

# Testing Procedure

Testing	Mode of Testing	Comments
DIO	Automatic	Run DIOTesting.m to automatically test whether Digital Inputs are working  The code will print out on MATLAB which DIO is currently in use
Communication protocol	Automatic	Print out hello world before reading input from GUI.  Look at communication protocol file.  Check on GUI for respective sent and received messages
	Automatic: Communication between server and client first established	server.py
	Automatic: Continuous communication	GUI has windows to display messages between platforms
	Automatic: Error checking	GUI displays messages and remarks when errors occur
GUI: Robot status - jogging	Manual	Using simulation and testing in labs Do move command and check position
GUI: Robot status - pose	Manual	Using simulation and testing in labs Do move command and check position
GUI: Pose & RAPID:Motion	Manual	Use latest version of GUI.m Do move command check position

	Manual: Moving pose relative to table home	<p>Before reading input from GUI, read status of the robot on the GUI</p> <p>Move the pose relative to Table and verify with inspection and using check status on the GUI that works also in individual directions</p> <p>Visually verify that pose motion works with respect to the Table Home</p> <p>Make sure position is within 5 mm of the actual value</p>
	Manual: Move pose relative to conveyor home	<p>Before reading input from GUI, read status of the robot on the GUI</p> <p>Move the pose relative to Conveyor on GUI and press move. Verify with inspection and using check status on the GUI that works also in individual directions</p> <p>Visually verify that jogging works with respect to the Conveyor Home</p> <p>Make sure position is within 5 mm of the actual value on the GUI</p>
	Manual: Reorienting the end effector	<p>Before reading input from GUI, read status of the robot on the GUI</p> <p>Reorient the end effector values on GUI and press move. Verify with inspection and using check status on the GUI that works also in individual angles</p>

		Visually verify that end effector has remained within the same position
	Manual: Move pose at different speeds	<p>Change speed on GUI to slow and test move Pose</p> <p>Change speed on GUI to regular and test move Pose</p> <p>Change speed on GUI to fast and test move Pose</p> <p>Upon manual inspection verify that the different speeds work as specified on the robot</p>
	Manual: Move joint angles	<p>Before reading input from GUI, read status of the robot on the GUI</p> <p>Move the Joint Angles on GUI and press move. Verify with inspection and using check status on the GUI that works also in individual directions</p> <p>Visually verify that pose motion works with respect to the Table Home</p> <p>Make sure position is within 5 mm of the actual value on the GUI</p>
GUI:Get coordinates from table camera	Automatic	testGUI.m & testGUI.fig Use 001_IMG.jpg
GUI: Get coordinates from conveyor camera	Automatic	testGUI.m & testGUI.fig Use conv_IMG.jpg
GUI check table camera working & detect blocks	Automatic	getBlocks() Sample images

GUI check conveyor camera working & detect box	Automatic	getBox() Sample images
GUI: Click and move to point in table frame	Automatic & Manual	Use automatic testGUI to double check coordinates, and then see if those coordinates are being moved to manually
GUI: Click and move to point in conveyor frame	Automatic & Manual	Use automatic testGUI to double check coordinates, and then see if those coordinates are being moved to manually
RAPID:Motion	Manual and Automatic	Simulation CAD

All automatic testing files are available in GitHub

## GUI: Robot status

### Linear Mode with respect to Base Frame

- Before reading input from GUI, read status of the robot on the GUI
- Jog X, Y and Z individually and verify with manual inspection and using check status on the GUI that jogging works in individual directions
- Visually verify that jogging works with respect to the End Effector Frame

### Linear Mode with respect to End Effector Frame

- Before reading input from GUI, read status of the robot on the GUI
- Jog X, Y and Z individually and verify with manual inspection and using check status on the GUI that jogging works in individual directions
- Visually verify that jogging works with respect to the End Effector Frame

### Reorient End Effector

- Before reading input from GUI, read status of the robot on the GUI
- Jog Pitch, Roll and Yaw individually and verify with manual inspection and using check status on the GUI that jogging works in individual directions
- Make sure with the status that the end effector position has stayed the same and only the angle joints have changed

## Jogging Joints

- Before reading input from GUI, read status of the robot on the GUI
- Jog each joint angle individually and verify with manual inspection and using check status on the GUI that jogging works in individual directions

