

TRABALHO DE GRADUAÇÃO

DESENVOLVIMENTO DE SOFTWARE DE FATIAMENTO DE SÓLIDOS TIPO CASCA E GERAÇÃO DE TRAJETÓRIAS PARA FABRICAÇÃO DE PEÇAS POR DEPOSIÇÃO DE METAL EM CAMADAS SUCESSIVAS UTILIZANDO O PROCESSO GMAW

Por, Rodrigo Castro Andrade

Brasília, abril de 2013



TRABALHO DE GRADUAÇÃO

DESENVOLVIMENTO DE SOFTWARE DE FATIAMENTO DE SÓLIDOS TIPO CASCA E GERAÇÃO DE TRAJETÓRIAS PARA FABRICAÇÃO DE PEÇAS POR DEPOSIÇÃO DE METAL EM CAMADAS SUCESSIVAS UTILIZANDO O PROCESSO GMAW

POR

Rodrigo Castro Andrade

Relatório submetido como requisito parcial para obtenção do grau de Engenheiro de Controle e Automação

Banca Examinadora

Prof. Ph.D Guilherme Caribé de Carvalho	
UnB/ENM (Orientador)	
Prof. Ph.D Sadek Crisóstomo Absi Alfaro UnB/ENM	
Prof. Ph.D José Maurício Santos Torres da Motta UnB/ENM	

Brasília, abril de 2013

FICHA CATALOGRÁFICA

ANDRADE, RODRIGO CASTRO Desenvolvimento de software de fatiamento de sólidos tipo casca e geração de trajetórias para		
fabricação de peças por deposição de metal em camadas sucessivas utilizando o processo GMAW		
[Distrito Federal] 2013.		
xiii, 82p., 297mm (FT/UnB, Engenheiro, Controle e Automação, 2013). Trabalho de Graduação – Universidade de Brasília.Faculdade de Tecnologia.		
1.Robótica 3.Soldagem	2.Prototipagem Rápida 4.Software	
I. Mecatrônica/FT/UnB	II. Título (série)	

REFERÊNCIA BIBLIOGRÁFICA

ANDRADE, R. C., (2013). Desenvolvimento de software de fatiamento de sólidos tipo casca e geração de trajetórias para fabricação de peças por deposição de metal em camadas sucessivas utilizando o processo *GMAW*. Trabalho de Graduação em Engenharia de Controle e Automação, Publicação FT. TG-nº 20/2012, Faculdade de Tecnologia, Universidade de Brasília, Brasília, DF, 82p.

CESSÃO DE DIREITOS

AUTOR: Rodrigo Castro Andrade.

TÍTULO DO TRABALHO DE GRADUAÇÃO: Desenvolvimento de software de fatiamento de sólidos tipo casca e geração de trajetórias para fabricação de peças por deposição de metal em camadas sucessivas utilizando o processo *GMAW*.

GRAU: Engenheiro de Controle e Automação ANO: 2013

É concedida à Universidade de Brasília permissão para reproduzir cópias deste Trabalho de Graduação e para emprestar ou vender tais cópias somente para propósitos acadêmicos e científicos. O autor reserva outros direitos de publicação e nenhuma parte desse Trabalho de Graduação pode ser reproduzida sem autorização por escrito do autor.

Rodrigo Castro Andrade

Condomínio Jardim das Paineiras, quadra 5, casa 71 – Jardim Botânico. 71680-366 Brasília – DF – Brasil.

AGRADECIMENTOS

À minha família, pelo apoio ao longo desses desafiadores anos no curso.

À todos os meus amigos que me acompanharam e incentivaram frente as diversas dificuldades naturalmente encontradas em um curso de engenharia.

Ao meu orientador, professor Guilheme Caribé de Carvalho, pelo auxílio e excelente orientação ao longo deste projeto.

Aos técnicos e alunos de graduação e pós-graduação do GRACO, que me ofereceram grande ajuda nas etapas finais deste trabalho.

E a todos que direta ou indiretamente contribuíram para o êxito desta missão.

Rodrigo Castro Andrade.

RESUMO

Este trabalho teve como objetivo o desenvolvimento de um programa de software capaz de gerar pontos de trajetória para o movimento de um robô soldador de modo que o mesmo reproduza, ao depositar cordões de solda em camadas, a geometria de uma peça tipo casca originalmente representada como um modelo computacional em 3 dimensões (deposição de metal em camadas sucessivas). O desenvolvimento do trabalho incluiu a definição da estratégia de fatiamento do corpo sólido, estratégias para planejamento do processo de construção da peça, e atribuição de orientação e velocidade a uma sequência de pontos pelos quais o efetuador do robô deverá passar para reconstruir fisicamente o corpo ao depositar o metal de solda. O software utiliza a plataforma de modelamento geométrico tridimensional *ACIS®*, gerando um arquivo texto com pontos da trajetória para programação off-line de um robô *ABB® IRB 2000*. O trabalho é finalizado com a realização de ensaios de deposição com o programa gerado, seguida de medições dos erros geométricos do sólido depositado em relação ao sólido projetado.

Palavras-Chave: deposição de metal em camadas sucessivas, prototipagem rápida, *GMAW*, soldagem robotizada

ABSTRACT

This project had as goal the development of a software program capable of generating trajectory points for the guiding of a welding robot so that it may reproduce, by depositing beads of weld in layers, the geometry of a shell-like object initially represented as a computational model in three dimensions (Shaped Metal Deposition - SMD). The project's development included the definition of the strategy for solid-body slicing, strategies for planning the object's construction process, and assignment of orientation and velocity to a sequence of points through which the robot's end-effector will pass to physically rebuild the body by depositing the metal weld. The software uses the three-dimensional geometric modeling engine ACIS®, creating a text file with the trajectory points for the offline programming of an ABB® IRB 2000 robot. The project is finalized with the execution of deposition experiments using the program created, followed by geometric error measurements of the deposited solid relative to the designed solid.

Keywords: Shaped Metal Deposition, rapid prototyping, GMAW, robotic welding

SUMÁRIO

1 - INTRODUÇÃO	1
1.1 - MOTIVAÇÃO	1
1.2 - OBJETIVO	2
1.3 - ESTRUTURA DO DOCUMENTO	2
2 - FUNDAMENTAÇÃO TEÓRICA	3
2.1 - SOLDAGEM	3
2.1.1 - Processo de soldagem GMAW	3
2.1.2 - Robotização de processos de soldagem	5
2.2 - PROTOTIPAGEM RÁPIDA	6
2.2.1 - Deposição de metal em camadas sucessivas	7
2.3 - PLATAFORMA ACIS®	9
2.3.1 - Interface para Scheme	11
2.4 - LINGUAGEM DE PROGRAMAÇÃO SCHEME	12
2.4.1 - Gerando programas em Scheme com ACIS	12
2.5 - ROBÔS MANIPULADORES	13
2.5.1 - ROBÔ <i>ABB</i> ® <i>IRB</i> 2000	14
2.6 - REPRESENTAÇÕES DE ORIENTAÇÃO NO ESPAÇO 3D	16
2.6.1 - RPY	17
2.6.2 - Quatérnions	18
2.7 - SOFTWARE WORKSPACE LT®	19
2.8 - SOFTWARE SPORT ^{S3} ®	20
2.9 - INTERFACE ROB5000®	21
2.9.1 - Entradas Digitais	21
3 - DESENVOLVIMENTO	23
3.1 - ESTRATEGIAS DE FATIAMENTO DO MODELO 3D	23
3.1.1 - Seleção final da estratégia	25
3.1.2 - Determinação do eixo de crescimento do modelo	27
3.1.3 - Obtenção das orientações para mesa posicionadora	
3.1.4 - Interpolação dos pontos da trajetoria	29
3.1.5 - Transformações de rotação dos fatiamentos	30
3.1.6 - Estrategias para o processo de construção das peças	
3.2 - DESENVOLVIMENTO DO APLICATIVO DE FATIAMENTO	31
3.2.1 - Organização do codigo	31
	33
3.3 - AMBIENTE PARA SIMULAÇAU CUM WURKSPACE LT®	37
3.5 - CONFIGURAÇÕES GERAIS PARA ENSAIOS DE DEPOSIÇÃO	41
3.5.1 - Connigurações do Tobo e da Tonte de alimentação	41 11
5.5.2 - Vallavels do processo de soldagem	41
4 - RESULTADOS	43
4 1 - INTERFACE COM O USUÁRIO	
4.2 - FATIAMENTO DOS MODELOS PROPOSTOS	43 47
4.2.1 - Modelos simples	
······································	vi

4.2.2 - Modelos com furos	51
4.2.3 - Modelos com mesa posicionadora	53
4.3 - SIMULAÇÕES COM WORKSPACE LT®	56
4.4 - ENSAIOS DE DEPOSIÇÃO	61
4.4.1 - Comparações com modelos 3D	65
5 - CONCLUSÕES	68
5.1 - DISCUSSÃO DOS RESULTADOS	68
5.2 - TRABALHOS FUTUROS	70
REFERÊNCIAS BIBLIOGRÁFICAS	71
APÊNDICES	74
APÊNDICE I - Código em C++ para atualização do endereço atual dos arquivos	75
APÊNDICE II - Código em "acisinit.scm"	77
APÊNDICE III - Exemplo de código com definição da trajetória gerado para simulação	no
WorkspaceLT	78
APÊNDICE IV - Exemplo de código para execução do processo gerado para simulaçã	io no
WorkspaceLT	79
APÊNDICE V - Exemplo de código gerado para envio ao robô pelo Sport ^{S3}	81
APÊNDICE VI - Exemplo de orientações para envio à mesa posicionadora	82

LISTA DE FIGURAS

2.1	Esquema do processo de soldagem GMAW [10] 4
2.2	Equipamento para soldagem GMAW [12] 4
2.3	Representação de um modelo esférico e o possível resultado de camadas excessivamente
	grandes na fabricação por prototipagem rápida [20]7
2.4	Célula de soldagem 3D com uso do processo TIG [4] 8
2.5	Representação esquemática do processo SMD [5]
2.6	Diagrama de dependências entre as bibliotecas fundamentais do ACIS [24]10
2.7	Diagrama de derivações das classes de objeto para um modelo em AC/S [24]10
2.8	Interface de um aplicativo em Scheme para ACIS [24]11
2.9	Desenho do IRB 2000, ilustrando as diferentes partes, motores e eixos de rotação [28]14
2.10	Gabinete de controle do IRB 2000 [29]15
2.11	Sistemas de coordenadas (ou frames) são fixados ao manipulador e aos objetos do ambiente,
	além de considerar o sistema de referência [19]16
2.12	Componentes da matriz de transformação 3D [27]16
2.13	Rotações RPY e correpondência usual no punho de um manipulador [27]17
2.14	Área de trabalho do software Workspace LT, onde pode-se visualizar uma célula de trabalho
	robotizada desenvolvida projetada pelo usuário [34]19
2.15	Interface de usuário do Sport ^{S3} [37]20
3.1	Exemplo de modelo 3D fatiado por planos paralelos: (a) modelo original; (b) modelo pós-
	fatiamento; (c) uma única camada do modelo fatiado [3]23
3.2	Vista do conjunto espiral de semi-retas [10]24
3.3	Corte superior de um fatiamento de um modelo com protuberância longa [10]25
3.4	Corte superior de um fatiamento de um modelo com protuberância crescendo ao longo da
	direção de uma semi-reta25
3.5	Ilustração de exemplo de um ciclo de fatiamentos. A distância entre planos de fatiamento está
	deliberadamente exagerada para facilitar a visualização27
3.6	Determinação do eixo de crescimento na região entre as seções de fatiamento
3.7	Diagrama exibindo a organização do código do software desenvolvido32
3.8	Modelo 3D de um cubo simples, salvo em formato ACIS como "cubo.sat"
3.9	Modelo 3D de um cilindro simples, salvo em formato ACIS como "cilindro.sat"
3.10	Modelo 3D de uma pirâmide, salvo em formato ACIS como "piramide.sat"
3.11	Modelo 3D de uma peça com crescimento no sentido de semi-retas partindo da origem, salvo
	em formato ACIS como "comp1.sat"
3.12	Modelo 3D de um cubo com furos, salvo em formato ACIS como "cilindro 2furos, sat"35
3.13	Modelo 3D de uma pirâmide com furo em uma lateral, salvo em formato ACIS como

3.14	Modelo 3D de uma peça com eixo de crescimento não coincidente com o eixo vertical, salvo
	em formato ACIS como "inclinado1.sat"
3.15	Modelo 3D com surgimento de protuberancia fora da base, salvo em formato ACIS como
	"protuberância.sat"
3.16	Modelo 3D utilizado para o ensaio principal de deposição, salvo em formato ACIS como
	"piramideS.sat"
3.17	Modelo 3D utilizado no ensaio auxiliar de deposição, salvo em formato ACIS como
	"bloco_reduzido.sat"
3.18	Célula de soldagem desenvolvida no software Workspace LT para simular ensaios de
	deposição38
3.19	Propriedades do robô inseridas no software <i>Sport</i> ^{S3}
3.20	Formato de um projeto do Sport ^{S3}
4.1	Etapa de aquisição do nome do modelo computacional 3D a ser fatiado43
4.2	Etapa de inicialização do Hoops 3D43
4.3	Etapa de exibição do modelo computacional 3D44
4.4	Etapa de aquisição da distância entre planos de fatiamento adjacentes45
4.5	Etapa de execução do fatiamento45
4.6	Etapa de aquisição da precisão desejada para os pontos da trajetória46
4.7	Etapa de exibição da trajetória obtida47
4.8	Resultado do fatiamento do modelo de cubo da Fig. 3.8, utilizando distância entre planos de
	fatiamento adjacentes de 10mm48
4.9	Resultado do fatiamento do modelo de cilindro da Fig. 3.9, utilizando distância entre planos de
	fatiamento adjacentes de 1mm48
4.10	Resultado do fatiamento do modelo da Fig. 3.10, utilizando distância entre planos de
	fatiamento adjacentes de 1mm49
4.11	Resultado do fatiamento do modelo da Fig. 3.11, utilizando distância entre planos de
	fatiamento adjacentes de 1mm49
4.12	Vista superior do resultado de fatiamento exibido na Fig. 4.1150
4.13	Resultado do fatiamento do modelo da Figura 3.16, utilizando distância entre planos de
	fatiamento adjacentes de 1mm50
4.14	Resultado do fatiamento do modelo na Fig. 3.17, utilizando distância entre planos de
	fatiamento adjacentes de 1mm51
4.15	Resultado do fatiamento do modelo da Fig. 3.12, utilizando distância entre planos de
	fatiamento adjacentes de 5mm52
4.16	Resultado do fatiamento do modelo da Fig. 3.13, utilizando distância entre planos de
	fatiamento adjacentes de 1mm52
4.17	Detalhe das falhas detectadas nas regiões com furos no fatiamento dos modelos das Fig.
	3.12 (esquerda) e 3.13 (direita)53

	2 Desultado do fotiomento do modelo evibido no Fin	1 10
ndo o auxilio da mesa	8 Resultado do fatiamento do modelo exibido na Fig.	4.18
e de variação de direção	posicionadora, distancia entre planos de fatiamento de	
54	dos planos de 10°	
ndo o auxílio da mesa	9 Resultado do fatiamento do modelo exibido na Fig.	4.19
e de variação de direção	posicionadora, distância entre planos de fatiamento de	
55	dos planos de 1°	
ndo o auxílio da mesa	D Resultado do fatiamento do modelo exibido na Fig.	4.20
e de variação de direção	posicionadora, distância entre planos de fatiamento de	
56	dos planos de 10°	
as diferentes57	1 Simulação pelo Workspace LT no estágio de construção	4.21
modelo da Fig. 3.8, com	2 Resultado após a simulação pelo Workspace LT da co	4.22
58	trajetória obtida após fatiamento com distância entre pla	
modelo da Fig. 3.9, com	3 Resultado após a simulação pelo Workspace LT da co	4.23
58	trajetória obtida após fatiamento com distância entre pla	
nodelo da Fig. 3.10, com	4 Resultado após a simulação pelo Workspace LT da cor	4.24
59	trajetória obtida após fatiamento com distância entre pla	
nodelo da Fig. 3.11, com	5 Resultado após a simulação pelo Workspace LT da cor	4.25
59	trajetória obtida após fatiamento com distância entre pla	
nodelo da Fig. 3.12, com	6 Resultado após a simulação pelo Workspace LT da cor	4.26
60	trajetória obtida após fatiamento com distância entre pla	
nodelo da Fig. 3.13, com	7 Resultado após a simulação pelo Workspace LT da cor	4.27
60	trajetória obtida após fatiamento com distância entre pla	
nodelo da Fig. 3.16, com	8 Resultado após a simulação pelo Workspace LT da cor	4.28
61	trajetória obtida após fatiamento com distância entre pla	
62	9 Peça resultante do ensaio auxiliar de deposição	4.29
62	 Peça resultante do ensaio auxiliar de deposição, após lir 	4.30
63	1 Imagens da peça resultante do ensaio principal de depo	4.31
64	2 Imagem do ensaio principal obtida por câmera termográf	4.32
comparação com modelo	3 Identificação dos lados da peca depositada no ensaio a	4.33
65	3D	
para comparação com	4 Identificação dos lados da peca depositada no ensa	4.34
	modelo 3D	

LISTA DE TABELAS

2.1	Modos de operação da fonte de soldagem pela interface ROB5000 [38]20
3.1	Nome e função dos arquivos ".scm" desenvolvidos no trabalho
3.2	Dados para configuração do TCP no IRB 200040
3.3	Variáveis iniciais do processo de soldagem utilizadas nos ensaios de deposição42
3.4	Variáveis controláveis iniciais e finais do processo de soldagem utilizadas nos ensaios de
	deposição64
3.5	Medições de espessura de camada e distância entre camadas na peça resultante do ensaio
	auxiliar65
3.6	Medições de espessura de camada, distância entre camadas, e altura final para a peça
	resultante do ensaio principal67

LISTA DE SÍMBOLOS

Siglas

3D	três dimensões ou tri-dimensional
ABB	ASEA Brown Boveri
AIDE	ACIS Interface Driver Extension
API	Application Programming Interface
CAD	Computer-aided Design
GMAW	Gas Metal Arc Welding
GRACO	Grupo de Automação e Controle
LISP	LISt Processing
RP	Rapid Prototyping
RPY	Roll-Pitch-Yaw
SAT	Standard ACIS Text
SMD	Shaped Metal Deposition
TCP	Tool Center Point
TIG	Tungsten Inert Gas

1 INTRODUÇÃO

1.1 MOTIVAÇÃO

O desenvolvimento tecnológico afeta diretamente os processos de produção existentes, e a maximização da eficiência destes é frequentemente foco de estudos pela comunidade acadêmica [1]. Em um mundo onde empresas precisam constantemente evoluir seus métodos de manufatura, seja minimizando custos e tempo de desenvolvimento ou aumentado a diversidade, técnicas de prototipagem rápida tornaram-se opção satisfatória para vários setores da indústria [2].

