

**NAZARENO DE OLIVEIRA PACHECO**

**Projeto e Implementação de um Controlador Numérico  
Computadorizado Aberto Baseado em STEP-NC**

**Joinville  
2009**

UNIVERSIDADE DO ESTADO DE SANTA CATARINA  
MESTRADO PROFISSIONAL EM ENGENHARIA ELÉTRICA

NAZARENO DE OLIVEIRA PACHECO

PROJETO E IMPLEMENTAÇÃO DE UM CONTROLADOR  
NUMÉRICO COMPUTADORIZADO ABERTO BASEADO EM  
STEP-NC

Dissertação submetida à Universidade do Estado de Santa Catarina como parte dos requisitos para a obtenção do grau de Mestre em Engenharia Elétrica

**Orientador:** Roberto Silvio Ubertino Rosso Jr., PhD

**Co-Orientador:** André Bittencourt Leal, Dr. Eng

Joinville  
2009

NAZARENO DE OLIVEIRA PACHECO

**PROJETO E IMPLEMENTAÇÃO DE UM CONTROLADOR  
NUMÉRICO COMPUTADORIZADO ABERTO BASEADO EM  
STEP-NC**

Esta Dissertação foi julgada adequada para a obtenção do título de Mestre em Engenharia Elétrica e aprovado em sua forma final pelo Curso de Mestrado Profissional em Engenharia Elétrica do CCT/UDESC.

**Banca Examinadora:**

Orientador:

---

Roberto Silvio Ubertino Rosso Jr., PhD

Co-Orientador:

---

André Bittencourt Leal, Dr. Eng

Membro:

---

João Carlos Espíndola Ferreira, PhD

Membro:

---

Marcelo da Silva Hounsell, PhD

**Joinville, 19 de maio de 2009**

*A Deus.*

*Aos meus avós*

*Aos meus pais*

*Aos meus familiares e amigos.*

## AGRADECIMENTOS

Aos professores dos Departamentos de Engenharia Elétrica e Ciência da Computação pelos seus ensinamentos e aos colegas e amigos de curso.

Aos meus pais, Seloé Pacheco e Nazarita de Oliveira Pacheco, minha filha Beatriz pelo tempo que tivemos privados do convívio, meus irmãos pelo grande incentivo na realização deste trabalho.

À minha noiva pelo incentivo e compreensão nos momentos de dificuldade nessa caminhada.

Ao meu orientador, Roberto S. U. Rosso Jr. que praticamente foi um pai em Joinville e ao meu co-orientador André B. Leal pelas conversas esclarecedoras e o incentivo.

Ao professor e amigo Fernando Lafratta.

A Empresa Júnior pela parceria formada.

Aos bolsistas Eduardo Harbs e Fernando Rebelo.

Aos irmãos de coração em Joinville Silvio Teston, Pedro Roman, Fernando Garcia, Daniel Joppi.

*Se nós soubéssemos o que estavam fazendo, isto não seria chamado de pesquisa, seria?*

*- Albert Einstein*

*O único modo de evitar os erros é adquirindo experiência, contudo, a única maneira de adquirir experiência é cometendo erros.*

*(Autor Desconhecido)*

## RESUMO

No trabalho desenvolvido foi utilizada a Norma ISO14649 (STEP-NC) que é o novo padrão para interface de dados de manufatura entre os sistemas CAM/CNC. Atualmente a troca dessas informações é feita usando a linguagem da Norma ISO6983, conhecida como Código-G/M. Isto significa que partes das informações geradas no ciclo de desenvolvimento, notadamente a geometria do produto e dados de ferramentas, são perdidas ao se gerar o programa em código G/M. Para a prova de conceito da funcionalidade desse novo padrão no controle e acionamento direto de máquinas-ferramenta CNC foram projetados e construídos *software*, *hardware* e uma estrutura mecânica simples de uma fresadora/-furadeira CNC. A arquitetura de *software* e *hardware* é aberta e aderente à STEP-NC. O sistema desenvolvido faz uso de arquivos ISO14649 para programação da máquina sem uso de linguagens legadas. Isto permitiu a validação do uso da norma no acionamento de máquinas CNC, provando a flexibilidade e eficácia para a troca de informações e controle de máquinas-ferramenta.

**PALAVRAS-CHAVE:** Automação da manufatura, CNC, CAD/CAM, STEP, ISO10303, STEP-NC, ISO14649, CNC aberto.

## ABSTRACT

This work was developed using the standard ISO14649 (STEP-NC), which is the new standard for manufacturing data interface to use in CAM and CNC systems. Currently the information exchange is done by the use of the ISO6983 standard known as G/M code. This means that parts of the information generated in the cycle of development, such as product geometry and tool data, are lost when the G/M code programme is generated. A simple mechanical structure of a milling/drilling machine, software and hardware were designed and built to proof the concept that ISO 14649 is functional in the control and direct drive of CNC machines-tool. The hardware and software architecture developed is open and STEP-NC compliant. The system developed uses ISO14649 files to program the machine without any legacy language. This provided the means to validate the use of the new standard to drive CNC machines, proofing its flexibility and efficacy in information exchange and control of machines-tool.

**KEYWORDS:** Manufacturing Automation, CNC, CAD/CAM, STEP, ISO10303, STEP-NC, ISO14649, Open CNC.

## LISTA DE FIGURAS

2.1	Ciclo de projeto tradicional.(BACK 1983) . . . . .	4
2.2	Ciclo de projeto tradicional com a introdução do computador.(BACK 1983)	4
2.3	<i>Features</i> encontradas em uma peça prismática. (ROSSO Jr. 2005) . . . . .	7
2.4	Determinação da trajetória.(SILVA 2001) . . . . .	10
2.5	Simulação da usinagem.(SILVA 2001) . . . . .	10
3.1	Arquitetura típica do CNC e periféricos. Adaptado de (CAPELLI 2006) . .	15
3.2	Arquitetura nova do CNC com CLP. Adaptado de (CAPELLI 2006) . . . .	15
3.3	Interface Homem-Máquina.(ROMI 2008) . . . . .	16
3.4	Servomotor e Servoconversor.(WEG 2008) . . . . .	17
3.5	Princípio de funcionamento do encoder. (CAPELLI 2006) . . . . .	19
3.6	Canais do encoder.(CAPELLI 2006) . . . . .	20
3.7	Geração do programa. (SCHÜTZER 2008) . . . . .	22
3.8	Linha de programa em ISO6983. (WILLIAMS 1988) . . . . .	23
3.9	Exemplo de programa CNC. (CAPELLI 2002) . . . . .	24
4.1	Tipos de Sistemas. (STEPTOOLS 2008) . . . . .	26
4.2	Comunicação entre sistemas utilizando tradução direta. (ZEID 1991) apud (ROSSO Jr. 2005) . . . . .	27
4.3	Comunicação entre sistemas utilizando tradução indireta.(ZEID 1991) apud (ROSSO Jr. 2005) . . . . .	28
4.4	Estrutura de alto nível da ISO10303.(LOFREDO 2008) . . . . .	30
4.5	Sistema Atual de integração CAD/CAPP/CAM/CNC.(NASSEHI 2006) . .	32

4.6	ISO6983 (G & M Codes) vs ISO14649 (STEP-NC). (XU 2006), (ALLEN 2003)	33
4.7	Relação entre ISO10303 e ISO14649. (ROSSO Jr. 2005)	34
4.8	Detalhes de uma estrutura típica de um programa STEP-NC.(ROSSO Jr. 2003)	36
4.9	Estrutura de Dados STEP-NC. (ISO14649-1 2003)	36
4.10	Exemplo resumido de programa STEP-NC.	37
4.11	Tipos de controladores. (SUH 2003)	39
5.1	Conceito de CNC aberto. Modificado de (OSACA 2008).	42
5.2	Arquitetura OSACA. (OSACA 2008)	44
5.3	Arquitetura aberta OSEC. (SAWADA 1997)	45
5.4	Arquitetura OMAC. (KOLLA 2002)	45
5.5	Arquitetura aberta MDSI. (MDSI 2008)	45
5.6	Arquitetura EMC. (NIST 2008)	46
6.1	Furadeira/Fresadora CNC aderente a STEP-NC.	47
6.2	Arquitetura Aberta desenvolvida neste trabalho	49
6.3	Placa de Controle.	50
6.4	Esquema elétrico da placa de controle.	50
6.5	<i>Drivers</i> dos eixos.	51
6.6	Esquema elétrico dos <i>drivers</i> dos motores	51
6.7	Driver para acionamento da ferramenta	52
6.8	Esquema elétrico do driver da ferramenta	52
6.9	Motor de passo utilizado no presente trabalho	54
6.10	Funcionamento básico de um compilador.(AHO 1995)	55
6.11	Estrutura do compilador desenvolvido	57
6.12	Compilador indicando que o arquivo está correto	58
6.13	Aviso de que a tabela com os dados de usinagem foi gerada	58

6.14	Compilador indicando que o arquivo está incorreto . . . . .	59
6.15	Indicação da <i>Home position</i> . . . . .	59
6.16	Indicação da Conclusão do trabalho . . . . .	60
6.17	Tela de configuração . . . . .	61
6.18	Usinagem sem otimizador X Usinagem com otimizador . . . . .	61
6.19	Esboço da peça de 5 furos . . . . .	62
6.20	Usinagem de 5 furos . . . . .	63
6.21	Esboço da peça de 13 furos . . . . .	63
6.22	Usinagem de 13 furos . . . . .	64
6.23	Usinagem sem otimizador X Usinagem com otimizador . . . . .	65

## LISTA DE TABELAS

2.1	Tabela comparativa entre estratégias de usinagem (SOUZA 2004) . . . . .	10
3.1	Comandos preparatórios (Código G) (ISO6983-1 1982) . . . . .	23
3.2	Comandos mistos (Código M) (ISO6983-1 1982) . . . . .	24
4.1	Equações para cálculo do número de tradutores (KERN 1997). . . . .	26
4.2	Quantidade de tradutores. . . . .	27
4.3	Estrutura dos protocolos da STEP (ROSSO Jr. 2005), (LOFREDO 2008) .	29
6.1	Meio passo ( <i>Half-step</i> ) . . . . .	54
6.2	Usinagem sem otimizador X Usinagem com otimizador . . . . .	65

## LISTA DE ABREVIATURAS

<b>ASIC</b>	<i>Circuito Integrado de Aplicação Específica</i>
<b>CAD</b>	<i>Computer Aided Design</i>
<b>CAE</b>	<i>Computer Aided Engineering</i>
<b>CAM</b>	<i>Computer Aided Manufacturing</i>
<b>CAPP</b>	<i>Computer Aided Process Planning</i>
<b>CLP</b>	<i>Controlador Lógico Programável</i>
<b>CNC</b>	<i>Computer Numerical Control</i>
<b>CPU</b>	<i>Unidade Central de Processamento</i>
<b>DSP</b>	<i>Digital Signal Processor</i>
<b>EDM</b>	<i>Electrical Discharge Machining</i>
<b>ISO</b>	<i>International Standards Organization</i>
<b>IHM</b>	<i>Interface Homem-Máquina</i>
<b>MES</b>	<i>Manufacturing Execution System</i>
<b>MDSI</b>	<i>Manufacturing Data Systems Inc</i>
<b>MIT</b>	<i>Massachusetts Institute of Technology</i>
<b>NC</b>	<i>Numerical Computer</i>
<b>NIST</b>	<i>National Institute of Standards and Technology</i>
<b>OAC</b>	<i>Open Architecture Controller</i>
<b>OMAC</b>	<i>Open Modular Architecture Controls</i>
<b>OSACA</b>	<i>Open System Architecture for Controls within Automation Systems</i>
<b>OSEC</b>	<i>Open System Environment for Controller</i>
<b>PC</b>	<i>Personal computer</i>
<b>RV</b>	<i>Realidade Virtual</i>
<b>SDAI</b>	<i>Standard Data Access Interface</i>
<b>SFP</b>	<i>Shop-floor programming</i>
<b>STEP</b>	<i>Standards for Exchange Product Data Model</i>
<b>STEP-NC</b>	<i>Standards for Exchange Product Data Model – Numerical Control</i>
<b>XML</b>	<i>Extensible Markup Language</i>

## SUMÁRIO

<b>1 INTRODUÇÃO</b> . . . . .	1
<b>2 PROJETO E MANUFATURA AUXILIADOS POR COMPUTADOR</b> .	3
2.1 PROJETO AUXILIADO POR COMPUTADOR . . . . .	5
2.1.1 <i>Wireframe</i> . . . . .	5
2.1.2 Modelos de superfície . . . . .	5
2.1.3 Modelos sólidos . . . . .	6
2.2 VISÃO GERAL SOBRE TECNOLOGIA DE <i>FEATURE</i> . . . . .	6
2.3 MANUFATURA AUXILIADA POR COMPUTADOR . . . . .	6
<b>3 MÁQUINAS-FERRAMENTA A CONTROLE NUMÉRICO</b> . . . . .	12
3.1 HISTÓRICO . . . . .	13
3.2 ARQUITETURA DE UMA MÁQUINA CNC . . . . .	13
3.2.1 Interface Homem-Máquina - (IHM) . . . . .	16
3.2.2 <i>Drives</i> de controle . . . . .	16
3.2.3 Entradas/Saídas (E/S) . . . . .	17
3.2.3.1 Sensores . . . . .	18
3.2.3.2 Atuadores . . . . .	20
3.3 FUNCIONAMENTO E PROGRAMAÇÃO DA MÁQUINA-FERRAMENTA .	21
<b>4 NORMA ISO10303 OU STEP E NORMA ISO14649 OU STEP-NC</b> . .	25
4.1 NORMA ISO10303 OU STEP . . . . .	25
4.1.1 Estrutura do STEP . . . . .	28
4.2 NORMA ISO14649 OU STEP-NC . . . . .	30
4.2.1 Estrutura da Norma ISO14649 . . . . .	34

4.2.2	Princípios Fundamentais da ISO14649 . . . . .	35
<b>5</b>	<b>CONTROLE NUMÉRICO DE ARQUITETURA ABERTA . . . . .</b>	<b>41</b>
5.1	PRINCIPAIS ARQUITETURAS ABERTAS . . . . .	43
<b>6</b>	<b>DESENVOLVIMENTO DE UMA MÁQUINA DE ARQUITETURA ABERTA ADERENTE A STEP-NC . . . . .</b>	<b>47</b>
6.1	ARQUITETURA ABERTA BASEADA EM STEP-NC . . . . .	48
6.2	<i>HARDWARE</i> DESENVOLVIDO . . . . .	48
6.2.1	Controladora . . . . .	49
6.2.2	<i>Drivers</i> dos acionamentos dos eixos coordenados . . . . .	51
6.2.3	<i>Driver</i> do acionamento do eixo árvore . . . . .	52
6.2.4	Motor de passo . . . . .	53
6.3	<i>SOFTWARE</i> DESENVOLVIDO . . . . .	54
6.3.1	Compilador . . . . .	54
6.3.2	Interfaces . . . . .	57
6.3.3	Otimizador de trajetória . . . . .	60
6.4	TESTES DE VALIDAÇÃO . . . . .	61
<b>7</b>	<b>CONCLUSÃO . . . . .</b>	<b>67</b>
7.1	TRABALHOS FUTUROS . . . . .	68
	<b>REFERÊNCIAS . . . . .</b>	<b>69</b>

# 1 INTRODUÇÃO

A concorrência global tem levado as empresas em busca de tornar seus métodos de produção mais flexíveis e ágeis a fim de satisfazer as exigências do mercado e vencer o desafio dos concorrentes.

Nas indústrias do setor metal-mecânico, o uso dos computadores tem melhorado a produtividade significativamente e encurtando o ciclo de vida do produto, planejamento e fabricação. Neste cenário, dados, informações e conhecimentos são essenciais para o sucesso. Atualmente a necessidade de dispor de sistemas integrados de manufatura é o diferencial da colocação de um produto com mais rapidez no mercado.

No centro da indústria de manufatura está o Controle Numérico (NC), que é a tecnologia utilizada há mais de cinquenta anos. Durante este período registrou-se uma evolução notável nas máquinas-ferramenta a controle numérico, tanto nos projetos de máquinas quanto nas capacidades de usinagem das mesmas. Entretanto a linguagem padrão usada na fabricação ainda é baseada no movimento e estado da máquina através do chamado código G/M (ISO6983). Do ponto de vista da integração da manufatura, essa é uma desvantagem importante, pois sendo a programação NC criada nesse ambiente tradicional, perde-se grande parte da informação como a descrição da geometria e ferramentas. Além disso, o padrão de linguagem ISO6983 é aberto a interpretações permitindo que cada fabricante de máquina-ferramenta a controle numérico crie seu próprio dialeto. O resultado disso tudo é uma perda de interoperabilidade e a incapacidade de ter bi-direcionalidade no fluxo de informação.

O objetivo geral deste trabalho consiste em demonstrar que o uso da norma ISO14649 é viável para o controle direto de máquinas-ferramenta, ou seja, sem o uso de linguagens intermediárias como a ISO6983. No desenvolvimento desse trabalho teve-se como objetivos específicos:

- Propor uma arquitetura aberta para CNC que seja simples e de baixo custo;
- Implementar uma solução de *hardware* para viabilizar a arquitetura proposta;
- Implementar uma solução de *software* que utilize o *hardware* desenvolvido e a

ISO14649 (STEP-NC);

- Demonstrar que a ISO14649 é viável para o controle de máquinas-ferramenta sem o uso de linguagens legadas como a ISO6983 (código-G) e
- Criar infra-estrutura para desenvolvimento de pesquisas em interfaces abertas no âmbito do PPGEEL.

O texto é organizado como segue. No Capítulo 2, é explanado sobre Projeto e Manufatura Auxiliado por Computador. No Capítulo 3, apresenta-se noções sobre Máquinas-ferramenta a Controle Numérico. No Capítulo 4, é explicada as normas ISO10303 e ISO14649 que tem por objetivo a integração total da manufatura. No Capítulo 5 é sobre Controle Numérico de Arquitetura Aberta, onde buscou-se ressaltar a importância do uso da tecnologia não proprietária. No Capítulo 6 é o Desenvolvimento de uma máquina de arquitetura aberta aderente a STEP-NC explica-se os procedimentos realizados no decorrer do desenvolvimento dessa aplicação da norma. Por fim, no Capítulo 7, resumem-se as conclusões e futuros trabalhos.

## 2 Projeto e Manufatura Auxiliados por Computador

O processo de se projetar algum produto é caracterizado como um processo iterativo, pois os resultados de uma fase subsequente realimenta as fases anteriores com o objetivo de otimizar o produto a ser desenvolvido. Esse processo é convencionalmente dividido em seis fases, ou passos (BACK 1983):

- **Fase 1 - Reconhecimento da necessidade:** Significa alguém perceber que um problema (uma necessidade) existe e precisa-se tomar alguma ação corretiva. Tal problema poderia ser um defeito numa máquina, ou um projeto de um novo produto;
- **Fase 2 - Definição do Problema:** Envolve uma especificação completa do produto a ser projetado, que inclui características funcionais e físicas, custo, qualidade e desempenho operacional;
- **Fase 3 - Síntese:** Essa etapa envolve basicamente a modelagem geométrica do produto;
- **Fase 4 - Análise e otimização:** Nessa fase existe a concepção de um componente, que é sujeito a análise, e que pode ser melhorado através desse procedimento de análise e reprojetado;
- **Fase 5 - Avaliação:** Significa quanto o projeto final refletiu as especificações iniciais estabelecidas na fase 2. Normalmente requer a fabricação e teste no protótipo para avaliar o desempenho operacional, qualidade e outros;
- **Fase 6 - Apresentação:** É a fase final do processo do projeto. Nela contém os documentos referentes ao projeto, tais como: desenhos do projeto, especificações de materiais, e assim por diante. Há uma criação da base de dados do projeto.

A Figura 2.1 apresenta os passos de um processo de um ciclo de projeto tradicional descrito por (BACK 1983)

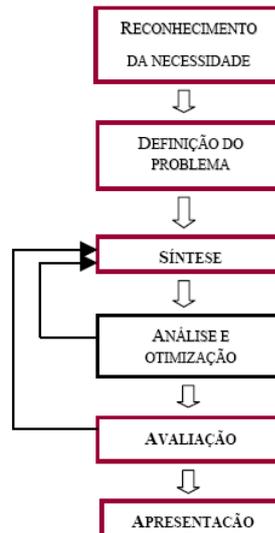


Figura 2.1: Ciclo de projeto tradicional.(BACK 1983)

Esse ciclo de projeto é um processo demorado, mas com a introdução do computador esse ciclo obteve mais dinamismo, otimizando assim o trabalho do projetista e possibilitando que ele utilize maior parte do seu tempo em trabalho criativo, testando com mais rapidez, por exemplo, várias opções construtivas. A Figura 2.2 apresenta o ciclo de projeto tradicional com a introdução do computador.

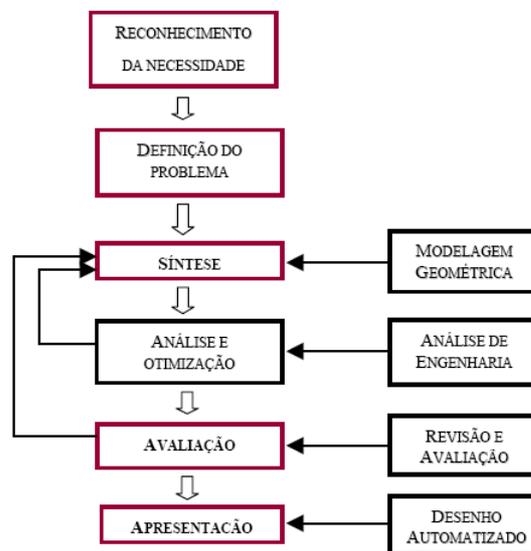


Figura 2.2: Ciclo de projeto tradicional com a introdução do computador.(BACK 1983)

## 2.1 Projeto Auxiliado por Computador

Embora o CAD (*Computer Aided Design*) não seja o foco deste trabalho, ele está intimamente relacionado com sistemas CAM, projetos baseados em *features* e manufatura. O início do CAD pode ser atribuídos ao trabalho desenvolvido pelo pioneiro Ivan Sutherland, que mostra o protótipo *SKETCHPAD* em 1963 (BEDWORTH 1991). Desde então, as atividades passaram de projetos em papel para projetos baseado em sistemas CAD bidimensionais e evoluiu para os sistemas tridimensionais onde se torna mais realista a representação e análise dos projetos. Modelos CAD têm sido usados como entrada para o processo de fabricação, quer para sistemas CAM (*Computer Aided Manufacturing*) ou sistemas de CAPP (*Computer Aided Process Planning*). No processo de evolução do CAD 3D, este foi desenvolvido utilizando *wireframe*, modelos de superfície e modelos sólidos. A seguir, estes tipos de modelos são discutidos de forma sucinta.

### 2.1.1 *Wireframe*

Modelos *wireframe* são baseados em linhas (retas ou curvas) que apontam para a construção de modelos 3D. Eles são a extensão lógica dos sistemas CAD bidimensionais desenvolvidos anteriormente. No entanto, a limitação destes modelos são variadas e sem nenhuma informação sobre superfície representada, o que provoca sérios problemas de ambiguidade na representação dos objetos reais (ZEID 1991), (BEDWORTH 1991).

### 2.1.2 Modelos de superfície

Esta abordagem é uma melhoria a partir dos modelos *wireframe* pois eles usam linhas e pontos como a base para criar as superfícies. Eles fornecem algum suporte em sistemas CAM que pode ser utilizado para traçar a trajetória da ferramenta. É, no entanto, limitado e traz algumas representações ambíguas, como a definição da superfície que limita o volume do objeto (ZEID 1991), (SILVA 2001). Hoje alguns produtos comerciais ainda usam modelos de superfície, mas com uma abordagem híbrida que use sólidos para descrever superfícies (Bentley System Inc. 2008), (Delcam Plc. 2008).

