

Crash Course: Mathematica and Robotica

1 Mathematica Basics

- To execute a cell press `<Shift>+<Enter>` on the keyboard or `<Enter>` on the numeric keypad. Pressing the keyboard `<Enter>` will simply move you to a new line in the same cell.
- Define a matrix using curly braces around each row, commas between each entry, commas between each row, and curly braces around the entire matrix. For example:

$$M = \{\{1, 7\}, \{13, 5\}\}$$

- To display a matrix use `MatrixForm` which organizes the elements in rows and columns. If you constructed matrix `M` as in the previous example, entering `MatrixForm[M]` would generate the following output:

$$\begin{pmatrix} 1 & 7 \\ 13 & 5 \end{pmatrix}$$

If your matrix is too big to be shown on one screen, Mathematica and Robotica have commands that can help (see the final section of this document).

- *To multiply matrices do not use the asterisk.* Mathematica uses the decimal for matrix multiplication. For example, `T=A1.A2` multiplies matrices `A1` and `A2` together and stores them as matrix `T`.
- Notice that Mathematica commands use square brackets and are case-sensitive. Typically, the first letter of each word in a command is capitalized, as in `MatrixForm[M]`.
- Trigonometric functions in Mathematica operate in radians. It is helpful to know that π is represented by the constant `Pi` (Capital 'P', lowercase 'i'). You can convert easily from a value in degrees to a value in radians by using the command `Degree`. For example, writing `90 Degree` is the same as writing `Pi/2`.

2 Writing a Robotica Source File

- Robotica takes as input the Denavit-Hartenberg parameters of a robot. Put DH frames on a figure of your robot and compute the table of DH parameters. Open a text editor (notepad in Windows will be fine) and enter the DH table. You will need to use the following form:

```
DOF=2
  (Robotica requires a line between DOF and joint1)
joint1 = revolute
a1      = 3.5
alpha1 = Pi/2
d1      = 4
theta1  = q1
joint2 = prismatic
a2      = 0
alpha2  = 90 Degree
d2      = q2
theta2  = Pi
```

- You may save your file with any extension, but you may find ‘.txt’ to be useful, since it will be easier for your text editor to recognize the file in the future.
- Change the degrees of freedom line (‘DOF=’) to reflect the number of joints in your robot.
- Using joint variables with a single letter followed by a number (like ‘q1’ and ‘q2’) will work well with the command that simplifies trigonometric notation, which we’ll see momentarily.

3 Robotica Basics

- Download ‘robotica.m’ from the website. Find the root directory for Mathematica on your computer and save the file in the /AddOns/ExtraPackages directory.
- Open a new notebook in Mathematica and load the Robotica package by entering

```
<< robotica.m
```

- Load your robot source file. This can be done in two ways. You can enter the full path of your source file as an argument in the `DataFile` command:

```
DataFile["C:\\directory1\\directory2\\robot1.txt"]
```

Notice the double-backslash after `C:` and single-backslashes elsewhere in the path. You can also enter `DataFile[]` with no parameter. In this case, Mathematica will produce a window that asks you to supply the full path of your source file. You do not need double-backslashes when you enter the path in the window. *You will likely see a message warning you that no dynamics data was found.* Don't worry about this message; dynamics is beyond the scope of this course.

- Compute the forward kinematics for your robot by entering `FKin[]`, which generates the A matrices for each joint, all possible T matrices, and the Jacobian.
- To view one of the matrices generated by forward kinematics, simply use the `MatrixForm` command mentioned in the first section. For example, `MatrixForm[A3]` will display the homogeneous transformation A3.

4 Simplifying and Displaying Large, Complicated Matrices

- Mathematica has a powerful function that can apply trigonometric identities to complex expressions of sines and cosines. Use the `Simplify` function to reduce a complicated matrix to a simpler one. *Be aware that Simplify may take a long time to run.* It is not unusual to wait 30 seconds for the function to finish. For example, `T=Simplify[T]` will try a host of simplification algorithms and redefine T in a simpler equivalent form. You may be able to view all of the newly simplified matrix with `MatrixForm`.
- Typically, matrices generated by forward kinematics will be littered with sines and cosines. Entering the command `SimplifyTrigNotation[]` will replace `Cos[q1]` with `c1`, `Sin[q1+q2]` with `s12`, etc. when your matrices are displayed. Executing `SimplifyTrigNotation` will not change previous output. However, all following displays will have more compact notation.

