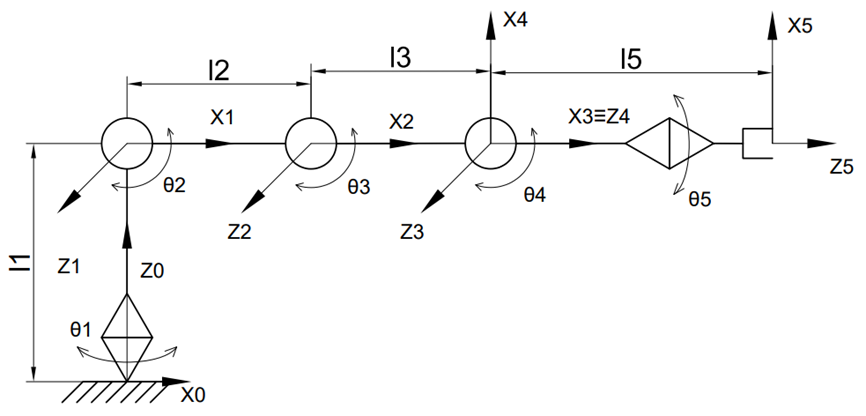
CHƯƠNG 3: ĐỘNG HỌC VÀ ĐỘNG LỰC HỌC TAY MÁY

* 1. . Động học vị trí

3.1.1 Động học thuận

**Sơ đồ động Robot:**



*Hình 3.1.1: Sơ đồ động Robot*

**Biểu diễn Denavit – Hartenberg:**

Mô hình hóa Denavit-Hartenberg (Viết tắt là phương pháp D - H) là cách biểu diễn đơn giản mô hình các khâu và khớp của robot và có thể sử dụng cho bất cứ cấu hình robot nào, kể cả bài toán phức tạp hay đơn giản.

Áp dụng phương pháp D – H, ta gắn một hệ trục tham chiếu tới mỗi khớp và sau đó xác định sự chuyển vị từ khớp này đến khớp kế tiếp, với mỗi khớp chúng ta sẽ gắn trục z và trục x lên chúng (hình 3.1).

Áp dụng phương pháp D - H cho khớp 1, tương tự với các khớp còn lại. Ta có:

* Đánh số trục z của khớp 1 là 0. Như vậy, trục z biểu diễn khớp 1 là z0. Trục x sẽ song song với đường vuông góc chung giữa các trục khớp của khâu, tương tự với các khớp còn lại.
* Với:
* l: biểu diễn khoảng cách dường vuông góc chung giữa trục z0­ và trục z1.
* θ1: góc quay từ trục x0 tới x1 xung quanh trục z­0.
* α1: góc quay của trục z0 tới z1 xung quanh trục x1.
* d1: khoảng cách từ gốc tọa độ thứ 0 tới giao điểm của của trục z0 và x1 dọc theo trục z0.

🡪 Từ đó, ta có thể lập ra bảng D – H với 5 khớp một cách tương ứng dựa theo hình 3.1:

**Bảng D-H:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Khâu** | **θ** | **l** | **α** | **d** | **Biến** |
| 1 |  | 0 |  | l1 |  |
| 2 |  |  | 0 | 0 |  |
| 3 |  |  | 0 | 0 |  |
| 4 |  | 0 |  | 0 |  |
| 5 |  | 0 | 0 | l5 |  |

*Bảng 3.1: Bảng Denavit–Hartenberg của robot*

Ở hệ trục thứ i sẽ mô tả ma trận vị trí và hướng so với hệ trục thứ i-1 như sau:

Ma trận chuyển trục Ai:

 (1)

Với .

Từ đó, ta có ma trận chuyển trục của 5 khớp từ A1 đến A5 như sau:



Với kết quả trên, ta có ma trận chuyển vị tổng giữa nền Robot và cánh tay là:



=

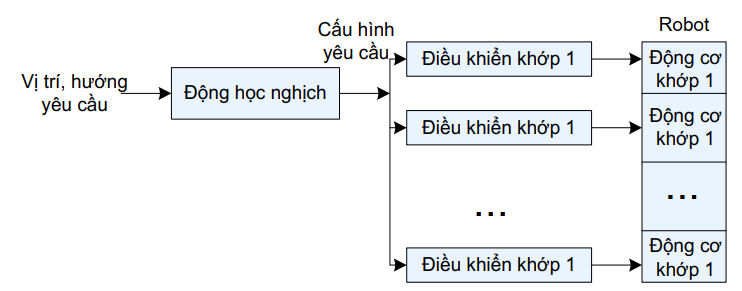
 (2)

Theo động học thuận, tọa độ điểm P là:



3.1.2. Động học nghịch

Bài toán động học thuận gồm việc giải quyết vị trí và hướng của tay gắp hay cơ cấu chấp hành cuối khi biết tất cả các biến khớp. Bài toán động học nghịch yêu cầu tìm tập hay nhiều tập nghiệm của các khớp khi biết vị trí và hướng tay gắp.



