Técnica de Controle Flexível para o Projeto e Desenvolvimento de Equipamentos Automáticos Programáveis

Alexandre dos Santos Roque, Alexandre Dias da Silva
Programa de Pós-Graduação em Engenharia de Produção, Núcleo de Automação e Projetos de Fabricação - NAFA, Universidade Federal de Santa Maria – UFSM, Santa Maria, RS, Brasil.

***Resumo*—A parametrização de configurações de equipamentos diversifica os modos de controle com menores custos e agilidade nas fases de projeto. Este trabalho propõe uma técnica de controle adaptativo/configurável, que possibilita o projeto e controle de equipamentos que são amplamente usados em aplicações dedicadas nas indústrias. A técnica é apresentada por meio de fluxogramas e detalhada com o auxílio de diagramas comportamentais da linguagem UML (Linguagem de Modelagem Unificada) para facilitar aplicações futuras. Os resultados são o desenvolvimento de um conjunto *software/hardware* de controle, com características parametrizáveis. A comunicação entre computador e equipamento realizada pelo protocolo USB, aumenta a portabilidade do sistema de controle e permite o acionamento de equipamentos por computadores pessoais.**

***Palavras Chave:* Sistemas de controle (*control systems*), tecnologia adaptativa (*adaptive technology*), sistemas reconfiguráveis (*reconfigurable systems*), equipamentos automáticos programáveis (*programmable automatic machines*).**

I. INTRODUÇÃO

S sistemas de controle (*hardware e software*) devem ser cada Ovez mais susceptíveis a adaptações ou configuráveis as necessidades de controle dos projetos. A agilidade e flexibilidade são conceitos reconhecidos como requisitos para os sistemas de controle aplicados em ambiente industrial [1]. Os equipamentos programáveis que fazem uso de motores de passo como meio de acionamento são os que mais se beneficiam de tais conceitos, pois, permitem um controle simplificado e de fácil compreensão, aliando movimentos precisos com o seu baixo custo. Freqüentemente, pesquisadores desenvolvem ou projetam equipamentos de pequeno porte, acionados por motores de passo, com os mais variados objetivos e aplicados em diferentes áreas [2], [3].

Os diversos equipamentos programáveis projetados possuem particularidades que muitas vezes não são atendidas em *softwares* comerciais, pois estes são sistemas fechados, e não permitem ao usuário realizar modificações ou adaptações, fato que faz com que os equipamentos e projetos se adaptem aos *softwares* de controle e não o contrário.

O controle de equipamentos com objetivos dedicados tomam como referência exemplos e características consolidadas na indústria, dentre os quais, as máquinas de comando numérico computadorizado – CNC se destacam, por possuírem características parametrizáveis em seu sistema de controle para se adaptar a um projeto ou processo de fabricação [4]. Nesse sentido, em muitos projetos existem características que podem ser benéficas ao controle de diversos equipamentos e configuradas de acordo com suas aplicações.

Outro aspecto importante para o controle de equipamentos diz respeito ao modo de funcionamento do seu sistema de controle. O sistema de controle que estabelece uma relação comparativa entre saída e entrada de referência, utilizando a diferença como meio de controle, caracteriza-se como sistema de controle com realimentação ou controle em malha fechada. De outra forma, o sistema de controle onde a saída não exerce ação sobre a entrada, ou seja, não é usada como referência para uma nova entrada, é denominado sistema de controle em malha aberta [5].

Dentre as características ou funcionalidades que um sistema de controle pode possuir, a forma de comunicação entre o equipamento e o computador é um fator crucial, pois, a evolução das portas de comunicação propiciou ao longo do tempo maior velocidade e confiabilidade na transmissão de dados, provendo portabilidade a diferentes equipamentos. Por ser um padrão de comunicação reconhecido e popularizado, o protocolo USB (*Universal Serial Bus*) constitui um diferencial para o desenvolvimento de uma técnica de controle flexível.

De acordo com [6], o desenvolvimento de uma tecnologia própria gera flexibilidade para a pesquisa, de modo que, diminui as limitações ou restrições impostas por um equipamento ou software comercial, sendo possível realizar alterações de acordo com as necessidades encontradas no decorrer do projeto ou do seu uso em campo.

Desta forma, este trabalho apresenta uma metodologia para o acionamento de equipamentos programáveis por meio da

---

Este trabalho consolida os resultados de pesquisa recentemente realizada no Núcleo de Automação e Projetos de Fabricação – NAFA, da Universidade Federal de Santa Maria, sendo parte integrante de dissertação de mestrado focada em Automação e Informática Industrial.

Alexandre dos Santos Roque possui graduação em Ciência da Computação, e concluiu recentemente Mestrado em Engenharia de Produção com Ênfase em Automação e Informática Industrial, pela Universidade Federal de Santa Maria - UFSM.
Alexandre Dias da Silva é Doutor em Engenharia Mecânica e Professor Adjunto da Universidade Federal de Santa Maria - UFSM.

técnica de controle em malha aberta, com comunicação através do protocolo USB. O sistema de controle é destinado ao projeto e desenvolvimento de equipamentos automáticos com aplicações dedicadas nas indústrias.

O artigo está dividido da seguinte maneira: na seção 2 são enfocados aspectos essenciais para a parametrização e flexibilidade de controle dos equipamentos; na seção 3 aspectos metodológicos são abordados; a seção 4 descreve a técnica de controle proposta; na seção 5 são abordados os resultados da pesquisa; por fim a seção 6 apresenta as conclusões e perspectivas do trabalho.

