Robot Control

EE468/CE468: Mobile Robotics

Dr. Basit Memon

Electrical and Computer Engineering
Habib University

September 18, 2023

Table of Contents

Table of Contents

Organization of autonomy across nested think-plan-act cycles. [2]

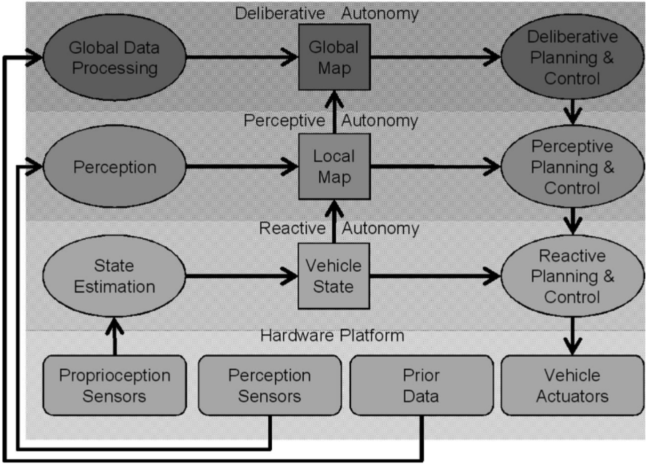


Figure 1.9 Layers of Autonomy. The entire mobile system can be described in terms of three nested perceive-think-act loops.

Definitions of autonomy layers. [2, 1.3.2]

- **Reactive Autonomy.** Responsible for controlling the motion of robot with respect to the environment. Requires motion state (position, velocity, heading, etc.) Examples: Follow a trajectory, Go to a desired pose, etc.

- **Perceptive Autonomy.** Responsible for responding to immediately perceivable environment. Requires feedback of state of the environment. Examples: Obstacle avoidance, Wall following, Line following, etc.

- **Deliberative Autonomy.** Responsible for achieving long-term goals, i.e. mission. Requires global position estimate and prediction extend far into future. Examples: Localization, Mapping, etc.

- Control exists at all layers.
- This module is about low-level control.
- Reactive autonomy and some perceptive autonomy.

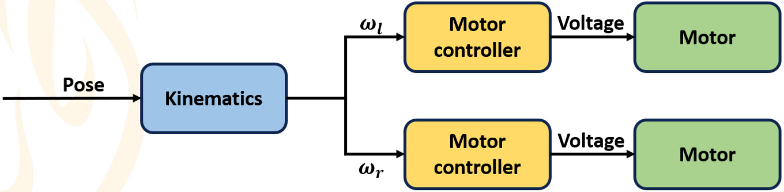


Figure: Motor controller ensures motor speeds and generates voltage commands.

Divide & Conquer: Organize control in terms of simple behaviors.

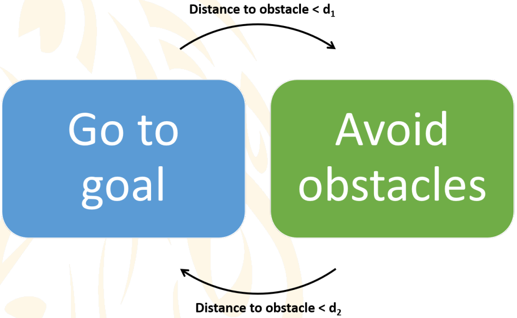


Figure: Switching between behaviors based on guard conditions.

- Go to goal
- Avoid obstacles
- Follow wall
- Track target
- ...
- Switch between behaviors as needed

Pure reactive control connects sense directly to act.

- Inspired from biology.
- Walter Grey's Tortoises (1949)

Braitenberg Vehicles

Figure: Simulator

MIT Braitenberg Creatures

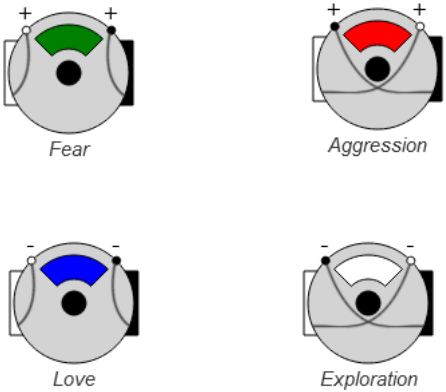


Figure: Four original modes of Braitenberg vehicles

Table of Contents

- We need control for precise motion despite modeling errors and perturbations.

