

MARCOS VINICIUS COELHO GEROMEL
VITOR SIQUEIRA DO CARMO

Desenvolvimento de uma máquina fresadora de arquitetura paralela para fins acadêmicos

São Paulo

2013

MARCOS VINICIUS COELHO GEROMEL
VITOR SIQUEIRA DO CARMO

Desenvolvimento de uma máquina fresadora de arquitetura paralela para fins acadêmicos

Trabalho de conclusão de curso apresentado
à Escola Politécnica da Universidade de São
Paulo para obtenção do título de Bacharel
em Engenharia Mecatrônica.

Orientador: Prof. Dr. Tarcísio Antônio Hess
Coelho

São Paulo
2013

MARCOS VINICIUS COELHO GEROMEL
VITOR SIQUEIRA DO CARMO

Desenvolvimento de uma máquina fresadora de arquitetura paralela para fins acadêmicos

Trabalho de conclusão de curso apresentado
à Escola Politécnica da Universidade de São
Paulo para obtenção do título de Bacharel
em Engenharia Mecatrônica.

Área de Concentração: Engenharia Me-
catrônica

Orientador: Prof. Dr. Tarcísio Antônio Hess
Coelho

São Paulo
2013

FICHA CATALOGRÁFICA

Geromel, Marcos Vinicius Coelho

Desenvolvimento de uma máquina fresadora de arquitetura paralela para fins acadêmicos / M.V.C. Geromel, V.S. do Carmo. -- São Paulo, 2014.

96 p.

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

1.Fresadoras 2.Precisão do teste 3.Arquiteturas paralelas I.Carmo, Vitor Siqueira do II.Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia Mecatrônica e de Sistemas Mecânicos III. t.

AGRADECIMENTOS

Agradecemos primeiramente ao nosso orientador Prof. Dr. Tarcisio Coelho pelos seus ensinamentos, sua cooperação e, principalmente, sua paciência. Somos gratos também ao aluno de mestrado Brian pela divisão de seus conhecimentos e por toda a ajuda prestada.

“A educação tem raízes amargas, mas seus
frutos são doces” – Aristóteles

RESUMO

Mecanismos seriais são os mais utilizados na indústria e vêm sendo usados há bastante tempo. Seus atuadores e peças movidas são dispostas em série, formando uma única cadeia cinemática aberta, na medida em que os braços se unem a convergir num “ponto” central, geralmente uma garra ou um eletrodo de solda. Porém, esses robôs apresentam desempenho insatisfatório quando utilizados para atividades que necessitam de precisão, rigidez, alta frequência natural e baixo tempo de ciclo. Por conta disto, os estudos sobre mecanismos paralelos vêm crescendo a cada dia tanto na indústria como na área acadêmica. Mecanismos paralelos são caracterizados pela presença de várias cadeias cinemáticas independentes, atuando de forma paralela e simultânea sobre o “ponto” central. Essas características apresentam grandes vantagens em relação aos robôs de arquitetura serial: alta rigidez, leveza, rapidez, precisão e alta capacidade de carga. O robô satisfaz os seguintes requisitos: apenas três eixos para a fresagem, contendo um atuador cada, operando paralelamente em conjunto e utilizando componentes padronizados de baixo custo.

Palavras-Chave: Arquitetura paralela. Fresadora. Precisão

ABSTRACT

Serial mechanisms are most commonly used in industry and have been used for quite some time. Its actuators and driven parts are arranged in series, forming a single open kinematic chain, in that bind to converge on a "dot" core, usually a clamshell or electrode welding. However, these robots have poor performance when used for activities that require precision, rigidity, high natural frequency and low-cycle time. Therefore, studies on parallel mechanisms have been growing every day both in industry and in academia. Parallel mechanisms are characterized by the presence of several independent kinematic chains, operating in parallel and simultaneously over a central point. These characteristics present major advantages over robots serial architecture: high rigidity, lightness, speed, accuracy and high load capacity.

The robot meets the following requirements: only three axes for milling, each containing a plunger, operating together in parallel and the use of inexpensive standard components.

Keywords: Parallel architectures. Milling. Precision.

LISTA DE FIGURAS

Figura 1: Robô paralelo idealizado por V.E. Gough	17
Figura 2: Plataforma de Stewart	17
Figura 3: Robô paralelo do tipo delta	18
Figura 4: Necessidades do projeto.....	20
Figura 5: Benefícios da máquina para os alunos	21
Figura 6: Exemplos de máquinas.....	22
Figura 7: Possíveis configurações de máquina.....	24
Figura 8: Deflexão na origem com carga de 3500N direção horizontal e 2 braços	25
Figura 9: Deflexão na origem com carga de 3500N direção vertical e 2 braços.....	25
Figura 10: Deflexão na origem com carga de 3500N a 45 graus e 2 braços	26
Figura 11: Deflexão na origem com carga de 3500N direção horizontal e 3 braços	26
Figura 12: Deflexão na origem com carga de 3500N direção vertical e 3 braços.....	27
Figura 13: Deflexão na origem com carga de 3500N a 45 graus e 3 braços	27
Figura 14: Cinemática de posicionamento	29
Figura 15: Conexões entre computador, driver e motores.....	29
Figura 16: Software controlador dos motores de passo	30
Figura 17: Diagrama do fluxo de comandos da máquina.....	31
Figura 18: Comunicação entre os programas e a máquina.....	31
Figura 19: Vista superior do robô.....	32
Figura 20: Vista Frontal (Motor de passo, Fuso, braço+guia)	33
Figura 21: Representação dos contrapesos nos braços.....	36
Figura 22: Utilização de uma Guia deslizante auxiliar.....	36
Figura 23: Proposta de fixação dos braços à ferramenta em um mesmo plano.....	37
Figura 24: Acoplamento com rolamentos de esferas em pontos espaçados.....	37
Figura 25: Cubo e rolamentos utilizados.....	38
Figura 26: Associação de porcas no fuso	39
Figura 27: Esquema da porca no fuso sem deflexão	40
Figura 28: Esquema da porca do fuso com deflexão máxima.....	40

Figura 29: Cubo do eixo usinado.....	42
Figura 30: Vista em corte do alojamento dos rolamentos no cubo	43
Figura 31: Vistas 3D do desenho de fabricação	43
Figura 32: Vista 3D do espaçador	44
Figura 33: Espaçador usinado em alumínio	44
Figura 34: Montagem do novo acoplamento. Guia, fuso e eixo do braço.....	45
Figura 35: Braços montados e área de trabalho.....	46
Figura 36: Configuração antiga	47
Figura 37: Cabo paralelo macho-macho.....	48
Figura 38: Caixa alojamento dos drivers.....	48
Figura 39: Driver Gecko G251.....	49
Figura 40: Drivers Gecko e distribuição de fios.....	50
Figura 41: Conector DB25 fêmea.....	51
Figura 42: Montagem para aplicação cargas nos braços.....	52
Figura 43: Montagem de três braços com laser e força aplicada na extremidade do braço.....	53
Figura 44: Controlador do laser e osciloscópio.....	54
Figura 45: Configuração do laser(1).....	55
Figura 46: Curva de deflexão no osciloscópio(1).....	55
Figura 47: Curva de deflexão no osciloscópio(2).....	56
Figura 48: Curva de deflexão no osciloscópio(3).....	56
Figura 49: Curva de deflexão no osciloscópio(4).....	57
Figura 50: Configuração no laser(2).....	58
Figura 51: Curva de deflexão no osciloscópio(5).....	58
Figura 52: Curva de deflexão no osciloscópio(6).....	59
Figura 53: Curva de deflexão no osciloscópio(7).....	59
Figura 54: Curva de deflexão no osciloscópio(8).....	60
Figura 55: Configuração no laser(3).....	61
Figura 56: Curva de deflexão no Matlab(1).....	61
Figura 57: Curva de deflexão no Matlab(2).....	62
Figura 58: Curva de deflexão no Matlab(3).....	62
Figura 59: Curva de deflexão no Matlab(4).....	63

Figura 60: Curva de deflexão no Matlab(5).....	63
Figura 61: Curva de deflexão no Matlab(6).....	64
Figura 62: Curva de deflexão no Matlab(7).....	64
Figura 63: Curva de deflexão no Matlab(8).....	65
Figura 64: Curva de deflexão no Matlab(9).....	65

LISTA DE TABELAS

Tabela 1: Precisão de máquinas seriais e paralelas	21
Tabela 2: Velocidades máximas de máquinas seriais e paralelas.....	22
Tabela 3: Relação entre as juntas e os ligamentos.....	23
Tabela 4: Pinos do conector fêmea.....	50
Tabela 5: Deflexão calculada para carga aplicada, em mm.....	66

SUMÁRIO

1. INTRODUÇÃO.....	16
1.1 Objetivos.....	19
2. REVISÃO BIBLIOGRÁFICA.....	23
2.1 Metodologia para análise e síntese de mecanismo	23
2.1.1 Síntese topológica da máquina	23
2.1.2 Simulação com 2 e 3 braços	24
2.1.2.1 Simulação com 2 braços	24
2.1.2.2 Simulação com 3 braços	26
2.2 Cinemática inversa	28
2.3 Elétrica.....	29
2.4 Softwares utilizados.....	29
2.5 Controle da fresadora.....	31
3. O PROJETO	32
3.1 Estágio inicial	32
3.2 Metodologia de projeto.....	33
3.3 Parte mecânica.....	34
3.4 Parte elétrica	34
3.5 Controle	34
4. ANÁLISE DAS SOLUÇÕES PROPOSTA.....	35
4.1 Mecânica.....	35
4.2 Elétrica.....	40
4.3 Controle	41
5. SOLUÇÕES ADOTADAS	41
5.1 Parte mecânica.....	41
5.1.1 Acoplamento.....	41
5.1.2 Espaçador	43
5.1.3 Montagem.....	45
5.2 Parte elétrica	47
5.3 Controle	51

6. TESTES E DISCUSSÕES DE RESULTADOS	52
7. CONCLUSÕES	67
8. REFERÊNCIAS BIBLIOGRÁFICAS	68
ANEXO A - DESENHOS DE FABRICAÇÃO	70
ANEXO B - CONFIGURAÇÃO DA MÁQUINA NO EMC2	72
ANEXO C - MANUAL DA FRESADORA	75
ANEXO D - PROGRAMAS DO SCILAB	77
ANEXO E - DATA SHEET- DRIVER GECKO G251	94

1. INTRODUÇÃO

O robô paralelo consiste principalmente na plataforma móvel, a plataforma fixa, o mecanismo paralelo e o fim do mecanismo de acionamento unido pelas várias partes chamado de órgão terminal.

A precisão do robô paralelo é, dentre outros fatores, o responsável por um dos seus principais índices de desempenho, dessa forma, ele pode ser classificado como de qualidade superior de usinagem ou de qualidade inferior; quanto maior a exatidão com que o robô paralelo executa os seus comandos maior a sua posição na escala de qualidade, aumentando assim o seu valor intrínseco, tanto para a indústria quanto para outras aplicações.

Dessa maneira, é necessário calcular a precisão do robô paralelo utilizando-se métodos simples e rápidos, a fim de usá-los em procedimentos de otimização de design em que se procura o máximo de precisão.

Erros na posição e orientação de um robô paralelo são devidos a vários fatores:

- Erros de fabricação, que podem ser tidos em conta através de calibração;
- Folgas, que pode ser eliminado por meio da escolha adequada dos componentes mecânicos;
- Conformidade, que também podem ser eliminados através da utilização de estruturas mais rígidas (embora isto faria aumentar a inércia e diminuir a velocidade de operação);
- Erros de sensor e os erros de controle.

Existe uma serie de protótipos de robôs paralelos com diferentes aplicações e uma alta aceitação no mercado industrial, pelas vantagens que oferecem. A partir dos estudos sobre poliedros e suas aplicações, o primeiro mecanismo paralelo foi patenteado em 1931. Se tratava de uma plataforma de movimento desenvolvida por James E. Gwinnett (Merlet, 2006) e era destinada à indústria de entretenimento.

Pouco tempo depois, em 1940, surge o primeiro robô industrial paralelo. Willard L.V. Pollard (Merlet, 2006) apresentou um robô de 5 graus de liberdade destinado a operações de pintura com *spray*. O robô consistia de três braços com duas ligações cada, unidas por juntas universais e três atuadores da base que comandavam a ferramenta. A orientação era fornecida por dois atuadores localizados na base que transmitiam o movimento para a ferramenta por rotação através de cabos.

Em 1947, Dr. Eric Gough (Merlet, 2006) inventou a plataforma paralela mais popular de todas, ilustrado na figura 1. Dr. Gough desenhou um hexápode octaedro com lados de comprimento variável como plataforma para verificar o comportamento dos pneus da casa Dunlop para a qual trabalhava, sob cargas aplicadas em diferentes eixos. Assim, tentava simular o processo de aterrissagem de um avião. Hoje em dia, há muitas plataformas com base nesta concepção, em muitas empresas, e são conhecidas sob o nome de MAST (*Multi-Axis Simulation Table*).

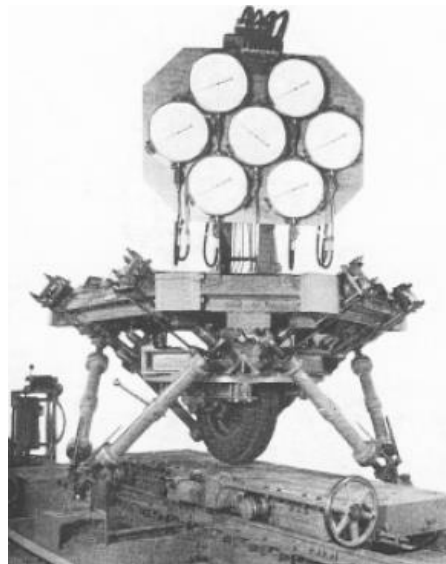


Figura 1: Robô paralelo idealizado por V.E. Gough (Merlet, 2006)

Em 1965, o Sr. Stewart (Merlet, 2006) apresentou um documento no qual descreve uma plataforma de movimento de 6 graus da liberdade projetada para funcionar como simulador de voo. veja a Figura 2.

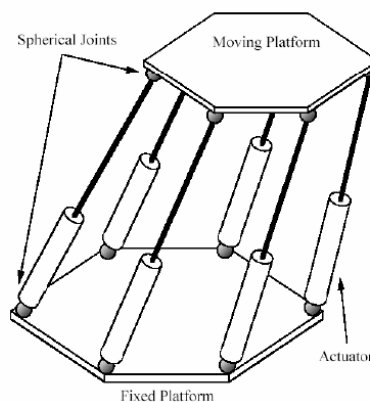


Figura 2: Plataforma de Stewart (Merlet, 2006)

Além da excelente capacidade de carga/peso em si, robôs paralelos têm outras características favoráveis. Em 1979, McCallion e Pham (Merlet, 2006) foram os primeiros que propuseram usar a plataforma de Stewart como manipulador paralelo para células de montagem robotizada, particularmente porque a posição do instrumento terminal é muito menos sensível a erros de sensores articulares quando comparado a dos robôs seriais. A rigidez assegura que as deformações dos atuadores são mínimas, e esta característica contribui grandemente para a precisão no posicionamento do manipulador.

Em 1990, Raymond Clavel (Merlet, 2006) realizou uma modificação no manipulador Stewart que apresentava desvantagens em comparação com a análise de cinemática e custo alto na fabricação de juntas esféricas por sua demanda em precisão quando construído. Assim surge o manipulador Delta de três e quatro graus de liberdade, o que supera os inconvenientes mencionados anteriormente, veja a Figura 3.



Figura 3: Robô paralelo do tipo delta: a) Três graus de liberdade b) Quatro graus de liberdade.

Fonte: <<http://www.robotmatrix.org/Parallelrobot.htm>> IRB 340 Flexpicker system – ABB

Os manipuladores paralelos têm atraído a atenção por causa das vantagens que eles oferecem em comparação com os manipuladores seriais em áreas como instrumentos de medicina e da máquina.

Existem três tipos de robôs manipuladores paralelos que são construídos em nível comercial e que possuem diferentes tipos de aplicações, são eles (Tsai, Lung-Wen. 1999):

- Do robô Delta desenvolvido por R. Carnation calcula-se que existem mais de 4000 unidades em operação em todo o mundo, e é usado principalmente em indústrias de manipulação de alimentos.

- O robô patenteado Tricept K.N. Neumann, que estima-se que existam cerca de 300 unidades, que é usado em máquinas-ferramentas de média precisão e ingressando no campo da medicina.

- A plataforma Gough é o manipulador paralelo mais integrado com cerca de 20.000 unidades e é utilizado para orientação de antenas, telescópios, painéis solares, isolamento e produção de vibração, posicionamento de microscópios e pacientes, simuladores de vôo de avião, empilhadeiras, montagem de componentes, posicionamento de peças e como máquina ferramenta de média precisão.

1.1 Objetivos

Este trabalho é uma continuidade de alguns trabalhos realizados previamente por alunos de graduação e de mestrado; partindo de um protótipo já projetado e concluído, pode-se aprofundar no desenvolvimento das etapas componentes do curso, criando condições favoráveis ao desenvolvimento da teoria mecânica, elétrica e de controle.

Os trabalhos que originaram o protótipo foram: “Desenvolvimento do Protótipo de um Robô de Cinemática Paralela para Usinagem 2D com uma Barra de Ancoragem” e “Uma contribuição para o desenvolvimento de uma máquina fresadora de arquitetura paralela” desenvolvidos pelo aluno Vitor Never Hartmann e “Fabricação de um robô de cinemática paralela para usinagem 2D” efetuado pelo aluno Rodrigo Yussuke Nojiri, todos orientados pelo Prof. Dr. Tarcísio Antonio Hess Coelho.

A máquina de arquitetura paralela foi projetada e construída por esses dois alunos e tinha o objetivo de elaborar um robô com alta robustez e grande precisão de operação. Para isso, elaboraram uma máquina com 2 graus de liberdade, utilizando 3 braços contendo um atuador em cada, trabalhando em conjunto, formando uma estrutura cinemática redundante. A base da ferramenta que a move na direção perpendicular à máquina foi projetada, porém não foi construída.

O projeto, segundo os critérios sugeridos por Hartmann para atingir os objetivos da academia, tem como requisito: 3 braços de usinagem (1 em redundância), construção modular e componentes com baixo custo. Expressos na figura 4.

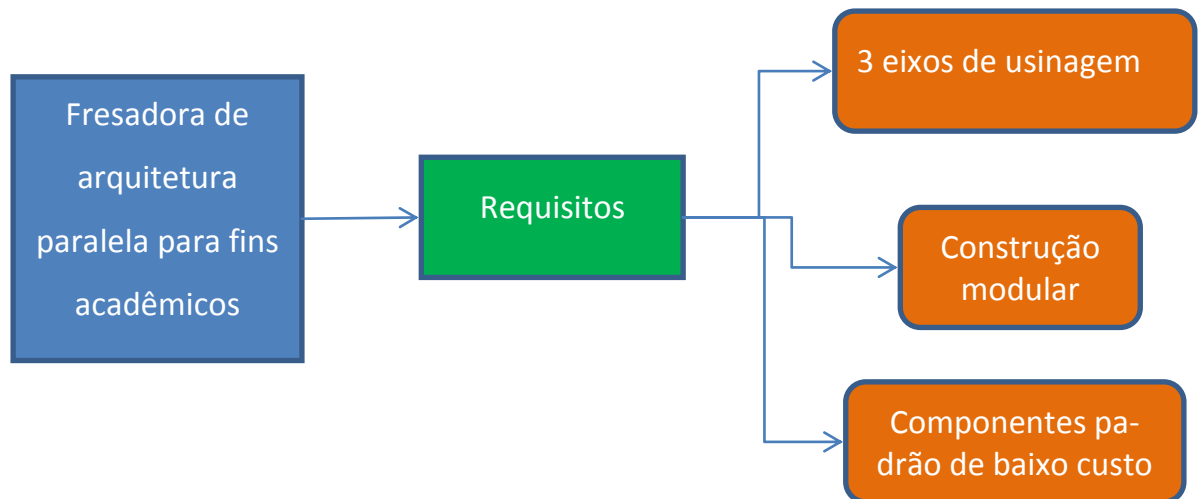


Figura 4 : Necessidades do projeto (Hartmann, 2013)

O objetivo final é a construção e o desenvolvimento de um robô fresador de arquitetura paralela para fins acadêmicos. Sendo a máquina, destinada principalmente para os alunos de graduação.

Primeiro de tudo, os alunos terão o primeiro contato com a arquitetura paralela, um modelo novo de mecanismo, se levado em consideração o tempo que a arquitetura serial foi implantada na indústria.

Um segundo passo, é o estudo das simulações de modelagem, desde as análises estáticas, cinemáticas e a avaliação do espaço de trabalho, até a parte experimental do mecanismo pronto, como a análise de erro e o controle do robô.

Por fim, os alunos poderão operar com a fresadora para conhecimento da linguagem de máquina, o código G, e usar algumas peças.

Em síntese, os benefícios para o meio acadêmico de um robô paralelo são apresentados no fluxograma da figura 5.

Com base nos dados da tabela 1, pode-se perceber que a precisão e a repetibilidade da Arrow 1000 (arquitetura serial) são bem melhores que a Tornado Hexel e a Tricept (ambas de arquitetura paralela). A partir desses resultados, a conclusão errada seria que se paras se de desenvolver os robôs paralelos e continuasse com os seriais. Porém foi visto que a nova arquitetura traz muitas vantagens que em relação à antiga, como a velocidade. As máquinas paralelas apresentam maiores velocidades de operação conforme ilustrado na tabela 2. Então, o desafio é devolver métodos para que a precisão dos mecanismos paralelos seja aprimorada e chegue próxima ao dos

mecanismos seriais; neste projeto, espera-se que a máquina atinja uma precisão de 0,2mm após as soluções desenvolvidas.

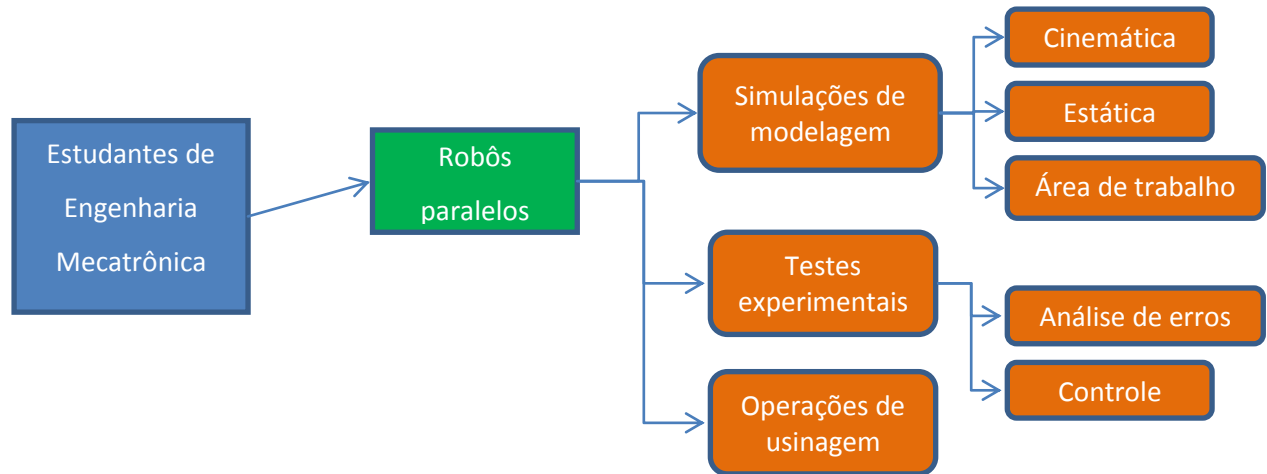


Figura 5: Benefícios da máquina para os alunos (Hartmann, 2013)

Portanto, o escopo deste trabalho abrange aproveitar os projetos e as análises desenvolvidas por esses alunos, usando-as de uma forma crítica para aferir possíveis erros e melhorias, na medida em que serão utilizadas novas soluções para diminuir as folgas do sistema, a calibração da máquina e o aperfeiçoamento das partes elétricas e de controle.

Tabela 1: Precisão de máquinas seriais e paralelas

Máquinas	Tipo de estrutura	Exatidão [μm]	Repetitividade [μm]
Arrow 1000	Serial	4	1
Tornado Hexel	Parallel	25	10
Tricept	Parallel	200	20

Fonte: Hartmann (2013)

Tabela 2: Velocidades máximas de máquinas seriais e paralelas

Máquinas	Tipo de estrutura	Velocidade (m/min)
Arrow 1000	Serial	25
SMT Tricept 805	Paralela	50
Genius 500	Paralela	180

Fonte: Hess-Coelho, Raszl (2006)

Exemplos de máquinas paralelas e seriais na figura 6.



Figura 6: Exemplos de máquinas a) Arrow 1000 b) Tornado Hexel c) Tricept

Fontes: a) <http://www.fpmiller.com/Inventory/Pages/3221.aspx>

b) <http://blog.naver.com/PostView.nhn?blogId=hcj2&logNo=24574190&redirect=Dlog&widgetTypeCall=tr>
[ue](#)

c) <http://www.abb.com.br/cawp/brabb155/4bdd5ed99f7eaa9203256d10005ddcdf.aspx>

REVISÃO BIBLIOGRÁFICA

Neste capítulo, são apresentadas as teorias, os métodos e os softwares que dão suporte ao projeto.

2.1 Metodologia para análise e síntese de mecanismos

2.1.1 Síntese topológica da máquina

O método usado para a síntese topológica é o critério de Kutzbach-Gruebler:

$$M = 3 \cdot (n - 1) - 2 \cdot n_{p1} - n_{p2}$$

Nessa equação, M é a mobilidade do sistema, n representa o número de ligamentos, np1 é o número de juntas com um grau de liberdade e np2, com dois graus de liberdade.

A partir dessa expressão, pode-se obter a relação entre o número de juntas e os ligamentos de acordo com a mobilidade desejada.

Nesta fresadora, deseja-se uma mobilidade igual a dois, referente aos eixos x e y. Assim, iterativamente foram calculadas as possíveis configurações da máquina, conforme a tabela 3.

Tabela 3: Relação entre as juntas e os ligamentos

np1	n
1	2,3
2	3,0
3	3,7
4	4,3
5	5,0
6	5,7
7	6,3
8	7,0

Visto que só se podem construir números inteiros de ligamentos, as três configurações possíveis estão esquematizadas na figura 7.

