ROB - Laboratory focused on Trilobot

Adam Fabo
Jan Beran

Brno University of Technology, Faculty of Information Technology
Božetěchova 1/2. 612 66 Brno - Královo Pole

xfaboa00@fit.vutbr.cz

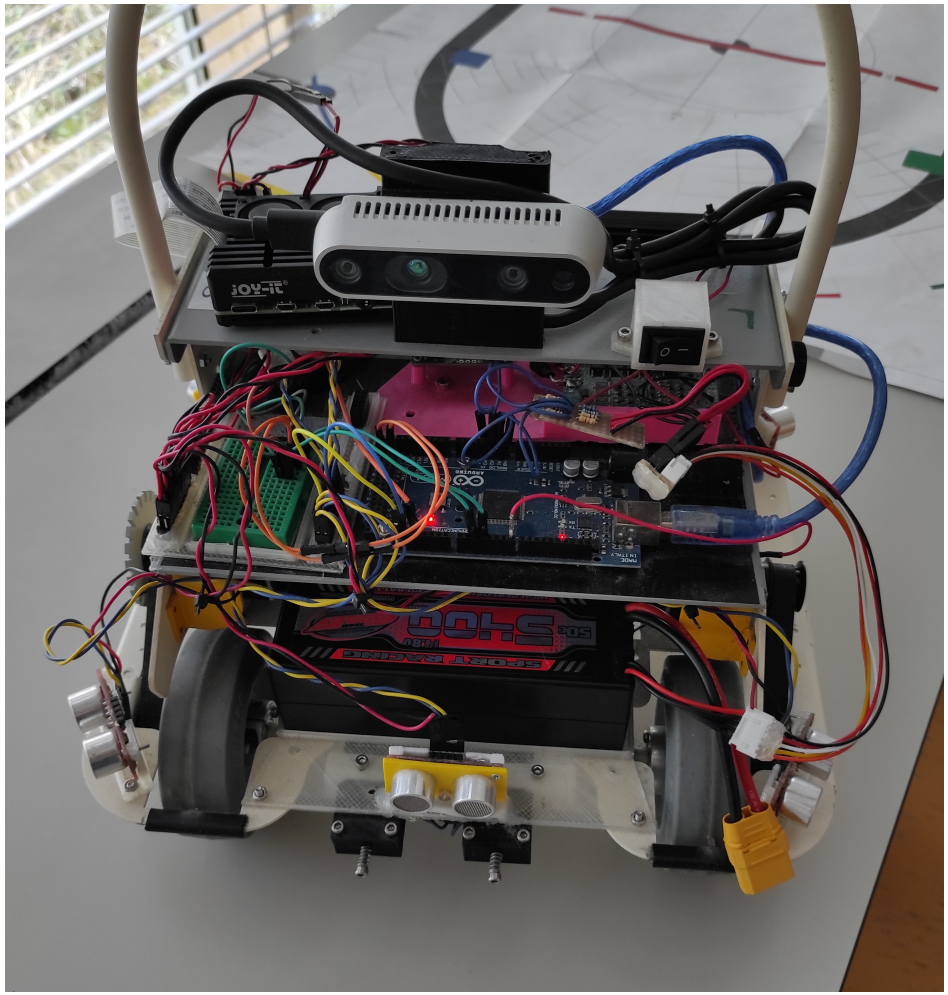
December 4, 2023

Today's lab will be focused on controlling robot Trilobot. You will learn:

- How robot Trilobot works, its architecture
- Controlling motors using ROS
- Reading data from sonars using ROS
- Making simple obstacle avoiding algorithm

