



Robotika: Organizace předmětu

Vladimír Petřík

vladimir.petrik@cvut.cz

21. 9. 2026

Přehled předmětu

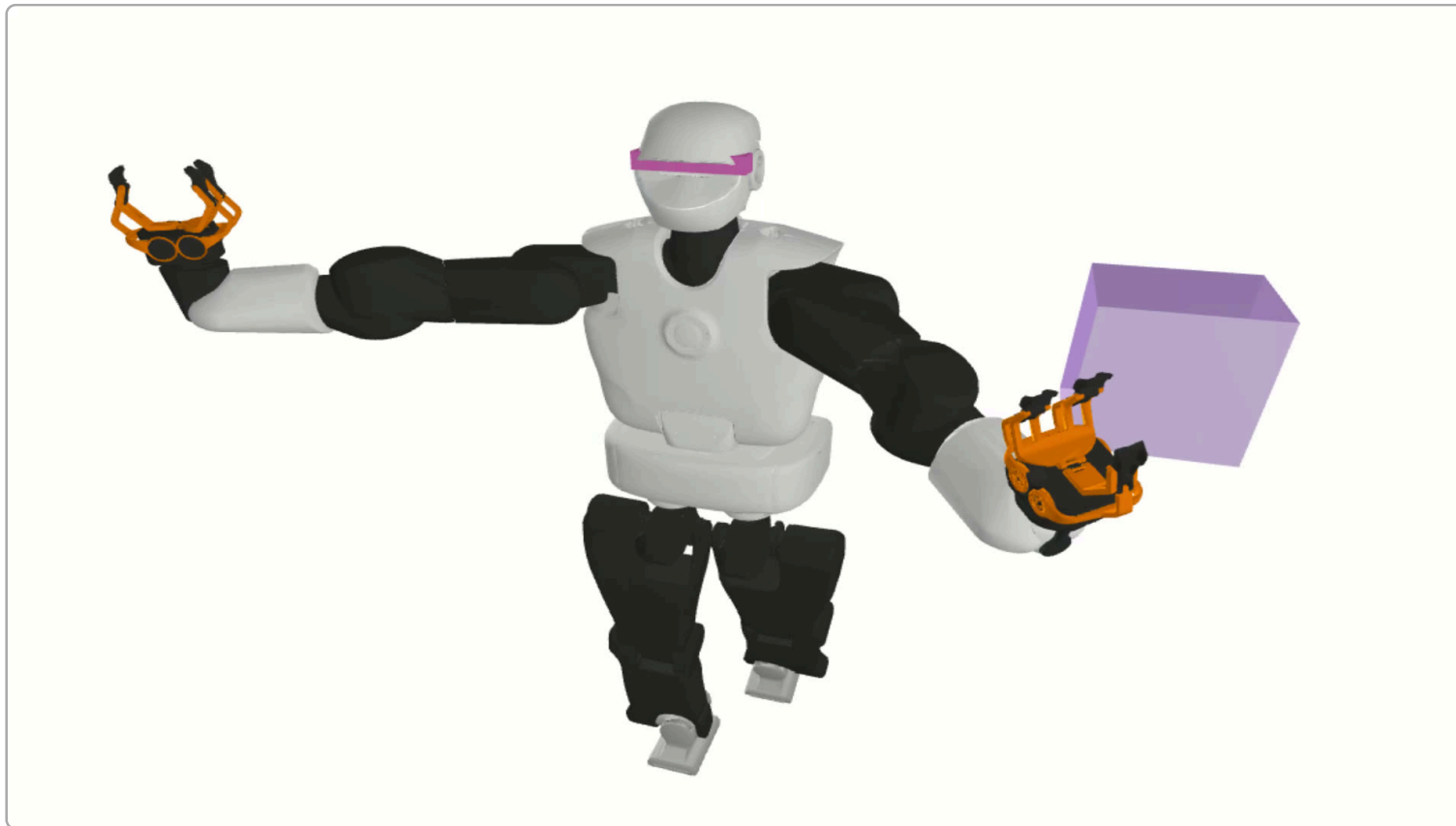
- Manipulační robotika
 - Literatura: Kevin M. Lynch and Frank C. Park: **Modern Robotics: Mechanics, Planning, and Control**

