



Robotics: Course organization

Vladimír Petřík

vladimir.petrik@cvut.cz

21 September 2026

Outline of the course

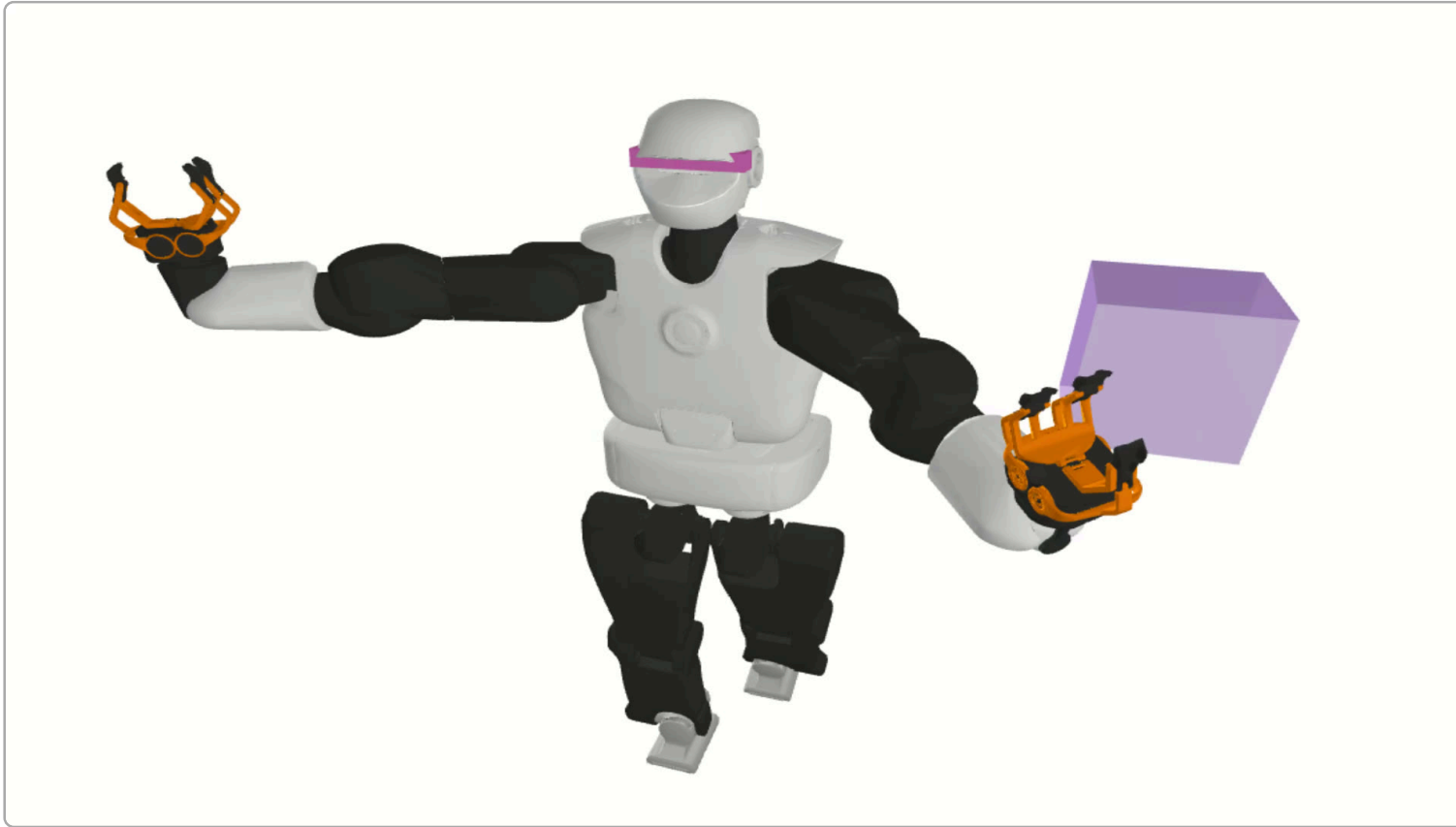
- Manipulation robotics
 - Literature: Kevin M. Lynch and Frank C. Park: **Modern Robotics: Mechanics, Planning, and Control**