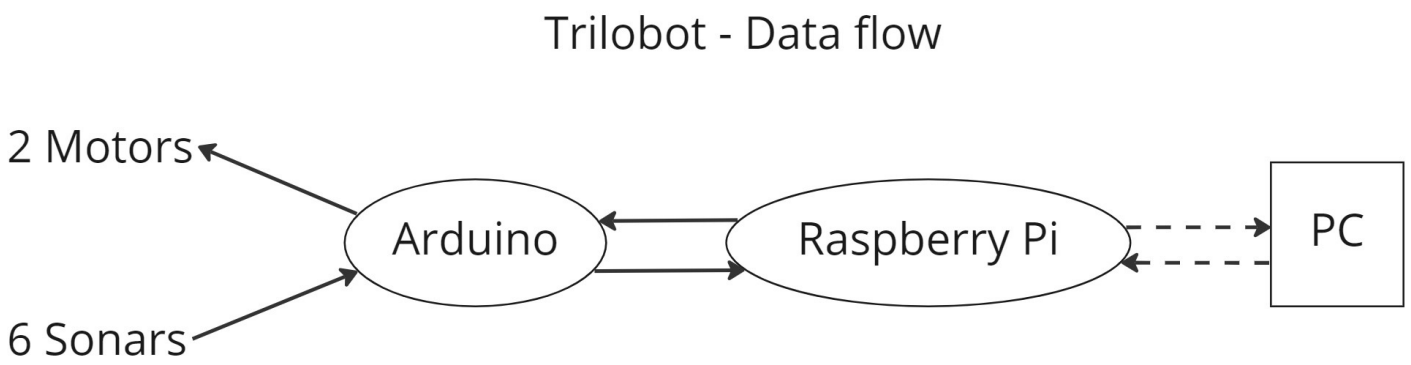
Trilobot

- Trilobot is a robot platform that is designed to be used by students that are learning robotics
- It has 2 motors, 6 sonars and depth camera



Trilobot - architecture

- Core of Trilobot is Raspberry pi (verzia here)
- It has 2 motors, 6 sonars and depth camera
- For motor control Arduino mega is used



- Trilobot uses more complex messages than simple String for communication
- Motor control uses Twist message. Its alternative Python structure can be seen below:

```
class Twist:
    def __init__(self):
        self.linear  = Linear()
        self.angular = Angular()


class Linear:
    def __init__(self, x = 0.0, y = 0.0, z = 0.0):
        self.x = x
        self.y = y
        self.z = z


class Angular:
    def __init__(self, x = 0.0, y = 0.0, z = 0.0):
        self.x = x
        self.y = y
        self.z = z
```

- Twist message can be used as following:

```
message = Twist()
message.linear.x  = 5
message.angular.z = 5

pub.publish(message)
```

- For getting data from sonars, Trilobot uses Sonar_data message. Its alternative Python structure can be seen below:

```
class Sonar_data:
    def __init__(self, front=0, front_left=0, front_right=0,
                       back=0, back_left=0, back_right=0,):
        self.front       = front
        self.front_left  = front_left
        self.front_right = front_right
        self.back        = back
        self.back_left   = back_left
        self.back_right  = back_right
```

