1 Introduction

This laboratory manual describes how to use the Quanser Hexapod QUARC® controllers. See the Hexapod User Manual 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 Hexapod.
- Running the supplied earthquakes on the Hexapod system (e.g. Northridge, El-Centro).
- Downloading new earthquakes from the PEER Strong Motion website.
- Transfer function model of the linear actuator.
- PID-based controller used to control position of stage.
- Description of the supplied Hexapod SIMULINK® models.

Caution

Before you begin this laboratory make sure you go through the setup information provided in the Hexapod User Manual. This includes how to install and setup the hardware and the software required for the PC/laptop used with the Hexapod.

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

1. Basics of MATLAB® and SIMULINK®.
2. Basics of QUARC®, i.e. how to build/run a Simulink model in QUARC.
3. Transfer function modeling and PID control is also needed to understand the supplied position control outlined in Section 6

2 Running the Hexapod

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

1. **Driver** - Interface to the Hexapod hardware and accepts commands from ALL Hexapod controllers described below.
2. **Calibration** - Runs an auto routine on all the drives and resets the encoders to zero. Required after system power up.
3. **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).
4. **Joint Level Control** - Control the position of the base joints at the linear actuator level.
5. **World Space Control** - Control the X, Y, Z, Roll, Pitch, and Yaw position of the top stage.
6. **Earthquake** - Run scaled down version of a recorded earthquake on the table, e.g. Northridge or Kobe earthquake.
7. **Brake Bypass** - Releases the brake system to move the stage manually.

Note: All the supplied controllers are already configured for the *hexapod2_usb* data acquisition (DAQ) device.

The sequence of operation of these controller is illustrated in Figure 2.1. After first powering up the Hexapod system, run the driver software followed by the calibration controller and the homing controllers to set the joint to the mid-stroke or *home* position. After that, you can run any of the position-based controllers: joint level control, world space control, or an earthquake.

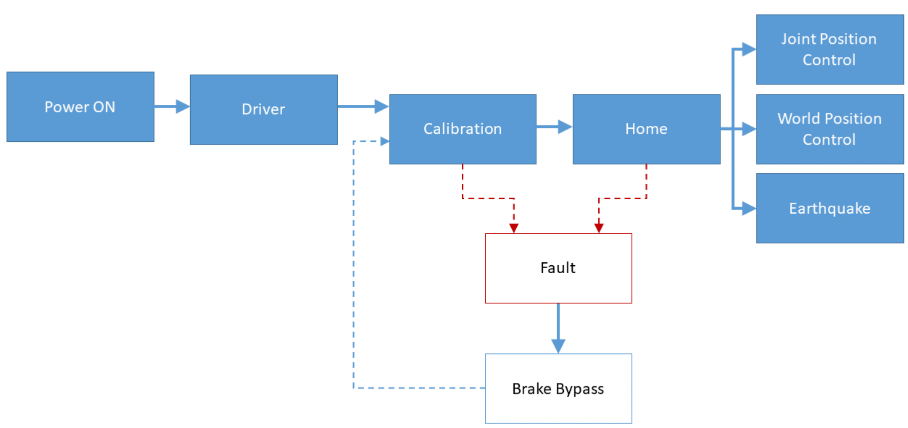


Figure 2.1: Hexapod operation sequence.

2.1 Initialization Procedure

Follow the initialization procedure before running the Hexapod2_Controller_Joint, Hexapod2_Controller_World, and Hexapod2_Controller_Earthquake controllers (or any other controller that send it position commands). This shows how to calibrate and home the Hexapod after power up.

2.1.1 Software Driver

Caution

Make sure Hexapod2_Driver QUARC controller is running BEFORE ANY of the controllers below!

1. Power ON the Hexapod system by placing the power switch to the ON position.
2. Make sure the red button on the Emergency Stop switch is in the released, upper position in order to enable the amplifier (twist the red knob clockwise to release). The *amplifier cannot be enabled when the red button is pressed in* (i.e. in the lower position).
3. Open the Simulink model called Hexapod2_Driver.
4. Run `setup_hexapod2.m` script in MATLAB®. Note: This step is not needed when using the supplied controllers. The Simulink model runs the script automatically when opened.
5. In the Hexapod2_Driver Simulink menu, click on QUARC | Build to generate the QUARC® controller.
6. Run the QUARC controller by going to QUARC | Start.

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. Keep the Hexapod2_Driver running and go to 2.1.2 to calibrate the Hexapod.

Hint: Once Hexapod2_Driver controller has been built in QUARC, you can run the QUARC® Hexapod2_Driver executable directly without having to go through SIMULINK®.

See Section Section 3 and Section Section 4 for more information on `setup_hexapod2.m` and Hexapod2_Driver.

2.1.2 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 Hexapod2_Controller_Calibration controller, shown in Figure 2.2, 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 2.1.3.

Caution

Running the driver and calibration must be performed EVERY TIME the Hexapod is powered ON!

Follow these steps to calibrate the system:

1. Make sure the Hexapod2_Driver model is running in QUARC® before running this controller, as described in Section 2.1.1. The Hexapod2_Driver QUARC executable can be ran directly without SIMULINK® if it has already been generated.

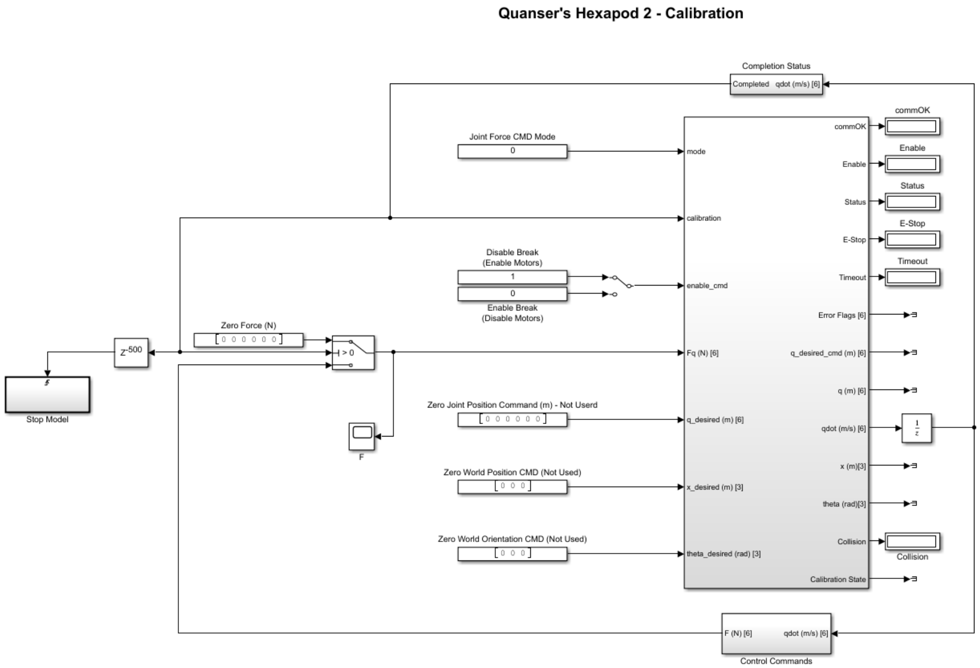


Figure 2.2: Simulink model used to calibrate the Hexapod using QUARC.

2. Make sure the red button on the Emergency Stop switch is in the released, upper position in order to enable the amplifier (twist the red knob clockwise to release). The *amplifier cannot be enabled when the red button is pressed in* (i.e. in the lower position).
3. Open the Simulink model called Hexapod2_Controller_Calibration, shown in Figure 2.2.
4. Run `setup_hexapod2.m` in MATLAB®. Note: This step is not needed when using the supplied controllers. The Simulink model runs the script automatically when opened.
5. Click on QUARC | Build to generate the controller.
6. Power ON the Hexapod system by placing the power switch to the ON position.
7. 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.

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

2.1.3 Homing

Before running Hexapod2_Controller_Joint, Hexapod2_Controller_World, or any other position-based experiments, the stage of the Hexapod must be in the HOME position. This is defined when each base joint is at its mid-stroke position. The Hexapod2_Controller_Home Simulink model, shown in Figure 2.3, is ran with QUARC to center all the linear joints at the base.

Follow these steps to home the system:

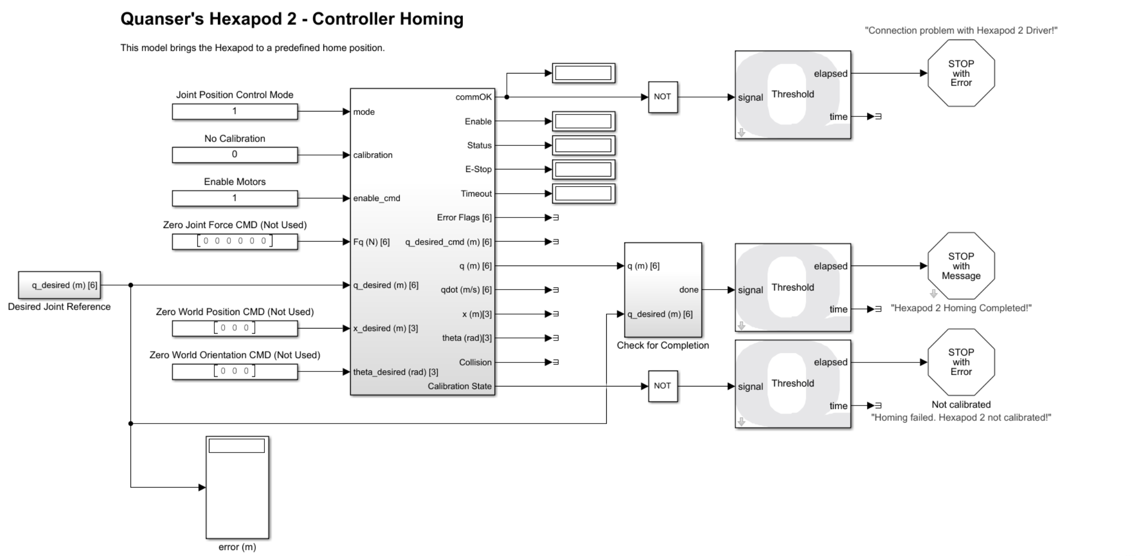


Figure 2.3: Simulink model used to home the Hexapod using QUARC.