Prototipagem rápida pode ser definida como a construção automática de objetos físicos por deposição de material a partir de um modelo computacional, e trata-se de um método eficiente no desenvolvimento de novos produtos ou na construção de objetos altamente customizados [3].

Em particular, técnicas capazes de construir produtos metálicos são de grande valia, dada a óbvia importância deste tipo de material para a indústria em geral. De tais técnicas, a produção de sólidos por deposição de metal em camadas sucessivas utilizando um robô soldador tem sido alvo frequente de pesquisas no meio acadêmico [4,5,6,7,8], visto o seu potencial e os muitos desafios envolvidos. Tal processo é também conhecido como soldagem 3D ou *Shaped Metal Deposition* (*SMD*).

O desenvolvimento de uma célula de manufatura por *SMD* envolve diversos obstáculos a serem superados. O planejamento do processo de produção da peça deve considerar as limitações da deposição de solda. As características mecânicas e geométricas do sólido depositado dependem de fatores como o resfriamento do sólido depositado, o fatiamento adequado do modelo computacional da peça, a velocidade de deposição da solda, etc [4,9].

Na Universidade de Brasília, o GRACO (Grupo de Automação e Controle) tem realizado pesquisas envolvendo este processo. Trabalhos incluem o desenvolvimento de um software para fatiamento do modelo computacional da peça [10], o desenvolvimento de um sistema de monitoramento e controle do arrefecimento e movimentação de uma bancada experimental para soldagem 3D [9], entre outros.

Tais pesquisas se justificam visto o potencial não-explorado deste processo, que pode ir além da fabricação de peças. Futuramente, *SMD* poderá ser aplicado no reparo de peças danificadas ou que tenham sido construídas erroneamente, ou acrescentando material a uma unidade já construída para que ela possa ser reutilizada de outra forma [4].

1.2 OBJETIVO

O objetivo deste trabalho é continuar [10] o desenvolvimento de um software capaz de, a partir do modelo computacional 3D de um sólido, gerar uma trajetória de pontos para guiar os movimentos do robô *IRB2000* de forma que este consiga reproduzir a geometria do sólido pela deposição de cordões de solda em camada.

Para tanto, deve-se definir a estratégia de fatiamento do corpo sólido, de modo a se produzir uma sequência de pontos pelos quais o robô deverá passar para reconstruir fisicamente o corpo, ao depositar o metal de solda. A tais pontos estarão atribuídos parâmetros de orientação e velocidade, para permitir a orientação do efetuador do robô e de uma mesa posicionadora que possa vir a ser utilizada.

A construção do sólido dependerá das estratégias para planejamento do processo de construção, considerando a possibilidade de serem realizadas operações de usinagem intermediárias para abertura de furos na superfície da peça, ou mudanças no eixo de crescimento caso a peça possua protuberâncias.

O trabalho é concluído ao visualizar a trajetória percorrida pelo efetuador do robô, usando o software *Workspace LT®*, e ao enviar os pontos para programação off-line do robô usando o software *Sport*^{S3}®, seguido pela realização de ensaios de deposição com o programa gerado e por uma análise comparativa entre o sólido obtido e o modelo 3D.

1.3 ESTRUTURA DO DOCUMENTO

Este documento foi organizado nos seguintes capítulos:

- Introdução (1): Termina nesta página. Consiste da motivação para o trabalho, os objetivos a serem atingidos por este, e a estrutura de apresentação deste documento.
- Fundamentação Teórica (2): Contém uma revisão bibliográfica dos assuntos necessários para compreensão dos métodos, processos e materiais (equipamentos e softwares) envolvidos neste trabalho.
- Desenvolvimento (3): Detalha a metodologia utilizada para atingir os objetivos estabelecidos na introdução.
- Resultados (4): Apresentação e análise dos resultados obtidos.
- Conclusões (5): Finaliza o trabalho com uma discussão sucinta dos resultados, seguida de considerações para possíveis trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 SOLDAGEM

A soldagem encontra aplicação extensa em quase todos os ramos da indústria e da construção mecânica e naval, além da engenharia civil. Trata-se do processo de juntar peças metálicas, permitindo contato físico, e aquecer as superfícies de contato de modo a levá-las a um estado de fusão ou de plasticidade [11].

Embora sejam usualmente utilizados como o método de união descrito acima, muitos processos de soldagem ou variações destes são usados para a deposição de material sobre uma superfície, com a intenção de recuperar peças desgastadas ou formar um revestimento com características especiais [12].

Os processos de soldagem são usualmente agrupados baseando-se no método dominante para produzir a solda: pressão ou fusão.

- Processos por fusão: A área da solda é aquecida por uma fonte concentrada de calor, causando a fusão e consequente união dos metais no local da solda.
- Processos por pressão (ou deformação): As peças são simultaneamente aquecidas (apenas até um estado plástico adiantado) e forçadas uma contra a outra (pela aplicação de força externa).

2.1.1 Processo de soldagem GMAW

GMAW (*Gas Metal Arc Welding*) é um processo de soldagem por fusão utilizando um arco elétrico entre um eletrodo metálico e a poça de fusão, diferenciando-se de outros processos similares por aplicar na poça uma proteção gasosa contra contaminação pelo ar externo, além de utilizar um eletrodo contínuo consumível [13]. O esquema do processo pode ser visualizado na Fig. 2.1.

Este processo de soldagem tem sido o mais dominante em várias indústrias (como a automotiva, a de fabricação de aço, construção de navios, pontes, tubulações, e assim por diante) devido à sua alta produtividade e o baixo custo dos equipamentos necessários [14]. Vantagens como altas taxas de deposição, ausência de escória pesada, velocidades superiores de soldagem, e a capacidade de depositar longos cordões de solda sem paradas são algumas das características que tornaram o processo *GMAW* o mais utilizado na soldagem automatizada [13].



Figura 2.1: Esquema do processo de soldagem GMAW [10].

A proteção do arco e da poça de fusão é obtida pela aplicação de gás (ou mistura de gases) cujo tipo utilizado define a sub-classificação do processo. Este será chamado *MIG* (*Metal Inert Gas*) se o gás de proteção for inerte (Ar/He), ou *MAG* (*Metal Active Gas*), caso o gás seja ativo (CO₂ ou misturas de Ar/O₂/CO₂). Gases inertes puros são em geral usados na soldagem de metais e ligas não ferrosas, enquanto que misturas de gases inertes com pequenas quantidades de gases ativos costumam ser usados com aços ligados. Já misturas mais ricas em gases ativos ou CO₂ puro são usados na soldagem de aços carbono [12].

A Figura 2.2 exibe o equipamento básico para a soldagem *GMAW*: fonte de energia, tocha de soldagem, fonte de gás e alimentador do arame (que é o eletrodo consumível do processo). A fonte de energia tem normalmente uma saída de tensão elétrica constante e regulável, sendo usada em conjunto com um alimentador de arame de velocidade também regulável [12]. A auto-regulagem do comprimento do arco elétrico pode ser obtida pela operação em conjunto do sistema alimentador de arame/fonte de energia [13].



Figura 2.2: Equipamento para soldagem GMAW [12].

As variáveis do processo de soldagem *GMAW* são definidas pelo processo ao qual este será aplicado. Previsões quanto à qualidade e características gerais do cordão de solda podem ser traçadas a partir da definição destas variáveis. Estas podem ser listadas como: tipo de eletrodo; gás de proteção utilizado; tensão elétrica da fonte; corrente elétrica de soldagem; velocidade de avanço da tocha, taxa de resfriamento da solda depositada, entre outros. Em processos automatizados de soldagem *GMAW*, o controle da tensão, corrente e posicionamento da tocha é essencial para a execução de um bom cordão de solda [9].

2.1.2 Robotização de processos de soldagem

Os benefícios de precisão e repetibilidade de robôs soldadores fazem deles uma alternativa atrativa para a automação de processos de soldagem em diversas áreas da indústria, particularmente na indústria automotiva. O uso de robôs pode ajudar a reduzir o tempo de ciclo na produção, sendo por isso preferidos para uso na produção em massa [15].

Soldagem tem sido umas das aplicações mais comuns para robôs industriais desde a sua introdução no final da década de 1970. A maioria das instalações nessa época utilizavam soldagem por resistência¹, e por volta da década de 1980 era comum encontrar em uma planta automotiva centenas de robôs soldadores [16].

Nas últimas décadas, a maior parte dos avanços tecnológicos nessa área consistiram basicamente de ampliar a robustez e performance dos subsistemas envolvidos, como por exemplo a evolução em controladores, manipuladores, e no sensoriamento das variáveis do processo [17].

Tendências futuras para a soldagem robotizada focam no aumento da flexibilidade do sistema. Simulações e programação on-line devem ganhar mais espaço, visto que a operação de definição da trajetória do efetuador² do robô (geralmente realizada por programação off-line³) é vista como o "gargalo" do processo [16, 18].

O método de prototipagem rápida por deposição de metal em camadas sucessivas, que é detalhado mais adiante (seção 2.2.1), exige um processo de soldagem obrigatoriamente robotizado. Isso se dá pela precisão, estabilidade e velocidade necessárias para o processo, além das longas exposições à radiação luminosa emitida pela soldagem.

¹ Grupo de processos de soldagem nos quais o calor necessário à formação da junta soldada é obtido pela resistência à passagem de corrente elétrica através das peças sendo soldadas [12].

² Dispositivo acoplado ao final do último elo do braço robótico, como por exemplo uma tocha de soldagem, ou uma garra [19]. Mais detalhes na seção 2.5.

³ Ambiente de programação robótica que foi suficientemente estendido, normalmente por gráficos computacionais, para permitir o desenvolvimento de programas para o robô sem precisar acessar o robô em si [19].

2.2 PROTOTIPAGEM RÁPIDA

Prototipagem rápida (ou *RP*, do inglês *Rapid Prototyping*) é um processo de fabricação que produz a peça desejada, camada por camada, a partir de modelos computacionais 3D controlados por computador. Prototipagem rápida vem se desenvolvendo desde a década de 1980, e pesquisa neste campo continua crescendo, visto seu potencial na redução de tempo e custos no desenvolvimento ou fabricação de produtos [3].

Existem vários métodos para a fabricação por prototipagem rápida, e a seleção adequada do método depende do material da peça desejada, entre outros fatores. A seguir, estão listados os métodos mais amplamente difundidos na indústria em geral [9]:

- Estereolitografia: Consiste na utilização de um laser para produzir peças a partir de polímeros fotossensíveis;
- Sinterização Seletiva a Laser: Consiste na utilização de laser para fundir finas camadas de pó para obtenção da geometria das seções transversais da peça. Apresenta menor custo e menor precisão quando comparada à estereolitografia;
- Modelagem por Deposição Fundida: Consiste na utilização de um fio de plástico ou cera que é fundido gradativamente de maneira similar ao processo de soldagem, formando a geometria das camadas de cada seção transversal da peça.
- Manufatura de Objetos Laminados: Consiste na deposição de camadas do material, seguida da passagem de um rolo. Um laser corta o contorno da camada da seção transversal, moldando o formato da peça. Encontra-se em desuso, por ser o processo listado aqui com maior perda de material [2].

Embora existam diferentes técnicas para a construção física do objeto, vários destes processos dividem entre si a forma como o modelo 3D computacional da peça é dividido em camadas (ou "fatiado"). De fato, a técnica de fatiamento utilizada é de importância vital em processos de prototipagem rápida. Isso ocorre devido ao desafio que é processar a informação geométrica do modelo e dividi-lo rapidamente em camadas que possam ser reproduzidas pelo maquinário. Para um mesmo modelo, diferentes métodos de fatiamento geram resultados variados de qualidade e produtividade [3].

Não importando o processo utilizado, prototipagem rápida gera um objeto que não é uma reprodução perfeita do modelo computacional. O principal fator que afeta todas as variadas técnicas é o *efeito degrau*, que ocorre devido ao fato de que uma dada camada é vertical, o que causa a inserção de uma característica rugosa a um modelo originalmente liso [10]. A Figura 2.3 exibe um exemplo deliberadamente exagerado para fácil visualização do efeito degrau.



Figura 2.3: Representação de um modelo esférico e o possível resultado de camadas excessivamente grandes na fabricação por prototipagem rápida [20].

O tamanho do efeito degrau depende principalmente de dois fatores [10]:

- Altura da camada: A Figura 2.3 facilita a compreensão de que o efeito degrau é fortemente ligado à altura da camada depositada, sendo que quanto menor a altura da camada, mais suave será a superfície resultante. Entretanto, deve-se considerar o aumento no tempo de execução do processo, dado o inevitável aumento na quantidade de camadas a serem depositadas.
- Orientação do fatiamento: A orientação de uma superfície particular ao plano das camadas afeta o efeito degrau, pois em geral, lados paralelos e perpendiculares ao plano das camadas sofrem menor efeito, mas apenas para superfícies planas. Degraus pronunciados tendem a surgir em superfícies curvas.

2.2.1 Deposição de metal em camadas sucessivas

A maioria dos processos existentes de prototipagem rápida utiliza resinas para a construção do objeto desejado. Embora elas sejam apropriadas para muitos casos na indústria em geral, o seu uso no desenvolvimento de peças metálicas é compreensivelmente limitado, visto as diferenças entre estes materiais (por exemplo: diferenças mecânicas e termodinâmicas). Entre outras técnicas, a prototipagem rápida por deposição de metal em camadas sucessivas surgiu na tentativa de contornar este problema [10].

Este método, que também é conhecido como soldagem 3D ou ainda *SMD* (do inglês *Shaped Metal Deposition*), consiste em depositar material derretido ao longo de uma trajetória utilizando equipamento de soldagem guiado por um braço robótico [6]. A Figura 2.4 exibe uma célula de soldagem 3D que utiliza o processo TIG.

Para a execução adequada de uma deposição de metal em camadas sucessivas, variáveis do processo GMAW foram combinadas com variáveis da prototipagem rápida, resultando em quatro parâmetros que foram identificados como sendo fundamentais para o controle adequado do procedimento [5]: a fonte de calor provida ao processo (corrente do arco elétrico), a taxa de massa do material injetado (taxa de alimentação do arame), velocidade da tocha de soldagem (velocidade do efetuador) e a espessura definida para cada camada

depositada (indicada pela altura entre duas camadas depositadas sequencialmente). Pesquisas revelam que esses parâmetros influenciam as dimensões, espessura e comprimento da peça final [5]. A Figura 2.5 exibe uma representação esquemática do processo *SMD*.



Figura 2.4: Célula de soldagem 3D com uso do processo TIG [4].



Figura 2.5: Representação esquemática do processo SMD [5].

Assim, estudos mais aprofundados sobre os parâmetros de soldagem GMAW e sobre as formas de transmissão de calor da tocha para a peças podem resultar em melhoria das qualidades geométricas e funcionais da peça, minimizando as etapas de acabamento por meios convencionais de usinagem [9].

Deve-se atentar ao fato de que a técnica nomeada Shaped Metal Deposition é patenteada pela Rolls-Royce® como sendo um processo que utiliza TIG (Tungsten Inert Gas) para a

deposição de metal [21]. Entretanto, como a expressão é consideravelmente genérica, ela é comumente utilizada em artigos científicos como a deposição de metal em camadas sucessivas utilizando qualquer processo de soldagem [4,8]. Assim sendo, ao longo deste trabalho a expressão *SMD* será utilizada apenas como deposição de metal em camadas sucessivas pelo processo *GMAW*.

2.3 PLATAFORMA ACIS®

O modelador 3D *ACIS®* é uma plataforma 3D orientada a objetos¹ para modelagem de sólidos ou geometrias, desenvolvido pela *Spatial Corp®*, tendo sido desenvolvido para uso como a fundação geométrica dentro de uma aplicação de modelagem 3D [22].

