

ESTE EXEMPLAR CORRESPONDE A REDAÇÃO FINAL DA  
TESE DEFENDIDA POR Luciano Antonio  
Frezzatto Santos ..... E APROVADA  
PELA COMISSÃO JULGADORA EM 26 / 08 / 2011  
.....  
João Maurício Rosário.....  
ORIENTADOR

UNIVERSIDADE ESTADUAL DE CAMPINAS  
FACULDADE DE ENGENHARIA MECÂNICA  
COMISSÃO DE PÓS-GRADUAÇÃO EM  
ENGENHARIA MECÂNICA

Luciano Antonio Frezzatto Santos

Planejamento de Trajetórias e  
Implementação de Técnicas de  
Posicionamento de Eixos para  
Dispositivo CNC com Arquitetura de  
Controle Aberta

Campinas, 2011.

Luciano Antonio Frezzatto Santos

# Planejamento de Trajetórias e Implementação de Técnicas de Posicionamento de Eixos para Dispositivo CNC com Arquitetura de Controle Aberta

Dissertação apresentada ao curso de Mestrado da Faculdade de Engenharia Mecânica da Universidade Estadual de Campinas, como requisito para obtenção do título de Mestre em Engenharia Mecânica.

Área de concentração: Mecânica dos Sólidos e Projeto Mecânico

Orientador: Prof. Dr. João Maurício Rosário

Campinas  
2011

FICHA CATALOGRÁFICA ELABORADA PELA  
BIBLIOTECA DA ÁREA DE ENGENHARIA E ARQUITETURA - BAE - UNICAMP

Santos, Luciano Antonio Frezzatto  
Sa59p      Planejamento de trajetórias e implementação de técnicas de posicionamento de eixos para dispositivo CNC com arquitetura de controle aberta / Luciano Antonio Frezzatto Santos. –Campinas, SP: [s.n.], 2011.

Orientador: João Maurício Rosário.  
Dissertação de Mestrado - Universidade Estadual de Campinas, Faculdade de Engenharia Mecânica.

1. Máquinas-ferramenta. 2. Controle em tempo real. 3. FPGA. 4. Robótica. I. Rosário, João Maurício. II. Universidade Estadual de Campinas. Faculdade de Engenharia Mecânica. III. Título.

Título em Inglês: Trajectory planning and implementation of axes positioning techniques for CNC device with open control architecture

Palavras-chave em Inglês: Machine tools, Real-time control, FPGA, Robotics

Área de concentração: Mecânica dos Sólidos e Projeto Mecânico

Titulação: Mestre em Engenharia Mecânica

Banca Examinadora: Ely Carneiro de Paiva, Pedro Luis Dias Peres

Data da defesa: 26/08/2011

Programa de Pós Graduação: Engenharia Mecânica

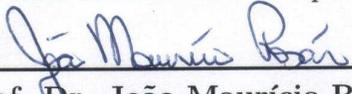
UNIVERSIDADE ESTADUAL DE CAMPINAS  
FACULDADE DE ENGENHARIA MECÂNICA  
COMISSÃO DE PÓS-GRADUAÇÃO EM  
ENGENHARIA MECÂNICA  
DEPARTAMENTO DE PROJETO MECÂNICO

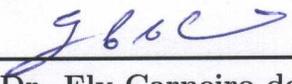
DISSERTAÇÃO DE MESTRADO ACADÊMICO

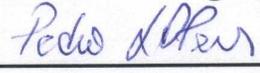
Planejamento de Trajetórias e  
Implementação de Técnicas de  
Posicionamento de Eixos para  
Dispositivo CNC com Arquitetura de  
Controle Aberta

Autor: Luciano Antonio Frezzatto Santos  
Orientador: Prof. Dr. João Maurício Rosário

A Banca Examinadora composta pelos membros abaixo aprovou esta Dissertação:

  
\_\_\_\_\_  
Prof. Dr. João Maurício Rosário, Presidente  
Faculdade de Engenharia Mecânica - UNICAMP

  
\_\_\_\_\_  
Prof. Dr. Ely Carneiro de Paiva  
Faculdade de Engenharia Mecânica - UNICAMP

  
\_\_\_\_\_  
Prof. Dr. Pedro Luis Dias Peres  
Faculdade de Engenharia Elétrica e de Computação - UNICAMP

Campinas, 26 de Agosto de 2011.

# Agradecimentos

Eu agradeço

a toda minha família pelo apoio e compreensão durante todos esses anos, que foram de extrema importância para alcançar este objetivo,

a minha namorada, Vanessa, por todos os momentos que estivemos juntos, pelas conversas, besteiras e apoio mútuo no desenrolar de nossos mestrados,

ao meu orientador pelos conselhos que foram úteis,

a todos os funcionários da Faculdade de Engenharia Mecânica que direta ou indiretamente ajudaram na concretização deste projeto,

ao Miro pela ajuda nos momentos difíceis, pelas discussões, cafés e “quebra-galhos”,

ao pessoal do Laboratório de Automação e Robótica, em especial ao Alvaro, Augusto e Silas pelo auxílio em algumas etapas do projeto, pelas conversas descontraídas e pela amizade,

a FAPESP pelo apoio financeiro concedido para a realização do projeto,

e, a todos os meus amigos.

*Tudo vale a pena se a alma não é pequena.*

*Fernando Pessoa*

# Resumo

Máquinas-ferramenta CNC são dispositivos complexos que executam movimentos automáticos, precisos e consistentes. Com o propósito de aprimorar o desempenho destes dispositivos face às mudanças bruscas de parâmetros agregados a posição, velocidade e aceleração no decurso de determinada trajetória torna-se imprescindível o estudo de novas arquiteturas de controle. A automação flexível permite que dispositivos robóticos mostrem-se cada vez mais velozes, motivando o desenvolvimento de técnicas que possibilitem de modo eficiente a substituição de controladores clássicos do tipo PID. Este estudo propôs-se a implementar um método alternativo para acionamento de um dispositivo cartesiano CNC de alta precisão com ênfase no desenvolvimento de um ambiente computacional para planejamento de movimentos e síntese de controladores. A geração de movimentos foi realizada por meio de um método de interpolação por *splines* que garantiu a continuidade e a suavidade dos perfis de trajetória. Para garantir que tais perfis fossem corretamente executados pelo dispositivo cartesiano, dois métodos de controle baseados no modelo do sistema foram implementados de forma a estabelecer comparação entre seus desempenhos. Em seguida, concebeu-se um ambiente de simulação com arquitetura aberta e flexível, o qual permitiu validar as estratégias propostas além de analisar a influência que perturbações externas causavam sobre o comportamento do dispositivo cartesiano. Tendo as estratégias sido validadas, procedeu-se à implementação em dispositivo físico utilizando uma placa FPGA para embarcar os controladores sintetizados. Os resultados mostraram que as estratégias propostas apresentaram bom desempenho quando empregadas junto ao dispositivo cartesiano CNC. Outrossim, o ferramental desenvolvido no estudo é suficientemente genérico para ser estendido a outras aplicações utilizando máquinas-ferramenta CNC.

**Palavras-chave:** máquinas-ferramenta, técnicas de controle, geração de trajetórias, prototipação em FPGA.

# Abstract

CNC machine tools are complex devices that execute automatic, precise and consistent movements. Aiming to improve the performance of these devices due to sudden changes on parameters associated to position, speed and acceleration while performing a given trajectory, it becomes indispensable the study of new control architectures. The flexible automation allows faster robotic devices, motivating the development of techniques that efficiently substitute classic PID controllers. This study aimed to implement an alternative method for driving a high-precision CNC cartesian device focusing on the development of a computing environment for movement planning and synthesis of controllers. The movement generation was performed by means of a spline interpolation method that ensured the continuity and smoothness of trajectory profiles. To ensure that these profiles were correctly executed by the cartesian device, two control techniques based on the model of the system were implemented in order to establish comparisons between their performances. Then, a simulation environment with open and flexible architecture was conceived, which allowed the validation of the proposed techniques besides analyzing the influence of external disturbances on the cartesian device behavior. Having the strategies been validated, the implementation in physical device was done using a FPGA board to embed the synthesized controllers. The results showed that the proposed strategies had good performance when applied to the CNC cartesian device. Furthermore, the tools developed in the study are general enough to be extended to other applications using CNC machine tools.

**Keywords:** machine tools, control techniques, trajectory generation, FPGA prototyping.

## Lista de Figuras

1.1	Dispositivo robótico CNC com três graus de liberdade. . . . .	4
2.1	Supervisão e arquitetura de controle de um dispositivo máquina-ferramenta CNC. . . . .	6
2.2	Forma canônica do controlador RST. . . . .	11
2.3	Princípio de funcionamento do controle preditivo. . . . .	13
3.1	Máquina-ferramenta CNC com identificação de seus componentes. . . . .	14
3.2	Junta robótica com perturbação. . . . .	15
3.3	Representação esquemática de um motor CC controlado por corrente de armadura. . . . .	18
3.4	Diagrama de blocos do motor de corrente contínua controlado por armadura. . . . .	19
4.1	Problema básico de interpolação. . . . .	21
4.2	Pontos nodais adicionais. . . . .	23
4.3	Curva genérica na qual a velocidade máxima pré-estabelecida é atingida. . . . .	26
4.4	Trajetoórias de referência utilizadas para interpolação. . . . .	31
4.5	Resultado da interpolação utilizando <i>splines</i> para diferentes trajetórias de referência. . . . .	31
4.6	Perfis de velocidade para as trajetórias triangular (a) e helicoidal (b) interpoladas por <i>splines</i> . . . . .	32
5.1	Estrutura geral de um controlador discretizado. . . . .	35
5.2	Esquema de funcionamento de um controlador preditivo. . . . .	38
6.1	Modelo de simulação da máquina-ferramenta CNC de três eixos utilizando motor CC. . . . .	45
6.2	Módulo de controle utilizando blocos Simulink <sup>TM</sup> . . . . .	45
6.3	Resposta ao degrau unitário e lugar das raízes para o sistema em malha fechada. . . . .	47
6.4	Comparação entre entrada e saída do sistema para trajetória triangular e erro cometido ao longo da trajetória. . . . .	47

6.5	Curvas de velocidade, corrente e potência na saída do motor para trajetória triangular. . . . .	48
6.6	Comparação entre entrada e saída do sistema para trajetória helicoidal e erro cometido ao longo da trajetória. . . . .	49
6.7	Curvas de velocidade, corrente e potência na saída do motor para trajetória triangular. . . . .	49
6.8	Resposta ao degrau unitário e pólos do sistema em malha fechada. . . . .	50
6.9	Comparação entre entrada e saída do sistema para trajetória triangular e erro cometido ao longo da trajetória. . . . .	51
6.10	Curvas de velocidade, corrente e potência na saída do motor para trajetória triangular. . . . .	52
6.11	Comparação entre entrada e saída do sistema para trajetória helicoidal e erro cometido ao longo da trajetória. . . . .	52
6.12	Curvas de velocidade, corrente e potência na saída do motor para trajetória triangular. . . . .	53
6.13	Influência de perturbações sobre o sinal de controle utilizando controlador por alocação de pólos. . . . .	54
6.14	Influência de perturbações sobre o sinal de controle utilizando controlador preditivo. . . . .	54
6.15	Influência de perturbações sobre o sinal de controle utilizando controlador por alocação de pólos. . . . .	55
6.16	Influência de perturbações sobre o sinal de controle utilizando controlador preditivo. . . . .	55
7.1	Bloco de controle utilizando blocos DSP Builder. . . . .	58
7.2	Comparação entre entrada e saída do sistema para trajetória triangular utilizando controlador por alocação de pólos implementado em DSP Builder. . . . .	59
7.3	Comparação entre entrada e saída do sistema para trajetória helicoidal utilizando controlador por alocação de pólos implementado em DSP Builder. . . . .	59
7.4	Comparação entre entrada e saída do sistema para trajetória triangular utilizando controlador preditivo implementado em DSP Builder. . . . .	59
7.5	Comparação entre entrada e saída do sistema para trajetória helicoidal utilizando controlador preditivo implementado em DSP Builder. . . . .	60
7.6	Bloco de controle utilizando código VHDL. . . . .	60
7.7	Comparação entre entrada e saída do sistema para trajetória triangular utilizando controlador por alocação de pólos implementado em VHDL. . . . .	61

7.8	Comparação entre entrada e saída do sistema para trajetória helicoidal utilizando controlador por alocação de pólos implementado em VHDL. . . . .	61
7.9	Comparação entre entrada e saída do sistema para trajetória triangular utilizando controlador preditivo implementado em VHDL. . . . .	62
7.10	Comparação entre entrada e saída do sistema para trajetória helicoidal utilizando controlador preditivo implementado em VHDL. . . . .	62
7.11	Esquemático em Quartus II para teste de controlador desenvolvido. . . . .	63
7.12	Placa de potência utilizada para acionamento do motor CC. . . . .	64

## Lista de Tabelas

3.1	Parâmetros do sistema. . . . .	17
6.1	Parâmetros do sistema de acionamento e do dispositivo robótico. . . . .	46

# Lista de Abreviações e Siglas

## Abreviações

A/D - Conversor analógico-digital

CC - Corrente contínua

CNC - *Computer Numerical Control*

D/A - Conversor digital-analógico

GDL - Grau de liberdade

GPC - *Generalized Predictive Control*

HIL - *Hardware in the loop*

NC - *Numerical Control*

## Letras Gregas

$\Delta(q^{-1})$  - Operador diferença

$\eta$  - Razão de redução

$\lambda$  - Fator de ponderação sobre os incrementos do sinal de controle

$\omega(t)$  - Velocidade angular do motor [rad/s]

$\tau$  - Torque [Nm]

$\tau_r$  - Torque resistente devido a perdas [Nm]

## Letras Latinas

$B$  - Atrito viscoso total do sistema [N·s/m]

$d_p$  - Diâmetro do fuso [m]

$h_p$  - Deslocamento por volta do fuso [m/rev]

$i(t)$  - Corrente gerada no motor [A]

$J$	Momento de inércia total do sistema	[kg·m <sup>2</sup> ]
$J_l$	Inércia do fuso esférico refletido no eixo do motor	[kg·m <sup>2</sup> ]
$J_m$	Inércia do motor	[kg·m <sup>2</sup> ]
$J_{tw}$	Inércia refletida no motor devido à mesa e à peça de trabalho	[kg·m <sup>2</sup> ]
$k_e$	Constante de força contra-eletromotriz	[V/(rad/s)]
$k_t$	Constante de torque do motor	[Nm/A]
$L_a$	Indutância de armadura do motor CC	[H]
$m_l$	Massa do fuso esférico	[kg]
$m_t$	Massa da mesa	[kg]
$m_w$	Valor máximo da massa da peça sobre a mesa	[kg]
$N_1$	Mínimo horizonte de predição	
$N_2$	Horizonte de predição sobre o controle	
$N_2$	Máximo horizonte de predição	
$q^{-1}$	Operador atraso	
$R_a$	Resistência de armadura do motor CC	[Ω]
$V(t)$	Tensão aplicada aos terminais do motor	[V]
$v^*$	Máxima velocidade atingida durante a execução de uma trajetória	[m/s]
$V_m$	Tensão aplicada entre os terminais do motor	[V]

### **Siglas**

CAD - *Computer-Aided Design*

FPGA - Field-Programmable Gate Array

PID - Proporcional-integral-derivativo

PWM - Pulse Width Modulation

RST - Estrutura polinomial empregada na síntese de controladores a tempo discreto

VHDL - VHSIC Hardware Description Language

VHSIC - Very-High-Speed Integrated Circuits

# Sumário

<b>1</b>	<b>Introdução</b>	<b>1</b>
1.1	Justificativa . . . . .	2
1.2	Objetivos . . . . .	3
1.2.1	Objetivos Específicos . . . . .	3
1.3	Metodologia . . . . .	3
1.4	Organização da Pesquisa . . . . .	4
<b>2</b>	<b>Revisão Bibliográfica</b>	<b>6</b>
2.1	Planejamento de Trajetórias . . . . .	7
2.2	Sistema de Controle . . . . .	10
2.3	Conclusões . . . . .	12
<b>3</b>	<b>Modelamento do Sistema</b>	<b>14</b>
3.1	Modelo Cinemático . . . . .	15
3.2	Modelo Dinâmico . . . . .	16
3.3	Sistema de Acionamento . . . . .	17
3.3.1	Motor de Corrente Contínua . . . . .	17
3.4	Conclusões . . . . .	18
<b>4</b>	<b>Planejamento de Trajetórias</b>	<b>20</b>
4.1	Interpolação utilizando <i>Splines</i> . . . . .	20
4.1.1	Síntese de Curvas Genéricas . . . . .	26
4.1.2	Avaliação dos Algoritmos . . . . .	30
4.2	Conclusões . . . . .	32
<b>5</b>	<b>Sistema de Controle</b>	<b>33</b>
5.1	Controlador com Alocação de Pólos . . . . .	33
5.2	Forma Polinomial RST . . . . .	35
5.3	Controlador Preditivo . . . . .	37
5.3.1	Preditor Ótimo . . . . .	39

5.3.2	Controlador Preditivo na Forma RST . . . . .	40
5.4	Conclusões . . . . .	43
<b>6</b>	<b>Teste e Validação das Estratégias Desenvolvidas</b>	<b>44</b>
6.1	Controlador por Alocação de Pólos . . . . .	46
6.2	Controlador Preditivo . . . . .	50
6.3	Influência de Perturbações Externas . . . . .	53
6.4	Conclusões . . . . .	56
<b>7</b>	<b>Implementação em FPGA</b>	<b>57</b>
7.1	Implementação em Blocos DSP Builder . . . . .	57
7.2	Implementação com Código VHDL . . . . .	60
7.3	Implementação em Ambiente Quartus II . . . . .	62
7.4	Conclusões . . . . .	65
<b>8</b>	<b>Considerações Finais e Perspectivas Futuras</b>	<b>66</b>
8.1	Perspectivas Futuras . . . . .	67
8.2	Trabalhos Produzidos . . . . .	68
8.3	Trabalhos Relacionados . . . . .	68
	<b>Referências Bibliográficas</b>	<b>68</b>
	<b>Anexos</b>	
<b>A</b>	<b>Algoritmos Implementados</b>	<b>73</b>
A.1	Splines Cúbicas com Restrição . . . . .	73
A.2	Busca Binária . . . . .	74
<b>B</b>	<b>Códigos Implementados</b>	<b>75</b>
B.1	Controle por Alocação de Pólos . . . . .	75
B.2	Controle Preditivo . . . . .	79
B.3	Leitura do Encoder . . . . .	84
B.4	Driver de PWM . . . . .	85
B.5	Geração do PWM . . . . .	86
<b>C</b>	<b>Vídeos</b>	<b>89</b>

# 1 Introdução

Máquinas-ferramenta são dispositivos empregados tipicamente na fabricação 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, formam a base da indústria sendo usadas tanto direta quanto indiretamente na manufatura de diversos produtos disponíveis (BOOTHROY; KNIGHT, 2006). Com a evolução da tecnologia, o controle dos eixos das máquinas que era feito de forma manual, passou a ser realizado utilizando Controle Numérico (*Numerical Control* - NC), no início programado em mídias como cartões perfurados ou fitas magnéticas, e posteriormente por meio de Controle Numérico por Computador (CNC). Nas máquinas-ferramenta CNC atuais são empregados microcontroladores ou placas de lógica reprogramável em substituição aos computadores inicialmente utilizados, que permitem tornar a aplicação dedicada além de melhorar o seu desempenho.

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, estes dispositivos possuem uma arquitetura de controle pouco flexível e fechada e, com o passar dos anos, podem tornar-se obsoletas necessitando de reajuste (*retrofitting*) do seu sistema de acionamento e controle.

Na presente pesquisa é proposta a concepção de um ambiente de simulação que reproduza o princípio de operação de uma máquina-ferramenta real e que possa, também, auxiliar na síntese de controladores, no estudo de sistemas de acionamento e na análise da influência de perturbações externas, como vibrações, folgas, desgastes.

## 1.1 Justificativa

A empregabilidade dos ambientes virtuais tem sido largamente estudada na literatura para fins diversos e, cada vez mais, esses ambientes vêm substituindo métodos de experimentação empírica além de agilizar processos de desenvolvimento, estudo e análise de diferentes dispositivos assim como dos elementos que os compõem (ALTINTAS et al., 2005). Outrossim, também é possível estudar como diferentes influências externas: forças cortantes, ruídos, atritos, vibrações, interferem no funcionamento ideal do dispositivo bem como auxiliam na proposição de meios que minimizem tais influências.

As vantagens da utilização de ambientes virtuais para o projeto de máquinas-ferramenta CNC, tais como redução de custos e economia de tempo, foram discutidas por Altintas e colaboradores (2005), ao mesmo tempo que, também, foram apresentados projetos de máquinas-ferramenta realizados dentro desses ambientes de maneira a ilustrar as vantagens mencionadas.

Yeung, Altintas e Erkorkmaz (2006) propuseram um sistema virtual baseado em uma máquina real que incluía a modelagem matemática de seus atuadores, suas forças de atritos, seu gerador de movimentos e suas leis de controle. Uma das principais contribuições deste trabalho centrou-se no desenvolvimento de um projeto com arquitetura aberta que possibilite ao usuário deter total liberdade para escolher a configuração que melhor se adequa às suas necessidades, além de realizar simulações e avaliações sobre o comportamento da máquina-ferramenta.

Por sua vez Kjellberg e colaboradores (2009) descrevem em seu trabalho uma aproximação na modelagem de máquinas-ferramenta utilizando de normas que regulam cada um dos componentes do dispositivo, cujo principal objetivo é a utilização destas máquinas em uma fábrica digital de forma a auxiliar o processo de manufatura real.

Em vista do exposto, a implementação de ambientes virtuais demonstra ser relevante na concepção e análise de novas máquinas e novos sistemas, uma vez que a reprodução correta do seu princípio de funcionamento permite a proposição de sistemas que melhor atendam as necessidades do usuário em um menor período de tempo se comparado aos métodos tradicionais empregados.

## 1.2 Objetivos

O objetivo geral desta pesquisa é propor um método alternativo para acionamento de um dispositivo cartesiano CNC de alta precisão com ênfase no desenvolvimento de um ambiente computacional para planejamento de movimentos e síntese de controladores, cujos resultados poderão ser estendidos para outras aplicações no campo da robótica.

### 1.2.1 Objetivos Específicos

Quanto aos objetivos específicos arrolam-se:

1. modelagem dinâmica do dispositivo robótico cartesiano similar a uma máquina-ferramenta CNC;
2. desenvolvimento de um planejador de movimentos;
3. desenvolvimento de um ambiente virtual para máquina-ferramenta CNC;
4. implementação de controladores sob a forma polinomial RST para implementação embarcada;
5. implementação dos controladores utilizando lógica reprogramável em FPGA (*Field-Programmable Gate Array*).

## 1.3 Metodologia

Inicialmente, esta investigação atentou para o estudo de máquinas-ferramenta para assim definir os parâmetros que permitissem realizar sua análise cinemática e dinâmica. Com o intuito de testar os modelos obtidos foi implementado um algoritmo para a geração de movimentos suaves, sem discontinuidades, a partir de pontos amostrados e, de modo a garantir que os movimentos fossem executados adequadamente, a fim de minimizar o erro cometido entre a entrada e a saída, além de inserir um controlador na malha do sistema. Duas classes de controladores foram alvo de estudo: os controladores clássicos, como o PID, que possuem característica reativa, os que respondem somente quando ocorre uma excitação em sua

entrada; e os controladores preditivos que possuem característica antecipativa, aqueles que conseguem prever o que ocorrerá no sistema antecipando sua ação de controle. A validação das metodologias desenvolvidas ocorreu, primeiramente, por meio da construção de um ambiente de simulação que incluía o gerador de movimentos, os controladores, os sistemas de acionamento e estrutura mecânica presentes em uma máquina-ferramenta e, posteriormente, efetivar sua implementação em dispositivo real, Figura 1.1, com auxílio de uma FPGA.



Figura 1.1: Dispositivo robótico CNC com três graus de liberdade.

#### 1.4 Organização da Pesquisa

Este estudo transcorreu segundo a metodologia preconizada.

O Capítulo 1 arrola a respeito da justificativa, dos objetivos, da metodologia e da organização da investigação.

O Capítulo 2 ateu-se à revisão bibliográfica sobre o sistema de controle e o planejamento de trajetórias focando sua aplicabilidade para máquinas-ferramenta CNC.

O Capítulo 3 trata do modelamento cinemático e dinâmico do dispositivo cartesiano CNC, disponível no Laboratório de Automação Integrada e Robótica da Faculdade de Engenharia Mecânica da UNICAMP, que constituiu-se no objeto de estudo desta dissertação.

Por fim efetivou-se a modelagem do sistema de acionamento empregado para movimentação de cada um dos eixos da máquina.

A geração de movimento que garante que os perfis de trajetória enviados à máquina-ferramenta sejam sempre suaves foi abordada no Capítulo 4. Neste capítulo, ainda, apresenta-se o algoritmo empregado para a geração de trajetórias para os eixos da máquina CNC estudada.

O Capítulo 5 centra-se nas sínteses dos controladores utilizados no dispositivo cartesiano em estudo. Neste capítulo, também, introduziu-se a forma polinomial RST utilizada como uma representação genérica para controladores na forma discreta para serem implementados em sistemas embarcados.

No Capítulo 6 é descrito o ambiente de simulação desenvolvido para a máquina-ferramenta CNC; a descrição dos resultados obtidos via simulação para diferentes tipos de controladores, considerando-se a presença e a ausência de perturbações externas.

A metodologia para implementação em uma placa de lógica reprogramável, apresentada no Capítulo 7, visou permitir a substituição do computador empregado, o que torna a aplicação independente.

Por fim, no Capítulo 8 são tecidas as considerações finais do estudo e possíveis propostas que possibilitem gerar novas investigações.

## 2 Revisão Bibliográfica

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 à estrutura do dispositivo e o sistema de controle é responsável pela geração de movimentos e pelo controle dos eixos do dispositivo.

