

Programação de robôs industriais em operações de maquinagem

João Pedro Moreira Dias Antunes

Dissertação do MIEM

Orientador: Professor Paulo Augusto Ferreira de Abreu

Coorientador: Professor Manuel Romano dos Santos Pinto Barbosa



Mestrado Integrado em Engenharia Mecânica

Ramo de Automação

Porto, Setembro de 2015

*“...Há sempre, na nossa infância,
um momento em que a porta se
abre e deixa entrar o futuro.”*

Graham Greene

Resumo

Esta dissertação aborda o problema da programação de robôs industriais em operações de maquinagem. As alternativas de *software* utilizadas para o presente estudo foram o Mastercam/RobotMaster, Sprutcam e o RobotStudio/Machining PowerPac. O objetivo principal desta dissertação foi comparar e aferir as implicações da utilização destas alternativas para gerar o programa para o robô no código RAPID, linguagem da ABB para os seus robôs industriais.

O processo de maquinagem abordado no presente trabalho é a fresagem. O trabalho começa por fazer uma apresentação das três soluções de programação escolhidas de modo a ilustrar as suas características principais. Seguidamente definiram-se diferentes peças e estratégias de maquinagem para se avaliar os processos de definição do plano de maquinagem e obtenção do código na linguagem RAPID. Os critérios de avaliação incluíram aspetos mais objetivos e quantificáveis como por exemplo: o número de linhas de código e tipo de instruções de movimento, potência, energia consumida e perfis de velocidade. E ainda critérios mais subjetivos como por exemplo as funcionalidades disponibilizadas por cada sistema e a facilidade de utilização. Para além da comparação ao nível da simulação e programação *off-line*, foram também efetuadas experiências com o robô real para validar os programas realizados.

Conclui-se com esta dissertação que a utilização de *softwares* CAM para criação de operações de maquinagem para robôs industriais ainda acarretam alguns problemas, sendo recomendado sempre a validação dos programas gerados no *software* proprietário do robô. Contudo é espectável que continue a haver avanços nesta área, devido às vantagens que os robôs proporcionam não só pela disponibilidade de um elevado volume de trabalho e flexibilidade, mas também a nível económico quando comparados a máquinas-ferramenta na maquinagem de peças que não requerem elevada precisão dimensional.

Programming of industrial robots for machining operations

Abstract

This dissertation focused on the use of CAM systems for programming industrial robots, particularly the ABB's IRB 2400 robot, for machining operations. The software alternatives used for this study were Mastercam with the add-on RobotMaster, Sprutcam and RobotStudio with the add-on Machining Powerpac. The main objective of this study was to compare and evaluate the implications of applying these alternatives to generate the program for the robot in Rapid code, ABB's language to their industrial robots.

The machining process approached in this study is milling. The thesis starts with a presentation of the three programming solutions chosen to illustrate its main characteristics. Thereafter, the different parts and machining strategies to evaluate the processes of defining the machining plan and obtaining the code in Rapid language were chosen. The evaluation criteria included more objective and quantifiable aspects such as: the number of code lines and type of movement instructions, power, energy consumption and speed profiles. In addition subjective criteria were used, such as the functionality provided by each software and their simplicity of use. Along with the tests in simulation and offline programming, experiments were also performed with the real robot to validate the programs implemented.

In conclusion the use of CAM software to create machining programs for industrial robots still carry some problems, being highly recommend the validation of programs generated in the robot proprietary software. However it is expected that continue to be advances in this area, due to the advantages that robots provide not only for their availability of a high workspace and flexibility, but also economically when compared to machine tools in machining parts that do not require high dimensional accuracy.

Agradecimentos

Gostaria de agradecer especialmente aos meus orientadores, os Professores Paulo Abreu e Manuel Romano Barbosa, pela enorme paciência, apoio e disponibilidade ao longo deste projeto.

Ao Professor Francisco Teixeira de Freitas pela constante dedicação em proporcionar aos alunos cada vez mais e melhores condições de aprendizagem.

Aos meus colegas e Amigos de curso Renato Ferreira, Daniel Queirós e Afonso Urbano, assim como ao Sr. Joaquim, pelas críticas construtivas ao longo deste trabalho e pelos momentos decididamente inesquecíveis.

À minha família, por sempre me terem dado condições e apoio durante todo o meu percurso escolar. Um muito obrigado.

Beatriz,

Obrigado.

Índice

1	Introdução.....	1
1.1	Objetivos do trabalho.....	1
1.2	Organização da dissertação.....	2
2	Maquinagem com robôs industriais.....	3
2.1	Breve introdução à robótica industrial.....	3
2.2	Principais componentes de um robô industrial.....	6
2.3	Maquinagem com robôs.....	7
2.4	Operações de fresagem.....	9
3	Programação de robôs industriais.....	13
3.1	Programação <i>on-line</i>	13
3.2	Programação <i>off-line</i>	14
3.3	Programação de robôs industriais para operações de maquinagem.....	15
3.4	Sprutcam 9.....	16
3.5	Mastercam X5/Robotmaster.....	17
3.6	RobotStudio/Machining PowerPac.....	18
3.7	Análise comparativa.....	19
4	Casos de estudo de programação de robôs para operações de maquinagem.....	21
4.1	Modelação da célula robótica.....	22
4.2	Análise preliminar de estratégias de maquinagem para operação de facejamento....	25
4.3	Peças desenvolvidas.....	31
4.4	Programação com Sprutcam.....	33
4.5	Programação com Mastercam/Robotmaster.....	41
4.6	Programação com RobotStudio/Machining PowerPac.....	46
4.7	Validação e análise dos resultados obtidos.....	50
5	Maquinagem com célula robótica real.....	53
5.1	Descrição da célula.....	53
5.2	Setup das operações de maquinagem.....	56
6	Conclusões e trabalhos futuros.....	61
	Referências.....	65
	Anexos.....	69

Índice de Figuras

Figura 2.1 - Número estimado anual de robôs industriais vendidos [6].....	4
Figura 2.2 - Número estimado anual de robôs industriais fornecidos por indústria [6]	4
Figura 2.3 - Fanuc R-2000iB numa operação de furação [7]	5
Figura 2.4 - Rebarbagem de um motor diesel utilizando robôs [8].....	5
Figura 2.5 - Componentes de um robô industrial articulado	6
Figura 2.6 - a) ABB IRB 6660 e b) Fanuc F-200Ib.....	8
Figura 2.7 - a) Kuka KR 500 R2830 MT e b) Stäubli RX 170 hsm.....	9
Figura 2.8 - Fresagem cilíndrica (a) e fresagem de topo (b)	9
Figura 2.9 - Fresagem “a empurrar” (a) e fresagem “a puxar” (b).....	10
Figura 2.10 - Parâmetros de corte (fresagem de topo)	11
Figura 3.1 - <i>Manual leadtrough</i> e <i>teach-pendant</i>	13
Figura 3.2 - Conversão de um programa de maquinagem.....	15
Figura 3.3 - Interface Sprutcam.....	17
Figura 3.4 - Interface Mastercam	18
Figura 3.5 - Interface RobotStudio/Machining PowerPac.....	18
Figura 4.1 - Esquema da estratégia adotada	21
Figura 4.2 - “Ferramenta” a montar no elemento terminal do robô	22
Figura 4.3 - <i>Tool0</i> (esquerda) e TCP segundo o referencial <i>tool0</i> (direita); em ambiente RobotStudio.....	23
Figura 4.4 - Representação do robô em ambiente SolidWorks	24
Figura 4.5 - <i>IRB 2400</i>	25
Figura 4.6 - Peça 0.....	25
Figura 4.7 - Exemplo de movimentação.....	27
Figura 4.8 - Trajetória gerada pelo Sprutcam e Mastercam	28
Figura 4.9 - Velocidade média e energia consumida nos percursos de maquinagem	30
Figura 4.10 - Percorso de maquinagem resultante da estratégia adaptativa.....	31
Figura 4.11 - Peça 1	31
Figura 4.12 - Peça 2.....	32
Figura 4.13 - Peça 3.....	32
Figura 4.14 - <i>Interface</i> após importação do modelo.....	34
Figura 4.15 - Posicionamento do <i>Workobject</i> em relação ao robô (Sprutcam).....	34
Figura 4.16 - Seleção da máquina a utilizar (Sprutcam)	35

Figura 4.17 - Geometria do bloco de trabalho (Sprutcam).....	35
Figura 4.18 - Parametrização da ferramenta (Sprutcam).....	36
Figura 4.19 - <i>Roughing Waterline</i> (Sprutcam)	36
Figura 4.20 - Erro na criação da trajetória (Sprutcam).....	37
Figura 4.21 - Correção do valor da coordena Z (Sprutcam).....	37
Figura 4.22 - Simulação do primeiro modelo (Sprutcam).....	38
Figura 4.23 - <i>Roughing Waterline</i> e <i>2D Contouring</i>	39
Figura 4.24 - <i>Finishing Plane</i>	39
Figura 4.25 - <i>Roughing Drive</i> (Sprutcam).....	39
Figura 4.26 - Seleção da orientação da ferramenta (Sprutcam)	40
Figura 4.27 - Colisão entre o <i>Spindle</i> e o robô	40
Figura 4.28 - Opções de posicionamento da peça (Mastercam).....	41
Figura 4.29 - Seleção do robô (Mastercam)	42
Figura 4.30 - Parametrização da ferramenta (Mastercam)	42
Figura 4.31 - Seleção da operação de desbaste (Mastercam)	43
Figura 4.32 - Seleção do robô nas definições do Robotmaster (Mastercam + Robotmaster) ..	43
Figura 4.33 - Funcionalidade <i>Workspace</i> (Mastercam/Robotmaster)	44
Figura 4.34 - <i>Optimization</i> (Mastercam/Robotmaster)	44
Figura 4.35 - Modificação da trajetória da ferramenta (Mastercam/Robotmaster).....	45
Figura 4.36 - Trajetória gerada (Mastercam).....	45
Figura 4.37 - Seleção e configuração da operação <i>Multiaxis</i> (Mastercam).....	45
Figura 4.38 - Célula robótica virtual (RobotStudio).....	46
Figura 4.39 - Posicionamento da peça (RobotStudio).....	47
Figura 4.40 - Seleção da face a maquinar (RobotStudio e Machining PowerPac).....	47
Figura 4.41 - Parametrização da operação de maquinagem e da ferramenta	48
Figura 4.42 - Parametrização da trajetória (RobotStudio e Machining PowerPac)	48
Figura 4.43 - Trajetória gerada (RobotStudio e PowerPac)	49
Figura 4.44 - Colisão entre o <i>Spindle</i> e a Peça 3	49
Figura 5.1 - Célula robótica da FEUP	53
Figura 5.2 - ABB <i>IRB 2400-16</i>	54
Figura 5.3 - Controlador <i>IRC5</i>	54
Figura 5.4 - Mesa ABB <i>IRBP C-500</i>	55
Figura 5.5 - ATI <i>Delta IP60</i>	55
Figura 5.6 - <i>Schunk SWS-011</i>	56
Figura 5.7 - <i>Spindle XLC70</i>	56

Figura 5.8 - Poliuretano expandido	56
Figura 5.9 - Fixação do bloco de trabalho	57
Figura 5.10 - Peça 1 (Sprutcam esquerda e Mastercam/Robotmaster direita)	58
Figura 5.11 - Peça 2 (Sprutcam esquerda e Mastercam/Robotmaster direita)	58
Figura 5.12 - 1ª Maquinagem da Peça 3 (Mastercam/Robotmaster).....	59
Figura 5.13 - Peça 3 (Sprutcam esquerda e Mastercam/Robotmaster direita)	59

Índice de Tabelas

Tabela 1 - Linguagem de programação de robôs industriais.....	14
Tabela 2 - <i>Software</i> de programação de robôs industriais.....	14
Tabela 3 - <i>Software</i> CAM dotado de pós-processadores para robôs.....	15
Tabela 4 - <i>Software</i> de conversão.....	16
Tabela 5 - Análise comparativa.....	20
Tabela 6 - Posicionamento e orientação do TCP em relação ao referencial tool0.....	23
Tabela 7 - Parâmetros de corte.....	26
Tabela 8 - Estratégias de maquinagem.....	26
Tabela 9 - Resultados de maquinagem da Peça 0 (Estratégia Paralela).....	27
Tabela 10 - Estratégia Paralela (após edição dos programas RAPID).....	29
Tabela 11 - Estratégia Equidistante (após edição dos programas RAPID).....	29
Tabela 12 - Estratégia Adaptativa (após edição dos programas RAPID).....	30
Tabela 13 - Parâmetros de corte.....	33
Tabela 14 - Resultados da simulação do programa de maquinagem da Peça 1.....	50
Tabela 15 - Resultados da simulação do programa de maquinagem da Peça 2.....	51
Tabela 16 - Resultados da simulação do programa de maquinagem da Peça 3.....	51
Tabela 17 - Características do robô <i>IRB 2400-16</i>	54
Tabela 18 - Dimensões das ferramentas.....	57

1 Introdução

Adaptabilidade, mudança e inovação são algumas características necessárias às empresas para conseguirem sobreviver no mundo competitivo atual. A constante evolução leva a que as indústrias apostem cada vez mais na otimização dos tempos de produção. Atualmente, a utilização de máquinas ferramenta CNC encontra-se implementada na maioria dos processos de maquinagem, devido às inúmeras vantagens que estas apresentam em relação à produção manual, tais como a qualidade do produto final com tempos de produção reduzidos. Porém, o desenvolvimento contínuo na área da robótica permite dispor de soluções versáteis como a aplicabilidade em operações de maquinagem. Contudo, a utilização de robôs industriais em processos de maquinagem apresenta ainda limitações que não lhes permite substituir as máquinas ferramenta CNC, sendo de destacar as seguintes: calibração do robô, vibrações e deformações associadas à sua baixa rigidez e a dificuldade de programação de percursos de maquinagem [1]. A utilização de *software* CAD/CAM permite facilitar a programação e será este o ponto estudado na presente dissertação.

1.1 Objetivos do trabalho

A maquinagem, em particular a fresagem, é um dos processos de fabrico em que o uso de robôs industriais tem vindo a ser adotado. Com o presente trabalho pretende-se estudar, analisar e simular o processo de programação de robôs industriais, recorrendo a diferentes aplicações informáticas que podem ser utilizadas na programação de robôs industriais.

Um dos objetivos é analisar as implicações das especificações particulares, associadas a cada estratégia de maquinagem, no programa que controla o robô na execução das operações de maquinagem. Pretende-se assim, estudar a aplicabilidade de robôs industriais a este processo de maquinagem e implementar uma demonstração de fresagem de peças envolvendo a maquinagem de 3 e 5 eixos.

Para esse efeito foram selecionadas três aplicações de cariz industrial que permitem a programação de robôs industriais, sendo estas: Mastercam/Robotmaster, Sprutcam e RobotStudio/Machining PowerPac.

Os dois primeiros *softwares* são exemplos de aplicações que podem ser utilizados para programar robôs de diferentes fabricantes em operações de maquinagem.

O último dos *softwares* referido é uma aplicação que só permite a programação de robôs do fabricante ABB e que tem uma aplicabilidade mais genérica, não estando vocacionado especificamente para operações de maquinagem. No entanto, possui algumas funcionalidades que permitem a sua utilização na programação de operações de maquinagem e, pelo facto de

ser o *software* proprietário utilizado pelos robôs da ABB, serve para validar a programação que é gerada a partir das outras duas aplicações, Mastercam/Robotmaster e Sprutcam.

1.2 Organização da dissertação

A presente dissertação está dividida em 6 capítulos. No capítulo 2, *Maquinagem com robôs industriais*, é feita uma breve apresentação da robótica industrial. São apresentados alguns robôs otimizados para processos de maquinagem, assim como as aplicações típicas de maquinagem com recurso a robôs industriais. Tendo escolhido a fresagem como operação de maquinagem a realizar com robôs, apresenta-se ainda uma descrição das operações de fresagem e caracterização dos parâmetros do processo.

No capítulo 3, denominado *Programação de robôs industriais* descrevem-se os diferentes métodos de programação de robôs e são apresentadas algumas soluções existentes no mercado. Também neste capítulo é apresentada uma introdução dos diferentes programas utilizados para a realização deste trabalho, assim como as funcionalidades mais pertinentes de cada um.

No capítulo 4, intitulado *Casos de estudo de programação de robôs para maquinagem* descreve-se o procedimento para a realização de peças envolvendo aplicações de maquinagem 2D, 3D e 5D, utilizando os programas: Sprutcam, Mastercam/Robotmaster e RobotStudio/Machining PowerPac.

No capítulo 5, denominado *Maquinagem com célula robótica real* é descrita a célula robótica utilizada assim como todo o equipamento utilizado na implementação prática de operações de maquinagem.

Finalmente no capítulo 6 são apresentadas as conclusões do trabalho, assim como algumas sugestões para trabalhos futuros.

2 Maquinagem com robôs industriais

Após uma breve referência ao aparecimento e evolução da robótica industrial, assim como aos componentes principais de um robô industrial (RI), este capítulo centra-se nas aplicações de RI para operações de maquinagem, tendo em conta as soluções disponibilizadas pelos fabricantes. Termina com a caracterização do processo de maquinagem considerado, a fresagem, de modo a estabelecer quais os critérios na escolha dos vários parâmetros e da sua importância para a qualidade e objetivos de maquinagem.

2.1 Breve introdução à robótica industrial

Em 1920 Karel Čapek, romancista e jornalista checo, criou uma obra de ficção científica intitulada “*Rossum’s Universal Robots*” onde usou pela primeira vez a palavra *Robota*, que em checo significa servidão ou trabalhador forçado [2]. Embora no final desta obra os robôs pudessem ser vistos como “seres malignos”, pois com o desenrolar dos atos tentam acabar com a vida humana, o propósito da sua invenção não estaria longe da realidade atual da palavra.

O desenvolvimento histórico dos robôs é um exemplo bastante interessante de como os mitos são capazes de se tornarem realidade e em como a ficção é capaz de motivar o ser humano a atingir algo nunca antes alcançado. Se em 1920 a palavra de algo inatingível era concebida apenas como uma ideia ou uma ficção, George Devol torna-a realidade em 1954 ao desenvolver o primeiro robô industrial [3]. Este robô era fabricado na empresa de George Devol e Joseph Engelberger, a Unimation, e seria denominado Unimate. A General Motors, em 1961, foi a primeira empresa a utilizar este robô na indústria, realizando a movimentação de peças obtidas por processos de fundição.

Onze anos após a primeira utilização de robôs na indústria a empresa germânica KUKA criou o primeiro robô industrial de 6 eixos [4]. O FAMULUS, manipulador em questão, permitiu que os robôs industriais realizassem processos mais complexos sem a necessidade constante da intervenção humana, predominantemente na soldadura por arco elétrico. Sendo na altura uma novidade tecnológica, foi necessário a implementação de medidas de segurança para evitar acidentes. As precauções tomadas passavam pela instalação de uma barreira física à volta do robô e a implementação de sensores nas portas de acesso.

Em 1974, a empresa sueca ASEA introduziu no mercado o primeiro robô industrial de atuação elétrica, um manipulador com 5 graus de liberdade. Este era controlado por um microprocessador e tinha uma capacidade de carga de 6 Kg [5].

A evolução desta área foi bastante rápida devido à capacidade de adaptação que os robôs proporcionavam e o crescente interesse na automatização por parte das empresas. Um caso

interessante passa por em 1988 o grupo Motoman apresentar um sistema capaz de controlar 12 eixos, mais do que qualquer outro controlador existente na altura. Apenas 6 anos depois, a mesma empresa, lança um novo controlador capaz de movimentar 21 eixos e de sincronizar o movimento de dois robôs industriais. Através da *International Federation of Robotics* é possível verificar, em particular nos últimos anos, que a Robótica é uma área em crescimento (Figura 2.1) com aplicabilidade em múltiplas atividades industriais (Figura 2.2), mas com a indústria automóvel a continuar como o grande utilizador de robôs industriais [6].

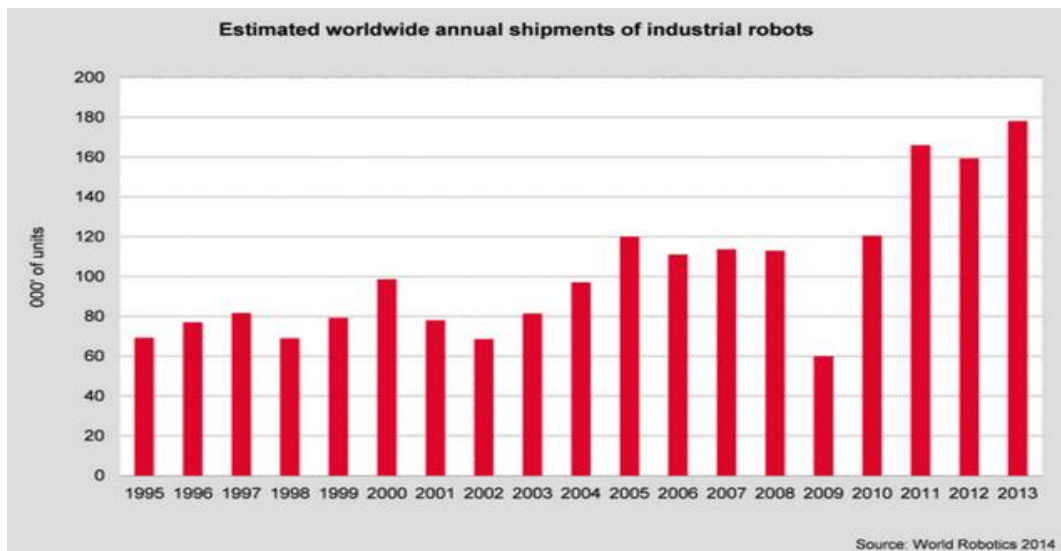


Figura 2.1 - Número estimado anual de robôs industriais vendidos [6]

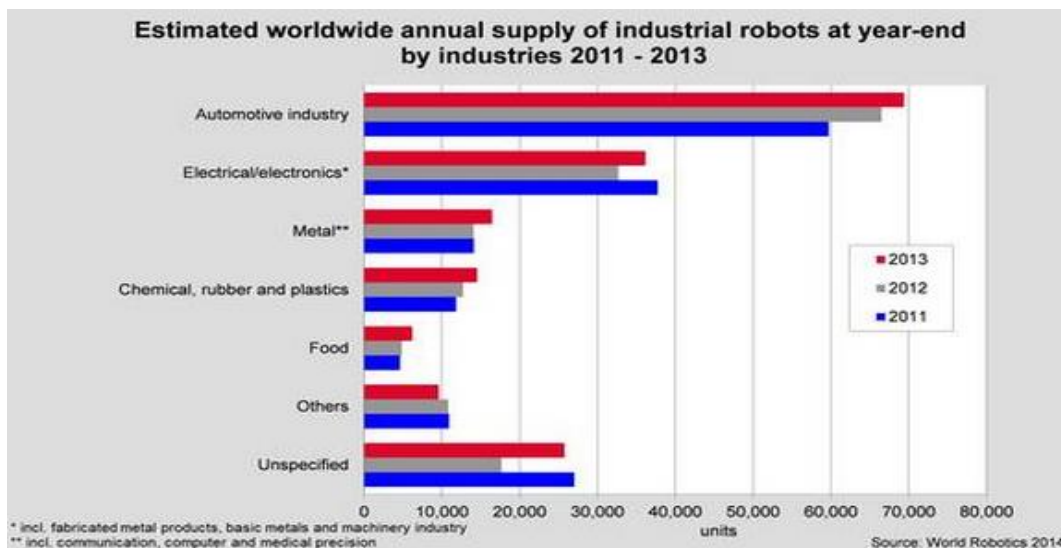


Figura 2.2 - Número estimado anual de robôs industriais fornecidos por indústria [6]

Das várias indústrias que utilizam robôs no seu dia-a-dia, além dos apresentados anteriormente, é possível destacar alguns exemplos:

- **Aeroespacial:** normalmente as peças produzidas nesta indústria são complexas e de uma dimensão considerável, sendo por isso especialmente importante a manobrabilidade dos robôs industriais. O facto de os robôs apresentarem uma grande versatilidade nas peças que podem produzir, mantendo as dimensões exigidas, ou até movimentar peças, representam vantagens importantes. Os processos executados por robôs nesta área são fundamentalmente os seguintes: furação (Figura 2.3), soldadura, rebarbagem, transporte de material e pintura.

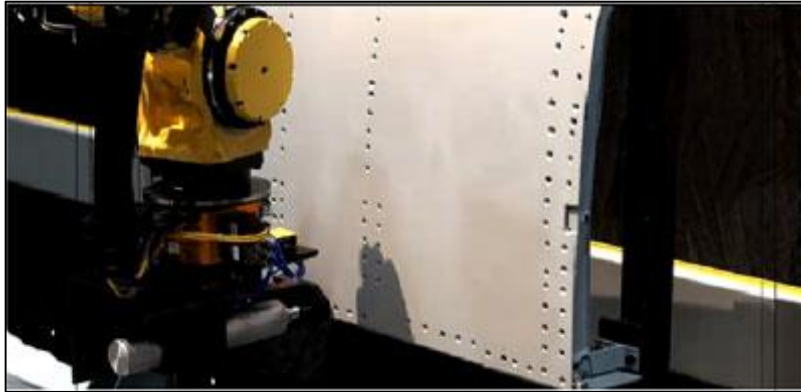


Figura 2.3 - Fanuc R-2000iB numa operação de furação [7]

- **Automóvel:** sendo uma indústria com grande volume de produção, os fabricantes de automóveis têm a necessidade de produzir com uma rapidez elevada e manter a fiabilidade do produto final. Com o uso de robôs, os trabalhadores deixam de realizar processos repetitivos ou eventualmente perigosos, como por exemplo a soldadura, e é ainda possível melhorar os tempos de produção. Através da interação dos robôs é possível executar processos desde polimento, corte, furação, rebarbagem (Figura 2.4) e soldadura.



Figura 2.4 - Rebarbagem de um motor diesel utilizando robôs [8]

- **Fundição:** operações de fundição são trabalhos muito pouco desejáveis pelos seres humanos devido às condições adversas existentes nesse tipo de ambiente. A sujidade, o perigo, e as temperaturas elevadas são algumas das razões para os seres humanos não quererem esse tipo de trabalho. Assim os robôs apresentam uma ótima alternativa para os empresários, conseguindo aumentar a produtividade e ainda precaver vidas humanas de eventuais acidentes. Os principais processos que utilizam robôs nesta área são: rebarbagem, polimento e transporte de material.

2.2 Principais componentes de um robô industrial

De acordo com a norma ISO 8373 um robô industrial é definido como: “*An automatically controlled, reprogrammable, multipurpose, manipulator programmable in three or more axes, which may be either fixed in place or mobile for use in industrial automation applications*”. Esta definição pode corresponder a diferentes estruturas, sendo no entanto possível identificar os seguintes elementos funcionais principais:

- Manipulador: a estrutura física do robô, muitas vezes referido como o “braço” do robô, é composto por um conjunto de corpos rígidos ligados por juntas, formando uma cadeia cinemática. No manipulador inclui-se, também, os atuadores e sensores necessários para cada eixo de movimento. O dispositivo ligado ao manipulador que interage com o ambiente e tem por objetivo executar as tarefas definidas pelo utilizador, é denominado de órgão terminal (ex. garra ou *Spindle*).
- Controlador: é o “cérebro” do robô industrial. Este dispositivo é baseado num computador com funções particulares. Inclui funções de controlo associadas aos eixos de movimento e outros dispositivos que podem estar associados às tarefas do robô. A linguagem de programação é específica para cada fabricante. Juntamente com o controlador existe uma consola portátil que permite a interação entre o robô e o utilizador. A programação pode ser feita *on-line* (na consola) ou *off-line* (no computador).
- Unidade de potência: são os dispositivos que integram o circuito de potência, de modo a proporcionar energia aos atuadores.

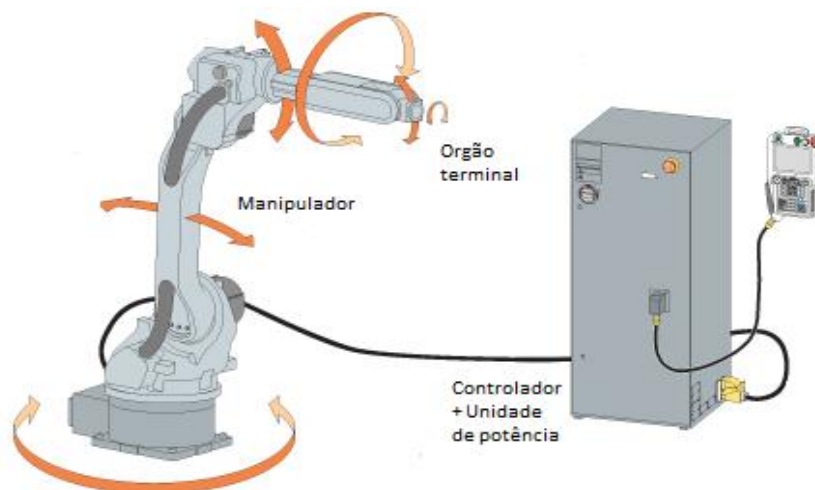


Figura 2.5 - Componentes de um robô industrial articulado

Na atualidade existem vários tipos de robôs industriais. Dependendo da sua configuração física é possível inseri-los em dois grandes grupos: estrutura em série ou estrutura em paralelo.

O robô utilizado na presente dissertação é um robô em série de estrutura revoluta (Anexo A). Este tipo de estrutura utiliza exclusivamente eixos de movimento rotativo, apresentando uma boa manobrabilidade, elevada velocidade de trabalho e um grande volume de trabalho para uma pequena área de montagem. Um robô com uma estrutura em série é caracterizado por [9]: “conjunto de corpos ligados por juntas, formando uma cadeia cinemática que definem uma estrutura mecânica. Uma das extremidades do manipulador encontra-se rigidamente ligada a uma base, enquanto a extremidade oposta suporta o órgão terminal, podendo mover-se livremente no espaço”.

2.3 Maquinagem com robôs

As técnicas convencionais de remoção de material, como a fresagem executada por máquinas CNC, têm provado serem capazes de responder com rapidez à produção de peças complexas garantindo a qualidade necessária em termos de acabamento superficial e toleranciamento dimensional. Por outro lado, o grande inconveniente do uso de máquinas CNC são a sua área ou volume de trabalho restrito e o custo do equipamento. Do ponto de vista conceptual, a utilização de robôs industriais pode ser vista como uma hipótese bastante interessante para a maquinagem devido às seguintes razões [10]:

- 1- O custo de um robô industrial para maquinagem é cerca de 30% inferior quando comparado a uma máquina CNC equivalente.
- 2- O robô pode movimentar os produtos, algo que na maquinagem executada por máquinas CNC é realizada por um operador.
- 3- Os robôs industriais pressupõem logo à partida máquinas com um nº elevado de graus de liberdade, dotando-os de uma grande manobrabilidade para as operações de maquinagem.
- 4- Um robô industrial pode ser facilmente instalado sobre uma plataforma com um eixo linear, por exemplo sobre carris, aumentando assim o seu volume de trabalho.

As desvantagens no uso de robôs industriais para maquinagem têm que ver com a sua baixa rigidez, pelo que é difícil absorver/rejeitar vibrações resultantes dos processos de maquinagem, dificultando a obtenção de uma boa exatidão no posicionamento do órgão terminal.

Para além dos problemas de rigidez, uma das dificuldades na utilização de robôs para aplicações de maquinagem convencionais (ex. fresagem), resulta de problemas particulares de programação neste tipo de aplicações. Os robôs foram pensados sempre como manipuladores para descrever trajetórias, ou percursos pré-definidos, bastando o conhecimento da “forma” a seguir, para a definição do seu programa. No caso da maquinagem, é necessário além da forma final, o conhecimento da forma inicial, e a estratégia a escolher para através de vários percursos retirar o volume de material necessário para obter a forma final. Havendo várias soluções possíveis, as ferramentas de *software* para programação de robôs industriais não estão normalmente vocacionadas para este tipo de aplicação.

Para a área da maquinagem, diversos fabricantes de robôs industriais disponibilizam robôs específicos, pelo que são apresentados alguns exemplos.

ABB (*ASEA Brown Boveri*)

Este grupo empresarial surgiu da fusão de duas grandes empresas: a ASEA (*Allmänna Svenska Elektriska Aktiebolaget*) e a Brown Boveri. Atualmente o grupo ABB tem a sede em Zurique, na Suíça, tendo representantes em mais de 100 países e emprega cerca de 145.000 pessoas na área de Automação e de Energia [10]. O robô mais adequado para operações de maquinagem deste grupo é o *IRB 6660-205/1.9*. A capacidade de carga deste robô é de 205 Kg, tem um alcance de 1.93 m, um peso de 1730 Kg, uma repetibilidade de 0.1 mm e apresenta uma estrutura revoluta com 6 graus de liberdade [11].

