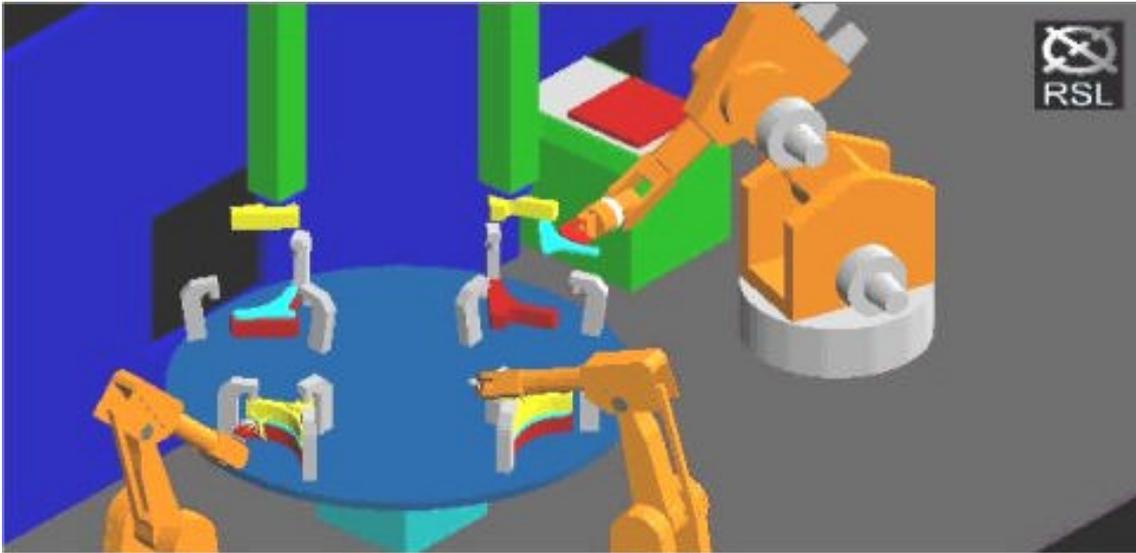


Simulação Robótica



WorkSpace

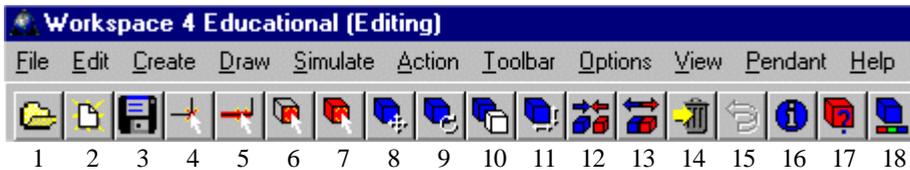
Laboratórios Integrados III

Departamento de Electrónica Industrial e de Computadores

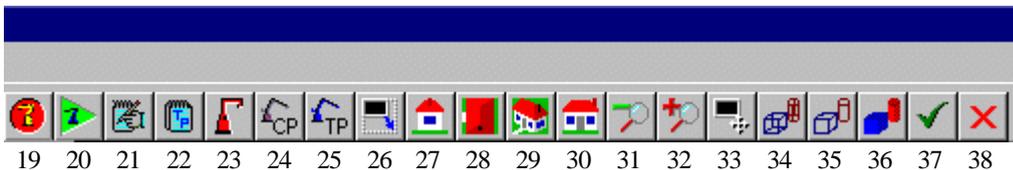
Universidade do Minho
Escola de Engenharia
Guimarães

Simulação de Robôs - Work Space 4

Função dos botões da toolbar



<p>Operação de ficheiros:</p> <ol style="list-style-type: none"> 1. Abrir Modelo 2. Novo Modelo 3. Gravar Modelo 	<p>Seleção de entidades:</p> <ol style="list-style-type: none"> 4. Um ponto 5. Aresta 6. Face 7. Objecto
<p>Edição de objectos:</p> <ol style="list-style-type: none"> 8. Mover objecto 9. Rodar 10. Duplicar 11. Editar dimensões 12. Juntar dois objectos 13. Separar dois objectos 14. Apagar um objecto 15. <i>Undo</i> 16. Informação de um objecto 17. Seleccionar um objecto pelo nome 18. Alterar a cor de um objecto 	



<p>Programar o Robô:</p> <ol style="list-style-type: none"> 19. Iniciar um programa novo 20. Executar um programa de robô 21. Editar um programa de robô 22. Editar Coordenadas no espaço 23. Mover Robô para <i>HOME</i> 24. Mover Robô para CP 25. Mover Robô para TP 	<p>Visualização do modelo:</p> <ol style="list-style-type: none"> 26. Escalar a imagem dentro do écran 27. Vista de Lado 28. Vista de Topo 29. Vista Ortogonal 30. Vista de Frente 31. Afastar 32. Aproximar 33. Mover a imagem dentro do écran 34. Vista <i>Wireframe</i> 35. Vista <i>HiddenLine</i> 36. Vista <i>Shade</i>
<p>Genéricos:</p> <ol style="list-style-type: none"> 37. Aceitar 38. Rejeitar 	

Simulação de Robôs - Work Space 4

Construção do modelo de um robô

Work Space (WS) para além de ser um simulador, tem ferramentas para desenhar, criar e calibrar robôs. Este exercício concentra-se na construção de um robô muito simples, apenas para exemplificar o funcionamento deste programa. A parte teórica da operação e controlo de robôs envolve estudos matemáticos bastante complexos, mas esses serão omitidos neste exercício. Apesar disso, será feita uma breve introdução à teoria dos robôs. Esta é necessária para entender todas as opções deste programa.

Ao nível mais simples, a estrutura mecânica dum robô pode ser considerada como uma serie de ligações (braços ou elos) normalmente metálicas (ou outros materiais) com articulações (ou juntas) que as ligam, tal como demonstra a figura 1. O exemplo simples da figura 1 consiste em três braços numerados de 0 a 2. A braço 0 é normalmente chamado base. Este sistema completo é conhecido por estrutura articulada.

Cada junta tem os seus próprios eixos de coordenadas (x, y, z). Exemplo para o braço 0 tem um sistema de coordenadas (x_0, y_0, z_0). Estes eixos são usados para definir a posição de qualquer ponto num espaço tridimensional em relação à junta 0.

