

Desenvolvimento de um software de simulação para o ensino de modelagem e análise de sistemas lineares

Ítalo A. Aguiar* Marcus V. Paula** Rodrigo A. Ricco***
Rodrigo G. Ribeiro****

* Programa de pós-graduação em Ciência da Computação (PPGCC),
Departamento de Computação – Universidade Federal de Ouro Preto
(UFOP), (e-mail: italo.aguiar@aluno.ufop.edu.br).

** Programa de pós-graduação em Engenharia Elétrica (PPGEE) –
Universidade Federal de Minas Gerais (UFMG), (e-mail:
marcusdepaula@ufmg.br)

*** Instituto de Ciências Exatas e Aplicadas (ICEA), Departamento de
Engenharia Elétrica – Universidade Federal de Ouro Preto (UFOP),
(e-mail: ricco@ufop.edu.br)

**** Programa de pós-graduação em Ciência da Computação
(PPGCC), Departamento de Computação – Universidade Federal de
Ouro Preto (UFOP) (e-mail: rodrigo.ribeiro@ufop.edu.br)

Abstract: Simulations applied in the educational context are capable of providing considerable support in the learning process of a given area of knowledge. This work proposes the development of an interactive, free and open source simulation *software* for educational use in the context of Linear Systems Modeling and Analysis. Within the scope of the discipline, the proposed *software* allows creating and graphically simulating variations of user-defined scenarios in real time or instantly. Using data from a plant of tanks created and measured in one of the related works, it was possible to validate the *software* implementation of this work by replicating the same experiment virtually. By obtaining a model response relatively similar to the original study, we concluded the validity and relevance of the simulation system in the target context.

Resumo: Simulações aplicadas no contexto educacional são capazes de prover um considerável apoio no processo de aprendizado de determinada área de conhecimento. Este trabalho propõe o desenvolvimento de um *software* de simulação interativo, gratuito e de código aberto para uso educacional no contexto da disciplina de Modelagem e Análise de Sistemas Lineares. O *software* proposto é capaz de compor e simular graficamente, dentro do escopo da disciplina, variações de cenários definidos pelos usuários em tempo real ou de forma instantânea. Utilizando-se de dados de uma planta de tanques criada e mensurada em um dos trabalhos relacionados, foi possível realizar a validação da implementação de *software* deste trabalho ao se replicar o mesmo experimento de forma virtual. Ao obter uma resposta do modelo relativamente próxima ao estudo original, concluímos a validade e a relevância do sistema de simulação no contexto educacional ao qual se propõe.

Keywords: Simulation; Software; Educational; Engineering; Physics; Modeling;

Palavras-chaves: Simulação; Software; Educacional; Engenharia; Física; Modelagem;

1. INTRODUÇÃO

De um modo geral, a matemática e a física são vistas como disciplinas de caráter intelectual avançado, e de considerável dificuldade em sua plena compreensão. A modelagem matemática de processos físicos do mundo real também é vista dessa forma, sendo por muitas vezes, um dos maiores obstáculos na discência (Marra et al., 2012). Embora toda a didática possa ser transmitida puramente de forma verbal e textual, o uso de metodologias gráficas e dinâmicas podem apresentar um impacto muito positivo para o discente.

Nas grades curriculares da engenharia elétrica e engenharias correlacionadas, tais como, a engenharia de computação e engenharia de controle e automação, um dos requisitos da graduação é o ensino de sistemas de controle e suas aplicações (ACM and IEEE, 2016). O estudo de sistemas de controle possui sua importância curricular ao permitir a concepção e criação de sistemas voltados para a medicina, indústria e economia (Kheir et al., 1996).

No ensino de sistemas de controle, parte da didática deve ser voltada para práticas e experimentação em laboratórios da instituição de ensino. A prática aplicada sobre os conceitos teóricos é um importante requisito na formação

de um profissional em engenharia (Kheir et al., 1996). Entretanto, em muitos casos, as práticas laboratoriais requerem o uso de equipamentos específicos ou de custo elevado. Isso impõe uma barreira para instituições com baixo poder aquisitivo ou mesmo estudantes que desejam conduzir algum tipo de implementação prática em sua própria residência. Uma proposta para tentar reduzir o impacto ao aprendizado causado por limitações de orçamento é o uso de *softwares* de simulação para emular as práticas laboratoriais.

Os *softwares* de simulação nascem da necessidade de realizar experimentos ou medições acerca do mundo real sem o uso de qualquer instrumento ou aparato, ou construção do objeto de estudo. Por exemplo, é possível simular o comportamento de um circuito sem a necessidade de implementá-lo e possuir uma complexa instrumentação para observá-lo.

Embora não seja o ideal, os simuladores podem permitir que o aluno observe na prática exatamente os conceitos matemáticos e físicos discutidos em sala de aula, sem necessariamente possuir toda uma infraestrutura laboratorial (Wang, 2021). Nesse sentido, este trabalho apresenta uma ferramenta de *software* desenvolvida com objetivos didáticos, capaz de simular interativamente o comportamento de sistemas de tanques de líquidos. Este é um cenário comumente presente em referências da área de modelagem e análise de sistemas lineares, como por exemplo o trabalho publicado por Cao et al. (2013) com ênfase específica neste tópico.

