

SIMULAÇÃO DA CINEMÁTICA DE UM MANIPULADOR ROBÓTICO

**VÍTOR GUSTAVO SEVERO PEREIRA¹; GILMAR PEREIRA DA CRUZ JÚNIOR²;
ALEXANDRE MOLTER³**

¹Instituto Federal de Educação, Ciência e Tecnologia Sul-rio-grandense - vitorgustavosp@live.com

²Universidade Federal de Pelotas – gilmarpcjunior@yahoo.com.br

³Universidade Federal de Pelotas – alexandre.molter@yahoo.com.br

1. INTRODUÇÃO

O trabalho desenvolvido trata de um código que, aplicando a teoria de rotação e translação de pontos, permite a visualização de um manipulador robótico de três elos no espaço. Dessa forma, o usuário do código irá possuir um recurso gráfico para visualizar o comportamento do mesmo, conforme altera-se o valor dos ângulos de suas juntas, ou até mesmo uma alteração no ordem das rotações, o que, de forma geral, produz resultados diferentes.

O arquivo principal faz chamada de outros módulos, tornando-o mais legível para quem quer simplesmente ver o resultado das rotações, mas não as matrizes que compõem o desenhos dos elos, por exemplo. No arquivo onde é montada a matriz de transformação homogênea, considerando as rotações desejadas em torno dos três eixos cartesianos, é possível identificar uma ordem específica de multiplicação dessas rotações para a formação de uma única. Esse ordenamento se dá devido à configuração específica do braço robótico, bem como ao fato de que as rotações estarem sendo realizados em relação a um sistema de coordenadas fixo, e não a sistemas sucessivamente alterados, caso em que a ordem da multiplicação seria invertida.

No presente estágio deste trabalho, só há preocupação com a cinemática do manipulador, enquanto simulações envolvendo a parte da dinâmica serão desenvolvidas futuramente, apresentando em forma de gráficos as respostas do sistema.

Os principais autores estudados para o desenvolvimento do código foram SPONG; MARK W. e VIDYASAGAR; M. (1989), com eventuais consultas à obra de CRAIG, JOHN J. (2005).

2. METODOLOGIA

O trabalho foi realizado primeiramente por meio de leituras e estudos dos autores apresentados na introdução, e a partir do entendimento do material estudado e analisando um código que também executava rotações e translações de pontos, foi escrito o código deste trabalho.

3. RESULTADOS E DISCUSSÃO

O código desenvolvido por esse trabalho resultou em uma ferramenta de fácil utilização, precisando o usuário apenas indicar os ângulos em que cada junta deve rotacionar, e ao mesmo tempo tendo acesso aos códigos que executam as operações matemáticas, que possibilitam as operações realizadas.

Além disso, o código foi inteiramente comentado, de forma a facilitar o entendimento do que está acontecendo.

Os códigos são apresentados abaixo:

Arquivo “segmento.m”:

```
%cria vetor que representa segmento que começa em (x0,y0,z0) e termina em
%(xf,yf,zf)
function seg = segmento(x0,y0,z0,xf,yf,zf)
seg = [x0 y0 z0; xf yf zf]';
```

Arquivo “espira_quadrada.m”:

```
%cria vetor que representa espira quadrada de base b e posiciona na
altura y
function espira = espira_quadrada (b, y)
espira=[0 0 y; b 0 y; b b y; 0 b y; 0 0 y]';
```

Arquivo “prisma.m”:

```
%cria vetor que representa prisma retangular, com duas espiras nas
extremidades e uma no meio,todas ligadas por 4 segmentos
function prism = prisma(b, h)
prism = [espira_quadrada(b, 0) espira_quadrada(b, h/2) espira_quadrada(b,
h) segmento(0, 0, h, b, 0, h) segmento(b, 0, h, b, 0, 0) segmento(b, 0,
0, b, b, 0) segmento(b, b, 0, b, b, h) segmento(b, b, h, 0, b, h)
segmento(0, b, h, 0, b, 0); ones(1,27)];
```

