**COMP0127 Robotic Systems Engineering**
**Coursework 1: Linear Algebra and Forward Kinematics**

Dr. Agostino Stilli
Department of Computer Science
University College London

October 17, 2022

To get full credit for an answer, you are *required* to provide a .pdf report, and a fully working coding solution by filling in the provided code templates. These templates provide additional information on how to implement each script. **Do not remove anything from the templates and try to only fill-in the code in the specified fields.** For the coding questions, you also are expected to include a simple breakdown of your algorithms in the report. When ready, *upload* your 'cw1' package on Moodle along with your submitted coursework report, in .zip or .rar extension. The necessary ROS packages are available on the course's *GitHub repository*.

# Linear Algebra

1. a. Given an arbitrary 3D rotation matrix,

$$\mathbf{R} = \begin{bmatrix} r_1 & r_2 & r_3 \\ r_4 & r_5 & r_6 \\ r_7 & r_8 & r_9 \end{bmatrix}$$

Prove that $||r_i|| \leq 1$ where $i = 1, 2, ..., 9$. [report - 3 pts]

b. For any rotation matrix $\mathbf{R}$, prove that $\mathbf{R}_{k,\theta} = \mathbf{R}_{-k,-\theta}$, where $k$ is the unit vector defined axis of rotation and $\theta$ is the angle of rotation. [report - 3 pts]

c. Given two arbitrary Cartesian coordinate frames $a$ and $b$, what does each row in a rotation matrix ${}^{a}\mathbf{R}_{b}$ represent? [report - 3 pts]

d. Identify the relationship between axis/angle of rotation and the eigenvector/eigenvalue of a rotation matrix. [report - 3 pts]

[12 pts]

2. a. Provide a matricial example, i.e. a succession of $3$ matrices along the $3$ different axes, of gimbal lock for the Y-Z-Y (proper Euler, extrinsic) and x-y-z (Tait-Bryan, intrinsic) rotations. Why do we need to avoid gimbal lock when controlling robotic arms? How is this achieved? [report - 4 pts]

b. Show how to pass from Quaternion representation to rotation matrix representation. (You will need to provide all steps, not just the formula). [report - 4 pts]

c. What rotation representation would you suggest to use in the following cases:

- Nano-robot with very limited memory storage
- Nano-robot with very limited computational power
- Iphone navigation system
- Robotic arm wth 6 DOF [report - 2 pts]

[10 pts]

3. a. Prove that a rotation quaternion $\mathbf{q}$ and $-\mathbf{q}$ are equivalent. [report - 3 pts]

   b. When do two arbitrary rotation matrices $\mathbf{R}_a$ and $\mathbf{R}_b$ become commutative? [report - 5 pts]

[8 pts]

4. Complete the following tasks by filling in the python code templates in the packages ”cw1/cw1q4_srv” and ”cw1/cw1q4” to create services that perform representation transformations.

   a. Fill in the template in package ”cw1/cw1q4_srv” with the appropriate request and response message types for each service. [code - 5 pts]

   - The quaternion message is ”geometry_msgs/Quaternion”, namely ”q”.
   - The Rodrigues representation message is three ”std_msgs/Float64” values, namely ”x” ,”y” and ”z”.
   - The Z-Y-X Euler angle representation message is three ”std_msgs/Float64” values, namely ”z” ,”y” and ”x”.

   b. Fill in the template in package ”cw1/cw1q4” to create a service that converts a quaternion representation to an Euler angle Z-Y-X representation (Tait-Bryan, extrinsic). Your request should contain the quaternion you need to convert, whereas your response should store the requested Euler angles. [report - 3 pts, code - 7 pts]

   c. Fill in the template in package ”cw1/cw1q4” to create a service that converts a quaternion representation to a Rodrigues representation. Your request should contain the quaternion you need to convert, whereas your response should store the requested Rodriguez representation. [report - 3 pts, code - 7 pts]

[25 pts]

