Sylvia Nathaly Rea Minango

APLICAÇÃO EM ROBÓTICA DO PADRÃO STEP-NC NA GERAÇÃO DE TRAJETÓRIAS DE USINAGEM

Dissertação submetida ao Programa de Pós-graduação em Engenharia Mecânica da Universidade Federal de Santa Catarina para a obtenção do Grau de Mestre em Engenharia Mecânica.
Orientador: Prof. João Carlos Espíndola Ferreira, Ph.D.

Florianópolis
2016

Ficha de identificação da obra elaborada pela autora
através do Programa de Geração Automática da Biblioteca Universitária
da UFSC.

Rea Minango, Sylvia Nathaly
Aplicação em robótica do padrão STEP-NC na geração de trajetórias de usinagem / Sylvia Nathaly Rea Minango ; orientador, João Carlos Espíndola Ferreira - Florianópolis, SC, 2016.
154 p.

Dissertação (mestrado) - Universidade Federal de Santa Catarina, Centro Tecnológico. Programa de Pós-Graduação em Engenharia Mecânica.

Inclui referências

1. Engenharia Mecânica. 2. Robôs industriais. 3. STEP-NC. 4. Geração de trajetórias. 5. Cinemática inversa. I. Ferreira, João Carlos Espíndola . II. Universidade Federal de Santa Catarina. Programa de Pós-Graduação em Engenharia Mecânica. III. Título.

Sylvia Nathaly Rea Minango

APLICAÇÃO EM ROBÓTICA DO PADRÃO STEP-NC NA GERAÇÃO DE TRAJETÓRIAS DE USINAGEM

Esta Dissertação foi julgada adequada para obtenção do Título de "Mestre em Engenharia Mecânica", e aprovada em sua forma final pelo Programa de Pós-graduação em Engenharia Mecânica.

Florianópolis, 29 de fevereiro de 2016

_____

Prof. Armando Albertazzi Gonçalves Jr., Dr.
Coordenador do Curso

Banca Examinadora:

_____

Prof. João Carlos Espíndola Ferreira, Ph.D. – Orientador
Universidade Federal de Santa Catarina

_____

Prof. Daniel Martins, Dr.
Universidade Federal de Santa Catarina

_____

Prof. Carlos Henrique Ahrens, Dr.
Universidade Federal de Santa Catarina

_____

Prof. Nilson Luiz Maziero, Dr.
Universidade de Passo Fundo

Este trabalho é dedicado a duas mulheres maravilhosas, minha mãe Sylvia e minha irmã Carolina, e a dois homens cujas lições estão gravadas no meu coração: meu pai, Luis W. Rea (*in memoriam*) e meu padrinho, Fausto Carrera R. (*in memoriam*).
Esta conquista é para vocês.

AGRADECIMENTOS

Agradeço profundamente a toda minha família, especialmente a minha mãe e minha irmã pelo apoio para cumprir esta meta, a paciência durante todo este processo, e o amor e suporte durante a vida toda.

Ao Prof. João Carlos Espíndola Ferreira, pela excelente disposição e guia durante todo este trabalho, e pela oportunidade dada para desenvolver o meu mestrado nesta prestigiosa universidade.

A Richard, pelas oportunas palavras, as ideias compartilhadas, o apoio constante, mas acima de tudo, pelo amor e a cumplicidade nesta aventura.

À Gisele Orgado, por sua ajuda ao longo da escrita deste trabalho, por sua dedicação e pelo carinho para seus alunos.

A todos os professores do POSMEC, pela dedicação e as valiosas lições, dentro e fora da sala de aula.

Aos colegas do GRIMA, especialmente a Julio, David e Paola, por seus aportes e conselhos oportunos.

Cabe destacar que o presente trabalho foi realizado com apoio do Programa Estudantes-Convênio de Pós-Graduação – PEC-PG, do Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) - Brasil.

One machine can do the work of fifty ordinary men. No machine can do the work of one extraordinary man.

(Elbert Hubbard)

RESUMO

A diversidade de formas de representação de dados, ao longo do ciclo de desenvolvimento de um produto, tem criado a necessidade de uma linguagem comum, capaz de descrever seus dados de projeto, fabricação e medição. A norma ISO 14649, conhecida como padrão STEP-NC, nasceu como um esforço para a padronização do formato de troca de dados de produtos no âmbito da fabricação tipicamente por controle numérico computadorizado (CNC). Porém, apesar de haverem diversos trabalhos envolvendo a aplicação do padrão STEP-NC em máquinas CNC, há uma lacuna em métodos que permitam a aplicação do padrão STEP-NC na geração de programas para robôs industriais, os quais vêm sendo cada vez mais utilizados nas linhas de produção em empresas de manufatura, tanto em quantidade quanto em variedade. Este trabalho propõe um método aplicável a vários tipos de robôs industriais, o qual permite receber informações aderentes ao padrão STEP-NC e gerar as trajetórias para movimentar o robô, acelerando a sua integração na manufatura. O método utiliza o arquivo físico no formato STEP-NC para a usinagem de uma peça e, após as informações no arquivo serem interpretadas, gera-se a movimentação do robô mediante um algoritmo de cinemática inversa, considerando-se os parâmetros cinemáticos específicos de cada robô. O resultado é o conjunto dos pontos da trajetória expresso em função das juntas do robô e das coordenadas dos pontos acompanhadas da orientação do efetuador final. Com esses dados pode-se simular o processo usando-se diferentes softwares e, se necessário, esses dados podem ser traduzidos para a linguagem própria do fabricante do robô mediante pós-processadores. Para implementar o método proposto foi criado um sistema computacional na linguagem Java, o qual foi usado para gerar as trajetórias para a fabricação de duas peças prismáticas, tendo como entrada arquivos no formato STEP-NC. Essas trajetórias foram testadas em três robôs industriais com diferentes morfologias, em um ambiente virtual, comprovando-se a viabilidade da aplicação do método proposto. Este trabalho pretende contribuir para a sistematização da geração de trajetórias para robôs industriais, aderentes ao padrão STEP-NC, visando reduzir significativamente o tempo de programação de robôs, constituindo-se em um trabalho de interesse e utilidade tanto em aplicações industriais quanto no setor acadêmico.

Palavras-chave: Robôs Industriais. STEP-NC. Geração de Trajetórias. Peças Prismáticas. Cinemática Inversa.

ABSTRACT

The different forms of data representation, along the product development process, have created the need for a common language capable of describing the design, manufacturing, and measurement data. The ISO 14649 standard, known as the STEP-NC standard, began as an effort to standardize the product data exchange format within manufacturing typically by computerized numerical control (CNC). However, although there are several studies involving the application of the STEP-NC standard in CNC machines, there is a gap in methods for the application of STEP-NC standard in the generation of programs for industrial robots, which are being increasingly used in production lines in manufacturing companies, both in quantity and variety. This work proposes a method applicable to various types of industrial robots, which allows STEP-compliant information to be received and generates the path along which the robot should move, accelerating the setup and integration of robots in manufacturing. The method uses the physical file in STEP-NC format for machining a workpiece, and after the information in the file is interpreted, the movement of the robot is generated by means of an inverse kinematics algorithm, considering the specific kinematic parameters of each robot. The result is the set of points along the path expressed in terms of the robot joints and the coordinates of the points, together with the orientation of the end effector. With these data one can simulate the process using different pieces of software and, if necessary, these data can be translated into the language of the robot by postprocessors. In order to implement the proposed method a computer program was developed using the Java language, which was used to generate the paths to manufacture two prismatic parts, having as input the files in the STEP-NC format. These paths were tested on three industrial robots with different morphologies, in a virtual environment, confirming the feasibility of the proposed method. This work seeks to contribute to the systematization of path generation for industrial robots, compliant with the STEP-NC standard, in order to reduce significantly the robot programming time, which makes this work important for both industry and academia.

Keywords: Industrial Robots. STEP-NC. Path Generation. Prismatic Parts. Inverse Kinematics.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

CAD	Projeto Assistido por Computador *(Computer-Aided Design)*
CAM	Manufatura Assistida por Computador *(Computer-Aided Manufacturing)*
CAPP	Planejamento de Processo Assistido por Computador *(Computer-Aided Process Planning)*
CAx	Tecnologias Assistidas por Computador (*Computer-Aided Technologies*)
CNC	Controle Numérico Computadorizado *(Computer Numerical Control)*
DH	Denavit-Hartenberg
DLS	Mínimos Quadrados Amortecidos *(Damped Least-Squares)*
IDE	Ambiente de Desenvolvimento Integrado *(Integrated Development Environment)*
ISO	Organização Internacional para Padronização *(International Organization for Standardization)*
JAXB	Arquitetura Java para enlaces XML (*Java Architecture for XML Binding*)
JDK	Kit de Desenvolvimento Java *(Java Development Kit)*
JRE	Ambiente de Execução Java *(Java Runtime Environment)*
JSDAI	*Java-Standard Data Access Interface*
NC	Controle Numérico
NIST	Instituto Nacional de Padrões e Tecnologia *(National Institute of Standards and Technology)*
OLP	Programação *off-line*
STEP	Padrão para Intercâmbio de Dados de Produtos *(Standard for the Exchange of Product model data)*
XML	*Extensible Markup Language*

LISTA DE SÍMBOLOS

Alfabeto latino:

a_e	Terceiro vetor coluna da matriz de rotação do efetuador final
A_i	Matriz de transformação do método Denavit-Hartenberg para o elo *i*
a_{i_DH}	Comprimento do elo *i* do método Denavit-Hartenberg
c	Círculo parametrizado
c_{DLS}, p_{DLS}	Coeficientes do método de mínimos quadrados amortecidos
$D(\lambda)$	Matriz diagonal de amortecimento
d_e, effl	Distância a partir do centro do punho esférico até o ponto extremo do efetuador final
deff	Distância a partir do centro da plataforma móvel até o centro do punho esférico
d_{i_DH}	*Offset* do elo *i* do método Denavit-Hartenberg
D_{max}	Maior movimento permitido ao efetuador final para uma iteração
dq	Variação das variáveis das juntas
dX, e	Vetor erro total
dX_p	Vetor erro de posição
dX_r	Vetor erro de orientação
J	Jacobiano
J_i	Aporte de cada junta para o jacobiano
n	Número de lados do polígono
O_e^0	Vetor posição da origem do sistema de referência do elo *e* com referência ao elo *0*
O_i	Origem do sistema de referência do elo *i*
p	Ponto
$\dot{p}$	Velocidade linear
p_e	Posição do efetuador final
p_w	Posição do punho esférico
q	Variáveis das juntas
q^c	Valor inicial das variáveis de juntas
R	Raio da circunferência correspondente ao polígono inscrito
r	Radio da circunferência estabelecido no arquivo STEP-NC
$R(\alpha,u)$	Matriz de rotação da representação ângulo-eixo

r_b	Raio da plataforma fixa
R_e^0	Matriz de rotação do elo e com referência no elo 0
r_p	Raio da plataforma móvel
S_p	Limite de tolerância da posição
S_r	Limite de tolerância da orientação
T_A^B	Matriz de transformação homogênea de A com referência à B
u	Vetor eixo de rotação na representação ângulo-eixo
X_d	Posição desejada
X_e	Posição atual

Alfabeto grego:

α	Ângulo de rotação em torno do vetor *u* na representação ângulo-eixo
α_{i_DH}	Torção do elo *i* do método Denavit-Hartenberg
θ	*Pitch*. Ângulo de rotação em torno do eixo Y
θ_{i_DH}	Ângulo da junta *i* do método Denavit-Hartenberg
λ_i	Coeficiente de amortecimento para a junta *i*
ρ	Ângulo de inclinação do vetor *u* em torno do eixo Z
σ	Ângulo de inclinação do vetor *u* em torno do eixo X
ϕ	*Roll*. Ângulo de rotação em torno do eixo Z
ψ	*Yaw*. Ângulo de rotação em torno do eixo X
ω	Velocidade angular

SUMÁRIO

1. INTRODUÇÃO

1.1 Apresentação do problema

A norma ISO 14649, chamada de STEP-NC, nasce como um esforço para a padronização do formato de troca de dados de produtos, ao longo do processo de fabricação. Na última década, vários trabalhos acadêmicos têm sido produzidos visando a aplicação deste padrão em programas CAD e CAM, na criação de controladores para máquinas de controle numérico aderentes a este padrão, e na análise de desempenho das máquinas ferramentas. Porém, no que diz respeito à integração da robótica e o STEP-NC, as pesquisas não seguiram a mesma tendência, apesar do grande número de robôs inseridos nas linhas de produção nas empresas de manufatura (Sääski; Salonen; Paro, 2005)

São vários os motivos para que a robótica não esteja sendo considerada na aplicação do STEP-NC. Um deles é a complexidade no controle destes equipamentos, pela ampla variedade de arquiteturas de robôs existentes. Cada tipo de robô precisa de um controlador especializado, e cada fabricante desenvolve seus próprios controladores para os robôs da sua marca. Além disso, para que o usuário possa controlar o robô, o controlador precisa receber comandos em uma linguagem específica, própria de cada fabricante, resultando em um número elevado de linguagens de programação de robôs e uma dificuldade a mais para atingir uma padronização.

Com a introdução de robôs nos processos de fabricação, ao problema de falta de padronização na comunicação entre o robô e o controlador somou-se a dificuldade da transmissão dos dados do processo de manufatura requerido para o controlador do robô. Muitos fabricantes criaram aplicações que recebem os dados da peça a ser fabricada em diferentes formatos, que incluem representações CAD (*Computer-Aided Design*) e até o código G, padronizado na norma ISO 6983 (2009). Posteriormente traduzem esses dados para a linguagem exclusiva dos seus robôs, tornando-se necessária a execução de um programa específico para os robôs de cada marca, aumentando a variedade de dados circulando ao longo do processo de fabricação de um produto.

Assim, pode-se concluir que os esforços de padronização não têm sucesso devido: (a) à grande quantidade de linguagens de programação de robôs, (b) aos controladores próprios de cada fabricante, (c) às diferentes representações de dados de usinagem, e (d) aos interesses dos fabricantes em manter sua tecnologia como proprietária. Com o padrão STEP-NC procura-se padronizar o formato para troca de dados, o qual

conta com o apoio de organizações como a NIST (*National Institute of Standards and Technology*), a Boeing, a General Electric e a Siemens, e espera-se assim um avanço mais significativo na implementação de um padrão nesta área (Nguyen, Stark. 2009).

Idealmente, como mostrado na Figura 1.1, o padrão STEP-NC visa ser interpretado diretamente pelo controlador da máquina de comando numérico, ou neste caso o robô manipulador. Em nível acadêmico há pesquisas focadas em controladores de arquitetura aberta para tonar aos robôs industriais mais inteligentes e capazes de interpretar informações aderentes ao padrão STEP-NC, por exemplo Calabrese; Celentano (2007) e Kovács; Szayer; Tajti (2012). A nível comercial, essas funcionalidades poderiam ser incluídas nos novos controladores de cada fabricante de robôs industriais.

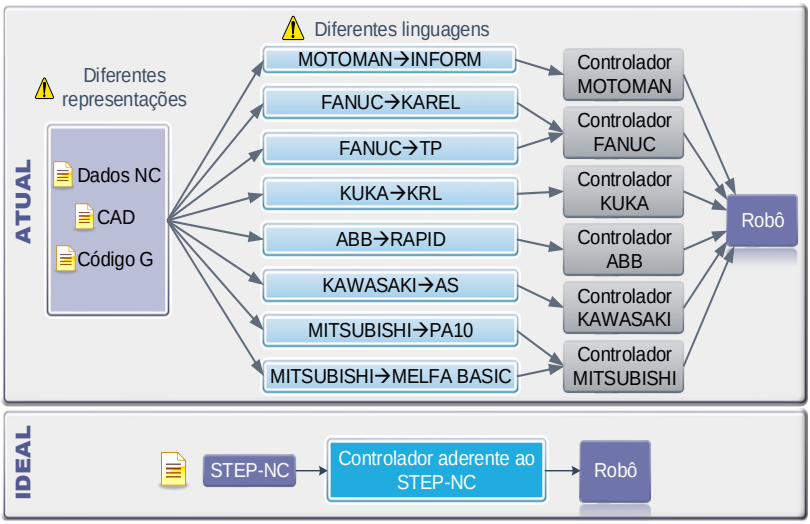


Figura 1.1. Cenários na integração da robótica com o padrão STEP-NC
FONTE: Da autora

Com o desenvolvimento deste trabalho, pretende-se contribuir na produção de métodos que permitam a aplicação do padrão STEP-NC na geração de trajetórias para robôs industriais, sem a necessidade de fazer mudanças no hardware do robô. Para isso propõe-se um método que permita aos robôs industriais receber informações aderentes ao padrão STEP-NC e gerar as trajetórias correspondentes ao processo de fresamento requerido. Essas trajetórias devem ser de fácil inserção em

ferramentas de simulação existentes para a sua visualização, ou para sua tradução para as diferentes linguagens proprietárias mediante a aplicação de pós-processadores correspondentes, como resumido na Figura 1.2.

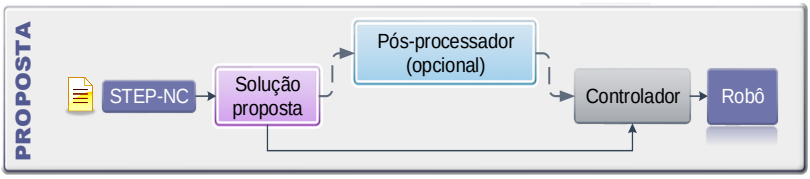


Figura 1.2. Solução proposta
FONTE: Da autora

Esta proposta é diferente do cenário ideal porque visa ser aplicada nos robôs que estão atualmente inseridos nas linhas de fabricação e robôs mais antigos, sem necessidade de adquirir um controlador novo ou hardware adicional, procurando diminuir o tempo de integração do padrão STEP-NC com robôs industriais.

Os trabalhos existentes que abordam a vinculação do STEP-NC com a robótica não se concentraram no desenvolvimento de um sistema computacional que relacione os parâmetros cinemáticos do robô com os dados fornecidos neste padrão. Este trabalho apresenta uma proposta de sistematização da geração de trajetórias para robôs industriais aderente ao padrão STEP-NC, constituindo-se em uma contribuição acadêmica nesta área, além de ser uma base para o desenvolvimento de soluções para a indústria.

1.2 Objetivos

O objetivo principal deste trabalho consiste no desenvolvimento de um método de geração de trajetórias para robôs industriais em ambiente virtual, aplicadas na usinagem de peças prismáticas aderentes ao padrão STEP-NC, para robôs de várias morfologias e independente do fabricante.

Para atingir o objetivo geral, foram estabelecidos os objetivos específicos a seguir:

- Estabelecer um algoritmo baseado em métodos numéricos, orientado à resolução da cinemática inversa de manipuladores industriais seriais, cinematicamente não redundantes;
- Criar uma rotina para geração de trajetórias de fresamento 2 ½D, com base na informação do arquivo físico aderente ao padrão STEP-NC, para a usinagem de cavidades, furos e faces planas;

- Desenvolver um sistema computacional no marco da programação *off-line* para robôs industriais que implemente o método proposto;
- Realizar testes no ambiente virtual utilizando vários tipos de robôs industriais e avaliar a exatidão das trajetórias obtidas mediante o método proposto;

1.3 Estrutura do trabalho

A presente dissertação está dividida conforme detalhado a seguir.

O capítulo 2 contém uma revisão bibliográfica que abrange os aspectos fundamentais do padrão STEP-NC, assim como tópicos relacionados à análise de manipuladores robóticos. Também é apresentada uma compilação dos trabalhos acadêmicos existentes sobre a temática do padrão STEP-NC e sua implementação para robôs industriais.

No capítulo 3 o método proposto é descrito e a metodologia aplicada na sua implementação é detalhada.

Os materiais e os métodos utilizados para a avaliação do trabalho proposto são citados no capítulo 4, incluindo uma descrição do sistema computacional desenvolvido com este fim.

Os resultados obtidos são listados e analisados no capítulo 5, tanto as observações correspondentes à avaliação no âmbito virtual, quanto os dados coletados na avaliação da trajetória obtida para o robô.

Finalmente, as conclusões deste trabalho são apresentadas no capítulo 6, assim como sugestões para trabalhos futuros. Informações adicionais sobre o sistema desenvolvido e os testes realizados podem ser encontrados nos apêndices.

2. FUNDAMENTAÇÃO TEÓRICA

Esta dissertação integra duas grandes temáticas: o padrão STEP-NC e a análise cinemática de manipuladores robóticos. Um breve panorama destes temas é apresentado neste capítulo, assim como uma revisão dos trabalhos existentes relacionados a eles.

Ademais, neste capítulo serão definidos alguns conceitos que são utilizados no decorrer da dissertação nas diferentes temáticas.

2.1 Padrão STEP-NC (ISO 14649)

A diversidade de formas de representação de dados ao longo do ciclo de vida do desenvolvimento de um produto tem criado a necessidade de uma linguagem comum, capaz de descrever os dados de projeto, fabricação e medição. O padrão STEP, descrito na norma ISO 10303, cobre em parte essa necessidade, no que se refere aos dados de projeto e representação geométrica do produto, segundo Hardwick et al. (2013).

O cenário na manufatura é similar. Para transmitir as informações desde o projeto do produto até a máquina ferramenta de controle numérico computadorizado (CNC) que a fabrica, foi estabelecida há várias décadas a norma ISO 6983, conhecida como código G. Porém, desde a sua origem ela não tem evoluído na mesma velocidade que os requerimentos da indústria, o que fez com que os fabricantes introduzissem suas próprias adaptações, criando um sem-número de versões diferentes do código G, dificultando assim a troca de informações entre máquinas de controle numérico (Sääski; Salonen; Paro 2005). Por este motivo, muitos esforços têm sido feitos visando criar um novo padrão para a troca de informações, e um deles é o padrão STEP-NC, padronizado na norma ISO 14649.

O padrão STEP-NC é descrito por Hardwick et al. (2013) como a aplicação dos métodos do padrão STEP em máquinas de controle numérico. De fato, o padrão STEP-NC integra as informações relativas à geometria da peça contidas na ISO 10303 com as operações de usinagem (ISO 14649-1, 2002), criando assim um novo cenário para o ciclo de fabricação de um produto, como mostrado na Figura 2.1. Dentro desse novo cenário, os objetivos do padrão STEP-NC são:

- Cobrir as atuais e futuras necessidades para a troca de dados;
- Apoiar o uso direto de dados de produtos gerados por computador a partir da norma ISO 10303;

– Criar um modelo de troca de dados orientado à peça para máquinas-ferramentas de controle numérico;
– Usar linguagens e bibliotecas modernas e padronizadas para a implementação do modelo de dados;
– Garantir a compatibilidade dos dados de entrada do CNC.

(ISO 14649-1, 2002, p.4. Tradução nossa)

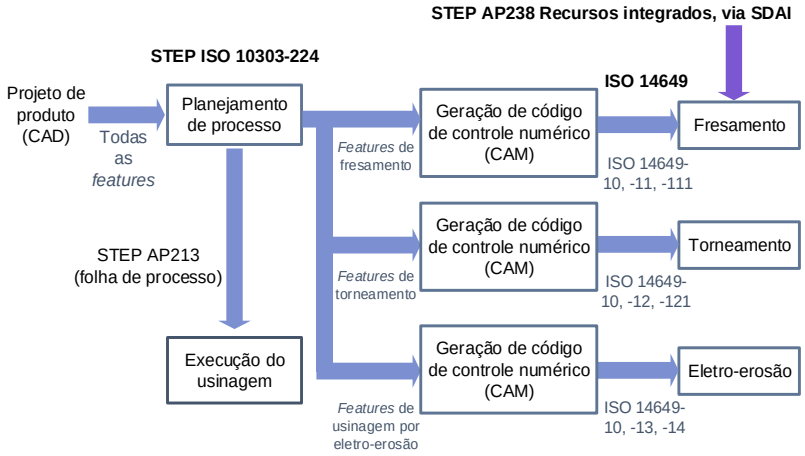


Figura 2.1. Ciclo de manufatura com o uso do padrão STEP-NC.
FONTE: Traduzido de ISO 14649-1 (2002)

2.1.1 Características e benefícios do padrão STEP-NC

Apesar de ser considerado o substituto do código G, o padrão STEP-NC possui características diferentes das do seu antecessor, as quais o tornam atrativo para seu uso na indústria e na academia, como referem Hardwick et al. (2013). Diferente da norma ISO 6983, a qual descreve explicitamente a trajetória da ferramenta de usinagem, o padrão STEP-NC descreve o processo de usinagem. Assim, o controlador da máquina ferramenta deve interpretar as informações do processo e transformá-las em movimentos da ferramenta e funções da máquina ferramenta CNC, imediatamente antes da usinagem, com base nas informações do arquivo STEP-NC (Sääski; Salonen; Paro, 2005; Pobozniak, 2013).

Além disso, o padrão STEP-NC permite que as informações possam ser utilizadas em qualquer máquina, tornando-o independente de

software e hardware proprietário, o que permite maximizar a eficiência da produção ao eliminar a necessidade de converter dados durante o processo de fabricação.

Como afirmam Newman; Allen; Rosso (2003), o padrão STEP-NC apresenta as informações em um modelo orientado a objetos e com dados estruturados e detalhados, o que faz com que seja mais compreensível para o usuário, facilitando a inserção de mudanças. Além disso, como as informações transmitidas são padronizadas e não dependem de uma linguagem ou códigos específicos do fabricante, elas podem ser codificadas mediante um arquivo neutro, como estabelecido na norma ISO 10303-21 (2002), auxiliando no compartilhamento de informações, como mostrado na Figura 2.2.

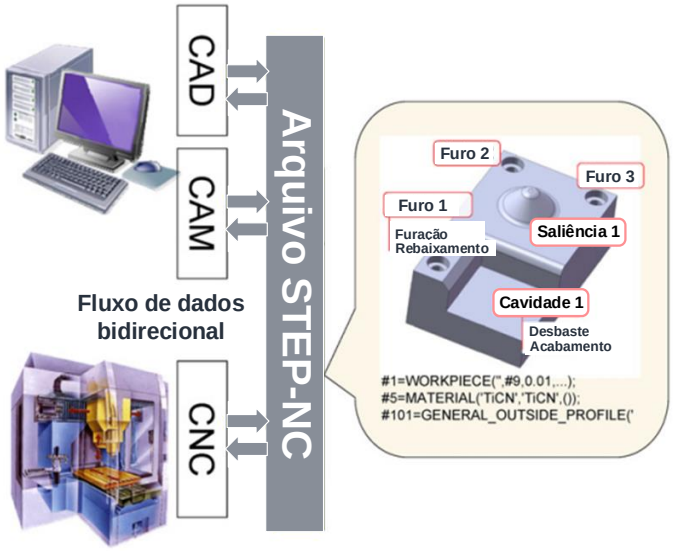


Figura 2.2. STEP-NC no compartilhamento de informações
FONTE: Traduzido de Rauch et al. (2012)

No que se refere aos benefícios da adoção do padrão STEP-NC, Xu; Newman (2006) indicam como principais:

- O modelo conecta as informações geométricas e tecnológicas, evitando a perda de informações durante o desenvolvimento do produto;
- Seus elementos são suficientes para descrever dados orientados a tarefas de controle numérico;

- O modelo é escalável e extensível a outras tecnologias;
- Elimina a necessidade de pós-processadores;
- Possibilita um fluxo de informações bidirecional entre o sistema CAD/CAM e as máquinas CNC, assim como entre os projetistas e o chão de fábrica.

Por último, Hardwick et al. (2013) sugerem que a adoção do padrão STEP-NC pode reduzir o tempo de programação da máquina CNC em 35% e os tempos de usinagem em até 50%, além de diminuir em 75% a quantidade de desenhos durante o desenvolvimento do produto.

2.1.2 Conceitos fundamentais do padrão STEP-NC

O padrão STEP-NC foi concebido como um padrão para a troca de informações de fabricação, pelo qual abrange uma quantidade significativa de informações. Por este motivo, a norma ISO 14649 foi dividida em várias partes envolvendo temas específicos, desde princípios fundamentais (parte 1) e dados gerais do processo (parte 10), até ferramentas e tecnologia próprias de um processo de fabricação determinado, como no caso da parte 11 (dados gerais para fresamento) e parte 12 (dados de processo para torneamento). Cada uma destas partes define conceitos importantes da sua área, os quais estabelecem elementos da nova estrutura de representação de dados de processo.

Neste trabalho será dada ênfase aos conceitos relacionados somente ao fresamento.

Antes de entrar nas definições próprias da norma ISO 14649 é preciso definir alguns termos contidos em normas anteriores ou relacionadas, que também foram utilizadas neste trabalho. Estes termos são:

- Arquivo físico – refere-se ao formato do arquivo descrito na norma ISO 10303-21 (2002) para a troca de dados;
- Entidade (em inglês: *entity*) - uma classe de informação definida por propriedades comuns (ISO 10303-11, 2004. Tradução nossa);
- *Feature* – volume de material a ser removido da peça mediante usinagem ou que é resultado da usinagem (ISO 10303-224, 2006. Tradução nossa);
- *Manufacturing feature* – "conjunto de informações geométricas e não geométricas importantes do ponto de vista do planejamento do processo de manufatura" (Pobożniak, 2013. p.50. Tradução nossa);

- Linguagem EXPRESS - linguagem de especificação de dados, constituída por elementos de linguagem que permitem uma definição inequívoca de dados e especificação de restrições sobre os dados definidos (ISO 10303-11, 2004. Tradução nossa);
- Trajetória da ferramenta - Caminho descrito por um ponto específico de uma ferramenta de corte (ISO 2806, 1994. Tradução nossa).

Os termos utilizados dentro do escopo norma ISO 14649 foram:

- *Workingstep* - Informação de usinagem para uma ferramenta de corte atuando em uma *feature* (ISO 14649-1, 2002. Tradução nossa);
- Operação de usinagem (em inglês: *machining operation*) - dados tecnológicos para um *workingstep* que detalha a operação (ISO 14649-1, 2002. Tradução nossa);
- Plano de trabalho (em inglês: *Workplan*) - coleção de *workingsteps* com uma sequência de execução (ISO 14649-1, 2002. Tradução nossa);
- Projeto (em inglês: *project*) – entidade que serve como ponto de partida para a execução do programa (ISO 14649-1, 2002. Tradução nossa);
- Usinagem 2½ D – usinagem de uma peça prismática, geralmente feita em camadas perpendiculares ao eixo da ferramenta (ISO 14649-10, 2002. Tradução nossa).

2.1.3 Modelo de dados do padrão STEP-NC

Como já mencionado, o padrão STEP-NC apresenta um modelo estruturado de dados orientado a objetos que define diversas entidades, bem como atributos que descrevem a peça e seus processos de usinagem.

A estrutura começa com a entidade *Project* que contém dados gerais do produto e seu projeto; um dos atributos dessa entidade é o *workplan*. O *workplan* contém uma lista de entidades, cujos elementos podem ser *workingsteps*, funções da máquina CNC ou estruturas de programação, os quais, ao serem executados sequencialmente, resultam no produto final. No caso dos *workingsteps* estes contêm a informação geométrica da *feature* e a informação da operação de usinagem.

Em relação às *features*, a norma ISO 14649-10 (2002) define três tipos: regiões, *features* 2½D e *features* de transição. Cada categoria possui um conjunto de entidades com as quais pode-se descrever

completamente a geometria do produto. Por exemplo, dentro das *features* 2½D estão definidas as seguintes entidades: cavidade, ranhura, furo, saliência, degrau e face plana. A entidade correspondente a cada *feature* contém informações da peça à qual ela pertence, a sua localização com referência ao sistema de coordenadas definido na peça, e as características geométricas próprias com referência ao sistema de coordenadas estabelecido na mesma *feature*, como é mostrado na Figura 2.3 para o caso de uma face plana.

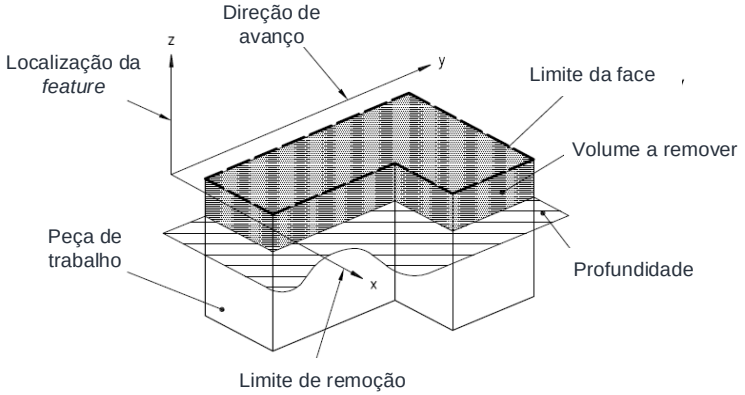


Figura 2.3. Atributos de uma face plana segundo a norma ISO 14649-10.
FONTE: Traduzido de ISO 14649-10 (2002)

As operações de usinagem também são definidas por entidades na parte correspondente da norma, cujos principais atributos são: o tipo de operação e a estratégia de usinagem, a ferramenta a ser utilizada, as funções auxiliares da máquina CNC executadas paralelamente e seus parâmetros tecnológicos, tais como velocidade de corte e avanço, entre outros.

Para o caso do processo de fresamento, a norma ISO 14649-11 (2002) diferencia dois tipos de processo: furação e fresamento propriamente dito. Dentro deste último distingue-se: faceamento, fresamento lateral e fresamento da base, os quais podem ser executados mediante as seguintes estratégias: paralelo ao contorno, unidirecional, bidirecional, espiral, contorno-bidirecional e bidirecional-contorno. No caso do fresamento, a norma ISO 14649-111 (2002) contém as entidades que descrevem totalmente a ferramenta de corte, tanto o tipo de ferramenta como suas dimensões principais.

Essa estrutura permite a descrição integral da peça e do seu processo de fabricação totalmente baseada em entidades, o que torna o padrão STEP-NC uma linguagem genérica e de alto nível, como afirmam Rauch et al. (2012).

2.1.4 Arquivo físico STEP-NC

A estrutura de dados apresentada na seção anterior é descrita mediante a linguagem EXPRESS, como estabelecido na norma ISO 10303-11 (2004) e codificada no arquivo físico padronizado na parte 21 da mesma norma. O arquivo resultante é similar ao mostrado na Figura 2.4.

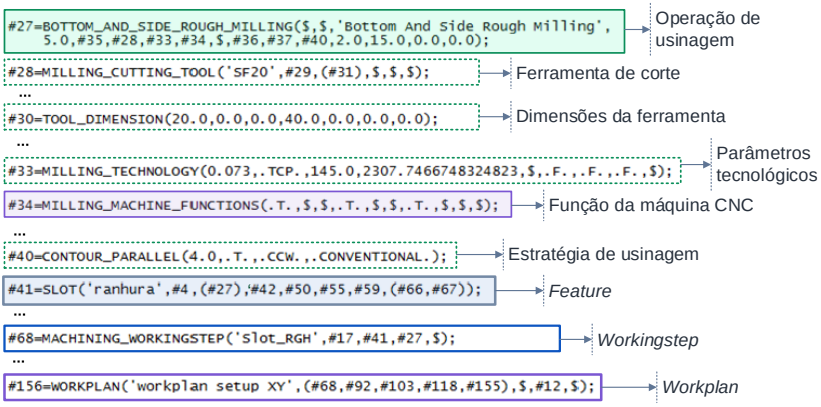


Figura 2.4. Exemplo do conteúdo do arquivo físico STEP-NC.
FONTE: Adaptado de Benavente; Ferreira (2013)

O arquivo físico STEP-NC mostra as informações em função da peça, porém não possui informações da trajetória que a ferramenta de corte deve percorrer para a sua usinagem. A tarefa de geração da trajetória é reservada ao controlador da máquina CNC, tornando o controlador uma parte central do processo de fabricação, com todos os desafios que isso implica. Contudo, os esforços na aplicação do STEP-NC ainda estão focados na implantação do modelo para determinados tipos de controladores CNC ou máquinas ferramentas específicas, fazendo com que todo o potencial do padrão não seja explorado, e equipamentos especializados são difíceis de serem integrados na manufatura, como ocorre no caso dos robôs industriais, sendo esta uma parte da motivação deste trabalho (Rauch et al., 2012; Kassim et al., 2015)

2.1.5 Tipos de STEP-NC

Com o surgimento do padrão STEP-NC, uma etapa de transição é necessária para garantir a aderência dos sistemas CAD/CAM e as máquinas CNC ao novo padrão. Sendo o controlador da máquina CNC um elemento essencial na fabricação do produto, Suh et al. (2003) diferenciam três tipos de controladores segundo o estágio de implementação do padrão, resumidos na Figura 2.5.

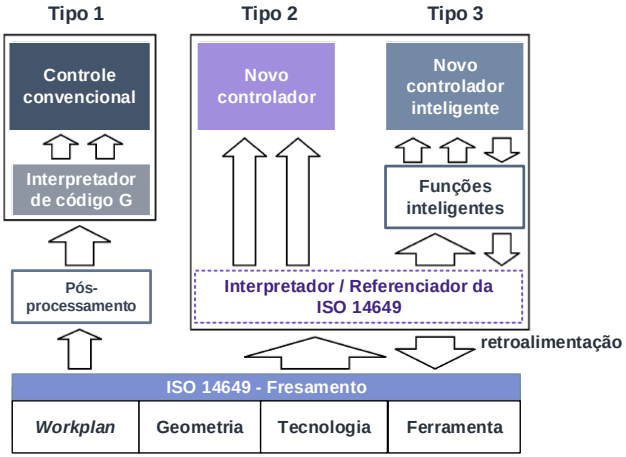


Figura 2.5. Tipos de STEP-NC.
FONTE: Traduzido de Suh et al. (2003)

Na Figura 2.5, o tipo 1 requer um pós-processador prévio que facilite a tradução do STEP-NC para código G, mantendo o fluxo de informações convencional, que é unidirecional. Já os tipos 2 e 3 consideram uma comunicação bidirecional e utilizam um interpretador de STEP-NC. No tipo 2 essa informação é transmitida diretamente para o novo controlador, o qual é capaz unicamente de gerar a trajetória da ferramenta segundo as informações de entrada, enquanto no tipo 3 o controlador inclui funções que permitem uma adequação e aprimoramento do processo de usinagem conforme as condições do momento. Rauch et al. (2012) denominam o tipo 3 como STEP-NC adaptativo, pois pode otimizar os parâmetros e as trajetórias de usinagem de acordo com as informações do processo adquiridas em tempo real, e o situam como o cenário alvo da implementação do padrão STEP-NC.

As categorias apresentadas permitem a aderência ao padrão STEP-NC tanto das máquinas CNC antigas, mediante o controlador tipo 1, das

máquinas CNC produzidas atualmente mediante o controlador tipo 2, e no projeto das novas máquinas CNC com um controlador tipo 3, o que facilita a adoção deste padrão pela indústria, como afirmam Cha et al. (2014).

No escopo deste trabalho, o método desenvolvido procura se enquadrar dentro do contexto do controlador tipo 2, porém orientado a manipuladores robóticos de tipo industrial, os quais serão abordados na próxima seção.

2.2 Conceitos e ferramentas da robótica

Na seção anterior foram apresentados os princípios fundamentais do padrão STEP-NC. Nesta seção serão apresentados os principais conceitos e ferramentas da área da robótica necessárias para a aplicação desse padrão nos robôs industriais.

2.2.1 Generalidades dos manipuladores robóticos

Segundo a norma ISO 8373 (2012), um robô industrial é definido como um manipulador automaticamente controlado, reprogramável, multiuso, que pode ser programado em três ou mais eixos, podendo ser fixo ou móvel, usado em aplicações industriais. Neste contexto, a robótica, como uma ciência que trata do projeto, fabricação e aplicação de robôs, é a responsável pelo desenvolvimento de ferramentas matemáticas que facilitem a análise de um robô e a factibilidade de sua aplicação para uma tarefa específica, com base nos seus parâmetros cinemáticos e dinâmicos (ISO 8373, 2012).

Do ponto de vista estrutural, Tsai (1999) define um manipulador como um conjunto de elos conectados por elementos denominados juntas (as quais podem ser ativas ou passivas). Um dos elos é fixo, e outro corresponde ao efetuador final[1] ou elo de saída, que executa a ação sobre um elemento do seu ambiente.

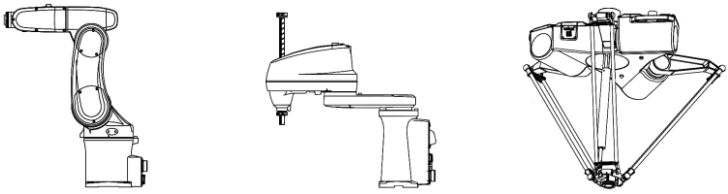
Com relação às juntas utilizadas para a conexão dos elos, estas podem ser: prismáticas ou rotativas (um grau de liberdade cada), cilíndricas ou planares (dois graus de liberdade), esféricas (três graus de liberdade), entre outras. Geralmente, o número de graus de liberdade

[1] Efetuador final *(end effector)*: termo utilizado na robótica para se referir aos dispositivos conectados ao elo de saída de um manipulador mecânico. Segundo Tsai (1999), o efetuador final pode-se considerar como a interface entre um manipulador e seu entorno.

(GDL) do manipulador corresponde ao número de juntas atuadas de um grau de liberdade presentes no robô. Não obstante, isso depende do tipo de cadeia cinemática presente no manipulador e do grau de redundância do robô. Segundo Siciliano (1990), a redundância cinemática ocorre quando um manipulador possui mais graus de liberdade do que os requeridos para executar uma tarefa específica. Segundo o tipo de cadeia cinemática, os robôs podem ser classificados em três grupos: (1) manipuladores seriais, (2) manipuladores paralelos, e (3) manipuladores híbridos.

