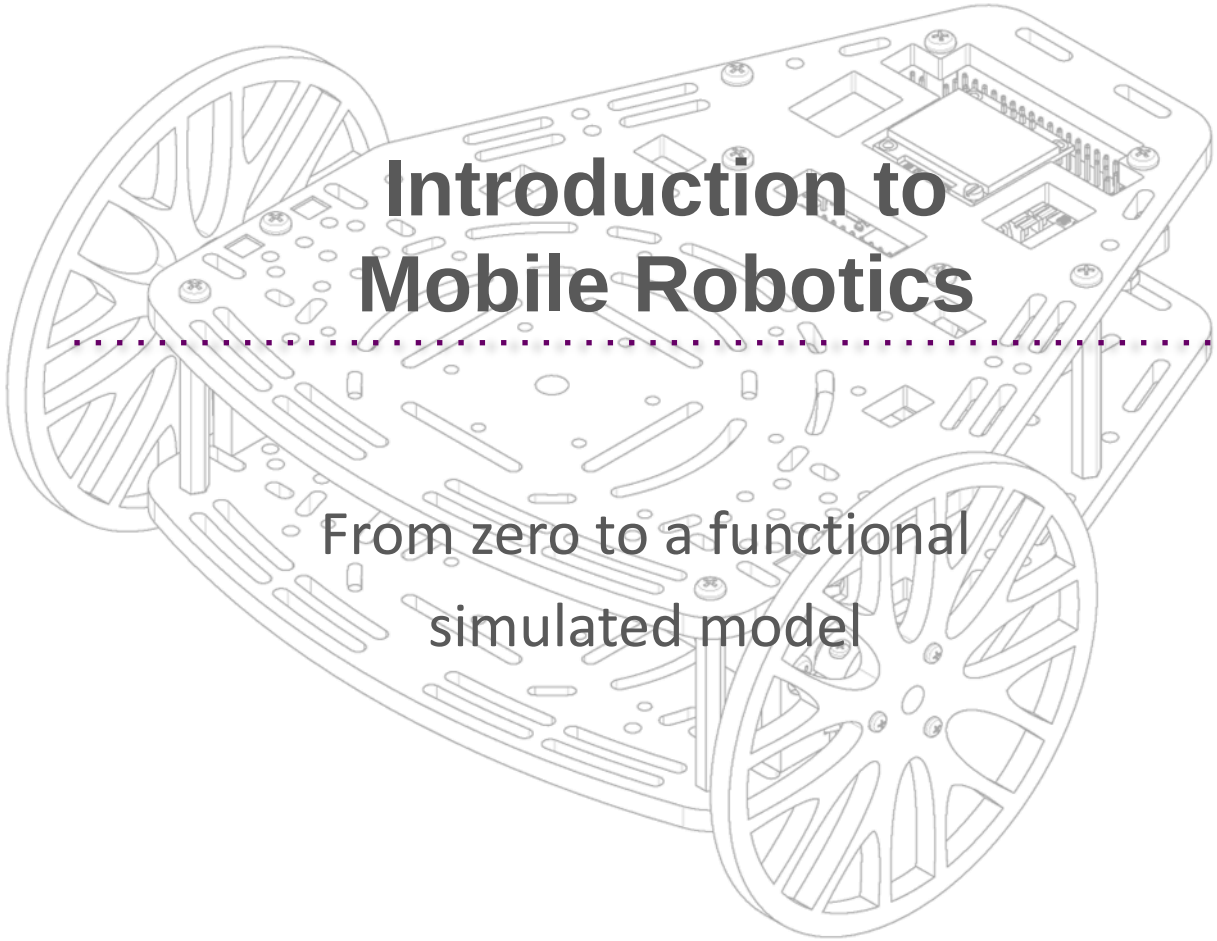


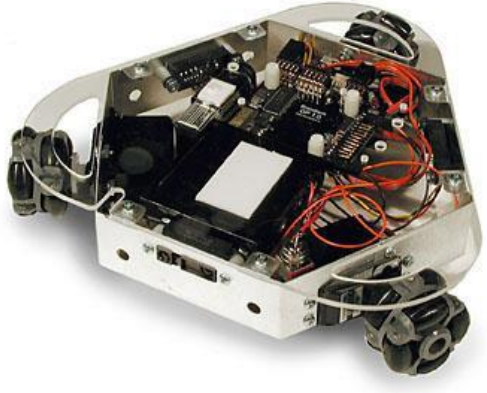
Introduction to Mobile Robotics

From zero to a functional simulated model

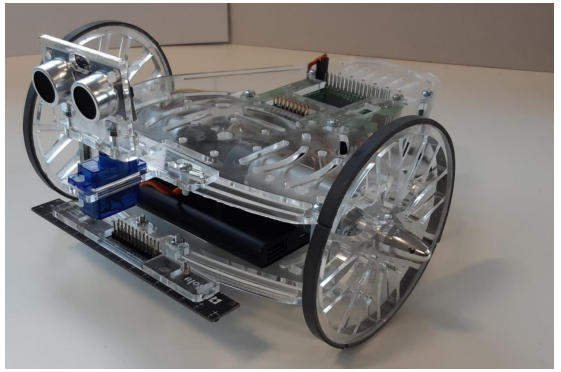
Mobile Robots

- There exists many types of wheeled robotic platforms
  - Differential-Drive robots
  - Omnidirectional robots
  - Ackermann-steering robots
  - and many others...

- In this course we will focus on differential drive robots, also known as “differential wheeled robots”.



Holonomic Robot
Acroname ©.



Differential-drive
Puzzlebot ©.

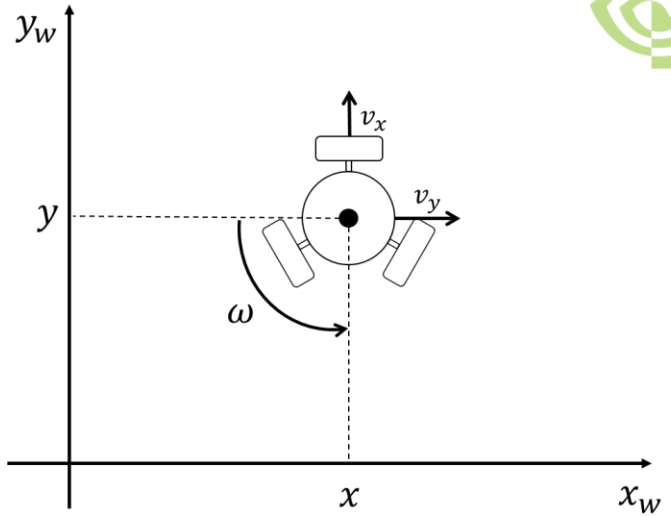
Holonomic vs. Nonholonomic Systems

Mobile robots can be classified as "Holonomic" or "Nonholonomic".

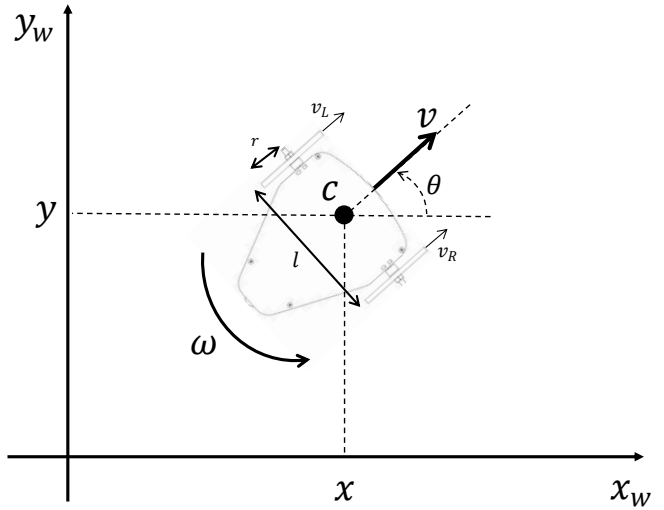
- This classification depends on the relationship between controllable and total degrees of freedom of a robot.

- **Holonomic Robots:** If the controllable degree of freedom is equal to total degrees of freedom, then the robot

- **Nonholonomic Robots:** If the controllable degree of freedom is less than the total degrees of freedom. Such systems are therefore called underactuated. Differential Drive Systems fall into this category.