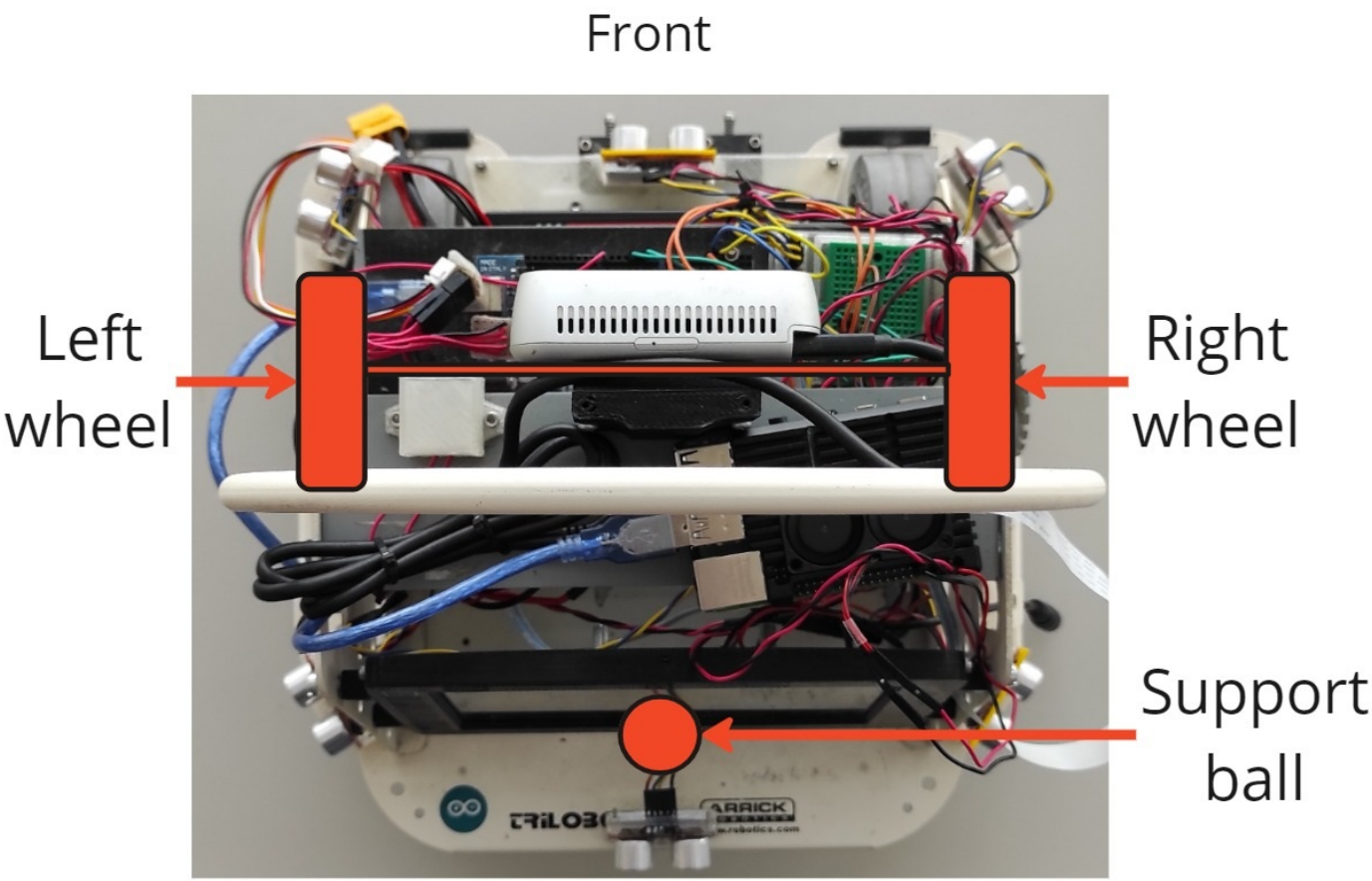
- Sonar_data message can be used as following:

```
message = Sonar_data()
message.front  = 5
message.back_left = 5

pub.publish(message)
```

Trilboot - Motors

- Trilobot has 2 Faulhaber 16002 motors on front axle
- Each wheel has its own motor
- Does not have rear wheels, has support ball instead



How to operate Trilobot

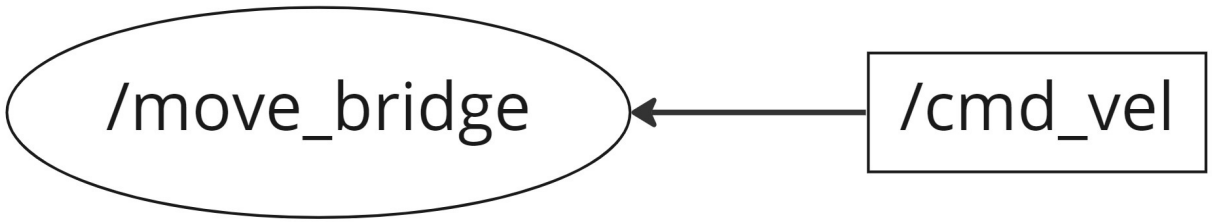
- Make sure you are connected to **TriloAPX**, where **X** is number 1,3 or 6.
- **SSH**: *ssh -X trilo6@10.42.0.1*, password: trilo6
  - Yes, even for Trilobots with number 1 and 3, name and password is trilo6.
- ROS master (and some other nodes) will be run on host machine
- Low level nodes (*roscore*) will run directly on Trilobot

- Code will be written both on Trilobot and host machine
- To write code directly on Tirlobot, use VSCode and Remote - SSH extension:
  - In VSCode, open command prompt with *CTRL + SHIFT + P*
  - Type *Connect Current Window to Host*
  - Choose either *Add New SSH Host* or use existing connection
  - ssh login for new connection: trilo6@10.42.0.1
  - Now you can use it as standard VSCode

