

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

**IMPLEMENTAÇÃO DE TÉCNICAS DE CONTROLE
DE MOTOR DE PASSO EM APLICAÇÕES CNC**

DISSERTAÇÃO DE MESTRADO

Edinara Filipiak de Cristo

Santa Maria, RS, Brasil

2009

IMPLEMENTAÇÃO DE TÉCNICAS DE CONTROLE DE MOTOR DE PASSO EM APLICAÇÕES CNC

por

Edinara Filipiak de Cristo

Dissertação apresentada ao Curso de Mestrado do Programa de Pós-Graduação em Engenharia de Produção, Área de Concentração em Gerência da Produção, 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

2009

Universidade Federal de Santa Maria
Centro de Tecnologia
Programa de Pós-Graduação em Engenharia de Produção

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

**IMPLEMENTAÇÃO DE TÉCNICAS DE CONTROLE
DE MOTOR DE PASSO EM APLICAÇÕES CNC**

elaborada por

Edinara Filipiak de Cristo

como requisito parcial para obtenção do grau de
Mestre em Engenharia de Produção

COMISSÃO EXAMINADORA:

Alexandre Dias da Silva, Dr.

(Presidente/Orientador)

Inácio da Fontoura Limberger, Dr. (UFSM)

Leandro Costa de Oliveira, Dr. (UFSM)

Santa Maria, 20 de fevereiro de 2009.

Dedico este trabalho a todas as pessoas
que são importantes na minha vida.

AGRADECIMENTOS

Elaborar uma dissertação não é algo que possa ser feito sozinho. Os resultados não são somente de quem assina o documento. Muitas pessoas são envolvidas e devem ser lembradas aqui.

A começar por Deus por permitir-me à vida, força e vontade necessárias para levar este trabalho até o fim.

Aos meus pais Carlos e Maria, a quem eu tenho o maior respeito e admiração e que sempre me apoiaram da melhor maneira possível.

À Universidade Federal de Santa Maria por ter me concedido a oportunidade de realizar este trabalho.

À Empresa Tecnopampa por abrir suas portas para que fossem efetuados testes em seus equipamentos.

Aos colegas, professores e funcionários do PPGEP-UFSM pelo companheirismo e receptividade.

Ao professor Nelson Zang que direta e indiretamente contribuiu para a concretização deste mestrado.

Aos professores Inácio da Fontoura Limberger e Leandro Costa de Oliveira pela participação na banca examinadora.

Ao meu Orientador, Prof. Dr. Alexandre Dias da Silva pelo apoio, dedicação e tempo disponibilizado para orientação deste trabalho. Muito obrigada pela ajuda.

Em especial agradeço ao meu esposo Fernando de Cristo pelo apoio em todos os momentos, por ter me incentivado a fazer o mestrado e por ter me mostrado que posso alcançar os objetivos que busco, só basta acreditar.

A todos aqueles que não foram citados, mas não foram esquecidos, dedico este trabalho e que bons frutos possam ser colhidos dele.

A todos meus sinceros agradecimentos.

RESUMO

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

IMPLEMENTAÇÃO DE TÉCNICAS DE CONTROLE DE MOTOR DE PASSO EM APLICAÇÕES CNC

Autora: Edinara Filipiak de Cristo
Orientador: Prof. Dr. Alexandre Dias da Silva
Data e Local da Defesa: Santa Maria, 20 de fevereiro de 2009.

O objetivo deste trabalho é desenvolver uma metodologia de controle de equipamentos acionados por motores de passo, via porta paralela de microcomputadores, com até três eixos programáveis, para utilização em processos de automação. Para implementar o sistema proposto, utilizou-se um ambiente de desenvolvimento de programas computacionais, através do qual foi possível fazer a comunicação com a porta paralela do computador para transferência de dados de controle aos circuitos de acionamento dos motores de passos. O programa desenvolvido possui as seguintes funções: prioridade de execução do processo no sistema operacional do microcomputador; interpretação de seqüências de operações programadas através de arquivos de dados; visualização do arquivo de dados para controle do equipamento durante sua execução; visualização em tempo real dos dados de controle enviados à porta paralela; controle manual e automático do equipamento; execução de forma reversa dos movimentos aplicados aos eixos do equipamento; ajuste de velocidade dos eixos de movimento do equipamento de forma automática ou manual; e sistema de ajuda *online*. Características importantes apresentadas pelo sistema proposto são a precisão no controle da velocidade, a capacidade de armazenar configurações diversas para diferentes equipamentos, possibilidade de reproduzir de forma automática até 100 vezes uma operação programada, facilitando a produção em série e capacidade de armazenar uma seqüência de comandos manuais, possibilitando a programação por aprendizagem. Testes práticos de operação do sistema demonstraram a viabilidade de aplicação da técnica em processos que demandam automação programável.

Palavras-chaves: Comando Numérico Computadorizado, Motor de Passo, Automação Programável.

ABSTRACT

Master's Degree Dissertation
Production Engineering Post-Graduation Program
Federal University of Santa Maria

IMPLEMENTATION OF CONTROL TECHNIQUES OF STEPPING MOTOR INTO CNC APPLICATIONS

Author: Edinara Filipiak de Cristo
Advisor: Prof. Dr. Alexandre Dias da Silva
Date and presentation place: Santa Maria, February 20th 2009.

The objective of this work is to develop a methodology to control of equipments powered by stepping motors, via the computer parallel port, with up to three programmable axes, for use in automated processes. To implement the proposed system, used to be an environment for developing computer programs, which could make communication with the computer's parallel port for data transfer control circuits to drive the stepping motors. The program developed has the following functions: priority for implementing the process in the microcomputer operating system, interpretation of sequences of planned operations through data files, view the data file to control the equipment during its implementation; view real-time control data sent to the parallel port, manual and automatic control of equipment, implementation of reverse form of movements applied to the axes of equipment, adjustment of speed of the movement axes of the equipment in automatic or manual and online help system. Important features presented by the proposed system is the precision control of speed, the ability to store different settings for different equipment, possibility of playing in an automatic way to 100 times an operation planned by facilitating mass production, and capacity to store a sequence of manual commands, enabling the program by learning. Practical tests of operation of the system demonstrated the feasibility of applying the technique in cases that require programmable automation.

Keywords: Computer Numerical Control, Stepping Motors, Programmable Automation.

LISTA DE FIGURAS

Figura 1 – Motores de Passo (Maxwell Bohr, 2006).....	18
Figura 2 – Sistema de Controle em Malha Aberta.....	19
Figura 3 – Sistema de Controle em Malha Fechada.....	20
Figura 4 – Aspecto Final do Robô (Barboza et.al, 2003).....	21
Figura 5 – Aspecto Físico da Porta Paralela.....	23
Figura 6 – Princípio de Funcionamento de uma Mesa de Coordenadas.....	28
Figura 7 – Fluxograma de Execução de um Arquivo de Comandos.....	37
Figura 8 – Fluxograma de Execução Inversa de um Arquivo de Comandos.....	38
Figura 9 – Fluxograma da Função de Retorno Automático.....	39
Figura 10 – Tempo Total, Intervalo entre Pulsos e Intervalo de Duração do Pulso..	41
Figura 11 – Estrutura do Programa.....	42
Figura 12 – Janela Principal.....	45
Figura 13 – Barra de Menus e Barra de Ferramentas.....	46
Figura 14 – Menu Arquivo e Barra de Ferramentas.....	46
Figura 15 – Menu Editar e Barra de Ferramentas.....	46
Figura 16 – Menu Editar Configurações.....	47
Figura 17 – Selecionar uma Configuração.....	49
Figura 18 – Menu Exibir e Barra de Ferramentas.....	50
Figura 19 – Status da Porta Paralela.....	50
Figura 20 – Arquivo de Passos.....	51
Figura 21 – Menu Ajuda e Barra de Ferramentas.....	51
Figura 22 – Tela de Ajuda <i>Online</i> do Sistema.....	51
Figura 23 – Controle de Execução.....	52
Figura 24 – Opções de Processamento do Arquivo de Pulsos.....	53
Figura 25 – Controle de Velocidade.....	53
Figura 26 – Controle Automático.....	54
Figura 27 – Confirmação de Retorno Automático.....	54
Figura 28 – Controle Manual.....	55
Figura 29 – Simbologia para Controle Manual de <i>Drive</i> e Ferramenta.....	55
Figura 30 – Velocidade Atual de 1.200 mm/min.....	58
Figura 31 – Velocidade Atual de 1.000 mm/min.....	59

Figura 32 – Tempo de Espera	59
Figura 33 – Contagem Regressiva	60
Figura 34 – Comando de Velocidade Máxima	60
Figura 35 – Velocidade Atual de 2.400 mm/min.	61
Figura 36 – Velocidade Padrão	61
Figura 37 – Velocidade Padrão Restabelecida.....	62
Figura 38 – Máquina Router CNC Industrial 3 Eixos (Tecnopampa, 2009)	63
Figura 39 – Configuração da Máquina Router CNC Industrial 3 Eixos	63
Figura 40 – Cortes Realizados pela Máquina Router CNC Industrial 3 Eixos	64
Figura 41 – Corte da Geometria de um Quadrado	65
Figura 42 – Corte da Geometria de um Triângulo	66
Figura 43 – Corte da Geometria de um Hexágono	67
Figura 44 – Interrupção no Corte da Geometria de um Hexágono	67
Figura 45 – Corte da Geometria de um Octógono.....	68
Figura 46 – Interrupção no Corte.....	69
Figura 47 – Execução Inversa	69
Figura 48 – Corte Utilizando a Execução Inversa.....	70

LISTA DE TABELAS

Tabela 1 – Descrição da Porta Paralela SPP	24
Tabela 2 – Tabela de Códigos do LPTControl.....	44
Tabela 3 – Simulação do Resultado da Execução de um Arquivo	57
Tabela 4 – Tabela Comparativa KCam4 x LPTControl.....	71

LISTA DE ABREVIATURAS E SIGLAS

μ s - Microssegundos

BIOS - *Basic Input/Output System* (Sistema Básico de Entrada e Saída)

CA - Corrente Alternada

CAD - *Computer Aided Design* (Projeto Assistido por Computador)

CAM - *Computer Aided Manufacturing* (Manufatura Assistida por Computador)

CC - Corrente Contínua

CNC - Comando Numérico Computadorizado

DMA - Acesso Direto à Memória

ECP - *Enhanced Capabilities Port* (Porta com Capacidade Melhorada)

EPP - *Enhanced Parallel Port* (Porta Paralela Melhorada)

ISO - *International Organization for Standardization* (Organização Internacional para Normalização)

MDI - *Manual Data Input* (Entrada Manual de Dados)

NAFA - Núcleo de Automação de Processo de Fabricação da UFSM

PAL - Aplicações Lógicas Programáveis

SPP - *Standard Parallel Port* (Porta Paralela Padrão)

UFSM - Universidade Federal de Santa Maria

LISTA DE ANEXOS

Anexo A – Requisitos para Instalação do Sistema	78
Anexo B – Ajuda <i>Online</i> do Sistema	80

SUMÁRIO

1	INTRODUÇÃO	13
1.1	Estrutura do Trabalho	15
2	REVISÃO BIBLIOGRÁFICA	16
2.1	Acionamentos em Sistemas Automatizados	16
2.2	Controle de Motores de Passo Via Porta Paralela.....	22
2.3	Aplicações de Comando Numérico Computadorizado.....	26
2.4	Programação de Equipamentos CNC	29
3	METODOLOGIA.....	34
3.1	Requisitos do Programa.....	34
3.2	Estrutura do Programa.....	36
3.2.1	Executando um Arquivo de Comandos	36
3.2.2	Executando um Arquivo de Comandos na Ordem Inversa.....	37
3.2.3	Executando um Arquivo de Comandos com Retorno Automático	38
3.2.4	Temporização dos Pulsos	40
4	DESENVOLVIMENTO DO PROGRAMA.....	42
4.1	Tabela de Códigos do Programa LPTControl	42
4.2	Interface Usuário/Programa	44
4.2.1	Estrutura da Barra de Menus e da Barra de Ferramentas.....	45
4.2.2	Controle Automático	52
4.2.3	Controle Manual	54
5	RESULTADOS	57
5.1	Teste do Modo Automático	58
5.2	Testes Práticos	62
5.2.1	Teste 1 – Geometria de um Quadrado	64
5.2.2	Teste 2 – Geometria de um Triângulo.....	65
5.2.3	Teste 3 – Geometria de um Hexágono.....	66
5.2.4	Teste 4 – Geometria de um Octógono	67
5.3	Comparativo com o Software Comercial KCam4.....	70
6	CONCLUSÕES	72
	REFERÊNCIAS BIBLIOGRÁFICAS	74
	ANEXOS.....	77

1 INTRODUÇÃO

Devido ao grande avanço da tecnologia, máquinas com Comando Numérico Computadorizado (CNC) vêm evoluindo bastante e com isso propiciando melhorias em diferentes processos produtivos. Esses equipamentos se caracterizam por possuírem automação programável, o que o proporciona flexibilidade do processo produtivo, redução de erros, minimização de desperdício de material, aumento da qualidade, precisão na fabricação de peças, agilidade na produção, redução de custos, longevidade da ferramenta, entre outras. São requisitos fundamentais para a indústria de manufatura, que cada vez mais exige produtos com projeto otimizado, alta qualidade, preços competitivos e produção no menor intervalo de tempo possível.

Távora (1998) observa que as exigências mercadológicas aumentam à medida que as empresas renovam e projetam novas e melhores estratégias de concorrência, fazendo com que projetos sejam elaborados visando melhores processos, novos produtos, ou até mesmo ampliação e modernização da empresa.

No caso das máquinas CNC, Pereira (2003) diz que o fator tempo de processo é bastante reduzido pela eliminação dos tempos improdutivos e reduzir tempo tornou-se uma tarefa imposta pelas circunstâncias em qualquer indústria moderna.

O CNC pode ser usado em vários tipos de máquinas-ferramenta, e que são destinadas a diferentes operações de fabricação, como torneamento, fresamento, furação, corte, entre outros. Ele tem como principais particularidades, automatização, custo reduzido, flexibilidade e produtividade adequada. Os equipamentos de corte, como oxicorte, por exemplo, devido as suas características, são máquinas que podem facilmente ser projetadas segundo a tecnologia CNC, com um custo relativamente baixo.

Segundo Costa e Pereira (2006) equipamentos CNC, associados à modelagem digital encontrada em sistemas CAD (Computer-Aided Design) e CAM (Computer-Aided Manufacturing), suportam em grande parte a transferência do modelo de um produto para a máquina com pouca intervenção humana, além de

propiciar a substituição do meio de transmissão, papel ou verbal, para o eletrônico.

O uso de CNC em aplicações específicas está se difundindo bastante, isto é facilmente detectado na literatura atual. Um exemplo é o artigo de Rodrigues e Lacerda (2003) que teve como objetivo desenvolver uma mesa de coordenadas XY CNC de baixo custo, para corte e solda de chapas metálicas. O acionamento dos eixos da mesa de coordenadas é realizado por motores de passo. Os autores concluem que o protótipo pode ser desenvolvido com um custo realmente baixo e, além disso, apontam que a utilização da solução proposta pode aumentar a competitividade, reduzir tempo de fabricação e padronizar as peças fabricadas.

A aplicação de motores de passo, como dispositivo de acionamento de eixos de movimentos na fabricação de máquinas CNC, é uma alternativa com restrições, porém, de baixo custo, pois, a utilização de motores de passo permite a criação de sistemas de malha aberta onde a utilização de sensores é dispensável. Esse tipo de motor tem como característica mover o seu eixo em incrementos angulares bastante exatos, conhecidos como passos, em resposta a pulsos digitais aplicados a um controlador digital (*driver*). Eles são diferentes dos motores elétricos de corrente contínua. Ao se aplicar em suas bobinas um impulso elétrico, eles giram num ângulo fixo, de forma que se pode controlar, mediante um circuito eletrônico, a quantidade, velocidade e sentido dos movimentos. O número de pulsos e a cadência com que estes pulsos são aplicados controlam a posição, velocidade e a aceleração do motor, respectivamente.

O objetivo deste trabalho é desenvolver uma metodologia de controle de equipamentos acionados por motores de passo, via porta paralela de microcomputadores, com até três eixos programáveis, para utilização em processos de automação. A técnica consiste em interpretar dados pré-processados, referentes a operações programadas, para enviá-los, via porta paralela, aos controladores de motores de passo.

Diferente de outros sistemas computacionais disponíveis para aplicação de microcomputadores no desenvolvimento de máquinas CNC, que processam os códigos de um programa padronizado e geram a correspondente seqüência de pulsos, a metodologia proposta interpreta apenas os códigos de pulsos previamente processados. Através dessa simplificação, o sistema torna-se dedicado principalmente à transmissão de dados entre computador e circuitos acionadores dos motores. A vantagem dessa estratégia é que ele pode ser utilizado como

interface de comunicação computador-máquina para viabilizar o desenvolvimento de programas computacionais direcionados a equipamentos especiais.

O programa implementado neste trabalho possibilitará a aplicação de técnicas desenvolvidas em pesquisas no Núcleo de Automação e Processos de Fabricação (NAFA/UFSM) quanto ao pré-processamento de dados para aplicação de motores de passo em projetos de equipamentos.

As características do sistema para atingir esses objetivos são discutidas detalhadamente no decorrer do trabalho.

1.1 Estrutura do Trabalho

Esta dissertação está dividida em seis capítulos, sendo o primeiro uma introdução sobre o trabalho desenvolvido, apresentando os objetivos e os principais assuntos que estão detalhados no decorrer do trabalho. O segundo capítulo apresenta uma revisão bibliográfica contendo textos de diversos autores que possuem correlação com o trabalho aqui apresentado. O terceiro capítulo destina-se a apresentação das técnicas e da metodologia utilizada no trabalho para o desenvolvimento do programa. Já o quarto capítulo mostra uma descrição detalhada de todos os passos usados para o desenvolvimento do programa. O quinto capítulo mostra os resultados finais e os compara aos objetivos pretendidos. No último capítulo encontram-se as conclusões desta dissertação e sugestões para futuros trabalhos.

2 REVISÃO BIBLIOGRÁFICA

Ao longo deste capítulo são apresentados vários artigos e dissertações além de outros textos relacionados ao trabalho. Os mesmos encontram-se agrupados nas seções: acionamentos em sistemas automatizados, controle de motores de passo via porta paralela, aplicações de comando numérico computadorizado e programação de equipamentos CNC.