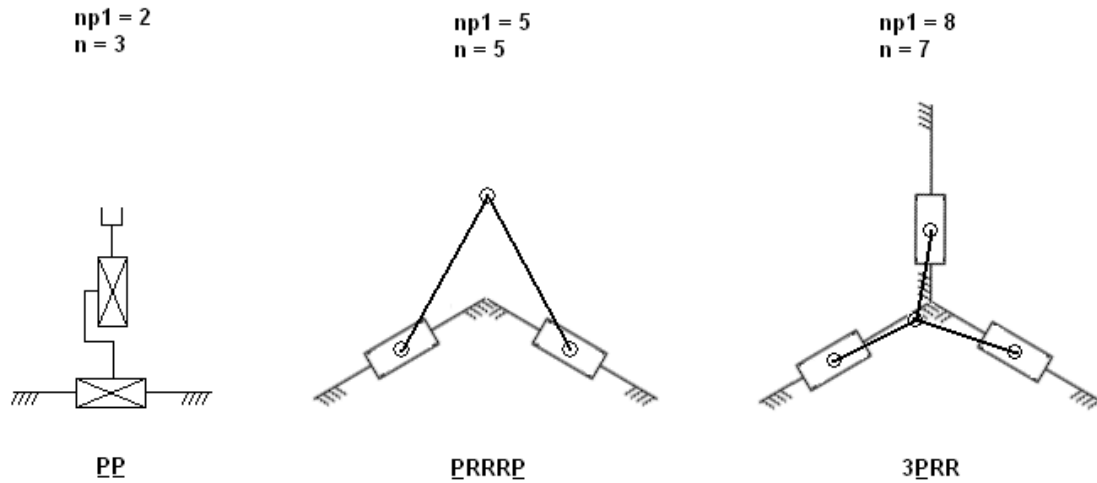


Figura 7 : Possíveis configurações de máquina (Hartman, 2011)

2.1.2 Simulação com 2 e 3 braços

Para a escolha da configuração final, os modelos de rigidez de 2 e 3 braços foram avaliados quando submetidos a cargas críticas (3500N) nas direções horizontal, vertical e 45 graus no plano 2D.

2.1.2.1 Simulação com 2 braços

Pela figura 8, pode-se perceber que quando aplicado um esforço na direção horizontal, não há deflexão na direção perpendicular a ela.

O mesmo acontece na direção vertical, visto na figura 9; não há deflexão na direção perpendicular a ela.

Já a 45 graus, figura 10, ocorre deflexão em ambas as direções.

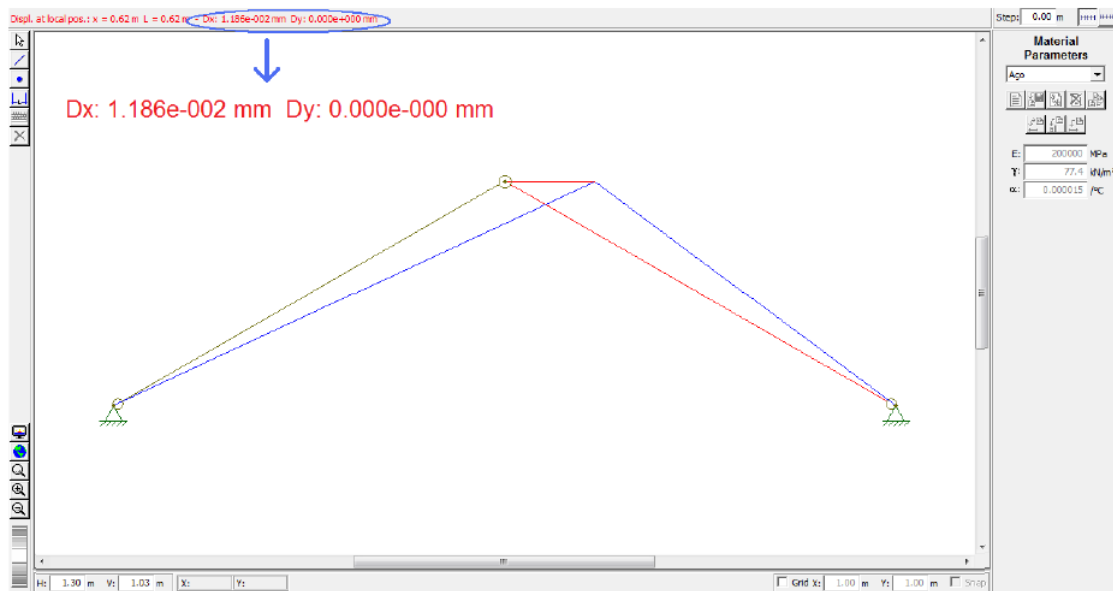


Figura 8: Deflexão na origem com carga de 3500N direção horizontal e 2 braços (Hartmann, 2011)

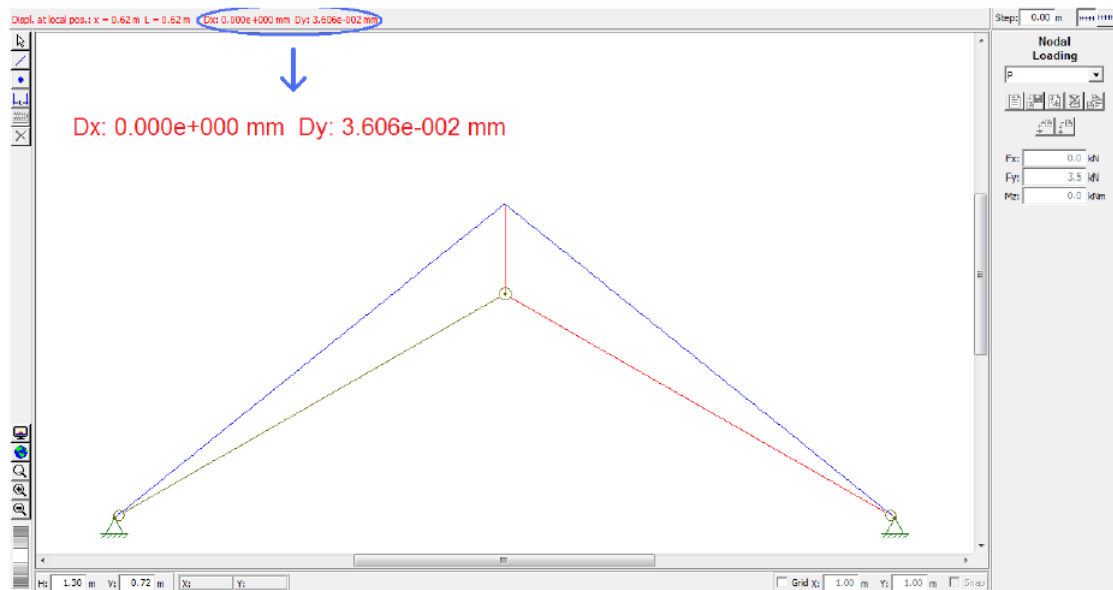


Figura 9: Deflexão na origem com carga de 3500N direção vertical e 2 braços (Hartmann, 2011)

A diferença ocorre quando há a presença de algum componente na direção vertical, apresentado na figura 12.

As deflexões, quando sujeito a uma força vertical, reduzem de uma ordem de 10 vezes e se distribuem nas direções x e y, devido a articulação da nova base.

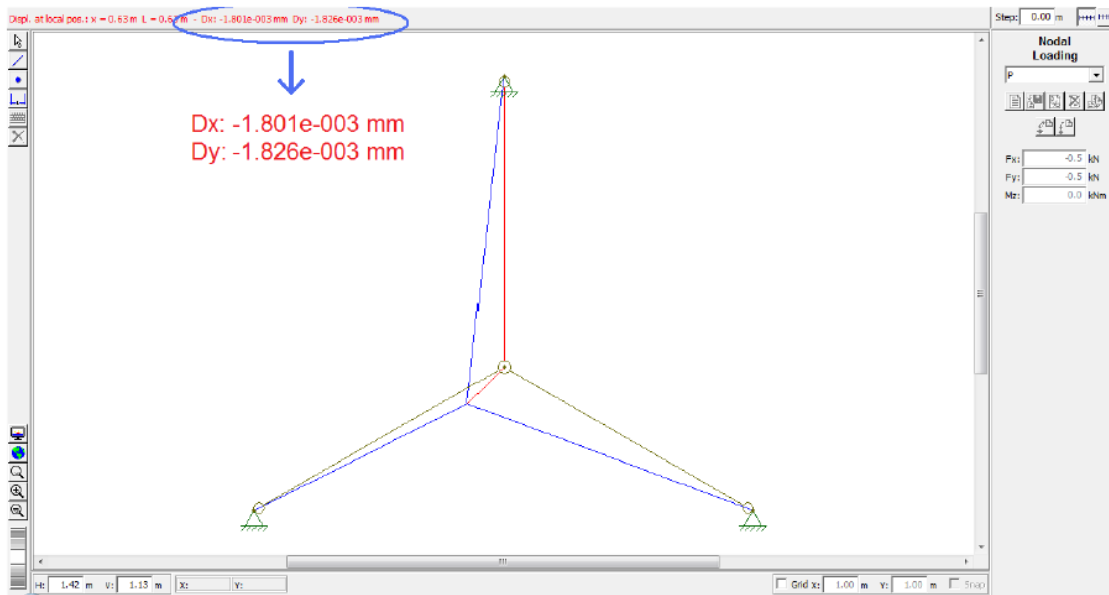


Figura 12: Deflexão na origem com carga de 3500N direção vertical e 3 braços (Hartmann, 2011)

Por último, em um esforço a 45 graus, a deflexão em x se mantém semelhante ao modelo com 2 braços, porém a deflexão em y se reduz em 2,5 vezes; figura 13.

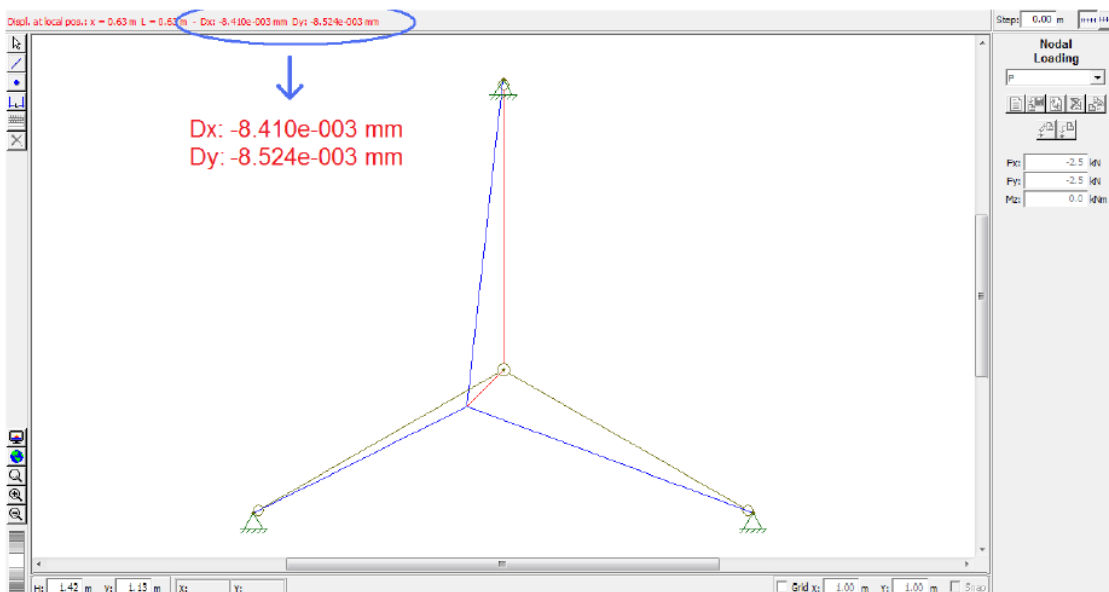


Figura 13: Deflexão na origem com carga de 3500N a 45 graus e 3 braços (Hartmann, 2011)

É importante salientar que essas simulações são simplificadoras pois foram feitas sem carga na direção z (pra dentro do plano) e não levam em consideração as folgas do sistema.

Com base nessas simulações, observa-se que a configuração de 3 braços apresenta uma melhor resposta às cargas aplicadas, sofrendo uma deflexão menor ou igual em todos os casos e, por isso, foi a configuração final escolhida.

A partir dessa escolha, surge a necessidade de analisar as cinemáticas desta máquina. tanto a direta, quanto a inversa.

2.2 Cinemática direta e inversa

O método direto consiste em determinar as coordenadas do ponto P , a partir de comprimentos l_1 , l_2 e l_3 conhecidos. Já no método inverso, conhece-se os comprimentos l_1 , l_2 e l_3 e as coordenadas do ponto $P(x,y)$ são determinadas.

Como o sistema tem redundância de 1 braço, pode-se resolver o sistema para dois atuadores e o terceiro fica em função deles.

A figura 14 ilustra a cinemática de posicionamento da configuração com 3 braços.

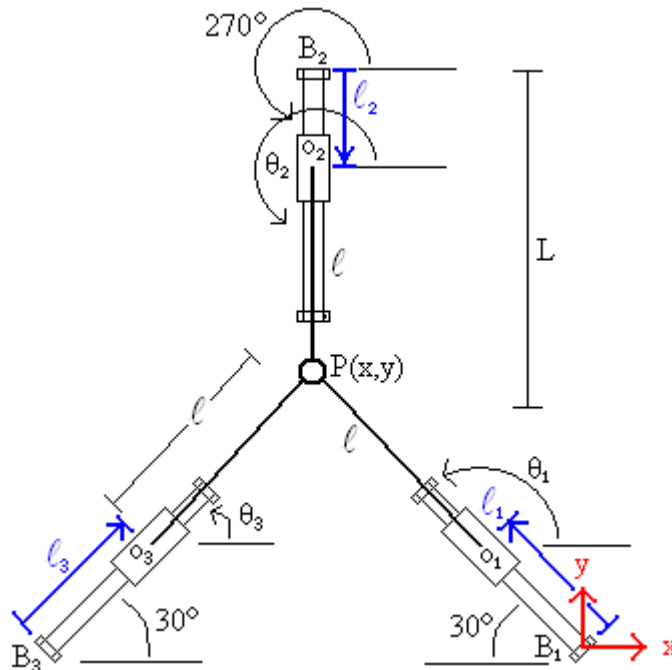


Figura 14: Cinemática de posicionamento (Gastineau,2011)

2.3 Eletrônica

A parte eletrônica da máquina está representada na figura 15.

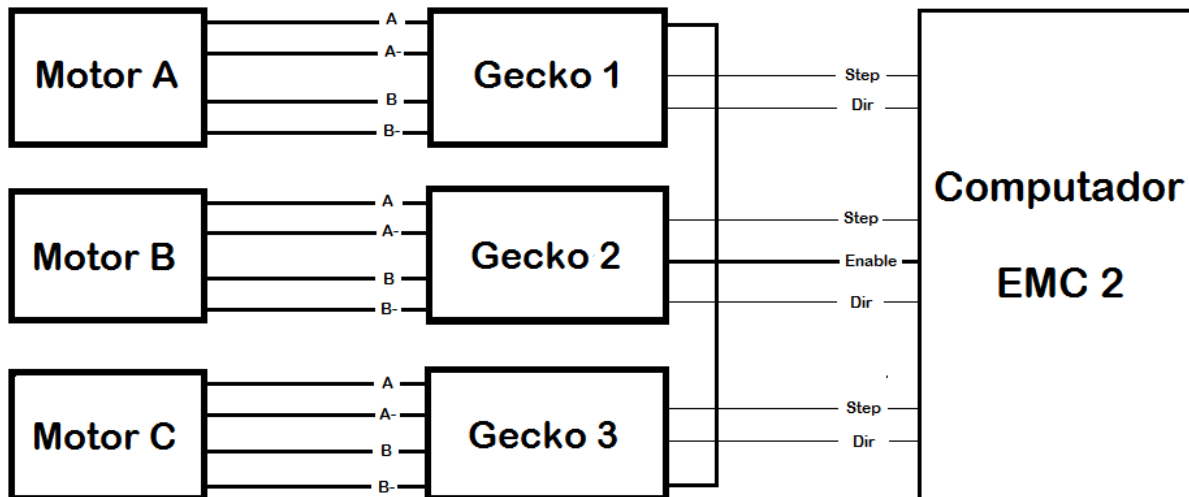


Figura 15: Conexões entre computador, driver e motores

Os sinais de comando são enviados do computador para os drivers Gecko pela porta paralela, sendo eles: o *Enable*, sinal que habilita todos os drivers; os *Steps*, que mandam os *clocks* e os *Directions*, que mandam as direções. O driver, por sua vez, interpreta esses sinais e controla os motores de passo através de 4 sinais (A, A-, B, B-), referentes às bobinas dos motores.

2.4 Softwares utilizados

O software EMC2 é baseado na metodologia de controle em tempo real do próprio NIST, e é programado usando as bibliotecas NIST RCS. O uso dessas bibliotecas na programação do software facilita o aporte dos códigos de controle em diversas plataformas Linux e Microsoft. O EMC2 pode controlar motores de passo, servo-motores, encoders, relés, e quaisquer outros equipamentos relacionados a máquinas de comando numérico. Nas próximas etapas, será mostrado como que o software foi configurado para este projeto, bem como instruções básicas de sua operação. A figura 16 mostra a participação do EMC2 com os *drivers* e os motores de passo.

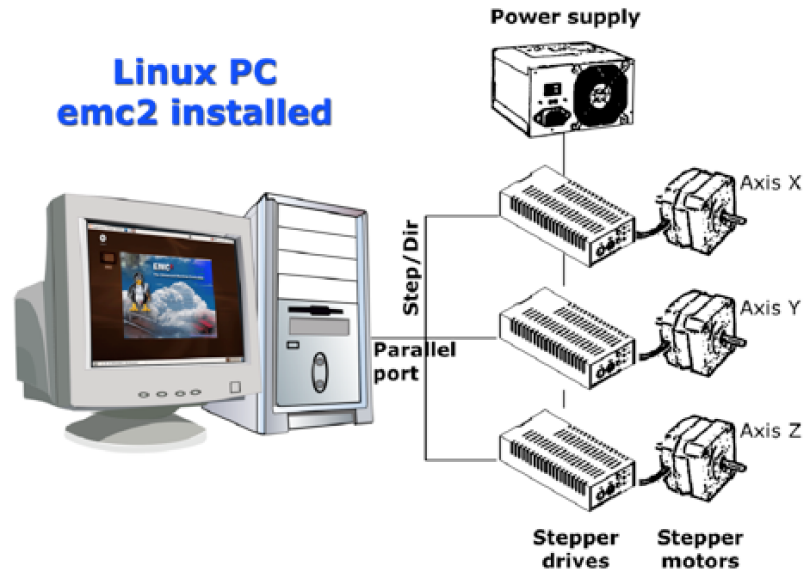


Figura 16: Software controlador dos motores de passo (Glogowsky, 2008)

Entender o software que responde pelo acionamento dos motores do robô paralelo é necessário para a completa percepção do trabalho, dessa maneira apresentamos o software EMC2 – Enhanced Machine Control, também conhecido como Linux CNC.

Partindo da análise histórica, o EMC2 teve o seu princípio na “Divisão de Sistemas Inteligentes” (Intelligent System Division) do “Instituto Nacional de Padrões e Tecnologia” (NIST - National Institute of Standards and Technology) dos Estados Unidos com o intuito de fornecer completa implementação via software para todos os membros da OMAC (“Organization for Machine Automation and Control”), colaboradora do NIST neste projeto, e, com isso, conseguir validação de interfaces de programação para máquinas e ferramentas. O outro objetivo era o de se desenvolver um canal de transferência de sistema de controle entre pequenos e médios fabricantes de máquinas e ferramentas.

O programa EMC2 foi idealizado seguindo a metodologia de controle em tempo real já existente no NIST, e sua programação se utiliza das bibliotecas NIST RCS. O uso dessas bibliotecas na linguagem do software facilita a combinação dos códigos de controle em diversas plataformas Linux e Microsoft.

O software EMC2 permite o controle de quaisquer equipamentos relacionados a máquinas de comando numérico. A configuração que foi necessária para que o programa se adequasse neste projeto, e também as instruções básicas para sua operação estão demonstradas no anexo B.

2.5 Controle da fresa

O diagrama de blocos da figura 17 apresenta como o controle do robô é feito.

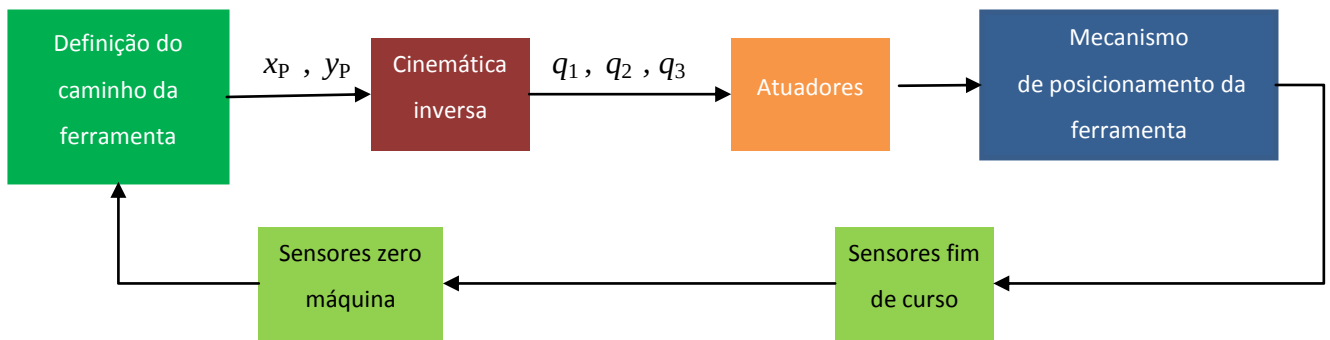


Figura 17: Diagrama do fluxo de comandos da máquina (Hartmann, 2013)

Primeiro, define-se a posição nos eixos x e y na interface do programa; este, que por sua vez, transforma as coordenadas recebidas em impulsos elétricos para os atuadores, que posicionam a ferramenta para a localização desejada.

Há um sistema de parada quando falha crítica, que são as chaves de fim de curso, assim que uma dessas chaves é tocada, a ferramenta volta para a posição inicial. O fluxograma é apresentado na figura 18.

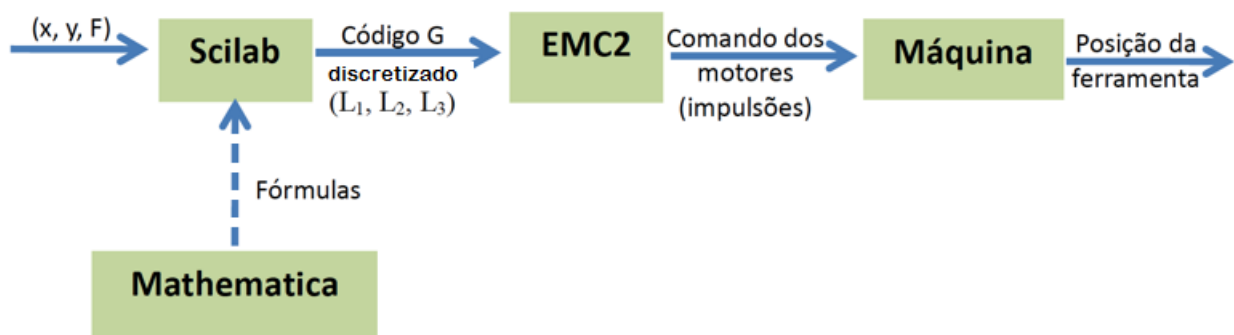


Figura 18: Comunicação entre os programas e a máquina (Gastineau, 2011)

Traduzindo o sistema em função dos programas utilizados, tem-se que as coordenadas de usinagem são recebidas pelo Scilab, traduzidas e transformadas em código G discretizado para (L_1, L_2 e L_3), captado pelo EMC2, que converte o código em sinais para os motores. Por final, a máquina efetua o procedimento desejado.

O PROJETO

3.1 Estágio inicial

A contribuição deste trabalho para o desenvolvimento da máquina fresadora tomou como partida o projeto ilustrado nas figuras 1 e 2. A situação inicial era uma estrutura com braços, guias, acoplamentos, fusos, porcas e “carrinhos” deslizantes devidamente construídos e montados. A fiação elétrica estava instalada e solta pelo robô. O controle da máquina também estava desenhado e projetado para a arquitetura paralela.



Figura 19: Vista superior do robô



Figura 20: Vista Frontal (Motor de passo, Fuso, braço+guia)

3.2 Metodologia do Projeto

Nesse projeto propõe-se, de uma forma resumida, realizar o aprimoramento e a conclusão de um protótipo de uma máquina fresadora de arquitetura paralela. Para tal, tem como principais necessidades a eliminação das folgas presentes nos braços do robô, a calibração da máquina, a reestruturação da parte elétrica e finalmente a usinagem controlada por Linux CNC.

O sucesso deste trabalho está diretamente ligado com a resolução dos problemas encontrados e os objetivos a serem atingidos. Esta máquina foi concebida para fins acadêmicos no intuito de analisar uma fresadora em arquitetura paralela com alta robustez e alta precisão. Para que essas características sejam alcançadas, foram definidas algumas metas como eliminação do erro no posicionamento da ferramenta, rigidez no acoplamento dos motores da ferramenta, eliminação das vibrações e otimização do volume de trabalho.

As necessidades do projeto são enumeradas abaixo com possíveis soluções. Estas serão descritas detalhadamente na seção 3.3.

3.3 Parte Mecânica

Para a eliminação das folgas, foram propostas algumas possíveis alternativas, que não são necessariamente exclusivas:

- Troca do acoplamento entre braços e suporte.
- Alteração das dimensões, do material, da estrutura do acoplamento e do encaixe.
- Construção e implantação de uma guia auxiliar, de forma que não prejudique o espaço de trabalho, fixa na base servindo de apoio para o braço.
- União dos braços no mesmo plano de forma a eliminar o apoio entre as extensões dos braços e então um momento maior nos braços de alturas inferiores.

3.4 Parte elétrica

A fiação elétrica é uma parte crítica do projeto; as conexões entre o computador e a PAMP (Placa Acionadora de Motor de Passo), construída pelo Professor Nilson em 2001, estavam se deteriorando, fios cortados, desencapados e soltos. A ligação entre essa placa e os motores também estavam na mesma situação, com conectividade duvidosa, assim como as próprias placas não funcionavam.

O trabalho nesta parte consiste em aprimorar a instalação elétrica de uma forma que melhore sua aparência e, principalmente, sua vida útil. Foi observado que quando os fios ficam expostos e soltos sem uma guia ou caminho e recipiente para serem guardados, o sistema tende a ficar menos estável, podendo até parar de funcionar.

3.5 Controle

Quanto à parte de controle, será feito o estudo dos parâmetros de entrada do sistema e serão modificados com os novos dados atribuídos às melhorias efetuadas.

4. ANÁLISE DAS SOLUÇÕES PROPOSTAS

4.1 Mecânica

Foram discutidas as ações viáveis a tomar visando a redução das folgas apresentadas pelo robô após analisar todas as soluções propostas no primeiro semestre. Destas, duas podem solucionar os problemas das folgas no mecanismo. As implementações citadas serão aqui desenvolvidas ilustrando os objetivos que serão alcançados, e posteriormente, validados através dos testes de deflexão com laser.