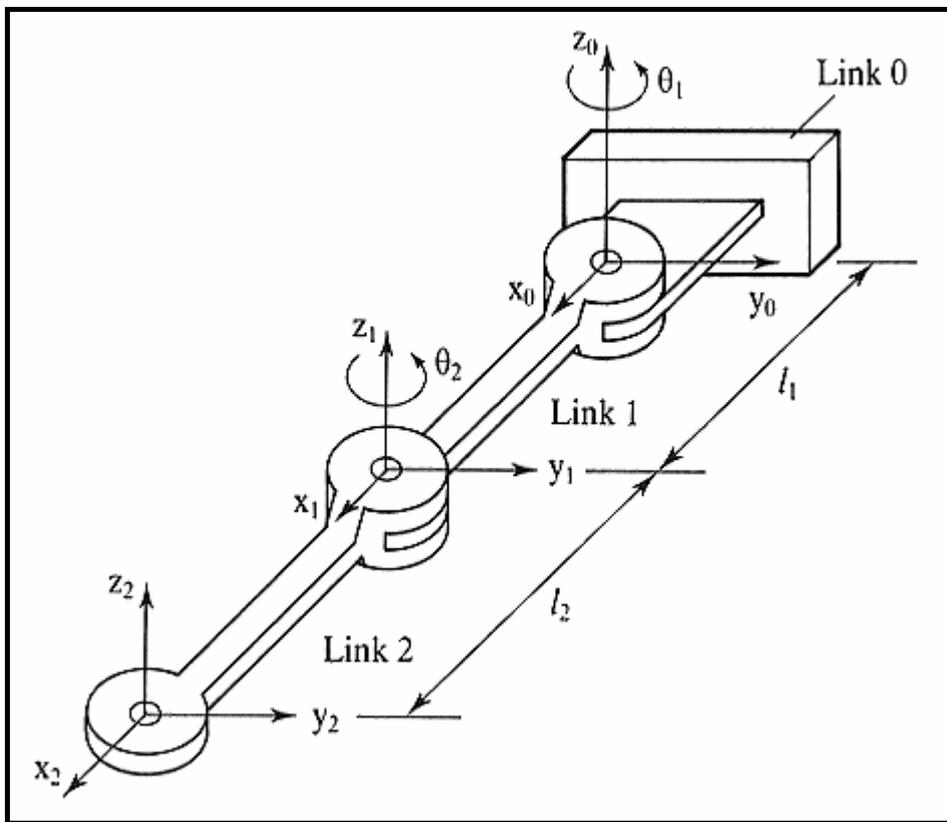


Figura 1 - Exemplo de uma estrutura de braço simples (com 2 Graus de Liberdade)

A figura 2 mostra uma perspectiva diferente de um robô simples com dois braços ligados a uma base (fixa). Também na figura encontra-se o diagrama do volume de trabalho e ainda uma representação dos sistemas de coordenadas.

Um certo numero de variáveis estão associadas a cada braço da estrutura. Para robôs mais complexos quatro variáveis são necessárias (por braço), mas neste exemplo da figura 2, apenas duas variáveis são relevantes. Estas são, o ângulo de rotação da junta θ (teta) e o comprimento do braço (l). A posição final do robô (extremo) é definida por estas duas variáveis. Isto é claramente visível na figura 2 onde o vector do ponto final do robô ${}^0P_{02}$, definido em termos de (x_0, y_0), é dependente dos valores de l_1 e l_2 em combinação com os ângulos θ_1 e θ_2 (${}^0P_{02}$ é definido como sendo um vector da origem de coordenadas 0 até à origem de coordenadas 2).

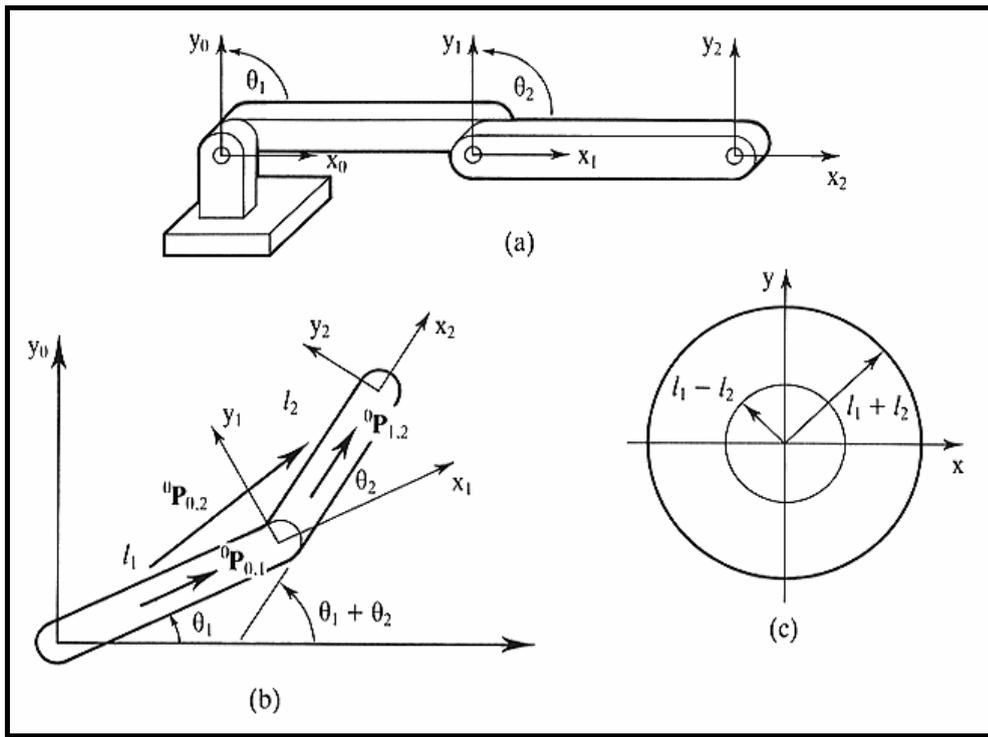


Figura 2 - Detalhes de uma estrutura de braço de robô simples

O volume de trabalho desta estrutura pode ser visto em c). Ambas as juntas rodam sobre o eixos dos Z, o qual produzem um volume de trabalho com a forma de um disco. Este volume de trabalho tem aplicações práticas muito limitadas. As juntas têm um grau de liberdade. Este tipo de junta é o mais comum nos robôs industriais.

Para se obter um robô com um volume de trabalho tridimensional mais útil, é necessário ter mais uma junta rotacional, que rode sobre o eixo dos X (ou Y). A Figura 3 mostra um dos mais populares robôs industriais (Unimation PUMA 550). Neste robô, as três maiores juntas são rotacionais (cinta, ombro, e cotovelo) e dão a posição, as outras três juntas são também rotacionais e dão a orientação do pulso.

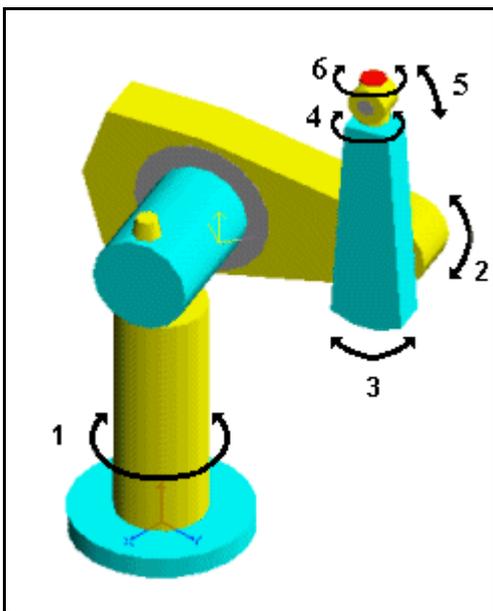


Figura 3 - Robô Unimation Puma 550

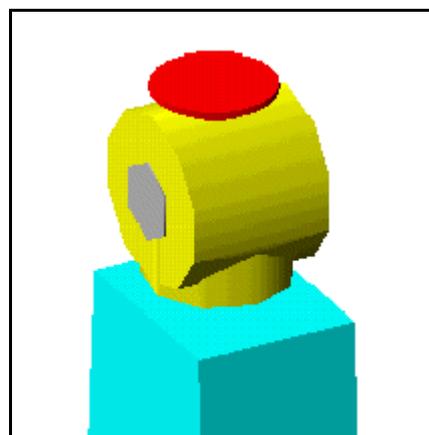


Figura 4 – Pormenor da garra do Puma 550

Para executar o software de simulação (WorkSpace 4) no PC, deve clicar-se no ícon **WorkSpace.exe** no directório **Programas\WorkSpace 4**. No início, este programa pergunta se o utilizador quer ver uma introdução inicial do programa, ao que é aconselhável responder que **sim** da primeira vez que se corre o programa para se obter as primeiras impressões do que o programa é capaz de fazer.

Muitos robôs consistem na combinação de juntas rotacionais e translacionais. As juntas translacionais originam um movimento linear.