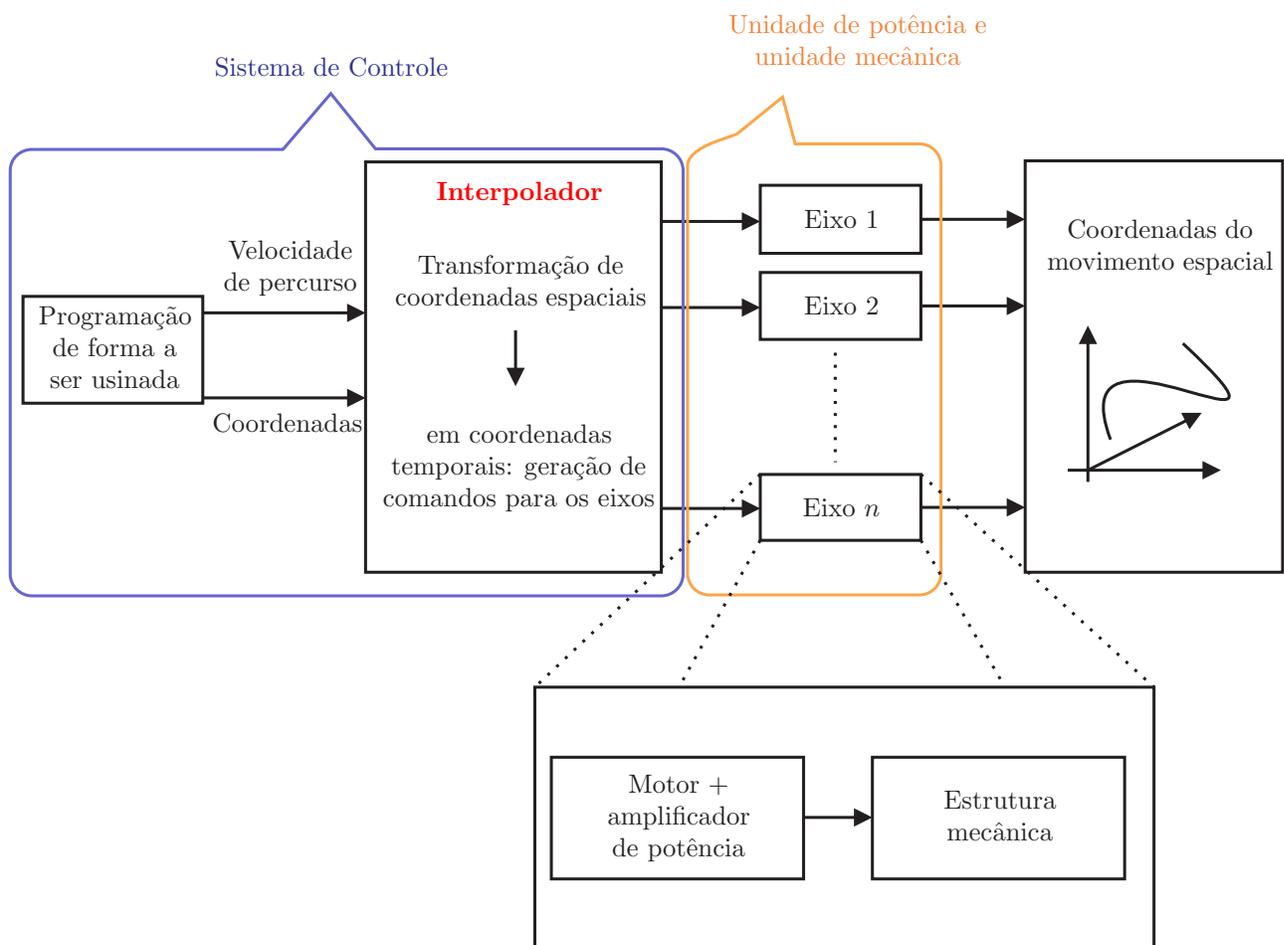


Figura 2.1: Supervisão e arquitetura de controle de um dispositivo máquina-ferramenta CNC.

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 desta configuração é deixar

um hardware dedicado para a realização da tarefa, aumentando a velocidade do processo e 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, conhecida como *código G*, ou por meio de sistemas CAD. 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-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 contar-se 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, buscaram-se na literatura especializada embasamentos que permitiram fundamentar análises e discussões que compõem as seções subsequentes.

## 2.1 Planejamento de Trajetórias

A geração de movimentos desempenha papel fundamental no controle computacional de máquinas-ferramenta, pois as trajetórias geradas devem não somente descrever o caminho da ferramenta, mas também apresentar perfis cinemáticos suaves de forma a manter alta precisão de rastreamento e evitar excitar os modos naturais da estrutura mecânica ou do sistema de servo-controle (ERKORKMAZ; ALTINTAS, 2001). Na geração de movimentos a partir de um conjunto de pontos de entrada são derivadas informações de posicionamento, velocidade e aceleração para cada um dos eixos da máquina-ferramenta no transcurso do tempo. Estas informações são necessárias, uma vez que estão diretamente relacionadas com o esforço de controle por parte do controlador caso haja alguma descontinuidade na velocidade e/ou na aceleração. Sob este cenário pode ocorrer esforço exagerado tanto por parte dos controladores quanto por parte dos motores para garantir o correto rastreamento da entrada podendo acarretar em danos sobre o equipamento.

Os geradores de movimento classificam-se em geradores *on-line* e geradores *off-line*.

Nos geradores *on-line*, a geração ocorre concomitantemente ao funcionamento da máquina-ferramenta, logo algoritmos otimizados e de rápida convergência devem ser utilizados com a finalidade de garantir bons resultados. Nos geradores *off-line*, parte-se do pressuposto que uma vez gerada a trajetória a ser executada pela máquina-ferramenta, esta terá pouca variação; assim é desnecessário refazer os cálculos a cada início de operação. Na literatura são citados os algoritmos baseados nas interpolações do tipo Bézier, NURBS (*Non-Uniform Rational B-Splines*) e *splines*, como os mais empregados para geração de trajetórias..

As curvas de Bézier foram desenvolvidas simultaneamente por Paul de Casteljaeu e Pierre Bézier ao final da década de 1950 e início da década de 1960. Elas resultam da demanda da indústria automotiva da época que buscava formas mais modernas para seus veículos (FIXTER, 2003). Estas curvas são descritas mais como um método para projeto de trajetória do que como um método para ajuste da mesma, dessa maneira a trajetória resultante exibe características da curva desejada sem se ajustar precisamente a esta curva.

Curvas de Bézier são largamente utilizadas em computação gráfica para a modelagem de curvas suaves onde, geralmente, são utilizadas curvas de baixa ordem para reduzir o custo computacional e aprimorar o desempenho dos aplicativos. Além desta aplicação, Chung e Hahn (1999) utilizaram curvas de Bézier para animação em ambientes virtuais do caminhar humano em terreno irregular. Já Lamiriaux e Kavraki (1999) usaram tal ferramenta a fim de modelar a deformação de placas elásticas presas por bordas opostas. Yau e Wang (2007) propuseram um algoritmo de interpolação rápida com função antecipativa em tempo real que identifica pequenos blocos de comandos G01, pertencentes ao código G, produzidos por modelos CAD e os interpola utilizando curvas de Bézier.

NURBS é uma generalização do algoritmo de Bézier capaz de descrever com precisão figuras em duas ou três dimensões. O algoritmo básico pode ser interpretado como a razão entre a soma ponderada dos pontos de entrada e o somatório de todos os pesos utilizados (PIEGL, 1991). As principais vantagens desta abordagem residem na grande flexibilidade para projetar uma larga gama de formas apenas manipulando os pontos de controle e os pesos e, a representação de contornos complicados que não exige o aumento da ordem da equação (WANG; YAU, 2009). Por outro lado as desvantagens deste método recaem sobre a alta carga computacional devido à recursividade do algoritmo e aos erros inerentes introduzidos pelo truncamento da série de Taylor (YAU; WANG, 2007).

A interpolação com NURBS é empregada em geradores de trajetória em tempo real

(*on-line*) devido as suas vantagens, porém ela também pode ser empregada para a geração de trajetórias *off-line* como proposto em (YANG; RED, 1997). No trabalho apresentado por Wang e Yau (2009) é proposta uma variação do algoritmo apresentado em (YAU; WANG, 2007) na medida em que os autores substituem a interpolação por curvas de Bézier pela interpolação utilizando NURBS. Como este método divide a curva em segmentos e faz a interpolação destes, o erro de interpolação causado por tal segmentação não é passível de ser totalmente controlado. Entretanto, esse erro pode ser compensado utilizando uma unidade de comprimento básica juntamente com um interpolador de dois estágios (PARK; NAM; YANG, 2005). Nesta proposta, primeiramente é feita uma interpolação padrão usando NURBS e, nos trechos onde o erro entre a curva desejada e a curva interpolada exceda o máximo permitido, é feita uma segunda interpolação mudando o parâmetro de curvatura daquele segmento e diminuindo o erro cometido.

Uma *spline* pode ser definida como a justaposição de polinômios de baixa ordem onde as condições de continuidade são satisfeitas entre os polinômios (SMITH, 1983). Normalmente assume-se que a ordem dos polinômios é 3, garantindo a continuidade do polinômio até sua segunda derivada, em termos físicos significa continuidade na aceleração, ou 5, assegurando a continuidade até a quarta derivada, isto é, a primeira, *jerk*, e a segunda derivadas da aceleração serão contínuas. O uso de um ou outro polinômio depende da aplicação e das limitações inerentes ao equipamento utilizado.

Em (DYLLONG; VISIOLI, 2003) as *splines* são empregadas como um método para alteração de trajetórias em tempo real de forma a evitar obstáculos. Baseando-se em *splines* de ordem 5 e trabalhando com a posição e a orientação final de um manipulador robótico, Herrera e Sidobre (2005) propuseram um método de planejamento de trajetórias em sete dimensões. Este trabalho foi posteriormente estendido para aplicações em tempo real em que há interação do operador com o dispositivo (HERRERA; SIDOBRE, 2006).

Nos trabalhos apresentados por Erkorkmaz e Altintas (2001) e Macfarlane e Croft (2003) são apresentados geradores de trajetórias que levam em consideração uma limitação do *jerk* do sistema. As vantagens de se limitar esta grandeza é garantir um melhor rastreamento da trajetória e diminuir o esforço de um dispositivo robótico. No primeiro trabalho é proposto um gerador *off-line* que realiza uma otimização da trajetória que será enviada ao dispositivo; já no segundo trabalho a geração é feita *on-line* sendo que a transição entre diferentes trechos da trajetória se realiza ao utilizar uma combinação parabólica dos pontos finais e iniciais do trecho origem e de destino, respectivamente.

O algoritmo OCAS (*Overlapping Cubic Arcs and cubic Splines*) proposto em (DU PLESSIS; SNYMAN, 2003) transforma uma trajetória qualquer, dada no plano cartesiano em perfis temporais que são enviados para cada eixo do dispositivo. A principal contribuição deste trabalho reside na adaptação do gerador ao dispositivo cartesiano em estudo, já que o algoritmo aceita como entradas informações sobre a cinemática do dispositivo e gera um perfil de trajetória adequado a cada caso.

## 2.2 Sistema de Controle

Um controlador é um dispositivo que monitora e influencia as condições operacionais de um sistema dinâmico que normalmente são as saídas do sistema que podem ser ajustadas por meio de certas variáveis de entrada.

O controlador PID é o controlador linear mais conhecido e utilizado industrialmente em decorrência de seu bom desempenho, desde que o sistema a controlar esteja adequadamente determinado, seja bem comportado (linear e invariante no tempo) e, os parâmetros do controlador bem ajustados (PASSOS, 2008). Com o propósito de garantir o desempenho do controlador PID, a correção do erro entre o sinal medido e o sinal de referência é estabelecida por meio de cálculo e determinação da ação que ajusta o processo rapidamente, mantendo o mínimo erro de rastreamento (OGATA, 1997).

No projeto de controladores clássicos, como o PID, há diversas técnicas empregadas para seu desenvolvimento: projeto via lugar das raízes, via resposta em frequência, alocação de pólos, critério de Routh, entre outras. Dentre elas, a técnica de alocação de pólos é de principal interesse, uma vez que possibilita que os pólos do sistema em malha fechada se situem em pontos pré-determinados dentro de uma região  $\Omega$  previamente escolhida, de forma que o sistema resultante apresente certas características desejadas, como máximo tempo de estabilização, margens de fase e de ganho e, máxima sobre-elevação em relação à entrada (GEROMEL; PALHARES, 2005).

No desenvolvimento deste tipo de controlador assume-se que sempre se trabalhe no espaço contínuo, porém, na maioria das aplicações em tempo real, é empregado algum tipo de micro-controlador que apresenta lógica discreta, o que torna imprescindível a conversão do espaço contínuo para o espaço discreto. Para realizar esta conversão pode ser empregada a

transformada bilinear (GEROMEL; PALHARES, 2005), também conhecida como transformação de Tustin, a qual estabelece uma transformação unívoca entre os dois espaços para um dado tempo de amostragem  $T_s$ .

A forma polinomial RST (LANDAU, 1990) é largamente empregada no desenvolvimento de controladores discretos, uma vez que todos os controladores a tempo contínuo podem ser discretizados sob essa forma possibilitando não somente uma análise mais aprimorada destes, mas também facilitando a substituição de um controlador por outro sem que haja necessidade de mudança da malha de controle. Outrossim, esta forma permite que qualquer controlador seja embarcado em dispositivo de lógica reprogramável.

A forma canônica de um controlador RST é retratada na Figura 2.2, sendo que  $q^{-1}$  corresponde ao operador atraso,  $R$  e  $S$  são filtros digitais projetados para atingir o desempenho de regulação desejado e o filtro digital  $T$  é, posteriormente, projetado para atingir o desempenho de rastreamento desejado (LANDAU, 1998).

Quando  $T = R$  o controlador está operando na forma de erro de regulação, o que não permite a especificação independente dos desempenhos de rastreamento e regulação. A discretização de controladores PID, geralmente, conduz a formas RST nas quais tal fenômeno ocorre, mostrando porque não é possível obter um bom desempenho simultaneamente em rastreamento e em regulação ao se empregar este tipo de controlador.

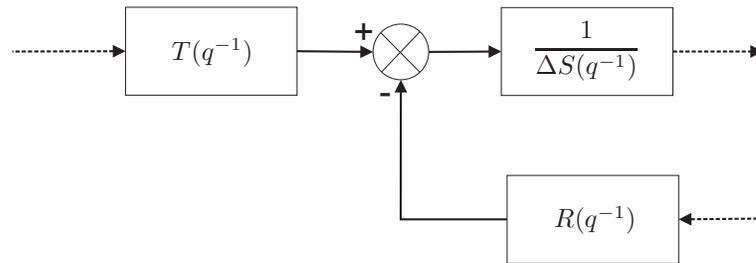


Figura 2.2: Forma canônica do controlador RST.

Apesar dos controladores clássicos apresentarem boas respostas para diversos processos, nem sempre todas as características dos sistemas a se controlar são bem conhecidas, o que dificulta a sintonização destes controladores de modo apropriado. Em decorrência deste fato, controladores mais robustos que possuam maior tolerância a incertezas do processo são, muitas vezes, mais interessantes do que os controladores clássicos.

A classe de controladores baseados no modelo vem sendo largamente empregada no

desenvolvimento de controladores mais robustos e com maior estabilidade. A ideia de utilizar o modelo com o intuito de prever o comportamento do sistema se tornou possível a partir do desenvolvimento do computador que forneceu os recursos necessários para a implementação desta técnica (SUSANU, 2005). O controle preditivo baseado no modelo, conhecido por “MPC” (*Model Predictive Control*), inclui as leis de controle baseadas na utilização explícita de um modelo do processo, a fim de desenvolver um controlador por meio da minimização de uma função de custo quadrática.

Os conceitos básicos dos controladores preditivos foram introduzidos primeiramente por (RICHALET et al., 1978) e (CUTLER; RAMAKER, 1980) no final dos anos 70. Nos dois trabalhos, os resultados obtidos são oriundos de aplicações industriais, cujas representações dos processos foram feitas por meio de modelos de resposta impulsiva. O maior interesse na teoria de controladores preditivos ocorreu no final na década de 1980 com a publicação do trabalho de Clarke, Mohtadi e Tuffs (1987) que introduziu o conceito de Controle Preditivo Generalizado (GPC) aplicável à maioria dos processos.

A metodologia GPC pode ser considerada uma extensão e evolução do controlador de variância mínima generalizado (GMV) (CLARKE; GAWTHROP, 1975) em virtude do incremento de previsões, tornando-o mais robusto quando aplicado a processos com atraso desconhecido ou variante (CLARKE; MOHTADI; TUFFS, 1987).

A ideia básica dos controladores preditivos reside no “saber” o que se sucederá no futuro tendo conhecimento somente do presente. Para tanto, é utilizado um modelo numérico do sistema a fim de prever a saída em um horizonte finito fazendo-a coincidir com a referência, tal como ilustrado na Figura 2.3. A existência de uma referência conhecida ou pré-calculada é um dos pré-requisitos para que a estratégia preditiva funcione.

## 2.3 Conclusões

Em suma, com o intuito de fundamentar o presente estudo uma investigação na literatura específica foi efetivada, de modo a propiciar referências teóricas que possibilitassem aprofundar conhecimentos a respeito da estrutura e funcionamento de máquinas-ferramenta CNC. O propósito deste estudo teórico ateve-se à análise da geração de movimentos e sistemas de controle, desenvolvidas por outros investigadores da área.

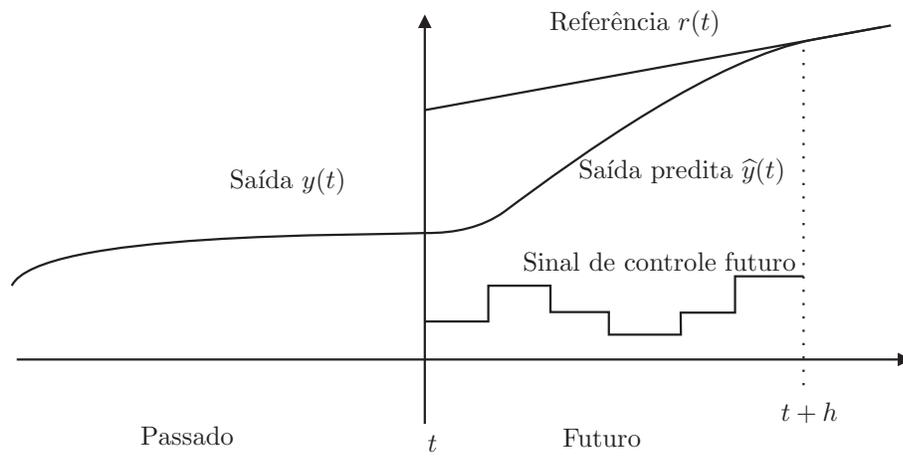


Figura 2.3: Princípio de funcionamento do controle preditivo.

Nos capítulos subsequentes prosseguir-se-á examinando e discutindo minuciosamente o modelamento do sistema, o planejamento de trajetórias, assim como os sistemas de controle.

### 3 Modelamento do Sistema

A máquina-ferramenta deste estudo corresponde a um dispositivo cartesiano de 3 graus de liberdade (GDL) em que os eixos  $X$  e  $Y$  do dispositivo constituem a mesa de trabalho, enquanto o eixo  $Z$  movimenta-se verticalmente para posicionamento da ferramenta. Cada um dos eixos é composto por um encoder, um motor, que é acoplado à carga por meio de um fuso esférico e, uma certa carga. Na Figura 3.1 é apresentada a máquina-ferramenta identificando cada um de seus componentes.

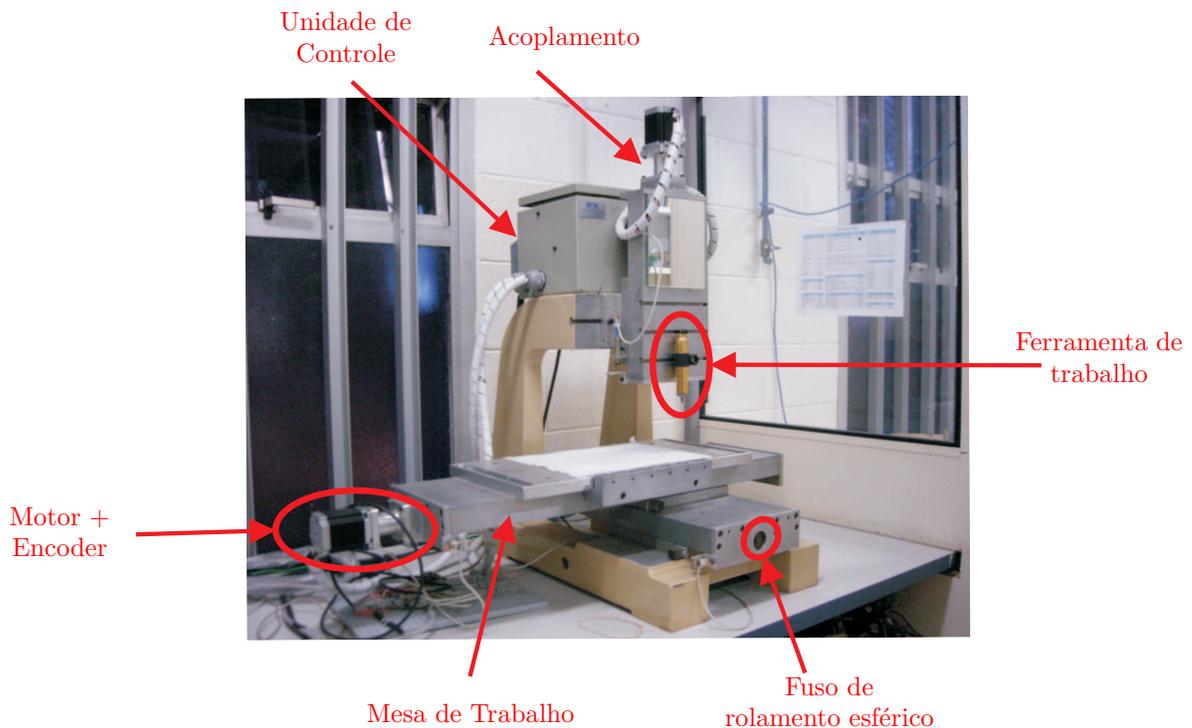


Figura 3.1: Máquina-ferramenta CNC com identificação de seus componentes.

Como há independência entre cada um dos eixos da máquina-ferramenta, uma bancada experimental, junta robótica com perturbação, representativa de um dos graus de liberdade do sistema em estudo, mostrado na Figura 3.2, representativo de cada um dos eixos do dispositivo cartesiano adotado foi utilizado, inicialmente, para estudo e validação dos algoritmos propostos. Este dispositivo robótico é constituído por um *encoder*, um motor, um fuso e cargas que também estão presentes em cada um dos eixos da máquina cartesiana. Além disso, há a possibilidade de imprimirem-se perturbações que simulam as interferências causadas tanto por forças externas quanto pelas forças de contato entre eixos da máquina-ferramenta.

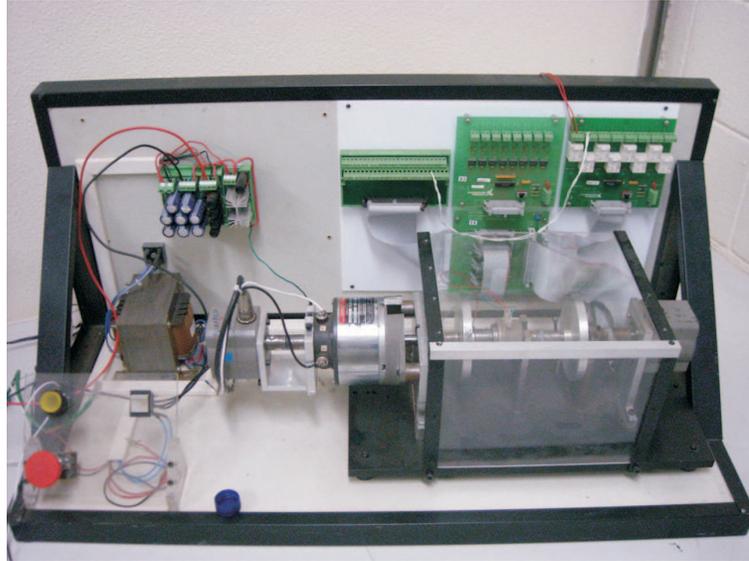


Figura 3.2: Junta robótica com perturbação.

Nas duas subseções seguintes serão tecidas considerações sobre as modelagens cinemática e dinâmica do dispositivo cartesiano, que também são válidas para a junta robótica com perturbação.

### 3.1 Modelo Cinemático

O modelo cinemático direto fornece a posição linear ( $\vec{X}$ ) de uma junta robótica em função da posição angular ( $\vec{\theta}$ ) e a cinemática inversa fornece a posição angular da junta em função da posição linear (ROSÁRIO, 2004).

No dispositivo cartesiano os modelos cinemáticos direto e inverso são simples de determinarem-se, pois há uma estreita relação entre o deslocamento angular do motor e o deslocamento linear do dispositivo. A seguinte equação representa esta relação:

$$\vec{X} = \frac{\alpha}{2\pi} \cdot \vec{\theta}, \quad (3.1)$$

em que  $\alpha$  é uma constante correspondente ao deslocamento linear por revolução do fuso esférico.

## 3.2 Modelo Dinâmico

O modelo dinâmico visa representar a evolução do sistema ao longo do tempo e, também, como os parâmetros do dispositivo influenciam nesta evolução. Além disso, busca-se modelar a maneira como o funcionamento de um eixo do dispositivo interfere no funcionamento dos demais. Estas interferências podem ser mais ou menos acentuadas dependendo do tipo de acoplamento existente entre eixos.

No caso de um dispositivo CNC com  $n$  GDL, as equações do modelo dinâmico são da forma:

$$\tau = J\ddot{q} + B\dot{q} + \tau_r, \quad (3.2)$$

em que  $J$  é o momento de inércia total do sistema,  $B$  o coeficiente de atrito viscoso do sistema,  $\tau_r$  o torque resistente,  $\tau$  o torque do sistema e  $q$  a coordenada generalizada que, neste caso, será a posição angular do motor. Os valores de  $J$  e  $B$  são dependentes tanto dos parâmetros do dispositivo robótico quanto dos parâmetros do motor.

Por sua vez, Altintas (2000) criou um método para calcular o momento de inércia total do sistema levando em consideração a inércia do motor ( $J_m$ ), o momento de inércia refletido no motor devido à mesa e à peça sobre a mesa ( $J_{tw}$ ) e o momento de inércia devido ao fuso esférico ( $J_l$ ). Esta relação é representada pela Equação (3.3). Os valores de  $J_{tw}$  e  $J_l$  são dados nas Equações (3.4) e (3.5), respectivamente. O significado de cada uma das constantes do dispositivo cartesiano CNC é mostrado resumidamente na Tabela 3.1.

$$J = \frac{J_{tw} + J_l}{\eta^2} + J_m \quad (3.3)$$

$$J_{tw} = (m_t + m_w) \left( \frac{h_p}{2\pi} \right)^2 \quad (3.4)$$

$$J_l = \frac{1}{2} m_l \left( \frac{d_p}{2} \right)^2 \quad (3.5)$$

Tabela 3.1: Parâmetros do sistema.

Parâmetro	Descrição
$\eta$	Razão de redução do fuso esférico
$m_t$	Massa da mesa
$m_w$	Valor máximo da massa da peça sobre a mesa
$m_l$	Massa do fuso esférico
$h_p$	Deslocamento por volta do fuso
$d_p$	Diâmetro do fuso

A determinação do parâmetro  $\eta$  é dependente do tipo de redutor utilizado. No caso do fuso de rolamento esférico, é determinado pela Equação (3.6) como demonstrado em (OKWUDIRE; ALTINTAS, 2009).

$$\eta = \frac{2\pi}{d_p} \quad (3.6)$$

### 3.3 Sistema de Acionamento

O acionamento de um dispositivo mecatrônico pode ser elétrico, hidráulico, pneumático ou mecânico. Em uma máquina-ferramenta CNC o acionamento geralmente ocorre sob a forma elétrica, pois, como são dispositivos de alta precisão, este tipo de acionamento permite respostas rápidas e precisas às variações em sua entrada. Para realizar o acionamento da máquina-ferramenta em estudo foi escolhido o motor de corrente de contínua (CC) devido às suas características e à facilidade de controlá-lo.

