

doi <https://doi.org/10.18265/2447-9187a2022id7858>
ARTIGO ORIGINAL

SUBMETIDO 14/08/2023

APROVADO 08/12/2023

PUBLICADO ON-LINE 28/12/2023

VERSÃO FINAL DIAGRAMADA 08/04/2025

EDITOR ASSOCIADO

Prof. Dr. Koje Daniel Vasconcelos Mishina

Proposta de uma biblioteca em Modelica para simulação de circuitos pneumáticos

 Tales Matheus Alves dos Passos ^[1] *

 Luciana Lima Monteiro ^[2]

 José Ângelo Peixoto da Costa ^[3]

 Gustavo Koury Costa ^[4]

[1] tmap1@discente.ifpe.edu.br

[2] lucianamonteiro@recife.ifpe.edu.br

[3] angelocosta@recife.ifpe.edu.br

[4] gustavokoury@recife.ifpe.edu.br

Instituto Federal de Pernambuco (IFPE), Recife, Pernambuco, Brasil

* Autor para correspondência.

RESUMO: Modelica é uma linguagem de programação orientada a objetos (POO) introduzida em 1996, que permite a reutilização de elementos programados individualmente. Devido a essa característica, tornou-se uma linguagem ideal para modelagem de vários sistemas físicos e matemáticos, incluindo campos da Engenharia como hidráulica, eletrônica e química. O software OMedit utiliza a linguagem Modelica, mas ainda carece de uma biblioteca pneumática. Portanto, este trabalho propõe a criação de uma biblioteca computacional para elementos pneumáticos, bem como a validação de vários modelos matemáticos utilizados. Foi realizada uma revisão da literatura para identificar as principais equações que regem os componentes pneumáticos. O comportamento do fluxo de massa através de válvulas solenoides e a dinâmica das pressões em câmaras de atuadores pneumáticos ao longo do tempo foram discutidos. Após cada um dos elementos ser modelado computacionalmente no software, experimentos de laboratório foram realizados para a aquisição de dados empíricos a fim de entender o comportamento dos componentes e obter os parâmetros que serão utilizados nos modelos computacionais. Uma placa de aquisição de dados e transdutores de pressão e posição foram usados para adquirir o comportamento dos componentes ao longo do tempo. O processamento de dados foi necessário, incluindo a eliminação de leituras de ruídos devido aos instrumentos de medição usados, e aproximações polinomiais foram desenvolvidas para aplicar métodos matemáticos de minimização de erros. Finalmente, curvas características da pressurização de um reservatório e do movimento de avanço e retorno de um atuador pneumático foram simuladas com sucesso, com as curvas experimentais e teóricas comparadas para destacar o grau de precisão obtido. Como resultado, uma ferramenta para a simulação computacional de um atuador pneumático usando a linguagem Modelica foi desenvolvida, juntamente com uma proposta de uma biblioteca pneumática para OMedit.

Palavras-chave: modelagem; Modelica; circuitos pneumáticos; POO; simulação.

Proposal of a library in Modelica for pneumatic circuits simulation

ABSTRACT: Modelica is an object-oriented programming (OOP) language introduced in 1996, which allows the reuse of individually programmed



elements. Due to this characteristic, it has become an ideal language for modeling various physical and mathematical systems, including fields of Engineering such as hydraulics, electronics and chemistry. The OMedit software uses the Modelica language but still lacks a pneumatic library. Therefore, this work proposes the creation of a computational library for pneumatic elements, as well as the validation of several mathematical models used. A literature review was conducted to identify the main equations governing pneumatic components. The behavior of mass flow through solenoid valves and the dynamics of pressures in pneumatic actuator chambers over time were discussed. After each of the elements was computationally modeled in the software, laboratory experiments were performed to collect empirical data to understand the behavior of the components and obtain the parameters that will be used in the computational models. A data acquisition board and pressure and position transducers were used to capture the components' behavior over time. Data processing was required, including eliminating noise readings due to the measuring instruments used, and polynomial approximations were developed to apply mathematical methods of error minimization. Finally, characteristic curves of the pressurization of a reservoir and the forward and return movement of a pneumatic actuator were successfully simulated, with the experimental and theoretical curves compared to highlight the degree of accuracy obtained. As a result, a tool for the computational simulation of a pneumatic actuator using the Modelica language was developed, along with a proposal for a pneumatic library for OMedit.

Keywords: Modelica; modeling; OOP; pneumatics; simulation.

1 Introdução

Nas últimas décadas, inúmeros aplicativos voltados à modelagem de sistemas físicos foram desenvolvidos. Entre eles, destacam-se o FluidSim, voltado para sistemas hidráulicos e pneumáticos, e o PipeFlow, utilizado em projetos de tubulações industriais. No entanto, observa-se que esses aplicativos foram projetados para resolver problemas específicos, como circuitos elétricos, pneumáticos, hidráulicos ou sistemas químicos, sendo sua aplicabilidade restrita em múltiplas áreas das Ciências e da Engenharia. Com o intuito de integrar essas áreas em uma única plataforma, Hilding Elmvist apresentou, em 1996, a linguagem Modelica (Tiller, 2001). Essa nova linguagem incorporou duas características fundamentais: (a) ser de alto nível, simplificando o uso em relação à linguagem de máquina; (b) possibilitar a integração de várias áreas da Engenharia.

Embora plataformas comerciais para a modelagem de sistemas pneumáticos estejam disponíveis no mercado há algum tempo, o custo elevado dessas ferramentas pode dificultar seu uso no meio acadêmico. Além disso, essas plataformas não permitem a inserção de outros modelos matemáticos para o cálculo de suas variáveis, o que torna seus códigos inflexíveis. Nesse contexto, a linguagem Modelica se apresenta como uma ferramenta relevante para atender a essa demanda. Neste artigo, utiliza-se uma plataforma de simulação e modelagem baseada em Modelica, denominada OpenModelica¹, que já vem equipada com mais de 50 bibliotecas de diferentes áreas da Engenharia e da Modelagem, mas ainda carece de uma biblioteca específica para sistemas pneumáticos. Assim, o objetivo deste trabalho é a criação de uma biblioteca preliminar para sistemas pneumáticos no OpenModelica. Alguns componentes são modelados, seguidos de uma comparação experimental para validação do código computacional. Inicialmente, serão

[1] Disponível em: <https://openmodelica.org>. Acesso em: 22 dez. 2023.

modelados apenas os elementos básicos para sistemas pneumáticos: compressor, escape, cilindro, reservatório e válvulas direcionais 5×2 e 3×2 acionadas por solenoide com retorno por mola.

O restante deste artigo está organizado da seguinte forma: na seção 2, são apresentados aspectos gerais da linguagem Modelica e do ambiente de simulação; na seção 3, é descrita a criação da biblioteca básica e de seus componentes, além da modelagem e da simulação computacional de dois circuitos pneumáticos distintos; na seção 4, são apresentados os testes experimentais realizados em bancada, para obtenção de resultados e comparação com a teoria; por fim, na seção 5, são apresentadas as conclusões deste trabalho.

2 Aspectos gerais da linguagem Modelica

Tinnerholm, Pop e Sjölund (2022) e Buffoni *et al.* (2021) afirmam que Modelica é uma linguagem de programação declarativa, orientada a objetos, baseada em equações, que permite a modelagem de sistemas de Engenharia com alto grau de complexidade e modelos físicos híbridos (como, por exemplo, sistemas elétricos e químicos). Por sua vez, o OpenModelica é um ambiente integrado de simulação e modelagem na linguagem Modelica, onde é possível escrever, depurar e analisar programas nessa linguagem. Além disso, o OpenModelica oferece suporte para simulação e modelagem, incluindo a interface gráfica OMedit.

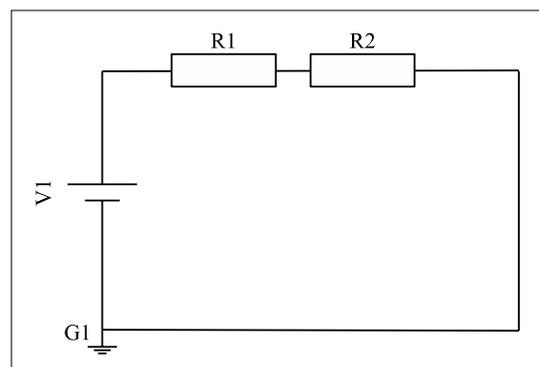
A Programação Orientada a Objetos (POO) baseia-se na ideia de que suas estruturas funcionam como elementos físicos (“objetos”). Cada objeto, por sua vez, pertence a uma determinada “classe”. Esse tipo de programação permite a modelagem individual de cada componente, bem como a reutilização de várias classes e objetos para diferentes sistemas. Por exemplo, um sistema composto por um ônibus, um carro e uma motocicleta pode ser representado como um sistema com três objetos pertencentes a uma classe mais geral denominada, por exemplo, “veículos terrestres”. Fiorina *et al.* (2022) enfatizam que, pelo fato de os objetos serem independentes e terem seu funcionamento compartimentalizado, a alteração local do funcionamento de um deles não impacta os demais, diminuindo possíveis erros e aumentando a velocidade na programação, com a possibilidade de reutilização.

