NYU TANDON SCHOOL OF ENGINEERING

ME GY-6933 Advance Mechatronics

TERM PROJECT PRESENTATION

Team Members:
Akshay Kumar (avk322)
Christian Lourido (cl4906)
Jordan Birnbaum (jbb498)

Instructor:
Prof Vikram Kapila

Date:
05/16/2019

NYU TANDON SCHOOL OF ENGINEERING

Goal Of the Project

- To build a miniature scale size automated construction crane
- Opensource platform for engineers willing to test and verify simulations and algorithms
- Implement a global vision system to assist user in defining goal coordinates

Objective of the Project

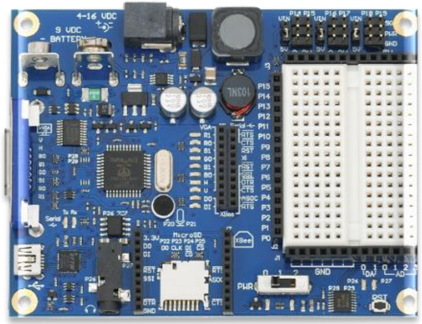
- The objectives for the project include:
  - To build a module with a scale size automated construction crane with its actuators
  - A defined workspace and a processing station
  - Develop a control algorithm using serial manipulator kinematics
  - Obtain goal coordinates through mouse clicking and marker placement
  - Manual operation to algorithms for autonomous operation

Design Iterations

Arduino Uno

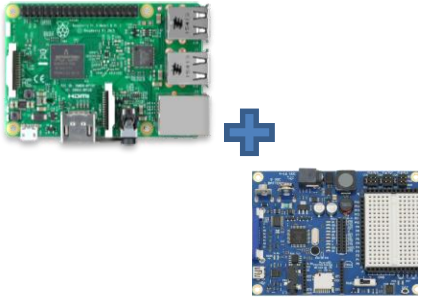
- Designing and fabricating the structure
- Controlling the cranes actuators manually using pots and buttons
- No feedback of any sort
- No user interface for controlling the crane

Propeller BoE



- Feedback for two of the actuators
- Forward kinematics code implemented
- A simple user interface for controlling
- More stable and rigid design parts added

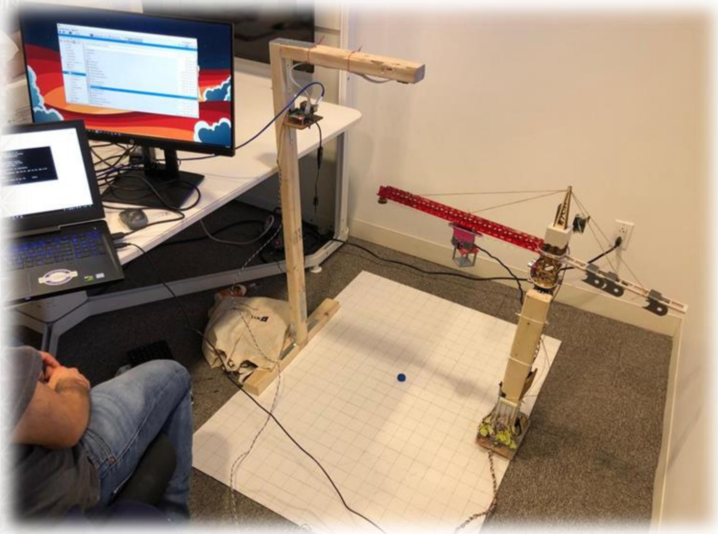
Raspberry pi 3 + Propeller BoE



- Use of camera to obtain the user input location for the crane to move
- Communication between Raspberry pi and Propeller
- Final and most robust mechanical design

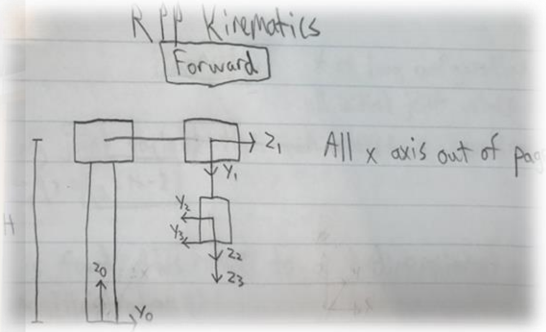
Mechanical Design

- Replacement and more robust design of mechanical weakness: the crane neck
- Replacement of carriage motor
- Redefined home pose
- Built camera stand