Os manipuladores seriais são os mais conhecidos na indústria e são compostos por uma cadeia cinemática aberta, isto é, os dois extremos da cadeia estão unidos por uma sequência única e consecutiva de elos, como afirma Siciliano et al. (2010). Esta configuração permite que o efetuador final possa cobrir um grande volume de trabalho. Dependendo do arranjo das juntas na cadeia cinemática, Spong; Hutchinson; Vidyasagar (2005) diferenciam alguns subtipos de manipuladores seriais:

- Antropomórficos: compostos unicamente por juntas rotativas, das quais três delas possuem eixos paralelos (Figura 2.6a);
- Esféricos: formados por duas juntas rotativas e uma junta prismática, cujos eixos são perpendiculares entre si;
- SCARA: também possuem duas juntas rotativas e uma junta prismática, porém seus eixos são paralelos entre si (Figura 2.6b);
- Cilíndricos: compostos por uma junta rotativa e duas juntas prismáticas; as variáveis das juntas descrevem o deslocamento do efetuador final em relação à base mediante coordenadas cilíndricas;
- Cartesianos: integrados por três juntas prismáticas com eixos perpendiculares entre si.



(a) (b) (c)

Figura 2.6. Tipos de robôs industriais: (a) antropomórfico, (b) SCARA, (c) delta
FONTE: Adept Technology Inc. (2013)

Em contrapartida, os manipuladores paralelos são formados por cadeias cinemáticas fechadas, nas quais o efetuador final está ligado à base mediante várias cadeias cinemáticas, o que proporciona vantagens como maior estabilidade, melhor distribuição de carga e redução do número de atuadores requeridos (Merlet, 2006). Os robôs mais conhecidos desta categoria são os robôs tipo delta, mostrado na Figura 2.6c, mas Tsai (1999) os classifica em manipuladores planares, esféricos e espaciais.

No caso dos manipuladores híbridos, Tanev (2000) os define como uma combinação de um manipulador serial e um paralelo, ou uma série de manipuladores paralelos, os quais apresentam as vantagens de ambos os tipos de manipuladores, tanto no volume de trabalho, quanto na exatidão e capacidade de carga. Por esses motivos, os manipuladores híbridos surgiram como uma opção para os centros de usinagem CNC convencionais (Harib et al., 2012).

Devido às grandes capacidades dos robôs e as vantagens da sua utilização em ambientes perigosos para os humanos, estes têm sido inseridos paulatinamente nas principais indústrias, atingindo níveis recorde no ano de 2014, segundo as estatísticas da Federação Internacional da Robótica (em inglês: *IFR - International Federation of Robotics*), mostradas na Figura 2.7. As tarefas em que são utilizadas nas linhas de produção vão desde montagem, pintura, soldagem, usinagem, inspeção, transporte, embalagem até manipulação de materiais, o que mostra os benefícios que eles têm trazido em termos de segurança para os trabalhadores e produtividade para as empresas.

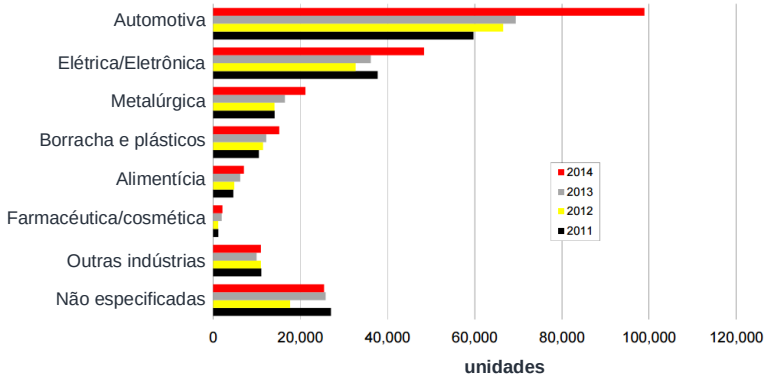


Figura 2.7 Fornecimento anual mundial de robôs industriais por principais indústrias 2011-2014.
FONTE: International Federation of Robotics (2015)

2.2.2 Componentes de um robô industrial

Segundo a norma ISO 8373 (2012), o termo robô industrial compreende tanto o manipulador como seu controlador, o qual inclui a interface de programação. O termo manipulador se refere à parte mecânica, encarregada de transformar os torques aplicados pelos atuadores em um movimento apropriado, enquanto o controlador interpreta as instruções dadas ao robô e gera os valores das variáveis de juntas e velocidades adequadas para transmitir aos atuadores (Horsch, 2000).

No que se refere ao controlador, Patrick; Fardo (2000) identificam três níveis de controle incluídos no controlador de um robô, resumidos na Figura 2.8.O nível mais baixo lida diretamente com os atuadores e controla o movimento do manipulador em cada eixo. O nível intermediário coordena o movimento dos atuadores do nível I com a trajetória. O nível mais alto de controle processa as instruções dadas na linguagem específica do robô (ou mediante a interface de programação) e as traduz para comandos e trajetórias que o nível II seja capaz de interpretar.

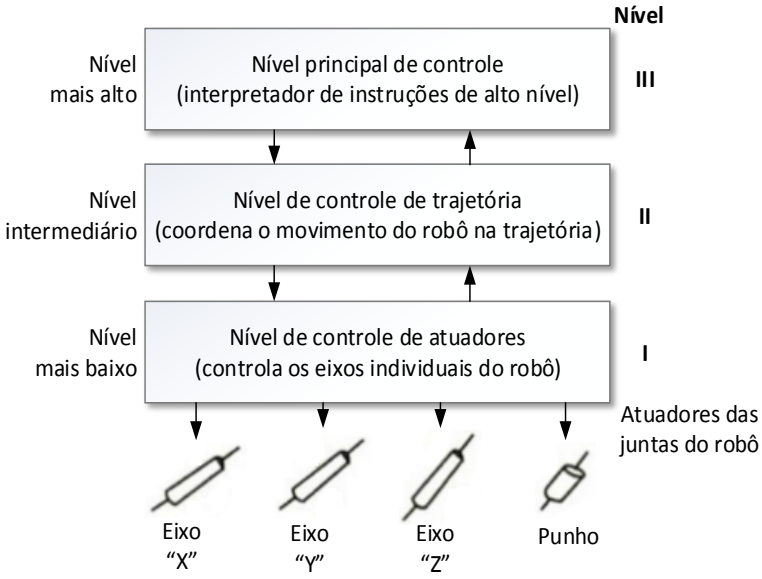


Figura 2.8 Níveis básicos de controle hierárquico
FONTE: Patrick; Fardo (2000)

O problema de planejamento de trajetórias é de grande importância para a robótica industrial, razão pela qual vários autores têm sugerido diferentes abordagens aplicáveis ao nível II de controle (Lin, Chang, Luh, 1983). Uma abordagem pode ser direta, resolvendo o planejamento da trajetória diretamente no espaço dimensional. Outra abordagem é a desacoplada, a qual resolve o planejamento de trajetórias em duas etapas: a primeira define uma trajetória geométrica no espaço cartesiano baseado na tarefa, no entorno de trabalho e obstáculos; enquanto uma segunda etapa resolve a trajetória anterior considerando aspectos dinâmicos do robô no espaço de juntas (Verscheure et al., 2009).

Segundo Ghazaei (2015) e Verscheure et al. (2009), a estratégia desacoplada é preferida por ter uma complexidade e requerimentos computacionais menores na resolução do problema de planejamento de trajetórias, pelo qual foi considerada no desenvolvimento deste trabalho.

2.2.3 Estratégias para programação de robôs

A integração dos robôs nas linhas de produção tem sido possível graças à sua capacidade para executar e repetir tarefas pré-programadas. Segundo Ang; Wei; Yong (2000), na indústria existem duas formas principais de programar um robô: ensinando-o ativamente (programação *online*), ou mediante a utilização de ferramentas computacionais especializadas ou linguagens de programação (programação *off-line*).

Embora a programação *online* requeira menor quantidade de recursos e de conhecimento especializado, ela depende diretamente das habilidades do operador, o que faz com que a qualidade do trabalho fique limitada, além de dificultar a modificação da tarefa programada, e geralmente o tempo de programação é muito mais longo. Para o caso da programação *off-line*, a programação é transferida a um ambiente virtual sem interferir na linha de produção, o que torna o processo mais flexível, factível de aplicar em tarefas mais complexas e mais economicamente eficiente para produção de grandes volumes (Pan et al., 2012)

No âmbito da programação *off-line* o software utilizado para simular adequadamente o comportamento do robô em um ambiente virtual é extremamente importante. Essas soluções de software comumente utilizam o que Deng et al. (2012) denominam de programação gráfica, que consiste em criar as trajetórias baseadas no modelo CAD do produto desejado e dos elementos do ambiente (Figura 2.9). Além disso, algumas soluções possuem a capacidade de transmitir o processo diretamente para o controlador do robô (Qi et al., 2008), como o

RobotStudio do fabricante ABB, KukaSim da Kuka e RoboGuide do Fanuc.

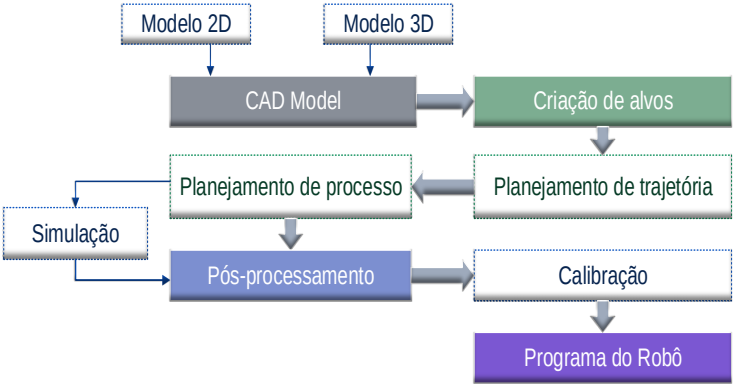


Figura 2.9. Passos da programação *off-line.*
FONTE: Adaptado de Pan et al. (2012)

A maioria dos sistemas computacionais para programação *off-line* relaciona-se com um fabricante específico, porém, existem outras soluções de software denominados genéricos, capazes de simular e gerar programas para robôs de diversos fabricantes, os quais utilizam pós-processadores. O pós-processador é uma aplicação que permite a escrita de dados na linguagem nativa do robô, desta maneira a tarefa pode ser carregada no controlador e executada com interação mínima do usuário (Pan et al., 2012). A utilização de um software genérico constitui uma solução vantajosa e eficiente para integrar robôs de diversos tipos e marcas no chão de fábrica, a qual foi uma opção explorada no desenvolvimento deste trabalho.

2.2.4 Análise cinemática dos manipuladores

Para que um manipulador robótico execute uma tarefa ele precisa se posicionar em pontos determinados com uma postura que não interfira com outros elementos no ambiente de trabalho. A capacidade de um robô de atingir um ponto com uma configuração específica em seus elos e juntas depende diretamente dos aspectos mecânicos que afetam o movimento do manipulador, os quais são referidos ao longo deste trabalho como parâmetros cinemáticos. A análise destes parâmetros cinemáticos é um requisito fundamental da robótica para o qual têm sido

desenvolvidas algumas ferramentas matemáticas que permitem estudar a cinemática dos manipuladores de diferentes tipos, descritas nesta seção.

2.2.4.1 Deslocamento rígido e transformações homogêneas

A localização de um objeto no espaço é determinada por uma posição e uma orientação com referência à origem de um sistema de coordenadas. Para a representação matemática da posição, geralmente é usado um vetor coluna de três componentes, enquanto para a representação da orientação é usada uma matriz de rotação que representa a orientação do objeto com respeito ao sistema de coordenadas.

Apesar de existirem várias formas de representação da rotação, no escopo deste trabalho serão usadas a representação por ângulos *yaw, pitch e roll* (ψ-θ-φ) e a representação ângulo-eixo (α, u).

A representação por ângulos *yaw, pitch e roll* utiliza uma série de rotações sucessivas para descrever a orientação do objeto. Como observado na Figura 2.10, a primeira rotação é ao redor do eixo X por um ângulo ψ, depois é executada uma segunda rotação ao redor do eixo Y um ângulo θ, e finalmente uma rotação de um ângulo φ ao redor do eixo Z (Spong et al., 2005).

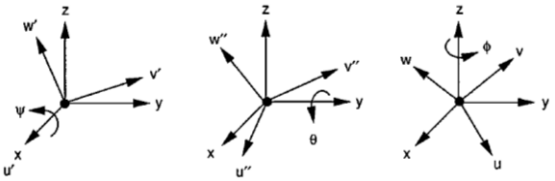


Figura 2.10. Rotações sucessivas da representação *yaw, pitch e roll.*
FONTE: Tsai (1999)

Essa sucessão de rotações pode ser descrita por uma matriz que agrupa as três rotações básicas ao redor dos eixos do sistema coordenado, como mostrado na equação (1) (Tsai, 1999), onde $c\theta$ representa $\cos\theta$, e $s\theta$ representa $\operatorname{sen}\theta$.

$$R_E^O = \begin{bmatrix} \mathrm{c}\,\phi\,\mathrm{c}\,\theta & -\mathrm{s}\,\phi\,\mathrm{c}\,\psi + \mathrm{c}\,\phi\,\mathrm{s}\,\theta\,\mathrm{s}\,\psi & \mathrm{s}\,\phi\,\mathrm{s}\,\psi + \mathrm{c}\,\phi\,\mathrm{s}\,\theta\,\mathrm{c}\,\psi \\ \mathrm{s}\,\phi\,\mathrm{c}\,\theta & \mathrm{c}\,\phi\,\mathrm{c}\,\psi + \mathrm{s}\,\phi\,\mathrm{s}\,\theta\,\mathrm{s}\,\psi & -\mathrm{c}\,\theta\,\mathrm{s}\,\psi + \mathrm{s}\,\phi\,\mathrm{s}\,\theta\,\mathrm{c}\,\psi \\ -\mathrm{s}\,\theta & \mathrm{c}\,\theta\,\mathrm{s}\,\psi & \mathrm{c}\,\theta\,\mathrm{c}\,\psi \end{bmatrix} \quad (1)$$

Enquanto a representação por ângulos *yaw, pitch e roll* utiliza somente três parâmetros, a representação ângulo-eixo precisa de quatro,

mas a rotação pode ser visualizada com maior facilidade no caso de aplicações aeronáuticas (Figura 2.11). O vetor *u* possui a orientação determinada pelos ângulos σ e ρ, com referência aos eixos do sistema de coordenadas, e α é o ângulo de rotação ao redor deste vetor em sentido anti-horário.

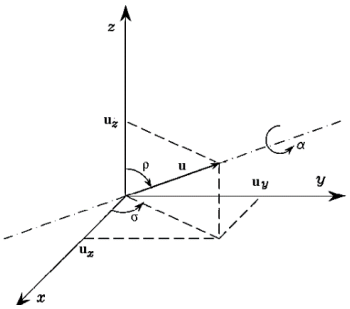


Figura 2.11. Representação ângulo-eixo.
FONTE: Siciliano et al. (2010)

A descrição em forma matricial da representação ângulo-eixo é expressa na equação (2), onde γ representa $(1-\cos\alpha)$.

$$R(\alpha,u)=\begin{bmatrix} u_x^2\gamma+c\alpha & u_xu_y\gamma-r_zs\alpha & u_xu_z\gamma+u_ys\alpha \\ u_xu_y\gamma+u_zs\alpha & u_y^2\gamma+c\alpha & u_yu_z\gamma-u_xs\alpha \\ u_xu_z\gamma-u_ys\alpha & u_yu_z\gamma+u_xs\alpha & u_z^2\gamma+c\alpha \end{bmatrix} \quad (2)$$

Essa representação é utilizada geralmente como passo intermediário para obter outros tipos de representação de rotação, como mencionado por Shuster (1993).

A forma de resolver o deslocamento de um objeto tanto em posição como em orientação é mediante matrizes de transformação homogêneas. Na matriz de transformação homogênea (4x4) podem ser identificados quatro componentes: a matriz de rotação (3x3) e o vetor de posição (3x1) na parte superior da matriz, e o vetor transformação de perspectiva (1x3), com seus elementos fixados em zero, e um fator de escala (1x1), fixo em 1, na parte inferior da matriz (Tsai, 1999).

Estas matrizes também são usadas para calcular as coordenadas de um ponto depois de uma mudança de sistema de referência. Por exemplo, se um ponto *p*, cujas coordenadas têm como referência a origem do sistema de coordenadas *A*, requeresse conhecer suas coordenadas em

relação a um sistema de coordenadas *C*, pode-se aplicar a equação (3), onde T_A^B descreve a localização de *A* com respeito a *B* e T_B^C a localização de *B* com respeito a *C*.

$$\begin{bmatrix} p^C \\ 1 \end{bmatrix} = T_B^C \cdot T_A^B \cdot \begin{bmatrix} p^A \\ 1 \end{bmatrix} \quad (3)$$

A multiplicação destas duas matrizes resulta na conversão das coordenadas do sistema de coordenadas *A* até o sistema *C*.

Embora a forma mais conhecida de representação de rotações e deslocamento rígido são as matrizes de transformação, existe uma outra abordagem que tem sido utilizada na robótica nas últimas décadas, a qual é o uso de quatérnios e quatérnios duais (Radavelli et al., 2015). Segundo Mukundan (2002), os quatérnios são números hipercomplexos de dimensão 4, geralmente usados na representação de rotações no espaço tridimensional. O autor assinala como vantagens do uso dos quatérnios na representação de rotações: (a) a representação com número reduzido de elementos; e, (b) o menor custo computacional decorrente de seu uso. No caso de deslocamento rígido podem ser usados os quatérnios duais, como mencionado por Kenwright (2012), os quais podem ser vistos como uma combinação da teoria de números duais e os quatérnios. Estes quatérnios duais podem representar uma translação e rotação mediante somente 8 elementos, sendo uma representação compacta similar aos quatérnios, como mostrado na Figura 2.12. Informação mais detalhada sobre o uso de quatérnios e quatérnios duais pode ser conferida nos trabalhos de Dooley; McCarthy (1991), Heidari; Daniali; Varedi (2014), Marinho (2014) e Radavelli et al. (2015).

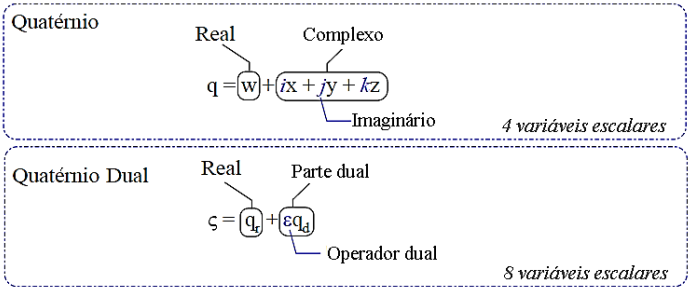


Figura 2.12. Visão geral dos componentes dos quatérnios e quatérnios duais.
FONTE: Kenwright (2012)

Todas as expressões apresentadas facilitam a representação da localização de um objeto, ou no caso dos manipuladores, do efetuador final, além de fornecer a sua posição e orientação com respeito a vários sistemas de coordenadas.

2.2.4.2 Cinemática direta

Como um robô é composto por um conjunto de elos interconectados por várias juntas, as quais podem ter diferentes níveis de complexidade, Spong; Hutchinson; Vidyasagar (2005) afirmam que o problema da cinemática direta reside em determinar a posição e orientação do efetuador final do robô em termos das suas variáveis de juntas. O problema de determinar o valor das juntas a partir de uma dada posição (x,y,z) é chamado de cinemática inversa.

Para a elaboração de soluções para os problemas de cinemática de manipuladores têm sido criadas convenções e algoritmos, sendo uma das mais importantes a convenção Denavit-Hartenberg (Hartenberg; Denavit, 1955). Segundo Spong; Hutchinson; Vidyasagar (2005), dado que o principal objetivo da análise da cinemática direta é determinar os efeitos acumulativos do conjunto de variáveis de juntas sobre a posição de qualquer elo, é necessário estabelecer sistemas de referência rigidamente conectados a cada um dos elos. Desta maneira, atuando-se sobre uma junta *i*, tanto o elo *i* como o seu sistema de referência experimentam o movimento resultante, mas o sistema de referência fixado na base do robô não experimenta movimento nenhum.

Logo, a posição de qualquer ponto do elo *n* expresso no sistema de referência preso ao próprio elo *n* é independente da configuração do robô, mas a posição de qualquer ponto do elo *n* expresso no sistema de referência da base depende da posição de cada um dos outros elos que compõem o robô. A posição desses elos, por sua vez, depende dos movimentos feitos pelas juntas que ligam cada um dos elos. Esta posição pode ser expressa em uma matriz de transformação homogênea mostrada na Equação (4), onde R_n^0 é a matriz de rotação de *n* com referência no elo 0 ou base, e O_n^0 é o vetor posição da origem do sistema de referência do elo *n*.

$$T_n^0 = \begin{bmatrix} R_n^0 & O_n^0 \\ 0 & 1 \end{bmatrix} \tag{4}$$

Nesse contexto, a convenção Denavit-Hartemberg é um método para escolher o sistema de referência de cada elo, e oferece uma forma de

representação da matriz de transformação homogênea baseada em quatro parâmetros: a_{i_DH}, comprimento do elo i; α_{i_DH}, torção do elo i; d_{i_DH}, *offset*; e, θ_{i_DH}, ângulo da junta i, como mostrado na Equação (5).

$$A_i = \begin{bmatrix} c\theta_{i\_DH} & -s\theta_{i\_DH}c\alpha_{i\_DH} & s\theta_{i\_DH}s\alpha_{i\_DH} & a_{i\_DH}c\theta_{i\_DH} \\ s\theta_{i\_DH} & c\theta_{i\_DH}c\alpha_{i\_DH} & -c\theta_{i\_DH}s\alpha_{i\_DH} & a_{i\_DH}s\theta_{i\_DH} \\ 0 & s\alpha_{i\_DH} & c\alpha_{i\_DH} & d_{i\_DH} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5)$$

Para estabelecer os sistemas de referência de cada elo segundo esta convenção, é preciso cumprir os passos descritos a seguir:

1. O eixo *Z* encontra-se na direção do eixo da junta *i*.
2. A origem do sistema de referência O_i localiza-se na intersecção entre o eixo Z_i e o eixo Z_{i-1}.
3. O eixo *Xi* localiza-se na direção da normal comum entre Z_{i-1} e Z_i.

Cada um desses parâmetros associados ao elo *i* é determinado após estabelecer o sistema de referência, como mostrado na Figura 2.13.

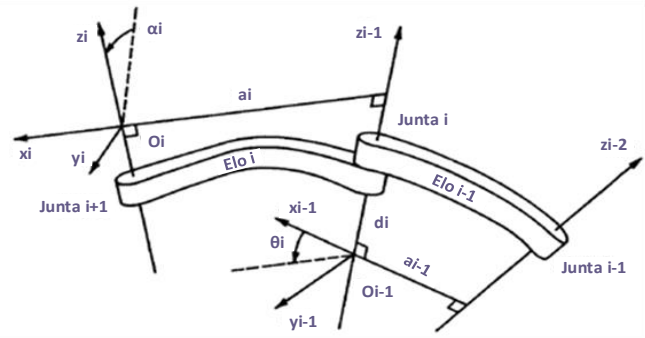


Figura 2.13. Convenção Denavit-Hartenberg.
FONTE: Traduzido de Spong; Hutchinson; Vidyasagar (2005)

Três dos parâmetros antes mencionados são fixos e um é variável, pelo qual a matriz A_i é função de uma variável só, que depende do tipo de junta: θ_{i_DH} se a junta for rotativa e d_{i_DH} se for uma junta prismática (Rocha et al., 2011). Para conhecer a posição de um elo *i* com referência ao outro elo *j*, deve-se multiplicar as matrizes obtidas correspondentes aos elos que estão entre eles na cadeia cinemática, começando pela matriz do

elo *i* multiplicada pela matriz do seu elo adjacente, e assim por diante até chegar ao elo *j*.

Cabe mencionar que a convenção Denavit-Hartenberg é aplicada geralmente para a solução da cinemática direta de manipuladores seriais. A análise de manipuladores paralelos mediante esta convenção também é possível com pequenas modificações, porém é mais complexa devido à morfologia destes manipuladores: quanto maior o número de cadeias fechadas, maior a complexidade (Siciliano et al., 2010).

Para o estudo da cinemática direta de manipuladores paralelos é preferível uma abordagem geométrica, como a apresentada por Merlet (2006), onde a estrutura do manipulador é analisada mediante relações trigonométricas entre seus elos de maneira a estabelecer expressões matemáticas que permitam determinar a posição do efetuador final, específicas para um robô específico.

2.2.4.3 Cinemática inversa

Como já mencionado, a solução de cinemática inversa consiste em determinar o valor das juntas a partir da posição do efetuador final do manipulador, correspondendo assim ao cenário oposto da cinemática direta.

Novamente, a análise geométrica é o caminho comumente adotado para obter uma solução de acordo com a morfologia do manipulador, mas no caso dos manipuladores seriais por exemplo, esta solução pode não ser única ou pode resultar em infinitas soluções, o que faz com que sejam utilizadas técnicas de solução numéricas aplicáveis a qualquer estrutura cinemática, como confirmado por Siciliano et al. (2010).

Os métodos numéricos utilizados para a solução da cinemática inversa normalmente requerem a utilização do jacobiano correspondente ao manipulador. Segundo Tsai (1999), o jacobiano *J* é uma matriz de transformação linear que mapeia a velocidade em função das variáveis das juntas em um vetor velocidade que contém a velocidade angular e linear do manipulador, como mostrado na equação (6)

$$\begin{bmatrix} \dot{p} \\ \omega \end{bmatrix} = J \begin{bmatrix} \dot{q_1} \\ \dot{q_2} \\ \vdots \\ \dot{q_n} \end{bmatrix} \tag{6}$$

Onde $\dot{p}$ corresponde à velocidade linear, ω é a velocidade angular, e $\dot{q_n}$ são as derivadas parciais das variáveis das juntas *i*.

Para o cálculo do jacobiano geométrico é utilizado o método proposto por Spong; Hutchinson; Vidyasagar (2005), que diferencia os aportes de cada junta, J_i, nas componentes da velocidade linear e angular, obtendo-se a equação (7) para as juntas rotativas e a equação (8) para as juntas de tipo prismáticas. Para conferir os parâmetros usados, deve-se referir à Figura 2.14.

$$J_i = \begin{bmatrix} z_{i-1} \times (o_n - o_{i-1}) \\ z_{i-1} \end{bmatrix} \tag{7}$$

$$J_i = \begin{bmatrix} z_{i-1} \\ 0 \end{bmatrix} \tag{8}$$

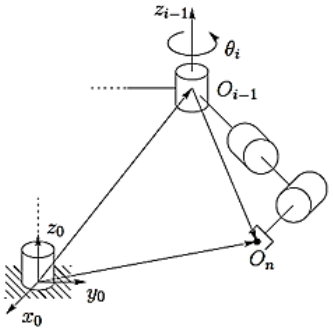


Figura 2.14 Parâmetros para determinação do jacobiano
FONTE: Spong; Hutchinson; Vidyasagar (2005)

Os parâmetros z_i correspondem aos três primeiros elementos da terceira coluna da matriz A_i^o, obtida do método Denavit-Hartenberg, e o_i corresponde aos três primeiros elementos da quarta coluna da mesma matriz. Calculando-se as componentes do jacobiano para cada junta obtém-se o jacobiano do manipulador.

Um fator importante relacionado com o jacobiano do manipulador são as singularidades. Siciliano et al. (2010) descrevem dois tipos de singularidades: (a) internas e (b) externas. As singularidades externas ou singularidades de borda ocorrem quando um manipulador está no limite do espaço de trabalho ou atinge o limite mecânico de movimento de uma junta. No caso das singularidades internas, estas ocorrem dentro do espaço de trabalho do manipulador e podem ser provocadas por vários

fatores relacionados com a configuração das juntas para uma posição específica. Segundo Tsai (1999), uma singularidade implica que o manipulador perde a capacidade de movimento em um ou mais graus de liberdade.

Uma configuração singular pode ser identificada mediante o jacobiano do manipulador, quando seu determinante seja igual a zero. Se esta condição é satisfeita, o jacobiano está em uma configuração singular, isso é, a matriz do jacobiano perde seu posto completo (Tsai, 1999). Durante a execução de uma tarefa deve-se evitar que o manipulador adote uma posição singular, o qual deve ser considerado na solução da cinemática inversa.

Para a utilização do jacobiano na solução da cinemática inversa é usada a relação diferencial do erro *e* entre a posição desejada x_d e a posição atual x_e, mostradas nas equações (9) e (10):

$$\dot{e} = \dot{x_d} - J\dot{q} \tag{9}$$

$$x_e = J\dot{q} \tag{10}$$

Sempre que o erro for minimizado, a configuração das juntas expressa por *q* terá como resultado a posição desejada x_d. Com base nesse princípio, têm sido desenvolvidos diversos algoritmos focados na resolução da cinemática inversa como:

- Jacobiano pseudoinverso presente nos trabalhos de Lin; Lin; Lo (2009), Meredith; Maddock (2004) e Khalil; Dombre; Nagurka (2004);
- Mínimos quadrados amortecidos (DLS), utilizado por Kržič; Stoic; Kopač (2009) e Na; Yang; Jia (2008);
- Jacobiano transposto, aplicado no trabalho de Taira; Sagara; Katoh (2000); entre outros.

Dentre os algoritmos referidos, os de maior contribuição para o presente trabalho são os apresentados por Khalil; Dombre; Nagurka (2004) e por Na; Yang; Jia (2008). Khalil; Dombre; Nagurka (2004) propõem uma solução baseada na utilização do jacobiano pseudoinverso para obter a posição das juntas considerando a posição e orientação desejadas para o efetuador final, o qual propõe os seguintes passos:

1. Dada uma posição inicial das juntas, q^c, calcular a matriz $A_n^{0\,c}$, obtida mediante o método Denavit-Hartenberg;
2. Calcular os vetores do erro de posição (dX_p) e orientação (dX_r) com relação à posição e orientação desejadas. O vetor erro de posição é calculado mediante a equação (11), enquanto o vetor erro de rotação é determinado usando-se as equações (12) e (13).

$$\mathrm{dXp} = x_d - x_e \tag{11}$$

$$\mathrm{dXr} = \alpha \cdot \mathrm{u} \tag{12}$$

$$\mathrm{R}(\alpha, \mathrm{u}) = A_n^{0\,d}\,(A_n^{0\,c})^{-1} \tag{13}$$

   Onde as expressões com letra sobrescrita *d* correspondem aos valores desejados, enquanto as que têm letra sobrescrita *c* correspondem aos valores atuais; α e u correspondem respectivamente ao ângulo e ao eixo para representação da rotação.
3. Estabelecem-se valores limite de tolerância (S_p *e* S_r) os quais são comparados em cada iteração com os vetores erro. Se a norma de cada vetor erro for menor do que o valor limite correspondente, então q^c é a resposta da cinemática inversa. Se forem maiores do que os valores limite, os valores dos vetores erros são atualizados usando as equações (14) e (15), as quais foram estabelecidas em Strandberg (2004).

$$Se: \quad \|dX_p\| > S_p \;\Rightarrow\; dX_p = \frac{dX_p}{\|dX_p\|} S_p \tag{14}$$

$$Se: \quad \|dX_r\| > S_r \;\Rightarrow\; dX_r = \frac{dX_r}{\|dX_r\|} S_r \tag{15}$$

4. O Jacobiano pseudoinverso e a variação das juntas (dq) são calculados usando-se a equação (16).

$$dq = J^+ \cdot dX \tag{16}$$

Onde dX é um vetor coluna de seis elementos cujos primeiros elementos correspondem ao vetor erro de posição e os últimos ao vetor erro de orientação.

5. Deve-se então atualizar o valor de q^c, e repetir o processo usando-se a equação (17).

$$q^c = q^c + dq \tag{17}$$

Segundo os autores, esse método tem um rendimento aceitável para robôs industriais de grande porte com valores limite de 0,2m para erro de posição e 0,2 radianos para erro de rotação. Porém, o algoritmo proposto não considera as singularidades do modelo do robô, nem os limites de movimento das juntas.

Para garantir uma resposta efetiva na solução da cinemática inversa também foi considerado o trabalho apresentado por Na; Yang; Jia (2008), o qual baseia-se no método de mínimos quadrados amortecidos. Neste método, calcula-se um fator de amortecimento que evita que o método forneça uma resposta próxima de uma singularidade: quanto mais perto da singularidade, maior o fator de amortecimento e, portanto, a variação entre cada iteração nesta zona amortecida é menor. Aliás, o método pode ser complementado com um coeficiente calculado a partir dos valores máximos e mínimos das juntas, de tal maneira que a resposta fique dentro do volume de trabalho do robô. O procedimento iterativo é similar ao método anterior, mas o cálculo da variação das juntas é feito usando-se a equação (18).

$$dq = J^T(J^T J + D(\lambda)^2)^{-1} \cdot \vec{e} \tag{18}$$

Onde $D(\lambda)$ é uma matriz diagonal composta pelos coeficientes de amortecimento para cada junta λ_i, calculados pela equação (19), e o vetor erro total ($\vec{e}$) está representado pela equação (20)

$$\lambda_i = c_{DLS}\left[\frac{2 \cdot q_i - q_{i\_max} - q_{i\_min}}{q_{i\_max} - q_{i\_min}}\right]^{p_{DLS}} \tag{19}$$

$$\vec{e} = \begin{cases} dX \; se \; \|dX\| \leq D_{max} \\ D_{max} \; se \; \|dX\| > D_{max} \end{cases} \tag{20}$$

Os coeficientes c_{DLS} e p_{DLS} são números reais, p é um número par; e, D_{max} é o maior movimento permitido ao efetuador final em uma iteração.

Apesar da grande quantidade de pesquisas que fazem uso dos métodos de resolução numérica da cinemática inversa, Kucuk; Bingul (2004) lembram que existem desvantagens importantes na sua utilização, como:

- Uma estimação inicial adequada é necessária para o sucesso do processo iterativo;
- Não poder ser garantida a obtenção de uma resposta;
- Somente é obtida uma solução por execução;
- Se o jacobiano for singular não é possível obter uma resposta.

Além disso, dependendo do tipo de algoritmo, o desempenho varia. Por exemplo, Buss (2004) relata em sua pesquisa que o método de jacobiano transposto tem a vantagem de ser rápido, mas em manipuladores com dois efetuadores finais esse método tem um rendimento baixo. O método do jacobiano pseudoinverso requer um controle rigoroso dos limites das juntas para obter uma resposta adequada. Por outro lado, o método de mínimos quadrados amortecidos (DLS) apresenta um bom rendimento, mas com velocidade menor do que os outros métodos.

Aristidou; Lasenby (2009) comparam o desempenho desses métodos e seus derivados na solução da cinemática inversa de manipuladores com quatro morfologias diferentes, concluindo que o método DLS é mais rápido, mais robusto frente a singularidades e requer menor quantidade de iterações que o método de jacobiano transposto, mas o tempo de cada iteração é maior. Siciliano et al. (2010) também sugerem a utilização do método dos mínimos quadrados amortecidos para evitar singularidades na solução da cinemática inversa. Por esses motivos, o método a ser usado depende das exigências da aplicação.

2.3 Pesquisas envolvendo o padrão STEP-NC e a robótica

A quantidade de pesquisas sobre o STEP-NC tem sido extensa, inclusive antes da padronização sob a norma ISO 14649, devido ao interesse que poderia gerar na indústria por suas vantagens e características distintivas frente a outros formatos de troca de informações

de fabricação. Os principais tópicos de pesquisa relacionados ao padrão STEP-NC são analisados nesta seção.

Os trabalhos prévios à padronização apresentavam o cenário da fabricação com a utilização do código G, suas limitações e a necessidade de ter-se um novo padrão, com base em modelo similar ao apresentado na ISO 10303 para dados de fabricação, como é o caso dos trabalhos de Allen; Rosso Jr; Newman (2001) e Newman; Allen; Rosso (2003). Nesses trabalhos foram analisadas as novas arquiteturas requeridas nos sistemas CAD/CAM para integrar os elementos do sistema de manufatura mediante um formato de troca de informações baseado em *features* e até tecnologia baseada em agentes.

Após a padronização, surgiram mais duas correntes principais de pesquisas: as focadas no desenvolvimento de sistemas CAD/CAM/CAPP aderentes ao novo padrão e novos cenários de integração de manufatura; e as que procuravam criar controladores para as máquinas ferramenta de comando numérico, como mostradas na Figura 2.15.

No primeiro grupo podem ser identificadas publicações que propõem arquiteturas ou cenários de integração do padrão, como nos trabalhos de Suh et al. (2003), Sääski; Salonen; Paro (2005), Xu; Newman (2006), Kržič; Stoic; Kopač (2009), Qi et al. (2008), Cha et al. (2014) e Kayani; Rico (2015). Há também aqueles que aplicam o novo padrão em sistemas CAD/CAM, fazendo uma análise da sua estrutura e o estudo das funcionalidades e requerimentos, como em Bruyninckx (2001), Pobozniak (2013), Sziebig (2013) e Xiao et al. (2015).

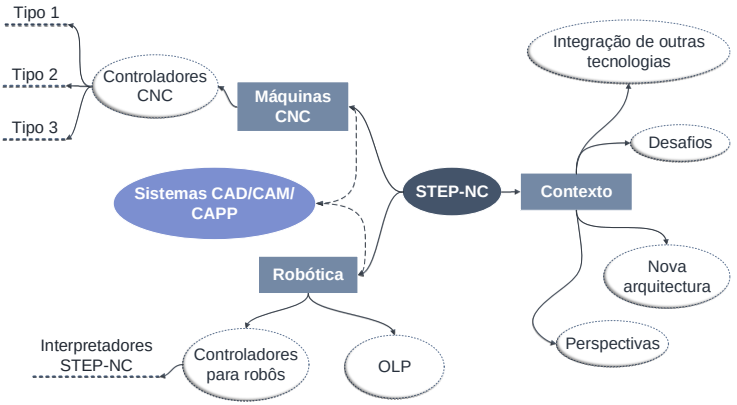


Figura 2.15. Principais temas de pesquisa relacionados com o STEP-NC
FONTE: Da autora

No segundo grupo, apesar do grande número de trabalhos existentes, a maioria aborda o desenvolvimento de controladores para máquinas CNC que se enquadram no tipo 1, apresentado na seção anterior. Entretanto, ultimamente têm surgido vários trabalhos que propõem a implementação de um controlador tipo 2, sendo o último de Kassim et al. (2015). Já tem surgido os primeiros controladores tipo 3, como os trabalhos de Zhao; Xu; Xie (2008) e Ridwan; Xu (2013).

No que se refere à robótica, embora alguns trabalhos façam alusão à utilização de robôs e sua integração mediante o uso do STEP-NC, como no caso de Vichare et al. (2009), Valilai; Nodeh; Houshmand (2010) e Song et al. (2012), esses trabalhos não abordam pontualmente os aspectos da geração da trajetória do robô. Já o trabalho de Solvang; Refsahl; Sziebig (2009) apresenta um sistema CAM para robôs industriais, o qual recebe o arquivo STEP-NC e gera a trajetória do robô em um ambiente virtual, logo essa trajetória é escolhida como caminho a ser seguido pelo efetuador final do robô. Neste trabalho, todos os cálculos são desenvolvidos no software Delmia IGRIP e os autores não abordam os temas relativos à cinemática dos manipuladores robóticos. Outro trabalho importante nesta área foi apresentado por Nagata et al. (2013), o qual apresenta o desenvolvimento de um sistema CAM para robôs servo-controlados. Nesse trabalho o modelo da peça é introduzido mediante o arquivo STEP-NC e o sistema envia diretamente as informações relativas ao movimento dos servomotores para o controlador do robô. Porém, o sistema está focado no robô RV1A, ficando limitado a robôs cujos controladores possuem arquitetura aberta, o que no ambiente industrial não é comum.

Por outro lado, Xiao; Huan; Dong (2014) apresentam um trabalho que procura estabelecer um modelo para a representação dos robôs industriais aplicando o padrão STEP (ISO 10303), e aplicam o modelo em um sistema CAM próprio, o qual recebe as informações em formato STEP-NC e calcula a trajetória para o robô. No trabalho são propostas novas entidades para descrever todos os parâmetros de um robô industrial e detalhadas na linguagem EXPRESS, mas não é mencionada a forma na qual a trajetória do robô é criada com base nos parâmetros cinemáticos que são considerados no modelo proposto.

No Grupo de Integração de Manufatura (GRIMA) da Universidade Federal de Santa Catarina (UFSC) foram desenvolvidas duas pesquisas relacionadas com o padrão STEP-NC (ISO 14649). A primeira foi apresentada por Benavente (2011), na qual foi criado um protótipo de sistema computacional CAD/CAPP/CAM aderente ao padrão STEP-NC, denominado STEP Modeler, desenvolvido usando-se a linguagem Java e

disponibilizado na Internet. Uma segunda pesquisa feita por Souza (2014) apresenta um controlador para uma máquina CNC de arquitetura aberta, desenvolvido na linguagem C/C++, o qual pode receber informações de entrada tanto mediante código G como mediante STEP-NC.

Pode-se perceber que existe uma lacuna na pesquisa relacionada à aplicação do padrão STEP-NC em manipuladores robóticos. Mesmo que surjam mais pesquisas voltadas ao desenvolvimento de controladores tipo 3 para robôs, aderentes ao padrão STEP-NC, estes seriam incluídos apenas na geração seguinte de controladores para robôs industriais, Assim, os robôs que operam atualmente na indústria permanecem aderentes ao processo convencional de programação, que é complexo e específico para cada robô.

Neste trabalho pretende-se desenvolver é um método de solução aplicável a vários tipos de robôs industriais, de maneira a facilitar a sua integração no novo ciclo de manufatura proposto pelos padrões STEP e STEP-NC. A solução procura se enquadrar como um controlador tipo 2, fornecendo um interpretador de STEP-NC capaz de calcular a trajetória de um robô industrial segundo seus parâmetros cinemáticos, tornando-se um método genérico que pode ser complementado com um *software* para simulação de robôs ou seus pós-processadores.

