Weekly Presentation

Week 40

Martin Blaszczyk
Edward Källstedt
Albin Martinsson
Måns Norell

Luleå University of Technology

September 22, 2020

Overview

1. Status update
   - Overall timetable
   - September planning
2. Hardware
   - New toys
3. Machine vision
   - Flows
   - Line following algorithm
4. Movable base
   - Arrowhead

Overall timetable

Sep	Oct	Nov	Dec
Concept generation	Evaluation	Evaluation	
Theory	Prototyping	Evaluation	Finishing up
Simulation	Evaluation	Evaluation	
Prototyping	Final Design	Evaluation	

Time plan for September

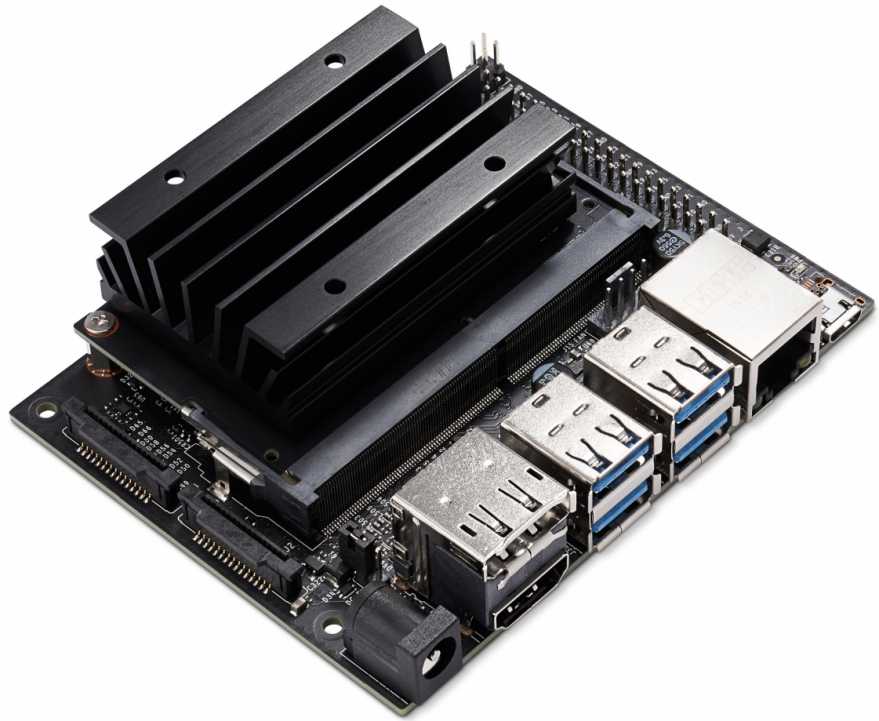
Subproject	Week 1	Week 2	Week 3	Week 4
Arrowhead	Reading	Setup	API	Prototyping
Movable base	Reading	Modeling	Simulation	Implementation
Arm and grip	Reading	Kinematics	Simulation	Prototyping
Object detection	Reading	Testing	Prototyping	Evaluation

New toys!

- NVIDIA Jetson Nano
- Cameras
- Dynamixel Smart motors
- Screws, cables and other goodies

