RBE 501 – Robot Dynamics

Spring 2022 – Instructor: L. Fichera

Mini-Project 2

PROBLEM STATEMENT

The goal of this project is to develop an algorithm to numerically calculate the mass matrix of any robotic manipulator. Knowledge of the mass matrix is useful to implement forward dynamics algorithms, e.g., for the purpose of simulating a robot; or to implement model-based control schemes (see Chapter 11 in the textbook).

Motivation: So far in the course, we have learned how to calculate the mass matrix of a robotic arm using the Lagrangian formulation of dynamics. As we have seen in class, this formulation is elegant and enables us to calculate the mass matrix in analytical form. Unfortunately, the Lagrangian approach can also be cumbersome to apply to robots with more than 3 degrees of freedom. It is therefore desirable to derive an alternative, more practical approach.

Recommended Approach: To calculate the mass matrix, we can implement the algorithm described in Section 8.4 of the textbook. Briefly, let us first note that the kinetic energy of a robotic arm can be expressed as the sum of the kinetic energies of each link:

$$\mathcal{K} = \frac{1}{2}\sum_{i=1}^{n} \mathcal{V}_i^{\mathrm{T}} \mathcal{G}_i \mathcal{V}_i$$

where $\mathcal{V}_i$ and $\mathcal{G}_i$ are the twist and spatial inertia matrix of link i, respectively, and n denotes the total number of links in the robot. In the expression above, both $\mathcal{V}_i$ and $\mathcal{G}_i$ are expressed with respect to the local link frame $\{i\}$.

The kinetic energy can also be expressed as a function of the joint velocities $\dot{\boldsymbol{\theta}} \in \mathbb{R}^n$ and the mass matrix $M(\boldsymbol{\theta})$:

$$\mathcal{K} = \frac{1}{2}\, \dot{\boldsymbol{\theta}}^{\mathrm{T}} M(\boldsymbol{\theta}) \dot{\boldsymbol{\theta}}$$

The two formulations of the kinetic energy that we have introduced are equivalent, therefore:

$$\sum_{i=1}^{n} \mathcal{V}_i^{\mathrm{T}} \mathcal{G}_i \mathcal{V}_i = \dot{\boldsymbol{\theta}}^{\mathrm{T}} M(\boldsymbol{\theta}) \dot{\boldsymbol{\theta}}$$

Now, let us observe that the relation between the twist $\mathcal{V}_i$ of each link and the vector of joint velocities is given by the following velocity kinematics equation:

$$\mathcal{V}_i = J_{ib}(\boldsymbol{\theta})\dot{\boldsymbol{\theta}}$$

where $J_{ib}(\boldsymbol{\theta})$ is the *Body Jacobian* of link i. Note that, in principle, $J_{ib}(\boldsymbol{\theta})$ would be defined as a $6 \times i$ matrix (the motion of joints $\{i+1, \dots, n\}$ has no effect on the velocity of the ith link). To turn $J_{ib}(\boldsymbol{\theta})$ into a $6 \times n$ matrix, we can add $n - i$ columns filled with zeros.

Considering the above, we can write:

$$\sum_{i=1}^{n} \mathcal{V}_i^{\mathrm{T}}\,\mathcal{G}_i \mathcal{V}_i = \dot{\boldsymbol{\theta}}^{\mathrm{T}}\left[\sum_{i=1}^{n} J_{ib}^{\mathrm{T}}(\boldsymbol{\theta})\,\mathcal{G}_i\,J_{ib}(\boldsymbol{\theta})\right]\dot{\boldsymbol{\theta}} = \dot{\boldsymbol{\theta}}^{\mathrm{T}} M(\boldsymbol{\theta})\dot{\boldsymbol{\theta}}$$

from which it finally follows that:

$$M(\boldsymbol{\theta}) = \sum_{i=1}^{n} J_{ib}^{\mathrm{T}}(\boldsymbol{\theta})\,\mathcal{G}_i\,J_{ib}(\boldsymbol{\theta})$$

The equation above is equivalent to Eq.(8.57) in the textbook, and it gives us a formula to numerically evaluate the mass matrix $M(\boldsymbol{\theta})$ based on the vector of joint variables $\boldsymbol{\theta}$ and the spatial inertia matrices $\mathcal{G}_i$.

<u>**Specifications**</u>: The algorithm must be implemented in the MATLAB programming language, and it must be encapsulated in a function with the following signature:

```
function M = MassMatrixCalculator(currentQ, S, M, G)
```

The expected inputs are:

- `currentQ:` nx1 vector of joint variables
- `S:` 6xn matrix whose columns are the screw axes of the robot, expressed in the space frame
- `M:` 4x4x(n+1) array containing the (n+1) homogeneous transformation matrices between consecutive frames
- `G:` 6x6xn array containing the n spatial inertia matrices

The function is expected to return a $n \times n$ mass matrix.

GRADING RUBRIC

The function will be tested on a set of 1000 randomly-generated vectors of joint variables. Tests will be carried out using the UR-5 robot (the starting code on Canvas provides the screw axes and inertial properties of the robot), but the code should work for any manipulator. Each correct solution will contribute 0.1 points to the final score (maximum possible score: 100).