



Matheus Rodrigues Goebel

**Projeto e acionamento de um robô lagartixa
pneumático com aplicação de aprendizado
computacional**

Dissertação de Mestrado

Dissertação apresentada como requisito parcial para
obtenção do grau de Mestre pelo Programa de Pós-graduação
em Engenharia Mecânica, do Departamento de Engenharia
Mecânica da PUC-Rio.

Orientador: Prof. Marco Antonio Meggiolaro
Coorientador: Prof. Giovane Quadrelli

Rio de Janeiro
Outubro de 2022



Matheus Rodrigues Goebel

**Projeto e acionamento de um robô lagartixa
pneumático com aplicação de aprendizado
computacional**

Dissertação apresentada como requisito parcial para
obtenção do grau de Mestre pelo Programa de Pós-graduação
em Engenharia Mecânica da PUC-Rio. Aprovada pela
Comissão Examinadora abaixo:

Prof. Marco Antonio Meggiolaro

Orientador

Departamento de Engenharia Mecânica – PUC-Rio

Prof. Giovane Quadrelli

Coorientador

Departamento de Engenharia Elétrica – UCP

Prof. Mauro Speranza Neto

Departamento de Engenharia Mecânica – PUC-Rio

Prof. Luís Alexandre Estevão da Silva

Diretoria de Pesquisas – IPJBRJ

Todos os direitos reservados. A reprodução, total ou parcial do trabalho, é proibida sem a autorização da universidade, do autor e do orientador.

Matheus Rodrigues Goebel

Autor graduado em Engenharia Mecânica pela Universidade Católica de Petrópolis em 2018.

Ficha Catalográfica

Goebel, Matheus Rodrigues

Projeto e acionamento de um robô lagartixa pneumático com aplicação de aprendizado computacional / Matheus Rodrigues Goebel ; orientador: Marco Antonio Meggiolaro ; co-orientador: Giovane Quadrelli. – 2022.

117 f. : il. color. ; 30 cm

Dissertação (mestrado)–Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Engenharia Mecânica, 2022.

Inclui bibliografia

1. Engenharia Mecânica – Teses. 2. Lagartixa pneumática. 3. Robô escalador. 4. Garra de vácuo. 5. Sistemas eletropneumáticos. 6. Aprendizado computacional. I. Meggiolaro, Marco Antonio. II. Quadrelli, Giovane. III. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Engenharia Mecânica. IV. Título.

Agradecimentos

Agradeço aos meus pais pela educação e valores passados, por sempre ajudarem e apoiarem em minhas decisões e por tudo que sempre fizeram para eu poder me tornar a pessoa que sou.

Agradeço ao meu irmão e à minha família em geral por todo o apoio durante toda a minha vida.

Agradeço aos meus amigos que durante os bons e os maus momentos, sempre estiveram ao meu lado para compartilhar conhecimentos e experiências, além de me ajudarem em diversas situações adversas da vida.

Agradeço aos professores e profissionais que ajudaram na minha formação acadêmica e profissional.

Agradeço ao Jorge Fontanella, que acreditou no meu potencial e sempre me aconselhou e ajudou em diversas situações dentro e fora da vida acadêmica, além de ter me ajudado na construção e elaboração do protótipo deste trabalho.

Gostaria de agradecer ao Dr. Marco Antonio Meggiolaro por me orientar neste trabalho, assim como o Dr. Giovane Quadrelli por ser meu coorientador.

Por último, mas não menos importante, gostaria de agradecer à PUC-Rio pela bolsa de isenção de mensalidades do mestrado.

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Código de Financiamento 001.

Resumo

Rodrigues Goebel, Matheus; Meggiolaro, Marco Antonio; Quadrelli, Giovane. **Projeto e acionamento de um robô lagartixa pneumático com aplicação de aprendizado computacional**. Rio de Janeiro, 2022. 117p. Dissertação de Mestrado – Departamento de Engenharia Mecânica, Pontifícia Universidade Católica do Rio de Janeiro.

Este trabalho apresenta um projeto mecânico de um robô lagartixa pneumática, capaz de se locomover em superfícies inclinadas em relação ao solo, através apenas de atuadores lineares que utilizam o ar comprimido como fonte de energia. Como parte fundamental do projeto mecânico neste trabalho, um sistema de garra é desenvolvido gerando vácuo mecanicamente, para haver uma economia de consumo energético no robô em comparação com os acessórios comerciais geralmente utilizados para esta tarefa de fixação. Com o protótipo de conceito fabricado e montado, o mesmo é submetido a uma bateria de testes com o intuito de posteriormente aplicar os dados obtidos em uma rede neural artificial, visando o aprendizado computacional dos movimentos do robô e, assim, sua otimização de velocidade em determinada sequência de movimentação. Após o treinamento desta rede neural, o protótipo é submetido a novos experimentos para verificar a eficiência do treinamento realizado e qual o impacto real obtido no robô. Finalmente, com a utilização de um sistema de câmeras, os deslocamentos do robô em diversas situações distintas são rastreados, visando gerar gráficos comparativos e analisar a repetibilidade e confiabilidade do sistema.

Palavras-chave

Lagartixa Pneumática; Robô Escalador; Garra de Vácuo; Sistemas Eletropneumáticos; Aprendizado Computacional; Redes Neurais Artificiais.

Abstract

Rodrigues Goebel, Matheus; Meggiolaro, Marco Antonio (Advisor); Quadrelli, Giovane (Co-Advisor). **Design and activation of a pneumatic gecko robot with application of machine learning.** Rio de Janeiro, 2022. 117p. MSc. Dissertation – Departamento de Engenharia Mecânica, Pontifícia Universidade Católica do Rio de Janeiro.

This work presents the mechanical design of a pneumatic gecko robot, capable of moving on inclined surfaces with respect to the ground, using only linear actuators with compressed air as a source of energy. As a fundamental part of the mechanical design in this work, a claw system is developed by generating vacuum mechanically, significantly reducing the energy consumption of the robot when compared to commercial accessories generally used for this clamping task. With the concept prototype manufactured and assembled, a series of tests are conducted to later apply the collected data in an artificial neural network. This network allows the computational learning of the robot movements, and thus its speed optimization for a certain defined gait. After training this neural network, the prototype is submitted to new experiments to verify the efficiency of the training performed and the real impact obtained on the robot. Furthermore, with the use of a camera system, the movements of the robot along several different situations are tracked, generating comparative graphs to analyze the repeatability and reliability of the system.

Keywords

Pneumatic Gecko; Climbing Robot; Vacuum Claw; Electropneumatic Systems; Machine Learning; Artificial Neural Networks.

Sumário

1. Introdução	16
1.1. Motivação e Objetivos	16
1.2. Revisão Bibliográfica.....	17
1.3. Organização da Dissertação	24
2. Projeto Mecânico	26
2.1. Fundamentos Teóricos	26
2.1.1. Pneumática.....	26
2.1.2. Pressão	27
2.1.3. Vácuo.....	28
2.1.4. Atuadores Pneumáticos.....	29
2.2. Desenvolvimento do Projeto.....	31
2.3. Modelagem do Sistema Pneumático	34
2.3.1. Modelo do Reservatório.....	36
2.3.2. Modelo dos Cilindros Principais.....	37
2.3.3. Modelo das Garras	38
2.3.4. Modelo dos Distanciadores.....	40
2.4. Simulação	42
2.5. Conceito das Garras	44
3. Redes Neurais	49
3.1. Neurônio Artificial e Função de Ativação	49
3.2. Aplicação da Rede Neural	51
3.3. Método de <i>Levenberg-Marquardt</i>	53
3.3.1. Método do <i>Gradient Descent</i>	54
3.3.2. Método de <i>Gauss-Newton</i>	55
3.3.3. Algoritmo de <i>Levenberg-Marquardt</i>	56
3.4. Especificações da Arquitetura da Rede Neural	57
4. Resultados Experimentais.....	58
4.1. Experimentos Realizados.....	58
4.2. Resultados.....	60
5. Conclusão e Trabalhos Futuros	67
6. Referências Bibliográficas.....	69
Anexo A: Gráficos de Consumo de Ar – <i>Vacuum Generators, Festo</i>	74
Anexo B: Programa de Captura da <i>JeVois</i>	76

Apêndice A: Desenhos de Fabricação das Peças em Alumínio.....	81
Apêndice B: Desenhos de Fabricação do Robô.....	83
Apêndice C: Algoritmo para simulação dinâmica no <i>Simulink</i>	88
Apêndice D: <i>Script</i> de Implementação da Rede Neural.....	98
Apêndice E: <i>Script</i> de Validação da Rede Neural.....	99
Apêndice F: Esquema Pneumático e Elétrico	100
Apêndice G: Programa do CLP	107
Apêndice H: Vídeo dos experimentos realizados com o protótipo.....	113
Apêndice I: Programa de Pós-processamento da <i>JeVois</i>	114

Lista de Figuras

Figura 1.1: <i>StickyBot</i> [1]	18
Figura 1.2: ROCR [2]	18
Figura 1.3: Robô eletroadesivo [3]	19
Figura 1.4: Representação do <i>gait</i> do robô limpador de janelas de vidro [4]	19
Figura 1.5: Robô escalador com rodas [5]	20
Figura 1.6: Esquema de montagem da garra robótica [6]	20
Figura 1.7: Robô quadrúpede [7]	21
Figura 1.8: Estágios do <i>gait</i> humano [8]	22
Figura 2.1: Ar comprimido em um volume confinado [37]	28
Figura 2.2: Atuador pneumático linear de simples ação	30
Figura 2.3: Atuador pneumático linear de dupla ação	30
Figura 2.4: Desenho da montagem do robô	31
Figura 2.5: Diagrama trajeto-passo do robô	33
Figura 2.6: Fluxo de ar através de pequeno orifício [43]	34
Figura 2.7: Diagrama de corpo livre de um cilindro principal	37
Figura 2.8: Diagrama de corpo livre de uma garra	39
Figura 2.9: Diagrama de corpo livre de um distanciador	41
Figura 2.10: Sinais de disparos das válvulas	43
Figura 2.11: Deslocamentos dos êmbolos dos cilindros principais	43
Figura 2.12: Deslocamentos dos êmbolos das garras	44
Figura 2.13: Deslocamentos dos êmbolos dos distanciadores	44
Figura 2.14: Tubo <i>Venturi</i>	45
Figura 2.15: Esboço de geração de vácuo mecanicamente	46

Figura 2.16: Protótipo de conceito da garra	46
Figura 2.17: Projeto da garra e do distanciador	47
Figura 3.1: Representação esquemática de um neurônio [49]	50
Figura 3.2: Função de ativação sigmoide unipolar [49]	50
Figura 3.3: Estrutura de redes neurais de <i>feedforward</i> [49]	51
Figura 3.4: Rede neural de <i>backpropagation</i> [49]	53
Figura 4.1: Teste do robô com inclinação	59
Figura 4.2: Treinamento da rede neural artificial	60
Figura 4.3: Teste de eficiência da rede neural	61
Figura 4.4: Esquema para testes com a <i>JeVois</i>	62
Figura 4.5: Gráfico de deslocamento com falha do robô	63
Figura 4.6: Deslocamento lento do robô em superfície sem inclinação	64
Figura 4.7: Deslocamento rápido do robô em superfície sem inclinação	64
Figura 4.8: Deslocamento lento do robô em superfície com inclinação	65
Figura 4.9: Deslocamento rápido do robô em superfície com inclinação	66

Lista de Abreviaturas

3D – Tridimensional

CLP – Controlador Lógico Programável

PLA – Políácido Láctico

PUC-Rio – Pontifícia Universidade Católica do Rio de Janeiro

Lista de Símbolos

A – Área

A_0 – Área do orifício

A_{CP} – Área do êmbolo do cilindro principal

A_{CP0} – Área do orifício para o cilindro principal

A_D – Área do êmbolo do distanciador

A_{D0} – Área do orifício para o distanciador

A_G – Área do êmbolo da garra

A_{G0} – Área do orifício para a garra

b – Coeficiente de atrito viscoso

C – Capacitância pneumática

C_d – Coeficiente de descarga

C_r – Razão crítica

D_{eG} – Diâmetro externo do êmbolo da garra

D_{hG} – Diâmetro da haste da garra

d_n – Saídas da rede neural desejadas

e_n – Erros de cálculo da rede neural

F – Força

F_G – Força da garra

g – Aceleração da gravidade

k_{CP} – Constante da mola do cilindro principal

k_D – Constante da mola do distanciador

k_G – Constante da mola da garra

L_G – Curso do êmbolo da garra

m – Massa do corpo da lagartixa

m_D – Massa das peças deslocadas pelo distanciador

m_G – Massa das peças deslocadas pela garra

n – Expoente politrópico

P – Pressão

P_1 – Pressão a montante

P_2 – Pressão a jusante

P_{atm} – Pressão atmosférica

P_{CP} – Pressão na câmara do cilindro principal

P_G – Pressão na câmara do distanciador

P_G – Pressão na câmara da garra

\dot{P}_R – Taxa de variação da pressão no reservatório

R – Constante dos gases

T – Temperatura do fluido

V – Volume

V_{CP0} – Volume inicial no cilindro principal

V_{D0} – Volume inicial no distanciador

V_G – Volume da câmara da garra

V_{G0} – Volume inicial na garra

V_R – Volume do reservatório

w – Vazão mássica

w_{atu} – Vazão mássica dos atuadores

w_C – Vazão mássica do compressor

w_{CP} – Vazão mássica do cilindro principal

w_D – Vazão mássica do distanciador

w_G – Vazão mássica da garra

w_{ent} – Vazão mássica entrando

w_{im} – Pesos das sinapses entre o neurônio anterior j e o de destino i

w_{liq} – Vazão mássica líquida

w_{Rliq} – Vazão mássica líquida do reservatório

w_{sai} – Vazão mássica saindo

x_{CP} – Posição do êmbolo do cilindro principal

\dot{x}_{CP} – Velocidade do êmbolo do cilindro principal

\ddot{x}_{CP} – Aceleração do êmbolo do cilindro principal

x_D – Posição do êmbolo do distanciador

\dot{x}_D – Velocidade do êmbolo do distanciador

\ddot{x}_D – Aceleração do êmbolo do distanciador

x_G – Posição do êmbolo da garra

\dot{x}_G – Velocidade do êmbolo da garra

\ddot{x}_G – Aceleração do êmbolo da garra

x_m – Entradas da rede neural

y_i – Saídas do neurônio i

y_n – Saídas da rede neural calculadas

λ – Inclinação da curva de ativação

λ_{lm} – Parâmetro de amortecimento de *Levenberg-Marquardt*

θ_i – Limite do neurônio i

η_i – Erros de medição

γ – Razão dos calores específicos

γ_{gd} – Comprimento do passo no método do *Gradient Descen*

1

Introdução

1.1

Motivação e Objetivos

A utilização de robôs para a execução de serviços cotidianos tem crescido muito nos últimos vinte anos, principalmente em tarefas que oferecem algum grau de risco à vida humana, como inspeção e pintura em prédios e cascos de navio, vistoria e limpeza de grandes estruturas de painéis solares, ou em dutos de prédios e fábricas, e.g.

Neste contexto, o objetivo principal deste trabalho é projetar um robô pneumático capaz de se deslocar em superfícies lisas com diversas inclinações. Para isso, o robô contará com juntas prismáticas e garras a vácuo especialmente desenvolvidas para a aplicação.

O desenvolvimento do trabalho conta com o projeto mecânico dos componentes para desenvolver um protótipo de conceito, sua modelagem dinâmica, testes funcionais com o protótipo para a obtenção de dados para aplicação em um algoritmo de redes neurais, a fim de otimizar a velocidade do robô e ter o aprendizado computacional do sistema.

A escolha por componentes pneumáticos faz-se principalmente por serem resistentes a ambientes hostis, como altas temperaturas, atmosfera corrosiva, e presença de umidade, além de não poluírem o meio ambiente, já que utilizam o ar comprimido como forma de energia. As garras do robô são projetadas de tal forma que o vácuo é gerado mecanicamente, a fim de otimizar o gasto energético durante sua utilização.

Além do projeto do robô, é feita a modelagem dinâmica que engloba a parte pneumática e mecânica. Assim, é possível conhecer o funcionamento do robô de

forma simplificada através das simulações. Tendo estas simulações como base, são gerados dados de entrada aleatórios para a realização de testes com o protótipo de conceito construído.

Os testes são de suma importância para a obtenção de um conjunto de dados confiável, haja vista que as simulações dinâmicas não englobam todas as variáveis do sistema. Este conjunto de dados é utilizado para treinar uma rede neural com aprendizagem supervisionada e *backpropagation*, no software *MATLAB*® [47], visando um aprendizado computacional da mecânica dos movimentos do robô projetado. Desta forma, mesmo se o *hardware* apresentar problemas, o computador aprenderá a compensá-los de maneira inteligente para garantir uma resposta adequada para o seu funcionamento.

1.2

Revisão Bibliográfica

Na literatura existem trabalhos relacionados ao sistema proposto utilizando diversos tipos de tecnologias de locomoção para ambientes inclinados, e com diferentes aplicações. Alguns projetos têm similaridade na questão conceitual do robô, que é se movimentar em superfícies com angulações variadas, já outros têm similaridade na parte da aplicação de redes neurais. Nesta seção, são apresentados alguns destes projetos que se relacionam de alguma forma com o sistema desenvolvido.

Começando pela parte conceitual do robô, Santos et al. [1] desenvolveram um robô inspirado em lagartixas, *StickyBot*, mostrado na Figura 1.1, que utiliza patas com microestruturas poliméricas para obter adesão com a superfície desejada, desde que esta seja lisa. O robô é capaz de se movimentar em várias direções, já que seu corpo é articulado.

Provancher et al. [2] utilizaram um sistema com pêndulo para se deslocar, e ganchos para formar as garras do robô, vide Figura 1.2. Este mecanismo foi batizado de *ROCR* e desenvolvido com o intuito de se mover em superfícies com rugosidade alta, como carpetes, haja vista que sua fixação é feita através de mini ganchos.

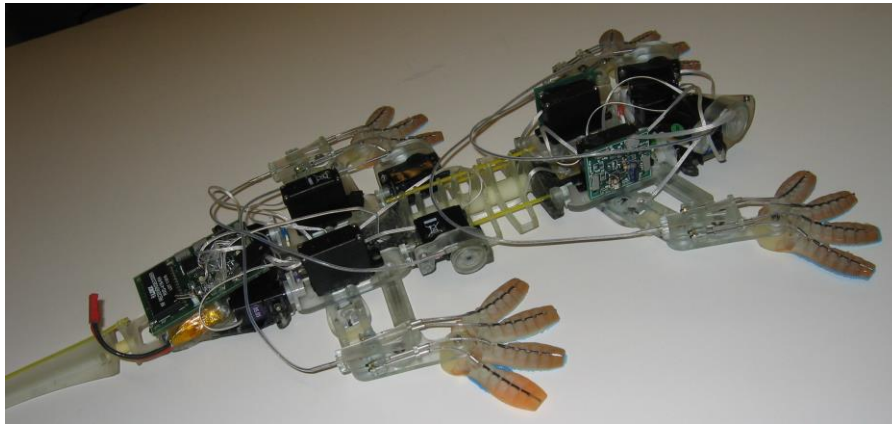


Figura 1.1: *StickyBot*. [1]

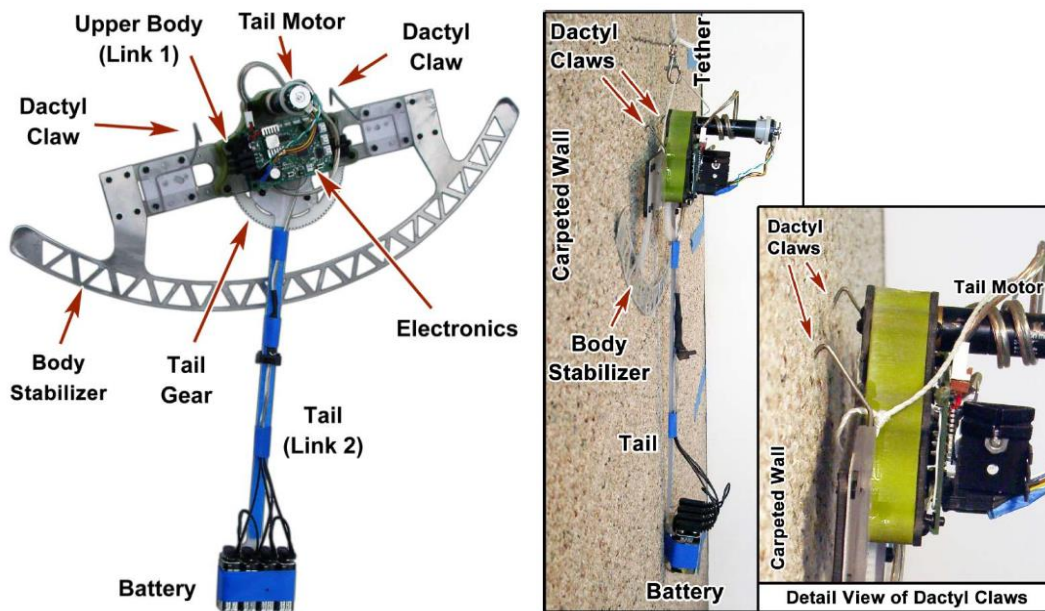


Figura 1.2: *ROCR*. [2]

Chen [3] utilizou a tecnologia de adesão eletrostática para que seu robô possa escalar superfícies de vidro. O robô é mostrado na Figura 1.3. Devido à sua articulação em seu corpo, ele pode executar uma mudança de direção em até 30° .

Mir-Nasiri et al. [4] projetaram um robô para limpeza de janelas de vidro utilizando ventosas e válvulas geradoras de vácuo, acionadas por meio de um sistema pneumático. Apesar de este robô utilizar ar comprimido para atuar seus componentes e gerar movimento, a forma de movimentação e de adesão diferem da apresentada neste trabalho. Seu movimento é feito através de rotação da base, e sua

adesão através de ventosas comerciais. O esquema do seu caminhar (*gait*) inicial pode ser visto na Figura 1.4.

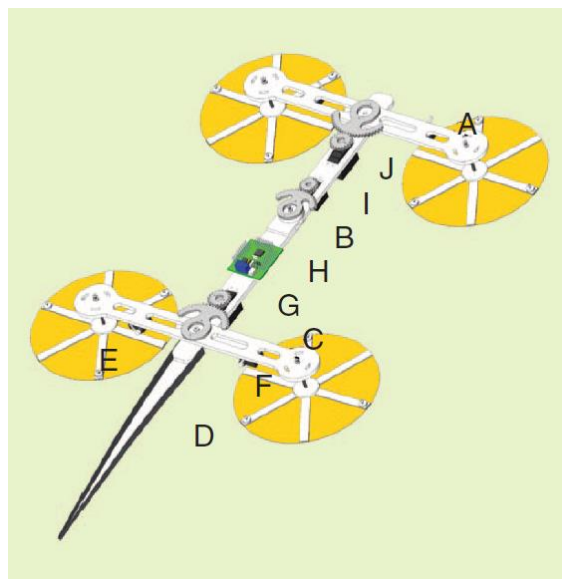


Figura 1.3: Robô eletroadesivo. [3]

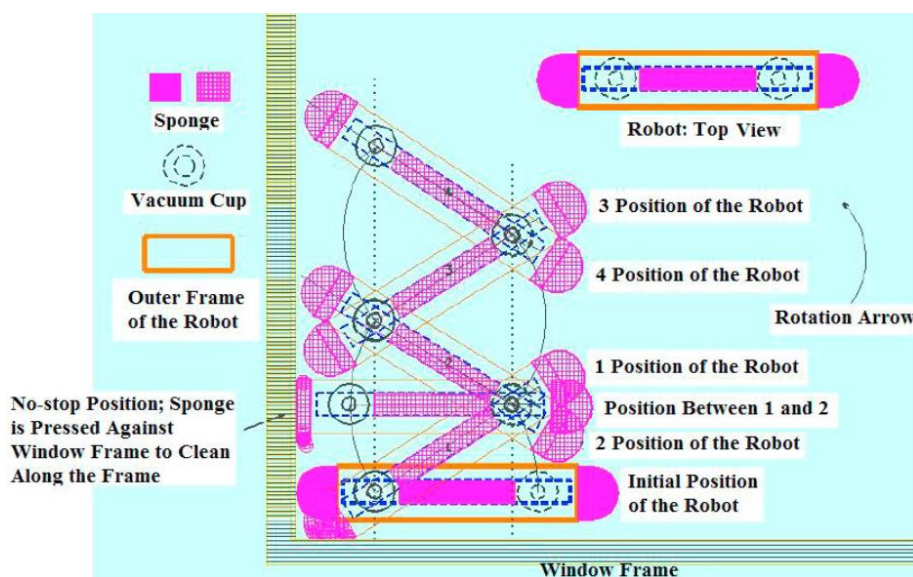


Figura 1.4: Representação do *gait* do robô limpador de janelas de vidro. [4]

Hossain et al. [5], como mostrado na Figura 1.5, projetaram um robô com rodas, capaz de se mover horizontal e verticalmente utilizando hélices em seu centro que, quando próximas à superfície em que se deseja a adesão, geram uma área de baixa pressão. A força causada pela diferença de pressão desta área e a pressão

atmosférica de seu exterior torna o robô capaz de se mover em superfícies inclinadas.

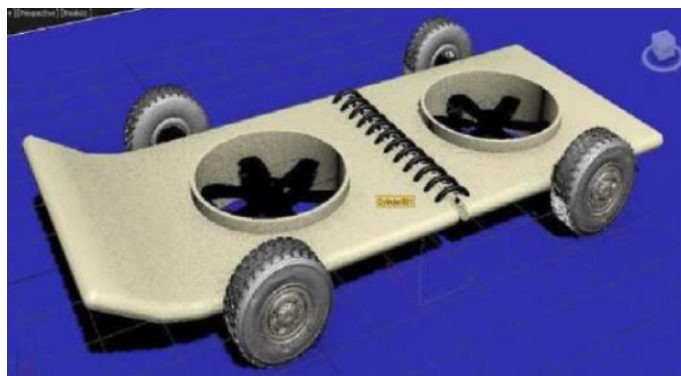


Figura 1.5: Robô escalador com rodas. [5]

Shi et al. [6] desenvolveram uma garra robótica cuja câmara de sucção funciona com dois fluidos, água e ar. As garras do robô usam a água para vedar e ajudar a manter a sucção gerada por uma bomba de vácuo, conseguindo aderência em superfícies com diferentes texturas através da diferença de pressão entre o seu interior e o meio externo. Um esquema de montagem desta garra robótica pode ser visto na Figura 1.6.

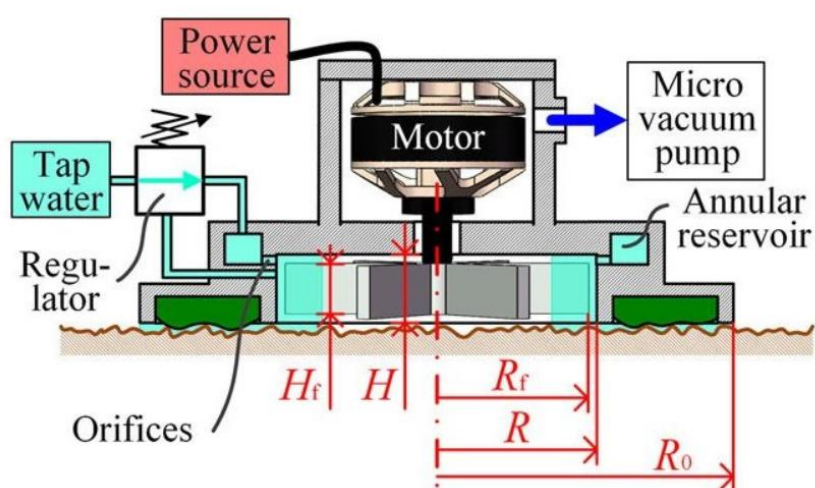


Figura 1.6: Esquema de montagem da garra robótica. [6]

Há também trabalhos na literatura na área de otimização do *gait*, ou seja, da maneira de andar de algum mecanismo. Não significa que estes estudos otimizaram

robôs que se enquadram na mesma categoria deste trabalho, ou que a otimização foi para o mesmo propósito buscado neste projeto apresentado, como discutido a seguir.

Masuri et al. [7] utilizaram um algoritmo genérico para autoaprendizagem dinâmica do movimento de caminhada de um robô quadrúpede, mostrado na Figura 1.7. As equações para simular esse movimento são complexas, por isso o protótipo foi utilizado para coletar dados experimentais com a finalidade de otimizar o *gait* do robô.

