

**INTRODUCTION TO ROBOTICS**

**PROJECT 2 REPORT**

**NS Raabid Hussain**

**Reg # 2011-NUST-CEME-BE-Mts-361**

**NS Raafay Ijaz**

**Reg # 2011-NUST-CEME-BE-Mts-409**

Syndicate A

Department of MECHATRONICS Engineering

Submitted To: Col K Faraz / LE Adnan Shuja

Date: 15/12/2014

**INTRODUCTION:-**

**MATLAB** is a powerful environment for linear algebra and graphical presentation that is available on a very wide range of computer platforms. The core functionality can be extended by application specific toolboxes. The Robotics Toolbox provides many functions that are required in robotics and addresses areas such as kinematics, dynamics, and trajectory generation. The Toolbox is useful for simulation as well as analyzing results from experiments with real robots, and can be a powerful tool for education.

The **Toolbox** is based on a very general method of representing the kinematics and dynamics of serial-link manipulators by description matrices. These comprise, in the simplest case, the Denavit and Hartenberg parameters of the robot and can be created by the user for any serial-link manipulator.

The manipulator description can be elaborated, by augmenting the matrix, to include link inertial, and motor inertial and frictional parameters. Such matrices provide a concise means of describing a robot model and may facilitate the sharing of robot models across the research community. This would allow simulation results to be compared in a much more meaningful way than is currently done in the literature.

The Toolbox also provides functions for manipulating datatypes such as vectors, homogeneous transformations and unit-quaternions which are necessary to represent 3-dimensional position and orientation. The routines are generally written in a straightforward, or textbook, manner for pedagogical reasons rather than for maximum computational efficiency.

**Robot Selection:-**

The robot selected for this project is Lab WheelBarrow MK9 by Northrup Grunmun which is a 5DOF robot with 4 revolute and 1 prismatic joint.

The Wheelbarrow was developed in the 1970s as part of the UK Government's intent to deploy remote control vehicles in the battlefield. It has been used by bomb disposal squads in Northern Ireland and Iraq. The UGV's development involved a consortium of companies, including Alvis Logistics, the predecessor of Remotec UK.

The Mk9 can remotely detonate IED and undertake several hazardous operations such as ground surveillance and chemical, biological, radiological and nuclear missions.

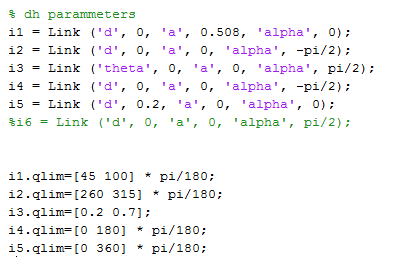
The uncluttered superstructure of the UGV has a wash-down capability with an improvised under-vehicle reach. The Mk9 has a modular telescopic arm with a maximum reach of about 6m. It can cross any terrain and climb stairways of 45° to reconnoitre suspect devices. The maximum speed of the Mk9 is 5km/hr. It has a unique design with a low profile three-fingered gripper. It can lift a payload of 150kg.



**DH Parameters:-**

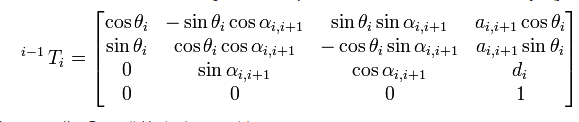
The dh parameters calculated were as follows:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **i** | **αi-1** | **Αi-1** | **di** | **ϑi** |
| 1 | 0 | 0 | 0 | Θ1 |
| 2 | 0 | L2 = 0.508m | 0 | Θ2 |
| 3 | -90 | 0 | D3 | 0 |
| 4 | 90 | 0 | 0k8 | Θ4 |
| 5 | -90 | 0 | L5 | Θ5 |

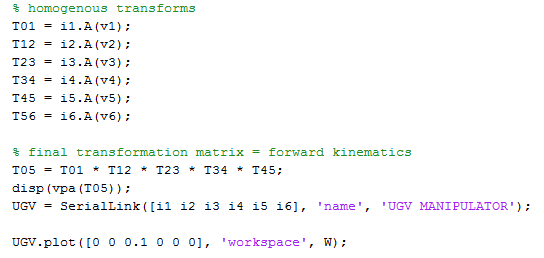


**Forward Kinematics:-**

**Forward kinematics** refers to the use of the [kinematic](http://en.wikipedia.org/wiki/Kinematic) equations of a [robot](http://en.wikipedia.org/wiki/Robot) to compute the position of the [end-effector](http://en.wikipedia.org/wiki/Robot_end_effector) from specified values for the joint parameters.