2.1 Acionamentos em Sistemas Automatizados

Pazos e Lovisolo (2002) entendem por acionamento o conjunto composto pelo motor, seu sistema de partida, o aparelho eletrônico de controle envolvido e o dispositivo de movimentação. Estes sistemas são utilizados para acionar componentes de máquinas ou equipamentos que requeiram algum tipo de movimento controlado e são formados normalmente pela combinação dos seguintes elementos:

- Conversor eletro-mecânico: A maioria das aplicações utiliza motor de corrente contínua (CC) (corrente na qual possui fluxo contínuo e ordenado de elétrons sempre na mesma direção) ou corrente alternada (CA) (corrente cuja magnitude e direção variam ciclicamente).
- Transmissão mecânica conversora: Polia, correia ou coroa, parafuso: adapta a velocidade e a inércia entre o motor e a máquina.
- Transformadores mecânicos: Engrenagens cilíndricas.
- Sensor de rotação: Tacogerador.
- Dispositivo eletrônico: Comanda e/ou controla a potência elétrica entregue ao motor.

Em muitas aplicações, têm-se como exigências de acionamento de avanço com precisão de posicionamento, homogeneidade e constância de velocidade, baixa

inércia, resposta dinâmica, alta capacidade de sobrecarga, movimentos e paradas rápidas com precisão, e reversão de movimentos. (PAZOS e LOVISOLO, 2002)

Os motores assíncronos ou de indução, por serem mais robustos e mais baratos, são os motores mais largamente empregados na indústria. Nestes motores o campo girante tem a velocidade síncrona, como nas máquinas síncronas. Por outro lado, os motores síncronos são motores de velocidade constante e proporcional com a frequência da rede. Os pólos do rotor seguem o campo girante imposto ao estator pela de alimentação trifásica. As máquinas de corrente contínua, em função do seu princípio de funcionamento, permitem variar a velocidade de zero até a velocidade nominal aliada com a possibilidade de se ter conjugado constante. Esta característica é de fundamental importância, pois dessa forma torna-se possível fazer o acionamento em várias aplicações que exigem ampla faixa de variação de velocidade com uma ótima regulação e precisão de velocidade, como acontece em máquinas de comando numérico (Pazos e Lovisolo, 2002).

Os mesmos autores (Pazos e Lovisolo, 2002) descrevem que, servomotores são motores utilizados nos servoacionamentos, onde os circuitos de alimentação dos servomotores encontram-se em uma unidade chamada servoconversor. Assim sendo, servoacionamento é o conjunto de servomotor e servoconversor, isto é, um sistema eletromecânico de controle de precisão. Os motores utilizados em servoacionamento podem ser de CA ou de CC, mas os servomotores CA apresentam uma precisão menor em relação ao servomotor CC. Segundo o trabalho o motor de passo, devido ao seu tamanho, custo e facilidade de controle digital têm aumentado muito seu uso, podendo substituir os servomotores convencionais em alguns casos.

Braga (2005), afirma que motores de passo são uma boa opção para acionamento em sistemas automáticos, preferencialmente especificados para aplicações que requerem baixo torque. Exemplos de motores de passo são mostrados na Figura 1.



Figura 1 – Motores de Passo (Maxwell Bohr, 2006)

Os motores de passo são usados em impressoras, *plotters*, *scanners*, *drives* de disquetes, discos rígidos e muitos outros aparelhos. Existem vários modelos de motores de passo disponíveis no mercado que podem ser utilizados para diversos propósitos. Podem ser utilizados para mover robôs, em equipamentos de clínicas radiológicas, câmeras de vídeo, ou brinquedos. Sua utilização é muito ampla e vai desde o controle de máquinas industriais (robôs) até pequenas demonstrações num curso de robótica. (BRAGA, 2005)

Segundo os autores Pazos e Lovisolo (2002), para trabalhar com motores de passo, é necessário conhecer algumas características de funcionamento, como a tensão de alimentação, a corrente elétrica suportada nas bobinas, o ângulo do passo e o torque. O torque do motor de passo depende da frequência aplicada à alimentação. Quanto maior a frequência, menor o torque, porque o rotor (conjunto eixo-imã que rodam na parte móvel do motor) tem menos tempo para mover-se de um ângulo para outro. A característica mais importante ao se escolher o motor é o número de graus por passo ou número de passos por volta. Os valores mais comuns para esta característica, também referida como precisão, são 0.72° , 1.8° , 3.6° , 7.5° , 15° e até 90° ou 500, 200, 100, 48, 24 e 4 passos por volta, respectivamente.

Em relação aos outros motores, o motor de passo apresenta vantagens como a total adaptação à lógica digital, o que permite o controle preciso da velocidade, direção e distância. Possuem ainda características de bloqueio, pouco desgaste e dispensa realimentação. O que mais o diferencia dos demais motores elétricos é o preciso controle sobre seus movimentos. Quando se quer obter um determinado ângulo de rotação em seu eixo, calcula-se o número de pulsos necessários de acordo com a resolução do motor e aplica-se o movimento. Outros motores podem

passar do ponto e, para voltar, precisam da realimentação negativa. (PAZOS e LOVISOLO, 2002)

Como desvantagens têm-se a má relação potência/volume. Para igual potência, motores de passo são maiores que outros tipos. Para maiores tamanhos, o controle também se torna mais complexo, pois um motor de passo tem que ser controlado por um dispositivo digital (*drive*), que na maioria das vezes é ligado a um sistema computadorizado. (PAZOS e LOVISOLO, 2002). Apesar das limitações, motores de passo são bastante importantes atualmente e sua utilização simplifica o desenvolvimento de soluções para diversas aplicações.

Segundo Silveira e Winderson (1999), o acionamento elétrico por motores de passos dividem-se em duas categorias:

- Sistemas de controle de malha aberta (Figura 2): neste caso, o sistema digital controla o ambiente através do acionamento dos atuadores de acordo com o que estiver programado. Este tipo de controle, embora bem mais econômico que o de malha fechada, se torna por vezes ineficiente, já que o sistema digital de controle não tem conhecimento de como estão sendo recebidas as ações dos atuadores por parte do ambiente. Ou seja, se os atuadores estão ou não obtendo sucesso em controlar o ambiente. É um sistema barato, de construção simples e fácil manutenção.



Figura 2 – Sistema de Controle em Malha Aberta

- Sistemas de controle de malha fechada (Figura 3): implementando-se esse tipo de sistema, a unidade de controle aciona os atuadores de acordo com os dados provenientes de sensores que, em tempo real, monitoram o sistema. Assim, todas as alterações no sistema geradas pelos atuadores são monitoradas pelos sensores, permitindo um controle mais rígido, seguro e eficiente do sistema. Isto torna esta técnica bem mais eficiente que a anterior, porém com a desvantagem de encarecer o processo de automação e torná-lo mais complexo para o desenvolvedor.

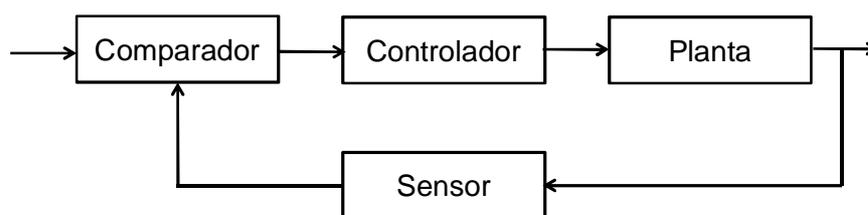


Figura 3 – Sistema de Controle em Malha Fechada

Segundo Pazos e Lovisolo (2002), os motores de passo podem ser bipolares ou unipolares. Em ambos os casos as fontes utilizadas são de tensão contínua e requerem um circuito digital que produza as seqüências de sinais para que o motor funcione corretamente. A forma com que o motor irá operar depende do que se deseja controlar. Existem casos em que o torque é mais importante, em outras a precisão ou mesmo a velocidade são mais relevantes. A principal vantagem de um motor de passo em relação a um motor CC é a precisão e a simplicidade do circuito e da lógica de acionamentos. O uso do motor de passo implica ainda na dispensa de sensores internos para odometria, que são usados para o controle da distância percorrida pelo robô desenvolvido no trabalho. Embora um motor CC possua maior velocidade e maior torque que um motor de passo, os requisitos estabelecidos para o projeto não inviabilizavam a utilização deste último.

Barboza et al. (2003) em seu artigo propõe uma metodologia de projeto e construção de robô móvel experimental, de baixo custo e para fins didáticos construído com motor de passo. O mesmo foi construído com o objetivo de se mover entre dois pontos pré-definidos em ambientes com presença de obstáculos fixos. A aplicação da metodologia de Kaikkonen et al. (1991) permitiu o sucesso no desenvolvimento do robô, consumindo-se para isto um tempo relativamente curto de execução do projeto. O projeto mecânico tinha como principal objetivo o custo reduzido. Os projetistas buscaram utilizar peças baratas e disponíveis. Na Figura 4, pode-se ver o robô construído pelo autores. Para fazer o robô se movimentar foi necessário dois motores de passos com acionamentos independentes, a fim de possibilitar a movimentação para qualquer direção. Com dois motores de passo atuando no mesmo sentido de giro, o robô efetuará movimentos para frente ou para trás. Em sentidos de giro contrários, o robô faz um movimento giratório sobre seu próprio eixo. Se um motor permanecesse parado e o outro girasse, então o robô poderia fazer uma curva para a esquerda ou para a direita.



Figura 4 – Aspecto Final do Robô (Barboza et.al, 2003)

Costa (2003) em seu trabalho desenvolveu um planejamento de trajetórias e simulação usando robôs que empregam motores de passo como força motriz em cada articulação. Tendo como dados uma trajetória ideal, a programação e simulação de movimento do robô. A trajetória ideal é substituída por outra capaz de ser executada pelo robô e otimizada, a programação é feita tendo em vista a trajetória aproximada percorrida e a simulação é realizada de tal forma que todos os componentes envolvidos devem ser considerados. Com essa complexidade é possível obter um nível de precisão bastante acentuado, que vai desde a trajetória exata até o movimento do robô. A saída da simulação pode ser visual, de tal forma que se possa observar a trajetória aproximada e a trajetória simulada. São analisados exemplos teóricos e é apresentado um exemplo prático de um braço com três barras e três motores.

Ferreira et al. (2001) descreve em seu artigo um processo de reconstrução tridimensional digital para construção de protótipos industriais, que é considerado de grande potencial em aplicações de diferentes campos da produção das indústrias. Este processo usa imagens de objetos iluminados com luz estruturada. O mesmo baseia-se em imagens múltiplas obtidas por um sistema de aquisição constituído por um plano de luz laser que executa um movimento de varredura de superfícies tridimensionais, controlado por um motor de passo. A geometria definida pelo sistema de aquisição de imagens permite, por triangulação, obter mapas densos tridimensionais. Estes mapas são modelos digitais de objetos reais e podem ser utilizados com eficácia na aquisição de modelos e protótipos tridimensionais de objetos. Este sistema utiliza tecnologia de baixo custo e é de fácil concretização. É um instrumento que pode ser utilizado em outras áreas, como as ciências físicas, médicas, forense ou ainda na produção cinematográfica e televisiva.

Na dissertação de mestrado de Sato (2002) foi desenvolvido um sistema de automação de ensaios estruturais, capaz de controlar a velocidade de carregamento. O sistema é composto por um quadro isostático, cilindros de carga, bomba hidráulica elétrica e um sistema de aquisição de dados. Após uma análise detalhada do sistema o autor chegou a conclusão que, com um custo baixo poderia, substituir o controle manual por um sistema controlado por computador. O sistema baseia-se na retro-alimentação e é ajustado com parâmetros de velocidade, carregamento e tempo. Inicialmente a informação parte de uma célula de carga e é enviada ao sistema de aquisição de dados, que transforma as informações em dados digitais. O computador processa as informações e de acordo com os ajustes iniciais faz as correções nos motores de passo acoplados às válvulas de pressão e vazão. Eliminando-se o sistema manual, certos parâmetros são mais fáceis de serem controlados automaticamente.

2.2 Controle de Motores de Passo Via Porta Paralela

Existem vários sistemas que propiciam a comunicação entre o computador e o meio exterior através da porta paralela para controlar vários tipos de processos.

Braga (2001) descreve técnicas da porta paralela que é uma interface de comunicação entre o computador e um dispositivo externo. A porta paralela (padrão DB-25 - Figura 5) pode transmitir 8 bits (1 byte) simultaneamente, levando cada bit por um condutor separado. A vantagem principal deste tipo de transmissão é que tem-se todos os bits de um byte ao mesmo tempo na entrada ou na saída do dispositivo, o que significa uma velocidade maior de transmissão de dados ou ainda o controle simultâneo de 8 linhas ou dispositivos. Este tipo de transmissão é indicado para comunicações de curtas distâncias, pois quanto mais curto o cabo menos interferências e maior a velocidade.



Figura 5 – Aspecto Físico da Porta Paralela

Além dos dados que são transferidos pelos condutores, têm-se sinais especiais de controle que também servem para informar ao computador o estado do dispositivo ligado a ele: se está ligado, se recebeu os dados transferidos, etc. Logo, através de sinais da porta paralela pode-se controlar diferentes funções de um projeto de mecatrônica ou ainda receber sinais de sensores ou de comandos externos. (BRAGA, 2001)

A porta paralela trabalha com 0V ou 5V, que significa desligado (bit 0) ou ligado (bit 1), ou ainda, nível baixo e nível alto. Para controle, todas as saídas da porta paralela são lógicas. Escrever “1” em um bit da porta significa que esse pino terá saída de +5V. No entanto existem alguns pinos que possuem lógica invertida, ou seja, para o sinal “0”, a voltagem enviada será de +5V. O computador nomeia as portas paralelas, chamando-as de LPT1, LPT2, LPT3, etc, mas, a porta física padrão é a LPT1, e seus endereços são:

- 378h, que é para escrever um byte de dados pela porta,
- 379h, para ler um byte de dados através da porta, e,
- 37Ah, para enviar e receber sinais de controle.

Esses dados podem ser utilizados por *drives* ligados a porta paralela para controle de diversos tipos de dispositivos. Entre esses dispositivos estão os motores de passo, os quais geralmente utilizam *drives* baseados em uma lógica de acionamento de três pinos para cada eixo. Esses pinos têm a função de ligar e desligar o motor, indicar o sentido da rotação do eixo e transmitir os pulsos para o motor. (LOPES, 2007)

A porta paralela possui três modos de operações que são configurados pela BIOS (*Basic Input and Output System* ou Sistema Básico de Entrada e Saída) Setup:
SPP (*Standard Parallel Port*) - Transmissão unidirecional 150KB/s (8 bits)

EPP (*Enhanced Parallel Port*) - Transmissão bidirecional 2 MB/s (8 bits)

ECP (*Enhanced Capabilities Port*) - Transmissão bidirecional+DMA 2 MB/s (8 bits)

A diferença entre EPP e ECP é que a ECP utiliza DMA (Direct Access Memory ou Acesso Direto a Memória). Na Tabela 1, tem-se a descrição da porta paralela SPP.

Tabela 1 – Descrição da Porta Paralela SPP

Sinal	Pinagem	Descrição
/STROBE	01	Indica se os dados estão prontos ou não para serem transmitidos. (0 = Dados prontos para serem transmitidos e 1 = Dados não prontos para serem transmitidos)
/ACK	10	Indica que a impressora está preparada para receber dados.
BUSY	11	Indica que a impressora não está preparada para receber dados.
PE	12	Indica que a impressora está sem papel para a impressão.
SELECT	13	Indica que a impressora está no estado "on line", pronta para receber informação.
/AUTO FD XT	14	O papel avança para o começo da próxima linha.
/ERROR	15	Indica quando ocorre algum tipo de erro (término do papel, impressora desativada).
/INIT	16	Reinicializa a impressora e limpa o buffer de impressão.
/SELECT INPUT	17	Os dados só podem ser transferidos para a impressora quando esta linha estiver em nível lógico baixo.
D0 a D7	2 a 9	Dados.
GND	18 a 25	Terra.

Para os quatro trabalhos relacionados na seqüência do texto é possível aplicar a técnica para controle de dispositivos acionados por motores de passo controlados via porta paralela desenvolvida neste trabalho.

No trabalho desenvolvido por Sperb (2008), é apresentado o projeto de um dispositivo programável, desenvolvido para tratamento fisioterápico na recuperação da articulação do joelho através da técnica de movimento passivo contínuo. O equipamento utiliza técnicas de automação flexível por meio da aplicação de motores de passo e é assistido por computador com interface de comunicação através da porta paralela. Segundo o autor, o aumento da prática esportiva também provoca um aumento considerável nas incidências de lesões, principalmente nos membros inferiores e, por isso, fica evidenciada a necessidade fisioterapêutica e

uma possibilidade de aplicação de princípios tecnológicos para otimização da mesma. A aplicação no projeto, de motores de passo controlados por computador, traz vantagens em comparação com equipamentos disponíveis atualmente no mercado para essa finalidade, pois além de possuírem preços elevados, devido à necessidade de importação, não suprem totalmente as necessidades fisioterápicas. Embora alguns modelos com custo mais elevado apresentem capacidade de memorização em seu leque de funções, esta não é flexível, ou seja, possuem limitações que comprometem o desempenho e a otimização do tratamento por parte dos fisioterapeutas.

Foi apresentada por Poll (2008), em sua dissertação de mestrado, uma metodologia para transformar calandras convencionais de tubos e chapas em equipamentos com sistema automático programável para execução de peças com raios fixos e/ou variáveis. O projeto conceitual também inclui motores de passo controlados via microcomputador e o programa de controle implementado com recursos computacionais para reconhecimento de desenhos definidos em sistema de projeto auxiliado por computador com o objetivo de gerar automaticamente o programa-tarefa capaz de executar a conformação da peça projetada. Desta maneira, além de satisfazer o requisito de operação programável, fazendo com que o equipamento trabalhe segundo os princípios da automação flexível, aspectos teóricos relativos ao processo de conformação mecânica para curvar tubos ou chapas também são considerados.

Fedel (2004) desenvolveu um projeto de um alimentador simples e com movimentos precisos para deficientes físicos. No desenvolvimento do trabalho constatou-se que uma mesa de coordenadas associada a um movimento rotacional, pode realizar todos os movimentos típicos da coleta do alimento. Diante disso, elaborou-se o projeto de automação usando uma planilha de cálculo para armazenar a velocidade e o tempo dos movimentos possibilitando simular as condições de projeto. Fisicamente o equipamento é composto de uma mesa XY projetada com elementos mecânicos padronizados e acionada por motores de passo operando em malha aberta, comandados digitalmente por um microcontrolador de oito bits, que controla os movimentos envolvidos.

Outro exemplo de aplicação de motores de passo controlados por microcomputador é apresentado por Peracetta e Uniandrade (2003), que desenvolveram um sistema para controle de uma estufa agrícola, que é composto

por uma interface e os módulos externos. A interface controla o tráfego de dados entre o computador e os módulos externos enquanto que os módulos externos servem como condicionadores de sinais provindos de sensores.