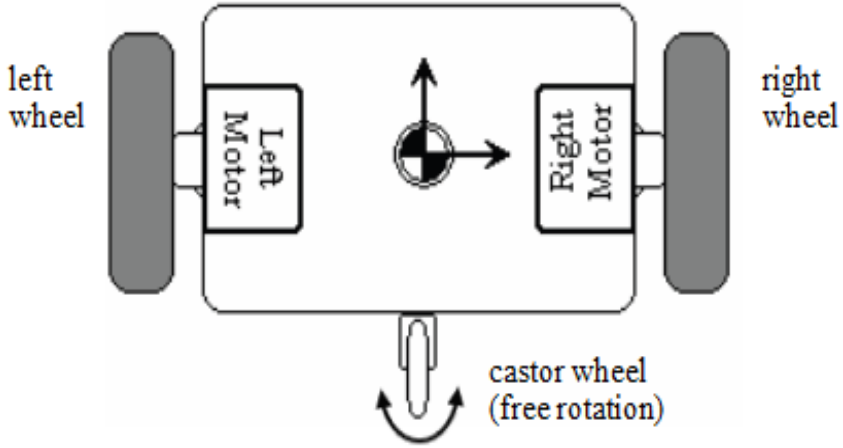
Holonomic Robot

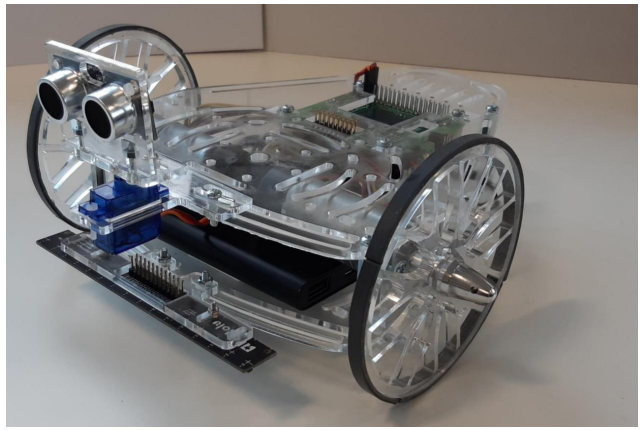


Nonholonomic Robot

Differential Drive

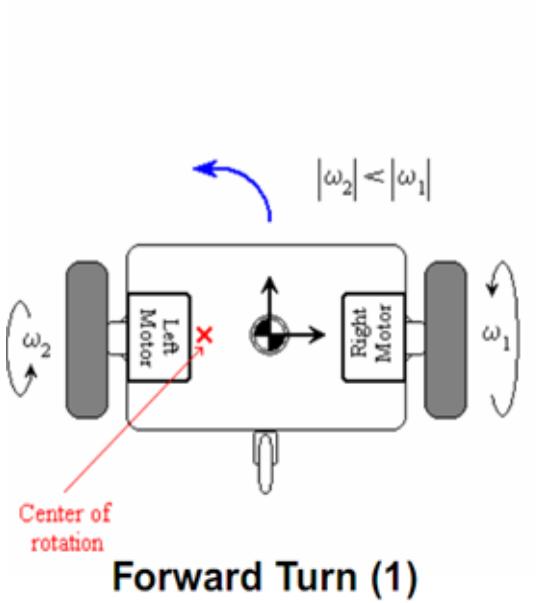
- These are mobile robots whose movement is based on two separately driven wheels placed on either side of the robot body.
- Two active wheels and one caster wheel (most common configuration)
- It can thus change its direction by varying the relative rate of rotation of its wheels, thereby requiring no additional steering motion.



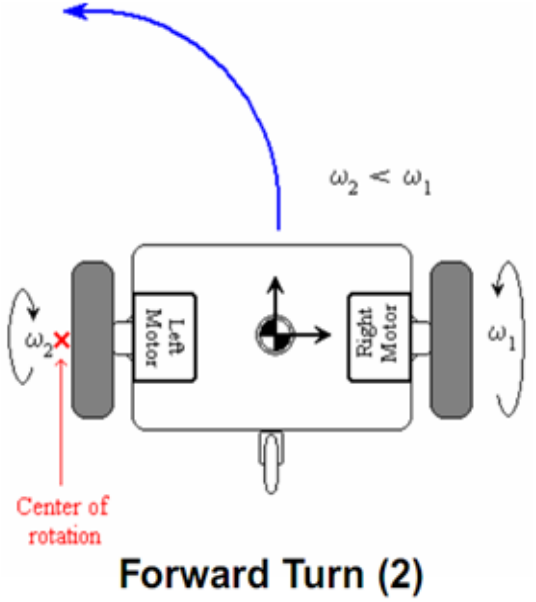


Differential drive robots

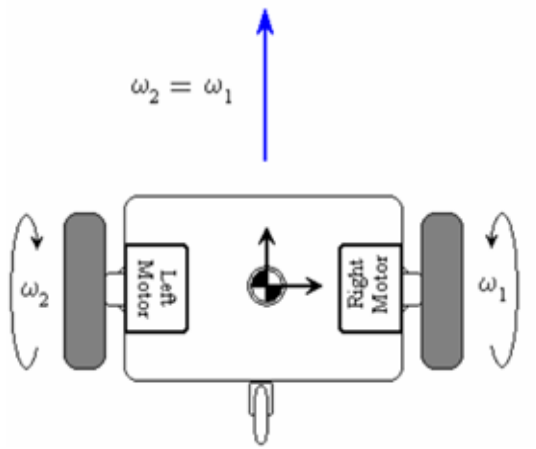
Differential Drive

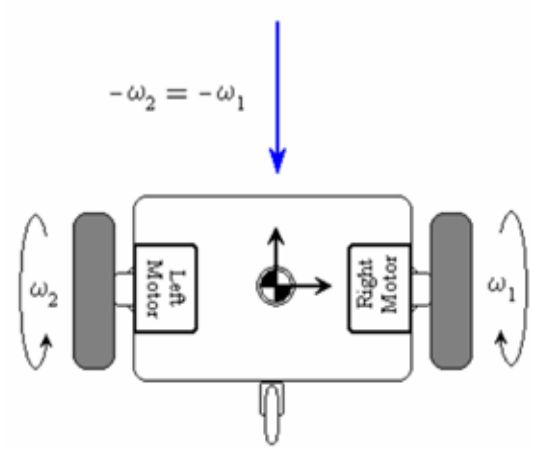


Forward Turn (1)



Forward Turn (2)





Robot Modelling (Kinematics)

- In engineering, a System is an entity that consists of interconnected components, built with a desired purpose. For this case, the system is said to be the differential drive robot.

- Systems can be modelled depending on their dynamical behavior. This models describe the behavior of a system using mathematical concepts and language.

    - Outputs depend on the present and past values of the inputs.
    - Changes over time.
    - Sometimes called dynamic systems or sequential systems.
    - Mathematically described with differential or difference equations.

- **Dynamic Modelling:** Considers different time varying phenomena and the interaction between motions, forces and material properties.

- **Kinematic modelling**: Studies the motion of a robot or mechanism under a set of constraints, regardless of the forces that cause it. Represent the relationship between the robot motor speeds (inputs) and the robot state.

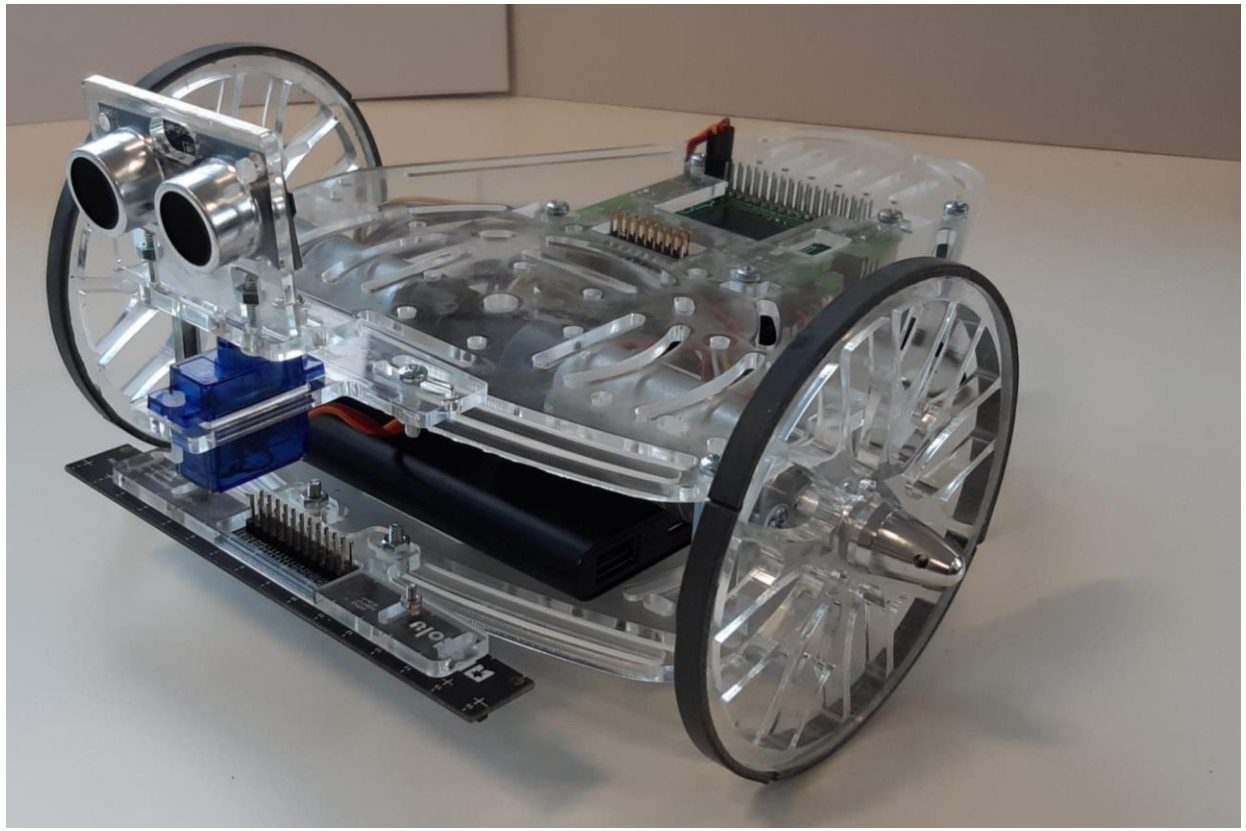
DC Brushed Motor with encoder

$$In \rightarrow \frac{K}{s^2 + 2\zeta\omega_n s + \omega_n^2} \rightarrow Out$$

2nd Order System

Sensors and Actuators for Differential Drive Robots

- As stated before, the kinematic model of a differential drive does not consider the physical characteristics and forces of the robot.
- This type of model is used when testing high level robotic algorithms (control) such as path planning, dead reckoning localization, trajectory tracking, AI, etc.
- Differential drive robots in reality, have sensors and actuators (motors, encoders, etc.) that allow us to control them to reach the correct speed required to correctly perform its functions.

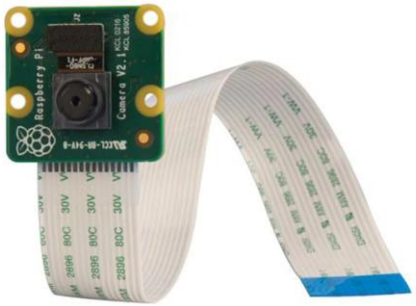


Differential Drive Robot Sensors and Actuators.

