

M



IMPLEMENTAÇÃO DE SOLUÇÃO CAM NA CÉLULA ROBÓTICA DO CINFU

LUÍS FILIPE TOMÉ DOMINGUES

DISSERTAÇÃO DE MESTRADO APRESENTADA À FACULDADE DE ENGENHARIA DA
UNIVERSIDADE DO PORTO EM ENGENHARIA METALÚRGICA E DE MATERIAIS

ORIENTADOR:

Professor Vitor Martins Augusto

Porto, 2015

“Have no fear of perfection, you’ll never reach it.”

Salvador Dali

Resumo

O objetivo da dissertação centrou-se na implementação da solução CAM *PowerMILL Robot Interface* na célula robótica do CINFU. Esta solução CAM permite simular e pós-processar percursos de maquinagem para serem executados por robôs, dando a estes funções similares aos centros de maquinagem CNC. A célula robótica do CINFU tem a particularidade de poder operar numa cinemática *part-on-robot*, ou seja, a peça fica montada no braço do robô, enquanto que a ferramenta se encontra em posição estática, o que se verificou ser um desafio devido a ser uma cinemática diferente especialmente quando se recorre ao *PowerMILL Robot Interface* para programar percursos de maquinagem. Após proceder à configuração da célula foi possível simular percursos de maquinagem e proceder à maquinagem de peças protótipo.

Abstract

The purpose of this dissertation was to implement the CAM solution PowerMILL Robot Interface in the CINFU's robotic cell. This CAM solution enables you to simulate and post-process machining paths to be run by robots, giving them similar functions to the CNC machining centers. The CINFU's robotic cell has the particularity of being able to operate in a part-on-robot kinematics, where the part is mounted on the robot's arm while the tool is static, which was considered challenging because it is a different robot kinematics especially when it relies on PowerMILL Robot Interface to program machining paths. Once the configuration of the cell was done machining paths were simulate and prototypes were created.

Agradecimentos

Ao professor Vitor Martins Augusto por ter aceitado orientar-me, pelo acompanhamento durante a realização desta dissertação de mestrado e por me proporcionar o contacto e a integração num ambiente empresarial.

Ao CINFU, que me acolheu nas suas instalações, pela forma como fui recebido por todos os seus colaboradores e pela disponibilização de todos os recursos necessários à realização deste projeto.

À NORCAM pela cedência da licença do software *PowerMILL Robot Interface* durante o período de realização da dissertação.

Aos engenheiros Paulo Aguiar Pinto e Filipe Cruz do CINFU e ao engenheiro Alexandre Bastos da DELCAM pelas suas disponibilidades e pelo acompanhamento durante o período de dissertação.

À minha família e à minha namorada.

Índice

Resumo	II
Abstract	III
Agradecimentos	IV
Abreviaturas e Siglas	X
Objetivo	1
As empresas.....	2
CINFU.....	2
NORCAM.....	3
Enquadramento.....	4
1. Célula robótica do CINFU	5
1.1. Estrutura	5
1.1.1. Robô	5
1.1.2. Controlador da célula	6
1.2. Descrição da movimentação do robô	7
1.2.1. Modos de funcionamento do controlador	7
1.2.2. <i>Jogging</i>	8
1.2.3. Referenciais	10
1.2.4. Cinemática.....	11
1.3. Segurança	12
2. Programação manual.....	13
2.1. Introdução	13
2.2. Calibração de um plano de ferramenta	15
2.3. Criação de objetos de trabalho.....	18
2.4. Programação	20
2.4.1. Módulos	21
2.4.2. Rotinas	21

2.4.3.	<i>MainModule</i>	21
2.4.4.	Instruções de programação.....	21
3.	Programação <i>offline</i>	23
3.1.	Introdução	23
3.2.	<i>PowerMILL Robot Interface</i>	23
4.	Trabalho realizado	25
4.1.	Formação no robô	25
4.2.	Formação PRI.....	26
4.3.	Primeiro projeto em PowerMILL Robot Interface	26
4.3.1.	Calibração da ferramenta	27
4.3.2.	Modelação da célula em CAD	28
4.3.3.	Exportar as alterações da célula robótica	29
4.3.4.	Peça e percursos de maquinagem	32
4.4.	Projeto de maquinagem a 3 eixos.....	35
4.4.1.	Peça e percursos de maquinagem.....	35
4.4.2.	PRI	36
4.4.3.	Simulação	38
4.4.4.	Pós-processamento	38
4.4.5.	Maquinagem no Robô.....	40
4.5.	Projeto de maquinagem a 5 eixos.....	42
4.5.1.	Otimização da célula.....	42
4.5.2.	Medições no robô:	43
4.5.3.	Modelos CAD e ficheiros .dmt.....	46
4.5.4.	Peça para maquinagem a 5 eixos.....	48
	Conclusões	53
	Referências Bibliográficas	54
A.	Anexos.....	1

Lista de Figuras

Figura 1 - Instalações do CINFU - Centro de Formação Profissional da Indústria da Fundição.....	2
Figura 2 - Logótipo da NORCAM.	3
Figura 1-1 - Célula robótica do CINFU.....	5
Figura 1-2 -Numeração dos eixos do robô para o <i>jogging</i> eixo a eixo [6].	8
Figura 1-3 - Correlação entre os movimentos do <i>joystick</i> e a movimentação do robô [5].	9
Figura 1-4 - Referenciais definidos pelo fabricante: a) <i>World</i> . b) TCP ou sexto eixo (no caso do robô do CINFU).....	10
Figura 1-5 - Representação esquemática das cinemáticas da célula robótica. A vermelho estão representados os referenciais definidos pelo fabricante e a amarelo os definidos pelo utilizador: a) Cinemática <i>part-on-table</i> ; b) Cinemática <i>part-on-robot</i>	11
Figura 1-6 - Botões de paragem de emergência da célula robótica do CINFU.....	12
Figura 2-1 - Editor de programas do <i>flexpendant</i> : 1) Instrução de movimento; 2) Funções de programação; 3) Exemplo de uma chamada de sinal da célula [5].....	14
Figura 2-2 - Menu de criação de nova ferramenta calibrada no controlador do robô.	15
Figura 2-3 - Esquema da medição dos pontos de aproximação numa calibração de um plano de ferramenta [5].	16
Figura 2-4 - Criação de um plano de ferramenta no <i>flexpendant</i> da célula robótica do CINFU.....	17
Figura 2-5 - Pontos de referência para a criação de objetos de trabalho.	18
Figura 2-6 - Vista simplificada da estrutura de um programa.	20
Figura 3-1 - Ambiente de trabalho do <i>PowerMILL Robot Interface</i>	24
Figura 4-1 - Calibração da ferramenta ponteira acoplada no sexto eixo.	25
Figura 4-2 - Bloco de madeira de suporte construído na oficina de carpintaria do CINFU.	26
Figura 4-3 - Ferramenta caneta montada no suporte do <i>spindle</i>	27
Figura 4-4 - Primeira versão da célula virtual com incorporação da banca do <i>spindle</i>	29
Figura 4-5 -Numeração dos eixos do IRB-2400.	30

Figura 4-6 - Determinação da inclinação do plano de ferramenta.	31
Figura 4-7 - Peça do primeiro projeto em PRI.	32
Figura 4-8 - Simulação em <i>viewmill</i> da estratégia de acabamento por padrão.	32
Figura 4-9 - a) Simulação em PRI. b) Execução do projeto no robô. A ferramenta caneta traça o padrão numa folha de papel.	34
Figura 4-10 - Peça “Cowling” para o novo projeto de maquinagem.	35
Figura 4-11 - Estratégia de desbaste. a) Percurso; b) <i>Viewmill</i>	35
Figura 4-12 - Estratégia de acabamento. a) Percurso; b) <i>Viewmill</i>	36
Figura 4-13 - Nova configuração da célula. a) Célula virtual. Eixo de ferramenta (Azul) localizado no centro do suporte do <i>spindle</i> . b) Calibração do <i>spindle</i> com o auxílio da ferramenta ponteira.	37
Figura 4-14 - Posicionamento da peça.	37
Figura 4-15 - Simulação dos percursos.	38
Figura 4-16 - Pós Processamento.	38
Figura 4-17 - Cálculo de planos de trabalho do robô.	39
Figura 4-18 - Ficheiros dos módulos criados pelo pós-processador.	39
Figura 4-19 - Bloco de Poliuretano expandido.	40
Figura 4-20 - Resultado do desbaste da peça “cowling” no robô.	41
Figura 4-21 - Menu de calibração de <i>spindle</i>	42
Figura 4-22 - Duas ponteiras de calibração fabricadas no torno CNC e uma fresa esférica de 6mm de diâmetro.	43
Figura 4-23 - Calibração da ferramenta e medição dos comprimentos das ponteiras.	44
Figura 4-24 - Preenchimento da tabela de calibração de ferramentas com os dados recolhidos.	44
Figura 4-25 - Detecção do erro de criação do eixo de ferramenta.	45
Figura 4-26 - Edição da calibração do plano de ferramenta no ficheiro <i>IRB2400-16_CINFU.mtd</i>	46
Figura 4-27 - Modelo CAD da garra de dois dedos.	47
Figura 4-28 - Novo modelo CAD da banca e do <i>spindle</i>	48
Figura 4-29 - Ângulos de saída da peça.	48
Figura 4-30 - Desbaste: a) Percurso; b) <i>Viewmill</i>	49
Figura 4-31 - Acabamento paramétrico: a) Percurso; b) <i>Viewmill</i>	49

Figura 4-32 - Acabamento raso: a) Percurso; b) Viewmill.....	50
Figura 4-33 - Nova célula do robô ABB do CINFU.	50
Figura 4-34 - Posicionamento da peça.	51
Figura 4-35 - Simulação dos percursos.	51
Figura 4-36 - Maquinagem do bloco de poliuretano na forma final.	52
Figura A-1 - Elementos constituintes do controlador da célula robótica: a) <i>Flexpendant</i> ; b) Painel de controlo; c) Botões de acionamento da célula, ligação dos motores, seleção do modo de funcionamento e interrupção de emergência. ...	1
Figura A-2- Alcance do robô ABB IRB 2400-16.....	2
Figura A-3 -Robô do CINFU a executar uma rotina de trabalho criada durante a formação.	2
Figura A-4 - Tipos de singularidades.....	3
Figura A-5 - Erro de singularidade do primeiro projeto.	3
Figura A-6 - Resultado do primeiro projeto. Após desenhar o padrão o robô coloca a peça num suporte posicionado numa mesa. Rotina criada com programação manual.....	4
Figura A-7 -Lista de parâmetros do pós processador no projeto <i>cowling</i>	4

Abreviaturas e Siglas

CAM - Computer Aided Manufacturing

CAD - Computer Aided Design

PRI - PowerMILL Robot Interface

CNC - Controlo Numérico Computorizado

NC - Numerical Control

TCP - Tool Center Point

CINFU - Centro de Formação Profissional da Indústria de Fundição

PLC - Power Line Communication

Objetivo

O objetivo da dissertação centrou-se na implementação da solução CAM *PowerMILL Robot Interface* na célula robótica do CINFU. Para o efeito foi realizado um estágio presencial que permitiu o desenvolvimento do tema proposto assim como a familiarização e integração em ambiente empresarial e industrial. Esta dissertação dá continuidade ao trabalho desenvolvido em seminário “Aprendizagem da aplicação CAM para geração de percursos de maquinagem de 3, 3+2 e 5 eixos” e traça novos desafios pois é necessário aplicar os conhecimentos adquiridos à tecnologia das células robóticas.

Este estágio visa assim a configuração da célula robótica de forma a esta poder executar operações de maquinagem que possam ser implementadas posteriormente na indústria pois existe, da parte da direção do CINFU, o objetivo de utilizar a célula robótica para rebarbagem de peças de fundição.

As empresas

CINFU

O CINFU - Centro de Formação Profissional da Indústria de Fundição - foi criado em 24 de Abril de 1981 por um protocolo assinado entre o Instituto de Emprego e Formação Profissional (IEFP) e a Associação Portuguesa de Fundição (APF), tendo-se tornado em 27 de Maio de 1987 um Centro de Gestão Participada [1].



Figura 1 - Instalações do CINFU - Centro de Formação Profissional da Indústria da Fundição.

O CINFU tem por objetivo a valorização do sector da fundição através da formação profissional dos seus ativos e daqueles que nele pretendam ingressar [1]. Disponibiliza formação profissional através de cursos de educação e formação de adultos, serviços de apoio às empresas, desenvolvimento de protótipos e ensaios mecânicos, químicos e metalográficos. De realçar que o CINFU foi pioneiro em Portugal na oferta de formação em CAD/CAM [2].

NORCAM

A NORCAM - Engenharia e Design Industrial foi criada em 1991 e é uma empresa que se dedica à implementação de soluções industriais como sistemas de CAD/CAM/CAE, prototipagem rápida, digitalizadores tridimensionais e outros meios tecnológicos avançados numa ótica de *Concurrent Engineering* [3].



Figura 2 - Logótipo da NORCAM.

A NORCAM integra um conjunto de especialistas com experiência comprovada, formando uma equipa coesa nas áreas de Design Industrial, CAD/CAM/CAE e CNC. Entre os produtos disponibilizados pela NORCAM estão os sistemas de CAD/CAM, como por exemplo o *PowerMILL*, *PowerSHAPE* ou *FeatureCAM* [3].

Enquadramento

A célula robótica foi adquirida pelo CINFU de forma a poder dar formação aos seus alunos na área da robótica industrial. O robô tem, por isso, sido utilizado apenas em modo *teach-and-learn*, onde o operador desloca manualmente o robô para cada posição pretendida, memorizando-a de forma a criar assim uma trajetória. Este processo não só é lento como também ocupa a célula robótica durante o processo de programação. Naturalmente, a programação *teach-and-learn* não é adequada para realizar operações de maquinagem, uma vez que o número de pontos necessários é demasiado elevado.

De forma a tirar maior rendimento deste equipamento foi então sugerido que se utilizasse esta célula para operações de maquinagem, utilizando a solução *CAM PowerMILL Robot Interface*.

Como existe por parte da direção do CINFU o objetivo de poder vir a utilizar esta célula para rebarbagem de peças de fundição, esta dissertação dá o primeiro passo nesse sentido ao criar os conhecimentos fundamentais necessários para, num posterior trabalho, se substituir a fresa do *spindle* por uma rebarbadora. Pelo mesmo motivo optou-se por configurar a célula utilizando o robô na cinemática *part-on-robot*, onde a peça é manipulada pelo robô perante uma ferramenta estática pois esta cinemática é a mais adequada à rebarbagem de peças.

1. Célula robótica do CINFU

1.1. Estrutura

A célula robótica do CINFU é constituída pelo robô ABB IRB 2400-16 de 6 eixos, uma banca com suporte de fixação de *spindle*, mesa de calibração e suporte de ferramentas bem como uma célula de segurança. A célula possui também um controlador com uma consola portátil, denominada *FlexPendant*, de ecrã tátil e com um *joystick* que permite movimentar o robô manualmente, programar e executar rotinas de trabalho bem como operações de troca e calibração de ferramentas.



Figura 1-1 - Célula robótica do CINFU.

1.1.1. Robô

O robô ABB IRB 2400-16 é um dos robôs mais populares do mundo na sua classe pois apresenta alta eficiência na realização de diversas tarefas industriais. Possui 6 eixos de rotação, tem uma capacidade de carga de 20 kg e um alcance de 1,55m (*Vid* Figura A-2 dos Anexos). Este tipo de equipamento está geralmente associado a operações de soldadura de arco, operações de corte e rebarbagem bem como a manusear e montar componentes [4].

A *Tabela 1* descreve sumariamente algumas das características deste equipamento [4].

Tabela 1 - Características do robô ABB IRB 2400-16 [4].

Capacidade de Carga	20 Kg
Alcance	1,55m
Número de Eixos	6
Peso	380 Kg
Velocidade Máxima	450°/s no Eixo 6, 360°/s nos Eixos 4 e 5 e 150°/s nos eixos 1, 2 e 3.