Dessa maneira, dois termos são essenciais para a compreensão da POO: classe e objeto. A classe corresponde à essência dos objetos, tipifica o seu comportamento e, normalmente, é algo mais abstrato. O objeto, por outro lado, é a concretização da classe. Em linguagem POO, objetos são referenciados como “instâncias” de uma determinada classe.

A Figura 1 ilustra um circuito elétrico que pode ser modelado utilizando a linguagem Modelica, com uma das bibliotecas disponibilizadas no OpenModelica. O código-fonte em Modelica referente ao circuito é mostrado no Código 1.

Figura 1 ▶

Fonte de tensão constante V1 com aterramento G1, conectada em série a dois resistores, R1 e R2.
Fonte: elaborado pelos autores



Código 1 ▶

Circuito elétrico da Figura 1 em Modelica.

Fonte: elaborado pelos autores

```

model CircuitoEletrico
  Modelica.Electrical.Analog.Basic.Resistor R1 annotation( ...);
  Modelica.Electrical.Analog.Basic.Resistor R2 annotation( ...);
  Modelica.Electrical.Analog.Basic.Ground G1 annotation( ...);
  Modelica.Electrical.Analog.Sources.ConstantVoltage V1 annotation( ...);
equation
  connect(R1.n, R2.p);
  connect(G1.p, V1.n);
  connect(V1.p, R1.p);
  connect(R2.n, V1.n);
  annotation( ...);
end CircuitoEletrico;
    
```

No circuito da Figura 1, foram utilizados objetos pertencentes a três classes diferentes: a fonte de tensão V1; os resistores R1 e R2; e o aterramento G1. Com base no Código 1, as instâncias da classe “Modelica.Electrical.Analog.Basic.Resistor” (disponível no código “Resistor” do Apêndice) foram denominadas R1 e R2. Observa-se que os terminais de cada objeto foram conectados por meio do método “connect”. Isso pode ser observado na linha “connect (R1.n, R2.p)”, em que dois terminais de polaridade diferente (*p* e *n*) são interconectados. Além da simplicidade de programação, o ambiente OpenModelica também oferece a possibilidade de modelagem gráfica, facilitando ainda mais para o programador a manipulação dos objetos, como será apresentado ao longo deste artigo.

A classe mais simples em Modelica, que precisa estar presente em cada simulação, é a classe “connector”, que é referenciada como um “método”. Na POO, métodos são procedimentos executados pelo compilador. Dessa forma, a classe “connector” é utilizada para estabelecer conexões entre os objetos do circuito. Nos conectores, não são utilizadas equações, apenas as variáveis que são transmitidas entre os objetos, que podem ser basicamente de dois tipos: potencial ou fluxo. Um exemplo de variáveis do tipo “potencial” são a tensão elétrica e a pressão, enquanto as variáveis típicas do tipo “fluxo” incluem a corrente elétrica e a vazão. No Apêndice PneumaticConnector, é apresentado o código criado para um conector pneumático, em que se observa que, para diferenciar a vazão mássica (*m*) da pressão (*p*), utiliza-se a palavra “flow”, indicando uma variável do tipo fluxo (as variáveis do tipo potencial são assumidas por padrão pelo compilador, sem necessidade de um prefixo).

Além da classe “connector”, outras duas classes merecem destaque em Modelica:

- A classe “package”, que corresponde a uma biblioteca, onde são alocados os diversos modelos;
- A classe “model”, que é utilizada para a modelagem dos componentes individuais por meio de equações descritivas.

3 Modelagem dos componentes

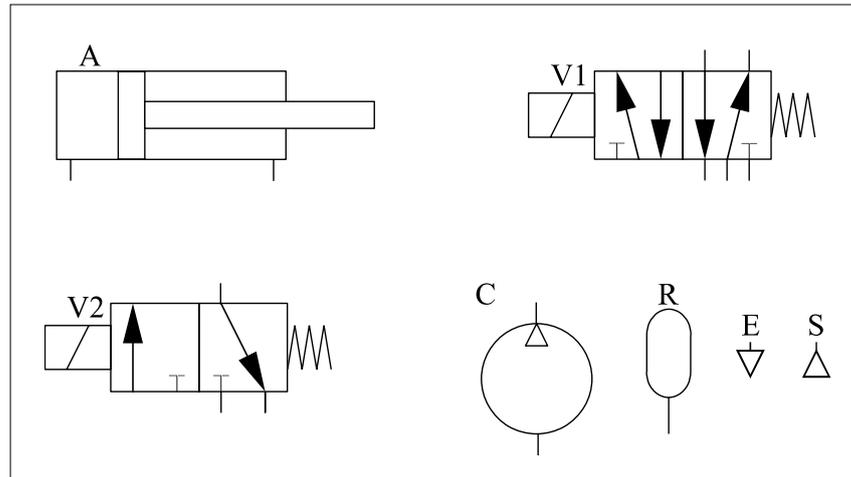
Todos os componentes apresentados na Figura 2 foram criados como parte de uma biblioteca (“package”) denominada “PneumaticPackage”², contendo dois experimentos e uma sub-biblioteca (“package”). Dentro dessa biblioteca, a parte denominada “Pneumatic” está organizada como uma subpasta contendo os componentes, cujos códigos se encontram descritos no Apêndice deste artigo. Cavalcante *et al.* (2019) utilizam três termos para a

[2] Os arquivos encontram-se disponibilizados na página deste artigo no site da Revista Principia.

modelagem de sistemas: caixa branca (baseado em equações fenomenológicas), caixa preta (utilizando-se equações matemáticas exclusivamente de dados obtidos) e caixa cinza (uma combinação de ambas as abordagens). A metodologia neste trabalho assemelha-se à de caixa branca, característica de muitas bibliotecas em Modelica, como observado na análise de Seabra *et al.* (2016) sobre bibliotecas do software Dymola, um ambiente de simulação baseado em Modelica, assim como o OpenModelica. No entanto, alguns autores, como Cavalcante *et al.* (2019), que realizaram experimentos semelhantes aos deste artigo, optaram por uma modelagem de caixa preta.

Figura 2 ►

Elementos desenvolvidos ao longo deste trabalho:
C - compressor,
E - escape, A - cilindro,
R - reservatório, S - fonte de pressão (compressor),
V1 e V2 - válvulas direcionais 5×2 e 3×2 acionadas por solenoide com retorno por mola, respectivamente.
Fonte: elaborado pelos autores



Inicialmente, foi adicionado à biblioteca um elemento da classe “connector” com as variáveis de vazão em massa (m) e de pressão (p). A vazão em massa possui o prefixo “flow”, por ser uma grandeza que atravessa o circuito. Esse elemento foi denominado “PneumaticConnector”, conforme o código disponível no Apêndice de mesmo nome.

Os primeiros elementos adicionados foram o escape e a fonte de pressão. O escape representa graficamente a conexão da rede de ar comprimido com o ar atmosférico. Na programação, esse componente possui apenas um conector, ao qual é atribuída uma pressão constante, igual à pressão atmosférica (101.325 Pa). De modo similar, a fonte de pressão simula uma alimentação de pressão constante em um circuito. Entretanto, nesse caso, em vez de uma pressão atmosférica, o componente recebe um parâmetro denominado “PI”, que contém o valor da pressão de trabalho a ser inserida pelo usuário. Esses dois primeiros modelos foram nomeados “Exhaust” e “PressureSource”, cujos códigos se encontram no Apêndice.

O próximo modelo a ser descrito é o reservatório de ar, um elemento pneumático que, além de atenuar oscilações de pressão ao longo da linha, atua como acumulador de energia. O comportamento da pressão no interior do reservatório pode ser descrito pela Equação dos Gases Perfeitos, demonstrada na Equação 1.

$$P V = \left(\frac{m}{M} \right) R T \quad (1)$$

onde: P é a pressão absoluta do ar no reservatório; V é o volume do reservatório; R é a constante universal dos gases (8,31 J/mol.K); T , a temperatura absoluta do ar no reservatório, considerada constante em 293,15 K; m é a massa de ar no reservatório; e M é a massa molecular do ar, que equivale a 0,02896 .