3. MÉTODO PROPOSTO PARA A GERAÇÃO DE TRAJETÓRIAS PARA ROBÔS

No capítulo anterior foram apresentadas as principais ferramentas conceituais no âmbito do padrão STEP-NC e da análise cinemática de manipuladores robóticos. Neste capítulo é descrita em detalhes a solução proposta para robôs industriais em operações de fresamento expressas sob o padrão STEP-NC.

3.1 Descrição do método proposto

Conforme o exposto no capítulo anterior com referência à pesquisa na área de STEP-NC aplicada à robótica e ao modelo apresentado por Pan et al. (2012) para a programação *off-line* (OLP) de robôs, a lacuna que o presente trabalho pretende preencher é ilustrada na Figura 3.1.

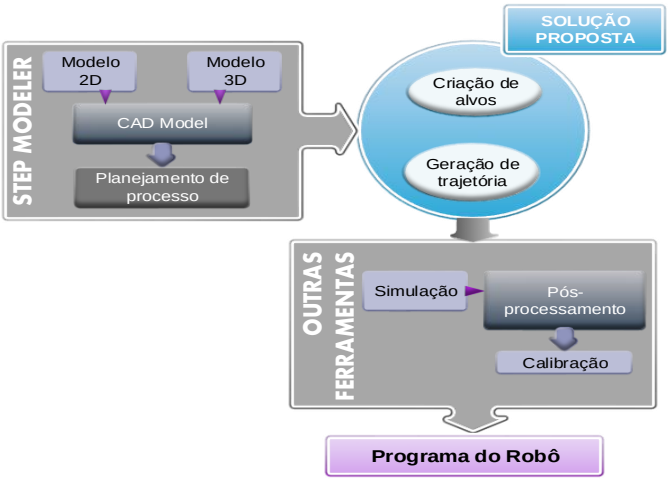


Figura 3.1. Situação do método proposto no ambiente da OLP
FONTE: Da autora

Percebe-se na Figura 3.1 que a solução proposta não interfere na estrutura existente, permitindo a conexão entre os sistemas cujas informações de fabricação são aderentes ao padrão STEP-NC (do lado esquerdo da figura) e as ferramentas já existentes para simulação e pós-processamento específicos para robôs industriais (na parte inferior da figura). Como essas ferramentas ainda não possuem a capacidade de interpretar as informações relacionadas a STEP-NC, tem-se neste caso um

controlador tipo 2 para robôs. Com isso, a usinagem de peças aderentes ao novo padrão é possível com os mesmos robôs industriais que já estão operando nas indústrias, sem incorrer em gastos decorrentes da compra ou modificação dos controladores dos robôs ou da aquisição de novas ferramentas de programação *off-line*. Além disso, esta solução procura ser aplicável a robôs do tipo serial, paralelo ou híbrido, desde que sejam não redundantes, e calcular sua trajetória baseando-se nos parâmetros cinemáticos próprios de cada robô.

Sendo assim, o método proposto neste trabalho utiliza como entrada os arquivos físicos do padrão STEP-NC criados com o software STEP Modeler (Benavente, 2011) para a usinagem de uma peça, interpreta suas informações e calcula a trajetória para cada robô, fornecendo o resultado em função das juntas que o compõem e em coordenadas XYZ acompanhadas da orientação do efetuador final. Com essas informações o processo pode ser simulado em um sistema computacional para a simulação de robôs e, se necessário, pode ser traduzido para a linguagem própria do fabricante mediante pós-processadores.

O método proposto neste trabalho é composto por etapas, as quais são resumidas na Figura 3.2.

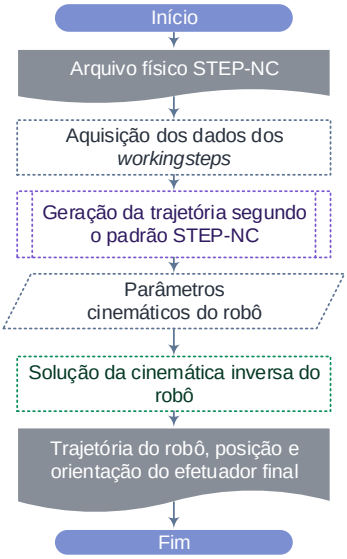


Figura 3.2. Diagrama de fluxo do método proposto.
FONTE: Da autora

Como mencionado no capítulo anterior, um arquivo físico aderente ao padrão STEP-NC contém as informações necessárias para a usinagem de uma peça. Porém, pelo fato de ser um padrão para troca de informações de alto nível, o arquivo não descreve explicitamente a trajetória que uma máquina ferramenta deve percorrer para executar o processo de usinagem. Por esta razão, o método proposto foi construído com base nas informações contidas no arquivo STEP-NC.

O primeiro passo consiste em adquirir e interpretar as informações do arquivo, identificando as *features* e os parâmetros das operações de usinagem de cada *workingstep*.

Uma vez coletados estes dados, uma trajetória de usinagem é criada, tomando como referência um ponto específico na peça. O tipo de trajetória é definido pela estratégia de fresamento, a qual consta no arquivo analisado, como é mostrado na Figura 3.3.

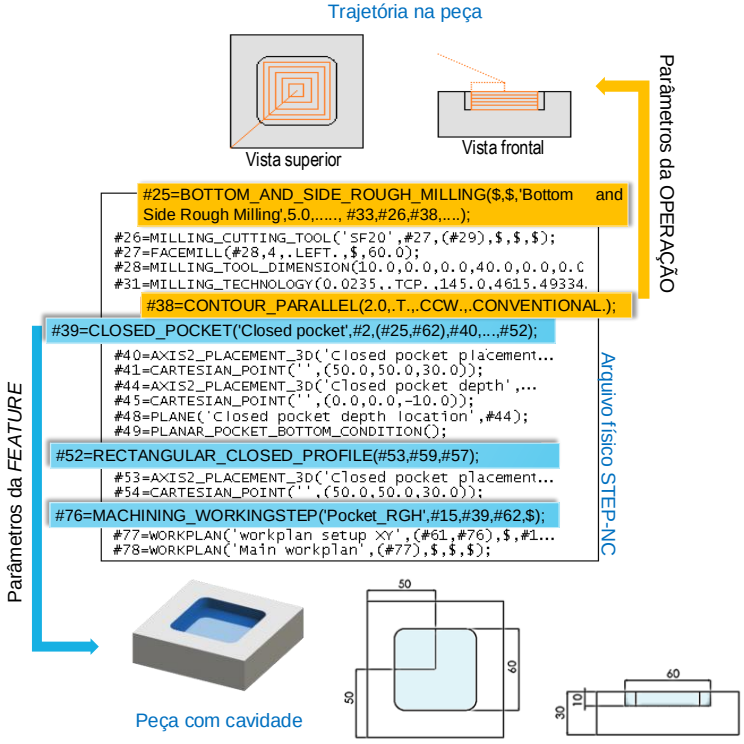


Figura 3.3. Exemplo de leitura e interpretação de dados do arquivo STEP-NC.
FONTE: Da autora

Após a geração da trajetória para a usinagem da peça é necessário obter os parâmetros correspondentes ao robô: a localização do robô com referência à peça, o tipo de robô a ser utilizado e seus parâmetros cinemáticos. Os parâmetros cinemáticos abrangem tanto os limites de movimento e tipo de cada junta, os graus de liberdade do robô e os parâmetros requeridos pelo método Denavit-Hartenberg para o caso dos manipuladores seriais. Como mencionado no capítulo 2, este método fornece um modelo matemático da cinemática do robô, o qual é a base para a solução da cinemática inversa mediante a utilização de métodos numéricos, que é o passo seguinte do método.

Para obter os pontos da trajetória que o robô irá percorrer deve-se converter os pontos da trajetória gerada para pontos com referência na base do robô. Cada um desses pontos são introduzidos para o cálculo da cinemática inversa com base no método de mínimos quadrados amortecidos apresentado por Na; Yang; Jia (2008). O resultado desses cálculos corresponde à configuração das juntas para cada ponto da trajetória do manipulador robótico.

Ao obter-se a posição das juntas para cada ponto da trajetória e a posição e orientação do efetuador final, o método proposto torna-se independente do fabricante e da linguagem de programação do robô ou de software proprietário. Já os dados podem ser inseridos em uma plataforma informática para a sua simulação ou tradução para uma linguagem específica.

3.2 Implementação do método proposto

Para a implementação do método descrito na seção anterior, foi desenvolvido um sistema computacional na linguagem Java, usando-se o ambiente de programação NetBeans v.8.0.2. e bibliotecas especializadas em cada uma das etapas citadas na Tabela 3.1.

Dentro das bibliotecas especializadas, a biblioteca JSDAI (LKSoftware GmbH, 2015) foi utilizada para a leitura e manipulação de dados definidos na linguagem EXPRESS, enquanto a biblioteca Symja (Axelclk, 2015) foi utilizada no âmbito deste trabalho nas operações envolvendo álgebra simbólica na linguagem Java.

O sistema desenvolvido requer como entrada um arquivo físico STEP-NC gerado pelo software STEP Modeler (Benavente, 2011), e tem como saída a trajetória de usinagem para o robô sendo utilizado.

Nas próximas seções serão apresentados os detalhes da implementação de cada uma das etapas mencionadas.

Tabela 3.1. Bibliotecas para implementação do método proposto

Etapa	Biblioteca	Versão	Licença
Leitura do arquivo STEP-NC	JSDAI	4.3	AGPL v3
Interpretação do arquivo STEP-NC			
Geração das trajetórias	Symja	2015-04-06	GNU v3
Cinemática inversa do robô			

3.2.1 Leitura do arquivo físico aderente ao padrão STEP-NC

Segundo o estabelecido pela norma ISO 14649-1 (2002), o arquivo físico STEP-NC é expresso na linguagem EXPRESS e, por essa razão, é necessário conhecer o *schema*[2] que define as entidades e atributos utilizados para descrever a informação no arquivo. Nesse caso foi criado um *schema* denominado "combined_schema" a partir da descrição das entidades citadas nas partes 1, 10, 11 e 111 do padrão ISO 14649.

A partir desse *schema* foi compilado um arquivo com extensão ".jar" mediante a biblioteca JSDAI. Este arquivo salva a informação descrita no *schema* como um dicionário que o programa em Java vai utilizar para interpretar as informações dos arquivos EXPRESS.

3.2.1.1 Obtenção dos dados de projeto e o *workplan*

Com o arquivo ".jar" o programa em Java pode ler as informações contidas no arquivo físico e interpretá-las adequadamente, começando pelos dados do projeto e o *workplan* estabelecido para a peça. Para facilitar a extração dos dados contidos no arquivo físico foram criadas classes em Java para algumas das entidades e seus atributos. Assim, as informações coletadas correspondentes ao projeto e ao *workplan* são mostradas na Figura 3.4.

[2] Um *schema* define um conjunto de objetos que têm um significado e propósito relacionado. Traduzido de Schenck; Wilson (1993)

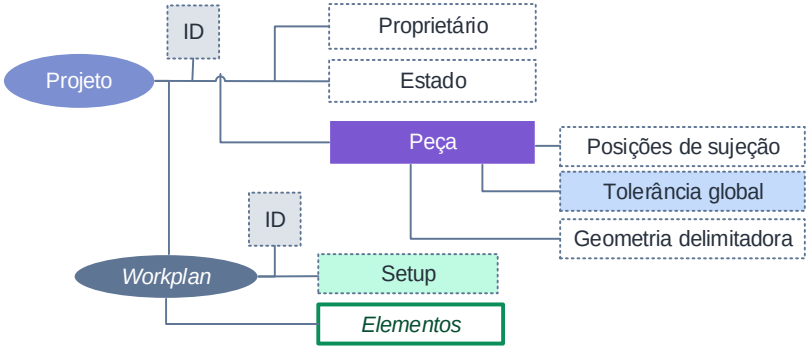


Figura 3.4. Informações coletadas referentes ao Projeto e *Workplan*.
FONTE: Da autora, baseado em ISO 14649-10 (2002)

Essas primeiras informações extraídas são unicamente de caráter informativo. No entanto, dentro do *workplan* encontram-se as entidades de maior interesse deste trabalho, descritas na próxima seção.

3.2.1.2 Obtenção dos *workingsteps*, *features* e operações

No capítulo 2 foram apresentadas as definições de *workplan* e *workingstep*, ressaltando-se a importância deste último na descrição das operações de usinagem que devem ser executadas na sequência especificada para obter a peça final. Tendo isso em vista, as informações relativas a cada *workingstep* foram extraídas respeitando-se a ordem dentro do *workplan* e os atributos de cada um deles, como indicado na seção anterior.

Segundo a norma ISO 14649-10 (2002), os elementos que compõem o *workplan* podem ser: *workingsteps*, funções de máquina ou estruturas de programa. Porém, tendo em vista o foco deste trabalho, serão analisados somente os *workingsteps*, especificamente os *machining workingsteps*. Cada um deles tem uma *feature* e uma operação de usinagem associadas, as quais são definidas como entidades na norma.

A totalidade de entidades foi analisada com o intuito de delimitar a quantidade de dados realmente necessários no contexto deste trabalho. A Figura 3.5 resume as principais entidades cujas informações foram extraídas, por serem vitais para a construção das trajetórias.

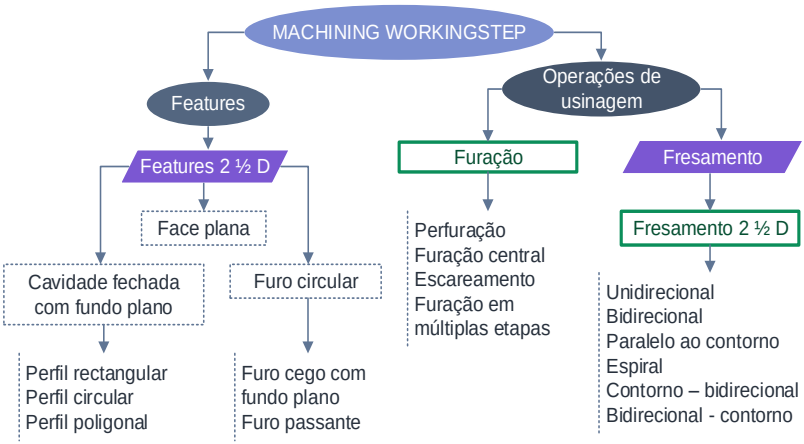


Figura 3.5. Entidades da norma ISO 14649 abordadas neste trabalho.
FONTE: Da autora

As informações das *features* e das operações disponíveis são armazenadas como atributos dos objetos da linguagem Java criados para cada entidade (consultar o Apêndice A para maiores detalhes). Dessa forma, o arquivo físico é lido na íntegra apenas uma vez no início do processo, a informação é extraída, armazenada na memória do computador, e fica disponível para consulta na forma dos tipos de dados suportados em Java. As entidades obtidas estão definidas nas normas ISO 14649-10 (2002), ISO 14649-11 (2002) e ISO 14649-111 (2002).

3.2.2 Geração das trajetórias definidas no padrão STEP-NC

Como apontado acima, o sistema computacional foi desenvolvido na linguagem Java. Entretanto, os algoritmos para a geração de trajetórias foram implementados inicialmente no software MatLab®, para uma visualização rápida dos caminhos obtidos. Depois de verificar os algoritmos, estes foram adaptados para Java e sua paridade verificada mediante comparação numérica dos pontos obtidos.

Para a geração das trajetórias, as *features* foram divididas em duas categorias principais: *features* com perfil retangular e *features* com perfil circular ou poligonal. Para cada uma destas categorias foram utilizadas formas diferentes para a geração das trajetórias segundo as estratégias de usinagem citadas na norma ISO 14649, as quais são descritas nas seções seguintes.

3.2.2.1 Trajetórias para *features* segundo seu tipo de perfil

A geração das trajetórias para *features* com perfil retangular foi implementada baseando-se nas definições apresentadas pela norma ISO 14649-11 (2002). Para todas as estratégias de usinagem, o primeiro passo foi calcular os pontos do perfil alvo. Depois, foi obtido um retângulo interno com os lados paralelos ao retângulo alvo, com uma separação igual ao raio da ferramenta a ser usada. Este retângulo interno é referido a partir daqui como primeiro contorno interno.

Com base neste primeiro contorno é calculada a trajetória interna à *feature*. Na obtenção da trajetória é considerado o parâmetro de sobreposição (em inglês: *overlap*) ilustrado na Figura 3.6, do qual dependem o sentido da trajetória (horário ou anti-horário) e o sentido de avanço da ferramenta, para todas as estratégias de usinagem.

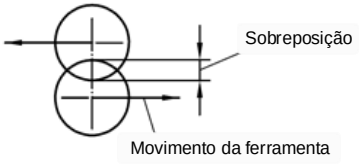


Figura 3.6. Parâmetro sobreposição como definido no padrão STEP-NC
FONTE: ISO 14649-11 (2002)

O procedimento é o seguinte:

- Para as trajetórias unidirecional e bidirecional, são traçadas linhas paralelas à base do retângulo, das quais são determinadas suas interseções com o primeiro perfil interno já definido, obtendo-se segmentos de reta. A distância d entre as linhas paralelas é calculada mediante a equação (21), uma vez que a sobreposição está definida na norma como uma porcentagem do diâmetro da ferramenta. Os pontos inicial e final de todos os segmentos de reta encontrados são ordenados segundo a estratégia desejada, de modo a formar o caminho da ferramenta para gerar o perfil retangular requerido.

$$d = (2 \times raio\ da\ ferramenta)(1 - sobreposição) \qquad (21)$$

- Para o caso das trajetórias paralela ao contorno e espiral foram traçados retângulos internos e paralelos ao primeiro contorno. Para determinar a distância entre os retângulos foi aplicada a mesma equação (21). Os retângulos obtidos são interligados um ao outro

dependendo da estratégia, seja com um segmento adicional ao retângulo (trajetória paralela ao contorno) ou como uma extensão de um dos lados do retângulo anterior (trajetória espiral).

- Para o caso das trajetórias mistas, primeiro são geradas as trajetórias simples separadamente, as quais são concatenadas posteriormente.

Como passo final são adicionados os pontos inicial e final da trajetória interna, com uma distância no eixo Z de cada *feature* igual ao valor estabelecido pelo atributo "plano de retração" (em inglês: *retract plane*).

A Figura 3.7 mostra todas as trajetórias obtidas para o caso de uma cavidade com perfil retangular, sendo que no primeiro ponto localiza-se a ferramenta, a qual é representada com o círculo da cor azul, enquanto o perfil alvo é representado na cor vermelha e a trajetória interna é mostrada na cor roxa.

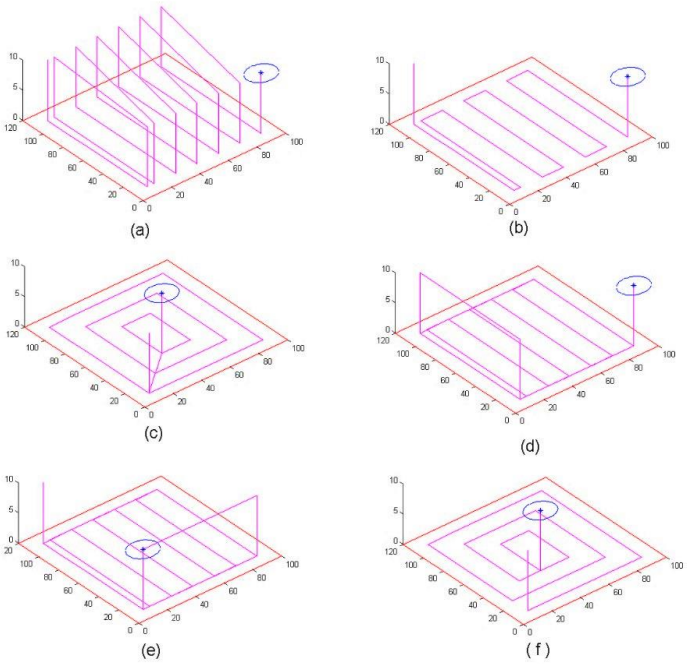


Figura 3.7. Trajetórias geradas para *features* com perfil retangular: (a) trajetória unidirecional, (b) bidirecional, (c) paralelo ao contorno, (d) bidirecional-contorno, (e) contorno-bidirecional; e, (f) espiral
FONTE: Da autora

Em caso de perfil poligonal não retangular, o primeiro passo consiste em identificar se o perfil está definido como inscrito ou circunscrito em relação ao círculo de raio especificado no arquivo. Se o perfil for inscrito, o primeiro perfil é traçado dentro do círculo com raio *r* definido no arquivo, porém, se for circunscrito, calcula-se o raio *R* do círculo no qual o polígono fica inscrito, usando-se a equação (22), onde *n* é o número de lados.

$$R = \sqrt{r^2 + \left(\frac{2 \cdot r \cdot \tan\left(\frac{\pi}{n}\right)}{2}\right)^2} \tag{22}$$

Calculado o valor do raio do círculo gerador dos pontos, o processo continua igual nos dois casos:

- O perfil alvo é traçado sobre o círculo gerador modelado usando-se a equação (23) onde o parâmetro *t* depende do número de lados do perfil, e fica dentro do intervalo $0 \leq t \leq 2\pi$, obtendo-se os pontos de cada vértice do polígono.

$$c = \begin{cases} r \times \cos t \\ r \times \operatorname{sen} t \end{cases} \tag{23}$$

- A partir do perfil alvo são traçadas linhas paralelas ao vetor *feed* (representado mediante um vetor unitário no padrão STEP-NC) ou círculos concêntricos internos que geram polígonos paralelos ao perfil alvo, dependendo da estratégia requerida. A Figura 3.8 mostra os passos do cálculo da trajetória para *features* com perfil poligonal, com estratégia espiral.

As trajetórias obtidas para o caso de um perfil poligonal hexagonal (n=6), para todas as estratégias, são apresentadas na Figura 3.9.

Para o caso das *features* com perfil circular, estas foram aproximadas mediante um perfil poligonal com n=40, uma vez que a equação utilizada para a geração dos seus pontos é a equação do círculo parametrizado, e, conforme Kumar; Sharma; Bhowmick (2010) a aproximação a partir de um polígono com $n \geq 20$ gera um erro menor ao 2%. Com este precedente, mediante uma relação de áreas entre um polígono regular inscrito com n=40 e um círculo, o erro estimado é de 0,41%.

Os passos para a obtenção da trajetória são os mesmos que para um perfil poligonal segundo a estratégia desejada.

Nota-se que todas as trajetórias, tanto para *features* com perfil retangular quanto circular ou poligonal, foram calculadas sobre o plano XY na etapa inicial. Contudo, em um passo posterior à obtenção dos pontos, as trajetórias são trasladadas até o ponto e plano desejados, o que faz com que todos os pontos da trajetória fiquem localizados adequadamente com referência à peça.

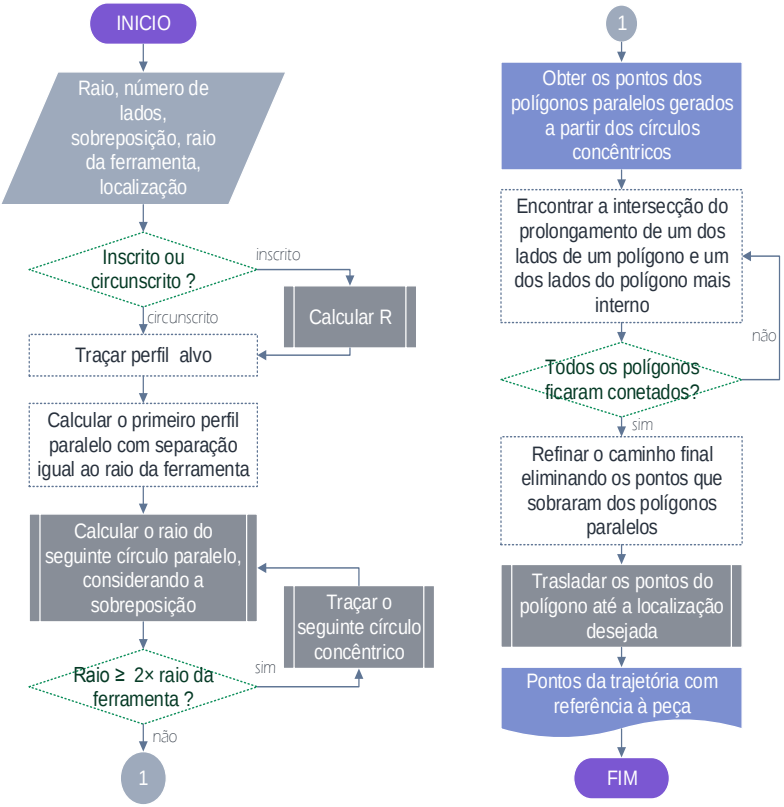


Figura 3.8. Algoritmo para calcular a trajetória espiral de um perfil poligonal
FONTE: Da autora

3.2.2.2 Trajetórias considerando a profundidade das *features* 2½D

Até aqui as trajetórias foram geradas somente para um plano, mas as *features* analisadas neste trabalho são do tipo 2½D e, nesse caso, é

necessário completar a trajetória na direção do eixo Z do sistema de referência da *feature*, tendo como resultado uma extração de volume propriamente dita.

Por esse motivo, as trajetórias geradas para um plano são repetidas em camadas separadas por uma distância igual à profundidade de corte, estabelecida nos atributos da operação de usinagem, até atingir a profundidade desejada. Os pontos de cada camada são concatenados e o resultado constitui a trajetória total de usinagem.

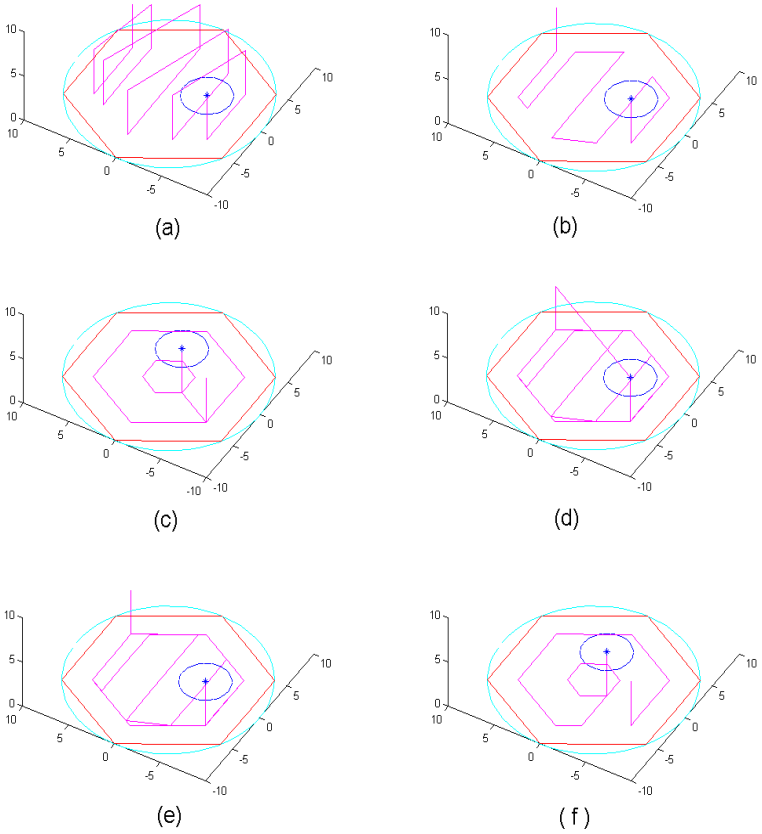


Figura 3.9. Trajetórias para *features* com perfil poligonal hexagonal: (a) trajetória unidirecional, (b) bidirecional, (c) paralelo ao contorno, (d) bidirecional-contorno, (e) contorno-bidirecional; e, (f) espiral
FONTE: Da autora

3.2.2.3 Conversão de coordenadas da trajetória gerada

Uma vez que a trajetória completa de usinagem foi gerada com referência na peça, esta é convertida para coordenadas com referência à base do robô. A Figura 3.10 apresenta a localização dos sistemas de coordenadas: na base do robô na cor verde, na mesa de trabalho na cor azul, e na peça na cor vermelha.

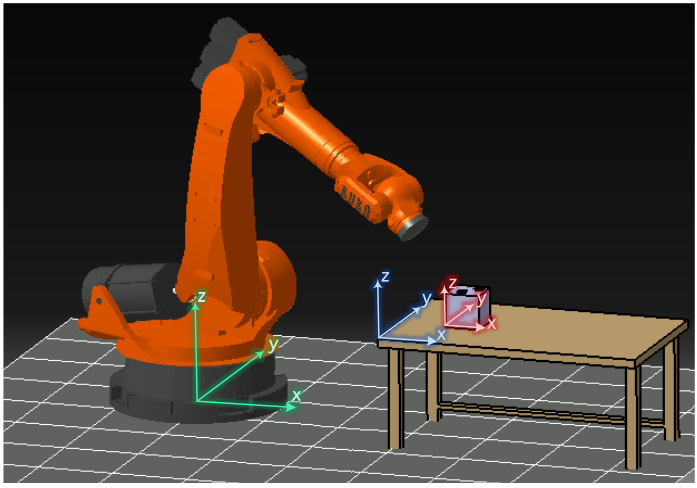


Figura 3.10. Localização dos sistemas de referência.
FONTE: Da autora

Para a conversão entre sistemas de referência usa-se as matrizes de deslocamento rígido mencionadas no capítulo 2, na sequência indicada na equação (24).

$$T_P^R = T_M^R \cdot T_P^M \tag{24}$$

Nessa expressão, R se refere ao robô, M à mesa de trabalho e P à peça, a letra sobrescrita indica a nova referência e a subscrita indica a referência de origem. Assim, por exemplo, T_P^R é a matriz que relaciona as coordenadas expressas com referência na peça com coordenadas com referência na base do robô.

Com a intenção de facilitar a especificação da localização dos elementos no ambiente de trabalho, no sistema implementado é foi considerada uma distância em X, Y e Z entre a origem do sistema de coordenadas do robô e a origem do sistema de coordenadas da mesa de

trabalho e, também, entre o sistema de coordenadas da mesa de trabalho e a origem do sistema de referência da peça. Além disso, estabeleceu-se uma rotação ao redor do eixo Z entre o robô e a mesa e entre a mesa e a peça. Com estes parâmetros são construídas as matrizes de deslocamento rígido, e cada um dos pontos a trajetória (p) é convertido mediante a equação (25) para obter a trajetória com referência na base do robô.

$$\begin{bmatrix} p_P^R \\ 1 \end{bmatrix} = T_P^R \cdot \begin{bmatrix} p^P \\ 1 \end{bmatrix} \tag{25}$$

3.2.3 Solução da cinemática inversa do robô

Tal como mencionado no capítulo 2, o problema central da cinemática inversa é determinar a posição das juntas do robô para atingir uma determinada posição e orientação no efetuador final. Neste trabalho, as posições alvo são os pontos da trajetória de usinagem. Assim, foi necessário implementar um método para a solução da cinemática inversa dos robôs não redundantes que contemplasse tanto a posição quanto a orientação do efetuador final, pois deve-se lembrar que as *features* podem estar em qualquer face da peça e o efetuador final do robô deve estar perpendicular àquela face.

Já foi apontado anteriormente que a análise da cinemática inversa é mais direta nos manipuladores paralelos, enquanto nos manipuladores seriais tal análise é mais complexa devido ao número de soluções possíveis. Por este motivo, a solução para a cinemática inversa foi dividida segundo o tipo de robô a ser utilizado.

3.2.3.1 Cinemática inversa para manipuladores seriais

Por este trabalho ser dirigido à programação *off-line* de robôs, a velocidade de processamento não é um fator crítico, mas pela quantidade de dados analisados é importante obter-se uma resposta rápida e que não precise de validações adicionais. Por essas razões foram utilizadas partes dos métodos citados por Khalil; Dombre; Nagurka (2004) e Na; Yang; Jia (2008), referidos nos capítulo 2. Juntos eles conduzem a uma solução rápida para a cinemática inversa, contemplando as limitações físicas das juntas, evitando singularidades e avaliando a posição e orientação do efetuador final.

Contudo, sendo um método de resolução numérica, foi necessário estabelecer um ponto de início, do qual depende o sucesso na busca da solução dentro de um número determinado de iterações. Também foi

adicionada uma etapa que controla o valor de cada junta e o atualiza caso uma delas atinja seu limite superior ou inferior, evitando a finalização do método devido ao coeficiente de amortecimento na parte de mínimos quadrados amortecidos. O procedimento utilizado é o seguinte:

1. Constrói-se o modelo cinemático do robô com base no método de Denavit-Hartenberg, e o Jacobiano em termos das variáveis das juntas;
2. Lê-se a posição desejada em termos das coordenadas (x, y, z) e orientação em termos dos ângulos (*yaw*, *pitch*, *roll*). Calcula-se a matriz de rotação da posição desejada;
3. Estabelece-se a configuração inicial das juntas nos seguintes casos:
   - Se for o primeiro ponto a ser calculado ou se não tiver uma configuração anterior definida, são estabelecidas cinco possíveis configurações iniciais: todas as juntas no valor máximo, no valor mínimo, no valor médio, no valor médio da parte positiva, e no valor médio da parte negativa. Dessa forma, o método pode encontrar uma solução considerando como suposição inicial cada uma das cinco configurações.
   - Se tiver uma configuração anterior definida, a suposição inicial do algoritmo é essa configuração anterior.
4. Calculam-se os erros de posição e orientação usando-se as equações (11), (12) e (13);
5. Comparam-se os erros com os valores limite: 0,0001 milímetros e 0,0001 radianos. Se o valor for menor ou igual ao limite, o processo termina; se for maior, o processo continua;
6. Calcula-se o vetor erro total $\vec{e}$ usando-se a equação (26), onde dX_p é o vetor erro de posição; dX_r, o vetor erro de orientação; S_p e S_r, o limite de tolerância da posição e rotação respetivamente.

$$\vec{e} = [dX_p; dX_r] \begin{cases} Se \left\| dX_p \right\| > S_p \Rightarrow dX_p = \dfrac{dX_p}{2} \\ Se \left\| dX_r \right\| > S_r \Rightarrow dX_r = \dfrac{dX_r}{2} \end{cases} \quad (26)$$

7. Calculam-se os coeficientes de amortecimento de cada junta mediante a equação (19), onde c_{DLS}= 0,5 e p_{DLS}= 8;
8. Avalia-se o jacobiano na posição atual e encontra-se a variação de cada junta usando-se a equação (18). Atualiza-se o valor de cada junta mediante a equação (17);

9. Compara-se o valor atualizado de cada junta com seus valores máximos e mínimos correspondentes. Se o valor atual for maior do que o limite máximo ou menor do que o limite mínimo, atualiza-se o valor da junta ao 75% do limite contrário. Por exemplo, se uma junta rotativa, cujos limites de movimento máximo e mínimo são 180° e -180° respetivamente, toma o valor de 181° (maior do que o limite máximo), o algoritmo substitui esse valor pelo 75% do limite mínimo que é -135°. Com este corte no valor da junta, garante-se que o valor fique dentro da faixa de movimento e que o método não continue em uma zona com alto coeficiente de amortecimento indefinidamente.
10. Depois da avaliação no valor das juntas, repete-se o processo a partir do passo 4.

As respostas obtidas mediante este método foram comprovadas com pontos aleatórios e trajetórias lineares curtas no software MatLab, antes do método ser implementado na linguagem Java.

3.2.3.2 Solução da cinemática inversa para manipuladores paralelos

Para o caso de manipuladores paralelos, tentou-se implementar uma solução mediante o método Denavit-Hartenberg, como apresentado por Tsai (1999). Entretanto, devido à complexidade relatada pelo autor, considerou-se que uma solução geral não seria a mais efetiva neste tipo de robô.

O método escolhido para a solução de manipuladores paralelos foi o método geométrico descrito por Merlet (2006), Tsai (1999), Siciliano (1999) e Isaksson; Eriksson; Nahavandi (2014).

3.2.3.3 Caso dos manipuladores híbridos

No caso de manipuladores híbridos, este trabalho limitou-se a analisar os robôs onde a estrutura paralela corresponde a um manipulador 3 U-P-S, e a estrutura serial corresponde a um punho esférico, similar ao robô Tricept, descrito no próximo capítulo.

Uma adaptação foi feita para possibilitar a fusão dos dois métodos de solução: a estrutura paralela foi modelada mediante o método geométrico e a parte serial pode ser resolvida com o método numérico apresentado anteriormente.

A adaptação baseia-se no método de resolução de manipuladores antropomórficos com punho esférico apresentado por Siciliano et al.

(2010). Porém, neste trabalho esse método foi aplicado à resolução de robôs híbridos, resumida dos seguintes passos:

1. Calcula-se a posição do punho esférico mediante a equação (27), onde p_w é a posição do centro do punho esférico, p_e é a posição do efetuador final, d_e é a distância do centro do punho esférico até o efetuador final; e a_e é o vetor coluna da matriz de rotação desejada para o efetuador final, a qual é indicada na equação (28) (confira a Figura 3.11 para maiores detalhes);

$$\overrightarrow{p_w} = \overrightarrow{p_e} - d_e \cdot \overrightarrow{a_e} \tag{27}$$

$$R_e^0 = [\overrightarrow{n_e} \quad \overrightarrow{s_e} \quad \overrightarrow{a_e}] = \begin{bmatrix} n_x & s_x & a_x \\ n_y & s_y & a_y \\ n_z & s_z & a_z \end{bmatrix} \tag{28}$$

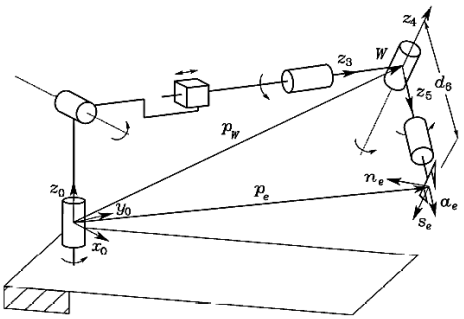


Figura 3.11. Solução de manipuladores com punho esférico.
FONTE: Siciliano et al. (2010)

2. Soluciona-se a cinemática inversa para p_w, correspondente à parte paralela do manipulador híbrido;
3. Calcula-se a matriz de rotação R_w^0 e a matriz homogênea T_w^0;
4. Obtém-se a matriz homogênea correspondente ao punho esférico mediante a equação (29);

$$T_e^w = (T_w^0)^{-1} \cdot T_e^0 \tag{29}$$

5. Resolve-se a cinemática inversa para a parte serial obtendo-se a rotação e posição desejadas a partir da matriz homogênea T_e^w, como indicado na equação (30).

$$T_e^w = \begin{bmatrix} R_e^w & p_e^w \\ 0 \quad 0 \quad 0 & 1 \end{bmatrix} \tag{30}$$

Com a introdução deste desacoplamento cinemático foi possível solucionar a cinemática inversa dos robôs híbridos utilizando os métodos já descritos.

Todos os algoritmos foram testados no software MatLab e, depois, reescritos completamente na linguagem Java, com o auxílio da biblioteca Symja.

3.2.3.4 Uniformidade na movimentação do robô durante a execução da trajetória

Durante os testes efetuados com robôs híbridos e seriais, foi observado que na execução de uma trajetória contínua, em algumas ocasiões, o robô requereu um ajuste na configuração das suas juntas subitamente para atingir o ponto seguinte da trajetória. Esta mudança na posição foi efetuada com um giro completo da articulação mais crítica para o robô, alterando irremediavelmente a trajetória. Se esses erros ocorressem durante a execução da usinagem, a peça e os elementos do ambiente de trabalho poderiam ser danificados.

A Figura 3.12 ilustra esta situação, onde a trajetória a ser seguida é a trajetória circular azul, mas devido à reconfiguração do robô, o efetuador final traça um círculo menor (vermelho) durante o ajuste da posição das suas juntas, provocando um desvio da trajetória na superfície a ser usinada.

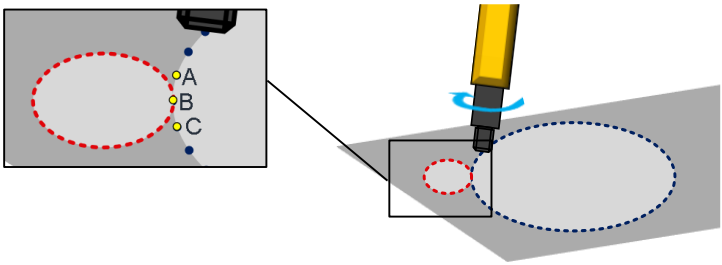


Figura 3.12. Ajuste súbito de configuração do robô.
FONTE: Da autora

Foi determinado que este movimento é produzido quando o algoritmo de cinemática inversa resolve o ponto atual (B) baseado no ponto anterior (A), mas para atingir o ponto seguinte (C) ele precisa configurar suas juntas muito perto dos limites de movimentação ou perto de uma singularidade, calculando-se outra configuração mais adequada para o robô, porém, implica um movimento de até 360° em uma das juntas.

Apesar desta situação não ser muito frequente na execução das trajetórias, buscou-se uma solução que foi adicionada como validação prévia à obtenção dos resultados finais da trajetória do robô. A solução consiste em adicionar três pontos à trajetória original do robô, conforme descrito abaixo e mostrado na Figura 3.13:

1. O primeiro ponto corresponde ao ponto B deslocado no eixo Z sobre a superfície, e a cinemática inversa para ele é calculada a partir da posição do robô para o ponto A;
2. O segundo ponto é o mesmo ponto B deslocado, mas a cinemática inversa é calculada a partir da posição do robô para o ponto C;
3. Finalmente, o terceiro ponto é o ponto B sem deslocar, onde a cinemática inversa foi calculada a partir da posição do robô para o ponto C.

Com este procedimento garantiu-se que o robô atingisse os pontos e mudasse de configuração se necessário, mas em uma posição que não prejudicasse nem a peça nem o processo.