Differential Drive Sensors and Actuators

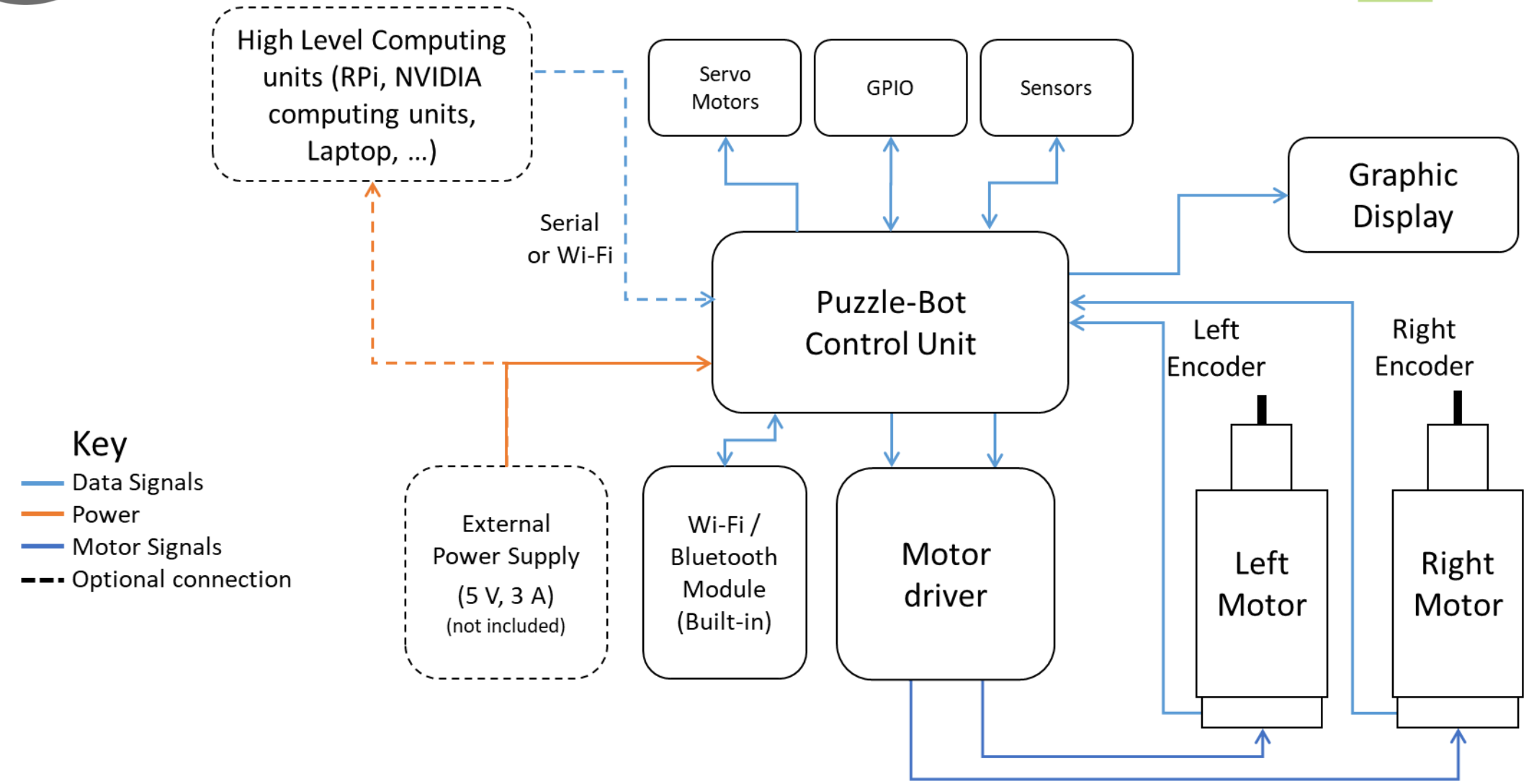
- For the case of the sensors, they can be classified as Exteroceptive and Proprioceptive.

- **Exteroceptive:** Used to measure the environment or the state of the environment, topology of the environment, temperature, etc. Some examples are Sonar, LiDAR, Light sensors, bumper sensors, magnetometers.
- **Proprioceptive:** Used to measure the state of the robot such as wheel position, velocity, acceleration, battery charge, etc. Some examples include, encoders, battery level, gyrometers, accelerometers.



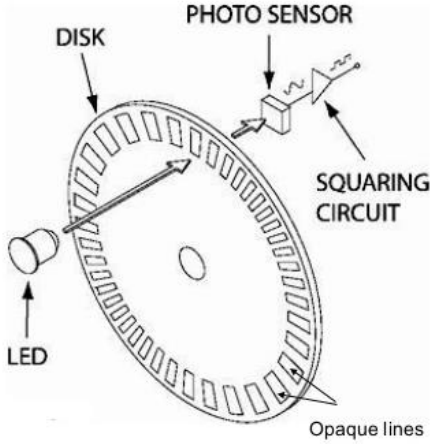
Sensors and Actuators.

Differential Drive Sensors and Actuators

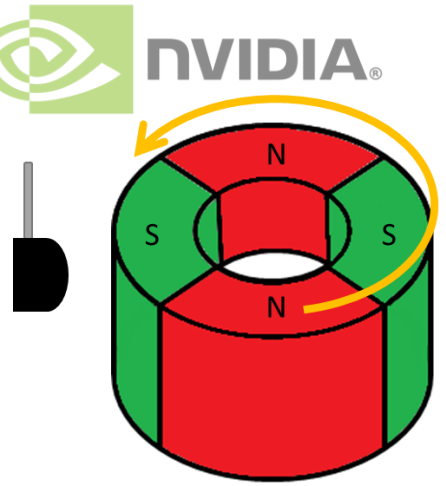


Encoders

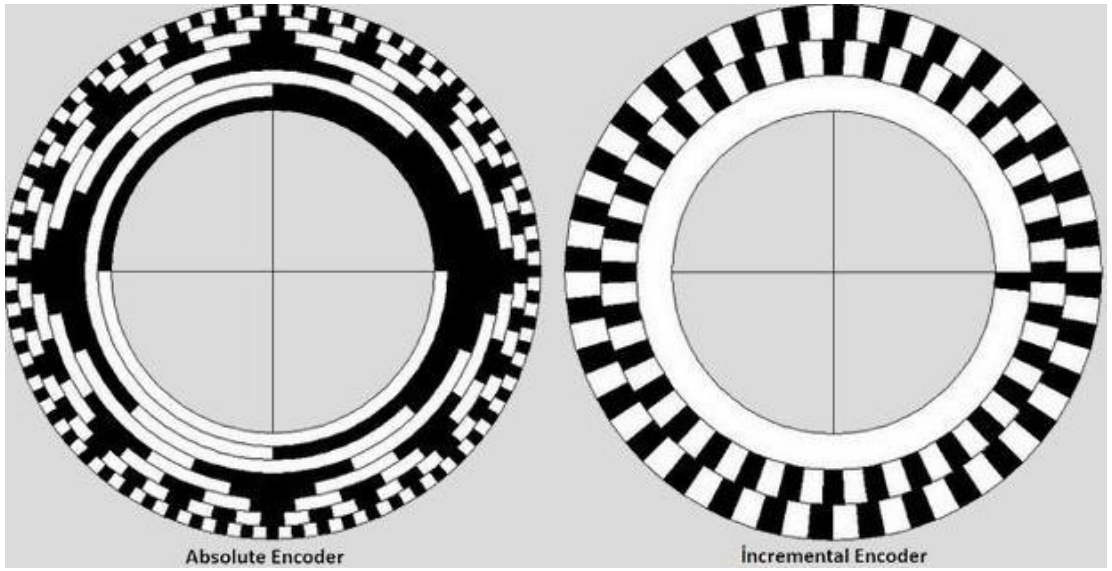
- Device that converts the angular position of a shaft (motor shaft) to an analogue or digital signal.
  - Absolute: Indicates the position of the shaft at all times by producing a unique digital code for each angle (Angle transducers).
  - Incremental: Record the changes in position of the motor shaft with no indication or relation to any fixed position of the shaft.
- Encoders in mobile robots are considered **proprioceptive** sensors because they only acquire information about the robot itself, not the structure of the environment.



Optical Rotary Encoder



Magnetic Rotary Encoder

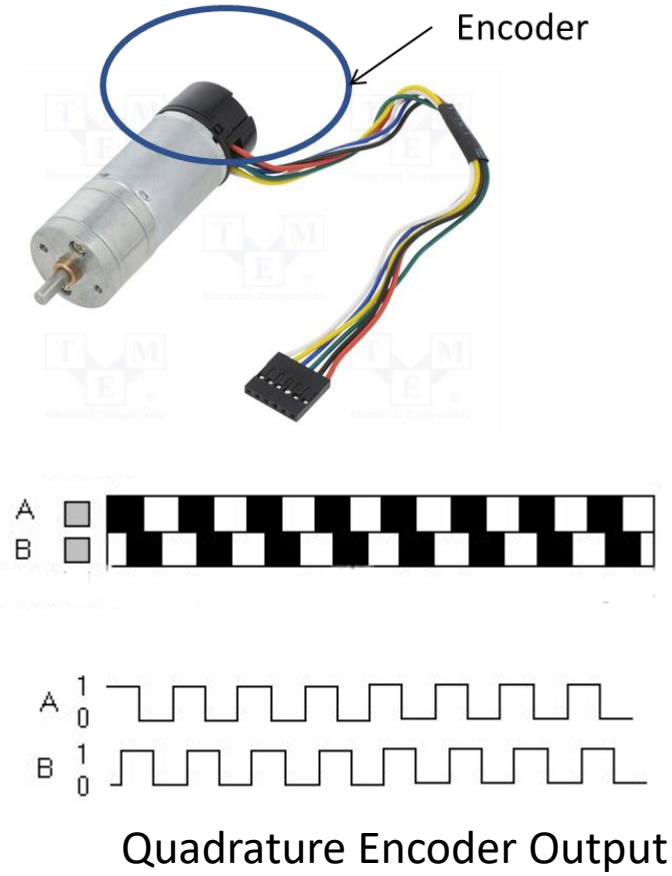


Encoders

- When an encoder is attached to the axle of each wheel in a differential-drive robot, it is possible to convert the number of pulses into useful information, such as the velocity or distance travelled by each wheel.

