

Robótica

Linguagens de Programação para Robótica (manipuladores)

- Orientados para o Robô
 - Tarefa descrita como sequência de movimentos
 - Ex: VAL (Puma), AL (IBM)...
- Orientada para as Tarefas (ou para os Objectos)
 - Descrição de posicionamentos de objectos e de objectivos

Características de linguagens nível robô:

- Setup do espaço de trabalho e posicionamento dos objectos
- Definição de "POSES" (posição + orientação) usando as Estruturas de dados da Linguagem
- Decomposição da tarefa em sequências de acções
- Acrescento de comandos sensoriais para as excepções
- "debug"

LEIC/2003

Eugénio Oliveira

Robótica

Estruturas:

Quadros de Coordenadas ("coordinate frames")

Representados por matrizes de transformação homogénea de 4×4

Especificando a orientação e (um vector) a posição relativa ao quadro de base

Podem ser coordenadas cartesianas nas matrizes e vectores

Operadores. Multiplicação
 TRANSFORMAÇÃO
 constituída por uma Rotação e uma Translacção

Programador tem controlo sobre detalhe dos movimentos:

Velocidade, Aceleração, desaceleração, ...

Ex: "WITH v=..."

Variáveis para valores extraídos dos sensores

Ex: Force (eixo) > 3

LEIC/2003

Eugénio Oliveira

Robótica

Embora os Quadros de Coordenadas sejam vulgares para representar uma configuração robótica, a maneira mais natural de o fazer é no espaço das Junções e não no espaço cartesiano

Isto porque o “problema cinemático inverso” não dá soluções únicas.
Isto é: dado um ponto cartesiano no espaço, a configuração do manipulador Robótico não tem uma solução única.

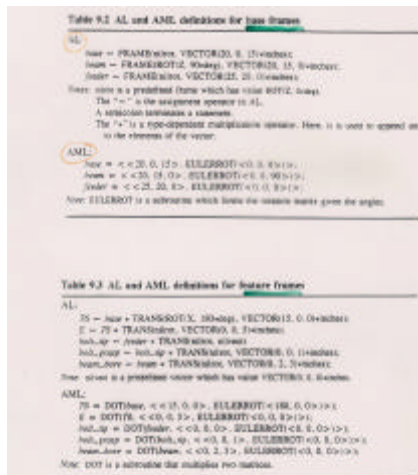
Quando se especifica um movimento deve dar-se pontos intermédios que Permita gerar uma trajectória livre de colisões

LEIC/2003

Eugénio Oliveira

Robótica

Definições de Quadros de Coordenadas



LEIC/2003

Eugénio Oliveira

Robótica

Exemplos de instruções sobre movimentos:

Table 9.4 Examples of AL and AML motion statements

AL:

```
{ Move arm from rest to frame A and then to bolt_grasp }
MOVE barm TO A;
MOVE barm TO bolt_grasp;

{ Another way of specifying the above movement }
MOVE barm TO bolt_grasp VIA A;

{ Move along the current Z axis by 1 inch. i.e., move relative }
MOVE barm TO @ - 1*Z=inches;
```

Notes: *barm* is the name of the robot arm.
@ indicates the current location of the arm which is equivalent to base + θ * E.
Statements inside brackets { . . . } are comments.

AML:

```
-- Move joint 1 and 4 to 10 inches and 20 degrees, respectively (absolute move)
MOVE(<1, 4>, <10, 20>);

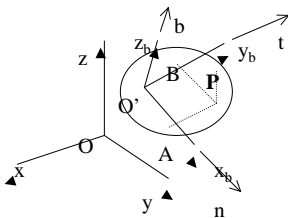
-- Move joints 1, 3 and 6 by 1 inch, 2 inches, and 5 degrees, respectively (relative move)
DMOVE(<1, 3, 6>, <1, 2, 5>);
```

Notes: Statements preceded by "--" are comments.

LEIC/2003

Eugénio Oliveira

Robótica



Referencial Fixo: origem O -x,y,z

Referencial da Base do Objecto O' é o vector $X_0 = \{x_0, y_0, z_0\}$

-Orientação segundo os eixos respectivos é dada pela matriz $R = [n, t, b]$

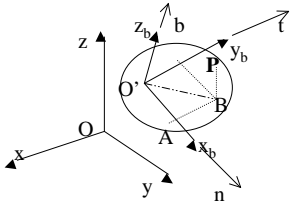
-Onde n,t,b são vectores unitários nas direcções x_b, y_b, z_b sendo os componentes os cosenos directores.

