Robotics Principles and Applications

Fumiya Iida

Bio-Inspired Robotics Lab, Department of Engineering

University of Cambridge

Lecture note available at:
http://mi.eng.cam.ac.uk/Main/FI224

Outline

- Introduction
- Kinematic Motion Control
- Discussion
  - Advanced Motion Control
  - Trends and Challenges

"Rise of the Robots"

is happening because:

1. Cost decrease of technologies
2. Readiness increase of technologies
3. Connected and shared technologies

The Economist, March 2014

Robotics Market and Opportunities

- Industrial manufacturing
- Cleaning
- Medical robotics
- Entertainment/edutainment
- Logistics/autonomous warehouse
- Autonomous cars
- Industrial inspection
- Surveillance and rescue
- Construction and mining
- Agriculture
- Health and elderly care
- Personal/service robots

H2020 euRobotics Multi-Annual Roadmap (2016)

Robotics Market and Opportunities

- Industrial manufacturing
- Cleaning
- Medical robotics
- Entertainment/edutainment
- Logistics/autonomous warehouse
- Autonomous cars

- Industrial inspection
- Surveillance and rescue
- Construction and mining
- Agriculture

- Health and elderly care
- Personal/service robots

H2020 euRobotics Multi-Annual Roadmap (2016)

Moravec's Paradox

It is comparatively easy to make computers exhibit adult level performance on intelligence tests or playing checkers, but difficult or impossible to give them the skills of a one-year-old when it comes to perception and mobility.

Moravec, Hans (1988), Mind Children, Harvard University Press

Industrial Environment

High predictability
High programmability
Algorithmic
Top-down design
Rigid interactions

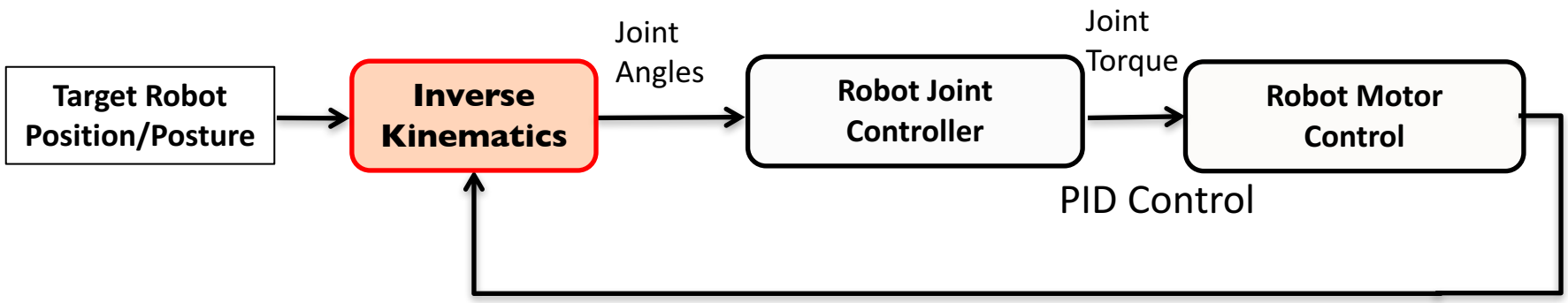
Real-World Environment

Low predictability
Low programmability
Creativity, adaptation
Design for emergence
Soft interactions

Outline

- Introduction
- Kinematic Motion Control
- Discussion
  - Advanced Motion Control
  - Trends and Challenges

What is Kinematic Control



<u>Kinematic Control:</u>
Specify desired robot position every time step, and calculate joint positions/angles from “Inverse Kinematics”. Position Controller then takes care of all the ”physics”.

<u>Forward and Inverse Kinematics:</u>
“Forward Kinematics” specifies position/posture of a robot given its joint angles, and “Inverse Kinematics” specifies joint angles given position/posture of a robot.

Try this before any other control methods!

Chapter 2 Kinematic Control

KINEMATIC CONTROL OF ROBOT MANIPULATORS

Kinematics of Robot Manipulators

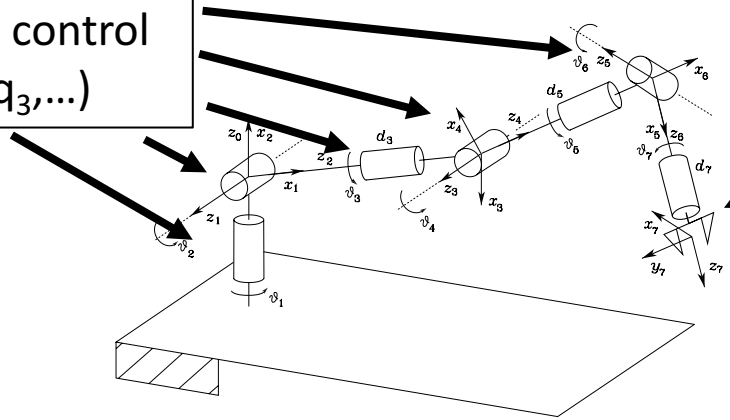
- Forward kinematics

$$[\dot{x}, \dot{y}, \dot{z}, \dot{\alpha}, \dot{\beta}, \dot{\chi}]^T = f(\dot{q}_1, \dot{q}_2, \ldots, \dot{q}_n)$$

- Inverse kinematics

$$[\dot{q}_1, \dot{q}_2, \ldots, \dot{q}_n]^T = f^{-1}(\dot{x}, \dot{y}, \dot{z}, \dot{\alpha}, \dot{\beta}, \dot{\chi})$$

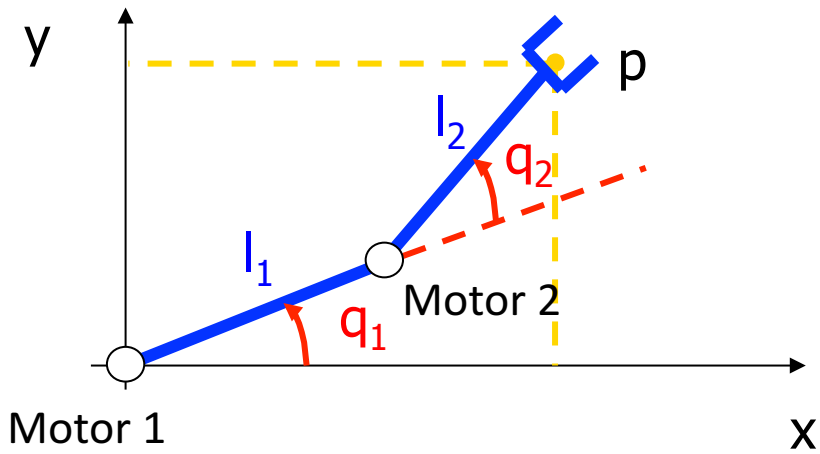
This is what you need to control (q_1, q_2, q_3,…)



This is what you want to achieve (x, y, z, etc.)

Simple Kinematic Control

- Specify joint angles q_1 and q_2 to achieve the position of end-effector at **x**



$$\mathbf{x}=\mathbf{f}(q_1, q_2)$$

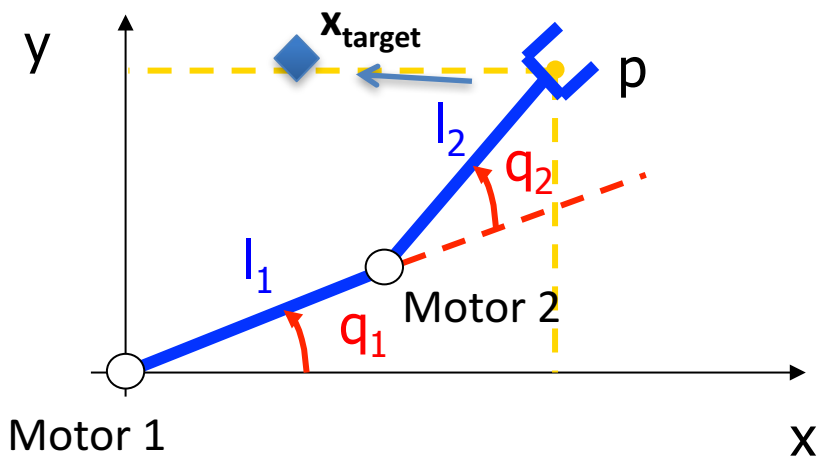
$$x = (l_1 \cos q_1 + l_2 \cos(q_1 + q_2))$$

$$y = (l_1 \sin q_1 + l_2 \sin(q_1 + q_2))$$

- Then think about position control of motor 1 and motor 2 to achieve the joint angles

Simple Kinematic Control

- How to determine q_1, q_2 for a desired target position of end-effector **x**?



