Elite Summer School

Digital Twins and simulation for robotic systems

Exam task description

Christian Schlette

Tasks

a) Implement your version of **one** of two common **sampling-based motion planning** methods

b) Implement a **visualization of the progress** of the planning method in cartesian space

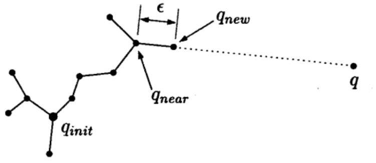


Figure 3: The EXTEND operation.

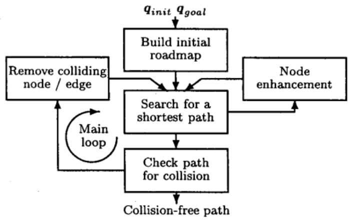
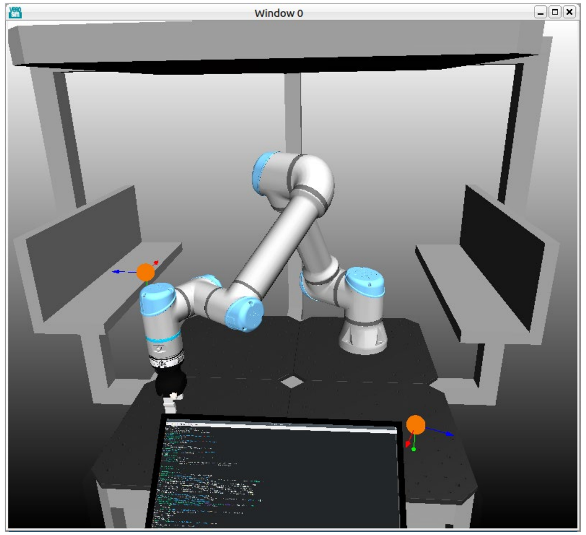
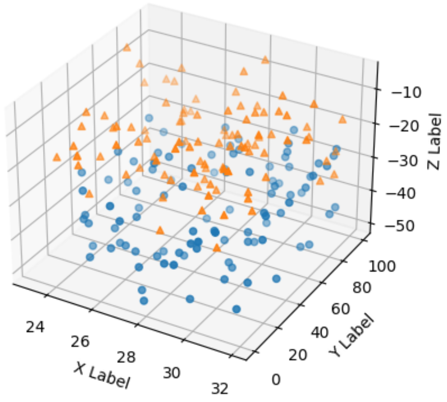


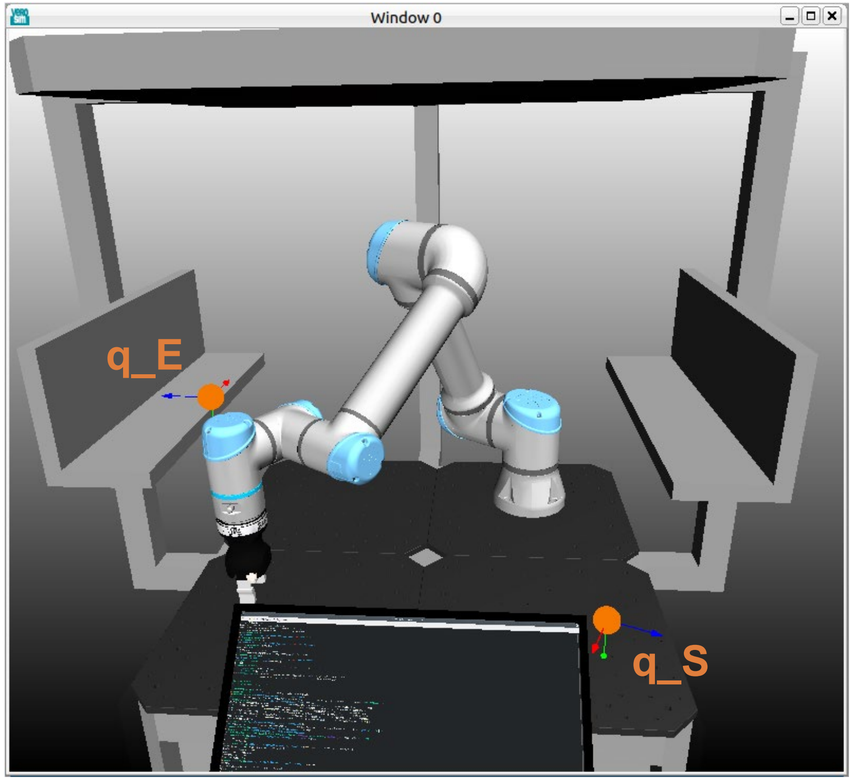
Figure 1: High-level description of Lazy PRM.





