



Daniel Kubiak

**Proposta de Sistema Embarcado para
Supervisão e Controle de Movimentos de um
Sistema de Posicionamento com 3 GL**

91/2014

**CAMPINAS
2014**



**UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE ENGENHARIA MECÂNICA**

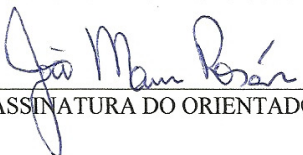
Daniel Kubiak

Proposta de Sistema Embarcado para Supervisão e Controle de Movimentos de um Sistema de Posicionamento com 3 GL

Dissertação de Mestrado apresentada à Faculdade de Engenharia Mecânica da Universidade Estadual de Campinas para obtenção do título de Mestre em Engenharia Mecânica, na Área de Mecânica dos Sólidos e Projeto Mecânico .

Orientador: Prof. Dr. João Maurício Rosário

ESTE EXEMPLAR CORRESPONDE À VERSÃO
FINAL DA DISSERTAÇÃO DEFENDIDA PELO
ALUNO DANIEL KUBIAK E ORIENTADO PELO
PROF. DR. JOÃO MAURÍCIO ROSÁRIO


ASSINATURA DO ORIENTADOR

**CAMPINAS
2014**

Ficha catalográfica
Universidade Estadual de Campinas
Biblioteca da Área de Engenharia e Arquitetura
Luciana Pietrosanto Milla - CRB 8/8129

K951p Kubiak, Daniel, 1962-
Proposta de sistema embarcado para supervisão e controle de movimentos de um sistema de posicionamento com 3 GL / Daniel Kubiak. – Campinas, SP : [s.n.], 2014.

Orientador: João Maurício Rosário.
Dissertação (mestrado) – Universidade Estadual de Campinas, Faculdade de Engenharia Mecânica.

1. Integração. 2. Automação industrial. 3. Prototipagem rápida. 4. Controle numérico. 5. Sistemas de controle. I. Rosário, João Maurício, 1959-. II. Universidade Estadual de Campinas. Faculdade de Engenharia Mecânica. III. Título.

Informações para Biblioteca Digital

Título em outro idioma: Proposal of onboard systems for supervision and control of system movements and positioning with 3 DOF

Palavras-chave em inglês:

Integration
Industrial automation
Rapid prototyping
Numerical control
Control systems

Área de concentração: Mecânica dos Sólidos e Projeto Mecânico

Titulação: Mestre em Engenharia Mecânica

Banca examinadora:

João Maurício Rosário [Orientador]
Paulo Roberto Gardel Kurka
Francisco Carlos Parquet Bizzaria

Data de defesa: 26-09-2014

Programa de Pós-Graduação: Engenharia Mecânica

**UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE ENGENHARIA MECÂNICA
COMISSÃO DE PÓS-GRADUAÇÃO EM ENGENHARIA MECÂNICA
DEPARTAMENTO DE PROJETO MECÂNICO**

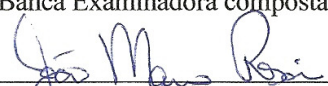
DISSERTAÇÃO DE MESTRADO ACADEMICO

**Proposta de Sistema Embarcado para
Supervisão e Controle de Movimentos de um
Sistema de Posicionamento com 3 GL**

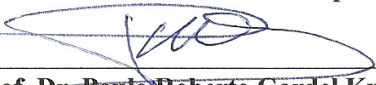
Autor: Daniel Kubiak

Orientador: Prof. Dr. João Maurício Rosário

A Banca Examinadora composta pelos membros abaixo aprovou esta Dissertação:



Prof. Dr. João Maurício Rosário, Presidente.
Universidade Estadual de Campinas – FEM/DPM



Prof. Dr. Paulo Roberto Gardel Kurka
Universidade Estadual de Campinas – FEM/DPM



Prof. Dr. Francisco Carlos Parquet Bizarria
Instituto de Aeronáutica e Espaço - DCTA

Campinas, 26 de setembro de 2014.

Dedicatória:

Dedico esse trabalho aos meus pais João (in memorium) e Wilma e a minha esposa Rosane que me apoiaram intensivamente em toda a minha vida, mesmo nos periodos que estive ausente da vida deles.

Agradecimentos

Esse trabalho não poderia ser terminado sem a ajuda de diversas pessoas às quais presto minha homenagem:

A meus pais e minha esposa, pelo incentivo em todos os momentos da minha vida.

Ao meu orientador, prof. Dr. João Maurício Rosário, pela forma como conduziu esse trabalho e principalmente pela confiança que depositou em mim.

Ao meu grande amigo Edgard de Oliveira e a todos os demais amigos do LAIR-UNICAMP que ajudaram de forma direta e indireta na conclusão desse trabalho.

Resumo

O objetivo dessa dissertação de mestrado é a proposta de sistema embarcado para supervisão e controle de movimentos de um sistema de posicionamento com 3 GL direcionado a aplicações em dispositivos máquinas ferramenta industriais, que apresentam arquitetura de controle e supervisão aberta, que poderão ser integradas a diferentes aplicações da Automação Industrial (sensores, atuadores, sistema de visão e outros), a partir da utilização de ferramentas para prototipagem de sistemas de automação industrial embarcados. O desenvolvimento desse trabalho de pesquisa permitiu a integração de conhecimentos nas diferentes áreas da engenharia, tendo como consequência a implementação de um sistema de controle de posicionamento embarcado, onde essa aplicação poderá ser estendida em outras aplicações de Automação Industrial.

No transcorrer desse trabalho foi implementada uma arquitetura de acionamento e controle, e integração com dispositivos embarcados de controle de juntas e sistema de visão e software para geração automática de movimentos, com validação a partir de situações reais.

Finalmente, esse trabalho permitiu o desenvolvimento de um ambiente aberto voltado à formação e capacitação profissional de projetos na área de automação industrial, onde os principais conceitos e problemas industriais poderão ser verificados e validados, fornecendo ainda subsídios para a análise e implementação de estratégias para concepção dessas aplicações.

Palavras Chave: Sistema Embarcado, Sistema de Posicionamento, Integração, Automação industrial

Abstract

The objective of this master's degree dissertation is the system proposal embarked for supervision and control of movements of a positioning system with 3 DOF addressed to applications in devices machines tool industrial, that present control architecture and open supervision that can be integrated into different applications of the Industrial Automation (sensor, actuators, vision system and other), starting from the use of tools for prototyping of systems of industrial automation embarked. The development of this research work allowed the integration of knowledge in the different areas of the engineering, tends as consequence the implementation of a system of control of embarked positioning, where this application can be extended in other applications of Industrial Automation.

In the Course of this work was implemented a drive and control architecture, and integration with embedded devices of control joints and vision system and software for automatic generation of movements, with validation from actual situations.

Finally, this work allowed the development of an open architecture returned to the formation and professional training of projects in the area of industrial automation, where the principal concepts and industrial problems can be verified and validated, still supplying subsidies for the analysis and implementation of strategies for conception of these applications.

Key words: Embarked System, Positioning System, Integration, Industrial Automation

Lista de Ilustrações

Figura 2.1: Supervisão e arquitetura de controle de um dispositivo máquina-ferramenta CNC.....	06
Figura 2.2: Sistema de controle modular e aberto.....	12
Figura 2.3: Comunicação entre os níveis.....	13
Figura 2.4: Utilização de redes de Petri no modelamento e análise de sistemas.....	18
Figura 3.1: Diagrama de blocos do controle de movimentos.....	20
Figura 3.2: Dispositivo Robótico Cartesiano.....	21
Figura 3.3: Fluxograma do software de funcionamento.....	22
Figura 3.4: Tipos de sistemas de transmissão mecânica.....	26
Figura 3.5: - Ligação entre o motor e a porta.....	28
Figura 3.6: Contador CMOS.....	28
Figura 3.7: Controle de direção utilizando-se flip-flops.....	29
Figura 3.8: Placa Motion PCI 7344 da National Instruments.....	31
Figura 3.9: Esquema do sistema controlado.....	32
Figura 3.10: Tela diagrama de blocos implementados em LabVIEW.....	33
Figura 3.11: Proposta de pagina HTML - Painel de Controle Implementado em LabView.....	34
Figura 3.12: Diagrama de blocos implementados em LabVIEW.....	34
Figura 3.13: Control Motion da NI.....	36
Figura 3.14: Sistema de Controle e etapas de implementação do Control Motion da NI.....	37
Figura 3.15: Control Motion da NI – assistente do software de prototipagem.....	38
Figura 3.16: Diferentes elementos de uma malha de controle.....	38
Figura 3.17: Gerador de movimentos para uma trajetória simples.....	39
Figura 3.18: Telas típicas do Control Motion da NI Instruments.....	42
Figura 3.19: Diferentes fases da implementação proposta.....	43
Figura 3.20: Sistema Implementado.....	44

Lista de Ilustrações

Figura 3.21: Tela típica do ambiente de desenvolvimento IDE MPLAB.....	46
Figura 3.22: Família de normas IEEE 1451.....	48
Figura 3.23: Principais aplicações em Visão de Máquinas em Automação.....	52
Figura 4.1: Estrutura do modelo do eixo de movimentação (YEUNG, 2006).....	58
Figura 4.2: Motor DC e acoplamentos com encoder.....	59
Figura 4.3: Interface de potência.....	59
Figura 4.4: Estrutura de polinômios RST.....	63
Figura 4.5: Estrutura em cascata com a síntese dos controladores RST.....	63
Figura 4.6: Gerador de trajetórias (ERKORMAZ; ALTINTAS, 2001).....	65
Figura 4.7: Perfis de Movimento máquinas CNC (ERKORKMAZ; ALTINTAS, 2006).....	66
Figura 5.1: Bancada utilizada para implementação do experimento.....	70
Figura 5.2: Bancada, parte de potência, gerador de PWM e borneira de conexão.....	70
Figura 5.3: Conjunto com Juntas X,Y e Z.....	71
Figura 5.4: Controlador Placa Motion PCI 7344 utilizada utilizada no experimento.....	72
Figura 5.5: Placa Interface de potência.....	72
Figura 5.6: Conjunto Interface de potência, PWM e Borneira.....	73
Figura 5.7: Borneira da Placa Motion.....	73
Figura 5.8: Tela do sistema motion para configuração de cada eixo independentemente.....	74
Figura 5.9: Tela do sistema supervisor utilizado juntamente com o controlador motion.....	74
Figura 5.10: Resultado de um desenho traçado pelo sistema utilizando controlador motion.....	75
Figura 5.11: Elementos Básicos do sistema de Controle (Blocos).....	77
Figura 5.12: Baixa velocidade (Eixo X).....	78
Figura 5.13: Alta Velocidade (Eixo Y).....	79
Figura 5.14: Baixa velocidade (Eixo Z).....	79
Figura 5.15: Trajetória: VB-Raio menor.....	82
Figura 5.16: Trajetória: VM-Raio menor.....	83
Figura 5.17: Trajetória: VA-Raio menor.....	83

Lista de Ilustrações

Figura 5.18: Trajetória VB-Raio maior.....	84
Figura 5.19: Trajetória VM-Raio maior.....	84
Figura 5.20: Trajetória VA-Raio maior.....	85
Figura 5.21: Implementação bancada com kit DE2-115 da Altera.....	87
Figura 5.22: kit de desenvolvimento DE2-115 da Altera.....	87
Figura 5.23: kit de desenvolvimento DE2-115 com cabo de comunicação.....	88
Figura 5.24: Detalhe do cabo de comunicação utilizado.....	88
Figura 5.25: Bloco de controle utilizando blocos DSP Builder.....	89
Figura 5.26: Comparação entre entrada e saída do sistema para trajetória triangular utilizando controlador por alocação de pólos implementado em DSP Builder.....	90
Figura 5.27: Comparação entre entrada e saída do sistema para trajetória helicoidal utilizando controlador por alocação de pólos implementado em DSP Builder.....	91
Figura 5.28: Comparação entre entrada e saída do sistema para trajetória triangular utilizando controlador preditivo em DSP Builder.....	91
Figura 5.29: Comparação entre entrada e saída do sistema para trajetória helicoidal utilizando controlador preditivo em DSP Builder.....	92
Figura 5.30: Bloco de controle utilizando código VHDL.....	92
Figura 5.31: Comparação entre entrada e saída do sistema para trajetória triangular utilizando controlador por alocação de pólos implementado em VHDL.....	93
Figura 5.32: Comparação entre entrada e saída do sistema para trajetória helicoidal utilizando controlador por alocação de pólos implementado em VHDL.....	94
Figura 5.33: Comparação entre entrada e saída do sistema para trajetória triangular utilizando controlador preditivo implementado em VHDL.....	94
Figura 5.34: Comparação entre entrada e saída do sistema para trajetória helicoidal utilizando controlador preditivo implementado em VHDL.....	95
Figura 5.35: Esquemático em Quartus II para teste de controlador desenvolvido.....	96
Figura 5.36: Placa de potência utilizada para acionamento do motor CC.....	97

Lista de Ilustrações

Figura 5.37: Metodologia para Identificação da Máquina-Ferramenta CNC.....	100
Figura 5.38: Trajetória linear para testes dos Controladores.....	101
Figura 5.39: Trajetória Circular para testes dos Controladores.....	102

Lista de Tabelas

Tabela 2.1: Requisitos do controlador de arquitetura aberta.....	14
Tabela 3.1: Critérios utilizados para Seleção de Motores NI.....	41
Tabela 3.2: Comparativo de Velocidade do Software de Visão NI em relação ao Líder do Mercado.....	53
Tabela 5.1: Erro de Seguimento em trajetória Linear utilizando controle PID.....	78
Tabela 5.2: Erro de Contorno em trajetória Circular com PID. Eixos X-Y.....	82
Tabela 5.3: Parâmetros de configuração para trajetória linear.....	101
Tabela 5.4: Parâmetros de configuração para a trajetória Circular.....	102

Lista de Abreviaturas e Siglas

CNC - Comando Numérico Computadorizado

CN - Comando Numérico

CCD - Charge-Coupled Device

CLP - Controladores Lógicos Programáveis

DSP – Digital Signal Processor

FDB - Function Block Diagram

FPGA - Field Programmable Gate Array

GL - Graus de Liberdade

GRAFCET - Grafo de Comando Etapa e Transição

HIL - Hardware In-the-Loop

HTML - HyperText Markup Language

IEC - International Electrotechnical Commission

IL - Instruction List

IP - Internet Protocol

LabVIEW - Laboratory Virtual Instrument Engineering Workbench

LAN - Local Area Network

LD - Ladder Diagram

NI – National Instruments

PAC - Programmable Automation Controller

PC - Personal Computer

PCI - Peripheral Component Interconnect

PXI - PCI eXtensions for Instrumentation

RdP – Redes de Petri

SCADA - Supervisory Control And Data Acquisition

SED - Sistemas a Evento Discreto

SFC - Sequential Function Chart

Lista de Abreviaturas e Siglas

USB - Universal Serial Bus

WWW - World Wide Web

Sumário

1	Introdução	1
1.1	Posicionamento do Problema	2
1.2	Objetivos	2
1.3	Objetivos Específicos	2
1.4	Delineamento do Trabalho	3
2	Revisão bibliográfica	5
2.1	Sistema Máquina-Ferramenta CNC	5
2.2	Arquitetura de Acionamento e Controle	7
2.3	Integração de Sistemas Mecatrônicos	8
2.4	<i>Retrofitting</i> de Sistemas Mecatrônicos	9
2.5	Sistemas de Controle de Arquitetura Aberta	10
2.6	Requisitos para o sistema de controle de arquitetura aberta	12
2.7	Ferramentas para integração de Sistemas Mecatrônicos	15
2.8	Simuladores Robóticos	15
2.9	Simuladores da modelagem cinemática	15
2.10	Grafo de Comando Etapa e Transição – GRAFCET	15
2.11	Redes de Petri - Conceitos Básicos e Definições	17
2.12	Conclusão e Próximas Etapas	18
3	Dispositivo Máquina Ferramenta CNC	19
3.1	Descrição do Dispositivo CNC em estudo	19
3.2	Sistemas de Acionamento e Controle	23
3.3	Sistema de Transmissão de Movimentos	26
3.4	Acionamento através de motor de passo	27
3.5	Acionamento através de motor cc	29

3.6	Software aberto para supervisão e controle de movimentos.....	30
3.7	Descrição do software Labview Motion.....	35
3.8	Descrição do hardware associado.....	44
3.9	Descrição de software MPLAB de acionamento e controle de movimentos.....	45
3.10	Automação e Integração com outros dispositivos.....	47
3.11	Sensores e dispositivos de segurança.....	47
3.12	Interface com Sistema de Visão Robótica.....	49
3.13	Conclusão e Próximas Etapas.....	56
4	Modelagem do sistema de acionamento e controle	57
4.1	Sistema de acionamento e transmissão.....	57
4.2	Projeto do Sistema de controladores PID.....	60
4.3	Gerador de Trajetórias.....	64
4.4	Conclusão e Proximas Etapas.....	67
5	Implementação experimental.....	69
5.1	Descrição da Plataforma Experimental implementada.....	70
5.2.	Interface de acionamento e controle utilizando plataforma National-Motion	71
5.3	Validação experimental.....	76
5.4	Validação das estratégias Propostas em Máquina-Ferramenta CNC.....	98
5.5	Conclusões.....	103
6	Resultados e discussões.....	106
7	Conclusões e sugestões para próximos trabalhos.....	107
7.1	Conclusões.....	107
7.2	Perspectivas Futuras.....	108

Referências109

Anexo A - Software em linguagem C.....116

1 Introdução

1.1 Posicionamento do Problema

Dispositivos Máquinas-ferramenta são equipamentos de alto custo, empregados tipicamente para a fabricação e usinagem de peças, geralmente metálicas, que requerem um alto nível de precisão. A fabricação das peças pode ser realizada pela remoção de material, usinagem ou por um processo de forjamento ou estampagem para que a peça adquira o formato desejado (SOUSA, 1998). Desde o seu surgimento, as máquinas-ferramenta têm passado por um constante processo de aprimoramento e, atualmente, constituem a base da indústria mecânica sendo usadas tanto direta quanto indiretamente na manufatura de diversos produtos disponíveis (BOOTHROY; KNIGHT, 2006).

Entretanto, esses dispositivos apresentam uma arquitetura de supervisão e controle pouco flexível e muito fechada e, com o passar dos anos, mais arquiteturas vão se tornando obsoletas em relação a parte mecânica do equipamento de grande precisão, justificando a necessidade de reajustes (*retrofitting*) do seu sistema de acionamento e controle, que será designado prototipagem rápida de arquiteturas de supervisão e controle.

Conseqüentemente, esse trabalho de pesquisa está direcionado no desenvolvimento de metodologias para a aplicação em dispositivos máquinas-ferramenta industriais, que apresentam arquitetura de controle e supervisão aberta, com objetivo de desenvolver dispositivos máquinas-ferramenta CNC com alto desempenho.

A partir da utilização dessa arquitetura, esses dispositivos, poderão ser integrados a diferentes ramos de aplicações da automação Industrial, e para isso, pode-se utilizar: sensores, atuadores, sistema de visão e outros, a partir da utilização de ferramentas para prototipagem de

sistemas de automação industrial embarcados, mantendo-se a garantia de robustez e estabilidade no sistema de controle.

1.2 Objetivos

Esse trabalho tem como objetivo, o estudo e implementação de um Sistema Embarcado para Supervisão e Controle de Movimentos de um Sistema de Posicionamento com três graus de liberdade.

1.3 Objetivos Específicos

Para esse estudo será tomado como objetivo específico o *retrofitting* de uma máquina com três graus de liberdade, como segue:

- ✓ Substituição os motores de passo existentes, em malha aberta, por motores de corrente contínua, que utilizam malha de controle fechada, através de geradores de pulso.
- ✓ Elaboração e construção e a implementação dos controladores e da eletrônica de potência.
- ✓ Implementação de controladores PID, sob a forma polinomial RST para implementação embarcada utilizando hardware embarcado baseado em FPGA.
- ✓ Desenvolvimento de um planejador de trajetórias.
- ✓ Implementação dos controladores utilizando plataforma National-Motion baseada em LabView.
- ✓ Implementação de interface de acionamento e controle em FPGA (*Field-Programmable Gate Array*).
- ✓ Validação das estratégias propostas em dispositivos máquina-Ferramenta CNC.

1.4 Delineamento do Trabalho

Para atingir os objetivos e metas delineados anteriormente, essa dissertação de mestrado será desenvolvida através da metodologia proposta a partir do desenvolvimento dos seguintes capítulos.

No Capítulo 2 é apresentada uma revisão da literatura contendo informações relevantes e discussão sobre outros trabalhos desenvolvidos na área de sistema máquina-ferramenta CNC e arquiteturas abertas de controle, onde é descrita a arquitetura de acionamento e controle, a Integração de Sistemas Mecatrônicos, o *Retrofitting* de sistemas mecatrônicos, os sistemas de controle de arquitetura aberta, os requisitos para o sistema de controle de arquitetura aberta.

São esplanadas as ferramentas para integração de Sistemas Mecatrônicos os simuladores Robóticos, os simuladores da modelagem cinemática, como também são esplanados o grafo de Comando Etapa e Transição GRAFCET as redes de Petri os conceitos Básicos e Definições.

No Capítulo 3, é apresentado o dispositivo máquina ferramenta CNC objeto desse estudo, e disponibilizado no Laboratório de Automação Integrada e Robótica da Faculdade de Engenharia Mecânica da UNICAMP. Serão apresentadas as principais características desse dispositivo dispositivo CNC, a proposta de arquitetura aberta de acionamento e controle, os estudos referentes ao sistema de transmissão de movimentos, e seu sistema acionamento implementado através de motor de passo, através de motores de corrente contínua (cc).

Será abordado o software aberto para supervisão e controle de movimentos, e a descrição do software Labview Motion da National Instruments, a descrição do hardware associado, a descrição de software MPLAB de acionamento e controle de movimentos. Finalmente será discutida a automação e integração com outros dispositivos, e sensores e dispositivos de segurança, e a interface com sistema de visão robótica.

No Capítulo 4 são apresentados aspectos relacionados a modelagem do sistema de acionamento e controle, descrevendo o sistema de acionamento e transmissão, onde também será discutido o projeto do Sistema de controladores PID, o gerador de trajetórias.

No Capítulo 5 é apresentada a descrição da plataforma experimental implementada, e será discutida a interface de acionamento e controle para Máquina-Ferramenta CNC utilizando inicialmente plataforma National-Motion que é um sistema de controle embarcado da National Instruments, baseado no ajuste de um regulador PID, com validação experimental baseada na implementação de trajetórias linear e circular e ajuste de parâmetros de sintonia do controlador PID. Nesse mesmo capítulo será realizado um estudo referente a implementação de interface de acionamento e controle embarcados em FPGA (sistema com arquitetura aberta), baseado na utilização de blocos DSP Builder, implementados utilizando código VHDL aplicado a plataforma Quartus, sendo apresentado resultados experimentais comparativos que mostram a validação da plataforma aberta implementada. No final desse capítulo os principais resultados obtidos são sintetizados e discutidos.

Finalmente, no Capítulo 6 dessa dissertação de mestrado são apresentadas as conclusões finais desse trabalho, apresentado ainda algumas perspectivas futuras de continuidade do tema apresentado, e a proposta de sugestões para os próximos trabalhos.

2 Revisão bibliográfica

2.1 Sistema Máquina-Ferramenta CNC

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.

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 desse processo.

Os dispositivos máquinas-ferramenta CNC (Controle Numérico Computadorizado) são sistemas mecatrônicos que trabalham na produção de peças complexas com precisão, otimização dos recursos e altas velocidades, especialmente quando é associado a programas de CAD (*Computer Aided Design*)/CAM (*Computer Aided Manufacturing*).

Máquinas-ferramenta CNC são dispositivos complexos constituídos por três unidades principais: sistema de controle, unidade de potência e unidade mecânica (ALTINTAS, 2000).

Uma representação genérica de máquina-ferramenta mostrando cada uma de suas unidades pode ser observada na Figura 2.1, a unidade de potência contém os motores e os amplificadores de potência; a unidade mecânica corresponde a estrutura do dispositivo e a unidade CNC é responsável pela geração de movimentos e pelo controle dos eixos do dispositivo.

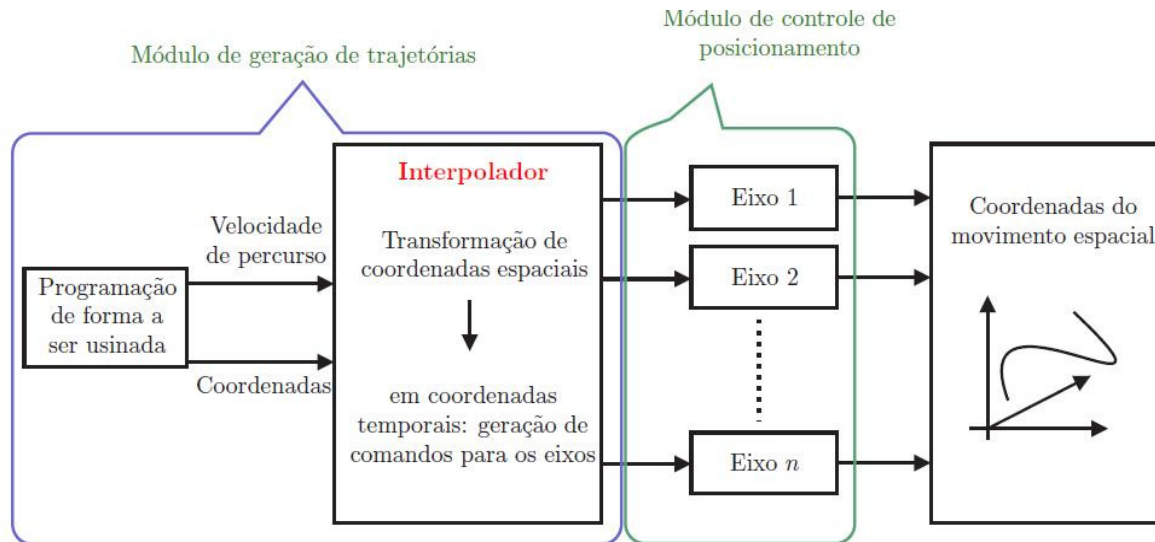


Figura 2.1: Supervisão e arquitetura de controle de um dispositivo máquina-ferramenta CNC.

Há casos em que a unidade de controle corresponde a um computador ligado ao dispositivo, no entanto na maioria das vezes, uma placa de lógica reprogramável possuindo um ou mais microprocessadores é conectada ao dispositivo. A vantagem dessa configuração é deixar um hardware dedicado para a realização da tarefa aumentando a velocidade do processo, liberando recursos computacionais para tarefas que exijam maior poder de processamento.

A geração de movimentos de tais dispositivos pode ser feita ou utilizando programação NC (*Numerical Control*), conhecida como *código G*, ou por meio de sistemas CAD (*Computer Aided Design*). No primeiro caso, o código contém a informação necessária para a correta usinagem de uma peça, como posicionamento, velocidade de usinagem e orientação da ferramenta. No segundo caso apenas alguns pontos da peça são fornecidos, desse modo um planejador de trajetórias deverá ser empregado para geração dos perfis de posição, velocidade e aceleração que serão enviados para a máquina de ferramenta.

Para garantir o bom desempenho do equipamento, as trajetórias de referência não devem apresentar modificações abruptas ao longo de sua execução, além de haver necessidade de se contar com um sistema de controle bem ajustado às condições de funcionamento da máquina.

Com a finalidade de compreensão mais apurada do funcionamento dos sistemas e de sua aplicabilidade a dispositivos CNC, buscou-se na literatura especializada referências que permitiram fundamentar análises e discussões que compõem as seções subsequentes.

2.2 Arquitetura de Acionamento e Controle

Atualmente dois níveis de abertura de arquitetura são considerados pelos fabricantes de máquina-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*)

Nível de abertura do *kernel* (núcleo de processamento do sistema operacional) do comando numérico (NCK – *Numerical Control Kernel*).

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.

Segundo PARK, a utilização do conceito *middleware* (programa que faz a comunicação entre outros *softwares*) no módulo do *kernel software*, este 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 novos métodos de configuração de *hardware* e *software* para viabilizar essa abertura.

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

2.3 Integração de Sistemas Mecatrônicos

As opções de integração e flexibilidade propostas pela nova geração de máquina-ferramenta são algumas 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.

A tendência em máquinas CNC vem visando desenvolver arquiteturas de controle aberto (*OAC-Open Architecture Control*), cuja principal contribuição é a facilidade e flexibilidade para integração de controladores conforme as especificações dos usuários, mediante interfaces abertas e métodos de configuração com ambientes standardizados e de fácil acesso. Interfaces de controle aberto para dispositivos CNC são necessárias para integrar novas e avançadas funções de máquina dentro dos sistemas de controle e são importantes para a criação de unidades de manufatura reconfigurável (PRITSCHOW, et al., 2001). Um sistema aberto segundo o estandarte IEEE (SUSANU et al., 2005), deverá ter a capacidade de implementar aplicações para funcionar em diferentes plataformas de vários tipos de vendedores e clientes, interoperando com outros aplicativos e outros sistemas, e mantendo um estilo consistente de interação com o usuário.

Segundo (WANG et al., 2004), 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.

2.4 Retrofitting de Sistemas Mecatrônicos

Retrofitting é um termo utilizado principalmente em engenharia para designar o processo de modernização de algum equipamento já considerado ultrapassado ou fora de norma.

As máquinas-ferramenta apresentaram um rápido desenvolvimento, devido principalmente a utilização dos sistemas de comando numérico computadorizado (CNC). Com o passar do tempo as máquinas-ferramenta passaram a apresentar sintomas que alertam para a inadequação ou fim de vida útil das mesmas. Do ponto de vista econômico, a simples troca desses equipamentos pode-se tornar um investimento extremamente dispendioso, no entanto, uma alternativa que vem sendo experimentada com muito sucesso por diversas empresas é a denominada modernização ou *Retrofitting (adaptação)*, que tem por objetivo a atualização tecnológica das máquina-ferramenta (DEMEC, 2011).

Mesmo máquinas mais antigas, podem ser reaproveitadas, pois possuem estruturas de elevada rigidez, proporcionando condições de permanecerem em operação e agregando novas funções e recursos. A modernização trata-se principalmente da inclusão ou substituição dos sistemas de comando e controle, acionamentos e sistemas de medição por equipamentos atuais. Dessa forma, a máquina passaria a apresentar melhores especificações (DEMEC, 2011).

O *retrofitting* de máquinas, também conhecido como reforma ou modernização, é muitas vezes a solução para empresas que desejam dar uma sobrevida para máquinas antigas, mantendo suas características periféricas, com perfeito estado de conservação mecânica. Essa necessidade é crescente, uma vez que a eletrônica das máquinas, equipamentos e sistemas evoluiu muito nos últimos anos e continua com rápida evolução.

Devido ao avanço tecnológico, os produtos estão se tornando defasados cada vez mais rápidos. Dessa maneira, a tendência é direcionar os esforços da indústria para se produzir produtos competitivos de acordo com as demandas do consumidor. Isto, por consequência, significará trabalhar com estoques menores e um sistema de produção que atenda a essas variações de um modo rápido e econômico (RUBIO, 1999).

O objeto desse estudo é realizar o *retrofitting* em uma máquina CNC com três graus de liberdade, que era equipada com três motores de passo, com seu respectivo sistema de controle e acionamento. O retrofitting dessa máquina consistiu na substituição dos motores de passo por motores de corrente contínua e também da substituição de seu sistema de acionamento por um sistema eletrônico com tecnologia PWM o sistema de controle foi substituído pelo sistema motion da National Instruments e implementado com software Labview.