FANUC (*Fujitsu Automated Numerical Control*)

Fundada em 1956, esta empresa entrou no mercado ao desenvolver e comercializar dois tipos de produtos: servomecanismos e sistemas de comando numérico. Atualmente sediada em Oshino, no Japão, oferece uma vasta lista de serviços e produtos na área dos servomotores, sistemas de comando numérico computadorizado e robôs industriais. Um dos robôs para utilização em processos de maquinagem fabricado por esta empresa é o *F-200iB*. Este robô de estrutura paralela tem 6 graus de liberdade com 6 atuadores idênticos, conferindo-lhe uma boa rigidez, capacidade de carga de 100 Kg e uma repetibilidade de 0.1 mm, embora possua um volume de trabalho mais reduzido (1040 x 1022 x 494 [mm]) [12].

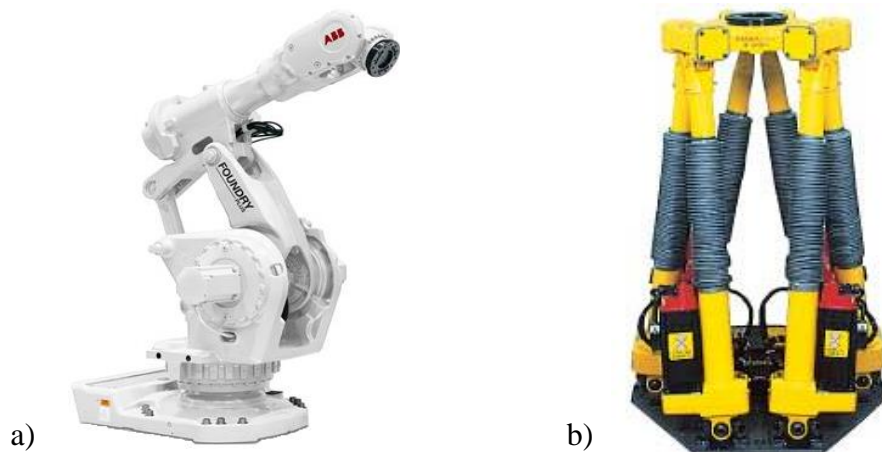


Figura 2.6 - a) ABB IRB 6660 e b) Fanuc F-200iB

KUKA (*Keller und Knappich Augsburg*)

Sediada em Augsburg, na Alemanha, a KUKA é uma empresa alemã fundada em 1898 por Johann Josef Keller e Jakob Knappich. Atualmente é líder de mercado na Alemanha e está dentro dos quatro primeiros fabricantes de robôs industriais em todo o mundo. Na área da Robótica, a KUKA Robot Group, tem como objetivos principais o desenvolvimento, produção e venda de robôs industriais, controladores e *software*. Esta divisão da empresa emprega cerca de 1.800 pessoas em todo o mundo. Um robô fabricado nesta empresa adequado para operações de maquinagem é o *KR 500 R2830 MT*. Apresenta uma repetibilidade de +/- 0.08 mm, 6 graus de liberdade, capacidade de mover cargas até 500 Kg e um alcance máximo de 2.826 m [13].

Stäubli

Stäubli, atualmente sediada em Pfäffikon na Suíça fornece soluções em duas áreas distintas: têxtil e robótica. Esta empresa, fundada em 1892, emprega atualmente mais de 4000 pessoas. No que diz respeito a robôs industriais para processos de maquinagem um robô com características interessantes deste grupo é o RX 170 hsm. Apresenta uma repetibilidade de ± 0.04 mm, capacidade de carga de 65 Kg e um alcance de 1.835 m. Tem 5 graus de liberdade e o motor-árvore vem integrado no braço do robô [14].

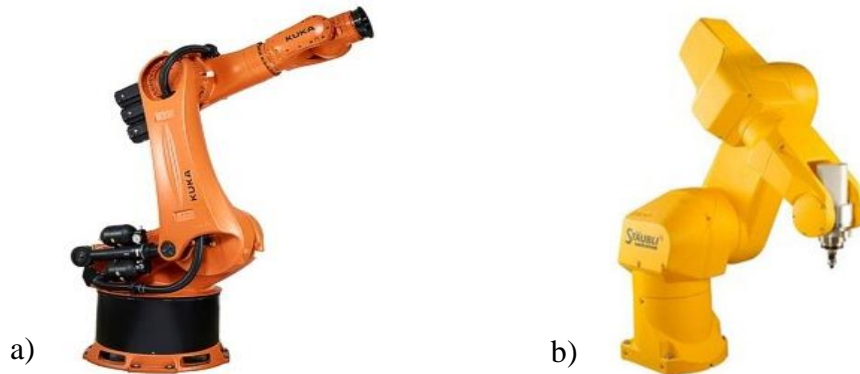


Figura 2.7 - a) Kuka KR 500 R2830 MT e b) Stäubli RX 170 hsm

2.4 Operações de fresagem

Na maquinagem, a fresagem é um dos processos mais importantes na indústria. A evolução tecnológica na área das ferramentas e equipamentos, tornam este processo cada vez mais abrangente e competitivo, atingindo níveis de tolerância cada vez mais exigentes. A elevada qualidade de acabamento e a possibilidade de se obterem geometrias cada vez mais complexas, fazem deste processo um dos mais utilizados na indústria automóvel, aeroespacial e na produção de moldes. Este processo consiste na remoção do excesso de material da superfície de uma peça, com o objetivo de dar a esta a forma ou acabamento desejado utilizando uma ferramenta (fresa) com múltiplas arestas ou pontos de corte.

A fresagem pode ser realizada de duas maneiras: fresagem de topo ou fresagem cilíndrica. Como é possível visualizar através da Figura 2.8 (a), na fresagem cilíndrica o eixo de rotação da fresa é paralelo à superfície da peça a maquinar. Devido à carga irregular exercida pela fresa, na fresagem cilíndrica, as peças apresentam no final uma superfície ondulada. Na fresagem de topo (Figura 2.8 (b)), o eixo de rotação da fresa encontra-se perpendicular à superfície a maquinar.1

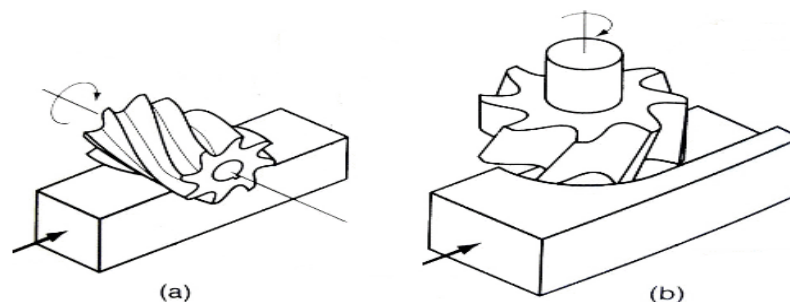


Figura 2.8 - Fresagem cilíndrica (a) e fresagem de topo (b)

Dentro da fresagem cilíndrica é possível maquinar a superfície da peça de duas maneiras: fresagem cilíndrica “discordante” ou fresagem cilíndrica “concordante”. No caso da fresagem “discordante”, o sentido do movimento de corte é contrário ao de avanço e o arranque da apara inicia-se na zona mais delgada. A ferramenta tende a levantar a peça, exigindo assim um sistema de fixação mais rígido. O escorregamento da fresa provoca um elevado atrito, traduzindo-se num maior desgaste das lâminas. Na fresagem cilíndrica concordante, o movimento de corte e avanço têm o mesmo sentido. Através da Figura 2.9 é possível compreender que uma das características deste processo é a ferramenta empurrar a peça contra a mesa de trabalho, sendo recomendado o uso desta técnica no caso de haver uma fixação menos rígida. Ao contrário da fresagem discordante, na fresagem concordante, o arranque da apara inicia-se na zona mais espessa. É normal utilizar-se este processo quando se pretende utilizar maiores profundidades de corte.

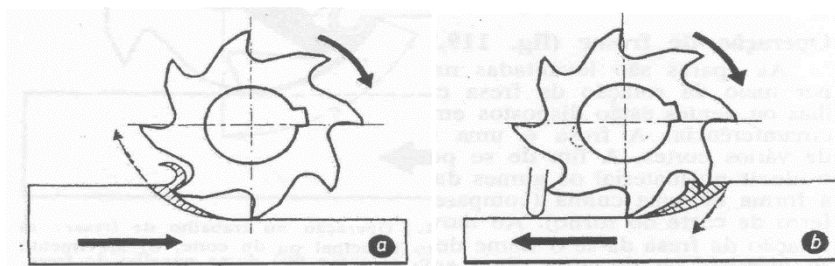


Figura 2.9 - Fresagem “a empurrar” (a) e fresagem “a puxar” (b)

Caso se pretenda um maior rendimento na remoção de material, a fresagem de topo será a mais eficaz pois, quando comparada à fresagem cilíndrica, apresenta uma taxa de remoção 20% maior [15]. Na fresagem a qualidade do produto final depende de vários parâmetros. Estes descrevem quantitativamente os movimentos, as dimensões e outras características da operação de corte. De seguida são enunciados os parâmetros mais importantes na fresagem, representados na Figura 2.10:

- **Velocidade de rotação (N) [rpm]:** velocidade angular da ferramenta (fresa).
- **Velocidade de corte (Vc) [m/min]:** velocidade tangencial instantânea do ponto selecionado sobre a lâmina, no movimento de corte, em relação à peça.
- **Passo [mm]:** distância que uma ferramenta se desloca entre as passagens de percurso adjacentes.
- **Avanço por dente (ft) [mm]:** distância em que dois dentes consecutivos entram em corte.
- **Velocidade de avanço (fm) [mm/min]:** velocidade instantânea do ponto selecionado sobre a lâmina, no movimento de avanço, em relação à peça.
- **Profundidade de corte (d) [mm]:** distância que a fresa penetra na peça (na direção do eixo da fresa, também denominado por *Stepdown*).
- **Largura de corte (w) [mm]:** poderá no limite ser igual ao diâmetro da ferramenta.

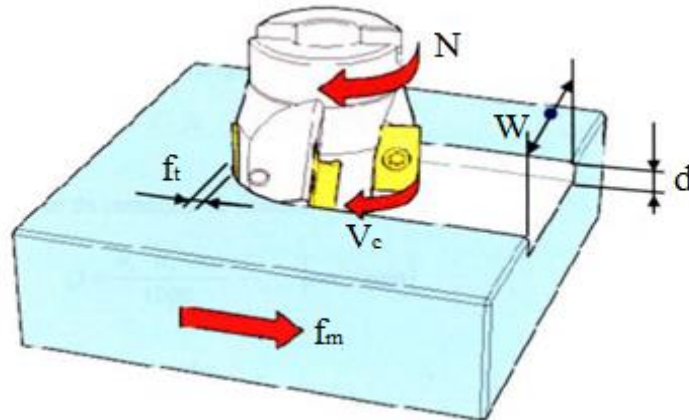


Figura 2.10 - Parâmetros de corte (fresagem de topo)

Nas operações de maquinagem, a escolha dos valores adequados para os vários parâmetros de corte, é obviamente um passo necessário e determinante para a produtividade e qualidade do processo. Isto é, a rapidez com que se remove um dado volume de material satisfazendo ao mesmo tempo as especificações de acabamento superficial e dimensionais da peça maquinada. Simultaneamente há ainda que considerar os efeitos na vida útil da ferramenta uma vez que, em geral, quanto maior for a taxa de remoção de material maior será a produtividade mas também, em geral, maior será o desgaste da ferramenta. O grande número de fatores envolvidos e parâmetros a considerar nas operações de maquinagem tornam difícil o estabelecimento de modelos que traduzam o comportamento real destes processos. A escolha de valores de referência para os parâmetros de corte baseia-se assim em tabelas obtidas com base em ensaios experimentais e modelos simplificados, baseados na geometria da ferramenta e movimentos do processo de maquinagem.

Em termos de produtividade o objetivo será utilizar a taxa de remoção de material máxima compatível com a potência disponível. Na fresagem a taxa de remoção de material (Q) pode ser estimada pela equação 1 com base na velocidade de avanço (f_m), largura de corte (w) e profundidade de corte (d).

$$Q = f_m * w * d \quad (1)$$

A potência de corte necessária para atingir esta taxa de remoção depende naturalmente do material a ser maquinado e outros fatores entre os quais o material da própria ferramenta. Usando o modelo proposto em [16], a potência de corte (P_m) pode ser estimada pela equação 2 aplicando um coeficiente de eficiência da ferramenta (E) à potência de corte na ferramenta (P_c). Esta é calculada pela equação 3, a partir da taxa de remoção de material (Q), um fator de potência (K_p), um fator de velocidade de avanço (C) e um fator de desgaste (W).

$$P_m = P_c / E \quad (2)$$

$$P_c = K_p * C * Q * W \quad (3)$$

Os coeficientes e fatores usados nestas equações podem ser obtidos através das tabelas disponíveis em [16]. Estes traduzem a influência da transmissão de potência da árvore da máquina para a peça através da ferramenta, determinada experimentalmente.

Tendo por base a limitação da potência disponível e os objetivos de taxa de remoção de material, há ainda que considerar várias condições do processo de fresagem na definição dos três parâmetros de corte em cada operação, velocidade de avanço, profundidade e velocidade de corte. Apesar da maioria dos materiais poderem ser maquinados numa gama de velocidades relativamente ampla, esse intervalo é mais reduzido quando se consideram outros fatores de eficiência e custos do processo, nomeadamente o desgaste ou tempo de vida útil da ferramenta. Dependendo da fresa a utilizar, do material a maquinar e da própria operação a realizar (desbaste ou acabamento), é possível selecionar uma velocidade maior ou menor, para prolongar o tempo de vida útil da ferramenta.

Um procedimento recomendado para escolha, de uma forma iterativa dos três parâmetros de corte, baseia-se na constatação de que o tempo de vida útil da ferramenta é mais afetado pela velocidade de corte, seguidamente pelo avanço e só depois pela profundidade de corte. Dessa forma, o procedimento envolve:

- 1) Escolher a profundidade de corte máxima, compatível com a operação de maquinagem;
- 2) Escolher a velocidade de avanço compatível com os requisitos de acabamento e rigidez da ferramenta, peça a maquinar e máquina-ferramenta;
- 3) Escolher a velocidade de corte.

A escolha da profundidade de corte é limitada pela quantidade de material existente no bloco de trabalho, potência disponível no motor-árvore, rigidez do material a ser maquinado e da ferramenta utilizada, A profundidade de corte é o parâmetro que menos influência tem no tempo de vida útil da ferramenta e, por isso, deve-se usar a maior profundidade de corte possível. O valor da profundidade de corte deverá, na maioria dos casos, ser 50% a 80% do valor do diâmetro da ferramenta de corte para operações de desbaste e 25% a 50% para operações de acabamento [17].

A determinação da velocidade de avanço (f_m) pode ser obtida a partir do avanço por lâmina (f_t , [mm]), do número de lâminas da fresa (n_t) e da velocidade de rotação (N), equação 4:

$$f_m = f_t * n_t * N \quad (4)$$

A potência disponível tem de ser suficiente para realizar a fresagem com a profundidade de corte e velocidade de avanço selecionados. Deve ser utilizada a maior velocidade de avanço possível desde que o acabamento superficial seja aceitável.

Por último a seleção da velocidade de corte é realizada com recurso a valores tabelados. Estes valores têm em conta estudos realizados em materiais com características diferentes utilizando ferramentas também diferentes. Quando não forem utilizadas as combinações de velocidade de avanço e de corte tabeladas e associadas a valores ótimos, será necessário ajustar os parâmetros de corte.

Em suma a escolha dos parâmetros de corte em operações de maquinagem tem uma importância fundamental na qualidade e produtividade do processo de maquinagem, sendo dependente de vários fatores e condições em que o processo pode decorrer. O recurso a dados experimentais para definir os valores desses parâmetros é uma prática habitual e deve ser tida em conta na avaliação das condições estabelecidas para comparar a maquinagem com robôs com a maquinagem em máquinas-ferramentas desde a elaboração do programa de maquinagem até a sua implementação.

3 Programação de robôs industriais

A programação de robôs industriais pode ser feita utilizando programação *on-line* ou programação *off-line*. No seguimento das alternativas de programação *off-line*, são apresentadas com maior detalhe as aplicações de *software* analisadas: Sprutcam, Mastercam/Robotmaster e RobotStudio/Machining PowerPac.

3.1 Programação *on-line*

A programação *on-line* requer a utilização física do robô, sendo este movimentado para as localizações necessárias que permitem definir as trajetórias a realizar. Para efetuar a movimentação do robô podem ser utilizados dois processos (Figura 3.1): através da consola portátil (*teach-pendant*) ou por *manual leadthrough*. O *teach-pendant* diz respeito ao uso da consola de interface do robô para o fazer movimentar de forma assistida e gravar as trajetórias desejadas. A programação do robô por *manual leadthrough* envolve a deslocação direta do manipulador por parte do operador. Este tipo de programação *on-line* tem como grande vantagem a facilidade de o operador criar o programa pretendido de forma expedita. As desvantagens deste processo incluem a necessidade de o robô estar disponível, uma vez que apenas se poderá programar interagindo diretamente com o robô, originando assim tempos não produtivos para a empresa. No que diz respeito à segurança também não é a melhor escolha, pois nem sempre é possível manter o programador fora do alcance do robô podendo desta maneira aumentar o risco de ocorrência de acidentes.



Figura 3.1 - Manual leadthrough e teach-pendant

Este tipo de programação não é utilizado para a programação de robôs industriais em processos de fresagem.

3.2 Programação *off-line*

Uma possível abordagem para resolver os problemas anteriormente referidos passa pelo uso da programação *off-line*. Este tipo de programação não necessita do uso direto do robô, permitindo assim a continuidade na produção enquanto se elabora um novo programa. Atualmente é possível realizar a simulação para verificação e validação das trajetórias do robô evitando assim possíveis colisões ou acidentes na célula real. Na Tabela 1 é possível verificar que consoante o fabricante a linguagem de programação dos robôs é distinta.

Tabela 1 - Linguagem de programação de robôs industriais

Fabricante	Linguagem
ABB	Rapid
KUKA	KRL
Motoman	INFORM
Fanuc	Karel
Stäubli	VAL3

Para a programação *off-line* de robôs industriais existem disponíveis no mercado diferentes aplicações informáticas, quer específicas para um só fabricante de robôs, quer genéricas. A Tabela 2 apresenta alguns dos *softwares* existentes. Estes permitem a programação de robôs em aplicações típicas, desde alimentação de máquinas, soldadura por pontos ou arco-elétrico, até pintura. Porém são limitadas quando se pretende programar para operações de maquinagem, pelo que é necessário considerar outras alternativas.

Tabela 2 - Software de programação de robôs industriais

<i>Software</i> genérico	Robôs compatíveis (exemplos)
Workspace 5 [18]	ABB
Octopuz [19]	KUKA
Delmia [20]	Fanuc
<i>Software</i> proprietário	Robôs compatíveis
RobotStudio	ABB
RoboGuide [21]	Fanuc
MotoSim EG [22]	Motoman

3.3 Programação de robôs industriais para operações de maquinagem

Genericamente o *software* CAM permite criar um programa de maquinagem numa linguagem normalizada (ISO 6983) para máquinas CN, frequentemente designado por código G&M, que não é compatível com a linguagem dos controladores dos robôs industriais. Apesar da linguagem com código G&M ser normalizada, admite especificações ou configurações particulares, o que leva à necessidade de nos sistemas CAM se gerar primeiro o programa num código independente da máquina (*CLData file*), posteriormente convertido através de um programa de pós-processamento (pós-processador) para cada máquina em particular. A utilização destes sistemas de CAM para gerar um programa de maquinagem para robôs envolve a utilização de um pós-processador específico para o robô a utilizar.

O modo como um pós-processador para um robô se integra no *software* CAM permite identificar duas alternativas para a programação de operações de maquinagem:

- ***Software* CAM genérico que disponibiliza pós-processadores integrados.**

Recorrendo ao uso de um *software* CAM genérico é possível fazer a programação de robôs de fabricantes distintos. Para criar um programa de maquinagem é necessário seleccionar o modelo do robô a utilizar e ter disponível o pós-processador respetivo para se gerar o programa na linguagem de programação do robô pretendida. Na Tabela 3 são apresentados alguns programas genéricos de CAM que disponibilizam pós-processadores integrados, para diversos fabricantes de robôs.

Tabela 3 - *Software* CAM dotado de pós-processadores para robôs

<i>Software</i>	Robôs compatíveis (exemplos)
Sprutcam	ABB, FANUC, KUKA, MOTOMAN, Stäubli
Mastercam/Robotmaster	
Delcam PowerMill Robot [23]	

- ***Software* CAM genérico e *software* de conversão.**

Esta opção envolve o uso de duas aplicações informáticas distintas. Em primeiro lugar é utilizado um *software* CAM para criar o programa de maquinagem. Posteriormente é importado para um outro *software* capaz de converter o programa gerado pelo CAM na linguagem de programação do robô seleccionado. Esquemáticamente esta hipótese de programação é representada na Figura 3.2.

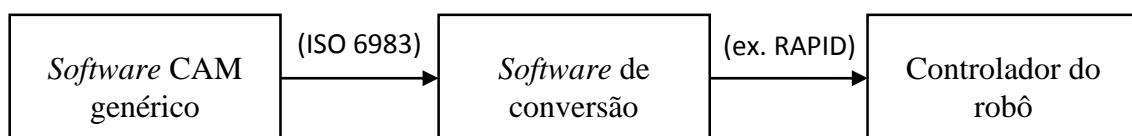


Figura 3.2 - Conversão de um programa de maquinagem

Seguidamente apresentam-se na Tabela 4 exemplos de *softwares* que possibilitam a referida conversão do código obtido no CAM para o código compatível com alguns robôs industriais.

Tabela 4 - Software de conversão

Exemplos de Software CAM	Software de conversão	Robôs Compatíveis
AlphaCam, Visi, RTM, EdgeCam, GO2CAM, CamWorks, Creo, MasterCAM, CIMATRON, GibbsCam, DelCam, WorkNC, Tebis, ZW3D, WINTOOL, SurfCAM, Esprit, ProEngineer, VisualMill, RhinoCAM, PowerMill, SolidCam	Eureka [24]	ABB, KUKA, Fanuc, Motoman, Kawasaki, Stäubli, Comau
	IRBCAM [25]	ABB, Comau, FANUC, Hyundai, KUKA, Mitsubishi, Motoman, Stäubli, Toshiba
	Conifer Rob [26]	ABB, KUKA, Motoman
	CAMRob [27]	KUKA
	G-Code Converter EG [28]	Motoman
	CAM Converter [29]	ABB

Existe ainda a possibilidade de utilização do *software* de programação proprietário do robô, que, mediante a utilização de *software* adicional, facilitam a programação de robôs em aplicações de maquinagem. No entanto, as funcionalidades disponibilizadas são muito reduzidas quando comparadas com as duas alternativas referidas anteriormente. Tipicamente, as operações de maquinagem estão limitadas a rebarbagem e polimento de superfícies. Um exemplo desta possível abordagem de programação de robôs, é a aplicação do *software* RobotStudio dotado do *add-on* Machining PowerPac, que só pode ser utilizado com robôs da ABB.

3.4 Sprutcam 9

Atualmente o Sprutcam 9 é a mais recente aposta da empresa SPRUT Technology, sediada em Naberezhnye Chelny, Rússia. O Sprutcam 9 é um *software* CAM que disponibiliza pós-processadores para diversas máquinas ferramenta e para robôs industriais. No âmbito deste trabalho, é utilizado este *software* de programação *off-line*. A empresa SPRUT Technology disponibiliza licenças académicas que contemplam todas as funcionalidades do programa, pelo que este *software* foi um dos escolhidos. As principais funcionalidades na utilização deste *software* são descritas de seguida e apresentada na Figura 3.3 a interface do mesmo.

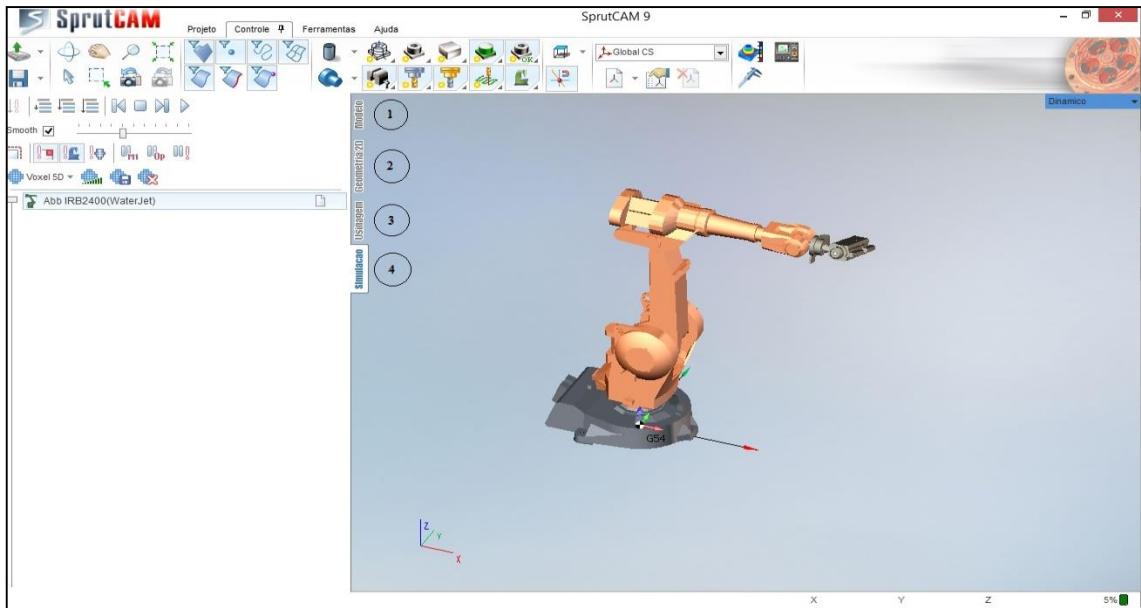


Figura 3.3 - Interface Sprutcam

- Importação de modelos gerados em vários *softwares* CAD e em diferentes formatos.
- Possibilidade de usar e criar diferentes robôs e ferramentas.
- Parametrização pormenorizada dos programas de maquinagem.
- Simulação virtual do programa gerado.
- Detecção de colisões.
- Edição e configuração de pós-processadores.
- Gestão da biblioteca de ferramentas

3.5 Mastercam X5/Robotmaster

O Mastercam é um *software* CAD/CAM genérico de larga utilização, para o qual existe uma aplicação dedicada à programação de robôs industriais, Robotmaster. Embora à data da realização deste trabalho já exista o Mastercam X8 e o Robotmaster V6 no mercado, a versão utilizada foi o Mastercam X5/Robotmaster X5. A funcionalidade CAD deste programa permite a criação de modelos 3D, porém para a criação de modelos com geometrias mais complexas não é tão expedito como um programa específico para CAD (por exemplo SolidWorks ou Inventor). O *add-on* Robotmaster disponibiliza diversos pós-processadores que permitem programar robôs de vários fabricantes. Repare-se que neste *software* não se vê o robô enquanto se cria o programa de maquinagem (Figura 3.4), sendo só possível a sua visualização aquando da simulação. O uso deste *software* permite ao utilizador definir vários parâmetros e usufruir de várias funcionalidades, nomeadamente:

- Importação de modelos gerados em vários *softwares* CAD e em diferentes formatos.
- Possibilidade de usar e criar robôs e ferramentas.
- Parametrização pormenorizada do programa de maquinagem.
- Otimização do movimento de cada eixo do robô.
- Simulação virtual do programa gerado.
- Edição e configuração de pós-processadores.
- Gestão da biblioteca de ferramentas

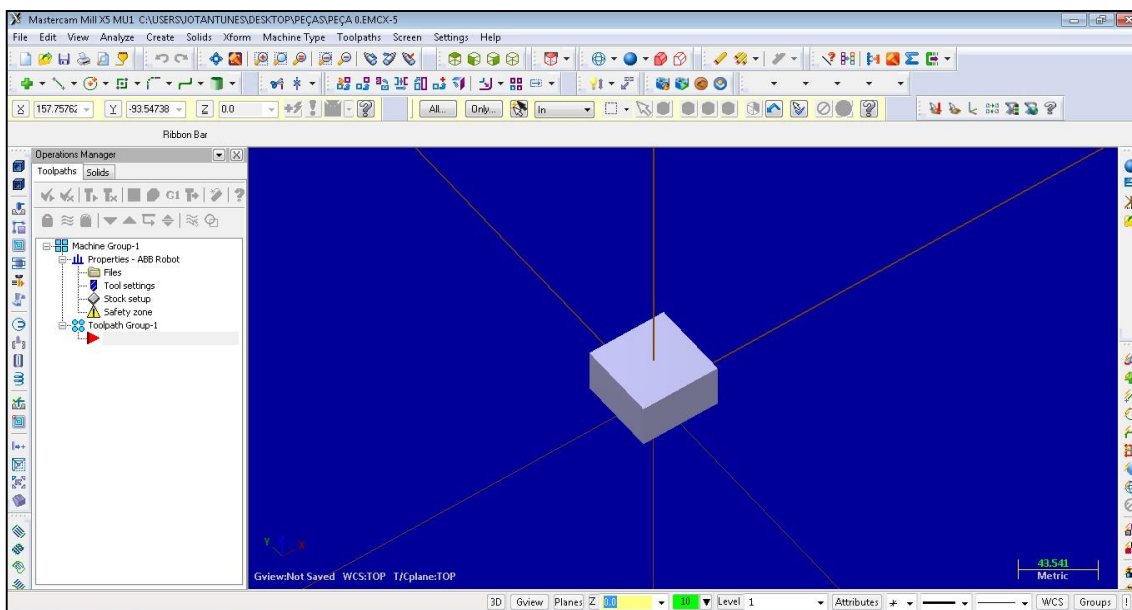


Figura 3.4 - Interface Mastercam

3.6 RobotStudio/Machining PowerPac

O RobotStudio é um *software* proprietário da ABB para a programação de robôs deste fabricante. Segundo o fabricante, este *software* é o mais indicado para a programação *off-line* dos seus robôs. Isto deve-se ao facto de o *software* apresentar uma cópia do controlador real, denominado RobotWare com o qual o RobotStudio é sincronizado. Desta maneira existe a garantia que o controlador real irá executar da mesma maneira as simulações executadas no controlador virtual. Este *software* dispõe de um *add-in*, o Machining PowerPac, para a criação de operações de maquinagem. Porém este *software* apresenta funcionalidades limitadas de CAM quando comparado aos restantes *softwares* utilizados neste trabalho, sendo este vocacionado para operações de acabamento superficial (polimento e rebarbagem). A interface deste *software* é a apresentada na Figura 3.5.

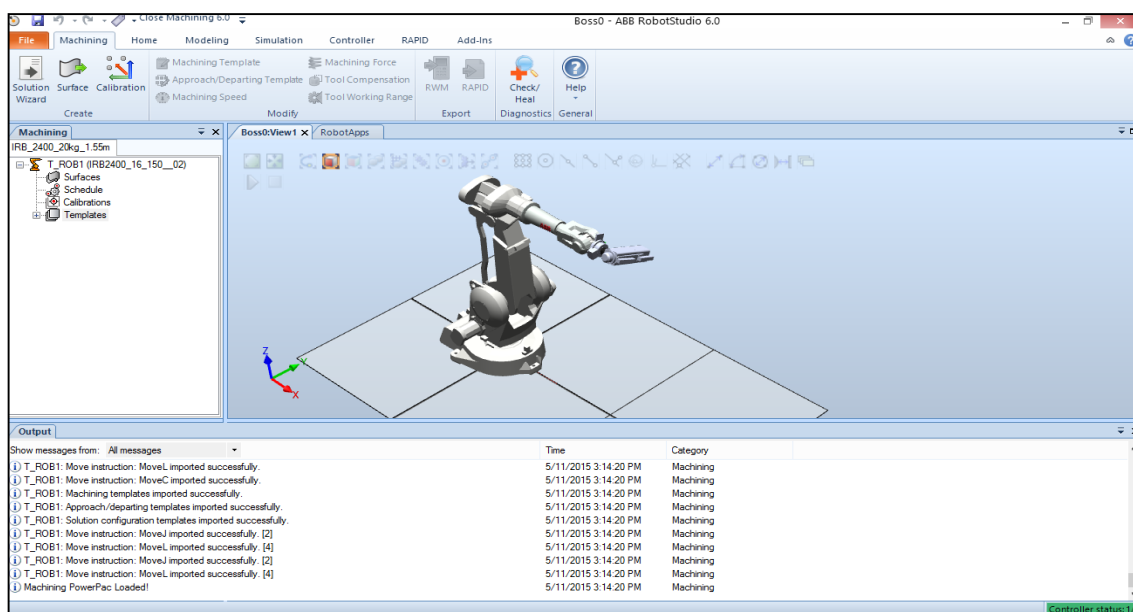


Figura 3.5 - Interface RobotStudio/Machining PowerPac

A utilização deste *software* permite a:

- Importação de modelos gerados em vários *softwares* CAD e em diferentes formatos.
- Usar vários robôs da ABB.
- Detecção de colisões.
- Simulação virtual do programa gerado.
- Verificação de programas em linguagem RAPID.
- Conversão de um programa em código G&M para RAPID (funcionalidade CAM Converter).

