RoboArm
Project Report #2

November 23, 2020

Justin Tam 300026741
Lucas Rahn 300021873

Forward Kinematic Transformation Matrix

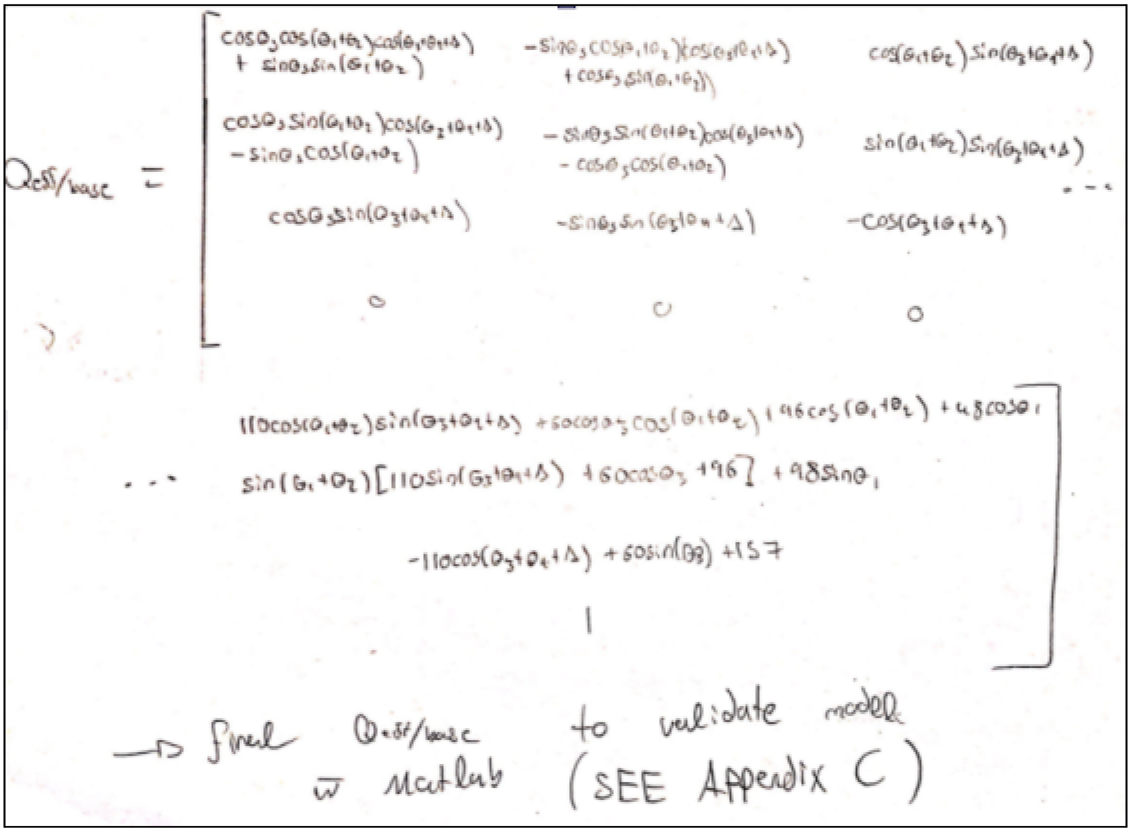


Figure 1: Simplified Q_eff/base Matrix

The final expression for the forward kinematic as obtained from Lab 1, is rewritten for readability

Column 1

$$[cos(\theta_5)cos(\theta_1+\theta_2)cos(\theta_3+\theta_4+\Delta)+sin(\theta_5)sin(\theta_1+\theta_2)$$

$$[cos(\theta_5)sin(\theta_1+\theta_2)cos(\theta_3+\theta_4+\Delta)-sin(\theta_5)cos(\theta_1+\theta_2) \quad ...$$

$$[cos(\theta_5)sin(\theta_3+\theta_4+\Delta)$$

$$[0$$

Column 2

$$-sin(\theta_5)cos(\theta_1+\theta_2)cos(\theta_3+\theta_4+\Delta)+cos(\theta_5)sin(\theta_1+\theta_2)$$

$$...\quad -sin(\theta_5)sin(\theta_1+\theta_2)cos(\theta_3+\theta_4+\Delta)-cos(\theta_5)cos(\theta_1+\theta_2) \quad ...$$

$$-sin(\theta_5)sin(\theta_3+\theta_4+\Delta)$$

$$0$$

Column 3

$$\begin{gathered} cos(\theta_1 + \theta_2)sin(\theta_3 + \theta_4 + \Delta) \\ \ldots \quad sin(\theta_1 + \theta_2)sin(\theta_3 + \theta_4 + \Delta) \quad \ldots \\ -\, cos(\theta_3 + \theta_4 + \Delta) \\ 0 \end{gathered}$$

Column 4

$$\begin{gathered} cos(\theta_1 + \theta_2)[110sin(\theta_3 + \theta_4 + \Delta) + 60cos(\theta_3) + 96] + 98cos(\theta_1)] \\ \ldots \quad sin(\theta_1 + \theta_2)[110sin(\theta_3 + \theta_4 + \Delta) + 60cos(\theta_3) + 96] + 98sin(\theta_1)] \\ -\, 110cos(\theta_3 + \theta_4 + \Delta) + 60sin(\theta_3) + 157] \\ 1] \end{gathered}$$

Inverse Kinematic Equation Derivation

The values of each joint angles ($\theta_1, \theta_2,$ $etc.$) were calculated assuming that the values within the forward kinematic matrix are known. These values can be calculated by inputting the value of each joint angle into the forward kinematic matrix defined above, and the reason we develop the model this way is discussed in the discussion section. Knowing the values in the forward model, we define each value by a variable which is depicted in Figure 2. The variables are used in developing the inverse kinematic equations.