ACIS é escrita em linguagem de programação C++² e consiste de um conjunto de classes (incluindo métodos) e funções em C++. Um desenvolvedor utiliza essas classes para criar a sua aplicação 3D [24]. A Figura 2.6 exibe o diagrama de dependências entre as bibliotecas fundamentais (*"core libraries"*) do *ACIS*. Estas bibliotecas fornecem funcionalidade de modelagem e não são simplesmente componentes de suporte ou interface (como é o caso de *Scheme*, descrito mais adiante).

Entretanto, o usuário da plataforma ACIS precisa compreender apenas superficialmente este diagrama, pois a interface entre a aplicação sendo desenvolvida e o *ACIS* é realizada através de funções *API* (*Application Programming Interface*), que gerenciam chamadas de construção, alteração ou retorno de dados para o desenvolvedor (usuário). Uma função API isola o usuário das complexidades da plataforma geométrica, combinando funcionalidades de modelagem com suporte para a aplicação (como verificação de erros).

Mais importante, para o uso de *ACIS*, é compreender a organização dos objetos que representam os vários componentes geométricos de uma dada estrutura. A figura 2.7 exibe as derivações das classes de objeto de um modelo em *ACIS*.

Uma *entidade* é o objeto mais básico em *ACIS*, sendo implementado pela classe *ENTITY*. Todas as entidades têm em comum um grupo de funcionalidades, como por exemplo a habilidade de restaurar a si mesmos para ou de um arquivo, ou ainda a habilidade de se copiar ou de buscar falhas em seus dados. Todos os outros objetos de modelos geométricos e de alto nível são derivados da classe *ENTITY*.

ACIS realiza a união de diferentes superfícies, sólidos e fios, permitindo que estas representações coexistam na estrutura de dados, armazenando-os em arquivos ".SAT" [24].

¹ Paradigma de programação que utiliza objetos (consistindo de campos de dados e funções junto com suas interações) para desenvolver aplicativos [9].

² Linguagem de programação que estende a linguagem C para uso orientado a objetos [23].



Figura 2.6: Diagrama de dependências entre as bibliotecas fundamentais do AC/S [24].



Figura 2.7: Diagrama de derivações das classes de objeto para um modelo em ACIS [24].

2.3.1 Interface para Scheme

Scheme, que será descrita mais detalhadamente na seção a seguir, é uma linguagem interpretativa de domínio público derivada de *LISP*¹ que provê uma capacidade de prototipagem rápida e fácil. Diferentemente da maioria das linguagens interpretativas, *Scheme* roda com muita eficiência, e muitos desenvolvedores que a usam encontram pouca necessidade de transcrever seu código para outra linguagem compilada [24].

Por ser interpretativa, não existem fases de compilação ou ligação para procedimentos *Scheme*, e ela pode ser usada imediatamente para treinamento. *Scheme* é particularmente útil para controlar tarefas, como por exemplo processamento de comando de interface para usuário. Ela é extensível e portável para todas as plataformas suportadas por *ACIS*.

Em ACIS, a interface Scheme é uma coleção de funções que permite à aplicativos baseados em Scheme acessar as funcionalidades do ACIS. Aplicativos interagem com ACIS usando o interpretador do ACIS, que processa comandos em Scheme. Esses comandos podem ser os usuais da linguagem, ou extensões do ACIS, ou ainda comandos específicos criados pelo desenvolvedor.

A Figura 2.8 ilustra como um aplicativo em *Scheme* é construído sobre o *ACIS*, interagindo com o componente de suporte para *Scheme* através do interpretador *Scheme*, que então interage com o resto do *ACIS* utilizando chamadas em C++.



Figura 2.8: Interface de um aplicativo em Scheme para ACIS [24].

¹ Família de linguagens de programação modelada em cálculo lambda cuja principais características são a recursividade, expressões condicionais e sintaxe com uso de parênteses em notação polonesa. [25].

2.4 LINGUAGEM DE PROGRAMAÇÃO SCHEME

O primeiro grande dialeto de *LISP* a distinguir procedimentos de expressões lambda¹ e símbolos, usando um único ambiente léxico para todas as variáveis, *Scheme* foi uma das primeiras linguagens de programação a incorporar procedimentos de primeira classe como no cálculo lambda, fornecendo utilidade a regras estáticas e estruturas de bloco em uma linguagem dinâmica [26].

Inventada por Guy Lewis Steele Jr. e Gerald Jay Sussman em 1978, *Scheme* confia inteiramente em chamadas de procedimentos para expressar iterações, e programas são normalmente salvos em arquivos (.SCM) ou inseridos interativamente em sistemas rodando *Scheme* [26].

2.4.1 Gerando programas em Scheme com ACIS

O Scheme ACIS Interface Driver Extension (Scheme AIDE) é um aplicativo de demonstração baseado em ACIS, e pode ser utilizado por desenvolvedores de muitas maneiras [24]:

- Auxilia no exercício de funcionalidades do ACIS sem a necessidade de escrever ou compilar um aplicativo em C++, o que ajuda desenvolvedores a aprender e prototipar novas funcionalidades;
- Como o código fonte em C++ é fornecido para extensões do Scheme usadas pelo Scheme AIDE, pode-se estudar como uma funcionalidade específica do ACIS pode ser implementada em um aplicativo desejado;
- Pode ser utilizado como exemplo de um aplicativo ACIS baseado em Scheme, ou o ponto de partida para criar um novo aplicativo;

O *Scheme AIDE* é um programa com linha de comando que aceita comandos em Scheme, possuindo uma janela para entrada dos comandos e uma janela de observação em separado. Os comandos podem ser tanto os nativos da linguagem Scheme como extensões criadas pelo *ACIS* [24].

Utilizar o *Scheme AIDE* para executar tarefas pré-definidas pode ser feito editando o arquivo "acisinit.scm". Este arquivo define quais os passos a serem executados pelo *Scheme AIDE*, e arquivos com código em *Scheme* (.scm) podem ser inseridos em "acisinit.scm" para executar o código automaticamente.

¹ Operador usado para vincular variáveis a valores, sendo então avaliados dentro de uma expressão. Este operador é usado para criar funções, recebendo argumentos e as expressões que a função criada avalia. [25].

2.5 ROBÔS MANIPULADORES

Os mais importantes dos robôs industriais [19], robôs manipuladores são frequentemente usados para manipulação de materiais, soldagem, pintura, entre outros. Alguns exemplos dos principais fabricantes são: *ABB, KUKA, STAUBLI, PUMA*, etc.

O manipulador mecânico robótico é essencialmente composto das seguintes partes [27]:

- Braço mecânico: Constituído por juntas e elos. Elos são usualmente blocos alongados rígidos conectados uns aos outros por juntas. Existem diversas variações de juntas, elos, e formas de combiná-los.
- Atuadores: Dispositivos que impõem movimento a qualquer parte mecânica móvel do robô. Trata-se usualmente de motores elétricos, hidráulicos ou pneumáticos, podendo ter associados elementos adicionais de transmissão mecânica.
- Sensores: Elementos destinados à medição do estado interno do manipulador (deslocamento dos atuadores), podendo estender-se ao ambiente exterior. Os mais comuns em manipuladores são os codificadores (*encoders*) e os sensores de fim-decurso.
- Controlador: Unidade capaz de gerar informação de ativação de um ou mais atuadores com base em algum algoritmo de controle. Este algoritmo pode levar em consideração comandos recebidos, o estado atual dos atuadores ou o próprio ambiente (com informações dos sensores).
- Ferramenta ou garra (*end-effector*): Componente ligado ao último elo do manipulador que insere funções adicionais de uso, como agarrar um objeto ou utilizar alguma ferramenta. Ferramentas usuais incluem tocha de soldagem, jato ou spray de água ou tinta, furadeiras, fresa, etc.

No estudo de robôs manipuladores, a localização de objetos no espaço tri-dimensional é de vital importância. Esses objetos são os elos do manipulador, as partes e ferramentas utilizadas e possivelmente elementos do ambiente. Em robótica, a descrição desses elementos é usualmente realizada com apenas dois atributos: posição e orientação. A descrição destes atributos para um dado corpo no espaço é feita anexando a ele um sistema de coordenadas próprio e rígido, e determinando a relação deste sistema de coordenadas a um sistema de referência [27].

Um sistema de coordenadas que exige especial atenção é o da ferramenta, visto que ferramentas podem ter diversos formatos e orientações. Este sistema é obtido pela calibração do *TCP (Tool Center Point)*, que diferentemente do que o nome pode implicar, trata-se de um conjunto de dados a partir do qual o robô pode inferir a posição e orientação do sistema de coordenadas de uma dada ferramenta com relação ao último elo do robô.

2.5.1 Robô ABB® IRB 2000

O laboratório do GRACO dispõe de uma unidade do robô *IRB 2000* (Fig. 2.9), da *ABB*. Trata-se de um robô manipulador articulado com seis graus de liberdade¹ criado, a princípio, para uso em soldagem a arco elétrico e colagem/selamento (mas podendo também ser utilizado em operações de montagem, corte por jato de água ou laser, manuseio de material, entre outros), e pode carregar objetos de até 10kg dentro de seu volume de trabalho [29].



Figura 2.9: Desenho do *IRB 2000*, ilustrando as diferentes partes, motores e eixos de rotação [28].

O sistema de controle do manipulador se encontra no gabinete de controle (Fig. 2.10), onde se encontram o painel de controle, a unidade de programação (*teach pendant*), a unidade de armazenamento, conexões para comunicação, além de outros acessórios opcionais. É neste gabinete que se encontra o controlador S3, que utiliza a linguagem de programação *ARLA* para programação simples e ágil do robô. O controlador possui entradas e saídas digitais e analógicas, permitindo a interface das operações do robô com equipamentos externos. Outra capacidade do controlador que merece destaque é a de se comunicar com computadores, permitindo controle externo ou envio de dados.

¹ Número de variáveis de posição independentes que precisam ser especificadas para localizar todas as partes de um mecanismo. É o número total de movimentos independentes que um dispositivo pode efetuar [19,27].



Figura 2.10: Gabinete de controle do IRB 2000 [29]

A performance mecânica do robô é excelente para a maioria das aplicações: resolução de posionamento de 0,125mm e repetibilidade inferior a $\pm 0,1$ mm, com atuadores podendo girar com velocidades máximas de 115º/segundo a 300º/segundo, dependendo do eixo. Seguindo a notação da Fig. 2.9, a capacidade de rotação nos eixos varia de $\pm 60^{\circ}$ (terceiro eixo - A) a até $\pm 300^{\circ}$ (sexto eixo - P).

Quanto à performance computacional, existem 200 registradores de posição (n° 0 a 199) e 120 registradores numéricos (n° 0 a 119) de 16 bits (podendo representar números de 0 a ± 32727), que podem ser utilizados para tarefas como a realização de operações matemáticas ou armazenamento de dados, ou ainda enviados para equipamentos externos. O controlador permite a gravação de 30 configurações diferentes de *TCP* (n° 0 a 29), sendo o *TCP* de número 0 reservado para o caso de ausência de qualquer ferramenta após o último elo do robô [29].

Ainda na performance computacional, o usuário pode inserir programas numerados de 1 a 9999, limitado apenas pela capacidade de memória para armazenamento de programas, que é de 64 *kbytes*. Este valor pode ser um forte limitante na sua utilização para operações que exigem programas longos, em que a trajetória não é repetida, sendo exatamente o caso da trajetória para *SMD*.

O sistema ainda permite a criação de cinco sistemas de coordenadas (*FRAME* 1 a 5) com posição ou orientação diferentes do sistema de referência (*FRAME* 0), que se encontra na base do robô.

2.6 REPRESENTAÇÕES DE ORIENTAÇÃO NO ESPAÇO 3D

Como visto na seção 2.5, em robótica utilizam-se sistemas de coordenadas para caracterizar objetos com posições e orientações.



Figura 2.11: Sistemas de coordenadas (ou *frames*) são fixados ao manipulador e aos objetos do ambiente, além de considerar o sistema de referência [19].

Para expressar esses atributos utilizando sistemas de coordenadas diferentes, utilizam-se matrizes de transformação para relacionar as mudanças de translação e rotação dos eixos coordenados de um sistema para outro. No caso tri-dimensional, que é a situação encontrada em robótica, utiliza-se a matriz exibida na Fig. 2.12.

Pode se observar na Fig. 2.12 a existência de uma sub-matriz de dimensões 3x3 que traduz a orientação do novo sistema de coordenadas com respeito ao sistema anterior. Essa matriz é conhecida como matriz de rotação, e indica a orientação do sistema após uma sequência de três rotações em torno de eixos distintos no espaço tri-dimensional, reproduzindo os três graus de liberdade rotacional distintos que existem no espaço [27].



Figura 2.12: Componentes da matriz de transformação 3D [27].

Matrizes de rotação podem ser utilizadas tanto como operadoras de transformação como descrições de orientação. Entretanto, embora elas sejam extremamente úteis como operadoras, elas podem se tornar de difícil compreensão ao especificar uma orientação [19]: inserir nove elementos para determinar uma orientação desejada pode confundir um usuário humano, por exemplo.

Assim, foram criadas outras formas de representar uma orientação qualquer no espaço, que facilitam a representação, mesmo podendo, em alguns casos, perder eficiência no tempo de execução dos cálculos. Algumas destas representações serão analisadas a seguir.

2.6.1 RPY

Os ângulos de rotação em torno de cada um dos três eixos coordenados do sistema cartesiano são normalmente chamados de ângulos de Euler, e o seu significado depende da combinação de rotações usadas (27).

Uma das combinações mais conhecidas é a *RPY* (*Roll-Pitch-Yaw*) e consiste de rotações sequenciais nos eixos x, y e z, nesta ordem. A Figura 2.13 ilustra a definição de *roll, pitch* e *yaw*, seguidos de suas matrizes de rotação simplificadas.



Figura 2.13: Rotações RPY e correpondência usual no punho de um manipulador [27].

Considerando que as rotações ocorrem nos eixos x (*yaw*), seguido de y (*pitch*) seguido de z (*roll*), e sabendo-se que as matrizes de rotação das rotações posteriores para um referencial fixo são pré-multiplicadas [19], obtemos a seguinte expressão matemática:

$$Rot(\phi, \theta, \psi) = RPY(\phi, \theta, \psi) = Rot(z, \psi)Rot(y, \theta)Rot(x, \phi)$$
(1)

A matriz de transformação para as rotações RPY é então obtida após o desenvolvimento das matrizes de rotação.

A convenção *RPY* é apenas uma de 12 combinações possíveis de ângulos de Euler fixos. Outras combinações comumente usadas são as rotações ZYX (rotações sequenciais nos eixos z, y e x, nessa ordem) e ZYZ (rotações sequenciais nos eixos z, y e novamente no eixo z, nessa ordem) [19].

2.6.2 Quatérnions

O quatérnion, introduzido por Hamilton em 1843 [30], é uma extensão de números complexos em quatro dimensões. O formalismo clássico define um quatérnion como:

$$\mathbf{q} = a_0 + a_1 i + a_2 j + a_3 k \tag{2}$$

onde $a_0, a_1, a_2 \in a_3 \in \Re$ e os três componentes imaginários *i*, *j* e *k* são definidos a seguir:

$$i^2 = j^2 = k^2 = -1 \tag{3}$$

possuindo então as seguintes propriedades:

$$ij = k, jk = i, ki = j, ji = -k, kj = -i, ik = -j$$
(4)

Uma representação alternativa de quatérnion é como um par (s,\vec{n}) , separando respectivamente a porção escalar da vetorial [30].

Quatérnions possuem uma regra para multiplicação associativa, mas não comutativa, podendo ser distribuída em adições. Além disso, cada vetor não-nulo possui um vetor inverso multiplicativo único. Assim tem-se que quatérnions formam uma álgebra de divisão própria [31].

Observando as equações (2), (3) e (4), fica clara a relação dos quatérnions com a análise vetorial usual, que representa elementos em bases ortonormais [32]. De fato, o quatérnion unitário ($||\mathbf{q}|| = 1$) pode ser utilizado para representar uma rotação de θ em torno de um vetor unitário \vec{n} no espaço euclidiano tri-dimensional como [30]:

$$\mathbf{q} = (\cos\left(\theta/2\right), \vec{n}\sin\left(\theta/2\right)) \tag{5}$$

de acordo com a representação por pares.

Em robótica e também em visão computacional, a natureza das ferramentas matemáticas disponíveis afeta diretamente a capacidade de realizar tarefas, tanto na teoria quanto na prática. Quatérnions ganharam destaque por ser uma poderosa ferramenta de notação vetorial, que permite operações que não costumam ser representadas diretamente por vetores, como rotações [32].

2.7 SOFTWARE WORKSPACE LT®

O software *Workspace LT*, da *WAT Solutions*, é um sofisticado sistema de simulação desenvolvido para propósitos educacionais. Nele é possível criar modelos de células de trabalho robotizadas e avaliar a sua performance, testando alcance e detecção de colisões [33].



Figura 2.14: Área de trabalho do software *Workspace LT*, onde pode-se visualizar uma célula de trabalho robotizada desenvolvida projetada pelo usuário [34].

O software combina propriedades de *CAD¹* e simulação [34]:

- Criação e modificação de objetos 3D, podendo importar dados de outros softwares similares;
- Criação de robôs e outros mecanismos, mas já possuindo uma extensa biblioteca de robôs encontrados no mercado;
- Modelagem cinemática de robôs e mecanismos, com estudos de alcance, ciclo de trabalho, otimização de trajetória e detecção de colisão, entre outros.