De acordo com a Lei da Conservação da Massa, a taxa de variação da massa, dm/dt , no interior do reservatório deve ser igual à vazão em massa no conector do terminal. A taxa de variação da massa pode ser obtida a partir da Equação 1, derivando-se ambos os membros em relação ao tempo, resultando na Equação 2.

$$\left(\frac{dP}{dt}\right)V = \frac{dm}{dt}\left(\frac{RT}{M}\right) \quad (2)$$

A representação da Equação 2 em Modelica é apresentada no código “Airtank” do Apêndice. Inicialmente, a variável “ Vm ” é definida como a vazão em massa por meio do conector do reservatório, sendo igualada à variação da massa dentro do reservatório, dm/dt . Em seguida, utiliza-se a função “ $der(p)$ ” para calcular a derivada temporal da pressão, dm/dt . O código resultante é derivado da reescrita da Equação 2, evidenciando-se “ $Vm = dm/dt$ ”, ou seja, “ $Vm = (V \times M) \times der(p) / (k \times T)$ ”.

O próximo componente a ser modelado é o compressor. De maneira geral, compressores são dispositivos pneumáticos responsáveis pela obtenção de ar pressurizado, admitindo determinado volume de ar atmosférico e pressurizando-o até atingir a pressão de trabalho exigida (Simões, 2016). Os compressores dividem-se basicamente em dois tipos:

- Compressores de deslocamento positivo: nestes, o volume é reduzido para aumentar a pressão;
- Compressores de deslocamento dinâmico: nestes, o aumento da pressão é obtido através da conversão de energia cinética em pressão durante a passagem do ar pelo rotor do compressor.

No trabalho de Maia (2005 *apud* Oliveira, 2013), descreve-se a vazão em massa na saída de um compressor. Dada uma pressão p (em Pa) na saída do compressor, a vazão em massa m (em kg/s) pode ser escrita em função: da frequência de rotação do motor elétrico (em Hz); do volume na câmara de compressão V (em m^3); do rendimento volumétrico n ; da pressão máxima de operação P_1 (em Pa); e da massa específica do ar atmosférico d , que é de $1,2041 \text{ kg/m}^3$ à temperatura de $20 \text{ }^\circ\text{C}$. O compressor é desligado quando o reservatório atinge a pressão máxima de operação. Assim, a vazão em massa na saída do compressor é descrita pela Equação 3, disponível em Modelica no código “Compressor” do Apêndice.

$$\begin{cases} \dot{m} = f \times V \times n \times d & \text{para } P < P_1 \\ \dot{m} = 0 & \text{para } P \geq P_1 \end{cases} \quad (3)$$

Passa-se agora à modelagem das válvulas 3×2 , que possuem três terminais (vias) e duas posições, havendo dois tipos de válvulas direcionais. As válvulas simples, ao serem acionadas eletricamente, movimentam um carretel interno que abre e fecha, permitindo a passagem ou o bloqueio do fluido entre as vias. As válvulas proporcionais possuem a mesma funcionalidade, porém com o adicional de controle fino dessa abertura, em função da tensão elétrica no solenoide. “Para fins de modelagem, assume-se normalmente que a vazão em massa em uma válvula proporcional se aproxima àquela de um orifício montado em uma linha [...]” (Bobrow; McDonell, 1998, p. 734, tradução nossa). Endler *et al.* (2008), Mandali e Dong (2022) e Laib *et al.* (2022) descrevem o comportamento da vazão em massa através de um orifício pela Equação 4.

$$\begin{cases} \dot{m} = A_0 P_u \sqrt{\frac{2\gamma}{(\gamma-1)KT} \left[\left(\frac{P_d}{P_u}\right)^{\frac{2}{\gamma}} - \left(\frac{P_d}{P_u}\right)^{\frac{\gamma+1}{\gamma}} \right]} & \text{se } \frac{P_d}{P_u} > b \\ \dot{m} = A_0 P_u \sqrt{\frac{\gamma}{KT} \left(\frac{2}{\gamma+1}\right)^{\frac{\gamma+1}{\gamma-1}}} & \text{se } \frac{P_d}{P_u} \leq b \end{cases} \quad (4)$$

onde:

- P_u e P_d e são as pressões na entrada e na saída da válvula, respectivamente;
- A_0 é a área de orifício efetivo da válvula, considerada igual a $8,5 \times 10^{-7} \text{ m}^2$, valor estimado a partir de experimentos de laboratório;
- γ é a constante adiabática (para o ar, $\gamma = 1,4$);
- $K = 287 \text{ J/KgK}$ é a constante do ar;
- T é a temperatura absoluta do ar na entrada da válvula, considerada constante e igual a $293,15 \text{ K}$;
- b é a razão entre as pressões na saída (P_d) e na entrada da válvula (P_u) quando a velocidade do ar é máxima, denominada relação crítica de pressão, que pode ser calculada pela Equação 5:

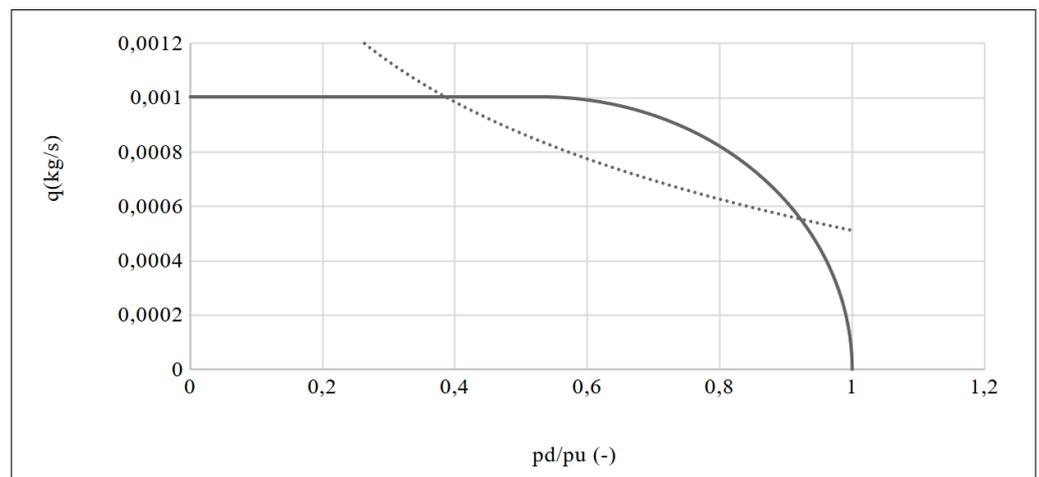
$$b = \left(\frac{2}{\gamma+1}\right)^{\frac{\gamma}{\gamma-1}} \quad (5)$$

Substituindo-se $\gamma = 1,4$ na Equação 5, obtém-se $b = 0,528$. Demonstra-se que, quando a relação entre as pressões de saída e de entrada da válvula P_d/P_u for menor que b , a vazão em massa será constante e atingirá um valor máximo. Após atingir a relação de pressão crítica, a vazão diminuirá gradativamente, conforme ilustrado na Figura 3.

Figura 3 ►

Variação da vazão através de um orifício em função da relação entre as pressões de entrada e de saída da válvula.

Fonte: dados da pesquisa



Bobrow e McDonell (1998) comentam que as Equações 4 são de baixa assertividade e incorretamente assumidas como válidas. Para contornar essa limitação, os autores propõem a Equação 6, revisada por outros pesquisadores, como Costa (2023), para o caso da vazão de enchimento e esvaziamento de um reservatório pressurizado com ar.

$$\begin{cases} \dot{m}_{ench} = B_{ench} \sqrt{P_u - P_d} & \text{no enchimento} \\ \dot{m}_{esv} = B_{esv} (P_u - P_d) & \text{no esvaziamento} \end{cases} \quad (6)$$

onde: \dot{m}_{ench} é a vazão em massa durante o enchimento; \dot{m}_{esv} é a vazão em massa durante o esvaziamento; B_{ench} e B_{esv} são constantes obtidas por meio da minimização do erro a partir de dados experimentais, sendo iguais a $2,577 \times 10^{-6}$ e $8,387 \times 10^{-9}$, respectivamente.

As Equações 4 e 6 foram utilizadas para modelar as válvulas direcionais 3×2 e 5×2 vias, nomeadas “ThreeXTwoA” (Equações 4), “ThreeXTwoB” e “FiveXTwoB” (Equações 6) nos códigos em Modelica disponíveis no Apêndice. Como o modelo das Equações 4 foi descartado após os testes com a válvula direcional 3×2, conforme descrito na próxima seção, a válvula 5×2 que utilizava essas equações não está incluída no Apêndice. A equação mais adequada para a simulação será discutida adiante.