II. REQUISITOS PARA AUTOMAÇÃO FLEXÍVEL

Para agregar características de flexibilidade aos equipamentos, alguns aspectos são importantes e considerados fundamentais, permitindo o controle de equipamentos programáveis por computadores pessoais [7]. O uso de motores de passo, a porta de comunicação usada, a visualização com simulação gráfica do projeto, e características como resolução (distância percorrida a cada comando de movimentação), aceleração, desaceleração e o modo de codificação (linguagem usada) dos passos que representam a trajetória do equipamento, podem ser parametrizáveis por meio de um sistema de controle (conjunto *hardware e software*).

Os motores de passo são um exemplo de movimentadores que agregam características fundamentais, tais como, grande precisão de movimentos, controle de aceleração e desaceleração, entre outras. Por conseguinte, são amplamente usados em sistemas robóticos e no controle de equipamentos, onde estes têm que realizar operações de posicionamento de alta precisão [8]. De acordo com [9], um motor de passo atua como um preciso acionador mecânico, tendo um dispositivo elétrico dividindo a rotação do motor em um grande número de passos, proporcionando movimentação precisa.

Diversos trabalhos utilizam estes motores como atuadores, sendo projetos com as mais variadas características e complexidades, como por exemplo, equipamentos que fazem a obtenção de propriedades mecânicas em dutos de petróleo [10]; um projeto de ambiente de aprendizagem remoto (via WEB) que permite a prática laboratorial e aprendizado de física [11]; um posicionador eletrônico para estruturas micrométricas, utilizado para posicionar objetos com passos tridimensionais e micrométricos [12]. Outro exemplo que destaca a importância da parametrização do controle é um dispositivo programável que permite a mobilização dos membros inferiores imediatamente no pós-operatório (exoesqueleto), promovendo a manutenção dos tecidos que envolvem a articulação, destacando o uso da tecnologia de comando numérico e propondo um modelo que opere de maneira flexível, fornecendo possibilidades de programação por parte do usuário [13].

A quantidade crescente de projetos e sistemas robóticos diferenciados alavanca a necessidade da existência de alternativas que possam acelerar o projeto destes equipamentos, focando na flexibilidade e portabilidade dos sistemas de controle. No âmbito de sistemas de controle para equipamentos programáveis é notável a necessidade de adaptação as mudanças de projeto, sendo necessário possuir características configuráveis.

III. MOTIVAÇÃO E METODOLOGIA

Este trabalho objetiva apresentar uma técnica de controle focada na flexibilidade e parametrização de configurações, para auxiliar no projeto e desenvolvimento de equipamentos programáveis.

O estudo iniciou com uma pesquisa bibliográfica sobre controle de equipamentos automáticos programáveis em âmbito industrial, constatando a realidade atual dos mesmos e explorando as suas sistemáticas de controle. Desta forma, do ponto de vista de seus objetivos, esta pesquisa é classificada como bibliográfico-exploratória, atuando em área onde existem estudos, mas ainda com possibilidades de pesquisa, que podem ser exploradas visando à obtenção de novas soluções e expandir os horizontes da área. Quanto à natureza, esta pesquisa é qualitativa, porque identifica características fundamentais para a melhoria dos sistemas de controle para equipamentos programáveis, focando na possibilidade de parametrização de configurações do equipamento, tornando-o assim, um sistema adaptativo aos projetos. Assim, é proposta e apresentada uma metodologia de controle flexível aplicando estas características, facilitando a sua compreensão e aplicabilidade, para agregar alternativas no processo de controle de equipamentos programáveis. Com tais características, a metodologia também objetiva integrar pesquisadores de outras áreas promovendo assim atividades interdisciplinares.

Para testes da metodologia proposta foi desenvolvido um circuito elétrico de controle, como interface de comunicação entre o computador e o equipamento programável. A sua composição permite o envio de comandos ao equipamento por meio da porta de comunicação USB.

Por fim, um *software* foi desenvolvido seguindo as características da metodologia proposta com capacidade para simular graficamente a trajetória do equipamento controlado, permitindo ainda correções e ajustes prévios, para em seguida enviar os comandos (pulsos elétricos) do computador para os atuadores do equipamento controlado.

Os testes para validação do sistema de controle foram realizados no Núcleo de Automação e Projetos de Fabricação - NAFA na Universidade Federal de Santa Maria – UFSM.

IV. TÉCNICA DE CONTROLE EMPREGADA

A metodologia de controle flexível proposta neste trabalho proporciona o projeto e desenvolvimento de equipamentos programáveis, aliando baixo custo com alta portabilidade. O foco é agregar valor e propor melhorias para os sistemas de controle desenvolvidos atualmente, aumentando as possibilidades de controle dos equipamentos, impulsionando atividades interdisciplinares e o aprendizado de automação e controle.

A informação de entrada do sistema proposto é um arquivo contendo uma codificação de pulsos que representa a trajetória do equipamento no ambiente de atuação. Por meio desta codificação, o sistema de controle faz a leitura e interpretação de cada código, que possui uma letra do alfabeto

representando um determinado movimento nos eixos/motores do equipamento. Esta forma de entrada de dados (Tabela 1) pode ser inserida manualmente no programa, representando uma trajetória conhecida e simplificada, movimentando os motores em sentido horário ou anti-horário. Outra forma de entrada de dados é o uso de um programa que converte a trajetória de um equipamento para estes códigos de movimentação [14], por meio de técnicas de interpolação linear e circular, gerando um arquivo que pode ser carregado no sistema de controle.

A Tabela 1 mostra a codificação da movimentação de um equipamento programável. O sinal de "+" e "-" na descrição do eixo controlado indica o sentido (horário ou anti-horário) de movimentação do eixo.

