

Ast x Oru Robotics Summer School

Andreas Persson, David Caceres Dominguez Pedro Zuidberg Dos Martires

Örebro University

github.com/pedrozudo/astxoru-roboticssummerschool

Robot = Forced Labour (in Czech)

Robot = Forced Labour (in Czech)



Robot = Forced Labour (in Czech)



Challenges in Robotics

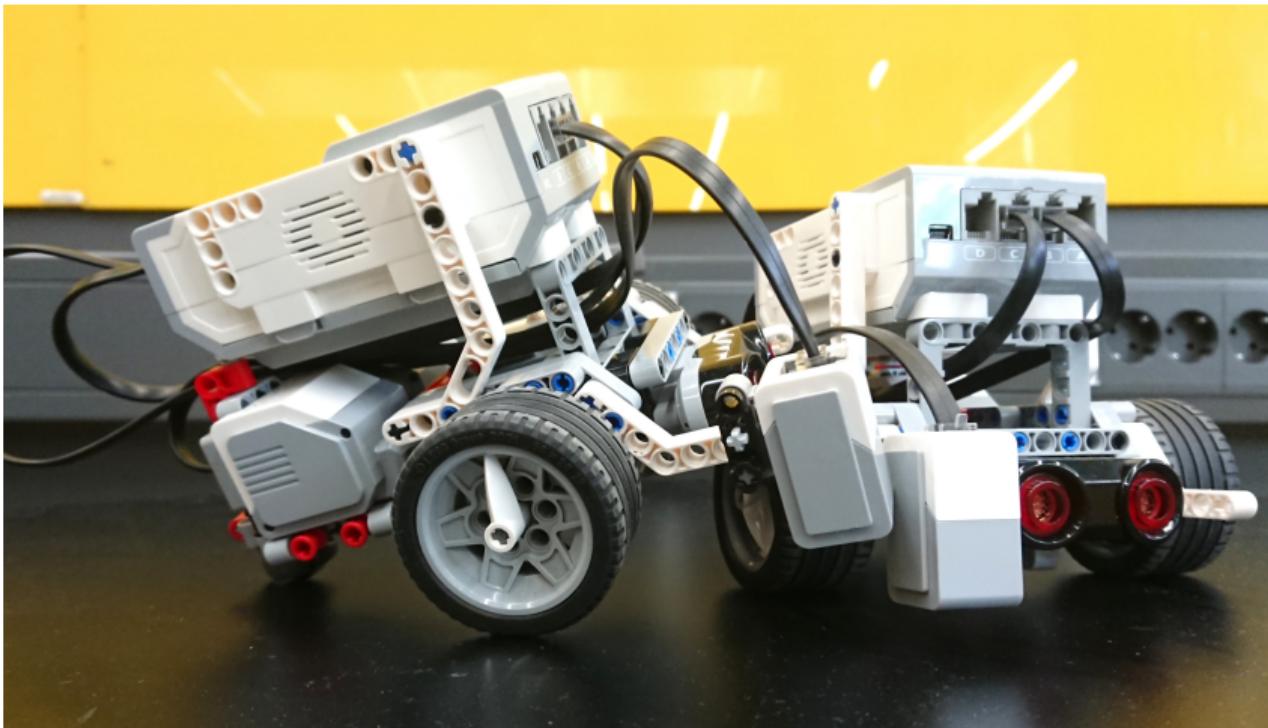
- building robots
- simulating robots
- robot control
- sensing
- inter-robot collaboration
- robot automation
- human-robot interaction
- ...

Challenges in Robotics

- building robots
- simulating robots
- robot control
- sensing
- inter-robot collaboration
- robot automation
- human-robot interaction
- ...

Even bigger problem: solve all at the same time

Our Robots



Coding in Python

Built-In Types

```
# This is a comment
n1 = 1 # an integer
n2 = 3 # an other integer

f1 = 0.5 # a float
f2 = 5. # an other float

b1, b2 = true, false # Booleans

s1 = "I am a string"
```

Collections

```
l = [1, 2, 10, "some-string"] #list  
  
t = (2, 4, 6.3) #tuple  
  
d = {"key1" : 5, 2 : "string value"} #dictionary/hashmap
```

Collections

```
l = [1, 2, 10, "some-string"] #list  
  
t = (2, 4, 6.3) #tuple  
  
d = {"key1" : 5, 2 : "string value"} #dictionary/hashmap  
  
l[0] #gives 1  
l[1] #gives 2  
l[-2] #gives 10  
l[3] = 5 # mutates the list to [1,2,10,5]
```

similar for tuples

Collections

```
d = {"key1" : 5, 2 : "string value"} #dictionary/hashmap  
  
d["key1"] # gives 5  
d[2] # gives "string value"  
  
d[2] = 10 # mutates dictionary into {"key1" : 5, 2 : 10}
```