1.1.2. Controlador da célula

O controlador ou painel de controlo é um dos equipamentos da célula robótica. Um dos componente mais importantes, o *flexPendant*, é a unidade de programação e manipulação do robô. Encontram-se também no controlador o interruptor de ligação da célula, seletor de modo de operação (manual, manual 100% e automático), botão de acionamento dos motores do robô, entradas USB e botão de paragem de emergência (*Vid* Figura A-1 dos Anexos). O *flexpendant* é uma unidade portátil que o utilizador deve transportar sempre consigo quando procede a operações com o robô. Possui um *joystick* com 3 graus de liberdade, o que significa que o robô pode ser movido em três direções diferentes simultaneamente segundo vários referenciais disponíveis, bem como um ecrã tátil para interação com o *software* do controlador. A velocidade do robô é proporcional à deflexão do *joystick*, mas nunca superior a 250 mm/s. É no *flexpendant* que se procede à programação de rotinas, calibração de ferramentas e criação de objetos de trabalho. Este equipamento possui também um botão de segurança conhecido como “botão do homem morto” que é responsável por acionar os motores do robô [5].

1.2. Descrição da movimentação do robô

O principal objetivo do *software* presente no controlador da célula robótica é o controlo dos movimentos do robô de forma a posicioná-lo corretamente consoante a tarefa que se pretende executar.

1.2.1. Modos de funcionamento do controlador

O controlador presente na célula robótica do CINFU permite três modos distintos de funcionamento: manual, manual 100% e automático. Para se seleccionar o modo de funcionamento com que se pretende trabalhar, roda-se uma chave de segurança no controlador na direcção do modo desejado (*Vid* Figura A-1 dos Anexos).

Em modo manual os movimentos do robô são controlados pelo *joystick* do *flexpendant* e os motores apenas se ativarão quando o “botão do homem morto” estiver pressionado. Este modo é utilizado para programação, verificação de rotinas e para reposicionamento do robô. Quando se opera em modo manual, o operador encontra-se perto do robô durante a sua manipulação. Para segurança dos operadores, a velocidade dos movimentos lineares do robô está reduzida para, no máximo, 250 mm/s conferindo assim uma maior segurança quando são executadas operações de alta precisão e que requerem proximidade ao robô para um melhor controlo. As operações normalmente feitas em modo manual são:

- *Jogging*;
- Criação e edição de programas em *Rapid*;
- Iniciar e parar a execução de programas.

Este é, portanto, o modo mais seguro para se manipular o robô quando o utilizador se encontra perto dele [5].

Em modo automático a função de segurança do “botão do homem morto” é ignorada e o robô opera à velocidade máxima sem qualquer interferência humana. A execução de programas apenas pode ser parada através do botão de stop do *flexpendant* ou dos botões de paragem de emergência presentes no controlador e no *flexpendant*. Não é possível programar nem movimentar o

robô usando o *joystick* enquanto este modo se encontra selecionado. Este é o modo de funcionamento do robô em serviço [5].

Em modo manual 100% o robô movimenta-se à velocidade máxima indicada no programa. Para além disso, para um programa ser executado é necessário não só pressionar o “botão do homem morto” mas também o botão *play* do *flexpendant* constantemente em simultâneo. Largar qualquer um destes botões resulta na interrupção imediata da rotina. Pelo perigo do robô a funcionar à velocidade máxima, este modo insere assim um nível de segurança extra permitindo observar a rotina a 100% da sua capacidade. Neste modo não é possível programar nem alterar parâmetros do sistema, servindo apenas para execução e verificação de programas *Rapid* [5].

1.2.2. *Jogging*

Para se movimentar o robô, é fundamental indicar segundo que referencial será feito este movimento. O termo *jogging* é usado para descrever a movimentação do robô e, geralmente, é feito de uma das seguintes formas:

- Movimentos eixo a eixo - neste tipo de movimentação seleciona-se qual o conjunto de eixos que se pretende manipular (1-2-3 ou 4-5-6) e move-se o *joystick* numa das suas 3 direções resultando na movimentação desse eixo;

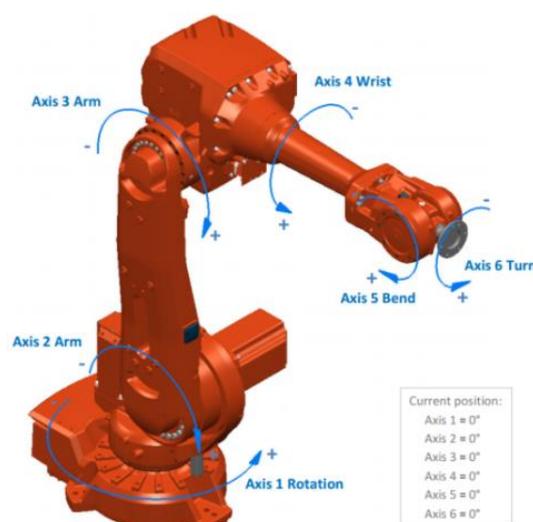


Figura 1-2 -Numeração dos eixos do robô para o *jogging* eixo a eixo [6].

- Movimentos lineares - movimentação do eixo de ferramenta ativo nas direções X, Y e Z do objeto de trabalho selecionado;
- Reorientação - o robô reorienta o eixo de ferramenta em torno de si próprio, mantendo as coordenadas da ponta da ferramenta sendo que apenas a inclinação do plano da ferramenta é afetada. Este método é mais comum para reorientação de ferramentas montadas no robô.

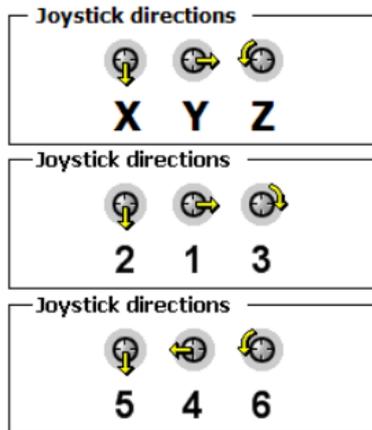


Figura 1-3 - Correlação entre os movimentos do *joystick* e a movimentação do robô [5].

1.2.3. Referenciais

O robô possui por defeito dois referenciais definidos pelo fabricante sobre os quais efetua os seus movimentos: *World* e TCP (*tool attachment point*). O referencial *World* está geralmente na base do robô. A frente do robô é na direção +X do referencial, Y é a direção perpendicular horizontal e Z a direção vertical, segundo a regra da mão direita. O referencial TCP está localizado na extremidade do último eixo do robô e é o local onde geralmente se acoplam as ferramentas, podendo estas serem garras, ponteiras, *spindles*, marcadores, sondas, eléctrodos, prensas, entre outros. No caso do robô do CINFU, o TCP está localizado na extremidade do sexto eixo e no controlador é sempre denominado dessa forma: o sexto eixo [5].

Na maioria dos robôs, a ferramenta é a entidade que se encontra em movimento e que se localiza no próprio robô, portanto, se a peça estiver acoplada na extremidade do robô, esta passa a ser considerada a ferramenta. No caso dos robôs ABB, a ferramenta é sempre a verdadeira ferramenta quer esta esteja numa banca de trabalho, acoplada no robô ou em qualquer outra parte da célula. Este detalhe irá ter influência na criação de planos de ferramenta e de objetos de trabalho pois estes são sempre criados em relação aos referenciais do fabricante e obedecem a regras muito restritas para evitar erros. Os objetos de trabalho são referenciais que indicam a localização de uma peça ou local de trabalho enquanto que os planos de ferramenta indicam onde se encontra a ponta da ferramenta. Estes referenciais são criados pelos utilizadores e vão depender também da cinemática da célula [6].

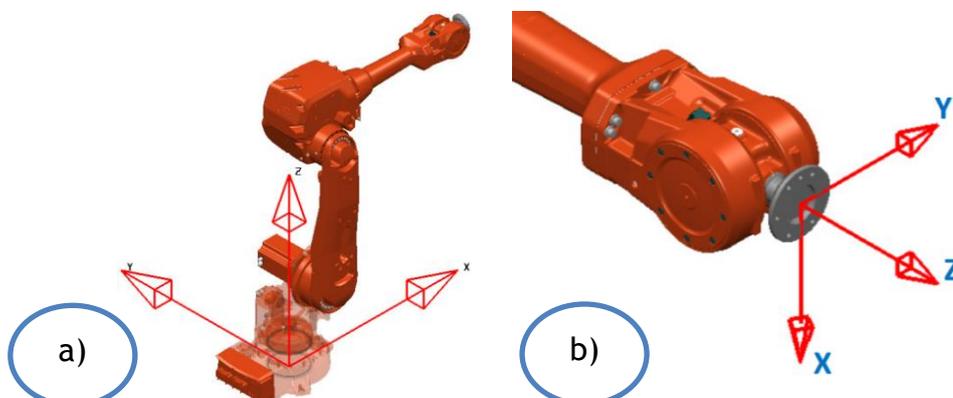


Figura 1-4 - Referenciais definidos pelo fabricante: a) *World*. b) TCP ou sexto eixo (no caso do robô do CINFU).

1.2.4. Cinemática

A célula robótica pode operar segundo duas cinemáticas diferentes: *part-on-robot* ou *part-on-table*. Na cinemática *part-on-robot*, o robô segura a peça enquanto que a ferramenta se encontra fixa. Na cinemática *part-on-table*, ocorre a situação contrária, o robô segura a ferramenta e a peça encontra-se fixa. Quando se opera na cinemática *part-on-table*, o referencial do sexto eixo (TCP) é utilizado como referência para criar os planos de ferramenta enquanto os objetos de trabalho são criados a partir do referencial do sexto eixo (Figura 1-5 a)). Quando se opera com a cinemática *part-on-robot*, o referencial do sexto eixo é utilizado como referência para criar objetos de trabalho, enquanto os planos de ferramenta são criados a partir do referencial *World* (Figura 1-5 b)). Este conjunto de regras resulta da característica única dos robôs ABB de reconhecer sempre a verdadeira ferramenta. Outras marcas consideram como ferramenta o objeto preso ao sexto eixo, quer seja uma peça ou uma ferramenta [6].

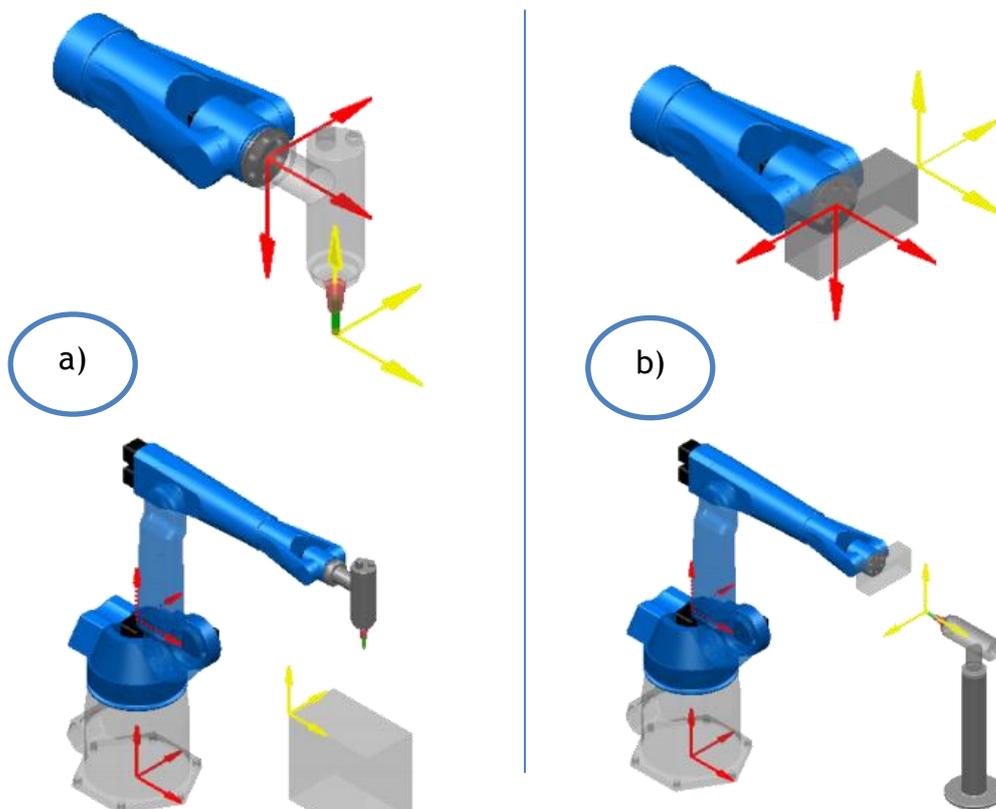


Figura 1-5 - Representação esquemática das cinemáticas da célula robótica. A vermelho estão representados os referenciais definidos pelo fabricante e a amarelo os definidos pelo utilizador: a) Cinemática *part-on-table*; b) Cinemática *part-on-robot*.

1.3. Segurança

Quando se trabalha em modo manual, deve-se ter em atenção a distância a que se está do robô e, sempre que possível, mantermo-nos fora da área de alcance deste a fim de evitar possíveis acidentes.

De forma a minimizar o risco de acidentes existe o já referido “botão do homem morto” no *flexpedant*. Quando pressionado até meia distância este ativa os motores permitindo a movimentação do robô. Caso o botão seja pressionado na totalidade ou largado, o robô automaticamente desliga os motores e interrompe o movimento. Isto é válido para os modos manual e manual 100% caso estejam a ser executados movimentos com o *joystick* ou esteja a correr uma rotina de trabalho. Em modo automático, é estritamente proibida a presença dos operadores dentro da célula. Existem também, quer no *flexpendant* quer no exterior da célula, botões de paragem de emergência que imediatamente desligam os motores do robô e interrompem as operações que estejam a decorrer.



Figura 1-6 - Botões de paragem de emergência da célula robótica do CINFU.

2. Programação manual

2.1. Introdução

A programação manual, também conhecida como programação *online* ou *teach-and-learn*, é efetuada diretamente no controlador do robô. Utilizando a *interface do flexpendant*, o operador desloca manualmente o robô para cada posição pretendida, memorizando-a para criar, assim, uma trajetória. Isto requer a utilização do próprio robô o que o ocupa durante o processo de programação.

No *software do flexpendant* existe um editor de programas que permite criar rotinas de trabalho, onde se memorizam as instruções dos movimentos, se inserem funções de programação e chamadas de sinais da célula. Quando é necessário adicionar a posição atual do robô ao programa, esta será dada segundo a posição e orientação do plano de ferramenta ativo em relação ao objeto de trabalho ativo. Isto significa que a posição do robô é sempre dada pela relação entre estes dois referenciais, ou seja, permite-nos saber onde está a ponta da ferramenta em relação ao objeto.

Memorizar uma posição instantânea do robô numa rotina é memorizar a localização da ferramenta em relação ao objeto de trabalho. Apesar de ser possível editar manualmente as coordenadas de uma posição, verifica-se que é pouco produtivo e muito menos preciso especialmente quando é necessário considerar a inclinação da ferramenta. Podem existir vários planos de ferramenta e objetos de trabalho gravados no *software* do robô mas apenas um plano de ferramenta e um objeto de trabalho podem estar ativos. As rotinas de trabalho são programadas em linguagem *Rapid* que é exclusivamente utilizada pelos robôs da marca ABB [5].

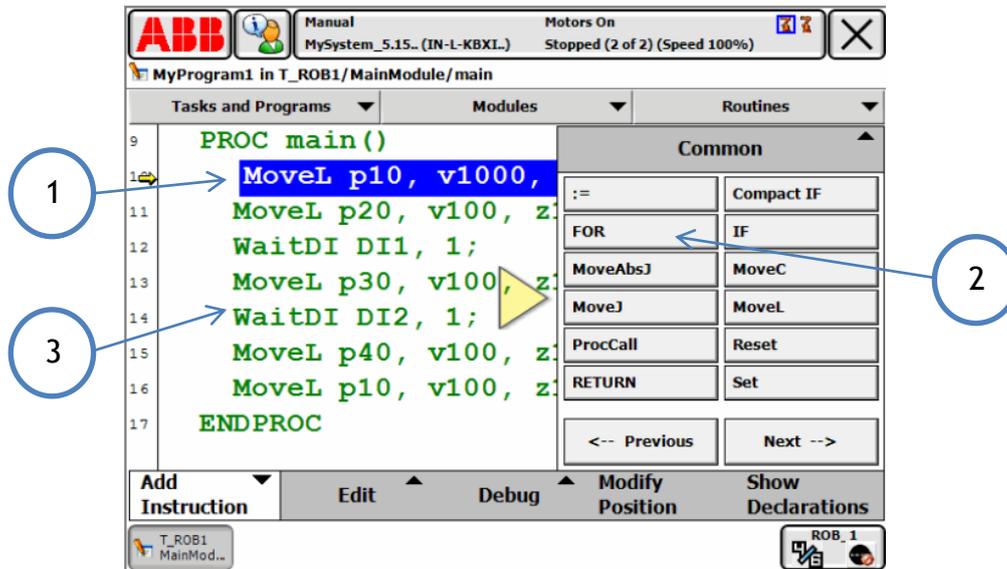


Figura 2-1 - Editor de programas do *flexpendant*: 1) Instrução de movimento; 2) Funções de programação; 3) Exemplo de uma chamada de sinal da célula [5].