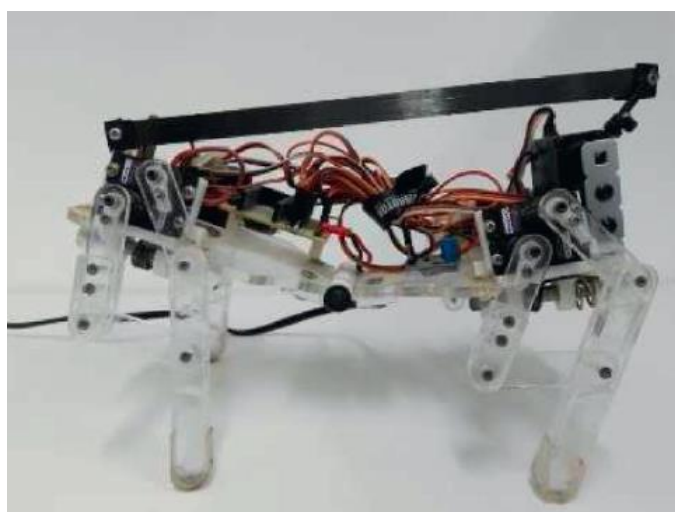


Figura 1.7: Robô quadrúpede. [7]

Alharbi et al. [8] utilizaram o método de *Levenberg-Marquardt* para trabalhar com previsões de ângulos da caminhada humana, considerando diversas velocidades. O trabalho é baseado em dados capturados com movimentos de pessoas reais, e os resultados do estudo mostram uma nova abordagem a ser utilizada em projetos de reabilitação protética e biomecânica. A Figura 1.8 representa vários estágios do *gait* humano.

Além destes, existem ainda na literatura outras pesquisas relacionadas com este trabalho. Em [9-12] discutem-se projetos de ventosas para fixações em diferentes tipos de superfícies, incluindo a modelagem e avaliação das mesmas.

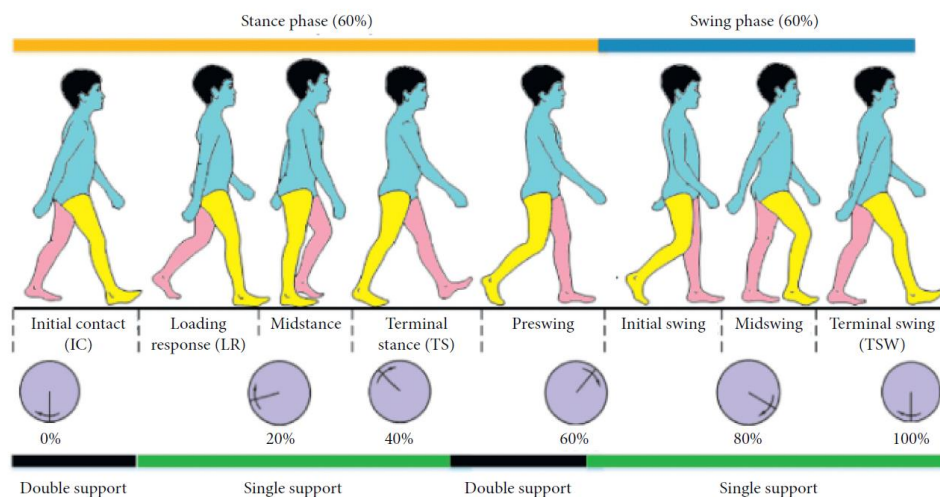


Figura 1.8: Estágios do *gait* humano. [8]

Koivikko et al. [9] desenvolveram uma ventosa macia acionada por vácuo, fabricada em 3D, capaz de se adequar e fixar em objetos com diferentes formas. Em Tsukagoshi et al. [10], uma ventosa acionada por vácuo é projetada, porém utiliza um anel de gel para ajudar na aderência e fixação da mesma em objetos variados. Liu, J. et al. [11] fizeram uma modelagem analítica para ventosas utilizadas em robôs limpadores de janelas de vidro e compararam os resultados com experimentos realizados. Em Ge et al. [12], fizeram uma análise de forças para ventosas disponíveis comercialmente e verificaram experimentalmente. Além disso, analisaram os fatores que influenciam na fixação das ventosas.

Com relação aos sistemas pneumáticos, em [13-20] é possível notar a versatilidade destes tipos de sistema, que utilizam ar comprimido como fonte de energia. Estas pesquisas contam com projetos e modelagem, controle e testes nos robôs, que servem para funções variadas.

Tuvayanond et al. [13] estudaram a implementação de controle em um robô pneumático e realizaram uma simulação da mesma. Herzig et al. [14] apresentaram um modelo matemático de um robô pneumático de dois graus de liberdade e um controle de posicionamento e ajuste de rigidez. Wang, Z. et al. [15] introduziram uma forma de fabricação de atuadores pneumáticos, especialmente para utilização em robôs macios, através de impressão 3D. Ugurlu et al. [16] propõem estruturas de controle para estruturas de músculos pneumáticos, visando posicionamento das juntas ou da rigidez do mecanismo.

Liu, Y. et al. [17] projetaram um robô paralelo pneumático de juntas prismáticas, além de terem sido feito seu modelo matemático e controle através de válvulas proporcionais de fluxo. Bai et al. [18] desenvolveram a modelagem completa de um robô pneumático de seis patas, além de implementação de controle. Liang et al. [19] propuseram um modelo de controle para um mecanismo com dois músculos pneumáticos é através de armazenagem de energia. Maeda et al. [20] projetaram um braço robótico pneumático, juntamente com uma mão pneumática. Além disso, desenvolveram um sistema de controle, fizeram simulações e experimentos.

Existem também outros tipos de robôs escaladores na literatura, além dos robôs já mostrados anteriormente com mais detalhes. Em [21-31] pode-se encontrar um leque amplo de tipos de robôs capazes de escalar superfícies com diferentes inclinações. Estes trabalhos indicados mostram os projetos mecânicos e aplicações dos robôs, modelagens matemáticas dos seus sistemas, controles de movimentação e experimentos realizados com os mesmos.

Han et al. [21] desenvolveram um sistema de controle para movimentação de um robô tipo lagartixa em ambientes complexos. Shao, D. et al. [22] projetaram um robô lagartixa capaz de se deslocar em terrenos acidentados com aplicação de controle, além de realizarem experimentos para medição das forças. Wang, W. et al. [23] focaram no estudo das superfícies adesivas secas desenvolvidas nos últimos anos para aplicação de robôs baseados em lagartixa. Chengwei et al. [24] modelaram e projetaram um robô capaz de se mover em superfícies de paredes utilizando uma ventosa e hélices para manter a estabilidade. Hernando et al. [25] desenvolveram um controle para robôs escaladores de múltiplas pernas, com compensação de possíveis movimentos errôneos. Nansai et al. [26] projetaram um robô escalador onde um único atuador aciona um mecanismo de quatro barras para executar a translação.

Vega-Heredia et al. [27] desenvolveram o projeto de um robô modular para limpeza de fachadas de prédios em vidro, além do controle e análise de resultados experimentais. Krosuri et al. [28] desenvolveram o projeto, controle e análise de testes feitos em um robô escalador bípede, além de estudarem a otimização considerando o tamanho e peso do robô comparados com a maximização do movimento. Zhong et al. [29] projetaram um robô escalador omnidirecional de três

rodas e seu controle de rastreamento de trajetória. Li et al. [30] desenvolveram o projeto de um robô escalador inspirado em uma lagartixa, seu *gait* é analisado e otimizado através de simulações matemáticas. Yu et al. [31] estudaram a aplicação de materiais adesivos secos em patas de robôs para deslocamento em superfícies verticais lisas.

Em [32-36] há pesquisas que apresentam discussões sobre o funcionamento de redes neurais para os fins de treinamento de aprendizado de movimento de robôs, além de mostrar aplicações com diversas técnicas.

Labruière [32] analisou a captura de dados para melhoria de movimentação de um robô utilizado para reabilitação clínica de pacientes que necessitam de movimentos de caminhada, através de aprendizado de máquina. Baiothman [33] aplicou e comparou algumas técnicas de aprendizado de máquinas em robôs bípedes, com objetivo de otimizar suas velocidades sem modificar o projeto físico, haja vista que os robôs foram projetados para jogar futebol. Shao, Y. et al. [34] fizeram um treinamento para aprendizado de movimentação de um robô quadrúpede com diferentes *gaits*. Ji et al. [35] projetaram um robô quadrúpede que utiliza atuadores não rígidos e treinaram seu *gait* para funcionar de forma otimizada nas tarefas de movimentação, além de fazerem testes para verificação de funcionamento. Yamamoto et al. [36] executaram uma captura de dados de movimentação em um robô quadrúpede, com diversos *gaits* distintos e os analisaram para achar padrões de movimentação através de aprendizado de máquina.

1.3

Organização da Dissertação

Para melhor compreensão das etapas de desenvolvimento deste trabalho, o mesmo foi dividido em 5 capítulos, dispostos da seguinte forma:

Capítulo 1 – O conceito do trabalho é introduzido, assim como um resumo da revisão bibliográfica é apresentado.

Capítulo 2 – Desenvolvimento do projeto mecânico, incluindo a apresentação dos conceitos utilizados para a elaboração dos componentes, modelagem dinâmica

dos mesmos e simulação, explicação de como é feita a movimentação do robô, ou seja, seu *gait*, além de comentar as etapas de fabricação das peças que compõem o protótipo de conceito.

Capítulo 3 – Utilização de redes neurais para obter um aprendizado computacional da movimentação do robô e dados sobre a otimização de sua velocidade. O capítulo é composto por uma breve introdução sobre a teoria de redes neurais, sua aplicação no trabalho e, por fim, uma explicação sobre o funcionamento do algoritmo utilizado.

Capítulo 4 – Mostra os testes feitos no protótipo de conceito para obtenção dos dados a serem aplicados na rede neural, além de discutir os resultados.

Capítulo 5 – Conclusão do trabalho e dos experimentos realizados, além de indicar propostas para uma futura continuação no projeto.

2

Projeto Mecânico

Neste capítulo o projeto mecânico é desenvolvido, desde a etapa dos conceitos fundamentais para a ideia do robô, passando pela sua modelagem, até a fabricação do protótipo de conceito funcional para ser testado posteriormente.

2.1

Fundamentos Teóricos

2.1.1

Pneumática

Pneumática é a tecnologia que utiliza os movimentos e fenômenos dos gases para atuar algum componente através da pressão do ar, exercendo trabalho. Antes de ser utilizado, o ar deve ser preparado: compressão, filtração, secagem, armazenamento e distribuição, simplificadamente representam algumas das etapas de preparação a que o ar é submetido até estar pronto para o uso de maneira adequada.

A pneumática, assim como outras tecnologias de atuação, possui vantagens e desvantagens [37]. Dentre suas vantagens, destacam-se:

- Como principal característica, apresenta altas velocidades de atuação.
- Seu elemento básico de funcionamento, o ar, não é limitado.
- É uma forma de energia limpa, portanto não gera poluição ao meio ambiente.
- Não necessita de reservatórios com grande resistência, devido à pressão de trabalho geralmente ser baixa (e.g. 6 bar). Consequentemente, os elementos pneumáticos têm construção

simplificada quando comparados a componentes similares de outras tecnologias, devido à utilização de baixas pressões.

- Pode trabalhar em temperaturas elevadas, uma vez que o ar tem poucas alterações em suas propriedades nestas condições necessárias para as tarefas.

Por outro lado, sistemas pneumáticos apresentam algumas desvantagens:

- Necessita de boa preparação do ar para não ter problemas de manutenção dos seus elementos.
- Não apresenta forças de trabalho elevadas, quando comparado com sistemas hidráulicos que normalmente trabalham a pressões muito mais elevadas.
- Se não contar com silenciadores em seus elementos, gera muitos ruídos.
- A compressibilidade do ar pode prejudicar o desempenho de técnicas de controle.

2.1.2 Pressão

Pressão, definida pela razão da força pela sua área de aplicação, pode ser calculada por:

$$P = \frac{F}{A} \quad (2.1)$$

Os gases têm por característica a tendência de se expandir infinitamente, porém o espaço é delimitado de acordo com o volume definido inicialmente. Quando o ar ocupa todo este volume de confinamento, tende a continuar sua expansão, exercendo pressão com mesma intensidade em todas as direções neste tal volume, como mostrado na Figura 2.1.

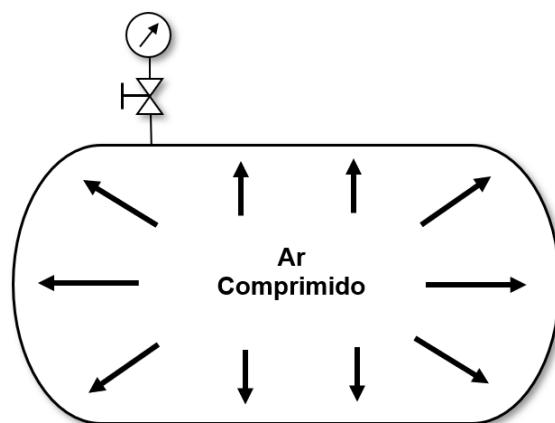


Figura 2.1: Ar comprimido em um volume confinado. [37]

Ao mesmo tempo, os gases apresentam capacidade de serem comprimidos com facilidade ao terem seu volume de confinamento diminuído ou quando o recipiente recebe um acréscimo de gás sem ter seu volume alterado.

Existem duas formas de se quantificar a pressão, podendo ser esta absoluta ou relativa. A pressão absoluta é medida a partir do ponto de zero absoluto (vácuo absoluto), enquanto a pressão relativa é medida a partir da pressão atmosférica.

2.1.3 Vácuo

Em termos práticos na pneumática, considera-se vácuo como uma pressão significativamente menor do que a pressão atmosférica. Quando existe um recipiente dito sob vácuo, a força resultante nas paredes deste recipiente é dada pela força exercida pela pressão externa (pressão atmosférica, por exemplo), subtraída da força causada pela pressão interna, que é assumidamente menor. Logo, a força resultante é do meio externo sobre o recipiente.

Sua geração geralmente é feita através de bombas de vácuo ou válvulas geradoras de vácuo que utilizam o princípio do tubo de *Venturi*. Porém ambos os métodos não são considerados no projeto proposto, o primeiro devido à sua forma construtiva ser complexa e o segundo pela alta vazão de ar demandada.

Outra forma de gerar vácuo em determinado corpo consiste em aumentar seu volume sem deixar que haja a adição de ar do meio. Essa forma será adotada neste

trabalho.

2.1.4

Atuadores Pneumáticos

Atuadores pneumáticos são elementos mecânicos capazes de transformar a energia do ar comprimido em expansão em trabalho, através de energia mecânica. Em geral, os atuadores pneumáticos podem ser lineares ou rotativos.

Neste trabalho, pretende-se utilizar atuadores lineares para gerar os movimentos do robô, tanto na fase de translação, quanto na fase de fixação e afastamento do corpo do robô em relação à superfície. Os atuadores rotativos não se encaixam neste projeto, por terem sua eficiência atrelada a processos como abertura e fechamento de comportas, por exemplo.

Neste trabalho, são abordados os dois tipos mais simples de atuadores pneumáticos lineares para fins de explicação de funcionamento: cilindros de simples ação e de dupla ação.

Partindo do princípio que estes cilindros têm seus movimentos de atuação alterando apenas o sentido em uma mesma direção, pode-se dizer que os atuadores de simples ação funcionam em um dos sentidos através da força gerada pelo ar comprimido, enquanto seu retorno à posição original depende de uma mola ou alguma força externa. A Figura 2.2 mostra de forma simplificada um atuador pneumático linear de simples ação em corte, para melhor compreensão do seu funcionamento.

Neste caso, quando o ar comprimido é injetado na entrada do cilindro, a pressão no êmbolo aumenta até que a força sobre o mesmo vença a força da mola, fazendo com que o conjunto êmbolo e haste se desloquem. Quando o ar comprimido deixa de ser fornecido e a conexão de entrada fica aberta para a atmosfera, a força da mola se torna a resultante e o conjunto êmbolo e haste volta para sua posição inicial. Este tipo de atuador pode ser normalmente recuado, como o representado na Figura 2.2, mas também pode ser normalmente avançado, dependendo de sua forma construtiva.

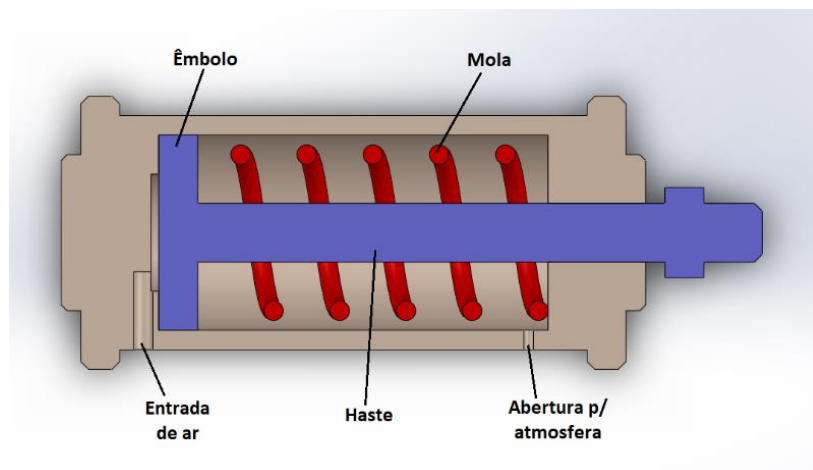


Figura 2.2: Atuador pneumático linear de simples ação.

Os atuadores pneumáticos lineares de dupla ação, por outro lado, precisam de duas entradas/saídas de ar em sua forma construtiva. O movimento destes tipos de cilindros é dado pela diferença de pressão entre as duas câmaras de ar em que o êmbolo está envolvido. Quando há injeção de ar comprimido em uma das duas entradas e a outra está aberta para a atmosfera, a força resultante atuando sobre o êmbolo faz com que haja a translação do êmbolo e da haste. A Figura 2.3 mostra de forma simplificada um atuador pneumático linear de dupla ação em corte. Este tipo de cilindro é recomendado para aplicações onde seja necessária a aplicação de força em suas duas fases de atuação, tanto no avanço quanto no retorno.

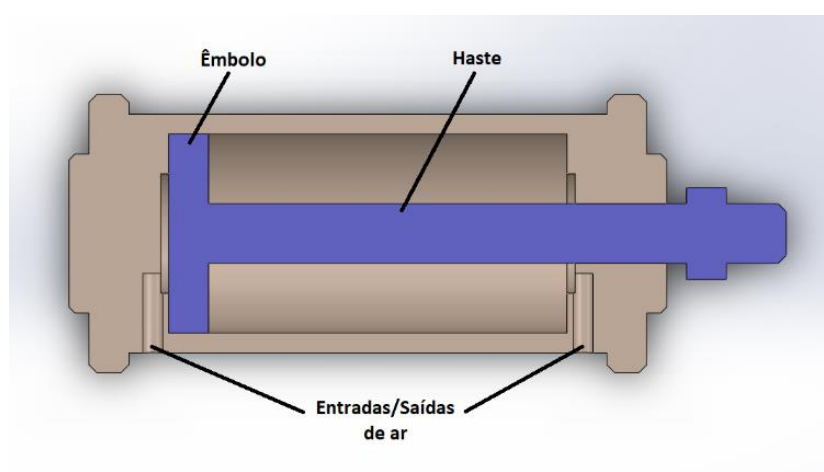


Figura 2.3: Atuador pneumático linear de dupla ação.

Neste trabalho, os atuadores utilizados são todos de simples ação, por terem um gasto energético menor, já que utilizam o ar comprimido apenas na fase de

avanço, e também por não haver a necessidade de exercer força nas fases de recuo dos cilindros utilizados, como será apresentado nas próximas seções.

2.2

Desenvolvimento do Projeto

Como já mencionado no Capítulo 1, o objetivo desse trabalho é desenvolver um robô com juntas prismáticas que seja capaz de se mover em superfícies lisas com diferentes inclinações em relação ao solo. Um desenho da montagem do robô “lagartixa” proposto é mostrado na Figura 2.4.

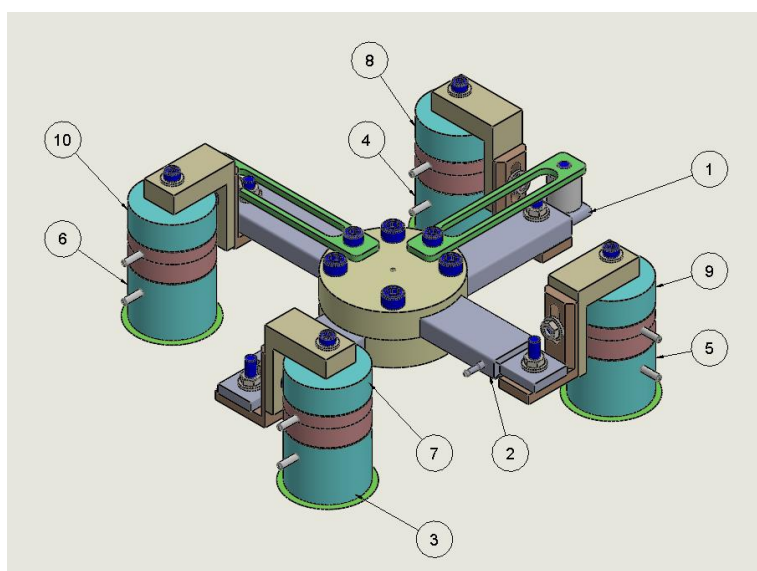


Figura 2.4: Desenho da montagem do robô.

A figura, além de mostrar a montagem geral da lagartixa, também indica a numeração dos atuadores e elementos, para que haja o entendimento da distribuição de cada item no conjunto, como descrito a seguir.

Os atuadores 1 e 2 são os cilindros principais do sistema, responsáveis pela translação de todo o corpo. Esses dois cilindros são comerciais, do tipo antigiro, da *Festo®* [45].

Os itens 3 a 6 são cilindros de fabricação própria, que constituem as garras. Estes atuadores têm por finalidade gerar vácuo mecanicamente para que o robô possa se prender à superfície desejada. Estes atuadores são explicados com mais

profundidade em uma seção futura, haja vista que são um dos principais componentes deste projeto.

Os itens 7 a 10 são os atuadores responsáveis por levantar o corpo da lagartixa durante a translação, para não haver atrito em relação à superfície pelo corpo e garras que estão desligadas neste momento. Estes cilindros são chamados de distanciadores, e também são de fabricação própria.

Os desenhos do projeto proposto foram desenvolvidos no software *SolidWorks®*. A ideia inicial seria fabricar todos os componentes em alumínio, porém para simplificação do desenvolvimento e redução de custos do protótipo optou-se pela fabricação da maioria das peças em polímero PLA (poliácido láctico) em uma impressora 3D (tridimensional). Apenas as peças internas onde ocorrem deslizamento, como os êmbolos e as camisas dos cilindros, foram fabricadas em alumínio.

No Apêndice A encontra-se o desenho detalhado das garras e dos distanciadores em sua versão em alumínio, apesar de a fabricação ter sido em PLA. No Apêndice B são mostrados todos os demais desenhos desenvolvidos para a construção do protótipo de conceito, desde os desenhos de conjunto até os desenhos de fabricação. Estes desenhos contam com as alterações feitas nas peças para a impressão 3D, assim como a especificação dos outros materiais utilizados.

Para explicação e entendimento da movimentação do robô e de como funciona cada passo dentro de um ciclo de movimentação em cada eixo, foi gerado um diagrama trajeto-passo, vide Figura 2.5. Note que os números dos itens na figura são correspondentes aos da Figura 2.4.

No diagrama da Figura 2.5, cada linha vertical representa o número de passos dentro de um ciclo de movimentação, enquanto as linhas horizontais indicam a numeração dos atuadores envolvidos no mecanismo analisado. Vale observar que, neste tipo de diagrama, existem duas linhas horizontais para cada atuador ou conjunto de atuadores, a linha inferior indica que o cilindro está na posição recuada, e a linha superior indica que está na posição avançada. Por exemplo, o atuador 1.0 está recuado nos passos 0, 1, 2, 8 e 9, enquanto está avançado nos passos 3, 4, 5, 6 e 7.

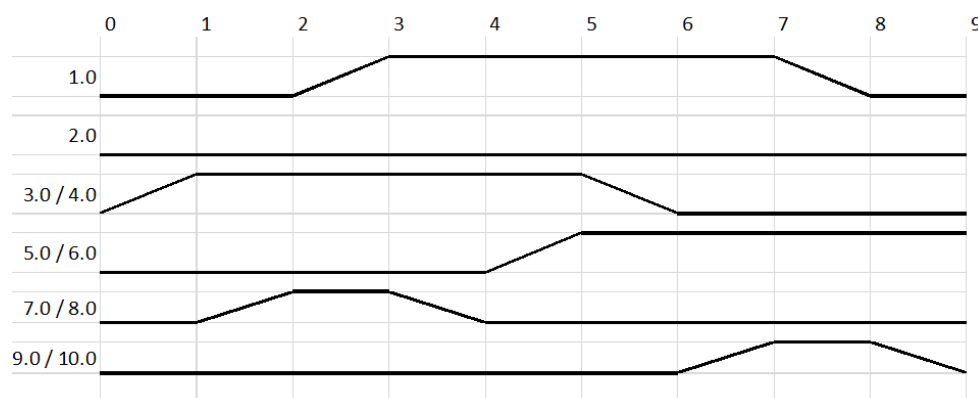


Figura 2.5: Diagrama trajeto-passo do robô.

No passo 0, que é o passo da inicialização, todos os atuadores estão recuados. Quando o comando de partida é acionado, os atuadores 3.0 e 4.0, que são garras, são acionadas para começarem a avançar. Em seguida, no passo 1 as garras dos atuadores 3.0 e 4.0 estão fixas na superfície.

O passo 2 consiste no momento em que os distanciadores 7.0 e 8.0 já ergueram o corpo da lagartixa, para assim ser iniciada a translação do mesmo com o avanço do cilindro 1.0.

No passo 3, a translação do corpo da lagartixa já ocorreu, com o avanço completo do atuador 1.0. Esta movimentação desloca inclusive os conjuntos dos atuadores 2.0, 3.0, 4.0, 9.0 e 10.0, além da peça central redonda de ligação de todos os subconjuntos.

Com a translação concluída, os atuadores 7.0 e 8.0 voltam às posições de recuados no passo 4, fazendo com que o restante dos componentes entre em contato novamente com a superfície, ou seja, os distanciadores abaixam o corpo da lagartixa novamente, como no posicionamento original.

No passo 5 as garras 5.0 e 6.0 se fixam à superfície também. Desta forma, as 4 garras do robô estão presas neste momento. As garras 3.0 e 4.0 então podem se soltar e voltar às suas posições recuadas, como pode ser visto no passo 6.

No passo 7, os distanciadores 9.0 e 10.0 erguem todo o corpo novamente, como preparativo para a finalização de um ciclo. Já no passo 8, o cilindro principal 1.0 termina seu recuo, levando o corpo do robô a ficar compacto novamente, como em seu posicionamento original.

Finalmente, o passo 9 é o responsável por aproximar novamente o corpo da lagartixa da superfície em que a mesma está se prendendo. Isso acontece com o recuo dos atuadores 9.0 e 10.0.

Caso seja desejado movimentar o robô durante apenas um ciclo em um dos eixos, a movimentação termina desta forma. Porém, se for desejada uma movimentação contínua, após estes passos mostrados, as garras 3.0 e 4.0 são acionadas novamente e em seguida, as garras 5.0 e 6.0 se soltam da superfície. Quando estas etapas são concluídas, os acionamentos voltam a acontecer a partir do passo 1.

2.3

Modelagem do Sistema Pneumático

Para modelar o sistema pneumático estudado, deve-se considerar alguns conceitos importantes para encontrar a equação fundamental que servirá de base para cada componente, como discutido a seguir.

O primeiro ponto importante a ser estudado é o fluxo de ar através de um pequeno orifício, uma vez que neste tipo de sistema este tipo de “obstáculo” está constantemente presente, seja nas válvulas, conexões ou nos próprios atuadores, dependendo de suas formas construtivas.

A Figura 2.6 exemplifica o fluxo de ar comprimido entre dois volumes de controle, através de um pequeno orifício ou área de estrangulamento em determinado componente pneumático. Este fluxo pode ser “limitado” ou “ilimitado”. O escoamento é dito “limitado” quando atinge condições sônicas na garganta.

