ECE 1675/2570: Robotic Control (Spring 2022)

Module II: Control of Mobile Robots

Lecture 5: Locomotion; Mobile Robot Control Architecture

February 9, 2022

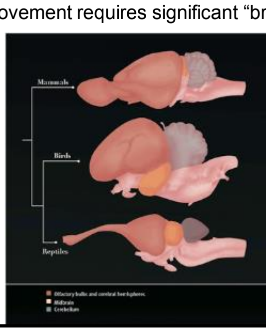
Zhi-Hong Mao
Professor of ECE and Bioengineering
University of Pittsburgh, Pittsburgh, PA

Outline

- Homework 3 (due Feb. 16)
- Locomotion
- Mobile robot components
- Behavior-based control of mobile robots
  - Go-to-goal behavior
  - Obstacle-avoidance behavior

Locomotion

- Introduction
  - Locomotion refers to the way a robot moves from place to place
  - In nature, movement requires significant “brain power”



Locomotion

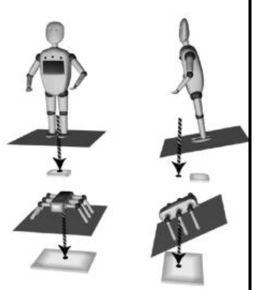
- Introduction
  - Locomotion refers to the way a robot moves from place to place
  - In nature, movement requires significant "brain power"
  - Moving the robot around becomes the first challenge for the robot "brains"—controllers
  - Effectors and actuators
    - *Legs*, for walking, crawling, climbing, jumping, etc.
    - *Wheels*, for rolling
    - *Arms*, for swinging, crawling, climbing, etc.
    - *Wings*, for flying
    - *Flippers*, for swimming

Note: Legged locomotion is more difficult than wheeled locomotion due to the challenge of stability and larger number of degrees of freedom.

4

Locomotion

- Introduction
- Stability
  - Static stability
    - A *statically stable* robot can stand still without falling over
    - The *center of gravity* of any body needs to be above *polygon of support*—the area that is covered by the ground points (legs or wheels)
    - If the robot can walk while staying balanced at all times, we call this *statically stable walking*



5

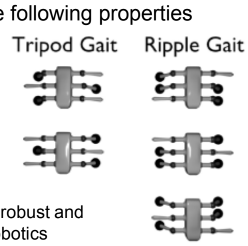
Locomotion

- Introduction
- Stability
  - Static stability
  - Dynamic stability
    - In *dynamic stability*, the body must actively balance or move to remain stable
    - A statically stable robot can use dynamically stable walking patterns, in order to be fast and efficient

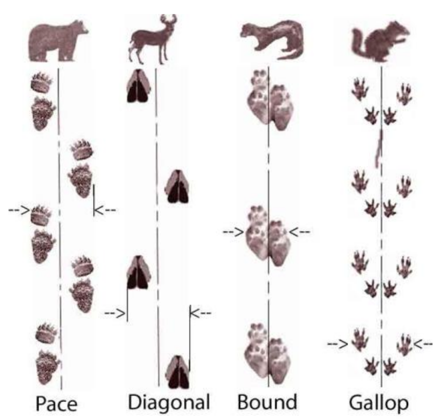
6

Locomotion

- Introduction
- Stability
- Moving and gaits
  - A *gait* is the particular way a robot moves, including the order in which it lifts and lowers its legs and places its feet on the ground
  - Desirable robot gaits have the following properties
    - *Stability*
    - *Speed*
    - *Energy efficiency*
    - *Robustness*
    - *Simplicity*



Note: Six-legged walking is highly robust and therefore common in nature and robotics



