



6 DOF Hexapod

Laboratory Guide

Hexapod

Quanser Inc.
2016

© 2016 Quanser Inc., All rights reserved.

Quanser Inc.
119 Spy Court
Markham, Ontario
L3R 5H6
Canada
info@quanser.com
Phone: 1-905-940-3575
Fax: 1-905-940-3576

Printed in Markham, Ontario.

For more information on the solutions Quanser Inc. offers, please visit the web site at:
<http://www.quanser.com>

This document and the software described in it are provided subject to a license agreement. Neither the software nor this document may be used or copied except as specified under the terms of that license agreement. All rights are reserved and no part may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of Quanser Inc.

Contents

1	Introduction	4
2	Running the 6 DOF Hexapod	5
2.1	Calibration	5
2.2	Homing	6
2.3	Joint-Level Position Control	7
2.4	World-Space Position Control	9
2.5	Brake Bypass	12
2.6	Earthquake	13
2.7	FFT Analysis	15
2.8	Hexapod Controllers	16
3	Background	23
3.1	Modeling	23
3.2	Position Controller Design	23
3.3	Downloading New Earthquakes	24
4	System Requirements	26
4.1	Overview of Files	27
4.2	Setup script: setup_hexapod.m	28
4.3	Creating an Earthquake: make_quake_xyz.m	30

1 INTRODUCTION

This laboratory manual describes how to use the Quanser 6 DOF Hexapod QUARC controllers. See the 6 DOF Hexapod User Manual [2] for information about the hardware setup (i.e., connections) and specifications.

Topics Covered

- Calibrating the Hexapod on power up.
- Moving the upper stage or platform to the HOME position (i.e., where linear actuator are in center or mid-stroke position).
- Running joint-level and world-space position controls on the 6 DOF Hexapod.
- Transfer function model of the linear actuator.
- PID-based controller used to control position of stage.

Prerequisites

In order to successfully carry out this laboratory, the user should be familiar with the following:

1. See the system requirements in Section 4 for the required hardware and software.
2. Modeling and transfer function representation.
3. PID control design.
4. Basics of **SIMULINK®**.
5. Basics of **QUARC®**, i.e., how to build/run a Simulink model in QUARC.

2 RUNNING THE 6 DOF HEXAPOD

This section explains how to use the supplied **SIMULINK®** models with the **QUARC®** software to run the following controllers on the Hexapod:

1. **Calibration** - Runs an auto routine on all the drives and resets the encoders to zero. Required after system power up.
2. **Home** - Positions the Hexapod to the HOME position where each joint is at the mid-stroke position. Make sure the Hexapod is at home BEFORE running the regular controller (e.g., joint position control).
3. **Joint Level Control** - Control the position of the base joints at the linear actuator level.
4. **World Space Control** - Control the X, Y, Z, Roll, Pitch, and Yaw position of the top stage.
5. **Brake Bypass** - Releases the brake system to move the stage manually.
6. **Earthquake** - Run scaled down version of a recorded earthquake on the table, e.g., Northridge or Kobe earthquake.

Note: All the supplied controllers are already configured for the *hexapod2_usb* data acquisition (DAQ) device. If you are using these files on different DAQ or system, then see the QUARC User Manual [1] for instructions on how to change *Board type* in the HIL Initialize block.

2.1 CALIBRATION

When the Hexapod is first powered ON, the positions measured from the encoders on all the joints are reset to 0, regardless of where the base joint of the Hexapod are actually at. The Hexapod_Calibrate.mdl controller, shown in Figure 2.1, positions each of the six joints along the linear guide until they reach the limit switch. Once all six joints have been placed at the limit switch positions, the encoders are reset and the model is stopped. If the joints are already at the calibrated position then the controller will only reset the encoders and stop the controller. Once calibrated, run the homing procedure outlined in Section 2.2.



Caution: Calibration must be performed EVERY TIME the Hexapod is powered ON and before ANY other controller!

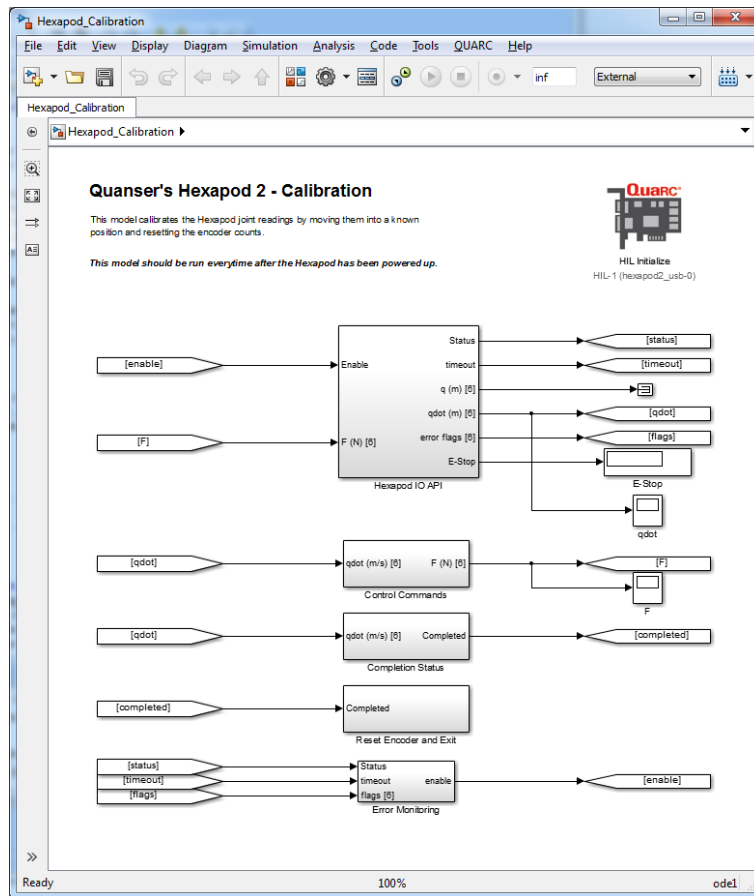


Figure 2.1: Simulink model used to calibrate the 6 DOF Hexapod using QUARC.

Follow these steps to calibrate the system:

1. Make sure the red button is in the upper position to enable the amplifier. The amplifier cannot be enabled when the button is in the lower position. Twist the button to ensure it is in the enabled position.
2. Open the Simulink model called Hexapod_Calibrate.mdl, shown in Figure 2.1.
3. Run setup_hexapod.m in MATLAB (note: script is ran automatically when Simulink model is ran).
4. Click on QUARC | Build to generate the controller.
5. Power ON the Hexapod system by placing the power switch to the ON position.
6. Run the QUARC controller by going to QUARC | Start. Each of the base joints should move to their limits until the limit switches are triggered.



Caution: PRESS DOWN on the RED BUTTON of the E-Stop switch to stop the Hexapod. This deactivates the amplifier and cuts off the DC motor power.

7. The controller stops automatically when each of the six joints have reached their limit switch.

2.2 HOMING

Before running Hexapod_Joint_Control.mdl, Hexapod_World_Control.mdl, or any other position-based experiments, the stage of the 6 DOF Hexapod must be in the HOME position. This is defined when each base joint is at its mid-stroke position. The Hexapod_Home.mdl Simulink model, shown in Figure 2.2, is ran with QUARC to center all the linear joints at the base.

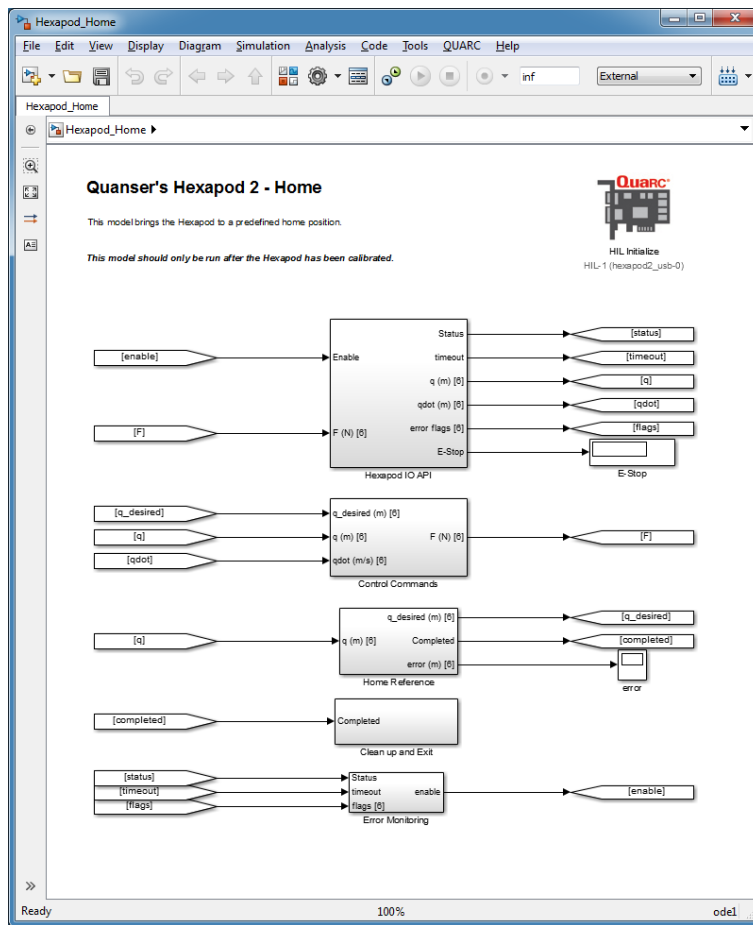


Figure 2.2: Simulink model used to home the Hexapod using QUARC.

Follow these steps to home the system:

1. Make sure the red button is in the upper position to enable the amplifier. The amplifier cannot be enabled when the button is in the lower position. Twist the button to ensure it is in the enabled position.
2. Make sure you have calibrated the Hexapod first, as described in Section 2.1.
3. Open the Simulink model called Hexapod_Home.mdl, shown in Figure 2.2.
4. Run setup_hexapod.m in MATLAB (note: script is ran automatically when Simulink model is ran).
5. Click on QUARC | Build to generate the controller.
6. Run the QUARC controller by going to QUARC | Start. The upper stage should move to the HOME position when are all the linear actuator are in the mid-stroke or center position.