- PuzzleBot uses 2 incremental dual channel, quadrature encoders with 400 Pulses per revolution, attached to the motor shafts to estimate the speed of the motors.

- By counting the number of pulses that occur in a certain period of time (Δt) it is possible to estimate the velocity of the motor.

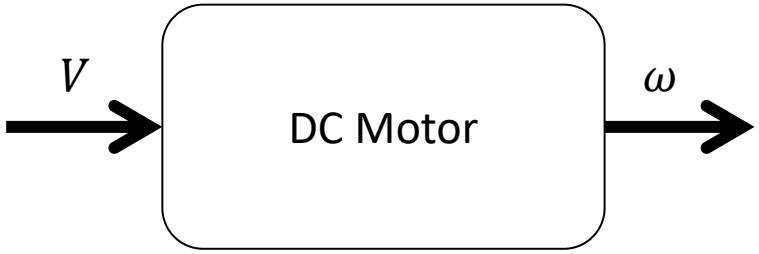


Quadrature Encoder Output

DC Motors

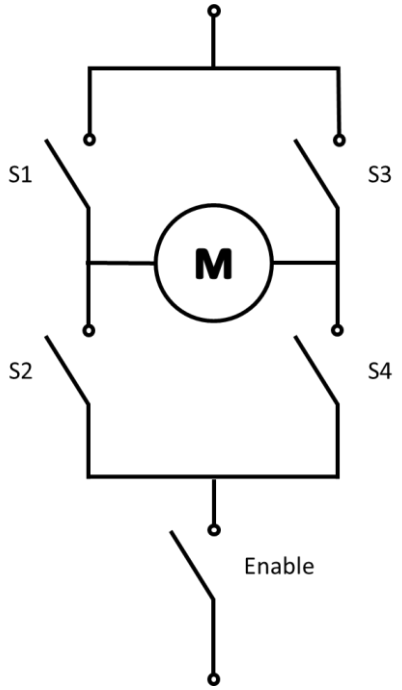
- A direct current (DC) motor is a type of electric machine that converts electrical energy into mechanical energy.
- DC motors take electrical power through direct current and convert this energy into mechanical rotation.
- This is done by using generated magnetic fields from the electrical currents, powering the movement of a rotor fixed within the output shaft. The output torque and speed depends upon both the electrical input and the design of the motor.
- In control and robotics DC motors are usually modeled as second order systems. The input to this system is the voltage (V) and the output is the angular velocity (ω).

DC Brushed Motor with encoder



Motor Driver

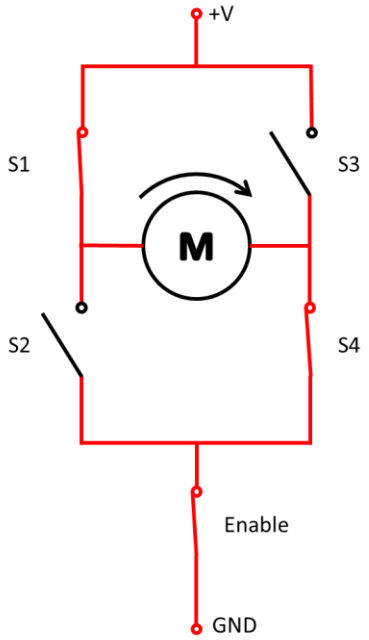
- A H-bridge is an electronic circuit that switches the polarity of a voltage applied to a load. They work using a combination of switching components (mechanical switches, transistors, etc.) as shown in the diagram to change the polarity to the load.

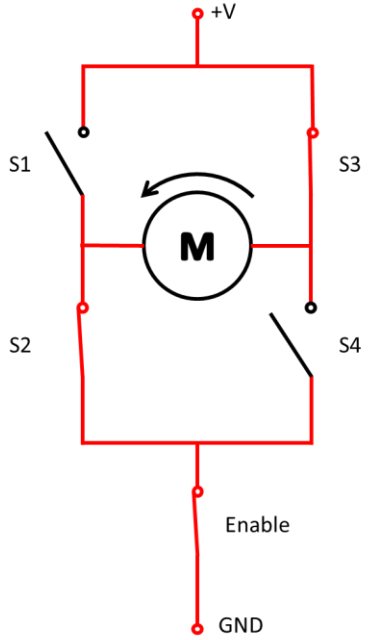


H-Bridge motor Driver Diagram

- H-Bridge Drivers are some of the most common motor drivers used in robotic applications such as the control DC motors to run forwards or backwards.

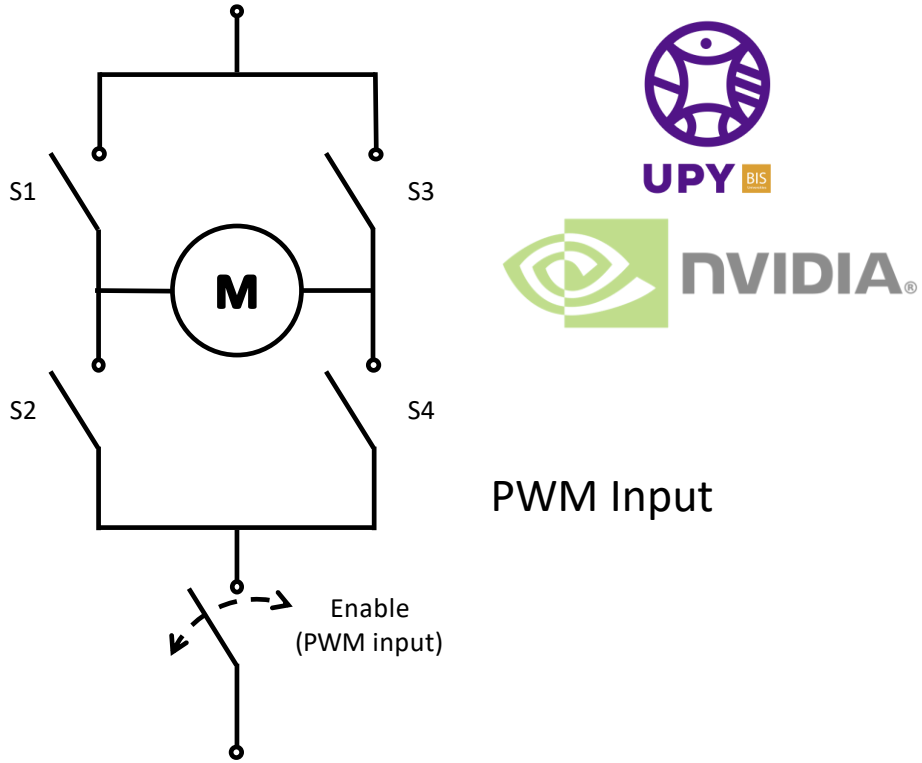
S1	S2	S3	S4	Motor
0	0	0	0	Motor Off
1	0	0	1	Right Turn
0	1	1	0	Left Turn
1	1	0	0	Short Circuit
0	0	1	1	Short Circuit



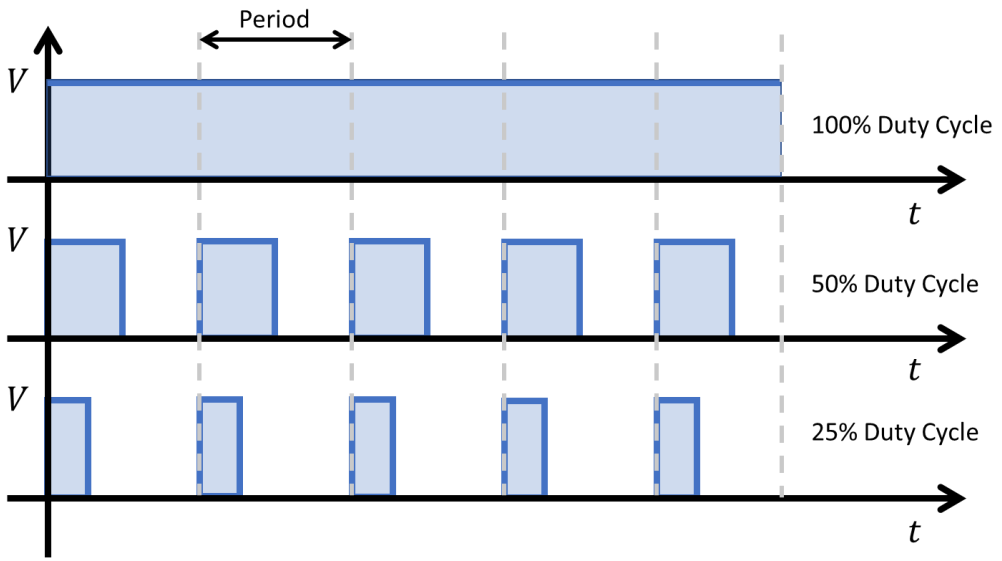


Motor Driver

- Another capability of the motor driver is to regulate the angular speed of the motor.
- There are many ways to obtain this result, one of the most common one is to send a PWM (Pulse width modulated signal) to the enable pin of the H-Bridge.
-
- PWM (Pulse Width Modulation): Is a technique used in engineering to control the average power delivered by an electrical signal, by dividing it into discrete parts.
- In practice this is accomplished by rapidly turning the switch between the load and the source (enable switch), ON and OFF.