Task a) Sampling-based motion planning

- Get **two collision-free poses** in joint space (q_S and q_E) such that the robot PTP motion **has collision** with the work cell
- Plan a **collision-free path** between q_S and q_E using the **python interface** in VEROSIM
- Return the list of **intermediate joint poses** to VEROSIM for execution in **simulation**



Task a) Sampling-based motion planning

You can choose to implement **one method between**:

- **Lazy PRM**
  - https://ieeexplore.ieee.org/document/844107
- **RRT-Connect**
  - https://ieeexplore.ieee.org/document/844730

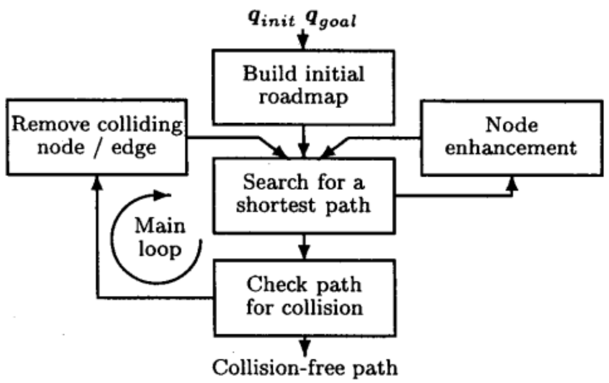


Figure 1: High-level description of Lazy PRM.

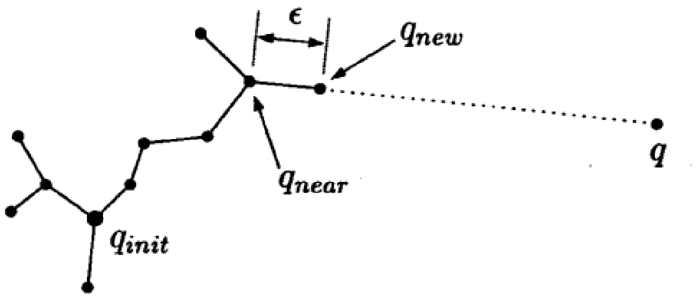


Figure 3: The EXTEND operation.

(Pictures taken from linked papers)
You can download the papers from the SDU library website:
https://syddansk.summon.serialssolutions.com/en-UK/#!/