TABELA 1
CODIFICAÇÃO DOS PULSOS PARA MOVIMENTAÇÃO DOS EIXOS

Cod	Descrição do Movimento	Eixo
A	Avanço no Eixo X	+X
B	Avanço no Eixo Y	+Y
C	Avanço no Eixo Z	+Z
D	Retorno no Eixo X	-X
E	Retorno no Eixo Y	-Y
F	Retorno no Eixo Z	-Z
G	Avanço no Eixo X e Y	+X e +Y
H	Avanço no Eixo X e Z	+X e +Z
I	Avanço no Eixo Y e Z	+Y e +Z
J	Retorno no Eixo X e Y	-X e -Y
K	Retorno no Eixo X e Z	-X e -Z
L	Retorno no Eixo Y e Z	-Y e -Z
M	Avanço em X e Retorno em Y	+X e -Y
N	Retorno em X e Avanço em Y	-X- e +Y
O	Avanço em X e Retorno em Z	+X e -Z
P	Retorno em X e Avanço em Z	-X e +Z
Q	Avanço em Y e Retorno em Z	+Y e -Z
R	Retorno em Y e Avanço em Z	-Y e +Z
S	Avanço nos Eixos X, Y e Z	+X, +Y,+Z
T	Retorno nos Eixos X, Y e Z	-X, -Y, -Z
U	Avanço Eixo X e Retorno Y e Z	+X, –Y, -Z
V	Avanço Eixo Y e Retorno X e Z	-X, +Y, -Z
W	Avanço Eixo Z e Retorno X e Y	-X, -Y, +Z
X	Retorno Eixo X e Avanço Y e Z	-X, +Y, +Z
Y	Retorno Eixo Y e Avanço X e Z	+X, -Y, +Z
Z	Retorno Eixo Z e Avanço X e Y	+X, +Y, -Z

Esta codificação serve de parâmetro para que sistemas de controle futuros possam ser desenvolvidos de maneira mais amigável, melhorando a característica de usabilidade, possibilitando maior flexibilidade de controle, bem como também a realização de atividades interdisciplinares.

Ao executar, por exemplo, a informação "AAAAAAAAAA BBBBB GGGGGGGGGG" o sistema de controle envia dez pulsos de movimentação, no eixo X, cinco pulsos, no eixo Y, e depois dez pulsos em ambos os eixos X e Y. Neste caso, a tabela trabalha com sistemas atuando com três eixos de coordenadas, cada eixo tendo um motor de passo como atuador.

A Fig. 1 apresenta um diagrama representativo da metodologia de controle proposta neste trabalho, representada por meio de um fluxograma dinâmico, mostrando o fluxo de dados de controle da metodologia, dividindo os seus principais componentes. No diagrama os atores principais são o operador ou usuário do sistema e o equipamento a ser controlado. Assim, um fluxo de informações de controle é intermediado por vários processos, como a entrada de dados, interpretação, simulações, correções e envio dos comandos para a porta USB. Estes são responsáveis por traduzir a entrada de dados feita pelo operador do sistema em movimentos nos atuadores do equipamento.

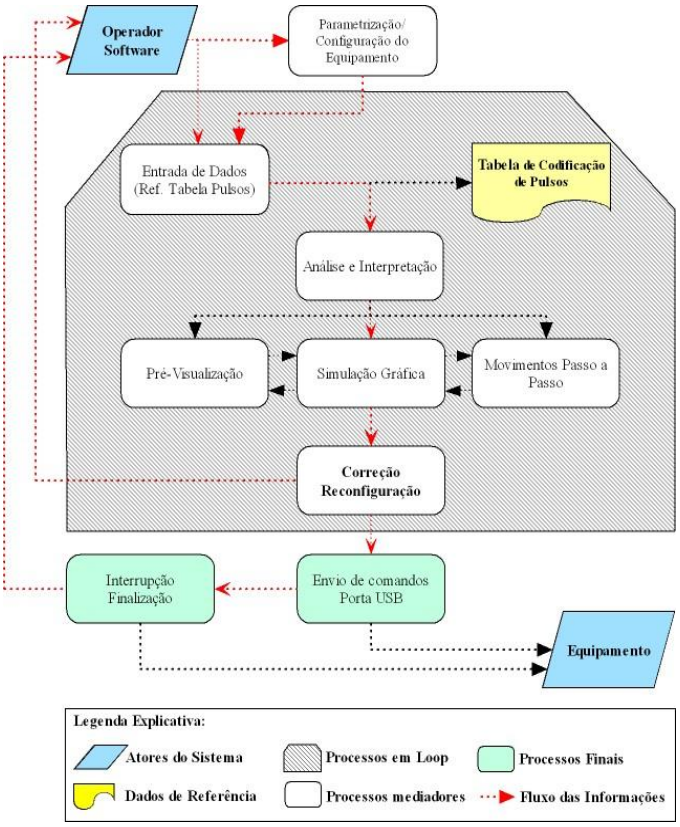

Fig. 1: Diagrama da metodologia em forma de modelo dinâmico.

Para contextualizar a metodologia proposta, descrevendo a sua dinâmica comportamental e mostrando a interação entre os atores do sistema, foi utilizada a modelagem UML – *Unified Modeling Language*, por meio de alguns de seus diagramas, diagrama de casos de uso e o diagrama de atividades. A linguagem UML utiliza representações gráficas para descrever os vários elementos que compõem um sistema, permitindo sua visualização sob diferentes perspectivas. O objetivo principal da UML é fornecer uma linguagem padrão que permita modelar um sistema, possibilitando a troca natural destes modelos entre os desenvolvedores de *softwares* [15]. Com a UML é possível descrever eficazmente requisitos de *software*, caracterizar a arquitetura de um sistema e direcionar programadores, aumentando a produtividade e diminuindo os riscos [16] [17].

