

Solução dos Exercícios de Programação e MATLAB Capítulo 4

Alisson Jaques*. Matheus Nogueira*. Sillas Francisco*.

*Universidade do Estado de Minas Gerais

Abstract: In this article you will find resolutions of the programming and MATLAB exercises proposed in chapter 4 of the book Robotics by author John J. Craig.

Resumo: Neste artigo encontra-se as resoluções dos exercícios de programação e MATLAB propostos no capítulo 4 do livro Robótica do autor John J. Craig.

Palavras-chaves: MATLAB; ferramenta; sub-rotina; orientação; sistema de referência; elos.

1. INTRODUÇÃO

Este artigo fornece explicações sobre os algoritmos e subrotinas, criados em MATLAB, que consistem em soluções para os exercícios do capítulo 4 do livro Robótica do autor John J. Craig (os algoritmos estão anexados numa pasta juntos com este artigo, aconselhamos a leitura em conjunto dos mesmos com este texto, para melhor compreensão).

2. EXERCÍCIOS DE PROGRAMAÇÃO PARTE 4

2.1 SOLUÇÃO

Desenvolvemos uma sub-rotina em MATLAB que recebe como argumentos o sistema de referência do punho em relação ao sistema de referência da base (WrelB), a posição atual do robô (*current*) que é um vetor de ângulos de juntas, e retorna a variável *sol*, que indica se as soluções foram encontradas, a solução mais próxima (*near*) e a segunda solução (*far*).

No escopo da função criada (*invkin*) são declaradas variáveis auxiliares que representarão o comprimentos dos elos 1 e 2 (linhas 16 e 17) bem como variáveis com valores de conversão para graus e radianos (linhas 19 e 20). Depois, utilizando as expressões definidas no capítulo 4 do livro, obtemos os valores dos ângulos procurados (linhas 55 e 56, por exemplo, onde obtemos os dois valores para θ_3) e, manipulando estruturas de decisão e repetição checamos as soluções (linhas 91 a 96, por exemplo, onde retornamos valores que indicam que não existe solução) e, com isso, obtemos os retornos procurados.

2.2 SOLUÇÃO

Criamos um sub-rotina em MATLAB que recebe como argumento o sistema de referência {T} especificado em relação a {S} (TrelS), uma matriz que corresponde ao sistema de referência da ferramenta (TrelW), uma matriz que corresponde ao sistema de referência da base do robô (SrelB) e a posição atual do robô (*current*) que é um vetor de ângulos de juntas. Essa sub-rotina retorna as soluções de ângulos de juntas (cinemática inversa).

No corpo da função, por meio de transformações, obtemos o sistema de referência {W} em relação a {B} (WrelB) e o passamos como argumento, assim como *current*, à função criada na solução 2.1 deste trabalho (*invkin*). Assim retornamos *sol*, *near* e *far* que fornecem as soluções, a solução mais próxima e a segunda solução, respectivamente.

2.3 SOLUÇÃO

Criamos uma sub-rotina que recebe um vetor de sistemas de referência meta (fornecidos pelo problema) e imprime os ângulos de juntas correspondentes ao mesmo. No escopo da função declaramos vetores que conterão os dados referentes à ferramenta (0.1, 0.2 e 30), estação (-0.1, 0.3 e 0) e *current* (0, 0 e 0). Depois, utilizando a função *utoi* (criada nos trabalhos anteriores) encontramos o sistema de referência da ferramenta (TrelW), estação (SrelB) e objetivo (TrelS) referentes aos vetores passados. Assim, em um for criamos uma expressão que imprime os ângulos de juntas correspondentes aos sistemas de referências meta informados.

A FIGURA 1 mostra os ângulos de juntas encontrados com a execução do programa.

```
>> principal
Ponto 1: uma solucao:
    Near: 57.01    115.26    67.73 graus
Ponto 2: duas solucoes:
    Near: -85.36    119.32    -18.96 graus
    Far : 33.96    -119.32    100.36 graus
Ponto 3: uma solucao:
    Near: 66.63    108.66    -85.30 graus
Ponto 4: sem solucao!
~~
```

FIGURA 1: Ângulos de juntas encontrados com a execução do programa.

3. EXERCÍCIOS PARA MATLAB 4

3.1 a) Solução

1)

$${}^0_H T = \begin{bmatrix} 1 & 0 & 0 & 9 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

então

$${}^0_3 T = {}^0_H T {}^3_H T^{-1} = \begin{bmatrix} 1 & 0 & 0 & 7 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Assim, concluímos os ângulos de juntas (TABELA 1).

Solução	$\Theta 1$	$\Theta 2$	$\Theta 3$
Subir ombro	0	0	0
Descer ombro	0	0	0

TABELA 1: Ângulos de juntas para o caso 1

2)

$${}^0_H T = \begin{bmatrix} 0,5 & -0,866 & 0 & 7,5373 \\ 0,866 & 0,6 & 0 & 3,2966 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

então

$${}^0_3 T = {}^0_H T {}^3_H T^{-1} = \begin{bmatrix} 0,5 & -0,866 & 0 & 6,5373 \\ 0,866 & 0,6 & 0 & 2,1946 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Assim, concluímos os ângulos de juntas (TABELA 2).