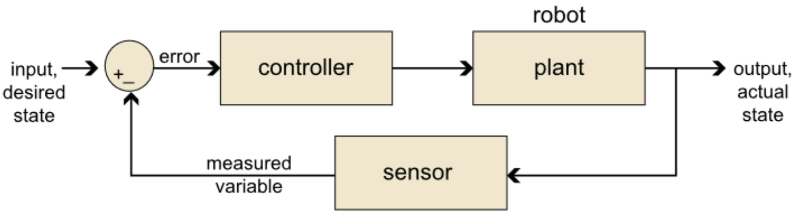


Figure: Basic model of feedback control [1]

Glossary of terms

- **Reference Signal or Desired state (y_r):** It specifies goal or target for the control system.
- **Output Signal (y):** What is the system actually doing.
- **Error Signal (e):** Difference between desired and actual state, $e = y_r - y$.
- **Feedback Controller or Closed Loop:** Control signal is based on error signal.
- **Plant:** Robot body (system) plus actuators.
- **Control Input (u):** Component(s) of system we can control.

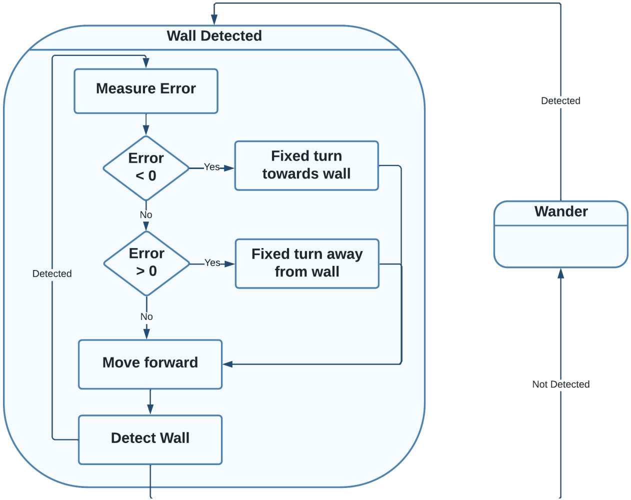
Create a wall-following robot [1]

- Maintain a distance of 5 in from a wall.
- If robot loses the wall, it randomly wanders till it detects the wall.

What will be open-loop control strategy?

- Given model of world (robot + environment), decide a path, determine wheel control commands using inverse kinematics, execute.

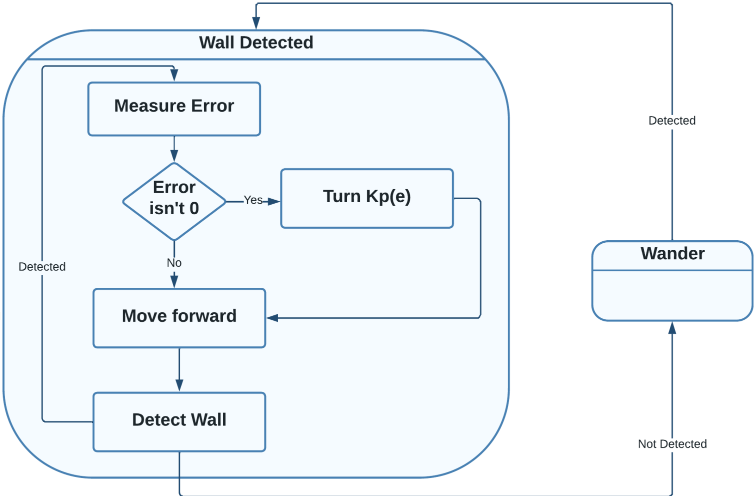
- Will it work?
  - Only if we had a perfect world model.
  - Modeling errors, Parameter changes, Unwanted inputs


Wall Detected
Measure Error
Error < 0
Yes
Fixed turn towards wall
No
Error > 0
Yes
Fixed turn away from wall
No
Move forward
Detect Wall
Detected
Detected
Wander
Not Detected

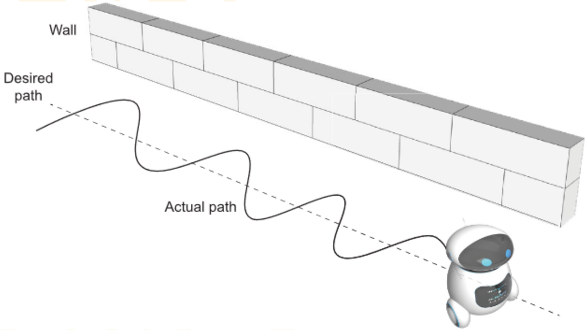
This is bang-bang or on-off control.

- Difficult to maintain exact distance.
- Odometry errors, Sensor errors, Wheel slippage cause robot to overturn.
- Oscillations may make system unstable.

How can we do better?