Na modelagem do cilindro pneumático, este é composto por duas câmaras: a câmara no lado do êmbolo (de volume maior) e a do lado da haste (de volume menor). A entrada de ar em uma das câmaras aumenta a pressão no lado correspondente, criando uma diferença de pressão entre as duas câmaras. Essa diferença de pressão, atuando sobre áreas distintas, gera uma força resultante F_p contrária à força de atrito F_{atr} entre o êmbolo e a camisa do cilindro. A Segunda Lei de Newton pode ser aplicada às partes móveis (êmbolo, haste e massa movida), cuja massa total é M . Adotando-se o deslocamento da haste como x e sua aceleração como a , a Segunda Lei de Newton pode ser escrita pela Equação 7:

$$Ma = F_p - F_{atr} \quad (7)$$

A Equação 7 pode ser escrita em função do deslocamento da haste, como na Equação 8:

$$\begin{cases} a = \frac{dv}{dt} \\ v = \frac{dx}{dt} \end{cases} \quad (8)$$

No estudo de Bavaresco (2007), foi utilizado o modelo proposto por Bobrow e McDonell (1998), que resulta no sistema de equações diferenciais apresentado na Equação 9:

$$\begin{cases} \frac{dx}{dt} = v \\ \frac{dv}{dt} = \frac{A_a}{M} P_a - \frac{A_b}{M} P_b - \frac{B}{M} v \\ \frac{dP_a}{dt} = - \left(\frac{A_a \gamma v}{V_{a0} + A_a \gamma} \right) P_a + \left(\frac{K \gamma T}{V_{a0} + A_a \gamma} \right) \dot{m}_a \\ \frac{dP_b}{dt} = \left(\frac{A_b \gamma v}{V_{b0} - A_b \gamma} \right) P_b - \left(\frac{K \gamma T}{V_{b0} - A_b \gamma} \right) \dot{m}_b \end{cases} \quad (9)$$

onde:

- B é o coeficiente de atrito viscoso entre o cilindro e o êmbolo;
- A_a e A_b e são as áreas expostas do cilindro nas câmaras anterior e posterior do êmbolo, respectivamente;
- V_{a0} e V_{b0} são as folgas volumétricas nas câmaras anterior e posterior do êmbolo, respectivamente;
- \dot{m}_a e \dot{m}_b são as vazões em massa nas vias das câmaras anterior e posterior do êmbolo, respectivamente.

Para a temperatura, adotou-se o valor de 293,15 K. O cilindro foi, portanto, modelado de acordo com as Equações 9, sendo denominado “Actuator” no código em Modelica, conforme consta no último item do Apêndice.

4 Resultados experimentais e modelagem numérica

O primeiro experimento realizado foi o proposto e representado na Figura 4, no qual se pressuriza um reservatório através do acionamento de uma eletroválvula 3×2 vias. Inicialmente, foram adquiridos os dados referentes ao reservatório, que foi pressurizado até uma pressão de 4 bar manométricos. Os parâmetros da simulação estão informados na Tabela 1, e os equipamentos utilizados no experimento, na Tabela 2.

Figura 4 ▶

Montagem em bancada do primeiro ensaio experimental:
 (1) Reservatório de ar.
 (2) Transdutor de pressão.
 (3) Manômetro.
 (4) Eletroválvula 3×2 vias.
 (5) Unidade de conservação.
 (6) Válvula redutora de vazão.
 (7) Comando elétrico.
 Fonte: arquivo dos autores

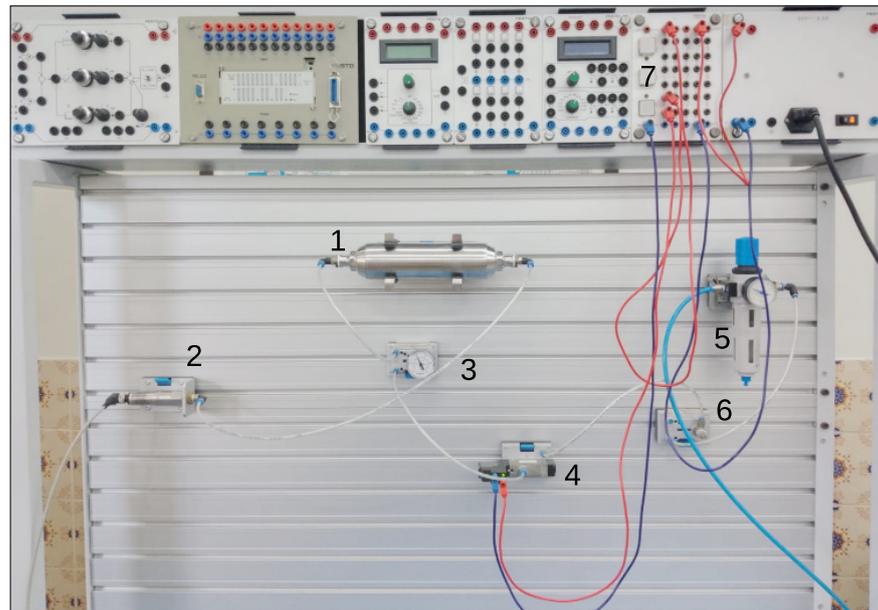


Tabela 1 ▶

Parâmetros utilizados na simulação computacional no primeiro experimento.
 Fonte: dados da pesquisa

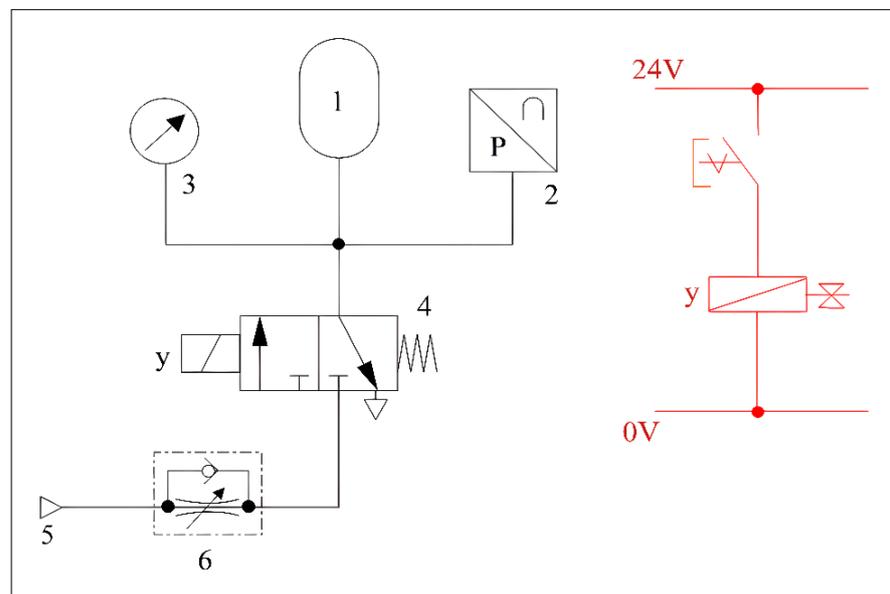
Parâmetro	Valor	Unidade
Pressão utilizada	4	bar
Pressão atmosférica	0	bar
Tempo para o início do sinal	0	s
Volume do reservatório	0,0004	m ³

Tabela 2 ▶
Instrumentação utilizada no primeiro experimento.
Fonte: dados da pesquisa

Equipamento	Identificação
Transdutor de pressão	547 482 SDET-22T-D10-G14-U-M12
Eletroválvula 3×2 vias	FESTO MFH-3-1/8 7802 BN 14
Reservatório de 0,4 litros	FESTO CRVZS-0,4 160234
Válvula redutora de vazão	FESTO 152881
Unidade de conservação	FESTO 159631
Placa de aquisição de dados	NI myDAQ SID 195509F-01L

O objetivo do experimento foi traçar a curva da pressão manométrica dentro do reservatório 1 em função do tempo, a partir da pressão inicial (pressão atmosférica). Para isso, a eletroválvula 4 foi acionada, permitindo a passagem de ar da válvula 6, conectada à unidade de conservação 5, para o interior do reservatório. O aumento da pressão foi acompanhado visualmente pela leitura do manômetro 3. O transdutor de pressão 2 mediu a pressão na entrada do reservatório e enviou um sinal elétrico correspondente (convertendo a medida em bar para um sinal em volt) para um dispositivo de aquisição de dados, conectado a um computador (não mostrado na Figura 4). A Figura 5 ilustra o esquema pneumático correspondente à montagem experimental da Figura 4.