O presente trabalho encontra-se dividido da seguinte maneira. A Seção 2 apresenta as justificativas para a realização do trabalho. Na Seção 3 é apresentado o *software* proposto. Na Seção 4, é apresentada a modelagem física do sistema utilizado na aplicação. A Seção 5 mostra os resultados obtidos, enquanto a Seção 6 discute sobre a possibilidade de trabalho futuros. Por fim, a Seção 7 apresenta as conclusões desse trabalho.

2. JUSTIFICATIVA

Não é muito difícil encontrar algum *software* de simulação capaz de ser aplicado aos requisitos curriculares de modelagem e análise de sistemas lineares. Fazendo-se uma revisão bibliográfica no contexto da disciplina de Modelagem e Análise de Sistemas Lineares, recorrentemente o *Matrix Laboratory - MATLAB*¹, juntamente com seu pacote chamado *Simulink*, é utilizado para a pesquisa acadêmica e o ensino (Oldaker and Gilbert, 1995; Rumanová and Ďuriš, 2016; Chekichev et al., 2020). Isso acontece, por exemplo, em Ottoni et al. (2020), onde os autores propõem uma ferramenta educacional implementada na plataforma do *MATLAB* para aprendizado por reforço, um campo da Inteligência Artificial. Isso demonstra que apesar de inicialmente o *MATLAB* não possuir as ferramentas em determinado campo de estudo, sua extensibilidade permite que novos componentes possam ser desenvolvidos para suprir possíveis demandas que não estão cobertas pelas funcionalidades iniciais do *software*.

¹ Disponível em: <<https://www.mathworks.com/products>> Acesso em: 10 mar, 2022.

O *MATLAB*, é um conjunto de ferramentas de *software* voltado para cálculo numérico (Moler and Little, 2020). Nas bibliotecas que integram o pacote do *software* estão a de processamento de sinais, simulação de circuitos elétricos, telecomunicações, sistemas financeiros, dentre muitas outras. Além disso, o *MATLAB* possui uma linguagem de programação própria que pode ser utilizada de forma geral por meio de comandos em um terminal ou por meio de um arquivo de *script*.

Outra ferramenta de simulação que pode ser utilizada com objetivos didáticos é o EcosimPro. O EcosimPro é um *software* comercial de simulação de sistemas contínuos e discretos, utilizado em aplicações espaciais, aeronáuticas e industriais. O EcosimPro fornece um ambiente gráfico de componentes reutilizáveis capaz de permitir a construção e simulação de sistemas complexos (Isselhorst, 2010). O EcosimPro é a ferramenta oficial da Agência Espacial Europeia, utilizada em simulações para vários estudos². O EcosimPro está disponível para licenciamento comercial, acadêmico e individual.

Ainda existem ferramentas matemáticas gratuitas e de código aberto, como o Octave³ ou o Scilab⁴. Essas ferramentas abrangem parte das funcionalidades observadas em soluções comerciais como, por exemplo, as soluções anteriormente descritas, *MATLAB* e EcosimPro (Sharma and Gobbert, 2010).

No caso das soluções comerciais existe uma barreira financeira em sua adoção. O custo de licenciamento pode ser demasiadamente elevado para, por exemplo, equipar todos os computadores dos laboratórios de uma instituição de ensino com recursos financeiros limitados. Ainda que consideremos que a instituição equipe seus laboratórios com estes *softwares*, não haveria espaço para sua utilização fora do ambiente institucional sem a concessão de algum licenciamento discente. Isso impede, por exemplo, que os estudantes possam utilizá-los em casa, em atividades extra-classe, nos casos em que a instituição de ensino não possua uma licença de *software* mais permissiva que englobe este uso.

Logo, se existem várias soluções presentes no mercado, comerciais ou gratuitas, por que desenvolver algo novo? A resposta para esta pergunta é voltada para o nível de customização de uma solução no campo de estudo. Em geral, as soluções existentes, sejam comerciais ou gratuitas, disponibilizam um conjunto de características e ferramentas de escopo genérico. Embora isso permita que seu uso consiga abranger um conjunto maior de possibilidades, a ausência de funcionalidades mais específicas e didáticas podem impor uma curva de aprendizado menos efetiva.

Ao observar o problema sob uma perspectiva diferente, o recente evento de pandemia global de COVID-19, além dos efeitos observáveis na saúde e economia (Ozili and Arun, 2020), também evidenciou o despreparo das instituições de ensino superior ao fornecer metodologias de ensino

² Disponível em: <<https://www.ecosimpro.com>> Acesso em: 10 mar, 2022.

³ Disponível em: <<https://www.gnu.org/software/octave/index>> Acesso em: 10 mar, 2022.

⁴ Disponível em: <<https://www.scilab.org>> Acesso em: 10 mar, 2022.

remoto capazes de substituir práticas requeridas pelas grades curriculares.

