

ESCOLA POLITÉCNICA DA UNIVERSIDADE DE SÃO PAULO



**PMR3550 — Trabalho de Conclusão de Curso II em Engenharia
Mecatrônica**

**PEDRO BERNARDO DOS SANTOS MELO – N.USP: 9887402
SAMUEL FILLIPINI MONÇÃO – N.USP: 8989321**

São Paulo - SP
18 de Novembro de 2018

PEDRO BERNARDO DOS SANTOS MELO
SAMUEL FILLIPINI MONÇÃO

**PROJETO E CONSTRUÇÃO DE PROTÓTIPO DE POSICIONADOR
CARTESIANO DE ALTA PRECISÃO CONTROLADO POR COMPUTADOR.**

São Paulo
2018

PEDRO BERNARDO DOS SANTOS MELO
SAMUEL FILLIPINI MONÇÃO

**PROJETO E CONSTRUÇÃO DE PROTÓTIPO DE POSICIONADOR
CARTESIANO DE ALTA PRECISÃO CONTROLADO POR COMPUTADOR.**

Relatório apresentado à Escola Politécnica da
Universidade de São Paulo como requisito da
disciplina PMR3550 – Trabalho de Conclusão
de Curso II em Engenharia Mecatrônica Área
de Concentração: Engenharia Mecatrônica

Orientador: Prof. Dr Rodrigo Lima Stoeterau

São Paulo
2018

Autorizo a reprodução e divulgação total ou parcial deste trabalho, por qualquer meio convencional ou eletrônico, para fins de estudo e pesquisa, desde que citada a fonte.

Catálogo-na-publicação

Melo, Pedro Bernardo dos Santos
Monção, Samuel Fillipini

PROJETO E CONSTRUÇÃO DE PROTÓTIPO DE POSICIONADOR
CARTESIANO DE ALTA PRECISÃO CONTROLADO POR COMPUTADOR /
P. B. S. Melo, S. F. Monção – São Paulo, 2018.
76 p.

Trabalho de Formatura - Escola Politécnica da Universidade de São
Paulo. Departamento de Engenharia Mecatrônica e de Sistemas Mecânicos.

1.Posicionador cartesiano controlado por computador 2.desenvolvimento
da parte mecânica de um posicionador 3.malha de controle de um posicionador
I.Universidade de São Paulo. Escola Politécnica. Departamento de
Engenharia Mecatrônica e de Sistemas Mecânicos II.t. III.Monção, Samuel Fillipini

RESUMO

Este projeto teve por objetivo descrever o projeto e a construção de um posicionador cartesiano de alta precisão, que poderá ser usado para posicionar a ferramenta de um torno de forma a obter um alto grau de qualidade de fabricação. Este projeto descreve o planejamento do mecanismo, junto com o desenvolvimento do projeto, mecânico, eletrônico e a programação e implementação do controle em malha fechada. Além disso, serão implementados a modelagem e análise cinemática e dinâmica para desenvolvimento e validação dos parâmetros de controle.

Palavras-chaves: Posicionador cartesiano, Posicionamento de ferramenta, Controle de 2 eixos.

ABSTRACT

This document aims to describe the design and construction of a high precision Cartesian positioner prototype, which can be used to position the tool of a lathe in order to obtain a high degree of manufacturing quality. This project describes the planning of the mechanism, along with project development, mechanical, electronic and the programming and implementation of closed-loop control. In addition, kinematic and dynamic modeling and analysis will be implemented for the development and validation of control parameters.

Key words: Cartesian positioner, Tool positioning, 2-axis control.

DEDICATÓRIA

Dedicamos este trabalho a Deus e as nossas famílias.

AGRADECIMENTOS

Gostaríamos de agradecer o Prof. Dr. Rodrigo Stoeterau pelo apoio, pelo conhecimento passado e pela estrutura fornecida para o desenvolvimento e montagem dos componentes.

Gostaríamos também de agradecer ao Prof. Dr. Marcilio Alves pelo empréstimo dos equipamentos, os quais possibilitaram o estudo dos resultados obtidos pelo mecanismo finalizado.

LISTA DE FIGURAS

Figura 2.1 - Evolução da usinagem e métodos inovadores desenvolvidos segundo TANIGUSHI (1983).....	15
Figura 2.2 – Recursos que entram na caracterização na retífica de precisão. Adaptado de BRINKSMEIER (2010).....	16
Figura 2.3 – Ferramentas de microlapidação de BRINKSMEIER (2010): CVD...18 coated de GÄBLE (2004), sintered de Haefeli (1997) and electroplated coating Aurich (2009) (da esquerda para a direita).....	19
Figura 2.4 – Exemplos de Máquinas Cranfield e Moore de alta precisão para espelhos.....	19
Figura 2.5 – Exemplos de posicionadores de alta precisão.....	20
Figura 3.1 – Lugar das raízes da guia inferior considerando a restrição de máximo sobressinal.....	22
Figura 5.1 – Exemplo do modelo de fuso de esferas escolhido.....	28
Figura 5.2 – Modelo completo do mecanismo em CAD.....	29
Figura 5.3 – Malha desenvolvida com o módulo MEF do software Solidworks.....	30
Figura 5.4 – Análise de convergência de acordo com o número de elementos.....	30
Figura 5.5 – Deslocamento obtido através da análise estática.....	32
Figura 6.1 – Sistema de medição de posição (ou realimentação de posição).....	33
Figura 6.2 – Sistema de Movimentação.....	34
Figura 6.3 – Detalhe do projeto da Placa de Alimentação.....	35
Figura 6.4 – Sinais de Movimentação.....	36
Figura 7.1 – Diagrama de Casos de Usos.....	38
Figura 7.2 – Diagrama de Componentes.....	39
Figura 7.3 – Resultado a uma entrada de posição sem controle.....	41
Figura 7.4 – Resultado a uma entrada de posição com controle.....	42
Figura 8.1 – Modelagem da guia inferior.....	43
Figura 8.2 – Comparação entre modelo teórico e curva experimental para a guia inferior para uma entrada de torque do motor.....	46
Figura 8.3 – Comparação entre modelo teórico e curva experimental para a guia superior para uma entrada de torque do motor.....	46
Figura 8.4 – Diagrama de blocos do sistema com controlador proporcional.....	47
Figura 8.5 – Diagrama de blocos do sistema com controlador PID.....	48
Figura 8.6 – Comparação entre as duas propostas de controlador sugeridas.....	48

Figura 8.7 – Aquisição de dados considerando entrada impulso de posição na guia superior.....	49
Figura 9.1 – Fluxograma de metodologia de desenvolvimento.....	50
Figura 9.2 – Exemplo montagem do ensaio de posicionamento para a guia inferior.....	51
Figura 9.3 – Equipamentos utilizados para o Teste 1.....	51
Figura 9.4 – Posição guia inferior, controlador desligado.....	52
Figura 9.5 – Posição guia superior, controlador desligado.....	52
Figura 9.6 – Posição guia inferior, controlador ligado.....	53
Figura 9.7 – Posição guia superior, controlador ligado.....	54
Figura 9.8 – Teste de repetibilidade da guia inferior.....	55
Figura 9.9 – Teste de repetibilidade da guia superior.....	56
Figura 9.10 – Teste de linearidade de movimento.....	57
Figura 9.11 – Modelo do método utilizado para verificar a linearidade das guias.....	57
Figura 9.12 – Reta característica guia inferior.....	59
Figura 9.13 – Reta característica guia inferior.....	60

LISTA DE TABELAS

Tabela 3.1 – Comparação entre os tipos de motores de BRITES (2008).....	24
Tabela 3.2 – Classificação de elementos de transmissão de movimento e suas características de acordo com STROETERAU (2004).....	25
Tabela 5.1 – Modelos considerados na escolha da transmissão de movimento.....	27
Tabela 5.2 – Modos de vibração identificados na Análise Modal.....	31
Tabela 8.1. – Variáveis da equação de movimento do sistema.....	44
Tabela 8.2. – Valores dos parâmetros teóricos para curva experimental.....	47
Tabela 8.3 – Parâmetros dos controladores PID implementados.....	49
Tabela 9.1 – Repetibilidade do controlador da guia inferior.....	55
Tabela 9.2 – Repetibilidade do controlador da guia superior.....	55
Tabela 9.3 – Linearidade da guia inferior.....	58
Tabela 9.4 – Linearidade da guia superior.....	59

SUMÁRIO

1.0 Introdução.....	13
2.0 – Estado da Arte.....	15
3.0 – Fundamentação Teórica.....	21
4.0 – Requisitos de Projeto.....	26
5.0 – Mecânica.....	27
6.0 – Eletrônica.....	33
7.0 – Programação.....	37
8.0 – Controle.....	43
9.0 – Validação.....	50
10.0 – Conclusão.....	61
11.0 – Sugestões Para Trabalhos Futuros.....	63
12.0 – Referências Bibliográficas.....	63
Apêndice A – Desenhos De Fabricação.....	66
Apêndice B – Algoritmos Utilizados.....	69

1.0 – INTRODUÇÃO

Posicionadores cartesianos controlados numericamente podem ser aplicados em diversas áreas da engenharia. Essa tecnologia pode ser encontrada na indústria biomédica como por exemplo em máquinas que necessitam de alta precisão no posicionamento de um laser no caso de uma cirurgia oftalmológica. Também pode ser encontrado posicionadores cartesianos, na produção dos semicondutores modernos, já que para o desenvolvimento e inspeção desses componentes, que possuem tamanho extremamente reduzido, há a necessidade de processos de alta precisão no posicionamento de instrumentos de medição. Processos mais tecnológicos são cada vez mais importantes na manufatura. A manufatura por adição, por exemplo, necessita de processos mais baratos e mais eficientes com menor custo energético, para isso há a necessidade de se conseguir desenvolver mecanismos de baixo atrito e alta precisão no posicionamento de ferramentas capazes de auxiliar na fabricação de objetos em escala reduzida.

No caso abordado, o objetivo do projeto foi o de posicionar a ferramenta de um torno com precisão suficiente para que a peça fabricada tenha um alto grau de qualidade de fabricação e um envelope de tolerância bem estreito. Para isso, houve a necessidade de projetar uma máquina que consistirá de dois conjuntos. O da base que possuirá duas guias cilíndricas, cada uma com dois mancais de deslizamento aerostáticos para a diminuição do atrito, sendo que para o acionamento desse conjunto usar-se-á um fuso de esferas recirculantes. Sobre este sistema de base haverá um outro conjunto com duas guias cilíndricas, cada uma com um mancal de deslizamento aerostático e acionamento também por fuso de esferas recirculantes, sendo que está última guia suportará o mecanismo que carregará a ferramenta.

1.2 – Objetivos

Dessa forma estabeleceu-se o objetivo geral do desenvolvimento de um posicionador cartesiano de alta precisão controlado por computador. E para que isso pudesse ser alcançado houve a necessidade, de forma específica, do desenvolvimento do controlador, da malha de controle, da sintonia do controle, do ajuste de ganhos do controlador e da integração mecatrônica dos sensores, controladores, acionamento e parte mecânica.

1.3 – Motivação

Cada vez mais máquinas de precisão são exigidas na sociedade moderna, devido as grandes demandas por equipamentos eletrônicos de maior desempenho e tamanho reduzido. Em contrapartida tem-se o alto custo das máquinas industriais de comando numérico. Assim, é de interesse o estudo e desenvolvimento dessa área para poder deixar mais acessível essa tecnologia.

Esse projeto consiste no estudo e desenvolvimento de um sistema que possa trabalhar com alta taxa de precisão, dentro de especificações mais rígidas, necessárias quando deseja-se trabalhar nas casas dos micrômetros. Assim, esse projeto sumariza a aplicação de tudo que foi ensinado no curso de engenharia mecatrônica, partindo inicialmente do desenvolvimento de uma máquina, seguindo com a sua automatização e controle em mais alto nível.

2.0 – ESTADO DA ARTE

O desenvolvimento de novas tecnologias na usinagem cresceu muito no último século. BRINKSMEIER (2010) constata que as necessidades de peças de alta precisão têm grande variedade de aplicação nas indústrias de ótica, automotiva, comunicações, principalmente aeroespacial, e ainda nas indústrias médicas e de ciências biológicas. Nesses setores, há a necessidade de que sejam produzidas peças com geometria bastante complexa e alto grau de precisão, o que torna muito alto o custo de produção desses componentes. Para que haja a redução no preço, aumento da qualidade e do acabamento das peças e ainda uma maior produtividade são aplicadas soluções derivadas de várias áreas de estudo na engenharia como aquelas equivalentes com a metrologia. O investimento nessas áreas de alta precisão cresceu muito nos últimos quarenta anos, graças a crescente necessidade de componentes e peças manipuláveis em escalas cada vez menores de tamanho as empresas investem e tentam adquirir tecnologias já que essas têm se desenvolvido rapidamente.

Em seu artigo, TANIGUSHI (1983) lista uma série de processos que foram marcos nesse sentido, estimando como se deu o desenvolvimento desta ramificação da engenharia ao longo dos anos através de uma classificação generalizada das máquinas de usinagem, dividindo as em usinagem comum, usinagem de precisão e usinagem de ultraprecisão. Com isso, podemos notar que essa grande evolução em métodos e tecnologias possibilitaram que o maquinário comum de usinagem atualmente tivessem tolerância equivalente a máquinas consideradas ultraprecisas desenvolvidas nas décadas de 1930 e 1940.

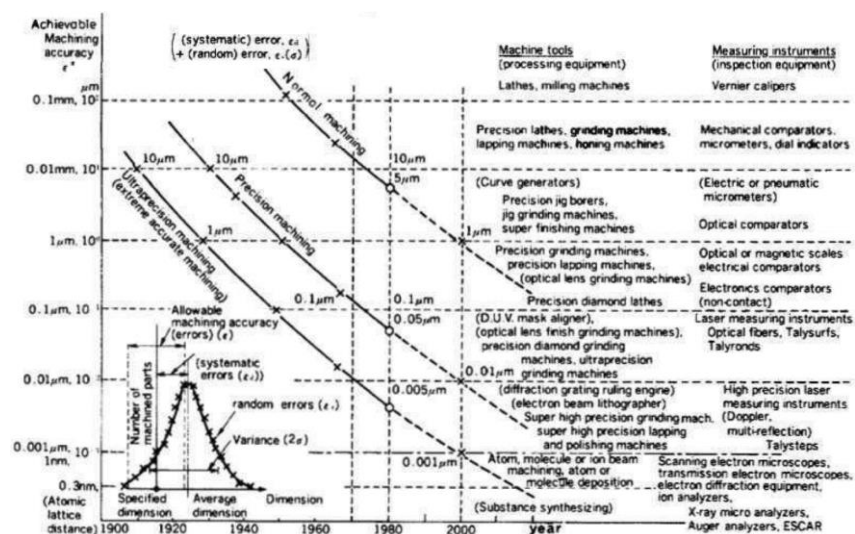


Figura 2.1 - Evolução da usinagem e métodos inovadores desenvolvidos segundo TANIGUSHI (1983).

Ainda segundo TANIGUSHI (1983), o próximo grande desafio da engenharia caminha na direção da nanotecnologia. Isso é exposto em um de seus trabalhos através do exemplo do circuito integrado de um transistor MOS, com dimensões especificadas de 0,1 μm e necessidade de uma acurácia na usinagem de ultra precisão da ordem de 0,005 μm , indicando que a engenharia dos sistemas precisam de um rápido crescimento em rigor e complexidade, onde a solução considerada mais provável seria integrar controle em circuito fechado a máquinas e ferramentas da área.

Mais recentemente, segundo LUO (2005) uma das principais forças motrizes para a fabricação de microcomponentes são os chamados MEMS, sigla para Micro-Electro-Mechanical Systems ou micro sistemas eletromecânicos. O silício é o principal material utilizado na fabricação desses dispositivos, no entanto a grande necessidade de utilização de outros materiais como vidros, cerâmicas, plásticos e metais em áreas como da biologia e da medicina torna extremamente relevante o desenvolvimento de estudos e desenvolvimento de tecnologias de usinagem de ultra precisão. Como exemplificado em WYBRANSKI (2004) a manufatura de alta precisão não somente oferece qualidade e confiabilidade para os produtos convencionais, mas também abre as portas para a possibilidade de produtos inteiramente novos, principalmente em áreas onde componentes mecatrônicos, miniaturizados e de alta *performance* são importantes. Exemplos impressionantes são aplicados em câmeras fotográficas digitais, celulares, equipamentos médicos de mínima invasão, assim como em áreas de biotecnologia ou ainda equipamento de processamento químico.

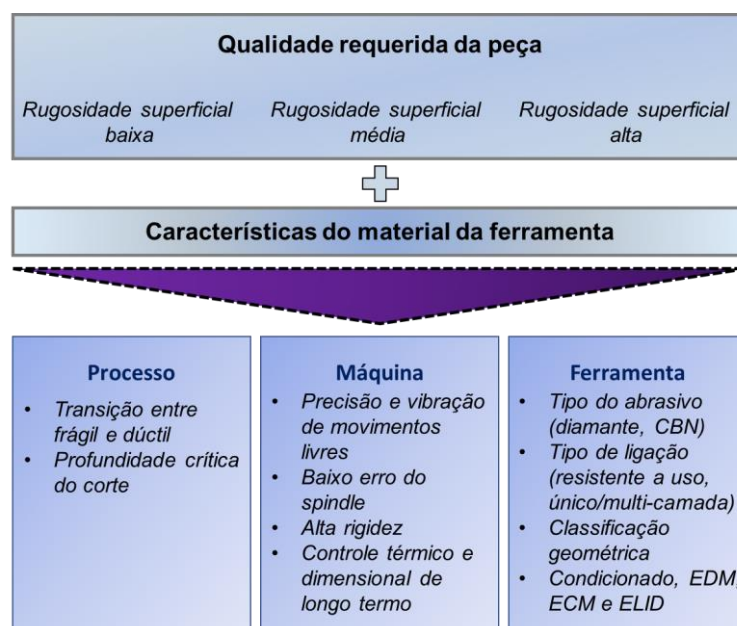


Figura 2.2 – Recursos que entram na caracterização na retífica de precisão. Adaptado de BRINKSMEIER (2010)

De acordo com BRINKSMEIER (2010) várias tecnologias possibilitaram o desenvolvimento da usinagem de ultra precisão, como por exemplo: estrutura de máquinas para usinagem de alta precisão, ferramentas de usinagem, condicionamento de ferramentas e fluidos para usinagem. Para se realizar uma usinagem de ultra precisão de qualidade uma série de fatores são relevantes. Primeiramente, uma precisa e suave vibração e movimento livre de folga. As máquinas Pneumo Ultra 200 e Nanoform 600 estão entre as primeiras capazes de fazer esse tipo de processo de fabricação. Elas possuíam mancais hidrostáticos pressurizados a óleo nas guias lineares. Possuíam ainda o carro da guia linear principal acionado por fuso de esferas com um acoplamento que não influenciava no movimento. Em segundo lugar, é importante que a máquina de usinagem não possua erros síncronos e assíncronos, para isso são utilizados fusos aerostáticos com erros de 5-10 nm e 25-50 nm para erros assíncronos e síncronos, respectivamente. Em terceiro, é necessário que a cadeia estática e dinâmica sejam altamente rígidas, como no caso do CUPE 7 da IBM que possuía 180 N/m de rigidez no laço estático. Por fim, é necessário que haja controle térmico e dimensional, para que as variações de temperatura do ambiente e da ferramenta não influenciem na qualidade da peça, visto que esse é o principal fator que contribui para imprecisão durante a operação.

Com o aumento da precisão e diminuição das tolerâncias, cresceu o desgaste nas pontas de ferramentas graças ao grande aumento de tensão. Inicialmente, estudos em materiais cerâmicos foram levados em conta por M. Sumiya (Toshiba Corporation) como mostrado em MCXEOWN (1982) possibilitando uma rugosidade de 20 nm em superfícies espelhadas com ferramenta de diamante de 5 mm de raio em sua extremidade. Hoje em dia utiliza-se bastante como material abrasivo o diamante devido a sua dureza, tais como em ferramentas diamantadas. No entanto, nesse caso pode haver a formação de átomos de carbono durante a retífica do aço, devido à alta temperatura do processo. Para que isso não acontecesse, foram desenvolvidas ferramentas a base de nitreto cúbico de boro, composto de alta estabilidade química em altas temperaturas.

O mesmo problema possibilitou o desenvolvimento de outras técnicas de retirada de material, principalmente para o polimento de superfícies, como a Emissão Elástica especificada em MORI (1987), que permitiu a retirada de material em nível atômico com o auxílio de grãos abrasivos imersos em líquido. Atualmente, essa tecnologia foi substituída em vários lugares pela técnica de laser de íons como dito em WILSON (1988), a qual não necessita de um posicionamento de alta precisão para retirada de material, mas que, em contrapartida, representa um custo alto pela necessidade de um ambiente controlado em vácuo.