Caution: PRESS DOWN on the RED BUTTON of the E-Stop switch to stop the Hexapod. This deactivates the amplifier and cuts off the DC motor power.

7. The controller stops automatically when the HOME position is reached.

2.3 JOINT-LEVEL POSITION CONTROL

This Hexapod_Joint_Control.mdl controller shown in Figure 2.3 applies position commands to the base-level linear actuators. The user can specify the amplitude and frequency of the sine wave on-the-fly and read corresponding position measurements.

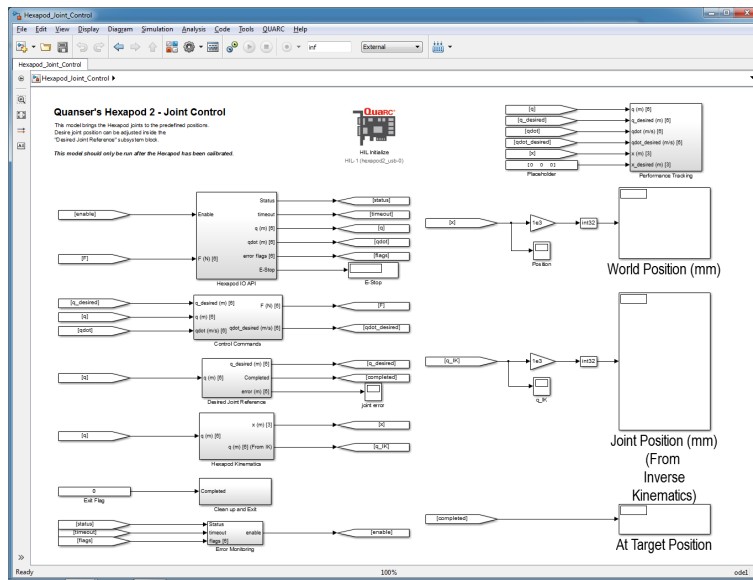


Figure 2.3: Simulink model used to command joint-level positions to the 6 DOF Hexapod using QUARC.

The joint commands can be found inside the *Desired Joint Reference* \Desired Joint Reference subsystem block shown in Figure 2.4. The home reference of 176.5 mm is applied to each joint .

The home reference command are set in the Slider Gain blocks shown in Figure 2.4a. The *Periodic Wave Commands* subsystem, shown in Figure 2.4b, has Slider Gain blocks that are, by default, all initially set to zero. Change their values in order to apply a sine wave command.

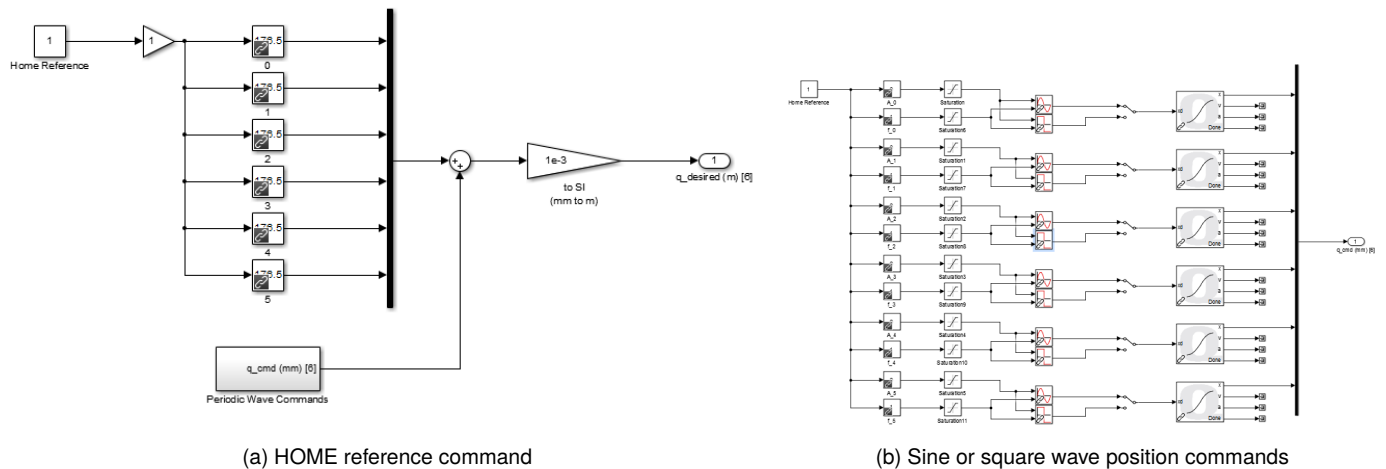


Figure 2.4: Hexapod position commands



Caution: Make sure the Hexapod is at the HOME position before running this experiment! If the joints are not at the mid-stroke or center position, then go through the homing procedure in Section 2.2.

Follow these steps to run the controller:

1. Make sure the Hexapod is at the HOME position, as described in Section 2.2.
2. Make sure the red button is in the upper position to enable the amplifier. The amplifier cannot be enabled when the button is in the lower position. Twist the button to ensure it is in the enabled position.

3. Open the Simulink model called Hexapod_Joint_Control.mdl, shown in Figure 2.3.
4. Run setup_hexapod.m in MATLAB (note: script is ran automatically when Simulink model is ran).
5. Click on QUARC | Build to generate the controller.
6. Run the QUARC controller by going to QUARC | Start.
7. Change the values of the Slider Gain blocks in in the Figure 2.4b gradually to begin commanding a sine wave to the linear actuators.



Caution: PRESS DOWN on the RED BUTTON of the E-Stop switch to stop the Hexapod. This deactivates the amplifier and cuts off the DC motor power.

8. Open the *Joint Position Tracking* and *Joint Speed Tracking* scopes, located in the *Performance Tracking* subsystem, to examine the position or speed response of the Hexapod. The yellow trace is the measured position/speed and the purple plot is the desired position/speed. To change which joint position/speed is being displayed, set the *Joint Index* block between 1 and 6 accordingly. See Section 2.8.5 for more information.
9. Before stopping the controller, it is recommended to set position commands back to the home reference. That is, set the Slider Gain blocks in the *Periodic Wave Commands* subsystem in Figure 2.4b all to 0 and the Constant block in Figure 2.4a to 140 mm to bring the stage back to HOME position and ready for the next experimental run.
Note: The Constant and Slider Gain block values for the setpoint commands are automatically reset to the home reference *automatically* when the controller is stopped. However, because this is performed after the controller is stopped, it does not physically move the platform back to HOME. This must be done manually as the controller is still running.
10. Click on Stop button in the Simulink diagram tool bar to stop running the controller.
11. Shut off the Hexapod power switch if no more experiments will be conducted.

Note: As explained in the Hexapod User Manual [2], if a limit switch is triggered the amplifier automatically stops accepting any motor/joint commands in that direction. However, position commands in the opposite direction are still accepted (e.g., joint can be moved back to home).

The forward and inverse kinematics are included in Hexapod_Joint_Control.mdl to provide a comparison between the measured joint positions, and those obtained using the derived kinematics. The *World Position* display block can be used to monitor this world coordinate position. The *Inverse Kinematics* block is used to find the six joint positions given the measured world coordinate position of the end-effector (i.e., the output from the *Forward Kinematics* block). This joint space position vector can be monitored in the *Hexapod Joints (From Inverse Kin)* display and compared to the actual joint positions shown in the *Hexapod Joints* display. See Section 2.8.4 for more information about the kinematic blocks.

2.4 WORLD-SPACE POSITION CONTROL

This Hexapod_World_Control.mdl controller shown in Figure 2.5 controls the position of the upper stage, i.e., the end-effector. The amplitude and frequency of the sine or square wave can be adjusted for each of the end-effector's axes: X, Y, Z, Pitch, Roll, and Yaw.

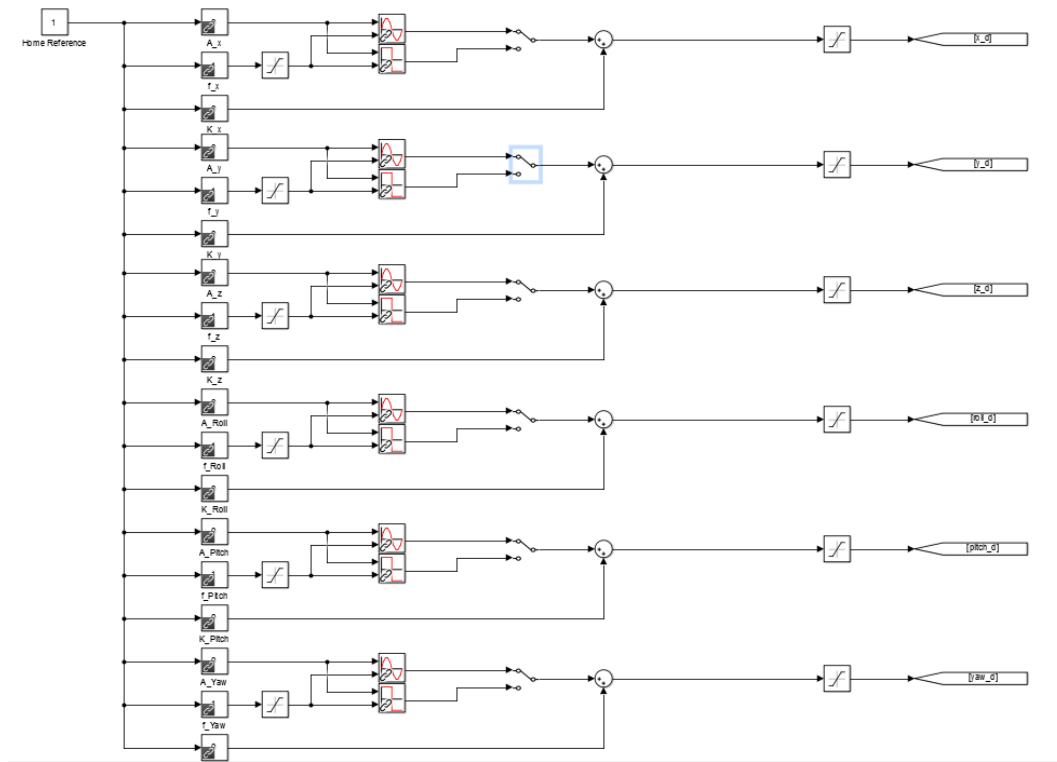


Figure 2.6: Hexapod world coordinate position commands



Caution: Make sure the Hexapod is at the HOME position before running this experiment! If the joints are not at the mid-stroke or center position, then go through the homing procedure in Section 2.2.

