

DAS Departamento de Automação e Sistemas
CTC **Centro Tecnológico**
UFSC Universidade Federal de Santa Catarina

Controle de Trajetória de um Manipulador Cartesiano de Dois Eixos

*Monografia submetida à Universidade Federal de Santa Catarina
como requisito para a aprovação da disciplina:*

DAS 5511: Projeto de Fim de Curso

Giulliano Carlo Jesus Pereira

Florianópolis, Maio de 1999

**Controle de Trajetória de um Manipulador
Cartesiano de Dois Eixos**

Giulliano Carlo Jesus Pereira

Esta monografia foi julgada no contexto da disciplina
DAS 5511: Projeto de Fim de Curso
e aprovada na sua forma final pelo
Curso de Engenharia de Controle e Automação Industrial

Banca Examinadora:

Jair Carlos Dutra
Orientador

Raul Gohr Jr.
Co-orientador

Prof. Augusto Humberto Bruciapaglia
Responsável pela disciplina e Coordenador do Curso

Prof. Antonio Augusto Rodrigues Coelho, Avaliador

Gil Riella, Debatedor

Fabrizio Carlo Mezzari, Debatedor

A meu pai

Agradecimentos

A Deus, por ter me dado a oportunidade de concluir mais este importante passo em minha vida.

A meu pai, pelo seu incentivo, apoio, orientação, e por ter sempre acreditado em mim, até mesmo quando eu não acreditei.

A minha família, por compreender a minha ausência ao longo destes cinco anos.

À jornalista Silvia Priscilla Argenta, pela correção gramatical deste trabalho e pelo carinho demonstrado no dia-a-dia.

Ao Instituto de Mecatrônica – LABSOLDA, pela oportunidade de realizar este projeto, em especial ao professor Jair Carlos Dutra e ao engenheiro Raul Gohr Jr. pela orientação.

Aos amigos do curso, pelas nossas “batalhas”.

E a todos a quem não posso citar individualmente, mas que não foram, de modo algum, menos importantes para que eu conseguisse alcançar este dia.

Resumo

O principal objetivo do presente trabalho é realizar o controle de trajetória de um manipulador cartesiano de dois eixos.

Um planejamento de trajetória é obtido, a partir de perfis de velocidade trapezoidais. As trajetórias desenvolvidas são movimentos de pendulamento comuns em processos de soldagem.

O sistema de acionamento é constituído de dois motores de passo, comandados por um computador pessoal.

São feitos ensaios e testes de laboratório, e apresentados os resultados do desempenho do sistema.

Abstract

The main objective of the present work is to realize the Trajectory Control of a Two Axis Cartesian Manipulator.

A trajectory planning is obtained, by a trapezoidal velocity profile. The developed trajectories are usual oscillating movements in welding processes.

Two stepper motors, commanded by a personal computer, compose the driver system.

Laboratorial tests are made, and results of system performance are shown.

Sumário

CAPÍTULO I CONSIDERAÇÕES GERAIS	1
1.1 INTRODUÇÃO	1
1.2 O INSTITUTO DE MECATRÔNICA - LABSOLDA	3
1.3 CONTEXTO DO PROJETO	6
CAPÍTULO II OS MOTORES DE PASSO	8
2.1 INTRODUÇÃO	8
2.2 ACIONAMENTO DOS MOTORES DE PASSO	9
2.3 VANTAGENS E DESVANTAGENS DOS MOTORES DE PASSO	10
2.4 CLASSIFICAÇÃO ESTRUTURAL DOS MOTORES DE PASSO	11
2.5 ESTÁGIO AMPLIFICADOR DE SAÍDA - <i>DRIVER</i>	11
2.6 MODOS DE EXCITAÇÃO DOS MOTORES DE PASSO	12
2.7 MODO DE ACIONAMENTO DE MOTORES DE PASSO – SENTIDO DA CORRENTE NAS FASES	12
2.8 SISTEMA DE COMANDO/CONTROLE	13
2.9 CONCLUSÃO	14
CAPÍTULO III METODOLOGIA DO PROJETO	15
3.1 INTRODUÇÃO	15
3.2 O “TARTÍLOPE” VERSÃO I	15
3.3 ETAPAS DO PROJETO	17
3.4 O “TARTÍLOPE” VERSÃO II	19
3.4.1 O SISTEMA MECÂNICO	20
A) AS REDUÇÕES	20
B) A ESTRUTURA DO SEGUNDO EIXO	20
3.4.2 O ACIONAMENTO - <i>DRIVERS</i>	21
A) O CONTROLADOR DE MOTORES DE PASSO – L297	22
B) O CONVERSOR DUAL DE PONTE COMPLETA - L298N	22

3.4.3 A UNIDADE DE CONTROLE _____	23
3.5 POR QUE ADOTAR ESTA METODOLOGIA DE PROJETO? _____	23
3.6 CONCLUSÃO _____	24
CAPÍTULO IV PLANEJAMENTO DE TRAJETÓRIA _____	25
4.1 INTRODUÇÃO _____	25
4.2 CAMINHO E TRAJETÓRIA _____	25
4.3 ESPAÇO OPERACIONAL E ESPAÇO DE JUNTAS _____	26
4.4 AS CURVAS DE VELOCIDADE _____	27
4.5 AS TRAJETÓRIAS IMPLEMENTADAS _____	30
4.5.1 TRAJETÓRIA DE PERÍODO TRIANGULAR _____	30
4.5.2 TRAJETÓRIA DE PERÍODO QUADRADO _____	34
4.5.3 TRAJETÓRIA DE PERÍODO TRAPEZOIDAL _____	36
4.6 CONCLUSÃO _____	39
CAPÍTULO V O DESENVOLVIMENTO DO PROJETO _____	40
5.1 INTRODUÇÃO _____	40
5.2 AMBIENTE DE DESENVOLVIMENTO _____	40
5.3 CONSIDERAÇÕES DE <i>HARDWARE</i> _____	41
5.4 SINAIS DE CONTROLE _____	41
5.4.1 O SINAL DE <i>CLOCK</i> _____	42
5.4.2 O SINAL DE DIREÇÃO _____	42
5.4.3 O SINAL DE LIBERA _____	42
5.4.4 O SINAL DE MODO DE VELOCIDADE _____	43
5.5 AS FREQUÊNCIAS DE <i>CLOCK</i> _____	43
5.6 AS ROTINAS DE <i>SOFTWARE</i> _____	45
5.6.1 AS FUNÇÕES DE QUE GERAM UM PULSO DE <i>CLOCK</i> _____	46
5.6.2 AS FUNÇÕES DE CONTROLE DOS SINAIS DE DIREÇÃO _____	47
5.6.3 AS FUNÇÕES DE CONTROLE DOS BITS DE MODO DE VELOCIDADE _____	49
5.6.4 AS FUNÇÕES DE CONTROLE DOS BITS DE LIBERA _____	50
5.7 GERAÇÃO DOS SINAIS DE <i>CLOCK</i> _____	51
5.8 CÁLCULO E ARMAZENAMENTO DA VELOCIDADE _____	52
5.9 ROTINA DE GERAÇÃO DAS FREQUÊNCIAS ELEMENTARES _____	55

5.10 O CONTROLE DE TRAJETÓRIA _____	56
5.10.1 O CONTROLE DE VELOCIDADE _____	57
5.10.2 O SEGUIMENTO DE CAMINHO _____	58
5.11 CONCLUSÃO _____	63
CAPÍTULO VI RESULTADOS OBTIDOS _____	64
6.1 INTRODUÇÃO _____	64
6.2 O PROTÓTIPO MECÂNICO _____	64
6.3 A INTERFACE DO <i>SOFTWARE</i> _____	67
6.4 ENSAIOS REALIZADOS _____	67
6.4.1 A TRAJETÓRIA DE PERÍODO QUADRADO _____	68
6.4.2 A TRAJETÓRIA DE PERÍODO TRIANGULAR _____	69
6.4.3 A TRAJETÓRIA DE PERÍODO TRAPEZOIDAL _____	70
6.4.4 A TRAJETÓRIA LINEAR _____	70
6.5 CONCLUSÃO _____	71
CAPÍTULO VII CONCLUSÕES E PERSPECTIVAS _____	72
REFERÊNCIAS BIBLIOGRÁFICAS _____	74

Capítulo I

Considerações Gerais

1.1 Introdução

Nos países desenvolvidos, há mais de duas décadas, vem sendo estudada a automação dos processos de soldagem que tem por objetivo obter melhoras significativas na qualidade da produção. Atualmente, no âmbito industrial brasileiro, também está muito em voga o conceito de qualidade total, o que motiva a automação dos processos de soldagem para um aumento da qualidade dos produtos aliado a uma diminuição nos custos de fabricação.

Quando juntas chanfradas ou em filete são unidas, resultados de melhor qualidade são obtidos se a tocha de soldagem realizar movimentos pendulares transversais às mesmas (ilustração 1-1). Estes movimentos são denominados de tecimento, e de acordo com o tipo de material e da junta a ser soldada, eles têm de ser cumpridos seguindo trajetórias definidas. Os processos de soldagem a arco, como o processo TIG, MIG/MAG e plasma utilizam o tecimento. As formas de tecimento mais comuns são mostradas na ilustração 1-2.

Assim, um controle de trajetória torna-se necessário para que a tocha realize movimentos axiais e transversais sobre a junta a ser unida, respeitando uma velocidade de soldagem e de tecimento predeterminada.

Outra aplicação do controle de trajetória é nas operações de corte, como o oxicorte e plasma. Nestes processos, figuras geométricas elementares são obtidas a partir de chapas de metal cortadas seguindo-se um perfil de velocidade desejado.

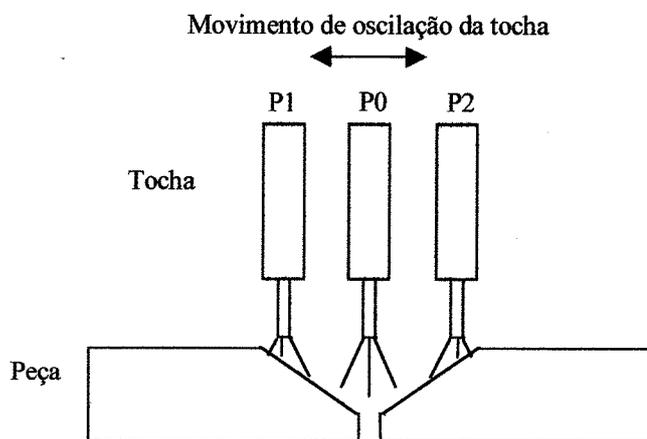


Ilustração 1-1: pendulamento da pistola de soldagem (tecimento)

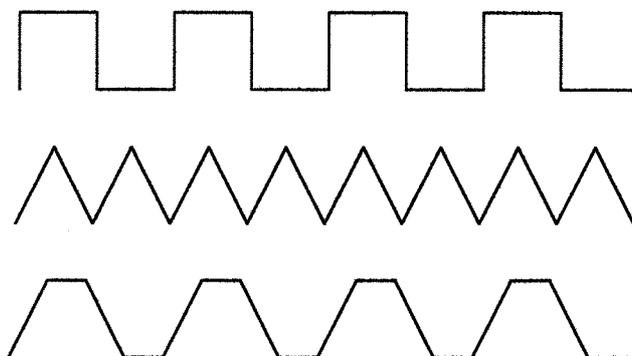


Ilustração 1-2: movimentos de tecimento mais comuns

Este projeto foi desenvolvido nesta área da automação, tendo por objetivo o desenvolvimento de algoritmos para o controle de trajetória de um manipulador cartesiano de dois eixos. O manipulador é capaz de realizar vários movimentos de tecimento seguindo trajetórias específicas. Além disso, seguindo-se a mesma "filosofia" de projeto, pode-se programar caminhos geométricos como quadrados, circunferências, triângulos etc.

Dispositivos similares são somente fabricados no exterior, de forma que o desenvolvimento de um equipamento nacional representaria a eliminação da importação, além de atender a grande demanda interna existente por máquinas do gênero.

O acionamento, bem como o mecanismo foram desenvolvidos no laboratório, promovendo a engenharia simultânea nas várias etapas de criação do sistema. Os motores utilizados foram os de passo híbridos, que devido a sua alta precisão dispensaram o uso de realimentação. Dessa forma, implementou-se um controle em malha aberta: as informações são processadas por um computador digital em tempo real, e enviadas a dois *drivers* bipolares que acionam os motores.

O projeto foi realizado no Instituto de Mecatrônica – LABSOLDA, localizado no Departamento de Engenharia Mecânica da Universidade Federal de Santa Catarina, tendo como base um deslocador automático da pistola de soldagem, anteriormente desenvolvido no laboratório. Este sistema, denominado “Tartílope”, somente permite movimentos lineares. Assim, o projeto objetiva o desenvolvimento da versão II desse deslocador, com a implementação de movimentos coordenados em um segundo eixo.

1.2 O Instituto de Mecatrônica - LABSOLDA

O Instituto de Mecatrônica - LABSOLDA foi criado em 1973 após um convênio de cooperação entre o hoje designado "Forschungszentrum" de Jülich/ Alemanha e o CNPq. Suas atividades foram, entretanto, intensamente iniciadas em 04/03/74. Neste período, seus integrantes só tinham a opção de desenvolver a pesquisa básica nos processos com os equipamentos disponíveis: uma máquina MIG, uma TIG, uma de solda a ponto e dois conversores rotativos de soldagem.

As atividades de pesquisa básica serviam para a formação acadêmica dos professores que aplicavam seus conhecimentos no ensino teórico e prático e possibilitaram a concretização da primeira dissertação de mestrado em 1976. Com a continuidade destas atividades foi escrito e publicado o livro “Tecnologia da Soldagem a Arco Voltaico”, em 1979.

Embora a participação do LABSOLDA sempre tenha sido ativa em todos os eventos de tecnologia de soldagem no Brasil e no exterior, era difícil o intercâmbio com empresas e indústrias. Esta dificuldade sempre desgostou a equipe do laboratório, que passou a buscar apoios institucionais para ministrar cursos de especialização para engenheiros. Tal apoio veio principalmente da Comissão Nacional de Energia Nuclear, não só para os citados cursos, mas também para o desenvolvimento de fontes de energia para soldagem a arco. Para isto houve também a participação do anteriormente denominado Laboratório de Máquinas e Eletrônica de Potência - LAMEP – do Departamento de Engenharia Elétrica da UFSC, denominado hoje como INEP - Instituto de Eletrônica de Potência.

As pesquisas em metalurgia da soldagem e o estudo das características dos revestimentos dos eletrodos revestidos também sempre estiveram presentes nas atividades do LABSOLDA.

Uma fase mais arrojada do laboratório se iniciou em 1983 com o início do doutorado de dois professores num programa cooperativo com a Alemanha sob o apoio da Sociedade Alemã de Cooperação Técnica (GTZ) e a participação do Instituto de Automatização de Processos de Soldagem (APS) da Universidade Técnica de Aachen - Alemanha. Foi em consequência deste cooperativismo que se criaram condições para a formação de uma infra-estrutura laboratorial de desenvolvimento de instrumentação eletrônica e de equipamentos de alta tecnologia.

Hoje o LABSOLDA mantém intercâmbio com as instituições de ensino e pesquisa a seguir descritas:

- CENPES (Petrobrás)
- PROSOLDA - Universidade Federal de Uberlândia
- Laboratório de Soldagem - Universidade Federal do Ceará
- Centro de Pesquisas KFA - Jülich - Alemanha
- Instituto de Tecnologia dos Materiais Uni BW - Hamburg - Alemanha

Além disso, o LABSOLDA tem fornecido equipamentos, instrumentação e prestado consultoria para as seguintes empresas:

- Acesita
- INCOR
- Senai/Florianópolis
- Mercedes Benz
- White Martins
- EMBRACO
- ELETROSUL
- Irmãos Fischer
- CENPES / PETROBRÁS
- COPESP (Ministério da Marinha)
- Centro Técnico Aeroespacial - CTA
- Companhia Paranaense de Energia - COPEL
- Escola Técnica Federal de Santa Catarina
- Escola Federal de Engenharia de Itajubá
- Universidade de Ijuí
- Universidade de São Paulo - USP
- Universidade do Vale do Rio Sinos - UNISINOS
- Universidade Federal de Uberlândia - UFU

- Universidade Federal do Ceará - UFC
- Universidade Federal de Minas Gerais - UFMG
- Universidade Federal do Rio Grande do Norte - UFRN
- Universidade Estadual de Campinas - Unicamp
- Universidade Estadual Paulista – UNESP

1.3 Contexto do Projeto

O desenvolvimento de um sistema de controle de trajetórias se enquadra no contexto das disciplinas oferecidas no Curso de Engenharia de Controle e Automação Industrial. Disciplinas como Microprocessadores, Eletrônica básica e Acionamentos Elétricos para Automação forneceram subsídios para o entendimento do sistema de acionamento dos motores. Já as disciplinas de Informática Industrial I e Engenharia de *Software* possibilitaram o estudo da linguagem C e de técnicas de concepção do *software* de controle. A Robótica contribuiu com a teoria referente ao planejamento das trajetórias implementadas. Finalmente, as disciplinas cursadas na área de Engenharia Mecânica, como Processos de fabricação metal mecânica, Planejamento do Processo e Tecnologia de Comando Numérico facilitaram a troca de informações com os demais técnicos e engenheiros envolvidos na área mecânica do projeto.

O trabalho é então dividido em capítulos, promovendo um entendimento pleno do desenvolvimento do sistema.

O Capítulo II traz uma breve revisão bibliográfica dos assuntos imprescindíveis para análise teórica e entendimento de um sistema de acionamento de motor de passo.

O Capítulo III descreve a metodologia do projeto, trazendo diferenças entre as duas versões do Tartilope.

O Capítulo IV faz uma abordagem sobre o planejamento de trajetória, mostrando as curvas dos movimentos implementados, bem como sua fundamentação matemática.

No Capítulo V é mostrado o modo como o sistema de controle processa e troca informações com o acionamento, sendo também feita uma descrição através de fluxogramas das funções mais relevantes do *software* de comando.

No capítulo VI serão explorados os resultados obtidos através de um protótipo implementado.

