



Universidade Federal de Ouro Preto  
Escola de Minas  
Engenharia de Controle e Automação



Júlia Elice Nunes Martins

**Modelagem, Controle e Prototipagem de um Sistema de Automação  
para Bolsa Válvula Máscara**

Ouro Preto, 2026

Júlia Elice Nunes Martins

**Modelagem, Controle e Prototipagem de um Sistema de Automação  
para Bolsa Válvula Máscara**

Monografia apresentada ao Curso de Engenharia de Controle e Automação da Universidade Federal de Ouro Preto como requisito parcial para obtenção do título de Engenheira de Controle e Automação.

Orientador: Prof. Dr. Paulo Marcos de Barros Monteiro

Coorientador: Prof. M.e João Carlos Vilela de Castro

Ouro Preto  
2026



MINISTÉRIO DA EDUCAÇÃO  
UNIVERSIDADE FEDERAL DE OURO PRETO  
REITORIA  
ESCOLA DE MINAS  
DEPARTAMENTO DE ENGENHARIA CONTROLE E  
AUTOMACAO



## FOLHA DE APROVAÇÃO

**Júlia Elice Nunes Martins**

### **Modelagem, Controle e Prototipagem de um Sistema de Automação para Bolsa Válvula Máscara**

Monografia apresentada ao Curso de Engenharia de Controle e Automação da Universidade Federal de Ouro Preto como requisito parcial para obtenção do título de Engenheiro de Controle e Automação

Aprovada em 29 de abril de 2026

#### Membros da banca

Dr. Paulo Marcos de Barros Monteiro - Orientador (Universidade Federal de Ouro Preto)  
Me. João Carlos Vilela de Castro - Coorientador (Universidade Federal de Ouro Preto)  
Dr. Agnaldo José da Rocha Reis - Convidado (Universidade Federal de Ouro Preto)  
Dra. Adrielle de Carvalho Santana - Convidada (Universidade Federal de Ouro Preto)

Paulo Marcos de Barros de Barros Monteiro, orientador do trabalho, aprovou a versão final e autorizou seu depósito na Biblioteca Digital de Trabalhos de Conclusão de Curso da UFOP em 30/04/2026



Documento assinado eletronicamente por **Paulo Marcos de Barros Monteiro, PROFESSOR DE MAGISTERIO SUPERIOR**, em 30/04/2026, às 13:37, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



A autenticidade deste documento pode ser conferida no site [http://sei.ufop.br/sei/controlador\\_externo.php?acao=documento\\_conferir&id\\_orgao\\_acesso\\_externo=0](http://sei.ufop.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0), informando o código verificador **1097228** e o código CRC **7E0CBFC9**.

**Referência:** Caso responda este documento, indicar expressamente o Processo nº 23109.005138/2026-30

SEI nº 1097228

R. Diogo de Vasconcelos, 122, - Bairro Pilar Ouro Preto/MG, CEP 35402-163  
Telefone: 3135591533 - [www.ufop.br](http://www.ufop.br)

# Agradecimentos

À minha avó, cuja luta por cada respiração fez da minha impotência um propósito.

# Resumo

Em ambulâncias e cenários de emergência com equipe reduzida, a ventilação manual com Bolsa Válvula Máscara (BVM) ocupa integralmente um socorrista que poderia estar realizando massagem cardíaca, administrando medicamentos ou monitorando sinais vitais. Além da indisponibilidade do profissional, a compressão manual prolongada resulta em fadiga e variabilidade no volume de ar entregue ao paciente, comprometendo a qualidade da ventilação. Este trabalho apresenta um sistema completo de automação para a BVM que integra, em um único protótipo, modelagem do sistema respiratório, projeto mecânico, controle em *malha fechada*, sistema de segurança e interface de operação. A garra mecânica, impressa em 3D e acionada por motor de passo, comprime a bolsa conforme parâmetros ventilatórios definidos pelo operador, enquanto um controlador proporcional integrativo regula a pressão alveolar com realimentação contínua. O controlador é projetado com um único conjunto de ganhos fixos e validado em três cenários clínicos distintos — adulto saudável, padrão obstrutivo (Doença Pulmonar Obstrutiva Crônica) e padrão restritivo (Síndrome do Desconforto Respiratório Agudo) —, demonstrando robustez ao atender os requisitos de desempenho em todas as condições, com tempos de acomodação compatíveis com os ciclos ventilatórios e erro em regime permanente nulo. A segurança do paciente é tratada como requisito central: o sistema incorpora intertravamentos, limites de pressão, alarmes de sobrepressão e de volume insuficiente, além de parada de emergência, garantindo operação dentro de faixas seguras mesmo em condições adversas. A modelagem do sistema respiratório por circuito resistência-complacência unicompartimental é validada em ambientes de simulação complementares, e o protótipo mecânico da garra é fabricado e integrado ao sistema de acionamento, com o firmware embarcado no microcontrolador. A validação experimental com instrumentação de pressão e volume e com carga real da bolsa é proposta como trabalho futuro. A solução desenvolvida demonstra a viabilidade de liberar um profissional para outras intervenções críticas durante o atendimento pré-hospitalar, contribuindo, em versões futuras validadas, para a segurança da ventilação e a eficiência da equipe de emergência.

**Palavras-chaves:** Bolsa Válvula Máscara. Ventilação Mecânica. Controlador Proporcional Integrativo. Modelagem Resistência-Complacência. Garra Mecânica.

# Abstract

In ambulances and emergency scenarios with reduced teams, manual ventilation with a Bag-Valve-Mask (BVM) fully occupies a rescuer who could otherwise be performing chest compressions, administering medication, or monitoring vital signs. In addition to the unavailability of the professional, prolonged manual compression leads to fatigue and variability in the air volume delivered to the patient, compromising ventilation quality. This work presents a complete automation system for the BVM that integrates, in a single prototype, respiratory system modeling, mechanical design, closed-loop control, a safety system, and an operator interface. The 3D-printed mechanical gripper, driven by a stepper motor, compresses the bag according to ventilatory parameters defined by the operator, while a proportional-integral controller regulates alveolar pressure with continuous feedback. The controller is designed with a single set of fixed gains and validated across three distinct clinical scenarios — healthy adult, obstructive pattern (Chronic Obstructive Pulmonary Disease), and restrictive pattern (Acute Respiratory Distress Syndrome) —, demonstrating robustness by meeting performance requirements in all conditions, with settling times compatible with ventilatory cycles and zero steady-state error. Patient safety is treated as a central requirement: the system incorporates interlocks, pressure limits, overpressure and insufficient volume alarms, and an emergency stop, ensuring operation within safe ranges even under adverse conditions. The respiratory system modeling through a single-compartment resistance-compliance circuit is validated in complementary simulation environments, and the mechanical prototype of the gripper is manufactured and integrated with the actuation system, with the firmware embedded in the microcontroller. Experimental validation with pressure and volume instrumentation and with actual bag load is proposed as future work. The developed solution demonstrates the feasibility of freeing a professional for other critical interventions during pre-hospital care, contributing, in future validated versions, to ventilation safety and emergency team efficiency.

**Key-words:** Bag-Valve-Mask. Mechanical Ventilation. Proportional-Integral Controller. Resistance-Compliance Modeling. Mechanical Gripper.

# Lista de ilustrações

Figura 1 – Bolsa Válvula Máscara ( <i>BVM</i> ) — Reanimador manual Foyomed. . . . .	22
Figura 2 – Circuito RC montado no Falstad. . . . .	32
Figura 3 – Formas de onda no Falstad - cenário saudável. . . . .	32
Figura 4 – Comparativo das formas de onda no Falstad para os três cenários clínicos. . . . .	33
Figura 5 – Diagrama de blocos do modelo RC em <i>malha aberta</i> no Simulink. . . . .	34
Figura 6 – Simulação em <i>malha aberta</i> — cenário saudável. . . . .	34
Figura 7 – Simulação em <i>malha aberta</i> — cenário DPOC. . . . .	36
Figura 8 – Simulação em <i>malha aberta</i> — cenário SDRA. . . . .	37
Figura 9 – Comparativo da pressão alveolar para os três cenários clínicos. . . . .	38
Figura 10 – Diagrama de blocos do sistema com controlador PI no Simulink. . . . .	45
Figura 11 – Polos em <i>malha fechada</i> para os três cenários clínicos e zero do controlador PI. . . . .	48
Figura 12 – Resposta ao degrau em <i>malha fechada</i> : controlador contínuo vs. discretizado - cenário saudável. . . . .	52
Figura 13 – Resposta em <i>malha fechada</i> para o cenário saudável: pressão, sinal de controle e volume pulmonar. . . . .	55
Figura 14 – Resposta em <i>malha fechada</i> para o cenário DPOC: pressão, sinal de controle e volume pulmonar. . . . .	56
Figura 15 – Resposta em <i>malha fechada</i> para o cenário SDRA: pressão, sinal de controle e volume pulmonar. . . . .	57
Figura 16 – Modelo 3D da garra mecânica projetada no Onshape - Geometria. . . . .	59
Figura 17 – Modelo 3D da garra mecânica projetada no Onshape - Conjunto. . . . .	60
Figura 18 – Protótipo da garra mecânica impressa em 3D, montada com o motor - Encaixe das engrenagens. . . . .	62
Figura 19 – Protótipo da garra mecânica impressa em 3D, montada com o motor - Completo. . . . .	62
Figura 20 – Exemplo de saída do Monitor Serial durante operação do protótipo. . . . .	67
Figura 21 – Bancada de testes com os subsistemas integrados: IHM, Arduino UNO, ponte H L298N e motor Nema 17. . . . .	70
Figura 22 – Verificação funcional — cenário saudável: pressão estimada, posição real e posição-alvo do motor. . . . .	72
Figura 23 – Verificação funcional — cenário DPOC: pressão estimada, posição real e posição-alvo do motor. . . . .	73
Figura 24 – Verificação funcional — cenário SDRA: pressão estimada, posição real e posição-alvo do motor. . . . .	74

# Lista de tabelas

Tabela 1 – Parâmetros do modelo RC por cenário clínico. . . . .	30
Tabela 2 – Analogia entre grandezas respiratórias e elétricas. . . . .	30
Tabela 3 – Indicadores das simulações em <i>malha aberta</i> por cenário. . . . .	38
Tabela 4 – Tempos inspiratório e expiratório por configuração de $f_R$ e I:E. . . . .	40
Tabela 5 – Ganhos PI e indicadores de desempenho por cenário clínico. . . . .	53
Tabela 6 – Limites de segurança do dispositivo. . . . .	63
Tabela 7 – Alarmes implementados no firmware. . . . .	65

# Lista de abreviaturas e siglas

ADRC	Controle por Rejeição Ativa de Perturbações ( <i>Active Disturbance Rejection Control</i> );
BVM	<i>Artificial Manual Breathing Unit</i> ;
APH	Atendimento Pré-Hospitalar;
BVM	Bolsa Válvula Máscara;
DPOC	Doença Pulmonar Obstrutiva Crônica;
ESO	Observador de Estado Estendido ( <i>Extended State Observer</i> );
FPGA	Arranjo de Portas Programável em Campo ( <i>Field Programmable Gate Array</i> );
IHM	Interface Homem Máquina;
IMC	Controle por Modelo Interno ( <i>Internal Model Control</i> );
PEEP	Pressão Expiratória Final Positiva ( <i>Positive End-Expiratory Pressure</i> );
PETG	Politereftalato de Etileno Glicol;
PI	Proporcional-Integral;
PID	Proporcional Integral Derivativo;
PLA	Ácido Polilático ( <i>Polylactic Acid</i> );
PSO	Otimização por Enxame de Partículas ( <i>Particle Swarm Optimization</i> );
PVC	Policloreto de Vinila ( <i>Polyvinyl Chloride</i> );
PWM	Modulação por Largura de Pulso ( <i>Pulse Width Modulation</i> );
RC	Resistência-Complacência;
SDRA	Síndrome do Desconforto Respiratório Agudo;
SMC	Controle por Modos Deslizantes ( <i>Sliding Mode Control</i> );
SRAM	Memória Estática de Acesso Aleatório ( <i>Static Random-Access Memory</i> );

# Lista de símbolos

$R$	Resistência das vias aéreas (cmH <sub>2</sub> O/(L/s))
$C$	Complacência pulmonar (mL/cmH <sub>2</sub> O)
$E$	Elastância pulmonar (cmH <sub>2</sub> O/mL), $E = 1/C$
$\tau$	Constante de tempo do sistema respiratório (s), $\tau = R \times C$
$\tau_{MF}$	Constante de tempo em malha fechada (s)
$P(t)$	Pressão nas vias aéreas (cmH <sub>2</sub> O)
$P_A(t)$	Pressão alveolar (cmH <sub>2</sub> O)
$P_0$	Pressão ao final da expiração / PEEP (cmH <sub>2</sub> O)
$P_{\text{ref}}$	Pressão de referência do controlador (cmH <sub>2</sub> O)
$P_{\text{med}}$	Pressão estimada / medida (cmH <sub>2</sub> O)
$P_{\text{aplicada}}$	Pressão aplicada pela compressão da bolsa (cmH <sub>2</sub> O)
$P_{\text{pico}}$	Pressão de pico durante a inspiração (cmH <sub>2</sub> O)
$P_{\text{max}}$	Pressão máxima de segurança (cmH <sub>2</sub> O)
$\Delta P$	Diferença de pressão (cmH <sub>2</sub> O)
$V(t)$	Volume pulmonar (mL)
$\dot{V}(t)$	Fluxo de ar (L/s)
$V_t$	Volume corrente (mL)
$V_{t,\text{ref}}$	Volume corrente de referência (mL)
$V_{\text{del}}$	Volume entregue conforme norma ISO (mL)
$T_{\text{ciclo}}$	Período total do ciclo respiratório (s)
$T_{\text{ins}}$	Tempo inspiratório (s)
$T_{\text{exp}}$	Tempo expiratório (s)
$T_{\text{pausa}}$	Pausa inspiratória (s)
$f_R$	Frequência respiratória (irpm)

$n$	Fator da relação I:E (adimensional)
$r(t)$	Referência ( <i>setpoint</i> )
$y(t)$	Saída (variável de processo)
$e(t)$	Erro entre referência e saída medida
$u(t)$	Sinal de controle
$K_p$	Ganho proporcional do controlador PI
$K_i$	Ganho integral do controlador PI
$K_d$	Ganho derivativo do controlador PID
$K$	Ganho estático da planta (adimensional)
$\alpha$	Polo desejado em malha fechada (rad/s)
$\zeta$	Fator de amortecimento (adimensional)
$\lambda$	Constante de tempo desejada em malha fechada, método IMC (s)
$G(s)$	Função de transferência da planta
$C(s)$	Função de transferência do controlador
$L(s)$	Função de transferência em malha aberta
$T(s)$	Função de transferência em malha fechada
$T$	Período de amostragem (s)
$z$	Variável do domínio discreto (transformada $z$ )
$a_0, a_1$	Coefficientes da equação de diferenças (Tustin)
$\theta$	Ângulo de compressão da garra (°)
$d$	Braço de alavanca do mecanismo da garra (m)
$F$	Força de compressão da bolsa (N)
$T_{\text{motor}}$	Torque do motor (N·m)
$\Delta t$	Intervalo de tempo discreto (s)

# Sumário

<b>1</b>	<b>Introdução</b>	<b>14</b>
1.1	Objetivos	15
1.1.1	Geral	15
1.1.2	Específicos	15
1.2	Metodologia	15
1.3	Justificativas e Relevância	16
1.4	Organização e estrutura	17
<b>2</b>	<b>Revisão de Literatura</b>	<b>18</b>
2.1	Sistema Respiratório	18
2.1.1	Ventilação pulmonar	18
2.2	Modelagem do Sistema Respiratório	20
2.3	Bolsa Válvula Máscara	21
2.3.1	Estrutura e Funcionamento da <i>BVM</i>	21
2.3.2	Parâmetros Clínicos da Ventilação com <i>BVM</i>	22
2.4	Controle Aplicado à Ventilação Mecânica	23
2.4.1	Arquiteturas de Controle e Estratégias Baseadas em Volume e Pressão	23
2.4.2	Controle PID e Otimização de Ganhos	24
2.4.3	Controle Inteligente	25
2.4.4	Controle Robusto e Estratégias de Rejeição de Perturbações	26
2.4.5	Aspectos Normativos e Confiabilidade	27
2.5	Projeto Mecânico e Prototipagem	27
<b>3</b>	<b>Desenvolvimento</b>	<b>29</b>
3.1	Materiais	29
3.2	Modelagem do Sistema Respiratório	29
3.2.1	Definição dos Parâmetros	29
3.2.2	Analogia Elétrica	30
3.2.3	Simulação no Falstad	31
3.2.4	Simulação no MATLAB	33
3.3	Projeto do Sistema de Controle	39
3.3.1	Definição dos Parâmetros Ventilatórios de Controle	40
3.3.1.1	Relação I:E	40
3.3.1.2	Frequência Respiratória ( $f_R$ )	41
3.3.1.3	Volume Corrente ( $V_t$ )	42
3.3.1.4	Pausa Inspiratória ( $T_{\text{pausa}}$ )	42
3.3.2	Estratégia de Controle de Frequência Respiratória	43
3.3.3	Estratégia de Controle de Volume	43

3.3.3.1	<i>Malha aberta</i> — Estimativa de volume pela geometria da bolsa . . . . .	43
3.3.3.2	<i>Malha fechada</i> — Controlador PI de pressão . . . . .	44
3.3.4	Sintonia do Controlador PI . . . . .	46
3.3.4.1	Função de Transferência da Planta . . . . .	46
3.3.4.2	Projeto por Alocação de Polos com Cancelamento de Polo e Zero . . . . .	46
3.3.4.3	Projeto Alternativo pelo Método IMC . . . . .	49
3.3.4.4	Discretização do Controlador — Transformação Bilinear (Tustin) . . . . .	50
3.3.4.5	Anti-Windup . . . . .	52
3.3.4.6	Procedimento de Sintonia e Validação . . . . .	52
3.3.4.7	Análise do Cenário DPOC . . . . .	54
3.3.4.8	Validação Cruzada de Parâmetros . . . . .	57
3.4	Projeto Mecânico . . . . .	58
3.4.1	Dimensionamento e Especificações . . . . .	58
3.4.2	Modelagem no Onshape . . . . .	58
3.4.3	Estimativa de Torque . . . . .	60
3.4.4	Fabricação e Testes Mecânicos . . . . .	61
3.5	Segurança e Interface Homem Máquina . . . . .	62
3.5.1	Limites de Segurança . . . . .	63
3.5.2	Intertravamentos e Sistema de Alarmes . . . . .	64
3.5.2.1	Parada de Emergência . . . . .	64
3.5.2.2	Sistema de Alarmes . . . . .	64
3.5.2.3	Detecção de Desconexão e Vazamento . . . . .	65
3.5.2.4	Registro de Eventos (Log de Alarmes) . . . . .	65
3.5.3	Interface Homem Máquina (IHM) . . . . .	65
3.5.3.1	Interface Serial (Monitor Serial / Serial Plotter) . . . . .	66
3.5.3.2	Interface Física . . . . .	68
3.5.4	Modo de Simulação Embarcada . . . . .	68
3.6	Validação . . . . .	69
3.6.1	Integração dos Subsistemas . . . . .	69
3.6.2	Verificação Funcional . . . . .	70
3.6.3	Limitações . . . . .	74
<b>4</b>	<b>Resultados</b> . . . . .	<b>76</b>
4.1	Resultados do Controlador PI . . . . .	76
4.2	Discussão . . . . .	77
4.2.1	Atendimento aos Objetivos . . . . .	77
4.2.2	Comparação com a Literatura . . . . .	78

4.2.3	Conformidade com Requisitos Normativos . . . . .	79
4.2.4	Efeito da Saturação e Desempenho Dinâmico . . . . .	80
4.2.5	Implicações Clínicas do <i>Overshoot</i> no Cenário DPOC . . . . .	80
4.2.6	Limitações da Validação . . . . .	80
<b>5</b>	<b>Conclusão . . . . .</b>	<b>82</b>
5.1	Trabalhos Futuros . . . . .	84
	<b>Referências . . . . .</b>	<b>85</b>
	<b>APÊNDICE A Firmware Principal . . . . .</b>	<b>88</b>
	<b>APÊNDICE B Módulo de Parser de Comandos Serial . . . . .</b>	<b>97</b>
	<b>APÊNDICE C Módulo de Alarmes . . . . .</b>	<b>102</b>

# 1 Introdução

Em situações de emergência como parada cardiorrespiratória ou insuficiência ventilatória, a ventilação manual com Bolsa Válvula Máscara (*BVM*) é um dos procedimentos mais frequentes no Atendimento Pré-Hospitalar (APH). Também conhecida como *BVM* (*Artificial Manual Breathing Unit*) ou Ressuscitador Auto-Insufável, a *BVM* é portátil, simples e de baixo custo. Porém, operá-la de forma contínua exige dedicação exclusiva de um socorrista, que precisa manter compressões rítmicas e consistentes por longos períodos (HAJJAR *et al.*, 2024; KIM *et al.*, 2024). A fadiga resultante compromete a regularidade do volume de ar entregue ao paciente, elevando o risco de hiperinsuflação ou hiperventilação (WARD; WARD; LEACH, 2012). Em ambulâncias, onde espaço e equipe são limitados, essa dedicação exclusiva reduz a disponibilidade para outras intervenções críticas: administração de medicamentos, massagem cardíaca, monitoramento de sinais vitais e preparo para transporte.

A interação entre ventilador e sistema respiratório é dinâmica e não linear, podendo variar ao longo do tempo (BATES, 2009). Na prática, dois parâmetros concentram essa variabilidade: a *Complacência Pulmonar* (C), que expressa a facilidade com que o pulmão se expande, e a *Resistência das Vias Aéreas* (R), que representa a oposição ao fluxo de ar. Ambos mudam não apenas entre pacientes, mas também no mesmo paciente ao longo do atendimento (ALMEIDA, 2007).

Esses parâmetros fundamentam tanto os modelos quanto os controladores que buscam fornecer volume de ar adequado com segurança. Para representar o comportamento mecânico do sistema respiratório, utiliza-se uma modelagem matemática baseada em *complacência* e *resistência* pulmonar, implementada por meio de um circuito elétrico RC (*Resistência-Complacência*) análogo. Como ensaios com pulmões reais apresentam limitações práticas e éticas, esse modelo computacional serve como ambiente de simulação para validação e testes do controlador sob condições controladas e reproduzíveis (BATES, 2009; ALMEIDA, 2007).

Diante das incertezas inerentes ao sistema respiratório, o controlador precisa ajustar dinamicamente volume, velocidade e frequência de compressão, com realimentação contínua para garantir estabilidade e tolerância a falhas (PEREIRA; GRANDE, 2021). Modelar a mecânica pulmonar e projetar o controlador responsável por esses ajustes são, portanto, passos centrais para um sistema semiautônomo capaz de operar em diferentes cenários clínicos.

## 1.1 OBJETIVOS

### 1.1.1 Geral

Projetar e implementar um sistema de automação para a Bolsa Válvula Máscara (*BVM*), composto por uma garra mecânica impressa em 3D com controle em *malha fechada*, capaz de comprimir a bolsa respeitando intervalos de tempo e volume adequados a diferentes condições clínicas. O trabalho abrange a modelagem matemática do sistema respiratório humano por meio de um circuito RC unicompartimental, a síntese e validação do controlador em ambiente de simulação, o projeto e fabricação do mecanismo de compressão, e a implementação de um sistema de segurança e interface de operação embarcados em microcontrolador.

### 1.1.2 Específicos

- Compreender os princípios fisiológicos da ventilação manual com *BVM*, incluindo volumes, frequências e tempos inspiratórios adequados, e modelar a dinâmica pulmonar;
- Projetar a garra para impressão em impressora 3D;
- Implementar o controle em tempo real que ajuste o padrão do motor para compressão da *BVM*;
- Implementar supervisão e intertravamentos (limites de pressão, alarmes) e propor mecanismos de detecção de desconexões e fugas para implementação futura;
- Projetar uma Interface Homem Máquina (IHM) para modificações do sistema em tempo real.

## 1.2 METODOLOGIA

A metodologia empregada neste trabalho caracteriza-se como pesquisa aplicada, de abordagem quantitativa, estruturada em cinco etapas agrupadas em três eixos principais, descritos a seguir.

O **Eixo I: Modelagem** compreende a representação matemática do sistema respiratório por meio de um circuito RC unicompartimental. Nesta etapa, o modelo é implementado e validado em dois ambientes de simulação complementares - Falstad e MATLAB -, contemplando cenários clínicos de adulto saudável, DPOC (Doença Pulmonar Obstrutiva Crônica) e SDRA (Síndrome do Desconforto Respiratório Agudo). O objetivo é obter uma planta virtual confiável que sirva de base para o projeto e a validação do controlador.

O **Eixo II: Mecânica e Atuação** abrange o projeto mecânico da garra impressa em 3D e a implementação do sistema de acionamento com motor e ponte H. Essa etapa inclui o dimensionamento do mecanismo de compressão, a modelagem no Onshape, o cálculo de torque e a fabricação do protótipo, garantindo que a garra seja capaz de realizar movimentos de compressão rítmicos de forma controlada e repetível.

O **Eixo III: Controle e Validação** engloba a síntese do controlador PI para regulação do padrão ventilatório - frequência respiratória, volume corrente, relação I:E e pausa inspiratória -, o projeto do sistema de segurança com intertravamentos e alarmes, e a Interface Homem Máquina (IHM). A validação consiste na verificação do funcionamento integrado dos três eixos e na análise dos resultados das simulações com base nos parâmetros fisiológicos de referência (BATES, 2009).

O desenvolvimento é conduzido em etapas sucessivas, conforme descrito a seguir:

1. Modelagem do Sistema Respiratório: desenvolvimento de um modelo unicompartmental do tipo RC e realização de simulações computacionais (Falstad e MATLAB) para representar e validar o comportamento ventilatório nos três cenários clínicos.
2. Projeto Mecânico: dimensionamento e construção da garra mecânica responsável pela compressão controlada da *BVM*, incluindo modelagem 3D, estimativa de torque e fabricação por impressão 3D.
3. Síntese de Controle: projeto do controlador PI por alocação de polos, discretização pela transformação bilinear de Tustin e validação em *malha fechada* no MATLAB.
4. Segurança e IHM: implementação dos intertravamentos, sistema de alarmes e interface de operação para monitoramento e ajuste de parâmetros em tempo real.
5. Integração e Testes: montagem do protótipo completo, integração dos subsistemas e verificação funcional do conjunto mecânico-eletrônico, com análise dos resultados das simulações.

### 1.3 JUSTIFICATIVAS E RELEVÂNCIA

Duas perspectivas complementares justificam a automação da ventilação com *BVM*. No aspecto social, liberar o socorrista da compressão manual permitiria que ele se dedicasse a outras intervenções críticas (administração de medicamentos, massagem cardíaca, monitoramento de sinais vitais), aumentando a eficiência da equipe em cenários com espaço e pessoal limitados (BORRELLO, 2021; KIM *et al.*, 2024). Ao mesmo tempo, o uso de controle em *malha fechada* com realimentação de pressão aplica conhecimentos de engenharia a um sistema biomédico crítico.

Do ponto de vista técnico e acadêmico, o trabalho desenvolve uma solução de baixo custo que reúne modelagem do sistema respiratório, controle PI e projeto mecânico simplificado em um único protótipo, demonstrando que é viável automatizar a *BVM* sem componentes de alto custo ou infraestrutura hospitalar.

## 1.4 ORGANIZAÇÃO E ESTRUTURA

Neste trabalho, a organização segue a seguinte forma: o Capítulo 2 apresenta a revisão de literatura; o Capítulo 3 descreve o desenvolvimento do protótipo, incluindo modelagem, controle, projeto mecânico e sistema de segurança; o Capítulo 4 apresenta os resultados das simulações e a discussão; e, por fim, as Conclusão sintetizam as contribuições, limitações e direções para trabalhos futuros.