Deve-se ressaltar que a escolha das soluções se deu devido a dois principais fatores. O primeiro relacionado à viabilidade do projeto. Foram analisadas quais permitiriam menor alteração do projeto inicial e que tinham maior aproveitamento do trabalho já realizado em todos os sentidos, de aproveitamento desde o mecanismo até o material já disponível, enfocando um real melhoramento da estrutura já desenvolvida. O segundo fator relevante para escolha das implementações foi o ataque direto às causas das folgas. Procurou-se neste trabalho tomar as melhores medidas corretivas, mantendo as características do projeto. Tais causas foram verificadas diretamente nas juntas do robô, a partir de uma inspeção manual.

Assim, as seguir expõe-se as medidas que foram adotadas para sanar as folgas verificadas.

- Troca do acoplamento entre braço e suporte.

A troca do acoplamento será estudada pela utilização da combinação de rolamentos de carreiras esféricas radiais ou cônicos de forma a suportar a carga axial e prover movimentação satisfatória dos braços. Para a seleção dos rolamentos, os principais requisitos a serem avaliados para o nosso projeto, especificamente, serão o espaço possível para o rolamento, a intensidade e direção da carga, fixação na direção axial e disposição.

- Alteração das dimensões e do material do robô.

Foi percebido que uma das causas da folga existente na base do braço do robô é causada pelo momento do peso do próprio braço. Por conta disso, foi cogitada a reconstrução dos braços com Alumínio, porém quando a máquina estiver em operação, estará sob uma força no órgão terminal que será muitas vezes maior que o peso do braço.

A solução escolhida foi a de aumentar o comprimento dos braços do robô servindo como contra peso para compensar o momento causado pelo peso do próprio braço, visto na figura 21.

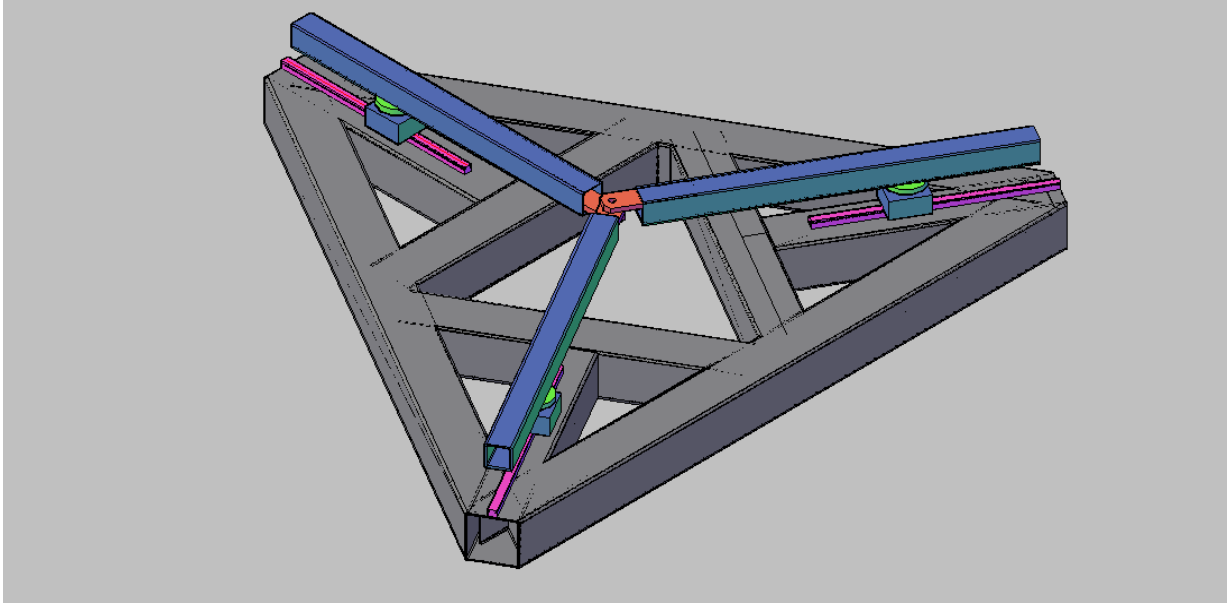


Figura 21: Representação dos contra-pesos nos braços

- Construção e implantação de uma guia auxiliar, de forma que não prejudique o espaço de trabalho, fixa na base servindo de apoio para o braço, visto na figura 22.

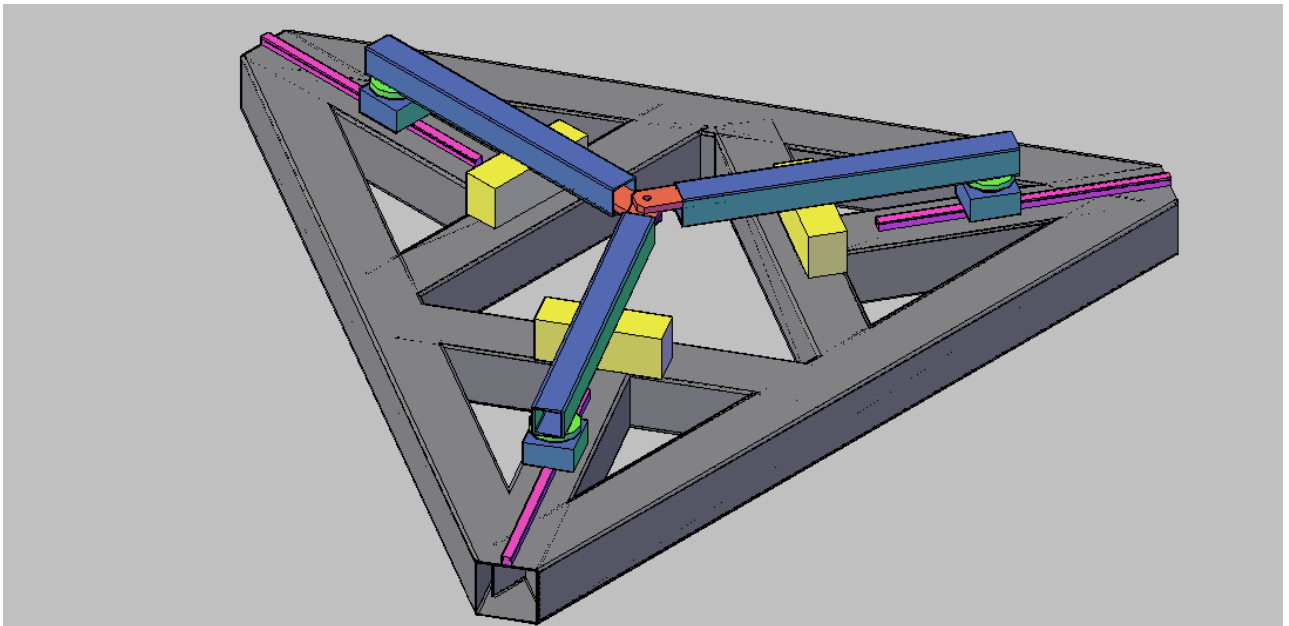


Figura 22: Utilização de uma Guia deslizante auxiliar

- União dos braços no mesmo plano de forma a eliminar o apoio entre as extensões dos braços e então um momento maior nos braços de alturas inferiores, visto na figura 23.

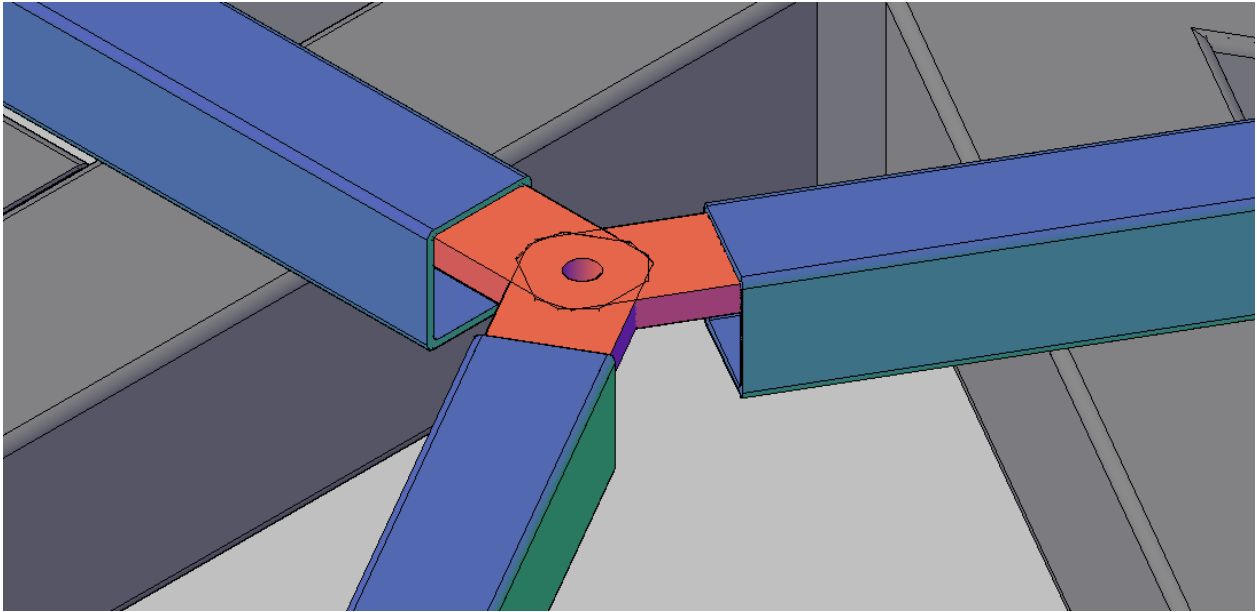


Figura 23: Proposta de fixação dos braços à ferramenta em um mesmo plano

- Substituição do acoplamento do eixo do braço ao cubo da guia

Troca do acoplamento com o rolamento existente por uma associação de rolamentos espaçados entre si, vista na figura 24.

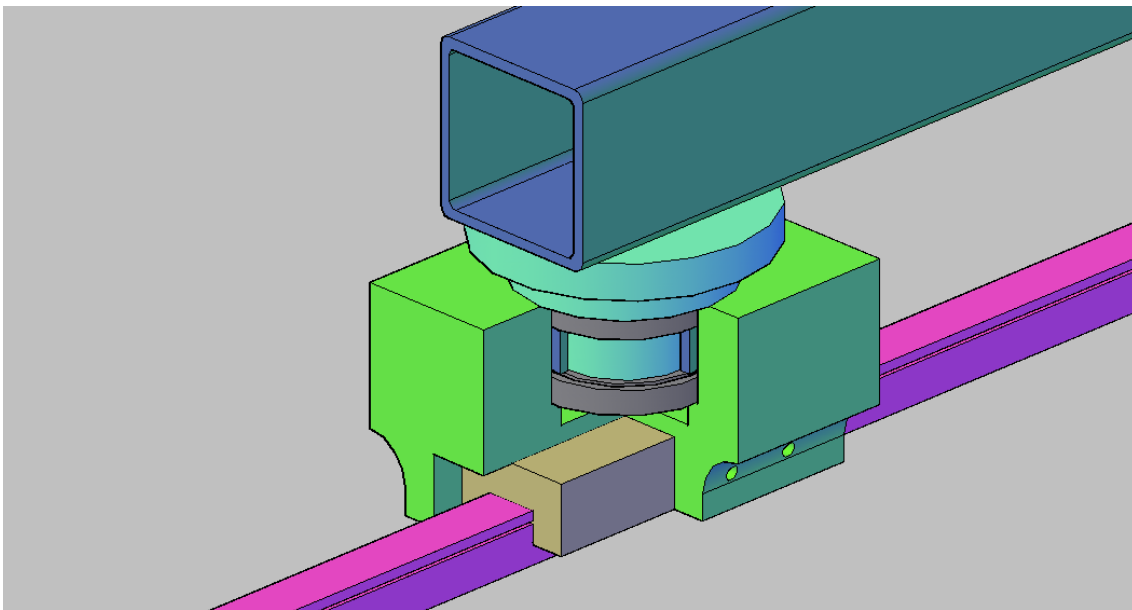


Figura 24: Acoplamento com rolamentos de esferas em pontos espaçados.

Para esta implementação, foi necessário projetar um novo cubo para o eixo dos braços no qual os novos rolamentos ficarão alojados. O novo cubo, além das novas dimensões para a nova montagem possui suas laterais estendidas para uma fixação nos carros das guias, de forma que garanta o paralelismo com o carrinho e evite folgas de encaixe.

A vista do cubo e a ilustração do rolamento utilizado estão na figura 25.

-Montagem Porca-fuso

Os problemas nas juntas porca-fuso são provocados pelo vão necessário entre elas para que haja deslizamento. Esta mínima folga se deve à tolerância de fabricação e, portanto, sempre vai existir. Com isso, visando utilizar o mesmo fuso, a implementação a ser feita aqui é simplesmente aumentar o comprimento da porca e da guia diminuindo assim o ângulo de folga em função desse aumento. A montagem é ilustrada na figura 34:

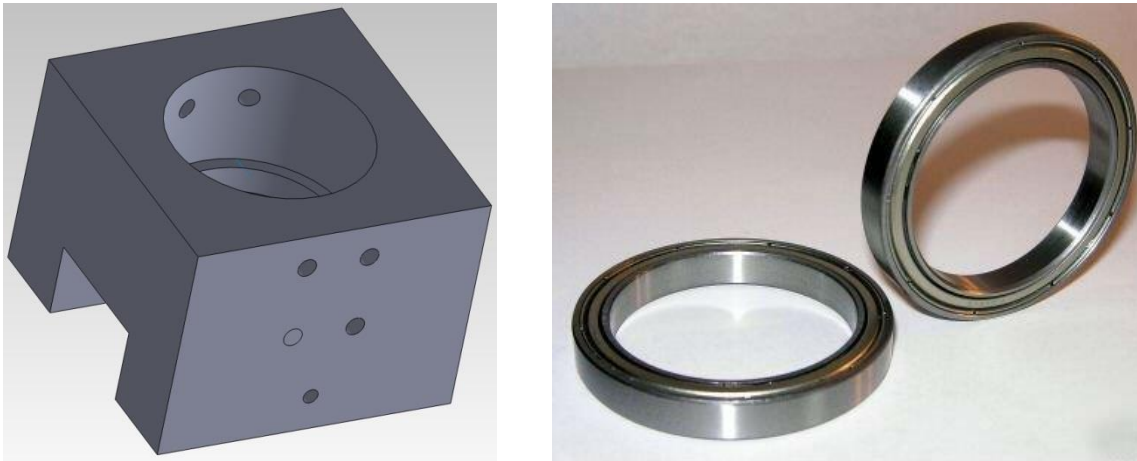


Figura 25: Cubo e rolamentos utilizados

Pode-se imaginar uma peça genérica de deslizamento linear como na figura 26:

Comprova-se facilmente pelos cálculos mostrados adiante, que o aumento do comprimento C , diminui o ângulo de deflexão α , ou seja, a folga existente.

Analisando a figura 27; por geometria simples, pode-se observar que mantendo-se constantes as dimensões B e b mostradas nos cálculos acima e variando o comprimento C (figura 28), tem-se uma significativa redução na deflexão α , sendo reduzido à quase a metade dobrando-se o comprimento da porca como mostrado nas equações em seguida.

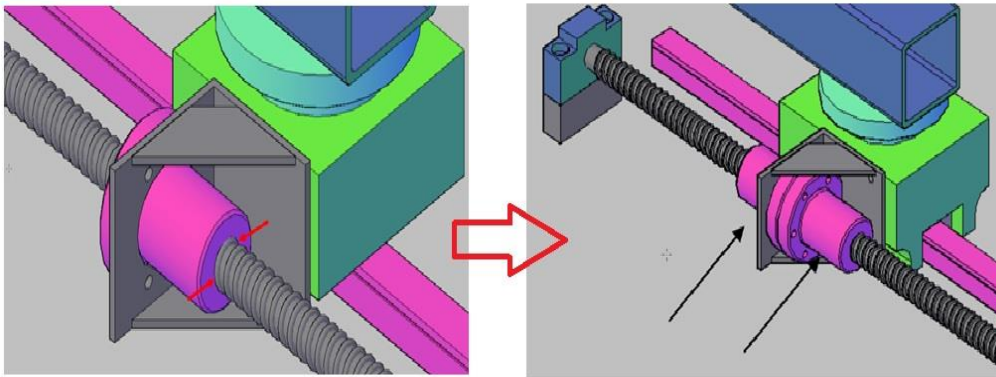


Figura 26: Associação de porcas no fuso

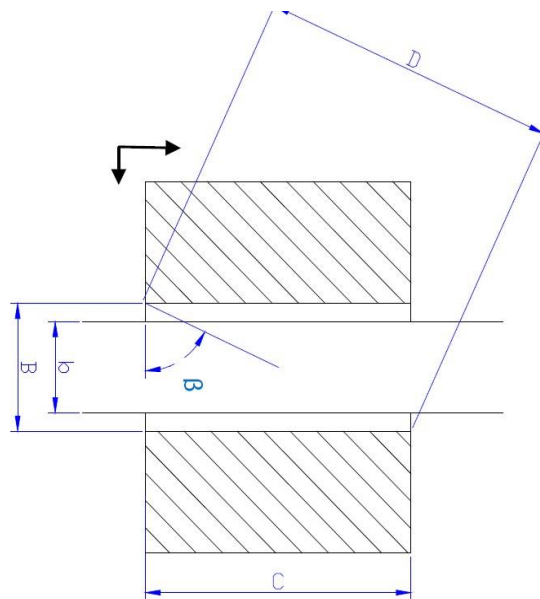


Figura 27: Esquema da porca no fuso sem deflexão

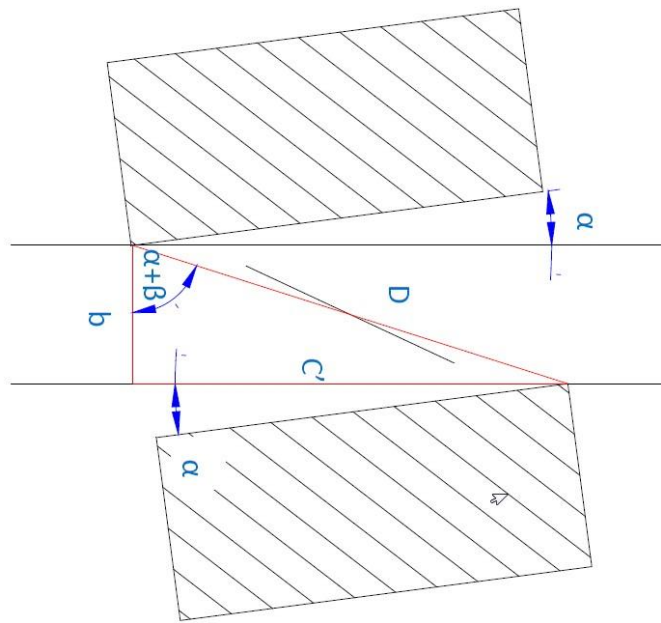


Figura 28: Esquema da porca do fuso com deflexão máxima

$$D = \sqrt{B^2 + C^2} \quad \beta = \tan^{-1} \left(\frac{C}{B} \right)$$

$$b = D \cos(\alpha + \beta)$$

$$b = \sqrt{B^2 + C^2} \cos(\alpha + \beta) \quad C = 45 \text{ mm} \quad \alpha = 1,27^\circ$$

$$C = 90 \text{ mm} \quad \alpha = 0,636^\circ$$

$$\cos^{-1} \left(\frac{b}{\sqrt{B^2 + C^2}} \right) - \tan^{-1} \left(\frac{C}{B} \right) = \alpha$$

4.2 Parte Elétrica

Para descobrir quais componentes não estavam funcionando, foram propostas algumas soluções:

- Substituição do cabo paralelo que liga o computador com a placa PAMP.
- Substituição dos fios que conectam a placa PAMP e os motores de passo.
- Substituição da placa PAMP por outro driver.

- Substituição da placa paralela do computador.

4.3 Controle

O problema inicial, para que se pudesse fazer os testes de controle, era fazer o sistema todo voltar a ser operante. Foi cogitado que o programa não estava enviando os sinais corretamente, por isso algumas soluções foram propostas.

- Testar com outro computador.
- Atualizar o EMC2 do computador atual.
- Atualizar o sistema operacional Ubuntu do computador atual.

Além disso, no programa EMC2, é necessário configurar virtualmente a máquina: limites físicos, tamanho de passo, posição do home. Assim como é necessária a escolha do driver utilizado e seu microstepping, no caso de substituição de drivers.

5. SOLUÇÕES ADOTADAS

5.1 Parte Mecânica

5.1.1 Acoplamento

A partir das possíveis soluções a serem implementadas, foi decidido por aquela que melhor se mostrou aplicável, segundo fatores como custo, tempo e gravidade do problema. Uma vez que a principal fonte de folga verificada foi no acoplamento, um novo tipo de acoplamento projetado, como mostrado anteriormente, foi usinado em alumínio.

O cubo do eixo, como se pode ver nas figuras abaixo, possui um alojamento mais profundo que o do projeto inicial de maneira que se possa fazer a montagem proposta utilizando dois rolamentos. O alojamento foi dimensionado com tolerância M6 para ajuste forçado furo-base segundo catálogo da NSK. O rolamento utilizado foi o 6808ZZ da NSK.

Como apresentado anteriormente tal solução propõe, além de uma interferência rolamento-alojamento mais eficiente que a anterior, o apoio do eixo em dois pontos de contato espaçados entre si a fim de minimizar o balanço dos braços devido ao peso e outras cargas verticais.

São mostradas nas figuras dos desenhos em CAD e de fabricação do cubo usinado para o novo acoplamento:

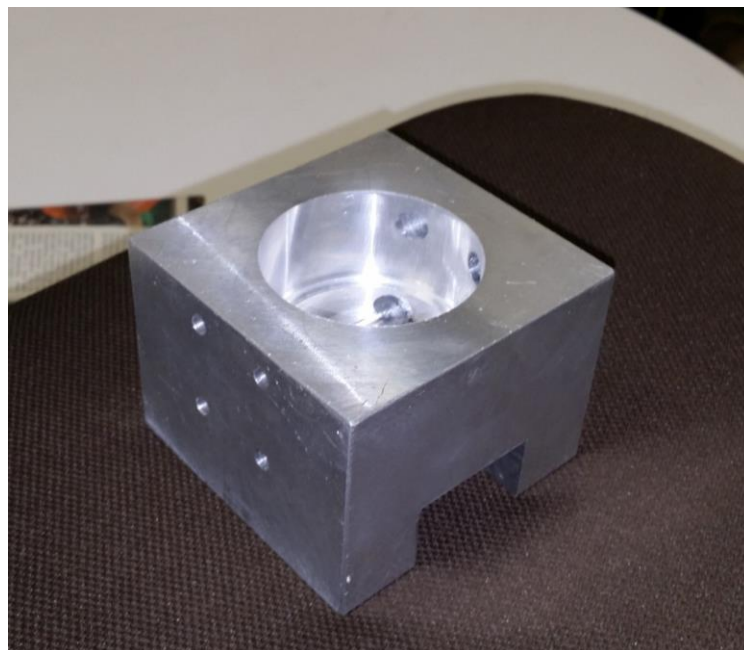


Figura 19: Cubo do eixo usinado

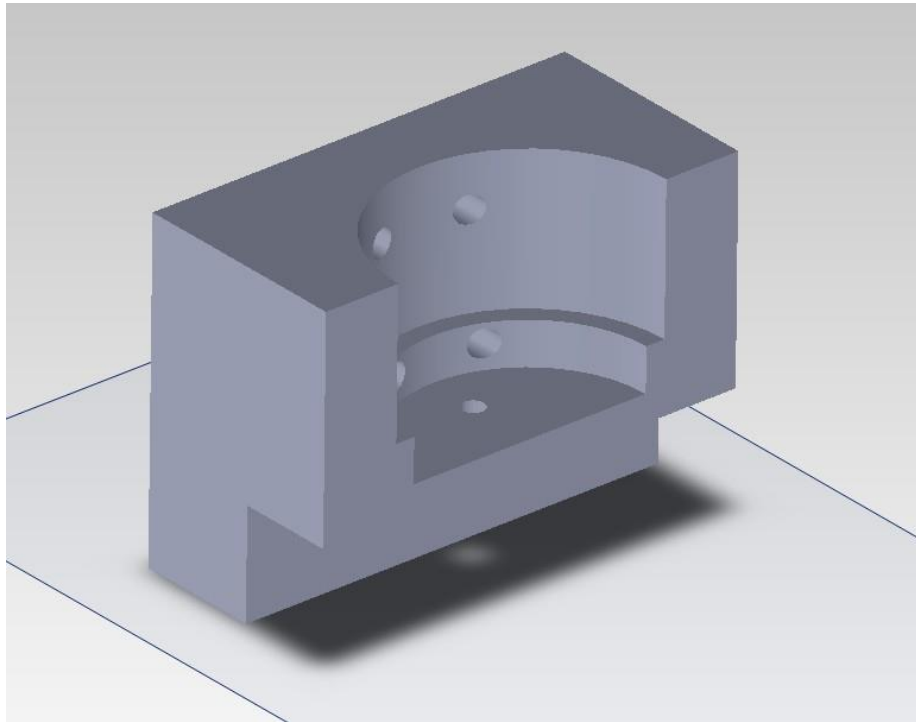


Figura 20: Vista em corte do alojamento dos rolamentos no cubo

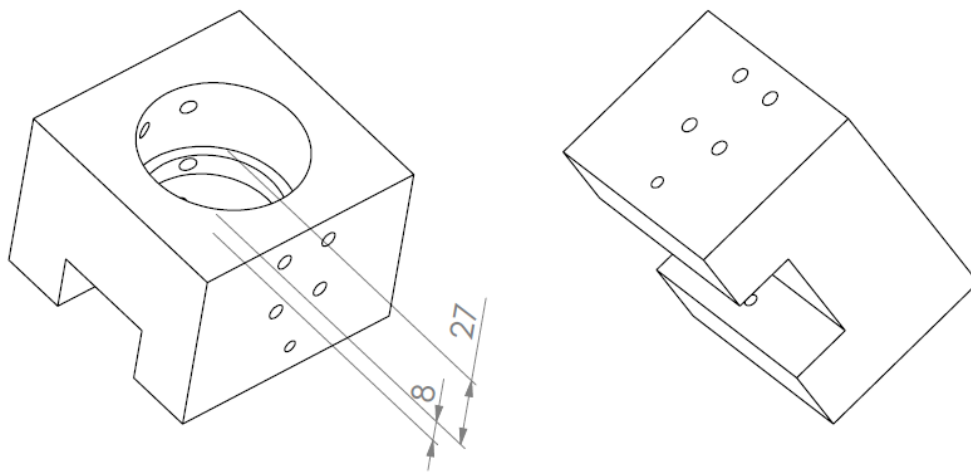


Figura 31: Vistas 3D do desenho de fabricação

5.1.2 Espaçador

Na montagem proposta, com o intuito de se manter os dois rolamentos de cada eixo o mais espaçados possível, sentiu-se a necessidade de se utilizar um “espaçador”. Este tem a única

função de manter os rolamentos em suas posições fixas após o conjunto montado no eixo ser encaixado no alojamento do cubo apoiando-se no rebaixo do mesmo.