Forward Kinematics

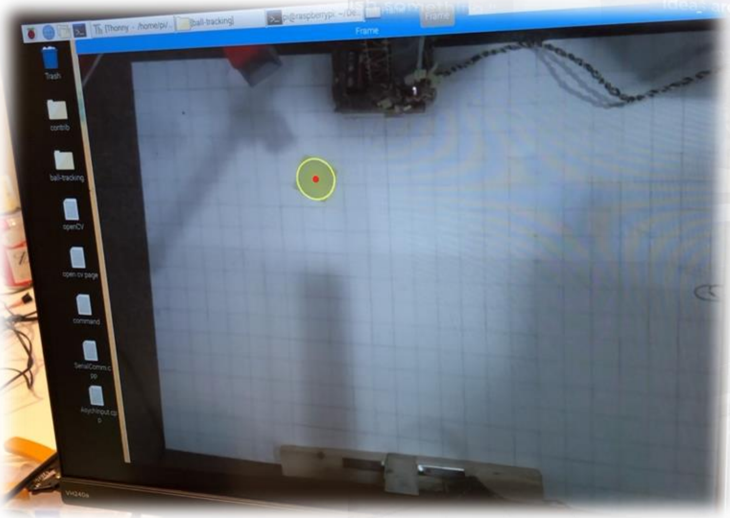
- Assigned frames using standard DH method
- Defined home configuration and obtained DH parameters
- Created DH matrices from DH parameters
- Obtained descriptor matrix from end effector to global frame

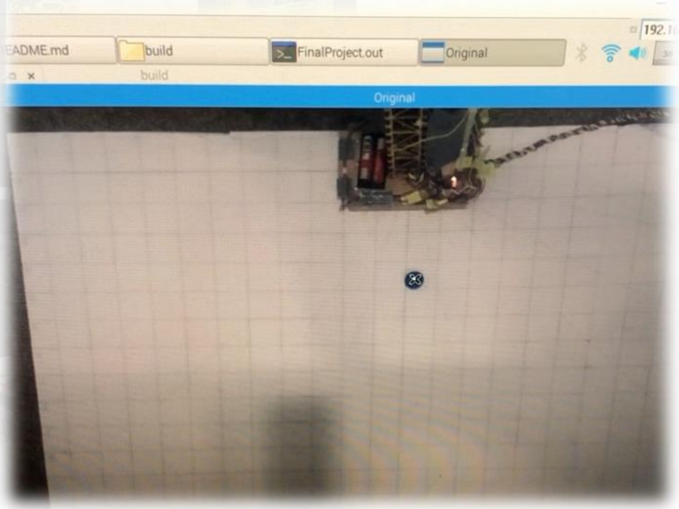


Joint #	Theta(i)	d(i)	a(i)	alpha(i)	q(i)
1	theta1=-90+q1	H	0	-90	q1
2	0	d2=15+q2	0	-90	q2
3	0	d3=9+q3	0	0	q3

Acquiring Coordinates

- User can obtain goal coordinates via three ways:
  - Manual input
  - Selecting the desired coordinates on the image provided by the pi cam
  - Placing a marker at the desired location – pi cam determines coordinates via a contour/ centroidal algorithm





NYU TANDON SCHOOL OF ENGINEERING

Methodology

Inverse Kinematics

Q3 is only joint variable that can affect z coordinate

$$Z=H-q$$

$$q3=H-z$$

Once z is found, this reduces to a Z dimensional problem (location of carriage)
q2 is only joint variables that can effect R distance ,
|R|=q2, solve for R