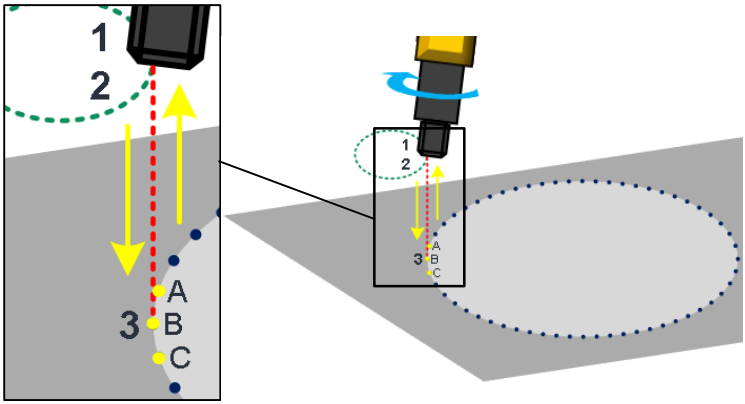


Figura 3.13. Solução proposta para manter a uniformidade na trajetória.
FONTE: Da autora

Neste capítulo foram descritas as etapas do método proposto neste trabalho, que são: leitura e interpretação do arquivo físico aderente ao padrão STEP-NC, geração de trajetórias de usinagem na peça e solução da cinemática inversa do robô para a trajetória. Com os dados resultantes deste processo é possível testar os métodos propostos em conjunto, e avaliar a solução proposta para o problema da aplicação de robôs em usinagem de peças expressas sob o padrão STEP-NC.

4. MATERIAIS E MÉTODOS EXPERIMENTAIS

No capítulo anterior foram descritos os métodos utilizados e algoritmos desenvolvidos na implementação da solução proposta. Neste capítulo será apresentado o sistema computacional desenvolvido com base nesses algoritmos, e os recursos e métodos utilizados na obtenção e avaliação dos dados obtidos nos testes.

4.1 Ferramenta desenvolvida

Como foi mencionado no capítulo anterior, o sistema computacional para a validação do método proposto foi desenvolvido na linguagem Java. No desenvolvimento foram utilizadas as seguintes ferramentas:

- Kit de desenvolvimento de software Java ou JDK (em inglês*: Java Development Kit*) v 8.0_25;
- Ambiente de execução de Java ou JRE (em inglês: *Java Runtime Environment*.) v 8.0_25;
- Ambiente de desenvolvimento integrado de Java ou IDE (em inglês *Integrated Development Environment*) Netbeans 8.0.2.

A ferramenta computacional utiliza as bibliotecas especializadas listadas no capítulo 3, além da biblioteca Swing utilizada para criar a interface gráfica. Entretanto, na versão final, a ferramenta toma o nome de *STEP-NC to Robot Trajectory*, como mostrado na Figura 4.1.

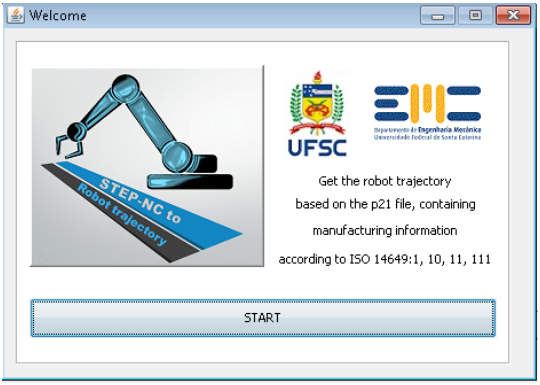


Figura 4.1. Tela de Início do sistema desenvolvido.
FONTE: Da autora

A interface gráfica implementada permite a entrada de dados pelo usuário, começando pelo arquivo físico STEP-NC na aba Arquivo STEP-NC (em inglês: *STEP-NC file*). O sistema lê o arquivo e apresenta as informações mais importantes de cada *workingstep* identificado, como mostra a Figura 4.2

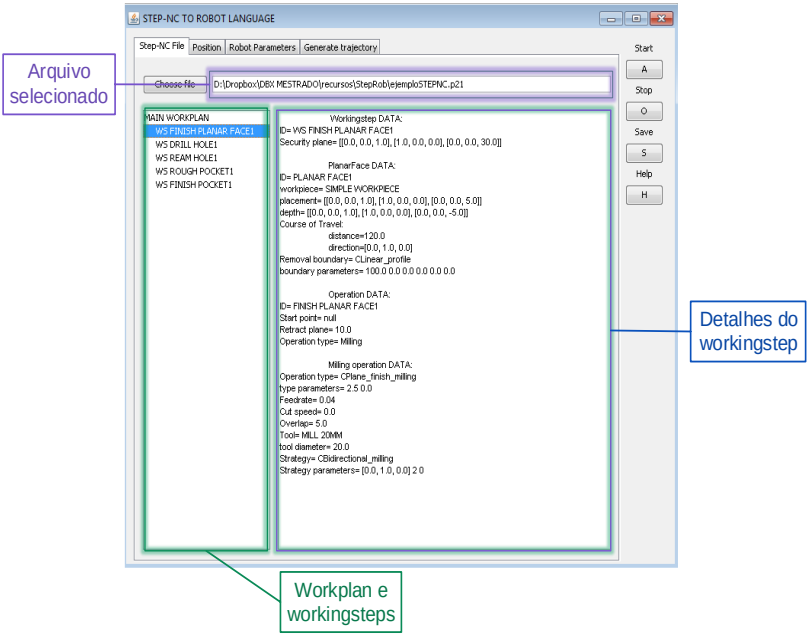


Figura 4.2. Entrada do arquivo STEP-NC no sistema desenvolvido.
FONTE: Da autora

Logo depois de entrar com o arquivo, o método de leitura e interpretação do arquivo físico descrito no capítulo anterior começa sua execução, auxiliado pela biblioteca JSDAI.

Na seguinte aba Posição (em inglês: *Position*) define-se os dados de posição e orientação dos elementos que constituem o ambiente de trabalho: o robô, a mesa e a peça, como mostrado na Figura 4.3.

Os dados introduzidos são coletados e armazenados para a construção das matrizes de deslocamento rígido, as quais são necessárias para a conversão das coordenadas da trajetória, desde o sistema de coordenadas na peça até a base do robô.

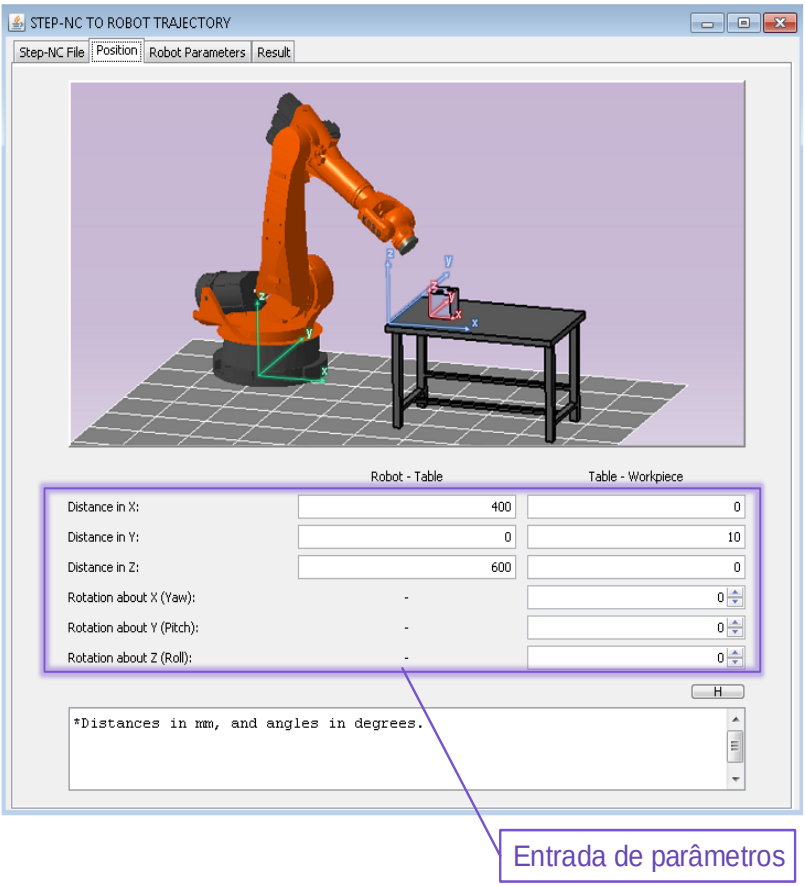


Figura 4.3. Entrada de parâmetros de posição dos elementos.
FONTE: Da autora

A Figura 4.4 ilustra a interface de entrada de parâmetros do robô a ser utilizado. No caso de robô serial, os parâmetros podem ser inseridos nas tabelas localizadas à esquerda da tela. Estes dados são armazenados para calcular o modelo do robô mediante as matrizes baseadas no método Denavit-Hartenberg, requerido para o método de resolução da cinemática inversa. No caso dos robôs paralelos e híbridos, o modelo geométrico dos robôs a testar foi resolvido e implementado na linguagem Java, sendo possível a alteração de determinados valores como distâncias e limites das juntas nas tabelas localizadas à direita da tela.

Finalmente, na parte inferior constam os tipos de efetuador final que podem ser aplicados.

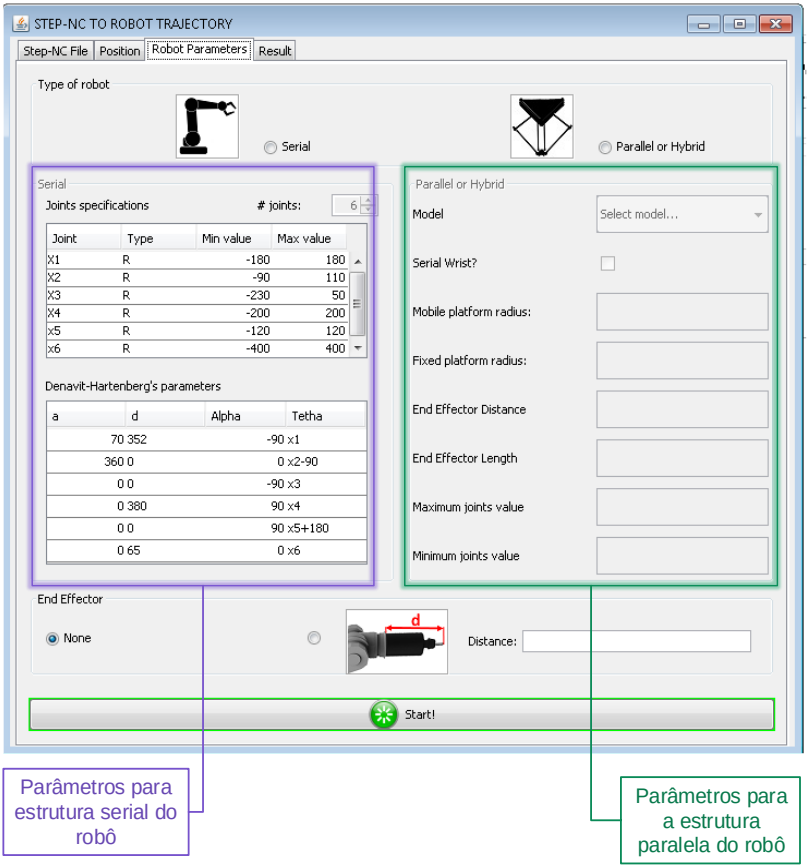


Figura 4.4. Entrada de parâmetros do robô.
FONTE: Da autora

Uma vez introduzidos os parâmetros pode-se iniciar o método proposto e obter os pontos da trajetória em função das juntas do robô.

Como o sistema computacional implementado foi programado na linguagem Java, ele pode ser executado em qualquer sistema operacional. Neste trabalho o sistema auxilia na validação do método proposto, além de permanecer disponível para a realização de trabalhos acadêmicos e servir como base para a criação de sistemas computacionais mais complexos.

4.2 Recursos utilizados nos testes

No que se refere aos testes e avalição dos resultados é necessário definir os recursos que foram aplicados, desde as ferramentas computacionais utilizadas até os robôs e as peças que foram utilizadas nas simulações. As características de todos esses elementos são especificadas nesta seção.

4.2.1 Software utilizado nos testes e simulações

Além do sistema computacional implementado, para a execução dos testes foram utilizados outros programas de suporte, tanto para a avaliação numérica do método implementado, quanto para as simulações dos robôs no ambiente de trabalho. Cabe destacar que todas essas ferramentas foram avaliadas em um computador com as características apresentadas na Tabela 4.1.

Tabela 4.1. Especificações do computador utilizado

Característica	**Valor**
Computador	DELL XPS 14z
Processador	Intel core i5-2430M
Velocidade do processador	2.40 GHz
Memória RAM	4.00 Gb
Sistema operacional	Windows 7 service pack 1
Tipo de sistema	64 bits

Para a execução das simulações foi utilizado o software Delmia (Dassault Systemes, 2014), e para a avaliação numérica e suporte em geral foram utilizados os programas Matlab versão R2013b e Microsoft Excel 2013.

Segundo a Dassault Systemes (2014), o *software* Delmia é uma solução computacional focada na produção e manufatura virtual, a qual inclui um módulo simulação de robôs industriais. Este programa foi escolhido principalmente pelo seu caráter genérico, o que oferece algumas vantagens como: grande quantidade de robôs disponíveis no ambiente virtual, disponibilidade de pós-processadores para um grande número de linguagens proprietárias de robôs e flexibilidade para a entrada de dados do processo a ser avaliado, como mencionado por Pan et al. (2012). Neste trabalho a versão utilizada foi a v5-6R2015.

Para a integração do sistema desenvolvido com o software Delmia podem ser utilizadas duas opções: introduzir manualmente as configurações das juntas para cada ponto ou criar um arquivo XML com uma estrutura aceita pelo Delmia, contendo todas as informações dos pontos da trajetória e as configurações das juntas do robô.

Com o intuito de facilitar o carregamento dos pontos, o arquivo XML foi o método escolhido, para o qual foi criada uma rotina de programação que gerasse diretamente o arquivo XML requerido, dentro do sistema desenvolvido. A Figura 4.5 mostra a aba adicionada na interface gráfica para gerar o arquivo e mostrar os resultados.

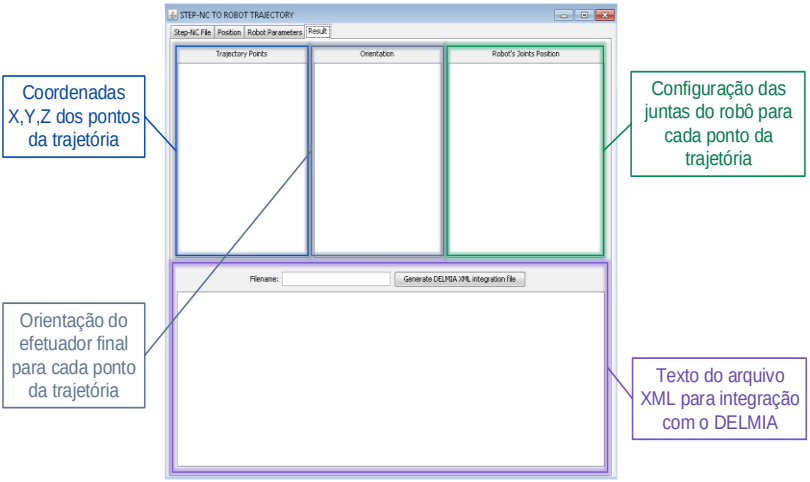


Figura 4.5. Geração de arquivo XML para integração com o Delmia.
FONTE: Da autora

Na programação da rotina especializada na geração do arquivo XML, foi utilizado o recurso JAXB, abreviatura para Arquitetura Java para enlaces XML (em inglês: *Java Architecture for XML Binding*), com o qual é possível acessar e escrever arquivos XML mediante a linguagem Java.

No caso do software Delmia é possível encontrar o esquema do arquivo XML na pasta da instalação do próprio software. A localização padrão é:

C:\ProgramFiles\DassaultSystemes\B25\win_b64\startup\OLP\
XSchemas\Upload.xsd.

A partir deste esquema foram geradas classes em Java de acordo com o método citado por Ort; Mehta (2003), mediante a ferramenta *xjc*,

disponibilizada na pasta de instalação do *JDK*, na localização padrão: C:\Program Files\Java\jdk1.8.0_25\bin\xjc.exe.

Com os objetos gerados pode-se inserir as informações dos pontos na estrutura determinada no esquema requerido pelo software Delmia, da qual foram utilizados os elementos mostrados na Figura 4.6.

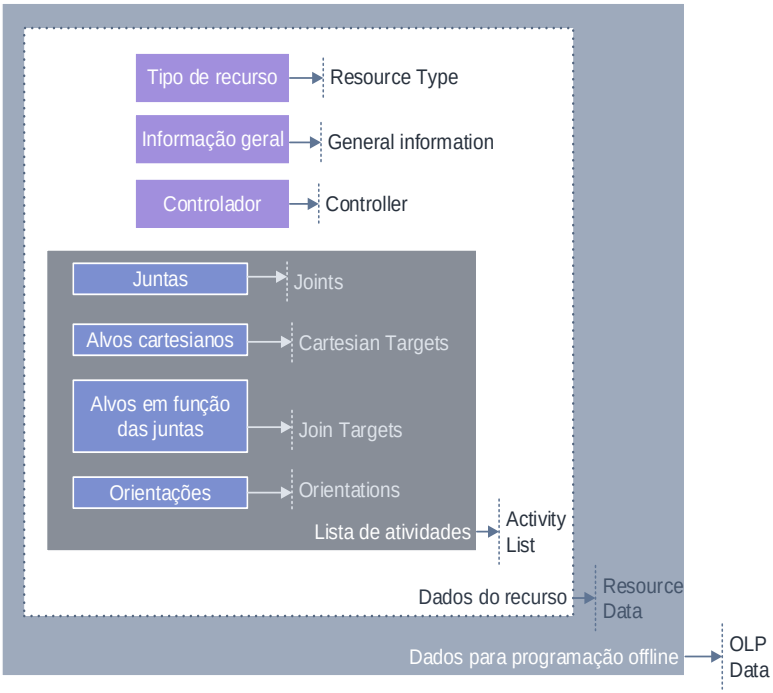


Figura 4.6. Elementos usados do esquema Upload.xsd.
FONTE: Da autora. Baseado em Dassault Systemes (2008)

Utilizando-se os recursos do *JAXB* gerou-se o arquivo XML que é reconhecido pelo software Delmia e permite carregar as informações da trajetória gerada em um só passo, utilizando-se a opção: Importar programa (*Import program*). Esse arquivo deve ser gerado para cada trajetória expressa em função das juntas do robô, simulada no ambiente virtual.

4.2.2 Robôs empregados nas simulações

Como mencionado anteriormente, no catálogo do Delmia existem vários tipos de robôs. No escopo deste trabalho foram utilizados três robôs para testar o método desenvolvido: dois robôs seriais de diferentes fabricantes e morfologias, e um robô do tipo híbrido. Esses robôs foram escolhidos por seu uso na indústria e por suas diferenças morfológicas, as quais deverão auxiliar na avaliação do método proposto.

Os parâmetros utilizados para a modelagem de cada robô são apresentados nas seções a seguir.

4.2.2.1 Adept One

O robô Adept 1 do fabricante Adept Technology é um robô tipo SCARA de quatro graus de liberdade, com três juntas rotativas e uma junta prismática, projetado principalmente para aplicações de posicionamento de peças (Adept Technology Inc., 1997).

A Figura 4.7 apresenta as dimensões principais do robô Adept 1, as quais foram utilizadas na obtenção dos parâmetros para a aplicação do método Denavit-Hartenberg, mostrados na Tabela 4.2.

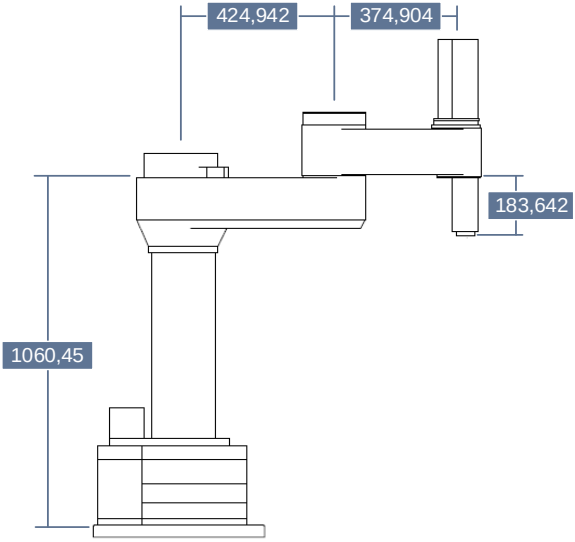


Figura 4.7. Robô Adept One.
NOTA: Distâncias em mm
FONTE: Adaptado de Adept Technology Inc. (1997)

Tabela 4.2. Parâmetros DH do Adept One

Junta	a_{DH} [mm]	d_{DH} [mm]	α_{DH} [°]	θ_{DH} [°]
1	424,942	1060,45	0	θ_1
2	374,904	0	180	θ_2
3	0	183,642	0	0
4	0	d_3	0	θ_4

4.2.2.2 ABB IRB-140

O robô IRB-140 é um braço robótico antropomórfico da ABB, com seis graus de liberdade, controlados mediante seis juntas rotativas. As principais aplicações deste robô são montagem, pintura, manuseio de materiais, rebarbação entre outras, conforme ABB (2004). As principais dimensões e os parâmetros cinemáticos utilizados são apresentados na Figura 4.8 e na Tabela 4.3, respectivamente.

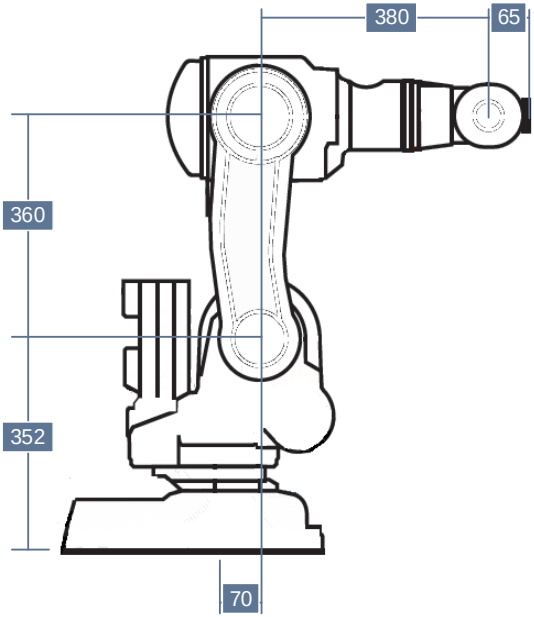


Figura 4.8. Robô ABB IRB-140.
NOTA: Distâncias em mm
FONTE: Adaptado de ABB (2004)

Tabela 4.3. Parâmetros DH doABB IRB -140

Junta	**a_{DH} [mm]**	**d_{DH} [mm]**	**α_{DH} [°]**	**θ_{DH} [°]**
1	70	358	-90	θ_1
2	360	0	0	θ_2-90
3	0	0	-90	θ_3
4	0	380	90	θ_4
5	0	0	90	θ_5+180
6	0	65	0	θ_6

4.2.2.3 COMAU Tricept 806

No catálogo do Delmia da versão utilizada existem dois robôs híbridos robustos: o Tricept 805 e Tricept 806, de cinco e seis graus de liberdade respectivamente, da empresa Comau. A diferença entre estes modelos está no punho utilizado em cada modelo, mas a estrutura paralela é igual, motivo pelo qual o robô Tricept 806 foi escolhido para os testes. Este robô é utilizado principalmente para operações de usinagem. Ele possui uma estrutura paralela com três juntas prismáticas controláveis e uma estrutura serial formada por um punho esférico, como mostrado na Figura 4.9.

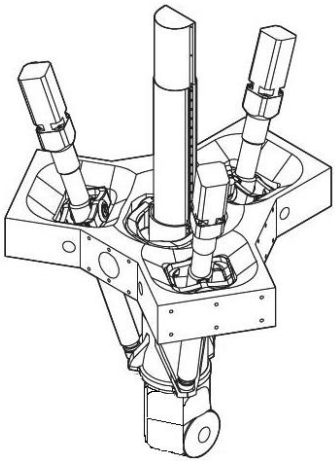


Figura 4.9. Robô Tricept 806.
FONTE: Adaptado de Lotura (2009)

Sendo o Tricept 806 um robô híbrido, é necessário conhecer os parâmetros tanto da estrutura paralela como da cadeia serial. Para obter o modelo da estrutura paralela, ela foi analisada mediante vetores e a origem *O* foi situada no centro da base fixa do robô. Os pontos de união entre a base fixa e os elos da plataforma estão localizados com uma separação de 120° numa circunferência de raio 500 mm, cujo centro coincide com a origem, como mostrado na Figura 4.10. O ponto A por exemplo está localizado no ponto [0,500,0] mm.

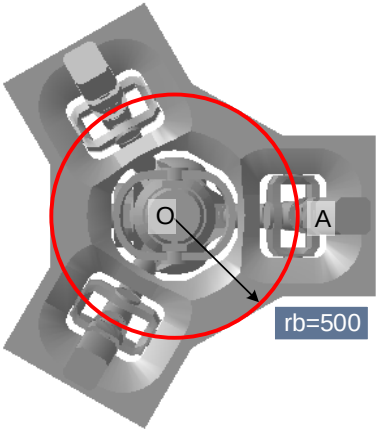


Figura 4.10. Base fixa do Tricept 806.
NOTA: Distâncias em mm
FONTE: Da autora

Depois foram estabelecidos os principais vetores que auxiliam na análise da plataforma paralela, como mostrado na Figura 4.11 para o caso da junta prismática entre A e B. Assim, o vetor $\overrightarrow{OA}$ é determinado usando-se a equação (31).

$$\overrightarrow{OA} = [500 \quad 0 \quad 0] \tag{31}$$

Segundo o método proposto, a posição do punho $\overrightarrow{OW}$ é determinada pela equação (32), equivalente à equação (27) citada no capítulo 3.

$$\overrightarrow{OW} = \overrightarrow{p_e} - effl * \overrightarrow{a_e} \tag{32}$$

A partir da posição do punho pode-se determinar o vetor $\overrightarrow{OC}$ mediante a equação (33).

$$\overrightarrow{OC} = \left(\left\|\overrightarrow{OW}\right\| - deff\right) * \frac{\overrightarrow{OW}}{\left\|\overrightarrow{OW}\right\|} \tag{33}$$

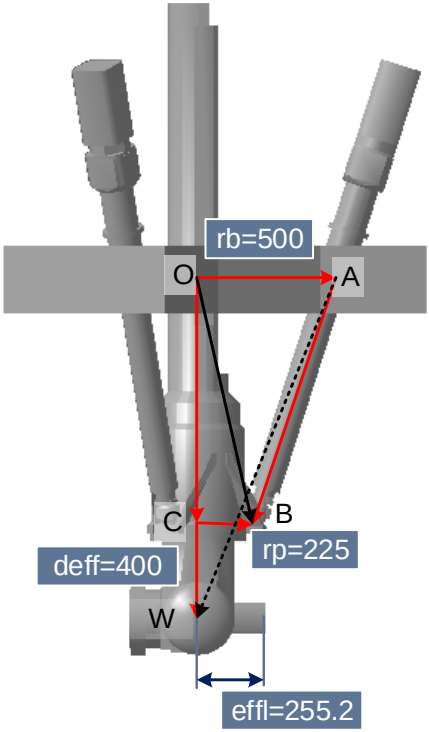


Figura 4.11. Plataforma paralela do Tricept 806.
NOTA: Distâncias em mm.
FONTE: Da autora

Observando-se novamente a estrutura paralela foram estabelecidas algumas relações entre os vetores, as quais são representadas pelas equações (34) a (38).

$$\overrightarrow{CB} = \overrightarrow{OB} - \overrightarrow{OC} \tag{34}$$

$$\overrightarrow{AW} = \overrightarrow{OW} - \overrightarrow{OA} \tag{35}$$

$$\overrightarrow{BW} = -\overrightarrow{CB} + \overrightarrow{CW} \tag{36}$$

$$\overrightarrow{OC} \perp \overrightarrow{CB} \Rightarrow \overrightarrow{OC} \cdot \overrightarrow{CB} = 0 \tag{37}$$

$$\left\| \overrightarrow{OB} - \overrightarrow{OC} \right\| = r_p \tag{38}$$

Além disso, deve-se observar que os vetores são coplanares, então tem-se a equação (39).

$$\det[\, \overrightarrow{CB} \quad \overrightarrow{OC} \quad \overrightarrow{OA} \,] = 0 \tag{39}$$

Resolvendo o sistema de equações composto por (37), (38) e (39) foi encontrado o vetor $\overrightarrow{OB}$, a partir do qual é possível calcular $\overrightarrow{AB}$ usando-se a equação (40).

$$\overrightarrow{AB} = \overrightarrow{AW} - \overrightarrow{BW} \tag{40}$$

O módulo do vetor $\overrightarrow{AB}$ é o comprimento da junta prismática localizada entre A e B necessário para atingir a posição desejada. O processo foi repetido para cada junta prismática, obtendo-se assim a solução para a estrutura paralela.

O modelo correspondente à parte paralela foi inserido diretamente no código no sistema computacional desenvolvido. No caso da estrutura serial, o modelo foi obtido mediante o método Denavit-Hartenberg (DH) com os parâmetros obtidos do punho esférico, mostrados na Tabela 4.4.

Tabela 4.4. Parâmetros DH para a estrutura serial do Tricept 806

Junta	$\mathbf{a_{DH}}$ **[mm]**	$\mathbf{d_{DH}}$ **[mm]**	$\boldsymbol{\alpha_{DH}}$ **[°]**	$\boldsymbol{\theta_{DH}}$ **[°]**
1	0	0	-90	θ_1+90
2	0	0	90	θ_2-90
3	0	effl	0	θ_3+180

4.2.3 Peças a serem executadas

Com o intuito de testar o método e os algoritmos propostos foram utilizados os arquivos físicos aderentes ao padrão STEP-NC de duas peças, descritas nas seções a seguir.

4.2.3.1 Exemplo 1 do padrão STEP-NC

Este primeiro arquivo é apresentado no texto da norma ISO 14649-11 (2002) e tem sido utilizado para testes básicos da maioria dos trabalhos focados na aplicação do padrão STEP-NC, devido à sua simplicidade.

Esta peça contém três *features*, as quais são usinadas mediante cinco *workingsteps*, e todas as operações são executadas em uma só face. O arquivo físico descreve uma peça com uma cavidade retangular e um furo circular, além de executar previamente o faceamento da peça, como mostrado na Figura 4.12.

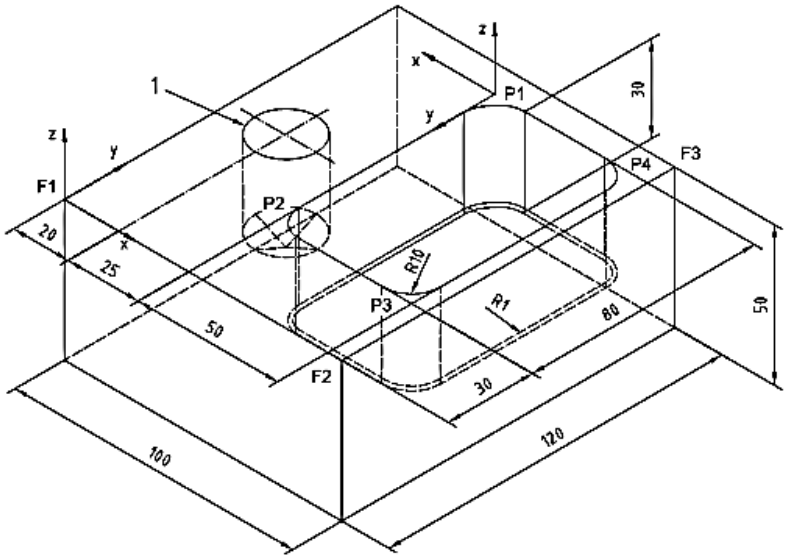


Figura 4.12. Peça exemplo 1.
NOTA: Distâncias em mm. FONTE: ISO 14649-11 (2002)

As informações do arquivo físico podem ser conferidas na norma ISO 14649-11 (2002).

4.2.3.2 Exemplo 2 gerado com o *software* STEP-MODELER

Com o objetivo de testar uma peça mais complexa e com operações executadas em outras faces da peça, foi projetada uma segunda peça no software STEP Modeler. Essa peça também contém cavidades retangulares e furos, mas estas *features* foram posicionadas em três faces da peça (Figura 4.13), o que permite avaliar o desempenho do método proposto em diferentes planos e usando robôs com diferentes graus de liberdade, os quais poderiam ou não atingir esses planos, dependendo da sua morfologia.

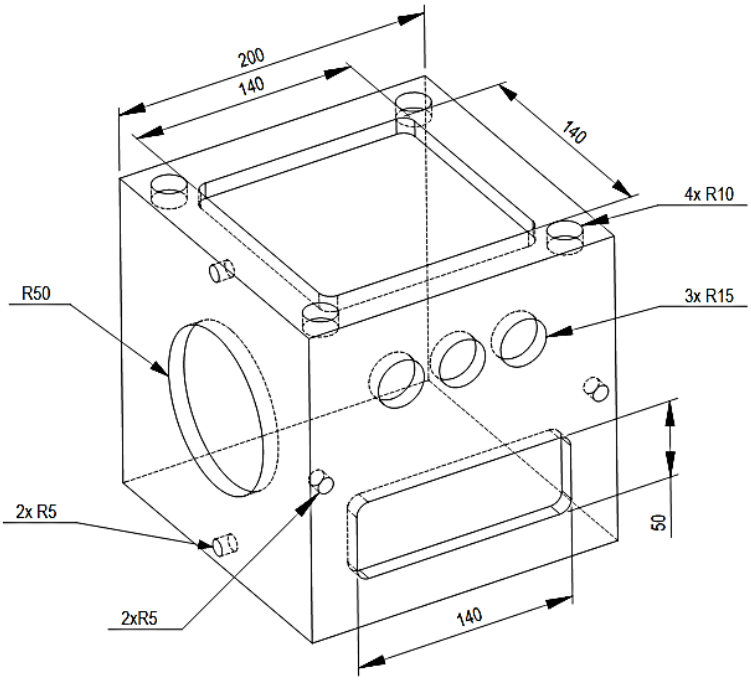


Figura 4.13. Peça exemplo 2.
NOTA: Distâncias em mm. FONTE: Da autora

As informações do arquivo físico correspondente à peça exemplo 2 podem ser conferidas no Apêndice B deste trabalho.

4.3 Métodos de avaliação da solução proposta

Nas seções anteriores foram descritos os recursos necessários para os testes do método proposto: o sistema computacional, os robôs e as peças a serem usinadas. Nesta seção será detalhado o procedimento para validar o método proposto e avaliar sua exatidão referente aos dados do arquivo STEP-NC de cada peça.

Deve-se lembrar que o escopo deste trabalho abrange desde a construção dos algoritmos do método proposto até os testes no ambiente virtual. Assim, os dados considerados nesta avaliação serão os obtidos das simulações dos processos de usinagem das peças já mencionadas.

4.3.1 Avaliação no ambiente virtual

A primeira fase da avaliação do método foi a simulação do processo de usinagem no entorno virtual. Dentro do entorno do *software* Delmia, no ambiente de programação *off-line*, colocaram-se os robôs a usar. Um modelo da peça final também foi inserido no ambiente virtual. Com os elementos já posicionados, o sistema computacional desenvolvido gerou a trajetória de usinagem a partir do arquivo STEP-NC correspondente à peça, bem como o arquivo XML, o qual foi carregado no robô simulado no software Delmia.

A simulação do processo de usinagem foi executada permitindo-se avaliar de forma preliminar a correspondência entre a trajetória obtida e a peça final. Pode-se avaliar também a trajetória gerada e o percurso que o robô deve seguir para usinar a peça, verificando-se a correspondência entre a localização dos elementos de trabalho no ambiente virtual e as posições atingidas pelo robô.

4.3.2 Avaliação numérica da trajetória gerada

Na segunda fase da avaliação do método foram obtidos indicadores da exatidão da trajetória gerada em relação à peça ideal detalhada no arquivo físico.

Foram geradas as trajetórias de usinagem no sistema computacional desenvolvido, para as duas peças consideradas e os robôs definidos, e foram coletados os dados das posições das juntas do robô correspondentes a cada ponto da trajetória.

Esses dados, juntamente com o modelo do robô correspondente, foram utilizados para obter as coordenadas dos pontos realmente

alcançados pelo robô, considerando-se que o cálculo da cinemática inversa poderia ter introduzido erros.

Estes cálculos foram executados mediante uma rotina de programação que foi criada para a resolução da cinemática direta no *software* Matlab, obtendo-se a trajetória realmente percorrida pelo robô, a qual foi comparada com a trajetória ideal calculada para cada uma das *features*, determinando-se o erro entre elas. Assim, conseguiu-se julgar se a peça usinada no ambiente virtual cumpre com a tolerância especificada no arquivo físico STEP-NC correspondente.

5. RESULTADOS E DISCUSSÃO

No capítulo anterior foi apresentada a forma de interpretação dos dados do arquivo físico STEP-NC, o cálculo e construção das trajetórias, e também foi descrito o método para solucionar a cinemática inversa para cada tipo de robô. Neste capítulo são apresentados os resultados obtidos a partir das simulações da usinagem das duas peças exemplo. Em primeiro lugar serão mostrados os resultados da simulação do processo e a correspondência visual entre as trajetórias geradas e as peças, como uma verificação preliminar. Posteriormente será apresentada a avaliação numérica feita sobre as trajetórias, comparando-se a trajetória ideal com a trajetória efetivamente percorrida e o erro entre elas.

5.1 Avaliação no ambiente virtual

A primeira etapa da avaliação das trajetórias consistiu na observação preliminar da correspondência geométrica entre a peça e a trajetória gerada. Para cada robô, a trajetória foi gerada usando-se o sistema computacional desenvolvido com os parâmetros cinemáticos apresentados na seção 4.2.2 e o arquivo físico STEP-NC correspondente a cada peça.

Os parâmetros de posição que foram aplicados em cada caso são mostrados na Tabela 5.1. Estes parâmetros foram inseridos na interface gráfica mostrada anteriormente na Figura 4.3.

No caso do robô Adept, a peça exemplo 2 precisou ser usinada em três etapas. A face XY foi usinada normalmente, mas devido às limitações cinemáticas do tipo de robô, a peça precisou ser girada para completar a usinagem nas faces laterais XZ e ZY, de maneira que o efetuador final do robô sempre conseguisse usinar a peça perpendicularmente ao plano de trabalho.

Uma vez ingressados os parâmetros de posição assim como os parâmetros cinemáticos do robô usado, as trajetórias foram geradas com o sistema computacional desenvolvido. O tempo que levou a geração das trajetórias em cada caso é mostrado na Tabela 5.2.

Com a trajetória gerada, o arquivo XML com a trajetória em função das juntas foi gerado e carregado no software Delmia. Um modelo da peça final foi inserido no ambiente virtual e posicionado de acordo com os parâmetros apresentados anteriormente. No caso do robô Adept, a peça foi posicionada segundo os parâmetros estabelecidos para cada face a usinar, respeitando as rotações mencionadas anteriormente.

Tabela 5.1 Parâmetros de posição usados

Robô		**Adept One**			
Peça exemplo		**1**	**2 (XY)**	**2 (XZ)**	**2 (ZY)**
Robô - mesa de trabalho	Distância em X	300	300	300	300
	Distância em Y	300	300	300	300
	Distância em Z	745	650	745	745
Mesa de trabalho - Peça	Distância em X	0	0	0	0
	Distância em Y	10	10	10	10
	Distância em Z	0	0	0	0
	Rotação em X	0	0	-90	0
	Rotação em Y	0	0	0	90
	Rotação em Z	0	0	0	0
Robô		**IRB-140**		**Tricept 806**	
Peça exemplo		**1**	**2**	**1**	**2**
Robô - mesa de trabalho	Distância em X	400	450	50	200
	Distância em Y	-200	0	50	300
	Distância em Z	450	350	-1900	-1900
Mesa de trabalho - Peça	Distância em X	0	0	0	0
	Distância em Y	10	10	10	10
	Distância em Z	0	0	0	0
	Rotação em X	0	0	0	0
	Rotação em Y	0	0	0	0
	Rotação em Z	0	0	0	0

NOTA 1: Distâncias em mm, ângulos em graus.
NOTA 2: Os sistemas de referência usados podem ser conferidos na Figura 3.10

Tabela 5.2 Tempo registrado para geração das trajetórias

Peça exemplo	**Quant. Pontos**	**Tempo [s]**		
		Adept One	**IRB-140**	**Tricept 806**
1	842	335	427	121
2	3106	784 (521+123+140)	1085	211

NOTA: Tempo obtido no IDE Netbeans durante a execução do sistema computacional desenvolvido

Finalmente, a simulação foi executada para cada robô e cada peça, observando-se que a operação de usinagem respeite a orientação desejada no efetuador final, traçando o caminho percorrido pelo efetuador.

Os resultados para as simulações correspondentes à peça exemplo 1 são apresentados na Figura 5.1 até a Figura 5.3.