Os autores demonstraram que é possível desenvolver um sistema de baixo custo capaz de fazer a comunicação entre um computador e o meio exterior propiciando a mensuração e o controle de processos comuns em experiências laboratoriais de cunho científico ou didático.

2.3 Aplicações de Comando Numérico Computadorizado

A forma mais comum de automação flexível no meio industrial é identificada pela aplicação de máquinas equipadas com CNC, que é definido por Machado (1990) como um sistema eletrônico capaz de receber informações, compilar essas informações e transmiti-las em forma de comando à máquina operadora, fazendo com que esta realize as operações na seqüência programada, sem a intervenção do operador.

Ramalho (1994) escreve que o corte de materiais é uma das mais importantes etapas na cadeia produtiva dos aços. Tanto as chapas prontas devem ser cortadas em peças para seu destino final, como as sucatas devem ser cortadas em peças de menores dimensões para facilitar seu processamento posterior. Os cortes dividem-se em: mecânicos, por fusão de metal, por combinação de fusão e vaporização e por reação química, que é um corte que envolve os mecanismos de aquecimento através de chama e reações exotérmicas, seguido de oxidação do metal e posterior expulsão através de jato de O_2 . Pode ser usado como exemplo o corte oxicom bustível, ou oxicorte. O processo de oxicorte pode ser classificado como um processo de remoção química. O autor também defende que de todos os métodos, há uma particular preferência pelo oxicorte devido a ser o processo mais barato de implementar, com equipamentos mais simples, com a maior facilidade de treinamento do operador e, particularmente na faixa de espessuras maiores que 30 mm, ser o processo que propicia o menor custo por metro cortado. O oxicorte é como um processo de seccionamento de metais pela combustão localizada e contínua devido à ação de um jato de O_2 de elevada pureza, agindo sobre um ponto

previamente aquecido por uma chama oxi-combustível. É um processo de combustão em que algumas condições são exigidas. Entre as quais podem-se citar: temperatura de início de oxidação viva deve ser inferior à temperatura de fusão do metal, a reação deve ser suficientemente exotérmica para manter a peça na temperatura de início de oxidação viva, os óxidos formados devem estar líquidos na temperatura de oxicorte para facilitar seu escoamento e dar continuidade ao processo, o material a ser cortado deve ter baixa condutividade térmica e os óxidos formados devem ter alta fluidez.

Excluindo os processos com princípio mecânico de corte, os equipamentos de corte metálico envolvem forças de acionamento que podem facilmente ser suprimidas por motores de passo. Mesas de coordenadas XY, além de serem utilizadas em muitas aplicações nas áreas comercial e de serviços, são um exemplo típico de aplicação da tecnologia CNC no meio industrial, principalmente em operações de corte. A grande variedade de mesas XY existentes no mercado usa basicamente dois tipos de acionamento: o acionamento por motor de passo, que trabalha em uma estrutura de malha aberta e o acionamento por servo mecanismo, que utiliza motores de corrente contínua ou de indução em malha fechada. Neste caso, há a necessidade de sensores de posição que servem para fornecer a posição angular e a velocidade do eixo do motor. (FILHO, 2007)

Rank e Batti (2007) descrevem que as máquinas CNC são comandadas por um sistema de coordenadas cartesianas na elaboração de qualquer trabalho. O sistema de coordenadas é definido no plano formado pelo cruzamento de uma linha paralela ao movimento longitudinal (X), com uma linha paralela ao movimento transversal (Y). Todo movimento da ponta da ferramenta é descrito no plano XY, em relação a uma origem preestabelecida (X_0, Y_0). Além dos eixos X e Y, existe o eixo Z, que define a altura da ferramenta em relação à peça a ser cortada (Figura 6).

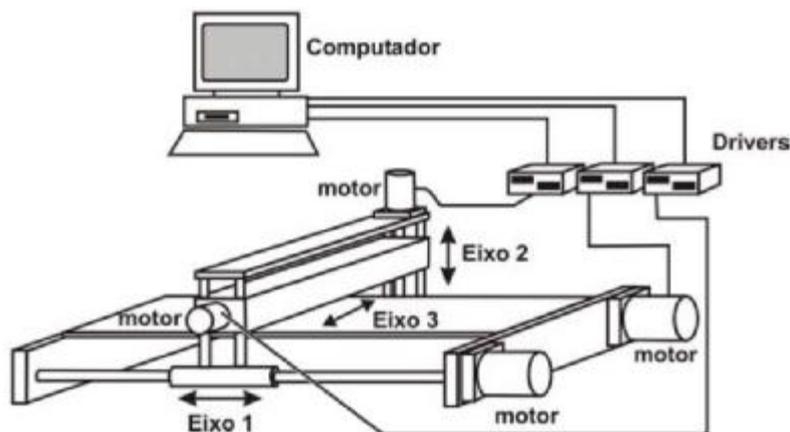


Figura 6 – Princípio de Funcionamento de uma Mesa de Coordenadas
(Modificado de: Rank e Batti (2007))

Os autores também dizem que precisam ser observados os sentidos positivos e negativos de cada eixo. Nem todas as máquinas possuem os sentidos dos sistemas de coordenadas iguais ao apresentado, portanto, é importante o profissional da programação saber identificar qual sistema de coordenadas a máquina a ser trabalhada possui. No sistema de programação CNC é possível utilizar dois tipos diferentes de sistema de coordenadas: as coordenadas absolutas que se referem sempre a um ponto fixo da peça, o ponto zero, e as coordenadas incrementais que se referem sempre ao posicionamento anterior. O trabalho desenvolvido por Bandeira (2005) apresenta o protótipo de uma mesa de coordenadas projetada para operar com motores de passo.

O processo de fabricação com maior adequação à tecnologia CNC é a usinagem. Penteado (2002) diz que máquinas modernas de usinagem conferem à peça uma precisão dimensional e um acabamento superficial que em muitos casos dispensam outros processos de fabricação. A maioria das peças produzidas industrialmente, mesmo quando obtidas através de outros processos, recebe seu formato final através de usinagem. É sabido que quanto maior o grau de precisão exigido no acabamento, mais aprimorado se torna o processo de usinagem e, conseqüentemente maiores os custos.

O processo de usinagem é um dos mais antigos métodos utilizados pelo homem para produção em série. A história da usinagem acompanha o desenvolvimento da indústria, desenvolvimento de novos materiais, necessidade de aumento de produção e produtividade. É um processo que depende de vários fatores, mas os mais importantes são a máquina-ferramenta, o material a ser

usinado, a ferramenta de corte e os fluidos de corte, incluindo o método de aplicação. A máquina-ferramenta evoluiu em vários aspectos, não apenas em termos estruturais, como o aumento de rigidez e com isto diminuindo vibrações mecânicas e melhorando o processo, mas também com relação à precisão de movimentos. Além disso, os recursos de comando numérico têm evoluído muito ultimamente. É um processo que usa grandes forças de cortes (SILVA, 2006) e, portanto, não é adequado à aplicação de motores de passo. Entretanto, equipamentos de pequeno porte ou dedicados a operações de acabamento, que exige esforços reduzidos, são perfeitamente adaptáveis a essa tecnologia.

2.4 Programação de Equipamentos CNC

Segundo Pereira (2003), o CNC é uma forma apropriada para a solução de problemas mais complexos de manufatura. As máquinas-ferramenta mais usadas nos processos de usinagem são tornos, furadeiras, fresadoras, plainas e retificadoras.

Uma máquina pode possuir vários movimentos, normalmente classificados em movimentos de translação ou rotação. Costuma-se dizer que cada um desses movimentos é um “eixo” da máquina, associando-se uma letra a ele. As letras mais usadas são X, Y, Z e W para movimentos lineares e A, B e C para movimentos angulares.

As técnicas de programação incluem uma variedade de procedimentos, que vão desde o método manual até o automatizado. Cada autor apresenta a sua classificação, com variações específicas. A maioria dos autores classifica as técnicas de programação em três tipos: programação manual, programação MDI (*Manual Data Input*) e programação via CAD/CAM.

Segundo Lynch (1994), a programação manual refere-se à elaboração de um programa, sem a utilização de recursos computacionais para determinar trajetórias de ferramentas, pontos de interseção de perfis, avanços, velocidades, etc. Nesse tipo de programação, as instruções de processamento são documentadas em um formulário chamado manuscrito do programa, que é uma listagem das posições da ferramenta em relação à peça, que a máquina precisa seguir para executar o

processamento. A listagem também inclui outros comandos com informações sobre velocidades, avanços, ferramental, etc. Este tipo de programação é indicado para aplicações em trabalhos de torneamento e fresamento que envolvam o processamento de peças geometricamente simples e que exijam poucas operações.

Segundo (Fortin, 2004), a programação manual é baseada no padrão “G”, da norma ISO 6983. A quantidade de máquinas equipadas com esta interface ainda é a maioria, mas futuramente será substituída pela norma ISO 14649. Pode-se dizer que existe uma grande quantidade de produtos que pode ser processado somente com a programação em 2 ½ eixos. Alguns autores estimam que essa quantidade pode ser superior a 80%, em se tratando de componentes mecânicos fabricados em grandes lotes.

Lynch (1997) escreve que o sistema MDI é um procedimento no qual o programa CNC é inserido diretamente no comando da máquina-ferramenta. Conseqüentemente, o uso de sistemas de transferência de dados é desnecessário. Os sistemas MDI são entendidos como sendo o caminho para as pequenas fábricas introduzirem a tecnologia CNC nas suas operações, sem a necessidade de adquirir equipamentos especiais de programação e contratar programadores. Isso permite à fábrica fazer um investimento inicial mínimo, para iniciar a transição em direção à tecnologia CNC.

A programação por aprendizagem é outra forma versátil de se criar um programa CNC. Bastante utilizada também em robôs de repetição, é um método em que um robô é movido fisicamente ponto-a-ponto sobre a trajetória desejada. Consiste em levar o braço do robô a mover-se na seqüência de movimentos requeridos e registrar os movimentos na memória do controlador que servirá para ele guiar-se sozinho posteriormente. Estudos atuais têm se concentrado no desenvolvimento da programação por aprendizagem, já que estão mais próximos da realidade do ser humano, e, portanto, permite simular as ações humanas com maior facilidade (COSTA, 2003). Em equipamentos CNC, a programação por aprendizagem consiste na memorização de operações executadas por entrada manual de dados (sistema MDI).

A programação automática de equipamentos CNC é realizada pelos sistemas CAD/CAM. O CAD é um *software* onde são criados os projetos de peças usando um computador e o CAM é o sistema que proporciona a fabricação de peças projetadas no CAD utilizando um computador. A tecnologia CAD/CAM corresponde à

integração das técnicas CAD e CAM num sistema único e completo. Isto significa, por exemplo, que se pode projetar um componente qualquer na tela do computador e transmitir a informação por meio de interfaces de comunicação entre o computador e um sistema de fabricação, onde o componente pode ser produzido automaticamente numa máquina CNC.

A programação CNC via CAD/CAM é uma forma mais avançada. Nele um sistema gráfico interativo, equipado com *software* de programação CNC, é usado para facilitar a atividade de programação. Neste método, o programador trabalha numa estação CAD/CAM para selecionar ferramentas, métodos e procedimentos para usinar a peça projetada no CAD. Os sistemas de programação CNC com base em sistemas CAD/CAM foram criados para solucionar as falhas da programação manual e automatizar o processo. São concebidos para programar tanto peças simples como complexas, de forma mais rápida, mais fácil e com um grau mais alto de precisão. Suas aplicações são necessárias nas indústrias de fabricação de moldes e aeroespacial, que trabalham com usinagem de peças tridimensionais complexas e cujos programas são muito extensos (LYNCH, 1994).

Embora os sistemas comerciais CAD/CAM sejam eficientes e forneçam recursos de programação relacionados aos mais variados processos, trabalhos de pesquisa nesta linha são paralelamente desenvolvidos com objetivo de ampliar a tecnologia, focar em aplicações específicas ou mesmo fornecer alternativas simplificadas visando diferentes tipos de usuários. Costa e Pereira (2006) apresentam um modelo de *software* de baixo custo desenvolvido para possibilitar a programação CNC para 2 ½ eixos. Neste caso, comanda-se apenas dois eixos ao mesmo tempo, e mantém o terceiro em posição constante. A implementação foi fundamentada em características programáveis e destinada ao conceito de um comando comercial. É um *software* para pequenas empresas que ainda utilizam a programação manual na fabricação de peças de baixa complexidade. A elaboração de um programa CNC passa pelas seguintes etapas: edição, simulação e transmissão do código para a máquina, independente da técnica a ser utilizada. A norma brasileira NBR 6162 (1989) define seis tipos de movimentos fundamentais na definição dos processos de usinagem. No contexto da programação, pode-se reduzi-los, sem perda significativa, a três tipos: o movimento de corte, o de posicionamento e o de avanço. O editor é representado por ícones gráficos, para os quais são criados formulários contendo os campos preenchidos pelo usuário, o que reduz a

necessidade e elimina erros de sintaxe de programação. A simulação baseia-se na representação gráfica da trajetória das ferramentas em um plano escolhido. A transferência do código para a máquina tem como objetivo a redução do tempo, quando comparado à introdução via teclado da máquina, e a eliminação de eventuais erros de digitação, a mesma é feita pela porta serial RS232. Os parâmetros de comunicação, tais como velocidade, paridade, tamanho de palavra, bits de parada, modo de transferência e o número da porta serial devem ser previamente definidos pelo usuário e o arquivo a ser transmitido deve conter apenas caracteres ASCII e possuir tamanho compatível com a memória disponível na máquina.

A tecnologia de baixo custo proposta foi implementada com o intuito de reduzir o tempo de programação e, reduzir o custo dos processos de usinagem em ambientes fabris. A avaliação foi realizada em três etapas: seleção de empresas que atendessem a um determinado perfil, treinamento e avaliação de técnicos dessas empresas e teste do protótipo no chão de fábrica. Este sistema, embora não possa ser considerado um sistema CAD/CAM (o usuário digita as coordenadas para gerar a trajetória da ferramenta), quando comparado àqueles observados em empresas que ainda fazem uso da programação manual, representa uma alternativa viável para redução do tempo total de usinagem e conseqüentemente de custos de fabricação. (COSTA e PEREIRA, 2006)

A utilização de microcomputadores como unidade de controle de equipamentos CNC tem a vantagem de poder disponibilizar no ambiente de trabalho, recursos de operação avançados ou dedicados. Sistemas computacionais comerciais disponíveis para esse fim é uma alternativa no desenvolvimento de projetos de equipamentos CNC. Kellyware (2007) descreve o funcionamento do *software* CNC KCam 4, que possui funções como estabelecimento de trajetórias de ferramentas e simulação, aplicadas a processos como fresamento 3D, furação e corte. Foi projetado para ler arquivos criados por aplicações CAD e controlar equipamentos CNC ligados à porta paralela do computador.

Outro exemplo de *software* comercial é o sistema Mach3 (2005). O programa interpreta arquivos de desenho do CAD em formato padronizado e converte os dados geométricos para o código de programação CNC. Possui simulação da trajetória da ferramenta e para realização de cortes são usadas funções para

executar e pausar o programa, e que permitem continuar ou parar o processo tendo como referência uma linha do programa CNC.

Para controlar motores de passo diretamente do ambiente CAD, Postal (2000) propôs uma metodologia que utiliza recursos computacionais disponíveis nesses sistemas para desenvolvimento de aplicativos. O programa implementado foi direcionado a mesas de coordenadas e é capaz de receber como dado um desenho feito no CAD, que é processado e traduzido em pulsos para cada um dos eixos controlados, que são enviados aos controladores dos motores pela porta paralela.

Goellner (2006) desenvolveu em sua dissertação de mestrado uma ferramenta computacional para operar como uma unidade de CNC. O sistema tem como função controlar a comunicação com os dispositivos da máquina, bem como possibilitar a criação ou a utilização de um arquivo de texto com a programação CNC, interpretar, analisar, simular e executar as informações de uma seqüência pré-programada de operações. A implementação de funções de controle, de recursos de operação do equipamento, de simulação do programa CNC em tela gráfica e de técnicas de interpolação linear e circular, mostrou a viabilidade de aplicação de computadores como unidade de comando em desenvolvimento de máquinas CNC com acionamento por motores de passo. A ferramenta tem uma precisão milesimal na interpretação dos programas CNC. É uma ferramenta que poderá ser utilizada em pequenas empresas da área metal-mecânica para modernização de máquinas-ferramenta que não possuam grande carga em seus eixos, como nos processos de corte por oxiacetileno, laser, plasma e jato-d'água. Nestes processos não existe atrito entre a ferramenta e a peça, deste modo não existe carga, além do peso do cabeçote da ferramenta, para a movimentação dos eixos. É uma ferramenta que também poderá ser usada para fins didáticos.

3 METODOLOGIA

O capítulo anterior mostra a diversidade de aplicações de equipamentos com automação programável e que a eficiência dos processos produtivos em que estão inseridos depende da qualidade dos programas de cada operação. O grande investimento em sistemas de programação automática, dedicados a equipamentos de comando numérico computadorizado, confirma esta tendência. Observa-se também que recursos computacionais permitem implementar funções de geração automática de programas para tarefas ou equipamentos especiais. Exemplo disso são as ferramentas disponíveis em sistemas CAD para desenvolvimento de aplicativos.

Neste capítulo é descrito o desenvolvimento do programa LPTControl que tem como finalidade viabilizar a especificação de motores de passo em aplicações que envolvam técnicas de automação programável. O nome LPTControl vem do termo utilizado pelos sistemas operacionais Windows para se referir a porta paralela do computador (LPT1, LPT2 e LPT3) e Control se refere ao controle exercido pelo programa nesta interface de hardware.

3.1 Requisitos do Programa

O programa LPTControl foi projetado levando-se em consideração necessidades apuradas junto a usuários de aplicações similares pelo NAFA/UFSM. Além disso, foi realizado um levantamento de características oferecidas por outros programas comerciais ou acadêmicos. Tomando por base estes dados definiram-se os seguintes requisitos para o programa proposto neste trabalho:

- Especificar códigos de pulsos para acionamento individual ou combinado dos motores controlados.
- Utilizar formato texto nos arquivos contendo os códigos de pulsos para interpretação pelo programa.

- Ter prioridade de execução no processo de transferência de dados. O programa deve ter prioridade de tempo real sobre os demais processos, evitando assim interrupções no processo de transmissão de dados.

- Possibilidade de visualizar o arquivo de pulsos durante sua execução, à medida que os pulsos estão sendo enviados para a porta paralela. Com esta visualização pode-se saber em que ponto o arquivo está sendo processado.

- Possibilidade de execução inversa do arquivo de pulsos, ou seja, execução do arquivo de pulsos até um ponto desejado e, posteriormente, repetir alguma parte do processo na seqüência inversa, ou então, pode-se interromper o processo ou até mesmo parar a execução do arquivo.