Uma característica importante do software é permitir a importação de programas de robôs de diversas linguagens de programação diferentes, permitindo a visualização da funcionalidade do programa em um ambiente simulado computacionalmente.

Embora, como dito anteriormente, diversas linguagens de programação de robôs possam

¹ Computer-aided design - uso de sistemas computadorizados no auxílio da criação, modificação ou análise de um projeto [35].

ser utilizadas, o *Workspace LT* utiliza como linguagem padrão o Karel 2, que é a linguagem utilizada por alguns controladores de robôs da *Fanuc*. Esta linguagem é a que se encontra melhor implementada no software, sendo a mais recomendada pelo desenvolvedor para uso extensivo [33].

2.8 SOFTWARE SPORT^{S3}®

O software *Sport*^{S3} é um pacote de software criado pela *PPM AS*® com o objetivo de aumentar a segurança e facilidade no uso dos já obsoletos sistemas robóticos S3 da *ABB*. O software oferece programação *offline* e permite o controle de múltiplos robôs [35] a partir da interface exibida na Fig. 2.15.



Figura 2.15: Interface de usuário do Sport S3 [37].

Através do software, o usuário pode alterar os valores dos diversos registradores diretamente nos robôs conectados, permitindo por exemplo a gravação de *TCP*'s ou novos sistemas de coordenadas sem a necessidade de movimentações para calibração.

Utilizando uma conexão serial, é possivel utilizar um computador para enviar ou receber programas da memória do robô, além de permitir o monitoramento em tempo real da

posição e orientação do efetuador, considerando o *TCP* e o sistema de coordenadas de refência ativados no momento.

A programação em *ARLA* é facilitada pela presença de tutoriais, janelas para auxiliar na utilização e inserção de comandos, e de um compilador que verifica a correção do código desenvolvido [36].

2.9 INTERFACE ROB5000®

Fabricada pela *Fronius*®, a interface *ROB5000* permite a comunicação entre dispositivos de soldagem da empresa com dispositivos eletrônicos externos, como por exemplo o controlador de um robô.

A interface possui entradas e saídas digitais e analógicas, com as quais o dispositivo externo pode se enviar ou receber sinais. Assim, torna-se possível o desenvolvimento de um programa na linguagem de programação do robô, capaz, por exemplo, de configurar os modos de operação da fonte e de ativar ou desativar a tocha de soldagem nos momentos em que o efetuador do robô estiver nas posições desejadas [9,38].

Neste trabalho, apenas os sinais de entrada digitais serão utilizados, o que naturalmente pode ser modificado em trabalhos futuros.

2.9.1 Entradas Digitais

Os sinais digitais de entrada da interface são vários, mas os de fundamental importância na operação de soldagem estão listados a seguir [38]:

- Source Error Reset: Sinal que deve ser acionado para continuação do funcionamento após a ocorrência de algum erro ser detectado pela fonte. Por questões de segurança, também recomenda-se ativá-lo momentaneamente (0,01s) antes de iniciar uma nova operação;
- Robot Ready/Quick Stop: Sinal enviado pelo controlador do equipamento que manipula a fonte (no caso, o robô) para avisar a fonte de que este está pronto para soldagem. Pode ser utilizado como uma parada de emergência ao se desativar o sinal, exigindo a seguir a aplicação de um pulso na entrada Source Error Reset;
- Mode bit 0-2: Seleção do modo de operação da fonte de soldagem, ocupando 3 bits, de acordo com a Tabela 2.1;
- Program/Job Number. Sinal de 8 bits para seleção do programa ou job específico a ser utilizado, dependo da configuração enviada às entradas Mode bit 0-2;

• *Welding Start*: Sinal para ativação da tocha de soldagem. Deve-se ser mantido em nível alto enquanto desejar-se a tocha em funcionamento;

	ROB	4000/	5000	
		MODE		
	"0"	"1"	"2"	
Standard program	0	0	0	
Pulsed arc program	1	0	0	
Job mode	0	1	0	
Parameter selection internal	1	1	0	
Manual	0	0	1	
CC/CV	1	0	1	
TIG	0	1	1	
CMT/special process	1	1	1	

Tabela 2.1 - Modos de operação da fonte de soldagem pela interface ROB5000 [38].

3 DESENVOLVIMENTO

3.1 ESTRATÉGIAS DE FATIAMENTO DO MODELO 3D

Como explicado anteriormente (seção 2.2), a seleção adequada da estratégia utilizada para o fatiamento do modelo computacional 3D em camadas para deposição de material é de fundamental importância para os processos de prototipagem rápida, e a técnica de deposição de metal por camadas sucessivas não é exceção.

No caso particular da técnica *SMD*, a estratégia de fatiamento deve considerar o efeito que as variáveis de processo da técnica de soldagem *GMAW* podem causar na peça final (seção 2.2.1).

Duas principais técnicas de fatiamento podem ser consideradas para uso em *SMD*: fatiamento por planos paralelos ou fatiamento em parafuso (plano espiral) [10].

O fatiamento por planos paralelos, ilustrado na Fig. 3.1, consiste em fatiar o modelo em planos perpendiculares à superfície de crescimento, e para cada plano perpendicular depositar material ao longo da intersecção entre o plano e o modelo. Repete-se o processo para todas as camadas, depositando material sobre a camada abaixo. Esta técnica de fatiamento é popular na maioria dos processos de *RP* usuais, tendo sido considerada inapropriada para uso em *SMD*. Limitações do processo *GMAW* exigem movimento contínuo do robô soldador, e a parada no ponto de subida para a próxima camada causaria um acúmulo de calor e solda excessivos nessa região [10].



Figura 3.1: Exemplo de modelo 3D fatiado por planos paralelos: (a) modelo original; (b) modelo pós-fatiamento; (c) uma única camada do modelo fatiado [3].

Já o fatiamento em plano espiral, que foi considerado solução para os problemas de descontinuidade no movimento do robô soldador, consiste em utilizar um conjunto de semiretas paralelas ao plano perpendicular ao eixo de crescimento do modelo. Após incrementar a altura e o ângulo de cada semi-reta, obtém-se o ponto de intersecção entre tais semi-retas e o modelo. O conjunto dos pontos, adquiridos na ordem de incremento na altura das retas (da base até o topo do modelo), consiste aproximadamente na trajetória que a tocha do robô soldador deveria percorrer para criar a peça [10].



A Figura 3.2 ilustra o resultado obtido como sendo um plano espiral.

Figura 3.2: Vista do conjunto espiral de semi-retas [10].

Embora tenha sido considerada superior ao fatiamento em planos paralelos, a técnica de fatiamento em plano espiral também possui desvantagens.

Ao utilizar incrementos angulares constantes para as semi-retas, peças com protuberâncias longas podem perder precisão. Esta falha está ilustrada na Fig. 3.3: à medida que a distância entre os pontos de intersecção e a origem de suas respectivas semi-reta aumenta, a distância entre os pontos obtidos também cresce, perdendo-se precisão na trajetória. No exemplo da figura em questão, a tocha de soldagem do robô percorreria o caminho em linha reta entre os pontos S3 e S4, e em seguida entre os pontos S4 e S5, o que causaria um erro grosseiro na peça resultante [10].

Uma solução possível para este problema reside em executar incrementos angulares que não sejam fixos, sendo inversamente proporcionais à distância entre a origem da semi-reta e o último ponto de intersecção. Dessa forma, após a ocorrência de um ponto de intersecção mais distante, o incremento angular para a próxima reta seria diminuído.


Figura 3.3: Corte superior de um fatiamento de um modelo com protuberância longa [10].

Outro problema que pode ocorrer no fatiamento em espiral é a ocorrência de crescimento repentino na direção das semi-retas, ilustrado na Fig. 3.4. Nessa situação, apenas um ponto de um reta seria acrescentado à trajetória, normalmente o ponto intermediário da reta. Na figura em questão, a tocha de soldagem se moveria do ponto S1 para o S2 e em seguida para o S3, novamente introduzindo um erro grosseiro no resultado final.



Figura 3.4: Corte superior de um fatiamento de um modelo com protuberância crescendo ao longo da direção de uma semi-reta.

3.1.1 Seleção final da estratégia

Após uma análise das estratégias de fatiamento já discutidas e considerando as funcionalidades que a plataforma *ACIS* fornece, decidiu-se por elaborar um procedimento de fatiamento próprio para uso neste trabalho, não adotando exatamente as estratégias apresentadas anteriormente.

Essa decisão foi tomada por influência das ferramentas disponibilizadas pela plataforma *ACIS*, cujas características auxiliam o desenvolvedor na manipulação da intersecção entre planos e o modelo computacional 3D.

A estratégia de fatiamento utilizada consiste em efetuar fatiamentos em planos paralelos, mas mantendo uma movimentação contínua do braço robótico. Ao invés de percorrer o perímetro da intersecção do modelo com cada plano, uma trajetória contínua é interpolada baseada na intersecção entre o modelo e planos de fatiamento consecutivos.

O procedimento para a obtenção dos pontos da trajetória a ser percorrida pelo efetuador do robô soldador é descrito a seguir:

- O procedimento começa na base do modelo 3D, que precisa necessariamente estar sobre o plano xy do sistema de coordenadas. Um plano para fatiamento é criado na base e outro criado paralelo ao primeiro plano, com a distância entre os dois planos paralelos tendo sido previamente definida pelo usuário do programa.
- Pontos são interpolados, a partir do plano inferior até o superior, baseados na intersecção dos planos. Esses pontos formam uma trajetória que conecta o plano inferior ao superior. A interpolação termina quando a trajetória atinge o plano superior.
- 3. A seguir, o plano superior se torna o novo plano inferior, e um novo fatiamento por plano é executado. Este novo plano estará deslocado com relação ao plano anterior ao longo do eixo de crescimento, com mesma orientação do eixo e transladado pela distância definida como "distância de fatiamento dos planos", definida previamente pelo usuário.
- 4. Com novos planos superior e inferior definidos, a etapa 2 se repete.

Este ciclo se repetirá até o momento em que o resultado da intersecção entre o plano de fatiamento e o modelo for nulo.

A Figura 3.5 ilustra um exemplo de um ciclo de fatiamentos, com distância entre planos deliberadamente exagerada para facilitar a visualização. As linhas laranjas indicam o resultado da intersecção entre o modelo e os planos de fatiamento, com exceção das linhas centrais, que foram geradas interpolando as linhas externas e internas. A linha azul corresponde à trajetória obtida e que deverá ser enviada ao robô, e foi gerada interpolando pontos entre as linhas laranjas centrais inferior e superior.



Figura 3.5: Ilustração de exemplo de um ciclo de fatiamentos. A distância entre planos de fatiamento está deliberadamente exagerada para facilitar a visualização.

3.1.2 Determinação do eixo de crescimento do modelo

O fatiamento adequado de modelos que possuam um eixo de crescimento que não seja paralelo ao eixo vertical depende da utilização de planos de fatiamento com orientação adequada: a direção normal dos planos deve ser a mesma do eixo de crescimento no local de interseção. Além disso, o plano deve passar pelo ponto definido pela distância entre planos de fatiamento, a partir do centro geométrico da seção de fatiamento anterior, ao longo do eixo de crescimento dessa seção anterior.

A interseção do plano de fatiamento com o modelo resulta em uma seção do modelo naquele plano. Para determinação da direção do eixo de crescimento, decidiu-se por utilizar os centros de área dessas seções: o vetor que conecta os centros de área de duas seções consecutivas é considerado a direção do eixo de crescimento naquela região. A Figura 3.6 exibe um exemplo, com distância entre planos também deliberadamente exagerada: o eixo de crescimento na região entre os planos de fatiamento (linhas azuis) possui a direção da reta que conecta os centros de área das seções de fatiamento (linha vermelha).

Deve-se salientar que na Fig. 3.6 a seção superior não será utilizada: um novo fatiamento será executado, agora com a orientação paralela ao eixo de crescimento detectado.



Figura 3.6: Determinação do eixo de crescimento na região entre as seções de fatiamento.

3.1.3 Obtenção das orientações para mesa posicionadora

Para a geração de uma trajetória adequada e que respeite as características do processo de soldagem, o software desenvolvido pode considerar a disponibilidade de uma mesa posicionadora, auxiliando na criação da peça quando esta possuir eixo de crescimento não alinhado com a vertical. Essa mesa seria responsável por alinhar o eixo de crescimento com a vertical, permitindo a deposição de solda sem alterar a orientação da tocha no efetuador do robô.

Considerando o eixo z do sistema de coordenadas da mesa posicionadora inicialmente idêntico ao eixo z do sistema de referência (vertical), o ajuste na orientação consiste em orientar o eixo z da mesa de forma a coincidir com o eixo de crescimento, mas espelhado no eixo vertical do sistema de referência; assim, a soma vetorial das direções resulta no próprio eixo vertical, como desejado.

Ao final do processo de fatiamento, tem-se uma lista das orientações da mesa para cada plano de fatiamento efetuado, que é trasformada da notação vetorial em bases ortonormais para representação em quatérnions. A lista é então gravada no arquivo-texto "orientacoes_mesa.txt".

A decisão de utilizar a representação em quatérnions para as orientações da mesa posicionadora foi feita beseada em alguns fatores:

- A transformação das orientações de vetores em base ortonomal para quatérnions, que pode ser efetuada pela Eq.5 (seção 2.6.2), é extremamente facilitada por ferramentas disponíveis no ACIS, sendo a transformação mais simples das representações discutidas na seção 2.6 e suas sub-seções.
- Como discutido na seção 2.6, a utilização de matrizes de rotação significaria gerar para cada ponto da trajetória nove elementos, que corresponderiam aos termos da matriz 3x3;
- Uma das três rotações da representação RPY jamais seria utilizada. Isso devido à natureza do processo em questão: não é necessário alterar a orientação da tocha de soldagem, e quaisquer translações causadas pelas rotações da mesa podem ser executadas pelo braço robótico.

3.1.4 Interpolação dos pontos da trajetória

Neste trabalho o resultado da intersecção entre o plano de fatiamento e o modelo 3D, que é de um sólido tipo casca, é uma seção do sólido naquela região do plano. Após um processo de interpolação, esta seção se torna um fio¹ contido no plano de fatiamento.

A interpolação dos pontos da trajetória entre dois planos perpendiculares tem início ao separar os fios do plano superior e inferior em entidades *ACIS* do tipo linha curva ou linha reta. Por exemplo, o fio do fatiamento da Fig. 3.3 seria dividido em uma entidade do tipo linha curva e três do tipo linha reta.

Na etapa seguinte, o programa relaciona cada entidade do fio na camada inferior com os da camada superior. Se planos adjacentes possuírem uma quantidade diferente de entidades, o fio obtido pelo plano inferior será substituído por uma cópia do superior. Essa etapa insere no processo um erro, que termina quando a interpolação atinge o plano superior.

A interpolação é então efetuada calculando pontos intermediários entre as entidades relacionadas dos fios inferior e superior, incrementando a altura de cada ponto interpolado. No casos de entidades curvas, interpolações são feitas ao longo das entidades. Para entidades retas, apenas os pontos de início da entidade inferior e de fim da entidade superior são inseridos na trajetória.

O incremento na altura de um ponto é dependente do comprimento da entidade com relação ao comprimento total do fio de intersecção; dessa forma, as extremidades de entidades mais longas recebem um incremento de altura maior com relação às entidades de menor comprimento.

¹ O termo "fio" é utilizado pela forma como o ACIS se refere a este tipo de entidade: wire [24].

3.1.5 Transformações de rotação dos fatiamentos

Sendo contidos em um plano, os fios de interseção tem uma orientação que coincide com o eixo de crescimento da peça naquela região. Nos casos em que tal orientação não seja paralela ao eixo vertical, a presença de uma mesa posicionadora pode ser considerada, como discutido na seção 3.1.3.

Nessas ocasiões, o fio deve sofrer uma rotação com relação ao eixo de rotação da mesa posicionadora. Como o eixo de rotação não intercepta o próprio fio, este sofre uma translação no espaço, além da própria rotação intencionada. Apenas após essa transformação é que o procedimento de interpolação dos pontos, descrito acima, tem início.

Deve-se chamar a atenção para o fato de que não existe interpolação de orientações da mesa posicionadora para cada ponto: a orientação da mesa é alterada apenas quando um novo fatiamento com orientação diferente for executado. Portanto, todos os pontos interpolados entre dois planos de fatiamentos terão a mesma orientação para a mesa posicionadora.

Para este trabalho, assume-se a utilização de uma mesa posicionadora cujos eixos de rotação passem pelo centro da mesa, e, portanto, este centro não pode sofrer translações causadas por quaisquer rotações. Cabe ao usuário determinar se a mesa utilizada permite as orientações para reprodução adequada da peça desejada.

3.1.6 Estratégias para o processo de construção das peças

Existem modelos que não podem ser completamente construidos por *SMD*, devido a algum atributo da geometria inviável de ser reproduzido por processos de soldagem. Exemplos incluem peças com furos ou com protuberâncias que surgem ou terminam de forma repentina, longe da base do modelo. A reprodução completa de tais modelos exigiria a deposição de solda sobre locais sem material para suportar o depósito.

Nesses casos, seria necessário considerar a alteração do modelo 3D ou a execução de um processo de usinagem posterior para finalização das peças.

Para as peças que apresentam furos, decidiu-se por criar uma trajetória que deposite solda nas regiões onde os furos estariam presentes, para que estes sejam criados por um processo de usinagem posterior à prototipagem rápida.