$$\begin{bmatrix} x_x & y_x & z_x & x_T \\ x_y & y_y & z_y & y_T \\ x_z & y_z & z_z & z_T \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Figure 2: Forward Matrix Parameter Definition

Solving for θ_1:

Using equations from the Simplified Q_{eff}/Q_{base} Matrix column 3:

$$cos(\theta_1 + \theta_2)sin(\theta_3 + \theta_4 + \Delta) = z_x$$

$$sin(\theta_1 + \theta_2)sin(\theta_3 + \theta_4 + \Delta) = z_y$$

$$tan(\theta_1 + \theta_2) = \frac{z_x}{z_y}\frac{sin(\theta_3+\theta_4+\Delta)}{sin(\theta_3+\theta_4+\Delta)} \quad \text{eq. 1}$$

Using equations from the Simplified Q_{eff}/Q_{base} Matrix column 4:

$$cos(\theta_1 + \theta_2)[110sin(\theta_3 + \theta_4 + \Delta) + 60cos(\theta_3) + 96] + 98cos(\theta_1) = x_T$$

$$sin(\theta_1 + \theta_2)[110sin(\theta_3 + \theta_4 + \Delta) + 60cos(\theta_3) + 96] + 98sin(\theta_1) = y_T$$

Substitute eq. 1 into above:

$$\frac{z_x}{z_y} = \frac{98sin(\theta_1) - y_T}{98cos(\theta_1) - x_T}$$

$$98z_y cos(\theta_1) - 98z_x sin(\theta_1) = x_T z_y - y_T z_x$$

Define:

$$U = 98z_y \qquad V = -98z_x \qquad W = x_T z_y - y_T z_x$$

Then as derived in class:

$$\theta_1 = 2 * atan2(-V \pm \sqrt{V^2 + (U+W)*(U-W)}, (-U-W))$$

* 2 Solutions for θ_1*

Solving for θ_2:

Using equation 1:

$$\theta_2 = atan2(z_x sin(\theta_3 + \theta_4 + \Delta), z_y sin(\theta_3 + \theta_4 + \Delta)) - \theta_1$$

Since $sin(\theta_3 + \theta_4 + \Delta)$ term can be positive or negative our equation becomes:

$$\theta_2 = atan2(\pm z_x, \pm z_y) - \theta_1$$

* 4 Solutions for θ_2*

Solving for θ_3:

Using equations from the Simplified Q_{eff}/Q_{base} Matrix column 4:

$$cos(\theta_1 + \theta_2)[110sin(\theta_3 + \theta_4 + \Delta) + 60cos(\theta_3) + 96] + 98cos(\theta_1) = x_T$$

$$sin(\theta_1 + \theta_2)[110sin(\theta_3 + \theta_4 + \Delta) + 60cos(\theta_3) + 96] + 98sin(\theta_1) = y_T$$

Define:

$$\gamma = 110sin(\theta_3 + \theta_4 + \Delta) + 60cos(\theta_3) + 96 = \frac{x_T - 98cos(\theta_1)}{cos(\theta_1 + \theta_2)} \text{ for } \theta_1 + \theta_2 \neq n\pi + \frac{\pi}{2}$$

$$\gamma = \frac{y_T - 98sin(\theta_1)}{sin(\theta_1 + \theta_2)} for\ \theta_1 + \theta_2 = n\pi + \frac{\pi}{2}$$

Using equations from the Simplified Q_{eff}/Q_{base} Matrix column 4:

$$-110cos(\theta_3 + \theta_4 + \Delta) + 60sin(\theta_3) + 157 = z_T$$

$$[110sin(\theta_3 + \theta_4 + \Delta)]^2 = [\gamma - 60cos(\theta_3) - 96]^2$$

$$[-110cos(\theta_3 + \theta_4 + \Delta)]^2 = [z_T - 60sin(\theta_3) - 157]^2$$

Adding above two equations:

$$110^2 = [z_T - 60sin(\theta_3) - 157]^2 + [\gamma - 60cos(\theta_3) - 96]^2$$

$$cos(\theta_3)(11520 - 120\gamma) + sin(\theta_3)(18840 - 120z_T) = -\gamma^2 + 192\gamma - z_T^2 + 314z_T - 25365$$

Define:

$$A = 11520 - 120\gamma \quad B = 18840 - 120z_T \quad C = -\gamma^2 + 192\gamma - z_T^2 + 314z_T - 25365$$

Again as derived in class, or the same procedure as for solving θ_1:

$$\theta_3 = 2 * atan2(-B \pm \sqrt{B^2 + (A + C) * (A - C)}, (-A - C))$$

* 8 Solutions for θ_3 *

Solving for θ_4:

Using equations from the Simplified Q_{eff}/Q_{base} Matrix column 3:

$$cos(\theta_1 + \theta_2)sin(\theta_3 + \theta_4 + \Delta) = z_x$$

$$sin(\theta_3 + \theta_4 + \Delta) = \frac{z_x}{cos(\theta_1 + \theta_2)}$$

$$cos(\theta_3 + \theta_4 + \Delta) = -z_z$$

$$tan(\theta_3 + \theta_4 + \Delta) = \frac{\frac{z_x}{cos(\theta_1 + \theta_2)}}{-z_z}$$

$$\theta_4 = atan2(\frac{z_x}{cos(\theta_1 + \theta_2)}, -z_z) - \theta_3 - \Delta \quad for\ \theta_1 + \theta_2 \neq n\pi + \frac{\pi}{2}$$