## 2 Revisão de Literatura

Neste capítulo apresenta-se os principais conceitos que embasam o desenvolvimento do presente trabalho, com uma revisão da literatura pertinente ao desenvolvimento de automatização de um respirador baseado no *BVM*. São abordados os seguintes temas: o sistema respiratório humano, com ênfase na ventilação pulmonar; a modelagem matemática do sistema respiratório por meio do modelo unicompartmental RC; a estrutura e o funcionamento da *BVM*, incluindo seus parâmetros clínicos; as estratégias de controle aplicadas à ventilação mecânica, desde arquiteturas baseadas em volume e pressão até técnicas inteligentes e robustas; os aspectos normativos e de confiabilidade; e o projeto mecânico e a prototipagem de dispositivos de compressão.

### 2.1 SISTEMA RESPIRATÓRIO

O sistema respiratório humano constitui um conjunto integrado de órgãos e estruturas cuja função principal é realizar as trocas gasosas entre o organismo e o ambiente, assegurando o suprimento de oxigênio ( $O_2$ ) e a remoção de dióxido de carbono ( $CO_2$ ), que é a hematose. Três processos interdependentes compõem a respiração: a ventilação pulmonar (entrada e saída de ar dos pulmões), a hematose e o transporte dos gases pelo sangue (WARD; WARD; LEACH, 2012).

Neste trabalho, a ventilação pulmonar é o mais diretamente relacionado ao mecanismo de compressão do dispositivo proposto, e por isso recebe ênfase. A hematose e o transporte de gases, embora constituam etapas subseqüentes do mesmo processo integrado, não são abordados em detalhe. Para uma descrição aprofundada, recomenda-se consultar West e Luks (2021) e Hall e Hall (2021).

#### 2.1.1 Ventilação pulmonar

A ventilação pulmonar é o processo mecânico pelo qual o ar entra e sai dos pulmões, envolvendo pressões, volumes e fluxos que dependem das propriedades físicas dos pulmões, das vias aéreas e da caixa torácica (WEST; LUKS, 2021). Em termos simples, o ar entra nos pulmões porque os músculos respiratórios expandem o tórax, criando uma pressão interna menor que a atmosférica; quando esses músculos relaxam, o ar sai naturalmente porque o pulmão tende a voltar ao seu tamanho original (HALL; HALL, 2021).

A *Complacência* expressa o quão facilmente o sistema respiratório se expande quando submetido a uma variação de pressão. Um pulmão com *complacência* elevada aumenta seu volume com menor necessidade de pressão, enquanto um pulmão com *com-*

*placência* reduzida se comporta como um sistema mais rígido, exigindo maior esforço inspiratório. A *elastância*, por sua vez, refere-se à tendência do pulmão de retornar ao estado inicial após ser distendido, sendo o comportamento oposto ao da *complacência* (LUCANGELO; BERNABÉ; BLANCH, 2005). Essas propriedades determinam o componente elástico do trabalho respiratório, isto é, a energia gasta para distender pulmões e caixa torácica (HESS, 2014). Cabe observar que a *elastância*, sendo o inverso da *complacência* ( $E = 1/C$ ), já está implicitamente representada no modelo RC adotado neste trabalho: ao utilizar a *complacência* como parâmetro do circuito análogo, a *elastância* é automaticamente contemplada, uma vez que ambas descrevem a mesma propriedade mecânica sob perspectivas complementares.

Complementarmente, a ventilação é influenciada pela *Resistência* das Vias Aéreas, que representa a oposição ao fluxo de ar ao longo da traqueia, brônquios e bronquíolos. Quando a *resistência* aumenta (por broncoconstrição, inflamação, edema ou presença de secreções), é necessário gerar maior diferença de pressão para manter o mesmo fluxo, o que eleva o componente resistivo do trabalho respiratório (LUCANGELO; BERNABÉ; BLANCH, 2005). Mesmo com *complacência* preservada a ventilação pode se tornar mecanicamente custosa se a *resistência* ao fluxo for elevada.

Há também uma relação com os volumes pulmonares e com o ar que permanece nos pulmões após uma expiração tranquila, frequentemente associado à capacidade residual funcional (WEST; LUKS, 2021). Esse “estoque” de ar tem importância na estabilidade do sistema, na distribuição da ventilação e na manutenção da troca gasosa entre as respirações. Alterações nesses volumes podem favorecer heterogeneidade ventilatória, especialmente quando diferentes regiões do pulmão ventilam de forma desigual (BATES, 2009). Para a automatização da *BVM*, essa heterogeneidade é relevante porque o dispositivo entrega um volume global ao paciente, sem possibilidade de direcionar o ar para regiões específicas do pulmão. Assim, o controlador deve considerar que o volume entregue pode não se distribuir uniformemente, especialmente em pacientes com patologias que alteram a capacidade residual funcional, reforçando a importância de respeitar limites de pressão e de incluir pausa inspiratória quando necessário para permitir a redistribuição do ar entre regiões com diferentes constantes de tempo (BATES, 2009; BRUNNER; PIQUILLOUD *et al.*, 2024).

Do ponto de vista energético, o trabalho respiratório se compõe de um componente elástico (relacionado à *complacência* e à distensão do sistema) e um componente resistivo (relacionado à *resistência* ao fluxo) (HESS, 2014). Quando a *resistência* aumenta, o esforço tende a se manifestar principalmente durante a expiração; já quando a *complacência* é reduzida, o custo mecânico se concentra na inspiração, devido à maior pressão necessária para expandir os pulmões (LUCANGELO; BERNABÉ; BLANCH, 2005).

Além dos aspectos estáticos (quanto o pulmão expande), há aspectos dinâmicos (quão rápido ele expande e esvazia). A constante de tempo ( $\tau$ ), aproximada por  $\tau \approx R \times C$ ,

quantifica essa velocidade: regiões com constante de tempo mais alta tendem a encher e esvaziar mais lentamente, o que pode contribuir para aprisionamento aéreo em algumas condições clínicas (BRUNNER; PIQUILLOUD *et al.*, 2024).

Essas propriedades mecânicas são determinantes para o projeto do dispositivo automatizado. Em condições predominantemente obstrutivas (DPOC, asma), a *resistência* elevada favorece expiração prolongada e aprisionamento de ar, de modo que o controlador deve permitir tempos expiratórios mais longos. Já em condições restritivas (SDRA, fibrose pulmonar), a *complacência* reduzida exige maiores pressões inspiratórias para atingir o volume corrente desejado, e o controlador precisa respeitar limites de pressão de pico para evitar barotrauma.

## 2.2 MODELAGEM DO SISTEMA RESPIRATÓRIO

Modelar matematicamente o sistema respiratório é um passo essencial no desenvolvimento de dispositivos de ventilação, pois permite prever o comportamento pulmonar em diferentes condições e projetar controladores mais confiáveis. O modelo mais difundido é o unicompartimental linear Bates (2009), que representa o pulmão como um circuito RC (*Resistência-Complacência*), descrito pela equação:

$$P(t) = R \cdot \dot{V}(t) + \frac{1}{C} \cdot V(t) + P_0 \quad (2.1)$$

onde  $P(t)$  é a pressão nas vias aéreas,  $R$  é a *resistência* das vias aéreas,  $\dot{V}(t)$  é o fluxo de ar,  $C$  é a *complacência* pulmonar,  $V(t)$  é o volume e  $P_0$  é a Pressão Expiratória Final Positiva (PEEP, do inglês Positive End-Expiratory Pressure), que corresponde à pressão mantida nas vias aéreas ao final da expiração para evitar o colapso alveolar.

Ao investigar o modelo dinâmico da *elastância* pulmonar, Almeida (2007) demonstrou que os parâmetros  $R$  e  $C$  variam significativamente entre pacientes e ao longo do tempo em um mesmo paciente, especialmente em condições patológicas como SDRA e DPOC. Essa variabilidade reforça a necessidade de controladores adaptativos capazes de ajustar seus parâmetros em tempo real.

Na obra de referência sobre mecânica pulmonar, Bates (2009) formalizou extensivamente o modelo unicompartimental e suas extensões, incluindo modelos multicompartimentais e modelos com comportamento não linear da *elastância*. Apesar de sua simplicidade, o modelo RC unicompartimental captura adequadamente a dinâmica respiratória para fins de controle de ventilação em muitas situações clínicas.

Ao comparar modelos lineares e não lineares, Faustino *et al.* (2019) avaliaram três modelos matemáticos aplicados à mecânica respiratória durante broncoconstrição. A conclusão foi que, embora o modelo linear unicompartimental seja o mais simples e

difundido, modelos não lineares oferecem melhor representação em condições patológicas específicas.

Em uma revisão sobre as propriedades mecânicas dinâmicas do sistema respiratório, [Bates e Irvin \(2011\)](#) abordaram modelagem tanto no domínio do tempo quanto da frequência, destacando que o monitoramento contínuo dessas propriedades já é possível em alguns ventiladores mecânicos modernos, graças à evolução do poder computacional. Esse modelo serve como base para estratégias de controle em ventiladores, permitindo regular parâmetros clínicos de interesse como o volume corrente ( $V_t$ ), a frequência respiratória, o tempo inspiratório ( $T_{ins}$ ), a pressão de pico e a PEEP. Monitorar e ajustar essas variáveis é indispensável para garantir ventilação eficaz, segura e adaptada às condições clínicas do paciente.

## 2.3 BOLSA VÁLVULA MÁSCARA

### 2.3.1 Estrutura e Funcionamento da *BVM*

A Bolsa Válvula Máscara, consiste, estruturalmente, em uma bolsa auto-inflável, válvulas unidirecionais, reservatório de oxigênio e máscara facial ou conexão para tubo endotraqueal. A bolsa auto-inflável, geralmente fabricada em silicone ou PVC, é o componente principal que armazena e fornece o ar ou mistura gasosa ao paciente. As válvulas unidirecionais garantem que o fluxo de ar ocorra apenas na direção correta, evitando a reinalação de gases expirados pelo paciente. O princípio de funcionamento do dispositivo baseia-se na compressão manual da bolsa pelo operador, que força o ar através das válvulas em direção ao paciente. Durante a fase de compressão (inspiração), a válvula de entrada se fecha e a válvula de saída se abre, direcionando o ar para os pulmões do paciente. Na fase de descompressão (expiração), a bolsa retorna à sua forma original, a válvula de saída se fecha e a válvula de entrada se abre, permitindo que a bolsa se encha novamente com ar ambiente ou oxigênio suplementar do reservatório. A [Figura 1](#) apresenta o reanimador manual Foyomed utilizado neste trabalho.



Figura 1 – Bolsa Válvula Máscara (*BVM*) — Reanimador manual Foyomed.

Fonte: autoria própria.

### 2.3.2 Parâmetros Clínicos da Ventilação com *BVM*

A ventilação com *BVM* deve respeitar limites que garantam a segurança do paciente. De forma simplificada, o objetivo é insuflar os pulmões com um volume de ar suficiente para expandir visivelmente o tórax, sem exagerar na quantidade ou na velocidade, pois tanto a falta quanto o excesso de ar podem ser prejudiciais. A cada compressão da bolsa, o dispositivo automatizado deve entregar o ar de forma suave, em cerca de um segundo, observando se o peito do paciente se eleva (HAJJAR *et al.*, 2024).

Em termos mais técnicos, durante a ressuscitação cardiorespiratória:

- Pacientes adultos sem via aérea avançada, recomenda-se ventilar com volumes de  $\approx$  500-600 mL (6-7 mL/kg), por  $\sim$ 1 s, apenas o suficiente para elevar visivelmente o tórax, a cada 30 compressões torácicas;
- Pacientes adultos com via aérea avançada, ventilar 1 vez a cada 6 s ( $\sim$ 10 irpm) de forma assíncrona, sempre evitando hiperventilação.

Projetos que visam automatizar a compressão do *BVM* precisam respeitar a ISO 10651-4 (INTERNATIONAL ORGANIZATION FOR STANDARDIZATION, 2023), já que várias limitações podem ser observadas no mesmo: alta variabilidade do volume/fluxo/pressão gerados manualmente, risco de hiperventilação, distensão gástrica e barotrauma, além de

falhas por vedação inadequada da máscara. Desse modo, é crucial que seja respeitado o parâmetro dos conectores (15 F/22 M), que designa o diâmetro interno de 15 mm na conexão fêmea e o diâmetro externo de 22 mm na conexão macho, conforme padronização internacional para interfaces de equipamentos respiratórios, garantindo compatibilidade entre a *BVM* e os dispositivos de via aérea do paciente. Além disso, a norma estabelece limitação de pressão, máxima zona morta, volumes mínimos entregues por faixa de massa corporal e robustez ambiental (choque, imersão, temperatura/umidade). Para pacientes com massa corporal inferior a 10 kg, a norma exige pressão limitada a 45 cmH<sub>2</sub>O e capacidade de gerar ao menos 30 cmH<sub>2</sub>O; para pacientes com massa corporal superior a 40 kg, especifica volume entregue ( $V_{del}$ ) mínimo de 600 mL em condições de teste. Esses requisitos normativos são utilizados como referência para o dimensionamento e validação do protótipo no capítulo [Desenvolvimento](#) (3).

## 2.4 CONTROLE APLICADO À VENTILAÇÃO MECÂNICA

Controlar ventiladores mecânicos é um problema desafiador, dada a natureza não linear e variante no tempo do sistema respiratório [Bates \(2009\)](#). Segundo [Pinto \(2005\)](#), o próprio corpo humano já funciona como um sistema de controle em *malha fechada*: os centros respiratórios no tronco encefálico atuam como controlador, recebendo informações dos quimiorreceptores centrais e periféricos sobre os níveis de O<sub>2</sub> e CO<sub>2</sub> no sangue, e ajustando frequência e profundidade da respiração para manter esses gases dentro de faixas fisiológicas adequadas. Os pulmões e as vias aéreas constituem a planta controlada, enquanto os quimiorreceptores fecham a malha de realimentação. Essa perspectiva fundamenta a aplicação de técnicas de controle automático à ventilação mecânica, e diversas estratégias têm sido propostas na literatura para garantir entrega segura e precisa de volume e pressão mesmo diante de variações nos parâmetros pulmonares do paciente.

### 2.4.1 Arquiteturas de Controle e Estratégias Baseadas em Volume e Pressão

Uma taxonomia para classificar os modos de ventilação mecânica foi proposta por [Chatburn e Mireles-Cabodevila \(2011\)](#), com base em variáveis de controle, fase e lógica condicional. Essa classificação é fundamental para o projeto de controladores, pois define se o sistema opera em modo controlado por volume (VCV) ou por pressão (PCV), cada um com implicações distintas para a estratégia de controle. No modo VCV, o controlador regula o fluxo de ar para entregar um volume corrente pré-definido; no modo PCV, mantém uma pressão inspiratória constante, sendo o volume resultante dependente da mecânica pulmonar do paciente. Para uma ventilação baseada na *BVM*, a escolha do modo de controle influencia diretamente o projeto do mecanismo de compressão e a instrumentação necessária.

Já [Borrello \(2021\)](#) investigou o controle de ventiladores *BVM* automatizados sem modelo, utilizando ciclos controlados por volume e por pressão. Seus resultados indicam que é possível obter desempenho satisfatório com controladores relativamente simples, desde que haja realimentação adequada de pressão e volume. Para projetos de baixo custo, esse achado é relevante: a ausência de um modelo preciso da planta não impede a obtenção de ventilação segura, desde que o sensoriamento seja adequado.

## 2.4.2 Controle PID e Otimização de Ganhos

Dentre as estratégias de controle, o PID é a mais difundida em ventiladores mecânicos, pela simplicidade de implementação e eficácia comprovada. A ação proporcional (P) responde ao erro instantâneo entre referência e saída medida, a ação integral (I) elimina o erro em regime permanente acumulando o erro ao longo do tempo, e a ação derivativa (D) antecipa variações futuras do erro com base em sua taxa de variação. A lei de controle PID é descrita por:

$$u(t) = K_p \cdot e(t) + K_i \cdot \int_0^t e(\tau) d\tau + K_d \cdot \frac{de(t)}{dt} \quad (2.2)$$

onde  $u(t)$  é o sinal de controle,  $e(t)$  é o erro entre a referência e a saída, e  $K_p$ ,  $K_i$  e  $K_d$  são, respectivamente, os ganhos proporcional, integral e derivativo.

Embora conceitualmente simples, o desempenho do PID depende fortemente da escolha adequada dos três ganhos. Métodos clássicos de sintonia baseiam-se em ensaios experimentais com a planta real e fornecem valores iniciais razoáveis, porém frequentemente exigem ajustes manuais posteriores. Na ventilação mecânica, essa sintonia é especialmente desafiadora: como *resistência* e *complacência* variam entre pacientes e ao longo do tempo em um mesmo paciente, ganhos sintonizados para uma condição podem apresentar desempenho insatisfatório em outra.

Para contornar essa limitação, [Ghafoor, Hayat e Hussain \(2021\)](#) propuseram o uso de Otimização por Enxame de Partículas (PSO) para sintonizar automaticamente os ganhos PID em sistemas de ventilação artificial. O PSO é um algoritmo de otimização bioinspirado que simula o comportamento coletivo de um bando de pássaros em busca de alimento: cada “partícula” representa uma combinação candidata de ganhos ( $K_p$ ,  $K_i$ ,  $K_d$ ) e se movimenta pelo espaço de busca, ajustando sua posição com base na melhor solução que ela própria encontrou e na melhor solução encontrada pelo grupo como um todo. A cada iteração, as partículas convergem para a região do espaço de parâmetros que minimiza uma função de custo, tipicamente o erro quadrático integrado ou o erro absoluto integrado entre a pressão de referência e a pressão medida nas vias aéreas.

A principal vantagem do PSO em relação a métodos analíticos de sintonia é que ele não requer um modelo matemático preciso da planta: basta simular o sistema com diferentes

combinações de ganhos e avaliar o desempenho resultante. Isso o torna especialmente adequado para o sistema respiratório, cuja dinâmica é não linear e variante no tempo. [Ghafoor, Hayat e Hussain \(2021\)](#) propuseram a utilização do modelo RC como planta de simulação e os resultados demonstraram que os ganhos obtidos via PSO proporcionaram melhor rastreamento da pressão de vias aéreas em comparação com os ganhos obtidos pelo método de Ziegler-Nichols, uma técnica clássica de sintonia de controladores PID que se baseia em levar o sistema ao limiar de oscilação sustentada para, a partir da frequência e do ganho críticos observados, calcular os ganhos do controlador por meio de relações empíricas tabeladas, com menor *overshoot* e menor tempo de acomodação.

Em uma abordagem distinta, [Al-Qassar et al. \(2023\)](#) implementou o controlador em FPGA (*Field Programmable Gate Array*). Um FPGA é um circuito integrado digital reconfigurável, composto por blocos lógicos que podem ser programados para executar operações específicas em paralelo. Diferentemente de um microcontrolador convencional (como o Arduino UNO, que executa instruções sequencialmente), o FPGA processa múltiplas operações simultaneamente em *hardware* dedicado, eliminando a latência associada à execução sequencial de código e ao sistema operacional.

Para ventiladores mecânicos, isso se traduz em três vantagens práticas: velocidade de processamento maior, com taxas de amostragem na ordem de microssegundos; eliminação de *jitter* (variação no tempo de execução entre ciclos), o que garante funcionamento correto das ações integral e derivativa; e redução de erros numéricos acumulados, já que a aritmética pode ser implementada em ponto fixo com precisão controlada. Os resultados de [Al-Qassar et al. \(2023\)](#) mostraram melhor desempenho dinâmico do controlador PID em FPGA em comparação com a implementação em microcontrolador, especialmente em cenários que exigem resposta rápida a variações bruscas de pressão.

### 2.4.3 Controle Inteligente

A lógica fuzzy é amplamente aplicada como forma de lidar com as não linearidades e incertezas do sistema respiratório, reduzindo a necessidade de um modelo matemático preciso. Diferentemente do controle clássico, que opera com valores numéricos exatos, o controle fuzzy utiliza variáveis linguísticas e regras do tipo SE-ENTÃO para mapear entradas em ações de controle. Por exemplo, em vez de calcular algebricamente o sinal de controle a partir do erro numérico, um controlador fuzzy pode conter regras como “SE a pressão está um pouco acima do alvo E está subindo devagar, ENTÃO reduza levemente o fluxo”. Essas regras são definidas por funções de pertinência que atribuem graus de verdade a cada proposição linguística, e o sinal de controle final resulta da combinação ponderada de todas as regras ativadas simultaneamente (processo de defuzzificação). Essa abordagem é particularmente vantajosa quando o modelo matemático da planta é impreciso ou quando os parâmetros variam de forma imprevisível, situação comum no sistema respiratório.

No trabalho de [Alam et al. \(2021b\)](#), um controlador Fuzzy PID foi desenvolvido para rastreamento da pressão de vias aéreas durante ventilação mecânica. Nesse trabalho, o sistema respiratório foi representado por um modelo em cadeia denominado *blower* mangueira paciente, que descreve o caminho completo do ar desde a fonte de pressão até os pulmões. O *blower* é um ventilador centrífugo (turbina) que gera o fluxo de ar pressurizado; a mangueira (circuito respiratório) conecta o *blower* ao paciente e introduz *resistência* ao fluxo e volume de compressão adicional; e o paciente é representado pelo modelo unicompartimental RC com *complacência* não linear, isto é, a *complacência* varia em função do volume pulmonar instantâneo, tornando-se menor à medida que o pulmão se aproxima de sua capacidade máxima. Essa modelagem em cadeia é mais realista do que considerar apenas o modelo RC isolado, pois incorpora as perdas de pressão e os atrasos introduzidos pelo circuito respiratório entre a fonte de ar e os pulmões.

O controlador Fuzzy-PID proposto combina a estrutura clássica do PID com um mecanismo fuzzy de ajuste *online* dos ganhos. Em vez de manter  $K_p$ ,  $K_i$  e  $K_d$  fixos, o sistema fuzzy monitora o erro de pressão e sua taxa de variação e ajusta os ganhos em tempo real conforme regras linguísticas pré-definidas. Quando o erro é grande, os ganhos são aumentados para acelerar a resposta; quando o erro é pequeno e estável, os ganhos são reduzidos para evitar oscilações. Os resultados demonstraram que o controlador Fuzzy-PID apresentou melhor desempenho que o PID convencional em cenários com variação de *complacência*, reduzindo o *overshoot* de pressão e melhorando o tempo de acomodação.

Posteriormente, [Alam et al. \(2021a\)](#) propuseram um controlador fuzzy adaptativo por modos deslizantes (*Adaptive Fuzzy Sliding Mode Control*) para ventilação controlada por pressão. O controle por modos deslizantes (SMC) é uma técnica robusta que força a trajetória do sistema a “deslizar” ao longo de uma superfície pré-definida no espaço de estados, tornando o comportamento independente dos parâmetros exatos da planta. Sua principal limitação é o *chattering* (oscilações de alta frequência no sinal de controle). A contribuição de Alam et al. consiste em substituir a comutação descontínua por um sistema fuzzy adaptativo, que suaviza a transição e ajusta seus parâmetros automaticamente por uma lei baseada na teoria de estabilidade de Lyapunov. Os resultados mostraram que o controlador mantém a pressão de pico abaixo de valores críticos mesmo com parâmetros desconhecidos do paciente e vazamentos no circuito.

Ambos os trabalhos de Alam et al. ilustram uma tendência no controle de ventiladores: a combinação de técnicas clássicas com inteligência computacional para obter controladores robustos e adaptáveis.

#### 2.4.4 Controle Robusto e Estratégias de Rejeição de Perturbações

A busca por desempenho e estabilidade mesmo diante de parâmetros incertos ou variáveis é o foco do controle robusto. Na ventilação mecânica, isso se traduz em manter a

entrega de volume e pressão dentro de limites seguros independentemente das variações de *complacência* e *resistência* do paciente.

Na revista *Mathematical Biosciences and Engineering*, [Hussain et al. \(2022\)](#) propuseram uma abordagem cascadeada de controle robusto utilizando modos deslizantes de ordem fracionária para regulação da pressão de vias aéreas. O trabalho utilizou técnicas de redução de modelo do *blower* para incorporar sua dinâmica no projeto do controlador principal. Os resultados indicaram menor *chattering* (oscilações de alta frequência no sinal de controle) e melhor rejeição de perturbações em comparação com controladores por modos deslizantes de ordem inteira, mantendo a pressão de vias aéreas dentro de limites seguros mesmo com variações de até 30% nos parâmetros pulmonares.

Já [Garmendia et al. \(2021\)](#) apresentaram o projeto mecatrônico de um ventilador baseado em *BVM* com controle por rejeição ativa de perturbações (ADRC, *Active Disturbance Rejection Control*). O ADRC é uma técnica que estima e compensa perturbações em tempo real sem necessidade de um modelo preciso da planta, utilizando um observador de estado estendido (ESO). O sistema utiliza uma correia para comprimir a bolsa e incorpora sensoriamento de pressão para realimentação, garantindo ventilação adequada mesmo diante de variações na mecânica pulmonar do paciente. Nos testes realizados, o ADRC superou o desempenho de controladores PID convencionais com diferentes valores de *complacência* e *resistência*.

#### 2.4.5 Aspectos Normativos e Confiabilidade

No âmbito normativo, [Lopes \(2023\)](#) analisa o desempenho essencial de ventiladores pulmonares no modo volume controlado segundo a norma ABNT NBR ISO 80601-2-12, fornecendo parâmetros de referência para garantir que sistemas automatizados atendam aos padrões de segurança exigidos. Complementarmente, [Uechi \(2012\)](#) discute a confiabilidade metrológica de ventiladores pulmonares para cuidados críticos, enfatizando a importância da calibração e manutenção dos sensores e atuadores, aspecto fundamental para assegurar que o sistema opere dentro de limites seguros.

## 2.5 PROJETO MECÂNICO E PROTOTIPAGEM

O projeto mecânico de dispositivos automatizados de compressão da *BVM* envolve o dimensionamento do mecanismo de atuação, a seleção de materiais e a validação estrutural, considerando as forças necessárias para comprimir a bolsa e a durabilidade frente a ciclos repetitivos. Diversos trabalhos na literatura abordam soluções mecânicas para esse fim, variando desde sistemas por engrenagens até mecanismos por correia e alavanca.

O projeto SVASTA, desenvolvido pela comunidade espacial indiana e descrito por [Venkatesan et al. \(2020\)](#), utilizou um sistema de atuação por engrenagens para comprimir

o BVM, garantindo longa vida cíclica à bolsa e controle fino dos parâmetros ventilatórios. O sistema incorpora controlador digital com *malha fechada*, representando uma abordagem robusta para automação de baixo custo.

O projeto mecânico do ventilador COMBIOVENT, apresentado por [Espinoza et al. \(2024\)](#), detalhou o dimensionamento do bloco de ventilação utilizando modelos 3D, incluindo análise de esforços mecânicos e seleção de atuadores. Os autores enfatizaram a importância de considerar a fadiga mecânica da bolsa BVM no dimensionamento do mecanismo de compressão.

## 3 Desenvolvimento

### 3.1 MATERIAIS

- Reanimador Manual Foyomed adulto (*BVM* em PVC, auto-inflável, registro Anvisa nº 10150470481);
- Peças prototipadas em impressora 3D (filamento PETG);
- Motor de passo Nema 17 (1,7 A, torque de retenção 0,56 N·m, ângulo de passo 1,8°);
- Arduino UNO R3 SMD (microcontrolador ATmega328P);
- *Driver* Ponte H L298N (corrente máxima de 2 A por canal)<sup>1</sup>;
- Fonte de alimentação chaveada (Sunny Computer Technology, entrada bivolt 100-240 V / 50-60 Hz, saída 12 V / 1 A);
- Protoboard, jumpers e componentes discretos para IHM (LEDs, *buzzer*);
- Parafusos, porcas e arruelas para fixação da garra e do motor;
- Arduino IDE.

### 3.2 MODELAGEM DO SISTEMA RESPIRATÓRIO

O modelo unicompartimental RC, apresentado na Seção 2.2, constitui a representação matemática adotada para descrever a dinâmica do sistema respiratório. A seguir, descreve-se sua implementação prática: definição dos parâmetros, simulações computacionais e validação do circuito análogo que serve como planta virtual para o controlador. Também é apresentado o projeto do sistema de controle que regula os parâmetros ventilatórios (frequência respiratória, volume corrente, relação inspiração:expiração e pausa inspiratória), visando ventilação adequada e segura para diferentes condições clínicas.