1. Make sure the Hexapod2_Driver model is running in QUARC® before running this controller, as described in Section 2.1.1. The Hexapod2_Driver QUARC executable can be ran directly without SIMULINK® if it has already been generated.
2. Make sure the red button on the Emergency Stop switch is in the released, upper position in order to enable the amplifier (twist the red knob clockwise to release). The *amplifier cannot be enabled when the red button is pressed in* (i.e. in the lower position).
3. Make sure you have calibrated the Hexapod first, as described in Section 2.1.2.
4. Open the Simulink model called Hexapod2_Controller_Home, shown in Figure 2.3.
5. Run `setup_hexapod2.m` in MATLAB®. Note: This step is not needed when using the supplied controllers. The Simulink model runs the script automatically when opened.
6. Click on QUARC | Build to generate the controller.
7. 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.

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

2.2 Joint-Level Position Control

This Hexapod2_Controller_Joint controller shown in Figure 2.4 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.

The joint-level commands can be found inside the Desired Joint Reference subsystem shown in Figure 2.5. The home reference command are set in the Slider Gain blocks shown in Figure 2.5 and set to 176.5 mm.

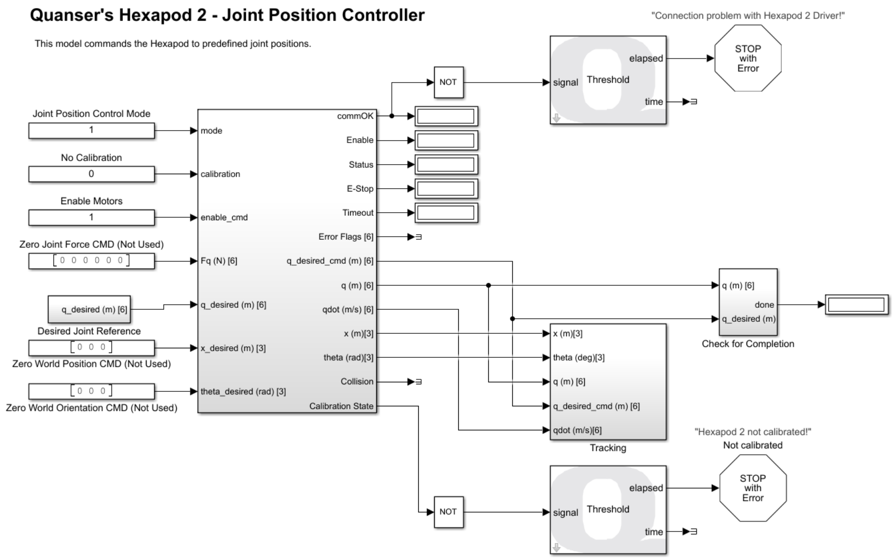


Figure 2.4: Simulink model used to command joint-level positions to the Hexapod using QUARC.

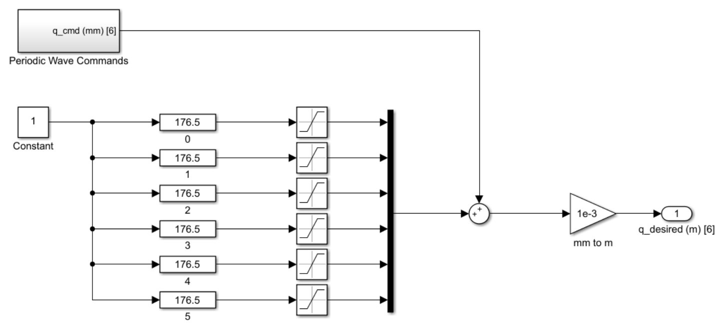


Figure 2.5: Joint-level controller HOME reference commands

The Periodic Wave Commands subsystem, shown in Figure 2.6, allow the user to apply a square or sine wave joint command by changing the Manual Switch. The amplitude and frequency of the sine or square wave can be changed using the Slider Gain blocks that are, by default, all initially set to zero. For example, set the A_0 block to 5 and the f_0 block to 2 to apply a ± 5 mm sine wave at 2 Hz to joint 0.

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 2.1.3.

Follow these steps to run the controller:

1. Make sure the Hexapod has is at the HOME position. Otherwise, go through the initialization procedure in Section 2.1.
2. Make sure the red button on the Emergency Stop switch is in the released, upper position in order to enable the amplifier (twist the red knob clockwise to release). The *amplifier cannot be enabled when the red button is pressed in* (i.e. in the lower position).
3. Open the Simulink model called Hexapod2_Controller_Joint, shown in Figure 2.4.

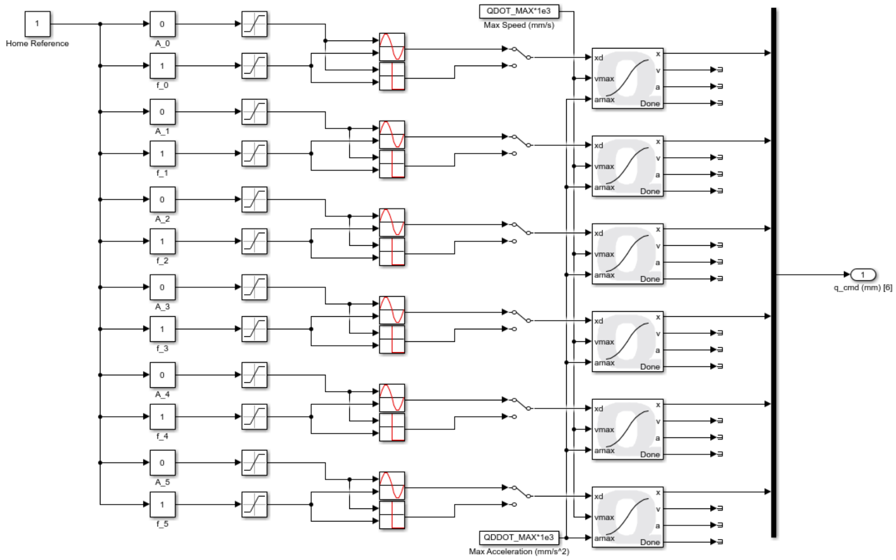


Figure 2.6: Sine or square wave position commands

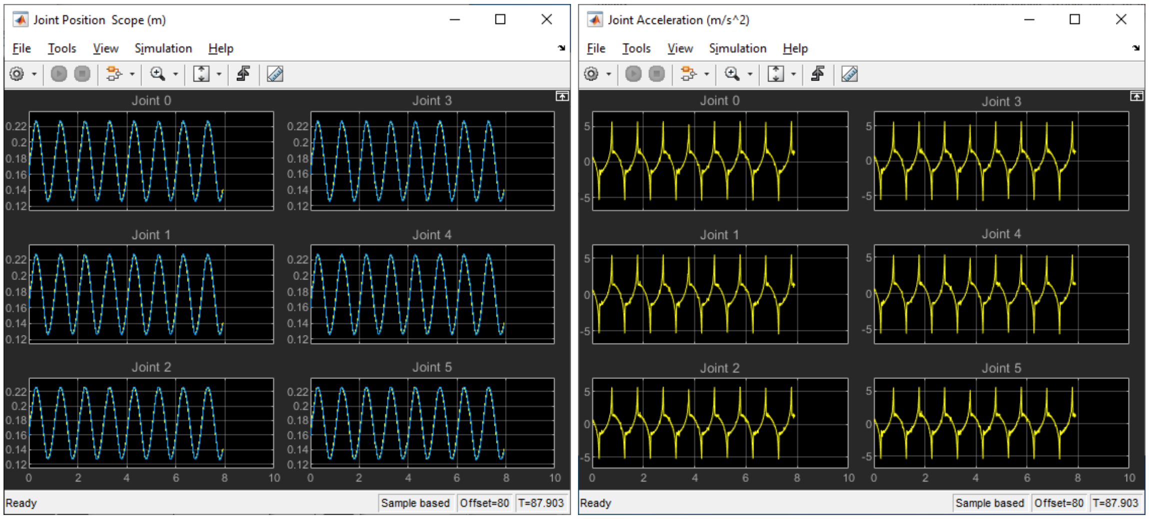
4. Run `setup_hexapod2.m` in MATLAB®. Note: This step is not needed when using the supplied controllers. The Simulink model runs the script automatically when opened.
5. Click on QUARC | Build to generate the controller.
6. Run the QUARC controller by going to QUARC | Start.

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. Change the values of the Slider Gain blocks in the Figure 2.6 gradually to begin commanding a sine wave to the linear actuators.
8. Open the Joint Position and Joint Acceleration scopes located inside the Tracking subsystem, to examine the position and acceleration of the joints. The yellow trace is the desired/commanded position and the light blue plot is the measured position. See Section 4.7 for more information about scopes and viewing data.
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 2.6 all to 0 and the Constant block in Figure 2.5 to 176.5 mm to bring the stage back to HOME position and ready for the next experimental run.
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, 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).



(a) Desired and measured joint positions

(b) Joint accelerations

Figure 2.7: Hexapod2_Controller_Joint scope response example

2.3 World-Space Position Control

This Hexapod2_Controller_World controller shown in Figure 2.8 controls the position of the upper stage. 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.

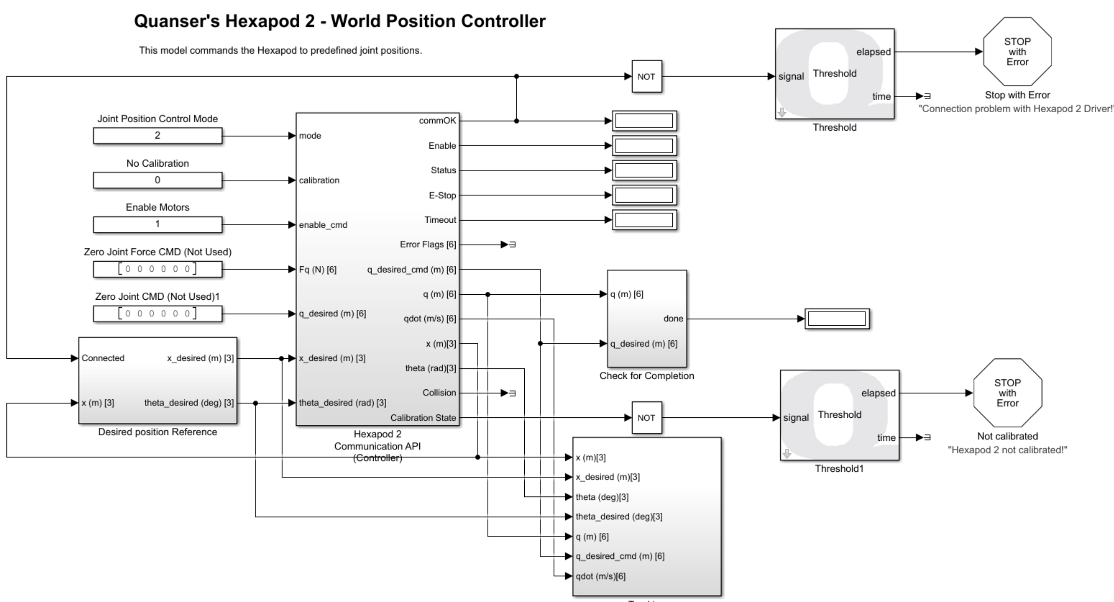


Figure 2.8: Simulink model used to command world-space positions to the Hexapod using QUARC.