No caso de modelos com protuberâncias, a estratégia consiste de verificar se esta afeta a detecção do eixo de crescimento de tal forma que uma solução de construção seja obtida com uso de uma mesa posicionadora. Na ocorrência de falha, uma solução alternativa seria

separar a protuberância do resto da peça em modelos diferentes, produzir as partes, e unilas em um processo de soldagem posterior.

3.2 DESENVOLVIMENTO DO APLICATIVO DE FATIAMENTO

A versão do ACIS utilizada para o aplicativo foi a R13, sendo que a versão mais recente é a R23, lançada em julho de 2012 [27].

Devido à utilização de uma versão desatualizada do *ACIS*, não foi possível compilar código em C++ para a criação de um aplicativo em separado que utilize a plataforma 3D. Sendo assim, decidiu-se por incorporar o *Scheme AIDE* no desenvolvimento do software de fatiamento, e por esse motivo toda a programação do aplicativo foi realizada em linguagem *Scheme*.

Como visto na seção 2.4.1, o arquivo "acisinit.scm" foi configurado com o local e nome dos códigos-fonte utilizados neste trabalho (Apêndice II). Entretanto, o local dos códigos precisava ser atualizado manualmente, o que gerava atraso ao tentar rodar o programa em computadores diferentes.

Por esse motivo, um programa em C++ (código no Apêndice I) foi escrito para editar o conteúdo de "acisinit.scm" com o local adequado dos arquivos necessários, para em seguida executar o *Scheme AIDE* dando continuidade ao processo de fatiamento.

3.2.1 Organização do código

O aplicativo tem seu código organizado de acordo com o diagrama na Fig 3.7, que exibe os arquivos ".scm" desenvolvidos para este trabalho e a forma como eles se comunicam.

Após o arquivo "acisinit.scm" ser editado com os endereços corretos dos modelos e dos outros arquivos necessários, o arquivo "parametros_basicos.scm" é inicializado. Este arquivo foi separado para que o usuário possa facilmente editar diversas configurações de fatiamento, simulação, soldagem e envio de dados.

A seguir, o arquivo "incializar.scm" é executado para preparar as configurações iniciais de seleção e visualização do modelo 3D, para então o arquivo "fatiamento.scm" dar continuidade aos procedimentos de fatiamento discutidos ao longo da seção 3.1.

A Tabela 3.1 apresenta a função de cada um dos arquivos ".scm" desenvolvidos neste projeto. Os arquivos ".scm" presentes na pasta "scm" e que não estão listados na Tab 3.1 são arquivos originais do *Scheme AIDE* para permitir o seu funcionamento básico.



Figura 3.7: Diagrama exibindo a organização do código do software desenvolvido.

Tabela 3.1 - Nome e função dos arquivos ".scm" desenvolvidos no trabalho

Nome (.scm)	Função		
acisinit	Executar todos os arquivos descritos a seguir.		
inicializar	Realizar ajustes iniciais para o processo.		
fatiamento	Gerenciar o processo de fatiamento como um todo.		
parametros_basicos	Concentrar configurações diversas para fácil edição pelo usuário.		
centro_geometrico	Determinar o centro geométrico de um fio de interseção. É análogo ao centro de área, neste caso.		
verifica_objeto	Determinar a presença do objeto em um dado plano. É utilizado para identificar o fim do fatiamento.		
constroi_fio_inter	Prepara um fio de intersecção a partir da região de interseção do modelo com o plano de fatiamento.		
constroi_fio_furos	ldem a "constroi_fio_inter.scm", mas considerando a presença de furos na região de interseção.		
detecta_furos	Verificar a presença de furos no objeto para um dado plano de fatiamento.		
otimizar_pontos	Ajustar uma lista de pontos de trajetória fornecidos de modo que pontos pertencentes a uma mesma reta sejam minimizados.		
ajusta_direcao_plano	Realizar ajustes de arredondamento para valores razoáveis das coordenadas do vetor que representa a direção do plano de fatiamento. Também aplica o limite máximo de variação de ângulo de uma direção para a próxima.		
transforma_fios	Rotacionar os fios de interseção em torno do eixo de rotação adequado para cada fio.		
obtencao_pontos	Separar as entidades do fio de intersecção, e verificar se são lineares ou curvas.		
entidade_linear	Realizar a interpolação dos pontos para uma entidade linear.		
entidade_nao_linear	Realizar a interpolação dos pontos para uma entidade não-linear (curva).		
texto_workspacelt	Gravar os pontos da trajetória em um arquivo texto no formato apropriado para leitura no software Workspace LT.		
texto_sports3 Gravar os pontos da trajetória em um arquivo texto no fo apropriado para leitura no software SportS3.			
escreve_quaternions	Escrever em arquivo-texto as orientações da mesa posicionadora, representadas em quatérnions.		
vetor_quaternion	Transformar uma orientação da mesa posicionadora de vetorial (representando a direção do eixo z de mesa com relação ao eixo vertical) para quatérnion.		

3.2.2 Modelos 3D para fatiamento

Para testar o aplicativo desenvolvido, foram criados variados modelos 3D de peças com diferentes formatos e características. Estes modelos foram gerados utilizando o próprio *Scheme AIDE* usado no desenvolvimento do aplicativo de fatiamento. Isso permitiu salvar os modelos no formato ".SAT", que é o formato para modelos do *ACIS*.

Segue agora uma sucinta descrição dos modelos utilizados para averiguar a eficácia do procedimento de fatiamento neste trabalho:



Figura 3.8: Modelo 3D de um cubo simples, salvo em formato AC/S como "cubo.sat".



Figura 3.9: Modelo 3D de um cilindro simples, salvo em formato AC/S como "cilindro.sat".

As Figuras 3.8 e 3.9 exibem os modelos mais simples gerados para este trabalho. Consistem respectivamente de um cubo e um cilindro, ambos vazados para representar sólidos do tipo casca. Estes modelos são de interesse para verificar a interpolação correta de entidades lineares ou curvas, e não possuem nenhuma característica que insira dificuldade na aquisição adequada da trajetória da tocha de soldagem. A execução correta do processo de fatiamento nestes modelos é o mínimo que se pode exigir do software sendo desenvolvido.

A Figura 3.10 exibe o modelo de uma peça piramidal vazada e cortada antes da ponta. Este modelo é de interesse para a verificação da interpolação correta de pontos da trajetória para um caso em que as dimensões da região de intersecção entre o modelo e os planos de fatiamento variem ao longo do modelo.



Figura 3.10: Modelo 3D de uma pirâmide, salvo em formato ACIS como "piramide.sat".

O modelo exibido na Fig. 3.11 é de interesse para averiguar se o erro exibido pela Figura 3.4 e discutido na seção 3.1 ocorre na estratégia adotada, enquanto que as Fig. 3.12 e 3.13 exibem peças com um ou mais furos, o que testará se o algoritmo de construção dos fios de intersecção obedece a estratégia para o processo de construção de peças com furos discutida na seção 3.1.6.



Figura 3.11: Modelo 3D de uma peça com crescimento no sentido de semi-retas partindo da origem, salvo em formato *ACIS* como "comp1.sat".



Figura 3.12: Modelo 3D de um cilindro com furos, salvo em formato ACIS como "cilindro_2furos.sat".



Figura 3.13: Modelo 3D de uma pirâmide com furo em uma lateral, salvo em formato AC/S como "piramide_1furo.sat".

Os procedimentos desenvolvidos para operação conjunta com uma mesa posicionadora (seções 3.1.2, 3.1.3 e 3.1.5) serão testados utilizando o modelo da Fig. 3.14, e a estratégia inicial para o processo de construção de peças com protuberâncias (seção 3.1.6) será analisada após o fatiamento do modelo exibido na Fig. 3.15.



Figura 3.14: Modelo 3D de uma peça com eixo de crescimento não coincidente com o eixo vertical, salvo em formato *ACIS* como "inclinado1.sat".



Figura 3.15: Modelo 3D com surgimento de protuberancia fora da base, salvo em formato ACIS como "protuberância.sat".

Para a realização do ensaio principal de deposição será utilizado o modelo exibido na Fig. 3.16. Decidiu-se por realizar, antes da execução deste ensaio, um ensaio auxiliar para determinar a taxa de crescimento da solda com os parâmetros de soldagem selecionados (seção 3.5), o que influenciará na definição da distância dos planos de fatiamento do modelo. Este ensaio auxiliar deve ser utilizado o modelo exibido na Fig. 3.17, que consiste de uma versão reduzida e cortada do modelo exibido nas Fig. 3.8.



Figura 3.16: Modelo 3D utilizado para o ensaio principal de deposição, salvo em formato ACIS como "piramideS.sat".



Figura 3.17: Modelo 3D utilizado no ensaio auxiliar de deposição, salvo em formato AC/S como "bloco_reduzido.sat".

3.3 AMBIENTE PARA SIMULAÇÃO COM WORKSPACE LT®

Neste trabalho, o software Workspace LT será utilizado para simular um robô manipulador percorrendo a trajetória obtida pelo software de fatiamento desenvolvido com uma tocha de soldagem *GMAW*. Com isso, busca-se confirmar se a trajetória obtida reproduz a geometria do modelo 3D desejado.

Para atingir esse objetivo, criou-se no ambiente de simulação a célula de soldagem robotizada exibida na Fig. 3.18. A célula consiste de um robô *IRB 2000* equipado com uma tocha de soldagem *GMAW* genérica com a ponta orientada na vertical, além de um simples bloco representando uma mesa de suporte para a peça depositada.

No software de fatiamento, existe a opção de gerar os pontos da trajetória em formato a ser reproduzido pelo simulador, utilizando a célula de soldagem da Fig. 3.18. Os Apêndices III e IV exibem exemplos dos códigos gerados para reprodução no *Workspace LT*, que são gravados em dois arquivos-texto nomeados "Track01.KL" e "Track01#.KL", ambos devendo ser movidos para a pasta "ROBOT" do simulador antes da execução da simulação.

O simulador foi configurado para exibir a trajetória percorrida durante a ativação da tocha, mas para não exibir a identificação individual dos pontos da trajetória, pois a sua grande quantidade afetaria a qualidade das simulações.



Figura 3.18: Célula de soldagem desenvolvida no software *Workspace LT* para simular ensaios de deposição.

3.4 CRIAÇÃO DO PROJETO EM SPORT^{S3}®

O software *Sport*^{S3} foi utilizado neste trabalho para a compilação e envio ao robô do código em *ARLA* gerado pelo software de fatiamento. As propriedades exibidas na Fig. 3.19 foram utilizadas para estabeler a comunicação entre o robô e o *Sport*^{S3}.

Após estabelecer contato com o robô, um projeto do SportS3 foi criado. Como visto na Fig. 3.20, no projeto é possível a edição de quaisquer registradores ou entradas e saídas do robô, além de ser possível inserir programas para envio.

Com definition Robot type : IRB 2000 Controller type: S3 M93	 IGRACU IRB 2000 pe: S3 M93 ly: 1 СОМЗ 	Custom name:	00.00	_
Com definition Robot type : IRB 2000 Controller type: S3 M93	ion IRB 2000 ре: S3 M93 lty: 1 СОМЗ 💌	Lustom name.	IGRACU	
Robot type : IRB 2000 Controller type: S3 M93	: IRB 2000 ре: S3 M93 ly: 1 СОМЗ 💌	- Com definition -		
Controller type: S3 M93	ре: S3 M93 ty: 1 СОМЗ 💌	Robot type :	IRB 2000	
	lty: 1 COM3 🔽	Controller type:	S3 M93	
Bobot identity:		Robot identity	1	
	СОМЗ 💌			
COM3 💌		Com port:	COM3 💌	
External axes – – Sensor – – – –				
External axes Sensor			Vision	
External axes Sensor	J Vision	U 🗾		

Figura 3.19: Propriedades do robô inseridas no software Sport^{S3}.



Figura 3.20: Formato de um projeto do Sport^{S3}.

Das opções exibidas na Fig. 3.20, apenas três são necessárias para o projeto desenvolvido:

 TCP: O registrador de TCP nº 1 (TCP 1) foi editado com os dados da tocha de soldagem GMAW da Fronius instalada no IRB 2000 para os ensaios de deposição. A Tabela 3.2 exibe os estes dados:

Número do Registrador	х	Y	Z	BX	BY	BZ
1	309,1	-0,5	-158,6	309,1	0,5	-150

Tabela 3.2 - Dados para configuração do TCP no IRB 2000.

- Frame: O projeto utiliza o registrador de sistemas de coordenadas nº 3 (FRAME 3) para definição do sistema de coordenadas a ser utilizado como referência pelo programa enviado. Esse sistema terá a mesma orientação do sistema de base do robô, mas posicionado sobre a mesa de trabalho, no centro da placa metálica utilizada como base da peça a ser depositada. A definição exata das coordenadas pode ser obtida ativando o *TCP* adequado e movendo o manipulador robótico até a posição desejada para o sistema de referência: as coordenadas serão exibidas na tela "Status" do *Sport*^{S3}, que monitora em tempo real a posição do robô com relação ao sistema de coordenadas de referência e levando em consideração o *TCP* ativado.
- Programs: O Sport^{S3} não permite a importação de arquivos contendo código, portanto deve-se criar um programa vazio e "colar" o conteúdo presente no arquivotexto gerado pelo software de fatiamento após a gravação da trajetória.

Ao fim do preparo do projeto, os códigos inseridos devem ser compilados, para em seguida o projeto ser enviado.

Para realizar o envio, o robô deve estar em modo automático de operação, e todo o projeto é enviado, editando os registradores e instalando na memória do robô todos os programas criados naquele projeto. Não é possível o envio de programas individuais dentro de um mesmo projeto, o que obriga o usuário a criar mais de um projeto caso os programas sejam muito extensos e não caibam todos na memória do robô.

No caso deste trabalho, os programas são tão extensos que cada peça completa exigiria um projeto individual no *Sport*^{S3}. O Apêndice V ilustra o exemplo de um programa gerado para envio ao robô pelo programa. O arquivo-texto é criado com o nome "trajetória_sports3.txt", e seu conteúdo deve ser copiado para um programa do projeto.

Os detalhes de ativação das saídas presentes no código serão discutidos na seção a seguir.

3.5 CONFIGURAÇÕES GERAIS PARA ENSAIOS DE DEPOSIÇÃO

Para execução correta e segura dos ensaios de deposição, diversas configurações nos equipamentos envolvidos precisam ser ajustadas, definindo o comportamento do robô e da fonte de alimentação da tocha ao longo do processo. Estas estão apresentadas nas sub-seções a seguir.

3.5.1 Configurações do robô e da fonte de alimentação

O Apêndice V exibe um exemplo de código gerado para envio ao robô.

Todas as movimentações pela trajetória são executadas com 10% da velocidade máxima, permitindo o ajuste da velocidade de soldagem pela alteração da velocidade máxima.

As saídas digitais de número 5 e 6 do robô estão conectadas respectivamente às entradas digitais *Welding Start* e *Source Error Reset* da interface *ROB5000*, enquanto que a porta de comunicação número 1 é utilizada para configurar o *Mode bit 0-2* (seção 2.9.1). A configuração utilizada foi a "Parameter Selection internal" (Tab. 2.1): nesta configuração, todas as variáveis de soldagem podem ser alteradas em tempo real pelo operador, através do console presente na fonte.

A entrada *Robot Ready/Quick Stop* está conectada à alimentação dos motores do robô, permitindo o desligamento da fonte de alimentação da tocha em caso de parada de emergência.

A ativação da tocha (*Welding Start*) só ocorre após o manipulador robótico atingir a primeira posição da trajetória, e após o usuário ativar uma das entradas do robô conectada a um botão (entrada digital número 5).

Todas as instruções de posicionamento utilizam a configuração *C*2. Nesta configuração, o robô realiza uma suavização da trajetória na proximidade de cantos, minimizando o tempo em baixa velocidade. Esta configuração é útil em soldagem por evitar o acúmulo excessivo de calor em bordas.

Após percorrer toda a trajetória, a tocha é desativada, e um programa já presente na memória do robô é chamado para retornar o manipulador à posição de repouso.

3.5.2 Variáveis do processo de soldagem

A Tabela 3.3 exibe as variáveis do processo de soldagem *GMAW* utilizadas no início do ensaio de deposição. Destas variáveis a tensão elétrica e a velocidade de alimentação do arame podem ser alteradas para manter a estabilidade da deposição.

A definição dessas variáveis para a execução adequada da prototipagem rápida por *SMD* está fora do escopo deste trabalho, e os dados presentes na Tab. 3.3 foram obtidos de uma pesquisa na área de soldagem em andamento nas dependências do GRACO¹. Essa combinação de dados foi caracterizada como sendo uma que proporciona boa estabilidade e baixo faiscamento durante a deposição.

Tabela 3.3 - Variáveis iniciais do processo de soldagem utilizadas nos ensaios de deposição

Espessura esperada do cordão de solda	Velocidade de alimentação do arame	Tensão elétrica	Velocidade de soldagem	Standoff	Corrente esperada
5mm	6m/min	19V	10mm/seg	16mm	~180A

Neste trabalho os ensaios de deposição têm como foco o percorrimento correto da trajetória obtida. Assim, decidiu-se por não tentar reproduzir a espessura da casca dos modelos 3D (10mm na maioria dos casos), utilizando a espessura na Tab 3.3.