#### 3.2.1 Definição dos Parâmetros

O modelo RC unicompartimental é governado pela Equação 2.1, cujas variáveis são definidas na Seção 2.2. Para a simulação, são adotados três cenários clínicos representativos,

<sup>1</sup> O L298N é um *driver* de ponte H projetado para motores DC. Embora funcione com motores de passo ao acionar duas bobinas independentes, seu desempenho é inferior ao de *drivers* dedicados como A4988 ou DRV8825, que oferecem *microstepping*, controle de corrente por *chopping* e melhor eficiência térmica. A escolha do L298N neste protótipo se justifica pela disponibilidade e pelo custo reduzido, sendo a substituição por um *driver* dedicado recomendada em versões futuras.

cujos parâmetros estão consolidados na Tabela 1. Os valores de *resistência* e *complacência* são grandezas obtidas empiricamente a partir de medições clínicas de pressão e fluxo em pacientes reais, e os intervalos adotados seguem os reportados por [Bates \(2009\)](#): para o adulto saudável,  $R = 5 \text{ cmH}_2\text{O}/(\text{L/s})$  — valor que representa a oposição típica das vias aéreas de um adulto em condições normais — e  $C = 50 \text{ mL}/\text{cmH}_2\text{O}$  — indicando que, para cada  $1 \text{ cmH}_2\text{O}$  de pressão aplicada, o pulmão saudável expande aproximadamente  $50 \text{ mL}$ ; para o cenário de DPOC,  $R = 20 \text{ cmH}_2\text{O}/(\text{L/s})$ , refletindo a *resistência* elevada das vias aéreas característica de condições obstrutivas; e para o cenário de SDRA,  $C = 20 \text{ mL}/\text{cmH}_2\text{O}$ , representando a *complacência* reduzida típica de condições restritivas. A PEEP de  $5 \text{ cmH}_2\text{O}$  é adotada como valor padrão para todos os cenários, em conformidade com as diretrizes de ventilação mecânica ([HAJJAR et al., 2024](#)).

Tabela 1 – Parâmetros do modelo RC por cenário clínico.

Cenário	R (cmH <sub>2</sub> O/(L/s))	C (mL/cmH <sub>2</sub> O)	$\tau$ (s)	P <sub>0</sub> (cmH <sub>2</sub> O)
Adulto saudável	5	50	0,25	5
DPOC (obstrutivo)	20	50	1,00	5
SDRA (restritivo)	5	20	0,10	5

A constante de tempo  $\tau = R \times C$  determina a velocidade de enchimento e esvaziamento do modelo. Cabe observar que, como a *complacência*  $C$  é expressa em mL/cmH<sub>2</sub>O na literatura clínica, é necessário convertê-la para L/cmH<sub>2</sub>O antes de multiplicar pela *resistência*  $R$  (em cmH<sub>2</sub>O/(L/s)), de modo que  $\tau$  resulte em segundos. Por exemplo, no cenário de DPOC:  $\tau = 20 \times 0,050 = 1,00 \text{ s}$  (onde  $50 \text{ mL}/\text{cmH}_2\text{O} = 0,050 \text{ L}/\text{cmH}_2\text{O}$ ). No cenário saudável,  $\tau = 5 \times 0,050 = 0,25 \text{ s}$  indica resposta rápida; no cenário de DPOC,  $\tau = 1,00 \text{ s}$  reflete o esvaziamento lento característico de condições obstrutivas; no cenário de SDRA,  $\tau = 5 \times 0,020 = 0,10 \text{ s}$  evidencia a rigidez pulmonar típica de condições restritivas, onde o pulmão atinge rapidamente seu volume máximo reduzido ([BATES, 2009](#)).

### 3.2.2 Analogia Elétrica

A implementação computacional do modelo baseia-se na analogia entre grandezas respiratórias e elétricas, muito utilizada na literatura de mecânica pulmonar ([BATES, 2009](#); [ALMEIDA, 2007](#)), conforme a Tabela 2.

Tabela 2 – Analogia entre grandezas respiratórias e elétricas.

Grandeza respiratória	Grandeza elétrica	Unidade elétrica
Pressão nas vias aéreas $P(t)$	Tensão $V(t)$	volt (V)
Fluxo de ar $\dot{V}(t)$	Corrente $I(t)$	ampere (A)
Resistência das vias aéreas $R$	Resistência $R$	ohm ( $\Omega$ )
Complacência pulmonar $C$	Capacitância $C$	farad (F)
PEEP ( $P_0$ )	Fonte DC de offset	volt (V)

Dessa forma, o circuito elétrico equivalente consiste em um resistor  $R$  em série com um capacitor  $C$ , alimentados por uma fonte de tensão que representa a pressão aplicada pela compressão do BVM. A fonte é configurada como onda quadrada com amplitude de 0 a 20 V e frequência de 0,25 Hz (período de 4 s, correspondente a 15 irpm), simulando o ciclo de inspiração (tensão alta, bolsa comprimida) e expiração (tensão baixa, bolsa relaxada). Uma fonte DC de 5 V em série com o capacitor representa a PEEP, garantindo que a tensão no capacitor (análoga à pressão alveolar) não caia abaixo de 5 V ao final da expiração.

### 3.2.3 Simulação no Falstad

A primeira etapa de simulação é realizada no Falstad Circuit Simulator, um simulador de circuitos *online* gratuito acessível em <https://www.falstad.com/circuit/>. O objetivo dessa etapa é validar qualitativamente o comportamento do circuito RC análogo e verificar se as formas de onda de tensão no capacitor (pressão alveolar) e corrente no resistor (fluxo de ar) correspondem ao esperado pela teoria.

O circuito montado consiste em:

- Fonte de tensão pulsada (onda quadrada): amplitude de 0 a 20 V, frequência de 0,25 Hz;
- Resistor em série:  $R = 5 \Omega$  (cenário saudável);
- Capacitor em série:  $C = 50 \text{ mF}$  (0,05 F). Na analogia adotada, a escala de conversão é  $1 \text{ mL/cmH}_2\text{O} \leftrightarrow 1 \text{ mF}$ , de modo que os valores numéricos de capacitância no circuito coincidem diretamente com os valores de *complacência* pulmonar. Embora capacitores de dezenas de milifarads sejam incomuns em circuitos eletrônicos convencionais, no simulador Falstad esses valores são configurados sem restrição, e a escolha dessa escala preserva a correspondência numérica direta com os parâmetros clínicos da Tabela 1;
- Fonte DC de 5 V em série com o capacitor (representando a PEEP).

A tensão no capacitor é monitorada via *scope*, representando a evolução temporal da pressão alveolar  $P_A(t)$ , enquanto a corrente no resistor representa o fluxo de ar  $\dot{V}(t)$ .

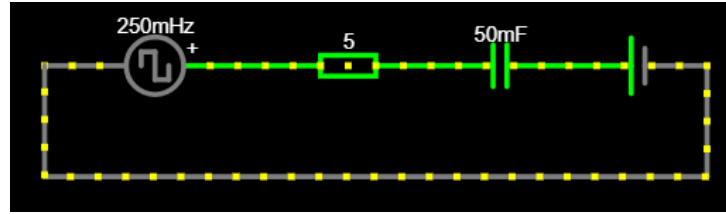


Figura 2 – Circuito RC montado no Falstad.

Fonte: elaboração própria (Falstad).

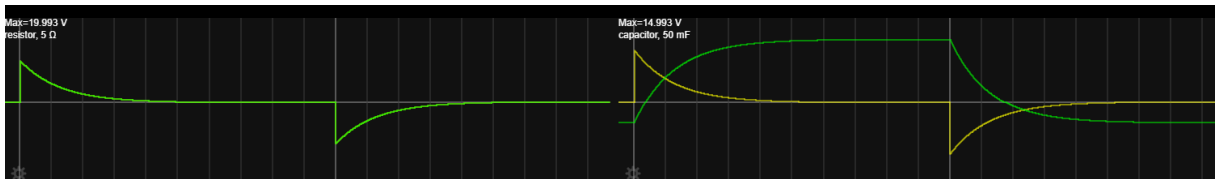


Figura 3 – Formas de onda no Falstad - cenário saudável.

Fonte: elaboração própria (Falstad).

Os resultados obtidos no Falstad permitem observar o comportamento exponencial de carga e descarga do capacitor, coerente com a constante de tempo  $\tau = 0,25$  s do cenário saudável. A tensão no capacitor apresenta elevação gradual durante a fase de inspiração (fonte em 20 V) e decaimento durante a expiração (fonte em 0 V), sem cair abaixo de 5 V devido à fonte DC de offset. A corrente no resistor apresenta picos no início de cada fase, decaindo exponencialmente à medida que o capacitor se aproxima do equilíbrio. Esse comportamento é análogo ao fluxo de ar, que é máximo no início da inspiração e diminui conforme o pulmão se enche.

Em seguida, os parâmetros são alterados para simular os cenários patológicos:

- DPOC:  $R = 20 \Omega$ ,  $C = 50 \text{ mF} \rightarrow \tau = 1,00$  s. Observa-se esvaziamento significativamente mais lento do capacitor, com a tensão não retornando completamente ao valor de offset dentro do período de expiração, coerente com o fenômeno de aprisionamento aéreo analisado quantitativamente na Subseção 3.2.4.
- SDRA:  $R = 5 \Omega$ ,  $C = 20 \text{ mF} \rightarrow \tau = 0,10$  s. Observa-se carga e descarga muito rápidas, com a tensão no capacitor atingindo valores mais elevados para o mesmo estímulo, refletindo a maior pressão necessária para ventilar um pulmão com *complacência* reduzida.

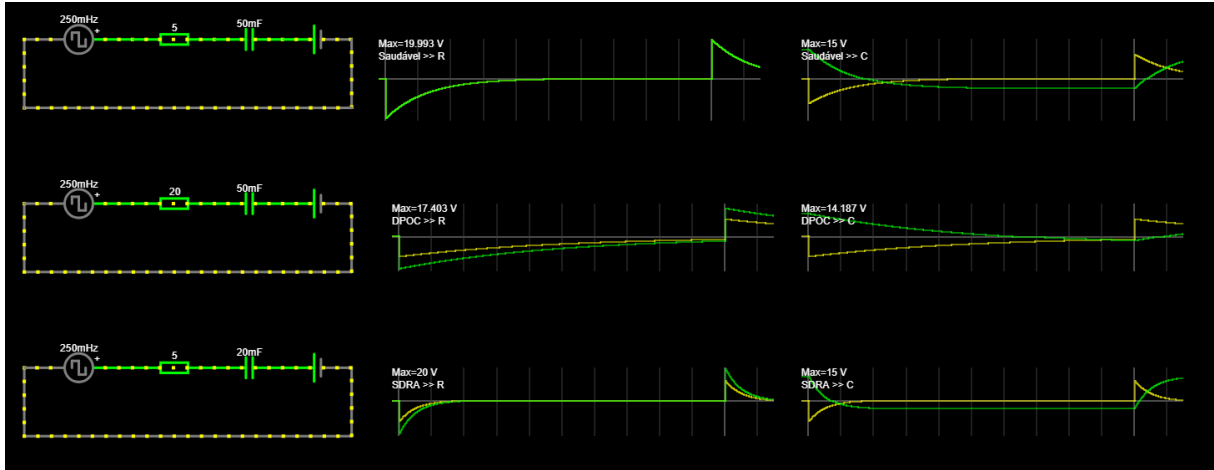


Figura 4 – Comparativo das formas de onda no Falstad para os três cenários clínicos.

Fonte: elaboração própria (Falstad).

Essa etapa confirma que o circuito RC série reproduz de forma satisfatória os comportamentos qualitativos esperados para cada condição clínica, validando a analogia elétrica como base para as simulações subsequentes.

### 3.2.4 Simulação no MATLAB

A segunda etapa de simulação é conduzida no MATLAB. O objetivo principal desta etapa é obter resultados quantitativos do modelo RC, gerar gráficos para análise detalhada e preparar a estrutura para a inserção do controlador PI.

O modelo é implementado em diagrama de blocos conforme a seguinte estrutura:

1. **Fonte de pressão** —  $R(s)$ : Pulse Generator, onda quadrada de 0 a 20, período de 4 s (15 irpm);
2. **Somador** —  $E(s)$ : calcula a diferença entre pressão aplicada e pressão alveolar realimentada;
3. **Resistência** —  $1/R$ : Gain com valor  $1/R$ , cuja saída é o fluxo  $\dot{V}(t)$ ;
4. **Complacência** —  $1/s$ : Integrator, cuja saída é o volume  $V(t)$ ;
5. **Pressão alveolar** —  $1/C$ : Gain com valor  $1/C$  (*elastância*), saída  $P_A(t) = V(t)/C$ ;
6. **PEEP** —  $D(s)$ : Constant inicialmente em 0 (modelo simplificado), alterado para 5 cmH<sub>2</sub>O após validação do modelo. Essa abordagem em duas etapas permite isolar problemas de sintonia antes de adicionar o offset de pressão;

7. **Realimentação** —  $H(s)$ : saída  $Y(s) = P_A(t) + P_0$  conectada à entrada negativa do somador, fechando a malha unitária.

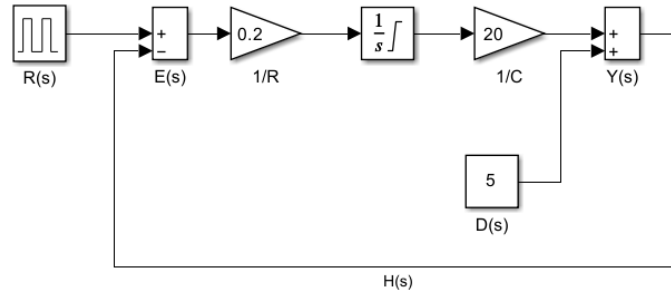


Figura 5 – Diagrama de blocos do modelo RC em *malha aberta* no Simulink.

Fonte: elaboração própria (MATLAB).

Blocos *Scope* são adicionados para monitorar simultaneamente a pressão aplicada, a pressão alveolar, o fluxo de ar e o volume pulmonar. A simulação é configurada com tempo final de 20 s e *solver* ode45 (passo variável).

#### (a) Resultados — Cenário Saudável.

Para o cenário de adulto saudável ( $R = 5 \text{ cmH}_2\text{O}/(\text{L}/\text{s})$ ,  $C = 50 \text{ mL}/\text{cmH}_2\text{O}$ ,  $P_0 = 5 \text{ cmH}_2\text{O}$ ), os resultados da simulação são apresentados na Figura 6.

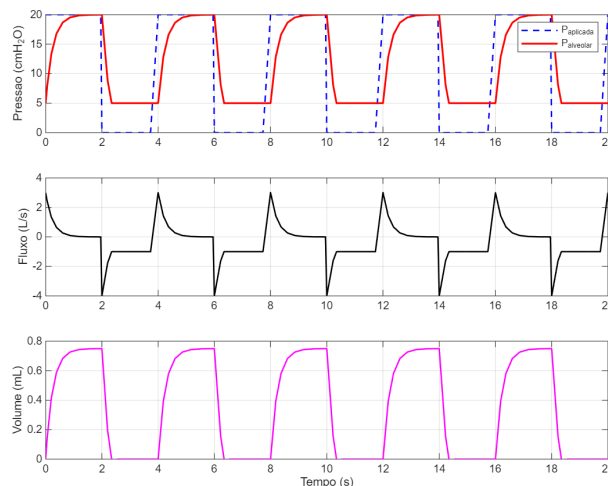


Figura 6 – Simulação em *malha aberta* — cenário saudável.

Fonte: elaboração própria (MATLAB).

A pressão alveolar acompanha a pressão aplicada com atraso determinado pela constante de tempo  $\tau = 0,25$  s, atingindo aproximadamente 95% do valor final em  $3\tau = 0,75$  s. Em menos de um segundo, o pulmão já está praticamente cheio. Esse comportamento é visível no gráfico de pressão: enquanto a pressão aplicada sobe instantaneamente de 0 para 20 cmH<sub>2</sub>O (onda quadrada), a pressão alveolar sobe de forma exponencial suave, como um capacitor carregando. Na expiração, quando a pressão aplicada cai a zero, a pressão alveolar decai exponencialmente até retornar ao valor de PEEP, completando o esvaziamento antes do próximo ciclo.

O fluxo de ar apresenta picos simétricos no início de cada fase (aproximadamente +3 L/s na inspiração e -3 L/s na expiração), decaindo exponencialmente à medida que a diferença de pressão entre a fonte e o pulmão diminui. Esse perfil é análogo à corrente em um circuito RC: máxima no instante da comutação e tendendo a zero conforme o capacitor se aproxima do equilíbrio.

O volume pulmonar oscila entre zero e aproximadamente 0,75 L (750 mL), retornando completamente a zero a cada ciclo — sem acúmulo de ar residual. Esse valor é coerente com a relação  $V_t = C \times \Delta P = 50 \times 15 = 750$  mL (onde  $\Delta P = 20 - 5 = 15$  cmH<sub>2</sub>O é a diferença entre a pressão de pico e a PEEP). Trata-se de um volume corrente adequado para ventilação de adultos, dentro da faixa recomendada pelas diretrizes clínicas ([HAJJAR et al., 2024](#)).

### (b) Resultados — Cenário DPOC.

Para o cenário de DPOC ( $R = 20$  cmH<sub>2</sub>O/(L/s),  $C = 50$  mL/cmH<sub>2</sub>O), a constante de tempo aumenta para  $\tau = 1,00$  s, de modo que o esvaziamento de 95% exige aproximadamente 3 s ( $3\tau$ ), valor próximo ao período de expiração disponível no ciclo de 4 s. A Figura 7 apresenta os resultados dessa simulação.

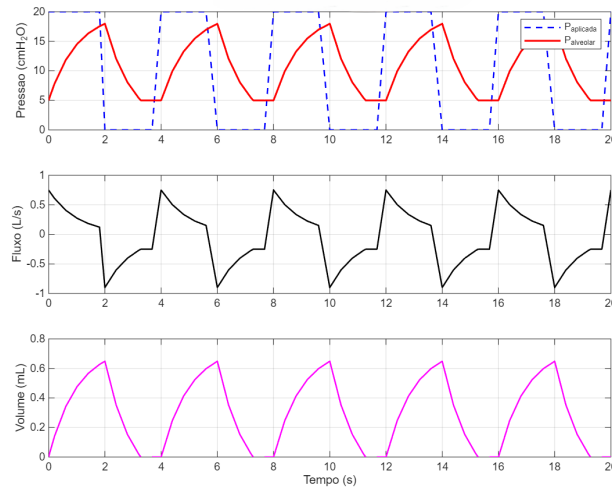


Figura 7 – Simulação em *malha aberta* — cenário DPOC.

Fonte: elaboração própria (MATLAB).

Observa-se que a pressão alveolar não retorna completamente ao valor de PEEP antes do início do próximo ciclo inspiratório, evidenciando aprisionamento aéreo. O resultado, a rigor, era previsível: no cenário saudável, a pressão alveolar atinge quase 20 cmH<sub>2</sub>O e retorna à PEEP; no cenário DPOC, a pressão de pico alcança apenas cerca de 15 a 17 cmH<sub>2</sub>O e, ao final da expiração, permanece acima da PEEP (em torno de 5 a 7 cmH<sub>2</sub>O residuais). A *resistência* elevada ( $R = 20$ ) retarda o esvaziamento, e o tempo disponível para expiração é inferior a  $3\tau$ , conforme a restrição formalizada na Equação 3.3.

O fluxo de ar confirma essa análise: os picos inspiratórios são significativamente menores (cerca de 0,75 L/s, contra 3 L/s no cenário saudável), pois a alta *resistência* limita a vazão de ar. Mais importante, o fluxo expiratório não chega a zero antes do início da próxima inspiração — o pulmão ainda está esvaziando quando o próximo ciclo começa.

O volume pulmonar oscila entre aproximadamente 0,1 a 0,2 L (residual) e 0,6 L, sem retornar a zero entre ciclos, confirmando o aprisionamento aéreo. Na prática clínica, esse fenômeno é característico de pacientes com DPOC e indica que o controlador deve utilizar frequências respiratórias mais baixas ou relações I:E mais favoráveis à expiração (como 1:3 ou 1:4).

### (c) Resultados — Cenário SDRA.

Para o cenário de SDRA ( $R = 5$  cmH<sub>2</sub>O/(L/s),  $C = 20$  mL/cmH<sub>2</sub>O), a constante de tempo reduz para  $\tau = 0,10$  s, resultando em resposta extremamente rápida. A Figura 8 apresenta os resultados dessa simulação.

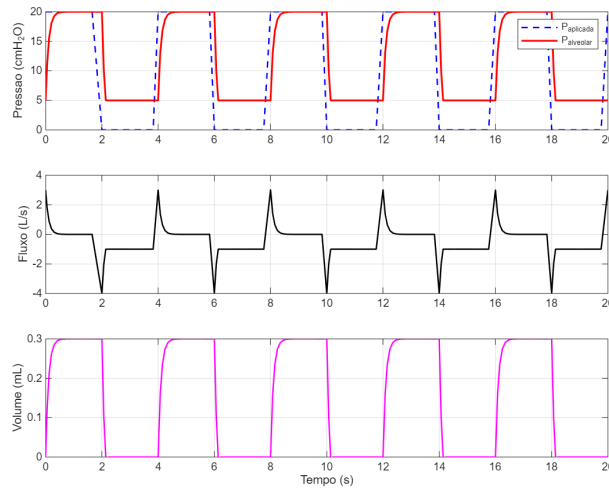


Figura 8 – Simulação em *malha aberta* — cenário SDRA.

Fonte: elaboração própria (MATLAB).

A pressão alveolar acompanha quase instantaneamente a pressão aplicada ( $3\tau = 0,30$  s), de modo que as curvas de pressão aplicada e pressão alveolar ficam praticamente sobrepostas. Com *complacência* reduzida, o pulmão oferece pouca elasticidade e a pressão interna sobe rapidamente assim que o ar começa a entrar.

O fluxo de ar apresenta picos altos e muito breves (semelhantes a impulsos), pois a diferença de pressão entre a fonte e o pulmão se anula quase instantaneamente. Os picos inspiratórios e expiratórios são da mesma ordem de grandeza do cenário saudável (cerca de  $\pm 3$  L/s), mas duram muito menos tempo, já que o equilíbrio é atingido rapidamente.

O volume corrente entregue é de aproximadamente 0,30 L (300 mL), significativamente menor que os 750 mL do cenário saudável, apesar de a mesma pressão de 20 cmH<sub>2</sub>O ser aplicada. Pela relação já apresentada,  $V_t = C \times \Delta P = 20 \times 15 = 300$  mL: com a *complacência* reduzida pela metade, o pulmão rígido aceita menos ar para a mesma pressão. Não há aprisionamento aéreo, pois o esvaziamento é extremamente rápido ( $3\tau = 0,30$  s  $\ll$   $T_{\text{exp}} = 2,67$  s). Embora 300 mL seja inferior ao volume corrente convencional de 6-8 mL/kg, esse valor é compatível com a estratégia de ventilação protetora recomendada para SDRA, que preconiza volumes de 4-6 mL/kg (aproximadamente 280-420 mL para um adulto de 70 kg) para minimizar a lesão pulmonar induzida por ventilação (HAJJAR *et al.*, 2024). O resultado, portanto, não indica ventilação inadequada, mas sim a necessidade de o controlador monitorar o volume entregue e, caso o operador configure um alvo superior, aumentar a pressão inspiratória de forma controlada, respeitando o limite de 40 cmH<sub>2</sub>O.

Realizando uma comparação entre os três cenários, a Tabela 3 resume os principais indicadores obtidos nas simulações.

Tabela 3 – Indicadores das simulações em *malha aberta* por cenário.

Indicador	Saudável	DPOC	SDRA
$\tau$ (s)	0,25	1,00	0,10
Tempo para 95% ( $3\tau$ ) (s)	0,75	3,00	0,30
Aprisionamento aéreo	Não	Sim	Não
Pressão de pico (cmH <sub>2</sub> O)	≈ 20	≈ 15-17	≈ 20
Volume corrente (L)	≈ 0,75	≈ 0,60	≈ 0,30
Pico de fluxo inspiratório (L/s)	≈ 3,0	≈ 0,75	≈ 3,0

A Figura 9 apresenta a sobreposição das curvas de pressão alveolar obtidas nos três cenários, permitindo visualizar diretamente o efeito da variação dos parâmetros  $R$  e  $C$  sobre a dinâmica do modelo.

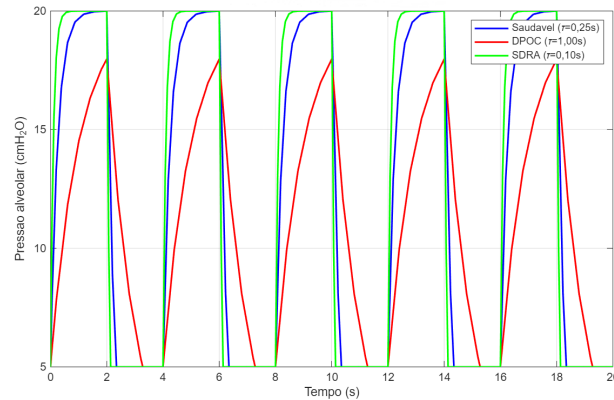


Figura 9 – Comparativo da pressão alveolar para os três cenários clínicos.

Fonte: elaboração própria (MATLAB).

O modelo unicompartimental RC adotado neste trabalho apresenta limitações inerentes à sua simplicidade: não representa a heterogeneidade regional da ventilação, não incorpora comportamentos não lineares da *elastância* em grandes volumes e não modela a interação entre compartimentos pulmonares. Contudo, esse modelo captura adequadamente a dinâmica respiratória para fins de projeto de controladores em muitas situações clínicas.

Cabe destacar que a *BVM* não é modelada nas simulações como um componente com dinâmica própria. A *complacência* da bolsa, a *resistência* das válvulas e as perdas no circuito respiratório entre a fonte de ar e o paciente não são representadas explicitamente. O circuito RC modela exclusivamente o sistema respiratório do paciente (vias aéreas e pulmão), e o papel da *BVM* na simulação difere conforme a configuração da malha:

- **Malha aberta:** o bloco Pulse Generator atua como fonte de pressão fixa e cíclica (0 a 20 cmH<sub>2</sub>O), representando diretamente a pressão que a compressão do BVM aplicaria ao paciente. Nessa configuração, o objetivo é apenas observar como o pulmão

responde a um estímulo de pressão constante nos três cenários clínicos, validando o modelo RC (não há controlador atuando nessa configuração).

- **Malha fechada (com PID):** o Pulse Generator passa a gerar a referência de pressão ( $P_{\text{ref}}$ ), ou seja, o perfil ventilatório desejado pelo operador (calculado a partir de  $V_t$ ,  $C$  e  $P_0$ ). Quem efetivamente aplica a pressão ao modelo do pulmão é a saída do controlador PI, limitada pelo bloco de Saturação (0 a 40 cmH<sub>2</sub>O). Nessa configuração, a saída do PI representa a ação do conjunto motor garra BVM: o controlador ajusta dinamicamente a pressão aplicada para que a pressão alveolar do paciente siga a referência, compensando as particularidades de cada condição clínica ( $\tau$  elevado na DPOC,  $C$  reduzido na SDRA).

Dessa forma, o sistema de controle simulado reproduz a lógica do protótipo físico: o operador define os parâmetros ventilatórios ideais (frequência respiratória, volume corrente, relação I:E, pausa inspiratória), o controlador PI calcula o esforço de compressão necessário e o atuador (combinação do motor com a garra e o BVM) entrega a pressão correspondente ao paciente. Na prática, a *BVM* apresenta comportamento não linear, pois a relação entre o deslocamento da garra e o volume de ar entregue varia ao longo do curso de compressão: no início, a geometria da bolsa ainda não está suficientemente deformada, de modo que a garra encontra pouca *resistência* e desloca pouco ar; no meio do curso, a deformação da bolsa atinge a faixa em que a relação força/volume é mais favorável, entregando maior volume por unidade de deslocamento; e no final da compressão, a bolsa se torna significativamente mais rígida, exigindo força muito maior para deslocar pouco volume adicional. A simplificação de tratar a *BVM* como fonte de pressão ideal é adequada ao escopo deste trabalho, cujo foco é a validação do controlador sobre a planta pulmonar.