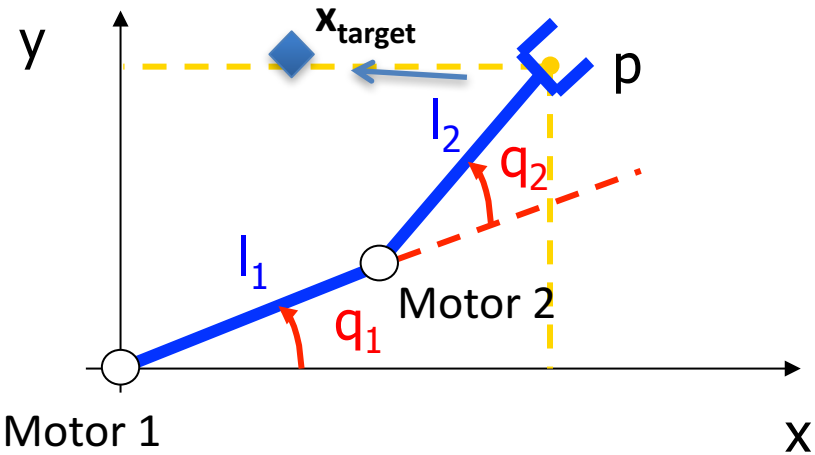
$$[q_1, q_2]^T = \mathbf{f}^{-1}(\mathbf{x})$$

$$x = (l_1 \cos q_1 + l_2 \cos(q_1 + q_2))$$

$$y = (l_1 \sin q_1 + l_2 \sin(q_1 + q_2))$$

$q_1 = ?$

$q_2 = ?$



$$x = (l_1 \cos q_1 + l_2 \cos(q_1 + q_2))$$

$$y = (l_1 \sin q_1 + l_2 \sin(q_1 + q_2))$$

$$x^2 + y^2 = l_1^2 + l_2^2 + 2 l_1 l_2 cos(q_2)$$

$$cos(q_2) = \frac{x^2 + y^2 - l_1^2 - l_2^2}{2 l_1 l_2}$$

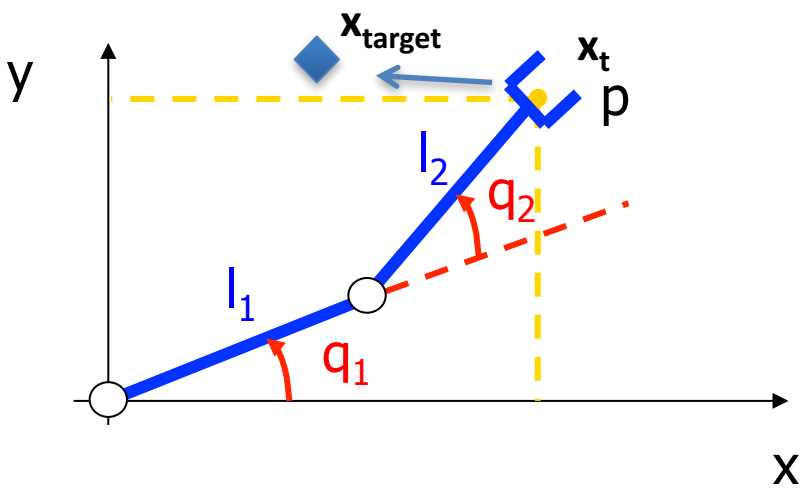
$$q_2 = \pm cos^{-1}\left(\frac{x^2 + y^2 - l_1^2 - l_2^2}{2 l_1 l_2}\right)$$

Given q_2,

$$q_1 = atan2(y, x) \pm atan2\left(\sqrt{x^2 + y^2 - d_1^2}, d_1\right)$$

Where, $$d_1 = \frac{x^2 + y^2 + l_1^2 - l_2^2}{2 l_1}$$

Numerical Approach to Kinematic Control



Coordinate of end-effector

$$\mathbf{x} = [x_1, x_2, \cdots x_n]^T$$

Coordinate of joint space

$$\mathbf{q} = [\theta_1, \theta_2, \cdots \theta_n]^T$$

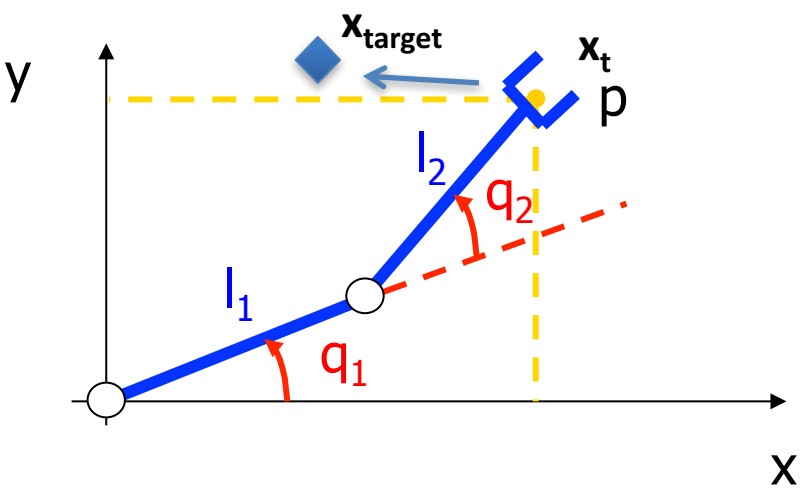
Kinematic constraints

$$\phi = [\phi_1(q), \phi_2(q), \cdots \phi_n(q)]^T$$

Relationship between velocities of q and x

$$\dot{\mathbf{x}} = \frac{\partial \phi}{\partial \mathbf{q}} \dot{\mathbf{q}} = \underbrace{\mathbf{J}(\mathbf{q})}_{\text{Jacobian}} \dot{\mathbf{q}} \qquad \text{or} \qquad \dot{\mathbf{q}} = \mathbf{J}^{-1}(\mathbf{q}) \dot{\mathbf{x}}$$

More Detailed Calculation



$$[x, y]^T = \phi(q_1, q_2)$$

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} l_1 cos(q_1) + l_2 cos(q_1 + q_2) \\ l_1 sin(q_1) + l_2 sin(q_1 + q_2) \end{bmatrix}$$

$$\mathbf{J}(\mathbf{q}) = \frac{\partial \phi}{\partial \mathbf{q}}$$

$$= \begin{bmatrix} \frac{\partial \phi_1}{\partial q_1} & \frac{\partial \phi_1}{\partial q_2} \\ \frac{\partial \phi_2}{\partial q_1} & \frac{\partial \phi_2}{\partial q_2} \end{bmatrix}$$

$$= \begin{bmatrix} -l_1 sin(q_1) - l_2 sin(q_1 + q_2) & -l_2 sin(q_1 + q_2) \\ l_1 cos(q_1) + l_2 cos(q_1 + q_2) & l_2 cos(q_1 + q_2) \end{bmatrix}$$

Once we obtain J(q), we can inverse it in matlab by
$J^{-1}(q)$ = inv(J(q))
or, if not invertible,
$J^{+}(q)$ = pinv(J(q)) => Pseudo-inverse