PWM Input

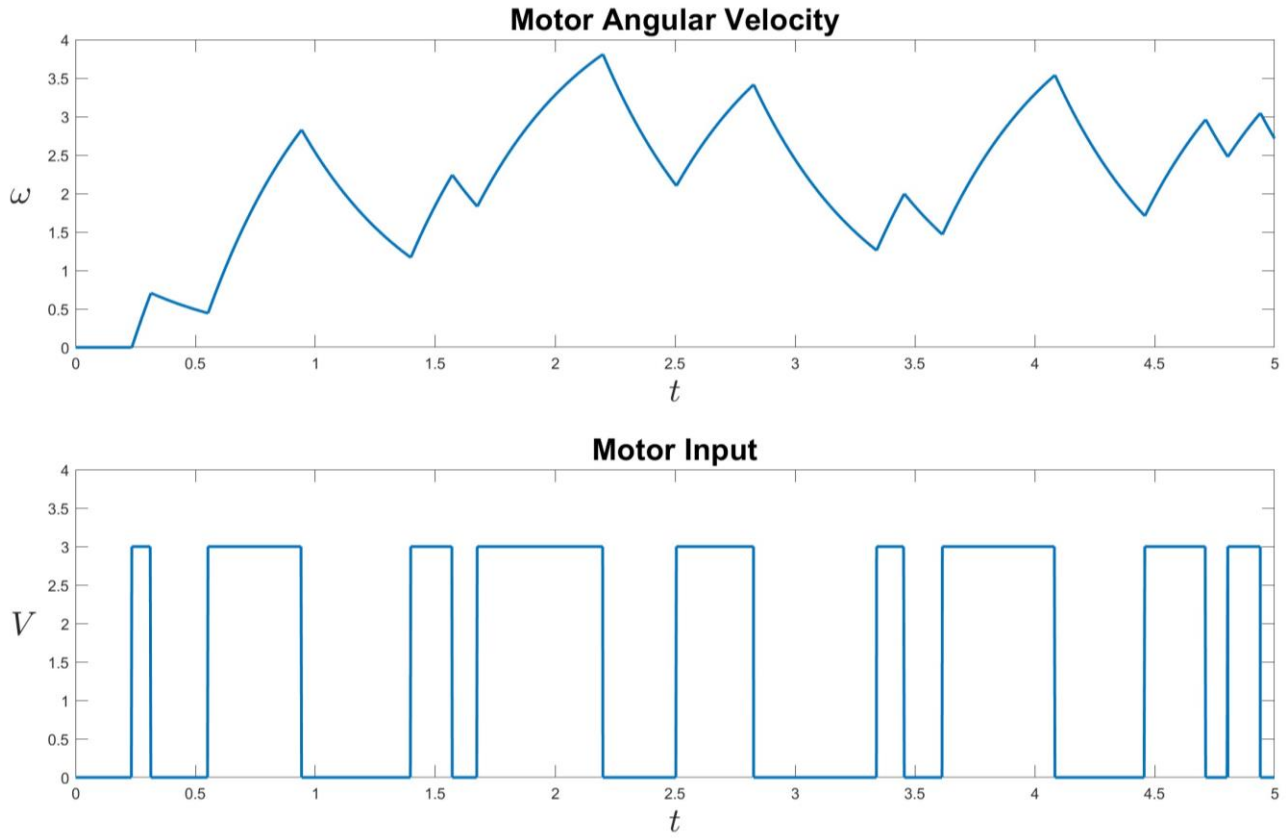


PWM Duty Cycle

Motor Driver

- Given that the motor can be modeled as a second order systems, when applying a PWM voltage as an input, it is possible to observe the output behaviour as in the figure.

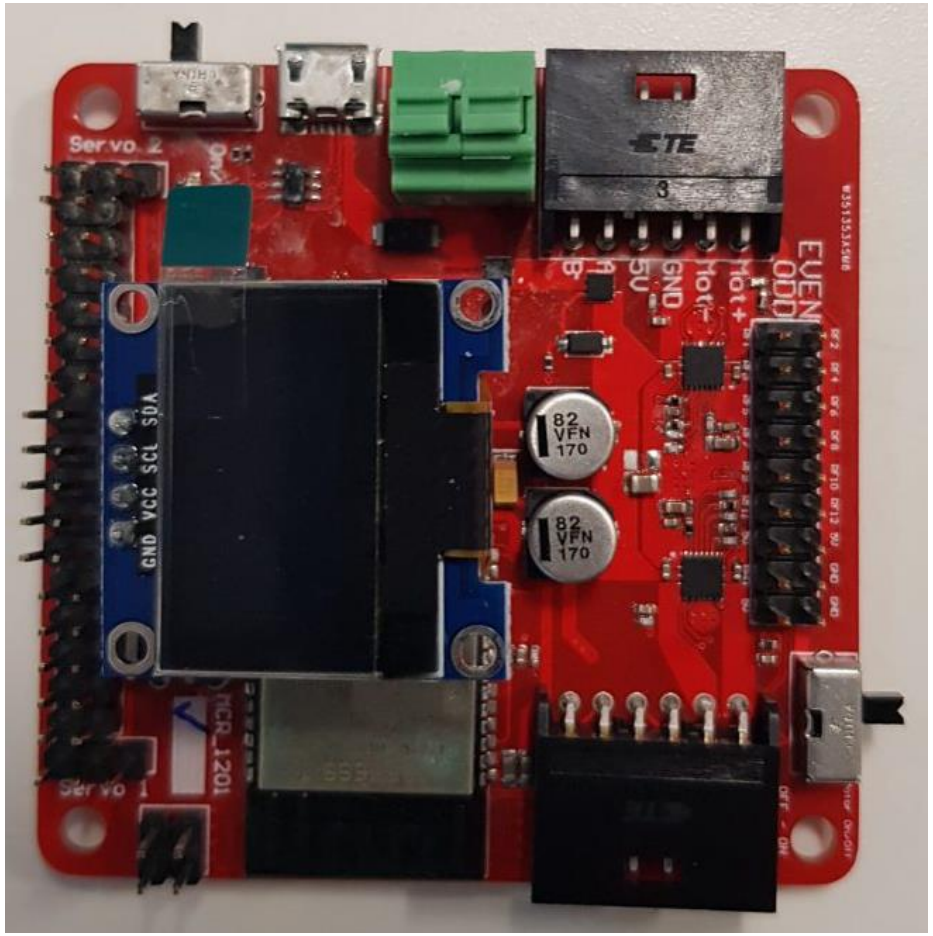
- This behavior, can be used to control the power give to the motor and therefore controlling the motor angular speed.



Motor Input/Output

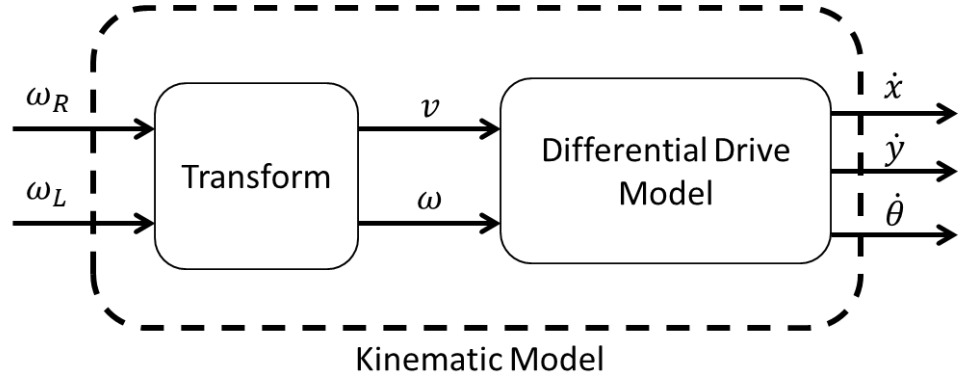
MCU

- A microcontroller is a compact integrated circuit. They are made to perform a specific operation in an embedded system.
- In robotics they are usually in charge of the low-level control of the robot, such as motors, and sensors or actuators that require a dedicated and fast controller to work.
- A typical microcontroller includes a processor, memory and input/output (I/O) peripherals on a single chip.
- For the case of the PuzzleBot:
  - ESP32-based Microcontroller
    - Xtensa dual-core 32-bit LX6 microprocessor
    - 520 KB of SRAM
    - WiFi & Bluetooth
  - DC-DC Converter
  - Motor Driver
  - 0.96” I2C LCD Display

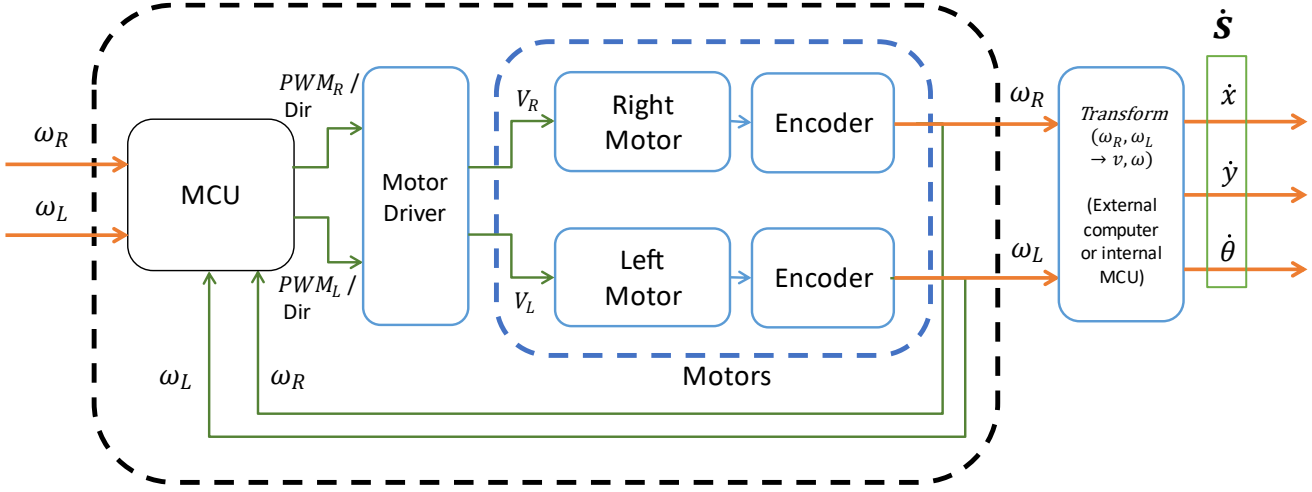


The Mobile Robot

- The real robot, in comparison with the kinematic model, requires the usage of the previously shown sensors and actuators to work.
- For the real robot, a closed control loop for each of the motors is required, in order to reach the required velocities, set by the user.
- In robotics this is called low level control. For the case of a wheeled mobile robot is a common practice to implement a PID control.
- It can be observed that the inputs and outputs are the same, but the inner loop controller and actuators will determine how close the Real Robot will resemble the Kinematic Model.



