

Exercícios de programação em TP no manipulador Fanuc

Introdução

A consola é utilizada para programar o manipulador de forma rápida e simples. Possui uma linguagem de programação própria, a TP. Permite ainda fazer jog manual (movimentação manual do manipulador) em vários sistemas de eixos.

Entre alguns dos elementos principais da consola são o “dead men switch”, botão amarelo localizado nas costas da consola, que funciona como botão de alarme pois tem que estar a ser pressionado para que o robô se movimente. A mesma funcionalidade têm o botão “shift”.

Quando existe um erro é necessário limpá-lo pressionando para isso os botões “shift” e “reset” simultaneamente.

Para navegar nos menus são utilizadas as setas e a tecla enter. Em cada menu aparecem no visor da consola opções contextuais que podem ser seleccionadas recorrendo às teclas F1 a F5.

O jog manual do manipulador é executado premindo o “dead man switch” e o “shift” (condição necessária para qualquer movimentação) junto com as teclas azuis -X X+ etc. Note-se que pode ser alterado o sistema de eixos com a tecla “coord” assim alterando as características do movimento.

As teclas com o simbolo % impõem a velocidade das movimentações.

Linguagem TP

Para criar um programa basta carregar em “select”. Aqui aparece uma lista dos ficheiros já existentes. Uma das opções contextuais é o “create”. Os passos seguintes são muito intuitivos sendo apenas necessário colocar o nome e pressionar enter para passar à edição. Em algumas consolas aparece uma opção de selecção das “groupmasks”. Neste caso deve ser colocado 1 no primeiro campo e * em todos os outros.

A linguagem TP é bastante simples sendo a sua estrutura básica a seguinte:

J P[1] 100% FINE

Em que J significa que o manipulador se vai mover para o ponto P[1] utilizando a trajectória mais imediata (movimento controlado pelas juntas). O valor da percentagem é o da velocidade que o manipulador terá ao ir para P[1]. Estas instruções indicam uma movimentação entre dois pontos: daquele em que o robô está para o ponto definido em P[1].

Para além do movimento de juntas existem ainda movimentos lineares ou circulares entre pontos.

O argumento FINE indica que o ponto será alcançado com precisão. No caso de ser, por exemplo, um ponto de passagem, pode ser utilizado o CNT que permite que o robô não pare nesse ponto e passe “perto” dele.

Para definir pontos de passagem é necessário fazer o chamado TOUCHUP. Este implica que o utilizador coloque o robô na posição P desejada e depois grave essa posição.

Exercício 1 – Definir dois pontos de trabalho e executar Trajectórias J e L entre eles.

Criar o programa "Test201"

Utilizando o *jog* manual armazenar dois pontos P[1] e P[2] com o comando TOUCHUP. No mesmo programa fazer um ciclo (utilizar os comandos LABEL e GOTO) em que o robô vai de P[1] para P[2] com um movimento de juntas e de P[2] para P[1] em trajectória linear.

Código exemplo:

```
1: LBL[1] //Label 1
2: L P[1] 1000mm/sec FINE //Vai para P[1] em mov. linear
3: J P[2] 100% FINE //Vai para P[2] em mov. de juntas
4: JMP LBL[1] //Vai linha com label 1
```

Exercício 2 – Execução de um trajectória entre dois pontos em coordenadas relativas

Criar o programa "Test202"

Neste exemplo pretende-se adaptar o programa anterior de modo a criar um novo ponto PR[3], que tenha os valores de (X, Y) de P[1] e o valor de Z de P[2].

Para isso, é necessário guardar num ponto de referência os valores (X,Y,Z) dos pontos:

PR[3] = LPOS

Depois de executado este comando, PR[3] passa a ser uma estrutura com 6 campos correspondentes às coordenadas (X,Y,Z) e às orientações do *end effector*.