Similarly:

$$\theta_4 = atan2\left(\frac{z_y}{sin(\theta_1+\theta_2)}, -z_z\right) - \theta_3 - \Delta \ for\ \theta_1 + \theta_2 = n\pi + \frac{\pi}{2}$$

* 8 Solutions for θ_4*

Solving for θ_5:

Using equations from the Simplified Q_{eff}/Q_{base} Matrix row 3:

$$-sin(\theta_5)sin(\theta_3 + \theta_4 + \Delta) = y_z$$

$$cos(\theta_5)sin(\theta_3 + \theta_4 + \Delta) = x_z$$

$$sin(\theta_5) = \frac{-y_z}{sin(\theta_3+\theta_4+\Delta)}$$

$$cos(\theta_5) = \frac{x_z}{sin(\theta_3+\theta_4+\Delta)}$$

$$\theta_5 = atan2\left(\frac{-y_z}{sin(\theta_3+\theta_4+\Delta)}, \frac{x_z}{sin(\theta_3+\theta_4+\Delta)}\right)$$

* 8 Solutions for θ_5*

Final equations:

$$\boldsymbol{\theta_1} = 2 * atan2(-V \pm \sqrt{V^2 + (U+W)*(U-W)}, (-U-W))$$

$$\boldsymbol{\theta_2} = atan2(\pm z_x, \pm z_y) - \theta_1$$

$$\boldsymbol{\theta_3} = 2 * atan2(-B \pm \sqrt{B^2 + (A+C)*(A-C)}, (-A-C))$$

$$\boldsymbol{\theta_4} = atan2\left(\frac{z_x}{cos(\theta_1+\theta_2)}, -z_z\right) - \theta_3 - \Delta \ for\ \theta_1 + \theta_2 \neq n\pi$$

Or

$$\boldsymbol{\theta_4} = atan2\left(\frac{z_y}{sin(\theta_1+\theta_2)}, -z_z\right) - \theta_3 - \Delta \ for\ \theta_1 + \theta_2 = n\pi$$

$$\boldsymbol{\theta_5} = atan2\left(\frac{-y_z}{sin(\theta_3+\theta_4+\Delta)}, \frac{x_z}{sin(\theta_3+\theta_4+\Delta)}\right)$$

For

$$U = 98z_y \qquad V = -98z_x \qquad W = x_T z_y - y_T z_x$$

$$\gamma = \frac{x_T - 98cos(\theta_1)}{cos(\theta_1+\theta_2)} \ for\ \theta_1 + \theta_2 \neq n\pi + \frac{\pi}{2}$$

Or

$$\gamma = \frac{y_T - 98sin(\theta_1)}{sin(\theta_1 + \theta_2)} \textit{for } \theta_1 + \theta_2 = n\pi + \frac{\pi}{2}$$

$$A = 11520 - 120\gamma \quad B = 18840 - 120z_T \quad C = -\gamma^2 + 192\gamma - z_T^2 + 314z_T - 25365$$

Demonstration

To demonstrate our inverse kinematic equations we use the forward model to generate a set of values that can be used for kinematic equations. Figure 3 - Figure 8 shows first, the values that are generated by the forward model defined in the first section of this document, or Lab 1. The forward model is generated by using the joint angles inputted, which is depicted in the second part of Figures 3-8 (Theta1F, Theta2F, …, Theta5F). Finally, the inverse kinematic equations defined above are used to obtain the values depicted in the third part of Figures 3-8 (Theta1I, Theta2I, …, Theta5I). The implementation of the inverse kinematic equations in Matlab code can be found in Appendix A.

These final values of the joint angles are used on the physical robot which can be seen in Appendix B (only Figures 5-8 are tested on the physical implementation). We can measure the coordinates from the physical implementation, and compare them to the coordinates used in the inverse kinematic calculation (x_T, y_T, z_T), but we refrain from doing this step since our values obtained from the inverse kinematic calculation are exactly the same as those used in the calculation of the forward model, and error calculations for this were already done in Lab 1.

```
    0.0349    0.9994         0   92.0899
    0.9842   -0.0344    0.1736  194.5819
    0.1735   -0.0061   -0.9848   10.1039
         0         0         0    1.0000

| Theta1F | Theta2F | Theta3F | Theta4F | Theta5F |
|    20   |    70   |   -40   |   -40   |    2    |
| Theta1I | Theta2I | Theta3I | Theta4I | Theta5I |
  20.0000   70.0000  -40.0000  -40.0000    2.0000
```

Figure 3: Inverse Kinematic Matlab Calculations for x_T=92.09 y_T=194.58 z_T=10.10

```
    0.3140    0.0611    0.9474  315.5847
   -0.9469    0.0932    0.3078   12.4160
   -0.0695   -0.9938    0.0872  198.3823
         0         0         0    1.0000

| Theta1F | Theta2F | Theta3F | Theta4F | Theta5F |
|    -43   |    61   |    32   |    -27   |    94   |
| Theta1I | Theta2I | Theta3I | Theta4I | Theta5I |
  -43.0000   61.0000   32.0000  -27.0000   94.0000
```

Figure 4: Inverse Kinematic Matlab Calculations for x_T=315.58 y_T=12.42 z_T=198.38

```
     0     0     1   364
     0    -1     0     0
     1     0     0   157
     0     0     0     1

| Theta1F | Theta2F | Theta3F | Theta4F | Theta5F |
|    0   |    0   |    0   |    0   |    0   |
| Theta1I | Theta2I | Theta3I | Theta4I | Theta5I |
     0     0     0     0     0
     0     0     0     0     0
```