$$q2=\sqrt{(x^2+y^2)}$$

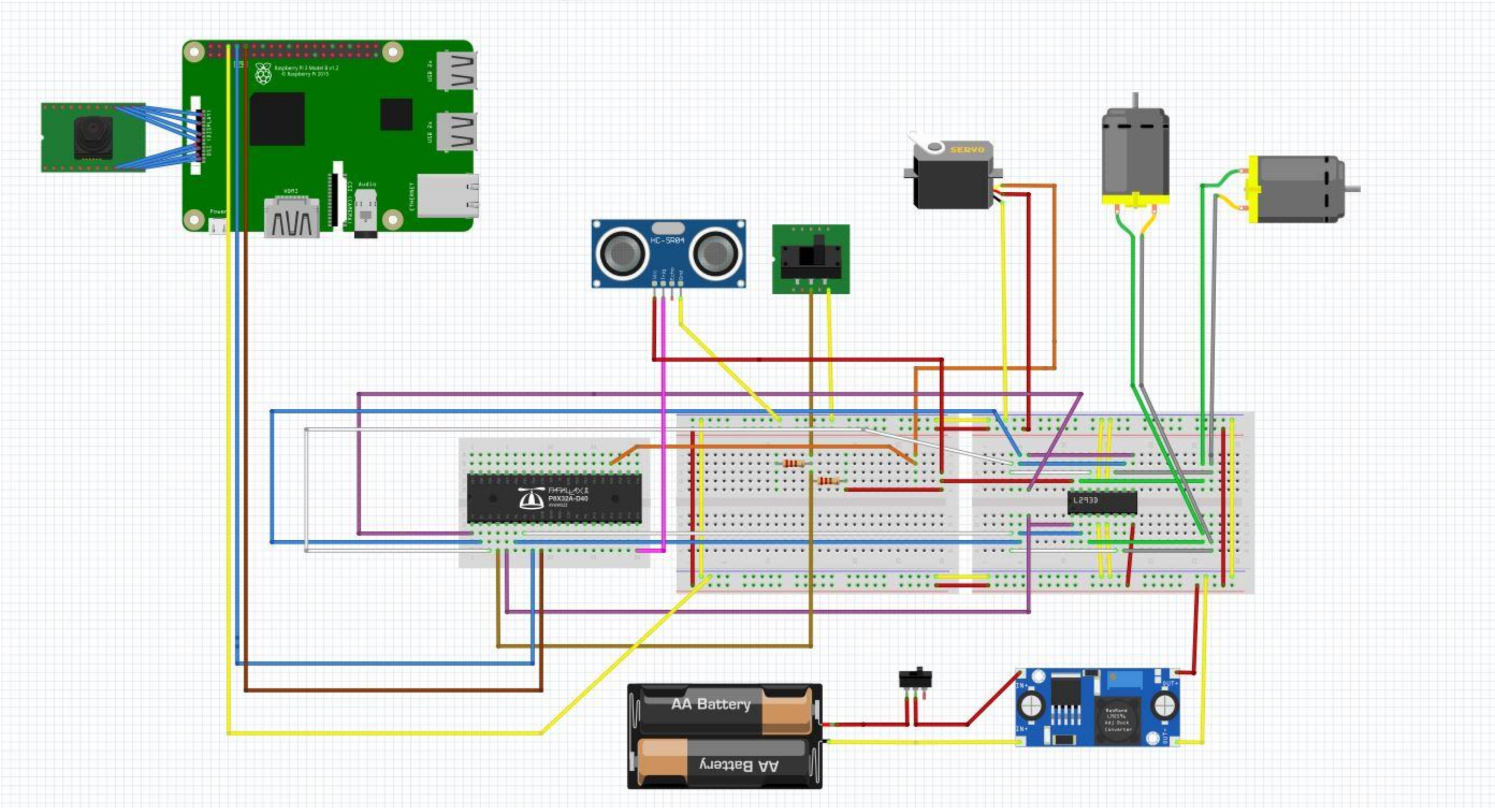
q1 is determined by taking the inverse tangent of the x and y values, the atan2 function is used to determine the precise q1 value.

$$\tan Q1=y/x$$

$$q1=\text{atan2}(y, x)$$

Note: At home configuration all initial joint values are zero

Electrical Circuit Design



Software used-*Fritzing*

Propeller Code

SimpleIDE D:\OneDrive\Documentos\SimpleIDE\FinalProject\Experiment_Servo_Carriage_IK.side