O espaçador mostrado abaixo foi usinado em alumínio com o cuidado de que não tocasse as pistas dos rolamentos e assim interferisse na rotação do eixo. Figuras 32 e 33.

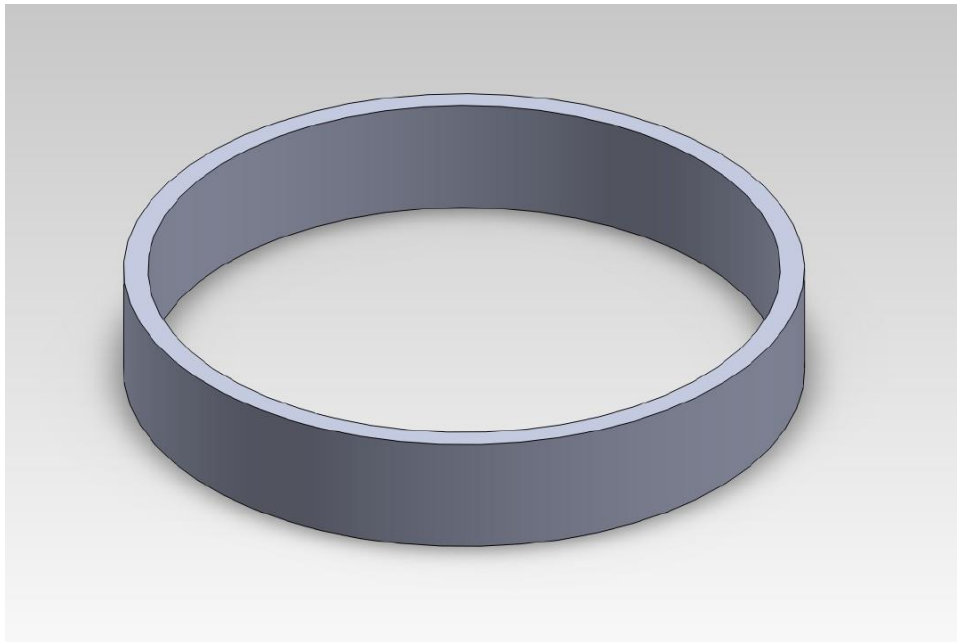


Figura 32: Vista 3D do espaçador



Figura 33: Espaçador usinado em alumínio

5.1.3 Montagem

Para a remontagem do robô, figuras 34 e 35, foram tomados os seguintes procedimentos:

- Furos adicionais foram feitos nos cubos para se ter acesso com as ferramentas aos parafusos;
- Os cubos foram primeiramente fixados às guias lineares e às porcas dos fusos de esferas, como mostrado na figura abaixo;
- Fixados os cubos, os eixos com os rolamentos foram encaixados até o rolamento inferior atingir o rebaixo do acoplamento e o espaçador se firmar entre os rolamentos;
- Finalmente, os braços foram fixados aos eixos finalizando a nova montagem.



Figura 34: Montagem do novo acoplamento. Guia, fuso e eixo do braço



Figura 35: Braços montados e área de trabalho

5.2 Parte elétrica

Devido ao limite de tempo de entrega do projeto e a busca pela melhora dos componentes elétricos, quase todas as soluções propostas de substituição foram aplicadas, com exceção da placa paralela do computador.

- Substituição do cabo paralelo

O cabo paralelo foi trocado e o modelo de conexão foi alterado.

A configuração antiga era o cabo paralelo cortado e todos fios soltos ficavam conectados diretamente com a placa PAMP (figura 36).

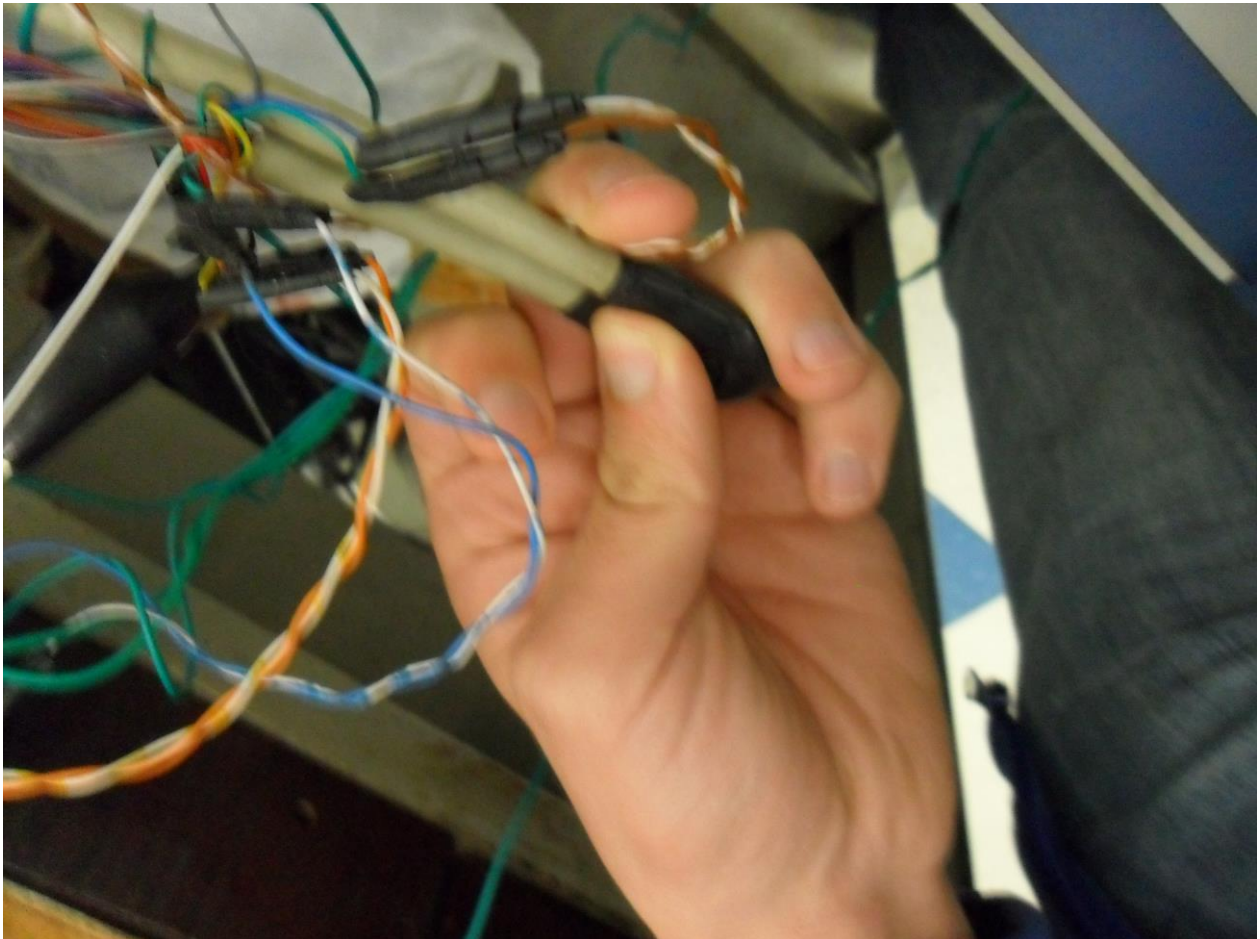


Figura 36: Configuração antiga

A nova configuração é o cabo (figura 37) conectado com o computador e com a caixa alojamento (figura 38) dos drivers de motor de passo.



Figura 37: Cabo paralelo macho-macho

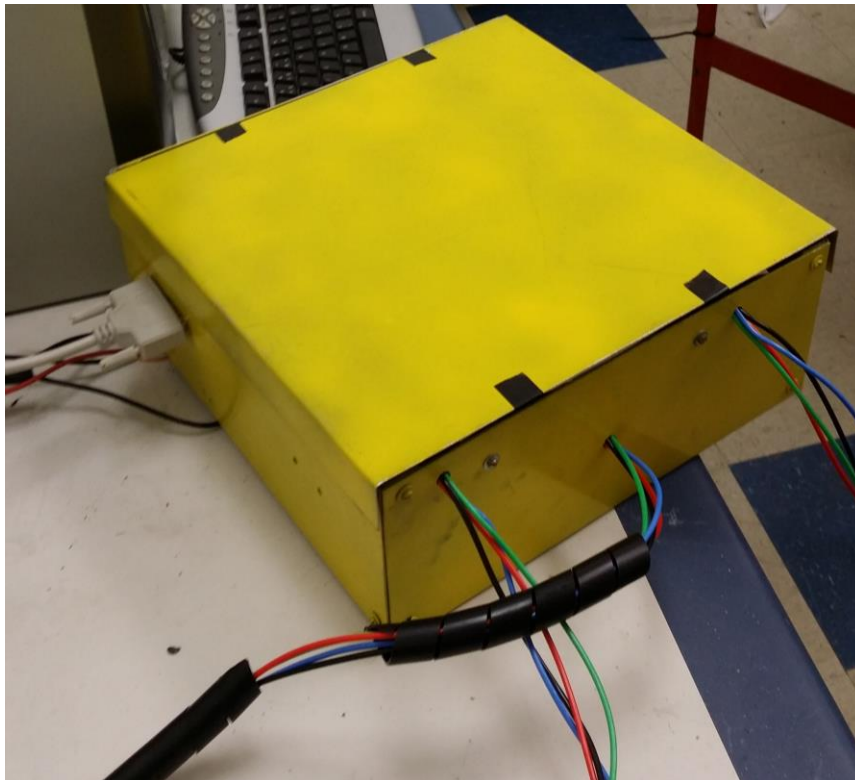


Figura 38: Caixa alojamento dos drivers

- Substituição da placa PAMP por outro driver.

A placa PAMP, por se mostrar instável e antiga, foi substituída por um novo driver de motor de passo, o Gecko G251 (figura 39).

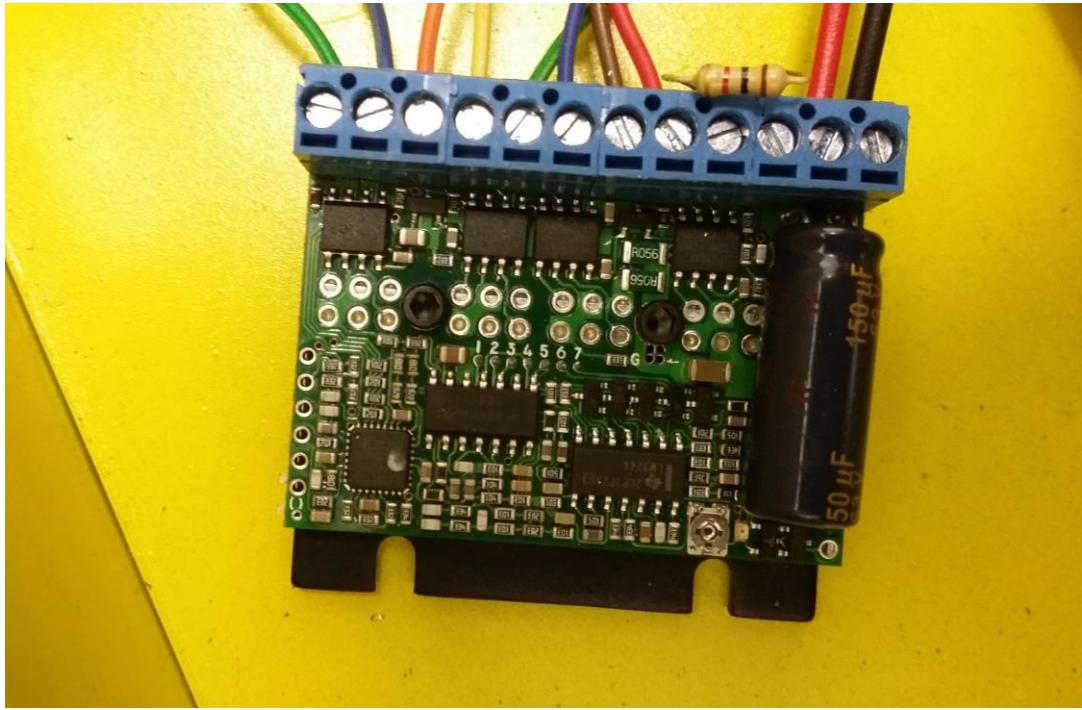


Figura 39: Driver Gecko G251

- Substituição dos fios que conectam a placa PAMP e os motores de passo.

Os fios que ligavam o driver com os motores de passo também foram substituídos, figura 40.

O conector DB25 (cabo paralelo) da caixa alojamento é um conector fêmea ilustrado na figura 41 com os pinos descritos na tabela 4.

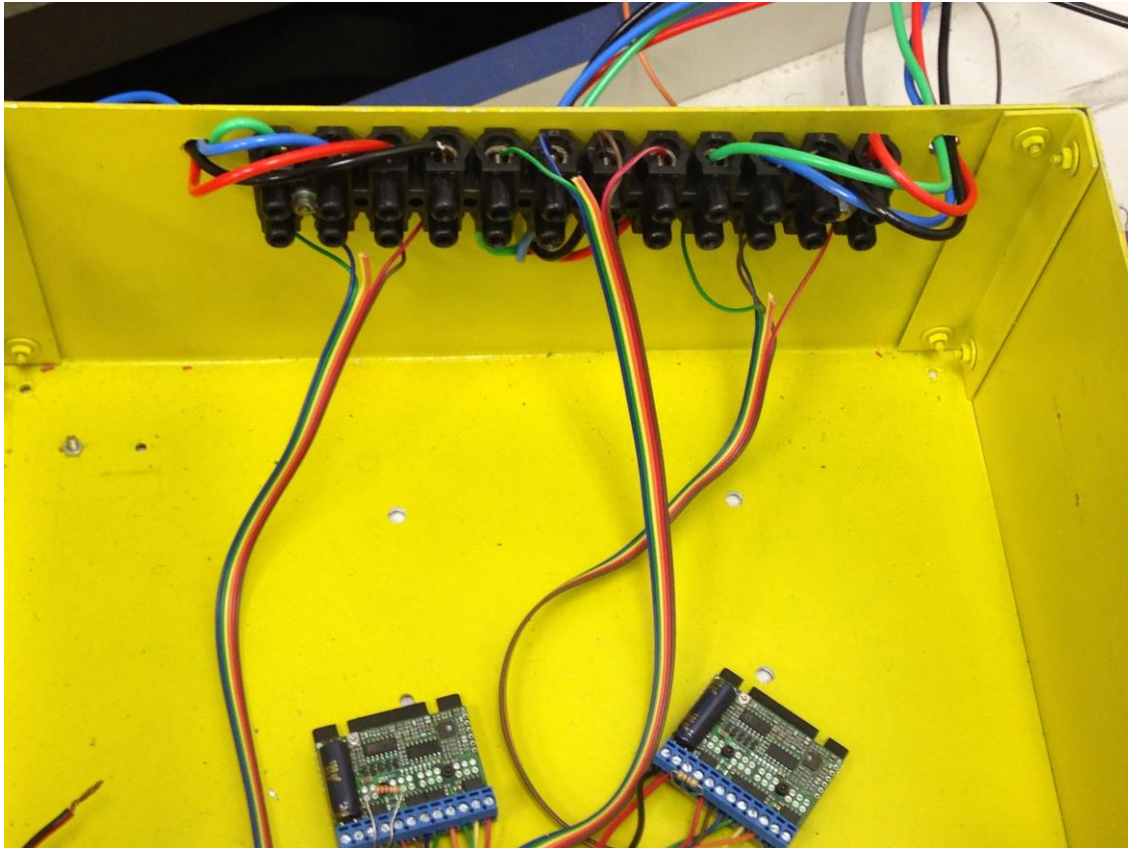


Figura 40: Drivers Gecko e distribuição de fios

Tabela 4 : Pinos do conector fêmea

Pinos	Cores	Funções
1	Marrom	<i>Enable</i>
2	Laranja	<i>Step</i> (motor L1)
3	Amarelo	<i>Direction</i> (motor L1)
4	Vermelho	<i>Step</i> (motor L2)
5	Azul	<i>Direction</i> (motor L2)
6	Verde	<i>Step</i> (motor L3)
7	Laranja	<i>Direction</i> (motor L2)
25	Amarelo e Verde	Terra

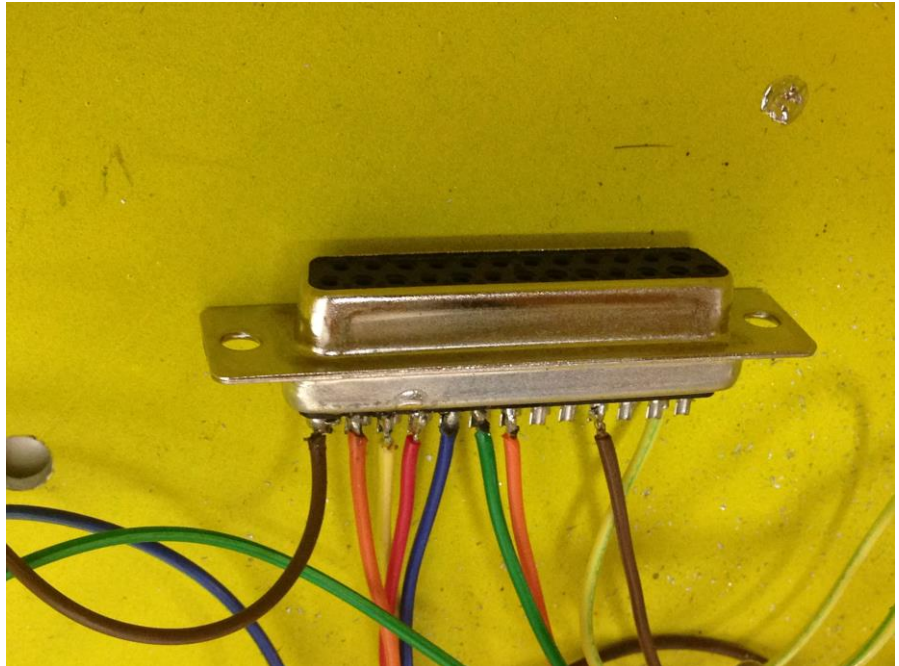


Figura 41: Conector DB25 fêmea

5.3 Controle

Depois de efetuadas todas as alterações na parte elétrica, somente foi necessário modificar as configurações no EMC2 referentes ao novo driver.

6. TESTES E DISCUSSÕES DE RESULTADOS

Após a montagem completa do protótipo com o novo acoplamento foram feitos os ensaios de deflexão para se observar tanto a precisão de um braço em movimento como a influência do braço redundante com a nova montagem dos eixos.

Foram feitos dois ensaios diferentes, estático nas direções de maior e menor rigidez, e dinâmico, verificando a precisão de movimento. Para ambos os casos, foi utilizado um laser para a medição e cargas foram aplicadas por meio de corda e polia. A saída do laser foi medida por osciloscópio verificando-se as deflexões ocasionadas.

A seguir são expostos os resultados das medições e a análise das deflexões para validação do trabalho e posterior conclusão das medidas tomadas. As figuras abaixo exibem a montagem feita para a tomada das deflexões, na qual as cargas foram aplicadas por meio de corda e polia, e o laser apontado no mesmo sentido da força aplicada. Na figura as setas indicam o local de aplicação da força e os pesos utilizados.

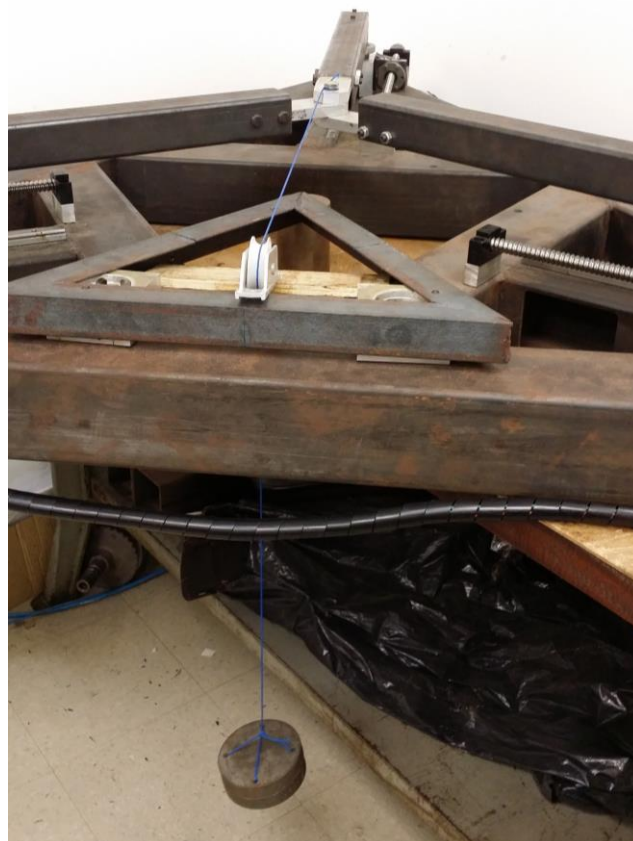


Figura 42: Montagem para aplicação de carga nos braços

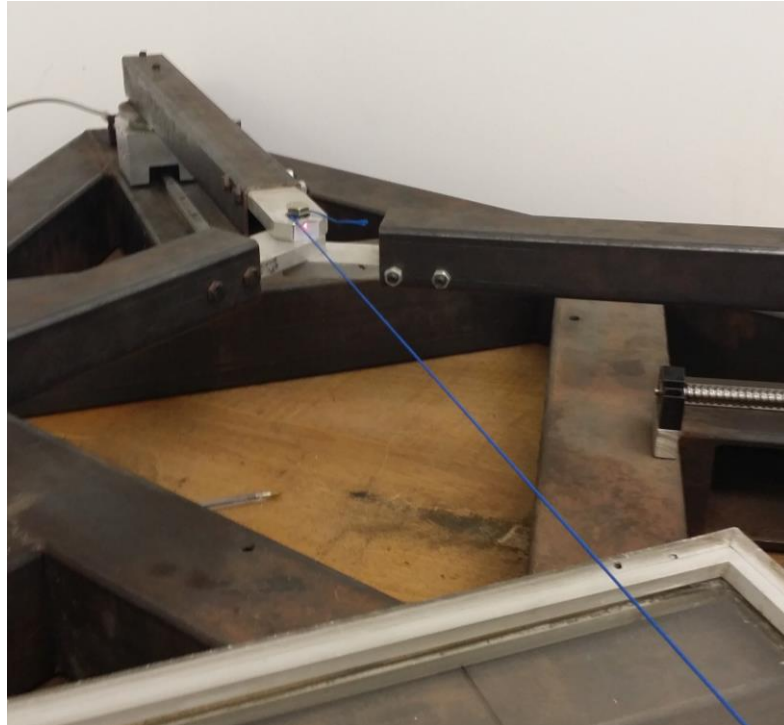


Figura 43: Montagem de três braços com laser e força aplicada na extremidade do braço

Medição estática

Como maneira de comprovar a influência de um terceiro braço com os novos acoplamentos, foi realizada a medição a laser com 2 e 3 braços montados, longitudinalmente e perpendicular a um braço. Para todas as medições foram analisados três casos: sem carga, 20.8N e 41.4N. As curvas obtidas e o valor da deflexão são expostos a seguir.

Deflexão longitudinal (maior rigidez):

Para esta configuração, curvas satisfatórias foram obtidas com o laser ajustado para uma variação de deslocamento de $40\mu\text{m/V}$ e de velocidade de 10mm/s/V , como visto na figura.

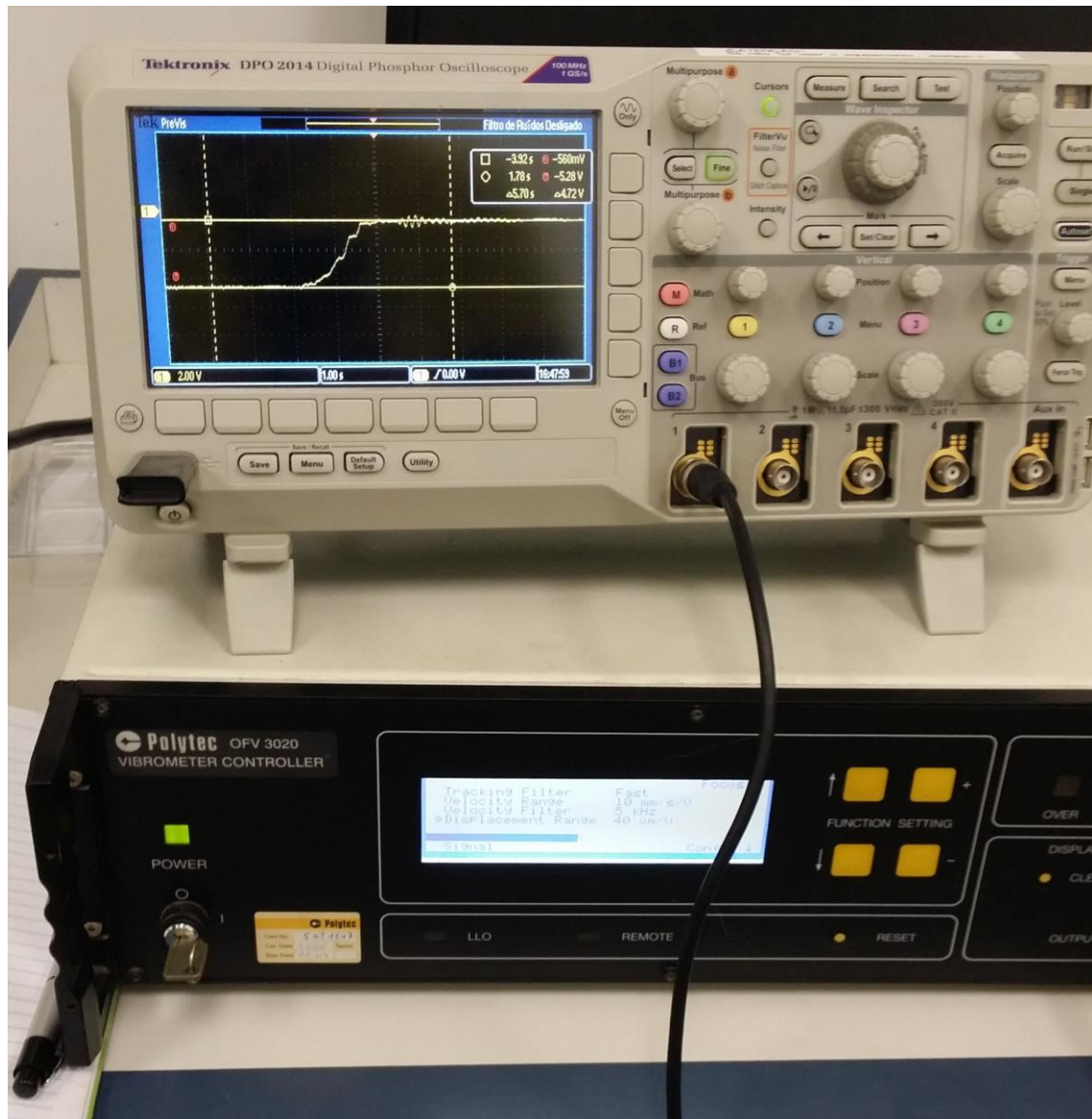


Figura 44: Controlador do laser e osciloscópio

Deflexão longitudinal (maior rigidez):

Para esta configuração, curvas satisfatórias foram obtidas com o laser ajustado para uma variação de deslocamento de $40\mu\text{m}/\text{V}$ e de velocidade de $10\text{mm}/\text{s}/\text{V}$, como visto na figura.