Follow these steps to run the controller:

1. Make sure you have home the Hexapod first, as described in Section 2.2.
2. Make sure the red button is in the upper position to enable the amplifier. The amplifier cannot be enabled when the button is in the lower position. Twist the button to ensure it is in the enabled position.
3. Open the Simulink model called Hexapod_World_Control.mdl, shown in Figure 2.5.
4. Run setup_hexapod.m in MATLAB (note: script is ran automatically when Simulink model is ran).
5. Click on QUARC | Build to generate the controller.
6. Run the QUARC controller by going to QUARC | Start.
7. Change the values of the Slider Gain blocks in the *Desired Position Reference* block, shown in Figure 2.6 subsystem, gradually to begin commanding a position command to the top stage.



Caution: PRESS DOWN on the RED BUTTON of the E-Stop switch to stop the Hexapod. This deactivates the amplifier and cuts off the DC motor power.

8. Open the *World Position Tracking* scopes in the *Performance Tracking* subsystem to examine the position response of the Hexapod end-effector. The yellow trace is the measured world coordinate position and the purple plot is the desired position. To change which axes is being displayed, set the *Position Index* block between 1 and 6 accordingly. See Section 2.8.5 for more information.
9. Before stopping the controller, it is recommended to set the sine/square wave amplitudes back to 0 to HOME the Hexapod so it is ready for the next experimental run.

Note: The Slider Gain values are actually reset to the home reference automatically when the controller is stopped. The signal amplitude gains are all reset to 0 and the frequency gain blocks are set back to 1 Hz. However, because this is performed after the controller is stopped, it does not physically move the platform back to HOME. This must be done manually as the controller is still running.

10. Click on Stop button in the Simulink diagram tool bar to stop running the controller.
11. Shut off the Hexapod power switch if no more experiments will be conducted.

Note: As explained in the Hexapod User Manual [2], if a limit switch is triggered the amplifier automatically stops accepting any motor/joint commands in that direction. However, position commands in the opposite direction are still accepted (e.g., joint can be moved back to home).

The measured end-effector position, from the *Forward Kinematics* block is shown in the *World Position (mm)* display block. The measured joint positions are shown in the *Joint Position* display.

2.5 BRAKE BYPASS

The Hexapod_Brake_Bypass.mdl model, shown in Figure 2.7 allows the joints to be moved manually. The amplifier drives are enabled and the brake system is disabled when the Digital Output Channels #12 and #13 are set to 0 and 1, respectively. The amplifiers are disabled when the DO #12 and #13 are both set to 1. The joints can be moved manually as long as the amplifiers are enabled (and no internal amplifier faults, such as over-current or over-temperature are triggered).

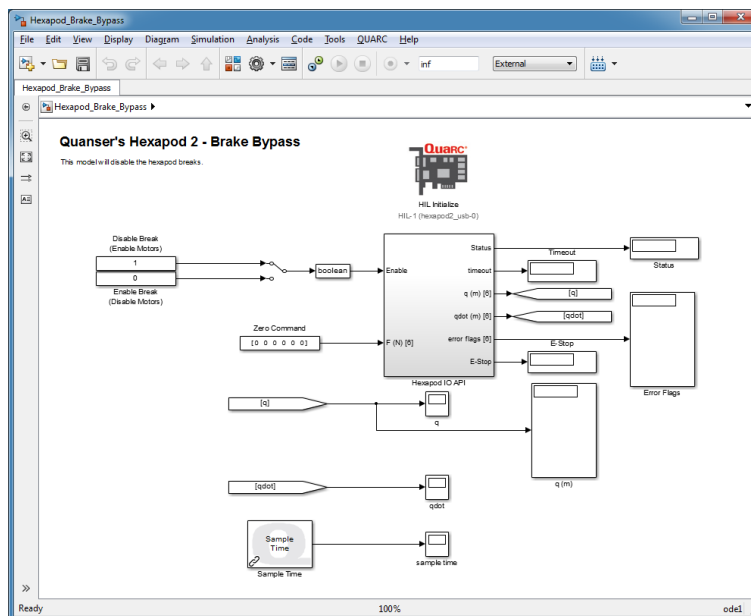


Figure 2.7: Simulink model used to disengage brakes on the Hexapod using QUARC

Follow these steps to disengage the brake system:

1. Make sure the red button is in the upper position to enable the amplifier. The amplifier cannot be enabled when the button is in the lower position. Twist the button to ensure it is in the enabled position.
2. Open the Simulink model called Hexapod_Brake_Bypass.mdl, shown in Figure 2.7.
3. Run setup_hexapod.m in MATLAB (note: script is ran automatically when Simulink model is ran)..

- Click on QUARC | Build to generate the controller.
- Power ON the Hexapod system by placing the power switch to the ON position.
- Run the QUARC controller by going to QUARC | Start. The base joints should now be free to move manually.



Caution: PRESS DOWN on the RED BUTTON of the E-Stop switch to stop the Hexapod. This deactivates the amplifier and cuts off the DC motor power.

- Click on Stop button in the Simulink diagram tool bar to stop running the controller. This engages the deactivates the amplifier and engages the brake system.
- Shut off the Hexapod power switch if no more experiments will be conducted.

2.6 EARTHQUAKE

The Hexapod_Earthquake.mdl Simulink diagram shown in Figure 2.8 can be used to replay earthquake on the 6 DOF Hexapod system using **QUARC®**. Recorded earthquake data, i.e., data collected when an actual earthquake occurred, can be scaled down and ran on the Hexapod. Four historical earthquakes have been supplied for the user: Northridge, Kobe, El-Centro, and Mendocino. For instructions on how to download new earthquakes, see Section 3.3. The user can also specify and create a pre-defined compound sine wave trajectory.

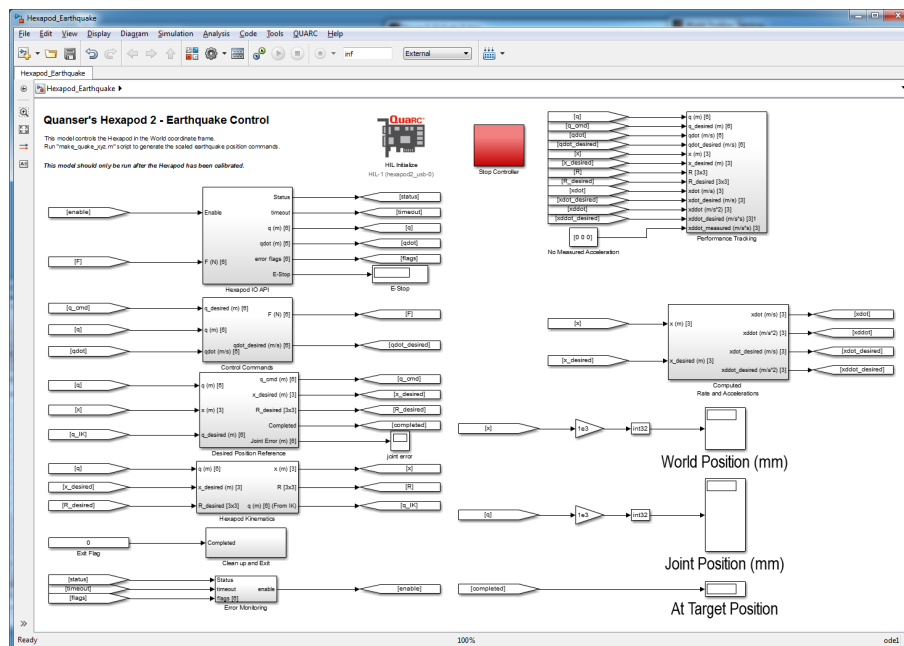


Figure 2.8: Simulink model used to command earthquake to the 6 DOF Hexapod using QUARC.



Caution: Make sure the Hexapod is at the HOME position before running this experiment! If the joints are not at the mid-stroke or center position, then go through the homing procedure in Section 2.2.

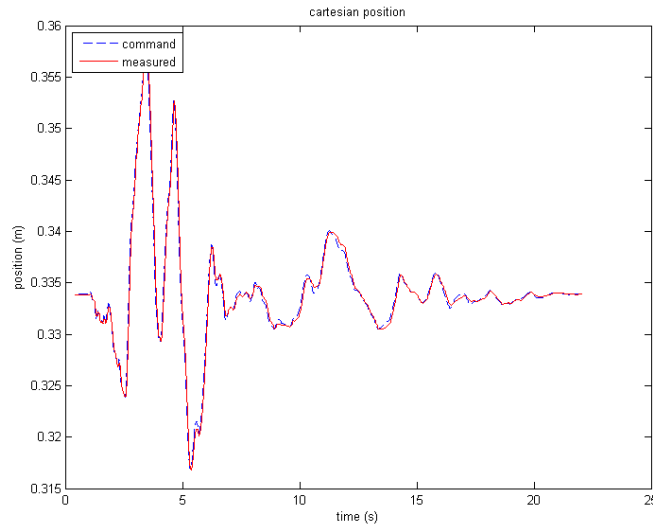
- Make sure the red button is in the upper position to enable the amplifier. The amplifier cannot be enabled when the button is in the lower position. Twist the button to ensure it is in the enabled position.
- Open the Simulink model called Hexapod_Earthquake.mdl, shown in Figure 2.8.
- Run setup_hexapod.m in MATLAB. **Note:** script is NOT ran automatically like other models - this must be ran manually.

4. The command position (and acceleration) must first be loaded into the MATLAB environment. **Select your earthquake record and run the `make_quake_xyz.m` script** to load the scaled data. For more information about using the `make_quake_xyz.m` script, see Section 4.3.
5. Click on QUARC | Build in Hexapod_Earthquake.mdl to generate the controller.
6. Run the QUARC controller by going to QUARC | Start. The top stage should begin tracking the loaded earthquake (or sine wave).