This should be enough to reach the target by running:

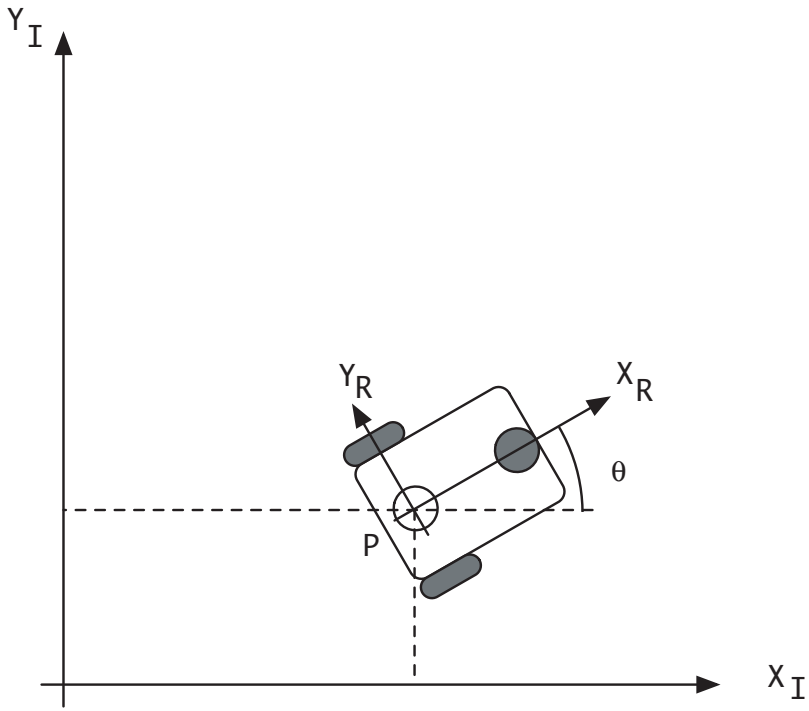
$q_{t+1} = q_t + J^{-1} (x_{target} - x_t)$
$= q_t + J^{-1} (x_{target} - f(q_t))$

Chapter 2 Kinematic Control

KINEMATIC CONTROL OF MOBILE ROBOTS

Kinematics of Mobile Robots

What is kinematics? Motion of system without forces, i.e. “geometry of motion”



Forward Kinematics

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = f(\dot{\varphi}_1, \ldots, \dot{\varphi}_n, \beta_1, \ldots, \beta_m, \dot{\beta}_1, \ldots, \dot{\beta}_m)$$

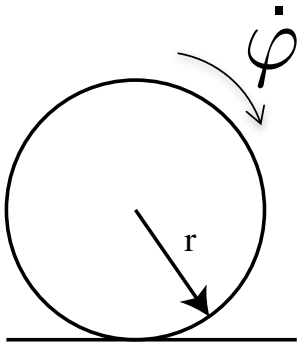
Inverse Kinematics

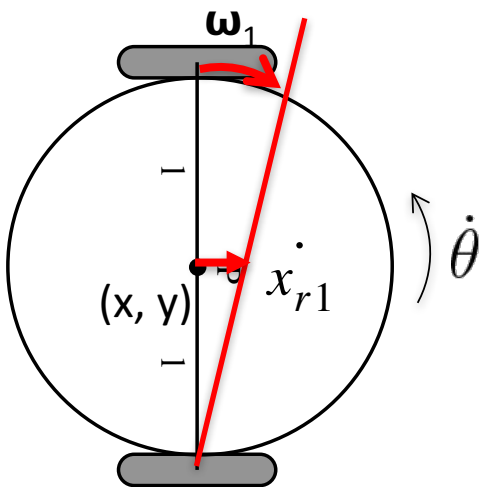
$$\begin{bmatrix} \dot{\varphi}_1 & \ldots & \dot{\varphi}_n & \beta_1 & \ldots & \beta_m & \dot{\beta}_1 & \ldots & \dot{\beta}_m \end{bmatrix}^T = f(\dot{x}, \dot{y}, \dot{\theta})$$

$\dot{\varphi}$ wheel speed β angle of steering wheels

Kinematics of Simple Mobile Robot

- Forward kinematics of a two wheeled robot





$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = f(l, r, \theta, \dot{\varphi}_1, \dot{\varphi}_2)$$

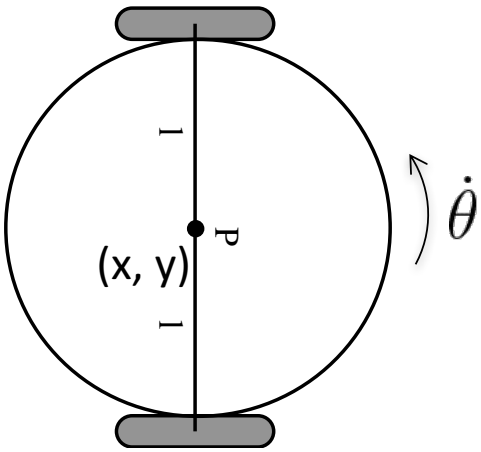
$$\omega_1 = \frac{r\dot{\varphi}_1}{2l} \quad \omega_2 = \frac{-r\dot{\varphi}_2}{2l} \quad \dot{x}_{r1} = \frac{1}{2}r\dot{\varphi}_1 \quad \dot{x}_{r2} = \frac{1}{2}r\dot{\varphi}_2$$

$$\dot{x} = \dot{x}_{r1} + \dot{x}_{r2}$$

$$\dot{y} = 0$$

$$\dot{\theta} = \omega_1 + \omega_2$$

Kinematics of Simple Mobile Robot

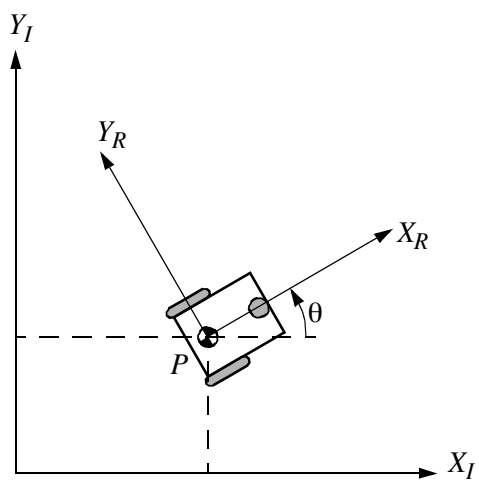


Forward kinematics (local coordinate)

$$\dot{x} = \dot{x}_{r1} + \dot{x}_{r2}$$
$$\dot{y} = 0$$
$$\dot{\theta} = \omega_1 + \omega_2$$

where $\omega_1 = \frac{r\dot{\varphi}_1}{2l}$ $\omega_2 = \frac{-r\dot{\varphi}_2}{2l}$ $\dot{x}_{r1} = \frac{1}{2}r\dot{\varphi}_1$ $\dot{x}_{r2} = \frac{1}{2}r\dot{\varphi}_2$

Forward kinematics (global coordinate)