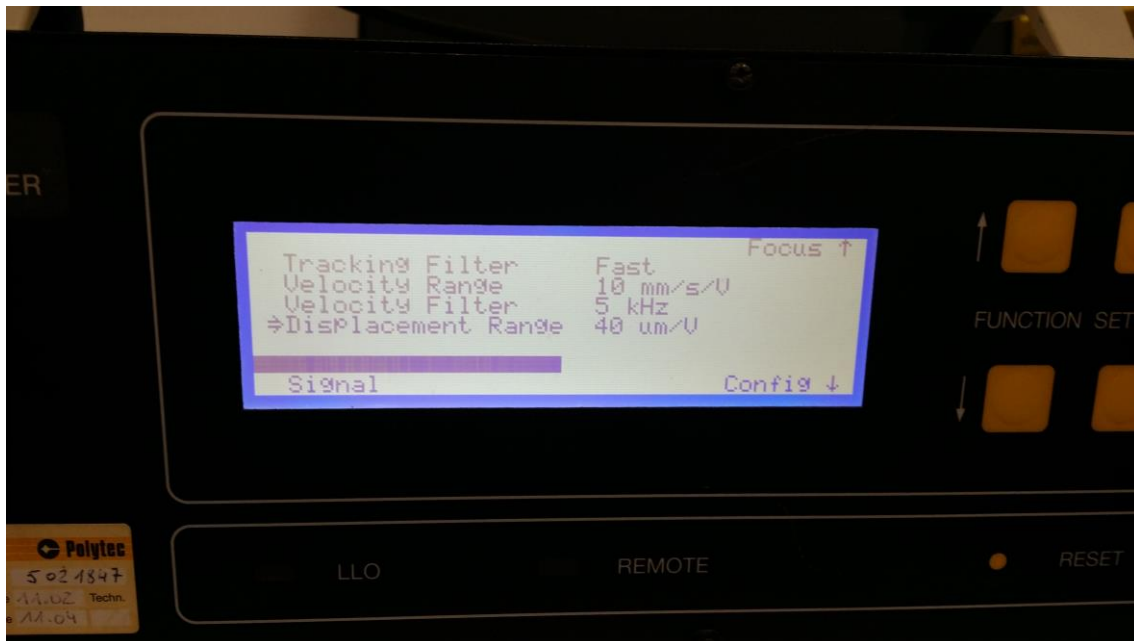


Figura 45: Configuração do laser(1)

- Carga de 20.8N

2 braços:

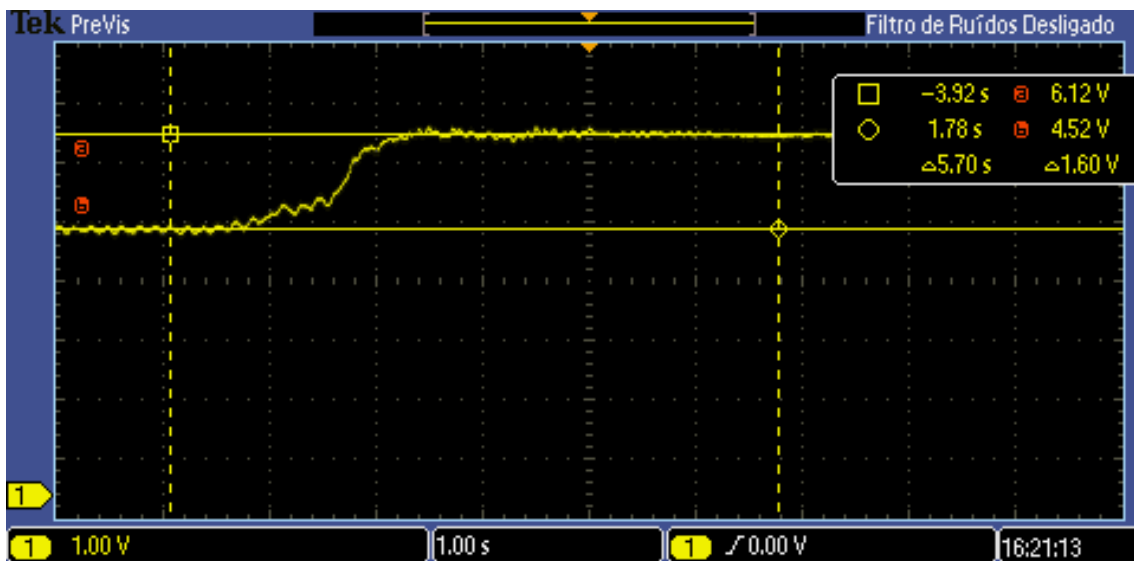


Figura 46: Curva de deflexão no osciloscópio(1)

3 braços:

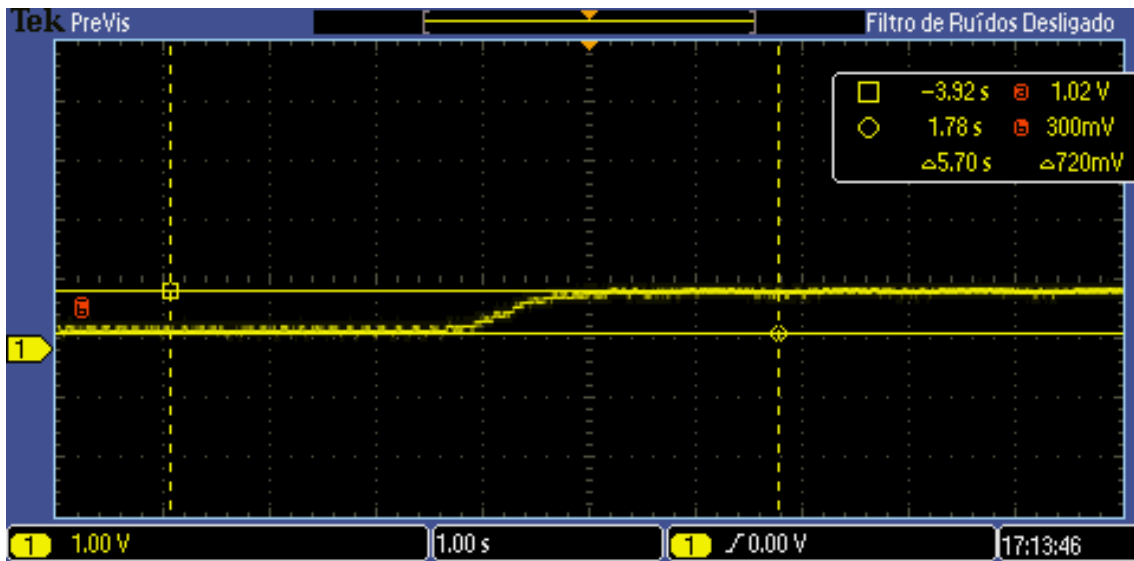


Figura 47: Curva de deflexão no osciloscópio(2)

Para situação de dois braços a deflexão calculada foi de 0.064mm enquanto na montagem de três braços o valor foi de 0.0288mm.

- Carga de 41.4N

2 braços:

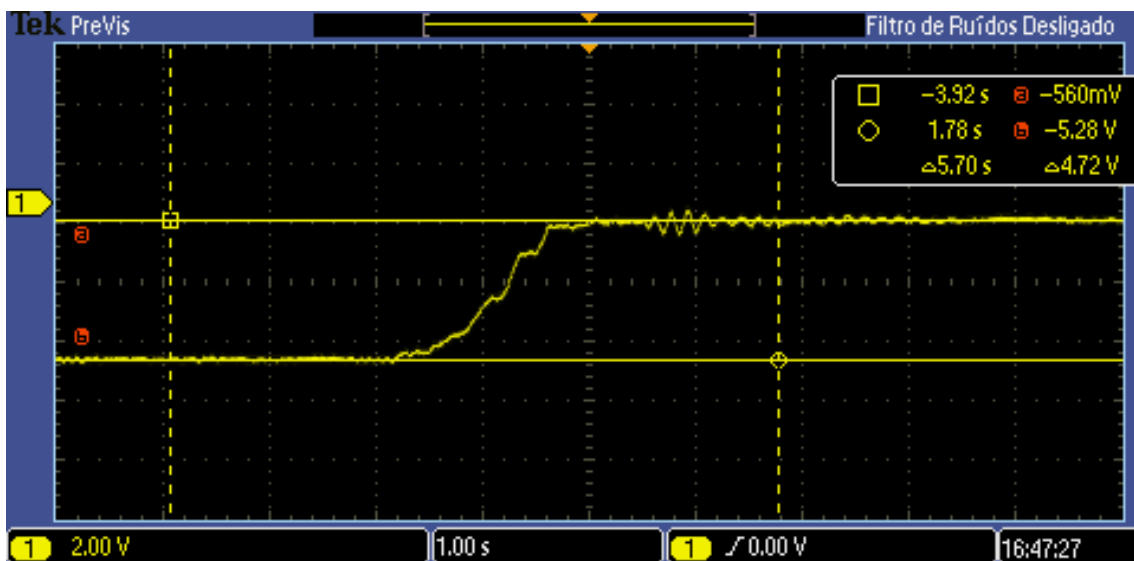


Figura 48: Curva de deflexão no osciloscópio(3)

3 braços:

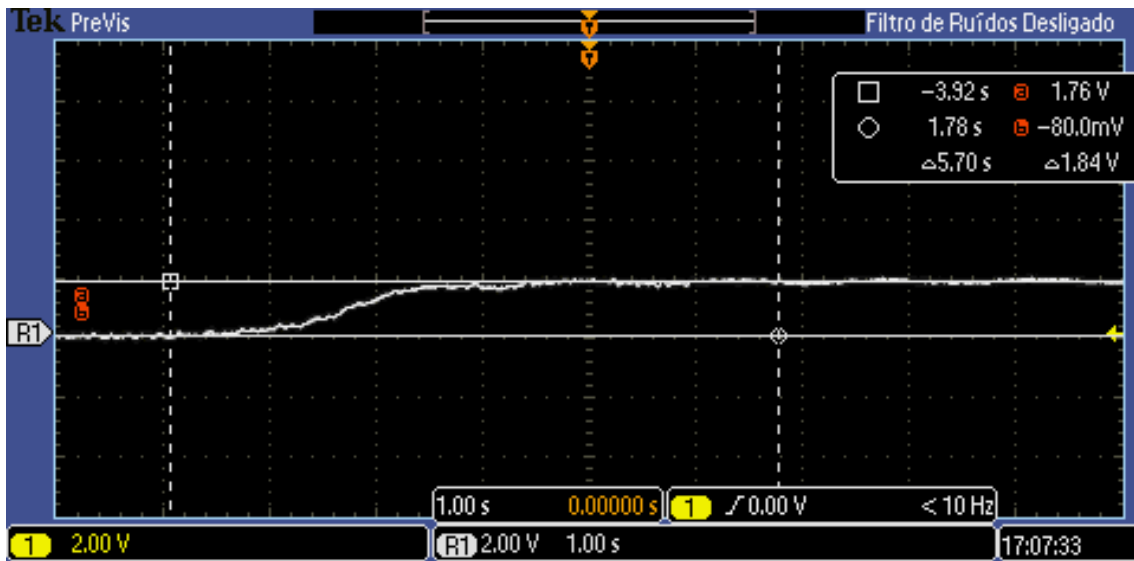


Figura 49: Curva de deflexão no osciloscópio(4)

Para situação de dois braços a deflexão calculada foi de 0.1888mm enquanto na montagem de três braços o valor foi de 0.0736mm.

Em ambos os casos, a montagem utilizando-se o braço redundante apresentou significativa melhora no valor defletido.

Deflexão para carga perpendicular ao braço (menor rigidez):

Para esta configuração, curvas satisfatórias foram obtidas com o laser ajustado para uma variação de deslocamento de $160\mu\text{m/V}$ e de velocidade de 10mm/s/V , como visto na figura 51.

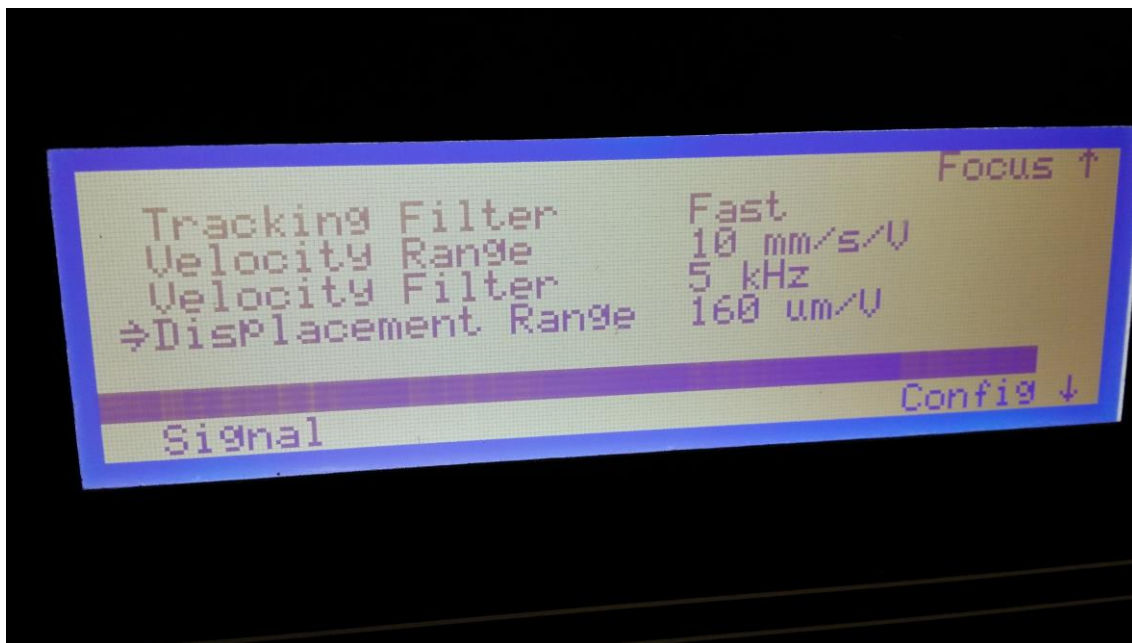


Figura 50: Configuração do laser(2)

- Carga de 20.8N

2 braços:

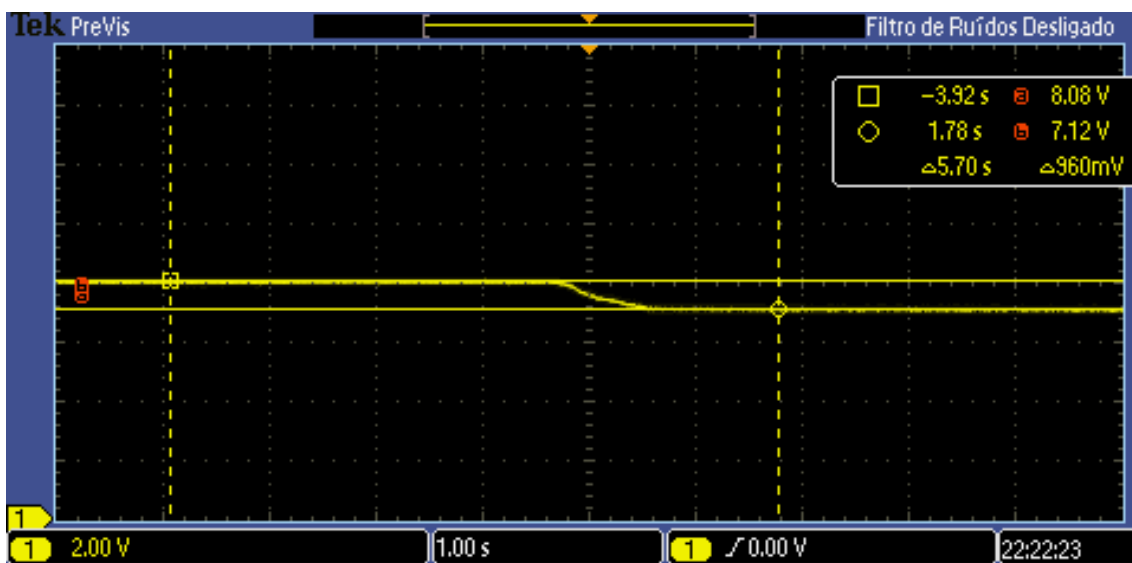


Figura 51: Curva de deflexão no osciloscópio(5)

3 braços:

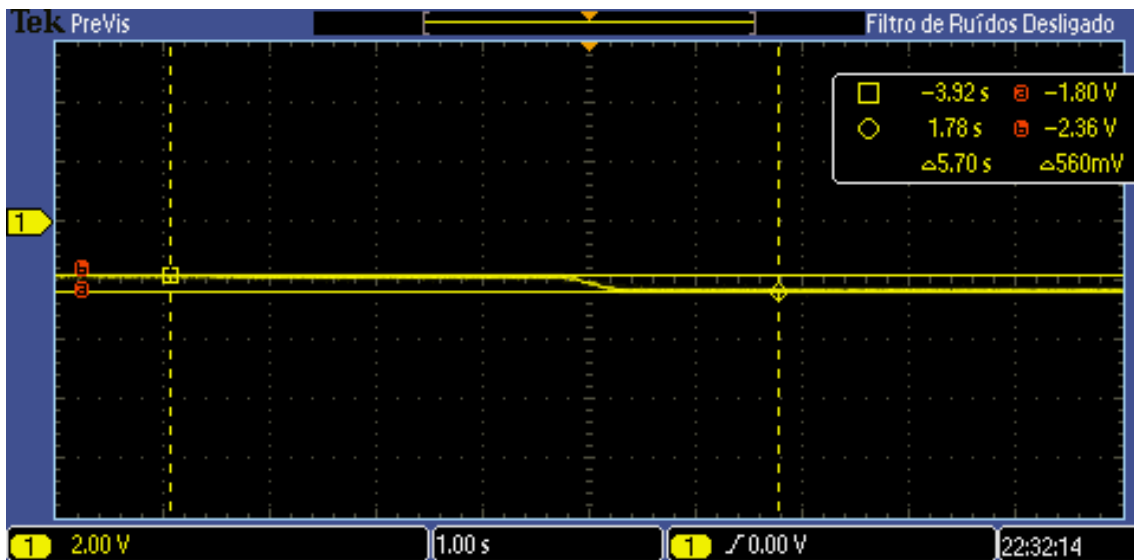


Figura 52: Curva de deflexão no osciloscópio(6)

Para situação de dois braços a deflexão calculada foi de 0.1536mm enquanto na montagem de três braços o valor foi de 0.0896mm.

- Carga de 41.4N

2 braços:

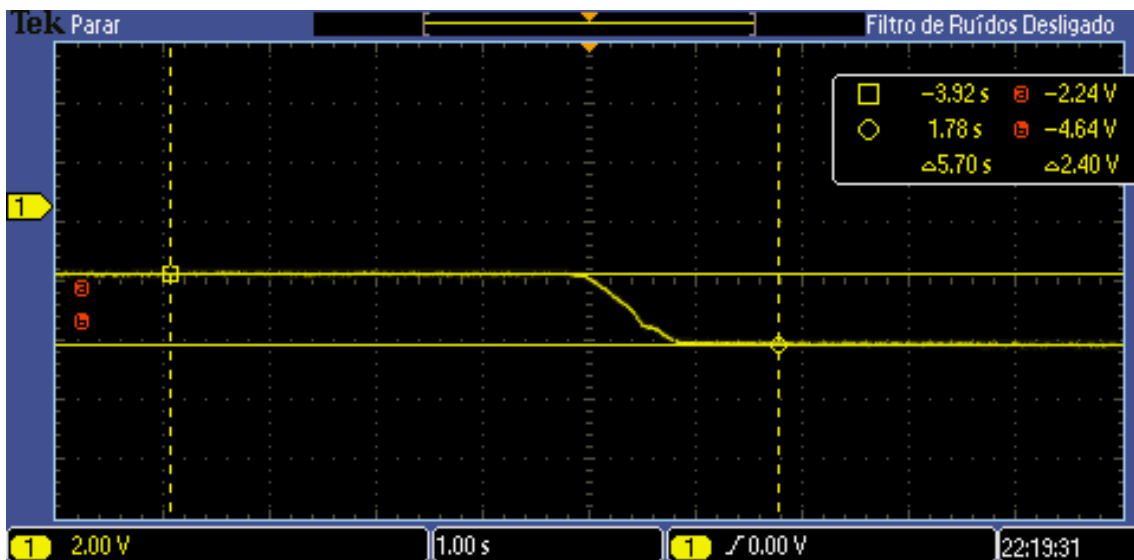


Figura 53: Curva de deflexão no osciloscópio(7)

3 braços:

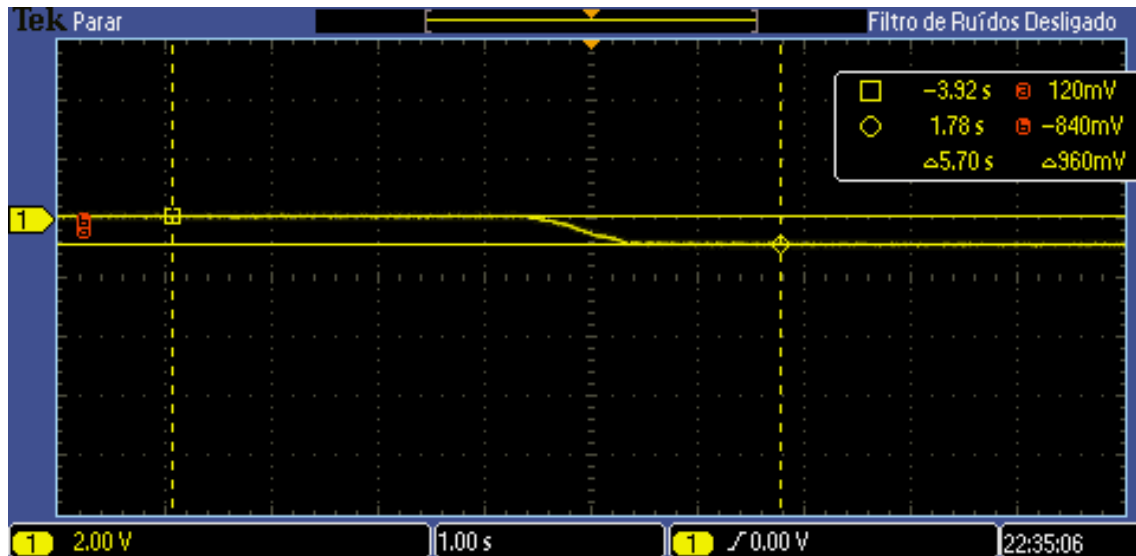


Figura 54: Curva de deflexão no osciloscópio(8)

Para situação de dois braços a deflexão calculada foi de 0.384mm enquanto na montagem de três braços o valor foi de 0.1536mm.

Assim como no sentido de maior rigidez, em ambos os casos, a montagem utilizando-se o braço redundante apresentou significativa melhora no valor defletido.

Medição Dinâmica

Para testar a precisão do movimento de um braço submetido à cargas, foi medida a deflexão na extremidade de um braço respondendo aos comandos de se deslocar 5mm, 1mm e 0,5mm. Os ensaios foram realizados para três casos: sem carga, 20.8N e 41.4N, aplicados no sentido contrario ao movimento.

Para esta configuração, como mostrado na figura 56, curvas satisfatórias foram obtidas com o laser ajustado para uma variação de deslocamento de 640 μ m/V e de velocidade de 50mm/s/V.

Em seguida, são apresentados os resultados dos ensaios em cada caso.