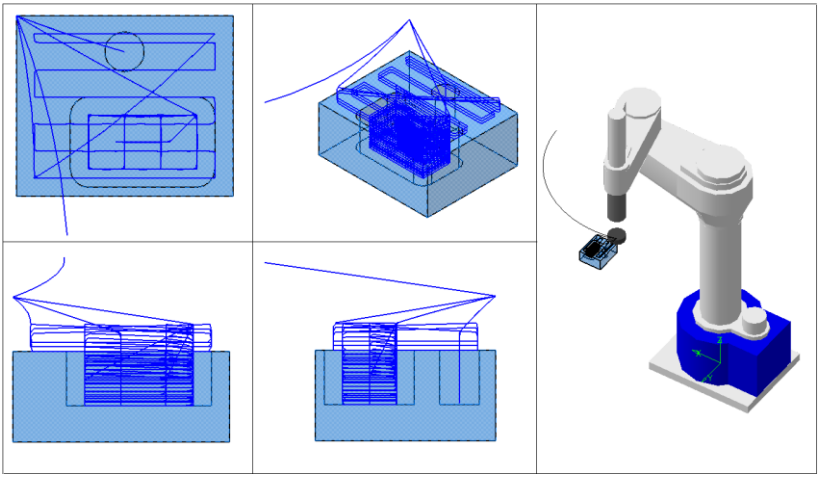


Figura 5.1 Simulação peça exemplo 1 - Adept One
FONTE: Da autora

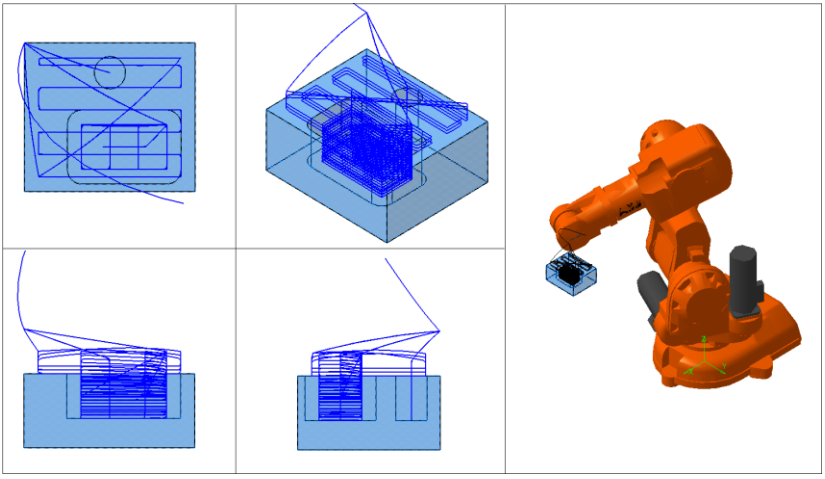


Figura 5.2 Simulação peça exemplo 1 - ABB IRB-140
FONTE: Da autora

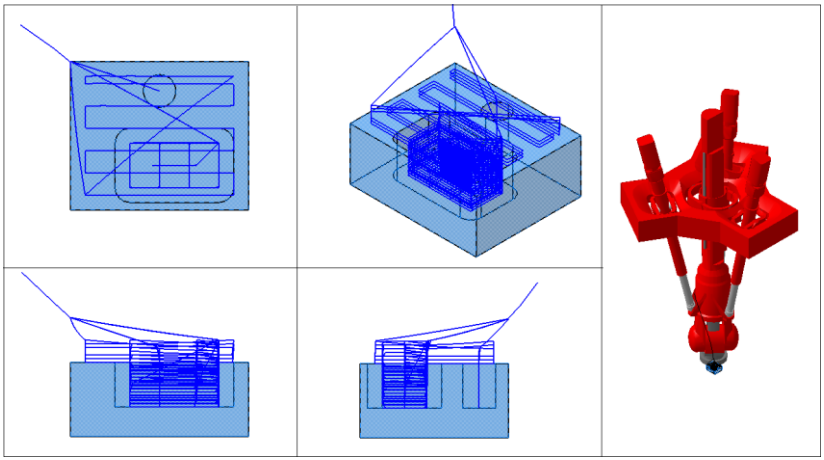


Figura 5.3 Simulação peça exemplo 1 - Tricept 806
FONTE: Da autora

Nestas simulações conseguiu-se observar que o efetuador final de cada robô manteve a orientação desejada, traçando a trajetória sempre perpendicular à face a ser usinada. Também, constatou-se que a trajetória foi executada de maneira contínua, partindo de um ponto no plano de retração definido. As trajetórias claramente correspondem às *features* resultantes do processo de usinagem, como no caso da cavidade com perfil circular, em que a trajetória fica visivelmente no centro do perfil. Como na peça exemplo 1 a usinagem é feita somente em uma face da peça, não são observadas correções drásticas na orientação das juntas do robô, mantendo-se um percurso sem desvios repentinos.

Os resultados para a peça exemplo 2 são apresentados na Figura 5.4 até a Figura 5.6.

Na execução da simulação observou-se que, no caso dos robôs IRB-140 e Tricept 806, foram necessárias algumas correções drásticas na orientação das juntas, como o caso detalhado na secção 3.2.3.4, pelo qual essas mudanças na trajetória foram executadas, como esperado, fora da peça, retomando-se a trajetória com a nova orientação. A peça 2 permitiu a visualização da usinagem em diferentes faces em um único um *setup*, o que ressalta a vantagem da utilização de robôs com 6 graus de liberdade para este tipo de peça. As trajetórias em todas as faces coincidem visualmente com as *features* resultantes da usinagem, e isso pode ser conferido com maior facilidade devido ao maior tamanho da peça. A orientação do efetuador final correspondeu à orientação desejada durante toda a trajetória.

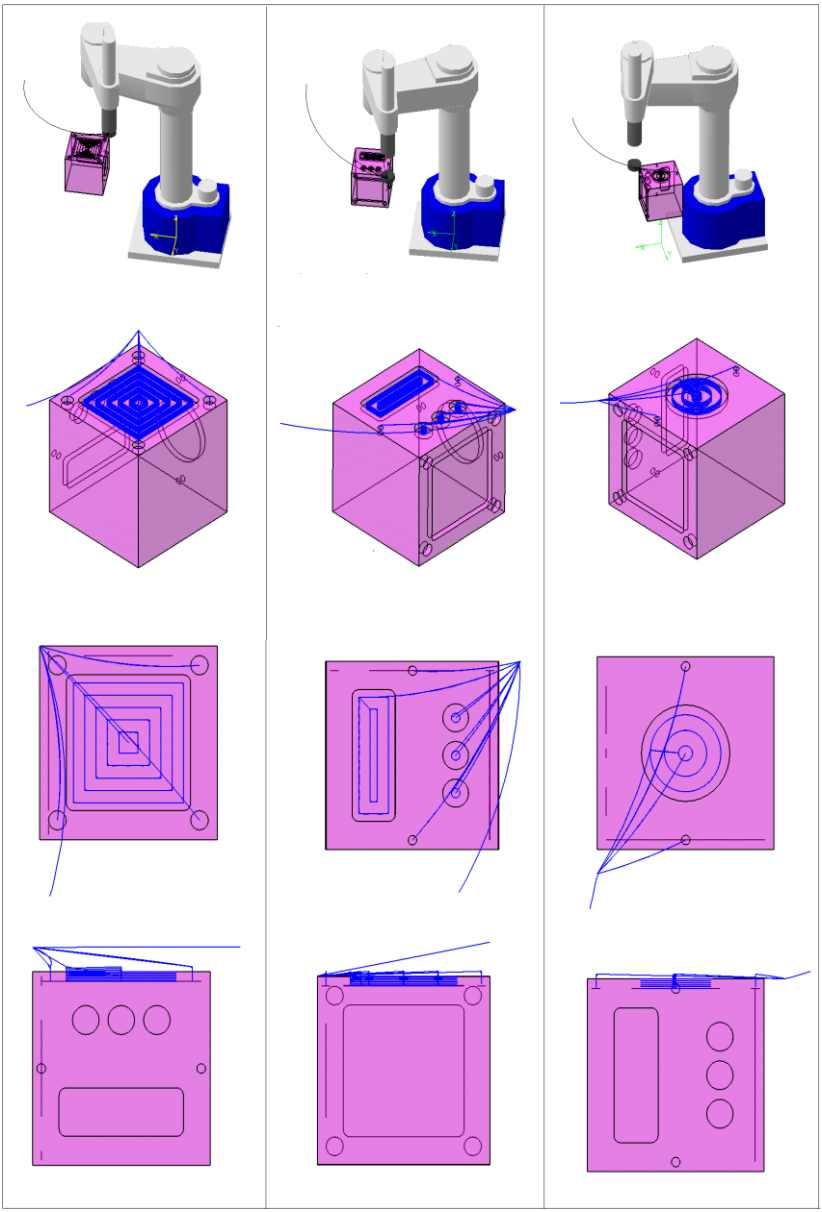


Figura 5.4 Simulação peça exemplo 2 - Adept One
FONTE: Da autora

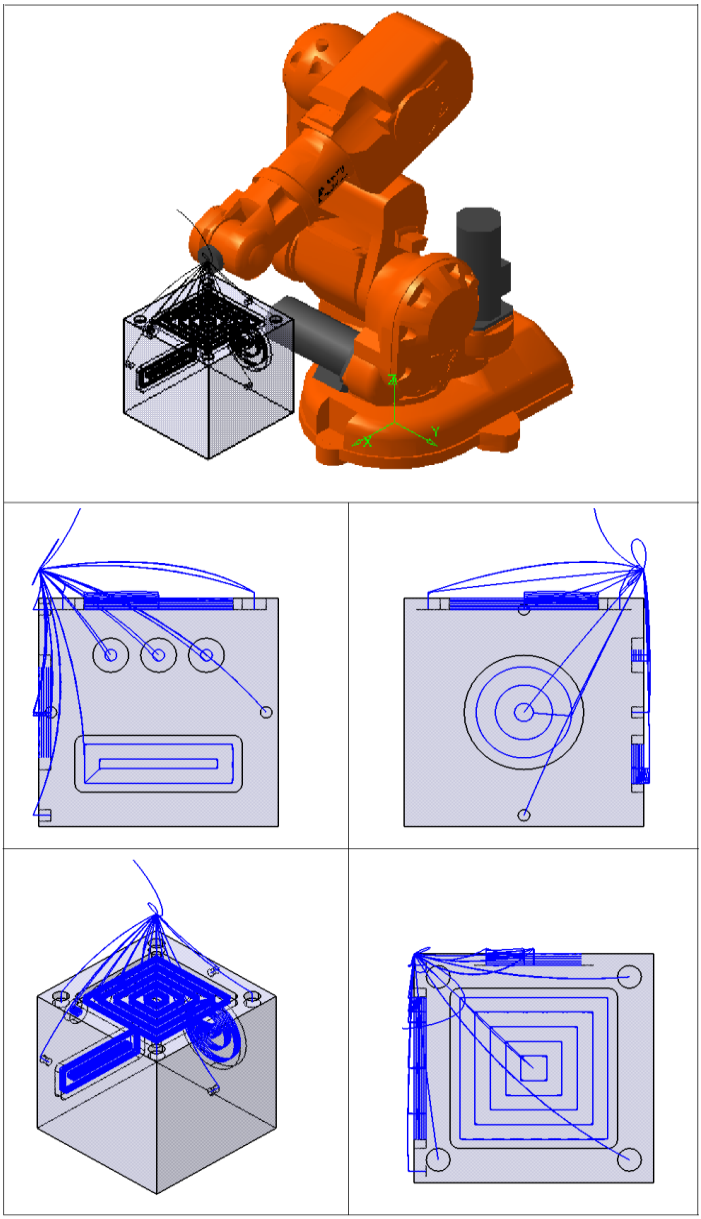


Figura 5.5 Simulação peça exemplo 2 - ABB IRB-140
FONTE: Da autora

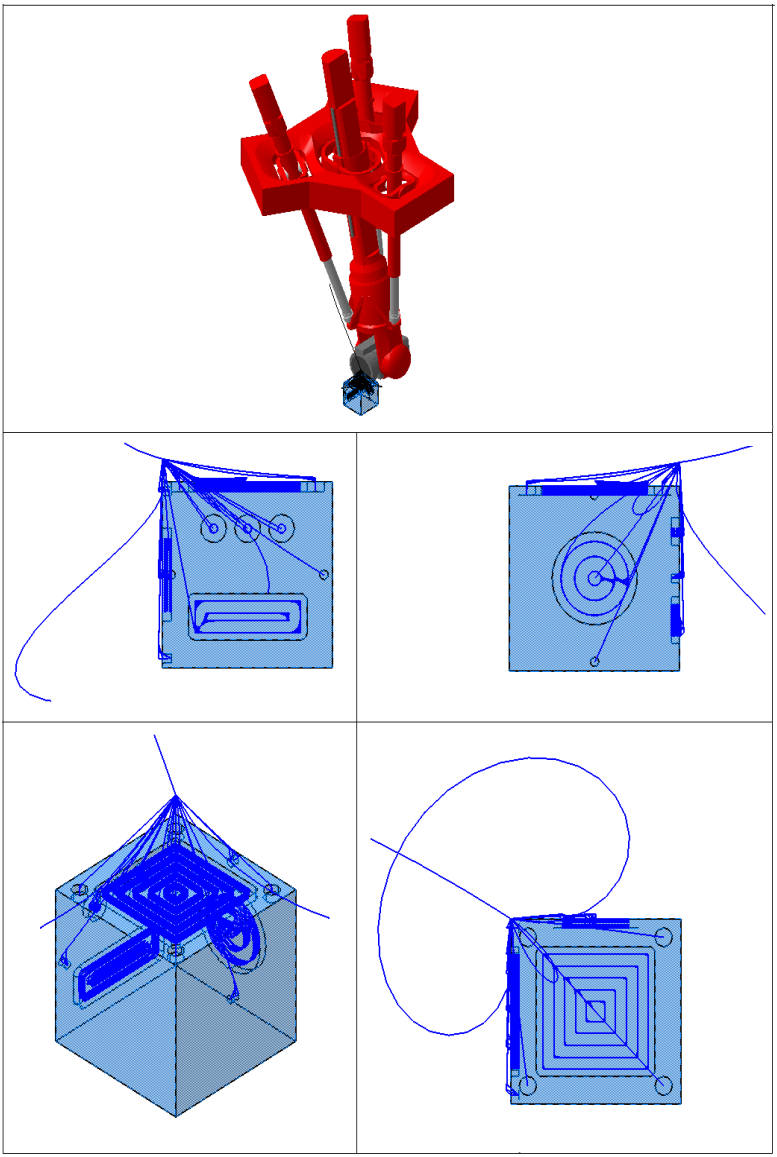


Figura 5.6 Simulação peça exemplo 2 - Tricept 806
Fonte: Da autora

Com a avaliação dos processos no ambiente virtual conseguiu-se concluir que:

- As trajetórias geradas respeitam os parâmetros de posicionamento definidos previamente;
- Os caminhos gerados correspondem às *features* que serão geradas nas peças;
- O método proposto é capaz de dar uma solução para a trajetória, ainda se o robô precisa ajustar drasticamente a orientação das suas juntas entre um ponto e outro do caminho definido, sem afetar o processo de usinagem. Este evento pode ser identificado claramente na vista superior da peça na Figura 5.6, onde o efetuador final do robô traça um laço fora da peça para ajustar a configuração das juntas em um ponto da trajetória e conseguir passar para o ponto seguinte, sem alterar a trajetória na peça.
- Com o método proposto é possível gerar uma trajetória para usinagem em diferentes faces em um único *setup*, sempre que usado um robô com a morfologia adequada.
- O método proposto consegue gerar uma trajetória adequada para cada tipo de robô em função das suas juntas, trajetória essa que é efetivamente percorrida durante a execução no ambiente virtual.

5.2 Avaliação numérica da trajetória

Para se ter uma melhor apreciação da exatidão da trajetória gerada, foi feita uma avaliação numérica usando os arquivos fornecidos pelo sistema desenvolvido e o software Matlab para o cálculo da cinemática direta. A comparação foi feita entre a trajetória alvo, calculada com base no perfil das *features* a serem usinadas, e a trajetória expressa em termos das juntas do robô, para conferir a correspondência entre elas com uma margem de erro menor do que a tolerância definida no arquivo físico STEP-NC.

Como a trajetória final é expressa em termos das juntas do robô, foi necessária a resolução da cinemática direta para cada um. No caso dos robôs seriais, as matrizes resultantes do método Denavit-Hartenberg foram utilizadas. Os valores das juntas foram substituídos e, assim, a posição do efetuador final foi obtida dos primeiros três elementos da quarta coluna da matriz final de cada robô, como explicado na seção 2.2.3.2.

No caso do robô Tricept 806, sendo um robô com estrutura híbrida, foi utilizado um modelo CAD da estrutura paralela para determinar a posição do centro da plataforma móvel, como mostrado na Figura 5.7.

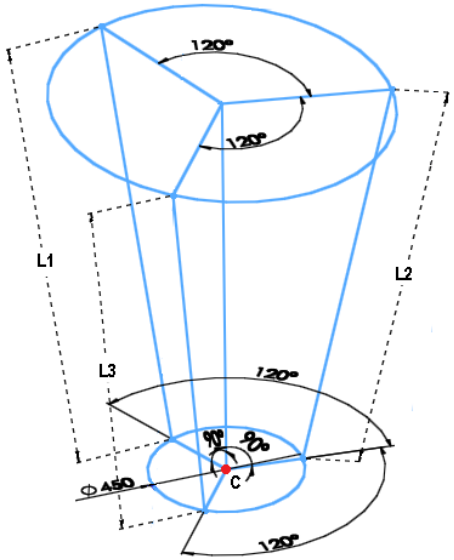


Figura 5.7 Modelo CAD da estrutura paralela do robô Tricept 806
FONTE: Da autora

Os valores das juntas prismáticas foram substituídos nas dimensões L1, L2 e L3 correspondentemente, obtendo-se a posição do ponto C e a orientação da estrutura paralela, com as quais pode-se obter a matriz T_e^w. Além disso, os valores das juntas que compõem o punho esférico foram substituídos na matriz resultante do método Denavit-Hartenberg da parte serial, obtendo-se T_w^0 e, mediante a equação (29), a posição do efetuador é obtida finalmente. Este método foi utilizado para agilizar a resolução da estrutura paralela, obtendo-se uma única solução, pois geralmente a cinemática direta do Tricept é resolvida mediante a utilização de métodos numéricos.

Com a cinemática direta resolvida, as trajetórias foram calculadas no software Matlab e comparadas com as trajetórias alvo, como mostrado na Figura 5.8 para o caso da peça exemplo 1 executada com o robô IRB-140. Nesta figura, o caminho vermelho representa a trajetória realmente percorrida, enquanto o caminho verde representa a trajetória ideal calculada com base no perfil das *features*. As trajetórias resultaram visivelmente coincidentes ao serem desenhadas, o qual pode ser considerado um resultado preliminar qualitativo. Porém, para obter o resultado quantitativo foi calculado o erro entre elas.

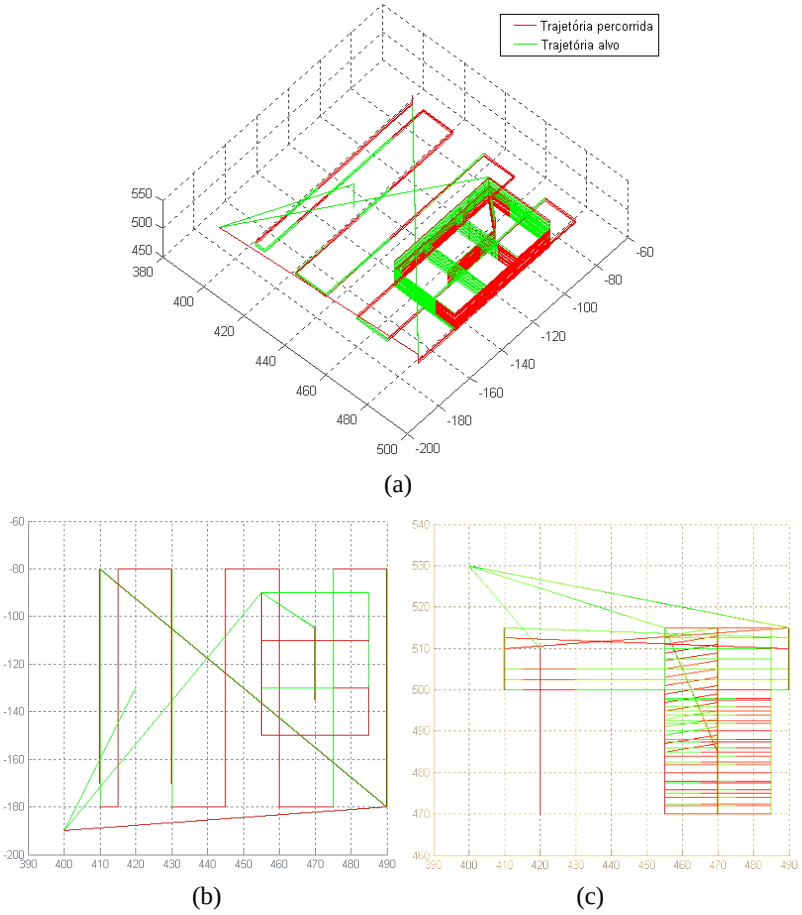


Figura 5.8 Trajetória percorrida e trajetória alvo para a peça exemplo 1, no caso do robô ABB IRB-140: (a) vista isométrica, (b) vista superior, (c) vista frontal.
FONTE: Da autora

Na Figura 5.9 até a Figura 5.14 são apresentados os gráficos obtidos da dispersão do erro de cada ponto da trajetória em cada eixo. Os gráficos mostram o valor do erro em cada componente da posição, para todos os pontos da trajetória gerada. Os valores obtidos podem ser conferidos no Apêndice C.

As trajetórias geradas para peça exemplo 1 com o robô Adept One (Figura 5.9) e o robô antropomórfico IRB-140 (Figura 5.10), mantêm um

comportamento uniforme em todos os pontos, resultando em um valor de erro médio muito próximo de zero.

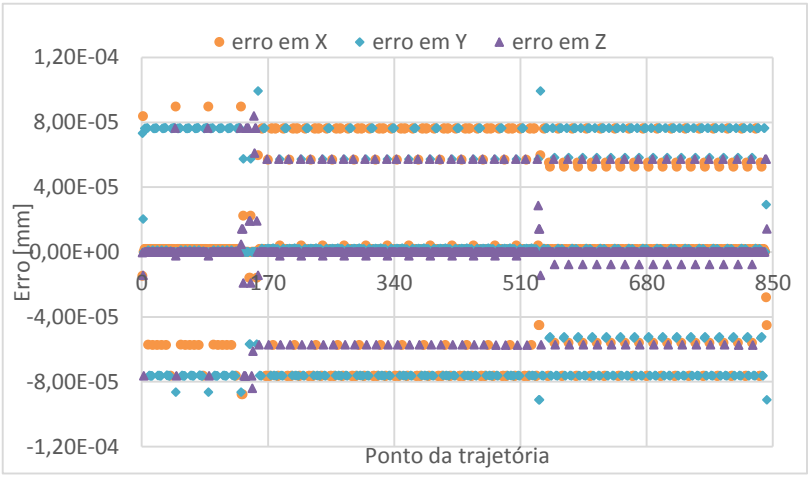


Figura 5.9. Erro da trajetória peça exemplo 1 - robô Adept One
FONTE: Da autora

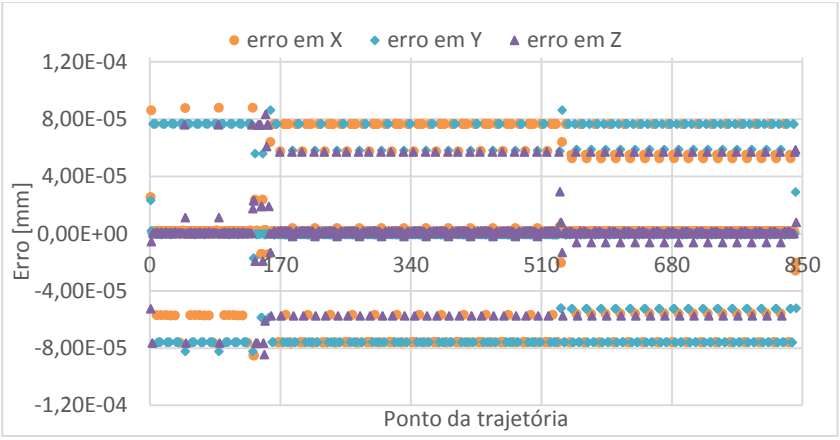


Figura 5.10. Erro da trajetória peça exemplo 1 - robô IRB-140
FONTE: Da autora

Nas trajetórias para a peça 2 com o robô Adept One (Figura 5.11) e o robô IRB-140 (Figura 5.12) pode-se notar uma diferença entre as trajetórias geradas nas faces XY (pontos 1-1015), XZ (pontos 1016-2257)

e ZY (pontos 2258-3106), apresentando um erro médio maior nas últimas duas faces e um comportamento menos uniforme na dispersão do erro.

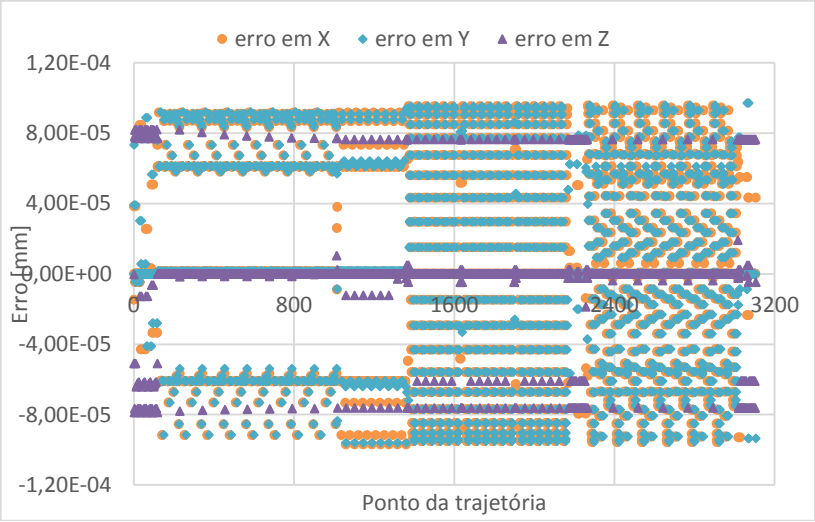


Figura 5.11. Erro da trajetória peça exemplo 2 - robô Adept One
FONTE: Da autora

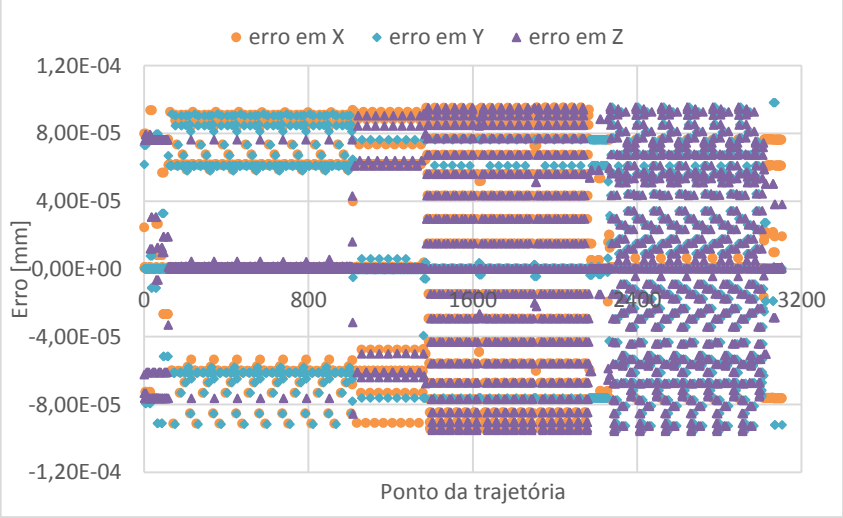


Figura 5.12. Erro da trajetória peça exemplo 2 - robô IRB-140
FONTE: Da autora

No caso do robô Tricept 806, para a trajetória da peça exemplo 1 (Figura 5.13) o erro está uniformemente distribuído nos três eixos e perto dos limites estabelecidos no algoritmo de cinemática inversa.

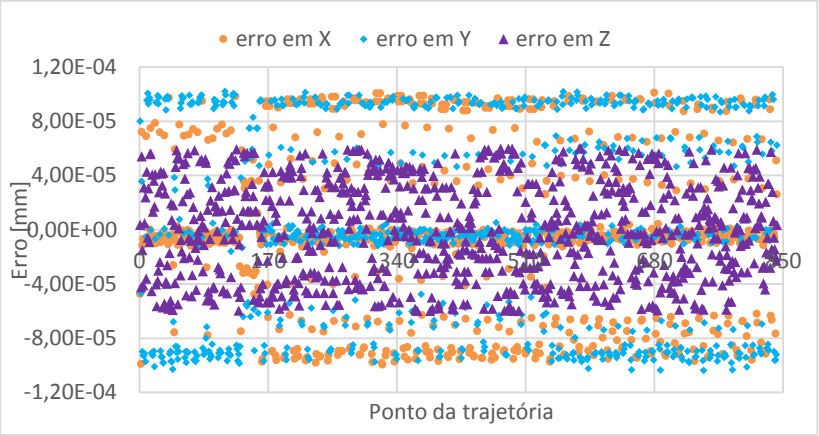


Figura 5.13. Erro da trajetória peça exemplo 1 - robô Tricept 806
FONTE: Da autora

Para a peça exemplo 2 (Figura 5.14), existem pontos cujo erro no eixo Z é maior do que os limites estabelecidos, e menores nos eixos X e Y para os pontos da face XZ (pontos 1016-2257).

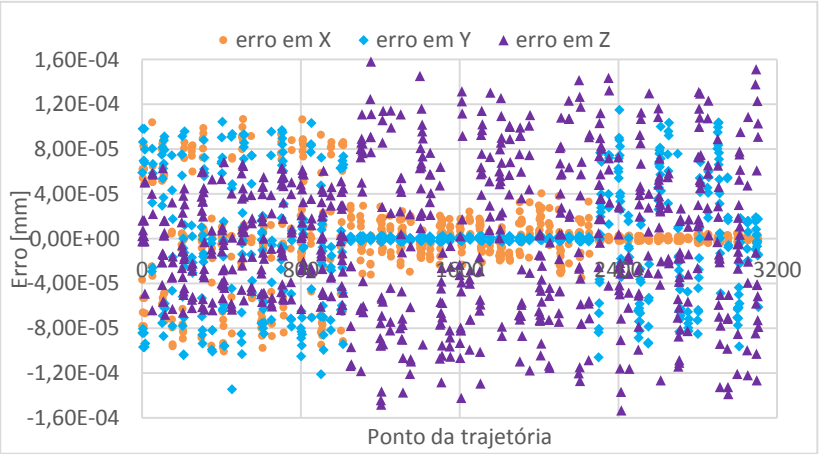


Figura 5.14. Erro da trajetória peça exemplo 2 - robô Tricept 806
FONTE: Da autora

Numericamente, o erro da trajetória no caso dos robôs seriais (Adept One e IRB-140) ficou dentro do intervalo de ±0,0001 mm (0,1 μm), como foi definido no método de resolução da cinemática inversa. Para o robô Tricept o erro final fica no intervalo de ±0,00016 mm (0,16 μm). Isso ocorre porque no robô Tricept o erro final abrange o erro da parte paralela, o erro da estrutura serial e o erro inserido no método de desacoplamento cinemático empregado para ligar a solução das duas partes do robô.

A Tabela 5.3 resume os valores mais importantes desta análise.

Tabela 5.3 Erro nas trajetórias

Robô		**Adept One**		**IRB-140**		**Tricept 806**	
Peça Exemplo		**1**	**2**	**1**	**2**	**1**	**2**
X	Erro médio [μm]	0,001	0,002	0,001	0,003	0,000	0,001
	Limite superior do erro [μm]	0,090	0,096	0,088	0,095	0,101	0,107
	Limite inferior do erro [μm]	-0,088	-0,097	-0,085	-0,095	-0,099	-0,101
Y	Erro médio [μm]	-0,003	0,003	-0,004	0,002	0,001	-0,001
	Limite superior do erro [μm]	0,099	0,097	0,086	0,098	0,102	0,115
	Limite inferior do erro [μm]	-0,091	-0,096	-0,082	-0,096	-0,104	-0,135
Z	Erro médio [μm]	0,000	-0,001	0,001	0,001	0,001	-0,001
	Limite superior do erro [μm]	0,084	0,082	0,083	0,095	0,060	0,158
	Limite inferior do erro [μm]	-0,084	-0,079	-0,085	-0,096	-0,060	-0,153

Os erros obtidos em todos os casos foram muito menores do que a tolerância especificada no arquivo físico STEP-NC das peças trabalhadas, as quais foram ±10μm para a peça exemplo 1, e ±50μm para a peça exemplo 2.

5.3 Discussão

O objetivo principal desta dissertação foi o desenvolvimento de um método que permitisse a geração de trajetórias para vários tipos de robôs industriais, a partir das informações contidas em um arquivo aderente ao

padrão STEP-NC. Considerando-se os resultados apresentados nas seções anteriores, pode-se concluir que o método proposto gerou trajetórias para os três robôs diferentes com um erro menor do que a tolerância requerida nas especificações das peças propostas.

Contudo, a correta geração dessas trajetórias depende da determinação adequada dos parâmetros cinemáticos e de localização dos elementos no ambiente de trabalho. No caso da localização, as trajetórias sempre serão geradas com referência na peça, depois giradas na ordem sequencial guinada-arfagem-rolagem (*yaw-pitch-roll*) e depois trasladadas de acordo com os parâmetros especificados. A correta interpretação destes parâmetros fornecerá uma trajetória espacialmente bem posicionada. Quanto mais precisos forem os parâmetros cinemáticos, melhor será o resultado da trajetória em função das juntas, pois um erro em um parâmetro cinemático poderia fornecer uma trajetória diferente da desejada.

Outro fator a ser considerado é o erro definido no método de cinemática inversa. Como neste trabalho foi considerada uma tolerância bilateral, uma tolerância unilateral também poderia ser especificada modificando-se o valor fixado no algoritmo de resolução da cinemática inversa. Adicionalmente, deve ser considerado o número de iterações permitidas, pois para uma solução da cinemática inversa muito próxima de uma singularidade, ou então mais exata, é necessário um maior número de iterações ou o resultado não poderia ser encontrado. Porém, um aumento no número de iterações afetaria o desempenho do algoritmo e resultaria em um tempo elevado para o cálculo das trajetórias.

Por outro lado, os resultados obtidos mostram que o método proposto é capaz de gerar trajetórias dentro do intervalo de tolerância para diversos tipos de robôs a partir do arquivo físico STEP-NC. Nos robôs seriais o cálculo é direto a partir dos parâmetros cinemáticos requeridos pelo método Denavit-Hartenberg. Para robôs híbridos utiliza-se o método de desacoplamento cinemático, podendo-se usar o algoritmo para a parte serial juntamente com o modelo da parte paralela adequada para o robô analisado.

O algoritmo proposto e os resultados obtidos mostram que a geração de trajetórias para robôs industriais, baseadas nas informações do arquivo físico STEP-NC, é possível sempre que o modelo do robô seja corretamente detalhado e tenha como base um método de resolução de cinemática inversa capaz de respeitar os limites das juntas e um desempenho tal que permita obtenção de resultados dentro da margem do erro em reduzidas iterações. O método de mínimos quadrados amortecidos apresentado por Na; Yang; Jia (2008), com as adaptações

feitas neste trabalho, mostrou ser efetivo na solução da cinemática inversa dentro do limite de 100 iterações por ponto, fornecendo como resultado uma configuração adequada ao limite das juntas do robô.

Uma vantagem do método proposto é a capacidade de gerar as trajetórias para diferentes faces de uma peça de trabalho em somente um *setup*, dependendo dos graus de liberdade do robô utilizado, reduzindo o tempo de fabricação da peça. Em outros trabalhos esta caraterística não é explorada, e poderia ser de grande utilidade para diminuir os erros na usinagem ao usar-se uma menor quantidade de *setups*.

Adicionalmente, diferentemente do trabalho de Solvang; Refsahl; Sziebig (2009), o método proposto nesta dissertação é independente de um software específico, pois a trajetória gerada em termos das juntas pode ser utilizada em diversas ferramentas de simulação. O software Delmia foi usado neste trabalho unicamente para executar as simulações devido a possuir os modelos dos robôs usados. Assim, o método pode ser aplicado em vários tipos de robôs, independente do fabricante e da topologia, desde que seja modelado adequadamente.

Em futuros trabalhos, a modelagem dos robôs poderia ser obtida mediante outros métodos como os helicoides sucessivos descrito por Tsai (1999), ou utilizando-se um modelo padronizado sob o padrão STEP como sugerido no trabalho de Xiao; Huan; Dong (2014). Porém, o método Denavit-Hartenberg aplicado nos robôs seriais conduziu a bons resultados, como descrito neste trabalho.

O método desenvolvido deve ser aprimorado de maneira a incluir robôs redundantes e robôs cooperativos, evitar colisões e otimizar as trajetórias, levando o método a um controlador tipo 3, como sugerido por Suh et al. (2003).

6. CONCLUSÕES E SUGESTÕES PARA TRABALHOS FUTUROS

Neste trabalho foi proposto um método para a geração de trajetórias para robôs industriais baseado nas informações contidas em um arquivo físico no padrão STEP-NC, para o fresamento de peças prismáticas. Um sistema computacional foi desenvolvido para gerar as trajetórias automaticamente, as quais foram simuladas em um ambiente virtual para a sua avaliação. A primeira avaliação consistiu na observação da simulação para determinar a correspondência entre a peça e a trajetória, assim como garantir a correta orientação do efetuador final durante a sua execução. Em uma etapa posterior, a trajetória foi avaliada numericamente obtendo-se o erro entre a trajetória ideal e a trajetória gerada em termos das juntas dos robôs.

A partir dos resultados obtidos são feitas as seguintes conclusões:

- O método proposto consegue gerar uma trajetória capaz de executar o processo de fresamento especificado no arquivo físico STEP-NC para robôs de diferentes tipos e morfologias, sem incorrer em custos adicionais derivados da atualização de hardware ou software proprietário.
- O algoritmo, baseado no método de mínimos quadrados amortecidos, aplicado na solução da cinemática inversa, forneceu bons resultados em termos de erro, número de iterações requeridas, prevenção de singularidades e consideração dos limites de movimento das juntas. Todas as soluções obtidas para a trajetória requerida respeitaram as limitações cinemáticas de cada tipo de robô.
- Apesar de não ter sido obtido um método geral para a resolução de cinemática inversa dos robôs paralelos, o desacoplamento cinemático proposto permite a integração do método de resolução para robôs seriais, com uma solução analítica para robôs paralelos, o que facilita a resolução dos robôs com estrutura híbrida, como no caso do robô Tricept analisado.
- A rotina estabelecida para a leitura e interpretação dos dados do arquivo STEP-NC mostrou ser efetiva na extração dos dados mínimos requeridos para a reconstrução das *features* a serem usinadas, as especificações das ferramentas e as operações envolvidas no processo. O *schema* compilado permite a interpretação das entidades contidas nas partes 10, 11, 111 da norma ISO 14649 sem inconvenientes, e o

interpretador desenvolvido pode ser estendido para incluir novas entidades e atributos, utilizando as classes já programadas.

- Ao gerar as trajetórias ideais em primeiro lugar e independente do algoritmo de solução da cinemática inversa, o método facilitou sua aplicação em robôs de diferentes morfologias, fornecendo os pontos da trajetória em forma de coordenadas cartesianas que poderiam ser usados em outras máquinas de comando numérico. Além disso, ao gerar a trajetória inicialmente no plano XY e ter a capacidade de girá-la e trasladá-la, o método permitiu a geração das trajetórias para qualquer face da peça prismática com a mesma exatidão, respeitando-se os graus de liberdade do robô usado e os parâmetros de posicionamento estabelecidos pelo usuário.
- O sistema computacional desenvolvido facilitou o processo de geração de trajetórias. A interface gráfica simplificou a entrada de parâmetros cinemáticos e de posição, assim como a leitura do arquivo físico. Os arquivos fornecidos pelo sistema computacional foram vitais para a execução da análise numérica da trajetória. Adicionalmente, a rotina criada para a geração do arquivo XML possibilitou o carregamento acelerado dos pontos da trajetória no software Delmia, necessário para executar as simulações.
- Os testes realizados para duas peças diferentes e três robôs distintos forneceram os resultados esperados e mostraram a aplicabilidade do método para a geração de trajetórias com uma margem de erro menor do que 1μm, para robôs não redundantes do tipo serial e híbridos do tipo Tricept. A execução das simulações permitiu determinar se as orientações do efetuador final e os parâmetros de posição foram realmente respeitados, facilitando a avaliação do método proposto em diferentes cenários, aproveitando-se as vantagens da programação *off-line*.

A seguir são apresentadas algumas sugestões para trabalhos futuros:

- Executar testes fora do ambiente virtual, isto é, utilizando robôs industriais reais.
- Estudar e aplicar outros métodos para a resolução da cinemática inversa e o modelamento do robô como a utilização de quatérnios e quatérnios duais, com o intuito de evitar singularidades no método proposto e reduzir o custo computacional.
- Ampliar a solução para incluir robôs com redundância cinemática;

- Incluir outras estratégias para a geração de trajetórias, como a apresentada por Pobożniak (2013) sobre trajetórias trocoidais no contexto do padrão STEP-NC;
- Aumentar as funcionalidades do sistema computacional, adicionando estratégias para evitar colisões com outros elementos do entorno de trabalho e a capacidade de sugerir o posicionamento adequado do robô com referência à peça, no entorno de trabalho.
- Expandir o método proposto de maneira a considerar os aspectos dinâmicos dos robôs;
- Adotar o modelo para troca de dados entre sistemas CAx e sistemas de programação *off-line* aderente ao padrão STEP, proposto por Xiao; Huan; Dong (2014) para a entrada dos parâmetros cinemáticos do robô, no método proposto;
- Atualizar o sistema computacional desenvolvido, adicionando opções de simulação e pós-processamento próprias, usando bibliotecas como as fornecidas pelo *software* de simulação e programação *off-line* RoboDK (RoboDK, 2015) ou pelo conjunto de ferramentas orientadas à robótica *Robotic toolkit* do Matlab.