Segundo uma pesquisa realizada em maio de 2020 pela *International Association of Universities – IAU*, uma associação da UNESCO, juntamente às instituições de ensino superior ao redor do mundo, várias das instituições entrevistadas declararam não estar preparadas para a mudança do ensino presencial para o *online*. Em parte este despreparo vinha da ausência de infraestrutura para prover o ensino ou mesmo o nível de preparo do corpo docente (Marinoni et al., 2020). Isso mostra que ferramentas de apoio ao ensino, capazes de serem executadas independentemente de localização geográfica, são adições úteis aos conjuntos de opções didáticas.

Este trabalho visa suprir parte das necessidades acadêmicas observadas na disciplina de Modelagem e Análise de Sistemas Lineares da Universidade Federal de Ouro Preto - UFOP. O ambiente de simulação virtual desenvolvido possui a capacidade de simular interativamente, de forma instantânea ou em tempo real, cenários configuráveis de plantas de tanques, com suporte a animações gráficas de eventos dos modelos. A ferramenta proposta possui um escopo de uso voltado à engenharia, onde os cursos de Engenharia Elétrica, Engenharia de Controle e Automação e Engenharia de Computação, por exemplo, podem se beneficiar de seu uso como ferramenta de apoio em parte de suas grades curriculares. Além disso, as principais barreiras financeiras, de geolocalização ou estrutura são superadas pela ferramenta desenvolvida ao ser disponibilizada de forma gratuita e ser passível de execução em diferentes sistemas operacionais.

3. MASLAB

Nomeado como MASLAB, um acrônimo de MASL (Modelagem e análise de sistemas lineares) e LAB (Laboratório) o *software* desenvolvido nasceu das necessidades acadêmicas no ensino da disciplina, também denominada Modelagem e análise de sistemas lineares, ou simplesmente, MASL, da Universidade Federal de Ouro Preto. A ideia por trás da ferramenta é oferecer um conjunto amplo de possibilidades, utilizando um sistema comumente abordado durante o ensino da disciplina: a modelagem de tanques de fluidos. A Figura 1 ilustra uma captura de tela da ferramenta desenvolvida durante a execução de uma simulação de exemplo.

O MASLAB foi desenvolvido em C# (*C Sharp*) dentro do .net Core da *Microsoft Corporation*. O .net Core é atualmente suportado pelos sistemas operacionais *Windows*, *macOS* e *Linux*. Porém, em sistemas operacionais diferentes do *Windows*, não há suporte oficial do *framework* para interfaces gráficas. Por este motivo, toda a interface gráfica foi desenvolvida utilizando o projeto *open-souce* Avalonia. Devido ao .net Core e Avalonia, MASLAB é totalmente compatível com os sistemas operacionais *Windows*, *macOS* e *Linux*.

O *software* desenvolvido permite que o usuário possa criar visualmente diagramas de tanques e suas inter-conexões. Cada um dos tanques da simulação possui duas entradas e duas saídas. A principal motivação para este número de entradas e saídas está na complexidade. Permitir um

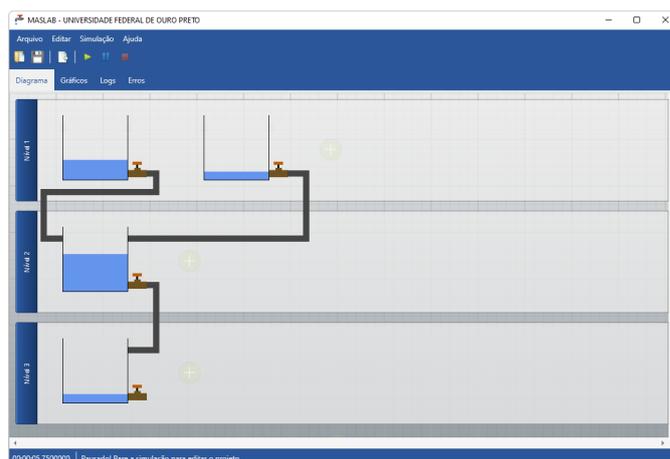


Figura 1. Tela principal do software desenvolvido.

número muito elevado de conexões poderia tornar o diagrama demasiadamente complexo, dificultando o objetivo didático proposto. No entanto, caso seja decidido aumentar o número de conexões, o código do *software* prevê mecanismos que facilitam a adaptação em uma nova versão.

O diagrama utiliza um sistema de níveis. Este sistema tem por objetivo refletir a posição relativa à altura em que cada tanque se encontra no mundo real. O objetivo do sistema de níveis é gráfico e não reflete diretamente no comportamento da simulação. Por exemplo, quando uma saída do tanque do nível inferior é conectada à entrada do tanque superior, é possível considerar na modelagem a força gravitacional exercida no deslocamento de massas entre os níveis.

O *software* desenvolvido foi construído para ser suficientemente genérico, de modo a permitir aplicar diferentes comportamentos a cada um dos tanques do diagrama. Para isso, cada um dos tanques possui um código na linguagem de programação C#. Dessa forma, é possível escrever códigos personalizados para motivações que não foram inicialmente previstas.