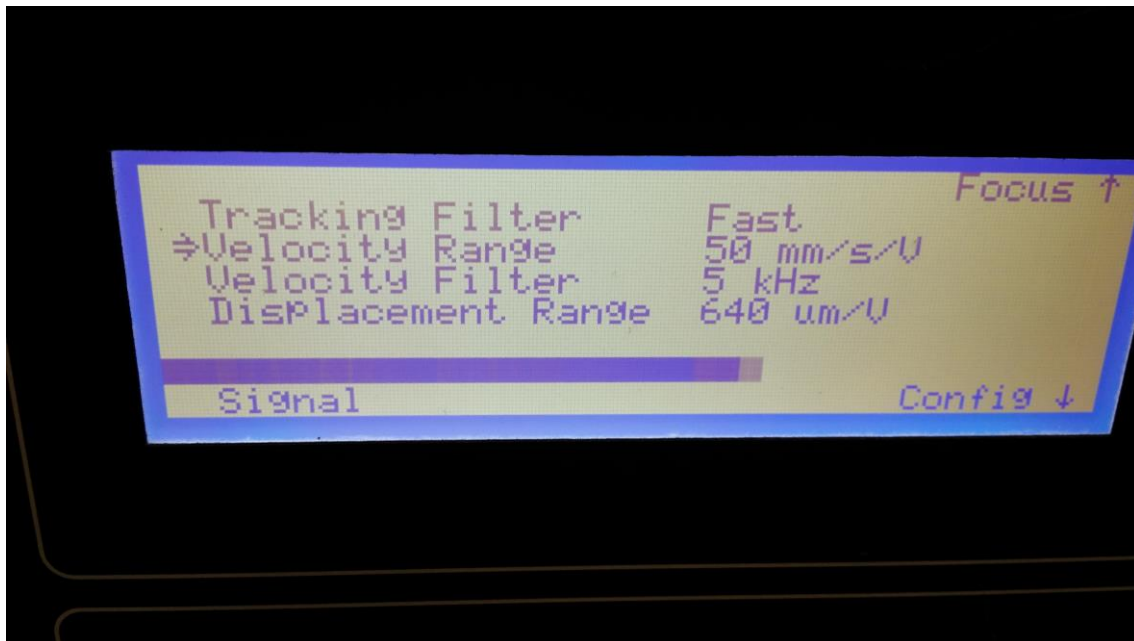


Figura 55: Configuração do laser(3)

- Avanço de 5mm

Sem carga:

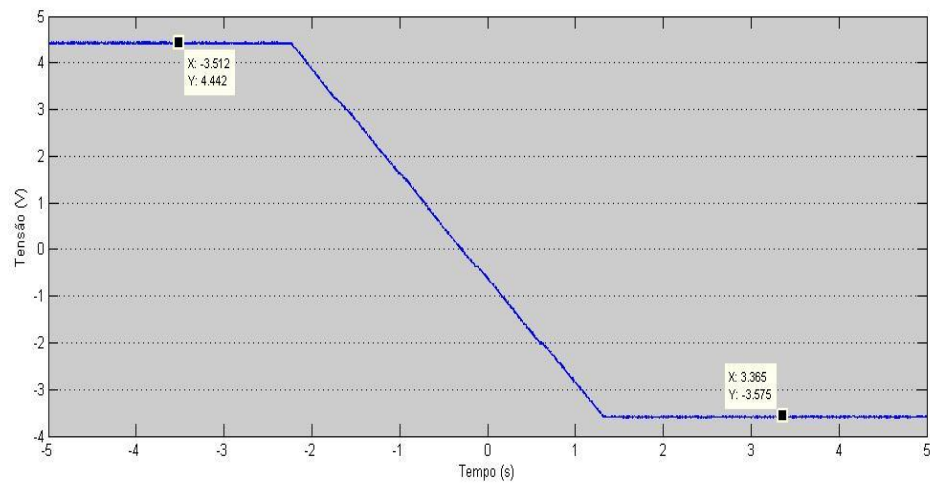


Figura 56: Curva de deflexão no Matlab(1)

Avanço real: 5.13mm

Carga de 20.8N:

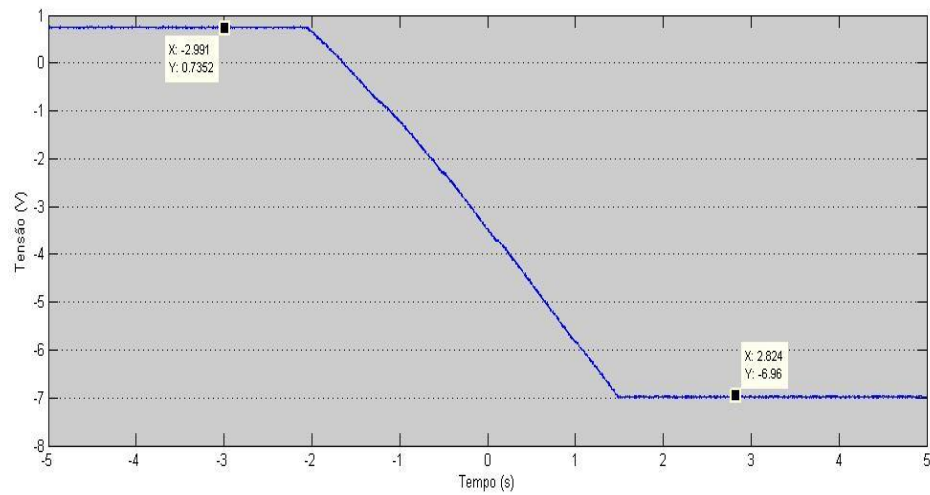


Figura 57: Curva de deflexão no Matlab(2)

Avanço real: 4.92mm

Carga de 41.4N

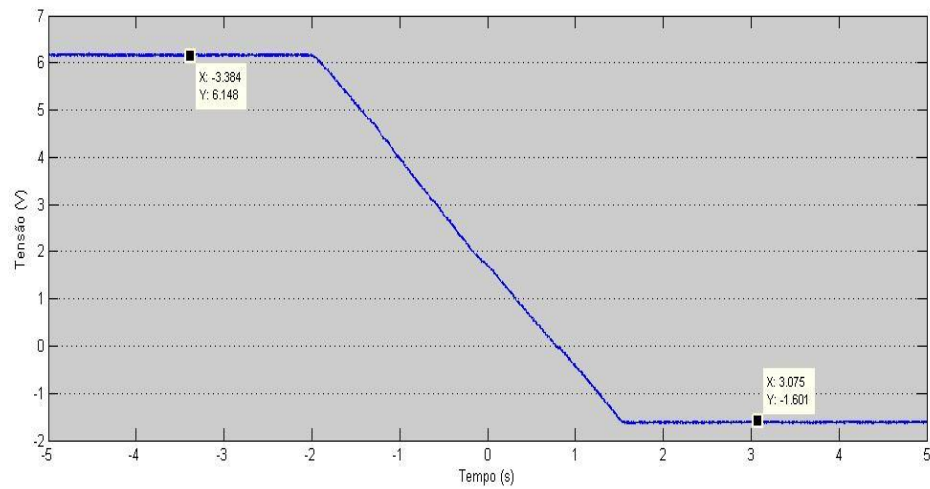


Figura 58: Curva de deflexão no Matlab(3)

Avanço real: 4.95mm

-Avanço de 1mm

Sem carga:

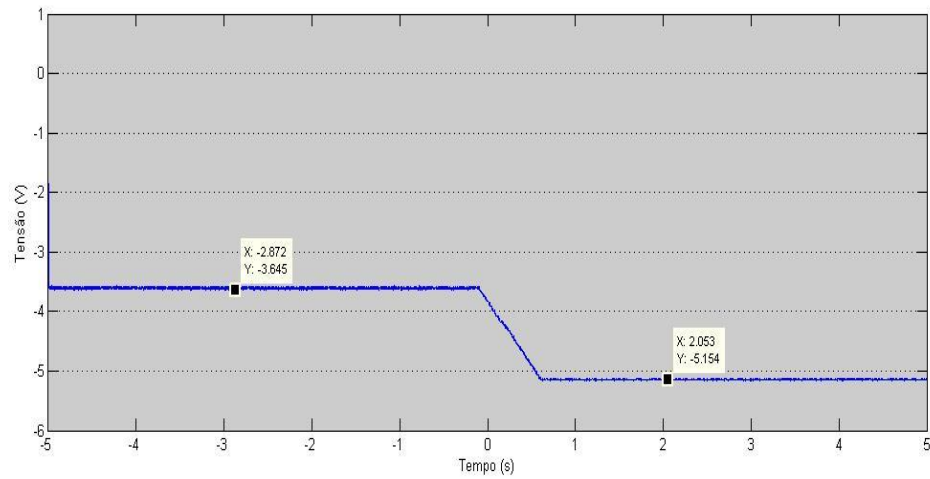


Figura 59: Curva de deflexão no Matlab(4)

Avanço real: 0.97mm

Carga de 20.8N:

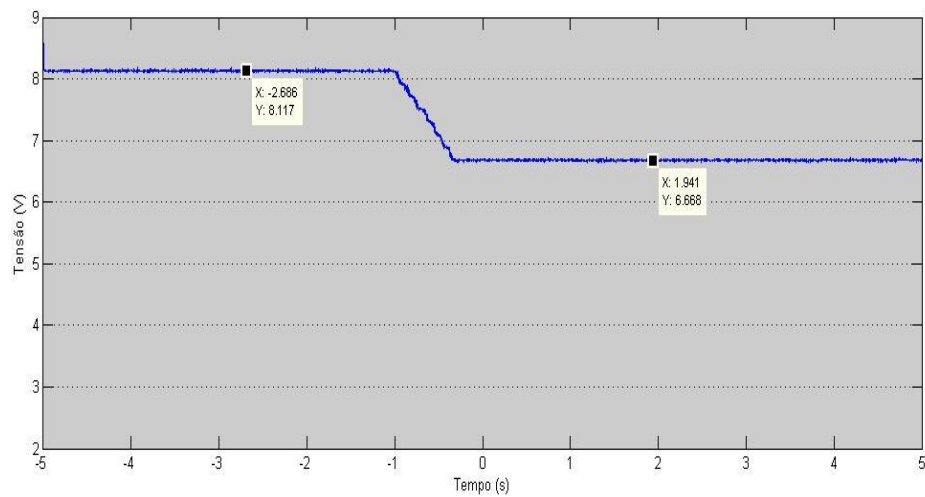


Figura 60: Curva de deflexão no Matlab(5)

Avanço real: 0.91mm

Carga de 41.4N:

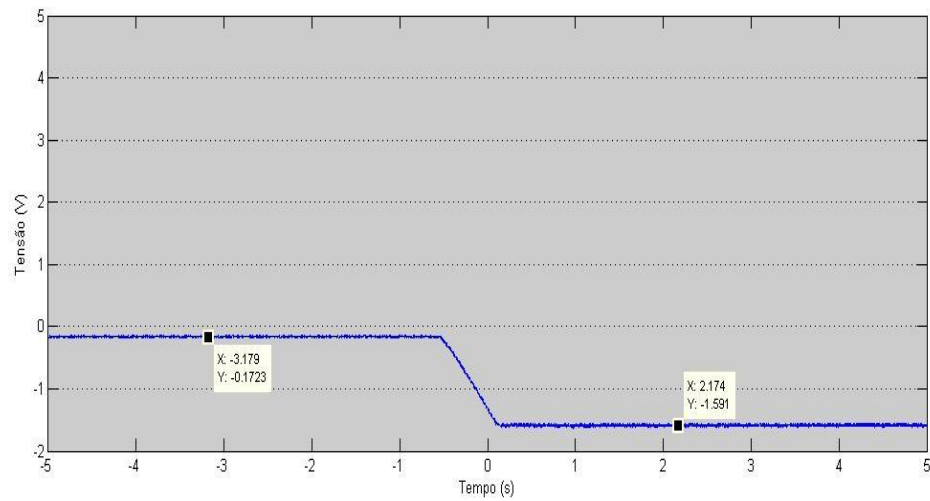


Figura 61: Curva de deflexão no Matlab(6)

Avanço real: 1.13mm

- Avanço de 0.5mm

Sem carga:

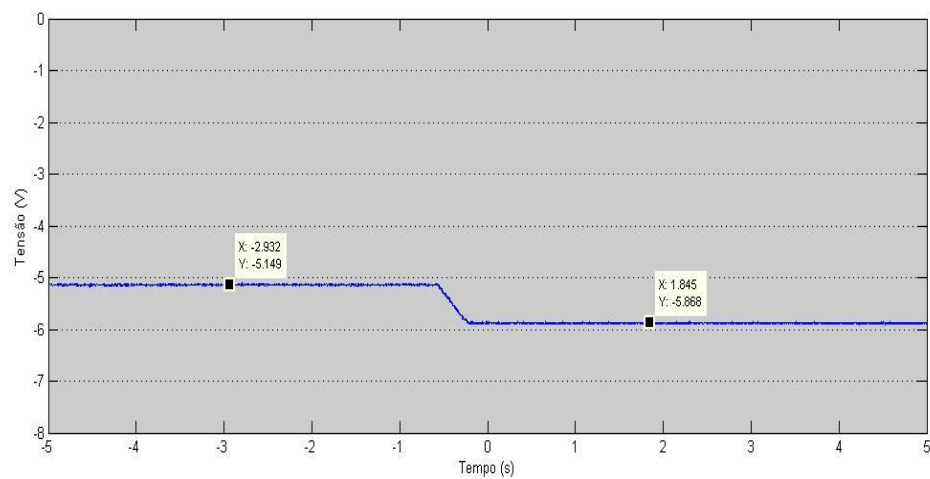


Figura 62: Curva de deflexão no Matlab(7)

Avanço real: 0.46mm

Carga de 20.8N:

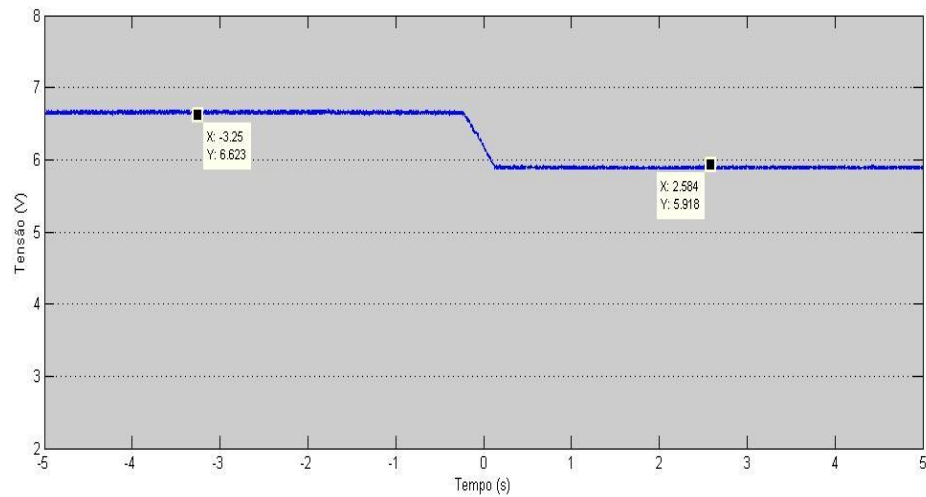


Figura 63: Curva de deflexão no Matlab(8)

Avanço real: 0.45mm

Carga de 41.4N:

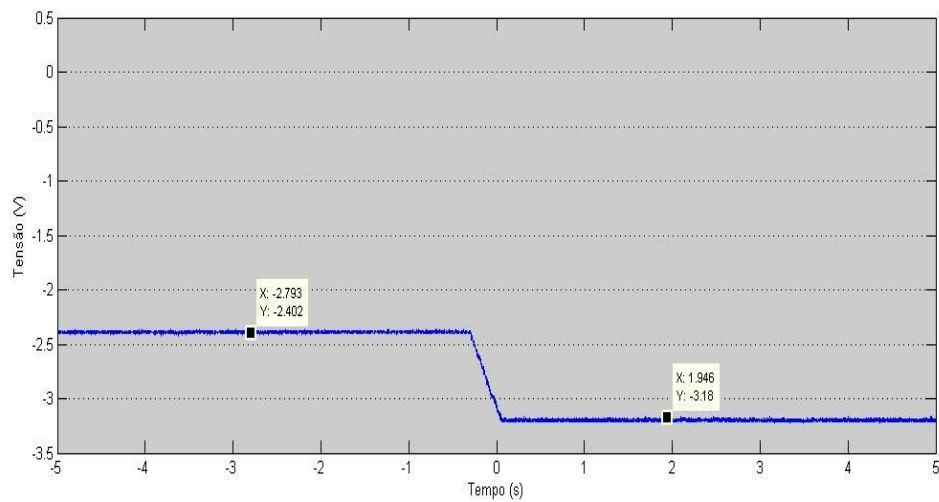


Figura 64: Curva de deflexão no Matlab(9)

Avanço real: 0.497mm

Analisando os resultados dos deslocamentos obtidos, verificou-se que com a montagem do novo acoplamento e aplicando-se as cargas mencionadas todos os valores de deflexão calculada ficaram dentro da faixa de 0,2mm de precisão desejada.

Os valores dos resultados de todos os ensaios apresentados anteriormente estão ilustrados na Tabela 5.

Tabela 5: Deflexão calculada para carga aplicada, em mm.

Sentido/ Comando	Geometria	Livre	20.8 N	41.4N
Longitudinal	2 Braços	-	0,064	0,189
	3 Braços	-	0,029	0,074
Perpendicular	2 Braços	-	0,153	0,384
	3 Braços	-	0,896	0,154
Avanço	Braço solto – 5mm	5,13	4,92	4,95
	Braço solto – 1mm	0,97	0,91	1,13
	Braço solto – 0.5mm	0,46	0,45	0,49

7. CONCLUSÕES

Neste trabalho apresentou-se uma série de alternativas de melhorias e aprimoramentos para uma máquina fresadora de arquitetura paralela, com o objetivo de torná-la primeiramente operável e depois com uma maior precisão de usinagem. Foi observado que algumas delas se tornaram inviáveis com um estudo mais aprofundado (construção de uma guia auxiliar acoplada ao fuso do motor) e outras não foram escolhidas para implementação por não se mostrarem tão eficazes quanto esperado (alteração do material dos braços da máquina para alumínio). No entanto, a solução escolhida para ser implementada no segundo semestre foi a substituição do acoplamento entre o eixo dos braços e o cubo fixado nas guias e no fuso do motor.

Esta medida adotada trouxe uma significativa mudança para a sustentação dos braços. Isto pode ser verificado pela validação feita através da medição de deflexão por laser medida no osciloscópio. Tanto de forma estática como dinâmica, a resposta às deflexões tiveram uma melhoria nos resultados medidos. É importante destacar que a outra medida sugerida para se obter uma maior melhora ainda, que é a associação de porcas no fuso não deve ser descartada. Sua única razão de não implementação foi a necessidade de um alto investimento inviável no momento, podendo ficar para possíveis trabalhos futuros.

Com a realização deste trabalho, espera-se que a máquina, uma vez operante de forma satisfatória, possa ser aproveitada para fins educacionais como forma de aprendizado para futuros alunos, por exemplo no estudo de controle, cinemática e arquitetura paralela.

Para trabalhos seguintes, é interessante mencionar a possível construção de uma base de ferramentas com mais um atuador para a mesma, o que concluiria o projeto inicial de uma máquina-fresadora e possibilitaria um estudo completo de suas funções.

8. REFERÊNCIAS BIBLIOGRÁFICAS

BONEV, I. ParalleMIC – The Parallel Mechanism Information Center; disponível em <http://www.parallemic.org/Terminology/General.html>, Acesso em Março 2013.

BRIOT, S. AND BONEV, I.A. (2008) ‘Accuracy analysis of 3-DOF planar parallel robots’, Mechanism and Machine Theory, Vol. 43, No. 4, pp.445–458.

CECCARELLI, M. Fundamentals of mechanics of robotic manipulation. Dordrecht: Kluwer, 2004.

GASTINEAU, MARIN. Máquina fresadora de arquitetura paralela. Trabalho de conclusão de curso. São Paulo, 2011.

GLOGOWSKY, PEDRO SPADA. Controle de robôs manipuladores paralelos de 3 graus de liberdade para operações pega-e-põe. Trabalho de conclusão de curso. São Paulo, 2008.

HARTMANN, VITOR NEVES. Desenvolvimento do protótipo de um robô de cinemática paralela para usinagem 2D com uma barra de ancoragem. Trabalho de iniciação científica. Processo FAPESP 07/50750-5 140p. 2007.

HARTMANN, VITOR NEVES. ‘Uma contribuição para o desenvolvimento de uma máquina fresadora de arquitetura paralela /V.N. Hartmann. – ed.rev. –São Paulo, 2011. 134p

HARTMANN, V.N., STIPKOVIC-FILHO, M., HESS-COELHO, T.A. AND ALMEIDA, S.L.R. (2013) ‘Development of a milling parallel robot for educational purposes’, Int. J. Mechanisms and Robotic Systems, Vol. 1, No. 1, pp.68–82.

HESS-COELHO, T. A. An alternative procedure for type synthesis of parallel mechanisms. In: 12th IFToMM World Congress, 2007, paper A612.

HESS-COELHO, T.A.; GOJTAN, G.E.E.; FURTADO, G. P. Kinematic analysis of a 3-dof parallel mechanism for milling applications. The Open Mechanical Engineering Journal. Special Issue on Kinematic Design of Manipulators, Marco Ceccarelli (org.) in press, corrected Proof, ISSN 1874-155X, Betham Science Publishers Ltd., 2010.

Jianye Guo; Jingkui Li; Jiashun Shi; Yanli Zhang, "Analysis on the Influence of Parallelogram Mechanism to the Posture Error of Movable Platform in the Parallel Robot," Intelligent Computation Technology and Automation (ICICTA), 2010 International Conference on , vol.1, no., pp.706,710, 11-12 May 2010

LinuxCNC. Apresenta informações sobre o sistema operacional Linux Ubuntu, assim como manuais de referências e dúvidas freqüentes. Disponível em: <<http://www.linuxcnc.org/>>. Acesso em: 25/09/2013.

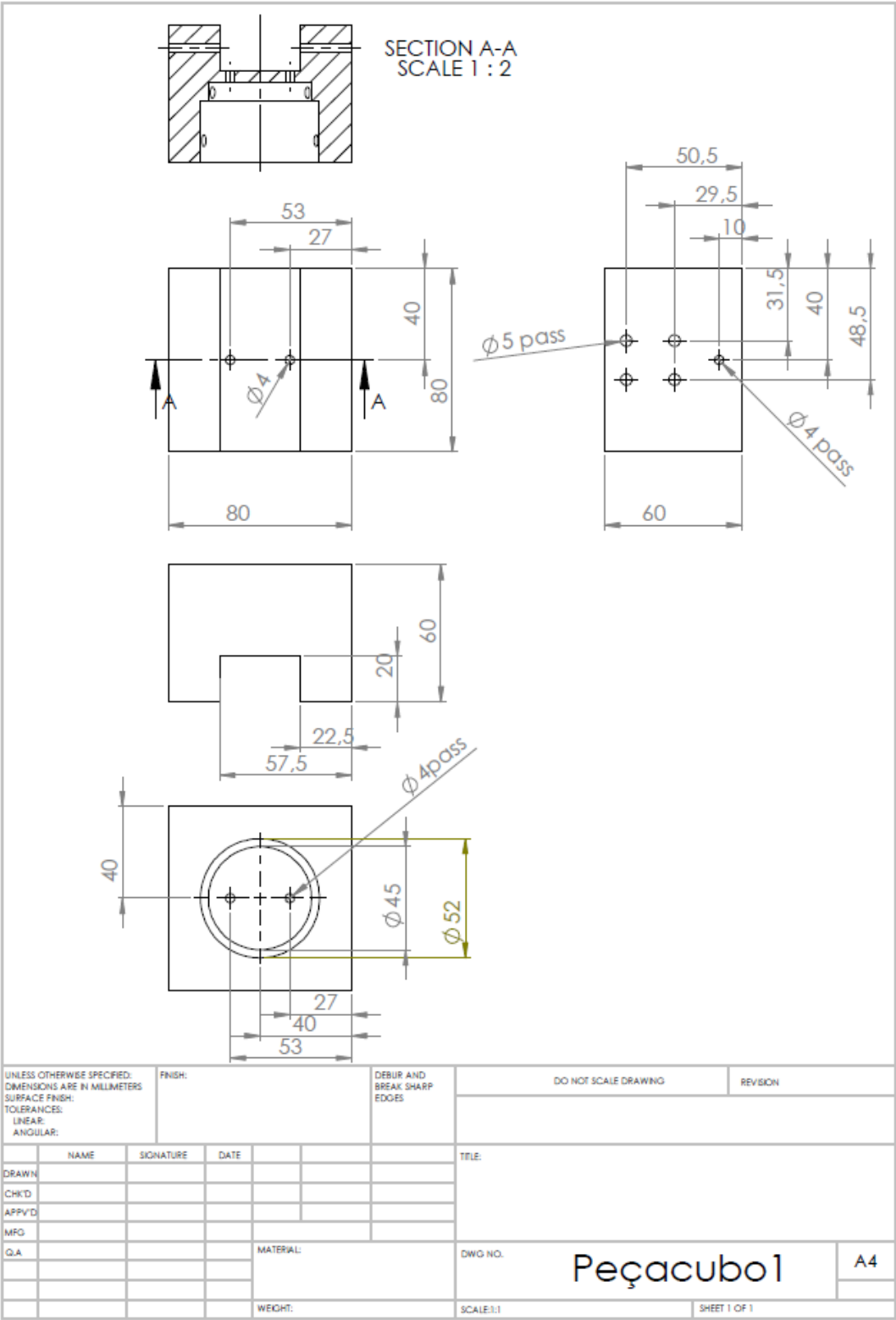
Merlet, J.P. Parallel Robots, Second Edition, Springer, 2006.

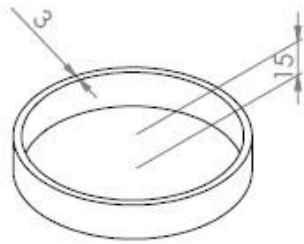
NOJIRI, R. Y. (2010) Fabricação de um robô de cinemática paralela para usinagem 2D. Trabalho de formatura.

Tsai, Lung-Wen. (1999). Robot Analysis: The mechanic of Serial and Parallel Manipulators, John Wiley & Sons, Inc.

Wang, J. and Masory, O. (1995). On the accuracy of a Stewart platform-part I: The effect of manufacturing tolerances. *IEEE Int. Conf. on Robotics and Automation*, pages 114-120, Atlanta.

ANEXO A – DESENHOS DE FABRICAÇÃO





SE NÃO ESPECIFICADO: DIMENSÕES EM MILÍMETROS ACABAM. SUPERFÍCIE: TOLERÂNCIAS: LINEAR: ANGULAR:			ACABAMENTO:			DEBUR AND BREAK: SHARP EDGES			NÃO MUDAR A ESCALA DO DESENHO			REVISÃO		
									TÍTULO:					
NOME			ASSINATURA			DATA								
DEL.														
VERIF.														
APROV.														
MANUF.									DES. Nº Espaçador A4					
QUALC.						MATERIAL:								
						PESO:			ESCALA:1:2			FOLHA 1 DE 1		

ANEXO B – CONFIGURAÇÃO DA MÁQUINA NO EMC2

Inicialmente, para a configuração do software EMC2 é preciso acessar a opção “StepConf Wizard” no menu de “applications -> CNC” do sistema Linux (que se equivale ao menu INICIAR do software Windows).

Em seguida é apresentada a opção de criação de uma nova configuração, ou a modificação de uma já existente. Deve-se optar pela criação de uma nova: “Create NewConfiguration”, figura 1.

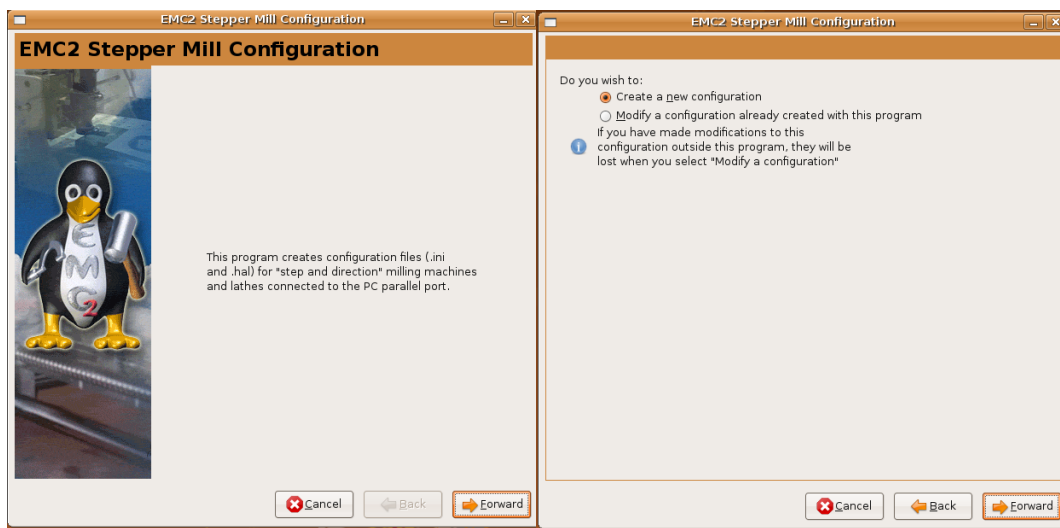


Figura 1: Configuração do EMC2- parte 1

Na tela subsequente, figura 2, são exibidas as opções para a definição das informações elementares a respeito do robô paralelo, como o seu nome, o conjunto dos eixos (espaço de trabalho esférico, cartesiano, etc.), as unidades métricas que serão utilizadas pela máquina, o modelo de driver de motor de passo que o software irá controlar (no caso o controlador L297) e o endereço da porta paralela, que será o principal dispositivo que propiciará a troca de informações entre o software EMC2 e o equipamento por ele controlado.