8

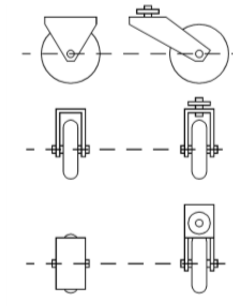
Locomotion

- Introduction
- Stability
- Moving and gaits
- Wheels and steering
  - Wheels in robotics
    - Wheel-like structures are very rare in biological locomotion (but do appear in certain bacteria)
    - Because of their efficiency and comparative simplicity of control, wheels are the major choice of locomotion effector in robotics
    - Wheeled robots are usually designed to be *statically stable*, but they are not necessarily *holonomic* (meaning they cannot control all of their available degrees of freedom)

9

Locomotion

- Introduction
- Stability
- Moving and gaits
- Wheels and steering
  - Wheels in robotics
  - Wheel types
    - *Standard wheel:* two degrees of freedom; rotation around the wheel axle and the contact point
    - *Castor wheel:* two degrees of freedom; rotation around the wheel axle and an offset steering joint



10

Locomotion

- Introduction
- Stability
- Moving and gaits
- Wheels and steering
  - Wheels in robotics
  - Wheel types
    - *Standard wheel*
    - *Castor wheel*
    - *Swedish* or *Mecanum wheel:* three degrees of freedom; rotation around the wheel axle, around the rollers, and around the contact point

11

Locomotion

- Introduction
- Stability
- Moving and gaits
- Wheels and steering
  - Wheels in robotics
  - Wheel types
    - *Standard wheel*
    - *Castor wheel*
    - *Swedish or Mecanum wheel*
    - *Ball* or *spherical wheel:* realization technically difficult

12

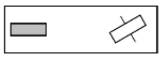
Locomotion

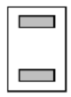
- Introduction
- Stability
- Moving and gaits
- **Wheels and steering**
  - Wheels in robotics
  - Wheel types
  - **Wheel configurations**
    - Two wheels

1st wheel for steering

Segway

Bicycle





Rear wheel for power

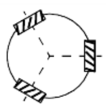
13

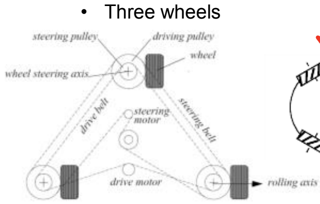
cleaning robot

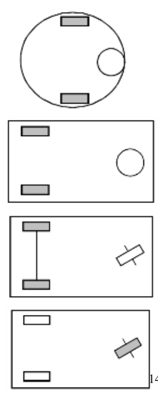
Locomotion

- Introduction
- Stability
- Moving and gaits
- **Wheels and steering**
  - Wheels in robotics
  - Wheel types
  - **Wheel configurations**
    - Two wheels
    - Three wheels

Swedish wheel



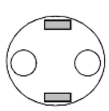


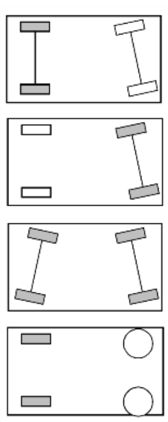


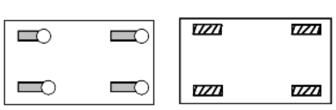
14

Locomotion

- Introduction
- Stability
- Moving and gaits
- **Wheels and steering**
  - Wheels in robotics
  - Wheel types
  - **Wheel configurations**
    - Two wheels
    - Three wheels
    - Four Wheels







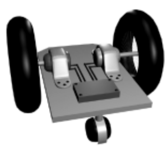
Holonomic robot - harder to control
→ Underactuated robots are easier to work with

Locomotion

- Introduction
- Stability
- Moving and gaits
- Wheels and steering
  - Wheels in robotics
  - Wheel types
  - Wheel configurations
  - Maneuverability and "controllability"
    - There is generally an *inverse correlation* between maneuverability and "controllability" (e.g., controlling an omnidirectional robot for a specific direction of travel is more difficult and often less accurate when compared to less maneuverable designs)