3.7 Análise comparativa

Com base na informação disponível referente a cada *software* e após a utilização dos mesmos, é possível identificar as principais funcionalidades (apresentadas na Tabela 5) que cada aplicação informática disponibiliza.

Na funcionalidade CAD, apenas são reportadas as informações disponibilizadas pelos *softwares* uma vez que, para a modelação das peças utilizou-se um *software* mais poderoso, o SolidWorks. O Sprutcam, em relação aos restantes *softwares* utilizados, é o único que não permite a criação de sólidos, pelo que se for necessário o utilizador realizar a modelação da peça pretendida, há que utilizar um *software* CAD externo. No entanto, é de destacar a capacidade de importação de diferentes tipos de ficheiros que cada *software* proporciona, quer seja por importação direta, quer seja após a utilização de alguma funcionalidade, ou programa associado, que permita a conversão para o tipo de ficheiro pretendido.

É pertinente para o utilizador saber quais as funcionalidades que cada *software* disponibiliza desde a criação ou seleção de robôs, gestão de bibliotecas de ferramentas, definição das operações de maquinagem, até à geração do código na linguagem do robô. Assim, são estes os parâmetros abordados na comparação realizada entre as diferentes aplicações informáticas. Embora o RobotStudio seja vocacionado para operações de acabamento (polimento e rebarbagem) e não contenha na sua biblioteca ferramentas para executar operações de fresagem, é possível criar ferramentas e gerar programas com essa finalidade. Contudo, este *software* não permite a criação de operações de desbaste de forma expedita. Relativamente aos pós-processadores, tanto o Sprutcam como o Mastercam permitem a integração de novos pós-processadores de diferentes fabricantes de robôs, algo que o *software* RobotStudio/Machining PowerPac não permite, sendo utilizado exclusivamente na programação de operações de maquinagem dos robôs da ABB.

Também se realizou a comparação em relação aos módulos de simulação que cada *software* providencia, destacando a capacidade que todos apresentam, de deteção de colisões. Uma funcionalidade bastante interessante num produto destes é a possibilidade de o utilizador ter uma representação da peça final (após todas as operações de maquinagem terem sido realizadas). Uma vez que o *software* RobotStudio/Machining PowerPac não permite a criação/parametrização do bloco de trabalho inicial, impossibilita a visualização da peça final.

Tabela 5 - Análise comparativa

	Sprutcam	Mastercam/ RobotMaster	RobotStudio/ Machining PP
Funcionalidade CAD			
Criação de geometrias	✓	✓	✓
Criação de sólidos	✗	✓	✓
Importação de diferentes formatos CAD	✓	✓	✓
Exportação de diferentes formatos CAD	✗	✓	Limitado
Funcionalidade CAM			
Biblioteca de robôs	✓	✓	Apenas robôs da ABB
Criação de robôs	✓	✓	✗
Parametrização do robô	✓	✓	✓
Parametrização do bloco de trabalho inicial	✓	✓	✗
Operações 2D de desbaste e acabamento	✓	✓	Limitado a operações de acabamento
Operações 3D de desbaste e acabamento	✓	✓	
Operações 5D de desbaste e acabamento	✓	✓	
Biblioteca de ferramentas	✓	✓	✗
Criação e parametrização de ferramentas	✓	✓	✓
Pós-processador para diferentes linguagens	✓	✓	Apenas Linguagem RAPID
Módulo de Simulação			
Visualização da trajetória gerada	✓	✓	✓
Deteção de colisões	✓	✓	✓
Visualização da remoção de material	✓	✓	✗

4 Casos de estudo de programação de robôs para operações de maquinagem

Este capítulo apresenta o processo de criação de programas de maquinagem, recorrendo ao uso de três aplicações informáticas distintas: Mastercam/Robotmaster, Sprutcam e RobotStudio/Machining PowerPac. Antes de ser possível criar o programa que contém a linguagem ou codificação dos movimentos do robô, é necessário modelar a célula virtual de modo a que esta seja a mais próxima possível da célula real. Para compreender melhor este processo de preparação e criação dos vários programas são apresentados os vários passos a seguir nos próximos subcapítulos. De referir que a estratégia adotada, considerando a utilização do robô e equipamentos presentes no laboratório de robótica, segue as etapas ilustradas na Figura 4.1.

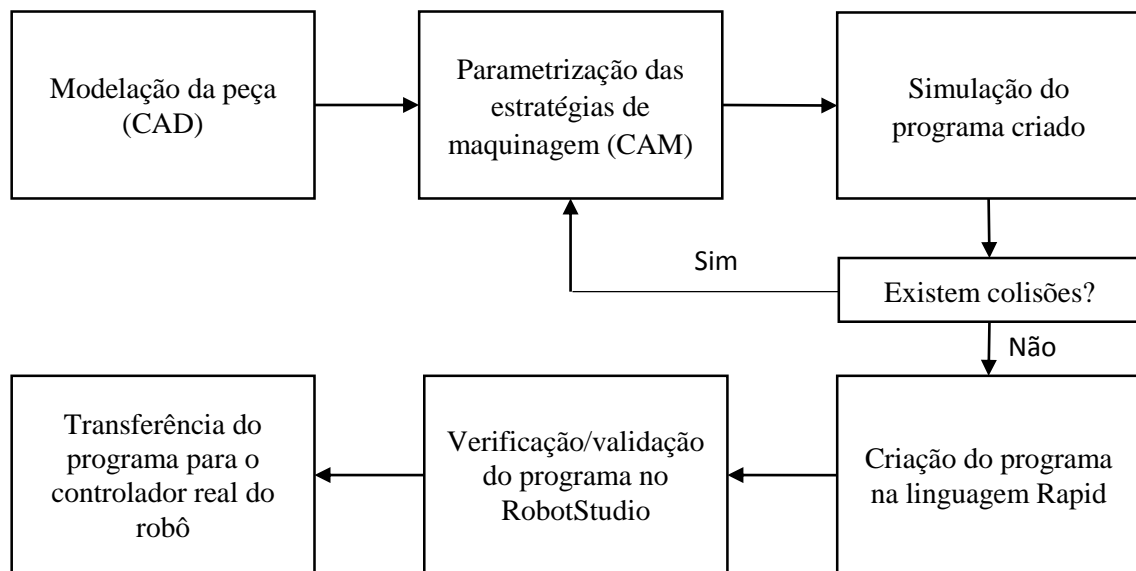


Figura 4.1 - Esquema da estratégia adotada

Para a modelação das peças o *software* utilizado foi o SolidWorks. Realizada a modelação transferem-se os ficheiros CAD para o *software* CAM onde são criados os vários programas de maquinagem e executadas as respetivas simulações. Após a verificação da inexistência de colisões, ou de qualquer outro erro que comprometa o processo de maquinagem, utiliza-se o pós-processador que cada *software* disponibiliza para gerar o programa na linguagem RAPID. Uma vez que a simulação no *software* proprietário será a mais próxima possível da realidade, por possuir todos os componentes da célula real na sua biblioteca, é realizada uma simulação neste *software* para validação do programa gerado. Após este último teste, o programa fica disponível para ser transferido para o controlador do robô.

4.1 Modelação da célula robótica

Para a utilização das diferentes aplicações de *softwares* utilizados neste trabalho procedeu-se à modelação geométrica da célula robótica existente no laboratório, sendo incluído os diversos componentes utilizados, tais como o motor-árvore (*Spindle*) e as ferramentas. Esta modelação é necessária para que seja possível tirar partido da funcionalidade de identificação de colisões que os diferentes programas possuem, assim como para ter uma melhor representação da célula real. A “ferramenta” que o robô movimenta é constituída por diversos componentes que foi necessário modelar. Os componentes são: o suporte da célula de carga, a célula de carga, adaptador para mudança de ferramenta automática e o *Spindle*. Com o objetivo de realizar um correto dimensionamento dos diferentes componentes utilizou-se os desenhos disponibilizados pelos respetivos fabricantes e posteriormente efetuou-se uma medição experimental para verificação e confirmação dos valores.

O resultado final da modelação da ferramenta movimentada pelo robô é apresentado na Figura 4.2, seguindo a respetiva ordem: suporte para célula de carga, célula de carga, adaptador para mudança automática de ferramenta e *Spindle*.

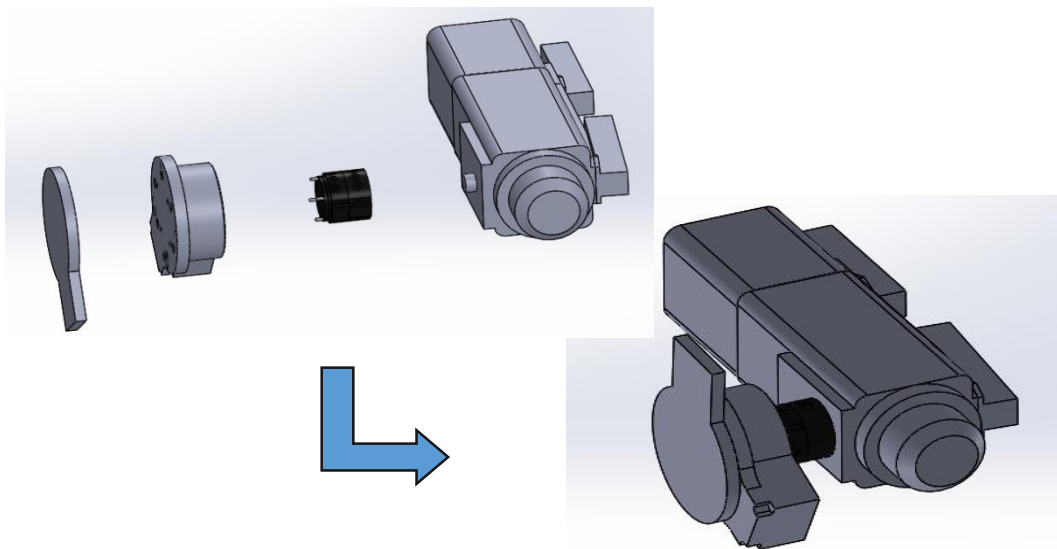


Figura 4.2 - “Ferramenta” a montar no elemento terminal do robô

O conjunto modelado em ambiente SolidWorks foi transferido para cada um dos 3 *softwares* CAM, usando-se diferentes formatos intermédios:

- *.SAT para utilização no RobotStudio
- *.STL e devida configuração no ficheiro de definições (ficheiro em formato *.XML) para utilização no Mastercam

A transferência para o *software* Sprutcam 9 é um caso particular em que não é necessário converter o ficheiro num determinado formato, isto porque após a instalação do *software* CAM (Sprutcam 9) é criada uma funcionalidade no *software* CAD (SolidWorks) que permite a passagem do modelo criado automaticamente. No entanto, é necessário realizar o correto posicionamento e orientação do modelo no ficheiro de definições (ficheiro em formato *.XML).

Um aspeto fundamental na modelação da ferramenta é a definição do referencial da ferramenta e respetivo *Tool Center Point* (TCP). A posição e orientação deste referencial é definido em relação ao referencial *tool0* (Figura 4.3). No entanto este precisa de ser validado quando a ferramenta se encontra montada no robô. A definição geométrica e a calibração experimental realizada, permitiu obter os valores apresentados na Tabela 6, para o referencial da ferramenta a utilizar. Estes valores foram obtidos através da consola do robô que permite aceder a uma funcionalidade para proceder à calibração da ferramenta utilizando o método dos 4 pontos. Sucintamente este método consiste em colocar a extremidade da ferramenta numa posição de referência fixa e de seguida reorientar o robô mantendo o mesmo ponto de referência. Este procedimento é repetido 4 vezes realizando assim a identificação do TCP.

Tabela 6 - Posicionamento e orientação do TCP em relação ao referencial *tool0*

X	-80.7942 mm
Y	-190.124 mm
Z	172.813 mm
Q1	0.692911404
Q2	0.692911404
Q3	-0.140974418
Q4	-0.140974418

(a orientação do referencial da ferramenta é expressa por quatérnios)

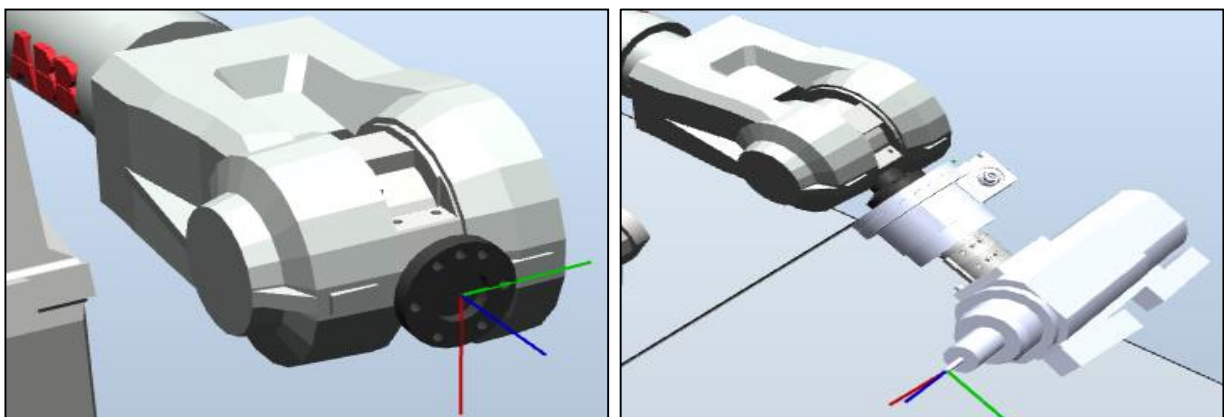


Figura 4.3 - *Tool0* (esquerda) e TCP segundo o referencial *tool0* (direita); em ambiente RobotStudio

4.1.1 Modelação do robô no Sprutcam

O programa Sprutcam tem uma biblioteca bastante extensa no que diz respeito a máquinas ferramentas CNC, mas o mesmo não acontece referentemente a robôs. A versão do *software* utilizado apenas disponibiliza 4 modelos de robôs industriais: ABB *IRB 2400 L*, Fanuc *M710 IC*, Kuka *KR150* e o Stäubli *RX170hsm*. Sendo o robô disponível no Laboratório de Robótica o ABB *IRB 2400*, que é uma versão distinta do *IRB 2400 L*, foi necessário proceder à sua modelação. O ABB *IRB 2400* apresenta uma estrutura semelhante à do *IRB 2400 L* mas possui características físicas distintas relativamente a distâncias entre eixos e limites de rotação de eixos. A modelação do robô *IRB 2400* no ambiente do *software* Sprutcam envolveu as seguintes três etapas:

1 - Construção do modelo de sólidos do robô em SolidWorks.

A partir dos modelos dos principais componentes do robô foi criado um “*assembly*” tendo em consideração a estrutura cinemática do robô. As partes constituintes do robô são disponibilizadas pela ABB em vários formatos, sendo utilizado nesta dissertação o formato *.IGES. O resultado final do modelo articulado do robô, apresentado na Figura 4.4, é gravado em formato proprietário do SolidWorks (*.SLDASM).

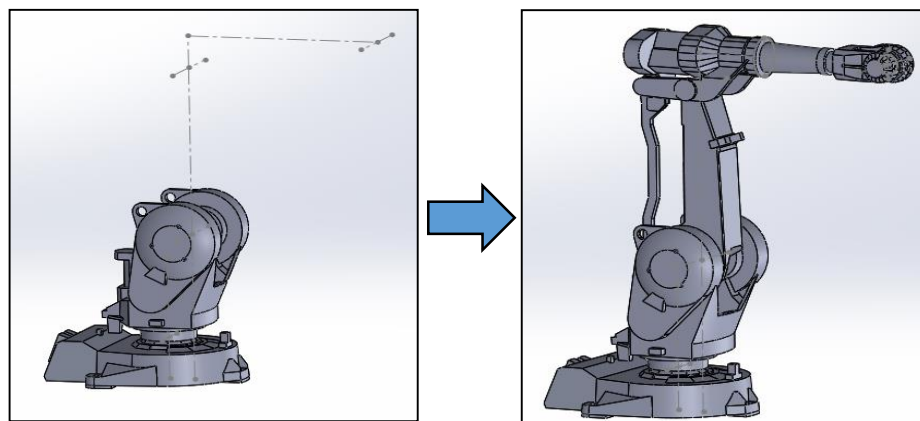


Figura 4.4 - Representação do robô em ambiente SolidWorks

2 - Conversão do modelo para formato compatível com a utilização do Sprutcam.

A conversão é realizada recorrendo às funcionalidades do *software* Sprutcam, sendo guardado um conjunto de ficheiros em formato *.OSD. Estes ficheiros contêm a representação dos diferentes componentes constituintes do robô.

3 - Configuração das definições do modelo do robô em ambiente Sprutcam.

Neste ficheiro é necessário definir a estrutura cinemática do robô, sendo especificado distâncias entre eixos, limites de rotação de cada eixo, posição e orientação das ferramentas. Ainda neste ficheiro é indicada a localização dos ficheiros que contêm os modelos físicos dos componentes do robô e do *Spindle* a utilizar (Anexo B).

Na Figura 4.5 é possível visualizar o modelo do robô ABB *IRB 2400* criado para a elaboração do trabalho.

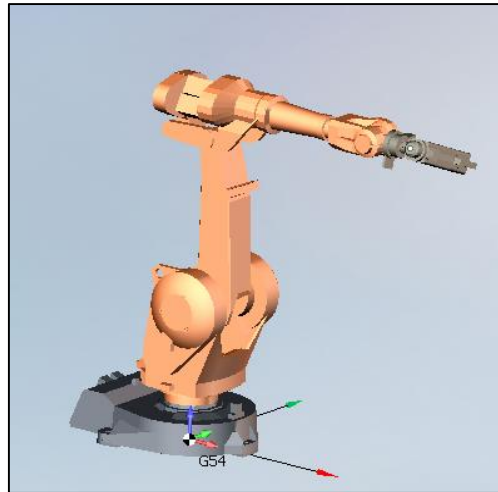


Figura 4.5 - IRB 2400

4.2 Análise preliminar de estratégias de maquinagem para operação de facejamento

Para comparar as três diferentes alternativas de programação de robôs consideradas, foi escolhida uma operação elementar de facejamento. O objetivo é testar para uma operação simples, maquinagem 2D ou no plano, analisando movimentos lineares e circulares da ferramenta de modo a mais facilmente identificar diferenças nos programas obtidos. A peça, de forma paralelepípedica e dimensões 50 mm x 50 mm x 30 mm (Figura 4.6) é posicionada no espaço de trabalho do robô e respeitando os seguintes valores, em relação à origem do robô: X= 1000 mm; Y= 0 mm; Z= 550 mm.

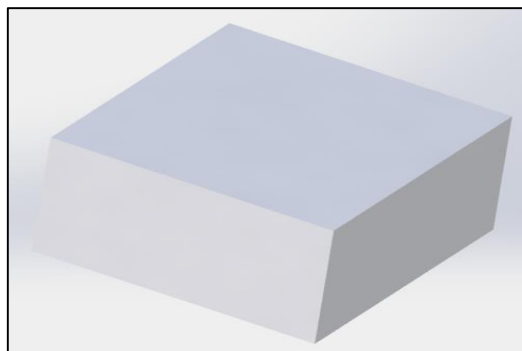


Figura 4.6 - Peça 0

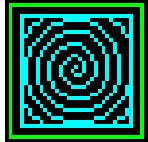
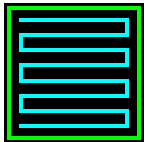
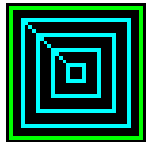
Uma vez que a presente análise tem como objetivo identificar algumas dissimilaridades que possam existir nos programas gerados em *softwares* distintos, optou-se por utilizar velocidades típicas de corte e de avanço para um material (alumínio). Na Tabela 7 são apresentados os valores utilizados para o presente teste, com base na informação disponibilizada pela HSM Works e que respeita os intervalos definidos pelo Machinery's Handbook [16].

Tabela 7 - Parâmetros de corte

Velocidade do <i>Spindle</i>	9703	RPM
Velocidade de Avanço	983	mm/min
Stepover (XY)	4.8	mm
Ø Fresa	6	mm
Nº de lâminas	2	---
Comprimento da Lâmina	30	mm
Comprimento da Fresa	100	mm

A velocidade de aproximação à peça em todas as simulações, ponto de início e final da operação de desbaste são idênticas. As estratégias de maquinagem selecionadas são as apresentadas na Tabela 8, que ilustram percursos baseados em movimentos lineares e circulares.

Tabela 8 - Estratégias de maquinagem

Estratégia	Adaptativa	Paralela	Equidistante
Representação			

Para o RobotStudio apenas é considerada a estratégia “Paralela”, uma vez que é a única estratégia disponibilizada pelo mesmo que pode ser diretamente comparável com as outras alternativas.

Para cada um dos *softwares* são gerados os programas em RAPID para a movimentação do robô. Estes, por sua vez, são utilizados no *software* RobotStudio para simular e analisar o desempenho do robô ao executar a estratégia/trajetória selecionada. Para tal, é utilizada uma funcionalidade, denominada “*Signal Analyzer*”, que permite para cada estratégia de maquinagem obter os seguintes parâmetros:

- Tempo de maquinagem: apenas é quantificado o tempo em que a ferramenta se encontra em corte.
- Velocidade média: velocidade linear da ferramenta expressa no referencial da peça.
- Potência máxima: potência instantânea do conjunto dos motores do robô.
- Energia consumida: energia despendida pelos motores do robô.

Na Tabela 9 são apresentados os resultados para a estratégia de maquinagem “Paralela”.

Tabela 9 - Resultados de maquinagem da Peça 0 (Estratégia Paralela)

Funcionalidade Signal Analyzer	RobotStudio	Sprutcam	Mastercam
Tempo de maquinagem [s]	29	40	9
Velocidade média [mm/s]	19.9	16	72.2
Potência máxima [W]	91	40	250
Energia Consumida [J]	2379	924	1653
Instruções de movimentação	22 MoveL	25 MoveL	24 MoveL

Após a obtenção destes resultados é necessário compreender certas disparidades. Uma vez que os parâmetros de maquinagem são iguais é espectável que os dados obtidos sejam no mínimo semelhantes. Ao analisar os resultados da Tabela 9 verifica-se que os tempos de maquinagem nos três programas utilizados são muito diferentes, devido a velocidades também diferentes. Para perceber a origem desta desigualdade é necessário analisar os programas gerados na linguagem RAPID.

Instruções básicas de movimento na linguagem RAPID

Nesta linguagem o robô pode ser programado com os seguintes tipos de movimentos [30]:

- **MoveJ** – interpolação ao nível de juntas: implica o movimento independente das juntas do robô de modo a que, quando a posição de destino é atingida, a paragem do movimento dos eixos ocorra em simultâneo.
- **MoveL** – interpolação linear: o TCP move-se descrevendo uma trajetória linear.
- **MoveC** – interpolação circular: o TCP move-se descrevendo uma trajetória circular.

Uma instrução de posicionamento contém a seguinte informação:

- Tipo de interpolação de movimento (apresentado anteriormente).
- A posição de destino que o robô deve atingir (*target*).
- Velocidade de execução do movimento.
- Dimensão da zona, qual o grau de aproximação ao ponto de destino que o robô deve executar antes de começar a mover-se para a posição seguinte. Por exemplo, se se escolhe “*fine*”, o robô move-se exatamente para o ponto de destino antes de executar o próximo movimento.
- Ferramenta ativa (TCP ativo).

Na Figura 4.7 é apresentado um exemplo de uma instrução de movimentação.

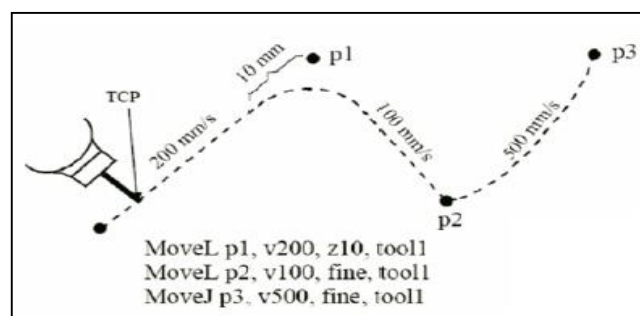


Figura 4.7 - Exemplo de movimentação

Para a trajetória representada na Figura 4.7 é necessário definir três instruções de movimento e posicionamento, como é possível observar. Desta maneira é possível compreender a definição de dimensão de zona (Z10 e fine), assim como a velocidade de movimentação que é expressa em mm/s e o tipo de movimento que é realizado pelo TCP dependendo da instrução MoveL ou MoveJ.

Analisando os códigos gerados torna-se claro que embora os parâmetros de maquinagem definidos sejam semelhantes, os pós-processadores não geram exatamente os mesmos códigos. Seguidamente serão evidenciadas algumas diferenças nos códigos gerados:

- **Diferença no peso do *Spindle* e centro de massa.** No código gerado pelo Sprutcam é atribuído ao *Spindle* um peso de 0.001 Kg enquanto que no Mastercam o peso é de 20 Kg e no RobotStudio o peso é de 12 Kg. Há que alterar as configurações dos programas para uniformizar estes valores ou editar os programas gerados, para que traduzam o valor efetivo das características do *Spindle*. Os valores obtidos experimentalmente com recurso à funcionalidade “*Machining Force Control*”, disponível no controlador real da célula robótica, indica que o *Spindle* tem um peso de 11.4 Kg e que o seu centro de gravidade é de: X= 18.9 mm; Y= 47.7 mm; Z= 124.1 mm
- **Ponto inicial da trajetória.** Analisando o primeiro movimento linear (*MoveL*) de maquinagem (visível na Figura 4.8) é possível verificar que o posicionamento do TCP é diferente.

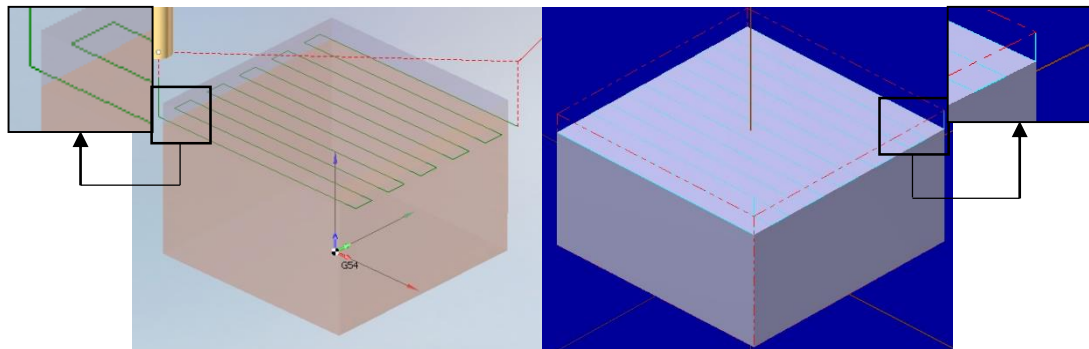


Figura 4.8 - Trajetória gerada pelo Sprutcam e Mastercam

Após a modificação do tipo de compensação da ferramenta na trajetória gerada pelo *software* Sprutcam é possível obter uma trajetória semelhante à do Mastercam.

- **Velocidades geradas pelos pós-processadores.** Recorde-se que a velocidade de avanço (ou velocidade do TCP) foi definida nos dois programas para 983 mm/min ou 16 mm/s. Ao examinar o código gerado é possível observar que os valores definidos para a velocidade também são diferentes: Sprutcam: [16,30,200,15] (Anexo C); Mastercam: [983.00,500,5000,1000]. Na instrução *MoveL* os pós-processadores de cada *software* definem a velocidade por um vetor de quatro componentes que representam respetivamente a velocidade do TCP, reorientação da ferramenta, velocidade linear dos eixos exteriores e velocidade de rotação dos eixos exteriores. Sendo a única variável de interesse para o presente caso de estudo a velocidade do TCP, torna-se evidente a origem do erro, isto é, embora no *software* Mastercam esteja bem definida a velocidade de avanço,

o pós-processador do RobotMaster não realiza a conversão de minutos para segundos. Em relação ao RobotStudio a definição do valor da velocidade utiliza valores pré-definidos pelo *software*.

- **Definição de dimensão de zona.** A definição do parâmetro *zone* a utilizar nas instruções de movimentação é feita de modo distinto nos dois *softwares*. No Mastercam/RobotMaster é possível parametrizar este valor, independentemente para as instruções do tipo *MoveJ*, *MoveL* e *MoveC*, enquanto que no Sprutcam tal funcionalidade não existe. Neste ultimo a definição do parâmetro *zone* é incorporada numa variável da linguagem Rapid do tipo *zonedata*, representada por um vetor de sete componentes, associado a cada instrução de movimentação, do tipo [FALSE,3,40,40,10,35,5]. No presente caso, sendo o primeiro elemento deste vetor “*false*” significa que o valor atribuído ao segundo elemento corresponde à utilização de uma aproximação de vetor equivalente a *Zone 3*, valor não parametrizável neste *software*. Contudo é possível editar o valor deste parâmetro, mas tem de ser feito manualmente no programa gerado. No caso do RobotStudio/Machining PowerPac a definição de zona também utiliza valores pré-definidos pelo programa.

Assim para poder efetuar uma comparação efetiva entre diferentes estratégias geradas por pós-processadores distintos, foi necessário proceder a uma harmonização dos códigos gerados, redefinindo os pontos abordados anteriormente e definindo a velocidade de avanço em 16 mm/s e parâmetro *Zone Z3* (Anexo C). Para efeito de comparação considerou-se apenas o percurso de maquinagem, tendo-se ignorado os movimentos de aproximação inicial e retorno final. Desta maneira foi possível obter os resultados apresentados nas tabelas 10, 11 e 12 para as diferentes estratégias de maquinagem.

Tabela 10 - Estratégia Paralela (após edição dos programas RAPID)

Funcionalidade Signal Analyzer	RobotStudio	Sprutcam	Mastercam
Tempo de maquinagem [s]	39	40	39
Velocidade média [mm/s]	16	16	16
Potência máxima [W]	80	61	103
Energia Consumida [J]	2625	2221	3569
Instruções de movimentação	22 MoveL	25 MoveL	24 MoveL

Tabela 11 - Estratégia Equidistante (após edição dos programas RAPID)

Funcionalidade Signal Analyzer	Sprutcam	Mastercam
Tempo de maquinagem [s]	39	39
Velocidade média [mm/s]	16	16
Potência máxima [W]	58	81
Energia Consumida [J]	2105	2852
Instruções de movimentação	41 MoveL	34 MoveL

Tabela 12 - Estratégia Adaptativa (após edição dos programas RAPID)

Funcionalidade Signal Analyzer	Sprutcam	Mastercam
Tempo de maquinagem [s]	41	57
Velocidade média [mm/s]	16	16
Potência máxima [W]	59	74
Energia Total Consumida [J]	2250	4190
Instruções de movimentação	570 MoveL	11 MoveL 28 MoveC

Ao analisar os resultados é possível constatar três pontos mais relevantes:

- 1- A velocidade do TCP em todas as simulações realizadas atingiram os valores pretendidos.
- 2- A energia consumida no processo de maquinagem é dependente da estratégia adotada sendo que para estratégias semelhantes o programa gerado com o Sprutcam conduz a um menor consumo energético.

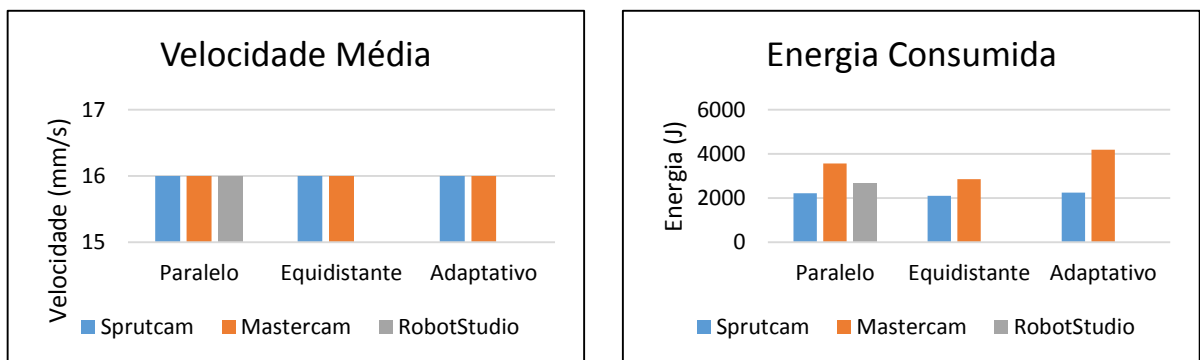


Figura 4.9 - Velocidade média e energia consumida nos percursos de maquinagem

- 3- O tamanho do código gerado pelo pós-processador do Sprutcam é sempre maior (mais instruções de movimentação) do que o código gerado pelo pós-processador do Mastercam/RobotMaster. O tipo de estratégia adotada condiciona o tipo de instruções de movimentação a utilizar no programa RAPID. Verifica-se que a estratégia “adaptativa” do Mastercam é a única que leva à utilização de interpolações circulares, o que permite reduzir o número de instruções utilizadas. A estratégia semelhante do Sprutcam apenas utiliza instruções de movimentação do tipo linear, fazendo com que o programa gerado seja, naturalmente mais extenso. Na Figura 4.10 é possível verificar que as trajetórias são efetivamente diferentes.

