



Rheinische
Friedrich-Wilhelms-
Universität Bonn

Prof. Dr. Maren Bennewitz

Institut für Informatik
Abteilung VI
Humanoid Robots Lab
Adresse:
Friedrich-Hirzebruch-Allee 8
53115 Bonn

Humanoid Robotics

Assignment 11

Due Thursday, July 17th, before class.

Legged Robots Locomotion

1. Footstep Planning with A* Search (Total: 9 points)

In the lecture, we discussed how the A* algorithm can be used to plan paths efficiently by combining actual path cost and heuristic estimates. This idea can be extended to footstep planning for humanoid or biped robots, where the state includes not only position but also which foot is currently on the ground.

In this exercise, we will implement a simplified A* footstep planner on a 2D grid world. The state is represented as a tuple (x, y, foot) , where (x, y) is the current grid cell, and foot is either "left" or "right", indicating the stance foot.

The robot alternates feet at every step and can move in 8 directions. Obstacles are present in the grid. Your goal is to find a feasible and efficient sequence of footsteps from a start state to a goal state.

- a. Implement the Heuristic Function (2 points)
Complete the `heuristic()` function using the Euclidean distance. This function guides the A* search toward the goal.
- b. Define the Neighboring Footsteps (2 points)
Implement the `get_neighbors()` function. It should return a list of valid next states and their corresponding costs. Ensure that you:
 - Alternate the stance foot (left -> right -> left),
 - Avoid stepping into obstacles,
 - Only allow valid steps (within grid bounds).
- c. Implement the A* Planner (3 points)
Complete the main `a_star_footstep()` function. This function should:
 - Track visited states and their costs,
 - Expand the most promising state at each step,
 - Reconstruct the path once the goal is reached.
- d. Visualization and Interpretation (2 points)
Once your implementation is complete, use the provided plot function to visualize the resulting footstep plan.
You should:
 - Observe the alternating foot placements (left in blue, right in green),
 - Verify that the path avoids obstacles,
 - Interpret how the footstep alternation and grid geometry affect the final trajectory.
 - What happens if you change the cost model (e.g., penalize diagonal steps)?



2. Operational Space Control

(Total: 6 points)

In this exercise, we will explore **Operational Space Control (OSC)** for a 3-DoF planar robot, where all links have equal length $l_1 = l_2 = l_3 = 1 (m)$. The task is to control the end-effector position (e.g., target position) while respecting the desired posture of the robot. To achieve this, we use Iterative Null Space Projection to prioritize tasks.

Primary task: Control the end-effector to reach a target position.

The control input is the joint velocity $\dot{\mathbf{q}}$. The desired end-effector velocity $\dot{\mathbf{x}}_{des}$ is related to joint velocities via the Jacobian $J_1 \in R^{m \times n}$

$$\dot{\mathbf{x}}_{des} = J_1 \dot{\mathbf{q}}_1$$

You can solve for $\dot{\mathbf{q}}_1$ using the Moore-Penrose pseudoinverse of the Jacobian.

Secondary task: Maintain a desired joint posture without interfering with the primary task.

First, compute the desired joint velocity for posture. Then project it into the null space of $N(J_1)$

$$\dot{\mathbf{q}}_2 = N \dot{\mathbf{q}}_{des}$$

The final joint velocity command is: $\dot{\mathbf{q}} = \dot{\mathbf{q}}_1 + \dot{\mathbf{q}}_2$

a. Implement the OSC

(4 points)

Fill out the ``os_control()`` function to implement the control for both tasks. Use the iterative null space projection method to solve the problem.

b. Visualization

(2 points)

Visualize the robot's final configuration using the provided plotting function. Test your controller with different end-effector targets and desired postures.

- Why can the robot reach the end-effector target with an arbitrary posture if Task 2 is not included?
- Do you observe that Task 1 (end-effector control) is always prioritized and accurately followed?
- Does Task 2 (posture control) only influence the motion and it does not interfere with Task 1?
- Can you identify situations where the posture task has little or no effect? Why?