Visualizing Manipulators in MATLAB using the Robotics Toolbox

Assuming you are running MATLAB 2010a or later:

1) Download the Robotics Toolbox from the ME 449 class website.
2) Unzip the folder into an empty directory and note where you unzipped the toolbox.
3) Start MATLAB and navigate to the folder where you unzipped the files (or double-click *wam_demo.m*).
4) In MATLAB, execute the script *wam_demo.m*. If you see an animated arm, then you're done!
5) Study the file and use it as a template for future homework problems.
6) Browse the documentation, *robot.pdf* in robot-9.10/rvctools/robot/, to learn more about the Robotics Toolbox. There are functions in the toolbox that may be useful throughout the term.

Directory Structure

File or Folder	Description
robot-9.10/	The Robotics Toolbox. You can learn more about the toolbox and download the latest version from http://petercorke.com/Robotics_Toolbox.html.
model_ur5.m	A script that creates and loads the Universal UR5 manipulator model into your MATLAB workspace as **ur5**.
model_wam.m	A script that creates and loads the Barrett 7-dof WAM manipulator model into your MATLAB workspace as **wam**.
wam_demo.m	An example script that plots an animated trajectory of the WAM arm given a matrix of configurations.

About the Robotics Toolbox

The Robotics Toolbox [1] is maintained and originally written by Prof. Peter Corke of Queensland University of Technology. The homepage is http://petercorke.com/Robotics_Toolbox.html.

The Robotics Toolbox represents serial-link manipulators using objects in MATLAB (specifically a SerialLink object). In MATLAB, an object has variables and methods that are accessed using a dot '.' notation. For example, to plot a robot's configuration q, we would call **robot.plot(q)** and a figure will pop up showing the robot's configuration. As another example, we can get the number of joints in the manipulator with the syntax **robot.n**. An entire list of variables and methods can be found in *robot.pdf* or by typing: **doc SerialLink** in the MATLAB command window. In ME 449, we will use kinematic and dynamic models of the Barrett WAM and Universal UR5 manipulators. These two robot models will be represented by SerialLink objects **wam** and **ur5**. They can be loaded into your workspace environment by calling *mdl_wam.m* and *mdl_ur5.m*, respectively. Example code can be found in *wam_demo.m*. If *wam_demo.m* does not work, then see if your issue is addressed in the toolbox's FAQ: https://code.google.com/p/matlab-toolboxes-robotics-vision/wiki/FAQ.

[1] P.I. Corke, "Robotics, Vision & Control", Springer 2011, ISBN 978-3-642-20143-1

Appendix: *wam_demo.m*

```
%% Load the Robotics Toolbox into Matlab's path
run('robot-9.10/rvctools/startup_rvc.m') % you only have to run this once

%% Load the WAM model into the workspace
% load the 7-dof Barrett WAM model into the workspace.
mdl_wam

% mdl_wam creates a variable "wam", which represents a WAM arm
wam % output the arm to the command window
% wam is a SerialLink object, which has various functions and properties

% learn more about the variables and functions of a SerialLink object
% doc SerialLink -or- help SerialLink

%% Animate a trajectory using the model
n = wam.n; % number of joints
q0 = zeros(1, n); % home configuration
qf = rand(1, n)*2*pi; % final configuration
k = 25; % create k points on a path from q0 to qf

% generate a linear path between q0 and qf
Q = zeros(k, n);
for i = 1:n
    Q(:, i) = linspace(q0(i), qf(i), k);
end
% Q is now a k x n matrix representing a path in configuration space of
% the manipulator.  Each row in the matrix is a configuration along the
% path, where Q(1,:) is the initial configuration q0 and Q(k,:) is the
% final configuration qf.

% import from an external file instead
% Q = dlmread('Q.csv'); % help dlmread to learn more

% animate the trajectory of the robot arm --remember each row is a new pose
wam.plot(Q) % all SerialLink objects, like wam, have a plot function
% wam.plot(Q, 'movie', 'wam_imgs') % save poses as PNGs in folder wam_imgs

%% Plot a single pose
% call plot with a 1 x n row vector for a single configuration
% wam.plot(q0)
% wam.plot(Q(1,:))

%% Done!
% see robot.pdf for more info about the Robotics Toolbox.
% open('robot.pdf')

%% The UR5 manipulator
% The other model we'll be working with is the Universal UR5 arm.
% Uncomment the last two lines in this file to load the UR5 model into the
% workspace or create a new file and, using the lines above as reference,
% replace ur5 with wam to get a plot for the Universal UR5 manipulator.

% mdl_ur5;
% ur5  % output the arm to the command window
```