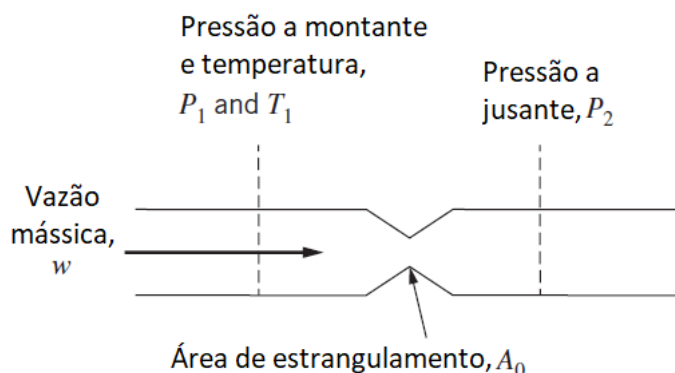


Figura 2.6: Fluxo de ar através de pequeno orifício. [43]

A maneira de calcular se este fluxo é “limitado” ou “ilimitado” consiste em comparar a razão entre as pressões a jusante e a montante com uma razão crítica, cuja equação é dada por:

$$C_r = \left(\frac{2}{\gamma + 1} \right)^{\frac{\gamma}{\gamma - 1}} \quad (2.2)$$

Com isso, para calcular a vazão mássica do fluxo de gás, têm-se:

- “Ilimitado”:

$$w = C_d A_0 P_1 \sqrt{\frac{\gamma}{RT_1} C_r^{\frac{\gamma+1}{\gamma}}} \text{ se } P_2/P_1 \leq C_r \quad (2.3)$$

- “Limitado”:

$$w = C_d A_0 P_1 \sqrt{\frac{2\gamma}{(\gamma - 1)RT_1} \left[\left(\frac{P_2}{P_1} \right)^{\frac{2}{\gamma}} - \left(\frac{P_2}{P_1} \right)^{\frac{\gamma+1}{\gamma}} \right]} \text{ se } P_2/P_1 > C_r \quad (2.4)$$

Outro conceito importante para a modelagem de um sistema pneumático é chamado de capacitância pneumática, que mede a capacidade de armazenamento de energia de um equipamento pneumático. Este conceito pode ser definido pela razão entre as variações de massa e pressão:

$$C = \frac{dm}{dP} \quad (2.5)$$

Desenvolvendo a equação e aplicando a lei dos gases ideais, obtêm-se:

$$C = \frac{V}{nRT} \quad (2.6)$$

Também podem-se separar as variáveis na equação inicial e dividir por dt para obter a equação diferencial:

$$C\dot{P} = \dot{m} = w \quad (2.7)$$

Aplicando os conceitos anteriores e partindo do princípio da continuidade de massa:

$$w_{VC} = \sum w_{ent} - \sum w_{sai} \quad (2.8)$$

obtem-se o modelo fundamental do sistema pneumático:

$$\dot{P} = \frac{nRT}{V} \left(w_{liq} - \frac{P}{RT} \dot{V} \right) \quad (2.9)$$

Com isso, pode-se aplicar este modelo nos componentes do robô lagartixa pneumática projetado que, como descrito anteriormente, conta com 2 atuadores comerciais da *Festo®* e 8 cilindros de fabricação própria, todos de simples ação com retorno por mola. Além disso, todas as válvulas utilizadas são 5-2 vias comuns, portanto nenhuma delas se enquadra na categoria proporcional, simplificando o modelo.

2.3.1

Modelo do Reservatório

Como o reservatório de ar comprimido não possui nenhuma parte mecânica móvel, seu modelo necessita apenas da formulação pneumática. Logo, seu modelo matemático fica simples quando comparado com outras partes ativas do sistema. Utilizando a Equação 2.9, temos:

$$\dot{P}_R = \frac{nRT}{V_R} w_{Rliq} \quad (2.10)$$

$$w_{Rliq} = w_C - \sum w_{atu} \quad (2.11)$$

2.3.2

Modelo dos Cilindros Principais

Para achar o modelo matemático completo de um atuador pneumático linear, deve-se considerar também a sua movimentação mecânica, através do diagrama de corpo livre mostrado na Figura 2.7, resultando em

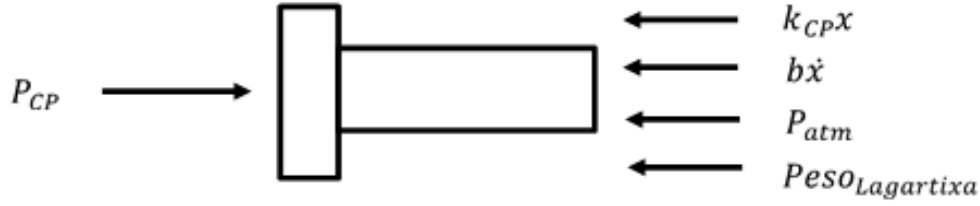


Figura 2.7: Diagrama de corpo livre de um cilindro principal.

$$\sum F = 0 \quad (2.12)$$

$$P_{CP}A_{CP} = P_{atm}A_{CP} + k_{CP}x_{CP} + b\dot{x}_{CP} + m\ddot{x}_{CP} + \text{Peso}_{Lagartixa} \quad (2.13)$$

Isolando \ddot{x}_{CP} e substituindo $\text{Peso}_{Lagartixa} = mg$:

$$\ddot{x}_{CP} = \frac{P_{CP}A_{CP}}{m} - \frac{P_{atm}A_{CP}}{m} - \frac{k_{CP}x_{CP}}{m} - \frac{b\dot{x}_{CP}}{m} - g \quad (2.14)$$

Para a parte pneumática do modelo, utiliza-se a Equação 2.9, obtendo:

$$\dot{P}_{CP} = \frac{nRT}{V_{CP0} + A_{CP}x_{CP}} \left(w_{CP} - \frac{P_{CP}}{RT} A_{CP} \dot{x}_{CP} \right) \quad (2.15)$$

Para obter a vazão mássica na câmara de cada cilindro principal, é necessário fazer a análise em duas situações distintas, quando a válvula é comutada para o

atuador avançar, e quando o sinal na válvula é zero. Em ambas situações, as Equações 2.2, 2.3 e 2.4 servem de referência, resultando em:

- Válvula comutada:

$$w_{CP} = C_d A_{CP0} P_R \sqrt{\frac{\gamma}{RT} C_r^{\frac{\gamma+1}{\gamma}}} \text{ se } P_{CP}/P_R \leq C_r \quad (2.16)$$

$$w_{CP} = C_d A_{CP0} P_R \sqrt{\frac{2\gamma}{(\gamma-1)RT} \left[\left(\frac{P_{CP}}{P_R} \right)^{\frac{2}{\gamma}} - \left(\frac{P_{CP}}{P_R} \right)^{\frac{\gamma+1}{\gamma}} \right]} \quad (2.17)$$

$\text{se } P_{CP}/P_R > C_r$

- Válvula sem sinal (aberta para atmosfera):

$$w_{CP} = C_d A_{CP0} P_{CP} \sqrt{\frac{\gamma}{RT} C_r^{\frac{\gamma+1}{\gamma}}} \text{ se } P_{atm}/P_{CP} \leq C_r \quad (2.18)$$

$$w_{CP} = C_d A_{CP0} P_{CP} \sqrt{\frac{2\gamma}{(\gamma-1)RT} \left[\left(\frac{P_{atm}}{P_{CP}} \right)^{\frac{2}{\gamma}} - \left(\frac{P_{atm}}{P_{CP}} \right)^{\frac{\gamma+1}{\gamma}} \right]} \quad (2.19)$$

$\text{se } P_{atm}/P_{CP} > C_r$

2.3.3

Modelo das Garras

O modelo matemático das garras deste projeto, que também são atuadores lineares de simples ação, deve considerar suas partes mecânicas móveis, através do diagrama de corpo livre mostrado na Figura 2.8, resultando em:

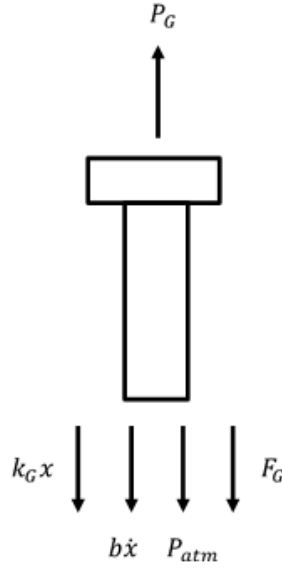


Figura 2.8: Diagrama de corpo livre de uma garra.

$$\sum F = 0 \quad (2.20)$$

$$P_G A_G = P_{atm} A_G + k_G x_G + b \dot{x}_G + m_G \ddot{x}_G + F_G \quad (2.21)$$

Isolando \ddot{x}_{CP} :

$$\ddot{x}_G = \frac{P_G A_G}{m_G} - \frac{P_{atm} A_G}{m_G} - \frac{k_G x_G}{m_G} - \frac{b \dot{x}_G}{m_G} - \frac{F_G}{m_G} \quad (2.22)$$

Para a parte pneumática do modelo, utiliza-se a Equação 2.9:

$$\dot{P}_G = \frac{nRT}{V_{G0} + A_G x_G} \left(w_G - \frac{P_G}{RT} A_G \dot{x}_G \right) \quad (2.23)$$

Assim como na modelagem dos cilindros principais, existem duas situações a serem estudadas: quando a válvula é comutada para o atuador avançar, e quando

o sinal na válvula é zero. Então, novamente, utilizam-se as Equações 2.2, 2.3 e 2.4 como referência, obtendo:

- Válvula comutada:

$$w_G = C_d A_{G0} P_R \sqrt{\frac{\gamma}{RT} C_r^{\frac{\gamma+1}{\gamma}}} \text{ se } P_G/P_R \leq C_r \quad (2.24)$$

$$w_G = C_d A_{G0} P_R \sqrt{\frac{2\gamma}{(\gamma-1)RT} \left[\left(\frac{P_G}{P_R} \right)^{\frac{2}{\gamma}} - \left(\frac{P_G}{P_R} \right)^{\frac{\gamma+1}{\gamma}} \right]} \quad (2.25)$$

se $P_G/P_R > C_r$

- Válvula sem sinal (aberta para atmosfera):

$$w_G = C_d A_{G0} P_G \sqrt{\frac{\gamma}{RT} C_r^{\frac{\gamma+1}{\gamma}}} \text{ se } P_{atm}/P_G \leq C_r \quad (2.26)$$

$$w_G = C_d A_{G0} P_G \sqrt{\frac{2\gamma}{(\gamma-1)RT} \left[\left(\frac{P_{atm}}{P_G} \right)^{\frac{2}{\gamma}} - \left(\frac{P_{atm}}{P_G} \right)^{\frac{\gamma+1}{\gamma}} \right]} \quad (2.27)$$

se $P_{atm}/P_G > C_r$

2.3.4

Modelo dos Distanciadores

Para o modelo dos distanciadores, leva-se em consideração o diagrama de corpo livre mostrado na Figura 2.9, resultando em:

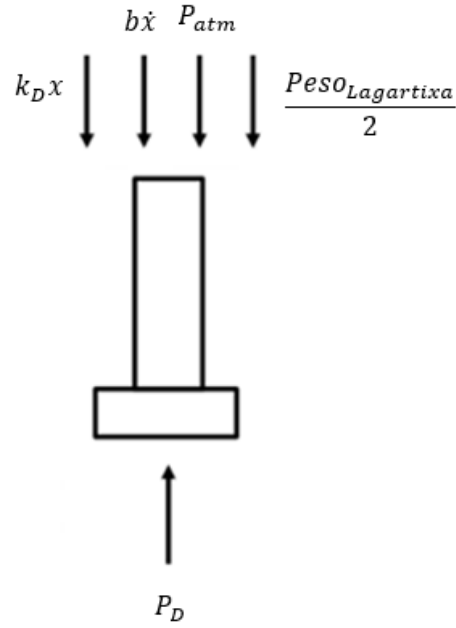


Figura 2.9: Diagrama de corpo livre de um distanciador.

$$\sum F = 0 \quad (2.28)$$

$$P_D A_D = P_{atm} A_D + k_D x_D + b \dot{x}_D + m_D \ddot{x}_D + Peso_{Lagartixa} \quad (2.29)$$

Isolando \ddot{x}_L substituindo $Peso_{Lagartixa} = mg$, além de $m_D = m/2$:

$$\ddot{x}_D = \frac{2P_D A_D}{m} - \frac{2P_{atm} A_D}{m} - \frac{2k_D x_D}{m} - \frac{2b \dot{x}_D}{m} - g \quad (2.30)$$

Para a parte pneumática do modelo, utiliza-se a Equação 2.9:

$$\dot{P}_D = \frac{nRT}{V_{D0} + A_D x_D} \left(w_D - \frac{P_D}{RT} A_D \dot{x}_D \right) \quad (2.31)$$

Para este tipo de cilindro, também é necessária a aplicação das duas situações já mostradas anteriormente, quando a válvula é comutada para o atuador avançar, e quando o sinal na válvula é zero. Então, utilizam-se as Equações 2.2, 2.3 e 2.4 como referência:

- Válvula comutada:

$$w_D = C_d A_{D0} P_R \sqrt{\frac{\gamma}{RT} C_r^{\frac{\gamma+1}{\gamma}}} \text{ se } P_D/P_R \leq C_r \quad (2.32)$$

$$w_D = C_d A_{D0} P_R \sqrt{\frac{2\gamma}{(\gamma-1)RT} \left[\left(\frac{P_D}{P_R} \right)^{\frac{2}{\gamma}} - \left(\frac{P_D}{P_R} \right)^{\frac{\gamma+1}{\gamma}} \right]} \quad (2.33)$$

$\text{se } P_D/P_R > C_r$

- Válvula sem sinal (aberta para atmosfera):

$$w_D = C_d A_{D0} P_D \sqrt{\frac{\gamma}{RT} C_r^{\frac{\gamma+1}{\gamma}}} \text{ se } P_{atm}/P_D \leq C_r \quad (2.34)$$

$$w_D = C_d A_{D0} P_D \sqrt{\frac{2\gamma}{(\gamma-1)RT} \left[\left(\frac{P_{atm}}{P_D} \right)^{\frac{2}{\gamma}} - \left(\frac{P_{atm}}{P_D} \right)^{\frac{\gamma+1}{\gamma}} \right]} \quad (2.35)$$

$\text{se } P_{atm}/P_D > C_r$

2.4 Simulação

Após a modelagem matemática dinâmica ser concluída, as equações são implementadas no ambiente *Simulink*, vide Apêndice C, a fim de gerar gráficos para

a observação do comportamento dos atuadores, assim como seus tempos de avanço e recuo.

Na simulação a seguir, utiliza-se a diferença entre tempos de disparos dos sinais para as válvulas iguais a 0,4s. A Figura 2.10 mostra o gráfico dos sinais gerados, com *gait* baseado no diagrama trajeto-passo da Figura 2.5.

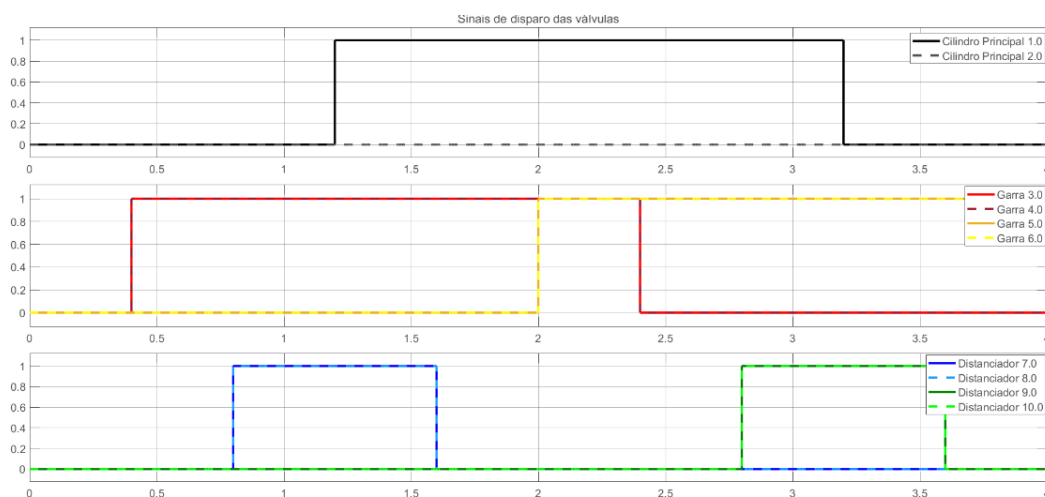


Figura 2.10: Sinais de disparos das válvulas.

As Figuras 2.11, 2.12 e 2.13 mostram, respectivamente, os deslocamentos dos êmbolos dos cilindros principais, das garras e dos distanciadores projetados, tomando como base os sinais da Figura 2.10. O comportamento exibido pelos atuadores é útil no capítulo seguinte, onde o protótipo passa por testes e necessita de tempos de atuação base para serem implementados na captura de dados.

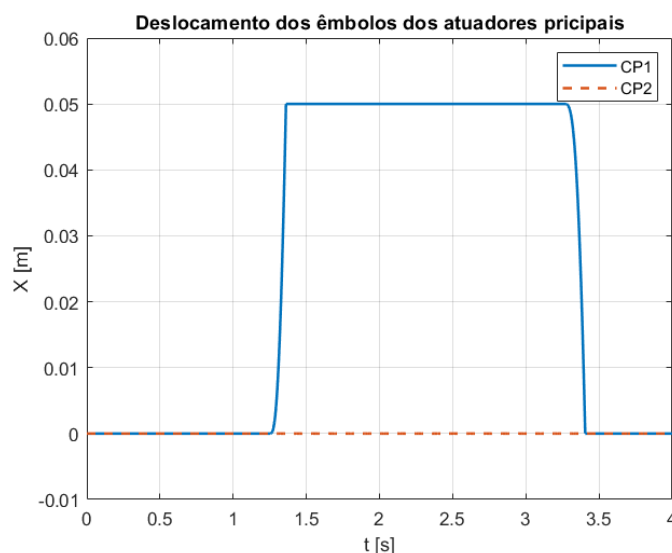


Figura 2.11: Deslocamentos dos êmbolos dos cilindros principais.

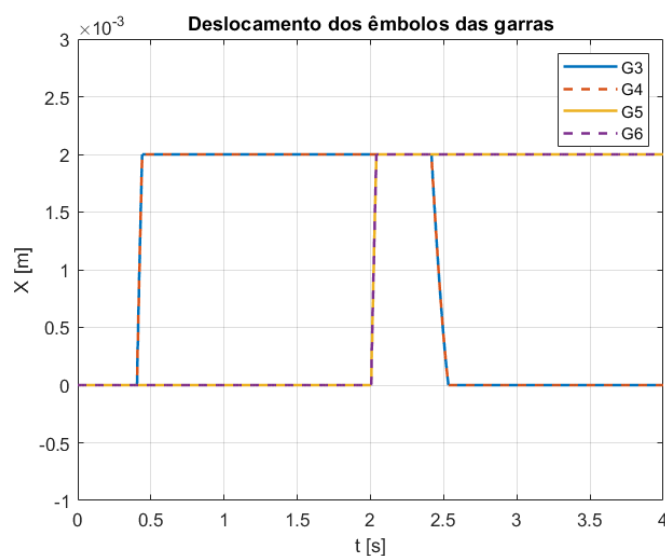


Figura 2.12: Deslocamentos dos êmbolos das garras.

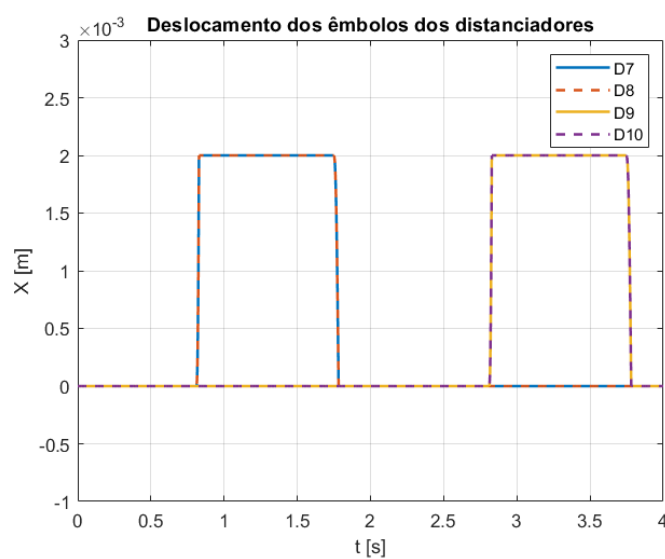


Figura 2.13: Deslocamentos dos êmbolos dos distanciadores.

2.5

Conceito das Garras

Uma das partes mais importantes do projeto mecânico do sistema proposto consiste no desenvolvimento das garras pneumáticas. Como a filosofia desse

projeto requer um dispositivo com economia energética, ou seja, que consuma menos ar comprimido, garras comerciais típicas não são uma boa escolha, pois exigem fluxo contínuo de ar, como discutido a seguir.

Em geral, quando é necessária a utilização de garras pneumáticas, utilizam-se ventosas e válvulas geradoras de vácuo comerciais, porém estas funcionam através do conceito do tubo *Venturi*, como mostra a Figura 2.14. Quando um fluxo de ar é submetido a um estrangulamento em uma garganta, a velocidade do mesmo aumenta durante a passagem por esta área menor, entretanto sua pressão neste momento diminui. As válvulas comerciais são dimensionadas de forma que a pressão neste momento de mudança de seção seja menor que a pressão atmosférica, desta forma o vácuo relativo é gerado.

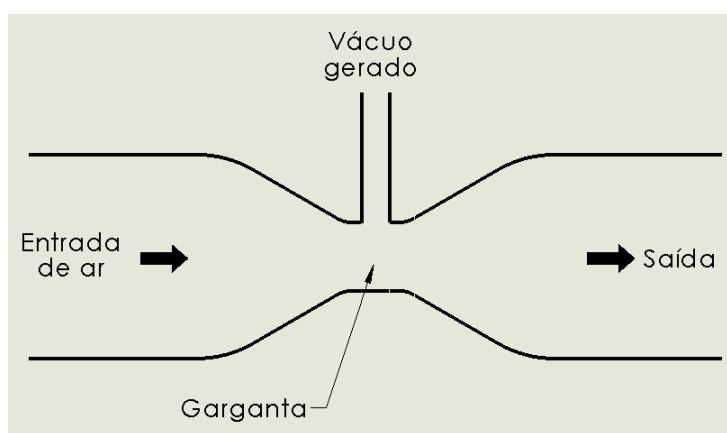


Figura 2.14: Tubo *Venturi*.

O problema deste tipo de geração de vácuo é a necessidade de haver um fluxo constante de ar comprimido através da válvula, enquanto se deseja ter o vácuo sendo gerado, acarretando um consumo energético maior. O Anexo A mostra os gráficos de consumo de ar comprimido de uma das fabricantes destas válvulas geradoras de vácuo. Pode-se observar que, a uma pressão de 6 *bar* (pressão padrão de aplicação em pneumática), o consumo mínimo, que varia conforme as dimensões do produto, é de aproximadamente 5,2 *l/min*. Isso significa que, se for exigido que a ventosa fique fixa a uma superfície durante 1 minuto, a válvula geradora de vácuo utilizará 5,2 litros de ar comprimido para executar sua função.

Como a ideia deste trabalho é projetar um robô que os gastos energéticos sejam baixos, opta-se por projetar uma garra que também utilize vácuo, entretanto com origem diferente. A Figura 2.15 mostra um esboço do conceito utilizado para

gerar o vácuo mecanicamente.

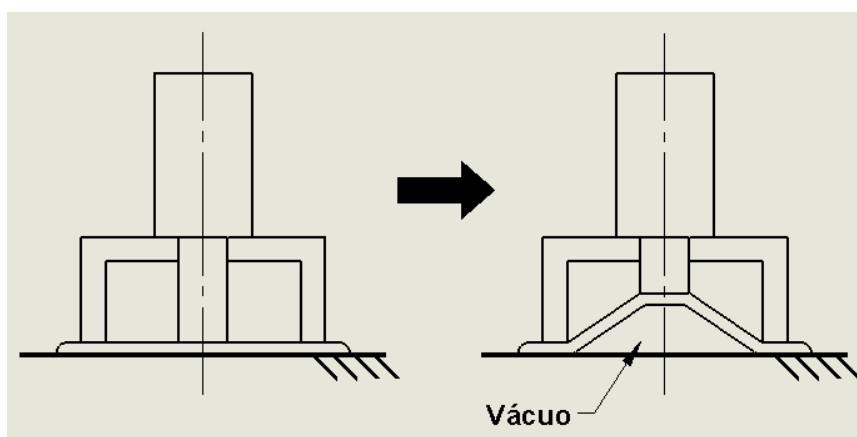


Figura 2.15: Esboço de geração de vácuo mecanicamente.

O funcionamento deste mecanismo consiste na variação do volume de controle entre uma borracha e superfície em que se deseja a fixação. Quando este volume de controle aumenta sem que a massa de ar seja alterada, ou seja, sem que haja a injeção ou retirada de ar deste volume, a pressão no mesmo diminui, gerando uma pressão menor do que a atmosférica. Como já descrito anteriormente, dentro deste volume de controle há um “vácuo relativo”, e a força resultante causada entre a pressão externa (atmosférica) e a interna faz com que a garra se fixe na superfície.

Antes de projetar todos os componentes da garra, é fabricado um pequeno protótipo da mesma para testar este conceito de vácuo mecânico, como pode ser visto na Figura 2.16.

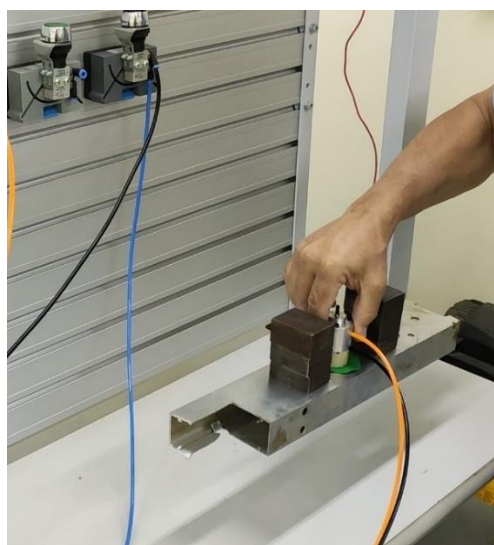


Figura 2.16: Protótipo de conceito da garra.

Após alguns testes com este protótipo de conceito, o projeto da garra foi desenvolvido para a versão final a ser implementada no robô. A Figura 2.17 apresenta o desenho em corte das garras e dos distanciadores. O conjunto de cima consiste nos distanciadores, que têm como função levantar o robô durante o movimento para não haver atrito com a superfície enquanto algumas partes da lagartixa pneumática se movem. O conjunto inferior da figura é o desenho das garras, cujo funcionamento já foi explicado.

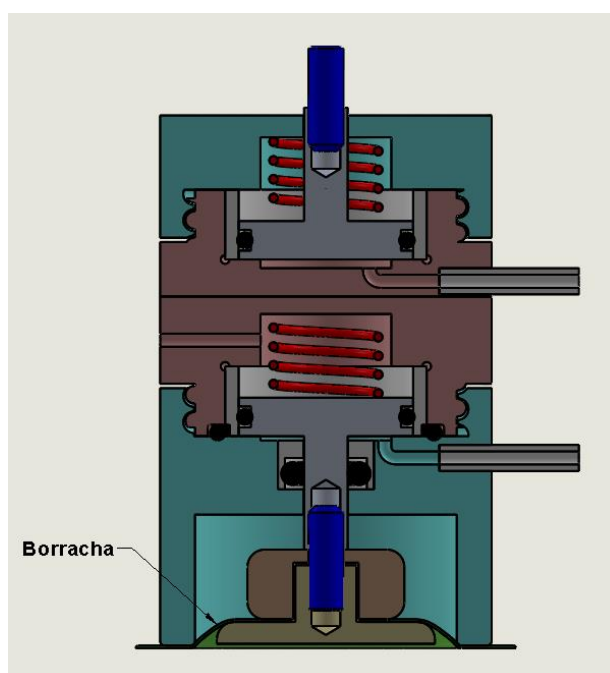


Figura 2.17: Projeto da garra e do distanciador.

Com este tipo de mecanismo, o gasto energético para gerar o vácuo e mantê-lo fica sendo apenas relativo ao volume de ar necessário para deslocar o êmbolo inferior, isto é:

$$V_G = A_G L_G \quad (2.36)$$

$$V_G = \left(\frac{\pi(D_{eG}^2 - D_{hG}^2)}{4} \right) L_G \quad (2.37)$$

Os desenhos com as dimensões detalhadas das garras e dos distanciadores são mostrados no Apêndice B, como mencionado anteriormente. As dimensões necessárias para este cálculo são $D_{eG} = 20mm$, $D_{hG} = 5mm$ e $L_G = 2mm$. Logo:

$$V_G = 589 \text{ mm}^3 \quad \therefore \quad V_G = 589 \cdot 10^{-6} \text{ l} \quad (2.38)$$

Isso significa que a fixação desta garra exigirá apenas $589 \cdot 10^{-6} \text{ l}$ de ar comprimido, independente do intervalo de tempo de fixação. Ou seja, não importa o tempo de duração de uso do vácuo, o gasto energético não mudará enquanto o atuador não for recuado e acionado novamente.

No próximo capítulo, técnicas de Redes Neurais serão discutidas para o acionamento do sistema experimental proposto.

3

Redes Neurais

De acordo com Lin & Lee [49], redes neurais são sistemas de processamento de sinais com capacidade de aprendizado e generalização a partir de bancos de dados fornecidos, existindo diversas maneiras de serem implementadas. As redes neurais são inspiradas nas redes de neurônios dos cérebros humanos, por isso também são chamadas de redes neurais artificiais.

As redes neurais fazem parte de um subconjunto de aprendizado de máquina construído para a realização de diversas tarefas, como organização e separação de dados, aproximação de funções, aprendizado de comportamentos através de padrões, dentre outras.

Estes sistemas contam com diversas estruturas e formas de funcionamento distintas, podendo variar o número de camadas e de neurônios artificiais presentes, a maneira com que estes neurônios estão conectados e processam as informações, os algoritmos de treinamento e aprendizado, etc.

