

**UNIVERSIDADE FEDERAL DE SANTA MARIA  
CENTRO DE TECNOLOGIA  
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE PRODUÇÃO**

**METODOLOGIA DE CONTROLE PARA EQUIPAMENTOS  
AUTOMÁTICOS PROGRAMÁVEIS ACIONADOS POR  
MOTORES DE PASSO**

**Dissertação de Mestrado**

**Alexandre dos Santos Roque**

**Santa Maria, RS, Brasil  
2010**

**METODOLOGIA DE CONTROLE PARA EQUIPAMENTOS  
AUTOMÁTICOS PROGRAMÁVEIS ACIONADOS POR  
MOTORES DE PASSO**

**por**

**Alexandre dos Santos Roque**

Dissertação apresentada ao Curso de Mestrado do Programa de Pós-Graduação em Engenharia de Produção, Área de Concentração em Gerência da Produção com Ênfase em Automação e Informática Industrial, da Universidade Federal de Santa Maria (UFSM, RS), como requisito parcial para obtenção do grau de **Mestre em Engenharia de Produção**

**Orientador Prof. Dr. Alexandre Dias da Silva**

**Santa Maria, RS, Brasil  
2010**

**Universidade Federal de Santa Maria  
Centro de Tecnologia  
Programa de Pós-Graduação**

A Comissão Examinadora, Abaixo assinada,  
aprova a Dissertação de Mestrado

**METODOLOGIA DE CONTROLE PARA EQUIPAMENTOS AUTOMÁTICOS  
PROGRAMÁVEIS ACIONADOS POR MOTORES DE PASSO**

Elaborada por  
**Alexandre dos Santos Roque**

Como requisito parcial para obtenção do grau de  
**Mestre em Engenharia de Produção**

**COMISSÃO EXAMINADORA:**

---

**Alexandre Dias da Silva, Dr. Eng. (UFSM)**  
Presidente

---

**Raul Ceretta Nunes, Dr. (UFSM)**

---

**Luiz Airton Consalter, Dr. (UPF)**

Santa Maria, 16 de Julho de 2010.

## **AGRADECIMENTOS**

A elaboração de uma dissertação é uma tarefa que representa o término de uma etapa de pesquisa muito importante, realizada em dois anos com dedicação e apoio de várias pessoas. Considero uma etapa porque a pesquisa não termina, e sim continua a partir das contribuições apresentadas. Assim, quero dedicar este trabalho a todos aqueles que contribuíram de alguma maneira para a sua realização, os quais serão citados a seguir.

Primeiramente agradeço a Deus pela saúde e força necessárias para a conclusão deste trabalho, sendo também o meu guia em todas as viagens realizadas a Santa Maria.

À Universidade Federal de Santa Maria pela oportunidade de realização deste mestrado.

Ao professor Alexandre Dias da Silva pela dedicação e orientação dada durante esses dois anos, os quais foram fundamentais para a realização deste trabalho.

A minha mãe (em memória) por toda a dedicação empenhada durante a minha vida para garantir os meus estudos.

Em especial a minha esposa pelo apoio, dedicação e compreensão, que foram importantes para alcançar este objetivo.

A todos os colegas e professores do PPGEP/NAFA da UFSM, pela amizade e companheirismo.

A todos os amigos da vida pessoal e do trabalho na 10ª CSM, que de alguma maneira contribuíram e me apoiaram nesta conquista.

Enfim, a todos os meus sinceros agradecimentos.

## RESUMO

Dissertação de Mestrado  
Programa de Pós-Graduação em Engenharia de Produção  
Universidade Federal de Santa Maria

### **METODOLOGIA DE CONTROLE PARA EQUIPAMENTOS AUTOMÁTICOS PROGRAMÁVEIS ACIONADOS POR MOTORES DE PASSO**

AUTOR: ALEXANDRE DOS SANTOS ROQUE  
ORIENTADOR: ALEXANDRE DIAS DA SILVA, Dr. Eng.  
Data e Local da Defesa: 16 de Julho de 2010, Sala 305 PPGEF.

O desenvolvimento de equipamentos programáveis requer técnicas de controle flexível. A possibilidade de parametrização nas configurações do equipamento resulta em menores custos e agilidade nas fases de projeto. Nas fases de testes comumente são usados *softwares* comerciais fechados ou é gasto tempo dispendioso no desenvolvimento de *hardware* e *software* de controle específico. Este trabalho propõe uma metodologia de controle flexível ou configurável, que possibilita o controle de equipamentos acionados por motores de passo, os quais são amplamente usados no projeto e desenvolvimento de equipamentos com aplicações dedicadas nas indústrias. A metodologia contempla características consideradas fundamentais para a parametrização do equipamento e do ambiente de atuação. A técnica é apresentada e descrita por meio de fluxogramas e diagramas comportamentais da linguagem UML (Linguagem de Modelagem Unificada) para facilitar aplicações futuras. Para demonstrar a viabilidade da técnica proposta foram desenvolvidos um circuito digital e um *software* de controle, implementado na linguagem de programação C++, seguindo as características apresentadas na metodologia. A comunicação realizada pelo protocolo USB (*Universal Serial Bus*) acompanha a tendência atual de utilização dessa interface, no acionamento de equipamentos através de microcomputadores.

Palavras Chave: Equipamentos Programáveis; Sistemas de Controle; Automação Parametrizável/Flexível.

## **ABSTRACT**

Master Dissertation  
Graduate Program of Production Engineering  
Federal University of Santa Maria

### **METHODOLOGY OF CONTROL FOR PROGRAMMABLE AUTOMATIC EQUIPMENTS POWERED BY STEPPER MOTORS**

AUTHOR: ALEXANDRE DOS SANTOS ROQUE  
ADVISER: ALEXANDRE DIAS DA SILVA, Dr. Eng.  
Date and Local: July, 16<sup>th</sup> of 2010, Santa Maria.

Programmable devices require flexible control techniques. The ability to customize settings of the equipment reduces the cost and time of the design phases. On testing phases are commonly used commercial software or spent a lot of time on developing hardware and software to specific control. This work proposes a methodology for flexible or configurable control, applied in equipments powered by stepper motors, which are widely used in academic projects and equipments with dedicated applications in industries. The methodology comprises characteristics considered fundamental to the parameterization of the equipment and the environment of operation. To facilitate future applications, the technique is presented using flowcharts and some behavioral diagrams of UML (Unified Modeling Language). To validate the proposed technique were developed a digital circuit and control software, implemented in the programming language C++, according to the characteristics presented in the methodology. The communication performed by USB protocol (Universal Serial Bus) follows the current trend of using this interface to control equipments through computers.

Keywords: Programmable Automatic Equipments; Control Systems; Automation flexible/configurable.

## LISTA DE FIGURAS

|  |    |
|--|----|
| Figura 1. Diagrama de blocos de um CLP (SILVEIRA e WINDERSON, 1999).....                                       | 20 |
| Figura 2. Fluxo de informações em uma máquina CNC.....   | 21 |
| Figura 3. Diagrama de sistemas controlados por computador (ASTROM, 1997). ....                                 | 23 |
| Figura 4. Controle em Malha Fechada (RIBEIRO, 2001).....   | 26 |
| Figura 5. Controle em Malha Aberta (Adaptado de RIBEIRO, 2001).....  | 27 |
| Figura 6. Funcionamento Motor Unipolar (JONES, 2004).....  | 31 |
| Figura 7. Funcionamento Motor Bipolar (JONES, 2004). ....  | 32 |
| Figura 8. Pinagem da Porta Paralela, conector DB-25.....   | 34 |
| Figura 9. Pinagem RS-232 padrão EIA/TIA-574, conector DB-9. ....   | 35 |
| Figura 10. Topologia de funcionamento do USB (USB.ORG, 2010).....  | 37 |
| Figura 11. Composição genérica de uma transmissão USB (USB.ORG, 2010).....                                     | 38 |
| Figura 12. Controle de motores de passo usado em <i>drives</i> comerciais. ....                                | 39 |
| Figura 13. Controle de motores de passo usando circuitos personalizados.....                                   | 39 |
| Figura 14: Diagrama representativo da metodologia em forma de modelo dinâmico. ....                            | 58 |
| Figura 15: Malha de controle aplicada no software desenvolvido. ....   | 59 |
| Figura 16: Diagrama de Casos de Uso para a Metodologia de Controle Flexível. ....                              | 61 |
| Figura 17: Diagrama de Atividades da Metodologia de Controle.....  | 62 |
| Figura 18: Estrutura típica de controle de uma máquina automática programável que usa<br>motores de passo..... | 63 |
| Figura 19: Circuito de Controle aplicado (LOGASHANMUGAM e SURESH, 2007).....                                   | 65 |
| Figura 20: Chip FTDI DLP-USB245M-G .....   | 67 |
| Figura 21: Circuito desenvolvido e usado nos testes.....   | 68 |
| Figura 22: Projeto para fabricação da placa com o circuito de controle. ....                                   | 69 |
| Figura 23: Tela inicial do software desenvolvido. ....   | 71 |
| Figura 24: Menu Arquivo – ênfase na abertura do código com a trajetória do equipamento. .                      | 72 |
| Figura 25: Menu Configurações – ênfase na parametrização do equipamento.....                                   | 72 |
| Figura 26: Menu Ajuda – descrição geral das funcionalidades do sistema.....                                    | 72 |
| Figura 27: Tela que faz a simulação de uma trajetória especificada. ....                                       | 73 |
| Figura 28: Tela do sistema que mostra a execução de comandos em três eixos/motores do<br>equipamento. ....     | 74 |

|   |    |
|---|----|
| Figura 29: Curva da relação do tempo entre pulsos e a velocidade do motor.....                                  | 75 |
| Figura 30: Exemplo do cálculo usado para regular a aceleração do motor.....                                     | 76 |
| Figura 31: Tela para configuração e ajustes do equipamento. ....  | 77 |
| Figura 32: Função para listagem de informações do dispositivo. ....   | 78 |
| Figura 33. Usando a função <i>FT_Open</i> . ....  | 79 |
| Figura 34. Dados de controle dos motores (LOGASHANMUGAM e SURESH, 2007).....                                    | 80 |
| Figura 35. Envio de pulsos com a composição do <i>byte</i> enviado para a porta USB. ....                       | 81 |
| Figura 36. Sequência de pulsos exemplificada na técnica de interpolação circular (Adaptado de WECK, 1989). .... | 81 |
| Figura 37. Usando a função <i>FT_Write</i> . ....   | 82 |

## LISTA DE TABELAS

|   |    |
|---|----|
| Tabela 1: Passo completo 1 (Full Step).....                                     | 31 |
| Tabela 2: Passo completo 2 (Full Step).....                                     | 31 |
| Tabela 3: Meio Passo ( <i>Half Step</i> ).....                                  | 32 |
| Tabela 4: Acionamento do motor bipolar.....                                     | 33 |
| Tabela 5: Versões e características do protocolo USB.....                       | 38 |
| Tabela 6: Principais características encontradas dos softwares estudados.....   | 52 |
| Tabela 7: Características para compor uma metodologia de controle flexível..... | 54 |
| Tabela 8: Codificação dos Pulsos para movimentação dos eixos. ....              | 56 |

## LISTA DE SIGLAS

|         |  |
|---------|--|
| AMS     | <i>Advanced Micro Systems Inc.</i>                                   |
| CAD/CAM | <i>Computer Aided Design / Computer Aided Manufacturing.</i>         |
| CI      | Circuito Integrado.  |
| CN      | Comando Numérico.  |
| CNC     | Comando Numérico Computadorizado.                                    |
| ECP     | <i>Enhanced Capabilities Port.</i>                                   |
| EIA     | <i>Electronics Industries Association.</i>                           |
| EPP     | <i>Enhanced Parallel Port.</i>                                       |
| FIFO    | <i>First In First Out.</i>   |
| FTDI    | <i>Future Technology Devices.</i>                                    |
| OMG     | <i>Object Management Group.</i>                                      |
| PCP     | Planejamento e Controle da Produção.                                 |
| PIC     | <i>Integrated Peripheral Controller.</i>                             |
| CLP     | Controlador Lógico Programável.                                      |
| RS      | <i>Recommended Standard.</i>   |
| SPP     | <i>Standard Parallel Port</i> (Porta Paralela Padrão).               |
| TTL     | <i>Transistor-Transistor Logic.</i>                                  |
| USB     | <i>Universal Serial Bus</i> (Barramento Serial Universal).           |
| UML     | <i>Unified Modeling Language</i> (Linguagem de Modelagem Unificada). |

## **LISTA DE ANEXOS**

|                             |    |
|-----------------------------|----|
| ANEXO A – PUBLICAÇÕES ..... | 91 |
|-----------------------------|----|

# SUMÁRIO

|   |           |
|---|-----------|
| <b>INTRODUÇÃO .....</b>   | <b>13</b> |
| 1.1 DEFINIÇÃO DO PROBLEMA .....   | 14        |
| 1.2 OBJETIVOS .....   | 14        |
| 1.3 JUSTIFICATIVA .....   | 15        |
| 1.4 METODOLOGIA DA PESQUISA.....  | 15        |
| 1.5 CONTRIBUIÇÕES DA DISSERTAÇÃO .....  | 17        |
| 1.6 ORGANIZAÇÃO DA DISSERTAÇÃO .....  | 17        |
| <b>2 REFERENCIAL TEÓRICO .....</b>  | <b>18</b> |
| 2.1 AUTOMAÇÃO FLEXÍVEL .....  | 18        |
| 2.2 MICROCOMPUTADOR COMO UNIDADE COMANDO .....  | 23        |
| 2.3. CONTROLE EM MALHA FECHADA E MALHA ABERTA .....   | 25        |
| 2.3.1 <i>Controle em Malha Fechada</i> .....  | 25        |
| 2.3.2 <i>Controle em Malha Aberta</i> .....   | 26        |
| 2.4 MOTORES DE PASSO EM EQUIPAMENTOS PROGRAMÁVEIS .....   | 28        |
| 2.4.1 <i>Tipos de Motores de Passo</i> .....  | 29        |
| 2.5 INTERFACES DE COMUNICAÇÃO.....  | 33        |
| 2.5.1 <i>A comunicação paralela e serial</i> .....  | 34        |
| 2.5.2 <i>Protocolo de Comunicação Universal Serial Bus – USB</i> .....                              | 36        |
| 2.5.3 <i>Comunicação entre computador e motores</i> .....   | 39        |
| 2.6 LINGUAGEM DE MODELAGEM UNIFICADA – UML.....   | 40        |
| 2.6.1 <i>Visão Estrutural/Estática e Visão Comportamental/Dinâmica</i> .....                        | 41        |
| <b>3 DESENVOLVIMENTO.....</b>   | <b>44</b> |
| 3.1 ANÁLISE DE REQUISITOS PARA AUTOMAÇÃO FLEXÍVEL .....   | 45        |
| 3.1.1 <i>O uso de motores de passo</i> .....  | 46        |
| 3.1.2 <i>Softwares de controle para equipamentos programáveis: Características principais</i> ..... | 47        |
| 3.1.3 <i>Análise das informações encontradas</i> .....  | 53        |
| 3.2 TÉCNICA DE CONTROLE E SUA REPRESENTAÇÃO .....   | 55        |
| 3.2.1 <i>Metodologia de controle flexível para equipamentos automáticos programáveis</i> .....      | 55        |
| 3.2.2 <i>Malha de Controle</i> .....  | 59        |
| 3.2.3 <i>Diagramas UML – Representação da Metodologia</i> .....                                     | 60        |
| <b>4 RESULTADOS.....</b>  | <b>63</b> |
| 4.1 HARDWARE DE CONTROLE .....  | 63        |
| 4.2 SOFTWARE DESENVOLVIDO.....  | 69        |
| 4.2.1 <i>Funções FT_ListDevices e FT_Open</i> .....   | 78        |
| 4.2.2 <i>Função FT_Write</i> .....  | 79        |

|  |           |
|--|-----------|
| <i>4.2.3 Funções FT_Close, FT_SetTimeouts e FT_Read.....</i> | <i>82</i> |
| <b>5 CONCLUSÃO .....</b>                                     | <b>84</b> |
| <b>REFERENCIAS BIBLIOGRÁFICAS.....</b>                       | <b>86</b> |
| <b>ANEXO A – PUBLICAÇÕES.....</b>                            | <b>91</b> |

## INTRODUÇÃO

Os sistemas de controle para equipamentos automáticos ou robóticos programáveis vêm evoluindo para atender as mais variadas necessidades de controle em nível de *hardware* e *software*, pois, estas necessidades demonstram a criatividade dos pesquisadores nos projetos e posteriormente no sucesso dos testes destes equipamentos. Contudo, para corroborar com este sucesso, os sistemas de controle (*hardware e software*) devem ser cada vez mais susceptíveis a adaptações ou configuráveis as necessidades de controle dos projetos. A agilidade e flexibilidade são conceitos reconhecidos como requisitos para os sistemas de controle com finalidade industrial (DE PAULA e SANTOS, 2008).

Os conceitos de agilidade e flexibilidade abrangem também projetos com outras finalidades, sendo fundamentais para muitos sistemas de controle usados em ambientes industriais, os quais possuem diversos objetivos. Os equipamentos programáveis que fazem o uso de motores de passo como meio de acionamento, são os que mais se beneficiam de tais conceitos, pois, permitem um controle simplificado, de fácil compreensão, aliando movimentos precisos com o seu baixo custo. Frequentemente, pesquisadores desenvolvem ou projetam equipamentos de pequeno porte, acionados por motores de passo, com os mais variados objetivos e aplicados em diferentes áreas (OLIVEIRA et al, 2009), (MAGALHÃES, 2009), (KAMOGAWA e TEIXEIRA, 2009).

Os diversos equipamentos programáveis projetados possuem particularidades que muitas vezes não são atendidas em softwares comerciais, pois estes são sistemas fechados que não permitem ao usuário realizar modificações ou adaptações, fato que faz com que os equipamentos e projetos se adaptem aos softwares de controle e não o contrário. Muitos trabalhos necessitam de mais objetividade e até simplicidade na fase de projeto e testes, possibilitando realizar correções e melhorias. Desta forma, um software de controle com opções de configuração ou parametrização se faz necessário.

O controle de equipamentos com objetivos dedicados tomam como referência exemplos e características consolidadas na indústria, como as que compõem as máquinas de comando numérico computadorizado – CNC, que possuem características de flexibilidade, parametrizáveis em seu sistema de controle, para se adaptar a um projeto ou processo de fabricação (RIBEIRO, 2009). Nesse sentido, em muitas aplicações existem características que podem ser comuns e úteis ao controle de vários equipamentos, podendo ser configurada de

acordo com a sua necessidade. Dentre as características ou funcionalidades que um sistema de controle pode possuir, a forma de comunicação entre o equipamento e o computador pessoal é um fator crucial, pois a evolução das portas de comunicação (serial, paralela, USB 1.0, USB 2.0), propiciaram ao longo do tempo maior velocidade e confiabilidade na transmissão de dados com portabilidade aos diferentes equipamentos. Por ser um padrão de comunicação reconhecido e popularizado (encontrado em todos os computadores pessoais), o protocolo USB (*Universal Serial Bus*) constitui um diferencial importante para o desenvolvimento de uma metodologia de controle flexível.

Este trabalho apresenta uma metodologia de controle flexível (adaptável e configurável) para equipamentos automáticos programáveis acionados por motores de passo, possibilitando a comunicação e controle destes equipamentos por meio de um software instalado em um computador pessoal.

### **1.1 Definição do Problema**

O problema de pesquisa deste trabalho é definido pela seguinte questão: *“Como desenvolver um sistema de controle para equipamentos automáticos programáveis, com baixo custo e possibilidade de adaptação a variados equipamentos e projetos?”*.

### **1.2 Objetivos**

Este trabalho tem como objetivo desenvolver uma metodologia para o acionamento de motores de passo por meio da técnica de controle em malha aberta, com interface de comunicação com o computador através do protocolo USB. A metodologia de controle é dedicada ao projeto e desenvolvimento de equipamentos automáticos programáveis.

Como objetivos específicos têm-se os seguintes:

- Efetuar uma análise bibliográfica sobre os sistemas de controle de máquinas automáticas programáveis e seus meios de comunicação com o computador;

- Desenvolver uma metodologia de controle flexível para máquinas automáticas programáveis usando o protocolo de comunicação USB;
- Analisar e validar a metodologia desenvolvida com o desenvolvimento de um *software* para testes.
- Testar o software de controle utilizando um dispositivo (circuito lógico de controle) de baixo custo para controle de motores de passo usando o protocolo USB;

### 1.3 Justificativa

A finalidade de um sistema de controle flexível é proporcionar maior adaptabilidade com parametrizações do equipamento a ser controlado. Diante disso, estes sistemas são aplicados em diferentes áreas, como por exemplo, no ambiente industrial, onde equipamentos são projetados e desenvolvidos objetivando a redução de custos e diminuindo atividades que prejudicam a saúde humana. Outro exemplo são pesquisas que objetivam o desenvolvimento de equipamentos robóticos que auxiliam no ensino, e que também buscam benefícios a atividades humanas ou sociais. Muitos destes projetos ou protótipos de equipamentos usam motores de passo como atuadores beneficiando-se de sua precisão e facilidade de controle. Estes equipamentos necessitam de *hardware* e *software* de controle específico devido as suas particularidades de funcionamento. Muitos projetos podem não sair do papel por falta de testes eficientes, ou podem demorar muito para serem testados, pois é gasto um tempo dispendioso no desenvolvimento de um sistema de controle dedicado.

Neste contexto surge a motivação de contribuir para a melhoria do projeto e desenvolvimento de novos equipamentos programáveis que usam motores de passo como atuadores, propondo uma metodologia de controle flexível e configurável.

### 1.4 Metodologia da Pesquisa

Este trabalho iniciou com a realização de uma pesquisa bibliográfica sobre controle de equipamentos automáticos programáveis que usam motores de passo como atuadores, em

âmbito industrial ou acadêmico, constatando a realidade atual dos mesmos e explorando as suas sistemáticas de controle.

De acordo com os seus objetivos, esta pesquisa é classificada como *bibliográfico-exploratória*, atuando em área onde existem pesquisas, mas ainda com possibilidades de melhorias, devido a sua amplitude, que podem ser exploradas visando à obtenção de novas soluções e expandir os horizontes de pesquisa da área.

Do ponto de vista da natureza, a pesquisa é classificada como *qualitativa*, porque identifica características fundamentais para a melhoria de tais sistemas de controle. Assim, é proposta e apresentada uma metodologia de controle flexível aplicando estas características, facilitando a sua compreensão e aplicabilidade, para agregar melhorias no processo de controle dos mais variados equipamentos programáveis. Com tais características a metodologia objetiva integrar pesquisadores de outras áreas (Engenharia Mecânica, Elétrica e Ciência da Computação) para projetar e testar diversos equipamentos, promovendo assim atividades interdisciplinares.

Para testes da metodologia proposta, foi utilizado e adaptado um circuito elétrico de controle, como interface de comunicação entre o computador e o equipamento programável. Um diferencial deste circuito de controle é o uso do protocolo USB, que aumenta a portabilidade do dispositivo. A sua composição permite o envio de comandos aos atuadores (motores de passo) do equipamento, por meio da porta de comunicação USB.

Para concluir, um *software* foi desenvolvido, seguindo as características da metodologia proposta, simulando graficamente a trajetória do equipamento controlado, e permitindo correções e ajustes, para em seguida enviar os comandos (pulsos elétricos) do computador para os atuadores do equipamento controlado, por meio do circuito de controle utilizado.

Os testes do *software e hardware* de controle foram realizados no Núcleo de Automação e Projetos de Fabricação - NAFA na Universidade Federal de Santa Maria - UFSM. Trabalhos futuros farão uso da metodologia de controle, permitindo a evolução e melhoria da técnica usada, e possibilitando o controle de diversos equipamentos automáticos programáveis.

## **1.5 Contribuições da Dissertação**

Esta pesquisa tem como principal contribuição à elaboração de uma metodologia de controle flexível para auxiliar no projeto e desenvolvimento de equipamentos automáticos programáveis, que utilizam motores de passo como atuadores, tendo como diferencial a comunicação pelo protocolo USB, metodologia esta, que contempla as características da pesquisa.

## **1.6 Organização da Dissertação**

O trabalho está dividido em cinco capítulos. O capítulo 2 apresenta o referencial teórico da pesquisa, destacando as principais áreas estudadas e trabalhadas. O capítulo 3 apresenta a metodologia de controle proposta, destacando o conjunto de características fundamentais para compor uma metodologia de controle flexível, que possibilite o controle de equipamentos automáticos programáveis acionados por motores de passo, descrevendo também a técnica usada para efetivar o controle. O capítulo 4 destaca os resultados do trabalho, apresentando o hardware de controle utilizado e o software desenvolvido com as características da metodologia, detalhando como ocorre a comunicação entre computador e equipamento. Finalmente, o capítulo 5 apresenta as conclusões e perspectivas futuras sobre o trabalho realizado.

## **2 REFERENCIAL TEÓRICO**

Este capítulo objetiva contextualizar o trabalho desenvolvido abordando os temas relacionados ao estudo. Primeiramente são apresentados os aspectos importantes sobre automação flexível (configurável e parametrizável) e equipamentos automáticos programáveis, que são utilizados em muitas pesquisas. Em seguida é apresentado um estudo sobre as técnicas de controle em malha aberta e fechada, o uso de motores de passo como atuadores, e os meios de comunicação entre computador e equipamentos automáticos programáveis. Por fim, é relatado um estudo sobre a representação e documentação de técnicas e metodologias, contribuindo para a aplicação futura das mesmas em sistemas de controle, destacando o uso da linguagem de modelagem unificada – UML (*Unified Modeling Language*). Segundo Kruchten (2000), a UML é uma linguagem de modelagem visual utilizada para especificar, representar visualmente, construir e documentar as partes que compõem uma técnica aplicada no desenvolvimento de um sistema.