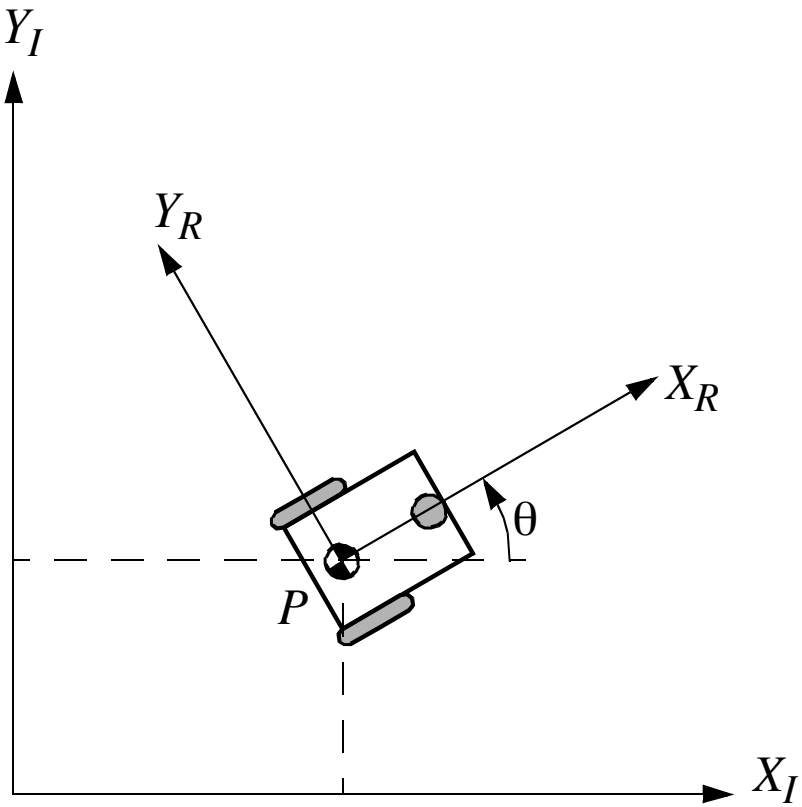
global coordinate X_I, Y_I

$$\dot{\xi}_I = R(\theta)^{-1}\dot{\xi}_R$$

where $\xi_I = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix}$ $R(\theta)^{-1} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$

$$\dot{\xi}_I = R(\theta)^{-1}\begin{bmatrix} \frac{r\dot{\varphi}_1}{2} + \frac{r\dot{\varphi}_2}{2} \\ 0 \\ \frac{r\dot{\varphi}_1}{2l} + \frac{-r\dot{\varphi}_2}{2l} \end{bmatrix}$$

Kinematics of Simple Mobile Robot



global coordinate X_I, Y_I

Forward kinematics

$$\dot{\xi}_I = R(\theta)^{-1}\dot{\xi}_R$$

Inverse kinematics

$$\dot{\xi}_R = R(\theta)\dot{\xi}_I$$

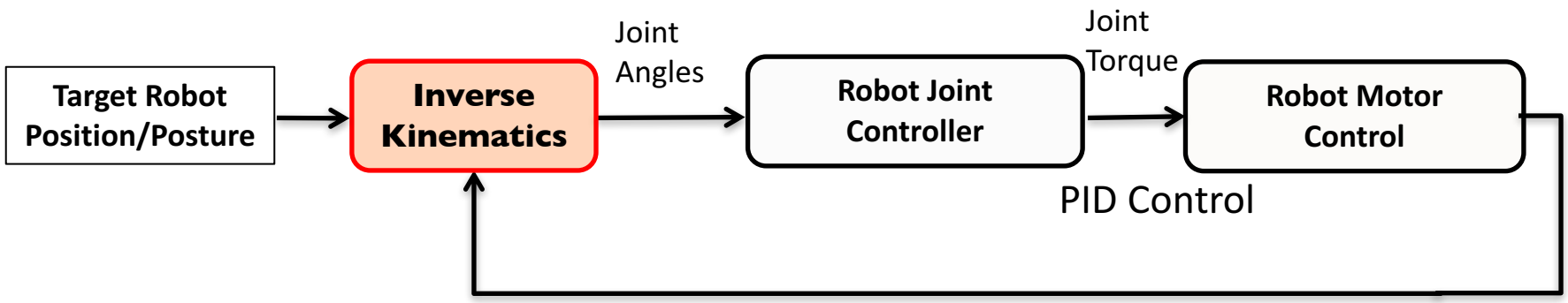
where $\xi_I = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix}$ $\quad R(\theta) = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$

$$\underbrace{\begin{bmatrix} \frac{r\dot{\varphi}_1}{2} + \frac{r\dot{\varphi}_2}{2} \\ 0 \\ \frac{r\dot{\varphi}_1}{2l} + \frac{-r\dot{\varphi}_2}{2l} \end{bmatrix}}_{\text{Motor Control}} = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \underbrace{\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix}}_{\text{Target velocity}}$$

Outline

- Introduction
- Kinematic Motion Control
- Discussion
  - Advanced Motion Control
  - Trends and Challenges

What is Kinematic Control

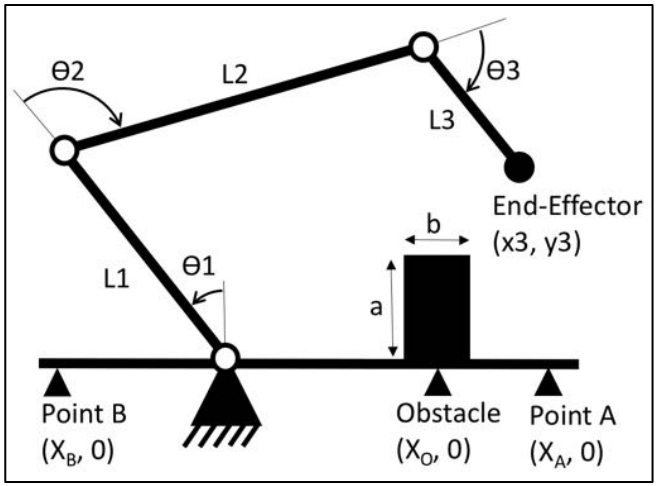


Designing your own robot:

1. Determine the kinematic chain of your robot
2. Calculate Inverse Kinematics of your robot
3. Determine target robot position/posture (can be time series)
4. Run the controller, then the robot follows the given trajectory

What can this approach NOT do?

Example Question 1



Given a three-link planar manipulator fixed on the base to the origin O, derive the time-series trajectories of three joints θ1(t), θ2(t), and θ3(t) to move the end-effector from Point A to Point B with a minimum traveling distance. L1 = 0.8m, L2 = 0.9m, L3 = 0.6m; X_A = 1.3m, X_B = -1.0m, X_O = 0.9m, a = 0.5m, b =0.3m.

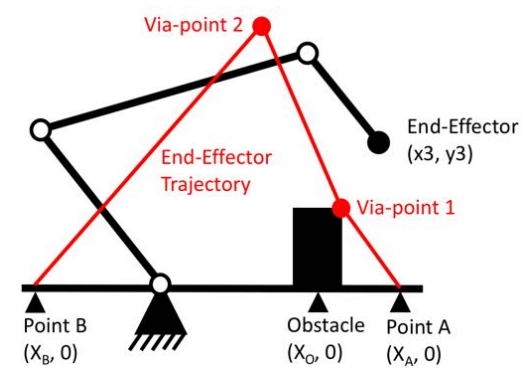
Answer:

First derive the Jacobian matrix:

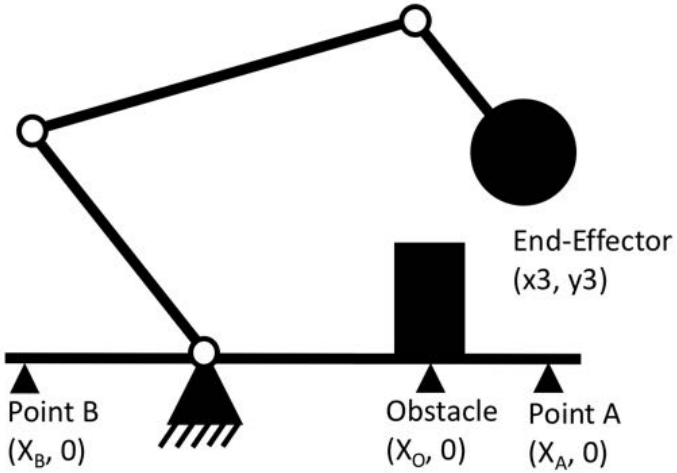
$$\mathbf{J} = \begin{bmatrix} -l_1c_1 - l_2c_{1-2} - l_3c_{1-2+3},\ l_2c_{1-2} + l_3c_{1-2+3},\ -l_3c_{1-2+3} \\ -l_1s_1 - l_2s_{1-2} - l_3s_{1-2+3},\ l_2s_{1-2} + l_3s_{1-2+3},\ -l_3s_{1-2+3} \end{bmatrix}$$