### 2.1.3 Modelos sólidos

Modelos sólidos têm o atributo de completar e fornecer a não ambiguidade como resultado dos dados, contidas em sua base de dados. Isto torna-os um meio mais adequado para representar geometrias para a concepção e fabricação dos produtos. Há uma série de técnicas que podem ser usadas para modelar sólidos tais como *Pure Primitive Instancing* (PPI), *Spatial Occupancy Enumeration* (SOE), *Cell Decomposition* (CD), *Sweeping* (S), *Constructive Solid Geometry* (CSG) e *Boundary representation* (BRep) (BEDWORTH 1991). Uma outra técnica, denominada *Destructive Solid Geometry* (DSG), mas pode ser considerado como um sub-conjunto de CSG onde só o operador booleano diferença é usado (HOUNSELL 1998). Os *Geometric Solid Modellers* (GSM) trouxeram uma melhor forma de representar as geometrias de um produto, entretanto eles estavam ainda longe de resolver o problema de concepção e fabricação.

Uma solução para estes problemas do ponto de vista de representar o produto foi o desenvolvimento da tecnologia de *features* que é discutida na próxima seção.

## 2.2 Visão Geral sobre Tecnologia de *Feature*

A Tecnologia de *Features* tem sido um tema de pesquisa desde a década de 1970 e, desde então uma série de diferentes utilizações e abordagens podem ser encontradas na literatura.

Segundo Rosso (2005), o conceito de *Features* pode ser resumido como: características que dizem respeito às formas básicas e atributos tecnológicos, que são associados a operações de manufatura.

Segundo Case (1992), o conceito de *Features* é que a descrição de um produto não só diz o que é o produto, mas também contém informações implícitas e explícitas sobre a forma como ele pode ser transformado a partir de ou para algum outro estado. A Figura 2.3 ilustra algumas das *features* típicas encontradas em peças prismáticas (CASE, K. 1992).

## 2.3 Manufatura Auxiliada por Computador

Em 1949 o *Massachusetts Institute of Technology* (MIT) foi convidado por *Parsons*

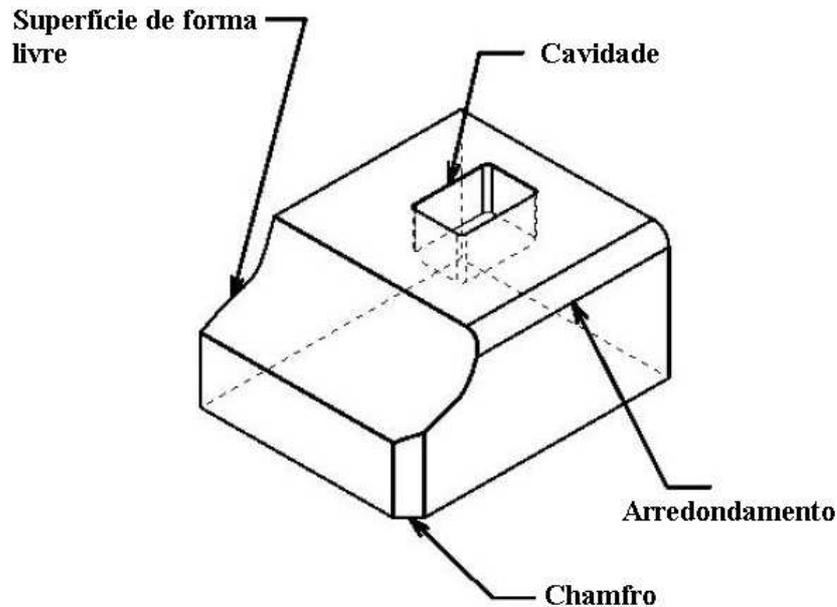


Figura 2.3: *Features* encontradas em uma peça prismática. (ROSSO Jr. 2005)

*Corporation* para resolver um problema de fabricação para a Força Aérea dos EUA. Foi para a usinagem de formas complexas para aeronaves, como helicópteros e os seus mísseis. Em 1953 a primeira máquina-ferramenta com controle numérico (NC) foi apresentada (uma Fresadora CNC Cincinnati Hydrotel adaptada), e ela foi demonstrada no MIT (ZEID 1991). O princípio básico dessa classe de máquina é o comando eletrônico do movimento da ferramenta (ZEID 1991). Houveram muitas melhorias nos aspectos de *hardware* e *software* das máquinas NC desde então. A primeira geração utilizava válvulas; a segunda as válvulas melhoraram, e mais tarde foi utilizada a tecnologia de estado sólido. A terceira geração utilizava circuitos integrados e *Read Only Memory* (ROM), que fez os controladores capazes de reter programas na memória, em vez da leitura direta das fitas perfuradas utilizadas nas 1ª e 2ª gerações. Isto levou ao aparecimento do que é conhecido como Controle Numérico Computadorizado (CNC) (ZEID 1991). Outras melhorias importantes surgiram mais tarde como a visualização e capacidade de edição em monitores.

Hoje, o *software* e *hardware* disponíveis em máquinas-ferramenta são capazes de simular graficamente o movimento da ferramenta, controlar a vida da ferramenta e de realizar Controle Adaptativo (AC) para melhorar *on-line* as condições de usinagem. Atualmente as tendências são para arquiteturas abertas, como OSACA (LUTZ 1997) e OMAC (OMAC 2008) sobre sistemas *software/hardware* que possam facilitar a utilização de tec-

nologias não proprietárias. Esse assunto sobre Controle Numérico de Arquitetura Aberta será explicado com mais detalhes na seção 5.

Nos últimos anos o CAM mudou significativamente, transformações que foram dirigidas pela evolução dos computadores, ferramentas de usinagem e sistemas CAD. Inicialmente o CAD era uma composição de linhas e superfícies, e o CAM era desenvolvido para usinagem 2D ou 3D, pois poucos apresentavam um bom desempenho em ambos. O sistema CAM apontava principalmente na habilidade de prover caminhos de ferramentas, sem haver colisões, erros de usinagem e vibrações, proporcionando um bom acabamento para modelos compostos por superfícies muitas vezes incompletas. Caso houvesse uma alteração no projeto, a programação do processo de fabricação NC era refeita para refletir a mudança (SIVITTER 2004).

O CAM é percebido como uma ferramenta de auxílio via computador para a preparação da fabricação, reproduzindo as tecnologias usadas no chão de fábrica. Muitas vezes o termo CAM é sinônimo da programação NC, concepção que ficou conhecida com a sigla CAD/CAM, que significa módulos de programação NC em sistemas CAM (RIBEIRO e ROZENFELD 2008).

Os sistemas CAD/CAM são atualmente usados no preparo de programas CNC. A partir da interpretação da geometria das peças armazenadas em arquivo gráfico gerado no CAD, são criados os programas NC segundo algumas informações fornecidas pelo programador usuário do sistema CAM.

Segundo (SIVITTER 2004), os sistemas CAD/CAM têm progredido com a garantia da integridade dos dados a partir do armazenamento de modelos sólidos gerados pelo sistema CAD. Os sistemas de projeto e manufatura auxiliados pelo computador mais avançados desfrutam de um vínculo interativo entre o modelo virtual do objeto e o caminho da ferramenta, ao alterar a geometria da peça o caminho da ferramenta é atualizado automaticamente. Esta associação que vincula os dois sistemas traz facilidades de comunicação entre os setores de projeto e produção, resultando em otimização no tempo de produção (MARCIANO 2008).

Segundo D'ISSY (2004), atualmente a oferta de *software* CAM se expande devido à modernização da indústria brasileira para competir no mercado globalizado.

A manufatura auxiliada por computador utiliza recursos computacionais no planeja-

mento, gerenciamento e controle da manufatura. Para o planejamento e gerenciamento da manufatura o computador é usado *off-line*, fornecendo informações para o gerenciamento e planejamento efetivo das atividades de produção. Entre as principais tarefas realizadas pelo CAM nessa área, tem-se (SIVITTER 2004), (D'ISSY 2004), (SILVA 2001):

- Estimativa do custo total (material e produção) de um novo produto a ser fabricado;
- No caso de processos que envolvam usinagem, o CAM calcula os parâmetros ótimos de usinagem, como rotação da ferramenta ou eixo de um torno, velocidade de avanço da ferramenta, número de passes, entre outros. Os cálculos são baseados em dados obtidos na fábrica ou em laboratório e essencialmente relacionam a vida da ferramenta com as condições de usinagem. Além disso, o CAM é capaz de simular a usinagem da peça na tela do computador;
- Gerar um programa para ser fornecido ao computador da máquina ferramenta de controle numérico que usinará a peça;
- Estudo de tempos de produção envolvidos na fabricação da peça.

Dentre as tarefas acima, a simulação de usinagem é sem dúvida a mais complexa computacionalmente. A idéia é que possuindo o modelo CAD da peça a ser usinada, o CAM irá calcular não somente os parâmetros ótimos de usinagem da peça, mas também especificamente a trajetória da ferramenta durante a usinagem e determinar se haverá colisões da ferramenta com partes da máquina. A Figura 2.6 mostra a determinação da trajetória da ferramenta. Após definida a usinagem o CAM deve poder simulá-la no PC. A Figura 2.7 mostra uma simulação da usinagem.

A escolha de diferentes trajetórias da ferramenta pode representar uma grande influência sobre o tempo necessário para a usinagem, desgaste de ferramentas de corte e qualidade superficial. Isso se agrava na usinagem de superfícies complexas, onde a qualidade do processo não é influenciada apenas pela ferramenta/peça, como é o caso do fresamento comum, mas depende fortemente da utilização adequada da estratégia de corte (WEINERT 2000).

A pesquisa realizada por WEINERT (2000) constatou uma grande diferença nos resultados obtidos por dois processos de usinagem, que se distinguiam apenas pelas estratégias de corte utilizadas para o acabamento de uma matriz de estampa.

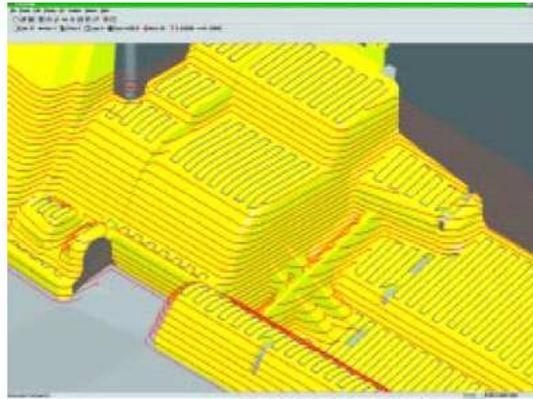


Figura 2.4: Determinação da trajetória.(SILVA 2001)

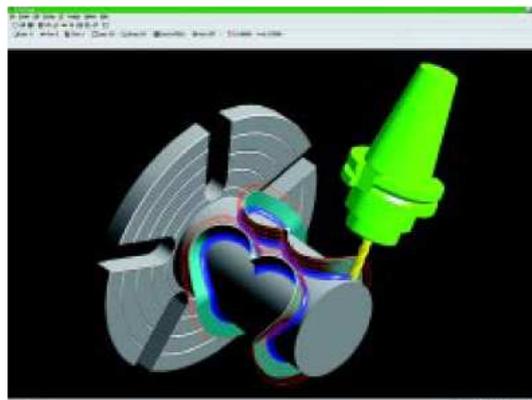


Figura 2.5: Simulação da usinagem.(SILVA 2001)

A Tabela 2.1 mostra o tempo gasto num fresamento convencional que utilizou passes paralelos, conhecidos como *Raster* e o fresamento com estratégia otimizada, composta por fresamento em curvas de nível nos setores mais íngremes e usinagem em *offset* nos setores planos. Nota-se a diminuição significativa do tempo de usinagem, proporcionando um processo de produção mais rápido.

Tabela 2.1: Tabela comparativa entre estratégias de usinagem (SOUZA 2004)

Fig a) Fresamento Convencional	Fig b) Estratégias Otimizadas
Tempo: 33,28 min.	Tempo: 26,41 min.

Nesse Capítulo abordou-se sobre as ferramentas computacionais, como o CAD, que é o início do ciclo de vida do produto, e o CAM, que gera o programa para o CNC, são de suma importância também num processo de produção, pois garantem uma diminuição no ciclo produtivo e garantem uma qualidade final superior, tornando os produtos mais

---

competitivos (FERNEDA 1999), (SOUZA 2004). No próximo Capítulo serão abordados os assuntos sobre as Máquinas-ferramenta a Controle Numérico, a fim de dar uma visão geral sobre as mesmas.

### 3 Máquinas-Ferramenta a Controle Numérico

Hoje, as máquinas ferramentas a controle numérico computadorizado são encontradas em quase todos lugares, das pequenas oficinas de usinagem às grandes companhias de manufatura. Na realidade quase não existem produtos fabris que não estejam de alguma forma relacionados à tecnologia destas máquinas ferramentas.

Segundo Groover (2001) sistemas de produção fundamentados em máquinas CNC podem ser classificados como de automação programável. Embora esse modelo apresente boa produtividade, os resultados ainda são inferiores àqueles obtidos na produção com equipamentos de automação fixa. Neste caso, máquinas dedicadas são ajustadas ou mesmo projetadas para um único tipo de peça. Esta limitação é compensada pela sensível redução do tempo de produção, obtida principalmente pela possibilidade de execução simultânea de diferentes operações de usinagem ou corte (GROOVER 2001).

A automatização de processos nas pequenas indústrias surge da necessidade de maior precisão e qualidade nos produtos. Mas geralmente esse investimento em maquinário mais moderno que possibilita uma melhora na produção tem seu custo elevado. Assim a grande maioria das pequenas indústrias utiliza uma prática conhecida como *retrofitting* que é uma automatização das máquinas ferramentas convencionais, adicionando equipamentos para satisfazer as necessidades do processo (PANSIERA 2002).

Com o advento do CNC foi possível criar peças com geometrias de extrema complexidade em tempos menores, peças cuja fabricação era possível apenas nas mãos experientes de um exímio mestre modelador. Pode se perceber isto nas linhas aerodinâmicas dos automóveis modernos com formas complexas e bonitas que se alteraram com muita rapidez. Ao contrário do que ocorria no início da fabricação dos automóveis em série, os modelos T de Henry Ford, por exemplo, todos os carros eram iguais e da mesma cor, o consumidor era obrigado a optar entre comprar aquele carro que todos tinham ou ficar sem. Hoje a colocação de um novo modelo é tão rápida, que o consumidor se sente às vezes até incapaz

de se decidir qual escolher (CAPELLI 2002).

### 3.1 Histórico

O NC (*Numerical Control*) é uma técnica de controle dos movimentos das máquinas-ferramentas pela interpretação direta dos programas, que representam instruções codificadas na forma de letras e números. Os dados são interpretados pelo sistema e o sinal de saída é gerado para controlar os componentes da máquina (ADAMOWSKI 2008). As pesquisas para o desenvolvimento do NC iniciaram-se durante a Segunda Guerra Mundial. A necessidade de se produzir peças de precisão e em grandes lotes para a fabricação de aviões de guerra uniram esforços de duas entidades: Instituto de Tecnologia de Massachusetts (MIT), e a *U.S. Air Force* (Força Aérea dos Estados Unidos). Em 1949 surgiu a primeira máquina equipada com controle numérico e, finalmente, em 1953 ocorreu sua demonstração prática. Já no final da década de 1950, os principais fabricantes de aviões utilizavam máquinas com controle numérico (FERNEDA 1999). A grande precisão de usinagem e alta capacidade produtiva eram, e até hoje são características fundamentais para a indústria aeronáutica. No início da década de 1960 essas máquinas começaram a se modernizar, e os padrões EIA 244 (*Environment Impact Assessment 244*) e ASCII (*American Standard Code for Information Interchange*), que em português significa "Código Padrão Americano para o Intercâmbio de Informação") já eram utilizados como sistemas de introdução de dados. Em 1977, as máquinas com CNC foram equipadas com microprocessadores. Hoje, essas máquinas podem operar com um número indefinido de eixos. O CNC é um computador dedicado ao movimento dos eixos de uma máquina operatriz. O movimento de cada eixo é traduzido em grandezas numéricas por dispositivos especiais, tais como sensores e atuadores, e então processado pelo controlador da máquina CNC (FERNEDA 1999).

### 3.2 Arquitetura de uma Máquina CNC

Atualmente, existem três principais tipos de controladores numéricos computadorizados.

- Multiprocessamento com ASIC (Circuito Integrado de Aplicação Específica): Isto permite integração e garante grande confiabilidade, em equipamentos comercial-

mente disponíveis tais como FANUC21T, FAGOR-800, MITSUBISHI C3/C3-S.

- *PC front end*: Este é um CNC tradicional, como uma caixa preta adicionada ao microcomputador, tais como os equipamentos Fanuc série 150/160/180/210 e Siemens 840D.
- Cartão de controle do movimento com PC: Este sistema tem uma placa de controle do movimento colocada em um microcomputador. Essa placa, geralmente baseada em DSP (*Digital Signal Processor*), executa as tarefas e melhora o desempenho da máquina e o microcomputador fica com as funções que não necessitam de processamento em tempo-real (WANG 2006).

Embora o primeiro tipo de estrutura permita alta integração e garanta uma grande confiabilidade, não é aberta e nem reconfigurável. No segundo tipo somente a IHM (Interface Homem-Máquina) é aberta, a parte fundamental de NC é ainda segredo para o usuário. O terceiro tipo é nomeado como controlador numérico baseado em PC ou PC-CNC. Atualmente, a solução para a abertura controlador é geralmente baseado nesta arquitetura. É mais flexível do que um sistema proprietário NC, mas estritamente falando, não está totalmente aberta, porque ainda contém os cartões de controle proprietário. No outro lado, se este controlador aberto usa *Microsoft Windows* enquanto sistema operacional, neste caso a abertura daqueles sistemas baseados no PC-CNC não é o OAC (*Open Architecture Controller*) do *hardware* ao nível de controle em *software* (WANG 2006). Uma solução para isto seria o uso de sistema operacional de arquitetura aberta tipo o Linux. Isto será discutido na seção 5.1

Baseado em (CAPELLI 2006) a Figura 3.1 mostra uma arquitetura típica do CNC e seus periféricos, tais como: o *hardware* IHM, os *drivers* dos eixos X, Y e Z, os sensores e atuadores. Nesta arquitetura todo o processamento é feito no próprio CNC. No exemplo a arquitetura refere-se a uma fresadora de 3 eixos.

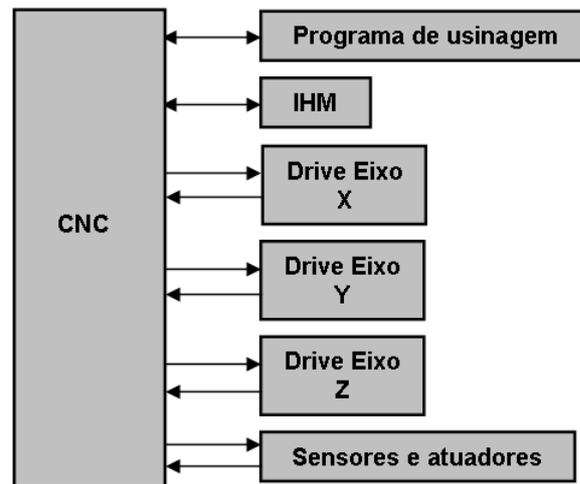


Figura 3.1: Arquitetura típica do CNC e periféricos. Adaptado de (CAPELLI 2006)

A Figura 3.2 mostra uma outra arquitetura CNC também baseada em (CAPELLI 2006), e para uma fresadora de 3 eixos que funciona em conjunto com um CLP (Controlador Lógico Programável) e seus periféricos, tais como: o *hardware* IHM, os *drivers* dos eixos X, Y e Z, os sensores e atuadores. Essa arquitetura tem o tratamento do arquivo de programa de usinagem na CPU, o controle e monitoramento dos *drivers*, sensores e atuadores é feito pelo CLP.

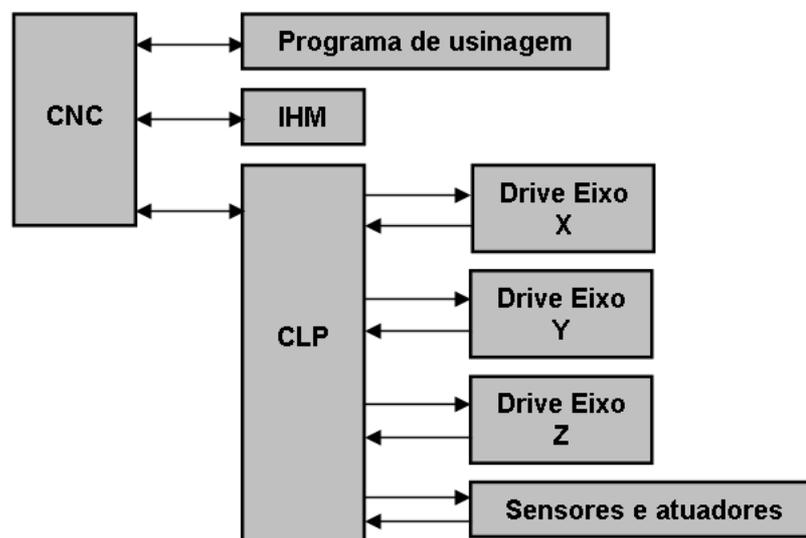


Figura 3.2: Arquitetura nova do CNC com CLP. Adaptado de (CAPELLI 2006)

A diferença principal entre as duas arquiteturas é a presença do CLP, que permite um melhor desempenho na máquina-ferramenta à CNC pois, ao invés de ter um processamento local, tem um processamento distribuído. No passado o controlador numérico gerenciava

todos os processos, desde as informações de programação até os controles dos eixos da máquina, sensores e atuadores. Com o advento dessa arquitetura o CNC fica responsável pela interpretação do programa NC e o CLP fica responsável pelo controle dos eixos da máquina, sensores e atuadores (CAPELLI 2006).

A seguir é mostrado e explicado o conjunto de componentes que constituem a arquitetura de um controle numérico computadorizado

### 3.2.1 Interface Homem-Máquina - (IHM)

A interface homem-máquina, mais conhecida como IHM, é a unidade responsável pela introdução e leitura de dados. É através dela que o programa é instalado na máquina. A Figura 3.3 mostra uma IHM da empresa ROMI.

A IHM pode ter um processador próprio. Essa técnica tem duas finalidades: poupar tempo de processamento ao CNC e evitar cabos longos entre o teclado alfanumérico e a CPU (SIEMENS 2008), (ROMI 2008).



Figura 3.3: Interface Homem-Máquina.(ROMI 2008)

### 3.2.2 *Drives* de controle

Esses são conjuntos de circuitos eletrônicos, destinados a controlar a rotação do motor e uma série de outros componentes auxiliares.

O CNC lê, interpreta e executa cada um dos códigos que compõem o programa de usinagem da peça. Por exemplo, supondo que o eixo longitudinal da fresadora se desloque para a coordenada 150 mm, com um avanço de 200 mm/min, para que o CNC compreenda isso, deve-se programar "G1 X150. F200" (SILVA 2001).

Para executar uma ordem, o CNC envia uma mensagem ao CLP, que por sua vez, envia uma mensagem ao *driver* de controle e o motor que está ligado a um eixo da máquina.

Para verificar se a ordem dada pelo CNC foi cumprida, usam-se os sensores. Eles informam ao CNC o que está ocorrendo com a máquina e permitem a correção de eventuais desvios entre o programado e o real.

O CLP comanda os servomotores através de *drivers* de potência: o servoconversor é que controla a potência do servomotor através de um sinal analógico de comando vindo do CLP, controlando assim o torque e a velocidade. A Figura 3.4 ilustra um servomotor e um servoconversor (WEG 2008).

Para os servomotores ligados aos eixos coordenados ( X, Y e Z ), têm-se dispositivos que indicam a posição em que o eixo se encontra. Esse dispositivo é geralmente o encoder, que será brevemente apresentado na seção 3.2.3.1.



Figura 3.4: Servomotor e Servoconversor.(WEG 2008)

### 3.2.3 Entradas/Saídas (E/S)

Cada eixo possui um sensor de fim-de-curso que serve para obter a posição de zero da máquina e também atua como elemento de segurança em relação aos limites físicos dos

eixos. Além disso, dispositivos e acessórios especiais (bombas de refrigeração, acionamento de contraponto) comunicam-se com o CNC através do módulo E/S. Normalmente, tem-se um recurso no CNC que permite visualizar o *status* de cada endereço de E/S na tela da IHM. Por exemplo, caso uma válvula não obedeça a um comando, pode-se entrar na tela de *status* e verificar se o defeito é devido à ausência de comando do CNC, ou se a própria válvula está com defeito (CAPELLI 2006).