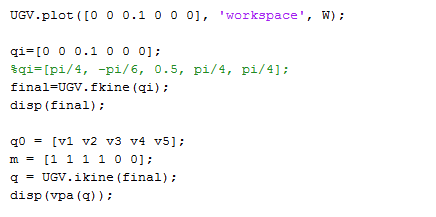


The forward kinematics can be found out by multiplying the transformation matrices for all the link / joints.



**Inverse Kinematics:-**

**Inverse kinematics** refers to the use of the [kinematics](http://en.wikipedia.org/wiki/Kinematics) equations of a robot to determine the joint parameters that provide a desired position of the [end-effector](http://en.wikipedia.org/wiki/Robot_end_effector).[[1]](http://en.wikipedia.org/wiki/Inverse_kinematics#cite_note-1) Specification of the movement of a robot so that its end-effector achieves a desired task is known as [motion planning](http://en.wikipedia.org/wiki/Motion_planning). Inverse kinematics transforms the motion plan into joint [actuator](http://en.wikipedia.org/wiki/Actuator) trajectories for the robot.



**Code:-**

% UGV Forward kinematics

close all;

clc;

clear;

syms v1 v2 v3 v4 v5 v6;

l2 = 0.508;

l5 = 0.2;

W = [-0.5 2 -0.5 2 -0.5 2];

% dh parammeters

i1 = Link ('d', 0, 'a', 0.508, 'alpha', 0);

i2 = Link ('d', 0, 'a', 0, 'alpha', -pi/2);

i3 = Link ('theta', 0, 'a', 0, 'alpha', pi/2);

i4 = Link ('d', 0, 'a', 0, 'alpha', -pi/2);

i5 = Link ('d', 0.2, 'a', 0, 'alpha', 0);

i6 = Link ('d', 0, 'a', 0, 'alpha', pi/2);

i1.qlim=[45 100] \* pi/180;

i2.qlim=[260 315] \* pi/180;

i3.qlim=[0.2 0.7];

i4.qlim=[0 180] \* pi/180;

i5.qlim=[0 360] \* pi/180;

i6.qlim=[0 0];

% homogenous transforms

T01 = i1.A(v1);

T12 = i2.A(v2);

T23 = i3.A(v3);

T34 = i4.A(v4);

T45 = i5.A(v5);

T56 = i6.A(v6);

% final transformation matrix = forward kinematics

T05 = T01 \* T12 \* T23 \* T34 \* T45;

disp(vpa(T05));

UGV = SerialLink([i1 i2 i3 i4 i5 i6], 'name', 'UGV MANIPULATOR');

UGV.plot([0 0 0.1 0 0 0], 'workspace', W);

qi=[0 0 0.1 0 0 0];

%qi=[pi/4, -pi/6, 0.5, pi/4, pi/4];

final=UGV.fkine(qi);

disp(final);

q0 = [v1 v2 v3 v4 v5];

m = [1 1 1 1 0 0];

q = UGV.ikine(final);

disp(vpa(q));

check=UGV.fkine(q);

disp(check);

%forward

f=[pi/4 -pi/6 0.5 pi/4 pi/4 0];

t=[80\*pi/180 -pi/6 0.1 pi/4 pi/4 0];

newpos=jtraj(f, t, 60);

UGV.plot(newpos, 'workspace', W);

%inverse

s=transl(0, 0, 0);

d=transl(0.508, 0.3, 0);