Capítulo II

Os motores de passo

2.1 Introdução

O crescimento explosivo da indústria de informática em anos recentes significou também um enorme crescimento para os motores de passos, porque esses motores fornecem a força que aciona muitos dispositivos periféricos de computador. Podem ser encontrados, por exemplo, acionando o mecanismo de alimentação de papel de impressoras. Motores de passo são também exclusivamente usados em *drivers* de discos flexíveis, onde proporcionam um posicionamento preciso da cabeça magnética nos discos. As penas de coordenadas x e y em *plotters* de elevado custo são acionadas por motores de passos. Nestes casos, eles oferecem bom desempenho dinâmico, enquanto eliminam a necessidade do pesado custo de manutenção associado com outros esquemas de acionamento que empregam engrenagens, fios e escovas.

O motor de passo é especialmente adequado para tais aplicações, pois é essencialmente um dispositivo que serve para converter informação de entrada em forma digital para uma saída mecânica. Desta forma, fornece uma interface natural com o computador digital.

O motor de passo, contudo, pode também ser encontrado desempenhando incontáveis tarefas fora da indústria da informática. Em muitas aplicações comerciais, militares e médicas, os motores de passo desempenham funções

como misturar, cortar, agitar, medir, combinar e purificar. Desempenham muitos papéis na produção de alimentos empacotados, produtos comerciais acabados e mesmo na produção de filmes de ficção científica. As vantagens oferecidas pelos motores de passo nestas aplicações podem facilmente ser exploradas com o uso de dois equipamentos auxiliares: o microprocessador e as chaves controladas na forma de transistores.

Embora o motor de passo seja acionado por um conjunto em sequência adequada de sinais digitais, o motor em si exibe as características de um motor síncrono. Em uma de suas formas mais populares, age como uma máquina de excitação dupla, o que por outro lado, significa que produz um torque em regime permanente de uma só velocidade [Toro, 1990].

2.2 Acionamento dos motores de passo

Os Motores de Passo, funcionalmente operam como conversores de energia elétrica em energia mecânica e também de sinais elétricos digitais (pulsos) em sinais mecânicos discretos (posição, velocidade). O deslocamento do elemento de saída de um motor de passo é diretamente proporcional ao número de pulsos aplicados à entrada e a velocidade à frequência dos pulsos. O deslocamento elementar correspondente a um pulso aplicado é denominado passo fundamental do motor [Scheffer, 1998].

Do ponto de vista construtivo, um acionamento para motor de passo pode ser dividido em três partes:

- fonte CC;
- estágio amplificador de saída (*driver*);
- sistema eletrônico de controle e comando (Unidade de Controle);

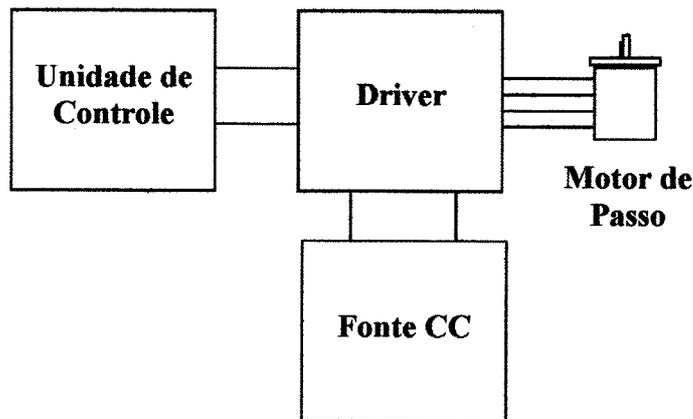


Ilustração 2-1: Diagrama do acionamento de um motor de passo

Os motores de passo são acionamentos que podem ser comandados de forma direta por computadores digitais em malha aberta. Para interfacear um motor de passo com um computador é necessário um circuito lógico digital que realiza o comando de chaveamento da corrente contínua de uma fonte CC para as bobinas do motor [Acarney,1992]. Utiliza-se uma porta de saída de 1 bit do computador para enviar os pulsos e outra para o sentido do movimento, podendo existir mais algumas, como para o modo de excitação, *reset*, etc.

2.3 Vantagens e desvantagens dos motores de passo

Como principais vantagens do uso dos motores de passo pode-se citar:

- Menor incidência de falhas em relação aos outros servomotores;
- ângulo de rotação do motor diretamente proporcional ao número de pulsos do sinal de entrada, facilitando a implementação de controle digital;
- robustez, baixo custo e pequena necessidade de manutenção;
- resposta rápida aos mais diversos comandos;

- alta confiabilidade, não necessitando na maioria das aplicações de realimentação.

Em contrapartida, pode-se enumerar como desvantagens:

- os efeitos de ressonância apresentados na frequência natural;
- o aquecimento existente quando operam com grandes cargas, devido ao alto consumo de corrente.

2.4 Classificação estrutural dos motores de passo

Os motores de passo rotativos podem ser classificados quanto ao princípio de funcionamento em

- motores de ímãs permanentes;
- motores de relutância variável ou
- motores híbridos.

Mais detalhes a respeito desses tipos de motores podem ser encontrados em [Lichtblau, 1989].

2.5 Estágio Amplificador de Saída - *Driver*

Para que o motor possa ser acionado pelos pulsos digitais na sequência e velocidade desejadas, faz-se necessário o uso de um circuito *driver*. O *driver* é um circuito que engloba um estágio amplificador permitindo excitar o motor através do fornecimento de energia elétrica na forma e quantidade adequadas para que este execute com precisão e rapidez os comandos recebidos. O *driver* deve amplificar o sinal de comando à sua entrada entregando-o ao motor. A definição da corrente

é de grande importância no funcionamento do motor, visto que é dela que depende o torque do mesmo.

As funções principais do *driver* são de sequenciar, distribuir e amplificar adequadamente os pulsos do sinal de comando, excitando cada bobina do motor de passo em valores específicos. O sequenciamento é realizado de acordo com a informação do sentido de deslocamento. As saídas deste circuito são tensão e corrente para cada uma das fases do motor .

O termo *driver* é confundido com o sistema completo de acionamento do motor, em vista de sua importância fundamental no funcionamento do circuito.

2.6 Modos de excitação dos motores de passo

Há diversas características de acionamentos para motores de passo, que são apresentadas a seguir:

O tipo de excitação pode ser:

- em passo completo, no qual todas as fases do motor sempre estão energizadas. O deslocamento correspondente a um pulso é igual ao passo fundamental do motor.
- Em meio passo, no qual duas ou uma fase estão energizadas. O deslocamento correspondente a um pulso é igual a metade do passo fundamental.
- Em micro passo, no qual é realizado controle analógico da corrente nas fases.

O tipo de excitação a ser selecionado depende da resolução requerida na aplicação.

2.7 Modo de acionamento de motores de passo – sentido da corrente nas fases

O modo de acionamento que diz respeito ao sentido da corrente nas fases pode ser:

- unipolar, quando a corrente circula somente em 1 sentido nas fases do motor.
- Bipolar, quando a corrente circula em ambos os sentidos nas fases do motor.

Nos acionamentos bipolares as fases são ligadas em paralelo duas-a-duas. O acionamento bipolar permite obter maior torque em baixa velocidade. Quando o principal requisito é alta velocidade, o acionamento unipolar é recomendado por apresentar menor constante de tempo.

2.8 Sistema de Comando/Controle

Para se acionar um motor de passo realizando o controle por passo completo, são necessários apenas dois sinais retangulares em quadratura. De acordo com cada sequência de fase ativa, o eixo do motor gira no sentido horário ou anti-horário, numa velocidade proporcional à frequência dos pulsos de entrada [Bento, 1989].

Geralmente estão presentes cinco sinais no controle de todas as operações básicas do motor:

- **Clock:** o sinal de *clock* dá ao motor o comando do passo. É a frequência com que o motor vai responder aos comandos.

- **Reset:** coloca todos os sinais do sistema numa posição de partida predefinida.
- **Direção:** determina o sentido do movimento de rotação do eixo do motor.
- **Modo de excitação:** decide se o sistema vai operar por passo completo, meio passo ou ainda micropasso.
- **Libera:** deixa o motor “livre” para um eventual deslocamento manual. Neste modo deve-se tomar algumas precauções visto que o motor funciona como gerador.

A interface entre um motor de passo e um computador é realizada através de um circuito lógico digital que realize o comando do chaveamento da corrente contínua de uma fonte CC para as bobinas do motor. Utiliza-se um bit da porta de saída do processador para enviar os pulsos, outro bit para enviar a informação de sentido de movimento ao circuito lógico e mais dois com a forma de onda da sequência de excitação nas fases.

Para gerar um passo do motor via *software* é preciso primeiro determinar a taxa de ocorrência desejada de passos e calcular o intervalo de tempo necessário entre os passos. O computador é então programado para contar este período antes de enviar os sinais ao motor para mover mais um passo.

2.9 Conclusão

Este capítulo permitiu uma noção do embasamento teórico necessário para o entendimento de um sistema de acionamento de motores de passo. Com esta breve revisão bibliográfica apresentada, tem-se subsídios para se fazer uma análise temática do projeto e compreender a metodologia empregada.

Capítulo III

Metodologia do projeto

3.1 Introdução

O projeto em questão propõe-se à realização do controle de trajetória de um manipulador cartesiano de dois eixos. Trata-se do melhoramento de um sistema de deslocamento automático da pistola de soldagem desenvolvido no laboratório: o Tartílope versão I. Este sistema de um único eixo somente permitia deslocamentos lineares segundo uma velocidade ajustada pelo usuário. Portanto, técnicas de controle de trajetória aliadas com um novo projeto mecânico foram utilizadas para a concepção de um novo mecanismo, agora com movimentos controlados em dois eixos. Este capítulo trata da discussão dos componentes selecionados para o desenvolvimento do projeto.

3.2 O “Tartílope” versão I

O “Tartílope” versão I é um equipamento desenvolvido no LABSOLDA que permite deslocar a pistola de soldagem linearmente, em posições horizontais ou verticais, com velocidade entre 5 e 160 cm/min. Seu acionamento é realizado através de um motor de passo da marca Sincro e por um *driver* bipolar desenvolvido no laboratório.

Um modelo tridimensional desse mecanismo foi inicialmente desenvolvido em AutoCAD (ilustração 3.1), o que facilitou a otimização das

dimensões e disposição dos componentes. As peças foram então fabricadas através dos processos de usinagem e fundição.

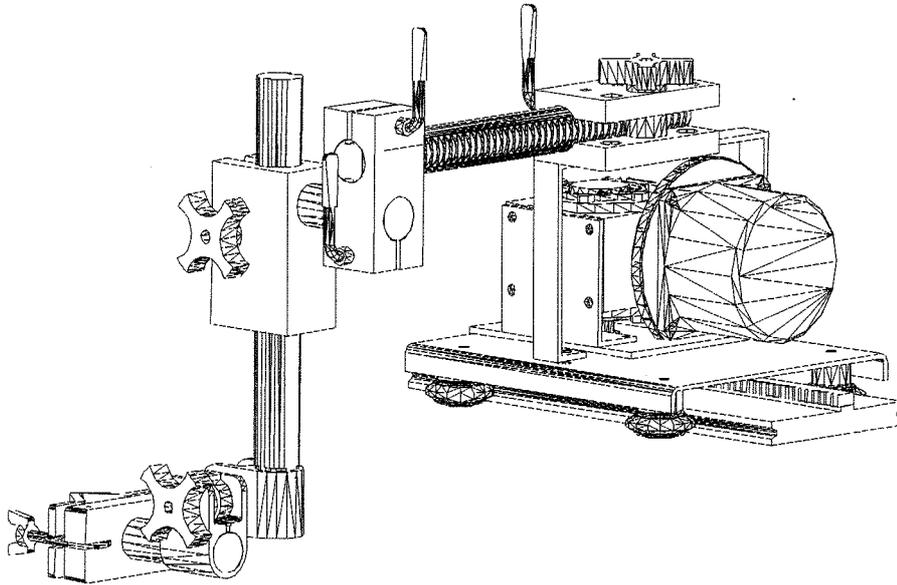


Ilustração 3-1: modelo em AutoCAD do Tartilope Versão I

O sistema é acionado por um motor de passo que através de um redutor sem-fim/coroa impulsiona uma engrenagem que está acoplada a uma cremalheira. Esta cremalheira é fixa em um trilho de alumínio, responsável também por guiar os roletes de sustentação do sistema. O curso de deslocamento não tem limitação física, pois vários trilhos podem ser acoplados. O projeto do trilho e dos roletes de sustentação permite operação do sistema em qualquer posição (vertical, horizontal, etc).

O mecanismo pode ser comandado através de um controle remoto onde são feitas as seleções de velocidade e posição desejadas.

Sua lógica de comando é realizada através do microcontrolador 8031 que reside em um módulo de controle juntamente com o *driver* do motor.

A seguir são apresentadas algumas fotos do equipamento na sua versão final. Na ilustração 3-2 são mostrados os componentes do sistema. Já

na ilustração 3-3 é exibido o Tartílope realizando uma soldagem em posição vertical.

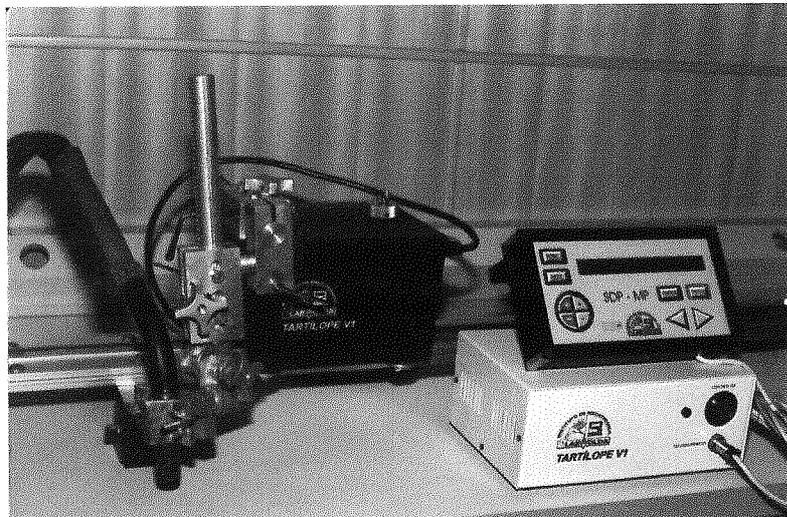


Ilustração 3-2: componentes do Tartílope versão I

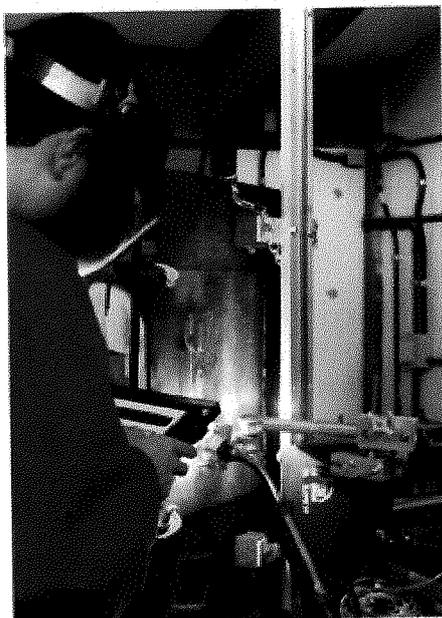


Ilustração 3-3: Tartílope soldando em posição vertical

3.3 Etapas do projeto

O projeto objetiva controlar posição e velocidade em dois motores de passo integrantes de um sistema mecânico de manipulação, de modo que o

efetuador final possa percorrer trajetórias definidas no plano. Neste caso o efetuador final é representado pelo mecanismo de fixação da pistola de soldagem.

Foram utilizados motores de passo híbridos, devido ao seu elevado torque e alta exatidão de posicionamento. Esse motor apresenta ângulo de passo fundamental de 1.8° , ou seja, 200 passos por volta.

Os dois motores foram acionados por dois *drivers* bipolares devido à boa relação de torque e velocidade. Seu projeto é idêntico ao utilizado na versão I.

Para se realizar o controle dos motores, foi utilizado um computador pessoal. Um sinal de alta frequência foi injetado no PC e de acordo com a velocidade desejada foi dividido em várias subfrequências que, devolvidas à porta paralela em conjunto com os outros sinais de comando, geraram as trajetórias estabelecidas.

O projeto teve de considerar a estrutura mecânica do sistema, levando em conta as reduções existentes no mecanismo.

O desenvolvimento do projeto realizou-se em três etapas fundamentais:

Primeira etapa: estudo do acionamento e da interação entre seus subsistemas. Em paralelo, estudou-se também o planejamento de trajetória de movimentos comuns em tectamento e de algumas formas geométricas simples, como o círculo, o quadrado e o triângulo.

Segunda etapa: implementação do controle das trajetórias anteriormente equacionadas via *software*, utilizando-se linguagem C e um processador *pentium* 200MHz. Nesta etapa, os sinais de comando como direção, *reset* e libera foram analisados com auxílio de um *protoboard* e

alguns *leds*. O sinal de alta frequência foi viabilizado através de um gerador de funções. Os sinais de *clock* foram medidos com auxílio de um osciloscópio.

Terceira etapa: teste do sistema de comando/controle, integrado ao sistema mecânico.

Paralelamente a estas etapas, foram realizadas, pela equipe de engenheiros e técnicos do laboratório, a fabricação e as implementações, tanto da estrutura mecânica como também do sistema de acionamento. Isso possibilitou uma interação entre as diversas partes, bem como o acompanhamento do desenvolvimento do projeto nas suas várias etapas de concepção.

3.4 O “Tartílope” versão II

A versão II do Tartílope se compõe das unidades representadas na ilustração 3-4.