Para processos de grande remoção de material e de dimensões maiores, o desbaste com o auxílio de ferramentas diamantadas foi o principal método desenvolvido ao longo dos anos. Mas isso também não seria possível se não tivessem pesquisas na área de materiais compósitos, já que esses materiais possuem duas ou mais fases com propriedades diferentes e sua união torna a ferramenta melhor por proporcionar a dureza dos grãos abrasivos junto com a propriedade de coesão da matriz. Nos casos de matrizes metálicas, a ferramenta passa a ter propriedades de alta condutividade térmica e boa resistência ao desgaste. Já as matrizes de resina de poliéster ou de epóxi garantem a ferramenta propriedades de alto acabamento, suavização de superfícies e polimento. Além disso são ferramentas que suportam altas velocidades de rotação aliada com a retirada de grande quantidade de material, já que são resistentes a choques e impactos, e ainda a pressões laterais.

Ferramentas de micro lapidação são possíveis de serem encontradas no mercado hoje em dia. De acordo com BRINKSMEIER (2010) essas ferramentas são fabricadas de diferentes formas, tais como matriz de carbetto de tungstênio com grão de carbetto cúbico de boro ou de diamante. Com a possibilidade de fabricar ferramentas de 5m de diâmetro. No entanto, nada disso é eficaz se a estrutura da ferramenta não for eficiente para o determinado processo de acordo com as características do material utilizado, ou ainda se a micro geometria não consegue manter o fio de corte afiado. A não utilização de fluidos de refrigerantes na usinagem também agrava a situação. Esse último reduz o atrito entre a ferramenta e a peça, ajuda na remoção do calor da área de usinagem de forma que o calor não danifique a peça ou a ferramenta. Este cuidado é importante já que o calor pode alterar as propriedades da peça negativamente.

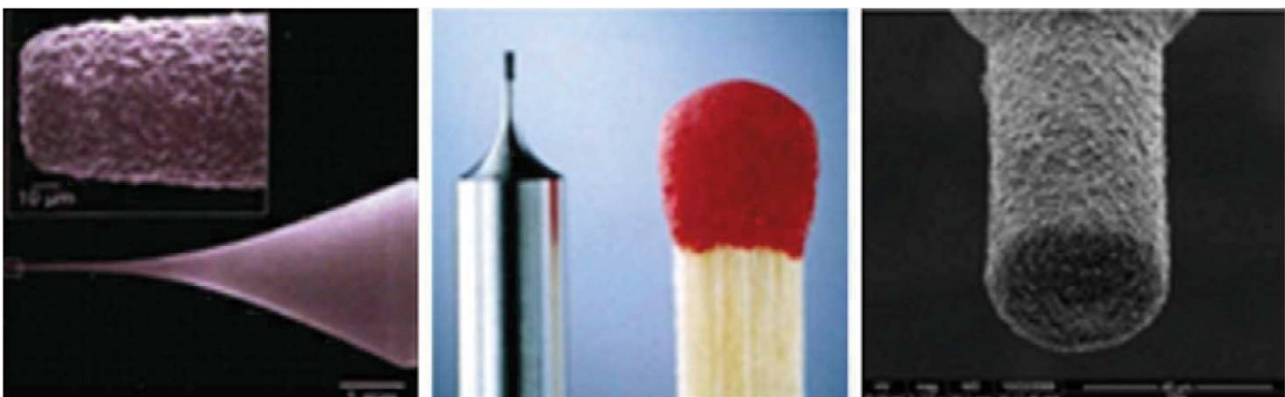


Figura 2.3 – Ferramentas de microlapidação de BRINKSMEIER (2010): CVD coated de GÄBLE (2004), sintered de Haefeli (1997) and electroplated coating Aurich (2009) (da esquerda a direita).

A fim de contornar o problema causado por altas-tensões durante o processo de usinagem, como mostrado em SCHINKER (1987), PUTTICK (1989), BLAKE (1990) e BIFANO (1991), as máquinas Cranfield foram desenvolvidas, sendo pioneiras nesse aspecto como mostrado em WILLS-MOREN (1982) e MCKEOWN (1986), o que possibilitou a fabricação de espelhos de alta precisão.



Figura 2.4- Exemplos de Máquinas Cranfield e Moore de alta precisão para espelhos

Como dito anteriormente, uma significativa evolução no posicionamento de ferramentas em instrumentos de medição ultra precisos andou em paralelo com a usinagem de precisão, graças principalmente ao desenvolvimento de estudos com Laser. Em 1957, Charles Townes, com o auxílio dessa tecnologia, desenvolveu máquinas capazes de medir objetos com alta precisão com sensores de interferência óptica, como mostrado em GARWIN (2003).



Figura 2.5- Exemplos de posicionadores de alta precisão

Portanto, como dito por SHORE (2012): Conclui-se que os engenheiros de ultra precisão possibilitaram grandes avanços científicos. Avanços nessa área, associados a sustentabilidade e grandes investimento, possibilitarão grandes mudanças em tecnologias de grande influência no mundo contemporâneo.

3.0 – FUNDAMENTAÇÃO TEÓRICA

Nesta seção, apresentaremos os principais conhecimentos aplicados durante o desenvolvimento do projeto. Como será mostrado futuramente, a teoria possibilitou o desenvolvimento dos parâmetros aplicados, sendo os mesmos validados e melhorados com o auxílio de testes no posicionador montado.

3.1 – Modelagem de sistemas mecânicos

Inicialmente, para que fosse possível desenvolver um controlador que cumprisse com os requisitos do projeto, o sistema foi equacionado de forma a identificar quais os principais parâmetros que afetam o seu desempenho dinâmico.

Para tanto, com o auxílio do conhecimento exposto em OGATA (2010), o sistema precisou ser descrito no modelo de diagrama de blocos, de modo a identificar a função transferência que rege o comportamento do posicionador desenvolvido.

Como aponta a teoria citada anteriormente, inicialmente aplicou-se a 2ª Lei de Newton de forma a encontrar a relação entre deslocamento da ferramenta e torque de entrada do motor. Dada a relação, a transformada de Laplace é então aplicada (considerando uma condição inicial igual a zero), o que possibilitou encontrar a função transferência que evidencia o comportamento das guias X e Y, que será demonstrada mais adiante:

$$\frac{Y(s)}{T(s)} = \frac{1}{\left(\frac{2\pi}{p}J + M\frac{p}{2\pi}\right)s^2 + B\theta\frac{2\pi}{p}s} \quad (1)$$

3.2 – Controladores PID

Dado o comportamento do sistema e a modelagem obtida, ficou evidente a necessidade do desenvolvimento de uma solução que minimiza-se ao máximo efeito de distúrbios externos a esta equação. Para tanto, de acordo com OGATA (2010), o caminho mais versátil que atenderia os requisitos propostos do projeto seria seguir com o projeto de um controlador Proporcional-Integral-Derivativo. Assim, através da correta calibragem dos parâmetros, conseguiríamos uma resposta rápida, sem sobressinal e com um erro de regime permanente muito abaixo da precisão desejada para o posicionador.

Inicialmente, foi considerado a utilização das regras de sintonia de Ziegler-Nichols para definir as constantes do controlador PID. Para que a estimativa inicial fosse precisa,

o lugar das raízes do sistema foi estudado para de modo que os requisitos de projeto fossem obedecidos, os quais serão discutidos mais adiante.

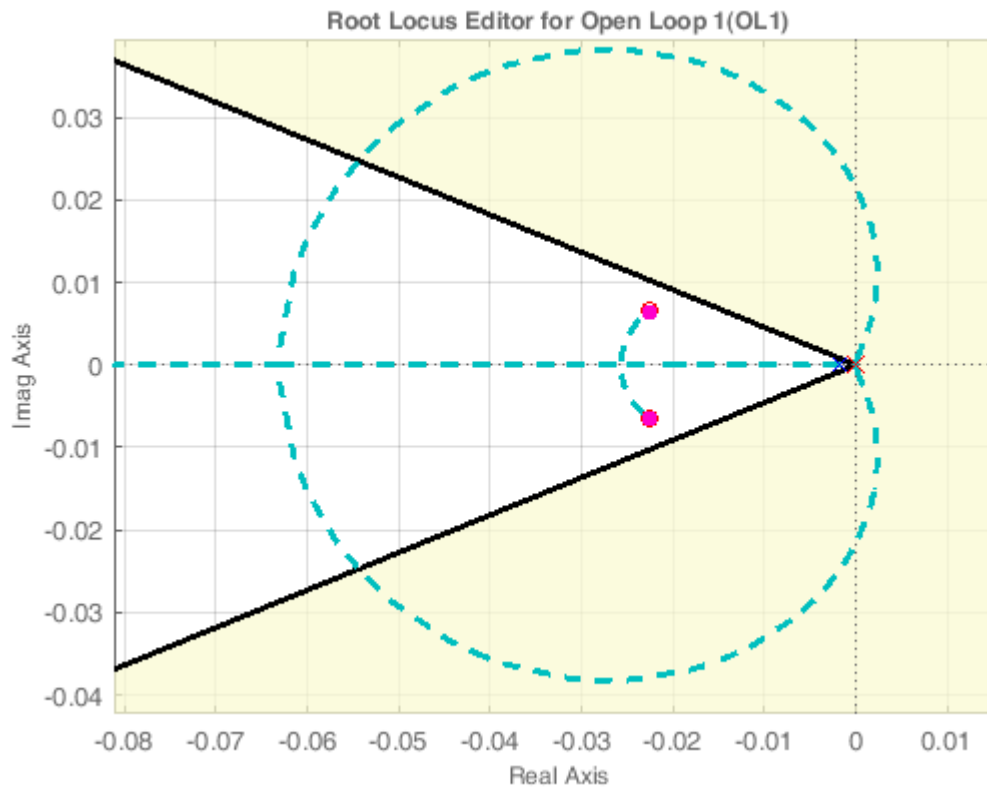


Figura 3.1 - Lugar das raízes da guia inferior considerando a restrição de máximo sobressinal

3.3 – Análise Modal (Método dos Elementos Finitos)

Durante a etapa de projeto mecânico na máquina, o principal critério que possibilitaria a robustez da solução adotada não seria a resistência a cargas estáticas, como é levado em conta em diversos projetos de portes maiores. Evidentemente, simulações nesse sentido também foram realizadas, mas como se trata de um mecanismo que precisa manter sempre a posição da ferramenta independente da situação, as vibrações acabam tendo um papel essencial na definição da estrutura mecânica.

Portanto, as velocidades de operação foram definidas de acordo com as frequências modais, encontradas com o auxílio do Método de Elementos Finitos. As equações utilizadas para cada elemento seguem a hipótese de Rayleigh para amortecimento. Com isso, as condições de manuseio da ferramenta que precisam ser evitadas foram definidas.

3.4 – Análise de Sistemas de Atuação

No desenvolvimento desse projeto foi necessário estudar 3 tipos principais de motores que poderiam ser utilizados para fazer a movimentação dos sistemas. Para atuação em sistemas de precisão podem ser utilizados os seguintes tipos: motor de passo controlado, motor de corrente contínua e servo motores. Cada um deles possui vantagens e desvantagens em relação aos outros como será explicado abaixo.

Nos motores de passo, o movimento ocorre por meio da conversão de pulsos elétricos em movimentos angulares determinados conhecidos por passo. Por meios da atuação do controlador do motor determinando a sequência de acionamento das bobinas do estator há o movimento do rotor e assim é possível controlar o ângulo que o dispositivo irá rotacionar e a direção dessa rotação. Já a velocidade de rotação é controlada por meio da frequência dos pulsos elétricos que indicam pro controlador a velocidade de comutação entre as bobinas.

Já nos motores de corrente contínua, o movimento ocorre por meio da força eletromagnética que um condutor, por onde uma corrente esteja passando, sofre ao ser submetido a um fluxo magnético. Esses motores são divididos em dois tipos: com escovas e sem escovas. No caso do primeiro, há a necessidade do anel comutador que é responsável por fazer a inversão do sentido da corrente da armadura: este é o enrolamento das bobinas do rotor. No segundo caso, os motores chamados brushless têm essa inversão da corrente da armadura feita de forma externa por meio de inversores de correntes dentro dos seus controladores. Os dois tipos possuem o enrolamento do estator que é responsável por criar o campo magnético dentro da estrutura.

No caso dos Servomotores, o movimento ocorre por meio da utilização, geralmente, de um motor de corrente contínua. No entanto, esse sistema trabalha em malha fechada pois, dentro do motor, existe sensores e um controlador que verifica se o motor chegou na posição desejada e assim controla a alimentação do mesmo para que ele fique mais devagar a medida que esteja mais perto da posição desejada.

Tabela 3.1 – Comparação entre os tipos de motores extraído de BRITES (2008)

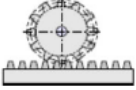

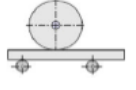
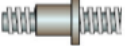
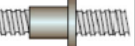


	Motor CC	Motor de passo	Servomotor
Velocidade	Alta	Baixa	Média
Torque	Alto	Médio	Alto
Facilidade de Controle	Fácil	Médio	Difícil
Precisão	Baixa	Alta	Muito Alta
Durabilidade	Média	Alta	Média
Manutenção	Sim	Não	Sim
Custo	Médio	Baixo	Alto

Ainda segundo BRITES (2008) os motores de passo quando trabalham em altas velocidades costumam perder a precisão do seu posicionamento, o que é sua característica mais importante. Apesar disso, esse modelo consegue se manter fixo na posição final mesmo estando ligado, diferente de um motor DC, onde é preciso desligar o mesmo quando este atinge a posição: esse controle é feito pelo servomotor. Motores de passo e motores CC trabalham em malha aberta, portanto a saída do acionamento não é utilizada para realimentar o controle o que facilita a utilização. No entanto, o motor de passo ainda é mais preciso que o CC devido a relação entre o pulso de controle e o passo de rotação do motor. Por fim, um fator que levou a utilização do motor de passo nesse projeto foi seu menor custo em comparação aos demais modelos citados.

3.5 – Análise de Sistemas de Transmissão de movimento

Como visto anteriormente, os motores têm seu acionamento por meio da rotação dos seus rotores, no entanto os deslocamentos dos eixos X e Y são lineares, o que gera a necessidade de conversão desse movimento. Para isso, podem ser utilizados diversas formas com descrito por STOETERAU (2004) na tabela à seguir.

Tabela 3.2 – Classificação de elementos de transmissão de movimento e suas características de acordo com STOETERAU (2004)

Tipo	Pinhão / Cremalheiras	Polias	Rodas de Atrito	Fusos			
				Trapezoidais	Esferas Recirculantes	Epicicloidais	Hidrostáticas
Requisitos							
Custo	Baixo	Baixo	Alto	Baixo	Médio	Alto	Muito Alto
Vibrações	Alto	Baixo	Baixo	Médio	Baixo	Baixo	Baixo
Exatidão no Posicionamento	Baixo	Médio	Muito Alto	Baixo	Alto	Alto	Alto
Capacidade de transmissão de Forças/Torque	Alto	Baixo	Muito Baixo	Alto	Alto	Alto	Alto
Facilidade de Manutenção	Alto	Alto	Baixo	Médio	Médio	Baixo	Baixo
Atrito	Baixo	Baixo	Alto	Alto	Baixo	Baixo	Muito Baixo
Vida útil	Médio	Alto	Baixo	Médio	Alto	Alto	Muito Alto

Para desenvolver um posicionador cartesiano de alta precisão é necessário que essa conversão seja de baixa vibração, alta precisão de posicionamento, alta capacidade de transmissão de torque e baixo atrito. Assim a transmissão que melhor se ajusta a essas exigências é o fuso de esferas recirculantes.

4.0 – REQUISITOS DE PROJETO

4.1 - Diretrizes do sistema

Inicialmente, definimos as diretrizes do projeto baseado nos objetivos anteriormente dispostos. Visto que se trata de um posicionador de alta precisão, levamos em consideração a teoria exposta em TANIGUSHI (1983) para obter métricas que servirão como bases comparativas. Levamos em consideração que se trata de um torno com método de usinagem convencional, ou seja, uma ferramenta que garante remoção de material considerável com ponta em aço. Além disso, estabelecemos, baseado nas limitações geométricas da mesa em que o mecanismo será posicionado, as características espaciais da área de trabalho e, por fim, considerando os raciocínios expostos em FERRARESI (1977) para minimizar o desgaste em uma ferramenta de metal duro, definimos a velocidade de avanço das guias para as duas direções de movimentação.

- Precisão entre 1 μm e 10 μm
- Área de trabalho mínima de 0,0625 m²
- Velocidade de trabalho menor que 15 m/min
- Baixo atrito nas guias
- Baixo Custo

Para garantir o baixo custo, uma das opções será implementar um controle em malha fechada, que, com a correta integração de sensores e acionamentos, possibilitaria simplicidade sem comprometer a qualidade de fabricação quando associado a um *spindle* que posicione a peça corretamente.

A velocidade também foi escolhida de modo que não seja alta, fazendo com que vibrações passadas ao sistema não afetem seu desempenho em futuras operações. Vale ressaltar que esse ponto será abordado mais à frente quando nos referirmos a Análise Modal, que garantirá que as excitações cíclicas existentes poderão ser desprezadas visto os materiais utilizados, não havendo a necessidade do uso de isoladores, como sugere MEKID (2011) para o exemplo de manufatura microeletrônica.

5.0 – MECÂNICA

5.1 - Direção de movimentação

Como se trata de um mecanismo onde o principal objetivo é garantir a precisão no posicionamento da ferramenta, para que o controle seja bem desenvolvido, foi escolhido a movimentação linear em eixos x e y . Para que as cargas dinâmicas de atrito fossem desprezadas, mancais aerostáticos serão utilizados para apoiar as mesas que se movimentarão, sendo os mesmos ligados a uma linha de ar comprimido durante a operação do mecanismo. Guias em aço SAE 1020 foram retificadas, garantindo uma folga média de 0,11 mm radialmente com os mancais acoplados.

5.2 - Transmissão de potência

Visto as limitações geométricas da mesa e da área de trabalho anterior, filtramos algumas opções possíveis de mancais de modo que a máquina não atinja alta complexidade de fabricação. Nos atentaremos principalmente ao acionamento por fuso de esferas recirculantes de modelo RFSIN, pelo fato de possuírem passo de 5 mm/volta, sendo esse o menor valor encontrado entre os demais modelos, além de obedecer o comprimento necessário para garantir a correta montagem da guia superior e inferior do mecanismo.

Tabela 5.1 – Modelos considerados na escolha da transmissão de movimento

<i>Modelo</i>	<i>Diam. Fuso (mm)</i>	<i>Passo</i>	<i>Diam. Esferas</i>	<i>Carga Din. (kgf)</i>	<i>Carga Estát. (kgf)</i>	<i>Diam. Base (mm)</i>
PS1405A	14	5	3175	515	990	11,42
PS1605A	16	5	3175	570	1130	13,42
PS2010A	20	10	6350	1100	2200	16



Figura 5.1 – Exemplo do modelo de fuso de esferas escolhido

Para garantir o correto posicionamento do fuso em relação às mesas do mecanismo, escolhemos mancais do tipo FK12 do mesmo fabricante que contêm duplo rolamento de contato angular, garantindo maior resistência a aplicação de cargas axiais e radiais. Além disso, o modelo escolhido obedece às limitações de diâmetro do fuso e de encaixe nas paredes do mecanismo, as quais serão mostradas mais à diante. Os acoplamentos foram escolhidos levando em conta o dimensionamento para o torque do motor de passo NEMA 23 escolhido, o qual a escolha será detalhada mais adiante ao tratarmos de todos os componentes eletrônicos do sistema.

5.3 - Modelagem em CAD

5.3.1 - Ferramentas utilizadas

Visto os componentes anteriormente citados, escolhemos desenvolver as peças restantes e a montagem do componente para a conferência das tolerâncias gerais no software *Solidworks® 2016*, por ser de fácil manuseio, além de possibilitar a criação de desenhos de fabricação de forma rápida, visto que ele interpreta as dimensões pedidas pelo usuário com funções específicas.

Além disso, o mesmo software consegue facilmente identificar interferências, além de possuir um módulo de montagem baseado em posicionamentos relativos de peças bem simplificado. Por fim, destaca-se a presença do módulo de MEF, que com o correto desenvolvimento das condições de contorno, possibilitou o desenvolvimento da Análise Modal, que será explicada mais a diante.

5.3.2 - Montagem em CAD

Dito isso, baseado nas peças já manufaturadas fornecidas ao nosso trabalho, desenhamos as mesmas no software citado a fim de verificar se modificações nestas seriam necessárias. Com isso, notamos a necessidade de fabricar novos suportes aos mancais dos fusos de esferas, que se tornaram maiores em relação ao utilizado anteriormente, além da necessidade de modificar o tamanho das guias, de forma que as mesmas não se movimentassem durante a operação do posicionador, garantindo uma área de trabalho total de 0,07 m² aproximadamente. A Figura 5.2 mostra o mecanismo finalizado em CAD.