For Loops

```
l = [1, 2, 10, "some-string"]
for i in l:
    print(i)
```

This will return the following:

```
1
2
10
"some-string"
```

For Loops

```
s = "some-string"  
for i in s:  
    print(i)
```

For Loops

```
s = "some-string"  
for i in s:  
    print(i)
```

This will return the following:

```
s  
o  
m  
e  
-  
...
```

Conditional Statements

```
b1 = True  
b2 = False  
if b1 and b2:  
    print("okay")  
else if not (b1 and b2) :  
    print("not okay")  
else:  
    print("somewhat okay")
```

Functions

```
def is_pair(x):
    if x % 2:
        return True
    else:
        return False
```

Classes

```
class Rectangle:  
    def __init__(self, length, width):  
        self.length = length  
        self.width = width  
  
    def area(self):  
        return self.length * self.width  
  
    def perimeter(self):  
        return 2 * self.length + 2 * self.width  
  
class Square(Rectangle):  
    def __init__(self, length):  
        super().__init__(length, length)
```

Classes

```
r = Rectangle(3,4)  
s = Square(4)  
  
print(r.area())  
print(s.perimeter())  
print(r.length)
```

The VS Code IDE

code0server: VS Code in the Browser

17:17 Mon 13 Jun

127.0.0.1

Microsoft Teams Standard de...ss Validated Gymsystem PaperCut MF Usage - co...v4.4.0 docs A Complete...- Dataquest Get Started...Code - OSS Fundament...- Wikipedia

astxoru_robotsics_lecture1 - Online LaTeX Editor Overleaf debd_trainer.py — distle — Code - OSS

EXPLORER

DISTLE

- datasets
 - __pycache__
 - DEBD
 - MNIST
- debd.py
- distle
 - __pycache__
 - nodes
 - __pycache__
 - __init__.py
 - bernoulli.py
 - nodes.py
 - product.py
 - shuffle.py
 - sum.py
- lightning_logs
- venv
- .gitignore
- debd_trainer.py
- LICENSE

108 def predict_dataloader(self):
109 return DataLoader(
110 self.predict_data,
111 batch_size=self.hparams.batch_size,
112 num_workers=self.hparams.num_workers,
113)
114
115
116 if __name__ == "__main__":
117
 clt = LightningCLI(
 DebdLitProbModel,
 seed_everything_default=42,
 save_config_overwrite=True,
)
 result = clt.trainer.test(clt.model, datamodule=clt.datamodule)
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

bash + ▾ □ ^ ×

tensor(-218.6564, device='cuda:0') | 22/25 [00:00<00:00, 72.52it/s]
Epoch 0: 98%|██████████| 123/125 [00:15<00:00, 8.05it/s, loss=nan, v_num=3, train/loss_step=5.15e+3, train/logp_step=-4.35e+3, tr
tensor(-212.7201, device='cuda:0') | 23/25 [00:00<00:00, 72.52it/s]
Epoch 0: 99%|██████████| 124/125 [00:15<00:00, 8.11it/s, loss=nan, v_num=3, train/loss_step=5.15e+3, train/logp_step=-4.35e+3, tr
tensor(-210.9488, device='cuda:0') | 24/25 [00:00<00:00, 72.52it/s]
Epoch 0: 100%|██████████| 125/125 [00:15<00:00, 8.16it/s, loss=nan, v_num=3, train/loss_step=5.15e+3, train/logp_step=-4.35e+3, tr
LOCAL_RANK: 0 - CUDA_VISIBLE_DEVICES: [0]
Testing DataLoader 0: 100% | 75/75 [00:03<00:00, 20.95it/s]

Test metric Data loader 0

15

The Command Line

Programming Robots

Robot Operating System

The Robot Operating System (ROS) is a flexible framework for writing robot software. It is a collection of tools, libraries, and conventions that aim to simplify the task of creating complex and robust robot behavior across a wide variety of robotic platforms.

ros.org (general information)

wiki.ros.org (practical information)

Robot Operating System

- A standard (more or less) for writing robot software.
- Encourages modular solutions.
- Targets different levels of abstraction:

Low: Communications infrastructure

Mid: Robot-Specific components (e.g. specific sensors)

High: Tooling (graphical interfaces, command-line tools, . . .)

Robot Specific Components

- Standard robot messages,
- Robot geometry library,
- Robot description language,
- Diagnostic tools,
- Pose estimation, localization, and Navigation,

Robot Specific Components

- Standard robot messages,
- Robot geometry library,
- Robot description language,
- Diagnostic tools,
- Pose estimation, localization, and Navigation,
- ...support for, e.g.:



ROS Nodes

Nodes are program that communicate with each other.

- Message Passing (through "topics")
- Publisher (talker)
- Subscriber (listener)
- Uses ROS 'msg'-files