NVIDIA Jetson Nano

- Runs Ubuntu
- Two camera ports (CSI)
- More powerful GPU than RPi



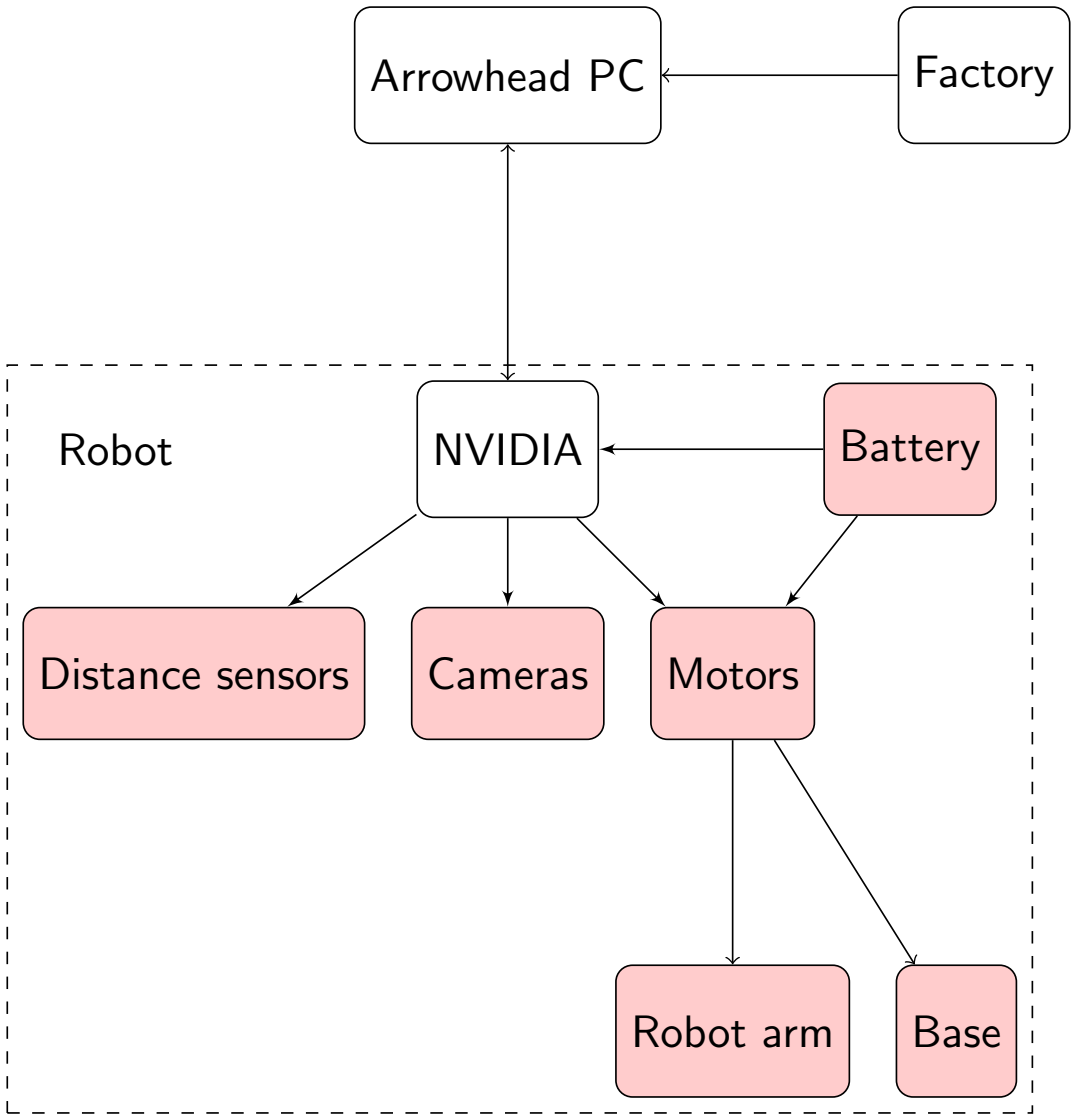
Cameras

- Compatible with NVIDIA and RPi
- Small package
- 8 megapixels
- Video:
  - 1080p @ 30 fps
  - 720p @ 60fps