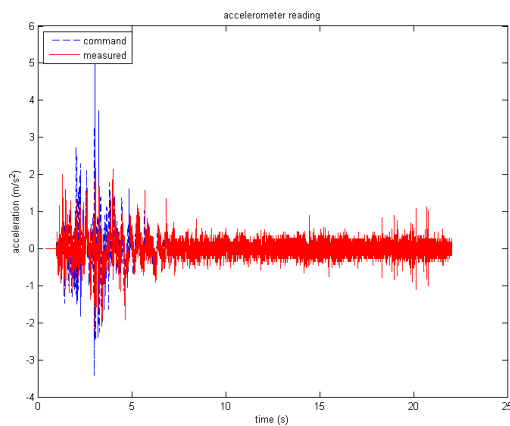


Caution: PRESS DOWN on the RED BUTTON of the E-Stop switch to stop the Hexapod. This deactivates the amplifier and cuts off the DC motor power.

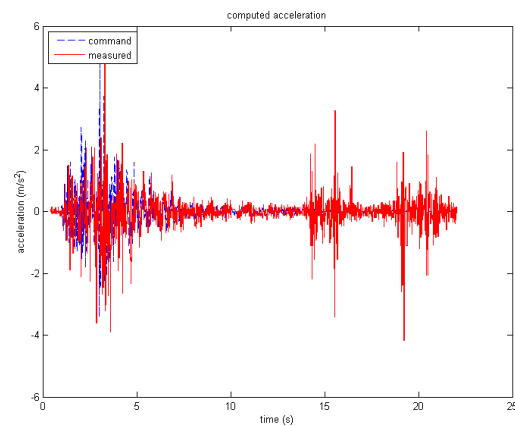
7. Typical position and acceleration responses when running the Northridge earthquake is shown in Figure 2.9 for the x-axis.



(a) Position along X-Axis



(b) Measured Acceleration along X-Axis



(c) Computed Acceleration along X-Axis

Figure 2.9: Hexapod stage position and acceleration when running Northridge earthquake.

8. The controller stops by itself when the duration of the earthquake (or sine) is reached. The controller can be stopped, however, by clicking on the Stop button in the Simulink diagram tool bar at any time.
9. Shut off the Hexapod power switch if no more experiments will be conducted.
10. To plot the FFT of the position or acceleration, run `fft_eval_pos.m` or `fft_eval_acc.m`. See Section 2.7 for instructions.

Note: As explained in the Hexapod User Manual [2], if a limit switch is triggered the amplifier automatically stops accepting any motor/joint commands in that direction. However, position commands in the opposite direction are still accepted (e.g., joint can be moved back to home).

2.7 FFT ANALYSIS

This section shows how to evaluate the **power spectrum** of the commanded and measured Hexapod position and acceleration signals using the `fft_eval_pos.m` and `fft_eval_acc.m` scripts. Given a time-based signal $g(t)$, the power spectrum shown on the plot generated is given by $|G(\omega)|$, where $G(\omega)$ is the Fast-Fourier Transform (FFT). The FFT, $G(\omega)$, is computed using the **MATLAB**® `fft` command. See the `power_spectrum.m` script for further details.

1. Run an earthquake or sine wave using the Hexapod_Earthquake.mdl QUARC controller, as described in Section 2.6. After Hexapod_Earthquake.mdl is ran, the commanded and measured position and acceleration data are automatically saved to the Matlab data files `data_xyz_pos.mat` and `data_xyz_acc.mat`. Only one axis is saved to these variables - either X, Y, or Z - depending on the selection in the *Position Index* block found in the *Performance Tracking* subsystem, as shown in Section 2.8.5.
2. To produce the $|X(\omega)|$ position power spectrum similar to Figure 2.10, run `fft_eval_pos.m`. The desired or commanded position is the blue trace and the measured position (from the encoder) is the red plot line.

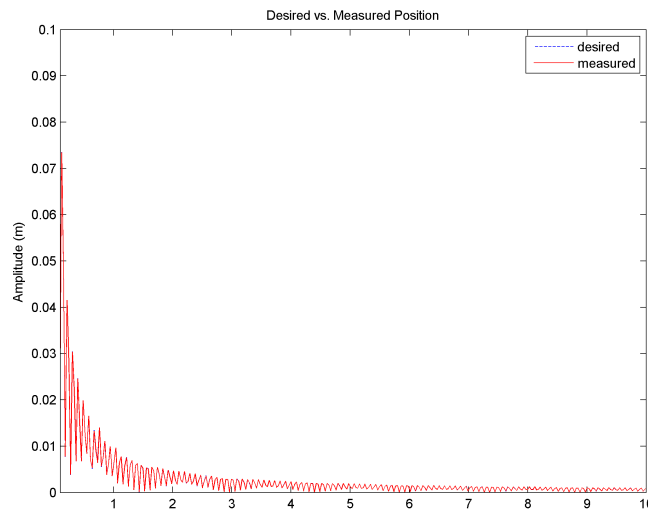


Figure 2.10: FFT of desired and measured position along x-axis after running Northridge earthquake

3. To produce the acceleration power spectrum, $A(\omega)$, similar to Figure 2.11, run `fft_eval_acc.m`. The blue and red traces are the desired and measured accelerations, respectively.

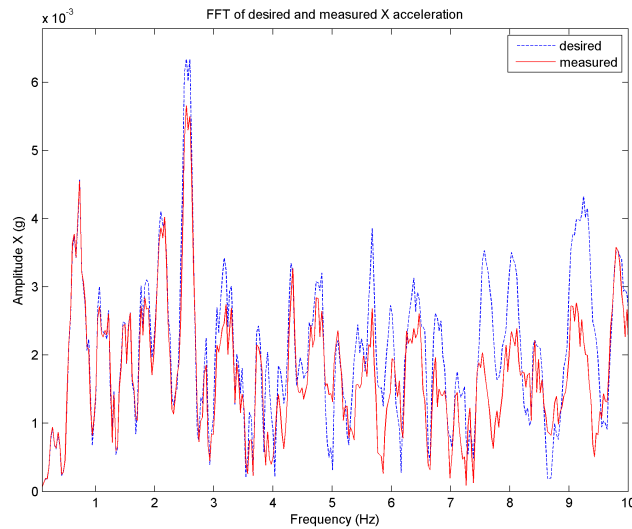


Figure 2.11: FFT of desired and measured acceleration along x-axis after running Northridge earthquake

4. In both the *fft_eval_pos.m* and *fft_eval_acc.m* scripts, you can adjust the frequency range by adjusting the *f_min* and *f_max* variables:

```
% min/max frequencies for plotting (Hz)
f_min = 1;
f_max = 10;
```

2.8 HEXAPOD CONTROLLERS

This section briefly describes some of the blocks used the Simulink models that are supplied with Hexapod, to be used with QUARC. Some of the blocks designed specifically for the Hexapod are found in the *Hexapod_Library* Simulink library, depicted in Figure 2.12.

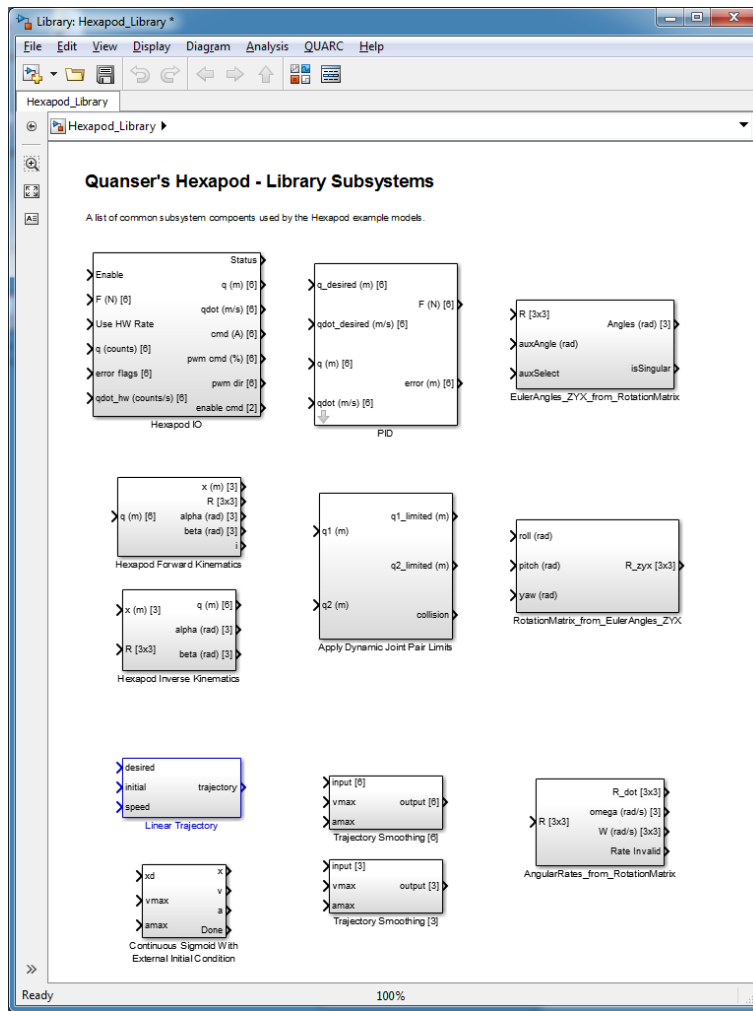


Figure 2.12: Hexapod library used in QUARC controllers supplied

2.8.1 Position Reference

As shown in Figure 2.13, the Hexapod positions commands are smoothed in the *Home Reference \Trajectory Smoothing* subsystem. The *Linear Trajectory* block takes in the desired position of each joint, the initial joint positions as well as the maximum allowed speed for each joint. The block then outputs a calculated trajectory for each joint to follow. This trajectory is the position command to each joint.

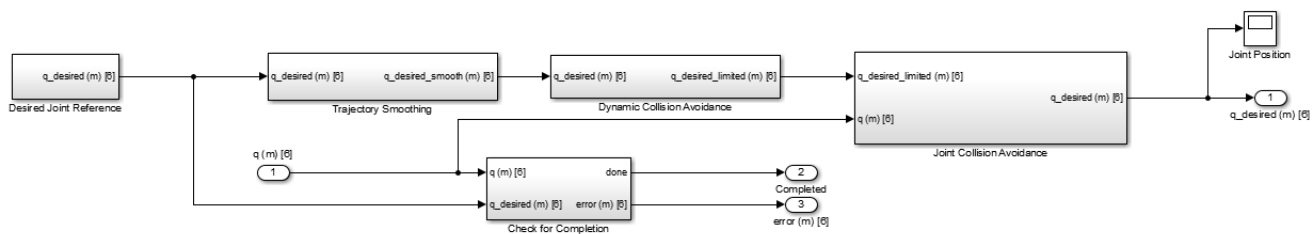


Figure 2.13: Hexapod position commands are smoothed and checked for collisions

The position references go through the collision avoidance logic to prevent adjacent joints from colliding with each.