Para se proceder à programação manual de uma rotina devem-se seguir os seguintes passos:

- Criar um plano de ferramenta calibrado;
- Criar um objeto de trabalho;
- Movimentar o robô para as posições adequadas às tarefas a realizar e memorizá-las;
- Inserir funções de programação e chamadas de sinais da célula se for necessário;
- Executar e analisar a rotina.

2.2. Calibração de um plano de ferramenta

O primeiro passo para se iniciar a programação de uma rotina de trabalho é a calibração de um plano de ferramenta. Utilizando o *flexpendant*, acede-se ao menu Painel de Controlo, seleciona-se *Tool Data* e cria-se uma nova ferramenta calibrada. Como referido anteriormente, o robô IRB 2400-16 pode operar segundo duas cinemáticas: *part-on-robot* e *part-on-table*.

Após a criação da nova calibração de ferramenta, existe uma variável booleana denominada *robhold* que pode assumir os valores *TRUE* ou *FALSE*. Se foi selecionado o valor *TRUE* significa que o robô segura a ferramenta, caso se selecione *FALSE* significa que a ferramenta está fixa e o robô segura a peça. Isto é muito importante para todo o processo de calibração pois definirá qual a cinemática que irá ser utilizada e, conseqüentemente, segundo que referencial do fabricante se criará o novo plano de ferramenta. Adicionalmente, quando o robô segura a ferramenta (*robhold=FALSE*), as coordenadas dos pontos de calibração são calculadas em função do referencial do sexto eixo e, no caso da cinemática de ferramenta fixa (*robhold=TRUE*), as coordenadas dos pontos de calibração são calculadas segundo o referencial *World* [5].

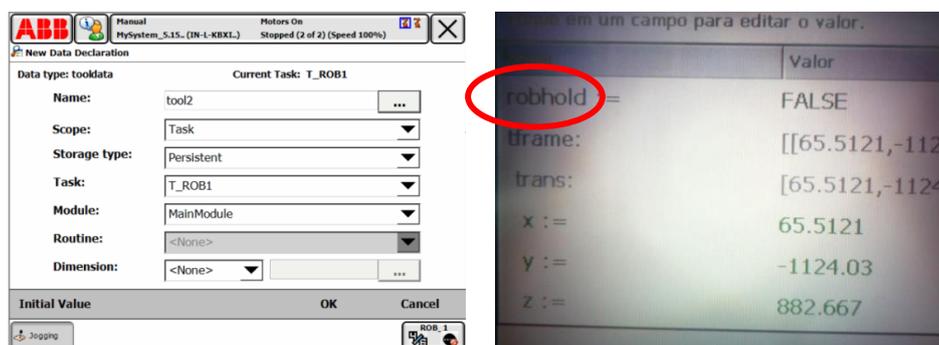


Figura 2-2 - Menu de criação de nova ferramenta calibrada no controlador do robô.

Depois deste passo, deve-se escolher um dos vários métodos de calibrar presentes no menu do programa: TCP, TCP Z e TCP XZ. Ambas as cinemáticas partilham estes métodos de calibração mas o utilizador deve proceder de forma diferente consoante a cinemática que está a utilizar. Estes métodos são de execução muito rápida graças à sua simplicidade e justificam, em parte, a razão pela qual este tipo de robôs é tão utilizado na indústria. Existem

também desvantagens relacionadas com a imprecisão do método visto que são muito mais suscetíveis a erros por parte do utilizador. Os três métodos são semelhantes e seguem um conjunto simples de passos para se proceder a uma calibração de um plano de ferramenta:

TCP

- Aproximar a ponta da ferramenta de um ponto de referência o mais perto possível (geralmente as células tem ponteiras de calibração no armazém de ferramentas próprias para estes processos);
- Registrar o ponto selecionando “Ponto 1” clicando “modificar posição” no menu;
- Alterar a direção da ferramenta e fazer nova aproximação;
- Registrar o ponto selecionando “Ponto 2” clicando “modificar posição”;
- Repetir este processo para o número de pontos desejado. O programa permite entre 3 e 9 pontos e quantos mais pontos se utilizarem mais precisa será a calibração.

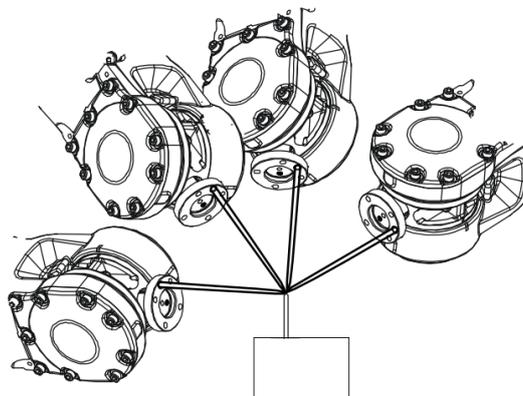


Figura 2-3 - Esquema da medição dos pontos de aproximação numa calibração de um plano de ferramenta [5].

TCP Z

- Fazer o mesmo processo que em TCP para o número de pontos seleccionados;
- Cria-se um ponto extra “Z” cuja posição deve ser medida na direcção perpendicular ao plano de ferramenta pretendido e indica a direcção do eixo dos Z através de um vetor entre o último ponto medido e este novo ponto, fixando apenas o eixo dos Z. Este ponto é chamado de ponto de alongamento e deve ter a mesma orientação do último ponto de aproximação utilizado.

TCP XZ

- Fazer o mesmo processo que em TCP Z para o número de pontos seleccionados;
- Ponto extra “X” cuja posição na horizontal do plano de trabalho no sentido positivo do eixo dos X. Este método fixa todos os eixos pois fixando Z e X, Y fica automaticamente fixo também.

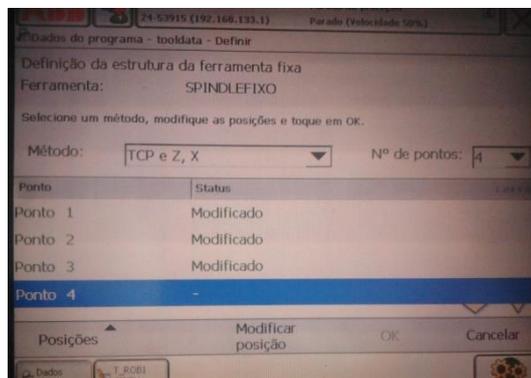


Figura 2-4 - Criação de um plano de ferramenta no *flexpendant* da célula robótica do CINFU.

No final o programa faz uma média das coordenadas X, Y e Z dos pontos TCP e daqui resultam as coordenadas da ponta da ferramenta e quanto mais pontos se usarem mais rigorosa será a medição. Os métodos TCP Z e TCP XZ fazem na mesma a média e para além disso fixam as direcções Z e X ficando o plano de ferramenta completamente definido.

2.3. Criação de objetos de trabalho

Um objeto de trabalho é um sistema de coordenadas que define um plano criado pelo utilizador no espaço da célula de forma a simplificar a programação de rotinas assim como facilitar o *jogging* do robô. Sempre que se pretende memorizar um movimento, deve estar sempre selecionado um objeto de trabalho e um plano de ferramenta.

Um objeto de trabalho é uma variável que guarda o nome do objeto de trabalho, as coordenadas X, Y, Z e o vetor de orientação. Para se criar um objeto de trabalho, acede-se ao menu de configuração e, após selecionar um plano de ferramenta (pode-se utilizar o sexto eixo definido pelo fabricante ou qualquer outro definido pelo utilizador), memorizam-se 3 pontos num plano que se pretenda que seja o nosso objeto de trabalho: X1, X2 e Y1. Após este procedimento, o robô vai gravar esta informação e criar naquela posição um objeto de trabalho segundo o qual se podem programar rotinas de trabalho [5].

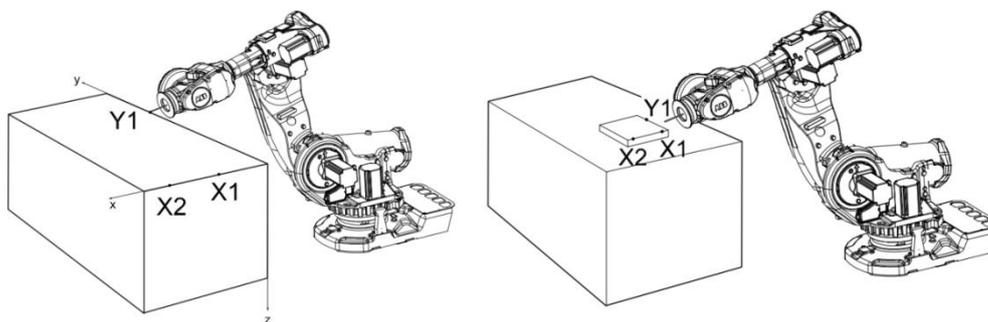


Figura 2-5 - Pontos de referência para a criação de objetos de trabalho.

A seleção dos pontos X1, X2 e Y1 vai determinar também a direção do eixo Z. O ponto X1 é sempre a coordenada (0,0,0) do referencial estando o X2 na direção positiva de X e o Y1 sempre na direção positiva de Y. No caso da Figura 2-5, como o ponto Y1 está à direita de X1, o sentido positivo do eixo do Z é para baixo pois segue a regra da mão direita.

A criação de objetos de trabalho não só simplifica o trabalho por se poder atribuir um referencial para cada parte da célula como também impede que se tenha de reprogramar na totalidade uma rotina caso seja necessário mover o local de trabalho. Se isso acontecer, apenas é necessário recalcular o

próprio objeto de trabalho e todos os movimentos da rotina ficarão corrigidos. Se for calculada uma rotina segundo o objeto de trabalho da base do robô (*world*) e seja necessário que a mesma rotina seja executada num local diferente da célula, tem de se corrigir ponto a ponto a rotina pois a sua localização mudou em relação à base do robô. Se a rotina tiver o seu próprio objeto de trabalho pode ser movida para qualquer parte da célula que os pontos estarão sempre corretos em relação ao objeto de trabalho pois este move-se em conjunto com eles, sendo apenas necessário recalcular a posição relativa do objeto de trabalho em relação à base do robô. Isto é muito comum na indústria quando é necessário mover uma banca onde se executa uma rotina para, por exemplo, se colocar uma outra banca com um novo trabalho para o robô [5].

2.4. Programação

Um programa consiste em instruções e dados codificados em linguagem de programação *Rapid*, que controlam o robô e equipamentos periféricos. A memória do programa contém um sistema de módulos.

O programa do robô é bastante complexo envolvendo conceitos de *task*, *program module*, *system modules*, entre outros. Do ponto de vista do utilizador, apenas serão abordados três conceitos principais:

- Módulos;
- Rotinas;
- *MainModule*.

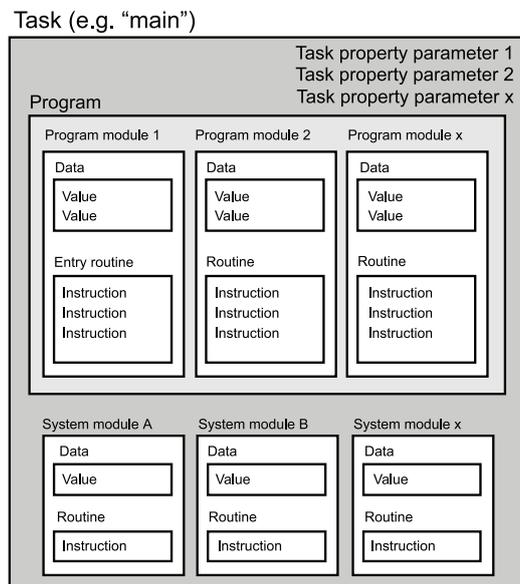


Figura 2-6 - Vista simplificada da estrutura de um programa.

2.4.1. Módulos

Módulos são ficheiros de texto (extensão .MOD) nos quais são criadas as rotinas de trabalho. Geralmente, no cabeçalho de um módulo são declaradas as variáveis das rotinas (isto inclui as variáveis objeto de trabalho e plano de ferramenta). É boa prática ter sempre as variáveis e as rotinas pelas quais são evocadas no mesmo módulo pois pode haver erros de privacidade das variáveis e estas não serem acessíveis a partir de outros módulos. Os módulos podem ser criados diretamente no programa ou carregados para o controlador provenientes de outras fontes (no caso de programação *Offline*) [5].

2.4.2. Rotinas

As rotinas são conjuntos de instruções escritas nos módulos em linguagem *Rapid* que podem ser chamadas para execução. Não é possível executar um módulo inteiro que contenha várias rotinas sendo apenas possível chamar e executar uma rotina de cada vez. As rotinas servem para subdividir o programa em pequenas partes para se obter um módulo em blocos, o que facilita a sua leitura [5].

2.4.3. MainModule

Nos robôs ABB existe um Módulo específico que, sem ele, não é possível executar rotinas em modo automático. Este ficheiro .MOD deve obrigatoriamente ter o nome *MainModule* e deve conter uma rotina com o nome *Main* [5].

2.4.4. Instruções de programação

Como já referido, pode-se inserir instruções de movimentos, funções de programação e chamadas de sinal do robô ao programar uma rotina de trabalho. Se uma nova instrução for adicionada, ela será colocada depois da instrução que estiver selecionada. Se a instrução selecionada está no início da rotina ou numa instrução composta (*IF*, *FOR*, *WHILE* ou *TEST*), é necessário escolher se a nova instrução será colocada antes ou depois desta instrução primária (no sentido de uma pergunta). Entretanto, se existir somente uma instrução nesta rotina ou numa instrução composta, a nova instrução será sempre acrescentada depois da anterior [5].

As instruções de posicionamento contêm as seguintes informações:

- Tipo de trajeto (linear, junta);
- Posição de destino para qual o robô se move;
- Velocidade;
- Tamanho da zona - Esta variável irá suavizar mudanças de direção bruscas traçando arcos de movimento de forma a diminuir o stress nas articulações do robô. Mudanças bruscas causam muita vibração e são prejudiciais para a precisão do robô;
- Plano de ferramenta atual;
- Objeto de trabalho atual.

O seguinte exemplo de uma linha de código de uma rotina é referente a um movimento de junta para o ponto *p1*, velocidade *v50*, tamanho de zona *z1*, utilizando o plano de ferramenta *tool1* e o objeto de trabalho *work1*.

```
MoveJ p1 v50, z1, Tool1\WObj:=Work1;
```

Movimento de junta - Quando a precisão do caminho não é muito importante, este tipo de movimento é utilizado para mover a ferramenta rapidamente a partir de uma posição para outra. Os movimentos de junta também permitem que um eixo se desloque de qualquer local para outro dentro do seu espaço de trabalho num único movimento e a uma velocidade constante [5].

Movimento linear - Durante a movimentação linear, o plano de ferramenta viaja ao longo de uma linha reta entre os pontos de início e de destino. A orientação da ferramenta permanece constante durante todo o movimento, a menos que tenha sido programada uma reorientação [5].

3. Programação *offline*

3.1. Introdução

A programação *offline* é feita no computador sem se recorrer ao controlador do robô. Existem diversos *softwares* que permitem programar rotinas para os robôs ABB, entre eles o *Robot Studio* da ABB e o *PowerMILL Robot Interface*. Neste projeto foi utilizado o *software PowerMILL Robot Interface* como ferramenta de programação de rotinas em modo *offline*.