2.5 Sistemas de Controle 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.

A definição do 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”.

Conforme a definição acima, um conjunto de especificações é necessário para que os fabricantes dos controladores utilizem-no.

Entre as iniciativas para tal padronização, destacam-se:

OSEC (Open System Environment for Manufacturing): esse 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.

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.

OMAC (*Open Modular Architecture Controllers*): as indústrias norte americanas criaram esse fórum com o propósito de definição de especificações de API através de classes de interface para a implementação nos controladores. Essa abordagem está baseada em componentes e visa a facilidade de modularização.

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.

No trabalho de (VALENTE et al. 2007) 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.

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.

2.6 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.

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 2.2 ilustra a estrutura do sistema de controle modular e aberto.

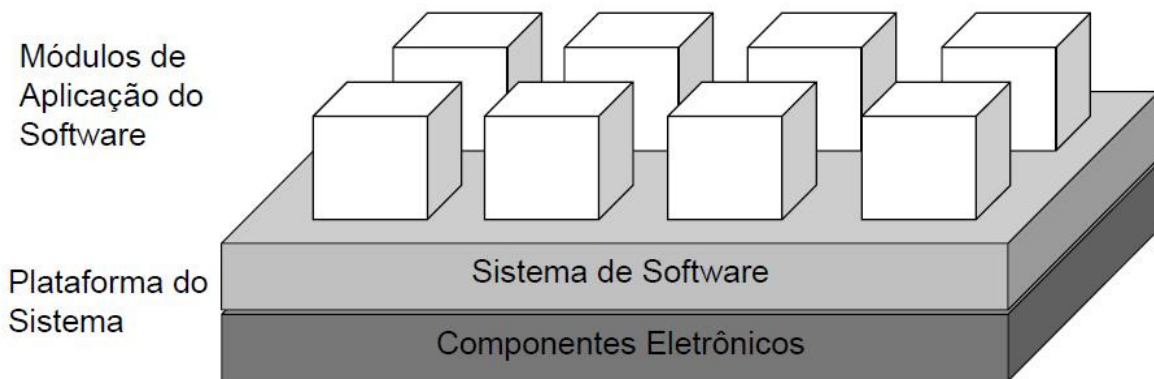


Figura 2.2: Sistema de controle modular e aberto

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.

Na estrutura da Figura 2.3: pode-se observar que as redes de comunicação são as interfaces padronizadas para a troca de dados entre os níveis especificados.

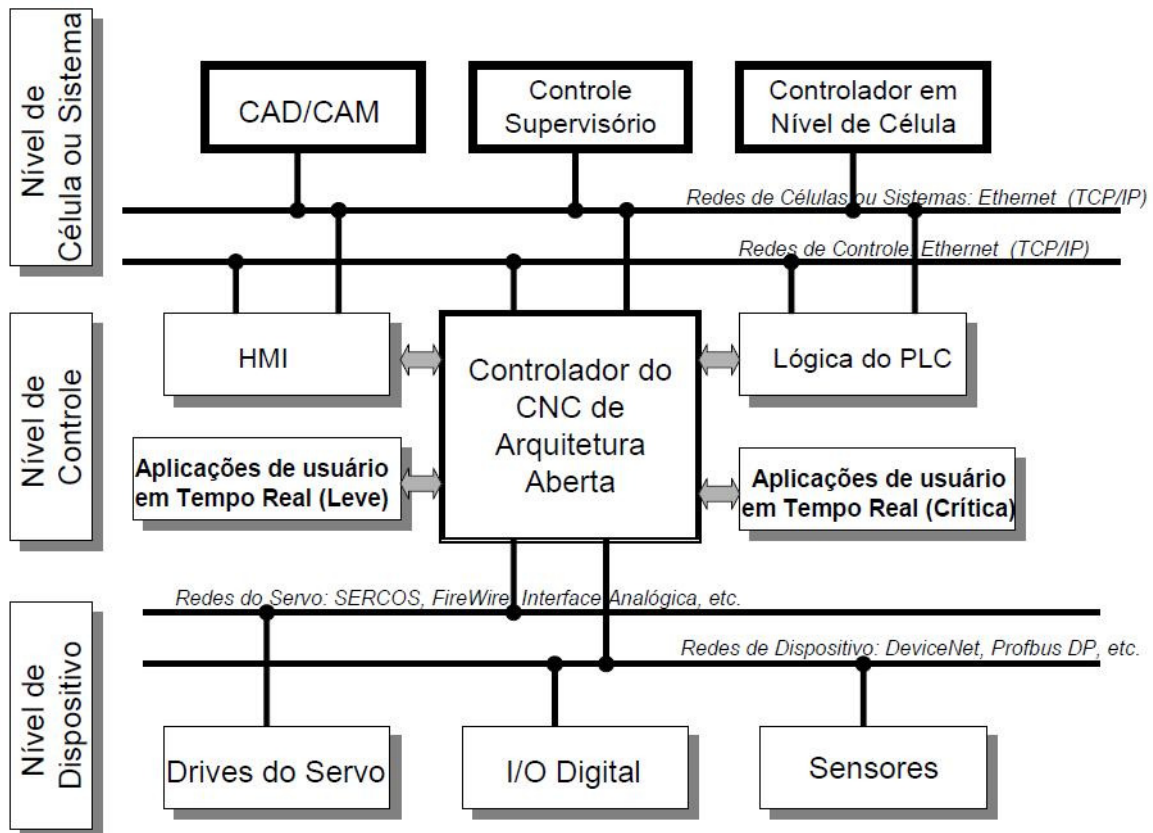


Figura 2.3: Comunicação entre os níveis

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 et al.20070 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

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.

2.7 Ferramentas para integração de Sistemas Mecatrônicos

Ao mesmo tempo que a integração de sistemas mecatrônicos têm sido uma área de grande interesse e desenvolvimento na robótica, e com o avanço da robótica surgem novas ferramentas computacionais de modelagem e simulação.

2.8 Simuladores Robóticos

A realização de etapas de modelagem, simulação e análise é o processo de criar e experimentar com modelos computadorizados matemáticos de sistemas físicos, nesse caso pode-se utilizar simuladores robóticos.

A simulação de manipuladores robóticos pode ser classificada em dois níveis: simulação gráfica e simulação numérica. A simulação gráfica de manipuladores abrange a visualização da movimentação do manipulador no seu ambiente de trabalho, sendo a mesma baseada CAD e na utilização de ferramentas de visualização gráfica. Ao mesmo tempo, a simulação de manipuladores utiliza conceitos de simulação numérica da dinâmica e o controle do manipulador.

2.9 Simuladores da modelagem cinemática

Algumas empresas desenvolveram um banco de provas para demonstrar como as máquinas, softwares e operadores podem estar integrados em rede para trabalhar de modo efetivo e eficiente.

O Banco de provas permite melhorar a produtividade e implementar inovações em todos os níveis da manufatura, onde para atender esses objetivos foi desenvolvida a caracterização, acesso remoto e simulação para máquinas

2.10 Grafo de Comando Etapa e Transição – GRAFCET

Devido à variedade de CLP's existentes no mercado, o desenvolvimento de programas e ferramentas de descrição devem ser adequadas permitindo o desenvolvimento de tarefas, independente da PLC utilizada e que adicionalmente ofereçam a opção de se programar em uma linguagem mais natural para os vários níveis de usuários.

Dessa forma pode-se dividir o problema em várias etapas, tornando-o mais simples e facilitando a visualização das seqüências de operações, alteração de especificação e detecção de falhas conceituais no programa.

O SFC - “Sequential Function Chart” ou Diagrama Funcional Seqüencial, também conhecida pelo nome de GRAFCET, oferece vantagens aos usuários e programadores, principalmente na modelagem de problemas complexos de automação, pois se pode dividir o problema em várias partes, tornando mais simples a programação, facilitando a visualização das sequencias de operações, alteração de especificação e a detecção de falhas conceituais no programa.

O GRAFCET (Grafo de Comando Etapa - Transição) foi criado na França em 1975, através de um grupo de pesquisadores e gerentes industriais, envolvidos com sistemas discretos de grande complexidade, sendo coordenados pela AFCET “*Association Française pour la Cybernétique, Economique, Technique*” e posteriormente padronizado pela ADEPA “*Agence Nationale pour la Developpement de la Production Automatisée*”, sendo considerado uma particularização das Redes de Petri, pois as redes possuem uma possibilidade de aplicações bem superior às que estão restritos os comportamentos cíclicos das máquinas e sistemas automáticos.

Com a combinação de elementos tem-se a representação estática do Sistema Automatizado. Aplicando-se as Regras de Evolução, obtem-se a visão dinâmica do mesmo.

Quando acionado um botão o motor liga e somente após a desativação da botoeira o motor é desligado e assim sucessivamente.

Algumas técnicas utilizadas atualmente para descrever comportamento seqüencial em sistemas automatizados incluem fluxogramas, diagramas de variáveis de estado, Rede de Petri, diagrama trajeto-passo e o GRAFCET.

2.11 Redes de Petri - Conceitos Básicos e Definições

A metodologia de modelagem de Sistemas a Eventos Discretos (SED) utilizando Redes de Petri, (RdP) foi proposta em 1962, por Carl Petri, matemático alemão, que através de uma tese de doutoramento criou esse método de estudo para sistemas dinâmicos a evento discreto, direcionado às Comunicações com Autômatos, originando posteriormente, duas grandes linhas de desenvolvimento nas áreas de Ciências da Computação e em Engenharia de Sistemas padronizando as Redes de Petri (Huber, Jensen e Shapiro, 1990).

Portanto as Redes de Petri são ferramentas gráficas e matemáticas de modelagem para descrição e/ou especificação que podem ser aplicadas a diversos tipos de sistemas apresentando um bom nível de abstração em comparação com outros modelos gráficos.

Além disso, as Redes de Petri possibilitam a verificação do sistema especificado. Usando-se RdP, pode-se modelar sistemas paralelos, concorrentes, assíncronos e não-determinísticos (Melo e Sobreira, 2003).

Dessa forma a RdP é uma linguagem formal que permite a modelagem de sistemas dinâmicos discretos com grande poder de expressividade, permitindo representar com facilidade todas as relações de causalidade entre processos em situações de: sequencialidade, conflito, concorrência e sincronização (Matos e Santos, 2004). Sendo que a sua aplicabilidade em diversas áreas transformou-o em tema alvo de investigação básica e aplicada, e sua utilização na Modelagem de Sistemas Automatizados apresenta algumas vantagens como:

- Captura das relações de precedência e os vínculos estruturais dos sistemas reais.
- Gráficamente expressivas, permitindo a modelagem de conflitos e filas.
- Possui fundamento matemático e prático.
- Admite várias especializações (RP's temporizadas, coloridas, estocásticas, de confiabilidade etc.).

Portanto, podem-se definir as Redes de Petri (RdP's) por meio de conjuntos, funções e também por grafos, de maneira que suas propriedades possam ser obtidas pela teoria dos conjuntos e/ou pela teoria dos grafos.

As redes de Petri permitem modelar sistemas constituídos por componentes que apresentem características de funcionamento concorrente e interagentes. Segundo Peterson (1981) a sua utilização poderá ser realizada de modos diversos, dependendo do objetivo em vista.

Uma abordagem possível de utilização das RdP considera-as como uma ferramenta auxiliar de análise, sendo que nessa abordagem, outras técnicas e formalismos são utilizados para especificar o sistema. Com base nessa especificação, o sistema é então modelado através de uma RdP que será posteriormente analisada. Se forem detectados problemas, procedem-se as alterações na especificação, e o ciclo será repetido até que mais nenhum problema seja detectado.

A figura 2.4 mostra esse processo.

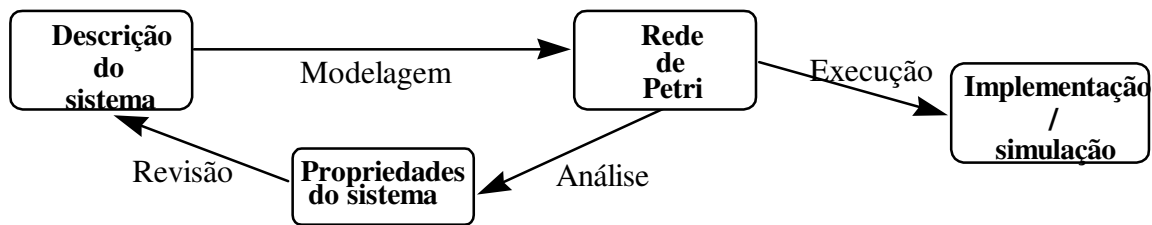


Figura 2.4: Utilização de redes de Petri no modelamento e análise de sistemas

2.12 Conclusão e Próximas Etapas

Esse capítulo apresenta a multidisciplinaridade que envolve a área de automação industrial através da apresentação dos diversos assuntos que se encontram inseridos no conhecimento de sistemas automatizados de produção. Dessa forma, ao permitir o conhecimento dos assuntos tratados nesse capítulo, permite-se também uma visão da decomposição do Sistema Automatizado, através dos elementos de cada parte e seu sistema de controle, de maneira a abordar através de uma visão geral do sistema para os subsistemas que o compõe esse todo.

3 Dispositivos Máquina-Ferramenta CNC

3.1 Descrição do Dispositivo CNC em estudo

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 motores. Os sinais dos sensores são utilizados pelos algoritmos de controle do CNC para manter o processo nas condições programadas.

A Figura 3.1 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. Essa taxa de atualização é tipicamente de 1 a 5 milisegundos. 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. Essa 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.

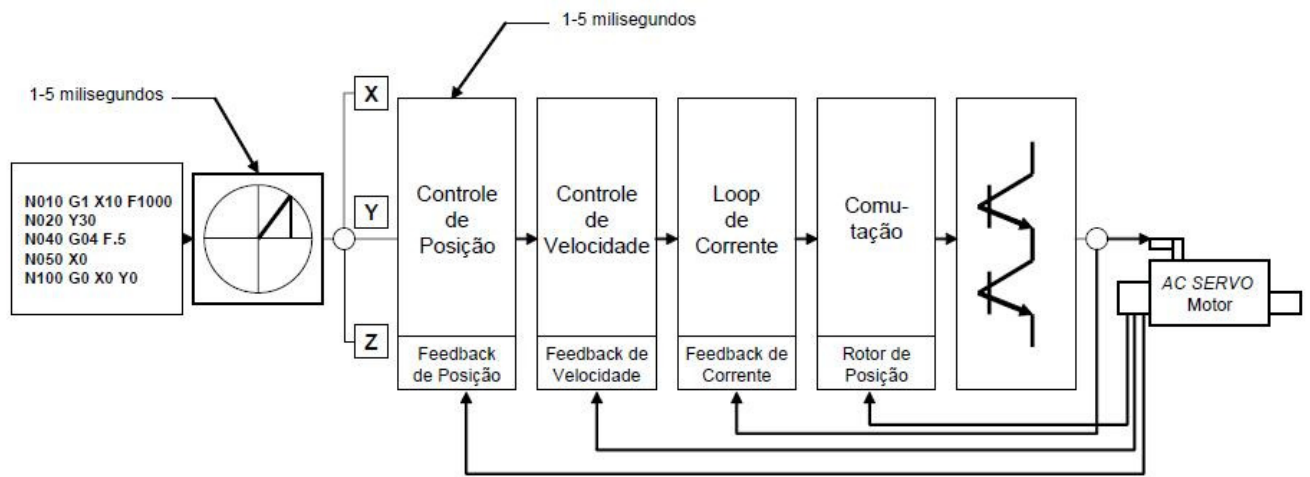


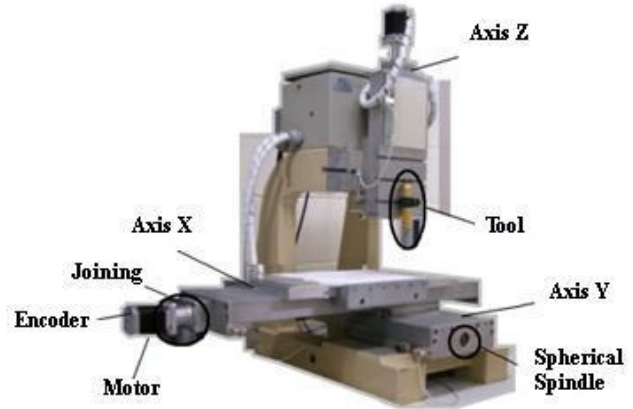
Figura 3.1: Diagrama de blocos do controle de movimentos

Dispositivo utilizado:

Um dispositivo robótico cartesiano Figura 3.2, é muito empregado industrialmente como uma máquina-ferramenta de controle numérico (CNC) para prototipagem de placas de circuito impresso – *Printed Circuit Board* (PCB), a partir da remoção mecânica de uma fina camada de cobre ao longo do contorno de cada trilha individual a fim de isolá-la eletricamente do restante do cobre da placa. Trata-se de um sistema mecânico de grande precisão, constituída de guias de rolamentos de fuso esférico e acionamento através de motores elétricos (Souza, 1998).



a) Dispositivo Robótico Cartesiano



b) Descrição dos movimentos do dispositivo

Figura 3.2: Dispositivo Robótico Cartesiano.

As máquinas-ferramenta devem executar movimentos automáticos, precisos e consistentes, de acordo com as instruções contidas no programa de usinagem previamente elaborado, sem necessidade de intervenções por parte do operador. Entretanto, esses dispositivos possuem uma arquitetura de controle pouco flexível e aberta, e com o passar dos anos podem se tornar obsoletas, necessitando de um *retrofitting* do seu sistema de acionamento e controle.

Um dos objetivos dessa aplicação é utilizar técnicas de prototipagem rápida, para implementação de eletrônica embarcada aberta, nesse dispositivo mecânico de grande precisão. Nesse caso, utiliza-se a solução da National Instruments – o Control Motion, que funciona na plataforma para o ambiente de programação visual LabVIEW™, tornando um elemento facilitador, para que essa plataforma industrial opere a partir de imagens geradas em arquivos,

O principal objetivo dessa aplicação é a geração automática das instruções de usinagem da máquina-ferramenta a partir dos dados gráficos fornecidos por um programa de CAD ou sistema de Visão Industrial, a partir de instruções geradas automaticamente pelo pós-processador implementado para geração de comandos utilizando o código G, linguagem padronizada de

operação de máquinas CNC (Souza, 1998). A figura 3.3, apresenta um fluxograma de funcionamento do sistema.

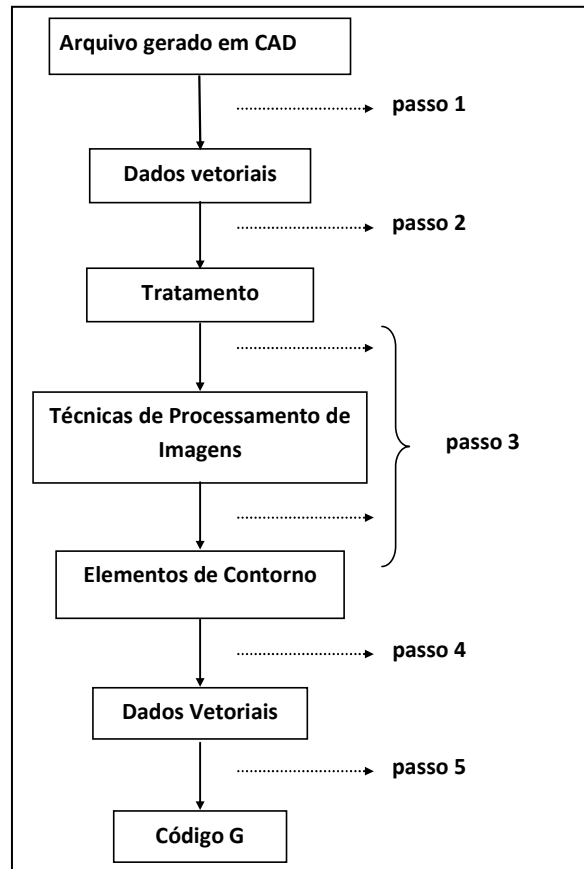


Figura 3.3: Fluxograma do software de funcionamento

3.2 Sistemas de Acionamento e Controle

Numerosas pesquisas têm sido feitas na área de desenvolvimento e implementação de controladores para máquinas CNC. Nos últimos anos são estudados diferentes tipos de modelos dinâmicos e várias estratégias de controle para melhorar seu desempenho. Os sistemas de controle estão incluindo complexidades da dinâmica do sistema, observando as perturbações e mudanças no movimento. Adicionalmente é reconhecida a diminuição da precisão devido aos efeitos dos atritos, da temperatura, das vibrações, forças de corte, entre outros.

Uma das problemáticas nos controladores é atingir alta precisão, mantendo a estabilidade e robustez no sistema. Devido a isso, novas técnicas de controle devem ser propostas. Controladores clássicos de tipo Proporcional (P), Proporcional Derivativo (PD) ou Proporcional Integral Derivativo (PID) são técnicas utilizadas frequentemente pela indústria. No entanto, esses tipos de sistemas de controle não possuem a capacidade de seguir comandos de posição em alta velocidade, minimizando o erro de trajetória (ALTINTAS, 2000). Segundo Van den Braembussche et al.(1996) alguns controladores clássicos não melhoram o desempenho dessas máquinas, quando são produzidos efeitos dinâmicos, resultantes de mudanças em velocidade, e quando é requerida alta velocidade em contornos e bordas. Para isso, aplica-se ao sistema baixa velocidade, a fim de minimizar o erro de seguimento. Os erros dinâmicos resultantes de mudanças rápidas nos comandos de referência podem ser cancelados parcialmente ou totalmente introduzindo o controle em *feedforward*.

Para dar solução aos problemas de precisão em termos de diminuição de erro de trajetória e altas velocidades, dois enfoques podem-se encontrar na literatura como são: O *Controle de Seguimento* o qual diminui o erro de seguimento em cada eixo de movimentação. Esse indiretamente resulta na redução do erro de contorno. Um segundo enfoque conhecido como o *Controle de Contorno*, o qual estima o erro de contorno em tempo real e utiliza essa estimativa na lei de controle em realimentação.

Algumas das estratégias de controle para melhorar o desempenho das máquinas mediante controle clássico são apresentadas por Altintas (2000). São avanços em controle preditivo (ALTINTAS, 2000; DUMUR et al., 2008; SUSANU et al., 2005), controle robusto e adaptativo (ERKORKMAZ; ALTINTAS, 2001; ALTINTAS, 2000; KAMALZADEH; ERKORKMAZ, 2007; VAN DEN BRAEMBUSSCHE et al., 1996; XI et al., 2010; RAMESH et al., 2005, PAPADOPOULOS; CHASPARIS, 2002), e controle por compensação (XI et al, 2010; KAMALZADEH; ERKORKMAZ, 2007, ZHANG et al., 2010, POGNET, P. et al., 1994; VARANASI, K.; NAYFEH, 1994).

É possível que os dispositivos máquinas-ferramenta CNC melhorem o desempenho utilizando controladores simultâneos para os eixos de movimentação, os quais são projetados como controladores tipo MIMO (múltiplas-entradas múltiplas-saídas), como é mencionado por Altintas e Sencer (2010), cujo trabalho desenvolveu uma estratégia de controle de contorno para alta velocidade em uma máquina de cinco eixos de movimentação, com o objetivo de minimizar erros de contorno, utilizando um controlador de modos deslizantes tipo MIMO. Ramesh e Mannan (2005) mencionam que é necessário gerar controladores que consigam monitorar e controlar posição e velocidade juntas, sincronizando os movimentos dos eixos com alta velocidade para seguimento de trajetórias complexas.

O desempenho das máquinas em termos de minimizar o erro de contorno, pode ser melhorado, utilizando estratégias de controle que sejam aplicadas por eixos independentes pelo sistema completo, por meio de estruturas de antecipação e por melhoras nas referências de trajetórias. Avanços nesse sentido são encontrados nos trabalhos de Uhlmann et al. (2012), que desenvolve um controlador com filtro de Kalman o qual possui o modelo mecânico da estrutura. Esse modelo é aplicado para calcular o deslocamento dinâmico dos eixos da máquina com relação ao ponto central da ferramenta e a sua velocidade. Esses dados de posição e velocidade são utilizados como referência do controlador. Altintas e Khoshdarregi (2012) propõem um controlador de erro de contorno com um sistema para evitar as vibrações da máquina, utilizando filtros na entrada da referência dos eixos de comando, evitando a excitação de modos estruturais no sistema.

Erkorkmaz e Altintas (2001) mencionam que a problemática de melhorar o erro de contorno através do incremento da largura de banda no sistema por um ganho alto de realimentação possui diferentes limitações em termos de sensibilidade na medição do ruído, excitando a dinâmica não modelada e causando oscilações indesejadas. Essa proposta da dinâmica de malha fechada em cascata emprega estratégias de filtragem para cancelar alguns sinais em frequência. No entanto seus componentes estáveis também são cancelados, obtendo uma função de transferência do sistema que anula grande parte de sua dinâmica.

Um dos controladores de seguimento é proposto por Tomizuka (1987 *apud* ALTINTAS et al., 2000) chamado **ZPTEC (Zero Phase Error Tracking Controller)**. Esse utiliza um algoritmo de controle digital em *feedforward* para seguimento de trajetória. Esse controlador permite cancelar a fase do sistema assegurando que sua resposta em frequência esteja entre a saída desejada e a saída atual. Outras propostas são feitas para melhorar esse tipo de controlador ZPTEC a fim de limitar o conteúdo de altas frequências, focados em melhorar erros nas trajetórias de bordas e contornos. A largura de banda geral do sistema e a precisão do sistema são incrementadas com esse controlador, desde que o modelo seja preciso e que não tenha variações com o tempo, mudanças da inércia, atritos e dinâmicas não modeladas. Mesmo assim, é mais difícil manter a precisão do ZPETC.

Algumas soluções para limitar conteúdos em altas frequências do sinal de controle é utilizar filtros antes do controlador; como é proposto por Butler et al. (1991). Outras opções para melhorar a precisão da resposta dos controladores é realizar uma identificação em baixas frequências e em malha da dinâmica do sistema com o intuito de adaptar os controladores mais próximo aos parâmetros reais do sistema, trabalhos nesse sentido são propostos por Tung e Tomizuka (1993 *apud* ERKORKMAZ e ALTINTAS 2001), soluções adaptativas foram realizadas por Tomizuka (1987) e Pristchow e Kosiedowski (1997 *apud* EKORMAKZ e ALTINTAS, 2001).

3.3 Sistema de Transmissão de Movimentos

Tipos de Sistemas de Transmissão Mecânica:

A energia produzida pelos motores de cc, necessita ser transmitida mecânicamente até a carga, para isso são empregados vários sistemas mecânicos, sendo por exemplo:

Cremalheira, que utiliza engrenagem linear e pinhão.

Correia e polia, utilizando várias polias interligadas por correias.

Engrenagem, utilizando-se de caixa de engrenagens.

Parafuso sem fim, provido de um fuso rotativo para transmissão de força.

Esteira, montagem mecânica composta de uma esteira com o objetivo de transporte da carga.

Como ilustrado na figura 3.4, pode-se observar os tipos de sistemas de transmissão mecânica.

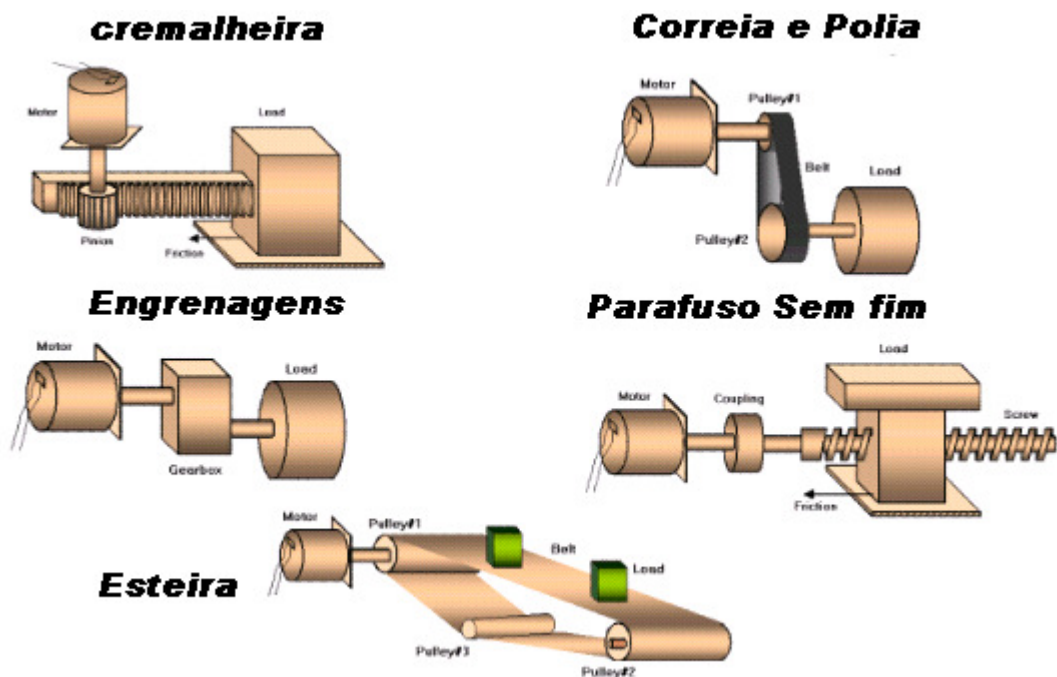


Figura 3.4: Tipos de sistemas de transmissão mecânica.

3.4 Acionamento através de motor de passo

O motor de passo é um transdutor que converte energia elétrica em movimento controlado através de pulsos, o que possibilita o deslocamento por passo, onde passo é o menor deslocamento angular.

Com o passar dos anos houve um aumento na popularidade desse motor, principalmente pelo seu tamanho e custo reduzidos e também a total adaptação por controle digitais.

Outra vantagem do motor de passos em relação aos outros motores é a estabilidade. Quando se deseja obter uma rotação específica de um certo grau, calcula-se o número de rotação por pulsos o que possibilita uma boa precisão no movimento.

Controladores para motor de passo

Nesta etapa serão abordados circuitos que podem controlar os motores adequadamente. Destacam-se que como as cores dos fios que levam energia às bobinas não são padronizadas. Portanto não há comentário sobre a ordem certa de polarização utilizando as cores dos fios.

A primeira etapa, aborda o controle de um motor de passo diretamente pelo computador.

Através da porta paralela, como visto na figura 3.5, tem-se como objetivo controlar um motor de quatro fases e unipolar através da excitação por passo-inteiro. Pesquisando os drivers existentes, foi descoberto o CI ULN2003 que é um 7bit 50V, 500mA TTL-input NPN darlington driver, que funciona como amplificador. Obviamente toda a lógica deve ser exercida pelo computador inclusive a da ordem de excitação das bobinas. A seguir o circuito elétrico mostrando a ligação entre o motor e a porta.

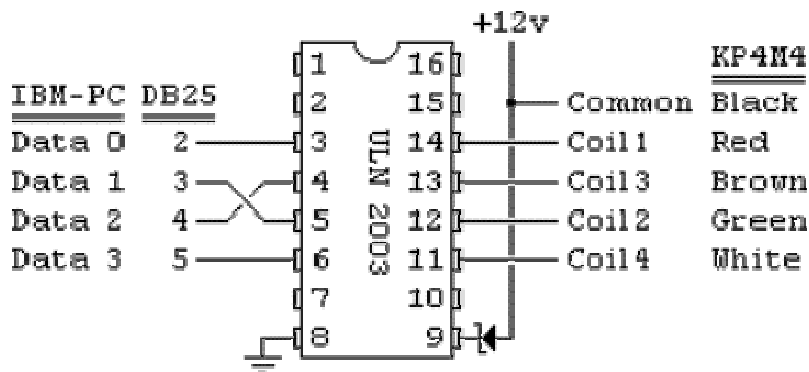


Figura 3.5: - Ligação entre o motor e a porta

Destaca-se que o diodo zener foi utilizado como intuito de absorver o campo eletromagnético reverso produzido pelo motor quando o mesmo é desligado. A inversão observada nos pinos de entrada 3 e 4 do ULN2003 são necessárias para manter a ordem certa de ativação das bobinas.

