Out: January 20, 2022
Due: February 4, 2022

NTNU – Trondheim
Norwegian University of
Science and Technology

TTK4130 Modeling and Simulation
Assignment 2

Introduction

Rotation matrices and angular velocity vectors are fundamental for describing the kinematics of a body's orientation. In the context of control theory, these quantities are usually described using parametrizations, such as Euler angles or unit quaternions, as explained in the book. Reasons for this approach are to reduce the computational complexity in real-time systems, and to relate the measurements as directly as possible to the states. However, in the context of modeling and simulation, a theoretically simpler and more direct approach can be used to simulate the rotation of a body, as we will study in Problem 2.

For this problem and the course in general, it will be very useful to make use of the Matlab Symbolic Math Toolbox™. The main commands of this toolbox include:

- `syms x y z real` declares scalar symbolic variables `x,y,z`.
- `A = sym('A',[n,m])` creates a matrix `A` of symbolic variables of size $n \times m$.
- `matlabFunction(expr1,expr2,'File','myfilename','Vars',t,[x y z])` exports a Matlab function called `'myfilename'`, which takes the two arguments specified by `t` and `[x y z]`. The function has (up to) two output arguments, evaluated as the symbolic expressions `expr1` and `expr2`. After the execution of this line, a new file `'myfilename'` will be created on your working folder with the corresponding function. This function can be called using `[expr1,expr2] = myfilename(t,[x,y,z])]`, where `t` and `[x,y,z]` are numerical values, and the numerical evaluations of `expr1` and `expr2` are returned.

In order to get used to manipulate this toolbox, you ought to try these different commands.

Problem 1 (Rotation matrices)

(a) Use the definition of SO(3) to find the missing values in the following rotation matrices:

$$\mathbf{R}_1 = \begin{bmatrix} * & 1 & * \\ 1 & * & * \\ * & * & * \end{bmatrix}, \quad \mathbf{R}_2 = \begin{bmatrix} \frac{5}{13} & * & * \\ * & 1 & * \\ \frac{12}{13} & * & * \end{bmatrix}.$$

Let $a = \{O, \vec{a}_1, \vec{a}_2, \vec{a}_3\}$ and $b = \{O, \vec{b}_1, \vec{b}_2, \vec{b}_3\}$ be two reference frames, where O is the common origin, $\vec{a}_1, \vec{a}_2, \vec{a}_3$ are the orthogonal unit vectors that give the axes of frame a, and $\vec{b}_1, \vec{b}_2, \vec{b}_3$ are the orthogonal unit vectors that give the axes of frame b.

Consider the rotation matrix from a to b, $\mathbf{R}_b^a$.

(b) The columns of $\mathbf{R}_b^a$ are the coordinates of some particular vectors in some particular frame. What vectors and what frame are these? Explain.

(c) Show that $(\mathbf{u}^a)^T \mathbf{v}^a = (\mathbf{u}^b)^T \mathbf{v}^b$ for all vectors $\mathbf{u}, \mathbf{v} \in \mathbb{R}^3$. What is the geometrical interpretation of this identity?

(d) For any rotation matrix $\mathbf{R}$ and vector $\mathbf{u}$, we have that $(\mathbf{R}\mathbf{u})^\times = \mathbf{R}\mathbf{u}^\times \mathbf{R}^T$. Use this identity to show that $\mathbf{u}^a \times \mathbf{v}^a = \mathbf{R}_b^a(\mathbf{u}^b \times \mathbf{v}^b)$. What is the geometrical interpretation of this identity?

Problem 2 (Angular velocities)

In this task, we will simulate the kinematic of a body's orientation for a given angular velocity $\vec{\omega}$. We will simulate the kinematic equations using the Matlab ODE integration function `ode45` (which we will discuss later in the course). You will find code templates on Blackboard to help you get started. `MainKinematic.m` provides a template for building the simulation and a 3D animation of the results. `Kinematics.m` is a Matlab function where the derivative of the state, $\dot{\boldsymbol{x}}$, is to be calculated as a function of the state and other parameters. For more information, see the file `ReadMe.txt`.

(a) Consider an SO(3) representation using Euler angles. More specifically we choose:

$$R_b^a = R_1(\rho)R_2(\theta)R_3(\psi)\,, \tag{1}$$

where

$$R_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\rho) & -\sin(\rho) \\ 0 & \sin(\rho) & \cos(\rho) \end{bmatrix}, \quad R_2 = \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{bmatrix}, \quad R_3 = \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

In order to compute $\dot{\rho}$, $\dot{\theta}$, $\dot{\psi}$ as a function of ρ, θ, ψ and $\vec{\omega}$, **complete the code of `SymbolicEuler.m`.** The routine `SymbolicEuler.m` builds a Matlab function that delivers $R_b^a = R_b^a(\rho, \theta, \psi)$ and a matrix $M = M(\rho, \theta, \psi)$ that fulfills

$$\boldsymbol{\omega}_{ab}^b = M \begin{bmatrix} \dot{\rho} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix}. \tag{2}$$

The parts of the code that are missing are labeled with `complete`.

(b) Implement/modify `Kinematics.m`. Hints are provided in the code draft, and in the ReadMe-file.

(c) Modify `MainKinematic.m` in order to simulate the body kinematics using different fixed vectors $\boldsymbol{\omega}_{ab}^b$. Parts to fill out are labeled with `MODIFY`.

Briefly explain your modifications in (a-c), and add them to your answer (code screen shot is fine, or in some other way).

(d) Try different fixed vectors $\boldsymbol{\omega}_{ab}^b$. Do the 3D-simulations give reasonable results? Explain.

(e) We will now consider a direct representation of R_b^a, which is often called **Direct Cosine Matrix** (DCM). Here, we will simply carry the entire matrix R_b^a as a state of the system, and proceed to simulate the dynamics:

$$\dot{R}_b^a = R_b^a \left(\boldsymbol{\omega}_{ab}^b\right)^\times \tag{3}$$

- Implement the DCM approach in Matlab by modifying `Kinematics.m` and modify `MainKinematic.m` accordingly, and perform the same simulations as you did in the previous part.
- Explain your modifications, and add them to your answer.
- Do the 3D-simulations give reasonable results? Explain.
- Compare the Euler angles and the Direct Cosine Matrix approaches. Are the simulations different? Which approach is simpler? Number of states?

Hint: `ode45` only works with state vectors. Therefore, we recommend to use the Matlab function `reshape` to convert R_b^a back-and-forth between a 3×3 matrix and a 9×1 vector.

Problem 3 (Angle-axis representation, Sheeperd's method)

Any rotation matrix can be represented as a rotation by an angle θ about an axis $\mathbf{k}$. This is known as the **angle-axis representation**. Moreover, the rotation matrix $\mathbf{R}$ can be written as

$$\mathbf{R} = \mathbf{R}_{\mathbf{k},\theta} = \cos\theta\mathbf{I} + \sin\theta\mathbf{k}^{\times} + (1-\cos\theta)\mathbf{k}\mathbf{k}^{T}. \tag{4}$$

(a) Show that $\mathbf{k} = \mathbf{R}\mathbf{k}$.
What is the geometrical interpretation of this identity?

Shepperd's method is an algorithm for calculating the angle-axis representation of a rotation matrix (see section 6.7 in the book).

(b)
- Implement a Matlab **function** that calculates the rotation angle θ and the rotation axis $\mathbf{k}$ for an arbitrary rotation matrix $\mathbf{R}$. Add the Matlab script to your answer.
- Find the rotation axis and rotation angle for each of the rotation matrices in part 1(a), and add to your answer.
- Are the obtained results reasonable? Explain.