O objetivo da diagramação é descrever a dinâmica comportamental da metodologia de controle proposta. Deste modo, dois diagramas da visão dinâmica da UML são utilizados, o diagrama de casos de uso e o de atividades.

O diagrama de ***casos de uso***, apresentado na Fig. 2, é usado para representar os atores, que desempenham papéis fundamentais no sistema, mostrando o fluxo entre os

processos ou funções que nesta técnica o sistema de controle deve realizar. O operador (ator inicial), por intermédio do *software* de controle, inicializa o processo com a configuração do equipamento e leitura do arquivo contendo os códigos de pulsos, este processamento pode ser simulado graficamente antes do envio dos comandos para a realização da movimentação do equipamento (ator final).

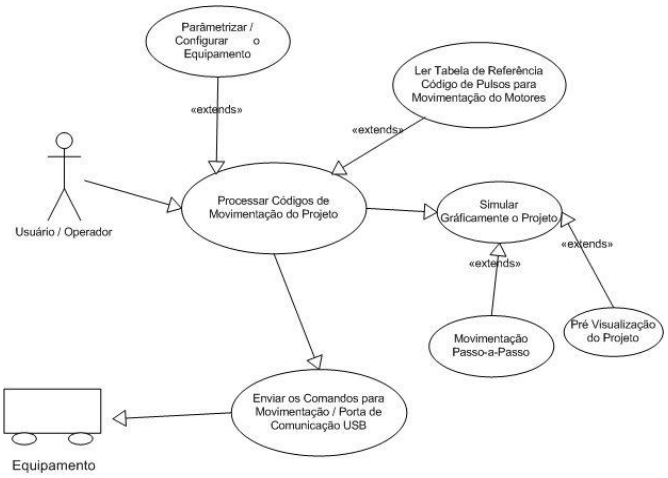

Fig. 2: Diagrama de Casos de Uso para a metodologia de controle.

O ***diagrama de atividades*** ilustra a seqüência de atividades realizadas no sistema, demonstrando o fluxo de trabalho de um ponto inicial até o ponto final, detalhando as decisões tomadas durante o caminho percorrido durante a execução das tarefas.

Dois pontos formados por círculos preenchidos na cor preta representam o estado inicial e o estado final do sistema. Entre eles os processos e as tomadas de decisão formam o fluxo de controle (*workflow*). No diagrama da Fig. 3, uma atenção especial deve ser dada ao processo de correção ou reconfiguração do projeto, pois, esta tomada de decisão gera um novo processamento no sistema de controle para permitir a adaptação da trajetória inicial especificada, fazendo assim o movimento esperado no equipamento.

A seguir, a Fig. 3 apresenta este diagrama.

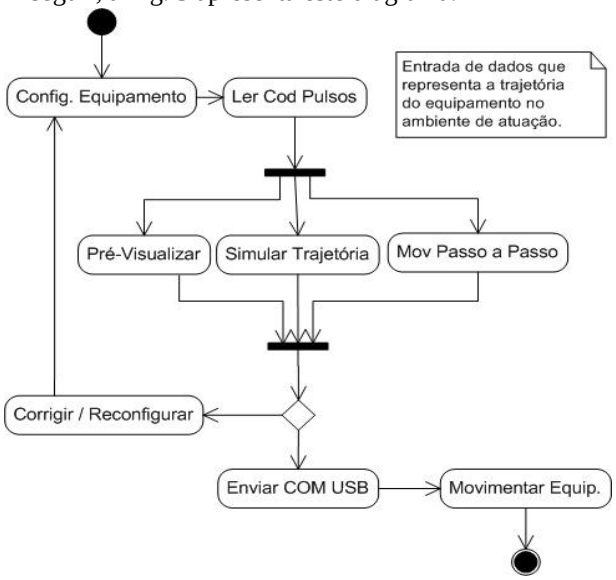

Fig. 3: Diagrama de Atividades da metodologia de controle.

O processo envia os comandos para o equipamento por meio do protocolo USB, utilizando uma biblioteca de funções de controle, tendo como resultado o movimento dos motores do equipamento. Na seção seguinte são detalhados os resultados da pesquisa, descrevendo o *hardware* e o *software* desenvolvido de acordo com a metodologia de controle.

V. RESULTADOS

O trabalho foi validado com o desenvolvimento de um *hardware* e um *software* para controle de equipamentos programáveis, os quais são apresentados a seguir.

A. Hardware de controle desenvolvido

Para efetivar o controle é necessário um dispositivo de *hardware* que faça o papel de interface de comunicação entre o *software* de controle instalado no computador e o equipamento a ser controlado. O *hardware* é composto de atuadores controlados e alimentados por meio de um circuito elétrico.

Com o advento da comunicação USB, desenvolvedores e projetistas obtiveram ganhos na conectividade ou portabilidade dos equipamentos projetados com a velocidade de transmissão de dados, conexão simplificada e suporte a software que este protocolo de comunicação permite.

O modelo apresentado possibilita o controle de um equipamento via *software*, seguindo as características enfocadas na técnica. O circuito desenvolvido permite a comunicação entre o computador e o equipamento por meio do protocolo USB [18].

O circuito de controle é baseado em um módulo *USB-to-FIFO* de interface paralela, dispondo de um barramento de dados bidirecional de 8 bits e sinais de controle, que são usados para controlar o fluxo de dados entre o computador e qualquer circuito externo utilizando o protocolo de comunicação USB.

O módulo utilizado é o componente modelo DLP-USB245M-G, visualizado na Fig. 4, da FTDI (*Future Technology Devices International*), que possui em seu *firmware* (programa controlador) toda a camada do protocolo USB. São fornecidos pelo fabricante os *drivers* para a instalação no computador, tendo sua biblioteca de funções e documentação detalhada para desenvolvimento na linguagem de programação *C++ Builder*® ou *Visual C++*®.

