,

Hochschule
Bonn-Rhein-Sieg
University of Applied Sciences

Bonn-Aachen
International Center for
Information Technology

Software Development Project

Motion primitives for Freddy

12th September, 2022

Bharath Kumar Adinarayan
Kevin Patel
Wing Ki Lau
Yashika Garg

Advisors

Sven Schneider

Definition

Motion primitives[1], skill or behavioural building block:

- atomic unit of robotic behaviour
- configurable and hence reusable in different contexts
- composable with other motion primitives
  to build more complex behaviour

[1] **Kinematic constraints and motion primitives** 2022.

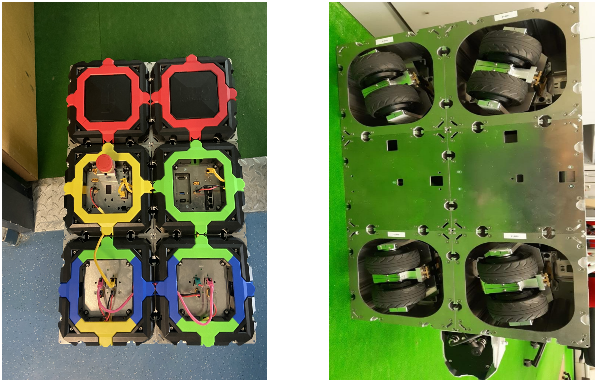
Robile platform of Freddy: Capabilities

- Modular mobile robot platform
- Four identical pair of wheels
  which can be actuated independently
- Communication with wheel-units (in master-slaves architecture) is made over EtherCAT
- Available sensors: motor encoder, IMU (Inertial Measurement Unit) - consists of gyroscope and accelerometer

Robile platform of Freddy: User interface

- Control interface: velocity(power limited) and force(current controlled)
- Programming language: C

Robile platform



(a) Top view (b) Bottom view

Figure 1: Top and bottom view of Robile platform

Problem definition

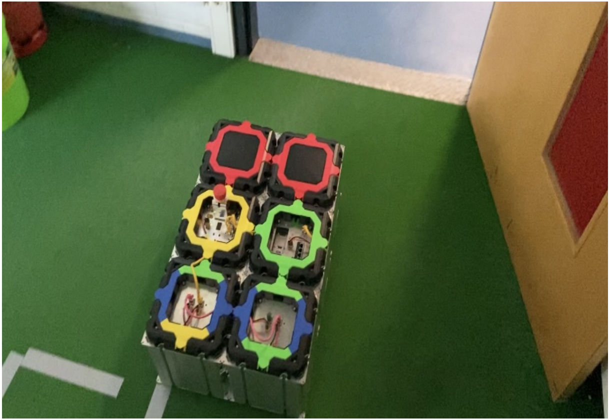
Controlled movement over the ramp using motion primitives for the Freddy robot

Project goal

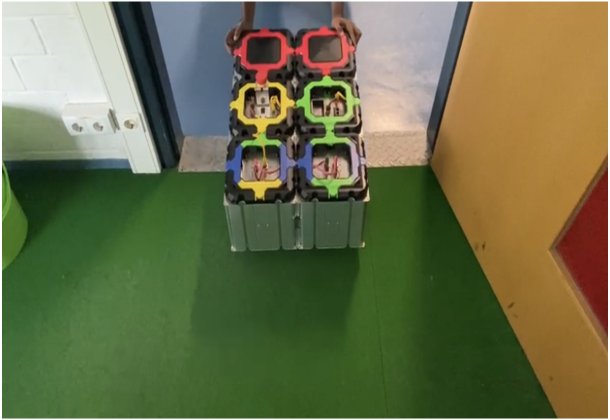
Develop a control interface for Freddy robot to perform a motion over the ramp [a]

[a] Considering environment is static

Velocity control - video



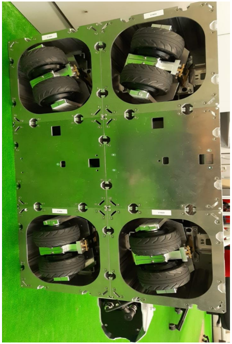
Safe ramping behaviour - video



Wheel configurations

- Possible different configuration of the platform i.e., 4 wheeled , 2 wheels + 2 castors, tricycle etc.

(a) 2 active wheels and 2 castor wheels



(b) 4 active wheels

Required libraries

- Simple Open EtherCAT Master (SOEM) - communication between robot and the actuators.[2]
- robif2b - robot control interface[3]
- GSL - GNU Scientific Library[4]
- WS21 SDP repository: Motion Control of the KELO 500[5]

[2] OpenEtherCATsociety 2022.