Agora pode-se observar o controle ser feito através de componentes discretos. Utilizando um contador Johnson CMOS, ilustrado na Figura 3.6, com o objetivo de controlar a etapa de potência não mostrada, pode-se exercer um simples comando ao motor somente através dos clock. Esse circuito tem a desvantagem de não exercer o controle da direção e observa-se que o mesmo executa o procedimento de passo inteiro.

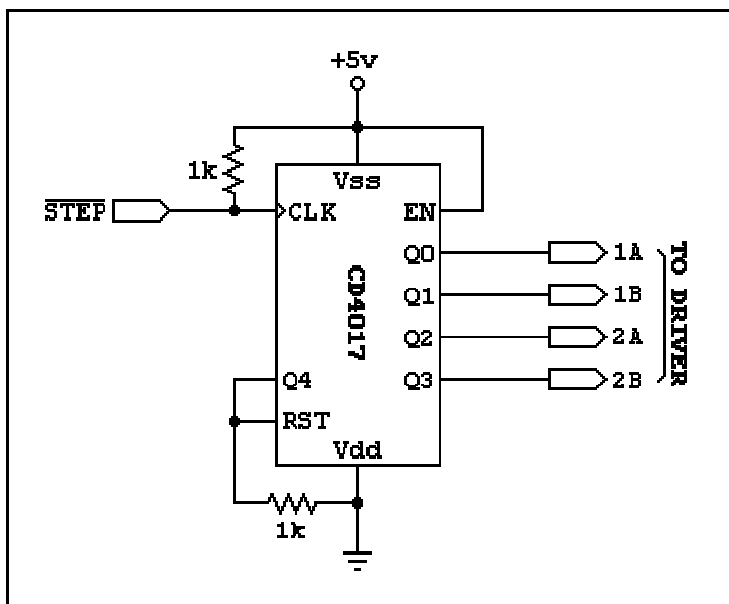


Figura 3.6: Contador CMOS

Utilizando agora portas lógicas e flip-flops, Figura 3.7, pode exercer-se o controle tanto da direção (através de DIR) quanto da velocidade do motor. Abaixo segue o esquema a ser analisado.

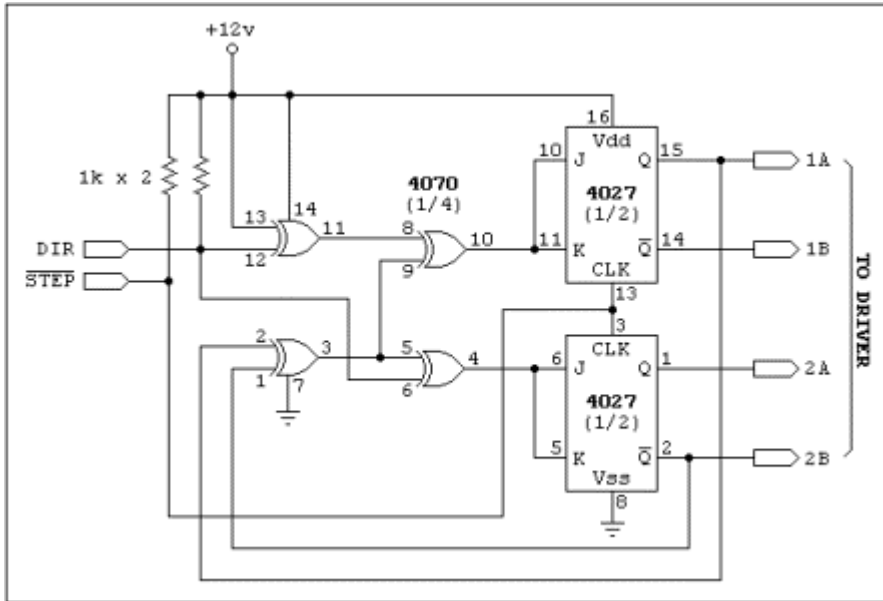


Figura 3.7: Controle de direção utilizando-se flip-flops

3.5 Acionamento através de motor cc

Nesta etapa será apresentado apenas um resumo do funcionamento do motor DC, já que esse não é o objeto principal desse estudo. Também é relevante falar desse motor, pois com ele pode-se introduzir os conceitos de controle e feedback, que servirão de introdução aos motores de passo.

O funcionamento básico do motor DC está fundamentado na Força de Lorentz aplicada em uma carga em movimento dentro de um campo magnético ($F = qvB$). Considerando-se uma espira de corrente inserida num campo magnético criado por um ímã permanente, em que há uma corrente criada por uma bateria (fonte DC).

De uma forma simplificada, a simples passagem dessa corrente faz com que apareçam duas forças de sentidos contrários, aplicadas uma em cada lado da espira. Essas forças criam um torque que, obviamente, faz a espira girar, transformando a energia elétrica da corrente em energia cinética num eixo acoplado às espiras. A direção da rotação depende da polaridade da bateria e da direção das linhas de campo magnético criadas pelo ímã.

Um motor real é composto de conjuntos múltiplos de espiras, dispostas de tal forma que as forças que agem em cada espira sejam somadas e produzam um torque significativo para uma possível aplicação.

Os motores DC são utilizados, por exemplo, em aplicações como o posicionamento de um braço de robô. Mas eles apresentam uma grande desvantagem. Para que um computador dê um comando para que o braço se mova para uma determinada posição com precisão, é necessário um complicado circuito externo provido de sensores de posição, que informe ao computador que o braço já está na posição determinada (feedback).

Como principais vantagens na utilização de motores de passo pode-se destacar como tamanho e custo reduzido, total adaptação a lógica digital, o que permite o controle preciso da velocidade direção e distância, características de bloqueio, pouco desgaste e dispensa realimentação, e como desvantagem pode-se destacar a má relação potência/volume e principalmente a necessidade de sistemas de controle relativamente complexo

3.6 Software aberto para supervisão e controle de movimentos

O sistema de supervisão e controle foi implementado em linguagem LabVIEW™, muito utilizado industrialmente, em projetos de dispositivos relacionados as áreas de medição e controle podem utilizar LabVIEW™ para criar aplicações personalizadas que operam em plataformas NI (National Instruments) com E/S reconfiguráveis baseadas em FPGAs. Juntos, o LabVIEW™ FPGA e o hardware NI para E/S reconfiguráveis permitem a criação de uma plataforma flexível

para o desenvolvimento de sistemas sofisticados que antes somente eram possíveis com hardware projetado de forma dedicada.

O software LabVIEW™ pode ser usado para controle e supervisão. O controle das juntas robóticas é realizado através de uma interface A/D que comanda os motores das juntas. Um programa de supervisão e controle, residente num computador é responsável pelo gerenciamento e controle das informações provenientes de sensores e atuadores do sistema.

Em uma interface de visualização implementada em ambiente Windows, foram desenvolvidas telas de supervisão que coletam as informações dos sensores, dos programas de tratamento matemático (modelagem cinemática da mesa), as informações sobre o sistema (velocidade, parâmetros do regulador, números de pontos da trajetória etc), da inicialização e da calibração automatizada.

A interface de aquisição de dados utilizada nesse trabalho foi a Placa Motion PCI 7344 da National Instruments, conforme ilustrado na Figura 3.8.

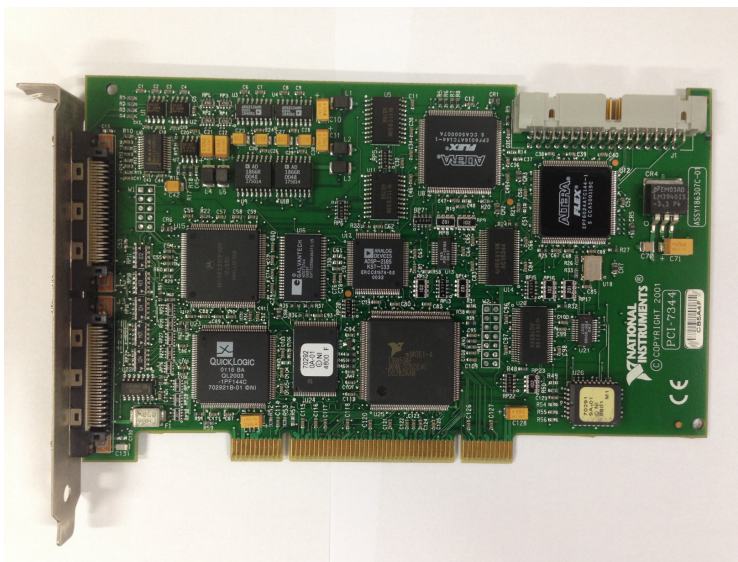


Figura 3.8: Placa Motion PCI 7344 da National Instruments.

Essa interface recebe as informações provenientes dos encoders para determinação da posição atual das juntas constituintes da mesa. Essas informações são comparadas com os valores

de referência, logo após essa interface calcula o algoritmo de controle envia para a saída do controlador enviando essas informações (posição, velocidade) aos acionadores de cada grau de liberdade da mesa Figura 3.9.

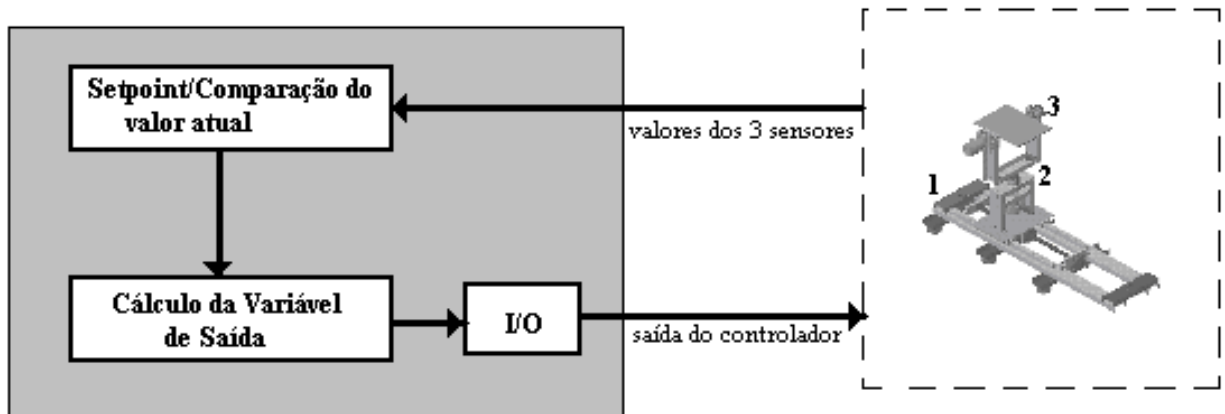


Figura 3.9: Esquema do sistema controlado.

Todos os valores de referência são calculados simultaneamente, mesmo a cinemática inversa com não linearidades. Outros tipos de sensores podem ser incluídos na bancada ou simplesmente através de uma VI (Virtual Interface) do software LabVIEW™. Normalmente, o projeto de uma VI, deverá conter os seguintes elementos de base:

Painel Frontal: É uma interface interativa entre o usuário e o programa. É onde o usuário entra com os dados utilizando o mouse ou o teclado, e então vê os resultados na tela do computador. Isto significa que o Painel Frontal é uma janela de execução, onde são encontradas as entradas de informação (control) e as saídas de informação (indicator)

Diagrama de blocos: É a representação de um programa ou algoritmo. É onde o programador cria o seu programa. O programa consiste de menus que definem a movimentação automática ou manual da mesa. A movimentação manual é determinada apenas indicando quais motores devem ser acionados. A movimentação automática é controlada pelo programa desenvolvido para tal fim Figura 3.10.

Tela principal: A ser utilizada para o controle direto da plataforma, foi elaborado um programa em linguagem visual. Esse programa de interface se comunica com a placa de interface com o hardware, a qual controla diretamente os motores, ilustradas na Figura 3.11.

Tela contendo os diagramas de conexão de cada bloco: Os diagramas são as conexões de cada bloco, e apresentam características específicas ao projeto implementado, de acordo com os elementos que serão implementados na VI, as conexões são realizadas nesse diagrama, ilustrado na Figura 3.12.

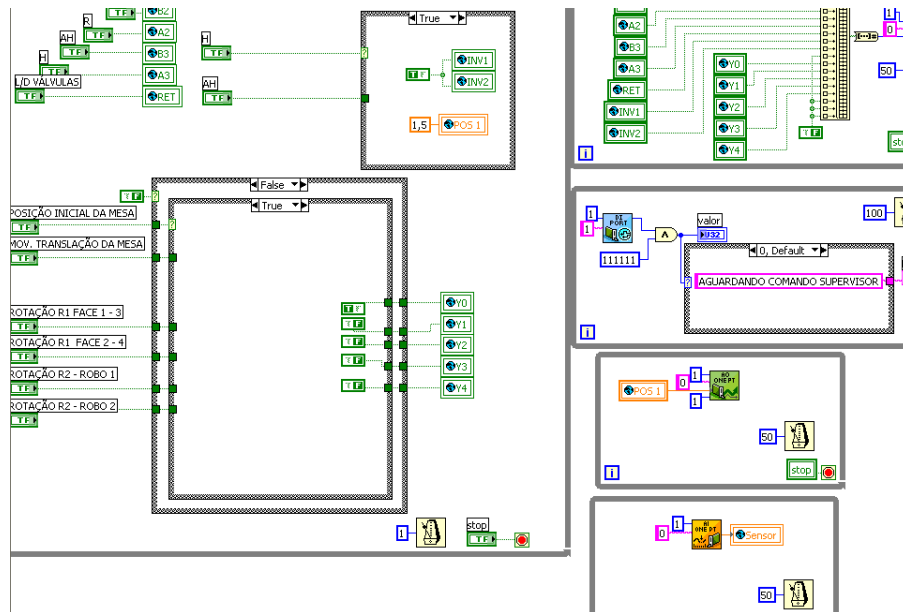
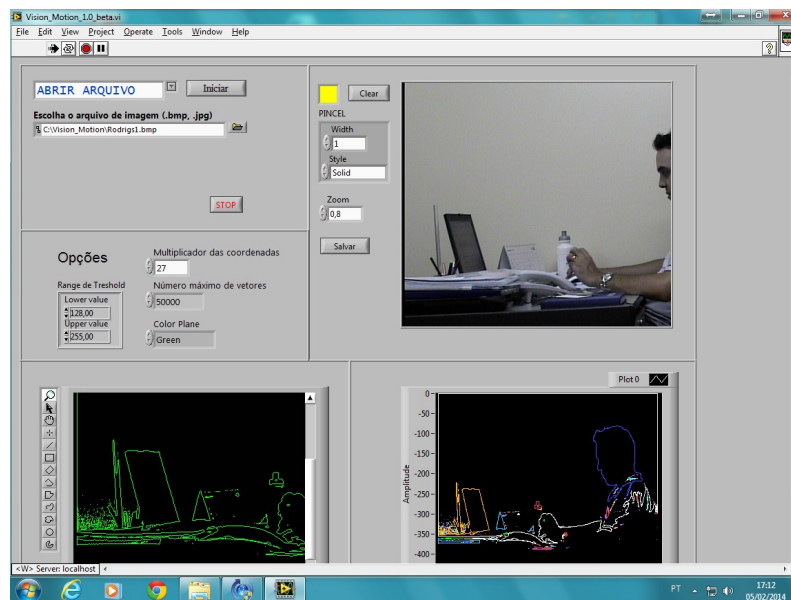


Figura 3.10: Tela diagrama de blocos implementados em LabVIEW.



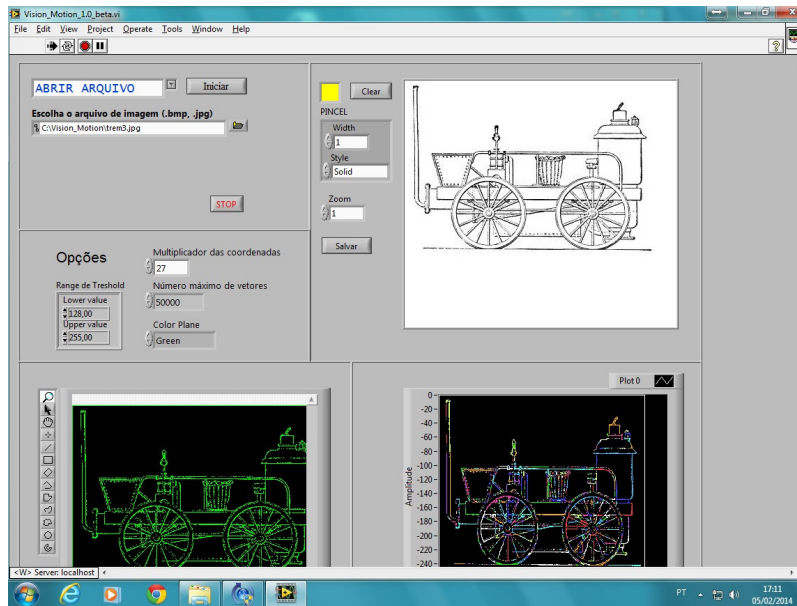


Figura 3.11: Proposta de pagina HTML - Painel de Controle Implementado em LabVIEW.

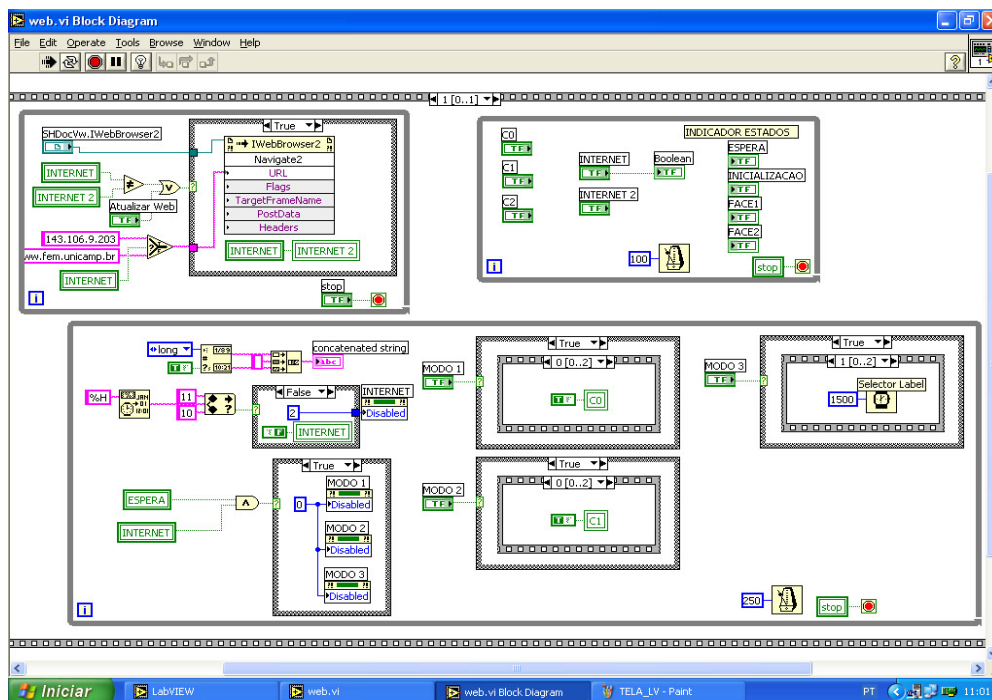


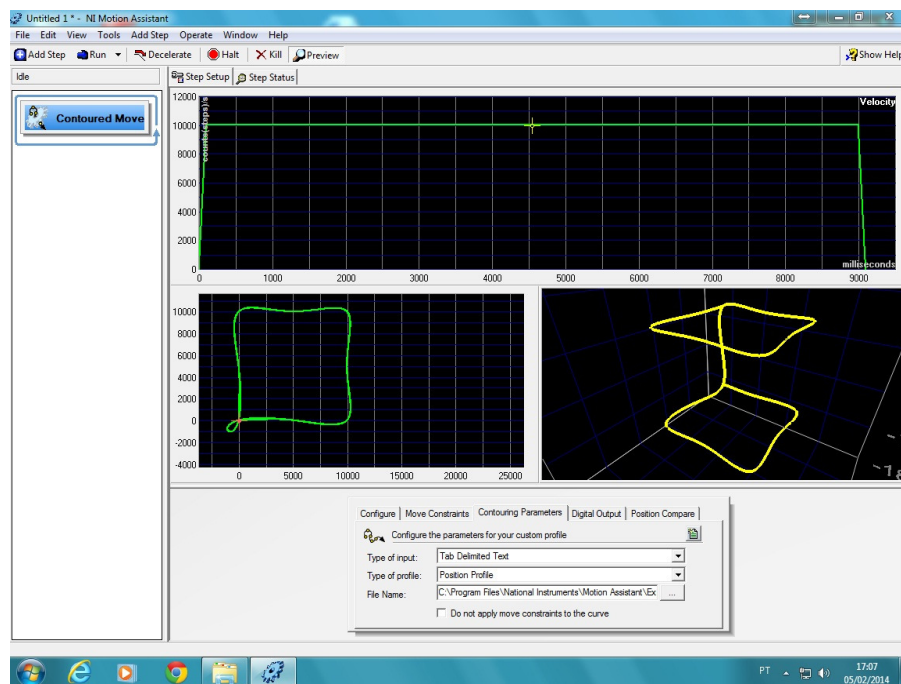
Figura 3.12: Diagrama de blocos implementados em LabVIEW.

3.7 Descrição do software Labview Motion

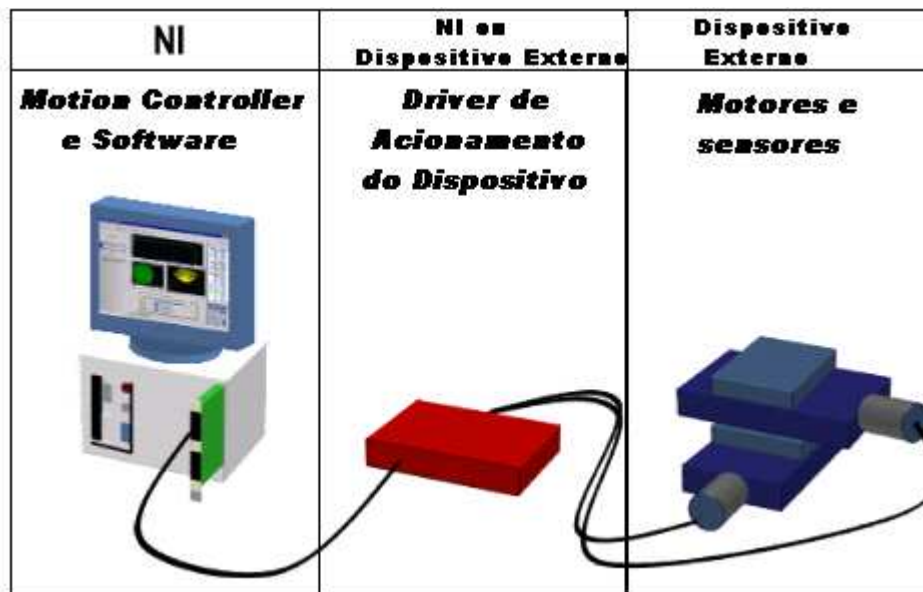
3.7.1 Plataforma NI Motion

O NI motion da National permite a integração de diferentes dispositivos que compõem o sistema de controle de uma aplicação, constituindo-se um elemento facilitador de integração, ilustrados na Figura 3.13 a e Figura 3.13 b.

Essa plataforma é baseada nos principais elementos constituintes do sistema de controle, sendo integrados através de etapas de configuração, protótipo e ambiente de aplicação e desenvolvimento, ilustrados na Figura 3.14 a e Figura 3.14 b.

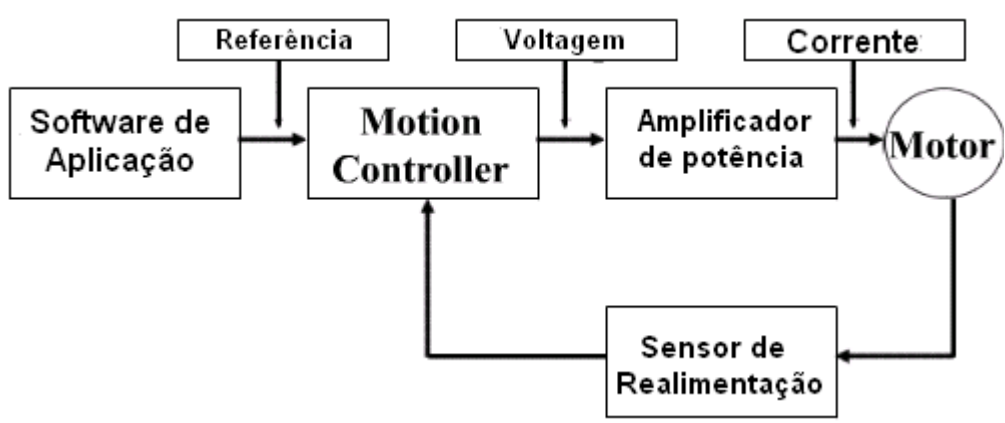


a) Tela típica do Aplicativo.

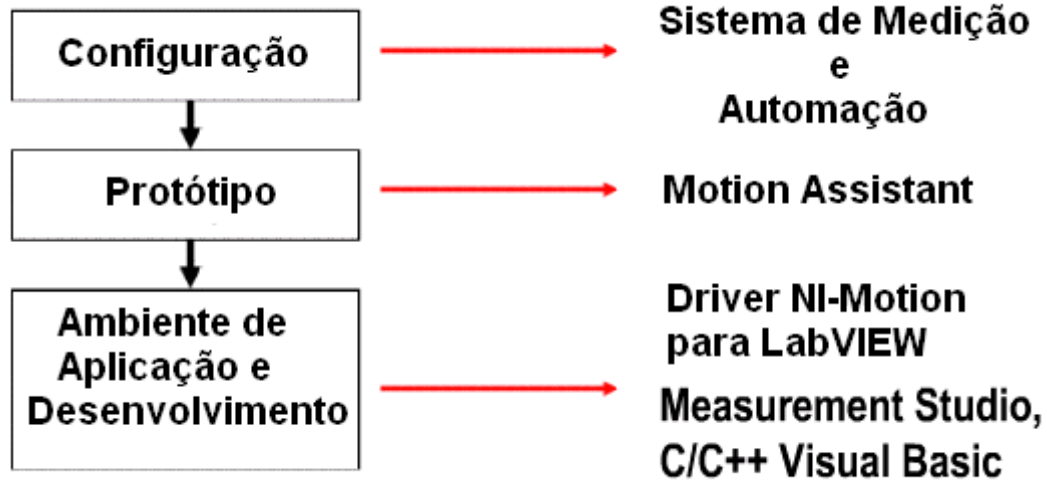


b) Conceito de utilização

Figura 3.13: Control Motion da NI



a) Elementos Básicos de um Sistema de Controle



b) Etapas de Prototipagem a partir do Software Control Motion.

Figura 3.14: Sistema de Controle e etapas de implementação do Control Motion da NI

3.7.2 Principais características do software NI Motion driver

Essa plataforma pode funcionar nos sistema operacional Windows 7 ou Linux, Figura 3.15. apresentando as seguintes características:

- a) Sistema Operacional Windows e LabVIEW Real-Time Systems;
 - VIs e funções para LabVIEW, Visual Basic e C.
 - Measurement and Automation Explores.
 - Configuração do Motion e outros componentes.
 - Ajuste de parâmetros de controle de motores.
- b) Outros Sistemas Operacionais;
 - Hardware de controle Motion DDK.
 - Linux e drivers VxWorks para sistemas (www.sensingsystems.com).

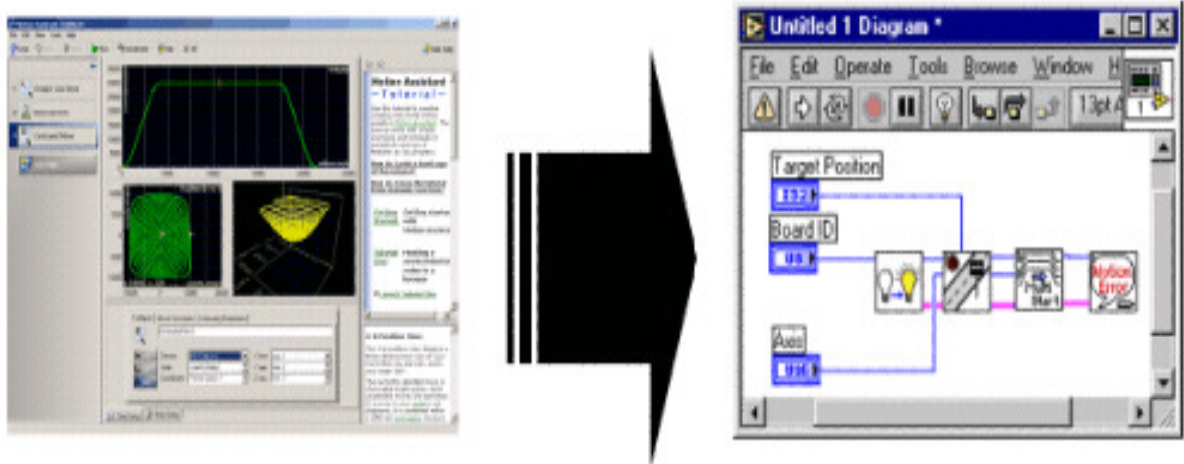


Figura 3.15: Control Motion da NI – assistente do software de prototipagem.

As principais características do software Motion Control estão baseadas na interatividade com os diferentes elementos constituintes de uma malha de controle, Figura 3.16, que serão descritas nos itens a seguir.

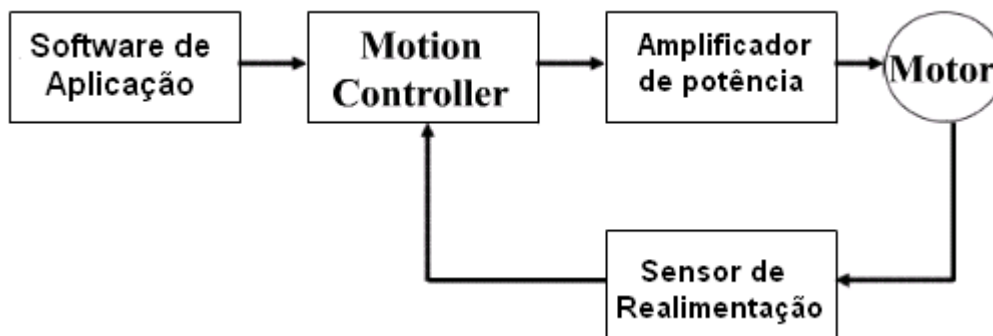


Figura 3.16: Diferentes elementos de uma malha de controle.

3.7.3 Software de Aplicação

O NI motion possui um assistente de software para prototipagem, que apresenta as seguintes características:

- a) **Configurabilidade:** fácil utilização, e interatividade com o usuário, não necessitando de nenhuma programação.

- b) **Programação:** alta flexibilidade e facilidade de integrar-se com outros componentes.

