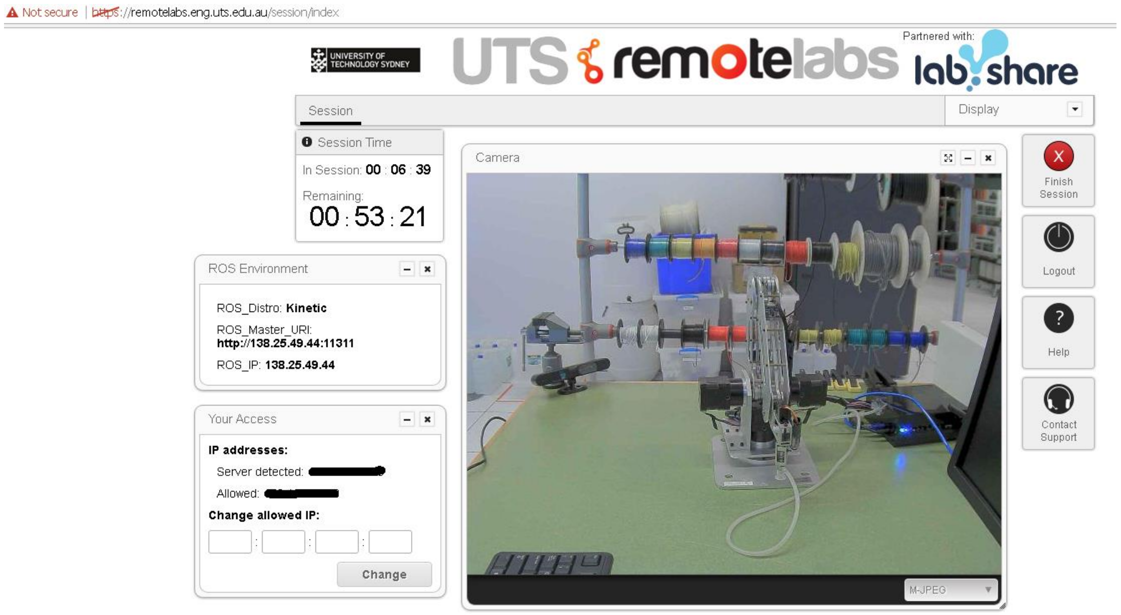
Introduction

This guide will help you use the Dobot Robot and RGBD camera in the remote lab, and get data in Matlab

Software

For this exercise you will need Matlab (with both the Robotics System Toolbox and Peter Corke's Robotic Toolbox), and a browser. Currently, you will also need to be within the UTS network (either wifi, Ethernet or VPN from home).

1. If you have never logged into remote labs then first you must reset your UTS password using https://email.itd.uts.edu.au/webapps/myaccount/ as this synchronises your password with remotelabs systems.
2. Log in to remotelabs website at https://remotelabs.eng.uts.edu.au. your username is your **student number**.
3. Click the tab "Dobot", then "Dobot" button and finally "Queue".
4. The Dobot page will launch.



Not secure | ~~https~~://remotelabs.eng.uts.edu.au/session/index

Session Time

In Session: 00 : 06 : 39

Remaining: 00 : 53 : 21

ROS Environment

ROS_Distro: Kinetic

ROS_Master_URI: http://138.25.49.44:11311

ROS_IP: 138.25.49.44

Your Access

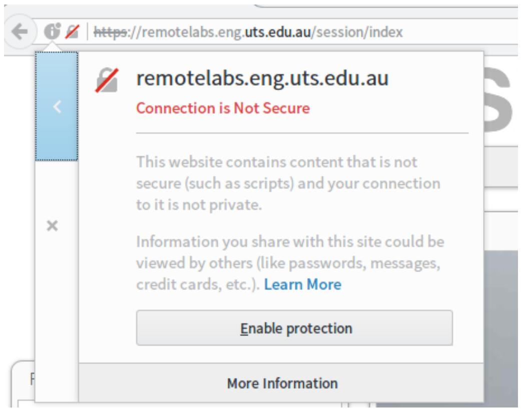
IP addresses:

Server detected: [redacted]

Allowed: [redacted]

Change allowed IP:

5. Enable unsafe scripts in your browser to allow your public IP to be detected by the server. The icon is to the left of the Firefox address bar and to the right of the address bar in Chrome, see the screenshots (Friefox on left, Chrome of right). The server needs to know your IP address so it can configure its firewall to allow your PC to contact ROS.