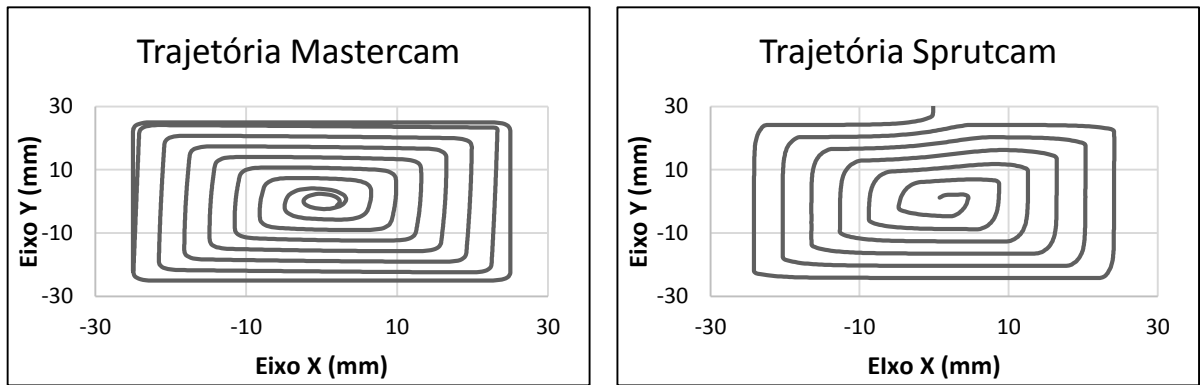


Figura 4.10 - Percurso de maquinagem resultante da estratégia adaptativa

4.3 Peças desenvolvidas

Para uma melhor compreensão das funcionalidades que cada programa proporciona, foram criados três outros modelos de peças para maquinar. O objetivo de usar peças de geometrias distintas é o de ilustrar algumas funcionalidades e tipos de maquinagem de 2, 3 e 5 eixos. Desta maneira é possível verificar as diferenças existentes na criação de um programa de maquinagem recorrendo à utilização de *software* distinto.

Peça 1

A peça representada na Figura 4.11 foi criada para a utilização de maquinagem de 2 eixos. A fresagem nestes casos consiste em apenas maquinar as peças segundo um plano. O eixo Z é usado para posicionar a ferramenta em profundidade. As geometrias diferentes em cada nível servem para demonstrar qual a resposta que cada *software* proporciona desde a criação do programa de maquinagem e geração do código RAPID até à qualidade final da peça. A peça tem uma dimensão máxima externa de 100 x 100 x 70 mm.

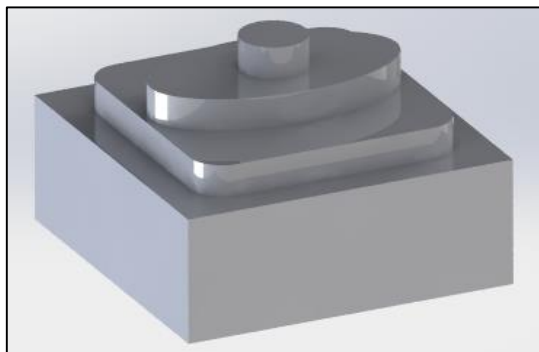


Figura 4.11 - Peça 1

Peça 2

O objetivo desta peça (Figura 4.12) é o permitir avaliar as funcionalidades de cada *software* em operações de desbaste e acabamento de superfícies mais complexas. Este modelo apresenta características geométricas que permitem a maquinagem 3D. A obtenção da superfície curva requer o movimento simultâneo em XYZ. As dimensões externas máximas desta peça são 100 x 100 x 40 mm.

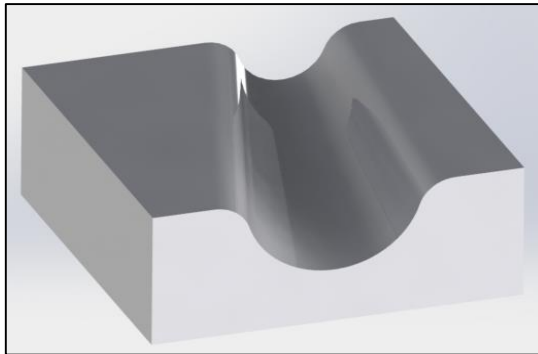


Figura 4.12 - Peça 2

Peça 3

A peça 3 foi desenhada para demonstração de maquinagem a 5 eixos. Este modelo permite ilustrar as maiores dificuldades na programação das operações de maquinagem a 5 eixos, uma vez que se pretende realizar o acabamento com a ferramenta sempre orientada segundo a direção normal em cada ponto da superfície. As dimensões máximas externas desta peça são 240 x 200 x 74 e o modelo é representado na Figura 4.13.

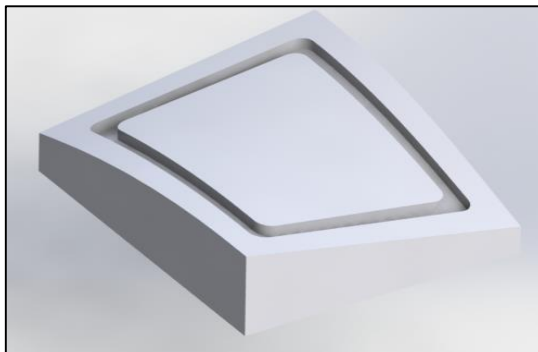


Figura 4.13 - Peça 3

Para a criação de um programa de maquinagem é necessário seguir independentemente do *software* utilizado, as seguintes etapas:

- Etapa 1 – Modelação da peça e importação para o *software* CAM.
- Etapa 2 – Escolha da máquina (neste caso o robô *IRB 2400* da ABB) e definição da geometria do bloco de material inicial.
- Etapa 3 – Seleção e parametrização das operações de maquinagem e da ferramenta.
- Etapa 4 – Verificação através de simulação.
- Etapa 5 – Geração do código em RAPID.

As etapas 1 e 2 serão pontos abordados apenas num caso de estudo em cada *software*, porque o procedimento é semelhante para todos os casos de estudo. Os parâmetros de corte para a criação dos programas são os apresentados na Tabela 13. As velocidades seleccionadas têm em conta o estudo realizado por [31], onde foram executados testes a diferentes velocidades utilizando uma fresa semelhante à que irá ser utilizada na implementação real e um bloco de trabalho também semelhante (Poliuretano expandido).

Tabela 13 - Parâmetros de corte

	Operações de desbaste	Operações de acabamentoo
<i>Stepdown</i>	6 mm	2 mm
<i>Stepover</i>	4.8 mm	1.2 mm
Velocidade do <i>Spindle</i>	10000 RPM	
Velocidade de corte	188 mm/min	
Velocidade de avanço	50 mm/s	
Ø Fresa	6 mm	

4.4 Programação com Sprutcam

Peça 1

Etapa 1 – Modelação da peça e importação para o *software* CAM

Após a criação dos modelos no programa SolidWorks é necessário fazer a passagem para os respetivos programas de CAM. Atendendo a que o Sprutcam é parceiro de desenvolvimento do SolidWorks, não é necessário proceder à gravação das peças modeladas num formato específico. Assim, o Sprutcam permite a incorporação de ficheiros em formato nativo do SolidWorks. Após a importação do modelo é necessário proceder ao seu correto posicionamento. Na Figura 4.14 é ilustrado o procedimento necessário a seguir: primeiro posicionar no zero do referencial utilizado e depois posicionar corretamente para se proceder às operações de maquinagem. A numeração apresentada na Figura 4.14 diz respeito aos seguintes conteúdos:

- 1- Árvore dos modelos criados e em utilização.
- 2- Faces seleccionadas em função de 1.
- 3- Posicionamento da peça em relação ao referencial pré-definido.
- 4- Visualização do modelo, assim como a previsão da sua posição final.

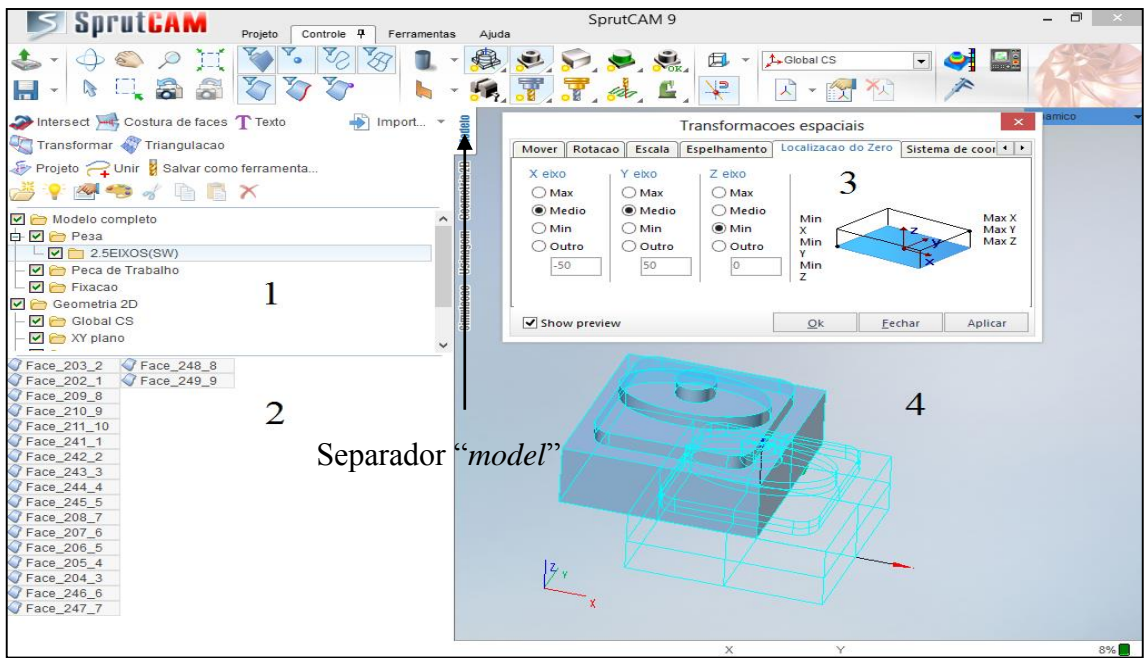


Figura 4.14 - Interface após importação do modelo

Uma vez realizado o correto posicionamento e determinado o “zero” do modelo é necessário posicionar a peça em relação ao referencial do robô. Para tal seleciona-se o separador *machining* e as funcionalidades da interface mudam. O posicionamento do modelo é agora alterado em relação ao robô seguindo os dois passos representados na Figura 4.15.

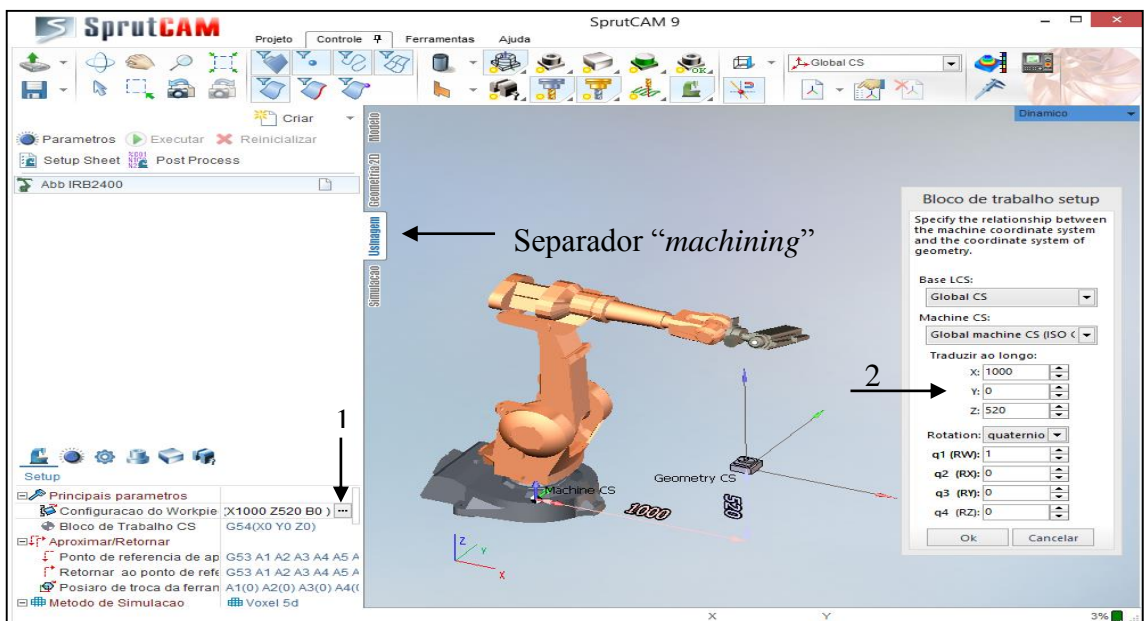


Figura 4.15 - Posicionamento do *Workobject* em relação ao robô (Sprutcam)

Etapa 2 - Escolha da máquina e definição da geometria do bloco de material inicial

Realizado o correto posicionamento do modelo, é necessário selecionar o robô a utilizar (Figura 4.16).

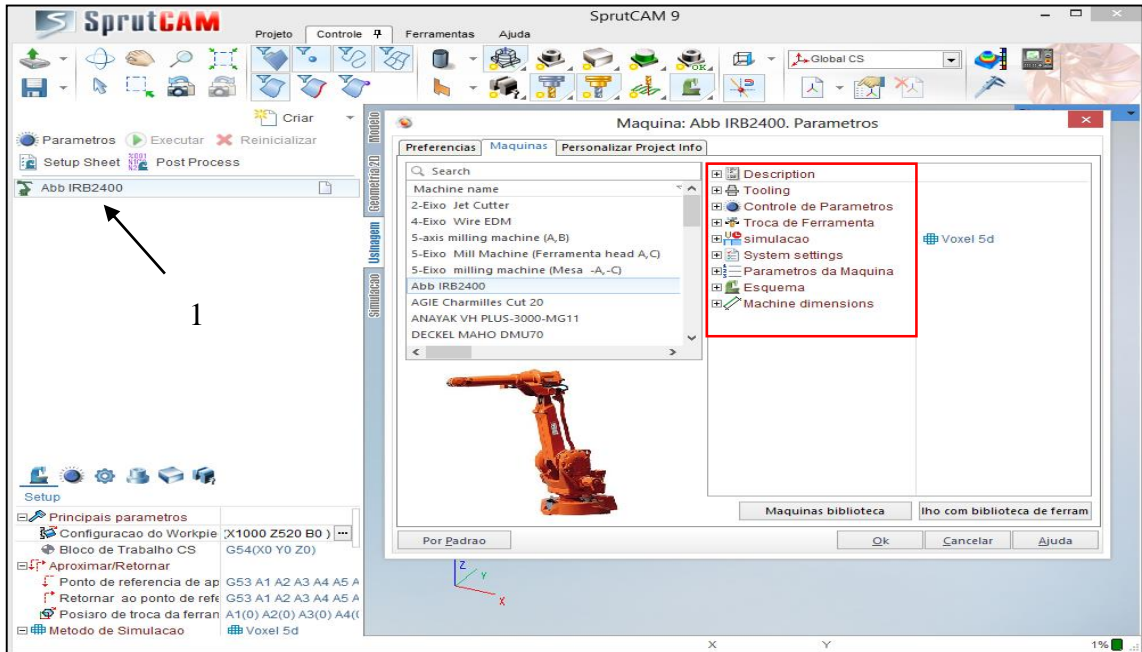


Figura 4.16 - Seleção da máquina a utilizar (Sprutcam)

Neste menu é possível mudar alguns parâmetros pré-definidos (quadrado a vermelho), como por exemplo a orientação do motor-árvore ou mudar os limites de rotação de cada eixo. A escolha feita é visível na Figura 4.16 e também é possível verificar o robô escolhido na árvore de trabalho (ponto 1 da mesma figura). Nesta fase da programação é necessário definir a geometria do bloco de material a maquinar. É selecionado o modo “around part” o que cria um bloco como representado na Figura 4.17 a sombreado.

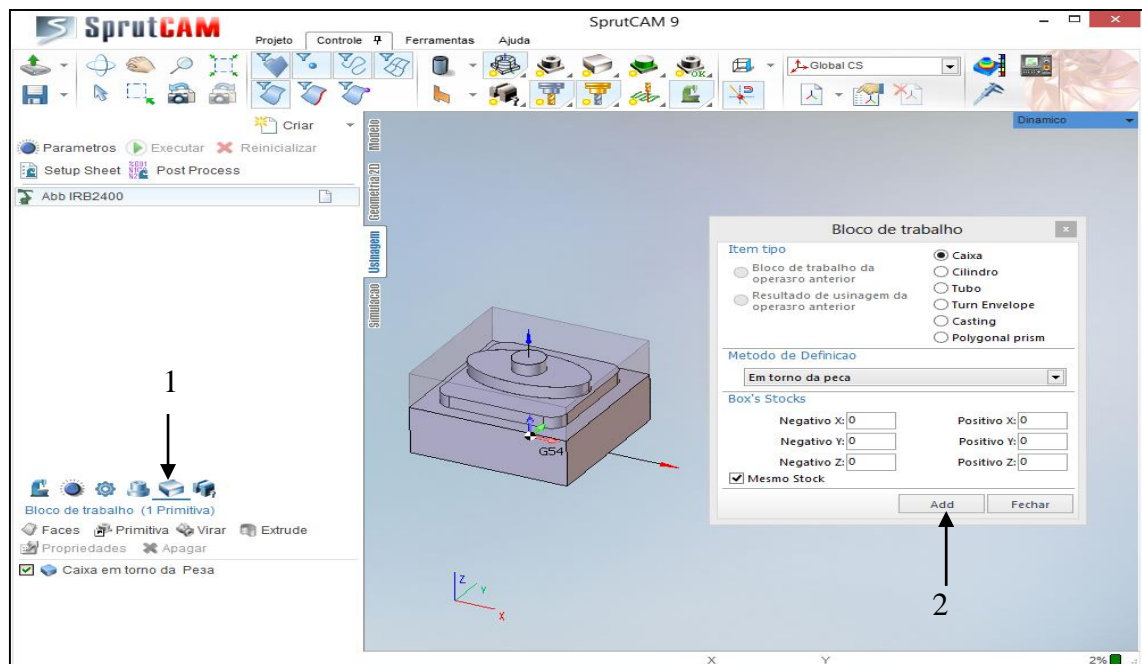


Figura 4.17 - Geometria do bloco de trabalho (Sprutcam)

Etapa 3 – Seleção e parametrização das operações de maquinagem

Ao selecionar qualquer operação de maquinagem é possível aceder a um menu bastante completo no que diz respeito à parametrização da operação selecionada. Neste menu é possível também aceder à parametrização da ferramenta a utilizar (Figura 4.18).

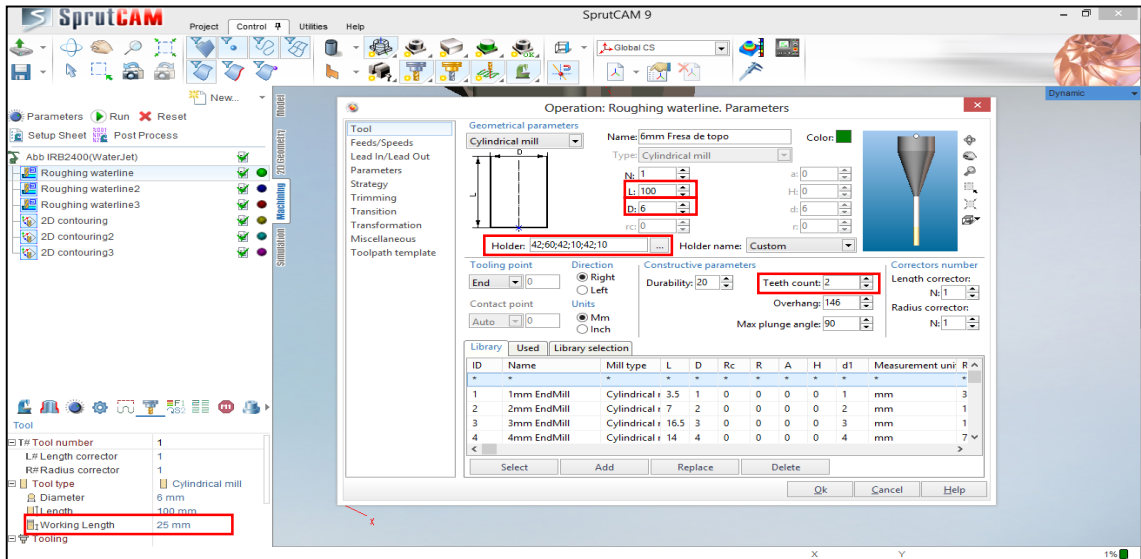


Figura 4.18 - Parametrização da ferramenta (Sprutcam)

Como se pode verificar, este *software* apresenta uma biblioteca bastante extensa no que diz respeito a ferramentas. Os retângulos a vermelho na Figura 4.18 mostram os parâmetros modificados para a criação da ferramenta pretendida. Esta pode ser utilizada posteriormente desde que seja gravada na biblioteca.

Neste momento é possível criar as operações de maquinagem. No presente caso as operações de desbaste e de acabamento selecionadas foram “*Roughing Waterline*” e “*2D Contouring*”, respetivamente. De salientar que ao selecionar este tipo de operação a ferramenta apenas tem movimento segundo um plano XY em cada passagem de maquinagem. Na Figura 4.19 é ilustrado a seleção das faces para a operação de desbaste (a azul) assim como a sequência para gerar a trajetória da ferramenta.

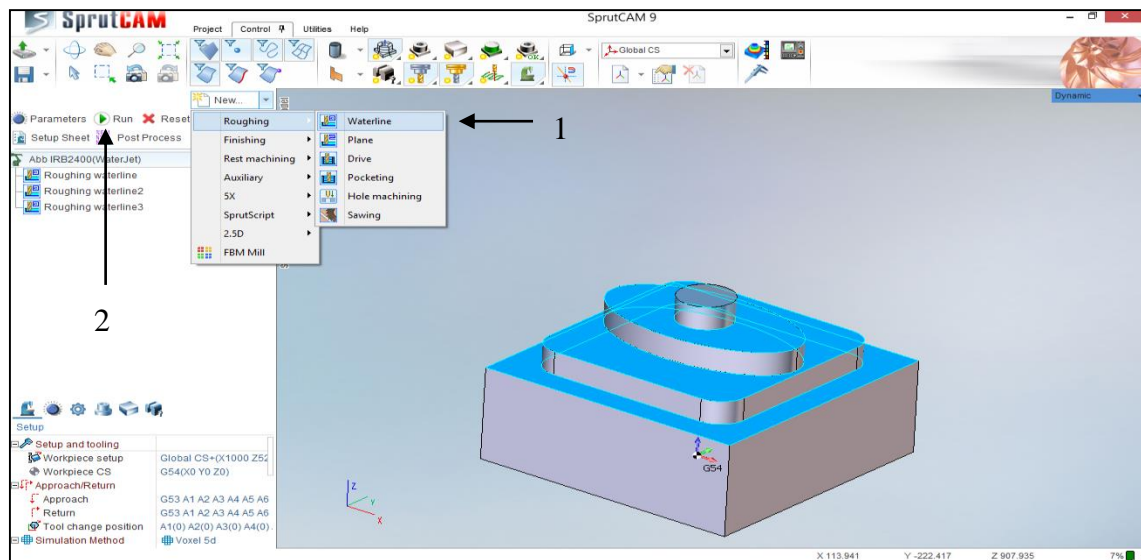


Figura 4.19 - *Roughing Waterline* (Sprutcam)

Embora seja possível seleccionar as três faces para maquinagem, o *software* Sprutcam não permite gerar automaticamente as três operações de faceamento de uma forma correta. Usando este caso de estudo, é possível visualizar que ao seleccionar as três faces são criadas na árvore de trabalho três operações, sendo a primeira referente ao nível superior da peça. No entanto, aquando da criação do programa a segunda e a terceira operação não são realizadas, porque contêm erros (Figura 4.20).

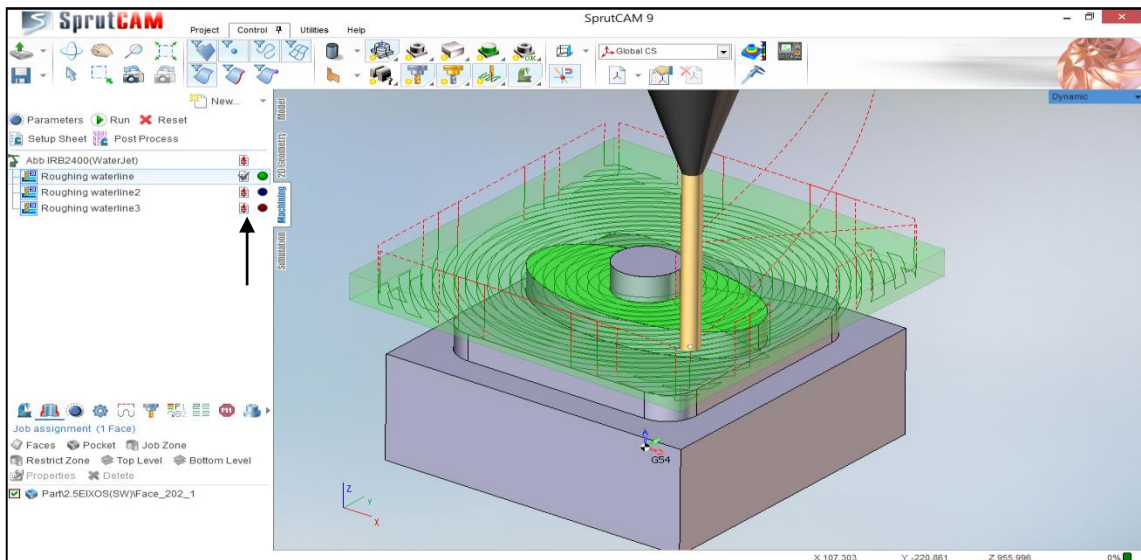


Figura 4.20 - Erro na criação da trajetória (Sprutcam)

A ocorrência deste erro deve-se ao facto do *software* definir alguns parâmetros automaticamente para a primeira operação e replicar esses parâmetros para as operações seguintes (no caso de seleccionar várias faces). Neste caso a razão pela qual não é possível criar o trajeto da ferramenta deve-se ao facto desta ter o seu movimento limitado para um intervalo definido da coordenada Z. A Figura 4.21 ilustra o menu correspondente à parametrização da operação de debaste. De referir que este menu é diferente consoante a operação seleccionada. A correção deste parâmetro passa por o utilizador definir o intervalo de valores para a coordenada Z que a ferramenta pode adotar.

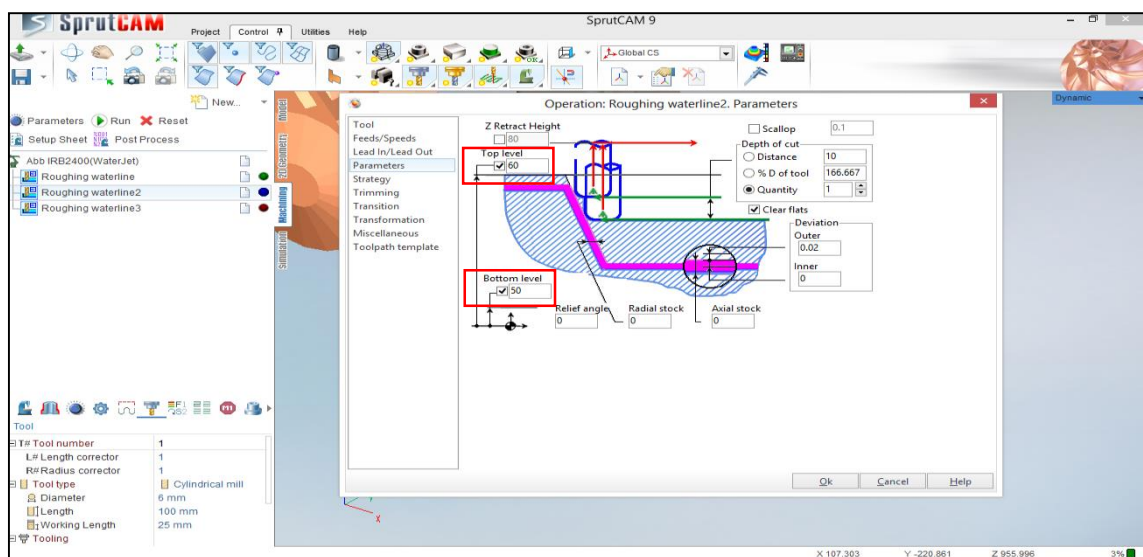


Figura 4.21 - Correção do valor da coordena Z (Sprutcam)

Após a criação da operação de desbaste, com sucesso, é criada a operação de acabamento (a utilizada foi a “2D Contouring”).

Etapa 4 – Verificação através de simulação

Concluída a fase de parametrização, é possível realizar a simulação das várias operações. Sendo uma simulação gráfica permite a identificação da origem de uma potencial irregularidade ou colisão. Neste caso de estudo foi possível verificar que não ocorriam problemas. Também neste módulo do *software* a presença de um ponto de exclamação a vermelho à frente de cada operação significa que existem erros. Caso não exista nenhuma irregularidade aparece um “visto a verde” (Figura 4.22). Também é possível prever o material que não foi removido ou o material removido em excesso.

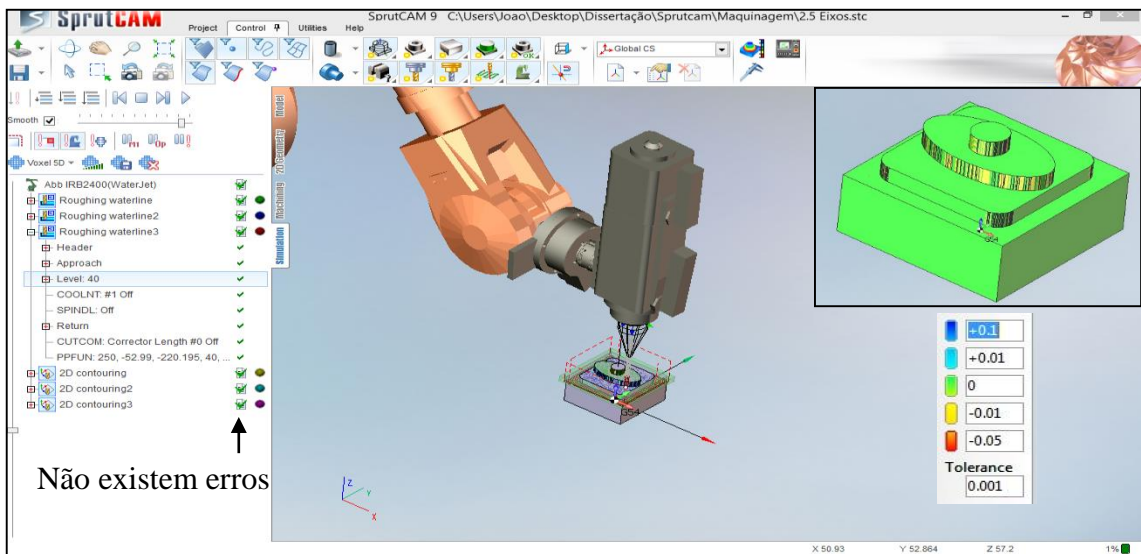


Figura 4.22 - Simulação do primeiro modelo (Sprutcam)

Etapa 5 – Geração do código

Este *software* permite a geração automática de um programa em linguagem Rapid. Para tal é necessário utilizar a funcionalidade “*Post Process*” e selecionar o pós processador pretendido (neste caso da ABB). O programa gerado é guardado num ficheiro, ficando disponível para ser utilizado diretamente no controlador do robô.

Peça 2

A peça 2 foi selecionada para constituir um exemplo de maquinagem a 3 eixos.