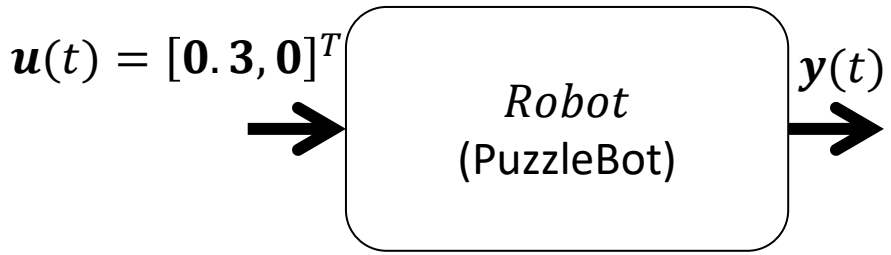
Kinematic Model



Real Robot Diagram*

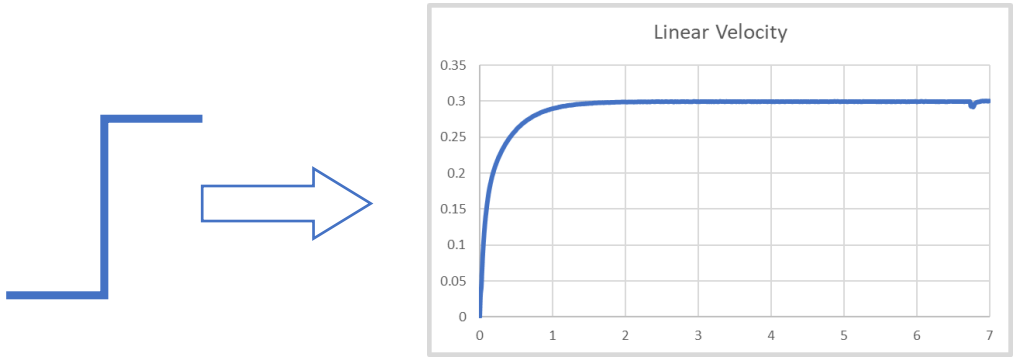
Open Loop Control

- Let the system (in this case the robot) to be represented as a black box and let only constant linear velocity of $v = 0.3\frac{m}{s}$ to be the input to the system (step input).



Black Box Model

- It can be observed that the output of the system matches the input reference at steady state. Therefore, the gain of the system is said to be 1. **This gain is only for the robot system for any other system the system's gain must be estimated.*



Input

Output

- A more in-depth analysis of this situation reveals that this is due to the inner controllers of the robot, allows the robot to follow the reference after some transient. The same thing happens with the angular speed.

Open Loop Control

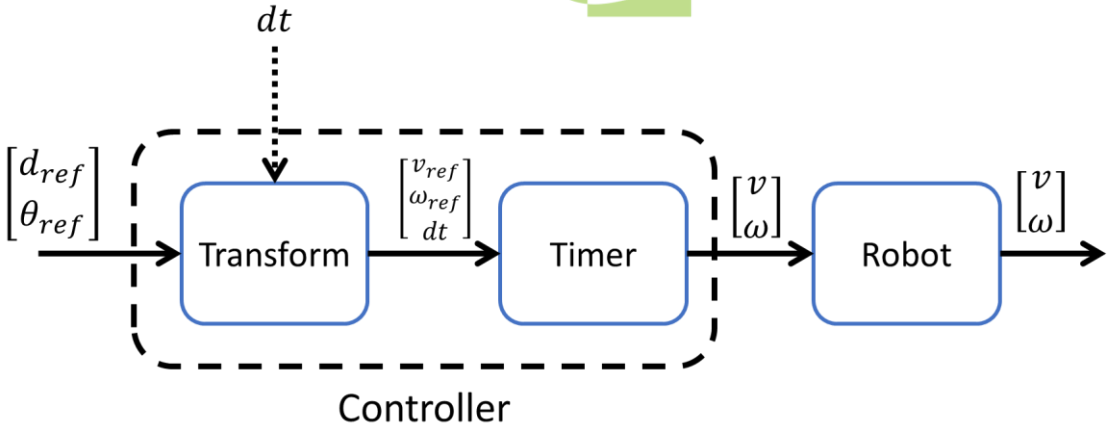
- Knowing that the system's output will follow the input after a transient, a question arises. Can we use this information to move the robot a certain distance?
- Since we know that the output follows the input, it is possible to approximate the distance and angle traveled by using the following formula.

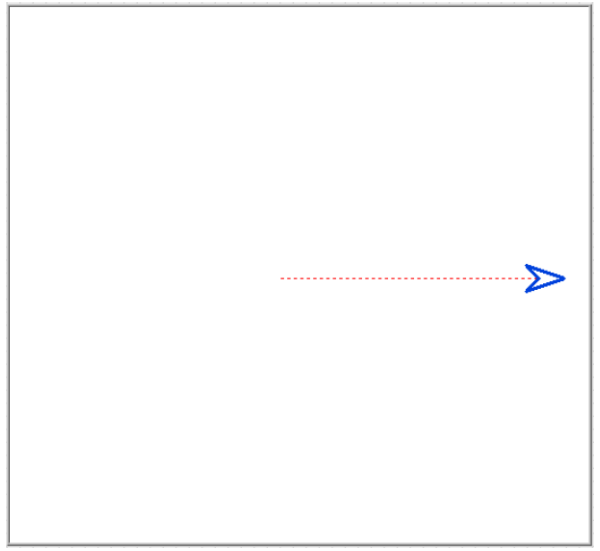
$$d = v \cdot dt = r\left(\frac{\omega_r + \omega_l}{2}\right) \cdot dt$$

$$\theta = \omega \cdot dt = r\left(\frac{\omega_r - \omega_l}{l}\right) \cdot dt$$

where dt is the time since the input was applied.

- Therefore, it is possible to use a timer to control the distance and angle travelled by the robot.
- This is called open loop control.

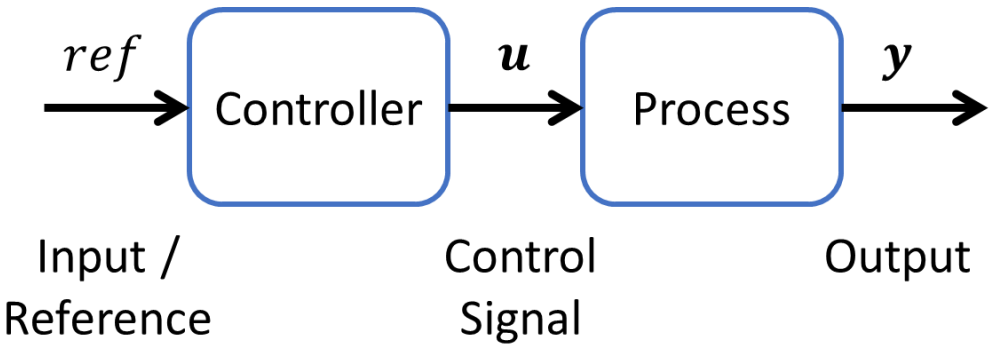


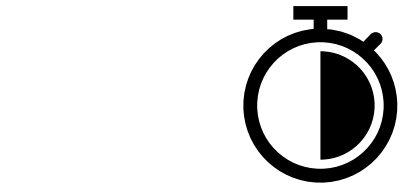


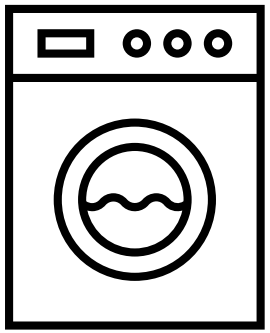
Open Loop Control

Open Loop Control

- Open Loop Control System is a system in which the control action is independent of the output of the system.
- In this type of control, the output is regulated by varying the input or reference.
- The output of the system is determined by the current state of the system and the inputs that are received from the controller.
- Some examples of this type of control systems are Windows, window blinds, washing machines, microwave ovens, hair drier, bread toaster, Door Lock System, some stepper motors, turning on/off lights, some remote-controlled applications, etc.