File Project Edit Tools Program Help

COM5

Project Manager

- Experiment_Servo_Carriage_IK.side
  - Experiment_Servo_Carriage_IK.c
  - HeaderFile.h
  - ServoMove.c
  - dcMove.c
  - dcMoveTimed.c

Experiment_Servo_Carriage_IK.c * | HeaderFile.h | ServoMove.c | dcMove.c | dcMoveTimed.c

```c
/*   Crane2.c

  ME GY-6933 Advanced Mechatronics
  Project-Propeller Based
  Instructor: Professor Vikram Kapila
  Date:04/18/2019
  Group Members: Christian Lourido(cl4906),Akshay Kumar Kutty(avk322), Jordan Birnbaum(jbb498)
*/

/*  Libraries */

#include "simpletools.h"
#include "ping.h"
#include "servo.h"
#include "HeaderFile.h"
#include <math.h>

/*Main Program*/

int main()
{
  int iuAngle; //Initial user angle (guess for q1) (user input)
  int q1new;
  int q2new;
  int q3new;
  int q1;
  int d2;
  int d3;
  int timeq3;
  char ui, ui2;
  double xnew,ynew,znew;
  int xu,yu,zu;

  cogstart(&pingfromcog, NULL, stack1, sizeof(stack1)); //Start ping sensor in Cog 1
  pause(500);
```

Project Options | Compiler | Linker

- [x] Math Lib
- [ ] Pthread Lib
- [x] Tiny Lib
- [ ] Create Project Library

Other Linker Options

Build Status

```
proploader.exe -r -I C:/Program Files (x86)/SimpleIDE/bin/../propeller-gcc/propeller-load/ -b activityboard -p COM5 cmm/Experiment_Servo_Carriage_IK.elf

Opening file 'cmm/Experiment_Servo_Carriage_IK.elf'Downloading file to port COM5
30176 bytes sent

Verifying RAM
Download successful!
```

Propeller Functions

Experiment_Servo_Carriage_IK.c | ServoMove.c

```c
void ServoMove(int pin, int iangle, int fangle,int delay){
  int fangleCorr= 5+fangle*(180.0/225.0);
  int iangleCorr= 5+iangle*(180.0/225.0);

  if (iangleCorr>fangleCorr){
    for(int n=iangleCorr; n>=fangleCorr;n--){
      servo_angle(pin,n*10);
      pause(delay);
    }
  }else if(iangleCorr<fangleCorr){
    for(int n=iangleCorr; n<=fangleCorr;n++){
      servo_angle(pin,n*10);
      pause(delay);
    }
  }else{
    }
  }
```

Experiment_Servo_Carriage_IK.c | dcMove.c

```c
void dcMove(int enPin, int dir1Pin, int dir2Pin, int direction, int onOff){

  if(onOff==0){
    low(dir1Pin); low(dir2Pin); low(enPin);
  }
  else{
    if(direction == 1){
      low(dir1Pin); high(dir2Pin); high(enPin);
    }else{
      high(dir1Pin); low(dir2Pin); high(enPin);
    }
  }
}
```

Experiment_Servo_Carriage_IK.c | dcMoveTimed.c

```c
void dcMoveTimed(int enPin, int dir1Pin, int dir2Pin, int direction, int time){

  if(direction == 1){
    low(dir1Pin); high(dir2Pin); high(enPin);
    pause(time);
    low(dir1Pin); low(dir2Pin); low(enPin);
  }else if(direction == 0){\
    high(dir1Pin); low(dir2Pin); high(enPin);
    pause(time);
    low(dir1Pin); low(dir2Pin); low(enPin);
    }

}
```

Propeller Functions

```
//Open serial communication
rpi = fdserial_open(9, 8, 0, 115200);
print("Run the executable file in the RPI. \nWating for RPI to send data...\n");
dprint(rpi, "\nHi from propeller!"); //To verify in RPI terminal

//Receive array of [20]
for(int i=0; i<20; i++){
  data_received[i]=fdserial_rxChar(rpi);
}
pause(1000);

//Data processing

/*Count digits until special character ":" is found. 1st value = X, 2nd value = Y */
int n=0, digitsx=0,digitsy=0;

while(data_received[n]!=':'){
  digitsx++;
  n++;
}
digitsx=n;
n++;

while(data_received[n]!=':'){
  digitsy++;
  n++;
}

/*Create arrays for X and Y based on the number of digits each has. Then, put data from buffer to new X and Y arrays*/
char x[digitsx-1],y[digitsy-1];
int m=0, o=0;

while(m<=digitsx-1){
  x[m]=data_received[m];
  m++;
}
m++;
```