Etapa 3 - Seleção e parametrização das operações de maquinagem e da ferramenta

A operação de desbaste utilizada nesta peça foi a “*Roughing Waterline*”. Para as operações de acabamento foram utilizadas duas operações distintas. A primeira é o facejamento lateral, onde se utilizou a operação “2D Contouring”. A trajetória das operações geradas são apresentadas na Figura 4.23.

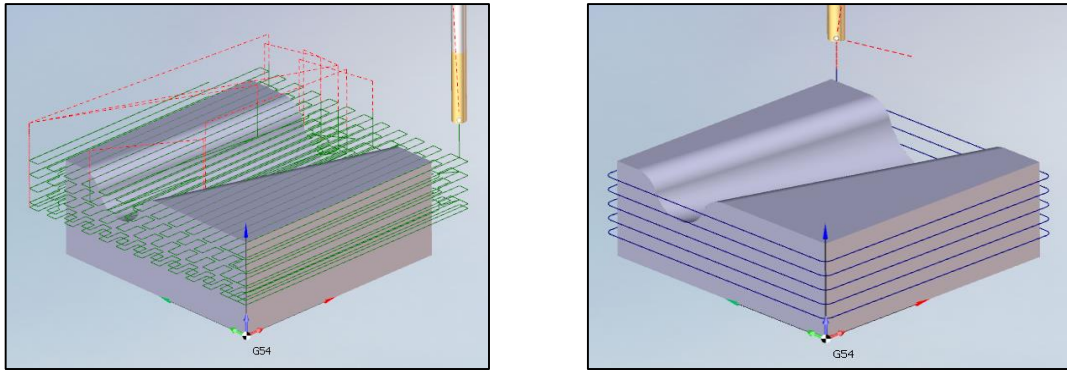


Figura 4.23 - Roughing Waterline e 2D Contouring

A operação de acabamento para a superfície curvilínea utilizada foi a “*Finishing Plane*”, que permite a utilização de 3 eixos em simultâneo na maquinagem. Selecionando a face superior obtém-se o percurso de maquinagem apresentado na Figura 4.24.

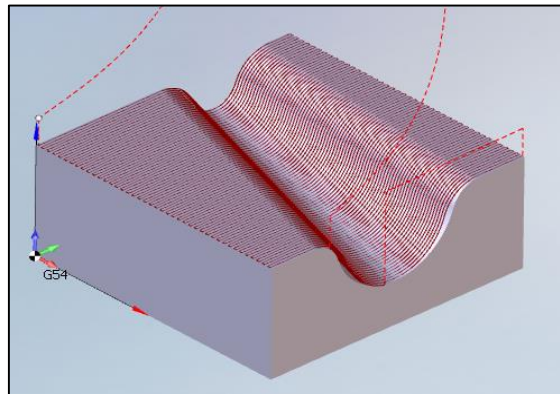


Figura 4.24 - Finishing Plane

Após a execução da etapa 4, onde se realiza a simulação virtual, foi possível verificar que trajetória criada não continha nenhuma colisão. Seguidamente utilizou-se a funcionalidade “*Post Process*” e gerou-se o programa na linguagem RAPID, completando assim a etapa 5.

Peça 3

A peça 3 é a mais complexa pois requer o uso de maquinagem de 5 eixos.

Etapa 3 - Seleção e parametrização das operações de maquinagem e da ferramenta

Para a primeira operação, que é de desbaste, selecionou-se “*Roughing Drive*” que permite a utilização de 3 eixos em simultâneo. A trajetória gerada é apresentada na Figura 4.25.

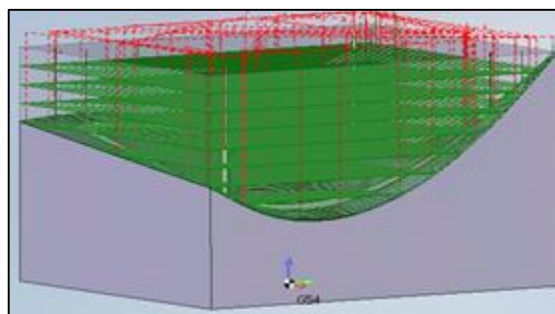


Figura 4.25 - Roughing Drive (Sprutcam)

Neste modelo as operações de acabamento são executadas recorrendo à maquinagem a 5 eixos. A operação selecionada foi a “5D Contouring”. Ao selecionar esta operação é de extrema importância definir a orientação da ferramenta. Existem três modos distintos: normal à superfície, paralela à superfície ou com um ângulo fixo face à superfície (as superfícies referidas são as que estão a ser maquinadas). A orientação selecionada foi a normal à superfície, visível na Figura 4.26.

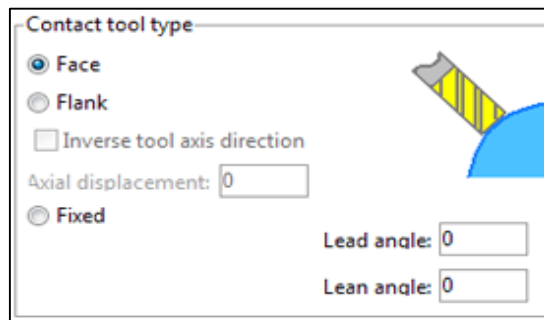


Figura 4.26 - Seleção da orientação da ferramenta (Sprutcam)

Após a seleção das faces a serem maquinadas, foi possível realizar a simulação desta operação e o *software* detetou um erro de colisão entre o *Spindle* e o robô (Figura 4.27).

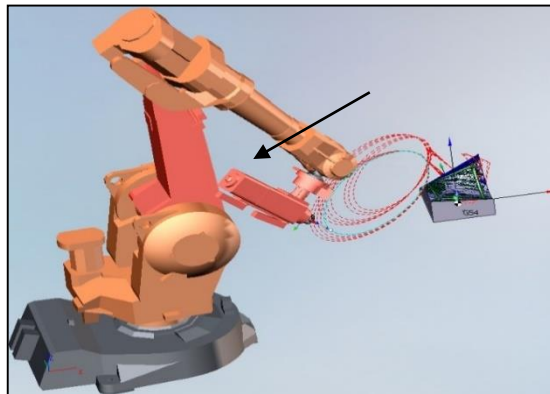


Figura 4.27 - Colisão entre o *Spindle* e o robô

Após uma análise a esta trajetória é possível identificar qual o problema e a causa. A colisão representada na Figura 4.27 é devido a uma rotação excessiva no eixo 4. Esta colisão ocorre no movimento entre as trajetórias geradas. Para solucionar este problema podem ser utilizadas duas estratégias:

- Mudar o tipo de movimento da ferramenta entre os diferentes percursos de maquinagem para que seja possível executar o trajeto sem qualquer colisão.
- Editar os movimentos gerados, o que neste caso passa pela eliminação dos mesmos.

Realizada a simulação verificou-se que ambas as soluções são possíveis. No entanto, a melhor opção é a eliminação dos movimentos desnecessários, o que permite diminuir o tempo de maquinagem.

4.5 Programação com Mastercam/Robotmaster

Após a realização das diversas operações de maquinagem em peças de geometria distintas recorrendo ao *software* Mastercam com o *add-on* RobotMaster foi selecionada a peça 3 para ilustrar a utilização do mesmo.

Peça 3

Etapa 1 - Modelação da peça e importação para o *software* CAM

Um aspeto a ter particular atenção na importação de um modelo de CAD para um programa de CAM diz respeito ao posicionamento/orientação da peça no referencial de base do *software* de CAM. Assim, uma definição/escolha do referencial da peça no *software* CAD pode simplificar o processo de posicionamento no *software* CAM. Contudo, o *software* Mastercam/RobotMaster providencia as ferramentas necessárias para efetuar o reposicionamento das peças importadas. Na Figura 4.28 são apresentadas as várias hipóteses de mudar a posição da peça, do separador “*Xform*”.

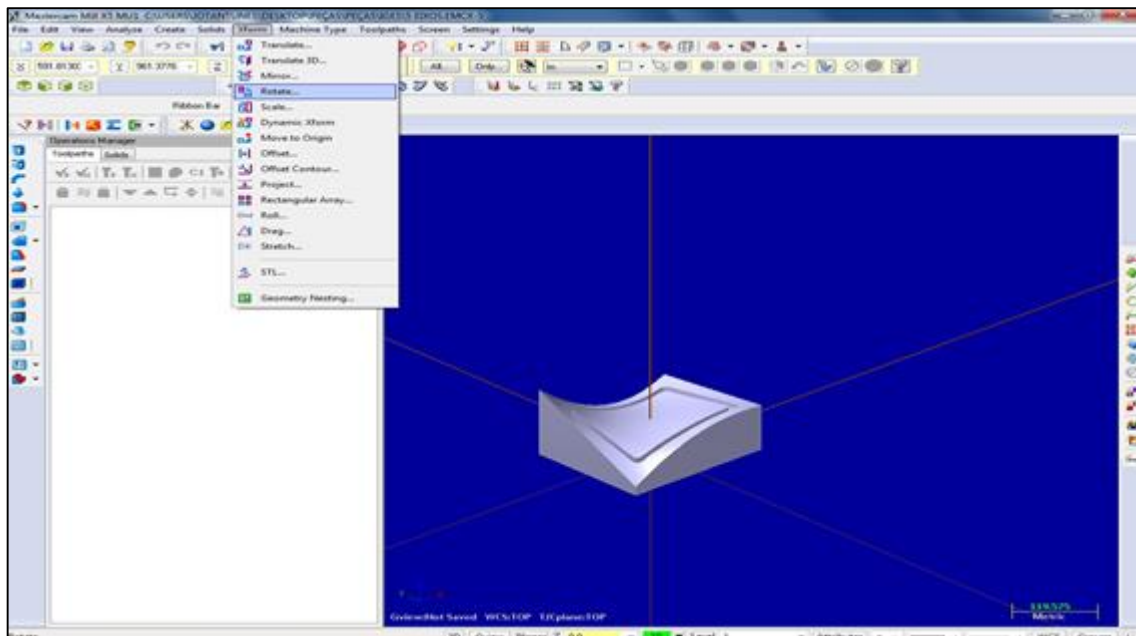


Figura 4.28 - Opções de posicionamento da peça (Mastercam)

Etapa 2 - Escolha da máquina e definição da geometria do bloco de material inicial

A seguir à importação da peça, é necessário selecionar qual a máquina a utilizar. Após a seleção da máquina de trabalho, aparecem as funcionalidades (representado na Figura 4.29) que permitem definir alguns parâmetros tais como as ferramentas a utilizar e o bloco de material a maquinar. Assim como no *software* apresentado anteriormente, a seleção da geometria do bloco de material é idêntica, neste caso com outra designação de “*Bounding box*”.

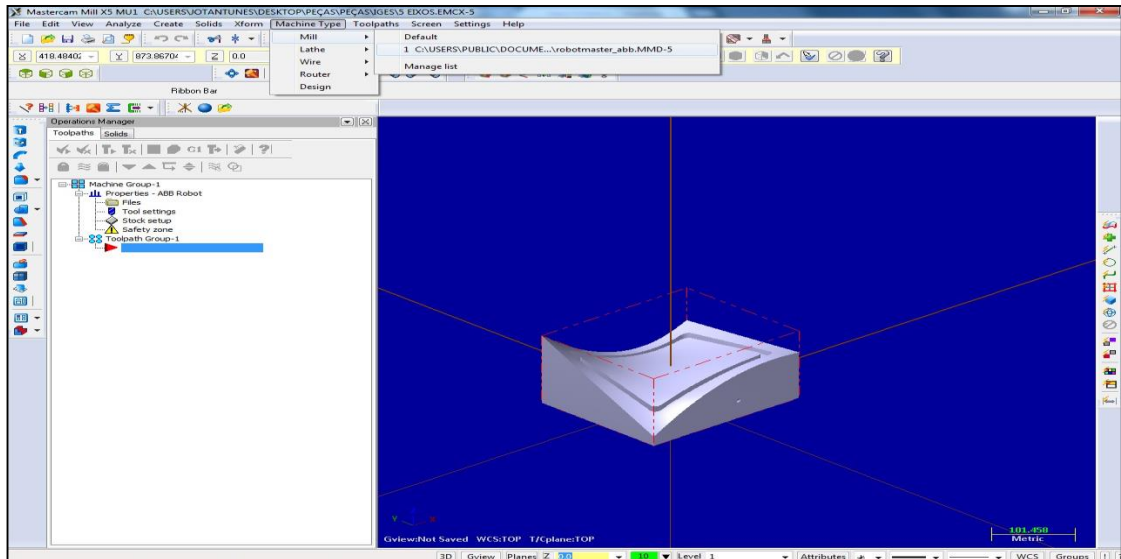


Figura 4.29 - Seleção do robô (Mastercam)

Etapa 3 - Seleção e parametrização das operações de maquinagem

A seleção da ferramenta é realizada a partir da aba “*Toolpath parameters*” que se pode aceder diretamente na árvore de trabalho. Na Figura 4.30 são apresentadas quais as definições possíveis de configurar no que diz respeito às ferramentas. De salientar que este *software* calcula as velocidades em função das características do material selecionado para a ferramenta (possui uma base de dados com as características de vários tipos de material).

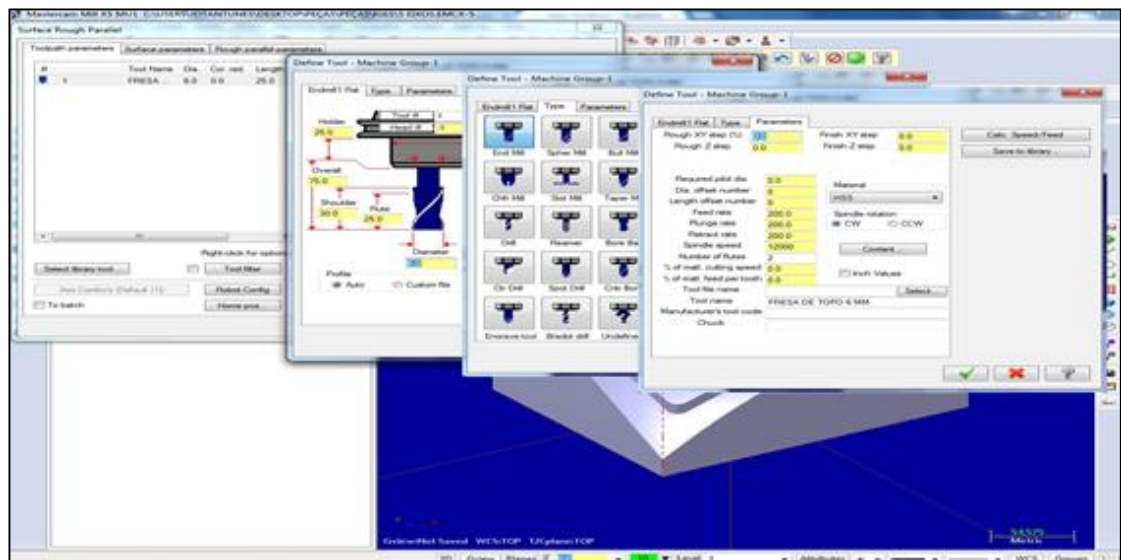


Figura 4.30 - Parametrização da ferramenta (Mastercam)

Para selecionar a operação de maquinagem pretendida é necessário utilizar-se o separador “*Toolpaths*”, representado na Figura 4.31, que disponibiliza as operações possíveis de realizar com este *software*. Foi selecionada a operação de desbaste “*Roughing Surface Flowline*” e iniciado o processo de parametrização desta operação. Neste tipo de operação é necessário definir a superfície condutora (“*Drive Surface*”), superfície limite (“*Check Surface*”) e “*Flowline*” (onde se define a direção do corte) [9]. Após o passo número 3 aparece automaticamente um menu onde é possível definir os parâmetros da operação. Nestes menus é

possível definir a profundidade de corte, o passo, as distâncias de aproximação e recuo, assim como a altura na qual a ferramenta se irá mover entre os movimentos de corte.

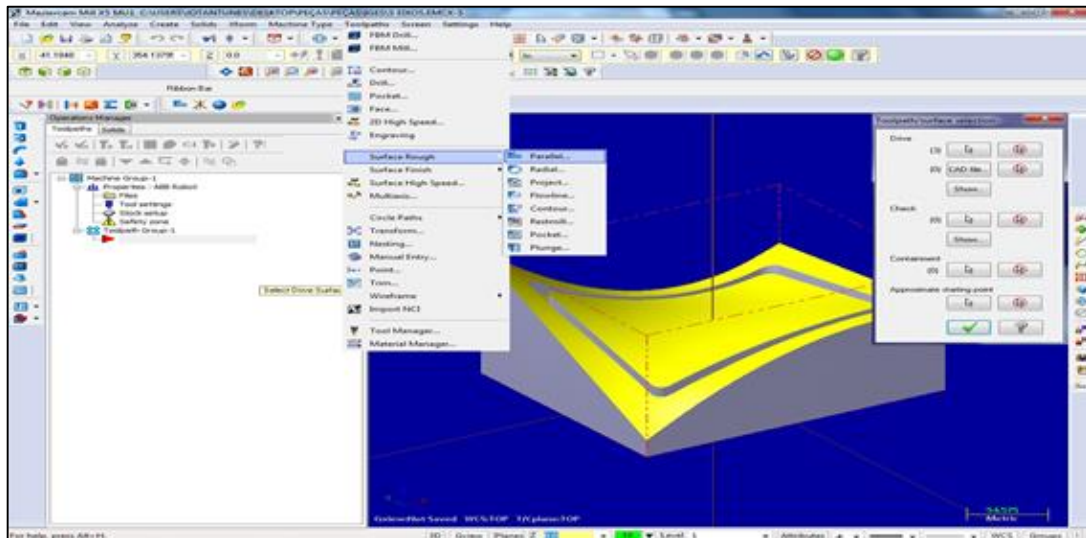


Figura 4.31 - Seleção da operação de desbaste (Mastercam)

Nas configurações do robô no *add-in* Robotmaster (accedidas através do *robot config*) existem dois tipos de definições: globais e locais. As globais, como o próprio nome indica, são utilizadas normalmente para todas as operações e dizem respeito à seleção do modelo robô, ao órgão terminal (neste caso o *Spindle* criado já foi introduzido na base de dados do programa e encontra-se disponível para utilização), posicionamento da peça em relação ao robô e posições de aproximação e retorno. Na Figura 4.32 está representada a seleção do robô e do *Spindle*, assim como o posicionamento da peça e a orientação do *Spindle*.

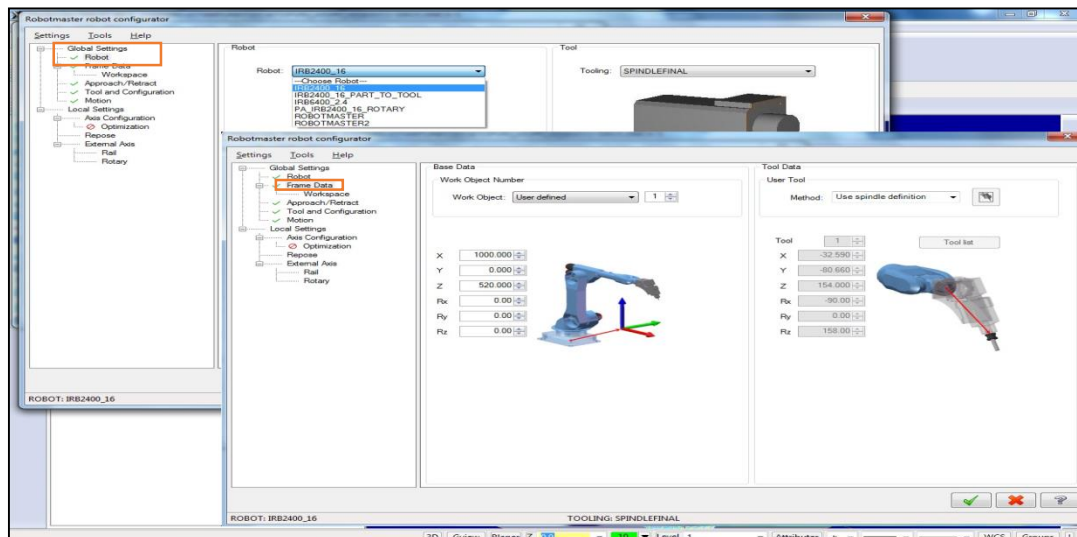


Figura 4.32 - Seleção do robô nas definições do Robotmaster (Mastercam + Robotmaster)

Ainda nestas definições é possível visualizar o volume de trabalho do robô seleccionado assim como o posicionamento da peça e das trajetórias geradas através da funcionalidade “workspace”. Desta maneira é possível ter uma noção do posicionamento relativo peça/robô (Figura 4.33).

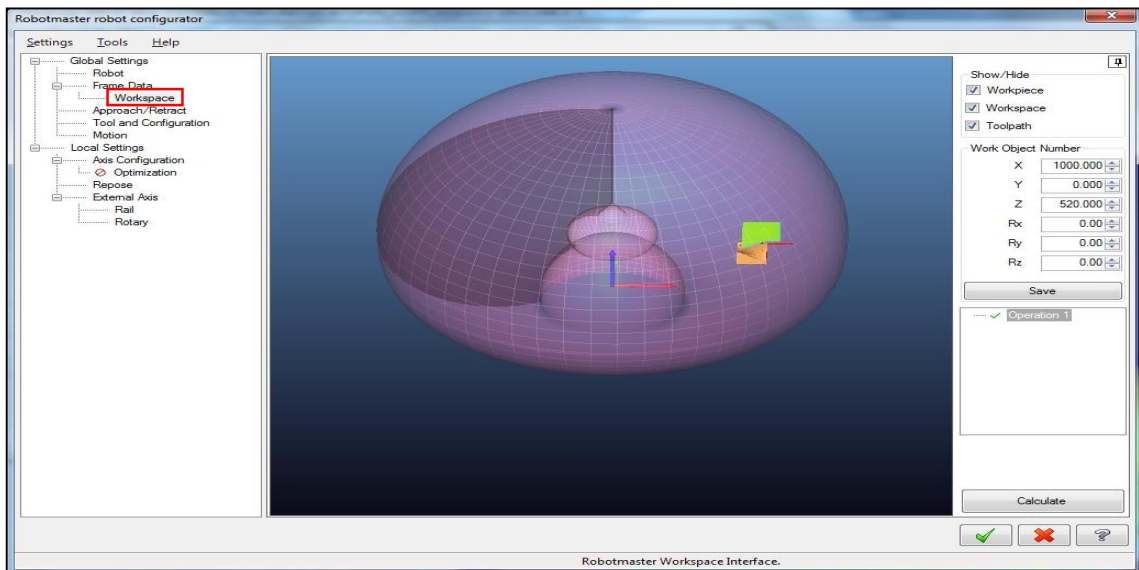


Figura 4.33 - Funcionalidade Workspace (Mastercam/Robotmaster)

Ao contrário das definições globais, as definições locais são referentes apenas à operação em questão. Uma funcionalidade bastante interessante nas definições locais é a “*Optimization*”, que permite visualizar graficamente colisões, singularidades, problemas de alcance do robô ou até limites excedidos de rotação dos eixos.

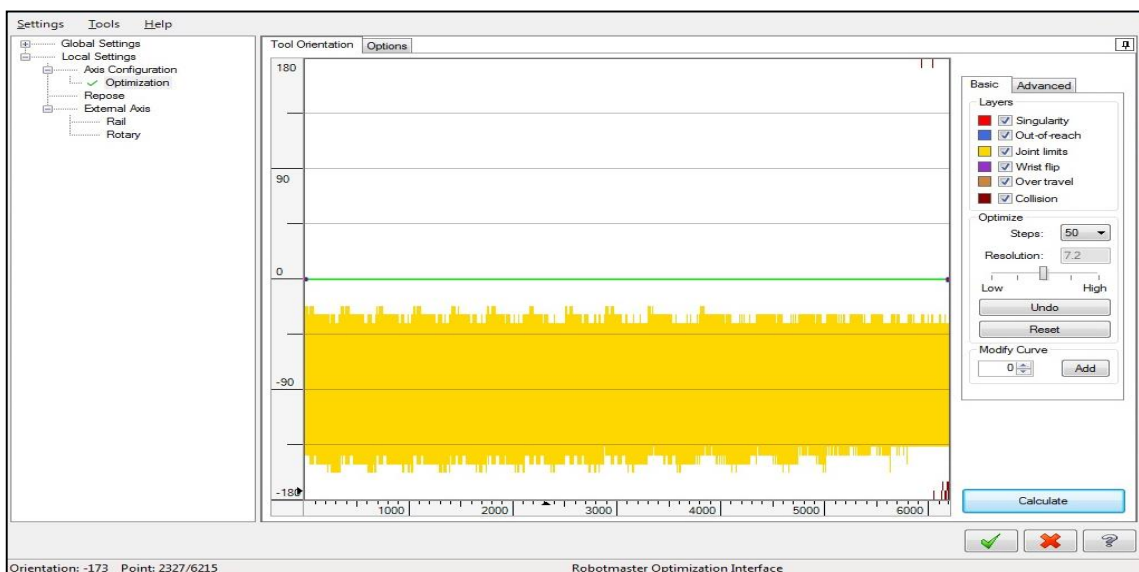


Figura 4.34 - Optimization (Mastercam/Robotmaster)

A linha horizontal a verde na Figura 4.34 representa a trajetória da ferramenta sem modificações nas configurações do robô. Por vezes é possível evitar alguns problemas ao mudar a trajetória da ferramenta. Por exemplo, na Figura 4.35 está representado a amarelo uma colisão. Neste caso é possível alterar as configurações do robô, de uma forma simplificada, de modo a que o robô consiga realizar a trajetória pretendida mas evitando a ocorrência de colisões.

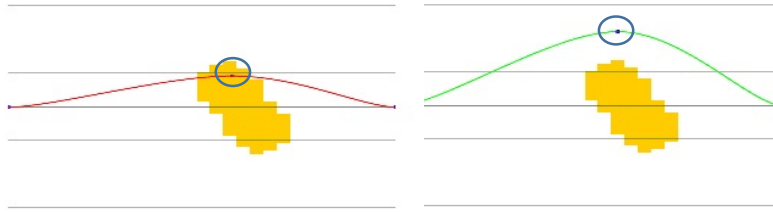


Figura 4.35 - Modificação da trajetória da ferramenta (Mastercam/Robotmaster)

A trajetória gerada para a operação de desbaste, após todas as parametrizações, é apresentada na Figura 4.36.

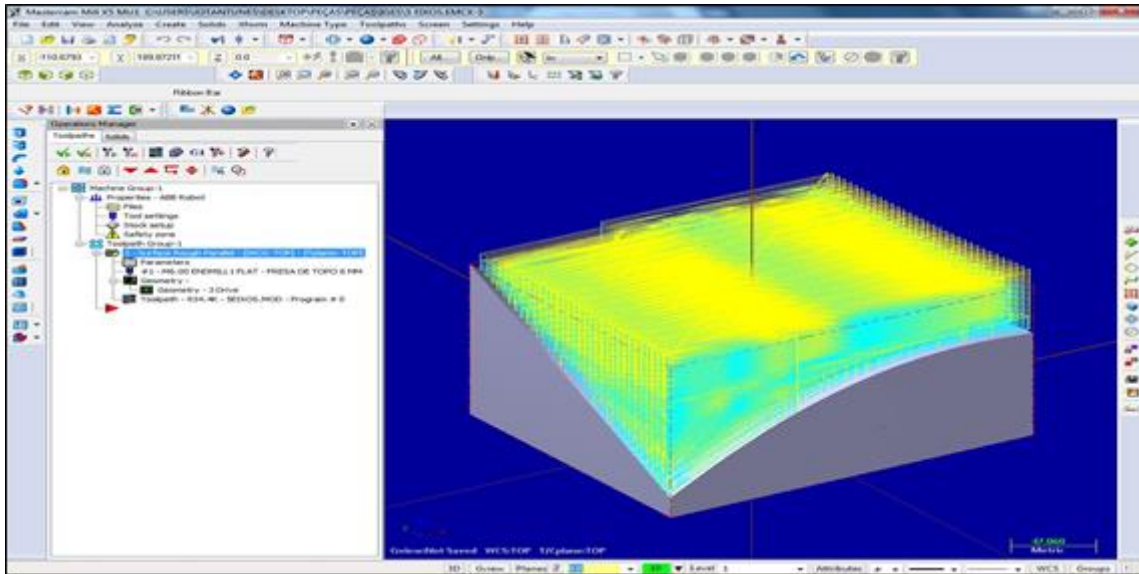


Figura 4.36 - Trajetória gerada (Mastercam)

A operação de acabamento é executada recorrendo à maquinagem com 5 eixos. Para seleccionar este tipo de operação, no separador “*Toolpaths*” é escolhida uma operação do tipo “*multiaxis*”, que disponibiliza várias opções de maquinagem. Para este exemplo seleccionou-se a “*MutipleSurface*”. Aquando da seleção da operação de maquinagem é possível aceder de imediato às configurações da mesma, como representado na Figura 4.37.

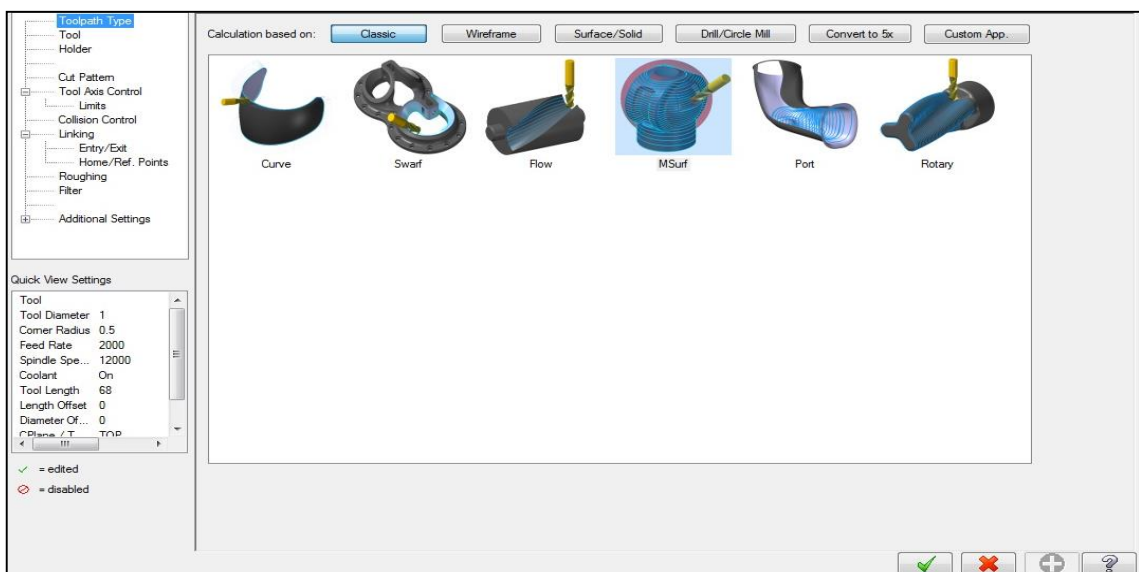


Figura 4.37 - Seleção e configuração da operação *Multiaxis* (Mastercam)

Etapa 4 – Verificação através de simulação

Numa primeira simulação foi possível verificar que para o posicionamento da peça, o robô não iria conseguir maquinar devido a um caso de singularidade. Utilizando a ferramenta “*optimization*” foi possível criar a operação com sucesso. Após a realização da Etapa 5 foi possível obter o código na linguagem pretendida (Rapid).

4.6 Programação com RobotStudio/Machining PowerPac

A descrição do percurso necessário a percorrer para ser possível obter uma operação de maquinagem com recurso a este *software* é idêntico aos anteriores, sendo expostas as diferenças mais relevantes. À semelhança da apresentação do *software* Mastercam/RobotMaster apenas é apresentado um exemplo, sendo escolhida mais uma vez a peça 3. De realçar que este estudo só aborda a operação de acabamento da superfície da peça 3, uma vez que o *software* tem limitações para ser usado de forma expedita em operações de desbaste que envolvam múltiplas passagens.

Etapa 1 – Escolha da máquina e preparação da estação de trabalho

Este *software* apresenta uma extensa biblioteca de robôs da ABB que é possível utilizar, assim como vários componentes fornecidos pela ABB, nomeadamente: mesa de trabalho, pedestal e sensor de força. Após o correto posicionamento do robô e dos vários componentes é possível criar a célula robótica representada na Figura 4.38. Note-se que nesta figura já está representado o *Spindle* criado em ambiente SolidWorks e importado para o *software* RobotStudio.