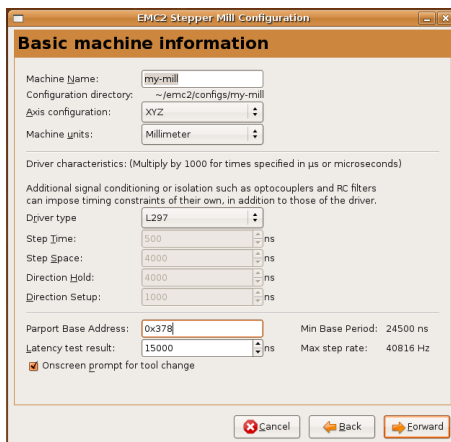


Figura 2: Configuração do EMC2-parte2

Na tela seguinte (figura 3), deve-se prestar especial atenção pois se estará diante de uma das etapas mais sensíveis da configuração do programa, chamada de pinagem, é nesta fase que se define quais pinos da porta paralela do PC serão responsáveis pelas funções do autômato. De forma que não ocorram erros que podem colocar em perigo a placa-mãe do PC, o usuário deve ter o domínio do equipamento que está controlando, pois nenhum sinal de INPUT deverá colocado em um pino errado, ou ter sua voltagem desrespeitada.

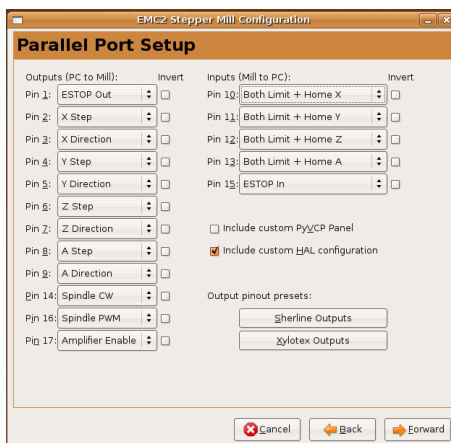


Figura 3: Configuração do
EMC2-parte3

Neste projeto foram utilizados os pinos 2 a 7 (sinais de clock e direção dos eixos X, Y e Z), e o pino 18 (GND de referência entre o PC e os drivers, que será explicado mais adiante) da porta paralela.

Seguindo com a última etapa de configuração, é apresentada a tela mostrada na figura 4, onde o eixo X deverá ser configurado (os eixos Y e Z serão configurados posteriormente). Nesta

etapa deverá ser fixado o número de passos por volta do motor, o avanço (em mm/rev), velocidade e aceleração máximas, e também a amplitude do espaço de trabalho e a localização inicial (zero-máquina).

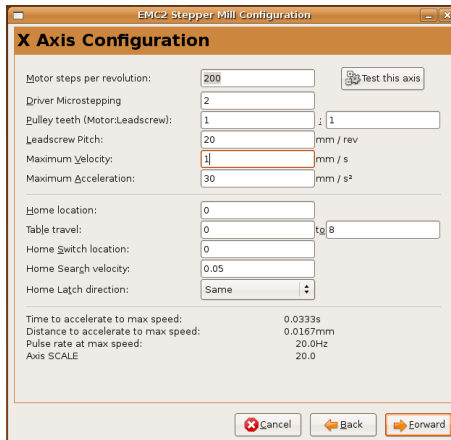


Figura 4: Configuração do EMC2-parte4

Superada a fase de configuração da máquina, já é possível carregá-la no software EMC2, que deverá ser iniciado no mesmo menu que o “StepConf Wizard”. Precedente à primeira tela do software, estará disponível uma janela de navegação por pastas, nela o utilizador poderá escolher o tipo de configuração de máquina CNC, ou robô, com a qual irá trabalhar. Seguindo corretamente os comandos de configuração apresentados anteriormente, na janela de seleção de máquinas haverá o nome da máquina criada pelo usuário na etapa “StepConf Wizard”.

Embora tenham sido mostradas todas as etapas de como criar uma configuração personalizada de máquina para o EMC2, neste projeto tal procedimento não foi realizado. Utilizou-se uma configuração pré-existente com o registro de “stepper_mm2” (em anexo). Esta configuração define uma máquina cartesiana de 3 eixos independentes, X, Y e Z, acionada por motores de passo. O leitor poderá estranhar tal escolha, visto que o robô deste projeto é de arquitetura paralela, e, portanto, não possui eixos independentes. Tal escolha é justificada, pois cada “eixo independente” no EMC2 representa um atuador (L1, L2, L3).

ANEXO C – MANUAL DA FRESADORA

- Ligar o computador e escolher o sistema operacional Ubuntu, kernel 2.6.15-**magma**.
- Abrir o Scilab, carregar o programa desejado (códigos no anexo C), escrever a matriz de pontos referentes às trajetórias a serem usinadas e, por fim, executar o programa com a matriz escolhida.

Programa	Matriz	Código	Arquivo salvo para o EMC2
geraG.sci	$A = [0 \ 0; \ 100 \ -50]$	writeCoord(A)	coord.ngc
geraG com coef.sci	Semelhante	writeCoord(A)	geraGcomcoef.ngc
linha sem F.sce	Semelhante	linhasemF(A)	linhasemF.ngc
linha com F.sce	$B = [0 \ 0 \ 200; \ -100 \ 40 \ 100; \ -20 \ -70 \ 150]$	linhacomF(B)	linhacomF.ngc
Linha com F e 0.sce	$D = [-100 \ 40 \ 100; \ -20 \ -70 \ 150]$	writeGcomFe0(D)	linhacomFe0.ngc
circulo em 0.sce	$E = [\text{Raio} \ \text{ângulo} \ F]$ $E = [60 \ 360 \ 150]$	Circuloem0(E)	Circuloem0.ngc

- Configurar o programa da máquina do EMC2

É nesse arquivo que são feitas as alterações básicas do robô como suas dimensões, o espaço de trabalho, a velocidade de usinagem, etc. A configuração padrão é a máquina “Vini”. Somente altere essa configuração caso queira fazer alguma modificação na máquina.

- Carregar o arquivo .ngf criado pelos exemplos do anexo C deste trabalho ou abrir o arquivo criado pelo programa em scilab de autoria do leitor.
- Pressionar a parada de emergência no EMC2 (canto superior esquerdo na figura 1).
- Ligar a fonte de tensão e apertar o botão “ON”.
- Verificar se todos os braços estão conectados e no centro da área de trabalho.
- Apertar o “Play”.

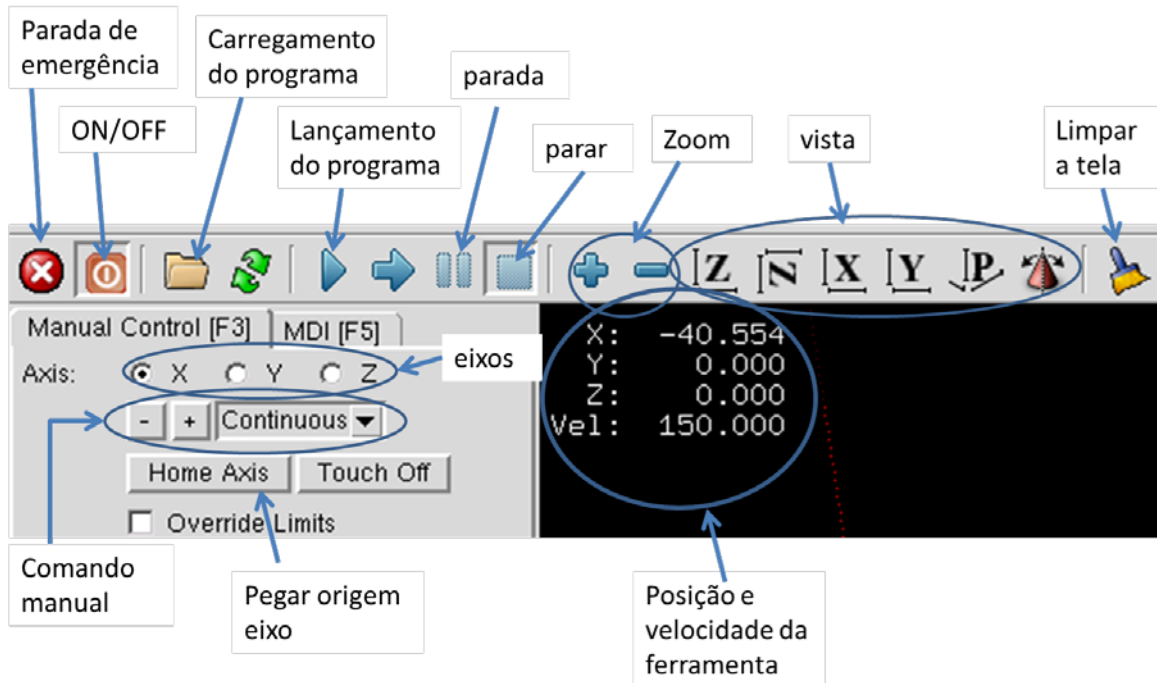


Figura 1: Explicação dos comandos do EMC2 (Gastineau, 2011)

ANEXO D – PROGRAMAS DO SCILAB

GeraG.sci

// Versão 10.09.12

// Arquivo OK!

// GeraG

function []=writeCoord(mat)

// As medidas de entrada devem ser dadas em mm.

// A matriz 'mat' é da forma (x1 y1, x2 y2, ...).

// A entrada deve ser feita baseada no zero peça, ou seja, (0,0) está no centro da máquina.

// Sua função é discretizar os pontos de entrada e gerar o arquivo em código G, não dependendo DIRETAMENTE das características da máquina.

// Dimensões da máquina:

Lf = 1417; *// mm*

x0 = Lf/2; *// Valor do referencial peça no referencial máquina.*

y0 = Lf*sqrt(3)/6; *// Valor do referencial peça no referencial máquina.*

// Correção de comprimento de braço para coordenadas incrementais:

corr1 = l1Inv(x0,y0);

corr2 = l2Inv(x0,y0);

corr3 = l3Inv(x0,y0);

//disp(corr1,corr2,corr3);

// Dados de avanço:

F=200; *// Unidade não especificada, pois é o avanço dos motores, e não da guia.*

tam=size(mat); *// O número de linhas fica em 'tam(1)', e o de colunas em 'tam(2)'.*

// Cria matriz 'disc' vezes maior (discretizado 'disc' vezes mais):

disc = 100;

mat2=zeros((tam(1)-1)*disc+1,2);

tam2=size(mat2);

// Uma maior discretização é requerida para desenhos grandes. Na dúvida, manter em 100.

// Atribui os valores discretizados para a matriz 'mat2':

cm2=1; *// A variável 'cm2' é relativa aos valores que serão preenchidos na matriz 'mat2'.*

for cont=1:(tam(1)-1) *// A variável 'cont' é relativa aos valores originais das coordenadas a serem seguidas ('mat').*

for step=0:(1/disc):(1-1/disc) *// A variável 'step' é relativa aos valores discretizados em 'disc' vezes.*

mat2(cm2,1)=mat(cont,1)+(mat(cont+1,1)-mat(cont,1))*step;

mat2(cm2,2)=mat(cont,2)+(mat(cont+1,2)-mat(cont,2))*step;

cm2=cm2+1;

end

end

mat2(cm2,1)=mat(tam(1),1);

mat2(cm2,2)=mat(tam(1),2); *// da a ultima valor de 'mat' para a ultima de 'mat2'*

// As variáveis 'cm2' e 'cont' NÃO são similares, e NÃO PODEM ser simplificadas em uma única.

//disp (mat2);

// Fecha arquivos de usuário que possam estar abertos:

file('close',file());

// Remove versão antiga do arquivo para evitar erros de execução:

mdelete('coord.ngc');

```

// Cria o arquivo em código G:
u=file('open','coord.ngc','unknown');
// A primeira linha é diferenciada pois contém o avanço 'F'.
fprintf(u, 'G91 G01 X%.1f Y%.1f Z%.1f F%d', l1Inv(mat2(1,1)+x0,mat2(1,2)+y0)-corr1,
l2Inv(mat2(1,1)+x0,mat2(1,2)+y0)-corr2, l3Inv(mat2(1,1)+x0,mat2(1,2)+y0)-corr3, F);
// Cria as demais linhas:
for cm2=2:tam2(1)
    fprintf(u, 'X%.1f Y%.1f Z%.1f', l1Inv(mat2(cm2,1)+x0,mat2(cm2,2)+y0)-corr1,
l2Inv(mat2(cm2,1)+x0,mat2(cm2,2)+y0)-corr2, l3Inv(mat2(cm2,1)+x0,mat2(cm2,2)+y0)-corr3);
end
// Cria o fechamento do arquivo e dados do programa de origem:
fprintf(u, 'M2'); //Que é o M2? para desligar a maquina, ver código G
fprintf(u, 'Arquivo gerado pelo geraG.sci, versão 10.09.12');
file('close',u);

endfunction

/*****

function [l1]=l1Inv(x, y)

// Medidas em mm.
// Sua função é determinar a posição do braço, dependendo DIRETAMENTE das características da máquina.

// Dimensões da máquina:
l = 625;

// Dados de compensação dos pulsos gerados:
VP1 = 100; // Valor Pedido. É o valor teórico de distância gerado no código G.
VR1 = 100; // Valor Real. É o valor experimental de distância observado e medido na máquina.
K1=VP1/VR1; // Compensação. Número que altera o valor escrito no arquivo 'coord.ngc' para se obter a res-
posta desejada.

// Resolve l1:
parc1 = roots([1 -2*(x*cos(30*pi/180)+y*sin(30*pi/180)) x^2+y^2-l^2]);
//disp(parc1);
l1 = K1 * min(real(parc1)); // A função 'real' é necessária pois 'roots' não retorna um número real.
//disp(l1);

endfunction

/*****

function [l2]=l2Inv(x, y)

// Medidas em mm.
// Sua função é determinar a posição do braço, dependendo DIRETAMENTE das características da máquina.

// Dimensões da máquina:
l = 625;

// Dados de compensação dos pulsos gerados:
VP2 = 100; // Valor Pedido. É o valor teórico de distância gerado no código G.
VR2 = 100; // Valor Real. É o valor experimental de distância observado e medido na máquina.

```

$K2=VP2/VR2$; // Compensação. Número que altera o valor escrito no arquivo 'coord.ngc' para se obter a resposta desejada.

// Para l2

$parc2 = \text{roots}([1 \ 2*((x-Lf)*\cos(30*\%pi/180)-y*\sin(30*\%pi/180)) \ (x-Lf)^2+y^2-l^2]);$
 //disp(parc2);

$l2 = K2 * \min(\text{real}(parc2));$ // A função 'real' é necessária pois 'roots' não retorna um número real.
 //disp(l2);

endfunction

/**

function [l3]=l3Inv(x, y)

// Medidas em mm.

// Sua função é determinar a posição do braço, dependendo DIRETAMENTE das características da máquina.

// Dimensões da máquina:

$l = 625$;

// Dados de compensação dos pulsos gerados:

$VP3 = 100$; // Valor Pedido. É o valor teórico de distância gerado no código G.

$VR3 = 100$; // Valor Real. É o valor experimental de distância observado e medido na máquina.

$K3=VP3/VR3$; // Compensação. Número que altera o valor escrito no arquivo 'coord.ngc' para se obter a resposta desejada.

// Para l3

$parc3 = \text{roots}([1 \ 2*(y-Lf*\cos(30*\%pi/180)) \ (x-Lf/2)^2+(y-Lf*\cos(30*\%pi/180))^2-l^2]);$
 //disp(parc3);

$l3 = K3 * \min(\text{real}(parc3));$ // A função 'real' é necessária pois 'roots' não retorna um número real.
 //disp(l3);

endfunction

GeraG com coof.sci

// geraG com coof (22/06/2011)

// Arquivo OK!

function []=writeCoord(mat)

// As medidas de entrada devem ser dadas em mm.

// A matriz 'mat' é da forma (x1 y1, x2 y2, ...).

// A entrada deve ser feita baseada no zero peça, ou seja, (0,0) está no centro da máquina.

// Sua função é discretizar os pontos de entrada e gerar o arquivo em código G, não dependendo DIRETAMENTE das características da máquina.

// Dimensões da máquina:

$Lf = 1417$; // mm

$x0 = Lf/2$; // Valor do referencial peça no referencial máquina.

$y0 = Lf*\sqrt{3}/6$; // Valor do referencial peça no referencial máquina.

// Correção de comprimento de braço para coordenadas incrementais:

```

//corr1 = l1Inv(x0,y0);
//corr2 = l2Inv(x0,y0);
//corr3 = l3Inv(x0,y0);
//disp(corr1,corr2,corr3);

// Dados de avanço:
F=200;      // Unidade não especificada, pois é o avanço dos motores, e não da guia.
tam=size(mat); // O número de linhas fica em 'tam(1)', e o de colunas em 'tam(2)'.

// Cria matriz 'disc' vezes maior (discretizado 'disc' vezes mais):
disc = 100;
mat2=zeros((tam(1)-1)*disc+1,2);
tam2=size(mat2);
// Uma maior discretização é requerida para desenhos grandes. Na dúvida, manter em 100.

// Atribui os valores discretizados para a matriz 'mat2':
cm2=1;      // A variável 'cm2' é relativa aos valores que serão preenchidos na matriz 'mat2'.
for cont=1:(tam(1)-1)    // A variável 'cont' é relativa aos valores originais das coordenadas a serem seguidas ('mat').
    for step=0:(1/disc):(1-1/disc)    // A variável 'step' é relativa aos valores discretizados em 'disc' vezes.
        mat2(cm2,1)=mat(cont,1)+(mat(cont+1,1)-mat(cont,1))*step;
        mat2(cm2,2)=mat(cont,2)+(mat(cont+1,2)-mat(cont,2))*step;
        cm2=cm2+1;
    end
end
mat2(cm2,1)=mat(tam(1),1);
mat2(cm2,2)=mat(tam(1),2); // da a ultima valor de 'mat' para a ultima de 'mat2'
// As variáveis 'cm2' e 'cont' NÃO são similares, e NÃO PODEM ser simplificadas em uma única.
//disp (mat2);

// Fecha arquivos de usuário que possam estar abertos:
file('close',file());
// Remove versão antiga do arquivo para evitar erros de execução:
mdelete('geraGcomcoof.ngc');

// Cria o arquivo em código G:
u=file('open','geraGcomcoof.ngc','unknown');
// A primeira linha é diferenciada pois contém o avanço 'F'.
fprintf(u, 'G91 G01 X%.1f Y%.1f Z%.1f F%d', l1Inv(mat2(2,1)+x0,mat2(2,2)+y0)-
l1Inv(mat2(1,1)+x0,mat2(1,2)+y0), l2Inv(mat2(2,1)+x0,mat2(2,2)+y0)-l2Inv(mat2(1,1)+x0,mat2(1,2)+y0),
l3Inv(mat2(2,1)+x0,mat2(2,2)+y0)-l3Inv(mat2(1,1)+x0,mat2(1,2)+y0), F);

// Cria as demais linhas:
for cm2=2:(tam2(1)-1)
    fprintf(u, 'X%.1f Y%.1f Z%.1f', l1Inv(mat2(cm2+1,1)+x0,mat2(cm2+1,2)+y0)-
l1Inv(mat2(cm2,1)+x0,mat2(cm2,2)+y0), l2Inv(mat2(cm2+1,1)+x0,mat2(cm2+1,2)+y0)-
l2Inv(mat2(cm2,1)+x0,mat2(cm2,2)+y0), l3Inv(mat2(cm2+1,1)+x0,mat2(cm2+1,2)+y0)-
l3Inv(mat2(cm2,1)+x0,mat2(cm2,2)+y0));
end

// Cria o fechamento do arquivo e dados do programa de origem:
fprintf(u, 'M2'); //Que é o M2? para desligar a maquina, ver codigo G
fprintf(u, 'Arquivo gerado pelo geraG com coef.sci, versão (22/06/2011)');
file('close',u);

```


endfunction

/**

function [l1]=l1Inv(x, y)

// Medidas em mm.

// Sua função é determinar a posição do braço, dependendo DIRETAMENTE das características da máquina.

// Dimensões da máquina:

l = 625;

// Dados de compensação dos pulsos gerados:

VP1 = 10; // Valor Pedido. É o valor teórico de distância gerado no código G.

VR1 = 25; // Valor Real. É o valor experimental de distância observado e medido na máquina.

K1=VP1/VR1; // Compensação. Número que altera o valor escrito no arquivo 'coord.ngc' para se obter a resposta desejada.

// Resolve l1:

parc1 = roots([1 -2*(x*cos(30*pi/180)+y*sin(30*pi/180)) x^2+y^2-l^2]);

//disp(parc1);

l1 = K1 * min(real(parc1)); // A função 'real' é necessária pois 'roots' não retorna um número real.

//disp(l1);

endfunction

/**

function [l2]=l2Inv(x, y)

// Medidas em mm.

// Sua função é determinar a posição do braço, dependendo DIRETAMENTE das características da máquina.

// Dimensões da máquina:

l = 625;

// Dados de compensação dos pulsos gerados:

VP2 = 10; // Valor Pedido. É o valor teórico de distância gerado no código G.

VR2 = 25; // Valor Real. É o valor experimental de distância observado e medido na máquina.

K2=VP2/VR2; // Compensação. Número que altera o valor escrito no arquivo 'coord.ngc' para se obter a resposta desejada.

// Para l2

parc2 = roots([1 2*((x-Lf)*cos(30*pi/180)-y*sin(30*pi/180)) (x-Lf)^2+y^2-l^2]);

//disp(parc2);

l2 = K2 * min(real(parc2)); // A função 'real' é necessária pois 'roots' não retorna um número real.

//disp(l2);

endfunction

/**

function [l3]=l3Inv(x, y)

// Medidas em mm.

// Sua função é determinar a posição do braço, dependendo DIRETAMENTE das características da máquina.

// Dimensões da máquina:

`l = 625;`

// Dados de compensação dos pulsos gerados:

`VP3 = 10;` *// Valor Pedido. É o valor teórico de distância gerado no código G.*

`VR3 = 25;` *// Valor Real. É o valor experimental de distância observado e medido na máquina.*

`K3=VP3/VR3;` *// Compensação. Número que altera o valor escrito no arquivo 'coord.ngc' para se obter a resposta desejada.*

// Para l3

`parc3 = roots([1 2*(y-Lf*sqrt(3)/2) (x-Lf/2)^2+(y-Lf*sqrt(3)/2)^2-l^2]);`

`//disp(parc3);`

`l3 = K3 * min(real(parc3));` *// A função 'real' é necessária pois 'roots' não retorna um número real.*

`//disp(l3);`

`endfunction`

Linhas sem F.sci

// linhas sem F (22/06/11)

// cinematica inversa

//metodo da intercecção do circulo com os vetores

`function []=linhasemF(mat)`

// As medidas de entrada devem ser dadas em mm.

// A matriz 'mat' é da forma [x1 y1 f1; x2 y2 f2; ...], fi= velocidade para chegar no ponto i.

// A entrada deve ser feita baseada no zero peça, ou seja, (0,0) está no centro da máquina.

// Dimensões da máquina:

`Lf = 1417;` *// mm*

`x0 = Lf/2;` *// Valor do referencial peça no referencial máquina.*

`y0 = Lf*sqrt(3)/4;` *// Valor do referencial peça no referencial máquina.*

//descritização à fazer veremos quando funcionar o programa

//origine no centro da maquina

// Correção de comprimento de braço para coordenadas incrementais:

`corr1 = l1Inv(x0,y0);`

`corr2 = l2Inv(x0,y0);`

`corr3 = l3Inv(x0,y0);`

//construção da matriz de coordenados: mat2

`tam=size(mat);` *// O número de linhas fica em 'tam(1)', e o de colunas em 'tam(2)'.*

`mat2=zeros(tam(1),3);`

`tam2=size(mat2);`

//Atribui os valor descritização para a matriz 'mat2'

`P2=1 //P2 = linha da matriz 'mat2'`

`for P1=1:(tam(1)-1) // para cada ponto da 'mat1' inseramos uma descretização`

[illegible]


```

for step=0:(1/disc):(1-1/disc)    // A variável 'step' é relativa aos valores discretizados em 'disc' vezes.
    mat2(cm2,1)=mat(cont,1)+(mat(cont+1,1)-mat(cont,1))*step;
    mat2(cm2,2)=mat(cont,2)+(mat(cont+1,2)-mat(cont,2))*step;
    mat2(cm2,3)=mat((cont+1),3);
    cm2=cm2+1;
end
end
mat2(cm2,1)=mat(tam(1),1);
mat2(cm2,2)=mat(tam(1),2);
mat2(cm2,3)=mat(tam(1),3);
// da a ultima valor de 'mat' para a ultima de 'mat2'
// As variáveis 'cm2' e 'cont' NÃO são similares, e NÃO PODEM ser simplificadas em uma única.
//disp (mat2);

// Fecha arquivos de usuário que possam estar abertos:
file('close',file());
// Remove versão antiga do arquivo para evitar erros de execução:
mdelete('linhacomF.ngc');

// Cria o arquivo em código G:
u=file('open','linhacomF.ngc','unknown');
// A primeira linha é diferenciada:
fprintf(u, 'G91 G01 X%.10f Y%.10f Z%.10f F%d', l1Inv(mat2(2,1)+x0,mat2(2,2)+y0)-
l1Inv(mat2(1,1)+x0,mat2(1,2)+y0), l2Inv(mat2(2,1)+x0,mat2(2,2)+y0)-l2Inv(mat2(1,1)+x0,mat2(1,2)+y0),
l3Inv(mat2(2,1)+x0,mat2(2,2)+y0)-l3Inv(mat2(1,1)+x0,mat2(1,2)+y0), mat2(2,3));
// Cria as demais linhas:
for cm3=2:(tam2(1)-1)
    fprintf(u, 'X%.10f Y%.10f Z%.10f F%d', l1Inv(mat2(cm3+1,1)+x0,mat2(cm3+1,2)+y0)-
l1Inv(mat2(cm3,1)+x0,mat2(cm3,2)+y0), l2Inv(mat2(cm3+1,1)+x0,mat2(cm3+1,2)+y0)-
l2Inv(mat2(cm3,1)+x0,mat2(cm3,2)+y0), l3Inv(mat2(cm3+1,1)+x0,mat2(cm3+1,2)+y0)-
l3Inv(mat2(cm3,1)+x0,mat2(cm3,2)+y0), mat2((cm3+1),3));
end

// Cria o fechamento do arquivo e dados do programa de origem:
fprintf(u, 'M2'); //Que é o M2? para desligar a maquina, ver codigo G
fprintf(u, 'Arquivo gerado pelo: linha com F.sci, versão (22/06/2011)');
file('close',u);

endfunction

//*****

function [l1]=l1Inv(x, y)

// Medidas em mm.
// Sua função é determinar a posição do braço, dependendo DIRETAMENTE das características da máquina.

// Dimensões da máquina:
l = 625;

// Dados de compensação dos pulsos gerados:
VP1 = 10;    // Valor Pedido. É o valor teórico de distância gerado no código G.
VR1 = 25;    // Valor Real. É o valor experimental de distância observado e medido na máquina.
K1=VP1/VR1;  // Compensação. Número que altera o valor escrito no arquivo 'coord.ngc' para se obter a res-
posta desejada.