*Hình 3.1.2: Điều khiển vị trí của cánh tay robot*

Với các giá trị đầu vào là Px, Py, P­z, φ, γ, Ta có:





 (4)



Ta đặt:





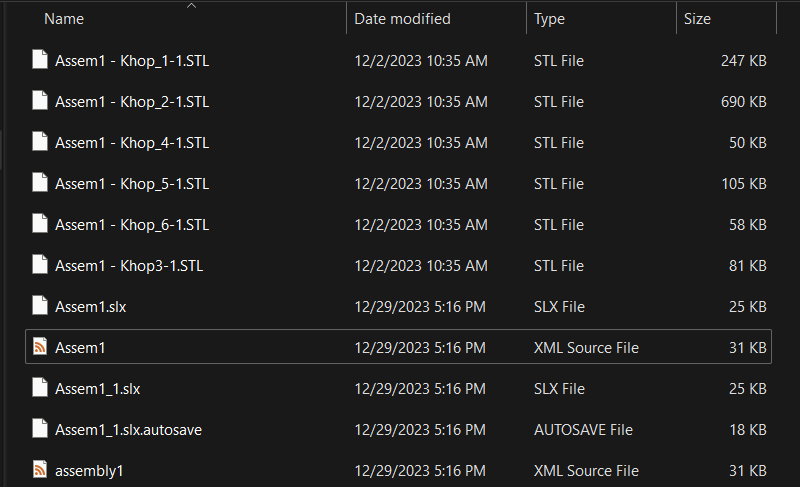


 (6)



* 1. . Mô phỏng Matlab
     1. Xây dựng mô hình robot trên Simulink

Sau khi đã thiết kế robot 3D bằng phần mềm SolidWorks, ta liên kết chúng với Matlab bằng thư viện Simscape Multibody Link trên phần mềm để tạo ra file xml và các files khác hỗ trợ cho việc mô phỏng robot bằng Matlab.