Outline of the course

- Manipulation robotics
 - Literature: Kevin M. Lynch and Frank C. Park: **Modern Robotics: Mechanics, Planning, and Control**
- **Kinematics and geometry of the camera**
 - how to describe the robot pose in space
 - how to find the robot pose for a given task, e.g. grasping
 - how to find the pose of an object

Outline of the course

- Manipulation
 - Literature
 - Planning
- Kinematics
 - how
 - how
 - how

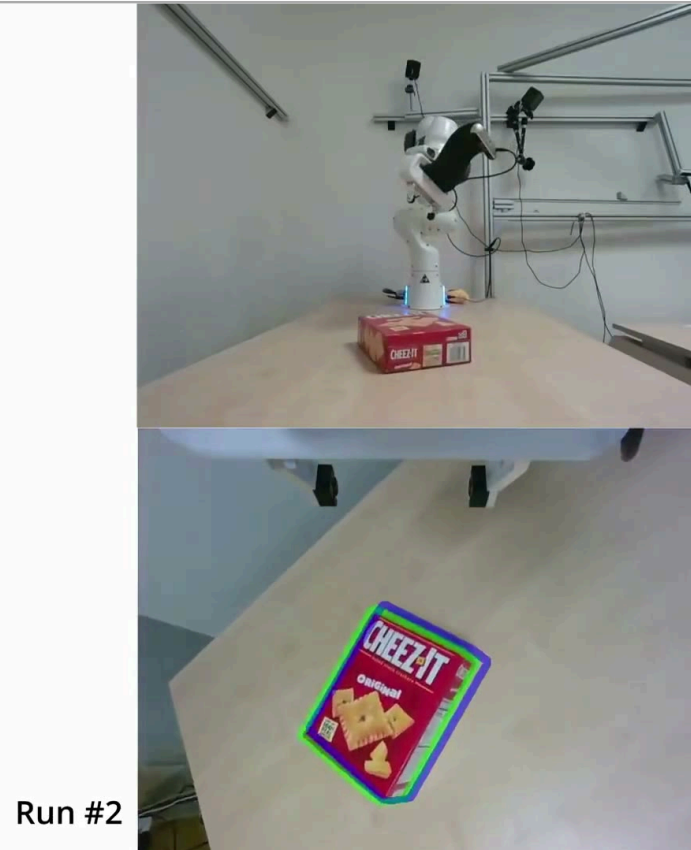
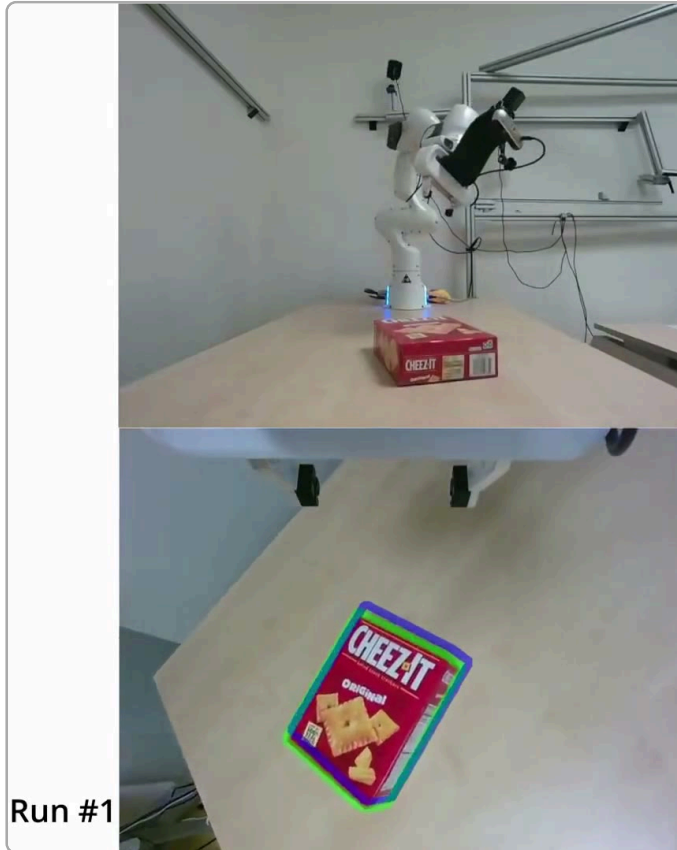