- Possuir uma ajuda *online* detalhada para cada item que compõe a estrutura do programa auxiliando nas dificuldades que o usuário possa ter com relação ao seu uso.

- Possibilidade de visualizar os sinais que estão sendo enviados pelo programa para a porta paralela em tempo real.

- Permitir controle manual do equipamento para operar individualmente cada um dos eixos de movimento.

- Programar o acionamento automático da ferramenta, isto é, aceitar função de ligar ou desligar a ferramenta como comando no programa CNC.

- Possibilitar ajuste de velocidade manual ou automaticamente, conforme a necessidade do usuário ou as características da máquina que será usada.

- Especificar o intervalo de duração dos pulsos em microssegundos (μs) para permitir a geração de um maior número de pulsos por segundo e conseqüentemente obter maior rotação e controle mais apurado da velocidade.

- Armazenar configurações diferentes para múltiplos equipamentos permitindo que diferentes equipamentos possam ser conectados a porta paralela do computador.

- Possibilidade de repetir um programa até 100 vezes. Esta característica é útil em aplicações onde pode ser necessário repetir um padrão várias vezes.

- Ter recurso para programação por aprendizagem.

- Apresentar todas as telas do programa em português, facilitando a utilização do usuário brasileiro.

3.2 Estrutura do Programa

A seguir estão descritos alguns recursos mais complexos que foram implementados no programa.

3.2.1 Executando um Arquivo de Comandos

Cada comando lido é convertido em sinais de acordo com as configurações do equipamento armazenadas no sistema e então é transmitido através da porta paralela. A seguir, é lido o próximo comando e o processo é repetido até atingir o fim do arquivo ou ser interrompido pelo usuário. O tempo que um comando leva para ser executado pode variar de acordo com a velocidade definida pelo usuário. Este tempo é calculado pelo sistema de acordo com o tamanho do deslocamento em cada eixo e a velocidade solicitada pelo usuário.

Através da Figura 7 pode-se visualizar que o primeiro processo na execução de um arquivo de comandos é realizar a leitura, a qual envia o comando lido para a memória do computador. Se durante o processo de leitura for verificado que o fim do arquivo foi atingido o processo é encerrado no passo 5. Senão, com o comando em memória o sistema converte este em sinais que possam ser compreendidos pelo equipamento que está sendo controlado. No passo 3 é calculado o tempo que estes sinais devem ficar disponíveis na porta paralela para serem lidos pelo equipamento. Então é feita a transmissão dos sinais para a porta paralela no passo 4, os quais ficam disponíveis durante o intervalo de tempo calculado no passo 3. Após este período de espera o processo é repetido até que se atinja o final do arquivo ou seja interrompido pelo usuário.

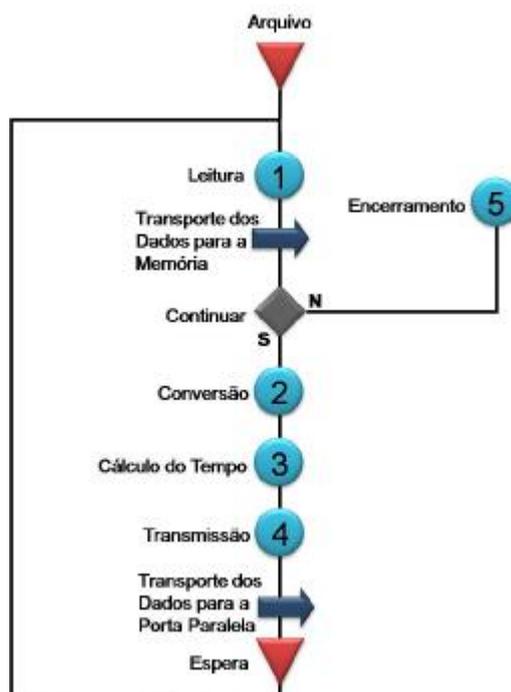


Figura 7 – Fluxograma de Execução de um Arquivo de Comandos

3.2.2 Executando um Arquivo de Comandos na Ordem Inversa

Para executar um arquivo na ordem inversa é necessário alterar a maneira como são lidos os comandos no arquivo. Na operação normal cada comando lido do arquivo é executado e então o processamento segue pela leitura do comando seguinte. Quando executando em ordem inversa, o último comando efetuado é lido e executado invertendo-se os valores dos bits que informam o sentido de deslocamento de cada eixo. Em seguida é lido o comando anterior e repetido o processo sucessivamente até este ser interrompido pelo usuário ou atingir-se o início do arquivo.

A Figura 8 exibe o fluxograma de execução inversa de um arquivo de comandos, onde o primeiro passo é realizar a leitura, a qual envia o comando que foi lido para a memória do computador. A decisão sobre continuar o processamento depende de o fim do arquivo ter sido atingido ou uma interrupção ter sido solicitada pelo usuário, caso isso ocorra o passo 6 é executado. Caso contrário, depois que os dados são enviados para a memória o sistema faz a inversão dos bits direcionais, alterando desta maneira o sentido do deslocamento em cada eixo. No passo 3 são

gerados os sinais de acordo com a configuração do equipamento. No passo 4 é calculado o tempo que os sinais devem permanecer na porta paralela. A seguir os sinais são transmitidos à porta paralela onde aguardam pelo tempo de espera calculado anteriormente. Assim que o tempo de espera acaba o processo retorna ao início para que seja repetido.

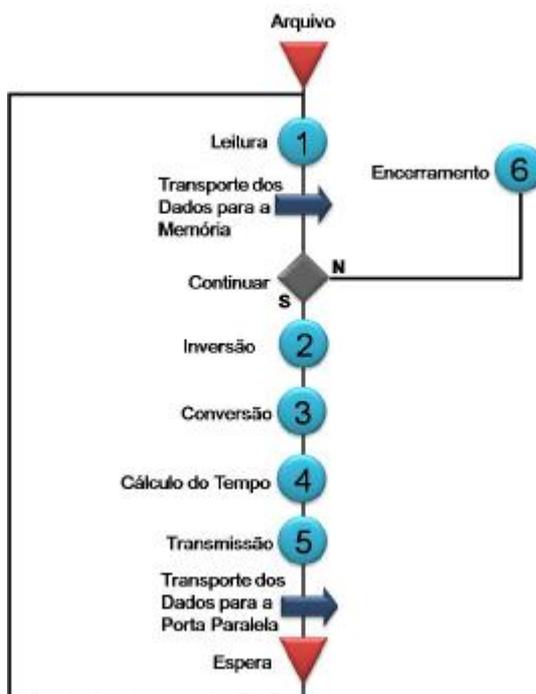


Figura 8 – Fluxograma de Execução Inversa de um Arquivo de Comandos

3.2.3 Executando um Arquivo de Comandos com Retorno Automático

O Retorno Automático é um recurso disponível a partir de um deslocamento manual efetuado pelo usuário durante uma pausa na execução de um arquivo de comandos. Sempre que houver uma pausa na execução de um arquivo de comandos, caso o usuário efetue comandos manuais, estes são armazenados em uma estrutura de dados temporária na memória. Quando o processamento do arquivo é retomado, caso seja solicitado pelo usuário, o equipamento pode retornar ao ponto onde a execução foi interrompida através da execução de todos os comandos armazenados na memória em ordem inversa e invertendo-se os bits de direção.

A Figura 9 traz o fluxograma da função de retorno automático. No processo 1

o usuário encontra-se executando um arquivo de comandos, caso efetue uma pausa na execução poderá acionar o módulo de controle manual. Todos os comandos efetuados no modo manual são armazenados pelo passo 3 em uma memória temporária. Assim que o usuário solicitar o retorno ao modo automático, será solicitado sobre efetuar ou não o retorno automático a posição onde o processo foi interrompido. Em caso positivo o processo segue pelo passo 4 onde é feita a inversão dos bits de direção dos eixos. No passo 5 os comandos são convertidos de acordo com a configuração especificada pelo usuário para o equipamento. Em seguida é calculado o intervalo de duração do pulso de acordo com a velocidade especificada pelo usuário. No passo 7 os dados são enviados a porta paralela e o processo é repetido até que atinja o ponto onde havia sido interrompido antes e então segue pela execução automática do arquivo de comandos. Por outro lado, caso usuário opte por não utilizar o recurso, ele poderá optar por continuar executando o arquivo a partir do ponto onde a ferramenta encontra-se no momento.

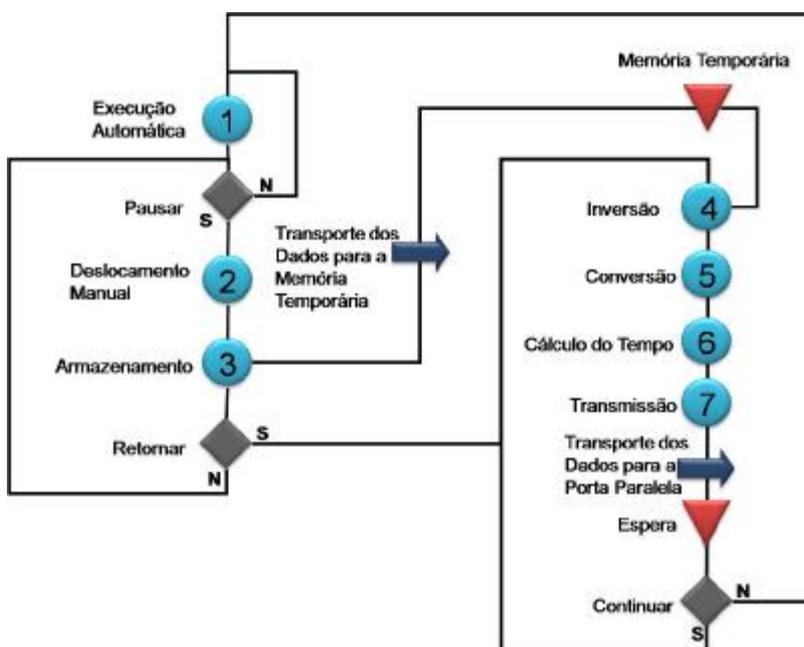


Figura 9 – Fluxograma da Função de Retorno Automático

3.2.4 Temporização dos Pulsos

O programa desenvolvido neste trabalho controla equipamentos baseados em motores de passo conectados a um *drive* que se comunica com o computador através da porta paralela. Para controlar esses motores devem-se energizar suas bobinas enviando para o pino de controle específico a quantidade de pulsos necessários para girar o rotor de passo para que possa produzir um movimento correspondente à distância que se quer percorrer.

Para que o *drive* do motor de passo consiga perceber a geração de um pulso, deve ser mantido o pulso por um determinado tempo na porta, que significa manter por certo tempo o pino da porta paralela energizado. Também é necessário que exista um tempo mínimo entre um pulso e outro, pois é necessário esse tempo em que o pino fica sem energia para que o *drive* perceba a existência de um novo pulso.

O tempo mínimo é informado pelo usuário na janela de configurações do sistema podendo variar de um equipamento para outro de acordo com o motor ou *drive* utilizado, características do equipamento ou aplicação.

O tempo total de duração de um pulso é calculado dividindo a resolução pela velocidade. Quanto maior o tempo total do pulso, menor a velocidade e vice-versa. Na Figura 10, tem-se um gráfico que indica o intervalo entre pulsos e o intervalo de duração do pulso. O primeiro pulso mostrado é executado em uma velocidade que equivale o dobro da velocidade do segundo pulso, isto porque, como pode-se observar o tempo total gasto no primeiro pulso é de $20\mu\text{s}$, enquanto que no segundo o tempo total é de $40\mu\text{s}$. Além disso, é possível perceber que no segundo pulso há uma desaceleração do equipamento que passa a operar em uma velocidade menor do que a anterior, uma vez que há uma relação inversamente proporcional entre o tempo total de um pulso e a velocidade.

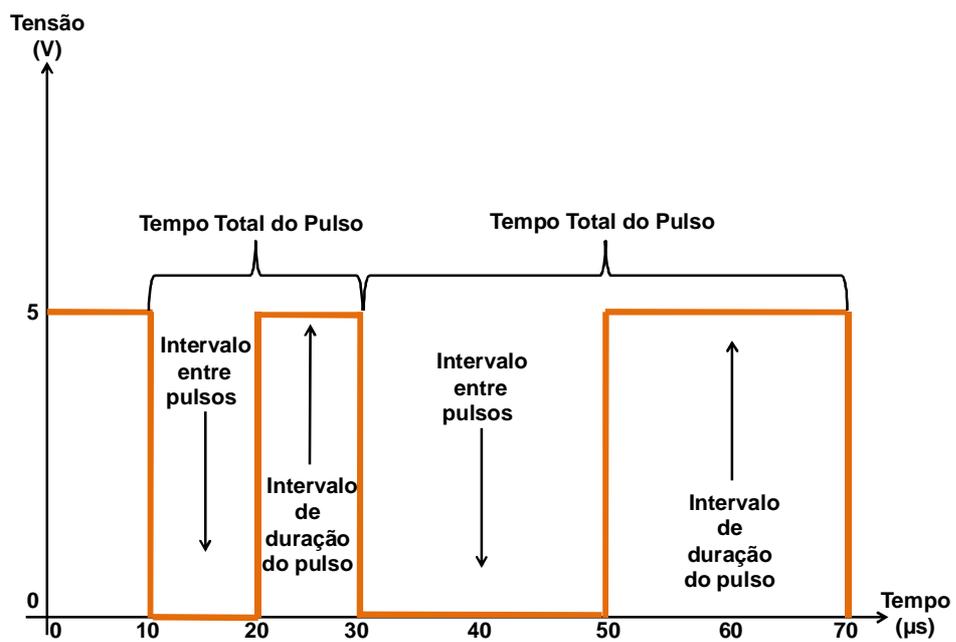


Figura 10 – Tempo Total, Intervalo entre Pulsos e Intervalo de Duração do Pulso

4 DESENVOLVIMENTO DO PROGRAMA

Para implementar o programa proposto utilizou-se o ambiente de desenvolvimento Borland Delphi 6, através do qual é possível fazer a comunicação com a porta paralela do computador para transferência dos dados para controlar os motores de passo. A Figura 11 apresenta um organograma contendo os principais módulos desenvolvidos para o programa.

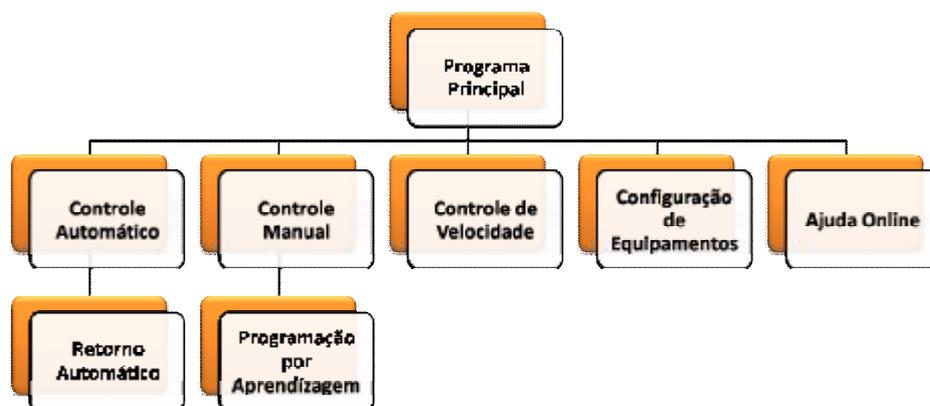


Figura 11 – Estrutura do Programa

A seguir, tem-se a relação completa dos códigos que podem ser usados para criar um arquivo de pulsos e uma descrição detalhada dos módulos que contém o programa.

4.1 Tabela de Códigos do Programa LPTControl

Na Tabela 2, pode-se observar os códigos utilizados para descrever um arquivo de pulsos para o sistema proposto. A coluna Código representa os dados armazenados em um arquivo de pulsos. A coluna Pulso ou Comando descreve a função executada por cada código. Os códigos de “A” até “Z” são utilizados para representar todas as combinações de deslocamentos nos eixos X, Y e Z.

O código “[” serve para enviar o sinal para ligar o equipamento (*drivers* dos motores), enquanto que “]” é utilizado para desligar o mesmo. Desta forma é possível ligar o equipamento no início do processo e desligar ao final ou mesmo ligá-lo e desligá-lo durante o processamento do arquivo de dados ou arquivo de passos, como também é chamado neste trabalho. O símbolo “(“ é utilizado para acionar a ferramenta do equipamento e “)” desliga a ferramenta, com estas funções é possível ligar e desligar a ferramenta quantas vezes se fizer necessário durante uma operação. O “*” oferece a opção de um tempo de espera onde o processamento é interrompido por um determinado intervalo e posteriormente prossegue normalmente. Este intervalo de tempo de espera é definido em milissegundos pelo usuário no arquivo de dados. Por exemplo, usa-se “*5000” para que o processo seja interrompido por um tempo de 5 segundos. Esta opção é interessante em diversas aplicações, como no exemplo de uma operação de oxicorte onde esta função pode ser utilizada para manter o processamento interrompido enquanto aguarda pelo aquecimento da superfície a ser cortada. O código “+” pode ser utilizado pelo usuário para configurar uma determinada velocidade de avanço para o equipamento, este valor é aplicado nos deslocamentos em quaisquer eixos. Para utilizar esta função basta digitar “+2000”, por exemplo, para uma velocidade de dois mil milímetros por minuto. O comando “+0” na listagem do arquivo provoca uma interrupção no processamento. Nesse caso, será necessária a intervenção do usuário para ajustar a velocidade e dar continuidade ao processo. Caso seja especificado um valor maior que a velocidade máxima do equipamento, o valor é substituído automaticamente pelo programa pelo valor da velocidade máxima da janela de configurações do sistema. O código “\$” é o comando utilizado para ajustar a velocidade na máxima possível para o equipamento. Esta opção é útil para efetuar deslocamentos quando a ferramenta encontra-se desligada e deseja-se apenas fazer o seu reposicionamento. O “@” ajusta a velocidade do equipamento para uma velocidade definida pelo usuário como padrão na janela de configurações do sistema. Isto é particularmente interessante em equipamentos onde a maioria dos processos é executada em uma mesma velocidade, ou quando o programador desconhece essa informação.