This is done by keeping a minimum distance between them. The separation distance can be set in `setup_hexapod.m` in the `min_separation` variable. Note that this is defined as the distance between the *joint center to joint center* - not the *side to side* distance. Furthermore, each joint is limited to maximum and minimum position using a Simulink Saturation block. Additional checks are included to detect and avoid potential collisions caused by the desired position references between the links and the mounting brackets at each joint location. The desired position references will be passed through the Joint Collision Avoidance subsystem if they do not result in a collision condition. Otherwise, the previous desired joint commands will be held until a set of collision-free desired position references are received (and be passed through).

2.8.2 Hardware Interface

The QUARC blocks used to interface to the Hexapod amplifier, linear actuators, and encoders is done, at a higher-level, in the *Hexapod IO API* subsystem shown in Figure 2.14.

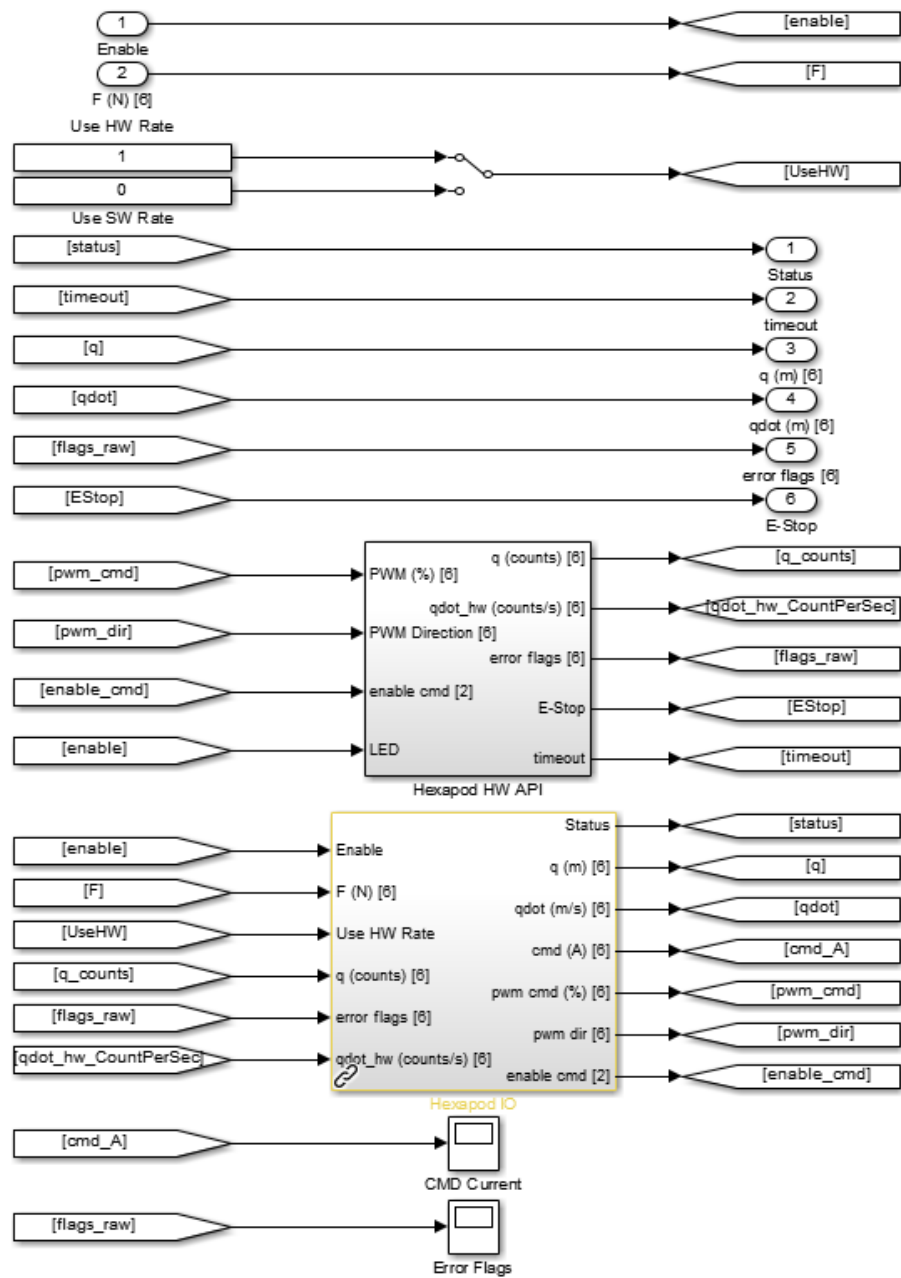


Figure 2.14: Hexapod IO API subsystem

The *Hexapod HW API* subsystem, depicted in Figure 2.15, includes the hardware blocks that interface directly to the hardware actuator and sensors. The HIL Read Timebase block reads the six encoders, status of all six amplifier drives (e.g., if a fault occurs), and the encoder velocities. Using the Timebase block forces the controller to use the clock from the data acquisition device for more deterministic sampling when running the controller (as opposed to the PC system timer). The HIL Write enables the amplifiers and applies the PWM amplitude and direction commands to the amplifier drives. Finally, HIL Watchdog monitors the controller performance and stops it if a timeout occurs.

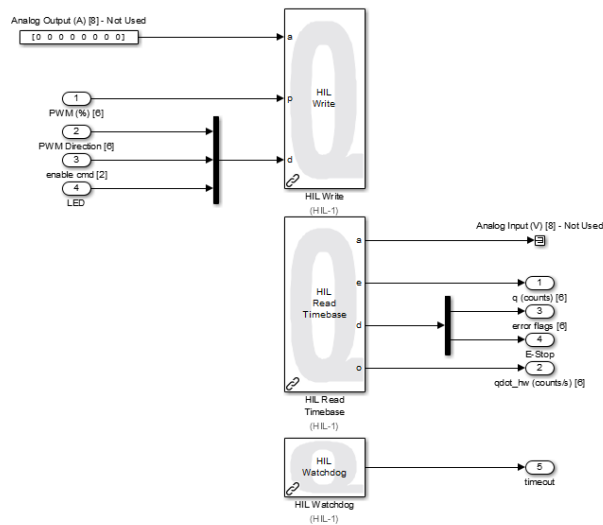


Figure 2.15: QUARC blocks used to interface to Hexapod actuators and sensors

The *Hexapod IO API* \Hexapod IO subsystem, shown in Figure 2.16, interfaces Hexapod position controller to the hardware I/O explained above. For instance, it converts the encoder counts to joint linear position (i.e., multiplied by the encoder resolution and motor shaft circumference constant). Based on the force computed by the Hexapod controller to reach the desired position, denoted as F , this subsystem computes the current required for the amplifier drives (see Section 3.2 for the force to current conversion).

The current is converted to the PWM signals required. The PWM magnitude is applied to the QUARC *HIL Write* block on PWM Channels #0 to #5 and the PWM direction is applied to DO Channel #0 to #5. Note that current commands are saturated according the configured amplifier limits to help prevent the motors or amplifier drives from over-heating (see Hexapod User Manual [2] for current limit specifications).

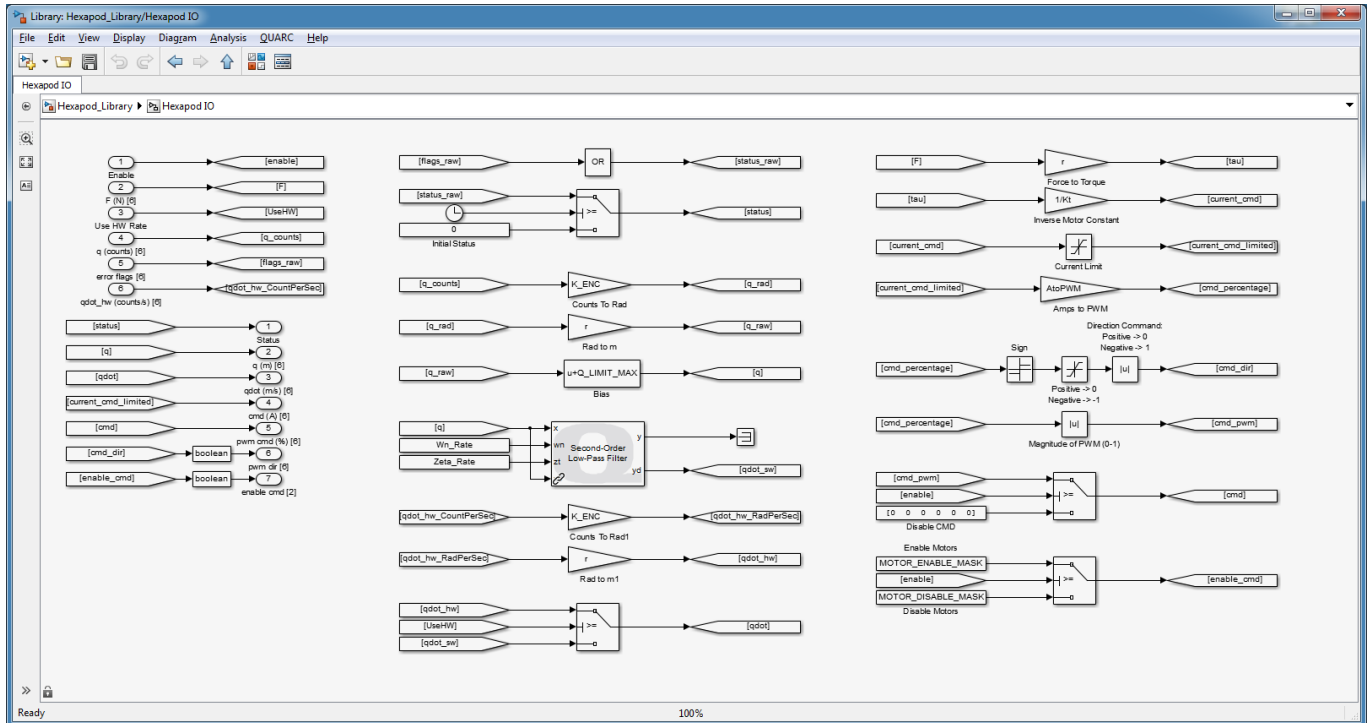


Figure 2.16: Hexapod IO subsystem

2.8.3 Position Control

The Hexapod implements the PID-based controller described in Section 3.2 in the *PID* subsystem block, shown in Figure 2.12. The control force, i.e., effort required to attain the desired position, is converted to current for the amplifier in the *Hexapod IO* subsystem.