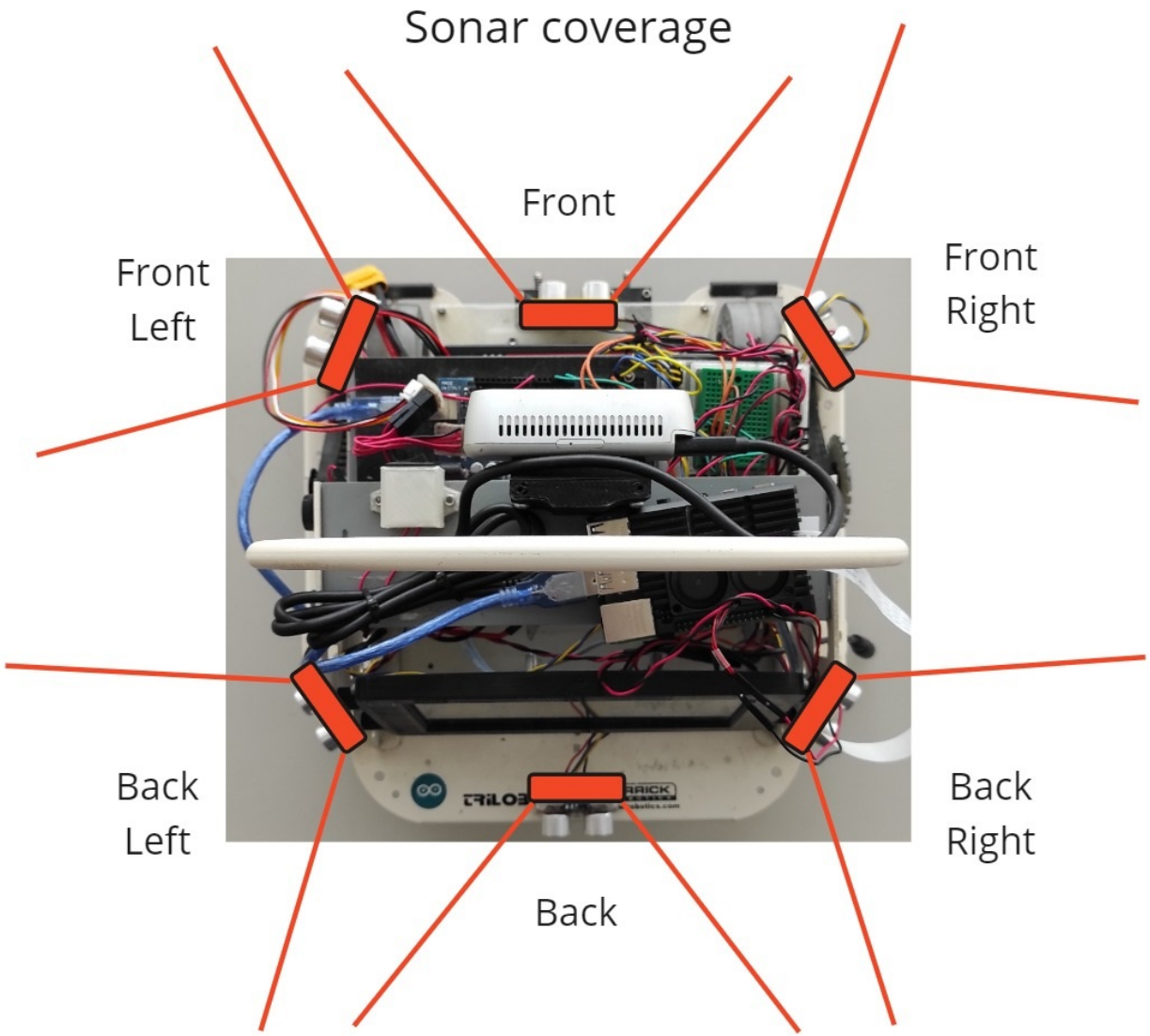
Trilobot and `rosserial`

- In order to enable Arduino and low level data (sonars, motor commands...), `rosserial` is needed
- (on trilobot) `roslaunch trilobot rosserial.launch`

- Movement of Trilobot can be controlled by sending Twist messages into /cmd_vel topic