### 3.2.3.1 Sensores

Os sensores são dispositivos sensíveis a alguma forma de energia do ambiente que pode ser luminosa, térmica, cinética, etc, e essas energias têm relação com as grandezas a serem medidas, como: temperatura, pressão, velocidade, corrente, aceleração, posição.

Um sensor nem sempre tem as características elétricas necessárias para ser utilizado em um sistema de controle. Normalmente o sinal de saída deve ser manipulado antes da sua leitura no sistema de controle. Isso geralmente é realizado com um circuito de interface para produção de um sinal que possa ser lido pelo controlador. Esse circuito de interface é denominado como transdutor (PAZOS 2002).

Os transdutores transformam grandezas físicas, tais como, temperatura ou pressão em um sinal de tensão ou corrente que pode ser facilmente entendido por um sistema de controle (TURBAN E. e RAINER 2003).

Muitas vezes os termos "sensor" e "transdutor" são usados sem distinção. Mas o transdutor é o instrumento completo que engloba sensor e todos os circuitos de interface necessários para que seja utilizado nas mais diversas aplicações (PAZOS 2002), (TURBAN E. e RAINER 2003).

Um dos sensores mais utilizados em máquinas CNC é o *encoder*, o qual consiste de um transdutor eletromecânico baseado em princípio óptico que converte rotação (movimento angular) em uma seqüência de pulsos eletrônicos. Ele pode ser agregado ao próprio motor, situação muito comum em servomotores, ou separado.

Existem dois tipo de *encoders*, o incremental e o absoluto. O princípio de funcionamento do *encoder* incremental é o mesmo encontrado em alguns tipos de *mouse* de PC, como ilustra a Figura 3.5.

Quando o disco sofre um deslocamento angular interrompe a passagem de luz, gerando

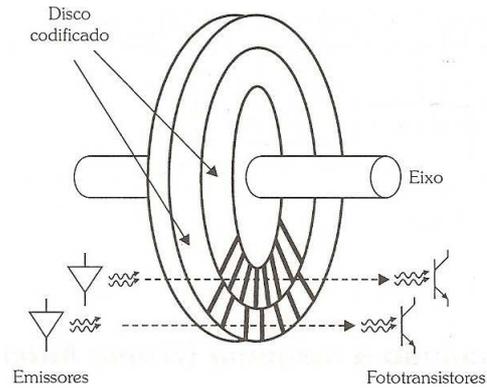


Figura 3.5: Princípio de funcionamento do encoder. (CAPELLI 2006)

um pulso. Este pulso representa um certo ângulo mínimo, que define a resolução do sistema. (THOMAZINI D. e ALBUQUERQUE 2005), (CAPELLI 2006)

Um *encoder* pode ter dois ou mais canais, os quais são chamados de A, B, C, sendo que em cada canal existe um par emissor-receptor de luz. A eletrônica interna do dispositivo se encarrega da geração de pulsos elétricos nestes canais. Esses pulsos por sua vez são enviados aos controladores que monitoram a velocidade e posição.

O sentido de giro é determinado pela fase entre "A" e "B". Já a medida de deslocamento pela quantidade de pulsos que chega ao CLP. Na Figura 3.6 são ilustrados os canais do encoder.

O princípio de funcionamento do *encoder* absoluto consiste em uma série de trilhas. Cada trilha tem seu próprio fotodetector, e são organizadas de forma que todos os fotodetectores leiam as trilhas ao mesmo tempo. O *encoder* absoluto tem uma vantagem em comparação ao incremental, por exemplo, quando há uma queda de energia e a máquina vier a desligar o *encoder* absoluto armazena-se a posição da máquina, não necessitando resetar a mesma.

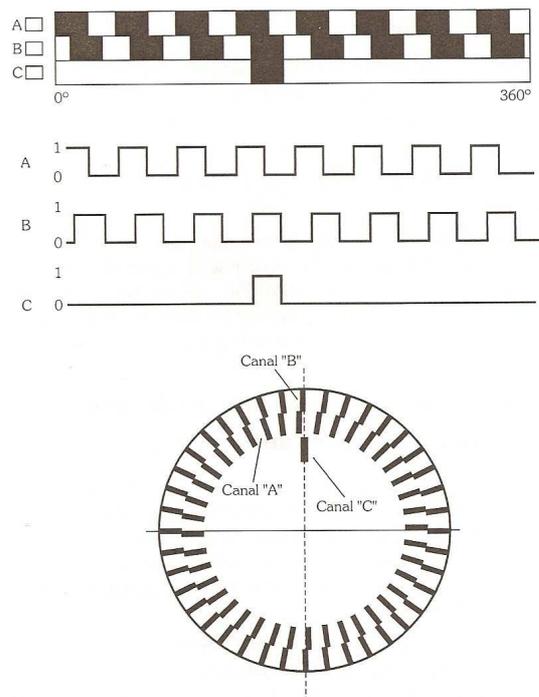


Figura 3.6: Canais do encoder.(CAPELLI 2006)

### 3.2.3.2 Atuadores

Os atuadores são dispositivos que modificam uma variável controlada. Recebem um sinal vindo do controlador e agem sobre o sistema controlado. Geralmente trabalham com potência elevada. Exemplos de alguns atuadores utilizados em máquinas-ferramenta à controle numérico são (TURBAN E. e RAINER 2003):

- Válvulas (pneumáticas, hidráulicas);
- Relés (estáticos, eletromecânicos);
- Cilindros (pneumáticos, hidráulicos);
- Motores (step-motor, servomotor);
- Solenóides.

Esses mecanismos são denominados atuadores, pois sua função é aplicar ou fazer atuar energia mecânica sobre uma máquina, levando-a a realizar um determinado trabalho.

(BONACORSO N.G. e NOLL 2007), (FIALHO 2003).

### 3.3 Funcionamento e Programação da Máquina-Ferramenta

A programação de uma máquina CNC segue um padrão internacional. O mais utilizado é a linguagem ISO6983-1, conhecida como código-G onde tem-se sempre uma letra que indica a operação, seguida de um número que indica a modalidade da operação (HOUNSELL 2008), (SCHÜTZER 2008).

A tarefa da programação é traduzir uma representação geométrica de um componente em movimentos realizados pela máquina. Primeiramente, em especificações para as operações da máquina-ferramenta, então em um programa com instruções para o controlador, que geram a geometria da peça.

Muitas máquinas NC ainda são programadas manualmente, em particular, aquelas que são relativamente simples. A programação de formas complexas sempre foi difícil, entretanto, os computadores tem auxiliado nesta tarefa. Em particular, eles auxiliam em cálculos matemáticos para determinar a trajetória da ferramenta, com compensação do raio e distribuição de passes para formar uma peça, e têm o mérito de reduzir erros na programação. O primeiro desses métodos envolve a linguagem de programação para definir a geometria da peça e o movimento da ferramenta de corte que é interpretado para resultar nos caminhos da ferramenta cortante em relação à peça. A segunda envolve a extração de dados diretamente do modelo vindo do CAD. O esquema é mostrado a Figura 3.8, inclusive com a parte de programação manual. No passo auxiliado por computador nota-se que o trajeto da ferramenta é produzido para o formato de uma máquina genérica, e depois convertida por um pós-processador para o formato específico da máquina (HOUNSELL 2008), (SCHÜTZER 2008)

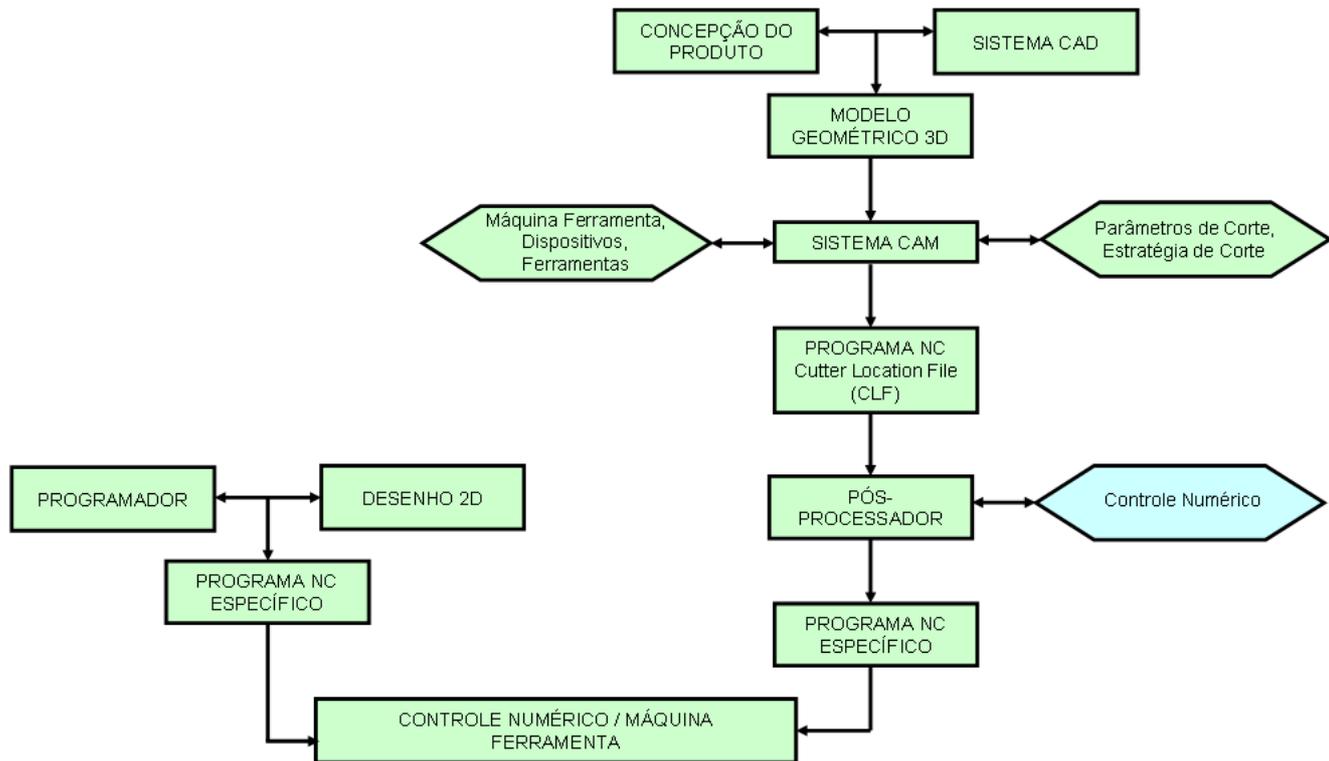


Figura 3.7: Geração do programa. (SCHÜTZER 2008)

A programação é gerada seguindo uma sintaxe bem definida. Basicamente, a máquina recebe instruções em seqüências de blocos contendo comandos para preparar a operação da máquina, parâmetros, dimensões e velocidade.

Eles são classificados em (HOUNSELL 2008), (SCHÜTZER 2008):

- Identificador N (*number sequence*): número de seqüência das instruções;
- Identificador G (*preparatory functions*): prepara a MCU (*machine control unit*);
- Identificadores X, Y, Z, A, B ou C (*dimensional data*): contém a localização e orientação do eixo para movimento da ferramenta cortante;
- Identificador F (*feed functions*): usado para especificar a taxa de alimentação do cortador a ser aplicada;
- Identificador S (*speed functions*): usado para especificar a velocidade de rotação, ou preparar a velocidade para operação em superfície;
- Identificador T (*tools functions*): especifica a ferramenta a ser utilizada;

- Identificador M (*miscellaneous functions*): designar um modo de operação.

A Figura 3.8, mostra uma linha de programa em código-G (ISO6983).

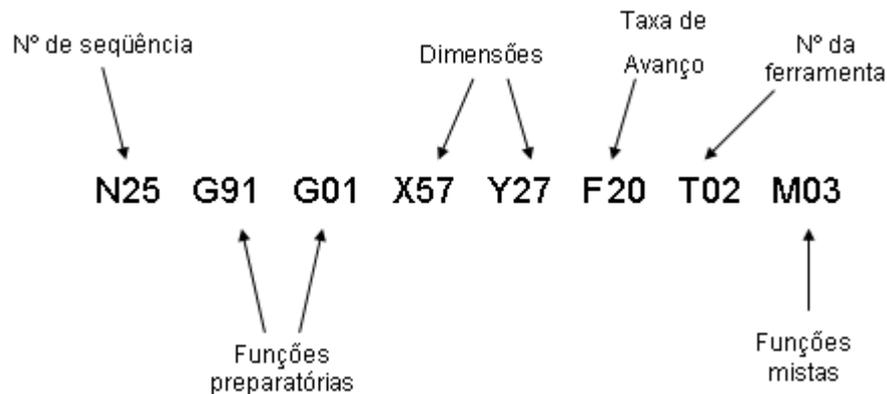


Figura 3.8: Linha de programa em ISO6983. (WILLIAMS 1988)

Nos tipos de operações que devem ser programadas, as funções preparatórias e as mistas são muito diversificadas. As Tabelas 3.1 e 3.2, mostram alguns comandos referentes ao código G (preparatórios) e o código M (Mistas), e na Figura 3.9 um exemplo de programa CNC.

Tabela 3.1: Comandos preparatórios (Código G) (ISO6983-1 1982)

G00	Movimento rápido ativo
G01	Movimento linear ativo
G02	Movimento circular horário ativo
G03	Movimento circular anti-horário ativo
G04	Ciclo de tempo ativo
G40	Cancela compensação automática do raio da ferramenta
G41	Ativa compensação automática do raio da ferramenta à esquerda
G42	Ativa compensação automática do raio da ferramenta à direita
G70	Dimensões em polegadas
G71	Dimensões em milímetros
G90	Dimensões absolutas
G91	Dimensões incrementais

Tabela 3.2: Comandos mistos (Código M) (ISO6983-1 1982)

M00	Parada do programa
M01	Parada opcional do programa
M03	Eixo árvore sentido horário
M04	Eixo árvore sentido anti-horário
M05	Parada do eixo árvore
M06	Troca de ferramenta
M08	Liga refrigeração da ferramenta
M09	desliga refrigeração da ferramenta
M30	Fim do programa

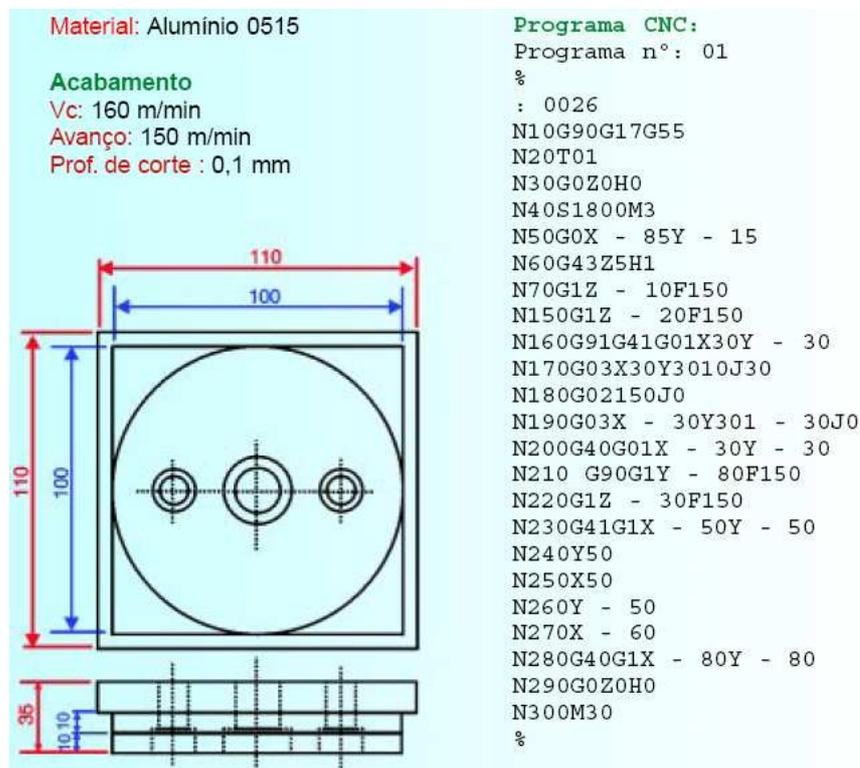


Figura 3.9: Exemplo de programa CNC. (CAPELLI 2002)

Nesse Capítulo abordou-se sobre Máquinas CNC abordando as arquiteturas típicas e componentes que fazem parte de uma máquina CNC, seu funcionamento e programação. No próximo Capítulo serão abordados assuntos sobre as Normas ISO10303 ou STEP e ISO14649 ou STEP-NC responsáveis pela padronização das trocas de informações entre os sistemas que compõem a manufatura.

## 4 Norma ISO10303 ou STEP e Norma ISO14649 ou STEP-NC

Os sistemas CAD, CAPP, CAE e CAM são muito utilizados no desenvolvimento de produtos de engenharia, e a troca de dados entre estes sistemas enfrenta em inúmeras dificuldades. Para ultrapassar estas dificuldades e prover um sistema neutro de registro das informações do produto, a ISO (*International Standards Organization*), desenvolveu o conjunto de normas 10303 – STEP (*Standards for Exchange Product Data Model*). E, posteriormente desenvolveu a norma ISO14649, conhecida como STEP-NC (*Standards for Exchange Product Data Model – Numerical Control*), que visa a construção de um padrão neutro de dados utilizados entre as máquinas CNC (NEWMAN 2003).

### 4.1 Norma ISO10303 ou STEP

No projeto e na fabricação, muitos sistemas são usados para controlar dados técnicos do produto. Cada sistema tem seus próprios formatos de dados, assim a mesma informação tem que ser atualizada muitas vezes nos vários sistemas que conduzem à redundância e aos erros. O *NIST* (*National Institute of Standards and Technology*) estimou que a incompatibilidade dos dados causa prejuízo em torno de 90 milhões de dólares por ano para a indústria de manufatura (BRUNNERMEIER. S. B. e MARTIN. S. A. 1999).

Entre as dificuldades encontradas nos sistemas de engenharia, estão as diferentes representações e interpretações dos modelos dos produtos utilizados por cada um dos sistemas, o problema do custo elevado para troca de dados e o crescimento acelerado da indústria de CAD/CAM. Desde os primórdios da indústria de CAD existe a necessidade da criação de um padrão de troca de informação. Soluções como *Initial Graphics Exchange Specifications* (IGES), *Standard d Echange et de Transfert* (SET) e *Verband Der Automobilindustrie* (VDA) foram propostas, mas não supriam as necessidades da indústria (ISO10303-1 1994). Os tipos de sistemas que usam a STEP/STEP-NC são mostrados na Figura 4.1.

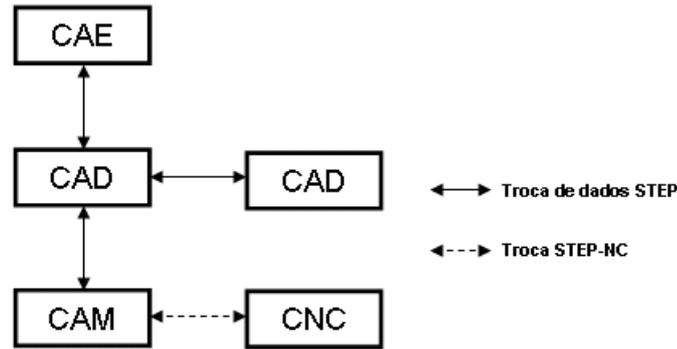


Figura 4.1: Tipos de Sistemas. (STEPTOOLS 2008)

A primeira solução encontrada para troca de dados entre sistemas, é chamada tradução direta, que implica no desenvolvimento de um par de tradutores para cada par de sistemas. A outra solução é a tradução indireta de dados, usando um formato intermediário chamado de formato neutro. O uso de um formato neutro implica que, cada sistema necessita somente de um par de tradutores para a importação/exportação dos dados, tornando-o utilizável a todos os modelos que seguem esse tipo de tradução (ROSSO Jr. 2005). A vantagem da tradução direta está no tamanho dos arquivos gerados, e na velocidade para leitura dos mesmos, pois é uma solução *one-stage*, sendo que, quando um tradutor é feito por uma empresa especializada, a perda semântica é baixa ou nenhuma. O inconveniente para esse tipo de tradução é quando o número de sistemas CAD/CAPP/CAE/CAM aumenta, fazendo com que o número de tradutores aumente significativamente, o que torna a solução muito cara. A Tabela 4.1 mostra em (a), a equação que calcula o número de tradutores necessários, quando há mais de 3 (três) sistemas envolvidos, usando tradutores diretos. Em (b) na Tabela 4.1, é apresentada a equação que calcula o número de tradutores necessários para utilizar tradutores indiretos. Têm-se “N” como número de tradutores necessários e “n” como número de sistemas envolvidos (KERN 1997).

Tabela 4.1: Equações para cálculo do número de tradutores (KERN 1997).

(a)	$N = n \times (n-1)$
(b)	$N = n \times 2$

Baseado na equação apresentada na Tabela 4.1 (a), a Figura 4.2 mostra um exemplo do número de traduções necessárias, utilizando tradução direta entre quatro sistemas. Nota-se que o crescimento do número de tradutores exponencial, à medida que novos

*software* são inclusos no processo de utilização dos dados. Na Tabela 4.2 é mostrado o número de tradutores diretos e indiretos necessários para a troca de informações entre os diferentes sistemas.

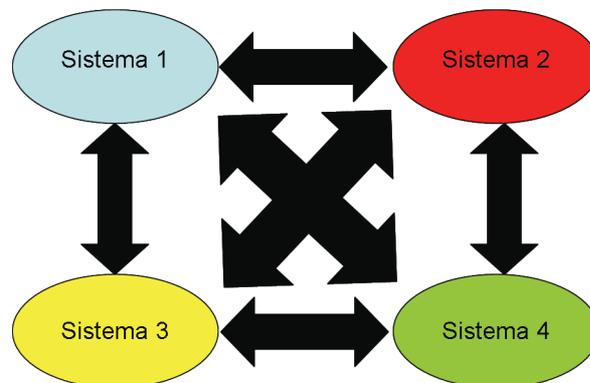


Figura 4.2: Comunicação entre sistemas utilizando tradução direta. (ZEID 1991) apud (ROSSO Jr. 2005)

Tabela 4.2: Quantidade de tradutores.

No de sistemas	Tradutores diretos	Tradutores indiretos
3	6	6
4	12	8
5	20	10
6	30	12
7	42	14
8	56	16
9	72	18
10	90	20

A Figura 4.3 apresenta como ficaria traduções indiretas, utilizando o formato neutro de dados entre quatro sistemas. Nota-se que, para a inclusão de mais um software no processo seriam necessários somente mais dois tradutores.

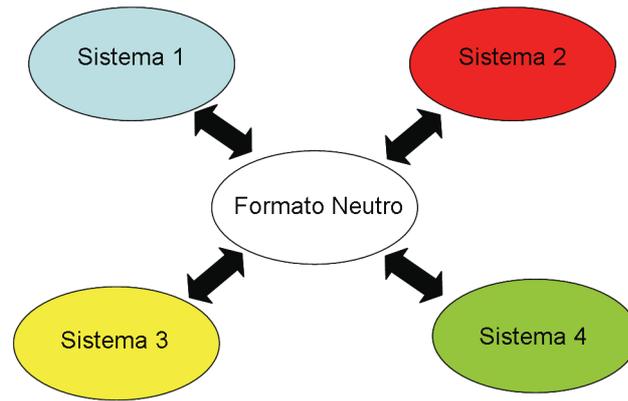


Figura 4.3: Comunicação entre sistemas utilizando tradução indireta.(ZEID 1991) apud (ROSSO Jr. 2005)

Para resolver o problema de falta de um padrão internacional aceito pela maior parte dos usuários, no início da década de 1990, a ISO lançou o Padrão para o Intercâmbio de Modelos de Dados de Produtos, conhecido informalmente como STEP, que tem como principal objetivo, a criação de um modelo de dados de produtos de forma a possibilitar a troca de dados, através de um sistema neutro. Possui suporte ao desenvolvimento durante toda a cadeia de processo, de maneira que as informações permaneçam integradas e consistentes durante todo o ciclo de vida do produto. Faz com que os sistemas de troca de dados se tornem eficientes entre diversos setores das empresas, fabricantes e fornecedores (KERN 1997). A norma é constituída por protocolos, onde cada um descreve um domínio de aplicação. Na prática o padrão é executado dentro do *software* em um computador e associado a aplicações particulares de engenharia, sendo seu uso transparente ao usuário. As descrições são modelos de informações que capturam a semântica e fornecem estruturas padronizadas para serem executadas por computador (XU 2004).