16

Locomotion



- Introduction
- Stability
- Moving and gaits
- Wheels and steering
  - Wheels in robotics
  - Wheel types
  - Wheel configurations
  - Maneuverability and "controllability"
  - Differential drive and differential steering
    - The ability to drive wheels separately and independently, through the use of separate motors, is called a *differential drive*
    - Being able to steer wheels independently is called *differential steering*

17

Locomotion

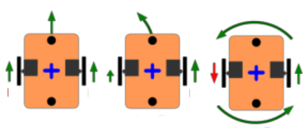
- Introduction
- Stability
- Moving and gaits
- Wheels and steering
- Navigation
  - *Navigation* refers to the problem of moving the robot's whole body to various destinations
  - Main issues
    - *Localization* and *map building:* to figure out where the robot is relative to some model of the environment
    - *Path planning*, also called *trajectory planning:* to search through all possible trajectories and evaluate them, in order to find one that will satisfy the requirements
    - *Motion control:* to get the robot to a particular location

Dynamics to follow path planning

18

Mobile robot components

- Body and dynamics
  - Model for a differential drive wheeled robot

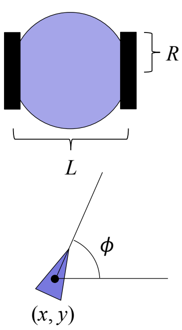


19

Mobile robot components

- Body and dynamics
  - Model for a differential drive wheeled robot

State space model



$$\dot{x} = \frac{R}{2}(v_r + v_l)\cos\phi$$
$$\dot{y} = \frac{R}{2}(v_r + v_l)\sin\phi$$
$$\dot{\phi} = \frac{R}{L}(v_r - v_l)$$

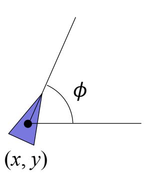
20

State Space: x is function of (x, u)

Inputs are v_R and v_L (angular velocity at each wheel)

Mobile robot components

- Body and dynamics
  - Model for a differential drive wheeled robot
  - The “unicycle” model
    - It is not very natural to “think” in terms of wheel velocities
    - Instead, go directly for translational velocity (v) and angular velocity (ω)



$$\dot{x} = v\cos\phi$$
$$\dot{y} = v\sin\phi$$
$$\dot{\phi} = \omega$$

21

Inputs: v, ω

ω: "heading's" angular velocity

Mobile robot components

- Body and dynamics
  - Model for a differential drive wheeled robot
  - The “unicycle” model
    - It is not very natural to “think” in terms of wheel velocities
    - Instead, go directly for translational velocity (v) and angular velocity (ω)
    - ***Idea of design:*** design for <u>the “unicycle” model</u> and then implement <u>the differential drive model</u>

$$\boxed{\begin{aligned}\dot{x} &= \frac{R}{2}(v_r + v_l)\cos\phi \\ \dot{y} &= \frac{R}{2}(v_r + v_l)\sin\phi \\ \dot{\phi} &= \frac{R}{L}(v_r - v_l)\end{aligned}}$$

$$v = \frac{R}{2}(v_r + v_l)$$
$$\omega = \frac{R}{L}(v_r - v_l)$$

$$\longleftarrow \quad \longrightarrow$$

$$v_r = \frac{2v + \omega L}{2R}$$
$$v_l = \frac{2v - \omega L}{2R}$$

$$\boxed{\begin{aligned}\dot{x} &= v\cos\phi \\ \dot{y} &= v\sin\phi \\ \dot{\phi} &= \omega\end{aligned}}$$

22

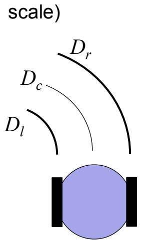
Mobile robot components

- Body and dynamics
- Sensors, actuators, and power
  - State estimation
    - The state of the robot is $[x, y, \phi]^T$
    - Coupling of proprioceptive and exteroceptive sensors for state estimation
    - Proprioceptive sensors
      - Orientation: compass
      - Position: accelerometers, gyroscopes,…
      - Wheel encoders
    - Exteroceptive sensors
      - A “skirt” of range sensors: IR, ultra-sound, LIDAR, …
      - Other standard exteroceptive sensors include vision, tactile, and “GPS”

