

Test Plan and Results

Overall Test Plan

Proper communication between the External Motion Control (EMC) server and the client application relies on correct handling of UDP packets and a strict order of operations. First, I individually test all of the changes made to the EMC server code. This mainly involves ensuring that UDP packets meant for the client are sent out at the correct time. Normal cases will be tested as well as checking that the code can properly handle abnormal user behavior. Next, I unit test each function of the client application individually. Simulated data will be sent by the server to ensure that the client handles the data as expected. After components from each side are tested individually, integration tests will be done to ensure proper communication between client and server during specific cases. Function of both components will be tested as well as the performance of the client while data is being continuously sent. Finally, a full simulation will be done where all parts are tested together in the expected sequential order.

Test Case Descriptions

CLI 1.1 **Client App Test 1**

CLI 1.2 This test will ensure that the client UI is not accessible before a run command is received.

CLI 1.3 This test will run the client application without sending any UDP signal from the server. No settings or commands should be able to be sent from the client until the Motion Control has sent a run setup command.

CLI 1.4 Input: None.

CLI 1.5 Output: The client should display a "Waiting for run setup" message to the user and hide all other UI elements.

CLI 1.6 Normal

CLI 1.7 Blackbox

CLI 1.8 Functional

CLI 1.9 Unit Test

CLI 1.10 Results: A "Waiting for run setup" message was displayed by the client. No other features of the app were accessible.

CLI 2.1 **Client App Test 2**

CLI 2.2 This test will ensure that the setup options are displayed to the user after the client receives the correct run setup command.

CLI 2.3 The server will send a UDP message to the client with a message saying, "Received MC Run Setup". The client will listen for packets and read their contents, looking for the "Received MC Run Setup" string. Once this string has been read, the client will hide the "Waiting for run setup" message from the screen and show the settings selection prompt to the user.

CLI 2.4 Input: A UDP message containing a string saying, "Received MC Run Setup".

CLI 2.5 Output: The settings selection options should be displayed onscreen.

CLI 2.6 Normal

CLI 2.7 Blackbox

CLI 2.8 Functional

CLI 2.9 Integration

CLI 2.10 Results: The server sends the correct run setup message to the client. The client listens for and receives this message and displays the settings window once it has been received.

CLI 3.1 **Client App Test 3**

CLI 3.2 This test will ensure that the client can properly read UDP messages and ignore packets that do not contain the required command.

CLI 3.3 The client will listen for the "Received MC Run Setup" message from the server with the waiting message displayed onscreen. The server will send a UDP packet containing a message other than the expected run confirmation. The client should be able to read the packet and tell that the run confirmation has not been received. The waiting message should stay onscreen, and the settings options should not be displayed.

CLI 3.4 Input: A UDP message containing a random string.

CLI 3.5 Output: The client should display a "Waiting for run setup" message to the user and hide all other UI elements.

CLI 3.6 Abnormal

CLI 3.7 Whitebox

CLI 3.8 Functional

CLI 3.9 Integration

CLI 3.10 Results: While the client is waiting for the run setup message from the server, any UDP message that does not contain run setup confirmation is ignored by the client.

CLI 4.1 **Client App Test 4**

CLI 4.2 This test will ensure that the user can select a “use default settings” option in the EMC run setup window that automatically saves the default settings.

CLI 4.3 The client will display a settings menu to the user. At the bottom of the menu, the user should be able to press a button that automatically stores all the default settings that are currently available in the server code. The settings menu will disappear, and the data display screen will be shown.

CLI 4.4 Input: A “Use Default Settings” button press.

CLI 4.5 Output: All the EMC settings variables will be set to the default values. The data display screen will appear.

CLI 4.6 Normal

CLI 4.7 Whitebox

CLI 4.8 Functional

CLI 4.9 Unit Test

CLI 4.10 Results: Pressing the “use defaults” button successfully sets all settings values to the default “1” and sends the user to the main window.