3.2. *PowerMILL Robot Interface*

O *PowerMILL Robot Interface* é um *plugin* criado para o *software* CAM *PowerMILL* que permite programar percursos de maquinagem para serem executados por robôs. Esta é, portanto, a grande vantagem deste *software* pois permite utilizar o robô para executar operações de maquinagem da mesma forma que um centro de maquinagem convencional. Outras vantagens são as dimensões das peças a serem maquinadas bem como o alcance de certas zonas na peça a maquinar onde os robôs têm, geralmente, superioridade sobre as CNC. Em contrapartida está a rigidez do robô que é menor que a das CNC porque, como possuem vários eixos e cada eixo deflete ligeiramente devido às folgas das articulações, vai existir uma maior vibração da peça ou da ferramenta presa no último eixo do robô. Esta vibração será maior quanto mais distendido estiver o robô.

Para simular um percurso em *PowerMILL Robots Interface*, existe uma sequência de passos mínima a seguir:

- Carregar o robô da base de dados;
- Definir a origem da peça;
- Definir as restrições iniciais;
- Simular o percurso;
- Analisar a simulação;
- Definir o número da ferramenta e a origem da peça (objeto de trabalho);
- Pós-processar o percurso de maquinagem.

O *PowerMILL Robot Interface* consiste em duas janelas incorporadas no *PowerMILL*. A janela vertical possui os menus Biblioteca de Robô, Célula de Robô, Controlo de Robô e Programa de Robô enquanto a janela horizontal possui as ferramentas para abrir e analisar simulações [6].

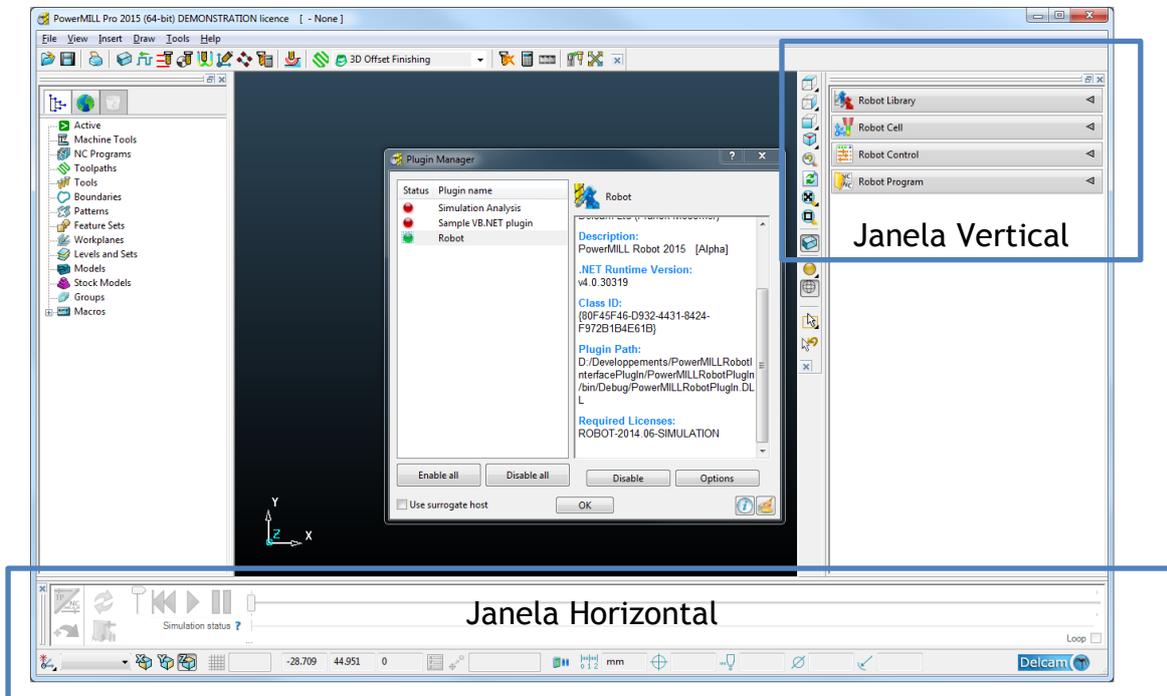


Figura 3-1 - Ambiente de trabalho do *PowerMILL Robot Interface*.

4. Trabalho realizado

4.1. Formação no robô

A formação no robô foi direcionada à programação online. Esta consistiu inicialmente em movimentar o robô segundo os seus diversos eixos bem como segundo os referenciais do fabricante. Esta primeira abordagem foi útil para treinar o *jogging* do robô e familiarizar com a sensibilidade do *joystick*. Posteriormente foram criados diversos objetos de trabalho usando o *flexpendant* do controlador e criadas também diversas calibrações de ferramenta. Para sumarizar esta formação inicial foi calibrada uma ponteira fixa no sexto eixo que iria posteriormente ser utilizado para calibrar as ferramentas fixas no *spindle* bem como efetuar medições na célula robótica por métodos de engenharia inversa. Esta ponteira é uma das ferramentas disponíveis no armazém de ferramentas da célula robótica.

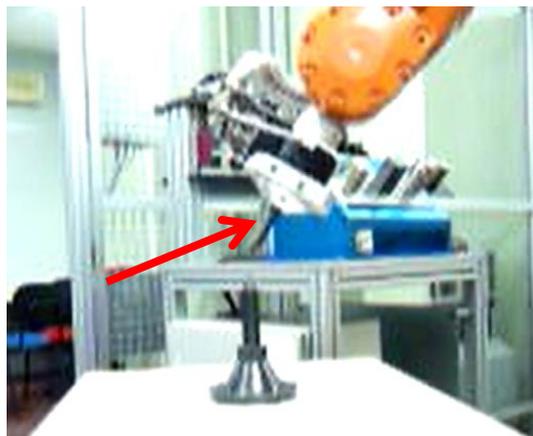


Figura 4-1 - Calibração da ferramenta ponteira acoplada no sexto eixo.

Para completar a formação foram também criadas várias rotinas de trabalho simples no controlador do robô que permitiram uma maior familiarização com a linguagem de programação Rapid e com os procedimentos de programação (Vid Figura A-3 dos anexos).

4.2. Formação PRI

A formação em *PowerMILL Robot Interface* foi feita através de cursos de formação disponibilizados com o software sob a forma de tutoriais e exercícios. Estes tutoriais visavam a familiarização com o software, permitiam a criação de projetos simples, análise de simulações e correção de erros. Foram necessários 9h de contacto com o software para completar a formação.

4.3. Primeiro projeto em PowerMILL Robot Interface

Após a realização dos tutoriais de formação procedeu-se à realização de um projeto em *PowerMILL Robot Interface* para testar o pós-processador do robô ABB do CINFU. Este primeiro projeto consistia em desenhar um padrão simples numa superfície plana. O objetivo era explorar as funcionalidades do PRI e proceder às primeiras configurações do robô. Em termos de complexidade era um projeto relativamente simples o que permitiu mais rapidamente obter resultados e familiarizar com as funcionalidades do PRI.

De forma a preservar a integridade da célula robótica e, como se estava ainda numa fase muito embrionária da configuração da célula, foi decidido que não se iria trabalhar com ferramentas de corte neste projeto inicial. Foi por isso utilizada como ferramenta fixa uma caneta de acetato montada no suporte do *spindle* do CINFU. O bloco foi montado numa placa de madeira fabricada na oficina de carpintaria do CINFU e preso na garra de dois dedos.



Figura 4-2 - Bloco de madeira de suporte construído na oficina de carpintaria do CINFU.

4.3.1. Calibração da ferramenta

Como já referido, a ferramenta utilizada foi uma caneta de acetato. Esta caneta está inserida num suporte e foi criada pelos operadores do CINFU para dar formação aos seus alunos em programação manual. Esta é montada no robô e utilizada para desenhar padrões em folhas de papel suportadas por caixas de cartão pois, devido à inexperiência dos alunos, eram comuns as colisões de ferramentas com as peças. Com esta solução, mesmo que haja uma colisão, a ferramenta não fica danificada pois as caixas de cartão absorvem o impacto sem haver danos significativos quer do robô quer das ferramentas.

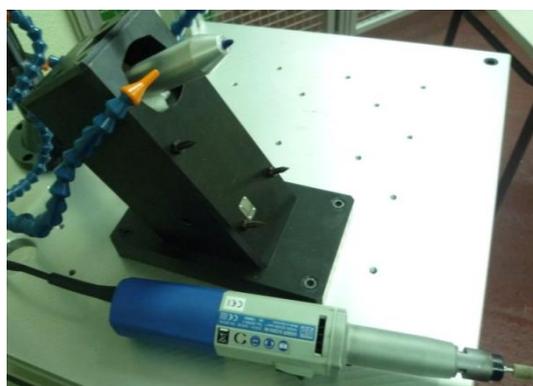


Figura 4-3 - Ferramenta caneta montada no suporte do *spindle*.

O primeiro passo foi retirar o *spindle* do suporte e substituí-lo pela caneta. De seguida montou-se no robô a ferramenta ponteira que foi calibrada durante a formação no robô. Como a ponteira se encontra calibrada, as coordenadas da ponta são conhecidas e vão servir como referência para a calibração da caneta e assim criar um plano de ferramenta na ponta desta. No menu de *jogging* deve-se indicar que se está a utilizar a ferramenta Ponteira selecionando o plano de ferramenta calibrado anteriormente, caso contrário não serão as coordenadas da ponta da ponteira que serão memorizadas. Em seguida foi utilizado o método TCP, com a variável *robhold* em *FALSE* para se calibrar o plano de ferramenta da caneta. Foram medidos quatro pontos fazendo *jogging* e aproximando o máximo possível a ponta da ponteira da ponta da caneta. Após calcular a média dos pontos memorizados é-nos dado

as coordenadas da ponta da caneta [5]. O suporte de *spindle* em que a caneta foi colocada tem uma elevação de 60° em relação ao plano XY da base do robô. Quando a ferramenta, neste caso a caneta, é colocada neste suporte vai ficar com uma elevação de -30° em relação ao mesmo plano. Este foi portanto o valor utilizado para calcular o vetor de direção da ferramenta, visto que o método TCP não nos permite obter uma direção, apenas as coordenadas de um ponto e os outros métodos são pouco precisos.

4.3.2. Modelação da célula em CAD

O PRI utiliza modelos virtuais da célula robótica para proceder à simulação de percursos de maquinagem. A célula virtual deve por isso ser modelada o mais próximo da realidade possível para que a simulação seja rigorosa. Os erros nas simulações mais comuns são singularidades, colisões e alcance dos limites dos eixos. Caso não esteja corretamente modelada a célula, pode acontecer que hajam simulações que não apresentem quaisquer erros mas que na realidade haja uma colisão, o que pode danificar o robô ou outros elementos da célula. O robô é “cego” e se não lhe for indicado onde se encontram os elementos da célula (bancas, paletes, *spindle*, etc) ele pode vir a chocar com os mesmos devido à trajetória que calculou não ter em consideração que deveria se desviar de um determinado objeto.

Utilizando o robô com o objeto de trabalho *World* e a ferramenta Ponteira selecionados fez-se engenharia inversa à célula. Para o efeito criou-se uma nova rotina no módulo pré existente “CINFU.MOD” e sempre que se pretendia saber a coordenada de um ponto algures na célula, era apenas necessário memoriza-lo com a instrução *MoveL*. Era posteriormente extraído das instruções *MoveL* as coordenadas X, Y e Z desses pontos memorizados. Uma vez obtidos todos os pontos necessários procedeu-se à alteração do ficheiro CAD do robô. Com o auxílio do software CAD *PowerSHAPE* e das coordenadas adquiridas criou-se então a banca e o suporte do *spindle*, acrescentando estes elementos ao modelo do robô que já existia, fornecido pelo engenheiro Alexandre Bastos da DELCAM.

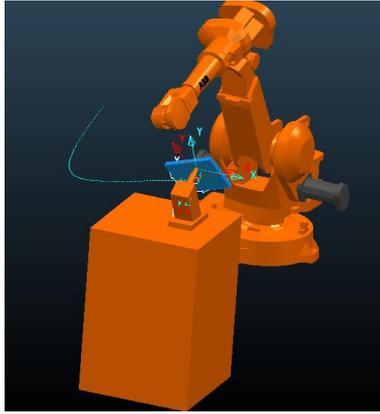


Figura 4-4 - Primeira versão da célula virtual com incorporação da banca do *spindle*.

4.3.3. Exportar as alterações da célula robótica

O modelo virtual da célula é guardado em vários ficheiro com extensão *.dmt*. Cada um destes ficheiros corresponde a uma parte do robô ou da célula. No caso da banca do *spindle* que foi criada, esta parte do desenho CAD será exportada num ficheiro independente denominado *spindle.dmt*. Este ficheiro será guardado com os restantes que compõe as várias partes do robô na pasta ABB CINFU. Como esta parte da célula não existia é necessário fazer o PRI reconhece-la. Para isso adiciona-se uma entrada no ficheiro *IRB2400-16_CINFU.mtd*. Este ficheiro organiza as várias partes da célula e correlaciona os movimentos dos vários eixos entre si. Para se adicionar o *spindle* acrescentam-se as seguintes linhas de código:

```
<!--Spindle fixo-->
<machine_part NAME="Spindle">
  <model_list>
    <dmt_file>
      <path FILE="ABB/Spindle.dmt" />
      <rgb R="255" G="110" B="0" />
    </dmt_file>
  </model_list>
</machine_part>
```

Esta sequencia de código XML contem um comentário inicial que indica que o *spindle* é fixo, define uma nova *machine part*, indica a localização do ficheiro *.dmt* correspondente à banca do *spindle* e a cor do mesmo em RGB. No caso dos elementos que compõe os eixos do robô existe uma secção de código sobre os limites de rotação [6].

```

<machine_part NAME="RobotAxis2">
  <axis>
    <control_info ADDRESS="B" MIN="-105" MAX="105" PRIORITY="AUTO"
HOME="0" VALUE="0"/>
    <simple_rotary X="100.0" Y="0.0" Z="615" I="0" J="1" K="0" />
  </axis>
  <model_list>
    <dmt_file>
      <path FILE="ABB/BC.dmt" />
      <rgb R="255" G="110" B="0" />
    </dmt_file>
  </model_list>
</machine_part>

```

Este código encontra-se encapsulado entre si de forma a que o movimento dos eixos posteriores sejam dependentes dos que lhes antecedem. Portanto, a informação do sexto eixo está encapsulada no quinto eixo, estando a do quinto eixo encapsulada no quarto e assim sucessivamente. Assim, movendo o eixo 1 fará mover todos os seguintes enquanto que mover o eixo 5 apenas fará mover o eixo 6 e não os anteriores. Isto é crucial para a sinergia da célula virtual.

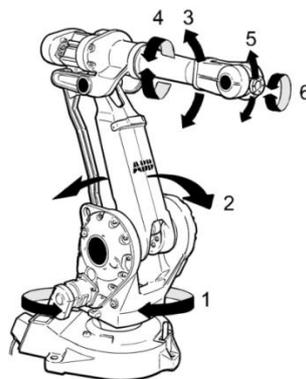


Figura 4-5 -Numeração dos eixos do IRB-2400.

No ficheiro *IRB2400-16_CINFU.mtd* é necessário alterar também as coordenadas do plano de ferramenta para as coordenadas da calibração da caneta. Para o efeito é necessário modificar a seguinte linha de código [6]:

```

<!-- Where the tool is attached (from world workplane) -->
  <head_attach_point PART="head" X="0" Y="0 Z="0" I="0" J="1" K="0" U="0"
V="0" W="0" />

```

Devido ao facto de ser necessário os valores de i, j e k correspondentes à orientação da ferramenta, é necessário converter os -30° nestes valores. A forma mais simples passa por criar uma linha no desenho CAD com origem no ponto correspondente à ponta da caneta e com 30° de elevação. Desenha-se sobre a linha um referencial orientado com o eixo dos Zs coincidente com a linha e obtêm-se os valores de i, j k pelas propriedades deste referencial criado.