Přehled předmětu

- Manipulační robotika
 - Literatura: Kevin M. Lynch and Frank C. Park: **Modern Robotics: Mechanics, Planning, and Control**
- **Kinematika a geometrie kamery**
 - jak popsat polohu robota v prostoru
 - jak najít polohu robota pro daný úkol, např. uchopování objektu
 - jak najít polohu a orientaci objektu

Přehled předmětu

- Manipulace
 - Liter
 - Plan
- Kinem
 - jak p
 - jak r
 - jak r

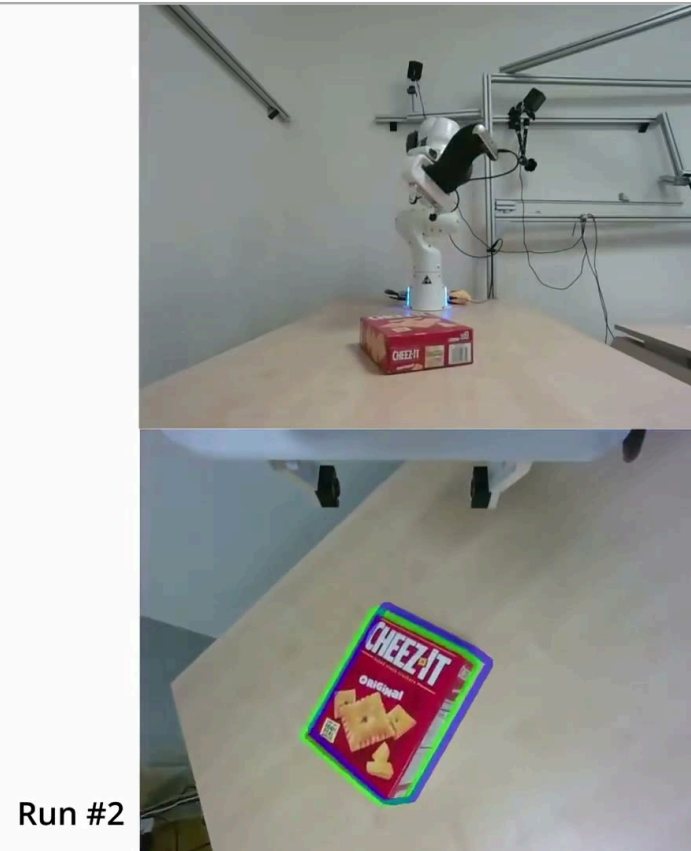
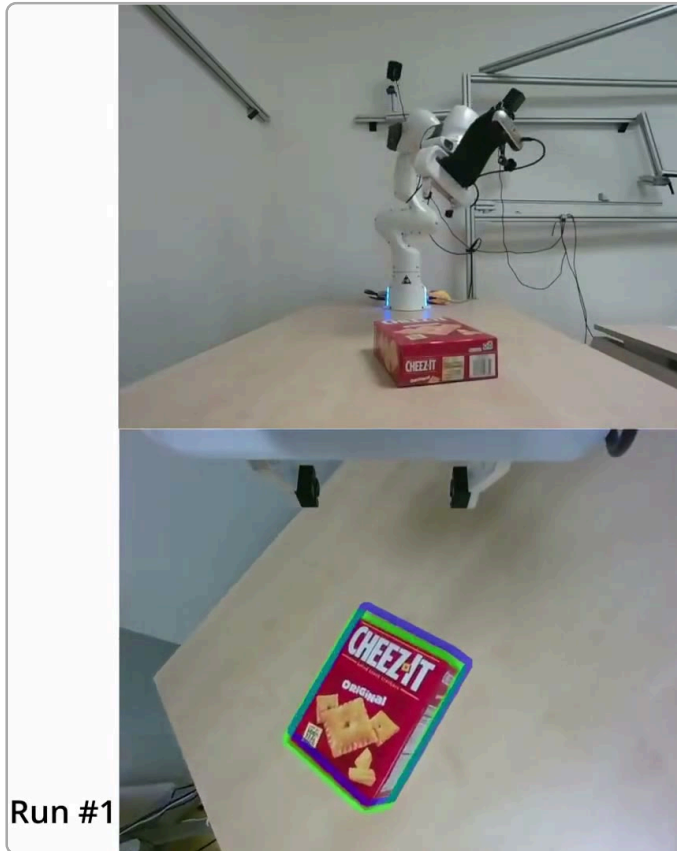