CLI 5.1 **Client App Test 5**

CLI 5.2 This test will ensure that the user can select one custom value for each of the available settings categories and save their selections.

CLI 5.3 The client will display a settings menu that allows the user to select one option for each of seven settings categories. Attempting to select multiple options per category will not work. The user should be allowed to select exactly one option for each category and press a “save” button, which will save the user selections and hide the settings menu. The data display screen will appear.

CLI 5.4 Input: The user’s selection of exactly one option per setting and a user confirmation via button press.

CLI 5.5 Output: The user’s selections saved as settings variables, and the data display screen.

CLI 5.6 Normal & Abnormal

CLI 5.7 Whitebox

CLI 5.8 Functional

CLI 5.9 Unit Test

CLI 5.10 Results: The user is allowed to select exactly one option for each category in settings. Selecting another option will remove the first selection.

CLI 6.1 **Client App Test 6**

CLI 6.2 This test will ensure that the user is not allowed to leave a settings category blank. All settings must be selected if the default option is not chosen.

CLI 6.3 The user will attempt to leave several settings options blank and press the confirmation button. An error message should appear asking the user to choose an option for each setting.

CLI 6.4 Input: Fewer than 7 settings selections, and a confirmation button press.

CLI 6.5 Output: An error message.

CLI 6.6 Abnormal

CLI 6.7 Whitebox

CLI 6.8 Functional

CLI 6.9 Unit Test

CLI 6.10 Results: If any number of settings categories does not have an option selected, a message will be displayed to the user indicating that they need to make a selection.

EMC 1.1 **Server Test 1**

EMC 1.2 After receiving a run command from the Motion Control, the server will send a single UDP packet to the client.

EMC 1.3 The EMC server code must be modified to interact with the client. For the client to recognize that a command has been received from Motion Control, the server must send a packet. To test this, the server code is run to check that a packet is sent after the "Received MC Run Setup Message" appears.

EMC 1.4 Input: None

EMC 1.5 Output: A UDP packet containing a "Received MC Run Setup" string.

EMC 1.6 Normal

EMC 1.7 Blackbox

EMC 1.8 Functional

EMC 1.9 Unit

EMC 1.10 Results: When the Run Setup command is received from the Motion Control, a packet is sent over UDP containing the correct message.

EMC 2.1 **Server Test 2**

EMC 2.2 This test ensures that the server has been properly modified to send runtime data to the client without impacting the server's original functions.

EMC 2.3 The server will send packets of data to the client containing EMC data, and packets containing MC data. These packets should be sent continuously without impacting the performance of the server or its communications with the Motion Control.

EMC 2.4 Input: Motion Control data.

EMC 2.5 Output: UDP packets containing Motion Control runtime data. UDP packets containing EMC runtime data.

EMC 2.6 Normal

EMC 2.7 Blackbox

EMC 2.8 Function and Performance

EMC 2.9 Unit

EMC 2.10 Result: The server is able to send EMC data to the client at a rate that is fast enough to keep up with the receiving of runtime data. Due to time constraints, integration with the Motion Control system was never tested.

EMC 3.1 **Server Test 3**

EMC 3.2 This test ensures that if a stop command is received from the Motion Control, a notification is sent to the client and the current run still stops as intended.

EMC 3.3 The server receives a stop command from Motion Control. When the command is received, a UDP packet should be sent by the server to the client containing a "Stop command received" notification.

EMC 3.4 Input: A stop command.

EMC 3.5 Output: Output: A UDP packet containing a stop command.

EMC 3.6 Normal

EMC 3.7 Blackbox

EMC 3.8 Function

EMC 3.9 Unit Test

EMC 3.10 Results: Due to time constraints, full integration with Motion Control has not been tested. A stop function will be added to the client in the future.

INT 1.1 **Integration Test 1**

INT 1.2 This test ensures that changes selected by the user in the client app are reflected by the server.

