Exercise 1b: Differential Kinematics of the ABB IRB 120

Marco Hutter, Michael Blösch, Dario Bellicoso, Samuel Bachmann

October 4, 2016

Abstract

The aim of this exercise is to calculate the differential kinematics of an ABB robot arm. You will practice on the derivation of velocities for a multi-body system, as well as derive the mapping of between generalized velocities and end-effector velocities. A separate MATLAB script will be provided for the 3D visualization of the robot arm.

1 Introduction

The following exercise is based on an ABB IRB 120 depicted in figure 1. It is a 6-link robotic manipulator with a fixed base. During the exercise you will implement several different MATLAB functions, which you should test carefully since the next exercises will depend on them. To help you with this, we have provided the script prototypes at `bitbucket.org/ethz-asl-lr/robotdynamics_exercise_1b` together with a visualizer of the manipulator.

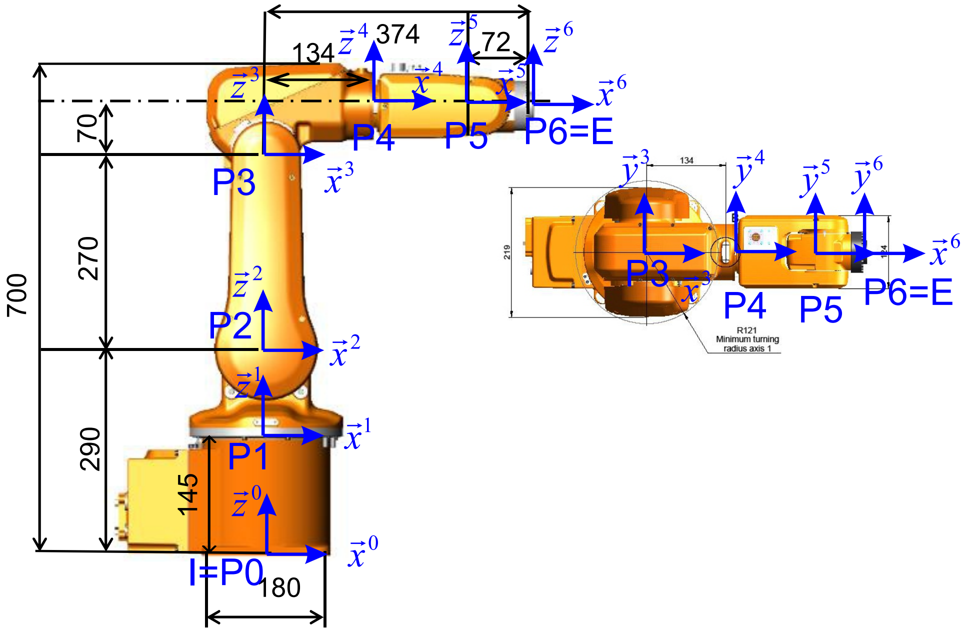

Figure 1: ABB IRB 120 with coordinate systems and joints

Throughout this document, we will employ I for denoting the inertial world coordinate system (which has the same pose as the coordinate system P0 in figure 1) and E for the coordinate system attached to the end-effector (which has the same pose as the coordinate system P6 in figure 1).

2 Differential Kinematics

Exercise 2.1

In this exercise, we seek to compute an analytical expression for the twist ${}_{\mathcal{I}}\mathbf{w}_E = \begin{bmatrix} {}_{\mathcal{I}}\mathbf{v}_E^T & {}_{\mathcal{I}}\boldsymbol{\omega}_E^T \end{bmatrix}^T$ of the end-effector. To this end, find the analytical expression of the end-effector linear velocity vector ${}_{\mathcal{I}}\mathbf{v}_E$ and angular velocity vector ${}_{\mathcal{I}}\boldsymbol{\omega}_{IE}$ as a function of the linear and angular velocities of the coordinate frames attached to each link. *Hint: start by writing the transport theorem and extend it to the case of a 6DoF arm.*

Exercise 2.2

This exercise focuses on deriving the mapping between the generalized velocities $\dot{\mathbf{q}}$ and the end-effector twist ${}_{\mathcal{I}}\mathbf{w}_E$, namely the *basic* or *geometric* Jacobian ${}_{\mathcal{I}}\mathbf{J}_{e0} = \begin{bmatrix} {}_{\mathcal{I}}\mathbf{J}_P^T & {}_{\mathcal{I}}\mathbf{J}_R^T \end{bmatrix}^T$. To this end, you should derive the translational and rotational Jacobians of the end-effector, respectively ${}_{\mathcal{I}}\mathbf{J}_P$ and ${}_{\mathcal{I}}\mathbf{J}_R$. To do this, you can start from the derivation you found in exercise 1. The Jacobians should depend on the minimal coordinates $\mathbf{q}$ only. Remember that Jacobians map joint space generalized velocities to operational space generalized velocities:

$$
\begin{aligned}
{}_{\mathcal{I}}\mathbf{v}_{IE} &= {}_{\mathcal{I}}\mathbf{J}_P(\mathbf{q})\dot{\mathbf{q}} && (1)\\
{}_{\mathcal{I}}\boldsymbol{\omega}_{IE} &= {}_{\mathcal{I}}\mathbf{J}_R(\mathbf{q})\dot{\mathbf{q}} && (2)
\end{aligned}
$$

Please implement the following two functions:

```matlab
function J_P = jointToPosJac(q)
  % Input: vector of generalized coordinates (joint angles)
  % Output: Jacobian of the end-effector translation which maps joint
  % velocities to end-effector linear velocities in I frame.

  % Compute the translational jacobian.
  J_P = ... ;

end

function J_R = jointToRotJac(q)
  % Input: vector of generalized coordinates (joint angles)
  % Output: Jacobian of the end-effector orientation which maps joint
  % velocities to end-effector angular velocities in I frame.

  % Compute the rotational jacobian.
  J_R = ... ;

end
```