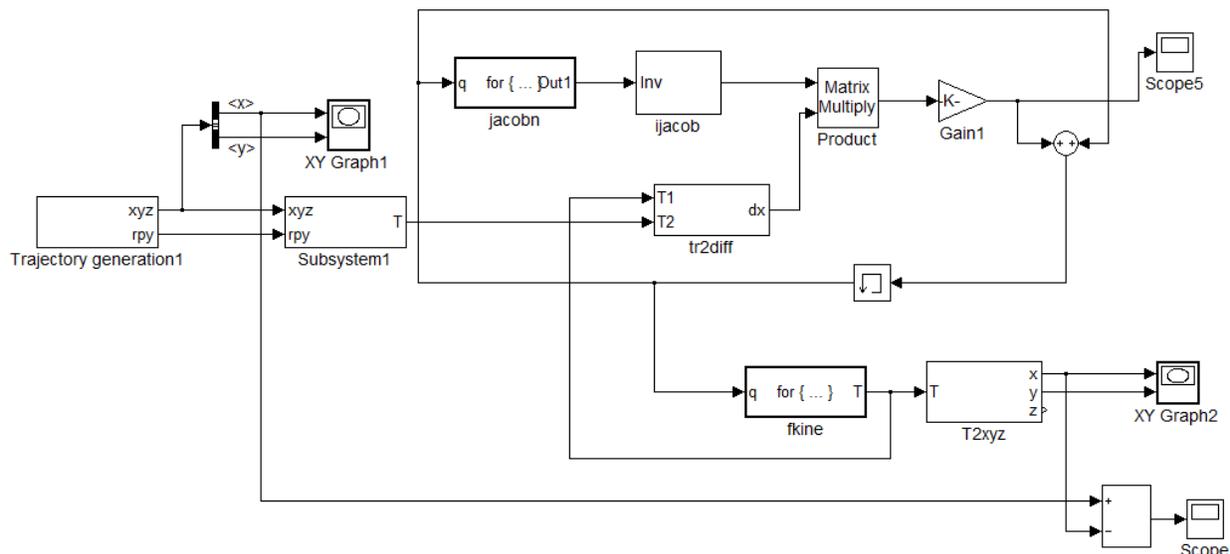


Assignment 3 (Revised). Software-In-The-Loop Simulation and Code Generation (100 points)

In this assignment, you will exercise model-based development and code generation process by implementing a software-in-the-loop simulation for robot motion control application. The application model is a 6 DOF robot from <http://www.csl.sri.com/~shankar/ModelRob.tar> which is built with Matlab Robotics Toolbox. The model robot_circular is illustrated in the following figure where the trajectory generation block puts out XY coordination of a circle and the robot model uses a simple close-loop control to rotate joints such that the end-effector follows the target trajectory. The major blocks in the model include:

- jacobn – to compute manipulator Jacobian in end-effector coordinate frame. The output is a 6x6 matrix
- ijacob – to compute inverse Jacobian for resolved-rate motion control. The output is a 6x6 matrix.
- fkine module – to compute forward kinematics to find the position and orientation of the end-effector. This is computed from a given set of joint angles. It returns a 4x4 matrix of the homogeneous transformation corresponding to the end-effector position.
- tr2diff module – to compute differential motion vector from homogeneous transformations. It produces a 6-vector of Cartesian differential, or scaled spatial velocity.
- Subsystem1 module – to compute inverse kinematics (calculate all possible sets of joint angles from a given position and orientation of the end-effector) and consist of xyz2T and rpy2T sub-modules for the computation of a 4x4 matrix of homogeneous transformation from Cartesian coordinate and from Roll/Pitch/Yaw angles.
- T2xyz – to compute Cartesian coordinate from homogeneous transformation for display.

For the details of robot motion control, please refer to “Robotics, Vision and Control “, by Peter Corke, <http://robotics.itee.uq.edu.au/~metr4202/tpl/Robotics%20Vision%20&%20Control.pdf>.