Tabela 2 – Tabela de Códigos do LPTControl

Código	Pulso ou Comando
A	X+
B	X-
C	Y+
D	Y-
E	Z+
F	Z-
G	X+Y+
H	X+Y-
I	X-Y+
J	X-Y-
K	X+Z+
L	X+Z-
M	X-Z+
N	X-Z-
O	Y+Z+
P	Y+Z-
Q	Y-Z+
R	Y-Z-
S	X+Y+Z+
T	X+Y+Z-
U	X+Y-Z+
V	X+Y-Z-
W	X-Y+Z+
X	X-Y+Z-
Y	X-Y-Z+
Z	X-Y-Z-
[Liga <i>Drive</i>
]	Desliga <i>Drive</i>
(Liga Ferramenta
)	Desliga Ferramenta
*	Tempo de Espera
+	Ajuste de Velocidade
\$	Velocidade Máxima
@	Velocidade Padrão

4.2 Interface Usuário/Programa

O programa LPTControl possui uma interface bastante simples e fácil de usar, o que proporciona uma interação bastante intuitiva por parte do usuário durante a execução dos comandos. Além disso, o programa conta com uma ajuda *online* que poderá ser utilizada sempre que o usuário estiver em dúvida sobre o preenchimento de algum campo ou mesmo para obter informações adicionais para um item do programa.

A janela principal (Figura 12) está dividida em: Barra de Menus, Barra de Ferramentas, Controle de Execução, Controle Manual e Controle de Velocidade. Esta organização da janela principal permite um acesso facilitado à maioria das funções, tornando o programa mais amigável, pois a grande parte das ações pode ser executada através de um único clique do mouse. A seguir serão detalhadas as funcionalidades de cada um dos componentes desta janela.

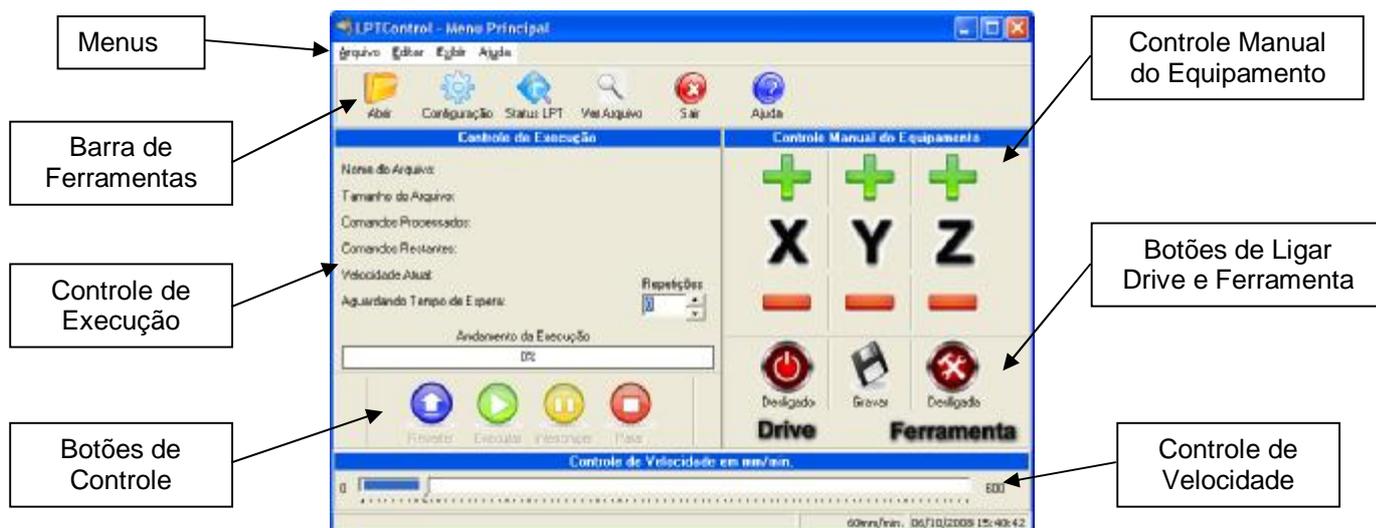


Figura 12 – Janela Principal

4.2.1 Estrutura da Barra de Menus e da Barra de Ferramentas

Como é mostrado na Figura 13, a estrutura da barra de menus e da barra de ferramentas está disposta em quatro menus subdivididos em sub-menus e uma barra de ferramentas.

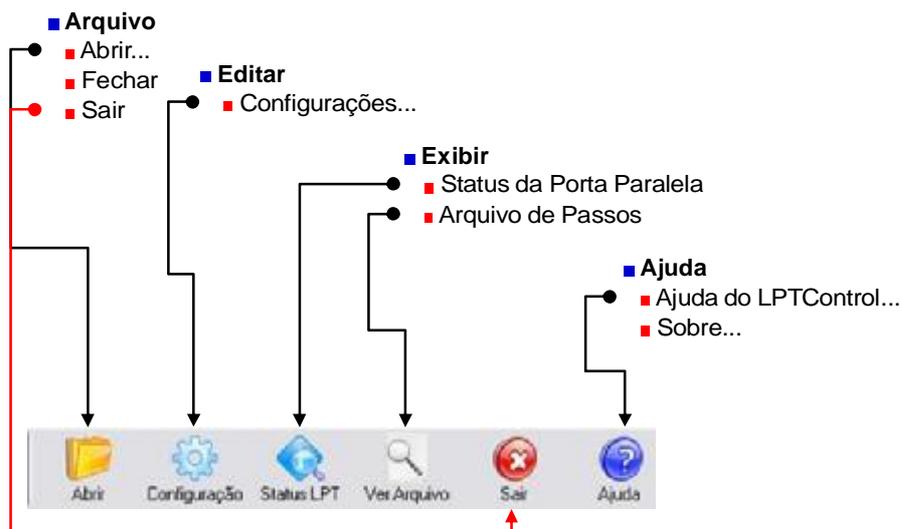


Figura 13 – Barra de Menus e Barra de Ferramentas

O menu **Arquivo** contém três opções, conforme apresentado na Figura 14: **Abrir**, **Fechar** e **Sair**. A opção **Abrir** permite selecionar um arquivo contendo uma seqüência de pulsos para ser executada pelo programa, utilizando os controles para execução automática. Através de **Fechar** pode-se fechar o arquivo de pulsos anteriormente aberto, permitindo a edição do mesmo através de outros programas, caso seja necessário. **Sair** serve para encerrar o programa. Tanto **Abrir** quanto **Sair** está disponível também na Barra de Ferramentas.



Figura 14 – Menu Arquivo e Barra de Ferramentas

A Figura 15 apresenta o menu **Editar**, que oferece a opção **Configurações**, a qual permite alterar as configurações do programa de acordo com o equipamento que estiver sendo operado. Na Barra de Ferramentas encontra-se um botão para acessar a mesma opção.



Figura 15 – Menu Editar e Barra de Ferramentas

Selecionando-se a opção **Configurações**, será exibida a janela de configurações do sistema, como ilustrado na Figura 16.



Figura 16 – Menu Editar Configurações

A tela de Configurações do Sistema divide-se em quatro partes, sendo estas, Barra de Ferramentas, Painel de Configurações Gerais, Painel de Configuração da Porta Paralela e Painel de Configuração dos Motores de passo. A seguir, encontram-se descritos de forma detalhada cada um dos componentes desta tela.

O painel de Configurações Gerais é composto pelos campos código, descrição, intervalo de duração do pulso e número de eixos. O campo Código é utilizado pelo programa para identificar, de forma única, cada configuração, já que o sistema suporta armazenar configurações específicas para vários equipamentos. Este campo é preenchido automaticamente pelo programa a cada vez que é criada uma nova configuração, não sendo permitida a entrada de dados pelo usuário. Descrição é o nome pelo qual uma determinada configuração é identificada pelo usuário. Serve para relacionar a configuração ao nome do equipamento.

O Intervalo de Duração do Pulso é o tempo de que o *drive* do motor de passo necessita entre um pulso e outro. Este intervalo é definido em microssegundos e pode variar de 1 a 250.000. A definição deste parâmetro depende de características dinâmicas dos *drives* dos motores. Um intervalo muito curto pode ocasionar a perda de pulsos, ou seja, o sistema pode gerar pulsos em um intervalo de tempo tão curto que os *drives* não consigam acompanhar. Isto pode gerar falhas no processo

controlado pelo programa. Por outro lado, intervalo de tempo excessivo pode ocasionar velocidade inadequada na execução de uma tarefa. Porém, permite um controle de deslocamento mais preciso. Esta característica diverge da maioria dos programas acadêmicos e até mesmo alguns profissionais produzidos para esta finalidade, que trabalham em milissegundos. O Número de Eixos é a quantidade de eixos utilizada pelo equipamento. O programa aceita trabalhar com máquinas compostas de 1, 2 ou 3 eixos, e esta informação deve ser especificada neste campo.

No painel de Configurações da Porta Paralela encontram-se os campos Sinal, Habilitado, Função e Invertido. A coluna Sinal serve para identificar cada um dos bits da porta paralela. Para cada um dos 8 sinais de comunicação da porta paralela, identificados com os bits que vão de 0 a 7, é possível configurar as opções habilitado, função e invertido. Na coluna Habilitado pode-se habilitar ou desabilitar cada uma das saídas da porta paralela de acordo com as necessidades do equipamento que está conectado. Desabilitando uma saída o programa não gerará nenhum tipo de sinal na mesma. Esta função pode ser utilizada, por exemplo, para desabilitar a saída responsável pelo acionamento da ferramenta, permitindo assim que se execute um arquivo de dados para um determinado processo percorrendo a trajetória descrita com a ferramenta desligada. Através da coluna Função é possível associar uma função a cada um dos bits de dados da porta paralela. Cada bit deverá ter uma função única associada. As funções disponíveis são oito:

- Direção Eixo X – Controla a direção de deslocamento no eixo X
- Direção Eixo Y – Controla a direção de deslocamento no eixo Y
- Direção Eixo Z – Controla a direção de deslocamento no eixo Z
- Habilita *Drive* – Habilita e desabilita os *drives* dos motores de passo
- Liga Atuador – Liga e desliga o atuador
- Pulso Eixo X – Transmite os pulsos para passos no eixo X
- Pulso Eixo Y – Transmite os pulsos para passos no eixo Y
- Pulso Eixo Z – Transmite os pulsos para passos no eixo Z

Ativando a caixa de verificação Invertido pode-se inverter o nível de sinal de saída, de um determinado bit. Ativando-se essa função, gera-se um sinal de nível lógico zero como saída para o bit selecionado. Caso contrário, ele permanece em nível um. Esta característica é utilizada em *drives* ou equipamentos que utilizam este tipo de lógica de acionamento.

No painel de Configurações dos Motores de Passo têm-se a especificação da resolução do equipamento, ou seja, a distância percorrida a cada pulso sobre um, dois ou três eixos. Enquanto as distâncias sob os eixos devem ser fornecidas pelo usuário durante a configuração, as diagonais são configuradas automaticamente pelo sistema aplicando-se o teorema de Pitágoras.

O painel de Ajuste de Velocidade conta com dois parâmetros, a Velocidade Máxima e a Velocidade Padrão. A Velocidade Máxima é calculada pelo programa de forma automática, levando-se em consideração o tamanho dos passos sobre cada eixo e o intervalo mínimo entre um pulso e outro. A Velocidade Padrão é definida pelo usuário como a velocidade utilizada com maior frequência no equipamento, sendo, aplicada em movimentos manuais, ou quando o arquivo de pulsos não especifica um valor em ambos os casos é possível alterar o valor da velocidade através do controle de velocidade.

Clicando no botão “Nova” será criado um novo registro no banco de dados de configurações que permitirá ao usuário editar e gravar uma nova configuração. O usuário pode adicionar quantas configurações desejar ao sistema, permitindo que o mesmo faça uma configuração para cada equipamento em que o programa for aplicado ou mesmo armazenar múltiplos ajustes para um mesmo equipamento. Para escolher uma entre as configurações disponíveis para o programa, ativa-se o botão “Selecionar”, conforme exemplificado na Figura 17. Ao selecionar uma configuração, esta será exibida na janela de configurações conforme ilustrado anteriormente na Figura 16.

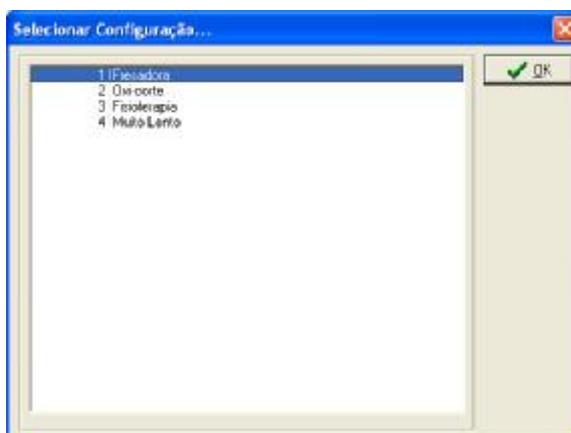


Figura 17 – Selecionar uma Configuração

A opção “Salvar” grava a configuração com os dados já especificados. O botão “Excluir” permite eliminar do cadastro uma configuração existente ou cancelar a inserção de uma nova configuração ainda não gravada. Através do botão “Fechar” retorna-se ao menu principal do programa.

O menu Exibir está dividido em dois submenus (Figura 18). O primeiro, para visualizar o Status da Porta Paralela e o segundo, para exibir o Arquivo de Passos. Têm-se as mesmas opções na barra de ferramentas.



Figura 18 – Menu Exibir e Barra de Ferramentas

Ao acionar **Status da Porta Paralela**, ou o menu exibir ou **Status LPT** no atalho da barra de ferramentas, são mostrados os sinais que estão sendo enviados pelo programa para a porta paralela, conforme exemplificado na Figura 19. No quadro (a) da Figura 19 têm-se os sinais enviados para deslocamento positivo em um único eixo e no quadro (b) têm-se os sinais utilizados para deslocamento simultâneo em dois eixos sendo um no sentido positivo e o outro no negativo, para a configuração definida na Figura 17.

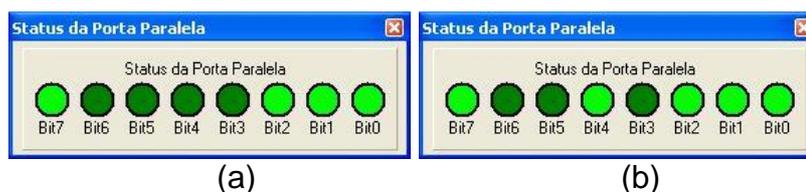


Figura 19 – Status da Porta Paralela

Clicando-se em **Arquivo de Passos** no menu exibir, ou **Ver Arquivo** no atalho da barra de ferramentas, é possível visualizar o arquivo que está aberto. Em operação automática, o sistema indica o ponto em que o mesmo está sendo executado (Figura 20). Este recurso é muito útil para uma possível depuração, pois, facilita a busca por erros no arquivo de passos.

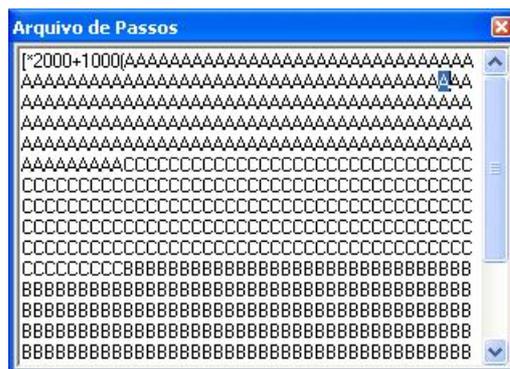


Figura 20 – Arquivo de Passos

O menu **Ajuda**, também disponível na barra de Ferramentas, tem dois submenus, como ilustrado na Figura 21, **Ajuda do LPTControl** e **Sobre**.



Figura 21 – Menu Ajuda e Barra de Ferramentas

Clicando em **Ajuda do LPTControl** no menu ou clicando em **Ajuda** na barra de ferramentas abrirá a ajuda *online* do sistema, a qual fornece uma descrição completa a respeito de todas as opções disponíveis no sistema. A opção **Sobre** abre uma janela que contém informações sobre o programa, tais como: versão, autor e uma breve descrição. Na Figura 22 é apresentada uma das várias telas da Ajuda *Online* do sistema.



Figura 22 – Tela de Ajuda *Online* do Sistema

4.2.2 Controle Automático

Na janela principal do programa (Figura 23), tem-se o controle de execução, onde encontram-se os seguintes itens:

Nome do Arquivo: Nome do arquivo que foi aberto para execução;

Tamanho do Arquivo: Tamanho do arquivo a ser executado;

Comandos Processados: Comandos processados até o momento;

Comandos Restantes: Comandos que restam para ser executados;

Velocidade Atual: Velocidade de avanço da ferramenta em mm/min;

Aguardando Tempo de Espera: Contagem regressiva do tempo durante o qual a execução do arquivo é interrompida.

Repetições: Permite que seja selecionado o número de vezes que o arquivo de pulsos será processado. O sistema admite entre 1 e 100 repetições. Esta característica é particularmente interessante para produção de lotes de uma mesma peça, por exemplo. Para usar esse recurso em uma mesma posição ou posições diferentes basta descrever juntamente no arquivo de comandos uma trajetória de deslocamento.

Andamento da Execução: Exibe através de uma barra de progresso o percentual já processado do arquivo.

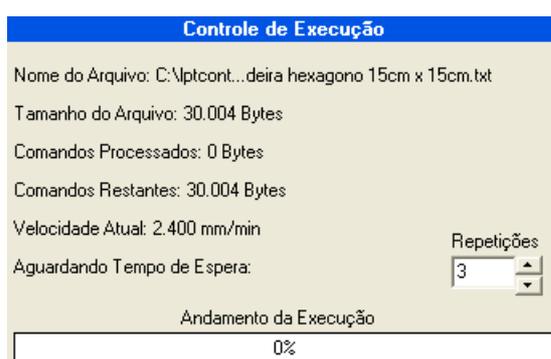


Figura 23 – Controle de Execução

Logo abaixo do controle de execução, têm-se os botões para controlar o processamento do arquivo de dados (Figura 24). O botão **Executar** processa o arquivo de dados. Acionando o botão **Reverter**, o arquivo de dados será executado na ordem inversa. Esta característica permite que a ferramenta retorne pela mesma

trajetória já percorrida. Isto torna-se particularmente interessante em processos como o oxicorte, onde pode haver uma extinção da chama no meio do processo. Com esse recurso, é possível retomar a operação a partir da posição em que a chama foi extinta. Em geometrias com trajetórias irregulares, a opção pode evitar a perda da peça em processamento. Para aplicações do sistema em processos de usinagem, esse recurso pode ser explorado para dar acabamento ou optar entre as alternativas de fresamento concordante e discordante (Costa e Santos, 2006).



Figura 24 – Opções de Processamento do Arquivo de Pulsos

A opção **Interromper** para temporariamente o processamento do arquivo de dados e aguarda alguma ação do operador, enquanto o no botão **Parar** aborta um processo.

Abaixo desses botões está o **Controle de Velocidade** (Figura 25), que permite alterar a velocidade de avanço da ferramenta, definida pela velocidade padrão no acionamento manual ou especificada no programa CNC.