Arquivo “eixo_config.m”:

```
%Configura espaço cartesiano, atualizando a imagem e colocando em uma
%perspectiva mais apropriada
h = newplot;
axis ([-5 11 -5 11 -5 11]);
axis equal;
grid on;
xlabel('x');
ylabel('y');
zlabel('z');
```

Arquivo “THomogen.m”:

```
%--MATRIZ DE TRANSFORMAÇÃO HOMOGÊNEA
%--A ordem das rotações é: j, k, i (em relação ao frame fixo do MATLAB
%--OBS.:Essa ordem de rotação se deve à forma como os prisma estão
%posicionados e à forma como podem rotacionar
function thomogen = THomogen (ri, rj, rk, dx, dy, dz)
%--Converte os ângulos de graus para radianos
ri = pi*ri/180;
rj = pi*rj/180;
rk = pi*rk/180;
%--Matrizes de rotação em torno de i, j e k
Ri = [1 0 0; 0 cos(ri) -sin(ri); 0 sin(ri) cos(ri)];
Rj = [cos(rj) 0 sin(rj); 0 1 0; -sin(rj) 0 cos(rj)];
Rk = [cos(rk) -sin(rk) 0; sin(rk) cos(rk) 0; 0 0 1];
%--Vetor com informação de translação
d = [dx; dy; dz];
%Concatenação da matriz de rotação resultante e vetor de translação
thomogen = [Ri*Rk*Rj d; 0 0 0 1];
```