3.1. Neurônio Artificial e Função de Ativação

Os neurônios artificiais básicos são os responsáveis pela parte de processamento de sinais, através de determinadas funções de ativação. Eles recebem os valores das entradas interligados a pesos antes de processá-los, gerando saídas que podem ser repassadas para outros neurônios ou não. Cada camada de uma rede neural pode ser composta por diversos neurônios. A Figura 3.1 apresenta de forma esquemática um neurônio de uma rede neural artificial para compreensão da função de ativação. Nesta figura, x_m são as entradas, y_i são as saídas do neurônio e w_{im} representam os pesos (força das sinapses) entre o neurônio anterior ou entrada j com o neurônio de destino i .

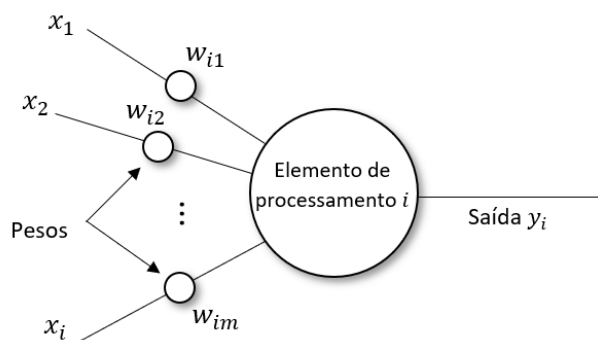


Figura 3.1: Representação esquemática de um neurônio. [49]

Para este trabalho, é escolhida a função de ativação sigmoide unipolar, que dá como resposta valores entre 0 e 1, com sua grande maioria ficando nos limites, como visto na Figura 3.2. Esta função foi escolhida devido à sua capacidade de oferecer valores variados entre os limites, diferente de uma função degrau, por exemplo. Logo, caso uma determinada saída não seja calculada com um valor exato 0 ou 1, pode-se classificar tal cálculo desta rede como não confiável.

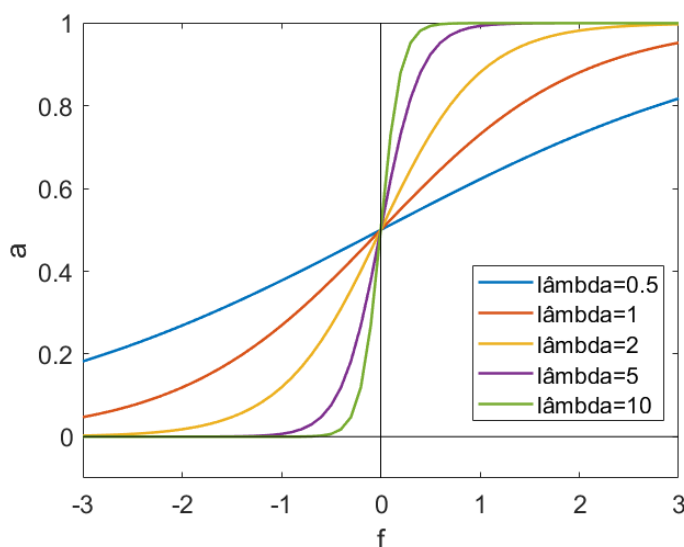


Figura 3.2: Função de ativação sigmoide unipolar. [49]

A equação que define a função de ativação sigmoide unipolar é dada por:

$$a(f) = \frac{1}{1 + e^{-\lambda f}} \quad (3.1)$$

$$f_i = \sum_{j=1}^m w_{ij}x_j - \theta_i \quad (3.2)$$

onde $\lambda > 0$ na Equação 3.1 determina a inclinação da curva próximo do ponto $f = 0$ e θ_i representa o limite do elemento de processamento (ou neurônio) i .

3.2

Aplicação da Rede Neural

Uma rede neural pode ter variadas quantidades e estruturas de arrumação de camadas. Estas variações classificam as redes neurais e são indicadas para diferentes aplicações. A Figura 3.3 mostra uma arrumação de rede neural genérica de múltiplas camadas de *feedforward*.

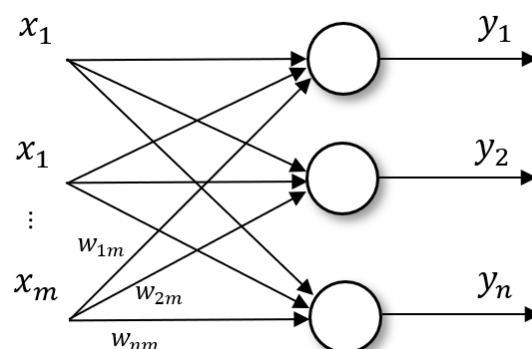


Figura 3.3: Estrutura de redes neurais de *feedforward*. [49]

Neste tipo de rede, os conjuntos de entradas x_n são injetados na primeira camada de neurônios para serem processadas, e os resultados são passados adiante para as próximas camadas de neurônios até chegar nas saídas y_n , não importando a quantidade de camadas ocultas presentes no sistema. Este tipo de rede neural de *feedforward* pode ter múltiplas camadas, ou apenas uma camada de neurônios.

Além das redes de *feedforward*, existem as redes neurais recorrentes. O que as diferencia é que, após o cálculo das saídas, é feito um cálculo dos erros para

atualizar a primeira camada da rede.

Além do tipo de estrutura, os aprendizados podem ser supervisionados ou não supervisionados. Em uma rede neural com aprendizagem supervisionada, os dados fornecidos para treinamento contemplam valores de saídas já definidos, logo a própria rede pode comparar os valores calculados com os valores desejados durante o treinamento, achando valores de erro. Por outro lado, quando uma rede neural utiliza aprendizagem não supervisionada, a própria rede precisa identificar e separar os dados em grupos de acordo com características em comum encontradas.

Neste trabalho, é utilizada aprendizagem supervisionada com um algoritmo de *backpropagation* modificado para aprender os movimentos da lagartixa pneumática, e otimizar os tempos de disparo dos sinais das válvulas pneumáticas.

Este tipo algoritmo trabalha com redes que utilizam múltiplas camadas em que os pesos se propagam adiante conforme os dados de entrada e funções de ativação, ou seja, como uma rede de *feedforward*. Chegando na camada de saída, compara os valores calculados com os valores de saída desejados e acha o valor do erro; se não for satisfatório, este valor é retro propagado para as camadas anteriores, para que, desta forma, atualizem seus pesos. Este ciclo continua até achar um erro satisfatório ou até chegar em um ponto em que o treinamento faça com que a rede comece a perder a capacidade de generalização.

Para a compreensão do funcionamento interno da rede utilizada neste trabalho, a Figura 3.4 apresenta de forma esquemática uma rede neural artificial que utiliza um algoritmo de *backpropagation*, onde x_m são as entradas, y_n são as saídas calculadas, d_n são as saídas desejadas e e_n são os erros calculados. O funcionamento do algoritmo acontece em alguns passos, descritos a seguir.

O passo 0 é a inicialização, onde se definem valores iniciais pequenos e aleatórios para os pesos, além de definir os limites desejados, como o valor do erro aceitável e limite de parada prematuro. Nesta rede especificada, coloca-se um limite de 6 épocas para uma parada prematura, ou seja, caso o aprendizado da rede piore seu desempenho por 6 épocas seguidas, o algoritmo pára de rodar e encerra o treinamento.

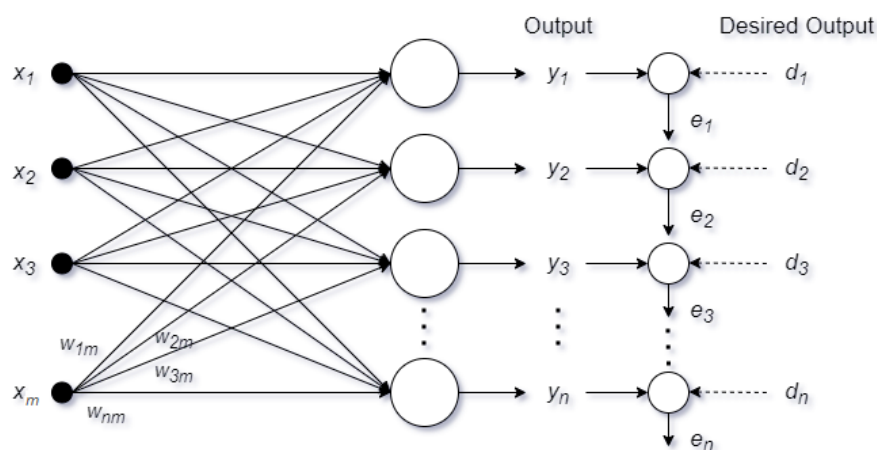


Figura 3.4: Rede neural de *backpropagation* [49].

No passo 1, aplica-se a matriz com os valores de entrada na primeira camada da rede, além do vetor com as saídas desejadas, valores estes obtidos por testes no protótipo de conceito. Já o passo 2, chamado de passo para frente, é responsável por propagar o sinal adiante entre cada camada conforme as funções de ativação dos neurônios. Após os dados terem passado por todas as camadas, a rede gera os valores de saída calculados.

Tendo estes valores computados, inicia-se o passo 3, onde se calcula a diferença entre a saída desejada e a obtida pela rede neural. O passo 4 é chamado de retro propagação onde, através de uma função de perda, atualizam-se os pesos das camadas anteriores. Esta atualização é feita também em cascata, ou seja, é feita de forma progressiva através de cada conexão utilizada no passo 2. Os passos 3 e 4 são realizados através do método de *Levenberg-Marquardt*, que é abordado na seção 3.3 deste trabalho.

Finalmente, no passo 5 é checado se o erro é aceitável. Se o erro obtido for menor que o erro máximo permitido, o problema termina, caso contrário volta-se ao passo 1 e inicia-se uma nova época.

3.3 Método de *Levenberg-Marquardt*

O algoritmo de *Levenberg-Marquardt* é utilizado para solução de problemas de mínimos quadrados não-lineares [50]. Ao ajustar uma função $y(\mathbf{x}; \mathbf{p})$ das variáveis \mathbf{x} e um vetor de parâmetro \mathbf{p} (que é relacionado com os pesos das conexões entre neurônios) a um conjunto de m dados, é conveniente minimizar a

soma dos quadrados ponderados dos erros entre os dados medidos d_i e a função $y(\mathbf{x}; \mathbf{p})$, que é a função responsável pelo cálculo das saídas, dado por

$$f_{lm}(\mathbf{p}) = \sum_{i=1}^m \left[\frac{d_i - y(\mathbf{x}_i; \mathbf{p})}{\eta_i} \right]^2 \quad (3.3)$$

onde η_i são os erros de medição. Como será explicado no Capítulo 4, neste trabalho desconsidera-se este parâmetro de erro de medição, resultando em:

$$f_{lm}(\mathbf{p}) = (\mathbf{d} - \mathbf{y}(\mathbf{p}))^T (\mathbf{d} - \mathbf{y}(\mathbf{p})) \quad (3.4)$$

$$f_{lm}(\mathbf{p}) = \mathbf{d}^T \mathbf{d} - 2\mathbf{d}^T \mathbf{y}(\mathbf{p}) + \mathbf{y}(\mathbf{p})^T \mathbf{y}(\mathbf{p}) \quad (3.5)$$

Se a função $y(\mathbf{x}; \mathbf{p})$ é não-linear no modelo do parâmetro \mathbf{p} , sua minimização deve ser feita de forma iterativa:

$$\mathbf{p} := \mathbf{p} + \Delta \mathbf{p} \quad (3.6)$$

O objetivo de cada iteração é achar o valor de $\Delta \mathbf{p}$ que minimize a função f . Para achar este valor, utiliza-se a combinação do método do *Gradient Descent* com o método de Gauss-Newton, descritos a seguir.

3.3.1

Método do *Gradient Descent*

O método *gradient descent* é um método de minimização que atualiza os valores dos parâmetros na direção mais íngreme, i.e. na direção oposta ao gradiente da função objetivo, cujo valor em relação aos parâmetros é dado por:

$$\frac{\partial}{\partial \mathbf{p}} f = 2(d - y(\mathbf{p}))^T \frac{\partial}{\partial \mathbf{p}} (d - y(\mathbf{p})) \quad (3.7)$$

$$\frac{\partial}{\partial \mathbf{p}} f = -2(d - y(\mathbf{p}))^T \left[\frac{\partial y(\mathbf{p})}{\partial \mathbf{p}} \right] \quad (3.8)$$

$$\frac{\partial}{\partial \mathbf{p}} f = -2(d - y(\mathbf{p}))^T \mathbf{J} \quad (3.9)$$

onde $\mathbf{J} = \frac{\partial y(\mathbf{p})}{\partial \mathbf{p}} \in R^{m \times n}$ é a matriz Jacobiana associada ao problema. O parâmetro de atualização na direção de descida mais íngreme é dado por:

$$\Delta \mathbf{p}_{gd} = -\gamma_{gd} \mathbf{J}^T (d - y(\mathbf{p})) \quad (3.10)$$

onde γ_{gd} é o comprimento do passo.

3.3.2

Método de *Gauss-Newton*

Este método é utilizado para minimizar uma função objetivo de soma de quadrados. Considera-se que a função objetivo é aproximadamente quadrática nos parâmetros próximos da solução ótima. A função avaliada pode ser aproximada localmente através de uma expansão em série de Taylor de primeira ordem:

$$y(\mathbf{p} + \Delta \mathbf{p}) \approx y(\mathbf{p}) + \left[\frac{\partial y(\mathbf{p})}{\partial \mathbf{p}} \right] \Delta \mathbf{p} = y(\mathbf{p}) + \mathbf{J} \Delta \mathbf{p} \quad (3.11)$$

Substituindo a aproximação na Equação 3.5:

$$f(\mathbf{p} + \Delta\mathbf{p}) \approx d^T d - 2d^T y(\mathbf{p}) + y(\mathbf{p})^T y(\mathbf{p}) - 2(d - y(\mathbf{p}))^T J\Delta\mathbf{p} + (J\Delta\mathbf{p})^T J\Delta\mathbf{p} \quad (3.12)$$

Igualando-se $\frac{\partial f}{\partial \Delta\mathbf{p}} = 0$, obtêm-se:

$$\frac{\partial}{\partial \Delta\mathbf{p}} f(\mathbf{p} + \Delta\mathbf{p}) \approx -2(d - y(\mathbf{p}))^T J + 2(J\Delta\mathbf{p})^T J = 0 \quad (3.13)$$

Logo, o parâmetro de atualização é obtido através da equação:

$$[J^T J]\Delta\mathbf{p}_{gn} = J^T (d - y(\mathbf{p})) \quad (3.14)$$

3.3.3

Algoritmo de *Levenberg-Marquardt*

O algoritmo de Levenberg-Marquardt combina os métodos do *Gradient Descent* e de *Gauss-Newton* para obter o parâmetro de atualização ou parâmetro de minimização:

$$[J^T J + \lambda_{lm} I]\Delta\mathbf{p}_{lm} = J^T (d - y(\mathbf{p})) \quad (3.15)$$

onde λ_{lm} é o parâmetro de amortecimento responsável pela magnitude do comprimento do passo, enquanto I é uma matriz identidade.

Como a rede neural artificial utilizada neste trabalho é a fornecida pela *toolbox* do *MATLAB*®, opta-se por utilizar os valores dos parâmetros predefinidos pelo *software*, como o valor de λ_{lm} , por exemplo, desde seu valor inicial até possíveis variações aplicadas automaticamente durante o processamento.

3.4

Especificações da Arquitetura da Rede Neural

Além da utilização da função de ativação sigmoide unipolar para o processamento dos dados, e do método de *Levenberg-Marquardt* para a parte da função de custo e retro propagação, a rede neural utilizada através da *toolbox* do *MATLAB*® conta com duas camadas ocultas em sua estrutura, sendo estas de 5 neurônios cada. O Apêndice D mostra o *script* de implementação de uma rede neural com um algoritmo de *backpropagation*, através de bibliotecas do *MATLAB*®.

O programa conta com um conjunto de dados obtidos através de testes no protótipo de conceito, explicado no Capítulo 4. Além disso, seu método de treinamento é definido por *Levenberg-Marquardt*, através de um algoritmo de *backpropagation* modificado.

No Apêndice E é mostrado o *script* da fase de validação da rede neural. Após o treinamento, uma nova base de dados também obtidos com experimentos no protótipo de conceito é aplicada e comparada com valores de saída calculados pela rede. Assim como o método de obtenção dos dados, os resultados obtidos no treinamento e na validação da rede neural criada são mostrados no Capítulo 4.

4

Resultados Experimentais

Com o projeto mecânico desenvolvido e o conceito de aplicação da rede neural compreendido, diversos testes são feitos no protótipo para a obtenção de uma base de dados e implementação computacional na rede neural artificial. Posteriormente, são feitas análises dos resultados de movimentação do robô para verificar a confiabilidade da rede e o grau de otimização obtido durante o processo.

4.1

Experimentos Realizados

Partindo do princípio que a estrutura de rede neural utilizada para este trabalho exige dados de entrada e saída, os parâmetros utilizados para a geração do conjunto de dados são os tempos de acionamento das válvulas eletropneumáticas e o funcionamento ou não do mecanismo durante a movimentação do robô.

Como mencionado, os tempos de acionamento do sistema são utilizados como as entradas da rede. Ao todo são 10 atuadores presentes no sistema, porém alguns sempre funcionam simultaneamente, como foi explicado na Figura 2.5. Portanto, em cada conjunto de valores de entradas, temos 6 variáveis distintas, ou seja, 6 tempos de disparo de sinais para as válvulas do sistema, válvulas estas que são as acionadoras dos atuadores pneumáticos.

Para o conjunto de dados não ficar com valores viciados, utilizam-se valores aleatórios para os tempos de acionamento das válvulas, porém valores estes que fiquem dentro de limites especificados com base na simulação dinâmica.

No total são gerados 250 conjuntos de tempos de disparo aleatórios entre $-0,1$ e $0,5$ s. Estes tempos representam o intervalo de tempo entre o sinal de cada válvula, isso significa que entre cada passo mostrado inicialmente na Figura 2.5 pode haver esta variação de tempo indicada. No caso de valores negativos, o diagrama trajeto-

passo mostrado é alterado.

É importante trabalhar com essa variação de tempo também negativa, alterando o algoritmo de movimentação do robô, haja vista que dependendo da aplicação pode se exigir um *gait* diferente do proposto inicialmente.

Com os conjuntos de valores de entradas já definidos, fez-se a montagem do sistema pneumático e elétrico para executar os testes no protótipo de conceito. O Apêndice F mostra o esquema pneumático e elétrico, apresentando os equipamentos e acessórios utilizados para estes experimentos, assim como as instruções da montagem utilizada.

Para aplicar estes tempos no protótipo, utiliza-se um CLP (controlador lógico programável), onde a inserção das entradas é programada manualmente para cada conjunto de entradas. O programa desenvolvido para executar o movimento desejado através dos tempos de disparo utilizando o CLP está disponível no Apêndice G.

Após a montagem de todo o sistema e programação do controlador, os testes utilizando os 250 conjuntos de tempos de disparo aleatórios foram efetuados, utilizando diversas inclinações da superfície em relação ao solo, como mostra a Figura 4.1 e o Apêndice H.

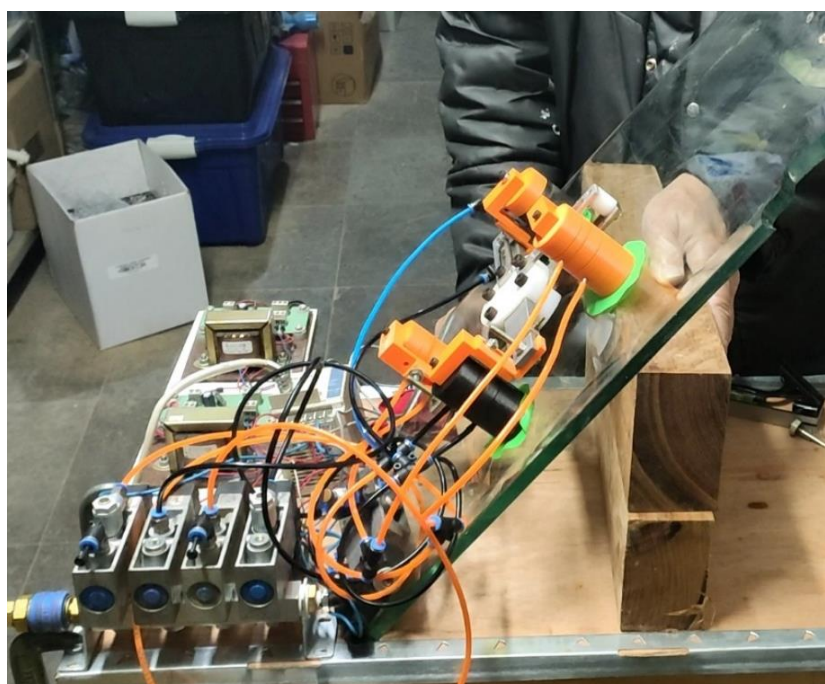


Figura 4.1: Teste do robô com inclinação.

Conforme os testes são feitos com o protótipo de conceito, as saídas são registradas manualmente sob supervisão humana. Caso o movimento programado do protótipo se complete sem problemas, o valor da saída para tal conjunto de entradas é anotado como 1. Se o movimento apresenta alguma falha, como deslizamento ou desprendimento total da superfície, o valor é anotado como 0. Após testar todos os 250 conjuntos de valores de entrada, tendo seus valores de saída vinculados, este conjunto de dados é aplicado à rede neural.

4.2 Resultados

Esta seção do trabalho é destinada a mostrar os resultados desde a parte do treinamento da rede neural até seu índice de acertos, além de um estudo da movimentação do robô através de testes com sensoramento.

Começando pelo treinamento da rede neural artificial, a Figura 4.2 mostra a média dos erros quadráticos durante as épocas do treinamento.

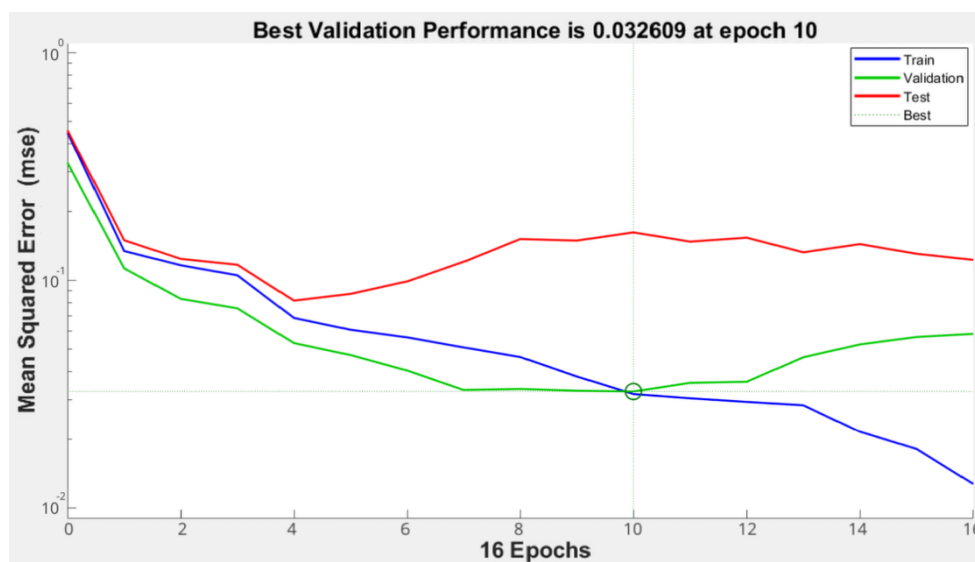


Figura 4.2: Treinamento da rede neural artificial.

Durante o treinamento, o algoritmo da *toolbox* do *MATLAB*® é executado em 3 fases: a primeira é a linha azul que representa o treinamento da rede propriamente dito; a linha vermelha representa os testes feitos com os próprios dados fornecidos para verificar o aprendizado da rede; e, por fim, a linha verde é a fase de validação,

responsável por verificar a capacidade de generalização da rede.

Neste treinamento, opta-se por um requisito de parada chamado *early stopping* ou parada precoce. Esta parada ocorre devido à curva de validação ter um aumento de erro em 6 épocas seguidas, o que faz com que a rede perca a capacidade de generalização.

Observa-se na Figura 4.2 que na época 16 as curvas de treinamento e de teste apresentam erros menores do que na época 10, porém, é importante esta parada precoce pois a curva de validação apresenta erros maiores depois deste ponto indicado como ponto de melhor resultado.

Com o treinamento da rede concluído, são gerados mais 25 conjuntos de entradas aleatórias com os valores na mesma grandeza dos 250 conjuntos iniciais, com o intuito de testar a eficiência da rede neural. Nesta fase, a rede neural calcula os valores de saída e compara com novos testes realizados no protótipo.

A Figura 4.3 apresenta o gráfico dos valores de saídas calculados entre 0 e 1 para os 25 conjuntos de valores utilizados nestes novos testes. A linha azul representa os resultados testados no protótipo de conceito, sendo 0 para falha e 1 para movimento correto completo, e a linha vermelha representa os valores calculados pela rede neural.

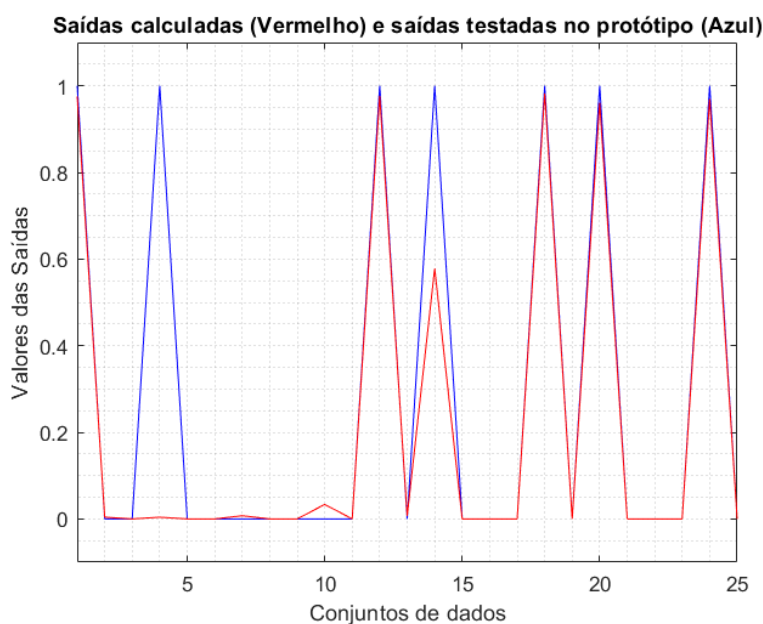


Figura 4.3: Teste de eficiência da rede neural.

Considera-se um erro tolerável máximo de 0,2 para os resultados gerados pela rede neural. Pelo gráfico da Figura 4.3 é possível observar quais são os valores fora da tolerância. Logo, as saídas calculadas para os conjuntos de entradas 4 e 14 são considerados errados, fazendo com que a rede neural tenha um índice de acerto de 92% (23 acertos e 2 erros) nesta amostragem.

Para verificar o funcionamento do robô, como seu descolamento e velocidade, experimentos utilizando sensores externos também são realizados. Para a obtenção destes dados, utiliza-se uma câmera *JeVois* [51] capaz de rastrear movimento de objetos atrelados a determinado padrão de cores ou píxeis. No Anexo B está disponível a parte de programação utilizada para a captura destes dados.

A Figura 4.4 apresenta o esquema montado para os testes de captura de dados com a câmera *JeVois*. Na imagem, a inclinação da superfície é 0° , porém também são realizados testes com inclinações e tempos de acionamento variados.

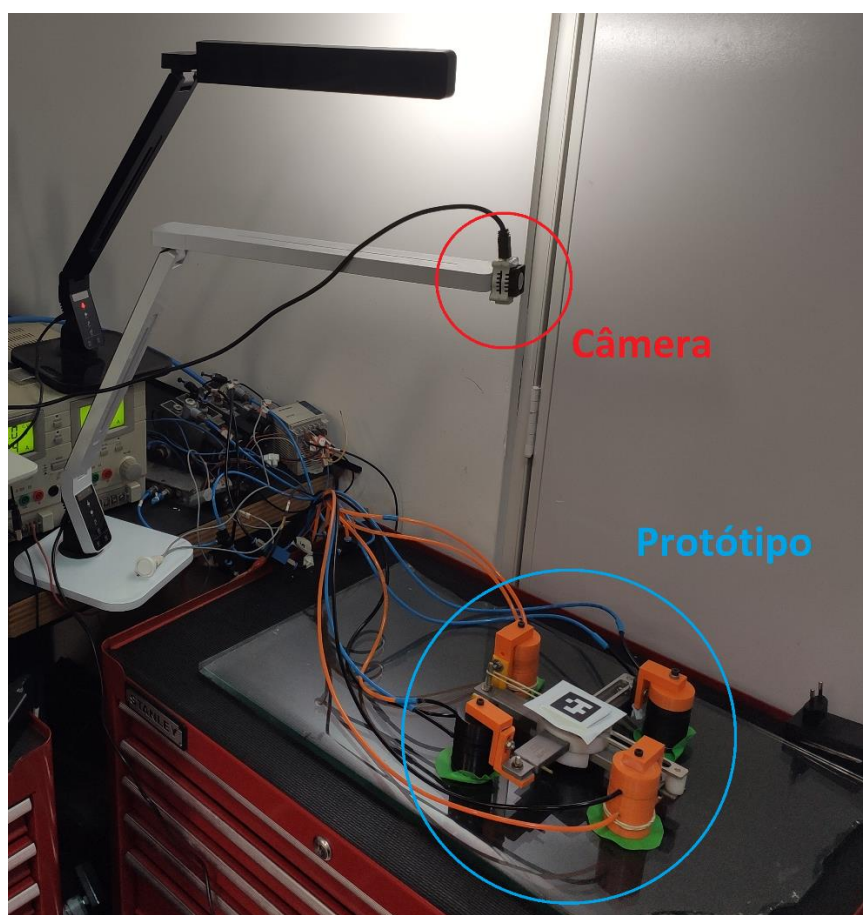


Figura 4.4: Esquema para testes com a *JeVois*.

