

Programação Gráfica

Neste capítulo procuraremos ilustrar o que se entende por Programação Offline de Robots e caracterizar quais as suas vantagens relativamente à programação convencional com uma “Teach Pendant”.

Procurar-se-à também focar quais os problemas associados e requisitos para implementar um sistema de programação offline.

Caracterização

A programação offline de um robot consiste em desenvolver um programa sem ocupar o robot, e poderá ser efectuada quer textualmente com o auxilio de um editor de texto normal ou software dedicado, quer graficamente com um programa que simule o robot e a célula de trabalho.

Como exemplos de sistemas offline textual e gráfico apontam-se o programa OLP3 da ABB e o programa **ROBOT3D**.

Os sistemas de programação offline textuais não incluem, na generalidade, a cinemática do robot e apenas permitem a introdução das instruções que compõem o programa com um controlo de sintaxe.

Por outro lado têm de ser conhecidas de antemão as coordenadas dos pontos característicos da tarefa a executar e a orientação do órgão terminal.

Estes sistemas não entram em linha de conta com os atravancamentos causados pelos objectos existentes na área de trabalho e com a possibilidade de se usar uma ou outra ferramenta.

Para se conseguir desenvolver um programa de trabalho de robot operacional é geralmente obrigatório recorrer a um sistema offline gráfico. Só deste modo é possível analisarem-se a capacidade de o robot alcançar a peça sem violar os limites mecânicos das juntas, os atravancamentos da célula, a orientação da pinça no espaço 3D e mesmo a solução adequada da transformação cinemática inversa.

PROGRAMAÇÃO OFFLINE

Os programas gráficos permitem geralmente calcular os tempos de ciclo, garantir a perpendicularidade a superfícies, e algumas funções não disponíveis em certos controladores como a execução de paletes e *mirrors*.

Estes programas dispõem geralmente de bases de dados de CAD dos robots e posicionadores dos fabricantes mais representativos.

O utilizador poderá completar esta base de dados introduzindo as robots, ferramentas e peças necessárias.

Um simulador gráfico permite ainda de uma forma muito eficiente testar e corrigir a posição dos elementos da célula que mais se adequa à execução do programa de trabalho, e na hipótese de se poderem utilizar vários robots, determinar qual o mais indicado para a tarefa em causa.

É também possível simular a movimentação de posicionadores, eixos externos e de objectos dentro da célula.

Requisitos

O desenvolvimento de um programa offline com o auxílio de um simulador gráfico presuppõe a existência de modelos tridimensionais do robot, posicionadores, ferramentas e peças. Estes modelos poderão ser do tipo *wireframe* (arames). ou sólidos.

Só com modelação de sólidos é possível determinar as colisões entre os vários componentes na célula de trabalho e obterem-se imagens realísticas por *shade* e *rendering*.

Um modelador de sólidos permite associar primitivas de objectos simples ex. CONE, BOX, através de operações booleanas como uniões e intersecções de modo a constituir objectos mais complexos (ex. uma placa com um furo cilíndrico).

Para a modelação destes objectos é geralmente necessário recorrer a um módulo de CAD que poderá acompanhar o package ou transferir os modelos a partir de um programa de CAD externo, mediante um formato adequado (IGES, DXF).

Para uma correcta simulação torna-se também necessário conhecer a transformação de coordenadas inversa para o robot que pretendemos utilizar. Esta transformação permite calcular a(s) configuração(ões) das juntas do robot a partir das coordenadas de posição e orientação.

PROGRAMAÇÃO OFFLINE

Os sistemas comerciais offline mais sofisticados incluem transformações de coordenadas para os robots industriais mais usuais. Em certos casos o próprio utilizador terá de deduzir a transformação de coordenadas específica do robot a utilizar.

Vantagens

As vantagens que advêm da utilização de um sistema gráfico para a programação de robots são as globais a qualquer sistema offline como:

- desenvolvimento seguro de programas sem ocupar o robot
- afastamento do operador de ambientes potencialmente perigosos
- minimização do risco de danificar o robot, as ferramentas e peças durante a fase de programação.

Por outro lado é possível:

- visualização dos movimentos do robot e posicionadores
- operação das pinças
- verificação da área de trabalho e colisões
- cálculo de tempos de ciclo
- ajuste do *layout* da célula
- projecto de pinças e *gabarits*
- projecto de peças adaptadas à robotização
- avaliação de diferentes robots para a mesma tarefa
- aprendizagem de apenas uma única linguagem neutra de programação offline mesmo que haja a necessidade de programar diversos robots de fabricantes distintos.