3.7.4 Motion Controller

A Figura 3.17, apresenta um exemplo de perfil de velocidade, disponível no Motion Controller para controle de trajetória para uma junta simples. As principais funcionalidades NI Motion Controller são apresentadas a seguir:

- a) Cálculo das trajetórias de referência para cada movimento comandado interagindo facilmente com o usuário.
- b) Cálculo do torque necessário para acionamento para o drive ou amplificador do motor.
- c) Sistema de Supervisão permite o monitoramento completo do sistema e paradas de emergência.
- d) Controlador PID em malha fechada.

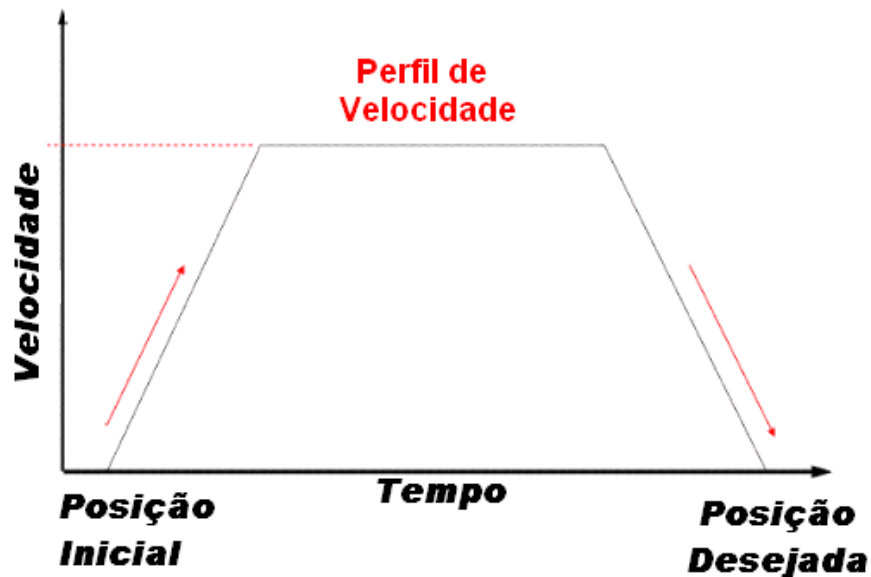


Figura 3.17: Gerador de movimentos para uma trajetória simples.

3.7.5 Requisitos do Amplificador de Potência (Driver de acionamento)

Os principais requisitos para especificação do sistema de acionamento (motores) e drive de potência são os seguintes:

- a) O drive do motor deverá ser especificado em função de sua tecnologia: Por exemplo motores de passo com 2 fases, requerem um drive de potência para motores de passo com 2 fases etc.
- b) O drive de acionamento do motor deverá ser compatível com as suas características tais como: corrente de pico, máxima corrente contínua, tensão de alimentação etc.

3.7.6 Seleção de Motores e Hardware Mecânico

A Tabela 3.1 sintetiza dos principais critérios que deverão ser utilizados no processo de seleção de Motores para acionamento e controle de dispositivos mecatrônicos. Os principais requisitos para especificação do sistema de acionamento (motores) e drive de potência são os seguintes:

- a) Análise do sistema mecânico.
- b) Seleção de Motores e sistema mecânico de transmissão de potência.
- c) Seleção do drive de potência compatível com o motor e escolha do controlador.

Acionamento elétrico	Características Positivas	Características Negativas	Aplicações
Motor de Passo (Stepper Motors)	<ul style="list-style-type: none"> ▪ Baixo Custo ▪ Não necessita de realimentação de posição ▪ Bom torque até o final do curso a baixa velocidade ▪ Bom para posicionamento básico 	<ul style="list-style-type: none"> ▪ Apresenta ruído e ressonância; ▪ Apresenta torque ruim a altas velocidades; ▪ Não é bom para cargas variáveis. 	<ul style="list-style-type: none"> ▪ Posicionamento ▪ Micromovimento
Brushed Motors (Motor com Escovas)	<ul style="list-style-type: none"> ▪ Custo Relativamente Baixo ▪ Velocidade Moderada ▪ Usa drives simples 	<ul style="list-style-type: none"> ▪ Necessita manutenção; ▪ Não é bom para ambientes limpos; ▪ Apresenta faíscas elétricas; ▪ Requer realimentação . 	<ul style="list-style-type: none"> ▪ Controle de Velocidade ▪ Controle de alto desempenho
Brushless Motors (Motor sem Escovas)	<ul style="list-style-type: none"> ▪ Sem manutenção ▪ Longo tempo de vida ▪ Alta velocidade ▪ Bom para ambientes limpos 	<ul style="list-style-type: none"> ▪ Alto custo; ▪ Usa drives mais complicados; ▪ Requer realimentação. 	<ul style="list-style-type: none"> ▪ Robótica ▪ Pick and place ▪ Torque muito alto ou aplicações rápidas

Tabela 3.1: Critérios utilizados para Seleção de Motores NI.

Para correta escolha do sistema de transmissão mecânica necessita-se das seguintes especificações funcionais e tecnológicas:

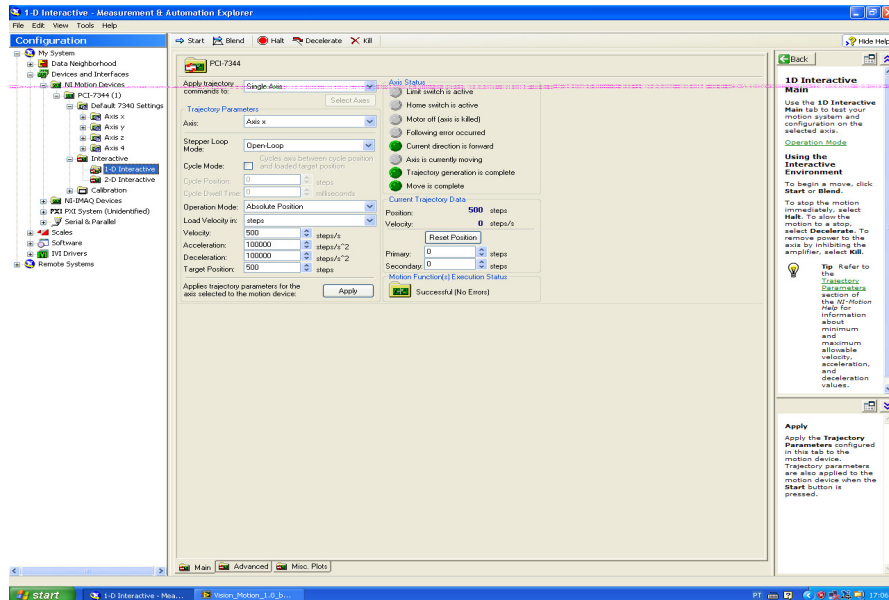
- a) Resolução desejada.
- b) Dimensões requeridas para operação.
- c) Repetibilidade.
- d) Capacidade de Carga.
- e) Tipo do motor.

3.7.7 Proposta de Aplicação

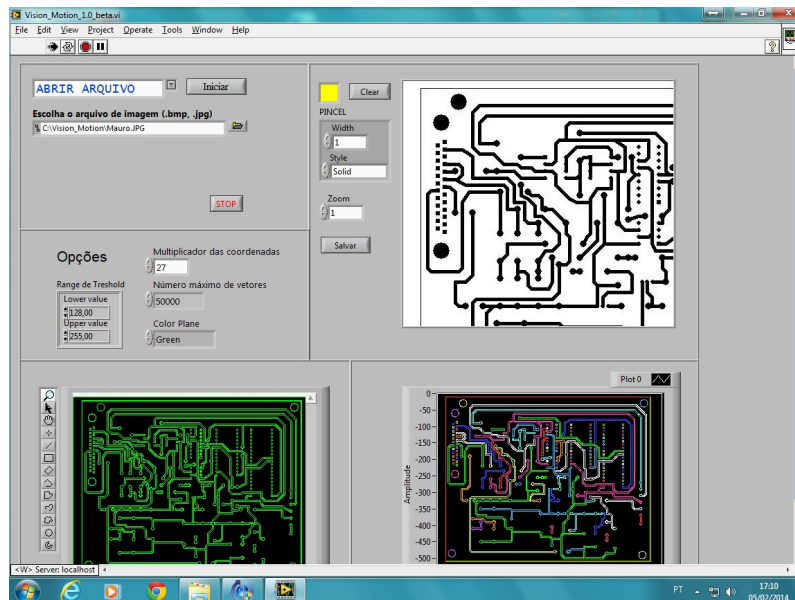
Como proposta de aplicação desse dispositivo foi implementada uma célula para projeto de circuitos impressos, onde a intenção não é a de substituir os métodos convencionais de fabricação, como os métodos químicos abrasivos e os de eletrodeposição seletiva de cobre, mas, apresentar uma técnica complementar para a prototipagem rápida e econômica de placas de circuito impresso, a partir da fabricação da placa definitiva pelos métodos convencionais.

Os resultados obtidos na etapa de usinagem são apresentados a seguir e comprovam a eficiência dessa aplicação em termos de precisão de posicionamento, tempo e custo de

implementação dessa nova arquitetura. A Figura 3.18 (a) e (b) apresentam algumas telas típicas do software implementado no NI Control Motion.



(a) Tela de configuração de parâmetros do sistema de acionamento.

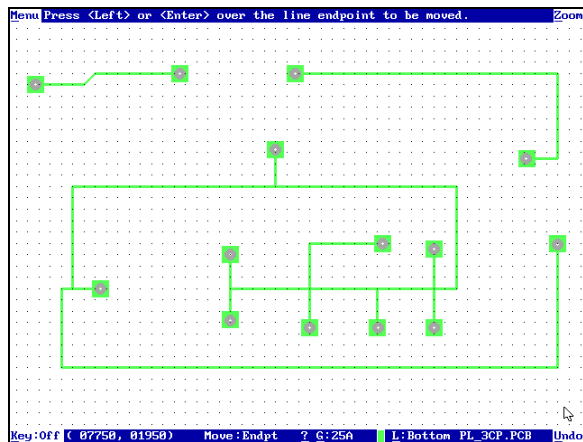


(b) Tela de aquisição e tratamento de imagens.

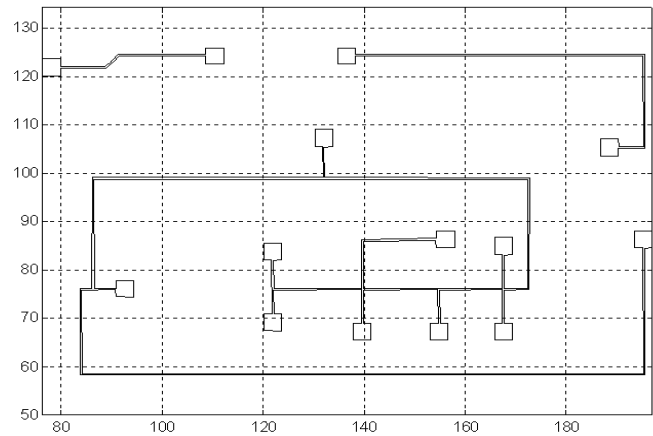
Figura 3.18: Telas típicas do Control Motion da NI Instruments.

3.7.8 Resultados

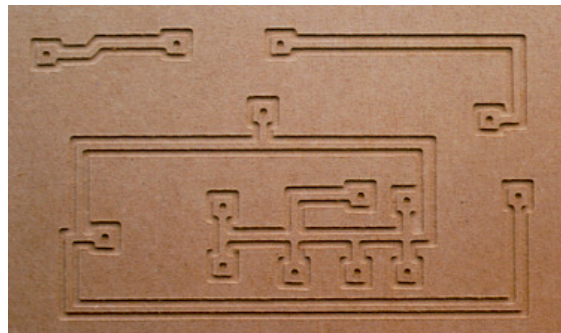
A Figura 3.19 mostra as diferentes fases da implementação proposta para essa aplicação, desde o layout do PCB gerado através de CAD (a), os resultados da digitalização realizada com extração de contornos (b), chegando na confecção final da placa de circuito impresso (c).



a) Layout PCB gerado em CAD.



b) Digitalização com extração de contornos.



c) Confecção Final da Placa de Circuito Impresso.

Figura 3.19: Diferentes fases da implementação proposta.

3.8 Descrição do hardware associado

Os experimentos para validação dos modelos, estratégia de identificação e sistema de controle foram realizados no sistema cartesiano máquina-ferramenta CNC fornecida pelo Laboratório de Automação e Robótica da Faculdade de Engenharia Mecânica da UMICAMP

Pode-se observar detalhe do hardware associado, para obtenção dos resultados mencionados na Figura 3.20.

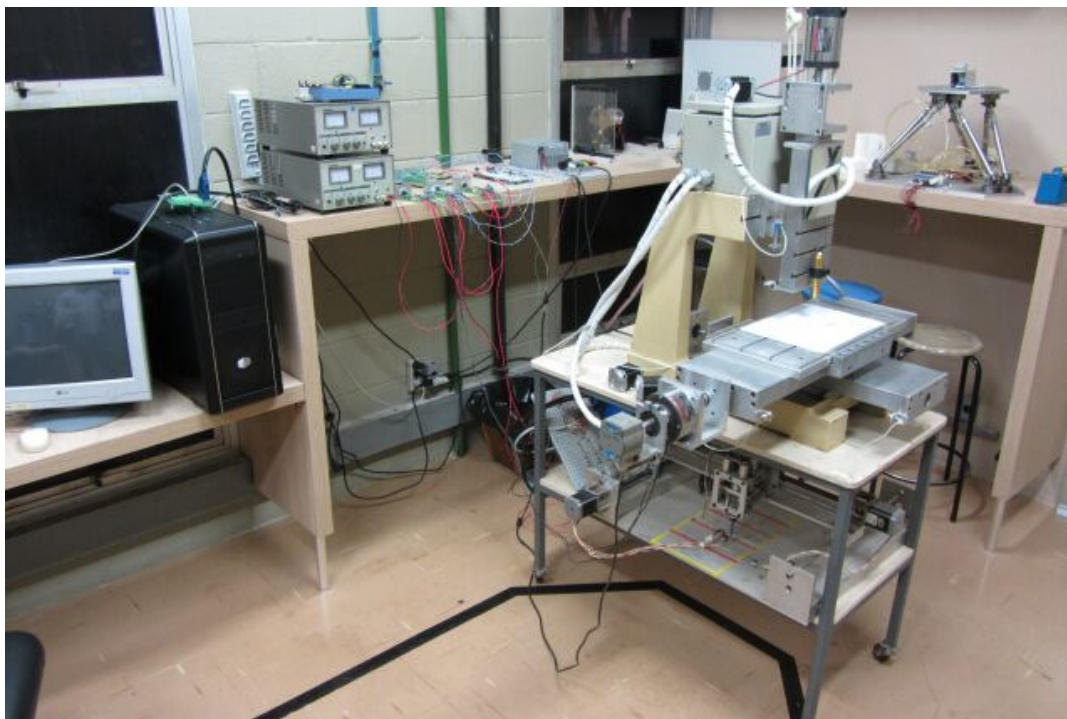


Figura 3.20: Sistema Implementado.

3.9 Descrição de software MPLAB de acionamento e controle de movimentos

O MPLAB IDE (*Integrated Development Environment*) é um software gerenciador de projetos e ambiente de programação para desenvolvimento de aplicações e sistemas embarcados, utilizando PIC Microchip, MCUs e DSCs dsPIC, integrando diversos ambientes de trabalho para programação, simulação e gravação de microcontroladores.

3.9.1 Linguagem de programação

MPLAB IDE foi projetado para trabalhar com vários modelos de Microchip e flexibilidade para uso de ferramentas de linguagem de programação de terceiros. Essas ferramentas de programação podem ser usadas para projetos escritos em linguagem de programação em assembly, C ou linguagem Basic; compilados em linguagem Hexadecimal o código executável pode ser gravado no dispositivo em uma memória flash ou EPROM. Para desenvolvimento de sistemas e análise dos programas. Possui debugger por software por ferramentas internas como o MPLAB SIM ou plugins externos como o Proteus (programa de computador VSM® e Matlab, ou ainda por módulos externos como MPLAB ICD2, MPLAB ICE 200 e PICKIT 2 que fazem o *debugger* no próprio microcontrolador enviando os valores dos registradores para o pc possibilitando a análise dos valores na memória RAM e avaliação do desempenho do programa. Para programação dos microprocessadores, o MPLAB possui interface de programação possibilitando que a mesma seja feita diretamente do MPLAB para a memória flash do micro controlador através da porta serial ou usb pela interface MPLAB ICD2 , MPLAB ICE 2000 e PICKIT 2

O ambiente MPLAB, Figura 3.21, foi utilizado nesse trabalho para elaborar o software, em linguagem “C” (Anexo A) que é utilizado nos microcontroladores responsáveis pela geração dos pulsos PWM (*Pulse with Modulation*), que são enviados as placas de controle dos motores de corrente contínua, instalados nos três fusos, X, Y e Z.

O software desenvolvido em linguagem “C” para geração do PWM, encontra-se disponível no (Anexo A).

```

//*****PIC MOTOR UNICAMP*****//
#include <18F2520.h>
#define device_adc 10
#define HS_NOMDT,PUT_NOLVP,NOPROTECT,BROWNOUT,DEBUG
#define HS_NOMDT,PUT_NOLVP,NOPROTECT,BROWNOUT,DEBUG
#define XT_NOMDT,PUT_NOLVP,NOPROTECT,BROWNOUT
#define delay(clock=20000000)
#define rs232(baud=9600, tx=PIN_C6, rcv=PIN_C7, PARITY=N, BITS=8) //seta rs 232
#define rs232(baud=19200, tx=PIN_C6, rcv=PIN_C7, PARITY=N, BITS=8) //seta rs 232

//*****Variaveis Globais*****//

long int loop = 0;
int32 contador_hora = 0; //Contador da hora para display e ler
long int val_temp = 0; // Valor temporário da Tensão
long int set_pwm1; // setpoint pwm
long int set_pwm2; // setpoint pwm
float real_tensao = 0; // Valor real da tensão

void main(void)
{
    //*****Inicializa como A/D *****//
    setup_adc_ports (RA0,RA1,ANALOG_RA3,REF);
    setup_adc_ports (ANO_TO_ANI_ANALOG);
    setup_adc(ADC_CLOCK_INTERNAL);
    set_adc_channel (0);

    //*****TIMER E PWM *****//
    // saída 1 ligada no Pino RC2 / CCP1
    // saída 2 ligada no pino RC1/ CCP2
    // Entrada analógica ligada em RA0 , valor de 0 até 5 do Motion

    // configura timer 2 de 8 bits *****//
    //setup_timer_1 ( T1_INTERNAL | T1_DIV_BY_1); //timer 1 clock externo div/1
    //setup_timer_3 ( T3_INTERNAL | T3_DIV_BY_1); //timer 1 clock externo div/1
    //setup_timer_1 ( T1_DIV_BY_4 ); //timer 1= 1,004 MHz
    setup_timer_2 ( T2_DIV_BY_16,248,1 ); //timer 2= 1,004 MHz
}

```

Figura 3.21: Tela típica do ambiente de desenvolvimento IDE MPLAB.

3.10 Automação e Integração com outros dispositivos

Para a obtenção dos dados a serem monitorados, pode-se ter uma integração com outros dispositivos. Essa integração pode utilizar hardware apropriado, como sensores ou atuadores, que coletam ou fornecem informações de estado das variáveis monitoradas.

Conjuntamente com esses dispositivos periféricos de campo, é necessário a utilização de um software, que trabalha em conjunto com esses dispositivos, executando a leitura de informações ou executando a atuação, enviando alguma informação de processo.

3.11 Sensores e dispositivos de segurança

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 serão adicionados contadores de pulsos, com pulsos A e B, defasados de noventa graus na máquina-ferramenta, pois os dados a serem monitorados são utilizados pelo sistema Labview Motion .

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.

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 essa série de requisitos pretende-se também obter a integração e a interoperabilidade dos transdutores via rede. Na Figura 3.22 é apresentada a família de normas IEEE 1451.

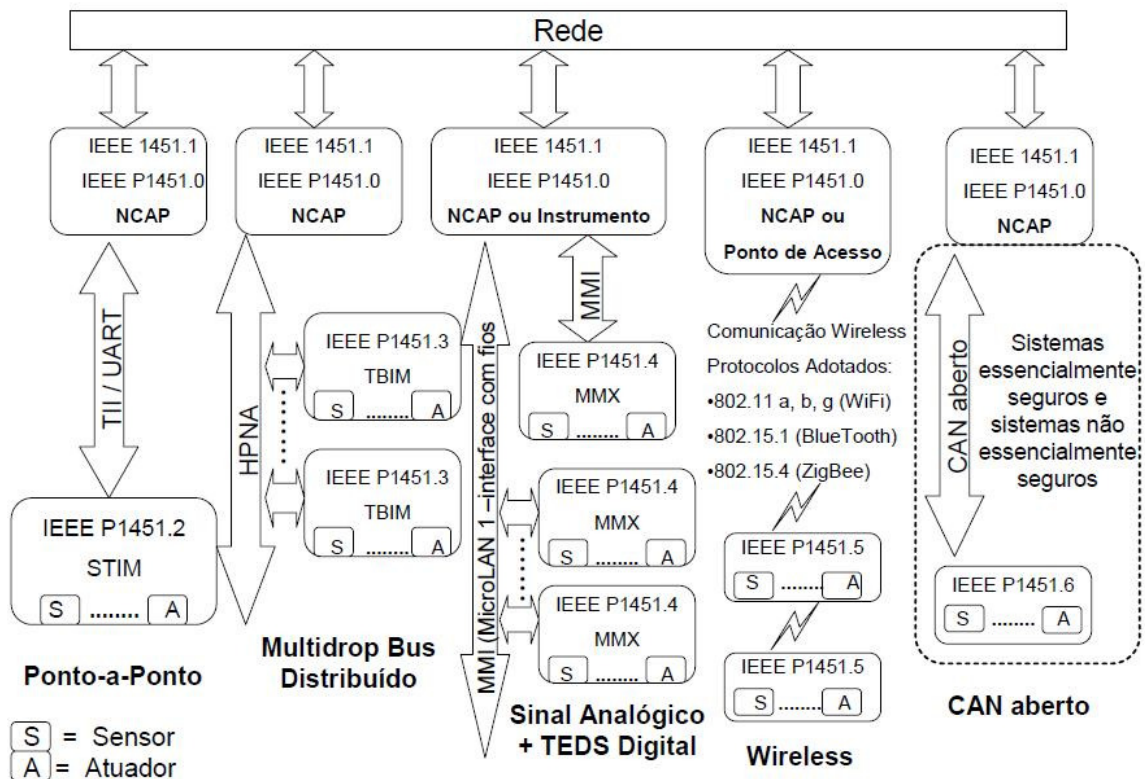


Figura 3.22: Família de normas IEEE 1451

3.12 Interface com Sistema de Visão Robótica

A utilização de um sistema de Visão Automatizada está cada vez mais frequente na área de Mecatrônica, e muitos fabricantes vem desenvolvendo sistemas dedicados para visão de máquina e processamento de imagens. Esses pacotes contêm um grande numero funções de visão, que poderão ser implementadas em outros softwares de Instrumentação Industrial, tais como o LabVIEW da National Instruments, que já possuem funções dedicadas para diferentes aplicações industriais tais como sistemas de inspeção, alinhamento, identificação e medição visual. Esses ambientes possuem interatividade para configurar, testar e distribuir aplicações de visão sem muita programação, trabalhando diretamente com muitas interfaces de aquisição de imagem disponíveis no mercado.

Considerando a rapidez e eficiência na sua manipulação e implementação, além da inúmera quantidade de recursos disponíveis nesses sistemas, que cobrem grande maioria das aplicações na área de Mecatrônica esses sistemas de Visão Automatizada são cada vez mais utilizados nessa área.

A escolha de um software de visão adequado para uma dada aplicação devera seguir as recomendações seguintes:

- a) Escolha da Câmera.
- b) Fator de Escala do Hardware.
- c) Software de simples utilização.
- d) Abrangência e exatidão do algoritmo implementado.
- e) Desempenho do algoritmo implementado.
- f) Facilidade de integração com outros dispositivos.
- g) Custo-Beneficio envolvido.
- h) Parceiros e Integradores.
- i) Suporte Técnico.
- j) Crescimento e Estabilidade da Empresa

3.12.1 Sistema de Visão da National Instruments

A National Instruments pode ser considerada um dos líderes do mercado em visão de máquina e processamento de imagens. O software NI Vision está disponível em dois pacotes:

a) NI Vision Development Module: contém centenas de funções de visão para serem utilizadas com o LabVIEW da National Instruments, NI LabWindows/CVI, C/C++ ou Visual Basic para programar aplicações de inspeção, alinhamento, identificação e medição visual.

b) NI Vision Builder for Automated Inspection (AI): é um ambiente interativo para configurar, testar e distribuir aplicações de visão sem programação. Ambos os pacotes de software trabalham com todas as placas de aquisição de imagem da National Instruments e com o NI Compact Vision System,

3.12.2 Abrangência e Exatidão do Algoritmo

Existem muitas características a serem consideradas na escolha de um software de visão, entretanto uma das mais importantes é saber se a ferramenta de software pode medir de maneira correta e com exatidão as características do objeto em subpixels. Se o software não é confiável ou exato, não importa o quão rápido seu computador é ou quantos pixels sua câmera possui.

Deve-se saber que é muito mais fácil tornar rápido um código preciso do que tornar mais preciso um código que é rápido.

O Vision Development Module e o Vision Builder AI incluem centenas de funções de visão que são precisas e confiáveis. Todas essas funções possuem a exatidão em subpixels para interpolar localizações, distâncias e medições em até um décimo de pixel e um décimo de grau. As cinco áreas mais comuns para aplicações de visão de máquina juntamente com a descrição de algoritmos mais comuns, estão listadas a seguir:

a) **Melhoria de uma imagem:** Utilização de ferramentas de filtro para delimitar bordas, remover ruídos ou extrair informação de frequência com exatidão. A utilização de ferramentas de

calibração de imagem permitirá a remoção de erros de não-linearidade e perspectiva causados por distorções da lente ou posicionamento da câmera. As ferramentas para calibração de imagem também poderão ser utilizadas para aplicar unidades de engenharia nas medições, ou seja, transformar os valores em pixels em microns, milímetros ou milhas.

b) **Verificação de presença:** Esse é o tipo mais simples de inspeção visual, onde para verificar-se a presença de uma característica, pode-se utilizar qualquer ferramenta de cor, reconhecimento de padrão ou histograma. Uma verificação de presença sempre resulta em um valor sim/não ou passa/falha.

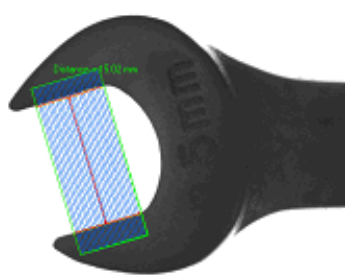
c) **Localização de padrões:** É importante no alinhamento de objetos ou na determinação do deslocamento dos objetos, servindo de padrão para todas as inspeções subsequentes. Para localizar padrões, essas ferramentas deverão permitir a detecção de: borda, escala de cinza, forma, geometria e padrões de cor. Essas ferramentas retornam a posição do objeto (X, Y) e o ângulo de rotação em até um décimo de pixel. A detecção de geometria é imune à sobreposição de objetos ou objetos que são alterados em escala.

d) **Medições:** É a razão mais comum na utilização de um sistema de visão, onde normalmente utilizam-se ferramentas com função para detecção de borda, análises de partícula ou geometria para medir distância, diâmetro, quantidade, ângulos e área. Caso seja efetuado o cálculo do total de número de células em um microscópio ou o ângulo entre duas bordas de um sistema de freio, essas ferramentas sempre retornam um número ao invés da localização ou um valor passa/falha.

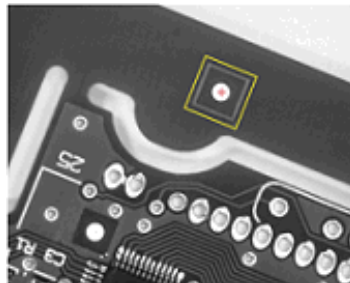
e) **Identificação de partes:** É importante para inspeção de conformidade, rastreabilidade e verificação. Métodos diretos de identificação incluem leitura de código de barras ou dados como DataMatrix e PDF 417. Novos métodos utilizam OCR “treinável” ou classificação de objetos. A Identificação de partes frequentemente resulta em texto ao invés de uma medição ou uma determinação passa/falha.

3.12.3 Análise de Desempenho do Algoritmo

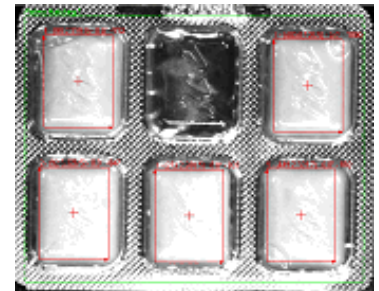
Enquanto a exatidão e a facilidade de uso frequentemente são os dois fatores mais importantes na escolha de um sistema de visão, a velocidade de execução é a terceira consideração. Não importa quantas centenas de algoritmos necessita-se para escolha ou quão rápido você pode construir uma aplicação com eles, se as ferramentas de inspeção são ineficientes e levam muito tempo para executar, muito do seu trabalho será perdido, Figura 3.23.



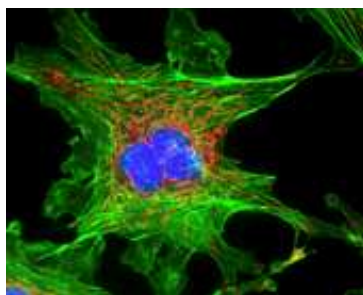
a) Medições



b) Melhoria de uma Imagem



c) Verificação de Presença



e) Localização de Padrões



d) Identificação de Partes

Figura 3.23: Principais aplicações em Visão de Máquinas em Automação.

O software de visão da NI é altamente otimizado para ter o melhor desempenho em qualquer possibilidade, resultando em um dos mais rápidos softwares do mercado, conforme mostra tabela 3.2 comparativa de características de tempo de processamento desse software em relação ao Líder do Mercado.

Características	NI	Líder do Mercado	No de Vezes Aumento Comparativo
Histograma	0.91	2.03	2.2
Transformadas Geométricas	3.1	10.3	3.3
Morfologia	1.8	5.9	3.3
OCR	3.3	5.9	1.8
Deteção de Geometrias	93.0	149.8	1.6
Classificação de Objetos	7.5	–	–

Tabela 3.2: Comparativo de Velocidade do Software de Visão NI em relação ao Líder do Mercado.

3.12.4 Integração com Outros Dispositivos

Em automação industrial, uma aplicação de visão está inserida como uma parte de um grande sistema de controle, podendo necessitar de atuadores para selecionar produtos; comunicar os resultados da inspeção para o controlador de um robô, CLP, ou Controlador Programável para automação; salvar imagens e dados em servidores de rede; ou comunicar os parâmetros e os resultados de inspeção para uma interface de usuário local ou remota. Frequentemente, para aplicações em imagens científicas, Deve-se integrar visão com estágios de movimento, sistemas de aquisição de dados, microscópios, óticos especializados e trigger avançado.

No caso da National Instruments, que pode ser considerada como um dos líderes do mercado em automação, controle industrial, aquisição de dados e controle de movimento, os produtos de visão da NI são projetados para trabalhar com esses e outros componentes comuns. Assim, no caso de uma aplicação que necessita-se comunicar com um CLP através de uma rede DeviceNet ou com um microscópio via barramento Serial, isso será facilmente implementado a partir de um Sistema de Visão da NI.