Figure 5: Inverse Kinematic Matlab Calculations for x_T=364 y_T=0 z_T=157

```
| Theta1F | Theta2F | Theta3F | Theta4F | Theta5F |
|    -20   |    0   |    -11   |    -79   |    0   |
| Theta1I | Theta2I | Theta3I | Theta4I | Theta5I |
Undefined function or variable 'EEO'.

Error in Untitled (line 117)
disp(EEO);

>> Untitled
    0.9395   -0.3420    0.0164  239.6294
   -0.3420   -0.9397   -0.0060  -87.2180
    0.0175         0   -0.9998   36.5979
         0         0         0    1.0000

| Theta1F | Theta2F | Theta3F | Theta4F | Theta5F |
|    -20   |    0   |    -10   |    -79   |    0   |
| Theta1I | Theta2I | Theta3I | Theta4I | Theta5I |
  -20.0000   -0.0000  -10.0000  -79.0000         0
```

Figure 6: Inverse Kinematic Matlab Calculations for x_T=239.63 y_T= -87.22 z_T=36.60

```
    0.0905    0.3046    0.9482  260.3451
   -0.7916   -0.5558    0.2541  163.1510
    0.6044   -0.7735    0.1908  147.0867
         0         0         0    1.0000

| Theta1F | Theta2F | Theta3F | Theta4F | Theta5F |
|    82   |    -67   |    -31   |    42   |    52   |
| Theta1I | Theta2I | Theta3I | Theta4I | Theta5I |
   82.0000  -67.0000  -31.0000   42.0000   52.0000
```

Figure 7: Inverse Kinematic Matlab Calculations for x_T=260.35 y_T=163.15 z_T=147.08

```
    0.0156    0.4540    0.8909  269.6986
   -0.0079    0.8910   -0.4539 -212.4299
   -0.9998         0    0.0175  169.3387
         0         0         0    1.0000

| Theta1F | Theta2F | Theta3F | Theta4F | Theta5F |
|    -70   |    43   |    10   |    -9   |    180   |
| Theta1I | Theta2I | Theta3I | Theta4I | Theta5I |
  -70.0000   43.0000   10.0000   -9.0000  180.0000
```

Figure 8: Inverse Kinematic Matlab Calculations for x_T=269.69 y_T= -212.43 z_T=169.34

Discussion

The forward kinematic model can be expressed in terms of three translation variables and three rotational variables (x_T, y_T, z_T, θ, φ, ψ) which is exactly expressed in Figure 9. The goal when using the inverse kinematic equations is that we wish to use these six variables to describe a location and angle of the end effector and in turn, the equations will produce an angle for each of the joints such that the end effector satisfies the location and rotation provided. Not all angles of the end effector can be satisfied when providing a location for the end effector. Because of this, a simple solution is to retrieve the location of the end effector from a pre-evaluated forward kinematics model, which is simply the last column of the forward matrix. The angle of the end effector could be extracted by comparing the equations in Figure 9 to the values in the computed forward matrix, but if we only care about the location of the end effector this would not matter. In turn this allows us to derive the inverse model in terms of the variables given in Figure 2. In future experiments, when we are looking to pick up objects, it might be ideal to have the end effector completely pointing down, in which case we would want to make φ or ψequal to 180° to have the end effector pointing down, which would greatly reduce the complexity of the equations to find the allowed angles of the end effector.

The inverse kinematic equations were developed from the forward kinematic matrix in terms of the values described above. We effectively used trigonometric reductions and substitutions of mainly column three and four to get our solutions. When initially developing the inverse kinematic equations we were unable to get correct values for some locations because of the incorrect use of the arctan function. We had decided to use the atan2 function that matlab provides to ensure that the full trigonometric circle was being used. Although, when using it we found that it was important to be precise about where the negative signs were placed, as well as to not simply cancel out a division of negative signs within the arctan function. This would result in an improper use of the atan2 function in some instances. One examples of this can be seen when deriving θ_2. Using equation 1, we normally would cancel out the$sin(\theta_3 + \theta_4 + \Delta)$ terms within the arctan, but this would be neglecting when the sin terms are negative which returns a different value than when the sin terms are positive. To overcome this, it requires doubling the number of solutions produced but we make it so the terms within the arctan function are plus or minus.

One of the mathematical restraints we have is that $\theta_3 + \theta_4 + \Delta \neq n\pi$. The reason for this is many of our equations depend on sin of that value and any multiple of $n\pi$would return 0 rendering the equations not usable. This can be seen in Figure 6, here $\Delta = \pi/2$ and the sum of $\theta_3 + \theta_4$are $\pi/2$, so we get an error. The solution is to simply move the location of the end effector, or in our case of predefining the forward kinematic matrix, slightly changing one of the angles. Here we change $\theta_3$to 10 degrees from 11 degrees, but it could be changed by even less than one degree. Initially we had another mathematical constraint that $\theta_1 + \theta_2 \neq n\pi + \frac{\pi}{2}$since cos of that value would return 0. We overcame this issue by simply using one of the other similar equations that had the sum of the two angles in terms of sin instead, when the inequality was true. Use of this can be seen in the calculation of gamma and θ_4.

$$A_1A_2\ldots A_n = \begin{bmatrix} \cos\theta\cos\varphi & \cos\theta\sin\varphi\sin\psi - \sin\theta\cos\psi & \cos\theta\sin\varphi\cos\psi + \sin\theta\sin\psi & x_T \\ \sin\theta\cos\varphi & \sin\theta\sin\varphi\sin\psi + \cos\theta\cos\psi & \sin\theta\sin\varphi\cos\psi - \cos\theta\sin\psi & y_T \\ -\sin\varphi & \cos\varphi\sin\psi & \cos\varphi\cos\psi & z_T \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Figure 9: Forward Kinematic Matrix in terms of end effector parameters