## Changing speed of jogging

- Change speed on GUI to slow and test jogging speed using X+ on Linear Mode with respect to Base Frame
- Change speed on GUI to regular and test jogging speed using X+ on Linear Mode with respect to Base Frame
- Change speed on GUI to fast and test jogging speed using X+ on Linear Mode with respect to Base Frame
- Upon manual inspection verify that the different speeds work on the robot

## GUI: Video feeds

### Checking Table Feed

- Place letter blocks with known letter and angle on the Robot Table. Also note down the reachability of the letter- if it is within 550mm radius of the robot base, then the block is reachable
- Click on the pushbutton "Find Blocks" to create a snapshot and image process of the video feed
- A figure should come up that visually shows the edge of the blocks as well as display the orientation and the letter. The figure should also visually display the reachability- red if out of bounds, blue if within bounds.
- Check if letters manually found match up with the detected letters on the GUI feed
- Also use testGUI.m to test image

### Checking coordinates

- Use testGUI.m to see if the points are individually reachable by robot.
- Make sure to load IMG\_001.jpg for this.
- Obtain the coordinates and check if it is indeed at that specified axes.
- Obtain the coordinates and check if it is actually reachable by robot manually.

### Checking Conveyor Feed

- Place box with known angle on the Robot Table. Also note down the reachability of the letter- if it is within 550mm radius of the robot base, then the block is reachable
- Click on the pushbutton "Find Box" to create a snapshot and image process of the video feed

- A figure should come up that visually shows the edge of the box as well as display the orientation. The figure should also visually display the reachability- red if out of bounds, blue if within bounds.
- Check if orientation manually found matches up with the detected orientation on the GUI feed

### **Checking coordinates**

- Use testGUI.m to see if the points are individually reachable by robot.
- Make sure to load conv\_IMG.jpg for this.
- Obtain the coordinates and check if it is indeed at that specified axes.
- Obtain the coordinates and check if it is actually reachable by robot manually.

## **Communication protocol**

- Print out hello world before reading input from GUI
- According to the communication protocol, follow the command within the protocol and type in the inputs as required by protocol
- Check that the message is sent in the sent messages on the GUI according to the sent command to verify that the communication protocol sent works

## **GUI: Pose and RAPID: Motion**

### **Move Pose relative to Table Home**

- Before reading input from GUI, read status of the robot on the GUI
- Move the pose relative to Table on GUI and press move. Verify with inspection and using check status on the GUI that works also in individual directions
- Visually verify that pose motion works with respect to the Table Home
- Make sure position is within 5 mm of the actual value on the GUI

### **Move Pose relative to Conveyor Home**

- Before reading input from GUI, read status of the robot on the GUI
- Move the pose relative to Conveyor on GUI and press move. Verify with inspection and using check status on the GUI that works also in individual directions
- Visually verify that jogging works with respect to the Conveyor Home
- Make sure position is within 5 mm of the actual value on the GUI

## Move Joint Angles

- Before reading input from GUI, read status of the robot on the GUI
- Move the Joint Angles on GUI and press move. Verify with inspection and using check status on the GUI that works also in individual directions
- Visually verify that pose motion works with respect to the Table Home
- Make sure position is within 5 mm of the actual value on the GUI

## Reorient End Effector

- Before reading input from GUI, read status of the robot on the GUI
- Reorient the end effector values on GUI and press move. Verify with inspection and using check status on the GUI that works also in individual angles
- Visually verify that end effector has remained within the same position
- Make sure position is within 5 mm of the actual value on the GUI

## Move Pose at different Speeds

- Change speed on GUI to slow and test move Pose
- Change speed on GUI to regular and test move Pose
- Change speed on GUI to fast and test move Pose
- Upon manual inspection verify that the different speeds work as specified on the robot

## Clicking on a position on the video feed and move to that position

### Checking coordinates on Table Feed

- Use testGUI.m to see if the points are individually reachable by robot.
- Make sure to load IMG\_001.jpg for this.
- Obtain the coordinates and check if it is indeed at that specified axes.
- Obtain the coordinates and check if it is actually reachable by robot manually.
- Make sure position is within 5 mm of the actual value

### Checking coordinates on Conveyor Feed

- Use testGUI.m to see if the points are individually reachable by robot.
- Make sure to load conv\_IMG.jpg for this.
- Obtain the coordinates and check if it is indeed at that specified axes.
- Obtain the coordinates and check if it is actually reachable by robot manually.

## DIO

- Run DIOTesting.m to automatically test whether Digital Inputs are working
- The code will print out on MATLAB which DIO is currently in use

- Manually make sure that these values are correct with real values