#### 4.1.1 Estrutura do STEP

Segundo Rosso Jr. (2005), a STEP foi estruturada para ser capaz de descrever os produtos de vários domínios e para testar se as aplicações são construídas utilizando os métodos aderentes à norma STEP.

O padrão STEP está dividido em várias partes. Estas partes abrangem tópicos como métodos utilizados para descrever a norma, arquiteturas de implementação, procedimentos de teste de conformidade, modelos de informação dos recursos e protocolos de aplica-

ção. As partes podem ser divididas em: Métodos de Descrição, Modelos de Informação, Protocolos de Aplicação, Métodos de Implementação e Ferramentas de Conformidade (LOFREDO 2008).

A estrutura da STEP é descrita na parte 3.2 da ISO10303 (ISO10303-1 1994), como mostra a Tabela 4.3 e detalhada em seguida (ROSSO Jr. 2005), (LOFREDO 2008).

Tabela 4.3: Estrutura dos protocolos da STEP (ROSSO Jr. 2005), (LOFREDO 2008)

<b>Documentos STEP</b>	<b>Partes</b>
Métodos de Descrição	11 a 19
Recursos Integrados:	
Recursos Genéricos	41 a 99
Recursos de Aplicação	101 a 199
Métodos de Implementação	21 a 29
Métodos e Testes de Conformidade	31 a 39
Protocolos de Aplicação	201 a 1199
<i>Suítes</i> de Testes Abstratos	1201 a 2199

- **Métodos de Descrição:** esses métodos são baseados na idéia de que a descrição de dados do produto não pode ser ambígua. Fornecem as especificações das linguagens que são usadas para a criação dos padrões;
- **Recursos Integrados:** divididos em recursos genéricos e recursos de aplicação. Os recursos genéricos fornecem um modelo único de informação para a manufatura do produto. Os recursos de aplicação fornecem meios flexíveis para criar a integração, através de diferentes aplicações;
- **Métodos de Implementação:** especificam uma técnica utilizada pelos sistemas computadorizados para troca de dados do produto, que são descritos utilizando a linguagem EXPRESS<sup>1</sup>. Possuem entre outras representações: texto simples (parte 21); SDAI *standard data access interface* (parte 22), linguagem de programação C++ (parte 23), linguagem de programação C (parte 24), linguagem de programação JAVA (parte 27) e representação através do XML<sup>2</sup>;

<sup>1</sup>Linguagem formal de especificação de dados definida na parte 11 do padrão STEP

<sup>2</sup>*Extensible Markup Language*

- **Métodos e Testes de Conformidade:** especificam como uma implementação da ISO10303 pode ser testada quanto à conformidade com os padrões. A falta destes métodos era um dos principais pontos de falhas de padrões como IGES, SET e VDA;
- **Protocolos de Aplicação:** especificam os requisitos para dados de aplicações para um domínio específico da engenharia. Descrevem as estruturas de dados para um modelo de produto;
- **Suítes de Testes Abstratos:** descrevem os testes que são usados para determinar se uma implementação está de acordo com os Protocolos de Aplicação;

Lofredo (2008) mostra uma estrutura de alto nível, que divide a norma em duas partes: Infra-Estrutura e Modelos de Informação, conforme ilustrado na Figura 4.4.

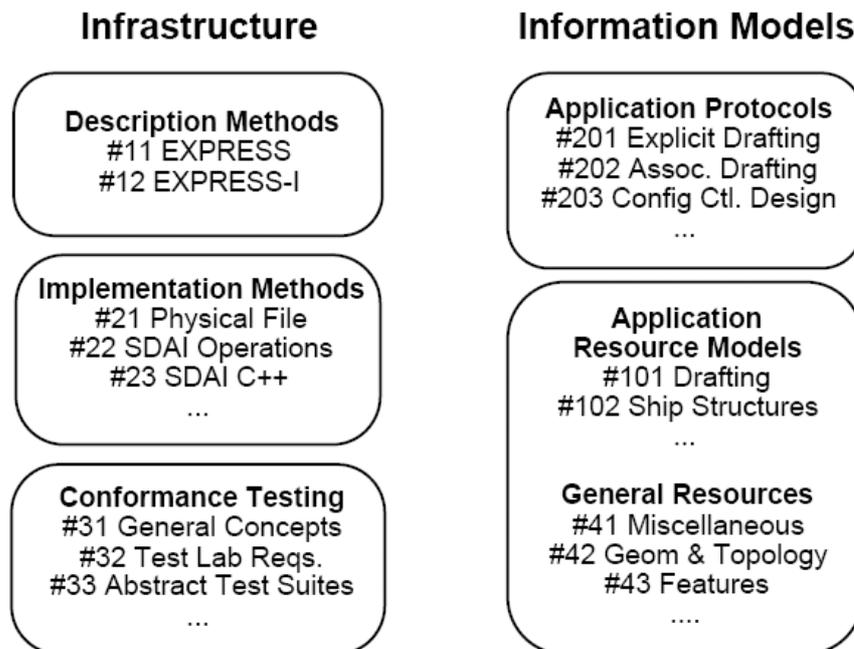


Figura 4.4: Estrutura de alto nível da ISO10303.(LOFREDO 2008)

## 4.2 Norma ISO14649 ou STEP-NC

Atualmente as máquinas CNC são desenvolvidas com capacidades tais como: controle multi-eixo, compensação de erro e a manufatura multi-processo. Entretanto, estas capacidades tornaram a tarefa de programação cada vez mais difícil. As máquinas CNC atuais usam uma linguagem baseada em tecnologia de 50 anos de idade, a ISO6983 (também

chamado G-code) (ISO6983-1 1982). Algum esforço foi feito para diminuir esse problema, levando para uma tendência de controle de arquitetura aberta. Embora estes desenvolvimentos tenham melhorado as ferramentas de *software* e a arquitetura de sistemas CNC, fabricantes e usuários ainda procuravam uma linguagem comum para o CAD, o CAPP, o CAM, e o CNC, que integrasse e traduzisse o conhecimento de cada estágio sem a perda de informação. Embora existam muitas ferramentas CAM que dão suporte para a manufatura NC, o problema da adaptação e interoperabilidade eram e ainda são, considerados como uma das questões básicas em limitar o uso mais amplo destas ferramentas (ROSSO Jr. 2003).

Algumas limitações derivadas da utilização do *G-code* são as seguintes (ROSSO Jr. 2003):

- Caso não possuam uma configuração semelhante e controladores fornecidos pelo mesmo fabricante, não é possível a troca de programas entre máquinas CNC;
- Não existe um padrão que atualize as bases de dados de processo e projeto do produto.

Para evitar esses tipos de limitações, com base na STEP, em meados da década de 1990 a ISO iniciou o desenvolvimento de uma nova norma para solução dos problemas apresentados pelo ISO6983. Iniciava-se a construção da ISO14649 ou STEP-NC. Esta norma proporciona uma facilidade para a manufatura baseada em CNC, com um modelo de dados orientado a objetos e uma estrutura detalhada de interface de dados com programação baseada em *Features* (ROSSO Jr. 2003). O padrão, ISO 14649 Part 1/10/11/12/111/121, conhecido informalmente como STEP-NC, que foi desenvolvido pelos fabricantes, usuários e os institutos acadêmicos em todo o mundo, visa fornecer um modelo de informação para uma nova geração de CNC's inteligentes. O modelo de informação representa um padrão comum que visa especificar a programação NC, facilitando assim a troca de dados entre os sistemas envolvidos e evitando a perda de dados ao exportar arquivos (XU 2006) (ROSSO Jr. 2002) (NEWMAN 2003).

Na Figura 4.5 ilustra-se uma integração típica da manufatura atualmente em uso. Nesta estrutura, o projeto é a parte informatizada, através da utilização do *software* CAD. O projeto CAD é então transferido para o CAPP/CAM por *software* através de um dos padrões da indústria, ou seja, o IGES, DXF ou STEP. O *software* CAPP/CAM então determina a trajetória da ferramenta para a fabricação da peça, baseado em uma

máquina pré-definida na controladora e na configuração de ferramentas. O engenheiro de processo supervisiona a geração de trajetória da ferramenta e depois de verificar as instruções passa pelo pós-processador que converte para a linguagem do controlador e algumas vezes otimiza as instruções de usinagem. O resultado são instruções de usinagem que são enviadas ao controlador (NASSEHI 2006).

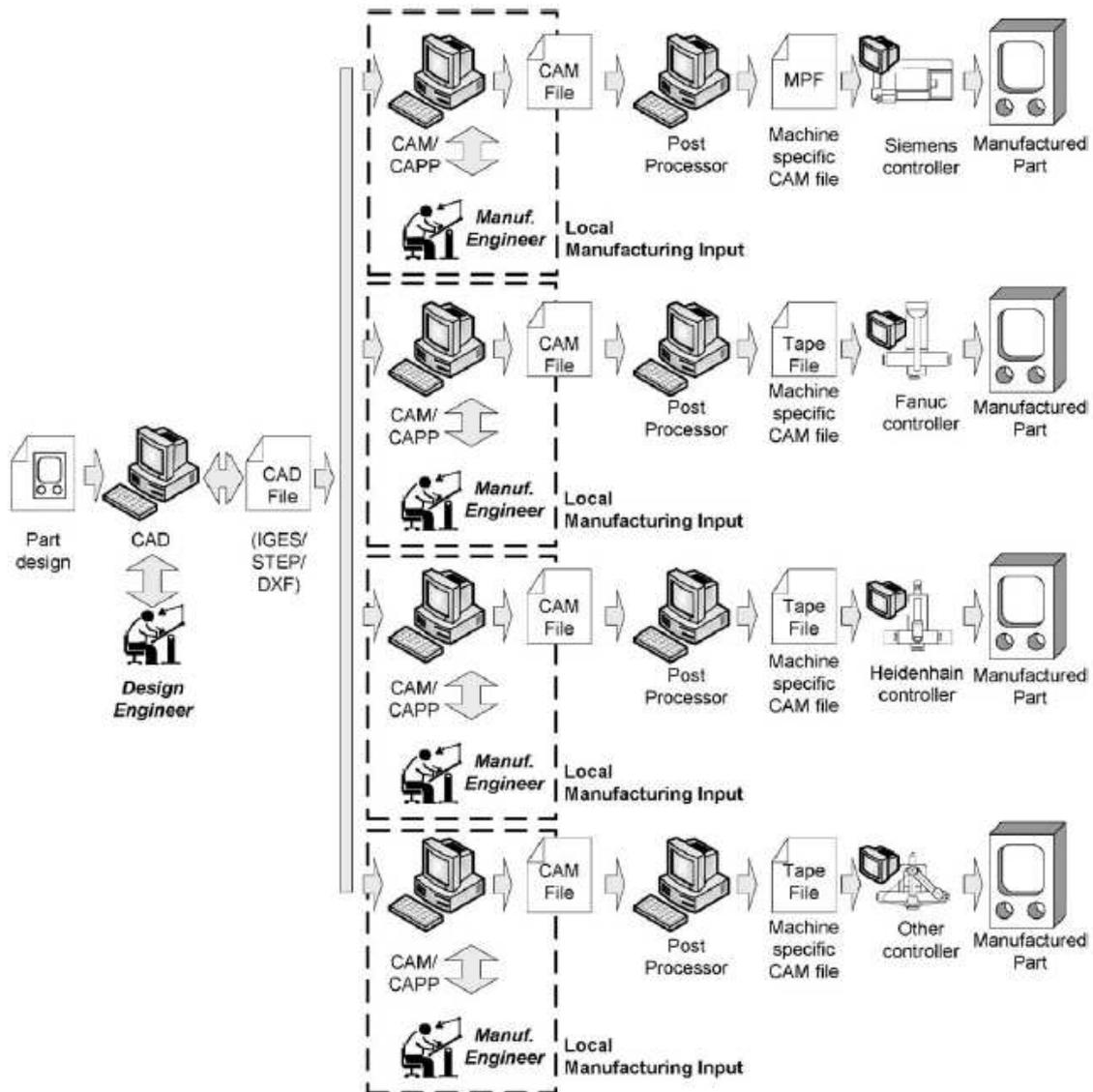


Figura 4.5: Sistema Atual de integração CAD/CAPP/CAM/CNC.(NASSEHI 2006)

Embora este tipo de solução possa ser praticada em alguns casos, existem porém muitas outras situações em que há uma necessidade de transferência de algumas instruções de usinagem juntamente com a geometria que não é possível de fazer através de modelos CAD. Além disso, a interpretação de modelos CAD em diferentes meios pode resultar em perdas de informação referentes ao projeto. A norma STEP surgiu como a plataforma comum de transferência dos dados de produtos e a norma STEP-NC para a engenharia

de processo e usinagem. STEP-NC propõe uma estrutura hierarquizada de dados para a informação de usinagem.

Um importante benefício que a norma STEP-NC traz, é a habilidade na representação da geometria de um componente nos controladores, através das *features* ou por regiões complexas. Possui ainda um modelo mais detalhado de dados que supera a falta de informação de processo dos arquivos da ISO6983. A utilização da STEP-NC cria a possibilidade de um fluxo de dados bi-direcionais entre sistemas CAD/CAPP/CAM/CNC, permitindo a alteração no projeto em qualquer estágio no ciclo de fabricação do produto, possibilitando uma integração entre os dados, fazendo com que não haja perda de informação (ROSSO Jr. 2005). Na Figura 4.6 é ilustrada a bi-direcionalidade de dados entre os sistemas utilizando STEP-NC na (direita da figura) e a impossibilidade de fluxo bi-direcional de dados na (esquerda da figura) que utiliza ISO6983.

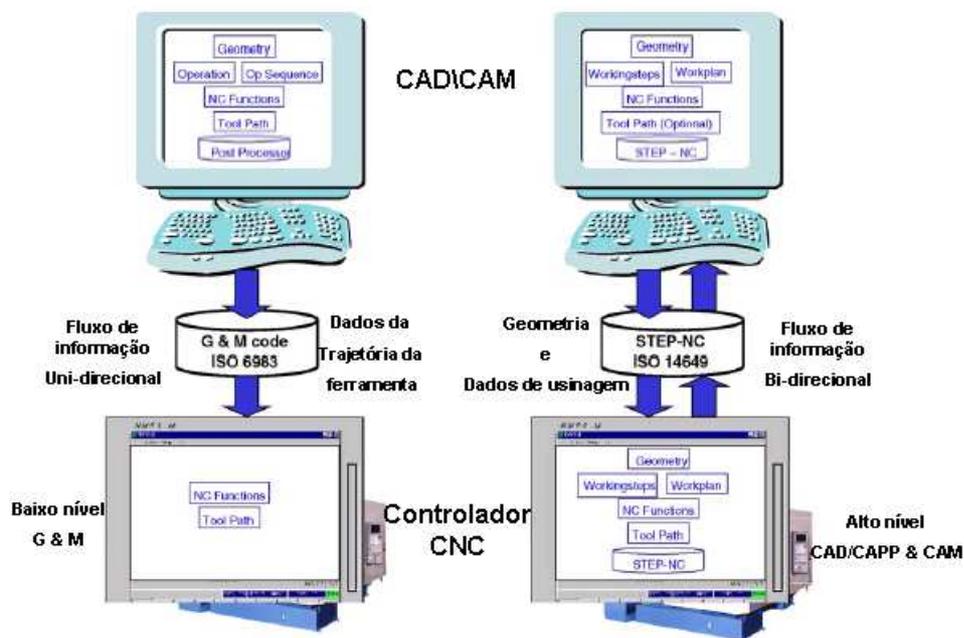


Figura 4.6: ISO6983 (G & M Codes) vs ISO14649 (STEP-NC). (XU 2006), (ALLEN 2003)

Ao contrário do padrão de programação NC em uso corrente (ISO6983), a ISO14649 não é um método de programação e descrição dos movimentos da ferramenta para uma máquina CNC. Em lugar disso, ela fornece um modelo de informação orientada a objeto com dados detalhados e estruturados, uma interface baseada em *features*, tais como: ferramentas usadas, operações de execução e a seqüência de operação. Embora seja possível definir a trajetória da ferramenta usando STEP-NC, o foco do padrão é permitir que estas decisões sejam feitas por uma nova geração de controladores inteligentes aderen-

tes a STEP-NC. Uma vez projetado o processo de fabricação de uma peça baseado em STEP-NC, o mesmo pode ser usado em diferentes tipos de máquinas, pois o controlador tem a capacidade de processar a informação. Ao fazer isto, as máquinas-ferramenta e seus programas serão adaptáveis e interoperáveis, desde que a máquina tenha condições de realizar as operações.

#### 4.2.1 Estrutura da Norma ISO14649

A norma segue a estrutura típica do STEP, onde existe uma norma geral para orientações e muitas partes que descrevem cada ramo de tecnologia e processos como fresamento, torneamento e EDM (*Electrical Discharge Machining*). A Figura 4.7 mostra essa relação entre a ISO10303 e a ISO14649.

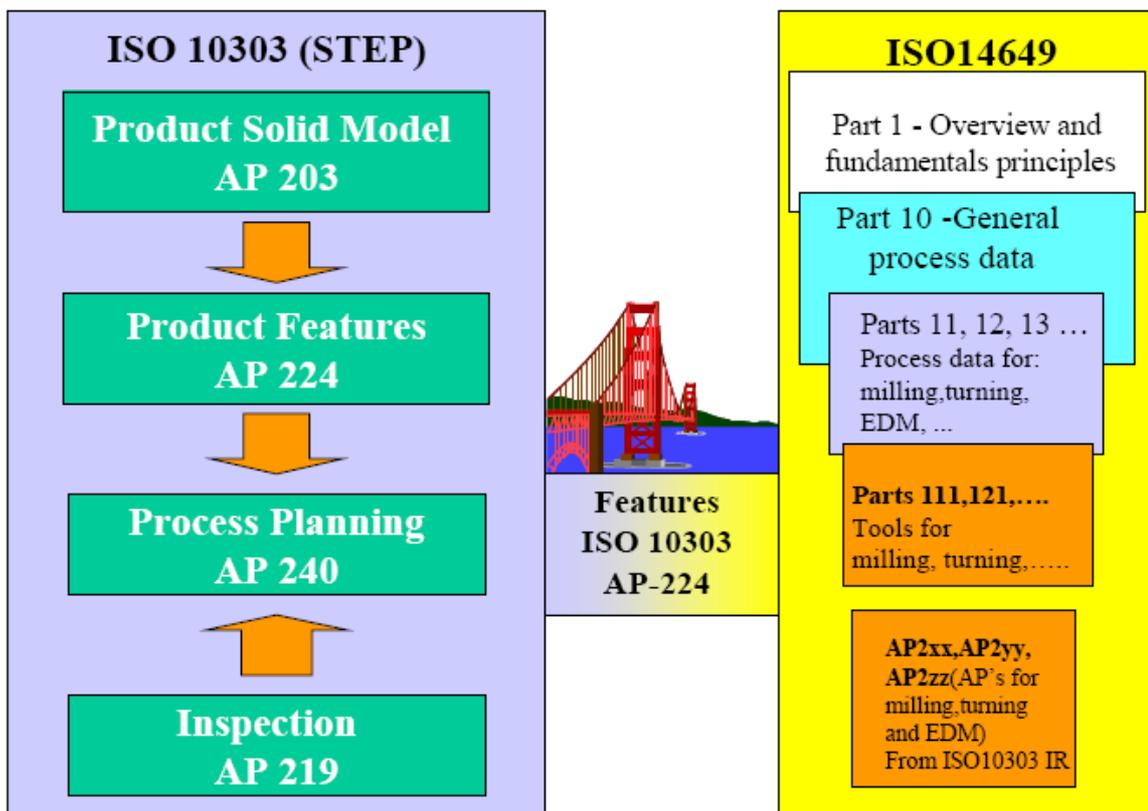


Figura 4.7: Relação entre ISO10303 e ISO14649. (ROSSO Jr. 2005)

A Figura 4.7 está representando a relação entre STEP e seus protocolos de aplicação e a estrutura corrente da norma ISO 14649. A ponte que liga essas duas normas é a tecnologia de *features* a qual serve para transportar um modelo de dados de um produto em um programa de usinagem que contém tecnologia de *features*. As informações do

produto podem ser armazenadas em um arquivo que é compatível com a AP 203 <sup>3</sup>. Suas *features* são extraídas no formato AP 224 <sup>4</sup> em seguida, o processo é planejado utilizando AP 240 <sup>5</sup>. Então, um sistema CAM deve ser utilizado e gera um Programa NC aderente a STEP. Se é necessário a inspeção, AP 219 <sup>6</sup> deve ser utilizada no processo.

#### 4.2.2 Princípios Fundamentais da ISO14649

A ISO 14649 Parte 1 (ISO14649-1 2003) define a base da norma, descrevendo a relação entre a ISO 14649 e ISO 10303, ou seja, a da AP 203, 213 <sup>7</sup>, 224, e 219. Ela estabelece a estrutura geral de um arquivo físico em duas seções: uma intitulada *HEADER*, onde as informações gerais sobre o programa da peça são armazenadas; uma seção chamada *DATA* na qual estão todas as informações sobre a fabricação, tarefas e geometria para produzir uma peça. A segunda seção contém três partes importantes denominadas de *Workplan and Executables*, *Technology Description* e *Geometric Description*. Também define basicamente que esta norma é uma abordagem diferente de programação NC tradição, pois é orientada a objetos de modelos de dados. A Figura 4.8 mostra a estrutura do modelo de dados orientada a objetos para a ISO 14649.

---

<sup>3</sup> *Configuration Controlled 3D Designs of Mechanical Piece Parts and Assemblies*

<sup>4</sup> *Mechanical Product Definition for Process Planning Using Machining Features*

<sup>5</sup> *Process Plans for Machined Products*

<sup>6</sup> *Dimensional Inspection Planning*

<sup>7</sup> AP 213 foi substituída pela AP 240

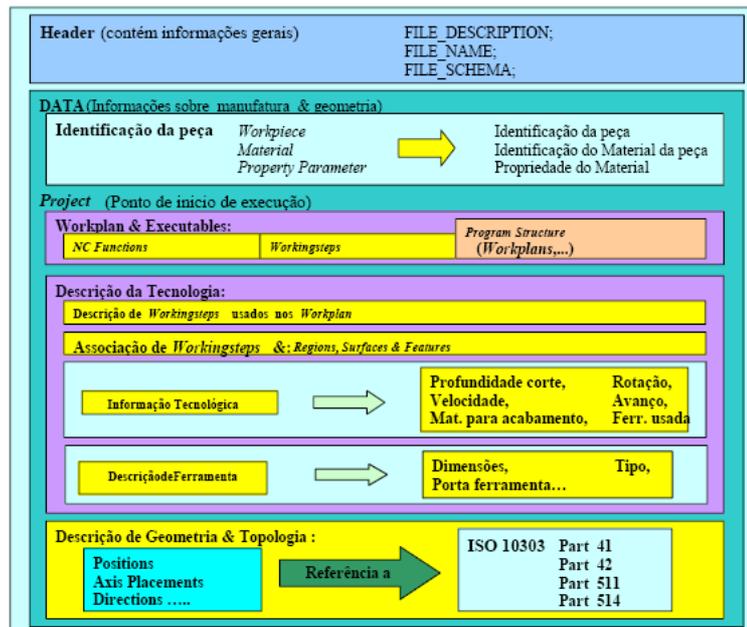


Figura 4.8: Detalhes de uma estrutura típica de um programa STEP-NC. (ROSSO Jr. 2003)

A Figura 4.9 descreve a estrutura geral da organização de dados na ISO14649 que foi detalhada na Figura 4.9 e a Figura 4.10 ilustra um exemplo resumido de programa STEP-NC.