Outline of the course

- Manipulation robotics
 - Literature: Kevin M. Lynch and Frank C. Park: **Modern Robotics: Mechanics, Planning, and Control**
- **Kinematics and geometry of the camera**
 - how to describe the robot pose in space
 - how to find the robot pose for a given task, e.g. grasping
 - how to find the pose of an object
- Dynamics and control
 - how to describe the dynamic properties of a robot
 - how to control the robot dynamically

Outline of the course

- Manipulation
 - Literature
 - Planning
- Kinematics
 - how
 - how
 - how
- Dynamics
 - how
 - how

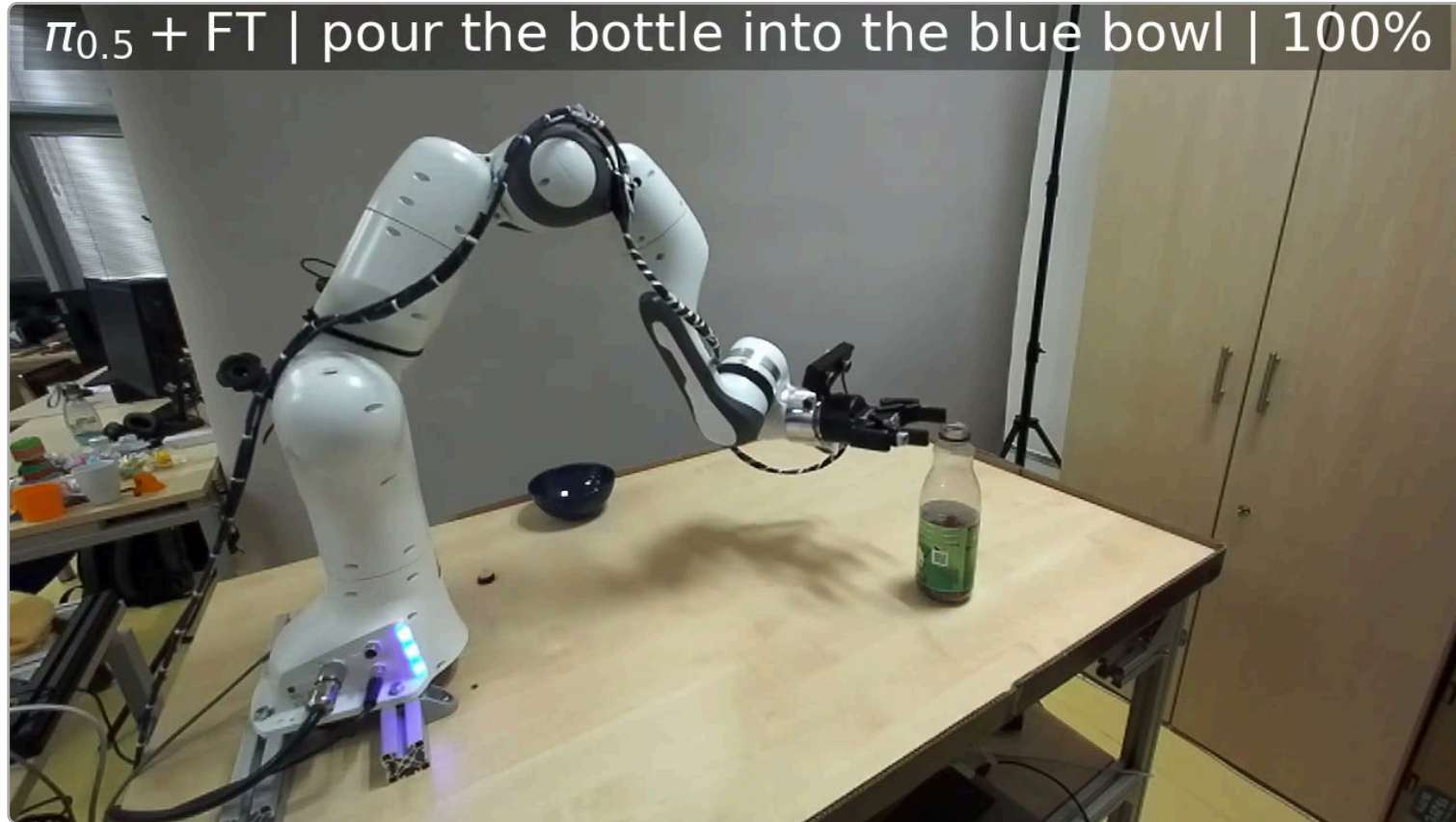


Outline of the course

- Manipulation robotics
 - Literature: Kevin M. Lynch and Frank C. Park: **Modern Robotics: Mechanics, Planning, and Control**
- **Kinematics and geometry of the camera**
 - how to describe the robot pose in space
 - how to find the robot pose for a given task, e.g. grasping
 - how to find the pose of an object
- Dynamics and control
 - how to describe the dynamic properties of a robot
 - how to control the robot dynamically
- AI in robotics
 - motion planning
 - modern AI applications (RL, VLA, ...)

Outline of the course

- Manipulation
 - Literature
 - Planning
- Kinematics
 - how
 - how
 - how
- Dynamics
 - how
 - how
- AI in robotics
 - motion
 - modeling



Lectures

Monday 12:45–14:15, KN

#	Date	Topic
1	21. 9.	Course organization, rigid body motion
–	28. 9.	public holiday
2	5. 10.	Forward kinematics of open chains
3	12. 10.	Vision in robotics, final project
4	19. 10.	Depth estimation and calibration [test 1]