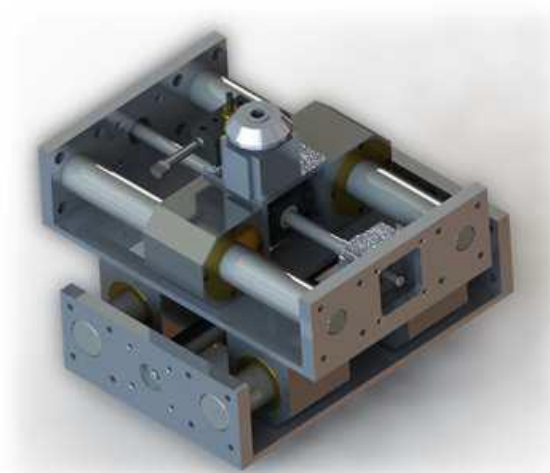


Figura 5.2 - Modelo completo do mecanismo em CAD

5.3.3 - Análise Modal com auxílio de MEF (Método dos Elementos Finitos)

Uma vez desenvolvida a montagem completa, foi separado do restante da mesma a estrutura que receberá as excitações de forma a não obter valores distantes da realidade ao considerar elementos que forneceram vibrações ao sistema. Diversas malhas com parâmetros distintos foram desenvolvidas, e a convergência foi avaliada de forma a obter um resultado confiável com baixo custo computacional durante o processamento das simulações. A Figura 5.3 evidencia a malha que foi escolhida ao fim das simulações

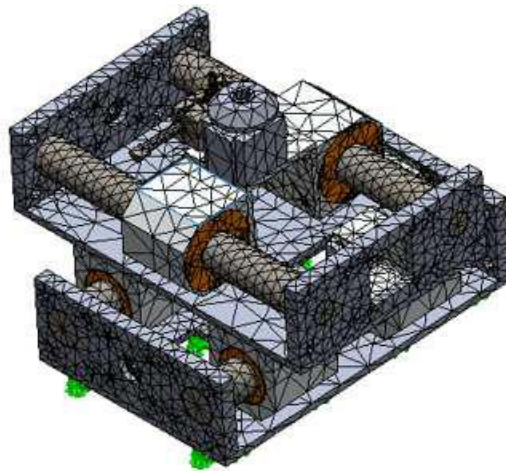


Figura 5.3 - Malha desenvolvida com o módulo MEF do software Solidworks 2016

Considerou-se como condição de contorno a base presa à mesa onde o mecanismo será montado nos furos determinados. Além disso, a mesa em que ocorrerá a montagem tem massa muito maior que o mecanismo, logo esta não foi considerada durante essa análise. Com isso, foi feita uma análise de convergência a fim de avaliar a influência do refinamento das malhas para garantir a convergência dos 5 modos de vibração. Mesmo sabendo que a máquina operará em baixas velocidades e, portanto, baixas frequências, diversos modos foram simulados para garantir a validade dos resultados, como exposto na Figura 5.4.

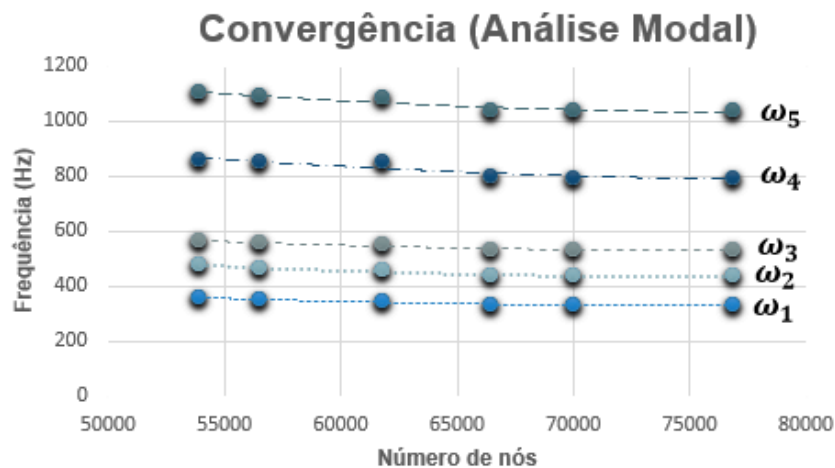
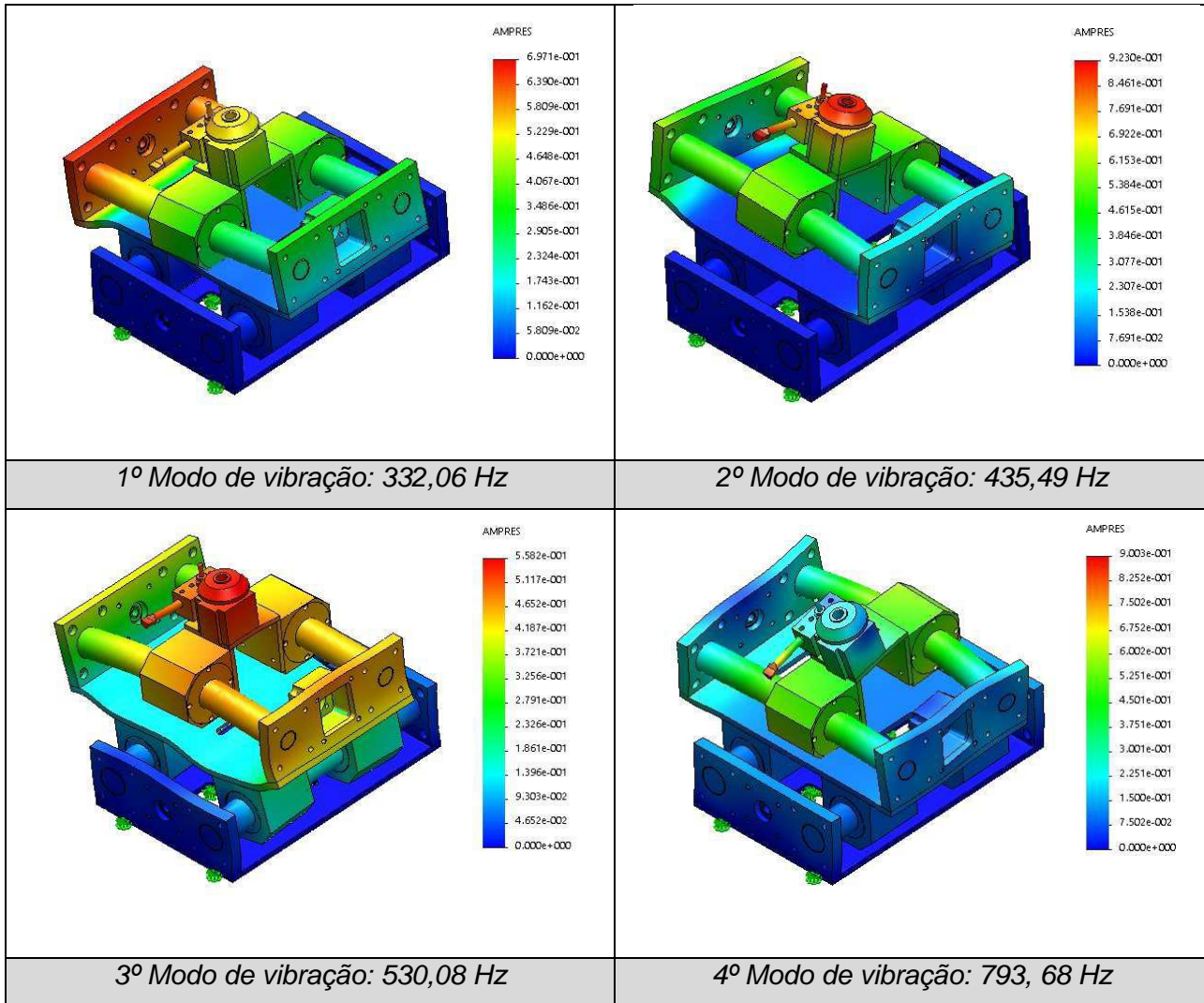


Figura 5.4 – Análise de convergência de acordo com o número de elementos

Os modos obtidos então foram analisados, sendo todos os considerados nessa análise de alto valor, como mostrado na Tabela 5.2. É importante ressaltar que, além dos modos de ressonância terem valores altos, o primeiro modo de vibração teve seu maior deslocamento longe da ferramenta, mostrando que, caso mantenhemos a operação do fuso em velocidades que estejam nessa faixa de frequências, não teríamos problemas de

posicionamento ocasionados por excitações cíclicas. Além disso, visto a velocidade de movimentação escolhida para o fuso em questão, atendendo os requisitos citados anteriormente como explicado no capítulo de eletrônica, atingimos uma frequência máxima de 1 Hz, valor que corresponde à apenas 0,3% do primeiro modo de ressonância.

Tabela 5.2 – Modos de vibração identificados na Análise Modal



Por fim, uma análise estática foi elaborada para avaliar a influência da força de corte de 50 N sobre o deslocamento da ferramenta. A mesma foi definida baseando-se na velocidade de acionamento das guias. Simplificações foram realizadas para que a simulação pudesse convergir: o porta ferramentas foi assumido como rígido, assim como o fuso de esferas.

A força definida anteriormente foi aplicada no local onde se encontraria a ponta da ferramenta, e como condição de contorno foi levado em conta que a mesa está preso em uma base rígida, o que aproxima o que acontece na realidade.

Realizado o estudo, ficou evidente que no pior caso o deslocamento da ferramenta foi de 4,08 μm . Conclui-se então que o erro de posicionamento causado pelo esforço de corte é pequeno quando comparado com a precisão desejada para o mecanismo.

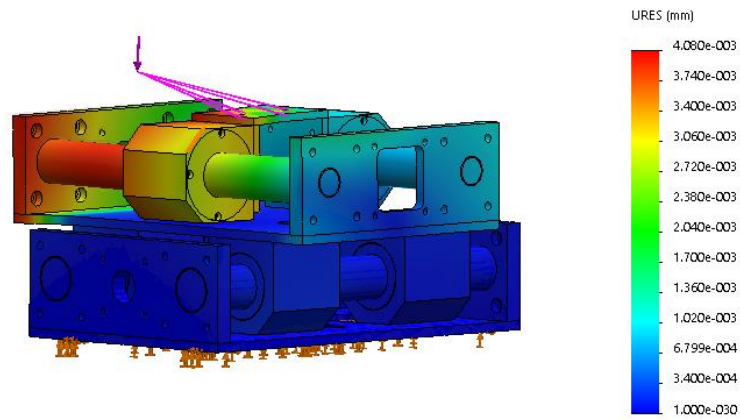


Figura 5.5 – Deslocamento obtido através da análise estática

6.0 – ELETRÔNICA

A eletrônica envolvida nesse projeto é composta de 5 partes. O primeiro componente é a placa Due do processador ARM-32 bits que recebe os dados e processa e envia os comandos para os drives dos motores. O segundo componente é a placa que alimenta os sensores das réguas magnéticas e envia os sinais para o cálculo da posição para a placa do processador. O terceiro componente é a placa FD-RAMPS que faz a alimentação dos motores, e que possui os drives de acionamento. O quarto componente são os dois motores de passo controlados. E por fim o quinto componente é a alimentação que vai fornecer energias para os motores. Para uma melhor compreensão e detalhamento essas partes foram agrupadas em dois sistemas, o primeiro sendo o Sistema de Sensoreamento, que é composto pelas duas primeiras placas, e o segundo o Sistema de Atuação, que é composto pelos três últimos componentes supracitados.

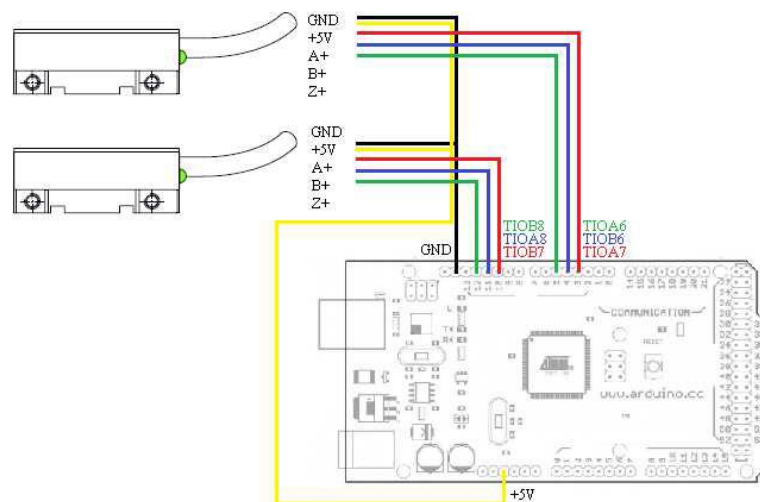


Figura 6.1 – Sistema de medição de posição (ou realimentação de posição)

Para medir a posição foi utilizado dois sensores LM13IC, que são sensores magnético lineares. Eles são alimentados com tensão entre 4,7 V a 7 V e consomem 35mA, por isso não foi necessário utilizar uma alimentação externa já que a placa Due consegue fornecer a tensão de 5 V com corrente máxima de 800mA. Já os níveis dos sinais A, B e Z são maiores ou iguais a 2,5 V no seu nível alto e menores ou iguais a 0,5 V em seu nível baixo, com isso não será necessário utilizar nenhum tipo de conversão nesse sinal já que a placa Due trabalha com tensão máxima de 3,3 V. Esses sensores trabalham com a leitura de 3 sinais na régua magnética, sendo que dois desses sinais (A e B) são vistos como ondas quadradas defasadas. O sinal Z é um sinal de referência a

cada 2 mm da régua. Com isso é possível calcular a distância que foi percorrida em cada eixo.

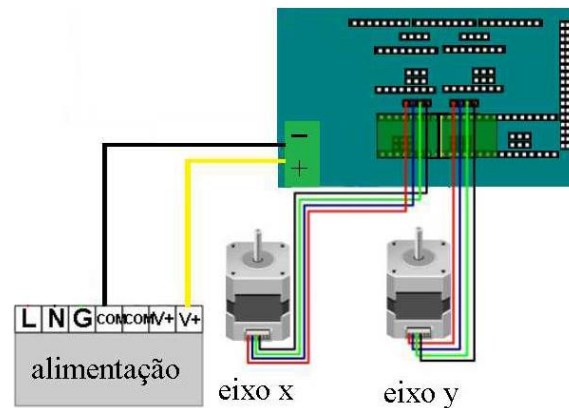


Figura 6.2 – Sistema de Movimentação

O sistema de atuação é composto primeiramente pela placa Ramps-FD que é uma placa que já está no mercado desde 2014 e é bastante utilizada em máquinas CNC. Sua principal característica é ser compatível com a placa Due que, por trabalhar com 3,3 V, poderia ser danificada se fosse utilizado outro modelo. A placa Ramps-FD pode ser alimentada com 12 V ou 24 V e consegue fornecer até 10 A para os motores de passo. Ela também é compatível com os principais Drives do mercado. O Drive que será utilizado é o DRV8825, o qual trabalha com micropasso e possui resolução de até 1/32, operando com tensões de 8,2 V a 45 V, podendo fornecer até 2,2 A. Os motores que serão utilizados são motores de passo NEMA 23 com torque de 2.35 Nm. Sua alimentação é de 8,6 V e necessita de uma corrente de 2 A. Por fim, será utilizado uma fonte comum de 12 V e corrente de saída de 10 A. Os dois sistemas são unidos por uma placa que possibilita alimentar e utilizar os pinos para a aquisição dos sinais dos sensores LM13, já que a placa FD-Ramps não possui entrada para leitor desse tipo de sensor. Essa placa foi desenvolvida utilizando-se o software *DipTrace*® na versão 3.2.0.1, o qual foi necessário para gerar os arquivos de fresamento da placa de circuito impresso.

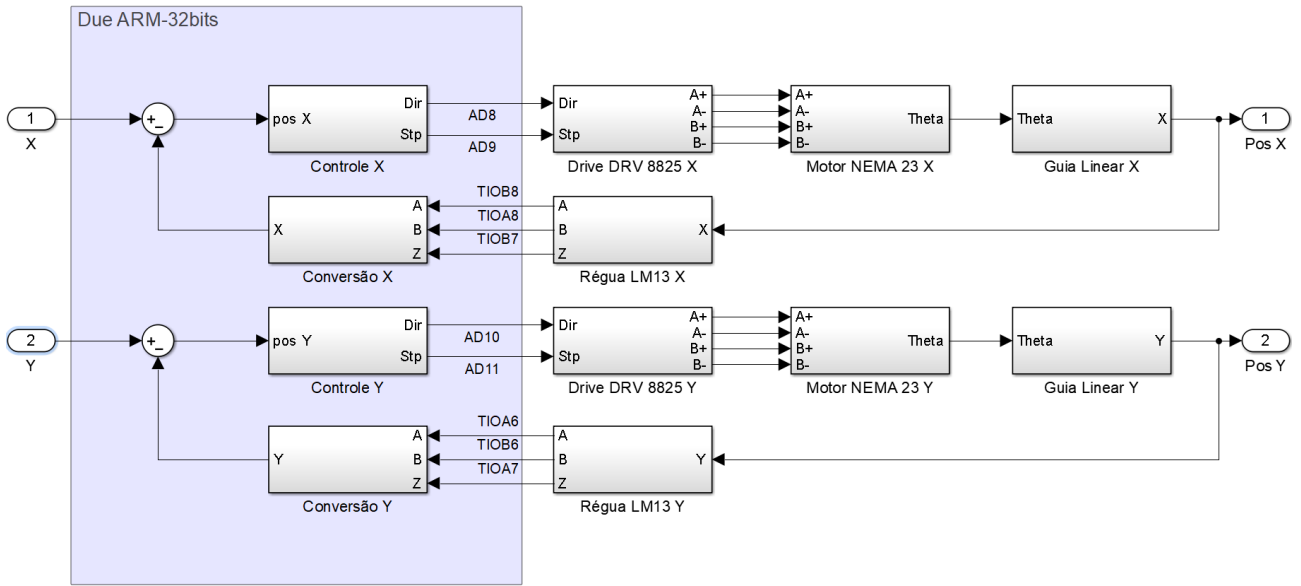


Figura 6.4 – Sinais de Movimentação

7.0 – PROGRAMAÇÃO

O desenvolvimento da rotina que foi embarcada no microprocessador se deu inicialmente com a definição dos requisitos que definem a atuação do programa. Com isso pode-se definir a arquitetura do programa e assim entender quais componentes deveriam enviar e receber sinais e como esses sinais deveriam ser tratados. Por fim, mostrou-se alguns resultados obtidos por essa rotina desenvolvida de forma a demonstrar se correto funcionamento.

7.1 – Projeto da Rotina embarcada

7.1.1 – Requisitos

No caso do sistema embarcado em questão, é necessário que haja um computador para que se possa desenvolver uma Interface Homem-Máquina (IHM) de forma a facilitar que o usuário determine os comandos e configurações que houverem. O controlador deve fazer os cálculos de controle de posição por meio da interpretação dos sinais de entrada vindos dos *encoders* magnéticos e envio dos sinais de comando para os drives dos motores.

No entanto, o sistema precisa ser modelado. Para isso foi necessário desenvolver um segundo programa que seja capaz de acionar o sistema com uma entrada conhecida e fazer a leitura e tratamento do sinal de saída dos sensores de posição. Com isso, foi desenvolvido dois modos de operação para acionamento do sistema: Modo de Aquisição da Curva do Sistema e Modo de Controle Automático.

No Modo de Aquisição da Curva do Sistema é possível dar ordens de entrada nos drives de forma que o motor percorra uma quantidade estipulada de passos. Conforme o movimento ocorre, o *encoder* faz a aquisição dos dados de posição. Esses dados são enviados para o processador que faz o cálculo da posição e manda para a IHM para que possa ser utilizada pelo usuário. Ou seja, determina se que o motor faça duas revoluções no sentido horário e duas revoluções no sentido anti-horário e a medida que os sensores fazem a aquisição dos dados de movimento, o microprocessador faz os cálculos da distância percorrida. Com isso é possível aplicar os estudos de controle para poder fazer uma regressão da curva do sistema para uma curva de primeira ou segunda ordem, conforme a teoria de controle.

Já no Modo de Controle Automático é possível determinar a posição desejada, e o microprocessador passa a fazer o cálculo de controle por meio dos parâmetros de Controle PID previamente configurados. Assim, novos comandos de posição são enviados para os drives a medida que o motor cumpre os deslocamentos determinados de forma a

chegar na posição final. Os dados adquiridos no sensor de posição são tratados pelo processador e agora são enviados para a IHM e voltam como realimentação para o controlador fechado a malha de controle. Ou seja, agora determina se uma posição, como por exemplo 10 mm, que é o equivalente as duas rotações no modo de aquisição. O sistema se desloca para aquela posição e espera pela determinação de uma nova posição.