- O ponto P é representado pelo vector relativamente a O-x,y,z

LEIC/2003 $X = \{x, y, z\}$

Eugénio Oliveira

Robótica



-O ponto P é representado relativamente a O' - x_b, y_b, z_b
 $X^b = \{u, v, w\}$ (coordenadas em O' - x_b, y_b, z_b)

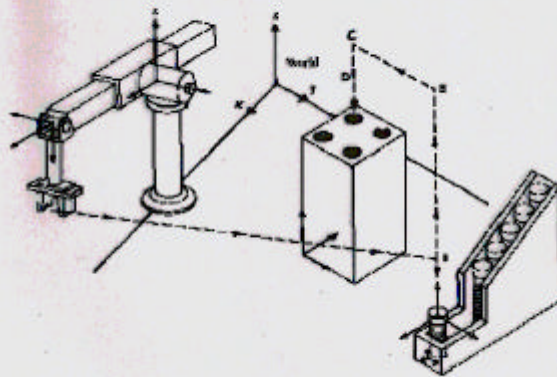
$$\vec{OP} = \vec{OO'} + \vec{O'A} + \vec{AB} + \vec{BP}$$

$$X = X^0 + u*n + v*t + w*b = X^0 + R * X^b$$

Conclusão: transformação de Coordenadas do corpo X^b para as coordenadas fixas X é feita em termos de X^0 e R . R representa a posição e orientação do corpo rígido (ou seu referencial) relativamente ao ponto fixo da origem O

Robótica

Exemplo de Tarefa de um Robô Manipulador



PROGRAMA EM "AL" PARA UMA TAREFA DE INSERÇÃO

BEGIN inserção

{ Definir as variáveis }

diametro_parafuso <--- 0.5*inches ;

altura_parafuso <--- 1*inches ;

tentativas <--- 0 ;

apanhado <--- false ;

{Definir as frames de base}

suporte <--- FRAME(ROT(Z,90*deg), VECTOR(20,15,0) *inches) ;

alimentador <--- FRAME(nilrot, VECTOR(25,20,0) *inches) ;

{Definir frames das partes dos objectos}

encaixe <--- alimentador * TRANS(nilrot,nilvect) ;

cabeça_parafuso <--- encaixe * TRANS(nilrot, VECTOR(0,0,0.5) * inches) ;

orificio <--- suporte * TRANS(nilrot, VECTOR(1.3,1) * inches) ;

{Definir frames dos pontos de passagem}

A <--- alimentador * TRANS(nilrot, VECTOR(0,0,0.5) * inches) ;

{Abrir a garra }

OPEN barm TO diametro_parafuso + 1 * inches ;

{Posicionar a garra}

MOVE garra TO cabeça_parafuso VIA A

WITH APPROACH = -Z WRT alimentador ;

{Tentativa de apanhar o parafuso}

DO

CLOSE barm TO 0.9 * diametro_parafuso ;

IF garra < diametro_parafuso

THEN

{falhou o agarramento; tentar de novo}

BEGIN

OPEN garra TO diametro_parafuso + 1 * inches ;

MOVE barm TO @-1 * Z * inches ;

END

ELSE

apanhado <--- true ;

tentativas <--- tentativas + 1 ;

UNTIL apanhado OR (tentativas > 3) ;

{Abortar a operação caso não chegaram tres tentativas }
IF NOT apanhado THEN ABORT ("Não conseguiu apanhar parafuso") ;

{Mover o braço para B}
MOVE barm TO B
VIA A
WITH DEPARTURE = Z WRT alimentador ;

{Mover o braço para D}
MOVE barm TO D VIA C
WITH APPROACH = -Z WRT orificio ;

{Testar se está bem posicionado}
MOVE barm TO @ -0.1 * Z * inches
ON FORCE (Z) > 10 * ounces DO ABORT("Mal posicionado") ;

{Inserir em conórdância}
MOVE barm TO orificio DIRECTLY
WITH FORCE(Z) = -10 * ounces
WITH FORCE(X) = 0 * ounces
WITH FORCE(Y) = 0 * ounces
WITH DURATION = 5 * seconds ;

Robótica

Características de Linguagens nível tarefa:

Em vez de descrever a tarefa por movimentos, descreve-se em termos dos objectos manipulados

Implica a existência de um Planeador e de um Modelo do Mundo
Transforma a especificação em linguagem de nível robô

4 fases: Modelação do Mundo,
 Especificação da Tarefa
 Síntese do Programa
 Execução

Robótica

Modelo do Mundo implica descrições das propriedades dos Objectos:
físicas
geométricas

Representação actualizada do Mundo

O Modelo geométrico inclui informação espacial: dimensão, volume, forma...

Programas CAD permitem a representação por.

arestas e vértices
superfícies
cilindros generalizados (CSG)
Decomposição celular

Outras informações importantes:

Estados estáveis
Centro de massa
Áreas de prensão

LEIC/2003

Eugénio Oliveira

Robótica

Especificação da tarefa:

ideal(?) Linguagem Natural
Seqüência de estados (de Inicial a Final) em lógica de predicados
Seqüência de operações simbólicas

Especificação da tarefa:

Seqüência de estados (de Inicial a Final)

Estado inicial

On(Ob1(Face1), Mesa)
On(Ob2(Face1), Ob1(Face3))
On(Ob3(face1), Mesa)

Estado final

On(Ob1(Face1), Mesa)
On(Ob2(Face1), Ob1(Face3))
On(Ob3(Face1), Ob2(Face3))



LEIC/2003

Eugénio Oliveira

Robótica

Síntese de Programas:

Implica planeamento mais fino de como:

“agarrar” e “largar”

(geometria, faces acessíveis, paralelas às garras)

estabilidade

movimentar sem colisões