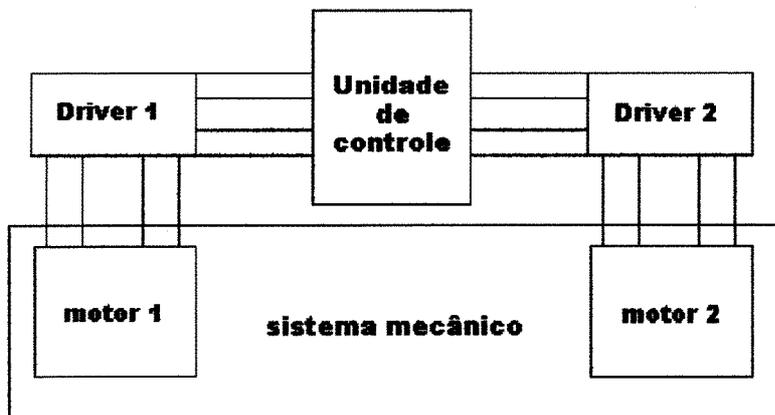


Ilustração 3-4: Composição do Tartílope versão II

As modificações referentes à primeira versão tornam-se evidentes, tanto na estrutura mecânica e no acionamento, como também no sistema de controle. Elas serão apontadas a seguir.

3.4.1 O sistema mecânico

A estrutura mecânica do sistema da versão I foi alterada de modo a permitir um movimento transversal, através da adição de um segundo motor de passo. Devido a limitações de torque de ambos os motores, reduções foram empregadas.

a) As reduções

É comum a utilização de reduções em sistemas mecânicos. Elas permitem alterar a relação entre uma volta completa do motor e o deslocamento efetivo do mecanismo. Outra finalidade do emprego das reduções é a redução do número de rotações do motor aliado com o aumento de torque.

Através de um sistema de reduções, obteve-se um deslocamento efetivo para o eixo X de 4 mm/ volta, e para o eixo Y de 16 mm/ volta, aproximadamente.

Uma caixa de redução aumentou em 15 vezes o torque para o eixo X e proporcionalmente diminuiu o número de rotações do motor. Sem esta redução seria impossível existir movimento do mecanismo devido ao baixo torque efetivo do acionamento.

Para o eixo Y utilizou-se um conjunto correia dentada e polia, aliado a uma redução de 2:1, permitindo uma transmissão precisa do movimento, além de facilitar o projeto do sistema mecânico.

b) A estrutura do segundo eixo

O segundo eixo foi projetado em AutoCAD e fabricado através do processo de usinagem em aço 1020. Trata-se de um protótipo para a

realização de testes. É esta a estrutura responsável pelo movimento transversal do braço. Seu modelo é mostrado na ilustração 3.4.

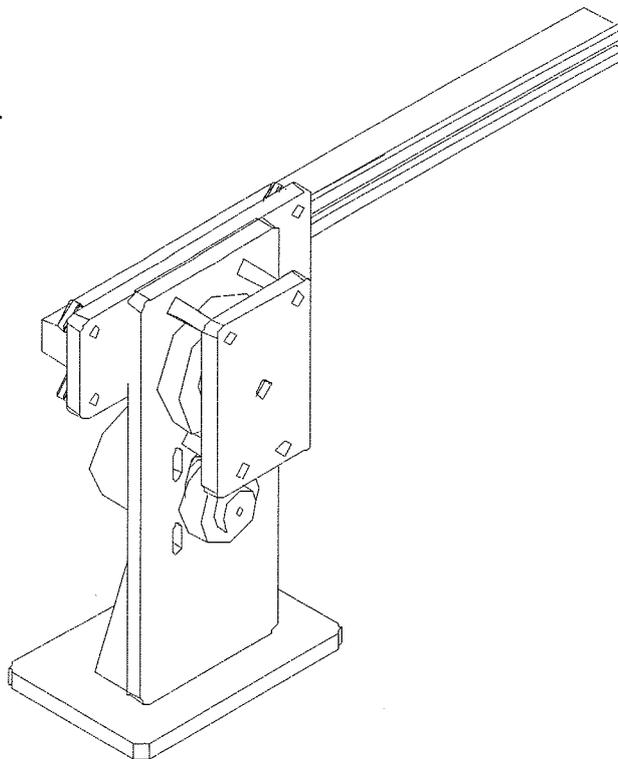


Ilustração 3-5: protótipo do segundo eixo

3.4.2 O acionamento - *drivers*

O acionamento foi realizado individualmente por dois *drivers* bipolares de projeto idêntico ao utilizado na versão I. Eles foram agrupados em um módulo de controle conforme mostra a ilustração 3.5.

Sua construção baseia-se em dois circuitos integrados: um controlador de motores de passo (L297) e um conversor dual de ponte completa (L298). A forma como o *driver* troca informações com a unidade de controle será mostrado no Capítulo V.

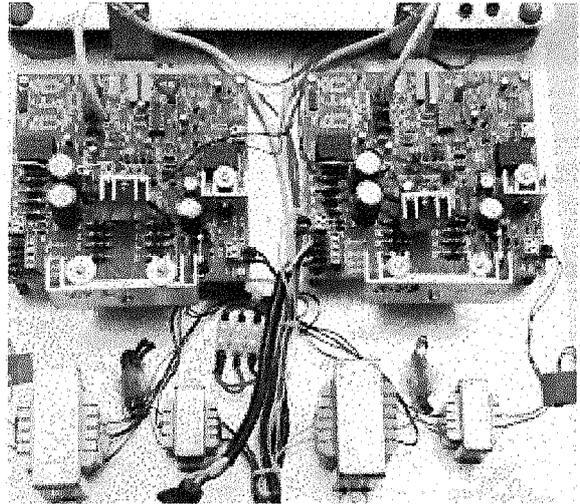


Ilustração 3-6: *drivers* do sistema de acionamento

a) O controlador de motores de passo – L297

O CI controlador de motores de passo L297 gera 4 sinais de fase para motores de passo de duas fases bipolares e para motores de quatro fases unipolares em aplicações de controle digital. O motor pode ser acionado no modo de meio passo, passo completo e em conjunto com circuitos *chopper* PWM, permitindo controle chaveado das correntes nas fases.

Este dispositivo se caracteriza por somente necessitar dos sinais de *clock*, direção e o modo de excitação. Isto reduz bastante a programação em baixo nível, visto que o chaveamento das fases é gerado internamente ao microprocessador.

b) O conversor dual de ponte completa - L298N

O L298N é um conversor dual de ponte completa de alta voltagem e alta corrente designado para acionar dispositivos indutivos como rês, solenóides, motores DC e de passo, através de níveis lógicos TTL.

Este circuito é o principal componente do módulo de potência do *driver*.

3.4.3 A Unidade de Controle

O controle foi realizado através de um computador pessoal. Para tanto, foi desenvolvido um *software* em linguagem C.

Um sinal de alta frequência é injetado através da porta paralela. Após ser adequadamente tratado pelo *software* de controle é devolvido em frequências determinadas aos *drivers*, gerando o movimento nos motores. Este sinal é concebido por um gerador de funções que pode ser substituído futuramente por um *timer* dedicado como o 8253.

O objetivo futuro será de transportar a plataforma de controle desenvolvida para uma CPU industrial, de modo a adequar o equipamento à utilização no chão de fábrica.

3.5 Por que adotar esta Metodologia de Projeto?

A existência de tecnologias já desenvolvidas no laboratório contribuiu para a escolha desta Metodologia de Projeto.

Como mostrado no item 3.4.2 os *drivers* utilizados foram os mesmos existentes na versão I. Somente a parte mecânica foi alterada, pelo acréscimo da estrutura exibida na ilustração 3-5.

O computador pessoal foi utilizado como unidade de controle devido a sua alta performance, facilidade de programação e de implementação de lógicas de controle que exigem um maior processamento de informações. Além disso, a complexidade dos movimentos especificados e o desejo de se implementar futuramente novos tipos de trajetórias, exclui a utilização de um microprocessador. A passagem para o modelo definitivo, com emprego de uma CPU industrial, também torna-se um processo mais natural com a utilização de um PC.

3.6 Conclusão

Este capítulo tratou da forma de implementação do acionamento da segunda versão do Tartílope, aproveitando-se de recursos existentes na versão I. Será necessário utilizar-se de alguma técnica para que este sistema possa realizar movimentos desejados respeitando uma velocidade determinada. Isto caracteriza um problema clássico de planejamento de trajetória.

Capítulo IV

Planejamento de trajetória

4.1 Introdução

A meta do controle de trajetória é gerar entradas de referência para o sistema de controle de movimento, permitindo ao manipulador / posicionador executar movimentos planejados.

Este capítulo abordará a forma como as trajetórias de tecimento serão planejadas, através de um perfil de velocidade trapezoidal. Neste caso o planejamento não consiste em apenas equacionar um movimento, de modo que uma dada velocidade seja seguida e que um caminho seja traçado. Consiste também em encontrar a melhor forma de se implementá-lo através de um computador.

4.2 Caminho e trajetória

A característica básica de um manipulador é a capacidade de mover-se de uma posição inicial para uma posição final estipulada [Craig, 1989]. A transição é caracterizada por leis de movimento que comandam os atuadores de modo a não ultrapassar os limites de saturação e também não excitar os modos ressonantes da estrutura. É então necessário estabelecer algoritmos de planejamento que gerem trajetórias “suaves” e que atendam as especificações.

Para se evitar confusão com termos frequentemente usados como sinônimos, a diferença entre caminho e trajetória será explicada. Um *caminho*

denota o local dos pontos do “espaço de juntas” ou do “espaço operacional”, no qual o manipulador deve seguir na execução de um movimento requerido: um caminho é então a pura descrição geométrica do movimento. Por outro lado, uma *trajetória* é um caminho onde uma lei no tempo é especificada em termos de velocidade e/ou acelerações em cada ponto [Brady et al., 1982].

4.3 Espaço operacional e espaço de juntas

Em princípio, se pode dizer que as entradas para um algoritmo de planejamento de trajetória são as descrições e as restrições do caminho, juntamente com as restrições impostas pela dinâmica do manipulador. Assim, as saídas são as trajetórias do efetuador final em termos de uma sequência de valores no tempo relacionados por posição, velocidade e aceleração.

Um caminho pode ser definido no espaço de juntas ou no espaço operacional. Usualmente o espaço operacional é preferível, visto que ele permite uma descrição natural da tarefa que o manipulador tem de realizar.

Não se especifica completamente um caminho geométrico. Geralmente, um reduzido número de parâmetros é levado em consideração tal como pontos extremos, pontos intermediários e primitivas geométricas interpolando os pontos.

Como a lei de movimento não é tipicamente especificada em cada ponto do caminho geométrico, esta leva em conta o tempo total da trajetória, as restrições de máximas velocidades e acelerações, e eventualmente o valor da velocidade e aceleração em pontos de interesse particular. Com base na informação acima, o algoritmo de planejamento de trajetória gera uma sequência no tempo de variáveis que descrevem a posição do efetuador final e a orientação no tempo em relação às restrições impostas.

O planejamento de trajetória no espaço operacional naturalmente permite computar a presença de restrições no caminho. De fato, algumas

restrições são melhores descritas no espaço operacional, visto que seus pontos correspondentes no espaço de juntas são difíceis de computar.

4.4 As curvas de velocidade

Uma aproximação alternativa das curvas de velocidade de um manipulador é frequentemente adotada na prática industrial, permitindo verificar de maneira direta se as velocidades e acelerações resultantes podem ser suportadas pela estrutura mecânica [Paul, 1979].

Neste caso, um modo de velocidade trapezoidal que impõe uma aceleração constante na fase inicial, uma velocidade intermediária constante e uma desaceleração constante na fase de parada é adotado. A trajetória resultante é formada por um segmento linear conectado por dois segmentos parabólicos nas posições finais e iniciais. A velocidade inicial é nula.

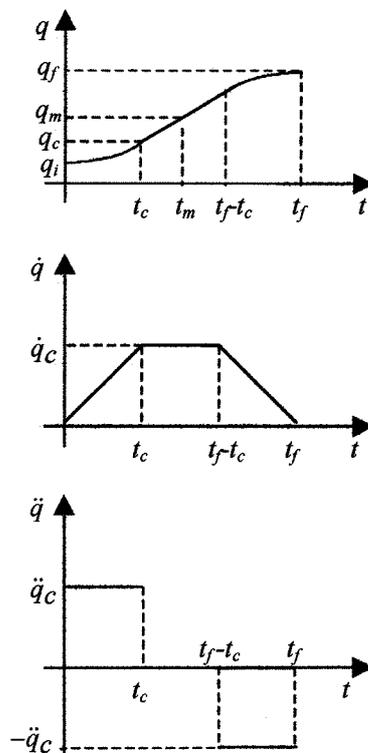


Ilustração 4-1: curvas de movimento adotadas em aplicações industriais

Como pode ser visto nas formas de velocidade na ilustração 4.1, é assumido que ambas as velocidades inicial e final são nulas e os segmentos com acelerações constantes têm a mesma duração de tempo; isto implica numa magnitude igual \ddot{q}_c nos dois segmentos. Nota-se também que a escolha acima leva a uma trajetória simétrica no que diz respeito ao ponto médio $q_m = (q_f + q_i)/2$ em $t_m = t_f/2$.

A trajetória tem de satisfazer algumas restrições para permitir a transição de q_i para q_f num tempo t_f . A velocidade ao final do segmento parabólico tem de ser igual à velocidade (constante) do segmento linear, isto é:

$$\ddot{q}_c t_c = \frac{q_m - q_c}{t_m - t_c}, \quad (4.1)$$

onde q_c é o valor associado ao tempo t_c ao final do segmento parabólico com aceleração constante \ddot{q} (lembrando que $\dot{q}(0) = 0$). No que resulta:

$$q_c = q_i + \frac{1}{2} \ddot{q}_c t_c^2. \quad (4.2)$$

Combinando (4.1) com (4.2) dá

$$\ddot{q}_c t_c^2 - \ddot{q}_c t_f t_c + q_f - q_i = 0. \quad (4.3)$$

Usualmente, \ddot{q}_c é especificado e então para dados t_f , q_i e q_f , o valor de t_c é computada de 4.3 como ($t_c \leq t_f/2$):

$$t_c = \frac{t_f}{2} - \frac{1}{2} \sqrt{\frac{t_f^2 \ddot{q}_c - 4(q_f - q_i)}{\ddot{q}_c}}. \quad (4.4)$$

A aceleração é então sujeita a restrição

$$|\ddot{q}_c| \geq \frac{4|q_f - q_i|}{t_f^2}. \quad (4.5)$$

Quando a aceleração \ddot{q}_c é escolhida de modo a satisfazer 4.5, na igualdade a trajetória resultante não mais apresenta o segmento de velocidade constante e possui apenas os segmentos de aceleração e desaceleração.

Dado q_i , q_f e t_f e desse modo a velocidade de transição média, a equação (4.5) permite calcular um valor de aceleração consistente com a trajetória. Então t_c é computado de (4.4) e a seguinte sequência de polinômios são gerados :

$$q(t) = \begin{cases} q_i + 1/2 \ddot{q}_c t^2 & 0 \leq t \leq t_c \\ q_c + \ddot{q}_c t_c (t - t_c/2) & t_c < t \leq t_f - t_c \\ q_f - 1/2 \ddot{q}_c (t_f - t)^2 & t_f - t_c < t \leq t_f \end{cases} \quad (4.6)$$

Especificar a aceleração no segmento parabólico não é o único modo de determinar trajetórias com curvas de velocidades trapezoidais. Com os valores q_i , q_f e t_f , pode-se obter a velocidade constante \dot{q}_c limitada às seguintes condições:

$$\frac{|q_f - q_i|}{t_f} < |\dot{q}_c| \leq \frac{2|q_f - q_i|}{t_f}. \quad (4.7)$$

Lembrando que $\dot{q}_c = \ddot{q}_c t_c$, a equação (4.3) permite computar t_c como

$$t_c = \frac{q_i - q_f + \dot{q}_c t_f}{\dot{q}_c}, \quad (4.8)$$

e dessa forma a aceleração resultante é

$$\ddot{q}_c = \frac{\dot{q}_c^2}{q_i - q_f - \dot{q}_c t_f}. \quad (4.9)$$

Assim, os valores de t_c e \ddot{q}_c encontrados em (4.8) e (4.9) permitem gerar a sequência de polinômios expressos em (4.6).

4.5 As trajetórias implementadas

As trajetórias foram implementadas, respeitando um perfil de velocidade trapezoidal, conforme anteriormente demonstrado.

São trajetórias muito utilizadas nos processos de tecimento em soldagem, de modo que são todos movimentos periódicos. Esses movimentos oscilatórios proporcionam uma melhor distribuição nas juntas do cordão de solda, melhorando a qualidade da soldagem.

Os movimentos mostrados serão referentes a movimentos periódicos em forma triangular, quadrada e trapezoidal.

4.5.1 Trajetória de período triangular

Uma das trajetórias desenvolvidas, possui período de acordo com a ilustração abaixo.

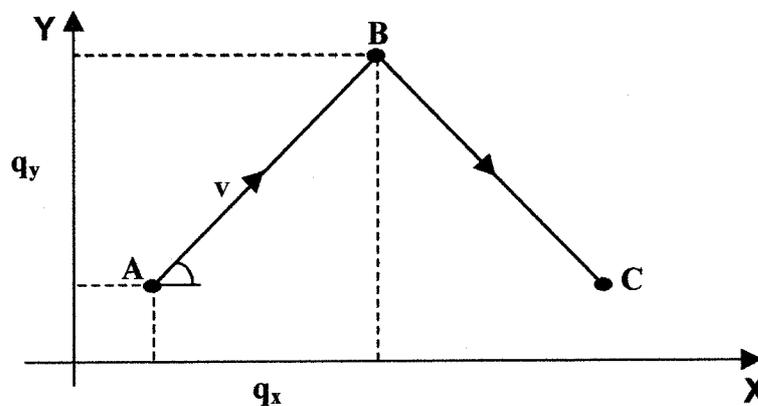
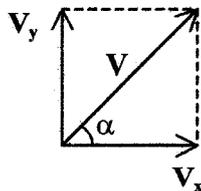


Ilustração 4-2: Trajetória triangular

A trajetória tem início no ponto A com velocidade inicial nula, percorre o trecho \overline{AB} com velocidade constante V e atinge B com velocidade novamente igual a zero. De maneira análoga ao trecho \overline{AB} realiza-se o trajeto \overline{BC} .