7.1.2 – Estrutura e Arquitetura

O Sistema precisa receber alguns comandos e configurações do usuário, necessitando responder adequadamente a esses comandos. Para isso, a rotina utiliza uma arquitetura que separa as informações. O projeto foi estruturado em duas partes: uma Interface Homem-Máquina (View) e um Microprocessador (Controller). A plataforma utilizada para programação e visualização dos dados foi o Arduino Software (IDE)

7.2 – Componentes da Programação

O desenvolvimento da rotina deve se dar seguindo os requisitos determinados. Portanto, o diagrama de Casos de Uso foi necessário para determinar como os componentes deveriam interagir com o controlador. Para isso, o usuário, utilizando a IHM, pode fazer as seguintes funções: ativar o controlador, ler a posição, enviar comando de movimento manual e ler dados. Já o drive receberá os sinais de comando do controlador. No caso do *encoder*, ele é quem enviará os dados de posição para que o controle faça os cálculos necessários.

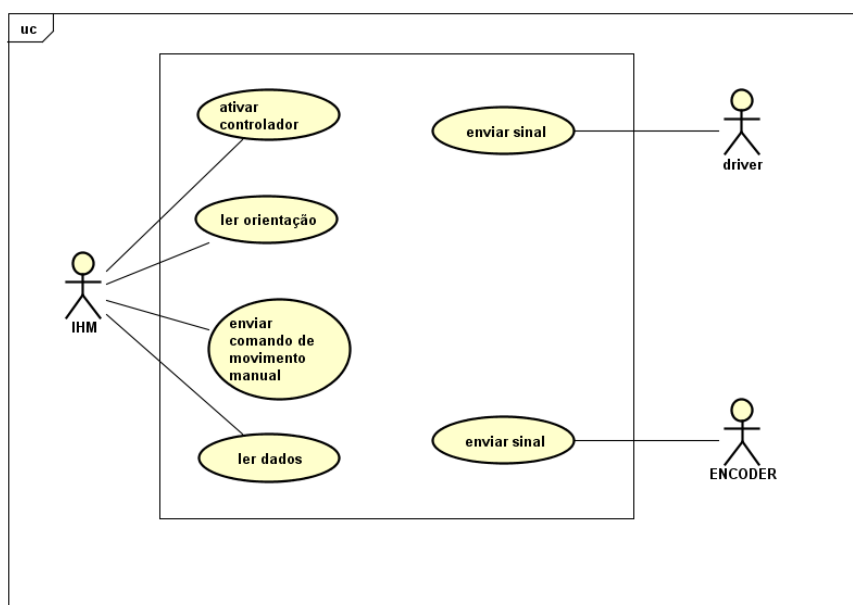


Figura 7.1 – Diagrama de Casos de Usos

Com isso foram determinados quais componentes deveriam ser incluídos na programação do sistema. Assim a programação da rotina foi dividido como mostra a figura 7.2, tendo seus componentes descritos a seguir.

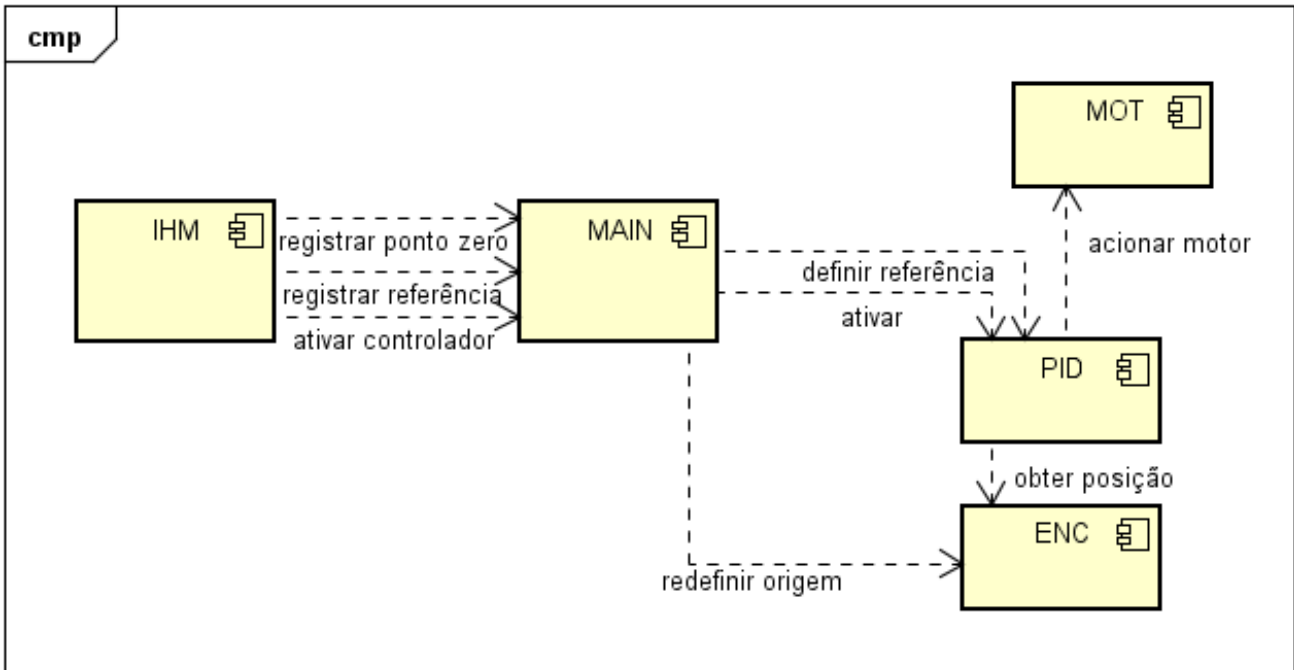


Figura 7.2 – Diagrama de Componentes

7.2.1 – COMPONENTE MAIN

A main tem por função controlar os demais componentes, e assim gerenciando a execução das rotinas que são processadas e executadas durante o funcionamento do sistema.

7.2.2 – COMPONENTE IHM

A IHM é muito importante pois ela se comunica via porta serial com o computador por onde o usuário pode verificar o correto funcionamento do programa e do sistema, mudar configurações e também desenvolver a programação. As principais funcionalidades que esse programa traz é o registro do ponto de início, determinação do ponto final e ativação do controlador.

7.2.3 – COMPONENTE ENC

A função *encoder* é exercida por meio de uma biblioteca chamada na main. Ela funciona em paralelo e sempre que ocorre uma interrupção vista como a mudança de estado lida em uma das portas em que o *encoder* está conectado, é calculado se o movimento foi de avanço ou de atraso e com isso soma se ou subtrai se do contador de posição o valor de um pulso. Dessa forma, a distância pode ser calculada.

7.2.4 – COMPONENTE PID

A função PID é efetuada por uma biblioteca que calcula o sinal de saída de acordo com o valor de entrada enviado pela main, que foi calculado a partir da resposta do *encoder* e a partir dos parâmetros de controle que foram configurados na main pelo usuário.

7.2.5 – COMPONENTE MOT

A função Motor é executada por uma biblioteca que controla os drives dos motores de acordo com os sinais de saída do controlador. Essa biblioteca envia por meio das portas em que o drive está conectado os sinais de habilitar o motor, direção e quantidade de passos que devem ser percorridos.

7.3 – Rotina do Sistema

7.3.1 – Estrutura do View

Como utilizou se basicamente as bibliotecas básicas que já estão incluídas com a instalação do programa da placa. Para visualizar os dados de interesse foi utilizado o comando *Serial.print()*. Para isso foi necessário configurar uma porta serial, no caso a 9600. Essa configuração se dá pelo comando *Serial.begin()* e assim é possível ler a posição calculada pelo controlador na função serial monitor dentro da aba de ferramentas do programa do Arduino.

7.3.2 – Estrutura do Controller

Aqui foi necessário utilizar quatro bibliotecas novas, *PID_v1*, *BasicStepperDriver*, *Encoder* e *DueTimer*. Cada uma dessas bibliotecas tem uma função importante no desenvolvimento do controlador. Inicialmente são definidos o passo do motor, a porta do sinal de direção, a porta do sinal de quantidade de passos que o motor deve dar, e a porta do sinal de habilitar o motor. Para que isso ocorra é necessário utilizar o comando *BasicStepperDriver stepper()*. Em seguida configurou se as portas de entrada do sinal do *encoder* por meio do comando *Encoder myEnc()*. E por fim configura se o controlador PID por meio da função *PID XaxisPID()*.

Após isso é preciso inicializar algumas funções para que as bibliotecas possam começar a funcionar adequadamente. Primeiramente utiliza se o comando *stepper.begin()* para inicializar o controle do motor de passo e para indicar o RPM desejado e qual tamanho de micropasso está sendo utilizado. Em seguida inicializa se a interrupção com a

função `Timer.attachInterrupt().start()`, aqui se configurado uma função de leitura do *encoder* para que os dados de posição sejam calculados. Por fim, é inicializado o controlador PID no modo automático e de acordo com a taxa de aquisição desejada, isso é feito pelas funções `XaxisPID.SetMode()` e `XaxisPID.SetSampleTime()`

Assim é possível utilizar a função `myEnc.read()` para calcular a posição do sistema. Essa posição é normalizada por meio da multiplicação por 0,04 vezes o valor do *microstep* que está sendo utilizado. Esse valor é passado para a IHM por meio do comando `Serial.println()` e também é levado como entrada para o cálculo do controlador. A função `XaxisPID.Compute()` é utilizada para calcular a saída do controlador a partir dos dados de entrada e por fim o movimento do motor ocorre com o envio do comando `stepper.move()` uma vez que ele deve mover o quanto o PID determinar.

7.4 – Resultado da Programação

Com os programas utilizados foi possível obter vários dados do sistema que deseja se fazer o controle em malha fechada. Primeiramente utilizou se o programa `AquiCur.ino` que encontra se no anexo C item 1. Com ele foi possível fazer a aquisição dos dados de saída *encoder* linear e assim gerar o gráfico de resposta do motor devido a uma entrada de indicação de posição

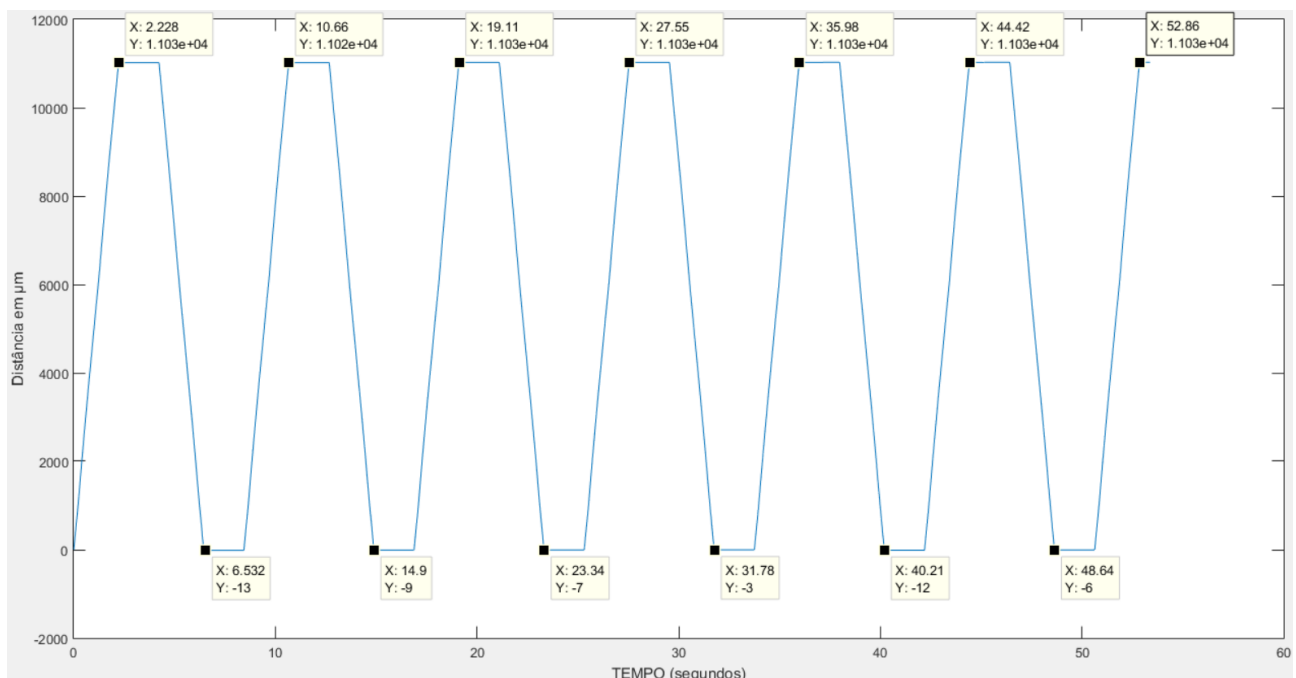


Figura 7.3 – Resultado a uma entrada de posição sem controle

Já com o programa `TF1v4.ino` foi possível implementar o controlador e obter a curva final do sistema. A medida que se alteram os parâmetros de controle é possível

verificar a mudança da resposta do sistema a uma entrada de posição final com o sistema se comportando com sobressinal maior ou ainda com a alteração do tempo de chegada ao regime estático, podendo este estar com erro estático ou não. Pode-se então confirmar que o controle está funcionando corretamente.

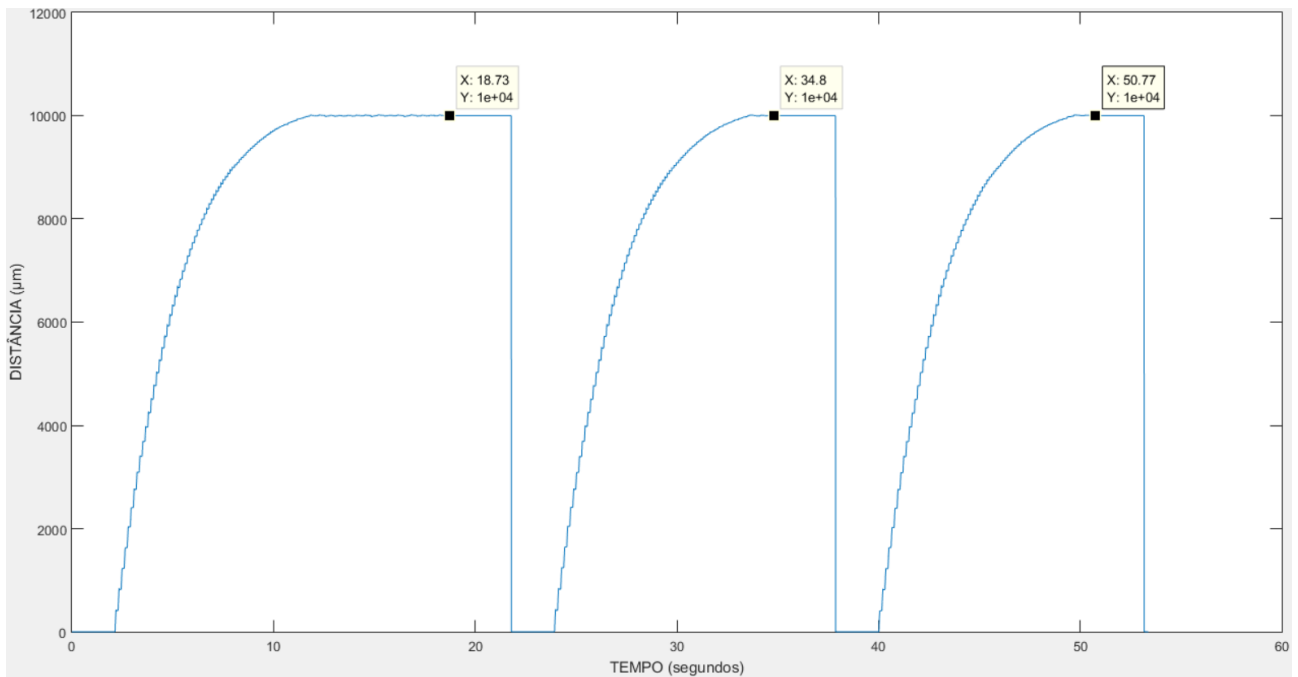


Figura 7.4 – Resultado à uma entrada de posição com controle

8.0 – CONTROLE

8.1 – Modelagem do sistema

Dado o sistema mecânico desenvolvido anteriormente, partimos para a modelagem do sistema para entender quais os principais parâmetros que afetam o controle que será desenvolvido, sendo o mesmo mostrado mais à frente.

As equações diferenciais que definem a dinâmica do sistema foram desenvolvidas de acordo com as forças atuantes nas mesas. Vale lembrar que se trata de um eixo de movimentação para cada mesa e, portanto, duas malhas de controle separadas. Para que os parâmetros reais sejam definidos precisamente, analisaremos como se comporta uma entrada degrau no sistema dado um torque no motor em malha aberta. Com isso, calibraremos a curva teórica do sistema dada o comportamento real das guias, em vez de medir cada parâmetro da equação separadamente com instrumentação de alto custo. É importante salientar que este método será aplicado nos dois eixos de movimentação, sendo diferentes os parâmetros encontrados em cada caso.

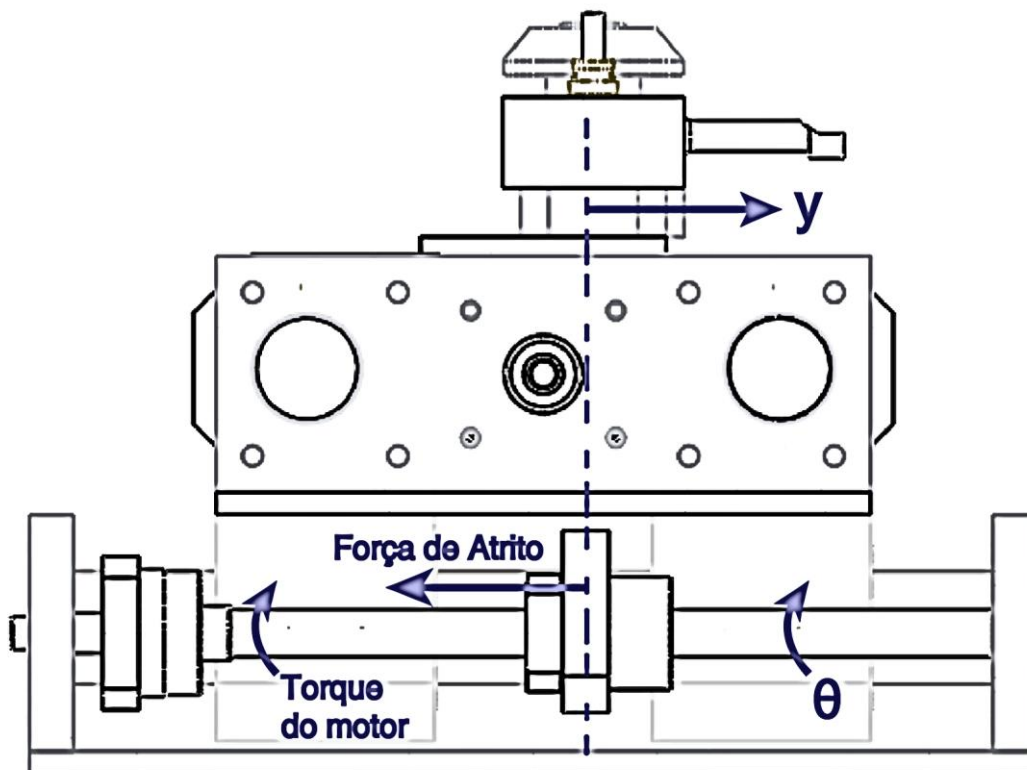


Figura 8.1 – Modelagem da guia inferior

Dado o diagrama de forças anterior, temos a seguinte equação de movimento:

$$M\ddot{y} + F_{atr} = \frac{2\pi}{p} (-T_m + B_\theta \dot{\theta} + J\ddot{\theta}) \quad (3)$$

Dada a relação:

$$\frac{p}{2\pi} \theta = y \quad (4)$$

Onde:

Tabela 8.1. – Variáveis da equação de movimento do sistema

<i>Variável</i>	<i>Descrição</i>
θ	Deslocamento angular do motor
y	Deslocamento linear da mesa
M	Massa da mesa
F_{atr}	Força de atrito do fuso e das guias cilíndricas com mancais aerostáticos
p	Passo do fuso
T_m	Torque do motor de passo
B_θ	Atrito do motor sem carga
J	Momento de Inércia da mesa em relação ao eixo central do fuso