Solução	$\Theta 1$	$\Theta 2$	$\Theta 3$
Subir ombro	10°	20°	30°
Descer ombro	27,1°	-20°	52,9°

TABELA 2: Ângulos de juntas para o caso 2

3)

$${}^0_H T = \begin{bmatrix} 0 & 1 & 0 & -3 \\ -1 & 0 & 0 & 2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

então

$${}^0_3 T = {}^0_H T {}^3_H T^{-1} = \begin{bmatrix} 0 & 1 & 0 & -3 \\ -1 & 0 & 0 & 4 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Assim, concluímos os ângulos de juntas (TABELA 3).

Solução	$\Theta 1$	$\Theta 2$	$\Theta 3$
Subir ombro	90°	90°	90°
Descer ombro	163,7°	-90°	163,7°

TABELA 3: Ângulos de juntas para o caso 3.

3.1 b) Solução

Criamos um programa em MATLAB para a solução da questão apresentada. Iniciamos declarando matrizes que recebem os sistemas de referência da ferramenta (fornecidos pelo problema), variáveis auxiliares que conterão os comprimentos fixados, entre outros. Depois, em uma estrutura de decisão, manipulando as expressões definidas no livro imprimimos os ângulos $\Theta 1$, $\Theta 2$ e $\Theta 3$, que são a solução procurada.

A FIGURA 2 fornece o resultado da execução do algoritmo criado para o caso de entrada *i*.

ferramentaPlano =

```

1      0      0      9
0      1      0      0
0      0      1      0
0      0      0      1

```

punhoPlano =

```

1      0      0      7
0      1      0      0
0      0      1      0
0      0      0      1

```

Theta1 = 0.000, 0.000

Theta2 = 0.000, -0.000

Theta3 = 0.000, 0.000

FIGURA 2: Resultados encontrados com a execução do programa

A FIGURA 3 fornece o resultado da execução do algoritmo criado para o caso de entrada *ii*.

```

Command Window

ferramentaPlano =

    0.5000    -0.8660         0    7.5373
    0.8660     0.6000         0    3.9266
         0         0    1.0000         0
         0         0         0    1.0000

punhoPlano =

    0.5000    -0.8660         0    6.5373
    0.8660     0.6000         0    2.1946
         0         0    1.0000         0
         0         0         0    1.0000

Theta1 = 10.000,    27.114
Theta2 = 20.000,   -20.000
Theta3 = 29.999,    52.885

```

FIGURA 3: Resultados encontrados com a execução do programa

A FIGURA 4 fornece o resultado da execução do algoritmo criado para o caso de entrada *iii*.

```

Command Window

ferramentaPlano =

     0     1     0    -3
    -1     0     0     2
     0     0     1     0
     0     0     0     1

punhoPlano =

     0     1     0    -3
    -1     0     0     4
     0     0     1     0
     0     0     0     1

Theta1 = 90.000,    163.740
Theta2 = 90.000,   -90.000
Theta3 = -270.000, -163.740

```

FIGURA 4: Resultados encontrados com a execução do programa

A FIGURA 5 fornece o resultado da execução do algoritmo criado para o caso de entrada *iv*.

```

Command Window

ferramentaPlano =

    0.8660    0.5000         0   -3.1245
   -0.5000    0.8660         0    9.1674
         0         0    1.0000         0
         0         0         0    1.0000

punhoPlano =

    0.8660    0.5000         0   -4.8565
   -0.5000    0.8660         0   10.1674
         0         0    1.0000         0
         0         0         0    1.0000

O objetivo esta fora de alcance (fora da area de trabalho alcancavel) - nao existe solucao

```

FIGURA 5: Resultados encontrados com a execução do programa

3.1 c) Solução

Foi desenvolvido o algoritmo, porém não foi possível executá-lo, para a obtenção dos resultados (tivemos problemas para a instalação do toolbox), mesmo assim o algoritmo foi enviado junto com os arquivos (caso o professor deseje testá-lo).

4. CONCLUSÃO

Neste artigo desenvolvemos algoritmos que manipulam a cinemática inversa dos robôs no espaço, fornecendo conhecimento para a criação de aplicações para robôs industriais.

AGRADECIMENTOS

Agradecemos ao professor Ângelo pelas aulas ministradas e aos nossos familiares pelo apoio e compreensão.

REFERÊNCIAS

Craig, John J. "Robótica. 3ª edição." (2012).

MathWorks. Disponível em: <www.mathworks.com>. Acesso em: 04 dez. 2020.

MATLAB Avançado. Disponível em: <http://mtm.ufsc.br/~melissa/arquivos/matlabpet/aula_01.pdf>. Acesso em: 30 set. 2020.