[3] Rosym-Project 2022.

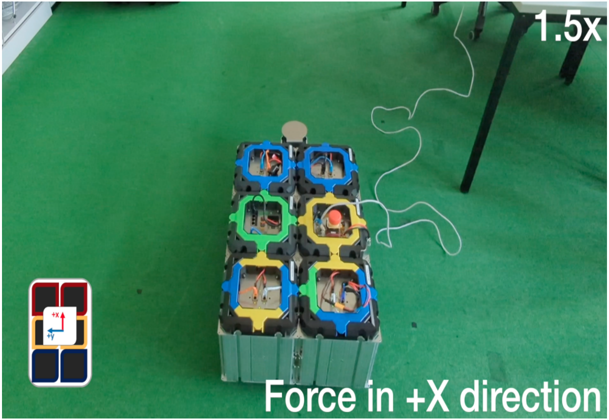
[4] Dr M. Galassi 2021.

[5] **ws21-kelo-500-motion-control** 2021.

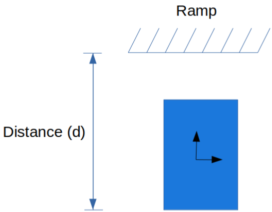
User story 1

Unique Identifier : D1	**Estimate :** 2 weeks
Task Description : • Understand force control distribution on Freddy robot. • Test previous SDP code on Freddy. • Evaluate overlap between previous SDP and ramping behaviour. • Refactoring the existing code.	**Acceptance Criteria :** • 1. Move robot in translational forward/backward and left/right as well as rotational clockwise/ anticlockwise manner. • 2. Drive up the ramp with the different platform-level force setpoints.
Risk : Low	**Real Effort :** 2 weeks

Acceptance criteria 1 - video



Robot distance w.r.t. ramp



Distance between robot and ramp

Robile platform

Acceptance criteria 2 (d=250cm) - video

Acceptance criteria 2 (d=125 cm) - video

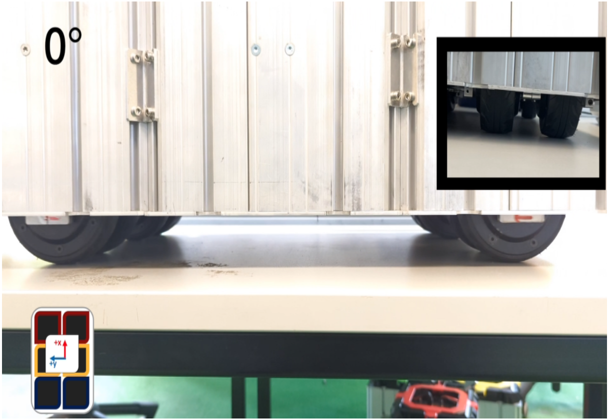
User story 2

Unique Identifier : D2	**Estimate :** 2 weeks
Task Description : • Orient wheel units to the desired configuration w.r.t. base of the robot.	**Acceptance Criteria :** • Bring the wheels to 0, 90, 180 and 270 degrees.
Risk : High	**Real Effort :** 2 weeks

Wheel orientation

- Robile platform has differential drive wheel units
- Using differential drive kinematics to rotate wheels around z-axis (no translation motion)
- Give both wheels same magnitude of torque but in opposite direction
- Controller:
  - Closed-loop controller
  - Set points are 0, 90, 180, and 270 degrees
  - Feedback signal from pivot angle measurement sensor (in rad)
  - Convert angle values in range of -PI to +PI
  - Find the shortest rotation direction
  - Implemented only Proportional (P) controller as of now