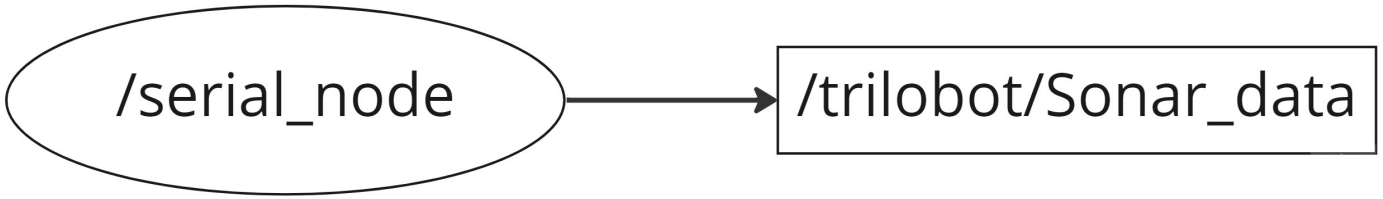


- Trilobot has 6 SRF08 sonars around its perimeter



Trilobot - Getting sonar data

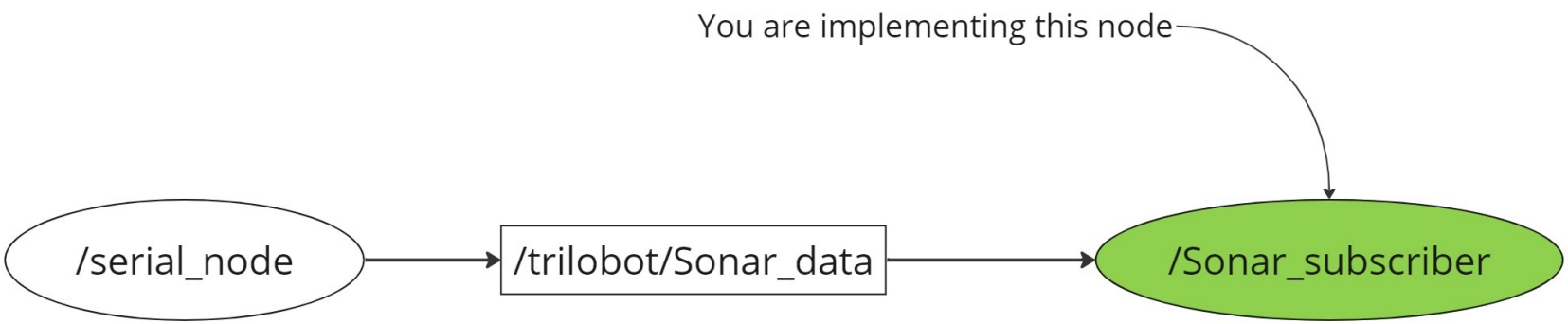
- Distances from sonars are sent periodically into /trilobot/Sonar_data topic
- Message type is Sonar_data



Excercise - Reading sonar data

Test if Trilobot's sonars are wroking. Read the data from topic /trilobot/Sonar_data and print them in console.

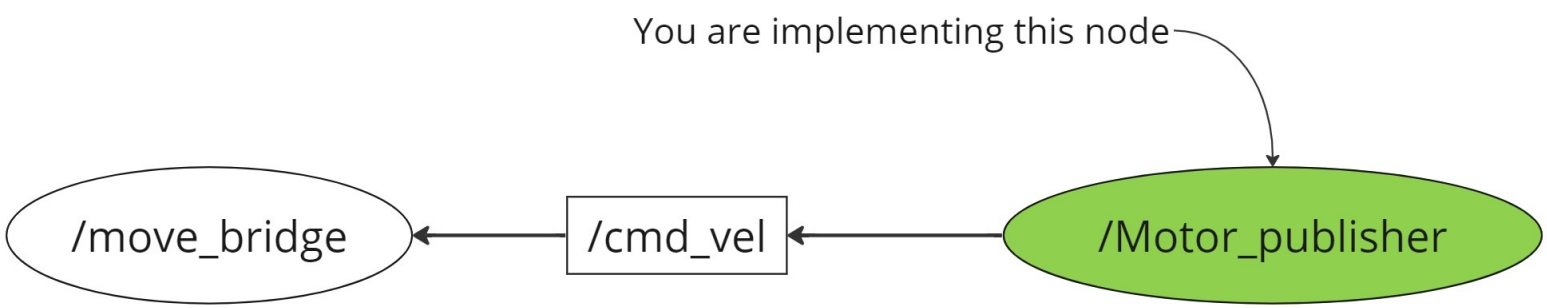
- Open file **trilobot_sonars/src/sonars.py**
- Implement subscriber that reads messages from topic /trilobot/Sonar_data
- Run your node with `rosrun trilobot_sonars sonars.py`
- Test your implementation by placing obstacles around Trilobot and observing read values.