Durante o desenvolvimento da plataforma, houve uma preocupação com a curva de aprendizado do usuário em relação à linguagem de programação C#. Desse modo, um editor de texto com destaque sintático foi adicionado ao *software*. Paralelamente, uma integração com a plataforma do compilador foi escrita com a finalidade de permitir que o editor pudesse auto sugerir trechos de código e destacasse códigos com erros de sintaxe. O recurso de auto sugerir consegue fornecer uma descrição das propriedades, métodos e eventos da linguagem, permitindo que o usuário compreenda seu funcionamento. Este recurso também permite visualizar a assinatura de um método, assim como o número de parâmetros e seus respectivos tipos. Os erros sintáticos destacados no editor são acompanhados por uma descrição visível ao passar o *mouse* sobre o erro, permitindo compreendê-lo.

Algumas bibliotecas foram incluídas no *software* de modo que o usuário possa utilizá-las diretamente na codificação de um tanque. A exemplo disso, temos as bibliotecas

math.net⁵, que permitem executar algoritmos numéricos e simbólicos de forma simplificada no *software*.

Alguns métodos especiais também foram implicitamente adicionados ao código de cada tanque, como, por exemplo, o método **Plot**. Este método permite desenhar um gráfico para qualquer valor que o usuário necessite. O método exige três parâmetros, sendo, o nome da série de valores, o instante de tempo (eixo X), e o valor a ser inserido (eixo Y). Para que o usuário adicione pontos a uma série existente, basta incluir o nome da série desejada ao chamar o método. Caso a série não exista, ela será criada automaticamente.

Atualmente, as novas *releases* do MASLAB, incluindo os binários executáveis e instaladores para cada sistema operacional, além do código-fonte completo, podem ser obtidos no repositório do projeto no *GitHub*⁶.

4. MODELAGEM FÍSICA

Cada um dos tanques adicionados na simulação possui seu próprio código-fonte editável. Isso significa que o comportamento de cada tanque pode ser customizado com diferentes lógicas, caso haja necessidade. Quando um tanque é criado, o mesmo é carregado com um código de partida, capaz de executar o funcionamento básico esperado.

Essa independência entre as instâncias de tanques adicionados torna o processo de modelagem mais simples. Ao invés de modelar todo o sistema baseado na configuração desenhada, somente é necessário se preocupar com a codificação de um único tanque. A modelagem do tanque deve então considerar o balanço de massa da diferença entre a vazão volumétrica das entradas e a vazão de saída, conforme descrito em (1)

$$\frac{\partial m}{\partial t} = \sum \omega_e - \sum \omega_s. \quad (1)$$

Considerando que m é a massa de água no tanque, em Kg, e ω é a vazão dada, em Kg/s, é necessário realizar a conversão dessas vazões mássicas em vazões volumétricas. Para se transformar a vazão dada em Kg/s para L/s, primeiramente, é necessário obter o volume do conteúdo do tanque. Isso pode ser feito por meio do produto entre a área da base do tanque C e o nível h , de modo que

$$V = Ch. \quad (2)$$

A massa ocupada pelo fluido contido no tanque pode ser obtida como o produto de seu volume V pela densidade ρ , conforme apresentado a seguir

$$m = V\rho. \quad (3)$$

Substituindo (2) em (3), temos

$$m = Ch\rho. \quad (4)$$

Aplicando (4) em (1), temos o modelo geral do tanque, cujas entradas e saídas são dadas em L/s

$$\begin{aligned} \rho C \frac{\partial h}{\partial t} &= \sum q_e \rho - \sum q_s \rho, \\ \frac{\partial h}{\partial t} &= \frac{\sum q_e - \sum q_s}{C}, \end{aligned} \quad (5)$$

em que q_e e q_s são as vazões de entrada e saída do tanque, respectivamente. De posse da Equação (5) que modela o tanque, o passo final necessário é calcular a vazão de saída do tanque para que possa ser utilizado em (5). Visualmente, cada um dos tanques exibe uma válvula ou registro. Os tanques virtuais apresentam uma propriedade denominada *ValveConstant* (K) para armazenar o valor da constante da válvula, caso seja utilizada na modelagem. Considerando essa informação, pode-se aplicar, em (5), a Equação de Bernoulli que descreve a vazão de fluidos através das válvulas

$$q_s = K\sqrt{h}. \quad (6)$$

O código individual de cada tanque faz herança direta com a classe base denominada *SimulationTank*. Esta é uma classe abstrata que define o contrato básico a ser implementado em cada tanque, além de oferecer vários métodos de suporte que podem ser utilizados na codificação dos tanques. Entre os métodos de suporte oferecidos estão os algoritmos de análise numérica *Runge-Kutta* de segunda ordem e *Runge-Kutta* de quarta ordem. Por padrão, o código inicialmente carregado nos tanques resolve a equação diferencial (5) utilizando o método de *Runge-Kutta* de quarta ordem. O código fonte, que descreve a modelagem padrão de um tanque, é apresentado no Código Fonte 1.

Como mencionado anteriormente, há somente a necessidade de se preocupar com a modelagem de um único tanque virtual e, posteriormente, replicar a lógica em outros tanques, caso seja necessário. Isso é possível porque ao iniciar a simulação, a primeira iteração é destinada a calcular a vazão de saída de cada tanque de forma independente, uma vez que $\sum q_e = 0$. A partir da segunda iteração do algoritmo as saídas reais podem ser observadas.