Dynamixel Smart Motors

- Connects in series
- Angle and wheel mode
- Feedback

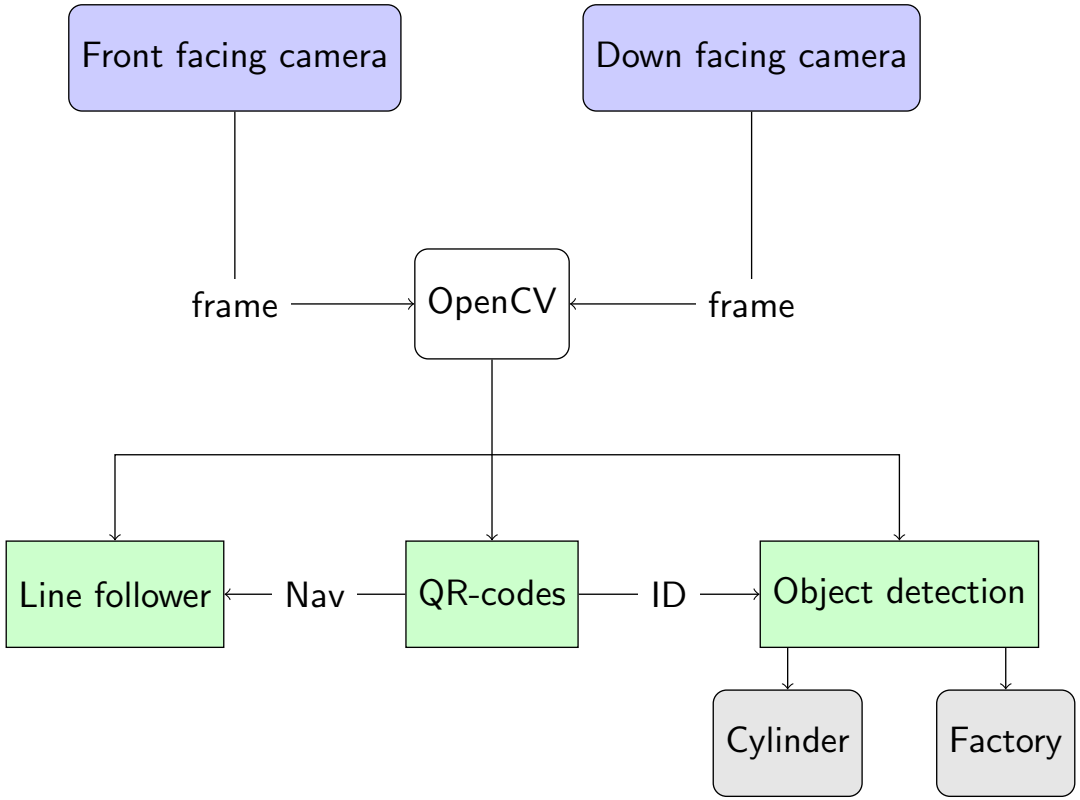
Hardware


Arrowhead PC
Factory
Robot
NVIDIA
Battery
Distance sensors
Cameras
Motors
Robot arm
Base