Como utilizaremos um *encoder* linear a fim de medir a posição da ferramenta do local mais próximo aonde os erros de posicionamento são gerados, descreveremos todas as equações em função da variável que representa o deslocamento linear para a confecção da malha de controle. Portanto, substituindo a variável “ θ ” por “ y ”, seguindo a relação anteriormente exposta, temos a equação final do movimento:

$$M\ddot{y} + F_{atr} = \frac{2\pi}{p} T_m - \left(\frac{2\pi}{p}\right)^2 B_\theta \dot{y} - \left(\frac{2\pi}{p}\right)^2 J \ddot{y} \quad (5)$$

$$\left(\frac{2\pi}{p} J + M \frac{p}{2\pi}\right) \ddot{y} + B_\theta \frac{2\pi}{p} \dot{y} = T_m - \frac{p}{2\pi} F_{atr} \quad (6)$$

Analisando a teoria exposta em OGATA (2010), vemos que a equação acima se assemelha muito aos modelos de servomotores, se tratando principalmente de sistemas de segunda ordem dada as transformações de Laplace. Para que a malha de controle fique simples e o controle seja rápido e efetivo, trataremos o atrito como um distúrbio ao sistema e inicialmente o torque como a entrada. Portanto, aplicando a transformada de Laplace, temos que:

$$\left(\frac{2\pi}{p}J + M\frac{p}{2\pi}\right)Y_o(s)s^2 + B_\theta\frac{2\pi}{p}Y_o(s)s = T(s) - \frac{p}{2\pi}F_{atr}(s) \quad (7)$$

Zerando o distúrbio ($F_{atr}(s) = 0$), temos a seguinte função transferência que representa o comportamento do sistema a uma entrada do tipo torque:

$$\frac{Y(s)}{T(s)} = \frac{1}{\left(\frac{2\pi}{p}J + M\frac{p}{2\pi}\right)s^2 + B_\theta\frac{2\pi}{p}s} \quad (8)$$

Se zerarmos o torque ($T(s) = 0$), temos a função transferência para o distúrbio:

$$\frac{Y(s)}{F_{atr}(s)} = \frac{\frac{-p}{2\pi}}{\left(\frac{2\pi}{p}J + M\frac{p}{2\pi}\right)s^2 + B_\theta\frac{2\pi}{p}s} \quad (9)$$

Considerando inicialmente um controlador proporcional simples, podemos ter uma ideia de como ficaria a equação acima em malha fechada, considerando como entrada a posição da mesa:

$$\frac{Y_o(s)}{Y_i(s)} = \frac{K}{\left(\frac{2\pi}{p}J + M\frac{p}{2\pi}\right)s^2 + B_\theta\frac{2\pi}{p}s + K} \quad (10)$$

Uma última simplificação será feita, uma vez que o parâmetro passado é na verdade o ângulo do motor, ou seja, " θ ". Fazendo essa substituição na equação anterior, temos o sistema em malha fechada completo:

$$\frac{Y_o(s)}{\theta_i(s)} = \frac{K}{\left(\left(\frac{2\pi}{p}\right)^2 J + M\right)s^2 + B_\theta\left(\frac{2\pi}{p}\right)^2 s + K\frac{2\pi}{p}} \quad (11)$$

8.2 – Definição dos parâmetros

Levando em consideração à função transferência do sistema sem nenhum controlador, analisamos a resposta em deslocamento para uma entrada em torque. Com isso, pudemos definir os parâmetros do sistema sem depende de medições específicas em cada peça.

Com a curva experimental de uma entrada em degrau, pudemos ajustar o modelo teórico a fim de descobrir o valor de cada parâmetro apontado anteriormente.

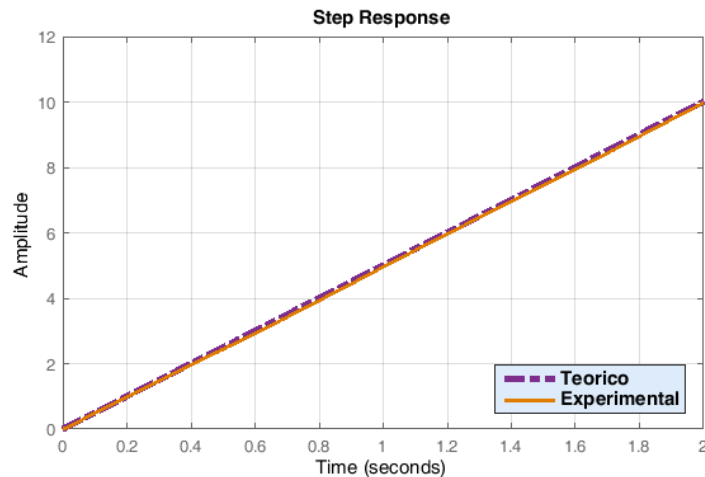


Figura 8.2 – Comparação entre modelo teórico e curva experimental para a guia inferior para uma entrada de torque do motor

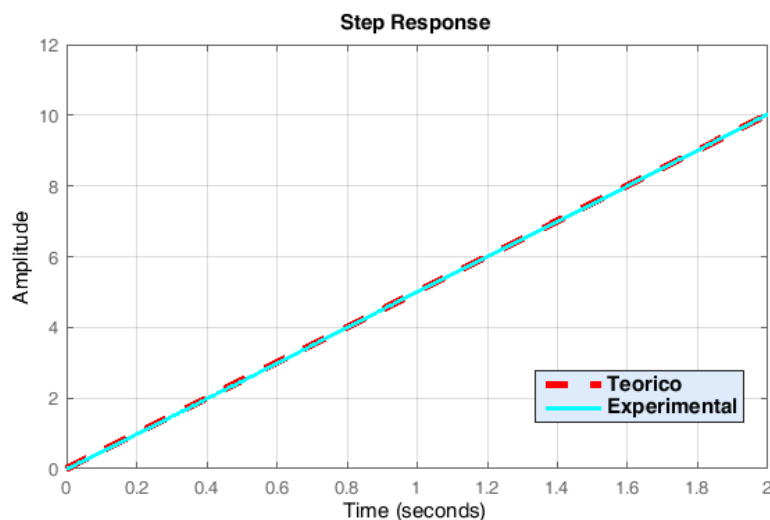


Figura 8.3 – Comparação entre modelo teórico e curva experimental para a guia superior para uma entrada de torque do motor

Tabela 8.2. – Valores dos parâmetros teóricos para curva experimental de resposta a um torque para a guia linear inferior

Variável	Descrição	Guia inferior	Guia superior
M	Massa da mesa	6,29 kg	0,33 kg
p	Passo do fuso	5 voltas/mm	5 voltas/mm
B_{θ}	Atrito do motor sem carga	0.0071	0.0071
J	Momento de Inércia da mesa em relação ao eixo central do fuso	$7.471 * (10^{-6})$ kg*m ²	$1.0536 * (10^{-6})$ kg*m ²

Com a metodologia utilizada, podemos avaliar até mesmo a influência de assimetrias no sistema montado. Dito isso, podemos partir para ajustar o controlador de acordo com as necessidades estabelecidas na próxima seção.

8.3 – Controlador

Dado o sistema exposto, podemos tratar o diagrama de blocos do mesmo, a fim de analisar a melhor proposta de controlador. Neste caso, precisamos nos atentar principalmente em garantir os seguintes requisitos, definidos com base nos requisitos previamente discutidos:

- $M_p \leq 0,1\%$
- $t_s \leq 8$ segundos

Inicialmente, o diagrama em malha fechada desenhado é colocado a seguir:

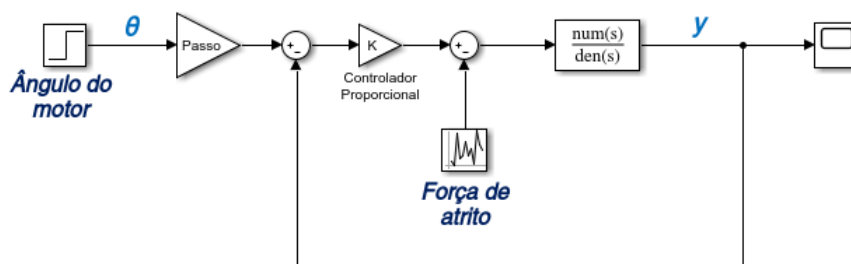


Figura 8.4 – Diagrama de blocos do sistema com controlador proporcional

Como dito anteriormente, trataremos o atrito como um distúrbio no sistema, e portanto, não levaremos em conta que há uma tendência para esse tipo de força.

Dito isso, para eliminarmos erros em regime permanente sem afetar o sistema obtido com sobressinal, escolhemos utilizar o controlador Proporcional Integral Derivativo.

O mesmo foi projetado para que, mesmo na presença de distúrbio de forma não conhecida, o sistema atinja a posição corretamente sem comprometer a qualidade do objeto usinado.

O Diagrama de blocos do mesmo é exposto a seguir:

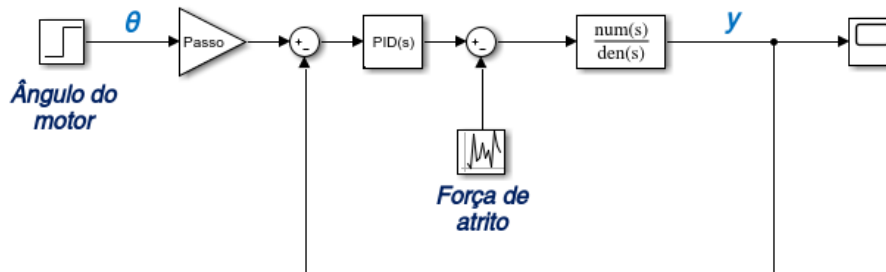


Figura 8.5 – Diagrama de blocos do sistema com controlador PID

Simulações iniciais possibilitaram estudar o comportamento do controlador escolhido variando os parâmetros P, I e D. Com isso, pudemos notar a grande diferença na estabilização do sistema visto as duas propostas de controlador sugeridas anteriormente.

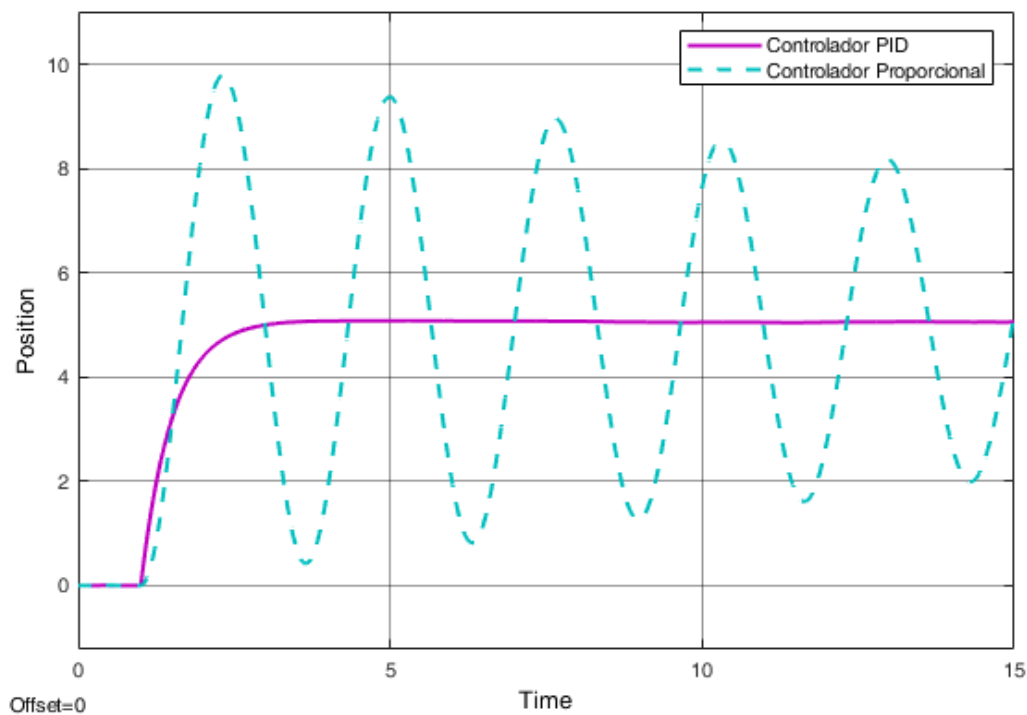


Figura 8.6 – Comparação entre as duas propostas de controlador sugeridas

Baseado no sistema teórico, calibramos o controlador implementado no sistema real com dados experimentais, visto que os parâmetros inicialmente definidos estavam

gerando um sobressinal de aproximadamente 2% para a guia inferior e 1,5% para a guia superior. Com isso, os parâmetros finais foram definidos como exposto a seguir:

Tabela 8.3. – Parâmetros dos controladores PID implementados

Variável	Guia inferior	Guia superior
K_p (P)	0,9	0,087
K_i (I)	0,011	0,001
K_d (D)	20	0,002

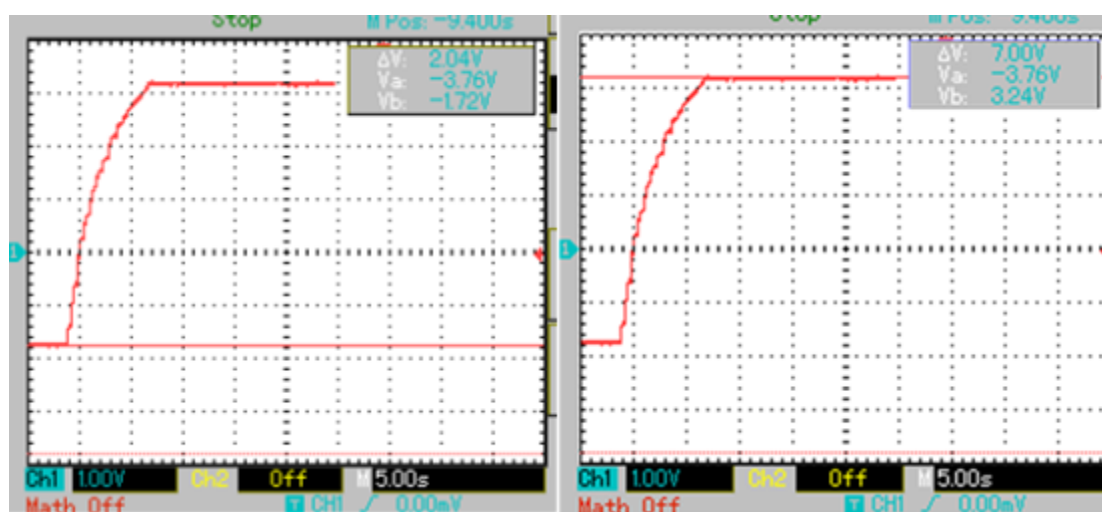


Figura 8.7 – Aquisição de dados considerando entrada impulso de posição na guia superior

(a) Resposta da guia inferior

(b) Resposta da guia superior

Dado que o *encoder* linear instalado acusava o posicionamento final dos eixos de movimentação corretamente, foi iniciado então o processo de validação do posicionamento, assim como a medição da linearidade do movimento

9.0 – VALIDAÇÃO

A construção e os testes foram realizados dentro do laboratório de Mecânica de Precisão no prédio da engenharia Mecânica, Mecatrônica e Naval ao longo do ano de 2018. A partir da construção da parte mecânica e da integração com a parte eletrônica do trabalho, realizaram-se vários testes de verificação para que houvesse certeza que os componentes projetados estavam funcionando corretamente e dentro dos parâmetros desejados que estão descritos no capítulo de requisitos de projeto.

Após essa configuração inicial, decidiu-se realizar a validação da máquina com base em dois testes, de forma similar ao que é proposto nas normas ISO 230-1:2012 e ISO 230-2:2014. O primeiro foi a verificação do posicionamento por meio de equipamento destinado a medição de deslocamento. E o segundo foi a verificação de linearidade da guia por meios da medição de um bloco padrão retangular de 100 mm realizada por um relógio comparador. Com esses testes deseja-se encontrar os desvios e com isso calcular os erros que são encontrados no posicionador cartesiano desenvolvido nesse trabalho.

9.1 – Metodologia:

A metodologia utilizada para validação do posicionador cartesiano proposto foi seguida de acordo a seguinte fluxograma:

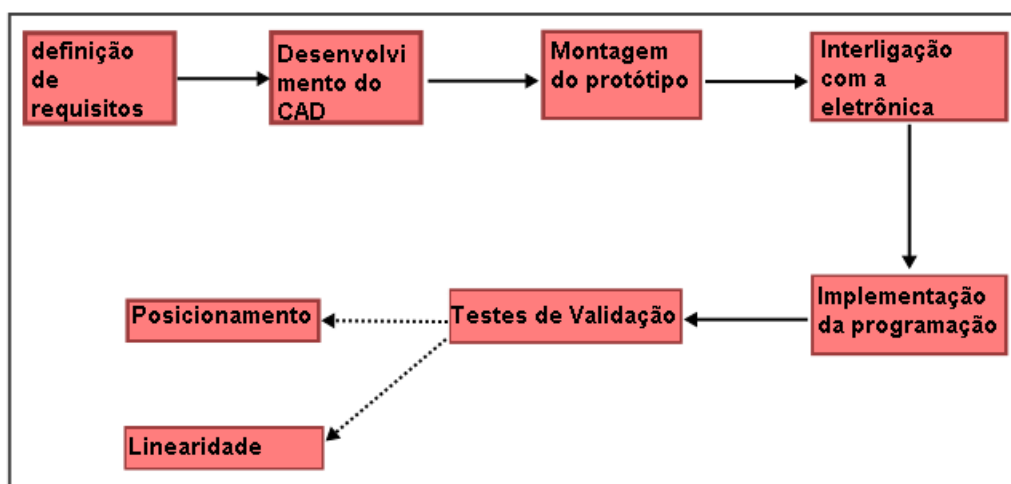


Figura 9.1 – Fluxograma de metodologia de desenvolvimento

9.2 – Teste de precisão de posicionamento:

Para validação nesse teste foi necessário utilizar o seguinte equipamento: Vibrômetro laser Polytec modelo OFV-323 e controlador modelo OFV-3020. Esse

equipamento é utilizado para medir vibrações e possui a capacidade de medir deslocamento e velocidade. Para isso foi posicionado o laser a uma distância de 1,5 m e o controlador foi configurado para fazer medições de 40 μ m por Volt com uma acurácia de 1,2%. O sinal de saída de distância lida pelo controlador do laser era lido por um osciloscópio digital MO-2100 da marca Minipa modelo MVB-DSO.

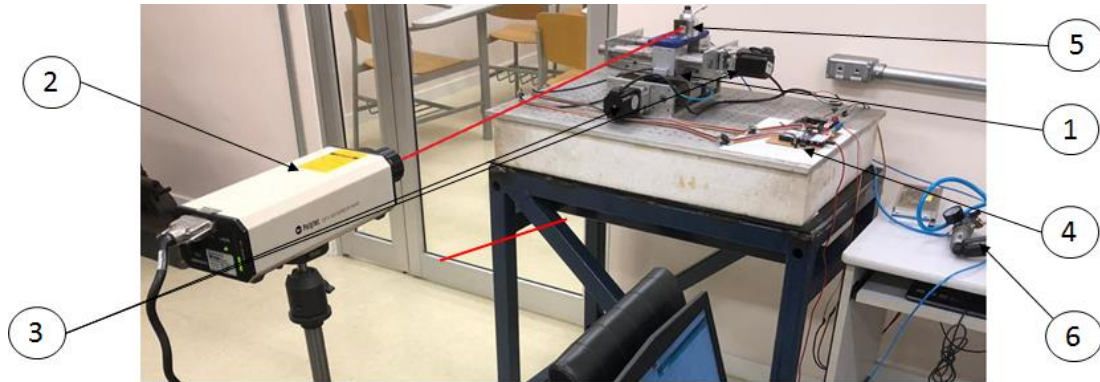


Figura 9.2 – Exemplo montagem do ensaio de posicionamento para a guia inferior, onde (1) Encoder linear da guia superior, (2) Vibrômetro laser Polytec modelo OFV-323, (3) Motor NEMA 23 da guia superior, (4) Placa manufaturada ligando o Arduino Due aos Drivers, (5) Superfície reflexiva e (6) Válvulas para controle da pressão de ar nas guias.