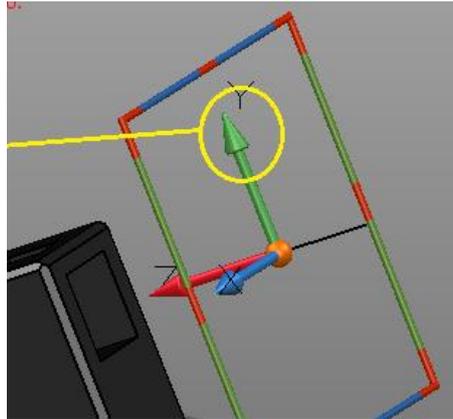


Figura 4-6 - Determinação da inclinação do plano de ferramenta.

Obtendo estes valores procede-se à alteração das coordenadas na linha de código correspondente ao plano do eixo de ferramenta:

```
<!-- Where the tool is attached (from world workplane) -->  
<head_attach_point PART="head" X="60.64" Y="-1173.23" Z="844.66" I="0"  
J="-0.866025" K="-0.5" U="1" V="0" W="0" />
```

4.3.4. Peça e percursos de maquinagem

A peça para o primeiro teste foi um paralelepípedo de 100x100x20 mm modelado em *PowerSHAPE* com um padrão em modelo de arame. Este padrão foi utilizado para criar uma estratégia de acabamento por padrão.

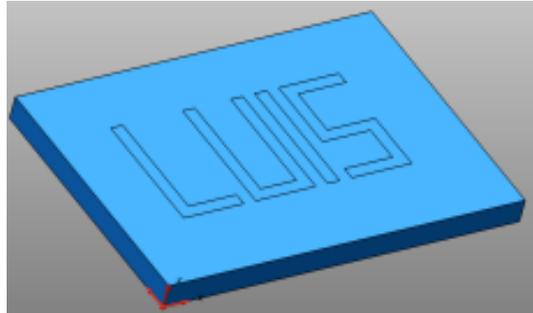


Figura 4-7 - Peça do primeiro projeto em PRI.

Seguiram-se um conjunto de passos para criar o percurso de maquinagem [7]:

- Importação do modelo CAD para o *PowerMILL*;
- Criação de uma ferramenta esférica de comprimento 2mm e de raio 2mm (dimensões da ponta da caneta);
- Criação do bloco;
- Calculo das alturas de movimentos rápidos da ferramenta;
- Alteração do ponto inicial e final para centro da peça seguro;
- Criação de um plano de trabalho no topo do bloco;
- Criação de um padrão utilizando o modelo de arames;
- Criação de uma estratégia de acabamento por padrão, com -1 mm de sobre espessura e 0 mm de tolerância;
- Simulação do percurso.

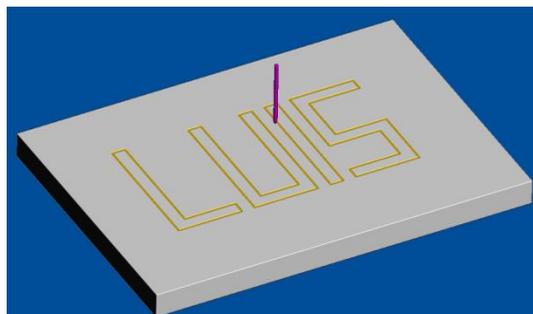


Figura 4-8 - Simulação em *viewmill* da estratégia de acabamento por padrão.

Caso este percurso fosse destinado a uma CNC seria agora necessário importar o modelo da máquina correspondente, proceder a uma simulação para verificar se existem colisões e criar um programa NC através do pós-processador da respetiva máquina. No caso dos robôs, o PRI é a ferramenta utilizada para esta etapa do processo.

O primeiro passo foi importar o robô modelado anteriormente. Na biblioteca de robô clicamos em adicionar novo robô e selecionamos a pasta correspondente ao robô ABB do CINFU. Esta pasta deve conter a sub pasta com os ficheiros *.dmt*, o ficheiro *IRB2400-16_CINFU.mtd* e o ficheiro *IRB2400-16_CINFU.RobConfig*. Sempre que se procedem a alterações quer nos ficheiros de configuração quer nos ficheiros *.dmt* das partes da célula é necessário importar novamente o robô de forma a atualizar a célula [6].

Após importar o robô verifica-se que existe uma colisão pois a robô e a peça partilham a mesma origem. Para evitar este incomodo pode-se alterar a origem das peças de forma a que não seja a mesma que o robô mas sim um local mais conveniente ao utilizador. Como estamos a trabalhar na cinemática *part-on-robot* seria ideal que as peças ao serem importadas ficassem localizadas no sexto eixo do robô. Para isso, no ficheiro *IRB2400-16_CINFU.mtd* procede-se à alteração da seguinte linha de código:

```
<table_attach_point PART="table" X="940" Y="0" Z="1455" I="1" J="0" K="0" U="0" V="0" W="-1" />
```

Esta linha de código define a localização da origem das peças quando são importadas. É possível confirmar esta informação pelo ficheiro *IRB2400-16_CINFU.RobConfig* que possui as coordenadas dos eixos definidos pelo fabricante para o respetivo modelo. No caso do robô ABB IRB2400-16 as coordenadas presentes neste ficheiro por defeito para os referenciais *World* e *6th Axis* são:

```
<Workplane Name="World Workplane" X="0" Y="0" Z="0" Rz="Auto" Ry="Auto" Rx="Auto" Type="RobotWorld" />
<Workplane Name="6th axis Workplane" X="940" Y="0" Z="1455" Rz="Auto" Ry="Auto" Rx="Auto" Type="RobotAxis6" />
```

Uma vez alterada a origem das peças, importa-se novamente o robô para atualizar e verifica-se que a peça aparece já localizada no sexto eixo do robô. No PRI procede-se à simulação do percurso gerada no *plugin* PRI [6].

Posteriormente analisa-se a simulação de forma a verificar a existência de erros. Se se verificarem colisões, singularidades ou erros de limite de eixo deve-se alterar a posição inicial da peça, rodando-a sobre o eixo dos Zs ou fazendo com que o robô pegue a peça numa posição diferente alterando a posição inicial com o menu de posicionamento da peça. No caso deste percurso foi necessário que o robô pegasse a peça de forma a esta ficar a 90° em relação ao sexto eixo. Isto permitiu eliminar uma singularidade de punho que era recorrente em todas as simulações e que não foi possível de corrigir apenas utilizando a rotação da posição inicial em torno de Z (*Vid* Figura A-5 dos anexos). Uma singularidade ocorre quando um movimento do robô pode ser executado por dois eixos diferentes. Isto produz um erro em que o robô não sabe qual dos eixos utilizar para efetuar o movimento e resulta num dos eixos rodar até ao seu limite (aparecendo uma janela de erro no *flexpendant*). Desta rotação até ao limite do eixo podem resultar colisões com a célula e com próprio robô bem como rebenamento de cabos, danos nas peças, entre outros. As singularidades mais comuns podem ser vistas na Figura A-4 dos Anexos [6]. Por fim, procedeu-se à exportação do projeto criando um ficheiro .MOD através do pós processador ABB CINFU. Carregou-se a rotina no robô e este executou-a na perfeição, em modo manual, produzindo o padrão do percurso numa folha de papel colocada no suporte de madeira (*Vid* Figura A-6 dos Anexos).

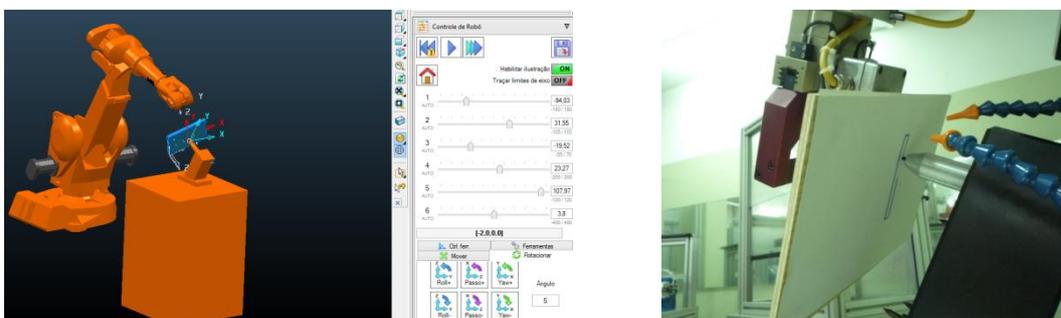


Figura 4-9 - a) Simulação em PRI. b) Execução do projeto no robô. A ferramenta caneta traça o padrão numa folha de papel.

4.4. Projeto de maquinagem a 3 eixos

Foi criado um segundo projeto em PowerMILL Robots Interface de forma a iniciar projetos de maquinagem utilizando ferramentas de corte.

4.4.1. Peça e percursos de maquinagem

O primeiro passo para a criação deste projeto de maquinagem foi a importação da peça para o *PowerMILL*. De seguida foram criadas uma estratégia de desbaste e uma estratégia de acabamento.

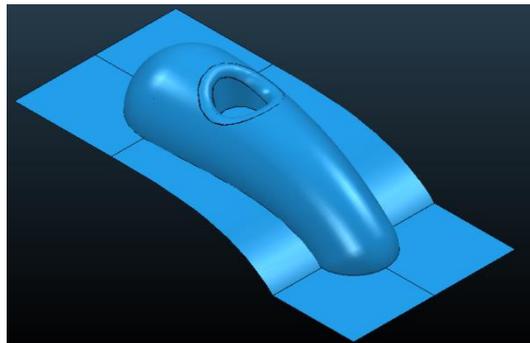


Figura 4-10 - Peça “Cowling” para o novo projeto de maquinagem.

Desbaste - A estratégia de desbaste selecionada foi um desbaste de modelo, com um passo lateral de 2 mm e um passo vertical de 8 mm. A ferramenta utilizada para esta estratégia foi uma fresa esférica de 8 mm. Após criar a estratégia esta foi simulada no *viewmill* [7].

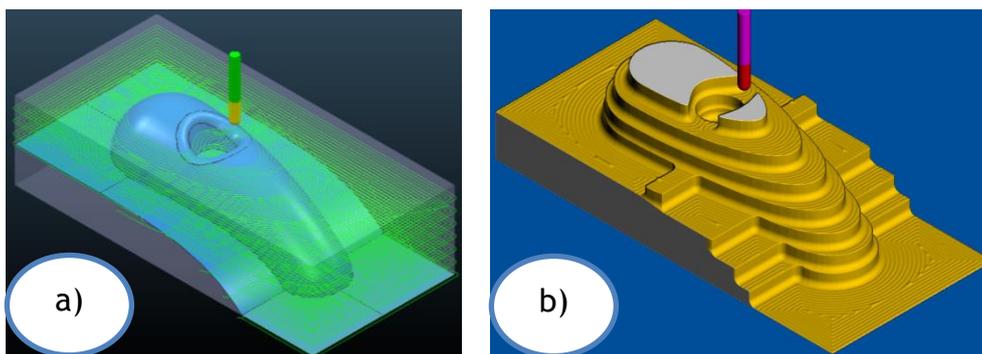


Figura 4-11 - Estratégia de desbaste. a) Percurso; b) *Viewmill*.

Acabamento - Utilizou-se a estratégia *step and shallow* para se proceder ao acabamento. Esta estratégia divide o modelo por zonas segundo a inclinação das superfícies. Neste caso escolheu-se um ângulo de 30° para distinguir zonas rasas de zonas inclinadas [7]. A ferramenta selecionada foi novamente a fresa esférica de 8 mm porque trocar de ferramenta entre percursos iria implicar uma nova calibração pois, devido ao facto da ferramenta ser estática, seria necessário substituí-la manualmente correndo o risco da localização da ponta da ferramenta não ficar exatamente no local onde a estratégia de desbaste assume que ela está. O percurso foi depois simulado no *viewmill*.

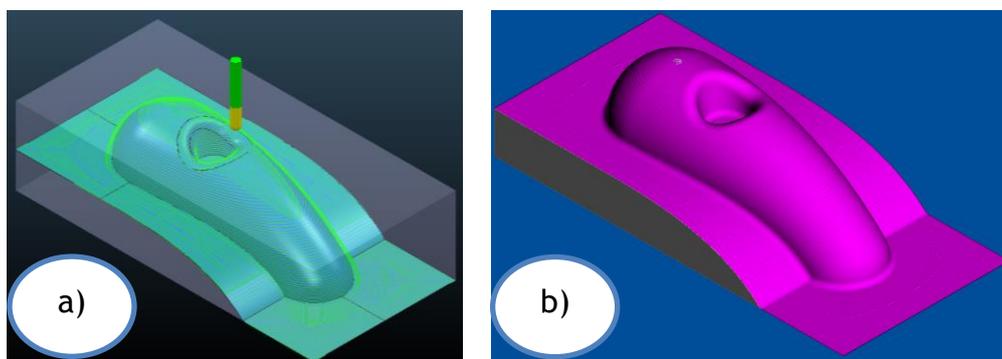


Figura 4-12 - Estratégia de acabamento. a) Percurso; b) *Viewmill*.

4.4.2. PRI

Uma vez concluído o projeto no *PowerMILL* é necessário simular os percursos mais uma vez no PRI. Procedeu-se então à importação do Robô. Foi necessário uma nova calibração de plano de ferramenta pois desta vez iria ser utilizada uma fresa em substituição da caneta. Efetuaram-se os mesmos passos de calibração pelo método TCP e alteração dos ficheiro CAD e do ficheiro *IRB2400-16_CINFU.mtd* como no projeto anterior (Vid 4.3.1 e 4.3.2). Daqui resultou uma nova célula virtual com o eixo de ferramenta localizado no centro do suporte do *spindle*.

Neste projeto decidiu-se que o modelo do *spindle* seria totalmente criado pelo *PowerMILL* através do menu de criação de ferramenta, adicionando hastes à ferramenta esférica.



Figura 4-13 -Nova configuração da célula. a) Célula virtual. Eixo de ferramenta (Azul) localizado no centro do suporte do *spindle*. b) Calibração do *spindle* com o auxílio da ferramenta ponteira.

Em seguida procedeu-se ao reposicionamento da peça para uma posição favorável à maquinação. Utilizou-se a posição inclinada a 90° em Z de forma a evitar singularidades. Esta posição foi calculada segundo um eixo criado na base da peça e movido 275 mm segundo o eixo Z e -90 mm segundo o eixo do X, com rotação de -90° em X e 90° em Z (usando o sexto eixo como referência).

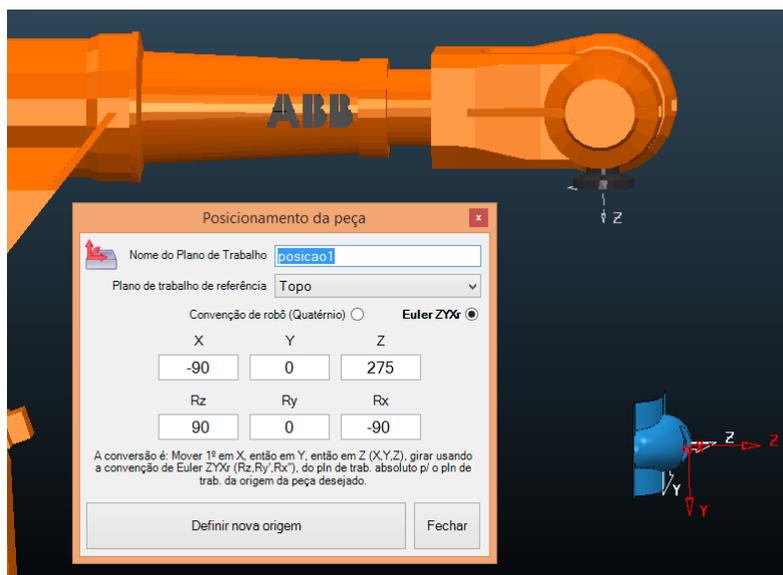


Figura 4-14 - Posicionamento da peça.

4.4.3. Simulação

Uma vez posicionada a peça procede-se à simulação dos dois percursos no PRI. Foi necessário rodar a peça em torno do eixo do Z da ferramenta de forma a obter uma posição inicial da qual não resultassem erros como, limites de eixo, singularidades ou colisões. Após várias tentativas e experimentando várias posições iniciais foi encontrada a posição mais favorável para cada uma das estratégias.

