

**UNIVERSIDADE METODISTA DE PIRACICABA**

**FACULDADE DE ENGENHARIA, ARQUITETURA E URBANISMO  
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE PRODUÇÃO**

**DESENVOLVIMENTO DE UM SISTEMA PARA MONITORAMENTO DE  
DADOS EM HSM (*HIGH SPEED MACHINING*)**

**ERIK GUSTAVO DEL CONTE**

**ORIENTADOR: PROF. DR.-ING. KLAUS SCHÜTZER**

Santa Bárbara d'Oeste, SP

2008

**UNIVERSIDADE METODISTA DE PIRACICABA**

**FACULDADE DE ENGENHARIA, ARQUITETURA E URBANISMO  
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE PRODUÇÃO**

**DESENVOLVIMENTO DE UM SISTEMA PARA MONITORAMENTO DE  
DADOS EM HSM (*HIGH SPEED MACHINING*)**

**ERIK GUSTAVO DEL CONTE**

ORIENTADOR: PROF. DR.-ING. KLAUS SCHÜTZER

Dissertação de Mestrado apresentada ao  
Programa de Pós-Graduação em  
Engenharia de Produção, da Faculdade de  
Engenharia, Arquitetura e Urbanismo, da  
Universidade Metodista de Piracicaba -  
UNIMEP

Santa Bárbara d'Oeste, SP

2008

---

**DESENVOLVIMENTO DE UM SISTEMA PARA MONITORAMENTO DE  
DADOS EM HSM (*HIGH SPEED MACHINING*)**

**ERIK GUSTAVO DEL CONTE**

Prof. Dr.-Ing. Klaus Schützer, Presidente  
PPGEP - UNIMEP

Prof. Dr. Alvaro José Abackerli  
PPGEP - UNIMEP

Prof. Dr. Jefferson de Oliveira Gomes  
CCM - ITA

## **Agradecimentos**

Ao Professor Dr.-Ing. Klaus Schützer pela orientação e incentivo para a conclusão deste trabalho.

Ao Professor Dr. Alvaro José Abackerli pelas discussões sobre este trabalho.

Aos companheiros de trabalho do Laboratório de Sistemas Computacionais para Projeto e Manufatura pelo apoio, incentivo e orientação.

À CAPES - Coordenação de Aperfeiçoamento de Pessoal de Nível Superior, pela concessão da bolsa de estudos.

Ao Centro de Competência em Manufatura pela execução de ensaios iniciais com o CNC Siemens 840D.

À empresa Siemens, pelo treinamento e disponibilização do pacote OEM para a comunicação com os CNCs 810D e 840D.

„Não olhes para trás, e não te detenhas em parte alguma da planície“  
(Gen 19,17)

## Sumário

Lista de Figuras .....	VII
Lista de Tabelas .....	IX
Lista de Siglas .....	X
Lista de Institutos e Laboratórios.....	XII
Resumo .....	XIII
Abstract.....	XIV
<b>1 Introdução .....</b>	<b>1</b>
1.1 A importância deste trabalho .....	4
1.2 Estrutura do trabalho .....	4
<b>2 Monitoramento do Processo de Usinagem HSM .....</b>	<b>6</b>
2.1 Monitoramento e controle do processo de usinagem .....	10
2.1.1 Aquisição de dados .....	12
2.1.2 Estratégias de tomada de decisão .....	14
2.1.3 Níveis de controle do processo de usinagem CNC .....	16
2.1.3.1 Controle do servo.....	17
2.1.3.2 Interpolador.....	18
2.1.3.3 Controle do processo.....	20
2.1.3.4 Controle de força .....	21
2.1.3.5 Controle de vibração.....	21
2.1.4 Arquitetura de controle distribuído.....	22
2.2 CNC Comando Numérico Computadorizado .....	24
2.2.1 Características para aplicação em HSM .....	25
2.2.2 Controlador de arquitetura aberta.....	26
2.2.3 Requisitos para o sistema de controle de arquitetura aberta .....	28
2.3 Sistemas baseados em computador.....	30
2.3.1 Requisitos de <i>software</i> .....	32
2.3.2 Projeto de arquitetura do sistema.....	33
2.3.3 Projeto de <i>software</i> de tempo real.....	34

---

2.3.4	Projeto estruturado .....	35
2.3.5	Teste de <i>software</i> .....	37
2.3.6	Teste de sistema .....	38
2.4	Protocolos de comunicação industrial .....	39
2.4.1	Fatores de escolha do protocolo de comunicação .....	40
2.4.2	Principais protocolos de redes de campo.....	42
3	Proposta do Trabalho .....	44
3.1	Objetivos.....	44
3.2	Metodologia .....	44
4	Desenvolvimento do Sistema de Monitoramento .....	46
4.1	Requisitos de <i>software</i> .....	46
4.2	Arquitetura do sistema.....	46
4.3	Diagrama de Fluxo de Dados (DFD) .....	48
4.4	Ensaio preliminares .....	51
4.5	Estratégias de monitoramento .....	52
4.5.1	Módulo de aquisição de dados.....	53
4.5.2	Módulo de transmissão de dados.....	54
4.5.3	Módulo de análise de dados.....	55
4.5.4	Interface com o usuário .....	56
4.6	Ensaio para verificação das premissas.....	57
5	Análise dos Resultados .....	60
5.1	Análise dos ensaios preliminares .....	60
5.2	Taxa de aquisição e resolução da variável monitorada .....	65
5.3	Identificação dos parâmetros ótimos de programação NC .....	67
5.4	Acesso e customização de funções no CNC aberto.....	71
6	Conclusões e Sugestões para Trabalhos Futuros .....	73
7	Bibliografia Referenciada .....	75

Anexo 1 – Programa NC com interpolação linear .....	82
Anexo 2 – Programa NC com interpolação polinomial .....	86



## Lista de Figuras

Figura 1.1: Conceito do sistema monitoramento .....	3
Figura 2.1: Características do processo HSM [16] .....	6
Figura 2.2: Tecnologias envolvidas no processo HSM [16] .....	7
Figura 2.3: Estrutura atual de controle na manufatura CNC [23] .....	8
Figura 2.4 : Esquema do monitoramento e controle de usinagem CNC [28] .....	11
Figura 2.5 : Família de normas IEEE 1451 [36] .....	13
Figura 2.6: Algoritmo do monitoramento de desgaste da ferramenta [43] .....	15
Figura 2.7 : Arquitetura de controle da máquina-ferramenta CNC [29] .....	16
Figura 2.8: Loop de controle de trajetória [45] .....	17
Figura 2.9: Influência do servo-lag no na exatidão da trajetória da ferramenta [28] .....	18
Figura 2.10: Proposta de cadeia CAD/CAM/CNC para a usinagem de curvas [20] .....	19
Figura 2.11 : Arquitetura de controle distribuído [53] .....	22
Figura 2.12 : Controle distribuído de um centro de torneamento [37] .....	23
Figura 2.13: Diagrama de blocos do controle de movimentos [56] .....	25
Figura 2.14 : Estrutura do sistema de controle modular e aberto [14] .....	28
Figura 2.15 : Estrutura típica de níveis do CNC de arquitetura aberta [15] .....	29
Figura 2.16: Elementos de um sistema baseado em computador [59] .....	30
Figura 2.17: O modelo em cascata [58] .....	31
Figura 2.18: Modelo de arquitetura [59] .....	33
Figura 2.19: Arquitetura de um sistema para aquisição e monitoramento de fluxo [60] ...	35
Figura 2.20: Buffer em anel para aquisição de dados [60] .....	35
Figura 2.21: Diagrama de fluxo de dados (adaptada de Pfleeger [58]). .....	36
Figura 2.22: Níveis e protocolos de comunicação na pirâmide de automação [64] .....	40
Figura 4.1: Diagrama de contexto de arquitetura do sistema .....	47
Figura 4.2: Diagrama de fluxo de dados inicial .....	48
Figura 4.3: Diagrama de fluxo de dados final .....	49
Figura 4.4: Fluxograma de operação do sistema de monitoramento .....	50
Figura 4.5 : Diagrama de blocos utilizado nos ensaios preliminares .....	52
Figura 4.6: Esquema de funcionamento das ações síncronas [75]. .....	53

---

Figura 4.7: Diagrama de blocos do módulo de transmissão de dados [76] .....	55
Figura 4.8: Módulo de análise de dados .....	56
Figura 4.9: Interface com o usuário .....	57
Figura 4.10: Modelo CAD utilizado para a programação NC.....	59
Figura 5.1: Gráfico de probabilidade normal com rede de 187,5 Kbps [72].....	62
Figura 5.2: Gráfico de probabilidade normal com rede de 1.5 Mbps [72]. .....	62
Figura 5.3: Gráfico de probabilidade normal rede de 1.5 Mbpps e função Write File .....	63
Figure 5.4. Influência da velocidade de avanço na resolução da variável [76].....	67
Figura 5.5: Comportamento da velocidade de avanço para interpolação linear.....	68
Figura 5.6: Comportamento da velocidade de avanço para a interpolação polinomial ....	69
Figura 5.7: Comportamento da aceleração para a interpolação linear .....	70
Figura 5.8: Comportamento da aceleração para a interpolação polinomial.....	70

## Lista de Tabelas

Tabela 2.1: Requisitos do controlador de arquitetura aberta [15] .....	29
Tabela 2.2: Classificação ortogonal de defeitos [58] .....	38
Tabela 5.1: Dados coletados nos ensaios preliminares.....	60
Tabela 5.2: Dados coletados nos ensaios preliminares com função Write File.....	63
Tabela 5.3. Velocidade de avanço e intervalo entre leituras [76] .....	65
Tabela 5.4. Tempo de usinagem .....	67
Tabela 5.5. Velocidade de avanço.....	68

## Lista de Siglas

API	Application Programming Interface
CAD	Computer Aided Design
CAM	Computer Aided Manufacturing
CAN	Controller Area Network
CNC	Comando Numérico Computadorizado
DDE	Dynamic Data Exchange
DFD	Diagrama de Fluxo de Dados
HMI	Human Machine Interface
HSM	High Speed Machining
I/O	Input/Output
IEC	International Electrotechnical Commission
ISA	Instrumentation, Systems and Automation Society
JOP	Japanese Open Promotion Group
MIT	Massachusetts Institute of Technology
MMC	Man Machine Communication
MPI	Multi Point Interface
NC	Numerical Control
NCDDE	Dynamic data exchange with the NC kernel
NCK	Numerical Control Kernel
NURBS	Non-Uniform Rational B-Spline
OEM	Original Equipment Manufacturer

OMAC	Open Modular Architecture Controllers
OSACA	Open System Architecture for Controls within Automation Systems
OSEC	Open System Environment for Manufacturing
OSI	Open Systems Interconnect
PCI	Peripheral Component Interconnect
PLC	Programmable Logic Controller
RPM	Rotação por minuto

## **Lista de Institutos e Laboratórios**

SCPM	Laboratório de Sistemas Computacionais para Projeto e Manufatura
UNIMEP	Universidade Metodista de Piracicaba
CCM	Centro de Competência em Manufatura
IEEE	Institute of Electrical and Electronics Engineers
DIN	Deutsches Institut für Normung

## Resumo

**DEL CONTE, Erik Gustavo. Desenvolvimento de um sistema para monitoramento de dados em HSM (*High Speed Machining*). 2007. 85p. Dissertação de Mestrado - Faculdade de Engenharia, Arquitetura e Urbanismo, Universidade Metodista de Piracicaba, Santa Bárbara d'Oeste.**

A necessidade de implementação de sistemas de monitoramento no processo de usinagem com altas velocidades tem aumentado devido aos altos níveis de desempenho e segurança requeridos nas máquinas ferramentas. Os parâmetros de programação NC são tradicionalmente baseados em manuais ou na experiência do usuário e, em muitos casos, tais parâmetros não exploram toda a capacidade da máquina-ferramenta. O conceito do CNC de arquitetura aberta está consolidado como meio para a integração de funções customizadas do usuário, porém não foi estabelecido um padrão comum de arquitetura aberta que seja aceito pelos fabricantes de CNC e utilizado em grande escala no ambiente industrial. Com os objetivos específicos de monitorar a velocidade de avanço e aceleração dos eixos, identificar os parâmetros ótimos da programação NC e avaliar a implementação de funções de monitoramento no CNC aberto, foi desenvolvido um sistema de monitoramento de dados do CNC de arquitetura aberta aplicado no processo de usinagem HSM. Ensaio considerando diferentes parâmetros de programação NC da interpolação da trajetória da ferramenta foram executados e os resultados demonstraram que através dos dados monitorados pode-se identificar os parâmetros ótimos de programação NC. O nível de implementação de funções de monitoramento de um CNC de arquitetura aberta foi avaliado, demonstrando-se as limitações para o desenvolvimento de funções customizadas.

**Palavras-chave:** Sistema de monitoramento, CNC de arquitetura aberta, usinagem com altas velocidades.

## **Abstract**

**DEL CONTE, Erik Gustavo. Desenvolvimento de um sistema para monitoramento de dados em HSM (*High Speed Machining*). 2007. 85p. Dissertação de Mestrado - Faculdade de Engenharia, Arquitetura e Urbanismo, Universidade Metodista de Piracicaba, Santa Bárbara d'Oeste.**

The necessity of monitoring systems implementation in High Speed Machining process (HSM) has increased with the high level of performance and security required in machine tools. The cutting process parameters of a CNC machine are traditionally based on handbooks or on user's practical experience. This approach results in conservative cutting parameters that do not explore the full performance of machine tools. The Open CNC concept is widely used to provide the integration of user's functions and communication with CNC data through the network, but a common standard with large application in industries has not been established. With the specific objectives of feedrate and axis acceleration monitoring, identifying optimized NC parameters and evaluating the implementation of open CNC monitoring functions, a monitoring system of open CNC data applied in High Speed Machining process was developed. Experiments considering different tool path interpolation parameters were executed and the results show that through the monitored data it's possible to identify the optimized parameters of NC programming. The level of monitoring functions implementation of an open CNC was evaluated, showing the limitations for customized functions development.

**Keywords:** Monitoring System, Open CNC, High Speed Machining.



# 1 Introdução

A usinagem com altíssima velocidade (*High-Speed Machining* - HSM) é amplamente utilizada na indústria de moldes e matrizes, para a usinagem de superfícies complexas, e na indústria aeronáutica, na usinagem de ligas não-ferrosas. Devido a obtenção de um alto nível de qualidade geométrica da superfície e das altas taxas de remoção de material, consegue-se redução no tempo e custo do processo de usinagem, principais fatores de aumento da competitividade da organização. O impacto da otimização do processo de usinagem nas indústrias de moldes e matrizes resulta em ciclos menores de desenvolvimento de produtos, contribuindo para a redução do tempo de lançamento de produtos no mercado [1, 2, 3, 4].

O processo HSM é composto por um conjunto de tecnologias que abrangem desde a programação NC, através de sistemas CAM, com metodologias de interpolação da trajetória da ferramenta e parâmetros adequados, até as tecnologias de fixação da ferramenta e da peça, sistemas de acionamento, dispositivos de segurança, monitoramento do processo, entre outras, que interagem com a finalidade de fornecer resultados específicos no processo de usinagem [5].

As opções de integração e flexibilidade propostas pela nova geração de máquinas ferramenta são uma das principais características que visam otimização do processo de manufatura. A necessidade de fabricação de pequenos lotes de peças em prazos reduzidos e a capacidade do sistema de manufatura em absorver rapidamente as modificações nos produtos existentes faz com que a flexibilidade, a integração e a reconfiguração do sistema sejam itens presentes no ambiente produtivo. Porém existe uma lacuna entre as tecnologias de integração e a capacidade da inteligência embarcada de capturar informações para a otimização do processo através de sistemas de controle e monitoramento, além de, em muitos casos, existir uma limitação no nível de customização de funções oferecida pelo sistema de manufatura [6, 7].

A importância do monitoramento do processo HSM se deve ao aumento dos requisitos de performance e segurança do processo *high speed*, que obriga os fabricantes de Comando Numérico Computadorizado (CNC) a incluir funções de monitoramento e proteção dos motores, *drivers* e ferramentas de corte, assim como possibilitar funções de controle e monitoramento das forças de corte, vibrações e velocidades que resultam na otimização e redução de custos do processo de fabricação [8].

---

Os parâmetros de programação são tradicionalmente baseados na experiência do profissional ou em manuais técnicos. Isso resulta na utilização de parâmetros conservadores, pois a prioridade é a prevenção de falhas no processo. O monitoramento *online* de dados considera o comportamento da máquina durante o processo de usinagem e permite o ajuste *online* dos parâmetros com o objetivo de otimizá-los [9].

Segundo Wang [6], ainda falta uma infraestrutura eficiente para a integração dos equipamentos automatizados que proporcione acesso aos dados do processo no ambiente digital, o que torna necessário o desenvolvimento de novas tecnologias que permitam a comunicação em todos os níveis de informação nas máquinas ferramentas CNC.

Atualmente dois níveis de abertura de arquitetura são considerados pelos fabricantes de máquinas ferramenta e de CNC. Esses níveis são diferenciados pelo seu grau de importância no sistema de controle e são assim classificados: nível de abertura da comunicação homem-máquina (MMC – *Man Machine Communication*) e nível de abertura do *kernel* (núcleo de processamento do sistema operacional) do comando numérico (NCK – *Numerical Control Kernel*) [10].

A abertura que é proporcionada pelos modernos controladores de máquinas permitem o desenvolvimento e aplicação de funções de usuário em alto nível, em muitos casos, essa abertura é suficiente para a aplicação do monitoramento do processo. Em baixo nível, como no acesso a dados do controle de velocidade e corrente, não é possível o desenvolvimento de funções customizadas [11].

Segundo Park [12], a utilização do conceito *middleware* (programa que faz a comunicação entre outros *softwares*) no módulo do *kernel software* viabiliza a compatibilidade de comunicação entre os controladores, possibilitando a integração de máquinas CNC geograficamente distribuídas. As arquiteturas abertas prévias têm a mesma proposta e funcionalidade porém são logicamente incompatíveis.

Assim, conforme foi apresentado, identifica-se que o CNC de arquitetura aberta permite acesso e manipulação de dados sem a necessidade de modificações no hardware, porém, o CNC de arquitetura aberta também não oferece um método de interfaceamento de dados em todos os níveis de informação e são propostos por [11, 12, 13, 14, 15] novos métodos de configuração de *hardware* e *software* para viabilizar esta abertura.

Com base nesse cenário, o objetivo geral desse trabalho é o desenvolvimento de um sistema para monitoramento de dados de um CNC de arquitetura aberta aplicado no processo HSM. Na proposta do trabalho apresentada no capítulo 3 são detalhados os objetivos específicos e a metodologia desse trabalho.

Pretende-se fazer a aquisição de dados através da comunicação de um computador pessoal com o comando numérico da máquina ferramenta. O comando numérico a ser utilizado é o Siemens 810D, de arquitetura aberta, instalado num centro de usinagem. O conceito envolvendo o contexto de aplicação do sistema de monitoramento é ilustrado na Figura 1.1.

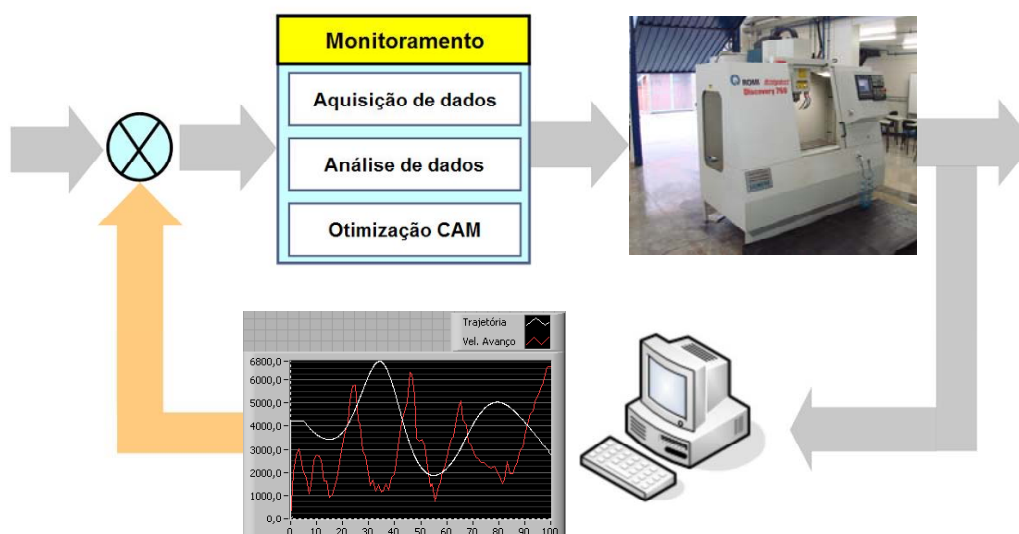


Figura 1.1: Conceito do sistema monitoramento

Na Figura 1.1 pode ser vista a associação dos componentes envolvidos com o conceito de um sistema de controle retroalimentado. O diagrama de blocos esquematizado representa a aplicação do monitoramento do processo de usinagem com o objetivo de desempenhar a função de uma interface que possibilite o acesso e a análise dos dados da máquina provindos do CNC de arquitetura aberta. A análise dos sinais que são disponibilizados para comunicação caracteriza o comportamento da máquina-ferramenta, que é influenciado pelos parâmetros de programação CAM.

O sistema de monitoramento obtém os sinais da máquina-ferramenta que são gerenciados pelo CNC de arquitetura aberta. Esses dados estão contidos em variáveis que são utilizadas no processamento de dados do CNC e disponibilizadas para acesso.

## 1.1 A importância deste trabalho

A importância deste trabalho está fundamentada no seguinte cenário:

- Atualmente, não existe um padrão estabelecido para o controlador de arquitetura aberta que seja amplamente aceito e utilizado pelas indústrias;
- O monitoramento da velocidade de avanço e aceleração dos eixos no processo de usinagem HSM permite identificar os parâmetros ótimos de programação NC;
- A avaliação da capacidade da arquitetura aberta de um CNC comercial para a implementação de sistemas de monitoramento é importante para o setor industrial, na medida em que são identificadas as limitações de customização de funções e oportunidades de otimização do processo de usinagem através da utilização de sistemas de monitoramento.

Com base no cenário acima, entende-se que o conceito de arquitetura aberta é o meio pelo qual a otimização do processo pode ser obtida, sendo o sistema de monitoramento de dados uma interface de avaliação do desempenho do processo, cujo comportamento é influenciado pela programação NC.

Paralelamente à otimização do processo, o desenvolvimento e a implementação de um sistema de monitoramento de dados permite a avaliação do nível de acesso de dados e de customização de funções de um CNC de arquitetura aberta comercial.

## 1.2 Estrutura do trabalho

A estrutura deste trabalho é composta de sete capítulos que são apresentados a seguir com uma breve abordagem do respectivo conteúdo destes.

### Capítulo 1 – Introdução

**Capítulo 2 – Monitoramento do Processo de Usinagem HSM** – Revisão bibliográfica sobre o monitoramento do processo de usinagem HSM, abordando a tecnologia de usinagem HSM, sistemas de monitoramento do processo de usinagem, comando numérico de arquitetura aberta, sistemas baseados em computador e protocolos de comunicação industrial.

**Capítulo 3 – Proposta do Trabalho** – Apresentação dos objetivos deste trabalho e a metodologia que determina como serão conduzidos os experimentos e a análise dos dados.

**Capítulo 4 – Desenvolvimento dos ensaios** – Desenvolvimento do sistema de monitoramento e execução dos ensaios conforme premissas elaboradas no Capítulo 3.

**Capítulo 5 – Análise dos resultados** – Nesse capítulo serão apresentadas a análise dos ensaios preliminares e da taxa de aquisição e resolução da variável monitorada, a identificação dos parâmetros ótimos de programação NC e a avaliação do nível de acesso e de customização de funções do CNC de arquitetura aberta.

**Capítulo 6 – Conclusões e sugestões para trabalhos futuros.**

**Capítulo 7 – Bibliografia Referenciada** – Lista das referências utilizadas durante o desenvolvimento deste trabalho.

## 2 Monitoramento do Processo de Usinagem HSM

A otimização do processo de manufatura nas indústrias de moldes e matrizes, aeroespacial e automobilística é obtida com o emprego da tecnologia de usinagem HSM. O aumento da velocidade de corte do processo de usinagem resulta na diminuição das forças de corte e da temperatura do processo. Superfícies de melhor qualidade são obtidas eliminando-se total ou parcialmente as operações seguintes de acabamento superficial. Estas características são apresentadas na Figura 2.1 [16].

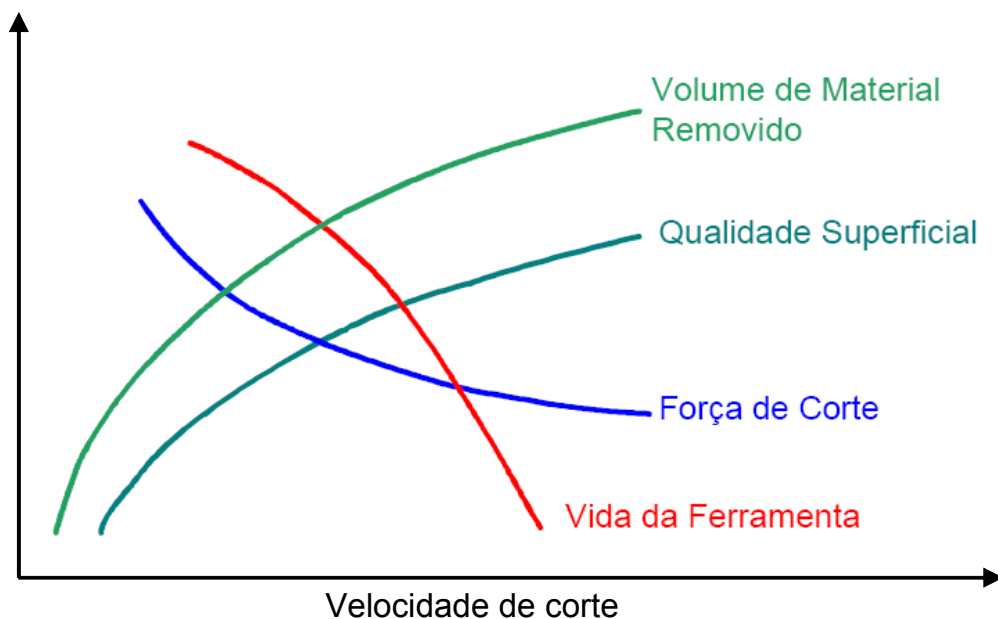


Figura 2.1: Características do processo HSM [16]

Conforme ilustrado na Figura 2.1, a otimização do processo de usinagem é obtida com o aumento do volume de material removido e pela melhor qualidade superficial da peça, porém, o tempo de vida da ferramenta diminui.

O contexto tecnológico que proporcionou o desenvolvimento e aplicação do processo HSM é apresentado na Figura 2.2.

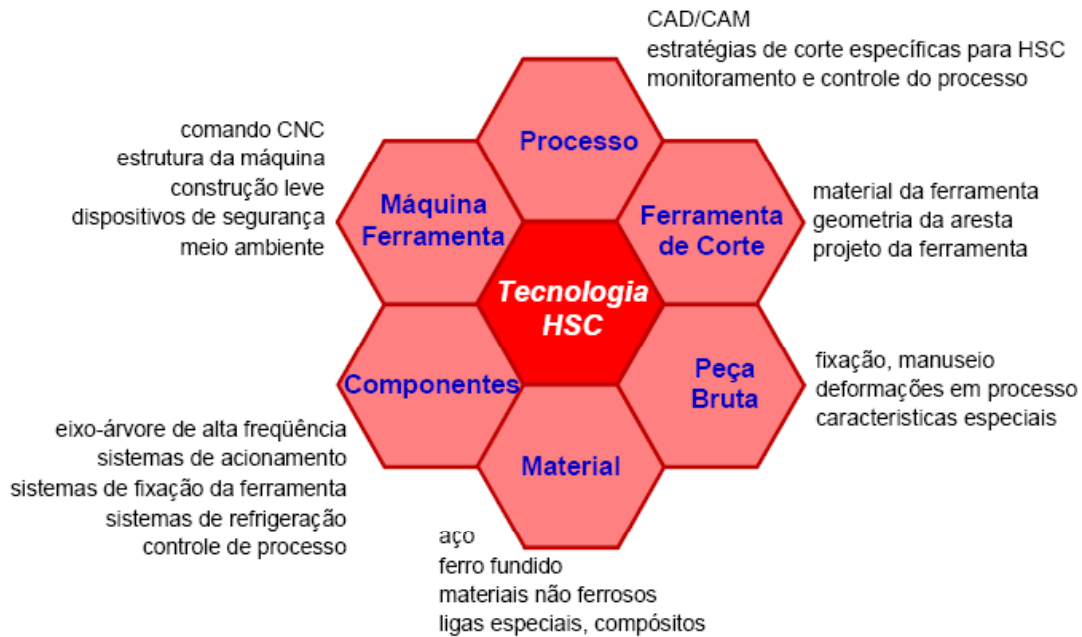


Figura 2.2: Tecnologias envolvidas no processo HSM [16]

Na Figura 2.2, pode ser observado que para a aplicação plena desta tecnologia, necessitou-se de avanços em componentes relacionados ao processo e a máquina-ferramenta. Destacam-se as estratégias de corte específicas para HSM, o CNC de alta performance, os sistemas de monitoramento e controle do processo, sistemas de acionamento, eixo-árvore de alta frequência e ferramentas de corte [16].

O conjunto de tecnologias de componentes, máquina-ferramenta e processo, apresentados na Figura 2.2 tem como finalidade principal a determinação dos movimentos da máquina ferramenta e a remoção de material que resultarão na peça acabada. Esses movimentos são conhecidos como sendo a trajetória da ferramenta de corte, que na usinagem de superfícies complexas com altas velocidades devem ser necessariamente especificadas e calculadas pelos sistemas CAM.

A trajetória da ferramenta de corte é determinada por métodos de interpolação. Na usinagem HSM o método interpolação linear pode alterar o comportamento dinâmico da máquina devido à descontinuidade gerada nas junções dos segmentos de reta, diminuindo significativamente a velocidade de avanço. Esta situação é evitada com a aplicação da interpolação polinomial que assegura a continuidade da trajetória, evitando limitações de tempo de processamento e gerando resultados como maior velocidade média de avanço [1].

Os seguintes autores [1, 17, 18, 19, 20, 21] abordam as influências das metodologias de interpolação na usinagem HSM. Nesses trabalhos pode-se

constatar que o comportamento da velocidade de avanço altera, assim como o tempo de usinagem, ao se aplicar diferentes métodos de interpolação. Tais fatores demonstram que com o monitoramento da velocidade de avanço, da aceleração dos eixos e do tempo de usinagem pode-se identificar os parâmetros de programação que resultem na otimização do processo.

As variações na velocidade de avanço durante a usinagem de superfícies complexas são influenciadas pelos seguintes fatores:

- Limitação dinâmica da máquina: o CNC limita a velocidade de avanço conforme parâmetros pré-determinados para a garantia da estabilidade dinâmica do processo [22];
- Tempo de processamento de bloco do CNC: se o tempo para a movimentação for menor que o tempo necessário para processamento do bloco, o sistema de movimentação reduz a velocidade de avanço, para que seja possível o processamento da informação [22].