5	26. 10.	Denavit-Hartenberg notation, differential kinematics
6	2. 11.	Inverse kinematics

#	Date	Topic
7	9. 11.	Motion planning
8	16. 11.	Path and trajectory generation, TAMP
9	23. 11.	Robot dynamics
10	30. 11.	Robot control
11	7. 12.	Parallel manipulators
12	14. 12.	AI in robotics I [test 2]
13	4. 1. 2027	AI in robotics II

Slides appear on the course page cw.fel.cvut.cz/b261/courses/b3b33rob1 before each lecture (also as handouts); lectures are streamed and recorded.

Laboratories

- Programming a robotics toolbox in Python
 - template repository github.com/CTURobotics/robotics_labs, documentation robotics-labs.readthedocs.io
 - work in the lab is finished as homework, evaluated automatically by unit tests in BRUTE
 - Tuesday 12:45–14:15, 14:30–16:00 and Wednesday 14:30–16:00, 16:15–17:45
 - labs follow up the lectures, **study the lecture before the lab**

Laboratories

- Programming a robotics toolbox in Python
 - template repository github.com/CTURobotics/robotics_labs, documentation robotics-labs.readthedocs.io
 - work in the lab is finished as homework, evaluated automatically by unit tests in BRUTE
 - Tuesday 12:45–14:15, 14:30–16:00 and Wednesday 14:30–16:00, 16:15–17:45
 - labs follow up the lectures, **study the lecture before the lab**
- Final project on a real industrial robot
 - **safety training in the 2nd week is mandatory** (JP:B-415); without it you cannot work with the robots
 - submission in BRUTE by 5 January 2027; up to +5 bonus points for early submission (17. 11.: +5, then –1 per week)