Appendix A

Implementation of the inverse kinematic model to get joint angles from the forward kinematics matrix in Matlab code

```
clear
%\\\\\\\\\\\\\\\\\\\\\\\\\\\\\Forward Model\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
%---------------------Theta Values-----------------------------------------
Theta=[20 70 -40 -40 2];
delta = 90;
%--------------------Calculate Q-------------------------------------------
a = Theta(3)+Theta(4)+delta; %arbitrary constant for simplicity
b = Theta(1)+Theta(2);
c = Theta(5);
Q = [cosd(c)*cosd(b)*cosd(a)+sind(c)*sind(b) -sind(c)*cosd(b)*cosd(a)+...
    cosd(c)*sind(b) cosd(b)*sind(a) cosd(b)*(110*sind(a)+60*...
    cosd(Theta(3))+96)+98*cosd(Theta(1));
    cosd(c)*sind(b)*cosd(a)-sind(c)*cosd(b) -sind(c)*sind(b)*cosd(a)-...
    cosd(c)*cosd(b) sind(b)*sind(a) sind(b)*(110*sind(a)+60*...
    cosd(Theta(3))+96)+98*sind(Theta(1));
    cosd(c)*sind(a) -sind(c)*sind(a) -cosd(a) -110*cosd(a)+...
    60*sind(Theta(3))+157;
    0 0 0 1];
disp(Q);
disp('| Theta1F | Theta2F | Theta3F | Theta4F | Theta5F |');
fprintf(['|    %d   |    %d   |    %d   |    %d   |    %d   |\n']...
    ,Theta(1),Theta(2),Theta(3),Theta(4),Theta(5));
%\\\\\\\\\\\\\\\\\\\\\\\\\\\\\Inverse Model\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
%------------------------Current Bounds------------------------------------
%sind(theta3+theta4+delta) != 90 Deg +-180
%---------------------End Effector Values----------------------------------
%{
Q=[0 0.7071 -0.7071 -118.7939;
    0 0.7071 0.7071 257.3869;
  1.0000 0 0 157.0000;
  0 0 0 1];
delta=90;
%}
%--------------------Define Output Matrix----------------------------------
%Using the given equations we expect 8 mathematically possible solutions.
%To organize this, we create a 8x10 matrix where the rows represent
%seperate solutions,and the columns represent various values that are
%either ouputs, values helpful in calculations, or indicators of nonviable
%solutions. These are defined as such:
%\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
%Columns: |theta1|theta2|theta3|theta4|theta5|gamma|A|B|C|rt|
%         |      |      |      |      |      |     | | | |  |
%         |      |      |      |      |      |     | | | |  |
%         |      |      |      |      |      |     | | | |  |...
```