Segundo Altintas e Erkokmaz [17], o desenvolvimento de uma técnica para otimização da velocidade de avanço na interpolação *spline* proporcionou a máxima utilização de recursos físicos dos *drives* e a redução do tempo de usinagem em alta velocidade.

A Figura 2.3 ilustra o *loop* de controle do processo de usinagem CNC que está inalterado há mais de vinte anos [23].

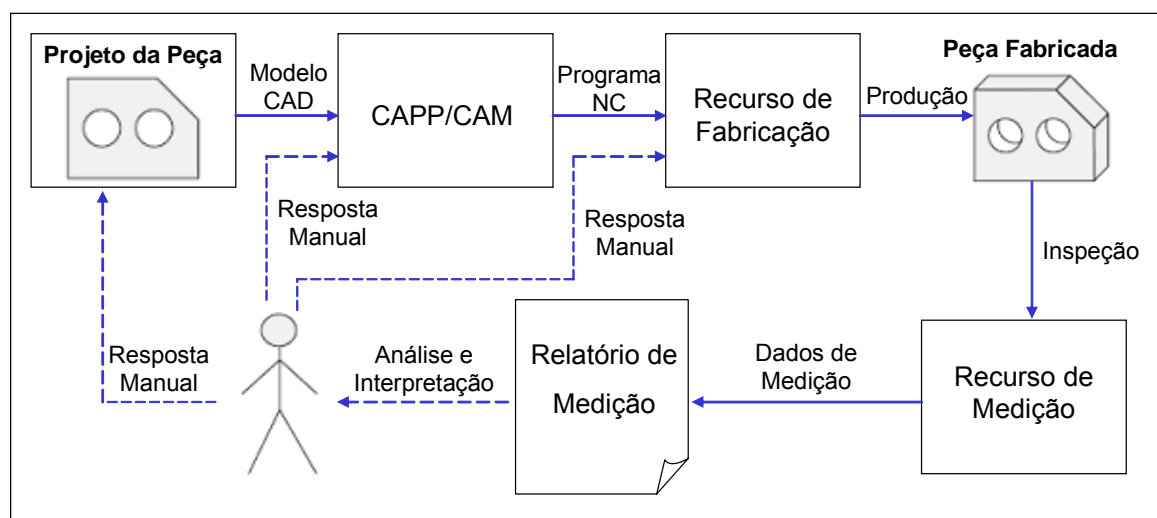


Figura 2.3: Estrutura atual de controle na manufatura CNC [23]

Na Figura 2.3, observa-se que a correção de parâmetros de programação CAM é feita de forma manual, e a fonte de informação que o usuário detém para a análise do desempenho da programação NC é o relatório de medição geométrica. Com



este tipo de análise não é possível identificar a influência dos métodos de interpolação no comportamento da velocidade de avanço e no tempo de usinagem, tornando necessário o desenvolvimento de um sistema que execute o monitoramento dessas variáveis, trate os dados e mostre-os ao usuário com uma interface de visualização adequada.

Mesmo com os CNC baseados em computadores pessoais, falta uma estrutura de comunicação entre as máquinas ferramentas e os sistemas computacionais de apoio a manufatura, principalmente com os sistemas CAM, que tem relação direta com o desempenho do processo de usinagem [23].

O tempo de usinagem calculado pelos sistemas CAM diverge do tempo real, pois no sistema CAM não são considerados os tempos de aceleração e desaceleração e de distância média a ser percorrida entre uma instrução programada e outra. Estudos prévios comprovam erros superiores a 36% em relação ao tempo calculado e real quando a velocidade de avanço programada é superior a 3 m/min [24].

Monreal [18], propôs um modelo para a predição do tempo de usinagem no processo HSM, considerando que os diferentes ângulos de orientação da trajetória da ferramenta em *zig-zag* e os altos valores de velocidade de avanço programados influenciam no tempo real de usinagem.

O caminho da ferramenta de corte pode ser otimizado com o uso do algoritmo de interpolação *spline*. Com esse algoritmo, define-se um parâmetro correspondente ao erro máximo de trajetória permitido e, baseado nesse erro, o algoritmo otimiza a trajetória da ferramenta com o uso da interpolação *spline* parametrizada, flexibilizando a otimização da programação NC para as operações de desbaste e acabamento [19].

Um novo método para a integração da experiência do operador na programação NC foi apresentado por Schulz [25]. Nesse trabalho a descrição do processo é feita pelas *features* de manufatura. As *features* são elementos de construção utilizados pelos sistemas CAD para a definição do modelo geométrico do produto e as *features de manufatura* contém informações sobre o processo de usinagem (tipo de ferramentas, estratégias de usinagem, velocidade de corte, etc). A integração do conhecimento do operador é feita através do uso de um modelo central de informação, desenvolvido com a linguagem descritiva *EXPRESS*.

A falta de um sistema para análise da influência da programação NC no processo de usinagem com altas velocidades direcionam a pesquisa para o desenvolvimento de um sistema de monitoramento de dados aplicado a um CNC

de arquitetura aberta comercial. Os dados de velocidade de avanço, de aceleração dos eixos e do tempo de usinagem foram escolhidos para serem monitorados, pois foi constatado que os métodos de interpolação modificam o comportamento desses dados. Dessa forma, alterando-se o método de interpolação e monitorando-se as variáveis descritas, pode-se identificar o método de interpolação que resulte num processo otimizado.

Em estudos já realizados [26, 27] é verificado que os métodos de interpolação podem influenciar na exatidão geométrica e na qualidade superficial da peça usinada. No presente trabalho essas influências não serão analisadas.

Os dados a serem monitorados são obtidos através de um CNC de arquitetura aberta comercial e não faz parte dos objetivos desse trabalho avaliar a exatidão dos dados fornecidos pelo CNC.

## **2.1 Monitoramento e controle do processo de usinagem**

O desenvolvimento do Comando Numérico – CN na década de 1950s pelo laboratório de servomecanismos do MIT (*Massachusetts Institute of Technology*) foi o marco inicial da automação de máquinas. Na sua concepção estão os conceitos básicos de controle e monitoramento do processo que aliados ao alto desempenho de processamento dos computadores obtido nos anos 1970s e 80s possibilitaram a implementação de controladores nos computadores, surgindo o Comando Numérico Computadorizado – CNC [28].

Desde o surgimento do CNC, grandes avanços foram obtidos nos mecanismos servocontrolados e na interpolação dos eixos. Os movimentos controlados pelo CNC geram o processo de corte e nas últimas décadas uma grande quantidade de pesquisas foram direcionadas para o monitoramento e controle deste processo [29].

As coordenadas de movimentação e os demais parâmetros do processo de usinagem estão contidos no programa NC. O CNC utiliza esses dados para a execução do processo de usinagem, onde surgem as variáveis de processo (força de corte, temperatura, etc.). De uma forma geral, os sistemas de monitoramento e controle utilizam o estado dessas variáveis em algoritmos de controle para atuação no processo, como pode ser observado no esquema da Figura 2.4 [28].

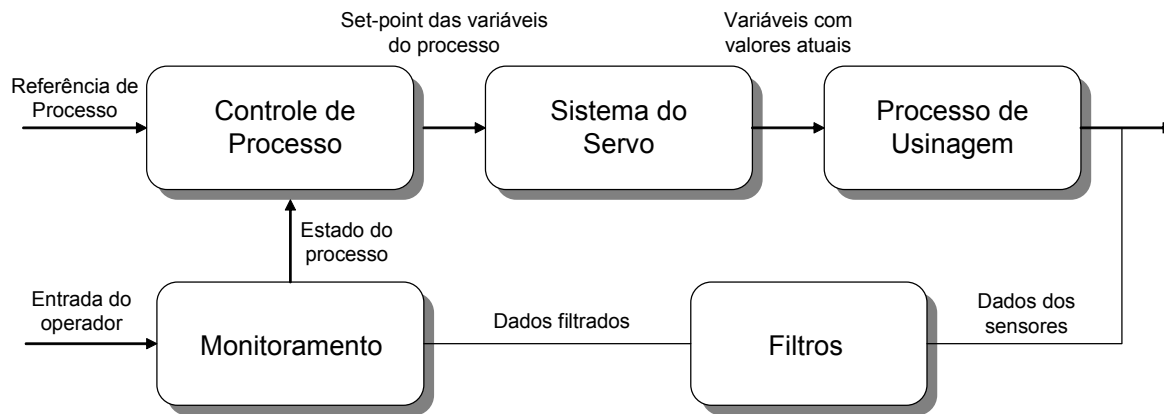


Figura 2.4 : Esquema do monitoramento e controle de usinagem CNC [28]

O contexto acima proporcionou o interesse de pesquisadores na implementação de funções de controle de processo de alto nível no CNC, conhecido como controle adaptativo. O controle adaptativo tem o objetivo de aumentar o volume de produção e a qualidade final dos produtos manufaturados. Seu funcionamento se baseia na medição e manipulação de variáveis do processo. Com a análise do comportamento do processo, obtêm-se as condições ótimas de operação [30].

As pesquisas de monitoramento do processo de usinagem tiveram e ainda têm em sua grande parte como foco principal o desenvolvimento de algoritmos e modelos matemáticos para o tratamento e análise de sinais. Estes sinais, que são obtidos por um sensor ou múltiplos sensores, são tratados pelos algoritmos e modelos matemáticos com a finalidade de extração de determinadas características que permitam a identificação do comportamento do processo [31, 32].

O monitoramento do processo de usinagem também é utilizado para a detecção e a correção de erros durante a execução do processo, evitando-se que o lote de peças seja fabricado fora das especificações de projeto [31].

No trabalho de Valente et al. [34] é proposto uma modificação na arquitetura de um CNC aberto para permitir a adição de sensores e disponibilizar os dados dos sensores adicionados e dados do CNC aberto para monitoramento remoto via rede de comunicação. A transmissão de dados via rede apresentou baixo *delay* (atraso), porém o atraso de comunicação não foi quantificado. A comunicação foi estabelecida pela rede Ethernet TCP/IP e protocolo proprietário do fabricante do CNC.

O desgaste e a quebra da ferramenta de corte são fatores que direcionam grande parte dos estudos de monitoramento do processo de usinagem. No entanto, a diversidade de máquinas e operações geralmente não é considerada nessas

pesquisas, além da grande complexidade desses sistemas, que dificultam sua aplicação no ambiente industrial [35].

Esta situação não é particularidade do monitoramento do estado da ferramenta de corte, e também é constatada nos demais trabalhos da área de monitoramento de processos de usinagem. Uma das alternativas, que visa permitir o acesso aos dados da máquina de forma padronizada para a implementação de algoritmos de controle, é o conceito de controlador de arquitetura aberta que será detalhado no tópico 2.2.2.

### 2.1.1 Aquisição de dados

Os dados que são utilizados para o monitoramento e o controle do processo de usinagem são obtidos através de sensores. O sensor tem a função de medir o estado da variável monitorada e disponibilizar o dado para comunicação com o sistema. No presente trabalho não serão adicionados sensores na máquina ferramenta, pois os dados a serem monitorados são do CNC aberto.

O Instituto dos Engenheiros Elétricos e Eletrônicos (IEEE) definiu o padrão IEEE 1451, denominado de interface inteligente de transdutores, para o monitoramento e controle de máquinas com o objetivo de estabelecer um conjunto de interfaces comuns de comunicação para conexão de sensores e atuadores inteligentes em um ambiente de rede independente. É basicamente dividida em duas partes, a primeira define uma interface padrão de *hardware* para conexão de transdutores com microprocessadores, redes, ou sistema de instrumentação. A segunda, define uma interface padrão de *software* para a conexão de transdutores em diferentes redes de comunicação via API (*Application Programming Interface*) em relação aos transdutores, e via interface lógica de protocolo de rede em relação à comunicação via rede [36].

A idéia principal do IEEE 1451 é a neutralidade na conexão via rede e a independência em relação a um vendedor. Para facilitar a migração e utilização no ambiente industrial, não existem requisitos quanto ao uso de diferentes conversores analógico digital, microprocessadores e protocolos de rede, o que permite os fabricantes adicionar flexibilidade aos produtos. Com esta série de requisitos pretende-se também obter a integração e a interoperabilidade dos transdutores via rede. Na Figura 2.5 é apresentada a família de normas IEEE 1451 [36].

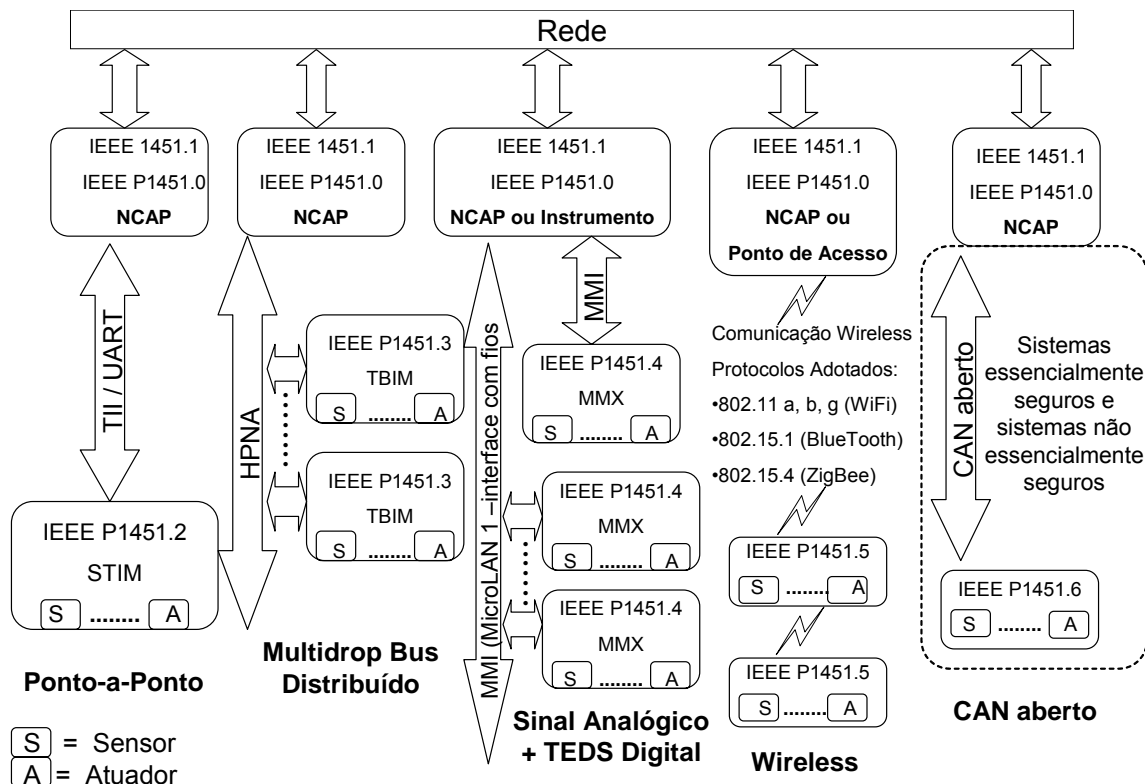


Figura 2.5 : Família de normas IEEE 1451 [36]

A arquitetura de comunicação da máquina-ferramenta está estruturada com a centralização da tomada de decisões pelo CNC. Desforges [37] apresenta uma proposta para viabilizar a descentralização do controle com o uso de sensores e atuadores inteligentes. Para habilitar o nível de processo com autonomia para a execução do controle do processo, desenvolveu-se uma estrutura de comunicação com um nível de decisão que é responsável pela ligação entre os níveis operacionais (Interface homem-máquina, CNC) e de processo (sensores e atuadores). Este nível de decisão permite que os sensores e atuadores inteligentes executem o controle do processo de maneira individual ou em cooperação com os demais componentes do sistema.

Nos processos de usinagem, os métodos de monitoramento estão classificados em dois tipos: direto e indireto. O relacionamento entre variável medida e variável monitorada determina o tipo de monitoramento e implica nas seguintes diferenças na aquisição do sinal [38]:

- O método direto, como o próprio nome expressa, faz a medição direta da variável monitorada. Como exemplo tem-se o monitoramento do desgaste da ferramenta de corte, onde a medição do desgaste é executada através da inspeção visual ou com sensores de visão;
- O método indireto mede variáveis que estão relacionadas com a variável de interesse. Neste caso, pode-se utilizar o exemplo anterior, porém com

---

a medição da potência elétrica consumida pelo eixo-motor, cujo valor é relacionado com o aumento do desgaste da ferramenta de corte.

Os sistemas de monitoramento tanto diretos como indiretos, podem ser implementados através de duas abordagens. A primeira considera a adição de sensores no processo a ser monitorado e, a segunda, chamada de *sensorless*, não necessita de adição de sensores pois os sinais são obtidos de sensores já presentes na máquina ferramenta. Para determinados sinais, como o sinal de corrente do *driver*, deve-se fazer o tratamento dos dados com filtros para eliminação de distúrbios que influenciam no sinal monitorado [39].

### 2.1.2 Estratégias de tomada de decisão

A decisão a ser tomada pelo sistema de monitoramento está baseada no estado da variável monitorada. Esse estado é extraído do sinal da variável, onde busca-se identificar os comportamentos que proporcionarão a tomada de decisões. As estratégias de tomada de decisão variam desde formas simplificadas como valores de limites máximo e mínimo até técnicas de aprendizado que detectam condições anormais de operação [40].

Os sistemas que monitoram o desgaste da ferramenta de corte utilizam a correlação de sinais monitorados com o estado da ferramenta, pois com o aumento do desgaste aumenta-se também a força necessária para execução do corte. Neste caso, pode-se utilizar a média da força necessária para o corte do material como variável para a representação de características do nível de desgaste da ferramenta [41].

Em estudos já realizados [42] o deslocamento do eixo-árvore foi utilizado para a elaboração de um modelo de previsão da rugosidade superficial obtida no processo de fresamento. O deslocamento do eixo-árvore foi medido durante a usinagem e utilizado para elaboração do modelo de regressão linear que foi validado com testes utilizando diferentes condições de corte. Os resultados obtidos demonstram que o modelo desenvolvido consegue fazer a previsão da rugosidade que será obtida na peça, de acordo com os parâmetros de corte adotados.

No processo de torneamento, a compensação automática do desgaste da ferramenta é implementada com o monitoramento do diâmetro da peça que está sendo usinada. Durante a execução da usinagem, uma rotina é acionada para medir o diâmetro da peça, essa medição é obtida com a utilização de um sensor que, ao tocar a peça, envia um sinal para o CNC gravar o posicionamento dos eixos da máquina obtendo-se a dimensão da peça. Comparando-se a dimensão obtida com o valor de *set point* calcula-se o erro correspondente ao desgaste da

ferramenta que é corrigido automaticamente, porém o limite máximo de desgaste da ferramenta deve ser informado pelo operador da máquina [43].

Na Figura 2.6 é mostrado o algoritmo de tomada de decisão do sistema descrito acima. O parâmetro CPW corresponde à somatória das correções equivalentes ao desgaste da ferramenta. O CPWmax é informado pelo operador da máquina e os parâmetros LSmax e LSmin significam respectivamente, peça com dimensão acima do limite máximo e abaixo do limite mínimo de especificação. Os parâmetros LCmax e LCmin correspondem ao limite superior e limite inferior de controle da variação natural de dimensão da máquina ferramenta.

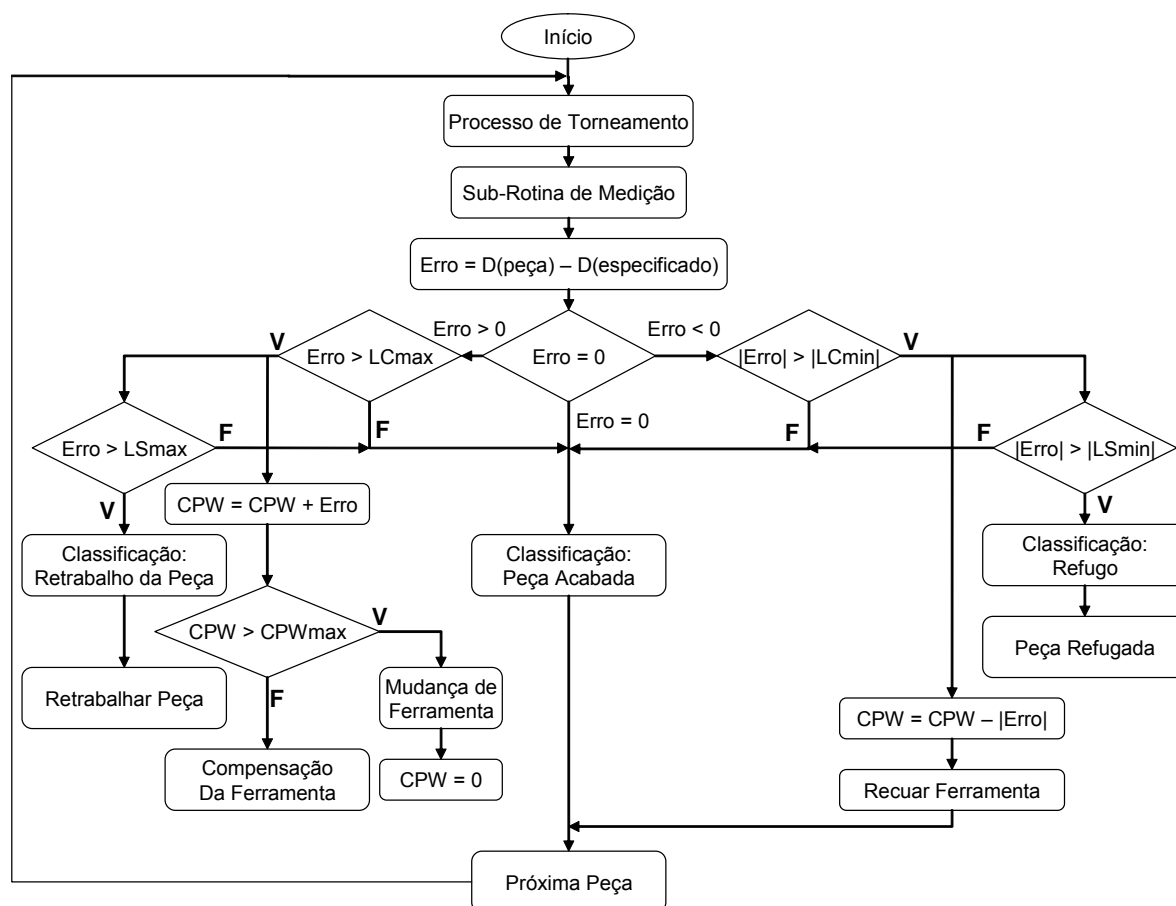


Figura 2.6: Algoritmo do monitoramento de desgaste da ferramenta [43]

Como mostrado na Figura 2.6, através da sub-rotina de medição compara-se o diâmetro da peça com o diâmetro especificado em projeto e determina-se o erro que pode ser menor, igual ou maior que zero, retornando os respectivos status: peça refugada, peça acabada e peça para retrabalho, onde são tomadas as devidas ações.

Um procedimento para a construção de um mapa de emissão acústica para o monitoramento do processo de retificação foi apresentado por Silva et al. [44]. No

seu trabalho a imagem do processo é construída em tempo real durante a operação de dressagem do rebolo. Essa imagem representa o estado da superfície do rebolo e pode ser analisada através da IHM do CNC aberto. Com esse sistema pode ser identificados ciclos ineficientes de dressagem, desbalanceamento, vibrações, etc.

### 2.1.3 Níveis de controle do processo de usinagem CNC

O controle do processo na máquina-ferramenta é constituído do PLC (*Programmable Logic Controller*), que gerencia funções de operação da máquina e do microprocessador, que executa funções de controle em tempo real. O controle no microprocessador é geralmente dividido em três níveis, que são: controle do servomecanismo, do interpolador e controle do processo [29]. Essa arquitetura é mostrada na Figura 2.7.

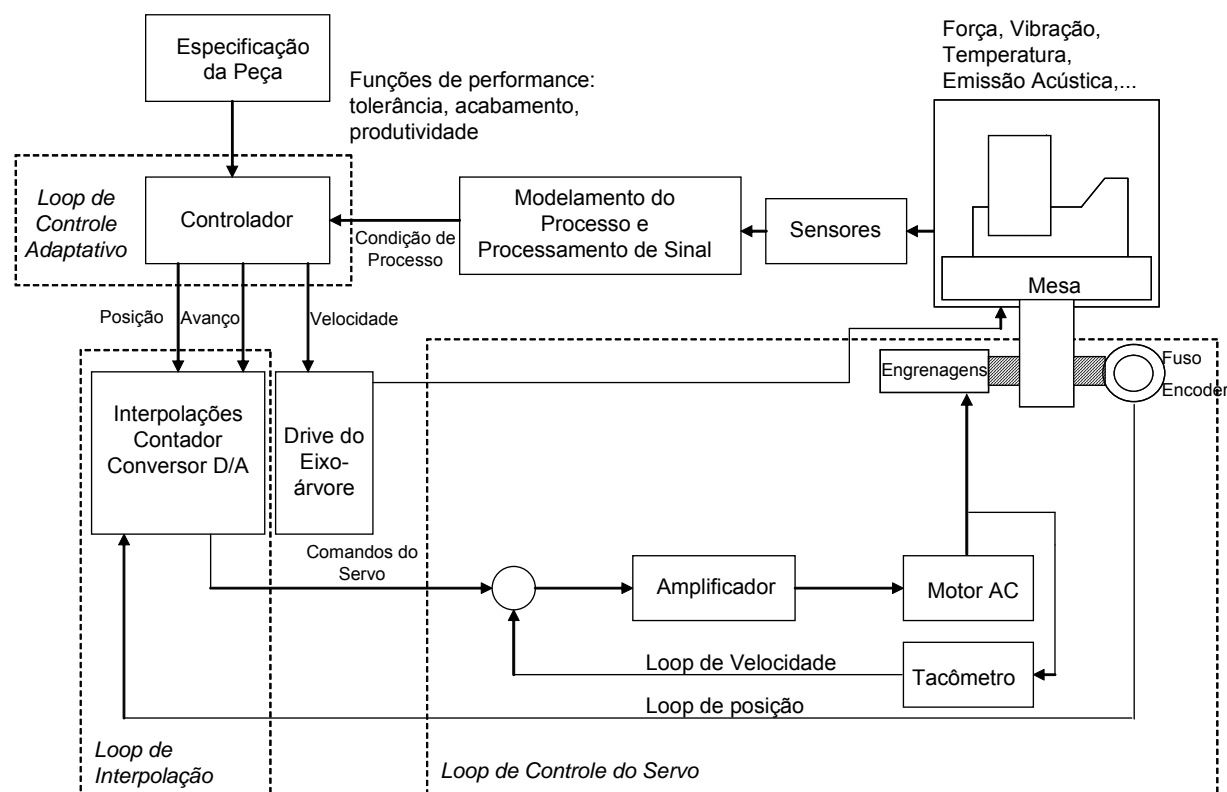


Figura 2.7 : Arquitetura de controle da máquina-ferramenta CNC [29]

A estrutura da arquitetura de controle ilustrada na Figura 2.7 é constituída pelo nível de controle do servo, pelo interpolador e pelo nível de controle do processo, que são abordados nos tópicos seguintes.



### 2.1.3.1 Controle do servo

O controle de movimentos dos eixos da máquina é executado pelo nível de controle do servo. Este é considerado o nível mais baixo do processo de usinagem CNC. Neste nível são executados os controles de posição e velocidade individual dos eixos, cujos valores de referência são informados pelo nível de controle do interpolador, descrito no próximo sub-tópico. O controle de posicionamento dos eixos é classificado em dois tipos: ponto a ponto e de contorno. No controle ponto a ponto somente as coordenadas dos pontos são controladas, sendo usado quando a velocidade de deslocamento entre os pontos não influenciam na qualidade da peça, pois não ocorre o processo de corte de material. Como exemplo tem-se o posicionamento da broca na operação de furação. No controle de contorno, a trajetória da ferramenta e a velocidade são controladas, pois nesse caso deseja-se obter geometrias usinadas, como ocorre no processo de fresamento [28]. Na Figura 2.8 é apresentado o *loop* de controle de trajetória.

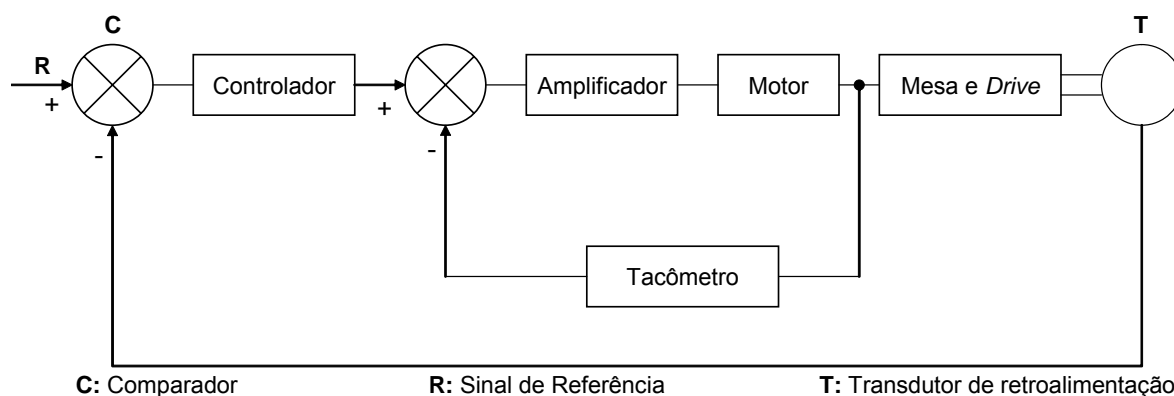


Figura 2.8: Loop de controle de trajetória [45]

O erro de contorno é um dos principais fatores para a avaliação do desempenho no controle de movimentos. Este erro é definido pela diferença entre trajetória programada e executada. Para a obtenção de contornos com elevada exatidão geométrica os valores de ganho dos controladores do servo devem ser corretamente ajustados para proporcionar ao sistema o comportamento de criticamente amortecido. Este comportamento é necessário devido as altas velocidades de deslocamento, que contribuem para o aumento do erro de contorno [45].

O controle do servo tem uma característica inerente aos sistemas de controle retroalimentados conhecida como *servo-lag*. O loop do servo trabalha com a diferença entre a posição de *set point* informada pelo programa NC e a posição atual da máquina, que é informada pela retroalimentação dos sinais do *encoder*. O

*servo-lag* é a diferença entre a posição desejada e a posição real e gera erros de posicionamento principalmente na usinagem de contornos [33]. A Figura 2.9 ilustra a influência do *servo-lag* na exatidão da trajetória da ferramenta.