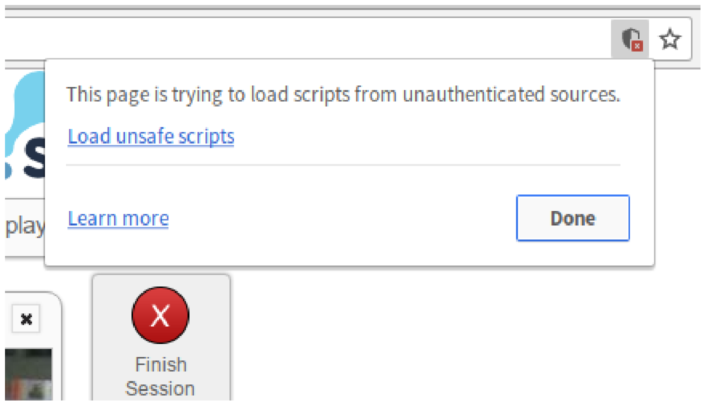
~~https~~://remotelabs.eng.uts.edu.au/session/index

remotelabs.eng.uts.edu.au

Connection is Not Secure

This website contains content that is not secure (such as scripts) and your connection to it is not private.

Information you share with this site could be viewed by others (like passwords, messages, credit cards, etc.). Learn More



This page is trying to load scripts from unauthenticated sources.

Load unsafe scripts

6. Alternate to autodetect, you can manually add your IP address to the "Your Access" form on the page and click "Change".
7. From Matlab you can connect to the Remote Lab ROS server (please change IP address if necessary) with

```
rosinit('138.25.49.44');
```

8. If using Linux and ROS then export the ROS_MASTER_URI environment variable in your terminal to the value shown on the screen so that your local ROS tools can access ROS running on the server.
9. The remote lab set up is very basic and there are still some issues outstanding:
   (1) Automated IP detection needs unsafe scripts to be enabled. We are trying to resolve this with UTS IT support (i.e. ServiceConnect).

   (2) The Dobot remotelab is only accessible within the UTS network. We have requested to have ServiceConnect open the ROS port and ephemeral ports to the public so you can connect from the internet.

Installing Custom ROS Messages in Matlab

Download matlab_gen.7z file for your version of Matlab from the resources folder on UTSOnline and unzip it

In Matlab add the msggen folder to the path or call:

C:\BitBucket\RemoteProtoLab\matlab_gen

```
addpath(genpath([pwd,'\msggen']));
```

Add [**pwd**,'\jar\dobot_ros-0.0.0.jar'] statically to the file javaclasspath.txt in the Matlab startup directory or app directory (mine is) *C:\Users\gapaul\AppData\Roaming\MathWorks\MATLAB\R2016a*
Note that **pwd** is the full path of where matlab_gen.7z was unzipped to, e.g.
C:\matlab_gen\jar\dobot_ros-0.0.0.jar

After changing javaclasspath.txt restart Matlab. Then check that the jar file has been added in Matlab by calling

```
javaclasspath
```

Connect Remotely to ROS from Matlab

On your or PC, open Matlab. Make sure you are connected to the remote lab and have an IP address that is allowable. For these commands you will need the Matlab Robotics System Toolbox (should be avialble by default from 2016 onwards). Note this is different to Peter Corke's Robotic Toolbox. You cannot use **Matlab Online**. Connect to ROS server with (ensure the IP address here matches the one of the remote lab from the browser)

```
rosinit('138.25.49.44');
```

Ensure required custom ROS messages are available (returns: **1** when dobot messages are found, and **[]** when not found)

```
find(cell2mat(strfind(rosmsg('list'),'dobot')),1)
```

Get the ROStopics available

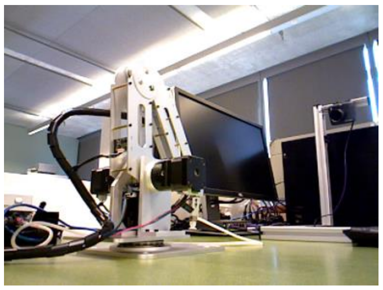
```
rostopic list
```

Make sure at least the following are available

```
/dobot/state
/camera/depth/camera_info
/camera/depth/image_raw
/camera/rgb/camera_info
/camera/rgb/image_raw
/rosout
/rosout_agg
```

Get started with ROS in Matlab: http://au.mathworks.com/help/robotics/examples/get-started-with-ros.html