REFERÊNCIAS

ABB. **IRB 140 Industrial Robot.** (Apresenta principais aplicações do robô). Disponível em: <http://www.feng.pucrs.br/~fkuhne/files/sistrob_em/notasdeaula/IRB 140 PR10031EN R10.pdf>. Acesso em: 10/10/2014.

ADEPT TECHNOLOGY INC. **Instruction Handbook AdeptOne-MV AdeptThree-MV Robot**. San Jose: ADEPT Technology, 1997. 183 p.

ADEPT TECHNOLOGY INC. **Portfolio of Industrial Intelligent Robotics.** (Apresenta catálogo de produtos). Disponível em: <http://www.adept.com/products/robots>. Acesso em: 10/10/2014.

ALLEN, R.; ROSSO JR, R.; NEWMAN, S. AB-CAM: an agent-based methodology for the manufacture of STEP compliant feature based components. In: International Conference on Metal Cutting and High Speed Machining, 3, 2001, **Anais...** Bath: Univesity of Bath, 2001. p. 351–362.

ANG, M. H.; WEI, L.; YONG, L. S. An industrial application of control of dynamic behavior of robots-a walk-through programmed welding robot. In: International Conference on Robotics and Automation, 2000, **Anais...** San Francisco: IEEE, 2000. p. 2352–2357.

ARISTIDOU, A.; LASENBY, J. **Inverse kinematics: a review of existing techniques and introduction of a new fast iterative solver. CUED/f-INFENG/TR-632**. Cambridge: University of Cambridge, 2009. 60 p.

AXELCLK. **Symja.** (Código e documentação da biblioteca Symja-Android). Disponível em: <https://bitbucket.org/axelclk/symja_android_library/wiki/Home>. Acesso em: 17/8/2015.

BENAVENTE, J. C. T. **Um Sistema para o projeto e fabricação de peças mecânicas a distância via internet aderente à norma ISO 14649 (STEP-NC)**. 2011. 239 f. Tese (Doutorado em Engenharia Mecânica) - Universidade Federal de Santa Catarina, Florianópolis.

BENAVENTE, J. C. T.; FERREIRA, J. C. E. A Web-Based System for Mapping Features into ISO 14649-Compliant Machining Workingsteps. In: International Conference on Manufacturing Systems Engineering, 2013, **Anais...** Veneza: World Academy of Science, Engineering and Technology, 2013. p. 2171–2179.

BRUYNINCKX, H. Open robot control software: the OROCOS project. In: International Conference on Robotics and Automation, 2001, **Anais...** Seoul: IEEE, 2001. p. 2523–2528.

BUSS, S. **Introduction to inverse kinematics with jacobian transpose, pseudoinverse and damped least squares methods**. San Diego: University of California, 2004. 19 p.

CALABRESE, F.; CELENTANO, G. Design and realization of a STEP-NC compliant CNC embedded controller. In: 2007 IEEE Conference on Emerging Technologies & Factory Automation (EFTA 2007), 2007, **Anais...** IEEE, 2007. p. 1010–1017.

CHA, J.-M. et al. A roadmap for implementing new manufacturing technology based on STEP-NC. **Journal of Intelligent Manufacturing**. 2014. p.1–15. Disponível em: <http://link.springer.com/10.1007/s10845-014-0927-2>. Acesso em: 26/10/2015.

DASSAULT SYSTEMES. **Delmia Capabilities.** (Apresenta características do software Delmia). Disponível em: <http://www.3ds.com/products-services/delmia/capabilities/>. Acesso em: 16/11/2015.

DASSAULT SYSTEMES. Delmia v5-6R2015, arquivo: Upload.xsd. 2008.

DENG, S. et al. Application of robot offline programming in thermal spraying. **Surface and Coatings Technology**. , v. 206, n. 19-20, 2012. p.3875–3882. Disponível em: <http://linkinghub.elsevier.com/retrieve/pii/S0257897212002071>. Acesso em: 20/10/2015.

DOOLEY, J. R.; MCCARTHY, J. M. Spatial rigid body dynamics using dual quaternion components. In: Proceedings. 1991 IEEE International Conference on Robotics and Automation, 1991, **Anais...** IEEE Comput. Soc. Press, 1991. p. 90–95.

GHAZAEI, M. **Topics in Trajectory Generation for Robots**. 2015. 121 f. Tese (Doutorado em controle automático) - Lund University, Lund.

HARDWICK, M. et al. A roadmap for STEP-NC-enabled interoperable manufacturing. **The International Journal of Advanced Manufacturing Technology**. , v. 68, n. 5-8, 2013. p.1023–1037. Disponível em: <http://link.springer.com/10.1007/s00170-013-4894-0>. Acesso em: 12/2/2015.

HARIB, K. et al. Parallel, Serial and Hybrid Machine Tools and Robotics Structures: Comparative Study on Optimum Kinematic Designs. In: **Serial and Parallel Robot Manipulators – Kinematics, Dynamics, Control and Optimization**. Rijeka: InTECH, 2012. p.109–124.

HARTENBERG, R. S.; DENAVIT, J. A kinematic Notation for Lower Pair Mechanisms Based on Matrices. **J. Appl. Mech. ASME**. 1955. p.215–221. .

HEIDARI, O.; DANIALI, H. M.; VAREDI, S. M. Geometric design of 3R manipulators for three precision poses using dual quaternions. In: 2014 Second RSI/ISM International Conference on Robotics and Mechatronics (ICRoM), 2014, **Anais...** IEEE, 2014. p. 601–606.

HORSCH, T. **Introduction to Robotics: Module Trajectory generation and robot programming.** (Apresenta conceito básicos de robótica). Disponível em: <http://www.easy-rob.com/uploads/media/LectureRobotics.pdf>. Acesso em: 6/3/2016.

INTERNATIONAL FEDERATION OF ROBOTICS. **World Robotics 2015 Survey: Industrial robots are conquering the world.** (Apresenta estatísticas do uso dos robôs na indústria). Disponível em: <http://www.worldrobotics.org/uploads/tx_zeifr/Charts_PC_09_30_2015_01.pdf>. Acesso em: 26/11/2015.

ISAKSSON, M.; ERIKSSON, A.; NAHAVANDI, S. Analysis of the inverse kinematics problem for 3-DOF axis-symmetric parallel manipulators with parasitic motion. In: International Conference on Robotics and Automation, 2014, **Anais...** Hong Kong: IEEE, 2014. p. 5736–5743.

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION. **ISO 10303-11**: Industrial automation systems and integration -- Product data

representation and exchange -- Part 11: Description methods: The EXPRESS language reference manual. Genebra, 2004. 255 p.

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION. **ISO 10303-21**: Industrial Automation Systems and Integration - Product Data Representation and Exchange - Part 21: Implementation Methods: Clear Text Encoding of the Exchange Structure. Genebra, 2002. 72 p.

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION. **ISO 10303-224**: Industrial automation systems and integration — Product data representation and exchange — Part 224: Application protocol: Mechanical product definition for process planning using machining features. Genebra, 2006. 1062 p.

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION. **ISO 14649-1**: Industrial automation systems and integration — Physical device control — Data model for computerized numerical controllers — Part 10: General process data. Genebra, 2002. 36 p.

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION. **ISO 14649-10**: Industrial automation systems and integration -- Physical device control -- Data model for computerized numerical controllers -- Part 10: General process data. Genebra, 2002. 184 p.

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION. **ISO 14649-11**: Industrial automation systems and integration — Physical device control — Data model for computerized numerical controllers — Part 11: Process data for milling. Genebra, 2002. 84 p.

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION. **ISO 14649-111**: Industrial automation systems and integration — Physical device control — Data model for computerized numerical controllers — Part 111: Tools for milling machines. Genebra, 2002. 28 p.

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION. **ISO 2806**: Industrial automation systems -- Numerical control of machines -- Vocabulary. Genebra, 1994. 23 p.

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION. **ISO 6983**: Automation systems and integration -- Numerical control of machines

-- Program format and definitions of address words. Genebra, 2009. 26 p.

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION. **ISO 8373**: Robots and robotic devices -- Vocabulary. Genebra, 2012. 38 p.

KASSIM, N. et al. Information Structure of STEP Convertor of STEP-CNC Mapping System. **Applied Mechanics and Materials**. , v. 773-774, 2015. p.38–42. Disponível em: <http://www.scientific.net/AMM.773-774.38>. Acesso em: 5/5/2015.

KAYANI, J.; RICO, P. STEP Compliant CAD/CAM – Challenges and the Future. **The Open Automation and Control Systems Journal**. , v. 7, n. 1, 2015. p.1335–1342. Disponível em: <http://benthamopen.com/contents/pdf/TOAUTOCJ/TOAUTOCJ-7-1335.pdf>. Acesso em: 17/10/2015.

KENWRIGHT, B. **Dual-Quaternions From Classical Mechanics to Computer Graphics and Beyond.** (Artigo digital). Disponível em: <http://www.xbdev.net/misc_demos/demos/dual_quaternions_beyond/paper.pdf>. Acesso em: 7/3/2016.

KHALIL, W.; DOMBRE, E.; NAGURKA, M. **Modeling, Identification and Control of Robots**. Oxford: Butterworth-Heinemann, 2004. 500 p.

KOVÁCS, B.; SZAYER, G.; TAJTI, F. Design of a universal robot controller. **Periodica Polytechnica Mechanical Engineering**. Budapest , v. 55, n. 2, 2012. p.95–100. Disponível em: <http://www.pp.bme.hu/me/2011_2/me2011_2_06.html>. Acesso em: 29/1/2016.

KRŽIČ, P.; STOIC, A.; KOPAČ, J. STEP-NC: A new programming code for the CNC machines. **Strojniski Vestnik/Journal of Mechanical Engineering**. , v. 55, n. 6, 2009. p.406–417. Disponível em: <http://en.sv-jme.eu/scripts/download.php?file=/data/upload/2009/SV-6-09/Temp/Krzic_Kopac_zl.pdf>. Acesso em: 29/12/2014.

KUCUK, S.; BINGUL, Z. The inverse kinematics solutions of industrial robot manipulators. In: International Conference on Mechatronics, 2004, **Anais...** Istambul: IEEE, 2004. p. 274–279.

KUMAR, G.; SHARMA, N. K.; BHOWMICK, P. Creating Wheel-Thrown Potteries in Digital Space. In: **Arts and Technology**. Heidelberg: Springer, 2010. p.181–189.

LIN, C.; CHANG, P.; LUH, J. Formulation and optimization of cubic polynomial joint trajectories for industrial robots. **IEEE Transactions on Automatic Control**. , v. 28, n. 12, 1983. p.1066–1074. Disponível em: <http://ieeexplore.ieee.org/articleDetails.jsp?arnumber=1103181>. Acesso em: 6/3/2016.

LIN, J.; LIN, C.; LO, H. Pseudo-inverse Jacobian control with grey relational analysis for robot manipulators mounted on oscillatory bases. **Journal of Sound and Vibration**. , v. 326, n. 3-5, 2009. p.421–437. Disponível em: <http://linkinghub.elsevier.com/retrieve/pii/S0022460X09004829>. Acesso em: 22/4/2015.

LKSOFTWARE GMBH. **JSDAI.** (Apresenta informações, código e documentação da biblioteca JSDAI). Disponível em: <http://www.jsdai.net/news>. Acesso em: 2/6/2015.

LOTURA, J. **Working in cartesians: Tricept Transformation.** (Apresenta características do robo Tricept). Disponível em: <http://www.pkmtricept.com/productos/index.php?id=en&Nproduct=12404 15079>. Acesso em: 16/11/2015.

MARINHO, M. M. **Controle de robô para auxílio em cirurgia laparoscópica usando quatérnios duais**. 2014. 77 f. Dissertação (Mestrado em Engenharia em Sistemas Eletrônicos e de Automação) - Universidade de Brasília, Brasilia.

MEREDITH, M.; MADDOCK, S. **Real-time inverse kinematics: The return of the Jacobian. CS-04-06**. Sheffield: University of Sheffield, 2004. 1-15 p.

MERLET, J. P. **Parallel Robots**. Dordrecht: Springer Science & Business Media, 2006. 402 p.

MUKUNDAN, R. Quaternions: From Classical Mechanics to Computer Graphics, and Beyond. In: Proceedings of the 7 th Asian Technology Conference in Mathematics, 2002, **Anais...** Melaka, 2002. p. 97–106.

NA, M.; YANG, B.; JIA, P. Improved Damped Least Squares Solution with Joint Limits, Joint Weights and Comfortable Criteria for Controlling Human-like Figures. In: IEEE Conference on Robotics, Automation and Mechatronics, 2008, **Anais...** Chengdu: IEEE, 2008. p. 1090–1095.

NAGATA, F. et al. Development of CAM system based on industrial robotic servo controller without using robot language. **Robotics and Computer-Integrated Manufacturing**. , v. 29, n. 2, 2013. p.454–462. Disponível em: <http://linkinghub.elsevier.com/retrieve/pii/S0736584512001184>. Acesso em: 1/10/2015.

NEWMAN, S.; ALLEN, R.; ROSSO, R. CAD/CAM solutions for STEP-compliant CNC manufacture. **International Journal of Computer Integrated Manufacturing**. , v. 16, n. 7-8, 2003. p.590–597. Disponível em: <http://www.tandfonline.com/doi/abs/10.1080/0951192031000115688>. .

NGUYEN, V. K.; STARK, J. STEP-compliant CNC Systems, Present and Future Directions. In: **Advanced Design and Manufacturing Based on STEP**. London: Springer London, 2009. p.215–231.

ORT, E.; MEHTA, B. **Java Architecture for XML Binding (JAXB).** (Apresenta um tutoral para o uso da ferramenta JAXB). Disponível em: <http://www.oracle.com/technetwork/articles/javase/index-140168.html>. Acesso em: 6/11/2015.

PAN, Z. et al. Recent progress on programming methods for industrial robots. **Robotics and Computer-Integrated Manufacturing**. , v. 28, n. 2, 2012. p.87–94. Disponível em: <http://linkinghub.elsevier.com/retrieve/pii/S0736584511001001>. Acesso em: 18/7/2015.

PATRICK, D.; FARDO, S. **Industrial electronics: devices and systems**. CRC Press, 2000. 671 p.

POBOZNIAK, J. The Use of Step-NC (ISO 14649) for the Integration of CAD / CAM / CNC Chain. **Czasopismo Techniczne**. , v. 5, n. 1-M, 2013. p.333–339. Disponível em: <http://www.ejournals.eu/sj/index.php/Cz/article/view/1830/1851>. Acesso em: 30/12/2014.

POBOŻNIAK, J. ALGORITHM FOR ISO 14649 (STEP-NC) FEATURE RECOGNITION. **Management and Production Engineering Review**. , v. 4, n. 4, 2013. p.50–58. Disponível em: <http://www.degruyter.com/view/j/mper.2013.4.issue-4/mper-2013-0037/mper-2013-0037.xml>. Acesso em: 3/11/2015.

QI, L. et al. Virtual engineering: challenges and solutions for intuitive offline programming for industrial robot. In: 2008 IEEE Conference on Robotics, Automation and Mechatronics, 2008, **Anais...** Chengdu: IEEE, 2008. p. 12–17.

RADAVELLI, L. A. et al. Cinemática posicional de robôs via iteração e quatérnios. In: Proceeding Series of the Brazilian Society of Computational and Applied Mathematics, 2015, **Anais...** 2015.

RAUCH, M. et al. An advanced STEP-NC controller for intelligent machining processes. **Robotics and Computer-Integrated Manufacturing**. , v. 28, n. 3, 2012. p.375–384. Disponível em: <http://linkinghub.elsevier.com/retrieve/pii/S0736584511001335>. Acesso em: 23/11/2015.

RIDWAN, F.; XU, X. Advanced CNC system with in-process feed-rate optimisation. **Robotics and Computer-Integrated Manufacturing**. , v. 29, n. 3, 2013. p.12–20. Disponível em: <http://linkinghub.elsevier.com/retrieve/pii/S0736584512000543>. Acesso em: 1/12/2015.

ROBODK. **RoboDK Home.** (Apresenta informação geral do software e funcionalidades). Disponível em: <http://www.robodk.com/>. Acesso em: 11/11/2015.

ROCHA, C. R.; TONETTO, C. P.; DIAS, A. A comparison between the Denavit–Hartenberg and the screw-based methods used in kinematic modeling of robot manipulators. **Robotics and Computer-Integrated Manufacturing**. , v. 27, n. 4, 2011. p.723–728. Disponível em: <http://linkinghub.elsevier.com/retrieve/pii/S073658451100010X>. Acesso em: 30/11/2015.

SÄÄSKI, J.; SALONEN, T.; PARO, J. **Integration of CAD, CAM and NC**

with Step-NC. Helsinki: VTT Information Service, 2005. 23 p.

SCHENCK, D. A.; WILSON, P. R. **Information Modeling the EXPRESS Way**. Nova Iorque: Oxford University Press, 1993. 416 p.

SHUSTER, M. D. Survey of attitude representations. **Journal of the Astronautical Sciences**. , v. 41, 1993. p.439–517. Disponível em: <http://www.dimnp.unipi.it/gabiccini-m/RAR/Hidden/___RotationsSurvey.pdf>. Acesso em: 27/11/2015.

SICILIANO, B. Kinematic control of redundant robot manipulators: A tutorial. **Journal of Intelligent and Robotic Systems**. , v. 3, n. 3, 1990. p.201–212. Disponível em: <http://link.springer.com/10.1007/BF00126069>. Acesso em: 9/2/2015.

SICILIANO, B. The Tricept robot: Inverse kinematics, manipulability analysis and closed-loop direct kinematics algorithm. **Robotica**. , v. 17, n. 4, 1999. p.S0263574799001678. Disponível em: <http://www.journals.cambridge.org/abstract_S0263574799001678>. Acesso em: 26/6/2015.

SICILIANO, B. et al. **Robotics: Modelling, Planning and Control**. Springer Science & Business Media, 2010. 632 p.

SOLVANG, B.; REFSAHL, L.; SZIEBIG, G. STEP-NC based industrial robot CAM system. In: International Symposium on Robot Control, 9, 2009, **Anais...** Gifu: The International Federation of Automatic Control, 2009. p. 361–366.

SONG, W. et al. Research on multi-robot open architecture of an intelligent CNC system based on parameter-driven technology. **Robotics and Computer-Integrated Manufacturing**. , v. 28, n. 3, 2012. p.326–333. Disponível em: <http://linkinghub.elsevier.com/retrieve/pii/S0736584511001207>. Acesso em: 2/12/2015.

SOUZA, F. J. **Usinagem remota de peças prismáticas via internet em uma máquina cnc aderente ao padrão STEP-NC**. 2014. Dissertação (Mestrado em Engenharia Mecânica) - Universidade Federal de Santa Catarina, Florianópolis.

SPONG, M. W.; HUTCHINSON, S.; VIDYASAGAR, M. **Robot Modeling and Control**. 1º ed. Nova Iorque: Wiley, 2005. 496 p.

STRANDBERG, M. **Robot path planning: an object-oriented approach**. 2004. Teses (Doutorado em Sinais e sistemas) - Royal Institute of Technology.

SUH, S. H. et al. Architecture and implementation of a shop-floor programming system for STEP-compliant CNC. **Computer-Aided Design**. , v. 35, n. 12, 2003. p.1069–1083. Disponível em: <http://linkinghub.elsevier.com/retrieve/pii/S0010448502001793>. Acesso em: 20/10/2015.

SZIEBIG, G. **Man-machine and intermachine interaction in flexible manufacturing systems**. 2013. 159 f. Tese (Doutorado em Produção e Qualidade) - Norwegian University of Science and Technology, Trondheim.

TAIRA, Y.; SAGARA, S.; KATOH, R. Digital kinematic control of space robot manipulators using transpose of generalized Jacobian matrix. In: International Conference on Industrial Electronics, Control and Instrumentation, 2000, **Anais...** Nagoya: IEEE, 2000. p. 596–601.

TANEV, T. K. Kinematics of a hybrid (parallel–serial) robot manipulator. **Mechanism and Machine Theory**. , v. 35, n. 9, 2000. p.1183–1196. Disponível em: <http://linkinghub.elsevier.com/retrieve/pii/S0094114X99000737>. Acesso em: 7/9/2015.

TSAI, L.-W. **Robot Analysis: The Mechanics of Serial and Parallel Manipulators**. John Wiley & Sons, 1999. 505 p.

VALILAI, O.; NODEH, J.; HOUSHMAND, M. RoboCAD INFELT STEP, Interoperable Platform to Manage Collaboration among CAD and Robot Programming Agents Integrated Based on STEP (ISO 10303) Standard. In: World Congress on Engineering and Computer Science, 2010, **Anais...** San Francisco, 2010. p. 350–356.

VERSCHEURE, D. et al. Time-Optimal Path Tracking for Robots: A Convex Optimization Approach. **IEEE Transactions on Automatic Control**. , v. 54, n. 10, 2009. p.2318–2327. Disponível em:

<http://ieeexplore.ieee.org/articleDetails.jsp?arnumber=5256286>. Acesso em: 6/3/2016.

VICHARE, P. et al. A Unified Manufacturing Resource Model for representing CNC machining systems. **Robotics and Computer-Integrated Manufacturing**. , v. 25, n. 6, 2009. p.999–1007. Disponível em: <http://linkinghub.elsevier.com/retrieve/pii/S073658450900043X>. Acesso em: 1/12/2015.

XIAO, W. et al. A complete CAD/CAM/CNC solution for STEP-compliant manufacturing. **Robotics and Computer-Integrated Manufacturing**. , v. 31, 2015. p.1–10. Disponível em: <http://linkinghub.elsevier.com/retrieve/pii/S073658451400043X>. Acesso em: 2/12/2015.

XIAO, W.; HUAN, J.; DONG, S. A STEP-compliant Industrial Robot Data Model for robot off-line programming systems. **Robotics and Computer-Integrated Manufacturing**. , v. 30, n. 2, 2014. p.114–123. Disponível em: <http://linkinghub.elsevier.com/retrieve/pii/S0736584513000665>. Acesso em: 1/12/2015.

XU, X. W.; NEWMAN, S. T. Making CNC machine tools more open, interoperable and intelligent—a review of the technologies. **Computers in Industry**. , v. 57, n. 2, 2006. p.141–152. Disponível em: <http://linkinghub.elsevier.com/retrieve/pii/S0166361505001089>. Acesso em: 13/10/2015.

ZHAO, F.; XU, X.; XIE, S. STEP-NC enabled on-line inspection in support of closed-loop machining. **Robotics and Computer-Integrated Manufacturing**. , v. 24, n. 2, 2008. p.200–216. Disponível em: <http://linkinghub.elsevier.com/retrieve/pii/S0736584506001141>. Acesso em: 4/11/2015.

APÊNDICE A – Diagramas UML da ferramenta desenvolvida

Estrutura da ferramenta computacional desenvolvida

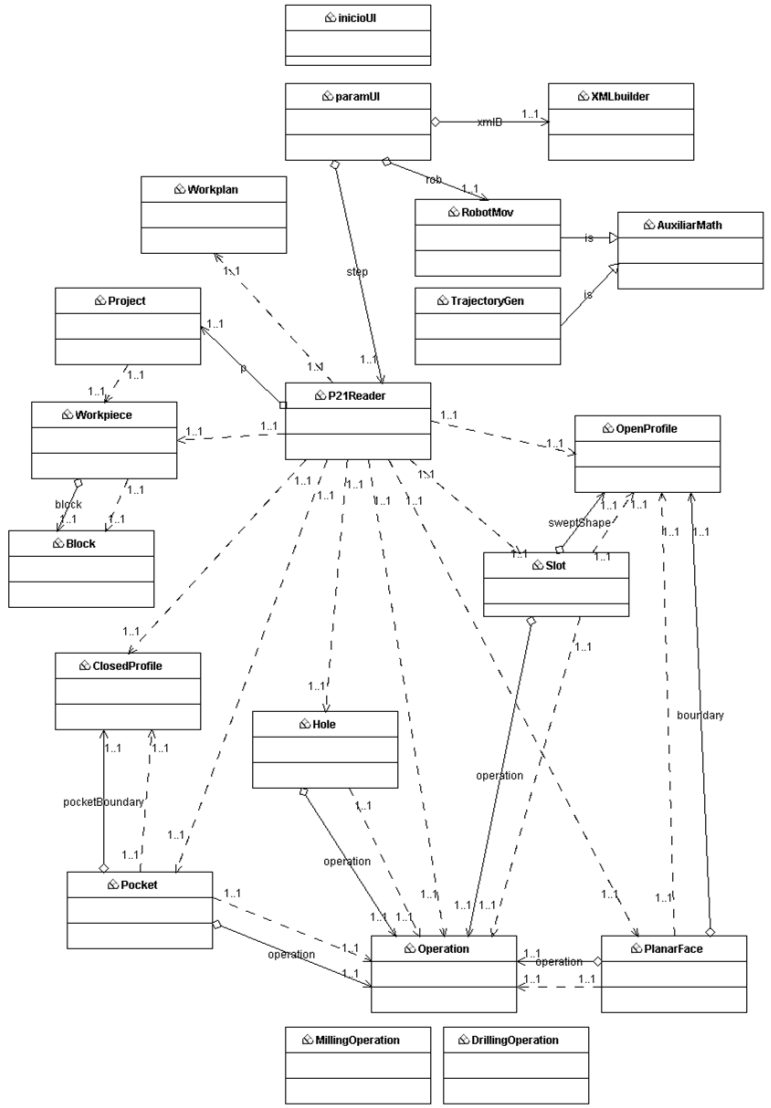


Figura A.0.1 Diagrama de relação de classes da ferramenta desenvolvida
FONTE: Da autora

Figura A.0.2 Variáveis e métodos das classes principais
FONTE: Da autora

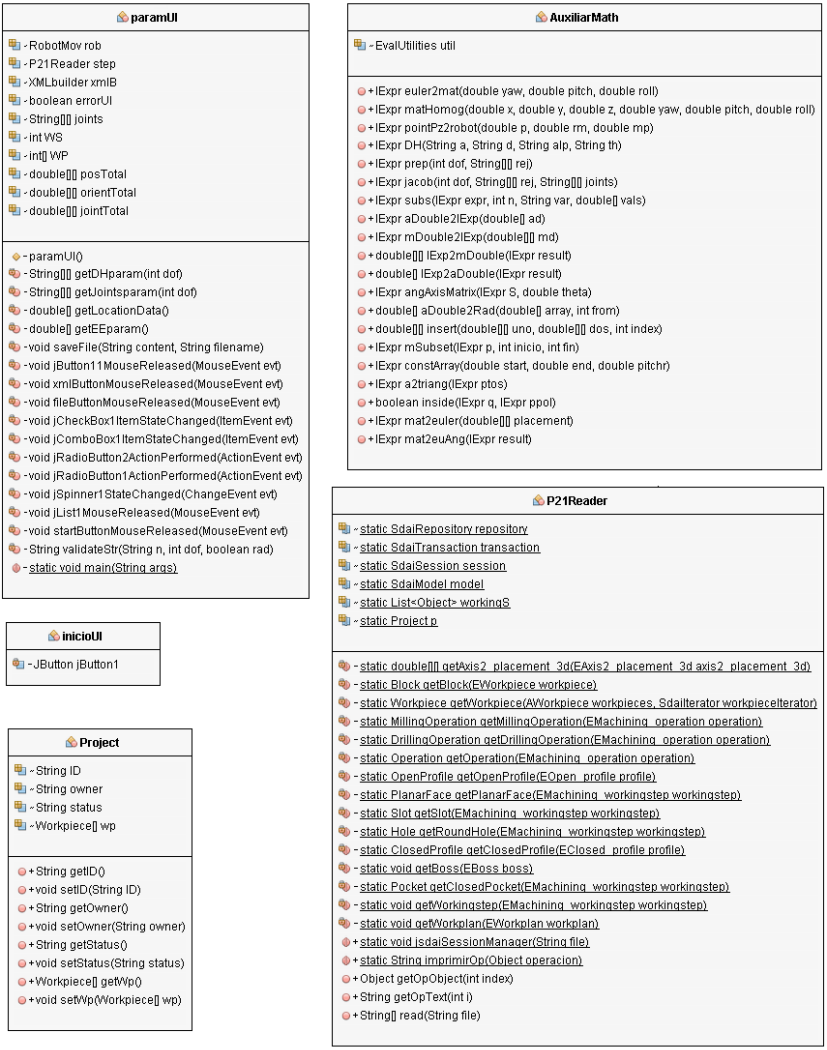


Figura A.0.3 Variáveis e métodos das classes principais (cont.)
FONTE: Da autora

Figura A.0.4 Variáveis e métodos das classes secundárias
FONTE: Da autora

Figura A.0.5 Variáveis e métodos das classes secundárias (cont.)
FONTE: Da autora

Figura A.0.6 Variáveis e métodos das classes secundárias (cont.)
FONTE: Da autora