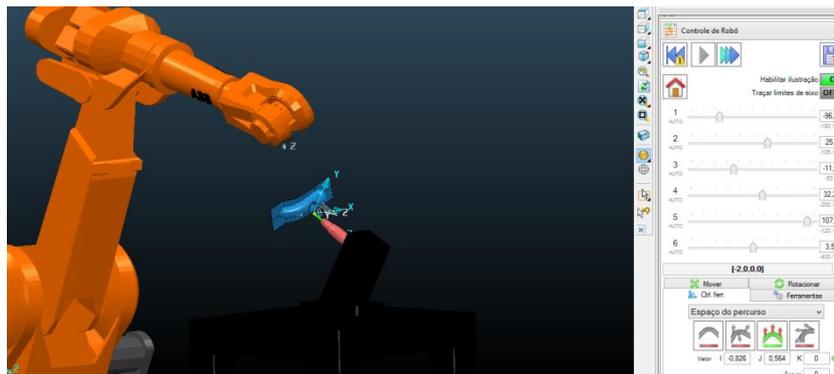


Figura 4-15 - Simulação dos percursos.

4.4.4. Pós-processamento

Para exportar os percursos para o robô é necessário proceder ao pós processamento. Cria-se um novo programa NC e adicionam-se ambos os percursos a este e, utilizando a ferramenta de cálculo de planos de trabalho, procede-se ao preenchimento da lista de parâmetros, nomeadamente calculando o plano de trabalho de ferramenta e o objeto de trabalho e copiando esses valores para as variáveis do ficheiro .MOD (Vid Figura A-7 dos Anexos) .

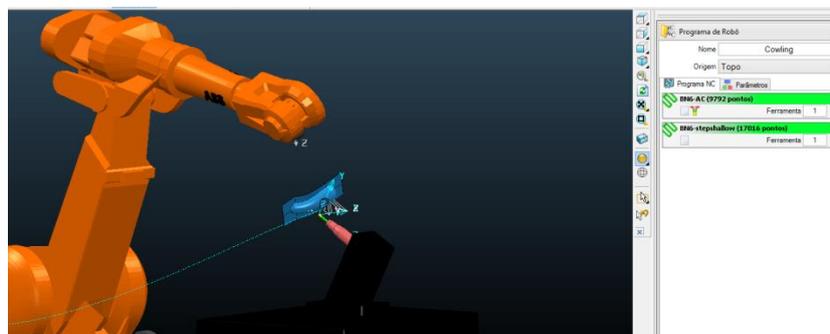


Figura 4-16 - Pós Processamento



Figura 4-17 - Cálculo de planos de trabalho do robô.

Por fim procede-se à exportação dos ficheiros do programa NC correspondentes ao desbaste, acabamento e a rotina de chamada. A rotina de chamada foi exportada com o nome “cowling”, que é o nome que foi dado ao programa NC mas, antes de a inserir no controlador do robô, foi alterado o nome do ficheiro para “MainModule” para que esta substituísse o modulo de chamada presente no robô. É no *MainModule* que estão declaradas as variáveis correspondentes ao plano de ferramenta e ao objeto de trabalho. De forma a este processo ser facilitado pode-se criar o programa NC já com o nome de *MainModule* mas pode implicar confusão entre ficheiro na hora de organizar os ficheiros no controlador do robô. Fica à decisão do utilizador utilizar ou não esta metodologia de trabalho.



Figura 4-18 - Ficheiros dos módulos criados pelo pós-processador.

4.4.5. Maquinagem no Robô

Para se proceder à simulação no robô foi necessário fabricar um bloco de trabalho com as mesmas dimensões do bloco da peça criado no *PowerMILL*. Procedeu-se ao corte de um bloco de poliuretano expandido na oficina com 210x100x55 mm. Este bloco foi montado no suporte de madeira utilizado no primeiro projeto na mesma posição que o bloco da simulação. Este é um passo bastante crítico pois caso este não esteja na posição correta a maquinagem não será bem executada e a peça será descartada por defeito.



Figura 4-19 - Bloco de Poliuretano expandido.

Neste fase do trabalho, e como era a primeira vez que se iria testar o *spindle* da célula, foi detetado um erro. Quando se arrancava com qualquer um dos programas de maquinagem o *spindle* não ligava e conseqüentemente a fresa não se movia. Após consulta de alguma bibliografia e de ficheiros MOD. de trabalhos anteriores de programação manual realizados pelo CINFU, verificou-se que era necessário acrescentar uma chamada de sinal nos ficheiros dos programas de maquinagem. O *spindle* está conectado numa banca com PLC e apenas se liga se for feita uma chamada de sinal específica feita pelo controlador. Para se continuar com o trabalho foi inserido manualmente com um editor de texto as linhas de código correspondentes ao sinal de ligar e desligar o *spindle* nos ficheiros dos percursos de maquinagem. As linhas de código inseridas foram as seguintes:

Set Q27P4ActivrRebrb; - Liga o *spindle*
Reset Q27P4ActivrRebrb; - Desliga o *spindle*

De forma a facilitar o trabalho, este código foi inserido no pós processador e vai ser incluído automaticamente em todos os programas exportados no futuro.

Os ficheiros dos módulos correspondentes ao projeto “cowling” foram carregados no controlador e o processo de maquinagem foi executado em modo automático com sucesso.

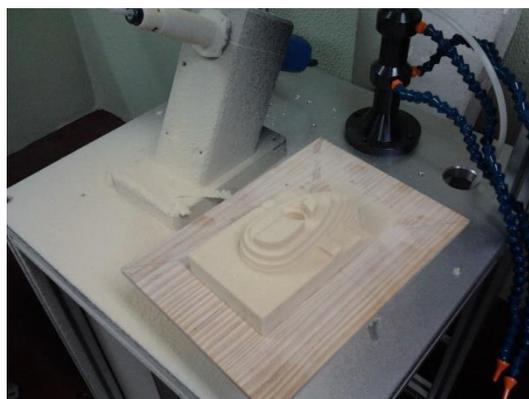


Figura 4-20 - Resultado do desbaste da peça “cowling” no robô.

4.5. Projeto de maquinagem a 5 eixos

De forma a simular e maquinar uma peça com estratégias de 5 eixos foi iniciado um novo projeto. Neste projeto procedeu-se também a uns ajustes finais na célula robótica de forma a que esta ficasse o mais otimizada possível na simulação e exportação de novos projetos. Isto permitiu simplificação e poupança de tempo na execução deste e de futuros projetos de maquinagem.

4.5.1. Otimização da célula

Calibração de *Spindle*

Para otimizar a célula foi necessário proceder a uma calibração de *spindle* utilizando a ferramenta de calibração do PRI.

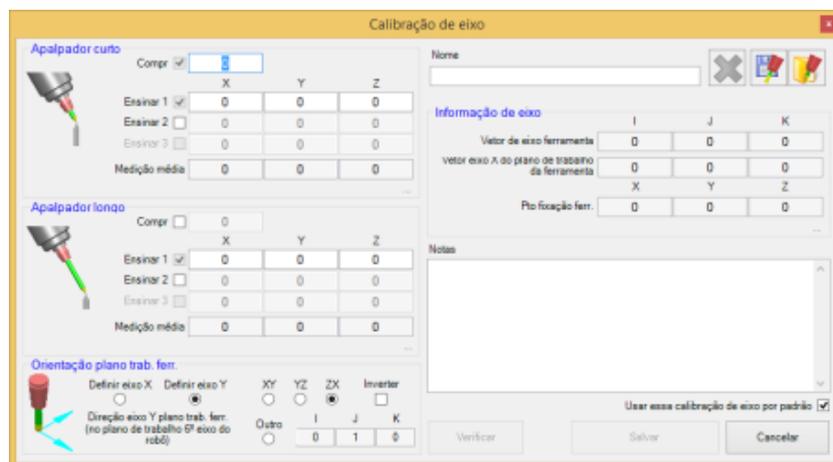


Figura 4-21 - Menu de calibração de *spindle*

Como foi descrito anteriormente, o robô possui 3 métodos para criar calibrações de ferramenta: TCP, TCP Z e TCP XZ. Este métodos utilizam entre 4 e 9 pontos para calcular a média das coordenadas X, Y e Z correspondentes à ponta da ferramenta. O problema que surgiu é que nenhum destes métodos pode ser diretamente usado com a ferramenta do PRI pois esta necessita de 3 médias para duas ponteiros de calibração diferentes. Devido a esta limitação do controlador do robô foi necessário fazer uma adaptação entre os dois métodos.

4.5.2. Medições no robô:

Para iniciar este processo de calibração, foi necessário fabricar no torno do CINFU duas ponteiros para servirem de referência.



Figura 4-22 - Duas ponteiros de calibração fabricadas no torno CNC e uma fresa esférica de 6mm de diâmetro.

Em seguida, montou-se a ferramenta ponteiro no robô, que estava calibrada segundo o sexto eixo. A ponta desta ferramenta irá ser o ponto de referência para medir as ponteiros agora fabricadas, fazendo *jogging* ao robô em torno destas.

O procedimento é o mesmo para as duas ponteiros, curta e longa. Após ter a ponteiro do robô acoplada neste, coloca-se a ponteiro curta no lugar da ferramenta, no *spindle*. Aproximam-se as duas pontas das duas ponteiros até estas estarem praticamente em contacto uma com a outra. No controlador, abre-se uma nova calibração de *spindle* do tipo TCP de 5 pontos e com a variável *robhold* definida como *FALSE*. Tendo a primeira posição preparada faz-se “ensinar” para gravar a coordenada da posição atual no ponto 1. De seguida, faz-se *jogging* do robô para outra posição e aproximam-se as pontas das ferramentas novamente até estas estarem praticamente em contacto. Tendo a nova posição pronta, faz-se novamente “ensinar” para gravar as coordenadas do ponto 2. Repete-se este processo de *jogging* e de gravação para os 3 pontos restantes. Estando os 5 pontos preenchidos faz-se “ok” e o controlador calcula o ponto médio. Este ponto médio agora calculado é apenas o “Ensaio 1” na tabela de calibração do PRI. Portanto, é necessário criar mais 2 calibrações no robô para a ponteiro curta, cada uma delas com 5 pontos e posteriormente criar mais 3 calibrações, também de 5 pontos cada

para a ponta longa. No final é também necessário medir o comprimento das ponteiros bem como o comprimento da ferramenta montada no *spindle* e inserir esses valores na tabela de calibração do PRI.



Figura 4-23 - Calibração da ferramenta e medição dos comprimentos das ponteiros.



Com as medições efetuadas obteve-se então a tabela de calibração do PRI completamente preenchida.

Calibração de eixo

Apalpador curto

Compr 13

	X	Y	Z
Ensinar 1 <input checked="" type="checkbox"/>	65.5121	-1124.03	882.667
Ensinar 2 <input checked="" type="checkbox"/>	64.9254	-1124.04	883.137
Ensinar 3 <input checked="" type="checkbox"/>	64.657	-1124.59	883.107
Medição média	65,031	-1124,22	882,97

Erro médio (mm): 0.47

Apalpador longo

Compr 68.9

	X	Y	Z
Ensinar 1 <input checked="" type="checkbox"/>	66.223	-1075.22	910.771
Ensinar 2 <input checked="" type="checkbox"/>	65.3883	-1076.32	911.023
Ensinar 3 <input checked="" type="checkbox"/>	65.7778	-1076.13	911.169
Medição média	65,796	-1075,89	910,988

Erro médio (mm): 0.573

Orientação plano trab. ferr.

Definir eixo X Definir eixo Y XY YZ ZX Inverter

Direção eixo Y plano trab. ferr. (no plano de trabalho global do robô)

Outro I J K

I: 0, J: 0, K: 1

Nome: SpindleCINFU

Informação de eixo

	I	J	K
Vetor de eixo ferramenta	-0.01369	-0.865059	-0.501483
vetor eixo X do plano de trabalho da ferramenta	0.999875	-0.015824	0
	X	Y	Z
Pto fixação ferr.	64,853	-1135,479	876,443

Erro médio (mm): 0.016

Notas

Usar essa calibração de eixo por padrão

Verificar Salvar Cancelar

Figura 4-24 - Preenchimento da tabela de calibração de ferramentas com os dados recolhidos.

Ao clicar no botão “Verificar” o PRI calcula o ponto de fixação da ferramenta e o vetor do eixo da ferramenta. Estes valores são depois utilizados no modelo CAD para reposicionar o ponto de fixação e o eixo da ferramenta e permitenos desenhar o *spindle* na posição correspondente à realidade da célula. O botão “Verificar” cria também dentro do *PowerMILL* um plano de trabalho localizado no ponto de fixação da ferramenta. Infelizmente foi detetado que este plano de trabalho era incorretamente calculado e criado pelo *software* pois, embora as coordenadas da ponta da ferramenta e a orientação estivessem corretas, este vetor era desenhado em função do sistema de coordenadas do sexto eixo quando deveria ser criado em função do sistema de coordenadas da base do robô, o referencial World, ou seja, não assumia que o *spindle* era fixo e calculava-o assumindo que este estava acoplado no sexto eixo do robô. Isto levou a um contacto com o engenheiro Alexandre da DELCAM que por sua vez entrou em contacto com os programadores do *PowerMILL*. Fomos então informados que era um erro que já tinha sido detetado e que uma correção iria acontecer na versão seguinte do programa.

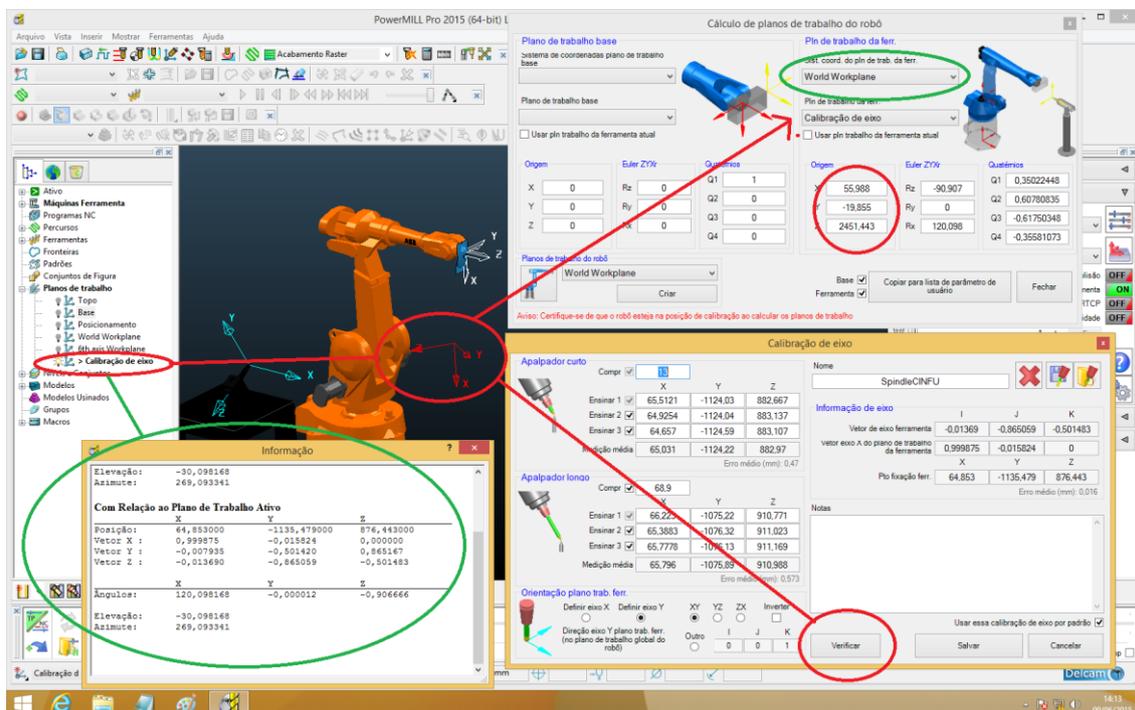


Figura 4-25 - Detecção do erro de criação do eixo de ferramenta.

Para inserir esta nova calibração de *spindle* no robô é necessário novamente alterar o ficheiro *IRB2400-16_CINFU.mtd*.