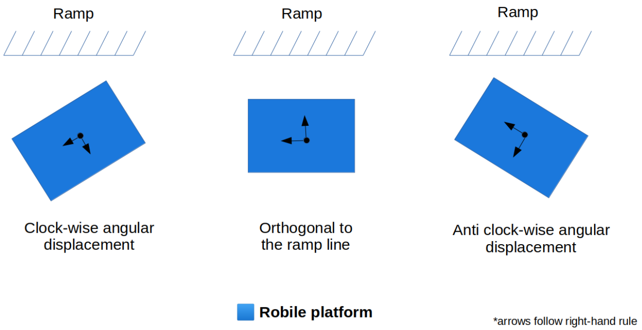
Acceptance criteria - video



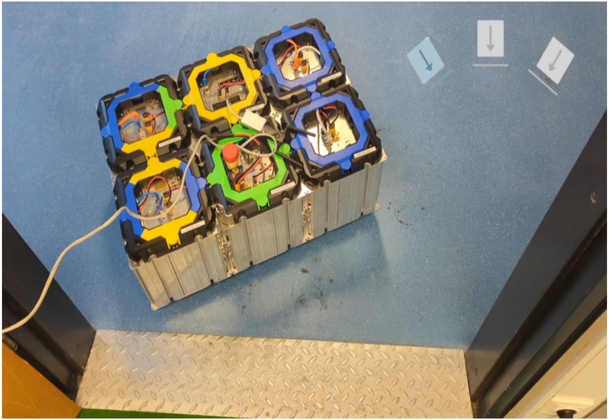
User story 3

Unique Identifier : D3	**Estimate :** 3 weeks
Task Description : • Align the robot with a ramp baseline.	**Acceptance Criteria :** • The robot should be aligned with the baseline of the ramp from different starting orientations: –> w.r.t. to the baseline of the ramp, 1) clockwise angular displacement, 2) anti–clockwise angular displacement, and 3) orthogonal to the line.
Risk : High	**Real Effort :** 3 weeks

Robot alignment w.r.t. ramp



Acceptance criteria - video



User story 4

Unique Identifier : D4	**Estimate :** 4 weeks
Task Description : • Understand the ramp-up behaviour for the Freddy robot. • Implement ramping behaviour on the robot.	**Acceptance Criteria :** • The robot should be able to complete the ramp slope. • The robot should be able to safely stop after finishing the ramp. • The robot should also be able to run over a small bump (ramp up and down).
Risk : High	**Real Effort :** 5 weeks

User story 4 explanation

- Proportional control based ramp up motion
- Stopping criteria is based on either distance, or velocity
- $f_{value} = f_{max} - d * vel_{w_i}$
- Where f = force(N),
  d = damping constant (ratio of f_{max} and velocity threshold),
  vel = wheel velocity (m/s),
  i = index of wheel

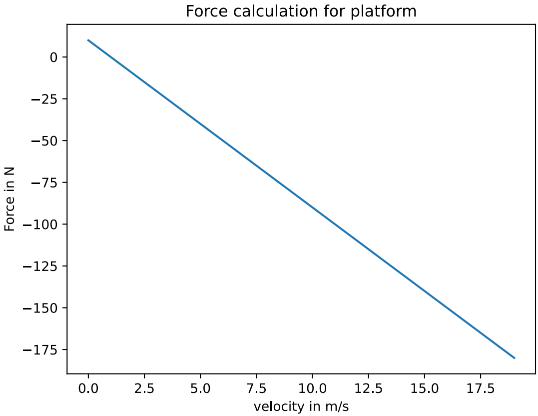

Force calculation for platform
Force in N
velocity in m/s
0
−25
−50
−75
−100
−125
−150
−175
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5

Figure 2

User story 5

Unique Identifier : D5	**Estimate :** 2 weeks
Task Description : • Integrate all sub–modules as a complete state machine.	**Acceptance Criteria :** • The robot should be able to autonomously drive over the ramp. • The robot should be stopped after the finishing ramp behaviour.
Risk : High	**Real Effort :** 1 week

State machine diagram

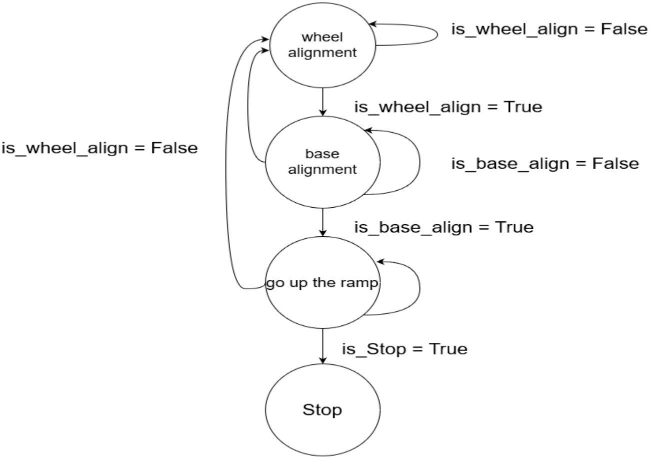


Figure 3: State machine

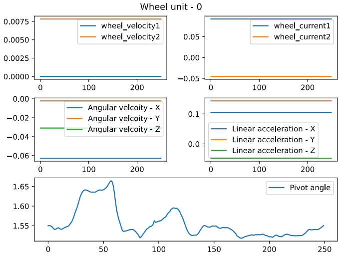
Features

- Force distribution(previous SDP) code as a library
- Force control behavior allows controlled movement and safe operation in the environment
- Active wheel control
- Proportional controller based ramping behaviour
- Using libraries as a package (modular code)
- Smooth ramp up behaviour