A possibilidade de se testarem várias soluções para gabarits através do CAD evita erros de concepção onerosos e atrasos significativos na construção dos gabarits.

É ainda possível o treino e formação de operadores e programadores.

No sistema **ROBOT3D** um operador com um conhecimento básico de computadores (DOS, editor), CAD (apenas para modelação) e com alguma experiência em programação on-line de robots está apto a produzir programas offline em apenas cerca de 16 horas.

PROGRAMAÇÃO OFFLINE

Sistemas Comerciais

Existem diversos sistemas comerciais disponíveis no mercado que diferem pelo grau de sofisticação, plataformas utilizadas e pelo preço.

Assim podem-se encontrar sistemas por *wireframe* ou modelação de sólidos, com interfaces para diversas aplicações tecnológicas como soldadura por resistência e por arco, manipulação, pintura, colagem etc, com valores que podem variar entre as poucas centenas e os milhares de contos.

As plataformas utilizadas são ou estações UNIX, ou menos frequentemente sistemas DOS. O sistema **ROBOT3D** a correr em DOS agrupa características importantes, aparentemente antagónicas, como:

- Investimento de baixo custo (*software e hardware*)
- Elevado desempenho
- Facilidade de utilização

representando uma abordagem bastante produtiva relativamente à necessidade de programação offline de robots.

Outro produtos conhecidos nesta área são:

- ROBCAD
- GRASP
- OPS
- SL ROBOTIK
- WORKSPACE

Desenvolvimento de Programas

O desenvolvimento de programas segue uma lógica idêntica ao do on-line utilizando um robot, geralmente diferindo apenas no modo de conclusão do programa que habitualmente se completa num editor de texto.

Como já foi referido, é necessário numa primeira fase efectuar a modelação geométrica das peças ou subconjuntos para que se vai efectuar o programa de trabalho.

Nesta fase, bastante significativa em termos de tempo, pode-se poupar bastante trabalho efectuando a modelação mais simples possível e

PROGRAMAÇÃO OFFLINE

desprezando todos os pormenores, que não sejam funcionais (ex. raios de curvatura em chapas de zonas não alcançáveis pelo programa).

Caso existam eixos externos é necessário ainda fazer o acoplamento entre o robot e os eixos externos.

À modelação dos elementos segue-se o seu posicionamento na célula de trabalho.

Ao desenvolver um programa offline começa-se por se analisar a tarefa no sentido de se determinarem partes repetitivas do programa que serão incluídas em subprogramas.

Segue-se uma fase de determinação dos pontos de aproximação e dos pontos do ciclo recorrendo-se à movimentação do robot eixo-a-eixo ou por coordenadas através de menus adequados que se assemelham geralmente à forma de uma consola de programação de robot (*teach-pendant*).

O posicionamento rigoroso poderá ser efectuado por determinação das coordenadas dos pontos de posicionamento recorrendo à base de dados de CAD, ou em certos casos por visualização em diferentes vistas e com *zoom's* adequados. Para cada ponto importa ainda reorientar a ferramenta de modo correcto.

Nesta fase são analisadas todas as possíveis colisões e atravancamentos e determinada qual a configuração das juntas mais adequada (solução da transformação cinemática inversa). É ainda durante esta fase que se poderá determinar qual a posição do posicionador e eixos externos (caso existam) que facilita mais a tarefa do robot.

Os tipos de movimentos a efectuar (linear, ponto-a-ponto ou circular) são determinados também nesta fase, de modo a se obter o programa mais eficiente.

A introdução da lógica do ciclo, com as chamadas dos subprogramas e os saltos, será introduzida posteriormente, através da interface de programação, ou com um editor de texto.

Instruções auxiliares que têm a ver com o processo (ex. fecho de pinças, arranque de arco eléctrico) poderão ser também introduzidas no programa, bem como instruções para processar entradas e saídas digitais e/ou analógicas.

Verificação de Programas

Finda a fase de construção do programa com o sistema de programação offline torna-se necessário simular a sua execução com vista à determinação de eventuais erros que tenham passado na fase anterior.

É possível durante a verificação determinar-se o tempo de ciclo e eventualmente reduzi-lo por optimização de alguns movimentos.

É ainda importante verificar a execução de todos os movimentos e alterar alguns que sejam impossíveis de realizar (ex. passagem por uma singularidade durante um movimento linear).