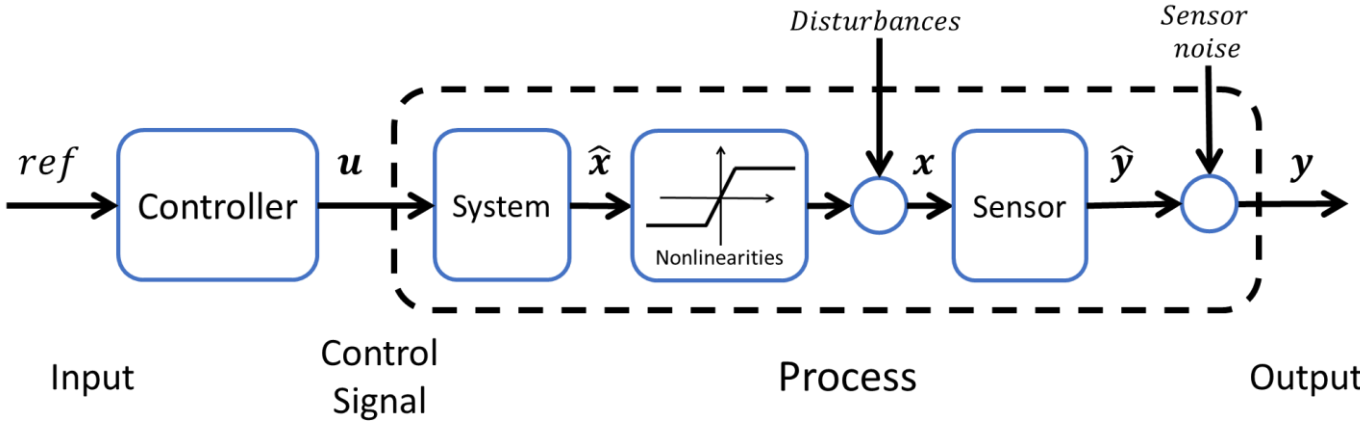


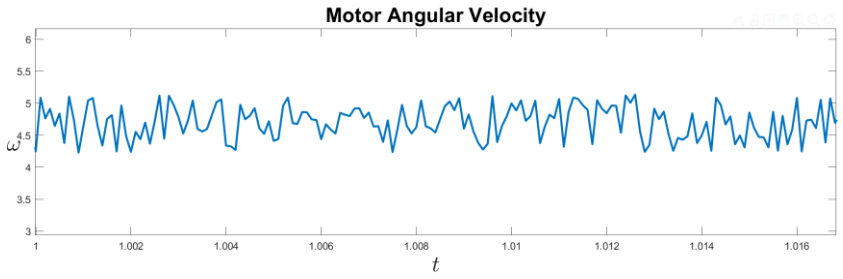




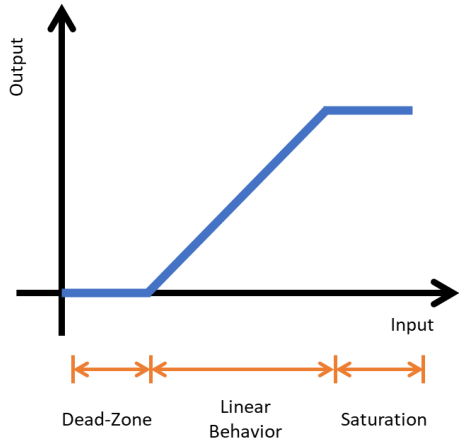
Open Loop Control

- In reality, every process present disturbances, nonlinearities and noise.
- Disturbances and noise come from the environment that surrounds the system.
  - On the differential drive robot, some disturbances can come from wheels slipping, obstacles on the environment, different types of terrain, etc.
  - On the other hand, the noise is present when reading the values of the motor encoders (Sensors).
- Nonlinearities are a intrinsic characteristic of a system in which the output does not linearly follow the input (output not proportional to the input).
  - For the case of the robot, the nonlinear behavior can be seen as a saturation and dead-zone in the motors and the robot itself.





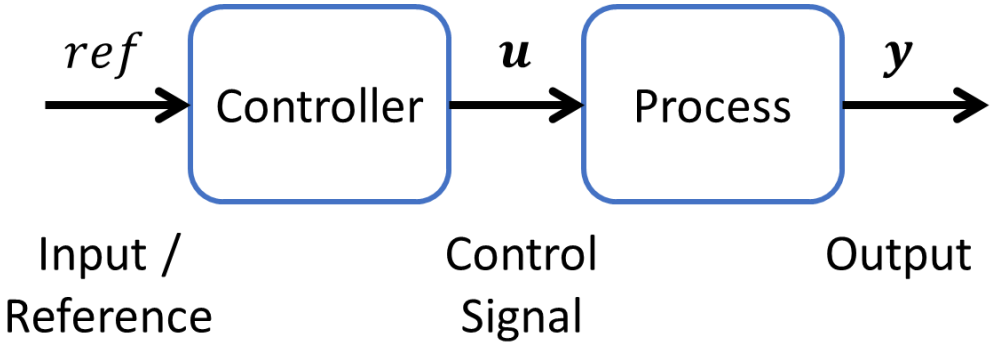
Noisy Signal (Angular velocity)



Nonlinear behavior

Open Loop Control

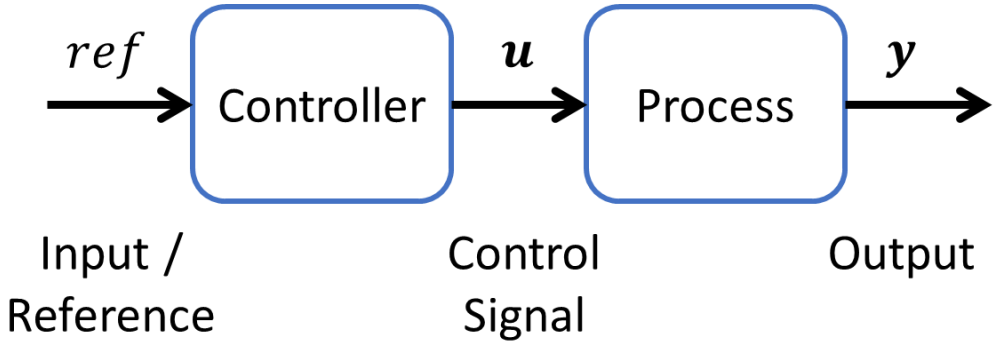
- Open Loop Advantages:
  - Simple to design and implement, provided that the user has some experience with the system.
  - The cost for design, implement and maintain is relatively low compared with other controllers.
  - Maintenance is considered simple, no high technical level required (for most of the controllers).
  - The behavior of the controller is quite stable, provided that the system is in a controlled environment, such that the process does not present big disturbances, or the process is not dangerous when unsupervised.

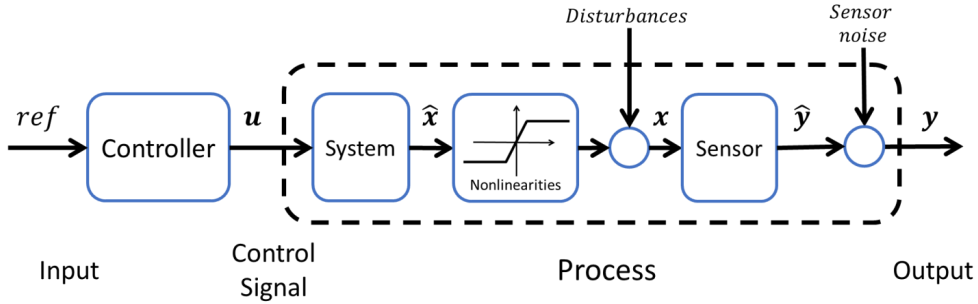


Open Loop Control

- Open Loop Disadvantages:

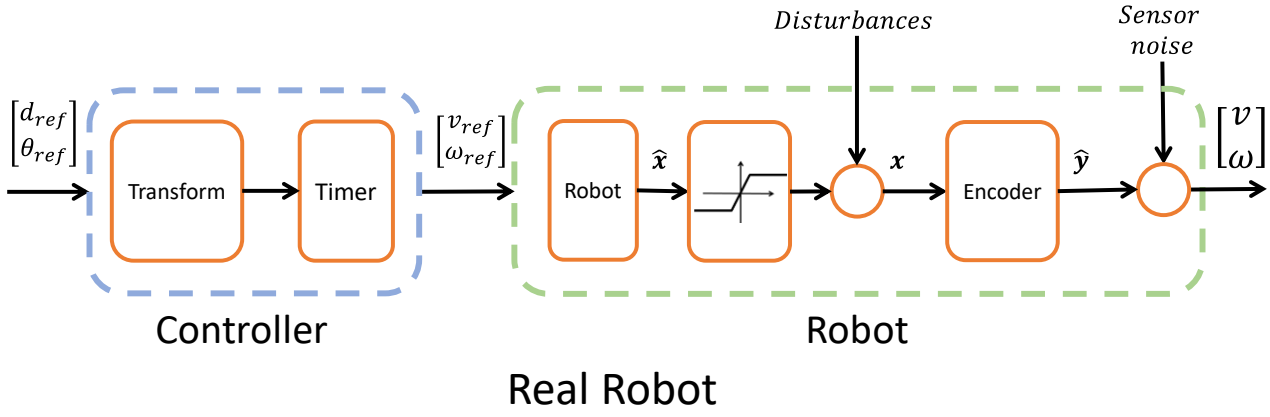
  - This type of control is not robust against disturbances (cannot correct the output in the presence of a disturbance).
  - An open-loop system has no self-regulation or control action over the output value.
  - Not reliable
  - The input depends on the experience of the user.
  - Each input determines a fixed operating point of the system.
  - Controller must be altered manually in case of an output disturbance or uncalibrated controller.
  - Requires re-calibration often.
  - Prone to errors in the output and control signal
  - Does not take into consideration changes on the process over time.
  - Also, there is no chance to correct the transition errors in open loop systems so there is more chance to occur errors.