```

// Resolve l1:

```
l1 = K1 * (1/2)*(sqrt(3)*x+y-sqrt((-sqrt(3)*x-y)^2-4*(-390625+(x^2)+(y^2))));
//disp(l1);
```

endfunction

//**

function [l2]=l2Inv(x**, **y**)**

// Medidas em mm.

// Sua função é determinar a posição do braço, dependendo DIRETAMENTE das características da máquina.

// Dimensões da máquina:

l = 625;

// Dados de compensação dos pulsos gerados:

VP2 = 10; // Valor Pedido. É o valor teórico de distância gerado no código G.

VR2 = 25; // Valor Real. É o valor experimental de distância observado e medido na máquina.

K2=VP2/VR2; // Compensação. Número que altera o valor escrito no arquivo 'coord.ngc' para se obter a resposta desejada.

// Para l2

```
l2 = K2 * 1/2*(1417*sqrt(3)-sqrt(3)*x+y-sqrt((-1417*sqrt(3)+sqrt(3)*x-y)^2-4*(1617264-2834*x+x^2+y^2))));
```

//disp(l2);

endfunction

//**

function [l3]=l3Inv(x**, **y**)**

// Medidas em mm.

// Sua função é determinar a posição do braço, dependendo DIRETAMENTE das características da máquina.

// Dimensões da máquina:

l = 625;

// Dados de compensação dos pulsos gerados:

VP3 = 10; // Valor Pedido. É o valor teórico de distância gerado no código G.

VR3 = 25; // Valor Real. É o valor experimental de distância observado e medido na máquina.

K3=VP3/VR3; // Compensação. Número que altera o valor escrito no arquivo 'coord.ngc' para se obter a resposta desejada.

// Para l3

```
l3 = K3 * 1/2*(1417*sqrt(3)-2*y-sqrt((-1417*sqrt(3)+2*y)^2-4*(1617264-1417*x+x^2-1417*sqrt(3)*y+y^2))));
```

//disp(l3);

endfunction

Linha com F e saio do 0.sci

// linha com F e saio do 0 (22/06/2011)

//partir da origem maquina

function []=writeGcomFe0(mat)

// As medidas de entrada devem ser dadas em mm.

// A matriz 'mat' é da forma (x1 y1, x2 y2, ...).

// A entrada deve ser feita baseada no zero peça, ou seja, (0,0) está no centro da máquina.

// Sua função é discretizar os pontos de entrada e gerar o arquivo em código G, não dependendo DIRETAMENTE das características da máquina.

// Dimensões da máquina:

Lf = 1417; *// mm*

x0 = Lf/2; *// Valor do referencial peça no referencial máquina.*

y0 = Lf*sqrt(3)/6; *// Valor do referencial peça no referencial máquina.*

// Correção de comprimento de braço para coordenadas incrementais:

corr1 = l1Inv(x0,y0);

corr2 = l2Inv(x0,y0);

corr3 = l3Inv(x0,y0);

//disp(corr1,corr2,corr3);

// Dados de avanço:

//F=200; *// Unidade não especificada, pois é o avanço dos motores, e não da guia.*

tam0=size(mat); *// O número de linhas fica em 'tam(1)', e o de colunas em 'tam(2)'.*

mat1=zeros((tam0(1)+1),tam0(2))

tam=size(mat1); *// O número de linhas fica em 'tam(1)', e o de colunas em 'tam(2)'.*

mat1(1,1)=0; *// a primeira linha é o ponto origem*

mat1(1,2)=0;

mat1(1,3)=0;*//o moteur não tem velocidade no inicio*

for linh=2:tam(1)

mat1(linh,1)=mat((linh-1),1);

mat1(linh,2)=mat((linh-1),2);

mat1(linh,3)=mat((linh-1),3);

end

// Cria matriz 'disc' vezes maior (discretizado 'disc' vezes mais):

disc = 100;

mat2=zeros((tam(1)-1)*disc+1,3);

tam2=size(mat2);

// Uma maior discretização é requerida para desenhos grandes. Na dúvida, manter em 100.

// Atribui os valores discretizados para a matriz 'mat2':

cm2=1; *// A variável 'cm2' é relativa aos valores que serão preenchidos na matriz 'mat2'.*

for cont=1:(tam(1)-1) *// A variável 'cont' é relativa aos valores originais das coordenadas a serem seguidas ('mat').*

for step=0:(1/disc):(1-1/disc) *// A variável 'step' é relativa aos valores discretizados em 'disc' vezes.*

mat2(cm2,1)=mat1(cont,1)+(mat1(cont+1,1)-mat1(cont,1))*step;

mat2(cm2,2)=mat1(cont,2)+(mat1(cont+1,2)-mat1(cont,2))*step;

mat2(cm2,3)=mat1((cont+1),3);

```

    cm2=cm2+1;
end
end
mat2(cm2,1)=mat1(tam(1),1);
mat2(cm2,2)=mat1(tam(1),2);
mat2(cm2,3)=mat1(tam(1),3);
// da a ultima valor de 'mat' para a ultima de 'mat2'
// As variáveis 'cm2' e 'cont' NÃO são similares, e NÃO PODEM ser simplificadas em uma única.
//disp (mat2);

// Fecha arquivos de usuário que possam estar abertos:
file('close',file());
// Remove versão antiga do arquivo para evitar erros de execução:
mdelete('linhacomFe0.ngc');

// Cria o arquivo em código G:
u=file('open','linhacomFe0.ngc','unknown');
// A primeira linha é diferenciada:
fprintf(u, 'G91 G01 X%.1f Y%.1f Z%.1f F%d', l1Inv(mat2(1,1)+x0,mat2(1,2)+y0)-corr1,
l2Inv(mat2(1,1)+x0,mat2(1,2)+y0)-corr2, l3Inv(mat2(1,1)+x0,mat2(1,2)+y0)-corr3, mat2(2,3));
// Cria as demais linhas:
for cm2=2:(tam2(1)-1)
    fprintf(u, 'X%.1f Y%.1f Z%.1f F%d', l1Inv(mat2(cm2,1)+x0,mat2(cm2,2)+y0)-corr1,
l2Inv(mat2(cm2,1)+x0,mat2(cm2,2)+y0)-corr2, l3Inv(mat2(cm2,1)+x0,mat2(cm2,2)+y0)-corr3, mat2((cm2+1),3));
end
//ultima linha:
fprintf(u, 'X%.1f Y%.1f Z%.1f F%d', l1Inv(mat2(tam2(1),1)+x0,mat2(tam2(1),2)+y0)-corr1,
l2Inv(mat2(tam2(1),1)+x0,mat2(tam2(1),2)+y0)-corr2, l3Inv(mat2(tam2(1),1)+x0,mat2(tam2(1),2)+y0)-corr3,
mat2(tam2(1),3));

// Cria o fechamento do arquivo e dados do programa de origem:
fprintf(u, 'M2'); //Que é o M2? para desligar a maquina, ver codigo G
fprintf(u, 'Arquivo gerado pelo linhacomFe0.sci, versão (22/06/2011)');
file('close',u);

endfunction

/*****

function [l1]=l1Inv(x, y)

// Medidas em mm.
// Sua função é determinar a posição do braço, dependendo DIRETAMENTE das características da máquina.

// Dimensões da máquina:
l = 625;

// Dados de compensação dos pulsos gerados:
VP1 = 10; // Valor Pedido. É o valor teórico de distância gerado no código G.
VR1 = 25; // Valor Real. É o valor experimental de distância observado e medido na máquina.
K1=VP1/VR1; // Compensação. Número que altera o valor escrito no arquivo 'coord.ngc' para se obter a res-
posta desejada.

// Resolve l1:

```



```

I1= K1 * (1/2)*(sqrt(3)*x+y-sqrt((-sqrt(3)*x-y)^2-4*(-390625+(x^2)+(y^2))));
//disp(I1);

endfunction

//*****

function I2=I2Inv(x, y)

// Medidas em mm.
// Sua função é determinar a posição do braço, dependendo DIRETAMENTE das características da máquina.

// Dimensões da máquina:
l = 625;

// Dados de compensação dos pulsos gerados:
VP2 = 10; // Valor Pedido. É o valor teórico de distância gerado no código G.
VR2 = 25; // Valor Real. É o valor experimental de distância observado e medido na máquina.
K2=VP2/VR2; // Compensação. Número que altera o valor escrito no arquivo 'coord.ngc' para se obter a res-
posta desejada.

// Para I2
I2 = K2 * 1/2*(1417*sqrt(3)-sqrt(3)*x+y-sqrt((-1417*sqrt(3)+sqrt(3)*x-y)^2-4*(1617264-2834*x+x^2+y^2)));
//disp(I2);

endfunction

//*****

function I3=I3Inv(x, y)

// Medidas em mm.
// Sua função é determinar a posição do braço, dependendo DIRETAMENTE das características da máquina.

// Dimensões da máquina:
l = 625;

// Dados de compensação dos pulsos gerados:
VP3 = 10; // Valor Pedido. É o valor teórico de distância gerado no código G.
VR3 = 25; // Valor Real. É o valor experimental de distância observado e medido na máquina.
K3=VP3/VR3; // Compensação. Número que altera o valor escrito no arquivo 'coord.ngc' para se obter a res-
posta desejada.

// Para I3
I3 = K3 * 1/2*(1417*sqrt(3)-2*y-sqrt((-1417*sqrt(3)+2*y)^2-4*(1617264-1417*x+x^2-1417*sqrt(3)*y+y^2)));
//disp(I3);

endfunction

```

Circulo em 0.sci

// Circulo em 0

//partir da origem maquina

function []=circuloem0(**matC**)

// As medidas de entrada devem ser dadas em mm.

// A matriz 'mat' é da forma (R,teta,F).

// R= rayon, teta=angulo do arco de circulo, F= velocidade

// Sua função é discretizar os pontos de entrada e gerar o arquivo em código G, não dependendo DIRETAMENTE das características da máquina.

// Dimensões da máquina:

Lf = 1417; *// mm*

x0 = Lf/2; *// Valor do referencial peça no referencial máquina.*

y0 = Lf*sqrt(3)/6; *// Valor do referencial peça no referencial máquina.*

// equações do circulo:

*// x=R*cos(teta)*

*// y=R*sin(teta)*

//discretização do circulo

//descretiza o arco em d pontos

d = 100;

raio = **matC**(1,1);

teta =((**matC**(1,2))*3.141592654/180);

F=**matC**(1,3);

// matrice de coordonada do circulo

mat=**zeros**(d+1,3) *// precise de voltar no ponto do inicio do circulo*

linha=1

for pas=0:1:d

mat(linha,1)=raio*cos(linha*teta/d);

mat(linha,2)=raio*sin(linha*teta/d);

mat(linha,3)=F

 linha=linha+1;

end

// Dados de avanço:

//F=200; // Unidade não especificada, pois é o avanço dos motores, e não da guia.

tam0=**size**(**mat**); *// O número de linhas fica em 'tam(1)', e o de colunas em 'tam(2)'.*

mat1=**zeros**((**tam0**(1)+1),**tam0**(2));

tam=**size**(**mat1**); *// O número de linhas fica em 'tam(1)', e o de colunas em 'tam(2)'.*

mat1(1,1)=0; *// a primeira linha é o ponto origem*

mat1(1,2)=0;

mat1(1,3)=0;*//o moteur não tem velocidade no inicio*

for linh=2:**tam**(1)

mat1(linh,1)=**mat**((linh-1),1);

mat1(linh,2)=**mat**((linh-1),2);

mat1(linh,3)=**mat**((linh-1),3);

end

// Cria matriz 'disc' vezes maior entre os 2 primeiros ponto (origem -> circulo) (discretizado 'disc' vezes mais):

disc = 100;

mat2=zeros(tam(1)+disc-1,3);

tam2=size(mat2);

// Uma maior discretização é requerida para desenhos grandes. Na dúvida, manter em 100.

// Atribui os valores discretizados para a matriz 'mat2':

cm2=1; *// A variável 'cm2' é relativa aos valores que serão preenchidos na matriz 'mat2'.*

for step=0:(1/disc):(1-1/disc) *// A variável 'step' é relativa aos valores discretizados em 'disc' vezes.*

mat2(cm2,1)=mat1(1,1)+(mat1(2,1)-mat1(1,1))*step;

mat2(cm2,2)=mat1(1,2)+(mat1(2,2)-mat1(1,2))*step;

mat2(cm2,3)=mat1(2,3);

cm2=cm2+1;

end

for cont=2:(tam(1)-1) *// A variável 'cont' é relativa aos valores originais das coordenadas a serem seguidas ('mat').*

mat2(cm2,1)=mat1(cont,1);

mat2(cm2,2)=mat1(cont,2);

mat2(cm2,3)=mat1(cont,3);

cm2=cm2+1

end

mat2(cm2,1)=mat1(tam(1),1);

mat2(cm2,2)=mat1(tam(1),2);

mat2(cm2,3)=mat1(tam(1),3);

// da a ultima valor de 'mat' para a ultima de 'mat2'

// As variáveis 'cm2' e 'cont' NÃO são similares, e NÃO PODEM ser simplificadas em uma única.

//disp (mat2);

// Fecha arquivos de usuário que possam estar abertos:

file('close',file());

// Remove versão antiga do arquivo para evitar erros de execução:

mdelete('circuloem0.ngc');

// Cria o arquivo em código G:

u=file('open','circuloem0.ngc','unknown');

// A primeira linha é diferenciada:

fprintf(u, 'G91 G01 X%.1f Y%.1f Z%.1f F%d', l1Inv(mat2(2,1)+x0,mat2(2,2)+y0)-

l1Inv(mat2(1,1)+x0,mat2(1,2)+y0), l2Inv(mat2(2,1)+x0,mat2(2,2)+y0)-l2Inv(mat2(1,1)+x0,mat2(1,2)+y0),

l3Inv(mat2(2,1)+x0,mat2(2,2)+y0)-l3Inv(mat2(1,1)+x0,mat2(1,2)+y0), mat2(2,3));

// Cria as demais linhas:

for cm2=2:(tam2(1)-1)

fprintf(u, 'X%.1f Y%.1f Z%.1f F%d', l1Inv(mat2(cm2+1,1)+x0,mat2(cm2+1,2)+y0)-

l1Inv(mat2(cm2,1)+x0,mat2(cm2,2)+y0), l2Inv(mat2(cm2+1,1)+x0,mat2(cm2+1,2)+y0)-

l2Inv(mat2(cm2,1)+x0,mat2(cm2,2)+y0), l3Inv(mat2(cm2+1,1)+x0,mat2(cm2+1,2)+y0)-

l3Inv(mat2(cm2,1)+x0,mat2(cm2,2)+y0), mat2((cm2+1),3));

end

// Cria o fechamento do arquivo e dados do programa de origem:

fprintf(u, 'M2'); *//Que é o M2? para desligar a maquina, ver codigo G*

fprintf(u, 'Arquivo gerado pelo Circulo em 0.sci, (22/06/2011)');

```

file('close',u);

endfunction

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [l1]=l1Inv(x, y)

// Medidas em mm.
// Sua função é determinar a posição do braço, dependendo DIRETAMENTE das características da máquina.

// Dimensões da máquina:
l = 625;

// Dados de compensação dos pulsos gerados:
VP1 = 10; // Valor Pedido. É o valor teórico de distância gerado no código G.
VR1 = 25; // Valor Real. É o valor experimental de distância observado e medido na máquina.
K1=VP1/VR1; // Compensação. Número que altera o valor escrito no arquivo 'coord.ngc' para se obter a res-
posta desejada.

// Resolve l1:
l1= K1 * (1/2)*(sqrt(3)*x+y-sqrt((-sqrt(3)*x-y)^2-4*(-390625+(x^2)+(y^2))));
//disp(l1);

endfunction

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [l2]=l2Inv(x, y)

// Medidas em mm.
// Sua função é determinar a posição do braço, dependendo DIRETAMENTE das características da máquina.

// Dimensões da máquina:
l = 625;

// Dados de compensação dos pulsos gerados:
VP2 = 10; // Valor Pedido. É o valor teórico de distância gerado no código G.
VR2 = 25; // Valor Real. É o valor experimental de distância observado e medido na máquina.
K2=VP2/VR2; // Compensação. Número que altera o valor escrito no arquivo 'coord.ngc' para se obter a res-
posta desejada.

// Para l2
l2 = K2 * 1/2*(1417*sqrt(3)-sqrt(3)*x+y-sqrt((-1417*sqrt(3)+sqrt(3)*x-y)^2-4*(1617264-2834*x+x^2+y^2)));
//disp(l2);

endfunction

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [l3]=l3Inv(x, y)

// Medidas em mm.

```

// Sua função é determinar a posição do braço, dependendo DIRETAMENTE das características da máquina.

// Dimensões da máquina:

l = 625;

// Dados de compensação dos pulsos gerados:

VP3 = 10; // Valor Pedido. É o valor teórico de distância gerado no código G.

VR3 = 25; // Valor Real. É o valor experimental de distância observado e medido na máquina.

K3=VP3/VR3; // Compensação. Número que altera o valor escrito no arquivo 'coord.ngc' para se obter a resposta desejada.

// Para l3

***l3** = K3 * 1/2*(1417*sqrt(3)-2*y-sqrt((-1417*sqrt(3)+2*y)^2-4*(1617264-1417*x+x^2-1417*sqrt(3)*y+y^2)));*

//disp(l3);

endfunction

ANEXO E - DATA SHEET - DRIVER GECKO G251



G251 STEP MOTOR DRIVE REV 9: November 16, 2009

GECKODRIVE, INC.
14662 FRANKLIN AVE
SUITE E
TUSTIN, CA 92780
(714) 832-6874

Thank you for purchasing the G251 drive. The G251 microstep drive is warranted to be free of manufacturing defects for 1 year from the date of purchase. Anyone who is dissatisfied with it or is unable to make it work will be cheerfully refunded the purchase price if the G251 is returned within 15 days of the purchase date and is in like-new condition.

PLEASE READ FIRST BEFORE USING THE G251

Before beginning, be sure to have a suitable step motor, a DC power supply suitable for the motor and a current set resistor. The motor's rated phase current must be between 0 amps and 3.5 amps and the current set resistor may be a 1/4 watt, 5% part. Finally have a STEP and DIRECTION pulse source available.

G251 TERMINAL WIRING

The G251 uses a 12 position 3.5mm terminal block. The wire gauge must be between 18 and 26 (AWG) and the wire must be stranded. Do not solder tin the wire strands as this will cause it to fatigue over time and it will eventually break. Do not exceed 30 in-oz (0.2 Nm) torque on the terminal screws.

TERMINAL 1	Power Ground Connect the negative (black) lead of your power supply to this terminal.
TERMINAL 2	Power (+) Connect the positive (red) lead of your power supply to this terminal. It must be between +15VDC to +50VDC.
TERMINAL 3	Current Set Connect one end of your current set resistor to this terminal.
TERMINAL 4	Current Set Connect the other end of your current set resistor to this terminal.
TERMINAL 5	Motor Phase A Connect one end of your "Phase A" motor winding here.
TERMINAL 6	Motor Phase /A Connect the other end of your "Phase A" motor winding here.
TERMINAL 7	Motor Phase B Connect one end of your "Phase B" motor winding here.
TERMINAL 8	Motor Phase /B Connect the other end of your "Phase B" motor winding here.
TERMINAL 9	Direction Connect the DIRECTION signal to this terminal.
TERMINAL 10	Step Connect the STEP signal to this terminal.
TERMINAL 11	Disable This terminal will force the winding currents to zero when shorted to ground (TERMINAL 12).
TERMINAL 12	Common Connect this to your logic (controller) ground.

POWER SUPPLY CONNECTION

TERMINAL 1	Power Ground Connect the power supply ground to this terminal
TERMINAL 2	Power (+) Connect the power supply "+" to this terminal



G251 STEP MOTOR DRIVE

REV 9: November 16, 2009

GECKODRIVE, INC.
14662 FRANKLIN AVE
SUITE E
TUSTIN, CA 92780
(714) 832-8874

The power supply voltage must be between 15VDC and 50VDC. The maximum power supply current required is 67% of the motor's rated phase current. An unregulated power supply may be used as long as the voltage stays between the limits; keep the ripple voltage to 10% or less for best results.

CAUTION! Power supply voltage in excess of 50VDC will damage the G251.

CAUTION! Never put a switch on the DC side of the power supply! This will damage, if not destroy, your drive!

A very accurate calculation of power supply voltage is to find your motor's inductance, and put it into the following equation.

$$32 * (\sqrt{\text{mH inductance}}) = \text{Power Supply Voltage}$$

If your motor has 2mH of inductance, the equation would look as follows.

$$32 * (\sqrt{2}) = 45.12\text{V}$$

In the example above 45.12V is the maximum voltage for that specific motor; any voltage above this will create unnecessary heating and any voltage below that will get you proportionally less speed than you could otherwise get at 45.12V.

CURRENT SET RESISTOR

TERMINAL 3 Current Set
Connect the current set resistor to this terminal.

TERMINAL 4 Current Set
Connect the other end of the current set resistor to this terminal.

This input programs the G251's current output to the motor windings. The current set resistor is a linear calculation of 1K ohms of resistance for every amp of motor winding current. This means that a 2.3A per phase motor will require a 2.3K ohm resistor on terminals 3 and 4. If your motor is 3.5A or above you can leave these terminals open and the drive will self limit to 3.5A; however, you will lose the auto current reduction feature of the drive. If you require this feature then a 3.5K resistor should be put on the drive.

MOTOR CONNECTION

TERMINAL 5 Phase A
Connect one motor winding to this terminal

TERMINAL 6 Phase /A
Connect the other end of the winding to this terminal

TERMINAL 7 Phase B
Connect the other motor winding to this terminal

TERMINAL 8 Phase /B
Connect the other end of the winding to this terminal

Connect one motor winding to terminals 5 and 6. Connect the other winding to terminals 7 and 8. Turn the power supply off when connecting or disconnecting the motor. If the motor turns in the wrong direction, reverse the motor winding connections to terminals 5 and 6.

CAUTION! Do not short the motor leads to each other or to ground; damage will result to the G251.

4-wire, 6-wire and 8-wire motors may be used. When 6-wire motors are used, they may be connected in half winding or full winding. This is equivalent to an 8-wire motor connected in parallel or series. If a motor is connected in series or full winding, the motor's phase current rating is half of its parallel or unipolar rating. The choice depends on the high-speed performance required; a parallel-connected motor will provide twice the power of a series-connected motor at the same power supply voltage.

STEP AND DIRECTION INPUTS

TERMINAL 9 Direction
Connect the DIRECTION line to this terminal.



G251 STEP MOTOR DRIVE

REV 9: November 16, 2009

GECKODRIVE, INC.
14662 FRANKLIN AVE
SUITE E
TUSTIN, CA 92780
(714) 832-8874

TERMINAL 10 Step
Connect the STEP line to this terminal.

These terminals can be driven with 3.3V or 5V logic. If 3.3V logic is used then the input current is -1mA for logic "0" and zero for logic "1". If 5V logic is used then the input current is -1mA for "0" logic and 0.87mA for logic "1".

DISABLE PIN

TERMINAL 11 Disable
This terminal will force the winding currents to zero when shorted to ground (TERMINAL 12).

Shorting this input to ground (terminals 11 to 12) forces winding currents to zero and stops all output switching activity. The G251 will continue totalizing step and direction inputs if any are sent. The power supply current drops to less than 15mA. The motor will return to its original position when the disable input is released if no step pulses have been sent and the motor has not been moved more than 2 full steps.

COMMON PIN

TERMINAL 12 Common
Connect this terminal to the controller signal ground

HEATSINKING

The G251 needs heatsinking for current settings greater than 3 amps. The case temperature (measured on the bottom plate) should not exceed 70 degrees C, and for best life should be kept to 50 degrees C or less. Use heatsink compound between the G251 and the heatsink.

CAUTION! Current settings above 3 Amps without a heatsink will result in damage to the G251.

The drive must be heatsinked to a piece of aluminum, preferably with fins and a fan to increase heat dissipation and surface area. Do not screw the drive directly to the door of your control cabinet, as this will typically not provide adequate heatsinking properties.

ADJUSTING THE TRIMPOT

This trimpot adjusts the motor for the smoothest possible low-speed operation. Set the motor speed to about 1/4 revolution per second, and then turn the trimpot until a distinct null is noted in the motor's vibration. This will result in the most even microstep placement for a given motor and power supply voltage.

DISCLAIMER

CERTAIN APPLICATIONS USING POWER PRODUCTS MAY INVOLVE POTENTIAL RISKS OF DEATH, PERSONAL INJURY OR SEVERE DAMAGE TO PROPERTY. GECKODRIVE INC. PRODUCTS ARE NOT DESIGNED, AUTHORIZED OR WARRANTED TO BE SUITABLE FOR USE IN LIFE-SUPPORT DEVICES OR OTHER CRITICAL APPLICATIONS. INCLUSION OF GECKODRIVE INC. PRODUCTS IN SUCH APPLICATIONS IS UNDERSTOOD TO BE FULLY AT THE PURCHASER'S OWN RISK.

In order to minimize risks associated with the purchaser's application, adequate design and operating safeguards must be provided by the purchaser to minimize inherent or procedural hazards. GECKODRIVE INC. assumes no liability for applications assistance or the purchaser's product design. GECKODRIVE INC. does not warrant or represent that any license, either express or implied, is granted under any patent right, copyright or other intellectual property right of GECKODRIVE INC.



G251 STEP MOTOR DRIVE

REV 9: November 16, 2009

GECKODRIVE, INC.
14662 FRANKLIN AVE
SUITE E
TUSTIN, CA 92780
(714) 832-8874

GECKODRIVE G251

MAIN CONNECTOR:

- 1 GND
- 2 +VDC
- 3 I-SET
- 4 I-SET
- 5 PHASE A
- 6 PHASE /A
- 7 PHASE B
- 8 PHASE /B
- 9 DIRECTION
- 10 STEP
- 11 DISABLE
- 12 SIGNAL GND

CURRENT SET RESISTOR:

CONNECT 1/4W RESISTOR FROM
'I-SET' (TERM. 3) TO 'I-SET' (TERM. 4).
MOTOR PHASE CURRENT = 1 AMP
PER 1,000 OHMS.

SPECIFICATIONS:

SUPPLY VOLTAGE	+5VDC MIN. +50VDC MAX
PHASE CURRENT	0A TO 3.5A
POWER DISSIPATION	3.3W AT 3.5A AND 50VDC
RESOLUTION	10 MICROSTEPS PER FULL STEP
INPUT LEVEL	0V TO 3.3V MIN, 0V TO 5V MAX
INPUT CURRENT	1mA
WEIGHT	0.45 OZ (12.5 GRAMS)
TEMPERATURE	0C TO 75C
STEP PULSE RATE	0Hz TO 300kHz
STEP PULSE WIDTH	1 MICROSECOND MIN
AUTO STANDBY	70% OF SET CURRENT