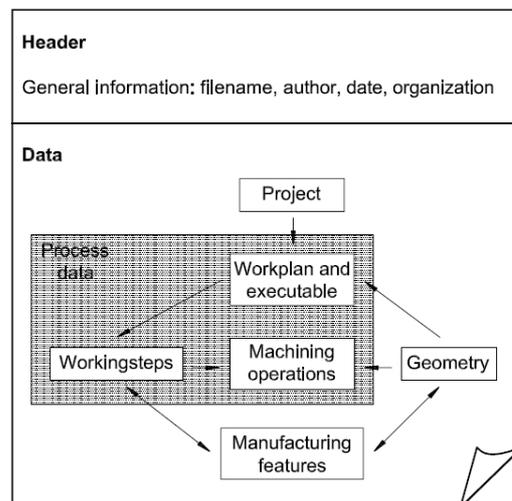


Figura 4.9: Estrutura de Dados STEP-NC. (ISO14649-1 2003)

```

ISO-10303-21;
HEADER;
FILE DESCRIPTION({'Nazareno Pacheco, Roberto Rosso, Eduardo Harbs ISO
14649-11 FURO5',
'PROGRAM WITH A ROUND_DRILL'}, '1');
FILE_NAME('FURO1.STP');

FILE_SCHEMA({'MACHINING_SCHEMA', 'MILLING_SCHEMA'});
ENDSEC;

DATA;
#1= PROJECT('EXECUTE FURO1', #2, (#4), $, $, $);
#2= WORKPLAN('MAIN WORKPLAN', (#11, #12, #13), $, #8, $);
#4= WORKPIECE('SIMPLE WORKPIECE', #6, 0.10, $, $, $, $, $, $);
#6= MATERIAL('ST-50', 'STEEL', (#7));
#7= PROPERTY_PARAMETER('E-20000N/M2');
#8= SETUP('SETUP1', #71, #62, (#9));
#9= WORKPIECE_SETUP(#4, #74, $, $, ());
#11= MACHINING_WORKINGSTEP('WS DRILL HOLE1', #62, #17, #20, $);
#12= MACHINING_WORKINGSTEP('WS DRILL HOLE2', #62, #18, #20, $);
#13= MACHINING_WORKINGSTEP('WS DRILL HOLE3', #62, #19, #20, $);
.....
#17= ROUND_HOLE('HOLE 1', #4, (#20, $), #81, #64, #58, $, #26);
#18= ROUND_HOLE('HOLE 2', #4, (#20, $), #82, #64, #58, $, #26);
#19= ROUND_HOLE('HOLE 3', #4, (#20, $), #83, #64, #58, $, #26);
.....
#20= DRILLING($, $, 'DRILL HOLE1', 10.000, $, #44, #45, #41, $, $, $, $, #46);
#26= THROUGH_BOTTOM_CONDITION();
#31= TWIST_DRILL(#32, 2, .RIGHT, .F, 0.840);
#32= MILLING_TOOL_DIMENSION(1.000, $, $, $, $, $);
#41= MILLING_MACHINE_FUNCTIONS(.T, $, $, .F, $, $, .T, $, $, ());
#44= MILLING_CUTTING_TOOL('SPIRAL DRILL D1MM', #31, (#126), 90.000, $, $);
#45= MILLING_TECHNOLOGY(0.030, .TCP, $, 16.000, $, .F, .F, .F, $);
#46= DRILLING_TYPE_STRATEGY(75.000, 50.000, 2.000, 50.000, 75.000, 8.000);
#56= PLUS_MINUS_VALUE(0.300, 0.300, 3);
#58= TOLERANCED_LENGTH_MEASURE(22.000, #56);
#62= ELEMENTARY_SURFACE('SECURITY PLANE', #73);
.....
#84= AXIS2_PLACEMENT_3D('HOLEDEPTH', #112, #113, #114);
#95= CARTESIAN_POINT('SETUP1: LOCATION',
(15.000, 9.000, 4.000));
#96= DIRECTION('
(0.000, 0.000, 1.000));
(0.000, 0.000, 1.000));
#127= CARTESIAN_POINT('HOLE4: LOCATION',
(8.000, 8.000, 0.000));
#128= DIRECTION('
(0.000, 0.000, 1.000));
(0.000, 0.000, 1.000));
#129= CARTESIAN_POINT('HOLE5: LOCATION',
(9.000, 6.000, 0.000));
#130= DIRECTION('
(0.000, 0.000, 1.000));
(0.000, 0.000, 1.000));

ENDSEC;
END-ISO-10303-21;

```

Figura 4.10: Exemplo resumido de programa STEP-NC.

O *Workplan and Executables* é parte da idéia central de como executar as tarefas. São 3 tipos de executáveis: *Workingstep*, *NC functions* e *program structure*. O *Workingstep* é geralmente uma descrição das operações realizadas em uma única operação utilizando uma única ferramenta (por exemplo, a operação de furação de um furo passante). *NC functions* são declarações de como exibir uma mensagem, uma parada opcional e outras operações que não precisam de interpolação de eixos. O *program structure* chamada *executable* é uma classe de entidades que são responsáveis pelo controle do fluxo do programa, tais como, *while*, *if*, *parallel*, e *Workplan*. O *Workplan* tem um papel especial no controle do fluxo do programa, uma vez que controla a seqüência dos *Workingsteps*. Isto implica que, se o usuário alterar a seqüência de *Workingsteps* no *Workplan*, é possível conseguir uma mudança na ordem das operações. Portanto, as definições restantes, como a geometria e a tecnologia continuam inalteradas. A descrição da tecnologia é uma definição dos *Workingsteps* encontrado no *Workplan*. Este inclui uma grande quantidade de dados como funções da máquina, dados de ferramenta e assim por diante.

Alguns dos benefícios com o uso do STEP-NC são (XU 2004):

- Fornece um modelo de dados completo e estruturado, ligando as informações geométricas e tecnológicas, assim nenhuma informação é perdida entre os estágios diferentes do processo de desenvolvimento de produtos;
- Seus elementos de dados são adequados para descrever a tarefa do NC;
- O modelo de dados é estendido para facilitar a tecnologia (com conformidade de classes) para combinar com as habilidades específicas do CAM, SFP ou NC;
- O tempo de usinagem para trabalhos pequenos e médios pode ser reduzido porque a otimização inteligente pode ser construída nos controladores STEP-NC;
- O mecanismo de pós-processamento é eliminado, pois a interface não exige informação de uma máquina específica;
- As máquinas-ferramenta são mais seguras e mais adaptáveis, porque o STEP-NC independe dos fabricantes das máquinas;
- As modificações no chão-de-fábrica podem ser realimentadas para o departamento de projeto, portanto teria o fluxo bi-direcional de informação entre os sistemas CAD/CAM ao CNC;
- O arquivo XML pode ser usado como um portador de informação, permitindo assim a fabricação distribuída com suporte da Internet.

Os controladores aderentes são geralmente classificados em três tipos, conforme ilustrado na Figura 4.12 (SUH 2003):

- O Tipo A simplesmente incorpora a STEP-NC para o controle convencional via pós-processador. Nesse caso, o CNC convencional pode ser usado sem modificação, isto não pode ser considerado como um CNC preparado para STEP-NC, mas deve ao menos ser capaz de ler código STEP-NC;
- O Tipo B, o “novo tipo de controle”, que tem um interpretador STEP-NC como parte do CNC, o *workingstep* programado é executado pelo *kernel* de CNC com capacidade de geração de trajetória da ferramenta. Este, é o tipo básico onde o

movimento é executado fielmente. É baseado na estratégia e seqüência de usinagem como especificado pelo STEP-NC. Em outras palavras, não tem outras funções inteligentes além da capacidade de geração da trajetória de ferramenta;

- O Tipo C, promete muito mais, que o antecessores, é o "novo controle inteligente", em que o CNC pode executar de maneira "inteligente" a tarefa de usinagem e "autonomamente" baseado na informação vinda do arquivo STEP-NC.

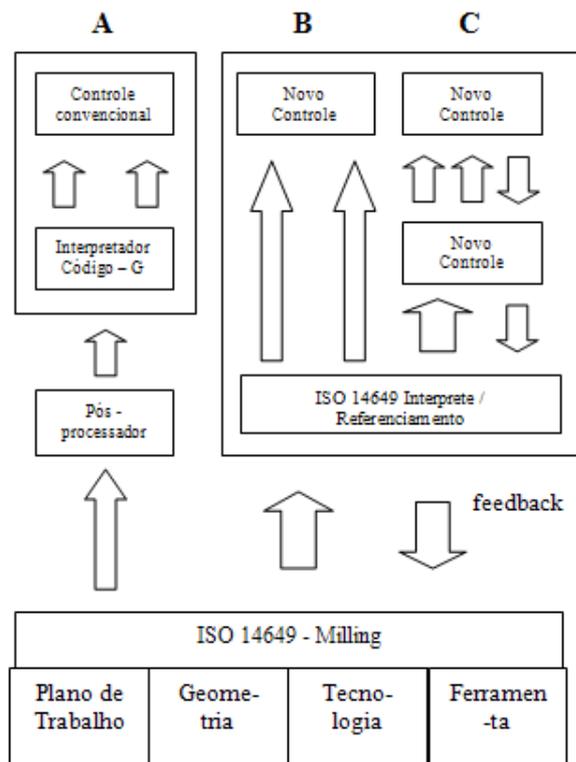


Figura 4.11: Tipos de controladores. (SUH 2003)

O uso de estruturas de dados aderentes a STEP-NC cria as bases para o uso de modelos de informação integrados e não sujeitos a erros de interpretação. Esta abordagem é especialmente apropriada para integração de sistemas CAD/CAPP/CAM/CNC utilizando modelos de dados de produto e de manufatura. O uso de dados aderentes a STEP desde o projeto até a usinagem da peça deverá diminuir sensivelmente o tempo de colocação do produto no mercado. Também deve-se conseguir uma queda de custos devido à menor circulação de papéis, menos erros de programação, melhor uso de recursos computacionais e de manufatura. Com o passar do tempo, mais empresas de *software* adotarão essa tecnologia, visando a melhoria da qualidade dos dados e uma troca de informações muito mais eficiente comparada com as normas existentes. Outro fator chave na utilização dos

padrões é a diminuição no tempo de criação e desenvolvimento de peças, gerando assim uma redução significativa nos custos de fabricação (ROSSO Jr. 2003), (NASSEHI 2006), (SUH 2003).

A ISO14649 é um ARM (*Application Reference Model*), enquanto a ISO10303-238 é um AP que implementa o ARM num contexto STEP, que permite uma melhor integração com outros Protocolos de Aplicação da ISO10303. Entretanto, o AP-238 é mais complexo e implica em maiores dificuldades na implementação que o ISO14649. Por este motivo, neste trabalho optou-se por realizar as implementações utilizando A ISO14649.

Nesse capítulo abordaram-se as Normas ISO10303 ou STEP e ISO14649 ou STEP-NC, foram discutidas as suas estruturas para a troca de informação para que haja uma integração da manufatura com uma perda mínima de informação entre os diferentes sistemas que a compõem. No próximo capítulo será explorada uma tendência que é o controle numérico de arquitetura aberta e também suas principais arquiteturas.

## 5 Controle Numérico de Arquitetura Aberta

Controle Numérico de Arquitetura Aberta é um sistema de controle CNC baseado em *software* e *hardware* não-proprietário. Foi concebido para eliminar a dependência dos usuários finais com relação ao *hardware* proprietário, incluindo placas de controle de movimentação para microcomputadores (MOREIRA 1998).

Poucos CNC's tomam esta abordagem de proporcionar uma menor dependência entre seus *hardware* e *software* num processo de manufatura. CNC's de arquitetura aberta consistem em aplicações que não estão dependentes de nenhum *hardware* específico e que podem funcionar em qualquer equipamento que o usuário escolha. A evolução desta abordagem é análoga ao percurso executado pelas ferramentas CAD, desde o início da década de 1980, quando a *Autodesk* começou a oferecer *software* de projeto que podia ser executado em qualquer tipo de PC, em oposição a terminais proprietários dedicados (MITSUSHI 1997). Ao escolher o *software* independente do *hardware* para CNC, o usuário pode escolher onde instalar, quem instala, treina e dá suporte. As empresas podem assim requisitar os serviços de uma terceira entidade, um integrador de sistemas, por exemplo. Podem instalar elas próprias o *software* e colocar o PC em rede. Com o CNC baseado em aplicações computacionais, a empresa pode adicionar sucessivamente novo *hardware* e *software* através de atualizações incrementais para melhorar a produtividade da máquina-ferramenta. Isto significa que a empresa pode proteger os seus investimentos em máquinas-ferramenta, formação dos colaboradores e dados do espaço fabril.

O computador pessoal é uma plataforma de custo reduzido para suporte à manufatura integrada. Uma visão de (MOREIRA 1998) já na época era que as máquinas-ferramenta deixariam de ser ilhas isoladas de automação, passando a ser dispositivos periféricos, controlados em tempo real por microcomputadores a partir de uma rede de longa distância que interliga as unidades fabris da empresa. Aplicações da computação, em oposição a *hardware* específico, controlariam as máquinas e a produção, de forma similar ao uso dos microcomputadores nas restantes funções de negócio da empresa (MOREIRA 1998).

O conceito de arquitetura aberta possibilita a integração física e funcional do equipamento produtivo com sistemas computacionais. Esta forma de pensar é oposta a dos

sistemas de controle baseados em *hardware* proprietário que não possuem abertura para possibilitar a integração completa dos sistemas. A MDSI (*Manufacturing Data Systems Inc.*) concebeu um produto para concretizar esta visão e denominou como OpenCNC (MDSI 2008). A Figura 5.1 ilustra o conceito de arquitetura aberta.

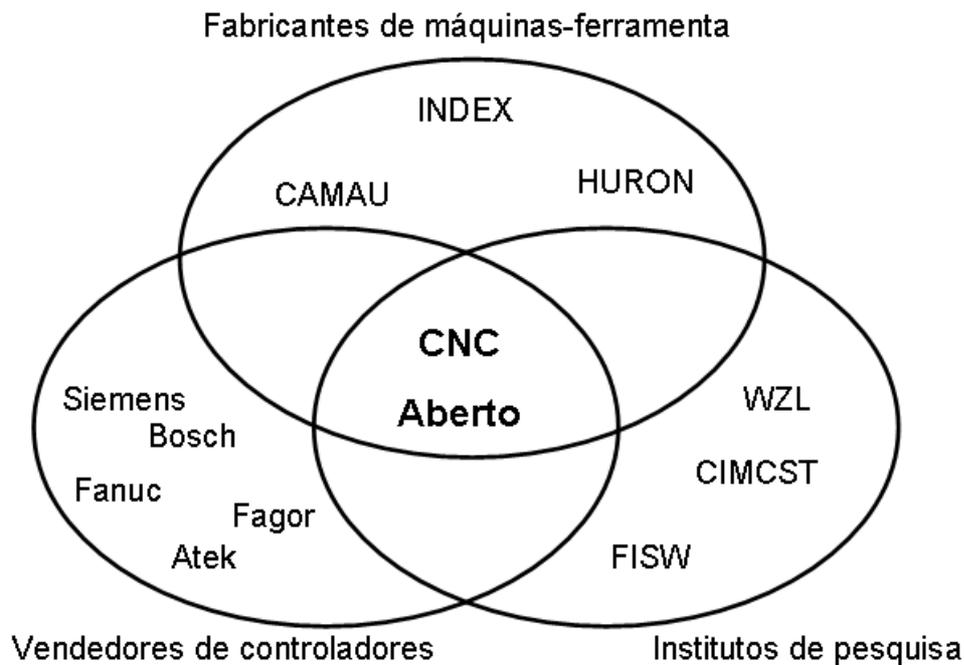


Figura 5.1: Conceito de CNC aberto. Modificado de (OSACA 2008).

O OpenCNC está sendo usado atualmente na produção em vários tipos de máquinas-ferramenta, incluindo centros de usinagem de dois a cinco eixos. As empresas estão demonstrando um interesse no OpenCNC, quer para a atualização de máquinas-ferramenta antigas, quer na implementação em máquinas novas. Muitas empresas possuem máquinas-ferramenta que ainda produzem peças boas, possuindo no entanto unidades de controle obsoletas. A implementação do OpenCNC nestas máquinas permite as empresas prolongar o seu tempo de vida útil (MDSI 2008).

Os equipamentos de controle, além das suas funções de controle das máquinas-ferramenta, possuem um potencial para aquisição e tratamento de informação do processo, como tempos de ciclo, velocidades de avanço, tempos de *setup*, etc. As empresas precisam dessas informações em tempo real, sem intervenção dos operadores, se quiserem melhorar continuamente os seus processos da qualidade, redução de custos e redução de tempo de projeto/produção dos produtos. Com a filosofia de arquitetura aberta esses equipamentos

de controle poderão ser de fabricantes diferentes, deixando assim a planta fabril flexível.

## 5.1 Principais arquiteturas abertas

O controlador de arquitetura aberta é uma tendência importante da tecnologia moderna. Muitas nações e organizações de pesquisa formaram equipes para desenvolvimento dessas arquiteturas e muitos métodos foram propostos (YONGLIN 2005), (NIST 2008), (EROL N. A. e ALTINTAS 2000).

- A plataforma OSACA (*Open System Architecture for Control with Automation System*) foi desenvolvida na Europa. A Figura 5.2 ilustra a arquitetura OSACA;
- A plataforma OSEC (*Open System Environment for Controller*) foi definida no Japão. A Figura 5.3 ilustra a arquitetura OSEC;
- Chrysler, Ford e General Motors fundaram o OMAC (*Open Modular Architecture Controls*). A Figura 5.4 ilustra a arquitetura OMAC. Baseado na OMAC;
- O NIST desenvolveu o EMC (*Enhanced Machine Controller*) que é um *software* que implementa em tempo real, o controle de equipamentos, tais como máquinas-ferramenta, robôs e máquinas de medição. Ele funciona em tempo real no Linux com o *RTlinux* ou *RTAI patch*. Esta arquitetura fornece um *software* também para CLP, e é ilustrado na Figura 5.6

Estes são sistemas de *software/hardware* que podem facilitar o uso de tecnologia não proprietária. Tais sistemas, por exemplo CNC MDSI (*Manufacturing Data Systems Inc.*) têm uma arquitetura que oferece muitas oportunidades para implementação de controle e também o reequipamento dos CNC's mais antigos. (MDSI 2008).

Na Figura 5.5 tem-se a arquitetura MDSI, que é uma arquitetura comercial. A MDSI é um fornecedor mundial de CNC de arquitetura aberta para controle de movimento, além de soluções de automatização de fábricas. Um exemplo de CNC baseado em *software* foi implantado pela *Great Lakes Industries* (GLI), que em 1993 decidiu usar o OpenCNC, tecnologia da MDSI, onde surgiu a primeira arquitetura aberta comercial propriamente dita, o *software* do CNC não exige um *hardware* proprietário (MDSI 2008).

As várias arquiteturas abertas de controladores, tais como: OSACA, OMAC, OSEC, EMC e MDSI, pode-se tirar as seguintes conclusões (YONGLIN 2005), (NIST 2008):

- OSACA e OMAC são semelhantes. Ambos provêm métodos eficazes para a configuração do sistema. Além disso, estas duas arquiteturas não definem a interação entre os componentes e os detalhes da plataforma. O OMAC descreve as funções dos componentes com mais detalhes do que OSACA;
- OSEC é uma arquitetura difícil de avaliar sistematicamente pelo fato de ainda possuir um grande número de dimensões indefinidas;
- MDSI é uma arquitetura comercial, tem toda uma estrutura para a integração desde o alto nível, como o *parser* para o código NC, planejamento de trajetória, planejamento da peça, até o nível mais baixo que é a máquina;
- EMC é uma arquitetura que funciona em Linux e controla em tempo real equipamentos, tais como: máquinas-ferramenta, robôs e máquinas de medição.

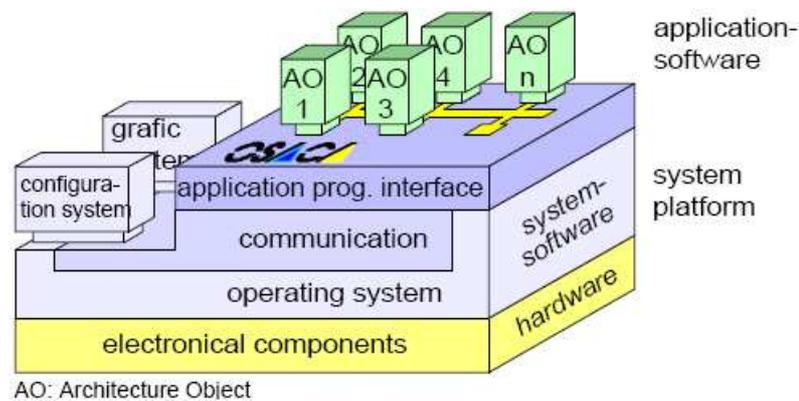


Figura 5.2: Arquitetura OSACA. (OSACA 2008)

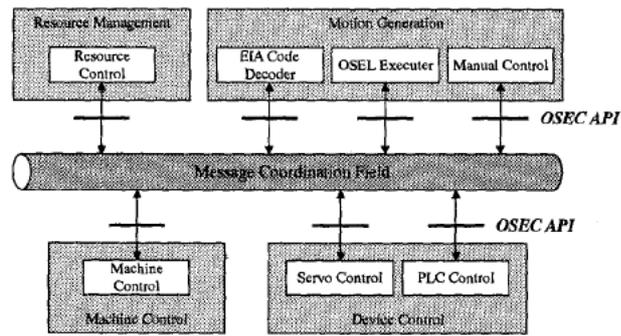


Figura 5.3: Arquitetura aberta OSEC. (SAWADA 1997)

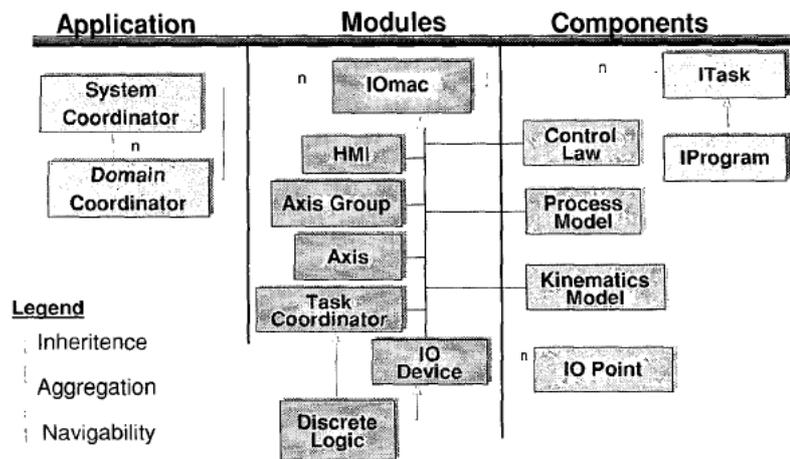


Figura 5.4: Arquitetura OMAC. (KOLLA 2002)

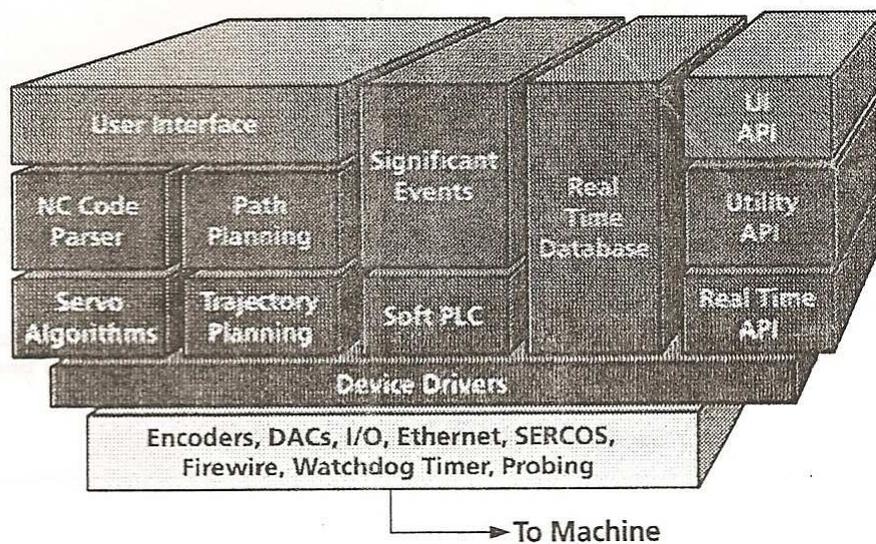


Figura 5.5: Arquitetura aberta MDSI. (MDSI 2008)

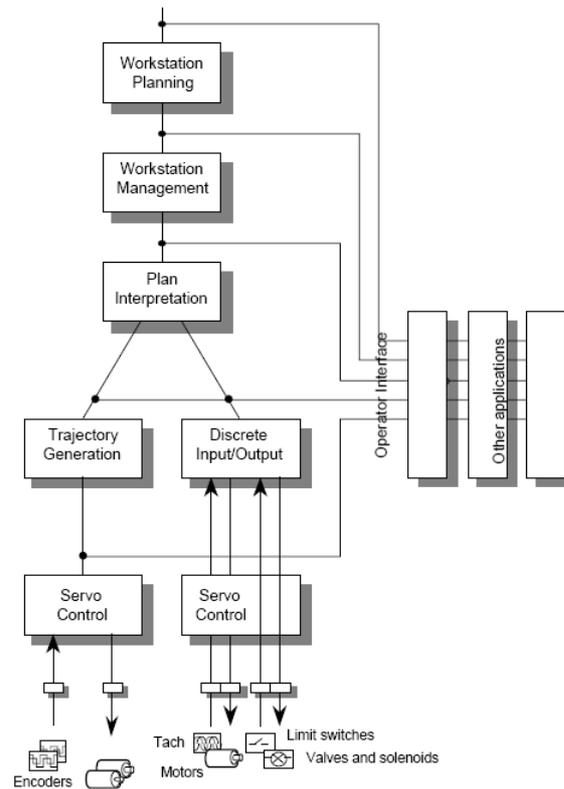


Figura 5.6: Arquitetura EMC. (NIST 2008)

Como visto no capítulo as arquiteturas abertas são uma tendência, que vão favorecer o cliente, pois eliminarão a dependência a um determinado fabricante, facilitando assim o uso da tecnologia não proprietária.