Após a captura dos dados de movimentação da lagartixa pneumática, os dados passam por um pós-processamento no *software MATLAB®*, para a retirada de *outliers* e limpeza dos dados, para que enfim os gráficos possam ser gerados. No Apêndice I está disponível o código utilizado para esta tarefa.

A Figura 4.5 mostra o gráfico de deslocamento do robô em um dos eixos, utilizando um conjunto de tempos de disparo em que o robô não funciona adequadamente, havendo deslizamento no terceiro ciclo de movimento. Este é um claro exemplo de falha de movimento que culmina em uma saída igual a 0 no sistema da rede neural treinada. A execução deste teste foi em uma superfície inclinada e com as variações de tempo entre os disparos dos sinais sendo lentas.

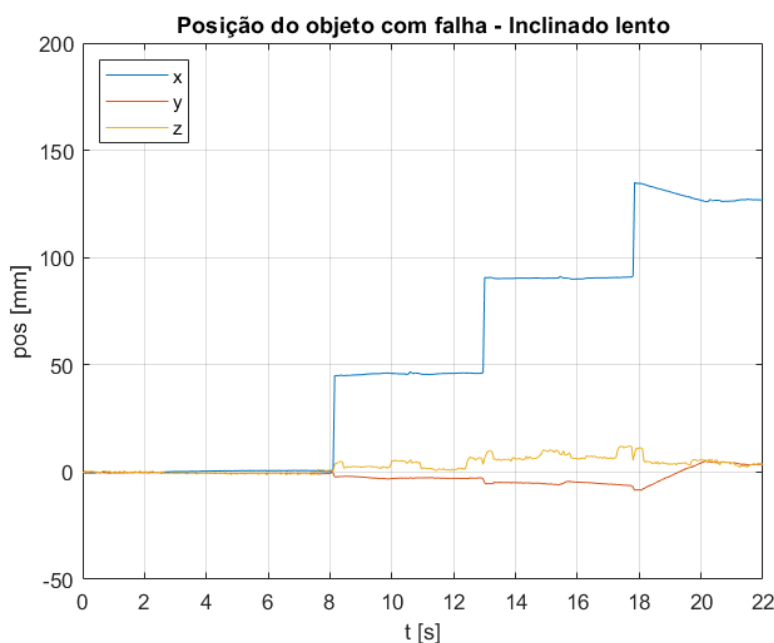


Figura 4.5: Gráfico de deslocamento com falha do robô.

Durante os experimentos com a câmera, comparações de velocidade entre os movimentos antes e depois de passar pela rede neural foram feitas. Nas Figuras 4.6 e 4.7 são mostrados os deslocamentos do robô em uma superfície sem inclinação de forma lenta e de forma rápida, respectivamente.

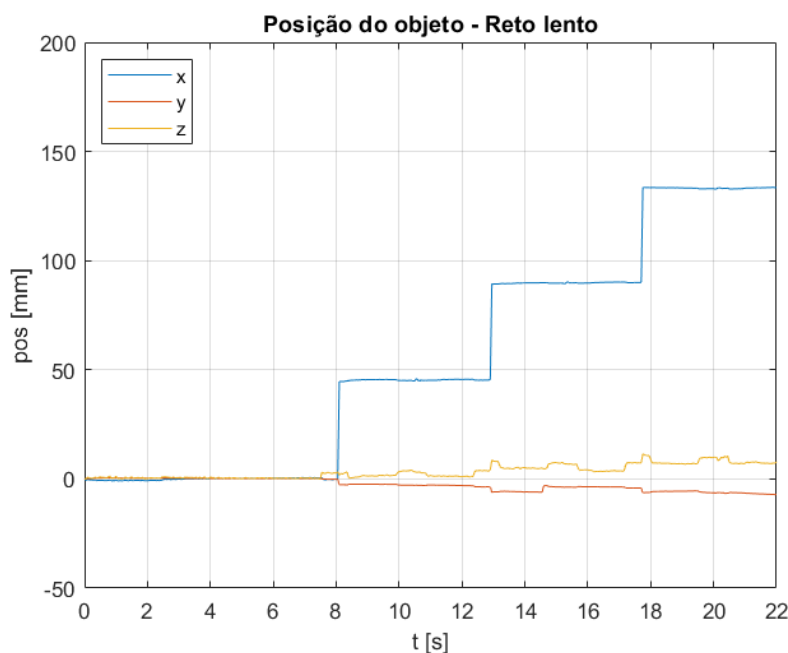


Figura 4.6: Deslocamento lento do robô em superfície sem inclinação.

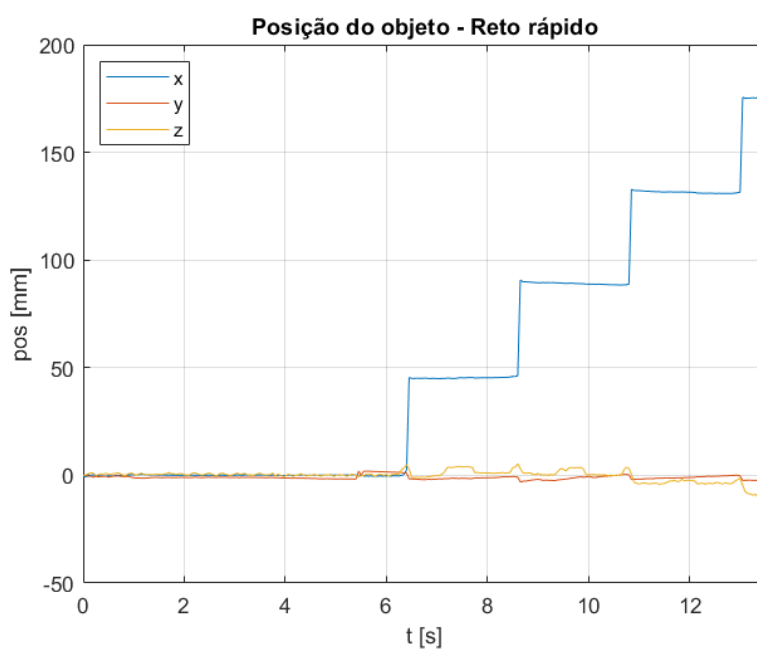


Figura 4.7: Deslocamento rápido do robô em superfície sem inclinação.

Pelos gráficos traçados, observa-se que o desempenho e velocidade da lagartixa pneumática foram melhorados com a rede neural. Utilizando dois ciclos completos como parâmetro de avaliação, na Figura 4.6 o primeiro passo é iniciado no segundo 8 e, após dois ciclos, o instante de tempo é 17,7s, ou seja, dois ciclos completos duraram 9,7s. Os tempos de disparo dos sinais das válvulas do sistema para a versão lenta foram valores padrão, com 0,5s de atraso entre cada passo do

gait do robô. A Figura 4.7 apresenta um acionamento mais rápido. O movimento é iniciado no instante 6,25s e, após dois passos, está no instante 10,75s. Logo, dois ciclos completos do movimento mais rápido duraram apenas 4,5s, cerca de 50% mais rápido do que o anterior.

Em outros experimentos, também são testados os mesmos valores de tempos de disparos, no entanto com a superfície inclinada em vez de horizontal. As Figuras 4.8 e 4.9 mostram os gráficos dos deslocamentos lento e rápido, respectivamente.

Estes dois gráficos de deslocamento em uma superfície inclinada seguem o mesmo padrão de movimentação e velocidade dos gráficos de deslocamento em superfície sem inclinação. Mesmo com as variações de inclinação da superfície, o desempenho da lagartixa pneumática é análogo em relação à sua velocidade. Com isso, pode-se afirmar que, em qualquer inclinação da superfície abaixo de 60° , o protótipo de conceito tem um desempenho padronizado de acordo com seus tempos de disparo programados, ou seja, a inclinação não é um fator influenciador na velocidade da lagartixa pneumática.

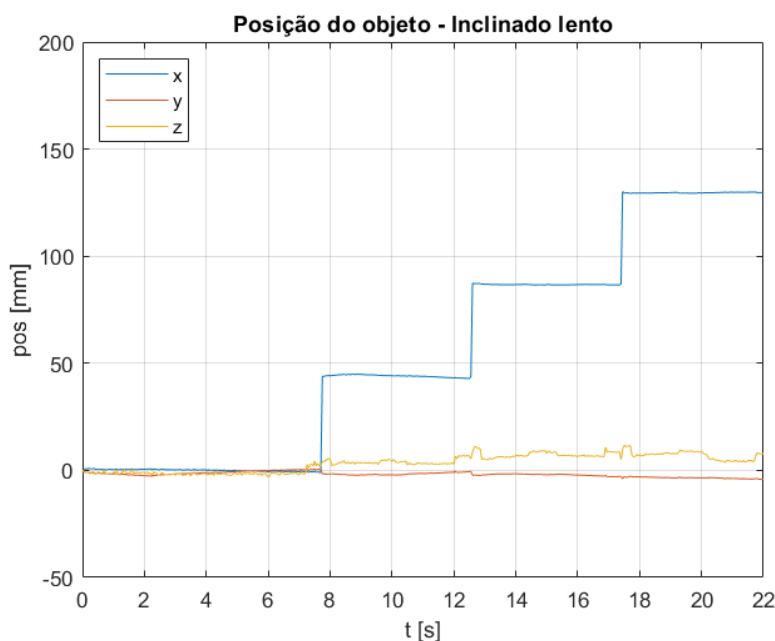


Figura 4.8: Deslocamento lento do robô em superfície com inclinação.

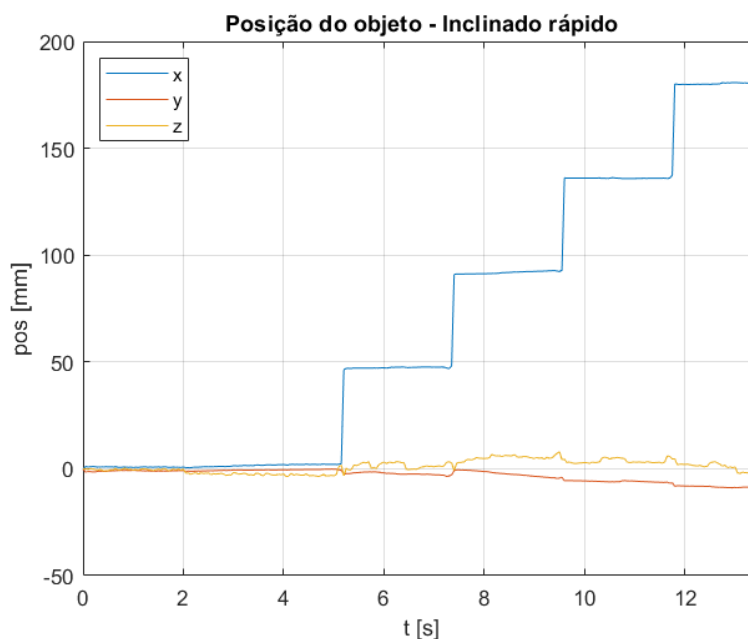


Figura 4.9: Deslocamento rápido do robô em superfície com inclinação.

No entanto, em todos os testes feitos com mais de 60° de inclinação, o robô se desprende entre os passos 5 e 6 do diagrama trajeto-passo mostrado na Figura 2.5, que é justamente a etapa em que se troca a fixação do robô na superfície. Esta falha na troca de fixação acontece porque os itens 1 e 2 do desenho MEM-01-04-001-4, do Apêndice B, funcionam como molas e possuem fixações articuladas, para poderem compensar qualquer problema de alinhamento das garras fabricadas em impressora 3D, além de ajudarem a compensar a deformação das borrachas utilizadas nas ventosas durante a troca de fixação entre os conjuntos de garras.

Quando há uma inclinação maior do que 60° , o peso das mangueiras que conduzem o ar comprimido até os atuadores já é suficiente para que haja uma rotação indesejada no ponto de articulação dos itens 1 e 2. Desse modo, no momento de troca de fixação, as garras até então livres não conseguem se alinhar com a superfície, e conseqüentemente não geram vácuo relativo suficiente entre a borracha e a superfície, fazendo o robô se desprender e cair.

Neste trabalho foi desenvolvido um projeto mecânico do robô lagartixa pneumática, capaz de se movimentar em superfícies lisas com diferentes inclinações, utilizando apenas atuadores pneumáticos lineares. No projeto mecânico, o foco foi o desenvolvimento das garras de vácuo, que são mais econômicas no consumo de ar comprimido quando comparadas com os conjuntos de geradoras de vácuo e ventosas presentes no mercado.

Através de uma rede neural, foi feito o aprendizado computacional da movimentação do robô para servir como uma simulação aproximada da funcionalidade real testada no protótipo de conceito. Essa otimização dos disparos permite que o robô seja usado em diferentes aplicações, podendo lidar com diferentes ordens de disparo dos sinais das válvulas que acionam o sistema de atuadores, dependendo da necessidade.

Utilizando uma amostragem de 250 conjuntos de valores de entradas para executar o treinamento da rede neural, testes posteriores foram feitos com mais 25 conjuntos de entradas para verificar a efetividade da rede nos cálculos de funcionamento do robô. Após os valores calculados serem comparados com os dados levantados com experimentos no protótipo, concluiu-se que a rede ficou com um índice de acertos de 92%, além de o robô ter tido um aumento de velocidade de mais de 50%, comparando a velocidade inicial com a mais rápida obtida na rede.

Trabalhos futuros sugeridos incluem o dimensionamento e projeto mecânico de um sistema de compressor e reservatório de ar comprimido para serem acoplados em seu corpo, fazendo com que o robô não necessite ter ligações com elementos externos. Além de um projeto mecânico e de acionamento embarcados, um outro desenvolvimento proposto envolveria um sistema de controle que permita que o robô seja autônomo, com auxílio de câmeras embarcadas ou sensores externos.

Para sequência do trabalho, sugere-se também uma simulação matemática completa do robô, incluindo a movimentação do mecanismo. Desta forma, poderá ser feita a comparação entre os resultados simulados através da modelagem matemática e os valores calculados pela rede neural.

Na parte de redes neurais, uma proposta futura é implementar outros algoritmos e métodos de aprendizado computacional, visando comparar os métodos existentes na literatura para gerar o melhor desempenho possível da rede neural no robô desenvolvido.

Referências Bibliográficas

- [1] SANTOS, DANIEL & HEYNEMAN, BARRETT & KIM, SANGBAE & ESPARZA, NOE & CUTKOSKY, MARK. (2018). **Gecko-inspired climbing behaviors on vertical and overhanging surfaces**. Proceedings – IEEE International Conference on Robotics and Automation, 1125-1131.
- [2] PROVANCHER, WILLIAM & JENSEN-SEGAL, SAMUEL & FEHLBERG, MARK (2011). **ROCR: An Energy-Efficient Dynamic Wall-Climbing Robot**. Mechatronics, IEEE/ASME Transactions on Mechatronics, 16, 897-906.
- [3] CHEN, RUI. (2015). **A Gecko-Inspired Electroadhesive Wall-Climbing Robot**. Potentials, IEEE, 34, 15-19.
- [4] MIR-NASIRI, NAZIM & SISWOYO JO, HUDYJAYA & ALI, MD. (2018). **Portable Autonomous Window Cleaning Robot**. Procedia Computer Science, 133, 197-204.
- [5] HOSSAIN, RIAD & CHISTY, NAFIZ. (2018). **Design and Implementation of a Wall Climbing Robot**. International Journal of Computer Applications, 179, 1-5.
- [6] SHI, KAIGE & LI, XIN (2020). **Vacuum suction unit based on the zero pressure difference method**. Physics of Fluids, 32. DOI: 10.1063/1.5129958.
- [7] MASURI, ARIEL & MEDINA, ODED & HACHOEN, SHLOMI & SHVALB, NIR. (2020). **Gait and Trajectory Optimization by Self-Learning for Quadrupedal Robots with an Active Back Joint**. Journal of Robotics. 2020. 1-7.
- [8] ALHARBI, ABDULLAH & EQUBAL, KAMRAN & AHMAD, SULTAN & RAHMAN, HASEEB & ALYAMI, HASHEM. (2021). **Human Gait**

Analysis and Prediction Using the Levenberg-Marquardt Method. Journal of Healthcare Engineering. 2021. 1-11.

- [9] KOIVIKKO, A. ET AL. (2021). **3D-Printed Pneumatically Controlled Soft Suction Cups for Gripping Fragile, Small, and Rough Objects.** Advanced intelligent systems, 3(9), pp.2100034-n/a.
- [10] TSUKAGOSHI, H. & OSADA, Y. (2021). **Soft Hybrid Suction Cup Capable of Sticking to Various Objects and Environments.** Actuators, 10(3), p.50.
- [11] LIU, J. ET AL. (2006). **Analytical modelling of suction cups used for window-cleaning robots.** Vacuum, 80(6), pp.593–598.
- [12] GE, D. ET AL. (2015). **Quantitative study on the attachment and detachment of a passive suction cup.** Vacuum, 116, pp.13–20.
- [13] TUVAYANOND, W. & PARNICHKUN, M. (2017). **Position control of a pneumatic surgical robot using PSO based 2-DOF H_{∞} loop shaping structured controller.** Mechatronics (Oxford), 43, pp.40–55.
- [14] HERZIG, N. ET AL. (2018). **Nonlinear position and stiffness Backstepping controller for a two Degrees of Freedom pneumatic robot.** Control engineering practice, 73, pp.26–39.
- [15] WANG, Z. ET AL. (2022). **Freeform Fabrication of Pneumatic Soft Robots via Multi-Material Jointed Direct Ink Writing.** Macromolecular materials and engineering, 307(4), p.n/a.
- [16] UGURLU, B. ET AL. (2019). **Stable Control of Force, Position, and Stiffness for Robot Joints Powered via Pneumatic Muscles.** IEEE transactions on industrial informatics, 15(12), pp.6270–6279.
- [17] LIU, Y. ET AL. (2021). **Three-dimensional loading control of a pneumatic three-universal–prismatic–universal robot.** International journal of advanced robotic systems, 18(5), p.172988142110449.
- [18] BAI, L. ET AL. (2017). **Kinematics, Dynamics, and Optimal Control of Pneumatic Hexapod Robot.** Mathematical problems in engineering, 2017, pp.1–16.

- [19] LIANG, D. ET AL. (2022). **Energy-Based Motion Control for Pneumatic Artificial Muscle Actuated Robots With Experiments.** IEEE transactions on industrial electronics (1982), 69(7), pp.7295–7306.
- [20] MAEDA, S. ET AL. (2012). **Development and Control of a Pneumatic Robot Arm for Industrial Fields.** International journal of advanced robotic systems, 9(3), p.59.
- [21] HAN, Q. ET AL. (2021). **Motion Control of a Gecko-like Robot Based on a Central Pattern Generator.** Sensors (Basel, Switzerland), 21(18), p.6045.
- [22] SHAO, D. ET AL. (2022). **A gecko-inspired robot with CPG-based neural control for locomotion and body height adaptation.** Bioinspiration & biomimetics, 17(3), pp.Bioinspiration & biomimetics, 2022, Vol.17 (3).
- [23] WANG, W., LIU, Y. & XIE, Z. (2021). **Gecko-Like Dry Adhesive Surfaces and Their Applications: A Review.** Journal of bionics engineering, 18(5), pp.1011–1044.
- [24] CHENGWEI HUANG ET AL. (2022). **Land–Air–Wall Cross-Domain Robot Based on Gecko Landing Bionic Behavior: System Design, Modeling, and Experiment.** Applied sciences, 12(8), p.3988.
- [25] HERNANDO, M. ET AL. (2021). **Behavior-Based Control Architecture for Legged-and-Climber Robots.** Applied sciences, 11(20), p.9547.
- [26] NANSAL, S. ET AL. (2021). **Design and Implementation of a Lizard-Inspired Robot.** Applied sciences, 11(17), p.7898.
- [27] VEGA-HEREDIA, M. ET AL. (2019). **Design and modelling of a modular window cleaning robot.** Automation in construction, 103, pp.268–278.
- [28] KROSURI, S.P. & MINOR, M.A. (2005). **Design, Modeling, Control, and Evaluation of a Hybrid Hip Joint Miniature Climbing Robot.** The International journal of robotics research, 24(12), pp.1033–1053.
- [29] ZHONG, Z. ET AL. (2021). **Design and Control of an Omnidirectional Mobile Wall-Climbing Robot.** Applied sciences, 11(22), p.11065.
- [30] LI, X. ET AL. (2018). **The Gait Design and Trajectory Planning of a Gecko-Inspired Climbing Robot.** Applied bionics and biomechanics, 2018,

pp.2648502–13.

- [31] YU, Z. ET AL. (2017). **Vertical Climbing Locomotion of a New Gecko Robot Using Dry Adhesive Material**. International Journal of Robotics and Automation, 32(4), pp.International Journal of Robotics and Automation, 2017, Vol.32 (4).
- [32] LABRUYÈRE, R. (2022). **Robot-assisted gait training: more randomized controlled trials are needed! Or maybe not?** Journal of neuroengineering and rehabilitation, 19(1), p.58.
- [33] BAOTHMAN, F.A. (2021). **A Machine Learning Approach for Improving the Movement of Humanoid NAO's Gaits**. Wireless communications and mobile computing, 2021, pp.1–14.
- [34] SHAO, Y. ET AL. (2022). **Learning Free Gait Transition for Quadruped Robots Via Phase-Guided Controller**. IEEE robotics and automation letters, 7(2), pp.1230–1237.
- [35] JI, Q. ET AL. (2022). **Synthesizing the optimal gait of a quadruped robot with soft actuators using deep reinforcement learning**. Robotics and computer-integrated manufacturing, 78, pp.Robotics and computer-integrated manufacturing, 2022, Vol.78.
- [36] YAMAMOTO, H. ET AL. (2020). **Generalization of movements in quadruped robot locomotion by learning specialized motion data**. ROBOMECH journal, 7(1), pp.1–14.
- [37] FIALHO, ARIVELTO BUSTAMANTE. **Automação Pneumática: Projetos, Dimensionamento e Análise de Circuitos** – 2. ed. - São Paulo: Érica, 2003.
- [38] FONTANELLA, JORGE LUIZ. **Apostila de Pneumática e Eletropneumática** – Petrópolis, RJ: [s.n.], 2017.
- [39] FESTO. **Basic Vacuum Technology Principles** – [S.I.: s.n.], [20--?]
- [40] Tecnologia de vácuo. FESTO. Disponível em: <https://www.festo.com/cms/pt-br_br/4292.htm> Acesso em: 25 mai. 2018.
- [41] JUVINALL, ROBERT C. **Fundamentos do Projeto de Componentes de**

Máquinas – 5. ed. – Rio de Janeiro: LTC, 2016.

- [42] COLLINS, JACK A. **Projeto Mecânico de Elementos de Máquinas: Uma Perspectiva de Prevenção da Falha** – 1. ed. – Rio de Janeiro: LTC, 2006.
- [43] KLUEVER, C. **Sistemas Dinâmicos: Modelagem, Simulação e Controle** – 1. ed – Rio de Janeiro: LTC, 2018.
- [44] SolidWorks version 2021. Waltham, Massachusetts: Dassault Systèmes SolidWorks Corporation, 2020.
- [45] Festo. Esslingen, Germany: Festo Corporate.
- [46] Festo. Vacuum Generators. Esslingen, Germany: Festo Corporate. Available at: <https://www.festo.com/media/pim/339/D15000100123339.PDF>
- [47] MATLAB version 9.10.0 (R2021a). Natick, Massachusetts: The MathWorks Inc, 2021.
- [48] The MathWorks, I. (2021). Deep Learning Toolbox. Natick, Massachusetts: The MathWorks Inc. Available at: <https://www.mathworks.com/help/deeplearning/>.
- [49] LIN, C.T., George Lee, C.S. **Neural Fuzzy Systems: A Neuro-Fuzzy Synergism to Intelligent Systems** – 1. ed. – New Jersey: Prentice Hall, 1996.
- [50] KIM, SEONGJAI. **Mathematical Foundations of Machine Learning** – Starkville, Mississippi: [s.n.], 2022.
- [51] JeVois Smart Machine Vision. Los Angeles, United States of America: JeVois Inc.
- [52] GOEBEL, M.R., FONTANELLA, J.L., MEGGIOLARO, M.A., QUADRELLI, G. (2021). **Project, Modeling and Application of Neural Networks in the Pneumatic Gecko Robot**. Simpósio Brasileiro de Automação Inteligente (SBAI), paper 216429, 17-20 de outubro de 2021.

Anexo A

Gráficos de Consumo de Ar – Vacuum Generators, Festo

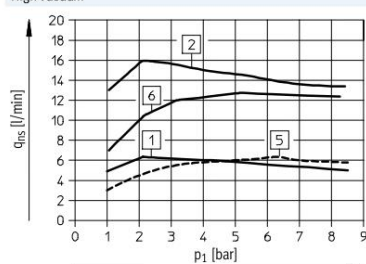
Vacuum generators VN

Technical data

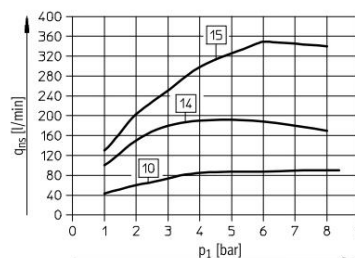
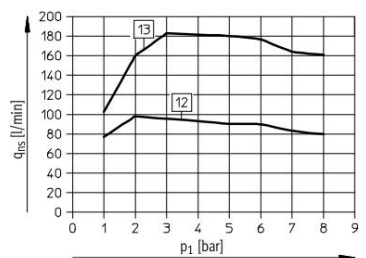
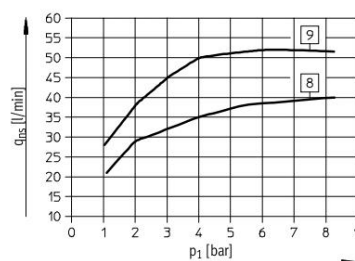
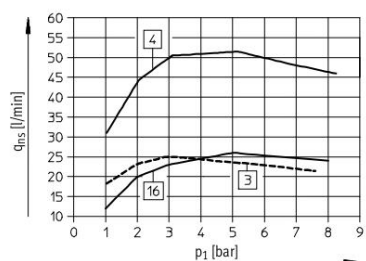
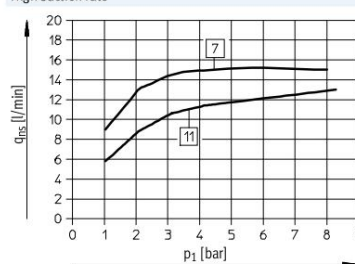
FESTO

Suction rate q_{ns} (with respect to atmosphere) as a function of operating pressure p_1

High vacuum



High suction rate



Standard:

- 1 VN-05-H...
- 2 VN-07-H...
- 3 VN-10-H...
- 4 VN-14-H...
- 12 VN-20-H...
- 13 VN-30-H...

Inline:

- 5 VN-05-M...
- 6 VN-07-M...
- 16 VN-10-M...

Standard:

- 7 VN-05-L...
- 8 VN-07-L...
- 9 VN-10-L...
- 10 VN-14-L...
- 14 VN-20-L...
- 15 VN-30-L...

Inline:

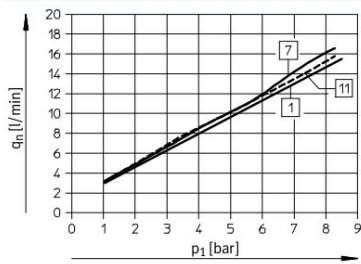
- 11 VN-05-N...