A abordagem de simulação em duas ferramentas complementares (Falstad para validação qualitativa da analogia elétrica e Simulink para análise quantitativa e projeto do controlador) permite uma transição gradual entre a modelagem teórica e a implementação prática, reduzindo riscos e facilitando a identificação de problemas em cada etapa.

### 3.3 PROJETO DO SISTEMA DE CONTROLE

Nesta seção apresenta-se o projeto do sistema de controle que regula a ventilação fornecida ao paciente. Dois mecanismos fundamentais são controlados: o número de compressões por unidade de tempo, que define a frequência respiratória efetiva, e a amplitude de compressão da bolsa, que determina o volume corrente entregue a cada ciclo. Combinados com a temporização das fases inspiratória e expiratória, esses dois graus de liberdade constituem as variáveis manipuladas pelo sistema.

### 3.3.1 Definição dos Parâmetros Ventilatórios de Controle

O controlador opera sobre quatro parâmetros ventilatórios fundamentais, cujos valores podem ser definidos pelo operador. Cada parâmetro possui uma faixa de operação delimitada por limites de segurança fisiológicos.

#### 3.3.1.1 Relação I:E

A relação I:E (inspiração:expiração) define a proporção entre o tempo dedicado à fase inspiratória e o tempo dedicado à fase expiratória dentro de cada ciclo respiratório. Para uma relação I:E de  $1 : n$  e um período de ciclo  $T_{\text{ciclo}}$ , os tempos são calculados como:

$$T_{\text{ins}} = \frac{T_{\text{ciclo}}}{1 + n} \quad \text{e} \quad T_{\text{exp}} = \frac{n \cdot T_{\text{ciclo}}}{1 + n} \quad (3.1)$$

onde  $T_{\text{ins}}$  é o tempo inspiratório (incluindo eventual pausa inspiratória) e  $T_{\text{exp}}$  é o tempo expiratório.

A faixa de operação é de 1:1 a 1:4, com valor padrão de 1:2. A escolha da relação I:E depende da condição clínica do paciente:

- Relação 1:2 (padrão): adequada para a maioria dos pacientes adultos em ventilação controlada;
- Relação 1:3 ou 1:4: indicada para pacientes com padrão obstrutivo (DPOC, asma), conforme a restrição de  $T_{\text{exp}} \geq 3\tau$  (Equação 3.3);
- Relação 1:1 ou 1:1,5: pode ser considerada em pacientes com padrão restritivo (SDRA, fibrose), nos quais o esvaziamento é rápido.

No diagrama de blocos do MATLAB, a relação I:E é implementada configurando o *duty cycle* da fonte de onda quadrada. Para uma relação 1:2, o *duty cycle* é de 33% (1/3 do período em nível alto, 2/3 em nível baixo); para 1:3, é de 25%; para 1:1, é de 50%.

A Tabela 4 apresenta os tempos resultantes para diferentes combinações de frequência respiratória e relação I:E.

Tabela 4 – Tempos inspiratório e expiratório por configuração de  $f_R$  e I:E.

$f_R$ (irpm)	$T_{\text{ciclo}}$ (s)	I:E	$T_{\text{ins}}$ (s)	$T_{\text{exp}}$ (s)
12	5,00	1:2	1,67	3,33
15	4,00	1:2	1,33	2,67
15	4,00	1:3	1,00	3,00
20	3,00	1:2	1,00	2,00
20	3,00	1:1	1,50	1,50

### 3.3.1.2 Frequência Respiratória ( $f_R$ )

A frequência respiratória define o número de ciclos inspiração e expiração por minuto (irpm). O período total de cada ciclo é dado por:

$$T_{\text{ciclo}} = \frac{60}{f_R} \quad (\text{s}) \quad (3.2)$$

A faixa de operação adotada é de 8 a 30 irpm, com valor padrão de 15 irpm, conforme recomendado para ventilação de adultos com via aérea avançada (HAJJAR *et al.*, 2024). Essa faixa é consistente com os requisitos da norma ISO 10651-4 (INTERNATIONAL ORGANIZATION FOR STANDARDIZATION, 2023).

No modelo RC implementado no MATLAB, a frequência respiratória é definida pela frequência da fonte de pressão (onda quadrada) no diagrama de blocos. Ao alterar  $f_R$ , o período total do ciclo muda conforme a Equação 3.2, e os tempos inspiratório e expiratório são redistribuídos de acordo com a relação I:E configurada (Equação 3.1).

O ponto central é que, durante a expiração, o modelo RC precisa de tempo suficiente para descarregar. Conforme discutido na Subseção 3.2.4, são necessários aproximadamente  $3\tau$  para esvaziamento de 95%. Portanto, o tempo expiratório deve satisfazer:

$$T_{\text{exp}} \geq 3\tau \quad (3.3)$$

Quando essa condição não é atendida, ocorre aprisionamento aéreo, conforme observado nas simulações do cenário DPOC (Subseção 3.2.4).

Essa restrição é particularmente crítica no cenário de DPOC, onde  $\tau = 1,00$  s e, portanto, o esvaziamento adequado exige  $T_{\text{exp}} \geq 3,00$  s. Considerando  $f_R = 15$  irpm com relação I:E de 1:2:

- Período do ciclo:  $T_{\text{ciclo}} = 60/15 = 4,00$  s;
- Tempo inspiratório:  $T_{\text{ins}} = 4,00/(1 + 2) = 1,33$  s;
- Tempo expiratório:  $T_{\text{exp}} = 4,00 - 1,33 = 2,67$  s.

Como  $T_{\text{exp}} = 2,67$  s  $<$   $3\tau = 3,00$  s, o tempo disponível para expiração é insuficiente. Nessa configuração, o pulmão retém ar a cada ciclo, elevando progressivamente a pressão residual, comportamento observado nas simulações do cenário DPOC (Subseção 3.2.4). Para evitar esse problema, o controlador deve utilizar frequências mais baixas ou relações I:E mais favoráveis à expiração (como 1:3 ou 1:4) em pacientes com padrão obstrutivo, conforme detalhado na Tabela 4.

### 3.3.1.3 Volume Corrente ( $V_t$ )

O volume corrente define o volume de ar entregue a cada compressão da bolsa. No protótipo, o volume é controlado indiretamente pelo curso de compressão da garra: quanto maior o deslocamento angular do motor, maior a deformação da bolsa e, conseqüentemente, maior o volume de ar deslocado. A relação entre o ângulo de compressão ( $\theta$ ) e o volume entregue é estabelecida com base nas dimensões geométricas da bolsa *BVM* Foyomed (capacidade total de aproximadamente 1500 mL) e no curso angular da garra, assumindo proporcionalidade linear entre o deslocamento do motor e o volume deslocado. Essa relação é parametrizada no firmware pela constante `VT_MAX`, que define o volume máximo correspondente ao curso completo do motor (`PASSOS_MAX`).

A faixa de operação é de 200 a 800 mL, com valor padrão de 500 mL (compatível com os limites estabelecidos pela norma ISO 10651-4 [International Organization for Standardization \(2023\)](#) de 6-7 mL/kg para um adulto de  $\approx 75$  kg). No modelo RC simulado no MATLAB, o volume corrente entregue ao final de cada inspiração depende da pressão de pico aplicada e da *complacência* do modelo, conforme a relação  $V_t = C \times (P_{\text{pico}} - P_0)$  já discutida na Subseção 3.2.4. Para o cenário saudável ( $C = 50$  mL/cmH<sub>2</sub>O), uma pressão de pico de 15 cmH<sub>2</sub>O acima da PEEP resulta em  $V_t = 750$  mL. Para o cenário de SDRA ( $C = 20$  mL/cmH<sub>2</sub>O), a mesma pressão entrega apenas 300 mL, valor compatível com a estratégia de ventilação protetora discutida na Subseção 3.2.4. Caso o operador configure um alvo de volume corrente superior, o controlador deve aumentar a pressão inspiratória de forma controlada, respeitando o limite de 40 cmH<sub>2</sub>O.

O controlador deve garantir que o volume entregue esteja dentro da faixa segura. Se o volume estimado for inferior ao mínimo (200 mL) por três ciclos consecutivos, o sistema aumenta progressivamente o curso de compressão. Se o volume estimado exceder o máximo (800 mL), o curso é reduzido imediatamente no ciclo seguinte.

### 3.3.1.4 Pausa Inspiratória ( $T_{\text{pausa}}$ )

A pausa inspiratória é um intervalo de tempo, ao final da fase inspiratória, durante o qual a garra mantém a bolsa comprimida na posição de máximo deslocamento antes de iniciar a fase expiratória. Durante essa pausa, o ar permanece nos pulmões sob pressão, permitindo uma distribuição mais homogênea do volume entregue entre as diferentes regiões pulmonares, especialmente relevante em pulmões com heterogeneidade ventilatória, onde regiões com constantes de tempo mais altas necessitam de mais tempo para se equilibrar ([LUCANGELO; BERNABÉ; BLANCH, 2005](#)).

A faixa de operação é de 0 a 1,0 s, com valor padrão de 0 s (sem pausa). A inclusão de pausa inspiratória reduz o tempo disponível para a expiração dentro do mesmo período de ciclo, de modo que o controlador verifica automaticamente se o tempo expiratório

resultante é suficiente (mínimo de  $3\tau$  para o cenário configurado) e rejeita configurações que resultem em tempo expiratório insuficiente.

### 3.3.2 Estratégia de Controle de Frequência Respiratória

O controle da frequência respiratória é realizado pela temporização do ciclo de compressão do motor. O firmware do Arduino implementa uma máquina de estados com três fases sequenciais:

1. **Inspiração ativa:** o motor avança, comprimindo a bolsa durante o tempo  $T_{\text{ins}} - T_{\text{pausa}}$ . A velocidade de avanço é calculada para que o curso de compressão correspondente ao volume alvo seja completado dentro desse intervalo. Caso uma pausa inspiratória esteja configurada ( $T_{\text{pausa}} > 0$ ), o motor mantém a posição de máxima compressão durante esse intervalo antes de prosseguir para a expiração;
2. **Expiração:** o motor retorna à posição de repouso, permitindo que a bolsa se expanda e o paciente expire passivamente durante  $T_{\text{exp}}$ ;
3. **Espera:** caso o retorno do motor seja mais rápido que  $T_{\text{exp}}$ , o sistema aguarda o tempo restante antes de iniciar o próximo ciclo.

O período total do ciclo é controlado por um temporizador baseado em `millis()`, garantindo que a frequência respiratória efetiva corresponda ao valor configurado. A cada início de ciclo, o firmware recalcula os tempos de cada fase com base nos parâmetros atuais ( $f_R$ , relação I:E,  $T_{\text{pausa}}$ ), permitindo ajustes em tempo real sem interrupção da ventilação.

No modelo MATLAB, essa estratégia é representada pela frequência e pelo *duty cycle* da fonte de onda quadrada. A validação consiste em verificar que, para cada combinação de  $f_R$  e relação I:E, o modelo RC atinge equilíbrio adequado em ambas as fases (inspiração e expiração), conforme os critérios de  $3\tau$  discutidos na subseção [Definição dos Parâmetros \(3.2.1\)](#).

### 3.3.3 Estratégia de Controle de Volume

O controle do volume corrente é o aspecto mais crítico do sistema, pois determina diretamente a eficácia da ventilação. A estratégia adotada opera em dois níveis complementares: controle em *malha aberta* (baseado na calibração da bolsa) e controle em *malha fechada* (baseado na realimentação de pressão).

#### 3.3.3.1 Malha aberta — Estimativa de volume pela geometria da bolsa

A relação entre o ângulo de compressão da garra ( $\theta$ ) e o volume de ar deslocado pela bolsa ( $V_t$ ) é estabelecida com base nas dimensões da bolsa *BVM* e no curso mecânico da

garra. Considerando que a bolsa para adulto possui capacidade total de aproximadamente 1500 mL e que a compressão completa pelo curso máximo do motor ( $\text{PASSOS\_MAX} = 120$  passos) corresponde à deformação máxima da bolsa, adota-se uma relação linear entre a posição do motor e o volume deslocado:  $V_t = (\theta/\theta_{\text{max}}) \times V_{t,\text{max}}$ , onde  $V_{t,\text{max}} = 800$  mL é o volume máximo configurado no firmware. Dado um volume alvo  $V_{t,\text{ref}}$ , o firmware calcula a posição correspondente do motor e comanda o avanço até essa posição.

Essa abordagem fornece o ponto de operação para o controle do atuador, sendo adequada ao escopo didático do protótipo. Variações decorrentes da geometria não linear da bolsa, do envelhecimento do material e das tolerâncias mecânicas da garra podem ser compensadas pela *malha fechada*, conforme descrito a seguir.

### 3.3.3.2 Malha fechada — Controlador PI de pressão

O controlador PI, cuja lei de controle é apresentada na Equação 2.2, atua sobre a posição do motor de passo para regular a pressão alveolar em torno da referência. O motor é acionado em modo *full-step* via ponte H L298N, e a saída do controlador determina o curso de compressão da garra. Essa abordagem é consistente com a estratégia de controle em *malha fechada* proposta por Borrello (2021), que demonstrou que controladores relativamente simples podem obter desempenho satisfatório em ventiladores *BVM* automatizados, desde que haja realimentação adequada de pressão. A pressão de referência ( $P_{\text{ref}}$ ) é calculada a partir do volume corrente desejado e da *complacência* estimada do modelo:

$$P_{\text{ref}} = \frac{V_{t,\text{ref}}}{C} + P_0 \quad (3.4)$$

O erro de pressão  $e(t) = P_{\text{ref}} - P_{\text{med}}(t)$  alimenta o controlador PI, cuja saída  $u(t)$  é a pressão aplicada em  $\text{cmH}_2\text{O}$ , saturada entre 0 e 40  $\text{cmH}_2\text{O}$ . No firmware embarcado, essa pressão é convertida em posição-alvo do motor de passo pela seguinte relação:

$$\theta_{\text{alvo}} = \frac{(u(t) - P_0) \cdot C}{V_{t,\text{max}}} \cdot \theta_{\text{max}} \quad (3.5)$$

onde  $\theta_{\text{alvo}}$  é a posição-alvo em passos,  $u(t) - P_0$  é a pressão relativa acima da PEEP,  $C$  é a *complacência* configurada,  $V_{t,\text{max}} = 800$  mL é o volume máximo parametrizado e  $\theta_{\text{max}} = 120$  passos é o curso máximo do motor. Essa conversão inverte a relação estática do modelo RC: a partir da pressão que o controlador deseja aplicar, calcula-se o volume correspondente ( $V = (u - P_0) \times C$ ) e, em seguida, a posição proporcional do motor. Quando  $e(t) > 0$  (pressão medida abaixo da referência), o controlador avança o motor; quando  $e(t) < 0$  (pressão acima da referência), o motor é recuado ou mantido em repouso.

No diagrama de blocos do MATLAB, a inserção do controlador PI segue a estrutura clássica de controle em *malha fechada*. Adotando a nomenclatura padrão da teoria de

controle:  $r(t)$  é a referência (*setpoint*),  $y(t)$  é a saída (variável de processo),  $e(t) = r(t) - y(t)$  é o erro,  $C(s)$  é o controlador e  $u(t)$  é o sinal de controle (variável manipulada). No contexto deste trabalho,  $r(t) = P_{\text{ref}}$ ,  $y(t) = P_A(t)$  e  $u(t) = P_{\text{aplicada}}(t)$ .

A estrutura requer dois somadores com funções distintas, conforme ilustrado na Figura 10: o comparador ( $\Sigma_e$ ), que calcula o erro  $e(t) = r(t) - y(t)$  e alimenta o controlador PI; e o somador da planta ( $\Sigma_p$ ), que calcula a diferença de pressão física  $u(t) - y(t)$  conforme a equação do modelo RC. Ambos subtraem  $y(t)$ , mas de sinais diferentes ( $r(t)$  versus  $u(t)$ ) e em pontos diferentes do diagrama: o comparador faz parte da malha de realimentação do controlador, enquanto o somador da planta faz parte da equação diferencial do modelo. A estrutura resultante é:

1. **Referência**  $r(t)$ : Pulse Generator ( $\Delta P = V_{t,\text{ref}}/C$ , período 4 s, Pulse Width 33%) somado a Constant ( $P_0 = 5 \text{ cmH}_2\text{O}$ ). A referência oscila entre  $P_0$  na expiração e  $V_{t,\text{ref}}/C + P_0$  na inspiração;
2. **Comparador** —  $e(t)$ : calcula  $e(t) = r(t) - y(t)$ ;
3. **Controlador PI** —  $G_c(s)$ : bloco PID Controller com  $K_p$ ,  $K_i$  (e  $K_d = 0$ );
4. **Saturação**: limita  $u(t)$  entre 0 e 40  $\text{cmH}_2\text{O}$  (elemento não linear);
5. **Somador da planta** —  $\Sigma_p$ : calcula  $u(t) - y(t)$ , preservando a dinâmica RC;
6. **Planta**  $G_p(s)$  —  $1/R$ ,  $1/s$ ,  $1/C$ : conforme Subseção 3.2.4, com perturbação  $d(t) = \text{PEEP}$ ;
7. **Realimentação** —  $H(s)$ :  $y(t) = P_A(t)$  realimentada ao comparador e ao somador da planta.

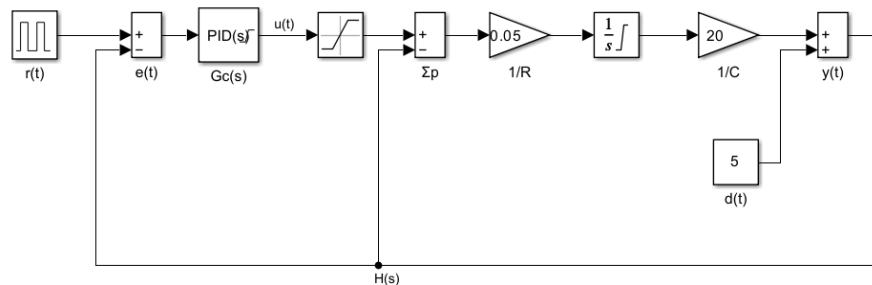


Figura 10 – Diagrama de blocos do sistema com controlador PI no Simulink.

Fonte: elaboração própria (MATLAB).

### 3.3.4 Sintonia do Controlador PI

A sintonia dos ganhos do controlador é realizada seguindo os pontos: análise da planta no domínio contínuo ( $s$ ), projeto do controlador por alocação de polos, e discretização para implementação no microcontrolador. Adicionalmente, os ganhos obtidos por alocação de polos são comparados com os ganhos fornecidos pelo método IMC (Internal Model Control), permitindo avaliar as diferenças entre as duas abordagens e justificar a escolha final.

#### 3.3.4.1 Função de Transferência da Planta

A função de transferência da planta RC em *malha aberta*, obtida a partir da Equação 2.1, é:

$$G(s) = \frac{P_A(s)}{P_{aplicada}(s)} = \frac{1}{RCs + 1} = \frac{1}{\tau s + 1} \quad (3.6)$$

que corresponde a um sistema de primeira ordem com ganho estático unitário ( $K = 1$ ) e constante de tempo  $\tau$ . Para o cenário saudável ( $\tau = 0,25$  s), a planta apresenta polo em  $s = -1/\tau = -4$  rad/s; para DPOC ( $\tau = 1,00$  s), polo em  $s = -1$  rad/s; para SDRA ( $\tau = 0,10$  s), polo em  $s = -10$  rad/s.

Como a planta é de primeira ordem e não apresenta *overshoot* natural (ação derivativa não é necessária), um controlador PI é suficiente. Essa escolha simplifica a implementação no microcontrolador, eliminando a necessidade de filtragem do termo derivativo, e reduz a sensibilidade a ruído no sinal de pressão. O controlador PI, no domínio  $s$ , tem a forma:

$$C(s) = K_p + \frac{K_i}{s} = \frac{K_p s + K_i}{s} \quad (3.7)$$

que introduz um polo na origem (garantindo erro nulo em regime permanente para entrada degrau) e um zero em  $s = -K_i/K_p$ .

#### 3.3.4.2 Projeto por Alocação de Polos com Cancelamento de Polo e Zero

A técnica adotada como método principal de projeto consiste em posicionar o zero do controlador PI para cancelar o polo da planta, e em seguida escolher o ganho que define a velocidade de resposta desejada em *malha fechada*.

(a) Cancelamento de polo e zero:

O polo da planta está em  $s = -1/\tau$ . Posicionando o zero do PI nessa mesma localização, impõe-se:

$$\frac{K_i}{K_p} = \frac{1}{\tau} \implies K_i = \frac{K_p}{\tau} \quad (3.8)$$

Com esse cancelamento, a função de transferência em *malha aberta*  $L(s) = C(s) \cdot G(s)$  simplifica-se para:

$$L(s) = \frac{K_p s + K_i}{s} \cdot \frac{1}{\tau s + 1} = \frac{K_p}{\tau} \cdot \frac{1}{s} = \frac{K_p}{\tau s} \quad (3.9)$$

resultando em um integrador puro com ganho  $K_p/\tau$ . A função de transferência em *malha fechada* é, portanto:

$$T(s) = \frac{L(s)}{1 + L(s)} = \frac{K_p/\tau}{s + K_p/\tau} = \frac{\alpha}{s + \alpha} \quad (3.10)$$

onde  $\alpha = K_p/\tau$  é o polo desejado em *malha fechada*. Trata-se de um sistema de primeira ordem com constante de tempo  $\tau_{MF} = 1/\alpha = \tau/K_p$ , cuja resposta é uma exponencial pura sem *overshoot*.

(b) Escolha do polo desejado:

O polo  $\alpha$  determina a velocidade de resposta em *malha fechada*. Para que o sistema atinja 95% do valor final em um tempo  $t_{95}$ , a relação é  $t_{95} = 3/\alpha$ . Segundo Ogata (2010), em projetos de controle é recomendável que o tempo de acomodação em *malha fechada* seja da ordem de um sexto a um décimo do período do sinal de referência mais rápido, de modo que a resposta do controlador se estabilize ainda no início da fase de interesse. Considerando que o tempo inspiratório mais curto previsto é de aproximadamente 1 s (Tabela 4), a aplicação dessa diretriz resulta em  $t_{95}$  entre  $1/10 = 0,10$  s e  $1/6 \approx 0,17$  s. Adota-se  $t_{95} = 0,15$  s como especificação de projeto, valor intermediário dentro dessa faixa:

$$\alpha = \frac{3}{t_{95}} = \frac{3}{0,15} = 20 \text{ rad/s} \quad (3.11)$$

(c) Cálculo dos ganhos para o cenário saudável:

Com  $\alpha = 20$  rad/s e  $\tau = 0,25$  s:

$$K_p = \alpha \cdot \tau = 20 \times 0,25 = 5 \quad (3.12)$$

$$K_i = \frac{K_p}{\tau} = \frac{5}{0,25} = 20 \quad (3.13)$$

O controlador PI projetado por alocação de polos é, portanto:

$$C(s) = 5 + \frac{20}{s} = \frac{5s + 20}{s} \quad (3.14)$$

Cabe observar que, como os ganhos são calculados para o cenário saudável ( $\tau = 0,25$  s), o cancelamento de polo e zero é exato apenas nesse cenário. Nos cenários DPOC ( $\tau = 1,00$  s) e SDRA ( $\tau = 0,10$  s), o zero do controlador ( $s = -K_i/K_p = -4$  rad/s) não coincide com o polo da planta ( $s = -1$  e  $s = -10$  rad/s, respectivamente), de modo que o cancelamento é apenas parcial. Essa é uma limitação inerente à adoção de ganhos fixos para múltiplos cenários, e seus efeitos são analisados na Subseção 3.3.4.7.

(d) Verificação de estabilidade:

A equação característica em *malha fechada*, com cancelamento exato, é  $s + \alpha = 0$ , cujo polo está em  $s = -20$  (semiplano esquerdo), garantindo estabilidade. Sem cancelamento exato (cenários DPOC e SDRA), a equação característica de segunda ordem resultante é:

$$\tau s^2 + (1 + K_p)s + K_i = 0 \quad (3.15)$$

Para o cenário DPOC ( $\tau = 1,00$  s):  $s^2 + 6s + 20 = 0$ , com polos em  $s = -3 \pm j\sqrt{11} \approx -3 \pm 3,32j$ , polos complexos conjugados com parte real negativa, portanto estável, com fator de amortecimento  $\zeta = 3/\sqrt{20} \approx 0,67$ . Para o cenário SDRA ( $\tau = 0,10$  s):  $0,1s^2 + 6s + 20 = 0$ , equivalente a  $s^2 + 60s + 200 = 0$ , com polos em  $s \approx -3,5$  e  $s \approx -56,5$ , dois polos reais negativos, portanto estável e sem *overshoot*. Em ambos os casos, todos os coeficientes da equação característica são positivos, satisfazendo o critério de Routh-Hurwitz para estabilidade.

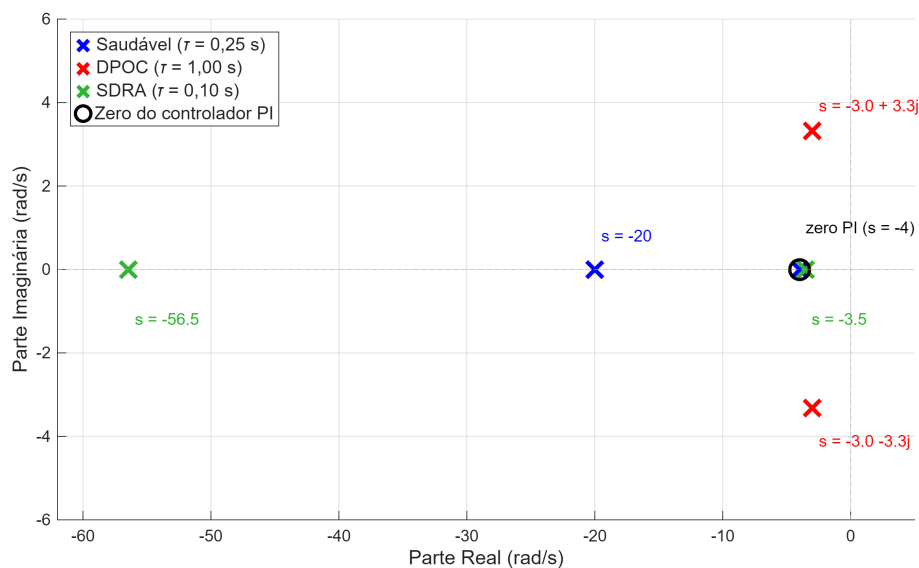


Figura 11 – Polos em *malha fechada* para os três cenários clínicos e zero do controlador PI.

Fonte: elaboração própria (MATLAB).

### 3.3.4.3 Projeto Alternativo pelo Método IMC

Para fins de comparação e verificação independente, os ganhos são também calculados pelo método IMC (Internal Model Control), técnica consolidada na indústria de controle de processos por fornecer fórmulas diretas de sintonia a partir dos parâmetros da planta, sem necessidade de ensaios iterativos. A escolha do IMC como método comparativo se justifica por três razões: é amplamente utilizado como referência em projetos de controladores para plantas de primeira ordem, permite verificar a consistência dos ganhos obtidos por alocação de polos de forma independente, e suas fórmulas explícitas facilitam a reprodutibilidade do projeto. Para uma planta de primeira ordem sem atraso,  $G(s) = K/(\tau s + 1)$ , o método IMC fornece os ganhos de um controlador PI como:

$$K_p^{\text{IMC}} = \frac{\tau}{K \cdot \lambda} \quad (3.16)$$

$$K_i^{\text{IMC}} = \frac{K_p^{\text{IMC}}}{\tau} = \frac{1}{K \cdot \lambda} \quad (3.17)$$

onde  $\lambda$  é a constante de tempo desejada em *malha fechada*. Valores menores de  $\lambda$  resultam em resposta mais rápida, porém com maior esforço de controle e menor robustez a incertezas no modelo.