The end-effector world coordinate commands are given in the Desired Position Reference subsystem block shown in Figure 2.9. The Slider Gain blocks set a constant world reference position. For example, set K_x to 5 to the stage move to 5 mm along the x-axis.

The setpoint for each of the end-effector's axes - X, Y, Z, Pitch, Roll, and Yaw - can be set to either a sine wave or square wave in the Periodic Wave Commands block illustrated in Figure 2.10. Set the amplitude and frequency of the signal using the Slider Gain blocks. For example, set gain block A_x to 2 and gain block f_x to 0.5 to apply a ± 2 mm sine wave at 0.5 Hz.

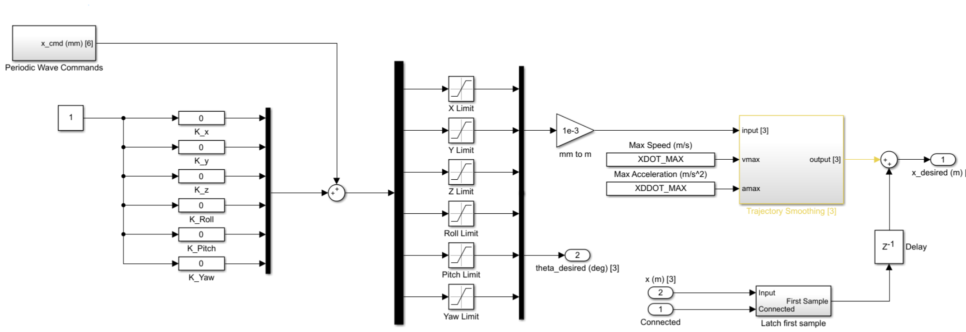


Figure 2.9: Desired position reference subsystem

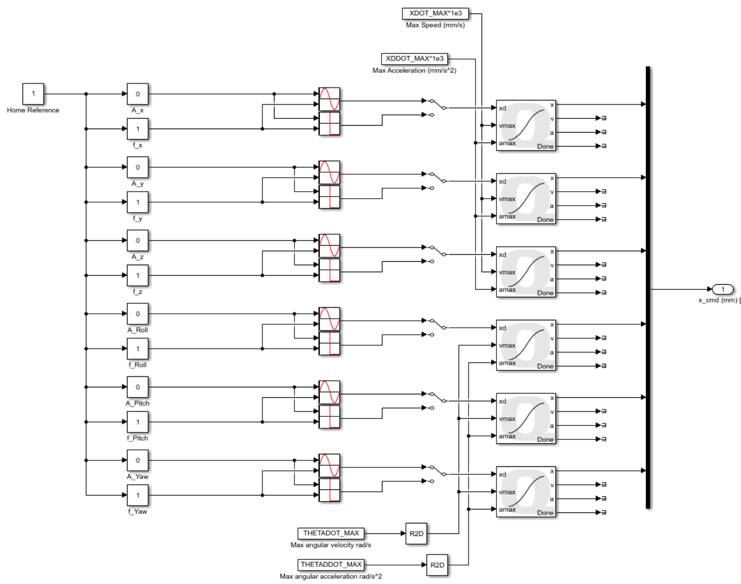


Figure 2.10: Periodic wave world position commands

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 2.1.3.

Follow these steps to run the controller:

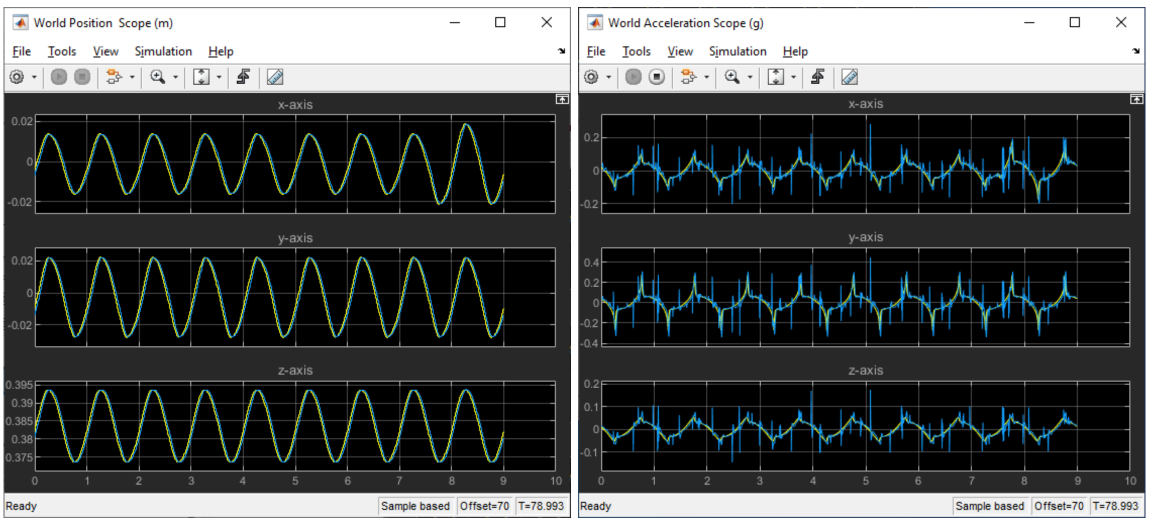
1. Make sure the Hexapod has is at the HOME position. Otherwise, go through the initialization procedure in Section 2.1.
2. Make sure the red button on the Emergency Stop switch is in the released, upper position in order to enable the amplifier (twist the red knob clockwise to release). The *amplifier cannot be enabled when the red button is pressed in* (i.e. in the lower position).
3. Open the Simulink model called Hexapod2_Controller_World, shown in Figure 2.8.
4. Run `setup_hexapod2.m` in MATLAB. Note: This step is not needed when using the supplied controllers. The Simulink model runs the script automatically when opened.
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 2.9 subsystem, gradually to begin commanding a position command to the top stage. See the example scope response shown in Figure 2.11

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.

Caution

8. Open the World Position and World Orientation scopes in the Tracking subsystem to examine the X, Y, Z, roll, pitch, yaw positions. The yellow trace is the desired world coordinate position and the light blue plot is the desired position. See Section 4.7 for more information about scopes and viewing data. The response shown in Figure 2.11 shows the world X, Y, and Z position and acceleration.



(a) Desired and measured x, y, and z positions (b) Measured x, y, and z accelerations (computed)

Figure 2.11: Hexapod2_Controller_World scope response example

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.
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, 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).

2.4 Brake Bypass

The Hexapod2_Controller_Brake_Bypass model, shown in Figure 2.12 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

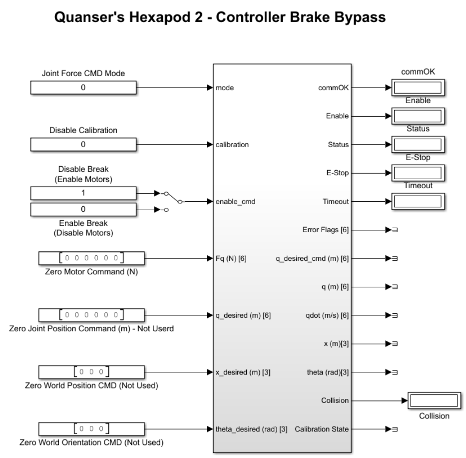


Figure 2.12: Simulink model used to disengage brakes on the Hexapod using QUARC

be moved manually as long as the amplifiers are enabled (and no internal amplifier faults, such as over-current or over-temperature are triggered).

Follow these steps to disengage the brake system:

1. Make sure the Hexapod2_Driver is running, as explained in Section 2.1.1.
2. Make sure the red button on the Emergency Stop switch is in the released, upper position in order to enable the amplifier (twist the red knob clockwise to release). The *amplifier cannot be enabled when the red button is pressed in* (i.e. in the lower position).
3. Open the Simulink model called Hexapod2_Controller_Brake_Bypass, shown in Figure 2.12.
4. Run `setup_hexapod2.m` in MATLAB®. Note: This step is not needed when using the supplied controllers. The Simulink model runs the script automatically when opened.
5. Click on QUARC | Build to generate the controller.
6. Power ON the Hexapod system by placing the power switch to the ON position.
7. 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.

8. 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.
9. Shut off the Hexapod power switch if no more experiments will be conducted.

2.5 Earthquake

The Hexapod2_Controller_Earthquake Simulink diagram shown in Figure 2.13 can be used to replay earthquake on the 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:

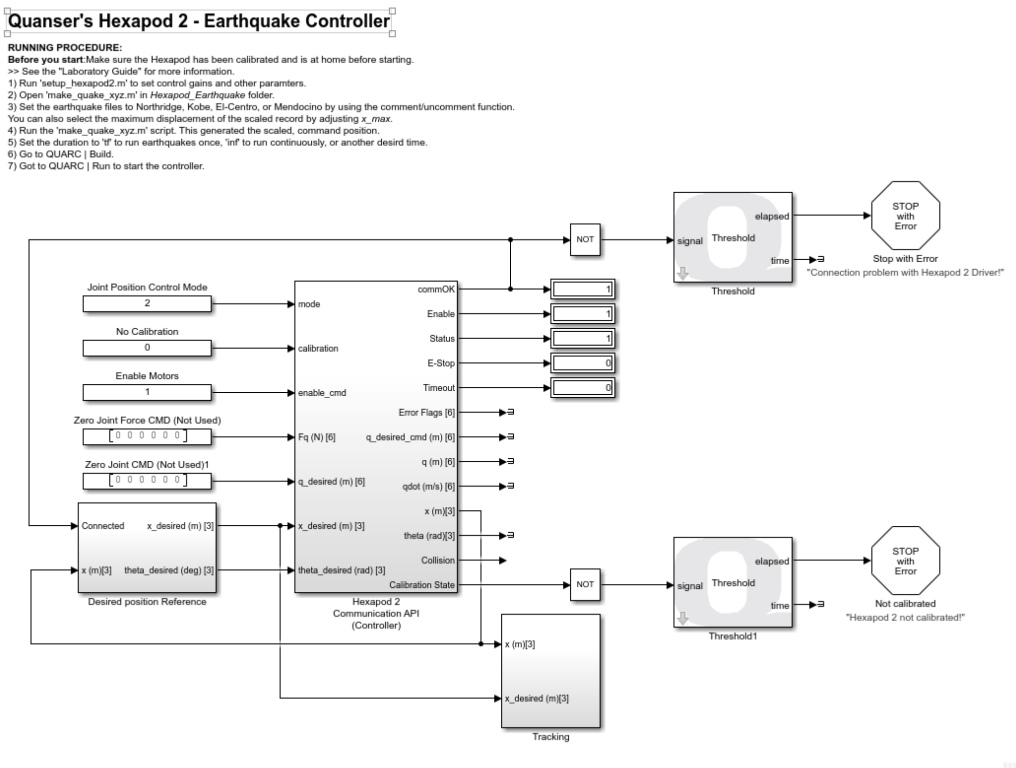


Figure 2.13: Simulink model used to command earthquake to the Hexapod using QUARC.