23

Mobile robot components

- Body and dynamics
- Sensors, actuators, and power
  - State estimation
  - Example of proprioceptive sensing: wheel encoders
    - Wheel encoders give the distance moved by each wheel
    - Assume the wheels are following an arc (short time scale)



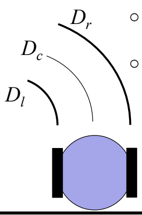
$$D_c = \frac{D_l + D_r}{2}$$

$$\boxed{\begin{aligned}x' &= x + D_c\cos(\phi) \\ y' &= y + D_c\sin(\phi) \\ \phi' &= \phi + \frac{D_r - D_l}{L}\end{aligned}}$$

24

Mobile robot components

- Body and dynamics
- Sensors, actuators, and power
  - State estimation
  - Example of proprioceptive sensing: wheel encoders
    - Wheel encoders give the distance moved by each wheel
    - Assume the wheels are following an arc (short time scale)
    - How do we know how far each wheel has moved?
      - Assume each wheel has N "ticks" per revolution
      - Most wheel encoders give the total tick count since the beginning
      - For both wheels: $\Delta\text{tick} = \text{tick}' - \text{tick}$



$$D_l = 2\pi R \frac{\Delta\text{tick}_l}{N}$$
$$D_r = 2\pi R \frac{\Delta\text{tick}_r}{N}$$

25

Mobile robot components

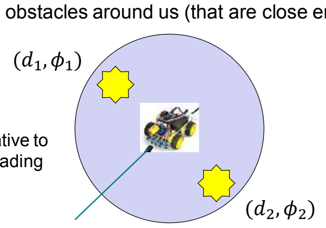
- Body and dynamics
- Sensors, actuators, and power
  - State estimation
  - Example of proprioceptive sensing: wheel encoders
    - Wheel encoders give the distance moved by each wheel
    - Assume the wheels are following an arc (short time scale)
    - How do we know how far each wheel has moved?
    - A major issue: **DRIFT!**

26

Mobile robot components

- Body and dynamics
- Sensors, actuators, and power
  - State estimation
  - Example of proprioceptive sensing: wheel encoders
  - Example of exteroceptive sensing: disk abstraction
    - Assumption: instead of worrying about the resolution of the sensors, assume we know the distance and direction to all obstacles around us (that are close enough)

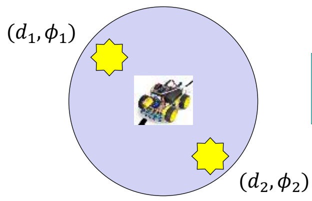
Angles are relative to the robot's heading



27

Mobile robot components

- Body and dynamics
- Sensors, actuators, and power
  - State estimation
  - Example of proprioceptive sensing: wheel encoders
  - Example of exteroceptive sensing: disk abstraction
    - Assumption
    - Coordinates of the obstacles (if we know our own position and orientation)



$$x_i = x + d_i \cos(\phi_i + \phi)$$
$$y_i = y + d_i \sin(\phi_i + \phi)$$
$$i = 1,2$$

28

Mobile robot components

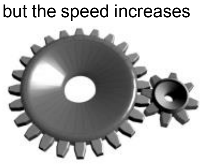
- Body and dynamics
- Sensors, actuators, and power
  - State estimation
  - Example of proprioceptive sensing: wheel encoders
  - Example of exteroceptive sensing: disk abstraction
    - Assumption
    - Coordinates of the obstacles
    - A case study

Multi-robot rendezvous task
(Li, Sun, and Yang, 2013)