Vacuum generators VN

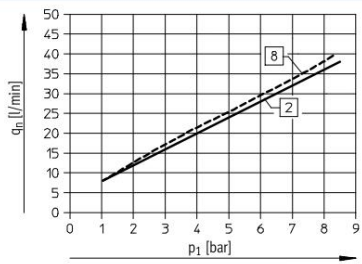
Technical data



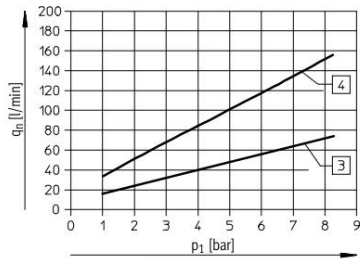
Air consumption q_n as a function of operating pressure p_1
High vacuum/high suction rate



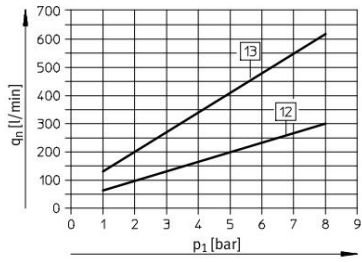
- | | |
|--------------|---------------|
| Standard: | Inline: |
| 1 VN-05-H... | 1 VN-05-M... |
| 7 VN-05-L... | 11 VN-05-N... |



- | | |
|--------------|--------------|
| Standard: | Inline: |
| 2 VN-07-H... | 2 VN-07-M... |
| 8 VN-07-L... | |



- | | |
|--------------|--------------|
| Standard: | Inline: |
| 3 VN-10-H... | 3 VN-10-M... |
| VN-10-L... | |
| 4 VN-14-H... | |
| VN-14-L... | |



- | |
|---------------|
| Standard: |
| 12 VN-20-H... |
| VN-20-L... |
| 13 VN-30-H... |
| VN-30-L... |

Anexo B

Programa de Captura da *JeVois*

Neste anexo é mostrado o programa feito na linguagem *Python*, cuja finalidade é apenas registrar os dados obtidos através da câmera *JeVois* durante os experimentos realizados. O algoritmo foi desenvolvido pela integrante do Laboratório de Robótica (LabRob) da Pontifícia Universidade Católica do Rio de Janeiro, Vivian Suzano Medeiros.

File - D:\DOWNLOADS\parseserial3D_log.py

```

1  #!/usr/bin/python
2
3  import serial
4  import time
5  import math
6  import csv
7  from datetime import datetime
8  import time
9
10 serdev = 'COM3' # serial device of JeVois
11 ser = serial.Serial(serdev, 115200, timeout=1)
12
13
14 def config_camera():
15     # Receive data from USB
16     ser.write("setpar serout USB\r".encode())
17     out = ser.readline().rstrip()
18     print (out)
19
20     # Initialize DemoArUco module
21     ser.write("setmapping2 YUYV 640 480 20.0 JeVois
DemoArUco\r".encode())
22     out = ser.readline().rstrip()
23     print (out)
24
25     # Set 3D pose messages
26     ser.write("setpar dopose true\r".encode())
27     out = ser.readline().rstrip()
28     print (out)
29
30     # Set 3D pose messages
31     ser.write("setpar markerlen 33\r".encode())
32     out = ser.readline().rstrip()
33     print (out)
34
35     # Set 3D pose messages
36     ser.write("setpar serprec 4\r".encode())
37     out = ser.readline().rstrip()
38     print (out)
39
40     # Set Detail style messages

```

File - D:\DOWNLOADS\parseserial3D_log.py

```

41     ser.write("setpar serstyle Detail\r".encode())
42     out = ser.readline().rstrip()
43     print (out)
44
45     # Set Detail style messages
46     ser.write("setpar serstamp FrameTime\r".encode())
47     out = ser.readline().rstrip()
48     print (out)
49
50     # Start image stream
51     ser.write("streamon\r".encode())
52     out = ser.readline().rstrip()
53     print (out)
54
55     time.sleep(1)
56     ser.flushInput()
57     ser.flushOutput()
58
59 def main():
60
61     config_camera()
62
63     # generic log file
64     f1 = open('csv_file.csv', 'w')
65     writer1 = csv.writer(f1)
66
67     # # log file with date
68     # now = datetime.now()
69     # file_time = now.strftime("%Y_%m_%d_%H-%M")
70     # filename = '/home/vivian/Dropbox/JeVois/Matlab/
data/matheus_'+file_time+'.csv'
71     # f2 = open(filename, 'w')
72     # writer2 = csv.writer(f2)
73
74     # Read data continuously
75     while True:
76         # Read a whole line and strip any trailing
line ending character:
77         line = ser.readline().rstrip().decode()
78         #print ("received: {}".format(line))
79

```

File - D:\DOWNLOADS\parseserial3D_log.py

```

80         # Split the line into tokens:
81         tok = line.split()
82         #print(f"token: {tok}")
83
84         # Skip if timeout or malformed line:
85         if (len(tok) < 1): continue
86         #print (len(tok))
87
88         # Skip if not a standardized "Detail 3D"
message:
89         # See http://jevois.org/doc/UserSerialStyle.
html
90         #if tok[1] != 'D3': continue
91
92         # From now on, we hence expect: timestamp D3
id x y z w h d q1 q2 q3 q4
93         #if len(tok) != 13: continue
94
95         # Assign some named Python variables to the
tokens:
96         timestamp, key, id, x, y, z, w, h, d, q1, q2
, q3, q4 = tok
97
98         # get datetime
99         now = datetime.now()
100         ts = now.strftime("%H:%M:%S.%f")
101         print("Timestamp jevois: {}, Timestamp
python {}".format(timestamp,ts))
102
103         print ("Found ArUco {} at ({:.2f},{:.2f},{:.
2f}) quat ({:.2f},{:.2f},{:.2f},{:.2f})".format(id,
float(x), float(y), float(z), float(q1), float(q2),
float(q3), float(q4)))
104
105         row = [ts, key, id, x, y, z, w, h, d, q1, q2
, q3, q4]
106         writer1.writerow(row)
107         #writer2.writerow(row)
108
109         f.close()
110 if __name__ == "__main__":

```

File - D:\DOWNLOADS\parseserial3D_log.py

```
111     main()  
112  
113  
114  
115
```

Apêndice A

Desenhos de Fabricação das Peças em Alumínio

Este apêndice apresenta detalhadamente os desenhos de fabricação das peças das garras e dos distanciadores em alumínio, referências para a fabricação de um exemplar para testes de funcionamento e de força do protótipo proposto.

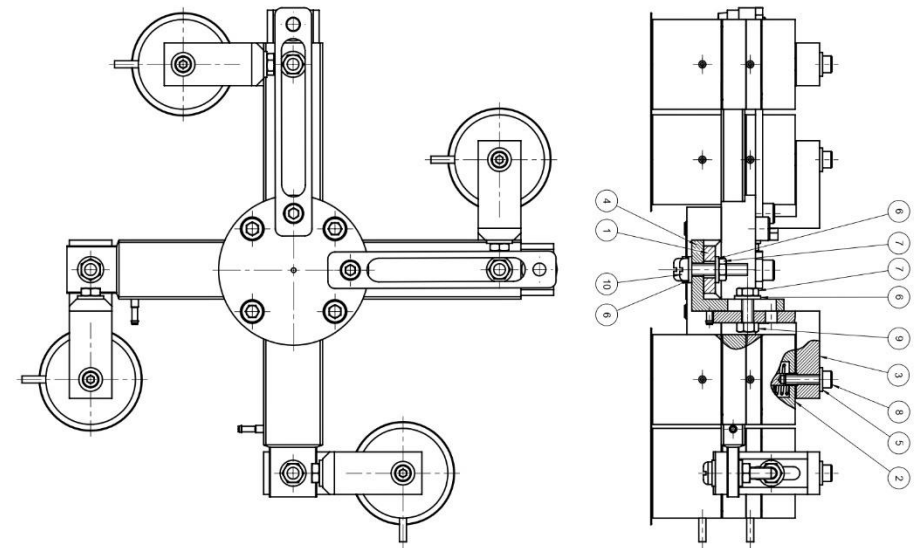
[illegible]

Apêndice B

Desenhos de Fabricação do Robô

Neste apêndice são mostrados os desenhos de conjunto e de fabricação das peças da versão final do protótipo de conceito. Nas partes dos conjuntos, o esquema de montagem é detalhado com o posicionamento de cada componente e suas formas de fixação. Os desenhos de fabricação das peças contam com o detalhamento de cotas e processos a serem feitos em cada item.

SALVO ESPECIFICAÇÃO CONTRÁRIA.		SALVO ESPECIFICAÇÃO CONTRÁRIA, TODOS OS ACABAMENTOS SÃO ALISADOS.							
DIMENSÕES EM MILÍMETROS. TOLERÂNCIAS: GERAL +0,1; ÂNGULO ± 30°; REMOVER REBARBAS; QUEBRAR CANTOS VIVOS.	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="text-align: center;"></td> <td style="text-align: center;">0,05/100</td> </tr> <tr> <td style="text-align: center;"></td> <td style="text-align: center;">0,05/100</td> </tr> <tr> <td style="text-align: center;"></td> <td style="text-align: center;">0,05/100</td> </tr> </table>		0,05/100		0,05/100		0,05/100	TRABALHE NAS MEDIDAS, NÃO USE ESCALA.	
	0,05/100								
	0,05/100								
	0,05/100								



Item	QTD	Descrição	Material
1	1	Corpo do robô	
2	4	Montagem da garra	
3	4	Cilindro de ar	
4	4	Carrocinha do corpo	
5	4	Articulação Isai	
6	12	Faixa articulada	
7	8	Faixa articulada com colchete	
8	4	Parede lateral esquerda	
9	4	Parede lateral direita	
10	4	Parede traseira	

CONTEÚDO DESDE O PROJETO	DATA	AUTORES	REVISÃO	APROVADO
PROJETO	13/01/22	MATHEUS MATHEUS		

TÍTULO:	REFERÊNCIA	SUBSTITUIÇÃO POR:
LAGARTIXA PNEUMÁTICA MONTAGEM GERAL		
ESCALA: SEM ESCALA		

DETALHAMENTO	DIMENSÃO
MEM-01-01-001-3	

SALVO ESPECIFICAÇÃO CONTRÁRIA.

DIMENSÕES EM MILÍMETROS.
TOLERÂNCIAS: GERAL $\pm 0,1$.
ÂNGULO $\pm 30^\circ$.
REMOVER REBARBAS.
QUEBRAR CANTOS VIVOS.

SALVO ESPECIFICAÇÃO CONTRÁRIA, TODOS OS ACABAMENTOS SÃO ALISADOS.

TRABALHE NAS MEDIDAS. NÃO USE ESCALA.

Figura 1: Perspectiva

Figura 2: Corte B-B

Figura 3: Corte A-A

Figura 4: Vista superior

Figura 5: Vista lateral

Item	QTD	Descrição	Material	Dimensões (BR/IT)
11	1	Parafuso de cabeça com cabeça	Comercial	M6 x 25
10	1	Parafuso de cabeça com cabeça	Comercial	M6 x 16
9	4	Parafuso allen com cabeça	Comercial	M6 x 30
8	1	Parafuso allen com cabeça	Comercial	M6 x 16
7	1	Parafuso allen com cabeça	Comercial	M5 x 8
6	2	Colinho anelão	FESTO	Código: E2H420-50B
5	2	Bucha guia	Alumínio	1/8" x 20 x 101
4	1	Espalhador menor	Nylon	606F x 17
3	1	Espalhador maior	Nylon	606F x 12
2	1	Bloco superior	Nylon	602 127 x 20
1	1	Bloco inferior	Nylon	602 127 x 20

Figura 6: Detalhe do bloco inferior

Figura 7: Detalhe do bloco superior

Figura 8: Detalhe do parafuso de cabeça

Figura 9: Detalhe do parafuso allen

Figura 10: Detalhe do colinho anelão

Figura 11: Detalhe da bucha guia

Figura 12: Detalhe do espalhador menor

Figura 13: Detalhe do espalhador maior

Figura 14: Detalhe do bloco superior

Figura 15: Detalhe do bloco inferior

Figura 16: Detalhe do parafuso de cabeça

Figura 17: Detalhe do parafuso allen

Figura 18: Detalhe do colinho anelão

Figura 19: Detalhe da bucha guia

Figura 20: Detalhe do espalhador menor

Figura 21: Detalhe do espalhador maior

Figura 22: Detalhe do bloco superior

Figura 23: Detalhe do bloco inferior

Figura 24: Detalhe do parafuso de cabeça

Figura 25: Detalhe do parafuso allen

Figura 26: Detalhe do colinho anelão

Figura 27: Detalhe da bucha guia

Figura 28: Detalhe do espalhador menor

Figura 29: Detalhe do espalhador maior

Figura 30: Detalhe do bloco superior

Figura 31: Detalhe do bloco inferior

Figura 32: Detalhe do parafuso de cabeça

Figura 33: Detalhe do parafuso allen

Figura 34: Detalhe do colinho anelão

Figura 35: Detalhe da bucha guia

Figura 36: Detalhe do espalhador menor

Figura 37: Detalhe do espalhador maior

Figura 38: Detalhe do bloco superior

Figura 39: Detalhe do bloco inferior

Figura 40: Detalhe do parafuso de cabeça

Figura 41: Detalhe do parafuso allen

Figura 42: Detalhe do colinho anelão

Figura 43: Detalhe da bucha guia

Figura 44: Detalhe do espalhador menor

Figura 45: Detalhe do espalhador maior

Figura 46: Detalhe do bloco superior

Figura 47: Detalhe do bloco inferior

Figura 48: Detalhe do parafuso de cabeça

Figura 49: Detalhe do parafuso allen

Figura 50: Detalhe do colinho anelão

Figura 51: Detalhe da bucha guia

Figura 52: Detalhe do espalhador menor

Figura 53: Detalhe do espalhador maior

Figura 54: Detalhe do bloco superior

Figura 55: Detalhe do bloco inferior

Figura 56: Detalhe do parafuso de cabeça

Figura 57: Detalhe do parafuso allen

Figura 58: Detalhe do colinho anelão

Figura 59: Detalhe da bucha guia

Figura 60: Detalhe do espalhador menor

Figura 61: Detalhe do espalhador maior

Figura 62: Detalhe do bloco superior

Figura 63: Detalhe do bloco inferior

Figura 64: Detalhe do parafuso de cabeça

Figura 65: Detalhe do parafuso allen

Figura 66: Detalhe do colinho anelão

Figura 67: Detalhe da bucha guia

Figura 68: Detalhe do espalhador menor

Figura 69: Detalhe do espalhador maior

Figura 70: Detalhe do bloco superior

Figura 71: Detalhe do bloco inferior

Figura 72: Detalhe do parafuso de cabeça

Figura 73: Detalhe do parafuso allen

Figura 74: Detalhe do colinho anelão

Figura 75: Detalhe da bucha guia

Figura 76: Detalhe do espalhador menor

Figura 77: Detalhe do espalhador maior

Figura 78: Detalhe do bloco superior

Figura 79: Detalhe do bloco inferior

Figura 80: Detalhe do parafuso de cabeça

Figura 81: Detalhe do parafuso allen

Figura 82: Detalhe do colinho anelão

Figura 83: Detalhe da bucha guia

Figura 84: Detalhe do espalhador menor

Figura 85: Detalhe do espalhador maior

Figura 86: Detalhe do bloco superior

Figura 87: Detalhe do bloco inferior

Figura 88: Detalhe do parafuso de cabeça

Figura 89: Detalhe do parafuso allen

Figura 90: Detalhe do colinho anelão

Figura 91: Detalhe da bucha guia

Figura 92: Detalhe do espalhador menor

Figura 93: Detalhe do espalhador maior

Figura 94: Detalhe do bloco superior

Figura 95: Detalhe do bloco inferior

Figura 96: Detalhe do parafuso de cabeça

Figura 97: Detalhe do parafuso allen

Figura 98: Detalhe do colinho anelão

Figura 99: Detalhe da bucha guia

Figura 100: Detalhe do espalhador menor

Figura 101: Detalhe do espalhador maior

Figura 102: Detalhe do bloco superior

Figura 103: Detalhe do bloco inferior

Figura 104: Detalhe do parafuso de cabeça

Figura 105: Detalhe do parafuso allen

Figura 106: Detalhe do colinho anelão

Figura 107: Detalhe da bucha guia

Figura 108: Detalhe do espalhador menor

Figura 109: Detalhe do espalhador maior

Figura 110: Detalhe do bloco superior

Figura 111: Detalhe do bloco inferior

Figura 112: Detalhe do parafuso de cabeça

Figura 113: Detalhe do parafuso allen

Figura 114: Detalhe do colinho anelão

Figura 115: Detalhe da bucha guia

Figura 116: Detalhe do espalhador menor

Figura 117: Detalhe do espalhador maior

Figura 118: Detalhe do bloco superior

Figura 119: Detalhe do bloco inferior

Figura 120: Detalhe do parafuso de cabeça

Figura 121: Detalhe do parafuso allen

Figura 122: Detalhe do colinho anelão

Figura 123: Detalhe da bucha guia

Figura 124: Detalhe do espalhador menor

Figura 125: Detalhe do espalhador maior

Figura 126: Detalhe do bloco superior

Figura 127: Detalhe do bloco inferior

Figura 128: Detalhe do parafuso de cabeça

Figura 129: Detalhe do parafuso allen

Figura 130: Detalhe do colinho anelão

Figura 131: Detalhe da bucha guia

Figura 132: Detalhe do espalhador menor

Figura 133: Detalhe do espalhador maior

Figura 134: Detalhe do bloco superior

Figura 135: Detalhe do bloco inferior

Figura 136: Detalhe do parafuso de cabeça

Figura 137: Detalhe do parafuso allen

Figura 138: Detalhe do colinho anelão

Figura 139: Detalhe da bucha guia

Figura 140: Detalhe do espalhador menor

Figura 141: Detalhe do espalhador maior

Figura 142: Detalhe do bloco superior

Figura 143: Detalhe do bloco inferior

Figura 144: Detalhe do parafuso de cabeça

Figura 145: Detalhe do parafuso allen

Figura 146: Detalhe do colinho anelão

Figura 147: Detalhe da bucha guia

Figura 148: Detalhe do espalhador menor

Figura 149: Detalhe do espalhador maior

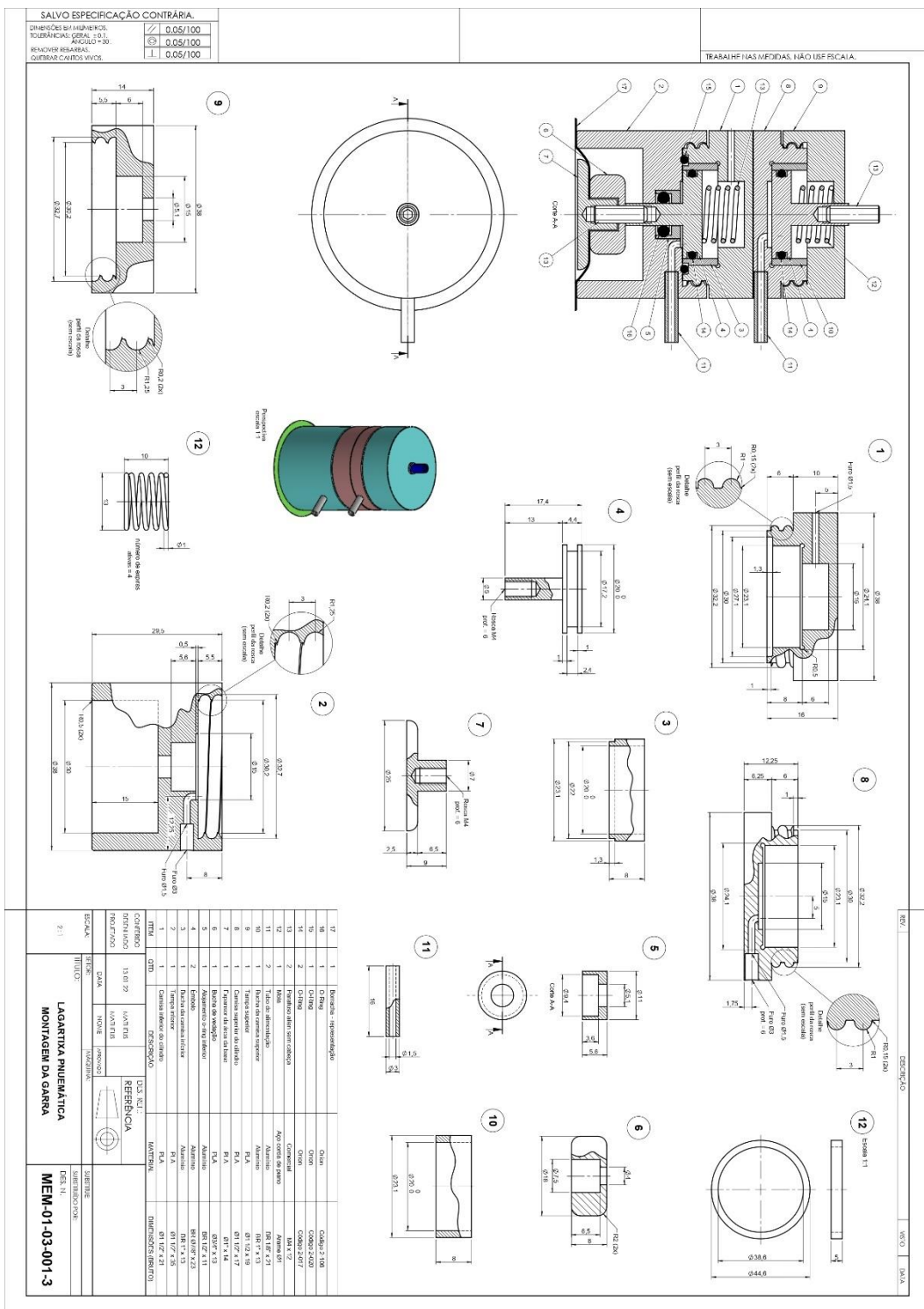
Figura 150: Detalhe do bloco superior

Figura 151: Detalhe do bloco inferior

Figura 152: Detalhe do parafuso de cabeça

Figura 153: Detalhe do parafuso allen

Figura 154



[illegible]

Apêndice C

Código para simulação dinâmica no *Simulink*

Neste apêndice são mostrados os códigos para implementação da modelagem dinâmica do robô no *MATLAB*® e no *Simulink*. Como cada um dos 10 atuadores presentes neste projeto foi modelado e implementado separadamente, alguns dos blocos mostrados são referentes a apenas um dos atuadores, haja vista que as lógicas são as mesmas, diferindo apenas algumas variáveis.

```

% Código para simulação da lagartixa pneumática

close all
clc
clear all

V = 0.028;           % volume do reservatório [m^3]
T = 298;             % temp do ar a montante [K]
Pr = 600000;         % pressao do reservatório [Pa]
Patm = 1.01325e5;    % pressao inicial da camara a jusante (atmosfera) [Pa]
Cd = 0.8;            % coeficiente de descarga
A0 = 7.95e-7;        % área do orifício das válvulas
A0c = 7.85e-7;       % área do orifício da válvula compressor --> reservatório
n = 1;              % expoente politrópico (processo isotérmico)
R = 287;            % cte do gás para o ar
gamma = 1.4;        % razão dos calores específicos
g = 9.81;           % aceleração da gravidade

m = 1.5;            % massa da lagartixa
A1_CP = 0.0001;     % área do êmbolo do cilindro
V0_CP = 1.5e-5;     % volume inicial de ar
b = 1.59;           % coeficiente de atrito viscoso
curso_CP = 0.05;    % curso do cilindro
x_pre_CP = 53/300;  % pré compressão da mola [m]
k_CP = 60;          % constante da mola

m_G = 0.02;         % massa das peças carregadas pelo êmbolo da garra
A1_G = 3.14e-4;     % área do êmbolo da garra
V0_G = 5e-6;        % volume inicial de ar na garra
curso_G = 0.002;    % curso da garra
x_pre_G = 0.002;    % pré compressão da mola da garra [m]
k_G = 3616.9;       % constante da mola da garra
A0_G = 2.6166e-07;  % área do orifício das válvulas da garra

m_D = m/2;          % massa das peças carregadas pelo êmbolo do distanciador
A1_D = 3.14e-4;     % área do êmbolo do distanciador
V0_D = 5e-6;        % volume inicial de ar no distanciador
curso_D = 0.002;    % curso do distanciador
x_pre_D = 0.002;    % pré compressão da mola do distanciador [m]
k_D = 3616.9;       % constante da mola do distanciador
A0_D = 2.6166e-07;  % área do orifício das válvulas do distanciador

dt = 5e-4;

sim('lagartixa_rev09') % roda o programa no Simulink

%subplot(3,1,1);
figure (1)
plot(t,X1,'LineWidth',1.5)
hold on
grid on
plot(t,X2,'--','LineWidth',1.5)
legend('CP1','CP2')
title('Deslocamento dos êmbolos dos atuadores pricipais')
xlabel('t [s]')
ylabel('X [m]')
vlim([-0.01 0.06])

```

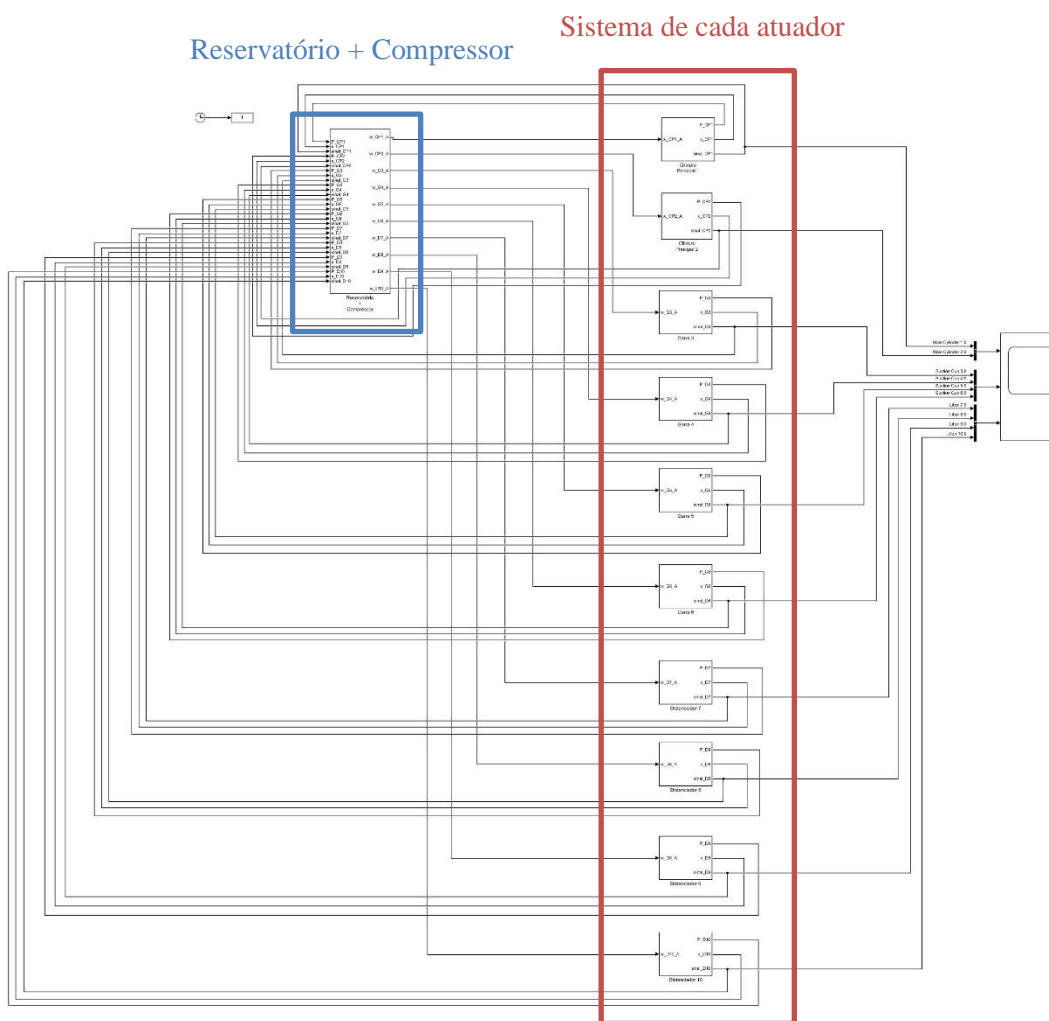
```

%subplot(3,1,2);
figure(2)
plot(t,X3,'LineWidth',1.5)
hold on
grid on
plot(t,X4,'--','LineWidth',1.5)
plot(t,X5,'LineWidth',1.5)
plot(t,X6,'--','LineWidth',1.5)
legend('G3','G4','G5','G6')
title('Deslocamento dos êmbolos das garras')
xlabel('t [s]')
ylabel('X [m]')
ylim([-0.001 0.003])

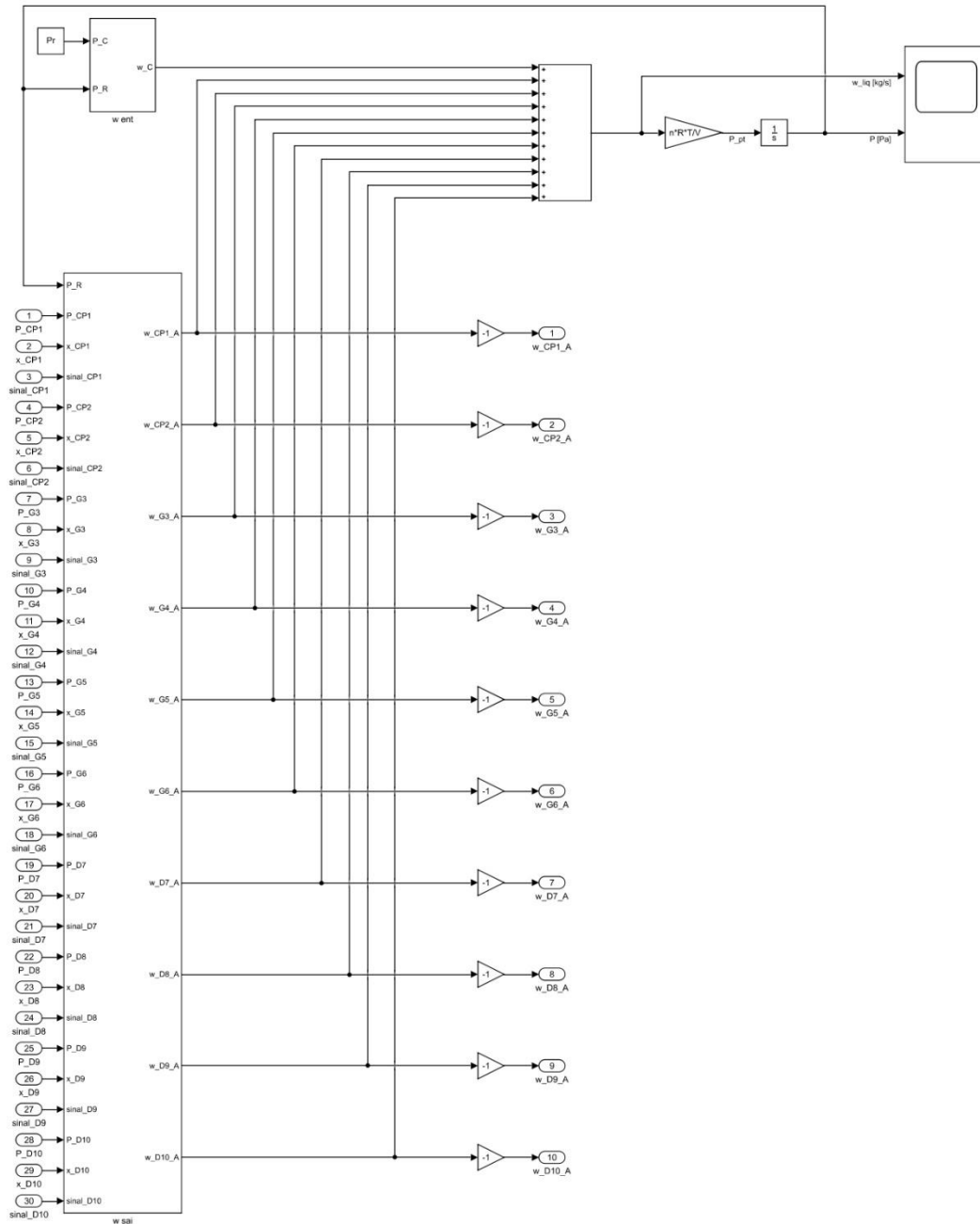
%subplot(3,1,3);
figure(3)
plot(t,X7,'LineWidth',1.5)
hold on
grid on
plot(t,X8,'--','LineWidth',1.5)
plot(t,X9,'LineWidth',1.5)
plot(t,X10,'--','LineWidth',1.5)
legend('D7','D8','D9','D10')
title('Deslocamento dos êmbolos dos distanciadores')
xlabel('t [s]')
ylabel('X [m]')
ylim([-0.001 0.003])