3.12.5 Custo-Benefício

Pacotes de software de visão possuem muitas variações. Muitos fornecem para clientes OEM (*Original Equipment Manufacturer*) separando suas bibliotecas e vendendo algoritmos individuais. Enquanto cada conjunto desses algoritmos individuais aparenta ter um custo reduzido, o pacote total de um sistema de visão possui um alto custo. Adicionado ao custo de uma licença para cada componente, a distribuição da aplicação se torna complicada e com um custo alto.

No caso do NI Vision Development Module, esse produto possui os algoritmos necessários para solucionar os desafios de visão mais complexos, assim você evita tempo de pesquisa, compra e manutenção de diversos pacotes de software. Assim, a distribuição das aplicações se torna de baixo custo, e através de uma única licença, pode-se distribuir uma versão executável que utiliza qualquer número de algoritmos.

3.12.6 Parceiros e Integradores

A National Instruments fornece software e hardware para processamento de imagem e visão de máquina. Pelo fato da NI não fornecer iluminação, câmeras e equipamentos óticos, a empresa trabalha em conjunto com outros especialistas que o fazem.

Através da última década, os produtos de visão da National Instruments têm auxiliado na solução de milhares de desafios em diversas aplicações de visão, desde inspeção em componentes automotivos a sistemas de pesquisa sobre câncer. Enquanto as ferramentas de visão da NI são projetadas para usuários finais, grandes aplicações podem necessitar da ajuda de um especialista em visão. Para auxiliar no desenvolvimento de sua aplicação, a NI trabalha com muitos parceiros que podem ajudar na seleção dos componentes corretos ou construir uma solução completa.

3.12.7 Suporte Técnico

Enquanto o software de visão da NI é projetado para ser de fácil utilização, é importante obter ajuda quando necessário. A NI fornece o software de visão diretamente para seus clientes, oferecendo também um suporte técnico direto aos seus produtos.

3.12.8 Crescimento e Estabilidade da Empresa

Um investimento num software de visão de máquina, deve-se levar em consideração utilizações futuras e necessidade de sempre mantê-lo funcionando. Existem diversas empresas pequenas e especializadas em visão de máquina onde, enquanto suas ferramentas podem funcionar para uma aplicação atual, quando você precisar atualizar uma estação de inspeção em cinco anos, você precisa saber se a empresa e o software ainda existirão e estarão evoluindo. No caso da NI, essa empresa tem um pesado investimento em Pesquisa e Desenvolvimento, e aos poucos tem se estabelecido como uma pioneira em software e hardware para visão, e com certeza que, nos próximos anos, essa empresa continuará a expandir e melhorar seus compromissos com visão de máquina e processamento de imagens.

3.13 Conclusão e Próximas Etapas

Esse capítulo descreveu alguns exemplos industriais implementados no Laboratório de Automação Integrada e Robótica da UNICAMP utilizando-se a ferramenta LabVIEWTM, que permitiram a validação experimental dos conceitos apresentados nessa dissertação. Através dos conceitos de prototipagem rápida em Mecatrônica, pode-se observar que em aplicações diferentes, a metodologia apresentada tem muitas semelhanças, facilitando a sua implementação rápida e com custo final reduzido. Contudo, cada sistema tem suas particularidades no que se refere ao local de operação, sistema de supervisão e controle e modelo de interface com o usuário. Essas particularidades inerentes a cada sistema são uma constante na utilização e implementação de técnicas de prototipagem rápida em mecatrônica, onde a elevada adaptabilidade do sistema proposto está condicionada aos requisitos de cada projeto.

4 Modelagem do Sistema de Acionamento e Controle

Nesse capítulo serão descritos o sistema de acionamento e transmissão, o projeto do sistema de acionamento e controle, o projeto do sistema de controladores PID, será ponderado sobre o gerador de trajetórias, e finalmente será abordada a conclusão e próximas etapas.

4.1 Sistema de acionamento e transmissão

Na Figura 4.1 é apresentado o sistema de acionamento do eixo de movimentação constituído pelo motor, drives de potência para acionamento, acoplamentos, mesa de trabalho, mecanismos de conversão de movimento como o fuso esférico, o sensor (encoder) para medição da posição e a ferramenta para o corte da peça.

As características desse modelo é estudado por (YEUNG et al. 2006), cujo trabalho projeta os modelos dinâmicos dos eixos de movimentação, utilizando o modelo linear do motor CC, o cálculo do momento de inércia com relação ao parafuso, a mesa, a peça de trabalho, além da influencia do torque estático do sistema. Esse trabalho utiliza os modelos de atritos representados por funções não lineares. O modelo inclui funções de ruído de medição, quantização dos sinais como os conversores de sinal digital-analógico, folga nos rolamentos e no parafuso dos eixos, e também contem os elementos de saturação para limite de corrente do atuador.

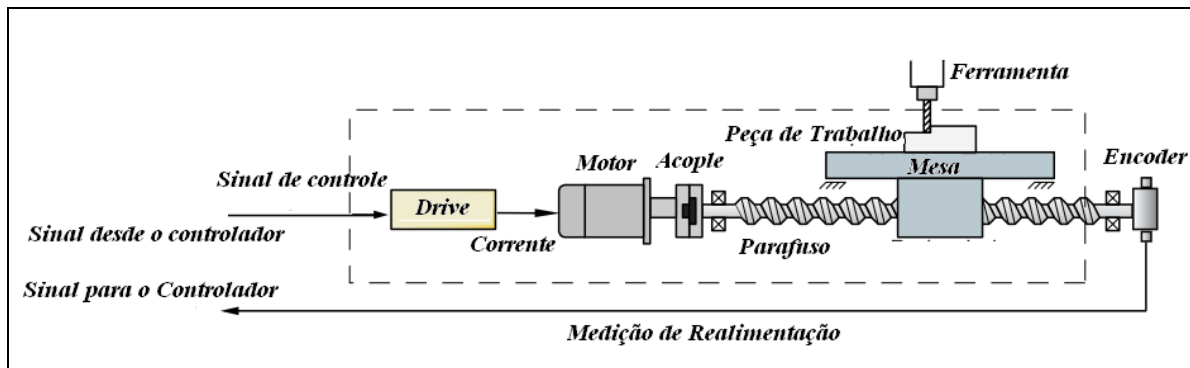


Figura 4.1: Estrutura do modelo do eixo de movimentação (YEUNG, 2006).

Outros projetos de sistemas de controle utilizam os modelos com a representação do drive de potência e o sistema atuador-amplificador, efeitos de atrito utilizando a função de Stribeck, modelos para folgas, ruído na medição e componentes de saturação como são apresentados nos trabalhos de Eckorkmaz e Wong(2007), Kamalzadeh et al. (2007) e Altintas (2000). O estudo dos efeitos de atritos em alguns mecanismos já tinha sido estudado como é apresentado no trabalho de Chen (1995), que realiza a modelagem dinâmica, detalhando as funções dos atritos no parafuso e nas guias mecânicas.

4.1.1 Projeto do Sistema de Acionamento e Controle

O sistema eletroeletrônico da máquina é configurado pela interface de potência e de controle, Figura 4.2. A interface de potência é composta por três interfaces para controle de corrente e tensão do sistema de acionamento, Figura 4.3, utilizando dispositivos para regular os picos de corrente e os giros de cada motor. A interface eletrônica de controle utiliza microcontroladores que realizam a regulação de potência do motor mediante o sinal de PWM. Esse dispositivo recebe os sinais de controle que são calculadas pelo computador, afim de gerar o comando de PWM. Esses sinais são enviados para a interface de potência fornecendo alimentação no sistema de acionamento. Os microcontroladores também recebem os sinais de cada encoder instalados nos três eixos independentemente, e envia os dados através da interface de aquisição de dados. A

energia é subministrada por uma fonte de alimentação CC de 50 V e 5 A, para controle dos três eixos de movimentação.

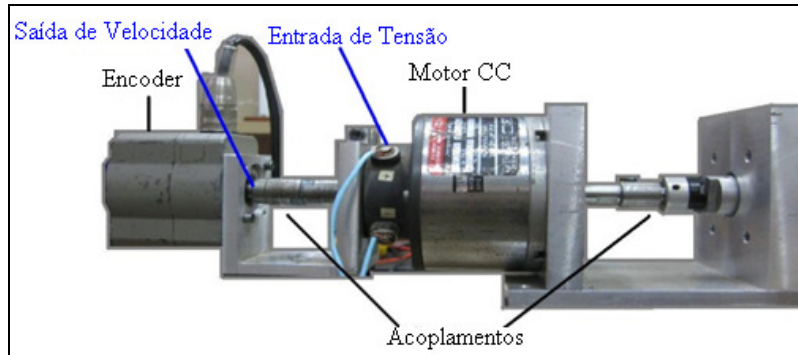


Figura 4.2: Motor DC e acoplamentos com encoder.

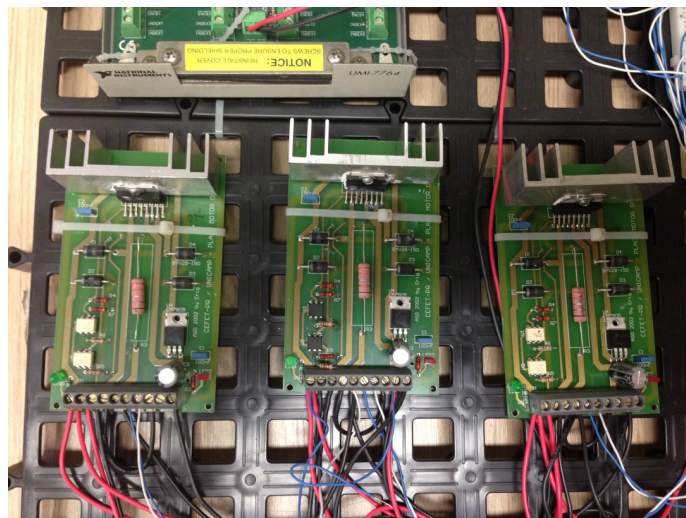


Figura 4.3: Interface de potência.

O sistema de Supervisão e Centro de Comando do dispositivo CNC está constituído pela interface de Motion da National Instruments configurado com programas de *software* realizados em LABVIEW.

4.2 Projeto do Sistema de controladores PID

O controle PID (Proporcional, Integral e Derivativo) é muito utilizado em aplicações industriais, onde diversos sistemas de máquinas e equipamentos mecatrônicos utilizam esse tipo de controlador por sua simplicidade para sintonizar, programar e implementar. Atualmente, esse tipo de controlador pode ser colocado em equipamentos do tipo analógico ou digital como microprocessadores, interfaces eletrônicas e drives por computador.

Um controlador PID é baseado na correção do erro do sistema através de uma ação proporcional, integral e derivativa. Sua expressão matemática é representada pela resposta $u(t)$ que é a saída do controlador e sua entrada $e(t)$ que é o erro do sistema, representado como $e(t) = y(t)_{referencia} - y_{atual}(t)$. A expressão matemática de um controlador PID é apresentada pela Equação (4.1), como função do sinal de controle no tempo, com K como o ganho proporcional, T_i o tempo de integração e T_d o tempo derivativo assim:

$$u(t) = K \left(e(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau + T_d \frac{de(t)}{dt} \right) \quad (4.1)$$

Nas máquinas-ferramenta CNC são utilizados métodos para calcular os PID aplicando algoritmos que permitam projetar os parâmetros a partir dos requerimentos desejados do sistema. Nesse sentido, os métodos analíticos como o Lugar Geométrico das Raízes ou os métodos por frequência podem ser escolhidos.

O sistema de controle é projetado com controladores baseados no modelo dinâmico, ou seja, que sua precisão e desempenho estão relacionados com os parâmetros modelados e a robustez das estratégias de controle. Para isso, os métodos empíricos de ajuste de controladores podem não satisfazer as necessidades de alto desempenho para velocidade e precisão do dispositivo CNC. Levando em consideração esse ponto, o método de lugar das raízes foi

escolhido para o projeto de controladores PID, devido a possibilidade de se projetar os parâmetros de controle baseado nos requerimentos da máquina.

Esse método permite ajustar os requerimentos de tempo de resposta e velocidade de cada eixo de movimentação com o erro desejado para a variável a controlar. A estabilidade do sistema é analisada pelo lugar geométrico das raízes e das margens de ganho e fase da função de transferência obtida com os controladores. Os parâmetros que o algoritmo de cálculo PID fornece são utilizados no simulador virtual da máquina e avaliados através da:

- Resposta em velocidade angular $\omega_j(t)$ e corrente $i_j(t)$ de cada eixo de movimentação.
- Tempo de resposta em cada variável de saída da máquina.
- Pico de corrente.
- E oscilações ou saturações presentes nas respostas dinâmicas.

4.2.1 Discussão do projeto de controladores PID

No projeto dos controladores PID por técnicas analíticas, mesmo que se tivessem os requerimentos desejados do sistema, é importante delimitar as restrições para o cálculo dos parâmetros. Os valores finais do PID podem satisfazer os requerimentos de estabilidade analítica e procedimento matemático. Entretanto, é possível que eles não satisfaçam o comportamento dinâmico real implícito no sistema dinâmico próprio da máquina. Assim, é importante a escolha dos requerimentos desejados ligados com as restrições da máquina em termos do tempo de resposta e erro estacionário. Pode-se precisar um erro em ultra precisão segundo o sistema, e o método de cálculo para os valores finais podem fornecer respostas válidas no cálculo do algoritmo matemático. No entanto, na aplicação real, esses parâmetros saturam o sistema e não respondem devidamente, para isso é fundamental testar o controlador no simulador para melhorar os parâmetros antes de utilizá-los na máquina real.

Embora o método do PID possa conseguir uma porcentagem de erro de contorno e de seguimento no sistema, não é possível atingir um erro em níveis altos de precisão, em termos de antecipar por si mesmo respondendo com maior velocidade. No caso de controladores PID, eles apresentam um grau de liberdade onde o seguimento da dinâmica se iguala ao seguimento da dinâmica de referência. Nesse sentido, deve ser considerado o controle preditivo para incluir o efeito antecipativo, melhorando o erro de seguimento em cada eixo de movimentação e uma diminuição do erro de contorno, aumentando a velocidade no sistema.

Como resposta a isso, é configurado o controlador PID utilizando uma estrutura de tipo *feedforward*. As técnicas utilizadas anteriormente são projetadas para o modelo dinâmico linear, conforme aos requerimentos da máquina. Contudo, a robustez é altamente afetada pelas mudanças no modelo, perturbações externas ou restrições. Os parâmetros calculados com modelos linearizados ou reduzidos são avaliados com a modelagem dinâmica completa no simulador.

Os requerimentos para o projeto são cuidadosamente fornecidos para evitar saturações da corrente, efeitos de temperatura nos mecanismos, oscilações e vibrações da máquina. Os parâmetros finais do projeto de cada controlador de corrente, velocidade e posição PI, são avaliados e ajustados no simulador virtual dessa dissertação. A fim de revisar o comportamento dinâmico que possui o controle com a modelagem não linear do sistema. Para implementação dos controladores em dispositivos eletrônicos digitais são sintetizados em polinômios RST como é explicado a continuação.

4.2.2 Estrutura de Controle na forma RST

Os sistemas de controle para as máquinas CNC utilizam dispositivos eletrônicos baseados em tecnologia digital. Devido a isso, as leis de controle foram calculadas em tempo discreto para ser implementados nesses dispositivos. A lei de controle RST apresenta uma relação de controle linear das leis de controle em tempo discreto, a qual facilita a programação dos controladores.

Essa lei de controle introduzida na malha do sistema de posição, e malhas internas da máquina de velocidade e corrente, possui a relação entre a saída $y(t)$, a variável de controle $u(t)$ e a referência $w(t)$. A implementação da equação RST com o operador em atraso q^{-1} está definida pela Equação (4.2) (CASEMIRO, 2005).

$$s(q^{-1})\Delta(q^{-1})u(t) = -R(q^{-1})u(t) + T(q^{-1})w(t) \quad (4.2)$$

Na Figura 4.4 está a representação da lei de controle RST.

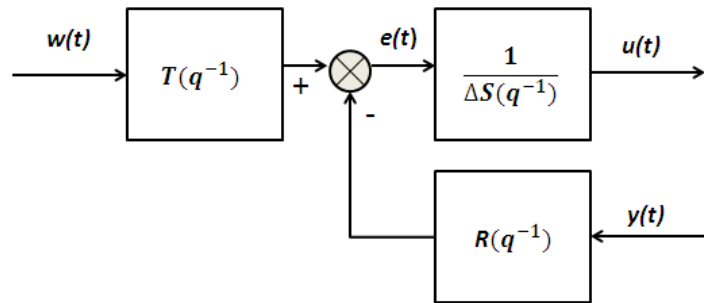


Figura 4.4: Estrutura de polinômios RST.

A estrutura em cascata aplicada para a regulação de posição e velocidade para a máquina CNC com os controladores RST é apresentada na Figura 4.5.

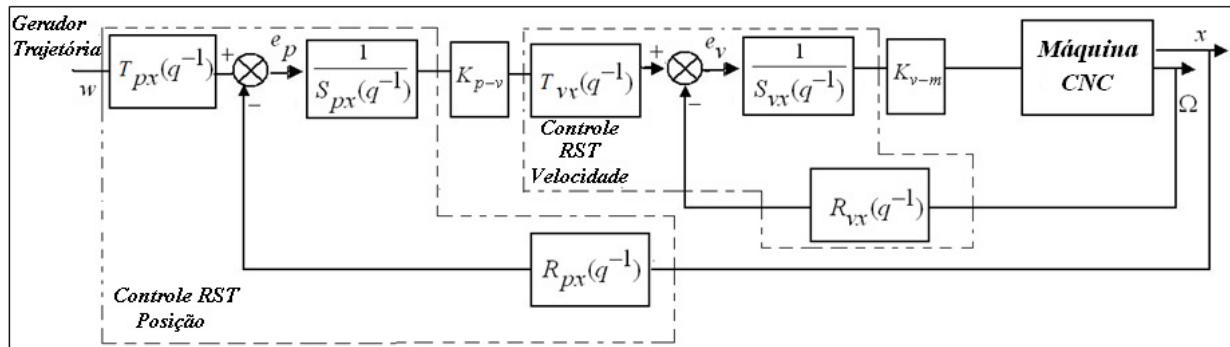
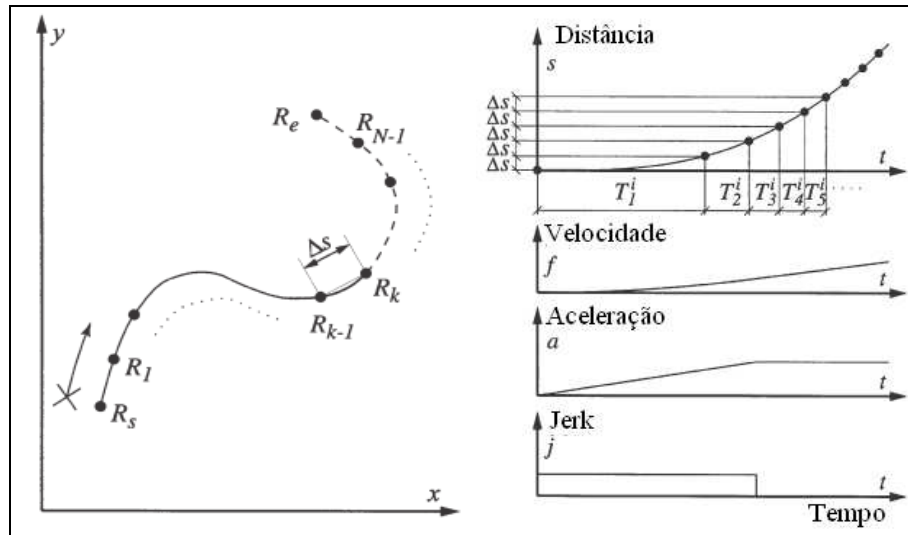


Figura 4.5: Estrutura em cascata com a síntese dos controladores RST.

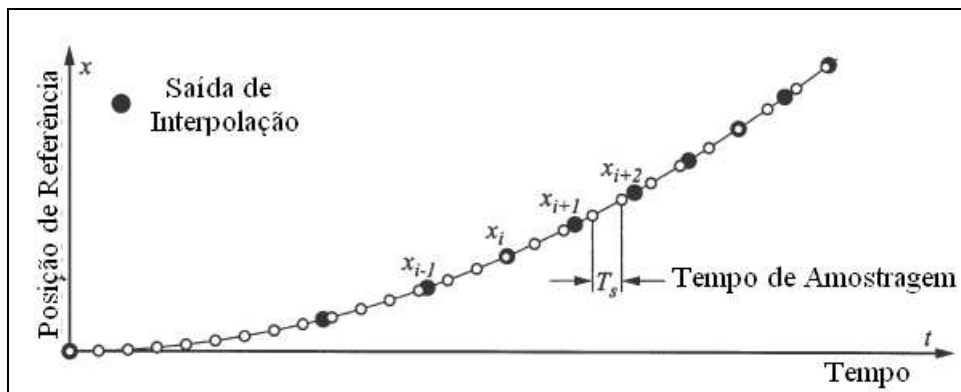
4.3 Gerador de Trajetórias

O gerador de trajetórias de dispositivos CNC, Figura 4.6, processa as informações de contorno da peça a ser produzida mediante a geração do caminho em coordenadas espaciais, e que utiliza técnicas de interpolação gera as trajetórias em coordenadas temporais. Finalmente são gerados os perfis de movimento, Figura 4.7, definidos pela posição, velocidade, aceleração e *jerk* do sistema. Os perfis de movimento representam as entradas de referência para o sistema de acionamento e controle dos eixos de movimentação da máquina-ferramenta. Os métodos mais empregados para a interpolação dos pontos são *Bezier*, *NURBS (Non-Uniform Rational B-Splines)* e *Splines*.

Os geradores de trajetória podem ser classificados em geradores *online* e *offline* (FREZZATTO et al., 2010). O gerador *online* executa as tarefas de interpolação e convergência dos dados ao mesmo tempo do funcionamento da máquina. Com isso, é necessário desenvolver algoritmos de alta velocidade de processamento para conseguir resultados satisfatórios. Ao contrario do gerador *offline*, esse calcula os perfis de movimento quando a máquina está sem funcionamento. No entanto, isso pode causar imprecisões nos parâmetros finais calculados, e que os resultados estejam afastados do comportamento real da máquina.



a. Geração de Caminho (coordenadas espaciais) b. Geração de Perfis (coordenadas temporais)



c) Geração de interpolação

Figura 4.6: Gerador de trajetórias (ERKORMAZ; ALTINTAS, 2001)

Os perfis de movimento (Figura 4.5) são uma representação das informações de velocidade com relação ao tempo e suas derivadas como aceleração e desaceleração, além do *jerk* como uma representação das variações na aceleração.

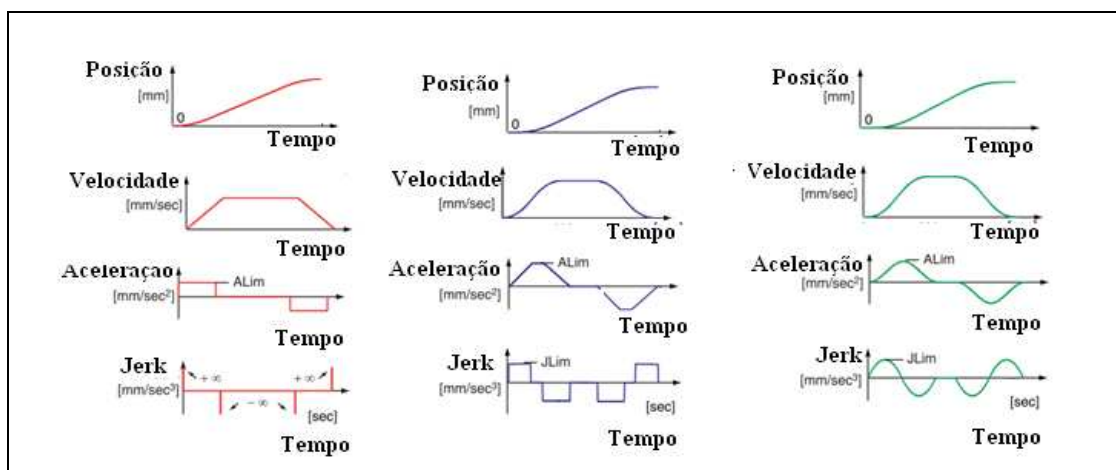


Figura 4.7: Perfis de Movimento para as máquinas CNC (ERKORKMAZ; ALTINTAS, 2006)

A geração de trajetória de referência desempenha o papel fundamental no controle computacional da máquina-dispositivo. As trajetórias geradas não apenas tem que descrever o passo desejado de dispositivo corretamente, porém, também tem que possuir perfis cinemáticos suaves a fim de manter alta precisão de rastreamento, e evitar excitar os modos naturais da estrutura ou do sistema de servo-controle (ERKORKMAZ; ALTINTAS, 2001, p. 1323).

Em termos de geração de trajetória, a tendência em novos métodos de otimização da trajetória da ferramenta são pesquisados, incluindo-se análise em superfícies de forma livre, erros geométricos sobre corte, otimização de energia em movimentos da ferramenta, desvios da superfície, adaptação da ferramenta, temperatura, e rugosidade superficial (HARIK et al., 2012).

O gerador de trajetórias é um modulo importante no desempenho dos dispositivos máquinas-ferramenta CNC, porque, uma vez dependendo dos algoritmos de construção dos perfis de movimento, esse conseguirá as entradas de referência adequadas para o sistema de controle, obtendo como resultado melhor desempenho no sistema. Devido a seu grau de importância, nota-se que os métodos para interpolação cada vez mais necessitam de algoritmos que consigam otimizar e melhorar os processos de análise de contorno, interpolação de pontos, e construção de perfis de movimento.

4.4 Conclusão e Proximas Etapas

Nesse capítulo foi realizado o estudo de motores de cc, seus aspectos construtivos abordando seu princípio de funcionamento e suas respectivas possibilidades de transmissão mecânica

Também foi apresentado nesse capítulo uma proposta da arquitetura de controle para regulação dos dispositivos máquinas-ferramenta CNC. Essa arquitetura possui o nível de referência, controle, compensação, diagnóstico e otimização. O nível de referência é constituído pela geração de trajetórias construindo os perfis de posição, velocidade e aceleração. O nível de controle é proposto com uma estrutura de cascata, a qual regula os parâmetros básicos da operação da máquina; as malhas dessa estrutura utilizam estratégias de controle clássico, preditivo e robusto sintetizados em polinômios de controle tipo RST para sua implementação. O nível de diagnóstico e otimização realiza a identificação dos parâmetros do sistema, além de utilizar métodos de otimização para a estimação e as trajetórias. O nível de compensação são os controladores de tipo *feedforward* que incluem o efeito antecipativo no sistema com o intuito de robustecer o sistema com relação ao efeito dos atritos e perturbações ajudando ao efeito dos controladores PID.

Utilizando os controladores em *feedforward* é possível melhorar as respostas para precisão das máquinas, devido aos efeitos na lei de controle de aceleração e velocidade nas malhas internas. Contudo, as sintonias desses controladores dependem da precisão dos modelos do sistema e sua robustez pode ser afetada pelas variações inerciais dos eixos ou mudanças da dinâmica.

A modelagem dinâmica de uma máquina CNC é fundamental para o projeto da arquitetura de controle com o objetivo de melhorar o desempenho em termos de precisão e velocidade. No entanto, o modelo deve ser complementado com os valores dos parâmetros da máquina CNC real,

para isso, é imprescindível desenvolver mecanismos de estimação e identificação de parâmetros que permitirão o diagnóstico completo e a simulação da máquina, além da possibilidade de detecção de algumas falhas, conseguindo assim o planejamento mais adequado do sistema de controle.

5 Implementação experimental

Nesse capítulo, são apresentadas a Descrição da Plataforma Experimental implementada ou seja o dispositivo máquina-ferramenta CNC utilizado para os experimentos e validação.

Ainda, nesse capítulo, são realizados os experimentos em duas etapas, a primeira, é a aplicação da estratégia de identificação e estimação de parâmetros da máquina-ferramenta CNC, utilizando-se controlador *Motion* da *National Instruments*, onde seram apresentadas, a trajetória linear com o PID de configuração inicial, a trajetória Circular com o PID de configuração inicial, a segunda etapa consiste nos experimentos do sistema de controle, utilizando os controladores por FPGA, fazendo-se o uso do kit de desenvolvimento DE2-115 utilizando FPGA da Altera, onde foi realizada a implementação em blocos DSP builder e a implementação com código VHDL, também será apresentada a implementação em ambiente Quartus II, e as conclusões.

Ainda, nesse capítulo, são apresentados a Validação de estratégias em máquina-ferramenta CNC para teste e validação das estratégias, apresentandas.

Continuando nesse capítulo, também, são apresentados os parâmetros obtidos através de identificação. E também, são apresentados a validação experimental do sistema de acionamento e controle, e finalmente tem-se as Conclusões.

5.1 Descrição da Plataforma Experimental implementada

Os experimentos para validação dos modelos, estratégia de identificação e sistema de controle foram realizados no sistema cartesiano máquina-ferramenta CNC fornecida pelo Laboratório de Automação e Robótica da Faculdade de Engenharia Mecânica da UNICAMP Figura 5.1 e Figura 5.2.



Figura 5.1: Bancada utilizada para implementação do experimento



Figura 5.2: Bancada, parte de potência, gerador de PWM e borneira de conexão

Na figura 5.3, pode-se observar o Conjunto com Juntas X,Y e Z, do dispositivo, utilizado.

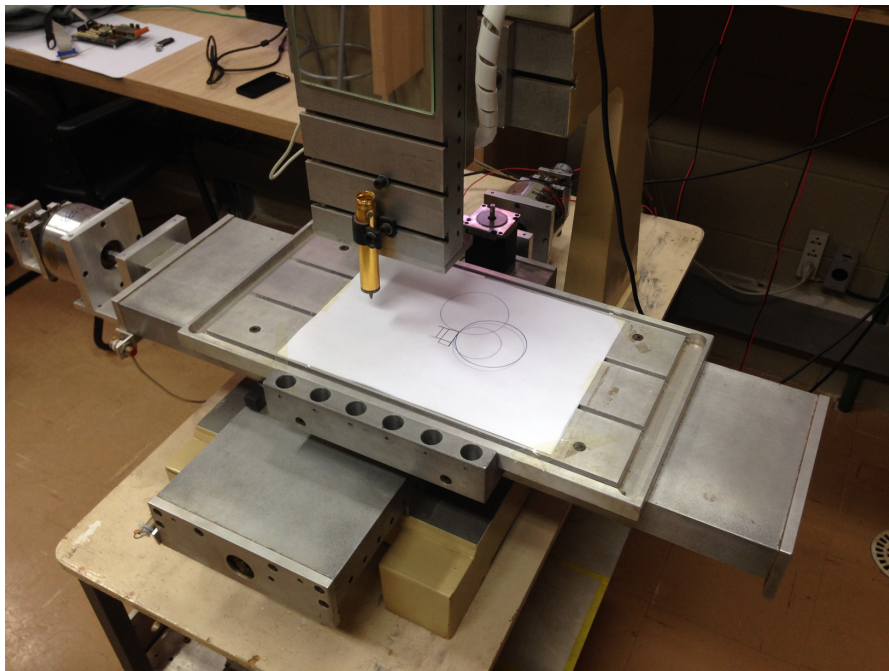


Figura 5.3: Conjunto com Juntas X,Y e Z

5.2 Interface de acionamento e controle utilizando plataforma National- Motion

Para implementação dos controladores é utilizada a interface de controle de movimentos *Motion* da *National Instruments* Figura 5.4. Nessa plataforma configura-se cada eixo de movimentação como mostra a Figura 5.8, e são introduzidos os dados da trajetória mediante um programa supervisor construído em *Labview* para gerar os perfis de movimento Figura 5.9. O *Motion* possibilita a programação dos controladores tipo PID para posição, regulando-se os parâmetros de corrente e velocidade

Como interface de controle potência foram implementados três módulos, que controlam os eixos X, Y e Z, e um sistema composto também de três geradores PWM, que tem por função a

geração dos pulsos necessários, para rotação dos três motores, sendo um para cada eixo, Figura 5.5 e Figura 5.6.