Get and Show Color Images



```
rgbSub = rossubscriber('/camera/rgb/image_color');
pause(1);
image_h = imshow(readImage(rgbSub.LatestMessage));
```

Make the figure large

```
set(gcf,'units','normalized','outerposition',[0 0 1 1]);
```

Now loop through and update the image data with the latest image data (ctrl + c to stop)

```
tic
while 1
    image_h.CData = readImage(rgbSub.LatestMessage);
    drawnow;
    toc;
end
```

Get and Show Depth Images

```
depthSub = rossubscriber('/camera/depth/image');
pause(1);
msg = depthSub.LatestMessage;
img = readImage(msg);
depthImage_h =imshow(img)
```

Make the figure large

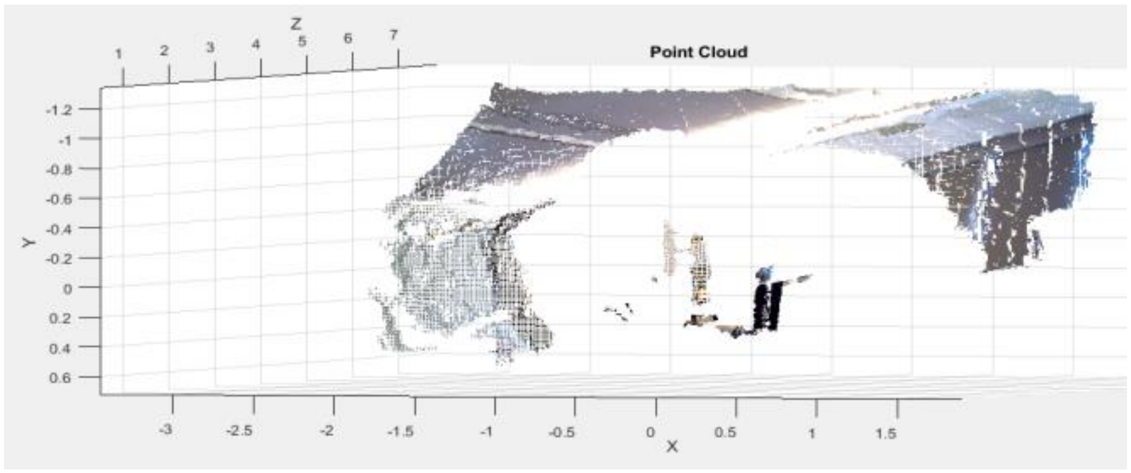
```
set(gcf,'units','normalized','outerposition',[0 0 1 1])
```

Now loop through and update the image data with the latest image data (ctrl + c to stop)

```
tic
while 1
    depthImage_h.CData = readImage(depthSub.LatestMessage);
    drawnow;
    toc;
end
```

Get and Show Point cloud

See the example on UTSOnline ColorPointTest.m to display something like below



Get Dobot State and Control from Matlab

Assuming that the custom messages are setup, roscore and the dobot node is started remotely, and /dobot/state is shown when `rostopic list` is called in Matlab. Then you can check the Dobot arm state with

```
dobotStateSub = rossubscriber('/dobot/state');
receive(dobotStateSub,2)
msg = dobotStateSub.LatestMessage;
```

To move the joints of the arm and ensure completion (remember **angles are in radians**)

```
msg = rosmessage('dobot_ros/SetPosAngRequest');
msg.RearArmAngle = 0.4;
msg.ForeArmAngle = 1.2;
msg.BaseAngle = 0;
srv = rossvcclient('/dobot/joint_angles');
srv.call(msg)
```

Note that once the above is called you could simply call the following to change the rear arm angle

```
msg.BaseAngle = -pi/4;
srv.call(msg)
```

To request the robot to move in Cartesian space (all in meters)

```
cartSvc = rossvcclient('dobot/cartesian');
cartMsg = rosmessage(cartSvc);
cartMsg.Pos.X = 0.2;
cartMsg.Pos.Y = 0.1;
cartMsg.Pos.Z = -0.02;
cartSvc.call(cartMsg);
```

To turn the pump on. Note: this may cause the robot to freeze, if so, logout of the remote lab and then back again to fix this.

```
msg2 = rosmessage('dobot_ros/EECtrlRequest');
msg2.Pump = 1;
eeSrv = rossvcclient('/dobot/ee_ctrl');
eeSrv.call(msg2)
```

Then to turn it off again

```
msg2.Pump = 0;
eeSrv.call(msg2);
```

After logging out the robot will automatically return to the home configuration and the pump will turn off.