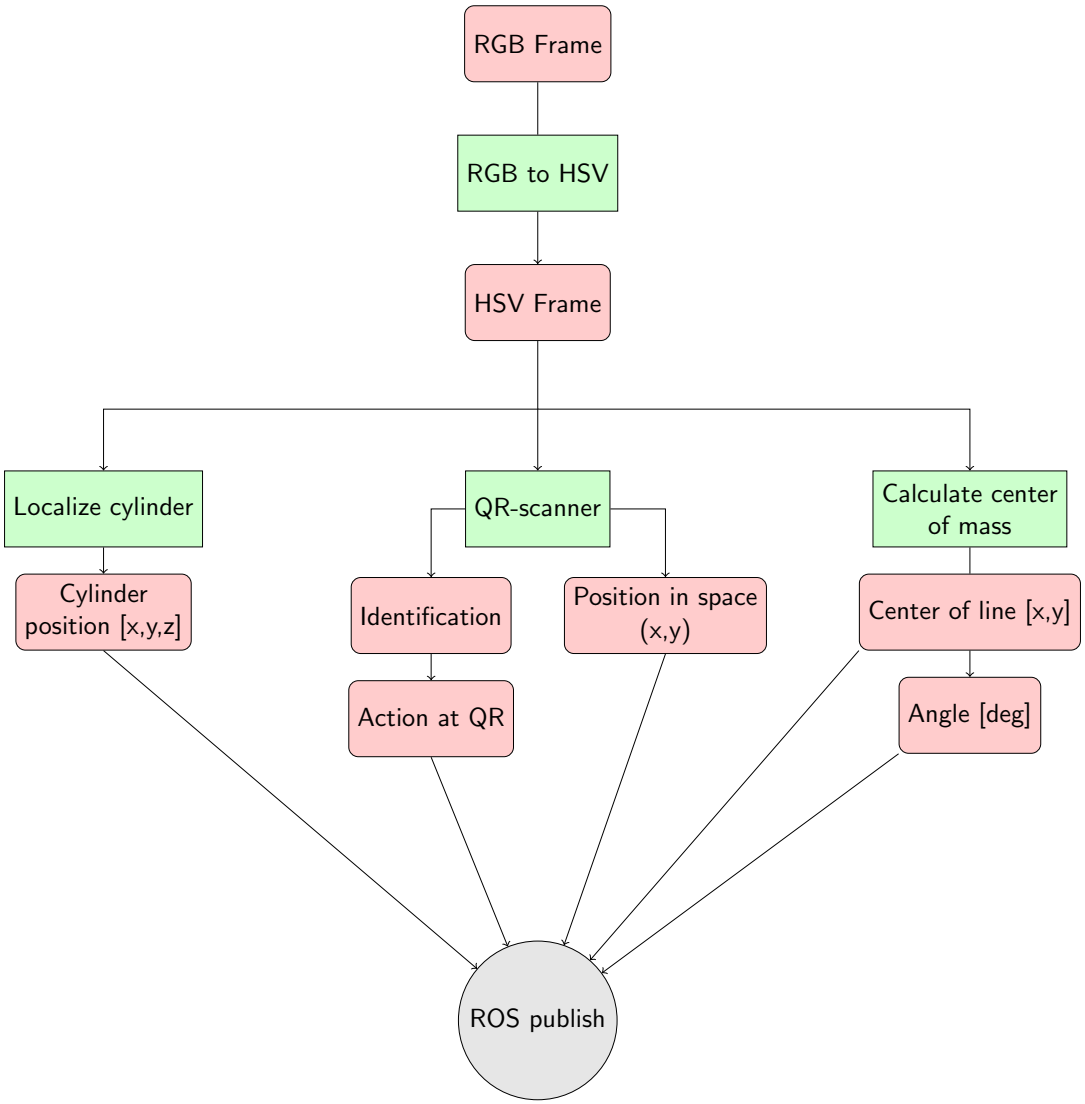
Object detection

- Line detection now works
- Testing cameras and real time performance on NVIDIA
- Will be using two cameras.
  - Front facing
  - Downwards facing

Machine vision



OpenCV


RGB Frame
RGB to HSV
HSV Frame
Localize cylinder
QR-scanner
Calculate center of mass
Cylinder position [x,y,z]
Identification
Position in space (x,y)
Center of line [x,y]
Action at QR
Angle [deg]
ROS publish

Line following algorithm

- https://www.youtube.com/watch?v=TdtPIdipSBY

Line following algorithm - Procedure

- HSV mask to select color space
- Crop image into horizontal slice
- Calculate image moment on slice to extract centroid
- Calculate angle between robot base and line

Line following algorithm - Optimization

- Only search adjacent to previous line position
- Implement CUDA support

CV - What's next?

- Implement QR code functionality
- Finalize API consumed by other parts of the system
- Clean up code and implement proper configuration capabilities

Movable Base

What has been done

- Mathematical model
- Controller selection
- Some simulations (more work needs to be done)

Mathematical Model

Actuators:

- The motors of the left and right track (v_L, v_R)

Parameters to Control:

- The angle to a coloured line (θ)

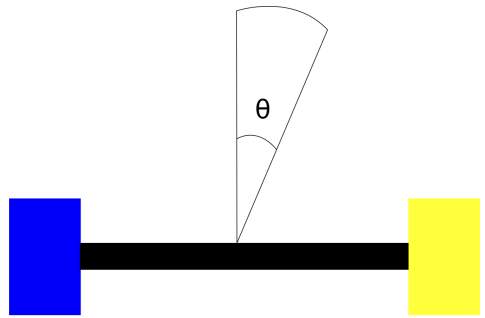


Figure: Model of base

Mathematical Model

Given the v_L, v_R and θ we can calculate

- $\overline{v} = ||v|| * (-\cos\theta * \hat{i} + \sin\theta * \hat{j})$
- $\overline{L} = \int \overline{v}$
- $\dot{\theta} = \frac{v_R - v_L}{2r}$

Controller Choice

The robot will use two **PD** controllers for the line following

Controller Choice

The robot will use two **PD** controllers for the line following

- Simple **P** controller is not smooth enough
- **PID** is more likely to overshoot