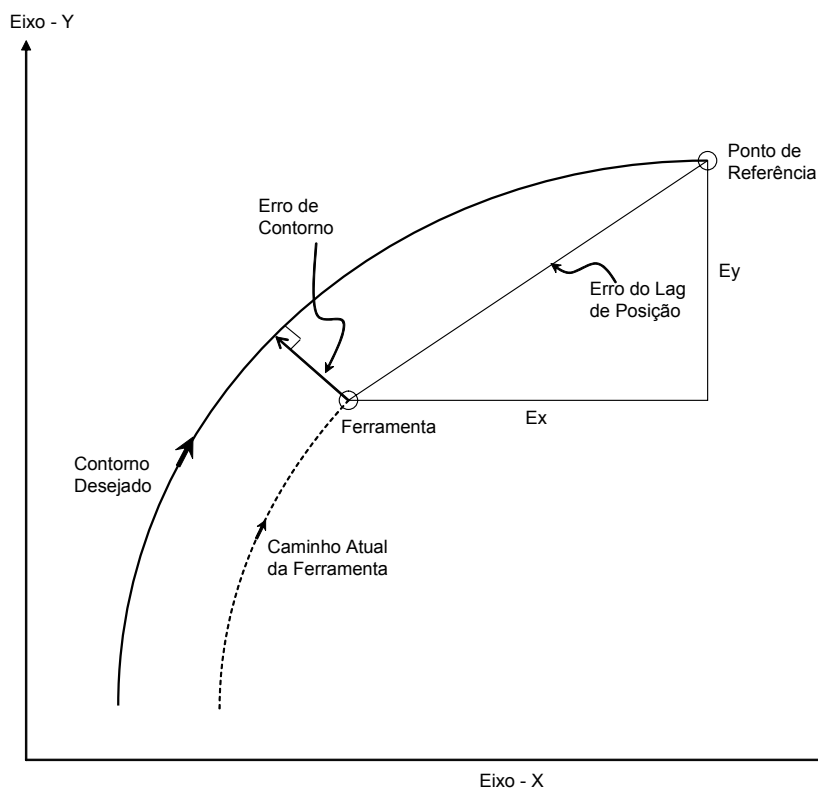


Figura 2.9: Influência do *servo-lag* na exatidão da trajetória da ferramenta [28]

Conforme ilustrado na Figura 2.9, o erro devido ao *lag* (defasagem) de posicionamento é responsável pela geração do erro de contorno. A redução do erro de contorno pode ou não significar o mesmo que a redução de erro individual dos eixos  $E_x$  e  $E_y$  [28]. No próximo tópico é abordado o interpolador e a tendência no desenvolvimento de algoritmos de interpolação de movimentos.

### 2.1.3.2 Interpolador

O movimento descrito pela ferramenta de corte é resultado dos movimentos individuais dos eixos da máquina. O Interpolador tem a função de descrever a informação contida no programa NC em coordenadas de referência para os eixos. Estas coordenadas determinam qual será o movimento a ser executado por cada eixo para que seja obtida a trajetória que a ferramenta deve percorrer para a usinagem da peça [46].

Segundo Cheng et al. [20], devido à estrutura convencional da cadeia CAD/CAM/CNC é necessária a segmentação dos segmentos de curva em segmentos de reta/arcos, pois no CNC convencional a interpolação dos eixos é

executada através de interpolações lineares e circulares. Devido à isso, a curva gerada no sistema CAD é segmentada no sistema CAM para o cálculo da trajetória da ferramenta [20].

Os principais motivos para a implementação de novos algoritmos para a substituição do interpolador linear/circular são:

- Eliminação de erros causados por ruídos e perda de informação na transmissão de elevada quantidade de dados entre os sistemas CAD/CAM/CNC [47];
- Perda de exatidão geométrica gerada pela descontinuidade da segmentação de curvas em segmentos de reta e arcos [47];
- Velocidade e aceleração não uniforme e sem suavização [47].

Para evitar as limitações causadas pela interpolação linear/circular são propostas modificações na cadeia CAD/CAM/CNC, principalmente no interpolador, conforme ilustra a Figura 2.10 abaixo [20].

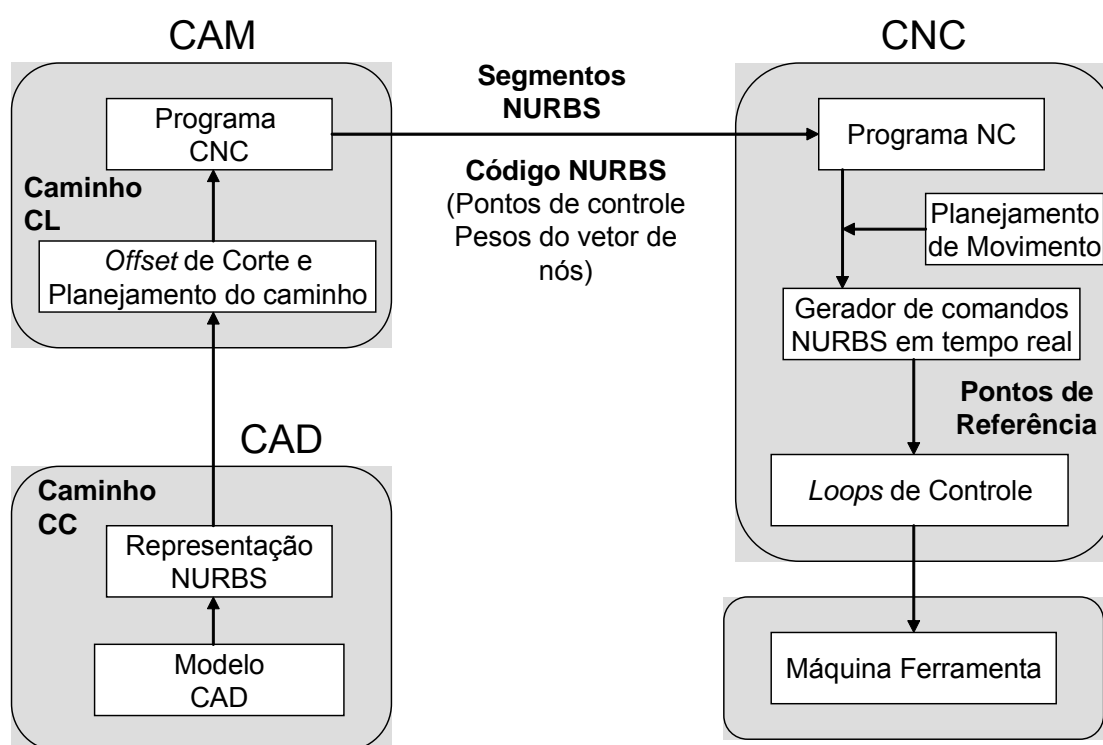


Figura 2.10: Proposta de cadeia CAD/CAM/CNC para a usinagem de curvas [20]

Conforme visto na Figura 2.10, a implementação de um interpolador NURBS de tempo real no CNC é uma proposta para eliminação das limitações da interpolação linear/circular e para viabilizar a integração da cadeia CAD/CAM/CNC, propõe-se a representação em NURBS do modelo CAD e a utilização de segmentos de curva em NURBS para o cálculo da trajetória da

ferramenta nos sistemas CAM. Essa proposta não inclui a influência do erro de *lag* na exatidão da trajetória da ferramenta.

Altintas e Erkorkmaz [17] otimizaram em até três vezes o tempo de ciclo de usinagem HSM com a utilização de uma técnica para otimizar a velocidade de avanço na interpolação *spline*. No seu trabalho, a capacidade total dos *drives* foi explorada, respeitando seus limites máximos de velocidade, aceleração e *jerk* (taxa de variação da aceleração). Através da *spline* de quinta ordem, a continuidade dos perfis de velocidade, aceleração e *jerk* foram mantidos e o perfil de velocidade de avanço foi otimizado com a manipulação do tempo necessário para a execução dos segmentos.

### 2.1.3.3 Controle do processo

Segundo Liang et al. [29], na terminologia de máquinas-ferramenta o termo controle do processo se referencia ao controle adaptativo. Atualmente, o controle adaptativo é encontrado como opcional em alguns CNCs modernos e devido aos bons resultados no aumento da qualidade e produtividade, este tema é abordado por uma quantidade significativa de pesquisas. A otimização do processo proveniente do controle adaptativo é obtida através da modificação dos parâmetros que influenciam na usinagem e sua classificação é apresentada abaixo:

- Controle adaptativo com limites: neste tipo de controle os parâmetros do processo são monitorados em tempo real e manipulados para manter determinada variável (ex: força de corte) dentro de limites pré-estabelecidos. Este tipo de controle é indicado para operações de desbaste devido à possibilidade de correlação da taxa de remoção do material com limites pré-estabelecidos de força de corte [29];
- Controle adaptativo com otimização: visa à otimização de indicadores de produção (ex: *lead time*, custo unitário) através da seleção e ajuste de parâmetros do processo (velocidade de avanço, rotação do eixo motor). Os parâmetros ótimos são obtidos de modelos matemáticos e de sinais de resposta do processo [29];
- Controle adaptativo geométrico: é aplicado em operações de acabamento com a finalidade de manter a qualidade da peça dentro das especificações de projeto. Os parâmetros de processo que afetam a qualidade da superfície usinada são correlacionados com parâmetros de qualidade (ex: velocidade de avanço correlacionada com rugosidade) e os valores dos parâmetros do processo são ajustados [29].

#### 2.1.3.4 Controle de força

O processo de corte do material é resultado das forças desenvolvidas entre a ferramenta e a peça durante a usinagem. A força necessária para o corte do material está diretamente relacionada com o estado da ferramenta de corte e com a deformação do conjunto máquina/peça/ferramenta, tornando o controle da força de corte um importante meio para a otimização do processo [48].

Li [48] apresentou um novo método para a medição da força de corte com o uso de sensores de baixo custo, pois uma das barreiras para a aplicação deste controle no ambiente industrial é o elevado custo dos instrumentos de medição de força. Nesse novo método, as correntes do eixo-motor e *servo* motor são medidas com sensores de efeito *hall* e os valores obtidos são utilizados para o cálculo do torque. Os valores de torque e corrente de armadura são utilizados pelos modelos de força de corte para a determinação das forças tangencial, axial e radial de corte. A relação de custo/performance demonstrou que com baixo custo, é possível a implementação do sistema para o monitoramento do estado da ferramenta e controle da força de corte no processo de torneamento.

Milfeiner et al. [49] utilizaram um dinamômetro para a medição da força de corte e monitoramento da quebra da ferramenta no processo de fresamento. Neste trabalho foi considerado que a força de corte é proporcional a área da seção do material removido, e que durante uma rotação completa da ferramenta o volume de material removido é constante. Partindo-se dessa premissa, considerou-se que ao quebrar um dente da ferramenta, o volume de material removido neste local será menor, diminuindo a força de corte, porém o dente subsequente removerá um volume maior de material aumentando o valor da força de corte. Com a análise do sinal de força pode-se identificar que tal quebra ocorreu.

#### 2.1.3.5 Controle de vibração

A vibração gerada entre a ferramenta e a peça durante o corte do material é uma das principais causas de perda de performance da máquina ferramenta. Esta perda de performance não somente limita a produtividade como também influencia em perda de qualidade do acabamento superficial, aumento do desgaste da ferramenta, aumento de ruídos no ambiente de trabalho e diminuição do tempo de vida da máquina-ferramenta [50].

Com o propósito de eliminar a vibração no processo de usinagem Al-Regib [50] desenvolveu um procedimento para a determinação da amplitude e da frequência de variação da velocidade do eixo-motor. Neste procedimento, considera-se que a variação da velocidade de rotação do eixo-motor influencia na espessura do

cavaco e conseqüentemente na força de corte. Calcula-se então a amplitude e freqüência necessária para diminuir as vibrações regenerativas.

Toh [51] utilizou a transformada rápida de Fourier para análise de vibração nas operações de desbaste e acabamento no processo HSM do aço endurecido AISI H13. A transformada rápida de Fourier foi aplicada nos sinais de força de corte obtidos de um dinamômetro e os gráficos dos espectros de freqüência foram analisados para identificação da influência da estratégia de corte e do desgaste da ferramenta no sinal de vibração do processo.

O modelo para o controle de vibrações proposto por Altintas e Ko [52] considera, além dos fatores dinâmicos do processo, a função da resposta em freqüência da ferramenta de corte fixada no mandril e montada na máquina ferramenta. A resposta em freqüência foi obtida através de um acelerômetro fixado num dente da ferramenta, que mediu a vibração do sistema resultante do impacto com um martelo no outro dente. Os valores de profundidade de corte e RPM que criticamente influenciam na vibração do sistema foram identificados.

#### 2.1.4 Arquitetura de controle distribuído

A arquitetura de controle distribuído descentraliza a tomada de decisões que tradicionalmente são executadas por um controlador dedicado a determinado processo. Com ele, o controle do sistema de manufatura passa a ser desempenhado por computadores ou dispositivos inteligentes que estão distribuídos entre os componentes do processo e são interligados por uma rede de comunicação digital (ex: *fieldbus*). Esta arquitetura é ilustrada na Figura 2.11 [53].

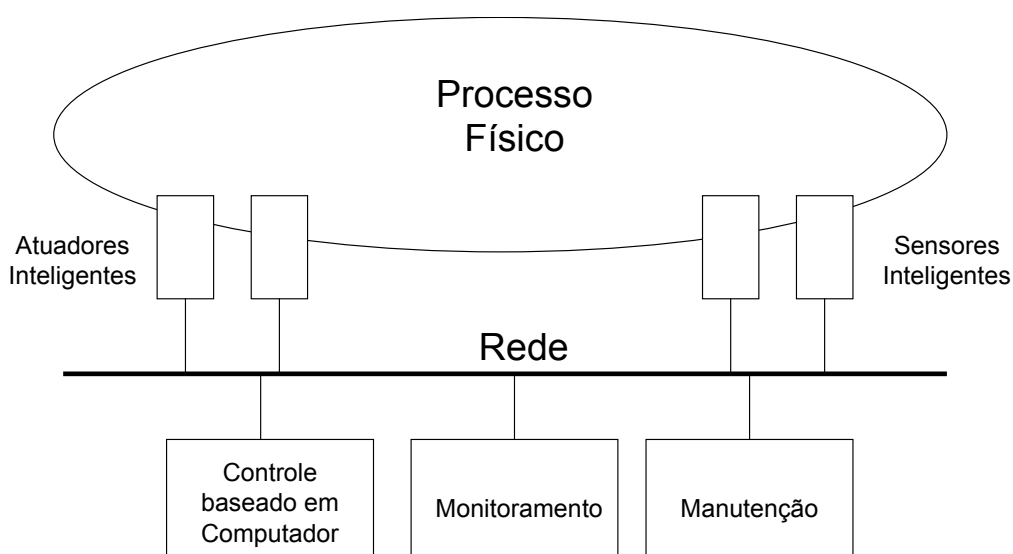


Figura 2.11 : Arquitetura de controle distribuído [53]

A integração entre os níveis operacionais e de decisão é um dos principais fatores que influenciam na implementação da arquitetura ilustrada na Figura 2.11. A proposta de descentralização do controle envolve a habilitação de sensores e atuadores inteligentes para a tomada de decisões. Isso é possível com a implementação de um nível intermediário de tomada de decisão. Este nível é chamado de nível adicional de decisão e está ilustrado na Figura 2.12 [37].

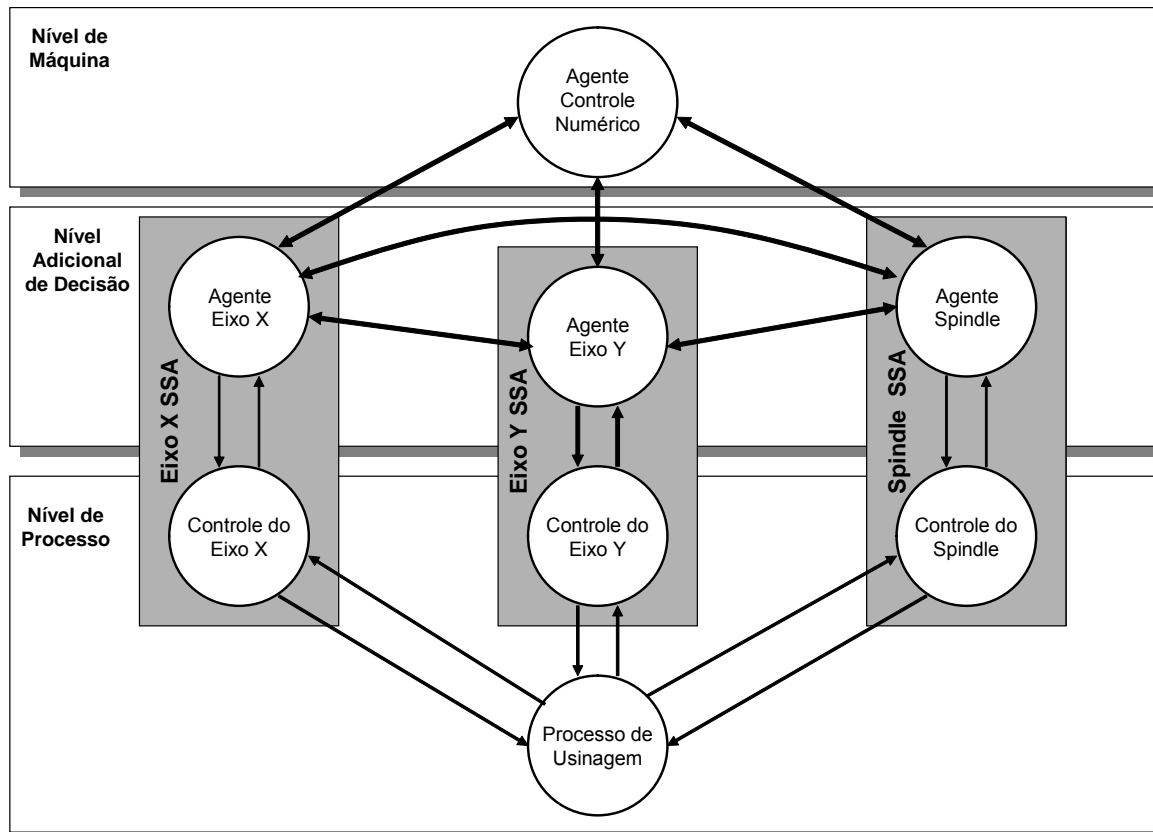


Figura 2.12 : Controle distribuído de um centro de torneamento [37]

Como pode ser visto na Figura 2.12, a tomada de decisões é distribuída entre os agentes e esses agentes trocam informações e cooperam para manter o controle do processo. O nível adicional de decisão faz a ligação entre os agentes dos três níveis e proporciona uma maior autonomia aos sensores e atuadores inteligentes, pois a distribuição do controle “aproxima” o nível de tomada de decisão para o nível de processo. A integração entre os níveis de controle e de processo é executada pelos sensores e atuadores inteligentes (*Smart Sensors and Actuators* – SSA) que possuem a autonomia para a tomada de decisões [37].

O sistema de manufatura distribuído desenvolvido por Cheng [54] possibilitou a flexibilização e meios de colaboração de uma máquina-ferramenta com os demais componentes do sistema (projeto, planejamento do processo, sistema de gerenciamento da produção). No seu trabalho, o agente máquina-ferramenta

executa funções de cooperação com outros agentes, comunicação com a máquina e monitoramento do estado atual de operação. A otimização do processo de usinagem foi obtida com o desenvolvimento de um módulo de controle inteligente implementado através de uma placa microcontrolada e estratégia de controle *fuzzy*.

## 2.2 CNC Comando Numérico Computadorizado

A função básica do CNC é comandar os movimentos dos eixos nas condições de posicionamento e velocidade determinados no programa NC. O CNC é constituído de um computador que converte as instruções do programa em sinais analógicos, e de sensores de posição e velocidade que executam a retroalimentação para o controle do processo. As instruções numéricas são convertidas em sinais analógicos e amplificadas para a tensão de funcionamento dos servo-motores. Os sinais dos sensores são utilizados pelos algoritmos de controle do CNC para manter o processo nas condições programadas [55].

A Figura 2.13 mostra o diagrama de blocos do sistema de controle de movimentos. O programa NC descreve os movimentos que a máquina-ferramenta deve executar e o controle de movimentos faz a tradução da interpolação da trajetória da ferramenta para a movimentação individual dos eixos com as velocidades programadas. A distância total a ser percorrida pelos eixos é quebrada em pequenos segmentos numa taxa repetitiva e de elevada exatidão. Esta taxa de atualização é tipicamente de 1 a 5 milissegundos. O controle aplica os movimentos interpolados na seção do *loop* de controle de posição, onde a posição determinada pelo dispositivo de retroalimentação é comparada com a posição de *set point* e determinada a diferença. Esta diferença é aplicada como comando de velocidade no servo e o valor é comparado com a velocidade atual, através dessa comparação é gerado o erro de velocidade que é utilizado pelo controle para gerar o comando de torque. Nos motores utilizados pelos acionamentos tradicionais dos eixos, o torque é proporcional a corrente e os semicondutores de potência são comutados para aplicar a corrente desejada nos motores [56].



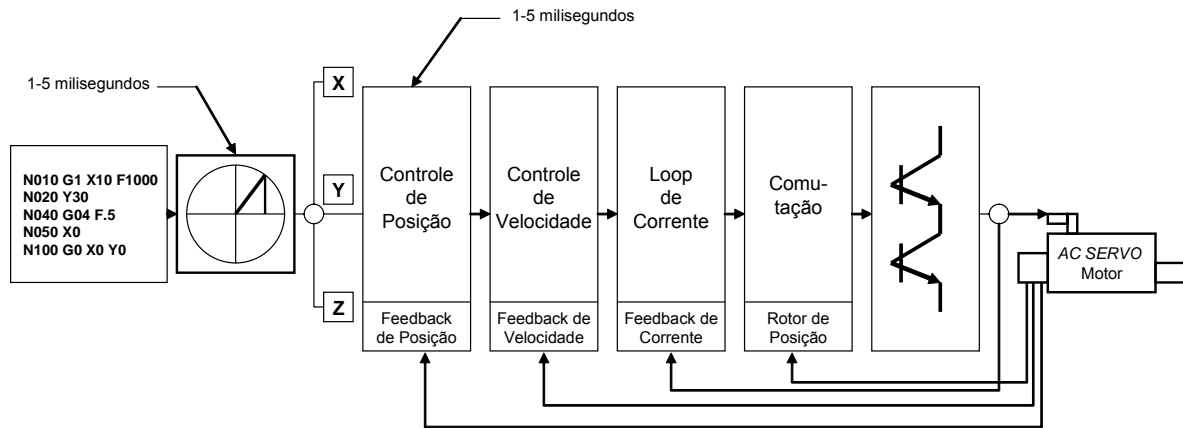


Figura 2.13: Diagrama de blocos do controle de movimentos [56]

Segundo Altintas [55], para percorrer a distância de 10 milímetros com uma velocidade de avanço de 200 milímetros por segundo, o *clock* de tempo real é setado para gerar 10.000 pulsos numa taxa de 200.000 pulsos por segundo (1 pulso = 0.001 milímetro de posicionamento). Os pulsos de posição (comandos de velocidade discretos) são convertidos em sinais analógicos equivalentes e amplificados para o acionamento dos motores e execução do movimento.

### 2.2.1 Características para aplicação em HSM

Os CNCs utilizados no processo HSM possuem funções especiais que os diferenciam dos comandos numéricos tradicionais. Devido à alta velocidade atingida pelas máquinas de alta performance são necessárias funções de auxílio ao processamento das informações e ao controle da estabilidade dinâmica do processo, dentre as quais destacam-se:

- Modificação da velocidade de avanço: essa função tem o objetivo de minimizar os efeitos de erro de contorno através da diminuição da velocidade de avanço em partes da peça onde há a mudança brusca no sentido da trajetória (exemplo: ângulos retos). Este conceito é aplicado com as funções *look ahead* e *contour optimization*. A função *look ahead* pré-processa uma nova trajetória da ferramenta com as velocidades de avanço modificadas e as armazena em um *buffer*, essa função não permite a especificação da tolerância de contorno e como ela não é executada em tempo real, não são possíveis alterações *online* na velocidade. A função *contour optimization* é executada em tempo real, permitindo ajustes *online* da velocidade e especificação de uma tolerância de contorno [33];
- Alteração do ganho: essa funcionalidade visa a obtenção de acelerações e velocidades ótimas da máquina de alta performance de acordo com limites de desvio da trajetória especificados para determinado tipo de

operação, como em operações de desbaste, onde a taxa de remoção de material é mais importante do que a precisão do contorno final. O CNC permite a seleção do valor do ganho que explore essa condição e resulte em menores tempos de usinagem. A mesma situação ocorre na operação de acabamento, porém, neste caso, a importância da qualidade dimensional é maior e o valor do ganho deve corresponder a esta condição [33];

- **NURBS (Non-Uniform Rational B-Spline):** a interpolação NURBS descreve a trajetória de superfícies complexas com menos instruções em comparação a interpolação linear. Dessa forma o processamento do programa no CNC é otimizado, eliminando alguns fatores que são fonte limitadoras da velocidade no processo de usinagem de alta performance. A Tecnologia NURBS foi criada principalmente para suprir as seguintes necessidades dos CNCs antigos: falta de memória para armazenamento de programas, baixa velocidade na transmissão de dados e baixa capacidade de processamento para conversão de pequenos segmentos de reta para o movimento dos motores [33]. A tecnologia NURBS e a interpolação *spline* são atualmente tema de diversos trabalhos que visam o desenvolvimento de um interpolador NURBS em tempo real e da otimização do processo de usinagem HSM [1, 17, 20, 21, 22, 47].

Dentre as características citadas do CNC para aplicação no processo de usinagem HSM, pode-se observar o quanto crítico é o tempo de processamento do CNC. A otimização dos ganhos do controlador é outra funcionalidade que resulta em ganhos de produtividade, porém pouco explorada no chão de fábrica pois grande parte dos CNCs não possuem uma interface amigável com facilidade de entendimento para o usuário final utilizar este recurso.

### **2.2.2 Controlador de arquitetura aberta**

O conceito do controlador de arquitetura aberta recebeu um maior destaque na década de 1990, pois a flexibilidade demandada pelos sistemas de manufatura e a necessidade de implementação de funções customizadas pelos usuários requeriam uma interface neutra de acesso aos dados.

Os requisitos de níveis de acessibilidade e customização de funções do CNC de arquitetura aberta são diferentes na opinião do usuário final em relação à opinião dos fabricantes dos comandos numéricos. Para o usuário final, os requisitos são: acesso e customização em todos os níveis de comunicação de uma forma padronizada e sem a necessidade de compra de módulos adicionais para tal finalidade. Para os fabricantes do CNC, o nível de acessibilidade é em grande parte limitado a customização da interface homem-máquina. Caso haja a

necessidade de implementação de funções em níveis de controle do Interpolador e do servo, o usuário depende de informações/módulos de *software* do fabricante do comando numérico para tal finalidade [13].

A definição do instituto IEEE para o sistema de arquitetura aberta é a seguinte: “Um sistema aberto possui características que habilitam as aplicações implementadas a serem executadas em várias plataformas de diferentes fabricantes, interoperáveis com outros sistemas e com uma interação consistente com o usuário” [57].

Conforme a definição acima, um conjunto de especificações é necessário para que os fabricantes dos controladores utilizem-no. Porém, muitas iniciativas já surgiram nesse sentido e constata-se que um consenso entre os fabricantes para a adoção de determinado padrão não foi atingido. Entre as iniciativas para tal padronização, destacam-se:

- OSEC (*Open System Environment for Manufacturing*): este consórcio foi estabelecido no Japão em 1994 com a proposta de definir uma plataforma padrão para os controladores industriais. Essa plataforma utiliza-se de APIs que encapsulam os dados e os dividem em blocos funcionais representados por objetos, essa representação elimina a necessidade de conhecimento detalhado sobre como o objeto executa determinada função [14];
- JOP (*Japanese Open Promotion Group*): paralelamente as iniciativas do OSEC, surgiu também no Japão dentro do JOP o comitê técnico de controlador aberto OC-TC (*Open-Controller Technical Committee*) com o objetivo de facilitar a discussão e o trabalho em conjunto dos fabricantes de controladores para a padronização da tecnologia do controlador aberto [14];
- OMAC (*Open Modular Architecture Controllers*): as indústrias norte americanas criaram este fórum com o propósito de definição de especificações de API através de classes de interface para a implementação nos controladores. Esta abordagem está baseada em componentes e visa a facilidade de modularização [14];
- OSACA (*Open System Architecture for Controls within Automation Systems*): foi criado em 1992 pelos principais institutos de pesquisa e fabricantes de controladores e máquinas ferramentas da Europa, com o objetivo de constituir-se em uma plataforma padronizada e neutra para os sistemas de controle de arquitetura aberta. A principal característica técnica do OSACA é a decomposição hierárquica das funções de controle em unidades funcionais (exemplos: controle de movimentos,

controle dos eixos, etc.), e a especificação das interfaces é feita através de modelos de informação baseados em orientação a objetos [14].

Conforme mostram as iniciativas acima, vê-se que todas têm como objetivo comum a definição de um padrão para os controladores de arquitetura aberta e esse padrão tem como características principais a neutralidade em relação aos fabricantes e a customização de funções através da modularização e da utilização de API. No entanto, não existe uma compatibilidade entre os padrões propostos, ou seja, mesmo com um objetivo em comum, esses consórcios concorreram entre si e o resultado dessa falta de integração é a não utilização em larga escala no ambiente industrial.

### 2.2.3 Requisitos para o sistema de controle de arquitetura aberta

Um dos requisitos mais importantes de um sistema de controle de arquitetura aberta é a modularização. A utilização de módulos permite a divisão do CNC em unidades funcionais e a integração desses módulos exige uma interface padronizada. A capacidade de tornar a arquitetura mais aberta está relacionada com a complexidade dos módulos, módulos menores permitem um maior detalhamento de funções, porém exigem maior custo de integração [14].

Devido à complexidade na utilização de módulos, é recomendada a definição de uma estrutura para o sistema. Essa estruturação consiste em dividir o sistema em plataformas que encapsulam os dados do sistema computacional como as características de hardware, sistema operacional e comunicação, e facilita a interoperabilidade entre os módulos. A Figura 2.14 ilustra a estrutura do sistema de controle modular e aberto.

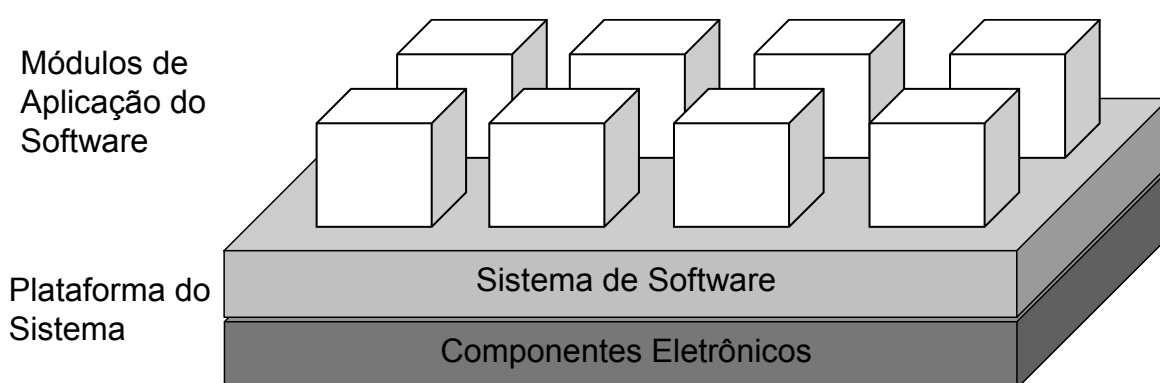


Figura 2.14 : Estrutura do sistema de controle modular e aberto [14]

A estrutura do sistema de controle com arquitetura aberta é geralmente constituída de três níveis, sendo: nível de dispositivos, de controle e de sistema. Para a comunicação entre os níveis, é requerida uma interface aberta e padronizada, ver Figura 2.15 [15].