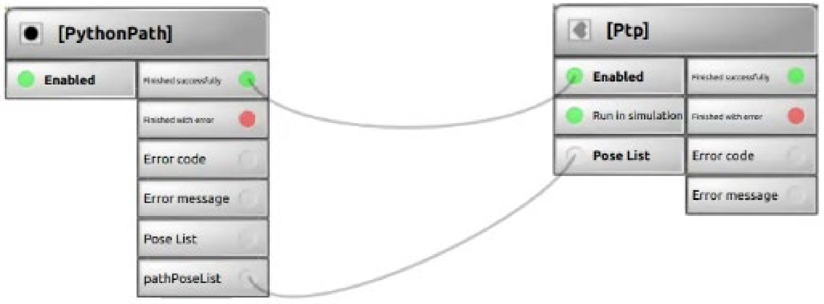
Task a) Sampling-based motion planning

Nextcloud

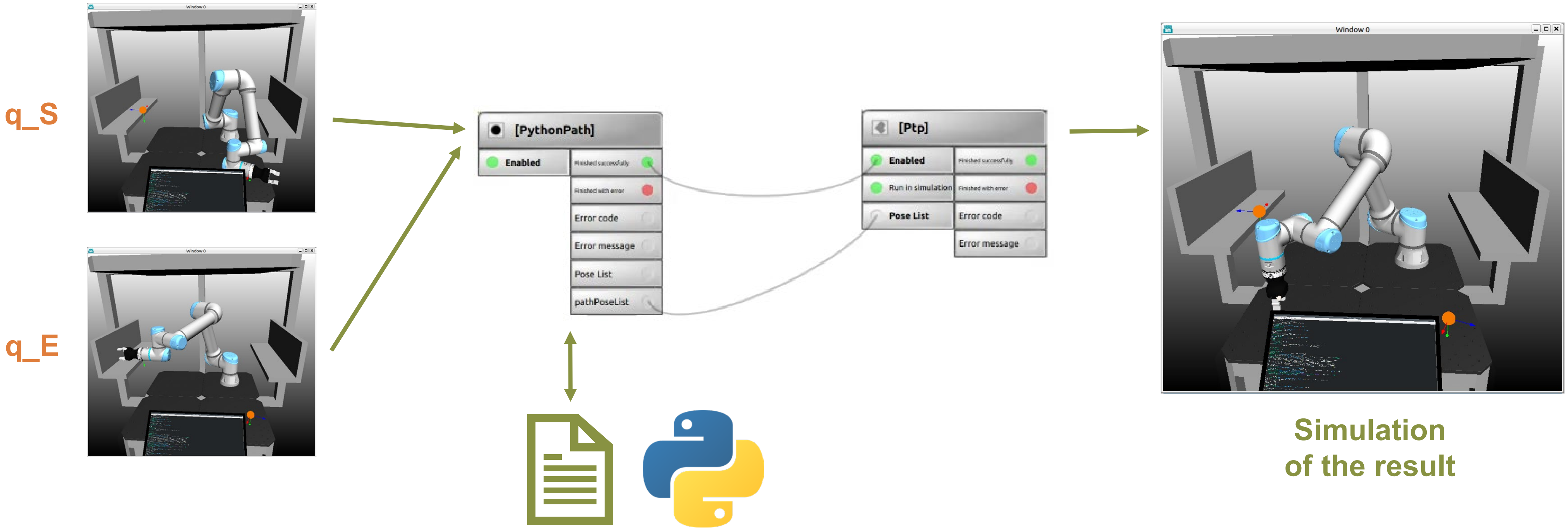
Start with the **template exercise** (MyExercise.MOD) and instruction video you can find in the shared NextCloud folder called "SummerSchool 2023 VM Ubuntu 20/ Exam task":

Use the python utility functions:

- com.hasCollision(poseCandidate)
  - Returns a Boolean for collision
- com.clearance(poseCandidate)
  - Returns Float for clearance (0 if in collision)