A construção de um robô implica três passos principais:

1. Criar a 'forma geométrica' do robô
2. Localizar e orientar os sistemas de eixos de cada junta
3. Testar o modelo final

Para começar este exercício, é necessário obter o ecrã inicial limpo. Para isso selecciona-se a opção **File / New Model**. Para já trataremos do primeiro passo (Criar a forma geométrica do robô). Para facilitar a construção deste modelo pode e deve usar os botões da toolbar (26 ao 33) os quais permitem várias vistas diferentes (Frente, Lado, Topo, Ortogonal, aproximação, afastamento, centrar modelo no ecrã e escalamento automático no ecrã).

1. Seleccionar a opção **Create / Create Cylinder**. Digite **MEUROBO** no nome do objecto. Escolha as dimensões: **Radius: 200, Length: 50 e Ratio: 1**. Digite ENTER.
2. Posicione o cursor nas coordenadas (0, 0, 50) para o localizar na superfície superior do cilindro. Para isso digite a tecla 'I' (depois de pressionar o rato em qualquer posição do ecrã, de modo a ver o cursor em forma de eixos coordenados) e digite depois as três coordenadas.
3. Seleccionar a opção **Create / Create Cylinder**. Digite **LINK1** no nome do objecto. Escolha as dimensões: **Radius: 75, Length: 1000 e Ratio: 1**. Digite ENTER.
4. Posicione o cursor nas coordenadas (0, 130, 550) para o localizar o cursor a meio do tronco do robô.
5. Seleccionar a opção **Create / Create Box**. Digite **LINK2** no nome do objecto. Escolha as dimensões: **Length 1: 600, Length 2: 100 e Length 3: 100**. Digite ENTER.
6. Posicione o cursor nas coordenadas (0, 130, 555).
7. Seleccionar a opção **Create / Create Box**. Digite **LINK3** no nome do objecto. Escolha as dimensões: **Length 1: 600, Length 2: 90 e Length 3: 90**. Digite ENTER.
8. De acordo com o gosto de cada um, diferentes cores podem ser atribuídas a cada braço do robô, bastando para isso usar o botão 18 ou a opção **Edit / Object / Colour / Colour of Object**, seleccionar o objecto (na opção **NAME**), seleccionar **CONTINUE** e escolher depois a cor pretendida num menu de cores que aparecerá no ecrã. Se o objecto pretendido estiver já seleccionado (fazendo duplo clique no objecto desejado) selecciona-se na mesma a opção **Edit / Object / Colour / Colour of Object** e escolhe-se a cor desejada que aparecerá numa paleta. Cores dão um maior realismo à simulação.
9. Para já, o que foi desenhado consiste apenas num conjunto de objectos sem 'relação' nenhuma entre eles. Para finalmente 'criar' o robô selecciona-se o objecto da base **MEUROBO** e depois selecciona-se a opção **Create / Create Robot**. O programa pergunta quantos graus de liberdade vai ter o robô (embora um valor por defeito seja automaticamente assumido (neste caso 3)) e consiste no número de objectos com o nome **LINKx** que se criou, sendo assim **LINKn** uma palavra reservada que indica que o objecto vai ter um movimento ou grau de liberdade). Para este exemplo, assume-se o valor por defeito (3 graus de liberdade). A partir daqui, o robô está operacional e deve ter o aspecto da figura 5.

Os eixos coordenados no centro do ecrã (à direita do robô) representam o cursor (current pointer). As três linhas horizontais no canto inferior direito do ecrã representam as três juntas deste robô. A pequena linha vertical no centro de cada uma das linhas horizontais, representam a posição actual da junta em relação aos limites máximo e mínimo desta.

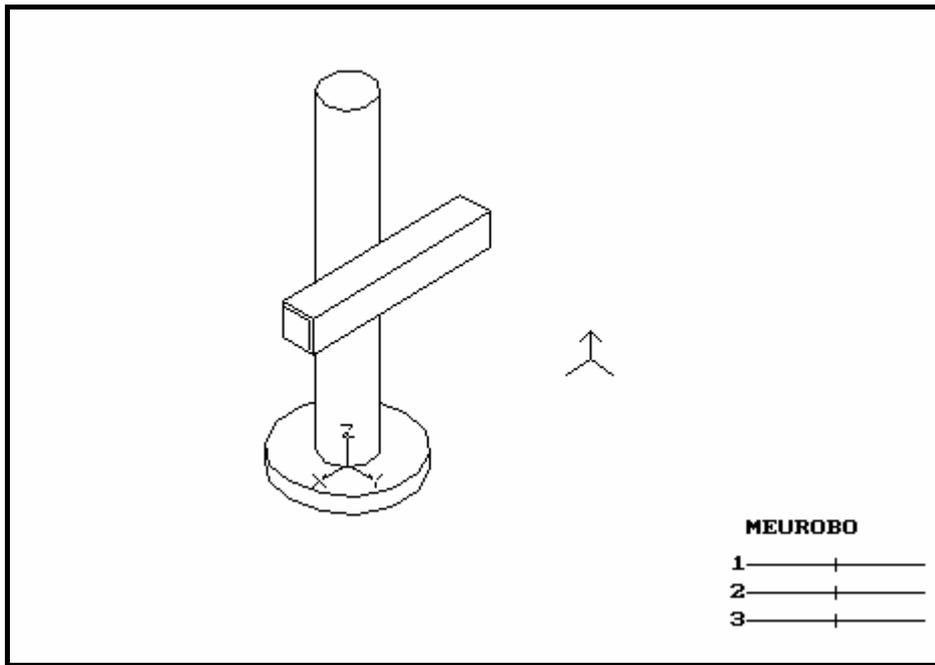


Figura 5 - Aspecto do Robô MEUROBO

Existem três modos de visualização do modelo, e estes podem ser seleccionados nas três primeiras opções do menu **View**. São eles **Shade**, **Hidden Line** e **Wireframe**. Descubra quais as diferenças entre eles seleccionando-as uma a uma. Para reduzir o tempo de construção deste modelo pode usar o modo **Wireframe**. As figuras 3 e 4 usam o modo **shade**, e a figura 5 e 6 usam o modo **Hidden Line**.

Considere-se agora o segundo passo, que consiste em localizar e orientar os sistemas de eixos de cada junta. Se testarmos o robô tal como está, verificaremos que as juntas não funcionam como nós pretendemos pois todas são criadas por defeito como juntas rotacionais. Para as testar basta ir ao menu **Pendant** e aumentar e/ou diminuir os valores para cada junta (**1**, **2** e **3**). O resultado não é o esperado. É necessário pois modificar os eixos de cada junta. A primeira junta deverá ser rotacional, a segunda translacional (mover-se-á para cima e para baixo) e a terceira translacional (mover-se-á para fora e para dentro). A Figura 6 mostra o movimento pretendido deste robô.

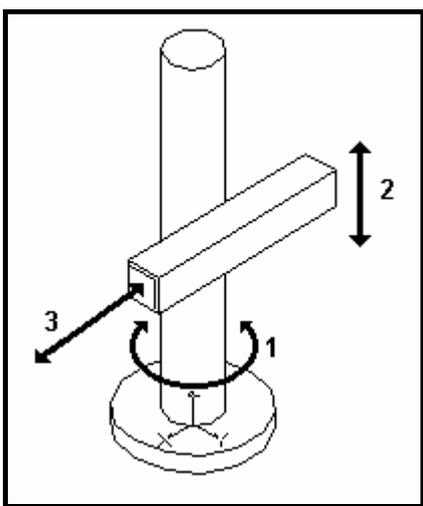


Figura 6 - Movimentação do Robô MEUROBO

Para alterar a orientação/ posição de uma determinada junta, deve seleccionar-se o objecto que a suporta (anterior), assim como um humano para mover a junta “cotovelo” tem que fazer força nos músculos do “braço” e não nos músculos do antebraço. Para editar a junta 2 deve seleccionar-se o **LINK1**.