O arame utilizado nos ensaios foi o Tubrod 410 NiMo MC, com 1,2mm de diâmetro.

¹ Trabalho em andamento: Edna M. M. Torres, Jorge A. G. Cruz, Jesus E. P. Lopera, Sadek A. Alfaro. "Parameter optmization in GMAW process with solid and metal cored wires", Departamento de Engenharia Mecânica, Universidade de Brasília.

² Distância entre a tocha de soldagem e a peça de trabalho.

4 **RESULTADOS**

4.1 INTERFACE COM O USUÁRIO

Ao rodar o executável "Software de Fatiamento.exe" o programa edita o arquivo "acisinit.scm" com o endereço correto dos arquivos necessários pelo *Scheme AIDE*, como explicado na seção 3.2, para então executar o console do *Scheme AIDE*.

A execução do aplicativo segue o procedimento sequencial descrito nas etapas a seguir:

 O console pede ao usuário que insira o nome do modelo computacional para fatiamento, seguindo as instruções no console (Fig. 4.1). Para serem localizados apenas pelo nome, os modelos devem se encontrar inseridos na pasta "modelos", que por sua vez se encontra dentro da pasta "FatiamentoCasca".



Figura 4.1: Etapa de aquisição do nome do modelo computacional 3D a ser fatiado.

Se o arquivo inserido de fato se encontrar na pasta "modelos", o Hoops 3D® é inicializado (Fig. 4.2). O Hoops 3D é um aplicativo da Spatial Corps®. utilizado pelo Scheme AIDE para visualização de modelos desenhados nele.



Figura 4.2: Etapa de inicialização do Hoops 3D.

 A janela do *Hoops 3D* abre exibindo o modelo 3D inserido pelo usuário na etapa 1, e o console aguarda resposta do usuário para prosseguir para a próxima etapa (Fig. 4.3).



Figura 4.3: Etapa de exibição do modelo computacional 3D.

- 4. O console pede que o usuário defina a distância que dois planos de fatiamento adjacentes estarão entre si (Fig. 4.4).
- 5. Após receber o valor da distância entre planos, o software indaga o usuário se a presença de uma mesa posicionadora deve ser considerada no fatiamento do modelo. O aplicativo então executa os procedimentos de fatiamento descritos na seção 3.1 e suas sub-seções, ainda sem exibir resultados ao usuário. Ao fim do processo, o console alerta o usuário para prosseguir para a próxima etapa (Fig. 4.5).
- 6. Ao fim do processo de fatiamento, os pontos da trajetória estão armazenados em uma lista. O aplicativo então pergunta ao usuário o programa desejado (*SportS3* ou *WorkspaceLT*) para gerar código com os pontos adquiridos, e o programa procede para gravar um arquivo texto com código, no formato correto para o programa previamente selecionado.



Figura 4.4: Etapa de aquisição da distância entre planos de fatiamento adjacentes.



Figura 4.5: Etapa de execução do fatiamento.

- 7. Os pontos obtidos durante o fatiamento podem conter até quinze casas depois da vírgula, o que ocupa espaço na memória e é exageradamente preciso para um processo *SMD*. Antes de gravar os dados no arquivo texto, o usuário é questionado se ele deseja limitar o número de casas decimais depois da vírgula.
- 8. Caso o usuário tenha escolhido limitar a precisão dos pontos da trajetória, o console pede que o número de casas após a vírgula seja inserido (Figura 4.6). Até este momento os pontos ainda não foram gravados nos arquivos-texto. Se o usuário escolher por não limitar as casas decimais dos pontos, o programa salta esta etapa.



Figura 4.6: Etapa de aquisição da precisão desejada para os pontos da trajetória.

- 9. Os pontos da trajetória são gravados no arquivo texto, e o usuário é questionado se ele deseja visualizar a trajetória obtida pelo processo de fatiamento, sendo alertado de que se a distância selecionada para planos de fatiamento adjacentes for pequena, o tempo de construção da trajetória na tela pode crescer consideravelmente, como elucidado na seção 2.2.
- 10. Se o usuário escolher por visualizar a trajetória obtida, o aplicativo altera a janela do Hoops 3D para um modo que permite a visualização da trajetória, para em seguida exibir na tela a trajetória obtida, que é apresentada como um caminho vermelho que percorre a geometria do modelo. Caso o usuário tenha selecionado o uso com mesa posicionadora, a tela também exibe (em amarelo) os fatiamentos executados no modelo antes da rotação causada pela mesa.
- 11. Ao fim da construção na tela, o usuário pode visualizar a imagem exibida, ou finalizar o programa como instruído (Figura 4.7).



Figura 4.7: Etapa de exibição da trajetória obtida.

4.2 FATIAMENTO DOS MODELOS PROPOSTOS

Todos os modelos computacionais propostos na seção 3.2.2 foram inseridos no aplicativo desenvolvido para verificar o resultado do processo de fatiamento.

4.2.1 Modelos Simples

A Figuras 4.8 exibe o resultado do fatiamento para o modelo do cubo (Fig. 3.8) utilizando distância entre planos de fatiamento consecutivos de 10mm. Esta distância foi selecionada deliberadamente grande para facilitar a visualização da trajetória obtida pelo software. Nenhuma anomalia foi detectada na trajetória desse modelo.

O mesmo pode ser dito para o fatiamento dos modelos exibidos nas Fig. 3.9 e 3.10, ambos utilizando uma distância entre planos de fatiamento de 1 mm, e cujos resultados estão exibidos na Fig. 4.9 e 4.10.



Figura 4.8: Resultado do fatiamento do modelo de cubo na Fig. 3.8, utilizando distância entre planos de fatiamento adjacentes de 10mm.



Figura 4.9: Resultado do fatiamento do modelo de cilindro na Fig. 3.9, utilizando distância entre planos de fatiamento adjacentes de 1mm.



Figura 4.10: Resultado do fatiamento do modelo da Fig.3.10, utilizando distância entre planos de fatiamento adjacentes de 1mm.

A estratégia de fatiamento selecionada, discutida na seção 3.1.1, obteve sucesso na interpolação de pontos para modelos com protuberâncias surgindo na direção de semi-retas partindo da origem (seção 3.1, Fig. 3.4), como comprovado pelas Fig. 4.11 e 4.12.



Figura 4.11: Resultado do fatiamento do modelo da Fig. 3.11, utilizando distância entre planos de fatiamento adjacentes de 1mm.



Figura 4.12: Vista superior do resultado de fatiamento exibido na Fig. 4.11.

Os resultados do fatiamento dos modelos selecionados para os ensaios de deposição (Fig. 3.16 e 3.17) estão exibidos nas Fig. 4.13 e 4.14, todos sendo bem-sucedidos e sem anomalias.



Figura 4.13: Resultado do fatiamento do modelo da Fig. 3.16, utilizando distância entre planos de fatiamento adjacentes de 1mm.



Figura 4.14: Resultado do fatiamento do modelo na Fig. 3.17, utilizando distância entre planos de fatiamento adjacentes de 1mm.

4.2.2 Modelos com furos

Nas Figuras 4.15 e 4.16 estão exibidos os resultados do fatiamento dos modelos com furos (Fig. 3.12 e 3.13, respectivamente). Durante o fatiamento do modelo exibido na Fig. 3.12, o programa foi interrompido por falta de memória *heap* disponível. A distância entre planos de fatiamento, que inicialmente era de 1mm, foi então ajustada para 5mm, reduzindo o custo computacional do fatiamento de entidades não-lineares e permitindo a finalização dos procedimentos de fatiamento.

Em ambos os modelos, foi percebida a ocorrência de erros na obtenção da trajetória nas regiões com furos (Fig. 4.17), resultando em uma geometria que desobedece a estratégia de construção de peças com furos discutida na seção 3.1.6.

O método de obtenção dos fios de interseção, detalhado na seção 3.1.4, não pôde ser utilizado para seções de fatiamento que possuíam aberturas, como furos. Assim, no caso de detecção de aberturas em uma dada seção de fatiamento, o fio de interseção é gerado a partir de pontos obtidos pelo disparo de semi-retas, partindo da origem, atingindo o modelo. Este método se assemelha ao discutido na seção 3.1 para a estratégia de fatiamento em espiral, exceto que todas as semi-retas são disparadas dentro do plano de fatiamento onde o furo foi detectado.

A falha detectada consiste de semi-retas atingindo a face interna dos furos, inserindo pontos que causam ligeiros deslocamentos da trajetória na região dos furos. Como pode ser observado na Fig. 4.16, a utilização de camadas de fatiamento (distância dos planos) relativamente pequenas diminui o efeito.



Figura 4.15: Resultado do fatiamento do modelo da Fig. 3.12, utilizando distância entre planos de fatiamento adjacentes de 5mm.



Figura 4.16: Resultado do fatiamento do modelo da Fig. 3.13, utilizando distância entre planos de fatiamento adjacentes de 1mm.



Figura 4.17: Detalhe das falhas detectadas nas regiões com furos no fatiamento dos modelos das Fig. 3.12 (esquerda) e 3.13 (direita).

4.2.2 Modelos com mesa posicionadora

Quando o usuário seleciona a execução do fatiamento em conjunto com uma mesa posicionadora, o software desenvolvido exibe tanto os fatiamentos no modelo ao longo de seu eixo de crescimento quanto a trajetória a ser percorrida pela tocha de soldagem, obtida após a rotação dos fios de interseção (seção 3.1.5).

Quando em operação com uma mesa posicionadora, a direção do vetor normal dos planos de fatiamento segue a orientação do eixo de crescimento naquela região do modelo. Entretanto, decidiu-se por limitar a variação de direção da normal de um plano para o próximo. Embora possa causar perda de precisão na reprodução do modelo, este limite foi implementado para impedir que fios de interseção cruzem um ao outro, o que resultaria em uma trajetória impossível de ser seguida pelo manipulador. Além disso, independentemente do eixo de crescimento inicial do modelo, a orientação da mesa posicionadora na base da peça é obrigatoriamente paralela ao eixo vertical.

Este limite é representado no código do software pela variável global *limite_angulo_maximo*, podendo ser editada pelo usuário no arquivo "parametros_basicos.scm" para a obtenção de um fatiamento razoável do modelo inserido.

A Figura 4.18 exibe o resultado do fatiamento do modelo inclinado exibido na Fig 3.14, utilizando distância entre planos de fatiamento de 1mm e limite máximo de mudança de direção de 10°. Pode-se observar que a variação permitida foi inapropriada, considerando o modelo e a distância entre planos utilizada. Os planos de fatiamento (em amarelo) cruzaram uns aos outros, resultando em uma trajetória que atravessa alguns pontos mais de uma vez.

A alteração do limite máximo de mudança de direção para apenas 1°, obtido experimentalmente, resultou no fatiamento visualizado na Fig. 4.19, que demonstra menos

falhas. Observa-se que a utilização de um limite ainda menor impediria a aproximação excessiva dos planos na região inferior da imagem.



Figura 4.18: Resultado do fatiamento do modelo exibido na Fig. 3.14, utilizando o auxílio da mesa posicionadora, distância entre planos de fatiamento de 1mm e limite de variação de direção dos planos de 10°.



Figura 4.19: Resultado do fatiamento do modelo exibido na Fig. 3.14, utilizando o auxílio da mesa posicionadora, distância entre planos de fatiamento de 1mm e limite de variação de direção dos planos de 1°.

O modelo exibido na Fig. 3.15, que possui uma protuberância repentina e distante da base, foi inserido diversas vezes no software de fatiamento, com diversas configurações de distância entre planos de fatiamento e limite de variação de direção entre ângulos. Todas as trajetórias obtidas foram insatisfatórias, indicando a ineficácia da estratégia inicial adotada para modelos desse tipo (seção 3.1.6).

A Figura 4.20 exibe o resultado do fatiamento do modelo da Fig. 3.15, utilizando distância entre planos de 5mm e limite de variação de direção de 10°. Analisando a figura, pode-se perceber que a única chance do fatiamento gerar uma trajetória que percorra o caminho da

protuberância seria aumentando tanto a distância entre planos de fatiamento quanto o limite de variação de direção entre planos. Entretanto, os valores utilizados já produzem resultados inaceitáveis: distância entre planos inapropriada para um processo de soldagem e seções de fatiamento cruzando umas às outras.



Figura 4.20: Resultado do fatiamento do modelo exibido na Fig. 3.15, utilizando o auxílio da mesa posicionadora, distância entre planos de fatiamento de 5mm e limite de variação de direção dos planos de 10°.

4.3 SIMULAÇÕES COM WORKSPACE LT®

Com exceção dos modelos que precisam de mesa posicionadora (Fig. 3.14 e 3.15) e do modelo criado para execução do ensaio auxiliar de deposição (Fig. 3.17), todos os modelos criados tiveram sua trajetória simulada no ambiente de soldagem desenvolvido no *Workspace LT* (seção 3.3). Modelos com elementos circulares foram simulados com trajetória resultante do fatiamento utilizando distância entre planos de 5mm, devido ao excesso de pontos sobrecarregar o computador onde as simulações foram realizadas.

A Figura 4.21 ilustra a simulação de duas trajetórias diferentes durante a movimentação do robô, e os resultados das simulações podem ser visualizados nas Fig. 4.22 a 4.28. Todas foram bem sucedidas, com exceção dos modelos com furos (simulações nas Fig. 4.26 e 4.27), cujas falhas foram discutidas na seção 4.2.1.



Figura 4.21: Simulação pelo *Workspace LT* no estágio de construção de duas peças diferentes.



Figura 4.22: Resultado após a simulação pelo *Workspace LT* da construção do modelo da Fig. 3.8, com trajetória obtida após fatiamento com distância entre planos de 1mm.



Figura 4.23: Resultado após a simulação pelo *Workspace LT* da construção do modelo da Fig. 3.9, com trajetória obtida após fatiamento com distância entre planos de 5mm.



Figura 4.24: Resultado após a simulação pelo *Workspace LT* da construção do modelo da Fig. 3.10, com trajetória obtida após fatiamento com distância entre planos de 1mm.



Figura 4.25: Resultado após a simulação pelo *Workspace LT* da construção do modelo da Fig. 3.11, com trajetória obtida após fatiamento com distância entre planos de 5mm.



Figura 4.26: Resultado após a simulação pelo *Workspace LT* da construção do modelo da Fig. 3.12, com trajetória obtida após fatiamento com distância entre planos de 5mm.



Figura 4.27: Resultado após a simulação pelo *Workspace LT* da construção do modelo da Fig. 3.13, com trajetória obtida após fatiamento com distância entre planos de 1mm.


Figura 4.28: Resultado após a simulação pelo *Workspace LT* da construção do modelo da Fig. 3.16, com trajetória obtida após fatiamento com distância entre planos de 1mm.

4.4 ENSAIOS DE DEPOSIÇÃO

As trajetórias utilizadas nos ensaios foram construidas com 1mm de distância entre planos de fatiamento, portanto antecipando uma taxa de crescimento da solda depositada de aproximadamente 1mm/volta.

Entretanto, em ambos os ensaios essa taxa se mostrou superior ao esperado, obrigando a interrupção da deposição antes da construção completa dos modelos devido a diminuição da distância entre o bocal da tocha e a solda depositada, tornando-se após um certo tempo excessivamente pequena (aproximadamente 5mm). Embora ajustes nas variáveis controláveis de soldagem (velocidade de alimentação do arame e tensão elétrica aplicada) tenham sido efetuados durante os ensaios para compensar esse efeito, eles eventualmente passaram a afetar apenas a espessura da camada depositada, não mais tendo efeito sobre a taxa de crescimento.

No início dos ensaios, os processos se mostraram inesperadamente instáveis e com excesso de faiscamento, exigindo constante monitoramento para ajustes na velocidade de alimentação do arame e na tensão elétrica aplicada. Tais condições adversas foram

possivelmente causadas por condições impróprias de limpeza das placas metálicas que receberam a deposição, pois percebeu-se melhoria nos estágios posteriores dos ensaios.

As Figuras 4.29 e 4.30 mostram a peça construída após a realização do ensaio auxiliar, cujo modelo 3D pode ser visto na Fig. 3.17.



Figura 4.29: Peça resultante do ensaio auxiliar de deposição.



Figura 4.30: Peça resultante do ensaio auxiliar de deposição, após limpeza.

Embora o objetivo deste ensaio tenha sido de obter uma taxa de crescimento adequada para o ensaio principal, isso não foi possível devido a dificuldade em monitorar simultaneamente as variáveis de processo, a distância entre o bocal e o depósito de solda, e o número de voltas realizadas pelo manipulador. Este último tendo sido perdido pela baixa velocidade de movimentação do robô.

Decidiu-se então realizar o ensaio principal, para construção da peça de modelo 3D visto na Fig. 3.16, utilizando a mesma distância de fatiamento entre planos (1mm). A Figura 4.31 exibe os resultados do ensaio, que também foi interrompido pela diminuição excessiva do *standoff*.



Figura 4.31: Imagens da peça resultante do ensaio principal de deposição.

Após os resultados obtidos no ensaio auxiliar deciciu-se por alterar as variáveis de soldagem iniciais, exibidas na Tab. 3.3, na tentativa de melhorar a estabilidade durante o ensaio principal. A Tabela 3.4 concentra as variáveis de soldagem iniciais e finais usadas em ambos os ensaios.

Tabela 3.4 - Variáveis controláveis iniciais e finais do processo de soldagem utilizadas nos ensaios de deposição.