É relevante se especificar os parâmetros q_y e q_x , bem como a velocidade V da trajetória para que se possa obter o perfil de velocidade de ambos os eixos (X e Y). A tangente do ângulo α é que relaciona ambas as funções de velocidade $V_x(t)$ e $V_y(t)$ de modo que a trajetória acima estabelecida seja seguida. De fato:



$$\tan \alpha = \frac{V_y}{V_x} \quad (4.10)$$

A mesma relação se aplica para as acelerações:

$$\tan \alpha = \frac{V_y}{V_x} = \frac{A_y}{A_x} \quad (4.11)$$

Resulta que as funções de velocidade e aceleração em ambos os eixos se diferem apenas pela constante $\tan \alpha$.

Adotando-se um perfil trapezoidal para $V_x(t)$ e $V_y(t)$ e estipulando uma aceleração constante para um dos eixos (geralmente, aquele de maior restrição à variações de velocidade), obtêm-se facilmente as curvas de velocidade, aceleração e espaço percorrido para ambos os eixos.

Os gráficos abaixo, comparam $V_x(t)$ e $V_y(t)$.

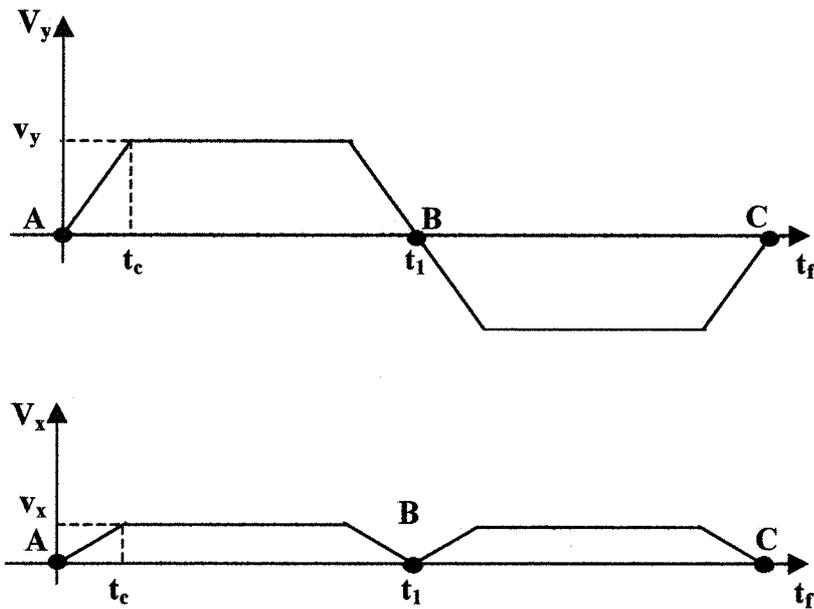


Ilustração 4-3: curvas de velocidade

Nota-se por intermédio dos gráficos acima, que em ambos os eixos, a velocidade nos pontos A, B e C é nula. A partir do ponto B, a velocidade mantém o seu sentido em X e o altera em Y. Além disso, em qualquer instante de tempo t_k , $V_y(t_k) = KV_x(t_k)$, onde $K = \tan \alpha$, o que garante que a trajetória é seguida.

As acelerações, bem como os trechos onde elas irão atuar são os primeiros parâmetros calculados para se obter a trajetória especificada. Nestas curvas, a proporcionalidade entre as duas acelerações é obviamente a mesma entre as velocidades.

A importância de se analisar estes gráficos (ilustração 4-4) reside no fato da necessidade de se observar se as acelerações calculadas podem ser alcançadas e mantidas constantes durante o seu tempo de atuação pelo sistema de acionamento.

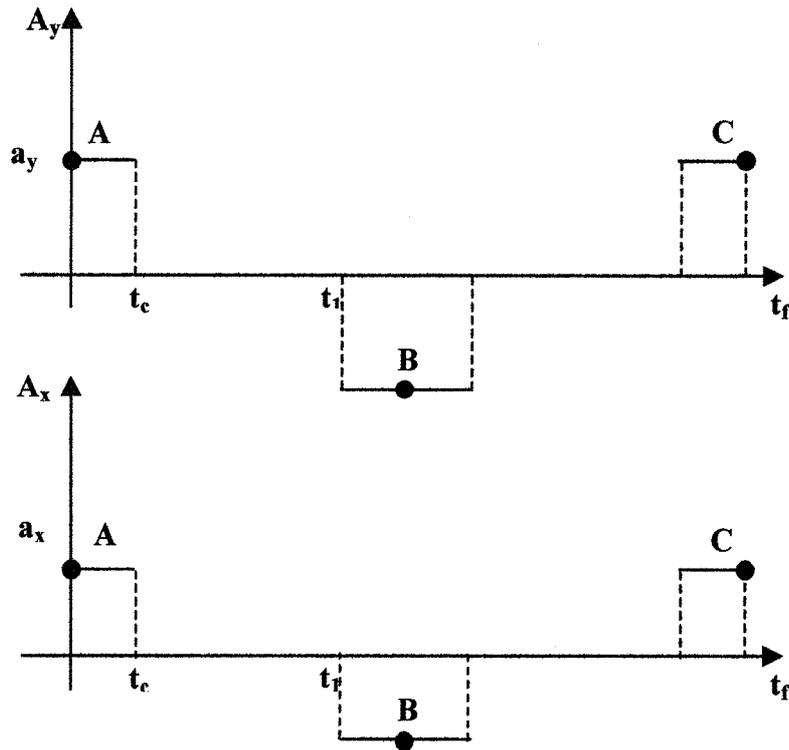


Ilustração 4-4: curvas de aceleração

Os gráficos de Q_y e Q_x são representados por parábolas nos trechos de aceleração constante e por retas onde a aceleração é nula, lembrando que a velocidade ao final de um segmento parabólico tem de ser igual à velocidade (constante) de um segmento linear tem-se os seguintes gráficos:

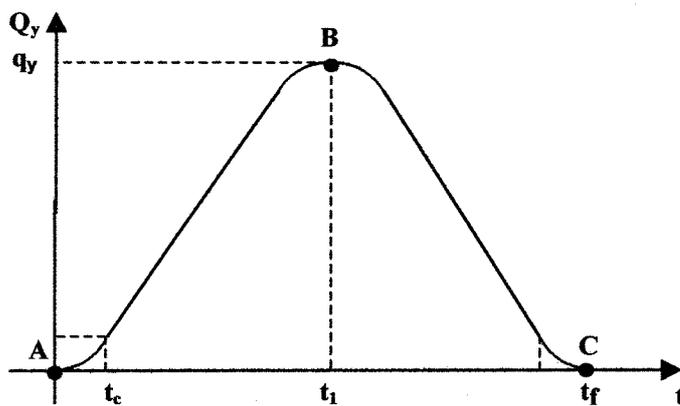


Ilustração 4-5 a: curva de deslocamento em Y

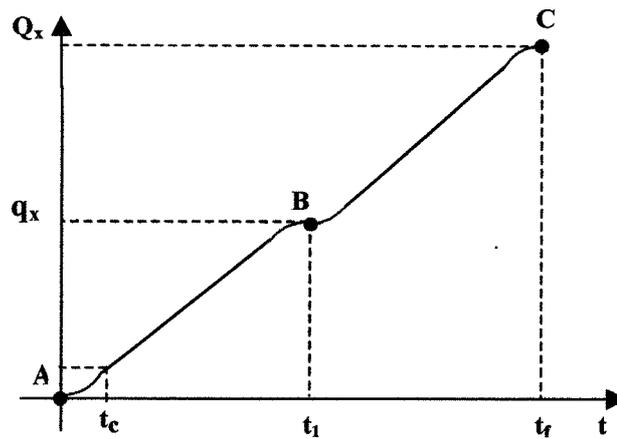


Ilustração 4-5 b: curva de deslocamento em X

4.5.2 Trajetória de período quadrado

A trajetória de período quadrado se desenvolve em 4 etapas. Um deslocamento vertical q_y no sentido \overline{AB} , um deslocamento horizontal q_x no sentido \overline{BC} , outro deslocamento q_y no sentido \overline{CD} e finalizando com um deslocamento horizontal q_x \overline{DE} .

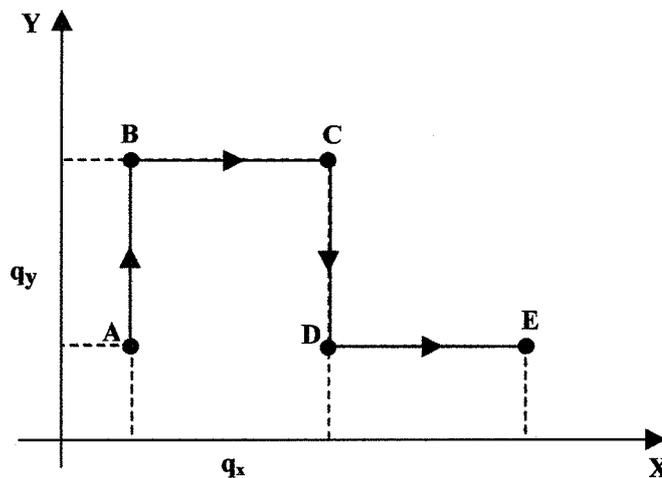


Ilustração 4-6: trajetória quadrada

Em todos os segmentos é desenvolvido um perfil de velocidade trapezoidal. O movimento dos eixos é alternado, não ocorrendo movimento simultâneo em momento algum.

O tratamento matemático é semelhante ao já apresentado na trajetória de período triangular, de modo que somente serão apresentadas as curvas de velocidade (ilustração 4-7), deslocamento (ilustração 4-8) e aceleração (ilustração 4-9) em ambos os eixos.

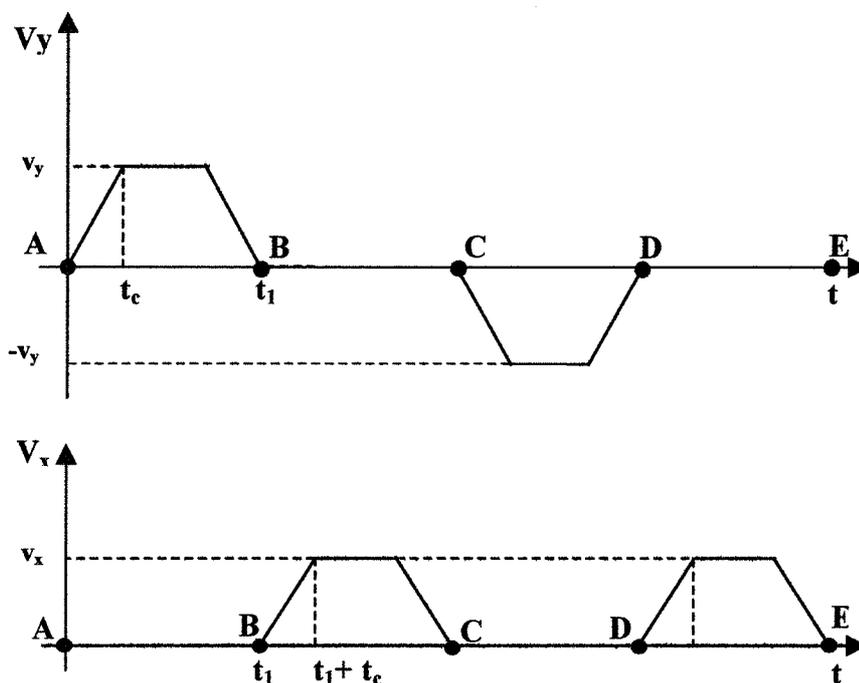


Ilustração 4-7: curvas de velocidade

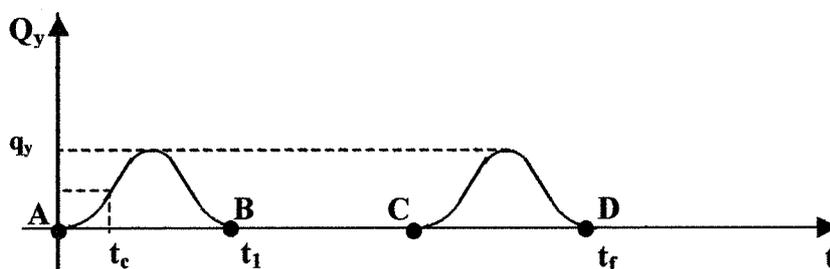


Ilustração 4-8 a: curva de deslocamento em Y

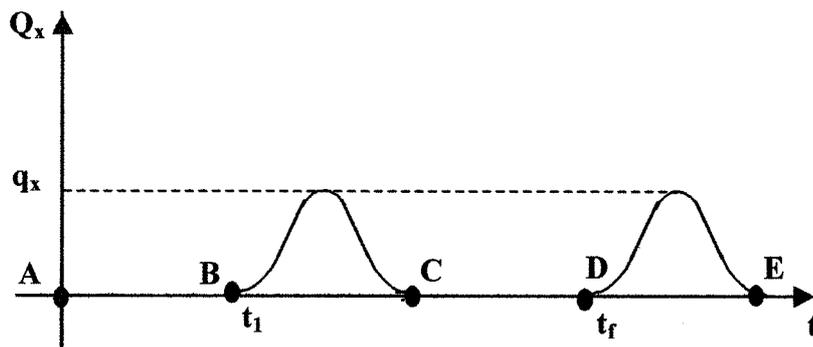


Ilustração 4-8 b: curva de deslocamento em X

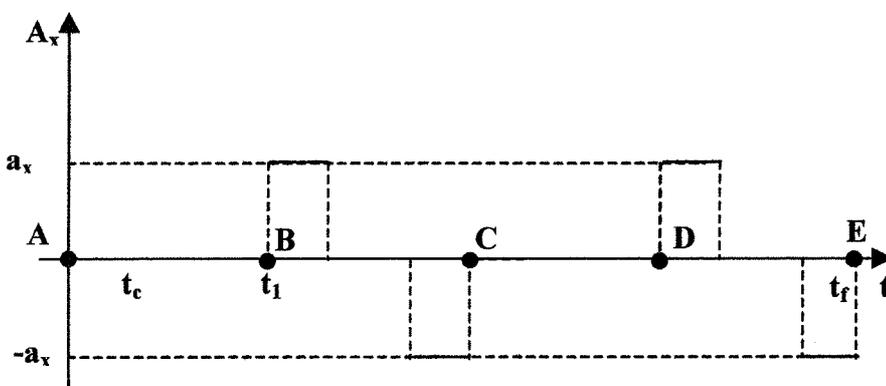
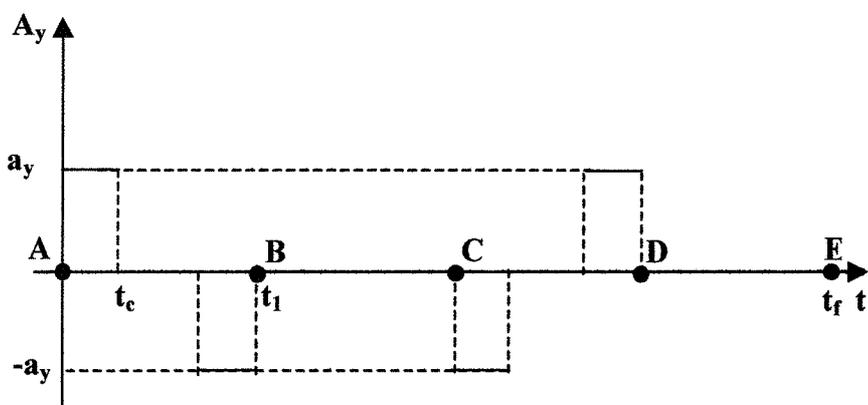


Ilustração 4-9: curvas de aceleração

4.5.3 Trajetória de período trapezoidal

A trajetória de período trapezoidal é composta por movimentos onde apenas um dos eixos se movimenta e também por aqueles onde os dois eixos se movimentam simultaneamente.

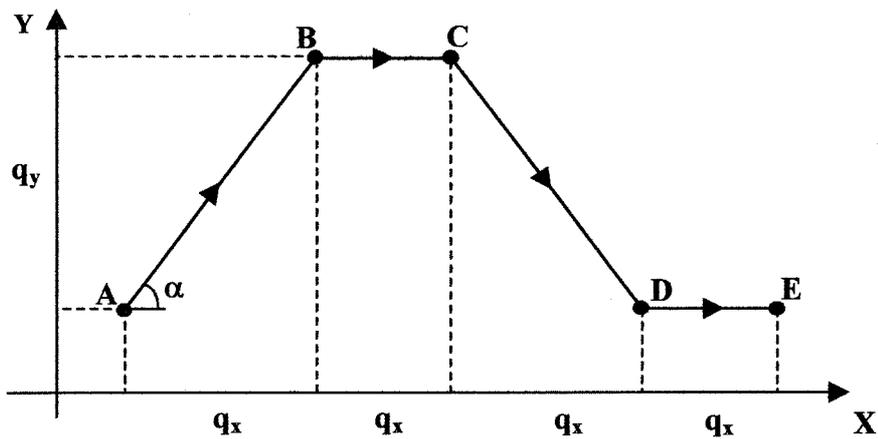


Ilustração 4-10: Trajetória trapezoidal

O movimento se inicia no ponto A, se desloca até o ponto B com velocidade nos dois eixos. Do ponto B ele se desloca até o ponto C com movimento apenas no eixo X. Novamente a trajetória se desenvolve no trecho \overline{CD} , agora com movimento nos dois eixos. Para finalizar é percorrido o trecho \overline{DE} , com movimento somente em X.

As curvas de velocidade, aceleração e deslocamento são mostradas respectivamente nas ilustrações 4.11, 4.12, 4.13.