Experimentation and Results

Inputs	I-click	II-click	III-click	Avg Diff.
Clicked	(25,30)	(25,30)	(25,30)	**-**
Comp.	(25,32)	(25,32)	(25,32)	**0.2**
Placed	(20,27.5)	(20,27.5)	(23,26)	**(4,3)**
Z-comp	5cm	5cm	5cm	**-**
Z-physical	7.5cm	7cm	7cm	**2.167**

I-click	II-click	III-click	Avg Diff.
(-20,15)	(-20,15)	(-20,15)	**-**
(-21,16)	(-21,16)	(-21,16)	**1.1**
(-17.5,15)	(-17.5,15)	(-17.5,15)	**(3,1.16)**
5cm	5cm	5cm	**-**
5cm	10cm	5.5cm	**1.83**

Experimentation and Results(Contd...)

Inputs	I-click	II-click	III-click	Avg Diff.
Clicked	(0,40)	(0,40)	(0,40)	-
Comp.	(0,42)	(0,42)	(0,42)	**0.2**
Placed	(0,36)	(0,36)	(0,35)	**(0,4.3)**
Z-comp	5cm	5cm	5cm	-
Z-physical	5cm	5cm	4cm	**0.33**

Avg. computational difference (x,y): (0.33,1,67) -1

Avg. physical difference (x,y): (2.33,2.83) -2.58

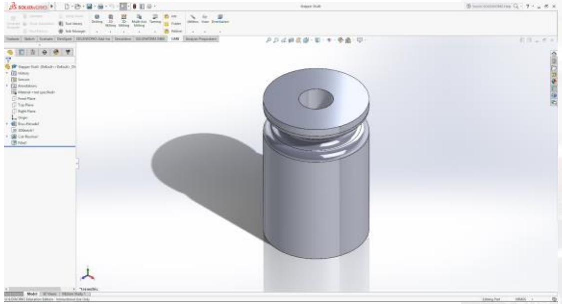
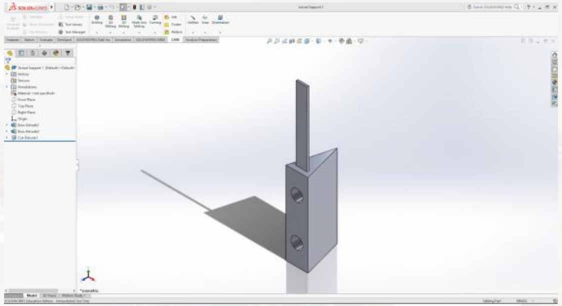
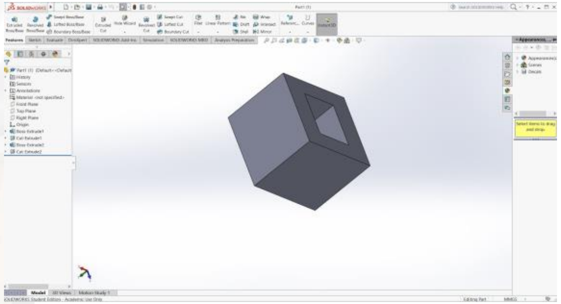
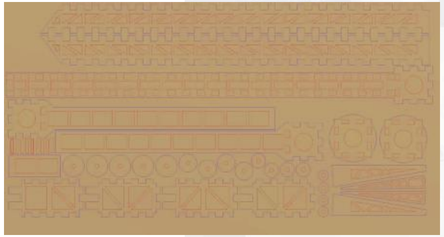
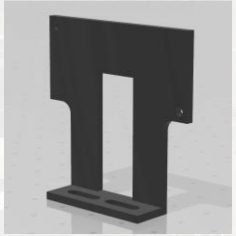
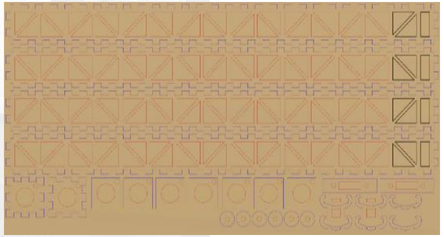
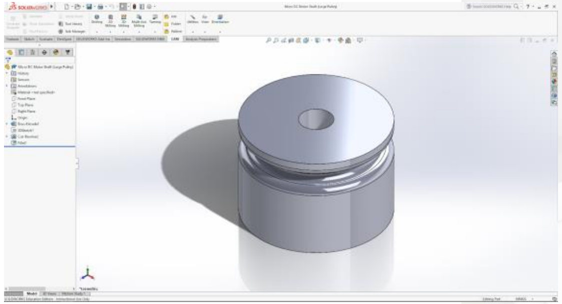
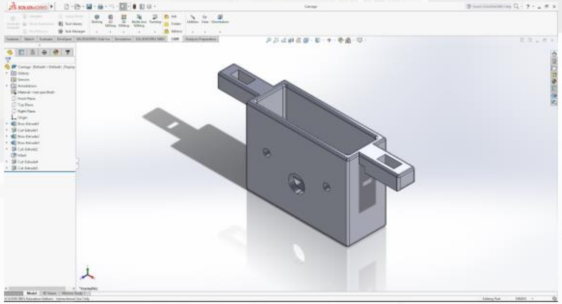
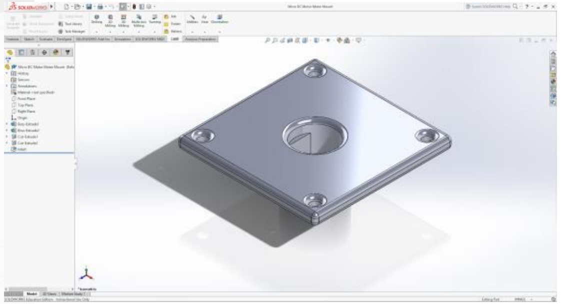
Avg. total Z difference : 1.44