5. METODOLOGIA DE TESTES

Uma etapa importante no desenvolvimento do *software* é a execução dos testes. Os testes asseguram que a implementação realizada atende aos requisitos mínimos obrigatórios para o aceite de uma nova versão ou *release*. No caso deste artigo, o sistema desenvolvido propõe emular virtualmente cenários customizados baseados nas leis da física. Desta forma, é necessário garantir que cada uma das variáveis apresentadas na simulação sejam suficientemente próximas do mesmo cenário executado em um contexto real.

Para a realização desta validação, o ideal é que seja feita a construção de uma planta de tanques capaz de ser mensurada para comparar resultados práticos e emulados. O problema desta abordagem é que para se construir tal planta,

⁵ Disponível em: <<https://www.mathdotnet.com/>> Acesso em: 01 dez, 2021.

⁶ Disponibilizado em: <<https://github.com/italoaguilar/MASLAB/>> Acesso em: 26 Fev, 2020.

```
public override SimulationData OnUpdate(
    double tempo, double passo,
    double entrada1, double entrada2,
    double entrada3, double entrada4) {

    double saidaE = 0; //saída esquerda
    double saidaD = 0; //saída direita

    // ∂h/∂t
    double level = RungeKutta4(Level, tempo,
    tempo + passo, (t, x) => {

        //vazão de saída 1 e 2 dada por  $K\sqrt{h}$ 
        saidaE = ValveConstant1 * Math.Sqrt(x);
        saidaD = ValveConstant2 * Math.Sqrt(x);

        //  $(\sum q_e - \sum q_s)/C$ 
        double xd = (entrada1 + entrada2
        + entrada3 + entrada4 - saidaE - saidaD)
        / TankArea;

        if (x < 0)
            return 0;

        return xd;
    });

    //plota o nível do tanque (Serie, X, Y)
    Plot("Tanque X", tempo, Level);

    return new
        SimulationData(level, saidaE, saidaD);
}
```

Código Fonte 1. Código padrão da simulação.

há sempre o fator financeiro intrínseco para a aquisição de todos os materiais envolvidos e toda a instrumentação necessária para se observar os resultados. Em Paula et al. (2016), os autores constroem uma planta de três tanques com objetivos didáticos voltados à engenharia. Neste artigo, os autores apresentam uma série de técnicas para a identificação do sistema representado na planta de tanques. Além disso, o artigo apresenta uma série de dados obtidos por meio de amostragem de sensores posicionados na planta de tanques.

Os autores deste trabalho, quando contactados, prontamente disponibilizaram todos os dados coletados utilizados na elaboração do artigo. A ideia aqui é utilizar os dados obtidos para encontrar uma aproximação virtual dos mesmos resultados encontrados pelos autores no estudo, ao invés de se construir uma planta própria para a validação do MASLAB. A planta de tanques acoplados desenvolvida em Paula et al. (2016) é apresentada na Figura 2 (a). O sistema consiste em um conjunto de três tanques, cada um com a área da base igual a 150 cm^2 e 30 cm de altura, interconectados entre si, pela base, por válvulas. A planta ainda possui um reservatório abaixo dos três tanques, responsável por coletar e redistribuir o fluido para o primeiro e o terceiro tanque por meio de duas bombas hidráulicas de corrente contínua, cuja tensão de alimentação máxima é de 12V.

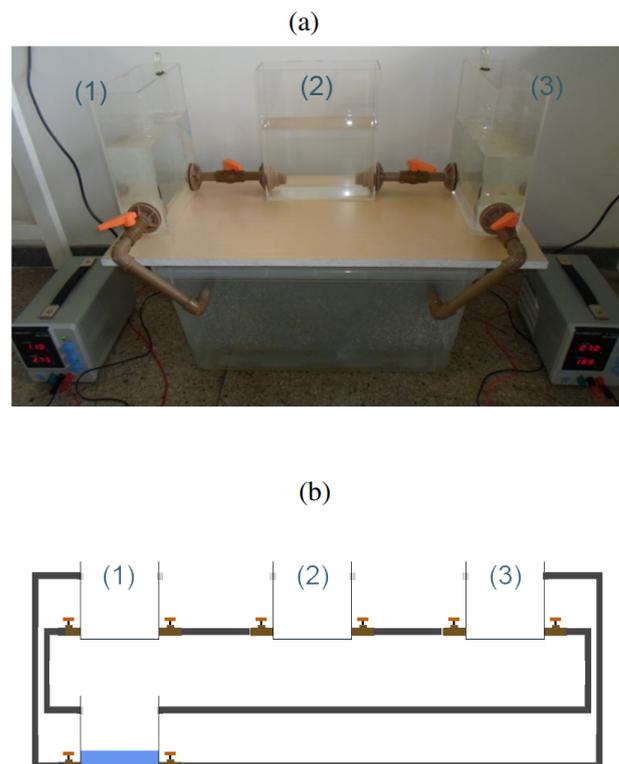


Figura 2. Sistema de três tanques acoplados. (a) Protótipo desenvolvido em Paula et al. (2016). (b) Implementação construída no MASLAB.