A interface de comunicação entre o módulo *Motion* e o sistema de controle e potência é feito através de uma borneira observada na Figura 5.6 e Figura 5.7.

Interface de controle de movimento de *Motion* da *National Instruments*



Figura 5.4: Controlador Placa Motion PCI 7344 utilizada no experimento

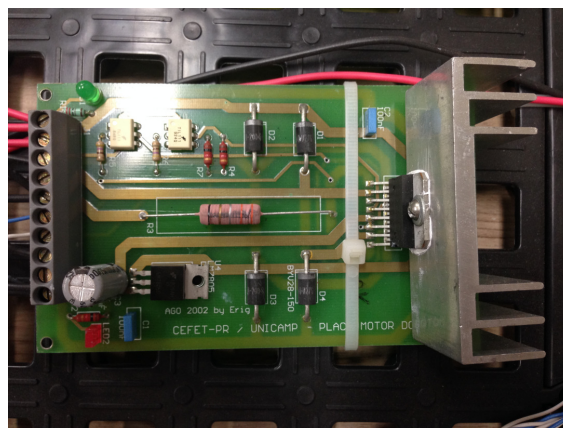


Figura 5.5: Placa Interface de potência

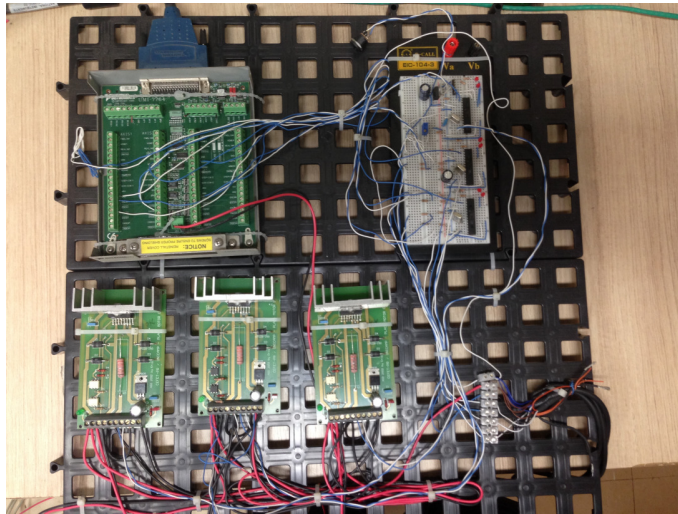


Figura 5.6: Conjunto Interface de potência, PWM e Borneira

Na figura 5.7, pode-se observar detalhe da borneira da placa motion PCI 7344, e na Figura 5.8 e Figura 5.10, observa-se o resultado de um desenho traçado pelo sistema utilizando-se controlador motion.

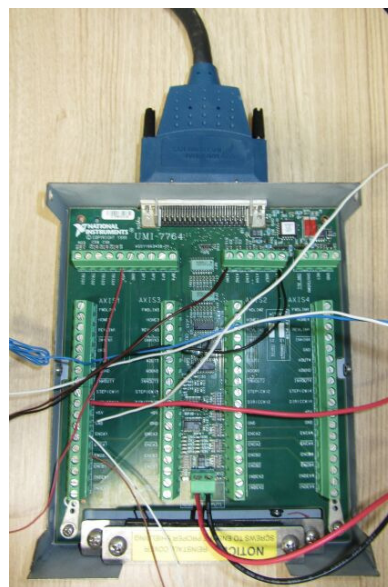


Figura 5.7: Borneira da Placa Motion.

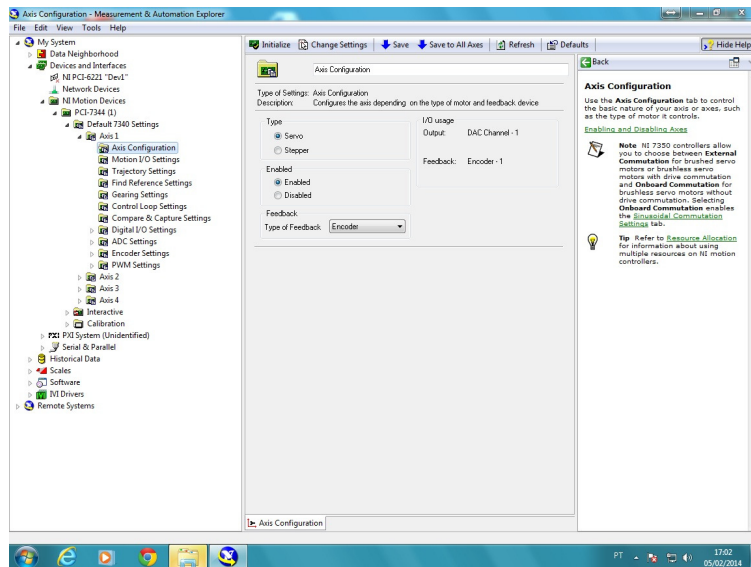


Figura 5.8: Tela do sistema motion para configuração de cada eixo independentemente.

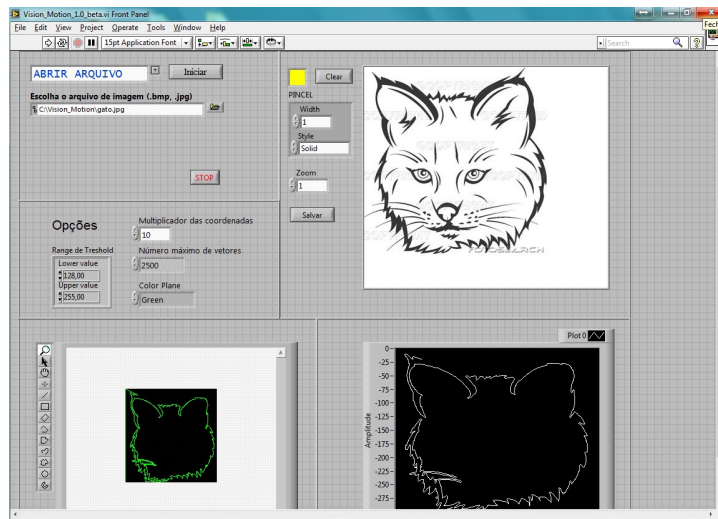


Figura 5.9: Tela do sistema supervisor utilizado juntamente com o controlador motion.

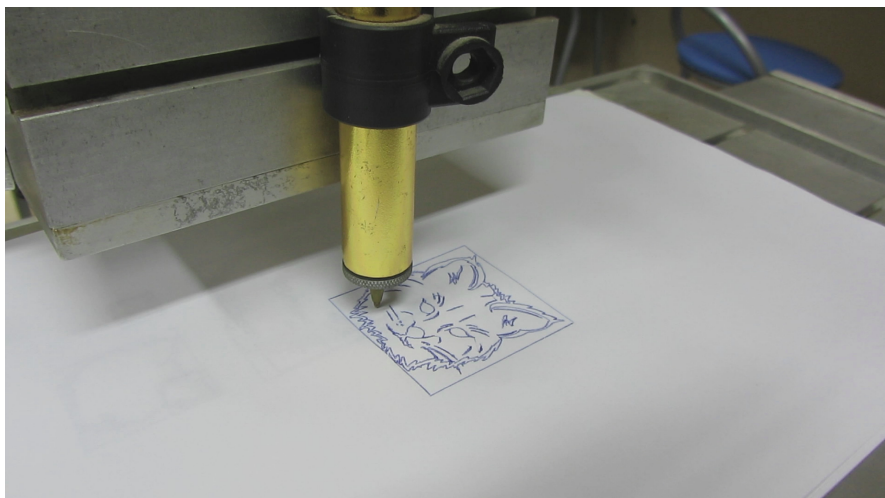


Figura 5.10: Resultado de um desenho traçado pelo sistema utilizando-se controlador motion.

5.3 Validação experimental

Nesse capítulo também é realizada a validação da plataforma de hardware associado, que permitirá o estudo do comportamento dinâmico baseado na modelagem e parâmetros estimados, e a simulação do projeto dos controladores para os dispositivos CNC em estudo.

A seguir tem-se a descrição de alguns experimentos realizados com a utilização do sistema cartesiano máquina-ferramenta CNC instalado no Laboratório de Automação e Robótica da Faculdade de Engenharia Mecânica da UNICAMP.

Para validação da plataforma de hardware, foram adotados dois experimentos, divididos em duas partes, como segue:

Parte 1 – Utilização Plataforma NI-Motion (ajuste parâmetros PID) (X-Y)

Trajectoria Linear (velocidades baixa, média e alta)

Trajectoria Circular (velocidades baixa, média e alta) – Raio Maior e Menor

Parte 2 – Sistema embarcado com FPGA (Z – X)

Trajectoria Linear com parâmetros PID – forma RST

Trajectoria Helicoidal com parâmetros PID – forma RST

Os resultados aqui descritos foram obtidos em conjunto, utilizando-se a mesma plataforma do trabalho de (LIZ KATHERINE RINCON ARDILA, 2013) utilizada neste trabalho

5.3.1 Trajetória linear

Para a obtenção e validação dos resultados, inicialmente foi implementada a plataforma NI Motion, onde abaixo, são mostrados os Elementos Básicos do sistema de Controle (Blocos), conforme descrito na Figura 5.11

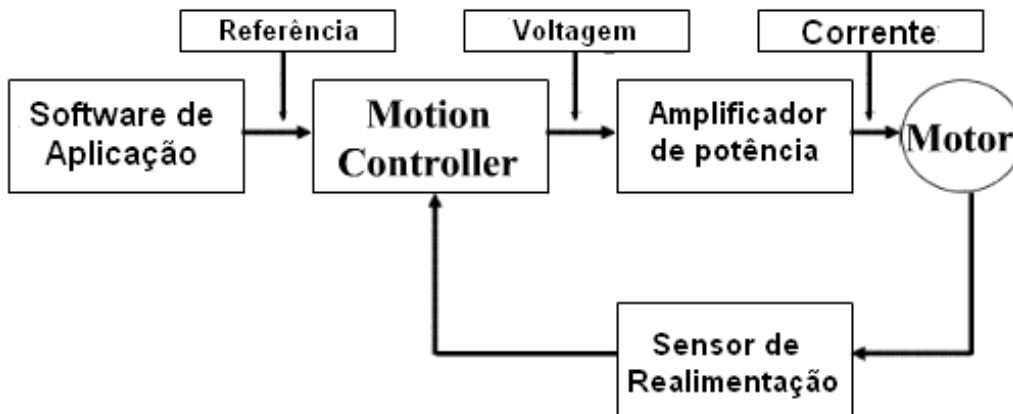


Figura 5.11: Elementos Básicos do sistema de Controle (Blocos).

Os eixos da máquina foram configurados inicialmente com um controlador PID de posição com os valores de $K_p=50$, $K_i=50$ e $K_d=10$. O controlador PID está configurado igual para os três eixos de movimento. Esse controlador foi ajustado diretamente na máquina com procedimentos empíricos e foi o controlador inicial que já possui o sistema para as operações de desenho e usinagem de peças. O comportamento desse sistema é apresentado na Figura 5.12 para o eixo X, Figura 5.13 para o eixo Y e Figura 5.14 para o eixo Z no seguimento da trajetória linear.

Os erros de seguimento para a trajetória linear com o controlador PID são apresentados na Tabela 5.1, para os três eixos de movimento. Os resultados desse experimento apresentam maior erro nas baixas e altas velocidades, e na média velocidade os eixos trabalham com menor erro. Quando o comprimento é maior e o eixo está em alta velocidade o erro se incrementa mais que nos menores comprimentos.

Tabela 5.1: Erro de Seguimento em trajetória Linear utilizando controle PID.

<i>Posição</i>	<i>Velocidade</i>	<i>Eixo X</i>	<i>Eixo Y</i>	<i>Eixo Z</i>
<i>Baixa</i> <i>5000 pulsos</i> <i>24.41mm</i>	Baixa	1.2259	0.3840	0.6167
	Média	0.7251	0.8103	3.9347
	Alta	1.8658	2.8225	3.4798
<i>Alta</i> <i>50000 pulsos</i> <i>244.14mm</i>	Baixa	3.1773	3.7506	2.3712
	Média	2.1041	2.5306	1.5188
	Alta	16.3986	13.2961	11.1561

Análise das Respostas da máquina-ferramenta, trajetória linear:

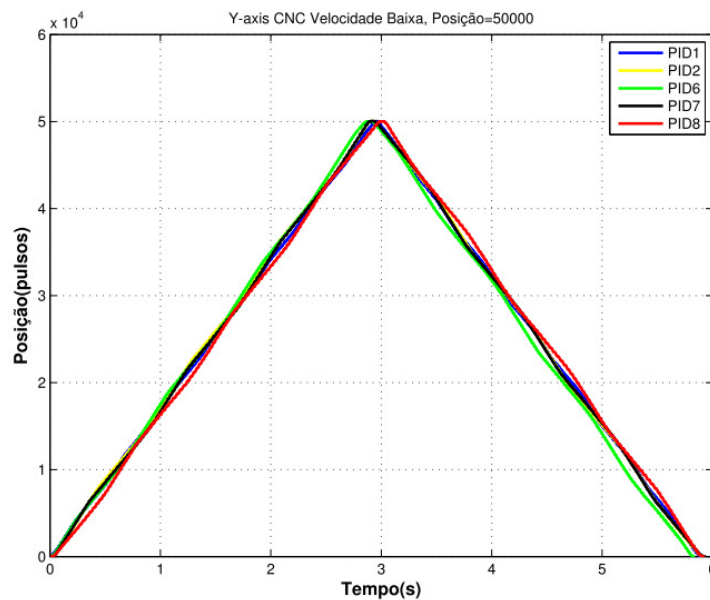


Figura 5.12: Baixa velocidade (Eixo X).

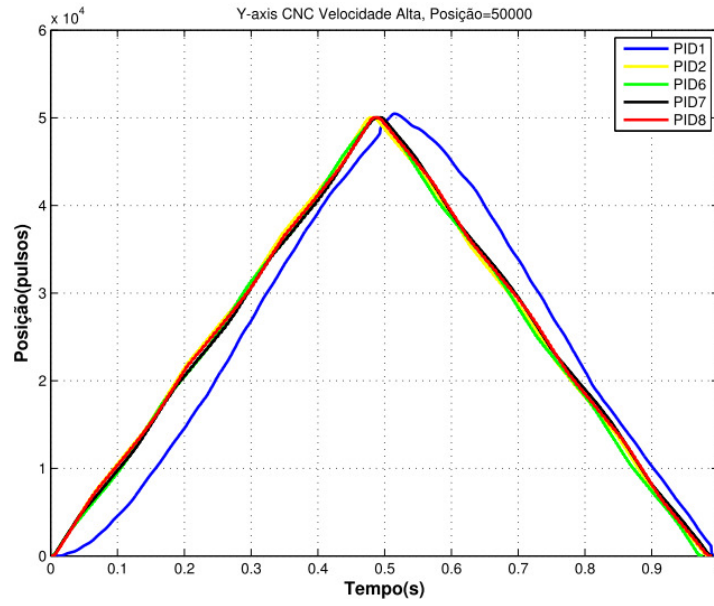


Figura 5.13: Alta Velocidade (Eixo Y).

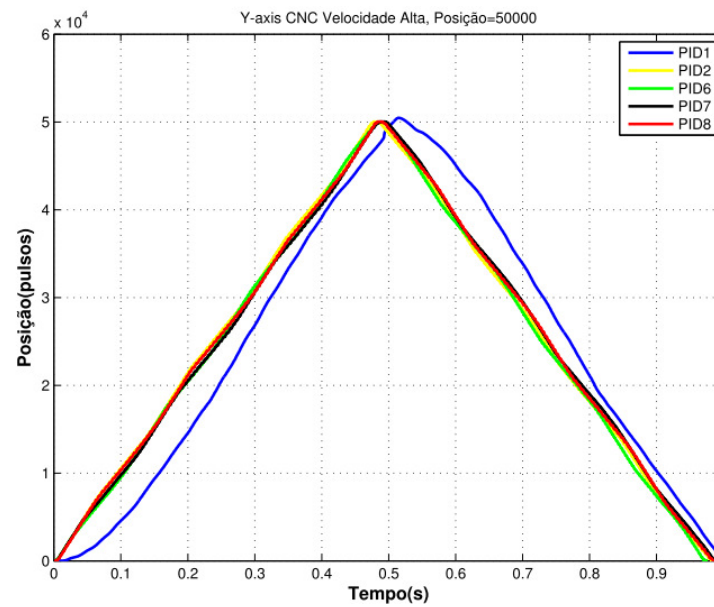


Figura 5.14: Baixa velocidade (Eixo Z).

Em baixas velocidades, o eixo X com o controlador PID conseguiu seguir a trajetória, com algumas oscilações no momento da mudança de giro de movimento, respondendo melhor para comprimentos maiores que para comprimentos de posição menor. Em velocidade média, o funcionamento do sistema apresenta melhor desempenho em termos de seguimento, oscilações e mudança de giro. Em velocidades altas, o eixo X não conseguiu seguir diretamente o perfil de trajetória e com um maior erro no momento da mudança de giro de movimento.

O eixo Y apresenta maiores oscilações em baixa velocidade e menor comprimento. Em média velocidade o eixo movimenta-se com menor erro de seguimento e em altas velocidades apresenta um erro maior no momento da mudança de movimento.

No eixo Z em baixa e média velocidade, o sistema manteve o seguimento da trajetória linear com algumas oscilações em baixa velocidade e comprimento curto, e apresenta maior erro em altas velocidades e mudança de movimento.

Assim, o eixo Z que é um eixo de movimentação vertical, o controlador precisa ir contra à variação de peso que tem o próprio eixo quando ele se movimenta em descida ou subida. Com isso, o sistema exige maior torque no momento de subida, porém, maior corrente para atingir a posição na velocidade desejada.

5.3.1.2 Resultados do controlador para trajetória Linear

O resultado com o PID em velocidade baixa apresenta um comportamento com baixas oscilações.

Na análise para os eixos X, Y e Z, o comportamento dos eixos com velocidade baixa mantêm um seguimento sem muitas oscilações, porém, em velocidades altas os eixos começam a exigir maior corrente e a vibração e as oscilações nas trajetórias são refletidas no desenho final da peça.

5.3.2 Trajetória Circular

Para este experimento realizado utilizou-se, a implementações de trajetória circular, com o objetivo de validar o desempenho do controlador PID, para isso foi inserida na máquina a trajetória circular com velocidades baixa= V_B , velocidade média= V_M e velocidade alta= V_A , e raio maior= R_M e raio menor= R_{MN} , onde os resultados obtidos, são apresentados nas **Erro! Fonte de referência não encontrada.**s 5.15,. 5.16, 5.17, 5.18, 5.19, e 5.20.

Os resultados de erro de contorno para uma trajetória circular, apresentaram maior erro em altas velocidades. O erro aproximadamente está ao redor de 80 pulsos ou 0.39mm para baixas velocidades e para altas ao redor de 150 pulsos ou 0.73mm.

Em altas velocidades, o sistema responde com maiores oscilações que em baixa velocidade. Para raios de circulo maior o erro também diminui. O erro maior acontece quando a máquina está com maior velocidade e para círculos de raio pequeno. Na Tabela 5.2, são apresentados os erros de contorno para o PID.

Tabela 5.2: Erro de Contorno em trajetória Circular com PID. Eixos X-Y.

<i>Raio do Círculo</i>	<i>Velocidade</i>	<i>Erro de Contorno MSE Trajetória Circular PID1</i>	<i>Máximo Erro de Contorno</i>
		<i>Eixos X-Y [mm]</i>	<i>Eixos X-Y[mm]</i>
Raio Menor 24.41 mm (5000pulsos)	Baixa	0.2184	0.5160
	Média	0.2140	0.6697
	Alta	0.3305	1.069
Raio Maior 97.67mm (20000pulsos)	Baixa	0.2143	0.4458
	Média	0.2458	0.6348
	Alta	0.3638	1.1279

Com o controlador PID o erro do sistema é maior em altas velocidades que em baixa velocidade. Quando o sistema realiza um círculo de raio pequeno, o erro cresce em velocidade alta com um valor máximo médio de 1mm para um raio de 24.4mm. Em velocidades baixas e raio pequeno o erro máximo está em 0.51mm. Para um círculo de raio maior o erro é maior com relação ao círculo pequeno. Em velocidades altas o sistema tem um erro máximo de 1.1279mm e um erro médio de 0.36, em baixas velocidades diminui a 0.44 o erro máximo e um erro médio de 0.21.

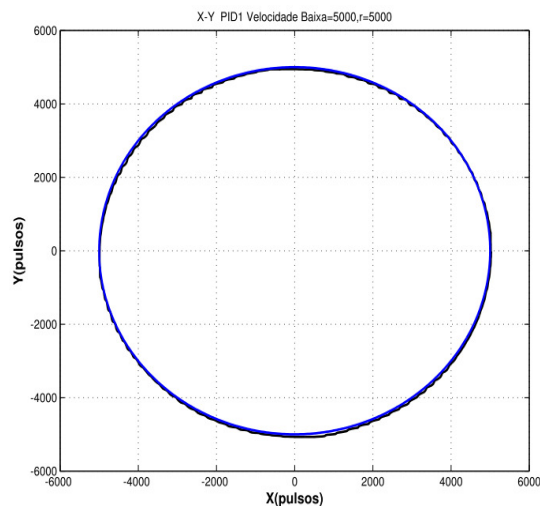


Figura 5.15: Trajetória: VB-Raio menor.

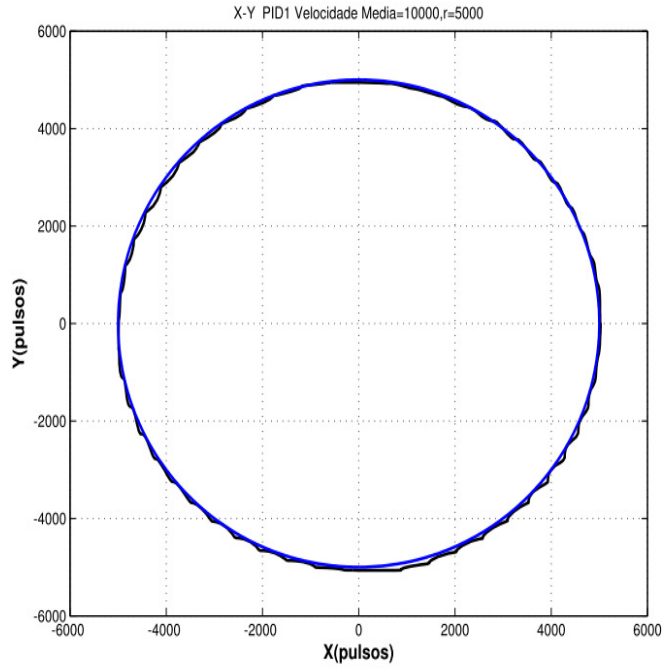


Figura 5.16: Trajetória: VM-Raio menor.

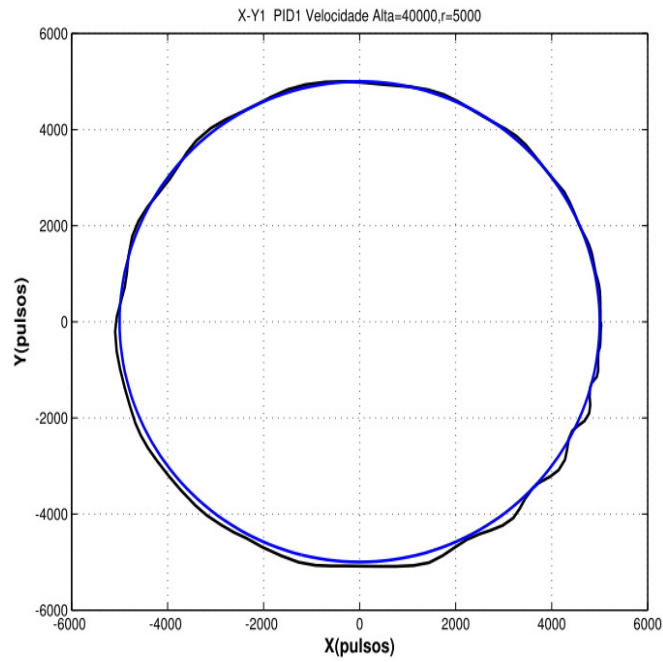


Figura 5.17: Trajetória: VA-Raio menor.

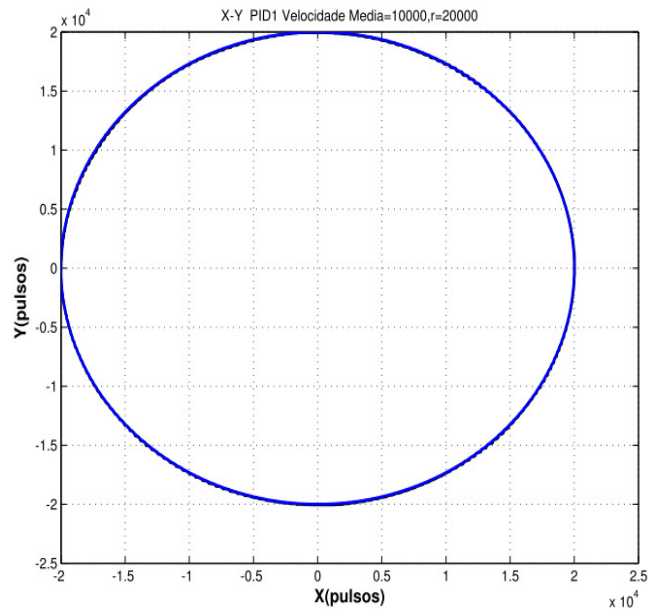


Figura 5.18: Trajetória VB-Raio maior.

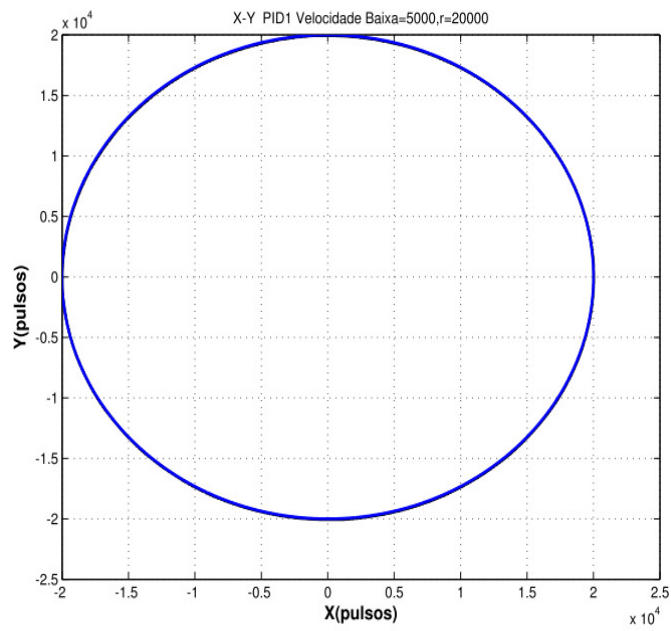


Figura 5.19: Trajetória VM-Raio maior.

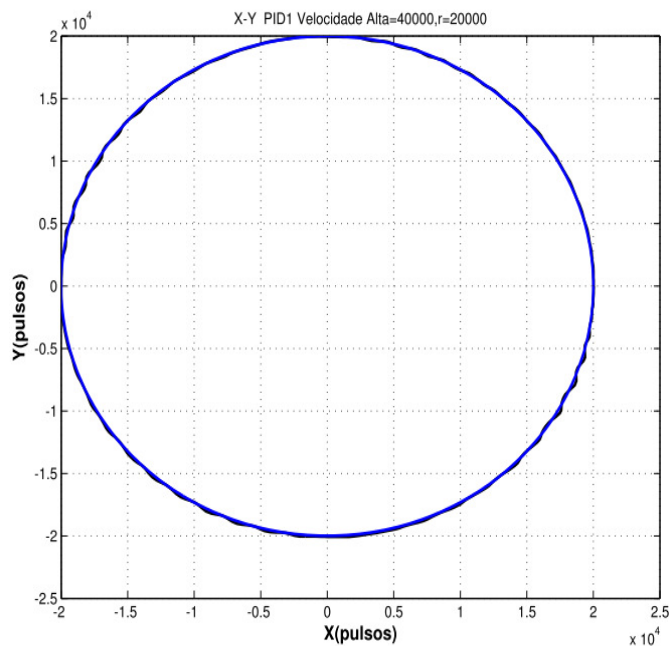


Figura 5.20: Trajetória VA-Raio maior

5.3.2.1 Resultados do controlador para trajetória Circular

Para validação do desempenho completo da máquina, foram realizados testes de seguimento circular para os eixos X-Y e Z-Y sendo realizados também, testes de trajetória circular utilizando para o eixo Y e para o eixo X, com o raio de 10000 pulsos e velocidades baixa, média e alta.

Com o propósito de analisar o sistema para maiores velocidades entre 10000 e maiores de 15000, também foi utilizando um raio menor de 20000 pulsos, para velocidades baixa, média e alta.

Os resultados, segundo o erro, nível de corrente e oscilações do sistema percebidas no seguimento de trajetórias circulares, apresentam maior erro em altas velocidades.

Para minimizar esse erro, foi incrementada a corrente no sistema, e ele conseguiu responder melhor na velocidade de 20000 pulsos, porém, para uma velocidade maior que 25000 pulsos, o sistema não conseguiu responder, devido ao incremento de corrente a qual as fontes do sistema de potência não fornecem. O movimento vertical do Eixo Z apresenta uma alta influência pela

variação no momento de inércia e efeito do peso do próprio eixo para responder em baixas velocidades. A vibração e oscilações são apresentadas na mudança de movimento, e o erro de contorno é afetado significativamente no limite de sua velocidade máxima, o qual também é diferente segundo o sentido de movimento do eixo.

5.3.3 Implementação de interface de acionamento e controle em FPGA

Nesse capítulo será abordada a implementação em dispositivo físico, mostrado na Figura 5.21, fazendo-se o uso do kit de desenvolvimento DE2-115 utilizando FPGA da Altera Figura 5.22.