Poucos fabricantes seguem esse caminho, mas o usuário final está buscando essa independência na escolha dos equipamentos, forçando o fabricante a aderir a essa tendência, visto que os projetos de arquiteturas abertas têm a participação efetiva das empresas.

Sobre controle numérico de arquitetura aberta, apesar dessa idéia apontar para o uso da tecnologia não proprietária, dada a complexidade de algumas estruturas eles se tornaram difíceis de implementar. Desta forma neste trabalho foi utilizado o conceito de arquitetura aberta para o desenvolvimento de uma estrutura mais simples que será apresentada no capítulo 6.

## 6 Desenvolvimento de uma máquina de arquitetura aberta aderente a STEP-NC

Este capítulo apresenta uma arquitetura aberta de CNC baseada em STEP-NC desenvolvida durante este trabalho. Os aspectos de *hardware* e *software* desenvolvidos para validação da arquitetura são descritos ao longo do capítulo.

Na estrutura mecânica foi realizado um trabalho de reengenharia, pois de um *scanner* e duas impressoras montou-se o protótipo. Usou-se a estrutura do *scanner* para construir o eixo X da máquina e das impressoras aproveitaram-se os locais onde vão os cartuchos de tinta para fazer os eixos Y e Z. Na Figura 6.1 é mostrada a máquina desenvolvida.

Na parte eletroeletrônica foram confeccionadas placas eletrônicas para o controle e acionamento dos eixos e da ferramenta de corte do "Frank", como é explicado com mais detalhes na seção 6.2.

Na área da computação foi desenvolvido um compilador em JAVA aderente a STEP-NC, e uma interface em C++ para fazer a comunicação entre o PC e a máquina (PC-CNC). Na seção 6.3.2 a parte de *software* deste trabalho é explicada com mais detalhes.

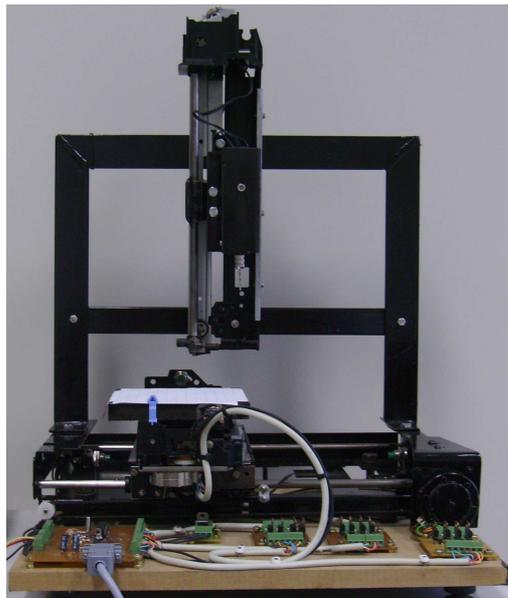


Figura 6.1: Furadeira/Fresadora CNC aderente a STEP-NC.

## 6.1 Arquitetura aberta baseada em STEP-NC

Neste trabalho foi desenvolvida uma arquitetura baseada na idéia do CNC aberto. Os blocos que formam essa arquitetura são:

1. **Compilador**, que é responsável pelo tratamento do arquivo STEP-NC;
2. **Tabela**, que organiza as informações de usinagem, tais como: tipos de ferramentas, dimensões dos furos, posição dos furos;
3. **Microcontrolador**, recebe informação de usinagem, e envio de comandos para o acionamento dos *drivers* dos motores da máquina-ferramenta;
4. **Comunicação serial, *drivers* dos motores de deslocamento e acionamento do motor do eixo de usinagem**, são responsáveis respectivamente pela troca de informação entre PC e placa de controle, acionamento dos motores dos eixos da máquina e o acionamento do eixo árvore da máquina-ferramenta.

O projeto foi dividido em dois grandes blocos, o *software* e o *hardware*. A arquitetura desenvolvida no projeto é mostrada na Figura 6.2. Essa arquitetura é aderente ao conceito de CNC aberto e a norma STEP-NC. A informação é introduzida no sistema sob a forma de arquivo STEP-NC que é compilado e gera um conjunto mínimo de informações necessárias ao micro-controlador para acionar e controlar uma máquina-ferramenta. Esta informação entra no micro-controlador na forma de uma tabela, conforme mostra o bloco na figura 6.2. A partir deste ponto o controle está a cargo do micro-controlador e independe de que máquina esteja sendo controlada, desde que esta tenha a configuração mecânica para qual o *software* de controle foi desenvolvido.

## 6.2 *Hardware* desenvolvido

No *hardware*, foi desenvolvida uma placa controladora, *driver* dos motores de passo e *driver* para o acionamento da ferramenta. Como explicado nas seções 6.2.1, 6.2.2, 6.2.3, 6.2.4.

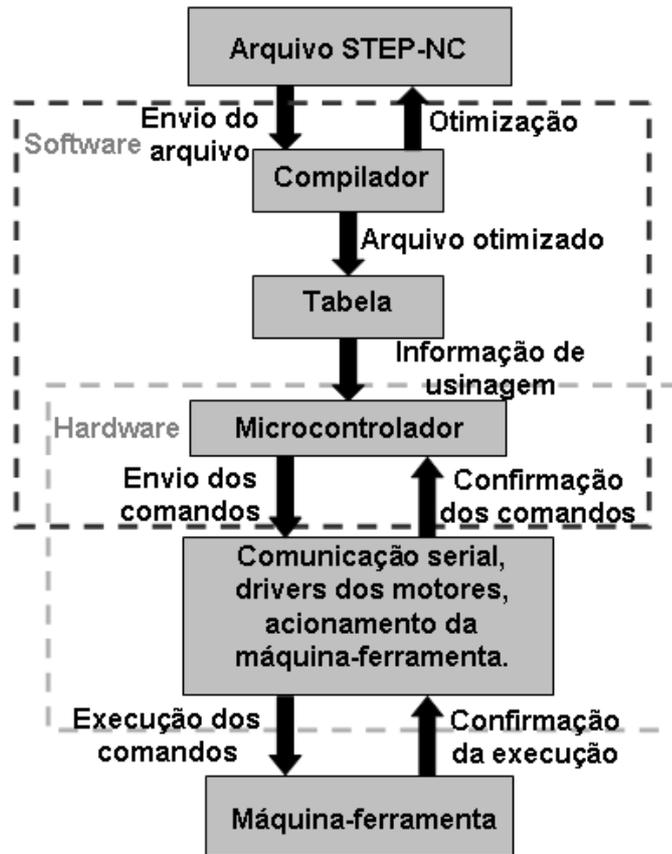


Figura 6.2: Arquitetura Aberta desenvolvida neste trabalho

### 6.2.1 Controladora

A placa de controle foi desenvolvida com o microcontrolador PIC18F252 da *Microchip* (MICROCHIP 2008). Ela é responsável pela comunicação via RS-232 entre o PC e/ ou CNC, pela movimentação dos eixos da máquina CNC e acionamento da ferramenta de usinagem. A movimentação dos eixos é comandada via PC, pois ao implementar toda a parte de cálculo, a relação entre as distâncias a percorrer e o valor de passo para cada milímetro fez com que o microcontrolador trabalhasse numa função que ultrapassava sua capacidade, onde teria que ser feito no código interrupções para um melhor desempenho. Assim decidiu-se fazer todo o processamento no PC e enviar somente a quantidade de passos referentes à posição desejada. Na Figura 6.3 é mostrada a placa de controle desenvolvida.

A Figura 6.4 mostra o esquema elétrico da placa de controle, onde se tem os componentes responsáveis pela comunicação serial entre o microcontrolador e o microcomputador um desses componentes é o MAX232, que implementa a interface de comunicação RS-232.



### 6.2.2 Drivers dos acionamentos dos eixos coordenados

Os *drivers* dos eixos são responsáveis pelos acionamentos dos motores para deslocamento dos eixos X, Y e Z. Para o desenvolvimento desses *drivers* foram utilizados transistores TIP 122, diodos FR107 para fazer a função chamada de "roda-livre" e os resistores para a polarização dos TIP 122. A Figura 6.5 mostra os *drivers* dos eixos e a Figura 6.6 mostra o esquema elétrico.

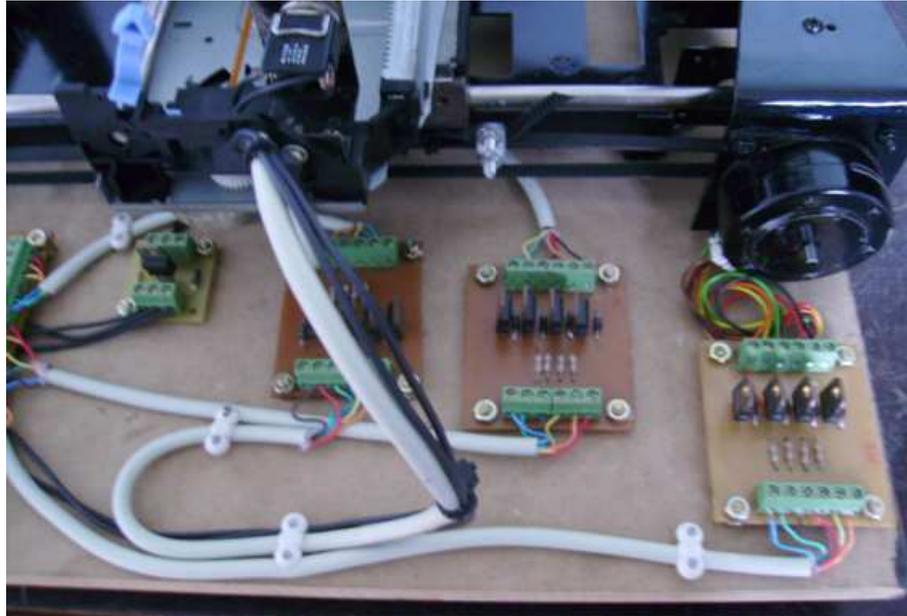


Figura 6.5: *Drivers* dos eixos.

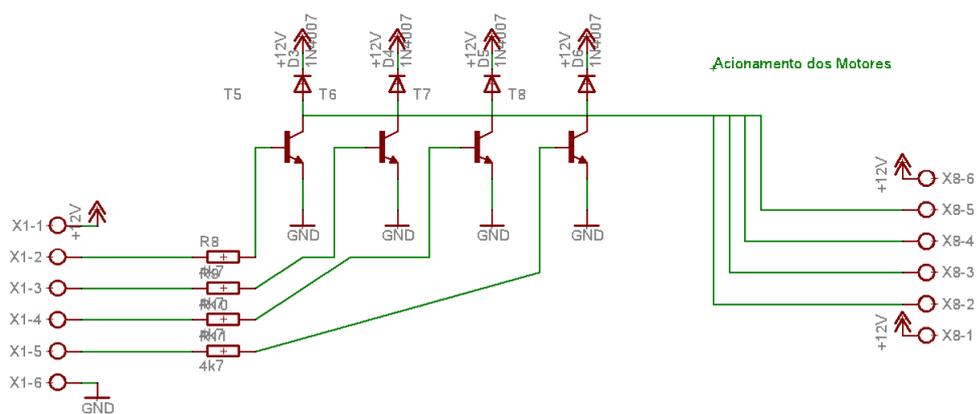


Figura 6.6: Esquema elétrico dos *drivers* dos motores

### 6.2.3 *Driver* do acionamento do eixo árvore

O *driver* de acionamento do eixo árvore é responsável pelo funcionamento da ferramenta de usinagem. Da mesma forma que nos *drivers* dos eixos X, Y e Z, para o desenvolvimento desse *driver* foi utilizado o transistor TIP 122, o diodo FR107 para fazer a função chamada de "roda-livre" e o resistor para a polarização do TIP 122. A Figura 6.7 mostra o *driver* para comando da ferramenta de usinagem. A Figura 6.8 mostra o esquema elétrico. O controle de velocidade de rotação da ferramenta de corte é feito através de modulação por largura de pulso (PWM). Quando aumenta a largura do pulso, diminui a velocidade de rotação da ferramenta de corte e quando diminui a largura do pulso aumenta a velocidade de rotação da ferramenta de corte.



Figura 6.7: Driver para acionamento da ferramenta

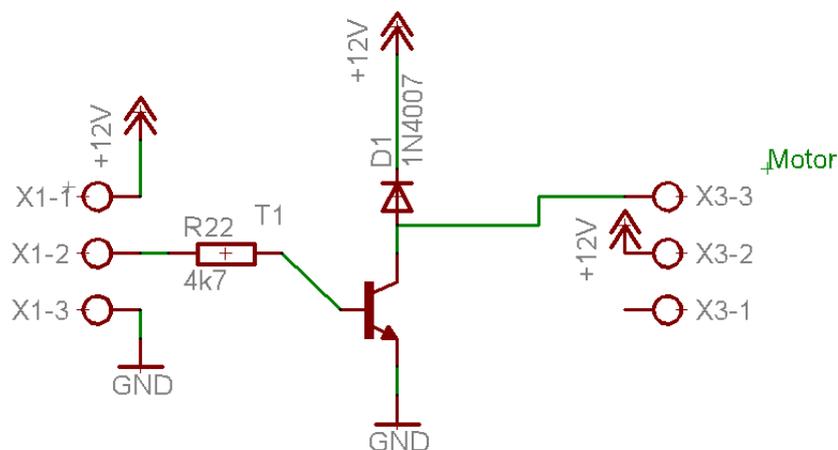


Figura 6.8: Esquema elétrico do driver da ferramenta

### 6.2.4 Motor de passo

Os motores de passo são dispositivos mecânicos eletro-magnéticos que podem ser controlados digitalmente através de um *hardware* específico ou através de *software*.

Para o acionamento dos eixos X, Y e Z foram usados motores de passos do *scanner* e das impressoras, o seu tipo de controle será abordado a seguir. Na Figura 6.9 é mostrado um motor de passo unipolar igual ao utilizado neste trabalho. Esses motores têm como características: 48 passos, sendo que cada passo equivale a  $7,5^{\circ}$ , unipolar, alimentação de 24V. Existem vários modos de operação de um motor de passo (MESSIAS 2008):

- Passo completo 1 (*Full-step*): Somente uma bobina é energizada a cada passo, isto confere menor torque, pouco consumo de energia e maior velocidade;
- Passo completo 2 (*Full-step*): Duas bobinas são energizadas a cada passo, isto confere maior torque, porém consome mais energia que o Passo completo 1, com maior velocidade que o Passo Completo 1;
- Meio Passo (*Half-step*): A combinação do passo completo 1 e do passo completo 2 gera um efeito de meio passo, que consome mais energia que os passos anteriores, é muito mais preciso que os passos anteriores, o torque é próximo ao do passo completo 2 e a velocidade é menor que as dos passos anteriores.

Através dessas características de cada modo de operação decidiu-se fazer o acionamento dos motores com o recurso de meio passo por ele ser mais preciso e ter um torque próximo ao passo completo 2. A Tabela 6.1 mostra as seqüências de pulsos digitais corretas utilizadas para controlar os motores, cada linha da tabela equivale a um passo. Como os motores têm as mesmas características de acionamento, é usada uma única tabela.

Para controlar a velocidade e o posicionamento do motor de passo é enviada uma seqüência de pulsos digitais, num determinado intervalo de tempo. Quanto menor esse intervalo de tempo, maior é a velocidade que o motor irá girar. Esse controle é feito em malha aberta.

Para mudar a direção de rotação do motor, é invertida a seqüência dos pulsos.

Tabela 6.1: Meio passo (*Half-step*)

1	0	0	0
1	1	0	0
0	1	0	0
0	1	1	0
0	0	1	0
0	0	1	1
0	0	0	1
1	0	0	1



Figura 6.9: Motor de passo utilizado no presente trabalho

## 6.3 *Software* desenvolvido

Na parte do trabalho referente ao *software*, foi desenvolvida uma interface de dados aderente a STEP-NC, através de um compilador que faz o tratamento necessário das informações de usinagem contidas no arquivo STEP-NC, conforme descrito nos itens de 1 a 6 na seção 6.3.1 e na Figura 6.11.

### 6.3.1 Compilador

Um compilador é um programa que lê um programa escrito numa linguagem - a linguagem *fonte* - e o traduz num programa equivalente numa outra linguagem - a linguagem *alvo* (AHO 1995), como mostrado na Figura 6.10. Como importante parte desse pro-

cesso de tradução, o compilador relata ao seu usuário a presença de erros no programa fonte (GRUNE 2001)

Existem milhares de linguagens fonte, tais como: Java, C e as linguagens especializadas que emergem virtualmente em quase todas as áreas de aplicação de computadores. As linguagens alvo são igualmente variadas; uma linguagem alvo pode ser uma outra linguagem de programação ou a linguagem de máquina de qualquer elemento entre microprocessador e supercomputador. Apesar dessa aparente complexidade, as tarefas básicas que qualquer compilador precisa executar são essencialmente as mesmas. Pelo entendimento delas, pode-se construir compiladores para uma ampla variedade de linguagens fonte e máquinas alvo, usando as mesmas técnicas básicas (AHO 1995), (GRUNE 2001).



Figura 6.10: Funcionamento básico de um compilador. (AHO 1995)

Análise léxica é o processo de analisar a entrada de linhas de caracteres (tal como o código-fonte de um programa de computador) e produzir uma seqüência de símbolos chamado "símbolos léxicos" (*lexical tokens*), ou somente "símbolos" (*tokens*), que podem ser manipulados mais facilmente por um *parser* (AHO 1995).

A análise léxica é uma forma de verificar um determinado alfabeto. Quando analisa-se uma palavra, pode-se definir através da análise léxica se existe ou não algum caracter que não faz parte do alfabeto, ou um alfabeto inventado (AHO 1995), (GRUNE 2001).

A análise sintática transforma um texto da entrada em uma estrutura de dados, em geral uma árvore, o que é conveniente para processamento posterior, e captura a hierarquia implícita desta entrada. Em ciência da computação e linguística, análise sintática (também conhecida pelo termo em inglês *parsing*) é o processo de analisar uma seqüência de entrada (lida de um arquivo de computador ou do teclado, por exemplo) para determinar sua estrutura gramatical segundo uma determinada gramática formal. Essa análise faz parte de um compilador, junto com a análise léxica e análise semântica

(GRUNE 2001), (AHO 1995). Através da análise léxica é obtido um grupo de símbolos, para que o analisador sintático use um conjunto de regras para construir uma árvore sintática da estrutura (AHO 1995), (GRUNE 2001).

A análise semântica é a verificação de tipos. Nela o compilador verifica se cada operador recebe os operandos que são permitidos pela especificação da linguagem fonte.

Na Figura 6.11 é ilustrada a estrutura do compilador desenvolvido neste trabalho, cujos blocos são descritos a seguir:

1. **Analisador léxico**, responsável pela verificação do alfabeto e lexemas;
2. **Analisador sintático**, responsável pela transformação do texto de entrada em uma estrutura de dados e verificação de conectude;
3. **Analisador semântico**, responsável pela verificação de tipos, ou seja, se cada operador recebe os operandos que são determinados pela especificação da linguagem fonte;
4. **Armazenamento de informações**, responsável por conter informações sobre *features*, nome do arquivo de entrada, entre outros parâmetros;
5. **Análise e armazenamento**, responsável pelas informações de cada *feature*; no caso implementou-se apenas a *feature* furo, e então tem-se os atributos diâmetro, posição, ferramenta;
6. **Tabela**, que organiza as informações de usinagem, tais como: tipos de ferramentas, dimensões dos furos, posição dos furos, para posterior envio à placa controladora.

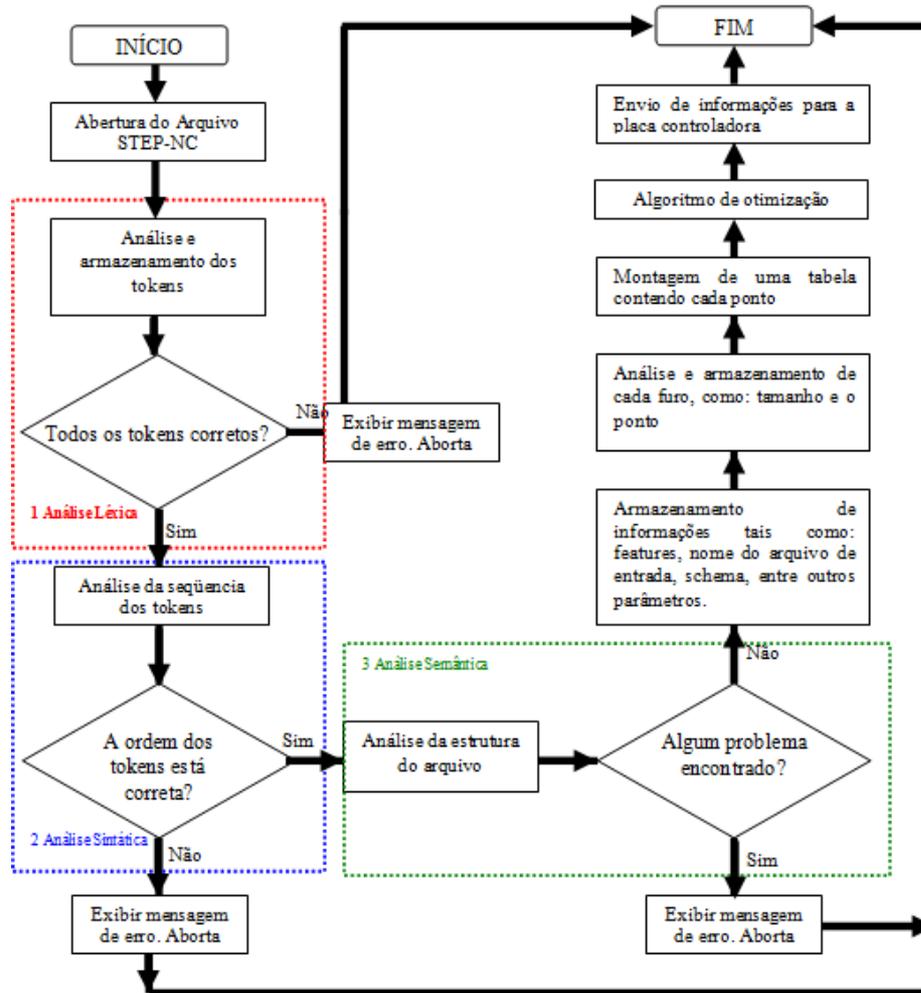


Figura 6.11: Estrutura do compilador desenvolvido

### 6.3.2 Interfaces

O compilador foi desenvolvido em plataforma JAVA. Tem-se também nessa parte um algoritmo de otimização de trajetória, que tem como objetivo traçar o melhor caminho, ou seja, a melhor rota para a usinagem da peça. Este algoritmo foi desenvolvido em PROLOG no contexto da disciplina de Inteligência Artificial Aplicada e depois foi adaptado para JAVA para integrar a interface STEP-NC.