Para verificar a consistência dos ganhos obtidos por alocação de polos, calcula-se o valor de  $\lambda$  correspondente ao  $K_p$  já projetado. No cenário saudável ( $K = 1$ ,  $\tau = 0,25$  s), a relação  $\lambda = \tau/K_p = 0,25/5 = 0,05$  s indica que a resposta em *malha fechada* é cinco vezes mais rápida que a planta em *malha aberta*. Substituindo nas fórmulas IMC:

- $K_p^{\text{IMC}} = 0,25/(1 \times 0,05) = 5$
- $K_i^{\text{IMC}} = 5/0,25 = 20$

Os ganhos IMC coincidem exatamente com os ganhos obtidos por alocação de polos ( $K_p = 5$ ,  $K_i = 20$ ), confirmando a consistência do projeto. Essa equivalência é esperada: para plantas de primeira ordem sem atraso, o método IMC com parâmetro  $\lambda$  é matematicamente equivalente à alocação de polos com polo desejado  $\alpha = 1/\lambda$ . A diferença reside na abordagem conceitual: o IMC parte de um parâmetro de sintonia ( $\lambda$ ) e fornece os ganhos diretamente por fórmulas, enquanto a alocação de polos parte da análise da função de transferência em *malha fechada* e da especificação explícita dos polos desejados. A alocação de polos é preferida neste trabalho por ser mais transparente quanto à dinâmica resultante e por seguir o fluxo convencional de projeto de controle: análise da planta em  $s$ , especificação de desempenho, obtenção formal do controlador e discretização.

### 3.3.4.4 Discretização do Controlador — Transformação Bilinear (Tustin)

O controlador PI projetado no domínio contínuo (Equação 3.14) deve ser discretizado para implementação no Arduino UNO. A escolha do período de amostragem segue a diretriz de Ogata (2010) para controle digital, segundo a qual o período de amostragem deve ser entre  $1/6$  e  $1/10$  do tempo de acomodação desejado em *malha fechada*. Com  $t_{95} = 0,15$  s (Equação 3.11), essa faixa resulta em  $T$  entre  $t_{95}/10 = 0,015$  s e  $t_{95}/6 = 0,025$  s. Adota-se  $T = 0,02$  s (20 ms), valor intermediário dentro dessa faixa e compatível com a capacidade de processamento do microcontrolador ATmega328P do Arduino UNO, que opera a 16 MHz e executa o *loop* de controle a uma taxa de 50 Hz. A discretização é realizada pela transformação bilinear (Tustin), que substitui a variável de Laplace  $s$  pela aproximação:

$$s = \frac{2}{T} \cdot \frac{z - 1}{z + 1} \quad (3.18)$$

A transformação bilinear é preferida em relação à aproximação de Euler (forward ou backward) por preservar a correspondência entre estabilidade no domínio  $s$  e no domínio  $z$ : polos estáveis no semiplano esquerdo de  $s$  são mapeados para o interior do círculo unitário em  $z$ , sem distorção de estabilidade. Além disso, a transformação de Tustin apresenta melhor preservação da resposta em frequência para períodos de amostragem relativamente grandes em relação à dinâmica do sistema.

Aplicando a Equação 3.18 ao controlador  $C(s) = K_p + K_i/s$ :

$$C(z) = K_p + K_i \cdot \frac{T}{2} \cdot \frac{z + 1}{z - 1} \quad (3.19)$$

Reorganizando em termos de potências de  $z^{-1}$  e convertendo para equação de diferenças, obtém-se a lei de controle recursiva implementável no microcontrolador:

$$u[k] = u[k - 1] + \left( K_p + \frac{K_i T}{2} \right) e[k] + \left( -K_p + \frac{K_i T}{2} \right) e[k - 1] \quad (3.20)$$

onde  $u[k]$  é o sinal de controle no instante  $k$ ,  $e[k] = r[k] - y[k]$  é o erro discreto e  $T = 0,02$  s. Substituindo os ganhos projetados ( $K_p = 5$ ,  $K_i = 20$ ):

- Coeficiente de  $e[k]$ :  $a_0 = K_p + K_i T/2 = 5 + 20 \times 0,01 = 5,2$
- Coeficiente de  $e[k - 1]$ :  $a_1 = -K_p + K_i T/2 = -5 + 0,2 = -4,8$

A equação de diferenças final é:

$$u[k] = u[k - 1] + 5,2 \cdot e[k] - 4,8 \cdot e[k - 1] \quad (3.21)$$

Essa forma recursiva é computacionalmente eficiente: requer apenas duas multiplicações, uma soma e o armazenamento de dois valores anteriores ( $u[k-1]$  e  $e[k-1]$ ), sendo plenamente compatível com as restrições de memória e processamento do Arduino.

Para confirmar que a discretização preserva a estabilidade, os polos do sistema em *malha fechada* no domínio  $z$  são verificados. O polo contínuo  $s = -\alpha = -20$  rad/s é mapeado pela transformação bilinear para:

$$z = \frac{1 + sT/2}{1 - sT/2} = \frac{1 + (-20)(0,01)}{1 - (-20)(0,01)} = \frac{0,8}{1,2} = 0,667 \quad (3.22)$$

Como  $|z| = 0,667 < 1$ , o polo está no interior do círculo unitário, confirmando estabilidade no domínio discreto.

A Figura 12 apresenta a comparação entre a resposta ao degrau do sistema em *malha fechada* contínua (solução analítica) e a resposta obtida com o controlador discretizado pela transformação de Tustin ( $T = 0,02$  s), para o cenário saudável. A sobreposição das curvas confirma que a discretização preserva a dinâmica do controlador contínuo, validando a equação de diferenças (Equação 3.20) como implementação fiel do projeto realizado no domínio  $s$ . Observa-se que a resposta discreta apresenta valores ligeiramente superiores à contínua nos primeiros instantes de amostragem. Esse comportamento é esperado e decorre da discretização da planta pelo método de Euler, que introduz um pequeno avanço de fase em relação à solução analítica exata. A diferença inferior a 5% do valor final se anula em poucos períodos de amostragem, confirmando que o período  $T = 20$  ms é adequado para a dinâmica da planta ( $T \ll \tau = 0,25$  s).

Cabe ressaltar que, na implementação embarcada no Arduino UNO, o período de amostragem não é garantido por interrupção de *timer*, sendo controlado por `millis()` com compensação no final do *loop*. As chamadas a `delayMicroseconds()` dentro da rotina de acionamento do motor introduzem *jitter* no período efetivo, de modo que  $T$  pode variar entre iterações. Essa variação não compromete a estabilidade do controlador para a dinâmica lenta da planta ( $\tau \geq 0,10$  s), mas uma implementação baseada em interrupções por *timer* seria mais adequada em versões futuras.

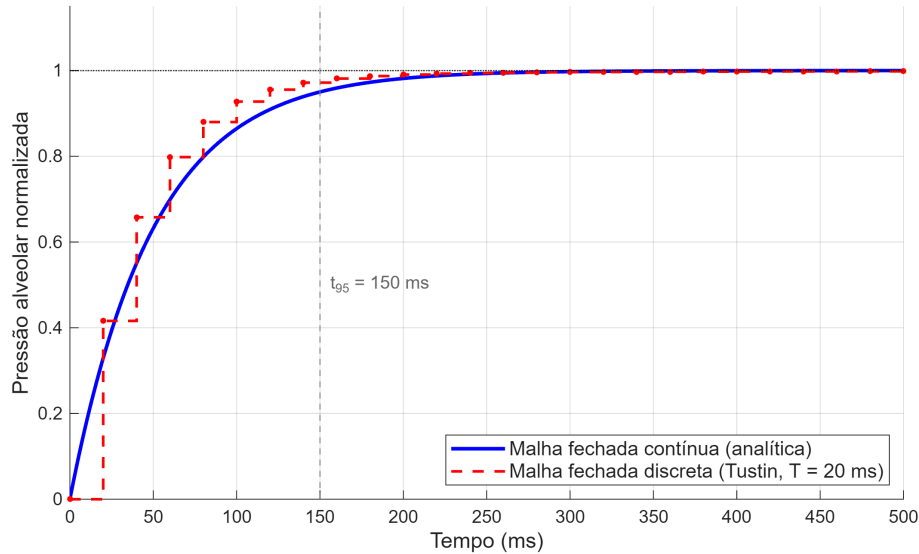


Figura 12 – Resposta ao degrau em *malha fechada*: controlador contínuo vs. discretizado - cenário saudável.

Fonte: elaboração própria (MATLAB).

#### 3.3.4.5 Anti-Windup

Em sistemas com saturação no sinal de controle (como o limite de 40 cmH<sub>2</sub>O imposto pelo bloco de Saturação), o integrador do controlador pode acumular erro durante os períodos em que a saída está saturada. Quando a saturação cessa, o integrador carrega um valor excessivo que provoca *overshoot* e oscilações transitórias.

Para mitigar esse efeito, é implementado o mecanismo de anti-windup por *clamping*: a atualização recursiva de  $u[k]$  (Equação 3.20) é interrompida sempre que a saída do controlador atinge os limites de saturação e o erro tem o mesmo sinal da saída saturada, impedindo o acúmulo desnecessário. No Simulink, essa configuração é feita com *Anti-windup method: clamping*. No firmware embarcado, a mesma lógica é implementada condicionando a atualização do termo integral.

#### 3.3.4.6 Procedimento de Sintonia e Validação

O procedimento de sintonia segue as seguintes etapas:

1. Obter a função de transferência da planta (Equação 3.6) e identificar o polo em  $s = -1/\tau$ ;
2. Especificar o desempenho desejado em *malha fechada*:  $t_{95} = 0,15$  s, resultando em polo desejado  $\alpha = 20$  rad/s (Equação 3.11);
3. Projetar o controlador PI por alocação de polos com cancelamento de polo e zero (Equações 3.12 e 3.13):  $K_p = 5$ ,  $K_i = 20$ ;

4. Verificar a estabilidade em *malha fechada* para os três cenários clínicos (Equação 3.15);
5. Discretizar o controlador pela transformação bilinear de Tustin (Equação 3.20) com  $T = 0,02$  s;
6. Verificar a estabilidade no domínio  $z$ ;
7. Executar a simulação em *malha fechada* no MATLAB, inicialmente sem PEEP ( $P_0 = 0$ ), e analisar o gráfico de  $r(t)$  versus  $y(t)$ , avaliando *overshoot*, tempo de acomodação e erro em regime permanente;
8. Reintroduzir a PEEP ( $P_0 = 5$  cmH<sub>2</sub>O) e verificar se o desempenho se mantém aceitável, já que a dinâmica RC não se altera e apenas o ponto de operação é deslocado;
9. Validar os ganhos nos três cenários clínicos (saudável, DPOC e SDRA).

A Tabela 5 apresenta os ganhos adotados e os indicadores de desempenho para cada cenário clínico. Os ganhos são obtidos pelo método de alocação de polos com cancelamento de polo e zero, projetados para o cenário saudável ( $\tau = 0,25$  s) e aplicados de forma fixa aos três cenários. Conforme demonstrado na Subseção 3.3.4.3, os ganhos IMC com  $\lambda = 0,05$  s são numericamente idênticos, confirmando a consistência entre os dois métodos para planta. O anti-windup por *clamping* é ativado em todos os casos.

Tabela 5 – Ganhos PI e indicadores de desempenho por cenário clínico.

Cenário	$\tau$ (s)	$K_p$	$K_i$	<i>Overshoot</i> (%)	$t_s$ (s)
Saudável	0,25	5	20	0	$\approx 0,66$
DPOC	1,00	5	20	$\approx 9,4$	$\approx 1,08$
SDRA	0,10	5	20	0	$\approx 1,00$

A adoção de ganhos fixos ( $K_p = 5$ ,  $K_i = 20$ ) para os três cenários representa uma escolha de projeto deliberada: em um sistema embarcado de baixo custo destinado a uso emergencial, a simplicidade e a previsibilidade do controlador são muito importantes. A equivalência entre alocação de polos e IMC reforça a robustez dessa escolha — dois métodos independentes convergem para os mesmos ganhos, indicando que os valores são consistentes com a dinâmica da planta.

Cabe observar que os tempos de acomodação obtidos nas simulações (0,66 s a 1,08 s) são superiores ao valor teórico de  $3/\alpha = 0,15$  s previsto pela análise no domínio  $s$ . Essa diferença decorre de dois fatores: a análise teórica considera um degrau unitário normalizado sem restrições, enquanto a simulação parte da PEEP (5 cmH<sub>2</sub>O) com saturação do sinal de controle em 40 cmH<sub>2</sub>O; e a presença da saturação limita o esforço de controle nos instantes iniciais, prolongando o transitório. Esse comportamento é esperado e não compromete a

operação do dispositivo nos cenários saudável e SDRA, cujos tempos de acomodação (0,66 s e 1,0 s, respectivamente) são inferiores ou iguais ao menor tempo inspiratório previsto de 1 s (Tabela 4), sendo ainda mais favoráveis nas configurações com tempos inspiratórios superiores (1,33 s a 1,67 s). No cenário DPOC, a constante de tempo elevada da planta ( $\tau = 1,00$  s) faz com que a pressão alveolar não atinja completamente a referência dentro do tempo inspiratório disponível, comportamento coerente com a fisiopatologia obstrutiva e contornável pelo ajuste de  $f_R$  e I:E via IHM, conforme discutido na Subseção 3.3.4.7.

No cenário DPOC, o *overshoot* de 9,4% é coerente com o fator de amortecimento  $\zeta \approx 0,67$  calculado na Subseção 3.3.4.2, resultante da ausência de cancelamento de polo e zero exato nesse cenário. Embora superior ao ideal, esse valor não representa risco ao paciente, pois a pressão de pico resultante (15,9 cmH<sub>2</sub>O) permanece dentro dos limites de segurança.

Para os cenários saudável e SDRA, o controlador apresenta *overshoot* nulo e erro em regime permanente desprezível, confirmando o cancelamento de polo e zero efetivo. O anti-windup por *clamping* é essencial para manter o desempenho consistente entre ciclos, evitando o acúmulo do integrador durante as transições entre inspiração e expiração.

### 3.3.4.7 Análise do Cenário DPOC

No cenário DPOC ( $R = 20$  cmH<sub>2</sub>O/(L/s),  $\tau = 1,00$  s), o comportamento do sistema em *malha fechada* com parâmetros ventilatórios padrão ( $f_R = 15$  irpm, I:E = 1:2) reflete fielmente a fisiopatologia obstrutiva, confirmando a coerência do modelo e do controlador. Dois fenômenos característicos são observados:

1. **Limitação de fluxo pela *resistência* elevada:** a *resistência* das vias aéreas ( $R = 20$ ) restringe o fluxo inspiratório a  $\dot{V} = (P_{\text{aplicada}} - P_A)/R$ . Mesmo com o sinal de controle no limite superior de 40 cmH<sub>2</sub>O, o fluxo máximo é de  $(40 - P_A)/20$  L/s, de modo que a pressão alveolar atinge tipicamente 13 a 14 cmH<sub>2</sub>O dentro do tempo inspiratório disponível ( $T_{\text{ins}} = 1,33$  s). Esse comportamento é esperado e reproduz a dificuldade clínica real de ventilar pacientes com padrão obstrutivo sob parâmetros convencionais.
2. **Aprisionamento aéreo:** conforme discutido na Subseção 3.3.1 (Equação 3.3), o tempo expiratório disponível com I:E de 1:2 é inferior a  $3\tau = 3,00$  s. O volume residual se acumula entre ciclos, elevando a pressão basal (BRUNNER; PIQUILLOUD *et al.*, 2024).

Esses resultados confirmam que o modelo RC e o controlador reproduzem corretamente os desafios ventilatórios do padrão obstrutivo. Na prática clínica, esses efeitos são contornados com frequências mais baixas (10-12 irpm) e relações I:E de 1:3 ou 1:4, ajustes

que o sistema proposto permite realizar em tempo real via IHM. A implementação de estratégias de controle adaptativo constitui uma evolução natural para trabalhos futuros, conforme discutido nas Conclusão.

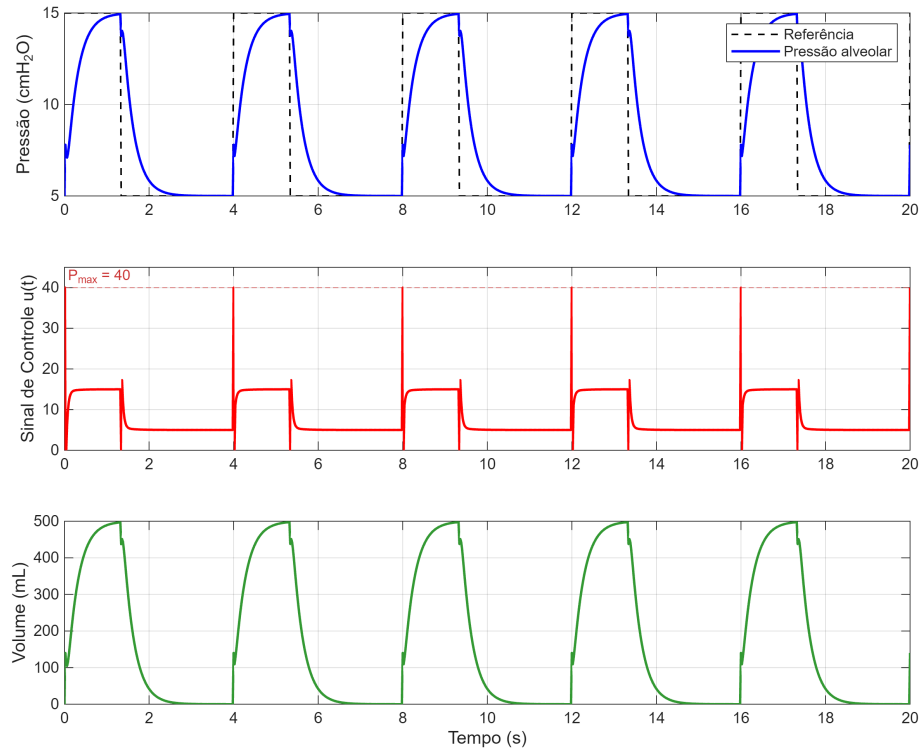


Figura 13 – Resposta em *malha fechada* para o cenário saudável: pressão, sinal de controle e volume pulmonar.

Fonte: elaboração própria (MATLAB).

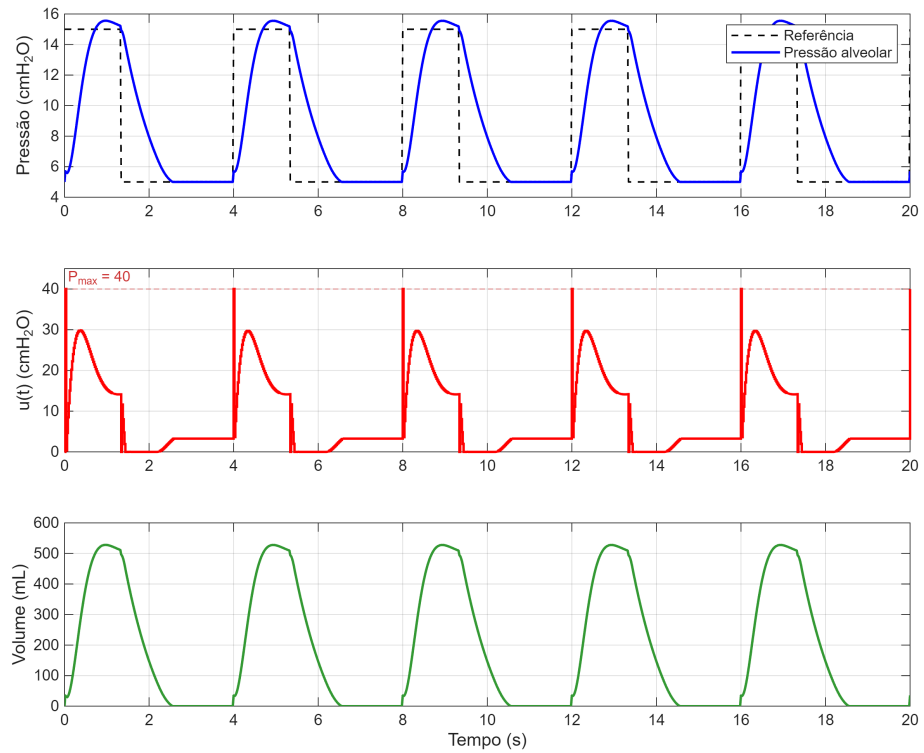


Figura 14 – Resposta em *malha fechada* para o cenário DPOC: pressão, sinal de controle e volume pulmonar.

Fonte: elaboração própria (MATLAB).

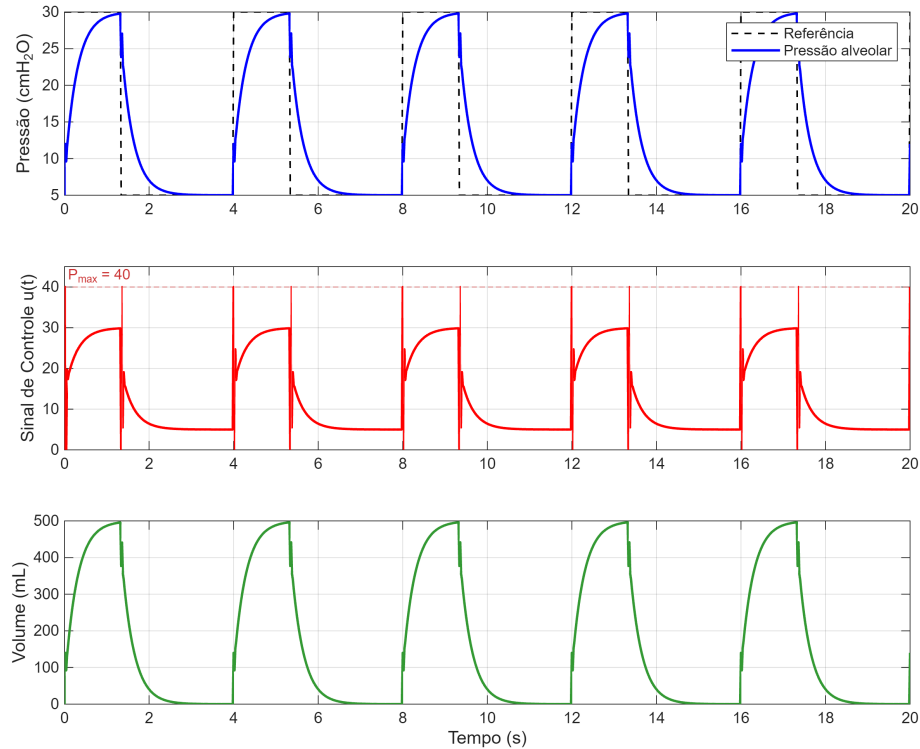


Figura 15 – Resposta em *malha fechada* para o cenário SDR: pressão, sinal de controle e volume pulmonar.

Fonte: elaboração própria (MATLAB).

### 3.3.4.8 Validação Cruzada de Parâmetros

O controlador implementa uma lógica de validação cruzada que verifica a consistência entre os parâmetros ventilatórios antes de aplicá-los. A cada alteração de parâmetro pelo operador via IHM, o firmware verifica três condições:

1. O tempo inspiratório efetivo ( $T_{\text{ins}} - T_{\text{pausa}}$ ) deve ser suficiente para o motor completar o curso de compressão correspondente ao volume alvo, considerando a velocidade máxima do motor;
2. O tempo expiratório ( $T_{\text{exp}}$ ) deve ser maior ou igual a  $3\tau$  para o cenário clínico configurado, garantindo esvaziamento adequado;
3. A pressão de pico estimada ( $V_{t,\text{ref}}/C + P_0$ ) não deve ultrapassar o limite de segurança de 40 cmH<sub>2</sub>O.

Se qualquer condição for violada, o ajuste é rejeitado e os parâmetros anteriores são mantidos, com mensagem explicativa enviada pela porta serial. Essa validação é essencial para evitar combinações perigosas. Por exemplo, frequência respiratória de 30 irpm com

relação I:E de 1:1 e pausa inspiratória de 0,5 s resultaria em  $T_{\text{exp}} = 0,5$  s, insuficiente para qualquer cenário clínico.

## 3.4 PROJETO MECÂNICO

A garra mecânica é o elo físico entre o sistema de controle e a compressão efetiva da bolsa *BVM*: ela converte o sinal de controle do motor em deformação mecânica da bolsa com amplitude e velocidade controladas. Fabricá-la por impressão 3D se justifica pela rapidez de prototipagem, pelo baixo custo de material e pela facilidade de iteração no projeto, características essenciais em um trabalho acadêmico com restrições de prazo e orçamento.

### 3.4.1 Dimensionamento e Especificações

Inicialmente, definem-se as dimensões do BVM a ser utilizado, incluindo o diâmetro da bolsa e o curso de compressão necessário para entregar o volume corrente desejado. Para uma bolsa adulto padrão (capacidade total de aproximadamente 1500 mL), o curso de compressão necessário para entregar 500-600 mL é estimado experimentalmente, servindo como referência para o dimensionamento da abertura e do curso angular da garra. Essa abordagem é consistente com a metodologia de dimensionamento descrita por [Venkatesan et al. \(2020\)](#) no projeto SVASTA e por [Espinoza et al. \(2024\)](#) no projeto COMBIOVENT, que enfatizam a importância de considerar a fadiga mecânica da bolsa no dimensionamento do mecanismo de compressão.

### 3.4.2 Modelagem no Onshape

A garra é modelada no Onshape, plataforma de modelagem 3D *online* desenvolvida pela PTC, acessível em <https://cad.onshape.com/>. A escolha por essa ferramenta se justifica pela facilidade de acesso e pela capacidade de gerar modelos exportáveis para impressão 3D sem necessidade de licenças comerciais. O mecanismo adotado é do tipo pinça biarticulada com braços côncavos e acionamento por engrenagens na base. Cada braço possui perfil côncavo que envolve a bolsa, e a articulação por engrenagens na base converte a rotação do motor em movimento de abertura e fechamento dos braços. Essa configuração permite amplificar o torque do motor na região de maior *resistência* da bolsa (início da compressão) e reduzir a velocidade angular na região de máxima compressão, favorecendo o controle fino do volume entregue.



Figura 16 – Modelo 3D da garra mecânica projetada no Onshape - Geometria.

Fonte: elaboração própria (Onshape).

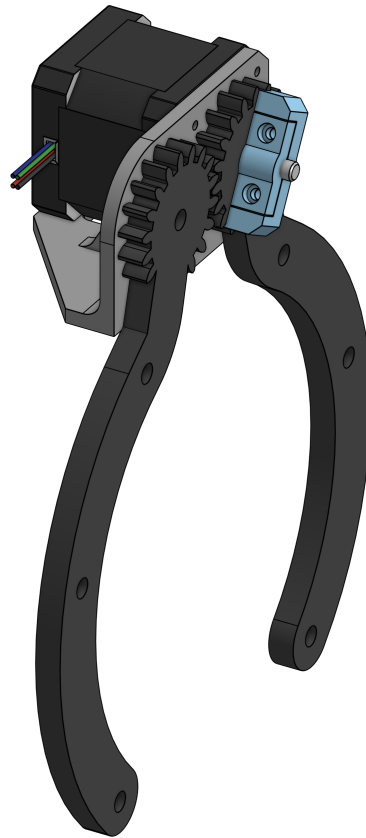


Figura 17 – Modelo 3D da garra mecânica projetada no Onshape - Conjunto.

Fonte: elaboração própria (Onshape).

### 3.4.3 Estimativa de Torque

Idealmente, a força de compressão da bolsa seria medida diretamente com um dinamômetro. Como não se dispunha desse instrumento, o torque necessário no motor é estimado com base em dados da literatura e nas características geométricas do mecanismo projetado. O objetivo dessa estimativa é verificar se o motor de passo Nema 17 selecionado (torque de retenção de 0,56 N·m) é capaz de comprimir a bolsa até o volume desejado.

A força necessária para comprimir uma bolsa *BVM* adulto padrão até o volume corrente de 500-600 mL é reportada na faixa de 20 a 40 N, dependendo do fabricante e do grau de deformação da bolsa (BORRELLO, 2021). Para a estimativa, adota-se o valor conservador de  $F = 40$  N (pior caso), garantindo margem de segurança para bolsas com maior rigidez ou envelhecimento do material.