Etapa da Deposição	Velocidade de alir (m/	mentação do arame /min)	Tensão elétrica (V)	
	Ensaio Auxiliar	Ensaio Principal	Ensaio Auxiliar	Ensaio Principal
Início	6	5	19	18
Fim	5,5	2,5	14,6	15,5

Antes da execução do ensaio principal, uma câmera termográfica presente no GRACO foi utilizada para filmar o processo. Uma das imagens obtidas pela câmera se encontra na Fig. 4.32. A análise desta imagem está fora do escopo deste trabalho, tendo sido inserida para possível utilização em estudos futuros.



Figura 4.32: Imagem do ensaio principal obtida por câmera termográfica.

4.4.1 Comparações com modelos 3D

Nos ensaios realizados, placas metálicas foram soldadas à mesa de trabalho para deposição da solda. Em ambos os ensaios, a dilatação das placas causada pelo calor acumulado foi suficiente para romper os pontos de solda, fato que só foi percebido após término dos procedimentos.

Esse fator, somado às instabilidades ocorridas durante o processo e à rugosidade natural do processo de soldagem, diminuem o significado e a utilidade de medições precisas realizadas nas peças depositadas. Ainda assim, algumas medições puderam ser realizadas com auxílio de um paquímetro padrão (precisão de 0,05mm) para que comparações sejam feitas com o modelo 3D.

Começando pelo ensaio auxiliar, a Fig. 4.33 exibe a convenção utilizada na identificação dos lados da peça.



Figura 4.33: Identificação dos lados da peça depositada no ensaio auxiliar, para comparação com modelo 3D.

Desconsiderando a espessura da casca, o modelo utilizado neste ensaio possui lados de 145mm de comprimento, sendo a distância percorrida no plano horizontal pelos pontos adjacentes da trajetória.

A Tabela 3.5 exibe os resultados de algumas medições efetuadas.

Número da Medição	Espessura da camada (mm)				Distância interna entre camadas (mm)	
	L1	L2	L3	L4	L1 a L3	L2 a L4
1	6,85	6,90	6,75	7,55	136,40	137,00
2	7,20	7,25	6,70	6,70	138,05	137,65
3	7,30	6,40	6,85	7,00	138,25	138,55
4	6,60	6,85	7,45	6,75	138,40	137,50
5	7,00	7,70	7,25	7,25	138,00	138,00
Média	6,99	7,02	7,00	7,05	137,82	137,74

Tabela 3.5 - Medições de espessura de camada e distância entre camadas na peça resultante do ensaio auxiliar.

Considerando a média das espessuras de camada (7,015mm) e a média das distâncias internas entre camadas (137,78mm), pode-se inferir uma distância média de depósito de solda em linha reta de 144,8mm. Trata-se de um valor próximo dos 145mm de lado do modelo, embora estes dados não tenham a confiabilidade desejada.

A Figura 4.34 exibe as identificações utilizadas para as medições do ensaio principal.

Figura 4.34: Identificação dos lados da peça depositada no ensaio principal, para comparação com modelo 3D.

Diferentemente do ocorrido no ensaio auxiliar, foi possível determinar neste ensaio a última posição do robô antes da interrupção do ensaio pelo número da instrução sendo executada no momento da parada e comparando com o arquivo enviado ao robô.

Assim observou-se que o robô se movia da posição X=64,135 Y=64,135 Z=35,35 para a posição X=-64,135 Y=64,135 Z=35,6. Como o modelo está centrado na origem do sistema de coordenadas adotado, e seus lados estão paralelos aos eixos x e y do sistema, pode-se então concluir que na altura da ocorrência da parada os lados opostos do modelo 3D estavam a uma distância de 128,27mm (desconsiderando a espessura da casca).

A Tabela 3.6 exibe os resultados das medições realizadas na peça depositada. Todas as medições foram realizadas na altura da peça no momento da interrupção do ensaio.

Tabela 3.6 - Medições de espessura de camada, distância entre camadas, e altura final para a peça resultante do ensaio principal.

Número da	Espessura da camada (mm)			Distância interna entre camadas (mm)		Altura (mm)	
Medição	L1	L2	L3	L4	L1 a L3	L2 a L4	
1	4,60	4,70	4,40	4,65	125,80	125,10	42,75
2	4,70	4,10	4,05	4,00	125,55	125,00	42,35
3	4,40	4,15	4,10	4,25	125,75	125,60	47,00
4	4,85	4,00	4,15	4,75	125,85	125,10	46,75
5	4,15	4,15	4,00	5,25	125,50	125,75	45,40
Média	4,54	4,22	4,14	4,58	125,69	125,31	44,85

Nesta peça tem-se uma média de 4,37mm de espessura de camada e 125,5mm de distância entre os lados opostos, ambos obtidos por medições realizadas na altura em que o ensaio foi interrompido. Com esses dados pode-se inferir um distância percorrida em linha reta pela solda de 129,9mm, sendo um valor que também se aproxima do valor pelo modelo computacional (128,27mm). A altura média obtida de 44,85mm, quando comparada a posição da tocha no eixo z (35,6mm), reflete a diminuição da distância entre o bocal da tocha de soldagem e o ponto de deposição, visto que o sistema de coordenadas adotado considerou o *stadoff* inicial desejado.

Novamente deve-se chamar atenção para o fato de que as medições apresentadas nesta seção são, independetemente da precisão adotada, de confiabilidade limitada visto as ocorrências discutidas anteriormente.

5 CONCLUSÕES

5.1 DISCUSSÃO DOS RESULTADOS

Este trabalho pode ser dividido em três etapas principais: desenvolvimento do software, simulações para verificação, e ensaios de deposição.

O aplicativo para fatiamento de modelos 3D desenvolvido utilizou uma estratégia de fatiamento exclusiva, resultando em vantagens e desvantagens quando comparada às estratégias discutidas na seção 3.1.

As vantagens percebidas consistem principalmente da ausência dos defeitos presentes no fatiamento por planos paralelos e por espiral, enquanto que o tempo de processamento foi a desvantagem mais acentuada.

A Figura 4.8 ilustra com clareza o comportamento contínuo da trajetória elaborada, que avança suavemente ao longo do eixo de crescimento, teoricamente não permitindo o acúmulo excessivo de calor ou solda em qualquer ponto percorrido (falha observada na estratégia de fatiamento por planos paralelos). Como a Figura 4.12 claramente confirma, a estratégia adotada anula por completo as falhas que ocorriam em modelos com protuberâncias longas (Figura 3.3) ou de crescimento na direção coincidente com a direção de semi-retas partindo da origem (Figura 3.4).

O processo de fatiamento dos modelos com presença de entidades curvas (Fig. 3.9, 3.11 e 3.12) é razoavelmente mais demorado do que para modelos sem entidades curvas. A interpolação de pontos adotada para entidades não-lineares é mais complexa, consumindo grande quantidade de memória do computador, tanto para obtenção dos pontos quanto para exibição da trajetória. De fato, a tentativa de executar o fatiamento desses modelos utilizando distâncias dos planos paralelos pequenas frequentemente resultou no software desenvolvido sendo abortado pelo sistema operacional por excesso de memória consumida.

A distância entre planos de 5mm, que foi utilizada para o fatiamento bem sucedido do modelo cilindrico com dois furos (Figura 4.15), pode ser considerada excessiva, dependendo das características do modelo e das variáveis do processo *SMD* (discutidas na seção 2.2.1).

Na interpolação de pontos para entidades curvas, cálculos são realizados para deixar uma distância fixa entre dois pontos adjacentes da curva. Isso significa que a tocha de soldagem executará um percurso não-linear como um conjunto de segmentos retos, o que já insere erro no processo. Para contornar essa situação, a variável *avanco_interpolacao* pode ser editada em "parametros_basicos.scm" para alterar a distância entre as interpolações de pontos, assim podendo diminuir o tempo de processamento sacrificando precisão.

Como discutido na seção 4.2.2, o software comete erros na obtenção da trajetória para modelos com furos, devido a estratégia utilizada deslocar a trajetória no interior da região com furo.

Na operação em conjunto com mesa posicionadora, a detecção do eixo de crescimento pela variação dos centros de área das seções fatiadas se mostrou eficiente, mas imperfeita. O fatiamento do modelo inclinado (Fig. 3.14) utilizando a mesa posicionadora obteve excelentes resultados, que foram obtidos após repetir o procedimento com diferentes combinações de distância de fatiamento entre planos e limite de variação na direção normal do plano de fatiamento. Entretanto, ao executar o fatiamento de modelos com furos percebe-se que a falha mencionada acima desloca o centro de área das seções, causando falsa detecção de variação no eixo de crescimento do modelo e incitando rotações desnecessárias na mesa posicionadora.

Após os resultados da seção 4.2.3 para o fatiamento de um modelo com protuberância, a estratégia definitiva para o processo de construção de peças com essa característica é a alternativa discutida na seção 3.1.6, que envolve a divisão do modelo em sub-modelos, para processos de usinagem posteriores à construção dos sub-modelos.

Na etapa de simulações pelo *Workspace LT*, todas as características mencionadas nos fatiamentos dos modelos pelo software desenvolvido (seção 4.2) foram reproduzidas. Assim conclui-se que as simulações confirmam a confiabilidade dos resultados exibidos.

O envio de códigos gerados pelo software ao robô foi bem sucedido, embora sua memória para armazenamento de programas fosse um limitante. Em particular, percebeu-se que modelos com regiões não-lineares possuem em geral uma quantidade de pontos que se encontra além da capacidade do robô.

As principais conclusões dos ensaios de deposição são a necessidade de fixação confiável da placa onde o depósito de solda será realizado e a seleção correta da distância de fatiamento (seção 4.4). Em ambos os ensaios, a distância entre planos de fatiamento se mostrou inferior à taxa de crescimento da peça depositada, causando interrupção antecipada do ensaio. Além disso, a fixação da placa metálica por pontos de solda à mesa de trabalho se mostrou insuficiente nas duas ocasiões, visto que a dilatação das placas causada pelo acúmulo de calor se mostrou suficiente para romper os pontos de solda.

Finalmente, embora de confiabilidade limitadas, as comparações geométricas das peças construídas com seus respectivos modelos 3D (seção 4.4.1) confirmam o bom funcionamento do aplicativo de fatiamento desenvolvido. Embora este ainda precise de melhorias, o objetivo de permitir a construção de peças por prototipagem rápida pelo processo *SMD* foi alcançado.

69

5.2 TRABALHOS FUTUROS

Com relação a melhorias possíveis no software desenvolvido, seguem abaixo sugestões do autor:

- Correção das falhas no fatiamento de modelos com furos, mencionados anteriormente;
- Implementar solução para fatiamento adequado de modelos com protuberâncias sem a necessidade de alterações no modelo original por parte do usuário;
- Obter uma versão mais atualizada do ACIS, que traria uma série de benefícios, entre eles: permitir a criação de modelos pelo SolidWorks (atualmente criados por linhas de comando no Scheme AIDE); alterar a programação do software para linguagem C++ (mais conhecida e que permitiria a criação de uma interface gráfica de mais fácil entendimento ao usuário); acesso a novas funções que possam facilitar alguma etapa do procedimento; entre outros;
- Inserir opções de geração de código para outros robôs;
- Elaborar solução para mesas posicionadoras de diferentes características, inclusive com opções para outras representações de orientação;
- Inserir no código enviado ao robô paradas para limpeza do bocal da tocha de solda, pois foi percebido que o acúmulo de sujeira durante os ensaios pode chegar ao nível de interromper o fluxo do gás de proteção;
- Quaisquer melhorias não mencionadas aqui;

A seguir estão algumas sugestões que concernem o processo SMD como um todo:

- Realizar análises metalurgicas e termográficas das peças depositadas;
- Determinar variáveis do processo GMAW para realização bem sucedida de ensaios de deposição;
- Estudar e implementar melhorias no ambiente do GRACO para realização dos ensaios;
- Utilizar técnicas de controle para diminuir a necessidade de constante monitoramento durante os ensaio;

REFERÊNCIAS BIBLIOGRÁFICAS

[1] Kalpakjian, Serope; Steven Schmid (August 2005). Manufacturing, Engineering & Technology. Prentice Hall. pp. 22–36, 951–988. ISBN 0-13-148965-8.

[2] Ian Campbell, David Bourell, Ian Gibson, (2012), "Additive manufacturing: rapid prototyping comes of age", Rapid Prototyping Journal [0268-3768], Vol. 18 Iss: 4 pp. 255 - 258

[3] Cao, W. Miyamoto, Y. (2003) "Direct Slicing from AutoCAD Solid Models for Rapid Prototyping", The International Journal of Advanced Manufacturing Technology, Vol. 21, Iss: 10 pp. 739 - 742

[4] Gustavo Escobar-Palafox, Rosemary Gault, Keith Ridgway, (2011), "Robotic manufacturing by shaped metal deposition: state of the art", Industrial Robot: An International Journal, Vol. 38 Iss: 6 pp. 622 - 628

 [5] Bonaccorso, Filippo. (2011), "An Arc Welding Robot Control for a Shaped Metal Deposition Plant: Modular Software Interface and Sensors", IEEE Transactions on Industrial Electronics [0278-0046], Vol: 58 Iss: 8 pp: 3126 - 3132

[6] Anca, A. (2001) "Computational modelling of shaped metal deposition", International journal for numerical methods in engineering [0029-5981], Vol: 85 Iss: 1 pp: 84 - 106

 [7] Skiba, Tomas. (2009), "Microstructure and Mechanical Properties of Stainless Steel Component Manufactured by Shaped Metal Deposition", ISIJ International [0915-1559], Vol: 49 Iss: 10 pp: 1588 -1591

[8] Clark, D ; Bache, MR ; Whittaker, MT. (2008), "Shaped metal deposition of a nickel alloy for aero engine applications", Journal of Materials Processing Technology, Vol.203 (1-3), pp.439-448

[9] CAMPOS, R. F. E., (2008). "Desenvolvimento e Implementação em Labview de Monitoramento e Controle dos Sistemas de Arrefecimento e Movimentação de uma Bancada Experimental para Soldagem 3D". Trabalho de Graduação em Engenharia de Controle e Automação, Publicação FT. TG-nº 24/2008, Faculdade de Tecnologia, Universidade de Brasília, Brasília, DF, 105p.

 [10] Silveira, S., (2003), "Desenvolvimento de um software em Java 3D para prototipagem rápida", Trabalho de Graduação em Engenharia de Controle e Automação, Publicação FT. TG-nº 009/2003, Faculdade de Tecnologia, Universidade de Brasília, Brasília, DF, 86p.

[11] Chiaverini, V., (1986) "Tecnologia Mecânica/Volume II: Processos de Fabricação e Tratamento",
 2^a edição - São Paulo : McGrall-Hill, pp. 161.

[12] Modenesi P., Marques P., (2000), "Soldagem I - Introdução aos Processos de Soldagem".Universidade Federal de Minas Gerais, Departamento de Engenharia Metalúrgica. Belo Horizonte.

[13] Bracarense, A. Q., (2003), "Gas Metal Arc Welding", Universidade Federal de Minas Gerais, Escola de Engenharia, Programa de Pós-Graduação em Engenharia Mecânica. Disponível em <http://www.infosolda.com.br/artigos/prosol18.pdf>. Acesso em 18/09/2012. [14] M. Tanaka, T. Tamaki, S. Tashiro, K. Nakata, T. Ohnawa, T. Ueyama, (2008), "Characteristics of ionized gas metal arc processing", Surface and Coatings Technology, Vol: 202, Iss: 22 - 23, pp. 5251-5254, ISSN 0257-8972

[15] "Global robotics market showed strength during 2010." Plant Engineering (2011): 26. Academic OneFile. Web. 17 Sep. 2012.

[16] Mike Wilson, (2002), "Robots at the heart of the welding system", Industrial Robot: An International Journal, Vol. 29 Iss: 2 pp. 91 - 92

[17] Gunnar Bolmsjö, Magnus Olsson, Per Cederberg, (2002),"Robotic arc welding - trends and developments for higher autonomy", Industrial Robot: An International Journal, Vol. 29 Iss: 2 pp. 98 - 104

[18] Chen, S., (2003), "Robotic Welding Systems with Vision-Sensing and Self-learning Neuron Control of Arc Welding Dynamic Process", Journal of intelligent & robotic systems [0921-0296], Vol:36 Iss:2 pp. 191 -208

[19] Craig, John J., (2005), "Introduction to Robotics: Mechanics and Control", 3^a edição, Pearson Education Internacional, USA, pp 3-5, pp 15-46

[20] Kulkarni, P., Dutta, D., (1996), "An accurate slicing procedure for layered manufacturing", Computer-Aided Design [0010-4485/96], Vol. 28, Iss: 9, pp. 683-697

[21] Baufeld, Bernd; Biest, Omer van der; (2009), "Mechanical properties of Ti-6Al-4V specimens produced by shaped metal deposition", Science and Technology of Advanced Materials [1468-6996], Vol. 10, Iss: 1

[22] Portal: ACIS. Spatial Docs: External Product Documentation. 10 de maio de 2010. Disponível em http://doc.spatial.com/index.php/Portal:ACIS. Acesso em: 13 de setembro 2012.

[23] Eckel, Bruce, (2000), "Thinking in C++", Vol: 1, 2nd edition, Prentice Hall, Upper Saddle River, New Jersey, pp 7

[24] "Online Help for 3D ACIS Geometric Modeller R13", CD de instalação do ACIS R13.