Após a verificação do programa offline, por simulação, este deverá ser pós-processado, ou seja convertido da linguagem neutra do sistema offline para a linguagem do controlador do robot, findo o qual o programa deverá ser transmitido ao controlador do robot por um módulo de comunicações ou através de disquete (*DOWNLOAD*).

Calibração

Concluído e verificado o programa offline por simulação torna-se necessário fazer a compatibilidade do programa com a realidade física.

Os desvios relativamente ao modelo teórico são provenientes de:

- Erros de zero mecânico do robot.
- Tolerâncias e folgas na estrutura do robot
- Falta de rigidez na estrutura do robot
- Resolução e exactidão numérica do controlador
- Dificuldade de efectuar um levantamento preciso da posição dos componentes na célula
- Falta de rigor dos manequins
- Variações ambientais (ex. temperatura)
- Precisão numérica do sistema de programação offline

Para isso são estabelecidos um ou mais pontos de referência no programa offline que posteriormente são testados de modo a ser determinado o desvio entre a realidade física e o programa produzido offline.

PROGRAMAÇÃO OFFLINE

Determinados os desvios, são efectuadas correções nos eixos X,Y,Z do sistema da base do robot (ex. instrução FRAME nos sistemas ABB e ZOF nos sistemas RCM). A linguagem neutra do **ROBOT3D** inclui instruções para deslocamento de referenciais. Uma outra abordagem consiste em efectuar uma correção imediata da listagem actuando nas coordenadas. O editor **RODIT** que faz parte do *package* **ROBOT3D** possui rotinas para efectuar este processamento.

Teste e Correções On-line

O programa deverá ser passado “em branco”, isto é sem soldadura nas aplicações de soldadura por arco e por resistência, e corrigido se necessário.

Nas aplicações de soldadura só após esta fase deverão ser corrigidos os parâmetros de processo (ex. Voltagem, Intensidade de corrente).

Simulação de Programas feitos On-line

Torna-se também necessária a simulação e correção de programas feitos on-line.

Para isso é necessário transferir os programas do controlador para o sistema offline. Esta operação é conhecida por *UPLOAD* e envolve a conversão da linguagem nativa do controlador para a linguagem neutra do sistema offline (pré-processamento)

As figuras seguintes ilustram o processo de desenvolvimento de programas offline.

PROGRAMAÇÃO OFFLINE

PROGRAMAÇÃO OFFLINE

Estrutura de um Simulador Gráfico

Nesta secção veremos como é que interagem os diversos módulos que compõem um sistema de simulação gráfico de robots.

Na figura seguinte apresenta-se um diagrama que mostra as diversas relações existentes entre os módulos de software e as bases de dados de entrada e saída. Esta figura ilustra apenas os aspectos funcionais e não implica nenhuma separação física entre estes elementos.

Por exemplo os blocos da figura são geralmente subrotinas num largo sistema integrado.

Muitos dos conceitos ilustrados aqui serão desenvolvidos no capítulo 4, onde se foca a operação do **ROBMOD**, o módulo de Simulação e Programação Offline.

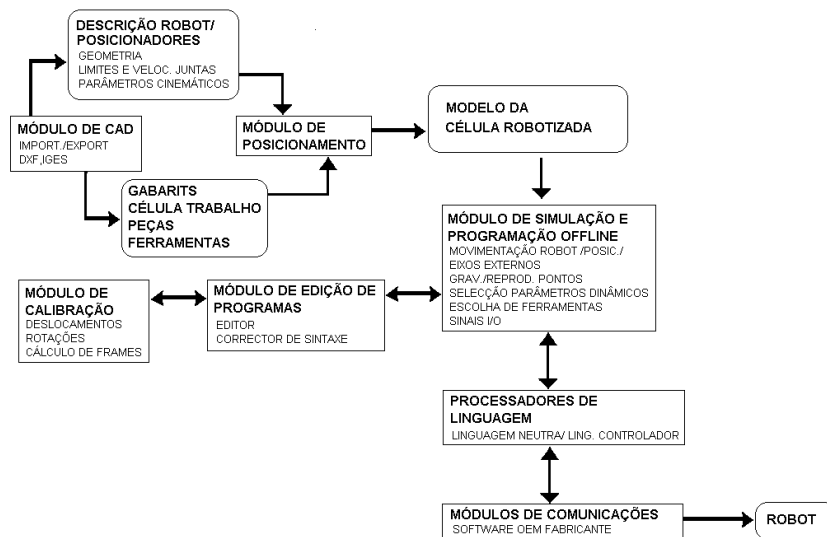


Fig. 3. Estrutura de um Simulador Gráfico

O primeiro requerimento de um simulador de robots é o de ter possibilidade de especificar qual o robot a ser simulado.