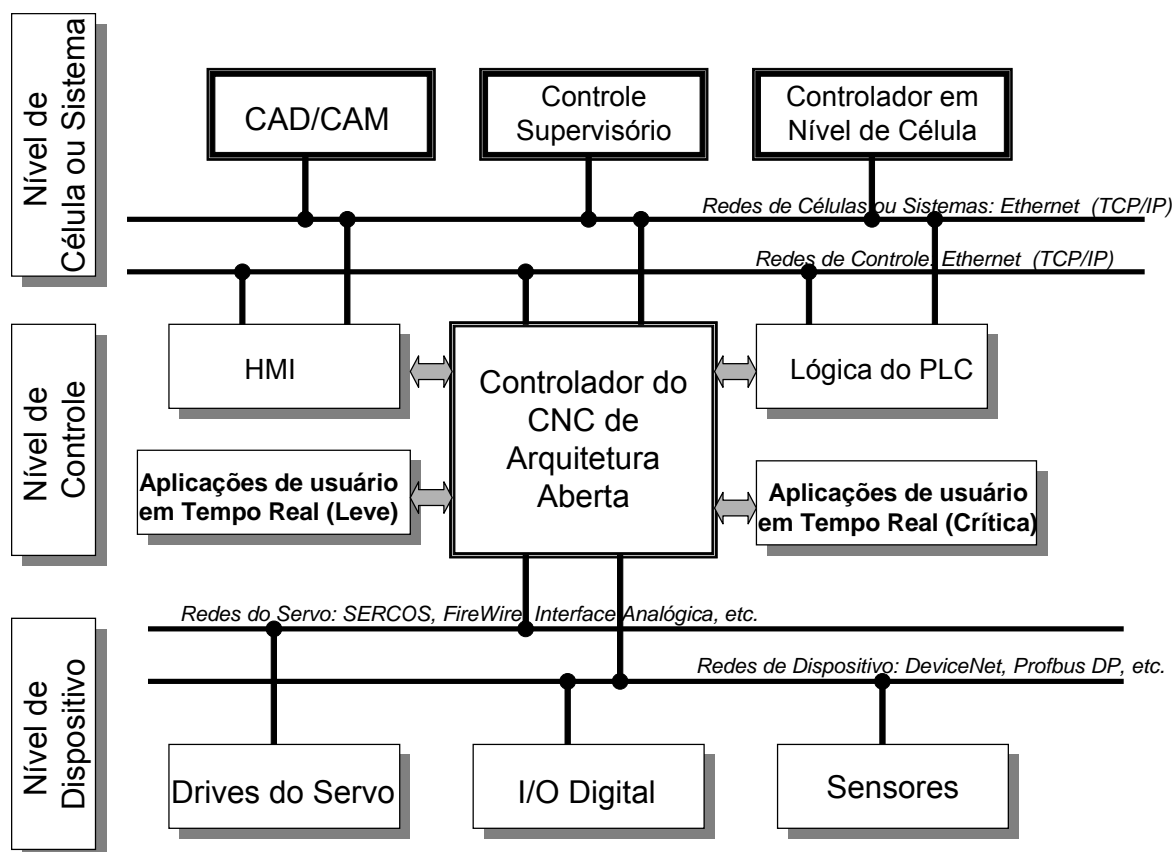


Figura 2.15 : Estrutura típica de níveis do CNC de arquitetura aberta [15]

Na estrutura da Figura 2.15 pode-se observar que as redes de comunicação são as interfaces padronizadas para a troca de dados entre os níveis especificados. No entanto, um sistema com uma interface de rede de comunicação não é sinônimo de um sistema totalmente aberto. A definição das plataformas do sistema, o encapsulamento dos dados e a modularização são os principais requisitos para a classificação de um sistema de arquitetura aberta.

A interface com dispositivos externos, a capacidade de customização de funções pelo usuário e a integração com outros softwares são os requisitos propostos por Yun [15] para um controlador de arquitetura aberta, e podem ser vistos na Tabela 2.1.

Item	Descrição
Interface com dispositivos externos	Interface do servo Interface de sinais I/O PLC
Capacidade de customização pelo usuário	Nível de HMI Nível de <i>software</i> em tempo real Nível de <i>hardware</i> em tempo real
Integração com outros <i>softwares</i>	Compatibilidade com outros <i>softwares</i> Conectividade com sistemas de gerenciamento

Tabela 2.1: Requisitos do controlador de arquitetura aberta [15]

Dentre os requisitos apresentados na Tabela 2.1, a capacidade de customização de funções pelo usuário é o que se relaciona diretamente com a otimização do processo de usinagem, pois através do acesso aos níveis de *hardware* e *software* em tempo real, pode-se implementar algoritmos de análise e tomada de decisão para otimização do processo.

A implementação de funções no CNC de arquitetura aberta está inserida no contexto de um sistema baseado em computador, que será apresentado no próximo sub-tópico.

### 2.3 Sistemas baseados em computador

Com base na definição do dicionário *Webster's*, Pressman [58], define um sistema baseado em computador como “um conjunto ou disposição de elementos que é organizado para executar certo método, procedimento ou controle ao processar informações”

Um sistema está inserido num contexto que envolve interações de *software* e *hardware* com os usuários e o ambiente. Nesse contexto, deve-se levar em consideração os serviços que o sistema oferecerá, as limitações de construção e funcionamento e o relacionamento entre o sistema e seus respectivos usuários e ambiente de operação [60].

O conjunto, ou elementos que constituem um sistema baseado em computador são apresentados na Figura 2.16.

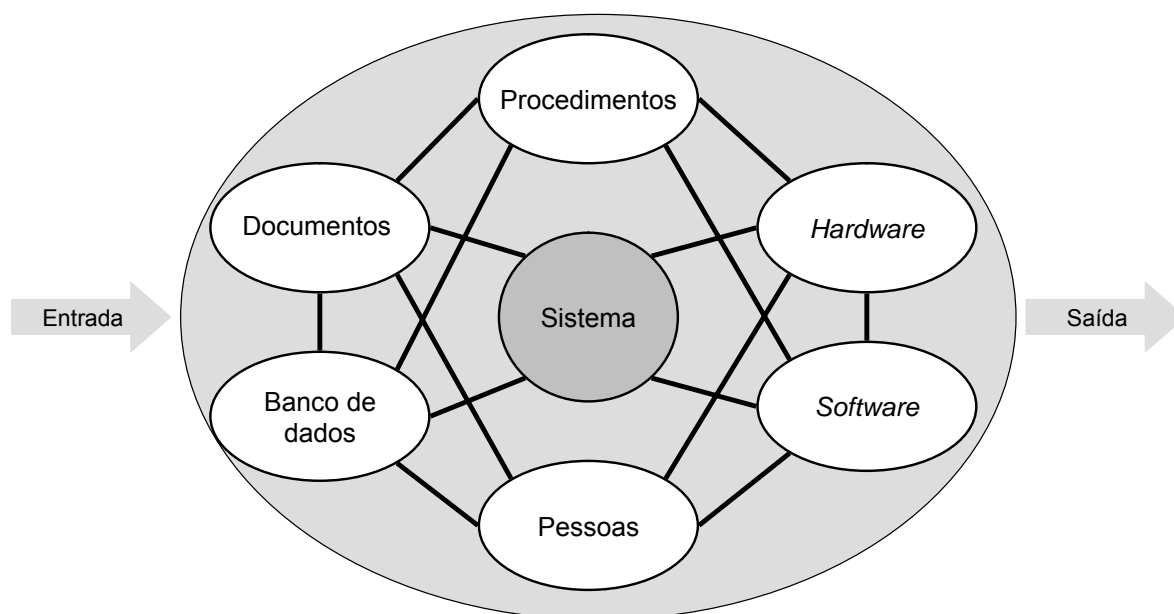


Figura 2.16: Elementos de um sistema baseado em computador [59]

Para a definição do conjunto de elementos ilustrados na Figura 2.16 deve-se determinar quais serão as entradas, as ações e as saídas produzidas pelos elementos do sistema, tanto na forma individual quanto na forma de relacionamento entre elementos. Essa análise de relacionamentos de elementos deve ser acompanhada da determinação das fronteiras do sistema, pois dessa forma pode-se estabelecer os limites de atuação do sistema e determinar o estado das entradas e saídas no contexto de aplicação ou funcionamento do sistema [58].

Os caminhos e os modos para o desenvolvimento do *software* são descritos pelos modelos de processo. O processo de desenvolvimento de *software* caracteriza-se pelos requisitos, que são as entradas do sistema e pelo produto final, que é a saída [58]. Existem vários modelos de processo para o desenvolvimento de *software* e um dos primeiros modelos foi o modelo cascata, apresentado na Figura 2.17.

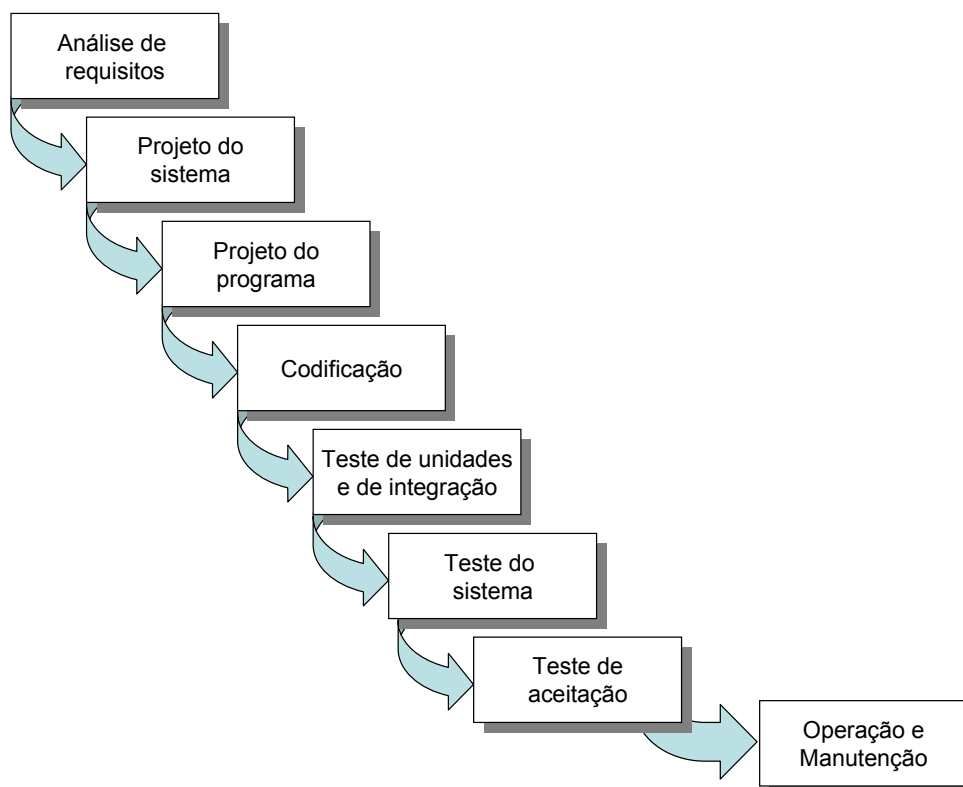


Figura 2.17: O modelo em cascata [58]

Na Figura 2.17 observa-se que no modelo em cascata as etapas do desenvolvimento do *software* são seqüenciais e o início da próxima etapa é somente feito com a conclusão da etapa anterior. Esse modelo proporciona uma visão geral do que acontece no desenvolvimento do *software* e auxilia na determinação do que é necessário estar pronto para o início do estágio posterior [58].

Nos subitens seguintes serão apresentadas as seguintes fases do desenvolvimento de *software*: análise e definição dos requisitos, modelando a arquitetura do sistema, projeto e teste de *software*.

### 2.3.1 Requisitos de *software*

A análise de requisitos determina a ligação entre as fases de definição do sistema e projeto de *software*. No processo de definição e análise dos requisitos, o desenvolvedor especifica a função e o desempenho do *software*, as interfaces entre os elementos do sistema e as limitações de projeto [59].

Segue abaixo a classificação dos requisitos de *software*:

- Requisitos funcionais: os requisitos funcionais declaram como o sistema deve responder as entradas e como ele deve se comportar, ou mesmo o que o sistema não deva executar [60];
- Requisitos não funcionais: os requisitos não funcionais são as limitações de serviços ou de funções desempenhadas pelo sistema, como limitações de tempo de execução, do processo de desenvolvimento, de padrões, etc [60];
- Requisitos do usuário: especificam de forma superficial os requisitos funcionais e não funcionais com uma linguagem compreensível para o usuário final. Os detalhes de projeto do sistema não devem ser abordados nestes requisitos [60];
- Requisitos de sistema: são o detalhamento das descrições do usuário, contendo uma especificação completa do sistema. A partir desses requisitos, é iniciado o projeto do sistema [60].

Na classificação dos requisitos do *software* pode-se constatar que ocorre um direcionamento do nível geral para o nível de detalhamento dos requisitos, até a obtenção dos requisitos de sistema. Isso não significa que os requisitos funcionais sejam menos importantes que os requisitos de sistema, porém os requisitos de sistema só serão mais precisos se os requisitos funcionais, não funcionais e de usuário foram bem elaborados.

A partir da definição e análise dos requisitos inicia-se a etapa do projeto do sistema, que é apresentada no tópico seguinte.



### 2.3.2 Projeto de arquitetura do sistema

A criação de um modelo da arquitetura do sistema visa demonstrar o inter-relacionamento dos componentes e servir de referência para as etapas seguintes do processo de desenvolvimento do sistema [59].

A arquitetura define o relacionamento entre os componentes do sistema associando o que foi identificado na especificação dos requisitos com o componente responsável pela execução de tal especificação. Os componentes são geralmente descritos por módulos e a ligação entre os módulos é representada pela arquitetura [58].

No projeto da arquitetura ocorre a alocação dos requisitos do sistema aos componentes, ou seja, para que seja assegurado determinado requisito do sistema, os componentes devem possuir requisitos correspondentes. Nesta atividade evidencia-se a importância que cada componente (ex: *hardware*, *software*, pessoas, etc.) possui, assim como o relacionamento entre componentes, para que sejam obtidos os requisitos do “conjunto” que constitui o sistema [61].

Os diagramas de arquitetura são ferramentas de auxílio para a criação do modelo da arquitetura do sistema. Os elementos do sistema são alocados no modelo padrão de arquitetura. Este padrão é constituído de cinco regiões, são elas: interface com o usuário, entrada, função de processo e controle do sistema, saída e manutenção e autoteste. O modelo padrão de arquitetura é ilustrado na Figura 2.18 [59].

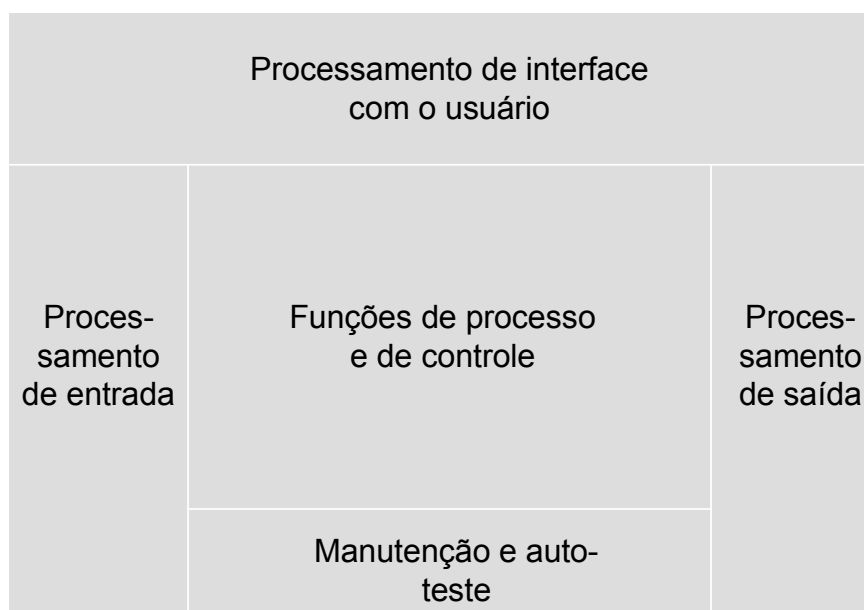


Figura 2.18: Modelo de arquitetura [59].

O diagrama de contexto da arquitetura é obtido com a utilização do modelo apresentado na Figura 2.18. Esse diagrama apresenta uma visão geral de como ocorre o relacionamento e fluxo de informação entre as interfaces de processamento, controle e diagnóstico do sistema. Para a definição dos subsistemas, utiliza-se o diagrama de fluxo de arquitetura, que é derivado do diagrama de contexto. O diagrama de fluxo de arquitetura é mais detalhado e evidencia os subsistemas e seus fluxos de informações [59].

### 2.3.3 Projeto de *software* de tempo real

Segundo Sommerville [60], “um sistema de tempo real é um sistema de *software* cujo funcionamento correto depende dos resultados produzidos por ele e do tempo em que esses resultados são produzidos”. Se o resultado da operação não for produzido conforme os requisitos de tempo, a operação pode ser degradada caracterizando um sistema de tempo real leve ou incorreta caracterizando um sistema de tempo real rígido [60].

Como visto no tópico 2.3, o sistema constitui-se de um conjunto de elementos e no caso de sistemas de tempo real, deve-se verificar logo no início as restrições de tempos e determinar o *hardware* adequado para atender a estas especificações. Um *hardware* de maior desempenho pode ser adotado para eliminação de gargalos de processamento e assim evitar-se o gasto de uma quantidade elevada de tempo na otimização do *software* [60].

Nos sistemas de tempo real a linguagem de programação influencia no projeto. No caso de sistemas de tempo real rígidos geralmente é utilizada a linguagem *assembler*. As linguagens de alto nível que possibilitam gerar códigos eficazes também podem ser utilizadas (ex: linguagem C) [60].

Como exemplo de sistemas de tempo real tem-se os sistemas de monitoramento e controle e sistemas de aquisição de dados. Nos sistemas de monitoramento a ação é tomada quando o valor da variável monitorada excede limites pré-estabelecidos, e nos sistemas de controle os atuadores são continuamente controlados para manter o controle das variáveis envolvidas no processo [60].

Os sistemas de aquisição de dados executam a coleta de dados de sensores para posterior processamento e análise. Na Figura 2.19 é ilustrado esse conceito com a representação de um sistema que faz a coleta de dados para o monitoramento de fluxo de nêutrons em um reator nuclear [60].

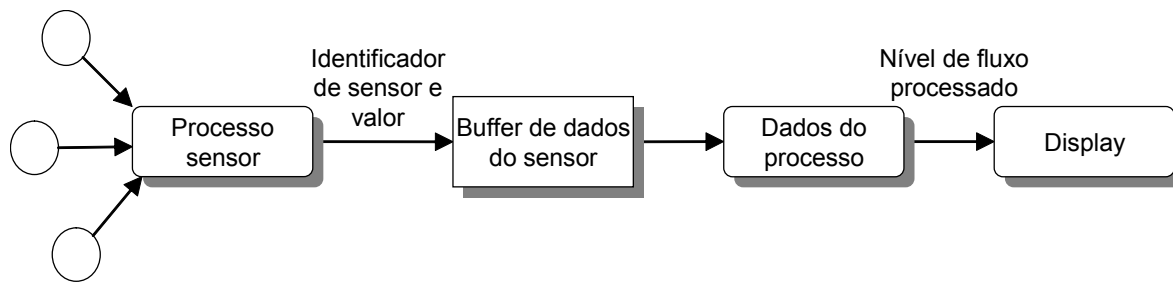


Figura 2.19: Arquitetura de um sistema para aquisição e monitoramento de fluxo [60]

Na Figura 2.19 os sensores são ilustrados pelos círculos e seus respectivos sinais são tratados pelo processo sensor e posteriormente armazenados no *buffer* de dados para processamento e análise.

Nos sistemas de tempo real, geralmente é utilizado um *buffer* circular ou em anel para minimizar diferenças na velocidade de aquisição e processamento de dados caracterizando a configuração produtor/consumidor. A Figura 2.20 ilustra um *buffer* em anel para aquisição de dados [60].

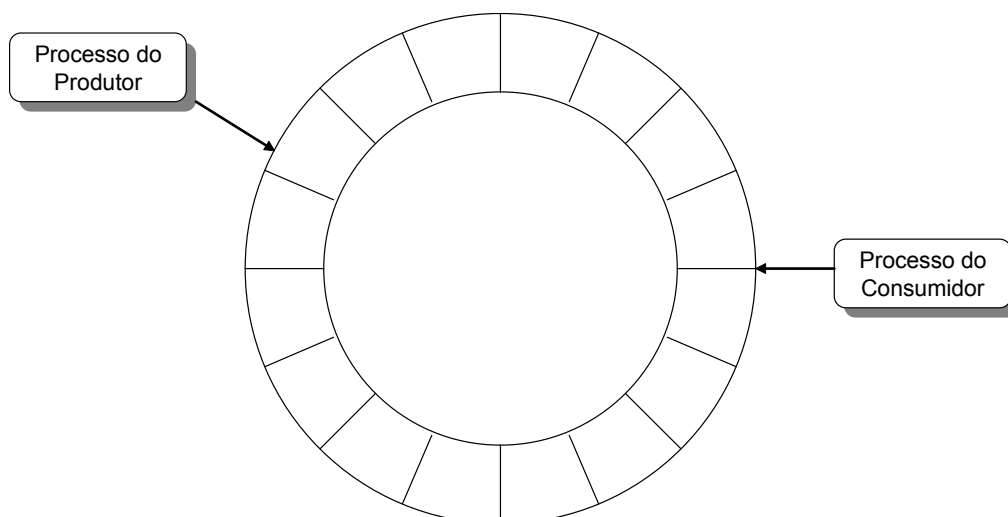


Figura 2.20: Buffer em anel para aquisição de dados [60]

Conforme visto na Figura 2.20, o processo produtor insere os dados coletados pelo sistema de aquisição no *buffer* e o processo consumidor retira esses dados para processamento e análise. Nesse modelo deve-se implementar condições que evitem o armazenamento de dados quando o *buffer* está cheio e evite a retirada de dados de um *buffer* vazio, além de restrições que garantam que o mesmo elemento não seja acessado de forma simultânea por ambos processos [60].

### 2.3.4 Projeto estruturado

Na análise estruturada são construídos modelos para representar o conteúdo e as transformações que acontecem com o fluxo de informação. No sistema baseado

em computador a informação é processada e modificada durante o seu fluxo, como exemplo tem-se o controle de temperatura do ambiente com ar condicionado, onde o valor de temperatura ambiente proveniente do sensor é uma informação que se modifica e é modificada durante o seu fluxo pelo sistema [59].

O diagrama de fluxo de dados (DFD) é uma ferramenta de auxílio para a análise do fluxo de informação e suas transformações, ele pode representar um sistema ou *software* em qualquer nível de abstração. A abstração é utilizada para a solução modular de um problema, geralmente parte-se de um nível generalizado para um nível aprimorado da abstração da solução, ou seja, a solução é proposta de forma genérica num primeiro nível e torna-se específica nos níveis seguintes até que seja obtida a representação com os procedimentos para implementação da programação [59].

O projeto estruturado é um dos métodos clássicos de projeto de *software* onde busca-se identificar as funções principais do sistema e em seguida refiná-las. É aplicado posteriormente a análise estruturada, através do detalhamento do DFD. O DFD tem a vantagem de ser intuitivo e possui uma representação de fácil entendimento [59, 60, 61].

A representação de descrições genéricas do sistema e seu respectivo fluxo de informações são obtidas pelo DFD, que é ilustrado na Figura 2.21.

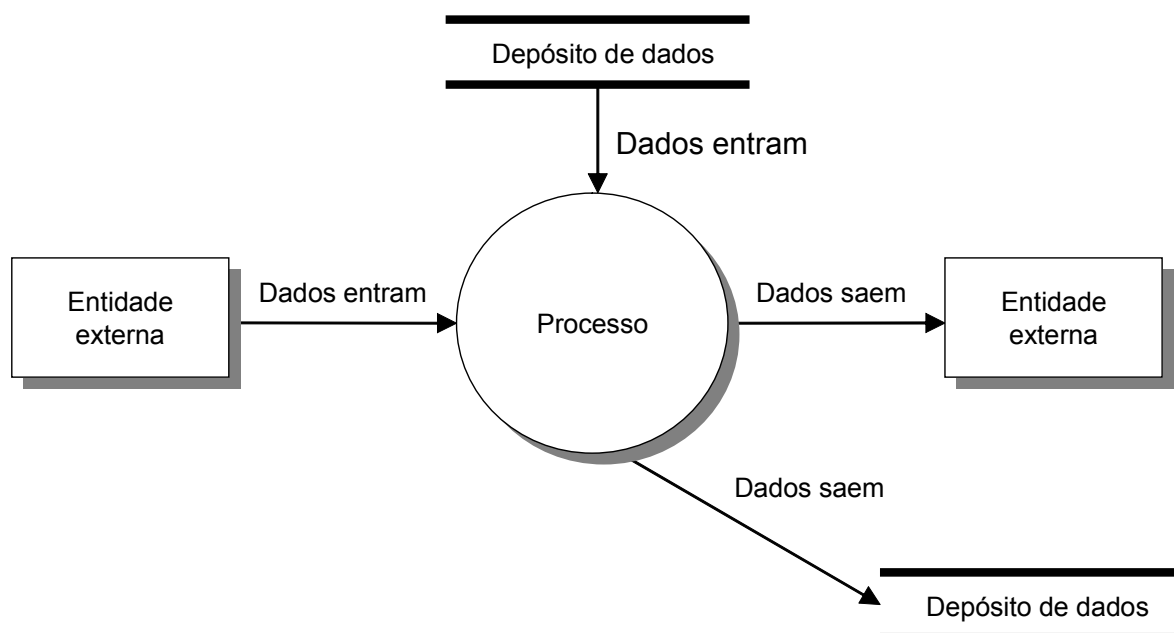


Figura 2.21: Diagrama de fluxo de dados (adaptada de Pfleeger [58]).

Conforme visto na Figura 2.21, o DFD é constituído dos seguintes elementos:

- Entidade externa: a entidade externa indica uma origem ou destino dos dados [62];
- Processo: indica a transformação dos dados de entrada para os dados de saída [62];
- Fluxo de dados: define a direção do fluxo de dados [62];
- Depósito de dados: indica o local de armazenamento dos dados [62].

As características do fluxo de informações determinados no DFD são utilizadas para o projeto do *software* através do método de orientação ao fluxo de dados. O projeto orientado ao fluxo de dados é indicado para as aplicações que utilizam o processamento seqüencial das informações, como exemplo tem-se as aplicações de controle em microprocessadores, análise numérica complexa, controle de processos e em grande parte das aplicações na área científica e de engenharia [59].

A orientação a objetos é uma abordagem que vem sendo muito utilizada no desenvolvimento de *software*. Nessa abordagem os problemas e as soluções são organizados em conjuntos de objetos que representam a estrutura e o comportamento dos dados. As principais características dessa representação são: identidade, abstração, classificação, encapsulamento, herança, polimorfismo e persistência [58].

Neste trabalho foi utilizada a abordagem de projeto orientada ao fluxo de dados, por ser uma aplicação com características de processamento seqüencial nos módulos de aquisição e de transmissão e análise dos dados.

### **2.3.5 Teste de *software***

Os testes de *software* têm o objetivo de identificar os erros gerados durante o ciclo de desenvolvimento do *software*. Os erros podem ser gerados na fase de definição de requisitos, no projeto do sistema, na documentação, na codificação e mesmo na manutenção, ao consultar uma documentação com erros. Na especificação pode haver erros na declaração das necessidades do cliente ou mesmo estar faltando declarações de requisitos [58].

Um único erro de *software* pode gerar um ou mais defeitos. Como exemplo, tem-se a interpretação incorreta dos requisitos pelo projetista do *software*. Esse erro pode gerar o defeito de que o *software* não executa o que o analista espera ou mesmo gerar outros defeitos como codificação incorreta e manual de usuário com descrições erradas. A existência de defeitos nem sempre significa a certeza da ocorrência de falhas, pois se o código defeituoso nunca for executado a falha não será perceptível pelo usuário. A falha é vista a partir de uma referência externa ao

sistema (ex: pelos usuários) e os defeitos a partir de uma referência interna (ex: pelos desenvolvedores) [58].

A classificação e análise dos defeitos auxiliam na elaboração de testes específicos das falhas de maior frequência de incidência e também auxiliam na avaliação do processo de desenvolvimento do *software*, apontando necessidades de reavaliação do projeto ou mesmo dos requisitos. A Tabela 2.2 mostra a classificação ortogonal de defeitos da IBM, onde os tipos de defeitos são agrupados em categorias [58].

<b>Tipo de defeito</b>	<b>Significado</b>
Função	Defeito que afeta a capacidade, as interfaces com o usuário final, as interfaces do produto, a interface com a arquitetura de <i>hardware</i> ou as estruturas globais de dados
Interface	Defeito na interação com outros componentes ou <i>drivers</i> , por meio de chamadas, macros, blocos de controle ou lista de parâmetros
Verificação	Defeito na lógica do programa, que falha ao validar dados e valores apropriadamente, antes de utilizá-los
Atribuição	Defeito na estrutura de dados ou na inicialização do bloco de código
Sincronia/sequência	Defeito que envolve a sincronia de recursos compartilhados e em tempo real
Versão/empacotamento	Defeito que ocorre devido a problemas nos repositórios, de mudanças de gerenciamento ou no controle de versões
Documentação	Defeito que afeta as informações nas publicações ou sobre a manutenção
Algoritmo	Defeito que envolve a eficiência ou a correção do algoritmo ou da estrutura de dados, mas não do projeto

Tabela 2.2: Classificação ortogonal de defeitos [58]

Como já foi visto no item 2.4, o *software* é um dos elementos do sistema baseado em computador e o teste do sistema tem o objetivo de verificar se os elementos que o constituem foram integrados corretamente e executam aquilo que o cliente espera [58, 59].

### 2.3.6 Teste de sistema

O teste de sistema visa a identificação de erros gerados pela integração do conjunto de elementos do sistema. Entre os testes existentes para o sistema estão os testes funcionais, de recuperação, de segurança, de estresse e de desempenho. Esses tipos de testes são abordados abaixo:

- Teste funcional: o teste funcional é baseado nos requisitos funcionais do sistema. Nesse tipo de teste, a estrutura interna do sistema é tratada como uma caixa-preta, pois o que interessa é testar se o sistema é capaz de fazer o que foi descrito nos requisitos funcionais [58];

- Teste de recuperação: o teste de recuperação avalia se o sistema é capaz de recuperar-se adequadamente após diversas falhas serem aplicadas no sistema. Essa recuperação deve ocorrer dentro de um tempo previamente especificado [59];
- Teste de segurança: o teste de segurança visa garantir a proteção do sistema no que se refere ao acesso e modificação de dados. Um possível indicador para o nível de segurança de um sistema é quando o custo do acesso aos dados do sistema passa a ser mais alto do que a informação a ser obtida [59];
- Teste de estresse: no teste de estresse o sistema é colocado em condições anormais, exigindo uma quantidade muito maior de recursos em frequência e volume anormais. Nessa situação busca-se verificar qual será o comportamento do sistema e as respectivas limitações [59];
- Teste de desempenho: o teste de desempenho avalia o *software* no contexto do sistema em execução. O sistema deve ser capaz de executar as funções oferecidas conforme os requisitos de desempenho especificados [59].