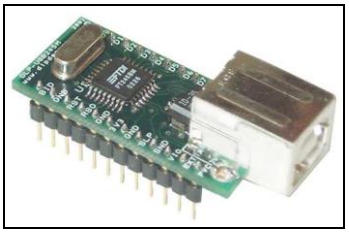

Fig. 4: Chip FTDI DLP-USB245M-G.

As principais características do componente são:

• Velocidade de transmissão e recebimento de dados de 1 MBps;
• Comunicação USB gerenciado pelo próprio chip;
• Compatível com padrão USB 1.1 e 2.0;
• 384 bytes reservados em *buffer* para transmissão;
• 128 bytes reservados em *buffer* para recepção;
• Montagem em soquete de 24 pinos, o que facilita a conexão em circuitos;
• Modo de transferência *Bulk* (em grandes quantidades e simultaneamente) e Isossíncrona (quantidade pré-negociável da banda de transmissão do barramento);
• *Drivers* de comunicação compatíveis com Windows 98/2000/CE/XP, MAC OS-8 e OS-9 e Linux 2.40 ou superior.

O dispositivo *USB-toFIFO*, DLP-USB245M-G, é usado como meio de comunicação em alguns trabalhos atuais [19], [20], pois, agrega um meio simples e de baixo custo para permitir a comunicação dos computadores atuais com o hardware de controle.

Durante os testes do circuito de controle, realizado no NAFA da UFSM, foi detectado que do modo em que o circuito elétrico está constituído, é necessário o envio de um ***pulso de controle*** que seleciona um motor diferente do atual, mas sem ativar bobinas, entre os pulsos de uma seqüência no mesmo eixo/motor (ex: pulsos "AAAAA"). Esse pulso é enviado para gerar uma transição de *clock* (sinal que habilita o envio de pulsos) entre comandos enviados no mesmo eixo. Quando pulsos são enviados alternadamente em diferentes eixos (ex: pulsos "ABABABCACA"), este pulso de controle não é necessário, pois a simples troca de eixo gera o *clock* necessário.

B. Software de controle desenvolvido

O *software* para controle desenvolvido neste trabalho foi chamado de "SCEAP – Sistema de Controle para Equipamentos Automáticos Programáveis". Este *software* foi desenvolvido focando o projeto, controle e testes de equipamentos com aplicações dedicadas nas indústrias que fazem o uso de motores de passo como atuadores, como mesas XY, XYZ, pequenos robôs, e outros equipamentos.

O *software* desenvolvido para testes da técnica compreende algumas etapas que foram seguidas para torná-lo simples e funcional facilitando o seu uso. Estas etapas são:
• Escolha da linguagem de programação;
• Estruturação da interface gráfica;
• Programação do ambiente gráfico de simulação;
• Programação da comunicação por meio da porta USB; e
• Testes com o *hardware* desenvolvido.

A base para seu desenvolvimento são características apresentadas em [7], as quais foram agregadas à técnica apresentada, para flexibilizar o controle dos equipamentos. A linguagem de programação utilizada foi a *C++*, por ser uma linguagem que permite a programação orientada a objetos, com amplo uso no meio acadêmico e a boa documentação do módulo DLP-USB245M-G agregado ao *hardware* de controle.

A possibilidade de parametrização de variáveis que compõem a planta de atuação do equipamento é importante, pois, permite aos pesquisadores, o projeto e testes de equipamentos em sua fase inicial. Assim, evita-se o gasto dispendioso de tempo no desenvolvimento de software de controle específico, contribuindo para obtenção de resultados em menor tempo. A seguir são apresentadas as principais telas do sistema desenvolvido, enfocando as suas particularidades de programação e controle.

A Fig. 5 mostra a tela inicial, onde é dada a entrada de um arquivo contendo a codificação da movimentação ou trajetória do equipamento. Usando a opção "Código de Pulsos - Abrir", o sistema faz a leitura e interpretação do código de pulsos, armazenando esta informação para uso na simulação e na movimentação efetiva do equipamento. Em seguida, o usuário pode escolher qual o tipo de simulação que deseja realizar.

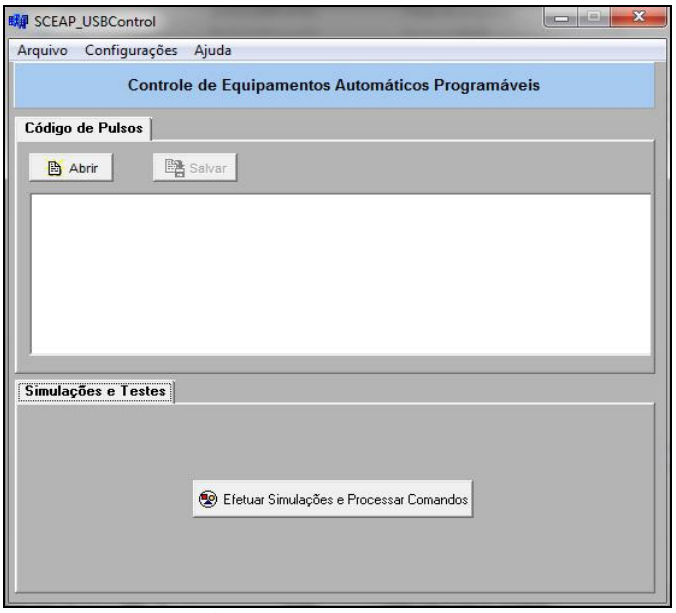

Fig. 5: Tela inicial do software desenvolvido.