#### 3.3.1 Motor de Corrente Contínua

Os motores CC são compostos por um estator e um rotor. O estator compõe-se por uma estrutura ferromagnética com pólos salientes aos quais são enroladas as bobinas que formam o campo magnético ou ímã permanente (ERICSSON, 2006). O rotor é um eletroímã constituído por um núcleo de ferro com enrolamentos em sua superfície alimentados por um sistema mecânico de comutação.

As vantagens da utilização de um motor CC decorrem do alto torque na partida e em baixas rotações, da ampla faixa de variação de velocidade, da facilidade em controlar a velocidade, do baixo nível de ruído e da alta resistência a vibrações. Por outro lado as desvantagens aludem à maior necessidade de manutenção e aos motores maiores e mais caros.

O modelamento de um motor CC controlado por corrente de armadura, cuja representação esquemática é dada na Figura 3.3, é feito partindo das equações elétrica, mecânica e de acoplamento eletromecânico clássicas para este motor (OGATA, 1997), que são respectivamente apresentadas na Equação (3.7).

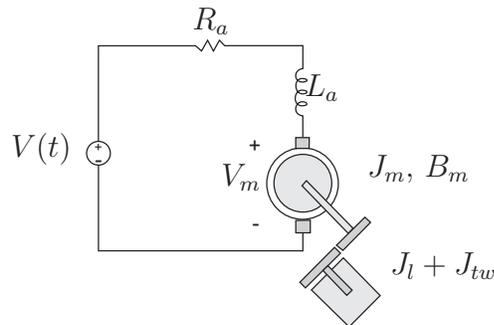


Figura 3.3: Representação esquemática de um motor CC controlado por corrente de armadura.

$$\begin{cases} V(t) = R_a i(t) + L_a \frac{di(t)}{dt} + k_e \omega(t) \\ \tau(t) = J \dot{\omega}(t) + B \omega(t) + \tau_r(t) \\ \tau(t) = k_t i(t) \end{cases} \quad (3.7)$$

A representação em diagrama de blocos do conjunto de equações apresentadas é dada na Figura 3.4, sendo  $s$  a variável de Laplace.

### 3.4 Conclusões

Neste capítulo discorreu-se a respeito dos aspectos de modelagem de uma máquina-ferramenta, considerando tanto a cinemática e a dinâmica do dispositivo quanto o sistema de acionamento utilizado. A modelagem cinemática mostra que há uma estreita relação entre o deslocamento angular do motor e o deslocamento linear causado sobre o eixo do dispositivo. O estudo do sistema de acionamento permitiu melhor compreensão do motor utilizado para

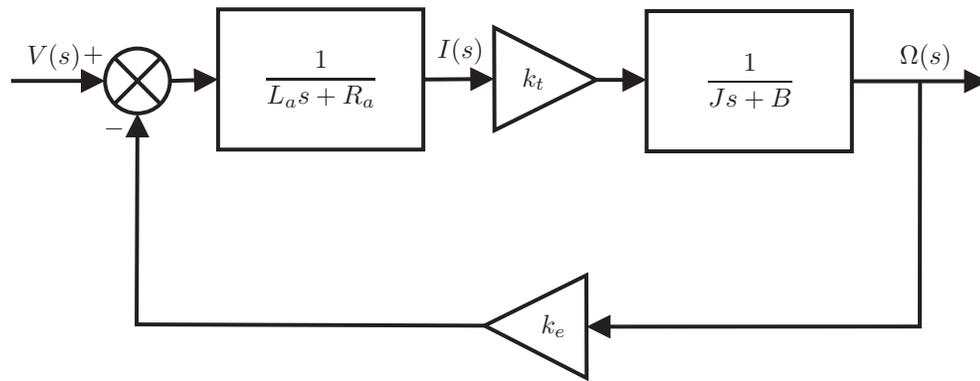


Figura 3.4: Diagrama de blocos do motor de corrente contínua controlado por armadura.

movimentação da máquina-ferramenta, elucidou a influência da estrutura mecânica do dispositivo robótico sobre o funcionamento do motor, além de indicar o que pode ser ajustado para otimizar o desempenho global do sistema.

## 4 Planejamento de Trajetórias

Para o funcionamento adequado do dispositivo cartesiano CNC é necessário que as trajetórias de referência sejam suaves em toda sua extensão e, também, que apresentem derivadas temporais igualmente suaves. Devido às *splines* satisfazerem tais pré-requisitos e por possuírem um baixo custo computacional, neste capítulo será apresentada a generalização do algoritmo OCAS (DU PLESSIS; SNYMAN, 2003) para três dimensões e sua aplicabilidade na síntese de curvas genéricas.

### 4.1 Interpolação utilizando *Splines*

Para solucionar o problema de geração de perfis temporais para cada eixo baseado em um conjunto de pontos de entrada, os seguintes dados devem ser fornecidos:

- Um conjunto de pontos de entrada,  $P_i = (x_i, y_i, z_i)$ ,  $i = 0, \dots, N$ , ao longo de uma curva  $C$  qualquer para ser seguido pela ferramenta de trabalho como ilustrado na Figura 4.1;
- Velocidades tangenciais previamente definidas nos pontos inicial e final;
- Acelerações tangenciais, também, previamente definidas nos pontos inicial e final;
- Derivadas cartesianas nos pontos inicial e final.

Então, é necessário determinar:

- um intervalo de tempo aceitável  $[0, T]$  durante o qual a curva é processada;
- curvas temporais parametrizadas  $X(t)$ ,  $Y(t)$  e  $Z(t)$ ,  $t \in [0, T]$ , que interpolam os pontos nodais  $(x_i, y_i, z_i)$  e;
- curvas temporais paramétricas contínuas para as velocidades  $\dot{X}(t)$ ,  $\dot{Y}(t)$  e  $\dot{Z}(t)$  e para as acelerações  $\ddot{X}(t)$ ,  $\ddot{Y}(t)$  e  $\ddot{Z}(t)$ ,  $t \in [0, T]$ .

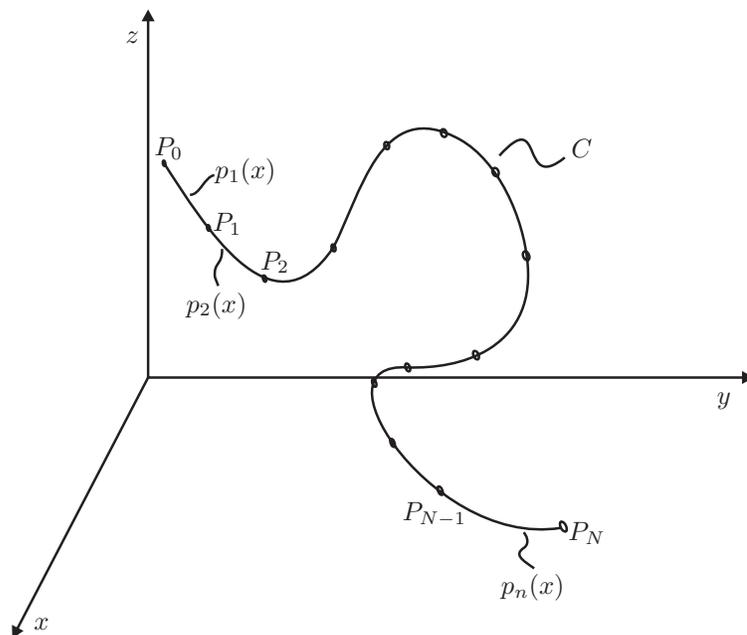


Figura 4.1: Problema básico de interpolação.

A partir dos elementos fornecidos a tarefa inicial é definir o comprimento  $S$  total da curva entre  $P_0$  e  $P_N$  somando os comprimentos dos segmentos  $s_i$  do arco entre  $(x_{i-1}, y_{i-1}, z_{i-1})$  e  $(x_i, y_i, z_i)$ , então, uma aproximação para a interpolação polinomial cúbica  $y(x) \approx p_i(x)$  e  $z(x) \approx q_i(x)$ ,  $x \in [x_{i-1}, x_i]$ , ou outra variável dependendo do que for mais conveniente, deve ser determinada.

Para determinar qual variável deve ser escolhida como parâmetro, pelo menos uma das seguintes condições deve ser satisfeita:

$$x_{i+1} > x_i > x_{i-1} \tag{4.1}$$

$$x_{i+1} < x_i < x_{i-1} \tag{4.2}$$

$$y_{i+1} > y_i > y_{i-1} \tag{4.3}$$

$$y_{i+1} < y_i < y_{i-1} \tag{4.4}$$

$$z_{i+1} > z_i > z_{i-1} \tag{4.5}$$

$$z_{i+1} < z_i < z_{i-1} \tag{4.6}$$

Um problema que pode advir é se nenhuma condição for satisfeita, significando que o conjunto de pontos fornecidos é insuficiente para expressar duas das coordenadas em função da terceira. Esta situação pode ser solucionada especificando pontos adicionais ao longo da curva. Outro inconveniente apresenta-se quando duas condições são satisfeitas; neste caso, a coordenada que será escolhida como variável é aquela que apresentar maior variação absoluta, por exemplo  $|x_{i+1} - x_{i-1}| > |y_{i+1} - y_{i-1}|$  e  $|x_{i+1} - x_{i-1}| > |z_{i+1} - z_{i-1}|$ .

Os polinômios cúbicos interpolados  $p_i(x)$  e  $q_i(x)$  podem ser expressos na forma

$$\begin{aligned} p_i(x) &= a_{y_i} + b_{y_i}(x - x_{i-1}) + c_{y_i}(x - x_{i-1})^2 + d_{y_i}(x - x_{i-1})^3 \\ q_i(x) &= a_{z_i} + b_{z_i}(x - x_{i-1}) + c_{z_i}(x - x_{i-1})^2 + d_{z_i}(x - x_{i-1})^3 \end{aligned}, \quad x \in [x_{i-1}, x_{i+1}]. \quad (4.7)$$

Os quatro coeficientes de cada equação podem ser unicamente determinados usando o gradiente no ponto  $P_{i-1}$  e os três pontos nodais de cada tripla, implicando nas seguintes expressões para cada coeficiente:

$$\begin{aligned} a_{y_i} &= y_{i-1}, \quad a_{z_i} = z_{i-1} \\ b_{y_i} &= \frac{d}{dx}p_i(x_{i-1}) \approx \left. \frac{dy}{dx} \right|_{P_{i-1}}, \quad b_{z_i} = \frac{d}{dx}q_i(x_{i-1}) \approx \left. \frac{dz}{dx} \right|_{P_{i-1}} \\ c_{y_i} &= e_{y_i} - d_{y_i}(x_i - x_{i-1}), \quad c_{z_i} = e_{z_i} - d_{z_i}(x_i - x_{i-1}) \end{aligned} \quad (4.8)$$

$$\begin{aligned} d_{y_i} &= \frac{y_{i+1} - a_{y_i} - b_{y_i}(x_{i+1} - x_{i-1}) - e_{y_i}(x_{i+1} - x_{i-1})^2}{(x_{i+1} - x_{i-1})^3 - (x_{i+1} - x_{i-1})(x_i - x_{i-1})^2}, \\ d_{z_i} &= \frac{z_{i+1} - a_{z_i} - b_{z_i}(x_{i+1} - x_{i-1}) - e_{z_i}(x_{i+1} - x_{i-1})^2}{(x_{i+1} - x_{i-1})^3 - (x_{i+1} - x_{i-1})(x_i - x_{i-1})^2} \end{aligned}$$

com,

$$\begin{aligned} e_{y_i} &= \frac{y_i - a_{y_i} - b_{y_i}(x_i - x_{i-1})}{(x_i - x_{i-1})^2}, \\ e_{z_i} &= \frac{z_i - a_{z_i} - b_{z_i}(x_i - x_{i-1})}{(x_i - x_{i-1})^2}. \end{aligned} \quad (4.9)$$

Os valores dos gradientes nos pontos inicial e final são conhecidos e fornecidos como entradas, mas, na prática, estes valores nem sempre estão disponíveis, de forma que devem ser estimados de alguma forma. Para aproximar adequadamente os valores dos gradientes alguns pontos adicionais devem ser inseridos entre os pontos  $P_0$  e  $P_1$ , e  $P_{N-1}$  e  $P_N$  como mostrado na Figura 4.2.

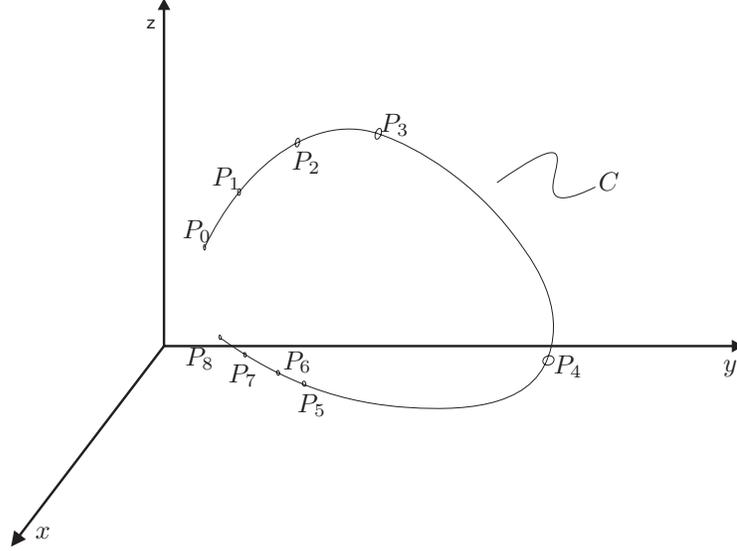


Figura 4.2: Pontos nodais adicionais.

Após a adição dos pontos os valores dos gradientes  $\partial y/\partial x$  e  $\partial z/\partial x$  no ponto  $P_0$  podem ser determinados resolvendo os sistemas de equações dados nas Expressões (4.10) e (4.11), respectivamente.

$$\begin{bmatrix} y_1 - y_0 \\ y_2 - y_0 \\ y_3 - y_0 \end{bmatrix} = \begin{bmatrix} \Delta x_1 & \frac{(\Delta x_1)^2}{2} & \frac{(\Delta x_1)^3}{6} \\ \Delta x_2 & \frac{(\Delta x_2)^2}{2} & \frac{(\Delta x_2)^3}{6} \\ \Delta x_3 & \frac{(\Delta x_3)^2}{2} & \frac{(\Delta x_3)^3}{6} \end{bmatrix} \begin{bmatrix} y'_0 \\ y''_0 \\ y'''_0 \end{bmatrix} \quad (4.10)$$

$$\begin{bmatrix} z_1 - z_0 \\ z_2 - z_0 \\ z_3 - z_0 \end{bmatrix} = \begin{bmatrix} \Delta x_1 & \frac{(\Delta x_1)^2}{2} & \frac{(\Delta x_1)^3}{6} \\ \Delta x_2 & \frac{(\Delta x_2)^2}{2} & \frac{(\Delta x_2)^3}{6} \\ \Delta x_3 & \frac{(\Delta x_3)^2}{2} & \frac{(\Delta x_3)^3}{6} \end{bmatrix} \begin{bmatrix} z'_0 \\ z''_0 \\ z'''_0 \end{bmatrix} \quad (4.11)$$

com  $\Delta x_i = x_i - x_0$ ,  $i = 1, 2, 3$ . Os valores dos gradientes  $y'_N$  e  $z'_N$  em  $P_N$  podem ser determinados da mesma forma. Se  $\partial x/\partial y$  ou  $\partial x/\partial z$  ou  $\partial z/\partial y$  em qualquer ponto forem necessários, eles podem ser obtidos do mesmo modo.

Após determinar as aproximações polinomiais  $p_i(x)$  e  $q_i(x)$  para serem usadas no intervalo  $[x_{i-1}, x_i]$ , o comprimento da curva  $s_i$  correspondente pode ser obtido via integração numérica usando a regra de Simpson composta para  $n$  subintervalos (BURDEN; FAIRES, 2001). Uma vez realizada a integração, o comprimento total  $S$  da trajetória é dado por

$$S = \sum_{i=1}^N s_i. \quad (4.12)$$

Assume-se que a distância  $s$  ao longo da curva possa ser representada por um polinômio cúbico no tempo  $t$  dado pela Equação (4.13). Esta expressão é associada com um dado comprimento inicial  $s_0$ , velocidade tangencial inicial  $v_0$  e o comprimento final da curva  $S$ , Equação (4.12), a velocidade tangencial ao longo de toda a trajetória é dada pela Equação (4.14) e a aceleração pela Equação (4.15), respectivamente.

$$s(t) = a_s + b_s t + c_s t^2 + d_s t^3, \quad t \in [0, T] \quad (4.13)$$

$$\dot{s}(t) = b_s + 2c_s t + 3d_s t^2, \quad t \in [0, T] \quad (4.14)$$

$$\ddot{s}(t) = 2c_s + 6d_s t, \quad t \in [0, T] \quad (4.15)$$

No tempo  $T$  o valor da velocidade tangencial deve atingir  $v^*$  ( $\dot{s}(T) = v^*$ ), a velocidade máxima previamente especificada, e o valor da aceleração deve ser zero ( $\ddot{s}(T) = 0$ ). Substituindo todas as condições das Equações (4.13), (4.14) e (4.15) e após algumas simplificações os valores de  $a_s$ ,  $b_s$ ,  $c_s$ ,  $d_s$  e  $T$  são dados por:

$$\begin{aligned}
a_s &= s_0, & b_s &= v_0, \\
T &= \frac{3(S - s_0)}{v_0 + 2v^*}, \\
d_s &= \frac{v_0 - v^*}{3T^2}, & c_s &= -3d_s T.
\end{aligned} \tag{4.16}$$

O comprimento do caminho até o ponto nodal  $i \in [1, N - 1]$  é dado por

$$S_i = \sum_{j=1}^i s_j. \tag{4.17}$$

Para determinar os tempos nodais  $t_i$  correspondentes aos pontos nodais  $P_i$  é necessária a resolução de um sistema de equações não-lineares, dado na Equação (4.18). É utilizado o método iterativo de Newton-Raphson (BURDEN; FAIRES, 2001) para atingirem-se valores adequados para cada um dos tempos.

$$s(t_i) - S_i = a_s + b_s t_i + c_s t_i^2 + d_s t_i^3 - S_i = 0 \tag{4.18}$$

Finalmente, com todos os tempos nodais  $t_i$  determinados, as representações por *splines* cúbicas de  $X(t)$ ,  $Y(t)$  e  $Z(t)$ , Equação (4.19), podem ser determinadas. O ponto principal para calcular a representação por *splines* cúbicas é que as derivadas temporais nos pontos extremos ( $\dot{x}_0$ ,  $\dot{x}_N$ ,  $\dot{y}_0$ ,  $\dot{y}_N$ ,  $\dot{z}_0$  e  $\dot{z}_N$ ) devem ser conhecidas.

$$\begin{aligned}
X_i(t) &= a_{X_i} + b_{X_i}(t - t_i) + c_{X_i}(t - t_i)^2 + d_{X_i}(t - t_i)^3, \\
Y_i(t) &= a_{Y_i} + b_{Y_i}(t - t_i) + c_{Y_i}(t - t_i)^2 + d_{Y_i}(t - t_i)^3, & i \in [0, N - 1] \\
Z_i(t) &= a_{Z_i} + b_{Z_i}(t - t_i) + c_{Z_i}(t - t_i)^2 + d_{Z_i}(t - t_i)^3,
\end{aligned} \tag{4.19}$$

Se as derivadas forem conhecidas explicitamente, o Algoritmo *Splines Cúbicas com Restrição* apresentado em (BURDEN; FAIRES, 2001) e resumido no Anexo A.1 pode ser aplicado para determinar as *splines* cúbicas; caso contrário, as derivadas temporais podem ser precisamente aproximadas usando expansão em série de Taylor para cada eixo e cada ponto extremo

(DU PLESSIS; SNYMAN, 2003) como apresentado na Equação (4.20).

$$\begin{bmatrix} x_1 - x_0 \\ x_2 - x_0 \\ x_3 - x_0 \\ x_4 - x_0 \end{bmatrix} = \begin{bmatrix} \Delta t_1 & \frac{(\Delta t_1)^2}{2} & \frac{(\Delta t_1)^3}{6} & \frac{(\Delta t_1)^4}{24} \\ \Delta t_2 & \frac{(\Delta t_2)^2}{2} & \frac{(\Delta t_2)^3}{6} & \frac{(\Delta t_2)^4}{24} \\ \Delta t_3 & \frac{(\Delta t_3)^2}{2} & \frac{(\Delta t_3)^3}{6} & \frac{(\Delta t_3)^4}{24} \\ \Delta t_4 & \frac{(\Delta t_4)^2}{2} & \frac{(\Delta t_4)^3}{6} & \frac{(\Delta t_4)^4}{24} \end{bmatrix} \begin{bmatrix} x'_0 \\ x''_0 \\ x'''_0 \\ x_0^{(4)} \end{bmatrix} \quad (4.20)$$

$$\Delta t_i = t_i - t_0, \quad i = 1, 2, 3, 4$$

#### 4.1.1 Síntese de Curvas Genéricas

Quando se lida com curvas reais, como dado na Figura 4.3, o movimento deve ser suave por toda a trajetória, além disso, o movimento deve partir do e chegar ao repouso. No caso de uma trajetória genérica, esta pode ser segmentada em três regiões básicas: aceleração tangencial ( $C_I$ ), velocidade tangencial constante ( $C_{II}$ ) e desaceleração tangencial ( $C_{III}$ ), os pontos  $N_I$ ,  $N_{II}$  e  $N_{III}$  delimitam cada uma dessas regiões.

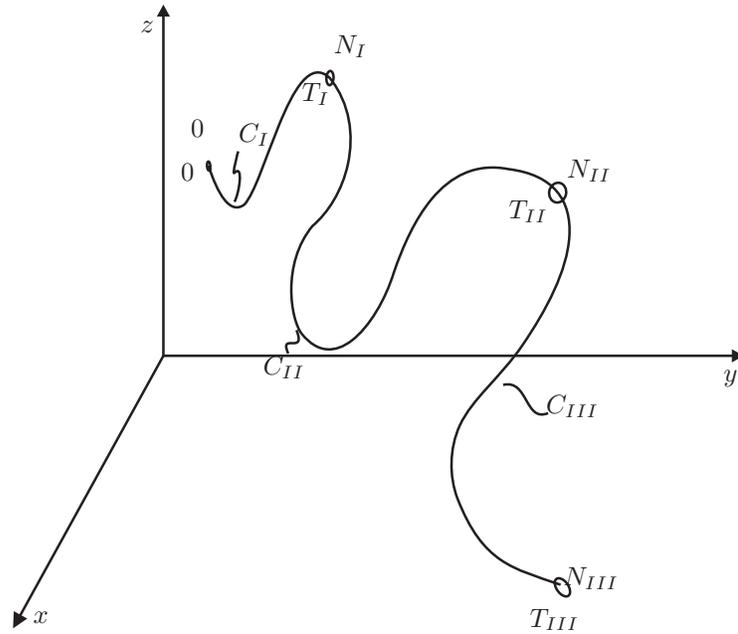


Figura 4.3: Curva genérica na qual a velocidade máxima pré-estabelecida é atingida.

A metodologia brevemente descrita aqui para a síntese de curvas reais foi proposta por DU PLESSIS; SNYMAN (2003), na qual a determinação dos pontos  $N_I$  e  $N_{II}$  é realizada baseada no valor máximo da velocidade tangencial  $v^*$  e na máxima aceleração tangencial permitida  $\ddot{s}_{allow}$ .

Antes de esclarecer o procedimento para determinar os pontos  $N_I$  e  $N_{II}$ , as expressões para os comprimentos de cada região da curva ( $S_I$ ,  $S_{II}$  e  $S_{III}$ ) serão determinadas. A curva de aceleração  $C_I$  corresponde à situação previamente demonstrada, em que um incremento gradual na velocidade é necessário durante o intervalo de tempo  $t \in [0, T_I]$  de forma a atingir a velocidade máxima desejada  $v^*$  no tempo  $T_I$ . A distância  $s_I$  ao longo da curva  $C_I$  é dada por

$$s_I(t) = a_{s_I} + b_{s_I}t + c_{s_I}t^2 + d_{s_I}t^3, \quad t \in [0, T_I]. \quad (4.21)$$

As condições de contorno são

$$\begin{aligned} s_I(0) = s_0, \quad s_I(T_I) = S_I, \quad \dot{s}_I(0) = v_0 \\ \dot{s}_I(T_I) = v^* \quad \text{e} \quad \ddot{s}_I(T_I) = 0. \end{aligned} \quad (4.22)$$

Os coeficientes  $a_{s_I}$ ,  $b_{s_I}$ ,  $c_{s_I}$ ,  $d_{s_I}$  e  $T_{s_I}$  podem ser determinados utilizando o procedimento previamente descrito; e os tempos nodais  $t_i$ ,  $i = 1, 2, \dots, N_I$ , podem ser obtidos empregando o método de Newton.

A distância  $s_{II}$  ao longo da curva de velocidade constante  $C_{II}$  é dada pela seguinte relação linear:

$$s_{II}(t) = v^*(t - T_I), \quad t \in [T_I, T_{II}]. \quad (4.23)$$

O comprimento total da trajetória  $S_i$ ,  $i = N_I + 1, N_I + 2, \dots, N_{II}$ , do nó  $P_0$  até o nó  $P_i$  pode ser determinado pela Equação (4.12) aplicada ao intervalo específico. Os tempos nodais

$t_i$  correspondentes são dados por

$$t_i = \frac{S_i - S_I}{v^*} + T_I, \quad i = N_I + 1, N_I + 2, \dots, N_{II}. \quad (4.24)$$

A distância  $s_{III}$  ao longo da curva de desaceleração  $C_{III}$  é dada pelo seguinte polinômio cúbico:

$$s_{III}(t) = a_{s_{III}} + b_{s_{III}}(t - T_{II}) + c_{s_{III}}(t - T_{II})^2 + d_{s_{III}}(t - T_{II})^3, \quad t \in [T_{II}, T_{III}], \quad (4.25)$$

cujas condições de contorno, considerando um decréscimo gradual da velocidade, são dadas por:

$$\begin{aligned} s_{III}(T_{II}) &= S_{II}, \quad s_{III}(T_{III}) = S_{III}, \quad \dot{s}_{III}(T_{II}) = v^* \\ \dot{s}_{III}(T_{III}) &= 0 \quad \text{e} \quad \ddot{s}_{III}(T_{II}) = 0. \end{aligned} \quad (4.26)$$

Portanto, os coeficientes  $a_{s_{III}}$ ,  $b_{s_{III}}$ ,  $c_{s_{III}}$ ,  $d_{s_{III}}$  e o tempo final  $T_{s_{III}}$  podem ser determinados por:

$$a_{s_{III}} = S_{II}, \quad b_{s_{III}} = v^*, \quad c_{s_{III}} = 0, \quad d_{s_{III}} = \frac{-v^*}{3(T_{III} - T_{II})^2} \quad (4.27)$$

$$\text{e } T_{III} = \frac{3(S_{III} - S_{II})}{2v^*} + T_{II} \quad (4.28)$$

O método de Newton é empregado mais uma vez para determinar os tempos nodais  $t_i$  correspondentes,  $i = N_{II} + 1, N_{II} + 2, \dots, N_{III}$ , como já explicitado.