Figura 25 – Controle de Velocidade

O sistema possui um recurso para retorno automático a partir de um deslocamento manual da ferramenta realizado pelo usuário após uma interrupção da execução de um arquivo de dados de controle do equipamento. A Figura 26 exemplifica o efeito desse recurso. O quadro (a) mostra o caso em que após a interrupção do programa e subsequente movimento manual, a ferramenta retorna automaticamente à posição onde o trabalho foi interrompido e a partir dali continua a tarefa. De outro modo, como apresentado no quadro (b), o usuário não utiliza o recurso de retorno automático e então a execução do arquivo prossegue a partir do ponto onde a ferramenta foi reposicionada manualmente pelo usuário.

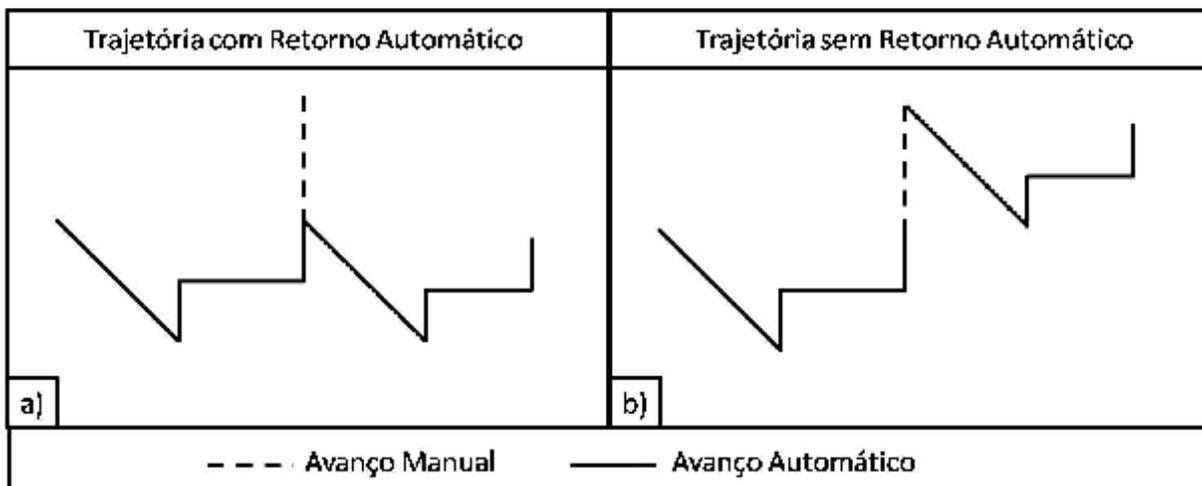


Figura 26 – Controle Automático

O usuário pode optar entre utilizar ou não o recurso de retorno automático através da tela de confirmação exibida na Figura 27.

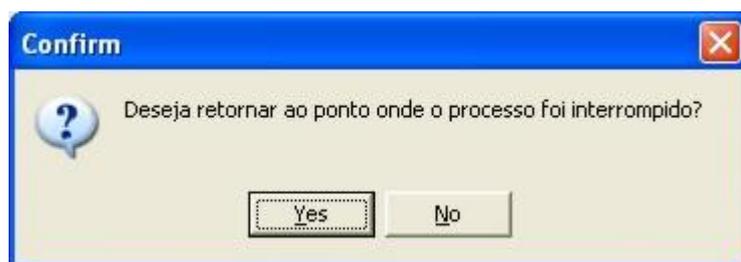


Figura 27 – Confirmação de Retorno Automático

4.2.3 Controle Manual

No lado direito do controle de execução tem-se o campo **Controle Manual** (Figura 28), que serve para controlar cada um dos eixos do equipamento individualmente.

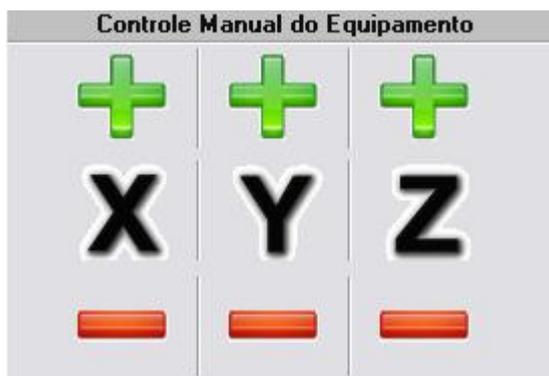


Figura 28 – Controle Manual

As letras **X**, **Y** e **Z** servem para fazer o controle manual dos eixos de movimentos do sistema. Ao clicar no sinal de mais + (na cor verde) ou - (na cor vermelha) de um dos eixos, é produzido um deslocamento na direção convencionalizada como positiva ou negativa deste eixo, respectivamente. O mesmo efeito pode ser produzido utilizando-se as setas de direção do teclado e as teclas “Page Up” e “Page Down”.

Tem-se ainda no controle manual, logo abaixo dos botões para operação dos motores, o botão para ligar e desligar o *drive* e a ferramenta, cuja condição é identificada pelas cores verde (Figura 29a) e vermelho (Figura 29b), respectivamente. A mesma convenção é adotada para o controle de acionar ou desligar a ferramenta.



Figura 29 – Simbologia para Controle Manual de *Drive* e Ferramenta

Entre os botões para ligar e desligar o *drive* e a ferramenta tem-se o botão **Gravar** (Figura 29), que serve para armazenar os passos que o usuário dará manualmente em um arquivo para um posterior uso. Esta técnica é conhecida como

programação por aprendizagem, descrita anteriormente. A gravação dos comandos manuais do usuário inicia-se assim que for pressionado o botão gravar, a partir deste momento todos os comandos são armazenados em um registro temporário. Para finalizar a gravação basta acionar novamente no botão gravar. Neste momento, o sistema solicita ao usuário que digite um nome de arquivo para salvar os registros temporários armazenados.

5 RESULTADOS

Este capítulo descreve os resultados obtidos nos testes efetuados com o programa desenvolvido neste trabalho. Os testes compreenderam a utilização do programa em modo automático e manual, em situações reais e simuladas.

Na Tabela 3, tem-se o resultado simulado da execução de um arquivo de dados de controle do equipamento. Os dados são processados e os sinais correspondentes são enviados a porta paralela.

Tabela 3 – Simulação do Resultado da Execução de um Arquivo

Arquivo de Dados de Controle		Saídas da Porta Paralela								
Código	Pulso ou Comando	Saída	HD	PZ	DZ	PY	DY	PX	DX	LF
[Liga <i>Drive</i>	80H	1	0	0	0	0	0	0	0
+	Ajuste de Velocidade	-	-	-	-	-	-	-	-	-
1		-	-	-	-	-	-	-	-	-
0		-	-	-	-	-	-	-	-	-
0		-	-	-	-	-	-	-	-	-
0		-	-	-	-	-	-	-	-	-
A	X+	86	1	0	0	0	0	1	1	0
A	X+	86	1	0	0	0	0	1	1	0
B	X-	84	1	0	0	0	0	1	0	0
X	X-Y+Z-	DC	1	1	0	1	1	1	0	0
P	Y+Z-	D8	1	1	0	1	1	0	0	0
(Liga Ferramenta	81	1	0	0	0	0	0	0	1
Q	Y-Z+	F1	1	1	1	1	0	0	0	1
R	Y-Z-	D1	1	1	0	1	0	0	0	1
M	X-Z+	E5	1	1	1	0	0	1	0	1
Y	X-Y-Z+	F5	1	1	1	1	0	1	0	1
Z	X-Y-Z-	D5	1	1	0	1	0	1	0	1
)	Desliga Ferramenta	80	1	0	0	0	0	0	0	0
]	Desliga <i>Drive</i>	00	0	0	0	0	0	0	0	0

Na coluna **Código** têm-se os comandos que se encontram descritos no arquivo de dados do controle. A coluna **Pulso ou Comando** contém a descrição referente a ação executada pelo código. A coluna **Saída** traz a palavra em hexadecimal gerada na porta paralela. As demais colunas: Habilita *Drive* (**HD**), Passo no Eixo Z (**PZ**), Direção no Eixo Z (**DZ**), Passo no Eixo Y (**PY**), Direção no

Eixo Y(DY), Passo no Eixo X (PX), Direção no Eixo X (DX), Liga Ferramenta (LF) correspondem aos sinais da porta paralela.

5.1 Teste do Modo Automático

Na Figura 30 tem-se duas telas do programa LPTControl. Na tela principal é possível verificar, no controle de execução, informações a respeito do arquivo de comandos que está sendo executado. Dentre estas informações encontra-se destacada a Velocidade Atual. É possível acompanhar na janela de passos que o próximo comando a ser executado altera esta velocidade para 1.000 mm/min.



Figura 30 – Velocidade Atual de 1.200 mm/min.

Quando é executado o comando para ajuste de velocidade (Figura 31) então a velocidade passa para 1.000,2 mm/min. Esta discrepância entre a velocidade solicitada pelo comando e a velocidade aplicada se deve à necessidade de ajustar a velocidade à resolução do equipamento.



Figura 31 – Velocidade Atual de 1.000 mm/min.

Na Figura 32 a execução já se encontra em um ponto mais adiantado onde será solicitada pelo arquivo de comandos uma pausa na execução. Na janela do Arquivo de Passos é possível visualizar que o próximo comando a ser executado é *10000, que indica uma pausa de 10 segundos na execução.



Figura 32 – Tempo de Espera

Após a execução do comando que aciona o tempo de espera, é possível visualizar a contagem regressiva em andamento (Figura 33). Neste ponto, a execução permanece interrompida por 10 segundos e a contagem regressiva vai sendo atualizada na tela até que atinja zero. Ao zerar a contagem, o programa prossegue com a execução normal do arquivo.



Figura 33 – Contagem Regressiva

Na Figura 34 o arquivo de passos está para executar o comando \$, que ajusta a velocidade para a máxima possível para o equipamento. No momento, a velocidade do equipamento indicada no controle de execução é de 1.000,2 mm/min., conforme ajustada anteriormente pelo sistema.



Figura 34 – Comando de Velocidade Máxima

Com a execução do comando \$, a velocidade é ajustada para 2.400 mm/min. (Figura 35), que é a máxima especificada para o equipamento utilizado no teste.



Figura 35 – Velocidade Atual de 2.400 mm/min.

A seguir, na Figura 36, o sistema encontra-se executando a uma velocidade de 2.400 mm/min., tendo processado 50% do arquivo de comandos. O comando seguinte a ser processado, conforme pode ser visto na janela do arquivo de passos, é o @ que ajusta a velocidade para o padrão do equipamento, que estava sendo empregada no início do processo.



Figura 36 – Velocidade Padrão

Com a execução do comando @, é possível visualizar na Figura 37 que a velocidade foi restabelecida a 1.200mm/min., conforme estipulado pelo usuário como velocidade padrão para o equipamento.



Figura 37 – Velocidade Padrão Restabelecida

Durante o teste, o programa LPTControl conseguiu executar todas as instruções contidas no arquivo, incluindo-se, os ajustes de velocidade e o tempo de espera. Testes similares a este foram desenvolvidos em várias plataformas diferentes de hardware e software para verificar a compatibilidade do sistema proposto com as diferentes versões do sistema operacional Windows e especificações de hardware. Os sistemas operacionais utilizados nos testes foram o Microsoft Windows Vista 64bits e 32bits, Microsoft Windows XP e Microsoft Windows 98. As plataformas de hardware utilizadas incluíram sistemas baseados nos processadores Intel Core2Duo, Intel Pentium IV, Intel Celeron, AMD Athlon 64 X2 e AMD K6 com quantidades de memória variando entre 128MB e 2GB. Não foram verificadas incompatibilidades entre o programa desenvolvido neste trabalho e as tecnologias de hardware e software utilizadas nos testes.

5.2 Testes Práticos

Para analisar o sistema em aplicações práticas, foram realizados testes em uma empresa fabricante de máquinas CNC (Tecnopampa, 2009). O trabalho desenvolvido foi aplicado em uma fresadora projetada para corte em madeira, conforme mostra a Figura 38.



Figura 38 – Máquina Router CNC Industrial 3 Eixos (Tecnopampa, 2009)

A Figura 39 mostra a configuração utilizada no programa LPTControl para realização dos testes na máquina Router CNC Industrial 3 Eixos da Empresa Tecnopampa.

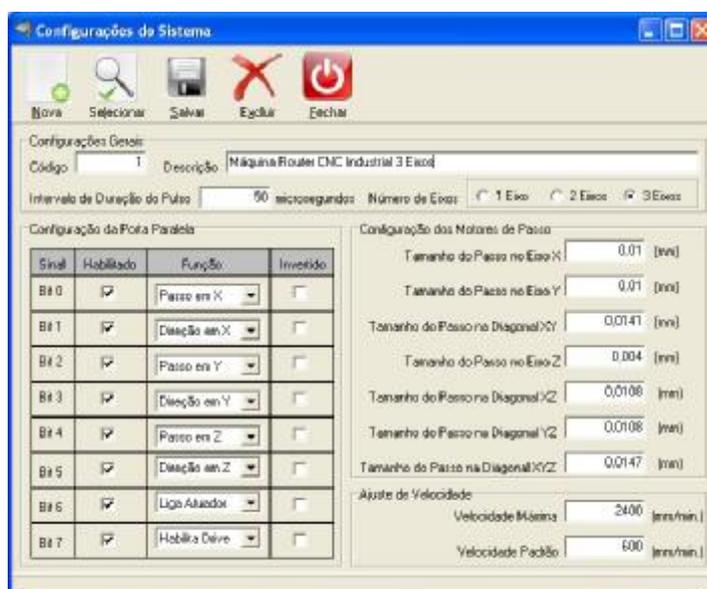


Figura 39 – Configuração da Máquina Router CNC Industrial 3 Eixos

Para avaliar o desempenho do sistema, foram desenvolvidas aplicações de corte em chapa de MDF, produzindo-se geometrias variadas para avaliar o sincronismo dos motores X e Y. Na Figura 40 são apresentados exemplos de cortes feitos pela máquina na qual foram realizados os testes.



Figura 40 – Cortes Realizados pela Máquina Router CNC Industrial 3 Eixos

Em todos os testes realizados foram criadas geometrias que possuem linhas retas sobre os eixos X e Y.

5.2.1 Teste 1 – Geometria de um Quadrado

No primeiro teste realizado, Figura 41, foi criada a geometria de um quadrado. O corte obedece a uma seqüência de passos descrita no arquivo de dados.

Inicialmente a ferramenta foi posicionada, através de acionamentos nos eixos X e Y, no local onde se deu início ao corte. Na seqüência, a ferramenta foi ligada e deslocada no eixo Z até atingir a profundidade desejada. Este posicionamento inicial foi feito através do controle manual do programa LPTControl. Após, foi utilizado um arquivo de dados que continha as instruções necessárias para que o programa manipulasse a máquina nos eixos X e Y de forma a produzir o corte de um quadrado de tamanho 10 cm por 10 cm.

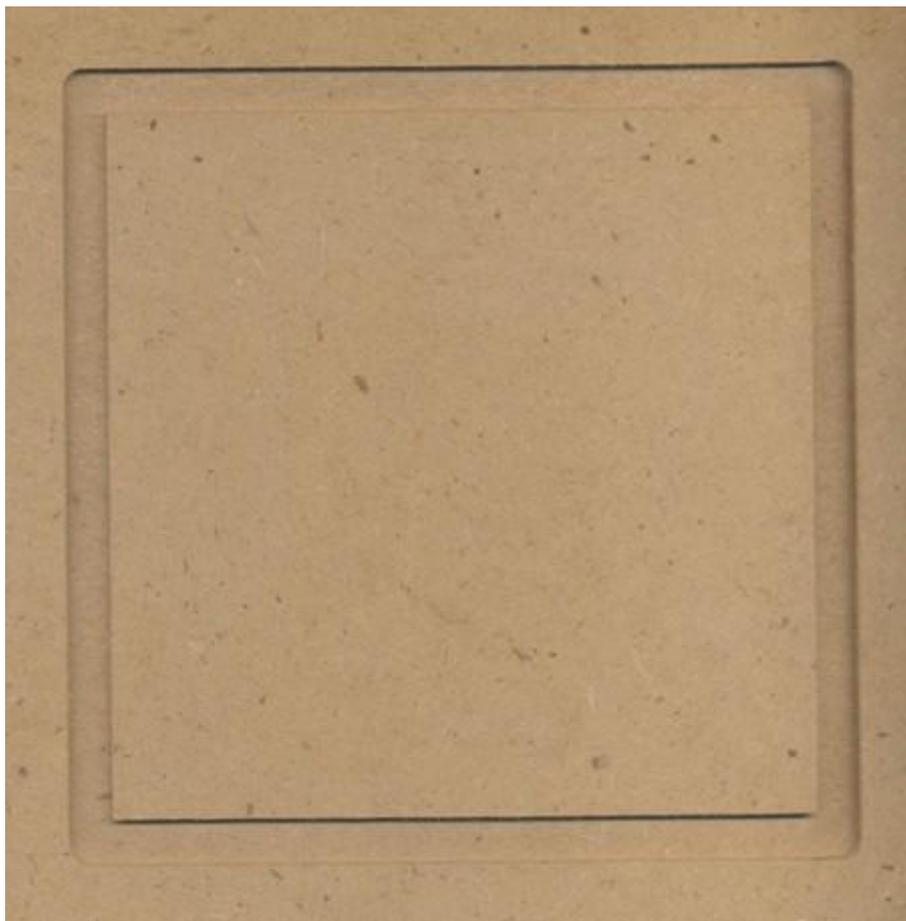


Figura 41 – Corte da Geometria de um Quadrado

5.2.2 Teste 2 – Geometria de um Triângulo

No segundo teste, foi criada a geometria de um triângulo, como pode-se observar na Figura 42. Este corte foi realizado para conferir a interpretação de códigos com acionamento simultâneo dos eixos e para analisar a manutenção da velocidade de avanço, quando a alteração na direção do movimento. O deslocamento na diagonal mostrou ter a mesma velocidade desenvolvida com o deslocamento em um único eixo. Embora a velocidade real não tenha sido medida, pode-se afirmar que o tempo entre pulso, para acionamento simultâneo dos eixos, foi corrigido proporcionalmente ao aumento na distância percorrida segundo o teorema de Pitágoras, porque a não aplicação desse recurso provocaria alteração facilmente perceptível. A geometria apresentada mede 10 cm por 10 cm de lados com uma diagonal de 14,14 cm.