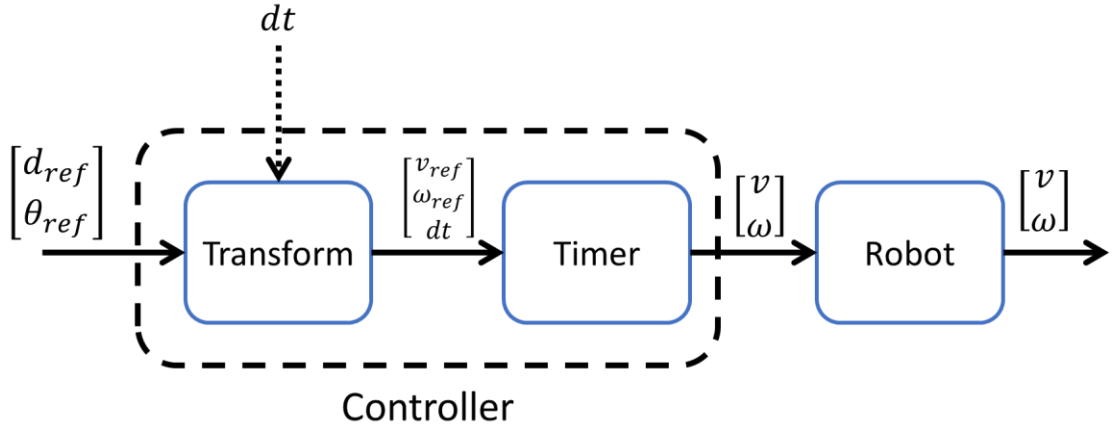


Open Loop Control: Mobile Robot

- As stated before, open loop control can be used to control de position of the robot.
- When designing this controller, the nonlinear behavior, perturbations and noise should be taken into consideration to make the controller work on the linear part of the robot and make it as robust as possible.
- In this case, the real mobile robot is a nonlinear system, due to the maximum and minimum velocities of the motors therefore it presents saturation and dead-zone.
  *Other nonlinear behavior are present, but they are not relevant in this case.
- The linear behavior is present within a region in which the actuators (motors) output are proportional to the input.



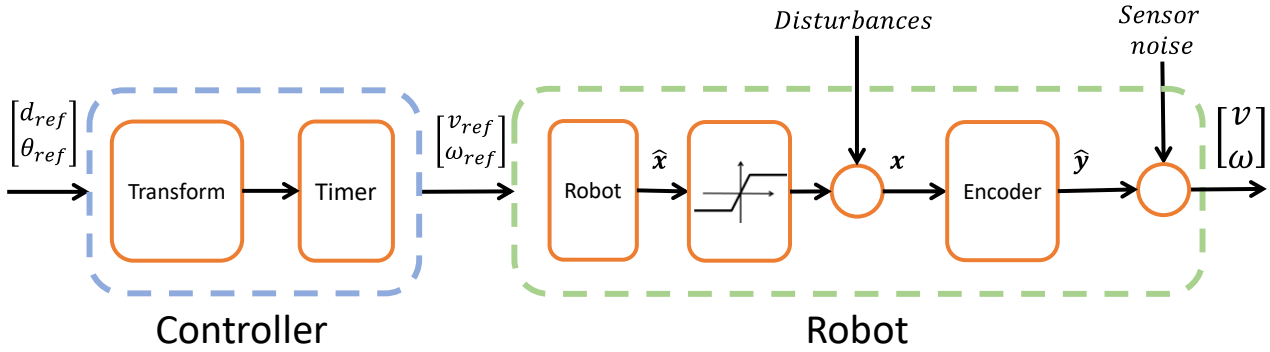
Real Robot

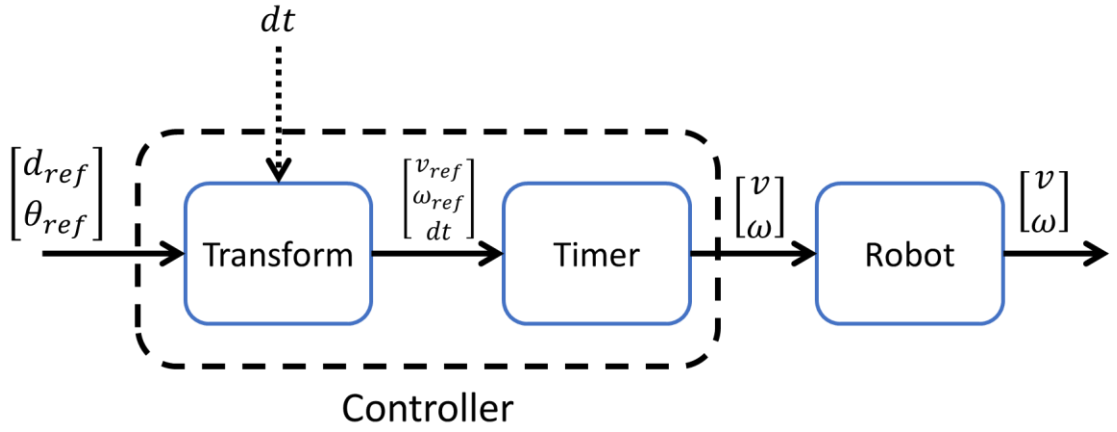


Simulated robot

Open Loop Control: Mobile Robot

- Noise and perturbations can also be found in the robot.
- As said before, noise can be found in the encoder reading, the noise can be due to electromagnetic noise or radiation, mechanical unbalance of encoders, etc.
- Disturbances are due to the mechanical unbalance of the wheels, wheel slippage with the floor, defects of productions, changes in the environment, etc.
- Depending on the simulator, the dynamical behavior of the robot can be linear or nonlinear and can consider the noise and disturbances .



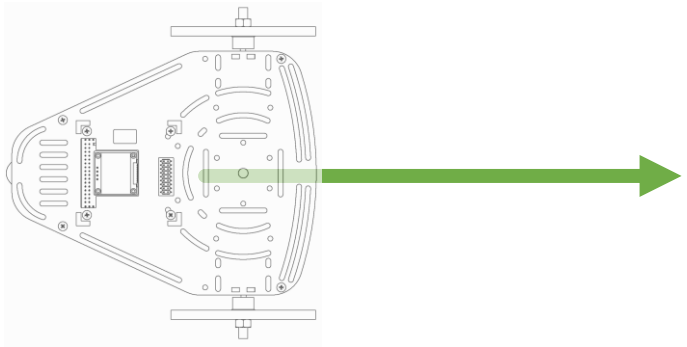


Now is your turn…

- *The following 2 activities will help you build an open loop controller for the Puzzlebot Gazebo simulation.*
  - *Activity 1: Create a node that sends commands to the robot gazebo simulation and move in a straight line for a period of time.*
  - *Activity 2: Using the previous created node, expand it to create a square path.*

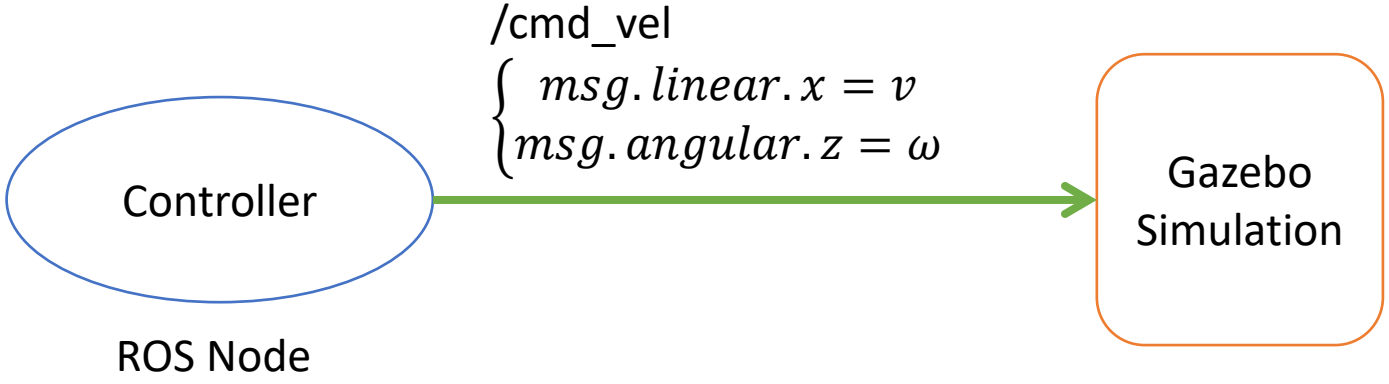
Activity 1
Moving a PuzzleBot

Straight line



Drive the robot 2 meters in a straight line.

- Objectives:
  - Create an open loop control as a ROS node that sends commands to the robot gazebo simulation and move in a straight line for 2 meters.



Activity
Some tips and tricks

- One publisher `cmd_vel`
- `cmd_vel` – `from geometry_msgs.msg import Twist` – 3 linear and 3 angular velocities.
  - `msg.linear.x`, `msg.linear.y`, `msg.linear.z`
  - `msg.angular.x`, `msg.angular.y`, `msg.angular.z`
- Use the equations below to compute the distance moved and the angle turned by the robot

$$d = v \cdot dt = r\left(\frac{\omega_r + \omega_l}{2}\right) \cdot dt$$

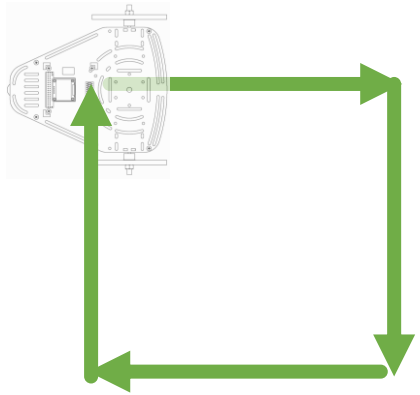
$$\theta = \omega \cdot dt = r\left(\frac{\omega_r - \omega_l}{l}\right) \cdot dt$$

where r is the radius of the wheels ($0.05\ m$) and l is the distance between the wheels ($0.18\ m$)

- Use `rospy.get_time()` to measure the time dt between each loop
- If the robot is not moving, check your topics with `rostopic echo` and `rostopic pub`
- Ensure your python file is executable: `sudo chmod +x <path_to_file>.py`
- Ensure you source your file: `source devel/setup.bash`

Activity
Moving a PuzzleBot

Square



Drive the robot making a square with a side length of 2m.

- Objectives:
  - Expand your previously created node or create a new node to make your robot drive in a square path of a side length 2 m.