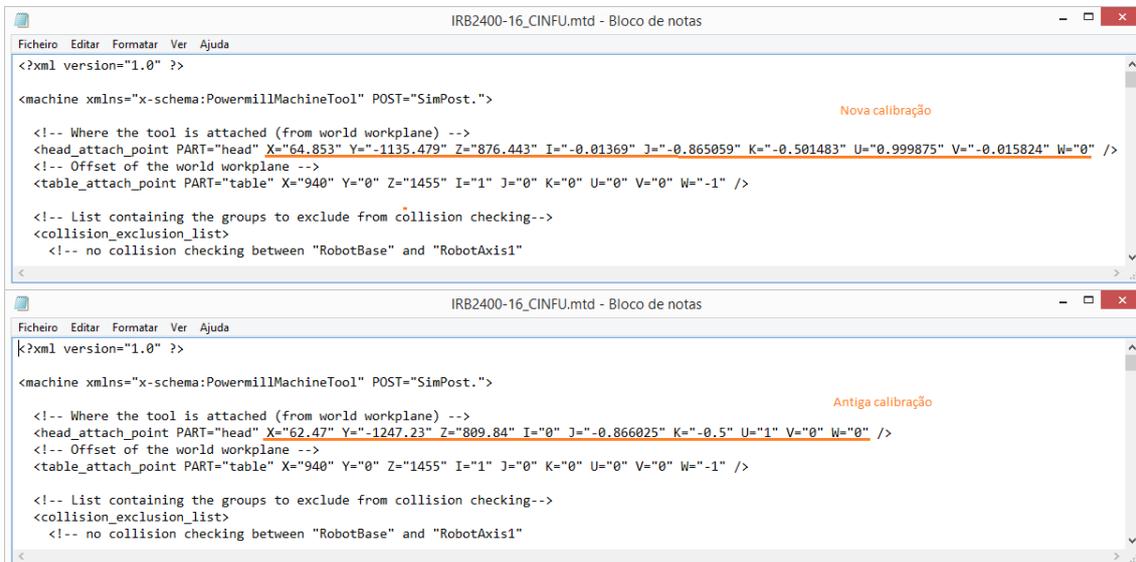


Figura 4-26 - Edição da calibração do plano de ferramenta no ficheiro *IRB2400-16_CINFU.mtd*.

4.5.3. Modelos CAD e ficheiros .dmt

Visando a otimização da célula robótica foi criado o modelo CAD da garra de dois dedos utilizada para segurar o suporte das peças. Como de momento não existe nenhum tipo de garra ou prensa disponível no CINFU específica para a função de pegar em peças para maquinaria, foi necessário continuar a trabalhar com a garra 2D. No caso de no futuro o CINFU vir a adquirir este tipo de equipamento, estão descritos nesta dissertação os procedimentos a executar de forma a incorpora-la na célula robótica e assim esta passar a fazer parte da célula virtual do PRI. A garra de dois dedos foi criada em *PowerSHAPE* e exportada sob a forma de um ficheiro .dmt e colocada na mesma pasta que os restantes ficheiros .dmt que compõe o robô.

O ficheiro *IRB2400-16_CINFU.mtd* foi também modificado de forma a que o PRI reconheça a adição da nova parte do robô:

```
<machine_part NAME="RobotAxis6">
  <axis>
    <control_info ADDRESS="F" MIN="-400" MAX="400" PRIORITY="AUTO"
HOME="0" VALUE="0"/>
    <simple_rotary X="0.0" Y="0.0" Z="1455" I="1" J="0" K="0" />
  </axis>
  <model_list>
    <dmf_file>
      <path FILE="ABB/F.dmf" />
      <rgb R="50" G="50" B="50" />
    </dmf_file>
    <dmf_file>
      <path FILE="ABB/GARRA.dmf" />
      <rgb R="80" G="80" B="80" />
    </dmf_file>
  </model_list>
</machine_part>
```

O ficheiro é adicionado ao sexto eixo (*RobotAxis6*) de forma a acompanhar os movimentos do componente **F**.

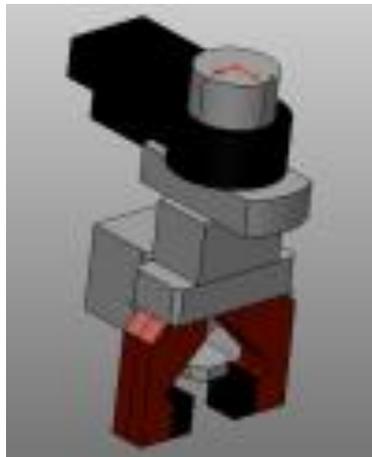


Figura 4-27 - Modelo CAD da garra de dois dedos.

Outra alteração efetuada na célula foi no *spindle*. Ao contrário do projeto anterior em que o suporte da ferramenta foi construído no *PowerMILL*, para este projeto, para além da banca e do suporte do *spindle*, desenhou-se em *PowerSHAPE* também o *spindle*.

Isto permite que a criação de ferramentas seja mais rápida visto o *spindle* já fazer parte da célula robótica.

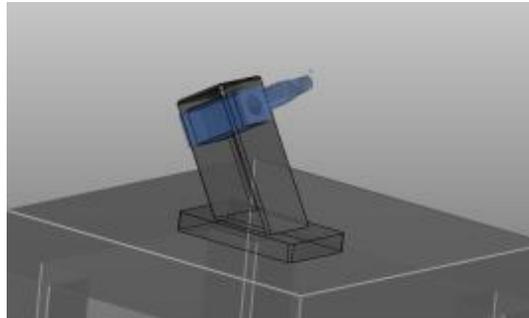


Figura 4-28 - Novo modelo CAD da banca e do *spindle*.

Com estas alterações ficou assim completa a célula robótica no PRI, melhorando as simulações pois irá a partir de agora ter em consideração possíveis colisões entre a garra e os restantes elementos da célula e mais facilmente simular os percursos de maquinagem gerados.

4.5.4. Peça para maquinagem a 5 eixos

Após ter a célula otimizada com os novos elementos foi iniciado um novo projeto de maquinagem. Este projeto tinha como objetivo simular um percurso de maquinagem a 5 eixos e exporta-lo para ser maquinado pelo robô. A peça utilizada está representada na Figura 4-29.

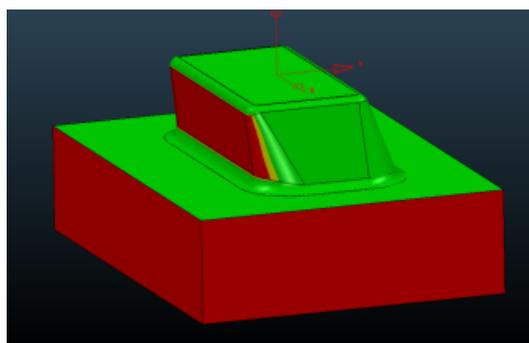


Figura 4-29 - Ângulos de saída da peça.

Como se pode ver pela figura, uma das paredes não possui ângulo de saída, não sendo possível maquina-la com estratégias de 3 eixos. Vai ser necessária uma estratégia de 5 eixos de forma a fazer o acabamento desta peça.

Após importar a peça para o *PowerMILL*, procedeu-se à criação de um plano de trabalho no topo do modelo, bem como a criação do Bloco definido por

caixa e utilizando o plano de trabalho criado como referência. Foram também definidas as alturas de movimentos rápidos e o ponto inicial e final foram alterados para Centro do Bloco seguro [8].

Estratégia de desbaste - A estratégia de desbaste utilizada foi um desbaste de modelo, com corte concordante, tolerância de 0,1 mm, sobresspessura de 0,5 mm, passo vertical de 8,0 mm e passo lateral de 2,0 mm. A estratégia foi calculada e simulada em *viewmill* [8].

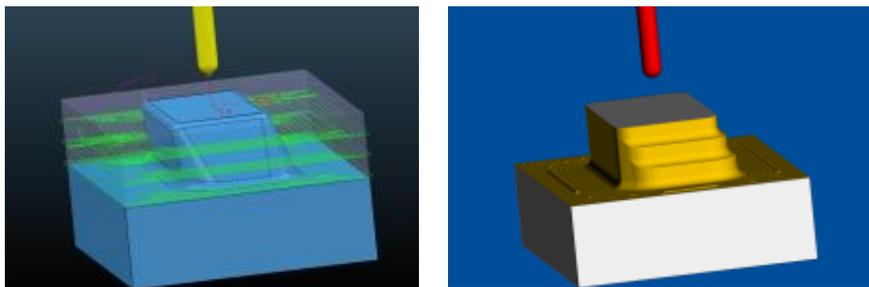


Figura 4-30 - Desbaste: a) Percurso; b) Viewmill.

Estratégia de acabamento - Utilizou-se uma estratégia de acabamento espiral paramétrica com tolerância de 0,01 mm, sobresspessura de 0 mm e um passo lateral máximo de 0,2 mm. Esta estratégia foi também simulada em *viewmill*. Neste tipo de estratégia a inclinação da ferramenta é definida como automática [8].

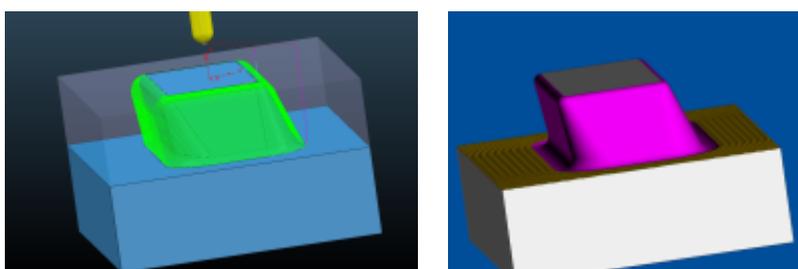


Figura 4-31 - Acabamento paramétrico: a) Percurso; b) Viewmill.

Para fazer o acabamento da zona plana fez-se uma segunda estratégia de acabamento do tipo *offset* plano, com tolerância de 0,1 mm, sobresspessura de 0 mm e passo lateral de 0,5 mm. Neste tipo de estratégia seria mais indicado utilizar uma fresa de topo mas isso iria implicar a troca de ferramenta.

Optou-se por não fazer troca de ferramenta pois de momento não existia nenhuma fresa de topo no CINFU capaz de ser adaptada no *spindle* da célula robótica e a troca de ferramenta iria implicar nova calibração [8].

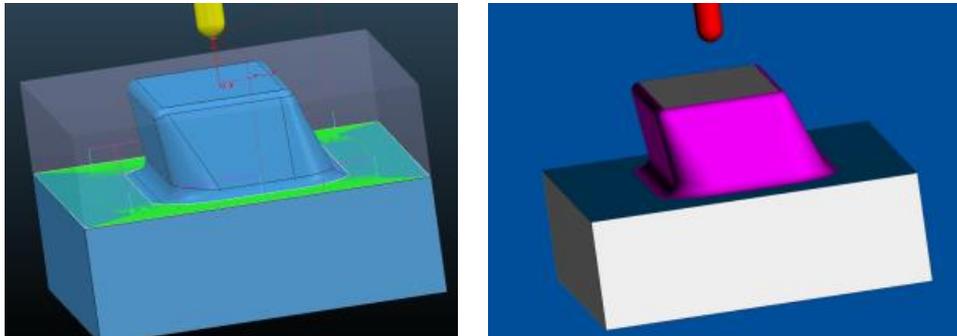


Figura 4-32 - Acabamento raso: a) Percurso; b) Viewmill.

PRI - Após a simulação de ambas as estratégias no *viewmill*, é necessário agora simula-las no PRI. Procedeu-se então à importação do novo robô com as alterações efetuadas, nomeadamente o novo *spindle* e a garra de dois dedos.



Figura 4-33 - Nova célula do robô ABB do CINFU.

De seguida é necessário posicionar a peça para a maquinação. É utilizada o mesmo método e posição do projeto de 3 eixos do capítulo 4.4.2. Assim, é criado um plano de trabalho na base do modelo e este é utilizado como referência para movimentar a peça para a posição ideal à maquinação. Move-se então a peça 275 em Z, -90 em X e efetua-se uma rotação em Z de 90° e em X de -90°.

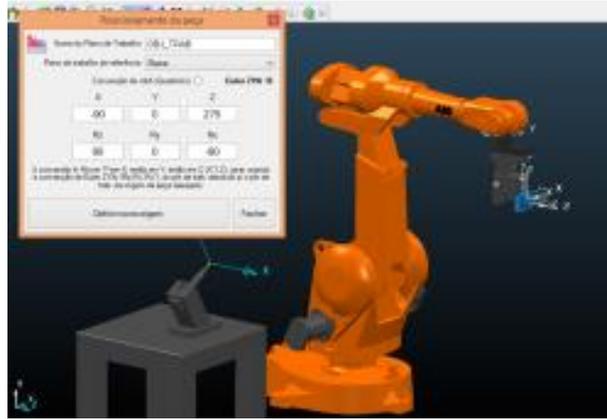


Figura 4-34 - Posicionamento da peça.

Procedeu-se de seguida à simulação dos percursos, fazendo *jogging* ao robô de forma a obter uma posição inicial que resulta-se numa simulação sem erros.

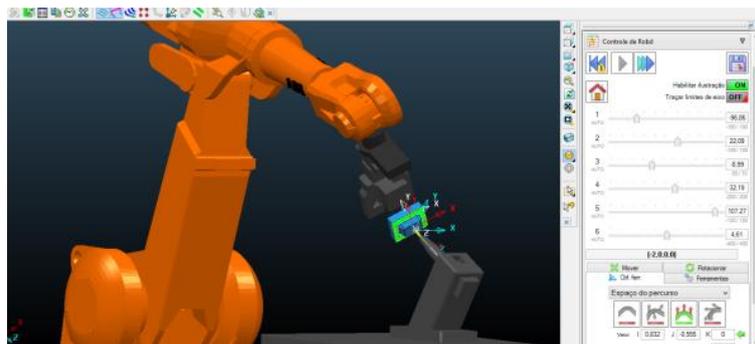


Figura 4-35 - Simulação dos percursos.

Uma vez obtidas as simulações sem erros é necessário criar um programa NC para exportação dos ficheiros MOD. Copiam-se os dados do plano de ferramenta e do objeto de trabalho através da ferramenta de cálculo de planos de trabalho do robô, seleciona-se a calibração de ferramenta feita no passo anterior e importam-se os percursos para o programa NC.

A maquinação foi executada em modo automático na célula do robô com sucesso e foi possível maquinar uma peça com de estratégias de 5 eixos utilizando o robô do CINFU.

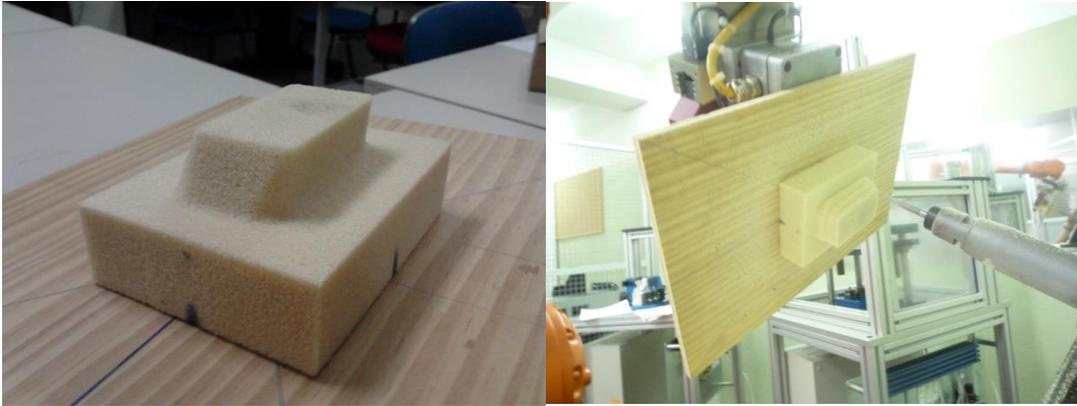


Figura 4-36 - Maquinação do bloco de poliuretano na forma final.

Conclusões

Embora tenha sido necessário ultrapassar inúmeros obstáculos, desde o relacionamento interpessoal até aos problemas técnicos inerentes à configuração da célula robô para esta nova aplicação, os resultados abrem as portas a novas aplicações.