29

Mobile robot components

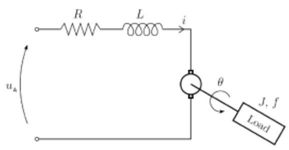
- Body and dynamics
- Sensors, actuators, and power
  - State estimation
  - Example of proprioceptive sensing: wheel encoders
  - Example of exteroceptive sensing: disk abstraction
  - Actuation
    - Gearing
      - If the output gear is larger than the input gear, the torque increases but the speed decreases
      - If the output gear is smaller than the input gear, the torque decreases but the speed increases



30

Mobile robot components

- Body and dynamics
- Sensors, actuators, and power
  - State estimation
  - Example of proprioceptive sensing: wheel encoders
  - Example of exteroceptive sensing: disk abstraction
  - Actuation
    - Gearing
    - Dynamics and time constant of a DC motor
    - About model order reduction



$$u_A(t) = L\frac{di(t)}{dt} + Ri(t) + k_v\omega(t)$$
$$k_a i(t) = J\dot{\omega}(t) + f\omega(t)$$

31

Mobile robot components

- Body and dynamics
- Sensors, actuators, and power
  - State estimation
  - Example of proprioceptive sensing: wheel encoders
  - Example of exteroceptive sensing: disk abstraction
  - Actuation
  - Power management

32

Control Architectures:

Reactive Behavior-based

Deliberative Hybrid

Mobile robot components

- Body and dynamics
- Sensors, actuators, and power
- Control architecture: behavior-based control
  - Motivation
    - The world is fundamentally unknown and changing
    - It does not make sense to over-plan
    - *Key idea:* develop a library of useful controllers (behaviors) and switch among controllers in response to environmental changes

33

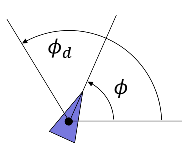
Mobile robot components

- Body and dynamics
- Sensors, actuators, and power
- Control architecture: behavior-based control
  - Motivation
  - Behaviors
    - Go-to-goal behavior
    - Avoid-obstacles behavior
    - Follow-wall behavior
    - Track-target behavior
    - …

34

Mobile robot components

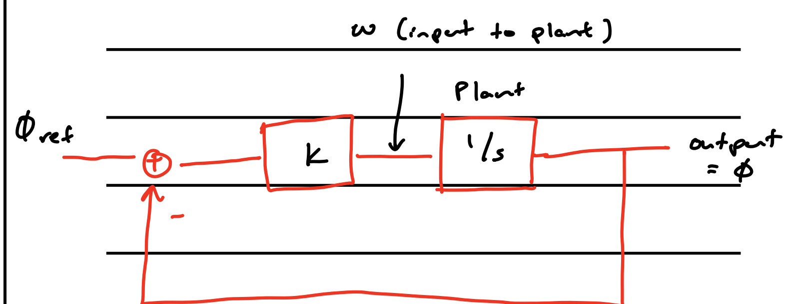
- Body and dynamics
- Sensors, actuators, and power
- Control architecture: behavior-based control
  - Motivation
  - Behaviors
  - Example of a behavior
    - Assume we have a differential-drive wheeled mobile robot driving at a constant speed
    - Want to drive in a desired heading



$$\dot{x} = v_0 \cos\phi$$
$$\dot{y} = v_0 \sin\phi$$
$$\dot{\phi} = \omega$$

35

input : ω output : ϕ



1/s because it is LT of diff eq.

Don't need PI because system itself is integrator

Mobile robot components

- Body and dynamics
- Sensors, actuators, and power
- Control architecture: behavior-based control
  - Motivation
  - Behaviors
  - Example of a behavior
    - Assume we have a differential-drive wheeled mobile robot driving at a constant speed
    - Want to drive in a desired heading
    - Use PID control

$r = \phi_d$

$e = \phi_d - \phi$

$\dot{\phi} = \omega$

$$\omega(t) = k_P e(t) + k_I \int_0^t e(\tau) d\tau + k_D \frac{de(t)}{dt}$$

or

$$\Omega(s) = (k_P + \frac{k_I}{s} + k_D s) E(s)$$

36

Slide 35:

$$T(s) = \frac{K/s}{1+K/s} = \frac{K}{s+K}$$ (1st order system)