Os possíveis erros nos quais os testes têm a função de identificar são gerados ao longo do desenvolvimento do *software* e do sistema. Assim, uma maneira de prevenir que falhas ocorram é através de utilização de técnicas de desenvolvimento que resultem em diminuição do número de falhas [63].

## 2.4 Protocolos de comunicação industrial

O protocolo é o principal elemento de uma rede de comunicação pois ele define as regras operacionais da rede, e devido esta importância, a denominação da rede passa a ser feita pelo protocolo que esta utiliza. Na Figura 2.22, são identificados na pirâmide de automação os cinco níveis de hierarquia de comunicação e os protocolos associados aos níveis em que são mais utilizados [64].

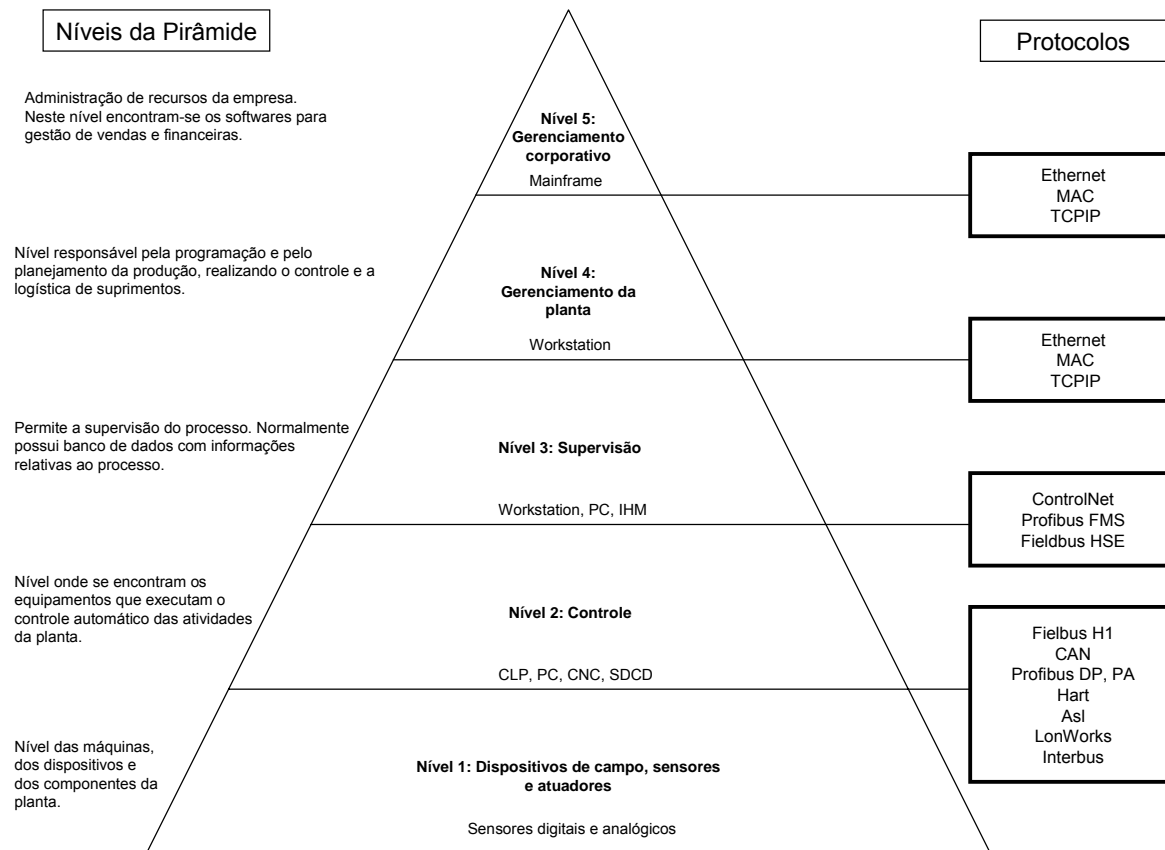


Figura 2.22: Níveis e protocolos de comunicação na pirâmide de automação [64]

Os níveis um e dois apresentados na Figura 2.22 são caracterizados da seguinte forma:

- Nível 1: protocolos para a manipulação de dados de natureza analógica, projetados para comunicação de instrumentos de campo [65];
- Nível 2: protocolos para a troca de informações através de pequenos pacotes de informações de natureza digital, projetados para comunicação de dispositivos [65].

Os protocolos dos níveis citados acima constituem a tecnologia conhecida como *fieldbus* ou redes de campo, que surgiu para substituir o padrão de instrumentação centralizada que utiliza o sinal analógico de quatro a vinte miliamperes por um padrão de comunicação digital global. Este padrão digital visa a redução de custos e a descentralização dos sistemas de controle [66].

#### 2.4.1 Fatores de escolha do protocolo de comunicação

A classificação dos protocolos de comunicação em níveis de aplicação pode resultar na escolha incorreta de determinada tecnologia de redes de campo, pois os protocolos podem possuir requisitos que os habilitem a serem aplicados em



---

mais de um nível da pirâmide de automação [66]. As questões seguintes podem servir de auxílio na escolha do protocolo de comunicação:

- A tecnologia oferecida pelo protocolo de comunicação atende as necessidades da aplicação? [66];
- Qual será a economia de dinheiro e tempo? [66];
- Possui conectividade e portabilidade de dados? [66];
- É oferecido suporte pós-venda? [66];
- Quais são os sistemas *fieldbus* disponíveis? [66];
- Qual melhor tecnologia se aplica nesta planta? [66];
- Esta tecnologia possui interoperabilidade com dispositivos de diversos fabricantes? [66];
- Qual foi o desempenho nos testes de interoperabilidade? [66];
- Qual deve ser a capacitação dos profissionais para a implementação da nova tecnologia? [66].

As questões apresentadas acima são somente recomendações para a escolha do protocolo de comunicação de campo a ser utilizado. A sociedade ISA elaborou um questionário para identificar as reais necessidades do usuário em relação aos requisitos funcionais de uma rede de campo e dentre os pontos identificados destacam-se:

- Facilidade para adição de novos componentes de campo [67];
- Possuir comunicação bidirecional [67];
- Permitir a adição de estratégias avançadas de controle [67];
- Tempo máximo de resposta de mensagem [67];
- Frequência de mensagens [67];
- Tamanho da rede em número de estações [67];
- Topologia da rede [67].

Dentre os fatores de escolha recomendados e os requisitos identificados pelo questionário da ISA, pode-se observar que a escolha do protocolo de comunicação envolve decisões estratégicas no que se refere a flexibilidade, interoperabilidade e custos do sistema de manufatura. No tópico seguinte são apresentados os principais protocolos utilizados em redes de campo.

## 2.4.2 Principais protocolos de redes de campo

Os protocolos abaixo são os padrões mais utilizados e encontrados nas redes de campo:

- Foundation Fieldbus: surgiu em 1994 com a união do WorldFIP e ISP (*Interoperable Systems Project*). É constituída de aproximadamente 140 membros que representam os principais fornecedores e usuários da área de automação e sistemas de controle de processo. Este protocolo é baseado no modelo de referência de sete camadas OSI (*Open Systems Interconnect*) e é projetado para suportar aplicações críticas, suas especificações são compatíveis com os padrões oficiais SP50 da ISA (*Instrumentation, Systems and Automation Society*) e do IEC (*International Electrotechnical Commission*) [66];
- Profibus: estabelecido em 1989 na Alemanha pela norma DIN 19245 e certificada pelo padrão europeu EN 50170 em 1996. Possui um centro de suporte internacional em Karlsruhe na Alemanha e conta com mais de 1000 membros pelo mundo com representação local em vinte e três países. O protocolo PROFIBUS suporta as tecnologias produtor/consumidor, *time stamping*, funções de segurança e sincronização de tempo, permitindo sua vasta aplicação no ambiente industrial [66];
- ControlNet: a ControlNet da Allen-Bradley é o estado da arte em redes de controle que atende a demanda de aplicações de tempo real com alta taxa de transmissão de dados. Todos os dispositivos da rede ControlNet podem ser configurados para acessar dados da rede Foundation Fieldbus através do dispositivo ControlNet-to-Foundation Fieldbus [66];
- WorldFIP: segue o padrão europeu EN50170 sendo aberto e estável, é o protocolo preferido nas aplicações no setor público e indústrias de utilidade da união europeia [66];
- Modbus: desenvolvido em 1979 pela Modicon é estruturado na comunicação mestre/escravo e cliente/servidor. É um protocolo de rede padronizado e aberto com grande utilização no ambiente industrial [66];
- Interbus: projetada para transmissão rápida de dados entre sensores e atuadores e devido ao seu procedimento de transmissão e topologia em anel, oferece recursos de transmissão de dados do processo de forma rápida, cíclica e com tempos equidistantes, diagnóstico, *plug and play*, assim como ótimos requisitos para uso de fibra ótica [66];

- CAN: o protocolo CAN (*Controller Area Network*) foi desenvolvido para aplicações automotivas e segue o padrão para comunicação serial ISO 11898. O barramento de comunicação CAN é do tipo *broadcast*, onde todos os nós da rede podem “ouvir” todas as transmissões e através de filtros de *hardware* executa-se a seleção do nó para responder somente as mensagens de interesse. A utilização do protocolo CAN não se limitou a indústria automotiva e também recebeu grande aceitação no setor de automação industrial [66];
- LonWorks: desenvolvido pela Echelon para prover soluções de controle para os setores industrial, de transporte, predial e residencial. Na rede LonWorks, cada nó possui inteligência embarcada e desempenham funções de controle do processo. Dispositivos como sensores de proximidade, chaves, detectores de movimento, instrumentos, entre outros podem servir como nós inteligentes na rede [66].

A aplicação da rede de comunicação ethernet em automação industrial tem se mostrado promissora. Com velocidades de 10 Gbps, estudos mostram que a ethernet industrial provavelmente unificará o padrão para redes de comunicação de campo [68].

## 3 Proposta do Trabalho

Neste capítulo são apresentados os objetivos e a metodologia desse trabalho.

### 3.1 Objetivos

Este trabalho tem como objetivo geral desenvolver um sistema para monitoramento de dados de um CNC de arquitetura aberta que seja aplicado no processo HSM. Os objetivos específicos são:

- Monitorar a velocidade de avanço e aceleração dos eixos;
- Identificar os parâmetros ótimos de programação dos métodos de interpolação da trajetória da ferramenta;
- Avaliar a interface de arquitetura aberta de um CNC comercial.

A interface de arquitetura aberta será avaliada pelo nível de acesso aos dados e pelo nível de customização de funções. Através dessa avaliação, pretende-se identificar as limitações e as possibilidades para a implementação de sistemas de monitoramento de dados no CNC aberto.

Não faz parte dos objetivos desse trabalho avaliar a exatidão dos dados fornecidos pelo CNC.

### 3.2 Metodologia

A pesquisa experimental é caracterizada neste trabalho, pois pretende-se selecionar as variáveis que influenciam no objeto de estudo e definir as formas de controle e análise dos efeitos produzidos pelas variáveis no objeto de estudo [69].

O desenvolvimento do experimento pode ocorrer em laboratório ou em campo. A pesquisa experimental em laboratório caracteriza-se pelo controle que o pesquisador tem sobre a representação da condição e sobre as variáveis do experimento. Seja em laboratório ou em campo a pesquisa experimental envolve o chamado “teste de hipóteses”, no qual o pesquisador formula uma ou mais hipóteses sobre o fenômeno e monta um experimento para a verificação da mesma, o teste de hipótese é conhecido como “método hipotético dedutivo” [70].

A hipótese tem como característica básica de ser uma suposta, provável e provisória resposta de um problema. Esta deve ser submetida à verificação para ser comprovada. Sua formulação mais comum se dá através de colocações

conjecturais da relação entre duas ou mais variáveis, sendo estas variáveis passíveis de serem medidas [71].

Baseando-se no método hipotético dedutivo foram formuladas as seguintes premissas para o sistema de monitoramento de dados.

**Premissa 1:** é possível executar a aquisição de dados da variável a ser monitorada com resolução adequada.

- Amostra: dados coletados com o sistema de monitoramento;
- Variável independente: velocidade de avanço programada;
- Variável dependente: resolução da variável monitorada.

**Premissa 2:** é possível identificar os parâmetros de programação CAM que resultem na otimização do processo

- Amostra: dados coletados com o sistema de monitoramento;
- Variável independente: métodos de interpolação da trajetória da ferramenta;
- Variável dependente 1: tempo de execução da usinagem;
- Variável dependente 2: média dos valores de velocidade de avanço;
- Variável dependente 3: comportamento da velocidade de avanço;
- Variável dependente 4: comportamento da aceleração dos eixos.

O procedimento experimental está dividido em duas etapas que estão descritas abaixo:

- Ensaios preliminares para avaliação da interface de comunicação;
- Ensaios para verificação das premissas formuladas.

Para a etapa de ensaios preliminares, executa-se a aquisição de valores da velocidade de avanço diretamente via rede de comunicação para o cálculo da taxa de aquisição do sistema e análise dos tempos entre leituras.

Para a etapa dos ensaios para verificação das premissas, o procedimento consiste em executar um programa NC de parte do corpo de prova da Associação Alemã de Comando Numérico (*NC-Gesellschaft*) e monitorar as variáveis de interesse conforme as condições estabelecidas para as premissas formuladas.

O desenvolvimento da proposta aqui discutida, seus testes e resultados são mostrados nos próximos capítulos.

## 4 Desenvolvimento do Sistema de Monitoramento

Neste capítulo é apresentado o desenvolvimento do sistema de monitoramento. As etapas de elaboração dos requisitos do software, da arquitetura do sistema e do diagrama de fluxo de dados foram feitas com o embasamento teórico do item 2.3 da revisão bibliográfica. Os ensaios iniciais que direcionaram o desenvolvimento das estratégias de monitoramento e os ensaios para verificação das premissas elaboradas no Capítulo 3 são também apresentados.

### 4.1 Requisitos de software

Conforme apresentado no item 2.3.1, a definição e análise dos requisitos de *software* determinam a ligação entre as fases de definição do sistema e projeto de *software*. Com base na classificação de requisitos apresentada no item 2.3.1, foram definidos os seguintes requisitos:

- Requisitos funcionais: monitoramento da velocidade de avanço e tempo de usinagem com resolução de dados adequada, para que através da análise de dados seja possível identificar a influência dos parâmetros de programação NC no desempenho do processo de usinagem;
- Requisitos não funcionais: limitação na taxa de aquisição de dados do sistema e limitação na quantidade de variáveis para manipulação de dados do CNC aberto;
- Requisitos do usuário: o sistema deve permitir a identificação de parâmetros ótimos da programação NC;
- Requisitos de sistema: o sistema deve fazer a aquisição de dados na seguinte condição: coletar 1 valor da variável velocidade de avanço por 1 milímetro percorrido pela ferramenta de corte. Os indicadores de desempenho (tempo de usinagem, velocidade de avanço e aceleração dos eixos) devem representar a influência dos parâmetros de programação NC na performance do processo.

A próxima etapa do desenvolvimento do sistema de monitoramento é o projeto do sistema, que é apresentado no próximo tópico.

### 4.2 Arquitetura do sistema

Conforme foi apresentado no item 2.3.2, a criação do modelo da arquitetura do sistema tem a proposta de mostrar como ocorre o relacionamento entre os

componentes do sistema e servir de referência para as próximas etapas do processo de desenvolvimento. Na Figura 4.1 é mostrado o diagrama de contexto de arquitetura desenvolvido para este trabalho. Para a elaboração desse diagrama foi utilizado o modelo de arquitetura proposto no item 2.3.2 da revisão bibliográfica.

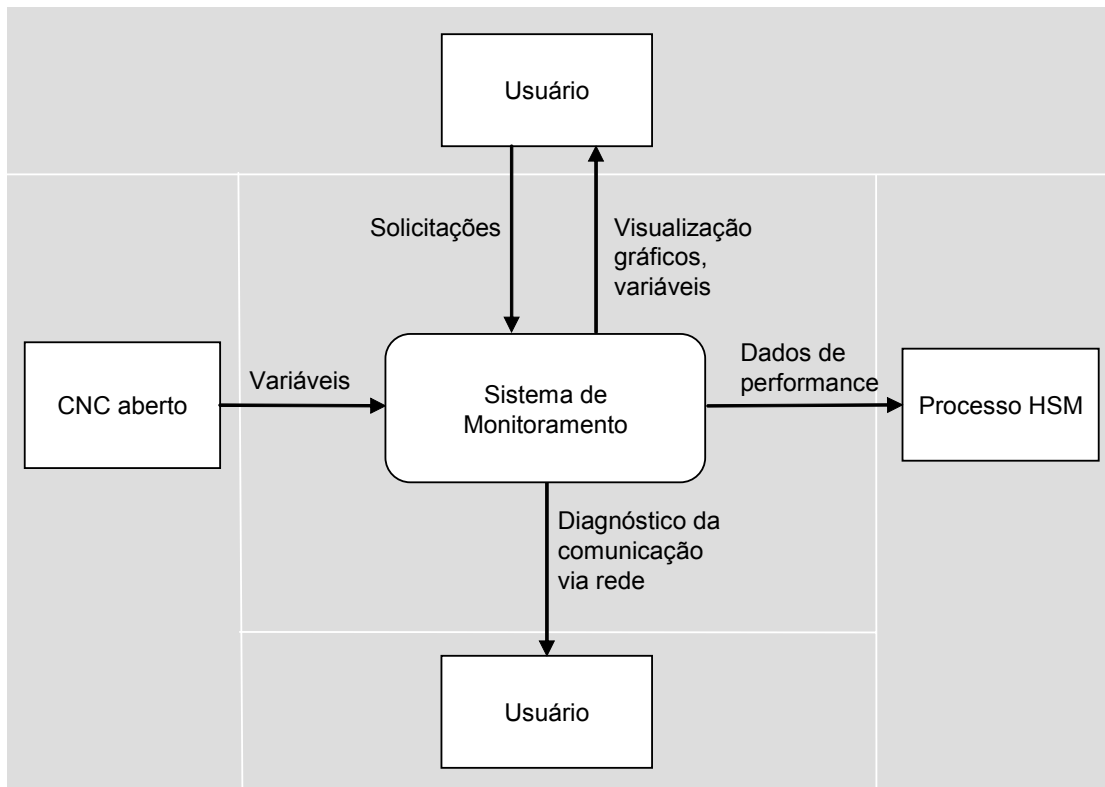


Figura 4.1: Diagrama de contexto de arquitetura do sistema

Conforme visto na Figura 4.1 o diagrama é constituído de regiões e seus respectivos componentes. A relação entre os componentes e suas funções com o sistema de monitoramento é ilustrada através de setas que representam o fluxo de informações. Abaixo são descritas as regiões do modelo de arquitetura com o respectivo componente do diagrama da Figura 4.1.

- Processamento de entrada: CNC aberto;
- Funções de processo e de controle: sistema de monitoramento;
- Processamento de interface com o usuário: usuário;
- Manutenção e auto-teste: usuário;
- Processamento de saída: processo HSM.

Os componentes de *hardware* a serem utilizados para o desenvolvimento do sistema são itens comerciais. O *software* Labview será utilizado para a criação do programa. Esse *software* possui grande aplicação na área de instrumentação e

controle de processos e seu funcionamento é baseado na lógica de fluxo de dados programáveis.

### 4.3 Diagrama de Fluxo de Dados (DFD)

No item 2.3.4 foi abordado o projeto estruturado de *software*, sendo o DFD uma ferramenta de auxílio para a análise do fluxo de informações e suas transformações. Nesse item são apresentados os DFDs iniciais e finais do sistema de monitoramento. O DFD inicial corresponde a versão do sistema utilizado nos ensaios preliminares que são descritos no item 4.4. Na Figura 4.2 é apresentado o DFD inicial.

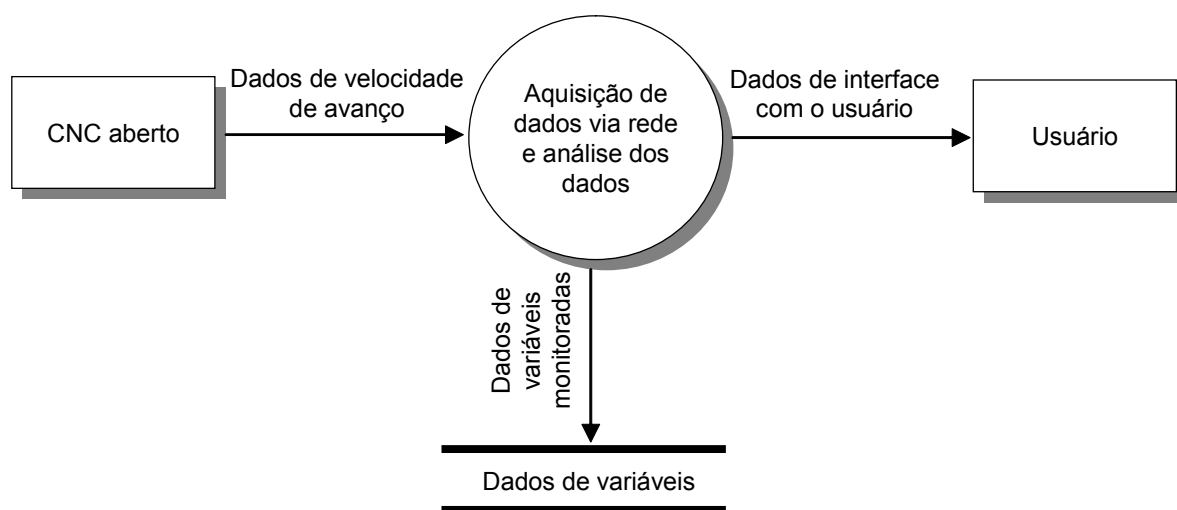


Figura 4.2: Diagrama de fluxo de dados inicial

Conforme visto na Figura 4.2 o DFD inicial é constituído da seguinte forma:

- Entidades externas: CNC aberto e Usuário;
- Processo: aquisição de dados via rede e análise dos dados;
- Depósito de dados: dados de variáveis.

O processo de aquisição de dados via rede e análise dos dados faz a leitura da variável a ser monitorada através de uma rede de comunicação. Através da interface de operação o usuário pode visualizar o gráfico de comportamento da variável monitorada e salvar os dados num arquivo texto.

Conforme mostra a análise dos resultados dos ensaios preliminares no item 5.1, foi necessário melhorar a taxa de aquisição do sistema. Para isso foi desenvolvido o DFD apresentado na Figura 4.3. Essa é a versão do DFD final do sistema que foi utilizado nos ensaios para verificação das premissas formuladas no item 3.2.



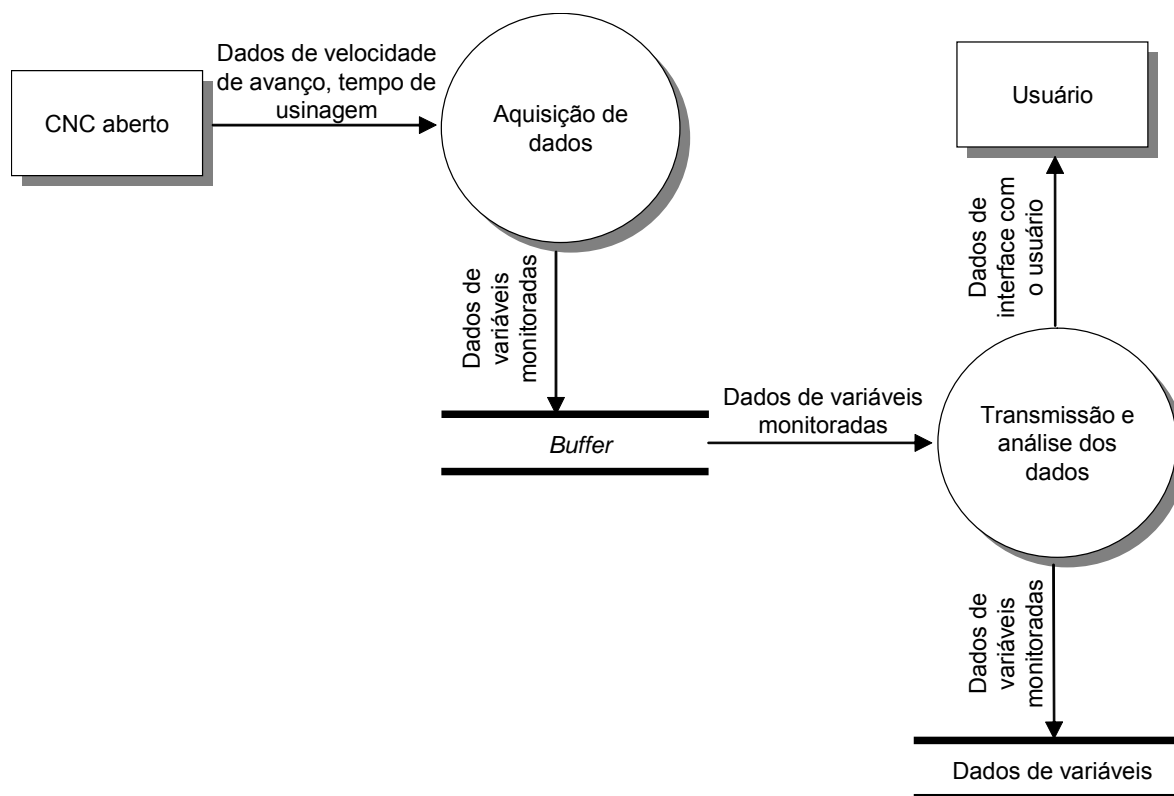


Figura 4.3: Diagrama de fluxo de dados final

O diagrama de fluxo de dados apresentado na Figura 4.3 é constituído dos seguintes componentes:

- Entidades externas: CNC aberto e usuário;
- Processos: aquisição de dados e transmissão e análise dos dados;
- Depósito de dados: *buffer* e dados de variáveis.

O processo de aquisição de dados faz a coleta de dados do *open* CNC e os armazena num *buffer*. O processo de transmissão e análise dos dados executa a coleta e processamento dos dados, e os valores de variáveis monitoradas podem ser armazenados num arquivo texto. O comportamento da variável monitorada e os índices de desempenho são apresentados através da interface com o usuário.

O módulo de aquisição de dados é implementado diretamente no programa NC e este é executado de forma automática em paralelo ao processamento do programa NC. O módulo de transmissão e análise dos dados é executado pelo usuário após a finalização do processo de usinagem. No item 4.5 é apresentado o funcionamento detalhado dos módulos de aquisição e de transmissão e análise dos dados.

O módulo de análise do sistema tem como princípio de funcionamento a comparação entre duas condições de parâmetros de programação NC. O usuário

executa dois programas NC com parâmetros diferentes e o sistema mostra qual condição proporcionou a otimização do processo. Caso o usuário deseje comparar mais condições, este deve salvar os dados obtidos para posterior comparação. Na Figura 4.4 é ilustrado o fluxograma de operação do sistema que considera a análise comparativa apresentada.

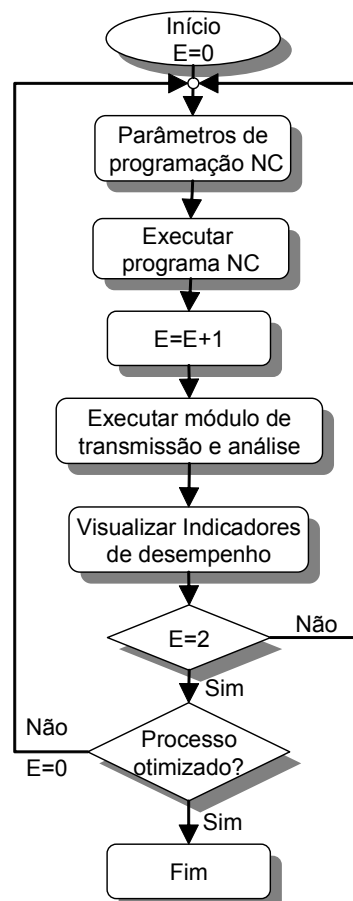


Figura 4.4: Fluxograma de operação do sistema de monitoramento

A Figura 4.4 ilustra a seqüência de operação do sistema. Nessa seqüência o usuário elabora um programa NC com os parâmetros de programação iniciais, executa o programa na máquina ferramenta, executa o módulo de transmissão e análise e visualiza os indicadores de performance. Necessariamente o usuário deve elaborar um segundo programa NC com parâmetros de programação alterados. Após o monitoramento da segunda condição o sistema indica qual foi a condição que resultou na otimização do processo.

A situação descrita acima é abordada no item 4.6 que corresponde aos ensaios para a verificação das premissas elaboradas no item 3.2

## 4.4 Ensaios preliminares

O sistema foi desenvolvido com a utilização do software LabVIEW<sup>(R)</sup> 7.1 da National Instruments. A comunicação do PC Intel Pentium III de 800 MHz e sistema operacional Windows XP com o CNC foi desempenhada pela placa CP5611 instalada no barramento PCI (*Peripheral Component Interconnect*) do PC. Na conexão da placa CP5611 com o CNC foi utilizado o cabo de rede MPI (*Multi Point Interface*), sendo esses três equipamentos da empresa SIEMENS [72]

A disponibilização dos dados para comunicação foi executada pelo NCDDE (*Dynamic data exchange with the NC kernel*), este servidor de dados faz parte do pacote OEM (*Original Equipment Manufacturer*) da SIEMENS [73].

Através das funções DDE (*Dynamic Data Exchange*) do LabVIEW é requisitada a leitura da variável de velocidade de avanço “vavactm” (este é o nome da variável que contém dados da velocidade real de avanço). Os softwares e equipamentos descritos acima são itens comerciais e estão disponíveis no Laboratório SCPM [72].

O comportamento da variável é visualizado através de um gráfico. Os valores da variável com o tempo gerado no contador em milissegundos são gravados em um arquivo texto separado por tabulações, sendo este arquivo a fonte de análise dos tempos relativos a taxa de aquisição determinada neste ensaio [72]. Na Figura 4.5 é apresentado o diagrama de blocos desenvolvido para os ensaios preliminares.

A leitura dos diagramas de blocos apresentados nas Figura 4.5, 4.7 e 4.8 deve ser feita da esquerda para a direita, seguindo-se a seqüência dos números indicados com suas respectivas funções. Essa é a lógica de funcionamento do fluxo de dados programáveis do Labview.