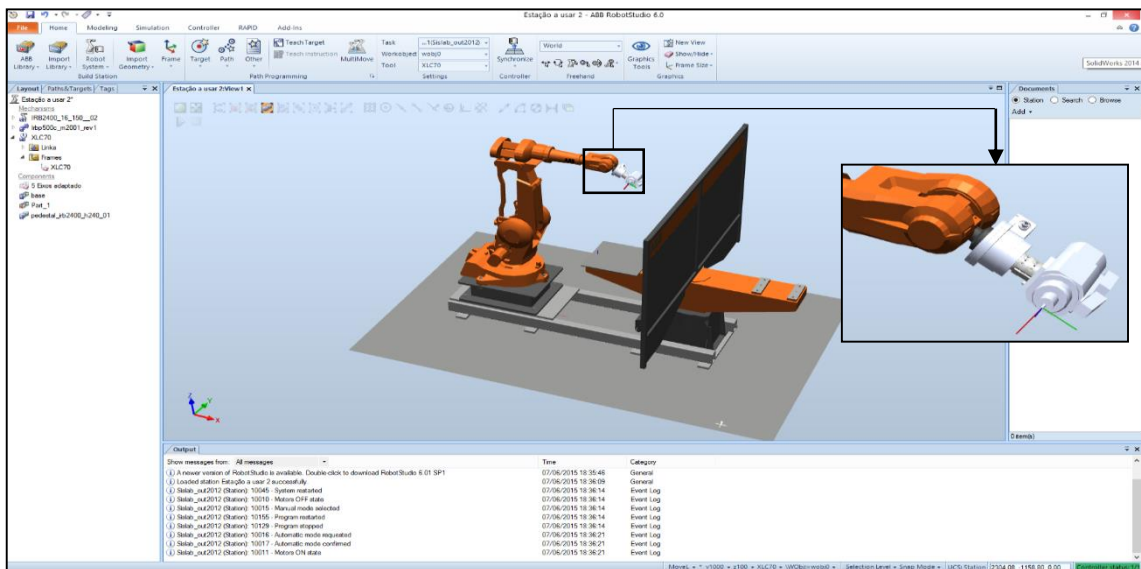


Figura 4.38 - Célula robótica virtual (RobotStudio)

Etapa 2 - Modelação da peça e importação para o *software* CAM

Este *software* também apresenta várias ferramentas que possibilitam posicionar/orientar a peça conforme as necessidades do utilizador. A utilizada neste caso foi a “*set position*”, onde é possível posicionar a peça em relação a vários referenciais, sendo selecionado o referencial de base do robô, como representado na Figura 4.39.

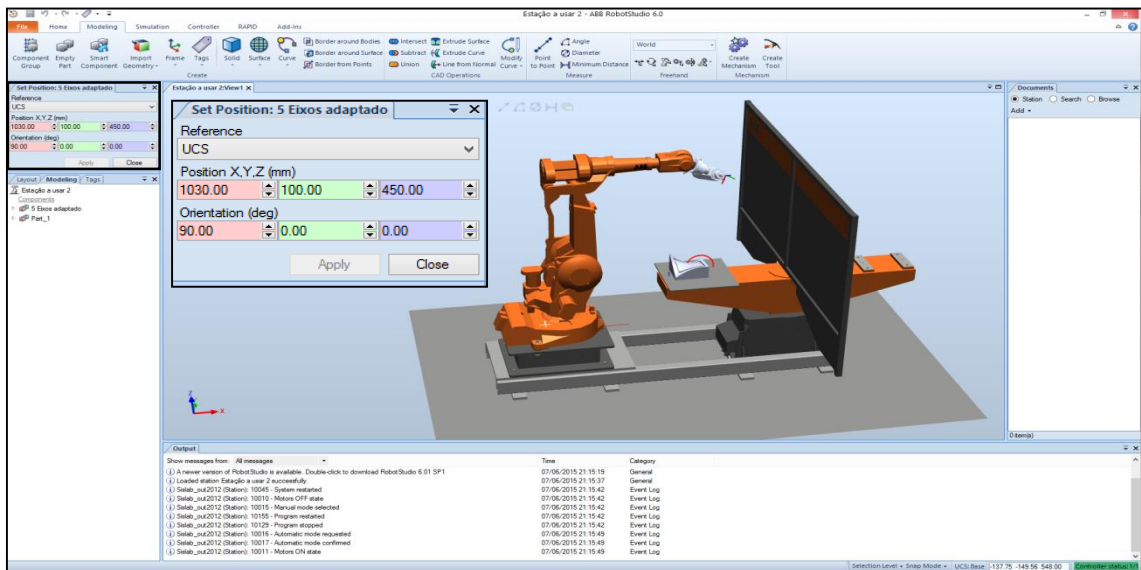


Figura 4.39 - Posicionamento da peça (RobotStudio)

Etapa 3 - Seleção e parametrização das operações de maquinagem

A seleção e parametrização das operações de maquinagem neste *software* são diferentes dos restantes *softwares*. O Machining PowerPac apresenta uma funcionalidade bastante interessante denominada de “*Solution Wizard*”. Ao aceder a esta funcionalidade é possível seleccionar e parametrizar uma operação de maquinagem de uma forma bastante expedita. Esta é separada por sete “passos” que serão apresentados de seguida.

O primeiro passo é seleccionar qual o controlador e robô a utilizar para a operação de maquinagem. O segundo é seleccionar qual a face a maquinar, como representado na Figura 4.40.

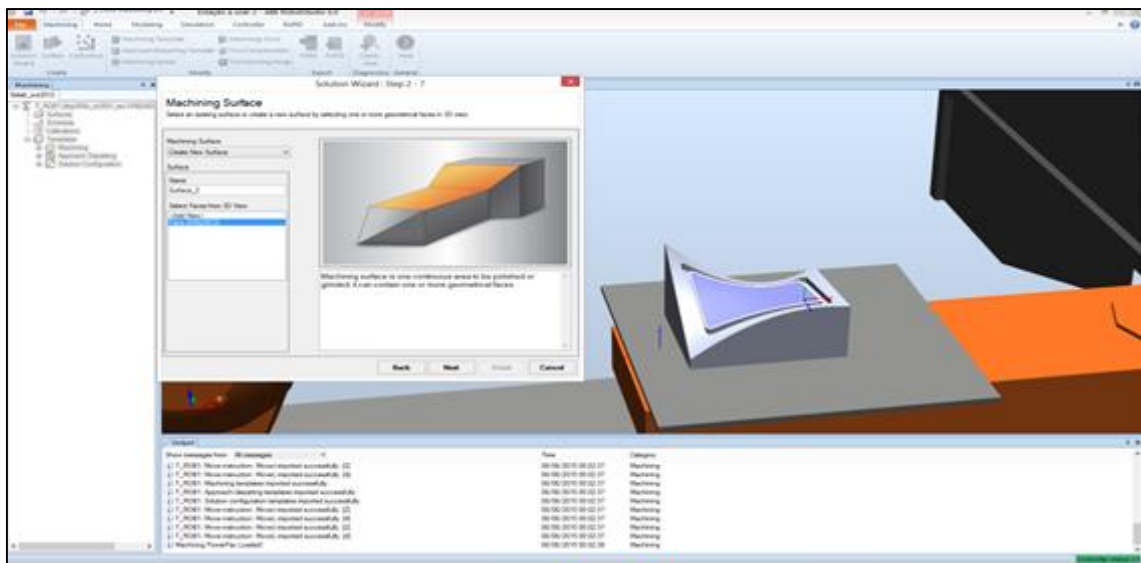


Figura 4.40 - Seleção da face a maquinar (RobotStudio e Machining PowerPac)

No terceiro passo define-se os parâmetros de maquinagem, sendo estes: a tolerância que neste caso é a máxima distancia que a trajetória da ferramenta pode ter em relação à superfície a

maquinar, o *Stepover* que neste *software* é denominado de *Machining overlap* e ainda é possível limitar a velocidade máxima e mínima do TCP. O quarto passo é a parametrização da ferramenta (onde é possível definir o raio e o comprimento da lâmina), a seleção do tipo de maquinagem (cilíndrica ou de topo) e as dimensões da mesma. O terceiro e quarto passos estão representados na Figura 4.41.

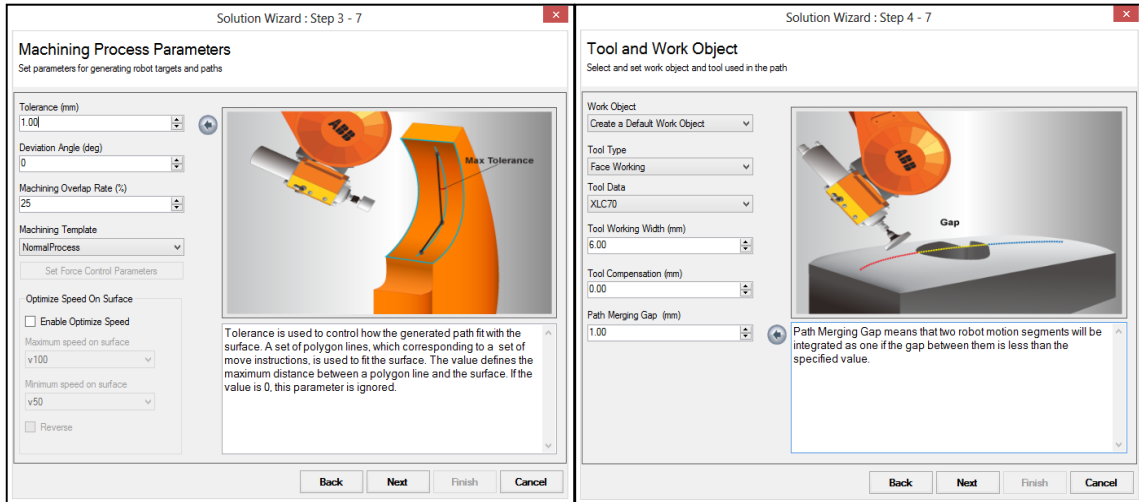


Figura 4.41 - Parametrização da operação de maquinagem e da ferramenta

Seguidamente, no quinto passo, é definida qual a estratégia a utilizar na geração da trajetória da ferramenta. É possível escolher, em função do tipo de superfície, uma de seis possíveis. No presente exemplo selecionou-se “*Cutting Planes Along Straight Line*”, demonstrado na Figura 4.42. O sexto passo diz respeito à definição da orientação da ferramenta ao longo da trajetória e ao tipo de movimentação entre os vários percursos adjacentes.

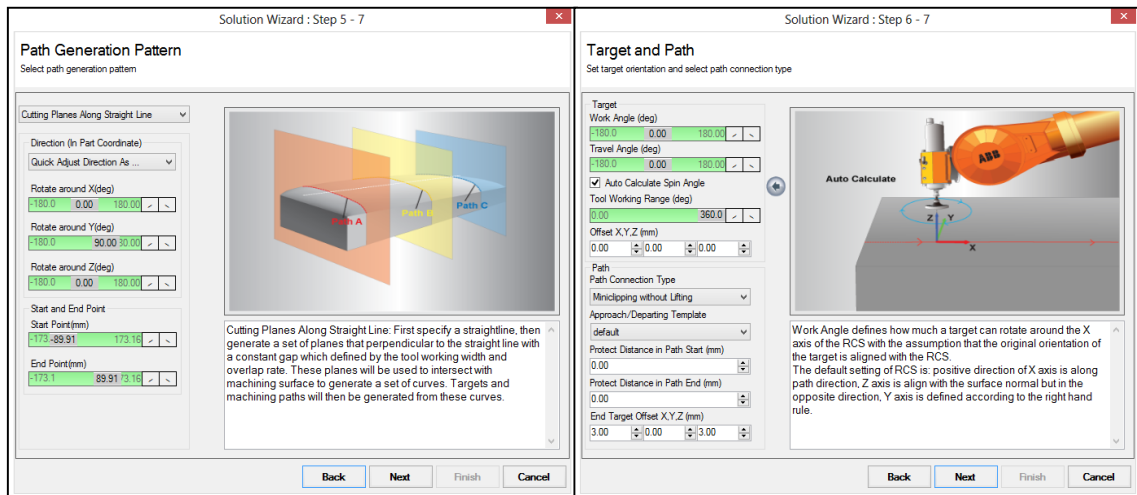


Figura 4.42 - Parametrização da trajetória (RobotStudio e Machining PowerPac)

No último menu, ou no passo número sete, é possível visualizar a trajetória gerada, Figura 4.43. De salientar que nesta pré-visualização não são representados os movimentos de aproximação nem de retorno.

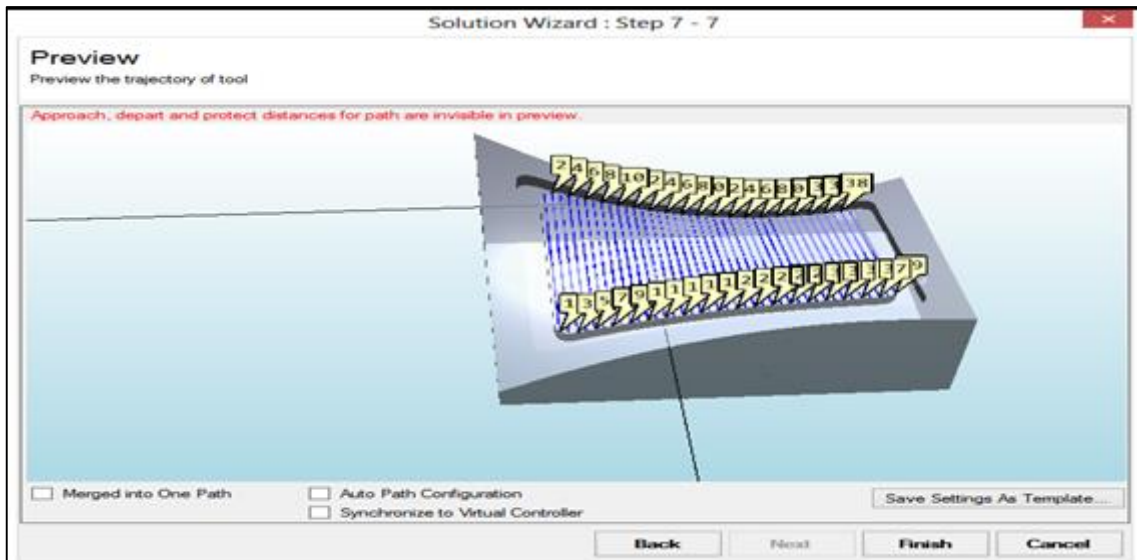


Figura 4.43 - Trajetória gerada (RobotStudio e PowerPac)

Etapa 4 – Verificação através de simulação e geração do código Rapid

Neste *software* para ser possível detetar possíveis colisões na trajetória gerada é necessário, antes da simulação, utilizar a funcionalidade “Collision Set” onde se definem quais os componentes que se pretende “controlar”. Os componentes seleccionados foram: a peça, o *Spindle*, base de trabalho e a mesa de trabalho. Repare-se que caso não seja definido este conjunto de componentes, o *software* não é capaz de identificar colisões como a assinalada na Figura 4.44.

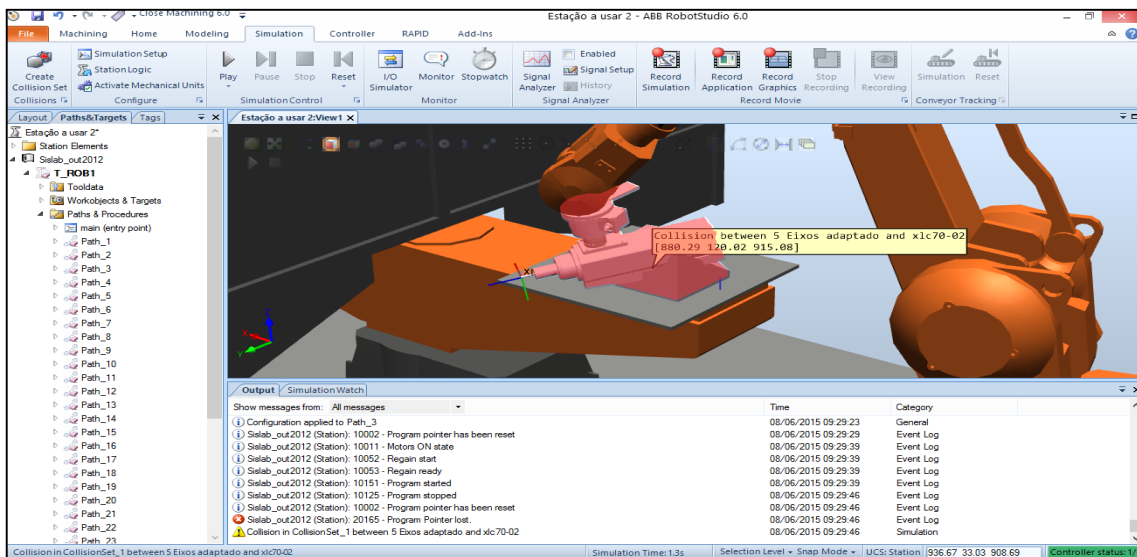


Figura 4.44 - Colisão entre o *Spindle* e a Peça 3

A origem desta colisão tem que ver com o tipo de movimento que o robô executa entre passagens consecutivas de maquinagem. É possível resolver esta colisão de duas maneiras distintas: modificar a trajetória gerada pelo *add-in* redefinindo os parâmetros da etapa 3 ou modificar o código RAPID gerado. No presente caso optou-se pela segunda solução e após algumas modificações no código relativamente à configuração do robô é possível criar a trajetória pretendida com sucesso.

4.7 Validação e análise dos resultados obtidos

Antes de utilizar o código RAPID gerado pelo *software* Mastercam/Robotmaster e Sprutcam no controlador real do robô é realizada uma validação, através de simulação do programa em ambiente virtual, disponibilizado no RobotStudio. Desta maneira, garante-se que tanto o percurso de maquinagem como os movimentos de posicionamento realizados pelo robô são exequíveis e que não contêm nenhuma colisão. À semelhança da comparação apresentada no subcapítulo 4.2, estes programas também foram analisados mais detalhadamente utilizando a funcionalidade *Signal Analyzer*, porém desta vez é comparado o tempo total dos programas gerados (movimentos de aproximação e de retorno incluídos).

As operações de desbaste e de acabamento utilizadas são semelhantes nos dois *softwares*. O número de operações apresentadas nas tabelas dizem respeito a quantas operações foram necessárias definir, nos diferentes *softwares*, para realizar a peça pretendida.

Peça 1

A estratégia de desbaste adotada foi a do tipo Paralela. O Sprutcam cria uma trajetória com um movimento mais constante na maquinagem (tem menos movimentos de aproximação e de retorno da ferramenta em relação à peça), algo que o Mastercam/Robotmaster não é capaz de gerar. Desta maneira é possível compreender as diferenças nos resultados obtidos nesta peça, uma vez que o Mastercam/Robotmaster cria uma trajetória maior na operação de desbaste, o tempo total será também maior. Em relação à operação de acabamento, a selecionada foi a do tipo “contorno” e ambos os programas têm comportamentos semelhantes. Os resultados obtidos para a Peça 1 são os apresentados na Tabela 14.

Tabela 14 - Resultados da simulação do programa de maquinagem da Peça 1

	Sprutcam	Mastercam/Robotmaster
Número de operações de desbaste	1	1
Número de operações de acabamento	3	3
Funcionalidade Signal Analyzer		
Tempo total [hh:mm:ss]	0:06:15	0:08:19
Velocidade média [mm/s]	49.7	45.9
Potência máxima [W]	124	133
Energia Consumida [KJ]	20.69	35.45

Peça 2

No presente caso, e à semelhança do anterior, o tempo despendido na operação de desbaste foi superior no Mastercam/Robotmaster em relação ao do Sprutcam. A operação de acabamento selecionada para a maquinagem das superfícies laterais foi a do tipo contorno. Para a superfície curvilínea foi a “*Finishing Drive*” no Sprutcam e “*Surface Finish Flowline*” no Mastercam/Robotmaster. Os resultados obtidos para a Peça 2 são os apresentados na Tabela 15.

Tabela 15 - Resultados da simulação do programa de maquinagem da Peça 2

	Sprutcam	Mastercam/Robotmaster
Número de operações de desbaste	1	1
Número de operações de acabamento	2	2
Funcionalidade Signal Analyzer		
Tempo total [hh:mm:ss]	0:05:08	0:06:25
Velocidade média [mm/s]	49.5	48.4
Potência máxima [W]	149	212
Energia Consumida [KJ]	21.97	58.61

Peça 3

Neste caso de estudo, onde foram seleccionadas operações de maquinagem a 5 eixos para as operações de acabamento, são realizados movimentos mais complexos por parte do robô. Após uma primeira simulação em ambiente RobotStudio dos programas gerados foi possível verificar que apenas o código gerado pelo Mastercam/Robotmaster era exequível e que não continha casos de singularidades como no código gerado pelo Sprutcam. Porém após algumas tentativas de reposicionamento do “*Workobject*” foi possível executar a trajetória de maquinagem da Peça 3. Os resultados obtidos para a Peça 3 são os apresentados na Tabela 16.

Tabela 16 - Resultados da simulação do programa de maquinagem da Peça 3

	Sprutcam	Mastercam/Robotmaster
Número de operações de desbaste	1	1
Número de operações de acabamento	3	4
Funcionalidade Signal Analyzer		
Tempo total [hh:mm:ss]	1:08:52	1:00:37
Velocidade média [mm/s]	49.8	47.7
Potência máxima [W]	482	299
Energia Consumida [KJ]	339.59	544.94

Ao analisar os resultados obtidos é possível constatar que embora se utilize parâmetros de corte e estratégias de maquinagem semelhantes, os programas gerados por pós-processadores de *softwares* distintos traduzem-se em códigos diferentes, originando tempos de maquinagem dissemelhantes. As diferenças verificadas devem-se, fundamentalmente, às trajetórias geradas nas operações de desbaste, assim como as trajetórias geradas para a aproximação da ferramenta à peça. Em relação à velocidade de avanço definida, 50 mm/s, é possível verificar que atingiu em todos os casos de estudo um valor bastante próximo do pretendido, realçando apenas o facto de o Sprutcam gerar programas com uma velocidade média mais próxima do especificado.

5 Maquinagem com célula robótica real

Mais importante que o tempo necessário para efetuar cada uma das peças, é avaliar se os programas gerados, por cada um dos *softwares*, cumprem as geometrias e acabamentos pretendidos. Neste capítulo é apresentada a célula robótica utilizada e alguns cuidados que se devem ter em consideração para a execução da maquinagem real.

5.1 Descrição da célula

A célula robótica (Figura 5.1) existente no laboratório de robótica do Departamento de Engenharia Mecânica da Faculdade de Engenharia da Universidade do Porto é constituída pelos seguintes elementos:

1. Robô Industrial ABB *IRB 2400-16*.
2. Controlador ABB *IRC5*.
3. Mesa posicionadora *IRPB C-500*.
4. Célula de carga ATI *Delta IP60*.
Sistema pneumático de mudança de ferramenta *SCHUNK SWS-011*.
Spindle PDC XLC70.
5. Controlador do *Spindle*.



Figura 5.1 - Célula robótica da FEUP

Robô industrial ABB IRB 2400-16

Este modelo da ABB de estrutura revoluta é constituído por 6 eixos rotativos em série, representado na Figura 5.2. Os dados técnicos deste modelo encontram-se na *datasheet* apresentado no anexo A, no entanto é de se destacar as seguintes características:

Tabela 17 - Características do robô IRB 2400-16

Repetibilidade	0.06 mm
Capacidade de carga	20 Kg
Alcance	1.55 m

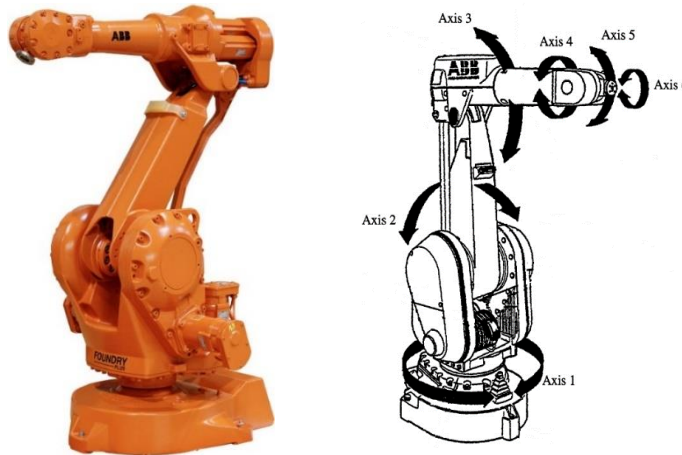


Figura 5.2 - ABB IRB 2400-16

Controlador IRC5

O controlador IRC5 é de grande referência na indústria robótica, e representa a última geração de controladores da marca ABB. Este controlador oferece uma grande flexibilidade e segurança.



Figura 5.3 - Controlador IRC5

O IRC5 otimiza automaticamente o desempenho do robô reduzindo os tempos de ciclo, denominado por *QuickMove*, e proporciona um trajeto com alta precisão, designado por *TrueMove*. Para além dos 2 modos de controlo de movimento referenciados anteriormente, existe ainda a opção de movimento *MultiMove*, que possibilita a utilização até 4 robôs em simultâneo. Informações mais detalhadas sobre este controlador estão presentes na *datasheet*, apresentado no anexo A.

Mesa posicionadora ABB IRBP C-500

Esta mesa posicionadora rotativa apresenta uma repetibilidade no seu movimento de 0,1 mm, e tem uma capacidade de carga até 500Kg. No entanto, não pode existir uma diferença entre cargas aplicadas nos dois lados opostos da mesa que ultrapasse os 350Kg. A sua utilização torna-se bastante fácil uma vez que é do mesmo fabricante do controlador utilizado. Na Figura 5.4 está representada a mesa posicionadora.



Figura 5.4 - Mesa ABB IRBP C-500

O painel preto localizado no meio da mesa separa duas áreas de trabalho. Este ecrã para além de proteger o operador de poeiras ou radiações provenientes de processos de soldadura, permite a realização de operações separadas. Porém na célula real este painel não se encontra montado. Informações mais detalhadas sobre a mesa posicionadora estão presentes na *datasheet* apresentado no anexo A.

Célula de carga ATI Delta IP60

Este sensor, representado na Figura 5.5, é fabricado pela ATI *Industrial Automation*. Esta empresa dedica-se à produção de componentes para robôs industriais.



Figura 5.5 - ATI Delta IP60

Este componente é construído com peças de alta resistência em aço inoxidável, apresentando uma distorção de ruído muito próxima de zero. A célula de carga é o modelo SI-330-30, e consegue medir forças até 330 N para F_x e F_y e forças até 990 N para F_z , Binários até 30 Nm para T_x , T_y e T_z . Tem uma resolução de 1/16 N para F_x e F_y , e 1/8N para F_z , e uma resolução de 5/1333 Nm para T_x , T_y e T_z . A sua frequência de ressonância ronda os 1100 Hz [32].

Sistema pneumático de mudança rápida de ferramenta *SCHUNK SWS - 011*

É possível efetuar uma mudança rápida de ferramenta através deste sistema de acionamento pneumático. Na Figura 5.6 está representado o sistema de mudança de ferramenta *Schunk SWS – 011* (*datasheet* apresentado no Anexo A).



Figura 5.6 - *Schunk SWS-011*

Spindle XLC70

O *spindle* ou motor-árvore disponível é o modelo *XLC70* da marca Precision Drive System, que é destinado a aplicações de maquinagem de alta velocidade. Apresenta uma potência de 2.2 KW, pode atingir velocidades de 40000 rpm e pesa 11 Kg.



Figura 5.7 - *Spindle XLC70*

5.2 Setup das operações de maquinagem

O material selecionado para maquinar foi uma espuma de poliuretano expandido de 55kg/m³ (Figura 5.8). Este material foi o escolhido por ser facilmente maquinado e ser uma espuma macia, o que contribui para assegurar condições de segurança nas operações de maquinagem.



Figura 5.8 - Poliuretano expandido

As ferramentas utilizadas são da marca Li-Hsing Precision Tool, referência LH2EMB-060 (fresa plana) e LH2BZB-060 (fresa boleada) com as dimensões apresentadas na Tabela 18.

Tabela 18 - Dimensões das ferramentas

	Fresa plana	Fresa boleada
Nº de lâminas de corte	2	2
Ø Fresa	6 mm	6 mm
Comprimento total	100 mm	100 mm
Profundidade da lâmina	30 mm	12 mm

Para realizar a implementação dos programas gerados foi necessário executar uma última configuração dos mesmos. Isto porque, embora em ambiente virtual seja possível ter uma representação da célula robótica bem próxima da realidade, nas simulações anteriormente realizadas não se teve em conta o sistema de fixação. O procedimento envolveu as seguintes etapas.

1- Calibração do TCP.

Realizou-se mais uma vez a calibração do TCP, utilizando o método dos 4 pontos para se ter a certeza que os dados das ferramentas introduzidos nos programas estão corretos.

2- Posicionamento do bloco de trabalho.

Uma vez que o bloco de trabalho não era igual ao criado em ambiente virtual foi necessário reposicionar o mesmo e realizar novos testes em ambiente virtual com os dados obtidos de medições experimentais. Na Figura 5.9 é possível observar um dos blocos de material utilizado para a maquinagem.



Figura 5.9 - Fixação do bloco de trabalho

3- Redefinição do eixo 5.

O suporte da célula de carga tem uma geometria que não permite que o eixo 5 utilize toda a sua amplitude de rotação ($\pm 120^\circ$). Para valores próximos de 95° , o suporte entra em contato com o braço do robô. Assim é necessário configurar este novo limite de rotação nos *softwares* e realizar uma última simulação em ambiente RobotStudio para validar os programas já definidos.

Após estes cuidados adicionais foi possível proceder à maquinagem real. Os programas gerados no capítulo 4 foram então reconfigurados e introduzidos no controlador do robô. Os resultados da maquinagem são os apresentados de seguida.

Peça 1

Os resultados de maquinagem da Peça 1 revelaram uma superfície bastante macia nas superfícies planas, como previsto em ambos os *softwares*. Em relação às superfícies curvas as peças criadas também tiveram bons resultados, apresentando superfícies sem irregularidades. Na Figura 5.10 é possível visualizar a peça obtida com cada um dos *softwares*.

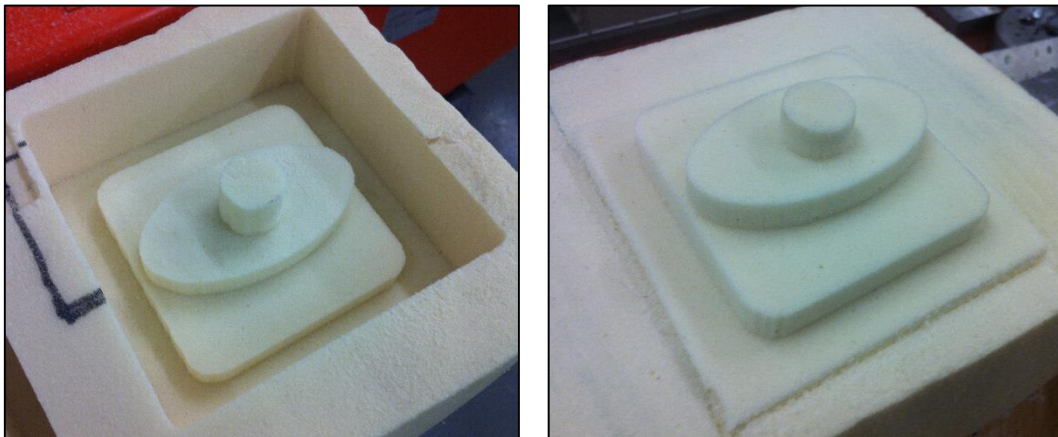


Figura 5.10 - Peça 1 (Sprucam esquerda e Mastercam/Robotmaster direita)

Peça 2

A maquinagem da peça 2 envolveu a utilização de uma fresa boleada para a operação de acabamento. A superfície curvilínea revelou uma superfície bastante regular, como é possível verificar na Figura 5.11. Ainda que os resultados da maquinagem de ambas as peças estejam perfeitamente de acordo com as simulações realizadas, nos dois *softwares* utilizados, é possível obter uma superfície ainda mais regular diminuindo o *stepover*. Contudo esta diminuição na distância dos percursos de maquinagem irão, naturalmente, aumentar o tempo de maquinagem.

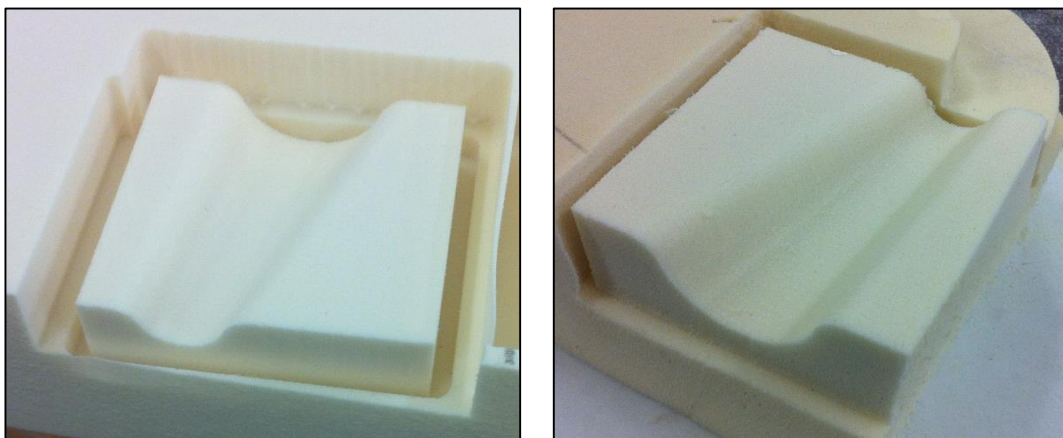


Figura 5.11 - Peça 2 (Sprucam esquerda e Mastercam/Robotmaster direita)

Peça 3

Na maquinagem da primeira peça 3, onde se utilizou o código gerado pelo pós-processador do Mastercam/Robotmaster, o acabamento da mesma apresentou uma superfície bastante regular. No entanto, existiu um movimento, entre operações de maquinagem diferentes, que o *software* CAM não representou na simulação. É possível observar este movimento indesejado na Figura 5.12.