Then calculate θ1(t), θ2(t), and θ3(t), with respect to targets (x3, y3)

Targets (x3, y3) need to be determined by considering the constraints of task-environment

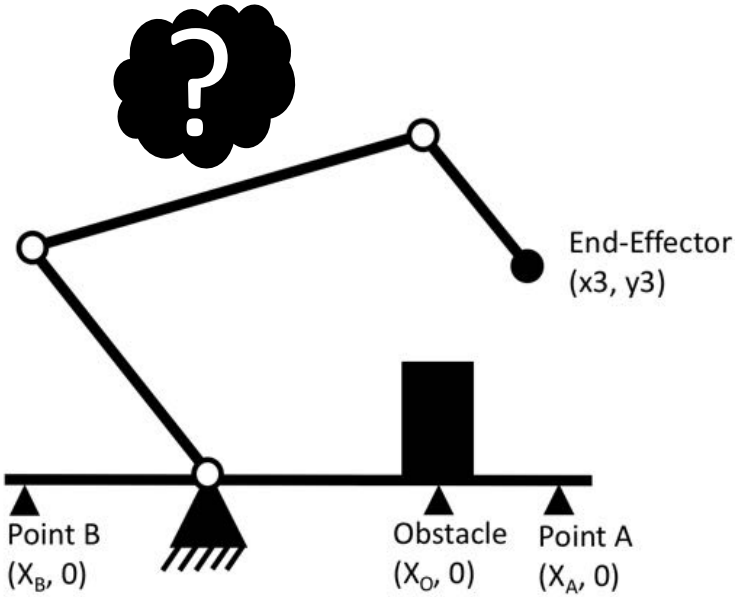


Example Question 2



What if there is a heavy object/end effector. Is Inverse-Kinematic Controller going to work? What is the limit of the payload?

Example Question 3

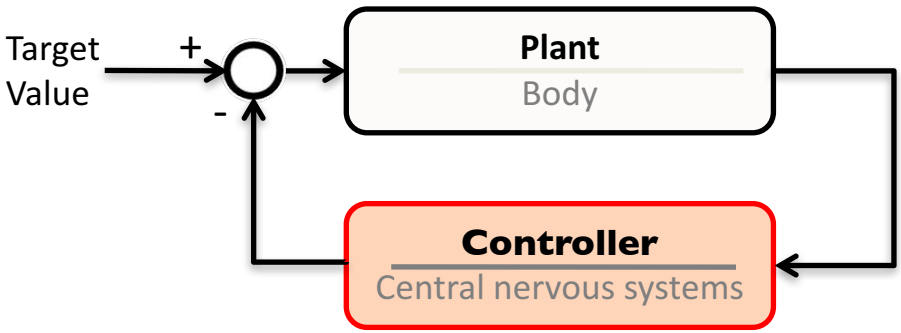


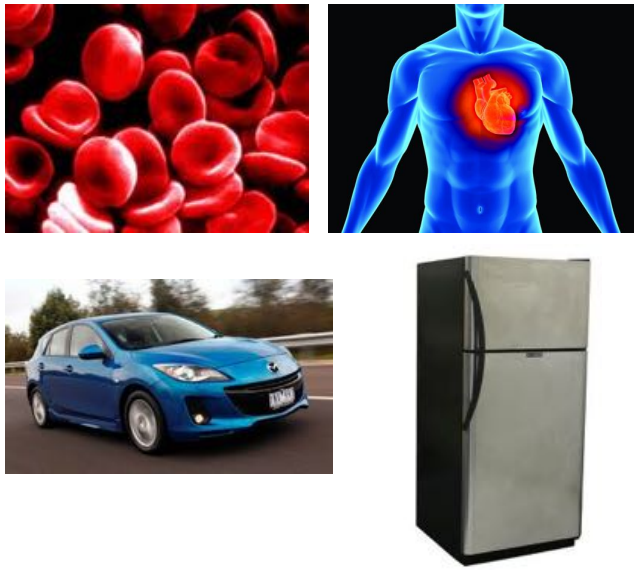
What if there is an unspecified object (or humans) in the environment? Any way to achieve the task without conflicting to the object?

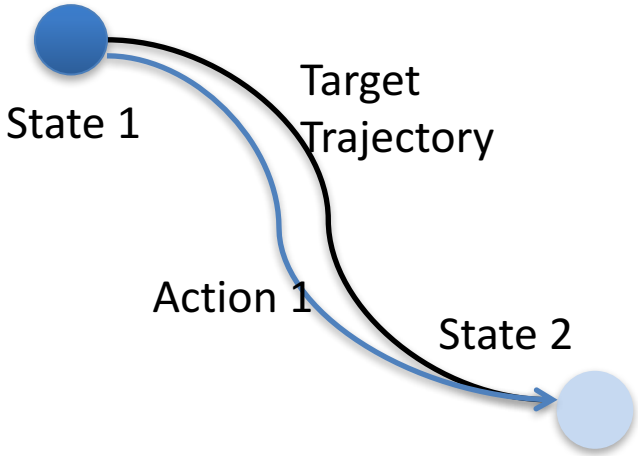
Various Approaches of Motion Control

1. Control engineering approach
2. Sense-think-act approach (CS approach)
3. Behaviour-based approach (Bio approach)
4. Passivity-based approach (MechE approach)
5. Soft approach (Material/Chem approach)

3.1. Control Engineering Approach



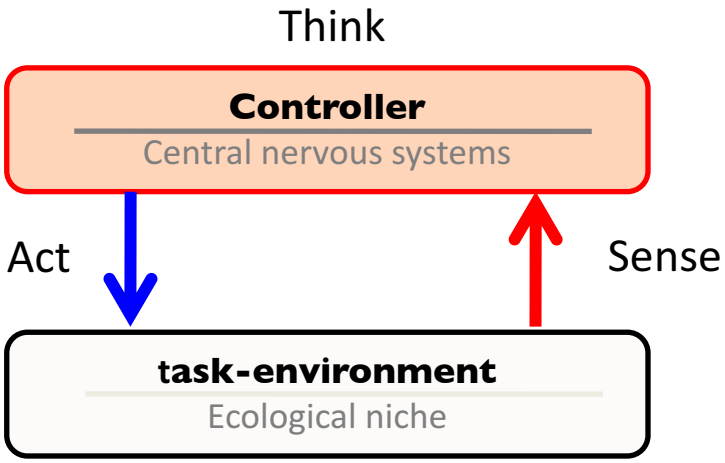


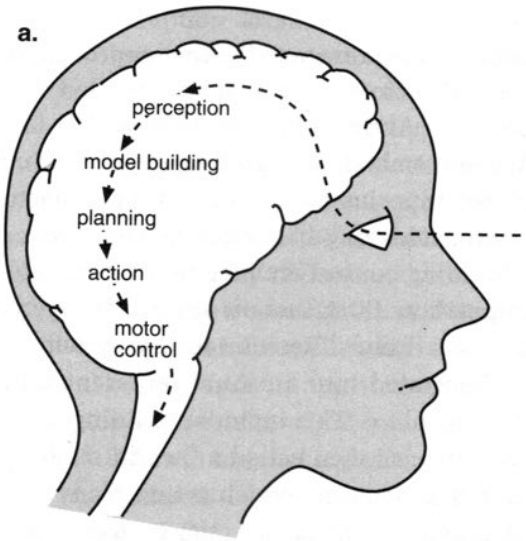


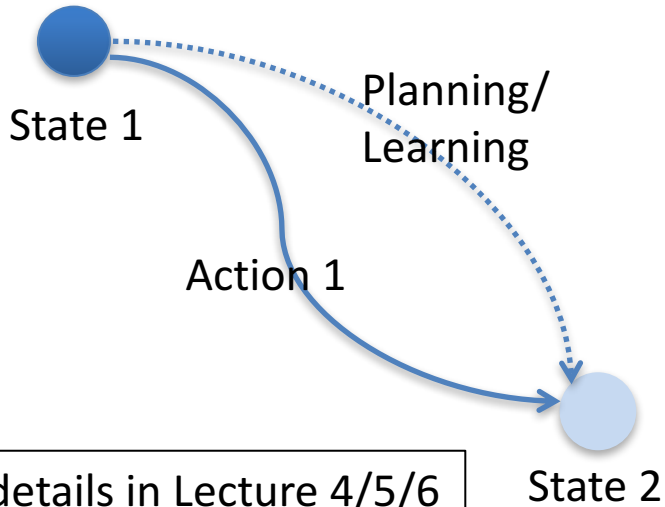
- States and actions are represented without ambiguity
- Model-based approach (out-of-model is regarded as noise)
- Target trajectory is given
- Limited to simple (linear) systems