Figura 5 ▶
Esquema eletropneumático correspondente à montagem experimental.
Fonte: elaborado pelos autores

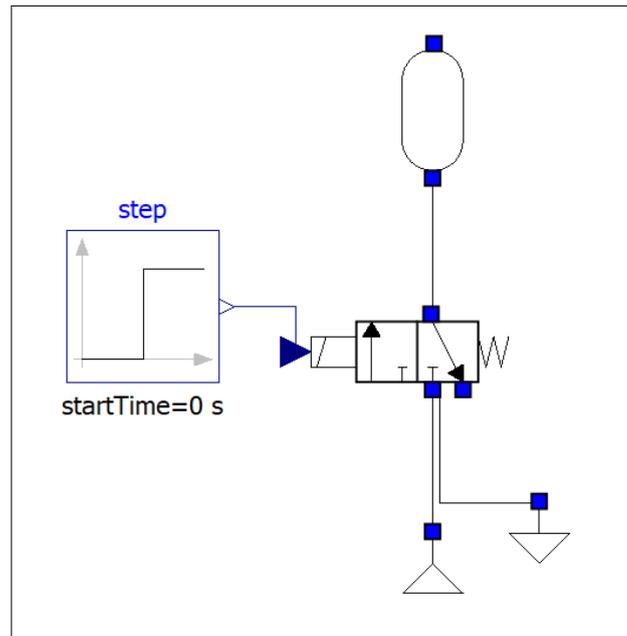


O circuito pneumático foi simulado com o auxílio da biblioteca desenvolvida. Cada elemento da Figura 4 foi representado em OpenModelica, conforme ilustrado na Figura 6: uma fonte de tensão (o sinal *step*), um reservatório (1), uma eletroválvula (4) e uma fonte de pressão (5) e (6).

Figura 6 ▶

Desenho do circuito pneumático em OpenModelica com a biblioteca desenvolvida.

Fonte: elaborado pelos autores

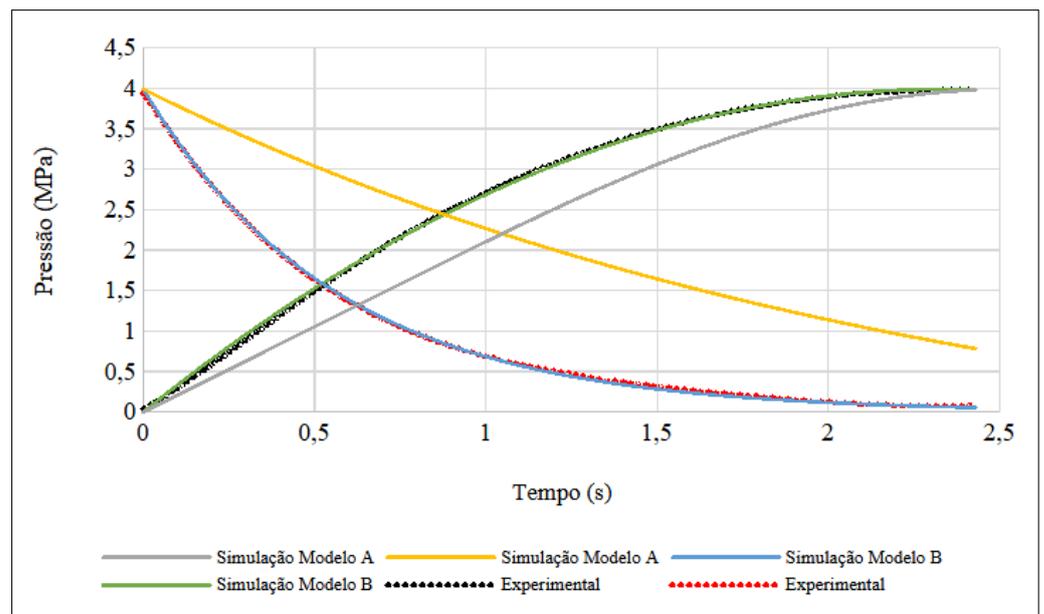


Para cada elemento do circuito na Figura 6, foi criado um ícone em Modelica, utilizando o IconEditor, presente no OMEdit. O circuito elétrico foi substituído por uma função degrau padrão, que, a partir de um instante de tempo, envia um sinal para a abertura da eletroválvula, reproduzindo o procedimento experimental. Detalhes da representação gráfica foram omitidos no código computacional presente no Apêndice, estando contidos nas seções intituladas “*annotations*”. Os resultados experimentais e da simulação computacional estão apresentados na Figura 7, que demonstra uma excelente concordância entre o circuito simulado e o real somente quando o modelo B (Equações 6) foi utilizado para a eletroválvula. Para a simulação numérica, foi utilizado o método de Runge-Kutta (quarta ordem) no OpenModelica.

Figura 7 ▶

Comparação entre os dados experimentais e a simulação em OpenModelica com os dados experimentais expressos em pontilhado.

Fonte: dados da pesquisa



O segundo experimento foi realizado para determinar o deslocamento de um atuador linear ao longo do tempo. As curvas referentes às Equações 4, modelo A, antes

de atingirem a relação de pressão crítica, comportam-se como uma função linear de primeiro grau, e após atingirem esse valor, seu comportamento varia gradativamente, com a pressão variando de forma menos acentuada. O modelo B, por outro lado, mantém o mesmo comportamento ao longo do tempo, formando uma curva contínua do início ao fim da simulação. Nesse caso, como o modelo simplificado para a eletroválvula proposto por Bobrow e McDonell (1998) – Equações 6 – mostrou-se extremamente assertivo no primeiro experimento, foi escolhido para a modelagem da eletroválvula de comando do cilindro no segundo experimento.

A Figura 8 ilustra a montagem experimental na bancada para o segundo experimento. Dessa vez, dois reservatórios de ar idênticos foram colocados entre as vias do cilindro e as vias de utilização da eletroválvula. O objetivo foi simular os volumes internos das redes de ar comprimido, que devem ser considerados em qualquer circuito pneumático, simulando um caso mais complexo para verificar a confiabilidade da biblioteca criada em Modelica. Para a simulação de um atuador sem haste, utilizou-se a mesma programação e ícone do atuador com haste, apenas zerando o parâmetro do diâmetro da haste. Os demais parâmetros utilizados na simulação e a descrição dos equipamentos estão listados nas Tabelas 3 e 4.

Figura 8 ►

Montagem em bancada do segundo ensaio experimental. (1) Cilindro sem haste. (2) Transdutor de posição acoplado ao cilindro. (3)-(4) Reservatórios de ar. (5) Eletroválvula 5x2 vias. (6) Unidade de conservação. (7) Válvula redutora de vazão. (8) Comando elétrico.
Fonte: arquivo dos autores

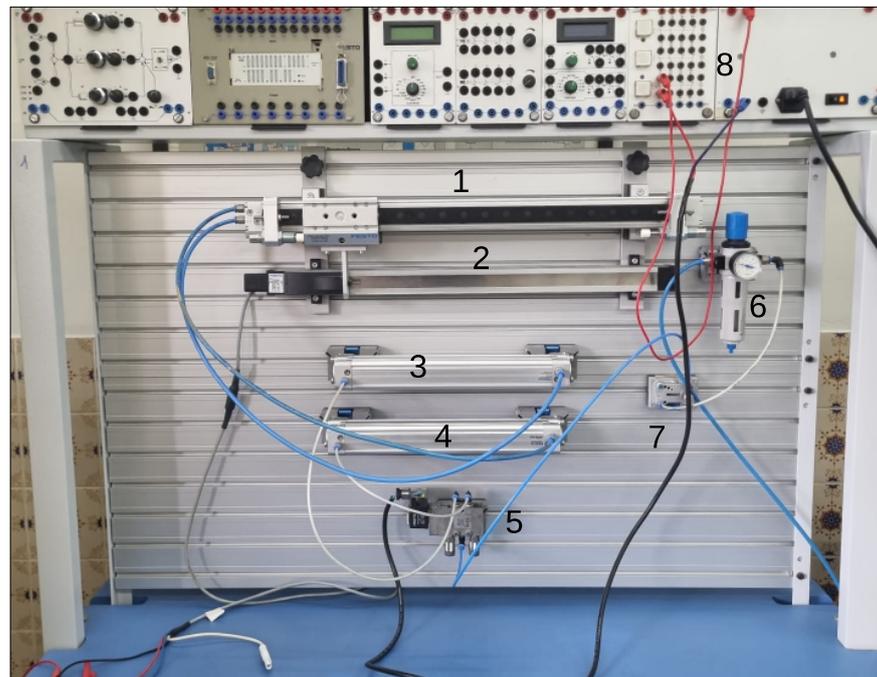


Tabela 3 ►

Parâmetros utilizados na simulação computacional no segundo experimento.
Fonte: dados da pesquisa