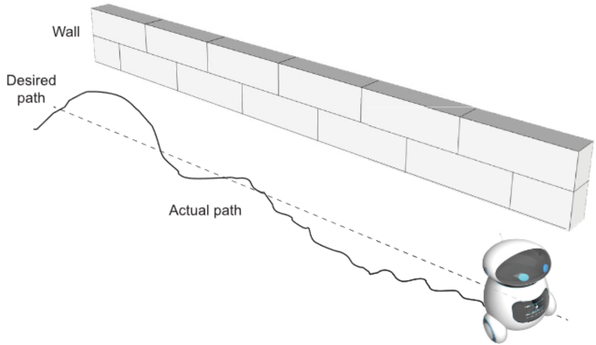


This is proportional control.



- Amount of turn is proportional to error.
- We still have oscillations.
- One way to reduce them is to introduce deadband - no action is taken for error values in this band.

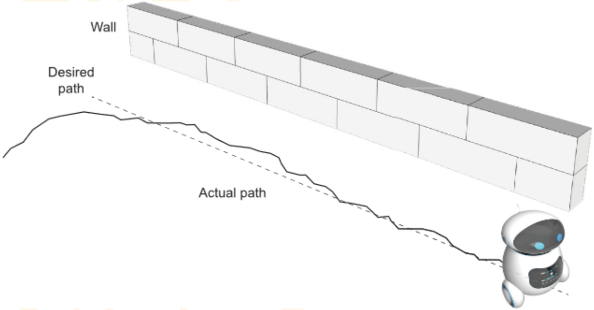
How can we decrease the oscillations?



- Add derivative control. PD control is:

$$u(t) = K_p\, e(t) + K_d\, \frac{de(t)}{dt}.$$

- Slows down rate of change in position as robot gets closer to desired state.
- Decreases sharpness of turns and reduces overshoot.
- Still has steady-state error. The rate of change of error and error are too small to remove this.



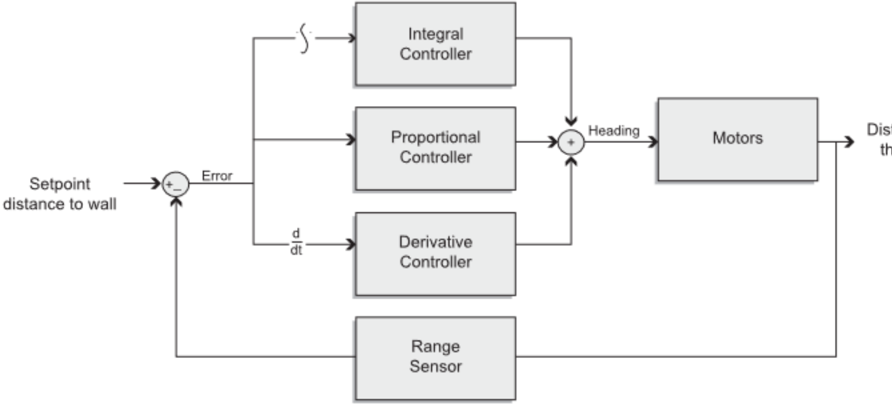
- Adding integral terms removes steady-state error:

$$u(t) = K_p\, e(t) + K_d \frac{de(t)}{dt} + K_i \int e(s)ds.$$

Possible problems with D and I

- **Issues with differential term**
  - Sudden change in the reference or current state (e.g. brakes) causes abrupt unwanted changes in control.
  - Current state measurements are noisy, resulting in high derivatives.
- **Issues with integral term**
  - *Windup:* The integral term is non-zero even when the desired state is achieved.

PID controller is sending heading commands down to motors!



Motion control for robots is structured hierarchically. [2, Chapter 7]

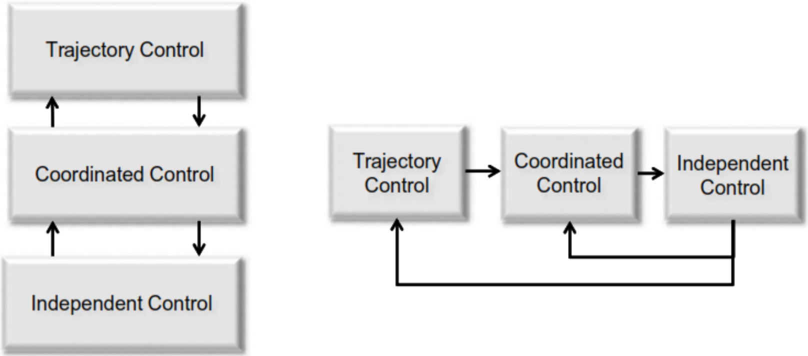


Figure 7.3 A Controller Hierarchy. Left: Higher levels produce the reference signals for lower levels. Right: The equivalent block diagram is a cascade configuration.