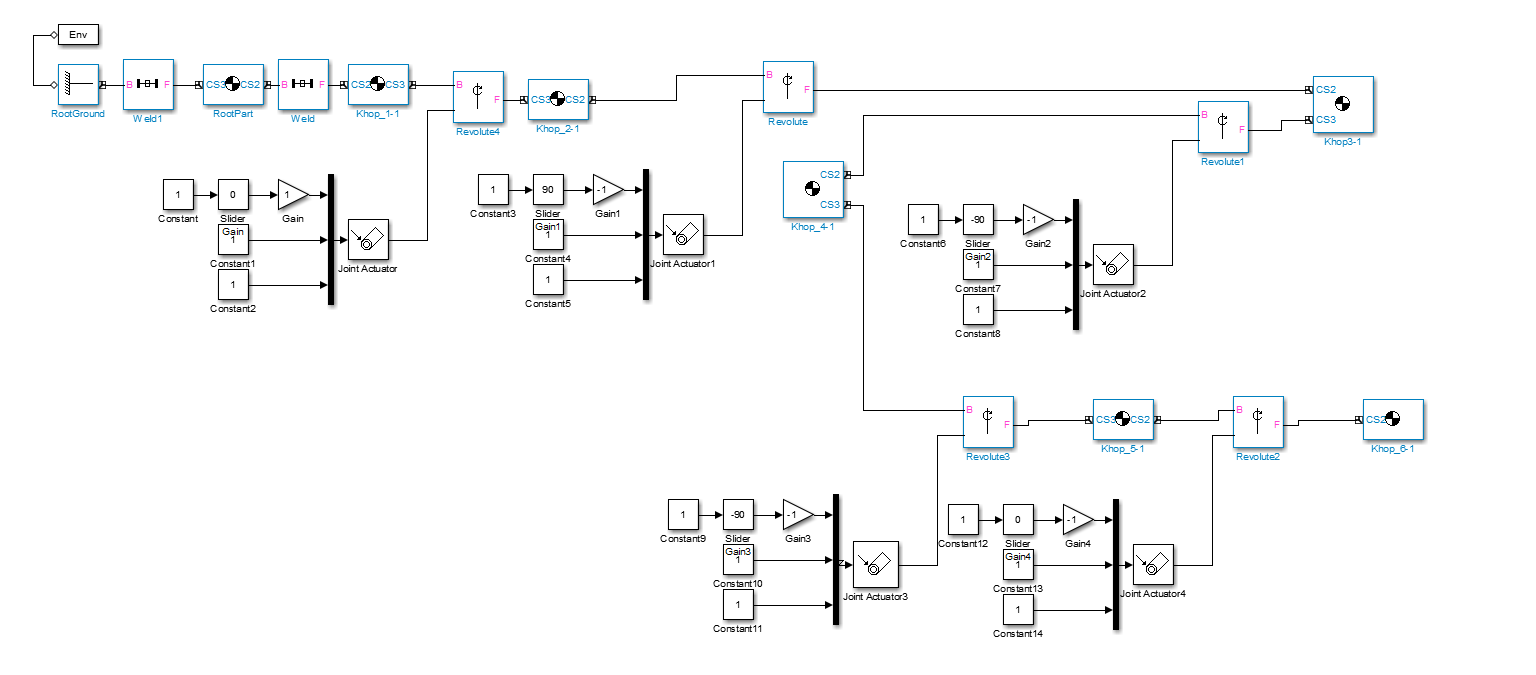


*Hình 3.2.1a: Files được tạo ra với Simscape Multibody Link*

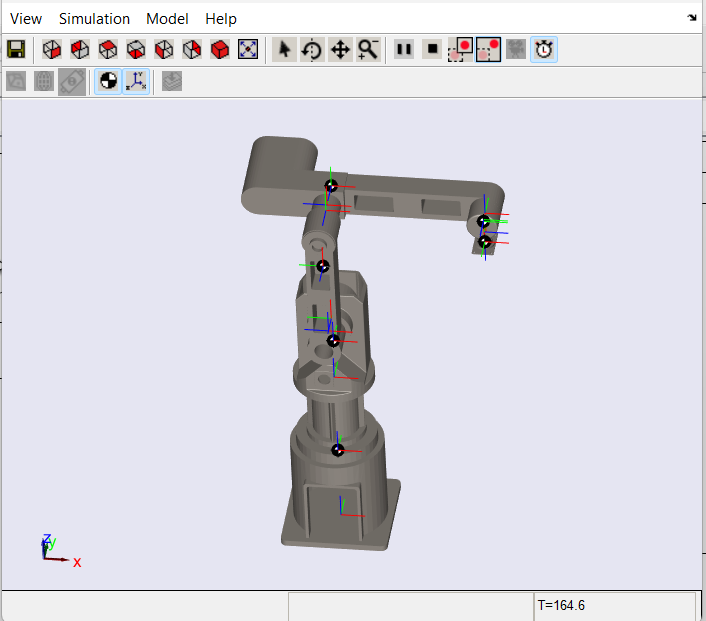
Đối với Matlab, bằng câu lệnh **mech\_import**, với đầu vào là file Assem1.xml, ta sẽ có được một file.slx với chức năng mô phỏng 3D robot trên phần mềm này.



*Hình 3.2.1b: Build model trên Matlab với lệnh mech\_import*



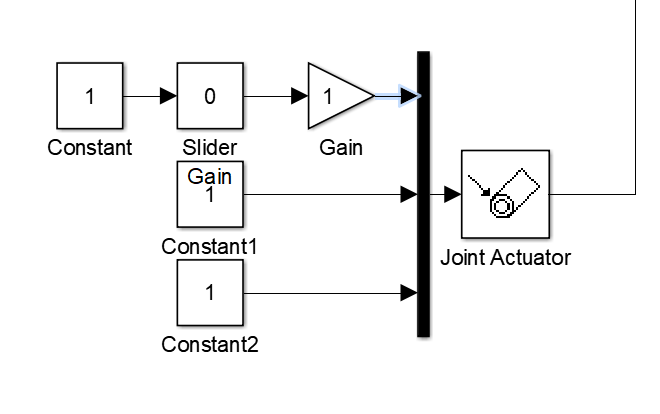
*Hình 3.2.1c: Các khối điều khiển các khớp của robot*



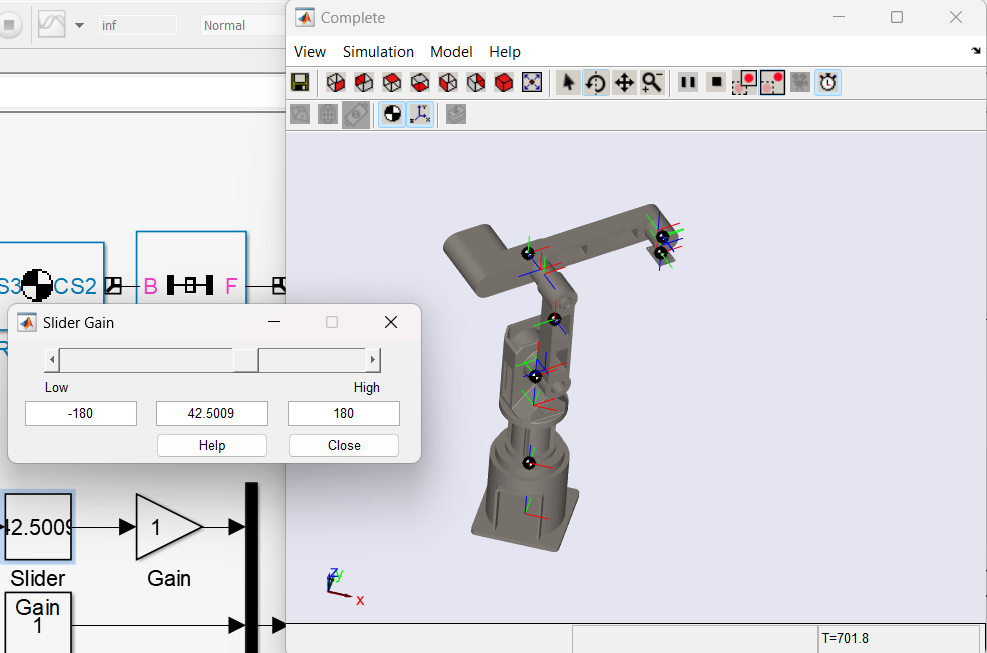
*Hình 3.2.1d: Robot biểu diễn 3D bằng Matlab*