DC Gain = 1 (replace s with 0)

$e_{ss} = 0$

$\tau = 1/K$

Mobile robot components

- Body and dynamics
- Sensors, actuators, and power
- Control architecture: behavior-based control
  - Motivation
  - Behaviors
  - Example of a behavior
    - Assume we have a differential-drive wheeled mobile robot driving at a constant speed
    - Want to drive in a desired heading
    - Use PID control

Caution: This typically will not work as we are dealing with angles!

$$\phi_d = 0, \qquad \phi = 100\pi \quad \Rightarrow \quad e = -100\pi$$

37

Robot may spin like crazy

Mobile robot components

- Body and dynamics
- Sensors, actuators, and power
- Control architecture: behavior-based control
  - Motivation
  - Behaviors
  - Example of a behavior
    - Assume we have a differential-drive wheeled mobile robot driving at a constant speed
    - Want to drive in a desired heading
    - Use PID control

Caution: This typically will not work as we are dealing with angles!

Solution: Ensure that $e \in (-\pi, \pi]$. Standard trick is to use atan2: (MATLAB)

$$\boxed{e' = \operatorname{atan2}(\sin(e), \cos(e)) \in (-\pi, \pi]}$$

38

Go-to-goal behavior

- How to drive a robot to a goal location?

$$\dot{x} = v_0 \cos\phi \qquad r = \phi_d$$
$$\dot{y} = v_0 \sin\phi \qquad e = \phi_d - \phi$$
$$\dot{\phi} = \omega \qquad \omega = \mathrm{PID}(e')$$

Start location: (x, y)
Goal location: (x_g, y_g)

$$\boxed{\phi_d = \arctan\left(\frac{y_g - y}{x_g - x}\right)}$$

39

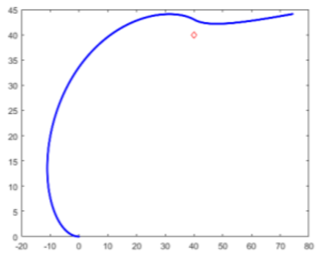
Go-to-goal behavior

- How to drive a robot to a goal location?
- Attempt 1

$$\phi_d = \arctan\left(\frac{y_g - y}{x_g - x}\right)$$

$$e = \phi_d - \phi$$

$$\omega = k_P e$$



40

Didn't use arctan2 → robot goes well past target

Not consistent with our estimation because it doesn't know difference between 0° and 180°

Go-to-goal behavior

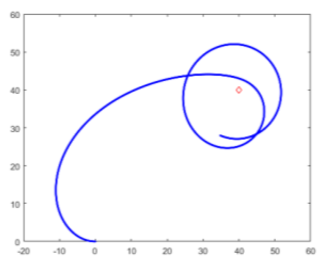
- How to drive a robot to a goal location?
- Attempt 1
- Attempt 2

$$\phi_d = \text{atan2}\,(y_g - y, x_g - x)$$

$$e = \phi_d - \phi$$

$$e' = \text{atan2}(\sin(e), \cos(e))$$

$$\omega = k_P e'$$



41

Still not what we want

There is also an outer loop that converts $(x, y) \longleftrightarrow \phi$

ϕ_d is not unit step because it depends on relative position of robot

ϕ_d is a ramp

Go-to-goal behavior

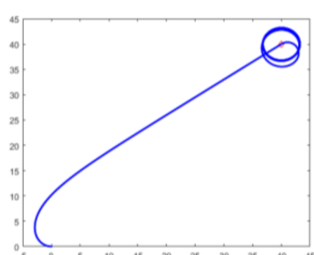
- How to drive a robot to a goal location?
- Attempt 1
- Attempt 2
- Attempt 3