Descrição da tela:
• Menu Arquivo – Opção de acesso a leitura do arquivo contendo o código de pulsos;
• Menu Configurações – Parâmetros que são configuráveis;
• Menu Ajuda – Descrição geral do sistema;
• Abrir Código de Pulsos – Códigos de pulsos que representam a trajetória que o equipamento irá realizar;
• Efetuar Simulações e Processar comandos – Esta opção possibilita simular e executar o controle de equipamentos que usam sistemas de coordenadas com dois eixos. Outra opção de simulação também mostra em uma linha de tempo a movimentação de um equipamento com três eixos;

Após a leitura do arquivo contendo o código de pulsos, a pré-visualização e simulação da movimentação do equipamento podem ser realizadas. A simulação gráfica foi programada usando a linguagem OpenGL, que é uma biblioteca de rotinas gráficas que permite a programação de diversas formas geométricas em 2D e 3D, a qual é muito utilizada em jogos e sistemas de visualização gráfica. Optou-se pela linguagem OpenGL, por ser facilmente integrada ao ambiente de programação *C++ Builder*®, onde permitiu a

programação e simulação gráfica da trajetória do equipamento controlado.

A programação realizada no sistema permite a visualização e correção pró-ativa do equipamento que está utilizando o sistema de controle. A Fig. 6 mostra a tela com a simulação gráfica de uma trajetória exemplificada em um sistema com dois eixos, podendo-se observar a esquerda do eixo Y do plano cartesiano as informações da simulação que está em andamento.

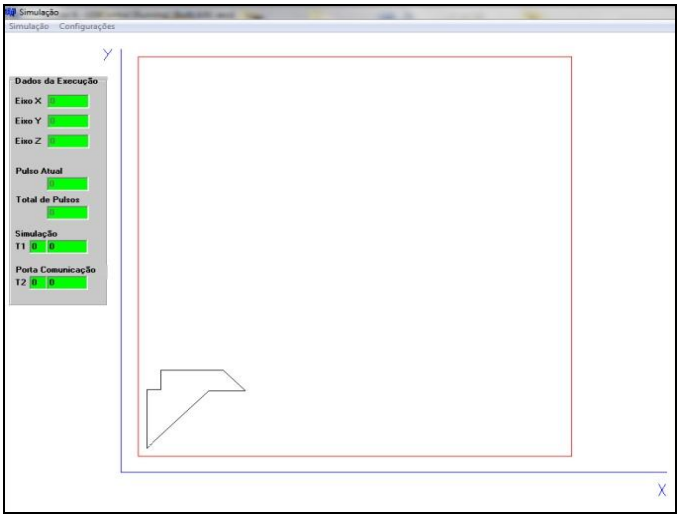

Fig. 6: Tela que faz a simulação da trajetória do equipamento.

Os dados atualizados durante a simulação são:

• Posição atual do Eixo X e do Eixo Y;

• Pulso atual – de acordo com o arquivo de codificação da trajetória; Total de pulsos que serão executados;

• T1 – Tempo de simulação;

• T2 – Tempo de envio dos comandos para o equipamento.

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

Outra opção de simulação é a execução do controle de três eixos/motores de um equipamento sendo visualizada em uma linha de tempo. A Fig. 7 ilustra este modo de simulação.

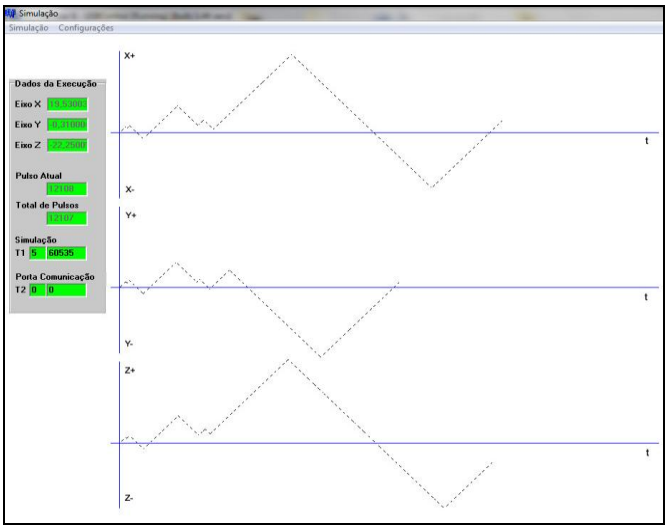

Fig. 7: Simulação do processamento dos pulsos em linha do tempo.

Esta opção pode ser usada em aplicações diversas, onde o monitoramento do envio dos comandos para os motores pode ser visualizado passo a passo. Pode-se citar como exemplo do uso deste tipo de simulação, o projeto de um equipamento para reabilitação passiva contínua em cotovelo, onde foram utilizados dois motores de passos realizando a movimentação de duas articulações [21]. Este tipo de equipamento não usa sistema de coordenadas, assim, pode-se testar e visualizar em uma linha do tempo os pulsos enviados aos motores.

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

São elas:

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

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

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

• **Aceleração inicial e desaceleração:** configuração por eixo controlado no sistema, onde é informada a aceleração inicial do equipamento. São configurações importantes e usadas de acordo com o tamanho do equipamento e características físicas dos motores, sendo informada pelo projetista, de acordo com variantes como inércia da carga e tamanho/limites da planta/ambiente de atuação.

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

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

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

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

FT_ListDevices *(PVOID pvArg1, PVOID pvArg2, DWORD dwFlags)*

A função *FT_ListDevices* retorna informações dos dispositivos conectados, como por exemplo, o número de dispositivos conectados, descrição do componente e identificador usado na conexão.