O torque mínimo no eixo do motor é estimado pela relação:

$$T_{\text{motor}} = F \times d \quad (3.23)$$

onde  $F$  é a força de compressão da bolsa (N) e  $d$  é o braço de alavanca do mecanismo da garra (m), ou seja, a distância entre o ponto de aplicação da força e o eixo de rotação do motor.

No protótipo fabricado, o braço de alavanca medido entre o eixo de rotação do motor e o ponto de contato do braço da garra com a bolsa é de  $d \approx 46,5 \text{ mm} = 0,0465 \text{ m}$ . Substituindo na Equação 3.23:

$$T_{\text{nec}} = F \times d = 40 \times 0,0465 = 1,86 \text{ N}\cdot\text{m} \quad (3.24)$$

Como o motor Nema 17 possui torque de retenção de  $0,56 \text{ N}\cdot\text{m}$ , o torque necessário ( $T_{\text{nec}} = 1,86 \text{ N}\cdot\text{m}$ ) excede em mais de três vezes a capacidade do motor. Para que o motor fosse suficiente sem redução mecânica, o braço de alavanca deveria satisfazer  $d \leq T_{\text{motor}}/F = 0,56/40 = 0,014 \text{ m} = 14 \text{ mm}$ , valor significativamente inferior ao braço de alavanca real do mecanismo.

Essa análise confirma que o protótipo atual, embora funcional para validação do firmware e do controlador com o motor operando sem carga da bolsa, não dispõe de torque suficiente para comprimir efetivamente a *BVM*. A inclusão de um sistema de redução mecânica (engrenagens redutoras ou sistema de alavanca) ou a substituição por um motor de maior torque é necessária para viabilizar o acoplamento da bolsa, conforme discutido nos Trabalhos Futuros. Essa abordagem de estimativa por dados da literatura é consistente com a metodologia adotada por Espinoza *et al.* (2024) no projeto COMBIOVENT, que também utilizou valores de referência para o dimensionamento do atuador.

#### 3.4.4 Fabricação e Testes Mecânicos

O protótipo é impresso em 3D utilizando filamento PETG (Politereftalato de Etileno Glicol), material que oferece maior resistência mecânica e térmica em comparação ao PLA (Ácido Polilático), além de boa flexibilidade e durabilidade sob esforços cíclicos. Após a fabricação, são realizados testes mecânicos de encaixe, movimentação e resistência cíclica para verificar a integridade estrutural da garra sob operação repetitiva. As Figuras 18 e 19 apresentam o protótipo da garra impressa e montada com o motor de passo Nema 17.

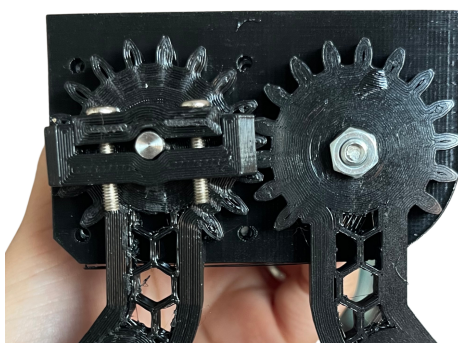


Figura 18 – Protótipo da garra mecânica impressa em 3D, montada com o motor - Encaixe das engrenagens.

Fonte: autoria própria.



Figura 19 – Protótipo da garra mecânica impressa em 3D, montada com o motor - Completo.

Fonte: autoria própria.

### 3.5 SEGURANÇA E INTERFACE HOMEM MÁQUINA

A segurança do paciente é o requisito mais crítico de qualquer dispositivo de ventilação. Diferentemente da operação manual, um sistema automatizado não possui a percepção tátil e visual do operador humano; falhas no firmware, no atuador ou nos

sensores podem resultar em barotrauma, hipoventilação ou hiperinsuflação sem que sejam imediatamente percebidas. Por essa razão, o sistema de segurança e a IHM são desenvolvidos de forma integrada ao controlador, de modo a respeitar limites fisiológicos em simulação, detectar e sinalizar condições anormais, e permitir intervenção e ajuste de parâmetros em tempo real.

### 3.5.1 Limites de Segurança

Os limites de segurança adotados neste projeto são definidos com base nas recomendações clínicas para ventilação de pacientes adultos e nos requisitos da norma ISO 10651-4 ([INTERNATIONAL ORGANIZATION FOR STANDARDIZATION, 2023](#)), que regula-menta ressuscitadores manuais. A Tabela 6 consolida os valores adotados.

Tabela 6 – Limites de segurança do dispositivo.

Parâmetro	Faixa / Limite	Justificativa
Pressão de pico máxima	Até 40 cmH <sub>2</sub> O	Prevenção de barotrauma
Volume corrente máximo	Até 800 mL	Prevenção de hiperinsuflação
Volume corrente mínimo	Mínimo de 200 mL	Garantia de ventilação alveolar mínima
Frequência respiratória mínima	8 irpm	Prevenção de hipoventilação
Frequência respiratória máxima	30 irpm	Prevenção de hiperinsuflação dinâmica
PEEP	5 cmH <sub>2</sub> O	Manutenção da estabilidade alveolar

O limite de pressão de pico em 40 cmH<sub>2</sub>O é amplamente aceito na literatura como limiar acima do qual o risco de lesão pulmonar induzida por ventilação aumenta significativamente, especialmente em pacientes com *complacência* reduzida ([HESS, 2014](#)). Esse valor é consistente com as recomendações de [Lopes \(2023\)](#) para ventiladores pulmonares no modo volume controlado e com os parâmetros de referência da norma ABNT NBR ISO 80601-2-12. Para pacientes pediátricos (abaixo de 10 kg), a norma ISO 10651-4 exige pressão limitada a 45 cmH<sub>2</sub>O ([INTERNATIONAL ORGANIZATION FOR STANDARDIZATION, 2023](#)); no presente protótipo, voltado para pacientes adultos (acima de 40 kg), adotou-se o limite mais conservador de 40 cmH<sub>2</sub>O.

A faixa de volume corrente entre 200 e 800 mL contempla desde pacientes de menor porte até situações que exijam volumes ligeiramente acima do padrão de 500-600 mL recomendado pelas diretrizes de ressuscitação ([HAJJAR \*et al.\*, 2024](#)). O limite superior de 800 mL previne hiperinsuflação e distensão gástrica, enquanto o limite inferior de 200 mL garante que a ventilação alveolar mínima seja mantida, considerando que parte do volume entregue corresponde ao espaço morto anatômico (aproximadamente 150 mL em adultos) e, portanto, não participa efetivamente da troca gasosa.

A frequência respiratória é limitada entre 8 e 30 irpm. O limite inferior de 8 irpm evita hipoventilação grave, enquanto o limite superior de 30 irpm previne hiperinsu-

flação dinâmica e aprisionamento aéreo, conforme discutido na Subseção 3.3.1 (BRUNNER; PIQUILLOUD *et al.*, 2024).

## 3.5.2 Intertravamentos e Sistema de Alarmes

O firmware do controlador implementa um conjunto de intertravamentos que monitoram continuamente as variáveis do sistema e atuam de forma autônoma quando condições anormais são detectadas. Esses mecanismos operam independentemente da IHM, garantindo proteção mesmo em caso de falha na interface de comunicação.

### 3.5.2.1 Parada de Emergência

O sistema dispõe de uma parada de emergência acionada via comando serial: ao digitar “1” no Monitor Serial da Arduino IDE, o firmware interrompe imediatamente o ciclo de compressão, desativa todas as bobinas do motor (*full-step* com todos os pinos em nível baixo) e aciona o LED vermelho e o *buzzer* intermitente. O sistema permanece em estado de emergência até que o operador digite “reset”, que retorna o sistema ao estado parado, permitindo reiniciar a ventilação com o comando “start”. Essa verificação é realizada por *polling* no início de cada iteração do *loop* principal, com prioridade sobre todas as demais rotinas (nenhuma outra lógica é executada enquanto o sistema estiver em emergência).

O firmware implementa também um alarme informativo de sobrepressão, acionado quando a pressão estimada ultrapassa o limite de 40 cmH<sub>2</sub>O. Esse alarme sinaliza a condição por meio do LED vermelho e do *buzzer*, sem interromper o motor. A decisão de não bloquear o motor pode parecer contraintuitiva, mas se justifica: como a pressão é estimada pelo modelo RC embarcado e não medida por sensor físico, o bloqueio do motor por sobrepressão simulada poderia causar comportamento indesejado. A parada efetiva do motor é reservada exclusivamente ao comando de emergência.

### 3.5.2.2 Sistema de Alarmes

O sistema de alarmes é projetado para sinalizar condições que exigem atenção do operador, classificadas em dois níveis de prioridade conforme a gravidade da condição. A Tabela 7 apresenta os alarmes implementados.

O alarme de sobrepressão, já descrito, é desativado automaticamente quando a pressão estimada cai abaixo de  $P_{\max} - 2$  cmH<sub>2</sub>O (histerese de 2 cmH<sub>2</sub>O).

Os alarmes de média prioridade indicam condições que comprometem a eficácia da ventilação se persistirem. O alarme de volume insuficiente é acionado quando o volume corrente estimado fica abaixo de 200 mL por três ciclos consecutivos. O alarme de volume excessivo atua quando o volume estimado excede 800 mL.

Tabela 7 – Alarmes implementados no firmware.

Prioridade	Condição	Critério	Ação automática
Alta (crítico)	Sobrepresão	$P_{est} > 40$ cmH <sub>2</sub> O	LED vermelho, <i>buzzer</i> , sinalização serial
Média (atenção)	Volume insuficiente	$V_t < 200$ mL por 3 ciclos	LED amarelo, <i>buzzer</i> breve
	Volume excessivo	$V_t > 800$ mL	LED amarelo, sina- lização serial

### 3.5.2.3 Detecção de Desconexão e Vazamento

Na etapa atual do desenvolvimento, o firmware não implementa detecção direta de desconexão ou vazamento no circuito respiratório, uma vez que o sistema opera com pressão estimada pelo modelo RC embarcado e não dispõe de sensor de pressão ou fluxo no circuito do paciente. A inclusão de um sensor de pressão diferencial e de um sensor de fluxo no circuito respiratório é prevista como evolução do projeto (Seção de Trabalhos Futuros), permitindo a detecção de desconexões e fugas por análise do perfil de pressão durante a compressão.

### 3.5.2.4 Registro de Eventos (Log de Alarmes)

Os alarmes gerados são transmitidos imediatamente pela porta serial no momento da detecção, no formato:

```
[ALARME] SOBREPRESSAO: 42.3
```

```
[ALARME] VOL_INSUF: 180
```

```
[ALARME] VOL_EXCESS: 850
```

Essa saída permite ao operador identificar a ocorrência e o valor da variável que disparou o evento. O registro pode ser capturado pelo Monitor Serial da Arduino IDE para análise posterior.

## 3.5.3 Interface Homem Máquina (IHM)

A IHM do protótipo é projetada considerando as restrições de hardware do Arduino UNO e o contexto de uso em ambulância, onde a interface deve ser simples, legível e permitir ajustes rápidos. A arquitetura da IHM combina dois canais complementares: uma interface serial via Arduino IDE para monitoramento detalhado e registro de dados, e uma interface física com componentes discretos para indicação de estado e interação básica.

### 3.5.3.1 Interface Serial (Monitor Serial / Serial Plotter)

O canal principal de monitoramento é a porta serial do Arduino (9600 baud), que transmite dados em tempo real para o computador conectado. O firmware opera em dois modos de saída, selecionáveis pelo operador:

- **Modo monitor** (padrão): a cada 2 segundos, transmite uma linha com o estado atual do ciclo, a pressão estimada, o volume corrente e a posição do motor, em formato legível com *labels*. Esse modo é adequado para acompanhamento textual e depuração via Monitor Serial;
- **Modo *plotter***: ativado pelo comando `plotter`, transmite a cada 100 ms três valores numéricos separados por tabulação (pressão estimada, posição real do motor e posição-alvo), compatíveis com o Serial Plotter da Arduino IDE para visualização gráfica em tempo real. O retorno ao modo monitor é feito pelo comando `monitor`.

Além da saída periódica, o firmware transmite mensagens de confirmação ([OK]), erro ([ERRO]) e aviso ([AVISO]) em resposta a cada comando recebido, bem como alarmes de sobrepresão e volume quando detectados. O comando `status` exibe um resumo completo de todos os parâmetros configurados, tempos do ciclo, velocidades calculadas e estado do motor. A Figura 20 apresenta um exemplo da saída do Monitor Serial durante a operação do protótipo.

```

=== BVM ===
Comandos: 'start', 'stop', 'reset', '1'
Perfil do paciente: 'saudavel', 'dpoc', 'sdra'
Parâmetros: 'fr=', 'vt=', 'ie=', 'pausa=', 'comp=', 'res='
[OK] Ventilacao iniciada.
INSP Pmed=5.0 Vt=0 Mot=2/75
EXP Pmed=12.1 Vt=500 Mot=42/0
INSP Pmed=5.0 Vt=0 Mot=0/0
EXP Pmed=12.0 Vt=500 Mot=41/0
[ERRO] FR: 8-30
INSP Pmed=5.0 Vt=0 Mot=2/75
[OK] FR=12
EXP Pmed=13.9 Vt=500 Mot=57/0
[OK] DPOC: C=50 R=20 FR=12 IE=1:3 VT=500
EXP Pmed=6.3 Vt=500 Mot=0/0
INSP Pmed=9.7 Vt=0 Mot=75/75
EXP Pmed=8.2 Vt=500 Mot=0/0
INSP Pmed=5.0 Vt=0 Mot=6/75
--- STATUS ---
FR=12 VT=500 IE=1:3.0
C=50 R=20
Tciclo=5.00s Ti=1.25s TiAt=1.25s
PosAlvo=75/120 VelIns=2 VelExp=1EXP Pmed=11.4 Vt=500 Mot=34/0
[OK] COMP=10
[AVISO] Ppico > Pmax
EXP Pmed=5.0 Vt=500 Mot=0/0
[ALARME] SOBREPRESSAO: 41.3
INSP Pmed=52.7 Vt=0 Mot=75/75

!!! EMERGENCIA ACIONADA !!!
Motor parado. Digite 'reset' para liberar.
[OK] Emergencia liberada. Digite 'start'.

```

Figura 20 – Exemplo de saída do Monitor Serial durante operação do protótipo.

Fonte: elaboração própria (Arduino IDE).

Para ajuste de parâmetros via serial, o operador pode enviar comandos no formato `param=VALOR`, por exemplo:

```

fr=12      (define frequência respiratória para 12 irpm)
vt=500     (define volume corrente alvo para 500 mL)
ie=1:2     (define relação I:E para 1:2)
pausa=0.5  (define pausa inspiratória de 0,5 s)
comp=20    (define complacência para 20 mL/cmH20)
res=20     (define resistência para 20 cmH20/(L/s))

```

Adicionalmente, os comandos de controle `start`, `stop`, `reset` e `1` (emergência) permitem operar a máquina de estados do firmware. O sistema inicia no estado parado e só opera após o comando `start`.

Cada comando recebido é submetido à validação cruzada descrita na subseção [Validação Cruzada de Parâmetros \(3.3.4.8\)](#) antes de ser aplicado. Valores fora das faixas de segurança definidas na Tabela 6 são automaticamente rejeitados, e o firmware retorna uma

mensagem de erro pela serial informando o motivo da rejeição e os limites permitidos. Essa verificação garante que o operador não consiga, inadvertidamente, configurar parâmetros que coloquem o paciente em risco. A implementação completa do parser de comandos serial é apresentada no Apêndice B.

### 3.5.3.2 Interface Física

A interface física do protótipo utiliza componentes discretos de baixo custo para fornecer indicação visual imediata do estado do sistema. Os componentes adotados são:

- LED verde: indica operação normal, piscando a cada ciclo respiratório completado com sucesso;
- LED amarelo: indica alarme de média prioridade ativo (volume insuficiente ou volume excessivo);
- LED vermelho: indica alarme de alta prioridade ativo (sobrepresão ou emergência), permanecendo aceso continuamente até reconhecimento;
- *Buzzer*: emite sinal sonoro intermitente para alarmes de alta prioridade e sinal breve para alarmes de média prioridade;

A pinagem adotada para os componentes da IHM é: LED verde no pino D4, LED amarelo no pino D5, LED vermelho no pino D6 e *buzzer* no pino D7. A parada de emergência é acionada via comando serial (digitando “1”).

### 3.5.4 Modo de Simulação Embarcada

O firmware implementa uma simulação embarcada que permite validar o comportamento do controlador PI em tempo real no próprio microcontrolador, utilizando o modelo RC como planta virtual com parâmetros controlados e reprodutíveis. A planta virtual resolve numericamente a equação diferencial do modelo RC unicompartmental a cada período de amostragem, utilizando os mesmos parâmetros  $R$  e  $C$  das simulações no MATLAB.

A pressão alveolar simulada é atualizada pela discretização de Euler da Equação 2.1:

$$P_A[k + 1] = P_A[k] + \frac{\Delta t}{\tau} \cdot (u[k] - P_A[k]) \quad (3.25)$$

onde  $P_A[k]$  é a pressão alveolar simulada no instante  $k$ ,  $u[k]$  é a saída saturada do controlador PI (pressão aplicada, limitada entre 0 e 40 cmH<sub>2</sub>O),  $\tau = R \times C/1000$  é a constante de tempo em segundos (com  $C$  em mL/cmH<sub>2</sub>O convertido para L/cmH<sub>2</sub>O) e  $\Delta t$  é o período de amostragem (20 ms). A PEEP é somada como offset:  $P_{\text{med}} = P_A + P_0$ .

Com essa abordagem, o controlador PI opera em *malha fechada* real sobre uma planta que reproduz a dinâmica validada nas simulações, enquanto o motor responde fisicamente ao sinal de controle. É possível, assim, verificar:

- Se o controlador PI embarcado reproduz o comportamento observado nas simulações MATLAB (rastreamento da referência, *overshoot*, tempo de acomodação);
- Se a máquina de estados do ciclo respiratório opera corretamente nas fases de inspiração ativa, pausa inspiratória e expiração;
- Se o sistema de alarmes detecta corretamente condições de sobrepresão, volume insuficiente e apneia;
- Se o motor responde de forma coerente ao sinal de controle em cada cenário clínico.

Os parâmetros do cenário clínico ( $R$  e  $C$ ) podem ser alterados em tempo real pelos comandos `COMP=valor` e `RES=valor`, permitindo ao operador alternar entre os três cenários (saudável, DPOC e SDRA) durante a operação e observar a resposta do controlador e do motor a cada transição.

Os dados transmitidos pela porta serial seguem o mesmo formato tabulado descrito na subseção 3.5.3. Essa saída serial pode ser capturada e exportada para análise posterior, possibilitando a geração de gráficos comparativos entre o comportamento do PI embarcado e as simulações MATLAB.

## 3.6 VALIDAÇÃO

O objetivo desta etapa é verificar o funcionamento integrado de todos os subsistemas do protótipo (modelo RC, controlador PI e garra mecânica) e analisar os resultados das simulações com base nos parâmetros fisiológicos de referência. Conforme demonstrado na Subseção 3.4.3, o torque do motor Nema 17 é insuficiente para comprimir efetivamente a bolsa *BVM*, de modo que os testes são realizados com a garra operando sem carga da bolsa, validando o funcionamento mecânico do mecanismo e o comportamento do controlador sobre a planta virtual embarcada. A validação é estruturada em duas fases.

### 3.6.1 Integração dos Subsistemas

Na primeira fase, os três eixos do projeto são integrados: o modelo RC (planta virtual), o controlador PI (firmware embarcado no Arduino) e a garra mecânica (atuador físico). O objetivo é verificar se o conjunto eletrônico-mecânico opera de forma consistente e dentro dos limites de segurança estabelecidos, com a garra executando os ciclos de compressão comandados pelo controlador.

O procedimento de integração segue a ordem descrita no documento complementar de montagem: alimentação elétrica (fonte → L298N), fiação de controle (Arduino → L298N → Nema 17), conexão da IHM (LEDs, *buzzer*) e carregamento do firmware. A Figura 21 apresenta a bancada de testes com todos os componentes conectados, incluindo a IHM, o Arduino UNO com as conexões de controle, a ponte H L298N e o motor de passo Nema 17.

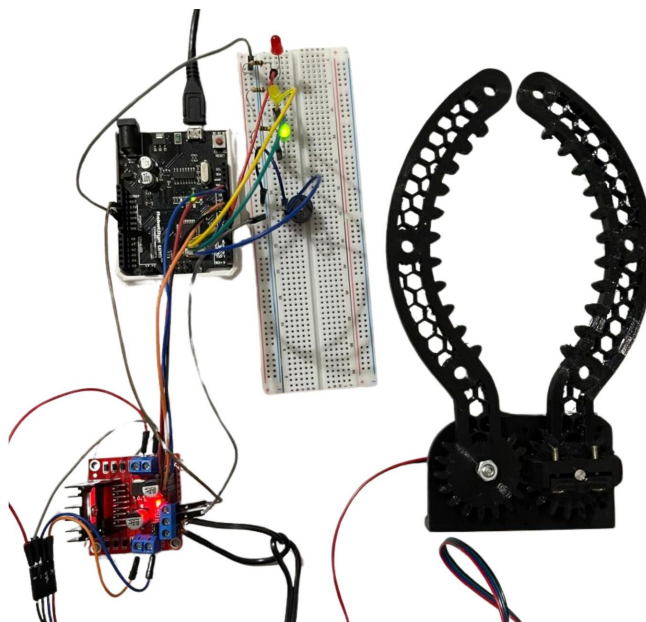


Figura 21 – Bancada de testes com os subsistemas integrados: IHM, Arduino UNO, ponte H L298N e motor Nema 17.

Fonte: autoria própria.

### 3.6.2 Verificação Funcional

A segunda fase avalia se a garra mecânica opera no ritmo e na amplitude esperados para cada configuração de parâmetros ventilatórios (frequência respiratória, volume corrente e relação I:E), executando os ciclos de abertura e fechamento sem a bolsa *BVM* acoplada. Dois aspectos são combinados: a observação qualitativa do comportamento mecânico da garra e a validação do controlador PI embarcado por meio do modo de simulação descrito na Subseção 3.5.4.

No modo de simulação embarcada, o controlador PI opera em *malha fechada* sobre a planta RC virtual executada no próprio Arduino, enquanto o motor responde fisicamente ao sinal de controle. Os dados de pressão de referência, pressão simulada, fase do ciclo e modo ativo são transmitidos pela porta serial a cada período de amostragem (20 ms), permitindo visualização em tempo real no Serial Plotter da Arduino IDE e exportação para análise posterior.

O firmware disponibiliza três perfis clínicos pré-configurados que configuram auto-

maticamente todos os parâmetros ventilatórios ao digitar o nome do cenário no Monitor Serial: `saudavel` ( $C = 50$ ,  $R = 5$ ,  $f_R = 15$ , I:E = 1:2,  $V_t = 500$ ), `dpoc` ( $C = 50$ ,  $R = 20$ ,  $f_R = 12$ , I:E = 1:3,  $V_t = 500$ ) e `sdra` ( $C = 20$ ,  $R = 5$ ,  $f_R = 15$ , I:E = 1:2,  $V_t = 400$ ). Alternativamente, cada parâmetro pode ser ajustado individualmente pelo formato `PARAM=VALOR`. O procedimento de verificação funcional consiste nos seguintes testes:

1. **Teste de ciclo respiratório (cenário saudável):** digitar `saudavel` na serial. Verificar no Serial Plotter que a pressão estimada acompanha o perfil esperado e que a garra abre e fecha no ritmo de 15 ciclos/min, reproduzindo qualitativamente o comportamento observado nas simulações MATLAB (Figura 13).
2. **Teste com cenário DPOC:** digitar `dpoc`. Verificar que a pressão estimada sobe mais lentamente ( $\tau = 1,00$  s), que o ciclo é mais longo e que o motor opera com a mesma amplitude mas em ritmo diferente, reproduzindo o padrão obstrutivo observado nas simulações (Figura 14).
3. **Teste com cenário SDRA:** digitar `sdra`. Verificar que a pressão estimada sobe rapidamente ( $\tau = 0,10$  s), que o pico de pressão é mais elevado e que o curso do motor é menor (400 mL), reproduzindo o comportamento da Figura 15.
4. **Teste de segurança:** durante operação, enviar o comando “1” pela porta serial e verificar que o motor para imediatamente e o LED vermelho acende. Verificar que os alarmes visuais e sonoros funcionam conforme a Tabela 7.
5. **Teste de ajuste em tempo real:** durante operação, alterar parâmetros via serial (por exemplo, `FR=20`) e verificar que o sistema transiciona suavemente para a nova configuração sem interrupção da ventilação.

Os testes funcionais foram realizados com o protótipo eletrônico-mecânico (Arduino UNO + L298N + Nema 17 + garra), sem a bolsa *BVM* acoplada, utilizando o modo de saída numérica do firmware (*plotter*) para visualização em tempo real no Serial Plotter da Arduino IDE. As três variáveis monitoradas são: pressão estimada  $P_{med}$  (cmH<sub>2</sub>O), calculada em tempo real pelo modelo RC embarcado no microcontrolador conforme a discretização, posição real do motor (passos) e posição-alvo (passos). No Serial Plotter da Arduino IDE, essas variáveis são exibidas na ordem de transmissão: a primeira (pressão estimada) em azul, a segunda (posição real do motor) em vermelho e a terceira (posição-alvo) em verde. Os resultados para cada cenário clínico são apresentados a seguir.

(a) Cenário saudável.

Os parâmetros de simulação considerados são:

$$C = 50 \text{ mL/cmH}_2\text{O}, R = 5 \text{ cmH}_2\text{O}/(\text{L/s}), f_R = 15 \text{ irpm}, \text{I:E} = 1:2, V_t = 500 \text{ mL}.$$

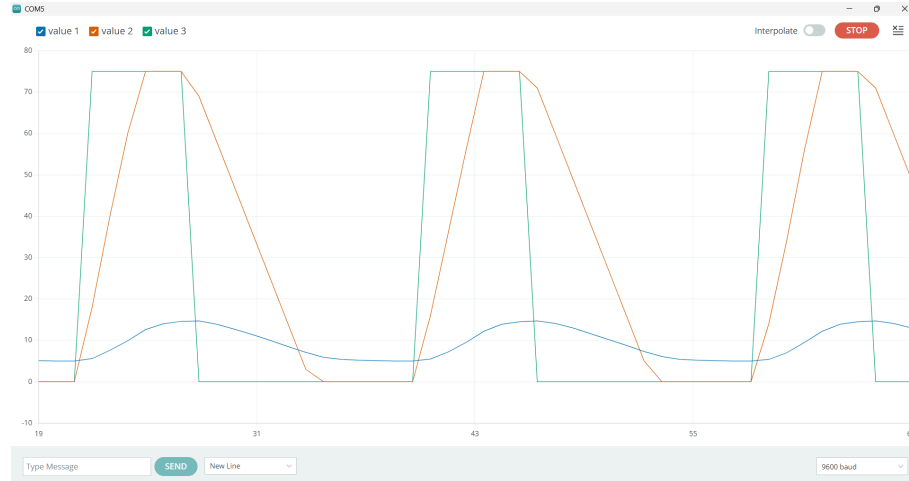


Figura 22 – Verificação funcional — cenário saudável: pressão estimada, posição real e posição-alvo do motor.

Fonte: elaboração própria (Arduino IDE).

Os resultados da verificação funcional para o cenário saudável são apresentados na Figura 22. O motor avança até aproximadamente 75 passos (correspondente ao volume alvo de 500 mL) e retorna completamente a zero a cada ciclo, com período de aproximadamente 4 s. A pressão estimada sobe de forma suave até cerca de 14 cmH<sub>2</sub>O, valor coerente com a relação estática  $P = V_t/C + P_0 = 500/50 + 5 = 15$  cmH<sub>2</sub>O. A pressão retorna ao valor de PEEP ( $\approx 5$  cmH<sub>2</sub>O) entre ciclos, sem acúmulo residual, confirmando a ausência de aprisionamento aéreo — comportamento esperado para  $\tau = 0,25$  s e  $T_{\text{exp}} = 2,67$  s  $\gg 3\tau$ .