Os valores da aceleração e da desaceleração nas curvas  $C_I$  e  $C_{III}$ , respectivamente, não devem ultrapassar a magnitude da aceleração máxima permitida  $\ddot{s}_{allow}$ . Assim, os nós  $N_I$  e

$N_{II}$  devem ser escolhidos de forma que tais restrições de aceleração não sejam violadas:

$$|\ddot{s}_I(0)| \leq \ddot{s}_{allow}, \quad (4.29)$$

$$|\ddot{s}_{III}(T_{III})| \leq \ddot{s}_{allow}. \quad (4.30)$$

Em uma primeira aproximação, supõe-se a existência de um ponto  $N_{mid}$  especial, correspondente ao ponto aproximadamente na metade da trajetória, isto é,  $S_{mid} \approx S_{III}/2$ , e que  $N_I = N_{II} = N_{mid}$ ; implicando que  $T_I = T_{II}$  e, assim,  $C_{II}$  desaparece.

Assumindo que as restrições de aceleração dadas nas Equações (4.29) e (4.30) não sejam violadas em  $N_{mid}$ , o nó  $N_I$  é continuamente modificado, como apresentado no Algoritmo A.2, Anexo A, até que um ponto adequado seja escolhido. O nó  $N_{II}$  é determinado de forma semelhante à empregada para o nó  $N_I$  considerando a parte final da curva.

Se qualquer uma das restrições for violada, implica que a velocidade desejada  $v^*$  não pode ser atingida, levando a propor uma outra estratégia que garanta que a máxima aceleração não seja excedida, ao custo de não atingir precisamente  $v^*$  e fazendo com que a curva  $C_{II}$ , Figura 4.3, desapareça.

As condições de contorno para a Equação (4.21) devem ser substituídas por

$$\begin{aligned} s_I(0) = s_0, \quad s_I(T_{mid}) = S_{mid}, \quad \dot{s}_I(0) = v_0, \\ \ddot{s}_I(0) = \ddot{s}_{allow} \quad \text{and} \quad \ddot{s}_I(T_{mid}) = 0 \end{aligned} \quad (4.31)$$

que deriva os seguintes coeficientes da *spline*:

$$\begin{aligned} a_{s_I} = S_0, \quad b_{s_I} = v_0, \quad c_{s_I} = \frac{\ddot{s}_{allow}}{2}, \\ d_{s_I} = \frac{-\ddot{s}_{allow}}{6T_{mid}} \quad \text{e} \quad T_{mid} = \sqrt{\frac{3S_{mid}}{\ddot{s}_{allow}}}. \end{aligned} \quad (4.32)$$

Os resultados acima fornecem a velocidade no nó  $N_{mid}$  como

$$v_{mid} = \dot{s}_I(T_{mid}) = \frac{\sqrt{3S_{mid}\ddot{s}_{allow}}}{2}, \quad (4.33)$$

e as condições de contorno para a curva de desaceleração  $C_{III}$ , Equação (4.25), são substituídas por:

$$\begin{aligned} s_{III}(T_{mid}) = S_{mid}, \quad s_{III}(T_{III}) = S_{III}, \quad \dot{s}_{III}(T_{mid}) = v_{mid} \\ \dot{s}_{III}(T_{III}) = 0 \quad \text{e} \quad \ddot{s}_{III}(T_{mid}) = 0. \end{aligned} \quad (4.34)$$

de onde chega-se a

$$a_{s_{III}} = S_{mid}, \quad b_{s_{III}} = v_{mid}, \quad c_{s_{III}} = 0, \quad d_{s_{III}} = \frac{-v_{mid}}{3(T_{III} - T_{mid})^2} \quad (4.35)$$

$$\text{e} \quad T_{III} = \frac{3(S_{III} - S_{mid})}{2v_{mid}} + T_{mid}. \quad (4.36)$$

#### 4.1.2 Avaliação dos Algoritmos

A avaliação do algoritmo implementado realizou-se por meio da utilização de dois tipos de trajetórias distintas, uma apresentando transições abruptas, perfil triangular, e outra com transições suaves, perfil helicoidal, como apresentado na Figura 4.4.

Para ambas as trajetórias foi adotada uma velocidade máxima  $v^* = 20 \text{ mm/s}$  e uma aceleração máxima  $\ddot{s}_{allow} = 8 \text{ mm/s}^2$ , sendo que a trajetória interpolada resultante foi avaliada de forma qualitativa comparando-se o resultado interpolado e a referência, além de analisar os perfis de velocidade resultantes.

A comparação entre a trajetória de referência (azul contínuo) e a trajetória interpolada (vermelho tracejado) por *splines* é apresentada na Figura 4.5, onde se observa, claramente, a

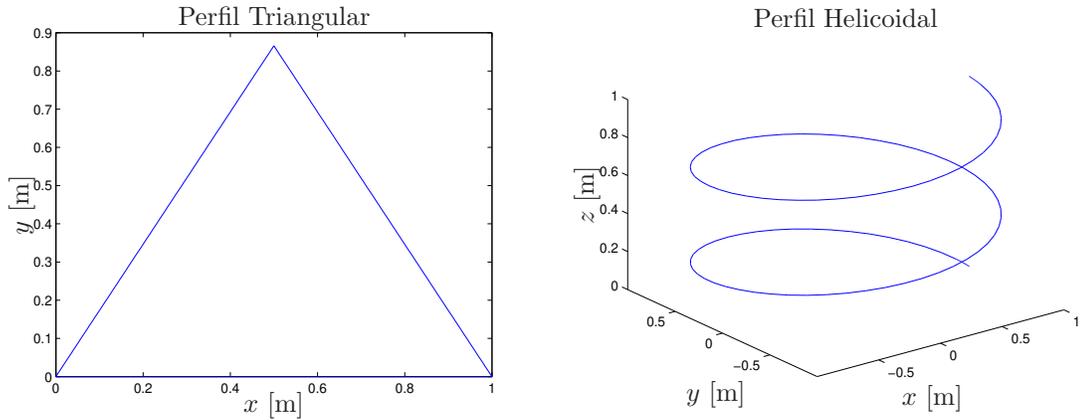


Figura 4.4: Trajetórias de referência utilizadas para interpolação.

boa aproximação entre ambas as curvas indicando que a metodologia implementada mostrou-se adequada para a descrição de trajetórias por meio de *splines* de baixa ordem.

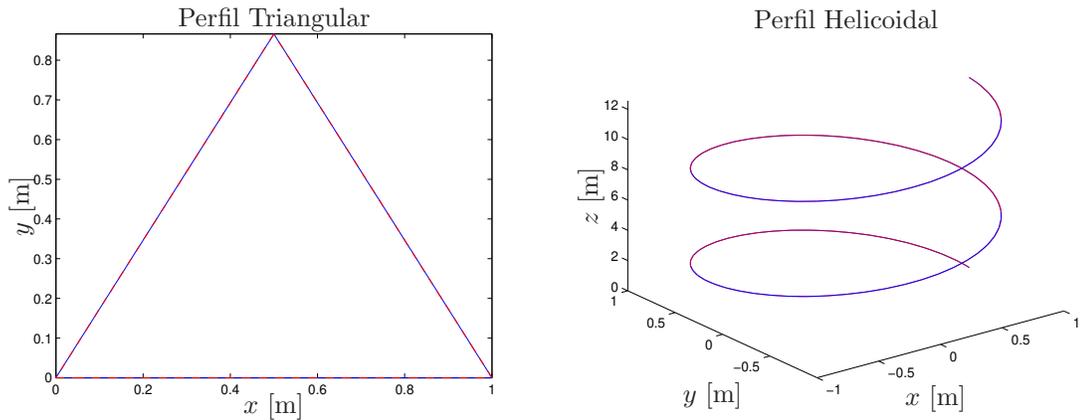
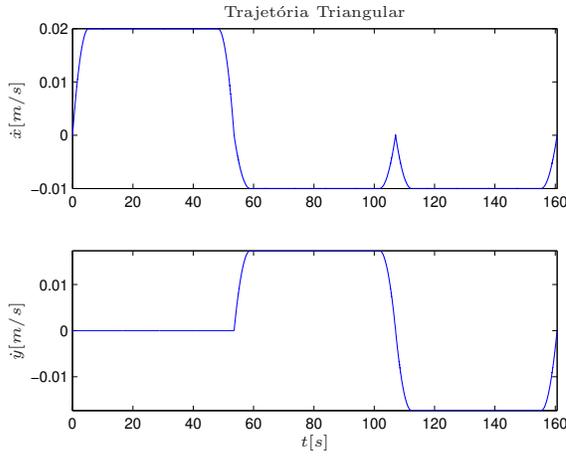


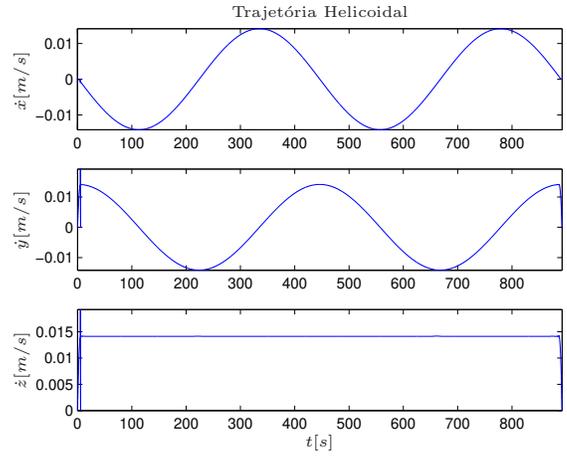
Figura 4.5: Resultado da interpolação utilizando *splines* para diferentes trajetórias de referência.

Os perfis de velocidade, Figura 4.6, para as trajetórias triangular e helicoidal foram analisados, a fim de averiguar a existência de possíveis descontinuidades. Como é possível observar, os perfis gerados são suaves em toda sua extensão, não apresentando nenhuma transição abrupta, garantindo, assim, o correto funcionamento do equipamento a ser conectado.

Os resultados obtidos dos perfis de velocidade são condizentes com curvas de velocidade reais. Assim, se esses perfis forem utilizados pelo sistema de atuação de uma máquina-ferramenta CNC, não haverá problemas de picos de corrente ou tensão, pois como não há descontinuidades nas curvas de velocidade, assegura-se que a trajetória seja corretamente executada sem ocasionar sobrecargas nos controladores ou no sistema de atuação.



(a)



(b)

Figura 4.6: Perfis de velocidade para as trajetórias triangular (a) e helicoidal (b) interpoladas por *splines*.

## 4.2 Conclusões

A geração de movimentos visa garantir boa precisão de rastreamento do caminho de entrada por meio da geração de perfis cinemáticos suaves. O método de interpolação de trajetórias baseado em *splines* garante a continuidade das derivadas temporais da posição cujas ordens sejam menores do que a ordem da *spline* utilizada, tal como foi constatado nos resultados obtidos. O principal motivo de utilizarem-se trajetórias contínuas suaves é garantir que não ocorressem picos de corrente, oriundos de discontinuidades, durante o funcionamento do equipamento, que poderiam causar danos irreparáveis.

## 5 Sistema de Controle

A inserção de um controlador na malha de um sistema visa garantir que a trajetória de referência seja corretamente seguida pelo sistema de acionamento. Alguns sistemas de acionamento, como o motor de passo, são menos dependentes do controlador para executar determinada trajetória do que outros. Vários tipos de controladores vêm sendo desenvolvidos com o intuito de otimizar o rastreamento da referência e diminuir a influência de perturbações externas durante o funcionamento do equipamento, como os controladores preditivos. Apesar disso, técnicas clássicas de controle ainda são muito empregadas por apresentarem bons resultados na maioria das aplicações e por possuírem um vasto ferramental para o projeto de controladores.

### 5.1 Controlador com Alocação de Pólos

Como já mencionado, o controlador PID é largamente utilizado pela sua simplicidade de implementação e por apresentar respostas adequadas para diversos processos. Por outro lado, quando se deseja impor restrições ao sistema de forma que o sistema apresente determinadas características de interesse, como tempo máximo de estabilização do sinal de saída e máxima sobre-elevação em relação à entrada, este controlador não mais supre as necessidades sendo necessário um controlador de maior ordem que permita satisfazê-las. A técnica de alocação de pólos vem de encontro a essa necessidade, pois permite ajustar os pólos do sistema estudado em pontos específicos para que o sistema resultante em malha fechada apresente as características desejadas.

Para o projeto do controlador utilizando alocação de pólos primeiramente define-se a função de transferência em malha fechada do sistema no domínio de Laplace como sendo

$$F(s) = \frac{C(s)G(s)}{1 + C(s)G(s)}, \quad (5.1)$$

sendo  $C(s)$  e  $G(s)$  as funções de transferência do controlador e da planta do sistema em

malha aberta, respectivamente, cujas representações genéricas são dadas nas Equações (5.2) e (5.3). Nestas equações assume-se que  $n_c \geq m_c$  e  $n_g \geq m_g$ .

$$C(s) = \frac{\overline{N}(s)}{\overline{D}(s)} = \frac{\overline{b}_0 + \overline{b}_1 s + \overline{b}_2 s^2 + \dots + \overline{b}_{m_c} s^{m_c}}{\overline{a}_0 + \overline{a}_1 s + \overline{a}_2 s^2 + \dots + \overline{a}_{n_c} s^{n_c}} \quad (5.2)$$

$$G(s) = \frac{N_G(s)}{D_G(s)} = \frac{b_0 + b_1 s + b_2 s^2 + \dots + b_{m_g} s^{m_g}}{a_0 + a_1 s + a_2 s^2 + \dots + a_{n_g} s^{n_g}} \quad (5.3)$$

De posse da região  $\Omega$  previamente determinada, o intuito é determinar o controlador  $C(s)$  que aloque os pólos de  $F(s)$  no interior desta região. A forma mais simples de satisfazer esta condição é definir um polinômio  $P(s)$  de ordem  $n_g + n_c$  cujas raízes encontram-se no interior de  $\Omega$  e resolver a Equação (5.4), conhecida como equação Diofantina, sendo  $\overline{N}(s)$  e  $\overline{D}(s)$  suas incógnitas.

$$D_G(s)\overline{D}(s) + N_G(s)\overline{N}(s) = P(s) \quad (5.4)$$

Uma forma computacionalmente eficiente de resolver a equação Diofantina é escrevê-la sob a forma matricial utilizando matrizes de Sylvester (CHEN, 1999) como a seguir:

$$\mathcal{S}_m \begin{bmatrix} \overline{a}_0 \\ \overline{b}_0 \\ \overline{a}_1 \\ \overline{b}_1 \\ \vdots \\ \overline{a}_{n_c} \\ \overline{b}_{m_c} \end{bmatrix} = \vec{p}, \quad (5.5)$$

sendo  $\vec{p}$  o vetor formado pelos termos do polinômio  $P(s)$  e  $\mathcal{S}_m$  formado pelos coeficientes da planta  $G(s)$  como dado na Equação (5.6).

$$\mathcal{S}_m = \begin{bmatrix} a_0 & b_0 & 0 & 0 & \cdots & 0 & 0 \\ a_1 & b_1 & a_0 & b_0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots \\ a_{n_g} & b_{m_g} & a_{n_g-1} & b_{m_g-1} & \cdots & 0 & 0 \\ 0 & 0 & a_{n_g} & b_{m_g} & \cdots & a_0 & b_0 \\ \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & a_{n_g} & b_{m_g} \end{bmatrix} \quad (5.6)$$

## 5.2 Forma Polinomial RST

O controlador apresentado na Seção 5.1 pressupõe a utilização de elementos analógicos, resistores, capacitores e indutores, para sua implementação; porém, em máquinas-ferramenta CNC, o controlador costuma estar embarcado em placas de lógica reprogramável (FPGAs) ou micro-controladores tornando necessária a discretização deste para que se possa implementá-lo.

A estrutura RST, Equação (5.7), é uma representação do sistema de controle possuindo como parâmetros os sinais de entrada  $r(t)$ , de saída  $y(t)$  e do controlador  $u(t)$ , como mostrado no esquema da Figura 5.1. Do esquema tem-se o par de equações dadas por (5.8) e (5.9).

$$S(q^{-1})\Delta(q^{-1})u(t) = -R(q^{-1})y(t) + T(q^{-1})r(t) \quad (5.7)$$

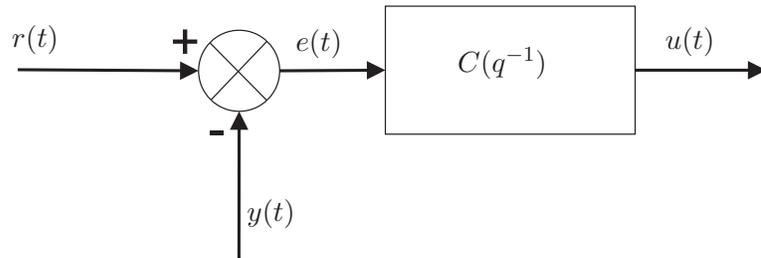


Figura 5.1: Estrutura geral de um controlador discretizado.

$$e(t) = r(t) - y(t) \quad (5.8)$$

$$u(t) = C(q^{-1})e(t) \quad (5.9)$$

Para discretizar o controlador será aplicada a transformação bilinear, Equação (5.10), com tempo de amostragem  $T_s$  sobre a expressão final do controlador obtido em tempo contínuo.

$$s = \frac{2}{T_s} \frac{1 - q^{-1}}{1 + q^{-1}} \quad (5.10)$$

A função de transferência discreta do controlador, tomando a Equação (5.2) como base, é dada por:

$$C(q^{-1}) = \frac{\sum_{j=0}^{m_c} \tilde{b}_j (1 - q^{-1})^j (1 + q^{-1})^{m_c - j}}{\sum_{i=0}^{n_c} \tilde{a}_i (1 - q^{-1})^i (1 + q^{-1})^{n_c - i}}, \quad (5.11)$$

$$\text{com } \tilde{a}_i = \bar{a}_i \left( \frac{2}{T_s} \right)^i \text{ e } \tilde{b}_j = \bar{b}_j \left( \frac{2}{T_s} \right)^j.$$

A partir do conhecimento da função de transferência discreta do controlador  $C(q^{-1})$  é possível representá-lo sob na forma polinomial RST:

$$C(q^{-1}) = \frac{u(t)}{e(t)} = \frac{\sum_{j=0}^{m_c} \tilde{b}_j (1 - q^{-1})^j (1 + q^{-1})^{m_c - j}}{\sum_{i=0}^{n_c} \tilde{a}_i (1 - q^{-1})^i (1 + q^{-1})^{n_c - i}} \Rightarrow$$

$$\left( \sum_{i=0}^{n_c} \tilde{a}_i (1 - q^{-1})^i (1 + q^{-1})^{n_c - i} \right) u(t) = \left( \sum_{j=0}^{m_c} \tilde{b}_j (1 - q^{-1})^j (1 + q^{-1})^{m_c - j} \right) [r(t) - y(t)] \quad (5.12)$$

Comparando a Equação (5.12) com a Equação (5.7), os polinômios  $R(q^{-1})$ ,  $\Delta(q^{-1})S(q^{-1})$  e  $T(q^{-1})$  são dados por:

$$R(q^{-1}) = T(q^{-1}) = \sum_{j=0}^{m_c} \tilde{b}_j (1 - q^{-1})^j (1 + q^{-1})^{m_c - j} \quad (5.13)$$

$$\Delta(q^{-1})S(q^{-1}) = \sum_{i=0}^{n_c} \tilde{a}_i (1 - q^{-1})^i (1 + q^{-1})^{n_c - i}$$

### 5.3 Controlador Preditivo

Apesar dos controladores clássicos serem de fácil sintonização e apresentarem boas respostas quando o modelo do sistema e seus parâmetros são bem conhecidos, sua sensibilidade à variação dos parâmetros é, geralmente, notória e os controladores tendem a apresentar resultados inferiores conforme ocorre a depreciação do equipamento. A implementação de um controle preditivo tem o intuito de garantir menor sensibilidade à variação dos parâmetros e ampliar a vida útil do equipamento.

Como previamente discutido, existem diversos tipos de controladores preditivos na literatura sendo que neste estudo foi abordado o controlador GPC, cujo esquema de funcionamento está ilustrado na Figura 5.2.

O algoritmo GPC considera um modelo do sistema sob a forma CARIMA (*Control-*

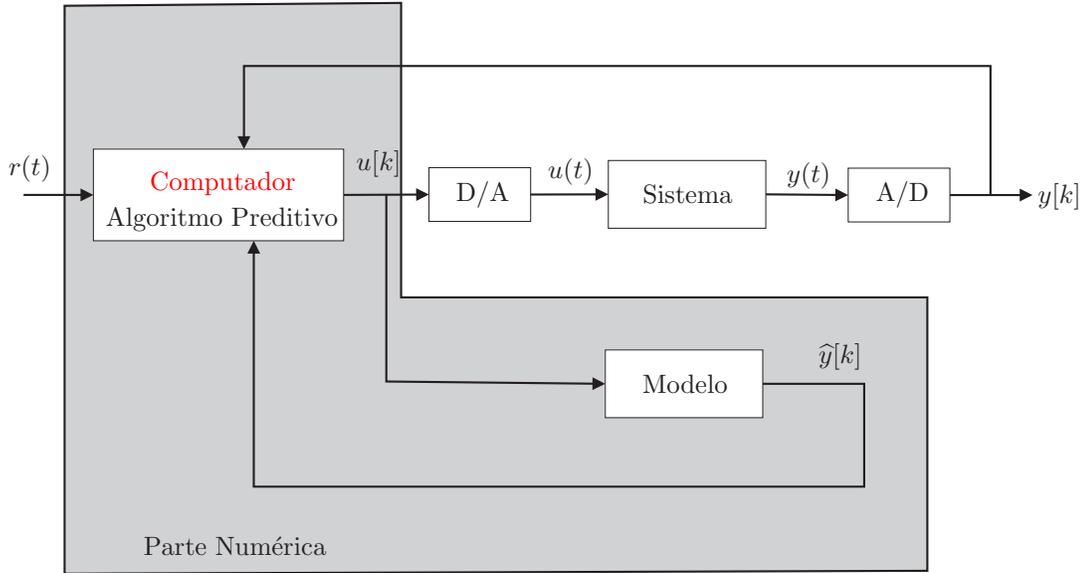


Figura 5.2: Esquema de funcionamento de um controlador preditivo.

*led Auto-Regressive Integrating Moving-Average*), que introduz um integrador na função de transferência e, com isso, anula os erros estáticos comuns em sinais de referência constante, presentes em aplicações industriais. O modelo, dado na Equação (5.14), é obtido a partir da discretização da função de transferência do sistema, quando conhecida, ou ser resultante de um método de identificação de parâmetros.

$$A(q^{-1})y(t) = B(q^{-1})u(t-1) + C(q^{-1})\frac{\xi(t)}{\Delta(q^{-1})} \quad (5.14)$$

Na Equação (5.14)  $u(t)$  e  $y(t)$  correspondem a entrada e a saída do modelo, respectivamente,  $\xi(t)$  modela as perturbações provenientes de um ruído externo,  $\Delta(q^{-1}) = 1 - q^{-1}$  é um operador diferença, os polinômios  $A(q^{-1})$  e  $B(q^{-1})$  contêm os pólos e zeros da função de transferência discreta e as raízes do polinômio  $C(q^{-1})$  são parâmetros de projeto que têm influência na modelagem de perturbações do processo e na robustez do sistema em malha fechada (SOUZA, 2001). Devido a dificuldade de estimar-se um polinômio  $C(q^{-1})$  adequado, considerou-se  $C(q^{-1}) = 1$  (CASSEMIRO, 2006).

O controlador preditivo calcula a sequência dos sinais de controle futuros  $\mathbf{u}$  que minimiza o erro quadrático entre uma sequência de saída desejada  $\mathbf{r}$ , referência de controle, e as saídas previstas  $\hat{\mathbf{y}}$  no horizonte de previsão  $N_2$  sendo que, dos sinais de controle calculados, somente o primeiro é aplicado sobre o sistema. No instante de tempo seguinte todo o procedimento

é repetido, em uma estratégia conhecida como *Receding Horizon*. A sequência de controles futuros ótimos é obtida a partir da minimização de um critério de custo quadrático (CLARKE; MOHTADI; TUFFS, 1987) dado por:

$$J_c = \sum_{j=N_1}^{N_2} (\hat{y}(t+j) - r(t+j))^2 + \lambda \sum_{j=1}^{N_u} \Delta u(t+j-1)^2, \quad (5.15)$$

considerando que  $\Delta u(t+j) = 0$  para  $j \geq N_u$  e sendo:

- $r$  – trajetória de referência,
- $\hat{y}$  – saída predita,
- $\Delta u$  – incrementos do sinal de controle,
- $N_1$  – mínimo horizonte de predição,
- $N_2$  – máximo horizonte de predição,
- $N_2$  – horizonte de predição sobre o controle,
- $\lambda$  – fator de ponderação sobre os incrementos do sinal de controle.

### 5.3.1 Preditor Ótimo

A equação do preditor ótimo que gera as saídas estimadas do modelo pode ser expressa sob a seguinte forma polinomial:

$$\hat{y}(t+j) = \underbrace{F_j(q^{-1})y(t) + H_j(q^{-1})\Delta u(t-1)}_{\text{resposta livre}} + \underbrace{G_j(q^{-1})\Delta u(t+j-1) + J_j(q^{-1})\xi(t+j)}_{\text{resposta forçada}} \quad (5.16)$$

sendo que  $G_j$  representa o futuro,  $F_j$  e  $H_j$  representam o presente e o passado, respectivamente, e  $J_j$  é um termo ligado às perturbações.

Combinando a equação do preditor ótimo com aquela do modelo CARIMA, Equação (5.14), no instante  $t+j$  e igualando os termos que acompanham os sinais de entrada e saída, obtém-se o par de equações Diofantinas dado em (5.17) cujas soluções são os polinômios  $F_j$ ,

$G_j$ ,  $H_j$  e  $J_j$ .

$$\begin{aligned}\Delta(q^{-1})A(q^{-1})J_j(q^{-1}) + q^{-j}F_j(q^{-1}) &= 1 \\ G_j(q^{-1}) + q^{-j}H_j(q^{-1}) &= B(q^{-1})J_j(q^{-1})\end{aligned}\tag{5.17}$$

Ao considerar que a melhor predição do termo ligado às perturbações possui valor médio nulo, o preditor ótimo é definido de forma única pela relação:

$$\hat{y}(t+j) = F_j(q^{-1})y(t) + G_j(q^{-1})\Delta u(t+j-1) + H_j(q^{-1})\Delta u(t-1),\tag{5.18}$$

com  $\text{grau}(G_j(q^{-1})) = \text{grau}(J_j(q^{-1})) = j-1$ ;  $\text{grau}(F_j(q^{-1})) = \text{grau}(A(q^{-1}))$  e  $\text{grau}(H_j(q^{-1})) = \text{grau}(B(q^{-1}))$ .