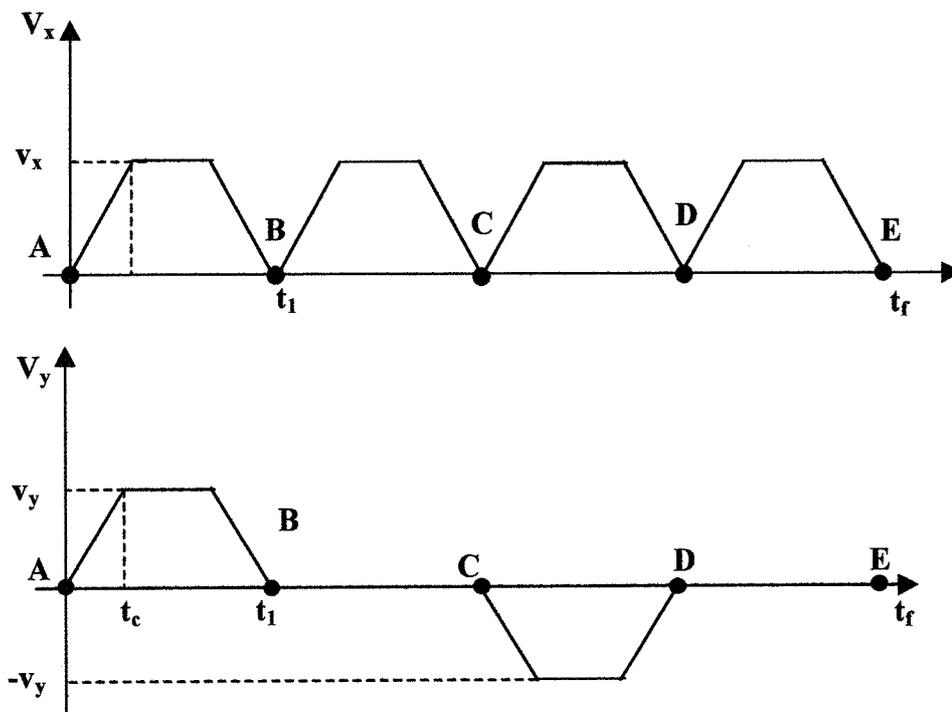


Ilustração 4-11: curvas de velocidade

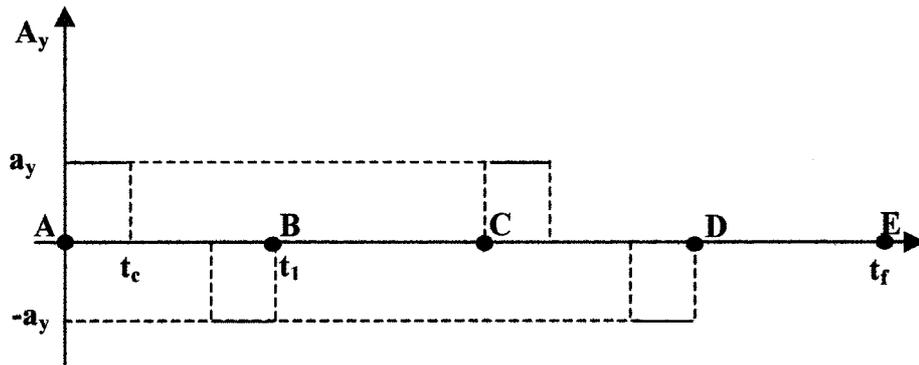
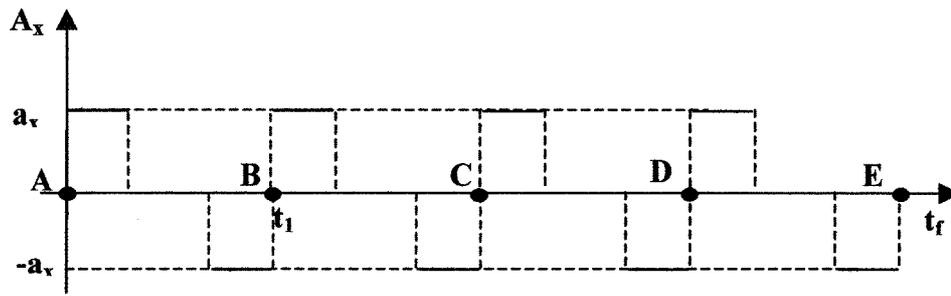


Ilustração 4-12: curvas de aceleração

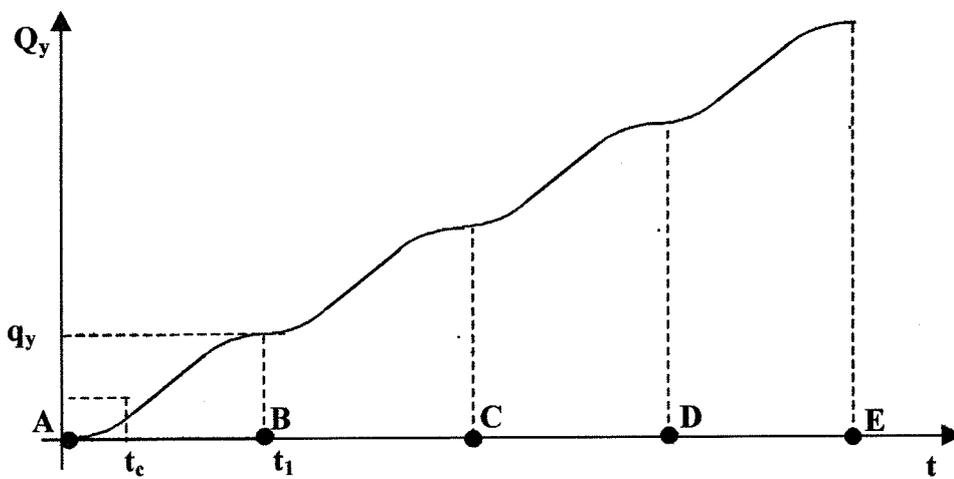
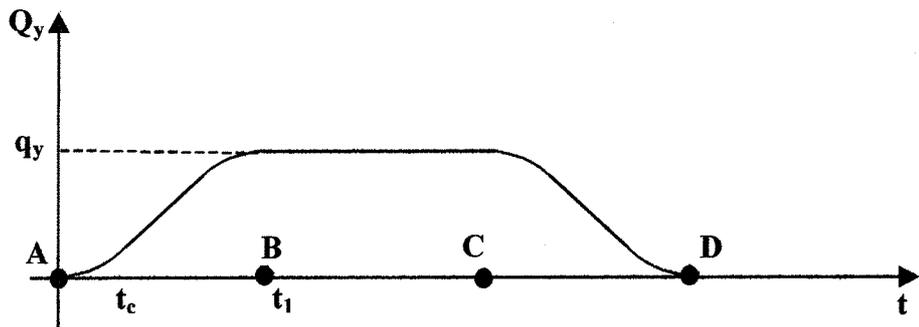


Ilustração 4-13: curvas de deslocamento

4.6 Conclusão

Utilizando-se de um perfil de velocidade trapezoidal, pôde-se facilmente planejar a trajetória de movimentos comuns em tecimento. Esta forma de velocidade facilita sua implementação no acionamento de motores de passo, pois permite que apenas uma rampa de aceleração seja calculada e armazenada para que se gere toda a curva de velocidade.

No próximo capítulo, será mostrada a implementação de funções de *software*, capazes de gerar as trajetórias aqui apresentadas no efetuator final do manipulador desenvolvido.

Capítulo V

O Desenvolvimento do Projeto

5.1 Introdução

Este capítulo tem como objetivo o relato do desenvolvimento do *software* de controle de trajetória do manipulador cartesiano. Aqui serão apresentados os fluxogramas dos processos relativos à construção dos perfis de velocidade trapezoidais, bem como aqueles responsáveis pelo correto sequenciamento do movimento dos eixos. O controle é realizado através de 1 byte proveniente da porta paralela que atua diretamente sobre os *drivers*. É também através da porta paralela que se dá a entrada de um sinal de alta frequência responsável pela geração dos diversos níveis de velocidade dos motores. Serão exploradas algumas particularidades referentes ao projeto de *software* e do *hardware* de que é composto o acionamento.

5.2 Ambiente de desenvolvimento

O projeto foi realizado utilizando-se o compilador Turbo C++ da Borland versão 3.0. Não foi empregada a programação para Windows, pois o objetivo é o de se transportar a plataforma definitiva para uma CPU industrial, viabilizando um comando via controle remoto. Este procedimento estaria mais de acordo com a realidade dos ambientes onde a soldagem e o corte são realizados, muitas vezes rústicos demais para o emprego de um computador pessoal. A teoria de referente à Engenharia de *Software* pode ser encontrada em [Pressman,1995].

Os testes foram realizados utilizando-se um computador Pentium 200 MHz. Enquanto a estrutura mecânica não estava disponível, verificou-se o estado dos sinais de controle através de um *proto-board* e alguns *leds*. A frequência de entrada foi obtida por um gerador de funções e as de saída foram avaliadas por um osciloscópio.

5.3 Considerações de *Hardware*

Como dito anteriormente, são os *drivers* que convertem a informação digital, proveniente de um processador qualquer, na corrente e tensão necessárias para movimentar o motor para uma posição preestabelecida, respeitando uma velocidade predeterminada.

Devido a necessidade de se acionar 2 motores, coordenando individualmente seus movimentos, foram utilizados dois *drivers* idênticos, desenvolvidos no laboratório.

O modo de excitação utilizado foi o de passo completo, pois satisfaz a precisão requerida nos processos de corte e soldagem, onde o equipamento será utilizado. Além disso, a excitação por micropasso não foi empregada, devido a limitações operacionais do *driver*.

Como uma das especificações do sistema é a de operar em alta velocidade, o emprego do micropasso necessitaria de frequências muito altas. Nessas frequências as correntes enviadas dos *drivers* ao motores sofrem alterações que causam perdas significativas de torque.

5.4 Sinais de controle

Para se acionar cada motor, oito sinais digitais devem ser enviados aos *drivers* (4 a cada um) via porta paralela, após serem determinados e atualizados, em tempo real, pela lógica do programa. São eles:

- *clock*;
- direção;
- libera;
- modo de velocidade.

5.4.1 O sinal de *Clock*

O *clock* é responsável pela velocidade de cada motor. A cada pulso, o motor gira o correspondente a um passo, que para o motor empregado é de 1.8° / passo (200 passos por volta). Dessa forma, a frequência com que esses pulsos são enviados ao *driver*, corresponde à taxa com que o motor realizará uma volta completa. O *clock* pode ser representado por uma onda quadrada, ou mesmo pela frequência no qual se *seta* e se *reseta* um bit.

5.4.2 O sinal de direção

Para a direção utiliza-se apenas um bit. É ele quem comandará a sequência pela qual as fases do motor serão acionadas. Geralmente, uma mudança de direção deve ser realizada em conjunto com a parada do motor para que se evite a perda de passos. Quando alto, o sinal de direção possibilita que o motor se mova no sentido anti-horário, quando baixo, o sinal de direção possibilita ao motor movimento no sentido horário.

5.4.3 O sinal de libera

O bit de libera quando ativo possibilita que o motor seja acionado e freado quando o sinal de *clock* está ausente. Quando não ativo deixa o motor com movimento livre, podendo ser posicionado manualmente. Esta operação não é recomendada, pois se o motor é movimentado por forças externas funciona como um gerador, podendo danificar o *driver*.

5.4.4 O sinal de modo de velocidade

O bit de modo de velocidade deve ser mantido baixo sempre que a velocidade for inferior a 0,5 cm/s, e alto quando superior. Isso permite uma certa segurança no motor, regulando o valor da corrente empregada.

5.5 As frequências de *clock*

O computador deve fornecer dois sinais em frequências diferentes para que se possa gerar o *clock* relativo ao comando de cada *driver*. Isso foi possibilitado dividindo-se, via *software*, um sinal de alta frequência F (20 kHz) injetado no PC, via porta paralela, nas várias frequências correspondentes às velocidades desejadas.

A maior limitação deste procedimento é a de se poder obter resolução de velocidade de 0.1cm/s somente para velocidades inferiores a 2.3 cm/s para o eixo X e 4.4 cm/s para o eixo Y, além de existirem velocidades associadas a altas frequências impossíveis de se atingir em ambos os eixos. De fato, cada frequência elementar f_{el} associada a uma determinada velocidade pode ser calculada como:

$$f_{el} = \frac{F}{n} = \frac{20\text{Khz}}{n}$$

onde n é um número inteiro maior ou igual a 1, que indica por quanto a frequência de entrada será dividida. Logo, sendo f_{xel} e f_{yel} as frequências elementares em X e Y, a velocidade de cada motor pode ser expressa em cm/s por:

$$V_x = \frac{\text{Deslocamento por volta em X}}{\text{Período por volta em X}} = \frac{4}{200 * \frac{1}{f_{xel}}} = \frac{4 \times 20000}{200n} = \frac{400}{n_x} \text{ mm/s}$$

$$V_y = \frac{\text{Deslocamento por volta em Y}}{\text{Período por volta em Y}} = \frac{16}{200 * \frac{1}{f_{yel}}} = \frac{16 \times 20000}{200n} = \frac{1600}{n_y} \text{ mm/s}$$

Demonstra-se a seguir duas tabelas, exibindo o comportamento para V_x e para V_y . Nota-se que para altas frequências elementares, existem velocidades que não podem ser atingidas.

Divisor (n_x)	Frequência elementar f_{xel}	Velocidade em X (mm/s)
1	20000	400,00
2	10000	200,00
3	6666,66	133,33
4	5000	100,00
5	4000	88,00
6	3333,33	66,66
7	2857,14	57,14
8	2500	50,00
9	2222,22	44,44
10	2000	40,00
11	1818,18	36,36
12	1666,66	33,33
13	1538,46	30,76
14	1428,57	28,57
15	1333,33	26,66
16	1250	25,50
17	1176,47	23,52
18	1111,11	22,22
19	1052,63	21,05
20	1000	20,00

Divisor(n_y)	Frequência elementar f_{yel}	Velocidade em Y(mm/s)
1	20000	1600,00
2	10000	800,00
3	6666,66	533,33
4	5000	400,00
5	4000	320,00
6	3333,33	266,67
7	2857,14	228,57
8	2500	200,00
9	2222,22	177,78

10	2000	160,00
11	1818,18	145,45
12	1666,66	133,33
13	1538,46	123,07
14	1428,57	114,28
15	1333,33	106,66
16	1250	100,00
17	1176,47	94,11
18	1111,11	88,89
19	1052,63	84,21
20	1000	80,00
21	952,38	76,19
22	909,090	72,72
23	869,56	69,56
24	833,33	66,67
25	800	64,00
26	769,23	61,53
27	740,74	59,25
28	714,28	57,14
29	689,65	55,17
30	666,66	53,33
31	645,16	51,61
32	625	50,00
33	606,06	48,48
34	588,23	47,05
35	571,42	45,71
36	555,55	44,44
37	540,54	43,24
38	526,31	42,10

Fica evidente, ao se comparar ambas as tabelas que para uma mesma velocidade, o eixo X necessita de uma frequência de excitação maior, devido ao seu maior deslocamento por volta em relação ao eixo Y.

5.6 As rotinas de Software

Para se realizar o controle dos motores, conforme já esclarecido anteriormente, é necessário enviar à porta paralela 1 byte de informação. Criou-se então rotinas de *software* para controlar os bit de direção, *clock*, libera e modo de velocidade de ambos os motores.

Cada sinal de saída da porta paralela (378) possui uma função específica, sendo o bit menos significativo o referente ao Clock X e o mais significativo o Libera Y. Estes sinais compõem o byte denominado `dado_saida`, que pode apresentar 256 possíveis combinações de bit.

Pino 9	Pino 8	Pino 7	Pino 6	Pino 5	Pino 4	Pino 3	Pino 2
LiberaY	M.VelY	DireçãoY	ClockY	LiberaX	M.VelX	DireçãoX	ClockX

Ilustração 5-1: disposição dos bits na porta paralela

Assim, definindo-se a porta de saída como 378 (`porta_saida = 0x378`) e *setando* inicialmente o dado de saída como FF (`dado_saida = 0xFF`), pode-se enviar bits de controle da porta paralela através da instrução `outportb`.

Será explicado o funcionamento de cada uma das funções utilizadas para controlar a saída destes bits.

5.6.1 As funções de que geram um pulso de *Clock*

As funções que enviam um único pulso de *clock* para o *driver* X e para o *driver* Y são respectivamente `pulsox()` e `pulsoy()`.

Elas funcionam anulando os bits de `clockX` e de `clockY`, e em seguida, dentro da mesma função, retornando-os a 1. Esta operação é realizada sem que os outros bit não sejam alterados.

A função `pulsox()`, responsável pelo pulso de *clock* em `clockX`, funciona realizando um E lógico entre `dado_saida` e o valor FE H que anula o bit associado a `clockX`. Em seguida, o `dado_saida` é enviado à porta paralela através da instrução `outportb`. Agora é realizado um OU lógico entre `dado_saida` e o valor 01 H para retornar o bit de `clockX` a 1.

A função `pulsox()`, implementada em linguagem C, é apresentada abaixo:

```

void pulsox ( )
{
    dado_saida=dado_saida & 0xfe;
    outportb(porta_saida,dado_saida);
    dado_saida=dado_saida | 0x01;
    outportb(porta_saida,dado_saida);
}

```

Do mesmo modo, a função pulsoy (), responsável pelo pulso de clock em clockY, funciona realizando um E lógico entre dado_saida e o valor EF H que anula o bit associado a clockY. Em seguida, o dado_saida é enviado à porta paralela através da instrução outportb. Agora é realizado um OU lógico entre dado_saida e o valor 10 H para retornar o bit de clockY a 1.

A função pulsoY (), implementada em linguagem C, é apresentada abaixo:

```

void pulsoy( )
{
    dado_saida=dado_saida & 0xef;
    outportb(porta_saida,dado_saida);
    dado_saida=dado_saida | 0x10;
    outportb(porta_saida,dado_saida);
}

```

5.6.2 As funções de controle dos sinais de direção

O bit que controla a direção do eixo X é gerenciado por duas funções: a dirxpos() e a dirxneg().

A função dirxpos() seta o bit de direção do eixo X em 1, realizando um OU lógico entre dado_saida e o valor 02 H.

```

void dirxpos( )
{
    dado_saida=dado_saida | 0x02;
    outportb(porta_saida,dado_saida);
}

```

Analogamente, a função `dirxneg()` *seta* o bit de direção do eixo X em 0, realizando um E lógico entre `dado_saida` e o valor FD H.

```

void dirxneg( )
{
    dado_saida=dado_saida & 0xfd;
    outportb(porta_saida,dado_saida);
}

```

O bit que controla a direção do eixo Y é gerenciado por duas funções: a `dirypos()` e a `diryneg()`.