Arquivo “manipulador_3_elos.m”:

```
clc;
clear all;
%-----%
%--DECLARAÇÃO DE ELOS E DEFINIÇÃO DE POSIÇÃO INICIAL
%--LINK 1
prism1 = prisma(1, 6); %Declaração do link 1
T = THomogen(0, 0, 0, -0.5, -0.5, 0); %Posicionamento do prisma na origem
prism1 = T * prism1;
```



```
prism1_init = prism1; %Posição inicial guardada em
prism1_init

%--LINK 2
prism2 = prisma(1, 4); %Declaração do link 2
T = THomogen(0, 0, 0, -0.5, 0.5, 0); %Posicionamento do prisma encostando
no primeiro
prism2 = T * prism2;
prism2_init = prism2; %Posição inicial guardada em
prism1_init

%--LINK 3
prism3 = prisma(1, 4); %Declaração do link 2
T = THomogen(0, 0, 0, -0.5, 1.5, 0); %Posicionamento do prisma encostando
no primeiro
prism3 = T * prism3;
prism3_init = prism3; %Posição inicial guardada em
prism1_init

%-----%
%-- INFORMAÇÕES PARA AS MATRIZES DE TRANSFORMAÇÃO HOMOGÊNEA
%--Ângulo de rotação da junta 1, em graus
q1 = linspace(0,-90); %vetor de 100 posições de 0 à q1
%--Ângulo de rotação da junta 2, em graus
q2 = linspace(0,45); %vetor de 100 posições de 0 à q2
%--Ângulo de rotação da junta 3, em graus
q3 = linspace(0,-45); %vetor de 100 posições de 0 à q3
%-----%

for n= 1:100%-- "n" vai de 1 até o tamanho dos vetors q1, q2 e q3 (todos
do mesmo tamanho)
    %--Movimenta elo 1
    prism1 = prism1_init;
    T = THomogen(0, 0, q1(n), 0, 0, 0);
    prism1 = T * prism1;

    %--Movimenta elo 2
    prism2 = prism2_init;
    T = THomogen(0, 0, 0, 0, 5.5)*THomogen(0, q2(n), q1(n), 0, 0, 0);%-
-Rotaciona depois translada
    prism2 = T * prism2;

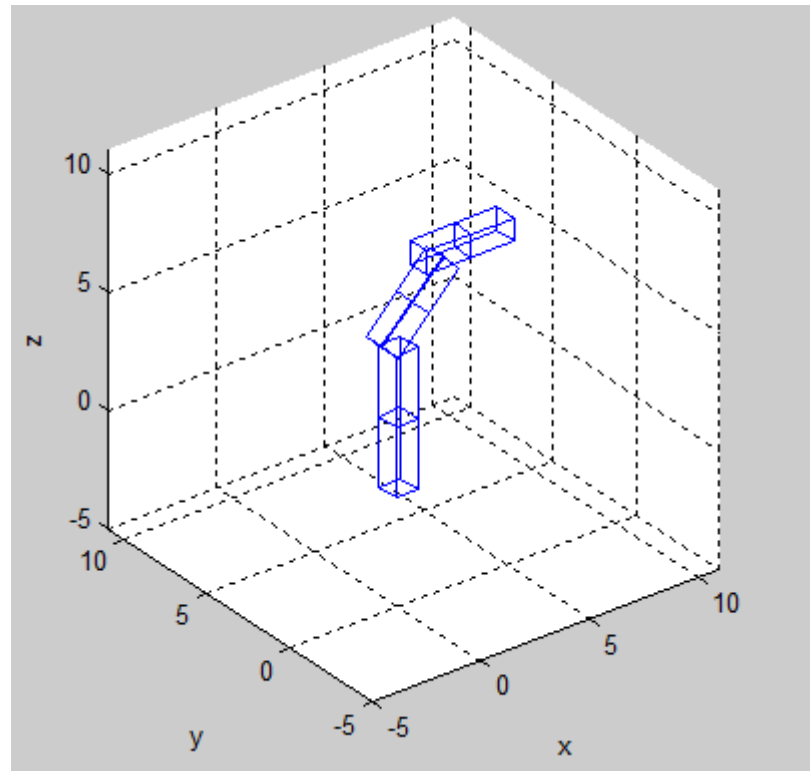
    %--Movimenta LINK 3
    %--O elo 3 deve ser corretamente ajustado até a extremidade do elo 2
    prism3 = prism3_init;
    T = THomogen(0, 0, q1(n), 0, 0, 0) * THomogen(0, 0, 0,
sin(pi*q2(n)/180)*3.5, 0, cos(pi*q2(n)/180)*3.5 + 5.5) * THomogen(0,
q3(n), 0, 0, 0, 0) * THomogen(0, q2(n), 0, 0, 0, 0);
    prism3 = T * prism3;

    eixo_config; %--configura os eixos para melhor visualização

    %--desenha os prismas, ligando ponto a ponto com um linha
    line(prism1(1,:), prism1(2,:), prism1(3,:));
    line(prism2(1,:), prism2(2,:), prism2(3,:));
    line(prism3(1,:), prism3(2,:), prism3(3,:));

    pause(1/1000000);%--Pequeno atraso para possibilitar visualização
end;
```

Representação gráfica do manipulador, gerada pelo código.



4. CONCLUSÕES

O código apresentado simula a cinemática de um manipulador robótico a partir de três entradas dos ângulos das juntas. Além disto, ele apresenta três elos do sistema graficamente.

Não foi considerada, entretanto, a dinâmica do manipulador. As simulações envolvendo esse aspecto são objetivos para os estágios seguintes do trabalho, em que as respostas do sistema serão exibidas por meio de gráficos, e posteriormente será ainda aplicada a teoria de controle nesses movimentos, podendo assim serem comparadas essas duas respostas.

5. REFERÊNCIAS BIBLIOGRÁFICAS

SPONG, MARK W.; VIDYASAGAR, M. **Robot Dynamics and Control**. Hoboken: Wiley, 1989.

CRAIG, JOHN J. **Introduction to Robotics: Mechanics and Control**. Upper Saddle River: Pearson, 2005.