APÊNDICE B – Peça Exemplo 2 e Arquivo Físico STEP-NC

Peça exemplo 2

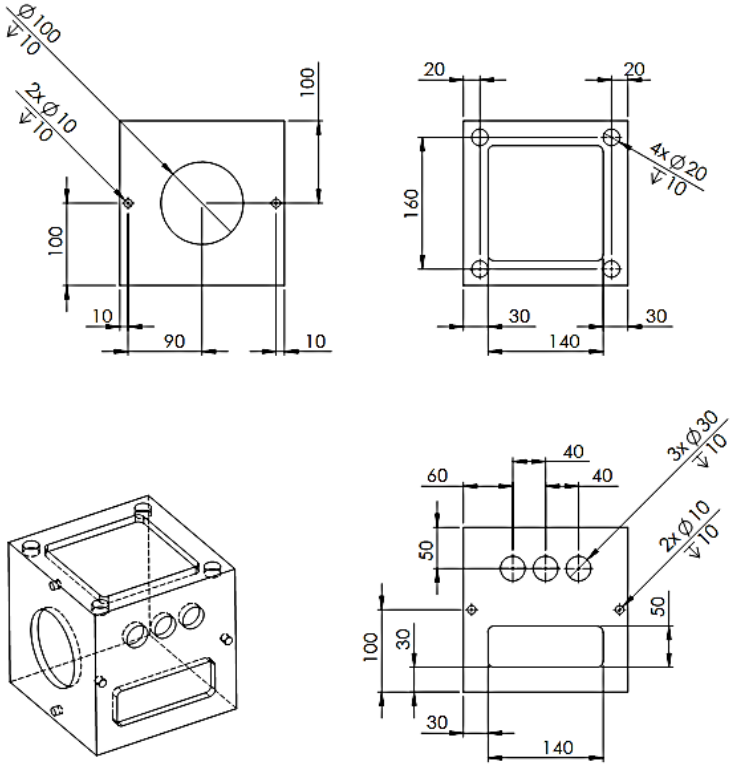


Figura B.0.1 Dimensões peça exemplo 2
FONTE: Da autora

Conteúdo do arquivo físico STEP-NC

```
ISO-10303-21;
HEADER;
 /*GRIMA*/
FILE_DESCRIPTION(
/* description */ (''),
/* implementation_level */ '2;1');
FILE_NAME(
/* name */ ' ',
/* time_stamp */ '2015-12-14T16:09:07',
/* author */ ('nr352br'),
/* organization */ (''),
```

```
/* preprocessor_version */ 'ST-GENERATOR 5.3',
/* originating_system */ ' ',
/* authorization */ ' ');
FILE_SCHEMA(('COMBINED_SCHEMA'));
ENDSEC;
DATA;
#1=PROJECT('3caras',#732,(#2),$,$,$);
#2=WORKPIECE('workpiece',#7,50.0,$,$,#9,());
#3=AXIS2_PLACEMENT_3D('workpiece placement',#4,#5,#6);
#4=CARTESIAN_POINT('',(0.0,0.0,0.0));
#5=DIRECTION('',(0.0,0.0,1.0));
#6=DIRECTION('',(1.0,0.0,0.0));
#7=MATERIAL('SAE 1020','ACO SEM LIGA',(#8));
#8=NUMERIC_PARAMETER('Hardness',250.0,'HB');
#9=BLOCK('piece',#3,200.0,200.0,200.0);
#10=SETUP('setup',#16,#15,(#20));
#11=AXIS2_PLACEMENT_3D('',#12,#13,#14);
#12=CARTESIAN_POINT('',(0.0,0.0,225.0));
#13=DIRECTION('',(0.0,0.0,1.0));
#14=DIRECTION('',(1.0,0.0,0.0));
#15=PLANE('security plane',#11);
#16=AXIS2_PLACEMENT_3D('origin',#17,#18,#19);
#17=CARTESIAN_POINT('',(0.0,0.0,0.0));
#18=DIRECTION('',(0.0,0.0,1.0));
#19=DIRECTION('',(1.0,0.0,0.0));
#20=WORKPIECE_SETUP(#2,#21,$,$,());
#21=AXIS2_PLACEMENT_3D('',#22,#23,#24);
#22=CARTESIAN_POINT('',(0.0,0.0,0.0));
#23=DIRECTION('',(0.0,0.0,1.0));
#24=DIRECTION('',(1.0,0.0,0.0));
#25=CENTER_DRILLING($,$,'Center Drilling',5.0,#26,#27,#32,#33,$,10.0,$,$,$,#34);
#26=CARTESIAN_POINT('start point',(0.0,0.0,0.0));
#27=MILLING_CUTTING_TOOL('center drill',#28,(#30),$,$,$);
#28=CENTER_DRILL(#29,2,.LEFT.,$,25.0);
#29=MILLING_TOOL_DIMENSION(10.0,0.5235987755982988,0.0,10.0,0.0,0.0,0.0,0.0);
#30=CUTTING_COMPONENT(50.0,#31,$,$,$);
#31=MATERIAL('P','CARBIDE',());
#32=MILLING_TECHNOLOGY(0.245,.TCP.,110.0,3501.4087480216976,$,.F.,.F.,.F.,$);
#33=MILLING_MACHINE_FUNCTIONS(.T.,$,$,.T.,$,(),.T.,$,$,());
#34=DRILLING_TYPE_STRATEGY($,$,$,$,$,$);
#35=ROUND_HOLE('Plane Base Hole',#2,(#25,#49,#60),#36,#46,#40,$,#47);
#36=AXIS2_PLACEMENT_3D('Plane Base Hole placement',#37,#38,#39);
#37=CARTESIAN_POINT('',(20.0,20.0,200.0));
#38=DIRECTION('',(0.0,0.0,1.0));
#39=DIRECTION('',(1.0,0.0,0.0));
#40=TOLERANCED_LENGTH_MEASURE(20.0,#41);
#41=PLUS_MINUS_VALUE(0.05,0.05,4);
#42=AXIS2_PLACEMENT_3D('hole depth',#43,#44,#45);
#43=CARTESIAN_POINT('',(0.0,0.0,-10.0));
#44=DIRECTION('',(0.0,0.0,1.0));
#45=DIRECTION('',(1.0,0.0,0.0));
#46=PLANE('hole depth plane',#42);
#47=FLAT_HOLE_BOTTOM();
```

```
#48=MACHINING_WORKINGSTEP('Plane Base Hole_RGH',#15,#35,#25,$);
#49=DRILLING($,$,'Drilling',5.0,#57,#50,#55,#56,$,10.0,$,$,$,#58);
#50=MILLING_CUTTING_TOOL('twist drill',#51,(#53),$,$,$);
#51=TWIST_DRILL(#52,2,.LEFT.,$,60.0);
#52=MILLING_TOOL_DIMENSION(10.0,0.3490658503988659,0.0,40.0,0.0,0.0,0.0);
#53=CUTTING_COMPONENT(70.0,#54,$,$,$);
#54=MATERIAL('P','CARBIDE',());
#55=MILLING_TECHNOLOGY(0.245,.TCP.,110.0,3501.4087480216976,$,.F.,.F.,.F.,$);
#56=MILLING_MACHINE_FUNCTIONS(.T.,$,$,.T.,$,(),.T.,$,$,());
#57=CARTESIAN_POINT('start point',(0.0,0.0,0.0));
#58=DRILLING_TYPE_STRATEGY($,$,$,$,$,$);
#59=MACHINING_WORKINGSTEP('Plane Base Hole_RGH',#15,#35,#49,$);
#60=BOTTOM_AND_SIDE_ROUGH_MILLING($,$,'Bottom And Side Rough
Milling',5.0,#68,#61,
    #66,#67,$,#69,#70,#73,2.0,15.0,0.0,0.0);
#61=MILLING_CUTTING_TOOL('SF20',#62,(#64),$,$,$);
#62=FACEMILL(#63,4,.LEFT.,$,60.0);
#63=MILLING_TOOL_DIMENSION(20.0,0.0,0.0,40.0,0.0,0.0,0.0);
#64=CUTTING_COMPONENT(70.0,#65,$,$,$);
#65=MATERIAL('P','CARBIDE',());
#66=MILLING_TECHNOLOGY(0.073,.TCP.,145.0,2307.7466748324823,$,.F.,.F.,.F.,$);
#67=MILLING_MACHINE_FUNCTIONS(.T.,$,$,.T.,$,(),.T.,$,$,());
#68=CARTESIAN_POINT('start point',(0.0,0.0,0.0));
#69=PLUNGE_TOOLAXIS(#71);
#70=PLUNGE_TOOLAXIS(#72);
#71=DIRECTION('approach strategy direction',(0.0,0.0,1.0));
#72=DIRECTION('retract strategy direction',(0.0,0.0,-1.0));
#73=CONTOUR_PARALLEL(4.0,.T.,.CCW.,.CONVENTIONAL.);
#74=MACHINING_WORKINGSTEP('Plane Base Hole_RGH',#15,#35,#60,$);
#75=CENTER_DRILLING($,$,'Center Drilling',5.0,#76,#77,#82,#83,$,10.0,$,$,$,#84);
#76=CARTESIAN_POINT('start point',(0.0,0.0,0.0));
#77=MILLING_CUTTING_TOOL('center drill',#78,(#80),$,$,$);
#78=CENTER_DRILL(#79,2,.LEFT.,$,25.0);
#79=MILLING_TOOL_DIMENSION(10.0,0.5235987755982988,0.0,10.0,0.0,0.0,0.0);
#80=CUTTING_COMPONENT(50.0,#81,$,$,$);
#81=MATERIAL('P','CARBIDE',());
#82=MILLING_TECHNOLOGY(0.245,.TCP.,110.0,3501.4087480216976,$,.F.,.F.,.F.,$);
#83=MILLING_MACHINE_FUNCTIONS(.T.,$,$,.T.,$,(),.T.,$,$,());
#84=DRILLING_TYPE_STRATEGY($,$,$,$,$,$);
#85=ROUND_HOLE('Plane Base Hole',#2,(#75,#99,#110),#86,#96,#90,$,#97);
#86=AXIS2_PLACEMENT_3D('Plane Base Hole placement',#87,#88,#89);
#87=CARTESIAN_POINT('',(180.0,20.0,200.0));
#88=DIRECTION('',(0.0,0.0,1.0));
#89=DIRECTION('',(1.0,0.0,0.0));
#90=TOLERANCED_LENGTH_MEASURE(20.0,#91);
#91=PLUS_MINUS_VALUE(0.05,0.05,4);
#92=AXIS2_PLACEMENT_3D('hole depth',#93,#94,#95);
#93=CARTESIAN_POINT('',(0.0,0.0,-10.0));
#94=DIRECTION('',(0.0,0.0,1.0));
#95=DIRECTION('',(1.0,0.0,0.0));
#96=PLANE('hole depth plane',#92);
#97=FLAT_HOLE_BOTTOM();
#98=MACHINING_WORKINGSTEP('Plane Base Hole_RGH',#15,#85,#75,$);
```

```
#99=DRILLING($,$,'Drilling',5.0,#107,#100,#105,#106,$,10.0,$,$,$,#108);
#100=MILLING_CUTTING_TOOL('twist drill',#101,(#103),$,$,$);
#101=TWIST_DRILL(#102,2,.LEFT.,$,60.0);
#102=MILLING_TOOL_DIMENSION(10.0,0.3490658503988659,0.0,40.0,0.0,0.0,0.0,0.0);
#103=CUTTING_COMPONENT(70.0,#104,$,$,$);
#104=MATERIAL('P','CARBIDE',());
#105=MILLING_TECHNOLOGY(0.245,.TCP.,110.0,3501.4087480216976,$,.F.,.F.,.F.,$);
#106=MILLING_MACHINE_FUNCTIONS(.T.,$,$,.T.,$,(),.T.,$,$,());
#107=CARTESIAN_POINT('start point',(0.0,0.0,0.0));
#108=DRILLING_TYPE_STRATEGY($,$,$,$,$,$);
#109=MACHINING_WORKINGSTEP('Plane Base Hole_RGH',#15,#85,#99,$);
#110=BOTTOM_AND_SIDE_ROUGH_MILLING($,$,'Bottom And Side Rough
Milling',5.0,#118,
    #111,#116,#117,$,#119,#120,#123,2.0,15.0,0.0,0.0);
#111=MILLING_CUTTING_TOOL('SF20',#112,(#114),$,$,$);
#112=FACEMILL(#113,4,.LEFT.,$,60.0);
#113=MILLING_TOOL_DIMENSION(20.0,0.0,0.0,40.0,0.0,0.0,0.0,0.0);
#114=CUTTING_COMPONENT(70.0,#115,$,$,$);
#115=MATERIAL('P','CARBIDE',());
#116=MILLING_TECHNOLOGY(0.073,.TCP.,145.0,2307.7466748324823,$,.F.,.F.,.F.,$);
#117=MILLING_MACHINE_FUNCTIONS(.T.,$,$,.T.,$,(),.T.,$,$,());
#118=CARTESIAN_POINT('start point',(0.0,0.0,0.0));
#119=PLUNGE_TOOLAXIS(#121);
#120=PLUNGE_TOOLAXIS(#122);
#121=DIRECTION('approach strategy direction',(0.0,0.0,1.0));
#122=DIRECTION('retract strategy direction',(0.0,0.0,-1.0));
#123=CONTOUR_PARALLEL(4.0,.T.,.CCW.,.CONVENTIONAL.);
#124=MACHINING_WORKINGSTEP('Plane Base Hole_RGH',#15,#85,#110,$);
#125=CENTER_DRILLING($,$,'Center Drilling',5.0,#126,#127,#132,#133,$,10.0,$,$,$,
    #134);
#126=CARTESIAN_POINT('start point',(0.0,0.0,0.0));
#127=MILLING_CUTTING_TOOL('center drill',#128,(#130),$,$,$);
#128=CENTER_DRILL(#129,2,.LEFT.,$,25.0);
#129=MILLING_TOOL_DIMENSION(10.0,0.5235987755982988,0.0,10.0,0.0,0.0,0.0,0.0);
#130=CUTTING_COMPONENT(50.0,#131,$,$,$);
#131=MATERIAL('P','CARBIDE',());
#132=MILLING_TECHNOLOGY(0.245,.TCP.,110.0,3501.4087480216976,$,.F.,.F.,.F.,$);
#133=MILLING_MACHINE_FUNCTIONS(.T.,$,$,.T.,$,(),.T.,$,$,());
#134=DRILLING_TYPE_STRATEGY($,$,$,$,$,$);
#135=ROUND_HOLE('Plane Base Hole',#2,(#125,#149,#160),#136,#146,#140,$,#147);
#136=AXIS2_PLACEMENT_3D('Plane Base Hole placement',#137,#138,#139);
#137=CARTESIAN_POINT('',(20.0,180.0,200.0));
#138=DIRECTION('',(0.0,0.0,1.0));
#139=DIRECTION('',(1.0,0.0,0.0));
#140=TOLERANCED_LENGTH_MEASURE(20.0,#141);
#141=PLUS_MINUS_VALUE(0.05,0.05,4);
#142=AXIS2_PLACEMENT_3D('hole depth',#143,#144,#145);
#143=CARTESIAN_POINT('',(0.0,0.0,-10.0));
#144=DIRECTION('',(0.0,0.0,1.0));
#145=DIRECTION('',(1.0,0.0,0.0));
#146=PLANE('hole depth plane',#142);
#147=FLAT_HOLE_BOTTOM();
#148=MACHINING_WORKINGSTEP('Plane Base Hole_RGH',#15,#135,#125,$);
```

```
#149=DRILLING($,$,'Drilling',5.0,#157,#150,#155,#156,$,10.0,$,$,$,#158);
#150=MILLING_CUTTING_TOOL('twist drill',#151,(#153),$,$,$);
#151=TWIST_DRILL(#152,2,.LEFT.,$,60.0);
#152=MILLING_TOOL_DIMENSION(10.0,0.3490658503988659,0.0,40.0,0.0,0.0,0.0);
#153=CUTTING_COMPONENT(70.0,#154,$,$,$);
#154=MATERIAL('P','CARBIDE',());
#155=MILLING_TECHNOLOGY(0.245,.TCP.,110.0,3501.4087480216976,$,.F.,.F.,.F.,$);
#156=MILLING_MACHINE_FUNCTIONS(.T.,$,$,.T.,$,(),.T.,$,$,());
#157=CARTESIAN_POINT('start point',(0.0,0.0,0.0));
#158=DRILLING_TYPE_STRATEGY($,$,$,$,$,$);
#159=MACHINING_WORKINGSTEP('Plane Base Hole_RGH',#15,#135,#149,$);
#160=BOTTOM_AND_SIDE_ROUGH_MILLING($,$,'Bottom And Side Rough
Milling',5.0,#168,
    #161,#166,#167,$,#169,#170,#173,2.0,15.0,0.0,0.0);
#161=MILLING_CUTTING_TOOL('SF20',#162,(#164),$,$,$);
#162=FACEMILL(#163,4,.LEFT.,$,60.0);
#163=MILLING_TOOL_DIMENSION(20.0,0.0,0.0,40.0,0.0,0.0,0.0,0.0);
#164=CUTTING_COMPONENT(70.0,#165,$,$,$);
#165=MATERIAL('P','CARBIDE',());
#166=MILLING_TECHNOLOGY(0.073,.TCP.,145.0,2307.7466748324823,$,.F.,.F.,.F.,$);
#167=MILLING_MACHINE_FUNCTIONS(.T.,$,$,.T.,$,(),.T.,$,$,());
#168=CARTESIAN_POINT('start point',(0.0,0.0,0.0));
#169=PLUNGE_TOOLAXIS(#171);
#170=PLUNGE_TOOLAXIS(#172);
#171=DIRECTION('approach strategy direction',(0.0,0.0,1.0));
#172=DIRECTION('retract strategy direction',(0.0,0.0,-1.0));
#173=CONTOUR_PARALLEL(4.0,.T.,.CCW.,.CONVENTIONAL.);
#174=MACHINING_WORKINGSTEP('Plane Base Hole_RGH',#15,#135,#160,$);
#175=CENTER_DRILLING($,$,'Center Drilling',5.0,#176,#177,#182,#183,$,10.0,$,$,$,
    #184);
#176=CARTESIAN_POINT('start point',(0.0,0.0,0.0));
#177=MILLING_CUTTING_TOOL('center drill',#178,(#180),$,$,$);
#178=CENTER_DRILL(#179,2,.LEFT.,$,25.0);
#179=MILLING_TOOL_DIMENSION(10.0,0.5235987755982988,0.0,10.0,0.0,0.0,0.0,0.0);
#180=CUTTING_COMPONENT(50.0,#181,$,$,$);
#181=MATERIAL('P','CARBIDE',());
#182=MILLING_TECHNOLOGY(0.245,.TCP.,110.0,3501.4087480216976,$,.F.,.F.,.F.,$);
#183=MILLING_MACHINE_FUNCTIONS(.T.,$,$,.T.,$,(),.T.,$,$,());
#184=DRILLING_TYPE_STRATEGY($,$,$,$,$,$);
#185=ROUND_HOLE('Plane Base Hole',#2,(#175,#199,#210),#186,#196,#190,$,#197);
#186=AXIS2_PLACEMENT_3D('Plane Base Hole placement',#187,#188,#189);
#187=CARTESIAN_POINT('',(180.0,180.0,200.0));
#188=DIRECTION('',(0.0,0.0,1.0));
#189=DIRECTION('',(1.0,0.0,0.0));
#190=TOLERANCED_LENGTH_MEASURE(20.0,#191);
#191=PLUS_MINUS_VALUE(0.05,0.05,4);
#192=AXIS2_PLACEMENT_3D('hole depth',#193,#194,#195);
#193=CARTESIAN_POINT('',(0.0,0.0,-10.0));
#194=DIRECTION('',(0.0,0.0,1.0));
#195=DIRECTION('',(1.0,0.0,0.0));
#196=PLANE('hole depth plane',#192);
#197=FLAT_HOLE_BOTTOM();
#198=MACHINING_WORKINGSTEP('Plane Base Hole_RGH',#15,#185,#175,$);
```

```
#199=DRILLING($,$,'Drilling',5.0,#207,#200,#205,#206,$,10.0,$,$,$,#208);
#200=MILLING_CUTTING_TOOL('twist drill',#201,(#203),$,$,$);
#201=TWIST_DRILL(#202,2,.LEFT.,$,60.0);
#202=MILLING_TOOL_DIMENSION(10.0,0.3490658503988659,0.0,40.0,0.0,0.0,0.0,0.0);
#203=CUTTING_COMPONENT(70.0,#204,$,$,$);
#204=MATERIAL('P','CARBIDE',());
#205=MILLING_TECHNOLOGY(0.245,.TCP.,110.0,3501.4087480216976,$,.F.,.F.,.F.,$);
#206=MILLING_MACHINE_FUNCTIONS(.T.,$,$,.T.,$,(),.T.,$,$,());
#207=CARTESIAN_POINT('start point',(0.0,0.0,0.0));
#208=DRILLING_TYPE_STRATEGY($,$,$,$,$,$);
#209=MACHINING_WORKINGSTEP('Plane Base Hole_RGH',#15,#185,#199,$);
#210=BOTTOM_AND_SIDE_ROUGH_MILLING($,$,'Bottom And Side Rough
Milling',5.0,#218,
    #211,#216,#217,$,#219,#220,#223,2.0,15.0,0.0,0.0);
#211=MILLING_CUTTING_TOOL('SF20',#212,(#214),$,$,$);
#212=FACEMILL(#213,4,.LEFT.,$,60.0);
#213=MILLING_TOOL_DIMENSION(20.0,0.0,0.0,40.0,0.0,0.0,0.0,0.0);
#214=CUTTING_COMPONENT(70.0,#215,$,$,$);
#215=MATERIAL('P','CARBIDE',());
#216=MILLING_TECHNOLOGY(0.073,.TCP.,145.0,2307.7466748324823,$,.F.,.F.,.F.,$);
#217=MILLING_MACHINE_FUNCTIONS(.T.,$,$,.T.,$,(),.T.,$,$,());
#218=CARTESIAN_POINT('start point',(0.0,0.0,0.0));
#219=PLUNGE_TOOLAXIS(#221);
#220=PLUNGE_TOOLAXIS(#222);
#221=DIRECTION('approach strategy direction',(0.0,0.0,1.0));
#222=DIRECTION('retract strategy direction',(0.0,0.0,-1.0));
#223=CONTOUR_PARALLEL(4.0,.T.,.CCW.,.CONVENTIONAL.);
#224=MACHINING_WORKINGSTEP('Plane Base Hole_RGH',#15,#185,#210,$);
#225=BOTTOM_AND_SIDE_ROUGH_MILLING($,$,'Bottom And Side Rough
Milling',5.0,#233,
    #226,#231,#232,$,#234,#235,#238,2.0,15.0,0.0,0.0);
#226=MILLING_CUTTING_TOOL('SF20',#227,(#229),$,$,$);
#227=FACEMILL(#228,4,.LEFT.,$,60.0);
#228=MILLING_TOOL_DIMENSION(20.0,0.0,0.0,40.0,0.0,0.0,0.0,0.0);
#229=CUTTING_COMPONENT(70.0,#230,$,$,$);
#230=MATERIAL('P','CARBIDE',());
#231=MILLING_TECHNOLOGY(0.073,.TCP.,145.0,2307.7466748324823,$,.F.,.F.,.F.,$);
#232=MILLING_MACHINE_FUNCTIONS(.T.,$,$,.T.,$,(),.T.,$,$,());
#233=CARTESIAN_POINT('start point',(0.0,0.0,0.0));
#234=PLUNGE_TOOLAXIS(#236);
#235=PLUNGE_TOOLAXIS(#237);
#236=DIRECTION('approach strategy direction',(0.0,0.0,1.0));
#237=DIRECTION('retract strategy direction',(0.0,0.0,-1.0));
#238=CONTOUR_PARALLEL(4.0,.T.,.CCW.,.CONVENTIONAL.);
#239=CLOSED_POCKET('Closed pocket',#2,(#225,#262),#240,#248,(),$,#249,$,#250,#252);
#240=AXIS2_PLACEMENT_3D('Closed pocket placement',#241,#242,#243);
#241=CARTESIAN_POINT('',(100.0,100.0,200.0));
#242=DIRECTION('',(0.0,0.0,1.0));
#243=DIRECTION('',(1.0,0.0,0.0));
#244=AXIS2_PLACEMENT_3D('Closed pocket depth',#245,#246,#247);
#245=CARTESIAN_POINT('',(0.0,0.0,-10.0));
#246=DIRECTION('',(0.0,0.0,1.0));
#247=DIRECTION('',(1.0,0.0,0.0));
```

```
#248=PLANE('Closed pocket depth location',#244);
#249=PLANAR_POCKET_BOTTOM_CONDITION();
#250=TOLERANCED_LENGTH_MEASURE(10.0,#251);
#251=PLUS_MINUS_VALUE(0.05,0.05,4);
#252=RECTANGULAR_CLOSED_PROFILE(#253,#259,#257);
#253=AXIS2_PLACEMENT_3D('Closed pocket placement',#254,#255,#256);
#254=CARTESIAN_POINT('',(100.0,100.0,200.0));
#255=DIRECTION('',(0.0,0.0,1.0));
#256=DIRECTION('',(1.0,0.0,0.0));
#257=TOLERANCED_LENGTH_MEASURE(140.0,#258);
#258=PLUS_MINUS_VALUE(0.05,0.05,4);
#259=TOLERANCED_LENGTH_MEASURE(140.0,#260);
#260=PLUS_MINUS_VALUE(0.05,0.05,4);
#261=MACHINING_WORKINGSTEP('Pocket_RGH',#15,#239,#225,$);
#262=BOTTOM_AND_SIDE_ROUGH_MILLING($,$,'Bottom And Side Rough
Milling',5.0,#270,
    #263,#268,#269,$,#271,#272,#275,1.6,12.0,0.0,0.0);
#263=MILLING_CUTTING_TOOL('SF20',#264,(#266),$,$,$);
#264=FACEMILL(#265,4,.LEFT.,$,60.0);
#265=MILLING_TOOL_DIMENSION(16.0,0.0,0.0,40.0,0.0,0.0,0.0);
#266=CUTTING_COMPONENT(70.0,#267,$,$,$);
#267=MATERIAL('P','CARBIDE',());
#268=MILLING_TECHNOLOGY(0.0505,.TCP.,145.0,2884.683343540603,$,.F.,.F.,.F.,$);
#269=MILLING_MACHINE_FUNCTIONS(.T.,$,$,.T.,$,(),.T.,$,$,());
#270=CARTESIAN_POINT('start point',(62.0,60.0,0.0));
#271=PLUNGE_TOOLAXIS(#273);
#272=PLUNGE_TOOLAXIS(#274);
#273=DIRECTION('approach strategy direction',(0.0,0.0,1.0));
#274=DIRECTION('retract strategy direction',(0.0,0.0,-1.0));
#275=CONTOUR_PARALLEL(3.2,.T.,.CCW.,.CONVENTIONAL.);
#276=MACHINING_WORKINGSTEP('Pocket_RGH',#15,#239,#262,$);
#277=WORKPLAN('workplan setup XY',(#59,#74,#109,#124,#159,#174,
    #209,#224,#276),$,#10,$);
#278=BOTTOM_AND_SIDE_ROUGH_MILLING($,$,'Bottom And Side Rough
Milling',5.0,#286,
    #279,#284,#285,$,#287,#288,#291,2.0,15.0,0.0,0.0);
#279=MILLING_CUTTING_TOOL('SF20',#280,(#282),$,$,$);
#280=FACEMILL(#281,4,.LEFT.,$,60.0);
#281=MILLING_TOOL_DIMENSION(20.0,0.0,0.0,40.0,0.0,0.0,0.0);
#282=CUTTING_COMPONENT(70.0,#283,$,$,$);
#283=MATERIAL('P','CARBIDE',());
#284=MILLING_TECHNOLOGY(0.073,.TCP.,145.0,2307.7466748324823,$,.F.,.F.,.F.,$);
#285=MILLING_MACHINE_FUNCTIONS(.T.,$,$,.T.,$,(),.T.,$,$,());
#286=CARTESIAN_POINT('start point',(0.0,0.0,0.0));
#287=PLUNGE_TOOLAXIS(#289);
#288=PLUNGE_TOOLAXIS(#290);
#289=DIRECTION('approach strategy direction',(0.0,-1.0,0.0));
#290=DIRECTION('retract strategy direction',(0.0,-1.0,-0.0));
#291=CONTOUR_PARALLEL(4.0,.T.,.CCW.,.CONVENTIONAL.);
#292=CLOSED_POCKET('Closed pocket',#2,(#278,#315),#293,#301,(),$,#302,$,#303,#305);
#293=AXIS2_PLACEMENT_3D('Closed pocket placement',#294,#295,#296);
#294=CARTESIAN_POINT('',(100.0,0.0,55.0));
#295=DIRECTION('',(0.0,-1.0,0.0));
```

```
#296=DIRECTION('',(1.0,0.0,0.0));
#297=AXIS2_PLACEMENT_3D('Closed pocket depth',#298,#299,#300);
#298=CARTESIAN_POINT('',(0.0,0.0,-10.0));
#299=DIRECTION('',(0.0,-1.0,0.0));
#300=DIRECTION('',(1.0,0.0,0.0));
#301=PLANE('Closed pocket depth location',#297);
#302=PLANAR_POCKET_BOTTOM_CONDITION();
#303=TOLERANCED_LENGTH_MEASURE(10.0,#304);
#304=PLUS_MINUS_VALUE(0.05,0.05,4);
#305=RECTANGULAR_CLOSED_PROFILE(#306,#312,#310);
#306=AXIS2_PLACEMENT_3D('Closed pocket placement',#307,#308,#309);
#307=CARTESIAN_POINT('',(100.0,0.0,55.0));
#308=DIRECTION('',(0.0,-1.0,0.0));
#309=DIRECTION('',(1.0,0.0,0.0));
#310=TOLERANCED_LENGTH_MEASURE(140.0,#311);
#311=PLUS_MINUS_VALUE(0.05,0.05,4);
#312=TOLERANCED_LENGTH_MEASURE(50.0,#313);
#313=PLUS_MINUS_VALUE(0.05,0.05,4);
#314=MACHINING_WORKINGSTEP('Pocket_RGH',#15,#292,#278,$);
#315=BOTTOM_AND_SIDE_ROUGH_MILLING($,$,'Bottom And Side Rough
Milling',5.0,#323,
    #316,#321,#322,$,#324,#325,#328,1.6,12.0,0.0,0.0);
#316=MILLING_CUTTING_TOOL('SF20',#317,(#319),$,$,$);
#317=FACEMILL(#318,4,.LEFT.,$,60.0);
#318=MILLING_TOOL_DIMENSION(16.0,0.0,0.0,40.0,0.0,0.0,0.0);
#319=CUTTING_COMPONENT(70.0,#320,$,$,$);
#320=MATERIAL('P','CARBIDE',());
#321=MILLING_TECHNOLOGY(0.0505,.TCP.,145.0,2884.683343540603,$,.F.,.F.,.F.,$);
#322=MILLING_MACHINE_FUNCTIONS(.T.,$,$,.T.,$,(),.T.,$,$,());
#323=CARTESIAN_POINT('start point',(62.0,15.0,0.0));
#324=PLUNGE_TOOLAXIS(#326);
#325=PLUNGE_TOOLAXIS(#327);
#326=DIRECTION('approach strategy direction',(0.0,-1.0,0.0));
#327=DIRECTION('retract strategy direction',(0.0,-1.0,-0.0));
#328=CONTOUR_PARALLEL(3.2,.T.,.CCW.,.CONVENTIONAL.);
#329=MACHINING_WORKINGSTEP('Pocket_RGH',#15,#292,#315,$);
#330=CENTER_DRILLING($,$,'Center Drilling',5.0,#331,#332,#337,#338,$,10.0,$,$,$,
    #339);
#331=CARTESIAN_POINT('start point',(0.0,0.0,0.0));
#332=MILLING_CUTTING_TOOL('center drill',#333,(#335),$,$,$);
#333=CENTER_DRILL(#334,2,.LEFT.,$,25.0);
#334=MILLING_TOOL_DIMENSION(10.0,0.5235987755982988,0.0,10.0,0.0,0.0,0.0,0.0);
#335=CUTTING_COMPONENT(50.0,#336,$,$,$);
#336=MATERIAL('P','CARBIDE',());
#337=MILLING_TECHNOLOGY(0.245,.TCP.,110.0,3501.4087480216976,$,.F.,.F.,.F.,$);
#338=MILLING_MACHINE_FUNCTIONS(.T.,$,$,.T.,$,(),.T.,$,$,());
#339=DRILLING_TYPE_STRATEGY($,$,$,$,$,$);
#340=ROUND_HOLE('Plane Base Hole',#2,(#330,#354,#365),#341,#351,#345,$,#352);
#341=AXIS2_PLACEMENT_3D('Plane Base Hole placement',#342,#343,#344);
#342=CARTESIAN_POINT('',(60.0,0.0,150.0));
#343=DIRECTION('',(0.0,-1.0,0.0));
#344=DIRECTION('',(1.0,0.0,0.0));
#345=TOLERANCED_LENGTH_MEASURE(30.0,#346);
```

```
#346=PLUS_MINUS_VALUE(0.05,0.05,4);
#347=AXIS2_PLACEMENT_3D('hole depth',#348,#349,#350);
#348=CARTESIAN_POINT('',(0.0,0.0,-10.0));
#349=DIRECTION('',(0.0,-1.0,0.0));
#350=DIRECTION('',(1.0,0.0,0.0));
#351=PLANE('hole depth plane',#347);
#352=FLAT_HOLE_BOTTOM();
#353=MACHINING_WORKINGSTEP('Plane Base Hole_RGH',#15,#340,#330,$);
#354=DRILLING($,$,'Drilling',5.0,#362,#355,#360,#361,$,10.0,$,$,$,#363);
#355=MILLING_CUTTING_TOOL('twist drill',#356,(#358),$,$,$);
#356=TWIST_DRILL(#357,2,.LEFT.,$,60.0);
#357=MILLING_TOOL_DIMENSION(10.0,0.3490658503988659,0.0,40.0,0.0,0.0,0.0);
#358=CUTTING_COMPONENT(70.0,#359,$,$,$);
#359=MATERIAL('P','CARBIDE',());
#360=MILLING_TECHNOLOGY(0.245,.TCP.,110.0,3501.4087480216976,$,.F.,.F.,.F.,$);
#361=MILLING_MACHINE_FUNCTIONS(.T.,$,$,.T.,$,(),.T.,$,$,());
#362=CARTESIAN_POINT('start point',(0.0,0.0,0.0));
#363=DRILLING_TYPE_STRATEGY($,$,$,$,$,$);
#364=MACHINING_WORKINGSTEP('Plane Base Hole_RGH',#15,#340,#354,$);
#365=BOTTOM_AND_SIDE_ROUGH_MILLING($,$,'Bottom And Side Rough
Milling',5.0,#373,
    #366,#371,#372,$,#374,#375,#378,2.0,15.0,0.0,0.0);
#366=MILLING_CUTTING_TOOL('SF20',#367,(#369),$,$,$);
#367=FACEMILL(#368,4,.LEFT.,$,60.0);
#368=MILLING_TOOL_DIMENSION(20.0,0.0,0.0,40.0,0.0,0.0,0.0,0.0);
#369=CUTTING_COMPONENT(70.0,#370,$,$,$);
#370=MATERIAL('P','CARBIDE',());
#371=MILLING_TECHNOLOGY(0.073,.TCP.,145.0,2307.7466748324823,$,.F.,.F.,.F.,$);
#372=MILLING_MACHINE_FUNCTIONS(.T.,$,$,.T.,$,(),.T.,$,$,());
#373=CARTESIAN_POINT('start point',(0.0,5.0,0.0));
#374=PLUNGE_TOOLAXIS(#376);
#375=PLUNGE_TOOLAXIS(#377);
#376=DIRECTION('approach strategy direction',(0.0,-1.0,0.0));
#377=DIRECTION('retract strategy direction',(0.0,-1.0,-0.0));
#378=CONTOUR_PARALLEL(4.0,.T.,.CCW.,.CONVENTIONAL.);
#379=MACHINING_WORKINGSTEP('Plane Base Hole_RGH',#15,#340,#365,$);
#380=CENTER_DRILLING($,$,'Center Drilling',5.0,#381,#382,#387,#388,$,10.0,$,$,$,
    #389);
#381=CARTESIAN_POINT('start point',(0.0,0.0,0.0));
#382=MILLING_CUTTING_TOOL('center drill',#383,(#385),$,$,$);
#383=CENTER_DRILL(#384,2,.LEFT.,$,25.0);
#384=MILLING_TOOL_DIMENSION(10.0,0.5235987755982988,0.0,10.0,0.0,0.0,0.0,0.0);
#385=CUTTING_COMPONENT(50.0,#386,$,$,$);
#386=MATERIAL('P','CARBIDE',());
#387=MILLING_TECHNOLOGY(0.245,.TCP.,110.0,3501.4087480216976,$,.F.,.F.,.F.,$);
#388=MILLING_MACHINE_FUNCTIONS(.T.,$,$,.T.,$,(),.T.,$,$,());
#389=DRILLING_TYPE_STRATEGY($,$,$,$,$,$);
#390=ROUND_HOLE('Plane Base Hole',#2,(#380,#404,#415),#391,#401,#395,$,#402);
#391=AXIS2_PLACEMENT_3D('Plane Base Hole placement',#392,#393,#394);
#392=CARTESIAN_POINT('',(100.0,0.0,150.0));
#393=DIRECTION('',(0.0,-1.0,0.0));
#394=DIRECTION('',(1.0,0.0,0.0));
#395=TOLERANCED_LENGTH_MEASURE(30.0,#396);
```

```
#396=PLUS_MINUS_VALUE(0.05,0.05,4);
#397=AXIS2_PLACEMENT_3D('hole depth',#398,#399,#400);
#398=CARTESIAN_POINT('',(0.0,0.0,-10.0));
#399=DIRECTION('',(0.0,-1.0,0.0));
#400=DIRECTION('',(1.0,0.0,0.0));
#401=PLANE('hole depth plane',#397);
#402=FLAT_HOLE_BOTTOM();
#403=MACHINING_WORKINGSTEP('Plane Base Hole_RGH',#15,#390,#380,$);
#404=DRILLING($,$,'Drilling',5.0,#412,#405,#410,#411,$,10.0,$,$,$,#413);
#405=MILLING_CUTTING_TOOL('twist drill',#406,(#408),$,$,$);
#406=TWIST_DRILL(#407,2,.LEFT.,$,60.0);
#407=MILLING_TOOL_DIMENSION(10.0,0.3490658503988659,0.0,40.0,0.0,0.0,0.0,0.0);
#408=CUTTING_COMPONENT(70.0,#409,$,$,$);
#409=MATERIAL('P','CARBIDE',());
#410=MILLING_TECHNOLOGY(0.245,.TCP.,110.0,3501.4087480216976,$,.F.,.F.,.F.,$);
#411=MILLING_MACHINE_FUNCTIONS(.T.,$,$,.T.,$,(),.T.,$,$,());
#412=CARTESIAN_POINT('start point',(0.0,0.0,0.0));
#413=DRILLING_TYPE_STRATEGY($,$,$,$,$,$);
#414=MACHINING_WORKINGSTEP('Plane Base Hole_RGH',#15,#390,#404,$);
#415=BOTTOM_AND_SIDE_ROUGH_MILLING($,$,'Bottom And Side Rough
Milling',5.0,#423,
    #416,#421,#422,$,#424,#425,#428,2.0,15.0,0.0,0.0);
#416=MILLING_CUTTING_TOOL('SF20',#417,(#419),$,$,$);
#417=FACEMILL(#418,4,.LEFT.,$,60.0);
#418=MILLING_TOOL_DIMENSION(20.0,0.0,0.0,40.0,0.0,0.0,0.0,0.0);
#419=CUTTING_COMPONENT(70.0,#420,$,$,$);
#420=MATERIAL('P','CARBIDE',());
#421=MILLING_TECHNOLOGY(0.073,.TCP.,145.0,2307.7466748324823,$,.F.,.F.,.F.,$);
#422=MILLING_MACHINE_FUNCTIONS(.T.,$,$,.T.,$,(),.T.,$,$,());
#423=CARTESIAN_POINT('start point',(0.0,5.0,0.0));
#424=PLUNGE_TOOLAXIS(#426);
#425=PLUNGE_TOOLAXIS(#427);
#426=DIRECTION('approach strategy direction',(0.0,-1.0,0.0));
#427=DIRECTION('retract strategy direction',(0.0,-1.0,-0.0));
#428=CONTOUR_PARALLEL(4.0,.T.,.CCW.,.CONVENTIONAL.);
#429=MACHINING_WORKINGSTEP('Plane Base Hole_RGH',#15,#390,#415,$);
#430=CENTER_DRILLING($,$,'Center Drilling',5.0,#431,#432,#437,#438,$,10.0,$,$,$,
    #439);
#431=CARTESIAN_POINT('start point',(0.0,0.0,0.0));
#432=MILLING_CUTTING_TOOL('center drill',#433,(#435),$,$,$);
#433=CENTER_DRILL(#434,2,.LEFT.,$,25.0);
#434=MILLING_TOOL_DIMENSION(10.0,0.5235987755982988,0.0,10.0,0.0,0.0,0.0,0.0);
#435=CUTTING_COMPONENT(50.0,#436,$,$,$);
#436=MATERIAL('P','CARBIDE',());
#437=MILLING_TECHNOLOGY(0.245,.TCP.,110.0,3501.4087480216976,$,.F.,.F.,.F.,$);
#438=MILLING_MACHINE_FUNCTIONS(.T.,$,$,.T.,$,(),.T.,$,$,());
#439=DRILLING_TYPE_STRATEGY($,$,$,$,$,$);
#440=ROUND_HOLE('Plane Base Hole',#2,(#430,#454,#465),#441,#451,#445,$,#452);
#441=AXIS2_PLACEMENT_3D('Plane Base Hole placement',#442,#443,#444);
#442=CARTESIAN_POINT('',(140.0,0.0,150.0));
#443=DIRECTION('',(0.0,-1.0,0.0));
#444=DIRECTION('',(1.0,0.0,0.0));
#445=TOLERANCED_LENGTH_MEASURE(30.0,#446);
```

```
#446=PLUS_MINUS_VALUE(0.05,0.05,4);
#447=AXIS2_PLACEMENT_3D('hole depth',#448,#449,#450);
#448=CARTESIAN_POINT('',(0.0,0.0,-10.0));
#449=DIRECTION('',(0.0,-1.0,0.0));
#450=DIRECTION('',(1.0,0.0,0.0));
#451=PLANE('hole depth plane',#447);
#452=FLAT_HOLE_BOTTOM();
#453=MACHINING_WORKINGSTEP('Plane Base Hole_RGH',#15,#440,#430,$);
#454=DRILLING($,$,'Drilling',5.0,#462,#455,#460,#461,$,10.0,$,$,$,#463);
#455=MILLING_CUTTING_TOOL('twist drill',#456,(#458),$,$,$);
#456=TWIST_DRILL(#457,2,.LEFT.,$,60.0);
#457=MILLING_TOOL_DIMENSION(10.0,0.3490658503988659,0.0,40.0,0.0,0.0,0.0);
#458=CUTTING_COMPONENT(70.0,#459,$,$,$);
#459=MATERIAL('P','CARBIDE',());
#460=MILLING_TECHNOLOGY(0.245,.TCP.,110.0,3501.4087480216976,$,.F.,.F.,.F.,$);
#461=MILLING_MACHINE_FUNCTIONS(.T.,$,$,.T.,$,(),.T.,$,$,());
#462=CARTESIAN_POINT('start point',(0.0,0.0,0.0));
#463=DRILLING_TYPE_STRATEGY($,$,$,$,$,$);
#464=MACHINING_WORKINGSTEP('Plane Base Hole_RGH',#15,#440,#454,$);
#465=BOTTOM_AND_SIDE_ROUGH_MILLING($,$,'Bottom And Side Rough
Milling',5.0,#473,
    #466,#471,#472,$,#474,#475,#478,2.0,15.0,0.0,0.0);
#466=MILLING_CUTTING_TOOL('SF20',#467,(#469),$,$,$);
#467=FACEMILL(#468,4,.LEFT.,$,60.0);
#468=MILLING_TOOL_DIMENSION(20.0,0.0,0.0,40.0,0.0,0.0,0.0,0.0);
#469=CUTTING_COMPONENT(70.0,#470,$,$,$);
#470=MATERIAL('P','CARBIDE',());
#471=MILLING_TECHNOLOGY(0.073,.TCP.,145.0,2307.7466748324823,$,.F.,.F.,.F.,$);
#472=MILLING_MACHINE_FUNCTIONS(.T.,$,$,.T.,$,(),.T.,$,$,());
#473=CARTESIAN_POINT('start point',(0.0,5.0,0.0));
#474=PLUNGE_TOOLAXIS(#476);
#475=PLUNGE_TOOLAXIS(#477);
#476=DIRECTION('approach strategy direction',(0.0,-1.0,0.0));
#477=DIRECTION('retract strategy direction',(0.0,-1.0,-0.0));
#478=CONTOUR_PARALLEL(4.0,.T.,.CCW.,.CONVENTIONAL.);
#479=MACHINING_WORKINGSTEP('Plane Base Hole_RGH',#15,#440,#465,$);
#480=CENTER_DRILLING($,$,'Center Drilling',5.0,#481,#482,#487,#488,$,10.0,$,$,$,
    #489);
#481=CARTESIAN_POINT('start point',(0.0,0.0,0.0));
#482=MILLING_CUTTING_TOOL('center drill',#483,(#485),$,$,$);
#483=CENTER_DRILL(#484,2,.LEFT.,$,25.0);
#484=MILLING_TOOL_DIMENSION(10.0,0.5235987755982988,0.0,10.0,0.0,0.0,0.0,0.0);
#485=CUTTING_COMPONENT(50.0,#486,$,$,$);
#486=MATERIAL('P','CARBIDE',());
#487=MILLING_TECHNOLOGY(0.245,.TCP.,110.0,3501.4087480216976,$,.F.,.F.,.F.,$);
#488=MILLING_MACHINE_FUNCTIONS(.T.,$,$,.T.,$,(),.T.,$,$,());
#489=DRILLING_TYPE_STRATEGY($,$,$,$,$,$);
#490=ROUND_HOLE('Plane Base Hole',#2,(#480,#504,#515),#491,#501,#495,$,#502);
#491=AXIS2_PLACEMENT_3D('Plane Base Hole placement',#492,#493,#494);
#492=CARTESIAN_POINT('',(10.0,0.0,100.0));
#493=DIRECTION('',(0.0,-1.0,0.0));
#494=DIRECTION('',(1.0,0.0,0.0));
#495=TOLERANCED_LENGTH_MEASURE(10.0,#496);
```

```
#496=PLUS_MINUS_VALUE(0.05,0.05,4);
#497=AXIS2_PLACEMENT_3D('hole depth',#498,#499,#500);
#498=CARTESIAN_POINT('',(0.0,0.0,-10.0));
#499=DIRECTION('',(0.0,-1.0,0.0));
#500=DIRECTION('',(1.0,0.0,0.0));
#501=PLANE('hole depth plane',#497);
#502=FLAT_HOLE_BOTTOM();
#503=MACHINING_WORKINGSTEP('Plane Base Hole_RGH',#15,#490,#480,$);
#504=DRILLING($,$,'Drilling',5.0,#512,#505,#510,#511,$,10.0,$,$,$,#513);
#505=MILLING_CUTTING_TOOL('twist drill',#506,(#508),$,$,$);
#506=TWIST_DRILL(#507,2,.LEFT.,$,60.0);
#507=MILLING_TOOL_DIMENSION(10.0,0.3490658503988659,0.0,40.0,0.0,0.0,0.0,0.0);
#508=CUTTING_COMPONENT(70.0,#509,$,$,$);
#509=MATERIAL('P','CARBIDE',());
#510=MILLING_TECHNOLOGY(0.245,.TCP.,110.0,3501.4087480216976,$,.F.,.F.,.F.,$);
#511=MILLING_MACHINE_FUNCTIONS(.T.,$,$,.T.,$,(),.T.,$,$,());
#512=CARTESIAN_POINT('start point',(0.0,0.0,0.0));
#513=DRILLING_TYPE_STRATEGY($,$,$,$,$,$);
#514=MACHINING_WORKINGSTEP('Plane Base Hole_RGH',#15,#490,#504,$);
#515=BOTTOM_AND_SIDE_ROUGH_MILLING($,$,'Bottom And Side Rough
Milling',5.0,#523,
    #516,#521,#522,$,#524,#525,#528,1.0,7.5,0.0,0.0);
#516=MILLING_CUTTING_TOOL('SF20',#517,(#519),$,$,$);
#517=FACEMILL(#518,4,.LEFT.,$,120.0);
#518=MILLING_TOOL_DIMENSION(10.0,0.0,0.0,40.0,0.0,0.0,0.0,0.0);
#519=CUTTING_COMPONENT(70.0,#520,$,$,$);
#520=MATERIAL('P','CARBIDE',());
#521=MILLING_TECHNOLOGY(0.0235,.TCP.,145.0,4615.493349664965,$,.F.,.F.,.F.,$);
#522=MILLING_MACHINE_FUNCTIONS(.T.,$,$,.T.,$,(),.T.,$,$,());
#523=CARTESIAN_POINT('start point',(0.0,0.0,8.180148828668989));
#524=PLUNGE_TOOLAXIS(#526);
#525=PLUNGE_TOOLAXIS(#527);
#526=DIRECTION('approach strategy direction',(0.0,-1.0,0.0));
#527=DIRECTION('retract strategy direction',(0.0,-1.0,-0.0));
#528=CONTOUR_PARALLEL(2.0,.T.,.CCW.,.CONVENTIONAL.);
#529=MACHINING_WORKINGSTEP('Plane Base Hole_RGH',#15,#490,#515,$);
#530=CENTER_DRILLING($,$,'Center Drilling',5.0,#531,#532,#537,#538,$,10.0,$,$,$,
    #539);
#531=CARTESIAN_POINT('start point',(0.0,0.0,0.0));
#532=MILLING_CUTTING_TOOL('center drill',#533,(#535),$,$,$);
#533=CENTER_DRILL(#534,2,.LEFT.,$,25.0);
#534=MILLING_TOOL_DIMENSION(10.0,0.5235987755982988,0.0,10.0,0.0,0.0,0.0,0.0);
#535=CUTTING_COMPONENT(50.0,#536,$,$,$);
#536=MATERIAL('P','CARBIDE',());
#537=MILLING_TECHNOLOGY(0.245,.TCP.,110.0,3501.4087480216976,$,.F.,.F.,.F.,$);
#538=MILLING_MACHINE_FUNCTIONS(.T.,$,$,.T.,$,(),.T.,$,$,());
#539=DRILLING_TYPE_STRATEGY($,$,$,$,$,$);
#540=ROUND_HOLE('Plane Base Hole',#2,(#530,#554,#565),#541,#551,#545,$,#552);
#541=AXIS2_PLACEMENT_3D('Plane Base Hole placement',#542,#543,#544);
#542=CARTESIAN_POINT('',(190.0,0.0,100.0));
#543=DIRECTION('',(0.0,-1.0,0.0));
#544=DIRECTION('',(1.0,0.0,0.0));
#545=TOLERANCED_LENGTH_MEASURE(10.0,#546);
```

```
#546=PLUS_MINUS_VALUE(0.05,0.05,4);
#547=AXIS2_PLACEMENT_3D('hole depth',#548,#549,#550);
#548=CARTESIAN_POINT('',(0.0,0.0,-10.0));
#549=DIRECTION('',(0.0,-1.0,0.0));
#550=DIRECTION('',(1.0,0.0,0.0));
#551=PLANE('hole depth plane',#547);
#552=FLAT_HOLE_BOTTOM();
#553=MACHINING_WORKINGSTEP('Plane Base Hole_RGH',#15,#540,#530,$);
#554=DRILLING($,$,'Drilling',5.0,#562,#555,#560,#561,$,10.0,$,$,$,#563);
#555=MILLING_CUTTING_TOOL('twist drill',#556,(#558),$,$,$);
#556=TWIST_DRILL(#557,2,.LEFT.,$,60.0);
#557=MILLING_TOOL_DIMENSION(10.0,0.3490658503988659,0.0,40.0,0.0,0.0,0.0);
#558=CUTTING_COMPONENT(70.0,#559,$,$,$);
#559=MATERIAL('P','CARBIDE',());
#560=MILLING_TECHNOLOGY(0.245,.TCP.,110.0,3501.4087480216976,$,.F.,.F.,.F.,$);
#561=MILLING_MACHINE_FUNCTIONS(.T.,$,$,.T.,$,(),.T.,$,$,());
#562=CARTESIAN_POINT('start point',(0.0,0.0,0.0));
#563=DRILLING_TYPE_STRATEGY($,$,$,$,$,$);
#564=MACHINING_WORKINGSTEP('Plane Base Hole_RGH',#15,#540,#554,$);
#565=BOTTOM_AND_SIDE_ROUGH_MILLING($,$,'Bottom And Side Rough
Milling',5.0,#573,
    #566,#571,#572,$,#574,#575,#578,1.0,7.5,0.0,0.0);
#566=MILLING_CUTTING_TOOL('SF20',#567,(#569),$,$,$);
#567=FACEMILL(#568,4,.LEFT.,$,120.0);
#568=MILLING_TOOL_DIMENSION(10.0,0.0,0.0,40.0,0.0,0.0,0.0);
#569=CUTTING_COMPONENT(70.0,#570,$,$,$);
#570=MATERIAL('P','CARBIDE',());
#571=MILLING_TECHNOLOGY(0.0235,.TCP.,145.0,4615.493349664965,$,.F.,.F.,.F.,$);
#572=MILLING_MACHINE_FUNCTIONS(.T.,$,$,.T.,$,(),.T.,$,$,());
#573=CARTESIAN_POINT('start point',(0.0,0.0,8.180148828668989));
#574=PLUNGE_TOOLAXIS(#576);
#575=PLUNGE_TOOLAXIS(#577);
#576=DIRECTION('approach strategy direction',(0.0,-1.0,0.0));
#577=DIRECTION('retract strategy direction',(0.0,-1.0,-0.0));
#578=CONTOUR_PARALLEL(2.0,.T.,.CCW.,.CONVENTIONAL.);
#579=MACHINING_WORKINGSTEP('Plane Base Hole_RGH',#15,#540,#565,$);
#580=WORKPLAN('workplan setup XZ',(#329,#364,#379,#414,#429,
    #464,#479,#514,#529,#564,#579),$,#10,$);
#581=CENTER_DRILLING($,$,'Center Drilling',5.0,#582,#583,#588,#589,$,10.0,$,$,$,
    #590);
#582=CARTESIAN_POINT('start point',(0.0,0.0,0.0));
#583=MILLING_CUTTING_TOOL('center drill',#584,(#586),$,$,$);
#584=CENTER_DRILL(#585,2,.LEFT.,$,25.0);
#585=MILLING_TOOL_DIMENSION(10.0,0.5235987755982988,0.0,10.0,0.0,0.0,0.0);
#586=CUTTING_COMPONENT(50.0,#587,$,$,$);
#587=MATERIAL('P','CARBIDE',());
#588=MILLING_TECHNOLOGY(0.245,.TCP.,110.0,3501.4087480216976,$,.F.,.F.,.F.,$);
#589=MILLING_MACHINE_FUNCTIONS(.T.,$,$,.T.,$,(),.T.,$,$,());
#590=DRILLING_TYPE_STRATEGY($,$,$,$,$,$);
#591=ROUND_HOLE('Plane Base Hole',#2,(#581,#605,#616),#592,#602,#596,$,#603);
#592=AXIS2_PLACEMENT_3D('Plane Base Hole placement',#593,#594,#595);
#593=CARTESIAN_POINT('',(0.0,100.0,100.0));
#594=DIRECTION('',(-1.0,0.0,0.0));
```

```
#595=DIRECTION('',(0.0,0.0,1.0));
#596=TOLERANCED_LENGTH_MEASURE(100.0,#597);
#597=PLUS_MINUS_VALUE(0.05,0.05,4);
#598=AXIS2_PLACEMENT_3D('hole depth',#599,#600,#601);
#599=CARTESIAN_POINT('',(0.0,0.0,-10.0));
#600=DIRECTION('',(-1.0,0.0,0.0));
#601=DIRECTION('',(0.0,0.0,1.0));
#602=PLANE('hole depth plane',#598);
#603=FLAT_HOLE_BOTTOM();
#604=MACHINING_WORKINGSTEP('Plane Base Hole_RGH',#15,#591,#581,$);
#605=DRILLING($,$,'Drilling',5.0,#613,#606,#611,#612,$,10.0,$,$,$,#614);
#606=MILLING_CUTTING_TOOL('twist drill',#607,(#609),$,$,$);
#607=TWIST_DRILL(#608,2,.LEFT.,$,60.0);
#608=MILLING_TOOL_DIMENSION(10.0,0.3490658503988659,0.0,40.0,0.0,0.0,0.0,0.0);
#609=CUTTING_COMPONENT(70.0,#610,$,$,$);
#610=MATERIAL('P','CARBIDE',());
#611=MILLING_TECHNOLOGY(0.245,.TCP.,110.0,3501.4087480216976,$,.F.,.F.,.F.,$);
#612=MILLING_MACHINE_FUNCTIONS(.T.,$,$,.T.,$,(),.T.,$,$,());
#613=CARTESIAN_POINT('start point',(0.0,0.0,0.0));
#614=DRILLING_TYPE_STRATEGY($,$,$,$,$,$);
#615=MACHINING_WORKINGSTEP('Plane Base Hole_RGH',#15,#591,#605,$);
#616=BOTTOM_AND_SIDE_ROUGH_MILLING($,$,'Bottom And Side Rough
Milling',5.0,#624,
    #617,#622,#623,$,#625,#626,#629,2.0,15.0,0.0,0.0);
#617=MILLING_CUTTING_TOOL('SF20',#618,(#620),$,$,$);
#618=FACEMILL(#619,4,.LEFT.,$,60.0);
#619=MILLING_TOOL_DIMENSION(20.0,0.0,0.0,40.0,0.0,0.0,0.0,0.0);
#620=CUTTING_COMPONENT(70.0,#621,$,$,$);
#621=MATERIAL('P','CARBIDE',());
#622=MILLING_TECHNOLOGY(0.073,.TCP.,145.0,2307.7466748324823,$,.F.,.F.,.F.,$);
#623=MILLING_MACHINE_FUNCTIONS(.T.,$,$,.T.,$,(),.T.,$,$,());
#624=CARTESIAN_POINT('start point',(0.0,5.0,0.0));
#625=PLUNGE_TOOLAXIS(#627);
#626=PLUNGE_TOOLAXIS(#628);
#627=DIRECTION('approach strategy direction',(-1.0,0.0,0.0));
#628=DIRECTION('retract strategy direction',(-1.0,0.0,-0.0));
#629=CONTOUR_PARALLEL(4.0,.T.,.CCW.,.CONVENTIONAL.);
#630=MACHINING_WORKINGSTEP('Plane Base Hole_RGH',#15,#591,#616,$);
#631=CENTER_DRILLING($,$,'Center Drilling',5.0,#632,#633,#638,#639,$,10.0,$,$,$,
    #640);
#632=CARTESIAN_POINT('start point',(0.0,0.0,0.0));
#633=MILLING_CUTTING_TOOL('center drill',#634,(#636),$,$,$);
#634=CENTER_DRILL(#635,2,.LEFT.,$,25.0);
#635=MILLING_TOOL_DIMENSION(10.0,0.5235987755982988,0.0,10.0,0.0,0.0,0.0,0.0);
#636=CUTTING_COMPONENT(50.0,#637,$,$,$);
#637=MATERIAL('P','CARBIDE',());
#638=MILLING_TECHNOLOGY(0.245,.TCP.,110.0,3501.4087480216976,$,.F.,.F.,.F.,$);
#639=MILLING_MACHINE_FUNCTIONS(.T.,$,$,.T.,$,(),.T.,$,$,());
#640=DRILLING_TYPE_STRATEGY($,$,$,$,$,$);
#641=ROUND_HOLE('Plane Base Hole',#2,(#631,#655,#666),#642,#652,#646,$,#653);
#642=AXIS2_PLACEMENT_3D('Plane Base Hole placement',#643,#644,#645);
#643=CARTESIAN_POINT('',(0.0,100.0,10.0));
#644=DIRECTION('',(-1.0,0.0,0.0));
```

```
#645=DIRECTION('',(0.0,0.0,1.0));
#646=TOLERANCED_LENGTH_MEASURE(10.0,#647);
#647=PLUS_MINUS_VALUE(0.05,0.05,4);
#648=AXIS2_PLACEMENT_3D('hole depth',#649,#650,#651);
#649=CARTESIAN_POINT('',(0.0,0.0,-10.0));
#650=DIRECTION('',(-1.0,0.0,0.0));
#651=DIRECTION('',(0.0,0.0,1.0));
#652=PLANE('hole depth plane',#648);
#653=FLAT_HOLE_BOTTOM();
#654=MACHINING_WORKINGSTEP('Plane Base Hole_RGH',#15,#641,#631,$);
#655=DRILLING($,$,'Drilling',5.0,#663,#656,#661,#662,$,10.0,$,$,$,#664);
#656=MILLING_CUTTING_TOOL('twist drill',#657,(#659),$,$,$);
#657=TWIST_DRILL(#658,2,.LEFT.,$,60.0);
#658=MILLING_TOOL_DIMENSION(10.0,0.3490658503988659,0.0,40.0,0.0,0.0,0.0);
#659=CUTTING_COMPONENT(70.0,#660,$,$,$);
#660=MATERIAL('P','CARBIDE',());
#661=MILLING_TECHNOLOGY(0.245,.TCP.,110.0,3501.4087480216976,$,.F.,.F.,.F.,$);
#662=MILLING_MACHINE_FUNCTIONS(.T.,$,$,.T.,$,(),.T.,$,$,());
#663=CARTESIAN_POINT('start point',(0.0,0.0,0.0));
#664=DRILLING_TYPE_STRATEGY($,$,$,$,$,$);
#665=MACHINING_WORKINGSTEP('Plane Base Hole_RGH',#15,#641,#655,$);
#666=BOTTOM_AND_SIDE_ROUGH_MILLING($,$,'Bottom And Side Rough
Milling',5.0,#674,
    #667,#672,#673,$,#675,#676,#679,1.0,7.5,0.0,0.0);
#667=MILLING_CUTTING_TOOL('SF20',#668,(#670),$,$,$);
#668=FACEMILL(#669,4,.LEFT.,$,120.0);
#669=MILLING_TOOL_DIMENSION(10.0,0.0,0.0,40.0,0.0,0.0,0.0);
#670=CUTTING_COMPONENT(70.0,#671,$,$,$);
#671=MATERIAL('P','CARBIDE',());
#672=MILLING_TECHNOLOGY(0.0235,.TCP.,145.0,4615.493349664965,$,.F.,.F.,.F.,$);
#673=MILLING_MACHINE_FUNCTIONS(.T.,$,$,.T.,$,(),.T.,$,$,());
#674=CARTESIAN_POINT('start point',(0.0,0.0,8.180148828668989));
#675=PLUNGE_TOOLAXIS(#677);
#676=PLUNGE_TOOLAXIS(#678);
#677=DIRECTION('approach strategy direction',(-1.0,0.0,0.0));
#678=DIRECTION('retract strategy direction',(-1.0,0.0,-0.0));
#679=CONTOUR_PARALLEL(2.0,.T.,.CCW.,.CONVENTIONAL.);
#680=MACHINING_WORKINGSTEP('Plane Base Hole_RGH',#15,#641,#666,$);
#681=CENTER_DRILLING($,$,'Center Drilling',5.0,#682,#683,#688,#689,$,10.0,$,$,$,
    #690);
#682=CARTESIAN_POINT('start point',(0.0,0.0,0.0));
#683=MILLING_CUTTING_TOOL('center drill',#684,(#686),$,$,$);
#684=CENTER_DRILL(#685,2,.LEFT.,$,25.0);
#685=MILLING_TOOL_DIMENSION(10.0,0.5235987755982988,0.0,10.0,0.0,0.0,0.0);
#686=CUTTING_COMPONENT(50.0,#687,$,$,$);
#687=MATERIAL('P','CARBIDE',());
#688=MILLING_TECHNOLOGY(0.245,.TCP.,110.0,3501.4087480216976,$,.F.,.F.,.F.,$);
#689=MILLING_MACHINE_FUNCTIONS(.T.,$,$,.T.,$,(),.T.,$,$,());
#690=DRILLING_TYPE_STRATEGY($,$,$,$,$,$);
#691=ROUND_HOLE('Plane Base Hole',#2,(#681,#705,#716),#692,#702,#696,$,#703);
#692=AXIS2_PLACEMENT_3D('Plane Base Hole placement',#693,#694,#695);
#693=CARTESIAN_POINT('',(0.0,100.0,190.0));
#694=DIRECTION('',(-1.0,0.0,0.0));
```

```
#695=DIRECTION('',(0.0,0.0,1.0));
#696=TOLERANCED_LENGTH_MEASURE(10.0,#697);
#697=PLUS_MINUS_VALUE(0.05,0.05,4);
#698=AXIS2_PLACEMENT_3D('hole depth',#699,#700,#701);
#699=CARTESIAN_POINT('',(0.0,0.0,-10.0));
#700=DIRECTION('',(-1.0,0.0,0.0));
#701=DIRECTION('',(0.0,0.0,1.0));
#702=PLANE('hole depth plane',#698);
#703=FLAT_HOLE_BOTTOM();
#704=MACHINING_WORKINGSTEP('Plane Base Hole_RGH',#15,#691,#681,$);
#705=DRILLING($,$,'Drilling',5.0,#713,#706,#711,#712,$,10.0,$,$,$,#714);
#706=MILLING_CUTTING_TOOL('twist drill',#707,(#709),$,$,$);
#707=TWIST_DRILL(#708,2,.LEFT.,$,60.0);
#708=MILLING_TOOL_DIMENSION(10.0,0.3490658503988659,0.0,40.0,0.0,0.0,0.0);
#709=CUTTING_COMPONENT(70.0,#710,$,$,$);
#710=MATERIAL('P','CARBIDE',());
#711=MILLING_TECHNOLOGY(0.245,.TCP.,110.0,3501.4087480216976,$,.F.,.F.,.F.,$);
#712=MILLING_MACHINE_FUNCTIONS(.T.,$,$,.T.,$,(),.T.,$,$,());
#713=CARTESIAN_POINT('start point',(0.0,0.0,0.0));
#714=DRILLING_TYPE_STRATEGY($,$,$,$,$,$);
#715=MACHINING_WORKINGSTEP('Plane Base Hole_RGH',#15,#691,#705,$);
#716=BOTTOM_AND_SIDE_ROUGH_MILLING($,$,'Bottom And Side Rough
Milling',5.0,#724,
    #717,#722,#723,$,#725,#726,#729,1.0,7.5,0.0,0.0);
#717=MILLING_CUTTING_TOOL('SF20',#718,(#720),$,$,$);
#718=FACEMILL(#719,4,.LEFT.,$,120.0);
#719=MILLING_TOOL_DIMENSION(10.0,0.0,0.0,40.0,0.0,0.0,0.0);
#720=CUTTING_COMPONENT(70.0,#721,$,$,$);
#721=MATERIAL('P','CARBIDE',());
#722=MILLING_TECHNOLOGY(0.0235,.TCP.,145.0,4615.493349664965,$,.F.,.F.,.F.,$);
#723=MILLING_MACHINE_FUNCTIONS(.T.,$,$,.T.,$,(),.T.,$,$,());
#724=CARTESIAN_POINT('start point',(0.0,0.0,8.180148828668989));
#725=PLUNGE_TOOLAXIS(#727);
#726=PLUNGE_TOOLAXIS(#728);
#727=DIRECTION('approach strategy direction',(-1.0,0.0,0.0));
#728=DIRECTION('retract strategy direction',(-1.0,0.0,-0.0));
#729=CONTOUR_PARALLEL(2.0,.T.,.CCW.,.CONVENTIONAL.);
#730=MACHINING_WORKINGSTEP('Plane Base Hole_RGH',#15,#691,#716,$);
#731=WORKPLAN('workplan setup ZY',(#615,#630,#665,#680,#715,#730),
    $,#10,$);
#732=WORKPLAN('Main workplan',(#277,#580,#731),$,$,$);
ENDSEC;
END-ISO-10303-21;
```

