# Robot Dynamics Quiz 1

Prof. Marco Hutter
Teaching Assistants: Dario Bellicoso, Jan Carius

October 17, 2018

First Name(s): ______________________________

Surname: ______________________________

Student ID: ______________________________

ETHZ-Email: ______________________________`@student.ethz.ch`

Duration: 1h 15min
Permitted Aids: Everything; no communication among students during the test

## 1 Instructions

1. Download the ZIP file for quiz 1 from Piazza. Extract all contents of this file into a new folder and set MATLAB's[1] current path to this folder.
2. Run `init_workspace` in the Matlab command line
3. All problem files that you need to complete are located in the `problems` folder
4. Run `evaluate_problems` to check if your functions run. We are not testing for correctness
5. When the time is up, zip the entire folder and email it to
   `robotdynamics@leggedrobotics.com`
   from your ETH email address with the subject line
   `[RobotDynamics] ETHStudentID - StudentName`

[1]Online version of MATLAB at `https://matlab.mathworks.com/`

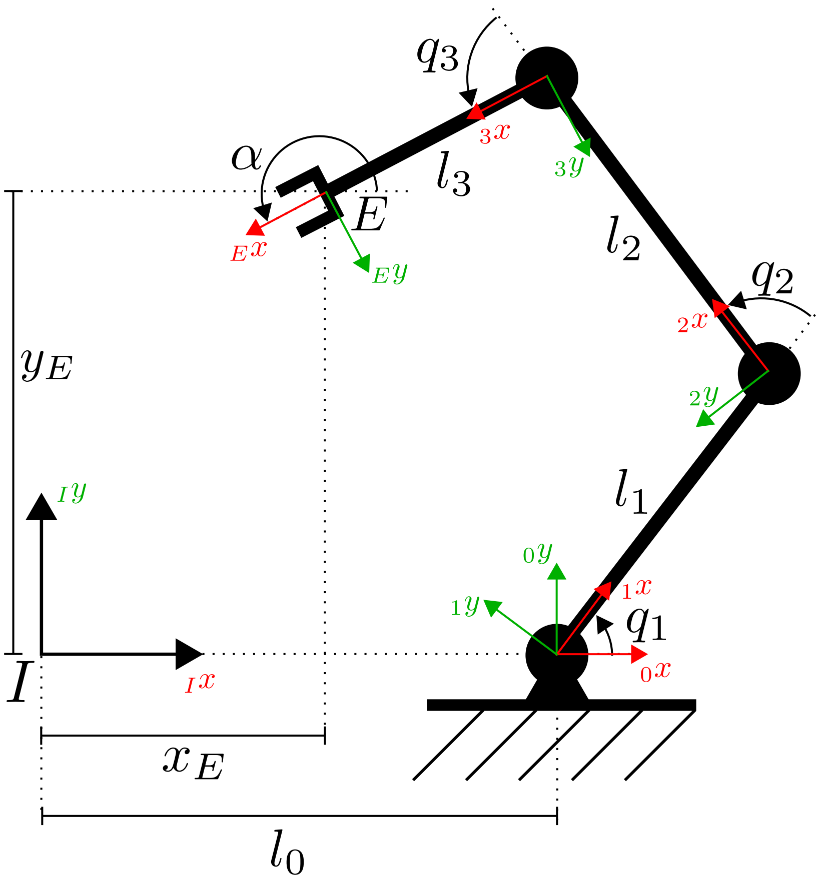


Figure 1: The three degree of freedom planar robotic arm considered in this document.

## 2 Questions

In this quiz, you will model the forward, differential, and inverse kinematics of the robot arm shown in Fig. 1. It is a fixed base manipulator with 3 degrees of freedom. All joints rotate around the positive $z$ axis. The generalized coordinates are defined as

$$\boldsymbol{q} = \left[ \begin{array}{ccc} q_1 & q_2 & q_3 \end{array} \right]^\top . \tag{1}$$

The arm is composed by three links with lengths $l_1$, $l_2$, and $l_3$ respectively and is displaced by $l_0$ from the inertial frame $I$ along the $_Ix$ axis.
In the following questions, all required parameters are passed to your functions in a structure called `params`. You can access it as follows:

```
l0 = params.l0 % length l0
l1 = params.l1 % length l1
l2 = params.l2 % length l2
l3 = params.l3 % length l3
```

**Question 1.** 6 P.

Find the homogeneous transform between the inertial frame $I$ and the end-effector frame $E$, i.e., the matrix $\mathbf{T}_{IE}$ as a function of the generalized coordinates $\boldsymbol{q}$.
*Hint:* Try to find the transforms of subsequent frames first.