* + 1. Giới thiệu về phần mềm điều khiển robot bằng Matlab

Bằng việc thêm những khối điều khiển với những robot trong file.slx, ta có thể tùy chỉnh được góc các khớp xoay bằng cách nháy đúp chuột vào thanh trượt (Slider) và điều chỉnh các thông số khác một cách phù hợp:



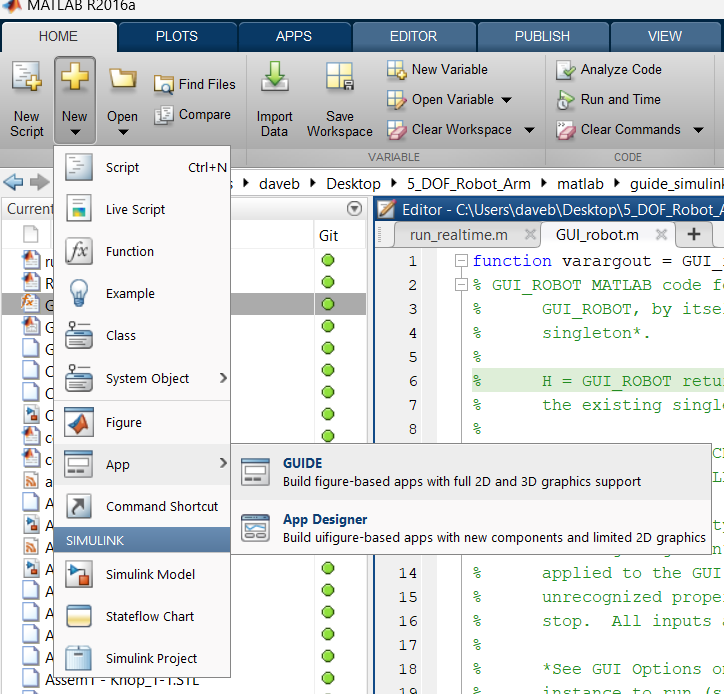
*Hình 3.2.2a: Khối điều khiển góc cho robot*



*Hình 3.2.2b: Điều khiển khớp robot bằng Slider*

* + 1. Thiết kế phần mềm điều khiển robot bằng Matlab

Ngoài ra, Matlab cũng hỗ trợ chúng ta trong việc xây dựng GUI để điều khiển bằng chức năng GUIDE trong Matlab:

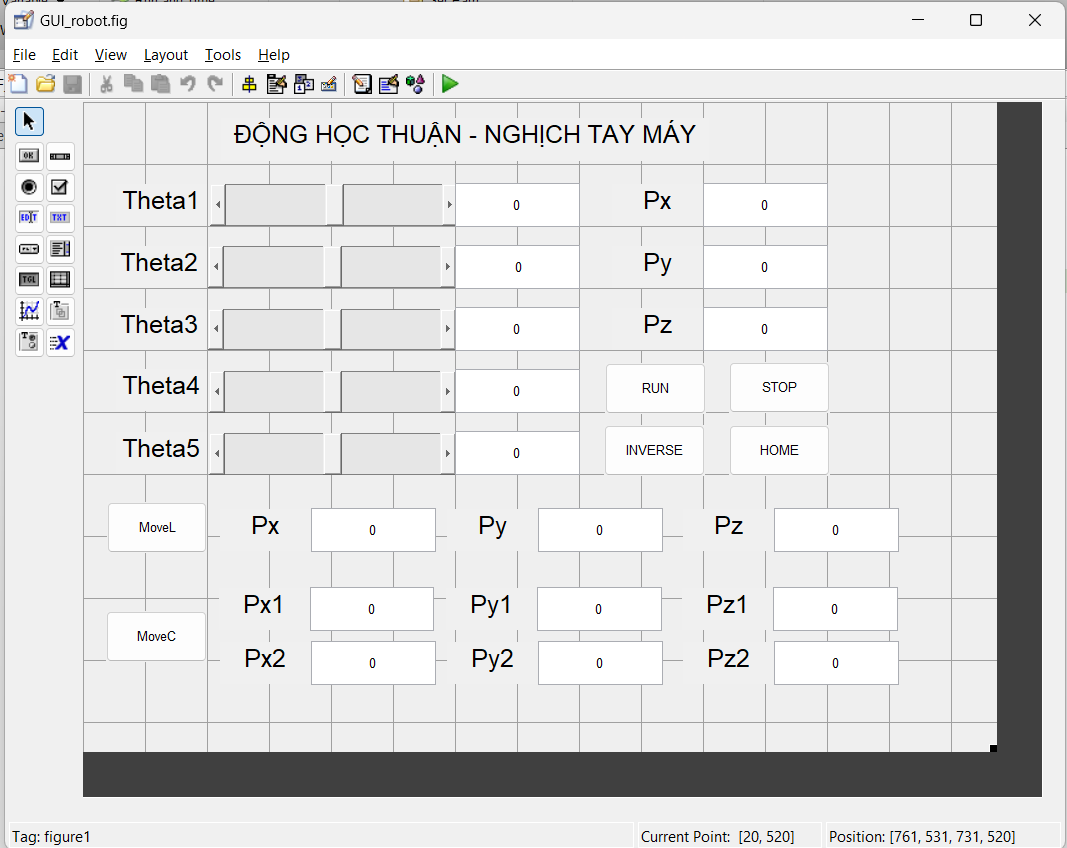


*Hình 3.2.3a: Tạo GUI điều khiển robot*

Từ đó, ta sẽ có được một giao diện cho phép chúng ta thiết kế bộ điều khiển.

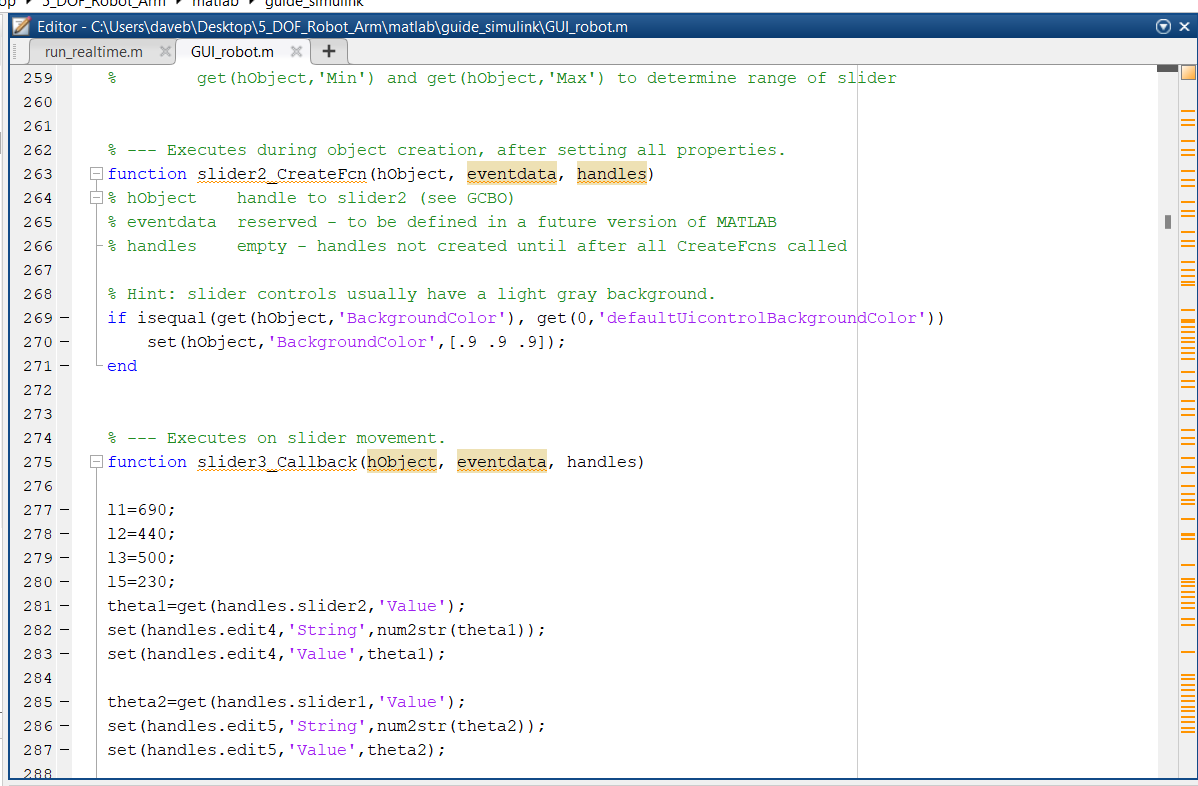
Như hình 3.2.3b, ta có thể thấy:

* Slider: Điều chỉnh góc quay của các khớp từ 1 đến 5.
* Px, Py, Pz: Hiển thị vị trí hiện tại/mong muốn của đầu tay gắp.
* Nút nhấn MoveL, MoveC: thực hiện những chức năng mong muốn.