A função `dirypos()` *seta* o bit de direção do eixo Y em 1, realizando um OU lógico entre `dado_saida` e o valor 20 H.

```

void dirypos( )
{
    dado_saida=dado_saida | 0x20;
    outportb(porta_saida,dado_saida);
}

```

Analogamente, a função `diryneg()` *seta* o bit de direção do eixo Y em 0, realizando um E lógico entre `dado_saida` e o valor DF H.

```

void diryneg( )
{
    dado_saida=dado_saida & 0xdf;
    outportb(porta_saida,dado_saida);
}

```

5.6.3 As funções de controle dos bits de modo de velocidade

O sinal que controla o modo de velocidade do eixo X é gerenciado por duas funções: a `modxpos()` e a `modxneg()`.

A função `modxpos()` *seta* o bit do modo de velocidade do eixo X em 1, realizando um OU lógico entre `dado_saida` e o valor 04 H.

```
void modxpos( )
{
    dado_saida=dado_saida | 0x04;
    outportb(porta_saida,dado_saida);
}
```

Analogamente, a função `modxneg()` *seta* o bit de modo de velocidade do eixo Y em 0, realizando um E lógico entre `dado_saida` e o valor FB H.

```
void modxneg( )
{
    dado_saida=dado_saida & 0xfb;
    outportb(porta_saida,dado_saida);
}
```

A função `modypos()` *seta* o bit do modo de velocidade do eixo X em 1, realizando um OU lógico entre `dado_saida` e o valor 40 H.

```
void modypos( )
{
    dado_saida=dado_saida | 0x40;
    outportb(porta_saida,dado_saida);
}
```

Analogamente, a função `modyneg()` *seta* o bit do modo de velocidade do eixo Y em 0, realizando um E lógico entre `dado_saida` e o valor BF H.

```
void modyneg ( )
{
    dado_saida=dado_saida & 0xbf;
    outportb(porta_saida,dado_saida);
}
```

5.6.4 As funções de controle dos bits de Libera

O sinal que controla a liberação do motor no eixo X é gerenciado por duas funções: a `liberax()` e a `n_liberax()`.

A função `liberax()` *seta* o bit que libera o motor no eixo X, em 1, realizando um OU lógico entre `dado_saida` e o valor 08 H.

```
void liberax( )
{
    dado_saida=dado_saida | 0x08;
    outportb(porta_saida,dado_saida);
}
```

Analogamente, a função `n_liberax()` *seta* o bit que libera o motor no eixo Y em 0, realizando um E lógico entre `dado_saida` e o valor F7 H.

```
void n_liberax( )
{
    dado_saida=dado_saida & 0xf7;
    outportb(porta_saida,dado_saida);
}
```

A função `liberay()` seta o bit que libera o motor no eixo Y, em 1, realizando um OU lógico entre `dado_saida` e o valor 80 H.

```
void liberay( )
{
    dado_saida=dado_saida | 0x80;
    outportb(porta_saida,dado_saida);
}
```

Analogamente, a função `n_liberax()` seta o bit que libera o motor no eixo Y em 0, realizando um E lógico entre `dado_saida` e o valor 7F H.

```
void n_liberay( )
{
    dado_saida=dado_saida & 0x7f;
    outportb(porta_saida,dado_saida);
}
```

5.7 Geração dos sinais de *Clock*

Cada vez que um pulso do sinal de alta frequência F é lido através da porta paralela, uma interrupção de *software* é gerada. Esta interrupção incrementa duas variáveis inteiras `freqx` e `freqy`.

Seta-se um valor desejado de frequências elementares para X e para Y através de n_x e n_y . Quando o valor de `freqx` é igual a n_x , um pulso de *clock* é gerado pela função `pulsox()`, e a variável `freqx` é zerada em seguida. De modo análogo para Y, quando o valor de `freqy` é igual a n_y , um pulso de *clock* é gerado por `pulsoy()`, e a variável `freqy` é zerada.

Dessa maneira, é possível dividir a frequência de entrada em duas frequências elementares f_{xel} e f_{yel} , proporcionais aos coeficientes n_x e n_y , gerando dois sinais de *clock* que acionarão os motores correspondentes.

Fica evidente que, alterando-se convenientemente o valor de n_x e n_y , pode-se controlar a velocidade dos dois motores. Assim, calculando-se e tabelando-se valores de n_x e n_y de acordo com uma trajetória preestabelecida, pode-se a cada passo dos motores, atualizar o valor da velocidade de cada eixo. Para efetivar o controle da trajetória, os bits de direção são alterados de acordo com condições específicas de cada movimento e sempre nos momentos onde a velocidade do eixo correspondente é nula.

Adiante será mostrado um fluxograma deste procedimento. Primeiramente será tratado o cálculo do perfil de velocidade.

5.8 Cálculo e armazenamento da velocidade

Conforme já explorado, o perfil de velocidade adotado para ambos os eixos nas trajetórias especificadas foi o trapezoidal. Assim, se faz necessário especificar a velocidade associada a cada passo dado pelo motor. Como os movimentos implementados são periódicos e de pouca complexidade, pode-se calcular e tabelar apenas uma única rampa de aceleração para cada eixo e, de acordo com a trajetória coordenar via *software* o modo como as direções e velocidades são *setadas*.

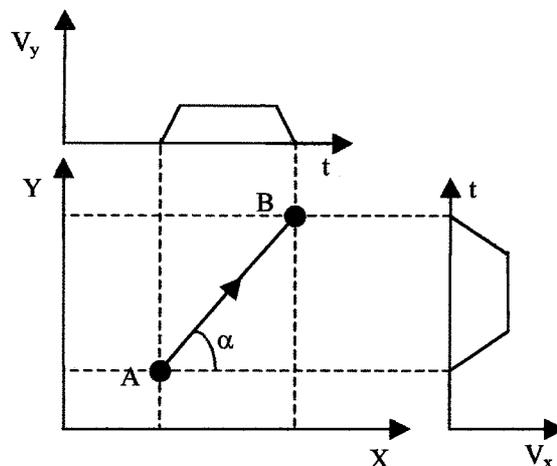


Ilustração 5-2: Trajetória, sentido A→B

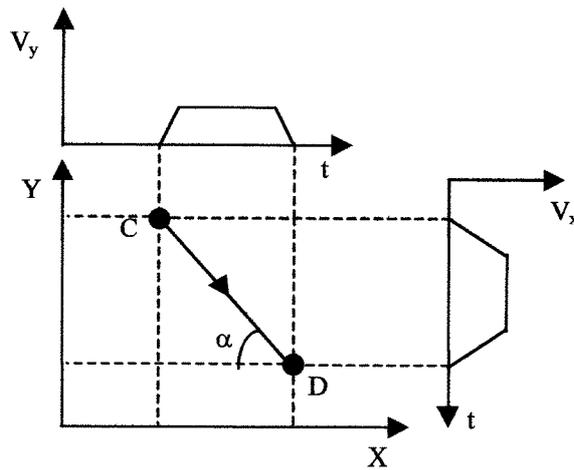


Ilustração 5-2: Trajetória, sentido C→D

Através das ilustrações 5-2 e 5-3, é fácil notar que os perfis de velocidade mantêm-se os mesmos, ainda que a direção e sentido da trajetória seja diferente.

Dispondo de valores como $dist_x$ e $dist_y$ (distância percorrida em um perfil de velocidade trapezoidal), a velocidade e o tipo da trajetória desejada, pode-se facilmente calcular as curvas de velocidade. Desejando-se calcular e armazenar um perfil de velocidade do eixo Y análogo ao abaixo,

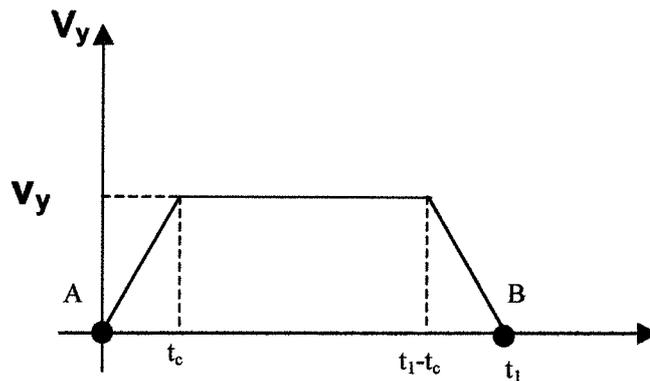


Ilustração 5-3: perfil de velocidade trapezoidal

com velocidade máxima v_y e cuja área representa a distância $dist_y$, é necessário especificar:

1) O número total de passos para percorrer $dist_y$ ($numpassy_t$) é obtido através da seguinte relação:

$$\text{numpassyt} = \frac{\text{disty}}{\text{precisaoy}}$$

2) Já a distância *disty1* percorrida em movimento acelerado e associada ao tempo t_c pode ser obtida por:

$$\text{disty1} = \frac{v_y^2}{a_y}$$

onde v_y^2 representa o quadrado da velocidade máxima v_y e a_y , a aceleração previamente calculada.

Pode-se então compor uma tabela com a velocidade e o divisor n_y referente a uma rampa de aceleração a_y , de acordo com o fluxograma referente à rotina de calculo_ay.

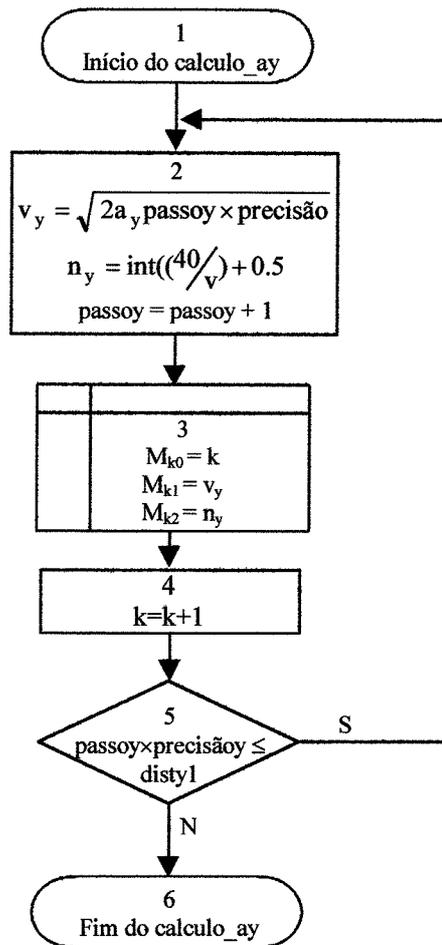


Ilustração 5-4: fluxograma da rotina calculo_ay

5.9 Rotina de Geração das frequências elementares

Pode-se gerar duas frequências diferentes de *clock* de acordo com o fluxograma abaixo:

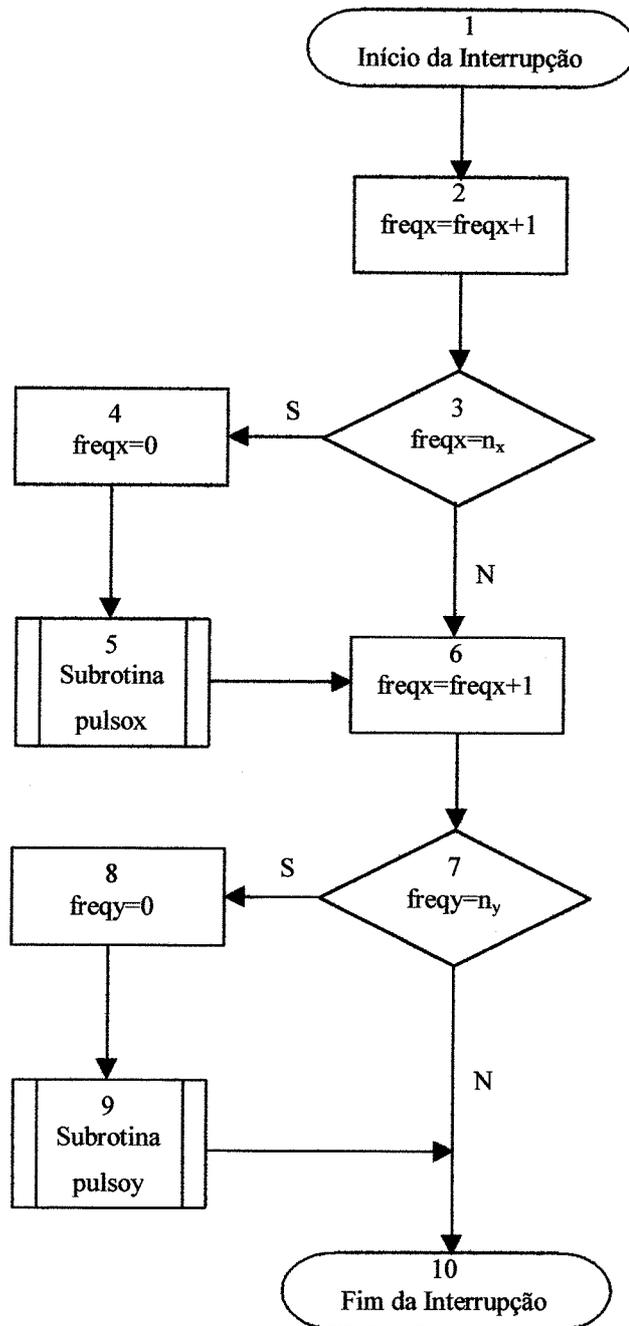


Ilustração 5-5: fluxograma da rotina de geração de frequências elementares

A interrupção ocorre numa frequência de 20 kHz. Existe então, tempo suficiente para a CPU realizar outras tarefas como: exibir valores internos e fazer alguns cálculos secundários.

Os bits de direção são mudados de acordo com condições específicas de cada movimento e sempre nos instantes onde a velocidade do eixo correspondente é nula.

O fluxograma apresentado não altera os valores de n_x e n_y , de modo que um controle da velocidade não pode ser viabilizado. Nem mesmo altera o sentido de rotação dos motores. Modificações devem ser realizadas no fluxograma para que se possa implementar uma lógica de controle.

5.10 O Controle de trajetória

O controle de n_x e n_y é realizado através de dois blocos que atualizam, de acordo com o *clock* dos respectivos motores, o valor de n_x e n_y : controle do eixo x e controle do eixo y . Estes blocos encontram-se inseridos dentro do procedimento de interrupção. Além desta atualização, *setam* ou *resetam* os bits de direção conforme a trajetória.

Uma rotina de controle de um eixo pode ser subdividido em mais duas subrotinas. Uma subrotina de controle de velocidade, que faz com que a curva de velocidade seja representada a partir da rampa de aceleração já armazenada, e outra subrotina de seguimento de caminho, que altera o valor dos bits de direção e habilita ou não o movimento de um determinado eixo. É essa função que determina o tipo de trajetória, de modo que ela difere de movimento a movimento.

Mostra-se, em seguida, o fluxograma já modificado do processo de interrupção no qual se encontra inserido o processo pertinente ao controle de trajetória de ambos os eixos.

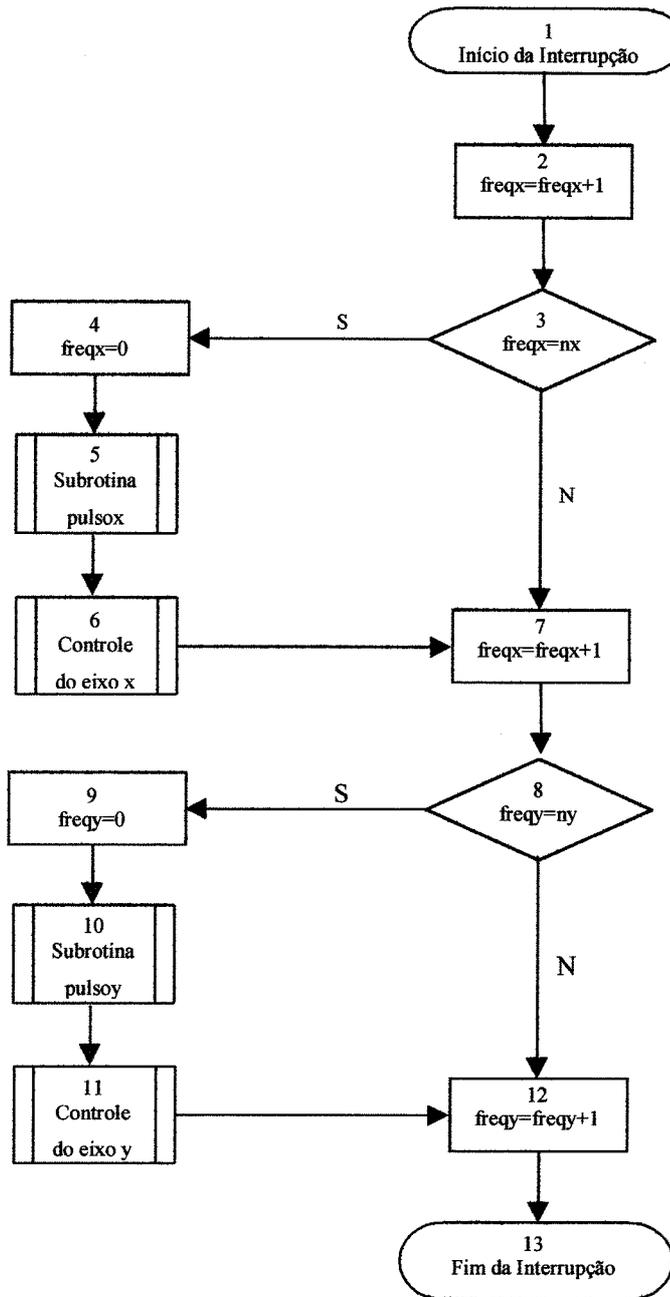


Ilustração 5-6: fluxograma da interrupção com rotinas de controle dos eixos X e Y

5.10.1 O controle de velocidade

O controle de velocidade é realizado por uma das 2 subrotinas que constituem as rotinas de controle dos eixos X e Y. É ela que, a partir da rampa de aceleração já calculada e armazenada, cria o perfil de velocidade trapezoidal adotado.

Sua lógica utiliza-se basicamente do número de passos total para se realizar uma trajetória (numpassyt ou numpassxt) e da quantidade de passos já percorrida (passox e passoy). Será mostrado o fluxograma relativo a rotina de controle de velocidade para o eixo y, denominada Velocidadey. A Rotina de controle para o eixo X é análoga.