You should implement your solution in the function `jointToEndeffectorPose.m`

```matlab
function [ T_IE ] = jointToEndeffectorPose( q, params )
  % q: a 3x1 vector of generalized coordinates
  % params: a struct of parameters

  % Link lengths (meters)
  l0 = params.l0;
  l1 = params.l1;
  l2 = params.l2;
  l3 = params.l3;

  % Joint positions
  q1 = q(1);
  q2 = q(2);
  q3 = q(3);

  T_IE = []; % implement your solution here
end
```

**Question 2.** 6 P.

Derive the geometric Jacobian ${}_I\mathbf{J}_0 \in \mathbb{R}^{6\times 3}$ of the end-effector. This Jacobian should map the generalized velocities $\dot{\boldsymbol{q}}$ to linear and angular velocities in $I$ frame, i.e.,

$$\begin{pmatrix} {}_I\boldsymbol{v}_E \\ {}_I\boldsymbol{\omega}_E \end{pmatrix} = {}_I\mathbf{J}_0 \, \dot{\boldsymbol{q}} \,. \tag{2}$$

*Hints:*

1. Use our provided functions for the transforms
   (Note: These transforms relate to a different choice of coordinate frames, hence you cannot use them to solve Question 1).
2. The MATLAB function for cross product $\boldsymbol{a} \times \boldsymbol{b}$ is `cross(a,b)`

You should implement your solution in the function `jointToGeometricJacobian.m`

```matlab
function I_J = jointToGeometricJacobian(q, params)
  % q: a 3x1 vector of generalized coordinates
  % params: a struct of parameters

  % Get the homogeneous transforms between each pair of coordinate ...
      frames.
  T_I0 = T_I0_solution(q, params);
  T_01 = T_01_solution(q, params);
  T_12 = T_12_solution(q, params);
  T_23 = T_23_solution(q, params);
  T_3E = T_3E_solution(q, params);

  % Implement your solution here.
  I_Jp = [];
  I_Jr = [];

  I_J = [I_Jp;
         I_Jr];
end
```

**Question 3.** 3 P.

Implement an iterative inverse kinematics algorithm for the robot arm. Your implementation should return the joint angles $\boldsymbol{q}$ that achieve a given position and orientation of the end-effector.
For this question, we provide

- the analytical Jacobian $\mathbf{J}_A \in \mathbb{R}^{3\times 3}$ that fulfills

$$\begin{pmatrix} \dot{x}_E \\ \dot{y}_E \\ \dot{\alpha} \end{pmatrix} = \mathbf{J}_A \, \dot{\boldsymbol{q}} \, . \tag{3}$$

  You can call it with `jointToAnalyticJacobian_solution(q, params)`

- a function for calculating damped pseudo-inverses, as you have seen in the exercise: `pseudoInverseMat_solution(J, lambda)`

To compute the forward kinematics needed in this exercise, you can use the function you implemented in Question 1 or use the transform functions provided in Question 2 (e.g., `T_01_solution(q, params)`).

You should implement your solution in the function `inverse_kinematics.m`

```
function [ q ] = inverse_kinematics( x_E_des, y_E_des, alpha_des, ...
    q_0, tol, params )
  % x_E_des: 1x1 desired x position of end-effector
  % y_E_des: 1x1 desired y position of end-effector
  % alpha_des: 1x1 desired alpha angle of end-effector
  % q_0: 3x1 initial guess for joint angles
  % tol: 1x1 tolerance to use as termination criterion
  %      The tolerance should be used as:
  %      norm([x_E_des; y_E_des; alpha_des] - [x_E; y_E; alpha]) < tol
  % params: a struct of parameters

  % Returns a vector of joint angles q (3x1) which
  % achieves the desired task space pose.

  q = []; % Implement your solution here.
end
```