A planta de tanques de Paula et al. (2016) tem como entrada do modelo os níveis de tensão u aplicados individualmente nas duas bombas hidráulicas do sistema, enquanto a saída é a altura h de cada um dos três tanques principais. Entre os dados fornecidos pelos autores, estão os valores de nível de cada um dos tanques e os níveis de tensão aplicados nas entradas do modelo do artigo, amostrados a uma taxa de 1Hz por aproximadamente 4:30h. As entradas e saídas obtidas pelos autores podem ser observadas na Figura 3 (a) e (b), respectivamente.

Junto aos dados fornecidos pelos autores de Paula et al. (2016), há também um conjunto de pontos que descrevem a relação estática *tensão* \times *vazão* para as bombas hidráulicas utilizadas na planta de tanques, conforme ilustrado na Figura 4.

Analisando o conjunto de pontos, é possível observar um comportamento não linear na saída produzida pela bomba hidráulica. Com objetivo de obter uma aproximação do modelo da bomba hidráulica, aplicou-se um ajuste de curva logarítmica da forma $y = a + b \ln(x)$. Estimando os parâmetros a e b , a curva logarítmica que descreve o sistema é dada por

$$y(x) = -0,0566338 + 0,0568869 \ln(x). \quad (7)$$

Como a curva resultante de (7), ilustrada em vermelho na Figura 4, não é linear, para se encontrar uma aproximação linear do modelo, deve-se obter a reta tangente da equação em torno de um ponto de operação da bomba hidráulica.

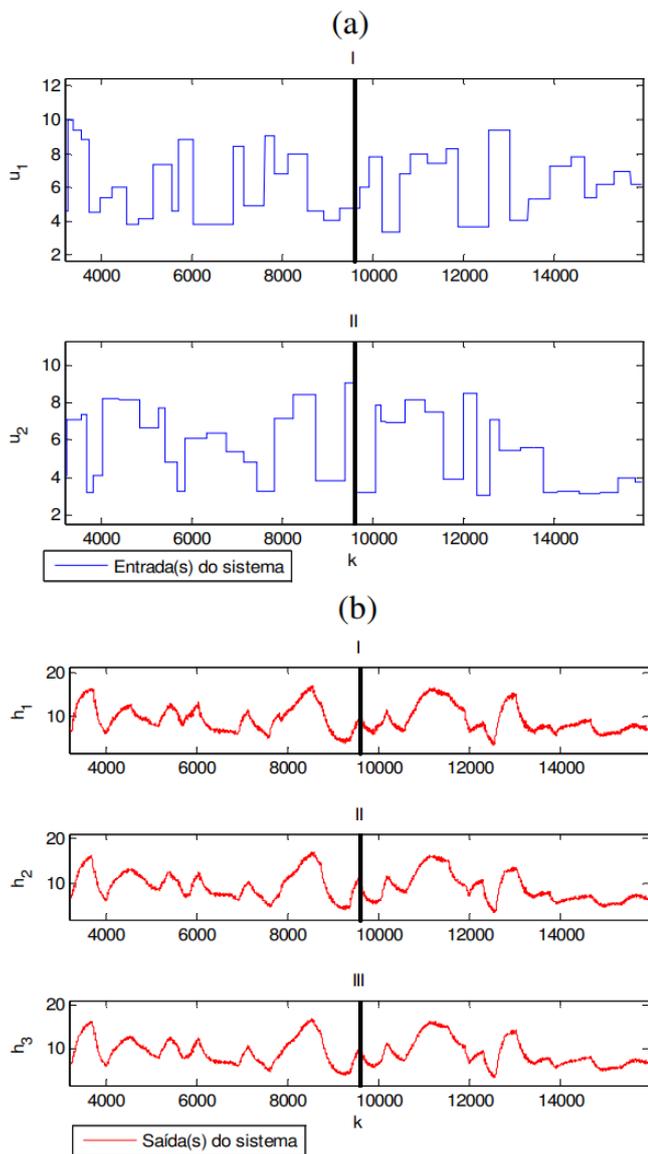


Figura 3. Entradas e saídas do sistema apresentado em Paula et al. (2016). (a) Sinais aleatórios de entrada (I) u_1 e (II) u_2 . (b) Sinais de saída (I) h_1 , (II) h_2 e (III) h_3 .

Derivando (7), obtemos a equação da reta tangente, de modo que

$$\frac{dy}{dx} = \frac{0,0568869}{x}. \quad (8)$$

Como a bomba hidráulica utilizada na planta opera em uma faixa de tensão de até 12V, foi estimada uma reta tangente, conforme ilustrado (em verde) na Figura 4, passando pelo ponto médio $x = 6$ da curva não linear (em vermelho).