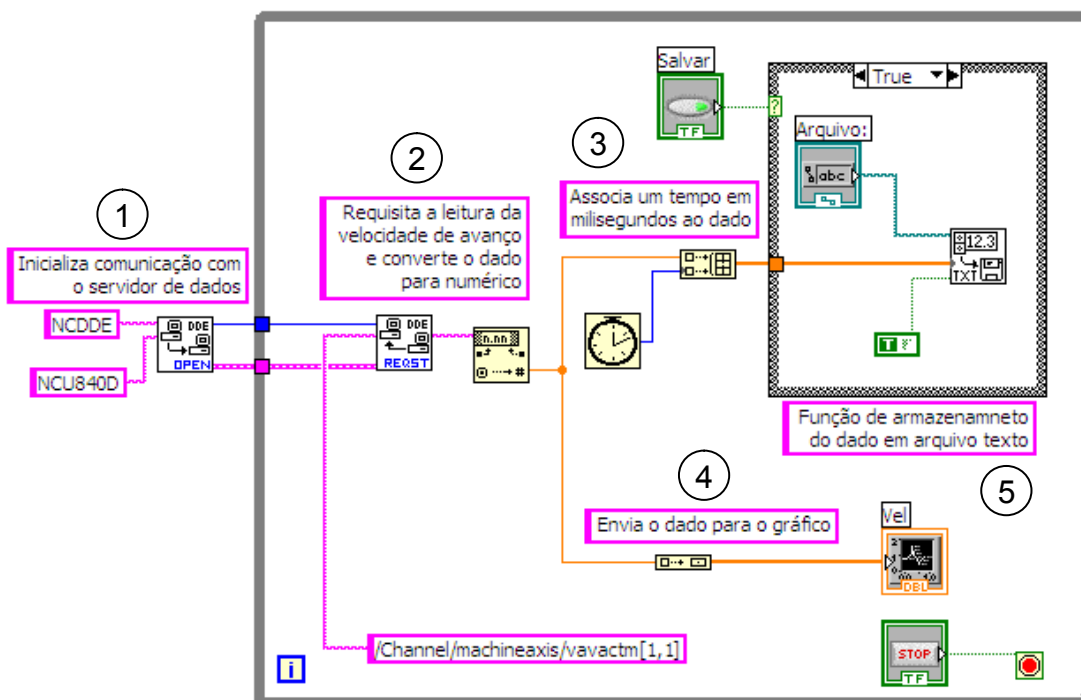


Figura 4.5 : Diagrama de blocos utilizado nos ensaios preliminares

O ensaio foi feito com a execução do sistema para leitura da variável “vavactm” do eixo X em um deslocamento de 100 mm com a velocidade de avanço de 1.000 mm/min no primeiro ensaio e 5.000 mm/min no segundo ensaio. Para o CNC 810D que possui velocidade de transmissão de dados de 187,5 Kbps e para o CNC 840D com velocidade de transmissão de 1,5 Mbps [73, 74]. Segue abaixo o programa NC utilizado nos ensaios [72]:

```
G54 D1
G0 X0
M00
G1 X100 F1000
M30
```

Conforme apresentado no Capítulo 5, devido à limitação causada pela frequência de aquisição de dados via rede de comunicação a pesquisa foi direcionada para o desenvolvimento de um módulo de aquisição de dados que foi programado diretamente no CNC aberto. O desenvolvimento do módulo de aquisição de dados é apresentado no item 4.5.1.

## 4.5 Estratégias de monitoramento

Nesse tópico são apresentadas as estratégias de aquisição, transmissão e análise dos dados desenvolvidos para o sistema de monitoramento.

#### 4.5.1 Módulo de aquisição de dados

A proposta do módulo de aquisição de dados é a criação de um *buffer* diretamente no CNC para armazenamento dos dados do nível de controle do servo. Dessa forma pretende-se evitar os atrasos de comunicação presentes na aquisição de dados via rede e aumentar a taxa de aquisição de dados.

A implementação dessa estratégia de aquisição de dados é feita através das funções de ações síncronas do CNC aberto utilizado nesse trabalho. Como definido no manual do fabricante [75] “As ações síncronas são instruções programadas pelo usuário que são avaliadas no ciclo de interpolação em sincronismo com a execução do programa NC. Se a condição programada nas ações síncronas é satisfeita ou caso não seja especificada nenhuma condição, então as ações determinadas pela instrução são ativadas em sincronismo com o programa NC que está em execução”.

Na Figura 4.6 é ilustrado o esquema de funcionamento das ações síncronas.

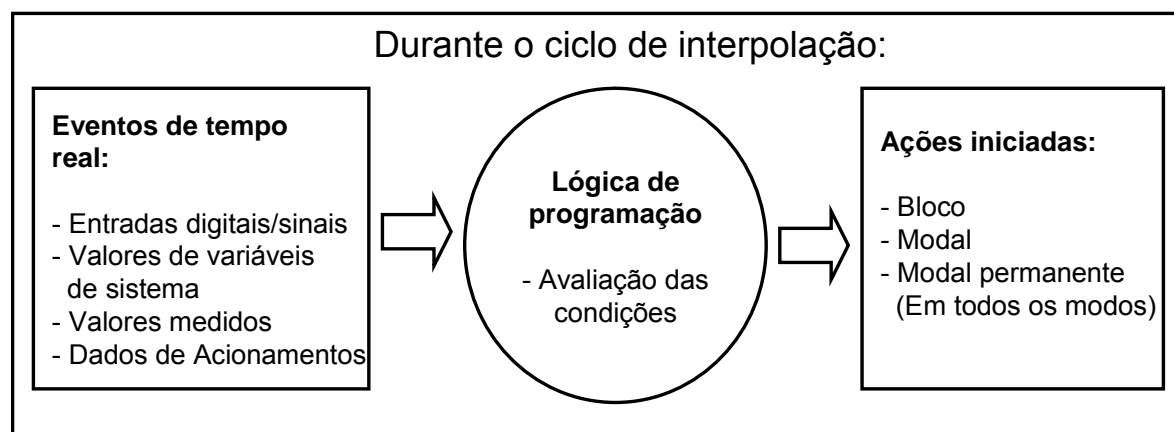


Figura 4.6: Esquema de funcionamento das ações síncronas [75].

A condição determinada na ação síncrona é avaliada no ciclo de interpolação e a ação ou ações são executadas conforme a condição é satisfeita. Para a construção do *buffer* foram utilizadas as variáveis do CNC chamadas de parâmetros R. Nos parâmetros R são armazenados os dados das variáveis de velocidade de avanço e tempo de usinagem. São 100 parâmetros R que podem ser expandidos até a quantidade de 1.000, porém não foi necessária essa expansão para os experimentos desse trabalho.

Abaixo é apresentada a programação da ação síncrona desenvolvida para o módulo de aquisição de dados. A ação síncrona é incluída no início do programa NC e ativada automaticamente durante a execução do programa [76].

```

ID=1 WHENEVER ($AA_IW[X1]>$AC_PARAM[1]) AND
($AA_IW[X1]<$AC_PARAM[2]) DO
$R[$AC_MARKER[1]]=SQRT($VA_VACTM[X1]*$VA_VACTM[X1]+$VA_VA
CTM[Z1]*$VA_VACTM[Z1]) $AC_MARKER[1]=$AC_MARKER[1]+1
$AC_PARAM[1]=$AC_PARAM[1]+1 $AC_PARAM[2]=$AC_PARAM[2]+1

```

A programação descrita acima é constituída da seguinte forma:

- ID=1: é o parâmetro de identificação da ação síncrona;
- WHENEVER (condição) DO: estabelece a condição para que a ação síncrona seja executada. Nesse caso, toda vez que a condição especificada entre parênteses for atingida a ação após a instrução DO é executada;
- DO ação: descreve qual será a ação a ser executada.

A lógica de funcionamento da ação síncrona desenvolvida é apresentada abaixo:

- Toda vez que a posição do eixo X for maior que o valor do parâmetro \$AC\_PARAM[1] e menor que o valor do parâmetro \$AC\_PARAM[2] é executada a ação que calcula o valor da velocidade de avanço resultante dos eixos X e Z e grava o resultado nos parâmetros R com o índice \$AC\_MARKER[1]. Os valores dos parâmetros condicionais são inicializados e incrementados para assegurar que a cada 1 milímetro de deslocamento do eixo X a ação seja executada. O índice dos parâmetros R também é incrementado para o armazenamento dos dados.

#### 4.5.2 Módulo de transmissão de dados

O módulo de transmissão de dados foi desenvolvido com o *software* Labview 7.1. A lógica de programação do Labview é baseada no fluxo de dados programáveis. Nessa lógica o usuário “monta” um diagrama de blocos com as funções de programação disponibilizadas pelo Labview e faz a conexão entre essas funções. A seqüência de execução do diagrama é da esquerda para a direita.

O módulo de transmissão de dados desenvolvido executa a comunicação entre o CNC e o PC conforme descrito no item 4.4. O programa em Labview foi alterado para ser executada a leitura dos dados contidos nos 100 parâmetros R do CNC.

A Figura 4.7 ilustra o diagrama de blocos desenvolvido no Labview para o módulo de transmissão dos dados [76].

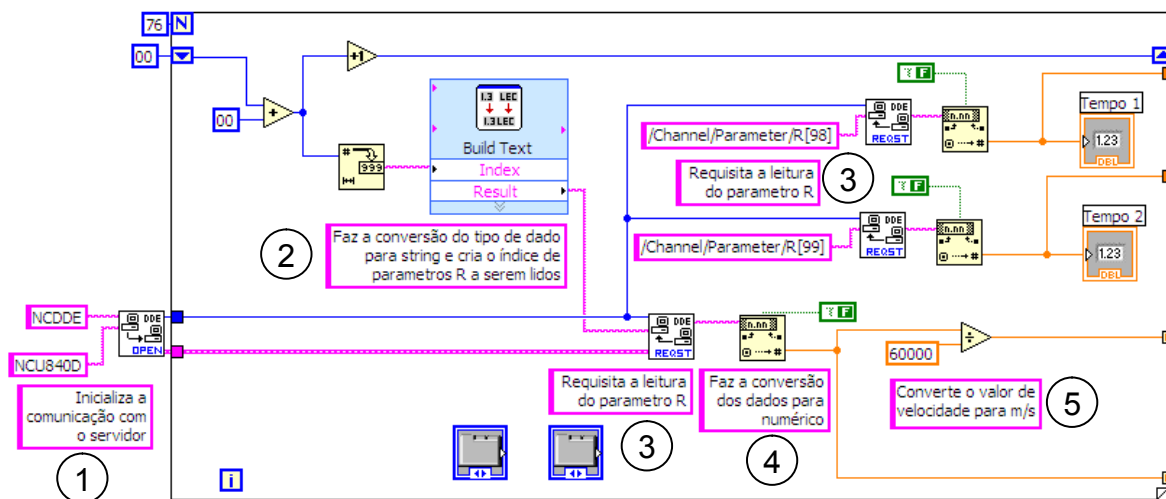


Figura 4.7: Diagrama de blocos do módulo de transmissão de dados [76]

Conforme visto na Figura 4.7 o diagrama de blocos do módulo de transmissão de dados funciona na seguinte seqüência lógica:

1. A função DDE *open* inicializa a comunicação com o servidor de dados NCDDE Server;
2. O dado com o tipo numérico é convertido para o tipo *string* e cria-se o índice de parâmetros R;
3. A função DDE *request* requisita a leitura dos valores contidos nos parâmetros R;
4. Os dados são convertidos para o tipo numérico;
5. Os valores de velocidade de avanço são convertidos para a unidade metros por segundo para o cálculo da derivada discreta no módulo de análise de dados.

As informações transmitidas para o PC são utilizadas pelo módulo de análise de dados, que é apresentado a seguir.

#### 4.5.3 Módulo de análise de dados

No módulo de análise de dados o comportamento do sinal monitorado é visualizado nos gráficos de velocidade de avanço e aceleração dos eixos. Os valores médio e mediano da velocidade de avanço, o valor do tempo de usinagem e a porcentagem de otimização do tempo são mostrados pela interface do usuário. O usuário tem a possibilidade de salvar os dados monitorados num arquivo texto.

Na Figura 4.8 é apresentada a parte do programa que corresponde ao módulo de análise de dados.

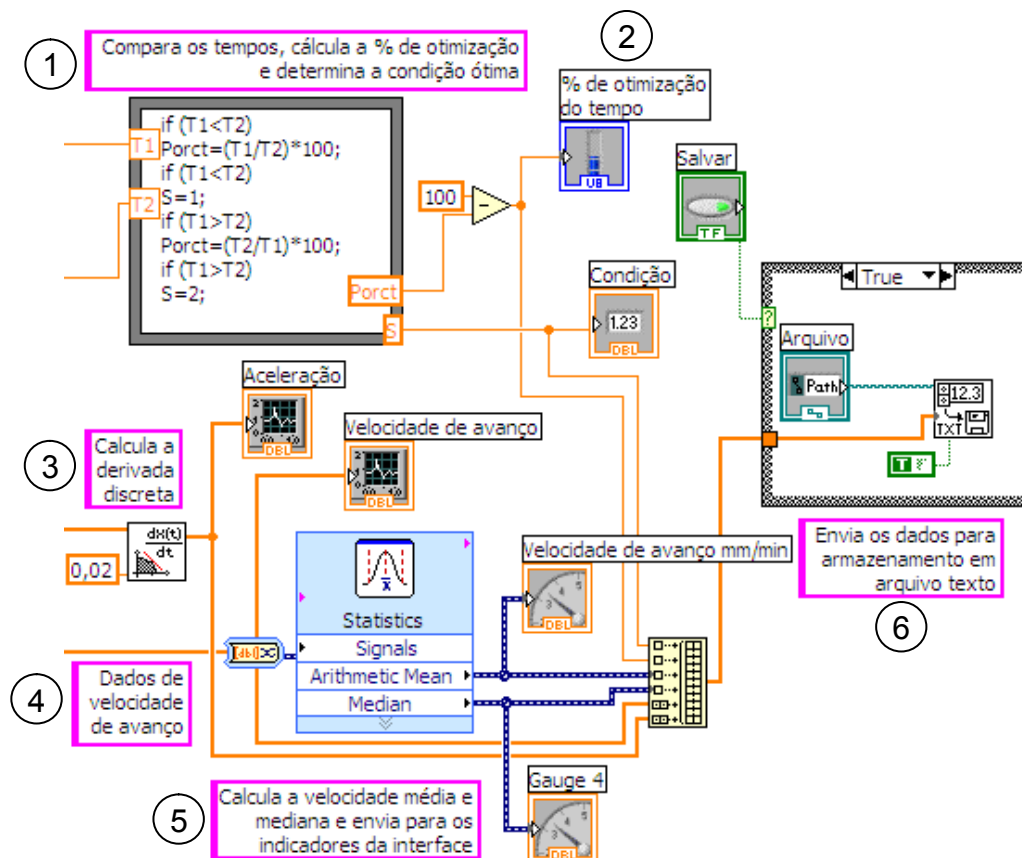


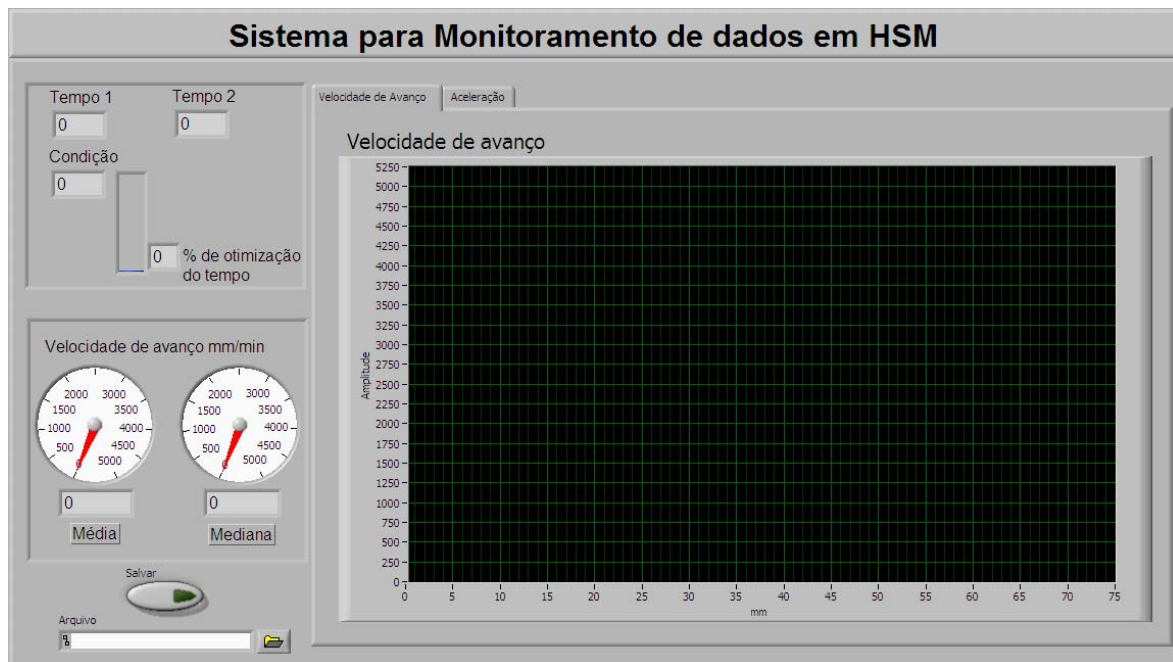
Figura 4.8: Módulo de análise de dados

Como visto na Figura 4.8, os dados do tempo de usinagem são utilizados para cálculo da porcentagem de otimização e determinação da condição otimizada. Os dados de velocidade de avanço são utilizados para cálculo da aceleração dos eixos. Esse cálculo é executado pela função de derivada discreta do Labview, o funcionamento dessa função será abordado no item 5.3. Os dados de velocidade de avanço e de aceleração são enviados para os respectivos gráficos e através das funções de estatística do Labview, são determinadas as velocidades média e mediana de avanço. A apresentação dos dados é feita pela interface com o usuário que é abordada a seguir.

#### 4.5.4 Interface com o usuário

A interface com o usuário foi desenvolvida no ambiente de programação do Labview destinado a criação da interface gráfica da programação. Nesse ambiente são disponibilizados controles e indicadores para a construção da interface. Os indicadores da condição otimizada (primeira ou segunda) e dos índices de performance, assim como do gráfico de comportamento da variável velocidade de avanço e aceleração dos eixos são visualizados na Figura 4.9.





*Figura 4.9: Interface com o usuário*

Como visto na Figura 4.9, é visualizado o indicador correspondente a condição otimizada de parâmetros de programação NC e os valores de tempo de usinagem, porcentagem de otimização do tempo, e velocidades média e mediana. No gráfico será visualizado o comportamento da velocidade de avanço e da aceleração dos eixos em relação a distância percorrida pelo eixo X da máquina ferramenta. Os dados da variável velocidade de avanço e dos indicadores de performance podem ser salvos em arquivos texto. Para que os dados sejam salvos deve-se especificar o caminho do arquivo e acionar o botão salvar.

A seguir serão apresentados os ensaios para a verificação das premissas formuladas no item 3.2.1.

## **4.6 Ensaios para verificação das premissas**

Para a verificação da premissa 1, o experimento foi executado num centro de usinagem de três eixos com o CNC Siemens 810D. Foi feito a aquisição de dados da variável de posição do eixo X, definida pela variável interna do CNC com o nome \$AA\_IW[X1]. O módulo de aquisição de dados foi executado em duas condições diferentes. Na primeira condição, a velocidade de avanço programada foi de 1.000 mm/min e na segunda condição foi de 6.000 mm/min [76]. Os parâmetros de programação NC utilizados nos ensaios foram elaborados com base nos estudos de Helleno [26] e Nunes [27].

O programa CNC apresentado abaixo foi utilizado para esse experimento [76].

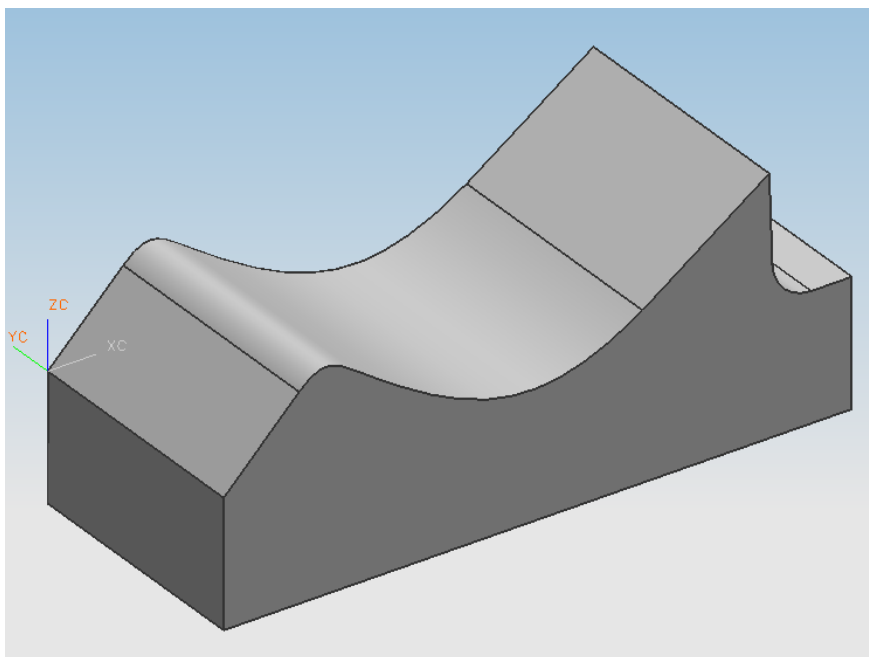
```
G54 D1
G1 X0 F1000
$AC_MARKER[1]=0
$AC_PARAM[1]=0
$AC_PARAM[2]=1
ID=1 WHENEVER ($AA_IW[X1]>$AC_PARAM[1]) AND
($AA_IW[X1]<$AC_PARAM[2]) DO $R[$AC_MARKER[1]]=$AA_IW[X1]
$AC_MARKER[1]=$AC_MARKER[1]+1 $AC_PARAM[1]=$AC_PARAM[1]+1
$AC_PARAM[2]=$AC_PARAM[2]+1
X100
M30
```

O programa NC executa um movimento linear no eixo X de 0 a 100 milímetros de distância e sempre que a condição descrita na ação síncrona é atingida, o valor de posição do eixo X é armazenado nos parâmetros R do CNC. A condição estabelecida assegura que um valor de posição do eixo X é gravado durante o intervalo de deslocamento de 1 milímetro do eixo X [76].

Para a verificação da premissa 2, foi elaborada a programação NC do corpo de prova da NCG apresentado na Figura 4.10. Os programas foram elaborados com *software Unigraphics NX 3*.

Para o ensaio foi utilizado o código de programação NC correspondente a uma faixa de deslocamento da operação de acabamento. Essa faixa corresponde a 1 deslocamento do eixo X da coordenada inicial até a coordenada final da peça.

Os dados monitorados de velocidade de avanço e aceleração são provenientes dos eixos X e Z.



*Figura 4.10: Modelo CAD utilizado para a programação NC*

Os parâmetros de programação NC alterados para a execução dos ensaios foram:

- Condição 1: interpolação linear e tolerância CAM de 0,005 mm;
- Condição 2: interpolação polinomial e tolerância CAM de 0,005 mm.

Os anexos 1 e 2 apresentam o código da programação NC que corresponde respectivamente as condições 1 e 2 apresentadas acima, e inclui a função de aquisição de dados do tempo de usinagem através do parâmetro \$AC\_TIMER[1].

## 5 Análise dos Resultados

Neste capítulo é apresentada a análise dos resultados dos ensaios preliminares e dos ensaios para a verificação das premissas elaboradas no item 3.2. O nível de acesso e customização de funções do CNC aberto é analisado no item 5.4.

### 5.1 Análise dos ensaios preliminares

A taxa de aquisição de dados do sistema é determinada pela quantidade de pontos (valores coletados) durante o intervalo de tempo de 1 segundo. A Equação 1 determina esse cálculo.

$$T_a = \frac{\Delta_p}{t} \quad (1)$$

Onde:

$T_a$  - Taxa de Aquisição [pontos/s];

$\Delta_p$  - Número de pontos;

$t$  - Tempo [s].

Conforme visto no item 4.4 os dados da variável velocidade de avanço e o tempo em milissegundos associado a cada dado são gravados num arquivo texto. Os dados gravados nesse arquivo foram utilizados para determinação da taxa de aquisição do sistema que foi calculada com a Equação 1. Os dados obtidos do cálculo e os tempos médio, mediano, mínimo e máximo entre leituras são apresentados na Tabela 5.1.

Descrição	810D F1000	810D F5000	840D F1000	840D F5000
Taxa (pontos/s)	9	8	56	60
$T_{\text{médio}}$ (s)	0,112	0,117	0,018	0,016
$T_{\text{mediano}}$ (s)	0,109	0,097	0,019	0,016
$T_{\text{mínimo}}$ (s)	0,055	0,066	0,008	0,011
$T_{\text{máximo}}$ (s)	0,215	0,219	0,039	0,022

Tabela 5.1: Dados coletados nos ensaios preliminares.

Conforme visto na Tabela 5.1 a taxa de aquisição de dados para o CNC 840D é maior. Isso ocorre devido à velocidade de transmissão da rede desse CNC ser 8

---

vezes maior do que a velocidade de transmissão do CNC 810D. Comparando as duas condições, é observada uma variação tanto na taxa de aquisição como no tempo médio entre leituras para ambos CNCs. Com isso pode-se verificar que ocorre uma influência do atraso de comunicação via rede e da operação do Labview que executa o armazenamento de dados no tempo entre leituras.

O comportamento dos tempos entre leituras para ambos comandos é analisado através da visualização da distribuição de tempos no gráfico de probabilidade normal. Esse gráfico é utilizado para verificar se a distribuição analisada possui a forma aproximada de uma distribuição normal. A representação característica de uma distribuição normal é constatada no caso dos pontos do gráfico seguirem aproximadamente uma linha reta [77].

Os gráficos de probabilidade normal apresentados nas Figuras 5.1, 5.2 e 5.3 foram obtidos com o *software* Matlab 7.4.0.

Conforme visto nas Figuras 5.1 e 5.2 os pontos do gráfico de probabilidade normal seguem aproximadamente uma linha reta, o que caracteriza de forma aproximada as distribuições como distribuições normais, no mínimo aproximadamente simétricas. Assim pode-se utilizar o tempo médio entre leituras como parâmetro de análise. No gráfico da Figura 5.3, a representação não é de uma distribuição normal, nesse caso foi adotada a mediana dos tempos entre leituras como parâmetro de análise.

Os gráficos obtidos também serviram para a análise comparativa apresentada a seguir.

Através da análise comparativa dos gráficos das Figuras 5.1 e 5.2, observa-se que para o comando com taxa de transmissão de 187,5 Kbps (ver Figura 5.1) não ocorre o empilhamento de pontos de leitura (que caracteriza a obtenção de tempos entre leituras iguais) e a distância entre leituras é irregular, ou seja, a resolução do sistema não é satisfatória com esta velocidade de transmissão [72].

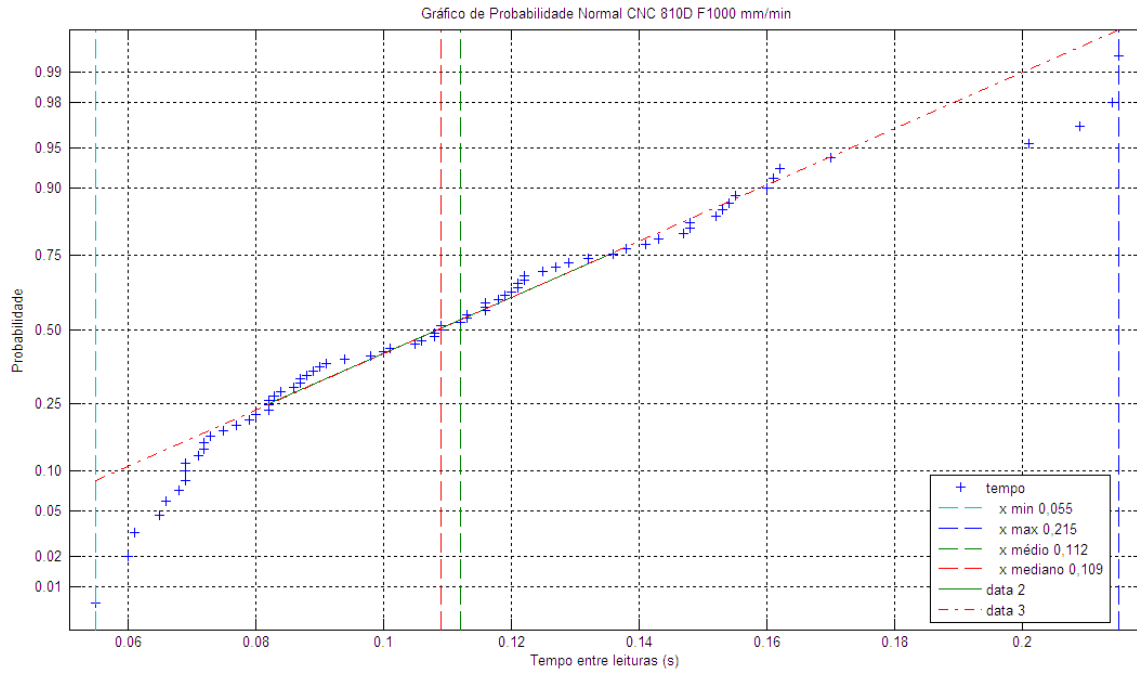


Figura 5.1: Gráfico de probabilidade normal com rede de 187,5 Kbps [72].

No comando com taxa de transmissão de 1,5 Mbps (ver Figura 5.2), ocorre a repetição de leituras com o mesmo tempo entre leituras, que é evidenciada pelo empilhamento de pontos. A resolução obtida entre os empilhamentos corresponde a resolução de 1 milissegundo do contador utilizado no sistema, demonstrando que esta velocidade de transmissão melhora a resolução da variável [72]. Para ambos os casos, o valor de velocidade de avanço programada é de 1.000 mm/min.

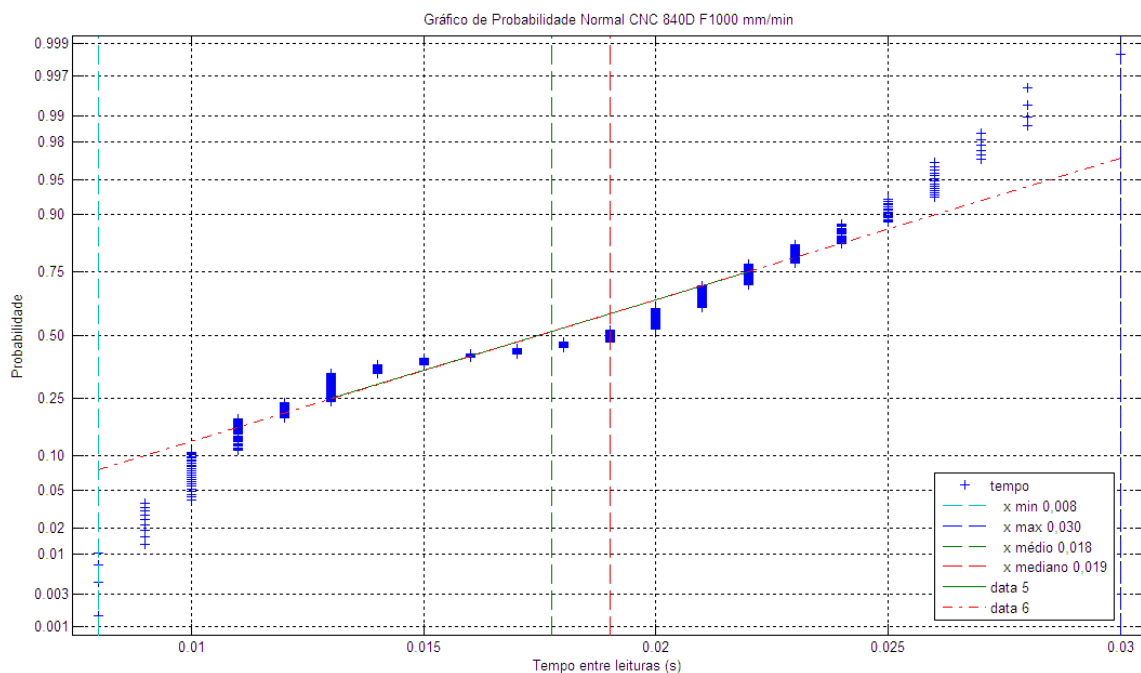


Figura 5.2: Gráfico de probabilidade normal com rede de 1.5 Mbps [72].