(b) Cenário DPOC. Os parâmetros de simulação considerados são: ( $C = 50$  mL/cmH<sub>2</sub>O,  $R = 20$  cmH<sub>2</sub>O/(L/s),  $f_R = 12$  irpm, I:E = 1:3,  $V_t = 500$  mL).

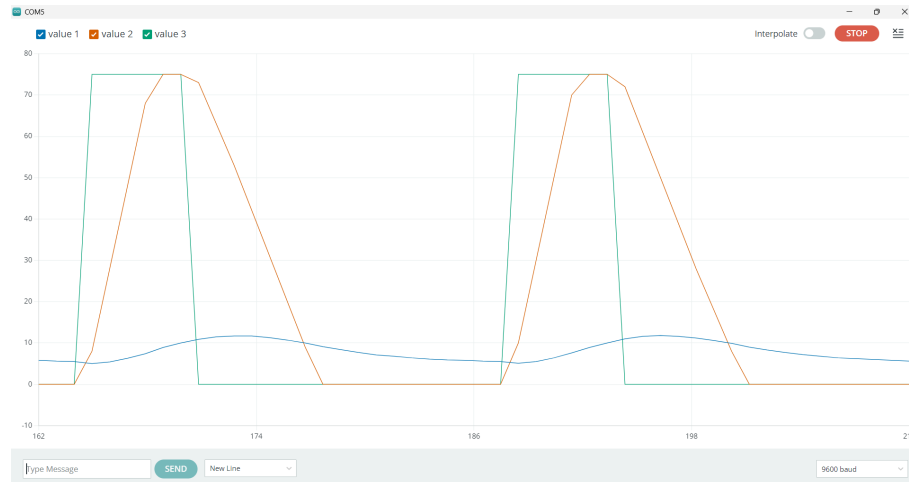


Figura 23 – Verificação funcional — cenário DPOC: pressão estimada, posição real e posição-alvo do motor.

Fonte: elaboração própria (Arduino IDE).

A Figura 23 apresenta os resultados para o cenário DPOC. O motor percorre o mesmo curso de 75 passos (mesmo  $V_t$ ), porém com ciclo mais longo ( $\approx 5$  s) devido à frequência reduzida. A pressão estimada sobe mais lentamente ( $\tau = 1,00$  s) e atinge apenas cerca de 11 cmH<sub>2</sub>O dentro do tempo inspiratório disponível, sem alcançar o valor estático de 15 cmH<sub>2</sub>O — reflexo da *resistência* elevada que limita o fluxo. A configuração com  $f_R = 12$  irpm e I:E = 1:3 proporciona  $T_{\text{exp}} = 3,75$  s  $>$   $3\tau = 3,00$  s, permitindo o retorno da pressão à PEEP entre ciclos.

(c) Cenário SDRA. Os parâmetros de simulação considerados são: ( $C = 20$  mL/cmH<sub>2</sub>O,  $R = 5$  cmH<sub>2</sub>O/(L/s),  $f_R = 15$  irpm, I:E = 1:2,  $V_t = 400$  mL).

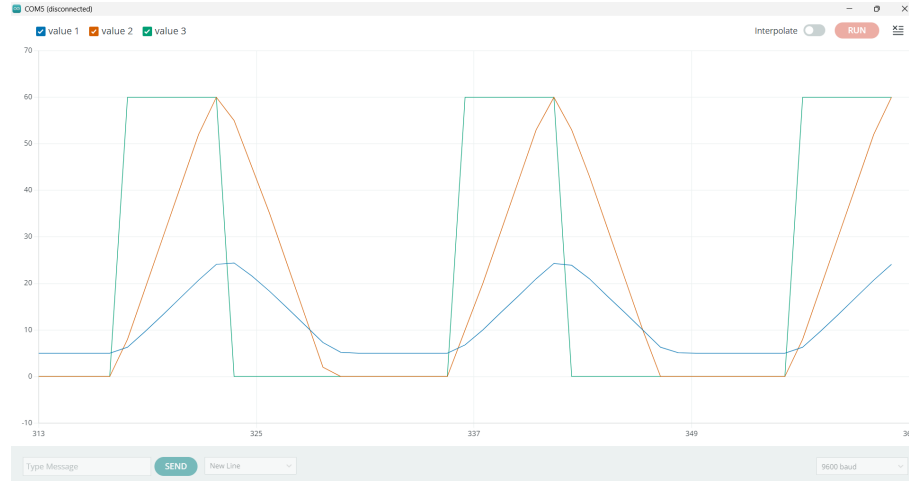


Figura 24 – Verificação funcional — cenário SDR: pressão estimada, posição real e posição-alvo do motor.

Fonte: elaboração própria (Arduino IDE).

A Figura 24 apresenta os resultados para o cenário SDR. O motor avança até aproximadamente 60 passos (correspondente a  $V_t = 400$  mL), curso visivelmente menor que nos cenários anteriores. A pressão estimada sobe rapidamente ( $\tau = 0,10$  s) e atinge cerca de 24 cmH<sub>2</sub>O, valor coerente com  $P = 400/20 + 5 = 25$  cmH<sub>2</sub>O. O pico de pressão é significativamente mais elevado que nos demais cenários, refletindo a *complacência* reduzida característica da SDR — o pulmão rígido exige maior pressão para acomodar o mesmo volume. A pressão retorna completamente à PEEP entre ciclos, sem aprisionamento aéreo.

Os três cenários apresentam diferenças visíveis e coerentes com a teoria: curso do motor proporcional ao volume alvo (75, 75 e 60 passos para 500, 500 e 400 mL), velocidade de subida da pressão determinada pela constante de tempo  $\tau$  (média, lenta e rápida para  $\tau = 0,25, 1,00$  e  $0,10$  s), e pico de pressão inversamente proporcional à *complacência* (14, 11 e 24 cmH<sub>2</sub>O). Esses resultados confirmam que o firmware embarcado reproduz qualitativamente os comportamentos observados nas simulações MATLAB e que os parâmetros ventilatórios ( $f_R, V_t, I:E, C, R$ ) afetam de forma coerente o comportamento mecânico do motor e a dinâmica da pressão estimada.

### 3.6.3 Limitações

As principais limitações identificadas nesta etapa são: a insuficiência de torque do motor Nema 17 para comprimir efetivamente a bolsa *BVM* (conforme demonstrado na Subseção 3.4.3), que restringiu os testes à operação da garra sem carga da bolsa; a influência de fatores não modelados (atrito, folgas mecânicas, variação da rigidez da bolsa com o uso) e a simplificação inerente ao modelo unicompartmental, que não representa a heterogeneidade regional da ventilação. A adequação do sistema de atuação e a integração

de sensores de pressão e fluxo no circuito respiratório, previstas como próximas etapas do projeto, permitirão a coleta de dados quantitativos para comparação direta com as simulações e o fechamento da malha de controle sobre variáveis clínicas reais. Essas direções fundamentam as propostas de evolução apresentadas nos trabalhos futuros.

## 4 Resultados

Os resultados são organizados em duas etapas: primeiro, os indicadores das simulações em *malha aberta*, já apresentados na Tabela 3 durante o desenvolvimento do modelo; em seguida, o desempenho do controlador PI em *malha fechada*, detalhado a seguir.

### 4.1 RESULTADOS DO CONTROLADOR PI

Os ganhos do controlador PI ( $K_p = 5$ ,  $K_i = 20$ ), obtidos por alocação de polos e confirmados pelo método IMC (Subseção 3.3.4.3), são aplicados de forma fixa aos três cenários clínicos. A lei de controle é discretizada pela transformação bilinear de Tustin (Equação 3.20) para execução no Arduino com período de amostragem de 20 ms.

Para o cenário saudável (Figura 13), a pressão alveolar  $y(t)$  acompanha a referência  $r(t)$  com tempo de acomodação de aproximadamente 0,66 s e *overshoot* nulo em regime cíclico. No primeiro ciclo, observa-se um leve transitório inicial (pressão de pico de  $\approx 15,5$  cmH<sub>2</sub>O) decorrente da condição inicial, que desaparece nos ciclos subsequentes. O sinal de controle  $u(t)$  apresenta pico de  $\approx 30$  cmH<sub>2</sub>O no início de cada inspiração — quando o erro é máximo — e decai suavemente à medida que a pressão alveolar se aproxima da referência, comportamento esperado de um controlador PI diante de uma variação em degrau. O bloco de saturação (0 a 40 cmH<sub>2</sub>O) garante que o sinal de controle permaneça dentro dos limites de segurança. O volume pulmonar oscila entre 0 e  $\approx 500$  mL, retornando completamente a zero a cada ciclo, sem acúmulo de ar residual.

Para o cenário SDRA (Figura 15), o controlador apresenta desempenho comparável ao cenário saudável, com tempo de acomodação de aproximadamente 1,0 s e *overshoot* nulo. A resposta naturalmente rápida da planta ( $\tau = 0,10$  s) facilita o rastreamento da referência, e a pressão alveolar retorna ao valor de PEEP a cada ciclo sem dificuldade. O sinal de controle apresenta perfil mais “quadrado” em comparação ao cenário saudável, pois a planta rápida exige menor esforço transitório do controlador — a pressão alveolar acompanha a referência quase instantaneamente, e o PI mantém  $u(t)$  próximo ao valor de regime durante a maior parte da inspiração. Cabe observar que o volume corrente neste cenário difere conforme a configuração: em *malha aberta* com pressão fixa de 20 cmH<sub>2</sub>O, o volume resultante é de  $\approx 300$  mL ( $V_t = C \times \Delta P = 20 \times 15$ ), conforme a Tabela 3; em *malha fechada*, o preset SDRA configura  $V_t = 400$  mL como alvo, e o controlador PI eleva a pressão aplicada acima de 20 cmH<sub>2</sub>O para atingir a referência  $P_{\text{ref}} = 400/20 + 5 = 25$  cmH<sub>2</sub>O, resultando em volume de  $\approx 400$  mL.

Para o cenário DPOC (Figura 14), o comportamento é compatível com a fisiopatologia obstrutiva. A pressão alveolar sobe mais lentamente ( $\tau = 1,00$  s) e atinge  $\approx 28-29$  cmH<sub>2</sub>O dentro do tempo inspiratório disponível, sem alcançar completamente a referência de 30 cmH<sub>2</sub>O. O sinal de controle atinge valores mais altos ( $\approx 30-35$  cmH<sub>2</sub>O) para compensar a limitação de fluxo, sem ultrapassar o limite de 40 cmH<sub>2</sub>O. O volume pulmonar não retorna completamente a zero entre ciclos, apresentando um residual de  $\approx 100$  mL, conforme a análise da Subseção 3.3.4.7. Esses efeitos são contornáveis pelo ajuste de  $f_R$  e I:E via IHM.

## 4.2 DISCUSSÃO

Os resultados obtidos neste trabalho permitem uma análise sob três perspectivas: o atendimento aos objetivos propostos, a comparação com os trabalhos da literatura revisada e a conformidade com os requisitos normativos.

### 4.2.1 Atendimento aos Objetivos

O objetivo geral do trabalho era projetar e implementar a automação de uma garra mecânica que comprimissem a bolsa da *BVM* respeitando intervalos de tempo e volume adequados a diferentes condições clínicas. Nesse sentido, o protótipo desenvolvido atende parcialmente ao proposto: a garra impressa em 3D, acionada pelo motor Nema 17 via ponte H L298N, opera em ciclos controlados pelo firmware embarcado no Arduino UNO, executando os movimentos de compressão no ritmo e na amplitude comandados pelo controlador. Contudo, conforme demonstrado na estimativa de torque (Subseção 3.4.3), o motor não dispõe de torque suficiente para comprimir efetivamente a bolsa *BVM* Foyomed, de modo que os testes foram realizados com a garra operando sem carga da bolsa. Os três cenários clínicos simulados (saudável, DPOC e SDRA) produzem comportamentos distintos e compatíveis com a fisiologia esperada, conforme demonstrado nas Figuras 13 a 15 e nas verificações funcionais (Figuras 22 a 24).

Quanto aos objetivos específicos, a modelagem da dinâmica pulmonar pelo modelo RC foi realizada e validada em dois ambientes de simulação (Falstad e MATLAB). A garra foi projetada e fabricada por impressão 3D. O controle em tempo real foi implementado com controlador PI discretizado por Tustin, operando sobre a planta virtual embarcada. O sistema de supervisão inclui alarmes de sobrepresão e volume, parada de emergência e validação cruzada de parâmetros. A IHM permite ajustes em tempo real via porta serial. A detecção de desconexões e fugas, entretanto, não foi implementada nesta etapa, por depender de sensores de pressão e fluxo ainda não integrados ao protótipo.

## 4.2.2 Comparação com a Literatura

O controlador PI com ganhos fixos ( $K_p = 5$ ,  $K_i = 20$ ) adotado neste trabalho representa a abordagem mais simples dentre as estratégias revisadas na literatura. Essa simplicidade é, ao mesmo tempo, sua principal vantagem e sua limitação mais evidente.

Em comparação com o trabalho de [Ghafoor, Hayat e Hussain \(2021\)](#), que utilizou PSO para otimizar os ganhos PID sobre o mesmo tipo de planta RC, o controlador aqui proposto dispensa qualquer algoritmo de otimização: os ganhos são obtidos analiticamente por alocação de polos, com resultado confirmado pelo método IMC. A contrapartida é que os ganhos de Ghafoor et al., por serem otimizados para minimizar uma função de custo global, tendem a oferecer melhor compromisso entre *overshoot* e tempo de acomodação quando os parâmetros da planta variam. No presente trabalho, o cancelamento polo-zero é exato apenas para o cenário saudável ( $\tau = 0,25$  s), resultando em *overshoot* nulo nesse caso. No cenário DPOC ( $\tau = 1,00$  s), o cancelamento parcial produz *overshoot* de 9,4%, valor que, embora não ideal, mantém a pressão de pico (15,9 cmH<sub>2</sub>O) dentro dos limites de segurança. Uma sintonia por PSO poderia, em princípio, reduzir esse *overshoot* ao custo de maior complexidade computacional, pouco compatível com as restrições do Arduino UNO.

O controlador Fuzzy-PID de [Alam et al. \(2021b\)](#) e o Adaptive Fuzzy Sliding Mode Control de [Alam et al. \(2021a\)](#) representam estratégias mais sofisticadas, capazes de ajustar ganhos em tempo real conforme a condição do paciente. Esses controladores foram validados sobre um modelo em cadeia (*blower-mangueira-paciente*) com *complacência* não linear, cenário mais realista que o modelo RC linear adotado neste trabalho. A vantagem do Fuzzy-PID e do SMC adaptativo é a capacidade de manter desempenho consistente mesmo quando  $R$  e  $C$  variam de forma imprevisível durante a ventilação. No protótipo aqui desenvolvido, essa adaptação não é automática: cabe ao operador selecionar o cenário clínico via IHM, e os ganhos permanecem fixos. Essa limitação é reconhecida, e a implementação de *gain scheduling* ou lógica fuzzy é proposta como evolução futura.

Quanto ao projeto mecânico, o trabalho de [Garmendia et al. \(2021\)](#) utilizou uma correia para comprimir a bolsa e incorporou controle ADRC com sensoriamento de pressão real, obtendo desempenho superior ao PID convencional. O protótipo aqui proposto adota mecanismo diferente (garra tipo pinça com engrenagens), sem sensoriamento de pressão no circuito do paciente. A ausência de sensor de pressão é a diferença mais significativa em relação a Garmendia et al.: enquanto o ADRC opera sobre variáveis clínicas medidas, o PI deste trabalho opera sobre uma planta virtual embarcada. Essa escolha, embora limite a validação clínica, permite o desenvolvimento e a verificação do controlador em ambiente controlado e reprodutível, como etapa prévia à integração com sensoriamento físico.

O projeto SVASTA ([VENKATESAN et al., 2020](#)) e o COMBIOVENT ([ESPINOZA et](#)

*al.*, 2024) abordam a premissa de automação de baixo custo para a *BVM*. O SVASTA utiliza engrenagens e controlador digital em *malha fechada*, enquanto o COMBIOVENT enfatiza a análise de esforços mecânicos e a fadiga da bolsa. O protótipo aqui desenvolvido se alinha a essas propostas no aspecto construtivo (impressão 3D, motor de passo, componentes de baixo custo), diferenciando-se pela formalização do projeto de controle (alocação de polos, discretização por Tustin, verificação de estabilidade nos domínios  $s$  e  $z$ ) e pela simulação embarcada como estratégia de validação.

Em relação à implementação em FPGA proposta por Al-Qassar *et al.* (2023), o Arduino UNO utilizado neste trabalho apresenta limitações conhecidas: execução sequencial, *jitter* no período de amostragem (agravado pelo uso de `delayMicroseconds`) e aritmética em ponto flutuante por *software*. Essas limitações não comprometeram o funcionamento do protótipo nos testes realizados, mas podem se tornar relevantes em cenários que exijam resposta mais rápida ou maior precisão numérica. A migração para um microcontrolador com timer dedicado (ou mesmo para FPGA, em versões futuras) é uma direção natural de evolução.

### 4.2.3 Conformidade com Requisitos Normativos

A norma ISO 10651-4 (INTERNATIONAL ORGANIZATION FOR STANDARDIZATION, 2023) estabelece requisitos para ressuscitadores manuais que servem como referência para o dimensionamento do protótipo. Para pacientes com massa corporal superior a 40 kg, a norma especifica volume entregue ( $V_{del}$ ) mínimo de 600 mL em condições de teste. Nas simulações do cenário saudável, o volume corrente obtido foi de aproximadamente 750 mL em *malha aberta* e 500 mL em *malha fechada* (com referência configurada para esse valor), ambos acima do limiar normativo. No cenário SDRA, o volume de 300 mL é inferior ao mínimo da norma, porém compatível com a estratégia de ventilação protetora (4-6 mL/kg) recomendada para essa condição (HAJJAR *et al.*, 2024). Essa aparente contradição entre o requisito normativo e a prática clínica decorre do fato de que a ISO 10651-4 define requisitos para o dispositivo (capacidade de entrega), não para a prescrição ventilatória (volume efetivamente administrado ao paciente). O protótipo é capaz de entregar volumes superiores a 600 mL quando configurado para tal, atendendo ao requisito de capacidade.

Quanto à limitação de pressão, a norma exige capacidade de gerar ao menos 30 cmH<sub>2</sub>O para pacientes pediátricos e limita a pressão a 45 cmH<sub>2</sub>O. O protótipo adota limite de 40 cmH<sub>2</sub>O, mais conservador que o normativo, implementado tanto no bloco de saturação do controlador quanto no sistema de alarmes. Em nenhum dos três cenários simulados a pressão de pico ultrapassou esse limite: 15 cmH<sub>2</sub>O no cenário saudável, 15,9 cmH<sub>2</sub>O no DPOC e 25 cmH<sub>2</sub>O no SDRA.

É necessário ressaltar, contudo, que a conformidade aqui analisada é conceitual. A norma ISO 10651-4 exige ensaios com pulmão de teste mecânico certificado, robustez

ambiental (choque, imersão, temperatura e umidade) e verificação dos conectores padronizados (15 F/22 M), requisitos que não foram avaliados experimentalmente neste trabalho. A validação normativa completa é proposta como trabalho futuro.

#### 4.2.4 Efeito da Saturação e Desempenho Dinâmico

Os tempos de acomodação obtidos nas simulações em *malha fechada* (0,66 s no cenário saudável, 1,08 s no DPOC e 1,00 s no SDRA) são significativamente superiores ao valor teórico de  $3/\alpha = 0,15$  s previsto pela análise no domínio  $s$ . Essa discrepância se explica pela presença do bloco de saturação (0 a 40 cmH<sub>2</sub>O), que limita o esforço de controle nos instantes iniciais de cada ciclo inspiratório. Na análise teórica, o controlador dispõe de esforço ilimitado para forçar a convergência; na simulação, o sinal de controle é restrito a 40 cmH<sub>2</sub>O, prolongando o transitório. Ainda assim, os tempos obtidos são inferiores ao tempo inspiratório mínimo previsto de 1 s (Tabela 4), de modo que a pressão alveolar atinge a referência antes do final da fase inspiratória nos cenários saudável e SDRA. No cenário DPOC, a constante de tempo elevada da planta ( $\tau = 1,00$  s) impede que a pressão atinja completamente a referência dentro do tempo inspiratório disponível, comportamento coerente com a fisiopatologia obstrutiva.

#### 4.2.5 Implicações Clínicas do *Overshoot* no Cenário DPOC

No cenário DPOC, o *overshoot* de 9,4% resulta em pressão de pico de aproximadamente 15,9 cmH<sub>2</sub>O, valor que permanece abaixo do limite de segurança de 40 cmH<sub>2</sub>O com margem considerável. Do ponto de vista clínico, *overshoot* de pressão em pacientes com padrão obstrutivo é uma preocupação quando a pressão de pico se aproxima de limiares de barotrauma (tipicamente acima de 35-40 cmH<sub>2</sub>O). No caso simulado, a pressão de pico está longe desse limiar, de modo que o *overshoot* observado não representa risco adicional. Cabe notar, porém, que esse resultado foi obtido com parâmetros ventilatórios padrão ( $f_R = 15$  irpm, I:E = 1:2), configuração que, conforme discutido na Subseção 3.3.4.7, não é a mais adequada para pacientes com DPOC. Com parâmetros ajustados ( $f_R = 12$  irpm, I:E = 1:3), o aprisionamento aéreo é reduzido e o comportamento do controlador tende a se aproximar do cenário saudável.

#### 4.2.6 Limitações da Validação

A principal limitação deste trabalho é a ausência de validação experimental com instrumentação de pressão e volume no circuito respiratório. Todos os resultados de *malha fechada* foram obtidos sobre a planta virtual (modelo RC embarcado), e não sobre um pulmão de teste mecânico ou paciente real. Adicionalmente, a insuficiência de torque do motor Nema 17 (Subseção 3.4.3) impediu a compressão efetiva da bolsa *BVM* durante os testes, de modo que a garra operou sem carga da bolsa. Essa abordagem permite verificar

a lógica do controlador, a estabilidade do sistema e a coerência dos resultados com a teoria, mas não captura efeitos práticos como: a dinâmica real da bolsa *BVM* (não linearidade da *complacência* da bolsa, histerese do material), as perdas de pressão nas válvulas e conexões, o atrito e as folgas mecânicas da garra, e a variabilidade do motor de passo sob carga. A adequação do sistema de atuação e a integração de sensores de pressão diferencial e fluxo, previstas como próximas etapas, permitirão fechar a malha sobre variáveis clínicas reais e confrontar os resultados simulados com medições experimentais.

## 5 Conclusão

Este trabalho teve como objetivo geral projetar e implementar a automação de uma garra mecânica para compressão da Bolsa Válvula Máscara (*BVM*), respeitando intervalos de tempo e volume adequados a diferentes condições clínicas. Para isso, foram desenvolvidos e integrados em um único protótipo de baixo custo: a modelagem matemática do sistema respiratório, o projeto mecânico da garra, o sistema de controle em *malha fechada*, o sistema de segurança com intertravamentos e alarmes, e a interface de operação.

O primeiro objetivo específico — compreender os princípios fisiológicos da ventilação manual e modelar a dinâmica pulmonar — foi atendido pela implementação do modelo unicompartimental RC, fundamentado nos parâmetros de *resistência* e *complacência* reportados na literatura (BATES, 2009). O modelo foi validado em dois ambientes de simulação complementares: o Falstad, para verificação qualitativa da analogia elétrica, e o MATLAB, para análise quantitativa. As simulações em *malha aberta* reproduziram de forma coerente os comportamentos clínicos esperados nos três cenários avaliados — adulto saudável ( $\tau = 0,25$  s), DPOC ( $\tau = 1,00$  s) e SDRA ( $\tau = 0,10$  s) —, incluindo o aprisionamento aéreo no cenário obstrutivo e a resposta rápida com volume reduzido no cenário restritivo.

O segundo objetivo — projetar a garra para impressão 3D — foi cumprido com o desenvolvimento de uma garra biarticulada com braços côncavos e acionamento por engrenagens, modelada no Onshape e fabricada em filamento PETG. A análise de torque revelou que o motor Nema 17 (0,56 N·m) é insuficiente para comprimir efetivamente a bolsa ( $T_{nec} = 1,86$  N·m), limitação que restringe a validação mecânica ao funcionamento sem carga da bolsa. Apesar dessa restrição, o protótipo demonstrou integridade estrutural e funcionamento mecânico adequado nos testes cíclicos.

O terceiro objetivo — implementar o controle em tempo real — foi alcançado pela síntese de um controlador PI com ganhos fixos ( $K_p = 5$ ,  $K_i = 20$ ), projetado por alocação de polos e confirmado pelo método IMC. O controlador foi discretizado pela transformação bilinear de Tustin com período de amostragem de 20 ms, conforme a diretriz de Ogata (2010), e embarcado no Arduino UNO. Os resultados das simulações em *malha fechada* demonstram que um único conjunto de ganhos atende aos três cenários clínicos: nos cenários saudável e SDRA, o controlador apresenta *overshoot* nulo e erro em regime permanente desprezível; no cenário DPOC, o *overshoot* de 9,4% resulta em pressão de pico de 15,9 cmH<sub>2</sub>O, dentro dos limites de segurança. A verificação funcional com o firmware embarcado e o motor operando sem carga confirmou que o controlador PI reproduz qualitativamente o comportamento observado nas simulações MATLAB.

O quarto objetivo — implementar supervisão e intertravamentos — foi atendido pelo sistema de segurança embarcado no firmware, que incorpora limites de pressão (40 cmH<sub>2</sub>O), alarmes de sobrepresão e de volume insuficiente, detecção de apneia e parada de emergência. O anti-windup por *clamping* garante que o integrador do controlador não acumule erro durante a saturação do sinal de controle. A validação de parâmetros no momento da configuração impede combinações que violem restrições fisiológicas. O quinto objetivo — projetar uma IHM para modificações em tempo real — foi cumprido pela interface serial que permite ao operador selecionar cenários clínicos pré-configurados, ajustar parâmetros individuais e monitorar o estado do sistema em dois modos (textual e gráfico), complementada por interface física com LEDs e *buzzer* para indicação de estado em campo.

A principal contribuição deste trabalho é a demonstração de que é viável integrar, em um único protótipo de baixo custo, modelagem do sistema respiratório, controle em *malha fechada*, projeto mecânico, sistema de segurança e interface de operação para automação da *BVM*. A robustez do controlador PI com ganhos fixos em três cenários clínicos distintos, validada tanto por simulação quanto por verificação embarcada, reforça a aplicabilidade da abordagem para um dispositivo destinado a cenários de emergência com equipe reduzida. A formalização do projeto de controle — alocação de polos, verificação de estabilidade nos domínios  $s$  e  $z$ , confirmação por IMC e discretização por Tustin — diferencia este trabalho de propostas similares na literatura que adotam controladores sem fundamentação analítica explícita. A estratégia de simulação embarcada, que utiliza o modelo RC como planta virtual no próprio microcontrolador, constitui uma abordagem de validação reprodutível e de baixo custo, aplicável a outros projetos de controle em estágio de prototipagem.

As principais limitações residem na ausência de validação experimental com instrumentação de pressão e volume no circuito respiratório — todos os resultados de *malha fechada* foram obtidos sobre a planta virtual embarcada, e não sobre um pulmão de teste mecânico ou paciente real. A insuficiência de torque do motor Nema 17 impediu a compressão efetiva da bolsa durante os testes, de modo que a garra operou sem carga, permitindo verificar a lógica do controlador e a coerência dos resultados com a teoria, mas sem capturar efeitos práticos como a não linearidade da bolsa, as perdas nas válvulas e o atrito mecânico da garra. O modelo unicompartmental, embora adequado para o escopo do trabalho, não representa a heterogeneidade regional da ventilação nem comportamentos não lineares da *elastância*. O tratamento da *BVM* como fonte de pressão ideal desconsidera a relação não linear entre deslocamento da garra e volume entregue. No firmware, o uso de `delayMicroseconds` compromete a regularidade do período de amostragem, e uma implementação baseada em interrupções por *timer* seria mais adequada em versões futuras (UECHI, 2012).