Considerando o par ordenado $x_1 = 6$, $y_1 = 0,045294$ (obtido avaliando $x_1 = x = 6$ em (7)) e o coeficiente angular $m = 0,00948$ (obtido avaliando $x = 6$ em (8)), obtemos a equação geral da reta que faz uma aproximação linear do modelo da bomba hidráulica em torno de um ponto de operação

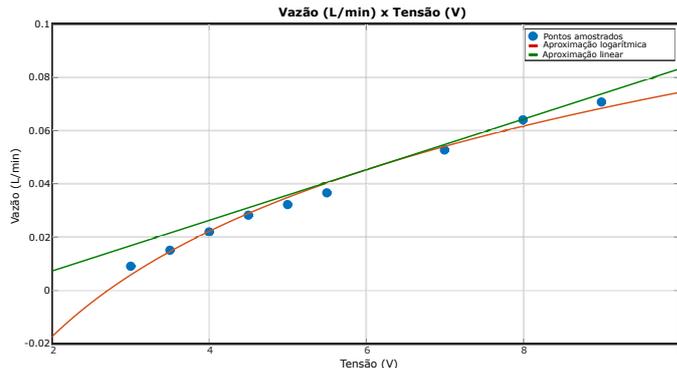


Figura 4. Vazão calculada para a bomba hidráulica em função da tensão de entrada. Em azul (●) têm-se as medições realizadas na bomba hidráulica. Em verde e vermelho tem-se, respectivamente, as curvas de aproximação linear e não linear para o conjunto de pontos medidos.

$$y = y_1 + m(x - x_1), \quad (9)$$

$$y = 0,00948x - 0,1135208. \quad (10)$$

De posse dos valores de entrada e características físicas da planta, o modelo de simulação exibido na Figura 2 (b) foi configurado no MASLAB. O tanque inferior, responsável por representar o reservatório, recebeu em seu código-fonte os conjuntos de dados das entradas do modelo para emular as bombas hidráulicas.

Os valores das constantes de Bernoulli para as válvulas do modelo foram estimados de forma experimental usando o próprio simulador. Para essa finalidade, foi necessário utilizar alguns pontos conhecidos da saída por meio dos dados apresentados na Figura 3 (b). Desta forma, o simulador varia os valores das constantes de modo a se aproximar dos pontos de referência.

O MASLAB foi configurado para simular o intervalo de 1s a 16000s a uma taxa de amostragem de 1Hz. Durante as simulações experimentadas houve a necessidade de se reduzir o valor do parâmetro a de 0,00948 para $\approx 0,005$, e b para 0. Uma possível justificativa para este ajuste é que a amostragem realizada com a bomba (Figura 4) não foi realizada no mesmo contexto da planta, além de desconsiderar a força gravitacional exercida sobre a massa de água da coluna formada pela tubulação da bomba entre o reservatório inferior e o tanque de destino, conforme pode ser observado na Figura 2 (a).

As saídas para o modelo simulado podem ser observadas na Figura 5. Foi realizada uma sobreposição dos dados obtidos no MASLAB com os dados do artigo de referência, para permitir comparar mais facilmente os resultados obtidos virtualmente e os resultados experimentais.

Para realizar a validação quantitativa do modelo estimado, foram calculados os índices RMSE (do inglês *root mean square error*) para as três saídas. A raiz quadrada do erro médio é uma métrica frequentemente utilizada em estudos de avaliação de modelos (Aguirre, 2007), sendo calculada do seguinte modo

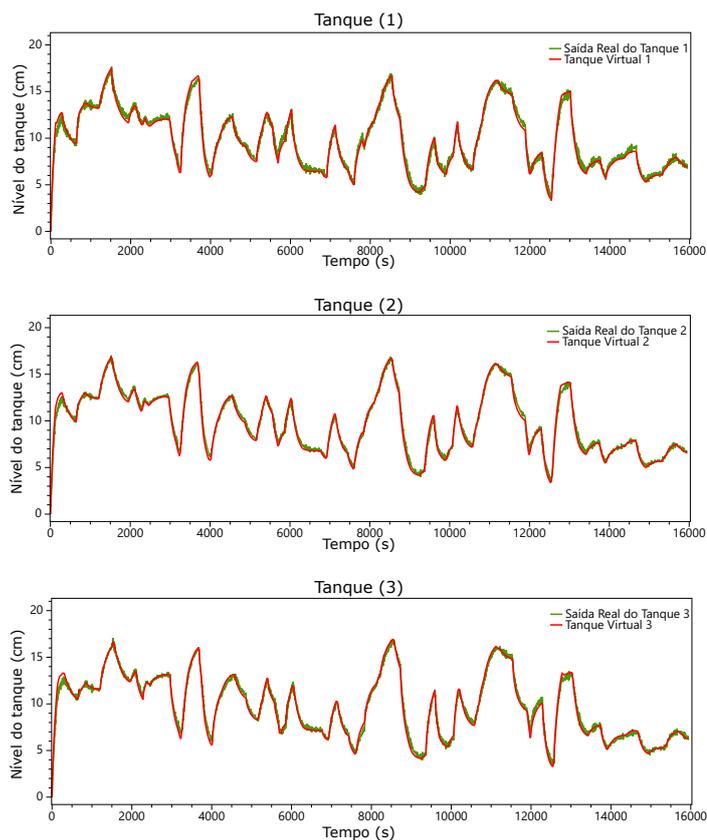


Figura 5. Comparação entre as saídas obtidas pelo MASLAB (em vermelho) e os dados do sistema real (em verde), obtidos do artigo de referência.