### **2.1 Automação Flexível**

Ao longo dos anos, o processo de automação sofre transformações que são influenciadas pelo avanço tecnológico. O crescente aumento na complexidade das atividades humanas impulsionou a necessidade de automatização de diversos processos na indústria e também em outras áreas. Com esta evolução aliada à tecnologia, há um incremento na capacidade e velocidade de processamento das informações, ocorrendo assim o aprimoramento dos equipamentos automáticos programáveis e de seus sistemas de controle. O objetivo destas melhorias é o aumento no desempenho, na velocidade da transmissão, leitura de informações entre computador e equipamento, e principalmente, na flexibilidade de controle dos equipamentos, que proporciona baixo custo e agilidade no desenvolvimento dos projetos.

A exigência de flexibilidade de controle implica na criatividade no projeto e desenvolvimento de diversos equipamentos automáticos, aumentando cada vez mais a possibilidade de adaptação e reconfiguração das características físicas e técnicas destes

equipamentos (tamanho do ambiente de atuação, número de atuadores, número de sensores, etc.). A adaptação (ou reconfiguração) as mudanças é uma característica humana. Desta forma, a automatização das atividades humanas, agrega características programáveis aos equipamentos, que adicionam inteligência na execução de várias tarefas.

O conceito de automação inclui a idéia de usar a potência elétrica ou mecânica para acionar algum tipo de máquina ou equipamento. Deve acrescentar à máquina algum tipo de inteligência, para que ela execute sua tarefa de modo mais eficiente e com vantagens econômicas e de segurança (RIBEIRO, 2001).

A automação industrial trouxe novos paradigmas para o setor, agregando soluções tecnológicas que transformaram os sistemas de produção, com agilidade, qualidade e segurança. Por outro lado, também surgiram novas áreas de atuação com inúmeras responsabilidades, como por exemplo, a administração dos recursos tecnológicos, o gerenciamento de projetos de fabricação e o planejamento e controle da produção - PCP.

Fernandes Junior (2006) corrobora que, o processo de automação não atinge apenas a produção em si, substituindo o trabalho braçal por robôs e máquinas computadorizadas, mas permite enormes ganhos de produtividade ao integrar tarefas distintas com a elaboração de projetos, o gerenciamento administrativo e a produção.

As responsabilidades geradas a partir do avanço da automação industrial compõem o gerenciamento administrativo (sistema de gestão das organizações), supervisionando e direcionando o trabalho de operação de equipamentos robóticos e de seus sistemas de controle, pois, estes afetam diretamente o sucesso do processo produtivo. De acordo com Stefanovitz (2006), estas soluções tecnológicas têm como objetivo principal o aumento da produtividade da planta industrial automatizada e, conseqüentemente, da competitividade da organização no ambiente ao qual ela se insere.

No processo de automação na indústria é importante mencionar uma técnica de controle chamada comando numérico. Este tipo de controle surgiu na década de 50 propiciando o desenvolvimento da indústria manufatureira. O princípio de funcionamento do comando numérico consiste no recebimento de um programa pela unidade de entrada, sua leitura, interpretação, armazenamento e execução. Essa execução corresponde ao movimento coordenado que a ferramenta faz, determinando a trajetória por interpolação via hardware ou software, por meio de rotinas de programação e algoritmos complexos. Simultaneamente com as máquinas de comando numérico, a robótica industrial começou a integrar o ambiente industrial. A robótica pertence ao tipo de automação programável que possui características de adaptabilidade ao produto, tornando o processo capaz de ser programado quando

especificações de fabricação ou de características do equipamento sofrem algum tipo de alteração (SILVEIRA e WINDERSON, 1999).

Uma técnica importante e muito usada em automação são os Controladores Lógicos Programáveis – CLP. O desenvolvimento do CLP foi crucial para a evolução tecnológica dos processos industriais, pois possibilitou a programação (através de linguagens oriundas dos diagramas elétricos de relés) e controle de processos repetitivos. Conceitualmente, Silveira e Winderson (1999) definem um CLP como um dispositivo eletrônico digital armazenando instruções de funções específicas, como de contagem e aritméticas, todas dedicadas ao controle de máquinas e processos. O CLP também é apresentado como um dispositivo com memória programável para o armazenamento de instruções para controle lógico programável, realizando operações lógicas e aritméticas, manipulação de dados e comunicação em rede, muito usado no controle de sistemas automatizados (SOUZA, 2006). Em sua estrutura, o CLP é basicamente composto por dois elementos principais: uma CPU (central de processamento unitário) e interfaces para os sinais de entrada e saída. A CPU é composta pela unidade de processamento, memória para instruções, e o barramento por onde trafegam dados e funções de controle e endereçamento. O seu princípio de funcionamento é constituído por uma memória programável que armazena internamente instruções e implementa várias funções específicas, tais como lógica, seqüenciamento, temporização, contagem e aritmética. Permite assim, o controle por meio de módulos de entrada e saída, de vários tipos de equipamentos ou processos. A figura 1 ilustra o diagrama de blocos de um CLP.

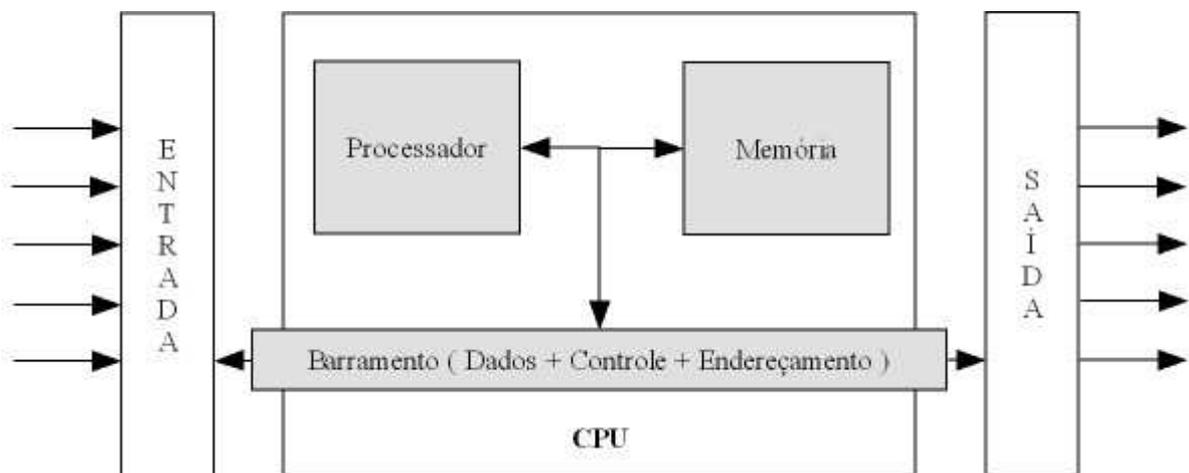


Figura 1. Diagrama de blocos de um CLP (SILVEIRA e WINDERSON, 1999).

O resultado do processo de automação, com características flexíveis, é o uso de equipamentos automáticos programáveis, que trazem benefícios para diversas áreas de pesquisa, sendo fundamentais para a indústria, por meio do processo de automação e controle empregado. Toda máquina ou equipamento que pode ser configurado e adaptado para a realização de uma determinada tarefa, agrega características como flexibilidade, robustez, confiabilidade e segurança, que são pontos chave para os sistemas de controle. Estas características ampliam as possibilidades de controle na realização de várias tarefas. Desta forma, muitos processos repetitivos, que foram por muito tempo, prejudiciais a saúde humana, agora são automatizados.

A automação flexível é composta de diversas técnicas, dentre elas, se destacam o Comando Numérico - CN, os Controladores Lógicos Programáveis – CLP, os softwares CAD/CAM (Projeto auxiliado por computador e Manufatura auxiliada por Computador) e a Robótica Industrial (SILVEIRA e WINDERSON, 1999), (ALMEIDA, 2006). Neste contexto, uma técnica consolidada, que proporciona ampla flexibilidade de controle, são as máquinas de comando numérico computadorizado – CNC. Segundo Groover (2000), a tecnologia CNC pode ser entendida como uma forma de automação flexível, que proporciona o acionamento de um equipamento através do processamento de um conjunto ordenado de símbolos, letras e números. Almeida (2006) confirma que a tecnologia CNC caracteriza um equipamento eletrônico que recebe informações sobre a maneira que a máquina vai realizar uma operação por meio de uma linguagem própria, o programa CNC, processando essas informações, e devolvendo-as ao sistema através de pulsos elétricos. Este conjunto de letras e símbolos compõe a programação em um dispositivo de entrada, sendo uma das partes do fluxo de informações que compõem a máquina CNC. Um dispositivo de entrada processa o código de acionamento da máquina, em seguida a unidade de controle converte estes códigos resultando em movimentos nos atuadores, podendo ainda receber o estado atual do equipamento por meio do uso de sensores. A figura 2 mostra o fluxo geral de informações em máquinas CNC.

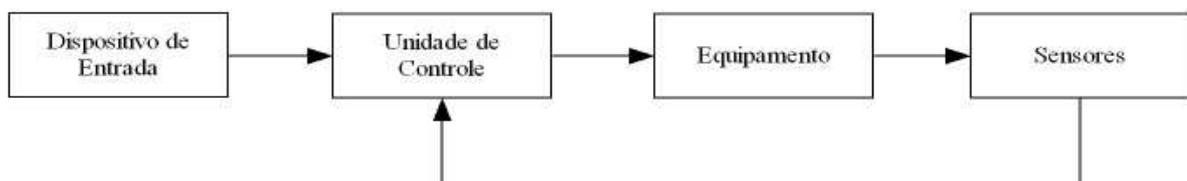


Figura 2. Fluxo de informações em uma máquina CNC.

O conjunto de letras e símbolos, que compõem um programa CNC, pode ser exemplificado pela linguagem padrão denominada *código G*, definida pela ISO 4342 – 1985 ou norma DIN 66025, que utiliza basicamente sentenças de programa composta de palavras reservadas, que informam sobre uma ação a ser realizada. Este código pode ser gerado por um *software* CAD/CAM, que auxilia a integração no projeto e fabricação, transformando a forma geométrica de uma peça na codificação CNC, enviando comandos para fazer a movimentação necessária no equipamento.

No processo de fabricação, relacionando o tempo de produção e a possibilidade de repetição de peças, a produtividade é impulsionada com o advento desta tecnologia. Com máquinas CNC automatizando os processos de fabricação, a indústria obteve um ganho potencial com a produção de peças de alta qualidade. Estas máquinas podem funcionar sem nenhum acompanhamento humano durante um ciclo de usinagem completo, resultando em um ganho de tempo que o operador poderá usar para desempenhar outras tarefas. Com o CNC, o usuário ganha vários benefícios que incluem fadiga de operador reduzida, diminuição dos enganos causados por erro humano (em tempo previsível para cada produto), e principalmente, a versatilidade que a máquina proporciona, ao usinar peças de geometrias complexas (SILVEIRA, 2007).

O amplo uso de programação computadorizada nas áreas de automação industrial e sistemas de controle trouxeram muitas vantagens, onde a versatilidade e flexibilidade despontam como pontos fundamentais, permitindo a realização de projetos cada vez mais complexos. A precisão aliada à segurança que os equipamentos programáveis proporcionam, formaram outro ponto de extrema importância para o ambiente industrial. A programação e configuração das características de controle de um equipamento, por meio de uma interface de comunicação com o computador, permitem a correção de erros com maior rapidez, aumentando a produtividade e dando mais liberdade ao usuário no projeto e desenvolvimento de novos equipamentos.

No contexto de projeto e controle de equipamentos programáveis, a teoria destaca o amplo uso de computadores para desenvolver sistemas controladores, pois proporcionam cálculos sofisticados, facilitando o uso de funções lógicas, usando tabelas para gravar o conhecimento acumulado sobre propriedades do sistema, e também permitindo o uso de interfaces amigáveis e eficientes (ASTRÖM, 1997).

O diagrama da figura 3 caracteriza os sistemas controlados por computador. O sistema contém basicamente três partes: o processo com os conversores A–D (analógico para digital) e D-A (digital para analógico), um algoritmo de controle, e o *clock* (sinal de controle). Os

tempos em que os sinais medidos são convertidos em formato digital são chamados de instantes de tempo, e o tempo entre sucessivos instantes são chamados de períodos de tempo. Estes processos ordenados compõem o hardware e software de controle do equipamento, os quais possuem dois níveis de comunicação, o primeiro entre o equipamento e o hardware de controle (circuito elétrico), e o segundo a comunicação entre operador e computador, por meio de uma interface gráfica.

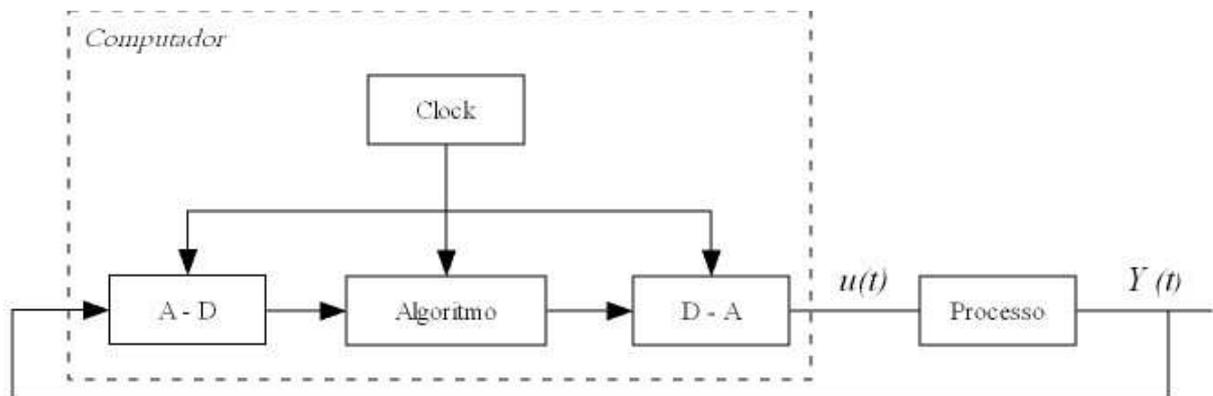


Figura 3. Diagrama de sistemas controlados por computador (ASTROM, 1997).

## 2.2 Microcomputador como unidade comando

Os equipamentos CNC normalmente possuem computadores dedicados para realizar as suas tarefas de controle. Estas tarefas exigem uma capacidade de processamento que, com a evolução tecnológica, é facilmente obtida com computadores pessoais (vendidos comercialmente). Assim, vários projetos de equipamentos programáveis, normalmente fazem o uso de computadores pessoais para efetivar o seu controle.

Muitos trabalhos são realizados focando na possibilidade de adaptação do sistema de controle ao equipamento, acelerando as fases de projeto e prototipação de novos equipamentos. O projetista efetua diversas mudanças para promover correções e melhorias no projeto de equipamentos programáveis, desta forma o sistema de controle deve ser adaptável ou configurável ao equipamento, e não o contrário, como ocorre quando se usam sistemas comerciais. Por serem fechados os sistemas comerciais acabam por não prover uma flexibilidade suficiente, o que acaba por dificultar ou mudar os projetos para que os testes

sejam realizados, possuindo assim muitas limitações e resultados imprecisos ou tendenciosos (segundo as características do sistema comercial).

De acordo com Dutra, Felipe e Carvalho (2009), o desenvolvimento de uma tecnologia própria gera flexibilidade para a pesquisa, pois não se está restrito as limitações impostas por um equipamento ou software comercial, sendo possível realizar alterações de acordo com as necessidades encontradas no decorrer do projeto e do uso em campo.

Os diversos tipos de equipamentos usados para automatizar processos, realizando tarefas repetitivas ou programadas, exigem *softwares* de controle que permitem a parametrização de variáveis que compõem características do equipamento e do ambiente de controle. O uso destes *softwares* de controle por meio de computadores pessoais resulta em economia de recursos, rapidez na execução dos projetos e facilidade de operação e aprendizado por parte dos usuários.

A *flexibilidade e modularidade* para lidar com vários tipos de tarefas, bem como a possibilidade de alteração ou adição de novos módulos de *hardware e software* são características relevantes no desenvolvimento de uma arquitetura de controle (BRITTO, 2008). Em diversas áreas de automação e robótica a flexibilidade desponta como sendo uma característica muito importante para os sistemas de controle, como por exemplo, um protocolo de comunicação para futebol de robôs, onde o protocolo deve permitir ser expansível ou facilmente adaptado para suportar mudanças em *hardware e software* em diferentes quantidades de robôs (MARTINS e TONIDANDEL, 2005). A parametrização do equipamento promove características de expansibilidade às arquiteturas de controle, independente de seu propósito principal.

As necessidades de controle dos equipamentos podem ser variadas e muitas podem ser previstas. Um sistema de controle (*software e hardware*) para equipamentos automáticos programáveis pode ser usado em muitos projetos, que podem possuir características diferentes. Para que este uso seja possível, parâmetros devem ser configuráveis pelo operador (usuário do sistema), como por exemplo, número de motores usados como atuadores no equipamento, as dimensões do ambiente de atuação, codificação da trajetória a ser percorrida (em pulsos/passos, por exemplo), velocidade, aceleração, desaceleração definidas para cada eixo controlável, entre outras informações que podem ser parametrizáveis. O uso de parâmetros e configurações em sistemas que podem ser operados em computadores pessoais comuns, com métodos de controle que aliam eficiência e facilidade de operação, agrega benefícios aos projetistas de diversas áreas, diminuindo custos, e promovendo atividades conjuntas entre pesquisadores de diferentes áreas. Trabalhos recentes focam em tecnologias

baseadas em computação reconfigurável aplicadas no desenvolvimento de sistemas de controle e na automação, pois, mostram-se como alternativa para sua implementação, combinando flexibilidade, confiabilidade e baixo custo (CORREA, 2007).

## 2.3. Controle em Malha Fechada e Malha Aberta

### 2.3.1 Controle em Malha Fechada

Um sistema de controle que estabelece uma relação de comparação entre saída e entrada de referência, utilizando a diferença como meio de controle, é normalmente chamado de **sistema de controle com realimentação**. Um exemplo que pode ser citado é um sistema de controle da temperatura ambiente, medindo a temperatura real com a temperatura de referência ou desejada. Os sistemas de controle com realimentação são mais comumente conhecidos por **sistemas de controle em malha fechada**.

No sistema de controle em malha fechada, o sinal de erro atuante, que é a diferença entre o sinal de entrada e o sinal de realimentação, realimenta o controlador, para minimizar o erro e corrigir a saída do sistema ao valor desejado. Desta forma, o controle em malha fechada objetiva a utilização da realimentação para reduzir o erro do sistema (OGATA, 2003).

A vantagem desta técnica de controle é o fato de que o uso da realimentação faz com que a resposta do sistema seja insensível a variações internas ou externas. É possível também utilizar componentes de baixo custo para obter controle preciso. Uma desvantagem é a estabilidade, pois estes sistemas de controle tendem a correção de erros em excesso, além do necessário, o que causa oscilações de amplitude constante ou variável (OGATA, 2003).

A figura 4 ilustra os elementos que compõem o controle em malha fechada.

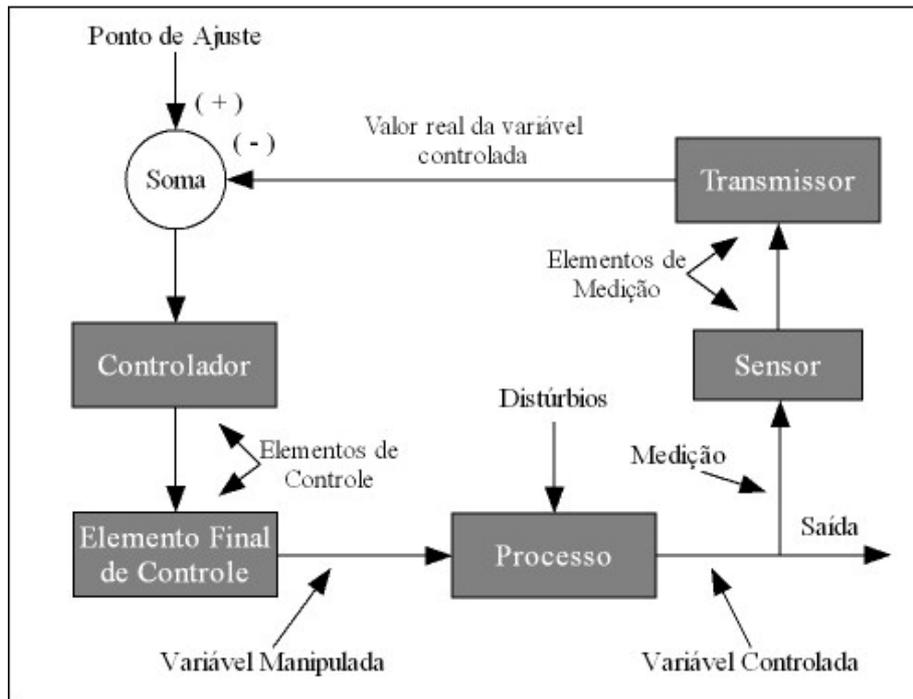


Figura 4. Controle em Malha Fechada (RIBEIRO, 2001).

Nos sistemas que fazem uso desta técnica de controle, os atuadores são acionados de acordo com dados provenientes dos sensores (realimentação), em tempo real. Todas as alterações geradas no sistema são monitoradas pelos sensores, permitindo um controle mais rígido e eficiente, porém tornando o sistema mais caro e complexo (SILVEIRA e WINDERSON, 1999). Existem vantagens e desvantagens no uso do controle em malha fechada, porém a melhor análise do sistema a controlar é que definirá a sua escolha ou não. Este controle é mais indicado quando existem distúrbios ou alterações desconhecidas ao sistema, devendo levar em conta também o custo, pois como enfatizado, os sistemas de controle em malha fechada são geralmente mais caros devido ao número maior de componentes necessários.

### 2.3.2 Controle em Malha Aberta

Os sistemas de controle em que a saída não exerce nenhuma ação sobre a entrada, ou seja, não é usada como referência para uma nova entrada, são denominados **sistemas de controle em malha aberta**. Um exemplo prático é a máquina de lavar roupas, onde as

operações de molho, lavar e enxaguar são executadas em uma seqüência de tempo. Estas operações de saída não são medidas, ou seja, não se verifica se as roupas estão bem lavadas.

De acordo com Ogata (2003), os sistemas de controle em malha aberta são aqueles em que a saída não exerce nenhuma ação de controle no sistema, ou seja, o sinal de saída não é medido nem realimentado para comparação com a entrada. Desta forma, a precisão de um sistema de controle em malha aberta depende de uma calibração, pois, qualquer distúrbio fará com que o sistema não execute a tarefa desejada. Este modo de controle é indicado quando a relação entre a entrada e saída é conhecida, e não há distúrbio interno ou externo.

A técnica de controle em malha aberta consiste no acionamento de atuadores de acordo com tarefas programadas, sendo mais econômico e simples de ser desenvolvido. Este sistema pode se tornar ineficiente em algumas situações, pois o sistema de controle não tem conhecimento de como os atuadores estão recebendo as informações, ou seja, se está efetuando o movimento desejado (SILVEIRA e WINDERSON, 1999).

A figura 5 demonstra como ocorre o controle em malha aberta, destacando os seus elementos principais.

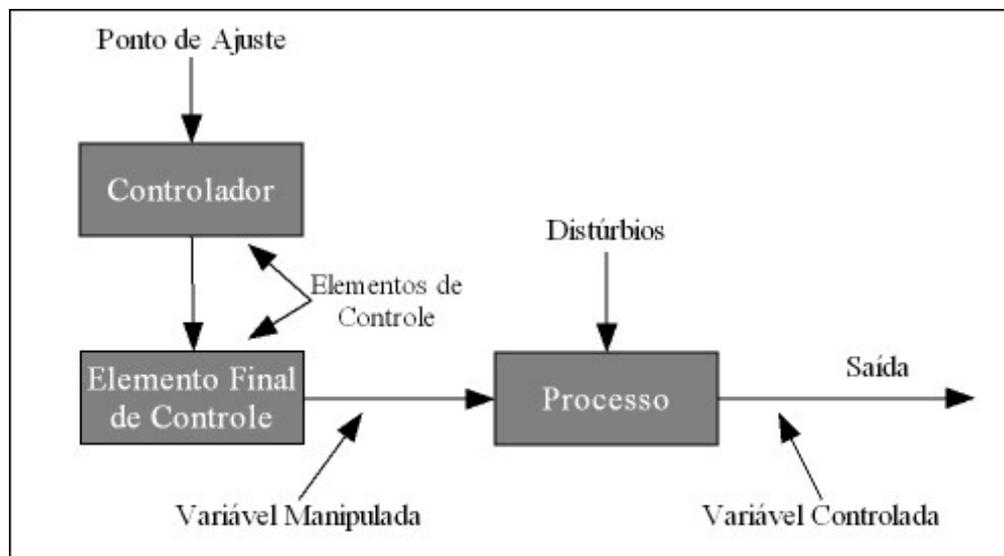


Figura 5. Controle em Malha Aberta (Adaptado de RIBEIRO, 2001).

A seguir são relacionadas algumas vantagens e desvantagens do uso do controle em malha aberta (OGATA, 2003).

Vantagens:

- São simples de serem construídos e têm fácil manutenção;
- São menos dispendiosos do que um correspondente em malha fechada;
- Não apresentam problema de estabilidade;
- São apropriados quando existem dificuldades de avaliação da saída, ou quando sua medição não é economicamente possível.

Desvantagens:

- Distúrbios e mudanças na calibração causam erros e a saída pode apresentar diferenças em relação ao padrão esperado; e
- Para que a saída mantenha a qualidade requerida, é necessária uma regulagem periódica.