Foi assim alcançado o objetivo de utilizar esta célula robótica para maquinagem, mas também para uma utilização genérica em programação *offline*: uma vez que o *PowerMILL Robot Interface* se encontra corretamente configurado e existindo agora os conhecimentos necessários, passa a ser possível criar trajetórias em ambiente CAM com completa simulação das trajetórias de fresagem, furação ou corte.

Estão assim criadas as condições para desenvolver a aplicação de rebarbagem de peças de fundição. Aqui os desafios prender-se-ão por exemplo com o suporte da peça, a programação de estratégias avançadas de 5 eixos e naturalmente o problema inerente às vibrações.

A nível pessoal, este estágio constituiu não só um desafio, mas sim a entrada numa nova área tecnológica de elevado potencial. A aprendizagem realizada e os conhecimentos adquiridos permitem encarar potenciais saídas profissionais nesta área.

Referências Bibliográficas

- [1] CINFU. (2010, 09-04-2015). *Bem-vindo ao Cinfu*. Available:
<http://www.cinfu.pt/index.php/component/content/article/3-breves/89-bem-vindo-ao-cinfu>
- [2] CINFU. (2010, 09-04-2015). *Formação Profissional*. Available:
<http://www.cinfu.pt/index.php/component/content/article/3-breves/3-forma-profissional>
- [3] NORCAM - *Engenharia e Design Industrial*. (2015, 09-04-2015). Available:
<http://www.norcam.pt/home/a-norcam/>
- [4] ABB, *Product manual - IRB 2400*. Västerås, Sweden: ABB AB - Robotics Products, 2013
- [5] ABB, *Operating manual - IRC5 with FlexPendant*. Västerås, Sweden: ABB AB - Robotics Products, 2013.
- [6] Delcam, "*PowerMILL Robot - Training*," [Apresentação]. Birmingham, UK: Delcam, 2012.
- [7] Delcam, "*PowerMILL - Training Course*," [Manual]. Birmingham, UK: Delcam, 2012.
- [8] Delcam, "*PowerMILL 5-axis - Training Course*," [Manual]. Birmingham, UK: Delcam, 2012.

A. Anexos



Figura A-1 - Elementos constituintes do controlador da célula robótica: a) *Flexpendant*; b) Painel de controlo; c) Botões de acionamento da célula, ligação dos motores, seleção do modo de funcionamento e interrupção de emergência.

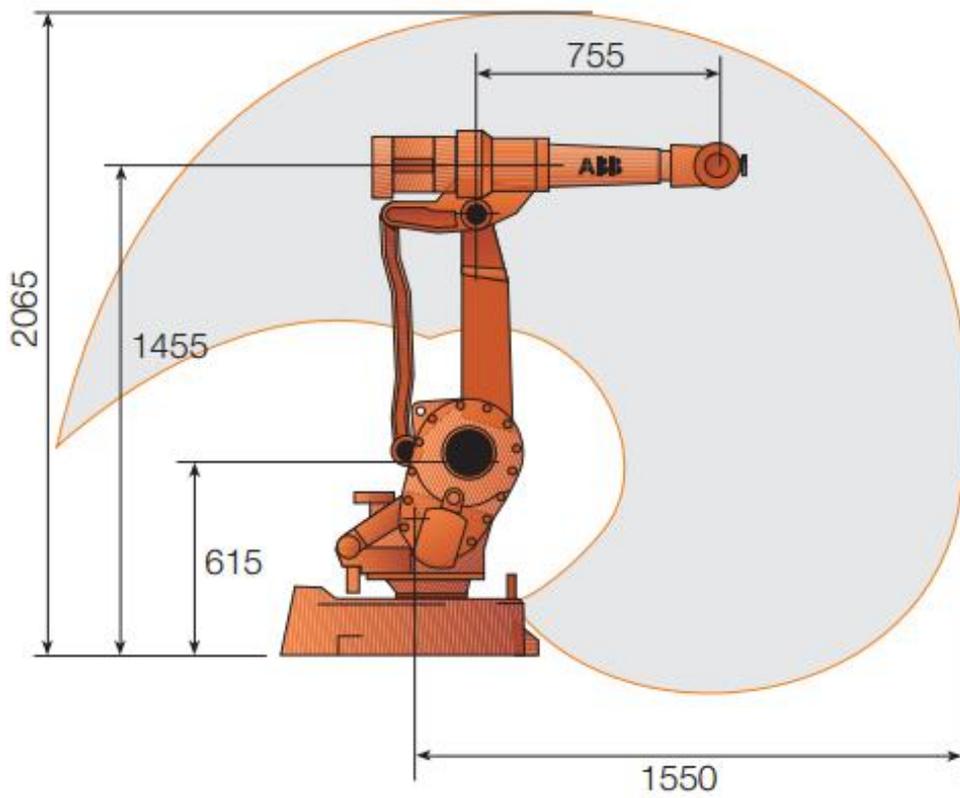


Figura A-2- Alcance do robô ABB IRB 2400-16.

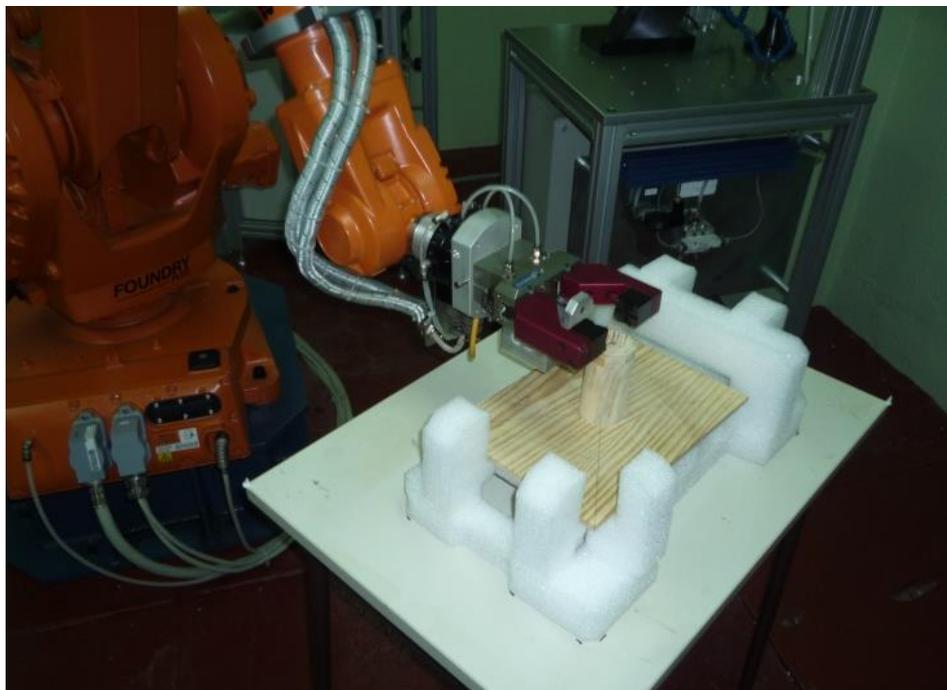


Figura A-3 -Robô do CINFU a executar uma rotina de trabalho criada durante a formação.

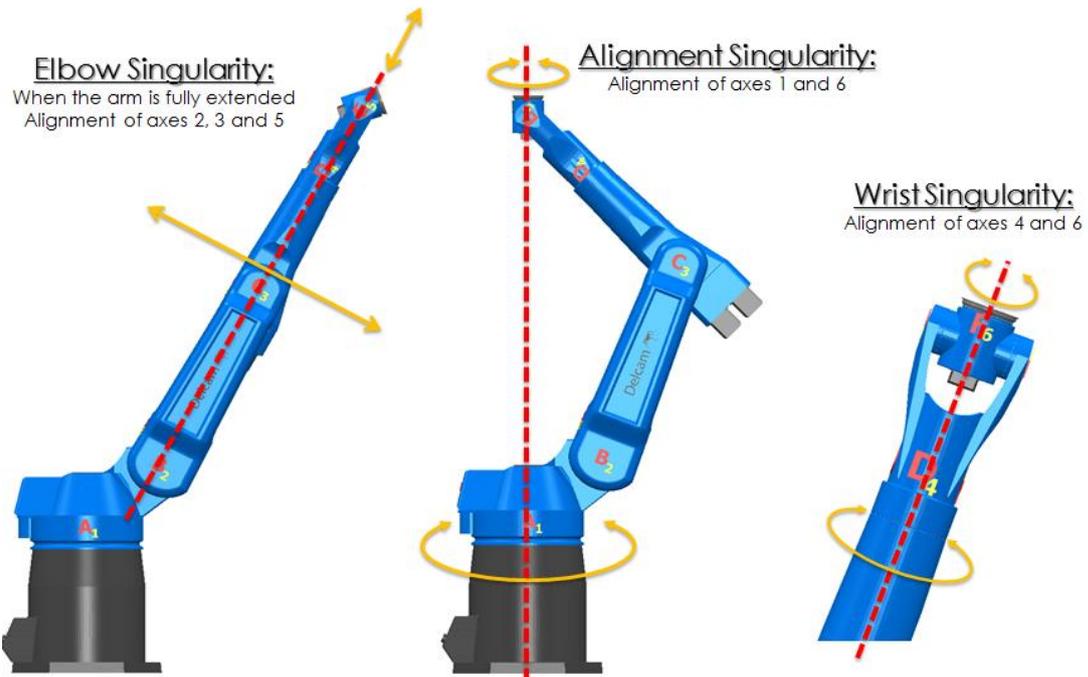


Figura A-4 - Tipos de singularidades

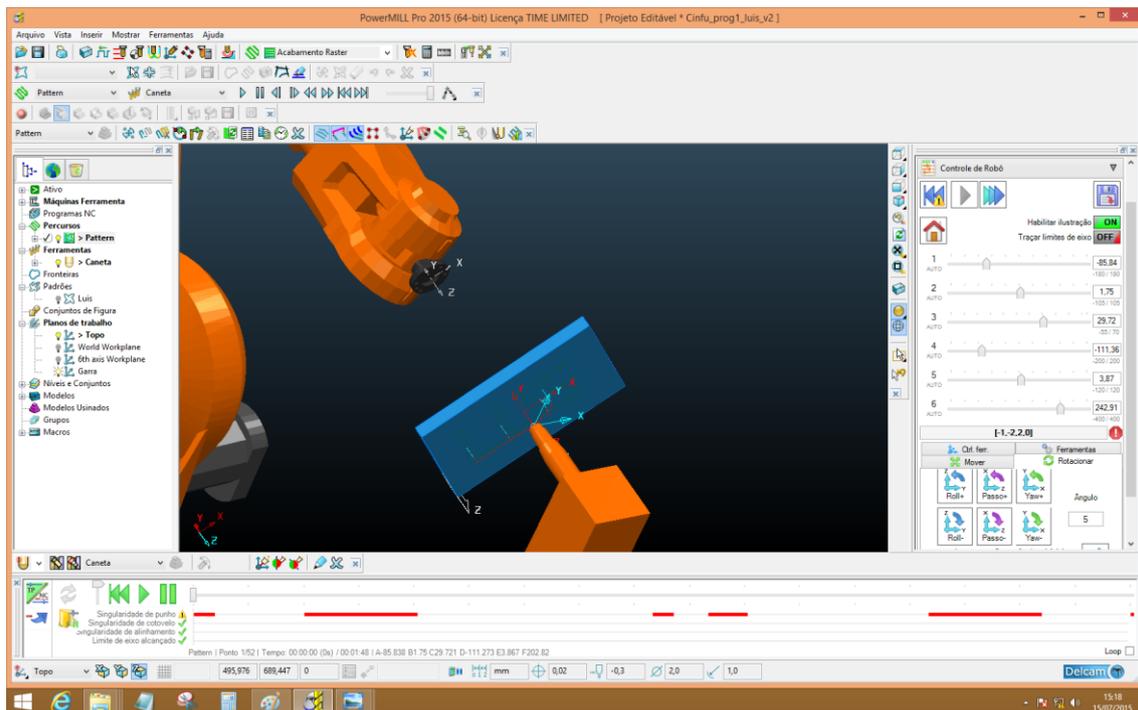


Figura A-5 - Erro de singularidade do primeiro projeto.

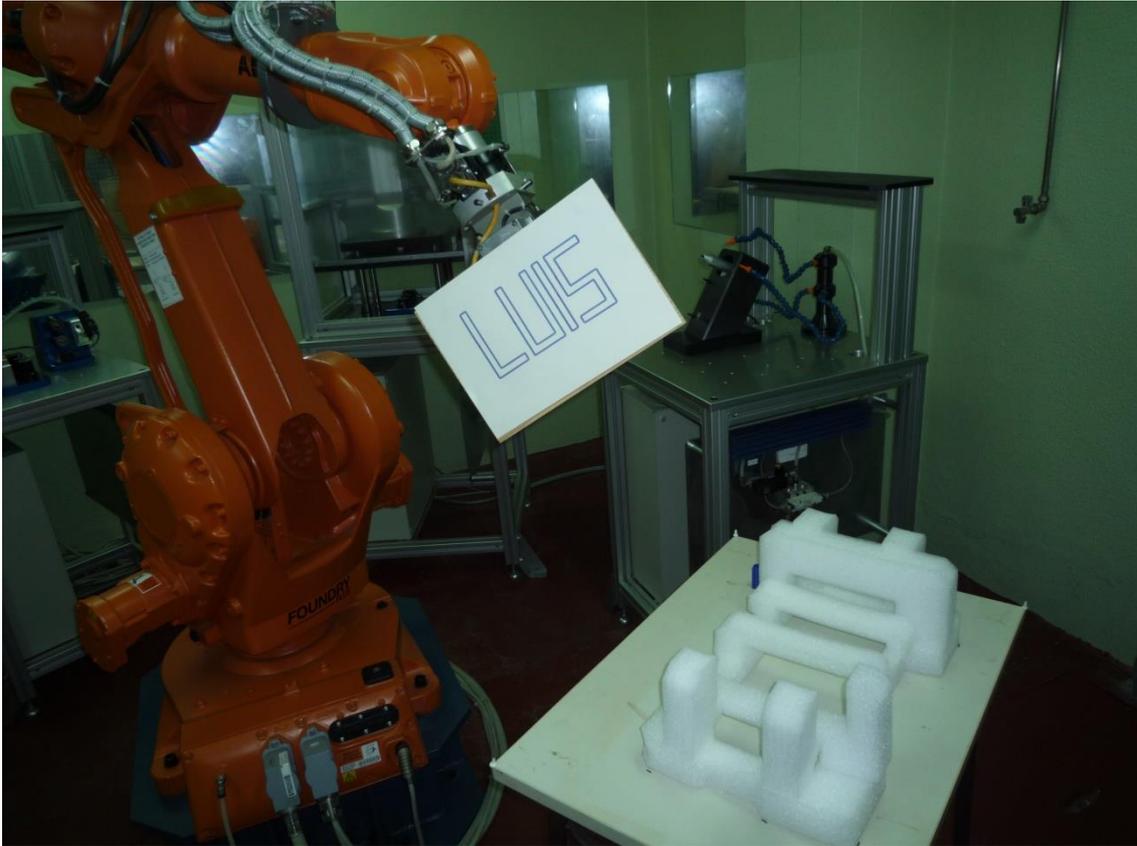


Figura A-6 - Resultado do primeiro projeto. Após desenhar o padrão o robô coloca a peça num suporte posicionado numa mesa. Rotina criada com programação manual.

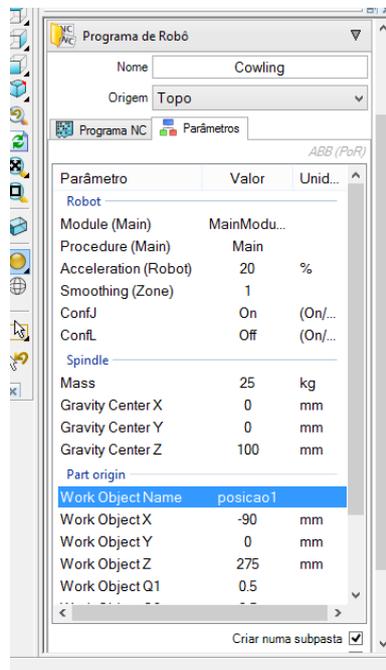


Figura A-7 -Lista de parâmetros do pós processador no projeto *cowling*.