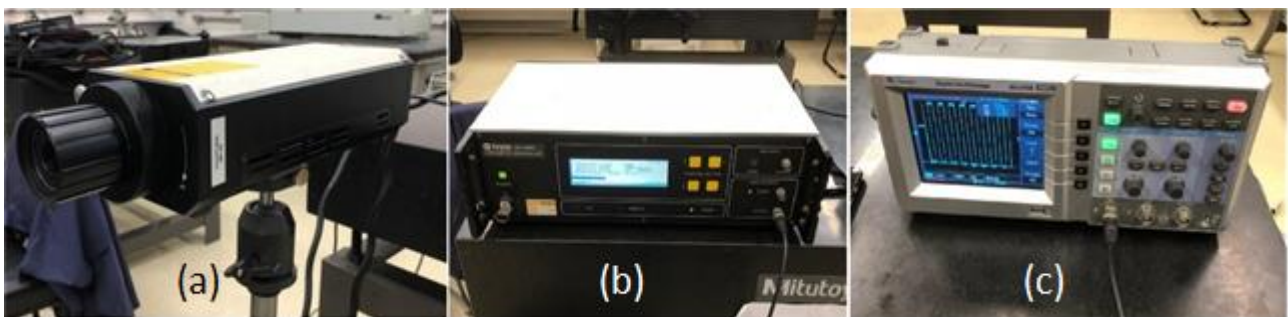


Figura 9.3 – Equipamentos utilizados para o Teste 1, onde (a) Vibrômetro laser Polytec modelo OFV-323, (b) Controlador modelo OFV-3020 e (c) Osciloscópio Digital MO-2100

Inicialmente foi passado comandos de deslocamento para a posição 400 μ m e de volta para a posição 0, sem que o controlador proposto estivesse ativo. Com isso pode-se obter a posição dada pelo sensor que estava acoplado à máquina e também pode-se obter a posição por meio dos dados de voltagem dada pelo sensor a laser fora do sistema.

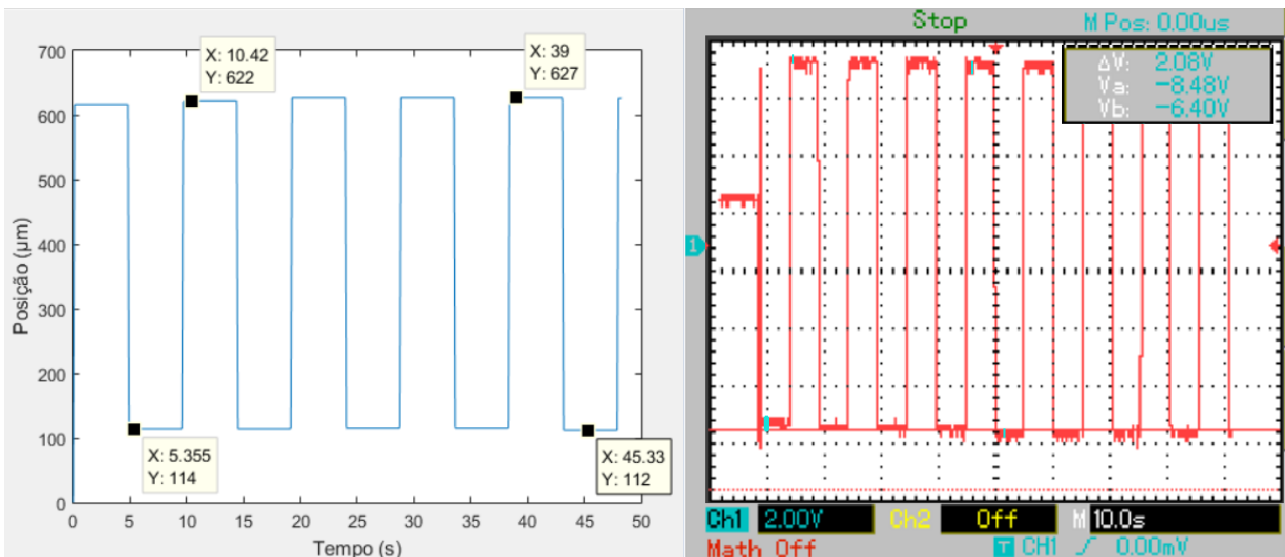


Figura 9.4 – Posição guia inferior, controlador desligado

De acordo com a figura 9.5 o sistema da guia inferior quando recebe o comando para deslocar um degrau de 400 µm e depois -400 µm para retornar a zero ciclicamente, acaba dando um salto para a posição 622 µm e depois retorna para a posição 114 µm com uma variação de 1,25% de faixa de erro tanto no limite superior quanto no limite inferior depois da estabilização, há também um erro estacionário de 25%. E como é possível perceber pela leitura do sensor laser há uma variação de 12,7 V entre a posição superior e a inferior, totalizando um deslocamento de 508 µm.

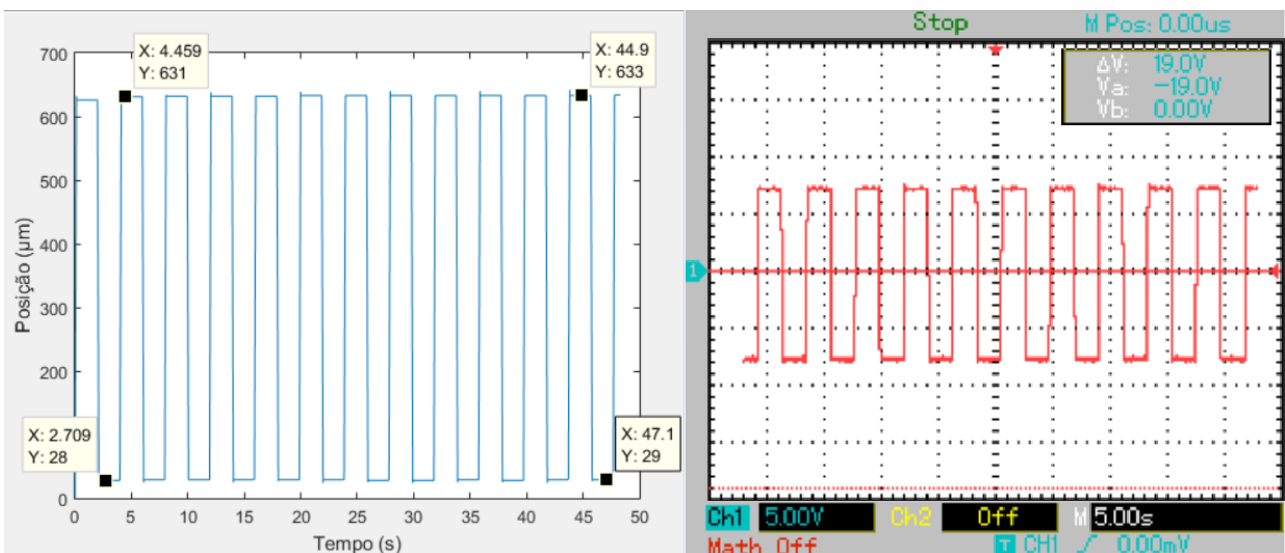


Figura 9.5 – Posição guia superior, controlador desligado

Já com relação a guia superior, verificou-se que para a mesma entrada inicial do ciclo de ida de volta de um deslocamento de 400 µm, o comportamento da guia se deu

com um salto para a posição de 631 μm e com o retorno a posição de 28 μm , com isso pode-se verificar que há uma faixa de erro de 0,5% tanto para o limite superior quanto para o inferior, fora o erro estacionário de 50%. Conforme a leitura do laser, há uma variação de voltagem de 15 V entre as posições superior e inferior o que significa uma diferença de deslocamento de 600 μm .

Em seguida foram feitas as medições para as guias com o controlador ligado. Desejava-se que o sistema tivesse um deslocamento máximo de 400 μm , para isso foi programado no controlador que houvesse um deslocamento para a posição 200 μm e que depois o sistema se deslocasse para -200 μm e assim sucessivamente. Foi feito a aquisição dos dados do laser e do sensor magnético do sistema, ao mesmo tempo o sensor magnético realimentava o sistema de forma a fechar a malha de controle.

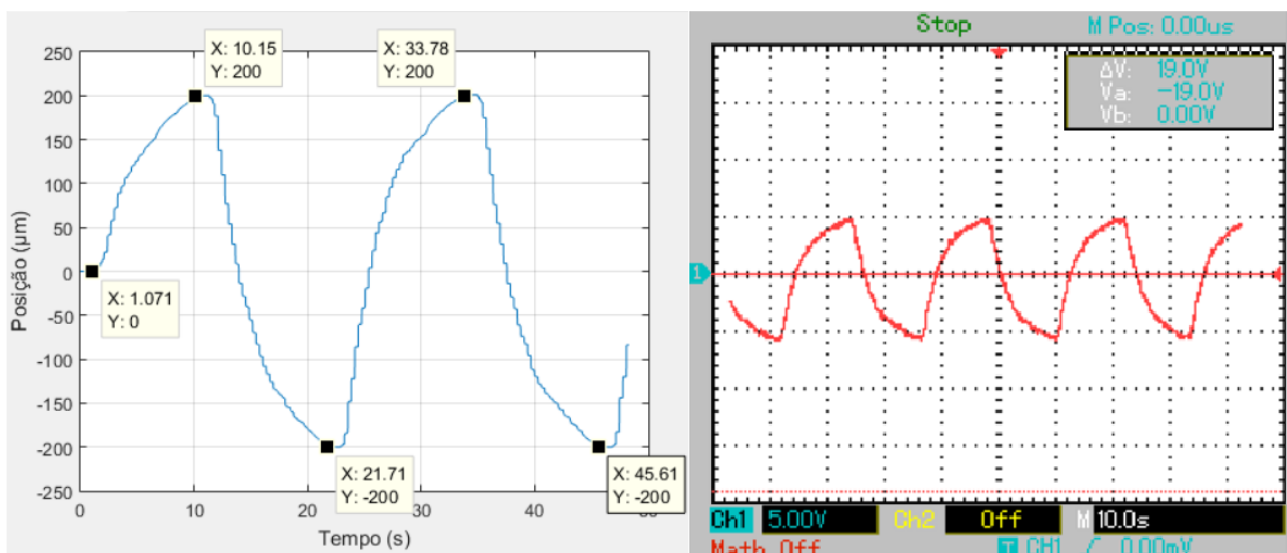


Figura 9.6 – Posição guia inferior, controlador ligado

Conforme é possível observar na figura 9.7 de acordo com o sensor o sistema sai da posição inicial zero e se desloca até a posição 200 μm após uma espera de 2 segundos quando a guia já se encontra na posição, esta passa então a se deslocar para a nova posição de -200 μm . Esses dados são confirmados quando se estuda a saída do osciloscópio, onde é possível verificar a diferença de voltagem de -5.4 V na posição inferior e de 4,6 V na posição superior, o que equivale a 10 V ou 400 μm .

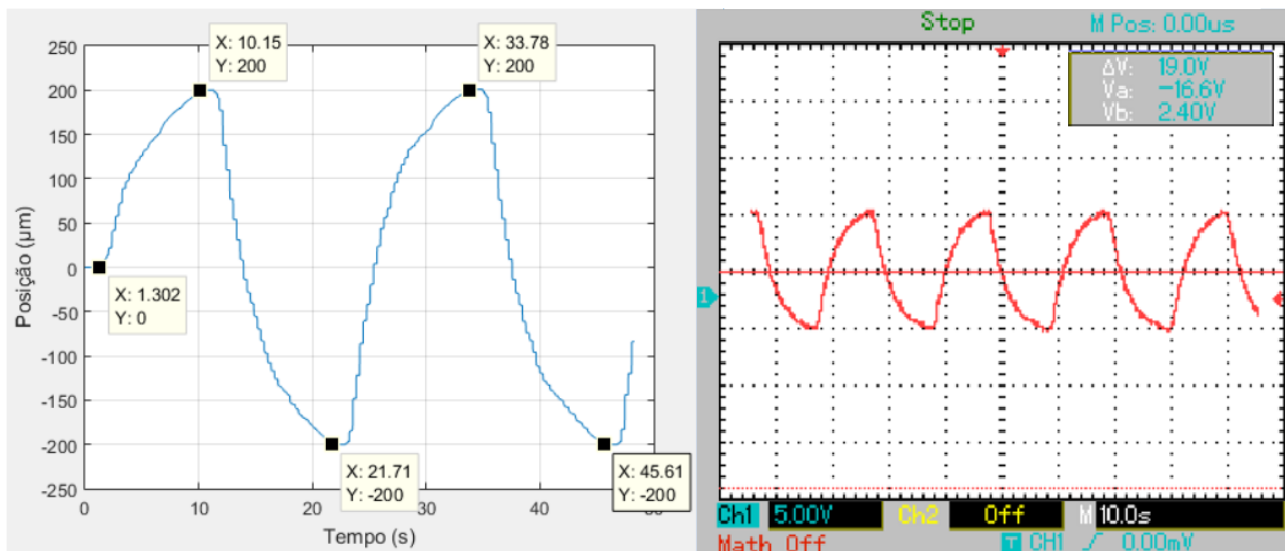


Figura 9.7 – Posição guia superior, controlador ligado

Também foi possível encontrar um resultado similar com a guia superior, conforme é possível verificar na Figura 9.8. Nela a guia também foi para a posição 200 µm, estabiliza por 2 segundos e passa a buscar a nova posição de -200 µm. Dados comprovados pela imagem do osciloscópio em que é possível ver que o sinal tem seu limite superior em 5 V e inferior em -5 V, assim o valor do deslocamento é de 400 µm, conforme esperava-se obter de acordo com o que foi especificado para o projeto.

Após obter os dados supracitados, houve-se a necessidade de testar a repetibilidade das aquisições. Os dados obtidos possibilitariam o cálculo da média do desvio de posicionamento unidirecional da guia, como sugerido na ISO 230-2:2014. Para isso o novo método consistia de uma espera de 5 segundos, e em seguida era enviado um sinal de degrau de módulo 200 µm. Após chegar na posição final, o sistema continuava sendo observado para o caso de haver alguma mudança de estado. Foi configurado no osciloscópio a maior resolução disponível para que o menor erro possível fosse observado. Além disso, eram dadas entrada de degrau positivo e negativo para que fosse possível verificar se havia algum movimento de *backlash* devido a possível existência de folga nos fusos de esferas recirculantes.

Iniciando-se pela guia inferior obteve-se os seguintes resultados:

Tabela 9.1 – Repetibilidade do controlador da guia inferior

Teste	Teste 1	Teste 2	Teste 3	Teste 4	Teste 5	Teste 6
Osciloscópio (ΔV)	4,98	4,96	5,04	5,00	4,98	5,02
Conversão (μm)	199,2	198,4	200,8	200	199,2	201,6
Sentido	+	-	-	+	-	+

A seguir está a imagem obtida do Teste 6 da guia inferior:

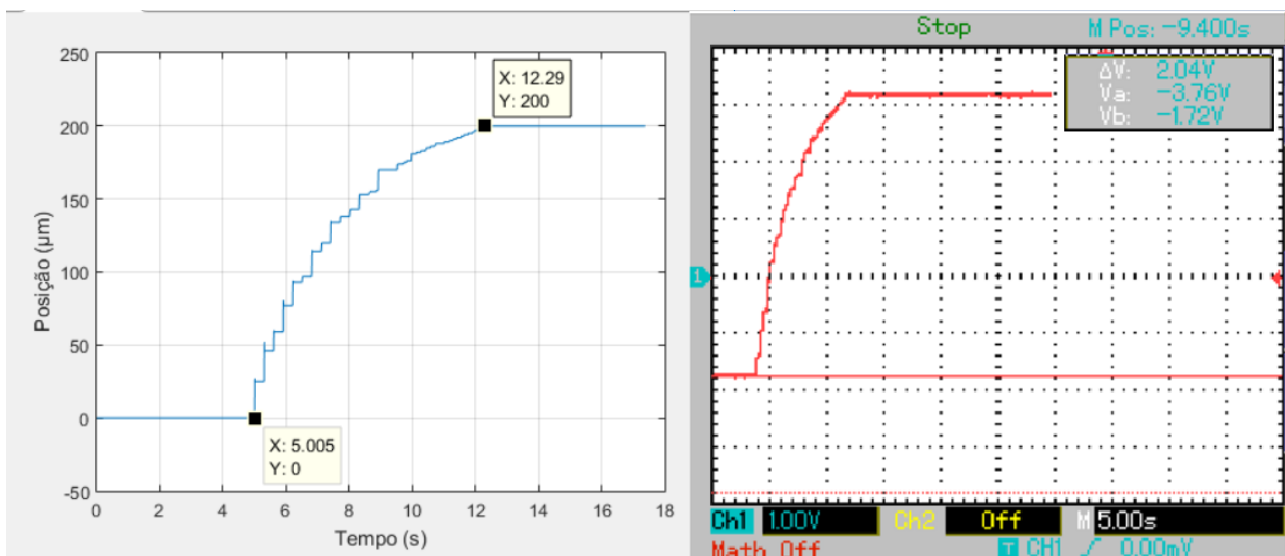


Figura 9.8 – Teste de repetibilidade da guia inferior

Devido a configuração que foi possível alcançar a melhor resolução no osciloscópio, foi possível verificar que, no teste de repetibilidade, embora o sensor da guia estivesse informando uma posição final de módulo 200 μm , havia um erro aleatório de no máximo 1,6 μm em módulo tanto para mais quanto para menos.

Já para a guia superior foi utilizado o mesmo método de análise e foram obtidos os seguintes resultados:

Tabela 9.2 – Repetibilidade do controlador da guia superior

Teste	Teste 1	Teste 2	Teste 3	Teste 4	Teste 5	Teste 6
Osciloscópio (ΔV)	4,98	4,98	5,02	5,02	4,98	4,98
Conversão (μm)	199,2	199,2	200,8	200,8	199,2	199,2
Sentido	+	-	+	-	+	-

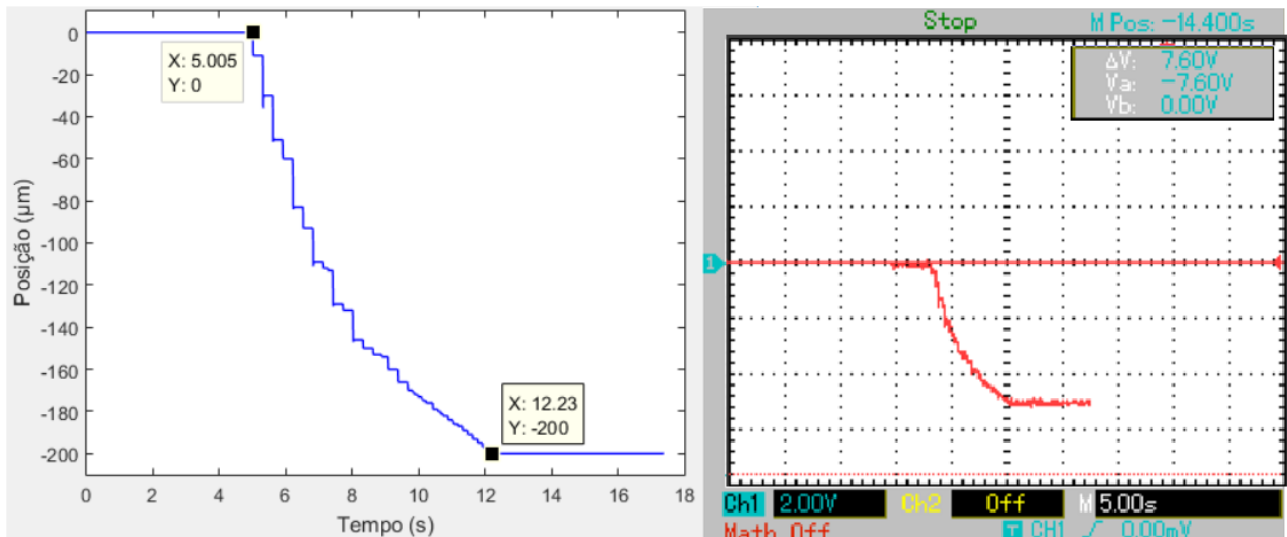


Figura 9.9 – Teste de repetibilidade da guia superior

No teste da guia superior também foi possível verificar que, embora o sensor da guia estivesse informando uma posição final de módulo 200 μm , havia um erro aleatório de no máximo 0,8 μm em módulo tanto para mais quanto para menos.

9.2 – Teste de linearidade das guias:

Para validação nesse teste foi necessário utilizar um relógio comparador modelo Padrão com graduação de 0,001 mm da marca Mitutoyo, que foi montado em um castelo porta ferramentas para torno. Para verificar a linearidade foi utilizado também um bloco padrão de 100 mm que foi montado em um suporte para que este pudesse ser alinhado e mantido fixo ao longo da guia.

O teste de linearidade foi feito de acordo com o sugerido na ISO 230-2:2012, embora o cálculo do erro foi avaliado em relação ao posicionamento absoluto da guia, não sendo subtraído deste valor a posição passada ao controlador.