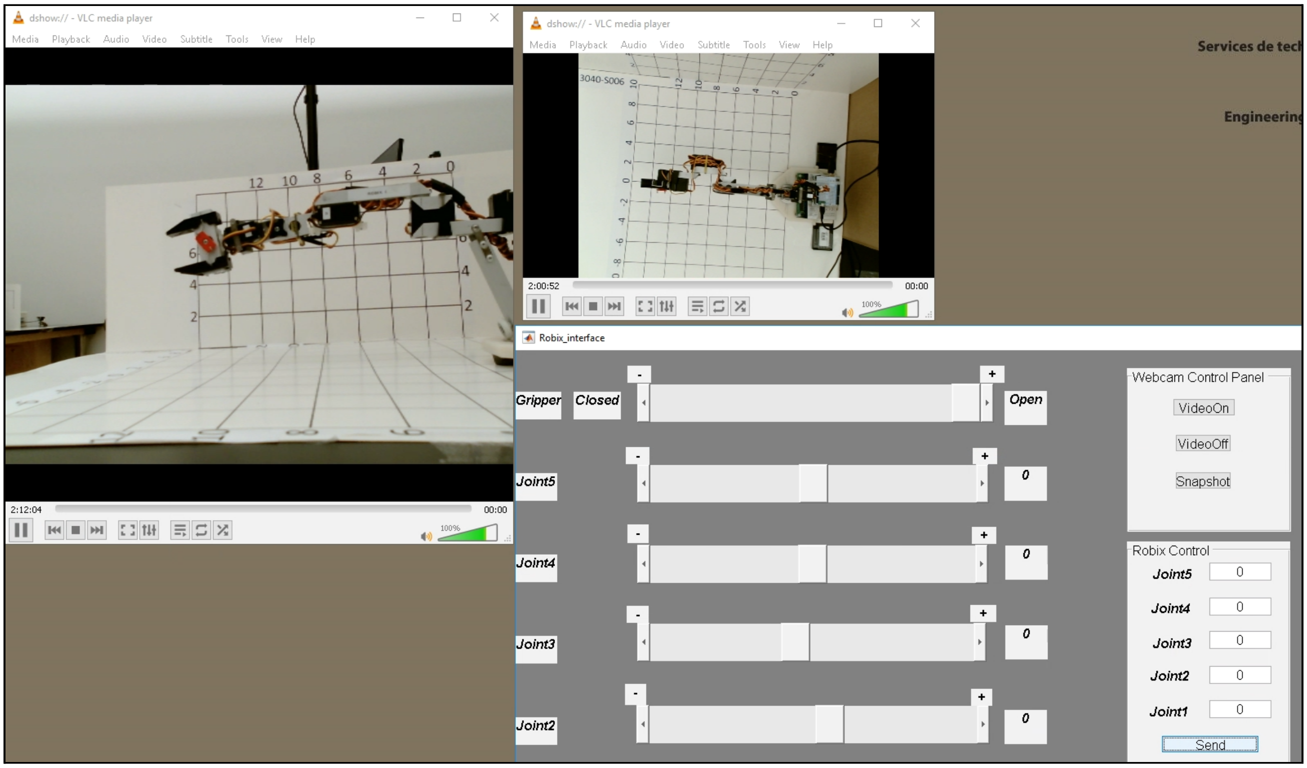
```
%Columns: |theta1|theta2|theta3|theta4|theta5|gamma|A|B|C|rt|
%         |      |      |      |      |      |     | | | |  |
%         |      |      |      |      |      |     | | | |  |
%         |      |      |      |      |      |     | | | |  |...
%\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
%where A,B,C, and gamma are used in calculations, and rt represents the
%terms being squarerooted during the calculation of theta3
EE=zeros(8,10);
zero_flag=zeros(1,8); %1 if solution involved atan2d of a complex number
%--------------------Calculate thetai--------------------------------------
U=98*Q(2,3);V=-98*Q(1,3);W=Q(1,4)*Q(2,3)-Q(2,4)*Q(1,3);
%Main for loop: calculates all 8 possible thetai solutions
for i=1:8
    j=fix((i+1)/2);
    k=fix((i+3)/4);
    EE(i,1)=2*atan2d((-V+((-1)^i)*sqrt(V^2+(U+W)*(U-W))),(-U-W)); %theta1
    EE(i,2)=atan2d(((-1)^j)*Q(2,3),((-1)^j)*Q(1,3))-EE(i,1); %theta2
    %to calculate gamma we alternate b/w equations obtained from the Q
    %matrix depending on if it will lead to an error. A similar strategy is
    %used in solving theta4.
    if cos_error(EE(i,1),EE(i,2))==1
        EE(i,6)=(Q(2,4)-98*sind(EE(i,1)))/(sind(EE(i,1)+EE(i,2))); %gamma
    else
        EE(i,6)=(Q(1,4)-98*cosd(EE(i,1)))/(cosd(EE(i,1)+EE(i,2))); %gamma
    end
    EE(i,7)=11520-120*EE(i,6); %A
    EE(i,8)=18840-120*Q(3,4); %B
    EE(i,9)=-(EE(i,6)^2)+192*EE(i,6)-(Q(3,4))^2+314*Q(3,4)-25365; %C
    EE(i,10)=EE(i,8)^2+(EE(i,7)+EE(i,9))*(EE(i,7)-EE(i,9)); %rt
    %to calculate theta3 we first check to see if the correspoding term
    %being squarooted is positive, and only proceed if that is the case
    if EE(i,10)>=0
        EE(i,3)=2*atan2d((-EE(i,8)+((-1)^k)*sqrt(EE(i,10))),...
            (-EE(i,7)-EE(i,9)));
    else
        zero_flag(1,i)=1;
    end
    %Knowing theta1-3 we can calculate a singular theta4/5 pair for every
    %solution calculated so far, and thus we end up with 8 solutions.
    if cos_error(EE(i,1),EE(i,2))==1
        EE(i,4)=atan2d(Q(2,3)/sind(EE(i,1)+EE(i,2)),(-Q(3,3)))-EE(i,3)...
            -delta;
    else
        EE(i,4)=atan2d(Q(1,3)/cosd(EE(i,1)+EE(i,2)),(-Q(3,3)))-EE(i,3)...
            -delta;
```

```
    else
        EE(i,4)=atan2d(Q(1,3)/cosd(EE(i,1)+EE(i,2)),(-Q(3,3)))-EE(i,3)...
            -delta;
    end
    EE(i,5)=atan2d(-Q(3,2)/(sind(EE(i,3)+EE(i,4)+delta)),...
        Q(3,1)/(sind(EE(i,3)+EE(i,4)+delta)));
end
%-----------Post Calculations to obtain realizable solutions--------------
EEO1=EE(:,1:5); %where EEO1 is an intermidiate EE matrix
cntr=1;
for q=1:8
    for m=1:5
        %convert any awkward angles by adding or subtracting 360
        if EEO1(q,m)<-180
            EEO1(q,m)=EEO1(q,m)+360;
        end
        if EEO1(q,m)>180
            EEO1(q,m)=EEO1(q,m)-360;
        end
    end
    %check if a solution involved a complex number, while also checking if
    %another condition of the Q matrix is satisfied by the theta1-5
    %solution
    %We also use fix() to convert the condition to be percise to 2 decimal
    %places.
    if zero_flag(q)==0 && fix((-110*cosd(EEO1(q,3)+EEO1(q,4)+delta)+...
            60*sind(EEO1(q,3))+157)*1e2)/1e2==fix(Q(3,4)*1e2)/1e2
        EEO(cntr,:)=EEO1(q,:); %Defines End-Effector_Output matrix
        cntr=cntr+1; %so the final matrix has as many terms as viable sols.
    end
end

%----------Final end effector location and angles-------------------------
disp('| Theta1I | Theta2I | Theta3I | Theta4I | Theta5I |');
disp(EEO);

%///////////////////////////Functions///////////////////////////////////////
function ce=cos_error(t1,t2)
    ce=0;
    for n=-10:10
        if t1+t2==90+n*180
            ce=1;
        end
```