Para guardar um destes valores numa variável real faz-se:

R[70] = PR[3, 3]

Para alterar apenas um dos campos de um PR[] faz-se:

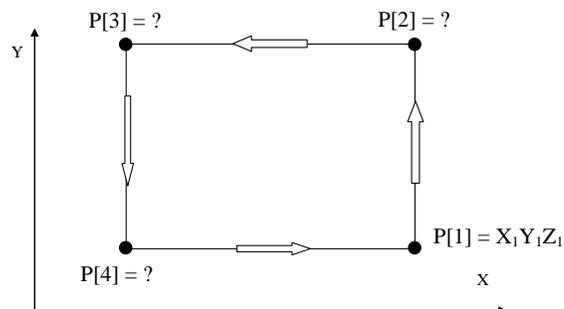
PR[3,campo] = 30

Depois de criar o PR[3], acrescentar ao programa uma linha para ir até PR[3]

Exercício 3 – Fazer a trajectória de um rectângulo a partir de um ponto inicial

Criar o programa "Test203"

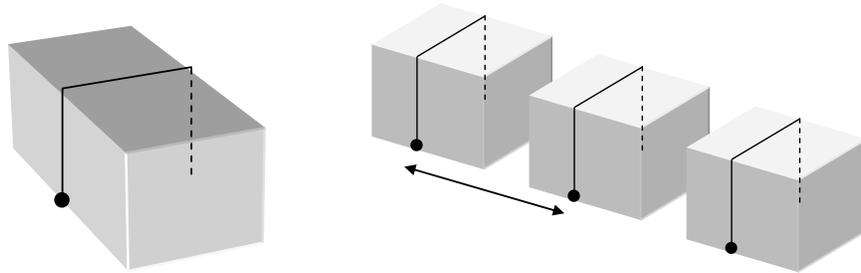
Utilizando as funcionalidades atrás descritas, pretende-se fazer um programa que ponha o robô a executar um rectângulo de 300 X 400 mm no plano XY com base num ponto inicial . Ilustração da trajectória requerida:



Exercício 4 – Simulação da realização de linhas nas faces de três caixa com base num único ponto inicial

Criar o programa "Test204"

Recorrer a uma subrotina para executar um comando repetitivo cujo início é função das coordenadas originais onde começar o traçado das linhas. Os parâmetros são: o ponto de partida, as medidas das duas faces relevantes, a separação das caixas.

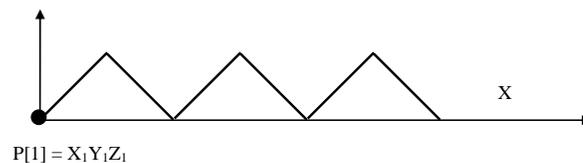


Criar uma rotina (programa comum para fazer um U invertido). É muito semelhante ao retângulo anterior.

Exercício 5 – Programa para simular movimento triangular

Criar o programa "Test205"

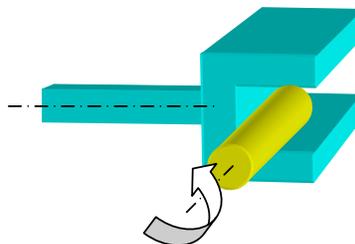
Fazer um programa que execute uma linha poligonal de forma triangular ao longo de uma linha recta conforme ilustrado de seguida:



Exercício 6 – Execução de orientações a 3D

Criar o programa "Test206"

Fazer um programa para incrementar sequencialmente (passos de 10°) a orientação de um cilindro ao longo do seu eixo de revolução mas admitindo que a garra segura no cilindro com o seu eixo de aproximação (eixo z da ferramenta) ortogonal ao eixo de revolução do cilindro:

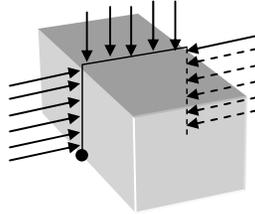


Admitindo que o eixo colinear com o eixo de revolução do cilindro é x.

Exercício 7 – Programa das linhas na caixa mas mantendo a ortogonalidade do end-effector

Criar o programa "Test207"

Exercício similar ao exercício anterior sobre as linhas na caixa, mas agora o end-effector deverá estar sempre normal às superfícies do objecto.



Admitindo que o eixo colinear com o eixo de revolução da ferramenta é sempre o eixo x, o que se pode garantir fazendo um touchup com isso em mente. Basta adicionar na rotina FAZERU as rotações em x em cada vértice do U.

Exercício 8 – Simulação de polimento de pedra em movimentos circulares cíclicos

Criar o programa "Test208"

Partindo de um ponto P1 genérico, realizar ciclicamente uma trajectória feita de meias circunferências, conforme ilustrado de seguida. O programa só dependerá de P1 e do diâmetro dos arcos circulares.

Numa segunda fase do exercício, condicionar a execução com base numa entrada digital do sistema que se pode simular através da consola.