*Hình 3.2.3b: Giao diện thiết kế bộ điều khiển*

Matlab sẽ tạo ra một file.m để ta có thể thao tác trực tiếp lên code (Timer, cập nhật giá trị Real-time, thực thi chức năng của nút nhấn…)



*Hình 3.2.3c: Phần mềm lập trình chức năng cho các đối tượng trong bộ điều khiển*

* + 1. Mô phỏng vị trí dựa trên kết quả tính toán

Bằng việc áp dụng cái phương trình động học thuận, động học nghịch, ta có thể dễ dàng mô phỏng được vị trí đầu cuối của tay gắp bằng việc áp dụng chúng vào trong chương trình.

**Khối điều khiển vị trí đầu tay gắp bằng thanh trượt áp dụng động học thuận:**

*function slider4\_Callback(hObject, eventdata, handles)*

*l1=690;*

*l2=440;*

*l3=500;*

*l5=230;*

*theta1=get(handles.slider2,'Value');*

*set(handles.edit4,'String',num2str(theta1));*

*set(handles.edit4,'Value',theta1);*

*theta2=get(handles.slider1,'Value');*

*set(handles.edit5,'String',num2str(theta2));*

*set(handles.edit5,'Value',theta2);*

*theta3=get(handles.slider3,'Value');*

*set(handles.edit6,'String',num2str(theta3));*

*set(handles.edit6,'Value',theta3);*

*theta4=get(handles.slider4,'Value');*

*set(handles.edit7,'String',num2str(theta4));*

*set(handles.edit7,'Value',theta4);*

*theta5=get(handles.slider5,'Value');*

*set(handles.edit8,'String',num2str(theta5));*

*set(handles.edit8,'Value',theta5);*

*set\_param('Complete/Slider Gain','Gain',num2str(theta1));*

*set\_param('Complete/Slider Gain1','Gain',num2str(theta2+90));*

*set\_param('Complete/Slider Gain2','Gain',num2str(theta3-90));*

*set\_param('Complete/Slider Gain3','Gain',num2str(theta4-90));*

*set\_param('Complete/Slider Gain4','Gain',num2str(theta5));*

*t2=theta2+90;*

*t3=theta3-90;*

*t4=theta4-90;*

*px=cosd(theta1)\*(l3\*cosd(t2+t3)+l2\*cosd(t2)+l5\*cosd(-90));*

*py=sind(theta1)\*(l3\*cosd(t2+t3)+l2\*cosd(t2)+l5\*cosd(-90));*

*pz=l1+l3\*sind(t2+t3)+l2\*sind(t2)+l5\*sind(-90);*

*set(handles.edit1,'String',num2str(px));*

*set(handles.edit2,'String',num2str(py));*

*set(handles.edit3,'String',num2str(pz));*



*Hình 3.2.4a: Thanh trượt điều khiển góc xoay của khớp 1*

Với:

|  |  |
| --- | --- |
| **Khối code** | **Chức năng** |
| *theta1=get(handles.slider2,'Value');*  *set(handles.edit4,'String',num2str(theta1));*  *set(handles.edit4,'Value',theta1);* | Lấy giá trị từ thanh trượt và hiển thị lên ô giá trị ở kế bên với tên *edit4.* |
| *set\_param('Complete/Slider Gain','Gain',num2str(theta1));*  *set\_param('Complete/Slider Gain1','Gain',num2str(theta2+90));*  *set\_param('Complete/Slider Gain2','Gain',num2str(theta3-90));*  *set\_param('Complete/Slider Gain3','Gain',num2str(theta4-90));*  *set\_param('Complete/Slider Gain4','Gain',num2str(theta5));* | Cài đặt lại parameter cho robot phù hợp với robot ngoài thực tế. |
| *t2=theta2+90;*  *t3=theta3-90;*  *t4=theta4-90;*  *px=cosd(theta1)\*(l3\*cosd(t2+t3)+l2\*cosd(t2)+l5\*cosd(-90));*  *py=sind(theta1)\*(l3\*cosd(t2+t3)+l2\*cosd(t2)+l5\*cosd(-90));*  *pz=l1+l3\*sind(t2+t3)+l2\*sind(t2)+l5\*sind(-90);*  *set(handles.edit1,'String',num2str(px));*  *set(handles.edit2,'String',num2str(py));*  *set(handles.edit3,'String',num2str(pz));* | Áp dụng động học thuận để tính vị trí đầu cuối của tay gắp và hiển thị lên giao diện. |

Thuật toán này sẽ áp dụng với việc điều khiển góc xoay của các khớp còn lại đồng thời cũng dùng cho nút nhấn RUN ở trên giao diện.