Přehled předmětu

- Manipulační robotika
 - Literatura: Kevin M. Lynch and Frank C. Park: **Modern Robotics: Mechanics, Planning, and Control**
- **Kinematika a geometrie kamery**
 - jak popsat polohu robota v prostoru
 - jak najít polohu robota pro daný úkol, např. uchopování objektu
 - jak najít polohu a orientaci objektu
- Dynamika a řízení
 - jak popsat dynamické vlastnosti robota
 - jak dynamicky řídit robota

Přehled předmětu

- Manipulace
 - Litera
 - Plan
- Kinematika
 - jak p
 - jak r
 - jak r
- Dynamika
 - jak p
 - jak c



Přehled předmětu

- Manipulační robotika
 - Literatura: Kevin M. Lynch and Frank C. Park: **Modern Robotics: Mechanics, Planning, and Control**
- **Kinematika a geometrie kamery**
 - jak popsat polohu robota v prostoru
 - jak najít polohu robota pro daný úkol, např. uchopování objektu
 - jak najít polohu a orientaci objektu
- Dynamika a řízení
 - jak popsat dynamické vlastnosti robota
 - jak dynamicky řídit robota
- AI v robotice
 - plánování pohybu
 - moderní AI aplikace (RL, VLA, ...)

Přehled předmětu

- Manipulace

- Liter

- Plan**

- **Kinem**

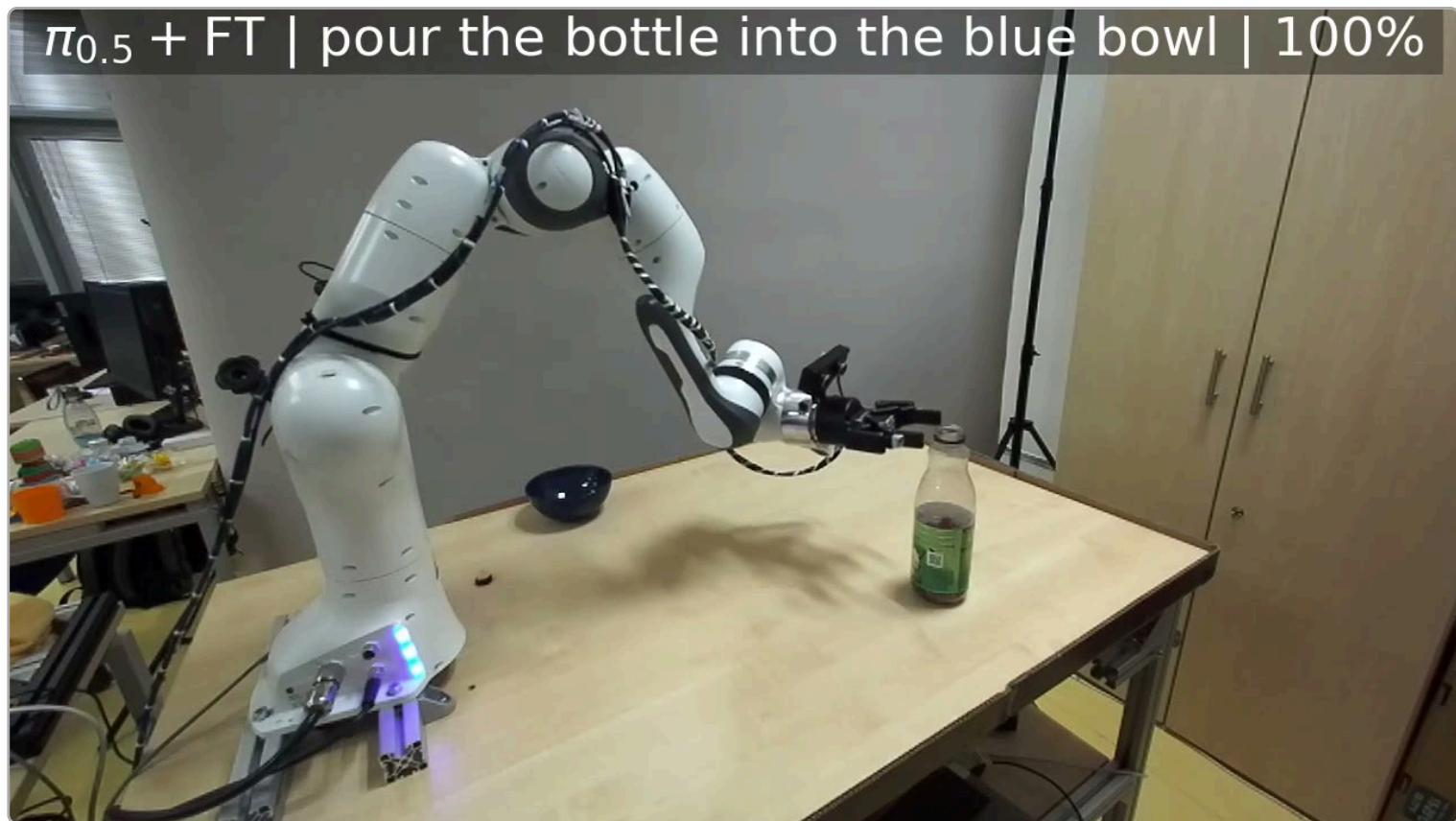
- jak p
 - jak r
 - jak r

- Dynamika

- jak p
 - jak c

- AI v roli

- plán
 - moc



Přednášky

Pondělí 12:45–14:15, KN

#	Datum	Téma
1	21. 9.	Organizace předmětu, reprezentace pohybu
–	28. 9.	státní svátek
2	5. 10.	Přímá kinematika otevřených řetězců
3	12. 10.	Vidění v robotice, semestrální projekt
4	19. 10.	Odhad hloubky a kalibrace [1. test]
5	26. 10.	Denavit-Hartenbergova notace, diferenciální kinematika
6	2. 11.	Inverzní kinematika

#	Datum	Téma
7	9. 11.	Plánování pohybu
8	16. 11.	Generování cest a trajektorií, TAMP
9	23. 11.	Dynamika robotů
10	30. 11.	Metody řízení robotů
11	7. 12.	Paralelní manipulátory
12	14. 12.	Úvod do AI v robotice I [2. test]
13	4. 1. 2027	Úvod do AI v robotice II

Prezentace jsou na stránce předmětu cw.fel.cvut.cz/b261/courses/b3b33rob1 před každou přednáškou (i jako handout); přednášky jsou streamovány a nahrávány.