Para analisar a influência da operação de armazenamento de dados do Labview no tempo entre leituras, foi alterada a função que salva os dados e realizado novos ensaios somente para o CNC 840D. A função *Write To Spreadsheet File* (grava em arquivo de planilha) foi substituída pela função *Write File* (grava arquivo). Os resultados dos ensaios são mostrados na Tabela 5.2.

Descrição	840D F1000	840D F5000
Taxa (pontos/s)	143	139
$T_{\text{mediano}}$ (s)	0,006	0,007
$T_{\text{mínimo}}$ (s)	0,004	0,005
$T_{\text{máximo}}$ (s)	0,025	0,023

Tabela 5.2: Dados coletados nos ensaios preliminares com função *Write File*.

Conforme visto na Tabela 5.2 os resultados demonstram que a função *Write To Spreadsheet File* consumia em torno de 10 milissegundos a mais que a função *Write File* (ver Tabela 5.1 e 5.2, colunas 840D F1000 e 840D F5000, linha  $T_{\text{médico}}$  na Tabela 5.1 e linha  $T_{\text{mediano}}$  na Tabela 5.2). Essa influência não foi verificada para o CNC 810D, pois o resultado dessa influência na taxa de aquisição (ver Tabela 5.1) não será significativo.

O gráfico da Figura 5.3 corresponde a distribuição de tempos entre leituras obtida no CNC 840D com a nova função de armazenamento de dados.

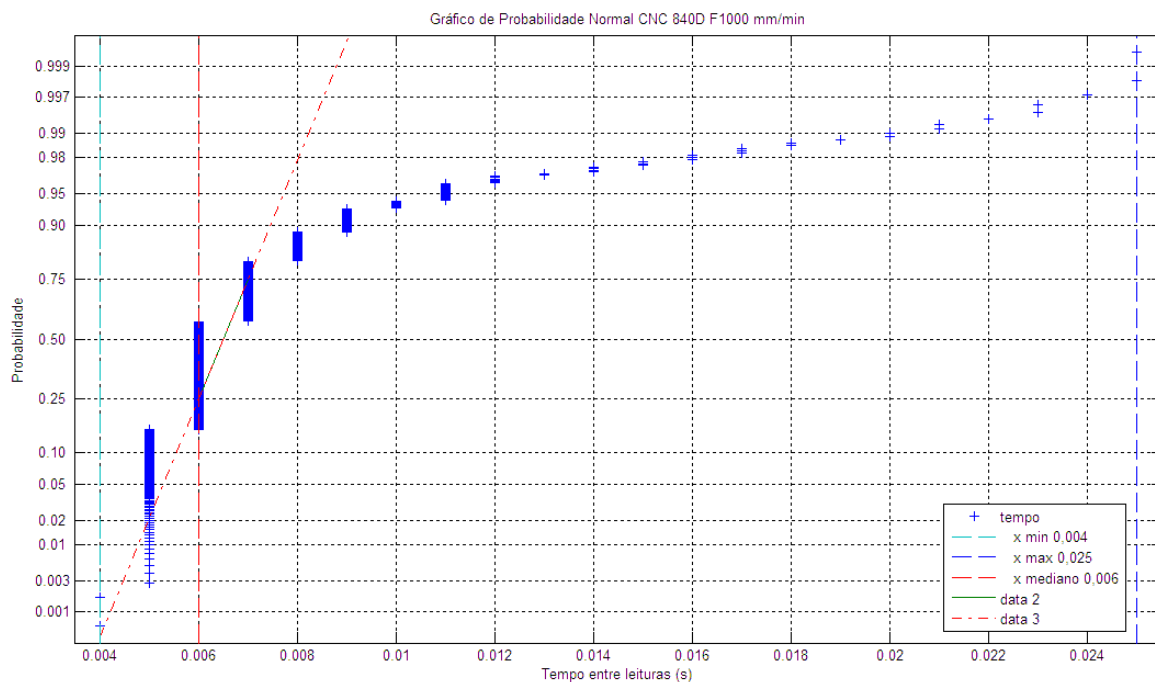


Figura 5.3: Gráfico de probabilidade normal rede de 1.5 Mbps e função *Write File*

Conforme visto na Figura 5.3 a distribuição obtida não é uma distribuição normal. Para esse caso foi adotado o valor da mediana de tempos entre leituras como dado de análise da influência da operação de armazenamento de dados nos tempos entre leituras. O valor de tempo mediano foi de 6 milisegundos. Isso demonstra que a operação de armazenamento de dados anteriormente utilizada consumia em torno de 10 milisegundos a mais.

A comparação do ganho de velocidade de transmissão da rede de comunicação entre o CNC 810D (187,5 Kbps) e o CNC 840D (1,5 Mbps) com o ganho obtido na taxa de aquisição de dados da Tabela 5.1 é determinada pela Equação 2. Os dados utilizados no cálculo do ganho de taxa de aquisição correspondem a condição de velocidade de avanço de 1.000 milímetros por minuto [72]:

$$G_v = \frac{1500}{187,5} = 8 \quad G_t = \frac{56}{9} = 6,2 \quad (2)$$

Onde:

$G_v$  - Ganho de velocidade de transmissão;

$G_t$  - Ganho de taxa de transferência.

O ganho de 8 vezes na velocidade de transmissão resultou no ganho de 6 vezes para a taxa de aquisição.

Conforme é apresentado na Equação 3 a velocidade de avanço é determinada pelo deslocamento em milímetros dividido pelo tempo em minutos (mm/min) [72].

$$V_a = \frac{\Delta_x}{t} \quad (3)$$

Onde:

$V_a$  - Velocidade de avanço [mm/min];

$\Delta_x$  - Deslocamento [mm];

$t$  - Tempo [s].

Para uma velocidade de avanço de 1.000 milímetros por minuto o sistema deve ter uma taxa de aquisição de 17 pontos por segundo para coletar no mínimo 1 ponto de leitura por 1 mm de deslocamento da ferramenta. A relação entre a Equação 1 e Equação 3 é utilizada para o cálculo do limite de velocidade de avanço para as taxas de aquisição obtidas nos CNCs 810D e 840D. Essa relação é apresentada a seguir:



$$T_a = \frac{\Delta_p}{t} \Rightarrow V_{amax} = \frac{\Delta_x}{t}$$

Onde:

$V_{amax}$  - Velocidade de Avanço Máxima [mm/min].

Para a taxa média de 9 pontos por segundo (ver Tabela 5.1) do CNC 810D a velocidade de avanço máxima para a aquisição de 1 ponto por milímetro percorrido é calculada através da conversão de unidade da taxa de aquisição de pontos por segundo para pontos por minuto, conforme mostra a Equação 4.

$$V_{amax} = T_a \times t \quad V_{amax} = 9 \times 60 \quad V_{amax} = 540 \text{ mm/min} \quad (4)$$

Com a taxa média máxima 56 pontos por segundo (ver Tabela 5.1) do CNC 840D, para a aquisição de no mínimo 1 ponto por milímetro percorrido a velocidade de avanço máxima deve ser [72]:

$$V_{amax} = 56 \times 60 \quad V_{amax} = 3.360 \text{ mm/min}$$

Os resultados dos ensaios preliminares direcionaram a pesquisa para o desenvolvimento de um módulo de aquisição de dados diretamente no CNC aberto. A seguir são apresentados os resultados obtidos com o módulo desenvolvido.

## 5.2 Taxa de aquisição e resolução da variável monitorada

A taxa de aquisição de dados para a programação do módulo de aquisição, diretamente no CNC aberto, está limitada ao tempo de ciclo do interpolador do CNC. No CNC utilizado o tempo de ciclo de interpolador está setado em 10 milissegundos, resultando em uma taxa de aquisição de 100 pontos por segundo. Porém, conforme foi apresentado no item 4.5.1, a condição estabelecida para a coleta de dados é que 1 dado da variável monitorada seja coletado por 1 mm de deslocamento do eixo X [76].

O intervalo entre leituras dos dados coletados através das ações síncronas são mostrados na Tabela 5.3 [76].

Velocidade de avanço	Intervalo entre leituras (mm)
F 1.000 mm/min	1
F 6.000 mm/min	1

Tabela 5.3. Velocidade de avanço e intervalo entre leituras [76]

Conforme mostrado na Tabela 5.3, identifica-se que o módulo de aquisição de dados assegura a condição especificada no ítem 4.1 para os requisitos de sistema.

Conforme abordado anteriormente, o tempo de ciclo do interpolador é o tempo mínimo para a aquisição de dados internos do CNC aberto que foi utilizado nos experimentos. Baseando-se na equação proposta por Arnone [31] para o cálculo do tempo de processamento de bloco, foi obtida a Equação 5 que determina a influência da velocidade de avanço programada na resolução da variável monitorada. Na Equação 5 o tempo de ciclo de interpolador substituiu o ítem tempo de processamento de bloco da fórmula original [76].

$$V_{amax} = \frac{\Delta_x}{TCI/60} \quad (5)$$

Onde:

- $V_{amax}$  - Velocidade máxima de avanço [mm/min];
- $\Delta_x$  - Distância entre aquisição de dados [mm];
- $TCI$  - Tempo de ciclo do interpolador [s].

Conforme a Equação 5, a velocidade máxima de avanço para assegurar a condição de aquisição de um valor da variável monitorada por um milímetro de deslocamento é calculada abaixo:

$$V_{amax} = \frac{1}{0,010/60} \quad V_{amax} = 6.000 \text{ mm/min}$$

Portanto, para o CNC 810D com tempo de ciclo do interpolador de 0,010 segundos a velocidade de avanço máxima é de 6.000 mm/min.

A Figura 5.3 mostra a influência de diferentes tempos de ciclo do interpolador na resolução da variável monitorada [76].

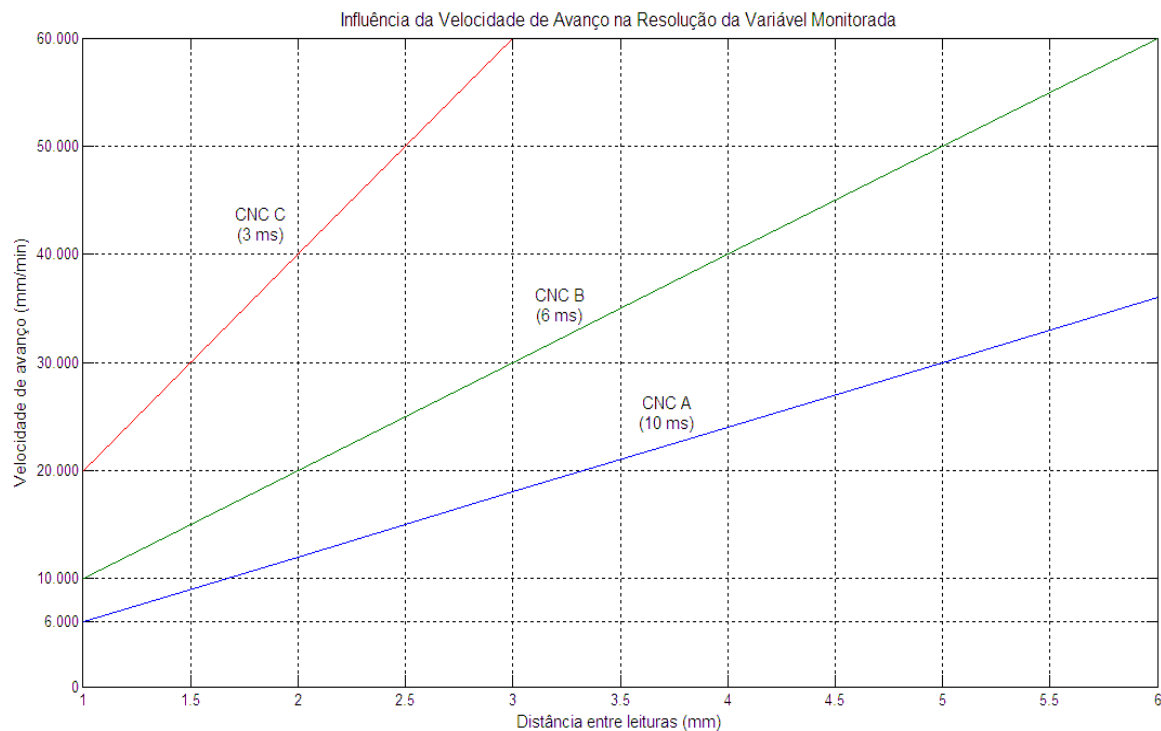


Figure 5.4. Influência da velocidade de avanço na resolução da variável [76]

Conforme visto na Figura 5.4 a resolução da variável monitorada diminui proporcionalmente com o aumento da velocidade de avanço. Para o exemplo do CNC B, com uma velocidade de avanço de 10.000 mm/min, a condição especificada é assegurada. Com o valor de 20.000 mm/min a resolução é afetada proporcionalmente. Nesse caso o dado é coletado a cada 2 mm de deslocamento do eixo X [76].

### 5.3 Identificação dos parâmetros ótimos de programação NC

Os parâmetros ótimos de programação NC foram identificados com os resultados dos experimentos descritos no item 4.6 para a verificação da premissa 2.

Os resultados em relação ao tempo de usinagem são apresentados na Tabela 5.4.

Interpolação	Tempo (s)
Linear	2,19
Polinomial	1,53

Tabela 5.4. Tempo de usinagem

Conforme visto na Tabela 5.4, o tempo de usinagem monitorado pelo sistema variou entre as duas condições de programação NC. A otimização do tempo de usinagem com a utilização da interpolação polinomial foi de 30% comparando-se com o tempo da interpolação linear.

Os resultados em relação à velocidade de avanço são apresentados na Tabela 5.5 e nas Figuras 5.4 e 5.5.

Interpolação	Velocidade média	Velocidade mediana
Linear	3.375 mm/min	3.498 mm/min
Polinomial	4.500 mm/min	4.953 mm/min

Tabela 5.5. Velocidade de avanço

Conforme visto na Tabela 5.5 a velocidade média e mediana de avanço aumentaram respectivamente em torno de 25% e 29% com a utilização da interpolação polinomial.

Nas Figuras 5.5 e 5.6 são apresentados o comportamento da velocidade de avanço monitorada pelo sistema.

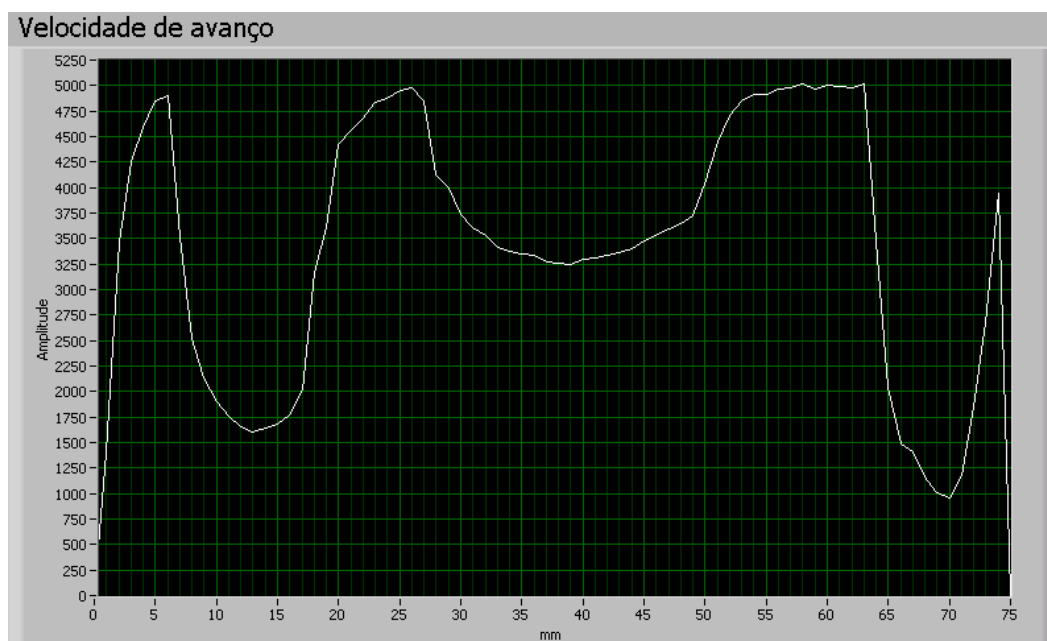


Figura 5.5: Comportamento da velocidade de avanço para interpolação linear

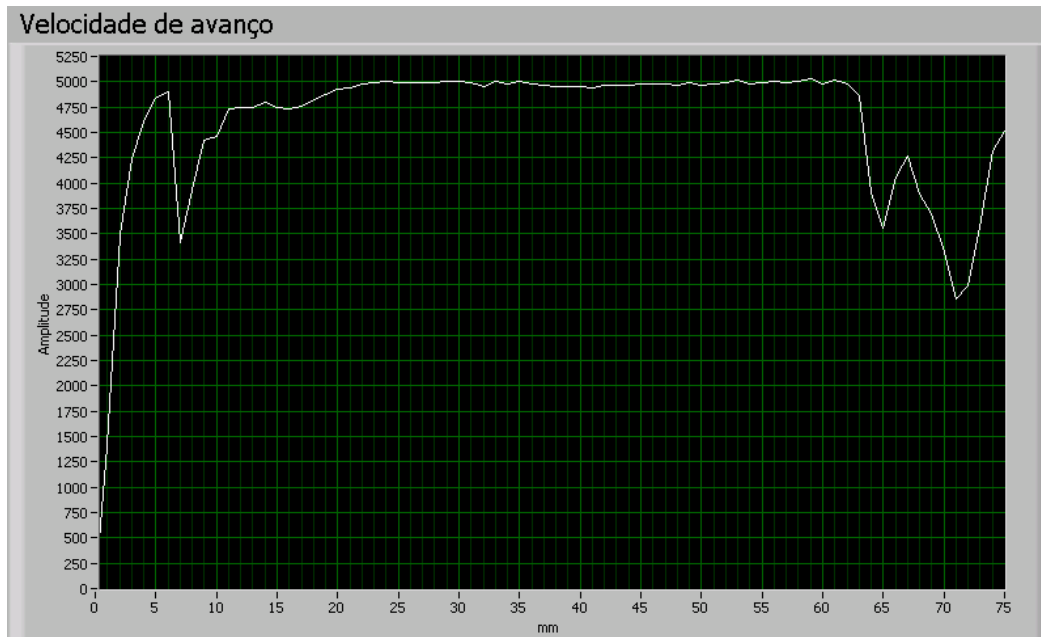


Figura 5.6: Comportamento da velocidade de avanço para a interpolação polinomial

Conforme visto nas Figuras 5.5 e 5.6 é identificado que o comportamento da velocidade de avanço se altera com os diferentes métodos de interpolação utilizados. Comparando-se os dois gráficos, é visualizado que com o uso da interpolação polinomial o comportamento da velocidade de avanço é mais estável e próximo do valor programado, no caso de 5.000 mm/min se comparado ao gráfico da interpolação linear.

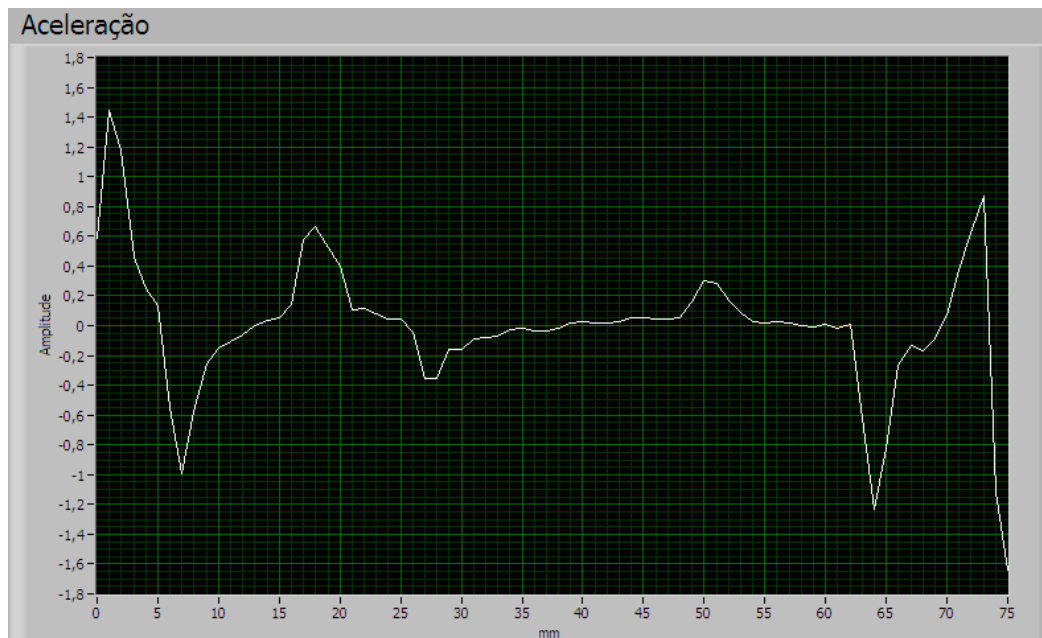
Nas Figuras 5.7 e 5.8 é apresentado o comportamento da aceleração obtida com o sistema. Para o cálculo da aceleração foi utilizada a função de derivada discreta do Labview. Essa função executa a derivada discreta dos dados de velocidade de avanço utilizando o método de aproximação centrada para a primeira derivada. Esse método de aproximação é mostrado na Equação 6 [78].

$$y'(x_i) \approx \frac{y_{i+1} - y_{i-1}}{2dt} \quad (6)$$

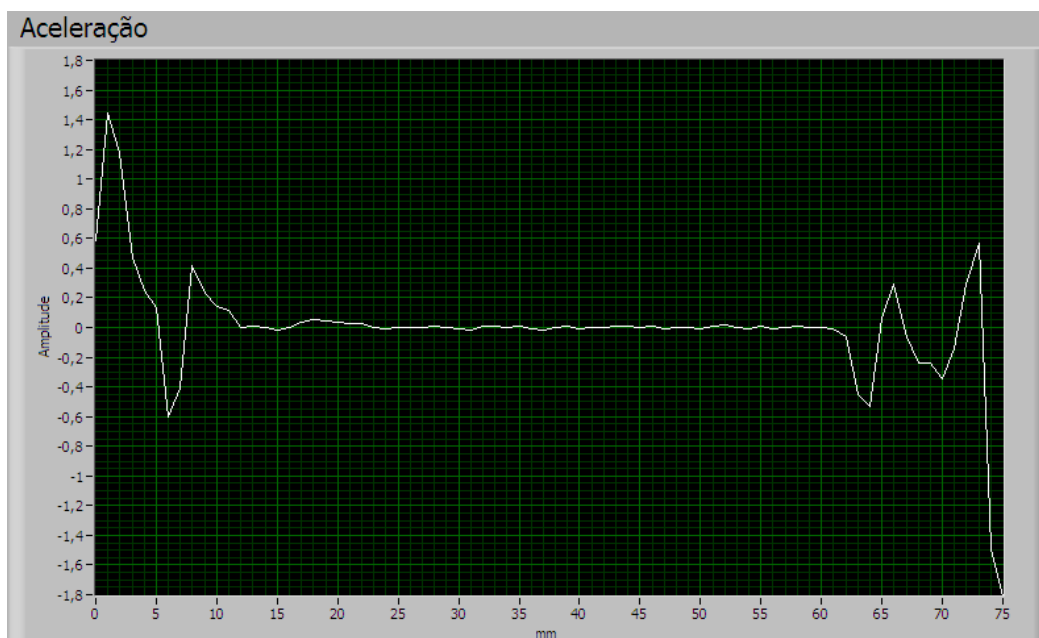
Segundo Ruggiero e Lopes [78] o erro devido à aproximação centrada é igual ao quadrado do intervalo de discretização. Para intervalo de discretização adotado de 0,020 s o erro da aproximação centrada é de 0,0004.

O intervalo de discretização foi obtido considerando-se o tempo médio entre leituras de dados de velocidade durante os 5 mm iniciais do corpo de prova. Esse trecho foi escolhido pois possui trajetória retilínea que resulta em comportamentos de velocidade e tempos entre leituras aproximadamente iguais para as duas condições de programação NC.

Para a determinação do intervalo de discretização foi feita a coleta dos dados do tempo entre leituras utilizando o módulo de aquisição de dados. A variável monitorada foi o contador de tempo `$AC_TIMER[1]` e os dados coletados foram salvos num arquivo texto. Os gráficos de aceleração para ambas condições de programação NC são apresentados a seguir.



*Figura 5.7: Comportamento da aceleração para a interpolação linear*



*Figura 5.8: Comportamento da aceleração para a interpolação polinomial*

Conforme visto nas Figuras 5.7 e 5.8 o comportamento da aceleração para os 5 milímetros iniciais do corpo de prova para ambas interpolações é praticamente o

mesmo. Isso ocorre devido a trajetória ser retilínea neste trecho. No decorrer da usinagem observa-se que o comportamento da aceleração para a interpolação polinomial (ver Figura 5.8) é suavizado. Essa suavização pode ser constatada através da comparação dos picos de aceleração e desaceleração. Comparando-se a região entre 13 e 60 milímetros, observa-se que a aceleração é aproximadamente constante para a interpolação polinomial (ver Figura 5.8).

Os resultados obtidos demonstram que o sistema tem a capacidade de identificar os parâmetros ótimos de programação NC através do monitoramento do tempo de usinagem e do comportamento da velocidade de avanço e da aceleração dos eixos.

#### **5.4 Acesso e customização de funções no CNC aberto**

Nesse tópico são analisados os níveis de acesso e de customização de funções do CNC aberto. As limitações constatadas podem variar para outros fabricantes de CNC, mas esse trabalho não tem o objetivo de fazer essa comparação. Como foi constatado na revisão bibliográfica, ainda não foi estabelecido um padrão de arquitetura aberta que seja amplamente aceito e utilizado pelos fabricantes de CNC.

O nível de acesso aos dados e de customização de funções está limitado ao nível do interpolador. Dessa forma a frequência máxima de manipulação dos dados é a frequência de trabalho do interpolador. Isso resulta na limitação da taxa de aquisição de dados e na implementação de sistemas de monitoramento em tempo real para usinagem com altas velocidades.

O nível permitido para a customização de funções também é o nível do interpolador. Para o desenvolvimento de funções nesse nível deve ser utilizada uma função de programação proprietária do fabricante do CNC, conforme visto no item 4.5.1.

A quantidade de variáveis internas e os recursos de programação oferecidos pelo CNC aberto podem limitar o desenvolvimento de funções customizadas. Para esse trabalho a quantidade de variáveis foi suficiente para a criação do *buffer* de aquisição de dados programado diretamente no CNC aberto.

No sistema de monitoramento desenvolvido o nível de abertura da arquitetura foi suficiente para o desenvolvimento do módulo de aquisição de dados. Os demais módulos foram desenvolvidos externamente com o *software* Labview. Isso demonstra que, dependendo da aplicação, existem oportunidades para o

desenvolvimento e implementação de sistemas de monitoramento de dados do CNC mesmo que este não seja totalmente aberto.



## 6 Conclusões e Sugestões para Trabalhos Futuros

Para os CNCs utilizados nos ensaios preliminares foi identificado que a velocidade de transmissão de dados via rede de comunicação influencia na taxa de aquisição de dados. Como o valor da taxa de aquisição de dados é constante, assim que a frequência associada a velocidade de avanço programada ultrapassa a frequência associada à taxa de aquisição a resolução da variável monitorada diminui.

A limitação de taxa de aquisição de dados evidenciada na situação acima direcionou a pesquisa para o desenvolvimento de um módulo de aquisição de dados diretamente no CNC aberto. Com essa estratégia de aquisição a influência da velocidade de transmissão de dados via rede foi eliminada.

Com o módulo de aquisição de dados desenvolvido a frequência de aquisição de dados aumentou em torno de 10 vezes. Isso diminui a influência da velocidade de avanço na resolução da variável monitorada mas não eliminou tal influência para valores altos de velocidade de avanço.

A condição de aquisição de dados especificada para a coleta de 1 valor da variável monitorada por 1 milímetro de deslocamento do eixo X é assegurada pelo sistema, desde que o valor de velocidade de avanço não ultrapasse o tempo mínimo para a coleta de dados que, para o CNC aberto utilizado, é de 0,010 s. Assim que a velocidade de avanço ultrapassa o tempo mínimo de aquisição de dados a resolução da variável monitorada diminui proporcionalmente.

Nos ensaios para verificação das premissas (vide cap. 3) constatou-se que o sistema pode identificar os parâmetros ótimos de programação NC através do monitoramento do tempo de usinagem, da velocidade de avanço e da aceleração dos eixos. O tempo de usinagem foi otimizado em 30% e as velocidades média e mediana de avanço foram otimizadas respectivamente em 25% e 29% com a mudança da interpolação linear para a interpolação polinomial.

A identificação dos parâmetros ótimos de programação NC também é visualizada através dos gráficos que descrevem o comportamento da velocidade de avanço. Comparando-se os gráficos pode-se visualizar que o comportamento da velocidade de avanço para a interpolação polinomial é mantido mais próximo ao valor de 5.000 mm/min (velocidade programada), em grande parte do deslocamento da trajetória de usinagem.

Através da aplicação da função de derivada discreta do Labview nos valores de velocidade de avanço foram obtidas as acelerações correspondentes as duas condições de programação NC. Nos gráficos de aceleração ficou evidenciado que o método de interpolação polinomial suaviza o perfil de aceleração, tornando-a mais estável quando comparada com a aceleração conseguida com a interpolação linear.

Foi constatado também que a otimização do processo HSM pode ser obtida através da identificação dos parâmetros ótimos de programação NC. A influência desses parâmetros no processo de usinagem foi demonstrada pelo monitoramento do tempo de usinagem e do comportamento da velocidade de avanço e da aceleração dos eixos.

O desenvolvimento de sistemas de monitoramento de dados de um controlador de arquitetura aberta depende do nível de abertura da arquitetura e do objetivo da aplicação do sistema. No presente trabalho foi verificado que para o nível de abertura do CNC utilizado os objetivos de monitoramento da velocidade de avanço e de identificação de parâmetros ótimos de programação NC foram atingidos dentro dos limites estabelecidos de valores de velocidade de avanço programada.