## 2.4 Motores de Passo em equipamentos programáveis

Em equipamentos programáveis onde a necessidade de precisão é o fator de maior importância, os motores de passo se destacam. Nestes projetos o controle em malha aberta é amplamente usado, pois a configuração prévia de características dos equipamentos é suficiente para gerar a movimentação desejada, além de possuir baixo custo e facilidade de implementação. Estes motores não são usados em projetos de equipamentos onde a necessidade de alto torque seja o ponto fundamental, pois esta é a sua limitação principal.

Os motores de passo são atuadores eletromagnéticos que trabalham convertendo pulsos elétricos em movimentos mecânicos. A cada pulso ocorre um deslocamento incremental de um pequeno ângulo, permanecendo nesta posição se mais nenhum pulso for aplicado. Este ângulo de movimentação a cada pulso é uma característica muito importante e inerente a todos os motores, pois define quantos pulsos são necessários para o motor dar uma volta completa em seu próprio eixo. Essa característica de movimentação define o que normalmente é denominado **resolução do sistema** em que o motor está atuando, por meio da distância (em milímetros, por exemplo) percorrida a cada pulso aplicado ao motor.

De acordo com a *AMS – Advanced Micro Systems INC*, um motor de passo é um dispositivo eletromagnético que converte sinais digitais em movimentos mecânicos. Como vantagens esses motores possuem baixo custo, alta confiabilidade, alto torque em baixas

velocidades e implementação simples e robusta, trabalhando em praticamente qualquer ambiente. A principal desvantagem da utilização do motor de passo é o *efeito de ressonância* (frequência natural característica, que quando o motor atinge resulta em aumento de ruído e vibração, podendo perder passos ou oscilar), freqüentemente exposto em baixas velocidades e diminuindo o torque com o aumento da velocidade. Este efeito ocorre em virtude da freqüência de trabalho do motor, pois, trabalhando em rotações muito altas, o motor pode perder passos e conseqüentemente sua precisão, assim é importante saber os limites de operação do motor aplicado ao projeto. A movimentação dos motores de passo se dá por meio da energização seqüencial de suas bobinas e também da constituição física do motor, sendo assim, o motor movimentará o seu eixo em determinado ângulo. A seguir serão abordados os tipos de motores de passo com suas características de funcionamento principais.

#### 2.4.1 Tipos de Motores de Passo

As características físicas e elétricas diferenciam os motores de passo, e desta forma elas devem ser levadas em consideração no momento da escolha do tipo de motor usado nos equipamentos. Os motores de passo podem ser classificados basicamente em três tipos, os de ***Relutância Variável, Imã Permanente e Híbridos***. Quanto ao seu modo de operação, eles podem ser ***unipolares e bipolares***, onde o modo de energização das bobinas é diferenciado. A seguir são apresentadas as principais diferenças entre estes tipos.

##### *Relutância Variável*

Os motores de relutância variável consistem de um rotor de ferro com múltiplos dentes e um estator com enrolamentos. Quando os enrolamentos do estator são energizados os pólos são magnetizados, desta forma, a rotação ocorre quando os dentes do rotor são atraídos para os pólos do estator energizado.

##### *Imã Permanente*

Os motores de imã permanente possuem baixa resolução, com passos em torno de 7.5° a 15° (48 – 24 passos/volta). O seu rotor é construído com imãs permanentes e não possui

dentos. Os pólos magnetizados do rotor provêm maior intensidade de fluxo magnético e por isso ele possui uma melhor característica de torque em relação ao motor de relutância variável.

### *Híbridos*

Os motores híbridos possuem melhor desempenho no que diz respeito à resolução do passo, torque e velocidade. Seus ângulos de passo estão entre  $3,6^\circ$  a  $0,9^\circ$  (100 – 400 passos/volta). Este tipo de motor combina as melhores características dos motores de imã permanente e de relutância variável, tendo o rotor multi-dentado e possuindo um imã permanente ao redor de seu eixo.

Quanto ao modo de operação os motores de passo podem ser *Unipolares* e *Bipolares*:

### *Motores Unipolares*

Os motores de passo unipolares possuem quatro bobinas e a energização das mesmas é realizada em um único sentido. Este motor tem como principais vantagens a facilidade de controle e maior precisão, e sua desvantagem está relacionada a aplicações onde o maior torque com precisão é fundamental. O número de fases do motor unipolar é duas vezes o número de bobinas, uma vez que cada bobina se encontra dividida. A derivação central dos enrolamentos é ligada ao terminal positivo da fonte de alimentação e os extremos de cada enrolamento são ligados alternadamente ao terra, para assim inverter a direção do campo gerado por cada um dos enrolamentos (JONES, 2004). A figura 6 ilustra o modo de funcionamento do motor unipolar.

A quantidade de bobinas do motor energizadas em uma seqüência define o tipo de passo aplicado e pode aumentar ou diminuir a precisão do movimento. As tabelas 1, 2 e 3 apresentam as seqüências corretas de energização das bobinas, onde o nível lógico “1” significa energizar, de acordo com o tipo de passo usado.

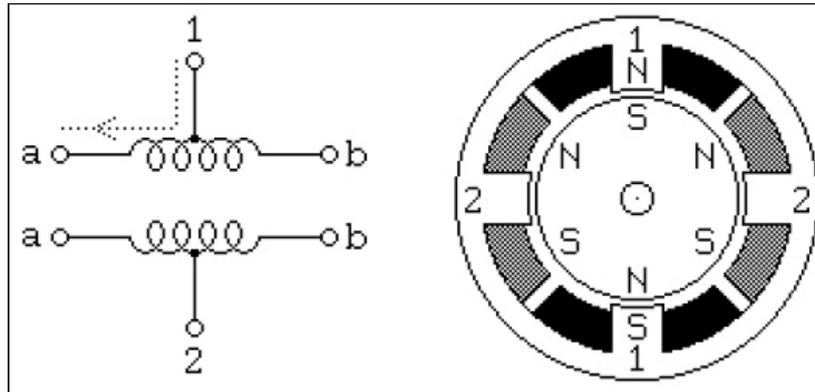


Figura 6. Funcionamento Motor Unipolar (JONES, 2004).

No **passo completo 1** (tabela 1), são energizadas uma bobina de cada vez, e no **passo completo 2** (tabela 2), duas bobinas de cada vez, fazendo com que aumente o torque do motor. O tipo de acionamento *Half Step* ou **meio passo** alterna a seqüência de acionamento, energizando uma bobina e depois duas, dobrando o número de passos e proporcionando maior precisão.

Tabela 1: Passo completo 1 (Full Step).

| Nº Passo | B4 | B3 | B2 | B1 | Decimal |
|----------|----|----|----|----|---------|
| 1        | 1  | 0  | 0  | 0  | 8       |
| 2        | 0  | 1  | 0  | 0  | 4       |
| 3        | 0  | 0  | 1  | 0  | 2       |
| 4        | 0  | 0  | 0  | 1  | 1       |

Tabela 2: Passo completo 2 (Full Step).

| Nº Passo | B4 | B3 | B2 | B1 | Decimal |
|----------|----|----|----|----|---------|
| 1        | 1  | 0  | 1  | 0  | 10      |
| 2        | 1  | 0  | 0  | 1  | 9       |
| 3        | 0  | 1  | 0  | 1  | 5       |
| 4        | 0  | 1  | 1  | 0  | 6       |

Tabela 3: Meio Passo (*Half Step*).

| Nº Passo | B3 | B2 | B1 | B0 | Decimal |
|----------|----|----|----|----|---------|
| 1        | 1  | 0  | 0  | 0  | 8       |
| 2        | 1  | 1  | 0  | 0  | 12      |
| 3        | 0  | 1  | 0  | 0  | 4       |
| 4        | 0  | 1  | 1  | 0  | 6       |
| 5        | 0  | 0  | 1  | 0  | 2       |
| 6        | 0  | 0  | 1  | 1  | 3       |
| 7        | 0  | 0  | 0  | 1  | 1       |
| 8        | 1  | 0  | 0  | 1  | 9       |

### *Motores Bipolares*

Os motores de passo bipolares possuem duas bobinas e a energização das mesmas deve ser realizada em ambas as direções para permitir o movimento de um passo. A corrente precisa ser invertida a fim de inverter o pólo magnético, desta forma o circuito de condução é mais complexo. Os motores bipolares são conhecidos pela excelente relação tamanho/torque, pois, proporcionam um maior torque comparativamente a um motor unipolar do mesmo tamanho (JONES, 2004). A figura 7 ilustra o modo de funcionamento do motor bipolar.

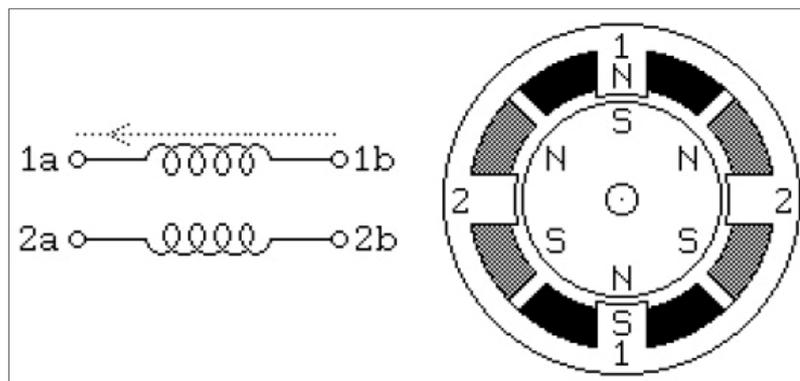


Figura 7. Funcionamento Motor Bipolar (JONES, 2004).

O padrão de acionamento é semelhante ao motor de passo unipolar, mas em vez de 0 e 1 é usado o sinal de polaridade aplicado as bobinas. A tabela 4 exemplifica este tipo de acionamento.

Tabela 4: Acionamento do motor bipolar.

| <b>Nº Passo</b> | <b>1<sup>a</sup></b> | <b>1b</b> | <b>2a</b> | <b>2b</b> |
|-----------------|----------------------|-----------|-----------|-----------|
| 1               | +                    | -         | -         | +         |
| 2               | +                    | -         | +         | -         |
| 3               | -                    | +         | +         | -         |
| 4               | -                    | +         | -         | +         |

## 2.5 Interfaces de Comunicação

A possibilidade de comunicação entre o computador e um dispositivo externo permitiu o desenvolvimento de vários tipos de equipamentos com aplicações dedicadas, como impressoras, scanners, equipamentos industriais, entre tantos outros. Atualmente os tipos de comunicação podem ser físicos (conexão com fios) ou sem fio (diversos tipos de conexão *Wireless*), mas o foco neste trabalho serão os tipos de comunicação física que são comumente usados para o controle dos mais variados equipamentos, como, a porta serial (*RS232*), a porta paralela, e principalmente a comunicação USB (*Universal Serial Bus*).

Devido à alta portabilidade aliada à alta taxa de transmissão de dados, a comunicação USB vem substituindo e extinguindo os outros meios de comunicação física do computador, tendo como prova simples dessa evolução o fato dos computadores pessoais atuais já saírem de fábrica somente com portas de comunicação USB. A seguir é apresentada uma síntese das características principais e do modo de controle destes meios de comunicação comumente usados.

### 2.5.1 A comunicação paralela e serial

#### *Porta Paralela*

Este meio de comunicação possui este nome devido ao fato da transmissão dos dados ser de forma paralela, pois transmite 8 *bits* de dados de uma só vez, formando um *byte* (BRAGA, 1996). A porta paralela de um computador geralmente consiste de 25 pinos em um conector DB-25 (Figura 8). Estes pinos podem interagir com a lógica TTL (*Transistor-Transistor Logic*) de um dispositivo externo, como entradas ou saídas. Alguns pinos podem ser utilizados como entradas apenas, enquanto outros podem ser alterados via software, entre o modo de entrada e saída (ELSON, 2000).

A porta paralela possui três modos de transmissão, um unidirecional e dois bidirecionais. O modo unidirecional SPP (*Standard Parallel Port*) possui taxa de transmissão de até 150 KB/s. No modo bidirecional EPP (*Enhanced Parallel Port*) ou ECP (*Enhanced Capabilities Port*) com acesso direto a memória, a porta pode atingir uma transferência de até 2 MB/s, mas fazendo uso de um cabo especial.

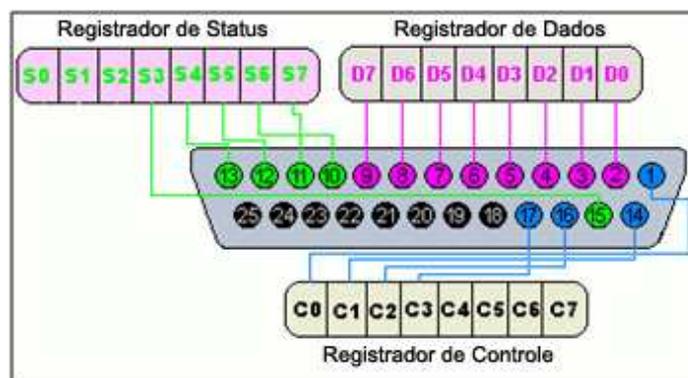


Figura 8. Pinagem da Porta Paralela, conector DB-25.

#### *Porta Serial*

Este padrão de comunicação é também conhecido como *RS232*, tendo sua transmissão de dados em série, ou seja, a porta recebe e envia sinais em seqüência. Os *bits* de dados precisam ser enfileirados e enviados um a um (BRAGA, 1996).

O padrão recomendado ou RS (*Recommended Standard*) define a padronização de uma interface de comunicação de dados entre equipamentos, criado pela EIA/TIA (*Electronics Industries Association / Telecommunications Industry Association*), sendo o RS-232 um padrão para troca serial de dados binários.

Neste padrão de comunicação os caracteres são enviados um a um como um conjunto de *bits*, e seu modo de transmissão pode ser síncrono ou assíncrono. No *modo síncrono*, canais separados são usados para transmissão de dados e informações de controle. Através da recepção periódica de um pulso de *clock*, o receptor lê o canal de dados e armazena o valor do *bit* encontrado. Como o receptor é responsável pelos pulsos de dados e de controle, o receptor irá ler o canal de dados apenas quando comandado pelo transmissor. Já em modo assíncrono, cada transmissão é iniciada com um *bit* de partida (*start*), que é enviado para cada *byte* ou caractere, e um sinal de parada (*stop*) é enviado após cada *byte* (DAVID, 2009).

Para a comunicação entre o computador e um equipamento, o padrão RS-232 pode ter conectores de 3 a 25 pinos, mas um conector muito usado é o DB-9 (Figura 9) que possui 9 pinos, tendo sido seu uso padronizado na norma TIA-574.

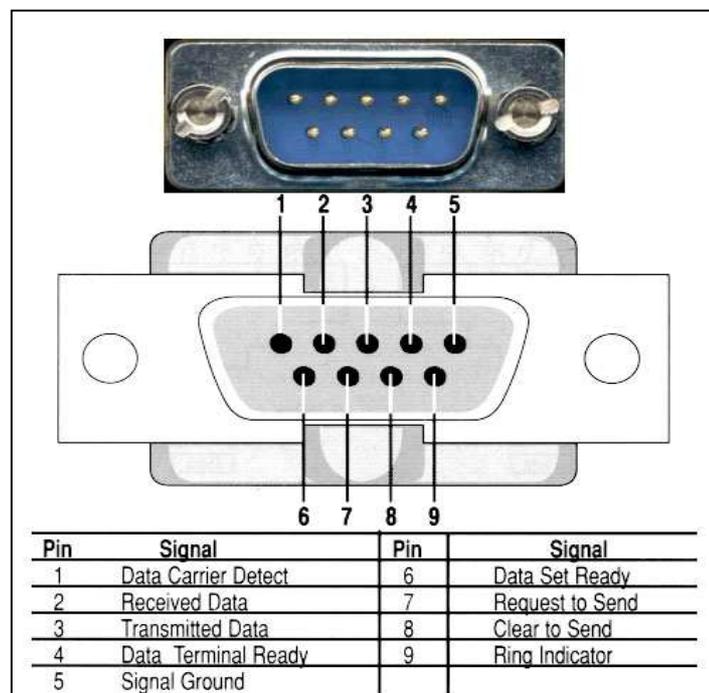


Figura 9. Pinagem RS-232 padrão EIA/TIA-574, conector DB-9.

## 2.5.2 Protocolo de Comunicação Universal Serial Bus – USB

O protocolo de comunicação USB surgiu da necessidade de um método de conexão de periféricos de computador mais rápido, simples e eficiente. Esta necessidade originou uma parceria de grandes empresas (como exemplo, *INTEL*, *HP*, e *Microsoft*) que desenvolveram em 1995 o barramento serial universal – USB (USB.ORG, 2010). Esse meio de comunicação se popularizou tanto que atualmente milhões de dispositivos ou equipamentos diversos usam a comunicação USB, tornando simples, prático e rápido o uso dos mesmos.

Desta forma, o USB é uma solução viável para comunicação entre computadores e dispositivos externos, sendo muito usado em diversos trabalhos acadêmicos (MARCHESAN, 2006) (CHELLA, 2006) (SILVA, 2007).

### **Arquitetura e visão geral de funcionamento**

Pode-se dividir o protocolo de comunicação em três partes, o *host*, o dispositivo e a interconexão entre eles. Na interconexão é definida a topologia de funcionamento do barramento USB. A topologia é do tipo estrela, onde um *hub* USB é o centro de cada estrela e os dispositivos conectados a ele são as pontas. Cada conexão entre um *host* e um *hub* ou função é uma conexão ponto a ponto, de acordo com a figura 10.

O *host* USB é responsável por detectar quando um dispositivo é conectado ou desconectado, pelo gerenciamento do fluxo de controle, pelo fluxo de dados, por consultar estados e ainda fornecer energia para os dispositivos a ele conectados. Os *hubs* são elementos fundamentais para a característica de *plug and play* do USB, pois permitem que um dispositivo seja instalado e automaticamente detectado pelo sistema operacional. São denominados concentradores de conexão, pois permitem a conexão múltipla do USB. Os pontos de conexão são chamados de portas e cada *hub* converte um único ponto em um múltiplo ponto de conexão. O protocolo de comunicação está tanto no *host* (computador) quanto no dispositivo. No primeiro, ele se apresenta no formato de *drivers* (especificações e funções de controle via software), já no dispositivo ele está presente em um microcontrolador específico, como um PIC (Controlador Integrado de Periféricos – *Integrated Peripheral Controller*) ou em um *chipset* (conjunto de circuitos integrados).

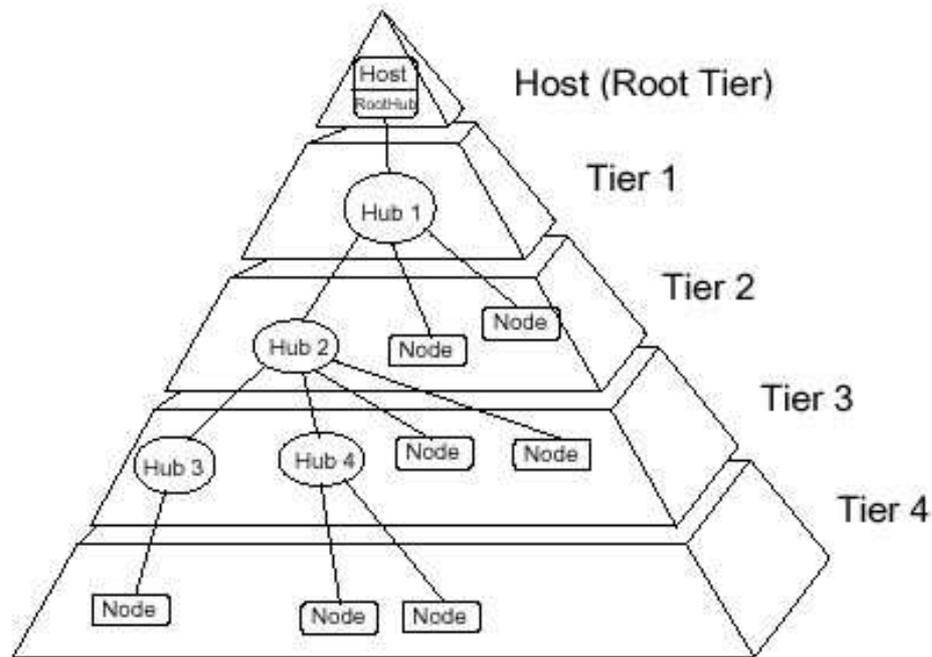


Figura 10. Topologia de funcionamento do USB (USB.ORG, 2010).

O protocolo USB possui quatro tipos de transferência:

- **Controle:** usadas pelo *host* para configurar o dispositivo conectado, de forma a requisitar informações sobre o mesmo.
- **Interrupção:** os dispositivos fazem as requisições para o *host*, utilizando pouca banda do sistema USB. Teclados e mouses usam esta forma de transferência, já que não necessitam enviar e receber dados de forma contínua.
- **Isossíncrona:** transferência de dados contínua, sem detecção de erro e checagem para chegada correta de dados. Dispositivos de vídeo e áudio utilizam este tipo, pois precisam de velocidade alta e constante.
- **Bulk:** transferência de grande volume de dados, com detecção de erro, correção e recuperação de dados corrompidos, largura de banda flexível, tráfego seguro e velocidade não contínua. Impressoras e discos de memória *flash* (*pen drives*) utilizam este tipo de transferência.

Na comunicação USB o protocolo faz a comunicação entre *host* e dispositivo através do envio de pacotes (Figura 11). Um pacote chamado *token* gerado pelo *host* inicia a comunicação, definindo se o próximo pacote é de escrita ou leitura. Existem dois pacotes na

fase *token*, que definem alta ou baixa velocidade de transmissão dependendo do tipo de dispositivo conectado. O próximo pacote é o *data* que contém os dados a serem lidos ou escritos, por fim o pacote *handshake* informa se houve falha ou sucesso na transmissão.

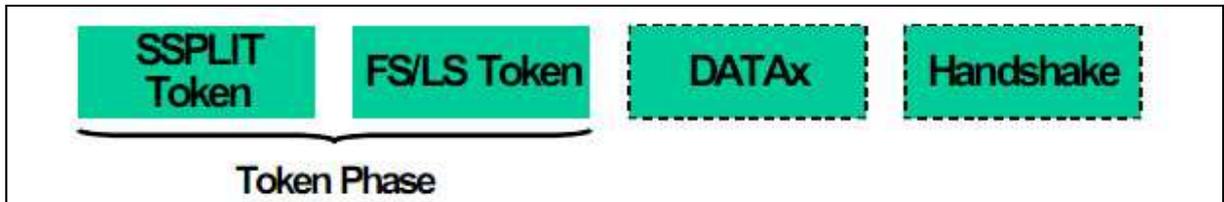


Figura 11. Composição genérica de uma transmissão USB (USB.ORG, 2010).

O protocolo USB sofreu modificações e melhorias ao longo dos anos, assim, surgiram algumas versões que definiram as características de seu funcionamento. As informações das principais versões estão descritas na tabela 5.

Tabela 5: Versões e características do protocolo USB.

| Versão        | Características  |
|---------------|--|
| USB 1.0 e 1.1 | V 1.0: Lançado em 1996 e V 1.1 em 1998 para corrigir problemas encontrados na versão anterior, com velocidade de transmissão de 1.5 ( <i>Low speed</i> ) a 12 Mbps ( <i>Full speed</i> ).  |
| USB 2.0       | Lançado em 2000 com velocidade de transmissão de até 480 Mbps ( <i>High Speed</i> ). Devido o aumento da velocidade apresentou uma solução viável para aplicações multimídia, o que aumentou o seu leque de utilidades.  |
| USB 3.0       | Lançado recentemente (2009) com velocidade de transmissão de 4.8 Gbps ( <i>SuperSpeed</i> ), sendo <i>full-duplex</i> transferindo dados bidirecionalmente. O seu uso ainda é restrito a poucos equipamentos, mas deve ser muito usado em vários equipamentos nos próximos anos, principalmente em dispositivos de armazenamento em massa. |

### 2.5.3 Comunicação entre computador e motores

Para efetivar a comunicação e controle de um motor de passo, por meio de um computador, existem duas formas de implementação comumente usadas, que compõem os *drives de controle dos motores de passo*. Na primeira forma, um componente que pode ser facilmente encontrado e vendido comercialmente, possui todo o circuito elétrico e de potência que permite o acionamento dos motores de passo, que normalmente utilizam três entradas de controle, *direção, pulso e habilita/desabilita*. Na segunda forma, se encaixam diversos trabalhos para aplicações específicas, onde um circuito elétrico e de potência específico é desenvolvido utilizando componentes de baixo custo. A lógica do circuito é composta de entrada de dados via *software* que especificam o acionamento de cada um dos motores, por meio da energização seqüencial de cada bobina dos motores controlados. A quantidade de *bits* de entrada pode variar nos circuitos personalizados, de acordo com as características do projeto. Abaixo, as figuras 12 e 13 detalham estas duas formas de controle.

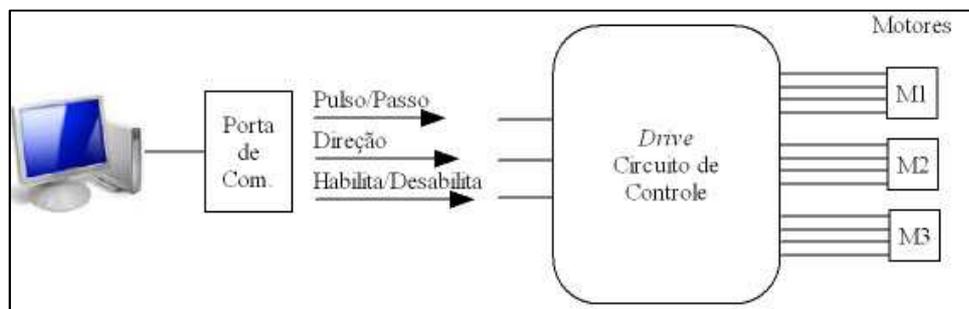


Figura 12. Controle de motores de passo usado em *drives* comerciais.