*Hình 3.2.4b: Nút nhấn để xác định đầu tay gắp với các góc biết trước*

**Khối điều khiển giá trị góc của các khớp bằng nút nhấn áp dụng động học nghịch:**

*function pushbutton3\_Callback(hObject, eventdata, handles)*

*syms theta1 theta2 theta3 theta4 theta5 c3 s3*

*l1=690;*

*l2=440;*

*l3=500;*

*l5=230;*

*theta234=-90;*

*px=str2double(get(handles.edit1,'String'));*

*py=str2double(get(handles.edit2,'String'));*

*pz=str2double(get(handles.edit3,'String'));*

*theta1=atan2d(py,px);*

*m= sqrt(px^2+py^2)-l5\*cosd(theta234);*

*%m1=l3\*cosd(theta2+theta3)+l2\*cosd(theta2);*

*n=pz-l1-l5\*sind(theta234);*

*%n1=l3\*sind(theta2+theta3)+l2\*sind(theta2);*

*c3=double((m^2+n^2-l3^2-l2^2)/(2\*l3\*l2));*

*s3=double(sqrt(1-c3^2));*

*theta3=atan2d(s3,c3);*

*if (theta3<80 && theta3 >-30)*

*theta3=atan2d(s3,c3);*

*else*

*s3=double(-sqrt(1-c3^2));*

*theta3=atan2d(s3,c3);*

*end*

*A= m\*(l3\*cosd(theta3)+l2)+n\*(l3\*sind(theta3));*

*B= n\*(l3\*cosd(theta3)+l2)-m\*(l3\*sind(theta3));*

*theta2=atan2d(B,A);*

*theta4=-90-theta2-theta3;*

*guidata(hObject,handles);*

*set(handles.edit4,'String',num2str(theta1));*

*set(handles.edit5,'String',num2str(theta2-90));*

*set(handles.edit6,'String',num2str(theta3+90));*

*set(handles.edit7,'String',num2str(theta4+90));*

*set(handles.slider2,'Value',round(theta1,2));*

*set(handles.slider1,'Value',round(theta2-90,2));*

*set(handles.slider3,'Value',round(theta3+90,2));*

*set(handles.slider4,'Value',round(theta4+90,2));*

*set\_param('Complete/Slider Gain','Gain',num2str(theta1));*

*set\_param('Complete/Slider Gain1','Gain',num2str(theta2));*

*set\_param('Complete/Slider Gain2','Gain',num2str(theta3));*

*set\_param('Complete/Slider Gain3','Gain',num2str(theta4));*

|  |  |
| --- | --- |
| **Khối code** | **Chức năng** |
| *l1=690;*  *l2=440;*  *l3=500;*  *l5=230;*  *theta234=-90;*  *px=str2double(get(handles.edit1,'String'));*  *py=str2double(get(handles.edit2,'String'));*  *pz=str2double(get(handles.edit3,'String'));*  *theta1=atan2d(py,px);*  *m= sqrt(px^2+py^2)-l5\*cosd(theta234);*  *%m1=l3\*cosd(theta2+theta3)+l2\*cosd(theta2);*  *n=pz-l1-l5\*sind(theta234);*  *%n1=l3\*sind(theta2+theta3)+l2\*sind(theta2);*  *c3=double((m^2+n^2-l3^2-l2^2)/(2\*l3\*l2));*  *s3=double(sqrt(1-c3^2));*  *theta3=atan2d(s3,c3);*  *if (theta3<80 && theta3 >-30)*  *theta3=atan2d(s3,c3);*  *else*  *s3=double(-sqrt(1-c3^2));*  *theta3=atan2d(s3,c3);*  *end*  *A= m\*(l3\*cosd(theta3)+l2)+n\*(l3\*sind(theta3));*  *B= n\*(l3\*cosd(theta3)+l2)-m\*(l3\*sind(theta3));*  *theta2=atan2d(B,A);*  *theta4=-90-theta2-theta3;* | Bằng việc lấy các giá trị đầu cuối mong muốn của tay gắp, cũng như giới hạn không gian hoạt động của robot 3D khớp với thực tế, ta có được các góc theta tương ứng bằng việc áp dụng phương trinh động học nghịch. |



*Hình 3.2.4c: Nút nhấn để xác định góc xoay của các khớp với vị trí đầu cuối*

**Khối set home và ngắt chương trình:** Bằng viêc áp dụng một cách tương tự chức năng của từng khối ở trên, ta có điều chỉnh để cài đặt vị trí ban đầu cho Robot cũng như tắt chương trình một cách tương ứng

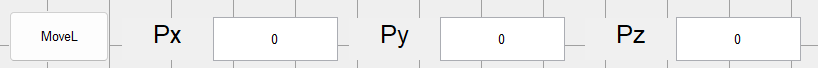
|  |  |
| --- | --- |
| **Khối code** | **Chức năng** |
| *function pushbutton2\_Callback(hObject, eventdata, handles)*  *close;* | Tắt chương trình |