Avg. spatial difference: 2.2

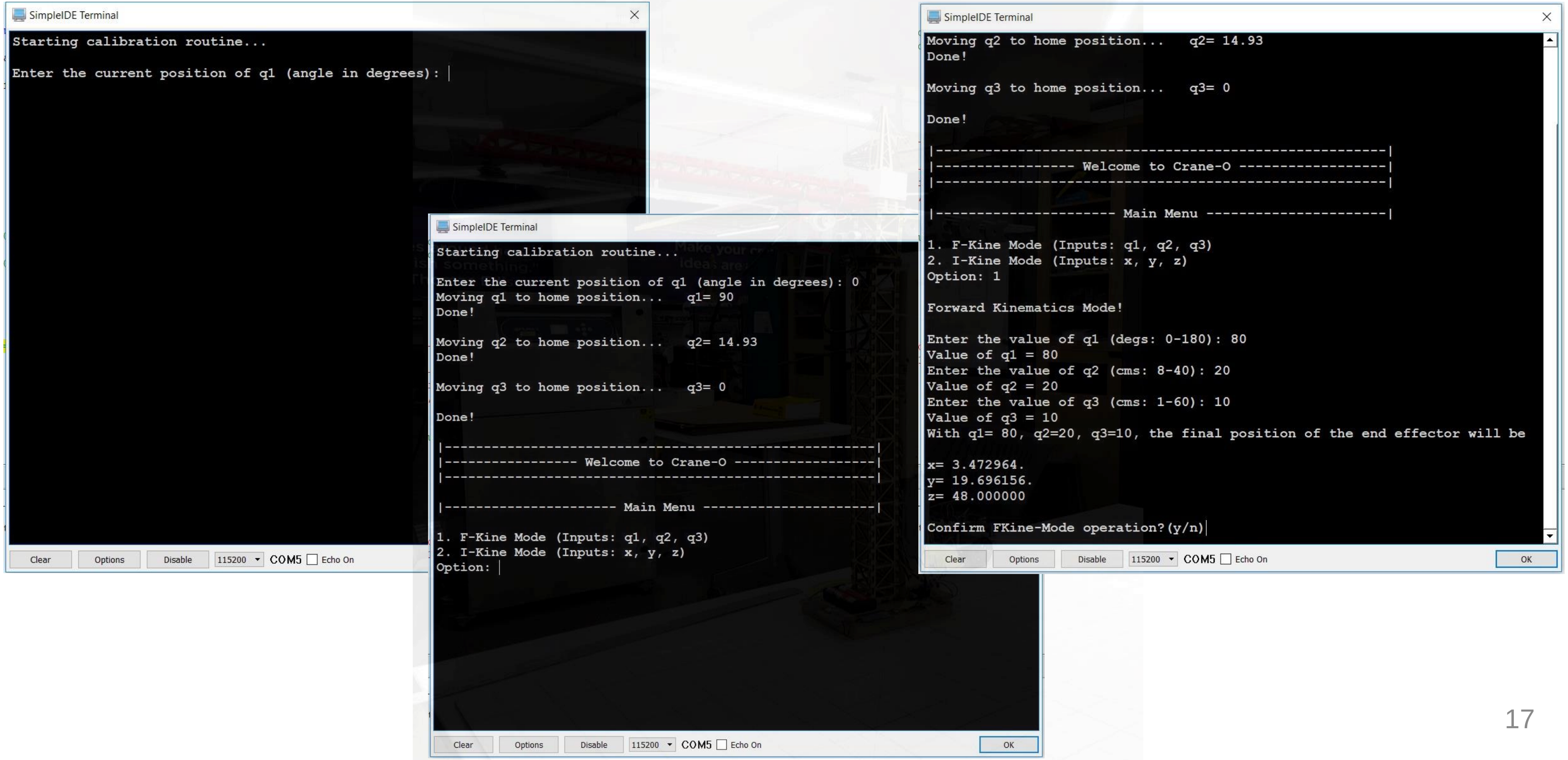
NYU | TANDON SCHOOL OF ENGINEERING

Thank You!

Appendix: Mechanical Solidworks Design

Appendix: User Terminal



Appendix: Experimentation and Results

q1=135 deg, **q2**= 30, **q3**=55

Inputs	X Dist	Y Dist	Z Dist	Avg Diff.
Comp Result	-21	21	-2	-
Physical Result	-18	23	5	2.67
	-18.5	22.2	6	1.57
	-18.5	22.5	3.5	3.83

q1=30 deg, **q2**= 35, **q3**=30

X Dist	Y Dist	Z Dist	Avg Diff.
30.3	17.5	28	-
23.5	18	30	4.13
29	20	31	1.5
26	19	31	2.67

q1=0 deg, **q2**= 25, **q3**=40

X Dist	Y Dist	Z Dist	Avg Diff.
25	0	18	-
22	2	22	1.83
26	3	22	2.67
26.5	3	21	3.67

Average Difference for all the trials:- ***x***:2.88, ***y***:1.91, ***z***:3.39

Appendix: Cost Structure of the project

Bills Of Materials(Prototype)			
Qty	**Components**	**Price**	**Total**
1	Raspberry Pi 3 B+	$ 30	$ 30
1	Raspberry Pi Cam V2.1	$18	$18
1	Servo Motor	$12	$12
1	Dc Motor for the carriage	$20	$20
1	Dc Motor for the gripper	$18	$18
1	Limit Switch	$4	$4
1	Voltage Regulator	$9	$9
2	Acrylic Sheet (24*12)	$15	$30
1	Motor Driver-L293D	$ 5	$ 5
4	Birch Wood (24*12)	$15	$60
1	Gorilla glue	$10	$10
1	Wood beams	$9	$9
1	Corner and L-clamps	$14	$14
1	Ping Sensor	$30	$30
1	Battery 7.2V	$17	$17
1	Breadboard	$6	$6
1	Wires	Free	Free
1	Fishing Ropes and Tension Wire	$11	$11
1	Miscelleneous	$ 30	$ 30
	TOTAL AMOUNT	**$273**	**$343**

This is the overall cost to build and fabricate the tower crane. The cost of building can be reduced if mass produced for educational purposes.