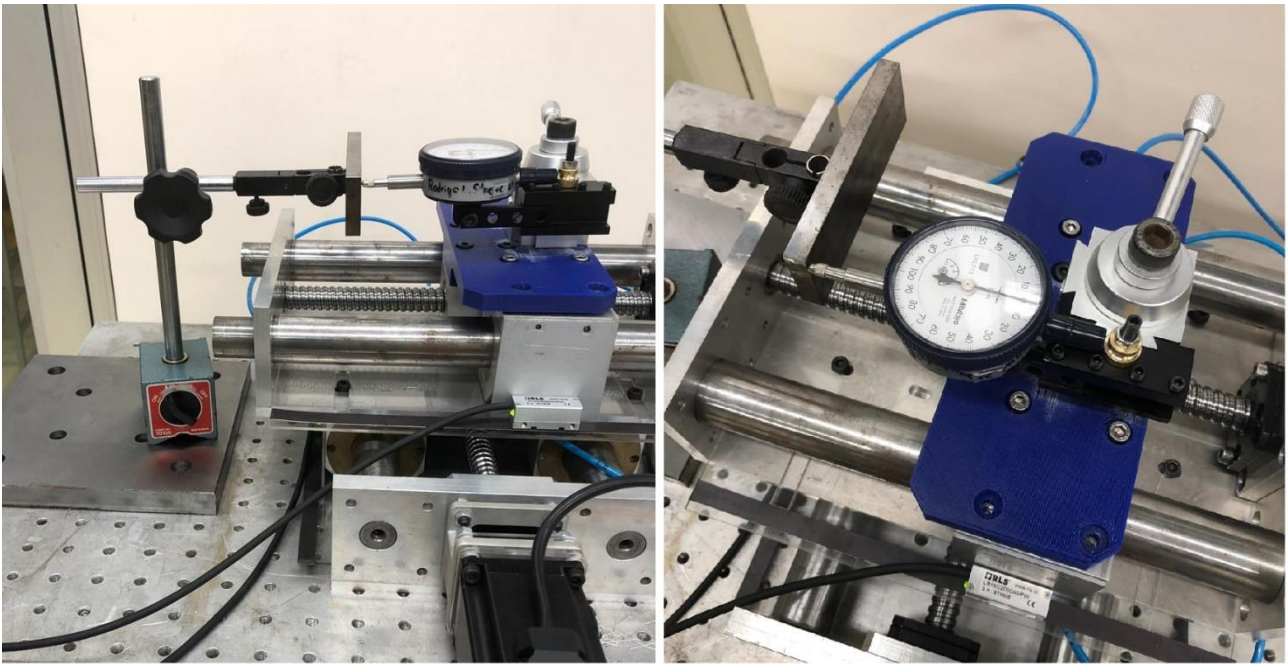


Figura 9.10 – Teste de linearidade de movimento

O relógio comparador e o bloco padrão foram montados de forma que houvesse alinhamento entre a lateral do bloco em relação a guia que estava sendo verificada. Após isso, programava-se um movimento controlado para afastar o relógio do bloco e deslocar-se a ferramenta para o outro lado do bloco. A partir daí, aproximava-se a ferramenta do bloco padrão de forma controlada até que o relógio marcasse 1,000 mm. Em seguida deslocava-se o relógio ao longo do bloco padrão de forma que o relógio estivesse sempre em contato com sua face e a cada 10 mm foi se medindo e catalogando a marcação que estava sendo mostrada. Quando o relógio chegava ao seu zero inicial calculava-se o erro de linearidade que o sistema possui.

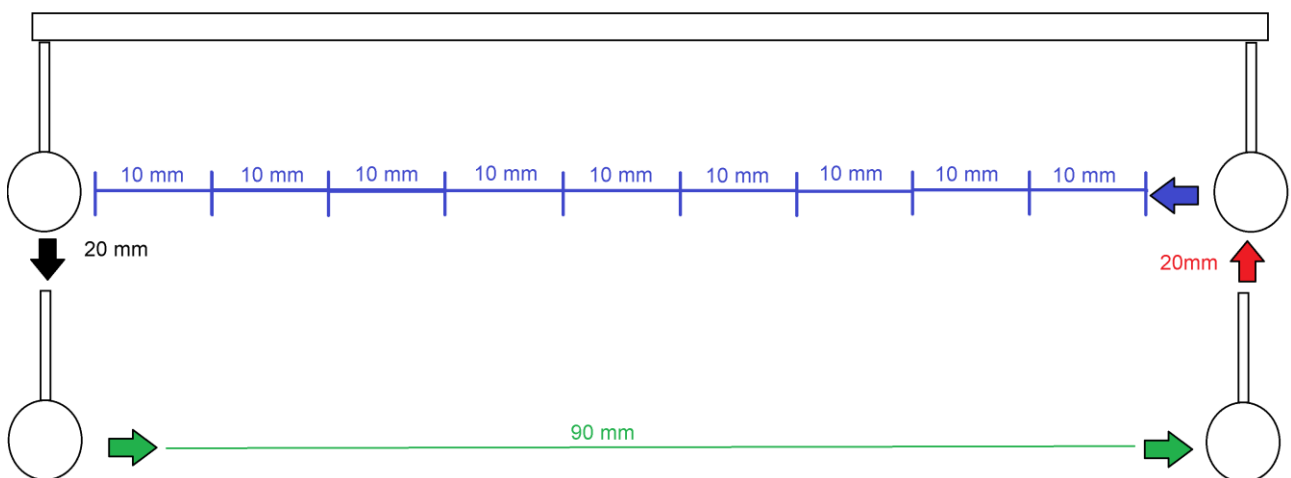


Figura 9.11 – Modelo do método utilizado para verificar a linearidade das guias

Para a guia inferior após a fixação da lateral do suporte no ponto zero, levou-se a ferramenta para a posição 90 mm, quando o relógio comparador estava na posição final, ajustou-se essa lateral para que o relógio marcasse 1000 μm . Após isso, a ferramenta foi programada para caminhar no eixo X e os dados X e Y foram obtidos.

Tabela 9.3 – Linearidade da guia inferior

X (mm)	Y (μm)	Fit (μm)	FitErro (μm)
0	107	110	-3
10	216	209	7
20	302	308	-6
30	404	407	-3
40	510	506	4
50	613	605	8
60	703	703	0
70	794	802	-8
80	903	901	2
90	1000	1000	0

Com os dados coletados, utilizou-se o software *MATLAB® R2015a* para obter a reta característica desses dados. Para isso foi necessário utilizar a função *polyfit* que calcula a reta que pode ser traçada que tenha os menores erros entre os pontos. Após isso foi preciso usar a função *polyval*, que calcula o ponto Fit da reta característica. Por fim foi calculado o erro final que é dado pela subtração do ponto Fit por Y.

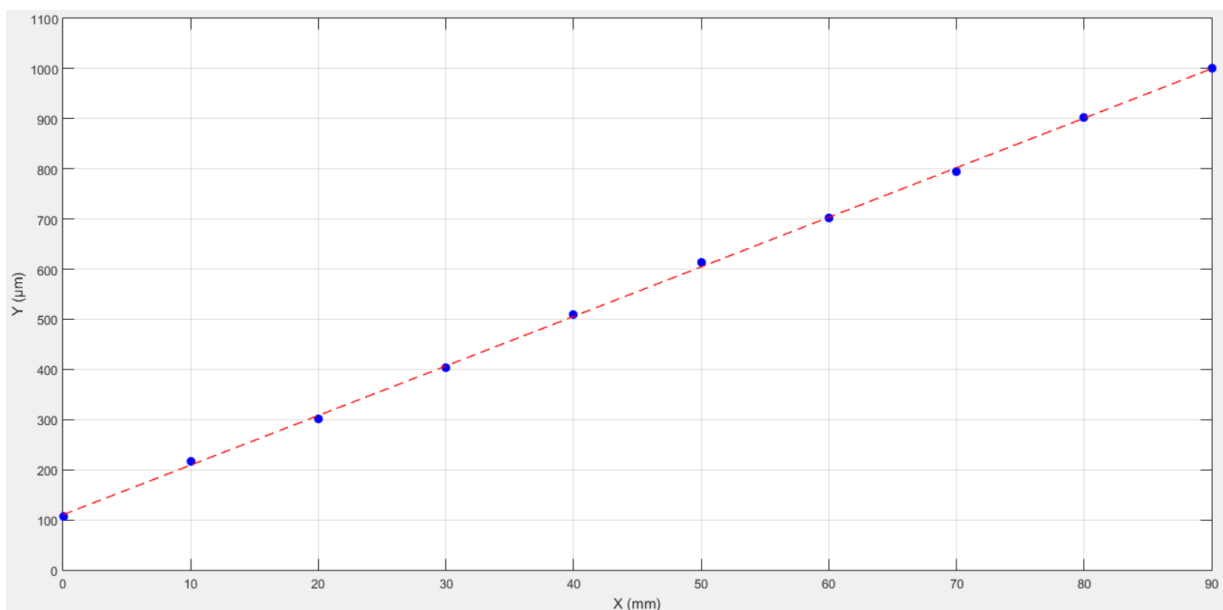


Figura 9.12 – Reta característica guia inferior

Para a guia superior utilizou-se o mesmo método para o cálculo do deslocamento no eixo perpendicular quando a ferramenta se desloca por ela. Assim, com deslocamentos no eixo Y foram coletados as marcações do relógio comparador, obtendo-se então os pontos X abaixo.

Tabela 9.4 – Linearidade da guia superior

Y (mm)	X (μm)	Fit (μm)	FitErro (μm)
0	57	59	-2
10	164	164	0
20	263	269	-6
30	379	374	5
40	484	479	5
50	586	583	3
60	685	688	-3
70	788	793	-5
80	904	898	6
90	1000	1003	-3

Utilizando o mesmo programa e realizando os mesmos cálculos feitos para a guia inferior, que foram realizados para a guia inferior, foi possível obter a reta característica para a guia superior.

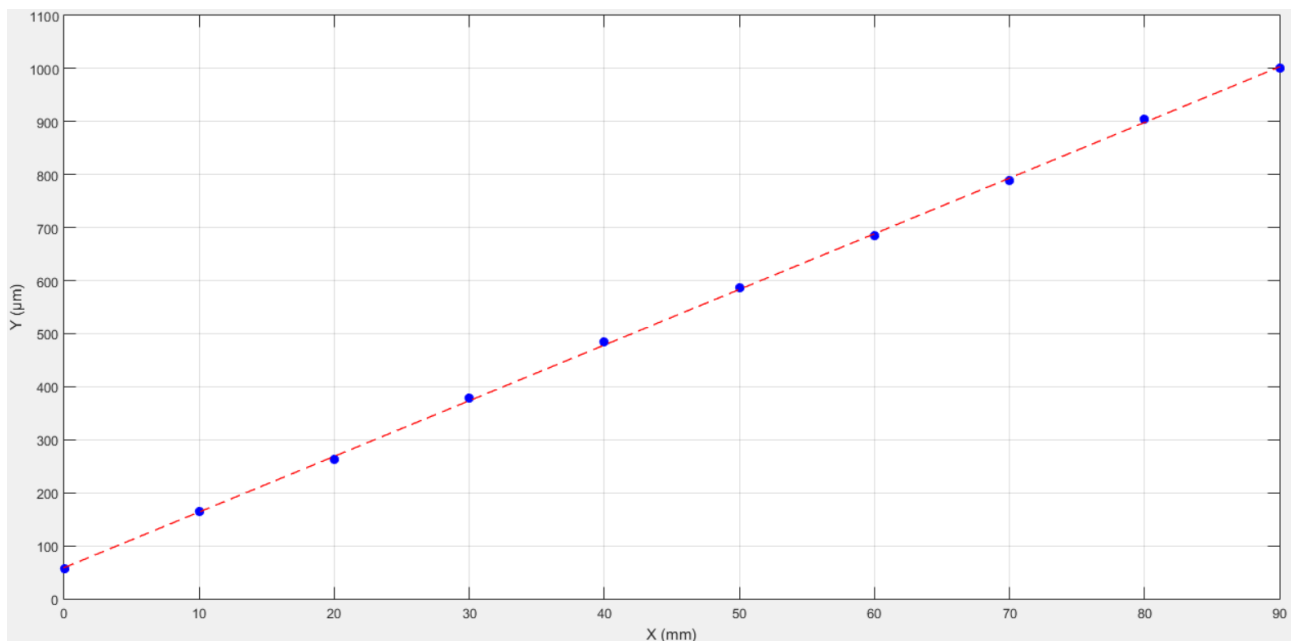


Figura 9.13 – Reta característica guia superior

Considerando os desvios obtidos em torno das retas, vemos uma variação máxima de 8 μm para a guia inferior e 6 μm para a guia superior durante os testes realizados. Assim, fica evidente que o erro obtido durante a movimentação do mecanismo ainda está da tolerância de movimentação proposta anteriormente como requisito de projeto.

10.0 – CONCLUSÃO

O desenvolvimento desse posicionador cartesiano de alta precisão controlado por computador é uma grande aplicação de todas as disciplinas administradas no curso de Mecatrônica. Os sistemas mecatrônicos são compostos principalmente por partes mecânicas e eletrônicas, sendo tudo isso integrado por uma parte computacional que controla o sistema.

Nesse trabalho teve-se o desenvolvimento do projeto mecânico, junto com as análises estáticas e dinâmicas da estrutura dessa máquina. Também foi necessário fazer o estudo das guias aerostática, dos fusos de esferas recirculantes que foram utilizados. Além disso foi necessário fazer o estudo dos motores de forma a escolher um que pudesse se adequar ao projeto mantendo os custos controlados.

Já na parte eletrônica foi necessário o estudo e obtenção dos drives controladores do motor junto com o desenvolvimento de uma placa que pudesse interligar os sensores das réguas magnéticas junto a placa de controle, para que assim pudesse ser desenvolvido o controlador em malha fechada de todos dos dois sistemas, inferior e superior.

Além disso foi necessário integrar todos os sistemas por meio do desenvolvimento de uma programação que pudesse receber os dados adquiridos no sensor de posição, fazer os cálculos de controle proporcional integral derivativo e que também enviasse os sinais para o controlador dos motores de forma que esse pudesse atuar corretamente no sistema.

Após esse longo processo, foi necessário fazer a validação da máquina. Para isso foram realizados dois experimentos: o primeiro deles utilizando-se um sensor de posição a laser, onde calculou-se a posição que a ferramenta estava quando esta se deslocava ao longo dos eixos. Para a guia inferior, verificou-se que esta, quando acionada em circuito de controle aberto, possuía um erro de 25% no posicionamento variando numa faixa de 1,25%, o tempo de subida que em média foi de 0,2 segundos. Já para a guia superior, encontrou-se um erro estacionário de 50% com uma faixa de variação de 0,5%, com tempo de subida médio de 0,2 segundos.

Decorrido o desenvolvimento da simulação do controlador de cada uma das guias, foi possível desenvolver um controle PID que melhorou sobremaneira os erros encontrados nas malhas abertas. Para a guia inferior o erro calculado por meio do sensor de posição a laser foi de 0,067% para o erro estacionário variando em 0,8%, com tempo de subida de 5 segundos. E para a guia superior, o novo erro estacionário passou a ser de 0,134% com uma variação de 0,4%, também com tempo de subida de 5 segundos. Os

valores obtidos para os erros foram considerados satisfatórios por estarem dentro das especificações propostas no início deste projeto. Já quando calcula-se o tempo de subida houve um aumento considerável, o que já era esperado e compreende-se já que para se conseguir tamanha precisão é necessário abrir mão da velocidade de posicionamento da ferramenta.

Já o segundo experimento foi desenvolvido para calcular o erro de linearidade que cada uma das guias possui de forma a verificar como a movimentação de cada eixo afeta na precisão do seu par. Para isso utilizou-se um bloco padrão e um relógio comparador. Após percorrer o bloco com o relógio, calculou-se que a guia inferior possui um erro máximo de 0,008 mm e a guia superior 0,006 mm.

Com isso, é possível afirmar que o trabalho alcançou seus objetivos e conseguiu entregar um posicionador cartesiano de precisão de posicionamento de 10 μm . Além disso, foi dado o passo inicial para o desenvolvimento de um torno completo, controlado por computador e com alta precisão de usinagem.

11.0 – SUGESTÕES PARA TRABALHOS FUTUROS

Tendo em vista que o projeto desenvolvido pode ser integrado a vários outros sistemas, para a construção de um torno de alta precisão ainda é necessário desenvolver o projeto de um spindle aerostático, sendo este integrado a placa de processamento do posicionador da ferramenta. É importante também que esses sistemas possam se comunicar e estar integrados com softwares de controle de máquinas CNC que já estão no mercado, como por exemplo LinuxCNC.

Para a validação de toda a máquina será necessário garantir o alinhamento do spindle com os eixos do posicionador da ferramenta, além de verificar a movimentação das guias quanto a movimentos em conjunto em trajetórias angulares, circulares e elípticas.

12.0 – REFERÊNCIAS BIBLIOGRÁFICAS

- BRINKSMEIER, E. et al. Ultra-precision grinding. *CIRP Annals - Manufacturing Technology* 59, 652–671, 2010
- TANIGUSHI N. Current Status in, and Future Trends of, Ultraprecision Machining and Ultrafine Materials Processing, *Annals of the CIRP Vol. 32/2/1983*.
- LUO, X. et al. Design of ultraprecision machine tools with applications to manufacture of miniature and micro components. *Journal of Materials Processing Technology* 167, 515–528, 2005.
- WYBRANSKI, B. Micro production technologies. *MstNews* 2, 43–45, 2004.
- MCXEOWN, P.A. The design and development of a large ultra precision CNC diamond turning machine SME technica paper, MR-82-931, 1982
- MORI, Y.; YAMAUCHI, K.; ENDO, K. *Elastic emission machining*, Butterworth & Co (Publishers) Ltd, 1987
- WILSON, S. R.; REICHER, D. W.; MCNEIL, J. R. 1988 Surface figuring using neutral ion beams. *Proc. SPIE* 966, 74–81.
- GÄBLER, J. *CompeDIA Micro Abrasive Pencils*. Jahresbericht Fraunhofer, IST 36–37, 2004
- HAEFELI, R. Anwendungsmöglichkeiten von Schleifstiften. *Industrial Diamond Review* 2(97):164–168. 1997
- AURICH, J. C.; ENGMANN, J.; SCHUELER, G., HABERLAND R.. *Manufacture of Complex Micro Grinding Tool for Structures in Brittle Materials*. *Annals of the CIRP*. 2009
- SCHINKER, M. G.; DOLL, W. Turning of optical glasses at room temperature. *Proc. SPIE* 802, 70–80, 1987.
- PUTTICK, K. E. et al. Single point diamond machining of glasses. *Proc. R. Soc. Lond. A* 426, 19–30, 1989
- BLAKE, P.N.; SCATTERGOOD, R.O. Ductile-regime turning of germanium and silicon. *J. Am. Ceram. Soc.* 73, 949–957, 1990.
- BIFANO, T. G.; DOW, T. A.; SCATTERGOOD, R.O.. Ductile-regime grinding: a new technology for machining brittle materials. *J. Eng. Ind. Trans. ASME* 113, 184–189. 1991
- WILLS-MOREN, W. J. et al. Some aspects of the design and development of a large high precision CNC diamond turning machine. *CIRP Ann. Manuf. Technol.* 31, 409–414, 1982
- MCKEOWN, P.A.; READ, R.F. & WILLS-MOREN, W.J.. Experiences in the precision machining of grazing incidence X-ray mirror substrates. *Proc. SPIE* 571, 42–50, 1986.

GARWIN, L.; LINCOLN, T.. A century of nature: twenty-one discoveries that changed science and the world. Chicago, IL: University of Chicago Press., 2003.

SHORE, P.; MORANTZ, P.. Ultra-precision: enabling our future. Precision Engineering Institute, Cranfield University, Cranfield, Bedfordshire, MK43 0AL, UK. 2012.

OGATA, K. Engenharia de Controle Moderno. 2010, Pearson Prentice Hall, São Paulo

BRITES, F. G.; ALMEIDA, V. P. S. Motor de Passo PET-Tele, Universidade Federal Fluminense, Centro Tecnológico, Departamento de Engenharia Telecomunicações, 2008.

STOETERAU, R. L. Introdução ao Projeto de Máquina-Ferramentas Modernas, Universidade Federal de Santa Catarina, Centro Tecnológico, Departamento de Engenharia Mecânica, 2004.

FERRARESI, D., Fundamentos da Usinagem dos Metais. Ed. Edgard Bluncher Ltda, 1977.

MEKID, S. Introduction to Precision Machining Design and Error Assessment. 2011, CRC Press, New York.

