HUMBOLDT-UNIVERSITÄT ZU BERLIN
MATHEMATISCH-NATURWISSENSCHAFTLICHE FAKULTÄT
INSTITUT FÜR INFORMATIK
KOGNITIVE ROBOTIK SOSE 2024

# 5. Übung - Zweibeiniges Laufen

*Authors:*
Goertz, Keno [572298]
Gyuler, Berin [635008]
Suttar, Artem [634758]

2024/07/04
Berlin

# 1 Subtask 1 (walk_fast_params.py)

The combinations of parameters suggested in the comments were considered:

| Combination | Time (min) | Observation |
|---|---|---|
| 1. Slow and Steady | $\sim 1.50$ | The robot maintains a steady pace with no significant deviations. |
| 2. Walk It | $\sim 1.11$ | The robot swings slightly to the left. |
| 3. Larger Steps | $\sim 0.51$ | The robot swings hard to the left. |
| 4. Tippel-Tappel | $\sim 0.51 - 0.52$ | The robot stays in the middle. |
| 5. Run!!! | $\sim 0.27$ | The robot swings hard to the right. |

Table 1: Robot Movement Parameters and Observations

The idea going forward is to work with values from 4 and 5, as they are the most successful. The best of the options would be to use the parameters of option 5, but stabilize them.

| Combination | Time (min) | Observation |
|---|---|---|
| 1. Version | $\sim 0.29$ | Better stabilization. |
| 2. Version | $\sim 0.24$ | Better stabilization + speed |
| 3. Version | $\sim 0.21$ | Best speed, robot comes too close to the line. |

Table 2: Parameters of robot motion. Results

| Version | f | robot_height | shift_y | step_height | step_length | arm_swing |
|---|---|---|---|---|---|---|
| 1 | 19 | 0.5 | 0.11 | 0.6 | 0.3 | 1.5 |
| 2 | 20 | 0.5 | 0.11 | 0.6 | 0.35 | 1.4 |
| 3 | 20 | 0.5 | 0.11 | 0.6 | 0.4 | 1.4 |

Table 3: Parameters for each version

So far, version number two has proven to be the best, as it has no issues with the robot going over the sidelines.

# 2 Subtask 2 (walk_side_steps.py)

The idea for accomplishing the task is very simple:
Step 1: Shift weight to the left leg
Step 2: Lift right leg
Step 3: Move right leg to the side
Step 4: Place right leg down
Step 5: Shift weight to the right leg
... Continue for the left leg ...
In the code provided, we already have a way to lift the legs in sequential order(**zLeft**, **zRight**).
It's necessary to stop moving forward: **xLeft = 0** and **xRight = 0**.
Only with the code provided, Step 3 doesn't work - the hips move in sync and this needs to be

changed. Instead of **yLeftRight** we use the variables **yLeft** and **yRight**.
Now you need to choose which direction to move. The sine will be the leading side. This means that the hip whose variable has the value of sine will be the leading one and the robot will move in this direction. Example for right-side steps:
**yLeft = math.cos(t)*shift_y**
**yRight = math.sin(t)*shift_y**
In the future, you can use this approach to adjust the robot's position in the lane.

## 3 Subtask 3 (walk_turn.py)

**Not done**

## 4 Subtask 4 (walk_sensor.py)

**Not done**