Hochschule
Bonn-Rhein-Sieg
University of Applied Sciences

b-it Bonn-Aachen International Center for Information Technology

Software Development Project

Motion primitives for Freddy

Bharath Kumar Adinarayan
Kevin Patel
Wing Ki Lau
Yashika Garg

Advisors

Sven Schneider

Definition

Motion primitives: Pre-computed motions that the robot can take

Freddy robot

[1]

- 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, gyroscope, accelerometer, IMU (Inertial Measurement Unit)
- Motion control: velocity and force
- Programming language: C

[1] EBuilding blocks for complicated and situational aware robotic and cyber-physical systems

(a) Top view

(b) Rear view

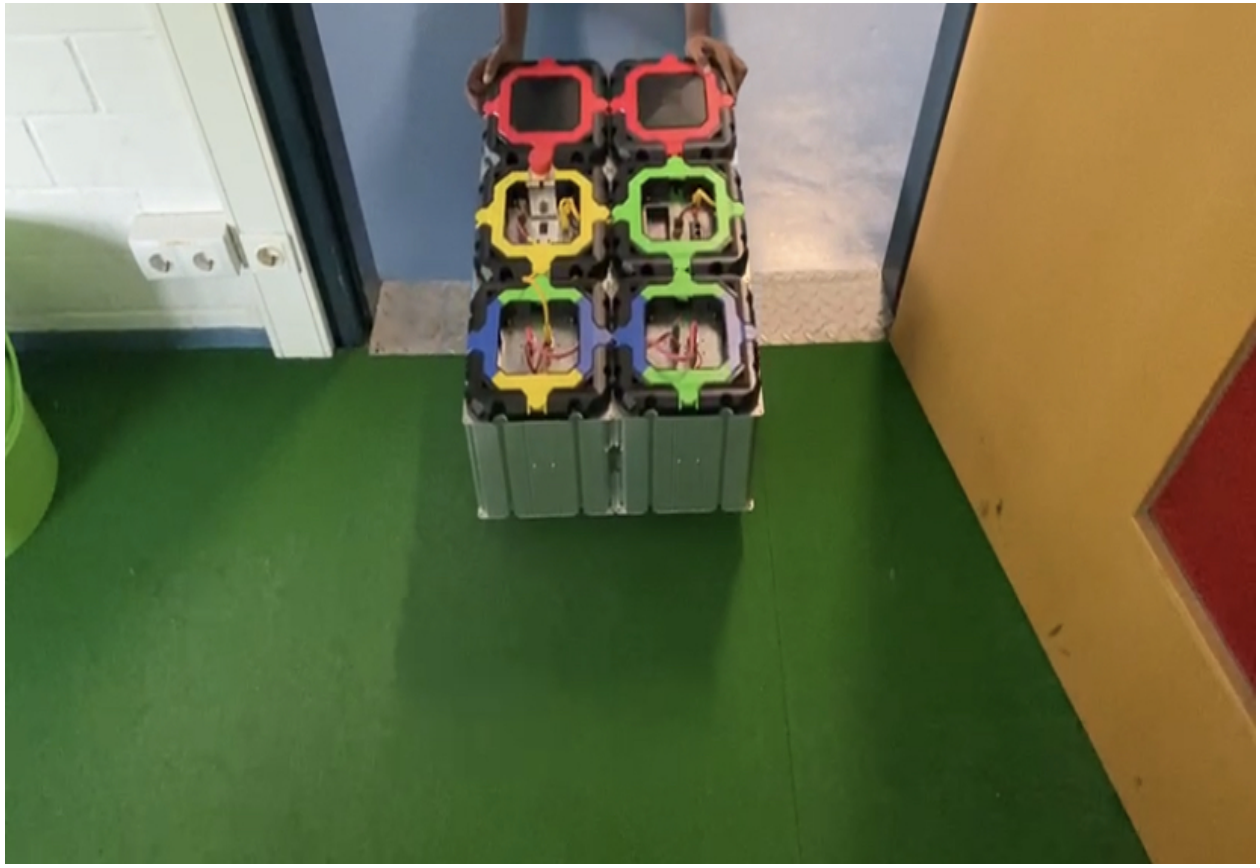
Figure 1: Top and rear view of Robot Freddy