APÊNDICE C – Dados Coletados das Trajetórias

Tabela C.0.1 Dados peça exemplo 1- robô Adept One

#	Trajetória percorrida			Trajetória alvo			Erro		
	X	Y	Z	X	Y	Z	X	Y	Z
1	300,00001	309,99993	775,00000	300	310	775	-1,47E-05	7,33E-05	-4,56E-07
2	389,99992	319,99998	760,00001	390	320	760	8,36E-05	2,03E-05	-1,43E-05
3	390,00000	320,00000	750,00008	390	320	750	6,38E-10	1,55E-10	-7,63E-05
4	390,00000	339,99992	750,00000	390	340	750	1,84E-06	7,61E-05	-2,91E-10
5	390,00000	359,99992	750,00000	390	360	750	1,82E-06	7,62E-05	0,00E+00
6	390,00000	379,99992	750,00000	390	380	750	1,80E-06	7,63E-05	0,00E+00
7	390,00000	399,99992	750,00000	390	400	750	1,78E-06	7,64E-05	0,00E+00
8	390,00000	419,99992	750,00000	390	420	750	1,77E-06	7,65E-05	0,00E+00
9	375,00006	420,00000	750,00000	375	420	750	-5,73E-05	1,13E-06	0,00E+00
200	385,00000	410,00000	742,50006	385	410	742,5	2,17E-10	8,64E-12	-5,72E-05
201	365,00008	410,00000	742,50000	365	410	742,5	-7,65E-05	2,01E-06	-2,18E-10
202	355,00008	410,00000	742,50000	355	410	742,5	-7,64E-05	1,00E-06	0,00E+00
203	355,00000	390,00008	742,50000	355	390	742,5	1,75E-06	-7,61E-05	0,00E+00
204	374,99992	390,00000	742,50000	375	390	742,5	7,60E-05	2,03E-06	0,00E+00
205	384,99992	390,00000	742,50000	385	390	742,5	7,62E-05	1,01E-06	0,00E+00
206	355,00006	390,00000	742,50000	355	390	742,5	-5,75E-05	2,29E-06	0,00E+00
207	355,00000	370,00008	742,50000	355	370	742,5	1,78E-06	-7,62E-05	0,00E+00
208	355,00000	370,00008	742,50000	355	370	742,5	1,78E-06	-7,62E-05	0,00E+00
209	374,99992	370,00000	742,50000	375	370	742,5	7,60E-05	2,06E-06	0,00E+00
400	355,00008	410,00000	725,00000	355	410	725	-7,64E-05	1,00E-06	0,00E+00
401	355,00000	410,00000	739,99994	355	410	740	-2,92E-10	3,81E-12	5,72E-05
402	384,99994	410,00000	740,00000	385	410	740	5,69E-05	2,26E-06	1,09E-10
403	385,00000	410,00000	725,00006	385	410	725	2,17E-10	8,64E-12	-5,72E-05
404	365,00008	410,00000	725,00000	365	410	725	-7,65E-05	2,01E-06	-2,19E-10
405	355,00008	410,00000	725,00000	355	410	725	-7,64E-05	1,00E-06	0,00E+00
406	355,00000	390,00008	725,00000	355	390	725	1,75E-06	-7,61E-05	0,00E+00
407	374,99992	390,00000	725,00000	375	390	725	7,60E-05	2,03E-06	0,00E+00
408	384,99992	390,00000	725,00000	385	390	725	7,62E-05	1,01E-06	0,00E+00
409	355,00006	390,00000	725,00000	355	390	725	-5,75E-05	2,29E-06	0,00E+00
830	385,00000	389,99992	715,00000	385	390	715	1,79E-06	7,63E-05	0,00E+00
831	385,00000	409,99992	715,00000	385	410	715	1,77E-06	7,64E-05	0,00E+00
832	365,00008	410,00000	715,00000	365	410	715	-7,65E-05	2,01E-06	0,00E+00
833	355,00008	410,00000	715,00000	355	410	715	-7,64E-05	1,00E-06	0,00E+00
834	369,14208	395,85792	715,00000	369,14	395,86	715	5,50E-05	-5,30E-05	0,00E+00
835	369,99995	395,00005	715,00000	370	395	715	5,24E-05	-5,23E-05	0,00E+00
836	370,00000	375,00008	715,00000	370	375	715	1,78E-06	-7,62E-05	0,00E+00
837	370,00000	365,00008	715,00000	370	365	715	8,98E-07	-7,63E-05	0,00E+00
838	370,00000	384,99992	715,00000	370	385	715	1,79E-06	7,63E-05	0,00E+00
839	370,00000	394,99992	715,00000	370	395	715	8,87E-07	7,63E-05	0,00E+00

NOTA: Dados em mm

Tabela C.0.2 Dados peça exemplo 1- robô ABB IRB-140

#	Trajetória percorrida			Trajetória alvo			Erro		
	X	Y	Z	X	Y	Z	X	Y	Z
1	399,99997	-190,00002	530,00005	400	-190	530	2,54E-05	2,32E-05	-5,26E-05
2	489,99991	-180,00000	515,00001	490	-180	515	8,61E-05	1,97E-06	-5,69E-06
3	490,00000	-180,00000	505,00008	490	-180	505	8,64E-07	-3,17E-07	-7,65E-05
4	490,00000	-160,00008	505,00000	490	-160	505	1,60E-06	7,67E-05	1,91E-07
5	490,00000	-140,00008	505,00000	490	-140	505	1,59E-06	7,67E-05	1,54E-07
6	490,00000	-120,00008	505,00000	490	-120	505	1,59E-06	7,66E-05	1,19E-07
7	490,00000	-100,00008	505,00000	490	-100	505	1,58E-06	7,66E-05	8,80E-08
8	490,00000	-80,00008	505,00000	490	-80	505	1,57E-06	7,66E-05	6,05E-08
9	475,00006	-80,00000	505,00000	475	-80	505	-5,70E-05	-3,19E-07	1,02E-06
200	485,00000	-90,00000	497,50006	485	-90	497,5	1,06E-06	-1,97E-07	-5,75E-05
201	465,00008	-90,00000	497,50000	465	-90	497,5	-7,60E-05	-6,47E-07	1,78E-06
202	455,00008	-90,00000	497,50000	455	-90	497,5	-7,62E-05	-3,34E-07	9,16E-07
203	455,00000	-109,99992	497,50000	455	-110	497,5	1,68E-06	-7,60E-05	8,90E-08
204	474,99992	-110,00000	497,50000	475	-110	497,5	7,65E-05	-7,92E-07	1,77E-06
205	484,99992	-110,00000	497,50000	485	-110	497,5	7,65E-05	-3,84E-07	8,65E-07
206	455,00006	-110,00000	497,50000	455	-110	497,5	-5,69E-05	-8,84E-07	1,97E-06
207	455,00000	-129,99992	497,50000	455	-130	497,5	1,69E-06	-7,59E-05	1,24E-07
208	455,00000	-129,99992	497,50000	455	-130	497,5	1,69E-06	-7,59E-05	1,24E-07
209	474,99992	-130,00000	497,50000	475	-130	497,5	7,65E-05	-9,15E-07	1,72E-06
400	455,00008	-90,00000	480,00000	455	-90	480	-7,62E-05	-3,38E-07	8,85E-07
401	455,00000	-90,00000	494,99994	455	-90	495	1,11E-06	-2,19E-07	5,69E-05
402	484,99994	-90,00000	495,00000	485	-90	495	5,75E-05	-7,36E-07	2,02E-06
403	485,00000	-90,00000	480,00006	485	-90	480	1,06E-06	-1,96E-07	-5,75E-05
404	465,00008	-90,00000	480,00000	465	-90	480	-7,59E-05	-6,54E-07	1,72E-06
405	455,00008	-90,00000	480,00000	455	-90	480	-7,62E-05	-3,38E-07	8,85E-07
406	455,00000	-109,99992	480,00000	455	-110	480	1,69E-06	-7,60E-05	8,61E-08
407	474,99992	-110,00000	480,00000	475	-110	480	7,66E-05	-8,01E-07	1,71E-06
408	484,99992	-110,00000	480,00000	485	-110	480	7,65E-05	-3,88E-07	8,32E-07
409	455,00006	-110,00000	480,00000	455	-110	480	-5,69E-05	-8,93E-07	1,90E-06
830	485,00000	-110,00008	470,00000	485	-110	470	1,60E-06	7,66E-05	9,75E-08
831	485,00000	-90,00008	470,00000	485	-90	470	1,59E-06	7,66E-05	6,98E-08
832	465,00008	-90,00000	470,00000	465	-90	470	-7,59E-05	-6,58E-07	1,68E-06
833	455,00008	-90,00000	470,00000	455	-90	470	-7,62E-05	-3,40E-07	8,65E-07
834	469,14208	-104,14208	470,00000	469,14	-104,14	470	5,50E-05	-5,27E-05	1,25E-06
835	469,99995	-104,99995	470,00000	470	-105	470	5,24E-05	-5,23E-05	7,33E-08
836	470,00000	-124,99992	470,00000	470	-125	470	1,64E-06	-7,59E-05	9,87E-08
837	470,00000	-134,99992	470,00000	470	-135	470	8,23E-07	-7,61E-05	6,13E-08
838	470,00000	-115,00008	470,00000	470	-115	470	1,65E-06	7,67E-05	1,15E-07
839	470,00000	-105,00008	470,00000	470	-105	470	8,21E-07	7,65E-05	4,53E-08

NOTA: Dados em mm

Tabela C.0.3 Dados peça exemplo 1- robô Tricept 806

#	Trajetória percorrida			Trajetória alvo			Erro		
	X	Y	Z	X	Y	Z	X	Y	Z
1	50,00005	59,99992	-1870,00000	50	60	-1870	-4,72E-05	8,02E-05	3,82E-06
2	140,00010	70,00005	-1884,99996	140	70	-1885	-9,90E-05	-4,67E-05	-4,23E-05
3	139,99993	69,99996	-1895,00005	140	70	-1895	7,21E-05	3,57E-05	5,42E-05
4	140,00001	90,00009	-1894,99999	140	90	-1895	-1,16E-05	-8,97E-05	-1,50E-05
5	140,00001	110,00009	-1894,99996	140	110	-1895	-6,58E-06	-9,25E-05	-4,09E-05
6	140,00001	130,00009	-1895,00003	140	130	-1895	-9,02E-06	-9,29E-05	3,14E-05
7	140,00001	150,00009	-1895,00001	140	150	-1895	-7,94E-06	-9,12E-05	5,00E-06
8	140,00000	170,00010	-1895,00002	140	170	-1895	2,04E-06	-9,71E-05	2,41E-05
9	124,99993	170,00001	-1894,99996	125	170	-1895	6,90E-05	-5,01E-06	-3,59E-05
200	134,99995	159,99994	-1902,50004	135	160	-1902,5	5,24E-05	5,91E-05	3,92E-05
201	114,99990	160,00001	-1902,50004	115	160	-1902,5	9,61E-05	-5,59E-06	3,86E-05
202	104,99991	160,00000	-1902,50004	105	160	-1902,5	9,35E-05	3,02E-06	4,33E-05
203	105,00001	139,99990	-1902,49995	105	140	-1902,5	-9,20E-06	9,64E-05	-4,89E-05
204	125,00009	140,00000	-1902,49996	125	140	-1902,5	-8,92E-05	-3,50E-06	-4,40E-05
205	135,00010	140,00001	-1902,49996	135	140	-1902,5	-9,51E-05	-6,19E-06	-3,65E-05
206	104,99993	140,00000	-1902,49995	105	140	-1902,5	6,82E-05	-4,94E-06	-4,89E-05
207	105,00001	119,99991	-1902,49998	105	120	-1902,5	-5,09E-06	9,25E-05	-2,07E-05
208	105,00001	119,99991	-1902,49998	105	120	-1902,5	-5,09E-06	9,25E-05	-2,07E-05
209	125,00009	120,00000	-1902,49998	125	120	-1902,5	-8,79E-05	-6,55E-07	-2,00E-05
400	104,99991	160,00001	-1920,00003	105	160	-1920	9,26E-05	-7,08E-06	3,14E-05
401	105,00003	160,00005	-1904,99997	105	160	-1905	-3,42E-05	-5,42E-05	-3,17E-05
402	135,00006	160,00001	-1904,99998	135	160	-1905	-6,42E-05	-9,39E-06	-1,53E-05
403	134,99996	159,99995	-1919,99997	135	160	-1920	4,37E-05	5,47E-05	-2,74E-05
404	114,99991	160,00000	-1919,99999	115	160	-1920	8,90E-05	-3,31E-06	-1,00E-05
405	104,99991	160,00001	-1920,00003	105	160	-1920	9,26E-05	-7,08E-06	3,14E-05
406	105,00000	139,99990	-1919,99998	105	140	-1920	-1,98E-06	9,74E-05	-2,10E-05
407	125,00009	140,00000	-1919,99998	125	140	-1920	-9,25E-05	-3,30E-06	-2,49E-05
408	135,00009	140,00000	-1920,00001	135	140	-1920	-9,40E-05	-4,77E-06	9,43E-06
409	104,99993	140,00000	-1919,99998	105	140	-1920	7,46E-05	-2,87E-06	-2,10E-05
830	135,00000	140,00008	-1930,00003	135	140	-1930	-3,83E-06	-8,42E-05	3,39E-05
831	135,00000	160,00009	-1930,00003	135	160	-1930	-8,70E-07	-8,87E-05	3,40E-05
832	114,99990	160,00001	-1929,99998	115	160	-1930	9,63E-05	-7,95E-06	-1,77E-05
833	104,99991	160,00000	-1929,99997	105	160	-1930	8,88E-05	-6,10E-08	-2,79E-05
834	119,14220	145,85780	-1930,00006	119	146	-1930	-6,32E-05	6,45E-05	5,74E-05
835	120,00007	144,99994	-1929,99999	120	145	-1930	-6,64E-05	5,82E-05	-1,43E-05
836	120,00001	124,99990	-1930,00003	120	125	-1930	-5,16E-06	9,97E-05	2,81E-05
837	120,00000	114,99990	-1930,00001	120	115	-1930	6,06E-07	9,63E-05	6,61E-06
838	120,00000	135,00009	-1929,99999	120	135	-1930	-2,64E-06	-8,98E-05	-5,81E-06
839	120,00001	145,00010	-1929,99999	120	145	-1930	-5,14E-06	-9,61E-05	-1,43E-05

NOTA: Dados em mm

Tabela C.0.4 Dados peça exemplo 2- robô Adept One

#	Trajetória percorrida			Trajetória alvo			Erro		
	X	Y	Z	X	Y	Z	X	Y	Z
1	300,00001	309,99993	875,00000	300	310	875	-1,47E-05	7,33E-05	-4,48E-07
2	319,99996	329,99996	855,00005	320	330	855	3,83E-05	3,90E-05	-5,09E-05
3	320,00000	330,00000	850,00008	320	330	850	5,84E-10	5,96E-10	-7,89E-05
4	320,00000	330,00000	840,00008	320	330	840	0,00E+00	0,00E+00	-7,67E-05
5	320,00000	330,00000	849,99992	320	330	850	0,00E+00	0,00E+00	7,82E-05
6	320,00000	330,00000	854,99992	320	330	855	0,00E+00	0,00E+00	8,19E-05
7	320,00000	330,00000	854,99992	320	330	855	0,00E+00	0,00E+00	8,19E-05
8	300,00000	310,00000	874,99992	300	310	875	-4,75E-06	-4,66E-06	7,72E-05
9	300,00000	310,00000	874,99992	300	310	875	-4,75E-06	-4,66E-06	7,72E-05
1010	389,20000	404,80006	840,00000	389,2	404,8	840	1,13E-06	-6,09E-05	0,00E+00
1011	389,20000	399,20009	840,00000	389,2	399,2	840	5,57E-07	-8,54E-05	0,00E+00
1012	399,99992	409,99992	840,00000	400	410	840	8,32E-05	8,38E-05	0,00E+00
1013	400,00000	410,00000	844,99992	400	410	845	1,27E-09	1,28E-09	7,69E-05
1014	338,00006	348,00005	854,99999	338	348	855	-5,67E-05	-5,39E-05	1,04E-05
1015	300,00001	310,00001	874,99992	300	310	875	-8,97E-06	-8,64E-06	7,72E-05
1016	299,99997	534,99994	745,00000	300	535	745	2,59E-05	5,68E-05	-1,47E-07
1017	337,99996	348,00008	750,00000	338	348	750	3,79E-05	-8,35E-05	2,38E-06
1018	338,00000	348,00000	745,00008	338	348	745	5,78E-10	-1,27E-09	-7,63E-05
1019	353,99994	348,00000	745,00000	354	348	745	6,08E-05	1,35E-06	-2,91E-10
2250	490,00000	410,00000	741,00006	490	410	741	0,00E+00	0,00E+00	-6,10E-05
2251	490,00000	410,00000	736,00008	490	410	736	0,00E+00	0,00E+00	-7,63E-05
2252	490,00000	410,00000	740,99992	490	410	741	0,00E+00	0,00E+00	7,63E-05
2253	490,00000	410,00000	740,00006	490	410	740	0,00E+00	0,00E+00	-6,10E-05
2254	490,00000	410,00000	735,00008	490	410	735	0,00E+00	0,00E+00	-7,63E-05
2255	490,00000	410,00000	739,99992	490	410	740	0,00E+00	0,00E+00	7,63E-05
2256	490,00000	410,00000	749,99992	490	410	750	0,00E+00	0,00E+00	7,63E-05
2257	300,00008	534,99992	745,00000	300	535	745	-7,95E-05	7,88E-05	-2,38E-06
2258	525,00001	309,99994	745,00002	525	310	745	-1,39E-05	6,04E-05	-1,89E-05
2259	400,00005	409,99994	750,00000	400	410	750	-5,36E-05	5,80E-05	2,38E-06
3090	490,00000	410,00000	744,00006	490	410	744	0,00E+00	0,00E+00	-6,10E-05
3091	490,00000	410,00000	739,00008	490	410	739	0,00E+00	0,00E+00	-7,63E-05
3092	490,00000	410,00000	743,99992	490	410	744	0,00E+00	0,00E+00	7,63E-05
3093	490,00000	410,00000	743,00006	490	410	743	0,00E+00	0,00E+00	-6,10E-05
3094	490,00000	410,00000	738,00008	490	410	738	0,00E+00	0,00E+00	-7,63E-05
3095	490,00000	410,00000	742,99992	490	410	743	0,00E+00	0,00E+00	7,63E-05
3096	490,00000	410,00000	742,00006	490	410	742	0,00E+00	0,00E+00	-6,10E-05
3097	490,00000	410,00000	737,00008	490	410	737	0,00E+00	0,00E+00	-7,63E-05
3098	490,00000	410,00000	741,99992	490	410	742	0,00E+00	0,00E+00	7,63E-05
3099	490,00000	410,00000	741,00006	490	410	741	0,00E+00	0,00E+00	-6,10E-05

NOTA: Dados em mm

Tabela C.0.5 Dados peça exemplo 2- robô ABB IRB-140

#	Trajetória percorrida			Trajetória alvo			Erro		
	X	Y	Z	X	Y	Z	X	Y	Z
1	449,99998	9,99994	575,00006	450	10	575	2,45E-05	6,18E-05	-6,24E-05
2	469,99992	29,99993	555,00007	470	30	555	7,99E-05	7,31E-05	-7,34E-05
3	470,00000	30,00000	550,00008	470	30	550	4,98E-07	3,28E-08	-7,64E-05
4	470,00000	30,00000	540,00008	470	30	540	9,94E-07	6,34E-08	-7,65E-05
5	470,00000	30,00000	549,99992	470	30	550	9,93E-07	6,34E-08	7,61E-05
6	470,00000	30,00000	554,99992	470	30	555	4,97E-07	3,17E-08	7,62E-05
7	470,00000	30,00000	554,99992	470	30	555	4,97E-07	3,17E-08	7,62E-05
8	450,00007	10,00008	574,99992	450	10	575	-7,26E-05	-7,94E-05	7,92E-05
9	450,00007	10,00008	574,99992	450	10	575	-7,26E-05	-7,94E-05	7,92E-05
1010	539,20000	104,80006	540,00000	539,2	104,8	540	9,26E-07	-6,12E-05	5,04E-08
1011	539,20000	99,20009	540,00000	539,2	99,2	540	4,52E-07	-8,55E-05	2,03E-08
1012	549,99992	109,99992	540,00000	550	110	540	8,38E-05	8,12E-05	1,49E-06
1013	550,00000	110,00000	544,99992	550	110	545	4,42E-07	8,94E-08	7,62E-05
1014	488,00005	48,00006	554,99998	488	48	555	-5,38E-05	-6,50E-05	1,60E-05
1015	450,00007	10,00008	574,99996	450	10	575	-6,83E-05	-7,80E-05	4,31E-05
1016	449,99991	9,99994	575,00003	450	10	575	9,37E-05	6,48E-05	-3,16E-05
1017	487,99996	5,00000	388,00009	488	5	388	3,99E-05	-4,98E-06	-8,56E-05
1018	488,00000	9,99992	388,00000	488	10	388	3,93E-07	7,63E-05	4,02E-09
1019	503,99994	10,00000	388,00000	504	10	388	6,16E-05	-2,75E-07	3,43E-07
2250	640,00000	13,99994	450,00000	640	14	450	4,81E-08	6,10E-05	8,26E-10
2251	640,00000	18,99992	450,00000	640	19	450	3,00E-07	7,63E-05	4,78E-09
2252	640,00000	14,00008	450,00000	640	14	450	3,00E-07	-7,63E-05	4,80E-09
2253	640,00000	14,99994	450,00000	640	15	450	4,81E-08	6,10E-05	8,05E-10
2254	640,00000	19,99992	450,00000	640	20	450	3,00E-07	7,63E-05	4,65E-09
2255	640,00000	15,00008	450,00000	640	15	450	3,00E-07	-7,63E-05	4,68E-09
2256	640,00000	5,00008	450,00000	640	5	450	6,02E-07	-7,62E-05	1,13E-08
2257	450,00007	10,00000	574,99991	450	10	575	-7,21E-05	-1,93E-06	8,83E-05
2258	450,00002	9,99999	575,00007	450	10	575	-1,93E-05	6,39E-06	-6,56E-05
2259	444,99998	109,99995	450,00006	445	110	450	1,57E-05	5,16E-05	-5,88E-05
3090	450,99994	110,00000	540,00000	451	110	540	6,10E-05	4,01E-08	8,20E-08
3091	455,99992	110,00000	540,00000	456	110	540	7,63E-05	2,49E-07	5,08E-07
3092	451,00008	110,00000	540,00000	451	110	540	-7,63E-05	2,49E-07	5,08E-07
3093	451,99994	110,00000	540,00000	452	110	540	6,10E-05	4,00E-08	8,18E-08
3094	456,99992	110,00000	540,00000	457	110	540	7,63E-05	2,48E-07	5,07E-07
3095	452,00008	110,00000	540,00000	452	110	540	-7,63E-05	2,48E-07	5,07E-07
3096	452,99994	110,00000	540,00000	453	110	540	6,10E-05	3,99E-08	8,16E-08
3097	457,99992	110,00000	540,00000	458	110	540	7,63E-05	2,47E-07	5,06E-07
3098	453,00008	110,00000	540,00000	453	110	540	-7,63E-05	2,47E-07	5,05E-07
3099	453,99994	110,00000	540,00000	454	110	540	6,10E-05	3,98E-08	8,14E-08

NOTA: Dados em mm

Tabela C.0.6 Dados peça exemplo 2- robô Tricept 806

#	Trajetória percorrida			Trajetória alvo			Erro		
	X	Y	Z	X	Y	Z	X	Y	Z
1	200,00004	309,99994	-1675,00000	200	310	-1675	-3,70E-05	5,88E-05	2,37E-06
2	220,00008	330,00008	-1695,00000	220	330	-1695	-7,82E-05	-8,44E-05	-2,13E-06
3	219,99994	329,99990	-1700,00002	220	330	-1700	6,20E-05	9,80E-05	1,64E-05
4	219,99995	329,99991	-1710,00001	220	330	-1710	5,40E-05	8,63E-05	7,51E-06
5	220,00007	330,00008	-1700,00002	220	330	-1700	-6,61E-05	-8,07E-05	1,64E-05
6	220,00005	330,00010	-1695,00000	220	330	-1695	-5,31E-05	-9,68E-05	-2,13E-06
7	220,00005	330,00010	-1695,00000	220	330	-1695	-5,31E-05	-9,68E-05	-2,13E-06
8	199,99992	309,99993	-1675,00000	200	310	-1675	8,28E-05	6,95E-05	2,37E-06
9	199,99992	309,99993	-1675,00000	200	310	-1675	8,28E-05	6,95E-05	2,37E-06
1010	289,20000	404,79993	-1709,99999	289,2	404,8	-1710	-3,12E-06	6,96E-05	-1,32E-05
1011	289,20000	399,19994	-1710,00003	289,2	399,2	-1710	-3,62E-06	6,32E-05	2,99E-05
1012	300,00007	410,00005	-1709,99997	300	410	-1710	-6,58E-05	-4,86E-05	-2,93E-05
1013	300,00009	410,00009	-1705,00004	300	410	-1705	-9,17E-05	-9,42E-05	4,48E-05
1014	237,99992	347,99993	-1694,99995	238	348	-1695	8,32E-05	6,61E-05	-4,83E-05
1015	199,99991	309,99993	-1675,00000	200	310	-1675	8,61E-05	7,36E-05	2,37E-06
1050	349,20000	310,00000	-1840,79989	349,2	310	-1840,8	-6,53E-07	1,86E-06	-1,14E-04
1051	333,19997	310,00000	-1840,79992	333,2	310	-1840,8	2,91E-05	-2,16E-06	-7,80E-05
1052	317,19997	310,00000	-1840,79998	317,2	310	-1840,8	2,61E-05	-1,83E-06	-2,45E-05
1053	301,19998	310,00000	-1840,79990	301,2	310	-1840,8	1,93E-05	1,60E-06	-9,70E-05
2250	389,99997	314,00000	-1799,99994	390	314	-1800	3,30E-05	1,21E-06	-5,80E-05
2251	389,99999	319,00000	-1799,99989	390	319	-1800	1,14E-05	-2,53E-06	-1,09E-04
2252	390,00000	314,00000	-1800,00009	390	314	-1800	7,86E-07	1,21E-06	9,05E-05
2253	389,99998	315,00000	-1799,99994	390	315	-1800	1,85E-05	-1,58E-06	-5,57E-05
2254	390,00000	320,00000	-1799,99993	390	320	-1800	4,98E-06	-8,04E-07	-7,12E-05
2255	390,00001	315,00000	-1800,00009	390	315	-1800	-1,37E-05	-1,58E-06	9,29E-05
2300	200,00000	400,66949	-1838,86470	200	400,67	-1838,87	1,77E-06	-8,38E-05	-2,53E-05
2301	200,00000	394,70460	-1836,92662	200	394,70	-1836,93	9,53E-07	-1,06E-04	4,94E-06
2302	200,00000	389,11628	-1834,07929	200	389,12	-1834,08	5,78E-07	-6,65E-05	4,80E-05
2303	200,00000	384,04224	-1830,39274	200	384,04	-1830,39	-2,05E-06	-8,07E-05	1,74E-05
3091	206,00000	409,99998	-1709,99992	206	410	-1710	8,46E-07	1,54E-05	-8,45E-05
3092	201,00000	410,00001	-1710,00014	201	410	-1710	-1,94E-06	-9,63E-06	1,38E-04
3093	202,00000	409,99998	-1709,99999	202	410	-1710	2,74E-07	1,88E-05	-1,26E-05
3094	207,00000	409,99999	-1709,99994	207	410	-1710	2,25E-06	8,45E-06	-5,79E-05
3095	202,00000	409,99998	-1710,00015	202	410	-1710	2,74E-07	1,88E-05	1,51E-04
3096	203,00000	410,00000	-1709,99990	203	410	-1710	2,59E-06	3,09E-06	-1,03E-04
3097	208,00000	410,00000	-1709,99987	208	410	-1710	2,79E-07	-4,31E-06	-1,26E-04
3098	203,00000	410,00000	-1710,00006	203	410	-1710	2,59E-06	3,06E-06	6,08E-05
3099	204,00000	410,00000	-1709,99996	204	410	-1710	-1,62E-06	-2,47E-06	-4,08E-05
3100	209,00000	410,00001	-1709,99996	209	410	-1710	2,98E-06	-6,83E-06	-4,13E-05

NOTA: Dados em mm