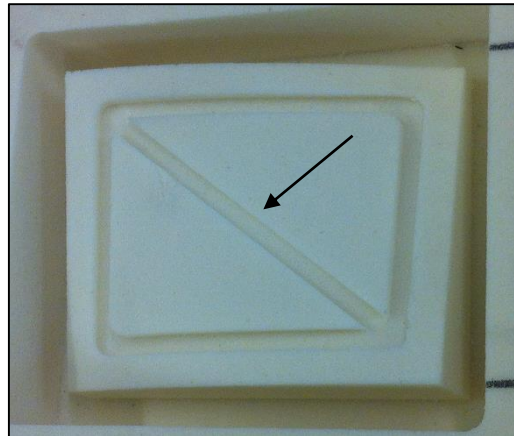


Figura 5.12 - 1ª Maquinagem da Peça 3 (Mastercam/Robotmaster)

Utilizando o RobotStudio identificou-se qual o movimento que criava este “rasgo” e editando o código manualmente foi possível obter a peça pretendida. Após estas modificações, e utilizando também o código gerado pelo pós-processador do Sprutcam, foi possível realizar as peças pretendidas. Na Figura 5.13, estão representadas os resultados de maquinagem.

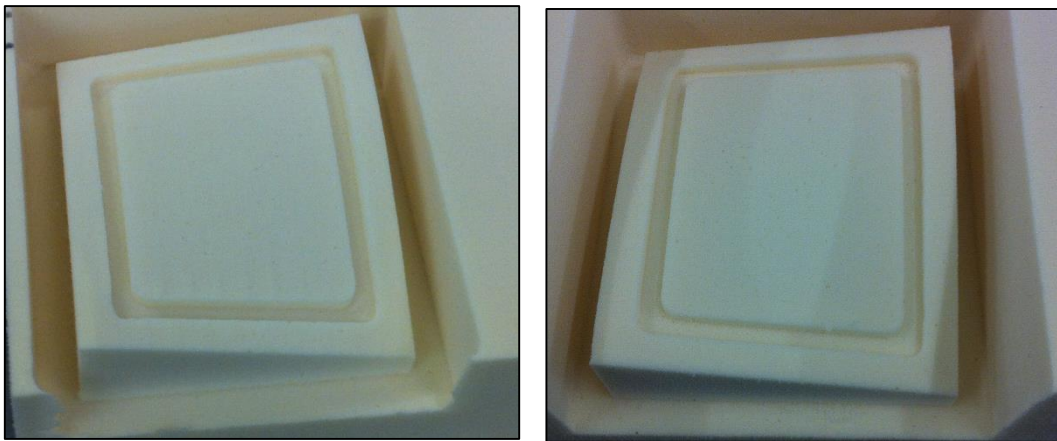


Figura 5.13 - Peça 3 (Sprutcam esquerda e Mastercam/Robotmaster direita)

À semelhança do Mastercam/Robotmaster, o código gerado pelo pós-processador do Sprutcam também se traduziu numa peça com uma superfície bastante regular, apresentado um resultado final idêntico em ambos os casos, conforme o esperado.

6 Conclusões e trabalhos futuros

O objetivo desta dissertação passa, principalmente, por demonstrar quais as facilidades/dificuldades que os *softwares* CAM proporcionam ao utilizador para gerar percursos de maquinagem quando aplicados a robôs industriais. Após a elaboração deste trabalho é possível obter algumas conclusões relacionadas com o uso de *software* CAM, nomeadamente Sprutcam, Mastercam/Robotmaster e RobotStudio/Machining PowerPac. Os dois primeiros constituem soluções genéricas que permitem a programação de robôs industriais, para operações de maquinagem. O *software* RobotStudio/Machining PowerPac não apresenta as mesmas potencialidades CAM, não sendo possível criar operações, por exemplo, de desbaste.

O problema mais relevante, quando se utiliza *software* CAM para a criação de operações de maquinagem com robôs industriais, diz respeito aos pós-processadores, uma vez que não é qualquer *software* CAM que permite a criação dos códigos na linguagem pretendida.

Ao analisar os resultados obtidos ao longo deste trabalho torna-se viável constatar algumas diferenças pertinentes, em relação aos *softwares* CAM utilizados.

Interface

Do ponto de vista do utilizador comum a interface é de extrema importância. Menus simples e expeditos são características quase que obrigatórias na atualidade para qualquer tipo de *software*. Comparando as aplicações informáticas utilizadas é possível afirmar, que provavelmente devido à diferença das datas de lançamento, a interface do Sprutcam e do RobotStudio apresentam uma interface bastante mais intuitiva ou “*user-friendly*” quando comparadas à do Mastercam.

Funcionalidade CAD

Para a criação de modelos de geometrias simples, o Mastercam e o RobotStudio conseguem executar facilmente, algo que não é exequível no Sprutcam. Porém para a modelação de peças com geometrias mais complexas, não é aconselhado a utilização dos mesmos, uma vez que o processo torna-se moroso e de difícil execução. Por esta razão, será interessante que contemplem uma grande variedade de formatos que consigam importar, possibilitando, desta maneira, a transferência de ficheiros gerados em *softwares* mais eficazes nesta área, característica que é comum a todos os *softwares* abordados no presente trabalho.

Funcionalidade CAM

Para a criação de operações de maquinagem utilizando *softwares* CAM, é necessário que estes contenham na sua biblioteca o modelo da máquina, ou neste caso do robô pretendido. Assim, no Sprutcam, e devido à versão utilizada, houve a necessidade de realizar a modelação do robô, uma vez que este não se encontrava disponível, enquanto no Mastercam/Robotmaster o *software* já continha na sua biblioteca o modelo pretendido, o *IRB 2400* da ABB.

Em relação às ferramentas que são possíveis de utilizar para as operações de maquinagem, tanto o Sprutcam como o Mastercam, apresentam uma biblioteca bastante completa. Ainda assim, permitem a edição ou criação de novas ferramentas que podem também ser utilizadas posteriormente desde que sejam guardadas na biblioteca.

Na maquinagem 2D ambos *softwares* permitem definir várias estratégias de maquinagem de uma forma expedita, sendo de destacar a facilidade com que se cria uma operação deste tipo.

Relativamente à maquinagem 3D, já existem algumas diferenças na criação das operações. A mais relevante é a maneira como o *software* CAM identifica as propriedades geométricas do modelo da peça a maquinar. Isto é algo que está diretamente relacionado com o formato do ficheiro importado. É de extrema importância que o utilizador tenha em atenção e/ou algum conhecimento sobre as características de cada tipo de ficheiro que importa para o *software* CAM para ser possível parametrizar a operação de maquinagem pretendida mais facilmente.

A maquinagem 5D apresenta um grau de dificuldade elevado na parametrização, o que obriga o utilizador a um estudo bastante mais intensivo e prolongado sobre as funcionalidades que cada *software* proporciona. Porém o *software* em que foi mais fácil realizar este tipo de operações foi o Mastercam. A informação disponibilizada nos menus referentes a este tipo de maquinagem é bastante completa, esclarecendo o utilizador de quais devem ser os parâmetros a definir e principalmente como os definir. Pelo que a apreciação final é que o Mastercam é uma solução mais interessante para a maquinagem 5D.

Códigos gerados

Um aspeto importante a referir é o facto do *add-on* Robotmaster, do Mastercam, ser o único que permite ao utilizador parametrizar qual o tipo de zona que pretende utilizar na definição das instruções de movimentação (do código RAPID). Outra funcionalidade muito interessante deste *add-on* é a “*optimization*”, que demonstrou ao longo deste trabalho ser uma funcionalidade bastante poderosa, permitindo de uma forma expedita, a reconfiguração de trajetórias que continham colisões e/ou singularidades. Contudo é necessário ter em atenção os códigos gerados pelo pós-processador do Robotmaster, isto porque, como foi demonstrado na presente dissertação, este pós-processador não realiza a conversão de mm/min para mm/s. Assim é necessário o utilizador, após a obtenção do código gerado pelo Mastercam/Robotmaster, editar o mesmo.

Outro aspeto a ter em conta é a extensão dos programas gerados. Utilizando a funcionalidade “*Signal Analyzer*” (do RobotStudio) foi possível ao longo do trabalho verificar que os códigos gerados pelo Robotmaster, normalmente, demoravam mais tempo a maquinar a peça e que consumia mais energia quando comparado ao Sprutcam. As diferenças nos consumos energéticos devem-se a diferentes trajetórias, isto é, para parâmetros de corte e estratégias de maquinagem idênticas o Mastercam gerou, na maioria dos casos, uma trajetória maior.

Os resultados dos ensaios experimentais comprovam a viabilidade da aplicação de *softwares* CAM para a criação de operações de maquinagem com robôs industriais. Porém ainda existe

uma grande dificuldade em confiar nestes sistemas. Recomendando-se sempre uma validação usando os *softwares* proprietários dos fabricantes dos robôs. A utilização do RobotStudio mostrou ser imprescindível para a realização deste trabalho, uma vez que para alguns códigos gerados (tanto do Sprutcam como do Mastercam e até do próprio *add-on* do RobotStudio, o Machining PowerPac) existiu a necessidade de reconfigurar alguns movimentos do robô para que fosse possível executar as trajetórias geradas.

Trabalhos Futuros

Como trabalhos futuros sugere-se:

- Programação do sistema automático de troca de ferramenta.
- A integração de um sistema de calibração automático da ferramenta.
- A instalação de um mecanismo de posicionamento da peça a maquinar.
- Investigar a possibilidade de adaptar/melhorar os pós-processadores para RI.

Referências

- [1] J. Pandremenos, C. Doukas, P. Stavropoulos e G. Chryssolouris, “Machining with Robots: A Critical Review,” em *7th International Conference on Digital Enterprise Technology*, Athens, 2011.
- [2] H. Abdul-Mushin e V. Patel, *Robotics in General Surgery*, Springer, 2013.
- [3] R. Paul, *Robots, Models, and Automation*, Purdue University, 1979.
- [4] K. R. Group, “Kuka,” Kuka, [Online]. Available: <http://www.kuka-robotics.com/usa/en/company/group/milestones/1973.htm>. [Acedido em 5 Março 2015].
- [5] W. F. Lages, *Robot Retrofitting: A Perspective to Small and Medium Size Enterprises*, Federal University of Rio Grande do Sul, 2004.
- [6] “International Federation of Robotics,” [Online]. Available: <http://www.ifr.org/service-robots/statistics/>. [Acedido em 27 Junho 2015].
- [7] Fanuc. [Online]. Available: http://robot.fanucamerica.com/robotics-articles/robotic_accuracy_improves_aerospace_manufacturing.aspx. [Acedido em 27 Julho 2015].
- [8] Midwest Engineered Systems Inc., [Online]. Available: <http://www.mwes.com/blog/tag/robotic-deburring/>. [Acedido em 20 Julho 2015].
- [9] M. R. S. Barbosa, *Acetatos de Apoio às Aulas Teóricas*, Porto: FEUP, 2014.
- [10] COMET, “Plug-and-produce COmponentes and METHods for adaptive control of industrial robots enabling cost effective, high precision manufacturing in factories of the future,” Setembro 2010.
- [11] ABB, “ABB,” [Online]. Available: [http://www09.abb.com/global/scot/scot230.nsf/veritydisplay/b93e2726298314b3c1257b130056d10a/\\$file/IRB%206600%20R%20%20US%2002_05.pdf](http://www09.abb.com/global/scot/scot230.nsf/veritydisplay/b93e2726298314b3c1257b130056d10a/$file/IRB%206600%20R%20%20US%2002_05.pdf). [Acedido em 28 04 2015].
- [12] R. A. Inc., “FANUC,” [Online]. Available: http://www.fanucamerica.com/cmsmedia/datasheets/F-200iB%20Series_9.pdf. [Acedido em 18 06 2015].
- [13] KUKA. [Online]. Available: http://www.kuka-robotics.com/usa/en/products/industrial_robots/heavy/kr500_fortec/kr500_r2830_mt/.

- [Acedido em 28 06 2015].
- [14] Stäubli. [Online]. Available: <http://www.staubli.com/en/robotics/6-axis-scara-industrial-robot/specialized-robot/high-speed-machining-robots/>. [Acedido em 06 26 2015].
- [15] H. Gerling, A volta da Máquina Ferramenta, 1967.
- [16] F. D. J. L. H. e. H. H. R. ERIK OBERG, Machinery's Handbook 29th Edition, New York: Industrial Press, 2012.
- [17] H. WORKS, AUTODESK, [Online]. Available: http://www.hsmworks.com/docs/cncbook/en/#Ch03_CuttingData. [Acedido em 21 06 2015].
- [18] Workspace, “WAT Solutions,” WAT Solutions, [Online]. Available: http://www.workspaceit.com/Workspace5/WS5_Index.htm. [Acedido em 17 Junho 2015].
- [19] Octopuz, “In-House Solutions,” In-House Solutions, [Online]. Available: <http://octopuz.com/>. [Acedido em 17 Junho 2015].
- [20] Delmia, “Dassault Systemes,” Dassault Systemes, [Online]. Available: <http://www.3ds.com/products-services/delmia/capabilities/robotics/robot-programming/>. [Acedido em 17 Junho 2015].
- [21] F. A. Corporation, “FANUC,” [Online]. Available: <http://robot.fanucamerica.com/products/vision-software/ROBOGUIDE-simulation-software.aspx>. [Acedido em 17 Junho 2015].
- [22] Motoman, “Yaskawa America Inc.,” [Online]. Available: <http://www.motoman.com/datasheets/MotoSim%20EG%20VRC.pdf>. [Acedido em 17 Junho 2015].
- [23] D. Ltd, “Delcam,” [Online]. Available: <http://www.delcam.com/partners/robotic/index.asp>. [Acedido em 17 Junho 2015].
- [24] Roboris, “Eureka Virtual Machining,” [Online]. Available: <http://www.roboris.it/en/product/eureka-robot>. [Acedido em 17 Junho 2015].
- [25] “IRBCAM,” [Online]. Available: <http://www.irbcam.com/>. [Acedido em 17 Junho 2015].
- [26] S. S. Inc., “Simtech,” [Online]. Available: http://www.easysimulation.com/web/html/coniferrob_en.htm. [Acedido em 17 Junho 2015].
- [27] KUKA.CAMRob, “KUKA,” [Online]. Available: http://www.kuka.be/main/products/applications/table/camrob22/en_cr_table.htm. [Acedido em 17 Junho 2015].
- [28] Motoman, “Yaskawa America Inc.,” [Online]. Available: <http://www.motoman.com/datasheets/G-CodeEG.pdf>. [Acedido em 17 Junho 2015].
- [29] A. Group, “ABB,” [Online]. Available:

<https://library.e.abb.com/public/1f3591528499cc43c1257e34002bfa9a/Machining%20PowerPac%20-%20EXTERNAL.pdf>. [Acedido em 17 Junho 2015].

- [30] P. Abreu, Manual de utilização - Parte 2 - Programação I, Edição 1.3, Porto: FEUP, 2012.
- [31] L. Oliveira, “Maquinagem de superfícies complexas com recurso a sistema robótico,” Porto, 2013.
- [32] “ATI Industrial Automation,” ATI, 9 Junho 2015. [Online]. Available: http://www.ati-ia.com/products/ft/ft_models.aspx?id=Delta+IP60.
- [33] J.-P. MERLET, Parallel Robots (Second editon), Springer, 2006.
- [34] A. M. Lopes, Modelação Cinemática e Dinâmica de Manipuladores de Estrutura em Série, Porto: FEUP, 2001.
- [35] ABB, ABB in brief, ABB [Online] ,<http://new.abb.com/about/abb-in-brief>, 2014.
- [36] S. T. Inc., Numerical Control Machining, 2014.

Anexos

Anexo A: Componentes da célula robótica

Datasheet robô IRB 2400

IRB 2400

Main applications

Arc welding, cutting/deburring, glueing/sealing, grinding/polishing, machine tending, material handling

Specification

Variants	Reach	Payload	Protection
IRB 2400-10	1.55 m	12 kg	FoundryPlus
IRB 2400-16	1.55 m	20 kg	FoundryPlus

Number of axes	6
Mounting	Floor and inverted all versions. Wall IRB 2400-10

Supplementary load	IRB 2400-10	IRB 2400-16
Upper arm		
-wrist end	2 kg	2 kg
-rear end	10 kg	10 kg
Base unit	35 kg	35 kg

Physical

Dimensions	IRB 2400-10	IRB 2400-16
Total height	1564 mm	1564 mm
Robot base	723x600 mm	723x600 mm
Robot weight	380 kg	380 kg

Performance (according to ISO 9283)

Position repeatability (RP)	0.03 mm
Path repeatability	0.11-0.15 mm depending on variant

Movements

Axis	IRB 2400-10	IRB 2400-16
Working range		
Axis 1	360**	360*
Axis 2	210*	210*
Axis 3	125*	125*
Axis 4	400*	400*
Axis 4, Option	Unlimited	Unlimited
Axis 5	240*	240*
Axis 6	800*	800*
Axis 6, Option	Unlimited	Unlimited

Max. speed

Axis 1	150*/s	150*/s
	90* **	
Axis 2	150*/s	150*/s
	90* **	
Axis 3	150*/s	150*/s
	90* **	
Axis 4	360*/s	360*/s
Axis 5	360*/s	360*/s
Axis 6	450*/s	450*/s

***) For wall mounted version

*) ± 30° for wall mounted robot

Electrical Connections

Supply voltage	200-600 V, 50/60 Hz
Power consumption	ISO-Cube at max speed 0.67 kW

Environment

Ambient temperature for mechanical unit:	
During operation	+5°C (41°F) to +45°C (113°F)
Relative humidity	Max. 95%

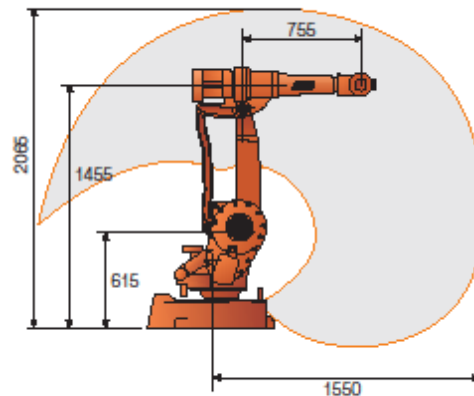
Degree of protection	IRB 2400-10	IRB 2400-16
Standard	IP54	IP54
Foundry Plus 2 version	IP67	IP67

Noise level	Max. 70 dB (A)
Emission	EMC/EMI-shielded

Data and dimensions may be changed without notice

Working range

IRB 2400-10, IRB 2400-16



© Copyright ABB Robotics. PR10034_EN_R7 June 2012.

Datasheet da mesa posicionadora IRBP C-500

IRBP C

Positioner Range

General Features

All ABB positioners gives users a complete and efficient solution. They are of a robust construction to ensure excellent stability.

All axes are fully coordinated with the robot when programming as well as during operation. The positioners are easy to use with clear, simple instructions for programming.

They use the same drive systems and software as the robot. The dynamic software (HPP, High Performance Positioner) automatically compensates for the effects of gravity, inertia and friction to provide fast movements. The robot software, Load Id, is used for calculating. The positioners are used for manipulating workpieces during arc welding, thermal cutting or other applications.

To meet users' requirements they can be supplied or retrofitted with a cable for bus communication (Profibus, DeviceNet, Interbus etc) and tubes for

pneumatic and hydraulic media transfer. ABB's positioners are designed to be highly functional yet compact thus making maximum use of available floor space.

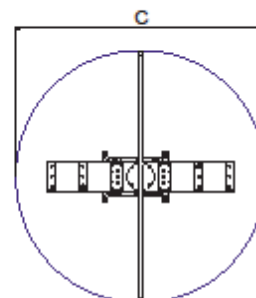
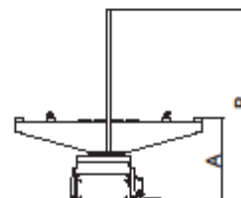
Standardised dimensions for all rotating plates greatly simplifies exchanging fixtures.

ABB positioners can be supplied with all necessary safety equipment. Users can get their positioner really soon as all ABB positioners comes with a very short delivery time.

		500C	1000C
Handling capacity	kg	500	1000
Max inertia	kgm ²	880	1600
Max bending torque	Nm	5000	15000
Min interchange time	s	3.3	3.1
Max interchange time	s	4	4
Repeatability (r=500)	mm	0.1	0.1
Max load diff	kg	350	500

Positioner 500C		
A	B	C
800	1852	2450

Positioner 1000C		
A	B	C
900	1952	2930



RR1013EN, 02 April 2003.
 Copyright © 2003 ABB Robotics. All Rights Reserved.

www.abb.com/robotics

ABB reserves the right to change specifications without notice.

SE-695 82 Laxå Sweden, Tel +46 584 820 00, Fax +46 584 131 80



Datasheet Sistema pneumático de mudança rápida de ferramenta SCHUNK SWS - 011

SWS-011

Tool Changing · Quick-change System · Light Load



Product description

Light, small and compact with extremely high locking force

No-Touch-Locking™

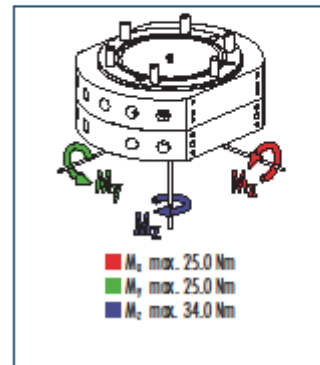
Locking without touching. Ensures that the SWS is securely locked even when the SWK and SWA do not touch. A maximum distance of 3 mm is possible.

Patented, self-retaining locking system

A larger piston diameter and the OD locking mechanism increase the permissible moment load. Steel components made from stainless Rc 58.

Air feed-through with specially developed rubber seals

Moment load



Ⓞ The dynamic moment load can be up to three times larger than the static moment load. Tests have shown that the system will only begin to fail in the event of 12-fold static moment.

Technical data

Designation		SWS-011	
Maximum payload	[kg]	16	A larger payload is possible with smaller moments
Static moment load M_x	[Nm]	25	
Static moment load M_y	[Nm]	34	
Dynamic moment load M_x	[Nm]	75	
Dynamic moment load M_y	[Nm]	102	
Locking force (at 6 bar)	[N]	1068	In the event of higher tensile forces the system "falls" into the self-locking position
Repeat accuracy	[mm]	0.01	Tested at 1 million cycles
Weight	[kg]	0.21	0.13 kg head; 0.08 kg adapter
Max. distance on locking	[mm]	3.0	No-TouchLocking™ technology allows the parts to be coupled without the head and the adapter touching
Pneumatic energy transmission		6x pneumatic M5	Max. 7 bar
Maximum permissible XY offset	[mm]	± 1	Maximum permissible XY offset when locking
Maximum permissible angular offset	[°]	± 2	Maximum permissible angular offset around the Z axis when locking

Information on moment load

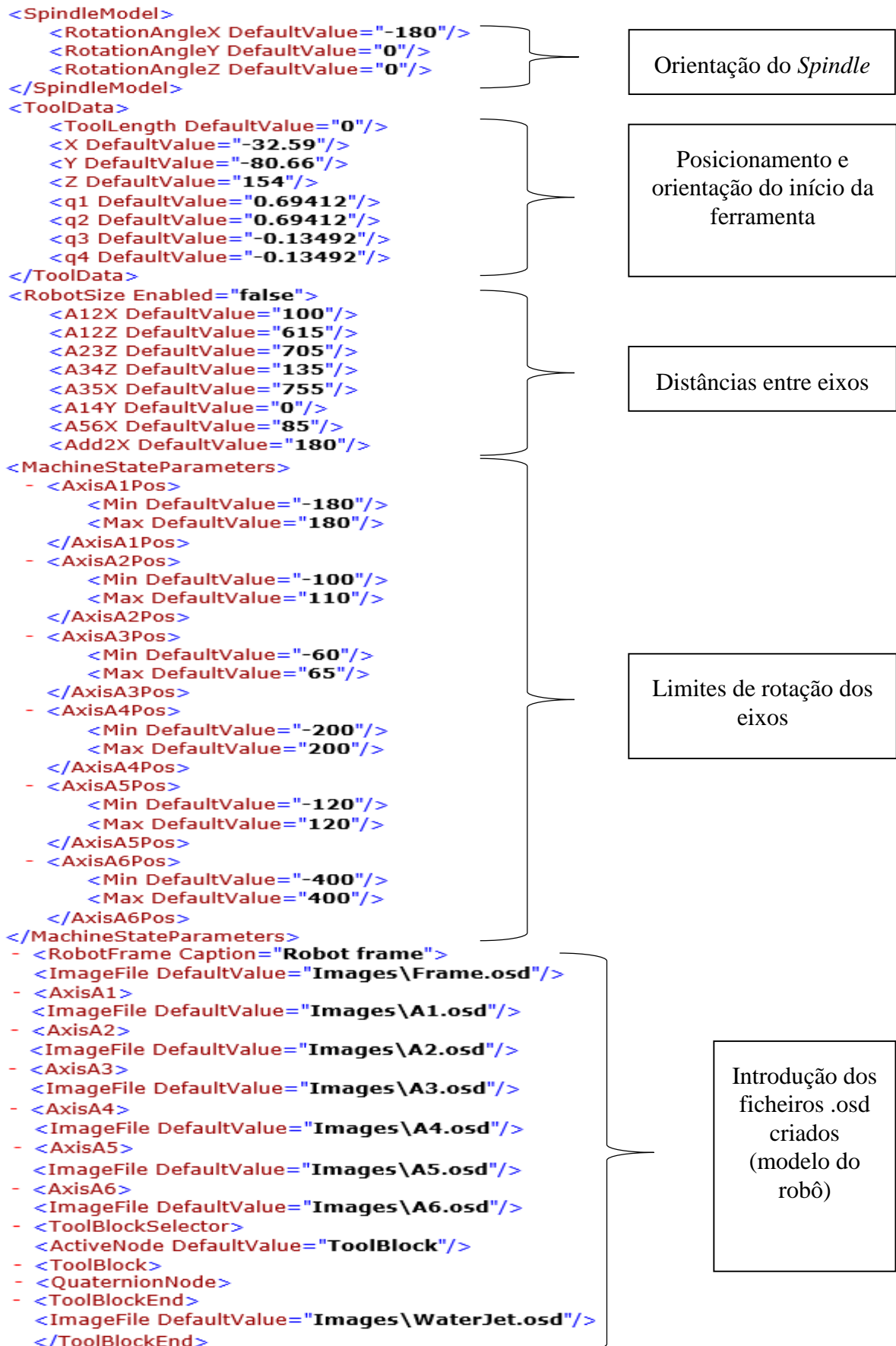
Selecting the correct quick-change system depends on the moment load which the system is subject to.

Proceed as follows to calculate the maximum moments.

- Determine the center of gravity and the weight (m in Newtons) of your heaviest tool (gripper, adapter plate and tool)
- Determine the distance (D) in meters) from the center of gravity to the underside of the quick-change adapter (SWA)
- Calculate the static moment (m x D)
- Select a quick-change system with a permissible moment that is equal to or greater than the moment you have calculated

Robot movements can also have an effect on the change system. Dynamic moments can come into effect which are 2 - 3 times greater than the static moments you have calculated. The SWS quick-change systems are designed for handling dynamic moments which can be up to three times greater than the static moments.