The joint velocities used in the PID controller can be generated through hardware or software. Use the manual switch in the *Hexapod IO API* subsystem shown in Figure 2.14 to select between the two sources for the joint velocity:

1. **Hardware-based velocity** obtained directly from the DAQ board using HIL Read Timebase block (see the Hexapod User Manual for channel details). **This is the default.**
2. **Software-based velocity** obtained by passing the joint values through a second-order high-pass filter.

2.8.4 Kinematics

The forward and inverse kinematic block are available in the *Hexapod Kinematics* block shown in Figure 2.17. Forward kinematics are used to find the world-based end-effector position (X, Y, Z, Roll, Pitch, Yaw) from the six joint positions, q , and the leg lengths. It also the rotation matrix, R .

Inverse kinematics calculates the six joint positions from the world coordinate of the end-effector position. The *Inverse Kinematics* block is used in the Hexapod_World_Control.mdl model to find the joint-level commands required to attain the desired end-effector position. But it is also used in Hexapod_Joint_Control.mdl to compare it again the actual, measure joint positions.

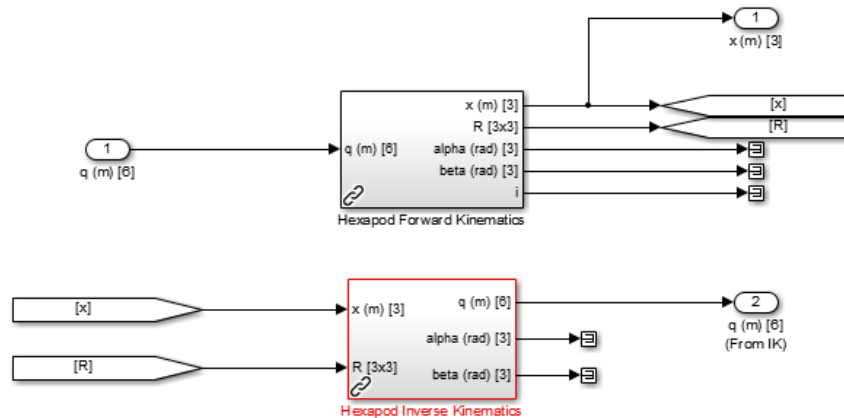


Figure 2.17: Forward and inverse kinematic blocks for the Hexapod

2.8.5 Performance Tracking

The *Performance Tracking* subsystem, depicted in Figure 2.18, includes various scopes to monitor the position and speed of the Hexapod.

- **Joint Position Tracking:** The yellow trace displayed in this plot is the measured position of the joint and the purple plot line is the commanded or desired position. The joint to be displayed on the scope can be selected using the *Joint Index* block (values between 0 and 5).
- **Joint Speed Tracking:** Similarly as *Joint Position Tracking* scope, but displays the desired and measured joint-level velocities instead.

- **World Position Routing:** The yellow trace displayed in this plot is the measured position of the top stage and the purple plot line is the commanded or desired position. The axis to be displayed on the scope - X, Y, Z, Roll, Pitch, or Yaw - can be selected using the *Position Index* block (values between 0 and 5). The measured position commands are generated by the forward kinematics.

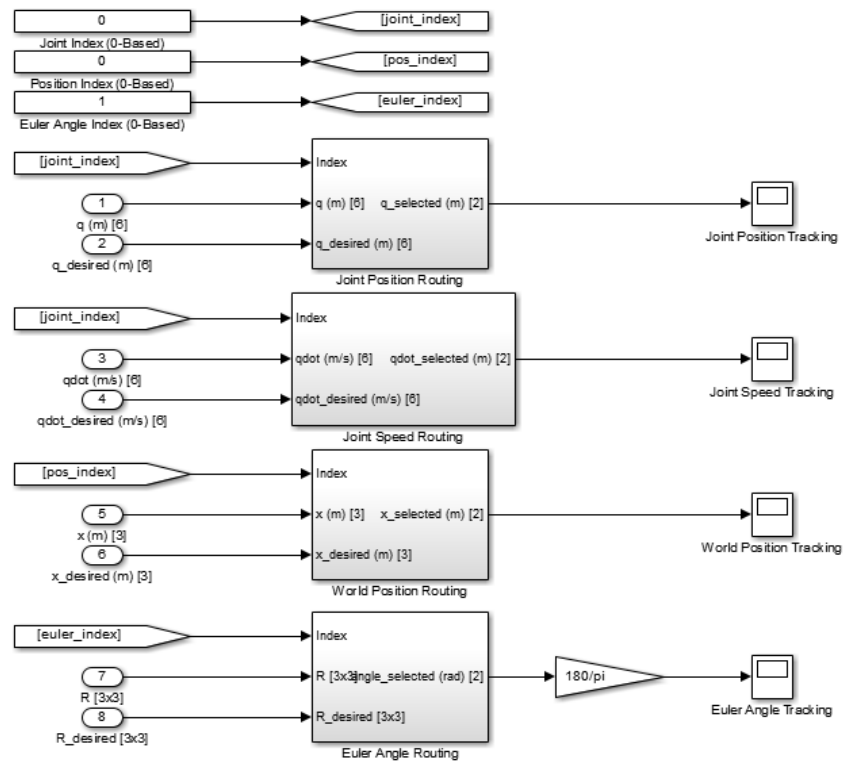


Figure 2.18: Performance Tracking subsystems includes scope to monitor position and speed of Hexapod

3 BACKGROUND

3.1 MODELING

The dynamics between the applied force to the linear position of each joint can be represented by the transfer function

$$q(s) = \frac{1}{M_t s^2} F(s) \quad (3.1)$$

where $q(s) = \mathcal{L}[q(t)]$ is the Laplace of the joint position $q(t)$, $F(s)$ is the Laplace of the applied linear force, and M_t is the total mass being moved by the motor (i.e., both pre-load and payload).

3.2 POSITION CONTROLLER DESIGN

The joint positions on the Hexapod system is controlled using a standard proportional-integral-derivative (PID) control. The PID control used has following structure:

$$F(t) = k_p (q_d(t) - q(t)) + k_i \int q_d(t) - q(t) dt + k_d (\dot{q}_d(t) - \dot{q}(t)) \quad (3.2)$$

where k_p is the proportional control gain, k_i is the integral gain, k_d is the derivative control gain, $q_d(t) \in \mathbb{R}^6$ is the setpoint or reference joint position (for all six joints), $q(t) \in \mathbb{R}^6$ is the measured joint position (for all six joints), and $F(t)$ is the force (i.e., control effort). The block diagram of the control is given in Figure 3.1.

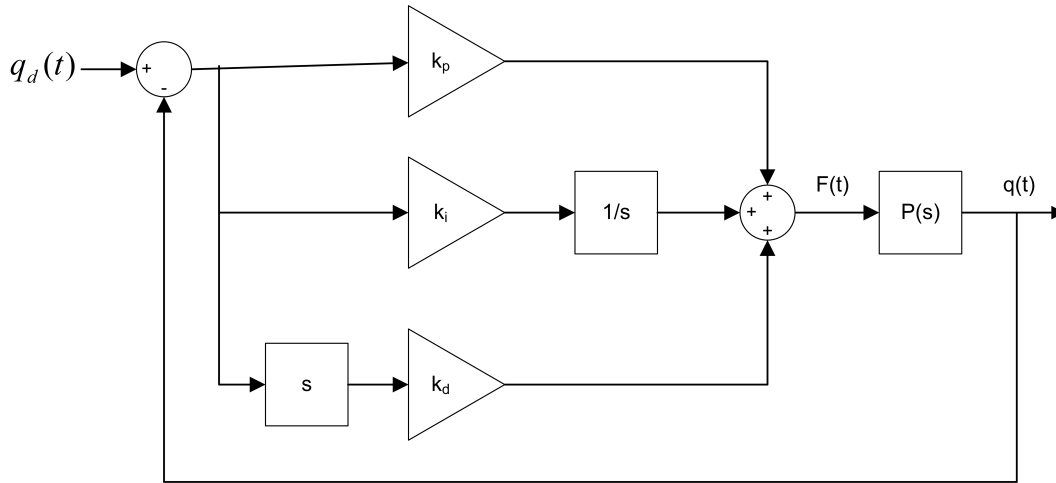


Figure 3.1: Block diagram of Hexapod position control.

In software implementation, the controller force is converted into motor current using

$$\begin{aligned} \tau_m &= \frac{F}{r} \\ I_m &= \frac{\tau_m}{k_t} \end{aligned}$$

Thus the controller force is divided by the lead-screw radius, r , to get the torque and this is multiplied by the current constant, k_t , to obtain the necessary motor current.

The default values used for the PID gains and cut-off filter are:

$$\begin{aligned}k_p &= 30000 \text{ N/m} \\k_i &= 40000 \text{ N/(m-s)} \\k_d &= 3000 \text{ N-s/m} \\\omega_n &= 1000 \text{ rad/s}\end{aligned}$$

The cut-off frequency, ω_n , is for the second-order filter that is used to compute the desired joint velocity and acceleration.



Caution: These are the default PD control gains that are used in the 6 DOF Hexapod software controller and **should NOT be tuned by the user** unless it is necessary for application-specific reasons.