Mesmo utilizando-se os meios proprietários do fabricante do controlador de arquitetura aberta para o desenvolvimento de um sistema de monitoramento foram encontradas as limitações citadas acima e enquanto os fabricantes de controladores não permitirem o acesso e customização em todos os níveis de controle esse cenário não será modificado.

Para trabalhos futuros é sugerido os seguintes tópicos:

- Implementação da análise de outras variáveis (ex: potência, torque, corrente) do CNC aberto e incluir a programação de um *buffer* emanel para aumento da quantidade de dados a serem monitorados pelo sistema.
- Desenvolvimento de uma proposta de arquitetura aberta para o monitoramento de dados de sensores e atuadores inteligentes aplicados em controle distribuído.

## 7 Bibliografia Referenciada

- [1] LARTIGUE, C. et al. High-Performance NC for HSM by means of Polynomial Trajectories. *CIRP Annals 2005*, Krakow, Vol. 53, No. 1, Aug. 2004.
- [2] ALBERTÍ, M., CIURANDA J., RODRIGUEZ, C. A. Experimental analysis of dimensional error vs. cycle time in high-speed milling of aluminium alloy. *International Journal of Machine Tools & Manufacture*, Vol. 47, No. 2, p. 236 – 246, 2007.
- [3] TZENG, Y., CHEN, F. Optimization of the High-Speed CNC Milling Process Using Two-Phase Parameter Design Strategy by the Taguchi Methods. *JSME Int Journal. Ser C. Mech Systems, Mach Elem Manuf*, Vol. 48, No. 4, 2005.
- [4] DAGILOKE, I. F. et al. High-speed machining: an approach to process analysis. *Journal of Materials Processing Technology*, Vol. 54, No. 1, 1995.
- [5] SCHULZ, H., SCHÜTZER, K. Histórico da Usinagem com Altíssima Velocidade até os Dias Atuais. In: *Usinagem em Altíssimas Velocidades*. São Paulo: Editora Érica, 2003. Cap. 1, p. 13-28.
- [6] WANG, L., et al. Remote real-time CNC machining for web-based manufacturing. *Robotics and Computer-Integrated Manufacturing* Vol. 20, No. 6, p. 563–571, Dec. 2004.
- [7] HABER, R. E., ALIQUÉ, J. R. Fuzzy Logic-Based Torque Control System for Milling Process Optimization. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, Vol. 37, No. 5, 2007.
- [8] HABER, R. E., CANTILHO, K., JIMÉNEZ, J. E. Networked sensing for high-speed machining processes based on CORBA. *Sensors and Actuators*, Vol. 119, p. 418–426, Apr. 2005.
- [9] CUS, F.; MILFELNER, M.; BALIC, J. An intelligent system for monitoring and optimization of ball-end milling process. *Journal of Materials Processing Technology*, Maribor, Vol. 175, No. 1 – 3, p. 90 – 97, June 2006.
- [10] HABER, R. E. et al. Embedded fuzzy-control system for machining processes Results of a case study. *Computers in Industry*, Vol. 50, No. 3, p. 353-366, Apr. 2003.
- [11] PRITSCHOW, G., KRAMER, C. Open system architecture for drives. *CIRP Annals 2005*, Antalya, Vol. 54, No. 1, Aug. 2005.
- [12] PARK, S., KIM, S., CHO, H. Kernel software for efficiently building, re-configuring, and distributing an open CNC controller. *International Journal of Advanced Manufacturing Technology*, Vol. 27, No. 7-8, p. 788-796, 2006.

- 
- [13] EROL, N.A.; ALTINTAS, Y.; ITO, M.R. Open system architecture modular tool kit for motion and machining process control. *IEEE/ASME Transactions on Mechatronics*, Vol. 5, No. 3, p. 281 – 291, Sept. 2000.
- [14] PRITSCHOW, G., et al. Open Controller Architecture – Past, Present and Future. *Annals of the CIRP*, Vol. 50/2, p1-8, 2001.
- [15] YUN, W., MIN, B., PASEK, Z. Open Architecture Controller Interface Implementation for Machine Tools. In: IPROMS 2007 Innovative Production Machines and Systems. 2007.
- [16] SCHULZ, H. The history of high speed machining. *Ciência e Tecnologia*, Piracicaba, v.7, n. 13, p.97-105, jun.1999.
- [17] ALTINTAS, Y., ERKORKMAZ, K. Feedrate Optimization for Spline Interpolation in High Speed Machine Tools, *Annals of CIRP*, Vol. 52, No 1, 2003.
- [18] MONREAL, M., RODRÍGUEZ, C. Influence of Tool Path Strategy on the Cycle Time of High-speed Milling. *Computer-Aided Design*, Vol. 35, No. 4, 2003.
- [19] MEO, F. Algorithms for the optimized generation of trajectories through splines, and their integration into open architecture control systems. In: IPROMS 2007 Innovative Production Machines and Systems. 2007.
- [20] CHENG, M., TSAI, M. KUO, J. Real-time NURBS command generators for CNC servo controllers. *International Journal of Machine Tools & Manufacture*, Vol. 42, No. 7, 2002.
- [21] LEI, W. T., et al. Fast real-time NURBS path interpolation for CNC machine tools. *International Journal of Machine Tools & Manufacture*, Vol. 47, No. 10, 2007.
- [22] SOUZA, A. F.; COELHO, R. T. Experimental investigation of feed rate limitations on high speed milling aimed at industrial applications. *The International Journal of Advanced Manufacturing Technology*, Vol. 32, No. 11-12, May 2007.
- [23] KUMAR, S., et al. Process control in CNC manufacturing for discrete components: A STEP-NC compliant framework. *Robotics and Computer-Integrated Manufacturing*, Vol. 23, No. 6, 2007.
- [24] YAN X., et al. NC Program Evaluator for Higher Machining Productivity. *International Journal of Machine Tools and Manufacture*, Vol. 39, No.10, 1999.
- [25] SCHULZ, H., SPATH, D. Integration of Operator's Experience into NC Manufacturing. *Annals of the CIRP*, Vol. 46, No. 1, 1997.
- [26] HELLENO, A. L. Investigação de Métodos de Interpolação para Trajetória da Ferramenta na Usinagem de Moldes e Matrizes com Alta Velocidade. Dissertação (Mestrado) – Universidade Metodista de Piracicaba, 2004.

- 
- [27] NUNES, J. M., et al. Analysis of the Tool Path Interpolation Results on the High Feed Milling of Free Form Surfaces with Sharp Edges. In: *COBEM 2007 - 19th International Congress of Mechanical Engineering*, 2007, Brasília, DF. Proceedings of COBEM 2007.
- [28] ULSOY, A. G. Monitoring and control of machining. In: *Condition Monitoring and Control for Intelligent Manufacturing*. Berlin: Springer, 2006. Cap. 1, p. 1-27.
- [29] LIANG, S. Y., HECKER, R. L., AND LANDERS, R. G. Machining Process Monitoring and Control: The State-of-the-Art. *ASME Transactions - Journal of Manufacturing Science and Engineering*, Vol. 126, p. 297-310, 2004.
- [30] ULSOY, A. G.; KOREN, Y., RASMUSSEN, F. Principal Developments in the Adaptive Control of Machine Tools. *Journal of Dynamic Systems, Measurement and Control*, Vol. 105, p. 107 – 112, 1983.
- [31] CHIN, H.; DANAI, K. A Method of Fault Signature Extraction for Improved Diagnosis. *Journal of Dynamic Systems, Measurement and Control*, Vol. 113, p. 634 – 638, 1991.
- [32] HOHWIELER, E. AND GEISERT, C. Intelligent Machines Offer Condition Monitoring and Maintenance Prediction Services. In: Roberto Teti (Editor), Proceedings of the 4<sup>th</sup> CIRP International Seminar on Intelligent Computation in Manufacturing Engineering (CIRP ICME '04). Sorrento. 2004. p. 599-604.
- [33] ARNONE, A. *High Performance Machining*. USA, Cincinnati: Hanser Gardner Publications, 1998. ISBN 1-56990-246-1.
- [34] VALENTE, C. M. O.; FERRAZ, F., BRAGA, A. P. S.; OLIVEIRA, J. F. G. Monitoramento Remoto de Máquina-Ferramenta. *Revista Máquinas e Metais*, No. 497, p. 174-193, jun. 2007.
- [35] REHORN, A. G.; JIANG, J., ORBAN, P. E. State-of-the-art methods and results in tool condition monitoring: a review. *The International Journal of Advanced Manufacturing Technology*, Vol. 26, No. 7-8, 2005.
- [36] LEE, K. B. Smart Transducer Interface Standards for Condition Monitoring and Control of Machines. In: *Condition Monitoring and Control for Intelligent Manufacturing*. Berlin: Springer, 2006. Cap. 14, p. 347 – 372.
- [37] DESFORGES, X., ARCHIMÈDE, B. Multi-agent framework based on smart sensors/actuators for machine tools control and monitoring. *Engineering Applications of Artificial Intelligence*, Vol. 19, No. 6, 2006.
- [38] JANTUNEN, E. A summary of methods applied to tool condition monitoring in drilling. *International Journal of Machine Tools e Manufacture*, Vol. 42, No. 9, p. 997 – 1010, 2002.

- 
- [39] ROMERO-TRONCOSO, R., et al. Driver current analysis for sensorless tool breakage monitoring of CNC milling machines. *International Journal of Machine Tools e Manufacture*, Vol. 43, p. 1529 – 1534, 2003.
- [40] ELBESTAWI, M. A., DUMITRESCU, M., NG, E. S Tool Condition Monitoring in Machining. In: *Condition Monitoring and Control for Intelligent Manufacturing*. Berlin: Springer, 2006. Cap. 3, p. 55 – 80.
- [41] BHATTACHARYYA, P., SENGUPTA, D., MUKHOPADHYAY, S. Cutting force-based real-time estimation of tool wear in face milling using a combination of signal processing techniques. *Mechanical Systems and Signal Processing*, Vol. 21, No. 6, p. 2665 – 2683, 2007.
- [42] CHANG, H., et al. In-process surface roughness prediction using displacement signals from spindle motion. *International Journal of Machine Tools & Manufacture*, Vol. 47, No. 6, p. 1021 – 1026, 2007.
- [43] FERRAZ, F.; COELHO, R. T. Data acquisition and monitoring in machine tools with CNC of open architecture using internet. *The International Journal of Advanced Manufacturing Technology*, London, Vol. 26, No. 1 – 2, July 2005.
- [44] SILVA, E. J.; BIFFI, M. ; OLIVEIRA, J. F. G. The Development Of An Open Architecture Control System For CBN High Speed Grinding. *Journal of the Brazilian Society of Mechanical Sciences*, Brasil, Vol. 26, No. 1, p. 51-55, 2004.
- [45] RAMESH, R., MANNAN, M. A., POO, A. N. Tracking and contour error control in CNC servo systems. *International Journal of Machine Tools & Manufacture*, Vol. 45, No. 3, 2005.
- [46] KOREN, Y. *Computer Control of Manufacturing Systems*. USA, New York: McGraw-Hill Inc, 1983. ISBN:0070353417.
- [47] YEH, S., HSU, P. Adaptive-feedrate interpolation for parametric curves with a confined chord error. *Computer-Aided Design*, Vol. 34, No. 3, 2002.
- [48] LI, X. Real-Time Prediction of Workpiece Errors for a CNC Turning Centre, Part 3. Cutting Force Estimation Using Current Sensors. *The International Journal of Advanced Manufacturing Technology*, Vol. 17, No. 9, 2001.
- [49] MILFEINER, M., CUS, F., BALIC, J. An overview of data acquisition system for cutting force measuring and optimization in milling. *Journal of Materials Processing Technology*, Vol. 164-165, 2005.
- [50] AL-REGIB, E., NI, J., LEE, S. Programming spindle speed variation for machine tool chatter suppression. *International Journal of Machine Tools and Manufacture*, Vol. 43, No. 12, 2003.

- 
- [51] TOH, C. K. Vibration analysis in high speed rough and finish milling hardened steel. *Journal of Sound and Vibration*, Vol. 278, No. 1-2, 2004.
- [52] ALTINTAS, Y., KO, J. H. Chatter Stability of Plunge Milling. *CIRP Annals - Manufacturing Technology*, Vol. 55, No. 1, 2006.
- [53] LEE, D., ALLAN, J., BANNETT, S. Distributed Real-Time Control Systems using CAN. In: *Fieldbus Technology*. Berlin: Springer, 2003. Cap. 15, p. 357 – 385.
- [54] CHENG, T. et al. Intelligent Machine Tools in a Distributed Network Manufacturing Mode Environment. *The International Journal of Advanced Manufacturing Technology*, Vol. 17, No. 3, 2001.
- [55] ALTINTAS, Y. Manufacturing Automation: Metal Cutting Mechanics, Machine Tool Vibrations, and CNC Design. USA, New York: Cambridge University Press, 2000.
- [56] LARSE, R. M. The SERCOS Interface for Digital Motion Control. In: *Fieldbus Technology: Industrial Network Standards for Real-Time Distributed Control*. Berlin: Springer, 2003. Cap. 8, p. 183 – 214.
- [57] IEEE Standards products catalog, 1998, IEEE/NEMI PR 1533-1998, IEEE/NEMI Low-Cost Open Architecture Controller Specification.
- [58] PFLEEGER, S. L. Engenharia de Software: teoria e prática. São Paulo: Prentice Hall, 2004.
- [59] PRESSMAN, R. S. Engenharia de Software. São Paulo: Makron Books, 1995.
- [60] SOMMERVILLE, I. Engenharia de Software. São Paulo: Addison Wesley, 2003.
- [61] BOURQUE, P., DUPUIS, R. (EDS.). Guide to the Software Engineering Body of Knowledge (SWEBOK). *IEEE Computer Society*, 2004. Disponível em: <<http://www.swebok.org>>. Acesso em 08 nov. 2007.
- [62] PETERS, J. F., PEDRYCZ, W. Engenharia de Software: Teoria e Prática. Rio de Janeiro: Campus, 2001.
- [63] SCHACH, S. R. Testing: principles and practice. *ACM Computing Surveys*, Vol. 28, No. 1, 1996.
- [64] MORAES, C. C., CASTRUCCI, P. L. Engenharia de Automação Industrial. 2ª ed. Rio de Janeiro: LTC, 2007.
- [65] ALVES, J. L. L. Instrumentação, Controle e Automação de Processos. Rio de Janeiro: LTC, 2005.
- [66] MURUGESHAN, R. Fieldbus and Contemporary Standards. In: *Fieldbus Technology: Industrial Network Standards for Real-Time Distributed Control*. Berlin: Springer, 2003. Cap. 1, p. 1 – 36.

- 
- [67] THOMESSE, J. Fieldbus Technology in Industrial Automation. *Proceedings of the IEEE*, Vol. 93, No. 6, 2005.
- [68] Zhang, X. L. et al. Hierarchical Real-time Networked CNC System based on the Transparent Model of Industrial Ethernet. *The International Journal of Advanced Manufacturing Technology*, Vol. 34, No. 1-2, 2007.
- [69] GIL, A. C. Como Elaborar Projetos de Pesquisa. São Paulo: Atlas, 1996.
- [70] MOREIRA, D.A. O método fenomenológico na pesquisa. São Paulo : Pioneira Thomson , 2002.
- [71] LAKATOS, E. M. Metodologia Científica. São Paulo: Atlas, 2000.
- [72] DEL CONTE, E. G., SCHUTZER, K., HELLENO, A. L. Análise da Influência da Velocidade de Avanço no Monitoramento de Dados do CNC. In: *4º Congresso Brasileiro de Engenharia de Fabricação, 2007*, São Pedro. Anais do 4º Congresso Brasileiro de Engenharia de Fabricação, 2007.
- [73] SIEMENS A.G., Sinumerik 840D / 810D / FM-NC OEM package MMC User's Manual, 12ª ed., 1997.
- [74] SIEMENS A.G., Sinumerik 810D Installation and Start-Up Guide Manual, 11ª ed., 2002.
- [75] SIEMENS A.G., Sinumerik 840D/840Di/810D Synchronized Actions, 08ª ed., 2005.
- [76] DEL CONTE, E. G., SCHUTZER, K. Data Acquisition Strategy for Open CNC Monitoring in HSM Process. In: *COBEM 2007 - 19th International Congress of Mechanical Engineering, 2007*, Brasília, DF. Proceedings of COBEM 2007.
- [77] FREUND, J. E., SIMON, G. A. Estatística Aplicada: Economia, Administração e Contabilidade. Porto Alegre: Bookman, 2000.
- [78] RUGGIERO, M. A. G., LOPES, V. L. R. Cálculo Numérico: Aspectos Teóricos e Computacionais. 2ª ed. São Paulo: Pearson Education do Brasil, 1996.



## Bibliografia Consultada

- OMIROU, S. L.; BAROUNI, A. K. Integration of new programming capabilities into a CNC milling system. *Robotics and Computer-Integrated Manufacturing*, Vol. 21, p. 518–527, 2005.
- YIH-FONG, T; MING-DER, J. Dimensional quality optimisation of high-speed CNC milling process with dynamic quality characteristic. *Robotics and Computer-Integrated Manufacturing*, Vol. 21, p. 506–517, 2005.
- ALVES, M.; TOVAR, E. Engineering PROFIBUS networks with heterogeneous transmission media. *Computer Communications*, Vol. 30, No. 1, p. 17 - 31, Julho de 2006.
- FERRAZ JUNIOR, F. Desenvolvimento de um sistema de monitoramento e supervisão para o processo de torneamento. Dissertação (Mestrado) – Escola de Engenharia de São Carlos, Universidade de São Paulo, 2002.
- HABER, R. E., ET AL. An investigation of tool-wear monitoring in a high-speed machining process. *Sensors and Actuators*, Vol. 116, No. 3, p. 539–545, Outubro de 2004.
- PUPO, M. S. Interface homem-máquina para supervisão de um CLP em controle de processos através da WWW. Dissertação (Mestrado) – Escola de Engenharia de São Carlos, Universidade de São Paulo, 2002, 33p.
- SWAIN, N. K. ET AL. Remote Data Acquisition, Control and Analysis using LabVIEW Front Panel and Real Time Engine. *Proceedings IEEE SoutheastCon 2003*, Orangeburg, 2003.
- TSENG, P. C., CHOU, A. The intelligent on-line monitoring of end milling. *International Journal of Machine Tools & Manufacture* Vol. 42, p. 89–97, 2002.
- ZONGYU, C., ET AL. The study of configuration-style CNC system based on CANBUS. *International Journal of Advanced Manufacturing Technology*, Vol. 28, p. 1129-1135, 2006.

## Anexo 1 – Programa NC com interpolação linear

```
N1 G17 G71 G90 G64 G94
N2 G0 G53 Z-110 D0 M5
N3 T02
N4 M06
N5 G54 D1
N6 G54 D01 S6000 M03
N7 G00 X-2.121 Y.001
N8 CFTCP
N9 G00 Z-11.636 M08
N10 G17 G01 X0 Y.001 Z-15.878 F5000
$AC_MARKER[1]=0
$AC_PARAM[1]=0
$AC_PARAM[2]=1
$AC_TIMER[1]=0
ID=1 WHENEVER ($AA_IW[X1]>$AC_PARAM[1]) AND ($AA_IW[X1]<$AC_PARAM[2])
DO
$R[$AC_MARKER[1]]=SQRT($VA_VACTM[X1]*$VA_VACTM[X1]+$VA_VACTM[Z1]*$VA
_VACTM[Z1]) $AC_MARKER[1]=$AC_MARKER[1]+1
$AC_PARAM[1]=$AC_PARAM[1]+1 $AC_PARAM[2]=$AC_PARAM[2]+1
N11 G01 X6.314 Y.001 Z-7.444
N12 G01 X6.916 Y.001 Z-6.84
N13 G01 X7.218 Y.001 Z-6.541
N14 G01 X7.519 Y.001 Z-6.262
N15 G01 X7.82 Y.001 Z-6.002
N16 G01 X8.121 Y.001 Z-5.761
N17 G01 X8.423 Y.001 Z-5.54
N18 G01 X8.724 Y.001 Z-5.337
N19 G01 X9.025 Y.001 Z-5.152
N20 G01 X9.326 Y.001 Z-4.985
N21 G01 X9.628 Y.001 Z-4.836
N22 G01 X9.929 Y.001 Z-4.705
N23 G01 X10.23 Y.001 Z-4.59
N24 G01 X10.531 Y.001 Z-4.492
N25 G01 X10.833 Y.001 Z-4.409
N26 G01 X11.134 Y.001 Z-4.343
N27 G01 X11.435 Y.001 Z-4.293
N28 G01 X11.736 Y.001 Z-4.259
N29 G01 X12.038 Y.001 Z-4.241
N30 G01 X12.339 Y.001 Z-4.239
N31 G01 X12.64 Y.001 Z-4.252
N32 G01 X12.941 Y.001 Z-4.281
N33 G01 X13.243 Y.001 Z-4.326
N34 G01 X13.544 Y.001 Z-4.387
```

---

N35 G01 X13.845 Y.001 Z-4.463  
N36 G01 X14.146 Y.001 Z-4.554  
N37 G01 X14.448 Y.001 Z-4.659  
N38 G01 X14.749 Y.001 Z-4.779  
N39 G01 X15.05 Y.001 Z-4.913  
N40 G01 X15.351 Y.001 Z-5.061  
N41 G01 X15.653 Y.001 Z-5.222  
N42 G01 X15.954 Y.001 Z-5.394  
N43 G01 X16.556 Y.001 Z-5.768  
N44 G01 X17.159 Y.001 Z-6.174  
N45 G01 X19.569 Y.001 Z-7.842  
N46 G01 X20.774 Y.001 Z-8.654  
N47 G01 X21.979 Y.001 Z-9.443  
N48 G01 X23.184 Y.001 Z-10.204  
N49 G01 X24.389 Y.001 Z-10.932  
N50 G01 X25.594 Y.001 Z-11.621  
N51 G01 X26.799 Y.001 Z-12.265  
N52 G01 X27.401 Y.001 Z-12.569  
N53 G01 X28.004 Y.001 Z-12.86  
N54 G01 X28.606 Y.001 Z-13.137  
N55 G01 X29.209 Y.001 Z-13.399  
N56 G01 X29.811 Y.001 Z-13.646  
N57 G01 X30.414 Y.001 Z-13.876  
N58 G01 X31.016 Y.001 Z-14.089  
N59 G01 X31.619 Y.001 Z-14.285  
N60 G01 X32.221 Y.001 Z-14.462  
N61 G01 X32.824 Y.001 Z-14.619  
N62 G01 X33.426 Y.001 Z-14.756  
N63 G01 X34.029 Y.001 Z-14.872  
N64 G01 X34.631 Y.001 Z-14.965  
N65 G01 X35.234 Y.001 Z-15.036  
N66 G01 X35.836 Y.001 Z-15.082  
N67 G01 X36.439 Y.001 Z-15.104  
N68 G01 X37.041 Y.001 Z-15.101  
N69 G01 X37.644 Y.001 Z-15.071  
N70 G01 X38.246 Y.001 Z-15.016  
N71 G01 X38.849 Y.001 Z-14.936  
N72 G01 X39.451 Y.001 Z-14.833  
N73 G01 X40.054 Y.001 Z-14.706  
N74 G01 X40.656 Y.001 Z-14.557  
N75 G01 X41.259 Y.001 Z-14.387  
N76 G01 X41.861 Y.001 Z-14.194  
N77 G01 X42.464 Y.001 Z-13.981  
N78 G01 X43.066 Y.001 Z-13.749  
N79 G01 X43.669 Y.001 Z-13.498  
N80 G01 X44.271 Y.001 Z-13.228

N81 G01 X44.874 Y.001 Z-12.94  
N82 G01 X45.476 Y.001 Z-12.634  
N83 G01 X46.079 Y.001 Z-12.311  
N84 G01 X46.681 Y.001 Z-11.972  
N85 G01 X47.284 Y.001 Z-11.616  
N86 G01 X47.886 Y.001 Z-11.246  
N87 G01 X48.489 Y.001 Z-10.853  
N88 G01 X62.949 Y.001 Z-.864  
N89 G01 X63.551 Y.001 Z-.452  
N90 G01 X63.852 Y.001 Z-.288  
N91 G01 X64.154 Y.001 Z-.165  
N92 G01 X64.455 Y.001 Z-.078  
N93 G01 X64.756 Y.001 Z-.024  
N94 G01 X65.057 Y.001 Z-.001  
N95 G01 X65.359 Y.001 Z-.008  
N96 G01 X65.6 Y.001 Z-.036  
N97 G01 X65.841 Y.001 Z-.084  
N98 G01 X66.082 Y.001 Z-.153  
N99 G01 X66.323 Y.001 Z-.245  
N100 G01 X66.564 Y.001 Z-.362  
N101 G01 X66.805 Y.001 Z-.508  
N102 G01 X67.046 Y.001 Z-.687  
N103 G01 X67.287 Y.001 Z-.91  
N104 G01 X67.467 Y.001 Z-1.114  
N105 G01 X67.648 Y.001 Z-1.362  
N106 G01 X67.738 Y.001 Z-1.51  
N107 G01 X67.829 Y.001 Z-1.681  
N108 G01 X67.919 Y.001 Z-1.884  
N109 G01 X68.01 Y.001 Z-2.143  
N110 G01 X68.055 Y.001 Z-2.313  
N111 G01 X68.1 Y.001 Z-2.546  
N112 G01 X68.123 Y.001 Z-2.732  
N113 G01 X68.145 Y.001 Z-3.252  
N114 G01 X68.484 Y.001 Z-12.957  
N115 G01 X68.507 Y.001 Z-13.492  
N116 G01 X68.552 Y.001 Z-13.752  
N117 G01 X68.597 Y.001 Z-13.899  
N118 G01 X68.642 Y.001 Z-14.011  
N119 G01 X68.733 Y.001 Z-14.186  
N120 G01 X68.823 Y.001 Z-14.322  
N121 G01 X68.913 Y.001 Z-14.435  
N122 G01 X69.004 Y.001 Z-14.53  
N123 G01 X69.094 Y.001 Z-14.611  
N124 G01 X69.275 Y.001 Z-14.743  
N125 G01 X69.456 Y.001 Z-14.842  
N126 G01 X69.636 Y.001 Z-14.914

---

N127 G01 X69.817 Y.001 Z-14.963  
N128 G01 X69.998 Y.001 Z-14.991  
N129 G01 X70.179 Y.001 Z-15.  
N130 G01 X75 Y.001  
R99=\$AC\_TIMER[1]  
N131 G00 Z-12.  
N132 G00 Z3.  
N133 G00 G53 D0 Z-110. M5  
N134 M30

## Anexo 2 – Programa NC com interpolação polinomial

```

N1 G17 G71 G90 G64 G94
N2 T02 M06
N3 G54 D1
N4 G00 X0 Y.001 S6000 M03
N5 Z3
N6 Z-11.636
N7 G01 Z-15.878 M08 F5000.
$AC_MARKER[1]=0
$AC_PARAM[1]=0
$AC_PARAM[2]=1
$AC_TIMER[1]=0
ID=1 WHENEVER ($AA_IW[X1]>$AC_PARAM[1]) AND ($AA_IW[X1]<$AC_PARAM[2])
DO
$R[$AC_MARKER[1]]=SQRT($VA_VACTM[X1]*$VA_VACTM[X1]+$VA_VACTM[Z1]*$VA
_VACTM[Z1]) $AC_MARKER[1]=$AC_MARKER[1]+1
$AC_PARAM[1]=$AC_PARAM[1]+1 $AC_PARAM[2]=$AC_PARAM[2]+1
N8 X6.314 Z-7.444
N9 POLY PO[X]=(7.812 ,-.119 ,.105) PO[Y]=(.001 ,0.0 ,0.0) PO[Z]=(-5.998 ,.087 ,-.092)
N10 POLY PO[X]=(12.632 ,1.246 ,-.431) PO[Y]=(.001 ,0.0 ,0.0) PO[Z]=(-4.249 , -1.204 , -
.444)
N11 POLY PO[X]=(15.66 ,-.018 ,-.111) PO[Y]=(.001 ,0.0 ,0.0) PO[Z]=(-5.221 ,-.933 ,.168)
N12 POLY PO[X]=(16.556 ,-.038 ,.011) PO[Y]=(.001 ,0.0 ,0.0) PO[Z]=(-5.776 ,-.047 ,.012)
N13 POLY PO[X]=(19.568 ,-.068 ,.037) PO[Y]=(.001 ,0.0 ,0.0) PO[Z]=(-7.842 ,-.132 ,.067)
N14 POLY PO[X]=(21.981 ,.027 ,-.002) PO[Y]=(.001 ,0.0 ,0.0) PO[Z]=(-9.437 ,.044 ,-.006)
N15 POLY PO[X]=(31.015 ,.263 ,.197) PO[Y]=(.001 ,0.0 ,0.0) PO[Z]=(-14.084 ,.313 ,.546)
N16 POLY PO[X]=(37.045 ,.311 ,-.123) PO[Y]=(.001 ,0.0 ,0.0) PO[Z]=(-15.102 ,.709 ,.228)
N17 POLY PO[X]=(46.686 ,-.166 ,-.353) PO[Y]=(.001 ,0.0 ,0.0) PO[Z]=(-11.969 ,3.868 , -
.978)
N18 POLY PO[X]=(48.489 ,-.053 ,.035) PO[Y]=(.001 ,0.0 ,0.0) PO[Z]=(-10.853 ,.04 ,.028)
N19 POLY PO[X]=(63.25 ,-.001 ,.002) PO[Y]=(.001 ,0.0 ,0.0) PO[Z]=(-.656 ,.003 ,0.0)
N20 POLY PO[X]=(63.7 ,0.0 ,.008) PO[Y]=(.001 ,0.0 ,0.0) PO[Z]=(-.366 ,0.0 ,-.016)
N21 POLY PO[X]=(66.693 ,.852 ,-.578) PO[Y]=(.001 ,0.0 ,0.0) PO[Z]=(-.428 , -1.589 ,.023)
N22 POLY PO[X]=(68.071 ,-.567 ,-.211) PO[Y]=(.001 ,0.0 ,0.0) PO[Z]=(-2.431 ,-.977 ,.26)
N23 POLY PO[X]=(68.118 ,-.061 ,.02) PO[Y]=(.001 ,0.0 ,0.0) PO[Z]=(-2.994 ,-.01 ,.003)
N24 POLY PO[X]=(68.502 ,-.229 ,.139) PO[Y]=(.001 ,0.0 ,0.0) PO[Z]=(-13.228 ,.002 , -
.011)
N25 POLY PO[X]=(68.571 ,.001 ,.043) PO[Y]=(.001 ,0.0 ,0.0) PO[Z]=(-13.826 ,0.0 ,.012)
N26 POLY PO[X]=(70.179 ,1.597 ,-.537) PO[Y]=(.001 ,0.0 ,0.0) PO[Z]=(-15. ,.435 ,.401)
N27 X75 Z-15
N28 R99=$AC_TIMER[1]
N29 G00 Z-12.
N30 Z3.
N31 M30

```