Na Figura 6.12 é ilustrado o funcionamento da interface do compilador, onde é feita a verificação do arquivo STEP-NC e retorna uma resposta referente ao arquivo, no caso do exemplo o arquivo está correto.

Quando o arquivo está correto é gerada uma tabela dos dados de usinagem para posterior processamento no microcontrolador, como é mostrado na Figura 6.13.

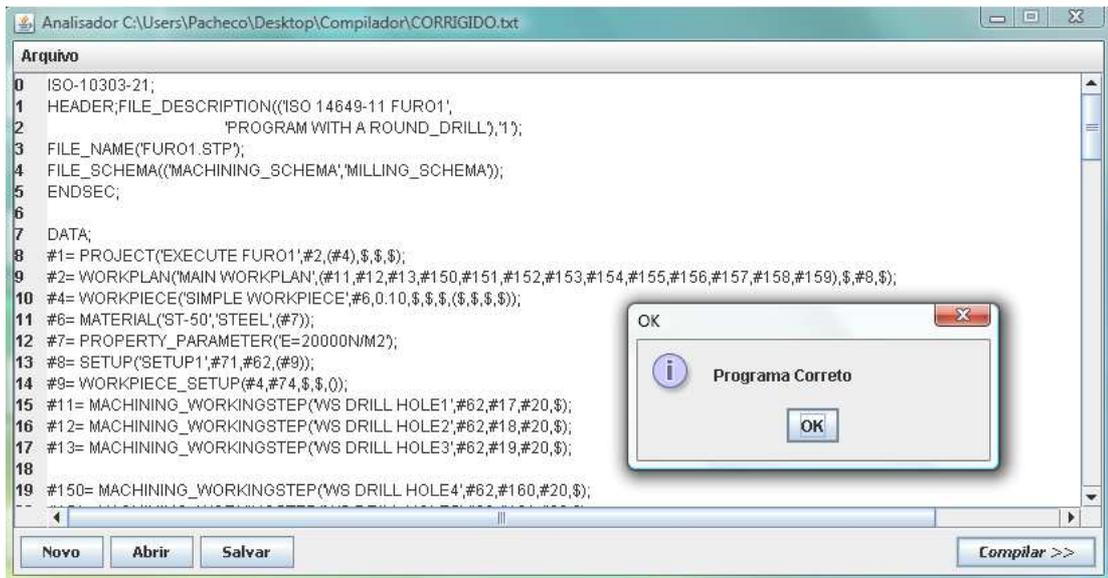


Figura 6.12: Compilador indicando que o arquivo está correto

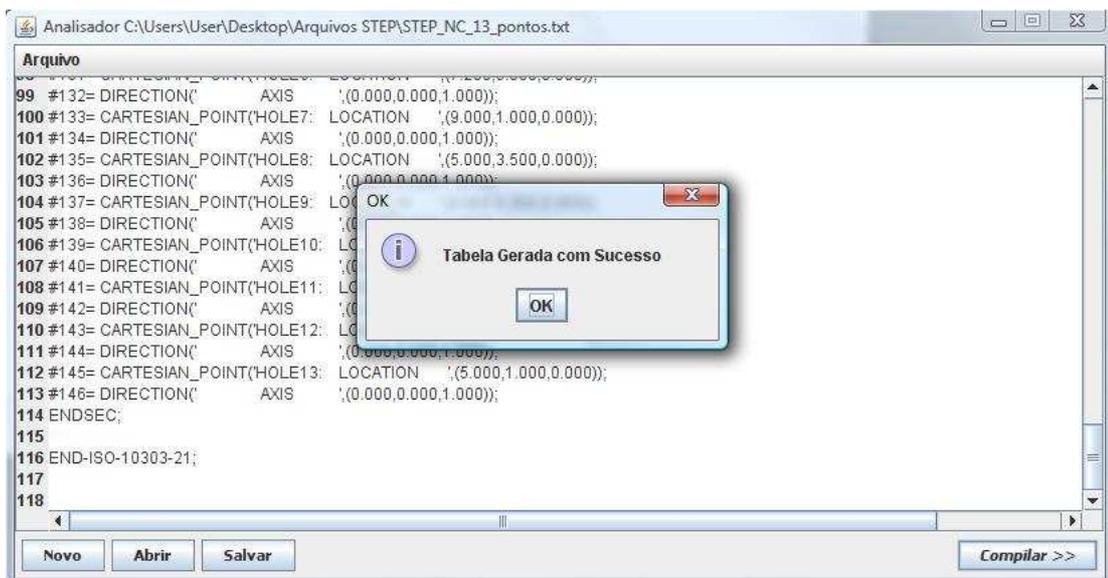


Figura 6.13: Aviso de que a tabela com os dados de usinagem foi gerada

Na Figura 6.14 é ilustrada uma resposta referente a um arquivo STEP-NC, com erro. Neste caso o compilador mostra um aviso informando a linha e coluna onde encontra-se o erro. O usuário também pode fazer as alterações diretamente na interface do compilador e salvar estas alterações no arquivo STEP-NC.

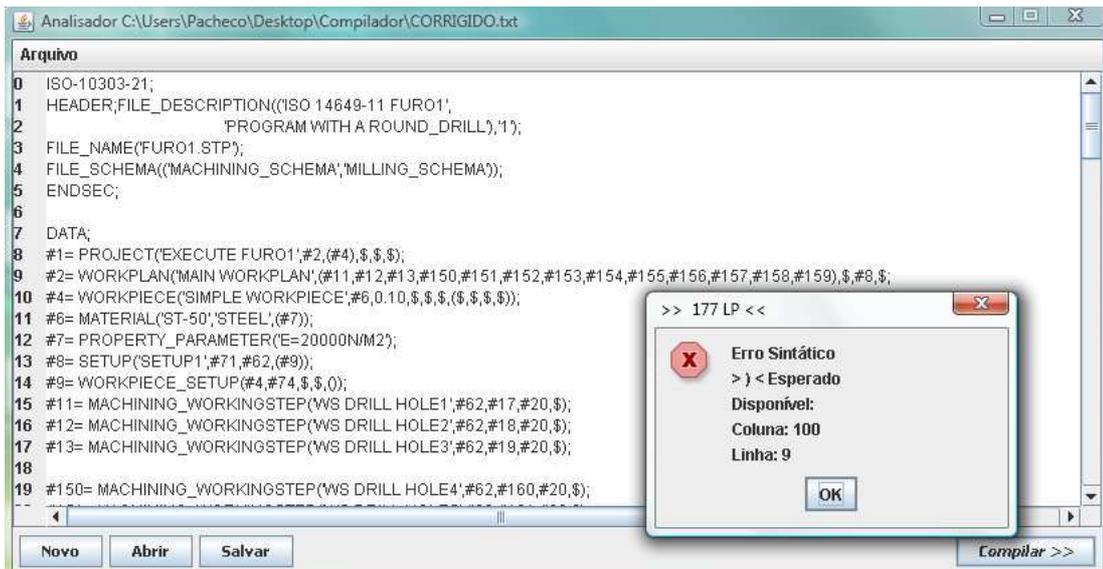


Figura 6.14: Compilador indicando que o arquivo está incorreto

Para a comunicação PC/CNC foi desenvolvida uma interface em C++. Essa interface é responsável pela comunicação e configurações da máquina, tais como: velocidade dos eixos, área de trabalho, posicionamento do zero da máquina.

A Figura 6.15 ilustra que a furadeira/fresadora CNC foi para a *Home position*, e está apta para executar a usinagem.

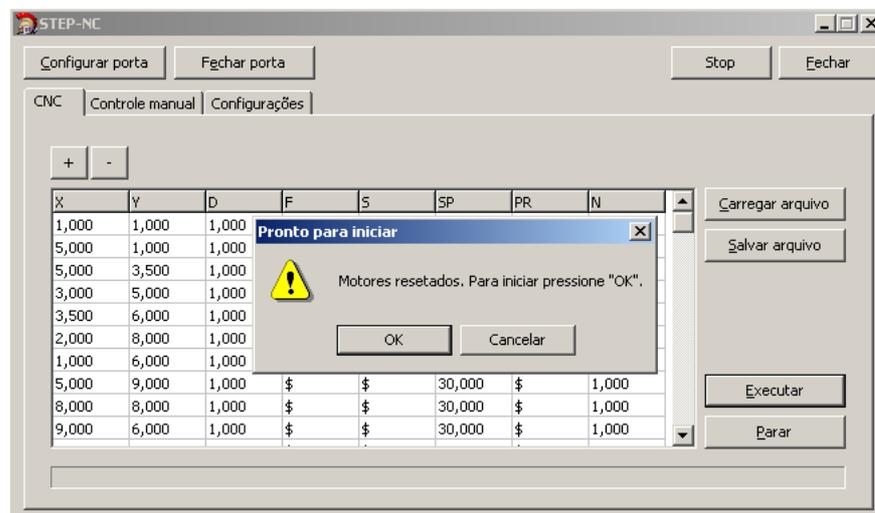


Figura 6.15: Indicação da *Home position*

A Figura 6.16 ilustra a conclusão dos trabalhos de usinagem.

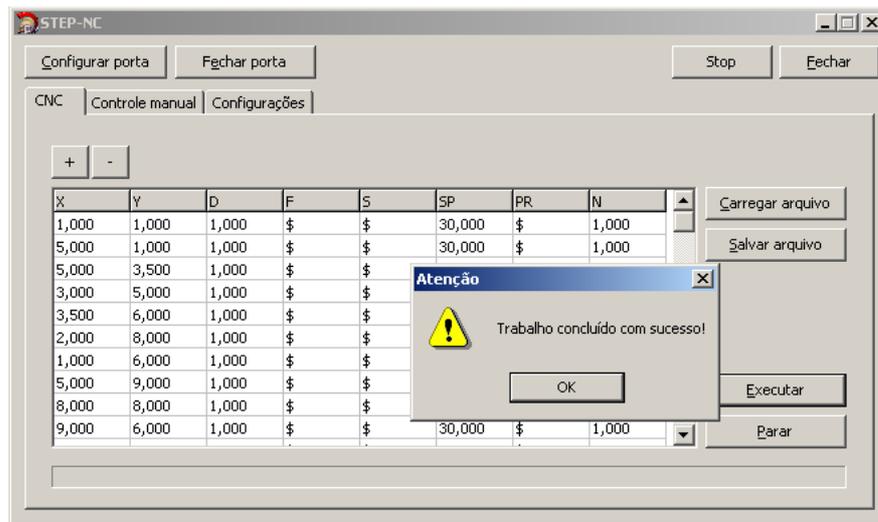


Figura 6.16: Indicação da Conclusão do trabalho

Na Figura 6.17 mostra-se a tela de configuração da máquina. Nessa tela pode-se configurar a área de trabalho, a velocidade de cada eixo com sua perspectiva rampa de aceleração, e a precisão dos passos nos eixos X, Y e Z e na aproximação em Z da ferramenta de corte. Por exemplo, quando é configurada uma área de trabalho em 130x130mm, que é o caso do protótipo desenvolvido, se o arquivo em STEP-NC tiver algum ponto fora dessa área de trabalho configurada é dado um aviso de erro.

### 6.3.3 Otimizador de trajetória

Esse otimizador analisa a tabela gerada pelo compilador e através das regras mostradas abaixo define uma trajetória otimizada. Ao definir uma melhor trajetória, será gerada uma nova tabela com as novas seqüências dos furos para usinagem. Havendo essa mudança na ordem das furações é enviado ao compilador um comando para a alteração do arquivo STEP-NC, na linha do *Workplan*, que é responsável pelo plano de trabalho da referida peça, como mostrada na Figura 6.18. Esta otimização foi feita apenas para furação, pois o protótipo, apesar de poder fazer fresamento, não teve implementado nenhuma *feature* que use este tipo de operação.

Para a melhoria do processo de usinagem, visando uma melhor trajetória, foi desenvolvido um algoritmo de otimização com duas regras:

- Sempre fazer o primeiro furo no ponto mais perto da *home position* da máquina;

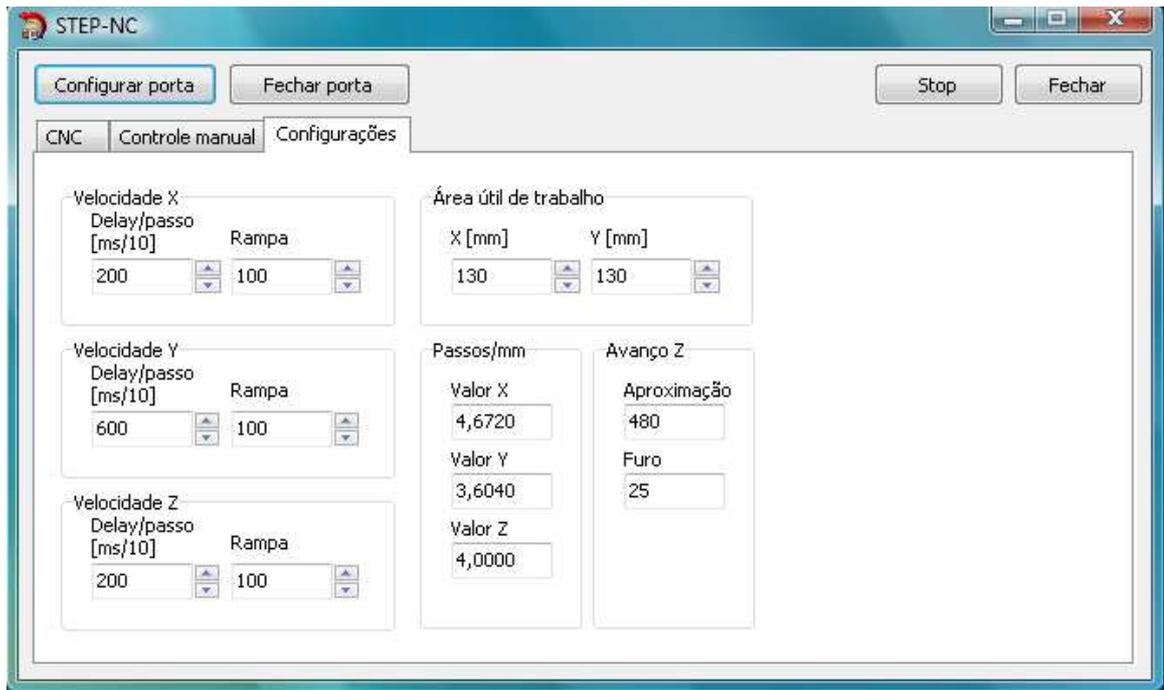


Figura 6.17: Tela de configuração

- Os furos subsequentes terão como referência a furação anterior, ou seja, uma posição relativa.

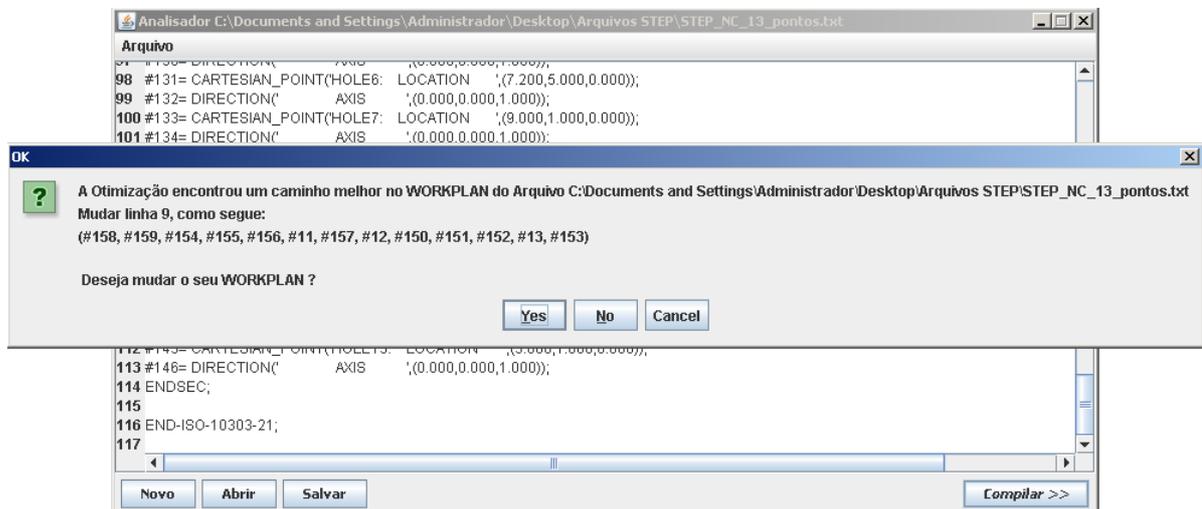


Figura 6.18: Usinagem sem otimizador X Usinagem com otimizador

## 6.4 Testes de Validação

Na realização dos testes a partir de arquivos em STEP-NC, o compilador faz toda a

análise léxica, sintática e semântica. Caso encontre algum discordância ele indica a linha e a coluna onde encontra-se o erro, caso o arquivo esteja correto, ele gera uma tabela com os dados de usinagem, tais como: posição dos furos, diâmetro dos furos, tipos de ferramentas, velocidade de avanço nos eixos, plano de segurança e velocidade da ferramenta.

Para verificar o funcionamento do *hardware* e *software* desenvolvidos foram realizados testes no Laboratório de Pesquisa em Automação de Sistemas - LAPAS, na Universidade do Estado de Santa Catarina.

Para os testes foram feitos arquivos contendo com cinco, treze, trinta e seis, cinqüenta e cem furos.

A Figura 6.19, mostra o esboço da peça com cinco furos e na Figura 6.20 ilustra a peça depois da usinagem.

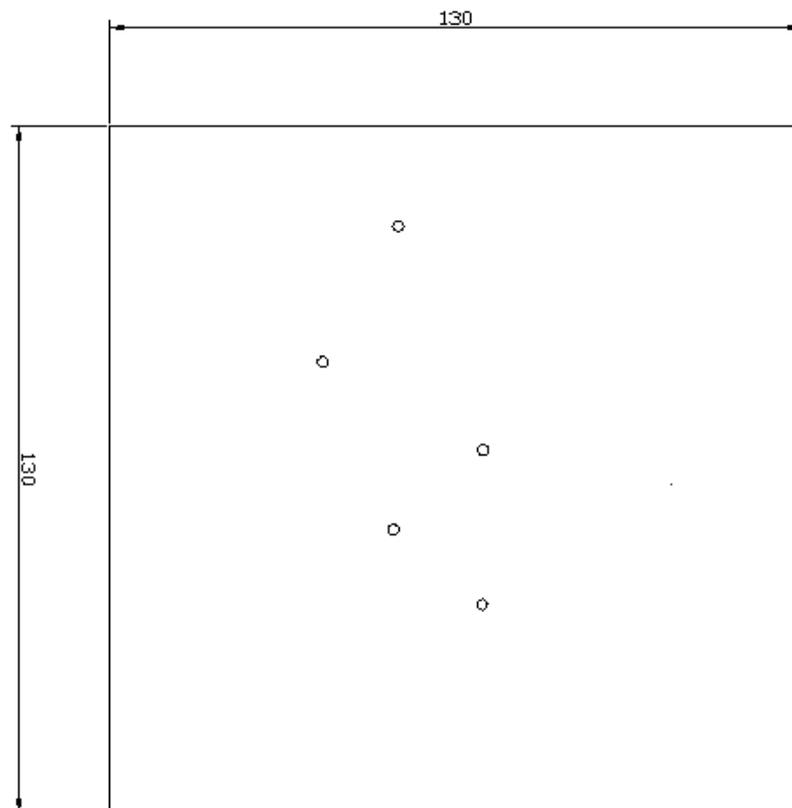


Figura 6.19: Esboço da peça de 5 furos

Na Figura 6.21, mostra o esboço da peça com treze furos e na Figura 6.22 ilustra a peça depois da usinagem. Na foto não aparece o furo no canto direito por causa da fixação da peça na mesa.

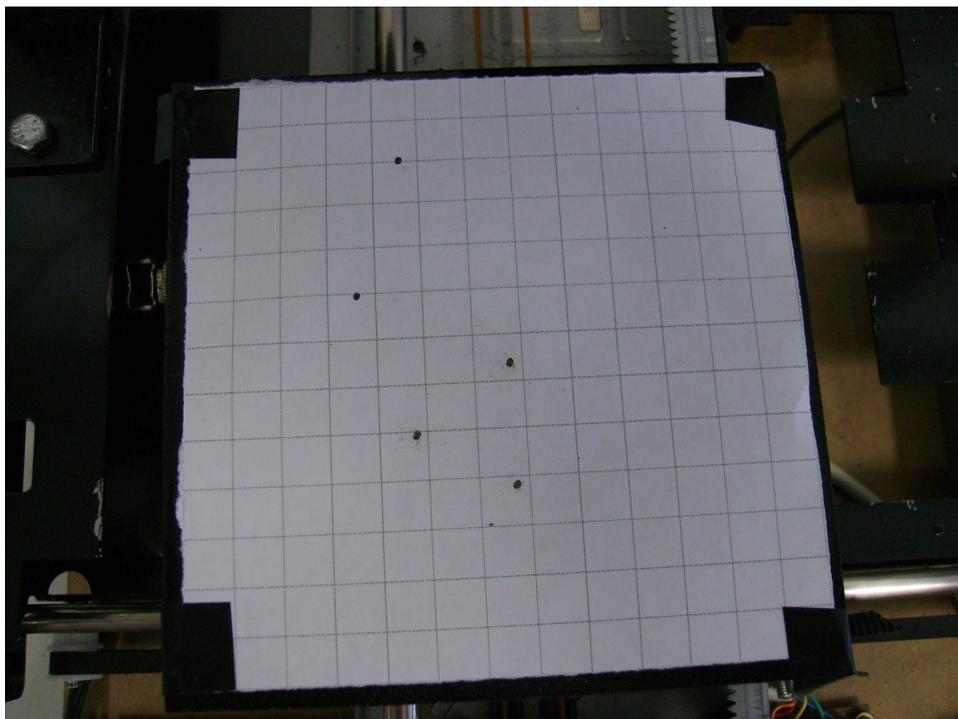


Figura 6.20: Usinagem de 5 furos

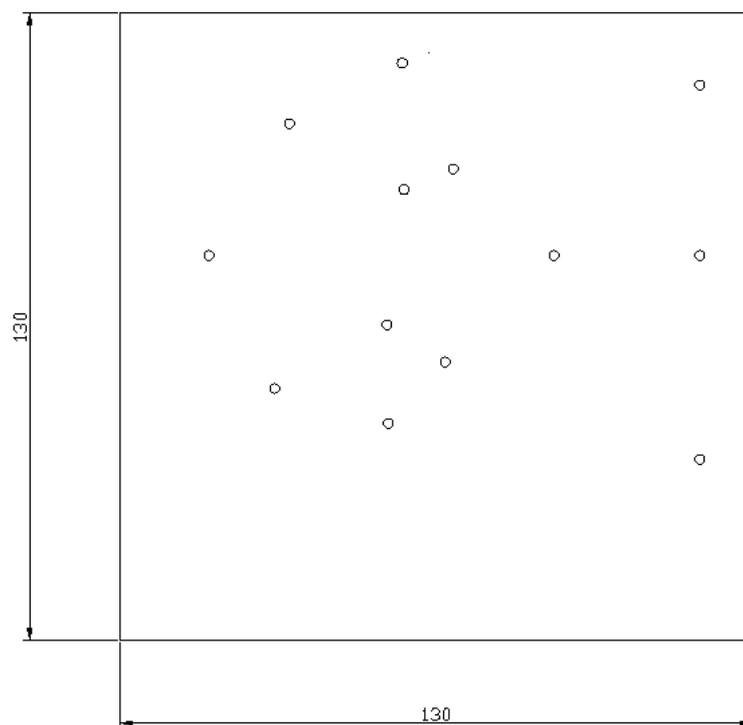


Figura 6.21: Esboço da peça de 13 furos

Foram realizados testes com quantidades de pontos diferentes para analisar o comportamento do algoritmo de otimização. Na Tabela 6.2 é mostrado um comparativo entre

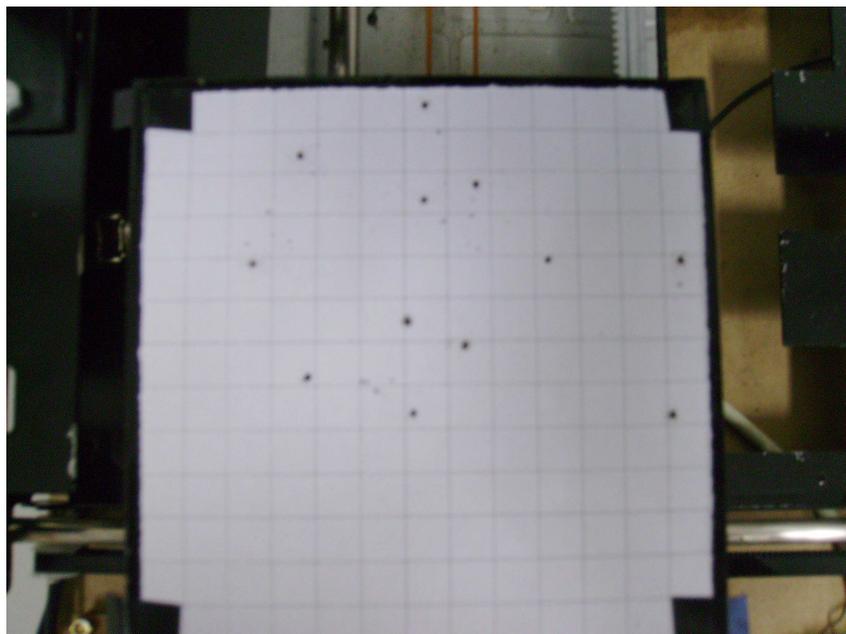


Figura 6.22: Usinagem de 13 furos

os testes feitos com os arquivos que não utilizaram o algoritmo e os que utilizaram o algoritmo de otimização. Na Figura 6.23 é mostrado o gráfico comparativo entre os tempos de usinagem que utilizaram o otimizador e os que não utilizaram o otimizador.