Antes de executar qualquer alteração em qualquer uma das juntas, ponha todas as juntas do robô na posição zero (usando o Pendant).

Para a correcta movimentação de todos os eixos devem seguir-se os seguintes passos:

1. Seleccionar o objecto **MEUROBO** e usar a opção do menu **Edit / Joint / Type of Joint**. Escolher depois a opção **Rotational**.
2. Fazer o mesmo para o objecto **LINK1** e **LINK 2** mas escolher o tipo **Translational**.
3. Se fôr agora testado este robô no menu **Pendant** verificar-se-á que a terceira junta ainda não está correcta pois em vez do braço se mover de dentro para fora, move-se para cima e para baixo. Isto tem a ver com a orientação dos eixos coordenadas. A direcção do Z indica a direcção do movimento linear. Para se proceder a esta correcção utiliza-se a opção **Edit / Joint / Turn Axes** e orienta-se o Z na direcção do movimento do terceiro braço ou seja para fora (a afastar-se do tronco do robô, ver figura 6). (Yaw = 90).
4. Se fôr agora testado, o robô irá comportar-se correctamente no que diz respeito às orientações e direcções de cada braço.

O próximo passo será impôr limites nas articulações, pois se exagerarmos os movimentos das juntas acontecem 'imprevistos' tais como o braço **LINK3** 'sair' do braço **LINK2**. Estes limites são definidos na opção do menu **Edit / Robot / Limits**. As células tem o seguinte significado:

- Min1 - Valor Mínimo da junta 1 (em graus ou mm conforme o tipo de junta)
- Max1 - Valor Máximo da junta 1 (em graus ou mm conforme o tipo de junta)
- Mxv1 - Valor da Velocidade Máxima da junta 1 (em mm/segundo)
- Mxa1 - Valor da Aceleração Máxima da junta 1 (em mm/segundo quadrado)

O numero após a palavra designa o numero da junta. Modifique estes valores de forma que o robô se comporte correctamente. Como sugestão pode usar os valores:

Min1	-150	Max1	150
Min2	-400	Max2	400
Min3	0	Max3	500

Grave o modelo através da opção do menu **File / Save Model as** e dê-lhe o nome **MEUROBO**.

Execute a opção do menu **Edit / Robot / Robot Settings** e active a janela **Show Kinematics**. As dimensões do modelo cinemático ficam agora visíveis no modelo.

O terceiro passo será agora levado a cabo e consiste no teste exaustivo deste modelo. Verifique todas as posições e o seu correcto funcionamento. Se surgirem problemas com o modelo tente resolvê-los através das instruções previamente descritas.

IMPORTANTE:

Como tarefa seguinte e para praticar o que foi descrito neste exercício, construa o modelo do robô descrito na figura 3 (PUMA 550) e quando terminar este, construa a célula do robô (ex.: paredes com janelas, gradeamentos de protecção, mesas de trabalho, alimentadores de peças, etc.).

Simulação de Robôs - Work Space 4

Movimentação de um Robô

A habilidade de mover um robô é fundamental para a operação do simulador WorkSpace (WS). Antes de continuar é necessário resumir a teoria da estrutura do robô. O robô usado neste exemplo é o IRb2000 e tem 6 graus de liberdade. As três primeiras juntas são o **Waist** ou cinta (No. 1), o **Shoulder** ou ombro (No. 2) e o **Elbow** ou cotovelo (No. 3). Estas juntas têm um movimento idêntico ao dos membros com o mesmo nome do corpo humano. Estas três juntas possibilitam o robô a posicionar-se em qualquer ponto tri-dimensional do seu volume de trabalho. As três juntas mais pequenas são chamadas **ROLL** (No. 4), **PITCH** (No. 5) e **YAW** (No. 6). Estes nomes derivam dos termos náuticos do movimento de um navio em diferentes condições de mar e vento. Estas três juntas estão associadas com o **Wrist** (pulso) do robô e controlam a orientação do ponto central deste. Todas as juntas são rotacionais. Cada junta tem o seu próprio eixo de rotação. Deste modo, as três primeiras juntas servem para posicionar a ferramenta do robô na posição correcta e as três seguintes asseguram a orientação da ferramenta na direcção correcta.

Há três métodos para mover o modelo do robô no WS:

1. 'Mostrar' ao robô a nova posição (através das teclas do cursor ou premindo o rato no local desejado) e este mover-se-á para lá (é importante salientar que nem todas as posições no ecrã têm solução).
2. Incrementar/Decrementar as posições das juntas individualmente (no *Pendant*) até que a nova posição seja alcançada.
3. Especificar a posição e orientação do ponto final do robô (coordenadas X, Y, Z e orientação A, B, C).

Cada um destes métodos será considerado individualmente.

Faça o Load do modelo **IRB2000.MOD**. Para isso use a opção do menu **File / Load Model**.

Para que o programa funcione de um modo mais rápido deve o modo de visão **Wireframe** (no menu **View**).

Primeiro método:

Posicione o cursor (eixos coordenados - **current pointer**) numa posição do ecrã qualquer e pressione o botão da esquerda. As novas coordenadas aparecem na **statusbar** (no fundo do ecrã). O cursor passa a ter uns eixos coordenados e esta posição actual do cursor é chamada **CP** (Current Pointer). Onde quer que o botão do rato seja pressionado, esta posição passa a ser a **CP**. O primeiro método de mover o robô é simplesmente comandá-lo para este se mover para a posição e orientação dados por CP. Para isso usa-se a opção do menu **Pendant / Followcp**.

Depois de seleccionar esta opção, o cursor posiciona-se no ponto final do robô, e a partir deste momento, sempre que se premir o botão do rato (ou mover o **CP** com as teclas do cursor) o robô move-se para o **CP** (excepto se a posição for impossível de alcançar, como exemplo é o caso do CP estar fora do volume de trabalho do robô). Neste caso aparecerá uma mensagem de erro a dizer que a posição é impossível. Mesmo que a posição pareça possível, esta pode não ser pois a terceira coordenada (profundidade no ecrã) não é perceptível aos nossos olhos. Para isso usa-se os ícones da toolbar para termos diferentes perspectivas (Frente, Lado, Topo, Diagonal, etc.). Explore os limites do volume de trabalho deste robô.

É de reparar que a face final do robô alinha-se sempre com a orientação dos eixos do **CP**. Uma posição pode estar dentro do volume de trabalho mas a orientação não ser possível de por parte do robô.

Um outro processo ainda no primeiro método consiste no seguinte:

Mova o robô de novo para a sua posição inicial. Para isso, use a opção do menu **Action / Robot move commands / Move Home**. Posicione o **CP** numa posição qualquer dentro do volume de trabalho do robô. Selecciona a opção **Action / Robot move commands / CP Move**. Se por acaso a mensagem de erro aparecer, mova o **CP** para outra posição possível (dentro do volume de trabalho do robô) e volte a tentar.