Anexo B: Alterações no ficheiro de definições do robô no *software* Sprutcam



Anexo C: Programas gerados e editados no Sub-capítulo 4.2

Estratégia Paralela Robotstudio.

```

MODULE Main
  PERS tooldata XLC70:=[TRUE,[-79.6979,-193.211,172.676]],[0.692911404,0.692911404,-0.140974418,-
0.140974418]],[11.4,[18.9,47.7,124.1],[1,0,0,0],[0,0,0]];
  PERS wobjdata Wobj_S_1:=[FALSE,TRUE,"",[[0,0,0],[1,0,0,0]],[[1000,-250,430],[1,0,0,0]];
  PROC main()
    Solution_1;
  ENDPROC
ENDMODULE
MODULE Surface_1
  CONST robtarg p4:=[-75.076530612,33.444503015,80],[0,1,0,0],[0,1,0,0],[9E9,9E9,9E9,9E9,9E9,9E9];
  CONST robtarg p5:=[-55.076530612,33.444503015,60],[0,1,0,0],[0,1,0,0],[9E9,9E9,9E9,9E9,9E9,9E9];
  CONST robtarg p6:=[-35.076530612,33.444503015,40],[0,1,0,0],[0,1,0,0],[9E9,9E9,9E9,9E9,9E9,9E9];
  CONST robtarg p1:=[-35.076530612,33.444503015,30],[0,1,0,0],[0,1,0,0],[9E9,9E9,9E9,9E9,9E9,9E9];
  CONST robtarg p2:=[14.923469388,33.444503015,30],[0,1,0,0],[0,1,0,0],[9E9,9E9,9E9,9E9,9E9,9E9];
  CONST robtarg p10:=[14.923469388,28.644503015,30],[0,1,0,0],[0,1,0,0],[9E9,9E9,9E9,9E9,9E9,9E9];
  CONST robtarg p11:=[-35.076530612,28.644503015,30],[0,1,0,0],[0,1,0,0],[9E9,9E9,9E9,9E9,9E9,9E9];
  CONST robtarg p19:=[-35.076530612,23.844503015,30],[0,1,0,0],[0,1,0,0],[9E9,9E9,9E9,9E9,9E9,9E9];
  CONST robtarg p20:=[14.923469388,23.844503015,30],[0,1,0,0],[0,1,0,0],[9E9,9E9,9E9,9E9,9E9,9E9];
  CONST robtarg p28:=[14.923469388,19.044503015,30],[0,1,0,0],[0,1,0,0],[9E9,9E9,9E9,9E9,9E9,9E9];
  CONST robtarg p29:=[-35.076530612,19.044503015,30],[0,1,0,0],[0,1,0,0],[9E9,9E9,9E9,9E9,9E9,9E9];
  CONST robtarg p37:=[-35.076530612,14.244503015,30],[0,1,0,0],[0,1,0,0],[9E9,9E9,9E9,9E9,9E9,9E9];
  CONST robtarg p38:=[14.923469388,14.244503015,30],[0,1,0,0],[0,1,0,0],[9E9,9E9,9E9,9E9,9E9,9E9];
  CONST robtarg p46:=[14.923469388,9.444503015,30],[0,1,0,0],[0,1,0,0],[9E9,9E9,9E9,9E9,9E9,9E9];
  CONST robtarg p47:=[-35.076530612,9.444503015,30],[0,1,0,0],[0,1,0,0],[9E9,9E9,9E9,9E9,9E9,9E9];
  CONST robtarg p55:=[-35.076530612,4.644503015,30],[0,1,0,0],[0,1,0,0],[9E9,9E9,9E9,9E9,9E9,9E9];
  CONST robtarg p56:=[14.923469388,4.644503015,30],[0,1,0,0],[0,1,0,0],[9E9,9E9,9E9,9E9,9E9,9E9];
  CONST robtarg p64:=[14.923469388,-0.155496985,30],[0,1,0,0],[0,1,0,0],[9E9,9E9,9E9,9E9,9E9,9E9];
  CONST robtarg p65:=[-35.076530612,-0.155496985,30],[0,1,0,0],[0,1,0,0],[9E9,9E9,9E9,9E9,9E9,9E9];
  CONST robtarg p73:=[-35.076530612,-4.955496985,30],[0,1,0,0],[0,1,0,0],[9E9,9E9,9E9,9E9,9E9,9E9];
  CONST robtarg p74:=[14.923469388,-4.955496985,30],[0,1,0,0],[0,1,0,0],[9E9,9E9,9E9,9E9,9E9,9E9];
  CONST robtarg p82:=[14.923469388,-9.755496985,30],[0,1,0,0],[0,1,0,0],[9E9,9E9,9E9,9E9,9E9,9E9];
  CONST robtarg p83:=[-35.076530612,-9.755496985,30],[0,1,0,0],[0,1,0,0],[9E9,9E9,9E9,9E9,9E9,9E9];
  CONST robtarg p91:=[-35.076530612,-14.555496985,30],[0,1,0,0],[0,1,0,0],[9E9,9E9,9E9,9E9,9E9,9E9];
  CONST robtarg p92:=[14.923469388,-14.555496985,30],[0,1,0,0],[0,1,0,0],[9E9,9E9,9E9,9E9,9E9,9E9];
  CONST robtarg p97:=[17.923469388,-14.555496985,43],[0,1,0,0],[0,1,0,0],[9E9,9E9,9E9,9E9,9E9,9E9];
  CONST robtarg p98:=[37.923469388,-14.555496985,63],[0,1,0,0],[0,1,0,0],[9E9,9E9,9E9,9E9,9E9,9E9];
  CONST robtarg p99:=[57.923469388,-14.555496985,83],[0,1,0,0],[0,1,0,0],[9E9,9E9,9E9,9E9,9E9,9E9];
  PROC Path_1()
    MoveJ p4,[16.30,500,5000,1000],[FALSE,3,40,40,10,35,5],XLC70\WObj:=Wobj_S_1;
    MoveJ p5,[16.30,500,5000,1000],[FALSE,3,40,40,10,35,5],XLC70\WObj:=Wobj_S_1;
    MoveJ p6,[16.30,500,5000,1000],[FALSE,3,40,40,10,35,5],XLC70\WObj:=Wobj_S_1;
    MoveL p1,[16.30,500,5000,1000],[FALSE,3,40,40,10,35,5],XLC70\WObj:=Wobj_S_1;
    MoveL p2,[16.30,500,5000,1000],[FALSE,3,40,40,10,35,5],XLC70\WObj:=Wobj_S_1;
    MoveL p10,[16.30,500,5000,1000],[FALSE,3,40,40,10,35,5],XLC70\WObj:=Wobj_S_1;
    MoveL p11,[16.30,500,5000,1000],[FALSE,3,40,40,10,35,5],XLC70\WObj:=Wobj_S_1;
    MoveL p19,[16.30,500,5000,1000],[FALSE,3,40,40,10,35,5],XLC70\WObj:=Wobj_S_1;
    MoveL p20,[16.30,500,5000,1000],[FALSE,3,40,40,10,35,5],XLC70\WObj:=Wobj_S_1;
    MoveL p28,[16.30,500,5000,1000],[FALSE,3,40,40,10,35,5],XLC70\WObj:=Wobj_S_1;
    MoveL p29,[16.30,500,5000,1000],[FALSE,3,40,40,10,35,5],XLC70\WObj:=Wobj_S_1;
    MoveL p37,[16.30,500,5000,1000],[FALSE,3,40,40,10,35,5],XLC70\WObj:=Wobj_S_1;
    MoveL p38,[16.30,500,5000,1000],[FALSE,3,40,40,10,35,5],XLC70\WObj:=Wobj_S_1;
    MoveL p46,[16.30,500,5000,1000],[FALSE,3,40,40,10,35,5],XLC70\WObj:=Wobj_S_1;
    MoveL p47,[16.30,500,5000,1000],[FALSE,3,40,40,10,35,5],XLC70\WObj:=Wobj_S_1;
    MoveL p55,[16.30,500,5000,1000],[FALSE,3,40,40,10,35,5],XLC70\WObj:=Wobj_S_1;
    MoveL p56,[16.30,500,5000,1000],[FALSE,3,40,40,10,35,5],XLC70\WObj:=Wobj_S_1;
    MoveL p64,[16.30,500,5000,1000],[FALSE,3,40,40,10,35,5],XLC70\WObj:=Wobj_S_1;
    MoveL p65,[16.30,500,5000,1000],[FALSE,3,40,40,10,35,5],XLC70\WObj:=Wobj_S_1;
    MoveL p73,[16.30,500,5000,1000],[FALSE,3,40,40,10,35,5],XLC70\WObj:=Wobj_S_1;
    MoveL p74,[16.30,500,5000,1000],[FALSE,3,40,40,10,35,5],XLC70\WObj:=Wobj_S_1;
    MoveL p82,[16.30,500,5000,1000],[FALSE,3,40,40,10,35,5],XLC70\WObj:=Wobj_S_1;
    MoveL p83,[16.30,500,5000,1000],[FALSE,3,40,40,10,35,5],XLC70\WObj:=Wobj_S_1;
    MoveL p91,[16.30,500,5000,1000],[FALSE,3,40,40,10,35,5],XLC70\WObj:=Wobj_S_1;
    MoveL p92,[16.30,500,5000,1000],[FALSE,3,40,40,10,35,5],XLC70\WObj:=Wobj_S_1;
    MoveL p97,[16.30,500,5000,1000],[FALSE,3,40,40,10,35,5],XLC70\WObj:=Wobj_S_1;
    MoveL p98,[16.30,500,5000,1000],[FALSE,3,40,40,10,35,5],XLC70\WObj:=Wobj_S_1;
    MoveL p99,[16.30,500,5000,1000],[FALSE,3,40,40,10,35,5],XLC70\WObj:=Wobj_S_1;
  ENDPROC
  PROC Solution_1()
    Path_1;
  ENDPROC
ENDMODULE

```

Estratégia Paralela Sprutcam

```

MODULE ABB
PERS tooldata tool1:=[TRUE,[-79.6979,-193.211,172.676],[0.694,0.694,-0.135,-0.135]],[11.4,[18.9,47.7,124.1],[1, 0, 0, 0], 0, 0, 0]);
PERS wobjdata wobj1:=[FALSE,TRUE,"",[[1000,-250,430],[1,0,0,0]],[[0,0,0],[1,0,0,0]]];
PROC main()
ConfL \Off;
MoveABSJ [-15.489,17.279,45.692,180,62.971,-68],[9E+009,9E+009,9E+009,9E+009,9E+009,9E+009],[16,30,200,15],[FALSE,3,40,40,10,35,5],tool1\WObj:=wobj1;
MoveL [-24.984,-20.19,40],[0,-0.796,-0.605,-0],[1,1,-1,0],[9E+009,9E+009,9E+009,9E+009,9E+009,9E+009],[16,30,200,15],[FALSE,3,40,40,10,35,5],tool1\WObj:=wobj1;
MoveL [-24.984,-20.19,35],[0,-0.796,-0.605,-0],[1,1,-1,0],[9E+009,9E+009,9E+009,9E+009,9E+009,9E+009],[16,30,200,15],[FALSE,3,40,40,10,35,5],tool1\WObj:=wobj1;
MoveL [-24.984,-20.19,30],[0,-0.796,-0.605,-0],[1,1,-1,0],[9E+009,9E+009,9E+009,9E+009,9E+009,9E+009],[16,30,200,15],[FALSE,3,40,40,10,35,5],tool1\WObj:=wobj1;
MoveL [[24.989,-20.378,30],[0,-0.792,-0.61,-0],[1,1,-1,0],[9E+009,9E+009,9E+009,9E+009,9E+009,9E+009],[16,30,200,15],[FALSE,3,40,40,10,35,5],tool1\WObj:=wobj1;
MoveL [[24.986,-24.983,30],[0,-0.793,-0.609,-0],[1,1,-1,0],[9E+009,9E+009,9E+009,9E+009,9E+009,9E+009],[16,30,200,15],[FALSE,3,40,40,10,35,5],tool1\WObj:=wobj1;
MoveL [-24.983,-24.986,30],[0,-0.797,-0.604,-0],[1,1,-1,0],[9E+009,9E+009,9E+009,9E+009,9E+009,9E+009],[16,30,200,15],[FALSE,3,40,40,10,35,5],tool1\WObj:=wobj1;
MoveL [-24.986,-15.39,30],[0,-0.795,-0.607,-0],[1,1,-1,0],[9E+009,9E+009,9E+009,9E+009,9E+009,9E+009],[16,30,200,15],[FALSE,3,40,40,10,35,5],tool1\WObj:=wobj1;
MoveL [[24.989,-15.578,30],[0,-0.791,-0.612,-0],[1,1,-1,0],[9E+009,9E+009,9E+009,9E+009,9E+009,9E+009],[16,30,200,15],[FALSE,3,40,40,10,35,5],tool1\WObj:=wobj1;
MoveL [[24.988,-10.778,30],[0,-0.789,-0.614,-0],[1,1,-1,0],[9E+009,9E+009,9E+009,9E+009,9E+009,9E+009],[16,30,200,15],[FALSE,3,40,40,10,35,5],tool1\WObj:=wobj1;
MoveL [-24.985,-10.589,30],[0,-0.793,-0.609,-0],[1,1,-1,0],[9E+009,9E+009,9E+009,9E+009,9E+009,9E+009],[16,30,200,15],[FALSE,3,40,40,10,35,5],tool1\WObj:=wobj1;
MoveL [-24.986,-5.789,30],[0,-0.792,-0.611,-0],[1,1,-1,0],[9E+009,9E+009,9E+009,9E+009,9E+009,9E+009],[16,30,200,15],[FALSE,3,40,40,10,35,5],tool1\WObj:=wobj1;
MoveL [[24.988,-5.978,30],[0,-0.788,-0.616,-0],[1,1,-1,0],[9E+009,9E+009,9E+009,9E+009,9E+009,9E+009],[16,30,200,15],[FALSE,3,40,40,10,35,5],tool1\WObj:=wobj1;
MoveL [[24.987,-1.178,30],[0,-0.787,-0.617,-0],[1,1,-1,0],[9E+009,9E+009,9E+009,9E+009,9E+009,9E+009],[16,30,200,15],[FALSE,3,40,40,10,35,5],tool1\WObj:=wobj1;
MoveL [-24.987,-0.989,30],[0,-0.79,-0.613,-0],[1,1,-1,0],[9E+009,9E+009,9E+009,9E+009,9E+009,9E+009],[16,30,200,15],[FALSE,3,40,40,10,35,5],tool1\WObj:=wobj1;
MoveL [-24.987,3.811,30],[0,-0.789,-0.614,-0],[1,1,-1,0],[9E+009,9E+009,9E+009,9E+009,9E+009,9E+009],[16,30,200,15],[FALSE,3,40,40,10,35,5],tool1\WObj:=wobj1;
MoveL [[24.986,3.622,30],[0,-0.785,-0.619,-0],[1,1,-1,0],[9E+009,9E+009,9E+009,9E+009,9E+009,9E+009],[16,30,200,15],[FALSE,3,40,40,10,35,5],tool1\WObj:=wobj1;
MoveL [[24.986,8.422,30],[0,-0.784,-0.621,-0],[1,1,-1,0],[9E+009,9E+009,9E+009,9E+009,9E+009,9E+009],[16,30,200,15],[FALSE,3,40,40,10,35,5],tool1\WObj:=wobj1;
MoveL [-24.988,8.611,30],[0,-0.788,-0.616,-0],[1,1,-1,0],[9E+009,9E+009,9E+009,9E+009,9E+009,9E+009],[16,30,200,15],[FALSE,3,40,40,10,35,5],tool1\WObj:=wobj1;
MoveL [-24.988,13.411,30],[0,-0.786,-0.618,-0],[1,1,-1,0],[9E+009,9E+009,9E+009,9E+009,9E+009,9E+009],[16,30,200,15],[FALSE,3,40,40,10,35,5],tool1\WObj:=wobj1;
MoveL [[24.985,13.222,30],[0,-0.783,-0.622,-0],[1,1,-1,0],[9E+009,9E+009,9E+009,9E+009,9E+009,9E+009],[16,30,200,15],[FALSE,3,40,40,10,35,5],tool1\WObj:=wobj1;
MoveL [[24.984,18.022,30],[0,-0.781,-0.624,-0],[1,1,-1,0],[9E+009,9E+009,9E+009,9E+009,9E+009,9E+009],[16,30,200,15],[FALSE,3,40,40,10,35,5],tool1\WObj:=wobj1;
MoveL [-24.989,18.211,30],[0,-0.785,-0.62,-0],[1,1,-1,0],[9E+009,9E+009,9E+009,9E+009,9E+009,9E+009],[16,30,200,15],[FALSE,3,40,40,10,35,5],tool1\WObj:=wobj1;
MoveL [-24.99,23.011,30],[0,-0.783,-0.622,-0],[1,1,-1,0],[9E+009,9E+009,9E+009,9E+009,9E+009,9E+009],[16,30,200,15],[FALSE,3,40,40,10,35,5],tool1\WObj:=wobj1;
MoveL [[24.984,22.822,30],[0,-0.78,-0.626,-0],[1,1,-1,0],[9E+009,9E+009,9E+009,9E+009,9E+009,9E+009],[16,30,200,15],[FALSE,3,40,40,10,35,5],tool1\WObj:=wobj1;
MoveL [[24.983,24.986,30],[0,-0.779,-0.627,-0],[1,1,-1,0],[9E+009,9E+009,9E+009,9E+009,9E+009,9E+009],[16,30,200,15],[FALSE,3,40,40,10,35,5],tool1\WObj:=wobj1;
MoveL [-24.986,24.983,30],[0,-0.783,-0.623,-0],[1,1,-1,0],[9E+009,9E+009,9E+009,9E+009,9E+009,9E+009],[16,30,200,15],[FALSE,3,40,40,10,35,5],tool1\WObj:=wobj1;
MoveL [-24.99,23.011,30],[0,-0.783,-0.622,-0],[1,1,-1,0],[9E+009,9E+009,9E+009,9E+009,9E+009,9E+009],[16,30,200,15],[FALSE,3,40,40,10,35,5],tool1\WObj:=wobj1;
MoveL [-24.99,23.011,40],[0,-0.783,-0.622,-0],[1,1,-1,0],[9E+009,9E+009,9E+009,9E+009,9E+009,9E+009],[16,30,200,15],[FALSE,3,40,40,10,35,5],tool1\WObj:=wobj1;
MoveL [-24.99,23.011,40],[0,-0.783,-0.622,-0],[1,1,-1,0],[9E+009,9E+009,9E+009,9E+009,9E+009,9E+009],[16,30,200,15],[FALSE,3,40,40,10,35,5],tool1\WObj:=wobj1;
MoveABSJ
[[0,0,0,0,0,0],[9E+009,9E+009,9E+009,9E+009,9E+009,9E+009],[16,30,200,15],[FALSE,3,40,40,10,35,5],tool1\WObj:=wobj1;
ENDPROC
ENDMODULE

```

Estratégia Paralela RobotMaster

MODULE P0

!GENERATED BY ROBOTMASTER

PERS wobjdata wob1:=[FALSE,TRUE,"",[1000.00,-250.00,430.00],[1,0,0,0],[0,0,0],[1,0,0,0]];

PERS tooldata tl:=[TRUE,[[-73.42,-181.72,154.00],[0.1349192,-0.1349192,-0.6941159,0.6941159]],
[11.4,[18.9,47.7,124.1],[1,0,0,0],0,0,0]];

PROC MAIN()

AccSet 10,10;

ConfL\On;

SingArea\Wrist;

```

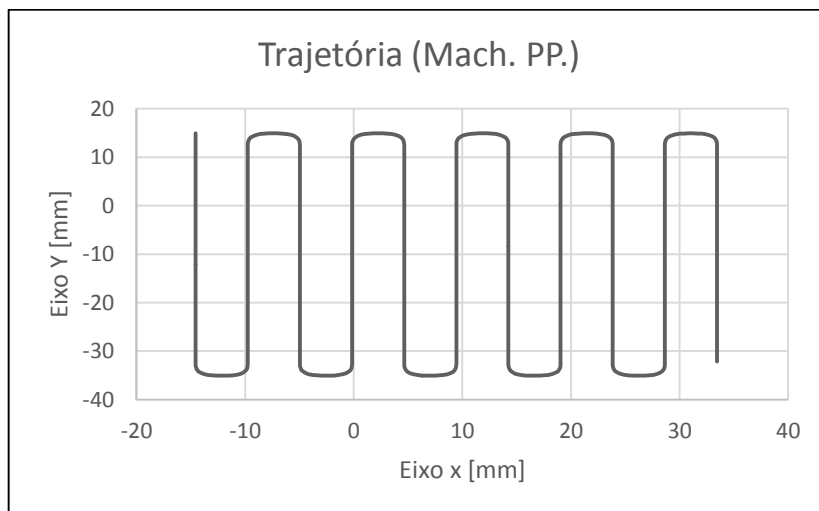
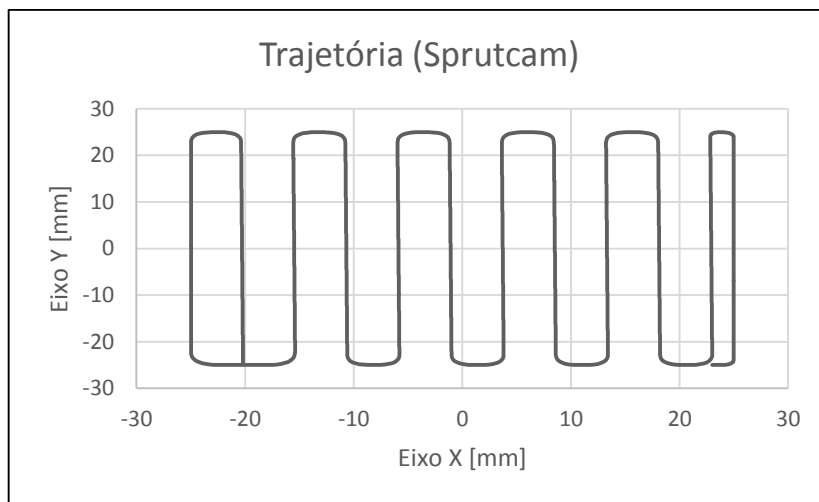
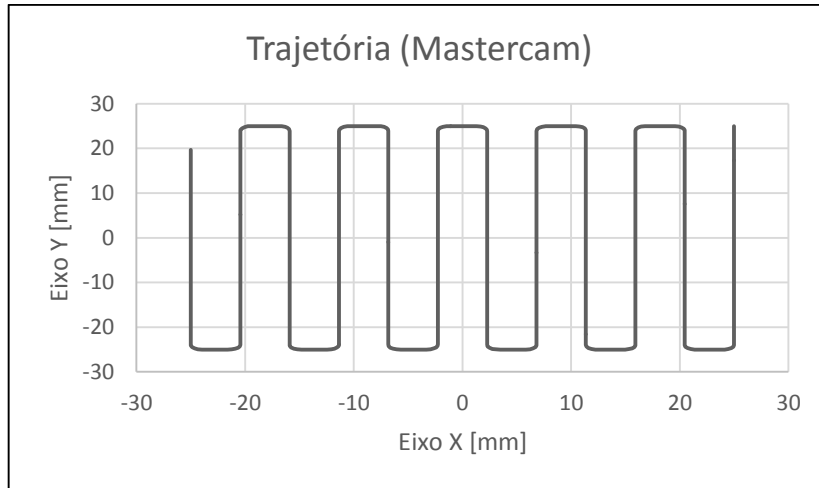
MoveL [[25.00,-25.00,250.00],[0,1,0,0],[-1,0,-
1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],v300,[FALSE,3,40,40,10,35,5],tl\_Wobj:=wob1;
MoveL [[25.00,-25.00,37.20],[0,1,0,0],[-1,0,-
1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],v300,[FALSE,3,40,40,10,35,5],tl\_Wobj:=wob1;
MoveL [[25.00,-25.00,30.20],[0,1,0,0],[-1,0,-
1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],[16.30,500,5000,1000],fine,tl\_Wobj:=wob1;
MoveL [[-25.00,-25.00,30.20],[0,1,0,0],[-1,0,-
1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],[16.30,500,5000,1000],[FALSE,3,40,40,10,35,5],tl\_Wobj:=wob1;
MoveL [[-25.00,-20.45,30.20],[0,1,0,0],[-1,0,-
1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],[16.30,500,5000,1000],[FALSE,3,40,40,10,35,5],tl\_Wobj:=wob1;
MoveL [[25.00,-20.45,30.20],[0,1,0,0],[-1,0,-
1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],[16.30,500,5000,1000],[FALSE,3,40,40,10,35,5],tl\_Wobj:=wob1;
MoveL [[25.00,-15.91,30.20],[0,1,0,0],[-1,0,-
1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],[16.30,500,5000,1000],[FALSE,3,40,40,10,35,5],tl\_Wobj:=wob1;
MoveL [[-25.00,-15.91,30.20],[0,1,0,0],[-1,0,-
1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],[16.30,500,5000,1000],[FALSE,3,40,40,10,35,5],tl\_Wobj:=wob1;
MoveL [[-25.00,-11.36,30.20],[0,1,0,0],[-1,0,-
1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],[16.30,500,5000,1000],[FALSE,3,40,40,10,35,5],tl\_Wobj:=wob1;
MoveL [[25.00,-11.36,30.20],[0,1,0,0],[-1,0,-
1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],[16.30,500,5000,1000],[FALSE,3,40,40,10,35,5],tl\_Wobj:=wob1;
MoveL [[25.00,-6.82,30.20],[0,1,0,0],[-1,0,-
1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],[16.30,500,5000,1000],[FALSE,3,40,40,10,35,5],tl\_Wobj:=wob1;
MoveL [[-25.00,-6.82,30.20],[0,1,0,0],[-1,0,-
1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],[16.30,500,5000,1000],[FALSE,3,40,40,10,35,5],tl\_Wobj:=wob1;
MoveL [[-25.00,-2.27,30.20],[0,1,0,0],[-1,0,-
1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],[16.30,500,5000,1000],[FALSE,3,40,40,10,35,5],tl\_Wobj:=wob1;
MoveL [[25.00,-2.27,30.20],[0,1,0,0],[-1,0,-
1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],[16.30,500,5000,1000],[FALSE,3,40,40,10,35,5],tl\_Wobj:=wob1;
MoveL [[25.00,2.27,30.20],[0,1,0,0],[-1,0,-
1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],[16.30,500,5000,1000],[FALSE,3,40,40,10,35,5],tl\_Wobj:=wob1;
MoveL [[-25.00,2.27,30.20],[0,1,0,0],[-1,0,-
1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],[16.30,500,5000,1000],[FALSE,3,40,40,10,35,5],tl\_Wobj:=wob1;
MoveL [[-25.00,6.82,30.20],[0,1,0,0],[-1,0,-
1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],[16.30,500,5000,1000],[FALSE,3,40,40,10,35,5],tl\_Wobj:=wob1;
MoveL [[25.00,6.82,30.20],[0,1,0,0],[-1,0,-
1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],[16.30,500,5000,1000],[FALSE,3,40,40,10,35,5],tl\_Wobj:=wob1;
MoveL [[25.00,11.36,30.20],[0,1,0,0],[-1,0,-
1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],[16.30,500,5000,1000],[FALSE,3,40,40,10,35,5],tl\_Wobj:=wob1;
MoveL [[-25.00,11.36,30.20],[0,1,0,0],[-1,0,-
1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],[16.30,500,5000,1000],[FALSE,3,40,40,10,35,5],tl\_Wobj:=wob1;
MoveL [[-25.00,15.91,30.20],[0,1,0,0],[-1,0,-
1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],[16.30,500,5000,1000],[FALSE,3,40,40,10,35,5],tl\_Wobj:=wob1;
MoveL [[25.00,15.91,30.20],[0,1,0,0],[-1,0,-
1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],[16.30,500,5000,1000],[FALSE,3,40,40,10,35,5],tl\_Wobj:=wob1;
MoveL [[25.00,20.45,30.20],[0,1,0,0],[-1,0,-
1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],[16.30,500,5000,1000],[FALSE,3,40,40,10,35,5],tl\_Wobj:=wob1;
MoveL [[-25.00,20.45,30.20],[0,1,0,0],[-1,0,-
1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],[16.30,500,5000,1000],[FALSE,3,40,40,10,35,5],tl\_Wobj:=wob1;
MoveL [[-25.00,25.00,30.20],[0,1,0,0],[-1,0,-
1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],[16.30,500,5000,1000],[FALSE,3,40,40,10,35,5],tl\_Wobj:=wob1;
MoveL [[25.00,25.00,30.20],[0,1,0,0],[-1,0,-
1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],[16.30,500,5000,1000],fine,tl\_Wobj:=wob1;
MoveL [[25.00,25.00,35.20],[0,1,0,0],[-1,0,-
1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],[16.30,500,5000,1000],fine,tl\_Wobj:=wob1;
MoveL [[25.00,25.00,250.00],[0,1,0,0],[-1,0,-1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],v300,fine,tl\_Wobj:=wob1;
ConfL\Off;
ENDPROC
ENDMODULE

```

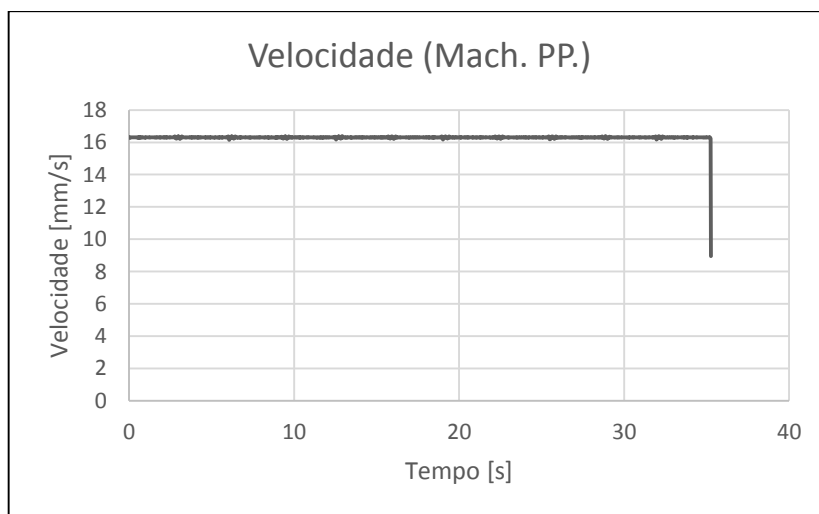
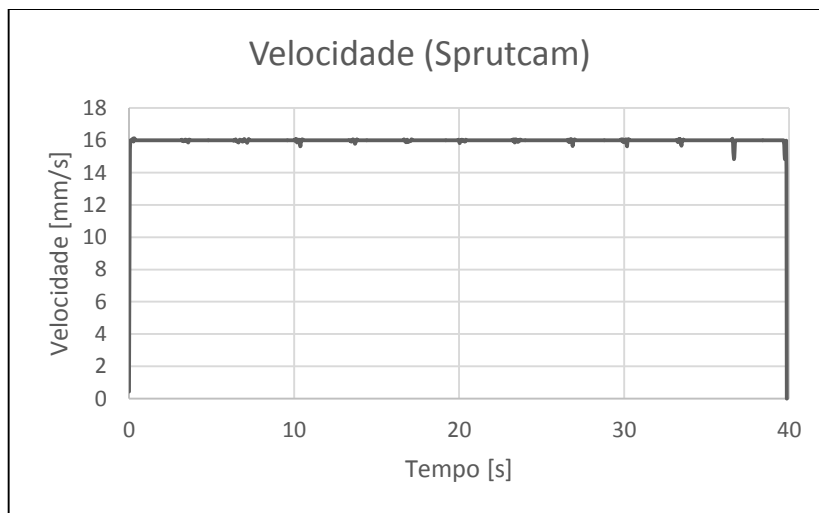
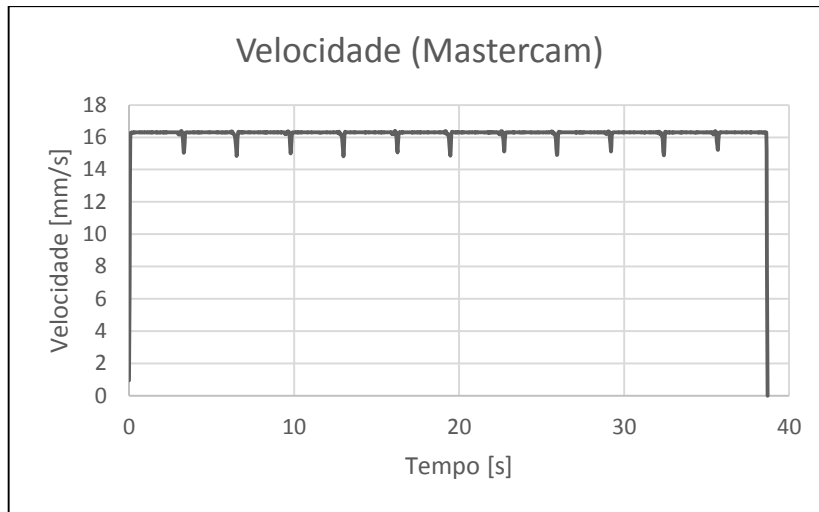
Dados obtidos pela funcionalidade **Signal Analyzer** nas diferentes estratégias de maquinagem do subcapítulo 4.2.

Estratégia Paralela

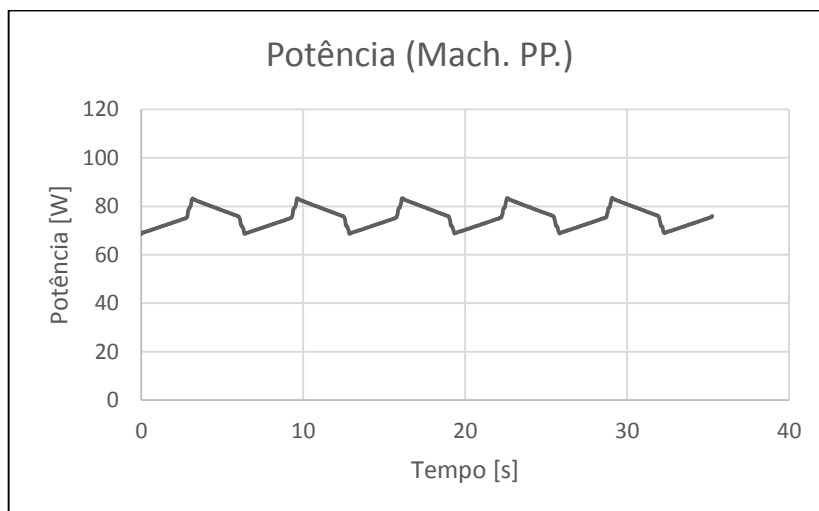
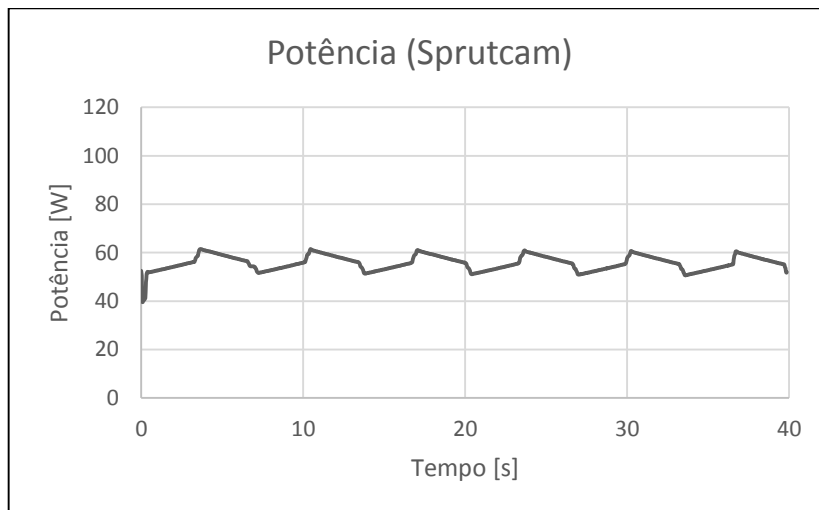
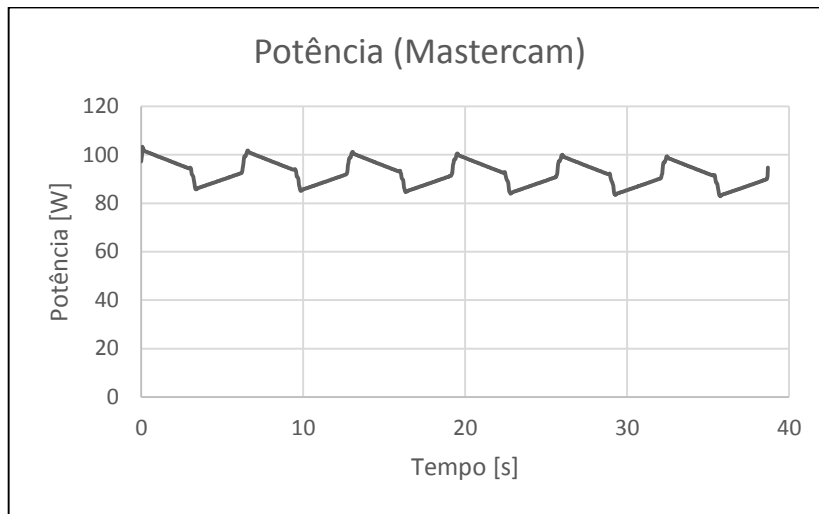
Trajétórias geradas:



Perfis de velocidade:



Potência:



Energia consumida:

