# C:\Users\Pedro Tomás\Documents\IST\logo_IST_A.emf

Mestrado Integrado em Engenharia Electrotécnica e de Computadores

**Robótica**

**2º Semestre – 2015/2016**

Direct and Inverse Kinematics of Serial Manipulators

Realizado por:

João Borges n.º 75588

Rui Guerra n.º 75737

Lisboa, Março de 2015

# 1. Introdução

# O problema proposto neste trabalho consiste na representação da pose (posição e orientação) da ponta de um braço robótico (denominado *end-effector*) recorrendo apenas aos graus de liberdade dados pelos ângulos das juntas deste, bem como a determinação das possíveis combinações destes ângulos para obter uma certa pose.

Numa primeira fase, pretende-se determinar esta pose a partir de um conjunto de 6 graus de liberdade (θ1, ..., θ6) recorrendo apenas a transformações de coordenadas baseadas na cinemática do braço robótico.

Numa segunda fase, partindo de uma pose (x, y, z, α, β, γ) conhecida, pretende-se conhecer todas as possíveis combinações de ângulos θ para que esta pose se verifique.

Com vista a solucionar estes problemas, implementaram-se dois programas de MATLAB que permitem determinar a pose a partir dos ângulos e vice-versa.

# 2. Cinemática Directa

# O problema de determinar a pose do *end-effector* do braço robótico a partir dos ângulos (θ1, ..., θ6) é um problema de cinemática directa que, para ser resolvido, é necessário percorrer um conjunto de passos.

Em primeiro lugar, é necessário estabelecer os referenciais que correspondem a cada junta e definir as relações entre estes. Este passo é importante porque uma boa definição de referenciais será útil para a simplificação do cálculo a ser feito. A partir destes referencias será, então, criada uma matriz de transformação geral de onde se poderá extrair a pose do *end-effector*.

## 2.1. Estabelecimento de Referenciais

# Os referenciais foram escolhidos tendo em vista a utilização da convenção Denavit-Hartenberg (D-H) para uma determinação simplificada das transformações entre estes. Uma representação esquemática das posições relativas dos referenciais encontra-se representada na Figura 1.

Para além do referencial da base (0), introduziram-se referenciais correspondendo a cada grau de liberdade (1 a 6) de forma que o eixo de rotação correspondesse sempre ao eixo z do referencial. O sentido dos ângulos (θ1, ..., θ6) é também visível no esquema. De notar que os referenciais 1 e 2 bem como os A, 5 e 6 têm a sua origem no mesmo ponto, mas encontram-se representados separadamente para maior facilidade de observação.

Um referencial auxiliar (A) foi também utilizado para representar uma translação para ser possível utilizar a convenção D-H para determinar a transformação entre todos os referenciais, que de outra forma não seria possível.

## 20160324_140720.jpg

Figura 1 - Representação gráfica dos referenciais usados.

## 2.2. Cálculo da Matriz de Transformação

# A partir dos referenciais escolhidos, criou-se uma tabela com os parâmetros a ser utilizados pela convenção D-H. Esta encontra-se representada na Tabela 1.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| i | ai-1 | αi-1 | di | θi |
| 1 | 0 | 0 | A = 99 mm | θ1 |
| 2 | 0 |  | 0 | θ2 |
| 3 | B = 120 mm | 0 | 0 | θ3 |
| 4 | C = 40 mm | - | 0 | θ4 |
| A | 0 | 0 | D = 195 mm | 0 |
| 5 | 0 |  | 0 | θ5 |
| 6 | 0 | - | 0 | θ6 |

Tabela 1 - Parâmetros da convenção D-H obtidos por inspecção dos referenciais.

# Uma vez obtidos estes parâmetros, é possível escrever as matrizes de transformação entre referenciais consecutivos

# , , , , , , .

# Para simplificação de notação, as funções trigonométricas sin(θx) e cos(θx) serão abreviadas para sx e cx e as constantes relativas a distâncias serão representadas por A, B, C e D, como se indica na Tabela 1.

A matriz de transformação geral pode ser então obtida por



## 2.3. Obtenção da Pose do *End-effector*

# A partir da matriz é possível determinar os parâmetros da pose do *end-effector*. A posição (x, y, z) é obtida directamente a partir das entradas (1,4), (2,4) e (3,4), respectivamente, desta matriz. Para definir os parâmetros da orientação (α, β, γ) escolheu-se uma convenção de ângulos de Euler Z-Y-X. Segundo esta convenção, estes ângulos podem ser obtidos somente a partir da matriz de rotação (com entradas representadas por ) a partir das expressões

 para 

ou, caso   para  respectivamente.

## 2.4. Testes Experimentais

# Após implementar a função direct\_kinematics.m em MATLAB, realizaram-se diversos testes experimentais para verificar o bom funcionamento do programa desenvolvido. Na Figura 2 apresentam-se uma representação do braço robótico em três dimensões, no referencial 0, com os graus de liberdade assinalados por circunferências, bem como o output de função para dois exemplos, por ordem, (θ1, ..., θ6) = (0, 0, 0, 0, 0, 0) e

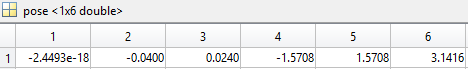
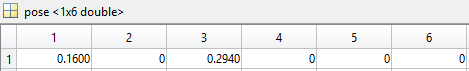
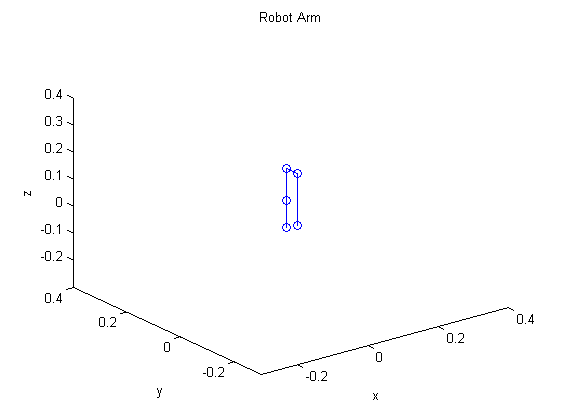
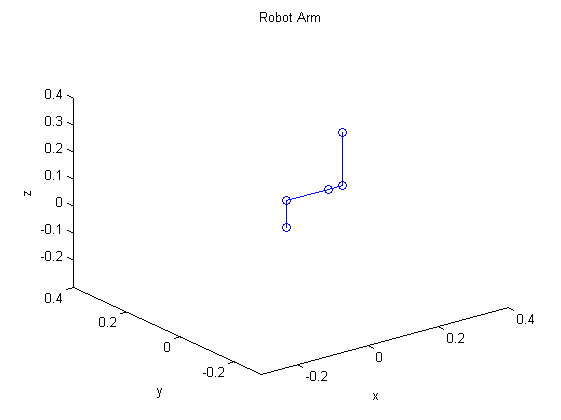


Figura 2 - Representação gráfica do braço robótico e output da função direct\_kinematics.m para dois exemplos.

É de notar que não é possível visualizar os efeitos dos três últimos graus de liberdade no gráfico devido a estes apenas influenciarem a orientação final do *end-effector*. De qualquer forma, o bom funcionamento do programa pode ser verificado de acordo com os referenciais estabelecidos anteriormente a partir do output da função.

# 3. Cinemática Inversa

# Determinar os valores dos ângulos (θ1, ..., θ6) do braço robótico a partir da pose do *end-effector* é um problema mais complexo, devido à multiplicidade de soluções e a complexidade do cálculo numérico para cada um dos ângulos.

## 3.1. Construção da Matriz de Transformação

# Em primeiro lugar, é necessário criar a matriz de transformação a partir da pose (x, y, z, α, β, γ) dada no input. Considerando novamente a convenção de ângulos de Euler Z-Y-X, é possível escrever esta matriz em função destes parâmetros através da expressão



## 3.2. Cálculo dos Ângulos das Juntas

# Utilizando é possível determinar todas as soluções possíveis para (θ1, ..., θ6). A fórmula geral usada para obter as expressões de cada θ foi



para diferentes valores de n escolhidos de forma a simplificar o cálculo simbólico para cada θ. As expressões são obtidas escolhendo uma ou duas entradas da matriz que resultem em equações simples que dependam de apenas um dos ângulos θ. Poe exemplo, as expressões obtidas para θ1 e θ3 foram:

 e  sendo

Os restantes ângulos têm expressões mais complexas mas que foram possíveis de obter apenas com dependências de outros ângulos já obtidos anteriormente. Estas poderão ser visualizadas no ficheiro inverse\_kinematics.m no MATLAB.

Como se pode observar, θ1 e θ3 têm duas soluções cada. Analisando os restantes ângulos também se poderá concluir que os ângulos θ4, θ5 e θ6 têm duas combinações de soluções. Assim, poder-se-á concluir que existem no total 8 soluções para (θ1, ..., θ6) que satisfaçam a pose desejada.

Contudo, é possível não haver solução para os ângulos, o que acontece quando a raiz quadrada utilizada no cálculo de θ3 produz um número imaginário. É possível também acontecer uma singularidade caso θ5 seja múltiplo de π. Neste caso, não é possível distinguir os efeitos de θ4 e θ6 pelo que os resultados obtidos são inconsistentes. Por fim, outra singularidade pode ocorrer caso  Quando isto sucede, θ1 poderá tomar qualquer valor para satisfazer a pose desejada.

## 3.3. Testes Experimentais

# De seguida demonstram-se vários exemplos de possíveis outputs para a função inverse\_kinematics.m desenvolvida. Na Figura 3 estão representadas as soluções obtidas para, em primeiro lugar, uma pose de entrada correspondendo à saída obtida no segundo exemplo da Figura 2, e para uma pose (0, 0, 0.01, 0, 0, 0).

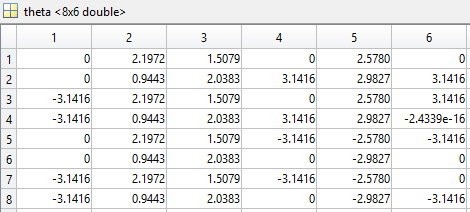
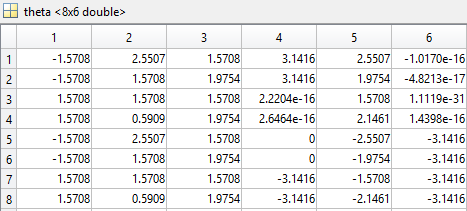


Figura 3 - Output da função inverse\_kinematics.m para dois exemplos.

Observa-se o bom funcionamento do programa desenvolvido no primeiro exemplo, já que a entrada usada na cinemática directa corresponde à terceira solução obtida, a menos de erros de arredondamento. No segundo exemplo observa-se o caso de existir a singularidade  Neste caso é dado um aviso a indicar que θ1 pode tomar qualquer valor para além dos indicados na solução.

No caso de existir a singularidade de θ5 ser múltiplo de π ou o ponto ser inválido (por ser impossível o braço robótico chegar a essa posição), o output da função é simplesmente o valor -1.

# 4. Instruções de Utilização das Funções MATLAB

# A função direct\_kinematics.m recebe como argumento um vector de dimensão 6 correspondendo aos ângulos (θ1, ..., θ6). A saída é também um vector de dimensão 6 correspondendo à pose (x, y, z, α, β, γ) do *end-effector* de acordo com a convenção de ângulos de Euler Z-Y-X. Um exemplo de utilização desta função é utilizando o comando "pose=direct\_kinematics(theta);", se theta corresponder ao vector de ângulos.

# A função inverse\_kinematics.m recebe como argumento um vector de dimensão 6 indicando a pose do *end-effector* desejada (x, y, z, α, β, γ) usando a mesma convenção. A saída é uma matriz de 8 linhas por 6 colunas correspondendo às 8 soluções possíveis de ângulos (θ1, ..., θ6), ou então apenas um valor igual a -1 caso o ponto seja inválido ou exista a singularidade de θ5 ser múltiplo de π. Um exemplo de utilização é o comando "theta=inverse\_kinematics(pose);".

# 5. Conclusões

# No trabalho realizado foi possível resolver o problema de determinar a pose do *end-effector* do braço robótico a partir dos ângulos das suas juntas (θ1, ..., θ6), bem como o problema inverso. Após a definição dos referenciais a usar e a escolha da convenção de ângulos, o problema a resolver resumiu-se em grande parte à realização de manipulações algébricas.

O resultado obtido foi duas funções MATLAB, direct\_kinematics.m e inverse\_kinematics.m, que produzem todas as soluções possíveis para cada problema, detectando singularidades e entradas inválidas.