[25] Edwin D. Reilly (2003), "Milestones in computer science and information technology", Greenwood Publishing Group. pp. 156–157. <u>ISBN 978-1-57356-521-9</u>.

[26] R. Kelsey, W. Clinger, J. Rees (eds.), (1998), "Revised⁵ Report on the Algorithmic Language Scheme", Higher-Order and Symbolic Computation, Vol. 11, Iss. 1, pp. 7 - 105

[27] Santos, Vítor M. F; (2003/2004), "Robótica Industrial", Universidade de Aveiro , Departamento de Engenharia Mecânica, pp. 1-4 a 2-2, 3-16 a 3-27

[28] "Programming Manual - Robot Control system S3/M93", (1993), 3HAB 0002-2, ABB Robotics Products

[29] "Product Manual IRB 2000 / M93A" (1993), 3HAB 0007-12, ABB Robotics Products

[30] Hoang-Lan Pham; Perdereau, V.; Adorno, B.V.; Fraisse, P., "Position and orientation control of robot manipulators using dual quaternion feedback," *Intelligent Robots and Systems (IROS), 2010*

IEEE/RSJ International Conference on, vol., no., pp.658,663, 18-22 Oct. 2010 doi: 10.1109/IROS.2010.5651097

[31] Goldman, Ron; (2011), "Understanding Quaternions", Graphical Models [1524-0703], Vol. 73, pp 21-49

[32] Pervin, Edward and Webb, Jon A., "Quaternions in computer vision and robotics" (1982). *Computer Science Department.* Paper 2476.

[33] "WorkspaceLT 1.2 User Guide", (2012), WAT Solutions

[34] Portal: Workspace LT - Robot and Automation Simulation Software for Education. Disponível em <<u>http://www.workspacelt.com/WorkspaceLT/Features.htm</u>>. Acesso em: 04 de abril de 2013.

[35] Narayan, K. Lalit (2008). "*Computer Aided Design and Manufacturing*." New Delhi: Prentice Hall of India. p. 3.

[36] Portal: SPORT^{S3} - New blood for old robots. Disponível em http://www.ppm.no/section.php?section=sport. Acesso em: 05 de abril de 2013.

[37] Ajuda HTML do software Sport^{S3}.

[38] Fronius ROB 4000/5000 MIG/MAG ROB 5000 OC MIG/MAG Operating Instructions Robot interface, 2005.

[40] Product Focus: R23. Spatial: Technology Solutions to Simplify 3D Application Development. Julho de 2012. Disponível em <u>http://www.spatial.com/resources/product-focus-r23</u>. Acesso em: 30 de setembro 2012.

APÊNDICES

Apêndice I - Código em C++ para atualização do endereço atual dos arquivos

Apêndice II - Código em "acisinit.scm"

- Apêndice III Exemplo de código com definição da trajetória, gerado para simulação no WorkspaceLT
- Apêndice IV Exemplo de código para execução do processo, gerado para simulação no Workspace LT
- Apêndice V Exemplo de código gerado para envio ao robô pelo Sport^{S3}

Apêndice VI - Exemplo de orientações para envio à mesa posicionadora

#include <windows.h>
#include <string>
#include <fstream>
#include <list>

using namespace std;

int main(){

// Prepara string para adquirir endereço dos arquivos TCHAR infoBuffer [1024];GetCurrentDirectory(1024, infoBuffer);string localInicial = infoBuffer;

```
string localFinalExecutavel = localInicial + "/FatiamentoCasca/SoftFat.exe";
```

//Prepara string com o endereço completo de SoftFat.exe
char * endereco_do_executavel = new char[localFinalExecutavel.size() + 1];
std::copy(localFinalExecutavel.begin(), localFinalExecutavel.end(), endereco_do_executavel);
endereco_do_executavel[localFinalExecutavel.size()] = '\0';

```
// Prepara strings para substituir linhas 2 e 3 de acisinit.scm com o endereço atualizado
localInicial = "" + localInicial;
string linha2 = "(set! load-path '(" + localInicial + "/FatiamentoCasca" + "/scm" + "" + "))";
string linha3 = "(set! part-load-path " + localInicial + "/FatiamentoCasca" + "" + ")";
```

```
char backward_slash = localInicial[3];

// Corrige barras \ em barras / para a linha 2

for( int i = 0; i < linha2.size(); i++)

if( linha2[i] == backward_slash )

linha2[i] = '/';
```

// Corrige barras \ em barras / para a linha 3 for(int i = 0; i < linha3.size(); i++) if(linha3[i] == backward_slash) linha3[i] = '/';

//Abrir acisinit.scm para obter conteudo

```
fstream acisinit;
string linha_atual;
list<string> lista_de_strings;
```

```
acisinit.open ( "acisinit.scm", fstream::in | fstream::out );
```

```
string linha_para_substituir1 = "(set! load-path";
string linha_para_substituir2 = "(set! part-load-path";
```

```
while ( acisinit.good() ){
```

getline (acisinit, linha_atual);

```
if( linha_atual.substr(0,15) == linha_para_substituir1 )
    lista_de_strings.push_back( linha2 );
```

else

else

```
lista_de_strings.push_back(linha_atual);
```

}

```
acisinit.close();
```

```
remove( "acisinit.scm" );
```

```
//Abri acisinit.scm para apagar conteudo atual e substituir com conteudo desejado
ofstream acisinit2( "acisinit.scm" );
list<string>::iterator it;
```

acisinit2.close();

```
//Executa o SCHEME AIDE do ACIS
```

ShellExecute(GetDesktopWindow(), "open", endereco_do_executavel, NULL, NULL, SW_SHOWNORMAL);

return 0;

}

;; Rodrigo Castro Andrade 07/52088

- ;; Trabalho de Graduação
- ;; Software de Fatiamento de sólidos tipo casca

;; Endereço dos arquivos precisa ser atualizado aqui

(set! load-path '("C:/Dropbox/Trabalho de Graduação/TG/FatiamentoCasca/FatiamentoCasca/scm")) (set! part-load-path "C:/Dropbox/Trabalho de Graduação/TG/FatiamentoCasca/FatiamentoCasca")

;; Arquivos padrão (load "acis.scm") (autoload 'describe "describe.scm") (load "trace.scm") (autoload 'entity-stats "entstats.scm") (autoload 'find-ents "findents.scm") (autoload 'env-stats "envstats.scm") (autoload 'viewmod "viewmod.scm") (load "param.scm") (viewmod #t)

;; Arquivos para processos auxiliares (load "parametros_basicos.scm")

;; Arquivos que executam o processo (load "inicializar.scm") (load "fatiamento.scm") APÊNDICE III: Exemplo de código com definição da trajetória gerado para simulação no *Workspace*

TP1 = POS(95,-95,0.1,0,0,0,'LUNT') TP2 = POS(95,95,0.35,0,0,0,'LUNT') TP3 = POS(-95,95,0.6,0,0,0,'LUNT') TP4 = POS(-95,-95,0.85,0,0,0,'LUNT') TP5 = POS(95,-95,1.1,0,0,0,'LUNT') TP6 = POS(95,95,1.35,0,0,0,'LUNT') TP7 = POS(-95,95,1.6,0,0,0,'LUNT') TP8 = POS(-95,-95,1.85,0,0,0,'LUNT') TP9 = POS(95, -95, 2.1, 0, 0, 0, 'LUNT')TP10 = POS(95,95,2.35,0,0,0,'LUNT') TP11 = POS(-95,95,2.6,0,0,0,'LUNT') TP12 = POS(-95,-95,2.85,0,0,0,'LUNT') TP13 = POS(95,-95,3.1,0,0,0,'LUNT') TP14 = POS(95,95,3.35,0,0,0,'LUNT') TP15 = POS(-95,95,3.6,0,0,0,'LUNT') TP16 = POS(-95,-95,3.85,0,0,0,'LUNT')

TP783 = POS(-95,95,195.6,0,0,0,'LUNT') TP784 = POS(-95,-95,195.85,0,0,0,'LUNT') TP785 = POS(95,-95,196.1,0,0,0,'LUNT') TP786 = POS(95,95,196.35,0,0,0,'LUNT') TP787 = POS(-95,95,196.6,0,0,0,'LUNT') TP788 = POS(-95,-95,196.85,0,0,0,'LUNT') TP789 = POS(95,-95,197.1,0,0,0,'LUNT') TP790 = POS(95,95,197.35,0,0,0,'LUNT') TP791 = POS(-95,95,197.6,0,0,0,'LUNT') TP792 = POS(-95,-95,197.85,0,0,0,'LUNT') TP793 = POS(95,-95,198.1,0,0,0,'LUNT') TP794 = POS(95,95,198.35,0,0,0,'LUNT') TP795 = POS(-95,95,198.6,0,0,0,'LUNT') TP796 = POS(-95,-95,198.85,0,0,0,'LUNT') TP797 = POS(95,-95,199.1,0,0,0,'LUNT') TP798 = POS(-37.33,0,773.40,0,0,0,'LUNT') APÊNDICE IV: Exemplo de código para execução do processo gerado para simulação no Workspace LT

PROGRAM Track01

-- Workspace LT KAREL 2 Program for IRB2001 Robot

-- TEACHPOINT DECLARATIONS

VAR

- **TP1 : POSITION**
- **TP2 : POSITION**
- **TP3 : POSITION**
- **TP4 : POSITION**
- TP5 : POSITION
- TP6 : POSITION
- **TP7 : POSITION**
- TP8 : POSITION
- **TP9 : POSITION**

TP10: POSITION

TP791 : POSITION

- **TP792 : POSITION**
- **TP793 : POSITION**
- TP794 : POSITION
- TP795 : POSITION
- TP796 : POSITION
- TP797 : POSITION
- TP798 : POSITION

BEGIN

\$USEMAXACCEL=TRUE %INCLUDE Track01# \$UFRAME=POS(1200,-0,600,0,-0,0,") \$UTOOL=POS(212.6682,0,-211.5995,0,0,0,") \$MOTYPE=LINEAR MOVE TO TP1 -- ! ARCWELDON 100,25 MOVE TO TP2 MOVE TO TP3

MOVE TO TP4 MOVE TO TP5 MOVE TO TP6 MOVE TO TP7 MOVE TO TP8 MOVE TO TP9 MOVE TO TP10 MOVE TO TP11 MOVE TO TP12 MOVE TO TP13 MOVE TO TP14 **MOVE TO TP15** MOVE TO TP16 MOVE TO TP17 MOVE TO TP18 MOVE TO TP19

.

MOVE TO TP783 MOVE TO TP784 MOVE TO TP785 MOVE TO TP786 MOVE TO TP787 MOVE TO TP788 MOVE TO TP789 MOVE TO TP790 MOVE TO TP791 MOVE TO TP792 MOVE TO TP793 MOVE TO TP794 MOVE TO TP795 MOVE TO TP796 MOVE TO TP797 --! ARCWELDOFF MOVE TO TP798 END Track01

COMMENT Ensaio SMD

V=100.0 MAX=1500.0 TCP 1 **RECT COORD** FRAME 0 **COMMENT Reset Weld Start RESET OUTP 5** COMMENT Pwr Src ErrReset SET OUTP 6 **RESET OUTP 6 DELAY 0.01 S COMMENT Sel Manual Mode** LET R 1 = 3TRANSFER R 1 TO PORT NO 1 **COMMENT** Config frame POS V=100.00% FINE X=1603.88 Y=50.88 Z=600 Q1=1.000000 Q2=0.000000 Q3=0.000000 Q4=0.000000 POS V=20.00% FINE X=1603.88 Y=50.88 Z=582.73 Q1=1.000000 Q2=0.000000 Q3=0.000000 Q4=0.000000 FRAME 3 COMMENT Posiciona manipulador POS V=100.00% C2 X=77.181 Y=-77.181 Z=50.1 Q1=1.000000 Q2=0.000000 Q3=0.000000 Q4=0.000000 POS V=20.00% C2 X=77.181 Y=-77.181 Z=0.1 Q1=1.00000 Q2=0.000000 Q3=0.000000 Q4=0.000000 COMMENT Ativa com input WAIT UNTIL INP 5 = 1 SET OUTP 5 POS V=10.00% C2 X=76.509 Y=76.509 Z=0.35 Q1=1.00000 Q2=0.000000 Q3=0.000000 Q4=0.000000 POS V=10.00% C2 X=-76.509 Y=76.509 Z=0.6 Q1=1.00000 Q2=0.000000 Q3=0.000000 Q4=0.000000 POS V=10.00% C2 X=-76.509 Y=-76.509 Z=0.85 Q1=1.00000 Q2=0.000000 Q3=0.000000 Q4=0.000000

POS V=10.00% C2 X=-76.863 Y=76.863 Z=221.6 Q1=1.00000 Q2=0.000000 Q3=0.000000 Q4=0.000000 POS V=10.00% C2 X=-76.863 Y=-76.863 Z=221.85 Q1=1.00000 Q2=0.000000 Q3=0.000000 Q4=0.000000 POS V=10.00% C2 X=76.863 Y=-76.863 Z=222.1 Q1=1.00000 Q2=0.000000 Q3=0.000000 Q4=0.000000 POS V=10.00% C2 X=77.216 Y=77.216 Z=222.35 Q1=1.00000 Q2=0.000000 Q3=0.000000 Q4=0.000000 POS V=10.00% C2 X=-77.216 Y=77.216 Z=222.6 Q1=1.00000 Q2=0.000000 Q3=0.000000 Q4=0.000000 POS V=10.00% C2 X=-77.216 Y=-77.216 Z=222.85 Q1=1.00000 Q2=0.000000 Q3=0.000000 Q4=0.000000 POS V=10.00% C2 X=77.216 Y=-77.216 Z=223.1 Q1=1.00000 Q2=0.000000 Q3=0.000000 Q4=0.000000 COMMENT Desativa tocha **RESET OUTP 5** COMMENT Posicao de repouso CALL PROG 50 STOP

Q1=1 Q2=0 Q3=0 Q4=0 Q1=0.283711 Q2=0 Q3=-0.95891 Q4=0 Q1=-0.83907 Q2=-0.000111 Q3=-0.544023 Q4=0 Q1=-0.759688 Q2=-0.000234 Q3=0.650288 Q4=0 Q1=0.408032 Q2=-0.000781 Q3=0.912967 Q4=0 Q1=0.991082 Q2=0.000739 Q3=-0.133251 Q4=0 Q1=0.004397 Q2=-0.046421 Q3=-0.998912 Q4=0 Q1=-0.131072 Q2=0.086069 Q3=0.98763 Q4=0 Q1=0.511297 Q2=-0.072881 Q3=-0.856308 Q4=0 Q1=-0.817644 Q2=0.085354 Q3=0.569362 Q4=0 Q1=-0.55199 Q2=0.094932 Q3=0.828429 Q4=0 Q1=-0.73003 Q2=-0.030127 Q3=0.682751 Q4=0 Q1=0.855755 Q2=0.002739 Q3=-0.517373 Q4=0 Q1=-0.254232 Q2=0.001609 Q3=-0.967142 Q4=0 Q1=-0.067091 Q2=-0.068092 Q3=0.995421 Q4=0 Q1=-0.990746 Q2=-0.004749 Q3=0.135646 Q4=0 Q1=0.879472 Q2=0.002651 Q3=-0.475943 Q4=0 Q1=-0.694428 Q2=0.047091 Q3=0.718019 Q4=0 Q1=-0.996145 Q2=0.009109 Q3=-0.087249 Q4=0 Q1=0.905884 Q2=0.005483 Q3=-0.42349 Q4=0 Q1=-0.706634 Q2=0.047387 Q3=0.705991 Q4=0 Q1=0.911935 Q2=-0.003594 Q3=-0.410319 Q4=0 Q1=-0.244681 Q2=0.0303 Q3=-0.96913 Q4=0 Q1=-0.885691 Q2=-0.02963 Q3=0.463328 Q4=0 Q1=-0.963063 Q2=0.015927 Q3=0.268806 Q4=0 Q1=0.612194 Q2=0.005711 Q3=0.790687 Q4=0 Q1=0.816008 Q2=-0.000328 Q3=-0.578041 Q4=0 Q1=0.767918 Q2=0.000605 Q3=-0.640548 Q4=0 Q1=-0.846339 Q2=-0.032678 Q3=0.531642 Q4=0 Q1=0.957019 Q2=0.000763 Q3=-0.290024 Q4=0 Q1=0.548791 Q2=0.000896 Q3=0.835959 Q4=0 Q1=0.836845 Q2=5e-005 Q3=-0.54744 Q4=0

Orientações da mesa para cada ponto da trajetória

```
Q1=-0.873305 Q2=0 Q3=-0.487175 Q4=0
Q1=-0.873305 Q2=0 Q3=-0.487175 Q4=0
Q1=-0.873305 Q2=0 Q3=-0.487174 Q4=0
Q1=-0.873305 Q2=0 Q3=-0.487175 Q4=0
Q1=-0.873305 Q2=0 Q3=-0.487175 Q4=0
Q1=-0.873305 Q2=0 Q3=-0.487174 Q4=0
Q1=-0.873305 Q2=0 Q3=-0.487174 Q4=0
Q1=-0.873305 Q2=0 Q3=-0.487175 Q4=0
Q1=-0.873305 Q2=0 Q3=-0.487174 Q4=0
Q1=-0.873305 Q2=0 Q3=-0.487175 Q4=0
Q1=-0.873305 Q2=0 Q3=-0.487175 Q4=0
```

FIM DO ARQUIVO