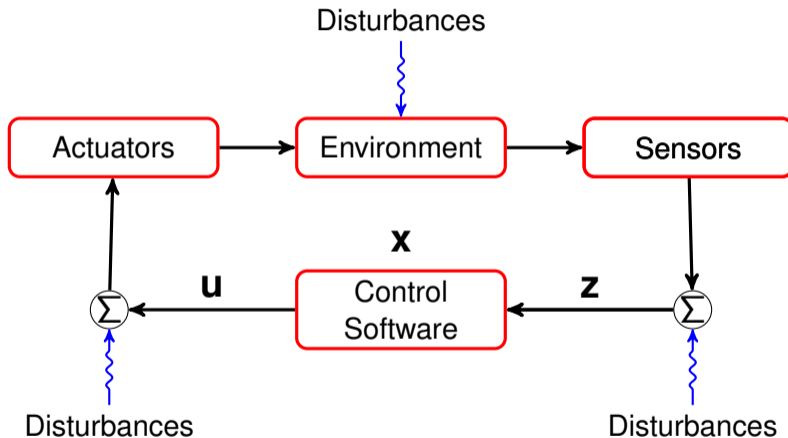
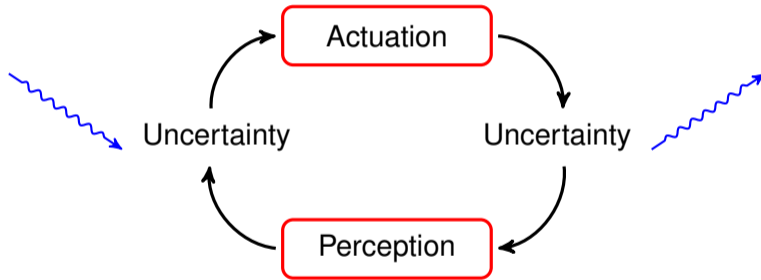


Figure: A dynamical system view of a mobile robot.



# Uncertainty Cycle



# Dynamical Systems Model

**Dynamical system** : a system whose behavior changes over time

**Feedback** : coupling between two systems influencing each other

State of a robot: position, velocity, joint values, ...

- x** state of the system
- u** input give to the system (actuators)
- z** information extracted from the system (sensors)



# Dynamical system equations

Continuous time model:

$$\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}, t) \quad (1)$$

$$\mathbf{z} = h(\mathbf{x}, \mathbf{u}, t) \quad (2)$$

Eq. (1) is the **state transition equation** and Eq. (2) is the **state observation equation**

Discrete time model:

$$\mathbf{x}_t = f(\mathbf{x}_{t-1}, \mathbf{u}_t, t) \quad (3)$$

$$\mathbf{z}_t = h(\mathbf{x}_t, \mathbf{u}_t, t). \quad (4)$$



# Time Invariant Systems

Continuous equations

$$\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}) \quad (5)$$

$$\mathbf{z} = h(\mathbf{x}, \mathbf{u}) \quad (6)$$

Discrete equations

$$\mathbf{x}_t = f(\mathbf{x}_{t-1}, \mathbf{u}_t) \quad (7)$$

$$\mathbf{z}_t = h(\mathbf{x}_t, \mathbf{u}_t) \quad (8)$$

Assumption of sensor reading independent from the input

$$\mathbf{z}_t = h(\mathbf{x}_t) \quad (9)$$

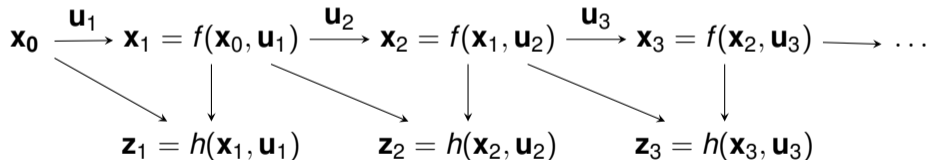
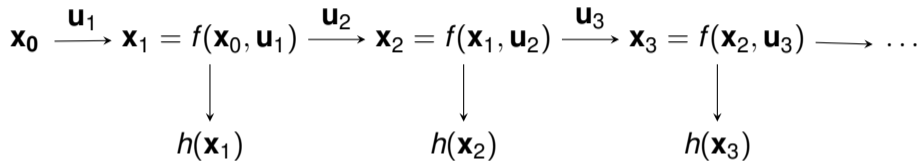


# State evolution

$$\mathbf{x}_0 \xrightarrow{\mathbf{u}_1} \mathbf{x}_1 = f(\mathbf{x}_0, \mathbf{u}_1) \xrightarrow{\mathbf{u}_2} \mathbf{x}_2 = f(\mathbf{x}_1, \mathbf{u}_2) \xrightarrow{\mathbf{u}_3} \mathbf{x}_3 = f(\mathbf{x}_2, \mathbf{u}_3) \rightarrow \dots \rightarrow \mathbf{x}_n = f(\mathbf{x}_{n-1}, \mathbf{u}_n)$$



# State and sensor acquisition



# Robot (software) Controller

- our eventual high level objective would be to get a function  $\pi$  that defines the action to take in each state so that the robot reaches a desired state (planning view)

$$\pi(\mathbf{x}_t) = \mathbf{u}_{t+1}$$

- the state, however, is almost invariably not known with certainty. So we also need an algorithm to determine the state (estimation view)

$$\mathbf{x}_t = g(\mathbf{x}_1, \dots, \mathbf{x}_{t-1}, \mathbf{z}_1, \dots, \mathbf{z}_{t-1})$$

- these functions are computed by the algorithms/programs we will study in this course
- must be pursued in parallel



# Robot Software Architecture

- principle for modular decomposition of a complex software system
- multiple approaches and solutions
- ROS based architectures emphasize data exchange and asynchronous interactions between modules