More details in Lecture 2

3.2 Sense-Think-Act Approach







More details in Lecture 4/5/6

- States and actions are represented without ambiguity
- Often include planning and learning (i.e. autonomously generate target trajectories)
- Sense-think-act scheme (the system always follows this order of processes)
- Also known as “Good Old Fashion AI” - GOFAI

Constructing World Model

Example: Shakey Robot (1966)

What is ***the world model*** for Shakey to navigate from Point A to B?

- Perception
  (understanding of sensor signals)
  Obstacles, Floors, Doors, Walls, Rooms, Humans, Slippage,
- Consequences of own motions
  Move fwd/bwd, turning, stuck
- Priorities
- Changes of tasks and environment
- …

Initial World Model

$(\forall x \forall y \forall z)[CONNECTS(x,y,z) \rightarrow CONNECTS(x,z,y)]$

CONNECTS(DOOR1,ROOM1,ROOM5)
CONNECTS(DOOR2,ROOM2,ROOM5)
CONNECTS(DOOR3,ROOM3,ROOM5)
CONNECTS(DOOR4,ROOM4,ROOM5)

LOCINROOM(f,ROOM4)	INROOM(BOX1,ROOM1)
AT(BOX1,a)	INROOM(BOX2,ROOM1)
AT(BOX2,b)	INROOM(BOX3,ROOM1)
AT(BOX3,c)	INROOM(ROBOT,ROOM1)
AT(LIGHTSWITCH1,d)	INROOM(LIGHTSWITCH1,ROOM1)
ATROBOT(e)	PUSHABLE(BOX1)
TYPE(BOX1,BOX)	PUSHABLE(BOX2)
TYPE(BOX2,BOX)	PUSHABLE(BOX3)

Operators

gotoI(m): Robot goes to coordinate location *m*.

Preconditions:

$(ONFLOOR) \wedge (\exists x)[INROOM(ROBOT,x) \wedge LOCINROOM(m,x)]$

Delete list: ATROBOT($),NEXTTO(ROBOT,$)

Add list: ATROBOT(*m*)

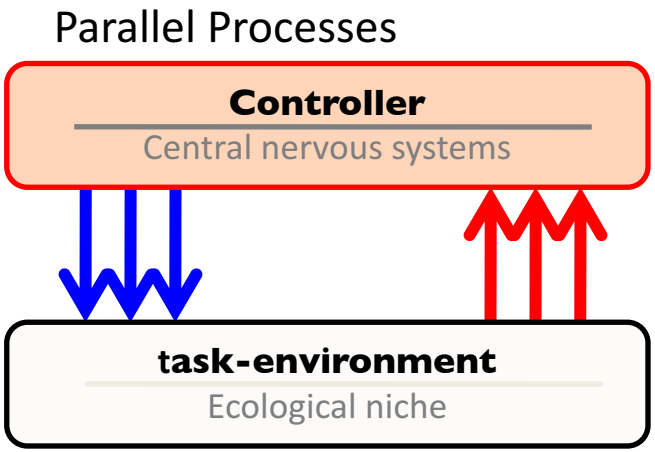
gotoN(m): Robot goes next to item *m*.

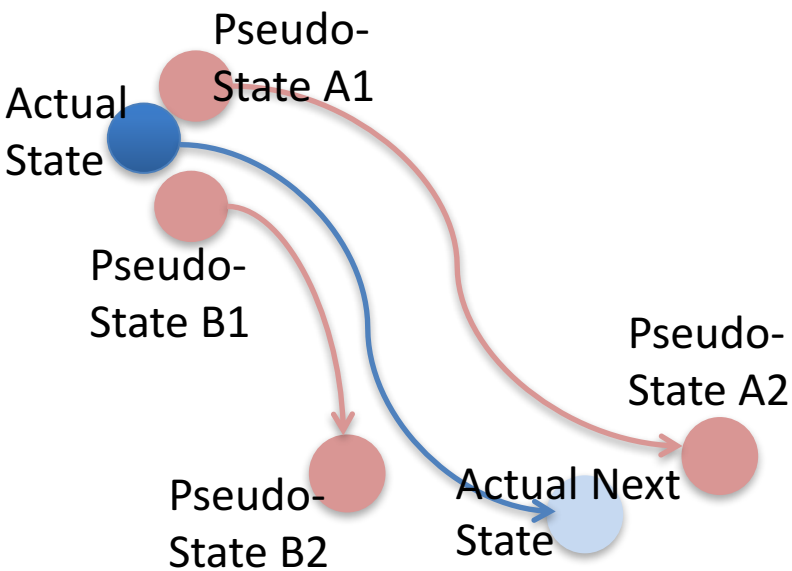
Preconditions:

$(ONFLOOR) \wedge [(\exists x)[INROOM(ROBOT,x) \wedge INROOM(m,x)] \vee (\exists x)(\exists y)$
$[INROOM(ROBOT,x) \wedge CONNECTS(m,x,y)]]$

Delete list: ATROBOT($),NEXTTO(ROBOT,$)

3.3 Behaviour-Based Approach

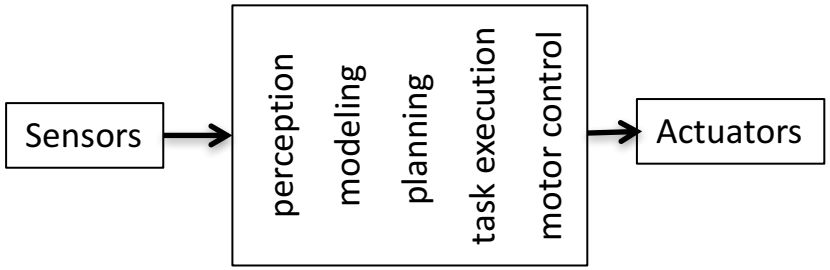


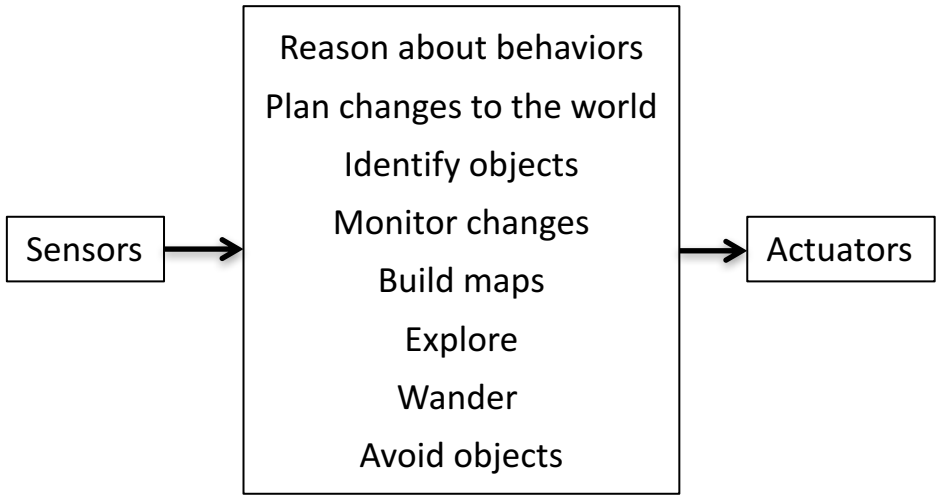


- Different processes have different state/action representations
- Parallel processes can run independently
- Sensing and action processes are asynchronous