Parâmetro	Valor	Unidade
Pressão utilizada	4	Bar
Pressão atmosférica	0	Bar
Diâmetro do êmbolo	0,025	m
Massa inercial	5	kg
Curso útil do atuador	0,45	m
Volumes dos reservatórios	0,0002	m ³
Coefficiente do atrito viscoso	90	Ns/m

Tabela 4 ▶

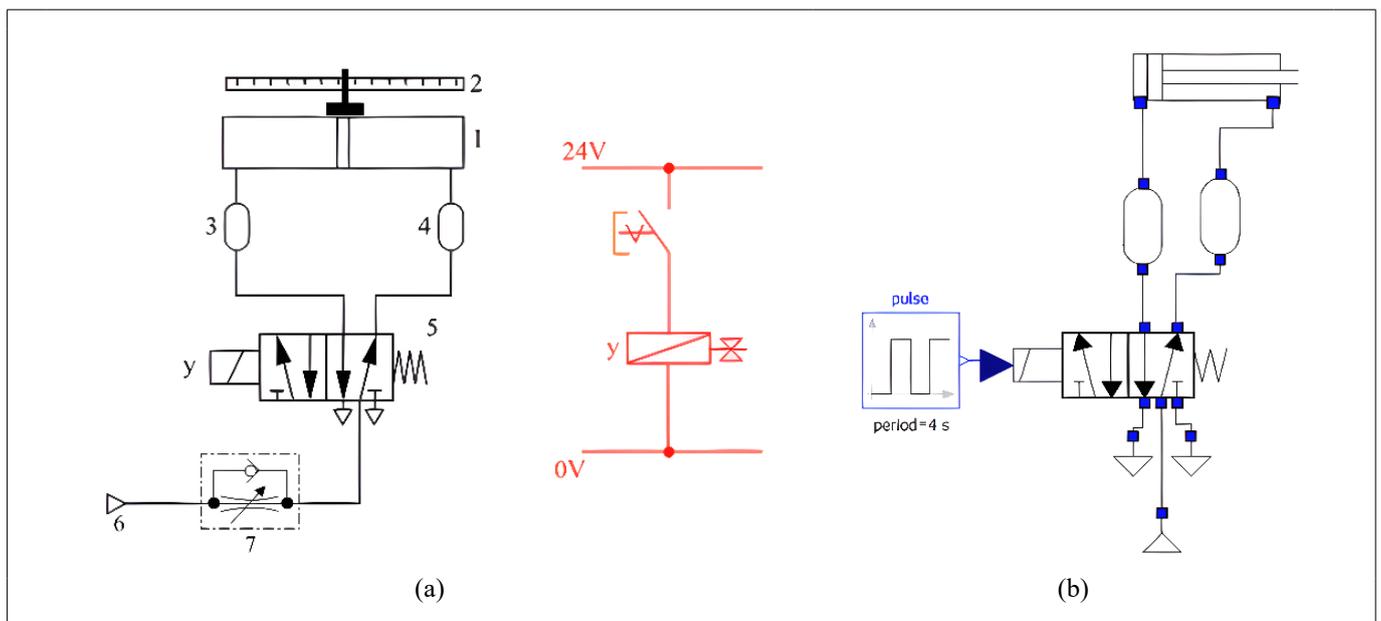
Instrumentação utilizada no segundo experimento.
Fonte: dados da pesquisa

Equipamento	Identificação
Transdutor linear de posição	MLO-POT-TLF
Válvula 5×2 vias	FESTO MFH-5-1/8 9982 X3 14
Atuador pneumático linear	DGPL-25-450-PPV-A-B-KF-GK-SH
Reservatórios de 0,2 L	VZSDNC-32-250-PPV-A
Placa de aquisição de dados	NI myDAQ SID 195509F-01L

Figura 9 ▼

(a) Esquema eletropneumático correspondente à montagem experimental mostrada na Figura 8, enquanto a Figura 9b exibe o desenho equivalente em OpenModelica, utilizando a biblioteca criada. Para simular o avanço e o retorno do cilindro, foi utilizada uma entrada em forma de pulso na eletroválvula, de tal forma que o atuador avance quando a eletroválvula receber um sinal positivo, e retorne quando o sinal for igual a zero. O período do pulso de sinal foi programado de maneira que o atuador percorresse todo o seu curso até o final, retornando automaticamente em seguida (ver detalhes no código “Actuator”, listado no Apêndice).
Fonte: elaborado pelos autores

A Figura 9a apresenta o esquema pneumático correspondente à montagem experimental mostrada na Figura 8, enquanto a Figura 9b exibe o desenho equivalente em OpenModelica, utilizando a biblioteca criada. Para simular o avanço e o retorno do cilindro, foi utilizada uma entrada em forma de pulso na eletroválvula, de tal forma que o atuador avance quando a eletroválvula receber um sinal positivo, e retorne quando o sinal for igual a zero. O período do pulso de sinal foi programado de maneira que o atuador percorresse todo o seu curso até o final, retornando automaticamente em seguida (ver detalhes no código “Actuator”, listado no Apêndice).

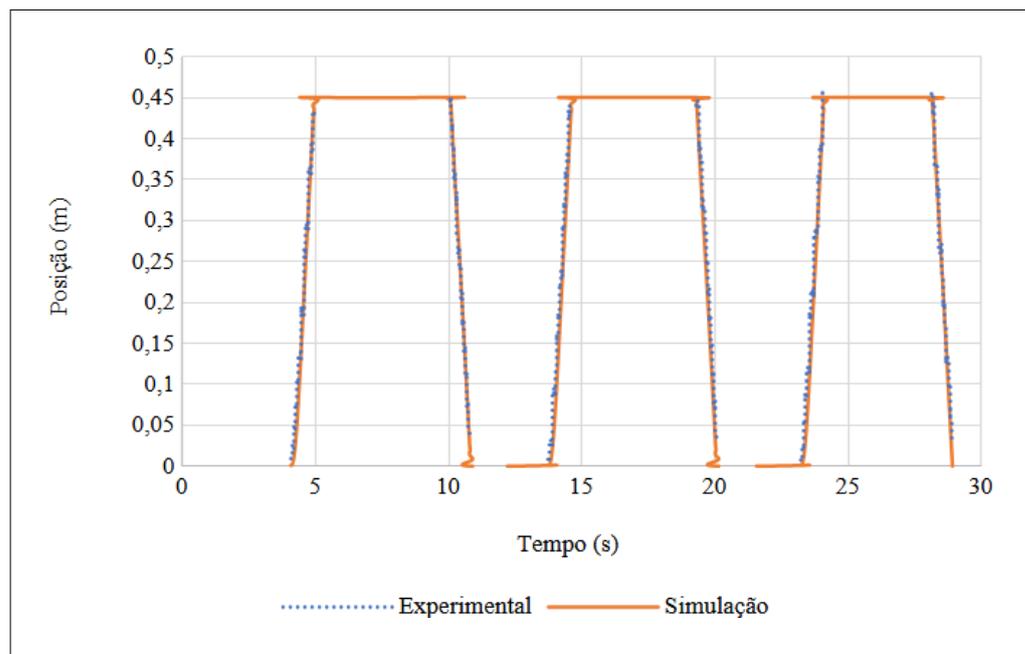


A Figura 10 compara os resultados obtidos experimentalmente com aqueles da simulação em OpenModelica. Como não foi possível produzir a mesma frequência de movimento no atuador, apenas os dados relativos às rampas de avanço e retorno do cilindro foram computados. Utilizando o mesmo solver que no primeiro experimento, observou-se novamente uma excelente concordância entre os resultados experimentais e a simulação computacional.

Figura 10 ►

Avanço e retorno do atuador ao longo do tempo experimental versus simulado.

Fonte: dados da pesquisa



5 Conclusões

Este trabalho apresentou uma proposta de biblioteca pneumática em linguagem Modelica para a simulação de um atuador pneumático, modelando-se cada elemento individualmente. Foram apresentadas as principais características da linguagem escolhida, assim como uma descrição geral de cada componente pneumático desenvolvido e dos experimentos realizados.

Foram discutidos modelos matemáticos que regem as vazões em válvulas pneumáticas e o comportamento dos reservatórios pneumáticos durante o processo de pressurização. Além disso, foi utilizado um modelo matemático de quarta ordem (Equação 9) para descrever o comportamento de um atuador pneumático linear. Também foram obtidos dados experimentais referentes à pressurização e despressurização do reservatório, bem como à posição do atuador ao longo do tempo. Foram gerados gráficos da simulação no programa OMedit para ambos os experimentos e comparados aos dados experimentais. O modelo com maior exatidão em relação aos dados experimentais, correspondendo às Equações 6, foi utilizado na simulação do curso do atuador ao longo do tempo, devido à maior proximidade entre a curva teórica e a experimental.

Dessa forma, os modelos dos elementos pneumáticos foram simulados com êxito e podem ser aplicados no controle de circuitos pneumáticos. Com os elementos desenvolvidos nesta biblioteca, propõe-se a ampliação do seu acervo, como a modelagem de válvulas proporcionais ou a aplicação de modelos matemáticos mais complexos, como o de quinta ordem proposto por Richter (2013).

Financiamento

Esta pesquisa não recebeu financiamento externo.

Conflito de interesses

Os autores declaram não haver conflito de interesses.

Contribuições ao artigo

PASSOS, T. M. A.: concepção ou desenho do estudo/pesquisa; análise e/ou interpretação dos dados; revisão final com participação crítica e intelectual no manuscrito. **MONTEIRO, L. L.; COSTA, J. A. P.:** análise e/ou interpretação dos dados; revisão final com participação crítica e intelectual no manuscrito. **COSTA, G. K.:** concepção ou desenho do estudo/pesquisa; revisão final com participação crítica e intelectual no manuscrito. Todos os autores participaram da escrita, discussão, leitura e aprovação da versão final do artigo.

Referências

BAVARESCO, D. **Modelagem matemática e controle de um atuador pneumático**. 2007. Dissertação (Mestrado em Modelagem Matemática) – Universidade Regional do Noroeste do Estado do Rio Grande do Sul, Ijuí, 2007. Disponível em: <http://www.simmer.unijui.edu.br/Dissertacoes/Dissertação%20Delair%20Bavaresco.pdf>. Acesso em: 12 ago. 2023.

BOBROW, J. E.; MCDONELL, B. W. Modeling, identification, and control of a pneumatically actuated, force controllable robot. **IEEE Transactions on Robotics and Automation**, v. 14, n. 5, p. 732-742, 1998. DOI: <https://doi.org/10.1109/70.720349>.

BUFFONI, L.; OCHEL, L.; POP, A.; FRITZSON, P.; FORS, N.; HEDIN, G.; TAHA, W.; SJÖLUND, M. Open source languages and methods for cyber-physical system development: overview and case studies. **Electronics**, v. 10, n. 8, 902, 2021. DOI: <https://doi.org/10.3390/electronics10080902>.

CAVALCANTE, G. B.; ARAÚJO, Í. B. Q.; AQUINO, A. S.; COSTA JUNIOR, A. G. Identificação de sistemas para obter modelos lineares via mínimos quadrados para um sistema eletropneumático. **Revista da Sociedade Brasileira de Automática**, v. 1, n. 1, p. 1-8, 2019. Trabalho apresentado no 22º Congresso Brasileiro de Automática, 2018, João Pessoa. Disponível em: https://www.sba.org.br/open_journal_systems/index.php/cba/article/view/841. Acesso em: 21 dez. 2023.

COSTA, G. K. Modelling air flow through pneumatic valves: a brief review with an experimental case study. **Eng**, v. 4, n. 4, p. 2601-2614, 2023. DOI: <https://doi.org/10.3390/eng4040149>.

ENDLER, L.; VALDIERO, A. C.; ANDRIGHETTO, P. L.; RAFIKOV, M. Simulação computacional de um modelo matemático para atuadores pneumáticos. **TEMA. Tendências em Matemática Aplicada e Computacional**, v. 9, n. 2, p. 213-222, 2008. Disponível em: <https://tcam.sbmac.org/tema/article/view/157>. Acesso em: 12 jun. 2023.

FIORINA, C.; CLIFFORD, I.; KELM, S.; LORENZI, S. On the development of multi-physics tools for nuclear reactor analysis based on OpenFOAM®: state of the art, lessons

learned and perspectives. **Nuclear Engineering and Design**, v. 387, 111604, 2022. DOI: <https://doi.org/10.1016/j.nucengdes.2021.111604>.

LAIB, K.; PHAM, M. T.; LIN-SHI, X.; MEGHNOUS, R. Average modeling and nonlinear observer design for pneumatic actuators with on/off solenoid valves. **Journal of Dynamic Systems, Measurement, and Control**, v. 144, n. 2, 021006, 2022. DOI: <https://doi.org/10.1115/1.4052394>.

MANDALI, A.; DONG, L. Modeling and cascade control of a pneumatic positioning system. **Journal of Dynamic Systems, Measurement, and Control**, v. 144, n. 6, 061004, 2022. DOI: <https://doi.org/10.1115/1.4053966>.

OLIVEIRA, R. N. **Modelo dinâmico e estudo experimental para um resfriador de uma bomba de calor operando com CO₂ para aquecimento de água residencial**. 2013. Tese (Doutorado em Engenharia Mecânica) – Escola de Engenharia, Universidade Federal de Minas Gerais, Belo Horizonte, 2013. Disponível em: <https://repositorio.ufmg.br/handle/1843/BUOS-9ABJ6E>. Acesso em: 12 jun. 2023.

RICHTER, R. R. M. **Modelagem matemática e controle de posição de um atuador linear acionado pneumáticamente**. 2013. Dissertação (Mestrado em Modelagem Matemática) – Universidade Regional do Noroeste do Estado do Rio Grande do Sul, Ijuí, 2013. Disponível em: <http://www.simmer.unijui.edu.br/Dissertacoes/Dissertação%20Rozimerli%20Raquel%20Milbeir%20Richter.pdf>. Acesso em: 6 ago. 2023.

SEABRA, E.; SILVA, L. F.; SILVA, F.; VIEIRA, J. Modelica como uma linguagem de engenharia: modelação de sistemas mecânicos/mecatrônicos. *In*: CONFERÊNCIA INTERNACIONAL DA SOCIEDADE PORTUGUESA PARA A EDUCAÇÃO EM ENGENHARIA (CISPEE 2016), 2., 2016, Vila Real. **Anais [...]**. Vila Real: SPEE, 2016. Disponível em: <https://repositorium.sdum.uminho.pt/handle/1822/50829>. Acesso em: 21 dez. 2023.

SIMÕES, R. M. I. **Sistemas hidráulicos e pneumáticos**. Londrina: Editora e Distribuidora Educacional, 2016.

TILLER, M. **Introduction to physical modeling with Modelica**. Dordrecht: Kluwer Academic Publishers, 2001.

TINNERHOLM, J.; POP, A.; SJÖLUND, M. A modular, extensible, and Modelica-standard-compliant OpenModelica compiler framework in Julia supporting structural variability. **Electronics**, v. 11, n. 11, 1772, 2022. DOI: <https://doi.org/10.3390/electronics11111772>.

Apêndice: códigos-fonte em Modelica

Resistor

```

within Modelica.Electrical.Analog.Basic;
model Resistor "Ideal linear electrical resistor"
parameter SI.Resistance R(start=1)
  "Resistance at temperature T_ref";
parameter SI.Temperature T_ref=300.15 "Reference temperature";
parameter SI.LinearTemperatureCoefficient alpha=0
  "Temperature coefficient of resistance (R_actual = R*(1 + alpha*(T_heatPort - T_ref)));

extends Modelica.Electrical.Analog.Interfaces.OnePort;
extends Modelica.Electrical.Analog.Interfaces.ConditionalHeatPort(T=T_ref);
SI.Resistance R_actual
  "Actual resistance = R*(1 + alpha*(T_heatPort - T_ref));

equation
assert((1 + alpha*(T_heatPort - T_ref)) >= Modelica.Constants.eps,
  "Temperature outside scope of model!");
R_actual = R*(1 + alpha*(T_heatPort - T_ref));
v = R_actual*i;
LossPower = v*i;
annotation (...);
end Resistor;

```

PneumaticConnector

```

connector PneumaticConnector
Modelica.Units.SI.Pressure p;
flow Modelica.Units.SI.MassFlowRate m;
annotation(...);
end PneumaticConnector;

```

Exhaust

```

model Exhaust
PneumaticPackage.Pneumatic.PneumaticConnector x1 annotation (...);
equation
x1.p = 101325; //Pressão atmosférica em Pascal
annotation (...);
end Exhaust;

```

PressureSource

```

model PressureSource
PneumaticConnector x1 annotation (...);
parameter Modelica.Units.SI.Pressure P1;
equation
x1.p = P1;
annotation (...);
end PressureSource;

```

Airtank

```

model Airtank
//..... constantes
constant Modelica.Units.SI.MolarMass M = 0.02896 "massa molar do ar";
constant Real k = 8.31 "constante universal dos gases";
constant Modelica.Units.SI.Temperature T = 293 "temperatura em K";
//.....parametros
parameter Modelica.Units.SI.Volume V "volume do reservatório";
parameter Modelica.Units.SI.Pressure pi = 101325 "Pressão inicial";
//.....Variáveis internas
Modelica.Units.SI.MassFlowRate Vm "vazao em massa na entrada";
Modelica.Units.SI.Pressure p "pressao no reservatorio";
// .....Conectores
PneumaticPackage.Pneumatic.PneumaticConnector x1 annotation(...);
initial equation
pi=p;
equation
Vm = x1.m + x2.m;
Vm = (V*M)*der(p)/(k*T);
x1.p = p;
x2.p = p;
annotation(...);
end Airtank;

```

Compressor

```

model Compressor
PneumaticPackage.Pneumatic.PneumaticConnector x1 annotation (...);
Modelica.Units.SI.Pressure p "Pressão do sist"ma";
constant Modelica.Units.SI.Density d = 1.20 "1 "densidade do"ar";
parameter Real f "Frequência do motor em"HZ";
parameter Real n "rendime"to";
parameter Modelica.Units.SI.Volume V "volume da câmara do compres"or";
parameter Modelica.Units.SI.Pressure p1 "Pressão máxima de opera"ão";
equation
x1.p = p;
    if p>p1 then
        x1.m = 0;
    else
        x1.m = -V*d*f*n;
    end if;
annotation (...);
end Compressor;

```

ThreeXTwoA

```

model ThreeXTwoA
Modelica.Blocks.Interfaces.RealInput r annotation(...);
PneumaticPackage.Pneumatic.PneumaticConnector x3 annotation(...);
PneumaticPackage.Pneumatic.PneumaticConnector x2 annotation(...);
PneumaticPackage.Pneumatic.PneumaticConnector x1 annotation(...);
Modelica.Units.SI.MassFlowRate q_ma, q_mac;
Modelica.Units.SI.Pressure P_d,P_u;
Modelica.Units.SI.Pressure Af, Amax = 0.000001;
constant Modelica.Units.SI.Temperature T = 293.15;
constant Real B = 0.528 "critical ratio", y = 1.4, R = 287"constante universal dos gases";
// Auxiliares
Real phi, a, b, c, z;
equation
q_mac = Af*P_u*sign(a)*sqrt(abs(a));
a = ((2*y)/(z))*b;
z = (y-1)*R*T;
b = ((phi)^(2/y)) - ((phi)^((y+1)/y));
q_ma = Af*P_u*sign(c)*sqrt(abs(c));
c = (y/(R*T))*((2/(y+1))^(y+1)/(y-1));
if P_d<P_u then
    phi = P_d/P_u;
else
    phi = 1;
end if;
Af = Amax; //abertura máxima e constante
if r<0 then
//desenergizado
if phi > B then
    x2.m = q_mac;
    x2.p = P_u;
    x3.p = P_d;
    x1.m = 0;
    x2.m + x3.m=0;
else
    x2.m = q_ma;
    x2.p = P_u;
    x3.p = P_d;
    x1.m = 0;
    x2.m + x3.m = 0;
end if;
else
//energizado
if phi > B then
    x1.m = q_mac;
    x1.p = P_u;
    x2.p = P_d;
    x3.m = 0;
    x1.m + x2.m = 0;
else
    x1.m = q_ma;
    x1.p = P_u;
    x2.p = P_d;
    x3.m = 0;
    x1.m + x2.m = 0;
end if;
end if;
annotation(...);
end ThreeXTwoA;

```

ThreeXTwoB

```

model ThreeXTwoB
  Modelica.Blocks.Interfaces.RealInput r annotation(...);
  PneumaticPackage.Pneumatic.PneumaticConnector x3 annotation(...);
  PneumaticConnector x2 annotation(...);
  PneumaticPackage.Pneumatic.PneumaticConnector x1 annotation(...);
  parameter Real Bf = 0.000002577"constante de enchimento";
  parameter Real Be = 0.00000008387"constante de esvaziamento";
equation
  if (r > 0) then //enchendo
    if x1.p >= x2.p then
      x3.m = 0;
      x1.m + x2.m = 0;
      x1.m = Bf*sqrt((abs(x1.p-x2.p)));
    else
      x3.m = 0;
      x1.m + x2.m = 0;
      x1.m = 0;
    end if;
  else //esvaziando
    x1.m = 0;
    x2.m + x3.m = 0;
    x2.m = Be*(x2.p - x3.p);
  end if;
  annotation();
end ThreeXTwoB;

```

FiveXTwoB

```

model FiveXTwoB
  Modelica.Blocks.Interfaces.RealInput r annotation(...);
  PneumaticPackage.Pneumatic.PneumaticConnector x1 annotation(...);
  PneumaticPackage.Pneumatic.PneumaticConnector x2 annotation(...);
  PneumaticPackage.Pneumatic.PneumaticConnector x5 annotation(...);
  PneumaticPackage.Pneumatic.PneumaticConnector x4 annotation(...);
  PneumaticPackage.Pneumatic.PneumaticConnector x3 annotation(...);
  parameter Real Bf = 0.000002577"constante de enchimento";
  parameter Real Be = 0.00000008387"constante de esvaziamento";
equation
  if (r > 0) then //Energizado
    x1.m = 0;
    x5.m + x3.m = 0;
    x5.m = Be*(x5.p - x3.p);
    if x4.p >= x2.p then
      x4.m + x2.m = 0;
      x4.m = Bf*sqrt((abs(x4.p-x2.p)));
    else
      x4.m + x2.m = 0;
      x4.m = 0;
    end if;
  else //Desenergizado
    x3.m = 0;
    x2.m + x1.m = 0;
    x2.m = Be*(x2.p - x1.p);
    if x4.p >= x5.p then
      x4.m + x5.m = 0;
      x4.m = Bf*sqrt((abs(x4.p-x5.p)));
    else
      x4.m + x5.m = 0;
      x4.m = 0;
    end if;
  end if;
  annotation(...);
end FiveXTwoB;

```

Actuator

```

model Actuator
  PneumaticPackage.Pneumatic.PneumaticConnector x1 annotation(...);
  PneumaticPackage.Pneumatic.PneumaticConnector x2 annotation(...);
  Modelica.Units.SI.Pressure Pa "pressão sobre A";
  Modelica.Units.SI.Pressure Pb "pressão sobre B";
  Modelica.Units.SI.Volume Va "volume de A";
  Modelica.Units.SI.Volume Vb "volume de B";
  Modelica.Units.SI.Area Aa "área de A";
  Modelica.Units.SI.Area Ab "área de B";
  Modelica.Units.SI.Distance y1 "posição do embolo";
  Modelica.Units.SI.Velocity y2 "velocidade do embolo";
  parameter Real B = 90 "Atrito Viscoso";
  constant Real Y = 1.4 "relação do calor específico do ar";
  constant Real R = 287 "constante universal dos gases";
  constant Modelica.Units.SI.Temperature T = 293.15 "temperatura";
  parameter Modelica.Units.SI.Diameter D "diâmetro do embolo";
  parameter Modelica.Units.SI.Diameter d "Diâmetro da haste";
  parameter Modelica.Units.SI.Mass M "massa na haste";
  parameter Modelica.Units.SI.Distance Alc "alcance";
initial equation
  y1 = 0.001;
  Pa = 101325;
  Pb = 101325;
equation
  //----- dinâmica das pressões -----
  der(Pa) = (Y/(Aa*y1))*(-y2*Pa*Aa + R*T*x1.m);
  der(Pb) = (Y/(-Ab*y1))*(y2*Pb*Ab + R*T*x2.m);
  der(y2) = ((Aa*(Pa)/M) - (Ab*(Pb)/M) - (B*y2/M));
  //----- Terminais -----
  Pa = x1.p;
  Pb = x2.p;
  // ----- áreas -----
  Aa = 3.14*(D^2)/4;
  Ab = Aa - (3.14*(d^2)/4);
  //----- volume em função da posição ---
  Va = Aa*y1;
  Vb = (Ab)*(Alc - y1);
  //----- Condicionais -----
  if ((Alc<y1 and y2>0) or (y1<0 and y2<0)) then
  der(y1) = 0;
  else
  der(y1) = y2;
  end if; annotation(...);
end Actuator;

```