### 5.3.2 Controlador Preditivo na Forma RST

A equação de predição (5.18) é utilizada dentro dos limites  $N_1$  e  $N_2$ , bem como a equação de custo (5.15). Assim, a equação de predição pode ser escrita na forma matricial

$$\hat{\mathbf{y}} = \mathbf{G}\tilde{\mathbf{u}} + \mathbf{I}\mathbf{F}y(t) + \mathbf{I}\mathbf{H}\Delta u(t-1),\tag{5.19}$$

sendo definidas as seguintes matrizes:

$$\tilde{\mathbf{u}} = [\Delta u(t) \cdots \Delta u(t + N_u - 1)]^T, \quad (5.20)$$

$$\hat{\mathbf{y}} = [\hat{y}(t + N_1) \cdots \hat{y}(t + N_2)]^T, \quad (5.21)$$

$$\mathbf{IF} = [F_{N_1}(q^{-1}) \cdots F_{N_2}(q^{-1})]^T, \quad (5.22)$$

$$\mathbf{IH} = [H_{N_1}(q^{-1}) \cdots H_{N_2}(q^{-1})]^T \text{ e} \quad (5.23)$$

$$\mathbf{G} = \begin{bmatrix} g_{N_1}^{N_1} & g_{N_1-1}^{N_1} & \cdots & 0 \\ g_{N_1+1}^{N_1+1} & g_{N_1}^{N_1+1} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ g_{N_2}^{N_2} & g_{N_2-1}^{N_2} & \cdots & g_{N_2-N_u+1}^{N_2} \end{bmatrix}, \quad (5.24)$$

e os termos  $g_i^j$  correspondem aos coeficientes do polinômio  $G_j$ , que equivalem aos valores  $g_i$  da resposta ao degrau do modelo do sistema.

Da forma semelhante a Equação (5.15) pode ser reescrita em termos matriciais, utilizando a Equação (5.19), como na seguinte relação:

$$\mathbf{J}_c = [\mathbf{G}\tilde{\mathbf{u}} + \mathbf{IF}y(t) + \mathbf{IH}\Delta u(t - 1) - \mathbf{r}]^T [\mathbf{G}\tilde{\mathbf{u}} + \mathbf{IF}y(t) + \mathbf{IH}\Delta u(t - 1) - \mathbf{r}] + \lambda \tilde{\mathbf{u}}^T \tilde{\mathbf{u}}, \quad (5.25)$$

com  $\mathbf{r} = [r(t + N_1) \cdots r(t + N_2)]^T$ .

A minimização do critério de custo, Equação (5.25), que dá-se por derivação analítica, fornece a lei de controle ótima para o sistema:

$$\begin{aligned}
\frac{\partial J_c}{\partial \tilde{\mathbf{u}}} &\equiv 0 \Rightarrow \\
\frac{\partial J_c}{\partial \tilde{\mathbf{u}}} &= 2\mathbf{G}^T \mathbf{G} \tilde{\mathbf{u}} + 2\mathbf{G}^T \alpha + 2\lambda \tilde{\mathbf{u}} = 0 \Rightarrow \\
\frac{\partial J_c}{\partial \tilde{\mathbf{u}}} &= 2[\mathbf{G}^T \mathbf{G} + \lambda \mathbf{I}_{N_u}] \tilde{\mathbf{u}} + 2\mathbf{G}^T \alpha = 0 \Rightarrow \\
\tilde{\mathbf{u}}_{\text{opt}} &= -[\mathbf{G}^T \mathbf{G} + \lambda \mathbf{I}_{N_u}]^{-1} \mathbf{G}^T \alpha = -\mathbf{M} \alpha
\end{aligned} \tag{5.26}$$

sendo que  $\mathbf{I}_{N_u}$  é uma matriz identidade de dimensão  $N_u$  e  $\alpha = \mathbf{I}\mathbf{F}y(t) + \mathbf{I}\mathbf{H}\Delta u(t-1) - \mathbf{r}$  representa os termos ligados ao presente.

Pela estratégia de *receding horizon*, somente o primeiro valor da sequência será utilizado para determinar a ação de controle sobre o sistema. Assim, substituindo o valor de  $\alpha$  e sabendo que  $\mathbf{M}$  é dado por

$$\mathbf{M} = [\mathbf{G}^T \mathbf{G} + \lambda \mathbf{I}_{N_u}]^{-1} \mathbf{G}^T = \begin{bmatrix} m_1 \\ \vdots \\ m_{N_u} \end{bmatrix}, \tag{5.27}$$

a Equação (5.26) pode ser reescrita como

$$\begin{aligned}
\Delta u_{\text{opt}}(t) &= -m_1 [\mathbf{I}\mathbf{F}y(t) + \mathbf{I}\mathbf{H}\Delta u(t-1) - \mathbf{r}] \Rightarrow \\
\Delta u_{\text{opt}}(t) [1 + m_1 \mathbf{I}\mathbf{H}q^{-1}] &= -m_1 \mathbf{I}\mathbf{F}y(t) + m_1 [q^{N_1} \dots q^{N_2}]^T r(t).
\end{aligned} \tag{5.28}$$

Comparando o resultado acima com a Equação (5.7), é possível determinar os polinômios  $R(q^{-1})$ ,  $S(q^{-1})$  e  $T(q)$  do controlador discreto.

$$\begin{aligned}
R(q^{-1}) &= m_1 \mathbf{I}\mathbf{F} \\
S(q^{-1}) &= 1 + m_1 \mathbf{I}\mathbf{H}q^{-1} \\
T(q) &= m_1 [q^{N_1} \dots q^{N_2}]^T
\end{aligned} \tag{5.29}$$

Ao analisar o polinômio  $T(q)$ , nota-se claramente que este é não causal e é responsável pelo aspecto antecipativo do GPC.

## 5.4 Conclusões

Os controladores analisados neste capítulo são sintonizados com base no modelo do sistema ao qual encontram-se conectados, portanto tendem a ter um melhor desempenho do que controladores que não utilizam informações do modelo para sua sintonização.

O que difere nos dois tipos de controladores analisados é o instante de sua resposta. No caso do controlador por alocação de pólos, primeiramente deve ser detectada uma alteração do sinal de entrada para que ele responda com uma ação de controle adequada; já no caso do controlador preditivo, como há um conhecimento prévio sobre a trajetória de referência, ele pode antecipar suas ações de comando minimizando efeitos indesejados de perturbações externas. Além disso, o controlador preditivo permite especificar separadamente os desempenhos de rastreamento e de regulação, o que não é possível com o controlador por alocação de pólos.

Por fim, discutiu-se a forma polinomial RST sob a qual quaisquer controladores podem ser discretizados. Devido a este fato, não há necessidade de se alterar a malha de controle para inserir um ou outro controlador, basta definir os polinômios  $R(q^{-1})$ ,  $S(q^{-1})$  e  $T(q^{-1})$  referentes e substituí-los na malha de controle.

## 6 Teste e Validação das Estratégias Desenvolvidas

Nos capítulos anteriores foram estudados, analisados e discutidos os elementos componentes de uma máquina-ferramenta CNC, individualmente. No presente capítulo é priorizada a análise da interação entre esses elementos e a análise dos resultados obtidos via simulação, tendo a finalidade de verificar se estes ratificam o esperado.

O ambiente virtual proposto para a simulação, desenvolvido em Matlab<sup>®</sup>/Simulink<sup>TM</sup>, inclui os elementos previamente estudados, módulo de geração de movimentos, módulo de controle e módulo mecânico, além de contar com módulo de geração de perturbações e de módulo para análise do erro cometido a fim de se comparar a referência com o resultado obtido. Na Figura 6.1 é ilustrado o ambiente de simulação proposto ressaltando cada um dos módulos citados.

Vale ressaltar que o módulo de controle foi implementado de forma que a substituição de um tipo de controlador por outro fosse simples e que não causasse alterações drásticas na malha de controle. A implementação do módulo de controle de cada eixo pode ser vista na Figura 6.2, onde se faz uso da estrutura RST para controle, de um segurador de ordem zero como conversor D/A do sinal a ser enviado para o equipamento e de um amostrador para conversão A/D do sinal proveniente da máquina.

Ainda, para efetivar a simulação utilizaram-se os parâmetros da máquina-ferramenta CNC e do motor CC que posteriormente foram utilizados para a experimentação, apresentados resumidamente na Tabela 6.1. E, os resultados obtidos para o controlador por alocação de pólos e para o controlador preditivo são arrolados nas Seções 6.1 e 6.2, respectivamente.

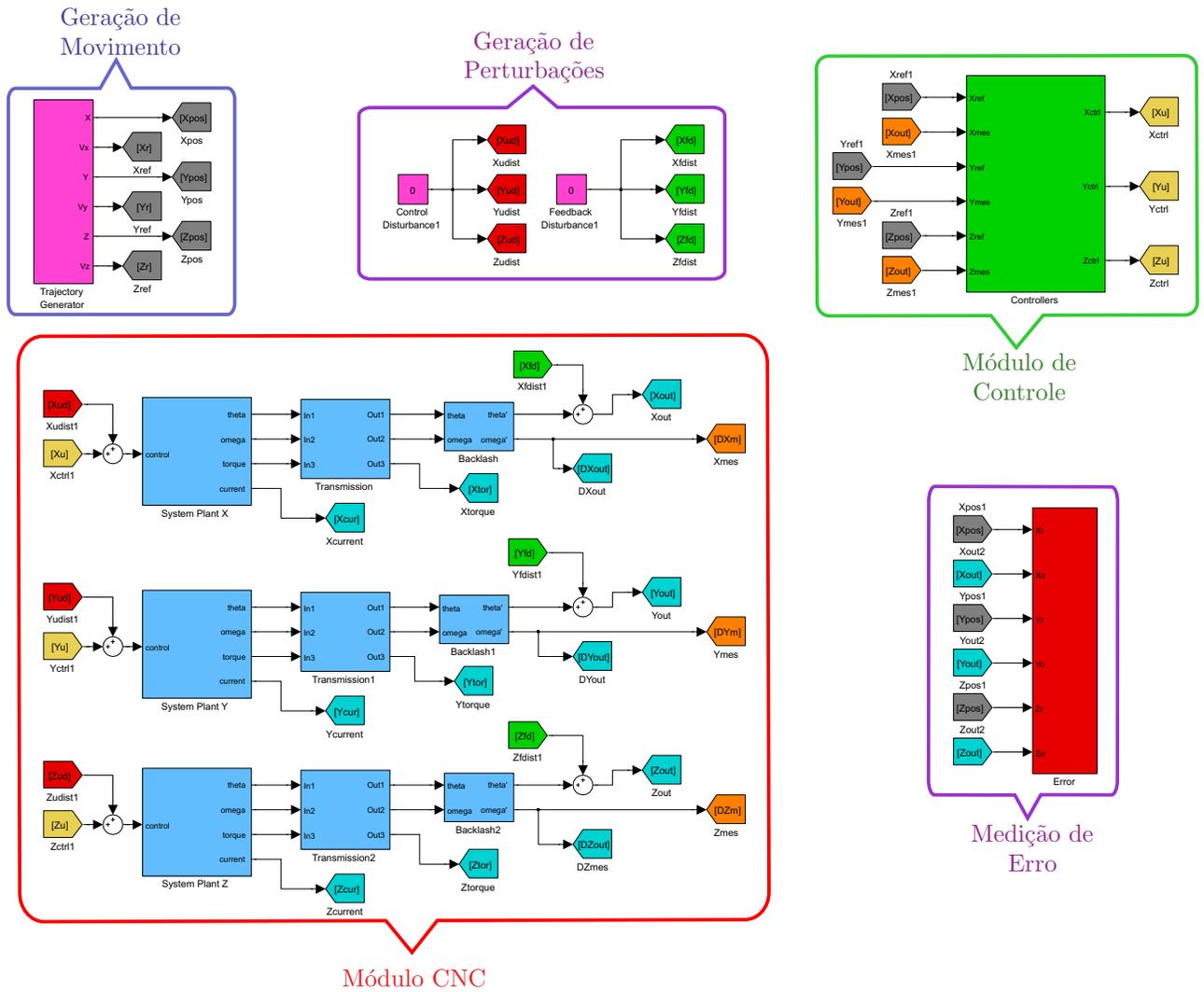


Figura 6.1: Modelo de simulação da máquina-ferramenta CNC de três eixos utilizando motor CC.

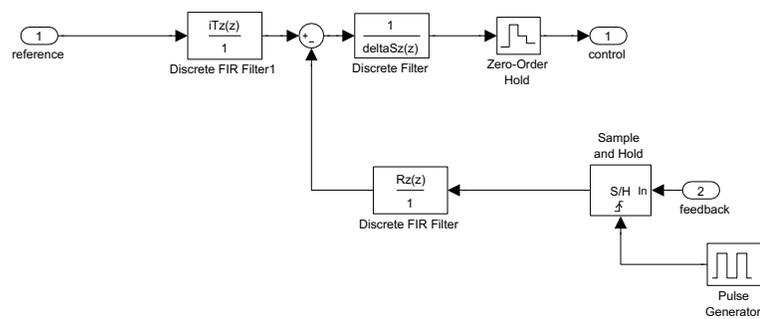


Figura 6.2: Módulo de controle utilizando blocos Simulink<sup>TM</sup>.

Tabela 6.1: Parâmetros do sistema de acionamento e do dispositivo robótico.

Parâmetro	Valor
$\eta$	610.02
$m_t$	5.47 kg
$m_w$	20 kg
$m_l$	0.505 kg
$h_p$	0.005 m/rev
$d_p$	0.0103 m
$V(t)$	24 V
$R$	0.85 $\Omega$
$L$	0.09 mH
$k_t$	$40.957 \cdot 10^{-3}$ Nm/A
$k_e$	$41.062 \cdot 10^{-3}$ V/(rad/s)
$J_m$	$4.802 \cdot 10^{-6}$ kg $\cdot$ m <sup>2</sup>
$B$	$6.743 \cdot 10^{-5}$ N $\cdot$ s/m

## 6.1 Controlador por Alocação de Pólos

O controlador por alocação de pólos foi ajustado de forma que o tempo de estabilização máximo fosse da ordem de 10 ms e que a máxima sobre-elevação em relação à entrada não ultrapassasse 20%. Assim, utilizando as equações apresentadas na Seção 5.1 chegou-se ao controlador dado na Equação (6.1), que conduz à resposta ao degrau e ao lugar das raízes do sistema em malha fechada mostrados na Figura 6.3, na qual é possível se observar que as restrições impostas são atendidas.

$$C(s) = \frac{5.581 \cdot 10^4 s + 3.298 \cdot 10^7}{5.467 \cdot 10^{-5} s^2 + 0.4829 s + 1230} \quad (6.1)$$

Com o controlador determinado, ele é inserido na malha do sistema a fim de averiguar o seu comportamento quando as trajetórias, calculadas no Capítulo 4, são empregadas na entrada do sistema.

A Figura 6.4 mostra a comparação entre entrada e saída do sistema para trajetória triangular de referência, a entrada representada pela cor azul, e a trajetória de saída, em cor vermelha, bem como o erro ponto-a-ponto cometido ao longo da execução. É possível

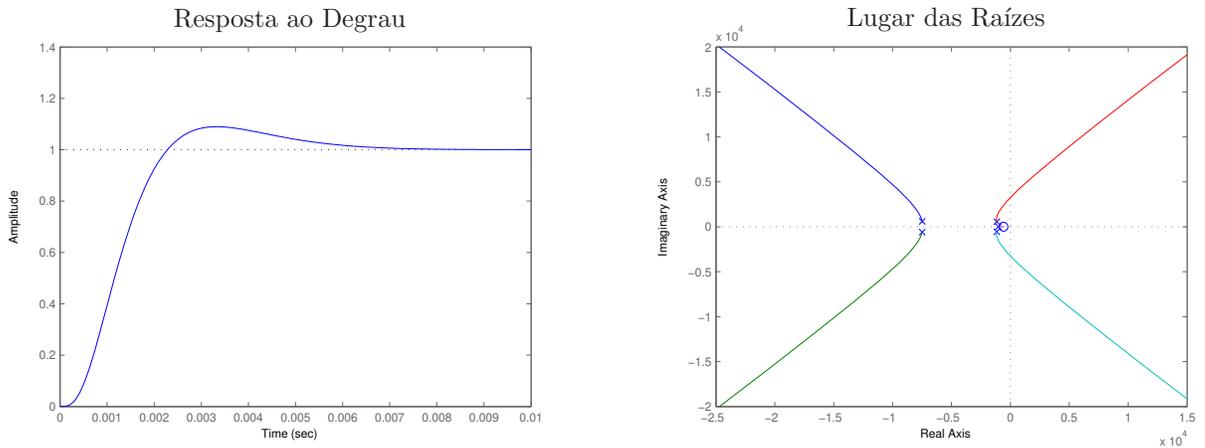


Figura 6.3: Resposta ao degrau unitário e lugar das raízes para o sistema em malha fechada.

observar que o erro cometido é bastante inferior à precisão do equipamento, que é da ordem de décimos de milímetros, demonstrando assim que o controlador atende as necessidades impostas.

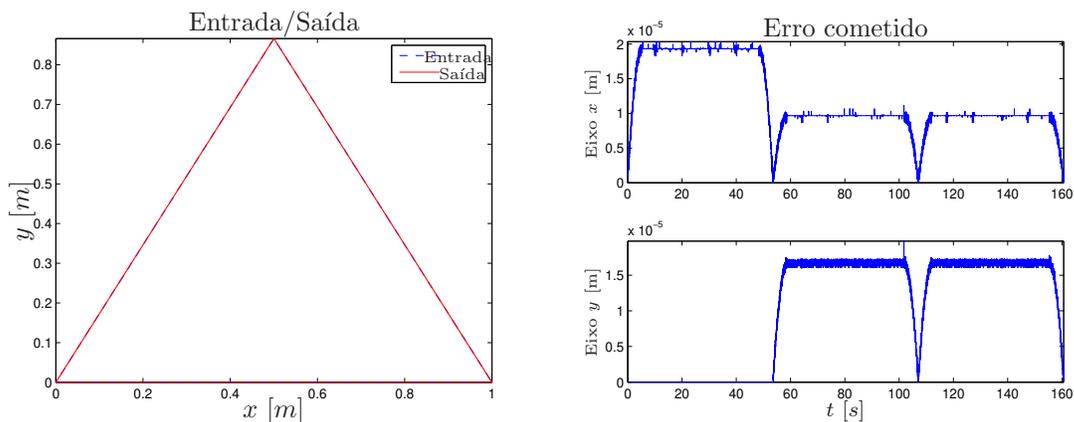


Figura 6.4: Comparação entre entrada e saída do sistema para trajetória triangular e erro cometido ao longo da trajetória.

Com a finalidade de averiguar se todos os parâmetros do motor permanecem dentro dos limites de operação, foram traçadas as curvas de corrente, velocidade e potência, como observado na Figura 6.5. Nesta pode-se apontar que a velocidade atingida pelo motor corresponde àquela previamente determinada e que as curvas de corrente e potência encontram-se dentro dos limites suportados pelo motor garantindo que não haja sobrecargas sobre o equipamento.

O mesmo procedimento empregado para a trajetória triangular foi aplicado para a trajetória helicoidal, tendo como objetivo averiguar o comportamento do controlador para

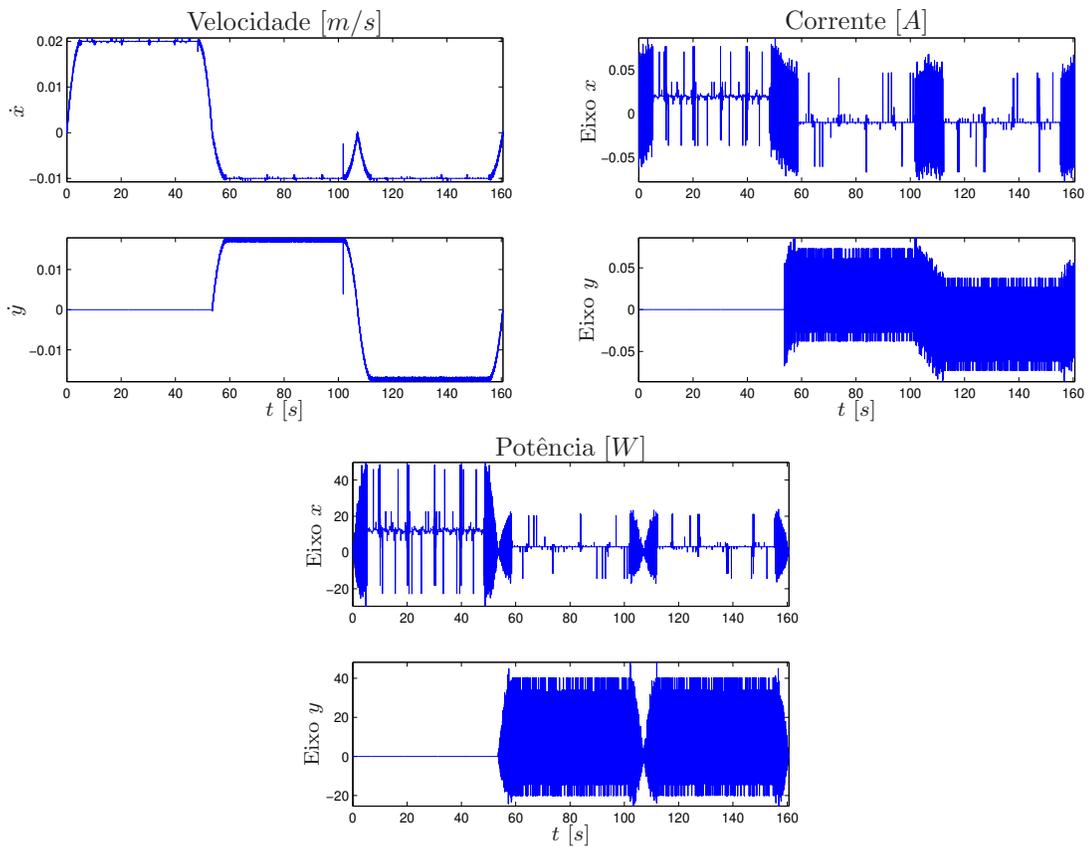


Figura 6.5: Curvas de velocidade, corrente e potência na saída do motor para trajetória triangular.

diferentes tipos de trajetória de entrada. A Figura 6.6 retrata a comparação entre a trajetória de entrada, representada em cor azul, e de saída, em cor vermelha, além do erro cometido durante sua execução. É possível observar a boa aproximação entre as curvas de entrada e saída obtidas, o que é evidenciado pela magnitude do erro cometido.

Curvas de velocidade, corrente e potência no motor foram também traçadas e os seus resultados estão retratados na Figura 6.7. A análise destas curvas comprovam que os valores obtidos encontram-se dentro dos limites do sistema, portanto, garantem o funcionamento adequado da máquina-ferramenta.

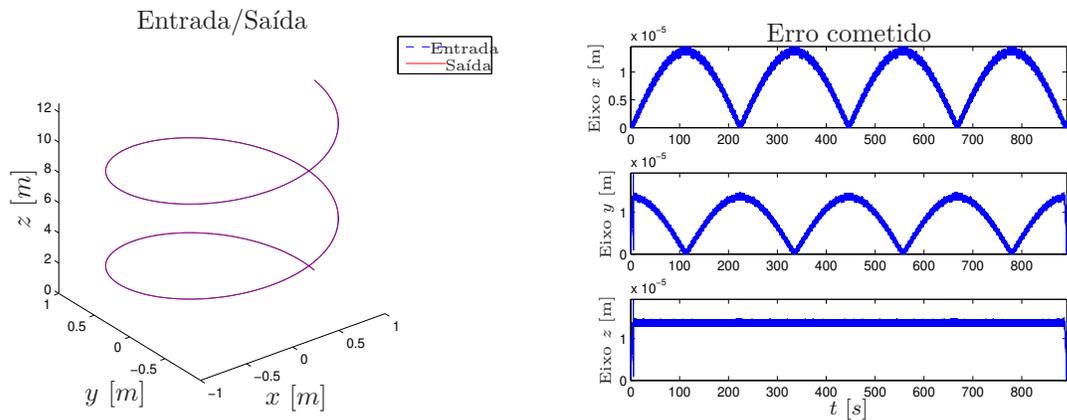


Figura 6.6: Comparação entre entrada e saída do sistema para trajetória helicoidal e erro cometido ao longo da trajetória.

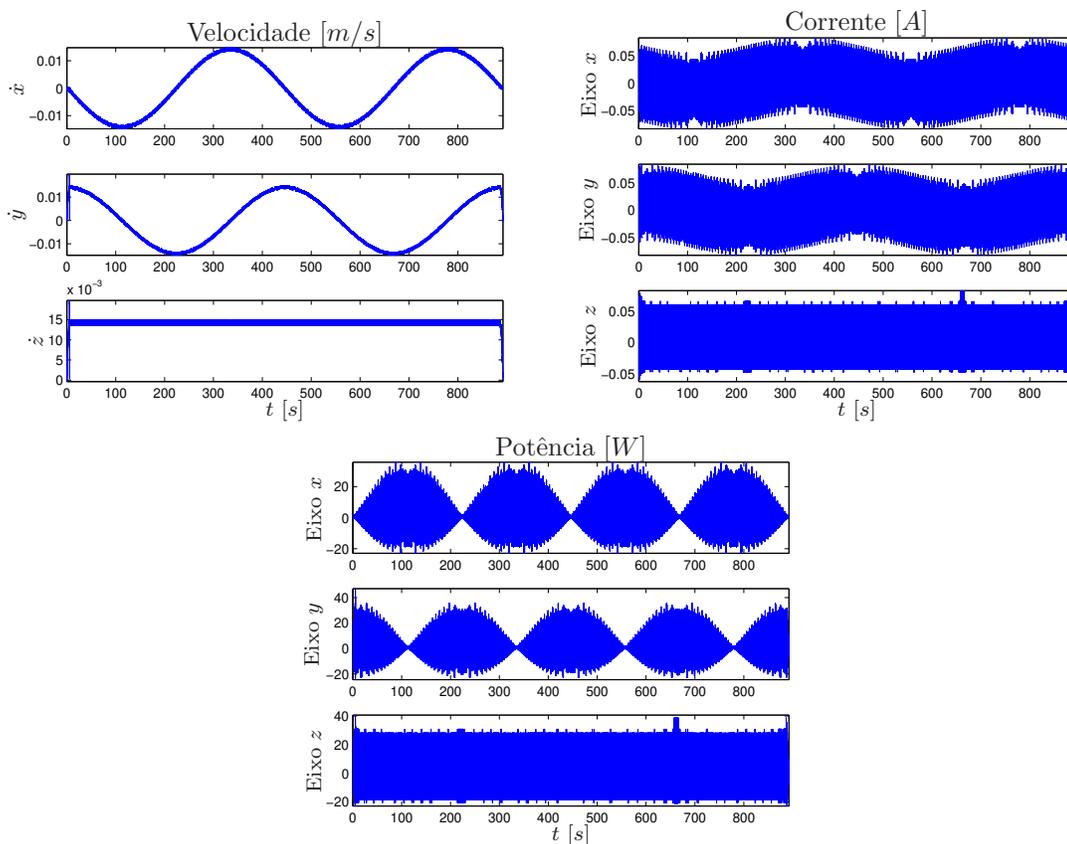


Figura 6.7: Curvas de velocidade, corrente e potência na saída do motor para trajetória triangular.