FT_Open *(int iDevice, FT_HANDLE *ftHandle)*

A função *FT_Open* é usada para abertura de um ou mais dispositivos, deixando-o pronto para os acessos seguintes.

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

FT_Write *(FT_HANDLE ftHandle, LPVOID lpBuffer, DWORD dwBytesToWrite, LPDWORD lpdwBytesWritten)*

As operações de escrita no dispositivo que resultarão em movimentos nos motores de passo, são processadas pela função *FT_Write*, a qual tem sua estrutura detalhada e definida de acordo com quatro parâmetros importantes. Estes parâmetros são definidos de acordo com as características do hardware a controlar.

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

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

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

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

Na Fig. 8 é visualizada uma sequencia de seis pulsos (AAAADD), os quais são representados por códigos hexadecimais, que no circuito de controle são convertidos em códigos de 8 bits, para assim representar o motor selecionado e acionado.

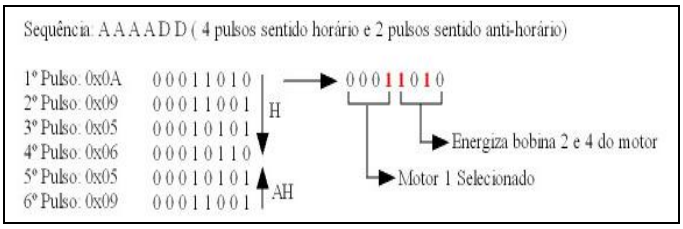

Fig. 8: Envio de pulsos com a composição do *byte* enviado para a porta USB.

Neste trabalho, o circuito de controle implementado permite o controle de até quatro motores. Para a movimentação correta dos motores, os dados devem ser enviados para a porta USB seguindo uma tabela de dados de controle, a qual é apresentada a seguir pela Fig. 9.

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

Fig. 9: Dados de controle dos motores [18].

A sequência de envio dos comandos define a direção de rotação do motor, e as configurações de velocidade, aceleração e desaceleração são especificadas no software de controle. Esta tabela contém os valores em notação hexadecimal que são usados para energizar as bobinas de um determinado motor, sendo que a sequência de ativação destas bobinas fará com que o motor gire em determinado sentido.

O arquivo de entrada contendo o código de pulsos é resultado de ***técnicas de interpolação linear e circular***, como já destacado anteriormente. Na Fig. 10 é visualizada uma seqüência gráfica que exemplifica o processamento de um arquivo de pulsos contendo uma técnica de interpolação circular [22].

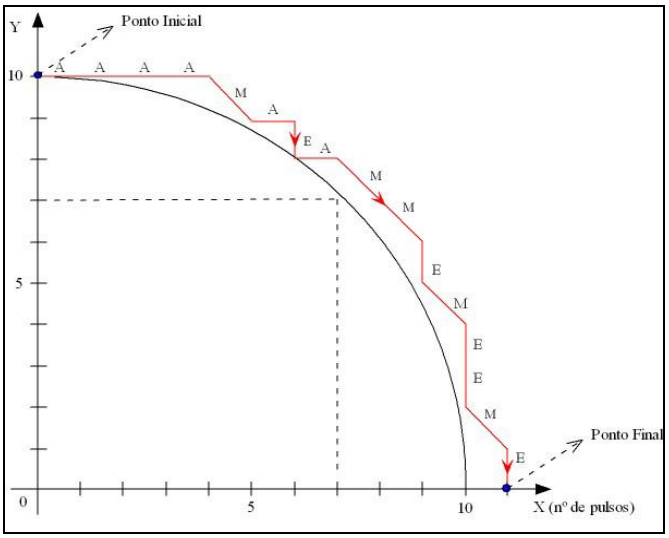

Fig. 10: Sequência de pulsos exemplificada na técnica de interpolação circular. Adaptado de [22].

Desta forma, o *software* faz a leitura dos códigos de envio para movimentar o equipamento em determinada trajetória. Todas as funções utilizadas são importantes na programação do software de controle, pois possibilitam o controle e parametrização de características de um equipamento.

VI. CONCLUSÕES E CONSIDERAÇÕES FINAIS

Esta pesquisa apresentou uma técnica de controle flexível/adaptativa para equipamentos automáticos programáveis, dando ênfase à flexibilidade de controle, com a possibilidade de configuração e parametrização de características dos equipamentos.

A parametrização destas características é delimitada pela complexidade dos projetos, que é resultado da criatividade dos projetistas. Muitas vezes esta criatividade é limitada às possibilidades de controle oferecidas por *softwares* comerciais fechados. Sendo assim, as características configuráveis destacadas e usadas, diversificaram as funcionalidades de controle.

A técnica proposta é aplicável em ambiente industrial como alternativa de sistema de controle, pois suas características permitem o desenvolvimento de equipamentos com funções dedicadas e peculiares. Considerando que para se fazer uso da técnica, é necessário que a sequência de pulsos para cada tarefa programada esteja de acordo com a codificação especificada, o sistema de controle exige processamento prévio desses dados. Porém, o uso de técnicas de interpolação linear e circular, implementados em trabalhos recentes [14], representam um meio eficiente de conversão dos movimentos desejados na codificação de pulsos usada (Tabela 1).

É importante destacar na técnica, a comunicação e o controle entre equipamento e computador pessoal, usando o protocolo USB, realizado por meio de um dispositivo eletrônico (chip USB245BM-G) programável via *software*. A biblioteca de funções desse componente traz possibilidades de controle por esta porta de comunicação, tornando o sistema de controle portável (usado em qualquer computador pessoal) e de acordo com os requisitos de comunicação atuais.