More details in Lecture 5

Principles of Parallel Processes

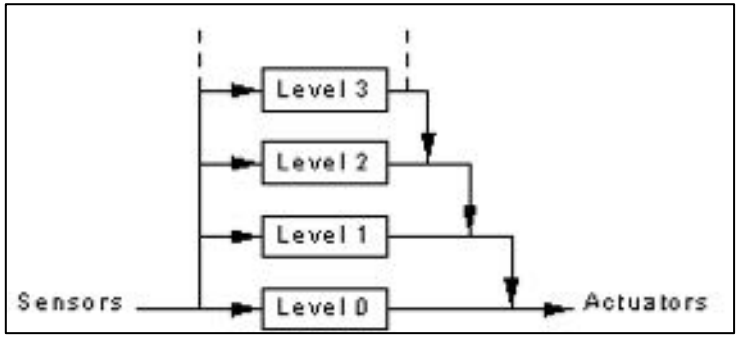




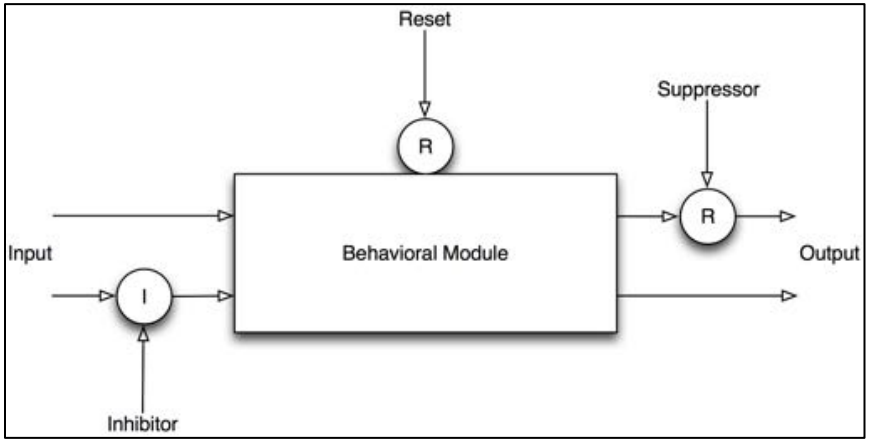
Brooks, R. (1985)

Subsumption Architecture

Overall architecture



Individual layer



- Overall structure consists of multiple layers
- Each layer works independently without others
- Layers can suppress the sensory input or inhibit motor output as necessary
- Layers can be extended or added anytime later

[Brooks, 1991]

Example: Robotic Vacuum Cleaner

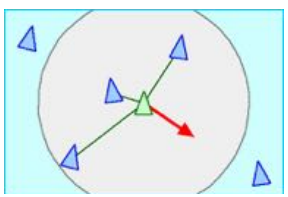
iRobot Roomba Clearning Robot

- First launched in 2002
- Sold over 10 million robots worldwide
- Sensors: bumper, infra-red proximity, cliff detector, dirt sensor, optic sensor
- Price: ca. £300-600

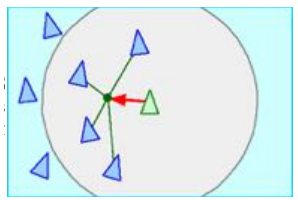
Example: Craig Reynolds Boids

Behavior of Boids depends on external states: Every agent has a “vision” to observe external states.