## 6.2 Controlador Preditivo

O ajuste do controlador preditivo foi realizado definindo-se, primeiramente, os valores dos horizontes mínimo e máximo de predição e o horizonte de controle dados, respectivamente, por  $N_1 = 1$ ,  $N_2 = 10$  e  $N_u = 1$ . A partir destes valores, foi possível determinar o fator de ponderação da ação de controle  $\lambda$  cujo valor é  $2.804 \cdot 10^{-7}$  e, após se efetuar os cálculos expressos na Seção 5.3.2, chega-se aos polinômios  $R(q^{-1})$ ,  $S(q^{-1})$  e  $T(q)$ , dados na Equação (6.2), que determinam o controlador a ser utilizado.

$$\begin{aligned}
 R(q^{-1}) &= 4.618 \cdot 10^4 - 6.909 \cdot 10^4 q^{-1} + 2.539 \cdot 10^4 q^{-2} - 0.0003 \cdot 10^4 q^{-3} \\
 S(q^{-1}) &= 1.000 + 0.310 q^{-1} + 0.006 q^{-2} \\
 T(q) &= \frac{10.821q + 42.287q^2 + 87.216q^3 + 140.761q^4 + 199.819q^5 +}{262.403q^6 + 327.245q^7 + 393.531q^8 + 460.740q^9 + 528.541q^{10}}
 \end{aligned} \tag{6.2}$$

Na Figura 6.8 são demonstradas a resposta ao degrau unitário e os pólos do sistema em malha fechada. Pode-se notar que a resposta do sistema é mais lenta quando comparado ao controlador por alocação de pólos, fato que deve-se ao aspecto antecipativo deste controlador inserir um atraso inerente à quantidade de predições realizadas. Quanto aos pólos do sistema, verifica-se que estes estão situados no interior do círculo unitário que, para um sistema discreto, indica sua estabilidade.

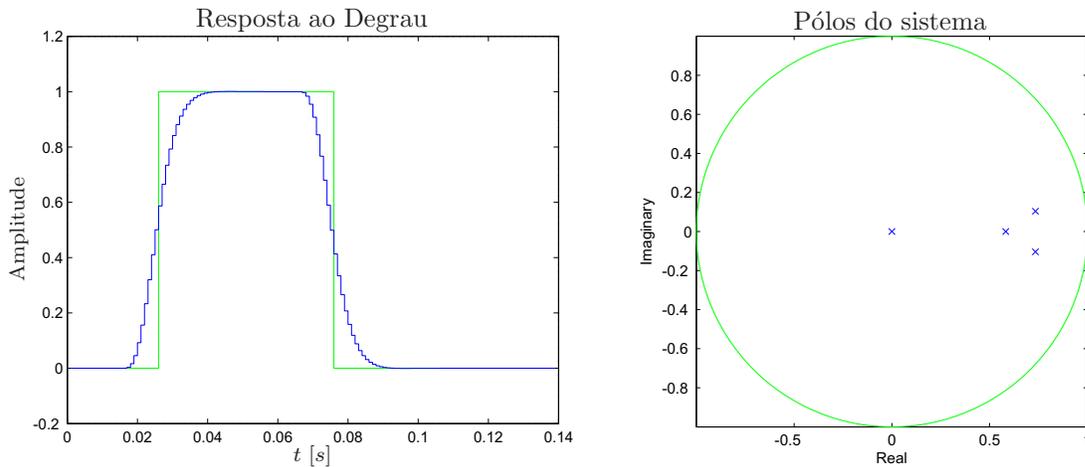


Figura 6.8: Resposta ao degrau unitário e pólos do sistema em malha fechada.

Assim como procedeu-se para o controlador por alocação de pólos, as simulações com as

trajetórias triangular e helicoidal foram aplicadas, também, para o controlador preditivo a fim de estabelecer parâmetros de comparação para o desempenho das duas estratégias propostas.

Na Figura 6.9 apresenta-se a comparação entre entrada e saída do sistema para trajetória triangular de referência, representada em cor azul para entrada e em cor vermelha para saída, bem como o erro ponto-a-ponto cometido ao longo da execução. O desempenho deste controlador pode ser comparado ao do controle por alocação de pólos, visto que a magnitude do erro cometido é de mesma ordem apesar de sua resposta mais lenta.

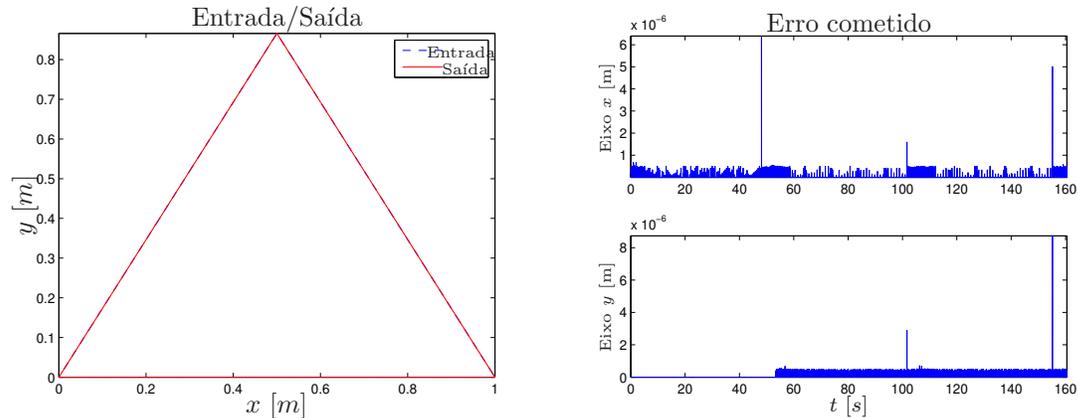


Figura 6.9: Comparação entre entrada e saída do sistema para trajetória triangular e erro cometido ao longo da trajetória.

Curvas de corrente, velocidade e potência do motor, também, foram traçadas e são mostradas na Figura 6.10, onde pode ser observado que tanto a velocidade atingida pelo motor quanto a corrente e a potência encontram-se dentro dos limites tolerados.

No caso da trajetória helicoidal o resultado encontrado, também, foi adequado, uma vez que houve uma boa aproximação entre as curvas de entrada e saída, apresentando um erro muito inferior à precisão da máquina-ferramenta real, como pode ser verificado na Figura 6.11, em que a trajetória de entrada é representada em azul e a de saída em vermelho.

Por fim, as curvas de velocidade, corrente e potência no motor foram traçadas e seus resultados estão sintetizados na Figura 6.12. Esta figura possibilita verificar que a velocidade não excede o limite previamente imposto e as demais curvas encontram-se dentro dos limites do sistema de atuação utilizado.

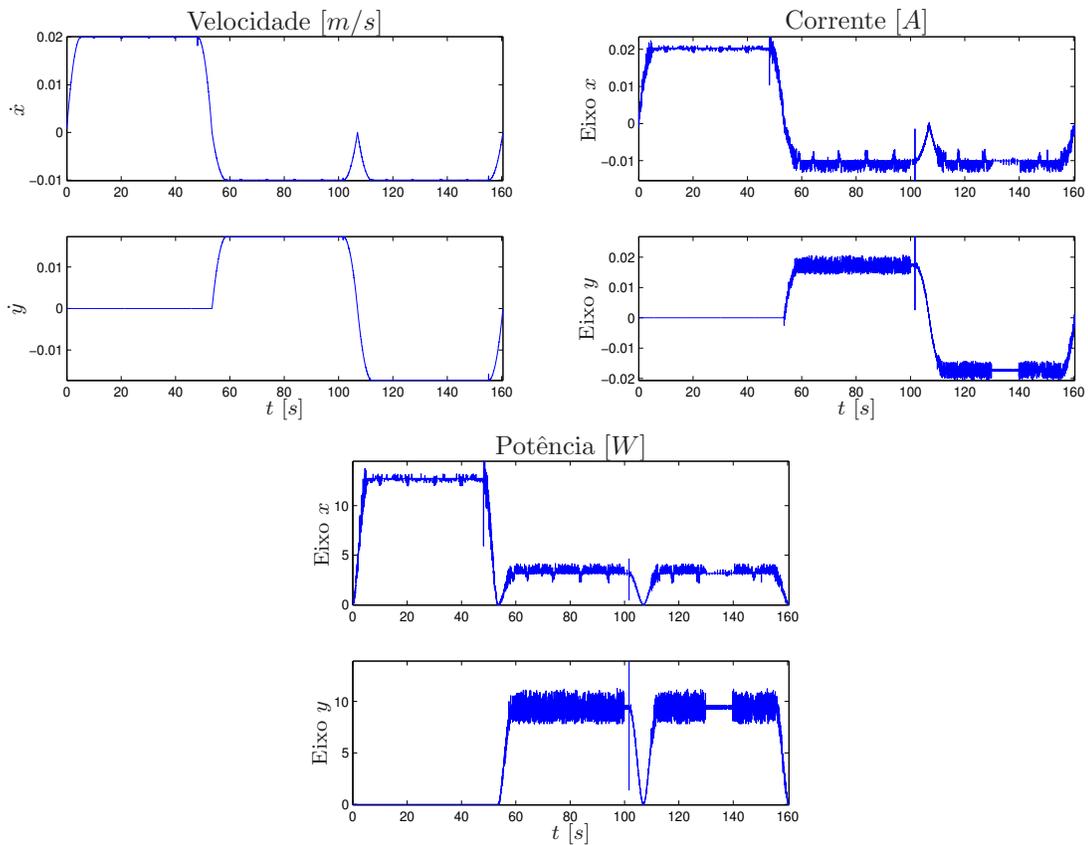


Figura 6.10: Curvas de velocidade, corrente e potência na saída do motor para trajetória triangular.

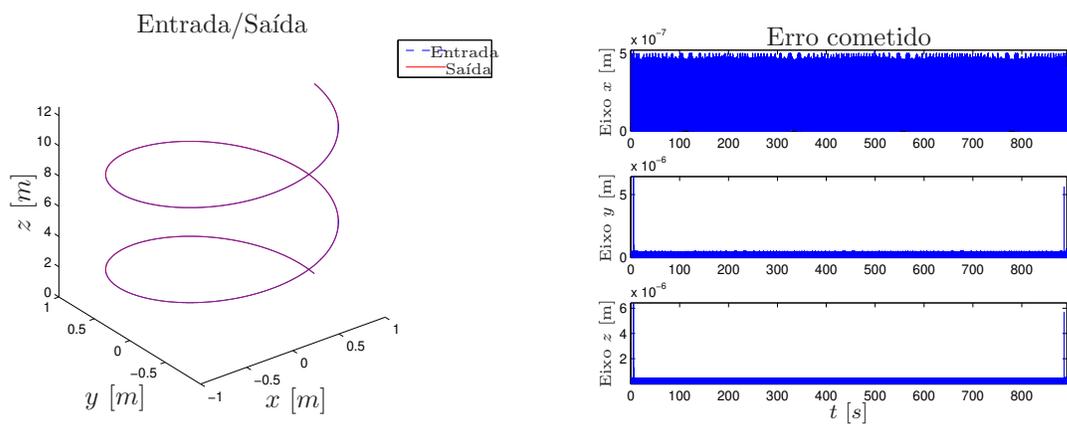


Figura 6.11: Comparação entre entrada e saída do sistema para trajetória helicoidal e erro cometido ao longo da trajetória.

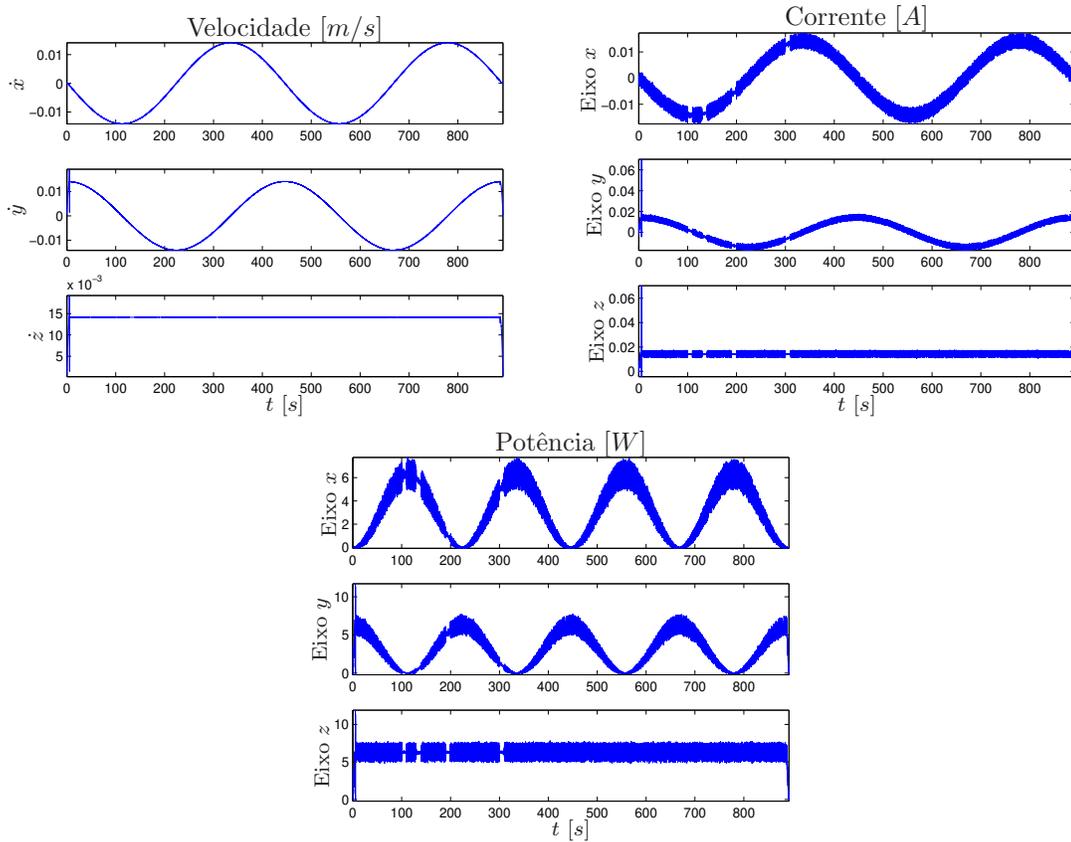


Figura 6.12: Curvas de velocidade, corrente e potência na saída do motor para trajetória triangular.

### 6.3 Influência de Perturbações Externas

Qualquer dispositivo em funcionamento está sujeito a diferentes perturbações, como vibrações, erros de medida e ruído de contato elétrico, que afetam sua operação normal e podem ocasionar divergências no resultado final. Outra função pertinente ao controle é buscar minimizar a influência destas perturbações tornando mais robusto o funcionamento do equipamento além de diminuir as fontes de erro. Devido a este fato, nesta seção, perturbações foram inseridas em dois pontos da malha de controle, no sinal de controle enviado ao dispositivo robótico e no sinal de realimentação do sistema, para analisar sua influência sobre a saída da máquina-ferramenta; para tanto utilizou-se o perfil helicoidal como referência para esta análise.

Inicialmente, um ruído colorido cuja amplitude é da ordem de  $10^{-1}$  foi adicionado ao sinal de controle durante um determinado período de tempo, dez segundos, em diferentes

pontos da trajetória concomitantemente ou separadamente nos eixos do dispositivo e a resposta de cada um dos controladores foi analisada com a finalidade de averiguar sua robustez na rejeição a perturbações.

Como pode ser observado nas Figuras 6.13 e 6.14 a presença de uma perturbação sobre o sinal de controle exerce pouca influência sobre o resultado final, uma vez que ao se efetuar a leitura do sinal de saída, a existência da perturbação foi notada e o controlador exerceu sua ação corretiva minimizando a influência de tal perturbação, além de garantir que o resultado obtido ratificou o desejado.

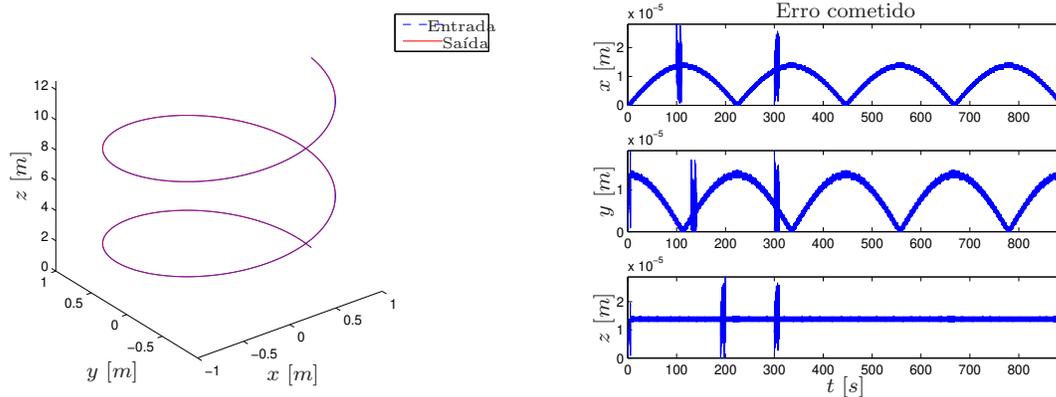


Figura 6.13: Influência de perturbações sobre o sinal de controle utilizando controlador por alocação de pólos.

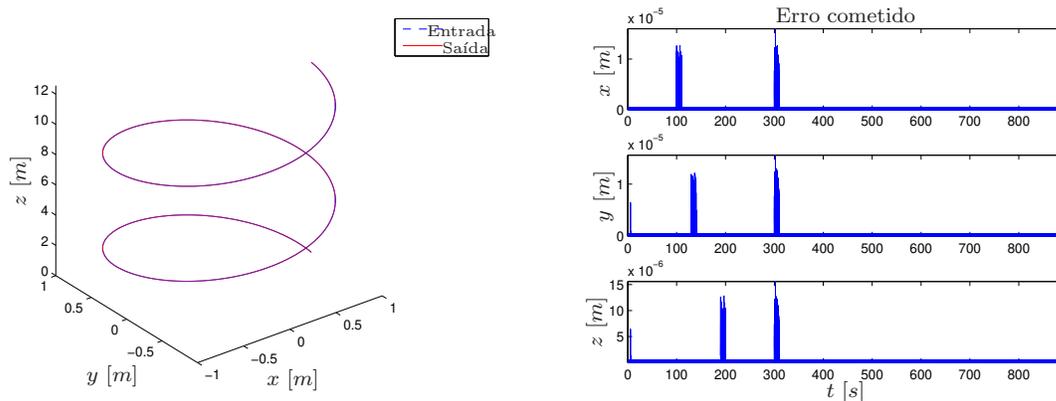


Figura 6.14: Influência de perturbações sobre o sinal de controle utilizando controlador preditivo.

O mesmo procedimento foi aplicado ao sinal de realimentação com o objetivo de averiguar em qual ponto a perturbação torna-se mais danosa ao sistema. Assim, nas Figuras 6.15 e 6.16 são retratados os resultados para os controladores por alocação de pólos e preditivo, respectivamente, quando há perturbações na realimentação do sistema.

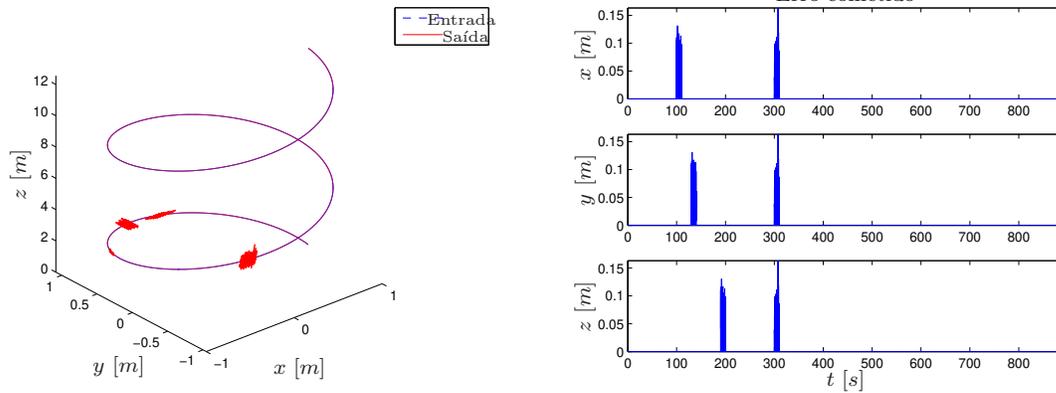


Figura 6.15: Influência de perturbações sobre o sinal de controle utilizando controlador por alocação de pólos.

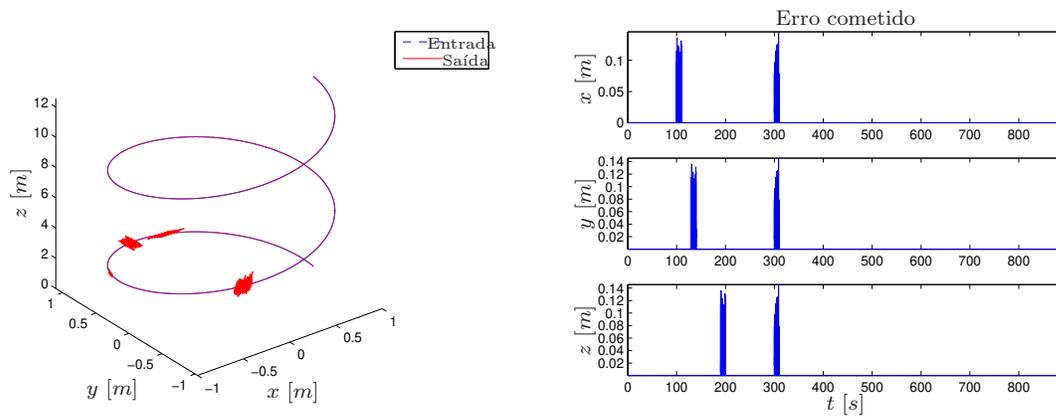


Figura 6.16: Influência de perturbações sobre o sinal de controle utilizando controlador preditivo.

Verifica-se que uma perturbação no sinal de realimentação é mais prejudicial para o correto funcionamento do sistema do que quando ocorre sobre o sinal de controle. Isto porque uma leitura errada faz o controlador executar uma ação de controle inadequada conduzindo o sistema a divergir além de acentuar o erro cometido. Logo, é preciso minimizar a influência de perturbações sobre o sinal de realimentação evitando que haja disparidades drásticas entre o sinal realmente lido e aquele na entrada de realimentação do controlador, conduzindo o sistema a responder da forma como foi inicialmente especificada.

## 6.4 Conclusões

Neste capítulo abordou-se a construção de um ambiente virtual com a finalidade de testar e analisar máquinas-ferramenta CNC, o que possibilitou avaliar o desempenho de um dispositivo cartesiano de três graus de liberdade. Do modo como se concebeu o ambiente foi possível substituir e/ou acrescentar módulos desde que os sinais de interação sejam preservados, possibilitando a sua utilização em uma gama de diferentes máquinas-ferramenta.

No que diz respeito ao dispositivo cartesiano priorizou-se o estudo de dois controladores, um por alocação de pólos e outro preditivo, a fim de avaliar a resposta do sistema na presença ou ausência de perturbações externas. Pelo que se constatou ambos os controladores responderam de modo eficiente aos testes propostos observando-se que o erro cometido ao se comparar o resultado obtido à referência manteve-se abaixo do erro inerente ao dispositivo, o qual é da ordem de décimos de milímetros.

Quanto à existência de perturbações na malha de controle, verificou-se que o pior cenário sucedeu quando as perturbações se apresentavam no sinal de realimentação do sistema conduzindo este a divergir da referência desejada. Do que foi exposto, pode-se considerar que utilizar ferramentas que minimizem a influência de tais perturbações e o uso de equipamentos de medição de maior precisão são duas condições a serem consideradas no projeto de dispositivos de alta precisão.

## 7 Implementação em FPGA

No capítulo anterior discutiu-se a respeito da validação das estratégias propostas com o intuito de avaliar a resposta do sistema com relação às trajetórias de referência e as influências exercidas pelas perturbações externas sobre a malha de controle. Neste capítulo será abordada a implementação em dispositivo físico fazendo uso do kit de desenvolvimento DE2-115 da Altera<sup>®</sup>. Inicialmente, o procedimento priorizará as validações utilizando Matlab/Simulink<sup>TM</sup> em conjunto com a biblioteca DSP Builder, tanto utilizando diagrama de blocos quanto utilizando código VHDL. O passo subsequente será a implementação no ambiente Quartus<sup>®</sup> II para efetiva utilização do hardware escolhido.

### 7.1 Implementação em 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 7.1 é representada uma implementação de controlador utilizando os blocos DSP Builder sendo que o número de coeficientes de cada polinômio (R, S e T) varia conforme o controlador desenvolvido. Os blocos de entrada não-sintetizáveis mostrados na figura 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 a contínuo. Estes 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 deste estudo utilizaram-se 32 bits para representação de todos

os valores.

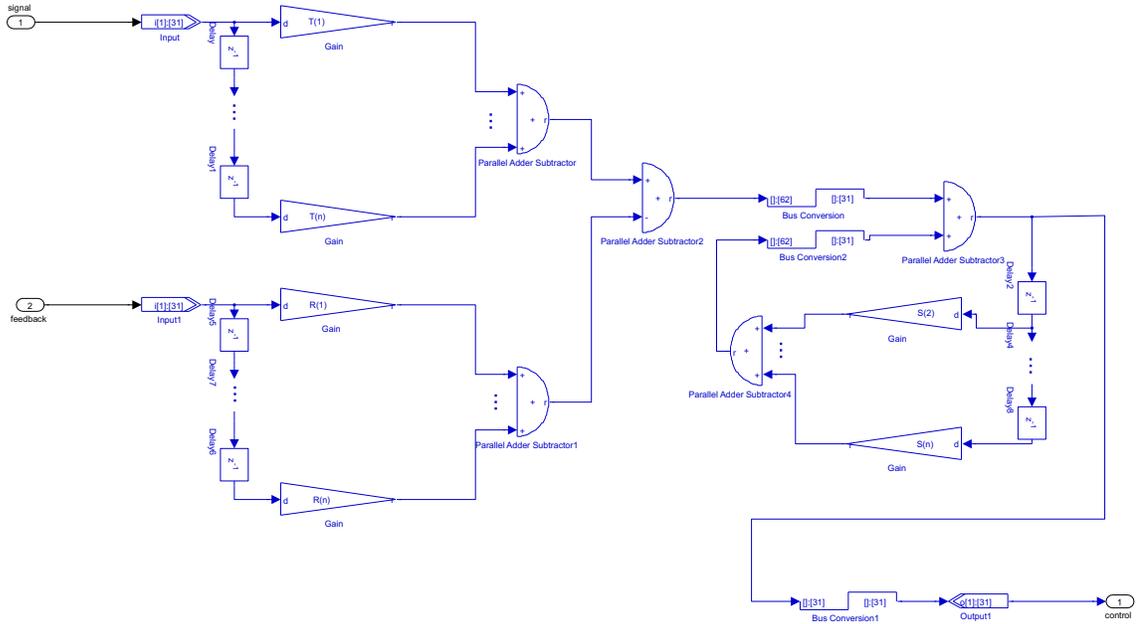


Figura 7.1: Bloco de controle utilizando blocos DSP Builder.