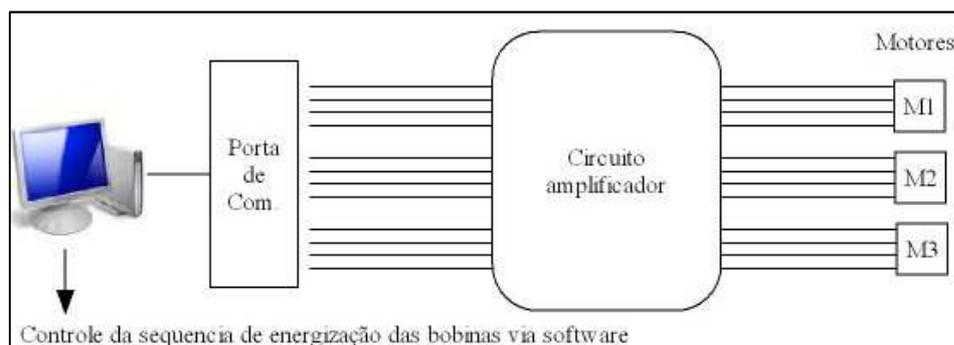


Figura 13. Controle de motores de passo usando circuitos personalizados.

## 2.6 Linguagem de Modelagem Unificada – UML

O uso de modelagem dinâmica e comportamental baseada em modelos, em sistemas/*softwares* aplicados a nível de hardware e software, é uma prática constante que auxilia no entendimento e compreensão da técnica de desenvolvimento usada. Essa compreensão objetiva facilitar o uso futuro da técnica por outros pesquisadores e desenvolvedores, possibilitando também a evolução e melhoria de um modelo ou metodologia. É com esse objetivo que alguns diagramas são usados para a representação da técnica de controle desenvolvida neste trabalho.

Neste contexto a Linguagem de Modelagem Unificada - UML (*Unified Modeling Language*) desponta como uma opção consistente para descrição estrutural ou comportamental de sistemas ou *softwares* para diversos objetivos, pois é amplamente usada em trabalhos atuais, como por exemplo, ambientes computacionais para práticas educacionais (OLIVEIRA et al, 2009), aplicações para monitoramento e controle de processos industriais (VILLEGAS et al, 2008), aplicações de automação industrial (MOURA e GUEDES, 2007) e sistemas de controle e gestão da produção (AMARAL FILHO, 2005).

Para apresentar a UML, dois conceitos de *Orientação a Objetos* são importantes: o conceito de classe e o conceito de objeto. Um *objeto* pode ser qualquer coisa que se possa descrever e ser representada como um objeto, e essa representação pode ser criada e manipulada, para representar como usar este objeto (uma cadeira, um livro, podendo ser também intangível, como um cargo ou presença). A representação de um conjunto de regras para gerenciar a representação de um objeto é chamada de *classe*. Uma classe define um *template* (modelo), que descreve como criar uma representação precisa de um tipo de objeto específico (PENDER, 2004).

De acordo com Guedes (2005), a UML é uma linguagem visual utilizada para modelar sistemas computacionais por meio do paradigma de Orientação a Objetos. Essa linguagem se tornou a linguagem-padrão de modelagem de software adotada internacionalmente pela indústria de Engenharia de Software.

Focando no desenvolvimento de sistemas, a linguagem UML permite especificar iterações de atores diretos, visualizar e documentar partes, modelar elementos, relacionamentos e diagramas de representação, para assim construir uma estrutura consistente que direcione os desenvolvedores a melhor prática de um sistema *software*. A UML surgiu da união de três metodologias de modelagem, o método de *Grady Booch*, o método *OMT*

(*Object Modeling Technique*) de Jacobson e o método **OOSE** (*Object-Oriented Software Engineering*) de Rumbaugh. O trabalho unificado dos métodos destes três profissionais resultou, em 1996, no lançamento da primeira versão da UML (GUEDES, 2005).

Em seguida, várias empresas atuantes na área de engenharia e desenvolvimento de *software* começaram a contribuir com o projeto, tendo a OMG (*Object Management Group*) adotado a UML como linguagem padrão para a modelagem de sistemas. Desta forma diversas versões revisadas, melhoradas e ampliadas foram surgindo, até chegar à versão atual, a UML 2.0.

Para atingir estes objetivos, os diagramas de representação UML são divididos em dois tipos de visões, a estrutural/estática e a comportamental/dinâmica. Na *modelagem estrutural* sistemas sem características dinâmicas podem ser modelados exclusivamente através da descrição de suas partes e de como elas se interligam, já a *modelagem dinâmica* é capaz de registrar o comportamento do sistema ao longo do tempo, mas também pode necessitar de modelagem estrutural (SILVA, 2007).

A seguir é relacionada uma breve descrição dos diagramas que compõem a UML, expondo o objetivo e o modo de funcionamento de cada tipo de diagrama (PENDER, 2004).

### 2.6.1 Visão Estrutural/Estática e Visão Comportamental/Dinâmica

#### **Visão Estrutural: Estática.**

- Diagrama de Classes: Um elemento abstrato que representa e especifica os objetos definindo suas características (atributos e métodos). É um diagrama muito importante que define as classes usadas no sistema, estabelecendo como elas se relacionam e trocam informações entre si;

- Diagrama de Objetos: Mostra os objetos que são instanciados das classes, servindo de complemento ao diagrama de classes, fornecendo uma visão dos valores armazenados pelos objetos;

- Diagrama de Pacotes: Descreve os pacotes ou pedaços do sistema divididos em agrupamentos lógicos, mostrando a dependência entre eles;

- Diagrama de Estrutura Composta: É proposto na versão 2 da UML, e visa descrever a visão cooperativa entre um conjunto de processos na execução de uma função específica;

- Diagrama de Componentes: Organiza as classes por meio de componentes de trabalho, sendo natural um componente corresponder a um conjunto de classes;
- Diagrama de Implantação: Descreve os componentes de hardware e software que representam a arquitetura de um sistema em que estão ligados seus respectivos componentes. São importantes porque modelam a plataforma de hardware para o sistema, identificando as características de hardware que afetam o desempenho e a configuração do *software*.

### **Visão Comportamental e de Iteração: Dinâmica**

- Diagrama de Casos de Uso: Mostra no mais alto nível a organização e o fluxo comportamental do sistema;
- Diagrama de Máquina de Estados: Foco nas relações de mudança de estado de um sistema orientado por eventos. Procura acompanhar as mudanças ocorridas nos estados, baseando-se em um caso de uso e se apoiando no diagrama de classes, para demonstrar a existência de um objeto;
- Diagrama de Atividades: Ênfase no fluxo de controle entre os objetos do sistema. Descreve o conjunto de ações ou passos a serem percorridos para a conclusão de uma atividade específica;
- Diagrama de Interatividade: Uma variação do diagrama de atividades, mostrando o fluxo de trabalho das atividades em uma seqüência de eventos, como realizar um pedido, finalizar um contrato, etc.;
- Diagrama de Seqüência: Focaliza a relação de tempo entre o envio e recebimento de mensagens entre os objetos que compõem o sistema, mostrando os objetos interagindo, e a interação entre um transmissor e um receptor;
- Diagrama de Comunicação: Mostra a organização estrutural dos objetos que enviam e recebem mensagens, assim como diagrama de seqüencia, mas representando um formato alternativo para descrever a interação de objetos. Não se preocupa com a temporalidade do processo e concentra-se em como os objetos estão vinculados;
- Diagrama de Tempo: Mostra o comportamento de objetos e sua interação em uma escala de tempo dando ênfase a mudanças ocorridas no período. São importantes para documentar claramente os requisitos de temporização que controlam as mudanças de estado.

O objetivo do uso destes diagramas é fornecer diversas visões do sistema que será modelado, observando vários aspectos e procurando a iteração entre os modelos,

possibilitando que um diagrama complemente o outro, ajudando a detectar e corrigir problemas durante a fase de desenvolvimento do *software*. Alguns diagramas enfocam o sistema de forma geral, com uma visão externa, como é o caso dos diagramas de Casos de Uso, e outros buscam uma visão mais específica e profunda do sistema. O uso de diversos diagramas é importante para a definição das características de um determinado sistema, mas um sistema não precisa ser modelado utilizando todos estes diagramas (GUEDES, 2005).

Cada diagrama possui um propósito e visão específica, assim como cada sistema a ser modelado também os possui. Desta forma, o uso de alguns destes diagramas é o suficiente para destacar o entendimento da técnica usada no desenvolvimento de um sistema.

### **3 DESENVOLVIMENTO**

A inovação crescente em diversos projetos de equipamentos automáticos programáveis exige melhorias nos sistemas de controle para atender as suas particularidades de funcionamento. Os sistemas automatizados são fundamentais para a redução de custos e para o aumento da segurança (substituindo atividades perigosas e trabalhos repetitivos humanos). Atualmente, estes sistemas exigem também alta flexibilidade e adaptabilidade a diversos equipamentos e ambientes de atuação. Esta flexibilidade possibilita o controle de um maior número de equipamentos existentes, com suas mais variadas características e configurações.

Outra característica importante é a usabilidade, pois um sistema além de prover inúmeras características, deve possibilitar ao operador/usuário um aprendizado rápido das suas funcionalidades, tendo como consequência o aumento da produtividade e também o maior controle sobre o projeto, permitindo ao operador detectar falhas, corrigi-las de maneira pró-ativa e propor melhorias no processo de controle. Melhorar o processo de controle destes equipamentos é fundamental, por exemplo, para o setor industrial, porque muitas vezes fazem uso de equipamentos personalizados ou dedicados a funções específicas e também equipamentos que são resultados de projetos realizados em ambiente acadêmico.

De acordo com De Paula e Santos (2008), o ambiente industrial deve continuamente adaptar-se a mudanças de mercado, pois o crescimento da competição global força as empresas a reduzir o tempo de resposta de lançamento de seus produtos e a oferecer preços competitivos. Um sistema de controle flexível, configurável e adaptável as características dos equipamentos, contribui para a competitividade deste setor, e também pode ser fundamental para o projeto e testes de novos equipamentos. No contexto da competitividade, os sistemas de controle usados nos equipamentos industriais, podem garantir o desempenho e a qualidade do processo de fabricação, pois, com maior flexibilidade os sistemas se adaptam as diversas características e/ou restrições existentes, como por exemplo, restrições técnicas relativas às características funcionais dos equipamentos (motores usados, ambiente de atuação, velocidade e aceleração, entre outras), a infra-estrutura da indústria e também restrições orçamentárias.

Este capítulo está dividido em duas partes: a primeira parte (seção 3.1) apresenta um estudo sobre as características fundamentais para compor uma metodologia de controle

flexível; a segunda parte (seção 3.2) descreve a metodologia de controle proposta, dando ênfase à técnica usada e aos seus modelos representativos.

### **3.1 Análise de requisitos para automação flexível**

Sistemas de controle flexíveis ou configuráveis são importantes para o projeto e desenvolvimento de diversos equipamentos de com objetivos específicos. O projeto de equipamentos programáveis fazendo uso de motores de passo para acionamento de mecanismos faz com que os sistemas de controle (*hardware e software*) sejam cada vez mais utilizados e exigidos. Alguns sistemas comerciais são usados para controlar estes equipamentos, mas muitas vezes, devido às particularidades dos projetos, algumas características e funcionalidades podem não ser atendidas plenamente, e o tempo gasto em desenvolver o conjunto *hardware/software* específico acaba dificultando testes e atrasando os projetos. Um exemplo prático é o ambiente acadêmico, onde muitos pesquisadores utilizam sistemas de controle que servem de base para aprendizado e impulsionam a imaginação de novos projetistas em áreas como, automação industrial, engenharia mecânica e elétrica, ciência da computação, entre outras.

Existem alguns exemplos de projetos de equipamentos desenvolvidos para aplicações específicas dentro da indústria, para várias finalidades, substituindo trabalhos repetitivos, de risco ou prejudiciais a saúde humana. Magalhães (2009) apresenta o desenvolvimento do protótipo de um robô, que consiste de um carrinho que transporta itens identificados com códigos de barras, e, guiado por um trilho em uma maquete desenhada no piso, sendo o seu objetivo identificar o compartimento correspondente ao item e descarregá-lo no local correto. Kamogawa e Teixeira (2009) descrevem o protótipo de um autoamostrador (química analítica) de baixo custo, movimentado por motores de passo, posicionado no centro de uma placa de fibra de madeira, com cinquenta furos para colocação das amostras, fazendo movimentos horizontais e verticais para se posicionar junto a cada um dos frascos de amostras.

Devido a quantidade crescente de projetos, no âmbito dos sistemas de controle para equipamentos automáticos programáveis, é notável a necessidade de adaptação constante às mudanças de projeto e dos próprios equipamentos. Desta forma, a evolução dos sistemas de

controle para sistemas cada vez mais flexíveis, torna-se um objetivo fundamental que traz benefícios a todos os projetistas.

Estes sistemas de controle também são amplamente utilizados em projetos acadêmicos, como por exemplo, um dispositivo programável que permite a mobilização dos membros inferiores imediatamente no pós-operatório, promovendo a manutenção dos tecidos que envolvem a articulação, destacando o uso da tecnologia de comando numérico e propondo um modelo que opere de maneira flexível, fornecendo possibilidades de programação por parte do usuário (SPERB, 2006). Outros projetos surgem com frequência e com as mais variadas complexidades de controle, tornando difícil e de alto custo encontrar sistemas de controle que atendam as necessidades específicas. Estas necessidades conflitam com a dificuldade de alteração de muitos *softwares*, por se tratarem de sistemas comerciais, diminuindo o nível de manipulação e adaptação, o que é considerado uma limitação para os projetos.

Este capítulo objetiva reunir as características fundamentais para os sistemas de controle de máquinas automáticas programáveis, servindo de base para o desenvolvimento de uma metodologia de controle, que contemple características de flexibilidade ou adaptabilidade. Tomando como base a análise de alguns *softwares* de controle para máquinas automáticas programáveis que são comumente usados, e técnicas propostas na literatura atual, foi realizado um estudo focando nas funcionalidades ou necessidades principais de controle, enfatizando a necessidade de flexibilidade para controle dos mais variados tipos de equipamentos programáveis.

### 3.1.1 O uso de motores de passo

Uma particularidade importante das máquinas automáticas programáveis é o uso de motores de passo em seus projetos. Estes motores podem agregar características fundamentais, como grande precisão de movimentos, controle de aceleração e desaceleração, entre outras. O seu controle requer sistemas elétricos que podem ser desenvolvidos de maneira configurável, incluindo diversas características, como o controle de vários motores de passo ao mesmo tempo e a porta de comunicação usada, o que aumenta a portabilidade a diferentes equipamentos.

Os motores de passo são amplamente usados em sistemas robóticos e no controle de máquinas, onde eles têm que realizar operações de posicionamento de alta precisão (MELIN e CASTILHO, 2004). De acordo com Muñiz et al (2008), um motor de passo atua como um preciso acionador mecânico, tendo um dispositivo elétrico dividindo a rotação do motor em um grande número de passos, proporcionando movimentação precisa. Velocidade e aceleração podem também ser precisamente controladas por meio de cálculos do tempo exato de envio dos comandos entre cada passo.

Há muitos trabalhos que utilizam motores de passo como atuadores devido a sua facilidade de controle, sendo projetos com as mais variadas características e complexidades, como por exemplo, equipamentos que fazem a obtenção de propriedades mecânicas em dutos de petróleo (DE SOUZA, SOARES e FRANCO, 2008); um projeto de ambiente de aprendizagem remoto (via WEB) que permite a prática laboratorial e aprendizado de física, tendo um equipamento controlado remotamente fazendo experimentos (OLIVEIRA et al, 2009); um posicionador eletrônico para estruturas micrométricas, utilizado para posicionar objetos com passos tridimensionais e micrométricos, aproveitando a facilidade de controle e a precisão dos motores de passo (CALIXTO, 2009); entre outros trabalhos que podem ser encontrados na literatura.

A forma de controle de motores de passo vem a ser uma das características que fazem com que os sistemas de controle para as máquinas automáticas programáveis sejam fundamentais para diferentes projetos. Sendo assim, a quantidade crescente de projetos e sistemas robóticos diferenciados alavanca a necessidade da existência de alternativas que possam acelerar o projeto e testes destes equipamentos, focando na flexibilidade e portabilidade dos sistemas de controle.

### 3.1.2 *Softwares* de controle para equipamentos programáveis: Características principais

Atualmente existem vários *softwares* que objetivam controlar equipamentos programáveis que usam motores de passo como atuadores. Muitas vezes estes softwares fornecem um número grande de funcionalidades e com interfaces complexas e pouco intuitivas, principalmente visando atender ao máximo as necessidades de controle dos mais variados equipamentos. Estas necessidades são muitas vezes mais objetivas do que a maioria das funções de alta complexidade oferecidas nos softwares comerciais. Em ambiente

acadêmico, por exemplo, existe a necessidade de demonstrar com clareza, objetividade, e com funções específicas, como pode ser feito o projeto e controle de equipamentos ou máquinas robóticas programáveis. Em pequenas e médias indústrias é importante o uso de softwares de baixo custo e que possibilitam alta portabilidade, pois, recursos para investimento em altas tecnologias neste setor são limitados. Seguindo estas características, observa-se que são pontos fundamentais na aprendizagem e na formação de novos projetistas, sendo necessário o uso de sistemas/*softwares* de controle flexíveis e portáveis a objetivos específicos.

O estudo toma como referência características encontradas em *softwares* comumente usados, destacando aqueles pontos considerados indispensáveis para compor um sistema de controle flexível, portátil e de fácil aprendizado. A seguir é apresentada uma descrição geral deste estudo, com suas principais características, bem como, aspectos considerados críticos para o controle de diversos equipamentos programáveis.

Os *softwares* de controle estudados são específicos para máquinas CNC, e amplamente usados em vários projetos relacionados à automação e controle.

## ***KCAM***

É um *software* desenvolvido independentemente pelo grupo americano *Kellyware*. Permite trabalhar com projetos de até 500 linhas de código G (linguagem de programação usada no controle de máquinas CNC) em sua versão para testes, pois, atualmente é um *software* proprietário, tendo custo de \$ 95 dólares por licença de uso.

O *KCAM* possibilita o controle de sistemas com até 3 eixos, com visualização gráfica em tempo real e podendo interromper e executar passo a passo o projeto. A sua interface gráfica agrega várias funções com boa disposição na tela, facilitando o acesso e o entendimento do operador, possibilitando controle de pulsos manuais nos motores, a visualização do código G na tela e o gráfico 3D do projeto com possibilidade de zoom (*KCAM User Manual*). Foi desenvolvido para o ambiente *Windows*, utilizando a porta paralela como meio de comunicação entre o equipamento e o computador. Este software trabalha com muitos tipos de máquinas programáveis, mas com recomendações de uso, pois é mais indicado para uso pessoal ou para estudantes do que para uso profissional.

Principais características relacionadas ao controle de motores de passo:

- Comunicação pela Porta Paralela, e controle de 3 motores/eixos;
- Visualização dos comandos que estão sendo executados;
- Movimentos Manuais;
- Configuração de *compensação (Backlash)* por eixo – compensar erros/folgas em todos os eixos.
- Aceleração e Desaceleração (*Ramping Rate*) por eixo ( $\text{mm}/\text{min}^2$ );

### ***MACH3***

O *software Mach3* desenvolvido pela empresa americana *ArtSoft*, possui licença de uso atualmente em \$ 175 dólares. Em sua versão para testes (gratuita) possibilita trabalhar com projetos de até 500 linhas de código G, controlando sistemas com até seis eixos. Durante o processo de controle permite a visualização do código G, o andamento das operações linha a linha, a visualização gráfica do projeto, e ainda, os pulsos manuais para possíveis ajustes no projeto. Foi desenvolvido para o sistema operacional *Windows*, utilizando a porta paralela como interface de comunicação (*Art Soft MACH3. Documentation*). De acordo com Carvalho, Dutra e Bonacorso (2008), a sua interface é flexível, podendo personalizar o ambiente de controle de acordo com as variações do projeto, apresentando também bibliotecas de controle para telas sensíveis ao toque (*touchscreen*).

De modo geral a sua interface é repleta de configurações, o que acaba tornando-a pouco intuitiva. O *Mach3* trabalha com diferentes equipamentos, com a utilização da porta paralela ou porta USB como interface de comunicação entre o equipamento e o computador pessoal. É muito utilizado no meio acadêmico para auxiliar no projeto e testes de novos equipamentos.

Principais Características relacionadas ao controle de motores:

- Controla 6 eixos/motores, 3 prismáticos e 3 rotacionais;
- Visualização 3D do projeto, e dos comandos que estão sendo executados;
- Controle dos motores através da porta paralela e porta USB;

- Configuração de *compensação Backlash* por eixo – compensar erros/folgas em todos os eixos;
- Controle de Velocidade Máxima em mm/min e Aceleração em mm/seg<sup>2</sup>;

### ***TurboCNC 4***

O software *TurboCNC* na versão 4.01, desenvolvido pela empresa americana *DAK Engineering* e com licença de uso de \$ 60 dólares, é um *software* proprietário muito utilizado em trabalhos acadêmicos e profissionais. O *software* permite o controle de sistemas com até 8 eixos simultaneamente. Permite a visualização do código G na tela, acompanhando a execução do programa linha a linha, interrompendo e continuando a execução em qualquer ponto, para possibilitar possíveis ajustes no projeto. Esta versão não permite visualização gráfica do projeto e a geração de pulsos manuais, pois o software trabalha em ambiente DOS, não possuindo muitas opções gráficas de visualização e controle. Desta forma a interface de controle não possui opções intuitivas ou personalização, o que acaba prejudicando o seu uso. A porta de comunicação utilizada para controle é a porta paralela, sendo facilmente configurado entre um computador pessoal e o drive de controle do equipamento (*TURBO CNC 4.01 User Manual*).

Atualmente existe uma nova versão (*Turbo CNC 5*) possuindo as mesmas características e funcionalidades, mas com um ambiente gráfico para *Windows* que proporciona maior produtividade. Esta nova versão foi desenvolvida pela empresa brasileira *A.S.T Indústria* em conjunto com a *DAK Engineering*.

Principais características relacionadas ao controle de motores de passo:

- Controle de sistemas com 8 eixos/motores;
- Comunicação através da porta paralela;
- Visualização dos comandos que estão sendo executados;
- *Backlash* - Configuração de compensação de folgas em todos os eixos;
- Recurso de operação manual de aceleração: Modo “*Jog*” – configuração de movimento contínuo ou discreto ao pressionar uma tecla de movimento do eixo. O modo

contínuo causa o movimento em velocidade contínua enquanto a tecla de movimento do eixo for pressionada e o modo discreto causa um movimento de desaceleração no eixo até parar.

## ***EMC2***

Desenvolvido inicialmente pela NIST (*National Institute of Standards and Technology*) uma organização do governo americano, atualmente disponível sob licença GPL como software livre, o software EMC2 possibilita controlar sistemas com até 9 eixos simultaneamente, podendo-se informar pré configurações na sua abertura, que podem servir de base para vários projetos. Possui acesso rápido na tela para várias configurações, permite a visualização do código G de acionamento dos eixos, a inserção de movimentos manuais e a visualização gráfica do projeto em tempo real. Sua interface gráfica de controle possui fácil acesso a maioria das configurações, além de ser muito intuitiva, contribuindo muito para a facilidade de uso do software (*EMC2 Documentation*).

Por ser um software livre, permite modificações e adaptações, trabalhando muito bem no projeto Linux UBUNTU, nas distribuições *Ubuntu 6.06 “Dapper Drake” LTS* e *Ubuntu 8.04 “Hardy Heron” LTS*, onde o EMC2 vem instalado e pronto para uso. Para a comunicação entre o sistema e o computador, a porta paralela é usada, como na maioria dos *softwares*. O EMC2 possui uma vantagem importante com relação a sua configuração, que é adaptar-se a máquina a ser controlada, pois possui o seu código fonte disponível para modificações.

Principais características relacionadas ao controle dos motores:

- Controla até 9 eixos/motores;
- Configuração de velocidade máxima do eixo e distância a percorrer até atingir esta velocidade, uma forma de aceleração;
- Compensação de folga em movimentos inversos (*backlash*) – com a edição de um arquivo “INI”, de configuração na inicialização;
- Comunicação pela porta paralela;
- Possui visualização 3D e acompanhamento em tempo real dos comandos executados.

A tabela 6 lista as principais características e funcionalidades dos quatro softwares estudados:

Tabela 6: Principais características encontradas dos softwares estudados.

| <b>Programas / Funções</b>                         | <b>KCAM<br/>v4.0.0</b>         | <b>MACH3<br/>v 3.042.020</b> | <b>Turbo CNC<br/>v 4.01</b> | <b>EMC2</b>                           |
|--|--------------------------------|------------------------------|-----------------------------|---------------------------------------|
| Valor da Licença                                   | \$ 95                          | \$ 175                       | \$ 60                       | \$ 0                                  |
| Número de Eixos controláveis do Sistema            | 3                              | 6                            | 8                           | 9                                     |
| Compensação de folga – <i>Backlash</i>             | Sim                            | Sim                          | Sim                         | Sim                                   |
| Controle de Aceleração e Desaceleração             | Sim                            | Sim                          | Sim, Manual                 | Sim                                   |
| Visualização e acompanhamento de comandos na tela. | Sim                            | Sim                          | Sim                         | Sim                                   |
| Geração de Pulsos Manuais                          | Sim                            | Sim                          | Não                         | Sim                                   |
| Visualização do Gráfico do Projeto                 | Sim                            | Sim                          | Não                         | Sim                                   |
| Porta de comunicação usada                         | Porta Paralela                 | Porta Paralela e USB         | Porta Paralela              | Porta Paralela                        |
| Simulação gráfica do projeto em tempo real         | Passo a Passo e Automática     | Automática                   | Não                         | Passo a Passo e Automática Tempo Real |
| Sistema Operacional suportado                      | Windows 98(ideal) ou superior. | Windows XP ou superior       | DOS                         | Linux                                 |

### 3.1.3 Análise das informações encontradas

Para levantar características consideradas fundamentais para compor metodologias de controle flexíveis, algumas funções são observadas e enfatizadas devido ao fato de todos os *softwares* apresentarem e também de serem fatores comumente encontrados em problemáticas de controle.