3.3 DOWNLOADING NEW EARTHQUAKES

There are a variety of resources on the Internet where real earthquake data can be downloaded. This section explains how to download and use the records from the Pacific Earthquake Engineering Research Center (PEER) Ground Motion Database website at: http://peer.berkeley.edu/peer_ground_motion_database/site

On the PEER website, each earthquake has various measurement stations and each station contains recorded displacement, velocity, and acceleration data of the tremor at different directions. The default download setting is acceleration, which is required when using the supplied `make_quake_xyz.m` file. To illustrate how to download and run an earthquake, the procedure below shows how to find the Kobe tremor recorded at the HIK station:

1. Go to the PEER Ground Motion Database at: http://peer.berkeley.edu/peer_ground_motion_database/site
2. Click on the *Unscaled* button.
3. In the *Event Name* field, enter *Kobe, Japan* as illustrated in Figure 3.2.

New Unscaled Search

PEER-NGA Spectrum

Additional Search Options

Event Name

NGA Sequence Numbers

Station Name

[Show chart controls](#)

Figure 3.2: Searching for the Kobe earthquake on PEER Ground Motion Database

4. Click on *Search*.
5. As shown in Figure 3.3, the *Results* section displays all the stations that recorded the Kobe earthquake.
6. Select Result #2 to download the recorded Kobe acceleration that was measured at the HIK station.

Results

*Click on the record below to display Spectra and Time series ☐ Plot Selected

<input type="checkbox"/>	Result#	Comp.	NGA#	Pulse	TP(s)	D5-95(s)	Event	Year	Station
<input type="checkbox"/>	1	GM	1103	0 0	-- --	34.4 38.2	Kobe, Japan	1995	FUK
<input checked="" type="checkbox"/>	2	GM	1105	0 0	-- --	18.5 10.5	Kobe, Japan	1995	HIK
<input type="checkbox"/>	3	GM	1106	1 0	1.0 --	9.6 8.1	Kobe, Japan	1995	KJMA
<input type="checkbox"/>	4	GM	1107	0 0	-- --	17.6 10.4	Kobe, Japan	1995	Kakogawa
<input type="checkbox"/>	5	GM	1109	0 0	-- --	26.4 21.2	Kobe, Japan	1995	MZH
<input type="checkbox"/>	6	GM	1111	0 0	-- --	9.5 11.1	Kobe, Japan	1995	Nishi-Akashi
<input type="checkbox"/>	7	GM	1112	0 0	-- --	14.6 17.8	Kobe, Japan	1995	OKA
<input type="checkbox"/>	8	GM	1113	0 0	-- --	58.6 71.8	Kobe, Japan	1995	OSAJ
<input type="checkbox"/>	9	GM	1116	0 0	-- --	13.3 9.4	Kobe, Japan	1995	Shin-Osaka
<input type="checkbox"/>	10	GM	1117	0 0	-- --	20.2 23.7	Kobe, Japan	1995	TOT
<input type="checkbox"/>	11	GM	1119	1 0	1.4 --	5.1 3.4	Kobe, Japan	1995	Takarazuka
<input type="checkbox"/>	12	GM	1120	1 0	1.6 --	10.8 11.9	Kobe, Japan	1995	Takatori

Figure 3.3: Kobe earthquake records on PEER Ground Motion Database

- Click on the *Save Original Unscaled Time Series Records* button.
- Save the ZIP file to your PC and extract its contents to the *6 DOF Hexapod\Controllers* folder. This particular record contains the measured acceleration for three different direction: NGA_no_1105_HIK000.AT2, NGA_no_1105_HIK090.AT2, and NGA_no_1105_HIK-UP.AT2.
- To use NGA_no_1105_HIK000.AT2, go to make_quake_xyz.m and set the *input_filename* variable as follows:

```
% name of data source file:
input_filename = 'NGA_no_1105_HIK000.AT2';
```

- Then follow the directions given in Section 2.6 to run this earthquake on the 6 DOF Hexapod system.

4 SYSTEM REQUIREMENTS

Required Software

- Microsoft Visual Studio (MS VS)
- **MATLAB**[®] with **SIMULINK**[®], Real-Time Workshop, and the Control System Toolbox
- **QUARC**[®]

See the **QUARC**[®] software compatibility chart in [3] to see what versions of MS VS and **MATLAB**[®] are compatible with your version of QUARC and for what operating system.

Required Hardware

- Quanser 6 DOF Hexapod. Note: already includes integrated amplifier and data acquisition (DAQ) device.

Before Starting Lab

Before you begin this laboratory make sure:

- **QUARC**[®] is installed on your PC, as described in [4].
- 6 DOF Hexapod is connected as described in the Hexapod User Manual [2].

4.1 OVERVIEW OF FILES

File Name	Description
Hexapod User Manual.pdf	This manual describes the hardware of the Hexapod system and explains how to setup and wire the system for the experiments.
Hexapod Laboratory Guide.pdf	This document demonstrates how to run the QUARC controllers, gives some background on the PID-based control design, and describes the Simulink diagrams supplied.
setup_hexapod.m	The main Matlab script that sets the Hexapod motor, sensor, and configuration-dependent parameters. Run this file only to setup the laboratory.
Hexapod_Brake_Bypass.mdl	Disengages the brake system and enables the amplifier to allow the joints to be moved manually.
Hexapod_Calibrate.mdl	Moves each joint to the limit switch and resets the encoders. Run calibration EVERY TIME the Hexapod is powered ON.
Hexapod_Home.mdl	Moves each of the joints to the mid-stroke position. Run this AFTER calibration and BEFORE any of the position-based controllers (e.g., Hexapod_Joint_Control.mdl or Hexapod_World_Control.mdl).
Hexapod_Joint_Control.mdl	Controls the position of the linear actuator joints at the base of the Hexapod system. User can change the amplitude and frequency of the sine wave position command.
Hexapod_World_Control.mdl	Controls the X, Y, and Z positions as well as the Roll, Pitch, and Yaw angles of the Hexapod top stage or end-effector. User can select between applying a sine or square wave for each axis and vary the amplitude and frequency of each.

Table 4.1: Files supplied with the 6 DOF Hexapod

File Name	Description
q_scale.p	Produces a scaled position trajectory given desired acceleration data. When ran on the Hexapod, the measured acceleration matches the given desired acceleration.
construct_quake_trajectory.m	Constructs a time-based array from a matrix of data and the sampling time.
init_earthquake_data.m	Re-formats raw earthquake data file.
NGA_no_1105_HIK000.AT2	Sample Kobe earthquake acceleration data file (direction 000).
NGA_no_1105_HIK090.AT2	Sample Kobe earthquake acceleration data file (direction 090).
NGA_no_1105_HIK-UP.AT2	Sample Kobe earthquake acceleration data file (direction UP).
NGA_no_1086_SYL090.AT2	Sample Northridge earthquake acceleration data file (direction 090).
NGA_no_1086_SYL360.AT2	Sample Northridge earthquake acceleration data file (direction 360).
NGA_no_1086_SYL-UP.AT2	Sample Northridge earthquake acceleration data file (direction UP).
NGA_no_180_H-E05140.AT2	Sample El-Centro earthquake acceleration data file (direction 140).
NGA_no_180_H-E05230.AT2	Sample El-Centro earthquake acceleration data file (direction 230).
NGA_no_180_H-E05-UP.AT2	Sample El-Centro earthquake acceleration data file (direction UP).
NGA_no_825_CPM000.AT2	Sample Mendocino earthquake acceleration data file (direction 000).
NGA_no_825_CPM090.AT2	Sample Mendocino earthquake acceleration data file (direction 090).
NGA_no_825_CPM-UP.AT2	Sample Mendocino earthquake acceleration data file (direction UP).

Table 4.2: Additional files supplied with the 6 DOF Hexapod

4.2 SETUP SCRIPT: SETUP_HEXAPOD.M

The setup_hexapod.m script loads MATLAB parameters that are used in the supplied Simulink model for QUARC. The setup_hexapod.m script should be ran before any of the QUARC controllers. However, it will be ran automatically by each supplied Simulink model using the *Model callbacks* feature.

Follow these steps:

1. Load the **MATLAB®** software.
2. In the *Current Directory* window, go to the 6 DOF Hexapod files on your PC (copied from the 6 DOF Hexapod CD).
3. Open the setup_hexapod.m script shown below:

```
%
%%
%%% Quanser's Hexapod Parameters Definition %%%
%%
%

%
% CAUTION: DO NOT EDIT THIS FILE.
%           Otherwise, Hexapod might not function properly.
%

clear all;

%%% Mechanical Parameters %%%

% Coupling Circumference
Circ = 10e-3;

% Radius
r = Circ/2/pi;
% Encoder Resolution: 2500 lines per rev at Quadrature
K_ENC = 2*pi/10000;

% Calibration Limit Position (m)
Q_LIMIT_MAX = 322.5e-3;
Q_LIMIT_MIN = 30.5e-3;

% Joint Home Position (m)
Q_HOME = 176.5e-3;

% Second Order Filter Coefficients - Software Encoder Rate
Wn_Rate = 1e3;
Zeta_Rate = 1;

% Dynamic Collision Avoidance
q_upper      = Q_LIMIT_MAX;
q_lower      = Q_LIMIT_MIN;
min_separation = 155e-3;

% Joint Collision Avoidance (deg)
```

```

RE_LIMIT_ROLL          = 30;
RI_LIMIT_ROLL          = 80;
R_LIMIT_PITCH_LOWER    = 35;
R_LIMIT_PITCH_UPPER    = 150;
BASE_LEAN_LIMIT        = 40;

%%% Motor Parameters %%%

% Motor Torque Constant
Kt = 19.3*0.0254*0.2780139;

% Max Current - Software (A)
CURRENT_MAX = 5;

%%% Electrical Parameters %%%

% Current to PWM Mapping (%/A)
AtoPWM = 1/5;

% Encoder / Motor Channels
Channels = [0:5];

%%% Controls Parameters %%%

% Initial delay in receiving status signals (s)
DELAY_INITIAL = 0.05;

% Logical signal masks for enable/disable motors
MOTOR_ENABLE_MASK  = [0 1];
MOTOR_DISABLE_MASK = [1 1];

% Maximum setpoint velocity and acceleration

% Max Joint Speed - Software (m/s)
QDOT_MAX          = 0.65;
% Max Joint Acceleration - Software (m/s^2)
QDDOT_MAX         = 6.5;

% Max Cartesian Linear Speed - Software (m/2)
XDOT_MAX          = 0.35;
% Max Cartesian Linear Acceleration - Software (m/s^2)
XDDOT_MAX         = 3.5;

% Max Rotational Speed - Software (rad/s)
THETADOT_MAX      = 60*pi/180;
% Max Rotational Acceleration - Software (rad/s^2)
THETADDOT_MAX     = 10*60*pi/180;

```

4. Run the `setup_hexapod.m` script (click on Debug | Run in the Editor menu bar or on the *Run* icon in the Editor

tool bar) to setup the Matlab workspace.