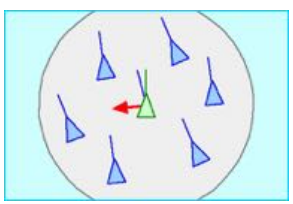
<u>5 Rules of Agent’s Motion</u>



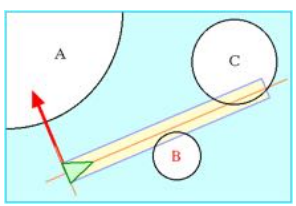
Separation



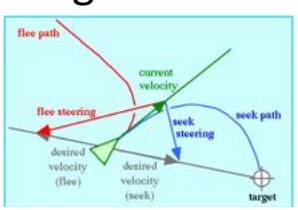
Cohesion



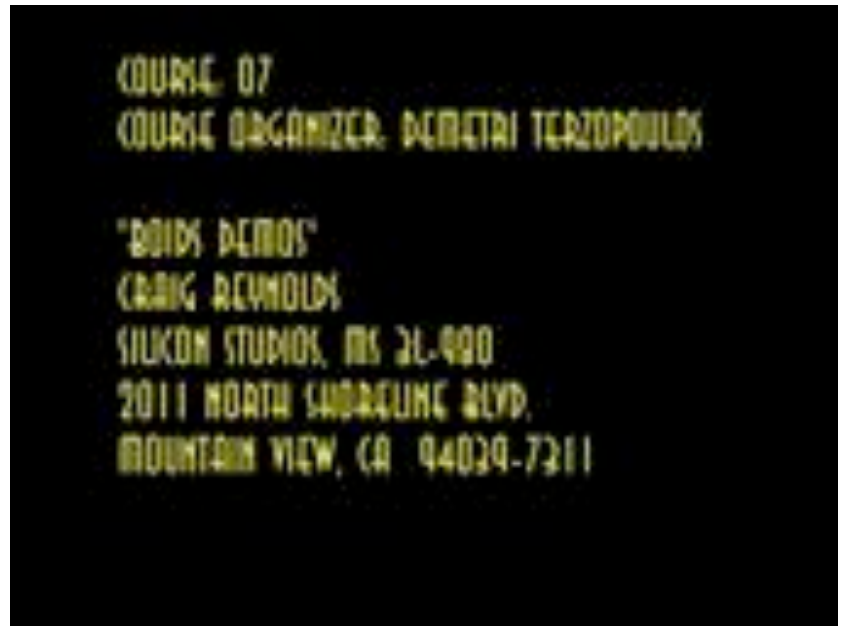
Alignment



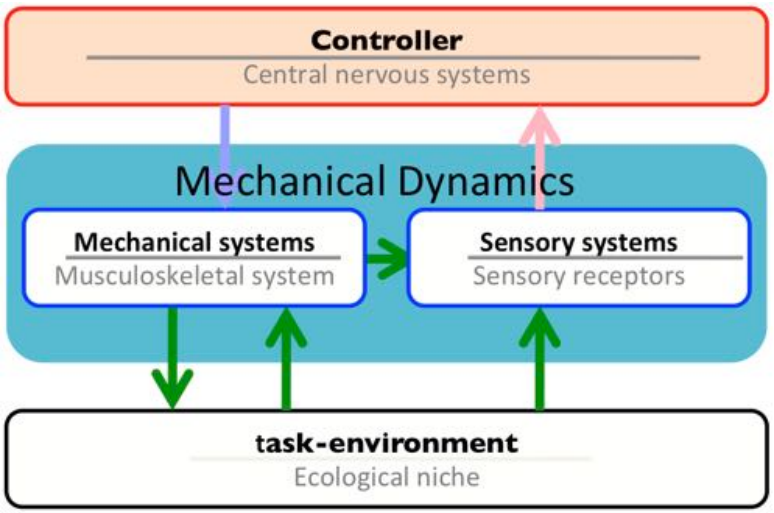
Obstacle avoidance

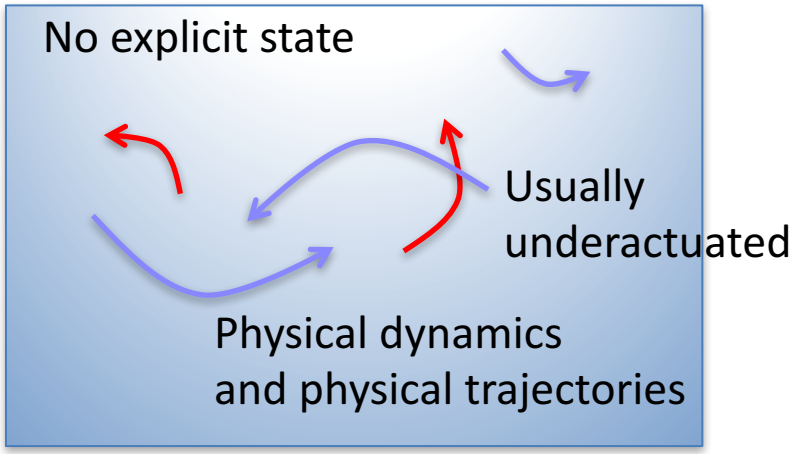


Goal seeking



3.4 Passivity-Based Architecture

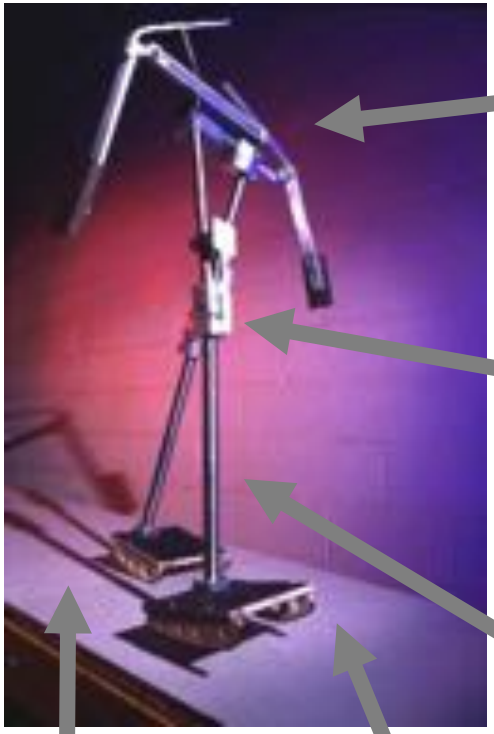




- Mechanical dynamics dominate behaviors of the systems
- Behaviors are often not fully controllable (i.e. underactuated systems)
- Often no state and no action are clearly defined in the controller

More details in Lecture 3

Example: Passive Dynamic Walker



No Motor
No Controller

Knee-lock mechanism

Pendulum dynamics

Slope

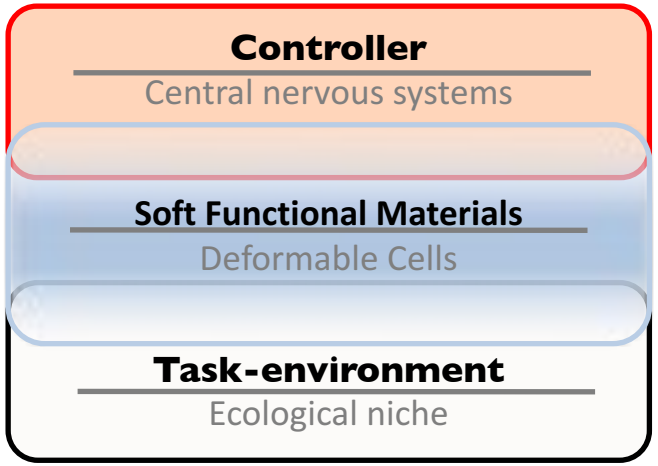
Ground friction

Passive Dynamic Walker, Cornell University

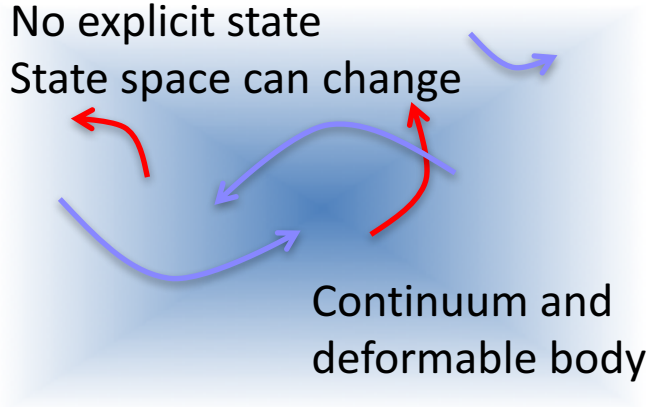
[Collins, et al. 2005]

3.5 Soft Robotics Approach

Functions are distributed in materials



No explicit state
State space can change

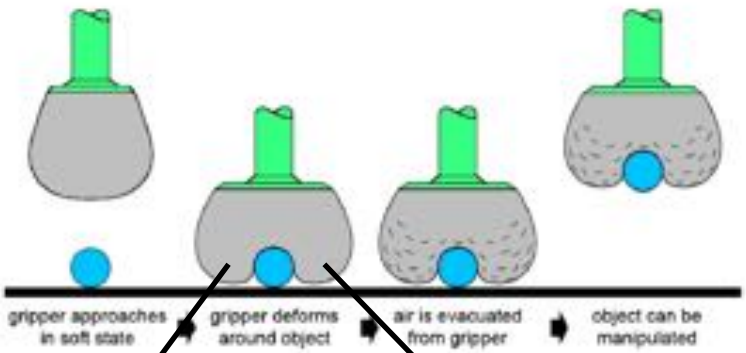


- No clear distinction between control, body, and environment
- All functions are results of physical/ material processes
- Continuum deformable body

[Pfeifer, et al. Communications of ACM, 2014]

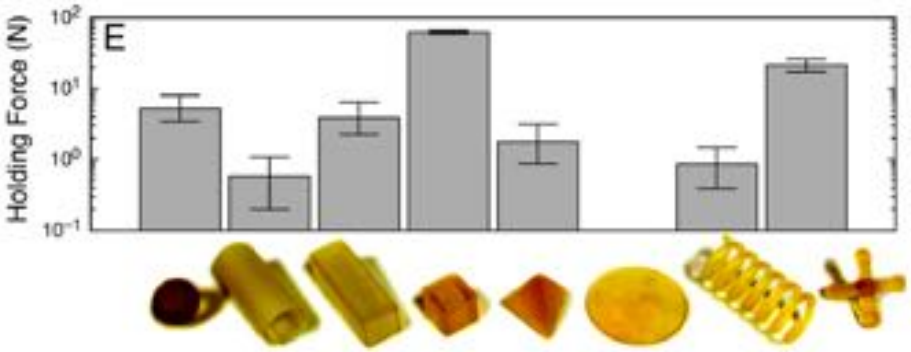
Universal Gripper

Physically Self-Adapting to Unknown Objects



Elastic bag
(radius = 4.3 [cm],
thickness = 0.3 [mm])

Inside:
granular
material
(ground coffee)



Brown et al (2010) *PNAS* 107 (44): 18809–18814

Take home messages

- Robot revolution is happening now but not all problems can be solved immediately
- Inverse-Kinematic Motion Control is the most powerful tool of robot applications. Try this first.
- Various other approaches exist for other challenges.

Thank you!

For publications, video, pictures:

Fumiya Iida
Bio-Inspired Robotics Laboratory
Dept. Engineering, Univ. of Cambridge

Email: fi224@cam.ac.uk
URL: http://divf.eng.cam.ac.uk/birl/

UNIVERSITY OF CAMBRIDGE

Bio-Inspired Robotics Lab