Excercise - Controlling motors

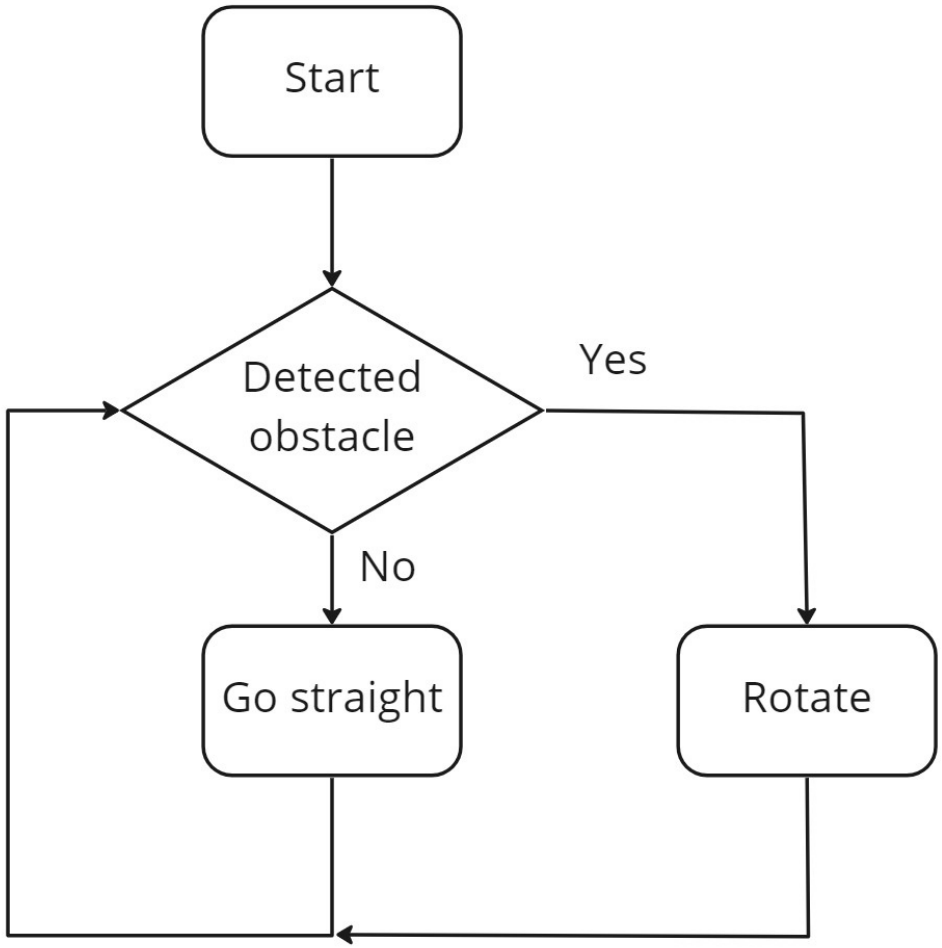
Test if Trilobot's motors are working

- Place Trilobot on a box in a way, that its wheels are in the air
- Open file **trilobot_motors/src/motors.py**
- Implement publisher that publishes Twist messages into /cmd_vel topic
- Run your node with `rosrun trilobot_motors motors.py`
- Test your implementation by writing different values into Twist message.



Simple obstacle avoiding algorithm

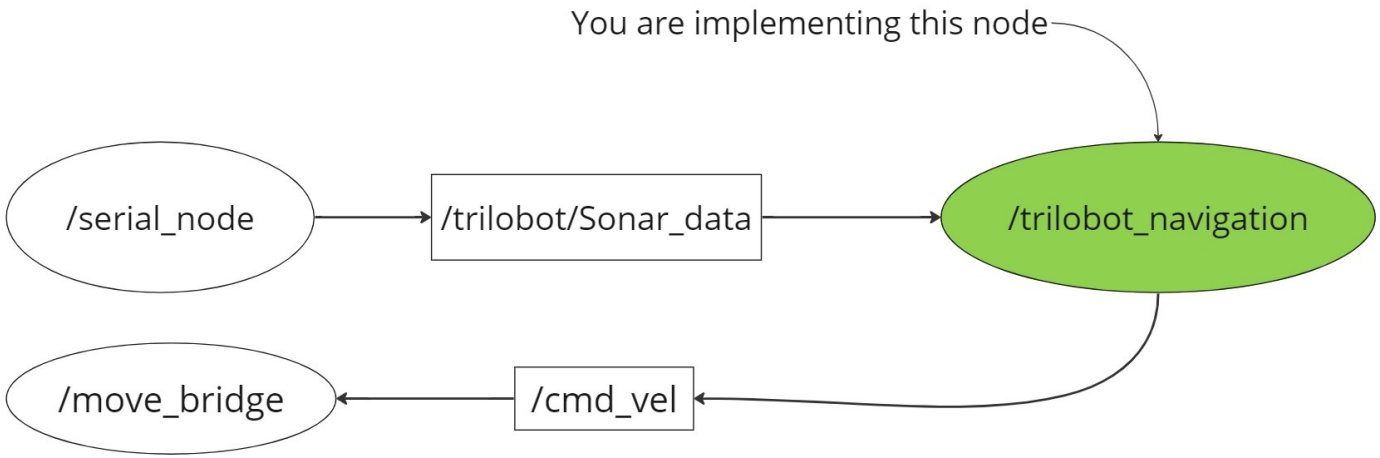
- Simplest obstacle avoiding algorithm can be described as following:



Simple obstacle avoiding algorithm

Program simple obstacle avoiding algorithm.

- Open file **trilobot_navigation/src/navigation.py**
- Implement node that reads sonar data and based on that controls motors
- Run your node with `rosrun trilobot_navigation navigation.py`
- Test your implementation by placing Trilobot on ground and placing an obstacle in front of it.



Pygame

- Pygame is a Python library that is used for making games
- In this laboratory it is used for getting key inputs
- https://www.pygame.org/news

Pygame - Initialization

- Before getting key inputs, Pygame needs to be initialized
- Pygame opens new window, this window needs to be in focus in order to get pressed keys

```
import pygame
from pygame.locals import *

# Initialize pygame window
pygame.init()
screen = pygame.display.set_mode((640, 480))
pygame.display.set_caption('Pygame - Trilobot')
pygame.mouse.set_visible(1)
```

Pygame - Getting pressed keys

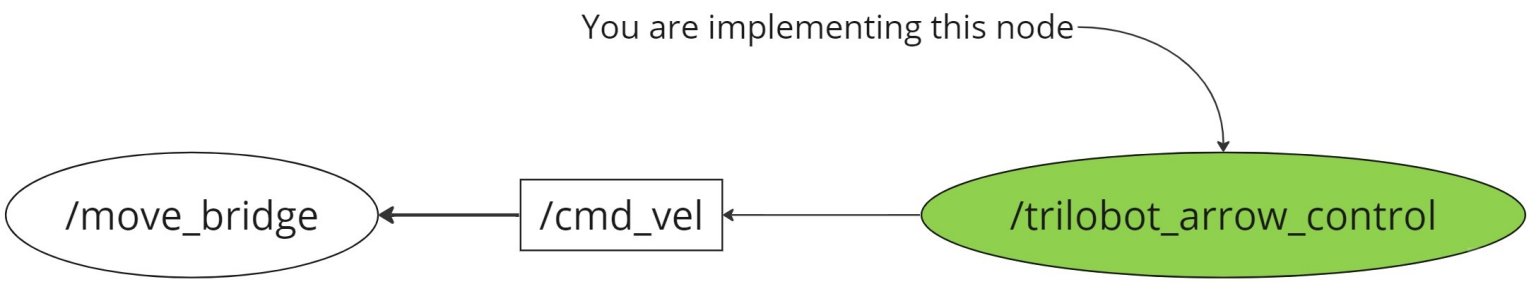
- Pressed keys can be obtained by function pygame.key.get_pressed()
- Function returns array of true/false for every key, true means key is pressed
- As indexes into array, following constants should be used: K_UP, K_DOWN, K_LEFT, K_RIGHT and K_LSHIFT

```
keys = pygame.key.get_pressed()

# If UP arrow is pressed
if keys[K_UP]:
    print("UP Arrow has been pressed!")
```

Make it possible to control Trilobot using keyboard arrows.

- Open file **trilobot_arrows/src/arrows.py**
- Implement arrow control using pygame library. Dictionary keys for keyboard keys are: K_UP, K_DOWN, K_LEFT and K_RIGHT.
- Run your code by running python command or running it from IDE (Do not run it using rosrun).
- Test your implementation by placing Trilobot on the ground and pressing keyboard arrow keys
- BONUS: make robot go even faster when left shift (K_LSHIFT) is pressed



Thank You For Your Attention !