INT 1.3 Run settings changes are applied by the server. The user will select settings and save their selections. The client will send these selections to the server via UDP. The server will receive the packet and apply its contents to the settings values. The server will display the new values to verify correctness.

INT 1.4 Input: Settings selections in the client.

INT 1.5 Output: The same settings selections in the server.

INT 1.6 Normal

INT 1.7 Blackbox

INT 1.8 Function

INT 1.9 Integration

INT 1.10 Results: The client is able to save settings selected by the user and send them to the server. The server can receive this message and assign the user selected settings to the proper values.

INT 2.1 **Integration Test 2**

INT2.2 This test will ensure that runtime data is sent to the client by the server and displayed by the client.

INT 3.3 The client receives two types of data while a simulation is running, External Motion Control and Motion Control data. The server will send these data to the client in separate packets. The client will receive these packets will be able to differentiate between EMC and MC data. Data will be labelled and displayed onscreen. Data is continuously sent and should be continuously displayed by the client to keep up with the server.

INT 2.4 Input: Server-side EMC and MC Data.

INT 2.5 Output: Properly labelled EMC and MC data values client-side.

INT 2.6 Normal

INT 2.7 Blackbox

INT 2.8 Functional and Performance

INT 2.9 Integration test

INT 2.10 Results: EMC data is continuously sent to the client. The client properly labels and displays the data. MC data has not been tested.

INT 3.1 **Integration Test 3**

INT 3.2 This test ensures that a stop command sent by the client is properly sent to the server and the simulation is stopped.

INT 3.3 A stop command can also be sent by the client. The user will press the stop button in the client. The client will send a UDP packet containing a stop command to the server. The server will receive the packet and respond by stopping the simulation.

INT 3.4 Input: The user presses the stop button.

INT 3.5 Output: A server-side stop command.

INT 3.6 Normal

INT 3.7 Blackbox

INT 3.8 Functional

INT 3.9 Integration Test

INT 3.10 Results: While the server is sending runtime data to the client continuously, it is currently unable to listen for a stop command at the same time. The next step of this project is to implement this function.

FS 1.1 Full Simulation Test 1

FS 1.2 This test ensures that all components of the client and server tested in individual tests above work together as they would in practice.

FS 1.3 The simulation should be run from start to finish. Starting with the initial run setup command, followed by settings setup, a run command, data display, and finally a stop command. All the expected outputs described in other tests should be expected with all parts being tested together.

FS 1.4 Inputs: A run setup command, user settings selections, a run command, Motion Control Data, a stop command.

FS 1.5 Output: Data displayed onscreen.

FS 1.6 Normal

FS 1.7 Blackbox

FS 1.8 Functional and Performance

FS 1.9 Integration

FS 1.10 Results: Integration with the Motion Control system has not been tested. Communications with the client and server have been tested. Run setup, settings selection, run command, and EMC data display all work as expected. MC data is not received and the stop command does not function.

Test Case Matrix

	Normal/ Abnormal	Blackbox/ Whitebox	Functional/ Performance	Unit/ Integration
CLI 1	Normal	Blackbox	Functional	Unit
CLI 2	Normal	Blackbox	Functional	Integration
CLI 3	Abnormal	Whitebox	Functional	Integration
CLI 4	Normal	Whitebox	Functional	Unit
CLI 5	Normal/Abnormal	Whitebox	Functional	Unit
CLI 6	Abnormal	Whitebox	Functional	Unit
EMC 1	Normal	Blackbox	Functional	Unit
EMC 2	Normal	Blackbox	Functional/ Performance	Unit
EMC 3	Normal	Blackbox	Functional	Unit
INT 1	Normal	Blackbox	Functional	Integration
INT 2	Normal	Blackbox	Functional/ Performance	Integration
INT 3	Normal	Blackbox	Functional	Integration
FS 1	Normal	Blackbox	Functional	Integration