Problem definition

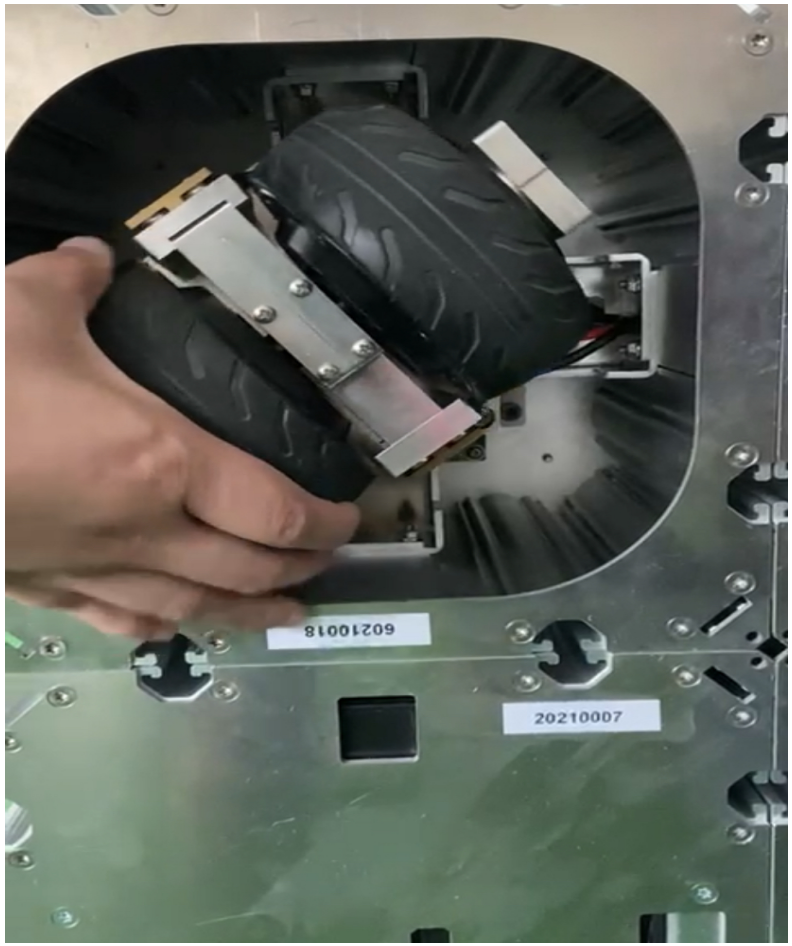
Safe ramping behaviour of Freddy robot using motion primitives.

Velocity control - video

Safe ramping behaviour - video



Push/pull configuration - video



Wheel configuration

Top view of 2 castor and 2 active wheels configuration

Project goal

Successfully perform the ramping motion on the Freddy robot safely.

Required libraries

- Simple Open EtherCAT Master (SOEM) - communication between robot and the actuators. [2]
- robif2b - robot control interface [3]

[2] SOEM github repository
[3] robif2b github repository

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 • Implement relevant code required for ramping.	**Acceptance Criteria :** • The code should run successfully on Freddy. • The overlapped code should be implemented for ramping behaviour.
Risk : Minimum	**Real Effort :**