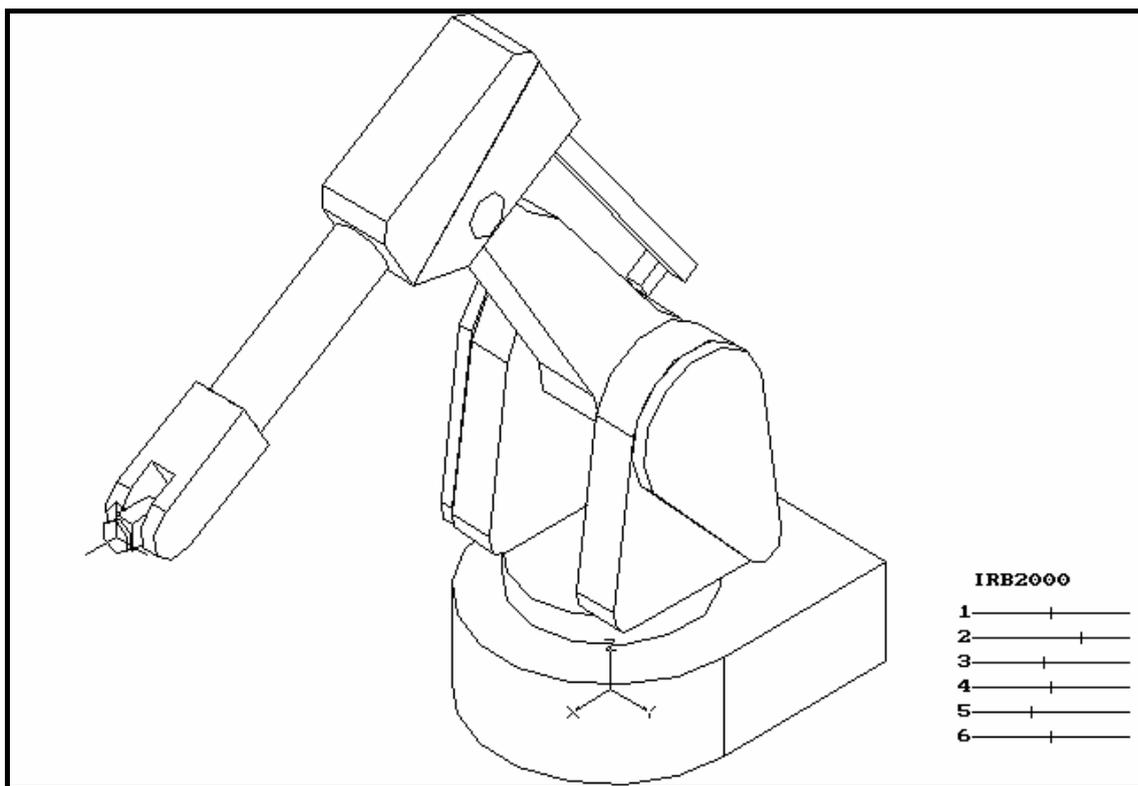


Figura 1 - Robô IRb2000 da ABB a posicionar-se no CP e onde a orientação é correspondente

Depois de fazer várias tentativas pelo mesmo método de posicionamento mova o robô de novo para a sua posição inicial. Para isso, use a opção do menu **Action / Robot move commands / Move Home** (ou o botão 23).

Os dois processos já demonstrados que representavam o primeiro método de posicionamento usam o **CP** para mostrar ao robô para onde se deve mover. O **CP Move** mostra o percurso que o robô realiza desde a sua posição actual até ao CP enquanto que o **Followcp** posiciona o robô no local desejado sem mostrar o seu percurso.

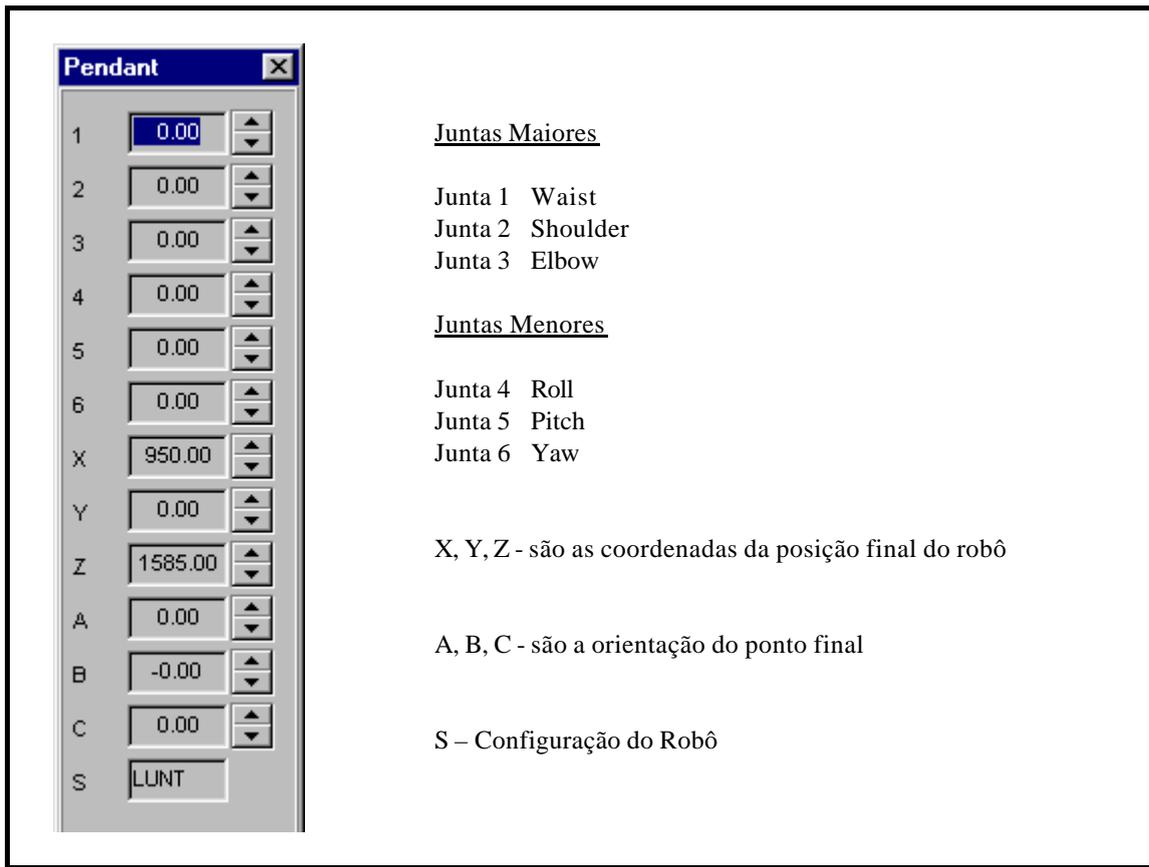


Figura 2 - Interpretação do menu **Pendant**

Segundo método:

Este consiste em alterar a posição individual de cada junta. Seleccione o menu **Pendant**. Este mostra (tal como na figura 2) o valor numérico da posição de cada junta, bem como a posição final do robô (X, Y, Z) e ainda a orientação (A, B, C).

Os valores mostrados na figura 2 representam o robô na sua posição inicial (**Home Position**), pois todas as juntas estão a zero. Os valores de X, Y, Z dão-nos uma perspectiva do tamanho do robô (aproximadamente 1.5 metros de altura).

Seleccione o menu **Pendant**. Há duas maneiras de mudar o valor de uma junta. Ou se pressiona com o botão do rato nas setas do lado direito de cada junta (cima para aumentar, baixo para diminuir) ou se pressiona com o botão do rato no próprio valor e se introduz o valor numérico exacto usando o teclado. Experimente mudar o valor de ambas as maneiras.

Para mudar o valor das três últimas juntas será mais conveniente fazer **zoom** do pulso do robô tal como demonstra a figura 3. Deste modo visualizará muito mais facilmente o seu comportamento.