# Forward Kinematics

5. Apply forward kinematics on the KUKA YouBot manipulator.

   a. Identify the standard Denavit-Hartenberg parameters for the simplified dimensions of the Youbot shown in Figure 1. Your report should include a picture with the frames on the robot joints, as well as a brief explanation of how the parameters were derived. [report - 10 pts]

   b. Complete this task by filling in the 'cw1q5b_node.py' code template, inside the package "cw1/cw1q5". Write a ROS script to compute the forward kinematics using the standard Denavit-Hartenberg convention. To complete this assignment, you must do the following:
      - Fill the "youbot_dh_parameters" dictionary with the youbot DH parameters you found in question 5a.
      - Implement the 'standard_dh()' function that defines the generic homogeneous transformation based on the four standard DH parameters.
      - Implement the 'forward_kinematics()' function that solves the forward kinematics by multiplying frame transformations.
      - Implement the 'fkine_wrapper()' function that integrates your robotics code with ROS and listens to the topic where joint states are published.
      - Initialize the subscriber to the topic that publishes joint states and its callback function fkine_wrapper() [report - 5 pts, code - 15 pts]

   c. Identify the standard D-H parameters following the complete Youbot dimensions found in the 'robot_description/youbot_description/robots/youbot_arm_only.urdf.xacro' URDF file. Based on the URDF description of each joint, you should be able to come up with a new set of DH parameters, as well as the joint offsets that the xacro file incorporates. Your report should include a brief explanation of how the parameters were derived. [report - 10 pts]

   d. Complete this task by filling in the 'cw1q5d_node.py' code template, inside the package "cw1/cw1q5". Write a ROS script to compute the forward kinematics based on the URDF description. To complete this assignment, you must do the following:
      - Fill the "youbot_dh_parameters" dictionary with the youbot DH parameters you found in question 5c.
      - Fill the "youbot_joint_offsets" dictionary to account for the joint offsets between the "youbot_dh_parameters" you found and the xarco representation.
      - Implement the 'fkine_wrapper()' function and initialize the subscriber [code - 5 pts]

[45 pts]

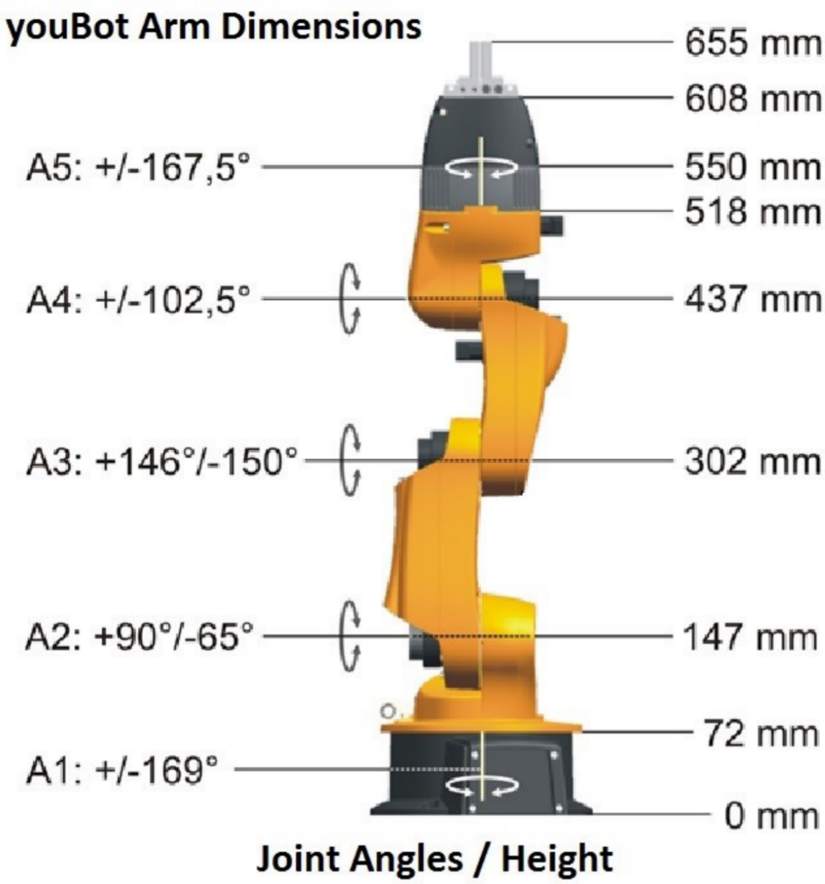


Figure 1: Kuka youBot Manipulator's simplified dimensions