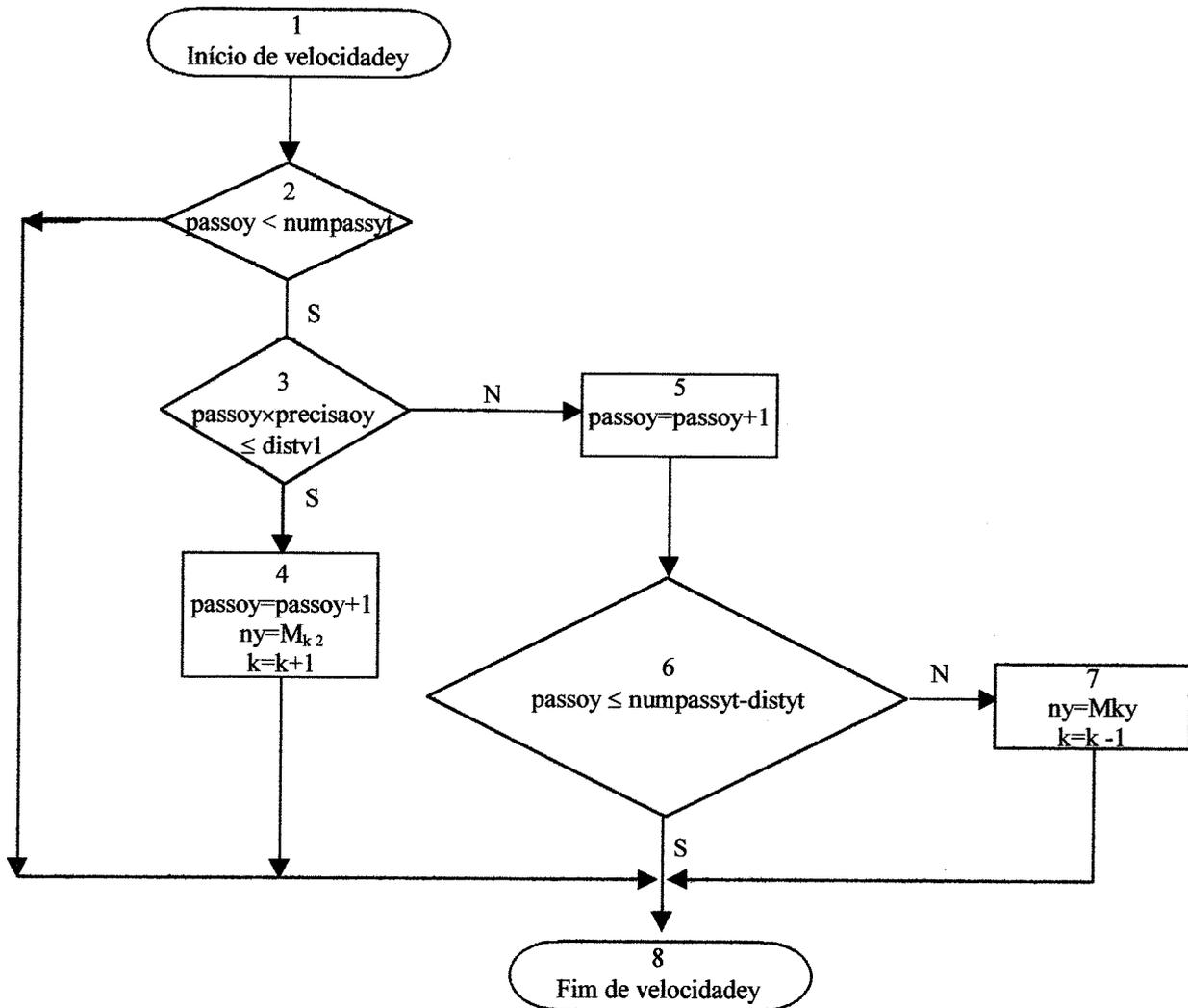


Ilustração 5-7: fluxograma da geração do perfil trapezoidal de velocidade

5.10.2 O seguimento de caminho

O seguimento de caminho é realizado por uma rotina que se difere de trajetória a trajetória. É nela em que são modificados os bits de direção e onde é habilitado ou não o movimento em um dos eixos.

O movimento é controlado pelo estado de 4 variáveis: *controldirx*, *controldiry*, *controlvelx* e *controlvely*. Conforme o estado de cada uma, o motor pode girar num sentido ou em outro, como também pode não se movimentar.

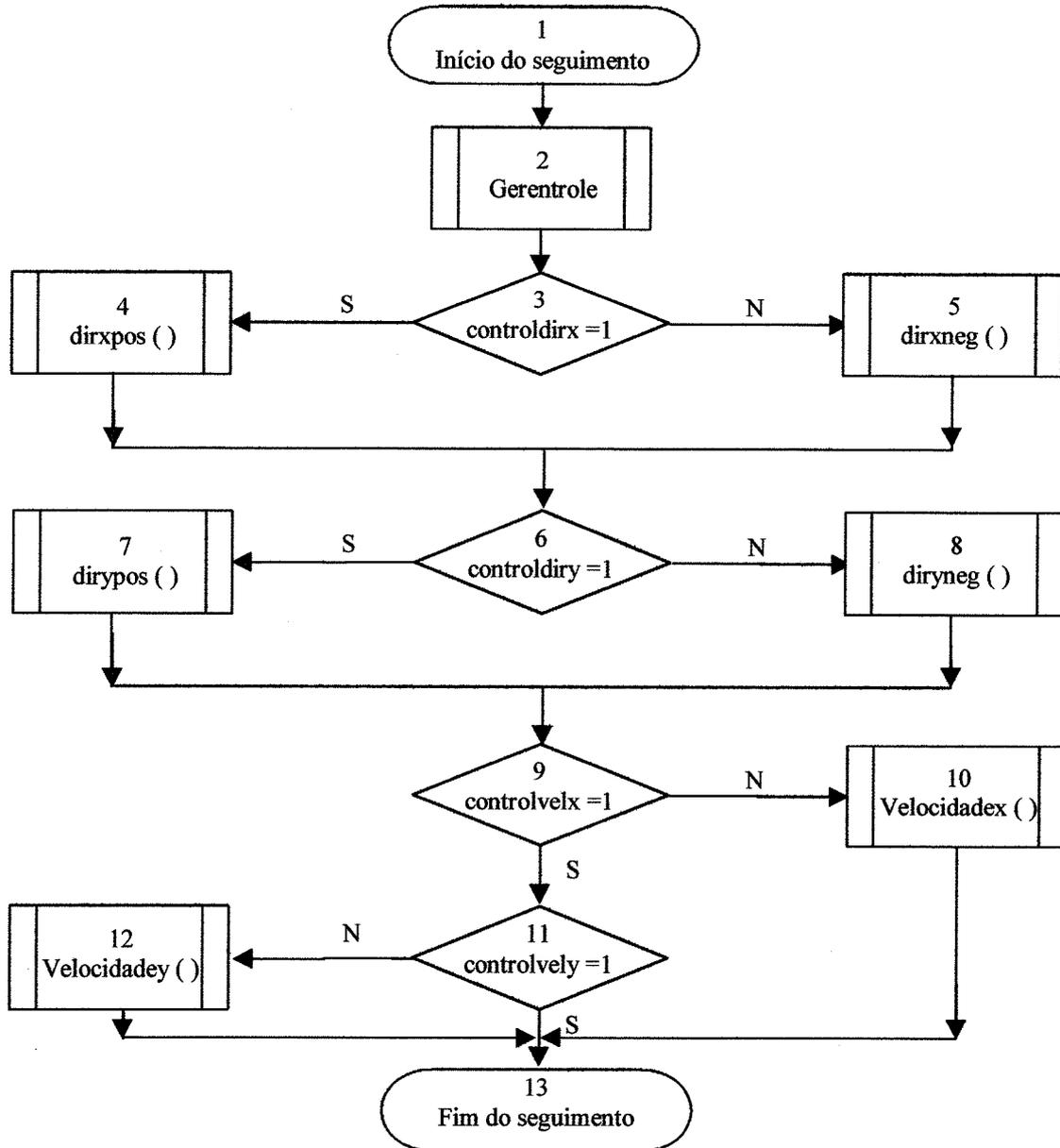


Ilustração 5-8: fluxograma da rotina de controle dos movimentos

A combinação dessas variáveis é estabelecida pela rotina interna gerentrole. É esta rotina que, levando em consideração o aspecto de cada trajetória e o número de passos já realizados, *seta* a direção e habilita ou não o movimento em cada eixo. Ela também atualiza os valores de variáveis como *disty1*, *disty2*, *numpassxt* e *numpassyt*.

Assim, um movimento de tecimento triangular pode ser gerado pelo seguinte fluxograma:

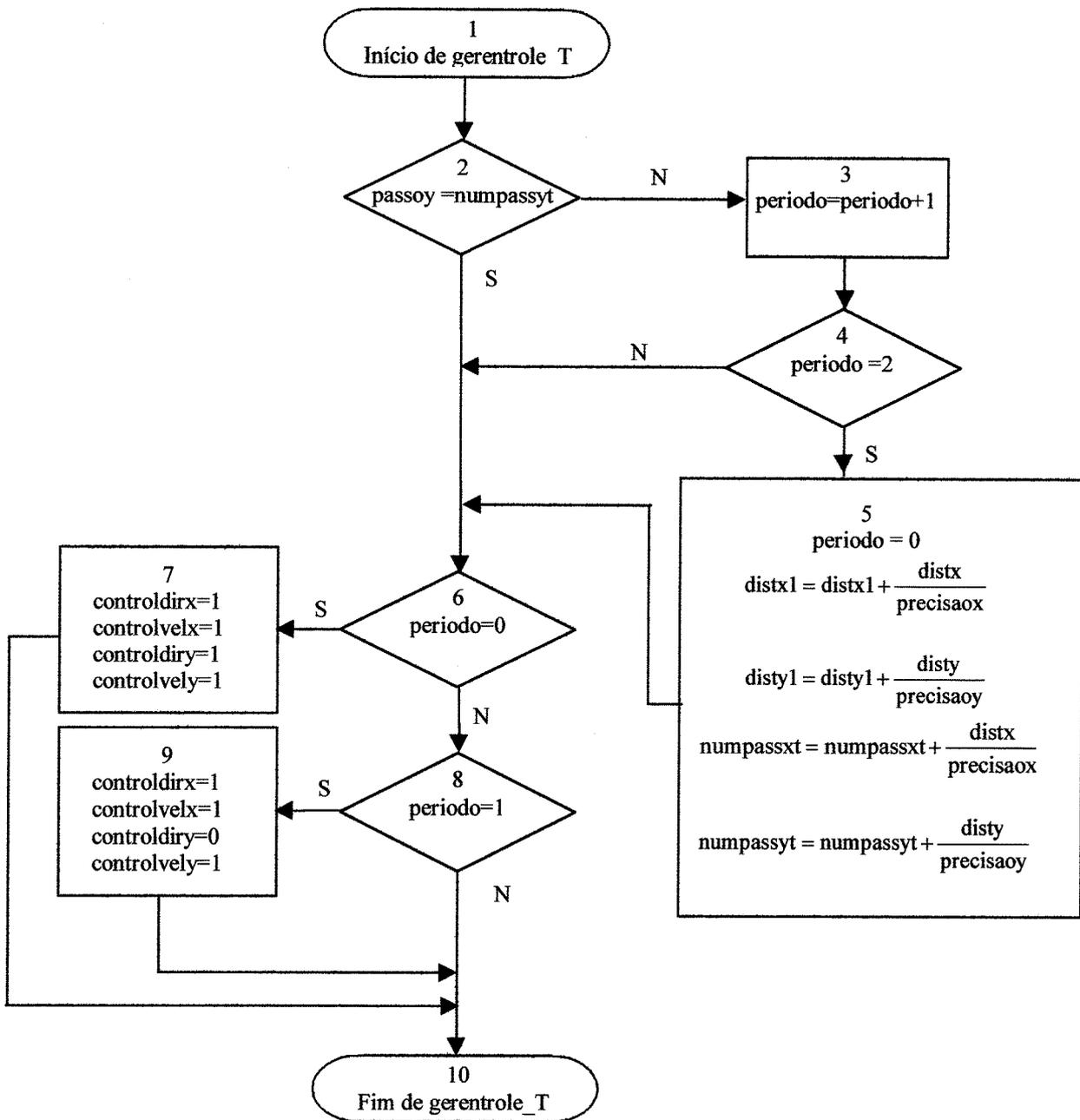


Ilustração 5-10: lógica da forma de tecimento triangular

Onde *período* é a variável que registra cada subperíodo de um movimento. Ela é incrementada no momento em que um dos eixos atingiu velocidade nula. Como ambos os eixos, nos movimentos estudados, anulam sua velocidade ao mesmo tempo, pode-se associar a mudança de valor da variável período a qualquer uma das variáveis *numpassxt* e *numpassyt*.

A forma de tecimento quadrada é comandada pela seguinte lógica:

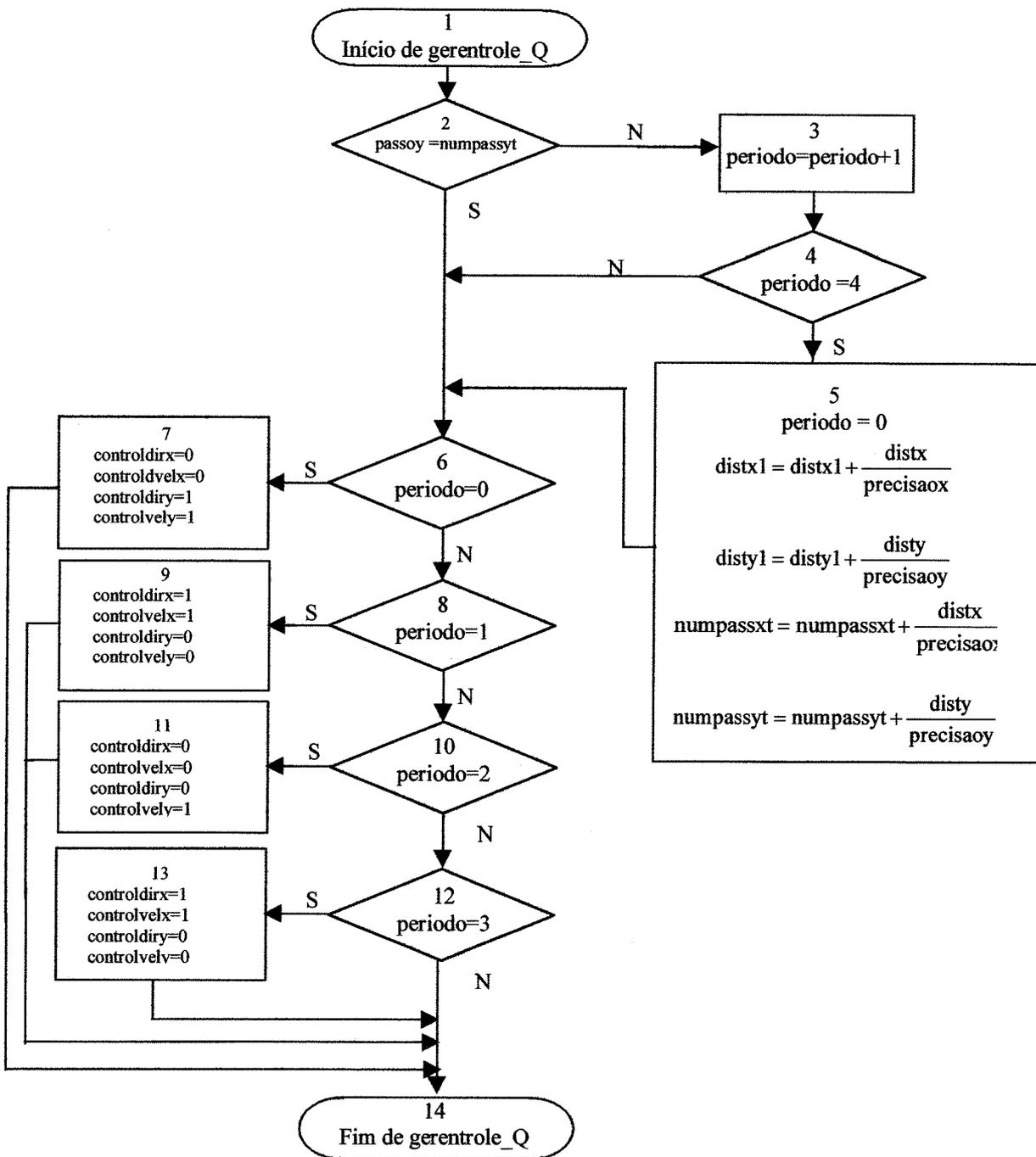


Ilustração 5-12: lógica da forma de tecimento quadrada

Analogamente, o tecimento trapezoidal também é representado pelo seguinte fluxograma:

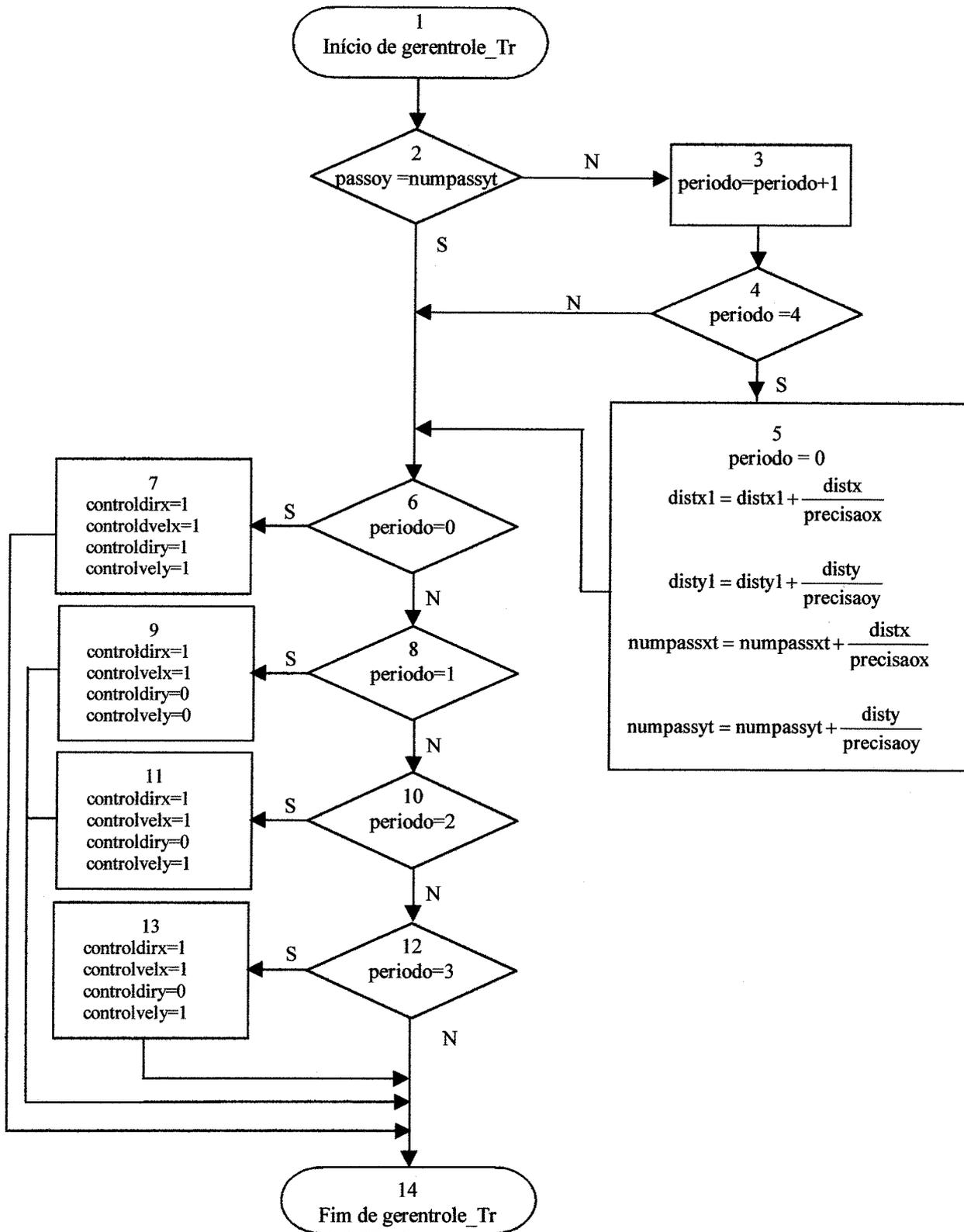


Ilustração 5-13: lógica da forma de tecimento trapezoidal

5.11 Conclusão

Este capítulo abordou a implementação via *software* de lógicas de controle de trajetórias comuns em movimentos de tecimento.