$$\phi_d = \text{atan2}\,(y_g - y, x_g - x)$$

$$e = \phi_d - \phi$$

$$e' = \text{atan2}(\sin(e), \cos(e))$$

$$\omega = k_{Big} e'$$

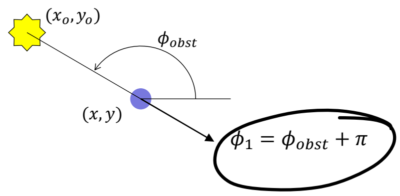


42

Error is getting smaller

Obstacle-avoidance behavior

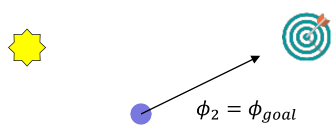
- How to avoid driving into obstacles?
  - We can use the same idea by defining a desired heading
    - “Pure” avoidance



43

Obstacle-avoidance behavior

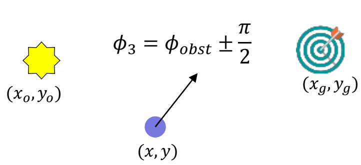
- How to avoid driving into obstacles?
  - We can use the same idea by defining a desired heading
    - “Pure” avoidance
    - “Pure” go-to-goal



44

Obstacle-avoidance behavior

- How to avoid driving into obstacles?
  - We can use the same idea by defining a desired heading
    - “Pure” avoidance
    - “Pure” go-to-goal
    - Not “pure” but “blended” (choice of + or – depends on direction to goal)

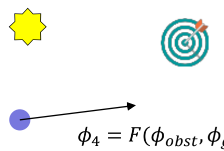


45

Sensor measures both goal and obstacle - how to combine $\phi_{obstacle}$ and ϕ_{goal}

Obstacle-avoidance behavior

- How to avoid driving into obstacles?
  - We can use the same idea by defining a desired heading
    - “Pure” avoidance
    - “Pure” go-to-goal
    - Not “pure” but “blended”



$\phi_4 = F(\phi_{obst}, \phi_{goal})$

46

Obstacle-avoidance behavior

- How to avoid driving into obstacles?
- Behavior coordination
  - The previous example illustrates two fundamentally different mechanisms for behavior coordination
    - Arbitration (winner-takes-all, hard switches)
    - Fusion (blending, combining behaviors)
  - Both approaches have merit in different situations
    - Performance
    - Analysis
  - We will see how to design systematic behaviors and behavior coordination mechanisms in the following lectures of Module II

47

References (I)

- M. Egerstedt, Lecture Notes for Control of Mobile Robots, Georgia Institute of Technology, 2016.
- K. J. Astrom and R. M. Murray, Feedback Systems: An Introduction for Scientists and Engineers, 2010.
- M. Mataric, The Robotics Primer, The MIT Press, 2007.
- R. Siegwart and I. R. Nourbakhsh, Introduction to Autonomous Mobile Robots, The MIT Press, 2004.
- C. L. Phillips and R. D. Harbor, Feedback Control Systems, 4th Edition, Prentice Hall, 2000.
- X. Li, D. Sun, and J. Yang, “Preserving multirobot connectivity in rendezvous tasks in the presence of obstacles with bounded control input,” IEEE Trans. Control Systems Technology 21(6), 2306-2314, Nov. 2013.
- https://42bots.com/tutorials/differential-steering-with-continuous-rotation-servos-and-arduino/
- https://ecowellness.wordpress.com/2012/02/20/animal-tracking-part-2-common-gait-patterns/

48

References (II)

- https://en.wikipedia.org/wiki/Mecanum_wheel
- https://en.wikipedia.org/wiki/Rotating_locomotion_in_living_systems
- https://howbirdsthink.blogspot.com/2012/09/bird-brains-are-different.html
- https://mobilerobotguide.com/2018/04/10/power-management-for-autonomous-mobile-robots/
- https://www.crazyengineers.com/threads/why-cant-the-spherical-wheels-be-used-in-automobiles.71189
- https://www.instructables.com/id/Making-a-Modular-Differential-Drive-Robot/

49