No caso das trajetórias de referência ocorre a mesma restrição dos coeficientes do controlador. Logo, estas também devem ser normalizadas. 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 recalculá-lo. 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 7.2 e 7.3 são apontados os resultados para o controlador por alocação de pólos. Nas Figuras 7.4 e 7.5 são exibidos os resultados para o controlador preditivo. Neste 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 os controladores apresentaram desempenhos semelhantes visto que o erro cometido 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 à obtida via simulação pura.

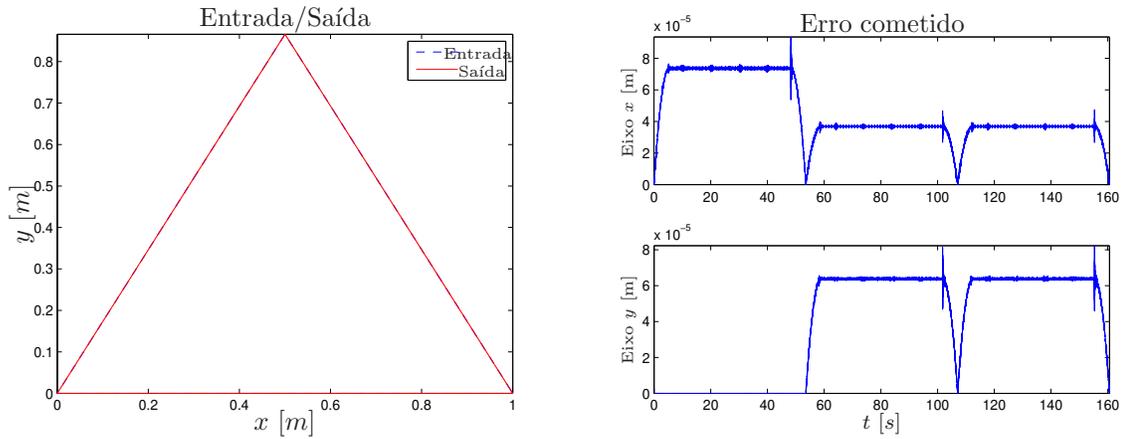


Figura 7.2: Comparação entre entrada e saída do sistema para trajetória triangular utilizando controlador por alocação de pólos implementado em DSP Builder.

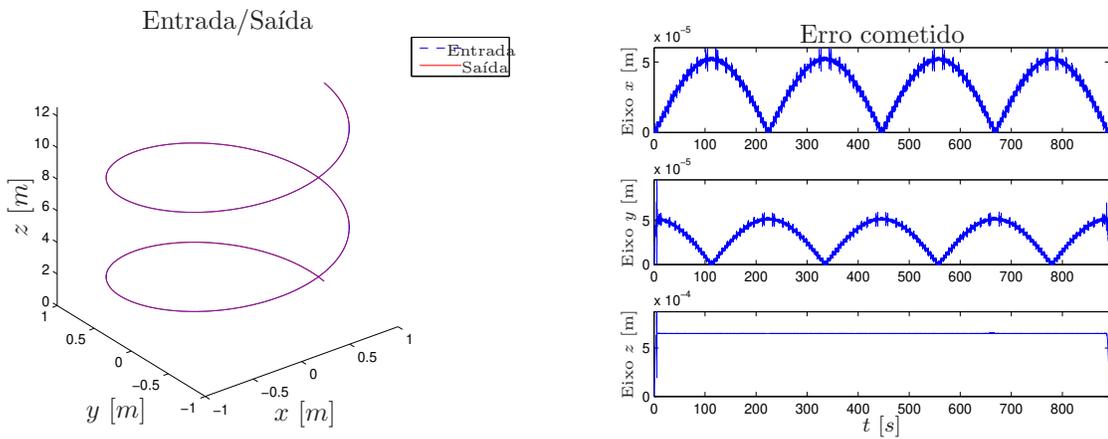


Figura 7.3: Comparação entre entrada e saída do sistema para trajetória helicoidal utilizando controlador por alocação de pólos implementado em DSP Builder.

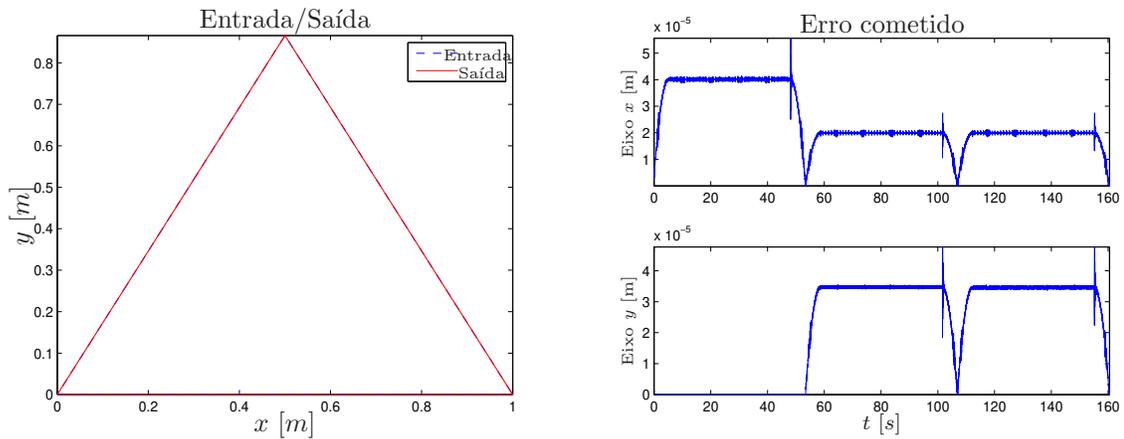


Figura 7.4: Comparação entre entrada e saída do sistema para trajetória triangular utilizando controlador preditivo implementado em DSP Builder.

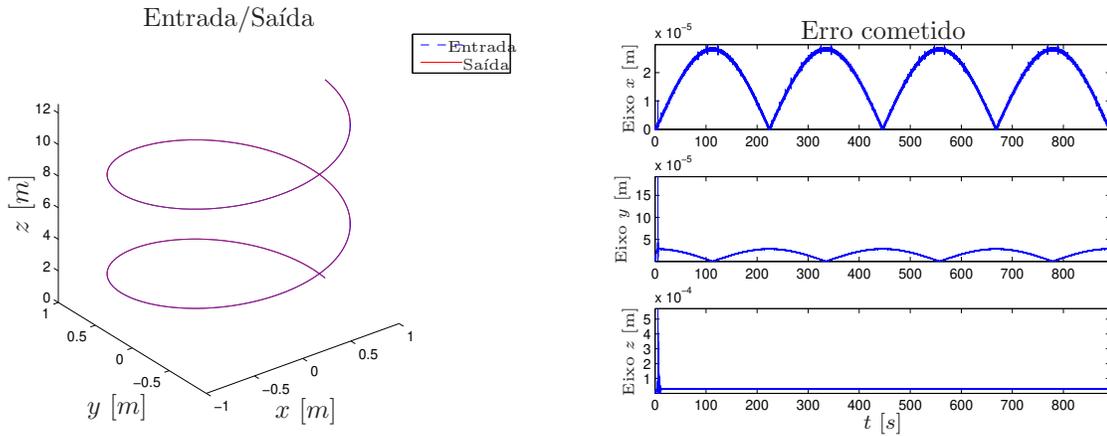


Figura 7.5: Comparação entre entrada e saída do sistema para trajetória helicoidal utilizando controlador preditivo implementado em DSP Builder.

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 esta independe de licença e pode ser aplicada em qualquer placa FPGA disponível.

## 7.2 Implementação com Código VHDL

Para embarcar o código VHDL no ambiente Simulink fez-se uso do bloco de importação de código VHDL disponível na ferramenta DSP Builder, como é mostrado na Figura 7.6.



Figura 7.6: 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, pois números muito próximos podem ser mapeados para o mesmo valor inteiro após o processo.

A partir desta consideração determinaram-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 7.7 e 7.8 são retratados os resultados para ambas as trajetórias utilizando o controlador por alocação de pólos. Nas Figuras 7.9 e 7.10 são mostrados os resultados para o controlador preditivo.

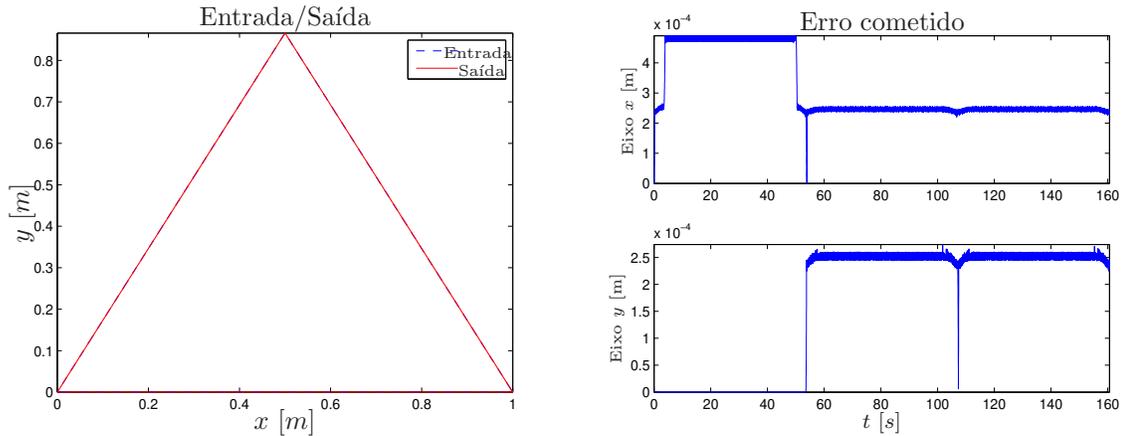


Figura 7.7: Comparação entre entrada e saída do sistema para trajetória triangular utilizando controlador por alocação de pólos implementado em VHDL.

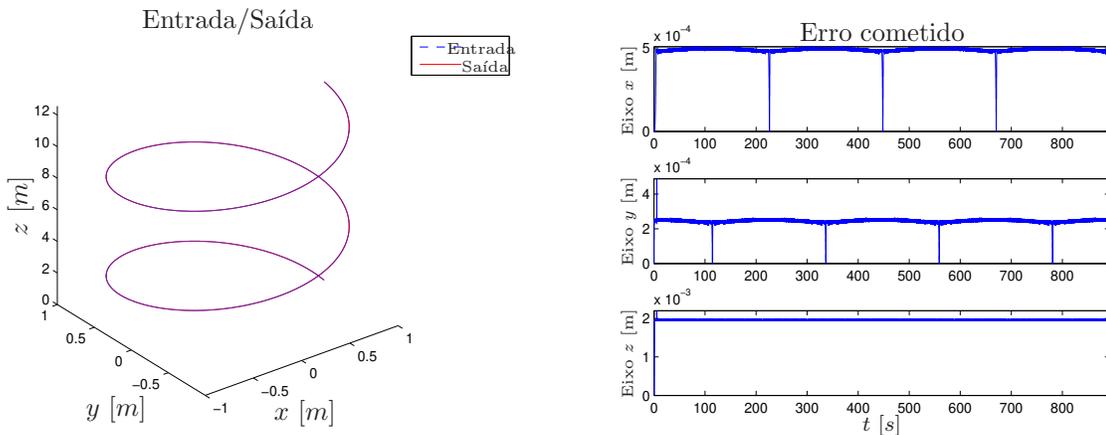


Figura 7.8: Comparação entre entrada e saída do sistema para trajetória helicoidal utilizando controlador por alocação de pólos implementado em VHDL.

Ao analisar os resultados mostrados nas figuras pode-se afirmar que estes 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. Apesar disso, o processo ainda pode ser aplicado para validar a metodologia proposta, pois qualitativamente, os erros cometidos ao utilizar blocos DSP Builder ou código VHDL apresentam

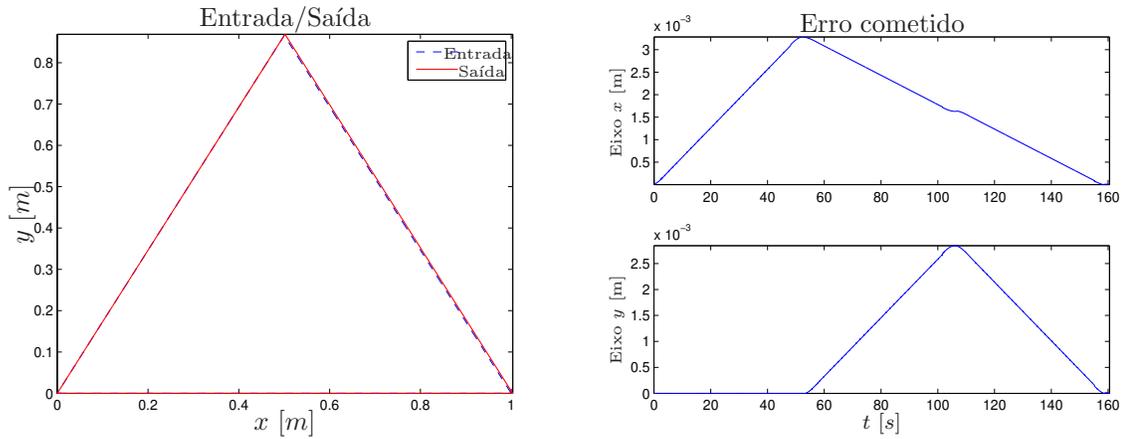


Figura 7.9: Comparação entre entrada e saída do sistema para trajetória triangular utilizando controlador preditivo implementado em VHDL.

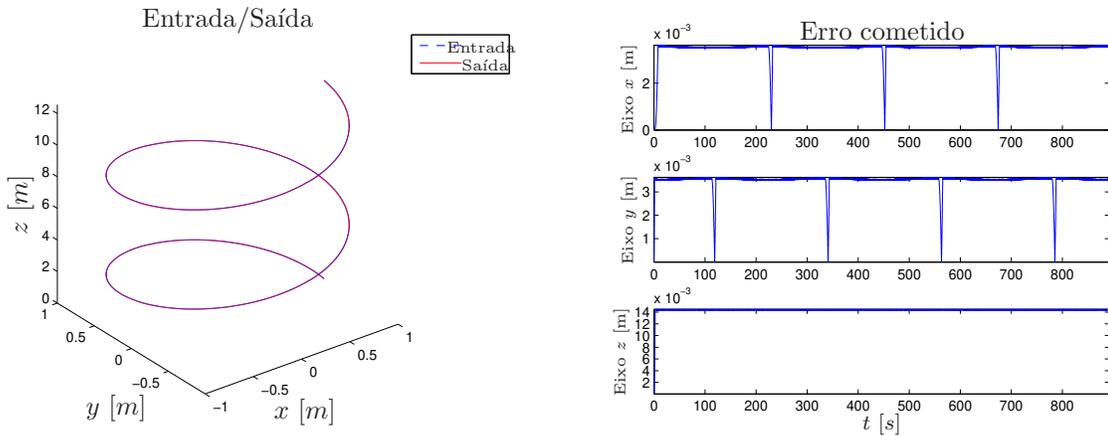


Figura 7.10: Comparação entre entrada e saída do sistema para trajetória helicoidal utilizando controlador preditivo implementado em VHDL.

aproximadamente o mesmo perfil.

### 7.3 Implementação em Ambiente Quartus II

Com a finalidade de testar a implementação do controlador desenvolvido anteriormente na placa FPGA foi construído o ambiente mostrado na Figura 7.11, em que podem ser observados os blocos de controle, de leitura do encoder, de memória, de driver do PWM e de geração de PWM.

No bloco de memória é armazenada a trajetória de referência para o sistema, tornando a



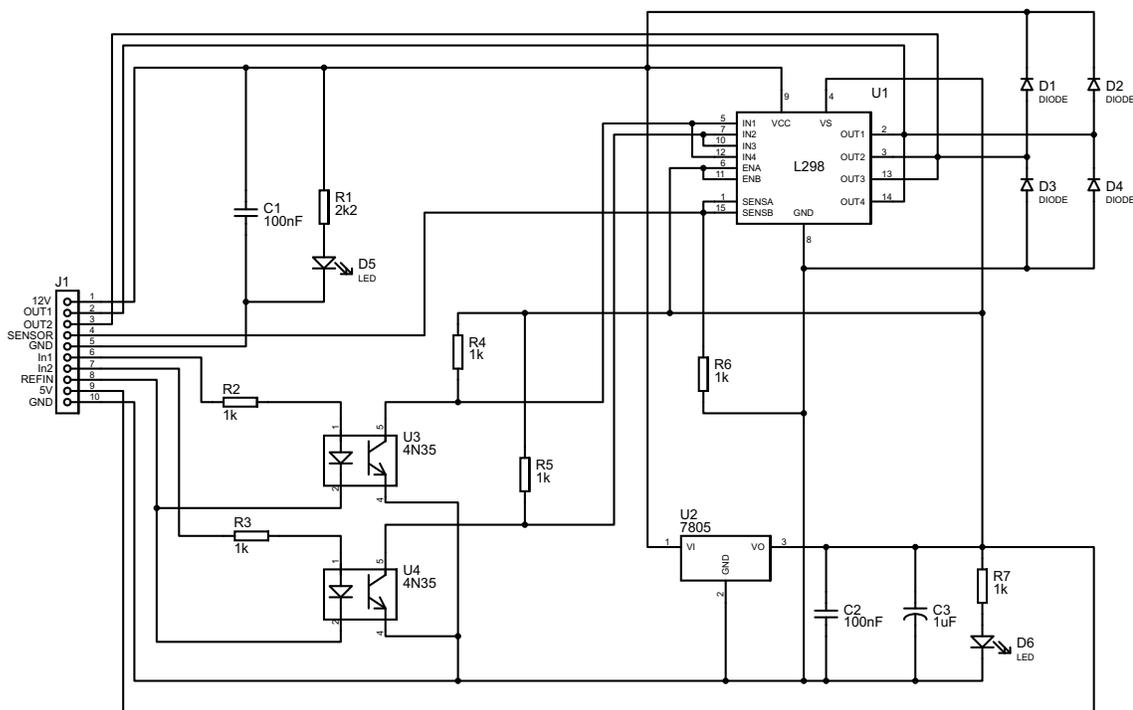


Figura 7.12: Placa de potência utilizada para acionamento do motor CC.

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 CNC. O funcionamento do dispositivo aplicando cada um dos controladores desenvolvidos foi gravado e os links para os vídeos encontram-se no Anexo C.

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 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 este 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.

## 7.4 Conclusões

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

## 8 Considerações Finais e Perspectivas Futuras

Esta pesquisa se propôs a desenvolver um ambiente computacional para planejamento de trajetórias e síntese de controladores, de forma a validar um método alternativo de acionamento de um dispositivo CNC de alta precisão. Para tal, inicialmente procedeu-se ao estudo e a modelagem da cinemática e da dinâmica do dispositivo e de seu sistema de acionamento provendo bases para um melhor entendimento de seu funcionamento e de suas restrições.

Conhecendo as restrições cinemáticas, velocidade e aceleração, optou-se pela escolha de um método de geração de movimentos que considerasse essas restrições durante o planejamento da trajetória garantindo que as restrições nunca fossem violadas. A interpolação por splines foi o método que melhor se ajustou às necessidades, pois além de considerar restrições cinemáticas durante o processo de interpolação, garante a continuidade dos perfis temporais gerados ao longo de toda a trajetória. Outrossim, o método implementado possui a capacidade de segmentar uma curva qualquer em três regiões: aceleração, velocidade constante e desaceleração, garantindo que os estados inicial e final da trajetória sejam sempre de repouso.

O controlador inserido na malha do sistema teve como principal finalidade garantir que a trajetória de referência gerada fosse corretamente executada durante a operação do equipamento. Dentre as técnicas de controle disponíveis na literatura, aquelas fundamentadas no modelo do sistema mostram-se ser as mais adequadas para serem empregadas em dispositivos CNC, uma vez que fornecem soluções mais robustas e de melhor desempenho justamente por utilizarem informações sobre o sistema no processo de síntese do controlador. Assim, optou-se pela escolha dos controladores por alocação de pólos e preditivo, sendo que o primeiro apresenta um comportamento reativo e o segundo antecipativo. Pelas simulações realizadas constatou-se que o controlador preditivo apresentou uma melhor resposta na medida em que o erro cometido ao longo da trajetória foi menor do que aquele cometido pelo controlador por alocação de pólos. Portanto, apesar da resposta do controlador preditivo ser mais lenta, em decorrência de seu aspecto antecipativo, a qualidade da mesma mostrou-se superior à resposta do controlador por alocação de pólos, indicando que a classe de controladores preditivos é adequada para se lidar com dispositivos CNC.

Quanto ao ambiente de simulação proposto, o seu principal mérito reside na sua aplicabilidade a uma gama de diferentes dispositivos, pois é possível trocar os componentes do

ambiente de forma que este se ajuste melhor ao problema que se deseja abordar. No ambiente também é possível inserir perturbações em diferentes pontos do sistema para analisar sua resposta e verificar os locais que, na ocorrência de uma perturbação, acarretarão em maiores prejuízos para a estratégia adotada.

Para efetivar a implementação em dispositivo real, inicialmente foram realizados testes no ambiente Matlab/Simulink com o intuito de averiguar a possibilidade dos controladores responderem como o desejado. No primeiro momento utilizaram-se blocos DSP Builder a fim de implementar os controladores e, em seguida, ocorreu a implementação em código VHDL. A principal vantagem deste método é sua independência do hardware utilizado e dispensar licença para ser compilado e carregado. Por outro lado, como há necessidade de trabalhar-se 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. Durante a execução dos experimentos notou-se que o ruído do sinal do encoder interferia na qualidade do resultado final. Assim, um fotoacoplador foi adotado a fim de minimizar a influência dessa perturbação o que otimizou a precisão de leitura dos pulsos do encoder. Após realizados os experimentos, constatou-se que o dispositivo cartesiano CNC respondia como desejado traçando os perfis da forma que havia sido planejada.

Assim, a análise do procedimento desenvolvido e os resultados obtidos permitiram concluir que foi possível implementar uma forma alternativa de acionamento de um dispositivo cartesiano CNC utilizando motores CC. Outrossim, as metodologias retratadas neste estudo foram implementadas de forma suficientemente genérica a fim de poderem ser utilizadas em outras aplicações. Por fim, espera-se que os métodos aqui desenvolvidos e demonstrados possam servir de base para futuros novos estudos.

## **8.1 Perspectivas Futuras**

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

- modelagem de folgas e atritos existentes em dispositivos CNC possibilitando ao modelo do sistema maior verossimilhança com dispositivo real;
- evolução do algoritmo de planejamento de trajetórias para operar de forma *on-line*;
- implementação de controle em cascata visando garantir não somente o correto rastreamento da posição de entrada, mas também a minimização de oscilações sobre a velocidade com a qual o sistema opera, acarretando numa diminuição da oscilação da corrente no motor e em melhor desempenho global;
- implementação de telas de interface com o usuário no ambiente de simulação de forma que seja possível visualizar o que está ocorrendo em cada etapa do processo e que o usuário possa aplicar modificações conforme necessário;
- utilização do processador Nios<sup>®</sup> possibilitando a otimização de precisão do sistema embarcado, além de prover funcionalidades mais complexas, como transferência de dados via USB.

## 8.2 Trabalhos Produzidos

- Luciano A. Frezzatto Santos, Liz K. R. Ardila, João M. Rosário, CNC Machine Tool - Actuator Selection with Emphasis in Dynamical Modeling and Control, 4th Workshop in Applied Robotics and Automation, Maio 2010, Bauru, SP.
- Luciano Frezzatto, João M. Rosário, Open Architecture For Virtual CNC Machine Tool, V Congreso Internacional de Ingeniería Mecánica y III de Ingeniería Mecatrónica, Agosto 2011, Bogotá, Colômbia.

## 8.3 Trabalhos Relacionados

- Alvaro Uribe, João Rosário, Luciano Frezzatto, Modeling, Control and Analysis of a Serial and Parallelogram Lower Member Mechanism, International Review of Mechanical Engineering (I.R.E.M.E.)

## Referências Bibliográficas

ALTINTAS, Y. *Manufacturing Automation: Metal Cutting Mechanics, Machine Tool Vibrations and CNC Design*. 1st. ed. New York: Cambridge University Press, 2000.

ALTINTAS, Y.; BRECHER, C.; WECK, M.; WITT, S. Virtual machine tool. *CIRP Annals - Manufacturing Technology*, v. 54, n. 2, p. 115 – 138, 2005.

Disponível em: <<http://www.sciencedirect.com/science/article/B8CXH-4SRVXM5-6/2/8cb118cd1705bf15452970fe8f917c08>>.

BOOTHROY, G.; KNIGHT, W. A. *Fundamentals of Machining and Machine Tools*. 3rd. ed. Florida: CRC Press, 2006. ISBN 1-57444-659-2.

BURDEN, R. L.; FAIRES, J. D. *Numerical Analysis*. 7th ed.. ed. [S.l.]: Brooks/Cole Publishing Company, 2001.

CASSEMIRO, E. R. *Proposta de um Ambiente Virtual de Simulação para Projeto de Controlador Preditivo Robusto para Juntas Robóticas*. Tese (Doutorado em Engenharia Mecânica) — Universidade Estadual de Campinas, Campinas, SP, 2006.

CHEN, C.-T. *Linear System Theory and Design*. 3rd. ed. New York: Oxford University Press, 1999.

CHUNG, S. kai; HAHN, J. K. Animation of human walking in virtual environments. In: *CA '99: Proceedings of the Computer Animation*. Washington, DC, USA: IEEE Computer Society, 1999. p. 4. ISBN 0-7695-0167-2.

CLARKE, D.; GAWTHROP, P. Self-tuning controller. *Electrical Engineers, Proceedings of the Institution of*, v. 122, n. 9, p. 929 –934, sep. 1975.