Northridge, Kobe, El-Centro, and Mendocino. For instructions on how to download new earthquakes, see Section Section 5. The user can also specify and create a pre-defined compound sine wave trajectory.

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 2.1.3.

1. Make sure the Hexapod has is at the HOME position. Otherwise, go through the initialization procedure in Section 2.1.
2. Make sure the red button on the Emergency Stop switch is in the released, upper position in order to enable the amplifier (twist the red knob clockwise to release). The *amplifier cannot be enabled when the red button is pressed in* (i.e. in the lower position).
3. Open the Simulink model called Hexapod2_Controller_Earthquake, shown in Figure 2.13.
4. Run `setup_hexapod2.m` in MATLAB. Note: This step is not needed when using the supplied controllers. The Simulink model runs the script automatically when opened.
5. 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 in `Hexapod_Earthquake` folder** to load the scaled data. For more information about using the `make_quake_xyz.m` script, see Section 3.3.
6. In MATLAB®, go back to the original `Controllers` folder where all Simulink models are located.
7. In Hexapod2_Controller_Earthquake, click on QUARC | Build in Hexapod2_Controller_Earthquake to generate the controller.

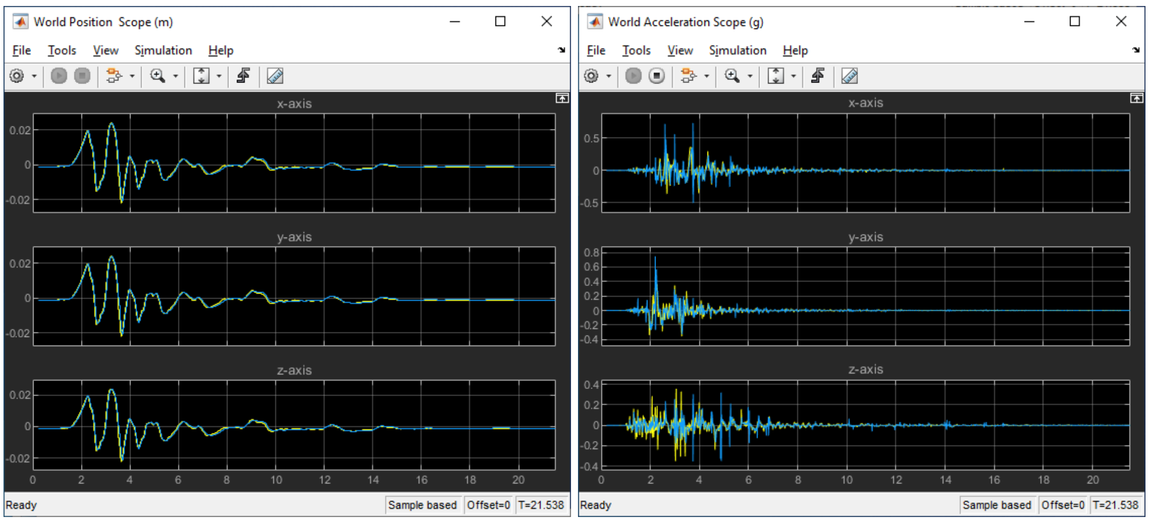
8. Run the QUARC controller by going to QUARC | Start. The top stage should begin tracking the loaded earthquake (or sine wave).

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.

9. Typical position and acceleration responses when running the Northridge earthquake is shown in Figure 2.14 for the x-axis.

   Note: The accelerations are computed from the measured position using second-order high-pass filters. For more information, see Section 4.7.



(a) Desired and measured x, y, and z positions from earthquake profile

(b) Resulting accelerations in x, y, and z axes of earthquake

Figure 2.14: Hexapod2_Controller_Earthquake scope response example

10. The controller stops by itself when the duration of the earthquake (or other loaded motion profile) is reached.
    - The controller can be stopped by clicking on the Stop button in the Simulink diagram tool bar at any time.
    - Set the duration to `tf` to run earthquakes once, `inf` to run them continuously (until the model is stopped), or another desired set time (e.g. 50).
11. Shut off the Hexapod power switch if no more experiments will be conducted.
12. To plot the power spectrum (FFT) of the position or acceleration, run `fft_eval_pos_xyz.m` or `fft_eval_acc_xyz.m`. See Section 2.6 for instructions.

Note: As explained in the Hexapod User Manual, 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).

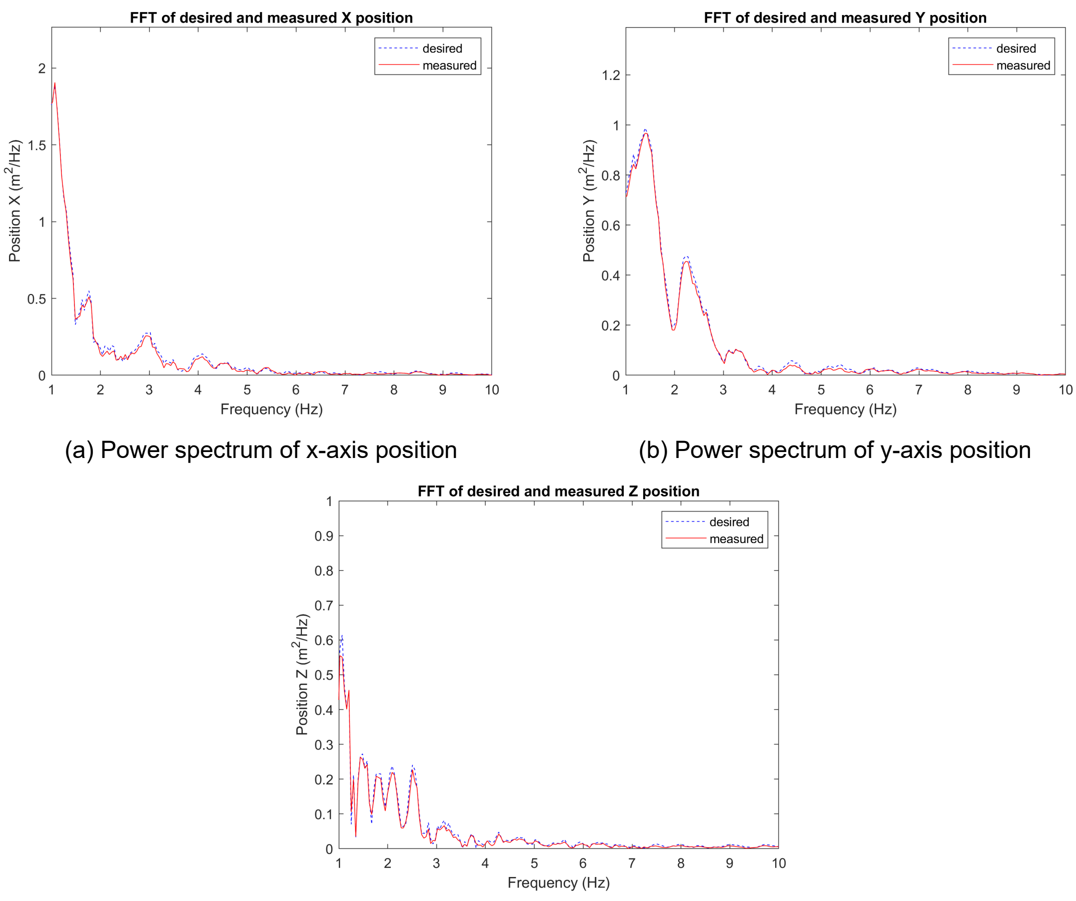
2.6 FFT Analysis

This section shows how to evaluate the **power spectrum** of the commanded and measured Hexapod position and acceleration signals using the `fft_eval_pos_xyz.m` and `fft_eval_acc_xyz.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 `d_power_spectrum.m` script for further details.

1. Run an earthquake using the Hexapod2_Controller_Earthquake QUARC controller, as described in 2.5.
2. After Hexapod2_Controller_Earthquake 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` in the `Hexapod_Earthquake\saved_data` folder.
3. To produce the $|X(\omega)|$ position power spectrum similar to Figure 2.15, run `fft_eval_pos_xyz.m` in the *Hexapod_Earthquake* folder. The desired or commanded position is the blue trace and the measured position (from the encoder) is the red plot line.

   Note: By default, the script loads the data from the saved `data_xyz_pos.mat` file in the `Hexapod_Earthquake\saved_data`. Modify the script if the saved file names or location have changed.



(a) Power spectrum of x-axis position

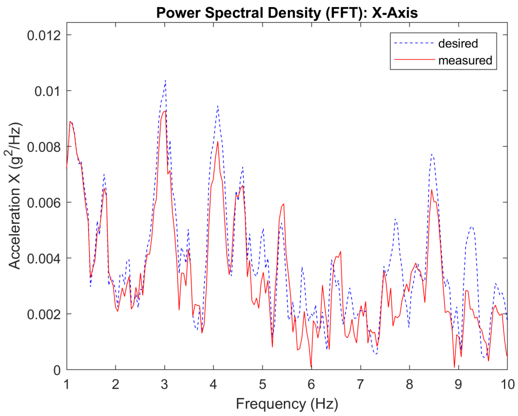
(b) Power spectrum of y-axis position

(c) Power spectrum of z-axis position

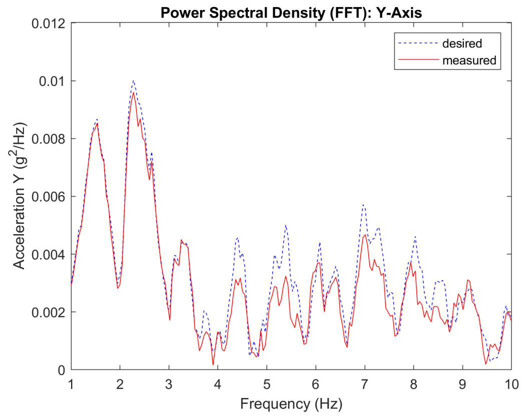
Figure 2.15: Power spectrum of desired and measured position after running Northridge earthquake

4. To produce the acceleration power spectrum, $A(\omega)$, similar to Figure 2.16, run `fft_eval_pos_xyz.m`. The blue and red traces are the desired and measured accelerations, respectively.
5. In both the `fft_eval_pos_xyz.m` and `fft_eval_acc_xyz.m` scripts, you can adjust the frequency range by adjusting the *f_min* and *f_max* variables:

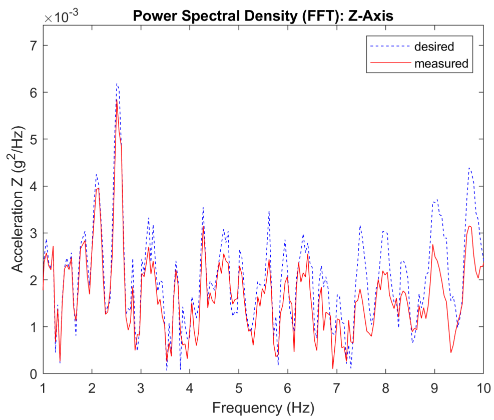
```
% min/max frequencies for plotting (Hz)
```



(a) Power spectrum of x-axis acceleration



(b) Power spectrum of y-axis acceleration



(c) Power spectrum of z-axis acceleration

Figure 2.16: Power spectrum of desired and measured acceleration after running Northridge earthquake

```
f_min = 1;
f_max = 10;
```

3 Supplied Files

3.1 Overview of Files

The main files used for the Hexapod is summarized in Table Table 3.1. The Matlab scripts used to generate earthquake for the Hexapod are listed in Table 3.2.