Figura 42 – Corte da Geometria de um Triângulo

5.2.3 Teste 3 – Geometria de um Hexágono

Para análise dos códigos de acionamento simultâneo dos eixos em todas as combinações possíveis com os eixos X e Y, foi feito o corte da geometria de um hexágono medindo 15 cm por 10 cm, como está ilustrado na Figura 43. Neste teste, também foi verificada a execução de um arquivo de dados com interrupção do processo (Figura 44), deslocamento manual da ferramenta para uma posição fora da geometria de corte e posterior retorno automático da ferramenta para a posição onde o processo foi interrompido, para prosseguimento do corte.



Figura 43 – Corte da Geometria de um Hexágono



Figura 44 – Interrupção no Corte da Geometria de um Hexágono

5.2.4 Teste 4 – Geometria de um Octógono

Neste teste realizado criou-se a geometria de um octógono medindo 15 cm por 15 cm, ilustrado na Figura 45.

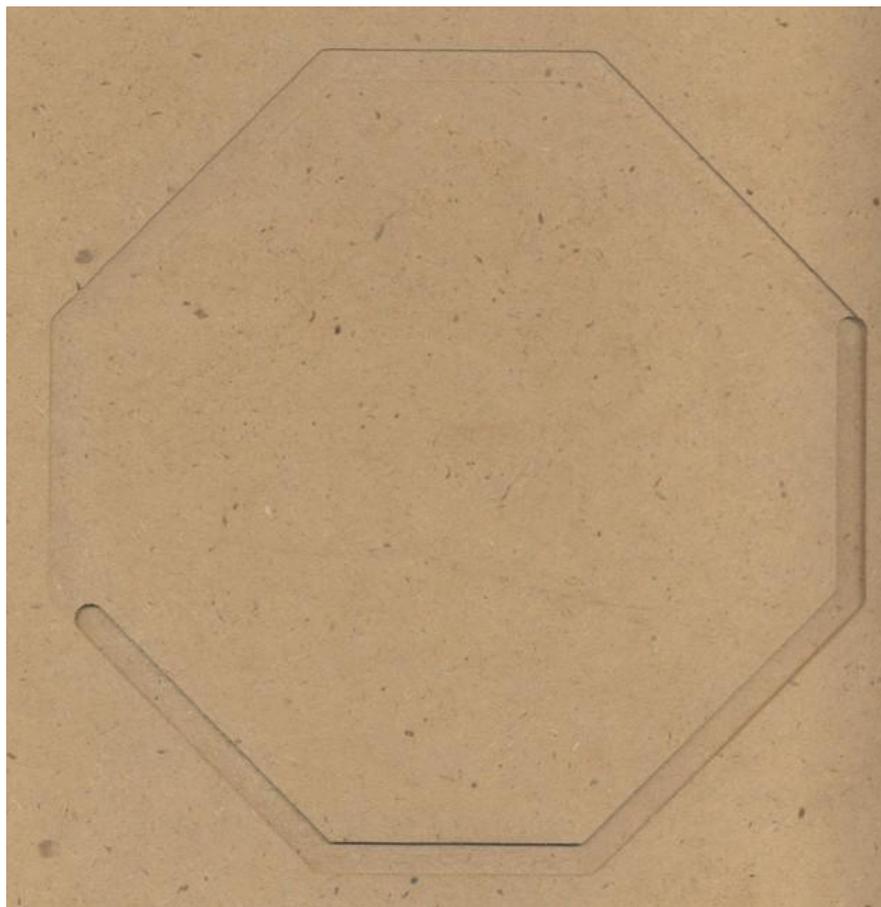


Figura 45 – Corte da Geometria de um Octógono

O corte iniciou-se pela diagonal inferior direita e quando a ferramenta encontrava-se na base da figura, interrompeu-se o processo (Figura 46) para utilizar o recurso de execução inversa (Figura 47) com o objetivo de retornar até um ponto próximo da intersecção da diagonal superior esquerda com o topo da figura. Neste ponto utilizou-se o módulo de controle manual do programa para aprofundar mais a cavidade e então a trajetória foi percorrida novamente a partir daquele ponto até o final. Na Figura 48 pode-se ver uma amostra do uso do recurso de execução inversa.



Figura 46 – Interrupção no Corte

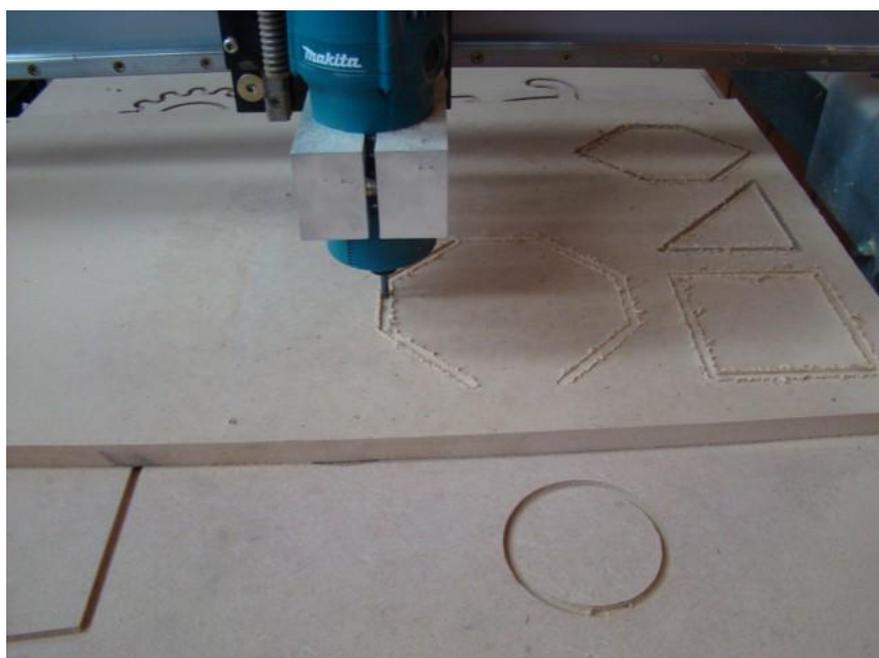


Figura 47 – Execução Inversa



Figura 48 – Corte Utilizando a Execução Inversa

Este recurso foi utilizado para aprofundar a marcação anterior, a qual havia ficado muito superficial, o que tornava difícil a visualização da geometria na placa de MDF. É possível perceber nitidamente no centro da figura o ponto em que a marcação foi aprofundada.

5.3 Comparativo com o Software Comercial KCam4

Na Tabela 4, tem-se uma comparação entre o software comercial KCam4 e o programa LPTControl. Entre as características avaliadas pode-se perceber algumas diferenças entre os dois programas.

Tabela 4 – Tabela Comparativa KCam4 x LPTControl

Característica	KCam4	LPTControl
Número de eixos	4	3
Status da porta paralela	Sim	Sim
Avanço automático	Sim	Sim
Retrocesso automático	Não	Sim
Pausa	Sim	Sim
Interrupção	Sim	Sim
Retorno automático após deslocamento manual	Não	Sim
Programação por aprendizagem	Não	Sim
Configurar múltiplos equipamentos	Sim	Sim
Sistema de posicionamento	Por coordenadas	Por pulsos
Controle de velocidade em tempo real	Não	Sim

Por se tratar de um software comercial e que se encontra em sua quarta versão o KCam4 possui algumas características interessantes como controle de até quatro eixos e processamento de arquivos GCode. Por outro lado o programa desenvolvido neste trabalho conta com características avançadas de processamento como retrocesso automático, retorno automático, programação por aprendizagem e controle de velocidade em tempo real, as quais não se encontram presentes no KCam4.

A programação por aprendizagem é uma característica particularmente interessante, pois, permite que o usuário realize uma série de comandos manualmente e os armazene na forma de um arquivo de pulsos, o qual consistirá numa seqüência exata de pulsos necessária para repetir de forma automática a operação manual realizada pelo usuário.

Estas funcionalidades extras do LPTControl tornam o seu uso interessante em vários processos onde tais características podem representar vantagens competitivas para os usuários.

6 CONCLUSÕES

A grande competitividade entre as empresas em busca da modernização dos equipamentos, diversificação da produção e a redução de custos e também pela exigência dos consumidores, por produtos variados e de qualidade as leva a busca de novos equipamentos para manterem-se no mercado atual. Pequenas e médias empresas procuram adaptar-se a essa situação usando tecnologias de baixo custo para desenvolver ou adequar equipamentos já existentes. Para atender ao objetivo geral proposto deste trabalho de apresentar uma metodologia de controle de motores de passo em aplicações de automação, primeiramente fez-se um estudo de diversas publicações a respeito de aplicações deste tipo de motor no desenvolvimento de equipamentos CNC.

O programa desenvolvido para demonstrar a metodologia realiza as tarefas que foram propostas nos objetivos e que se limitou ao desenvolvimento de uma metodologia de software diferenciada para controle de equipamentos de até três eixos programáveis construídos a partir de motores de passo, controlados via porta paralela, para utilização em processos de automação industrial. Esta metodologia interpreta dados pré-processados de programas CNC. Todos os requisitos do programa foram integralmente atendidos na versão final.

A técnica, implementada na linguagem Borland Delphi 6, mostra que recursos simples para fazer a comunicação entre a porta paralela de microcomputadores e *drives* de motores de passos, podem ser eficientes para o desenvolvimento de sistemas automáticos dedicados com software embarcado. Entretanto, a aplicação da metodologia proposta, que interpreta diretamente a seqüência de pulsos lida de um arquivo texto, exige implementação de módulos adicionais para operação. Porém, com a alternativa de desenvolvimento próprio de programas computacionais que gerem a seqüência de pulsos desejada, permite que as particularidades de desempenho dos equipamentos possam ser mais precisamente direcionadas.

A interpretação de dados pré-processados reduz a complexidade do programa de controle. Esta característica também indica a aplicação da técnica para o projeto de equipamentos com o desenvolvimento da unidade de comando a partir

de circuitos eletrônicos com microprocessadores. A dificuldade, nesse caso, é implementar em linguagem computacional de baixo nível, técnicas como interpolações lineares ou circulares.

Esta ferramenta poderá ser usada em empresas da área metal-mecânica, principalmente em processos que utilizem mesas de coordenadas para corte, furação, fresamento e torneamento. Também poderá ser usada para fins de pesquisa, pois serve de módulo básico para o desenvolvimento de sistemas mais complexos.

Algumas sugestões que poderiam ser implementadas em trabalhos futuros são:

- Adaptar o sistema LPTControl para ser controlado via porta USB;
- Implementar rotinas de aceleração e desaceleração para equipamentos que possuem uma inércia muito grande ou para processos em que necessita-se diminuir a velocidade para fazer curvas ou cantos formas geométricas;
- Reconhecer entidades geométricas na programação por aprendizagem;
- Fazer a compensação do raio da ferramenta;
- Desenvolver a simulação gráfica para o programa.

REFERÊNCIAS BIBLIOGRÁFICAS

- BANDEIRA, M. A. G. **Desenvolvimento de um Equipamento Programável para Micro e Pequena Empresa.** Dissertação de Mestrado da Universidade Federal de Santa Maria, Curso de Engenharia de Produção, Santa Maria-RS, 2005.
- BARBOZA, F. J. R.; AZEVEDO, J.; OLVEIRA, L. R.; ALBUQUERQUE, M. L. **Uma Metodologia para Construção de Robôs Móveis.** Programa de Pós-Graduação em Mecatrônica, Departamento de Ciência da Computação Departamento de Engenharia Mecânica, Universidade Federal da Bahia, I Seminário Técnico-Científico, I STEC, Salvador, 2003, 7p. Disponível em: <http://www.lucianooliveira.eti.br/publicacoes_files/Ogumbot.pdf>. Acesso em: 24 abr. 2007.
- BRAGA, C. N. **Os Segredos da Porta Paralela.** Revista Mecatrônica Atual, São Paulo-SP, ano 1, n.º 1, nov/2001, 32p.
- BRAGA, N. C. **Eletrônica Básica para Mecatrônica.** 1.ed. São Paulo: Editora Saber, 2005, 160p.
- COSTA, D. D.; PEREIRA, A. G. **Desenvolvimento e Avaliação de uma Tecnologia de Baixo Custo para Programação CNC em Pequenas Empresas.** Revista Produção, Abr 2006, Vol.16, Nº1, p.48-63. Universidade Federal do Paraná, 16p. Disponível em: <http://www.scielo.br/>
- COSTA, E. R. F. **Simulação de Movimento e Planejamento de Trajetórias para Robôs Manipuladores.** Dissertação de Mestrado, Universidade Federal de Pernambuco, Recife-PE, 2003, 74p.
- COSTA, E. S.; SANTOS, D. J. **Processos de Usinagem.** Centro Federal de Educação Tecnológica de Minas Gerais, Curso Técnico em Eletromecânica. Divinópolis, 2006.
- FEDEL, R. **Metodologia de Desenvolvimento de Equipamento para Alimentar Deficientes Físicos.** Dissertação de Mestrado da Universidade Estadual Paulista Júlio Mesquita/Ilha Solteira, Curso de Engenharia Elétrica, São Paulo-SP, 2004, 172p.
- FERREIRA, J.; MARTINS, N.; AGNELO, L.; DIAS, J. **Imagiologia Tridimensional Digital para Construção de Protótipos Industriais.** Coimbra-Portugal, Instituto de Sistemas e Robótica, Universidade de Coimbra, 2001, 11p.
- FILHO, J. B. M. **Controlador Adaptativo Neural para Mesa de Coordenadas X-Y.** Curso de Pós-Graduação em Engenharia Mecânica, CT- UFPB, 2007.
- FORTIN, E. **An Innovative Software Architecture to Improve Information flow from CAM to CNC.** Computers & Industrial Engineering, Vol. 46, 2004, p. 655-667.

- GOELLNER, E. **Ferramenta Computacional para Acionamento de Motores de Passo Aplicados ao Projeto de Equipamentos CNC**. Dissertação de Mestrado, Santa Maria-RS, 2006, 130p.
- ISO. **International Organization for Standardization**. Disponível em: <http://www.iso.org>. Acesso em: 31 mai. 2008.
- KAIKKONEN, J.; MAKELAINEN, T.; HAKALA, H. **Advanced Design Methods and Tools for Mobile Robot Development, Intelligenc for Mechanical Systems**. Proceedings IROS, 1991, 7p.
- KELLYWARE. **Manual KCAM 4 - CNC CONTROL SOFTWARE**. 2007, 31p.
- KORTENKAMP, D.; BONASSO, R. P.; MURPHY, R. **Artificial Intelligence and Mobile Robots: case studies of successful robot systems**, The MIT Press, 1998.
- LOPES, L. C. G. **Programando para Controle de Dispositivos pelo Computador**. Informática Industrial/Automação, SDM - Sistemas Digitais e Microprocessados CEFET-MG Campus III - Uned Leopoldina, 2007, 36p.
- LYNCH, M. **Computer Numerical Control: Acessory Devices**. New York: McGraw-Hill, 1994, 262p.
- LYNCH, M. **The Key Concepts of CNC**. Modern Machine Shop, Cincinnati, Vol. 69, Nº 11A, 1997, p.81-144.
- MACH3. **Manual de Utilização do Software Mach3**. 2005, 43p.
- MACHADO, A. **Comando Numérico Aplicado às Máquinas-Ferramenta**. 4ª ed., São Paulo, Editora Ícone, 1990, 312 p.
- NBR. **Associação Brasileira de Normas Técnicas**. Disponível em: <http://www.abnt.org.br>. Acesso em: 31 mai. 2008.
- PAZOS, F. A.; LOVISOLO, L. **Automação de Sistemas e Robótica**. 1 ed. Rio de Janeiro: Editora Axcel Books do Brasil, 2002, 384p.
- PENTEADO, F. A. C. A. **Processo de Usinagem dos Metais**. Revista CADware(r) Publishing Brazil. Agosto/2002.
- PERACETTA, L. F.; UNIANDRADE, A. R. **Sistema de Aquisição de Dados e Controle de Processos Através da Porta Paralela**. Simpósio, 2003, 10p. Disponível em: <http://www.uniandrade.br/simpósio/pdf/comp105.pdf>. Acesso em: 23 set. 2008.
- PEREIRA, A. G. **Desenvolvimento e Avaliação de um Editor para Programação CN em Centros de Usinagem**. Dissertação de Mestrado, Curitiba-PR, 2003, 122p.

- POLL, M. T. **Uma Metodologia para Automação do Processo de Conformação por Calandras**. Dissertação de Mestrado da Universidade Federal de Santa Maria, Curso de Engenharia de Produção, Santa Maria-RS, 2008, 129p.
- POSTAL, M. **Desenvolvimento e Implementação de CNC para Motores de Passo**. Dissertação de Mestrado da Universidade de Uberlândia, Curso de Engenharia Mecânica, Uberlândia-MG, 2000.
- RAMALHO, J. **Corte Plasma x Oxicorte e Algumas Considerações**. Anais do XX Congresso Brasileiro de Soldagem, 1994. São Paulo: ABS, p.125-138.
- RANK, A.; BATTI, C. B. **Usinagem de Madeira em Máquinas CNC**. Porto Alegre-RS, SENAI - Projeto de Ampliação de Produtos em Informação Tecnológica nas Áreas Madeira e Mobiliário, 2007, 31p.
- RODRIGUES, R. S. A.; LACERDA, H. B. **Projeto e Construção de uma Mesa de Coordenadas XY de Baixo Custo para Corte e Solda**. Revista Horizonte Científico, Universidade Federal de Uberlândia, 2003, 12p. Disponível em: www.propp.ufu.br/revistaeletronica/exatas2003/projeto.pdf. Acesso em: 06 nov. 2008.
- SATO, F. H. **Automação do Sistema de Carregamento em Ensaios Estruturais**. Dissertação de Mestrado da Universidade Estadual Paulista Júlio Mesquita/Ilha Solteira, Curso de Engenharia Civil, São Paulo-SP, 2002, 71p.
- SILVA, M. B. **Processos de Usinagem**. Doutor em Engenharia Mecânica, University of WalWick, Inglaterra, 2006. Disponível em: <http://www.buzin.ind.br/destaques.asp?idDestaque=6>. Acesso em: 05 mai. 2007.
- SILVEIRA, P. R.; WINDERSON, E. **Automação e Controle Discreto**. 1 ed., São Paulo, Editora Érica, 1999, 229p.
- SPERB, D. Q. **Desenvolvimento de um Dispositivo Programável de Movimento Passivo Contínuo para Membros Inferiores**. Dissertação de Mestrado da Universidade Federal de Santa Maria, Curso de Engenharia de Produção, Santa Maria-RS, 2008.
- TÁVORA, J. L. **Crítérios de Competitividade na Análise de Projetos**. Anais do XVIII Encontro Nacional de Engenharia de Produção, Niterói-RJ: ABEPRO, 1998, 7p. Disponível em: <http://publicacoes.abepro.org.br/index.asp>. Acesso em: 25 abr. 2007.
- TECNOPAMPA, INDÚSTRIA DE MÁQUINAS LTDA. **Tecnopampa**. 2009. Disponível em: <http://www.tecnopampa.com.br>. Acesso em: 20 jan. 2009.