Cvičení

- Programování robotické sady nástrojů (toolbox) v jazyce Python
 - šablona repozitáře github.com/CTURobotics/robotics_labs, dokumentace robotics-labs.readthedocs.io
 - práce na cvičení pokračuje domácím úkolem, hodnocení automaticky unit testy v BRUTE
 - úterý 12:45–14:15, 14:30–16:00 a středa 14:30–16:00, 16:15–17:45
 - cvičení navazují na přednášky, **nastudujte přednášku před cvičením**

Cvičení

- Programování robotické sady nástrojů (toolbox) v jazyce Python
 - šablona repozitáře github.com/CTURobotics/robotics_labs, dokumentace robotics-labs.readthedocs.io
 - práce na cvičení pokračuje domácím úkolem, hodnocení automaticky unit testy v BRUTE
 - úterý 12:45–14:15, 14:30–16:00 a středa 14:30–16:00, 16:15–17:45
 - cvičení navazují na přednášky, **nastudujte přednášku před cvičením**
- Semestrální práce na reálném průmyslovém robotu
 - **školení bezpečnosti ve 2. týdnu je povinné** (JP:B-415); bez něj nemůžete pracovat s roboty
 - odevzdání v BRUTE do 5. 1. 2027; bonus až +5 bodů za předčasné odevzdání (17. 11.: +5, pak –1 každý týden)

Plán cvičení

#	Út	St	Náplň	Domácí úkol	Odevzdat do
1	22. 9.	23. 9.	Struktura toolboxu, automatické testy, transformace v SE(2)/SE(3)	hw01 opt_hw01	4. 10.
2	29. 9.	30. 9.	Bezpečnost práce (JP:B-415), povinné	–	–
3	6. 10.	7. 10.	Přímá kinematika planárního manipulátoru, vizualizace	hw02	18. 10.
4	13. 10.	14. 10.	Transformace kamera–robot, homografie	hw03	25. 10.
5	20. 10.	21. 10.	individuální práce na projektu		
6	27. 10.	28. 10.	individuální práce na projektu		
7	3. 11.	4. 11.	Jakobián planárního robota numericky a analyticky	hw04	15. 11.
8	10. 11.	11. 11.	Inverzní kinematika	hw05 opt_hw05	22. 11.
9	17. 11.	18. 11.	individuální práce na projektu		
10	24. 11.	25. 11.	Plánování pohybu	opt_hw06	6. 12.
11–13	1. 12.	2. 12.	individuální práce na projektu (také 8.–9. 12., 15.–16. 12.)		
14	5. 1.	6. 1.	Odevzdání semestrální práce	projekt	5. 1. 2027

hwNN povinný, opt_hwNN dobrovolný, 1 bod za úkol; termín neděle 23:59.