Caution: Most of the settings in `setup.m` should not be changed. This script has been configured for the 6 DOF Hexapod system that was shipped and therefore most the parameters should not be changed.

4.3 CREATING AN EARTHQUAKE: MAKE_QUAKE_XYZ.M

The `make_quake_xyz.m` script builds a trajectory that can be used in the `Hexapod_Earthquake.mdl` Simulink diagram and ran on the shake table. The resulting trajectory created is the setpoint or command position that is to be tracked by the stage in order to achieve the same accelerations as the recorded earthquake.

As listed in Table 4.2, the Northridge, Kobe, El-Centro, and Cape Mendocino raw earthquake acceleration files are already supplied. Additional earthquake data files can be downloaded from the Internet from locations such as the PEER Ground Motion website, as explained in Section 3.3.

Follow these steps to run the `make_quake_xyz.m` file:

1. Load **MATLAB®**.
2. In the *Current Directory* window, go to the 6 DOF Hexapod files on your PC (copied from the 6 DOF Hexapod CD).
3. Open the `make_quake_xyz.m` script.
4. Set the `input_filename_x` and `input_filename_y` variables in the script to the name of the earthquake file that is to be replayed on the shake table. As shown below, set the variable to `NGA_no_1806_SYL090.AT2` and `NGA_no_1806_SYL360.AT2` for the Northridge earthquake. The `x_max` parameter determines the maximum position of the scaled setpoint trajectory.

```
%% INPUT
% name of data source file:
% Northridge
input_filename_x = 'NGA_no_1086_SYL090.AT2';
input_filename_y = 'NGA_no_1086_SYL360.AT2';
input_filename_z = 'NGA_no_1086_SYL-UP.AT2';
% Kobe
% input_filename_x = 'NGA_no_1105_HIK000.AT2';
% input_filename_y = 'NGA_no_1105_HIK090.AT2';
% input_filename_z = 'NGA_no_1105_HIK-UP.AT2';
% Cape Mendocino
% input_filename_x = 'NGA_no_825_CPM000.AT2';
% input_filename_y = 'NGA_no_825_CPM090.AT2';
% input_filename_z = 'NGA_no_825_CPM-UP.AT2';
% El Centro
% input_filename_x = 'NGA_no_180_H-E05140.AT2';
% input_filename_y = 'NGA_no_180_H-E05230.AT2';
% input_filename_z = 'NGA_no_180_H-E05-UP.AT2';
%
% Maximum scaled position (cm). NOTE: Do not exceed stroke limit. It is
% recommend you keep this value below 2.5 cm for most trajectories.
x_max = 2.5;
%
```



Caution: Do not set `x_max` greater to value greater then the maximum stroke of the table! In general, keep the variable under 2.5 cm for most trajectories.

- Run the `make_quake_xyz.m` script (click on Debug | Run in the Editor menu bar or on the *Run* icon in the Editor tool bar) to generate the earthquake wave. This will generate a plot similar to Figure 4.1. The desired acceleration and recorded earthquake acceleration is shown in the top plot (in gravitational units) and the scaled position setpoint in the bottom plot (in centimeters).

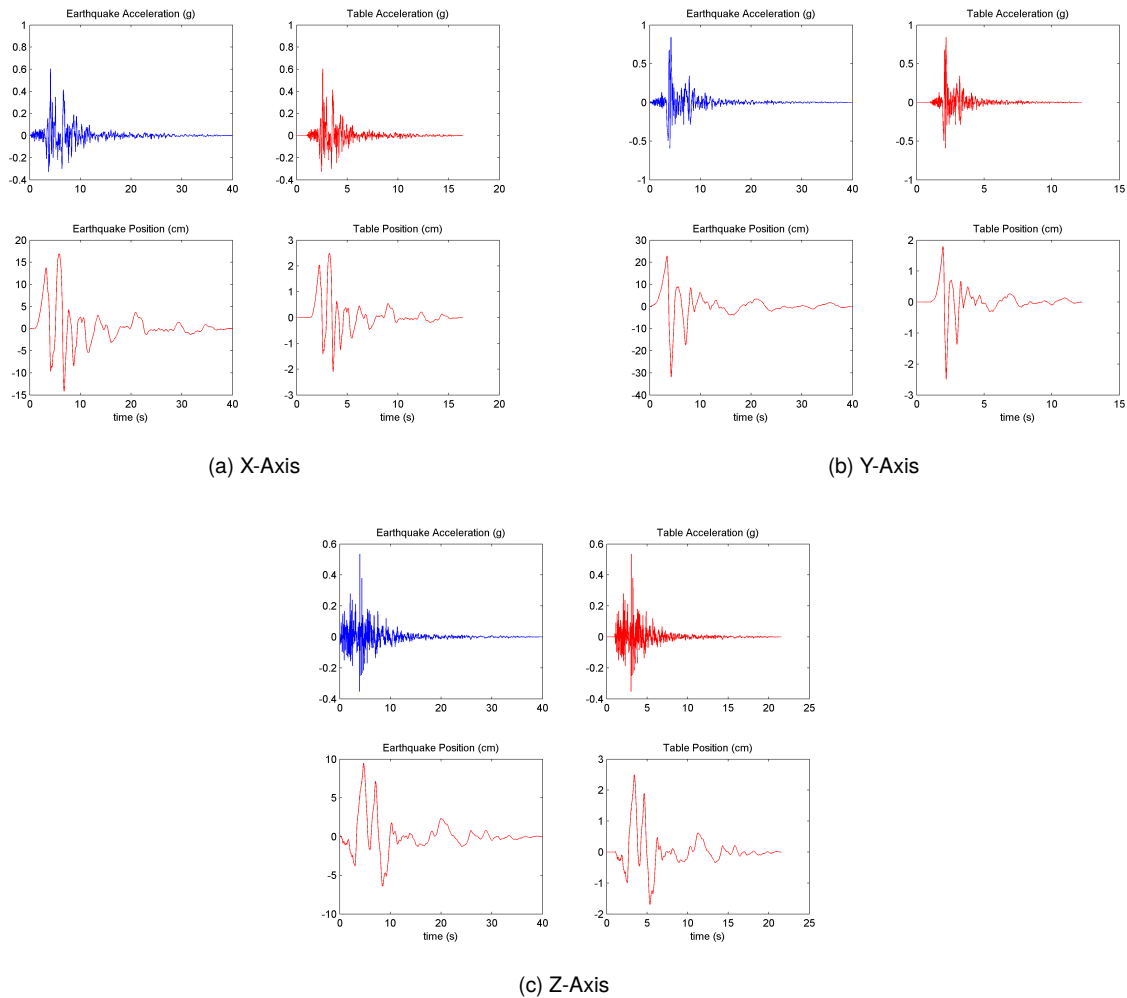


Figure 4.1: Northridge earthquake plot created by `make_quake_xyz.m` script

Note: The desired acceleration, which is computed from the scaled position setpoint, is the same as the actual recorded acceleration of the earthquake, a .

- After running `make_quake_xyz.m` with the files `NGA_no_1806_SYL090.AT2`, `NGA_no_1806_SYL360.AT2`, and `NGA_no_1806_SYL-UP.AT2` at $x_{\text{max}} = 2.5$ cm, the following output will be displayed in the Matlab Command Window *for the x-axis*:

```
*** Quanser Consulting Inc.
*** Use of this Software is under license from Quanser Consulting Inc.
*** Any results derived from this use should be duly acknowledged by the statement:
*** Acceleration and position scaling performed using software licensed from Quanser Consulting Inc.
```

```
[Tc,Xc,Ac,Xu,Au] = q_scale(t,a,xmax)
t = array of time at equal sampling intervals (s)
a = array of acceleration record that matches "t" above (g)
xmax = maximum amplitude of motion (cm)
Do NOT exceed the limits of the table with xmax!
```

```

Tc = Scaled command time array (s)
Xc = Scaled table position command array (cm)
Ac = Acceleration array, found by differentiating Xc twice (g)
Tu = Earthquake time array (s)
Xu = Computed earthquake displacement (not scaled) (cm)
Au = Earthquake acceleration array (g)

Original time step:    0.02000
Step 1 of 3: Get displacements
Step 2 of 3: Scale records
Ratio of table displacement to ground displacement: 0.147731

Step 3 of 3: Scaling time
Time step after scaling = 0.007687

*** Done ***
Displacement scaled from original movement of 16.92 cm to 2.50 cm
Time scaled from original duration of 39.98 s to 15.37 s
Record size = 2130 samples

```

The Northridge earthquake had a maximum displacement of 16.92 cm and this was scaled down to 2.5 cm (as set by `x_max`). In order to achieve the same acceleration, the time of the generated trajectory is compressed from 39.98 to 15.37 seconds.

7. Once the script is ran, the `make_quake_xyz.m` Simulink Model can be used to replay the earthquake on the 6 DOF Hexapod. See Section 2.6 for the procedure to run the tremor on the table.

Files used for Earthquake Scaling

Three functions are called in `make_quake_xyz.m` to create the scaled position command from the recorded earthquake data:

1. `init_earthquake_data.m`: this script extracts the sampling time information of the recorded earthquake, dt , and compiles the acceleration data from the AT2 file, which is the four column format, into an array called `acc_data`.
2. `construct_quake_trajectory.m`: creates a trajectory containing the recorded earthquake acceleration data, $[t, a]$.
3. `q_scale.p`: goes through a scaling algorithm and outputs the time of the trajectory t , the scaled position setpoint in centimeters xd_{cm} , the desired acceleration ad , and the duration of the tremor tf .

These are listed and explained in Table 4.2 as well.

REFERENCES

- [1] Quanser Inc. *QUARC User Manual*.
- [2] Quanser Inc. *Quanser 6 DOF Hexapod User Manual*, 2016.
- [3] Quanser Inc. *QUARC Compatibility Table*, 2016.
- [4] Quanser Inc. *QUARC Installation Guide*, 2016.