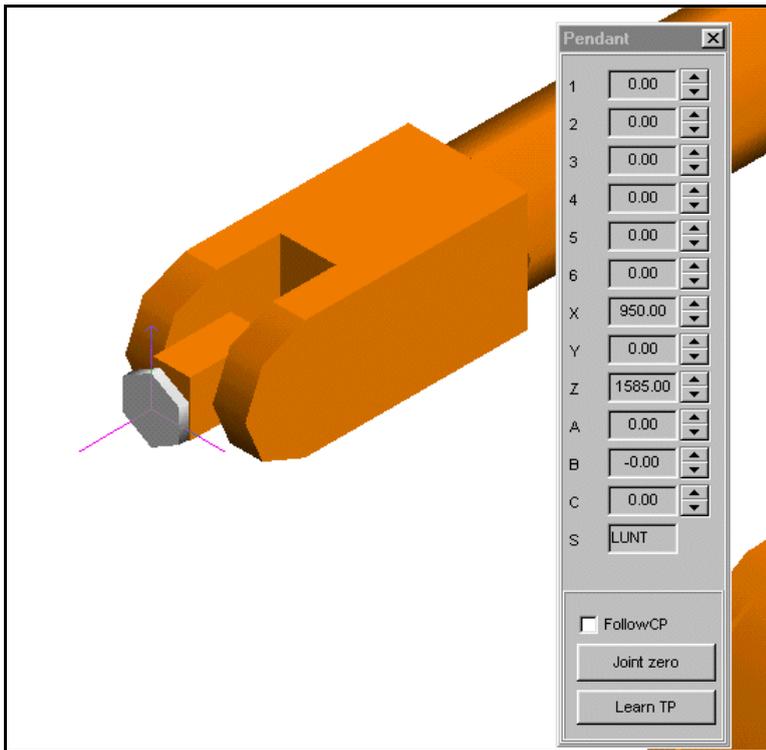


Figura 3 - Detalhe do pulso do IRb2000

Terceiro método:

Este consiste em alterar as coordenadas do ponto final do robô (X, Y, Z) através do teclado e ainda a orientação do mesmo modo (A, B, C). Seleccione o menu **Pendant**. Experimente mudar o valor das coordenadas (X, Y, Z) e da orientação (A, B, C) pressionando com o botão do rato nas setas do lado direito de cada variável (cima para aumentar, baixo para diminuir) ou pressionando o botão do rato no próprio valor e introduzindo um valor numérico exacto.

Como continuação deste exercício, estude e verifique o menu **Action**. Tente familiarizar-se com todas as instruções que lá existem, especialmente as do menu **Robot Move Commands**, **Gripper Commands**, **Miscellaneous**, **Object Commands**. Entre outras, existem instruções para pegar ou largar objectos. Tente usá-las depois de criar um objecto fictício (ex.: caixa).

Simulação de Robôs - Work Space 4

Criação de uma animação

Animação é o aspecto mais importante e o principal objectivo da utilização deste programa. Mais do que qualquer outra coisa, ela define a utilidade máxima deste programa e consiste na habilidade de simular o mundo real na forma de uma animação gráfica em computador. Outras vantagens, apesar de também serem importantes, são simplesmente a possibilidade e facilidade de criar e implementar estas animações. Nesta aplicação pode simular-se para além de robôs, tapetes rolantes, e outras estruturas ou objectos relacionados com a célula robótica.

Há três métodos de criar uma animação no WS:

1. Animação de objectos que não sejam robôs.
2. Animação de robôs usando o **CP Move**.
3. Animação de robôs usando **Teach Points**.

Para começar, e como introdução a este exercício, será examinado um exemplo previamente definido.

No menu **Simulate / Simulate Options / AVI File / *.AVI** seleccione o ficheiro **CTRLARM.AVI**. Seleccione depois a opção do menu **Simulate / Media Player**.

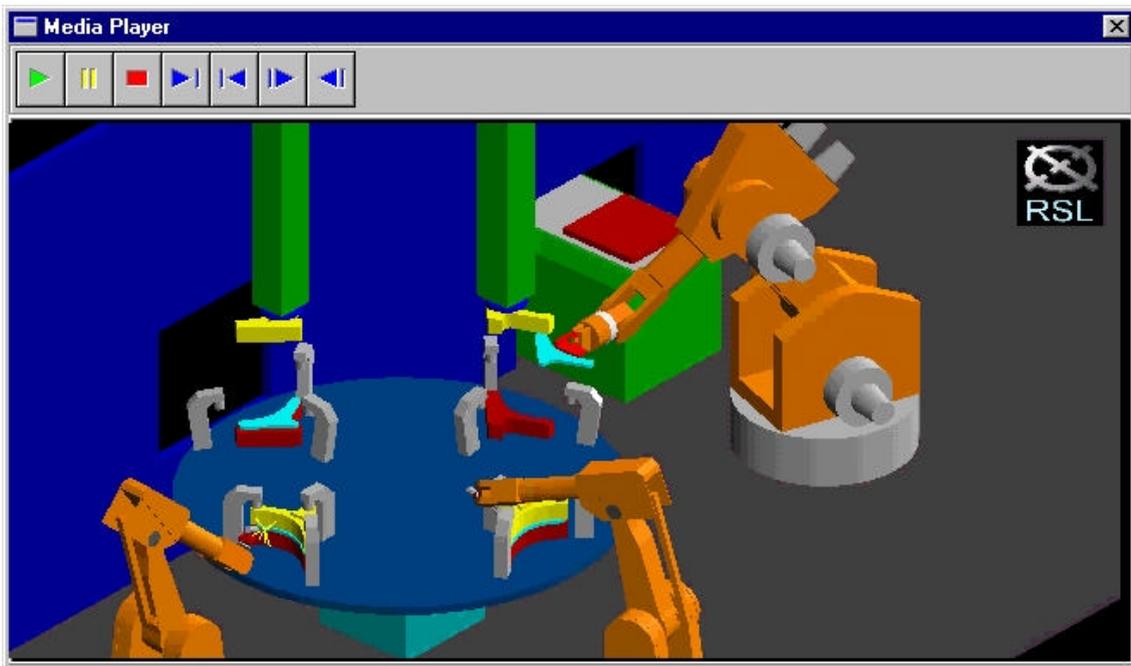


Figura 1 - Animação da célula CTRLARM

Esta célula mostra uma mesa rotativa, com dois robôs a soldar peças e um outro a colocar as peças num tapete rolante para saírem da célula. Verifique outros exemplos existentes (*.AVI).

A construção de exemplos como este requer a criação de um mundo virtual, isto é, tapetes rolantes, grades de segurança, mesas, etc. Neste exercício não se vai demonstrar como se constroem estes objectos mais complexos, mas sim como se animam.

Selecione no menu a opção **File / New Model**. Seleccione depois a opção **Create / Create Box** e dê-lhe o nome de **BOX1** e o tamanho (200 200 200). Seleccione a vista ortogonal. O resultado deve ser parecido com a figura 2.

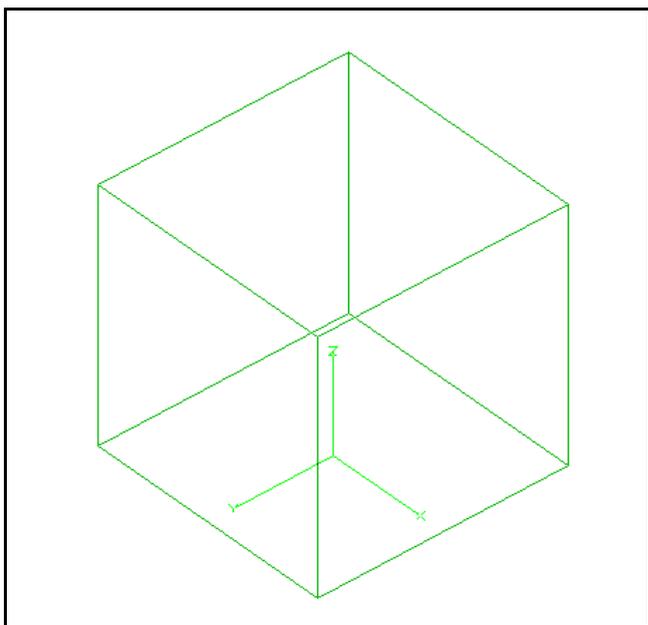


Figura 2 - Criação de um cubo para animação

Uma vista ortogonal Wireframe do objecto **BOX1** preenche o ecrã. Os eixos coordenados principais estão posicionados no centro da base do cubo e são os eixos à volta dos quais **BOX1** será rodado.