```

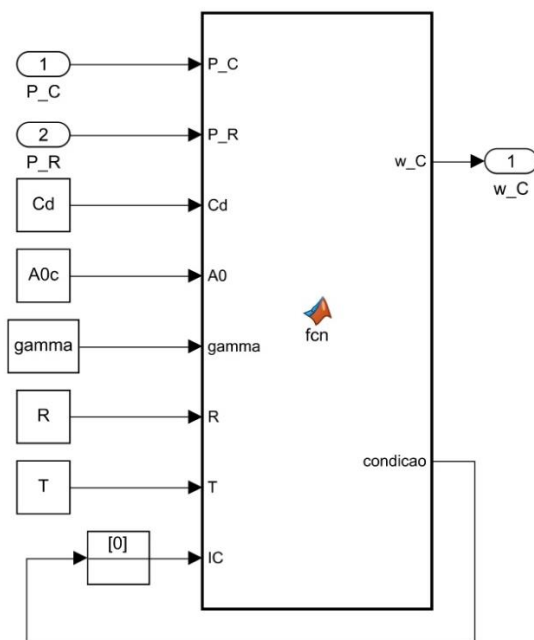
Tela principal do programa do *Simulink*:



Dentro do bloco “Reservatório + Compressor” representado pelo bloco azul na tela principal do programa do *Simulink*:



Dentro do bloco “w ent” ocorre o enchimento do reservatório de ar comprimido através do compressor. Esta parte está simplificada, haja vista que o dimensionamento do compressor e do reservatório não foram feitos neste trabalho.



```
function [w_C,condicao]= fcn(P_C,P_R,Cd,A0,gamma,R,T,IC)
```

```
% Pr = pressao do reservatório
% Pc = pressao do compressor
% w > 0 : fluxo compressor->reservatório
% w < 0 : fluxo reservatório->compressor
```

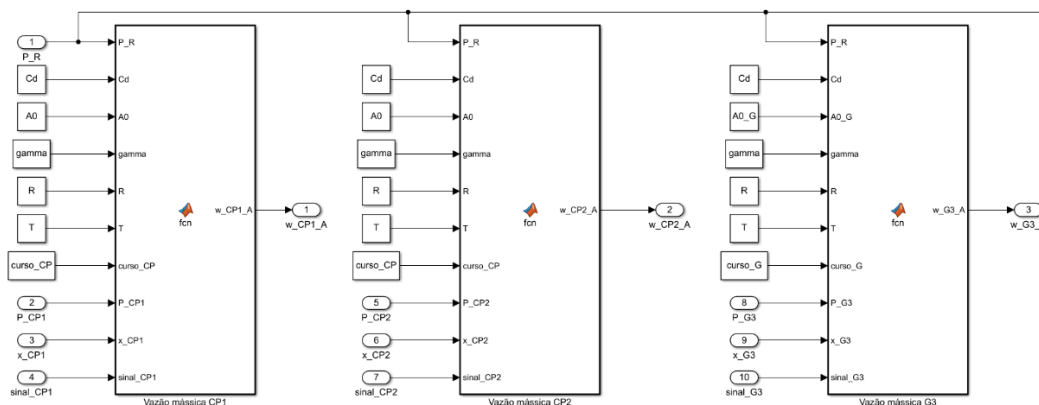
```
P1 = max(P_C,P_R);
P2 = min(P_C,P_R);
```

```
Cr = (2/(gamma + 1))^(gamma/(gamma-1));
```

```
condicao = IC;
if P_R <= 550000
    condicao = 1;
end
if P_R >= 590000
    condicao = 0;
end
```

```
if condicao == 1
%     if P2/P1 > Cr
%         %ilimitado
%         w_C = sign(P_C-P_R) * Cd*A0*P1*sqrt( 2*gamma/((gamma-1)*R*T)* ( (P2/P1)^(2/gamma) - (P2/P1)^((gamma+1)/gamma) ) );
%     else
%         %limitado
%         w_C = sign(P_C-P_R) * Cd*A0*P1*sqrt( gamma/(R*T) * Cr^((gamma+1)/gamma) );
%     end
    w_C = 17e-4; %[kg/s]
else
    w_C = 0;
end
```

Dentro do bloco “w sai”, ocorrem os cálculos das vazões mássicas para os momentos de avanço de cada um dos 10 atuadores. Na imagem abaixo apenas 3 blocos estão representados, porém cada um dos atuadores tem seu próprio bloco.



```
% As variáveis apresentadas neste script são referentes ao Cilindro
% Principal 1. (curso_CP, P_CP1, x_CP1, sinal_CP1)

% Os códigos matemáticos para os outros atuadores é igual, alterando-se
% apenas as variáveis.

function w_CP1_A = fcn(P_R,Cd,A0,gamma,R,T,curso_CP,P_CP1,x_CP1,sinal_CP1)

% Pr = pressao do reservatório
% Pc = pressao da camara
% w > 0 : fluxo reservatório->atmosfera
% w < 0 : fluxo atmosfera->reservatório

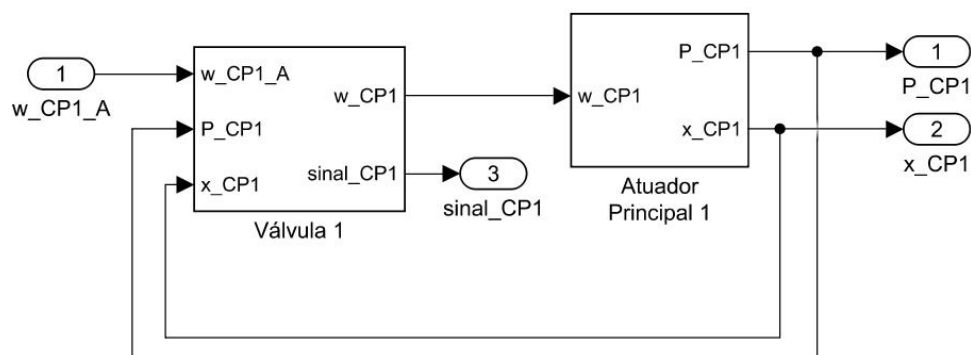
Cr = (2/(gamma + 1))^(gamma/(gamma-1));

P1 = max(P_R,P_CP1);
P2 = min(P_R,P_CP1);

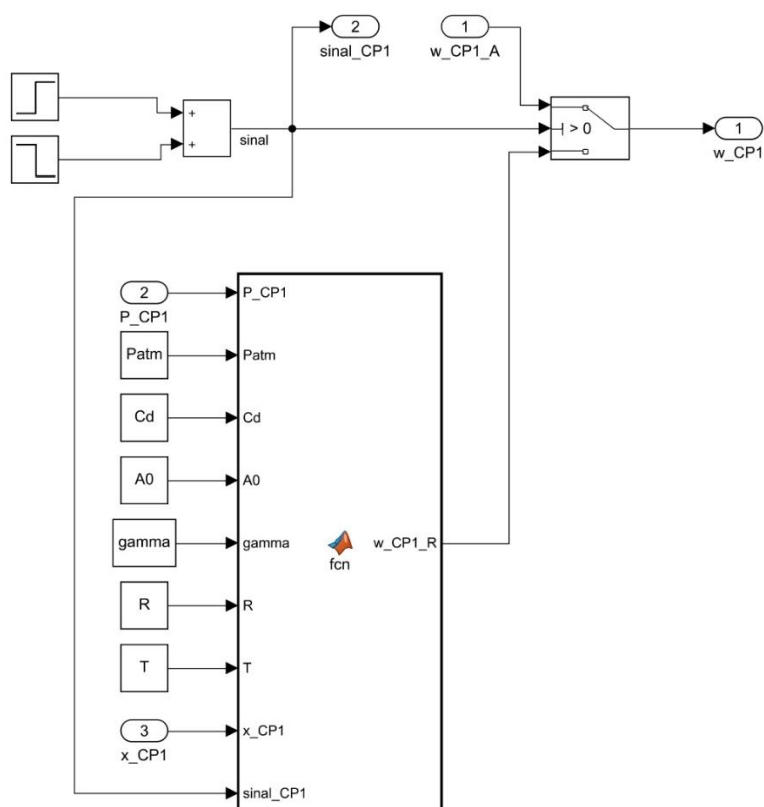
if P2/P1 > Cr
    %ilimitado
    w_CP1_A = sign(P_CP1-P_R) * Cd*A0*P1*sqrt( 2*gamma/((gamma-1)*R*T)* ( (P2/P1)^(2/gamma) - (P2/P1)^((gamma+1)/gamma) ) );
else
    %limitado
    w_CP1_A = sign(P_CP1-P_R) * Cd*A0*P1*sqrt( gamma/(R*T) * Cr^((gamma+1)/gamma) );
end

if (P_CP1 >= P_R) || (sinal_CP1 == 0)
    w_CP1_A = 0;
end
```

O bloco “Sistema de cada atuador” representado pelo bloco vermelho na tela principal do programa do *Simulink*, aglutina os blocos de cada um dos atuadores do robô. Abaixo é detalhado o bloco do atuador “Cilindro Principal 1”:



Dentro do bloco “Válvula 1”, os sinais para a válvula de comando são gerados, além do cálculo de vazão na câmara do atuador durante a fase de recuo.



```

% As variáveis apresentadas neste script são referentes ao Cilindro
% Principal 1. (P_CP1, x_CP1, sinal_CP1)

% Os códigos matemáticos para os outros atuadores é igual, alterando-se
% apenas as variáveis.

function w_CP1_R = fcn(P_CP1, Patm, Cd, A0, gamma, R, T, x_CP1, sinal_CP1)

% Pr = pressao do reservatório
% Pc = pressao da camara
% w > 0 : fluxo reservatório->atmosfera
% w < 0 : fluxo atmosfera->reservatório

P1 = max(P_CP1, Patm);
P2 = min(P_CP1, Patm);

Cr = (2/(gamma + 1))^(gamma/(gamma-1));

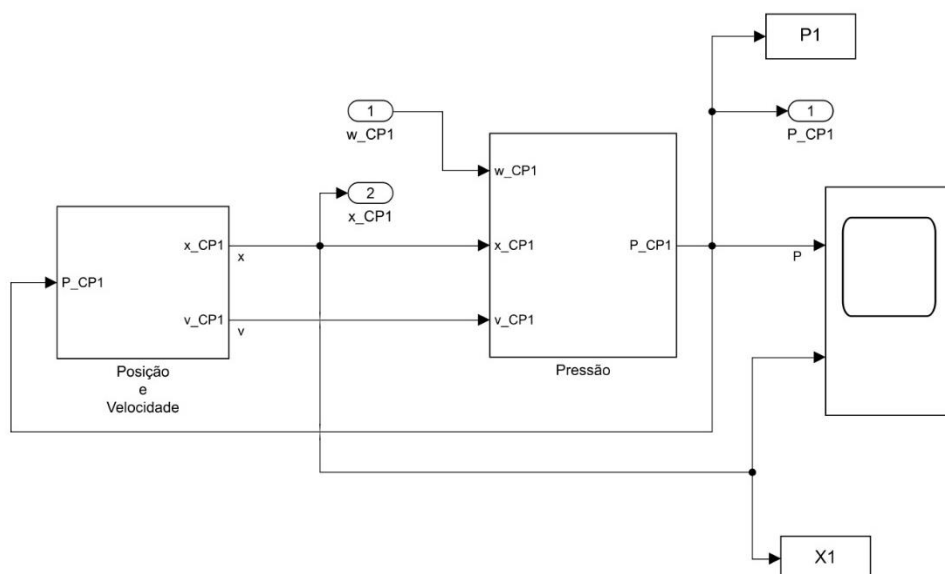
if P2/P1 > Cr
    %ilimitado
    w_CP1_R = sign(P2-P1) * Cd*A0*P1*sqrt( 2*gamma/((gamma-1)*R*T) * ( (P2/P1)^(2/gamma) - (P2/P1)^((gamma+1)/gamma) ) );
else
    %limitado
    w_CP1_R = sign(P2-P1) * Cd*A0*P1*sqrt( gamma/(R*T) * Cr^((gamma+1)/gamma) );
end

if (P_CP1 <= Patm) || (sinal_CP1 == 1)
    w_CP1_R = 0;
end

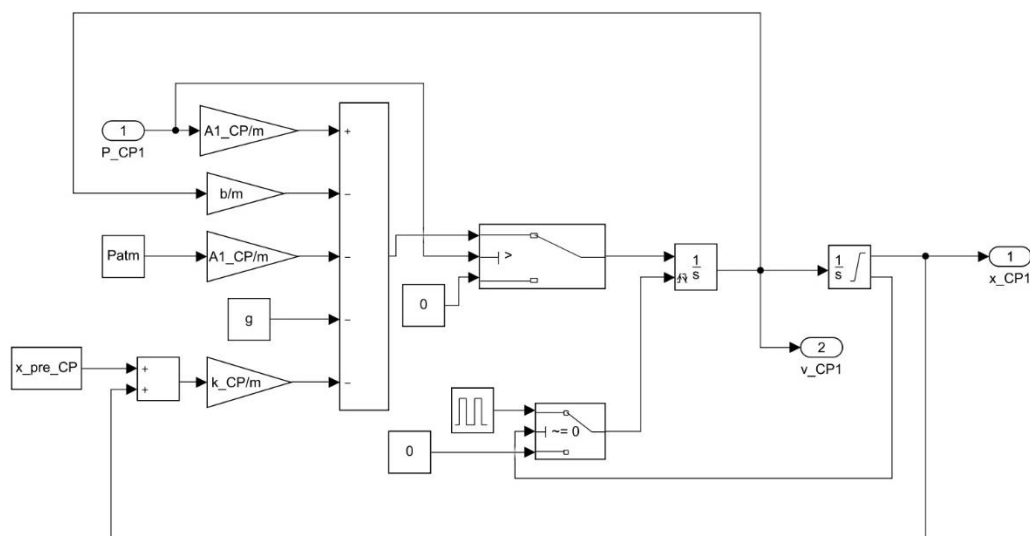
end

```

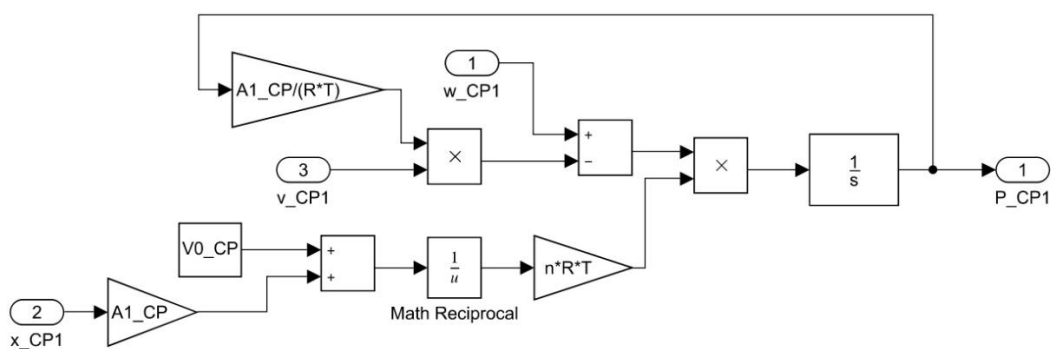
Dentro do bloco “Atuador Principal 1”, temos:



Onde o bloco “Posição e Velocidade” contém:



E o bloco “Pressão” contém:



Os algoritmos para cada um dos 10 atuadores do robô seguem a mesma lógica apresentada para o “Atuador Principal 1”, alterando-se apenas as variáveis.

Apêndice D

Script de Implementação da Rede Neural

```
1
2     clc, close all, clear all
3
4     % 1) Carregamento dos dados de entrada (X)
5     %     e dos dados de saída (target) (t):
6
7     dadosdeentradasaida_lagartixa % dados obtidos com testes no protótipo
8
9     % 2) Criação da Rede Neural:
10
11     nnc=[5 5]; % número de neurônios (ni) por cada camada ncc=[n1 n2 n3..]
12     net = feedforwardnet([nnc], 'trainlm'); % rede neural feedforward com
13                                             % método de treinamento
14                                             % levenberg-marquardt
15
16     % 3) Treinar (in sample) a rede neural usando os
17     %     dados X e t de treinamento:
18
19     net = train(net,x,t); % treinar a rede
20
21     view(net) % Ver a Rede Neural
22     save('net10') % salva os parâmetros (pesos, bias, topologia etc)
23     pb=getwb(net); % Valores dos pesos e bias após treinamento
24
25
```

Apêndice E

Script de Validação da Rede Neural

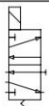

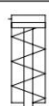



```
1
2 - dadosdeentradasaída2_lagartixa % outros dados obtidos no protótipo
3
4 - load('net')
5
6 - y2 = net(x2); % Apicando novas entradas (dadosdeentradasaída2)
7               % na RN net já treinada para achar valores calculados
8 - ct=length(t2); % Tamanho (número) de valores da saída (dadosdeentradasaída2)
9 - np=1:1:ct;
10
11 % Comparação entre a saída calculada pela RN já treinada (y2) e a saída
12 % testada no protótipo (t2) utilizando o banco de dadosdeentradasaída2:
13 - plot(np,t2,'b',np,y2,'r')
14
15 - figure(1)
16 - grid minor
17 - xlabel('Conjuntos de dados')
18 - ylabel('Valores das Saídas')
19 - title('Saídas calculadas (Vermelho) e saídas testadas no protótipo (Azul)')
20 - ylim([-0.1 1.1])
21 - xlim([1 ct])
22
```

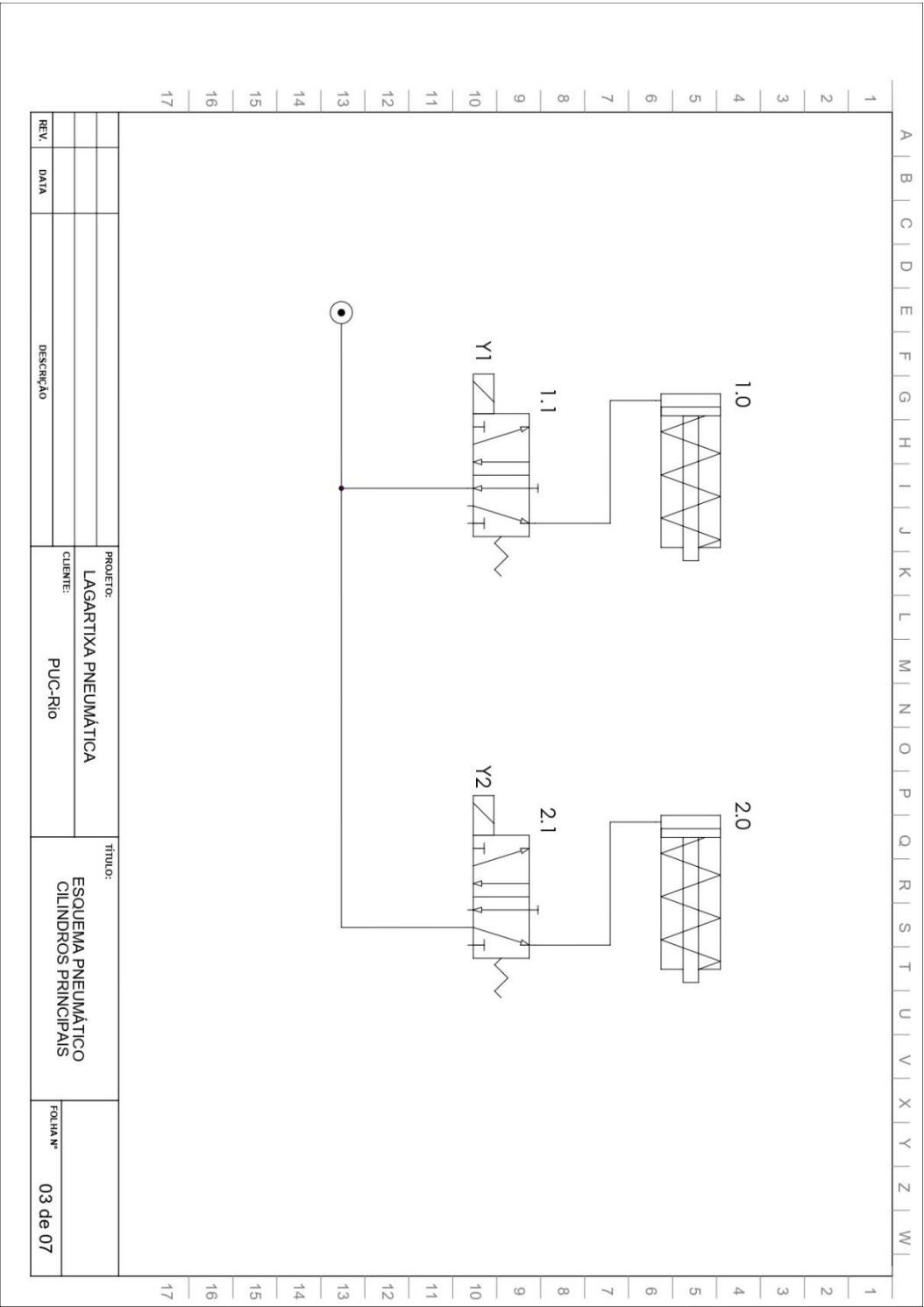
Apêndice F

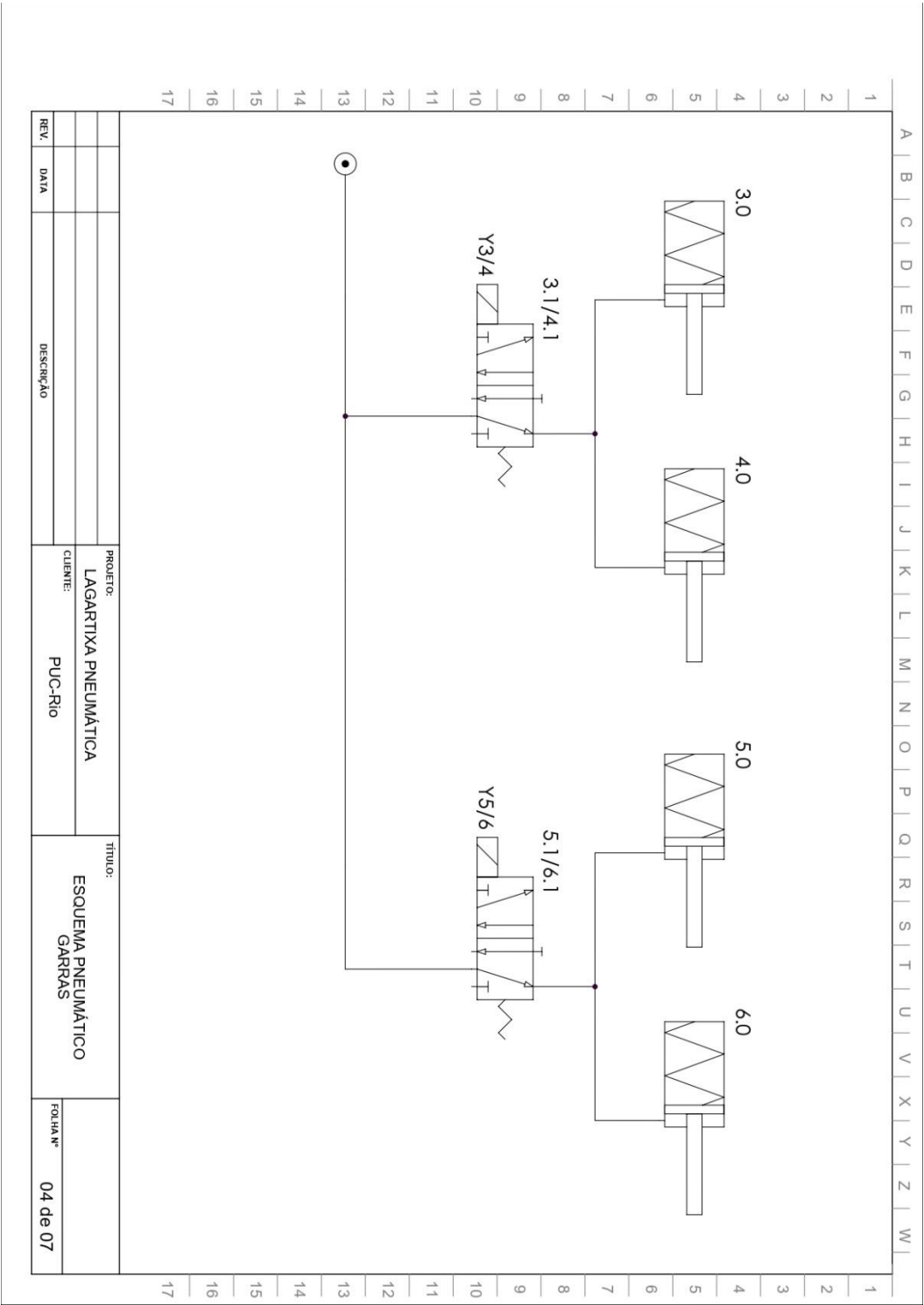
Esquema Pneumático e Elétrico

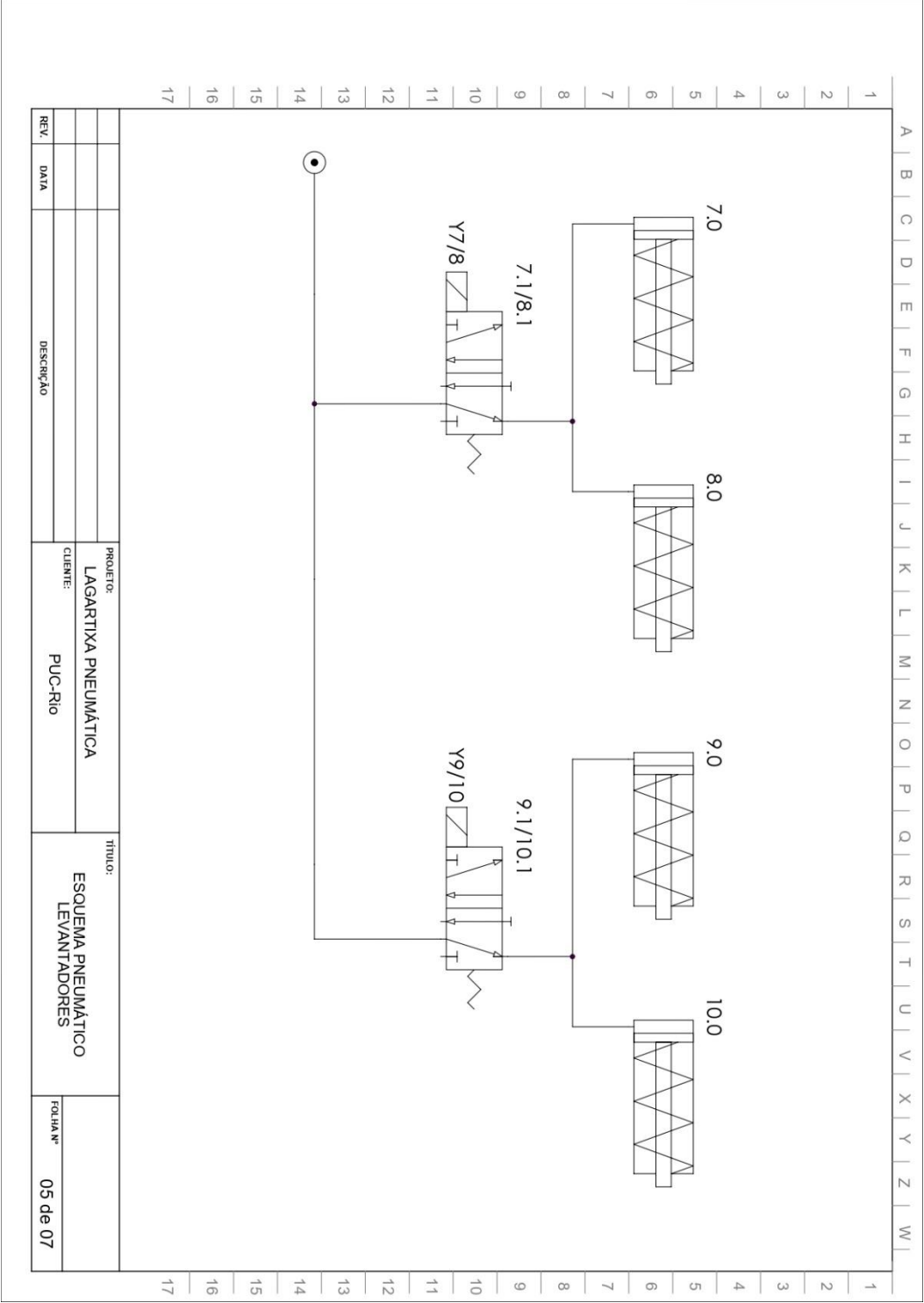
Neste apêndice é mostrado o esquema pneumático e elétrico montado para a execução dos experimentos com o protótipo de conceito. O esquema conta com indicação e especificação dos elementos utilizados, além de sua integração.