Trajectory control

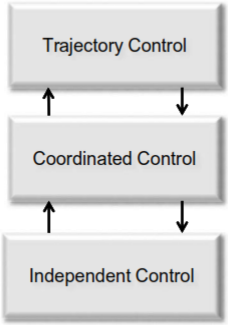


Figure: Hierarchical Control

- Causes robot to follow a trajectory over a period of time.
- Requires observations or predictions of the robot with respect to the environment.
- Examples: Path following, Move robot to specified pose, Follow leader robot, Line following

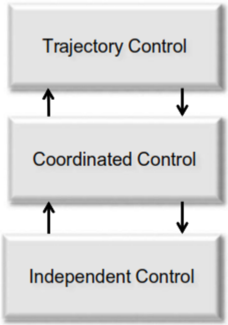


Figure: Hierarchical Control

- Instantaneous control of entire robot as an entity.
- MIMO: Consistent control of independent dof.

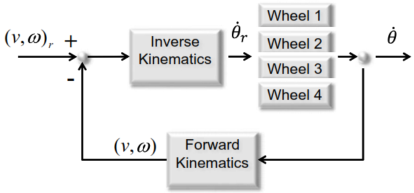


Figure: Wheels are coordinated to produce desire robot linear and angular velocity.

Independent control

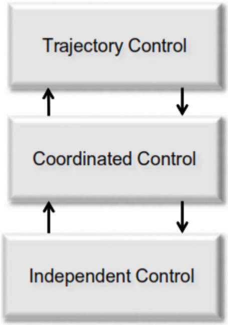


Figure: Hierarchical Control

- SISO: Controls a single degree of freedom.
- Relies only on sensing connected to that DOF.
- Connected directly to actuators.
- Example: Controllers for motor of differential drive that accepts wheel velocity and converts to torque/voltage signals.

Table of Contents

Feedforward control makes system reach goal faster.

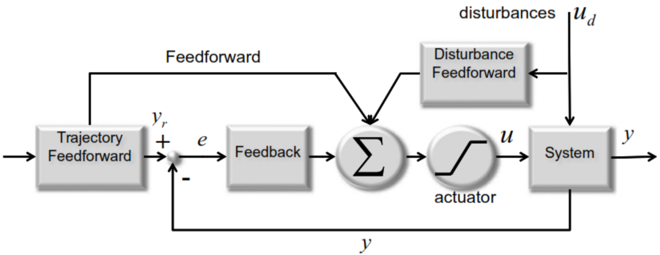


Figure 7.2 Generic Controller Block Diagram. This diagram summarizes most cases of interest.

Glossary of terms

- **Disturbance (u_d):** System inputs we cannot control.
- Actuators have saturation limits.
- **Feedforward Control or Open loop:** Control signal is based on system model.

Table of Contents

Mathematical model of system for control

- Recall that continuous-time control requires a differential model of the system, which is called *dynamical system model* in control literature.
- The word *dynamical* here simply refers to time-evolving nature of system captured by differential equations.
- We can use kinematic model of robot (also differential equations) for control, and don't require model to include forces (dynamics model).
- Kinematic model is sufficient, if we have good velocity control and dynamic effects are not vital.

Differential Drive

$$\dot{q} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} \frac{r}{2}\cos\phi & \frac{r}{2}\cos\phi \\ \frac{r}{2}\sin\phi & \frac{r}{2}\sin\phi \\ -r/L & r/L \end{bmatrix} \begin{bmatrix} u_L \\ u_R \end{bmatrix}$$

- We can rewrite it as:

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

if

$$u_L = \frac{2v - \omega l}{2r} \text{ and } u_R = \frac{2v + \omega l}{2r}.$$

Unicycle Model

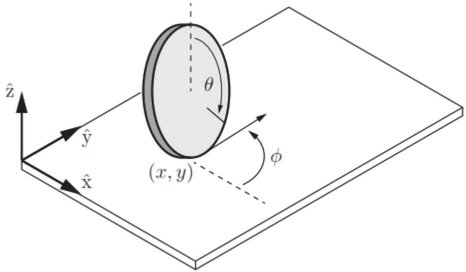


Figure: Wheel rolling without slipping

$$\dot{q} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} r\cos\phi & 0 \\ r\sin\phi & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

- u_1 is the wheel's driving speed and u_2 is heading direction turning speed.

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

Car-like robot (Ackermann Steering)

- The model for car can also be written as:

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

with the expressions

$$v = v$$
$$\psi = \arctan\left(\frac{\omega d}{v}\right)$$

converting the controls (v, ω) to actual controls (v, ψ).

- **Assumption:** We're able to measure the controlled variables, typically position and orientation of the robot, with respect to either a fixed frame or a path that the robot should follow.
- **Assumption:** Observations are continuous in time.
- **Assumption:** Observations are not corrupted by noise.

Table of Contents

[1] **Carolotta Berry.**
Mobile robotics for multidisciplinary study.
Morgan & Claypool Publishers, 2012.

[2] **Alonzo Kelly.**
Mobile robotics: mathematics, models, and methods.
Cambridge University Press, 2013.