Lab Course
Planning for Robotics

Syllabus - Summer 2025

Daniel Swoboda, Lecturer

Abhirup Das, Assistant

Course Overview

Planning for Robotics is a lab course (Software Projektpraktikum) at RWTH Aachen University, valued at 6CP. The course aims to teach foundations of algorithmic planning techniques, relevant to robotics applications (i.e., motion planning, path planning, and task planning). Students work in groups (2-3 students) to solve exercises related to each of the topics during the lecture period, applying techniques from software engineering. In the lecture-free period, students will work on a final project. Students are expected to propose their own final project topic. During weekly mandatory meetings (during the lecture period), students are asked to update on their progress, hand-in their solutions to exercise sheets, and are able to ask questions.

Phase	Duration	Focus
Block A: Planning for Motion	April 10 - end of May	Kinematics, Probabilistic Motion Planning, Navigation Planning
Block B: Planning for Tasks	June - mid July	STRIPS/PDDL Planning, LLMs in Planning, Integrated Task and Motion Planning
Final Project	Mid July - end of September	Student-defined integrated planning project

Weekly Curriculum Plan

Block A - Planning for Motion

Week	Topics	Deliverables
1 (April 8th - April 13th)	**Kick-off**	Kick-off Meeting
2 (April 14th - April 20th)	**Setup and Basics**	Introductory Lecture
3-4 (April 21st - April 27th)	**Kinematics and Frames**	**Exercise 1:** Setup the simulation, navigate in PyRoboSim, forward/ inverse kinematics, transformation matrices

Week	Topics	Deliverables
5-7 (April 28th-May 18th)	**Probabilistic Motion Planning**	**Exercise 2:** Implement/configure RRT in PyRoboSim to navigate environments, task on "truly understand probabilistic motion planning“
8 (May 19th - May 25th)	**Navigation Planning**	**Exercise 3:** Grid map-based path planning & obstacle avoidance, local/global planners, etc.

Block B - Planning for Tasks

Week	Topics	Deliverables
9-10 (May 26h - June 8th)	**STRIPS/PDDL Planning: States, actions, goals, domains, search**	**Exercise 4:** Domain modelling, search-based planner implementation, (simple) learned heuristics
11-13 (June 9th - June 29th)	**Generalized Planning: Sub-goals, structures, sketches, policies**	**Exercise 5:** Multi-step goal domain, hand-crafted sketch rules + hybrid planning/policy execution
14 (June 30th - July 6th)	**Planning and Execution: Symbol grounding, sim-to-plan connection, ROS2**	**Exercise 6:** Trigger symbolic plan from PyRoboSim robot, execute plans, introduction to ROS2
15 (July 7th - July 13th)	**LLMs + PDDL Excursion: Use language as an interface to planners**	**Exercise 7:** LLM as an interface to a planner/agent
16 (July 14th - July 20th)	**Intro to Task & Motion Planning: Concepts of PDDLStream, grounding challenges**	**Exercise 8:** Basic TAMP use case (e.g., picking reachable objects with kinematic constraints)

Block C - Final Project

Each group designs and implements a full pipeline (motion + task planning) in PyRoboSim (+ROS2) at their own timeframe. An initial proposal (3-4 pages) is to be submitted until week 14, the final proposal is to be submitted until week 16. The results will be presented in 15 minute talks at the end of the semester.

Example Project Ideas

- Mobile robot-based factory automation
- Kitchen assistant: symbolic recipe planner, multi-room spatial execution

- Object rearrangement with reachability constraints using simple TAMP
- Integration of LLMs and planners: grounding, world-knowledge, problem generation and integration

Tooling and Resources

Tools: PyRoboSim, ROS2, Python, PDDL/STRIPS, etc.

Resources:

- PyRoboSim Documentation
- Hector Geffner - A Concise Introduction to Models and Methods for Automated Planning
- Howie Choset - Principles of Robot Motion: Theory, Algorithms, and Implementations
- Russel and Norvig - Artificial Intelligence: A Modern Approach, 4th edition
- Scaramuzza - Introduction to Mobile Autonomous Robots
- Siddharth Srivastava's Tutorial on TAMP

Assessment

The lab course will be graded based on the performance of the group. The exercise sheets can be handed it at any time during the weekly meetings by presenting the code ant the solutions to the teaching staff. Exercises are graded based on a pass/fail scheme.

The final project is graded based on performance relative to the project proposal, the technical implementation, the proper use of software development techniques (e.g., Git, software project structure, code readability and documentation), and students final presentations. Students will present their results in a small presentation at the end of the semester.

The attendance grade is based on the percentage of attended weekly meetings. Attendance is mandatory during the lecture period. Students are

free to skip up to 3 meetings without an excuse. A properly excused missed attendance (e.g., health and family issues, problems with public transportation, conflicting important appointments) are treated as if the student was presented. In the lecture free period, the weekly meetings continue with voluntary attendance.

Grading		
Component	**Contribution to Final Grade**	**Points Formula**
Exercises (Pass/Fail)	60 %	passed_exercises/ total_exercises
Final Project	30 %	Performance based grading (0 to 100 points)
Attendance	10 %	floor((attended_meetings - 3)/total_meetings)