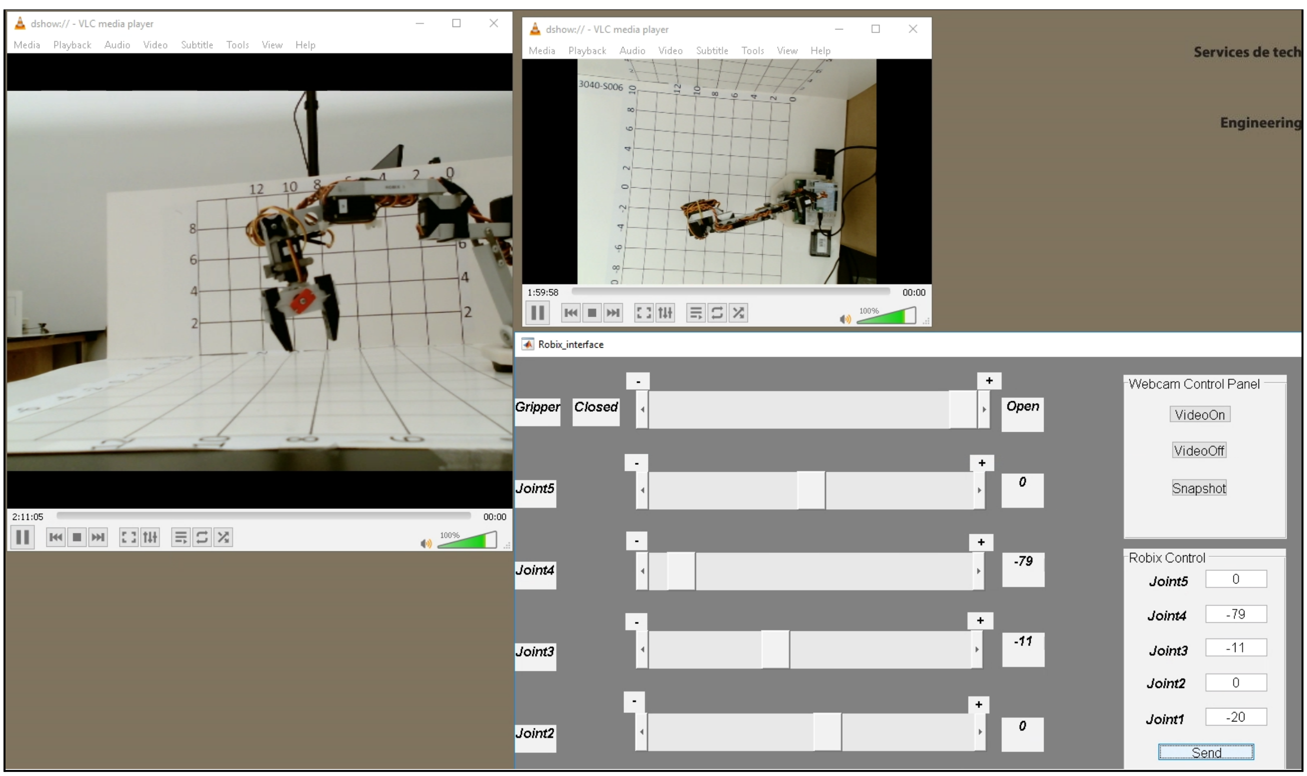
```
%----------Final end effector location and angles--------------------------
disp('| Theta1I | Theta2I | Theta3I | Theta4I | Theta5I |');
disp(EEO);

%//////////////////////////Functions/////////////////////////////////////
function ce=cos_error(t1,t2)
    ce=0;
    for n=-10:10
        if t1+t2==90+n*180
            ce=1;
        end
    end
end

function se=sin_error(t1,t2)
    se=0;
    for n=-10:10
        if t1+t2==n*180
            se=1;
        end
    end
end
```