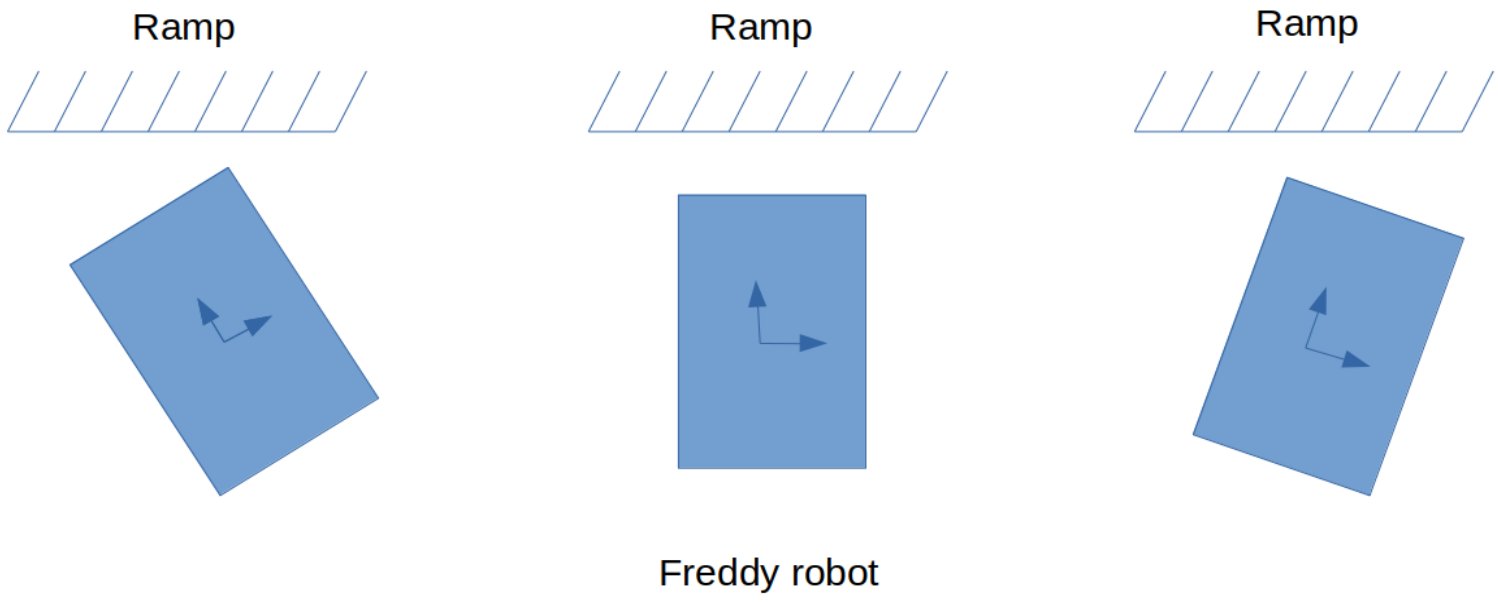
User story 2

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 :	**Real Effort :**

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) positive slope, 2) negative slope, and 3) parallel to the line.
Risk :	**Real Effort :**

Robot alignment w.r.t. ramp



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 :	**Real Effort :**

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 :	**Real Effort :**

Collaboration plans

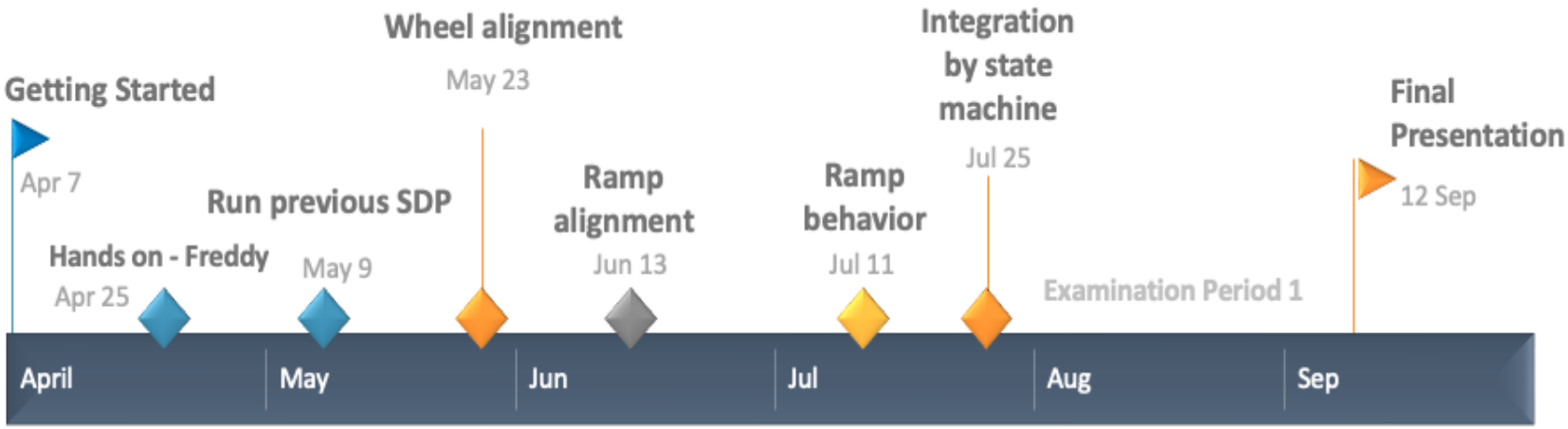


Figure 2: Project Roadmap

Version Control : GIT: Motion Primitive Freddy repository
Communication medium: Slack
Meeting frequency : Internal meeting twice a week and with the Advisor, every Monday (in-person/online)