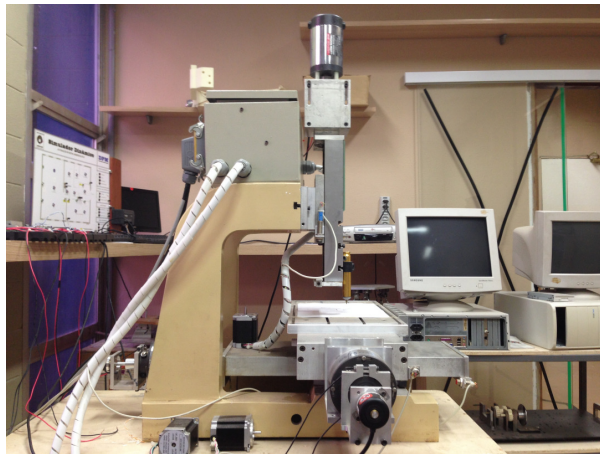


Figura 5.21: Implementação bancada com kit DE2-115 da Altera

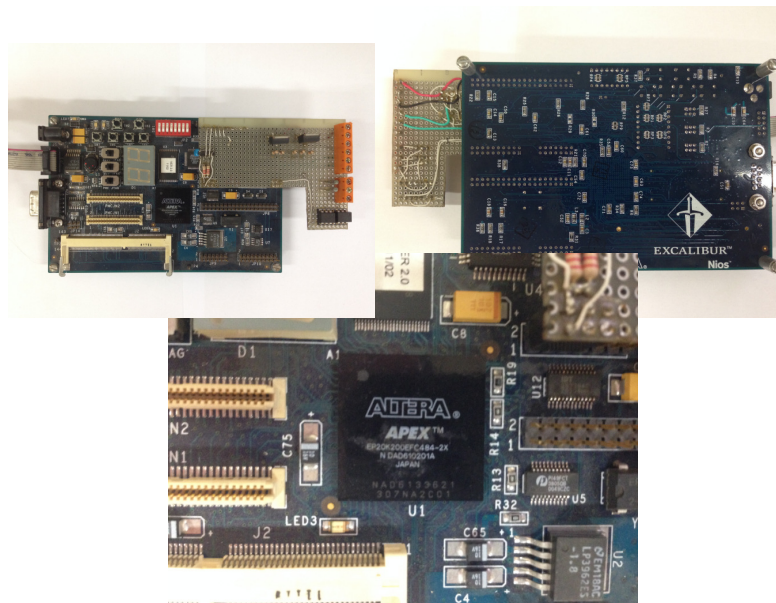


Figura 5.22: kit de desenvolvimento DE2-115 da Altera

Inicialmente, o procedimento priorizará as validações utilizando Matlab/Simulink em conjunto com a biblioteca DSP Builder, utilizando-se blocos e utilizando-se código VHDL.

O passo subsequente será a sua implementação no ambiente Quartus II para efetiva utilização do hardware escolhido.

Os resultados aqui descritos foram obtidos, utilizando-se a mesma plataforma do trabalho de (LUCIANO ANTONIO FREZZATTO SANTOS, 2011).utilizada neste trabalho

A comunicação do kit de desenvolvimento DE2-115 com o computador foi estabelecida através dos cabos de comunicação, conforme as Figuras 5.23 e 5.24

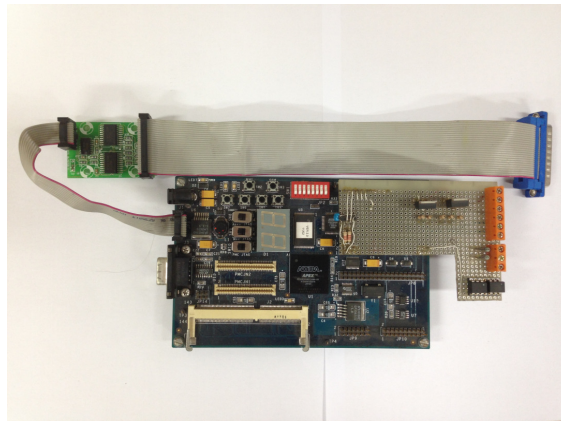


Figura 5.23: kit de desenvolvimento DE2-115 com cabo de comunicação.

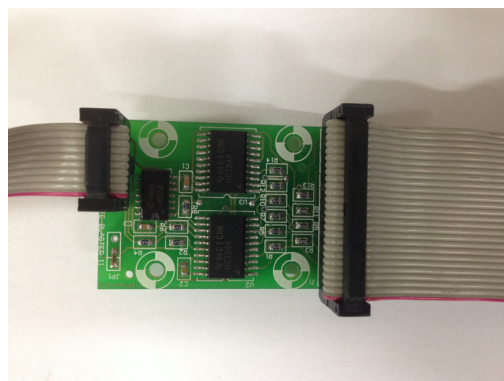


Figura 5.24: Detalhe do cabo de comunicação utilizado.

5.3.4 Blocos DSP Builder

A ferramenta DSP Builder disponibiliza uma biblioteca com blocos Simulink que permite utilizar a lógica discreta da FPGA dentro do ambiente Matlab podendo, inclusive, compilar o projeto implementado e carregá-lo na placa para incorporá-la à simulação, técnica conhecida como hardware in the loop (HIL) ou, aplicá-la para acionamento de um dispositivo externo. Porém, para realizar a compilação e carregamento do código é necessária a aquisição de uma licença específica.

Quando se trabalha com lógica discreta em sistemas de controle existe a restrição dos coeficientes estarem compreendidos entre -1 e 1. Assim, faz-se necessária a normalização de todos os valores para que fiquem dentro dessa faixa, caso contrário pode ocorrer overflow o que implica que o comportamento do sistema não esteja de acordo com o esperado.

Na Figura 5.25 é representada uma implementação de controlador utilizando os blocos DSP Builder sendo que o numero de coeficientes de cada polinômio (R, S e T) varia conforme o controlador desenvolvido.

Os blocos de entrada não-sintetizáveis mostrados na figura 5.30, fazem a conversão do sinal contínuo em Simulink para um sinal discreto que possa ser interpretado na FPGA e o bloco de saída não-sintetizável reconverte o sinal discreto em contínuo. Esses blocos permitem também especificar a quantidade de bits a ser utilizada para representação interna dos sinais, o que deve corresponder à quantidade de bits empregada na arquitetura dos blocos internos do controlador. Para o escopo desse estudo utilizou-se 32 bits para representação de todos os valores.

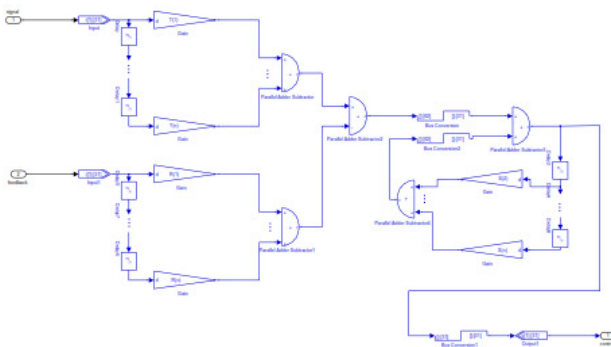


Figura 5.25: Bloco de controle utilizando blocos DSP Builder.

No caso das trajetórias de referência ocorre a mesma restrição dos coeficientes do controlador, logo essas também devem ser normalizadas. Além de que essa normalização deve ser posteriormente compensada na saída do controlador. O fator de normalização é variável de trajetória para trajetória, em decorrência disto, sempre que a trajetória de referência é alterada, deve-se recalcular esse fator. No presente estudo aplicou-se um método de aproximações sucessivas para determinação do fator que proporcionasse os melhores resultados na saída.

Após realizados os ajustes necessários foram efetivadas as simulações para ambas as trajetórias descritas anteriormente fazendo uso dos dois controladores desenvolvidos. Nas Figuras 5.26 e 5.27 são apontados os resultados para o controlador por alocação de pólos. Enquanto nas Figuras 5.28 e 5.29 são exibidos os resultados para o controlador preditivo, nesse caso somente foram avaliados os perfis de entrada e saída e o erro cometido ao longo da trajetória.

Pelos resultados obtidos, pode-se afirmar que ambos os controladores apresentaram desempenhos semelhantes visto que o erro cometido por eles é da mesma ordem de grandeza.

Ainda, cabe ressaltar que o processo de normalização dos sinais de referência e dos coeficientes dos controladores não ocasionou maiores impactos sobre o resultado final, já que a magnitude do erro cometido ficou próxima a obtida via simulação pura.

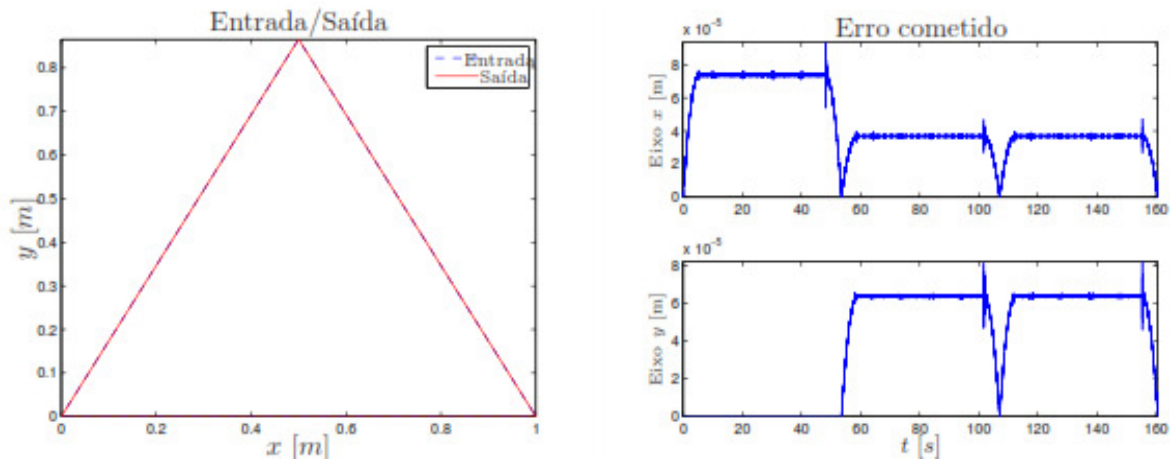


Figura 5.26: Comparação entre entrada e saída do sistema para trajetória triangular utilizando controlador por alocação de pólos implementado em DSP Builder.

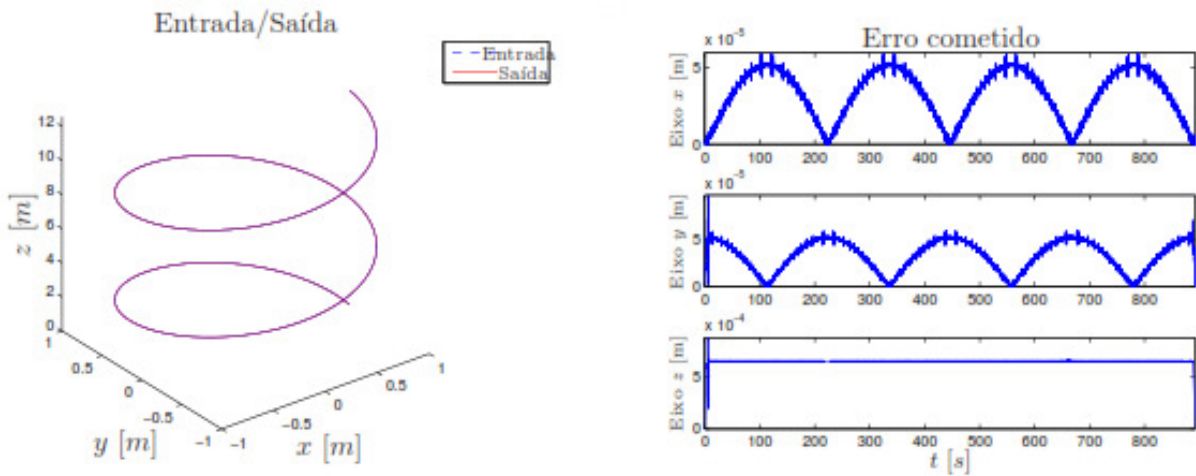


Figura 5.27: Comparação entre entrada e saída do sistema para trajetória helicoidal utilizando controlador por alocação de pólos implementado em DSP Builder.

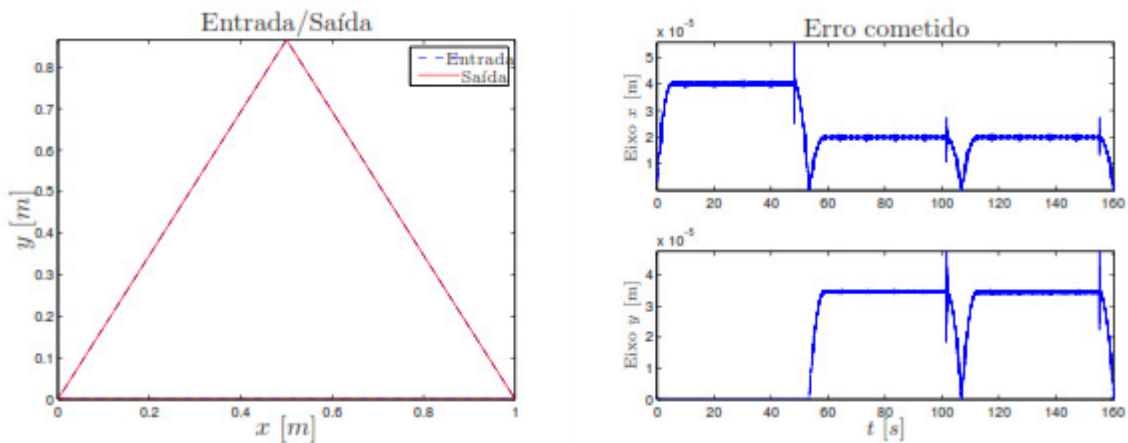


Figura 5.28: Comparação entre entrada e saída do sistema para trajetória triangular utilizando controlador preditivo em DSP Builder.

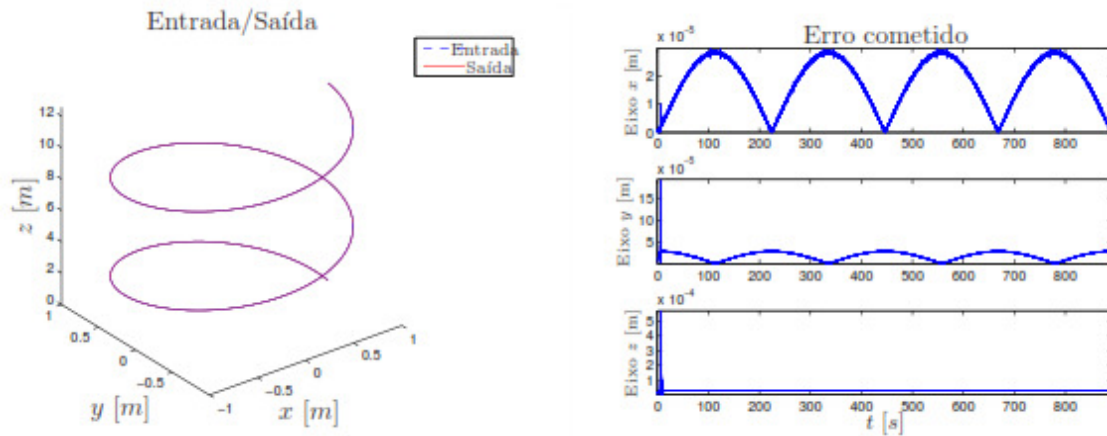


Figura 5.29: Comparação entre entrada e saída do sistema para trajetória helicoidal utilizando controlador preditivo em DSP Builder.

Em decorrência da necessidade de aquisição de uma licença para compilar os blocos de controle desenvolvidos e carregá-los na FPGA, adotou-se a linguagem VHDL para implementação dos controladores, pois essa independe de licença e pode ser aplicada em qualquer placa FPGA disponível.

5.3.5 Código VHDL

Para embarcar o código VHDL no ambiente Simulink se fez uso do bloco de importação de código VHDL disponível na ferramenta DSP Builder, como é mostrado na Figura 5.30.



Figura 5.30: Bloco de controle utilizando código VHDL.

Diferentemente do que ocorre nos blocos DSP Builder, que trabalham com ponto flutuante, o código VHDL faz uso de valores inteiros em sua lógica. Portanto, torna-se indispensável a conversão dos valores em ponto flutuante, determinados anteriormente, para valores inteiros por meio da multiplicação por uma constante, que depende da quantidade de bits utilizada para representação, com posterior arredondamento do resultado. Em contrapartida, há uma perda de precisão na representação do valor final já que números muito próximos podem ser mapeados para o mesmo valor inteiro após o processo.

A partir dessa consideração determinou-se os ajustes dos coeficientes dos controladores e das trajetórias de referência para que seus valores fossem representados como inteiros de 32 bits, com isso, foram rodadas simulações com ambos os blocos de controle para validação do algoritmo implementado e a avaliação dos resultados obtidos. Nas Figuras 5.31 e 5.32 são retratados os resultados para ambas as trajetórias utilizando o controlador por alocação de pólos. Nas Figuras 5.33 e 5.34 são mostrados os resultados para o controlador preditivo.

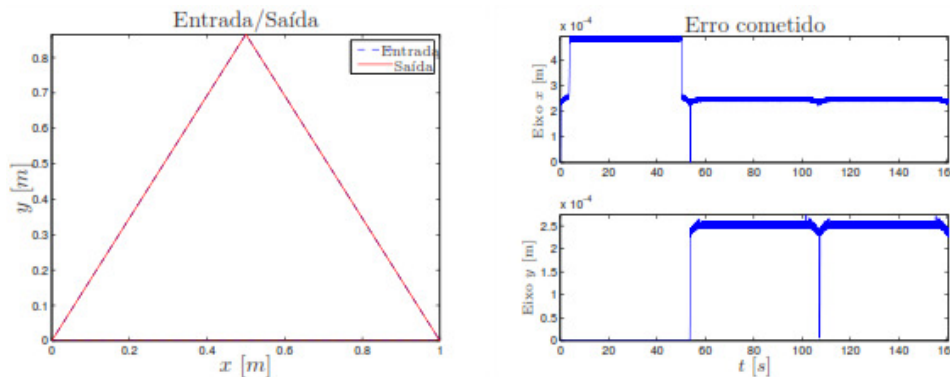


Figura 5.31: Comparação entre entrada e saída do sistema para trajetória triangular utilizando controlador por alocação de pólos implementado em VHDL.

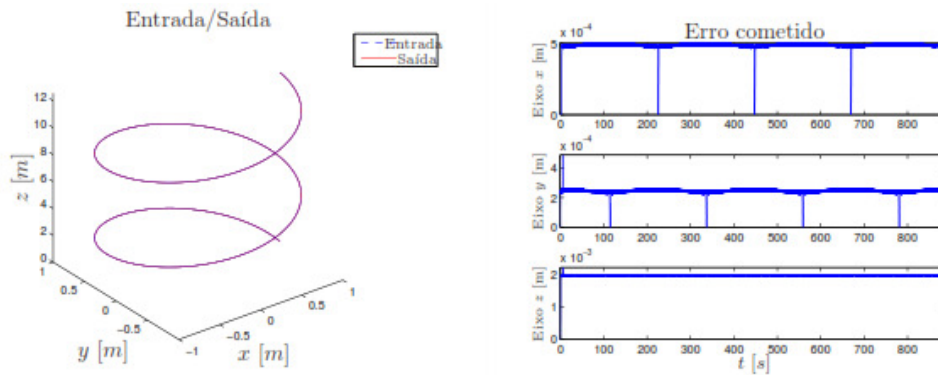


Figura 5.32: Comparação entre entrada e saída do sistema para trajetória helicoidal utilizando controlador por alocação de pólos implementado em VHDL.

Ao analisar os resultados mostrados nas Figuras 5.33 e 5.34 pode-se afirmar que esses estão aquém dos resultados obtidos ao utilizar blocos DSP Builder. Esse fato decorre do processo de transformação de número em ponto flutuante para número inteiro, que reduz a precisão da representação e, por consequência, aumenta o erro cometido no traçado da trajetória.

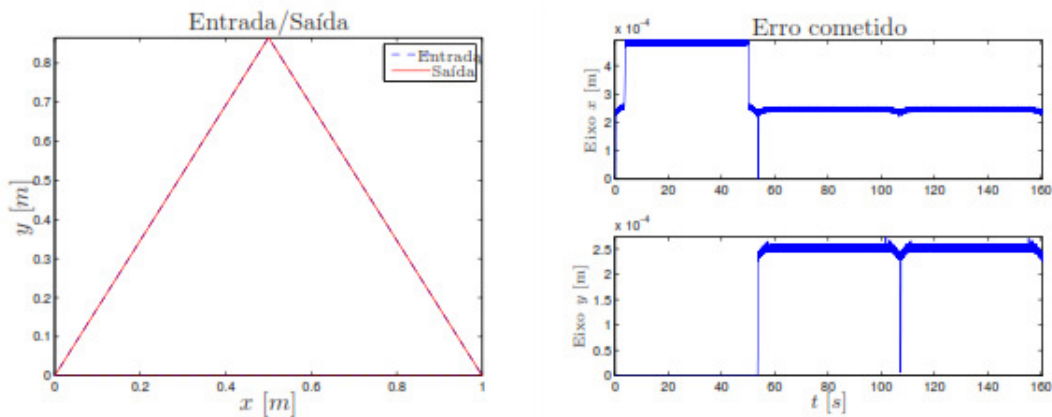


Figura 5.33: Comparação entre entrada e saída do sistema para trajetória triangular utilizando controlador preditivo implementado em VHDL.

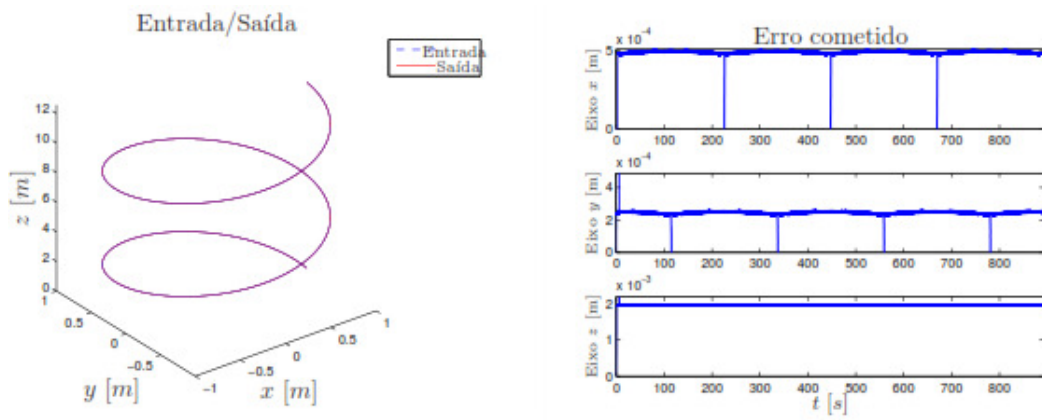


Figura 5.34: Comparação entre entrada e saída do sistema para trajetória helicoidal utilizando controlador preditivo implementado em VHDL.

5.3.6 Ambiente Quartus II

Com a finalidade de testar a implementação do controlador desenvolvido anteriormente na placa FPGA foi construído o ambiente mostrado na Figura 5.35, em que podem ser observados os blocos de controle, de leitura do encoder, de memória, de driver do PWM e de geração de PWM.

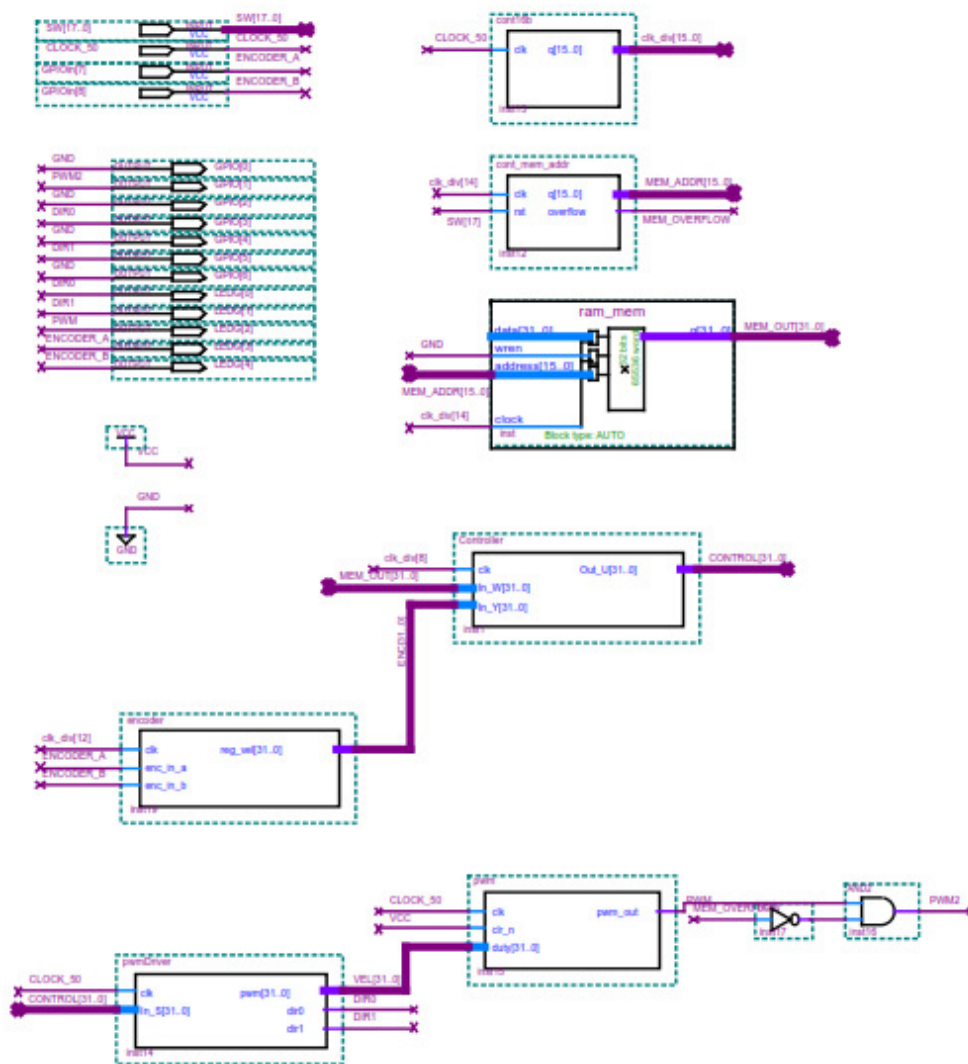


Figura 5.35: Esquemático em Quartus II para teste de controlador desenvolvido.

No bloco de memória é armazenada a trajetória de referência para o sistema, tornando a aplicação standalone, já o driver de PWM é responsável por determinar o sentido de rotação do

motor a partir de um valor dado em sua entrada e, por sua vez, o bloco de leitura do encoder armazena a quantidade de pulsos gerada pelo encoder em um contador que é incrementado ou decrementado dependendo do sentido de rotação do motor. Por fim, o bloco de geração de PWM é responsável por modificar o *duty-cycle* do trem de pulsos variando, assim, a velocidade de rotação do motor. Os códigos implementados de todos os blocos estão expressos no (Anexo A).

Além desses blocos, os sinais de direção e de PWM foram conectados á uma placa de potência, cujo esquemático encontra-se na Figura 5.36, para amplificá-los de forma a acionar o motor CC com a tensão e a corrente necessárias.

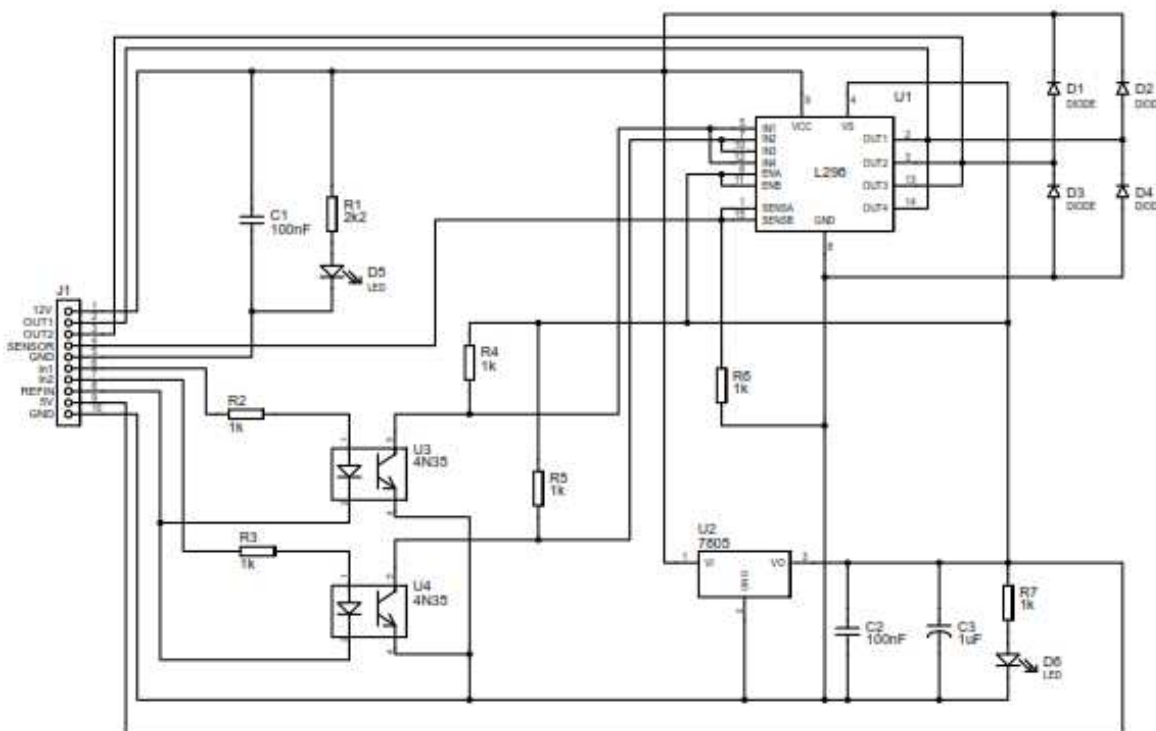


Figura 5.36: Placa de potência utilizada para acionamento do motor CC.

A montagem experimental para teste da metodologia desenvolvida procedeu-se por meio da aplicação dos componentes citados conectados ao motor CC para acionamento do dispositivo

No decurso da realização dos experimentos observou-se que a contagem dos pulsos do encoder não ocorria da forma esperada, pois havia ruído no sinal gerado. Por esse motivo, adotou-

se um fotoacoplador com a finalidade de isolar o sinal de entrada do sinal de saída e que reduzisse a influência do ruído, o que otimizou a precisão da contagem.

Pelo que foi exposto, pode-se afirmar que ambos os controladores exibiram desempenhos semelhantes, no entanto em decorrência do efeito antecipativo do controlador preditivo, o funcionamento do dispositivo cartesiano ocorreu de modo mais suave, sem trepidações e sem a necessidade de retorno da ferramenta para correção da trajetória executada. Frente a esse resultado e aos demais resultados apontados anteriormente, conclui-se que a classe de controladores preditivos mostra-se mais adequada para ser aplicada conjuntamente com dispositivos CNC quando acionados por motores CC.

5.4 Validação das estratégias Propostas em Máquina-Ferramenta CNC

A máquina-ferramenta CNC utilizada para a validação experimental dos modelos dinâmicos e do sistema de controle foi apresentada no capítulo 3, com as características mecânicas, eletroeletrônicas e de controle. Essa máquina é um sistema mecânico linear cartesiano constituído por três eixos de movimentação X, Y, Z. A máquina CNC utiliza a interface de *Motion* da National Instruments para programação dos perfis de movimento e sinais de controle para os eixos de movimentação.

Para estimação dos parâmetros é utilizado a configuração de *Motion* e *Labview* para as entradas de identificação, controle de posição e aquisição de dados. O processamento de dados é realizado *offline* com o sistema de estimação de parâmetros em MATLAB.

A validação dos controladores pode ser avaliada experimentalmente com os controladores PID de posição, devido a que *Motion* somente possui possibilidade de configurar os parâmetros para controladores tipo PID, sem flexibilidade de arquiteturas abertas de controle. Para implementação de outras estratégias avançadas de controle, o *Motion* restringe a programação. No entanto, a avaliação experimental do PID é suficiente para avaliar a modelagem, identificação

e simulador do sistema, com o que os outros controladores sejam avaliados através de um simulador.