Selecione no menu a opção **Simulate / Start Track**. Responda depois ao nome do programa de robô 'TESTE.KL' (a extensão KL varia conforme a linguagem do robô seleccionada). Depois será perguntado se queremos gravar o modelo (caso este não esteja já gravado). Não confundir *Modelo* com *Track*.

Aparecerá uma janela de menus **ACTION** com os comandos possíveis.

BOX1 deverá ser o objecto seleccionado mas pelo sim pelo não, use o rato para seleccionar esse objecto (clitando sobre ele duas vezes).

Selecione a opção **Begin** (da tal janela ACTION que apareceu) para começar o programa propriamente dito.

Selecione depois a opção **Object commands / Turnobj**, pressione o botão do rato no campo **Roll** e escreva 180 no teclado seguido de ENTER. Depois aceite o valor de 20 para velocidade angular. O cubo rodará agora 180 graus sobre o eixo do Z. Selecione a opção **End** para indicar que acabou o programa. Uma janela aparecerá a dizer **Track Recording Halted**. Está criado um programa (com apenas uma instrução de rotação de um objecto). Para visualizar novamente a animação selecione a opção **Simulate / Replay Track**. Para gerar um ficheiro de animação (extensão **.AVI**) selecione a opção **Simulate / Replay Track and Creat AVI**. Responda depois ao nome do ficheiro filme (**.AVI**).

Este ficheiro **AVI** Pode ser visto por qualquer Media Player que interprete este formato. Para obter uma animação mais real com shade, selecione a opção **View / Shade**, e gere outra vez um novo ficheiro **AVI**. Repare que não foi necessário criar outra vez o programa com o comando **Simulate / Start Track**.

Como resumo, as opções seleccionadas foram:

1. **Create Object**
2. **Simulate / Start Track**
3. **Action / Begin**
4. **Action / Turnobj**
5. **Action / End**
6. **Simulate / Replay track** ou **Simulate / Replay track and Creat AVI**
7. **Simulate / Media Player**

Durante este processo, perguntas tiveram de ser respondidas acerca de ficheiros de diferentes formatos (diferentes extensões). Assim, os três formatos podem desde já ser definidos: **.MOD** é o ficheiro onde fica

guardado o modelo geométrico e cinemático do robô, célula e outros objectos; **.KL** é o ficheiro onde fica guardado o programa do robô gerado automaticamente (esta extensão varia com a linguagem de robô usada); **.AVI** é o ficheiro onde fica guardada a sequência de imagens que dá origem ao 'filme' ou animação.

O ficheiro **.KL** pode ser também editado através de um editor normal de texto ou editado directamente através da opção **Simulate / Edit Track** (ou ainda através do botão 21 da toolbar). Como exemplo, seleccione essa opção para ver as instruções que foram automaticamente geradas. O resultado está na figura 3.

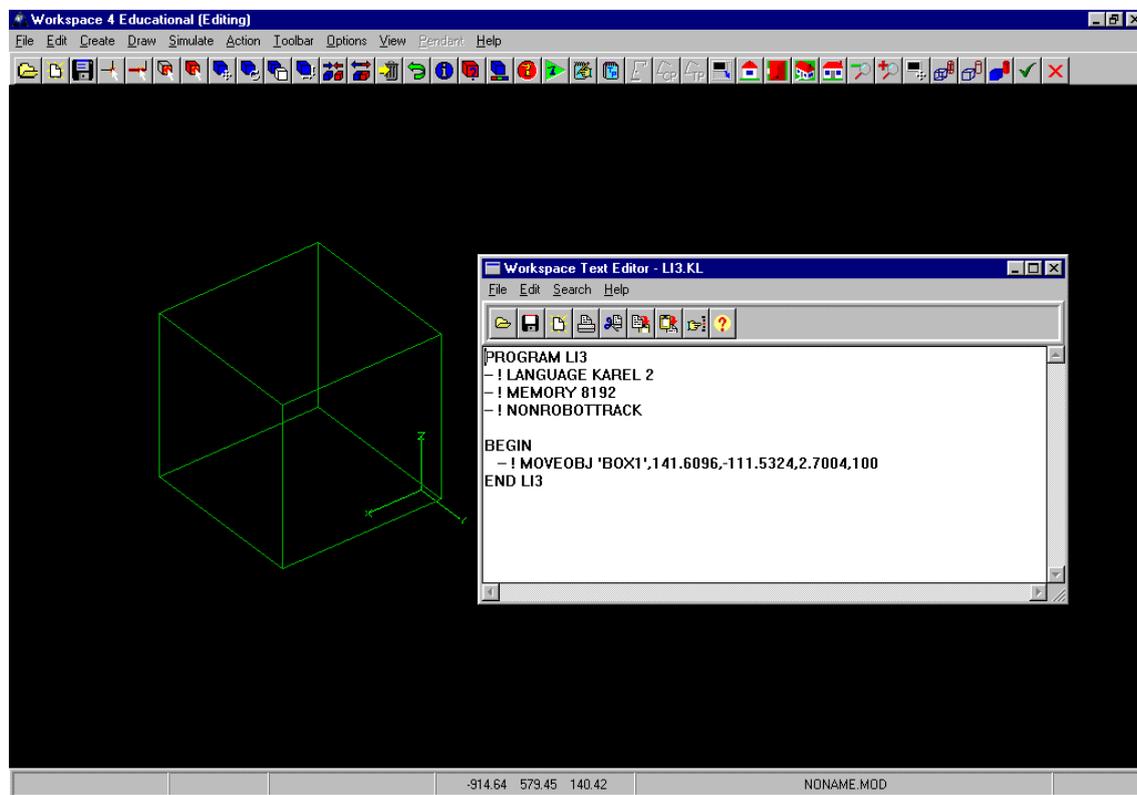


Figura 3 - Instruções de um programa de robô gerado automaticamente (ficheiro com extensão .KL)

O programa para mover um cubo, consiste apenas numa linha com uma instrução (**Moveobj**), seguida do nome do objecto (**Box1**), um vector com três componentes que definem o vector translação, e finalmente o valor da velocidade de translação. Como exercício, experimente modificar (editar) alguns destes valores para ver o que acontece.

O segundo método a ser considerado para criar uma animação é a animação de um modelo de robô usando o **CP Move**. Neste exemplo vai ser criado um objecto para ser movido pelo robô em vez de se mover apenas o braço do robô. Para isso, começaremos por seleccionar um modelo já existente. **File / Load model / *.MOD / IRB2000.MOD**.

Crie uma caixa através do menu **Create / Create Box** e aceite o nome por defeito de **BOX1** e com as dimensões (200 200 200). Uma caixa aparecerá onde o cursor se encontrava. Visualize o modelo e a caixa, nas diversas perspectivas possíveis (frente, lado, topo, ortogonal) para ter a noção exacta onde a caixa se encontra. Agora, posicione a caixa no 'chão' em frente ao robô. É importante realçar que não existe gravidade no ambiente desta aplicação. Se um objecto é criado no 'ar' ele fica lá até que seja movido para outro sítio qualquer. Para se colocar a caixa no chão basta para isso seleccionar a caixa, e pressionar o 8o. botão da toolbar (mover objectos) pega-se no objecto e move-se para baixo. Se usarmos a vista de lado, facilita-nos a sua deslocação para o chão. Use todas as vistas necessárias para posicionar esta caixa em frente ao robô. Seguidamente grave este modelo (que inclui a caixa) com um novo nome, por exemplo **OUTRO.MOD**. NÃO GRAVE por cima do modelo **IRB2000.MOD**.