Task a) Sampling-based motion planning



q_S

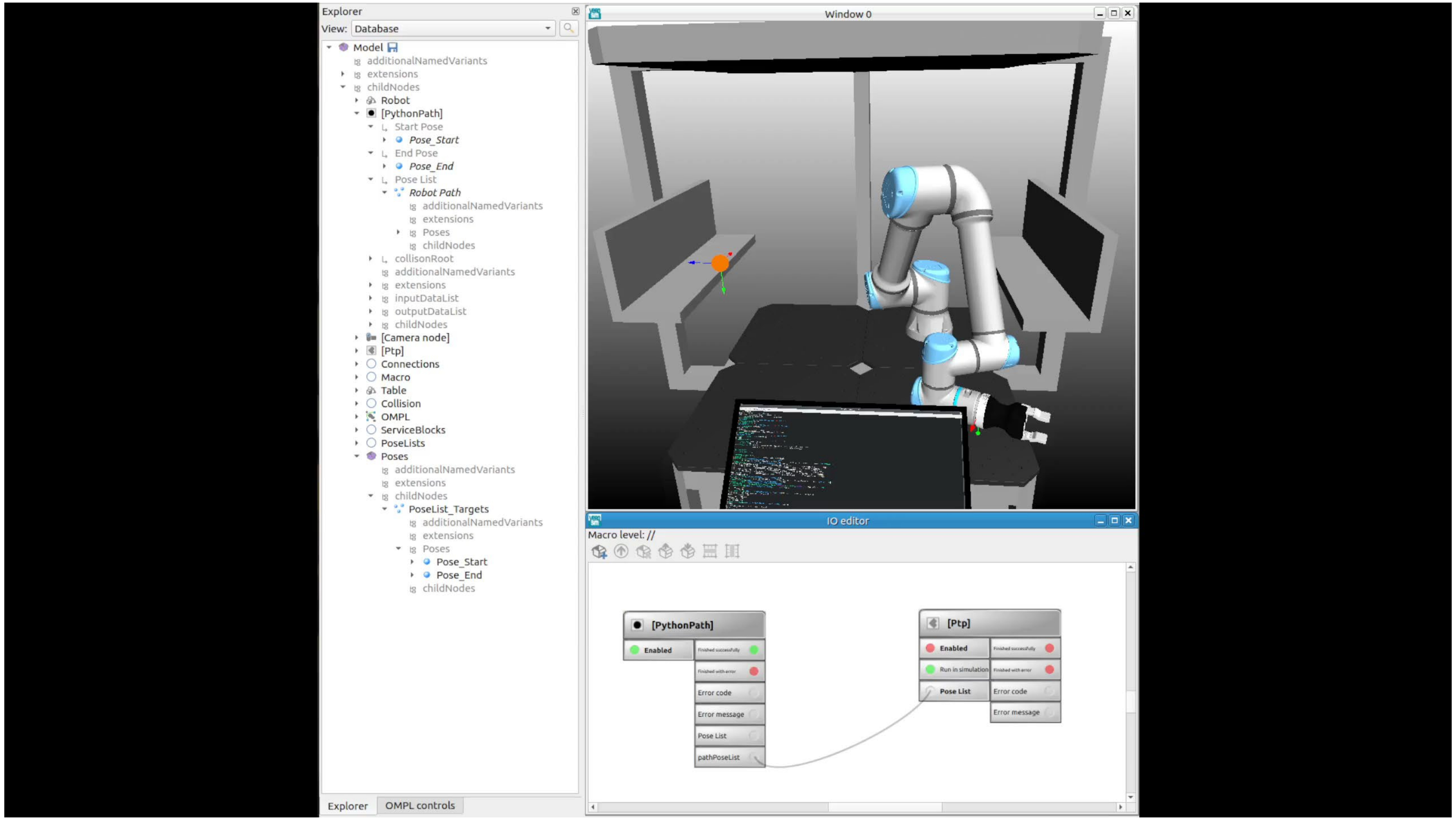
q_E

Simulation
of the result

Task a) Sampling-based motion planning

Example of result for RRT-Connect:

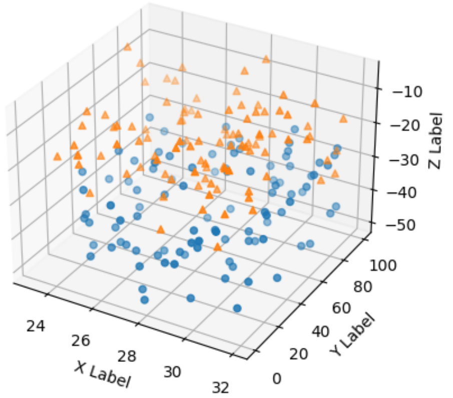
- K = 100
- epsilon = 0.01
- Python script: ~140 lines of code



Task b) Progress visualization

Visualize the progress of the selected sampling-based planner:

- Implement a function to calculate the **forward kinematics** for the UR5e robot (obtain cartesian poses for all the tested joint poses)
- Visualize the start and end cartesian poses in a **2D or 3D diagram** (use Python libraries for plotting the data – ex. Matplotlib https://matplotlib.org/)
- Visualize all the cartesian poses for **all the joint candidates** which were tested in VEROSIM using different colors, depending on the presence of collisions or the available clearance



Task b) Progress visualization

Visualize the progress of the selected sampling-based planner:

- Implement a function to calculate the **forward kinematics** for the UR5e robot (obtain cartesian poses for all the tested joint poses)
- Visualize the start and end cartesian poses in a **2D or 3D diagram** (use Python libraries for plotting the data – ex. Matplotlib https://matplo
- Visualize all the cartesian poses for **all the joint candidates** whic tested in VEROSIM using different colors, depending on the pres collisions or the available clearance

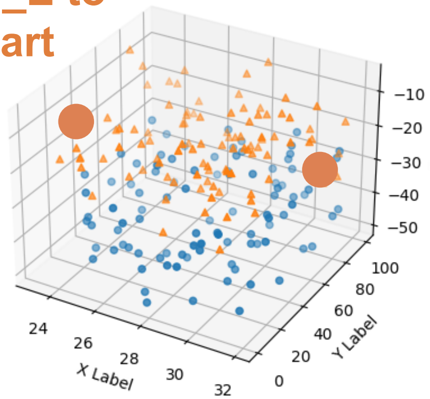
Have a look at: https://www.universal-robots.com/articles/ur/application-installation/dh-parameters-for-calculations-of-kinematics-and-dynamics/

Task b) Progress visualization

Visualize the progress of the selected sampling-based planner:

- Implement a function to calculate the **forward kinematics** for the UR5e robot (obtain cartesian poses for all the tested joint poses)
- Visualize the start and end cartesian poses in a **2D or 3D diagram** (use Python libraries for plotting the data – ex. Matplotlib https://matplotlib.org/)
- Visualize all the cartesian poses for **all the joint candidates** which were tested in VEROSIM using different colors, depending on the presence of collisions or the available clearance

Just show the position part (not orientation)



q_E to cart

q_S to cart

Task b) Progress visualization

q_i not collision and great clearance (to cart)

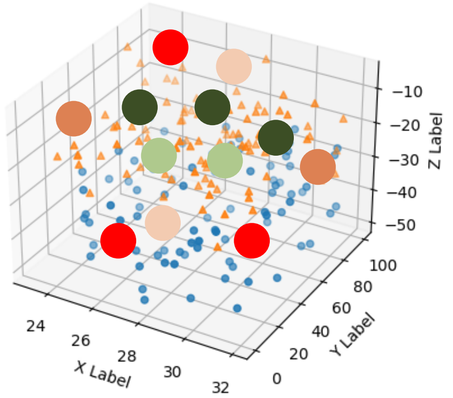
q_i not collision and medium clearance (to cart)

q_i not collision and low clearance (to cart)

q_i in collision (to cart)

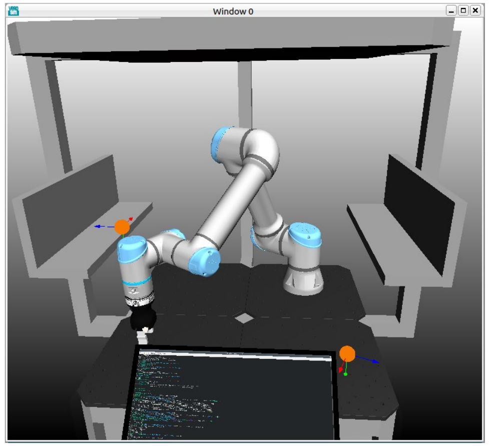
Visualize the progress of the selected sampling-based planner:

- Implement a function to calculate the **forward kinematics** for the UR5e robot (obtain cartesian poses for all the tested joint poses)
- Visualize the start and end cartesian poses in a **2D or 3D diagram** (use Python libraries for plotting the data – ex. Matplotlib https://matplotlib.org/)
- Visualize all the cartesian poses for **all the joint candidates** which were tested in VEROSIM using different colors, depending on the presence of collisions or the available clearance



Solution presentation for the exam

- **Group presentation** of solution with explanation and demonstration
  - Use a **live demo or recording** of previous runs
- Share the **presentation** in a digital format
- Share the **VEROSIM project** (.MOD)
- Share the **python files** with your code



Christian Schlette
Professor

SDU ROBOTICS
http://robotics.sdu.dk
The Mærsk Mc-Kinney Møller Institute (MMMI)
University of Southern Denmark (SDU)

chsch@mmmi.sdu.dk
T +45 6550 7916
M +45 9350 7377