ANEXOS

Anexo A – Requisitos para Instalação do Sistema

Para o funcionamento do sistema LPTControl é necessário que o computador tenha alguns componentes instalados que são: o *driver* para acessar a porta paralela, o Borland Database Engine e a biblioteca “qtintf.dll” . Caso o computador não possua esses três componentes é necessário instalá-los.

A seguir, seguem os passos de instalação de cada um deles:

Instalando um *driver* no Windows NT/2000 ou XP para rodar programas que acessem a Porta Paralela

Para seguir os passos abaixo certifique-se de ter efetuado *logon* no Windows NT, 2000 ou XP como administrador, ou no mínimo ter privilégios para instalar programas no sistema.

Siga os passos abaixo para instalar o *driver*:

- 1) Descompacte o arquivo UserPort.zip;
- 2) Copie o *Driver* UserPort.sys para a seguinte pasta:

Nas versões do windows NT ou 2000

C:\WINNT\system32\drivers

Na versão do windows XP Service Pack (SP1 e SP2)

C:\WINDOWS\system32\drivers

3) Rode o programa UserPort.exe, selecione os endereços padrão e use o botão Remove para removê-los, conforme a figura abaixo:

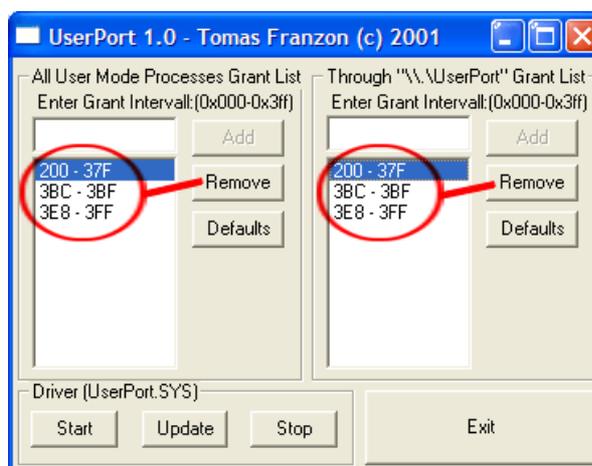


Figura 1

Agora entre com os endereços pretendidos, para isso digite na caixa de edição o endereço inicial e final, assim: 378-37A, depois é só clicar no botão Add, conforme mostra a figura abaixo:

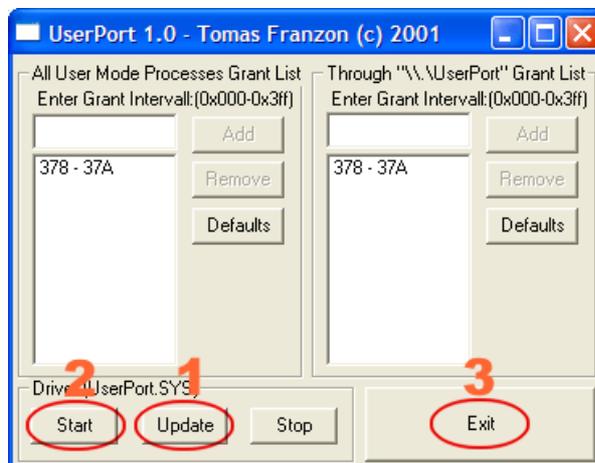


Figura 2

5) Para ativar o *driver* e fechar o programa, clique nos botões: Update, Start e por fim no Exit.

Pronto, agora já é possível executar qualquer programa que acesse a Porta Paralela, no windows NT/2000 ou XP sem nenhuma restrição.

Instalando o Borland Database Engine

Para que o programa LPTControl possa acessar a sua base de dados é necessária a instalação do Borland Database Engine. O procedimento de instalação é muito simples, basta executar o arquivo “Instalador_BDE_5.2_Compacto.exe” e seguir os procedimentos do assistente de instalação.

DLL

O Borland Delphi utiliza a biblioteca “qtintf.dll” para funções que acessa o hardware. Caso o computador onde o sistema esteja sendo instalado não possua esta DLL será necessário copiá-la na pasta “C:\WINDOWS\System32”.

Com esses componentes instalados é só clicar em “LPTControl.exe” para executar o sistema.

Anexo B – Ajuda *Online* do Sistema

Bem-Vindo ao Sistema de Ajuda do Programa LPTControl

Esta ajuda *online* contém todas as instruções necessárias a correta e eficiente utilização do programa para um melhor proveito dos recursos disponíveis por parte do usuário.



Figura 1 - Tela Principal do Programa LPTControl

Menu Principal

Através do menu principal pode-se ter acesso as configurações do programa, abertura do arquivo de passos entre outras opções.



Figura 1 - Menu Principal

O Menu Principal está dividido em quatro submenus:

Arquivo - Permite abrir e fechar um arquivo e sair do programa.

Editar - Dá acesso aos comandos relativos às configurações do programa.

Exibir - Possui comandos para exibir dados sobre a execução do programa.

Ajuda - Permite acesso a ajuda *online* do programa.

Clique sobre uma das opções acima se desejar maiores detalhes.

Menu Arquivo

Através do menu arquivo pode-se ter acesso as opções Abrir, Fechar e Sair.



Figura 1 - Menu Arquivo

O menu arquivo está dividido em três submenus:

Abrir - Permite abrir um arquivo.

Fechar - Fecha o arquivo atual.

Sair - Encerra o programa.

Menu Arquivo Abrir

Clicando em "Arquivo-> Abrir"...



Figura 1 - Arquivo Abrir

Será exibida uma janela para selecionar um arquivo a ser aberto.

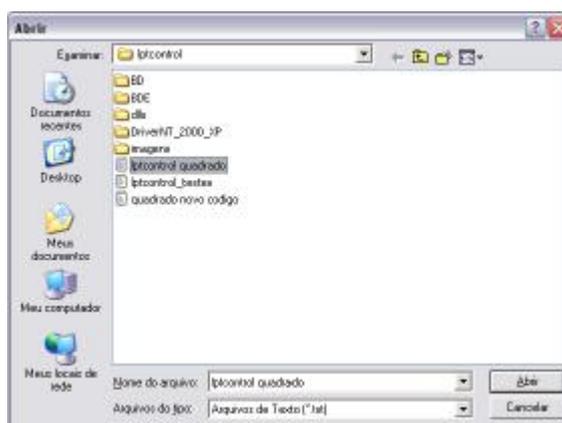


Figura 2 - Caixa de Seleção

Após selecionar o arquivo a ser aberto o usuário deverá clicar no botão abrir para concluir a tarefa.

Menu Arquivo Fechar

Clique em "Arquivo->Fechar" para fechar o arquivo atual.



Figura 1 - Fechar o Arquivo

Menu Arquivo Sair

Clique em "Arquivo->Sair" para sair do programa.



Figura 1 - Sair do Programa

Menu Editar

No menu editar temos a opção de editar as configurações do programa.

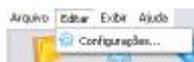


Figura 1 - Menu Editar

Menu Editar Configurações

Clicando em "Editar->Configurações"



Figura 1 - Menu Editar Configurações

Será exibida a janela de configurações do programa.

Configurações Gerais

Nas Configurações Gerais, como vemos na Figura abaixo temos:

O Código é utilizado pelo programa para identificar cada configuração sendo gerado automaticamente a cada vez que é criada uma nova configuração.

A Descrição é o nome pelo qual uma determinada configuração é identificada.

O Intervalo de Duração do Pulso é o tempo de que o *drive* do motor de passo necessita entre um pulso e outro. Este intervalo é definido em microssegundos e pode variar de 1 a 250.000.

O Número de Eixos é a quantidade de eixos utilizada pelo equipamento. Este programa aceita trabalhar com 1, 2 ou 3 eixos, conforme a ferramenta a ser utilizada.

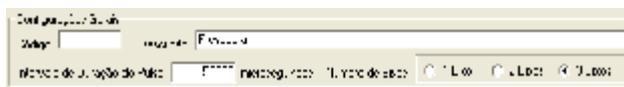


Figura 1 - Configurações Gerais

Configurações da Porta Paralela



Figura 1 - Configurações da Porta Paralela

Nas Configurações da Porta Paralela:

Sinal: A coluna sinal serve para identificar cada um dos bits da porta paralela.

Habilitado: Nesta coluna podemos habilitar ou desabilitar cada uma das saídas da porta paralela de acordo com as necessidades do equipamento que está conectado.

Função: Nesta coluna é possível selecionar uma entre as funções disponíveis para cada bit da porta. Conforme Figura 2.

Invertido: Ao marcar a caixa de verificação invertido pode-se inverter o nível de sinal de saída de um determinado bit.



Figura 2 – Funções

Configurações Motores de Passo

Nas Configurações dos Motores de Passo:

Temos que digitar o tamanho do passo dos eixos x, y e z. Os tamanhos dos passos nas diagonais são preenchidos automaticamente usando a fórmula de Pitágoras.

Figura 1 - Configurações dos Motores de Passo

Ajuste de Velocidade

No Ajuste de Velocidade:

A velocidade máxima é calculada pelo programa.

A velocidade padrão é definida pelo usuário.

Figura 1 - Ajuste de Velocidade

Nova Configuração

Clicando no botão "Nova" será criado um novo registro em branco que permitirá o usuário editar e gravar uma nova configuração.

Figura 1 - Nova Configuração em Branco

Selecionar Configuração

Clicando no botão "Selecionar" terá a opção de selecionar uma das configurações já existentes. Ao selecionar uma configuração será aberta a janela com a configuração escolhida, como no exemplo da Figura 1.

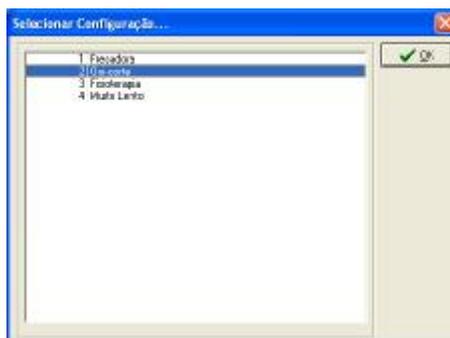


Figura 1 - Selecionar uma Configuração

Salvar Configuração

Clique no botão  para salvar uma configuração nova ou as alterações feitas em uma configuração já existente.

Excluir Configuração

Clique no botão  para excluir uma configuração existente.

Fechar Configuração

Clique no botão  para fechar a janela de configurações.

Menu Exibir

Clicando no menu exibir temos opções para visualizar o status da porta paralela e o arquivo de passos.

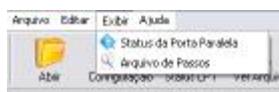


Figura 1 - Exibir Dados do Programa

Status da Porta Paralela

Clicando em "Exibir->Status da Porta Paralela"



Figura 1 - Menu Exibir Status da Porta Paralela

Será exibida a janela de Status da Porta Paralela.



Figura 2 - Status da Porta Paralela

Ela mostra os sinais que estão sendo enviados pelo programa para a porta paralela.

Arquivo de Passos

Clicando em "Exibir->Arquivo de Passos"



Figura 1 - Menu Exibir Arquivo de Passos

Será exibida a janela mostrando o Arquivo de Passos.



Figura 2 - Arquivo de Passos

Através desta janela é possível acompanhar o andamento do arquivo que está sendo executado.

Menu Ajuda

O menu ajuda oferece acesso as opções de ajuda *online* do programa.



Figura 1 - Menu Ajuda

Ajuda do LPTControl

Clicando em "Ajuda->Ajuda do LPTControl"



Figura 1 - Menu Ajuda do LPTControl

Você terá acesso a ajuda *online* do programa.



Figura 2 - Ajuda Online do Programa LPTControl

Sobre

Clicando em "Ajuda->Sobre"



Figura 1 - Opção Sobre do Menu Ajuda

Você abrirá uma janela que contém informações sobre o programa, tais como: versão, autor e uma breve descrição.

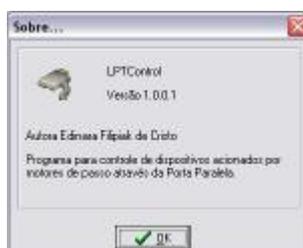


Figura 2 – Sobre

Botão Abrir

Clicando no botão  que está na barra de ferramentas será exibida uma janela para selecionar um arquivo a ser aberto.

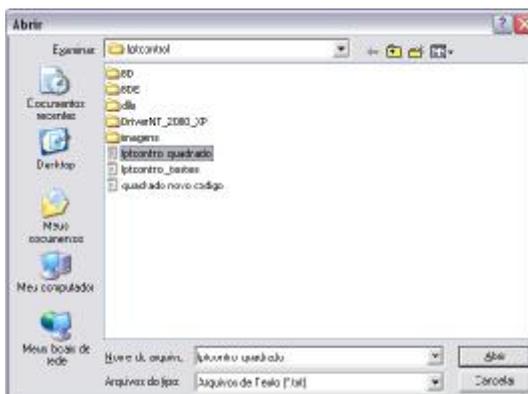


Figura 1 - Caixa de Seleção

Após selecionar o arquivo a ser aberto o usuário deverá clicar no botão abrir para concluir a tarefa.

.....

Botão Configuração

Clicando no botão  que está na barra de ferramentas será exibida a janela de configurações do programa.

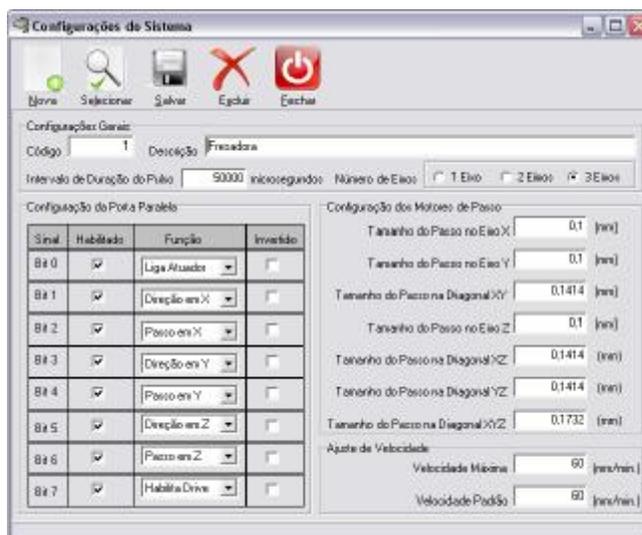


Figura 1 - Editar Configurações

.....

Botão Status LPT

Clicando no botão  que está na barra de ferramentas será exibida a janela de Status da Porta Paralela.



Figura 1 - Status da Porta Paralela

Ela mostra os sinais que estão sendo enviados pelo programa para a porta paralela.

Botão Arquivo de Passos

Clicando no botão  que está na barra de ferramentas será exibida a janela mostrando o Arquivo de Passos.



Figura 1 - Arquivo de Passos

Através desta janela é possível acompanhar o andamento do arquivo que está sendo executado.

Botão Sair

Clique no botão  para sair do programa.

Botão Ajuda

Clique no botão  para ter acesso as opções de ajuda *online* do programa.

Controle de Execução

No controle de execução são mostrados os seguintes itens:

Nome do Arquivo: Nome do arquivo que foi aberto para execução;

Tamanho do Arquivo: Tamanho do arquivo a ser executado;

Comandos Processados: Comandos processados até o momento;

Comandos Restantes: Comandos que restam para ser executados;

Velocidade Atual: Velocidade em mm/min que estão sendo executados os comandos naquele momento;

Aguardando Tempo de Espera: É o intervalo de tempo durante o qual a execução do arquivo é interrompido.

Ao lado de "Aguardando Tempo de Espera" temos o número de repetições do programa, que pode ser entre 1 e 100 repetições.



Figura 1 - Controle de Execução

Botão Reverter

Clicando no botão  o arquivo de passos será executado na ordem inversa.

Botão Executar

Clicando no  será executado o arquivo de passos.

Botão Interromper

Clicando no  será interrompido o arquivo de passos.

Botão Parar

Clicando no  a execução do arquivo de passos será cancelada.

Controle Manual

É o controle manual para cada eixo do equipamento.

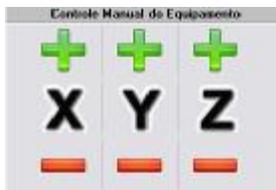


Figura 1 - Controle Manual do Equipamento

Controle Manual Eixo X

É o controle manual do eixo X. Ao clicar no sinal de mais (+) na cor verde é produzido um deslocamento na direção positiva do eixo X. De outra forma, ao clicar

no sinal de menos (-) na cor vermelha é produzido um deslocamento na direção positiva do eixo X.



Figura 1 - Controle Manual do Eixo X

.....

Controle Manual Eixo Y

É o controle manual do eixo Y. Ao clicar no sinal de mais (+) na cor verde é produzido um deslocamento na direção positiva do eixo Y. De outra forma, ao clicar no sinal de menos (-) na cor vermelha é produzido um deslocamento na direção positiva do eixo Y.



Figura 1 - Controle Manual do Eixo Y

.....

Controle Manual Eixo Z

É o controle manual do eixo Z. Ao clicar no sinal de mais (+) na cor verde é produzido um deslocamento na direção positiva do eixo Z. De outra forma, ao clicar no sinal de menos (-) na cor vermelha é produzido um deslocamento na direção positiva do eixo Z.



Figura 1 - Controle Manual do Eixo Z

Drive Ligado

Pressione o botão para ligar o *drive* da ferramenta. Quando o botão estiver verde é porque o *drive* está ligado.



Figura 1 - Drive Ligado

Drive Desligado

Pressione o botão para desligar o *drive* da ferramenta. Quando o botão estiver vermelho é porque o *drive* está desligado.



Figura 1 - Drive Desligado

.....

Ferramenta Ligada

Pressione o botão para ligar a ferramenta. Quando o botão estiver verde é porque a ferramenta está ligada.



Figura 1 - Ferramenta Ligada

.....

Ferramenta Desligada

Pressione o botão para desligar a ferramenta. Quando o botão estiver vermelho é porque a ferramenta está desligada.



Figura 1 - Ferramenta Desligada

.....

Botão Gravar

Entre os botões para ligar e desligar o *drive* e a ferramenta temos o botão Gravar que serve para armazenar os passos que o usuário dará manualmente em um arquivo para um posterior uso. Esta técnica é conhecida como programação por aprendizagem.



Figura 1 - Botão Gravar

Controle de Velocidade

É a velocidade com que é executado o arquivo de passos. O controle de velocidade pode ser feito manualmente ou calculado pelo programa.



Figura 1 - Controle de Velocidade