CLARKE, D.; MOHTADI, C.; TUFFS, P. Generalized predictive control—part i. the basic algorithm. *Automatica*, v. 23, n. 2, p. 137 – 148, 1987. Disponível em: <<http://www.sciencedirect.com/science/article/B6V21-47X24MT-DT/2/f3be1c0b788febdce8968c0e9d4d6f2c>>.

CUTLER, C. R.; RAMAKER, B. L. Dynamic matrix control - a computer control algorithm. American Control Conference, San Francisco, 1980.

- DU PLESSIS, L. J.; SNYMAN, J. A. Trajectory-planning through interpolation by overlapping cubic arcs and cubic splines. *International Journal for Numerical Methods in Engineering*, v. 57, n. 11, p. 1615–1641, 2003.
- DYLLONG, E.; VISIOLI, A. Planning and real-time modifications of a trajectory using spline techniques. *Robotica*, Cambridge University Press, New York, NY, USA, v. 21, n. 5, p. 475–482, 2003. ISSN 0263-5747.
- ERICSSON. *Motores de Corrente Contínua*: Guia rápido para uma especificação precisa. 2006. [http://www.siemens.com.br/medias/FILES/2910\\_20060505141908.pdf](http://www.siemens.com.br/medias/FILES/2910_20060505141908.pdf).
- ERKORKMAZ, K.; ALTINTAS, Y. High speed CNC system design. part i: jerk limited trajectory generation and quintic spline interpolation. *International Journal of Machine Tools and Manufacture*, v. 41, n. 9, p. 1323 – 1345, 2001. ISSN 0890-6955.
- FIXTER, M. Fast trajectory generation using bezier curves. In: RUBINSTEIN, B. I. P.; CHAN, N.; KSHETRAPALAPURAM, K. K. (Ed.). *Proceedings of the First Australian Undergraduate Students' Computing Conference*. [S.l.]: Australian Undergraduate Students' Computing Conference, 2003. p. 34–39.
- GEROMEL, J. C.; PALHARES, A. G. B. *Análise Linear de Sistemas Dinâmicos*. [S.l.]: Editora Blucher, 2005.
- HERRERA, I.; SIDOBRE, D. On-line trajectory planning of robot manipulator's end effector in cartesian space using quaternions. In: *15th International Symposium on Measurement and Control in Robotics*. Brussels, Belgium: [s.n.], 2005.
- HERRERA, I.; SIDOBRE, D. Soft motion trajectory planning and control for service manipulator robot. In: *Workshop on Physical Human-Robot Interaction in Anthropic Domains at IROS*. Beijing, China: [s.n.], 2006. p. 13–22.
- KJELLBERG, T.; EULER-CHELPIN, A. von; HEDLIND, M.; LUNDGREN, M.; SIVARD, G.; CHEN, D. The machine tool model—a core part of the digital factory. *CIRP Annals - Manufacturing Technology*, v. 58, n. 1, p. 425 – 428, 2009. Disponível em: <<http://www.sciencedirect.com/science/article/B8CXH-4W1SGP9-G/2/725ddad6ba71adb44a955ff46acd9124>>.
- LAMIRAUX, F.; KAVRAKI, L. E. Path planning for elastic plates under manipulation constraints. In: *In Int. Conf. on Robotics and Automation*. [S.l.: s.n.], 1999. p. 151–156.

- LANDAU, I. D. *System Identification and Control Design*. Englewood Cliffs, New Jersey: Prentice Hall, 1990. (Prentice Hall information and system sciences series).
- LANDAU, I. D. The R-S-T digital controller design and applications. *Control Engineering Practice*, v. 6, n. 2, p. 155 – 165, 1998. ISSN 0967-0661.
- MACFARLANE, S.; CROFT, E. A. Jerk-bounded manipulator trajectory planning: Design for real-time applications. *IEEE Transactions on Robotics and Automation*, v. 19, n. 1, p. 42–52, Feb 2003.
- OGATA, K. *Modern Control Engineering*. 3rd. ed. Nova Jersey: Prentice Hall, 1997.
- OKWUDIRE, C. E.; ALTINTAS, Y. Hybrid modeling of ball screw drives with coupled axial, torsional, and lateral dynamics. *Journal of Mechanical Design*, ASME, v. 131, n. 7, p. 071002, 2009.
- PARK, J.; NAM, S.; YANG, M. Development of a real-time trajectory generator for nurbs interpolation based on a two-stage interpolation method. *The International Journal of Advanced Manufacturing Technology*, v. 26, n. 4, p. 359–365, 2005.
- PASSOS, W. de A. *Utilização de Ferramentas de Prototipagem Rápida direcionada à Concepção de Sistemas Embarcados baseados em Computação Reconfigurável*. 163 p. Dissertação (Mestrado em Mecânica) — Universidade Estadual de Campinas, Campinas, SP, 2008.
- PIEGL, L. On nurbs: a survey. *IEEE Comput. Graph. Appl.*, IEEE Computer Society Press, Los Alamitos, CA, USA, v. 11, n. 1, p. 55–71, 1991. ISSN 0272-1716.
- RICHALET, J.; RAULT, A.; TESTUD, J.; PAPON, J. Model predictive heuristic control: Applications to industrial processes. *Automatica*, v. 14, n. 5, p. 413 – 428, 1978. Disponível em: <<http://www.sciencedirect.com/science/article/B6V21-47WVRPB-42/2/4b67a6cad9f0f6e3997aa298985e6a1b>>.
- ROSÁRIO, J. M. *Princípios de Mecatrônica*. 1st. ed. São Paulo: Prentice Hall, 2004.
- SMITH, A. R. *Spline Tutorial Notes*. [S.l.], 1983.
- SOUSA, F. M. G. de. *Controle de Fresadora para a Prototipagem de Circuitos Impressos*. 160 p. Dissertação (Mestrado em Engenharia Elétrica) — Universidade Estadual de Campinas, Campinas, SP, 1998.

SOUZA, J. P. de. *Implementação de Algoritmos Preditivos para Controle de Juntas Robóticas*. Tese (Doutorado em Engenharia Mecânica) — Universidade Estadual de Campinas, Campinas, SP, 2001.

SUSANU, M. *Commande Predictive Hierarchisee Pour Le Positionnement des Axes de Machine-Outil à Architecture Ouverte*. Tese (PhD) — Supélec, Paris, France, 2005.

WANG, J.-B.; YAU, H.-T. Real-time nurbs interpolator: application to short linear segments. *The International Journal of Advanced Manufacturing Technology*, v. 41, n. 11-12, p. 1169–1185, 2009.

YANG, Z.; RED, E. On-line cartesian trajectory control of mechanisms along complex curves. *Robotica*, Cambridge University Press, New York, NY, USA, v. 15, n. 3, p. 263–274, 1997. ISSN 0263-5747.

YAU, H.-T.; WANG, J.-B. Fast bezier interpolator with real-time lookahead function for high-accuracy machining. *International Journal of Machine Tools and Manufacture*, v. 47, n. 10, p. 1518 – 1529, 2007. ISSN 0890-6955.

YEUNG, C.-H.; ALTINTAS, Y.; ERKORKMAZ, K. Virtual CNC system. part i. system architecture. *International Journal of Machine Tools and Manufacture*, v. 46, n. 10, p. 1107 – 1123, 2006. ISSN 0890-6955.

# ANEXO A – Algoritmos Implementados

## A.1 Splines Cúbicas com Restrição

---

**Algorithm A.1** Splines Cúbicas com Restrição

---

*cubic\_spline*( $n, [t_0..t_n], [x_0..x_n], FPO = x'_0, FPN = x'_n$ )

```
for  $i \leftarrow 0$  to  $n$  do
   $a_i \leftarrow x_i$ 
end for
for  $i \leftarrow 0$  to  $n - 1$  do
   $h_i \leftarrow t_{i+1} - t_i$ 
end for
 $\alpha_0 \leftarrow \frac{3(a_1 - a_0)}{h_0} - 3FPO$ 
 $\alpha_n \leftarrow 3FPN - \frac{3(a_n - a_{n-1})}{h_{n-1}}$ 
for  $i \leftarrow 1$  to  $n - 1$  do
   $\alpha_i \leftarrow \frac{3}{h_i}(a_{i+1} - a_i) - \frac{3}{h_{i-1}}(a_i - a_{i-1})$ 
end for
 $l_0 \leftarrow 2h_0$ 
 $\mu_0 \leftarrow 0.5$ 
 $z_0 \leftarrow \alpha_0/l_0$ 
for  $i \leftarrow 1$  to  $n - 1$  do
   $l_i \leftarrow 2(t_{i+1} - t_{i-1}) - h_{i-1}\mu_{i-1}$ 
   $\mu_i \leftarrow h_i/l_i$ 
   $z_i \leftarrow (\alpha_i - h_{i-1}z_{i-1})/l_i$ 
end for
 $l_n \leftarrow h_{n-1}(2 - \mu_{n-1})$ 
 $z_n \leftarrow (\alpha_n - h_{n-1}z_{n-1})/l_n$ 
 $c_n \leftarrow z_n$ 
for  $j \leftarrow n - 1$  to  $0$  do
   $c_j \leftarrow z_j - \mu_j c_{j+1}$ 
   $b_j \leftarrow (a_{j+1} - a_j)/h_j - h_j(c_{j+1} + 2c_j)/3$ 
   $d_j \leftarrow (c_{j+1} - c_j)/(3h_j)$ 
end for
OUTPUT( $a_j, b_j, c_j, d_j$  for  $j \leftarrow 0, 1, \dots, n - 1$ )
```

---

## A.2 Busca Binária

---

**Algorithm A.2** Busca Binária

---

*binary\_search*( $[P_0..P_{N_I}], S_0, v^*, \ddot{s}_{allow}$ )

$N_{init} \leftarrow 0$

$N_{end} \leftarrow N_I$

**while**  $N_{init} < N_{end}$  **do**

$N \leftarrow \left\lfloor \frac{N_{init} + N_{end}}{2} \right\rfloor$

$\ddot{s} \leftarrow \text{determine\_acceleration}([P_0..P_N], S_0, v^*)$     % (Equation 4.21)

**if**  $\ddot{s} \leq \ddot{s}_{allow}$  **then**

$N_{end} \leftarrow N$

**else**

$N_{init} \leftarrow N$

**end if**

**end while**

OUTPUT( $N_{end}$ )

---

# ANEXO B – Códigos Implementados

## B.1 Controle por Alocação de Pólos

```
LIBRARY ieee;
USE ieee.std_logic_1164.all;
USE ieee.std_logic_signed.all;
USE ieee.std_logic_arith.all;

ENTITY polePlacementController IS

    PORT(
        clk          : IN STD_LOGIC;
        In_W,In_Y    : IN STD_LOGIC_VECTOR(31 DOWNT0 0);
        Out_U        : BUFFER STD_LOGIC_VECTOR(31 DOWNT0 0)
    );

END polePlacementController;

ARCHITECTURE contr OF polePlacementController IS
    TYPE data IS ARRAY(0 TO 9) OF STD_LOGIC_VECTOR(31 DOWNT0 0);
    TYPE data32 IS ARRAY(0 TO 9) OF STD_LOGIC_VECTOR(31 DOWNT0 0);

    SIGNAL Wt,Rt,St : data; -- (Delay Line z-n)

    SIGNAL Out_T,Out_R,In_S : STD_LOGIC_VECTOR(31 DOWNT0 0);

    -- COEFICIENTES POLINOMIO 'T'
    SIGNAL Coef0_T      : STD_LOGIC_VECTOR(31 DOWNT0 0);
    SIGNAL Coef1_T      : STD_LOGIC_VECTOR(31 DOWNT0 0);
    SIGNAL Coef2_T      : STD_LOGIC_VECTOR(31 DOWNT0 0);

    -- COEFICIENTES POLINOMIO 'S'
```

```

SIGNAL Coef0_S      : STD_LOGIC_VECTOR(31 DOWNT0 0);
SIGNAL Coef1_S      : STD_LOGIC_VECTOR(31 DOWNT0 0);
SIGNAL Coef2_S      : STD_LOGIC_VECTOR(31 DOWNT0 0);

-- COEFICIENTES POLINOMIO 'R'
SIGNAL Coef0_R      : STD_LOGIC_VECTOR(31 DOWNT0 0);
SIGNAL Coef1_R      : STD_LOGIC_VECTOR(31 DOWNT0 0);
SIGNAL Coef2_R      : STD_LOGIC_VECTOR(31 DOWNT0 0);

BEGIN

    Coef0_T <= CONV_STD_LOGIC_VECTOR(1073723893,32);
    Coef1_T <= CONV_STD_LOGIC_VECTOR(489760577,32);
    Coef2_T <= CONV_STD_LOGIC_VECTOR(-583963315,32);

    Coef0_S <= CONV_STD_LOGIC_VECTOR(17931,32);
    Coef1_S <= CONV_STD_LOGIC_VECTOR(15025,32);
    Coef2_S <= CONV_STD_LOGIC_VECTOR(3588,32);

    Coef0_R <= CONV_STD_LOGIC_VECTOR(1073723893,32);
    Coef1_R <= CONV_STD_LOGIC_VECTOR(489760577,32);
    Coef2_R <= CONV_STD_LOGIC_VECTOR(-583963315,32);

    -- Implementacao dos atrasos das amostras Bloco T (z-n)
    process(clk)
    begin
        if clk'event and clk='1' then
            Wt(2) <= Wt(1);
            Wt(1) <= In_W;
        end if;
    end process;

    -- Implementacao dos atrasos das amostras Bloco R (z-n)
    process(clk)
    begin

```

```

        if clk'event and clk='1' then
            Rt(2) <= Rt(1);
            Rt(1) <= In_Y;
        end if;
end process;

-- Implementacao dos atrasos das amostras Bloco S (z-n)
process(clk)
begin
    if clk'event and clk='1' then
        St(2) <= St(1);
        St(1) <= Out_U;
    end if;
end process;

-- equacao saida bloco T
process(In_W,Wt,Coef0_T,Coef1_T,Coef2_T)
    variable aux : STD_LOGIC_VECTOR(63 DOWNT0 0);
    variable saida : STD_LOGIC_VECTOR(63 DOWNT0 0);
begin
    aux := In_W * Coef0_T;
    saida := aux;

    aux := Wt(1) * Coef1_T;
    saida := saida + aux;

    aux := Wt(2) * Coef2_T;
    saida := saida + aux;

    Out_T <= saida(62 DOWNT0 31) + saida(63);
end process;

-- equacao saida bloco R
process(In_Y,Rt,Coef0_R,Coef1_R,Coef2_R)
    variable aux : STD_LOGIC_VECTOR(63 DOWNT0 0);
    variable saida : STD_LOGIC_VECTOR(63 DOWNT0 0);

```

```

begin
    aux := In_Y * Coef0_R;
    saida := aux;

    aux := Rt(1) * Coef1_R;
    saida := saida + aux;

    aux := Rt(2) * Coef2_R;
    saida := saida + aux;

    Out_R <= saida(62 DOWNTO 31) + saida(63);
end process;

-- funcao Soma
In_S <= Out_T - Out_R;

-- equacao saida bloco U
process(In_S,St,Coef0_S,Coef1_S,Coef2_S)
    variable aux : STD_LOGIC_VECTOR(63 DOWNTO 0);
    variable saida : STD_LOGIC_VECTOR(63 DOWNTO 0);
begin
    aux := In_S * Coef0_S;
    saida := aux;

    aux := St(1) * Coef1_S;
    saida := saida + aux;

    aux := St(2) * Coef2_S;
    saida := saida + aux;

    Out_U <= saida(62 DOWNTO 31) + saida(63);
end process;

END contr;

```

## B.2 Controle Preditivo

```
LIBRARY ieee;
USE ieee.std_logic_1164.all;
USE ieee.std_logic_signed.all;
USE ieee.std_logic_arith.all;

ENTITY predictiveController IS

    PORT(
        clk          : IN STD_LOGIC;
        In_W,In_Y    : IN STD_LOGIC_VECTOR(31 DOWNTO 0);
        Out_U        : BUFFER STD_LOGIC_VECTOR(31 DOWNTO 0)
    );

END predictiveController;

ARCHITECTURE contr OF predictiveController IS
    TYPE data IS ARRAY(0 TO 9) OF STD_LOGIC_VECTOR(31 DOWNTO 0);
    TYPE data32 IS ARRAY(0 TO 9) OF STD_LOGIC_VECTOR(31 DOWNTO 0);

    SIGNAL Wt,Rt,St : data; -- (Delay Line z-n)

    SIGNAL Out_T,Out_R,In_S : STD_LOGIC_VECTOR(31 DOWNTO 0);

    -- COEFICIENTES POLINOMIO 'T'
    SIGNAL Coef0_T      : STD_LOGIC_VECTOR(31 DOWNTO 0);
    SIGNAL Coef1_T      : STD_LOGIC_VECTOR(31 DOWNTO 0);
    SIGNAL Coef2_T      : STD_LOGIC_VECTOR(31 DOWNTO 0);
    SIGNAL Coef3_T      : STD_LOGIC_VECTOR(31 DOWNTO 0);
    SIGNAL Coef4_T      : STD_LOGIC_VECTOR(31 DOWNTO 0);
    SIGNAL Coef5_T      : STD_LOGIC_VECTOR(31 DOWNTO 0);
    SIGNAL Coef6_T      : STD_LOGIC_VECTOR(31 DOWNTO 0);
    SIGNAL Coef7_T      : STD_LOGIC_VECTOR(31 DOWNTO 0);
    SIGNAL Coef8_T      : STD_LOGIC_VECTOR(31 DOWNTO 0);
```

```

SIGNAL Coef9_T      : STD_LOGIC_VECTOR(31 DOWNT0 0);

-- COEFICIENTES POLINOMIO 'S'
SIGNAL Coef0_S      : STD_LOGIC_VECTOR(31 DOWNT0 0);
SIGNAL Coef1_S      : STD_LOGIC_VECTOR(31 DOWNT0 0);
SIGNAL Coef2_S      : STD_LOGIC_VECTOR(31 DOWNT0 0);
SIGNAL Coef3_S      : STD_LOGIC_VECTOR(31 DOWNT0 0);

-- COEFICIENTES POLINOMIO 'R'
SIGNAL Coef0_R      : STD_LOGIC_VECTOR(31 DOWNT0 0);
SIGNAL Coef1_R      : STD_LOGIC_VECTOR(31 DOWNT0 0);
SIGNAL Coef2_R      : STD_LOGIC_VECTOR(31 DOWNT0 0);
SIGNAL Coef3_R      : STD_LOGIC_VECTOR(31 DOWNT0 0);

BEGIN

    Coef0_T <= CONV_STD_LOGIC_VECTOR(16427797,32);
    Coef1_T <= CONV_STD_LOGIC_VECTOR(14320451,32);
    Coef2_T <= CONV_STD_LOGIC_VECTOR(12231481,32);
    Coef3_T <= CONV_STD_LOGIC_VECTOR(10171230,32);
    Coef4_T <= CONV_STD_LOGIC_VECTOR(8155864,32);
    Coef5_T <= CONV_STD_LOGIC_VECTOR(6210646,32);
    Coef6_T <= CONV_STD_LOGIC_VECTOR(4375058,32);
    Coef7_T <= CONV_STD_LOGIC_VECTOR(2710806,32);
    Coef8_T <= CONV_STD_LOGIC_VECTOR(1314327,32);
    Coef9_T <= CONV_STD_LOGIC_VECTOR(336334,32);

    Coef0_S <= CONV_STD_LOGIC_VECTOR(31081,32);
    Coef1_S <= CONV_STD_LOGIC_VECTOR(-21459,32);
    Coef2_S <= CONV_STD_LOGIC_VECTOR(-9448,32);
    Coef3_S <= CONV_STD_LOGIC_VECTOR(-175,32);

    Coef0_R <= CONV_STD_LOGIC_VECTOR(1435331493,32);
    Coef1_R <= CONV_STD_LOGIC_VECTOR(-2147452567,32);
    Coef2_R <= CONV_STD_LOGIC_VECTOR(788471183,32);

```

```

Coef3_R <= CONV_STD_LOGIC_VECTOR(-96115,32);

-- Implementacao dos atrasos das amostras Bloco T (z-n)
process(clk)
begin
    if clk'event and clk='1' then
        Wt(9) <= Wt(8);
        Wt(8) <= Wt(7);
        Wt(7) <= Wt(6);
        Wt(6) <= Wt(5);
        Wt(5) <= Wt(4);
        Wt(4) <= Wt(3);
        Wt(3) <= Wt(2);
        Wt(2) <= Wt(1);
        Wt(1) <= In_W;
    end if;
end process;

-- Implementacao dos atrasos das amostras Bloco R (z-n)
process(clk)
begin
    if clk'event and clk='1' then
        Rt(3) <= Rt(2);
        Rt(2) <= Rt(1);
        Rt(1) <= In_Y;
    end if;
end process;

-- Implementacao dos atrasos das amostras Bloco S (z-n)
process(clk)
begin
    if clk'event and clk='1' then
        St(3) <= St(2);
        St(2) <= St(1);
        St(1) <= Out_U;    -- z-1
    end if;
end process;

```

```

end process;

-- equacao saida bloco T
process(In_W,Wt,Coef0_T,Coef1_T,Coef2_T,Coef3_T,Coef4_T,
        Coef5_T,Coef6_T,Coef7_T,Coef8_T,Coef9_T)
    variable aux : STD_LOGIC_VECTOR(63 DOWNT0 0);
    variable saida : STD_LOGIC_VECTOR(63 DOWNT0 0);
begin
    aux := In_W * Coef0_T;
    saida := aux;

    aux := Wt(1) * Coef1_T;
    saida := saida + aux;

    aux := Wt(2) * Coef2_T;
    saida := saida + aux;

    aux := Wt(3) * Coef3_T;
    saida := saida + aux;

    aux := Wt(4) * Coef4_T;
    saida := saida + aux;

    aux := Wt(5) * Coef5_T;
    saida := saida + aux;

    aux := Wt(6) * Coef6_T;
    saida := saida + aux;

    aux := Wt(7) * Coef7_T;
    saida := saida + aux;

    aux := Wt(8) * Coef8_T;
    saida := saida + aux;

    aux := Wt(9) * Coef9_T;

```

```

    saida := saida + aux;

    Out_T <= saida(62 DOWNTO 31) + saida(63);
end process;

-- equacao saida bloco R
process(In_Y,Rt,Coef0_R,Coef1_R,Coef2_R,Coef3_R)
    variable aux : STD_LOGIC_VECTOR(63 DOWNTO 0);
    variable saida : STD_LOGIC_VECTOR(63 DOWNTO 0);
begin
    aux := In_Y * Coef0_R;
    saida := aux;

    aux := Rt(1) * Coef1_R;
    saida := saida + aux;

    aux := Rt(2) * Coef2_R;
    saida := saida + aux;

    aux := Rt(3) * Coef3_R;
    saida := saida + aux;

    Out_R <= saida(62 DOWNTO 31) + saida(63);
end process;

-- funcao Soma
In_S <= Out_T - Out_R;

-- equacao saida bloco U
process(In_S,St,Coef0_S,Coef1_S,Coef2_S,Coef3_S)
    variable aux : STD_LOGIC_VECTOR(63 DOWNTO 0);
    variable saida : STD_LOGIC_VECTOR(63 DOWNTO 0);
begin
    aux := In_S * Coef0_S;
    saida := aux;

```

```

        aux := St(1) * Coef1_S;
        saida := saida + aux;

        aux := St(2) * Coef2_S;
        saida := saida + aux;

        aux := St(3) * Coef3_S;
        saida := saida + aux;

        Out_U <= saida(62 DOWNTO 31) + saida(63);
    end process;

END contr;

```

### B.3 Leitura do Encoder

```

library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_signed.all;

entity encoder is
port (
    clk,enc_in_a,enc_in_b : in std_logic;
    reg_vel : out std_logic_vector(31 downto 0)
);
end encoder;

architecture behave of encoder is
signal enccounter : std_logic_vector (31 downto 0) := x"40000000";
constant MAX_VALUE : std_logic_vector (31 downto 0) := x"40000432";
constant MIN_VALUE : std_logic_vector (31 downto 0) := x"C0000432";

begin

```

```

count: process (enc_in_a)
begin
    if enc_in_a'event and enc_in_a = '1' then
        if enc_in_b = '1' then
            enccounter <= enccounter - x"0947";

            if enccounter < MIN_VALUE then
                enccounter <= MIN_VALUE;
            end if;
        else
            enccounter <= enccounter + x"0947";

            if enccounter > MAX_VALUE then
                enccounter <= MAX_VALUE;
            end if;
        end if;
    end if;
end process;

```

```

output: process (clk)
begin
    if clk'event and clk = '1' then
        reg_vel<= enccounter;
    end if;
end process;

```

```

end behave;

```

## B.4 Driver de PWM

```

LIBRARY ieee;
USE ieee.std_logic_1164.all;
USE ieee.std_logic_signed.all;

```

```

USE ieee.std_logic_arith.all;

ENTITY pwmDriver IS

    PORT(
        clk          : IN STD_LOGIC;
        In_S         : IN STD_LOGIC_VECTOR(31 DOWNTO 0);
        pwm          : OUT STD_LOGIC_VECTOR(31 DOWNTO 0);
        dir0         : OUT STD_LOGIC;
        dir1         : OUT STD_LOGIC
    );

END pwmDriver;

ARCHITECTURE conv OF pwmDriver IS

BEGIN
    process(clk, In_S)
    begin
        if In_S(31) = '1' then
            pwm <= not(In_S) + '1';
            dir0 <= '1';
            dir1 <= '0';
        else
            pwm <= In_S;
            dir0 <= '0';
            dir1 <= '1';
        end if;
    end process;
END conv;

```

## B.5 Geração do PWM

```

library ieee;

```

```

use ieee.std_logic_1164.all;
use ieee.std_logic_signed.all;

entity pwm is
port (
    clk,clr_n : in std_logic;
    duty : in std_logic_vector (31 downto 0);
    pwm_out : out std_logic);
end pwm;

architecture behave of pwm is

signal counter : std_logic_vector (15 downto 0);
signal pwm_on : std_logic;

begin
divider: process (clk, clr_n)
begin
    if clr_n = '0' then
        counter <= (others => '0');
    elsif clk'event and clk = '1' then
        counter <= counter + '1';
        if counter >= x"0160" then
            counter <= (others => '0');
        end if;
    end if;
end process;

duty_cyle: process (clk, clr_n)
begin
    if clr_n = '0' then
        pwm_on <= '1';
    elsif clk'event and clk = '1' then
        if counter >= duty(15 downto 0) then
            pwm_on <= '0';
        elsif counter = x"0000" then

```

```
        pwm_on <= '1';
    else
        pwm_on <= pwm_on;
    end if;
end if;
pwm_out <= pwm_on;
end process;

end behave;
```

## ANEXO C – Vídeos

- Funcionamento do dispositivo cartesiano CNC utilizando controlador por alocação de pólos

[http://www.youtube.com/watch?v=dxRZG\\_1Ud1o](http://www.youtube.com/watch?v=dxRZG_1Ud1o)

- Funcionamento do dispositivo cartesiano CNC utilizando controlador preditivo

<http://www.youtube.com/watch?v=19XQNe4v0DY>