## 5.1 TRABALHOS FUTUROS

Como perspectivas para trabalhos futuros, propõem-se as seguintes direções:

- Inclusão de sensores de pressão diferencial e fluxo no circuito respiratório, permitindo realimentação direta das variáveis clínicas e melhorando a precisão do controlador e dos alarmes;
- Implementação de controle adaptativo com *gain scheduling* ou técnicas baseadas em lógica fuzzy, permitindo ajuste automático dos ganhos PI conforme a condição clínica estimada do paciente;
- Substituição dos dados virtuais gerados pelo circuito análogo por sinais reais adquiridos diretamente do paciente, por meio de sensores de SpO<sub>2</sub> e capnografia;
- Validação do protótipo com pulmão de teste mecânico certificado, conforme os requisitos da norma ISO 10651-4;
- Desenvolvimento de uma IHM dedicada com *display* gráfico e botões físicos, eliminando a dependência do computador para monitoramento e ajuste de parâmetros;
- Estudo de fadiga mecânica da garra e da bolsa *BVM* sob operação prolongada, visando garantir a confiabilidade do sistema em cenários reais de emergência;
- Adequação do sistema de atuação para viabilizar a compressão efetiva da bolsa *BVM*, seja pela inclusão de um estágio de redução mecânica (engrenagens redutoras) entre o motor e a garra, seja pela substituição do motor Nema 17 por um atuador de maior torque, conforme a análise da Subseção 3.4.3.

# Referências

- ALAM, M.S. *et al.* Adaptive Fuzzy Sliding Mode Control of a Pressure-Controlled Artificial Ventilator. *Journal of Healthcare Engineering*, Hindawi, v. 2021, p. 1926711, 2021. DOI: [10.1155/2021/1926711](https://doi.org/10.1155/2021/1926711). Citado 2 vezes nas páginas 26, 78.
- ALAM, M.S. *et al.* Fuzzy PID Control for Respiratory Systems. *Journal of Healthcare Engineering*, Hindawi, v. 2021, p. 7118711, 2021. DOI: [10.1155/2021/7118711](https://doi.org/10.1155/2021/7118711). Citado 2 vezes nas páginas 26, 78.
- ALMEIDA, RG de. Modelo dinâmico da elastância pulmonar. [peb.ufjf.br](http://peb.ufjf.br), 2007. Citado 4 vezes nas páginas 14, 20, 30.
- BATES, J.H.T. *Lung Mechanics: An Inverse Modeling Approach*. Cambridge University Press, 2009. Citado 12 vezes nas páginas 14, 16, 19, 20, 23, 30, 82.
- BATES, J.H.T.; IRVIN, C.G. Respiratory system dynamical mechanical properties: modeling in time and frequency domain. *Biophysical Reviews and Letters*, Springer, 2011. DOI: [10.1007/s12551-011-0048-5](https://doi.org/10.1007/s12551-011-0048-5). Citado 1 vez na página 21.
- BORRELLO, M. Model-free volume and pressure cycled control of automatic bag valve mask ventilator. *AIMS Bioengineering*, v. 8, n. 3, p. 192–207, 2021. DOI: [10.3934/bioeng.2021017](https://doi.org/10.3934/bioeng.2021017). Citado 4 vezes nas páginas 16, 24, 44, 60.
- BRUNNER, Josef X.; PIQUILLOUD, Lise *et al.* Six methods to determine expiratory time constants in mechanically ventilated patients: a prospective observational physiology study. *Intensive Care Medicine Experimental*, Springer, v. 12, p. 25, 2024. DOI: [10.1186/s40635-024-00612-z](https://doi.org/10.1186/s40635-024-00612-z). Citado 4 vezes nas páginas 19, 20, 54, 64.
- CHATBURN, R.L.; MIRELES-CABODEVILA, E. Closed-loop control of mechanical ventilation. *Respiratory Care*, v. 56, n. 1, p. 85–98, 2011. Citado 1 vez na página 23.
- ESPINOZA, J. *et al.* Mechanical Design and Modeling of COMBIOVENT Pulmonary Ventilator. *In: LECTURE Notes in Networks and Systems*. Springer, 2024. DOI: [10.1007/978-3-031-49401-7\\_1](https://doi.org/10.1007/978-3-031-49401-7_1). Citado 4 vezes nas páginas 28, 58, 61, 78.
- FAUSTINO, A.C. *et al.* Comparison of Three Different Mathematical Models Applied to Respiratory Mechanics During Bronchoconstriction. *In: IFMBE Proceedings*. Springer, 2019. p. 7–11. DOI: [10.1007/978-981-13-2119-1\\_2](https://doi.org/10.1007/978-981-13-2119-1_2). Citado 1 vez na página 20.
- GARMENDIA, N. *et al.* Mechatronic Design and Active Disturbance Rejection Control of a Bag Valve-Based Mechanical Ventilator. *Journal of Medical Devices*, ASME, v. 15, n. 3, p. 031006, 2021. DOI: [10.1115/1.4050271](https://doi.org/10.1115/1.4050271). Citado 2 vezes nas páginas 27, 78.

GHAFOOR, U.; HAYAT, M.F.; HUSSAIN, S.N. Swarm optimization approach to design PID controller for artificially ventilated human respiratory system. *IEEE Access*, 2021. Citado 3 vezes nas páginas [24](#), [25](#), [78](#).

HAJJAR, L.A. *et al.* *Emergências Cardiovasculares e Ventilação no Atendimento Pré-Hospitalar*. Atheneu, 2024. Citado 8 vezes nas páginas [14](#), [22](#), [30](#), [35](#), [37](#), [41](#), [63](#), [79](#).

HALL, John E.; HALL, Michael E. *Guyton and Hall Textbook of Medical Physiology*. 14. ed.: Elsevier, 2021. Citado 2 vezes na página [18](#).

HESS, Dean R. Respiratory mechanics in mechanically ventilated patients. *Respiratory Care*, v. 59, n. 11, p. 1773–1794, 2014. DOI: [10.4187/respcare.03410](https://doi.org/10.4187/respcare.03410). Citado 3 vezes nas páginas [19](#), [63](#).

HUSSAIN, S.N. *et al.* Cascaded robust control of mechanical ventilator using fractional order sliding mode control. *Mathematical Biosciences and Engineering*, AIMS Press, v. 19, n. 2, p. 1332–1354, 2022. DOI: [10.3934/mbe.2022061](https://doi.org/10.3934/mbe.2022061). Citado 1 vez na página [27](#).

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION. *Lung ventilators — Part 4: Particular requirements for user-powered resuscitators*. ISO, 2023. Citado 6 vezes nas páginas [22](#), [41](#), [42](#), [63](#), [79](#).

KIM, T.Y. *et al.* Automatic Mechanical Ventilation vs Manual Bag Ventilation During CPR in Out-of-Hospital Cardiac Arrest. *Resuscitation*, Elsevier, 2024. Citado 2 vezes nas páginas [14](#), [16](#).

LOPES, F. *Desempenho Essencial de Ventiladores Pulmonares no Modo Volume Controlado Segundo a Norma ABNT NBR ISO 80601-2-12*. 2023. Diss. (Mestrado) – Universidade Federal. Citado 2 vezes nas páginas [27](#), [63](#).

LUCANGELO, Umberto; BERNABÉ, Franco; BLANCH, Lluís. Respiratory mechanics derived from signals in the ventilator circuit. *Respiratory Care*, v. 50, n. 1, p. 55–67, 2005. Citado 4 vezes nas páginas [19](#), [42](#).

OGATA, Katsuhiko. *Engenharia de Controle Moderno*. 5. ed. São Paulo: Pearson Prentice Hall, 2010. Citado 3 vezes nas páginas [47](#), [50](#), [82](#).

PEREIRA, ML; GRANDE, C. Modelagem e Controle Preciso do Tempo de Inspiração de um Ventilador Pulmonar do tipo Ambu Automatizado. *Trabalho de Conclusão de Curso (Engenharia Elétrica)–Universidade Federal de Mato Grosso do Sul, Campo Grande, MS*, 2021. Citado 1 vez na página [14](#).

PINTO, J.R. Controlo da ventilação. *Revista Portuguesa de Pneumologia*, Elsevier, 2005. Citado 1 vez na página [23](#).

AL-QASSAR, A.A. *et al.* Enhanced performance of the mechanical respiratory system by FPGA-digital based on PID Controller. *Cogent Engineering*, Taylor e Francis, v. 10, n. 2, p. 2247865, 2023. DOI: [10.1080/23311916.2023.2247865](https://doi.org/10.1080/23311916.2023.2247865). Citado 3 vezes nas páginas 25, 79.

UECHI, E.Y. *Confiabilidade Metrológica de Ventiladores Pulmonares para Cuidados Críticos*. 2012. Diss. (Mestrado) – Pontifícia Universidade Católica do Rio de Janeiro. Citado 2 vezes nas páginas 27, 83.

VENKATESAN, R. *et al.* SVASTA, PRANA and VaU: Three Novel Ventilators from Space Community. *Transactions of the Indian National Academy of Engineering*, Springer, v. 5, p. 355–364, 2020. DOI: [10.1007/s41403-020-00149-z](https://doi.org/10.1007/s41403-020-00149-z). Citado 3 vezes nas páginas 27, 58, 78.

WARD, Jeremy P. T.; WARD, Jane; LEACH, Richard M. *Fisiologia Básica do Sistema Respiratório*. Manole, 2012. Citado 2 vezes nas páginas 14, 18.

WEST, John B.; LUKS, Andrew M. *West's Respiratory Physiology: The Essentials*. 11. ed.: Wolters Kluwer, 2021. Citado 3 vezes nas páginas 18, 19.

# APÊNDICE A – Firmware Principal

O código-fonte a seguir implementa o firmware completo do protótipo, incluindo a máquina de estados do ciclo respiratório, o controlador PI discretizado por Tustin com anti-windup, a estimativa de pressão pelo modelo RC embarcado, o sistema de alarmes e a transmissão de dados pela porta serial.

```
#include <stdlib.h>

#define PIN_IN1  8
#define PIN_IN2  9
#define PIN_IN3  11
#define PIN_IN4  12

#define PIN_LED_VERDE    4
#define PIN_LED_AMARELO  5
#define PIN_LED_VERMELHO 6
#define PIN_BUZZER       7

// ===== CONSTANTES =====

#define DT_MS          20
#define DT_S           0.02f
#define PASSOS_MAX     120
#define VT_MAX         800.0f
#define PMAX_DEFAULT  40.0f
#define PEEP_DEFAULT   5.0f
#define HISTERESE_P    2.0f
#define PRINT_INTERVAL 2000

#define VEL_MIN        1
#define VEL_MAX        6
#define DELAY_PASSO_US 1500

// Controlador PI (Tustin): Kp=5, Ki=20, T=0.02s
// a0 = Kp + Ki*T/2 = 5.2
// a1 = -Kp + Ki*T/2 = -4.8
```

```
#define PI_A0 5.2f
#define PI_A1 -4.8f

// ===== ESTADOS =====

enum Estado {
    ST_PARADO,
    ST_INSPIRACAO,
    ST_PAUSA_INSP,
    ST_EXPIRACAO,
    ST_EMERGENCIA
};

// ===== VARIAVEIS GLOBAIS =====

volatile Estado estado = ST_PARADO;

float freqResp = 15.0f;
float volumeAlvo = 500.0f;
float relacaoIE = 2.0f;
float pausaInsp = 0.0f;
float PEEP = PEEP_DEFAULT;
float Pmax = PMAX_DEFAULT;
float comp = 50.0f;
float res = 5.0f;

const int SEQ_PASSOS[4][4] = {
    {HIGH, LOW, HIGH, LOW},
    {LOW, HIGH, HIGH, LOW},
    {LOW, HIGH, LOW, HIGH},
    {HIGH, LOW, LOW, HIGH}
};

int passoAtual = 0;
int posicaoMotor = 0;
int posicaoAlvo = 0;
int velIns = 0;
int velExp = 0;

unsigned long tCicloInicio = 0;
```

```

unsigned long ultimaPrint = 0;
float PpicoAtual = 0.0f;
float VtEstimado = 0.0f;

float PestRel = 0.0f;

float piU = 0.0f; // saida do PI u[k]
float piEprev = 0.0f; // erro anterior e[k-1]

bool modoPlotter = false;

// ===== MOTOR =====

void darPasso(int dir) {
    passoAtual = (passoAtual + dir + 4) % 4;
    digitalWrite(PIN_IN1, SEQ_PASSOS[passoAtual][0]);
    digitalWrite(PIN_IN2, SEQ_PASSOS[passoAtual][1]);
    digitalWrite(PIN_IN3, SEQ_PASSOS[passoAtual][2]);
    digitalWrite(PIN_IN4, SEQ_PASSOS[passoAtual][3]);
}

void pararMotor() {
    digitalWrite(PIN_IN1, LOW);
    digitalWrite(PIN_IN2, LOW);
    digitalWrite(PIN_IN3, LOW);
    digitalWrite(PIN_IN4, LOW);
}

void moverMotor() {
    if (estado == ST_EMERGENCIA) return;

    int delta = posicaoAlvo - posicaoMotor;
    if (delta > 0) {
        int v = (velIns > 0) ? velIns : VEL_MIN;
        int n = min(delta, v);
        for (int i = 0; i < n; i++) {
            darPasso(1);
            posicaoMotor++;
            if (i < n - 1) delayMicroseconds(DELAY_PASSO_US);
        }
    }
}

```

```

    }
} else if (delta < 0) {
    int v = (velExp > 0) ? velExp : VEL_MIN;
    int n = min(-delta, v);
    for (int i = 0; i < n; i++) {
        darPasso(-1);
        posicaoMotor--;
        if (i < n - 1) delayMicroseconds(DELAY_PASSO_US);
    }
    if (posicaoMotor <= 0) {
        posicaoMotor = 0; pararMotor();
    }
}
}

void retornarMotorZero() {
    while (posicaoMotor > 0 && estado != ST_EMERGENCIA) {
        darPasso(-1);
        posicaoMotor--;
        delayMicroseconds(DELAY_PASSO_US);
    }
    posicaoMotor = 0;
    pararMotor();
}

// ===== EMERGENCIA =====

void ativarEmergencia() {
    estado = ST_EMERGENCIA;
    pararMotor();
    posicaoAlvo = 0;
    PestRel = 0.0f;
    piU = 0.0f; piEprev = 0.0f;

    digitalWrite(PIN_LED_VERMELHO, HIGH);
    digitalWrite(PIN_LED_VERDE, LOW);
    digitalWrite(PIN_LED_AMARELO, LOW);

    Serial.println();
}

```

```

Serial.println(F("!!! EMERGENCIA ACIONADA !!!"));
Serial.println(F("Motor parado. Digite 'reset' para liberar."));
}

// ===== POSICAO E PRESSAO =====

int calcularPosicaoVolume() {
    float frac = volumeAlvo / VT_MAX;
    if (frac > 1.0f) frac = 1.0f;
    return (int)(frac * PASSOS_MAX);
}

int calcularVel(int passos, float tempo) {
    if (tempo < DT_S) tempo = DT_S;
    if (passos <= 0) return VEL_MIN;
    float iteracoes = tempo / DT_S;
    int vel = (int)ceil((float)passos / iteracoes);
    if (vel < VEL_MIN) vel = VEL_MIN;
    if (vel > VEL_MAX) vel = VEL_MAX;
    return vel;
}

void atualizarPressaoEstimada() {
    float volEntregue =
        ((float)posicaoMotor / PASSOS_MAX) * VT_MAX;
    float PestatRel = volEntregue / comp;

    float tau = res * comp / 1000.0f;
    if (tau < 0.01f) tau = 0.01f;
    PestRel += (DT_S / tau) * (PestatRel - PestRel);
    if (PestRel < 0.0f) PestRel = 0.0f;
}

float pressaoMedida() {
    return PEEP + PestRel;
}

// ===== CONTROLADOR PI =====

```

```

float calcularPref() {
    return volumeAlvo / comp + PEEP;
}

int pressaoParaPosicao(float pressao) {
    float pRel = pressao - PEEP;
    if (pRel < 0.0f) pRel = 0.0f;
    float vol = pRel * comp;
    float frac = vol / VT_MAX;
    if (frac > 1.0f) frac = 1.0f;
    return (int)(frac * PASSOS_MAX);
}

float executarPI(float Pref, float Pmed) {
    float e = Pref - Pmed;
    float uNovo = piU + PI_A0 * e + PI_A1 * piEprev;
    float uSat = uNovo;
    if (uSat > Pmax) uSat = Pmax;
    if (uSat < 0.0f) uSat = 0.0f;
    // Anti-windup por clamping
    if (uNovo == uSat) {
        piU = uNovo;
    } else if ((uNovo > Pmax && e < 0.0f)
        || (uNovo < 0.0f && e > 0.0f)) {
        piU = uNovo;
    }
    piEprev = e;
    return uSat;
}

void resetarPI() {
    piU = 0.0f; piEprev = 0.0f;
}

// ===== SETUP =====

void setup() {
    Serial.begin(9600);
    Serial.setTimeout(100);
}

```

```

pinMode(PIN_IN1, OUTPUT); pinMode(PIN_IN2, OUTPUT);
pinMode(PIN_IN3, OUTPUT); pinMode(PIN_IN4, OUTPUT);
pinMode(PIN_LED_VERDE, OUTPUT);
pinMode(PIN_LED_AMARELO, OUTPUT);
pinMode(PIN_LED_VERMELHO, OUTPUT);
pinMode(PIN_BUZZER, OUTPUT);
pararMotor();

Serial.println(F("=== BVM ==="));
Serial.println(F("Comandos: 'start', 'stop', 'reset', '1'"));
Serial.println(F("Presets: 'saudavel', 'dpoc', 'sdra'"));
Serial.println(F("Params: 'fr=', 'vt=', 'ie=', 'pausa=',"));
Serial.println(F("          'comp=', 'res='"));
}

// ===== LOOP PRINCIPAL =====

void loop() {
  if (estado == ST_EMERGENCIA) {
    pararMotor();
    static unsigned long tBuzz = 0;
    if (millis() - tBuzz > 1000) {
      tone(PIN_BUZZER, 2000, 500);
      tBuzz = millis();
    }
    processarSerial();
    delay(50);
    return;
  }

  processarSerial();

  if (estado == ST_PARADO) {
    pararMotor();
    delay(50);
    return;
  }

  float Tciclo = 60.0f / freqResp;

```

```

float Tins    = Tciclo / (1.0f + relacaoIE);
float Texp    = Tciclo - Tins;
float TinsAt  = Tins - pausaInsp;
if (TinsAt < 0.1f) TinsAt = 0.1f;
if (Texp < 0.1f) Texp = 0.1f;

float tSeg = (millis() - tCicloInicio) / 1000.0f;

switch (estado) {
  case ST_INSPIRACAO:
    if (tSeg >= TinsAt) {
      if (pausaInsp > 0.001f) {
        estado = ST_PAUSA_INSP;
      } else {
        estado = ST_EXPIRACAO;
        PpicoAtual = pressaoMedida();
        VtEstimado =
          ((float)posicaoMotor / PASSOS_MAX) * VT_MAX;
        checarVolume();
        velExp = calcularVel(posicaoMotor, Texp);
      }
    } else {
      // --- Controlador PI em malha fechada ---
      float Pref = calcularPref();
      float Pmed = pressaoMedida();
      float uPI  = executarPI(Pref, Pmed);
      posicaoAlvo = pressaoParaPosicao(uPI);
      velIns = calcularVel(posicaoAlvo, TinsAt);
    }
    break;

  case ST_PAUSA_INSP:
    {
      float Pref = calcularPref();
      float Pmed = pressaoMedida();
      float uPI  = executarPI(Pref, Pmed);
      posicaoAlvo = pressaoParaPosicao(uPI);
    }
    if (tSeg >= Tins) {

```

```

    estado = ST_EXPIRACAO;
    PpicoAtual = pressaoMedida();
    VtEstimado =
        ((float)posicaoMotor / PASSOS_MAX) * VT_MAX;
    checarVolume();
    velExp = calcularVel(posicaoMotor, Texp);
}
break;

case ST_EXPIRACAO:
    posicaoAlvo = 0;
    if (tSeg >= Tciclo && posicaoMotor == 0) {
        estado = ST_INSPIRACAO;
        tCicloInicio = millis();
        PpicoAtual = 0.0f;
        VtEstimado = 0.0f;
        PestRel = 0.0f;
        resetarPI();
        pararMotor();
        digitalWrite(PIN_LED_VERDE,
            !digitalRead(PIN_LED_VERDE));
    }
    break;

default:
    break;
}

atualizarPressaoEstimada();
checarSobrepresao();
moverMotor();
imprimirCiclo();

static unsigned long tIter = 0;
unsigned long elapsed = millis() - tIter;
if (elapsed < DT_MS) delay(DT_MS - elapsed);
tIter = millis();
}

```

# APÊNDICE B – Módulo de Parser de Comandos Serial

O código-fonte a seguir implementa o parser de comandos serial para ajuste de parâmetros ventilatórios em tempo real via Monitor Serial da Arduino IDE.

```
// =====
// Modulo de Parser de Comandos Serial
// Formato: PARAM=VALOR (ex: FR=12, VT=500, IE=1:2)
// =====

bool converterFloat(const String& txt, float& val) {
    char buf[24];
    txt.toCharArray(buf, sizeof(buf));
    char* fim;
    val = strtod(buf, &fim);
    while (*fim == ' ' || *fim == '\t'
           || *fim == '\r' || *fim == '\n') fim++;
    return (*fim == '\0' && fim != buf);
}

void processarSerial() {
    if (!Serial.available()) return;

    String cmd = Serial.readStringUntil('\n');
    cmd.trim();
    if (cmd.length() == 0) return;

    if (cmd == "1") { ativarEmergencia(); return; }

    String cmdUp = cmd;
    cmdUp.toUpperCase();

    if (cmdUp == "RESET") {
        if (estado == ST_EMERGENCIA) {
            estado = ST_PARADO;
        }
    }
}
```

```

    pararMotor();
    posicaoMotor = 0;
    posicaoAlvo = 0;
    PestRel = 0.0f;
    digitalWrite(PIN_LED_VERMELHO, LOW);
    noTone(PIN_BUZZER);
    Serial.println(F("[OK] Emergencia liberada.));
}
return;
}

if (estado == ST_EMERGENCIA) {
    Serial.println(F("[BLOQUEADO] Em emergencia.));
    return;
}

if (cmdUp == "START") {
    if (estado == ST_PARADO) {
        estado = ST_INSPIRACAO;
        tCicloInicio = millis();
        PestRel = 0.0f;
        posicaoMotor = 0;
        resetarPI();
        Serial.println(F("[OK] Ventilacao iniciada.));
    }
    return;
}

if (cmdUp == "STOP") {
    estado = ST_PARADO;
    posicaoAlvo = posicaoMotor;
    pararMotor();
    Serial.println(F("[OK] Ventilacao parada.));
    return;
}

// Presets clinicos
if (cmdUp == "SAUDAVEL" || cmdUp == "NORMAL") {
    comp = 50.0f; res = 5.0f; freqResp = 15.0f;
    relacaoIE = 2.0f; volumeAlvo = 500.0f;
}

```

```

Serial.println(F("[OK] SAUDAVEL: C=50 R=5 FR=15"
                " IE=1:2 VT=500"));
return;
}
if (cmdUp == "DPOC") {
    comp = 50.0f; res = 20.0f; freqResp = 12.0f;
    relacaoIE = 3.0f; volumeAlvo = 500.0f;
    Serial.println(F("[OK] DPOC: C=50 R=20 FR=12"
                    " IE=1:3 VT=500"));
    return;
}
if (cmdUp == "SDRA") {
    comp = 20.0f; res = 5.0f; freqResp = 15.0f;
    relacaoIE = 2.0f; volumeAlvo = 400.0f;
    Serial.println(F("[OK] SDRA: C=20 R=5 FR=15"
                    " IE=1:2 VT=400"));
    return;
}
if (cmdUp == "STATUS") {
    imprimirStatus();
    return;
}

// PARAM=VALOR
int sep = cmd.indexOf('=');
if (sep == -1) {
    Serial.println(F("[ERRO] Use PARAM=VALOR"));
    return;
}

String p = cmd.substring(0, sep);
String v = cmd.substring(sep + 1);
p.toUpperCase();
float val;

if (p == "FR") {
    if (!converterFloat(v, val) || val < 8 || val > 30)
        { Serial.println(F("[ERRO] FR: 8-30")); return; }
    freqResp = val;
}

```

```

} else if (p == "VT") {
    if (!converterFloat(v, val) || val < 200 || val > 800)
        { Serial.println(F("[ERRO] VT: 200-800")); return; }
    volumeAlvo = val;
} else if (p == "IE") {
    int dp = v.indexOf(':');
    if (dp == -1) {
        Serial.println(F("[ERRO] Use IE=1:2")); return;
    }
    if (!converterFloat(v.substring(dp + 1), val)
        || val < 1 || val > 4)
        { Serial.println(F("[ERRO] IE: 1:1 a 1:4")); return; }
    relacaoIE = val;
} else if (p == "PAUSA") {
    if (!converterFloat(v, val) || val < 0 || val > 1)
        { Serial.println(F("[ERRO] PAUSA: 0-1")); return; }
    pausaInsp = val;
} else if (p == "COMP") {
    if (!converterFloat(v, val) || val < 10 || val > 100)
        { Serial.println(F("[ERRO] COMP: 10-100")); return; }
    comp = val;
} else if (p == "RES") {
    if (!converterFloat(v, val) || val < 1 || val > 50)
        { Serial.println(F("[ERRO] RES: 1-50")); return; }
    res = val;
} else {
    Serial.print(F("[ERRO] Param: "));
    Serial.println(p);
    return;
}

Serial.print(F("[OK] ")); Serial.print(p);
Serial.print('='); Serial.println(v);

// Validacao cruzada
float Tc = 60.0f / freqResp;
float Ti = Tc / (1.0f + relacaoIE);
float Te = Tc - Ti;
if ((Ti - pausaInsp) < 0.3f)

```

```
    Serial.println(F("[AVISO] Tins ativo < 0.3s"));
float tau = comp * res / 1000.0f;
if (Te < 3.0f * tau)
    Serial.println(F("[AVISO] Texp < 3*tau"));
if ((volumeAlvo / comp + PEEP) > Pmax)
    Serial.println(F("[AVISO] Ppico > Pmax"));
}
```

## APÊNDICE C – Módulo de Alarmes

O código-fonte a seguir implementa o sistema de alarmes com sinalização por prioridade.

```
// =====
// Modulo de Alarmes
// =====

void checarSobrepresao() {
    float Pmed = pressaoMedida();
    static bool alarmeAtivo = false;

    if (Pmed > Pmax && !alarmeAtivo) {
        alarmeAtivo = true;
        digitalWrite(PIN_LED_VERMELHO, HIGH);
        tone(PIN_BUZZER, 2000, 500);
        Serial.print(F("[ALARME] SOBREPRESSAO: "));
        Serial.println(Pmed, 1);
    } else if (alarmeAtivo
                && Pmed < (Pmax - HISTERESE_P)) {
        alarmeAtivo = false;
        digitalWrite(PIN_LED_VERMELHO, LOW);
    }
}

void checarVolume() {
    static int cntBaixo = 0;
    if (VtEstimado > 0 && VtEstimado < 200.0f) {
        if (++cntBaixo >= 3) {
            Serial.print(F("[ALARME] VOL_INSUF: "));
            Serial.println(VtEstimado, 0);
            digitalWrite(PIN_LED_AMARELO, HIGH);
            tone(PIN_BUZZER, 1000, 200);
            cntBaixo = 0;
        }
    } else { cntBaixo = 0; }
```

```
if (VtEstimado > 800.0f) {  
    Serial.print(F("[ALARME] VOL_EXCESS: "));  
    Serial.println(VtEstimado, 0);  
    digitalWrite(PIN_LED_AMARELO, HIGH);  
} else {  
    digitalWrite(PIN_LED_AMARELO, LOW);  
}  
}
```