Laboratories plan

#	Tue	Wed	Topic	Homework	Due
1	22. 9.	23. 9.	Toolbox structure, automatic tests, transformations in SE(2)/SE(3)	hw01 opt_hw01	4. 10.
2	29. 9.	30. 9.	Safety training (JP:B-415), mandatory	–	–
3	6. 10.	7. 10.	Forward kinematics of a planar manipulator, visualization	hw02	18. 10.
4	13. 10.	14. 10.	Camera–robot transformation, homography	hw03	25. 10.
5	20. 10.	21. 10.	individual work on the project		
6	27. 10.	28. 10.	individual work on the project		
7	3. 11.	4. 11.	Jacobian of a planar robot, numerically and analytically	hw04	15. 11.
8	10. 11.	11. 11.	Inverse kinematics	hw05 opt_hw05	22. 11.
9	17. 11.	18. 11.	individual work on the project		
10	24. 11.	25. 11.	Motion planning	opt_hw06	6. 12.
11–13	1. 12.	2. 12.	individual work on the project (also 8.–9. 12., 15.–16. 12.)		
14	5. 1.	6. 1.	Final project submission	project	5. 1. 2027

hwNN mandatory, opt_hwNN optional, 1 point each; due Sunday 23:59.

How (not?) to solve homework

The screenshot shows a VS Code editor window with the following components:

- Explorer:** A file tree on the left showing the project structure. The file `01_so2_example.py` is selected.
- Editor:** The main workspace shows the content of `01_so2_example.py`. The code is a Python script that defines a `RendererPlanar` class and includes comments about its purpose and usage.
- hw01 tests implementation:** A panel on the right showing the test results for the homework. It includes a task description, rules, and a list of failed tests.
- Terminal:** A terminal window at the bottom showing the command `uv run pytest tests/hw01` and its output, which lists 23 failed tests and 15 passed tests.

The test results in the terminal are as follows:

```
(robotics-labs) petrivl3@Vladimir--MacBook-Pro robotics_labs %  
(robotics-labs) petrivl3@Vladimir--MacBook-Pro robotics_labs % uv run pytest tests/hw01  
FAILED tests/hw01/mandatory/test_so3.py::TestS03::test_exp - AssertionError: <robotics_toolbox.core.so3.S03 object at 0x110ee4770> != <robotics_toolbox.core.so3.S03 object at 0x110ee46e0>  
FAILED tests/hw01/mandatory/test_so3.py::TestS03::test_inverse - AssertionError:  
FAILED tests/hw01/mandatory/test_so3.py::TestS03::test_log - AssertionError: <robotics_toolbox.core.so3.S03 object at 0x110ee5460> != <robotics_toolbox.core.so3.S03 object at 0x110ee53d0>  
FAILED tests/hw01/optional/test_so3_opt.py::TestS030pt::test_from_angle_axis - NotImplementedError: Needs to be implemented  
FAILED tests/hw01/optional/test_so3_opt.py::TestS030pt::test_from_ea - NotImplementedError: Needs to be implemented  
FAILED tests/hw01/optional/test_so3_opt.py::TestS030pt::test_from_q - NotImplementedError: From quaternion needs to be implemented  
FAILED tests/hw01/optional/test_so3_opt.py::TestS030pt::test_rx - NotImplementedError: RX needs to be implemented.  
FAILED tests/hw01/optional/test_so3_opt.py::TestS030pt::test_ry - NotImplementedError: RY needs to be implemented.  
FAILED tests/hw01/optional/test_so3_opt.py::TestS030pt::test_rz - NotImplementedError: RZ needs to be implemented.  
FAILED tests/hw01/optional/test_so3_opt.py::TestS030pt::test_to_angle_axis - NotImplementedError: Needs to be implemented  
FAILED tests/hw01/optional/test_so3_opt.py::TestS030pt::test_to_q - NotImplementedError: To quaternion needs to be implemented  
===== 23 failed, 15 passed in 0.90s =====  
(robotics-labs) petrivl3@Vladimir--MacBook-Pro robotics_labs %
```