Your task in the assignment is to split the model execution in both simulation host and a target system. The goal is to allow user inputs to control a simulated robot in real time. The target is a Linux system while the simulated robot runs in Simulink host environment. In the target, the user can use a mouse to enter XY trajectory. At the same time, the robot’s generalized joint coordinates, i.e., the angles

between joints, are read from the simulated robot. Then, a spatial velocity is computed in the target and sent to the simulated robot, to control robot movement.

For this assignment, we will keep Jacobn, iJacob, Product, Gain1, Add, memory, and XY Graph blocks in the Simulink host. All other coordinator transformation, fkine, tr2diff blocks are moved to the target. Thus, a simulated robot is in the host and a robot controller is implemented in the target. Simulink UDP send and receive blocks should be added in both host and target models. In addition, a mouse block should be implemented in the target to receive the events of mouse movement and button click. The input XY trajectory and the robot end effector position (the output of T2xy) should be sent to the host and displayed in 2 XY graph blocks. Thus, you will be able to control robot end effector position via a mouse in the target, and display the input XY and robot positions in the host. The XY displays should be refreshed (cleared) when the right button of the mouse is clicked.

To accomplish the task, you may consider 3 incremental steps:

1. Run the trajectory generation in the target and the rest of robot model in the simulation host. You will need to establish communication between the host and the target, generate code for the target environment, and manage the execution on the host and the target.
2. Add asynchronous interface for mouse input in the target and use mouse inputs, including button press and mouse movement, to control robot movement. The control functions are:
 - The right button generates a clear signal to XY Graph blocks of the simulation model to refresh the display. For a Simulink XY graph with refresh, please refer to <http://www.mathworks.com/matlabcentral/answers/91467-is-it-possible-for-the-xy-graph-block-in-simulink-7-5-r2010a-to-have-the-capability-to-draw-until>.
 - Mouse XY movement data is used as the input of robot trajectory.
3. Spilt the robot control model as required.

The asynchronous interface for mouse inputs should be implemented as an S-function block in which your C/C++ code can be included. UDP communication channels can be used to connect two separate processes of the robot in the simulation host and the robot controller in the target. The execution of the Simulink model in the host and of the controller in the target can be time or event triggered. For instance, the controller can run an iteration every 100ms or whenever a mouse event and/or a UDP packet arrives. Please note that, when you use Embedded Coder to generate code for Linux target, you can see a comment in `etr_main.c` suggesting that “Attaching `rt_OneStep` to a real-time clock is target specific.” Thus, for a time-trigger approach, you will need to incorporate POSIX timer(s) in the generated code.

Simulink software is available for engineering students at http://help.asu.edu/sims/selfhelp/SelfhelpKbView.seam?parature_id=8373-8193-4466. However, you will need to use Embedded Coder for code generation. Embedded Coder is only available at BYENG 217/215 lab and at CIDSE virtual lab, <http://cidse.engineering.asu.edu/forstudent/labs/>. After the code for the robot controller is generated, you can move (manually) the software to any Linux box and run the controller. Additional information on code generation for Linux target will be made available later.

Due Date

This assignment is due at 11:59pm on April 11.

What to Turn in for Grading

- Create a working directory similar to Assignment 1 to include your Simulink models (of host and target environment) and the generated code for the target, makefile, readme.txt, and a report (in pdf) to describe the structure of the target code and what are happened during its execution. The steps of generating and executing the target code should be included in the report too.
- Compress the directory into a zip file named *cse522-26805-3-firstname-lastname.zip* for online section and *cse522-15970-3-firstname-lastname.zip* for on-campus section. Please note that, for convenience, we only accept zip files. Points will be deducted if the name conversion and the directory structure are not followed.
- Comment your source files properly and write the readme file to describe how to use your software. Also, make sure there will be no warnings when compiling your source code.
- Submit the zip archive to Blackboard by the due date and time.