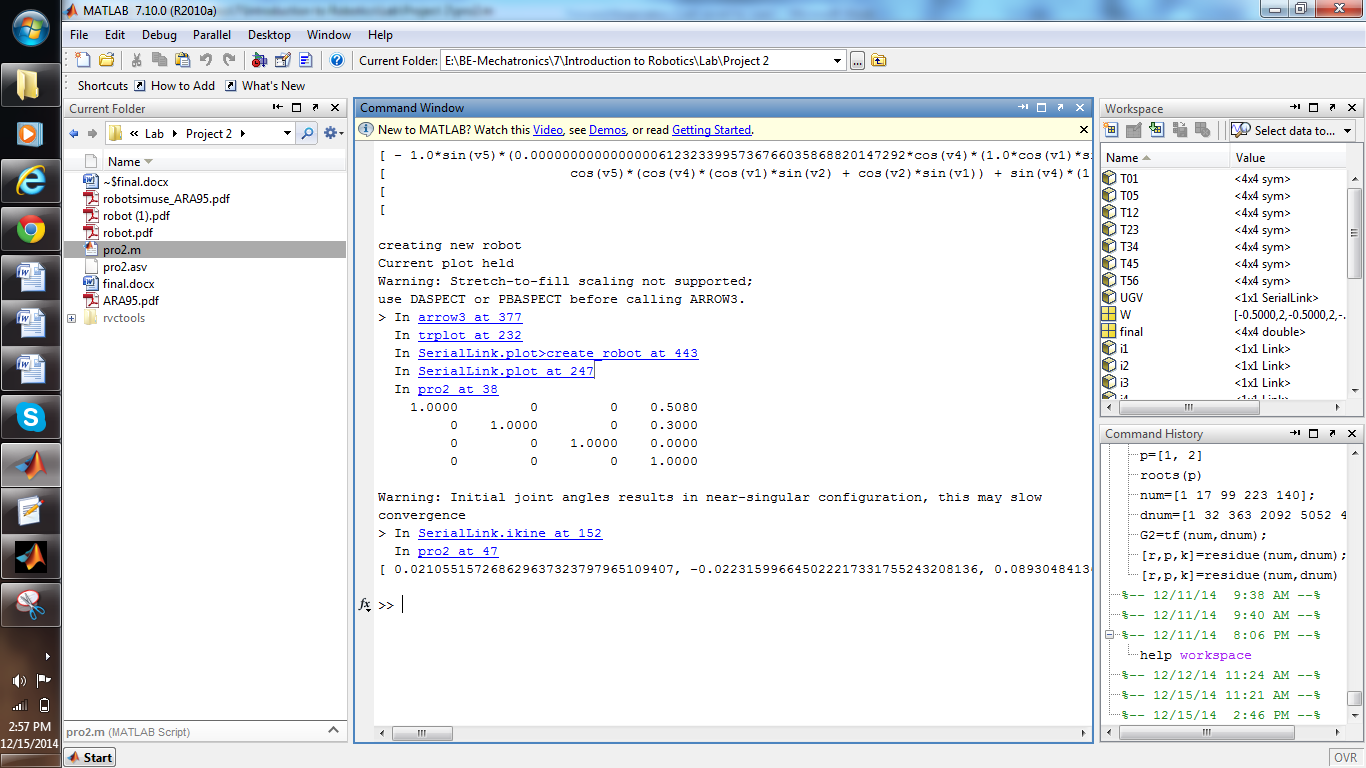
inv=ctraj(s, d, 50);

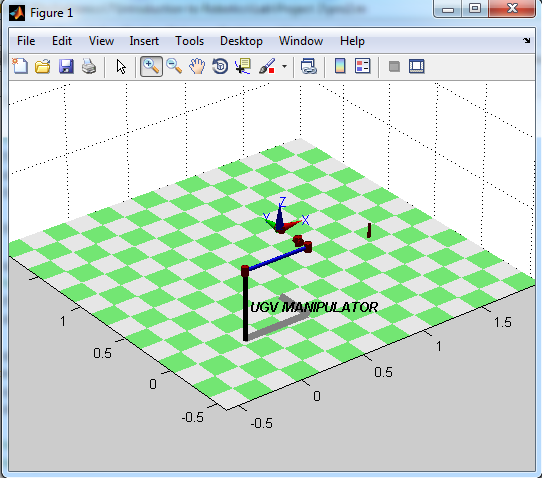
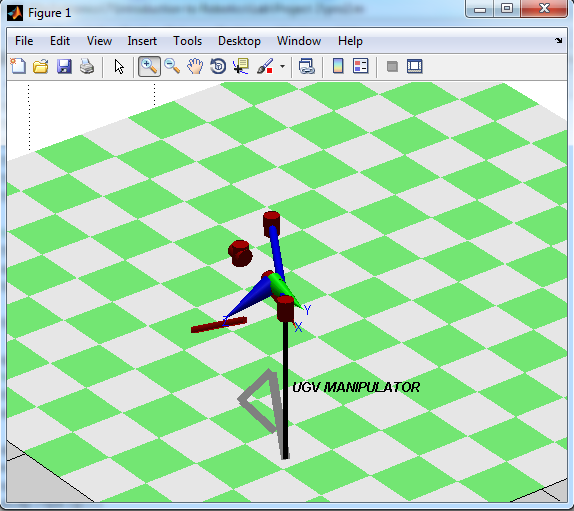
show=UGV.ikine(inv);