How (not?) to solve homework

The screenshot shows a VS Code editor window with the following components:

- Explorer:** A sidebar on the left showing a file tree for a project named 'robotics_labs'. The tree includes files like '.github', '.pytest_cache', '.venv', 'docs', 'exercises', 'lab01', 'lab02', 'lab03', 'lab04', 'lab05', 'lab06', 'src', 'tests', and 'LICENSE'. The 'lab01' directory is expanded, showing files like '01_so2_example.py' and '02_mobile_robot.py'.
- 01_so2_example.py:** A Python file in the center editor. It contains a docstring with copyright information and a comment about a transformation. The code imports 'numpy' and 'RendererPlanar' from 'robotics_toolbox.core.so2'. It defines a 'Renderer' class and a 'render' method. The file is 17 lines long.
- hw01 tests implementation:** A panel on the right showing the results of running tests. It includes a task description: 'Your task is to make all tests for hw01 pass by implementing the required functionality. Rules: 1. Do NOT modify, delete, skip, or comment out any test files or test cases. The tests are the specification and must remain exactly as they are.' Below this, it shows a 'Bash' command: 'Run hw01 tests for baseline'. The output shows several test failures, including 'test_so2.py::TestS02::test_reference_rotations', 'test_so2.py::TestS02::test_so2_initialization', and 'test_so3.py::TestS03::test_composition'. A 'Thinking...' status is shown with '50 tokens' and a 'Marinating...' status.
- Terminal:** A terminal window at the bottom showing the command 'uv run pytest tests/hw01' and its output. The output shows a list of failed tests and their error messages, including 'AssertionError: <robotics_toolbox.core.so3.S03 object at 0x110ee4770> != <robotics_toolbox.core.so3.S03 object at 0x110ee46e0>', 'NotImplementedError: Needs to be implemented', and 'NotImplementedError: RX needs to be implemented'. The summary at the bottom indicates '23 failed, 15 passed in 0.90s'.

Evaluation

- Homework (robotics toolbox): 5 mandatory + 3 optional, 1 point each
- Final project: 20 points (+ up to 5 bonus for early submission)
- Two tests during the lectures (19. 10. and 14. 12.): 2 × 16 points, pen only
- Exam: 40 points

Part	Points
Homework	8
Final project	20
Tests (2 × 16)	32
Exam	40
Total	100

Requirements

- Homework
 - all mandatory homework must pass the tests in BRUTE
 - deadline Sunday 23:59, about 1.5 weeks after the lab
 - late mandatory homework: -0.1 point per day; optional homework cannot be late

Requirements

- Homework
 - all mandatory homework must pass the tests in BRUTE
 - deadline Sunday 23:59, about 1.5 weeks after the lab
 - late mandatory homework: -0.1 point per day; optional homework cannot be late
- Final project: at least 1 point
- Tests are optional, no minimum
- Exam: at least 10 of 40 points; registration only after the credit

Teachers

- Lectures: Vladimír Petřík, vladimir.petrik@cvut.cz

Teachers

- Lectures: Vladimír Petřík, vladimir.petrik@cvut.cz
- Laboratories
 - Robotics toolbox and homework
 - Martin Cífka
 - Petr Vanc
 - David Kovář

Teachers

- Lectures: Vladimír Petřík, vladimir.petrik@cvut.cz
- Laboratories
 - Robotics toolbox and homework
 - Martin Cífk
 - Petr Vanc
 - David Kovář
 - Final project on real robots
 - Pavel Krsek
 - Vladimír Smutný