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_hexapod2.m`	The main Matlab script that sets the Hexapod motor, sensor, and configuration-dependent parameters. **Run this file only to setup the laboratory**. See Section 3.2 for more information.
Hexapod2_Driver	Interfaces to the Hexapod hardware and accepts commands from higher-level controller, e.g. Hexapod2_Controller_Joint.
Hexapod2_Controller_Calibration	Moves each joint to the limit switch and resets the encoders. **Run calibration EVERY TIME the Hexapod is powered ON.**
Hexapod2_Controller_Home	Moves each of the joints to the mid-stroke position. **Run this AFTER calibration and BEFORE any of the position-based controllers** (e.g. Hexapod2_Controller_Joint or Hexapod2_Controller_World).
Hexapod2_Controller_Joint	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.
Hexapod2_Controller_World	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.
Hexapod2_Controller_Earthquake	Runs an earthquake on the Hexapod. Run the `make_quake_xyz.m` script first to load the scaled earthquuake trajectories into the Matlab workspace.
Hexapod2_Controller_Brake_Bypass	Disengages the brake system and enables the amplifier to allow the joints to be moved manually.
`make_quake_xyz.m`	Generates a scaled earthquake trajectory based on an actual earthquake record (.AT2 file). Run this script before using Hexapod2_Controller_Earthquake. See Section 3.3 for more information.
`make_sine_xyz.m`	Generates a compound sine wave according to user-defined amplitudes and frequencies. Similarly to `make_quake_xyz.m`, this script is ran prior to Hexapod2_Controller_Earthquake.

Table 3.1: Files supplied with the Hexapod

File Name	Description
RSN1105_KOBE_HIK000.AT2	Sample Kobe earthquake acceleration data file (direction 000).
RSN1105_KOBE_HIK090.AT2	Sample Kobe earthquake acceleration data file (direction 090).
RSN1105_KOBE_HIK-UP.AT2	Sample Kobe earthquake acceleration data file (direction UP).
RSN1086_NORTHR_SYL090.AT2	Sample Northridge earthquake acceleration data file (direction 090).
RSN1086_NORTHR_SYL360.AT2	Sample Northridge earthquake acceleration data file (direction 360).
RSN1086_NORTHR_SYL-UP.AT2	Sample Northridge earthquake acceleration data file (direction UP).
RSN180_IMPVALL.H_H-E05140.AT2	Sample El-Centro earthquake acceleration data file (direction 140).
RSN180_IMPVALL.H_H-E05230.AT2	Sample El-Centro earthquake acceleration data file (direction 230).
RSN180_IMPVALL.H_H-E05-UP.AT2	Sample El-Centro earthquake acceleration data file (direction UP).
RSN825_CAPEMEND_CPM000.AT2	Sample Mendocino earthquake acceleration data file (direction 000).
RSN825_CAPEMEND_CPM090.AT2	Sample Mendocino earthquake acceleration data file (direction 090).
RSN825_CAPEMEND_CPM-UP.AT2	Sample Mendocino earthquake acceleration data file (direction UP).
q_scale.p	Produces a scaled position trajectory from the recorded acceleration data of the earthquake. The scaled position is used as the position command and produced the same acceleration as the actual recorded earthquake.
init_earthquake_data.m	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*.
construct_quake_trajectory.m	Constructs a time-based array (i.e. a trajectory) containing the recorded earthquake acceleration data.

Table 3.2: Matlab scripts used for running earthquakes on the Hexapod

3.2 Setup script: `setup_hexapod2.m`

The `setup_hexapod2.m` script loads Matlab parameters that are used in the supplied Simulink model for QUARC. The `setup_hexapod2.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 Hexapod files on your PC.
3. Run the `setup_hexapod2.m` script to load all the necessary Hexapod system parameters in the Matlab workspace.

Caution

Most of the settings in `setup_hexapod2.m` should not be changed. Many of these variables are set to the Hexapod system specifications.

3.3 Loading an Earthquake: make_quake_xyz.m

The make_quake_xyz.m script builds a trajectory that can be used in the Hexapod2_Controller_Earthquake 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 3.2, 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 Section 5.

Follow these steps to run the make_quake_xyz.m file:

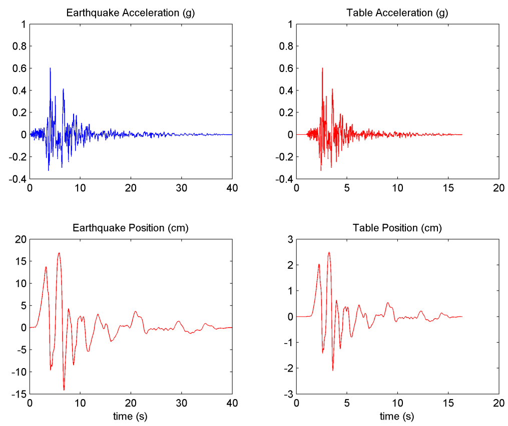
1. Load MATLAB®.
2. In the *Current Directory* window, go to the Hexapod Simulink models folder on your PC/laptop.
3. Open the make_quake_xyz.m script in the Hexapod_Earthquake folder.
4. Set the input_filename_x and input_filename_y variables in the script to the name of the earthquake file that is 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;
%
```

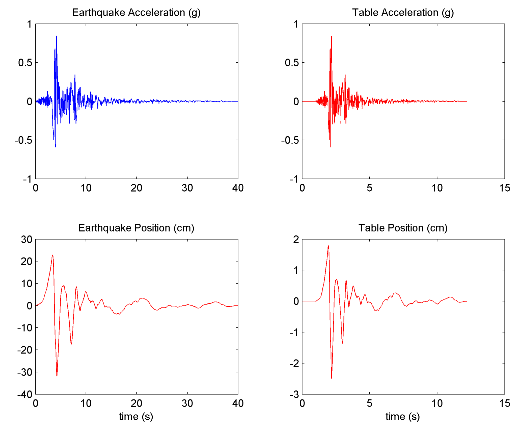
Caution

Do not set x_max to a value greater then the maximum stroke of the Hexapod in any of the x, y, or z directions! The maximum x, y, and z positions is given in the Hexapod User Manual.

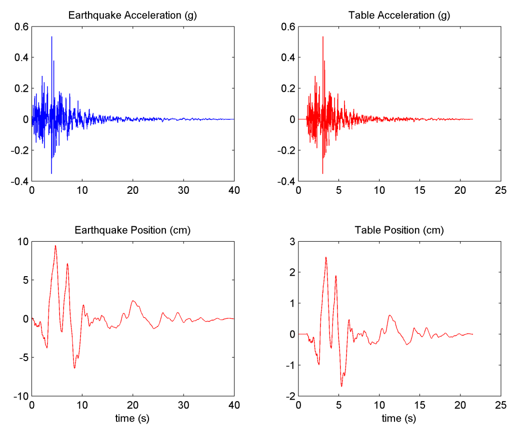
5. Run the make_quake_xyz.m script (click on Debug | Run in the Editor menu bar or on the Run icon in the Editor tool bar) to generate the earthquake wave. This will generate a plot similar to Figure 3.1. The desired acceleration and recorded earthquake acceleration is shown in the top plot (in gravitational units) and the scaled position setpoint in the bottom plot (in centimeters).



(a) X-Axis



(b) Y-Axis



(c) Z-Axis

Figure 3.1: Northridge earthquake plot created by `make_quake_xyz.m` script

Note: As shown in Figure 3.1, the acceleration of the scaled position, `Ac`, is the same as the acceleration of the recorded earthquake, i.e. the unscaled acceleration `Au`.

6. 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_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 Hexapod. See Section 2.5 for the procedure to run the tremor on the table.

4 Controller Description

This section describes some of the blocks that are used in the SIMULINK® QUARC® controller models supplied with Hexapod. Some of the blocks designed specifically for the Hexapod are found in the Hexapod_Library SIMULINK® library, depicted in Figure 4.1.

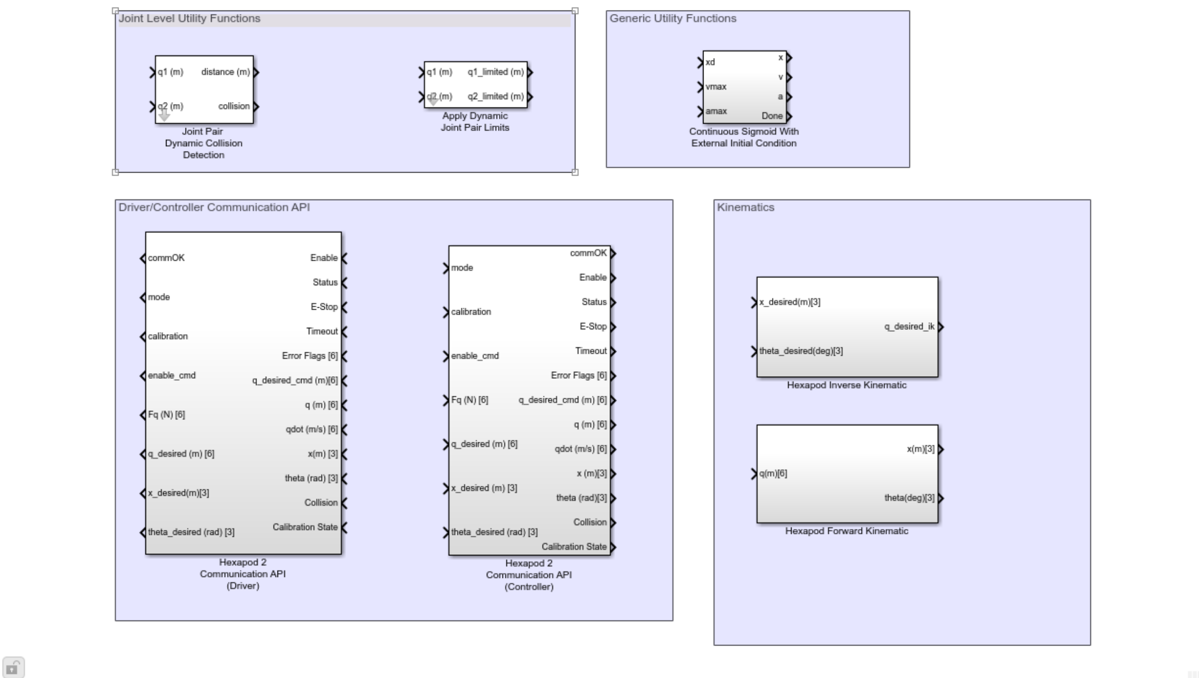


Figure 4.1: Hexapod library used in QUARC controllers supplied

The Hexapod Simulink models use the client-server paradigm shown in Figure 4.2. The Hexapod Simulink controllers interface to the Hexapod2_Driver model using the Hexapod Communications API (Controller) block in Hexapod2_Library, as shown in Figure 4.1. The Hexapod2_Driver model shown in Figure 4.3 includes the Hexapod Communication API (Driver) block that accepts the connections from the controller models (i.e. the client).

Note: Only one controller model can be connected to the driver at one time.

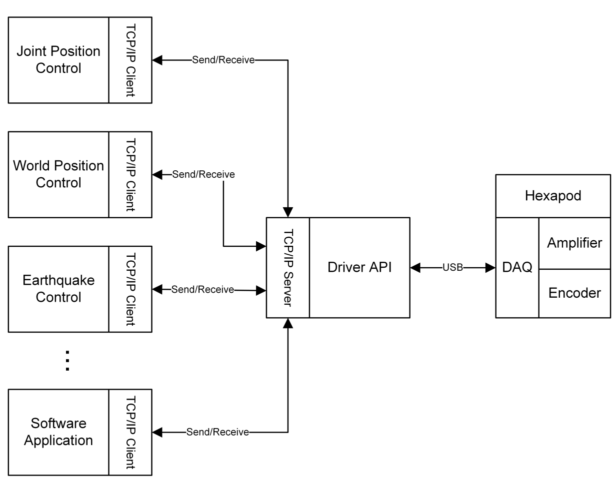


Figure 4.2: Hexapod client-server software architecture

The Hexapod2_Driver model interfaces directly to the Hexapod hardware (e.g. sensors and actuators), includes the kinematics, and implements various motion-based functions. The Hexapod client controller execute higher-order operations, e.g. sending commands, monitoring measurements.

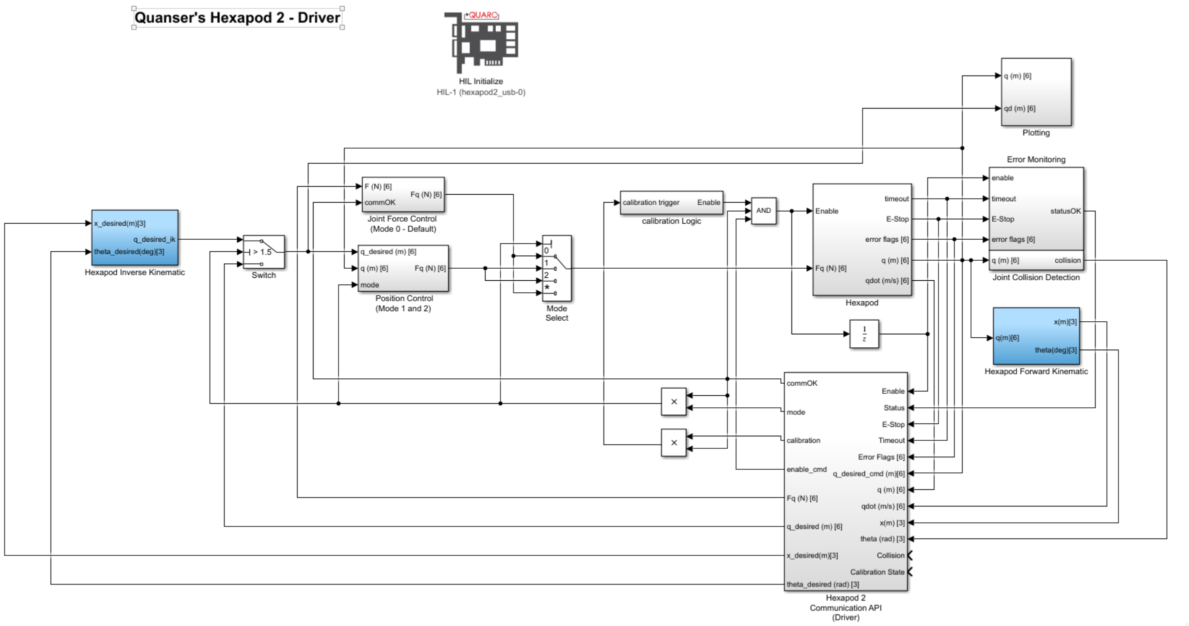


Figure 4.3: Hexapod Hexapod2_Driver interfaces to hardware and is used by all controller models

4.1 Communication API

The Hexapod Communications API (Controller) block in the Hexapod controller models, e.g. Hexapod2_Controller_Joint, uses the QUARC Stream Client block to connect to the Hexapod2_Driver. The Hexapod Communications API (Driver) block in Hexapod2_Driver uses the QUARC Stream Server block to receive information from the controller model. See the QUARC® block documentation for more information.

The description of the input and output ports of the Hexapod Communications API (Controller) subsystem, which are used by all the controller models, are summarized in Table 4.1 and Table 4.2, respectively.

4.2 Hardware Interface

The Hexapod block shown in Figure 4.4 is used in the Hexapod2_Driver. It interfaces to the Hexapod hardware actuators and sensors (e.g. amplifier, encoders), converts the desired force to motor commands and reads the linear joint positions and velocities from the encoder measurements.

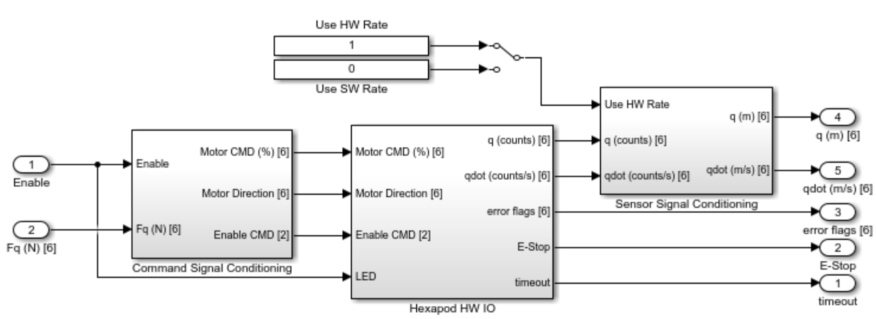


Figure 4.4: Hexapod HW IO subsystem

Input Signal	Description
mode	Set to 0 for force control mode, 1 for joint-Space control mode, and 2 for world space control.
calibration	This signal is set to 1 by the Hexapod2_Controller_Calibration model to indicate that the Hexapod has been calibrated to the Hexapod2_Driver model.
enable_cmd	Enable the amplifiers when set to 1.
Fq (N)	Force vector input that applies forces to the Hexapod joint when in force control mode.
q_desired (m)	Sets the desired joint positions when in joint-space control mode.
x_desired (m)	Sets the desired x, y, and z Cartesian position of the stage (i.e. end-effector) when in world-space control mode.
theta_desired (rad)	Sets the roll, pitch, yaw of the stage (i.e. end-effector) when in world-space control mode.

Table 4.1: Hexapod Communication API (Controller) subsystem input signals

4.2.1 Hexapod HW IO

The Hexapod HW IO subsystem, depicted in Figure 4.5, 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.

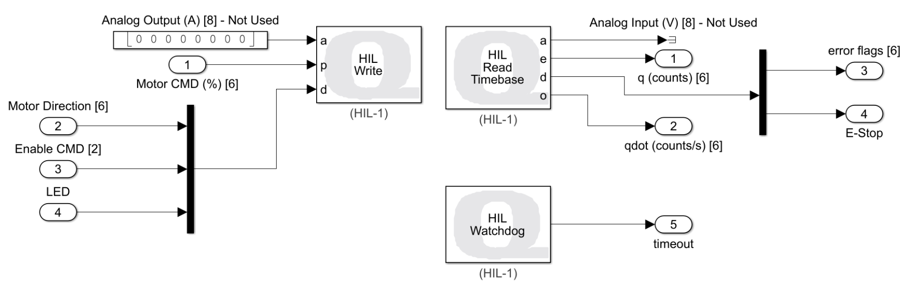


Figure 4.5: QUARC blocks used to interface to Hexapod actuators and sensors

4.2.2 Command Signal Conditioning

The Command Signal Conditioning block used in the Hexapod subsystem is shown in Figure 4.6. Based on the force computed by the Hexapod controller to reach the desired position, denoted as Fq (N), this subsystem computes the current required for the amplifier drives (see Section 6.2 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

Input Signal	Description
CommOK	Outputs 1 when there is a connection established with the Hexapod2_Driver model.
Enable	Outputs 1 when the motors can be driven, 0 otherwise. Motors can be driven when the API is running, the enable command sent, valid data is received, and the E-Stop is connected and in upright, released position.
Status	Output 1 when running normally and 0 if an amplifier fault occurs.
E-Stop	Outputs 0 if E-Stop is connected and in released position. Outputs 1 if E-Stop switch is either not connected to the Hexapod or connected and pressed down to (i.e. disables the amplifiers).
Timeout	Output 1 if the timeout is triggered by the HIL Watchdog block, e.g. due to the controller running too slowly on the PC/laptop.
Error Flags	Vector indicating the status of the amplifiers for each joint. 0 indicates no error.
q_desired_cmd (m)	Outputs the desired joint positions. In joint-space mode, this is the same as the q_desired (m) input. In world-space mode, this is the output of the inverse kinematics.
q (m)	Measured positions of the joints.
qdot (m/s)	Hardware velocity measurement of the joints from the encoder input. Computed on the Hexapod data acquisition device.
x (m)	XYZ Cartesian position of the stage calculated from the measured joint positions.
theta (rad)	Roll, pitch, and yaw angles of the stage calculated from the measured joint positions.
Collision	Outputs 1 if any joint collisions is detected.
Calibration State	Indicates if the Hexapod has been calibrated since the Hexapod2_Driver model has been started, i.e. using the Hexapod2_Controller_Calibration model. Outputs 0 if the system has NOT been calibrated and 1 if it has.

Table 4.2: Hexapod Communication API (Controller) subsystem output signals

over-heating. See Hexapod User Manual for current limit specifications.

4.2.3 Sensor Signal Conditioning

The Sensor Signal Conditioning block used in the Hexapod subsystem is shown in Figure 4.7.