SAM4E Series Datasheet: http://www.atmel.com/Images/Atmel-11157-32-bit-Cortex-M4-Microcontroller-SAM4E16-SAM4E8_Datasheet.pdf

Atmel Corporation. / Rev.: Atmel-42706A-Quadrature-Decoder-(QDEC)-for-SAM3/4-Devices_AT11483_Application Note-04/2016

LM13D04_02Datasheet.[http://resources.renishaw.com/details/Data+sheet3A+LM13+linear+magnetic+encoder+system+\(DPI\)\(22880\)](http://resources.renishaw.com/details/Data+sheet3A+LM13+linear+magnetic+encoder+system+(DPI)(22880))

RAMPS-FD Datasheet: <https://www.geeetech.com/wiki/index.php/Ramps-FD>

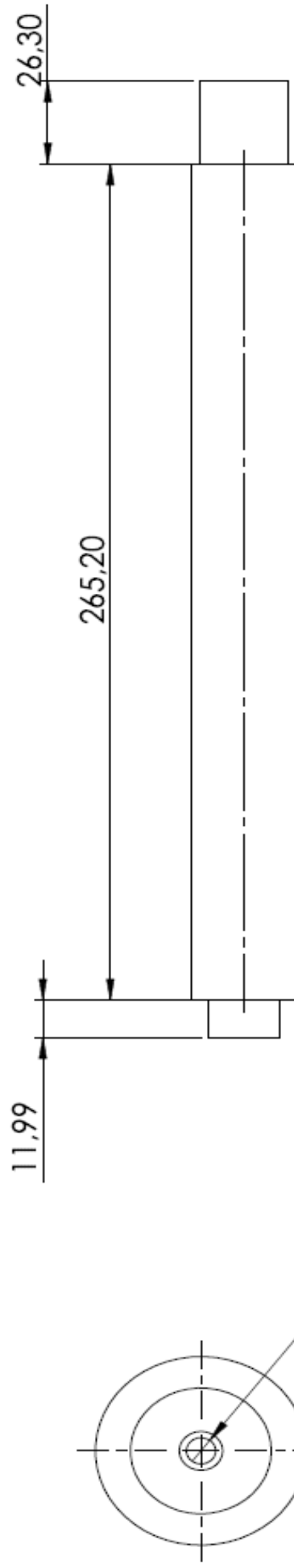
BEAUREGARD, B. Arduino PID Library: <https://playground.arduino.cc/Code/PIDLibrary>

BADEA, L. Generic Stepper Motor Driver Driver: https://github.com/laurb9/Stepper_Driver/blob/master/src/BasicStepperDriver.cpp

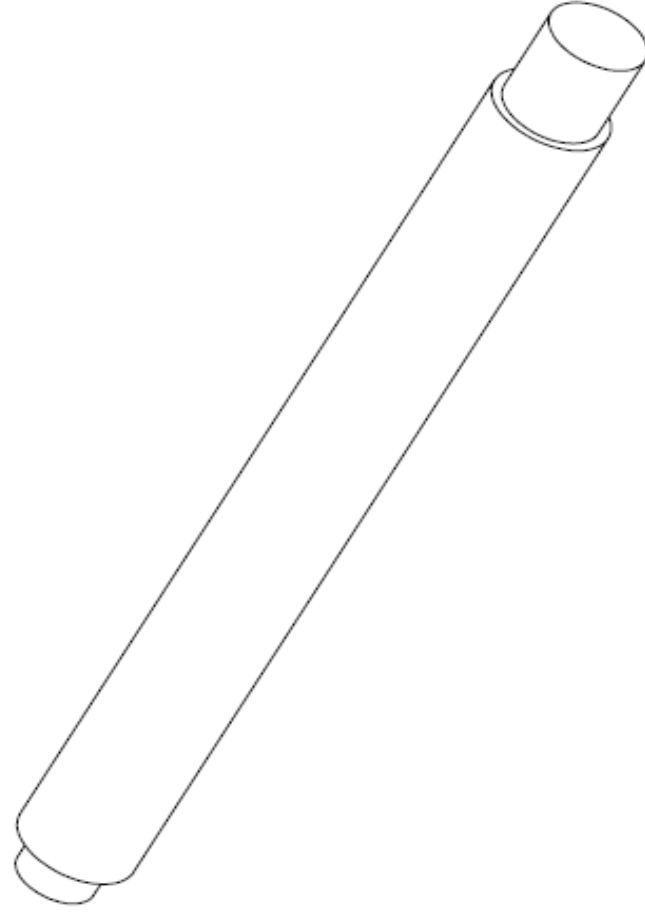
STOFFREGEN, P. Encoder Library: https://www.pjrc.com/teensy/td_libs_Encoder.html

GOMES, I. S.; KLAUS, P.. Implementation of Timers defined on DueTimer: <https://github.com/ivanseidel/DueTimer>

APÊNDICE A – DESENHOS DE FABRICAÇÃO



Medidas a serem modificadas	
Medida Atual	Valor modificado
11,99 mm	9,40 mm
4,18 mm	M6x1.00 - 7.4mm



Tolerância geral de 0,1mm

Escola Politécnica da Universidade de São Paulo
Engenharia Mecatronica

Subsistema: Projeto: TCC 2018

Folha Título

1/1

Guia para Mancais da Mesa Superior

Escala
1:2

Material
AÇO

Quantidade: 2

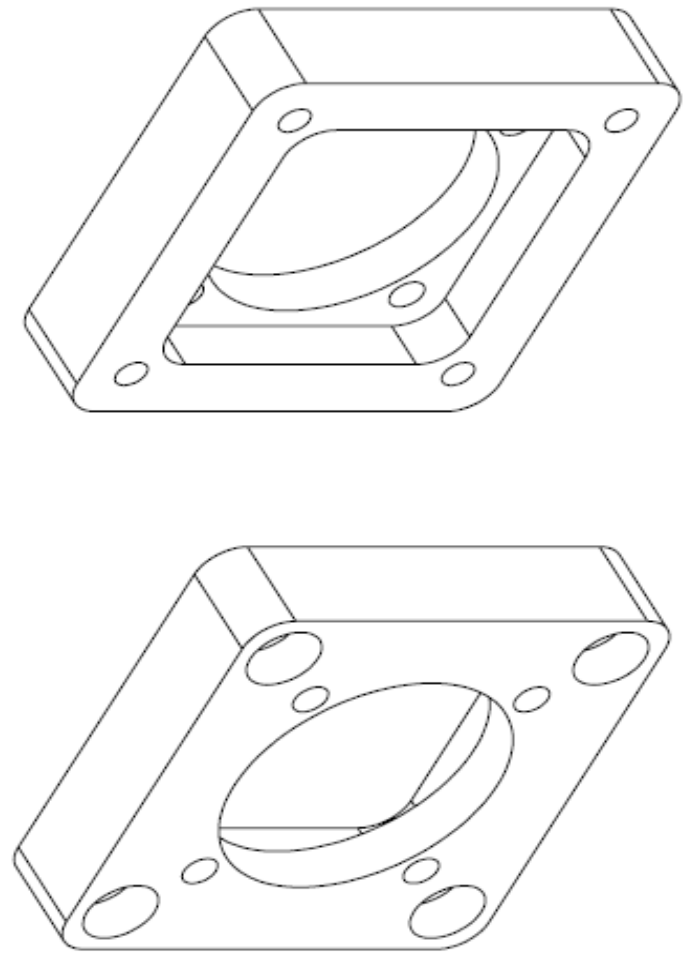
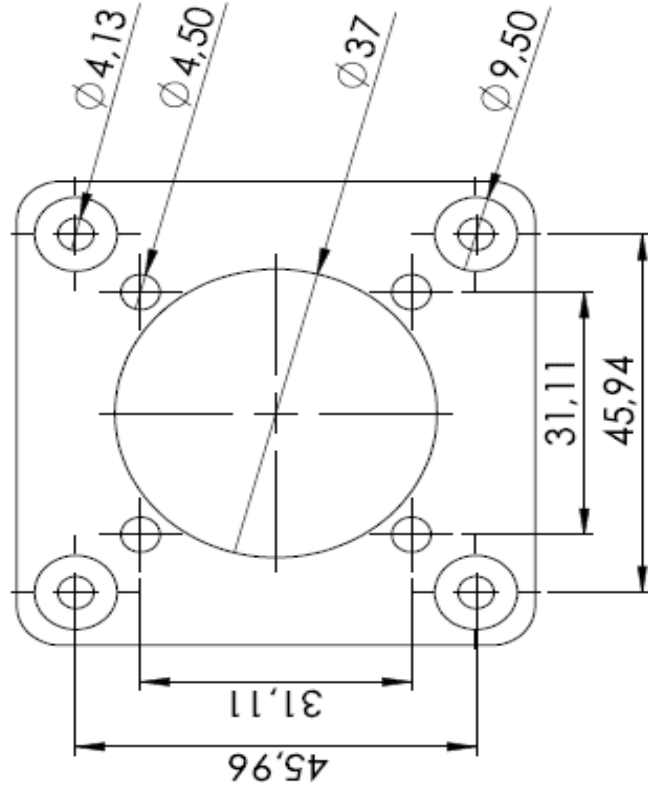
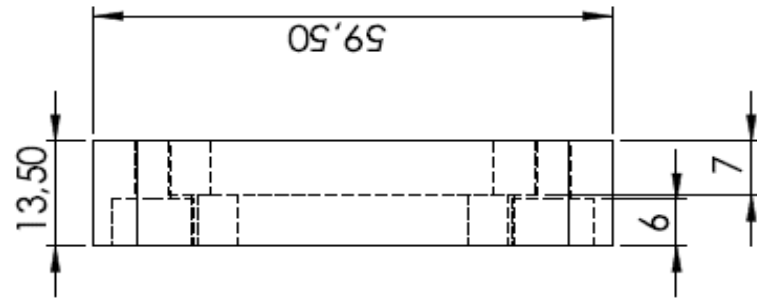
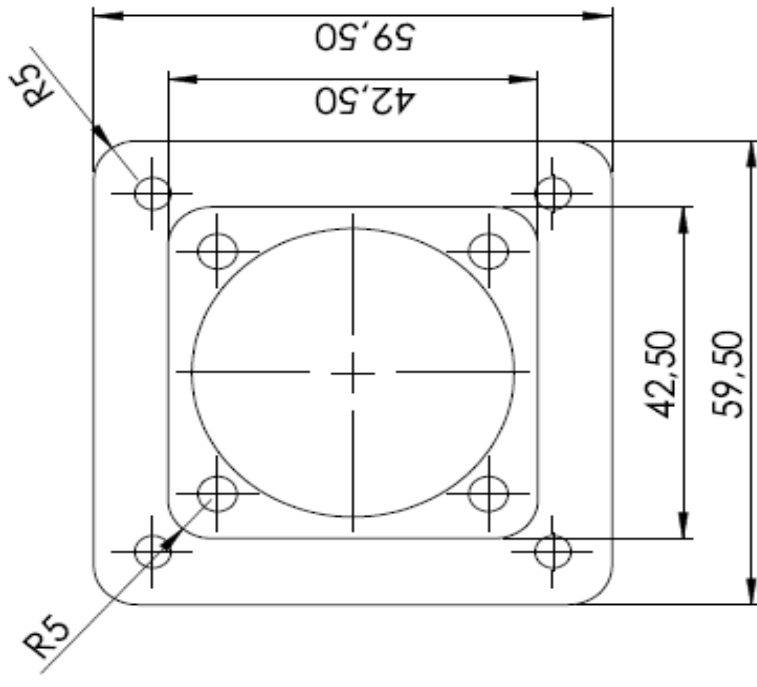
Revisão: 1

Data
19/04/2018

Samuel Filippini

Pedro Bernardo

Testemunha:



Tolerância geral de 0,1mm

Escola Politécnica da Universidade de São Paulo PMR3500		Projeto: Posicionador Cartesiano	
Trabalho Final	Título		
Folha	1/1		
Material	Alumínio 6351	Quantidade:	3
Revisão:	1	Revisão:	1
Escala	1:1	Data	31/05/2018
Responsável:	Samuel Monção		
Orientador:	Prof. Rodrigo Stoeterau		

1/1 Suporte Mancal FK12

APÊNDICE B – ALGORITMOS UTILIZADOS

AQUICURv2

```
#include <Arduino.h>
#include "BasicStepperDriver.h"
#include "Encoder.h"
#include "DueTimer.h"
#define MOTOR_STEPS 200
#define MOTOR_X_RPM 60
//#define DIR A8
//#define STEP A9
//#define ENABLE 48
//#define DIR A10
//#define STEP A11
//#define ENABLE 46
#define MICROSTEPS 16
int rev = 200;
BasicStepperDriver stepper(MOTOR_STEPS, DIR,
STEP, ENABLE);
//Encoder myEnc(10, 11);
Encoder myEnc(3, 4);
void setup() {
Serial.begin(9600);
stepper.begin(MOTOR_X_RPM, MICROSTEPS);
Timer3.attachInterrupt(myHandler);
Timer3.start(7500);
}
void loop() {
stepper.enable();
```

```
stepper.move(rev);
delay(2000);
stepper.move(rev);
delay(2000);
}
void myHandler(){
long newPosition = myEnc.read();
long newPositionN = newPosition*0.04*16;
Serial.println(newPositionN);
}
```

TF1V8

```
#include <Arduino.h>
#include "PID_v1.h"
#include "BasicStepperDriver.h"
#include "Encoder.h"
#include "DueTimer.h"
#define PASSO_DO_MOTOR 200
#define RPM_DO_MOTOR 60
#define DIR_X A10
#define PASSO_X A11
#define ABILITA_X 46
#define DIR_Y A8
#define PASSO_Y A9
#define ABILITA_Y 48
#define MICROPASSO 16
BasicStepperDriver stepper_X(PASSO_DO_MOTOR, DIR_X, PASSO_X,
ABILITA_X);
```

```
BasicStepperDriver stepper_Y(PASSO_DO_MOTOR, DIR_Y, PASSO_Y,
ABILITA_Y);
Encoder myEnc_X(3, 4);
Encoder myEnc_Y(10, 11);
int XaxisENCPoS;
int XaxisENCPoSN;
int pos_X = 640;
int rev_X;
int XaxisInput;
int XaxisInputNew;
int XaxisOutput;
int XaxisSetpoint;
int YaxisENCPoS;
int YaxisENCPoSN;
int pos_Y = 640;
int rev_Y;
int YaxisInput;
int YaxisInputNew;
int YaxisOutput;
int YaxisSetpoint;
float KpX=0.9;
float KiX=0.011;
float KdX=0.002;
float KpY=0.087;
float KiY=0.001;
float KdY=0.002;
PID XaxisPID(&XaxisInput, &XaxisOutput, &XaxisSetpoint, KpX, KiX,
KdX, DIRECT);
```

```
PID YaxisPID(&YaxisInput, &YaxisOutput, &YaxisSetpoint, KpY, KiY,
KdY, DIRECT);
const int sampleRate = 1;
long int reportTime;
int pronto = 0;
void setup() {
  Serial.begin(9600);
  stepper_X.begin(RPM_DO_MOTOR, MICROPASSO);
  stepper_Y.begin(RPM_DO_MOTOR, MICROPASSO);
  Timer3.attachInterrupt(LER_ENCODER).start(7000);
  XaxisPID.SetMode(AUTOMATIC);
  XaxisPID.SetSampleTime(sampleRate);
  YaxisPID.SetMode(AUTOMATIC);
  YaxisPID.SetSampleTime(sampleRate);
}
void loop() {
  int x1 = 640;
  int y1 = 640;
  int x2 = 0;
  int y2 = 1280;
  int x3 = -640;
  int y3 = 640;
  int x4 = 0;
  int y4 = 0;
  VAI_X_VAI_Y(x1, y1);
  delay(1000);
  RET_X_VAI_Y(x2, y2);
  delay(1000);
```

```

RET_X_RET_Y(x3, y3);
delay(1000);
VAI_X_RET_Y(x4, y4);
delay(1000);
}
void LER_ENCODER(){
int XaxisENCPos = myEnc_X.read();
int YaxisENCPos = myEnc_Y.read();
XaxisENCPosN = XaxisENCPos*(0.04*16);
YaxisENCPosN = YaxisENCPos*(-0.04*16);
Serial.print("POS X");
Serial.print("\t");
Serial.print(XaxisENCPosN);
Serial.print("\t");
Serial.print("POS Y");
Serial.print("\t");
Serial.print(YaxisENCPosN);
Serial.println();
}
void VAI_X_VAI_Y(int x, int y){
XaxisSetpoint = x;
YaxisSetpoint = y;
XaxisPID.SetControllerDirection(DIRECT);
YaxisPID.SetControllerDirection(DIRECT);
while(XaxisInput != XaxisSetpoint && XaxisInputNew !=
XaxisSetpoint && YaxisInput != YaxisSetpoint && YaxisInputNew !=
YaxisSetpoint) {
XaxisInput = XaxisENCPosN;

```

```

YaxisInput = YaxisENCPosN;
XaxisPID.Compute();
YaxisPID.Compute();
rev_X = XaxisOutput;
rev_Y = YaxisOutput;
stepper_X.enable();
stepper_Y.enable();
if(XaxisSetpoint > XaxisInput) stepper_X.move(rev_X);
else if(XaxisSetpoint < XaxisInput) stepper_X.
move(-rev_X);
if(YaxisSetpoint > YaxisInput) stepper_Y.move(rev_Y);
else if(YaxisSetpoint < YaxisInput) stepper_Y.move(-rev_Y);
delay(100);
XaxisInputNew = XaxisENCPosN;
YaxisInputNew = YaxisENCPosN;
}
stepper_X.disable();
stepper_Y.disable();
}
void RET_X_VAI_Y(int x, int y){
XaxisSetpoint = x;
YaxisSetpoint = y;
XaxisPID.SetControllerDirection(REVERSE);
YaxisPID.SetControllerDirection(DIRECT);
while(XaxisInput != XaxisSetpoint && XaxisInputNew !=
XaxisSetpoint && YaxisInput != YaxisSetpoint && YaxisInputNew !=
YaxisSetpoint) {
XaxisInput = XaxisENCPosN;

```

```

YaxisInput = YaxisENCPosN;
XaxisPID.Compute();
YaxisPID.Compute();
rev_X = XaxisOutput;
rev_Y = YaxisOutput;
stepper_X.enable();
stepper_Y.enable();
if(XaxisSetpoint > XaxisInput) stepper_X.move(rev_X);
else if(XaxisSetpoint < XaxisInput) stepper_X.
move(-rev_X);
if(YaxisSetpoint > YaxisInput) stepper_Y.move(rev_Y);
else if(YaxisSetpoint < YaxisInput) stepper_Y.move(-rev_Y);
delay(100);
XaxisInputNew = XaxisENCPosN;
YaxisInputNew = YaxisENCPosN;
}
stepper_X.disable();
stepper_Y.disable();
}
void RET_X_RET_Y(int x, int y){
XaxisSetpoint = x;
YaxisSetpoint = y;
XaxisPID.SetControllerDirection(REVERSE);
YaxisPID.SetControllerDirection(REVERSE);
while(XaxisInput != XaxisSetpoint && XaxisInputNew !=
XaxisSetpoint && YaxisInput != YaxisSetpoint && YaxisInputNew !=
YaxisSetpoint) {
XaxisInput = XaxisENCPosN;

```

```

YaxisInput = YaxisENCPosN;
XaxisPID.Compute();
YaxisPID.Compute();
rev_X = XaxisOutput;
rev_Y = YaxisOutput;
stepper_X.enable();
stepper_Y.enable();
if(XaxisSetpoint > XaxisInput) stepper_X.move(rev_X);
else if(XaxisSetpoint < XaxisInput) stepper_X.
move(-rev_X);
if(YaxisSetpoint > YaxisInput) stepper_Y.move(rev_Y);
else if(YaxisSetpoint < YaxisInput) stepper_Y.move(-rev_Y);
delay(100);
XaxisInputNew = XaxisENCPosN;
YaxisInputNew = YaxisENCPosN;
}
stepper_X.disable();
stepper_Y.disable();
}
void VAI_X_RET_Y(int x, int y){
XaxisSetpoint = x;
YaxisSetpoint = y;
XaxisPID.SetControllerDirection(DIRECT);
YaxisPID.SetControllerDirection(REVERSE);
while(XaxisInput != XaxisSetpoint && XaxisInputNew !=
XaxisSetpoint && YaxisInput != YaxisSetpoint && YaxisInputNew !=
YaxisSetpoint) {
XaxisInput = XaxisENCPosN;

```

```
YaxisInput = YaxisENCPosN;
XaxisPID.Compute();
YaxisPID.Compute();
rev_X = XaxisOutput;
rev_Y = YaxisOutput;
stepper_X.enable();
stepper_Y.enable();
if(XaxisSetpoint > XaxisInput) stepper_X.move(rev_X);
else if(XaxisSetpoint < XaxisInput) stepper_X.
move(-rev_X);
if(YaxisSetpoint > YaxisInput) stepper_Y.move(rev_Y);
else if(YaxisSetpoint < YaxisInput) stepper_Y.move(-rev_Y);
delay(100);
XaxisInputNew = XaxisENCPosN;
YaxisInputNew = YaxisENCPosN;
}
stepper_X.disable();
stepper_Y.disable();
}
```