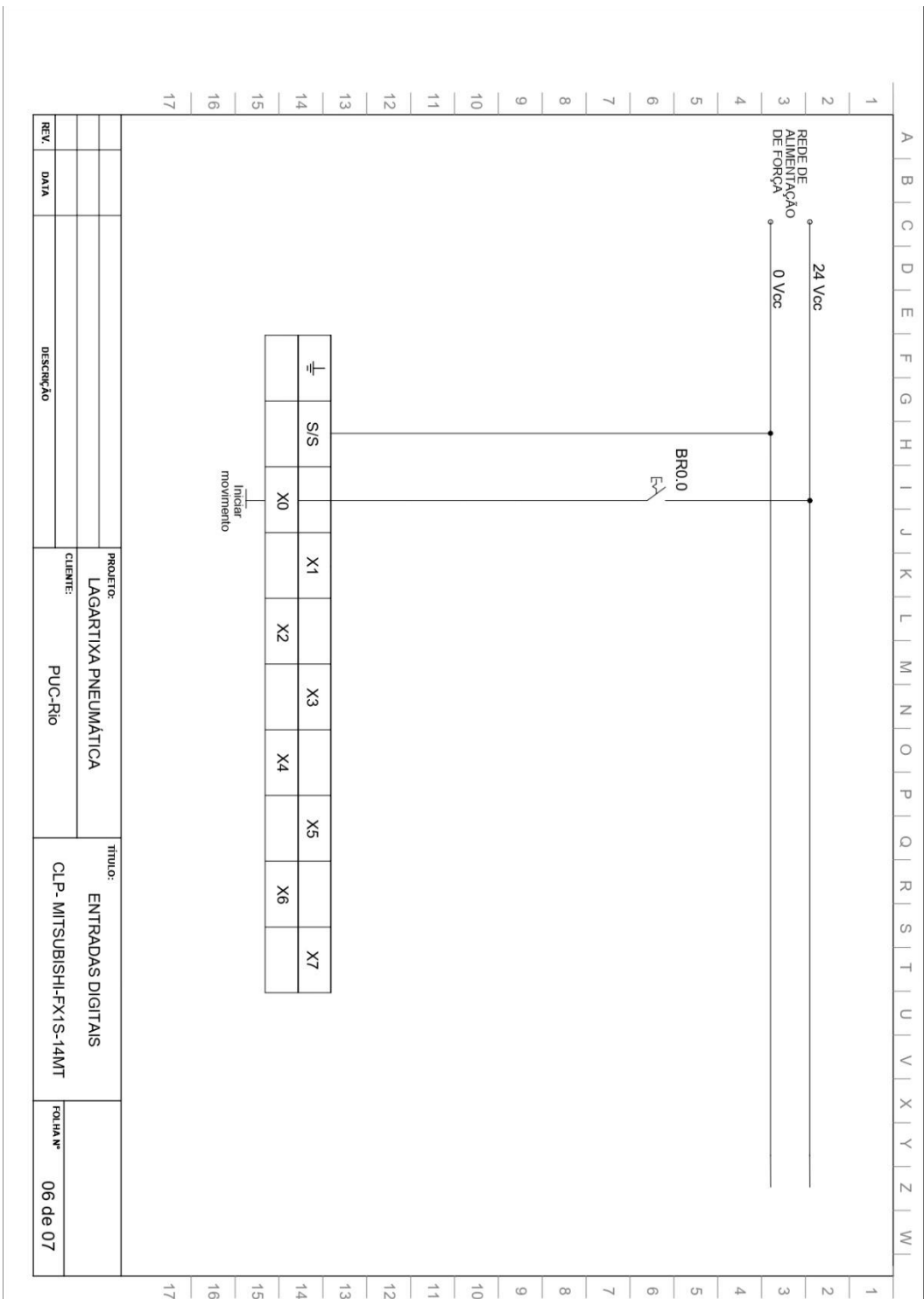
				PROJETO:		TÍTULO:	
				LAGARTIXA PNEUMÁTICA			
				CLIENTE:		SIMBOLOGIA	
				PUC-Rio		FOLHA Nº	
						02 de 07	
REV.	DATA	DESCRIÇÃO					

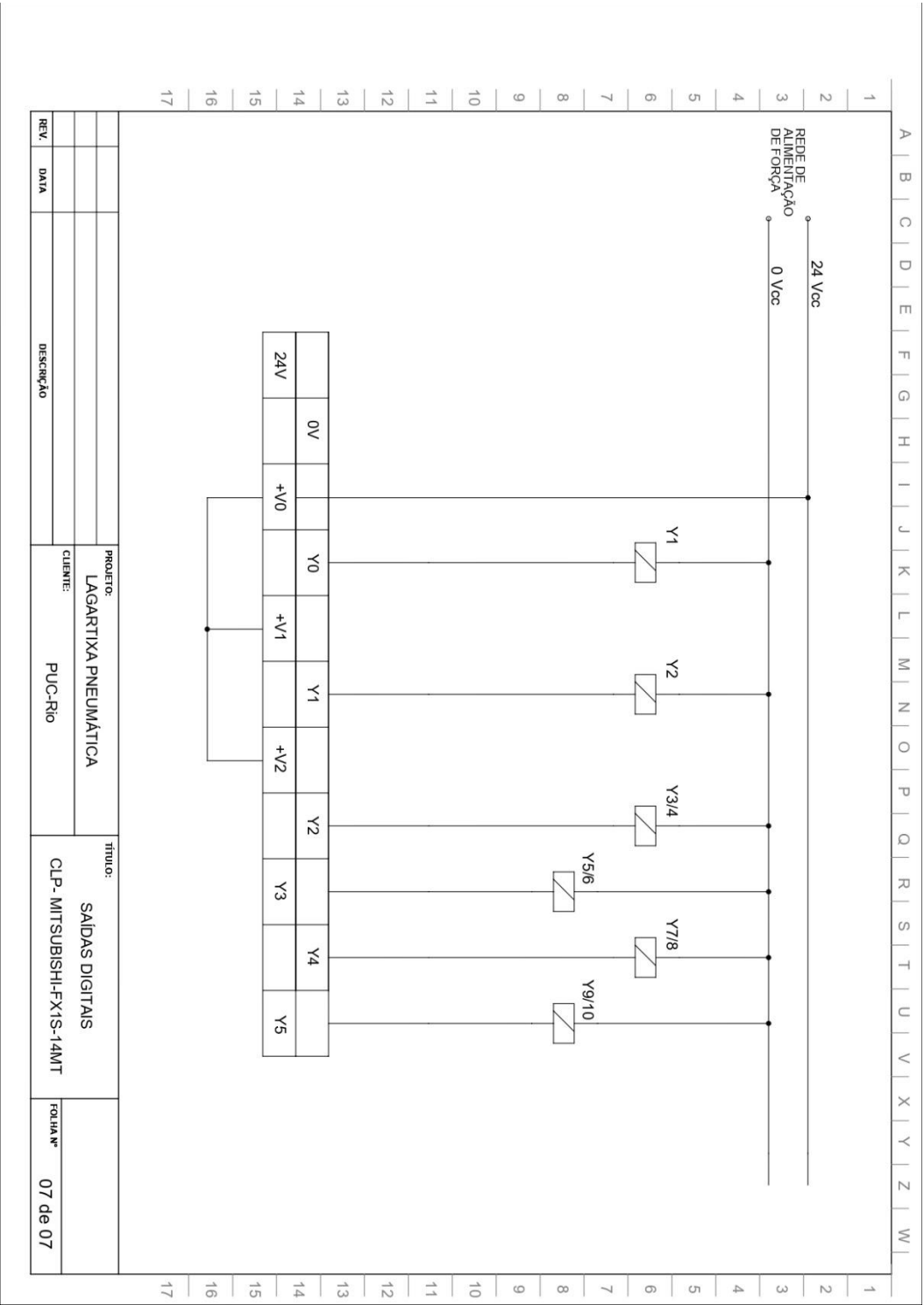
	Válvula pneumática 3/2 vias simples solenóide		Solenóide						
	Cilindro pneumático simples ação								
	Cilindro pneumático simples ação normalmente avançado								
	Fonte de corrente contínua 24 volts								
•	Ligação entre fios ou mangueiras								
	Bolão retilíneo 1/4								







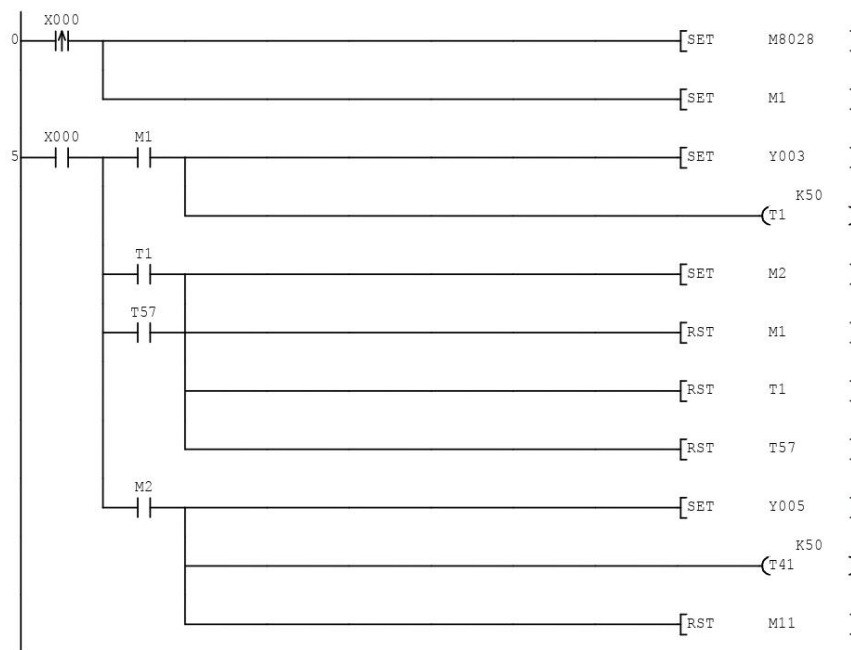


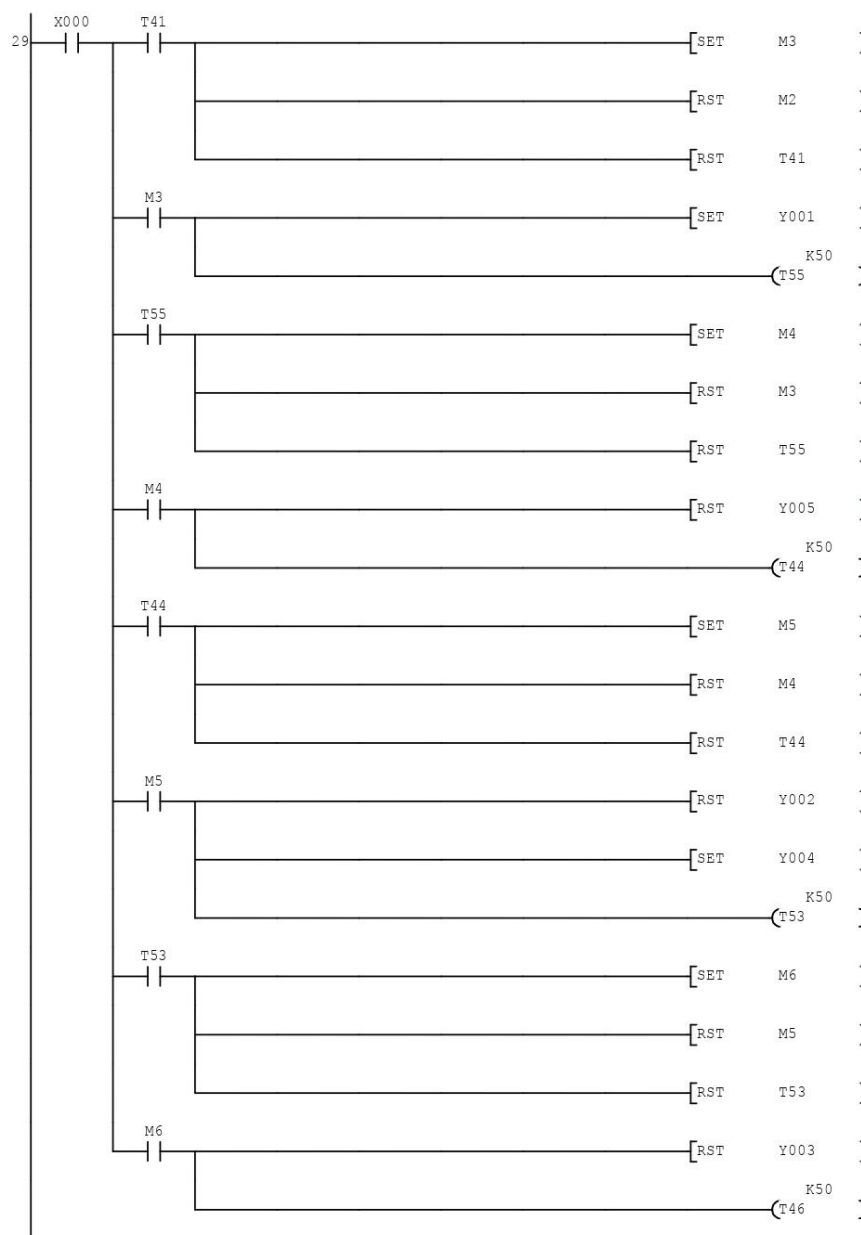


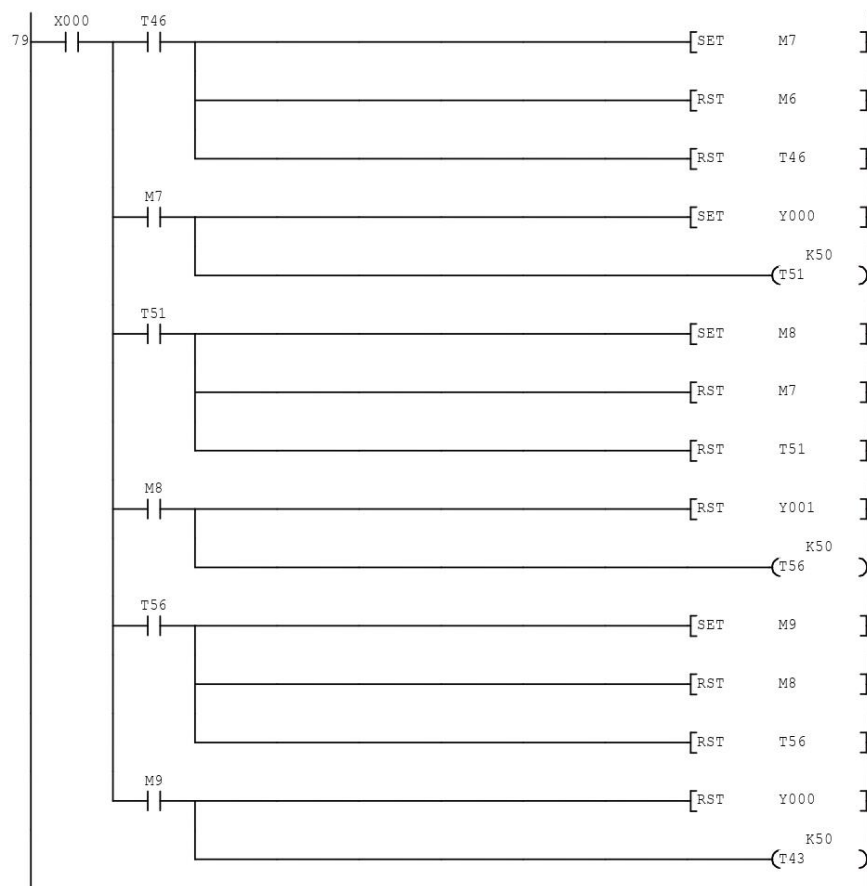
Apêndice G

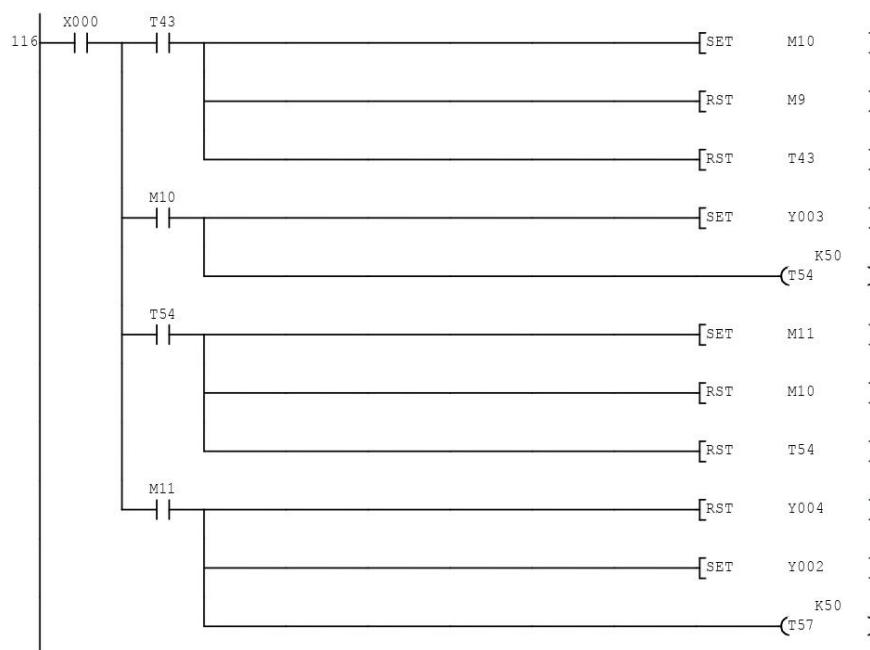
Programa do CLP

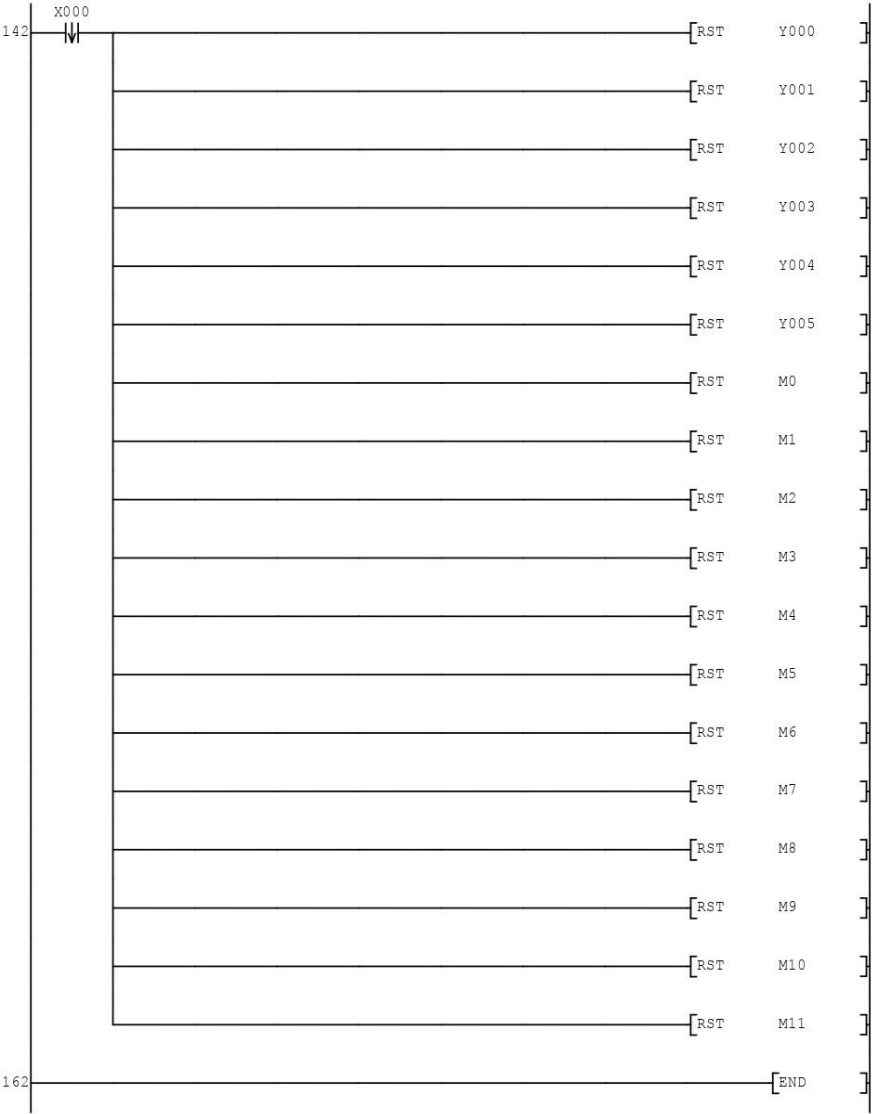
Neste apêndice é mostrado o programa na linguagem *Ladder* desenvolvido para CLP, programa este que é o responsável por enviar os sinais para as válvulas de controle. O algoritmo mostrado neste apêndice está na configuração inicial mais lenta, com tempo de disparo padrão de 0,5s entre todos os passos.











Apêndice H

Vídeo dos experimentos realizados com o protótipo

Neste apêndice encontra-se o *link* para o vídeo de alguns testes realizados no protótipo de conceito com a finalidade de obter o conjunto de dados para o treinamento da rede neural. Para acessar o vídeo, basta realizar a leitura do *QR Code* abaixo.



Apêndice I

Programa de Pós-processamento da *JeVois*

Neste apêndice encontra-se o programa desenvolvido no software *MATLAB*® para realizar o pós-processamento e tratamento dos dados obtidos pela câmera *JeVois* no programa apresentado no Anexo B durante os experimentos, além de gerar os gráficos necessários para análise dos movimentos do robô. No código é feita uma transformação de coordenadas, do sistema de referência da câmera para um sistema de referência global, e uma limpeza dos dados com retirada de outliers e aplicação de média móvel nas curvas, uma vez que o sensoriamento utilizado para a captura dos dados não oferece uma resolução livre de ruídos.

```

% Read data from JeVois

clc; clear; close all;

%% Read data from JeVois

% Using Matlab rotation functions: quaternions are defined as q = (w, x, y, z)
% Data from JeVois: quaternions are defined as q = (w, x, y, z)

A = readmatrix('dados/matheus_2021_09_14_17-30');

% The 3D marker position in the camera coordinate system
C_p = A(:,4:6)';
C_p = C_p*1e-3; % in m (to improve visualization)

% Quaternion that represents the 3d transformation from the camera coordinate
system
% to the marker coordinate system
q_OC = A(:,10:13);

%% Retirando os outliers dos dados capturados:

% Formar matriz juntando posição e quaternios
T = [q_OC,C_p'];

% Retirar outliers (Tout)
% Criar um vetor com as posições dos outliers (Vout)
[Tout,Vout] = rmoutliers(T);

% Substituir os outliers por uma média móvel de 3 pontos anteriores e 3
% pontos posteriores ao ponto fora da curva:
posx = C_p(1,:);
posy = C_p(2,:);
posz = C_p(3,:);
quatw = q_OC(:,1);
quatx = q_OC(:,2);
quaty = q_OC(:,3);
quatz = q_OC(:,4);

for j = 1:(length(Vout)-1)
    if Vout(j) == 1
        % posições:
        posx_novo(j) = (posx(j-3)+posx(j-2)+posx(j-1)+posx(j+1)+posx(j+2)+posx(j+3))/6;
        posy_novo(j) = (posy(j-3)+posy(j-2)+posy(j-1)+posy(j+1)+posy(j+2)+posy(j+3))/6;
        posz_novo(j) = (posz(j-3)+posz(j-2)+posz(j-1)+posz(j+1)+posz(j+2)+posz(j+3))/6;
        % quaternios:
        quatw_novo(j) = (quatw(j-3)+quatw(j-2)+quatw(j-1)+quatw(j+1)+quatw(j+2)+quatw(j+3))/6;
        quatx_novo(j) = (quatx(j-3)+quatx(j-2)+quatx(j-1)+quatx(j+1)+quatx(j+2)+quatx(j+3))/6;
        quaty_novo(j) = (quaty(j-3)+quaty(j-2)+quaty(j-1)+quaty(j+1)+quaty(j+2)+quaty(j+3))/6;
        quatz_novo(i) = (quatz(i-3)+quatz(i-2)+quatz(i-1)+quatz(i+1)+quatz(i+2)+quatz(i+3))/6;
    end
end

```

```

+quatz(j+3))/6;
    else
        % posições:
        posx_novo(j) = posx(j);
        posy_novo(j) = posy(j);
        posz_novo(j) = posz(j);
        % quatérnios:
        quatu_novo(j) = quatu(j);
        quatx_novo(j) = quatx(j);
        quaty_novo(j) = quaty(j);
        quatz_novo(j) = quatz(j);
    end
end

% Pegar matriz de posições já sem os outliers
C_p = [posx_novo ; posy_novo ; posz_novo];
C_p = C_p*1e3; % Passar para mm

% Pegar matriz de rotação já sem os outliers
q_OC = [quatu_novo ; quatx_novo ; quaty_novo ; quatz_novo]';

% Fazer uma média móvel das rotações sem outliers
q_OC = movmean(q_OC, size(q_OC(:,1)));

%% Data plots

% Object position in marker frame
O_p = zeros(size(C_p));
rpy_OC = zeros(3, size(C_p, 2));
for i = 1:size(C_p, 2)
    % quaternion (from marker frame to camera frame)
    q = q_OC(i, :);
    qn = quatnormalize(q);

    % Rotation matrix from camera frame to marker frame
    R_OC = quat2rotm(qn);

    % Roll, pitch and yaw angles
    rpy_OC(:, i) = (quat2eul(qn, 'ZYX')*180/pi)';

    % Object position in the marker frame
    O_p(:, i) = inv(R_OC)*C_p(:, i);
end

% translation
O_p = O_p - O_p(:, 1);

%% Criação do vetor tempo (t):

dt = 0.05; %Diferença de tempo em segundos entre cada frame capturado
t = zeros(length(A(:, 1))-1, 1);

for k = 2:length(A(:, 1))-1
    t(k) = t(k-1)+dt;
end

```

```
%% Plots

figure()
plot(t,-O_p(1,:)); hold on;
plot(t,O_p(2,:)); hold on;
plot(t,O_p(3,:)); grid on;
legend('x','y','z','Location','northwest');
xlabel('t [s]')
ylabel('pos [mm]')
xlim([0 22])
ylim([-50 200])
title('Object position in the object frame - Slow with angle')
```