A análise da portabilidade e adaptabilidade foi definida de acordo com duas características predominantes em todos os sistemas, a quantidade de motores controláveis e a porta de comunicação usada. Estas duas características fazem a diferença no momento de definir quais os equipamentos e projetos que o sistema pode controlar, diferenciando e destacando estas funcionalidades, com o intuito de verificar a sua indispensabilidade em sistemas de controle atuais.

Outras funções parametrizáveis são consideradas diferenciais para a eficiência no controle dos equipamentos, como compensação de folga para a inversão de movimentos (configuração de *backlash*), controle de aceleração e desaceleração, velocidade máxima, todos configuráveis por eixo/motor controlado. Também foram consideradas outras características relacionadas ao modo de visualização do andamento do projeto, como a visualização gráfica, comandos que estão sendo executados, interrupção para movimentos manuais, além do custo para uso destes sistemas. Finalmente chega-se a um conjunto de características indispensáveis para compor metodologias que possibilitem o controle dos mais variados projetos, contribuindo para sistemas de controle flexíveis, parametrizáveis e fáceis de usar.

Constatou-se que as funcionalidades dos *softwares* analisados aliadas a experiências existentes em trabalhos atuais e relacionadas na tabela 6, resultaram em um conjunto de características consideradas indispensáveis para a composição de uma metodologia mais flexível e suscetível a adaptações (tabela 7). Desta forma, estes dados servem de base para o desenvolvimento de uma metodologia de controle flexível, contribuindo e ajudando os projetistas nos testes de novos equipamentos, com maior rapidez, proporcionando ganho de tempo e diminuindo custos.

A tabela 7 mostra as características ou funcionalidades para compor sistemas de controle mais flexíveis.

Tabela 7: Características para compor uma metodologia de controle flexível.

| <b>Característica</b>                                    | <b>Obs:</b>  |
|--|--|
| Número de Eixos/Motores Controláveis                     | O número de motores controláveis aumenta as possibilidades de controle de acordo com a complexidade dos mais variados projetos.  |
| Porta de comunicação Usada                               | A porta de comunicação usada é um diferencial entre os sistemas, pois atualmente a porta USB é um padrão de comunicação superior (velocidade de transmissão e conectividade), e que é encontrada com facilidade em todos os computadores pessoais, fato que aumenta a portabilidade dos sistemas.  |
| Visualização Gráfica do Projeto                          | A possibilidade de visualizar graficamente a movimentação do equipamento e poder fazer modificações antes da execução final caracteriza também um importante diferencial para a correção e manutenção dos programas, diminuindo custos e facilitando o uso dos sistemas de controle por pessoas com pouca experiência na área.                                     |
| Compensação de folga - <i>Backlash</i>                   | Configuração de <i>compensação de folga em movimentos – Backlash</i> - por eixo, usada para compensar erros/folgas, perceptíveis quando o eixo faz movimentos inversos, eliminando a folga e aumentando a precisão do sistema de controle.   |
| Controle de Aceleração e Desaceleração por eixo          | Configuração do tempo e distância percorrida pelo motor para alcançar velocidade máxima constante e também para desacelerar até parar.   |
| Velocidade máxima por eixo                               | Limitação da velocidade de cada eixo/motor.  |
| Resolução de movimentação a cada pulso enviado, por eixo | Configuração da relação entre pulso e deslocamento individualmente em cada eixo, em milímetros.  |
| Codificação de pulsos/passos de forma simplificada       | A metodologia deve possuir um sistema de codificação de pulsos simplificado, facilitando a compreensão dos pesquisadores de diversas áreas. No caso dos <i>softwares</i> estudados a linguagem de codificação é o código G. A criação e utilização de linguagens de codificação simplificadas, com foco interdisciplinar facilita o uso de um sistema de controle. |

Com o uso de metodologias de controle flexíveis, pesquisadores de diversas áreas podem trabalhar com o projeto de equipamentos para os mais variados fins, como equipamentos industriais de pequeno porte, equipamentos que ajudam na recuperação de pessoas com deficiência física, equipamentos que auxiliam na aprendizagem com experimentos e simulações de situações reais, entre outras, promovendo a interdisciplinaridade.

### **3.2 Técnica de Controle e sua Representação**

Nesta seção são detalhados os aspectos metodológicos do trabalho, demonstrando as técnicas usadas e aplicadas. Na seção 3.2.1 é apresentada a metodologia de controle usada; na seção 3.2.1 é abordado à malha de controle usada; em seguida, a seção 3.2.3 apresenta os diagramas representativos da metodologia de controle proposta, para facilitar a sua aplicação futura.

#### **3.2.1 Metodologia de controle flexível para equipamentos automáticos programáveis**

Uma metodologia de controle flexível traz alternativas que proporcionam baixo custo e fácil adaptação aos mais variados equipamentos. A metodologia de controle flexível proposta neste trabalho, proporciona o projeto, controle e testes de equipamentos programáveis, aliando baixo custo com alta portabilidade. O foco é agregar valor e propor melhorias para os sistemas de controle desenvolvidos atualmente, aumentando as possibilidades de controle dos equipamentos, impulsionando atividades interdisciplinares e o aprendizado de automação e controle.

A metodologia proposta contempla características consideradas fundamentais em um sistema de controle, tornando-o parametrizável para controlar diversos equipamentos programáveis (ROQUE, 2010). Dentre estas, destacam-se a maneira ou linguagem de codificação de movimentação dos atuadores, considerada como dado de entrada do sistema, e a comunicação principal usada entre controlador e computador, proporcionando mais velocidade e confiabilidade no processamento das operações.

A informação de entrada do sistema é um arquivo contendo uma codificação de pulsos que representa a trajetória que o equipamento irá executar no ambiente de atuação. Por meio desta codificação o sistema de controle faz a leitura e interpretação de cada código, que possui uma letra do alfabeto representando um determinado movimento nos eixos/motores do equipamento. Esta forma de entrada de dados ou codificação de pulsos de movimentação (tabela 8) pode ser inserida manualmente no programa, representando uma trajetória conhecida, usada para testes nos equipamentos ou para movimentos de um determinado motor em sentido horário ou anti-horário. Outra forma de entrada de dados é o uso de um programa que converte a trajetória de um equipamento para códigos de movimentação (DE CRISTO, 2009), gerando um arquivo que pode ser carregado no *sistema/software* de controle.

Esta codificação simplificada serve de parâmetro para que sistemas de controle futuros possam ser desenvolvidos de maneira mais amigável, melhorando a característica de usabilidade, possibilitando maior flexibilidade de controle, bem como promovendo as atividades interdisciplinares. A tabela 8 mostra a codificação da movimentação para um equipamento programável através do uso de letras do alfabeto que representam avanços no sentido horário e anti-horário dos motores usados como atuadores. O sinal de “+” e “-” na descrição do eixo controlado indica o sentido (horário ou anti-horário) de movimentação do eixo.

Tabela 8: Codificação dos Pulsos para movimentação dos eixos.

| <b>Código do Pulso</b> | <b>Descrição do Movimento</b>        | <b>Eixo Controlado</b> |
|------------------------|--------------------------------------|------------------------|
| A                      | Avanço no Eixo X                     | +X                     |
| B                      | Avanço no Eixo Y                     | +Y                     |
| C                      | Avanço no Eixo Z                     | +Z                     |
| D                      | Retorno no Eixo X                    | -X                     |
| E                      | Retorno no Eixo Y                    | -Y                     |
| F                      | Retorno no Eixo Z                    | -Z                     |
| G                      | Avanço no Eixo X e Y                 | +X e +Y                |
| H                      | Avanço no Eixo X e Z                 | +X e +Z                |
| I                      | Avanço no Eixo Y e Z                 | +Y e +Z                |
| J                      | Retorno no Eixo X e Y                | -X e -Y                |
| K                      | Retorno no Eixo X e Z                | -X e -Z                |
| L                      | Retorno no Eixo Y e Z                | -Y e -Z                |
| M                      | Avanço no Eixo X e Retorno no Eixo Y | +X e -Y                |
| N                      | Retorno no Eixo X e Avanço no Eixo Y | -X e +Y                |
| O                      | Avanço no Eixo X e Retorno no Eixo Z | +X e -Z                |
| P                      | Retorno no Eixo X e Avanço no Eixo Z | -X e +Z                |
| Q                      | Avanço no Eixo Y e Retorno no Eixo Z | +Y e -Z                |

|   |                                      |             |
|---|--------------------------------------|-------------|
| R | Retorno no Eixo Y e Avanço no Eixo Z | -Y e +Z     |
| S | Avanço nos Eixos X, Y e Z            | +X, +Y e +Z |
| T | Retorno nos Eixos X, Y e Z           | -X, -Y e -Z |
| U | Avanço Eixo X e Retorno Eixos Y e Z  | +X, -Y e -Z |
| V | Avanço Eixo Y e Retorno Eixos X e Z  | -X, +Y e -Z |
| W | Avanço Eixo Z e Retorno Eixos X e Y  | -X, -Y e +Z |
| X | Retorno Eixo X e Avanço Eixos Y e Z  | -X, +Y e +Z |
| Y | Retorno Eixo Y e Avanço Eixos X e Z  | +X, -Y e +Z |
| Z | Retorno Eixo Z e Avanço Eixos X e Y  | +X, +Y e -Z |

Ao executar, por exemplo, a informação “AAAAAAAAAA BBBB GGGGGGGGGG” o sistema de controle envia dez pulsos de movimentação, no eixo X, cinco pulsos, no eixo Y, e depois dez pulsos em ambos os eixos X e Y. Esta codificação simplifica a leitura e adaptação dos sistemas de controle desenvolvidos, com melhoria no processamento, na simulação gráfica, e no envio dos pulsos para a porta de comunicação. Neste caso, a tabela trabalha com sistemas atuando com três eixos de coordenadas, cada eixo tendo um motor de passo como atuador, podendo ainda ser adaptada para uma quantidade maior de motores.

A figura 14 apresenta um diagrama representativo da metodologia de controle proposta neste trabalho, representada por meio de um fluxograma dinâmico, mostrando o fluxo de dados de controle da metodologia, dividindo em regiões os seus principais componentes. No diagrama os atores principais são o operador ou usuário do sistema de controle e o equipamento a ser controlado. Desta forma, um fluxo de informações de controle é intermediado por vários processos, estes são responsáveis por traduzir a entrada de dados feita pelo operador do sistema em movimentos nos atuadores do equipamento.

O processo inicial é à entrada de dados baseado na tabela de codificação de pulsos, em seguida, outro processo interpreta estes códigos para assim representar o movimento desejado no equipamento. Os processos de pré-visualização, simulação gráfica e movimentos passo a passo, são importantes para a representação visual prévia do movimento desejado e também para a correção proativa de possíveis erros de projeto. Estes processos formam um ciclo (*loop*) de operações entre o sistema e o operador, para posteriormente confirmar o envio dos pulsos para a porta de comunicação. Esta porta envia os dados para o circuito de controle, que converte os códigos em pulsos elétricos para realizar a movimentação dos motores no equipamento.

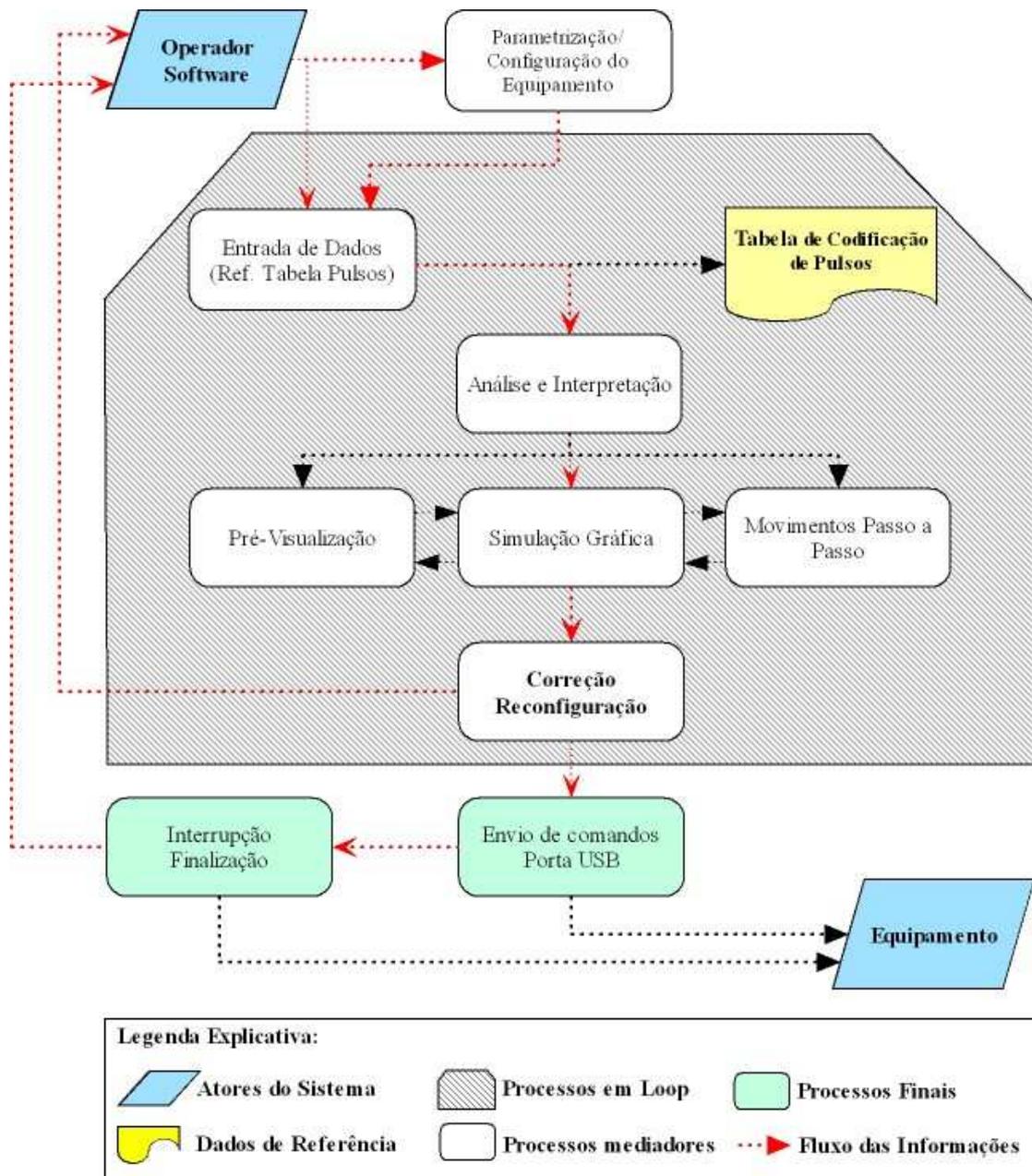


Figura 14: Diagrama representativo da metodologia em forma de modelo dinâmico.

O processo de *correção/reconfiguração* demonstra uma característica importante do modelo, pois neste ponto podem-se conferir as simulações realizadas, para em seguida decidir se os comandos podem ser enviados para a porta, ou se deve ocorrer uma interrupção para solucionar problemas de projeto.

A seção seguinte abordará aspectos relativos a malha de controle usada.

### 3.2.2 Malha de Controle

O foco da metodologia proposta é o projeto e desenvolvimento de *equipamentos automáticos programáveis*, comumente encontrados em diversos trabalhos. A base para a metodologia são os requisitos de desempenho e parametrização que foram especificados na seção 3.1, que demonstraram a necessidade de flexibilidade de controle com a parametrização dos atuadores. A movimentação de saída é geralmente conhecida e quando ocorrem distúrbios/erros no sistema, estes podem ser configurados e alterados previamente, fazendo uso da simulação gráfica em computador para correção de erros, dispensando assim uma correção automática em tempo de execução. Neste trabalho, o controle de equipamentos programáveis acionados por motores de passo, foi realizado usando a técnica de *controle em malha aberta*, devido à rapidez de implementação, baixo custo e principalmente porque os equipamentos projetados podem ter um operador realizando as correções de trajetória. Nestes equipamentos é muito importante permitir ao usuário/operador o ajuste e correção rápida de uma trajetória diretamente no sistema/*software* de controle.

A figura 15 ilustra como ocorre o controle na metodologia proposta.

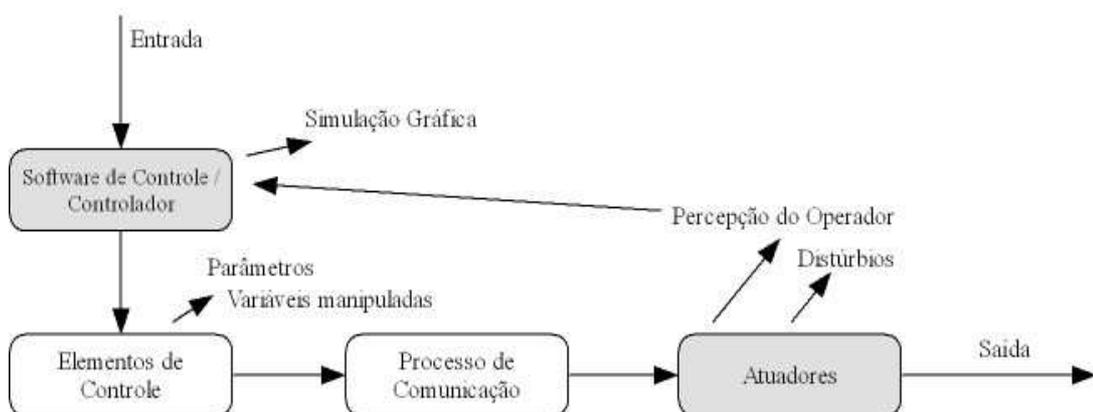


Figura 15: Malha de controle aplicada no software desenvolvido.

A técnica usada é destinada e aplicada no controle de diferentes tipos de equipamentos programáveis, aliando a facilidade de uso com as características de controle da metodologia. Estas características incluem:

- Tamanho do ambiente ou planta de atuação;
- Velocidade máxima (tempo de envio entre pulsos);
- Resolução do equipamento (distância em *milímetros* que o motor se move a cada pulso), usado para representação na simulação gráfica;
- Controle de aceleração e desaceleração por eixo (em  $\text{mm/s}^2$ );
- *Backlash* – pulsos de folga aplicados em movimentos inversos.
- Movimentos manuais ou passo a passo nos eixos X e Y.

### 3.2.3 Diagramas UML – Representação da Metodologia

Para contextualizar a metodologia proposta, descrevendo a sua dinâmica comportamental e mostrando a interação entre os atores do sistema, foi utilizada a modelagem UML – *Unified Modeling Language* (GUEDES, 2005).

A linguagem UML utiliza representações gráficas para descrever os vários elementos que compõem um sistema, permitindo sua visualização sob diferentes perspectivas. O seu uso é destinado à especificação de modelos, os quais posteriormente servirão de base para o desenvolvimento de um sistema executável. O objetivo principal da UML é fornecer uma linguagem padrão que permita modelar um sistema, permitindo a troca natural destes modelos entre os desenvolvedores de *softwares*. Com a UML é possível descrever eficazmente requisitos de *software*, caracterizar a arquitetura de um sistema e direcionar programadores, aumentando a produtividade e diminuindo os riscos (PENDER, 2004).

As diferentes partes que compõem um sistema/*software* são definidas em UML como partes estáticas (diagrama de classes, de objetos, de pacotes, estrutura composta, de componentes e de implantação) e partes dinâmicas comportamentais (diagrama de *casos de uso*, máquina de estados, *de atividades*, interatividade, de seqüências, de comunicação, e de tempo). Como já especificado, o objetivo da diagramação desta metodologia é descrever a dinâmica comportamental da metodologia de controle flexível para equipamentos automáticos programáveis, deste modo, dois diagramas da visão dinâmica da UML são usados, o diagrama de casos de uso e o de atividades.

A seguir a figura 16 apresenta o diagrama de casos de uso da metodologia.

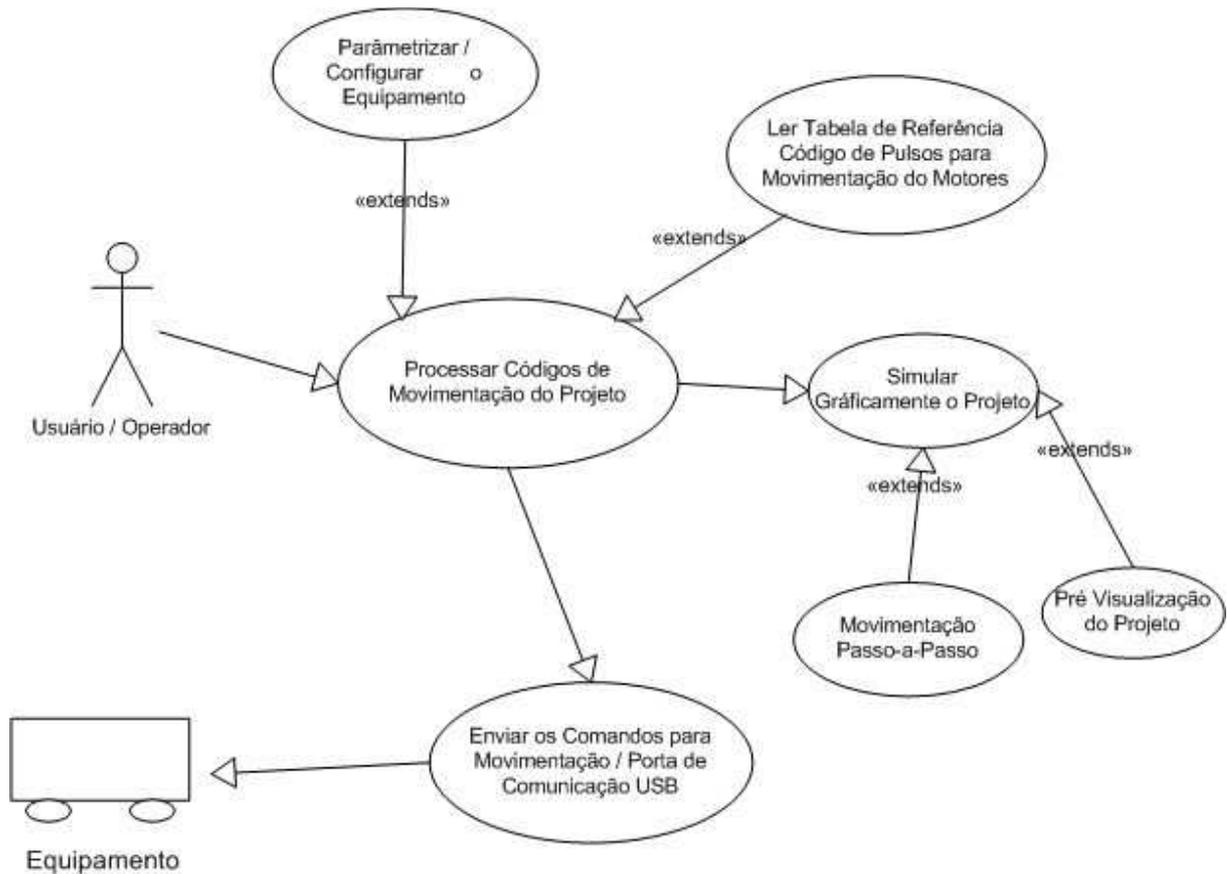


Figura 16: Diagrama de Casos de Uso para a Metodologia de Controle Flexível.

O diagrama de *casos de uso* (figura 16) é usado para representar os atores, que desempenham papéis fundamentais no sistema, mostrando o fluxo entre os processos ou funções que nesta metodologia o sistema de controle deve realizar. O operador (ator inicial), por intermédio do *software* de controle, inicializa o processo com a configuração do equipamento e leitura do arquivo contendo os códigos de pulsos. Este processamento pode ser simulado graficamente antes do envio dos comandos para a realização da movimentação do equipamento (ator final).

O *diagrama de atividades* (figura 17) é usado para mostrar a seqüência de atividades realizadas no sistema, mostrando o fluxo de trabalho de um ponto inicial até o ponto final, detalhando as decisões tomadas durante o caminho percorrido durante a execução das tarefas.

A figura 17 apresenta o diagrama de atividades, mostrando a dinâmica seqüencial de atividades da metodologia de controle, definindo o fluxo básico de controle, suas tomadas de decisões e processamentos ou ações.