Jak (ne)řešit domácí úkoly

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 visualizes a few frames.
- hw01 tests implementation:** A panel on the right showing the test results for the `hw01` tests. It includes a task description, rules, and a list of failed tests.
- Terminal:** A terminal window at the bottom showing the output of the command `uv run pytest tests/hw01`. The output indicates that 23 tests failed and 15 passed.

01_so2_example.py

```
1 #!/usr/bin/env python
2 #
3 # Copyright (c) CTU — All Rights Reserved
4 # Created on: 2023-07-4
5 # Author: Vladimír Petrik <vladimir.petrík@cvut.cz>
6 #
7 # This example shows what transformation does with a vector and basic properties of
8 # transformation.
9 # Note, you need to complete robotics_toolbox/core/so2.py in order for script this to
10 # work.
11 import numpy as np
12
13 from robotics_toolbox.core.so2 import S02
14 from robotics_toolbox.render import RendererPlanar
15
16 # Let's visualize a few frames
17 renderer = RendererPlanar(xlim=(-0.25,
```

hw01 tests implementation

Your task is to make all tests for hw01 pass by implementing the required functionality.

Rules:

- 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.

Bash Run hw01 tests for baseline

```
IN .venv/bin/pytest tests/hw01 -q 2>61 | tail -15
OUT FAILED tests/hw01/mandatory/test_so2.py::TestS02::test_reference_rotations - ...
    FAILED tests/hw01/mandatory/test_so2.py::TestS02::test_so2_initialization - A...
    FAILED tests/hw01/mandatory/test_so3.py::TestS03::test_composition - Assertio...
```

Thinking... · 50 tokens

Marinating...

Queue another message...

+ [60m Fable 5.1 High Auto

Problems Output Debug Console **Terminal** Ports

```
(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_so2.py::TestS02::test_reference_rotations - ...
FAILED tests/hw01/mandatory/test_so2.py::TestS02::test_so2_initialization - A...
FAILED tests/hw01/mandatory/test_so3.py::TestS03::test_composition - Assertio...
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 0x110ee4770> != <robotics_toolbox.core.so3.S03 object at 0x110ee46e0>
FAILED tests/hw01/optional/test_so3_opt.py::TestS03Opt::test_from_angle_axis - NotImplementedError: Needs to be implemented
FAILED tests/hw01/optional/test_so3_opt.py::TestS03Opt::test_from_ea - NotImplementedError: Needs to be implemented
FAILED tests/hw01/optional/test_so3_opt.py::TestS03Opt::test_from_q - NotImplementedError: From quaternion needs to be implemented
FAILED tests/hw01/optional/test_so3_opt.py::TestS03Opt::test_rx - NotImplementedError: RX needs to be implemented.
FAILED tests/hw01/optional/test_so3_opt.py::TestS03Opt::test_ry - NotImplementedError: RY needs to be implemented.
FAILED tests/hw01/optional/test_so3_opt.py::TestS03Opt::test_rz - NotImplementedError: RZ needs to be implemented.
FAILED tests/hw01/optional/test_so3_opt.py::TestS03Opt::test_to_angle_axis - NotImplementedError: Needs to be implemented
FAILED tests/hw01/optional/test_so3_opt.py::TestS03Opt::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 %
```

Jak (ne)řešit domácí úkoly

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 visualizes a few frames.
- hw01 tests implementation:** A panel on the right showing the test results for the `hw01` tests. It includes a task description, rules, and a list of failed tests.
- Terminal:** A terminal window at the bottom showing the output of the command `uv run pytest tests/hw01`. The output indicates that 23 tests failed and 15 passed.

01_so2_example.py

```
1 #!/usr/bin/env python
2 #
3 # Copyright (c) CTU — All Rights Reserved
4 # Created on: 2023-07-4
5 # Author: Vladimír Petrik <vladimir.petrík@cvut.cz>
6 #
7 # This example shows what transformation does with a vector and basic properties of
8 # transformation.
9 # Note, you need to complete robotics_toolbox/core/so2.py in order for script this to
10 # work.
11 import numpy as np
12
13 from robotics_toolbox.core.so2 import S02
14 from robotics_toolbox.render import RendererPlanar
15
16 # Let's visualize a few frames
17 renderer = RendererPlanar(xlim=(-0.25,
```

hw01 tests implementation

Your task is to make all tests for hw01 pass by implementing the required functionality.