5.4.1 Parâmetros obtidos através de Identificação

Os experimentos realizados para a identificação de parâmetros utilizam a estratégia proposta e aplicam a metodologia definida na Figura 5.37. As entradas no primeiro nível são de tensão para os motores. Os motores trabalham até 34.4V, no entanto, para a máquina são disponibilizados desde 1V até 12V, obtendo velocidades desde 100 rpm até 2500 rpm. No segundo nível de identificação, a entrada é uma trajetória programada diretamente no ambiente de *Motion*, fornecendo os parâmetros de velocidade e aceleração para gerar os perfis de movimento. Para aquisição dos dados, o sistema é configurado em um tempo de amostragem de $T_s=1\text{ms}$.

Os dados são processados por programas em MATLAB®. Os modelos de referência escolhidos estão definidos no Capítulo 4 de Identificação e Estimação de parâmetros segundo o nível da estratégia. Finalmente o processo de validação tem como objetivo comparar os modelos propostos e os parâmetros obtidos com a máquina CNC real. A validação utiliza os seguintes critérios:

- 1) Erro de Seguimento.
- 2) Erro de contorno.
- 3) Erro entre a medição e a simulação.

Depois da validação, se os parâmetros precisam ser ajustados, o processo de estimação é retornado para realizar novas iterações até conseguir o mínimo erro dos modelos e dos parâmetros. Nesse passo, é importante o ajuste das restrições e valores limite para o problema de otimização.

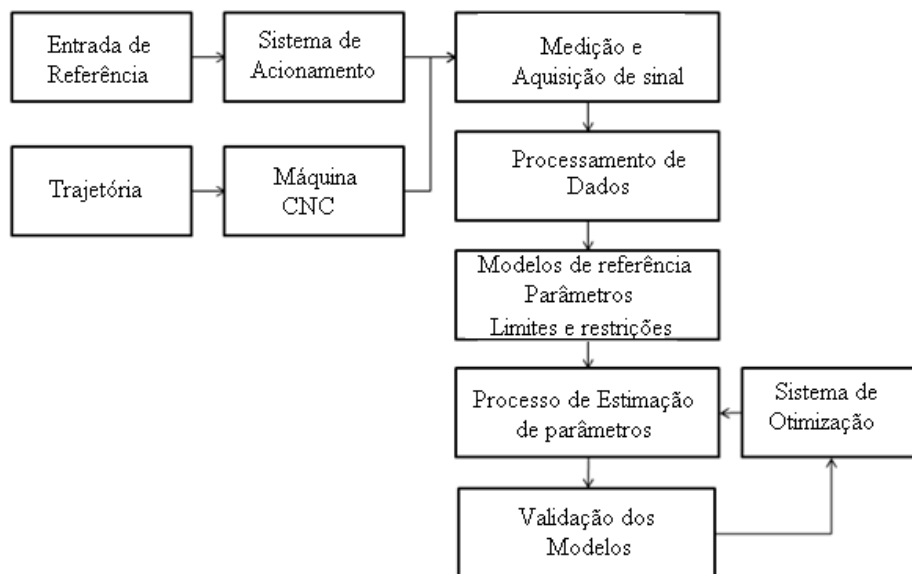


Figura 5.37: Metodologia para Identificação da Máquina-Ferramenta CNC.

5.4.2 Validação experimental do sistema de controle

O sistema de controle utiliza a interface de *Motion* da *National Instruments* para configurar as características elétricas do sistema de medição e dos drives de potência dos eixos da máquina, além de possibilitar a programação de controladores tipo PID. No entanto, esse possui restrições para implementar outro tipo de controladores. Porém, o projeto dos controladores PID são testados experimentalmente na máquina real, e os outros controladores foram validados na simulação, para trabalhos posteriores em implementação com outras tecnologias.

Os testes do sistema de controle são configurados para realizar uma trajetória linear para três posições (baixa, média, alta) e três tipos de velocidades (baixa, média, alta). A configuração de aceleração e desaceleração é uma constante nas diferentes configurações. Para teste de cada eixo, é utilizada essa trajetória linear de ida e retorno como é apresentada na Figura 5.38, com o objetivo de gerar movimento do sistema no sentido horário e anti-horário, passando por

velocidade zero e mudança de giro de movimento. Na Tabela 5.3, são apresentadas as características da trajetória linear configurada para os três eixos de movimento.

Tabela 5.3: Parâmetros de configuração para trajetória linear

<i>Saída</i>	<i>Eixo X</i>			<i>Eixo Y</i>			<i>Eixo Z</i>		
	<i>Baixa</i>	<i>Média</i>	<i>Alta</i>	<i>Baixa</i>	<i>Média</i>	<i>Alta</i>	<i>Baixa</i>	<i>Média</i>	<i>Alta</i>
<i>Velocidade</i>	5000	10000	40000	5000	10000	40000	5000	10000	18000
<i>Posição</i>	5000	20000	50000	5000	20000	50000	5000	20000	80000

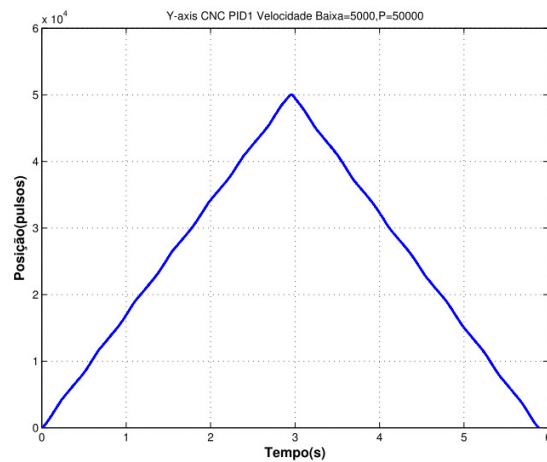


Figura 5.38: Trajetória linear para testes dos Controladores.

Para a análise da máquina utilizando movimentos que incluem dois eixos de movimento, é feito o experimento de uma trajetória circular como é apresentado na Figura 5.39, com o movimento do eixo X-Y, e outros testes com o eixo Y-Z. A trajetória circular é igualmente configurada em termos de três velocidades (baixa, média, alta) e dois valores de radio do círculo. Na Tabela 5.4, são apresentados os valores para a trajetória circular.

Tabela 5.4: Parâmetros de configuração para a trajetória Circular

<i>Saída</i>	<i>Eixo X</i>			<i>Eixo Y</i>			<i>Eixo Z</i>		
	<i>Baixa</i>	<i>Média</i>	<i>Alta</i>	<i>Baixa</i>	<i>Média</i>	<i>Alta</i>	<i>Baixa</i>	<i>Média</i>	<i>Alta</i>
<i>Velocidade</i> <i>e</i> <i>[pulsos/s]</i>	5000	10000	4000 0	5000	10000	4000 0	5000	10000	1800 0
<i>Raio</i> <i>[pulsos]</i>	5000		2000 0	5000		2000 0	5000		2000 0

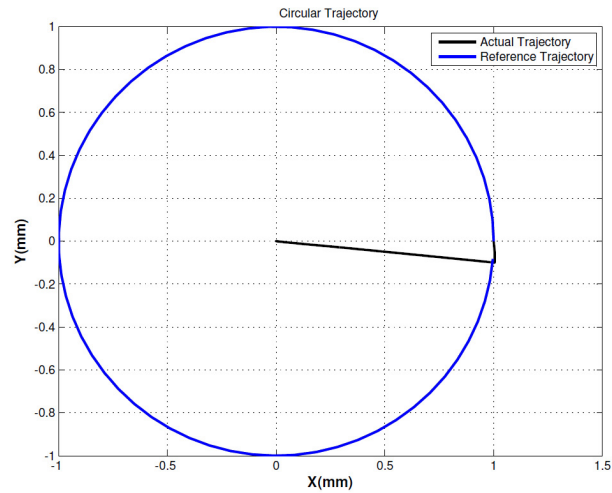


Figura 5.39: Trajetória Circular para testes dos Controladores.

5.5 Conclusões

Nesse capítulo a ênfase recaiu sobre os aspectos relativos a montagem experimental que teve como finalidade a validação das estratégias propostas.

A arquitetura de interface *National-Motion* possibilita a programação efetiva dos parâmetros do controle PID para cada eixo, no entanto, essa interface não permite configurar outras estruturas avançadas de controle, limitando o sistema de controle e a implementação completa da arquitetura proposta nessa dissertação de mestrado, o que seria considerado para uma futura proposta de programação em sistemas embarcados para dispositivos CNC.

Após realizados os experimentos, constatou-se que o dispositivo cartesiano CNC respondia como desejado traçando os perfis da forma que havia sido planejada ao utilizar-se a interface *National-Motion*.

Na validação do PID, é percebido que as vibrações do sistema são evidenciadas em baixas velocidades para o eixo X-Y, quando apresentam-se mudanças de movimento, isso como resposta aos efeitos de atrito e de mudança no torque dos atuadores. Com relação ao eixo Z a resposta em baixa velocidade é mais afetada pela variação inercial que nos outros eixos, devido a seu movimento vertical. Quando o sistema apresenta movimentos de subida e descida, o torque que necessita para movimentar-se é alto, e em baixas velocidades o controle precisa de maior ajuste.

Em movimentos verticais, o maior efeito é apresentado em baixas velocidades para trajetórias lineares, devido à alta variação inercial, e em altas velocidades em trajetórias circulares. Com relação ao sistema de movimento horizontal em altas velocidades, é apresentado o maior erro de contorno e de seguimento, com presença de oscilações na velocidade limite. Mesmo que o controle conseguisse ajustar-se adequadamente ao sistema, ele continuaria apresentando erro com o controle PID de posição. Em mudanças de movimento, há a maior

influência, e oscilações em baixa velocidade, como produto dos atritos presentes no eixo do motor que incrementa o torque dinâmico do sistema.

Para a implementação de interface de acionamento e controle em FPGA no primeiro momento utilizou-se blocos DSP Builder a fim de implementar os controladores e, em seguida, ocorreu a implementação em código VHDL.

A principal vantagem desse método é sua independência do hardware utilizado e dispensar licença para ser compilado e carregado. Por outro lado, como há necessidade de se trabalhar com números inteiros, a precisão inerente do método demonstra ser menor do que no caso do DSP Builder; acarretando em resultados de menor qualidade. Apesar disso, a validade da metodologia ainda é preservada.

Com os blocos de controle implementados em VHDL foi constituído um programa no ambiente Quartus II para acionamento dos motores, o qual foi carregado na placa FPGA.

Os resultados obtidos por meio da utilização da ferramenta DSP Builder mostram-se próximos daqueles obtidos via simulação, porém ao se empregar o código VHDL observou-se um incremento do erro cometido, esse incremento pode ser em decorrência do processo de transformação de números em ponto flutuante para números inteiros, sem comprometer a validade da metodologia.

Por fim, efetivou-se a montagem experimental para acionamento do dispositivo cujos resultados permitiram concluir que o controle preditivo produz resultados mais satisfatórios do que o controle por alocação de pólos. Além de demonstrar ser mais adequado para o tipo de aplicação desenvolvida.

O sistema motor e acoplamentos mostrou uma alta influência na variação da velocidade quando o motor está em vazio ou quando utiliza os acoplamentos, ao verificar-se essa parte é possível encontrar-se fontes de folgas e maior ruído na medição. A importância de realizar testes segundo os níveis da estratégia de estimação possibilita encontrar valores dos parâmetros mais próximos dos valores reais.

A validação experimental do sistema de acionamento e controle projetados, também foram apresentados nesse capítulo. A validação experimental foi realizada através do teste do controlador PID na máquina CNC, devido às interfaces disponíveis para o teste, considerando que os resultados desse controlador permitem validar a modelagem e identificação de parâmetros do sistema, assim é possível ter uma validação em simulação dos outros controladores, que em trabalhos futuros poderão ser implementados com tecnologias para sistemas de arquitetura aberta.

6 Resultados e discussões

Nesse trabalho foi realizado uma proposta de implementação da arquitetura de controle aberta para regulação e controle dos dispositivos máquinas-ferramenta CNC. Essa arquitetura possui o nível de referência, controle, compensação, diagnóstico e otimização totalmente abertas, possibilitando com isso o livre acesso tanto ao software quanto ao hardware das máquinas ferramentas CNC, com o objetivo de obter-se uma melhoria constante em seu desempenho e funcionabilidade, também obtendo-se uma maior facilidade sua de programação para execução de tarefas com maior grau de dificuldade e complexidades.

Uma máquina-ferramenta CNC foi desenvolvida para realizar um estudo comparativo entre dois tipos de sistemas de acionamento (motor de passo e motor de c.c.).

Esse tipo da máquina é largamente empregado em processos devido a sua alta precisão de atuação.

O motor de passo forneceu um resultado melhor do que o motor da c.c., mas a velocidade de trabalho do motor de passo é menor e não é recomendado para aplicações que necessitam de altas velocidades.

Outra vantagem do motor de passo é a ausência de um controlador, reduzindo o custo total do projeto com esse tipo de motor.

Por outro lado, a vantagem de usar o motor da c.c. é a velocidade mais elevada durante o processo de amostragem.

Ao longo desse trabalho obteve-se os resultados, abaixo descritos:

Otimização do Controle do Dispositivo.

Implementação de novos tipos de acionamentos.

Implementação de nova metodologia de controle com *labview*.

Obtenção de uma bancada de simulação flexível a diversas aplicações.

Construir uma ferramenta para auxílio em futuras dissertações.

Difundir o conceito de Arquitetura Aberta.

7 Conclusões e sugestões para próximos trabalhos

7.1 Conclusões

Essa pesquisa se propôs a desenvolver um sistema embarcado para supervisão e controle de movimentos de um sistema de posicionamento com 3 GL, que apresentam arquitetura de controle e supervisão aberta.

Para tal, inicialmente procedeu-se ao estudo de métodos de prototipagem rápida, tais como, a ferramenta *Labview* a interface *National-Motion* e o FPGA, e demais estudos citados anteriormente.

Esses exemplos utilizam conceitos apresentados nos capítulos anteriores dessa dissertação, sendo adicionados blocos de hardware e software dentro do conceito do sistema aberto proposto nesse trabalho.

A escolha desses sistemas práticos para validação desses conceitos pode ser justificada por fatores como facilidade de implementação, capacidade de agregar sensores e atuadores em diferentes aplicações, sendo apresentadas detalhadamente as diferentes etapas de implementação de diferentes plataformas baseadas em aplicações industriais, demonstrando pelos resultados obtidos, que as ferramentas de prototipagem rápida utilizada se mostram eficientes e de fácil implementação. Assim, esses conceitos podem ser sintetizados sob a forma das aplicações experimentais apresentadas, provendo bases para um melhor entendimento de seu funcionamento e de suas restrições.

No transcorrer desse trabalho foi implementada uma arquitetura de acionamento e controle, e integração com dispositivos embarcados de controle de juntas e sistema de visão e software para geração automática de movimentos, com validação a partir de situações reais.

Finalmente, esse trabalho permitiu o desenvolvimento de um ambiente aberto fornecendo ainda subsídios para a análise e implementação de estratégias para concepção dessas aplicações.

7.2 Perspectivas Futuras

Tendo o estudo atingido os objetivos pretendidos, delineam-se como perspectivas de futuros trabalhos algumas extensões e evoluções dos resultados aqui apresentados, que são:

Flexibilidade, estudo de outras possíveis configurações para implementação de soluções mais abertas para diversos problemas associados com desenvolvimento de protótipos.

Adaptabilidade a novas tarefas: apenas os recursos necessários a execução de uma tarefa são utilizados. Sistemas mais simples são tratados com versões mais simples do ambiente proposto, evitando-se perda de recursos.

Ambiente aberto, permitir que seus blocos de componentes sejam totalmente acessados e eventualmente modificados para agregar novas soluções.

Facilidade de expansão, novos recursos podem ser facilmente adicionados a um projeto, permitindo a adição de novas funcionalidades.

Sistemas Cooperativos, desenvolvimento de interfaces para operação de sistemas embarcados cooperativos.

Referências

ALTINTAS, Y., KO, J. H. Chatter Stability of Plunge Milling. *CIRP Annals - Manufacturing Technology*, Vol. 55, No. 1, 2006.

Acar, M. Mechatronics Engineering Education in the UK. In: Joint Hungarian British International Mechatronics Conference, 1994, Budapeste. *Proceedings...* Budapest:

Computational Mechatronics Publ., 1994, p 763- 769.

Adam, Nabil R., Dogramaci, Ali., Current issues in computer simulation. New York : Academic, 1979. 292 p.

Aihara, C. K., *Projeto e Implantação de Plataforma Didática aplicada ao Ensino e Pesquisa em Automação*, Campinas. Faculdade de Engenharia Mecânica, Universidade Estadual de Campinas, 2000, 104p. Tese (Mestrado).

Aihara, C. K., Cosso, S. G., Saramago, M. A. P. Rosário, J. M. Desenvolvimento de Aplicativos para Monitoramento de Variáveis de Controle de Processos Industriais. In: *Aplicon 2001*, EEUSP São Carlos, Julho 2001.

Aihara, C. K., Rosário, J.M., Desenvolvimento de Metodologias Aplicadas ao Ensino e Pesquisa em Automação, In: *World Congress on Engineering and Technology Education, WCETE2004*, Santos, São Paulo, Março 2004.

Aihara, C. K., Rosário, J. M., A Methodology Proposal For Teaching And Research In Automation Using Computer Mediatededucation Tool, In: *11th IFAC Symposium on Information Control Problems in Manufacturing INCOM2004*, Salvador, Brazil, Abril 2004.

Araujo, Emerson dos Santos, Modelagem e Descrição da Parte Comando de um Sistema Automatizado de Produção utilizando o GRAFCET - Aplicado à uma Plataforma Industrial em Automação. Campinas: Faculdade de Engenharia Mecânica, Universidade Estadual de Campinas, 1997, 91 p. Tese (Mestrado).

- Battesini, M., Sistemas Produtivos, Disponível em: www.producao.ufrgs.br/webgrad/ENG09014/SP_II_C1.pdf. Acessado em novembro de 2003.
- Bosnardo, R. C., Um sistema de videoconferência par educação à distância baseado em padrões abertos. Dissertação de mestrado. UNICAMP. Faculdade de Engenharia Elétrica, 2001.
- Brams G. W. Réseaux de Petri: “Théorie et pratique, tomos 1 et 2, Masson Editions, 1983.
- Cardoso, Janete, Valette, Robert, Redes de Petri, Florianópolis: Editora da UFSC, 1997, 212 p.
- Cassandras, Christos G., Discrete event systems: modeling and performance analysis. Burr Ridge : Irwin, 1993, 790p.
- Chang de Zhang and Shin-Min Song. Forward kinematics of a class of parallel (Stewart) platforms with closed-form solutions. *Journal of Robotic Systems*, 9(1), February 1992.
- Cong-Xin Li Min-Jie Liu and Chong-Ni Li. Dynamics analysis of the Gough-Stewart platform manipulator. *Journal of Robotic Systems*, 9(1), February 1992.
- Coriat, B., Automação Microeletrônica e Competitividade: Tendências no Cenário Internacional. In, *Automação, Competitividade e Trabalho: A Experiência Internacional*. Schmitz, H. & Carvalho, R. Q., Humanismo, Ciência e Tecnologia “Hucitec”, São Paulo, 1988.
- Cosso, S. G., Integração de Ferramentas de Automação direcionadas à Aplicações de Telerobótica - Implementação de um Sistema de Supervisão e Controle num Sistema Teleoperado, Campinas, Faculdade de Engenharia Mecânica, Universidade Estadual de Campinas, 2002. 134p. Tese (Mestrado).
- Coutinho, L, A F.: “Um Ambiente Integrado de Desenvolvimento de Software a Robótica”, mestrado, UNICAMP, Setembro de 1993.
- Cruz, J. M.: “Projeto e Desenvolvimento de um Sistema de Geração Automática de Trajetória para Manipuladores”, UNICAMP, mestrado, Outubro de 1993.

D'Abreu, J. V. V., Integração de Dispositivos Mecatrônicos para Ensino-Aprendizagem de Conceitos na Área de Automação, Campinas, Faculdade de Engenharia Mecânica, Universidade Estadual de Campinas, 2002. Tese (Doutorado).

D'Abreu, João Vilhete Viegas, Construção de um Traçador Gráfico para fins Educacionais, Campinas, Faculdade de Engenharia Elétrica e Computação, Universidade Estadual de Campinas, 1994. Tese (Mestrado).

Dorf, R.C. e Bishop, R.H., 2001, "Sistemas de Controle Modernos", Vol. 8, Ed. LTC, Rio de Janeiro, Brasil.

Farina, E., 2009, "Modelagem Matemática de sistemas dinâmicos". 15 de maio de 2010, <<http://www.google.com.br/url?sa=t&source=web&cd=3&ved=0CB0QFjAC&url=http%3A%2F%2Fwww.marcos>

cassiano.com%2Farquivos%2FTURMAEM6X01%2F%3Fdownload%3DModelagem%2BMatem_tica%2Bde%2B

sistemas%2Bdinamicos.pdf&rct=j&q=modelagem%20matematica%20de%20sistemas%20dinamicos&ei=6DPWT

LelA4aBlAf01tmBCQ&usg=AFQjCNEdAo1e1Eb8iiGaUrjDeWVTw4DZng. .

Ferrer, J.A.G., 1993, "Controle de Posição de Motores de Corrente Contínua utilizando Microcontrolador Dedicado",

Dissertação de Mestrado, Escola Politécnica da USP, São Paulo, Brasil, 143p.
Matlab, V7.8, The MathWorks Inc.

Ogata, K., 1998, "Engenharia de Controle Moderno", Vol. 3, Ed. LTC, Rio de Janeiro, Brasil.
Santana, D.D., Silva, P.A. e Oliveira de, A.L., 2008, "Sistema de controle de velocidade e posição para mesa

Coordenada cartesiana utilizada em máquinas ferramenta", Revista de Ciências Exatas e Tecnologia, Vol.3, No.3,

pp 7-25.

Siemens.,1990, “Electrical Feed Drives for Machine Tools”, Ed. John Wiley & Sons.

Damásio, D. Tecnologia Educacional. Disponível em:

http://www.superobra.com.br/admin/news.asp?ID_New=1180&Pag=all_news.asp&ID_Sessao_New=1&ID_ANew=11. Acessado em: Outubro de 2004.

Daumas, M., Las grandes Etapas Del Progreso Técnico. Fondo de Cultura Económica, p. 63-123, México, 1983.

Demongodin, I., Koussoulas, N. T., Differential Petri Nets: Representing Continuous Systems in a Discret-Event World. IEEE Transactions on Automatic Control, vol. 43, n. 4, pp 573-579., 1998

Dias, C. H.: “Implementação de um Supervisor de Controle para Robôs Industriais”, mestrado , UNICAMP, Junho de 1993.

Eby, F. História da Educação moderna. 2. ed, Porto Alegre: Globo, 1976.

Ferreira, Edson P., *Robótica Básica Modelagem de Robôs*, R. Vieira Gráfica e Editora Ltda. Versão Preliminar Publicada para a V Escola Brasileiro-Argentina de Informática, Rio de Janeiro, 1991.

Frigotto, G., As mudanças tecnológicas e Educação da Classe Trabalhadora: Politécnia, Polivalência ou Qualificação Profissional?, In: Trabalho e Educação, p. 45-52, Papirus, Campinas, SP, 1992.

Fayan, B. L.: “Estudo e Especificação de um Supervisor de Controle para um Robô Industrial”, mestrado, UNICAMP, Dezembro de 1992.

Groover, M., et al. Robótica: Tecnologia e Programação. McGraw-Hill, São Paulo, 1988.

Groover, Mikell P. *Automation, Production Systems, and Computer Integrated Manufacturing* USA: Prentice-Hall International, Inc., 1987, 808 p.

Guimarães, E. et al. Real: A Virtual Laboratory for Mobile Robot Experiments, IEEE Transactions on Education, vol. 46 n.1 February, 2003.

Hartley, S. et al. Enhancing teaching using the Internet: report of the working group on the World Wide Web as an interactive teaching resource. In: Conference on Integrating Technology into computer Science Education, Barcelona, 1996. Disponível em: <http://doi.acm.org/10.1145/237466.237649>. Acessado em agosto de 2002.

Heer, D., Traylor, R.L., et al. Enhancing the Freshman and Sophomore ECE Student Experience Using a Platform for Learning™. In: IEEE Transactions on Education, vol 46, n. 4, p. 434-446, novembro 2003.

Hervella, C.: “Projeto e Desenvolvimento de um Sistema de um Controlador Programável Flexível para Manipuladores e Robôs Industriais”, mestrado, UNICAMP, maio de 1995.

Huber P., Jensen K., Shapiro R. M.; "Hierarchies in Coloured Petri Nets"; In: G. Rozenberg (ed.);

Advances in Petri Nets 1990; Lecture Notes in Computer Science, vol. 483, Springer, 1990, pp. 313-341 e em "High-level Petri Nets - Theory and Application"; K. Jensen, G. Rozenberg (Eds.); Springer-Verlag; 1991; pp. 215-243

Lobosco, Orlando Silvio & Dias, José Luiz P. da Costa; Seleção e Aplicação de motores

Elétricos; Makron Books - Gajski, Daniel D.; Principles of Digital Designer; Prentice-Hall; 1997

Taub, Herbert; Circuitos Digitais e Microprocessadores; McGraw-Hill; 1984

PRITSCHOW, G., KRAMER, C. Open system architecture for drives. *CIRP Annals* 2005, Antalya, Vol. 54, No. 1, Aug. 2005.

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.

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.

YUN, W., MIN, B., PASEK, Z. Open Architecture Controller Interface Implementation for Machine Tools. In: IPROMS 2007 Innovative Production Machines and Systems. 2007.

ABRÃO, A. M., RUBIO, J. C., **Modernização de máquinas-ferramenta “Retrofitting”**. Disponível em:

<http://www.demec.ufmg.br/Grupos/Usinagem/RETROFITTING.htm> , Acesso 08/08/2011.

<http://www.mundocnc.com.br/conceito4.php> Acessado em: 06 de Setembro de 2012

<http://pt.wikipedia.org/wiki/Retrofit> Acessado em: 06 de Setembro de 2012

<http://sine.ni.com/nips/cds/print/p/lang/pt/nid/13824> Acessado em: 06 de Setembro de 2012

<http://www.eletronica.org/modules.php?name=News&file=article&sid=15> Acessado em: 06 de Setembro de 2012

<http://pt.scribd.com/doc/18503391/SISTEMAS-EMBARCADOS> Acessado em: 06 de Setembro de 2012

http://www.schneider-electric.com.br/documents/cadernos-tecnicos/apostila_procobre_eficienc.pdf Acessado em: 06 de Setembro de 2012

<http://alvarestech.com/temp/cobef2011/grima.ufsc.br/cobef2011/media/trabalhos/COF11-0157.pdf> Acessado em: 07 de Setembro de 2012

http://www.fee.unicamp.br/revista_sba/volumes/v7n1/v7n1a04.pdf Acessado em: 07 de Setembro de 2012

<http://www.dee.ufc.br/anexos/TFCs/2010-1/Matheus%20Sales%20Bezerra.pdf> Acessado em: 07 de Setembro de 2012

Anexo A – Software em linguagem C

Software em linguagem C embarcado nos microcontroladores PIC 18F2520

Função:

A sua função é receber um sinal de 0 até 5Vdc provindo da Placa Motion e converter esse sinal em 2 PWM's que fornecem set-point de Velocidade para a placa de potência.

```
*****PIC MOTOR UNICAMP*****//

#include <18F2520.h>
#define adc=10
#define fuses HS,NOWDT,PUT,NOLVP,NOPROTECT,BROWNOUT,DEBUG
#define use delay(clock=20000000)
#define use rs232(baud=19200, xmit=PIN_C6, rcv=PIN_C7, PARITY=N, BITS=8) //seta rs 232

*****Variaveis Globais*****//

long int loop =0;
int32 contador_hora =0; //Contador da hora para display
long int val_temp =0; // Valor temporário da Tensão
long int set_pwm1; // setpoint pwm

long int set_pwm2; // setpoint pwm
float real_tensao =0; // Valor real da tensão

void main(void)
{
*****inicializa conv A/D *****//
```

```

setup_ADC_ports (AN0_TO_AN1_ANALOG);
setup_adc(ADC_CLOCK_INTERNAL);
set_adc_channel (0);

//*****TIMER E PWM *****/

setup_timer_2 ( T2_DIV_BY_16,248,1 ); //timer 2= 1,004 kHz
setup_ccp1 (ccp_pwm); // configura ccp1 para modo PWM
setup_ccp2 (ccp_pwm); // configura ccp2 para modo PWM
set_pwm2_duty (0); // configura o ciclo ativo em zero desligado
set_pwm1_duty (0); // configura o ciclo ativo em zero desligado

// The cycle time will be (1/clock)*4*t2div*(period+1)
// In this program clock=20000000 and period=127 (below)
// For the three possible selections the cycle time is
// (1/20000000)*4*1*128 = 51.2 us or 19.5 khz
// (1/20000000)*4*4*128 = 204.8 us or 4.9 khz
// (1/20000000)*4*16*128= 819.2 us or 1.2 khz

while (true)
{
  // Início
  //*****Rotina de Pisca Pisca *****/
  contador_hora = contador_hora + 1;
  if(contador_hora > 350) // valor 350 aprox = 1 seg.
  {
    contador_hora = 0; // zera contador
    output_bit ((PIN_B5),!input(PIN_B5)); // inverte valor do led de indicação
  }
}

```

```

/**rotina que lê a entrada analógica***/
val_temp = 0;
for (loop=0; loop <=59;loop++) // Lê 60 valores para fazer média
{
    val_temp = val_temp + read_adc();
}
// val(volts)=(5/1024) * Val lido
//(5/1024) = 0.0048828125
//real_tensao = k0 * 0.0048828125 * ((float)val_temp)/ 60.0; //média de 60
real_tensao = 0.0048828125 * ((float)val_temp)/ 60.0; //média de 60
/**final rotina que lê entrada analógica ***/
if ( 2.4 <= real_tensao <= 2.6 ) // Banda morta
{
    set_pwm1 = 0;
    set_pwm2 = 0;
}
if ( real_tensao < 2.4 ) // nível abaixo
{
    // Y=Ax + b ---> Y = ((real_tensao * -409.2) + 1023)
    set_pwm2 = (long int)((real_tensao * -409.2) + 1023);
    output_low (PIN_B4);
    // Escreve no pwm 1
    set_pwm1 = 0;
}
if ( real_tensao > 2.6 ) // nível acima
{
    // Y=Ax + b ---> Y = ((real_tensao * 409.2) - 1023)
    set_pwm1 = (long int)((real_tensao * 409.2) - 1023);
}

```



```

output_high (PIN_B4);
// Escreve no pwm 2
set_pwm2 = 0;
}

//*****Rotina que escreve no PWM*****//
//set_pwm1 = 0;// setpoint PWM1
// 0 < PWM1 < 1024
if (set_pwm1>1023) set_pwm1 = 1023;
if (set_pwm1<0) set_pwm1 = 0;
set_pwm1_duty (set_pwm1); // seta PWM saida 1

//set_pwm2 = 0;// setpoint PWM2
// 0 < PWM2 < 1024
if (set_pwm2>1023) set_pwm2 = 1023;
if (set_pwm2<0) set_pwm2 = 0;
set_pwm2_duty (set_pwm2); // seta PWM saida 2
//*****//

} //Fecha com while true

} // Fecha com void main

```

Fluxograma