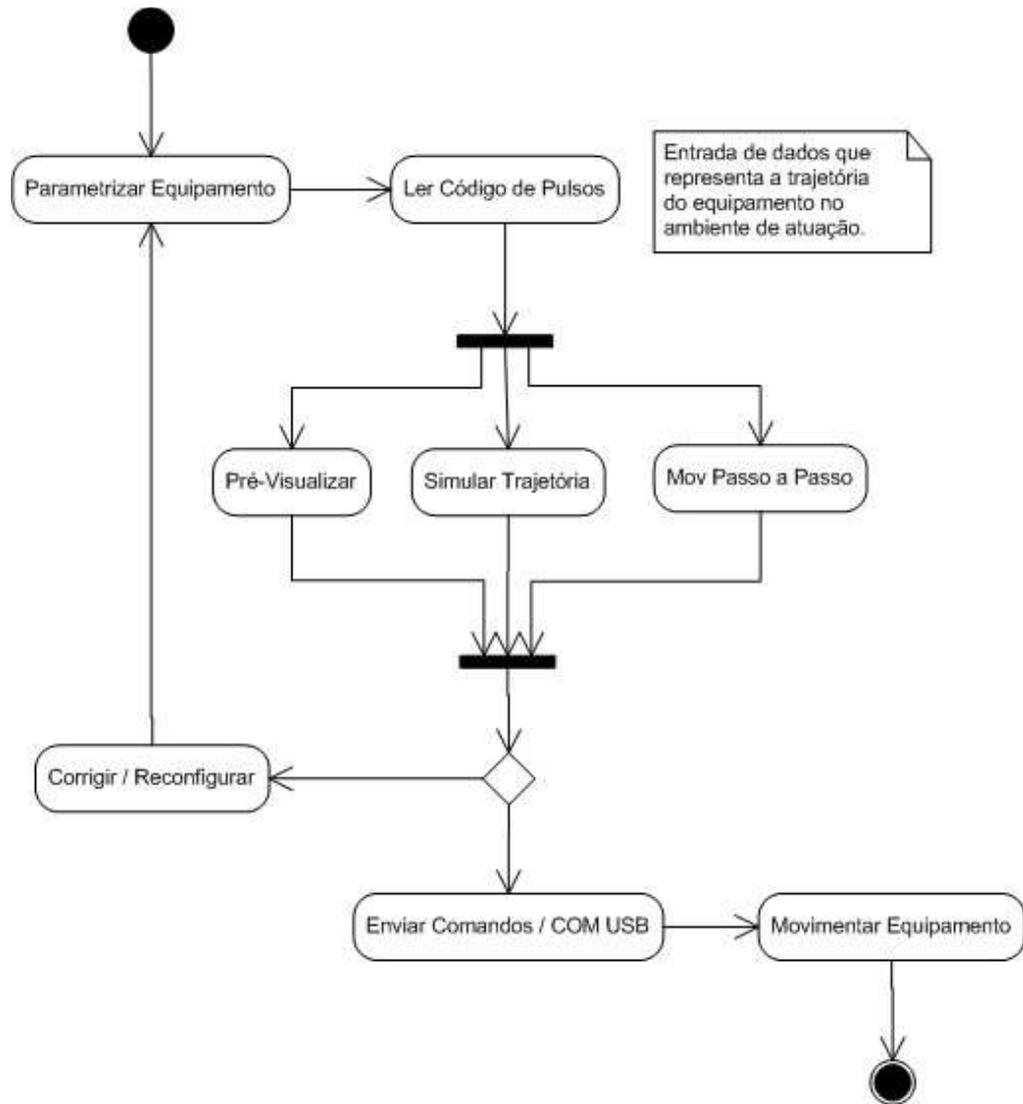


Figura 17: Diagrama de Atividades da Metodologia de Controle.

Na figura 17, dois pontos formados por círculos preenchidos na cor preta representam o estado inicial e o estado final do sistema, entre eles os processos e as tomadas de decisão formam o fluxo de controle (*workflow*). No diagrama de atividades uma atenção especial deve ser dada ao processo de correção ou reconfiguração do projeto, pois, esta tomada de decisão gera um novo processamento no sistema de controle para permitir a adaptação da trajetória inicial especificada, fazendo assim o movimento esperado no equipamento. O processo envia os comandos para o equipamento usando a comunicação por meio do protocolo USB, utilizando uma biblioteca de funções de controle (especificadas na seção seguinte), tendo como resultado o movimento dos motores do equipamento. Na seção seguinte são detalhados os resultados da pesquisa, descrevendo o *hardware* e o *software* desenvolvido de acordo com a metodologia de controle.

## 4 RESULTADOS

Nesta seção serão abordados os resultados da pesquisa, mostrando como foi aplicada e validada a metodologia. Na primeira parte será detalhada a composição do hardware de controle usado como meio de comunicação entre computador e equipamento, para em seguida, apresentar o software de controle desenvolvido para controlar equipamentos programáveis.

### 4.1 Hardware de controle

Para o controle de uma máquina automática programável é necessário um dispositivo de *hardware* que faça o papel de interface entre o *software* de controle configurado e instalado no computador e o equipamento a ser controlado. O *software* é responsável por toda a programação que proporciona um nível de flexibilidade para o controle do equipamento. O *hardware* é composto dos atuadores e sensores controlados e alimentados por meio de um circuito elétrico. A figura 18 demonstra a comunicação típica entre o computador e uma máquina automática programável que usa motores de passo.

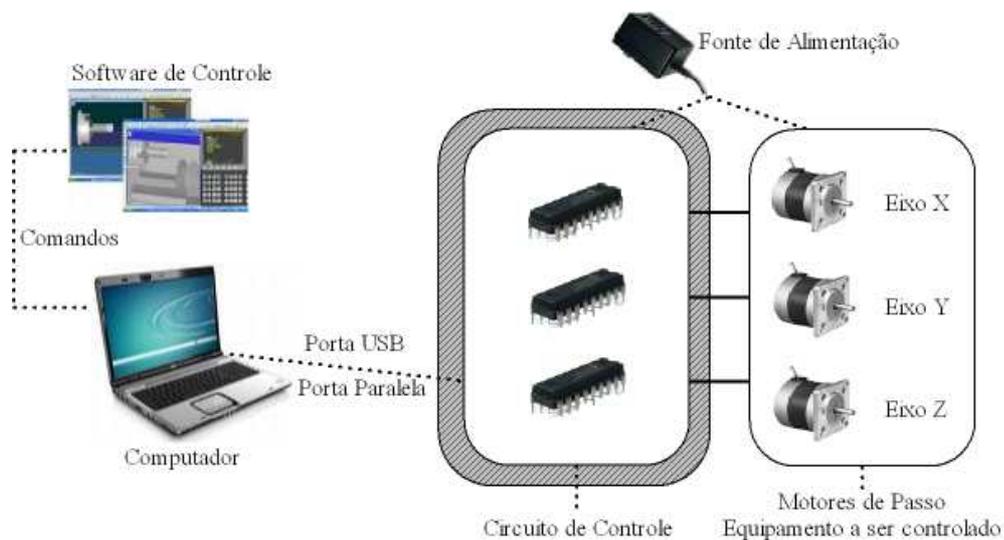


Figura 18: Estrutura típica de controle de uma máquina automática programável que usa motores de passo.

O modelo de controle apresentado contempla características importantes que compõem os sistemas de controle, possibilitando o controle de um equipamento via *software*. Desta forma, o *hardware* de controle deve estar harmônico as características enfocadas no modelo. Neste trabalho foi desenvolvido um *hardware* de controle adequado a metodologia, o qual é composto de um circuito elétrico que permite a comunicação entre o computador e o equipamento por meio da porta de comunicação USB (LOGASHANMUGAM e SURESH, 2007).

O circuito de controle é baseado em um módulo *USB-to-FIFO* de interface paralela (DLP-USB245M-G), dispendo de um barramento de dados bidirecional de 8 bits e sinais de controle (tabela 9), que são usados para controlar o fluxo de dados entre o PC e qualquer circuito externo utilizando o protocolo de comunicação USB.

Tabela 9: Sinais de controle do Chip DLP-USB245M-G.

| <b>Sinal Controle</b> | <b>Descrição</b>   |
|-----------------------|--|
| WR                    | Quando ocorre a mudança de nível lógico alto para baixo, WR lê 8 linhas de dados e escreve este <i>byte</i> em um <i>buffer</i> de transmissão FIFO ( <i>First In First Out</i> ). O dado escrito no <i>buffer</i> é enviado para o Host PC dentro do <i>timeout</i> limite TX (padrão 16ms) e colocados no <i>buffer</i> RS-232 aberto pelo programa de controle. Este <i>timeout</i> pode ser reprogramado entre 1 e 255ms dependendo da necessidade da aplicação. |
| RD                    | Quando em nível lógico baixo, RD prepara as 8 linhas de dados do corrente <i>byte</i> do <i>buffer</i> de recepção FIFO. Retornando a nível lógico alto RD prepara o próximo <i>byte</i> (se disponível) para ser lido.  |
| RXF                   | Quando em nível lógico baixo, há pelo menos um <i>byte</i> no <i>buffer</i> FIFO de recepção (de 128 bytes), pronto para ser lido com RD. RXF vai estar em nível lógico alto quando o <i>buffer</i> estiver vazio.   |
| TXE                   | Quando em nível lógico alto o <i>buffer</i> FIFO de transmissão (de 385 bytes) está cheio ou ocupado gravando o último <i>byte</i> a ser escrito. Neste caso não tentar gravar dados no <i>buffer</i> de transmissão.  |

É com esta visão, que utilizando o circuito de controle (figura 19) apresentado por LOGASHANMUGAM e SURESH (2007), é desenvolvido um *hardware* que possibilita a comunicação entre um computador pessoal e uma máquina automática programável.

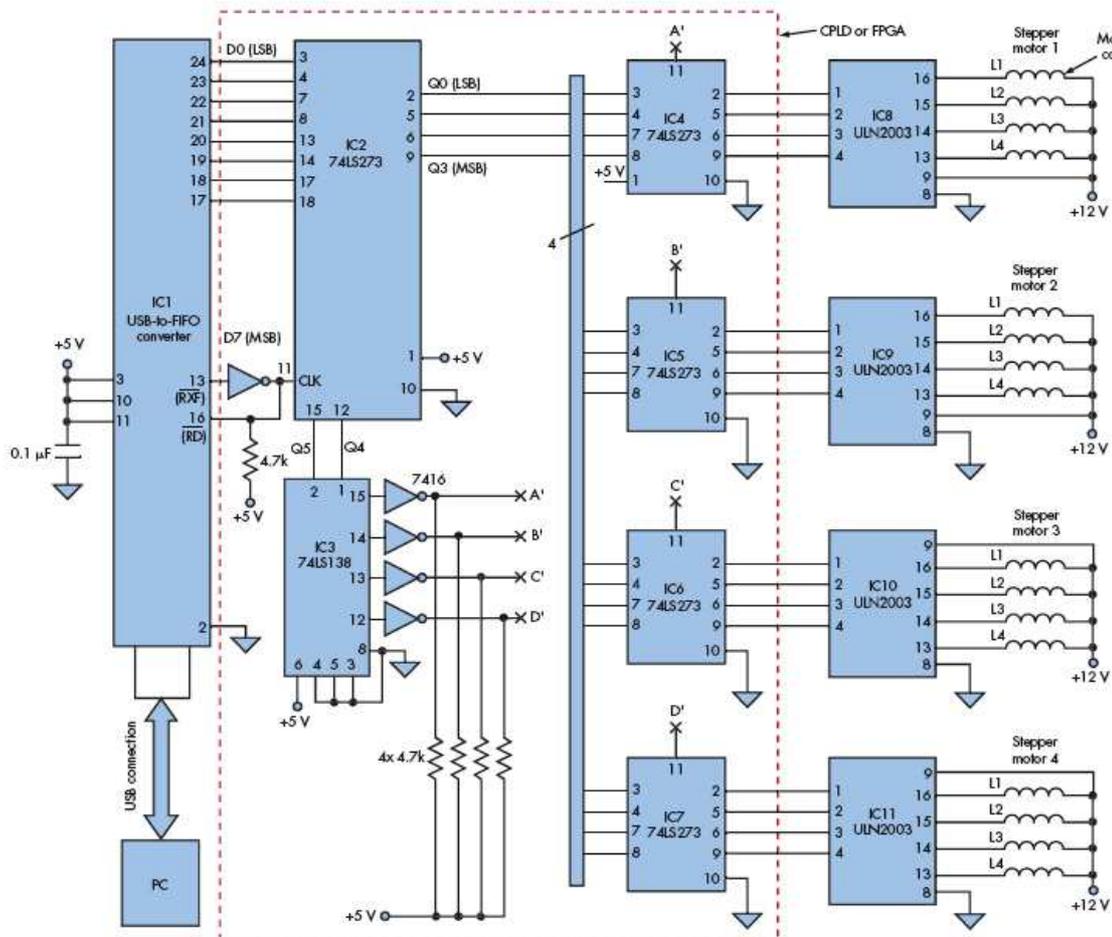


Figura 19: Circuito de Controle aplicado (LOGASHANMUGAM e SURESH, 2007).

Durante os testes do circuito de controle foi detectado que do modo em que o circuito elétrico está constituído, é necessário o envio de um *pulso nulo* (pulso que seleciona um motor diferente do atual, mas sem ativar bobinas) entre os pulsos de uma seqüência no mesmo eixo/motor (ex: pulsos “AAAAA”). Esse *pulso nulo* é enviado para poder gerar uma transição de *clock* (sinal que habilita o envio de pulsos) entre comandos enviados no mesmo eixo, sendo constatado, nos testes de controle realizados no NAFA, que não afeta o tempo de processamento e o desempenho do sistema. Quando pulsos são enviados alternadamente em

diferentes eixos (ex: pulsos “ABABABCACA”), este *pulso nulo* não é necessário, pois a simples troca de eixo gera o sinal de controle necessário.

O hardware de controle foi desenvolvido e testado no NAFA na UFSM, o qual possui a seguinte composição:

- Uma placa/matriz de contato usada para montagem do circuito elétrico;
- Um Chip da FTDI modelo DLP-USB245M-G;
- Cinco CIs 74LS273 (registrador de 8 bits, *octal D flip-flops*);
- Um CI 74LS138 (demultiplexador);
- Quatro CIs ULN2003 (driver de potência);
- Um inversor CI 7416;
- Cinco resistores de 4.7 k;
- Três capacitores de 0,47 uF;
- Uma fonte externa de 5 V;
- Uma fonte externa de 12 V.

Para estabelecer a *interface* de comunicação entre o *hardware* e o *software*, é utilizado o chip da FTDI (*Future Technology Devices International*) modelo DLP-USB245M-G (figura 20), que é um módulo de comunicação USB que possui em seu *firmware* (programa controlador do chip) toda a camada do protocolo USB. São fornecidos pelo fabricante todos os *drivers* para a instalação no computador, tendo sua biblioteca de funções e documentação detalhada para desenvolvimento na linguagem de programação C++ *Builder*® ou *Visual C++*®.

As principais características do chip DLP-USB245M-G são:

- Velocidade de transmissão e recebimento de dados de 1 MBps;
- Protocolo de comunicação USB gerenciado pelo próprio chip;
- Compatível com padrão USB 1.1 e 2.0;
- 384 bytes reservados em *buffer* FIFO para transmissão;
- 128 bytes reservados em *buffer* FIFO para recepção;
- Montagem em soquete de 24 pinos, o que facilita a conexão em circuitos;
- Modo de transferência *Bulk* (em grandes quantidades e simultaneamente) e Isossíncrona (quantidade pré-negociável da banda de transmissão do barramento);

- *Drivers* de comunicação compatíveis com Windows 98/2000/CE/XP, MAC OS-8 e OS-9 e Linux 2.40 ou superior.

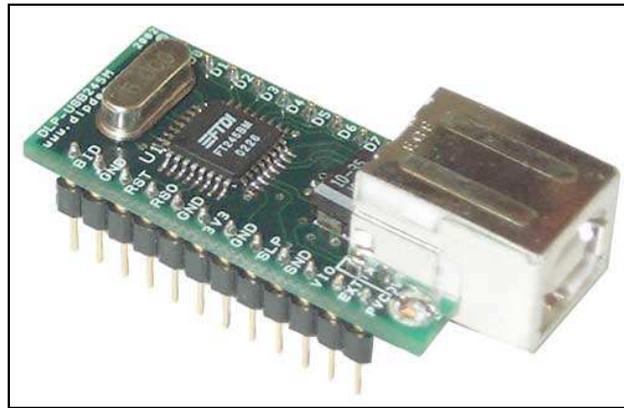


Figura 20: Chip FTDI DLP-USB245M-G

A biblioteca de funções (*D2XX Programmer's Guide*) deste chip apresenta um guia completo das suas funcionalidades de controle. Abaixo são listadas suas principais funções, sendo mais detalhadas na seção seguinte, que trata da programação do software desenvolvido.

*FT\_ListDevices* – Retorna informações dos dispositivos conectados, como número de dispositivos, descrição, identificador de conexão, entre outros.

*FT\_Open* – Requisição para deixar o dispositivo pronto para leitura ou escrita.

*FT\_Read* – Ler dados no dispositivo.

*FT\_Write* – Escrever dados no dispositivo.

*FT\_Close* – Fecha o dispositivo ao final das operações.

*FT\_SetTimeouts* – Define um *timeout* (tempo para resposta) para leitura e escrita no dispositivo.

Um dispositivo *USB-FIFO*, como o DLP-USB245M-G, já foi usado como meio de comunicação em alguns trabalhos atuais (NAGEB et al., 2005), (GIASSI JUNIOR, 2006), pois agrega um meio simples e de baixo custo para permitir a comunicação dos computadores atuais com o hardware de controle, possibilitando altas taxas de transmissão de dados, como 8 milhões de *bits* por segundo. O chip foi interligado a placa de circuito integrado montada, por meio de um soquete de 24 pinos e um cabo USB 2.0 de uso normal em impressoras.

A figura 21 mostra a placa com o circuito elétrico montado e usada nos testes.

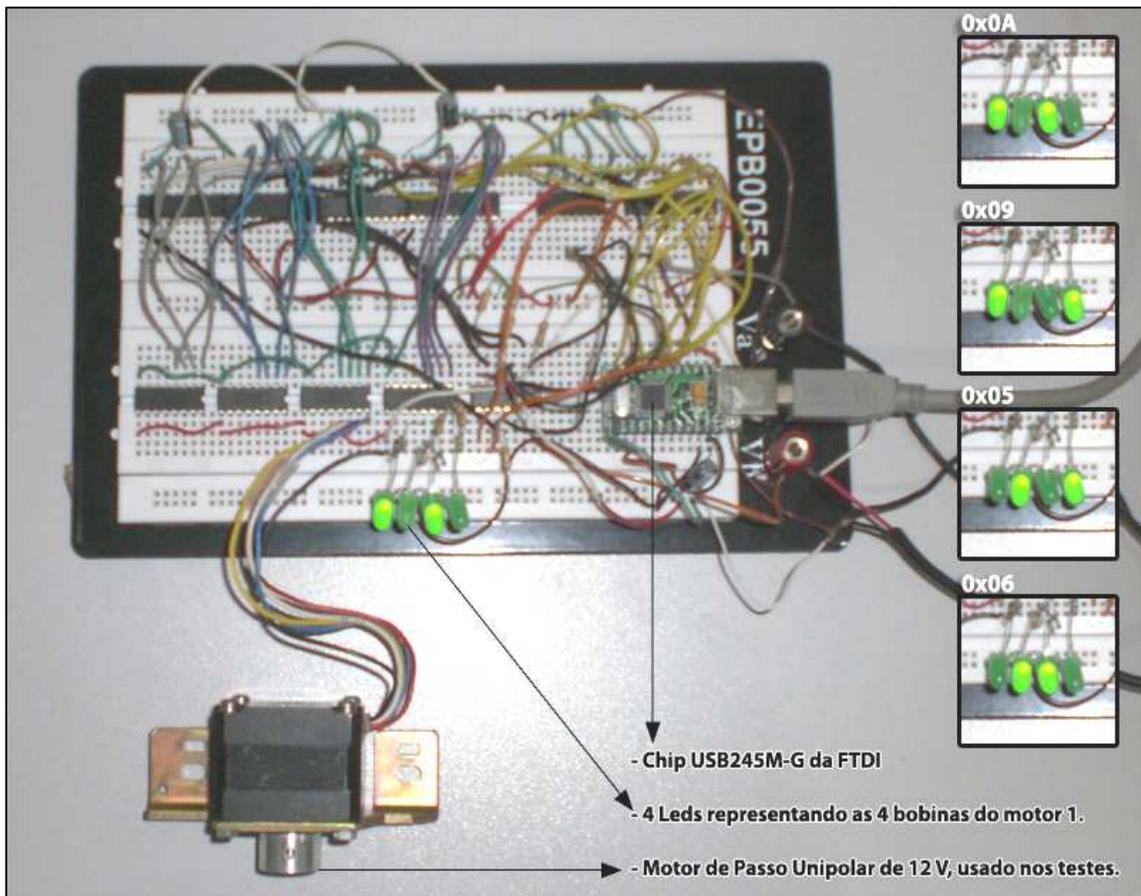


Figura 21: Circuito desenvolvido e usado nos testes.

Por meio da comunicação estabelecida entre o computador e o hardware desenvolvido, cada *byte* de dados (que representam determinados pulsos nos motores) enviado pelo software de controle, por meio da porta USB, forma os sinais da seqüência de ativação das bobinas do motor de passo, fazendo-o se movimentar da maneira desejada.

A cada *clock* (sinal) do circuito de controle, uma das quatro saídas do decodificador libera um sinal de 5VDC, sendo o sentido determinado pela seqüência de ativação das bobinas (seqüência definida no *software* de controle). Cada sinal liberado tem a função de habilitar a passagem de corrente para alimentar uma das bobinas do motor de passo, e esta variação no acionamento das bobinas provoca a rotação do motor.

Para facilitar a aplicação e desenvolvimento do *hardware* de controle, foi desenhado o projeto de fabricação de uma placa com a lógica do circuito. A figura 22 mostra o projeto desta placa, ilustrando a interligação dos componentes, de acordo com a figura 19.

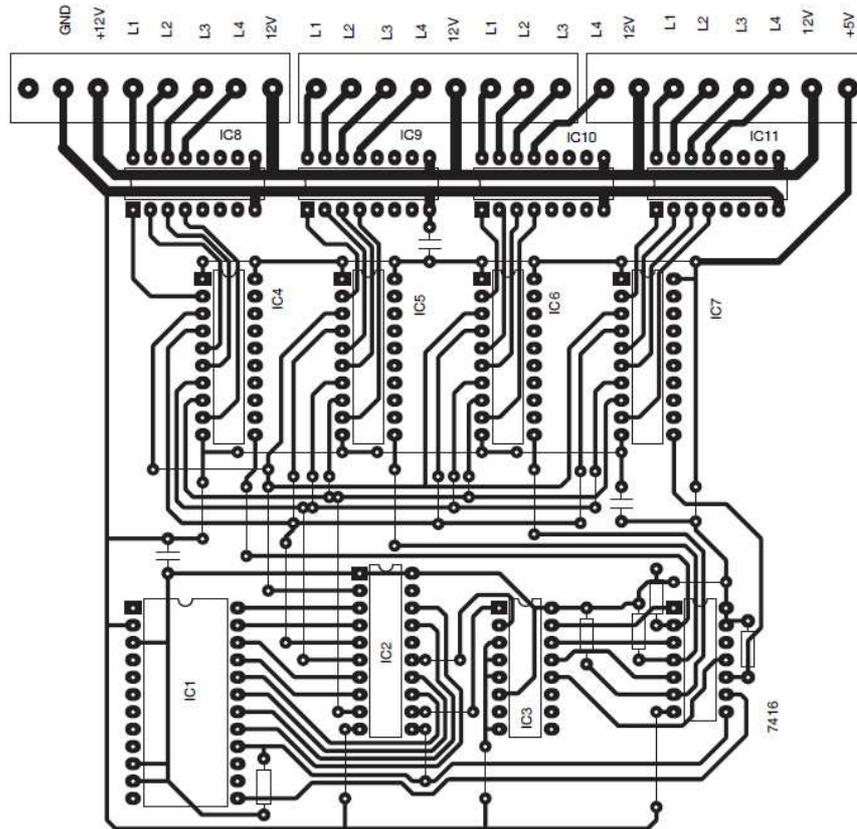


Figura 22: Projeto para fabricação da placa com o circuito de controle.

## 4.2 Software desenvolvido

O *software* para controle de equipamentos automáticos programáveis por meio da porta USB, desenvolvido neste trabalho, foi chamado de “SCEAP”. Este *software* foi desenvolvido focando o projeto, controle e testes de equipamentos diversos que fazem o uso de motores de passo como atuadores, como mesas XY, XYZ, pequenos robôs, e outros equipamentos usados em diversas pesquisas.

O *software* desenvolvido para testes da metodologia compreende algumas etapas que foram seguidas procurando torná-lo simples e funcional facilitando o seu uso. Estas etapas são:

- Escolha da linguagem de programação;
- Estruturação da interface gráfica;
- Programação do ambiente gráfico de simulação;
- Programação da comunicação por meio da porta USB; e
- Testes com o *hardware* desenvolvido.

A base para seu desenvolvimento segue as características listadas no capítulo 3 na seção 3.1 (flexibilidade e adaptabilidade), bem como a metodologia descrita na seção 3.2. A linguagem de programação usada foi o *C++ Builder*®, por ser uma linguagem que permite a programação orientada a objetos, com amplo uso no meio acadêmico e a boa documentação do *hardware* de controle (chip DLP-USB245M-G) nesta linguagem.

O ponto principal e considerado como um diferencial em relação a muitos métodos e sistemas de controle atuais é a comunicação entre computador e equipamento por meio do protocolo de comunicação USB. Comumente se encontram *softwares* e diversos trabalhos, todos usando a porta de comunicação serial ou paralela, efetivando sua comunicação nos testes, mas limitando muitas vezes a comunicação com outros equipamentos, diminuindo a continuidade destes trabalhos por outros pesquisadores (nível de particularidade muito grande).

A possibilidade de parametrização de muitas variáveis que compõem a planta de atuação do equipamento é muito importante, pois permite que pesquisadores testem vários equipamentos em sua fase inicial, evitando o gasto dispendioso de tempo em desenvolvimento de software específico de controle nas fases de projeto, contribuindo para obtenção de resultados em menor tempo. A seguir são apresentadas as telas do sistema desenvolvido, com as suas particularidades de programação e controle.

A figura 23 mostra a tela inicial, onde é dada a entrada de um arquivo contendo a codificação da movimentação ou trajetória do equipamento. Usando a opção “Código de Pulsos - Abrir”, o sistema faz a leitura e interpretação do código de pulsos, armazenando esta informação para uso na simulação e na movimentação efetiva dos eixos no equipamento. Em seguida, o usuário pode escolher qual o tipo de simulação realizar; neste ponto o sistema possui duas opções, a simulação de equipamentos que usam sistemas de coordenadas XY ou

equipamentos diversos com até três eixos de atuação, onde a movimentação dos motores será visualizada em uma linha de tempo.