A construção de um protótipo do sistema mecânico possibilitou a realização de ensaios experimentais das funções de controle implementadas. Estes testes serão apresentados no capítulo seguinte, bem como a análise dos resultados obtidos.

Capítulo VI

Resultados Obtidos

6.1 Introdução

Neste capítulo são apresentados os resultados experimentais obtidos a partir da implementação de um sistema de controle de trajetória, cujo projeto foi tratado no capítulo anterior. Os testes foram realizados em um protótipo do sistema mecânico acionado por dois motores de passo.

As trajetórias foram traçadas em papel milimetrado, com emprego de uma caneta esferográfica presa ao efetuador final. As velocidades dos eixos X e Y, associadas aos divisores n_x e n_y , foram exibidas em tempo real, na tela do computador.

6.2 O Protótipo mecânico

Os testes foram realizados em um protótipo de um manipulador cartesiano desenvolvido no laboratório que, conforme abordado no capítulo III, consiste na versão I do Tartílope acrescida de um segundo eixo. Este acréscimo, possibilita à versão II realizar movimentos axiais e transversais, representando um mecanismo dois graus de liberdade.

A estrutura do segundo eixo é mostrada nas ilustrações 6-1 e 6-2, ficando em evidência o sistema, de correia dentada e polia, acionado pelo motor de passo.

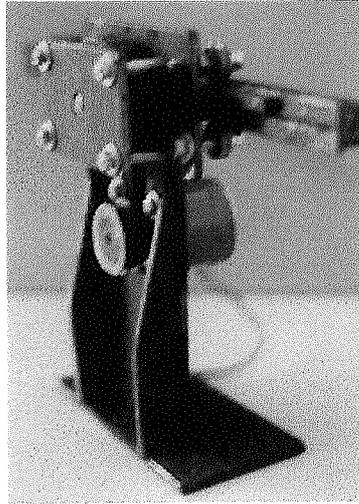


Ilustração 6-1: O eixo transversal do Tartílope versão II (visão lateral)

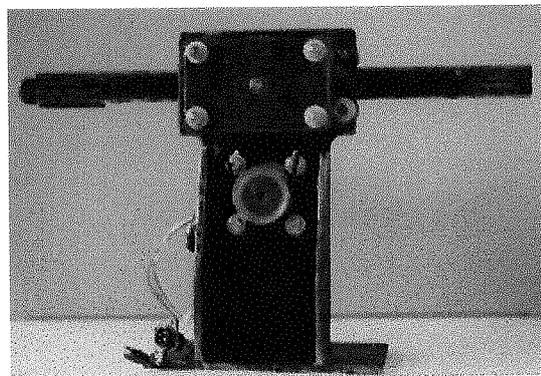


Ilustração 6-2: O eixo transversal do Tartílope versão II (visão frontal)

Este segundo eixo foi fixado na parte superior do Tartílope versão I para a realização das trajetórias de tecimento.

O protótipo do sistema mecânico, onde se realizaram os testes, é exibido nas ilustrações 6-3 e 6-4. Na figura 6-3 tem-se o manipulador mecânico,

juntamente com o sistema de acionamento. Já na 6-4, são destacados os dois eixos de movimento: o trilho de movimento axial e o sistema mecânico de movimento transversal.

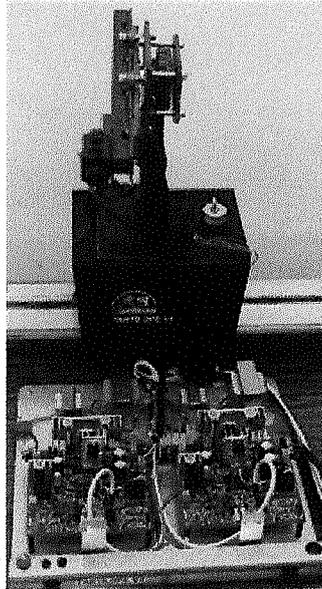


Ilustração 6-3: Sistema mecânico e acionamento da versão II

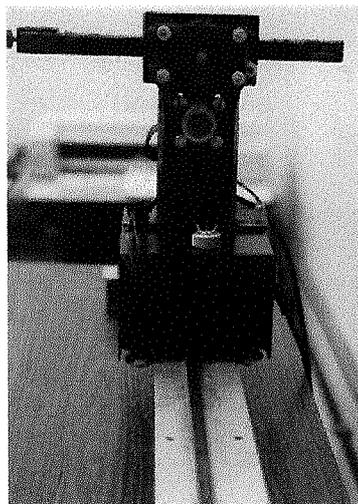


Ilustração 6-4: Sistema mecânico sobre trilho de movimento axial

6.3 A interface do Software

Por não consistir em um objetivo primário do projeto, não houve preocupação da interface do *software* com o usuário. Dados como a velocidade e a posição do efetuador final relativos a cada eixo, bem como o tempo da trajetória em segundos, são mostrados em campos definidos da tela do computador. Um esboço do caminho traçado também é exibido.

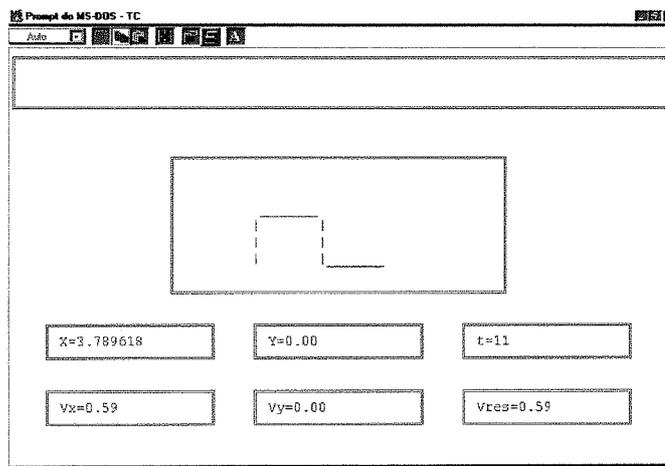


Ilustração 6-5: Tela do computador que exhibe a saída dos dados

6.4 Ensaios realizados

As trajetórias das formas de tecimento tratadas neste trabalho foram traçadas em papel milimetrado, onde pôde-se analisar se o caminho percorrido pelo efetuador final estava de acordo com o desejado .

Os ensaios foram realizados utilizando-se uma caneta esferográfica fixada ao atuador final conforme mostra a ilustração 6-6.

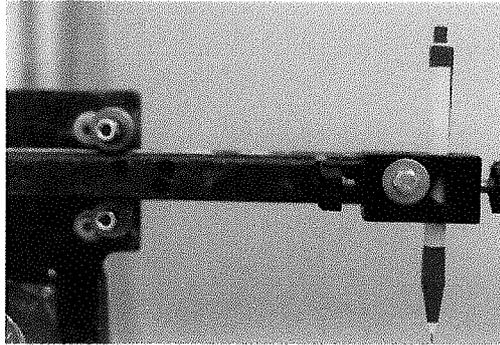


Ilustração 6-6: Esferográfica presa ao efetuador final para a realização de testes

Como a velocidade de cada motor é calculada dividindo-se a frequência de entrada ($F=20$ kHz) por um número inteiro, este valor tem seu resultado o mais próximo possível da realidade se F é bem regulada.

Além disso, se a velocidade de trajetória escolhida pelo usuário não possuir valores associados de n_x e n_y inteiros, estes serão aproximados aos inteiros mais próximos. Isso indica que, muitas vezes a velocidade da trajetória desejada será diferente da mostrada: esta diferença para velocidades inferiores a 2 cm/s não ultrapassa 2%. O que garante que a velocidade exibida tem seu valor bastante próximo da realidade.

6.4.1 A trajetória de período quadrado

Esta trajetória foi traçada em papel milimetrado de acordo com os seguintes parâmetros:

- velocidade de curso de 1 cm/s;
- período quadrado de tamanho 2 cm;

O traçado é mostrado na ilustração 6-7.

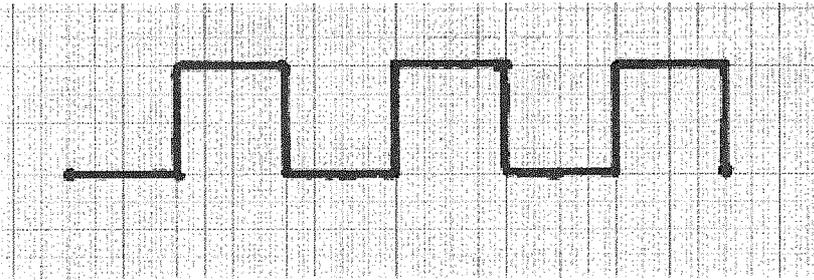


Ilustração 6-7: trajetória traçada pelo Tartilope: período quadrado

A velocidade da trajetória correspondeu a desejada, com uma velocidade de 1.00 relativa a cada eixo.

6.4.2 A trajetória de período triangular

Da mesma maneira, traçou-se a trajetória triangular, de acordo com os seguintes parâmetros:

- velocidade de curso de 1 cm/s;
- período triangular de base igual a 2 cm e altura igual a 2 cm;

O traçado é mostrado na ilustração 6-8.

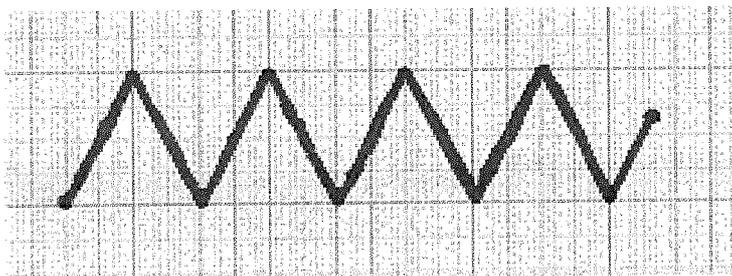


Ilustração 6-8: trajetória traçada pelo Tartilope: período triangular

A velocidade da trajetória não apresentou diferença da desejada, tendo uma velocidade associada ao eixo X de 0.71cm/s e 0.71cm/s para Y, de modo que a velocidade resultante foi de 1.00 cm/s.

6.4.3 A trajetória de período trapezoidal

Este movimento apresentou problemas durante os ensaios realizados de modo que não será mostrada a trajetória resultante.

6.4.4 A trajetória linear

Embora não explorada nos capítulos anteriores, a trajetória linear foi desenvolvida para que o Tartilope versão II pudesse realizar também os movimentos da versão I.

Sua lógica consiste em acionar apenas um dos eixos (X), segundo um perfil trapezoidal. Um deslocamento de 9 cm, realizado com uma velocidade de 1 cm/s é exibido na ilustração 6-9. A velocidade da trajetória correspondeu ao valor desejado.

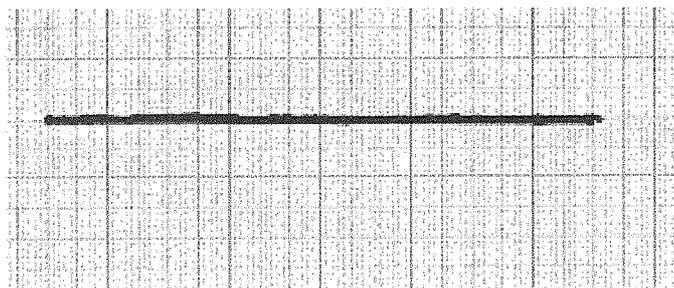


Ilustração 6-9: trajetória traçada pelo Tartilope: movimento linear

6.5 Conclusão

Pelos resultados apresentados neste capítulo, pôde-se verificar o correto funcionamento do manipulador cartesiano através das trajetórias obtidas. As perdas de passos foram minimizadas pela adoção de um perfil de velocidade trapezoidal, pois as mudanças no sentido de rotação dos motores foram realizadas somente em velocidades nulas.

Os resultados são bastante satisfatórios para a precisão requerida em movimentos de tecimento, apesar das folgas existentes no protótipo mecânico.

A lógica de controle da trajetória trapezoidal será reavaliada, visando encontrar e corrigir o problema apresentado.

Capítulo VII

Conclusões e Perspectivas

Neste trabalho foi apresentado o estudo, desenvolvimento e implementação do Controle de Trajetória de um Manipulador Cartesiano de Dois Eixos. A idéia básica consistia em verificar a viabilidade da técnica empregada, fundamentada na divisão via software de um sinal de alta frequência para gerar as várias velocidades de rotação dos motores de passo.

A estrutura mecânica e de acionamento do manipulador se baseou em um sistema de deslocamento linear da tocha de soldagem, anteriormente desenvolvido no laboratório: o Tartilope versão I. Foi acrescido a esse sistema um segundo eixo, de modo a permitir que o efetuador final realizasse trajetórias de tecimento definidas em um plano. Assim, o objetivo do projeto consistiu em realizar a versão II do Tartilope, coordenando o movimento de dois eixos para a realização de pendulamentos comuns em soldagem.

Através da adoção de um perfil de velocidade trapezoidal, durante o planejamento dos movimentos estudados, pôde-se obter trajetórias que apresentavam a precisão adequada para a aplicação. As perdas de passos foram minimizadas, pois uma inversão rotação, somente ocorria após a total parada dos motores.

As reduções existentes na estrutura mecânica, permitiram o torque necessário para a movimentação de uma tocha de soldagem pelo efetuador final.

A impossibilidade de se obter uma boa resolução de velocidade nas altas frequências não constituiu um problema relevante, visto que os movimentos de tecimento são realizados em baixas velocidades.

A programação desenvolveu-se de forma modular, de modo que novos movimentos periódicos possam ser programados. Funções que realizem figuras geométricas elementares (circunferências, quadrados, triângulos, etc.), podem ser implementadas futuramente, apenas com a adição de novas funções de planejamento de trajetória. Estas funções gerariam uma tabela, definindo a velocidade adequada para cada eixo, associada a cada passo do motor. Entretanto, melhores resultados serão obtidos se houver a mesma precisão de deslocamento para cada eixo, visto que a necessidade de coordenação e simetria para esses movimentos é maior.

A diversidade dos assuntos envolvidos com o tema, abrangeu uma grande quantidade de disciplinas ministradas na graduação. Isto proporcionou ao acadêmico uma aplicação prática da teoria estudada, bem como uma boa experiência profissional, através do contato com profissionais mais experientes e pela responsabilidade do desenvolvimento do projeto.

Referências Bibliográficas

[Acarnley, 1992]

ACARNLEY, P.P.. **Stepping Motors: a guide to modern theory and practice.**
IEE Control Engineering. 1992.

[Bento, 1989]

BENTO, Celso R.. **Sistemas de Controle – Teoria e Projetos.** Editora Érica.
São Paulo, 1989.

[Brady et al., 1982]

BRADY, M. **Trajectory Planning, in Robot Motion: Planning and Control.** MIT
Press. Cambridge, 1982.

[Craig, 1989]

CRAIG, J.J.. **Introduction to Robotics: Mechanics and Control.** Addison-
Wesley, Reading, Mass. 1989

[Lichtblau, 1989]

LICHTBLAU, Marcos. **Uma contribuição ao estudo da dinâmica de
posicionadores.** Dissertação de mestrado. UFSC, 1989.

[Paul, 1979]

PAUL, R.P. **Manipulator Cartesian Path Control.** IEEE Trans. **Systems, Man,
and Cybernetics**, vol 9, pp702 –711, 1979.

[Pressman, 1995]

PRESSMAN, Roger S. **Engenharia de Software**. Makron Books. São Paulo, 1995.

[Scheffer, 1998]

SCHEFFER, Cynthia B. **Estudo e Implementação de um Acionamento Microprocessado para motores de Passo**. Monografia de conclusão de curso. UFSC, 1998.

[Toro, 1990]

TORO, Vincent. **Fundamentos de máquinas elétricas**. Editora Santuário, 1989.