Simulations

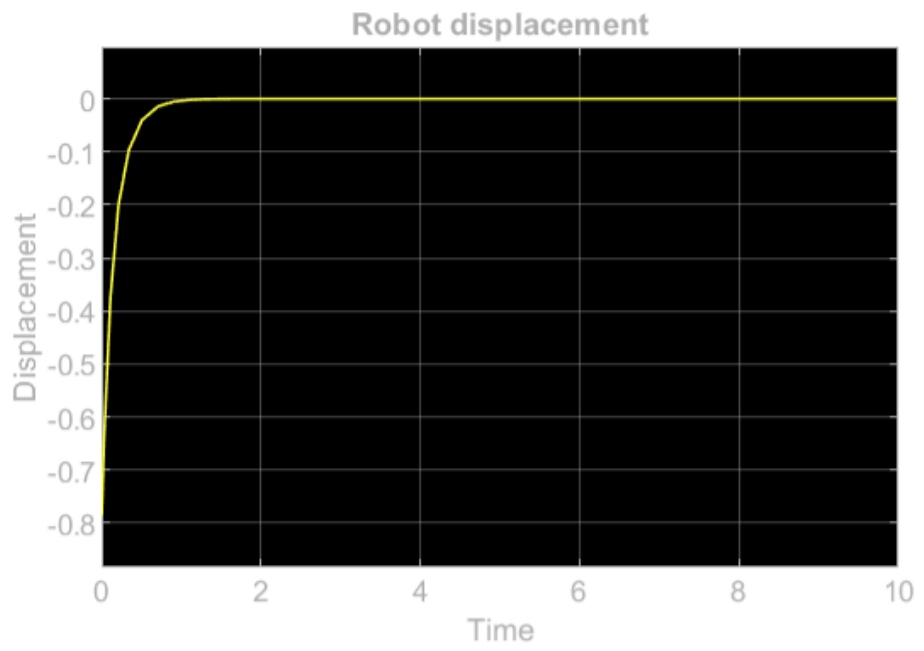


Figure: Angle from line

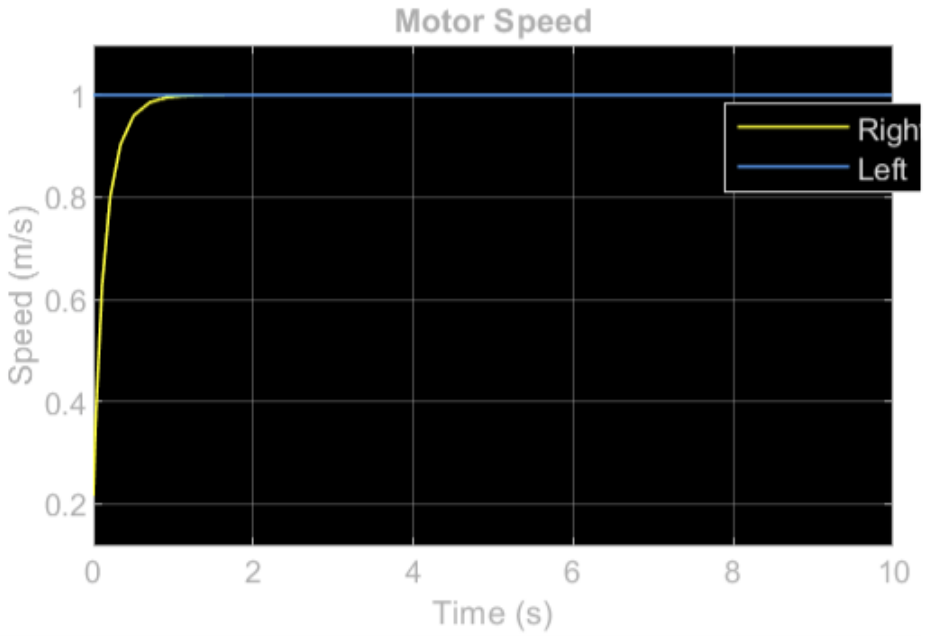


Figure: Motor speed for left and right motor

Simulations

To be done

- Import real values from the motors and base for applicable simulations
- Insert boundaries for control signals

Arrowhead

Questions?