Resumo das opções da tela apresentada na figura 23:

- Menu Arquivo – Opção de acesso a leitura do arquivo contendo o código de pulsos.
- Menu Configurações – Parametrização do equipamento (intervalo entre pulsos, resolução dos motores, folga backlash, aceleração e desaceleração, ambiente de atuação).
- Menu Ajuda – Descrição Geral do Funcionamento do Sistema.
- Abrir Código de Pulsos – Arquivos contendo os códigos de pulsos que representam a trajetória que o equipamento irá realizar.
- Efetuar Simulações e Processar comandos – Esta opção possibilita simular e executar o controle de equipamentos que usam sistemas de coordenadas com dois eixos XY. Outra opção de simulação também mostra em uma linha de tempo a movimentação de um equipamento com três eixos.

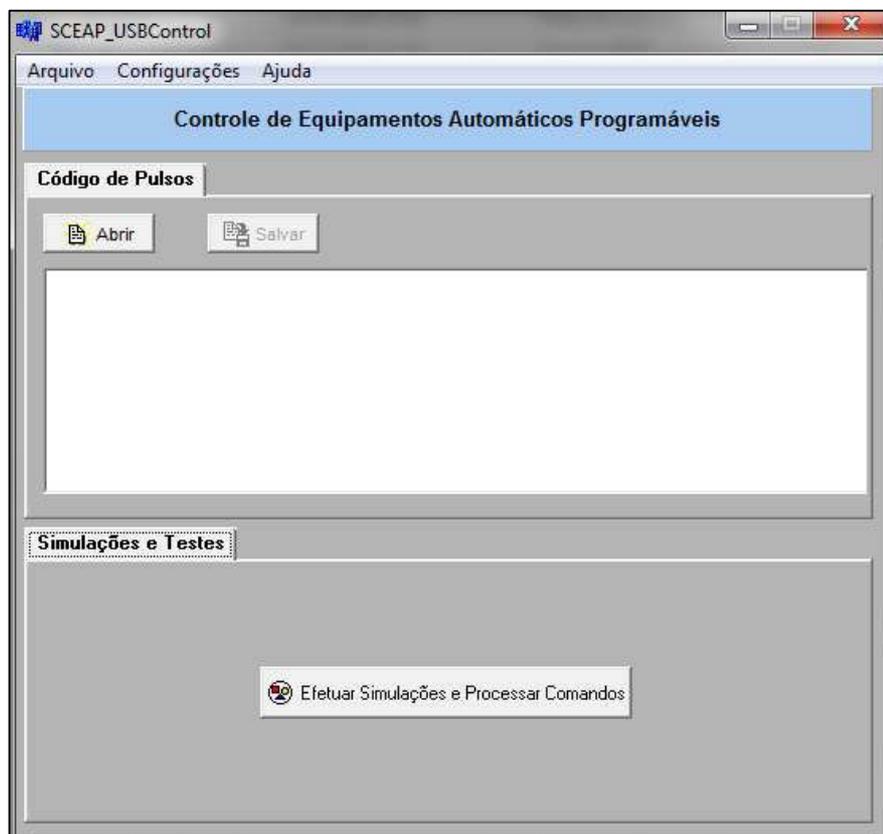


Figura 23: Tela inicial do software desenvolvido.

As figuras 24, 25 e 26, mostram os acessos pelos menus da tela inicial, que representam a leitura de código de pulsos, configurações do equipamento e informações sobre o uso do sistema (opção ajuda), respectivamente.

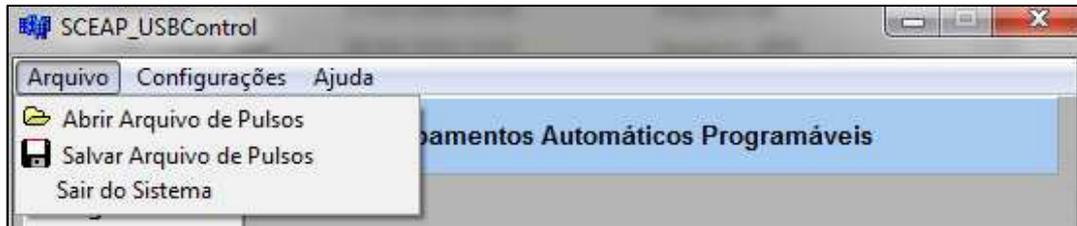


Figura 24: Menu Arquivo – ênfase na abertura do código com a trajetória do equipamento.

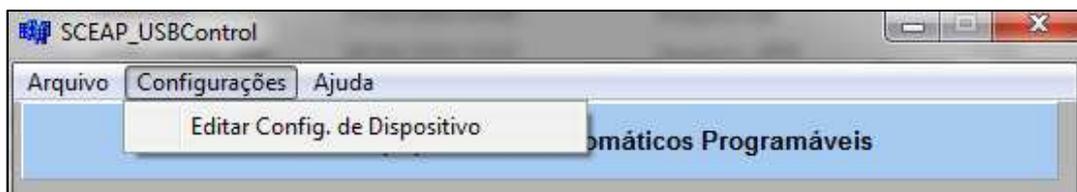


Figura 25: Menu Configurações – ênfase na parametrização do equipamento.

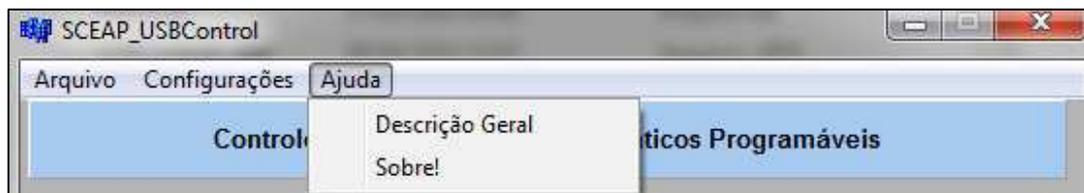


Figura 26: Menu Ajuda – descrição geral das funcionalidades do sistema.

Após a leitura do arquivo contendo o código de pulsos, a pré-visualização e simulação da movimentação do equipamento podem ser realizadas. A simulação gráfica foi programada usando a linguagem **OpenGL**, que é uma biblioteca de rotinas gráficas que permite a programação de diversas formas geométricas em 2D e 3D, sendo muito utilizada em jogos e sistemas de visualização gráfica. A **OpenGL** foi usada por ser facilmente integrada ao ambiente de programação C++ *Builder*®, onde permitiu a programação e simulação gráfica da trajetória do equipamento controlado. De acordo com Cohen e Manssour, 2006, pode-se definir OpenGL como uma especificação aberta e multiplataforma de uma biblioteca de

rotinas gráficas e de modelagem utilizada para o desenvolvimento de aplicações de computação gráfica.

A programação realizada no sistema permite a visualização e correção pró-ativa do equipamento que está utilizando o sistema de controle. A figura 27 mostra a tela com a simulação gráfica de uma trajetória exemplificada em um sistema com dois eixos, podendo se observar a esquerda do eixo Y do plano cartesiano, as informações da simulação que está em andamento.

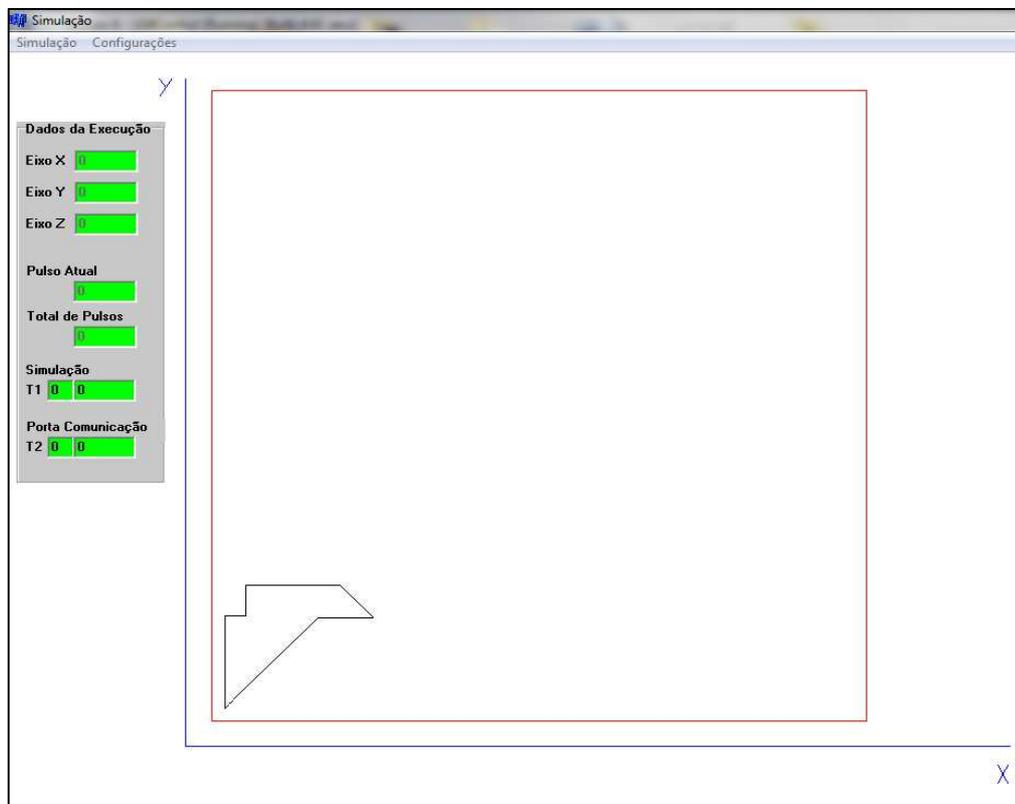


Figura 27: Tela que faz a simulação de uma trajetória especificada.

Dados atualizados durante a simulação:

- Posição atual do Eixo X.
- Posição atual do Eixo Y.
- Pulso Atual – de acordo com o arquivo de codificação da trajetória.
- Total de Pulsos que serão executados.
- T1 – Tempo de simulação.
- T2 – Tempo de envio dos comandos para o equipamento.

Os tempos de execução (*timers*) T1 e T2 são equivalentes e proporcionais a quantidade de pulsos que serão executados, permitindo executar a simulação e o envio dos comandos para a porta USB ao mesmo tempo, para assim, visualizar em tempo de execução a movimentação no equipamento.

Na figura 28 é apresentada a tela que mostra outra opção de simulação, onde a execução do controle de três eixos/motores de um equipamento é visualizada em uma linha de tempo. Esta opção pode ser usada em aplicações diversas, onde o monitoramento do envio dos comandos para os motores pode ser visualizado passo a passo. Pode-se citar como exemplo do uso deste tipo de simulação, o projeto de um equipamento para reabilitação passiva contínua em cotovelo (para reabilitação no pós-operatório), onde foram utilizados dois motores de passos realizando a movimentação de duas articulações (CALLEGARO, 2010). Este tipo de equipamento não usa sistema de coordenadas, assim, pode-se testar e visualizar em uma linha do tempo os pulsos enviados aos motores.

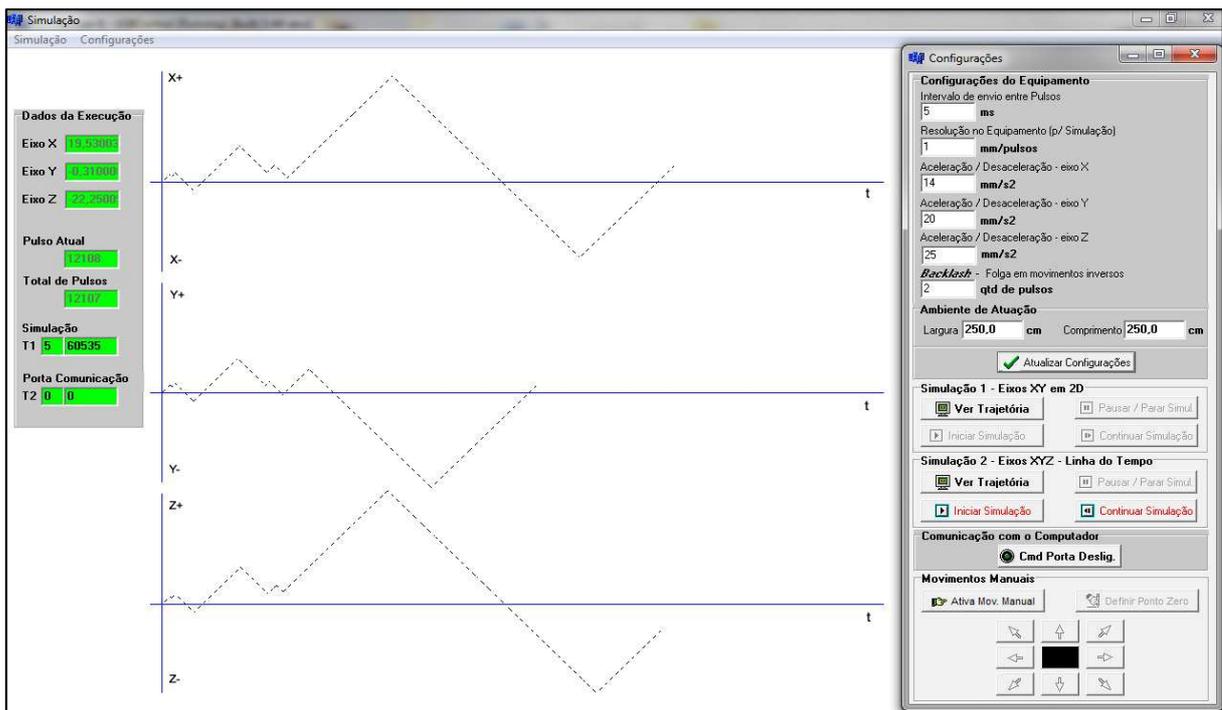


Figura 28: Tela do sistema que mostra a execução de comandos em três eixos/motores do equipamento.

Quando ajustes na trajetória forem necessários, algumas configurações podem ser alteradas, podendo-se simular várias vezes o projeto antes de enviar o comando para a porta USB. Estas configurações são descritas a seguir:

- **Intervalo de envio entre pulsos:** É o intervalo de tempo entre o envio de pulsos/comandos para a porta USB, o qual é informado em milissegundos. Este intervalo deve ser ajustado de acordo com as características de funcionamento do equipamento, onde o incremento e decréscimo deste parâmetro representam a velocidade dos motores.

- **Resolução no equipamento:** representa a distância que o equipamento percorre a cada pulso enviado, definido em milímetros por pulso (mm/pulso). Esta configuração é importante para a representação gráfica, ao simular a movimentação do equipamento, auxiliando assim na detecção de erros.

- **Distância para aceleração:** esta informação é usada para o cálculo da aceleração dos motores, onde o usuário pode definir de acordo com características físicas do equipamento, qual a distância que o motor percorrerá até chegar à velocidade constante (intervalo de envio de pulsos).

- **Aceleração inicial e desaceleração:** configuração por eixo controlado no sistema, onde é informada a aceleração inicial do equipamento. São configurações importantes e usadas de acordo com o tamanho do equipamento e características físicas dos motores, sendo informada pelo projetista, de acordo com variantes como inércia da carga e tamanho/limites da planta de atuação. De acordo com Souza (2008), a relação do tempo entre pulsos, em função da velocidade gerada no motor de passo, apresenta duas regiões distintas. A **região de arranque e parada**, onde o motor pode ser acionado ou parado com o envio de pulsos a uma frequência constante, podendo acelerar a sua própria inércia sem que haja perda de sincronismo. Na **região de alta velocidade** é impossível parar ou inverter o sentido de rotação do motor sem perder sincronismo no sistema de controle. A figura 29 mostra a curva destas duas regiões.

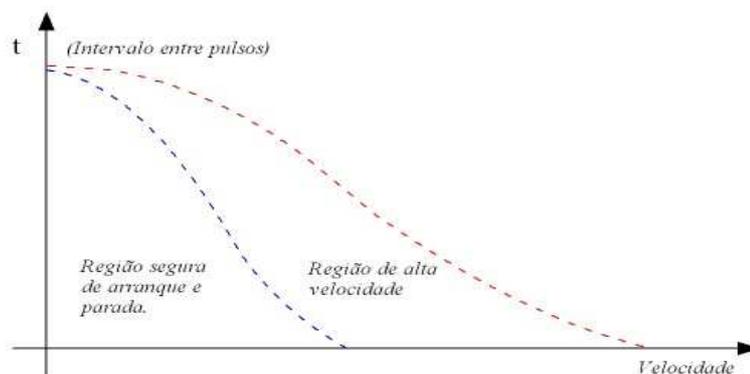


Figura 29: Curva da relação do tempo entre pulsos e a velocidade do motor.

O motor de passo deve iniciar o seu deslocamento na região de arranque/parada e ter seu intervalo de pulsos aumentado **progressivamente** até a região de alta velocidade. Por outro lado, quando for preciso parar ou inverter o sentido de rotação do motor, deve-se modificar a velocidade do motor para a região de arranque/parada, evitando a perda de sincronismo. Sendo assim, pode-se configurar a aceleração do motor, que resultará em um determinado tempo que o motor levará para chegar a sua velocidade de trabalho. Cabe ao projetista definir estas regiões de acordo com os motores usados no equipamento e configuração no sistema de controle. No *software* desenvolvido, a aceleração e desaceleração são definidas em função do incremento e decremento de uma constante de tempo (intervalo entre pulsos), tempo este, informado para cada eixo na tela de configurações. O resultado é um tempo decrementado para acelerar, ou incrementado para desacelerar, em um intervalo fixo até chegar ao intervalo constante de envio de pulsos informado no sistema. A figura 30 descreve, em um exemplo, como o sistema regula a aceleração dos motores.

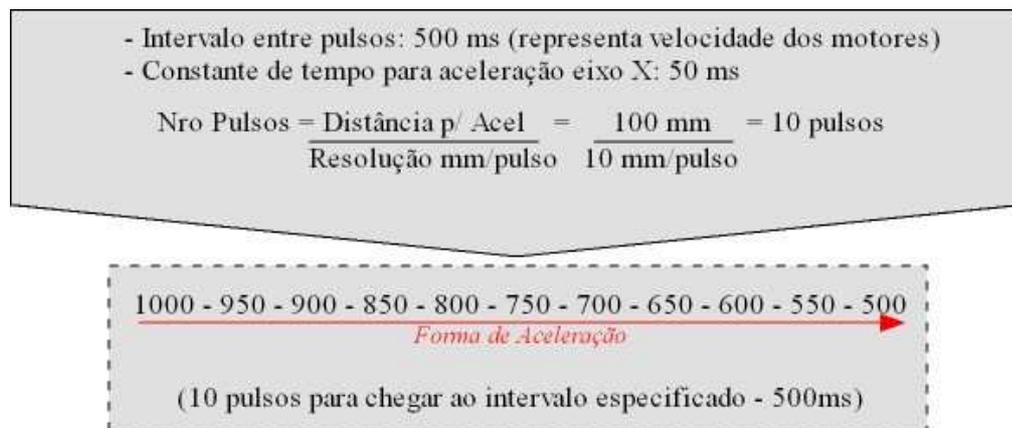


Figura 30: Exemplo do cálculo usado para regular a aceleração do motor.

- **Folga *backlash***: configurar um número de pulsos para compensar a folga do equipamento em movimentos inversos;

- **Limites de atuação**: tamanho do ambiente ou planta de atuação, definida em centímetros. Importante para a representação gráfica durante a simulação da trajetória, verificando assim possíveis erros de movimentação.

- **Movimentos manuais**: realização de movimentos com o envio passo a passo (ou pulso a pulso) na porta de comunicação USB, de acordo com os eixos X e Y da primeira opção de simulação.

A seguir a figura 31 apresenta a tela do sistema que permite as configurações anteriormente descritas.



Figura 31: Tela para configuração e ajustes do equipamento.

Como já especificado na seção 4.1, a programação entre computador e equipamento foi realizada usando o protocolo USB por meio do chip USB245BM da FTDI. O custo atual deste chip é de \$ 25 dólares e possui *drivers* de controle para os sistemas operacionais atuais (*Windows*, *Linux* e *Mac*). Após o dispositivo estar montado e alimentado no circuito elétrico, ao conectar o cabo USB no computador, o dispositivo é reconhecido e solicita a identificação dos *drivers*, neste caso a *DLL* (*Dynamic Link Library* – Biblioteca de ligação dinâmica) “*ftd2xx.dll*” usada no sistema operacional *Windows* é solicitada para concluir a instalação.

A seguir são descritas as principais funções do chip DLP-USB245M-G que foram usadas na programação do *software* de controle desenvolvido, detalhando como as mesmas são processadas e como ocorre o envio dos comandos para a porta de comunicação USB. Maiores detalhes e também outros exemplos de uso das funções podem ser encontrados na biblioteca de funções do *chip*.

#### 4.2.1 Funções *FT\_ListDevices* e *FT\_Open*

Esta função retorna informações dos dispositivos conectados, como por exemplo, o número de dispositivos conectados, descrição do componente e identificador usado na conexão. Abaixo segue o detalhamento de sua estrutura.

*FT\_ListDevices* (*PVOID pvArg1*, *PVOID pvArg2*, *DWORD dwFlags*)

A figura 32 ilustra o exemplo de uso desta função dentro do software desenvolvido, onde a variável *dwFlags* faz referência a alguns *bits*, que por sua vez, indicam as informações dos dispositivos. No exemplo abaixo, se o bit *FT\_LIST\_NUMBER\_ONLY* estiver em nível lógico 1, em *dwFlags*, o primeiro argumento (*numDevs*) retornará o número de dispositivos conectados. O mesmo modo de referência será usado para outras variáveis/*bits* que representam outras informações do dispositivo.

```
ftStatus = FT_ListDevices(&numDevs, NULL, FT_LIST_NUMBER_ONLY);
if (ftStatus == FT_OK)
    ; // FT_ListDevices OK, number of devices connected is in numDevs
else
    ShowMessage("FT_ListDevice Error");
    //Erro na tentativa de listar os dispositivos;
```

Figura 32: Função para listagem de informações do dispositivo.

A função *FT\_Open* é usada para abertura de um ou mais dispositivos, deixando-o pronto para os acessos seguintes. Abaixo segue a estrutura da função *FT\_Open*.

***FT\_Open*** (*int iDevice*, *FT\_HANDLE \*ftHandle*)

Esta função possui apenas dois parâmetros, o primeiro indica o número do dispositivo e o segundo é a variável usada para acessar o dispositivo nas operações seguintes. Se apenas um dispositivo estiver conectado o primeiro parâmetro (*iDevice*) deve ser 0, quando se estiver usando múltiplos dispositivos, estes serão referenciados por números inteiros, como 1, 2 e etc. A figura 33 mostra como foi usada a função *FT\_Open* neste trabalho.

```
ftStatus = FT_Open(0, &ftHandle);
if (ftStatus == FT_OK)
    ; // FT_Open OK, use ftHandle to access device
else
    ShowMessage("FT_OpenDevice Error"); // FT_Open failed
```

Figura 33. Usando a função *FT\_Open*.

#### 4.2.2 Função *FT\_Write*

As operações de escrita no dispositivo que resultarão em movimentos nos motores de passo, são processadas pela função *FT\_Write*, a qual tem sua estrutura detalhada e definida de acordo com quatro parâmetros importantes.

***FT\_Write*** (*FT\_HANDLE ftHandle*, *LPVOID lpBuffer*, *DWORD dwBytesToWrite*, *LPDWORD lpdwBytesWritten*)

- *ftHandle*: identificador do dispositivo em que será efetuada a operação de escrita. O mesmo identificador usado na operação abertura na função anterior.

- *lpBuffer*: apontador para o *buffer* (variável que armazena em memória os dados que serão usados na operação) que contém os dados que serão escritos no dispositivo.

- *dwBytesToWrite*: número de bytes que serão escritos no dispositivo.

- *lpdwBytesWritten*: apontador para a variável que retorna o número de bytes escrito no dispositivo.

Os parâmetros acima são definidos de acordo com as características do hardware a controlar. Neste trabalho o circuito de controle apresentado permite o controle de até 4 motores. Para a movimentação correta dos motores os dados devem ser enviados para a porta USB seguindo uma tabela de dados de controle (figura 34), onde a seqüência de envio define a direção de rotação do motor, e as configurações de velocidade, aceleração e desaceleração são especificados no software de controle.

| STEPPER-MOTOR-CONTROL BYTE VALUES |                        |                 |    |    |    |           |         |
|-----------------------------------|------------------------|-----------------|----|----|----|-----------|---------|
| Data byte to USB-to-FIFO          | Stepper motor selected | Coils energized |    |    |    | Direction |         |
|                                   |                        | L1              | L2 | L3 | L4 | Forward   | Reverse |
| 0x0A                              | SM1                    | 1               | 0  | 1  | 0  | ↓         | ↑       |
| 0x09                              | SM1                    | 1               | 0  | 0  | 1  | ↓         | ↑       |
| 0x05                              | SM1                    | 0               | 1  | 0  | 1  | ↓         | ↑       |
| 0x06                              | SM1                    | 0               | 1  | 1  | 0  | ↓         | ↑       |
| 0x1A                              | SM2                    | 1               | 0  | 1  | 0  | ↓         | ↑       |
| 0x19                              | SM2                    | 1               | 0  | 0  | 1  | ↓         | ↑       |
| 0x15                              | SM2                    | 0               | 1  | 0  | 1  | ↓         | ↑       |
| 0x16                              | SM2                    | 0               | 1  | 1  | 0  | ↓         | ↑       |
| 0x2A                              | SM3                    | 1               | 0  | 1  | 0  | ↓         | ↑       |
| 0x29                              | SM3                    | 1               | 0  | 0  | 1  | ↓         | ↑       |
| 0x25                              | SM3                    | 0               | 1  | 0  | 1  | ↓         | ↑       |
| 0x26                              | SM3                    | 0               | 1  | 1  | 0  | ↓         | ↑       |
| 0x3A                              | SM4                    | 1               | 0  | 1  | 0  | ↓         | ↑       |
| 0x39                              | SM4                    | 1               | 0  | 0  | 1  | ↓         | ↑       |
| 0x35                              | SM4                    | 0               | 1  | 0  | 1  | ↓         | ↑       |
| 0x36                              | SM4                    | 0               | 1  | 1  | 0  | ↓         | ↑       |

Figura 34. Dados de controle dos motores (LOGASHANMUGAM e SURESH, 2007).