Conceptualmente a representação de um robot é dividida em duas componentes: uma descrição funcional dos seus parâmetros cinemáticos e dados associados, como os limites das juntas; e uma descrição física da geometria actual dos elos da cadeia do robot, actuadores etc.

PROGRAMAÇÃO OFFLINE

Um módulo de software que possa definir e modificar os parâmetros cinemáticos é útil na determinação de limitações ao espaço de trabalho e é de capital importância durante a fase de projecto. A modelação da geometria física do robot é levada a cabo por software de CAD para engenharia mecânica.

Este software de modelação serve um duplo propósito dado que também serve para modular o produto final e os gabarits com que o robot vai operar. A descrição funcional e geométrica do robot geralmente é guardada numa biblioteca de robot existentes no mercado. Isto simplifica o processo de selecção e permite flexibilizar a escolha de um dado robot para uma dada tarefa ou arranjo de célula.

A possibilidade de modificar as especificações dos robots continua a ser importante, no entanto, dado que os modelos *standart* são geralmente adaptados para aplicações especiais.

Dado que uma das maiores aplicações dos simuladores gráficos é modelar a integração de robots num processo de fabrico é necessário ter software que possa ajudar na especificação do arranjo da célula. As entradas para um programa deste tipo são a descrição completa do robot, gabarits, e outros componentes requeridos pelo processo e uma descrição do próprio produto.

O posicionamento do robot e outros objectos móveis e sistemas de movimentação pode então ser especificado. Este procedimento de posicionamento dos objectos é em parte iterativo dado que várias soluções são testadas até que seja encontrada uma solução óptima. Isto pode implicar a utilização de geometrias de robots alternativas, ilustrando a interacção entre o módulo de software descrito anteriormente.

O posicionamento estático dos objectos dentro da célula é apenas uma pequena parte do projecto. A maior responsabilidade dos sistemas de simulação de robots é modelar características dependentes do tempo para assegurar uma correcta operação. Isto requer uma descrição geométrica de todos os objectos existentes na célula, incluindo o modo como mudam com o tempo; a informação da cinemática, dinâmica, do controlador e da tarefa a ser executada. É este o nível a que corresponde a programação online do robot real.

A simulação do robot por meio da programação offline apresenta uma série de vantagens em termos de custo, tempo e segurança. O nível em que o projectista interage com este módulo varia com a sofisticação do sistema. Muitos sistemas usam a técnica de menus, que se assemelha ao ensinamento de um robot com uma consola dado que apenas escolhas discretas podem ser efectuadas.

PROGRAMAÇÃO OFFLINE

O sistema por menus de um simulador é, no entanto, mais poderoso dado que certas opções correspondem a operações não facilmente executáveis com o um robot real.

Muitos simuladores usam uma linguagem de programação neutra para especificar a movimentação do robot. Esta linguagem pode ser relativamente de baixo nível, requerendo apenas o posicionamento do órgão terminal. Linguagens mais sofisticadas podem levar em linha de conta toda a informação disponível da célula de trabalho, do produto, e controlo do processo, e produzir independentemente a trajectória do órgão terminal a partir de comandos de alto nível.

Novamente a optimização dos movimentos é mais ou menos um procedimento iterativo que poderá requerer modificação da trajectória do robot ou rearranjo da célula. Pode mesmo ser necessário voltar atrás e experimentar um outro robot. Uma vez o processo completo a maior parte dos simuladores pode traduzir os comandos específicos de movimentação para um controlador específico (pós-processamento) para *download* no robot existente na célula.

O último módulo de software usado concorrentemente em todos os procedimentos anteriores, é o display gráfico e o software de animação.

Enquanto se passam pelos anteriores estágios de design o projectista usa os gráficos por computador para visualizar os progresso interactivamente. Dados gravados podem mesmo ser pós-processados para serem produzidas representações mais realísticas para documentação.

As simulações gráficas permitem ao projetista pegar em modelos geométricos e visualizá-los a três dimensões segundo qualquer ângulo e grau de detalhe.

Deste modo são usadas durante a fase inicial de modelação CAD para fornecer feedback visual no produto ou especificação do robot. No estágio seguinte, proporcionam a interface ideal para avaliar o projecto da célula.

Com a informação das variáveis dependentes do tempo do módulo de controle, podem-se produzir animações para detecção de colisões entre elementos, optimização do movimento do órgão terminal, e verificação da qualidade do processo de fabrico global.