- If your matrix is still too large to view on one screen when you use `MatrixForm`, the `EPrint` command will display each entry in the matrix one at a time. The `EPrint` command needs two parameters. The first is the name of the matrix to be displayed, the second is the label used to display alongside each entry. For example, entering `EPrint[T, "T"]` will display all sixteen entries of the homogeneous transformation matrix `T` individually.

5 Example

Consider the three-link revolute manipulator shown in Figure 1. The figure shows the DH frames with the joint variables $\theta_1, \theta_2, \theta_3$ and parameters a_1, a_2, a_3 clearly labeled. The corresponding table of DH parameters is given in Table A.1.

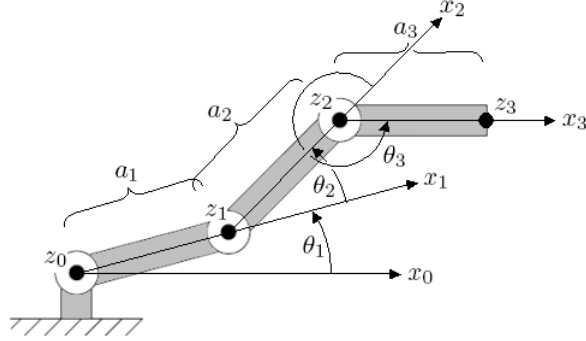


Figure 1: Three-link revolute manipulator with DH frames shown and parameters labeled. The z axes of the DH frames are pointing out of the page.

joint	a	α	d	θ
1	a_1	0	0	θ_1
2	a_2	0	0	θ_2
3	a_3	0	0	θ_3

Table 1: Table of DH parameters corresponding to the frames assigned in Figure 1.

We open a text editor and create a text file with the following contents.

```
DOF=3
  (Robotica requires a line between DOF and joint1)
joint1 = revolute
a1      = a1
alpha1 = 0
d1      = 0
theta1  = q1
joint2 = revolute
a2      = a2
alpha2 = 0
d2      = 0
theta2  = q2
joint3 = revolute
a3      = a3
alpha3 = 0
d3      = 0
theta3  = q3
```

We save the file as `c:\example.txt` . After loading Mathematica, we enter the following commands.

```
<< robotica.m
DataFile["C:\\example.txt"]
```

If all goes well, the Robotica package will be loaded successfully, and the example datafile will be opened and its contents displayed in a table.

No dynamics data found.

Kinematics Input Data

```
-----
Joint   Type      a    alpha  d    theta
1       revolute  a1   0       0    q1
2       revolute  a2   0       0    q2
3       revolute  a3   0       0    q3
```

Now generate the forward kinematic matrices by entering the following command.

```
FKin[]
```

Robotica will generate several status lines as it generates each A and T matrix and the Jacobian. Now we will view one of these matrices. We decide to view the T_3^0 matrix. Entering

```
MatrixForm[T[0,3]]
```

generates the output

$$\begin{pmatrix} \cos[q_1 + q_2 + q_3] & -\sin[q_1 + q_2 + q_3] & 0 & a_1 \cos[q_1] + \dots \\ \sin[q_1 + q_2 + q_3] & \cos[q_1 + q_2 + q_3] & 0 & a_1 \sin[q_1] + \dots \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

which is too wide to fit on this page. We wish to simplify the notation, so we enter

```
SimplifyTrigNotation[]
```

before displaying the matrix again. Now, entering the command

```
MatrixForm[T[0,3]]
```

generates the much more compact form

$$\begin{pmatrix} c_{123} & -s_{123} & 0 & a_1 c_1 + a_2 c_{12} + a_3 c_{123} \\ s_{123} & c_{123} & 0 & a_1 s_1 + a_2 s_{12} + a_3 s_{123} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

6 What Must Be Submitted with Robotica Assignments

For homework and lab assignments requiring Robotica, you must submit each of the following:

1. figure of the robot clearly showing DH frames and appropriate DH parameters
2. table of DH parameters
3. matrices relevant to the assignment or application, simplified as much as possible and displayed in `MatrixForm`.

Do not submit the entire hardcopy of your Mathematica file. Rather, cut the relevant matrices from your print out and paste them onto your assignment. Also, remember to simplify the matrices as much as possible using the techniques we have presented in section A.4 of this Appendix.

- Mathematically simplify the matrix using `Simplify`.
- Simplify the notation using `SimplifyTrigNotation`. Remember that all mathematical simplification must be completed *before* running `SimplifyTrigNotation`.
- If the matrix is still too large to view on the screen when you use `MatrixForm`, use the `EPrint` command to display the matrix one entry at a time.