Seleccione agora a opção do menu **Simulate / Start Track**. Escolha o nome desejado para o ficheiro **.KL**. É conveniente ver o objecto e robô pelo menos através de duas perspectivas. Depois faça **Action / Begin**. O CP deve agora ser posicionado no centro da face superior da caixa para facilitar a tarefa do robô de pegar na caixa. Use todas as vistas necessárias para posicionar o CP. Quando o CP estiver na posição e orientação correctas seleccione a opção **Action / CP Move** para mover o robô para o CP. O robô move-se da sua posição original para o CP numa sequência de imagens. A orientação do pulso do robô em relação à caixa é feita de acordo com a orientação dos eixos coordenados do CP. Se por acaso, o braço do robô não estiver na orientação correcta, terá de reorientar-se os eixos coordenados do CP através do menu **Edit / CP / Turn Cp**.

Seleccione agora a opção **Action / Gripper Commands / Close Hand** e responda sim à pergunta 'OK to grasp BOX1' (se o programa não fizer esta pergunta é porque o objecto não está seleccionado). A caixa está agora 'colada' à ferramenta do robô e qualquer que seja o movimento que se efectue com o robô, a caixa vai junto. Isto pode parecer anormal pois o robô não tem nenhuma ferramenta para pegar em objectos, mas isto é perfeitamente possível. Cada aplicação pode ter uma ferramenta diferente e por isso cada pessoa pode desenhar a sua (não havendo nenhuma standard). Deste modo o programa tem que aceitar qualquer tipo de ferramenta (mesmo quando esta é 'nenhuma'). Outro detalhe importante é que este simulador não se importa com o peso do objecto. Se fizéssemos este robô pegar num objecto maior que ele, não havia limitações e ele pegava mesmo nele enquanto que na realidade isso impossível.

A seguir move-se o CP para cima (cerca da altura do ombro do robô) e executa-se outra vez a opção **Action / Robot Move Commands / CP Move**. Depois selecciona-se a opção **Action / Gripper Commands / Open Hand** para largar outra vez a caixa. A seguir move-se o robô para a sua posição inicial (**Home Position**) e acaba-se o programa com a opção **Action / End**.

Para gerar o ficheiro **.AVI** desta animação selecciona-se a opção **Simulate / Replay Track and Creat AVI**. Para visualizar a animação final selecciona-se a opção **Simulate / Media Player**.

Agora se o ficheiro **.KL** for editado, podem ver-se as instruções que o compõem. Se o número de frames que compõem a animação são poucas (quando se verifica uma descontinuidade no filme) este pode ser melhorado através da opção **Simulate / Simulate Options / Frame Interval**. Este valor representa o tempo entre imagens. Quanto mais imagens mais lenta se torna a animação (embora não seja um valor muito significativo) mas mais real esta se torna.

O terceiro método de criar uma animação no WS é usando **Teach Points**. O robô (e não o CP) é primeiro movido para as posições tri-dimensionais desejadas e estas são gravadas (uma a uma) num ficheiro com o nome **NOME#.KL** (NOME é variável, mas o simbolo # e a extensão são obrigatórios). Este nome é definido no menu **Simulate / Simulate Options / TeachPoint file**. Quando todas as posições estiveram gravadas, poder-se-á então começar a criar o programa do robô movendo-o para os pontos já definidos. Estes **Teach Points** são gravados com a opção **Pendant / Learntp** após se ter movido o robô para a posição desejada.

Para este exemplo vamos usar o mesmo modelo (IRb2000) usado para demonstrar o método anterior. Move-se então o robô para a posição desejada (pode ser o centro da face superior da caixa), e selecciona-se a opção **Pendant / Learntp**. É-nos perguntado o nome do ficheiro onde as coordenadas (TPs) vão ser guardadas (caso este ainda não tenha sido definido no menu **Simulate / Simulate Options / TeachPoint file**) e depois o nome da coordenada (por defeito o sistema usa **TP1, TP2, etc.**). Depois posiciona-se o robô noutra posição tri-dimensional e grava-se outro ponto da mesma maneira que o ponto anterior. No espaço tri-dimensional onde cada ponto é gravado aparece um sistema de eixos coordenados para cada ponto com o nome ao lado. Deste modo sabe-se sempre onde fica cada ponto.

Quando todos os pontos necessários estiverem guardados, pode começar-se a criar o programa do robô tal como no método anterior, seguindo os passos resumidos a seguir:

1. **Simulate / Start Track**
2. **Action / Begin**
3. **Action / Robot move commands / TP Move** (escolha o ponto TP1)
4. **Action / Gripper commands / Close Hand** (antes de executar esta instrução seleccione o objecto que vai pegar. Depois de executar este comando responda 'Yes' à pergunta 'OK to grasp Box1?')
5. **Action / Robot move commands / TP Move** (escolha o ponto TP2)
6. **Action / Gripper commands / Open Hand**

7. **Action / Robot move commands / TP Move** (escolha o ponto TP3) ou **Move Home**
8. **Action / End**
9. **Simulate / Replay Track** (ou **Replay Track and Creat AVI**)

A simulação pode agora ser vista. Se por acaso alguma coordenada estiver incorrecta, esta pode ser modificada através do re-posicionamento do robô (e nova gravação da coordenada com o mesmo nome que a incorrecta (TP?)) ou pode ainda editar-se o ficheiro de coordenadas tal como um ficheiro de texto através da opção **Simulate / Edit Teachpoints**.

Depois do programa estar acabado, e para criar uma simulação mais realista, pode usar-se *shading* através da opção **View / Shade** e voltar a criar o ficheiro de animação.

IMPORTANTE

Como exercício final, pede-se a criação de uma animação onde um robô (**PUMA260**) resolve o problema das torres de hanoi através do uso de **TP**'s. Pretende-se que seja criada a mesa, as peças e o programa de robô para simular toda a sequência. Um exemplo é dado pela figura 4. Verifique que o volume de trabalho do robô consegue englobar a zona dos cilindros de Hanoi, para garantir que o robô consiga chegar às peças, e ainda que a movimentação das peças não provoque colisões entre elas.

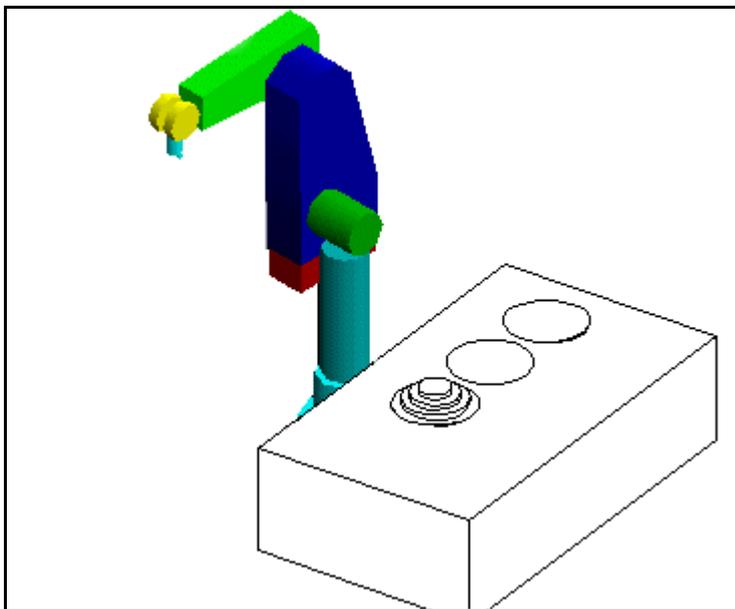


Figura 4 - Modelo do PUMA260 para resolução das Torres de Hanoi