*Hình 3.2.4d: Nút nhấn set Home và tắt chương trình*

**Khối chạy chức năng MoveL/MoveC:**

|  |  |
| --- | --- |
| **Khối code** | **Chức năng** |
| *function MoveL(hObject, event, handles)*  *global t;*  *global tar\_pos;*  *global cur\_pos;*  *global vect\_u;*  *global tm;*  *if t <= 10*  *x = cur\_pos(1,1) + (vect\_u(1,1) / 10) \* t;*  *y = cur\_pos(1,2) + (vect\_u(1,2) / 10) \* t;*  *z = cur\_pos(1,3) + (vect\_u(1,3) / 10) \* t;*  *value\_angles = convert\_position\_angle(x, y, z)*  *t1 = value\_angles(1,1);*  *t2 = value\_angles(1,2);*  *t3 = value\_angles(1,3);*  *t4 = value\_angles(1,4);*  *t5 = value\_angles(1,5);*  *ret = Check\_angle(t1, t2, t3, t4, t5);*  *if ret ~= 0*  *theta = 0.0;*  *if ret == 1*  *theta = t1;*  *elseif ret == 2*  *theta = t2;*  *elseif ret == 3*  *theta = t3;*  *elseif ret == 4*  *theta = t4;*  *elseif ret == 5*  *theta = t5;*  *end*  *disp(['Error: P2P: theta', num2str(ret), ' = ', num2str(theta), ' out of range']);*  *return;*  *end*  *set(handles.edit1,'String',num2str(x));*  *set(handles.edit2,'String',num2str(y));*  *set(handles.edit3,'String',num2str(z));*  *% guidata(hObject,handles);*  *set(handles.edit4,'String',num2str(t1));*  *set(handles.edit5,'String',num2str(t2 - 90.0));*  *set(handles.edit6,'String',num2str(t3 + 90.0));*  *set(handles.edit7,'String',num2str(t4 + 90.0));*  *set(handles.edit8,'String',num2str(t5));*    *set(handles.slider2,'Value',round(t1,2));*  *set(handles.slider1,'Value',round(t2 - 90.0,2));*  *set(handles.slider3,'Value',round(t3 + 90.0,2));*  *set(handles.slider4,'Value',round(t4 + 90.0,2));*  *set(handles.slider5,'Value',round(t5,2));*    *set\_param('Complete/Slider Gain','Gain',num2str(t1));*  *set\_param('Complete/Slider Gain1','Gain',num2str(t2));*  *set\_param('Complete/Slider Gain2','Gain',num2str(t3));*  *set\_param('Complete/Slider Gain3','Gain',num2str(t4));*  *set\_param('Complete/Slider Gain4','Gain',num2str(t5));*  *t = t + 1;*  *else*  *% Stop the timer*  *stop(tm);*  *% Delete the timer object*  *delete(tm);*  *end* | Hiển thị giá trị trên GUI cũng như robot 3D với mỗi 0.1s tương ứng với 10 điểm được tính toán trước đó. |
| *% --- Executes on button press in MoveL\_btn.*  *function MoveL\_btn\_Callback(hObject, eventdata, handles)*  *% hObject handle to MoveL\_btn (see GCBO)*  *% eventdata reserved - to be defined in a future version of MATLAB*  *% handles structure with handles and user data (see GUIDATA)*  *global t;*  *t = 1;*  *global tar\_pos;*  *global cur\_pos;*  *global vect\_u;*  *global tm;*  *tar\_pos = zeros(1, 3); % Initializing vect\_u array with zeros*  *cur\_pos = zeros(1, 3); % Initializing vect\_u array with zeros*  *% Referred vector*  *vect\_u = zeros(1, 3); % Initializing vect\_u array with zeros*  *tar\_pos(1,1) = str2double(get(handles.MvLx\_tb,'String'));*  *tar\_pos(1,2) = str2double(get(handles.MvLy\_tb,'String'));*  *tar\_pos(1,3) = str2double(get(handles.MvLz\_tb,'String'));*  *cur\_pos(1,1) = str2double(get(handles.edit1,'String'));*  *cur\_pos(1,2) = str2double(get(handles.edit2,'String'));*  *cur\_pos(1,3) = str2double(get(handles.edit3,'String'));*    *% value\_angles = convert\_position\_angle(x, y, z)*  *for i = 1:3*  *vect\_u(1,i) = tar\_pos(1,i) - cur\_pos(1,i);*  *% Uncomment the following line if you want to print the values*  *% disp(['vect\_u(', num2str(i), ') = ', num2str(vect\_u(i))]);*  *end*  *vect\_u*  *tm = timer('ExecutionMode', 'FixedRate', 'Period', 0.1, 'TimerFcn', {@MoveL, handles});*  *start(tm);* | Xác định vị trí ban đầu và mong muốn để tính toán được vector chỉ phương u, đồng thời start Timer để hiển thị Robot 3D trên Matlab |



*Hình 3.2.4e: Nút nhấn MoveL để điều khiển robot đi theo đường thẳng*