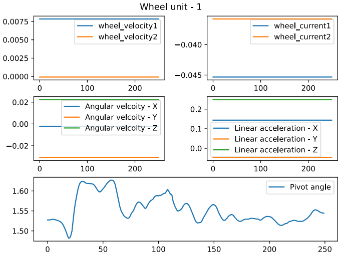
Challenges

- Realise the physical expected behaviour on coding
- Hardware issues - wheel malfunction
- Instability associated with proportional controller
- Uneven surfaces cause issues during wheel alignment
- Aligning robot vertically with the base of the ramp
- Rotation w.r.t the corner of a robot
- Using IMU sensor for ramping behaviour
- Misalignment of the wheels on the ramp
- Iteration based code

Challenges (continue)

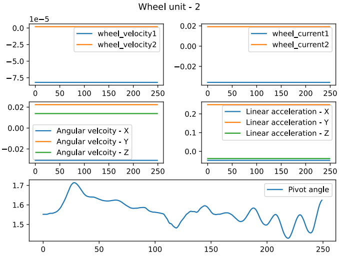


(a) Wheel unit 0

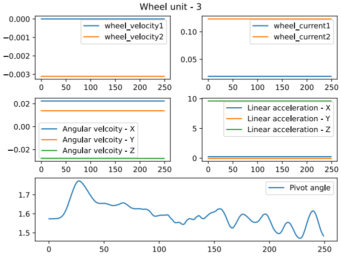


(b) Wheel unit 1

Challenges (continue)



(c) Wheel unit 2



(d) Wheel unit 3

Collaboration plans

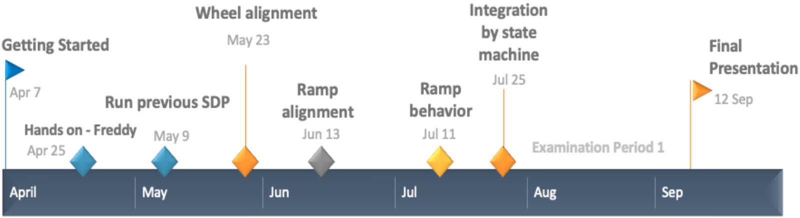


Figure 4: Project Roadmap

- GIT version control: Motion Primitive Freddy repository
- Communication medium: Slack
- Meeting frequency: internal meeting twice a week and with the advisor, every Thursday (in-person/online)

Future work

- Implement more robust controllers, example - PID, Fuzzy logic (FLC), etc.
- Use of various sensors for assisting the ramping behaviour, example - Lidar, camera, etc
- Generalisation of the current implementation for different wheel configurations and different types of ramps
- Moving the robot over the ramp in vertical orientation
- Using smart pointers in state machine implementation

Experiences

- Working with C project structure and building custom libraries
- Integrating sub modules on github repository
- Velocity vs Force control behaviour
- Learnt new wheel kinematics
- Learnt about PID controller
- Working with low-level hardware/software stack
- Refactoring existing code base

References

- Bruyninckx, Herman (2022). **Building blocks for complicated and situational aware robotic and cyber-physical systems**. Accessed on 21.04.2022. URL: https://robmosys.pages.gitlab.kuleuven.be/composable-and-explainable-systems-of-systems.pdf.
- Dr M. Galassi, Dr J. Theiler (2021). **GSL - GNU Scientific Library**. Accessed on 21.04.2022. URL: https://www.gnu.org/software/gsl/.
- **Kinematic constraints and motion primitives** (2022). Accessed on 21.04.2022. URL: http://sbpl.net/node/48.
- OpenEtherCATsociety (2022). **OpenEtherCATsociety/Soem: Simple Open Source ethercat master**. Accessed on 21.04.2022. URL: https://github.com/OpenEtherCATsociety/SOEM.
- Rosym-Project (2022). **ROSYM-project/ROBIF2B: Building blocks for robot interfaces**. Accessed on 21.04.2022. URL: https://github.com/rosym-project/robif2b.
- **ws21-kelo-500-motion-control** (2021). Accessed on 21.04.2022. URL: https://github.com/HBRS-SDP/ws21-kelo-500-motion-control.

Thank you for your attention!
Questions?