UGV.plot(show, 'workspace', W);

disp(v1);

**Robotic Simulations:-**



PS: The ctraj and jtraj functions were used to show the movement of the robot.

**PART 2: ROBOCIM**

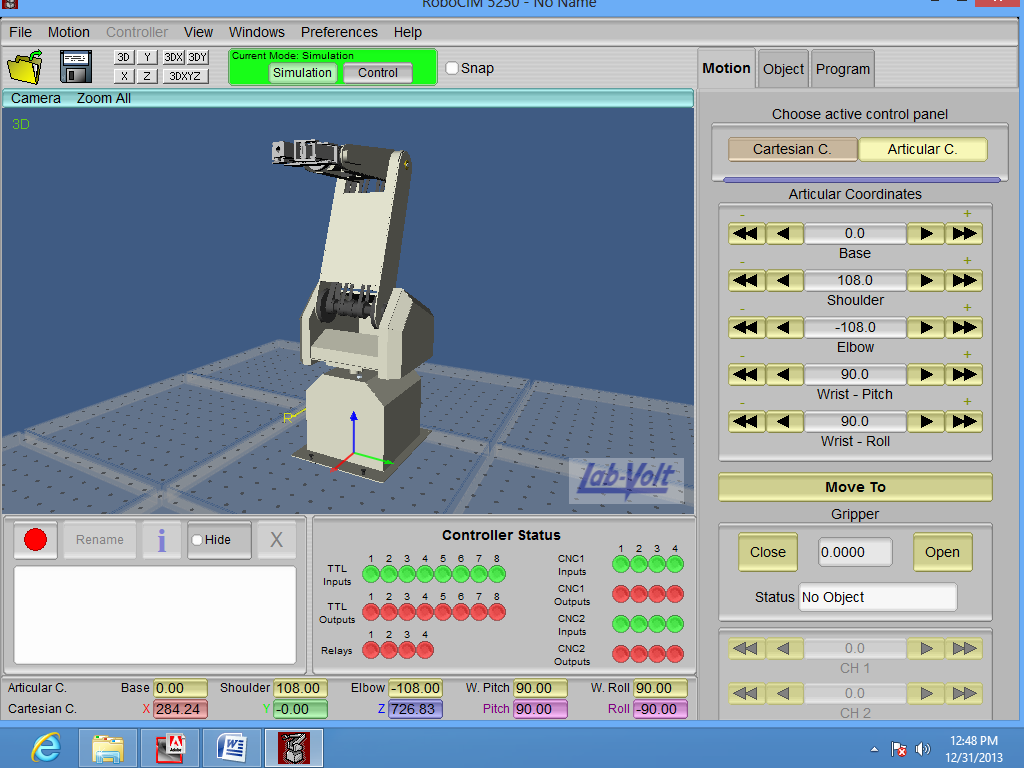
**Introduction:-**

The RoboCIM 5250 Software is used to simulate and control the operation of the Servo Robot System, Model 5250, and optional external devices such as Gravity Feeders, Belt Conveyors, or Linear Slides. One of the key features of the RoboCIM 5250 Software is that it simulates the actual equipment with three-dimensional representations. Sophisticated mathematical models accurately simulate the mechanical and electrical characteristics of the equipment. The RoboCIM 5250 Software allows users to interactively control and view the motion of the system. Programs can be created with the RoboCIM 5250 Software to control the equipment using either the text programming mode or the icon programming mode.