Rules:

- 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.

Bash Run hw01 tests for baseline

```
IN .venv/bin/pytest tests/hw01 -q 2>61 | tail -15
OUT FAILED tests/hw01/mandatory/test_so2.py::TestS02::test_reference_rotations - ...
    FAILED tests/hw01/mandatory/test_so2.py::TestS02::test_so2_initialization - A...
    FAILED tests/hw01/mandatory/test_so3.py::TestS03::test_composition - Assertio...
```

Thinking... · 50 tokens

Marinating...

Queue another message...

+ [60m Fable 5.1 High Auto

Problems Output Debug Console **Terminal** Ports

```
(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_so2.py::TestS02::test_reference_rotations - ...
FAILED tests/hw01/mandatory/test_so2.py::TestS02::test_so2_initialization - A...
FAILED tests/hw01/mandatory/test_so3.py::TestS03::test_composition - Assertio...
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 0x110ee4770> != <robotics_toolbox.core.so3.S03 object at 0x110ee46e0>
FAILED tests/hw01/optional/test_so3_opt.py::TestS03Opt::test_from_angle_axis - NotImplementedError: Needs to be implemented
FAILED tests/hw01/optional/test_so3_opt.py::TestS03Opt::test_from_ea - NotImplementedError: Needs to be implemented
FAILED tests/hw01/optional/test_so3_opt.py::TestS03Opt::test_from_q - NotImplementedError: From quaternion needs to be implemented
FAILED tests/hw01/optional/test_so3_opt.py::TestS03Opt::test_rx - NotImplementedError: RX needs to be implemented.
FAILED tests/hw01/optional/test_so3_opt.py::TestS03Opt::test_ry - NotImplementedError: RY needs to be implemented.
FAILED tests/hw01/optional/test_so3_opt.py::TestS03Opt::test_rz - NotImplementedError: RZ needs to be implemented.
FAILED tests/hw01/optional/test_so3_opt.py::TestS03Opt::test_to_angle_axis - NotImplementedError: Needs to be implemented
FAILED tests/hw01/optional/test_so3_opt.py::TestS03Opt::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 %
```

Hodnocení

- Domácí úkoly (toolbox): 5 povinných + 3 dobrovolné, 1 bod za úkol
- Semestrální práce: 20 bodů (+ až 5 bonus za předčasné odevzdání)
- Dva testy na přednášce (19. 10. a 14. 12.): 2 × 16 bodů, pouze pero
- Zkouška: 40 bodů

Část	Body
Domácí úkoly	8
Semestrální práce	20
Testy (2 × 16)	32
Zkouška	40
Celkem	100

Podmínky zápočtu a zkoušky

- Domácí úkoly
 - všechny povinné úkoly musí projít testy v BRUTE
 - termín neděle 23:59, cca 1,5 týdne po cvičení
 - povinný úkol po termínu: $-0,1$ bodu za den; dobrovolný po termínu odevzdat nelze

Podmínky zápočtu a zkoušky

- Domácí úkoly
 - všechny povinné úkoly musí projít testy v BRUTE
 - termín neděle 23:59, cca 1,5 týdne po cvičení
 - povinný úkol po termínu: $-0,1$ bodu za den; dobrovolný po termínu odevzdat nelze
- Semestrální práce: alespoň 1 bod
- Testy jsou dobrovolné, bez minima
- Zkouška: alespoň 10 ze 40 bodů; přihlášení až po získání zápočtu

Vyučující

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

Vyučující

- Přednášky: Vladimír Petřík, vladimir.petrik@cvut.cz
- Cvičení
 - Robotický toolbox a domácí úkoly
 - Martin Cífk
 - Petr Vanc
 - David Kovář

Vyučující

- Přednášky: Vladimír Petřík, vladimir.petrik@cvut.cz
- Cvičení
 - Robotický toolbox a domácí úkoly
 - Martin Cířka
 - Petr Vanc
 - David Kovář
 - Semestrální práce na reálných robotech
 - Pavel Krsek
 - Vladimír Smutný