Na Tabela 6.2, pode-se notar que com o aumento do número de furos o otimizador teve um melhor desempenho. Essa diferença de tempo na usinagem pode ser percebida ao observar o funcionamento do protótipo, pois quando sai da *home position* a primeira vez gasta um tempo fixo para executar o primeiro furo, logo quando se tem poucos furos esse tempo se torna significativo, porém fica evidente que quando aumenta-se o número de furos o tempo diminui, pelo fato do tempo de descida ser diluído na quantidade de furos existentes no projeto.

Num dos testes foi feito um arquivo com 150 furos, mas por limitações elétricas e mecânicas, não foi possível a realização do teste, uma vez que o motor de passo do eixo Z, responsável pela descida da ferramenta de corte, teve um aquecimento além do esperado.

Tabela 6.2: Usinagem sem otimizador X Usinagem com otimizador

Item	Quantidade de pontos	Sem otimizador (s)	Com otimizador (s)	Otimização %
1	13	112	110	1,79
2	36	367	288	21,53
3	50	501	382	23,75
4	100	972	754	22,43

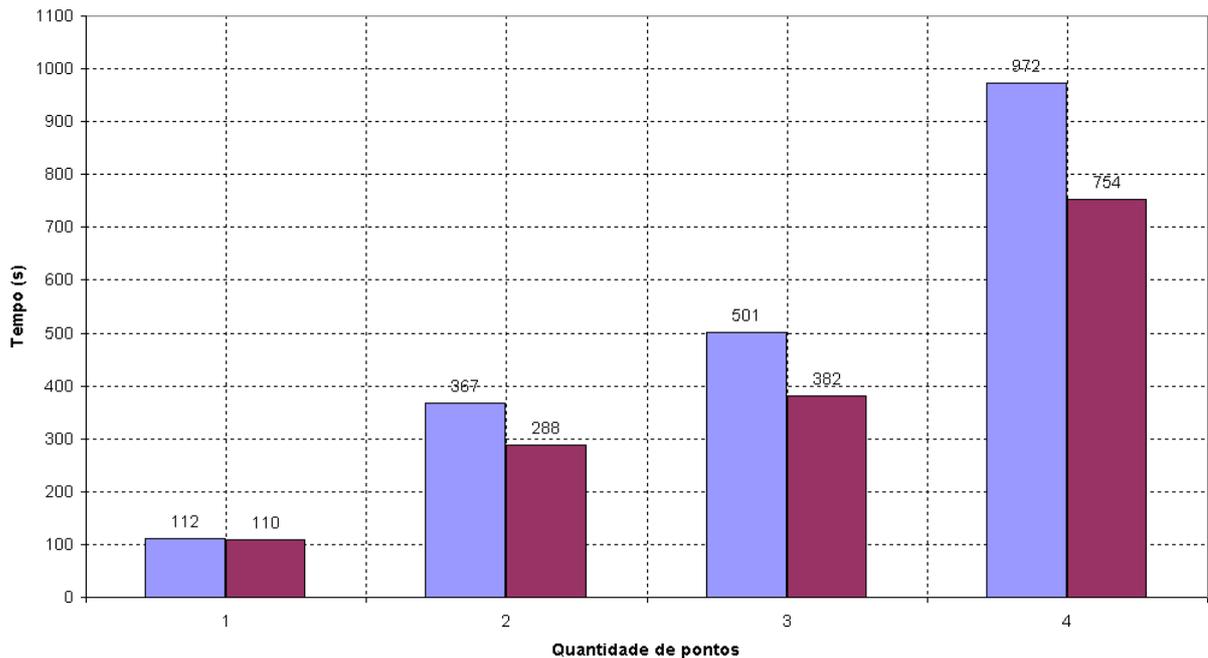


Figura 6.23: Usinagem sem otimizador X Usinagem com otimizador

No Capítulo 6 foram mostrados e discutidos todos os aspectos nas áreas envolvidas para a realização do projeto. Na área mecânica se teve problemas de transmissão de torque, atrito, precisão, porém como o foco do presente trabalho não era construção da máquina-ferramenta e sim a validação do conceito do STEP-NC para a troca de informação no processo de manufatura. Na área elétrica as placas de controle e os *drivers* desenvolvidas atenderam as necessidades elétricas, onde através do microcontrolador se conseguiu minimizar o problema de precisão. Na área de *software* o compilador desenvolvido atendeu as expectativas do projeto, tendo em vista que segundo Suh (2003) o trabalho desenvolvido atende as características das soluções referentes à implementação do STEP-NC no âmbito industrial, ver na seção 4.2.2 maiores explicações.

No desenvolvimento desse trabalho buscou-se publicações sobre o que está sendo desenvolvido no mundo, foram encontrados alguns trabalhos que utilizam abordagem semelhante, como Calabrese (2007), que fez a implementação da norma ISO14649. Mas essa implementação utilizou um pós-processador do STEP-NC para Código-G. Esse tipo de implementação tem suas limitações pelo fato da não paridade entre a ISO14649 e a ISO6983, assim não utilizando todos os recursos disponíveis na norma. Suh (2003) fez uma implementação semelhante a desenvolvida no trabalho (CALABRESE F. e CELENTANO 2007), (SUH 2003).

## 7 CONCLUSÃO

A importância da portabilidade de informações de usinagem seja via programas ou interfaces de dados vem crescendo constantemente no cenário da manufatura. A interoperabilidade entre sistemas computacionais de apoio à manufatura é essencial para atingir o objetivo de integração da manufatura auxiliada por computador. Este trabalho explorou a literatura no que tange a sistemas CAD, CAM, Máquinas-ferramenta a Controle Numérico e as normas STEP (ISO10303) e STEP-NC(ISO14649).

A estrutura de dados pode ser facilmente utilizada em modelos de informação para projeto e manufatura, integrados e aderentes a STEP, criando uma via livre de erros de interpretação desde o projeto até a usinagem da peça. A norma ISO 14649 define um conjunto de estratégias de usinagem a serem utilizadas. Algumas destas com definições muito próximas aos ciclos fixos na ISO 6983. Apesar da tarefa ser realizada de maneira semelhante, nem sempre é possível fazer uma relação 1:1 entre ciclos de ISO6983 e *workingsteps* do ISO14649. Para garantir o máximo de aderência à norma, é desejável que os sistemas CAD/CAM possam capturar informações das operações/estratégias de acordo com a ISO 14649. Sem este nível de conhecimento este sistema apenas traduz os ciclos do código G/M em funções básicas do STEP-NC perdendo muitas das possibilidades de operação utilizando controladores inteligentes como previsto no modelo.

No desenvolvimento do protótipo de um CNC baseado em Arquitetura Aberta aderente a STEP-NC, buscou-se publicações sobre o que está sendo feito no mundo, foram encontrados alguns trabalhos que utilizam abordagem semelhante como comentado no final do Capítulo 6. Entretanto, para efetivar essa implementação no CNC, a grande maioria fez a interpretação do arquivo STEP-NC(ISO14649) para código-G (ISO6983), esse tipo de implementação tem suas limitações pelo fato da não paridade entre a ISO14649 e código G, assim não utilizando todos os recursos disponíveis na norma. Já o desenvolvido no presente trabalho tem um interpretador STEP-NC como parte do CNC, com capacidade de geração de caminho de ferramenta, é o tipo básico onde o movimento é executado fielmente, é baseado na estratégia de usinagem e seqüência como especificado pelo STEP-NC. Ou seja, não utiliza sistemas legados sendo uma solução que avança na integração da

manufatura garantindo interoperabilidade via o padrão internacional ISO14649.

Apesar das limitações do equipamento, principalmente das mecânicas, pois o protótipo foi construído de sucatas de *scanner*, impressoras, teve problemas com atrito, alinhamento, mas eletronicamente e através de *software* foram compensados esses erros.

Todavia, o foco do trabalho não foi a construção do protótipo propriamente dito, mas sim a aplicação dos conceitos de arquitetura aberta, visto que baseado nessas idéias foi desenvolvido um CNC de arquitetura aberta aderente a STEP-NC. Assim os objetivos traçados no início do trabalho foram alcançados, provando a eficácia da troca de informações de manufatura no que diz respeito a máquinas-ferramenta, a viabilidade da norma ISO14649 para controle de máquinas-ferramenta. Tendo esse trabalho já desenvolvido criou-se uma estrutura para pesquisas de interfaces abertas no âmbito do PPGEEL.

## 7.1 Trabalhos futuros

Atualmente o protótipo faz trabalhos de usinagem bem limitados, no caso furações. Para trabalhos futuros, poderia ser feito:

- Desenvolver a partes de *software* e mecânica para a realizar o fresamento de placas de circuito impresso;
- Ter um controle de posicionamento através de encoders;
- Ter um controle de temperatura nos motores dos eixos;
- Fazer uma integração de manufatura através do protocolo TCP/IP para proporcionar a usinagem de placas via internet;
- Melhorar a dissipação de calor do motor que sofreu sobre-aquecimento;
- Melhorar o algoritmo de otimização;
- Melhorar os aspectos mecânicos e
- Desenvolver um novo protótipo onde o circuito integrado possa assumir a maior parte do processamento, ficando o PC somente para interface.

## Referências

- ADAMOWSKI 2008 ADAMOWSKI, J. C. *Princípios de projetos de máquinas*. 2008. Disponível em: <http://www.poli.usp.br>. Acesso em: 01/03/08.
- AHO 1995 AHO, A. e. a. *Compiladores: Princípios, Técnicas e Ferramentas*. 1. ed. Rio de Janeiro: Guanabara Koogan, 1995.
- ALLEN 2003 ALLEN, R. D. e. a. The Design of a STEP-NC Compliant Agent Based CAD/CAM System. *Internacional Conference on Flexible Automation and Intelligent Manufacturing (FAIM2003)*, p. 530–540, 2003.
- BACK 1983 BACK, N. *Metodologia de projeto de produtos industriais*. 1. ed. Rio de Janeiro: Guanabara Dois, 1983.
- BEDWORTH 1991 BEDWORTH, D. D. e. a. *Computer Integrated Design and Manufacturing*. 1. ed. [S.l.]: McGraw-Hill, 1991.
- Bentley System Inc. 2008 Bentley System Inc. 2008. Disponível em: <http://www.bentley.com>. Acesso em: 10/05/08.
- BONACORSO N.G. e NOLL 2007 BONACORSO N.G. E NOLL, V. *Automação Eletropneumática*. 11. ed. São Paulo: Érica, 2007.
- BRUNNERMEIER. S. B. e MARTIN. S. A. 1999 BRUNNERMEIER. S. B. e MARTIN. S. A. Interoperability Cost Analysis of the U.S. Automotive Supply Chain. *RESEARCH TRIANGLE INSTITUTE*, 1999.
- CALABRESE F. e CELENTANO 2007 CALABRESE F. E CELENTANO, G. Design and Realization of a STEP-NC Compliant CNC Embedded Controller. *Emerging Technologies Factory Automation*, p. 1010–1017, 2007.
- CAPELLI 2002 CAPELLI, A. Máquinas CNC. *Revista Mecatrônica Atual*, p. 30–35, 2002.

- CAPELLI 2006 CAPELLI, A. *Automação Industrial: Controle do Movimento e Processos Contínuos*. 1. ed. São Paulo: Érica, 2006.
- CASE, K. 1992 CASE, K. Feature Technology - an Integration Methodology for CAD and CAM. *International Conference on Manufacturing Automation*, p. 613–624, 1992.
- Delcam Plc. 2008 Delcam Plc. 2008. Disponível em: <http://www.delcam.co.uk/>. Acesso em: 10/05/08.
- D'ISSY 2004 D'ISSY, M. Indústria automatiza processos produtivos. *Revista CADesign*, p. 20–22, 2004.
- EROL N. A. e ALTINTAS 2000 EROL N. A. E ALTINTAS, Y. e. I. M. Open System Architecture Modular Tool Kit for Motion and Machining Process Control. *IEEE/ASME Transactions on Mechatronics*, p. 281–291, 2000.
- FERNEDA 1999 FERNEDA, A. B. *Integração Metrologia, CAD e CAM: Uma Contribuição ao Estudo da Engenharia Reversa*. Dissertação (Mestrado) — Escola de Engenharia de São Carlos da Universidade de São Paulo, São Paulo, 1999.
- FIALHO 2003 FIALHO, A. *Automação Pneumática - Projetos, Dimensionamento e Análise de Circuitos*. 4. ed. São Paulo: Érica, 2003.
- GROOVER 2001 GROOVER, M. P. *Automation, production systems, and computer-integrated manufacturing*. 3. ed. New Jersey: Prentice-Hall, 2001.
- GRUNE 2001 GRUNE, D. *Projeto Moderno de Compiladores: Implementações e Aplicações*. 1. ed. Rio de Janeiro: Campus, 2001.
- HOUNSELL 1998 HOUNSELL, M. d. S. *Feature-based Validation reasoning for Intent-driven Engineering Design*. Tese (Doutorado) — Loughborough University, UK, agosto 1998.
- HOUNSELL 2008 HOUNSELL, M. S. *Básico de programação de CNC (linguagem G)*. 2008. Disponível em: <http://www.joinville.udesc.br/departamentos/dcc/professores/marcelo/disciplinas/AIN/cam/programacao.html>. Acesso em: 01/03/08.
- ISO10303-1 1994 ISO10303-1. *Industrial automation systems and integration Product data representation and exchange - Overview and Fundamental Principles, International Standard, ISO 10303-1, ISO TC184/SC4*. [S.l.], 1994.

- ISO14649-1 2003 ISO14649-1. *Data Model for Computerized Numerical Controllers: Part 1 - Overview and fundamental principles*. [S.l.], 2003.
- ISO6983-1 1982 ISO6983-1. *Numerical control of machines-program format and definition of address words-part 1: data format for positioning, line and contouring control systems*. [S.l.], 1982.
- KERN 1997 KERN, V. M. *Manutabilidade da Semântica de Modelos de Dados de Produtos Compartilhados em Rede Interoperável*. Tese (Doutorado) — Universidade Federal de Santa Catarina, Brasil, agosto 1997.
- KOLLA 2002 KOLLA, S. Evaluation of component-based reconfigurable machine controllers. *World Automation Congress Proceedings of the 5th Biannual*, p. 625–630, 2002.
- LOFREDO 2008 LOFREDO, D. *Fundamentals of STEP Implementation*. 2008. Disponível em: <http://www.steptools.com/library/fundimpl.pdf>. Acesso em: 01/03/08.
- LUTZ 1997 LUTZ, P. e. a. OSACA - the vendor neutral Control Architecture. *Conference on Integration in Manufacturing*, 1997.
- MARCIANO 2008 MARCIANO, J. P. P. *Introdução ao Controle Numérico*. 2008. Disponível em: <http://www.poli.usp.br>. Acesso em: 01/03/08.
- MDSI 2008 MDSI. *OpenCNC Architecture*. 2008. Disponível em: <http://www.mdsi2.com>. Acesso em: 01/03/08.
- MESSIAS 2008 MESSIAS, A. R. *Controle de motor de passo através da porta paralela*. 2008. Disponível em: <http://www.rogercom.com/pparalela/IntroMotorPasso.html>. Acesso em: 10/05/08.
- MICROCHIP 2008 MICROCHIP. *PIC18FXX2-Data Sheet*. 2008. Disponível em: <http://www.microchip.com/>. Acesso em: 21/01/08.
- MITSUBISHI 1997 MITSUBISHI, M. e. a. An Open Architecture CNC CAD-CAM Machining System with Data-Base Sharing and Mutual Information Feedback. *Annals of the International Institution for Production Engineering Research (CIRP Annals.)*, p. 269–274, 1997.

- MOREIRA 1998 MOREIRA, J. F. P. *Operação E Controlo de Equipamentos Industriais à Distância - Desenvolvimento de Teleserviços para Sistemas FAC*. Dissertação (Mestrado) — Universidade do Minho, Portugal, 1998.
- NASSEHI 2006 NASSEHI, A. e. a. STEP-NC compliant process planning as an enabler for adaptive global manufacturing. *Robotics and Computer-Integrated Manufacturing*, p. 456–467, 2006.
- NEWMAN 2003 NEWMAN, S. T. e. a. CAD/CAM solutions for STEP-compliant CNC manufacture. *Computer Integrated Manufacturing*, p. 590–597, 2003.
- NIST 2008 NIST 2008. *Enhanced Machine Controller*. Disponível em: <http://sourceforge.net/projects/emc>. Acesso em: 10/11/08.
- OMAC 2008 OMAC 2008. *Business Justification of Open Architecture Control*. Disponível em: <http://www.arcweb.com>. Acesso em: 15/02/08.
- OSACA 2008 OSACA 2008. *Open System Architecture for Controls within Automation Systems*. Disponível em: <http://www.osaca.org/>. Acesso em: 21/01/08.
- PANSIERA 2002 PANSIERA, P. E. Retrofitting. *Revista Mecatrônica Atual*, p. 48–54, 2002.
- PAZOS 2002 PAZOS, F. *Automação de Sistemas e Robótica*. 1. ed. Rio de Janeiro: Axcel Books, 2002.
- RIBEIRO e ROZENFELD 2008 RIBEIRO, D. D.; ROZENFELD, H. *CAM (Computer Aided Manufacturing)*. 2008. Disponível em: <http://www.numa.org.br/>. Acesso em: 02/03/08.
- ROMI 2008 ROMI. *Centros de Usinagem Verticais*. [S.l.], 2008.
- ROSSO Jr. 2002 ROSSO Jr., R. S. e. a. Future Issues for CAD/CAM and Intelligent CNC Manufacture - IMC-19. *Internacional Manufacturing Conference*, p. 115–124, 2002.
- ROSSO Jr. 2003 ROSSO Jr., R. S. e. a. Estrutura de Dados para Sistemas CAD/CAM aderente à STEP. *Livro de Actas do VI Congresso Ibero-Americano de Engenharia Mecânica - Universidade de Coimbra*, p. 1019–1024, 2003.

- ROSSO Jr. 2005 ROSSO Jr., R. S. U. *STEP Compliant CAD/CAPP/CAM System For Rotational Asymmetric Parts*. Tese (Doutorado) — Loughborough University, UK, agosto 2005.
- SAWADA 1997 SAWADA, C. e. a. Open Controller Architecture OSEC-II: Architecture Overview and Prototype Systems. *Emerging Technologies and Factory Automation, 1997. Proceedings. ETFA '97. 1997 6th IEEE International Conference on Manufacturing*, p. 543–550, 1997.
- SCHÜTZER 2008 SCHÜTZER, K. *Programação CNC*. 2008. Disponível em: <http://www.unimep.br/feau/scpm>. Acesso em: 01/03/08.
- SIEMENS 2008 SIEMENS 2008. *Controle Numérico - CNC*. Disponível em: <http://www.siemens.com.br/templates/homeune.aspx?channel=3878>. Acesso em: 10/05/08.
- SILVA 2001 SILVA, E. C. N. CAD/CAE/CAM. *Revista Mecatrônica Atual*, p. 38–47, 2001.
- SIVITTER 2004 SIVITTER, S. Usinagem baseada em modelos sólidos será padrão. *Revista CADesign*, p. 14–15, 2004.
- SOUZA 2004 SOUZA, A. F. *Contribuições ao Fresamento de Geometrias Complexas Aplicando a Tecnologia de Usinagem com Altas Velocidades*. Dissertação (Mestrado) — Escola de Engenharia de São Carlos da Universidade de São Paulo, São Paulo, 2004.
- STEPTOOLS 2008 STEPTOOLS. *STEP e STEP-NC*. 2008. Disponível em: <http://www.steptools.com>. Acesso em: 02/03/08.
- SUH 2003 SUH, S. e. a. Architecture and implementation of a shop-floor programming system for STEP-compliant CNC. *ELSEVIER*, p. 1069–1083, 2003.
- THOMAZINI D. e ALBUQUERQUE 2005 THOMAZINI D. E ALBUQUERQUE, P. *Sensores Industriais - Fundamentos e Aplicações*. 1. ed. São Paulo: Érica, 2005.
- TURBAN E. e RAINER 2003 TURBAN E. E RAINER, R. e. P. R. *Administração da Tecnologia da Informação (TI): Teoria e Prática – Introduction to Information Technology*. 1. ed. Rio de Janeiro: Campus, 2003.

- WANG 2006 WANG, W. e. a. A research on reconfigurable numerical controller based on embedded system. *3rd International Conference on Mechatronics*, p. 189–193, 2006.
- WEG 2008 WEG. *Servoconversores SCA-05 e Servomotores SWA*. 2008. Disponível em: <http://catalogo.weg.com.br/FILES/Artigos/4-346.pdf>. Acesso em: 10/06/08.
- WEINERT 2000 WEINERT, K. e. a. Usinagem de superfícies complexas. *Revista Máquinas e Metais*, p. 18–27, 2000.
- WILLIAMS 1988 WILLIAMS, D. J. *Manufacturing Systems: an introduction to technologies*. 1. ed. [S.l.]: Stony Stratford: Open University Press, 1988.
- XU 2004 XU, X. W. Striving for a total integration of CAD, CAPP, CAM and CNC. *Robotics and Computer Integrated Manufacturing*, p. 101–109, 2004.
- XU 2006 XU, X. W. Making CNC machine tools more open, interoperable and intelligent - a review of the technologies. *ELSEVIER - Computers in Industry*, p. 141–152, 2006.
- YONGLIN 2005 YONGLIN, C. An evaluation space for open architecture controllers. *The International Journal of Advanced Manufacturing Technology*, p. 351–358, 2005.
- ZEID 1991 ZEID, I. *CAD/CAM Theory and practice*. 1. ed. New Jersey: McGraw-Hill, 1991.