Appendix B

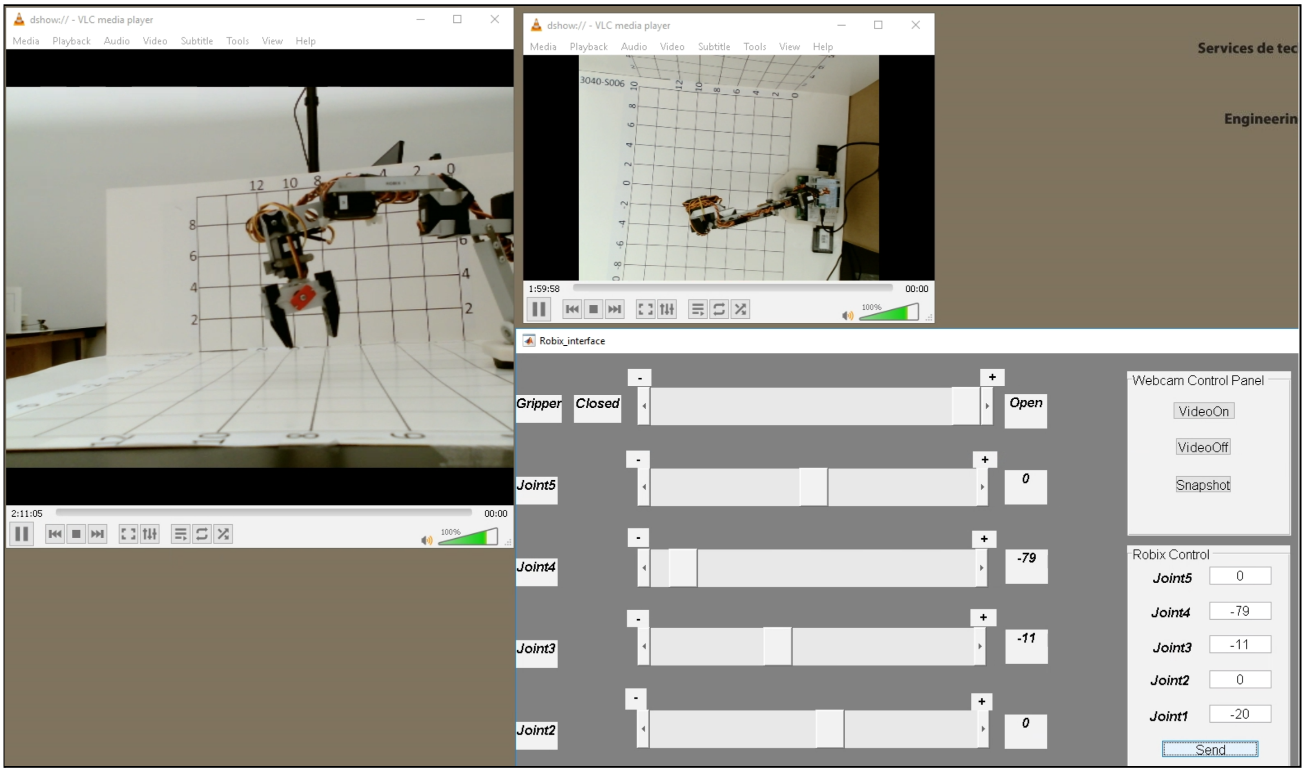
Set 1



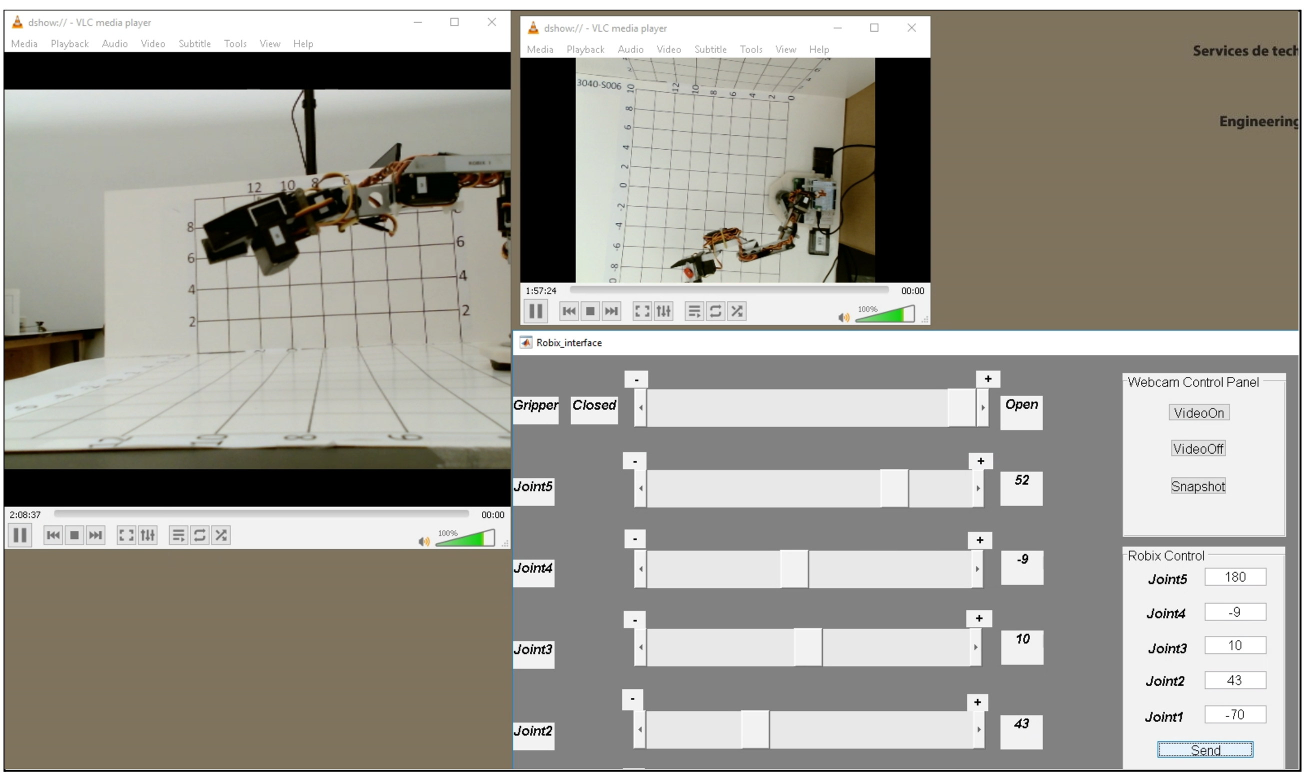
Set 2



Set 3


dshow:// - VLC media player
Media Playback Audio Video Subtitle Tools View Help
2:11:05
00:00
dshow:// - VLC media player
Media Playback Audio Video Subtitle Tools View Help
1:59:58
00:00
Services de tec
Engineerin
Robix_interface
Gripper
Closed
Open
Joint5
0
Joint4
-79
Joint3
-11
Joint2
0
Webcam Control Panel
VideoOn
VideoOff
Snapshot
Robix Control
Joint5 0
Joint4 -79
Joint3 -11
Joint2 0
Joint1 -20
Send

Set 4


dshow:// - VLC media player
Media Playback Audio Video Subtitle Tools View Help
2:08:37
00:00
dshow:// - VLC media player
Media Playback Audio Video Subtitle Tools View Help
1:57:24
00:00
Services de tech
Engineering
Robix_interface
Gripper
Closed
Open
Joint5
52
Joint4
-9
Joint3
10
Joint2
43
Webcam Control Panel
VideoOn
VideoOff
Snapshot
Robix Control
Joint5 180
Joint4 -9
Joint3 10
Joint2 43
Joint1 -70
Send