It converts the encoder counts to joint linear position, i.e. multiplied by the encoder resolution and motor shaft circumference constant. The joint velocities can be generated through hardware or software. Use the manual switch in the Hexapod subsystem shown in Figure 4.4 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.

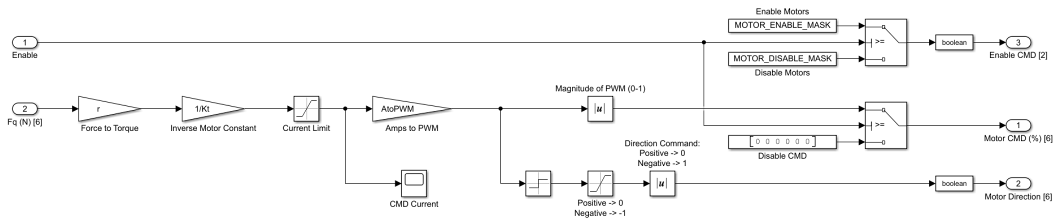


Figure 4.6: Command Signal Conditioning block: converts desired joint forces to motor commands

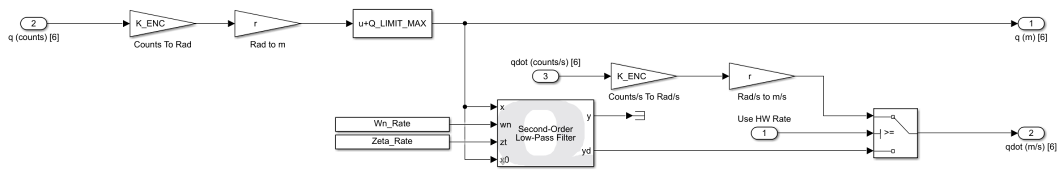


Figure 4.7: Sensor Signal Conditioning block: measures joint position and velocities from encoders

4.3 Position Control

The desired joint positions are either taken directly from the model when using the Hexapod in joint position mode or from the inverse kinematic block when in world control mode. The Joint Position Control block in the Hexapod2_Driver implements the PID controller described in Section 6.2, as shown in Figure 4.8. The control force, i.e. effort required to attain the desired position, is converted to current for the amplifier in the Hexapod subsystem described in Section 4.2.

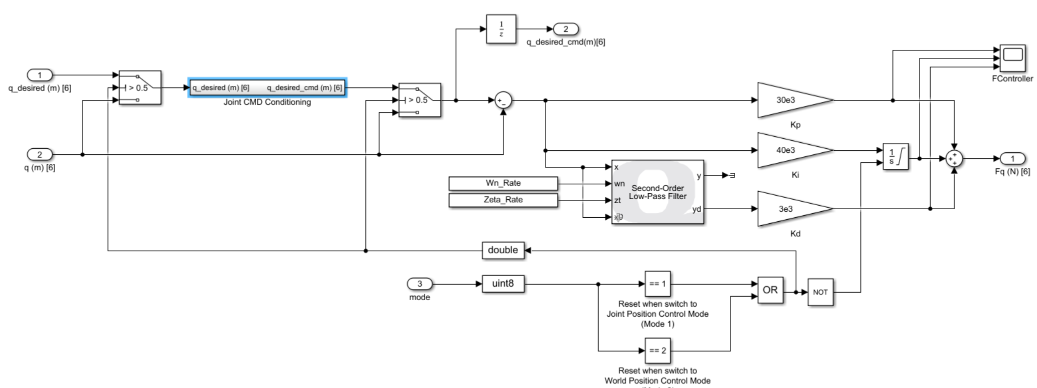


Figure 4.8: PID control is implemented in the Joint Position Control block in the Hexapod2_Driver

Joint Command Limits and Collision Detection The Joint CMD Conditioning block incorporates the Continuous Sigmoid with External Initial Condition block form the Hexapod2_Library to limit the velocity and acceleration of the joint commands. 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. See Section 4.4 for more information.

4.4 Collision Avoidance

The Joint Position Control and Joint Collision Detection blocks in the Hexapod2_Driver model include collision avoidance logic to prevent adjacent joints from colliding with each other. The Joint Position Control detects if any collision will occur based on the desired joint angles while the Joint Collision Detection block checks, based on the measured joint angles, if the joint will collide.

This is implemented in the Apply Dynamic Joint Pair Limits subsystem from the Hexapod2_Library. It does this by keeping a minimum distance between them based on the upper and lower joint limits. The input position references will be passed through if they do not result in a collision condition. Otherwise, the previous desired joint commands will be held until a set of collision-free position inputs are received (and be passed through). The separation distance and joint limits are be set in the `setup_hexapod2.m` script in the `min_separation`, `q_upper`, and `q_lower` variables.

Note that this is defined as the distance between the *joint center to joint center* - not the *side to side* distance. The Joint Pair Dynamic Collision Detection block in Hexapod2_Library is used in the Joint Collision Detection block in Hexapod2_Driver to detect if any collision .

4.5 Kinematics

The forward and inverse kinematic block are available in the Hexapod2_Library shown in Figure 4.1. 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. The orientation the x, y, and z axes are illustrated in Figure 4.9.

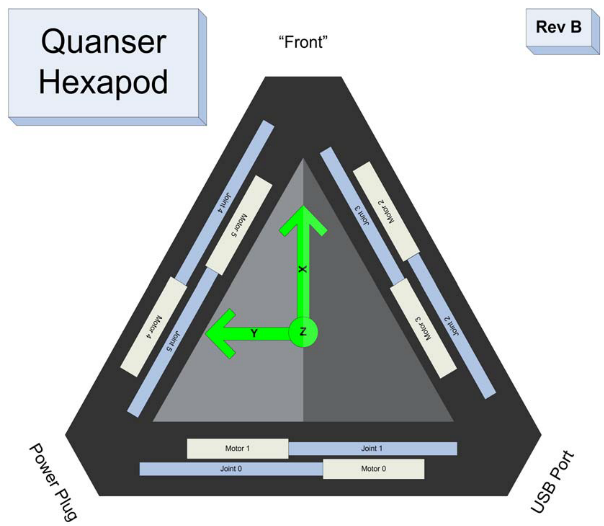


Figure 4.9: X, Y and Z axes orientation relative to the base of the Hexapod

Inverse kinematics calculates the six joint positions from the world coordinate of the end-effector position. The Inverse Kinematics block is used in the Hexapod2_Controller_World model to find the joint-level commands required to attain the desired end-effector position. But it is also used in Hexapod2_Controller_Joint to compare it again the actual, measure joint positions.

4.6 Position Reference

The desired position commands used in the Hexapod2_Controller_Joint, Hexapod2_Controller_World, and Hexapod2_Controller_Earthquake controllers all go through a trajectory smoothing. The Desired position reference block used in the Hexapod2_Controller_World controller is shown in Figure 4.10.

As shown in Figure 4.10, 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.

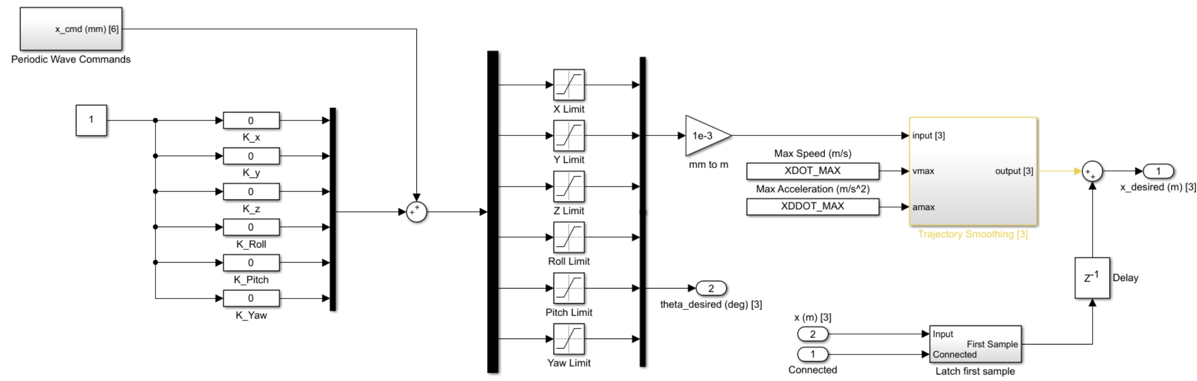


Figure 4.10: The Desired position reference block used in Hexapod2_Controller_World implements position, velocity, and acceleration limits to the desired world commands

4.7 Tracking

The Tracking subsystem, depicted in Figure 4.11, includes various scopes to monitor the position, velocity, and acceleration of the joints and the x, y, z, roll, pitch, and yaw positions of the stage. The To Host File blocks saves the position, velocity, and acceleration data to a Matlab MAT data file. This can be used to plot the response and conduct further analysis in MATLAB®, e.g. FFT shown in Section 2.6.

Note that in controller such as Hexapod2_Controller_World, the World Position Scope plots both the desired world commands and the measured world positions (i.e. x, y, and z). Similarly, the World Orientation Scope monitor both the desired and measured roll, pitch, and yaw angles of the Hexapod. Only the measured world coordinate velocities and accelerations are displayed.

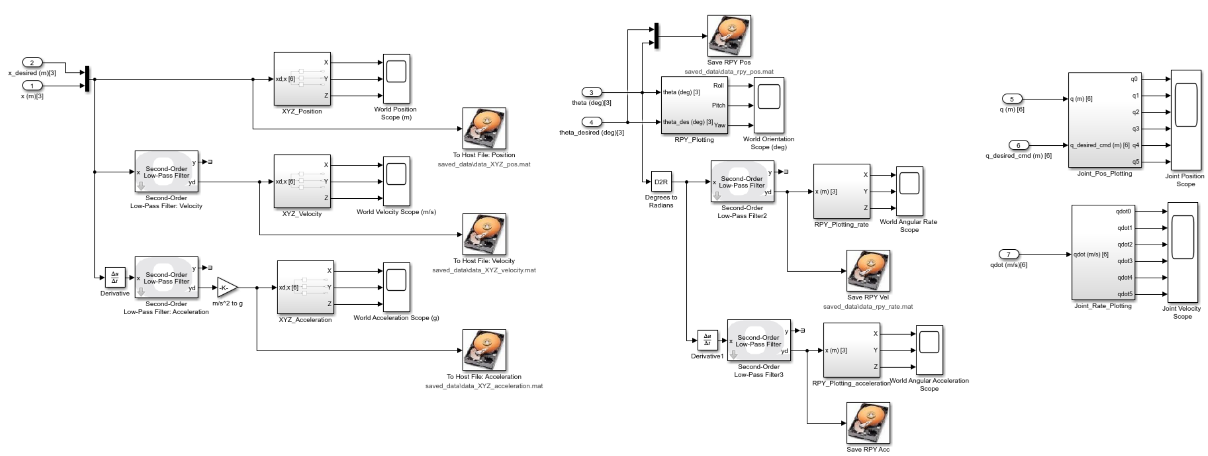


Figure 4.11: Tracking subsystem in the Hexapod2_Controller_World model includes scopes to monitor position and speed of Hexapod

5 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://ngawest2.berkeley.edu/site. Users will have to register on the PEER website to access two ground motion databases: NGA-West2 and NGA-East.

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. You must only use the downloaded acceleration data when using the supplied `make_quake_xyz.m` file detailed in Section 3.3.

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://ngawest2.berkeley.edu/site
2. Sign-up for an account and after successfully logging in, click on the NGA-West2 ground motion database icon (shown in Figure Figure 5.1).

Figure 5.1: The NGA West2 icon

3. Select *No Scaling* from the drop-down menu and click the *Submit* button.
4. In the *Event Name* field, enter *Kobe* and in the *Station Name* filed, enter *HIK* as illustrated in Figure 5.2.

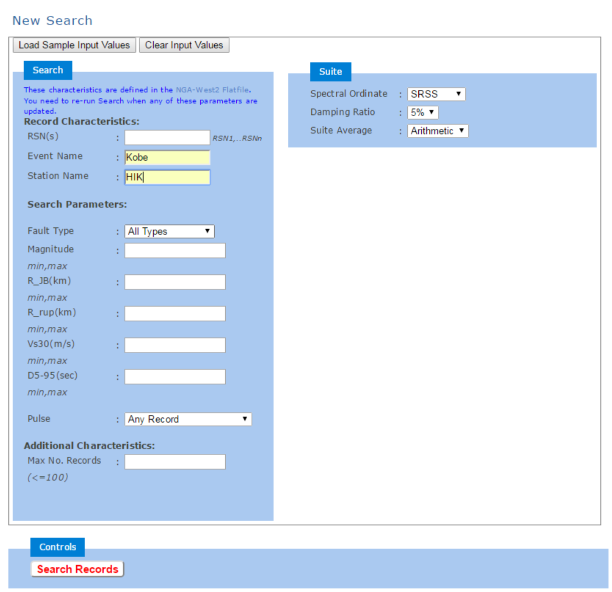


Figure 5.2: Searching for the Kobe earthquake on PEER Ground Motion Database

5. Click on *Search Records* button.
6. After the PEER website successfully processes your search request, scroll down to see the search result under the *Results – Metadata* section as shown in Figure 5.3.
7. Select the check box next to Result ID #1.

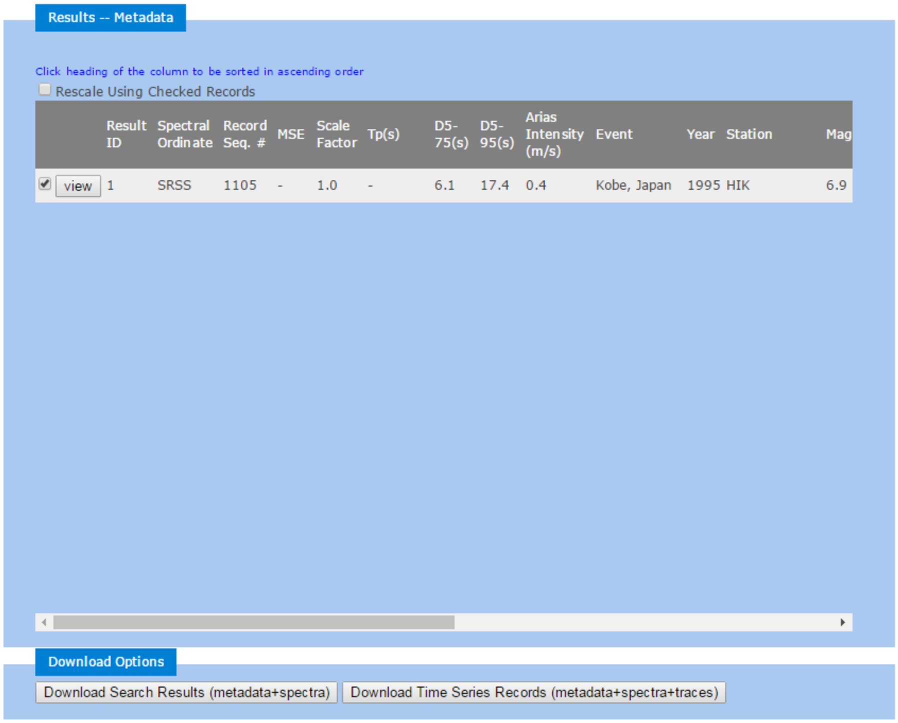


Figure 5.3: Kobe earthquake data on PEER Ground Motion Database as recorded by the HIK station

8. Click on the *Download Time Series Records (metadata+spectra+traces)* button.

9. A ZIP file containing displacement, velocity, and acceleration data for the Kobe earthquake as recorded by the HIK station will be saved to a location on your PC depending on your web browser settings. Extract the contents of the ZIP file and only copy files containing acceleration data (file extension .AT2) to the `Hexapod \Controllers_Hexapod_Earthquake` folder on your PC. This particular record contains the measured acceleration for three different direction: RSN1105_KOBE_HIK000.AT2, RSN1105_KOBE_HIK090.AT2, and RSN1105_KOBE_HIK-UP.AT2.

10. To scale the Kobe earthquake, for example, go to `make_quake_xyz.m` and set the input file variables to:

```
% name of data source file:
input_filename_x = 'RSN1105_KOBE_HIK000.AT2';
input_filename_y = 'RSN1105_KOBE_HIK090.AT2';
input_filename_z = 'RSN1105_KOBE_HIK-UP.AT2';
```

11. Run `make_quake_xyz.m` (see Section 3.3 for details) and then follow the directions given in Section 2.5 to run this earthquake on the Hexapod system.

6 Position Control Design

6.1 Modeling

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

$$q(s) = \frac{1}{M_t\, s^2} F(s) \tag{6.1}$$

where $q(s) = \mathcal{L}\left[q(t)\right]$ 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).

6.2 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:

$$F(t) = k_p\left(q_d(t) - q(t)\right) + k_i \int q_d(t) - q(t)dt + k_d\left(\dot{q}_d(t) - \dot{q}(t)\right) \tag{6.2}$$

where k_p is the proportional control gain, k_i is the integral gain, k_d is the derivative control gain, $q_d(t) \in \Re^6$ is the setpoint or reference joint position (for all six joints), $q(t) \in \Re^6$ 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 6.1.

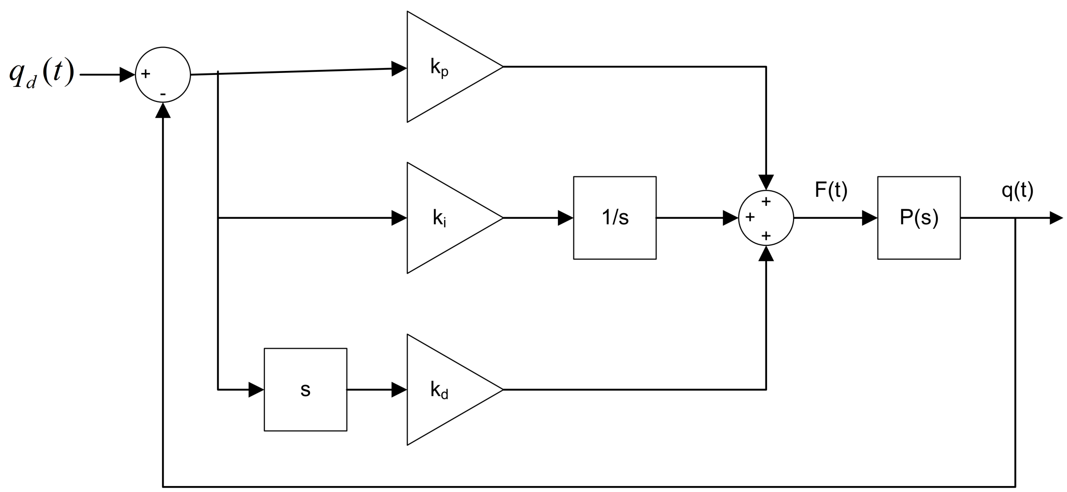


Figure 6.1: Block diagram of Hexapod position control.

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:

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

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 Hexapod software controller and should NOT be tuned by the user unless it is necessary for application-specific reasons.

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

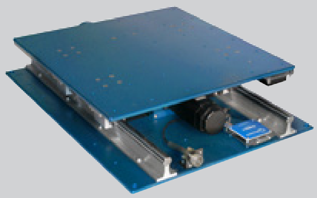
Printed in Markham, Ontario.

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

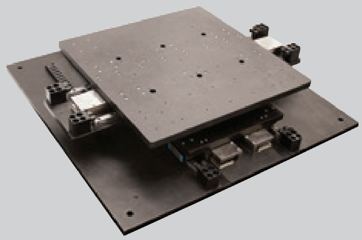
Systems for structural dynamics and analysis teaching and research

Shake Table I-40

Shake Table II

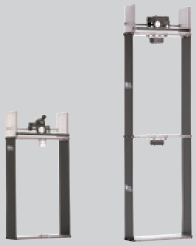


XY Shake Table III



Hexapod

One- or Two-floor Active Mass Damper



Smart Structure

Precise, robust, and flexible, Quanser shake table and smart structure platforms meet the needs of educators and researchers for reliable, low maintenance and cost-effective devices. With these systems you can study various control techniques used to manipulate and dampen structural vibrations and introduce more advanced multi-dynamic analysis. Quanser shake tables and smart structures are also ideally suited for researchers in engineering fields focusing on advanced vibration analysis and isolation, the study of response and elastic limits of structures and geo-materials subjected to dynamic loads.

For more information, please contact info@quanser.com.

QUANSER
INNOVATE·EDUCATE

INFO@QUANSER.COM +1-905-940-3575 QUANSER.COM

Solutions for teaching and research. Made in Canada.