A tabela contém os valores em notação hexadecimal que são usados para energizar as bobinas de um determinado motor, sendo que a seqüência de ativação destas bobinas fará com que o motor gire em um sentido. Cada *byte* de dados enviado para o dispositivo DLP-USB245M-G contém em seus 4 bits mais significativos a escolha de qual motor será acionado, e os 4 bits menos significativos representam as bobinas que serão energizadas. A figura 35 ilustra com um exemplo, como este controle ocorre.

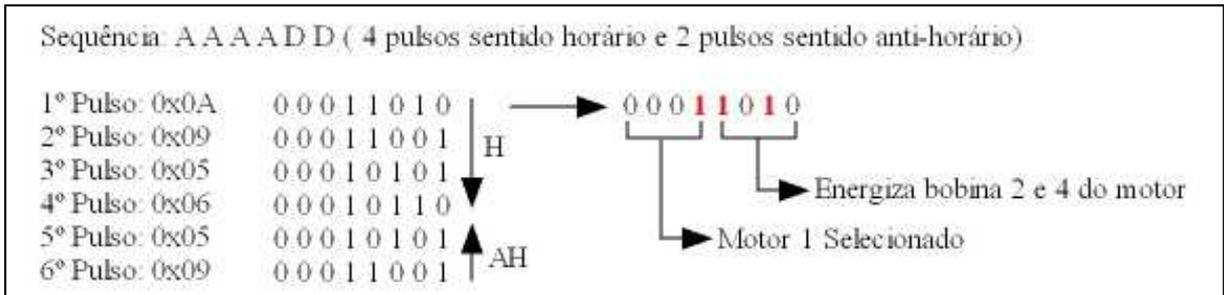


Figura 35. Envio de pulsos com a composição do *byte* enviado para a porta USB.

Na figura 35 é visualizado uma sequência de 6 pulsos (A A A A D D), os quais são representados por códigos hexadecimais, que no circuito de controle são convertidos em códigos de 8 bits, para assim representar o motor selecionado e as bobinas ativadas.

O arquivo de entrada contendo o código de pulsos é resultado de técnicas de interpolação linear e circular. Desta forma, o *software* faz a leitura dos códigos de envio para movimentar o equipamento em determinada trajetória. Na figura 36 é visualizada uma sequência que exemplifica o processamento de um arquivo de pulsos contendo a técnica de interpolação circular (WECK, 1989).

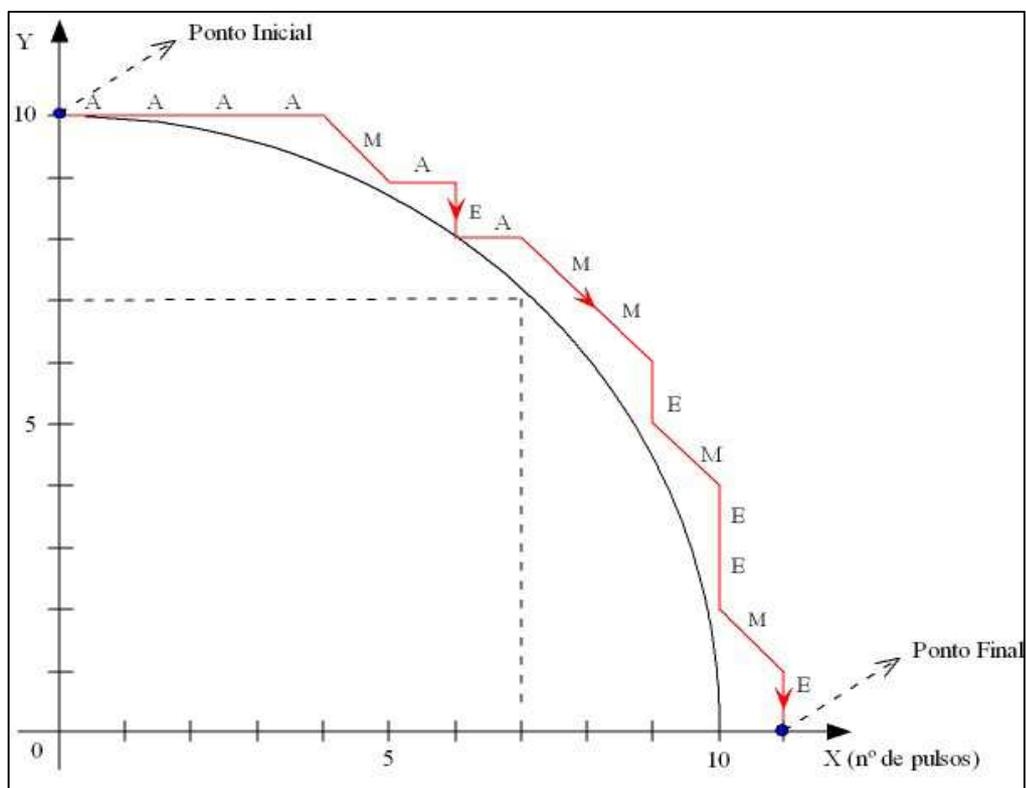


Figura 36. Sequência de pulsos exemplificada na técnica de interpolação circular (Adaptado de WECK, 1989).

A figura 37 mostra como foi usada a função *FT\_Write*, a qual contém o array *mData1* com os dados para movimentação do motor de passo e a variável *codData1* que controla em qual ponto da sequência de pulsos o array *mData1* se encontra. Deste modo é possível controlar se o próximo movimento do equipamento será em sentido horário ou anti-horário, quando a sequência de pulsos lida no arquivo representa movimentos dos motores em várias direções.

```

mData1[0] = 0x0A; // 0A, 09, 05, 06 - Forward
mData1[1] = 0x09; // 06, 05, 09, 0A - Reverse
mData1[2] = 0x05;
mData1[3] = 0x06;

ftStatus = FT_Write(ftHandle, &mData1[codData1], 1, &BytesWritten);
if (FT_OK == 0){
    // enviou dado
    //ShowMessage("Dado Escrito: "+BytesWritten);
    if(codData1==3) codData1=0; else codData1++;
}
else
    ShowMessage("FT_Write Error"); // erro no envio

```

Figura 37. Usando a função *FT\_Write*.

#### 4.2.3 Funções *FT\_Close*, *FT\_SetTimeouts* e *FT\_Read*

Após uma sequência de operações a função *FT\_Close* é usada para fechar a comunicação com o dispositivo. A sua descrição é simples, pois é composta apenas de um parâmetro identificador do dispositivo aberto anteriormente, a variável *ftHandle*.

***FT\_Close*** (*FT\_HANDLE ftHandle*)

Exemplo:

```
FT_Close(ftHandle);
```

Para saber se o envio de um comando de leitura ou escrita foi executado com sucesso, é usada a função ***FT\_SetTimeout***. Esta função é muito importante para a detecção de erros no processamento dos comandos, agregando uma característica fundamental para garantir o envio correto dos pulsos pelo software. A seguir é detalhada a estrutura desta função.

***FT\_SetTimeouts*** (FT\_HANDLE *ftHandle*, DWORD *dwReadTimeout*, DWORD *dwWriteTimeout*)

Define um *timeout* (tempo para resposta) nas operações de leitura e escrita no dispositivo. Novamente o parâmetro *ftHandle* define em qual dispositivo será enviado o comando, e nos próximos dois parâmetros deve-se informar o *timeout* em milissegundos.

Exemplo:

Definindo um timeout de 3 segundos para leitura e 1 segundo para escrita.

```
ftStatus = FT_SetTimeouts(ftHandle,3000,1000);
```

A função ***FT\_Read*** apesar de não usada no software desenvolvido, merece destaque pela possibilidade de controle que ela pode adicionar em desenvolvimentos futuros. Esta função habilita o sistema de controle a fazer leituras de sensores e outros dispositivos por meio da porta USB. É possível ler informações e tratá-las no software, para assim gerar novos comandos no dispositivo.

***FT\_Read*** (FT\_HANDLE *ftHandle*, LPVOID *lpBuffer*, DWORD *dwBytesToRead*, LPDWORD *lpdwBytesReturned*)

Esta função possui os mesmos parâmetros da função de escrita, permitindo ler um número de *bytes* do dispositivo e armazenar em um *buffer* de dados, que pode ser processado via software.

Todas as funções descritas são importantes na programação do software de controle, pois possibilitam o controle e parametrização de características de um equipamento. O software de controle foi testado no Núcleo de Automação e Projetos de Fabricação - NAFA da UFSM e demonstrou a viabilidade de controlar equipamentos automáticos programáveis.

## 5 CONCLUSÃO

Esta pesquisa apresentou uma metodologia de controle para equipamentos automáticos programáveis, dando ênfase a sua flexibilidade de controle, com a possibilidade de configuração e adaptação às características dos equipamentos. Estas características tornam a metodologia aplicável no projeto de novos equipamentos, diminuindo o tempo gasto no desenvolvimento e impulsionando atividades conjuntas entre diversas áreas de pesquisa.

A parametrização destas características ou funcionalidades é delimitada pela complexidade dos projetos, que é resultado da criatividade dos projetistas. Muitas vezes esta criatividade é limitada às possibilidades de controle oferecidas por *softwares* comerciais fechados (*softwares* proprietários). Desta forma, as características configuráveis destacadas e usadas neste trabalho (como por exemplo, velocidade, aceleração, resolução dos motores, folga em movimentos inversos, etc.), diversificaram as funcionalidades de controle, permitindo que vários trabalhos possam ser desenvolvidos com efetividade e menor custo.

A técnica proposta é aplicável em ambiente industrial como opção de sistema de controle, pois suas características permitem o desenvolvimento a baixo custo de equipamentos com funções dedicadas.

Considerando que para se fazer uso da técnica, é necessário que a sequência de pulsos para cada tarefa programada esteja de acordo com a codificação especificada, o sistema de controle exige processamento prévio desses dados. Porém, métodos de interpolações linear e circular implementados em trabalhos atuais desenvolvidos no NAFA/UFSM, representam um meio eficiente de conversão de movimentos desejados na codificação de pulsos usada.

Um ponto importante a destacar da metodologia, que forma um diferencial para o processo de comunicação e controle entre equipamento e computador pessoal, foi a comunicação usando o protocolo USB, realizado por meio de um dispositivo eletrônico (circuito integrado FTDI USB245BM-G) programável via *software*. A biblioteca de funções desse componente traz possibilidades de controle por esta porta de comunicação, tornando o sistema de controle portátil (usado em qualquer computador pessoal) e de acordo com os requisitos de comunicação atuais.

A metodologia de controle flexível foi testada usando o *software* de controle desenvolvido na linguagem C++ *Builder*®, e possibilitou o controle de equipamentos com três motores de passo de maneira independente, podendo ainda, de acordo com o *hardware*

especificado, controlar um quarto motor. A correlação entre os pulsos elétricos identificados na saída para os motores e os dados codificados na tabela de pulsos, demonstra a eficiência do software e do circuito de controle desenvolvidos.

Por fim, conclui-se que os sistemas de controle flexíveis e parametrizáveis permitem que os projetistas tenham mais liberdade nas operações de acionamento, eliminando o tempo gasto no desenvolvimento de *software e hardware* de controle específico, nas fases de projeto e desenvolvimento de novos equipamentos, podendo assim diminuir os custos relacionados.

Esta dissertação abre um leque de possibilidades de pesquisas na área de sistemas de controle flexíveis destinados ao desenvolvimento de equipamentos programáveis. Desta forma, algumas atividades podem ser realizadas para dar continuidade nesta pesquisa, como as descritas a seguir:

- Acrescentar ao sistema a conversão de programas CN para a codificação de pulsos especificada na metodologia, através de técnica de interpolação linear e circular;
- Desenvolver outras possibilidades de simulação da trajetória do equipamento, como a implementação da parte tridimensional de movimentação;
- Acrescentar a possibilidade de leitura de sensores no sistema, para controlar eventos, através da adaptação do hardware de controle;
- Acrescentar um código a tabela de pulsos para acionamento e desligamento de dispositivos, a partir de um determinado ponto de execução;
- Acrescentar à metodologia, a possibilidade de controlar *servomotores* (motores com alto torque e larga faixa de rotação), para acionamento de equipamentos de maior porte;
- Implementar uma rotina para efetuar a leitura reversa do arquivo de pulsos, a partir de qualquer ponto de execução.

## REFERENCIAS BIBLIOGRÁFICAS

ALMEIDA, Sérgio L. R.; **Módulo Computacional baseado em Redes Neurais para a Força de Corte e Rugosidade, em Torneamento.** Universidade Estadual de Campinas - UNICAMP, Faculdade de Engenharia Mecânica, Departamento de Engenharia de Fabricação, Tese de Doutorado, 2006.

AMARAL FILHO, Guy C. **Requisitos para Sistemas de Controle de Sistemas Produtivos Integrados à Gestão.** Universidade de São Paulo - USP, Dissertação de Mestrado, 2005.

AMS – Advanced Micro Systems INC. *Stepper Motor System Basics* (Rev. 2 / 2010). Acessado em 06/04/2010, disponível em: <http://www.ams2000.com/pdf/step101.pdf>.

Art            Soft            **MACH3.**            **Documentation.**            Disponível            em:  
<http://www.machsupport.com/documentation.php>.

ASTRÖM, Karl J. **Computer-Controlled systems: theory and design.** 3<sup>rd</sup> ed. Prentice Hall, 1997.

BRAGA, Newton C., **As Portas Paralelas do PC.** Saber Eletrônica, Saber Ltda, Ano 32, número 281, São Paulo, 1996.

BRITTO, Ricardo S. **Uma Arquitetura Distribuída de Hardware e Software para Controle de um Robô Móvel Autônomo.** Universidade Federal do Rio Grande do Norte - UFRN, Programa de Pós-Graduação em Engenharia Elétrica e de Computação, Dissertação de Mestrado, 2008.

CALLEGARO, Aline M.; **Desenvolvimento de um Equipamento Computadorizado de Movimentação Passiva Contínua para Cotovelo e Antebraço.** Universidade Federal de Santa Maria – UFSM, Engenharia de Produção – Gerência da Produção, Dissertação de Mestrado, 2010.

CALIXTO, Gustavo M.; **Desenvolvimento de um posicionador eletrônico para estruturas micrométricas.** Universidade Estadual de Campinas - UNICAMP, Faculdade de Engenharia Elétrica e Computação, Departamento de Microonda e Óptica, Dissertação de Mestrado, 2009.

CARVALHO, Renon S.; DUTRA, Jair C.; BONACORSO, Nelso G.; **Implementação de controlador CNC de baixo custo em manipulador robótico para soldagem.** Universidade Federal de Santa Catarina – UFSC – LABSOLDA, em: V Congresso Nacional de Engenharia Mecânica - CONEM, 2008.

CHELLA, Marco T. **Arquitetura para laboratório de acesso remoto com aplicações educacionais**. Universidade Estadual de Campinas – UNICAMP, Tese de Doutorado, 2006.

COHEN, Marcelo. MANSSOUR, Isabel H. **OpenGL, Uma Abordagem Prática e Objetiva**. Novatec Editora, São Paulo, 2006.

CORREA, Anderson P. **Um projeto de controle de movimentação veicular projetado em um processador embarcado em FPGA com ambiente de simulação usando instrumentação virtual**. Universidade de Brasília - UnB, Departamento de Engenharia Mecânica, Dissertação de Mestrado, 2007.

DAVID, Raphael P. **Desenvolvimento de software emulador de terminal de vídeo compatível com PC de baixo custo**. Universidade de Brasília – UnB, Departamento de Engenharia Elétrica, Monografia, 2009.

DE CRISTO, Edinara F. **Implementação de Técnicas de Controle de Motor de Passo em Aplicações CNC**. Universidade Federal de Santa Maria - UFSM, Engenharia de Produção – Gerência da Produção, Dissertação de Mestrado, 2009.

DE PAULA, Marco A. B.; SANTOS, Eduardo A. P.; **Uma abordagem metodológica para o desenvolvimento de sistemas automatizados e integrados de manufatura**. Revista Produção, v. 18, n. 1, p. 008-025, Jan./Abr. 2008.

DE SOUZA, Jeovane V.; SOARES, Alcimar B; FRANCO, Sinésio D.; **Desenvolvimento de um Sistema de Hardware e Software para Otimização das Unidades de Controle, Monitoração, Coleta e Processamento de Dados de um Macroidentador Portátil para Avaliação In-Situ de Propriedades Mecânicas de Dutos Metálicos**. Revista Horizonte Científico, Volume 1, N. 9, 2008.

DUTRA, Jair C.; FELIPPE, Luiz; CARVALHO, Renon S. **Sistema robótico de quatro graus de liberdade e processos de soldagem dedicados para o revestimento de tubos de caldeiras**. Universidade Federal de Santa Catarina - LABSOLDA, em: V Congresso de Inovação Tecnológica em Energia Elétrica – V CITENEL, Belém/PA, 2009.

ELSON, Jeremy. *PARAPIN: A Parallel Port Pin Programming Library for Linux*. Information Sciences Institute University of Southern California, USA, 2000.

**FTDI Chip**, Future Technology Devices. <http://www.ftdichip.com/>.

**EMC2 Documentation**; Disponível em: <http://www.linuxcnc.org/>.

FERNANDES JUNIOR, Francisco G. **Metodologia para Re-Sintonia de Controladores PID Industriais**. Universidade Federal do Rio Grande do Norte – UFRN, Programa de Pós-Graduação em Engenharia Elétrica, Dissertação de Mestrado, 2006.

GIASSI JUNIOR, Pedro. **Etofisiógrafo: Protótipo de um Sistema Telemétrico para Registro de Variáveis Fisiológicas em Pequenos Animais**. Universidade Federal de Santa Catarina – UFSC, Programa de Pós-Graduação em Engenharia Elétrica, Dissertação de Mestrado, 2006.

GROOVER, M. *Automation, Production Systems and Computer Integrated Manufacturing*, 2ª ed, Prentice Hall, 2000.

GUEDES, Gilleanes T. A. **UML - Uma abordagem prática**. 2 ed. Novatec, 2005.

JONES, Douglas W. *Stepping Motors Fundamentals*. MICROSHIP AN907, Microship Technology Inc., 2004.

KAMOGAWA, Marcos Y.; TEIXEIRA, Marcelo A.; **Autoamostrador de Baixo Custo para Análise por Injeção em Fluxo**. Revista Química Nova, Vol. 32, No 6, 1644 a 1646, 2009.

**KCAM** User Manual; Disponível em: [http://www.kellyware.com/anonymous/KCAM4\\_User\\_Manual\\_4.0.25.pdf](http://www.kellyware.com/anonymous/KCAM4_User_Manual_4.0.25.pdf).

KRUTCHEN, P. *The Rational Unified Process. An Introduction*. 2. Ed. Addison Wesley, 2000.

LOGASHANMUGAM, E.; SURESH, K. **Control Multiple Stepper Motors Through A PC's USB Port**, Ed. Online – Id 16125, Eletronic Design, <http://eletronicdesign.com>, 2007.

MAGALHÃES, L. S.; **Abastecimento de Estoque por Robô Microcontrolado**, em 8º Congresso Anual de Iniciação Científica – CONIC – SEMESP, 2008, Anuário da Produção de Iniciação Científica Discente, Vol XI, Nº 12, 2008, p695-705, Anhanguera Educacional S. A. São Paulo, Março de 2009.

MARCHEZAN, André R. **Ferramentas aplicadas no desenvolvimento de laboratório remoto e/ou presencial no ensino de engenharia eletrônica**. Universidade Estadual de Campinas – UNICAMP, Dissertação de Mestrado, 2006.

MARTINS, Murilo F; TONIDANDEL, Flávio. **Um Protocolo Confiável e Flexível de Comunicação para Futebol de Robôs**. Centro Universitário da FEI, Em: VII Simpósio Brasileiro de Automação Inteligente / II Latin American Robotics Symposium – VII SBAI / II IEEE LARS, São Luis, 2005.

MELIN, Patricia; CASTILHO, Oscar; *Intelligent control of a stepping motor drive using adaptive neuro-fuzzy inference system*. Journal Information Sciences, 170 133-151, Elsevier – ScienceDirect, 2004.

MOURA, R. S., GUEDES, L. A. **Modelagem de aplicações de automação industrial usando diagramas UML e reuso de componentes**. Universidade Federal do Piauí – UFPI, Simpósio Brasileiro de Automação Inteligente – SBAI, 2007.

MUÑIZ, Carlos; LEVI, Rafael; BENKRID, Meriem; RODRÍGUEZ, Francisco B.; VARONA, Pablo; **Real-time control of stepper motors for mechano-sensory stimulation**. Journal of Neuroscience Methods, Elsevier – ScienceDirect, 2008.

NAGEB, J. M.; RUHULLAH, Arief.; SALLEH, Sh-Hussain. **12 Channel USB Data Acquisition System For QT Dispersion Analysis**. Centre for Biomedical Engineering, University Technology Malaysia, Proceedings of the International Conference on Robotics, Vision, Information and Signal Processing, ROVISIP, 2005.

OGATA, Katsuhiko. **Engenharia de Controle Moderno**. 4ª ed. Pearson Prentice Hall, São Paulo, 2003.

OLIVEIRA, C. R.; OLIVEIRA, I.; SANTOS, H.; PEREIRA, A.; **Um Ambiente para a Prática Remota de Aulas Laboratoriais de Física (determinação da viscosidade de líquidos)**, Revista Brasileira de Informática na Educação - RBIE, v.17, N. 1 – 2009.

PENDER, Tom. **UML a Bíblia**. Elsevier, Rio de Janeiro, 2004.

RIBEIRO, Alexandre S. **Modelagem e Especificação de Controle de Sistemas Flexíveis de Manufatura, Utilizando Redes de Petri de Alto Nível**. Programa de Pós-Graduação em Mecatrônica, Universidade Federal da Bahia, Dissertação de Mestrado, 2009.

RIBEIRO, Marco A. **Automação Industrial**. 4ª ed, Tek Treinamento & Consultoria Ltda, Salvador, 2001.

ROQUE, A. S.; SILVA, A. D. **Sistemas de Controle para Equipamentos Automáticos Programáveis: Características Fundamentais para Compor uma Metodologia de Controle Flexível**. ROBOCONTROL, UNESP - Bauru - SP. 4º Workshop de Robótica Aplicada e Automação, 2010.

SILVA, Ricardo P. **UML2 em Modelagem Orientada a Objetos**. Florianópolis, Visual Books, 2007.

SILVA, Tony I. **Implementação de um sistema para geração e avaliação de movimentos em pacientes hemiplégicos**. Universidade Estadual Paulista – UNESP, Tese de Doutorado, 2007.

SILVEIRA, Paulo R; WINDERSON, E. **Automação e Controle Discreto**, Érica, São Paulo, 1999.

SILVEIRA, Ricardo C. A.; **Desenvolvimento de um Equipamento Mecânico com Controle Numérico Computadorizado para Produção de Protótipos em Escala.** Universidade Federal de Minas Gerais – UFMG, Programa de Pós-Graduação em Engenharia Mecânica, Dissertação de Mestrado, 2007.

SOUZA, Fábio, C. **Desenvolvimento de Metodologia de Aplicação de Redes de Petri para Automação de Sistemas Industriais com Controladores Lógicos Programáveis (CLP).** Universidade de São Paulo – USP, Departamento de Engenharia de Energia e Automação Elétrica, Dissertação de Mestrado, 2006.

SOUZA, Pedro E. **Accionamento de um Motor Passo a Passo com Velocidade e Direção Controladas e com Binário de Arranque Aumentado.** Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa, Engenharia Electrotécnica e de Computadores, Dissertação de Mestrado, Portugal, 2008.

SPERB, Daniel Q.; **Desenvolvimento de dispositivo programável de movimento passivo contínuo para membros inferiores.** Universidade Federal de Santa Maria - UFSM, Engenharia de Produção – Gerência da Produção, Dissertação de Mestrado, 2006.

STEFANOVITZ, Juliano P. **Criação de Conhecimento e Inovação na Indústria de Alta Tecnologia.** Universidade de São Paulo – USP, Departamento de Engenharia de Produção, Dissertação de Mestrado, 2006.

**TURBO CNC 4.01 User Manual;** Disponível em: <http://www.dakeng.com/man/turboenc.html>.

USB.ORG, **USB Especification 2.0.** Disponível em: <http://www.usb.org/developers/docs/>.

VILLEGAS, Angel; HERRERA, Isaac; GÓMEZ, Gerardo; RODRIGUEZ, José; LUGO, Edgar; PACHECO, José. **Aplicación para el monitoreo y control de procesos industriales basada em el estándar de comunicaciones OPC.** Facultad de Ingeniería, Valencia, Venezuela, Revista Ingenieria UC. Vol. 15, No 3, 7-18, 2008.

WECK, Manfred.; **Werkzeugmaschinen Band 3 – Autommatisierung und Steuerungstechnik.** Düsseldorf: VDI-Verl, Germany, 1989.

## **ANEXO A – PUBLICAÇÕES**

### **PUBLICAÇÕES**

**ROQUE, A. S.; SILVA, A. D. Sistemas de Controle para Equipamentos Automáticos Programáveis: Características Fundamentais para Compor uma Metodologia de Controle Flexível. ROBOCONTROL 2010, UNESP - Bauru - SP. 4º Workshop de Robótica Aplicada e Automação, 2010.**