**Methodology & Simulations:-**

We can open the software by double clicking the RoboCIM icon and its default homescreen look like this.

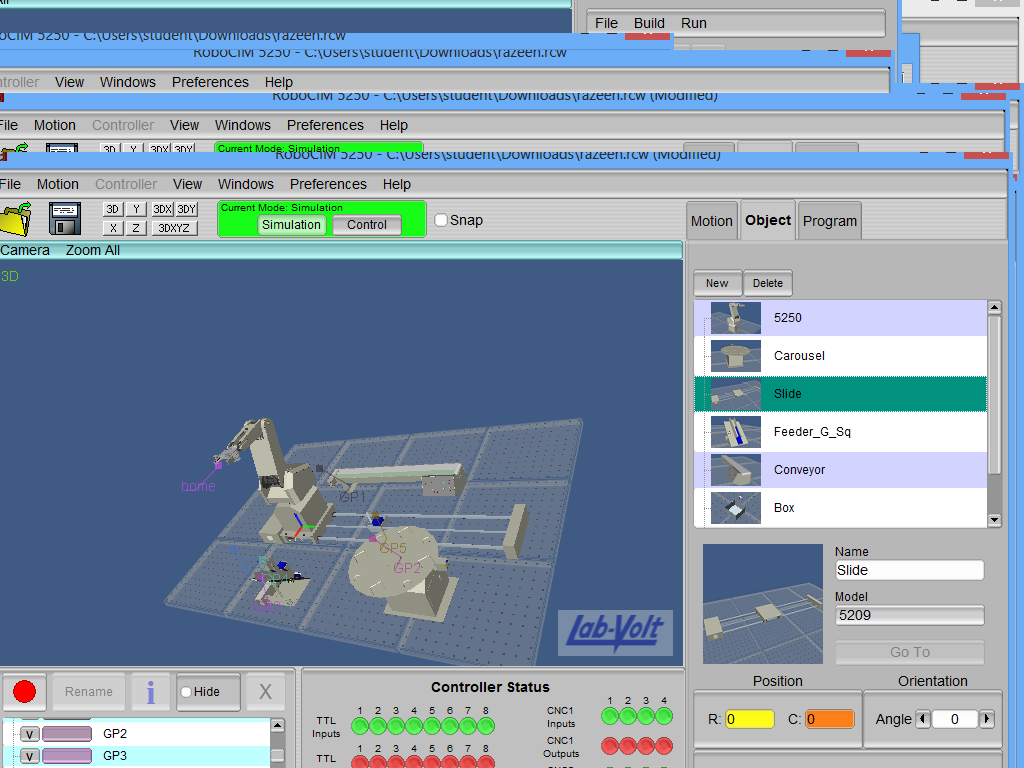


* The first thing that we needed was to place different objects that will be used in the environment:

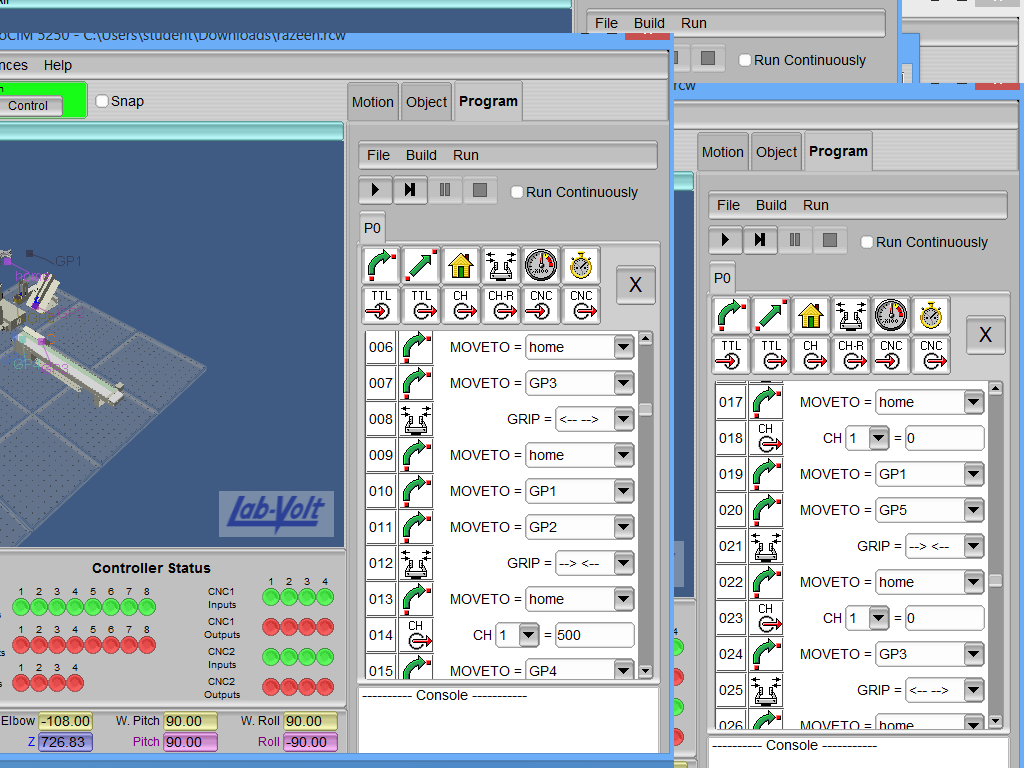
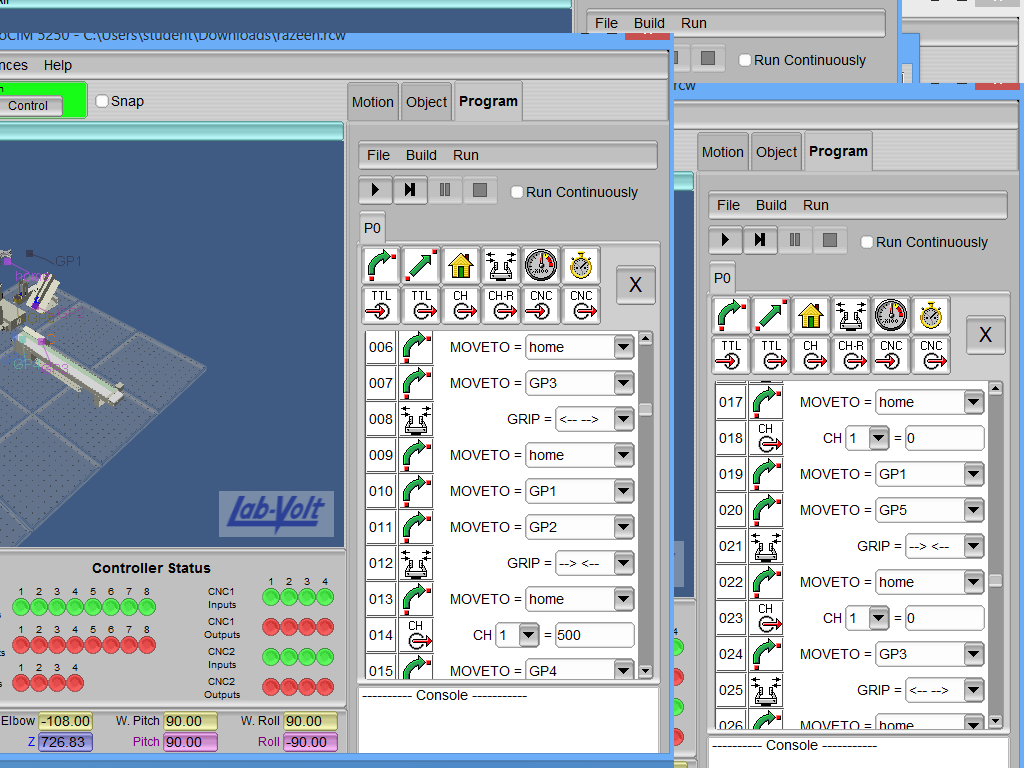


We had to design a work cell in which all the available lab volt equipment is utilized this equipment includes:

* Robot
* Linear Slide
* Conveyor belt
* Gravity feeder
* Rotary carousel
* Parts (Square /cylindrical)
* Then, needed to define the GP points in order to simulate the robot:-



* We used icon programming to program the software:

**Final Simulation:-**