Os testes do conjunto *software/hardware* desenvolvidos foram realizados em laboratório, com motores de passo comumente usados em diversos projetos, sendo também usado para testes de trabalhos recentes realizados no Núcleo de Automação e Projetos de Fabricação – NAFA/UFSM [14] [21], demonstrando a sua viabilidade no projeto e desenvolvimento de novos equipamentos.

Por fim, conclui-se que a técnica apresentada permite aos projetistas maior liberdade nas operações de acionamento, diminuindo o tempo gasto no desenvolvimento de *software e hardware* de controle específico (usados nos testes e projeto de novos equipamentos), possibilitando assim, diminuir os custos relacionados.

Este trabalho possibilita a realização de outras pesquisas na área de sistemas de controle flexíveis destinados ao projeto e desenvolvimento de equipamentos programáveis. Desta forma, algumas atividades podem ser realizadas para dar esta continuidade, como as descritas a seguir:

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

REFERÊNCIAS

[1] M. A. B. De Paula, E. A. P. Santos, "Uma Abordagem Metodológica para o Desenvolvimento de Sistemas Automatizados e Integrados de Manufatura" *Revista Produção*, 2008.

[2] L. S. Magalhães, "Abastecimento de Estoque por Robô Microcontrolado", *Anhanguera Educacional S. A.*, São Paulo, 2009.

[3] M. Y. Kamogawa, M. A. Teixeira, "Autoamostrador de Baixo Custo para Análise por Injeção em Fluxo", *Revista Química Nova*, 2009.

[4] A. S. Ribeiro, "Modelagem e Especificação de Controle de Sistemas Flexíveis de Manufatura, Utilizando Redes de Petri de Alto Nível", Programa de Pós-Graduação em Mecatrônica, Universidade Federal da Bahia , Dissertação de Mestrado, 2009.

[5] K. Ogata, *Engenharia de Controle Moderno*. Pearson Prentice Hall, 4ª ed., SP, 2003.

[6] J. C. Dutra, L. Felippe, R. S. Carvalho, "Sistema robótico de quatro graus de liberdade e processos de soldagem dedicados para o revestimento de tubos de caldeiras", Universidade Federal de Santa Catarina – UFSC, *V Congresso de Inovação Tecnológica em Energia Elétrica – V CITENEL*, Belém/PA, 2009.

[7] A. S. Roque, A. D. Silva. "Sistemas de Controle para Equipamentos Automáticos Programáveis: Características Fundamentais para Compor uma Metodologia de Controle Flexível", *4º Workshop de Robótica Aplicada e Automação, Robocontrol*, 2010.

[8] P. Melin, O. Castilho, "Intelligent Control of a Stepping Motor Drive Using Adaptive Neuro-fuzzy Inference System", *Journal Information Sciences*, 170 133-151, Elsevier – ScienceDirect, 2004.

[9] C. Muñiz, R. Levi, M. Benkrid, F. B. Rodriguez, P. Varona, "Real-time control of stepper motors for mechano-sensoryvstimulation", *Journal of Neuroscience Methods*, Elsevier – ScienceDirect, 2008.

[10] J. V. De Souza, A. B. Soares, S. D. Franco, "Desenvolvimento de um Sistema de Hardware e Software para Otimização das Unidades de Controle, Monitoração, Coleta e Processamento de Dados de um Macroidentador Portátil para Avaliação In-Situ de Propriedades Mecânicas de Dutos Metálicos", *Revista Horizonte Científico*, Volume 1, N. 9, 2008.

[11] C. R. Oliveira, I. Oliveira, H. Santos, A. Pereira, "Um Ambiente para a Prática Remota de Aulas Laboratoriais de Física (determinação da viscosidade de líquidos)", *Revista Brasileira de Informática na Educação - RBIE*, v.17, N. 1 – 2009.

[12] G. M. Calixto, "Desenvolvimento de um posicionador eletrônico para estruturas micrométricas", Universidade Estadual de Campinas - UNICAMP, Dissertação de Mestrado, 2009.

[13] D. Q. Sperb, "Desenvolvimento de dispositivo programável de movimento passivo contínuo para membros inferiores", Universidade Federal de Santa Maria - UFSM, Dissertação de Mestrado, 2006.

[14] E. F. De Cristo, "Implementação de Técnicas de Controle de Motor de Passo em Aplicações CNC", Universidade Federal de Santa Maria - UFSM, Dissertação de Mestrado, 2009.

[15] R. P. Silva, *UML2 em Modelagem Orientada a Objetos*, Florianópolis, Visual Books, 2007.

[16] T. Pender, *UML a Bíblia*. Elsevier, RJ, 2004.

[17] R. S. Moura, L. A. Guedes, "Modelagem de aplicações de automação industrial usando diagramas UML e reuso de componentes", Universidade Federal do Piauí – UFPI, *Simpósio Brasileiro de Automação Inteligente – SBAI*, 2007.

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

[19] J. M. Nageb, A. Ruhullah, S. Salleh, "12 Channel USB Data Acquisition System For QT Dispersion Analysis", Proceedings of the International Conference on Robotics, Vision, *Information and Signal Processing, ROVISP*, 2005.

[20] P. Giassi Junior, "Etofisiógrafo: Protótipo de um Sistema Telemétrico para Registro de Variáveis Fisiológicas em Pequenos Animais", Universidade Federal de Santa Catarina – UFSC, Dissertação de Mestrado, 2006.

[21] A. M. Callegaro, "Desenvolvimento de um Equipamento Computadorizado de Movimentação Passiva Contínua para Cotovelo e Antebraço", Universidade Federal de Santa Maria – UFSM, Dissertação de Mestrado, 2010.

[22] M. Weck, *Werkzeugmaschinen Band 3 – Autommatisierung und Steuerungstechnik*. Düsseldorf: VDI-Verl, Germany, 1989.