Tanque	Índice
Tanque 1	0,10945744067430496
Tanque 2	0,08493320643901825
Tanque 3	0,11031370609998703

Tabela 1. Índices RMSE obtidos para os três tanques.

$$RMSE = \frac{\sqrt{\sum_{k=1}^n (y(k) - \hat{y}(k))^2}}{\sqrt{\sum_{k=1}^n (y(k) - \bar{y})^2}} \quad (11)$$

Seu objetivo é comparar o quão próximo um sinal se encontra do outro. O RMSE é um índice normalizado (varia entre 0 e 1). Quanto mais próximo de zero for seu valor, melhor será o resultado da comparação. Os índices RMSE calculados para as três saídas mostradas na Figura 5 são apresentados na Tabela 1.

Analisando os resultados qualitativos (Figura 5) e quantitativos, observa-se que as saídas do modelo acompanham de forma satisfatória os dados gerados pelo sistema real.

6. TRABALHOS FUTUROS

O *software* desenvolvido foi projetado para atender a uma necessidade da disciplina de Modelagem e Análise de Sistemas Lineares. Nesta implementação, um dos cenários de estudo mais comuns, a modelagem de tanques de fluidos, foi escolhida para ser o ponto de partida de todas as funcionalidades subsequentes. Como parte dos trabalhos futuros, pretende-se expandir o ambiente de simulação para outros cenários também abordados na disciplina,

como por exemplo, a modelagem de circuitos elétricos ou modelagem de sistemas massa-mola.

De um ponto de vista técnico, atualmente, a aplicação é capaz de ser executada nos sistemas operacionais *Windows*, *Linux* e *MacOS*. Embora isso permita sua execução na grande maioria dos *desktops* e *notebooks* que tenham estes sistemas instalados, para a utilização do *software* desenvolvido ainda é necessário seu prévio *download*. Com a popularização do *Web assembly* (Haas et al., 2017) e o seu suporte adicionado ao *.net Core* no projeto *Blazor*⁷ da Microsoft, cabe ao mínimo a criação de uma *POC (Proof of Concept)*, para determinar se a execução do ambiente de simulação utilizando o *web browser* é viável. Isso tornaria o uso da ferramenta desenvolvida ainda menos restritivo.

7. CONCLUSÃO

Emular as práticas laboratoriais vivenciadas no ambiente acadêmico pode ser uma alternativa viável nas ocasiões em que o acesso aos recursos ou às instalações laboratoriais estiver indisponível, como no caso dos efeitos colaterais do COVID-19 na educação (Marinoni et al., 2020).

Neste trabalho apresentamos uma ferramenta de *software* desenvolvida a partir de algumas das necessidades observadas no ensino da disciplina de modelagem e análise de sistemas lineares. A partir do experimento realizado foi possível observar o êxito ao emular um cenário de estudo real com resultados relativamente próximos. Isso permite que qualquer pesquisador, estudante ou professor possa experimentar cenários de planta de tanques, independentemente de localização, sem ter tal infraestrutura de projeto, bastando apenas um computador e o *software* de simulação.

Todos os dados utilizados no experimento de validação, juntamente com o arquivo de simulação criado para o MASLAB, podem ser obtidos gratuitamente no repositório do projeto⁸.

AGRADECIMENTOS

Gostaríamos de agradecer à Universidade Federal de Ouro Preto - UFOP e a Pró-Reitoria de Pós Graduação - PROPP pela criação e disponibilização do Programa de Pós Graduação em Ciência da Computação - PPGCC.

REFERÊNCIAS

- ACM and IEEE (2016). Curriculum Guidelines for Undergraduate Degree, Programs in Computer Engineering. ACM and IEEE. doi:10.1145/3025098.
- Aguirre, L. (2007). *Introdução à Identificação de Sistemas - Técnicas Lineares e Não-Lineares Aplicadas a Sistemas Reais*, volume 3. Editora UFMG.
- Cao, Y., Cao, W.H., Yuan, Y., and Wu, M. (2013). Dynamic modeling and model mismatch of coupled water tanks system. In *Proceedings of the 32nd Chinese Control Conference*.

⁷ Disponível em: <<https://dotnet.microsoft.com/en-us/apps/aspnet/web-apps/blazor>> Acesso em: 25 fev, 2022.

⁸ Disponibilizado em: <<https://github.com/italoaguiar/MASLAB/tree/experimento>> Acesso em: 20 abr, 2022

- Chekichev, A.H., Shehova, D.A., Lyubomirov, S.Y., Ase-
nov, S.M., and Asparuhova, K.K. (2020). Researching
and modeling of discrete linear time-invariant systems
with difference equations in matlab. In *2020 XXIX In-
ternational Scientific Conference Electronics (ET)*. doi:
10.1109/ET50336.2020.9238252.
- Haas, A., Rossberg, A., Schuff, D.L., Titzer, B.L., Holman,
M., Gohman, D., Wagner, L., Zakai, A., and Bastien, J.
(2017). Bringing the web up to speed with webassembly.
In *Proceedings of the 38th ACM SIGPLAN Conference
on Programming Language Design and Implementation*.
- Isselhorst, A. (2010). Hm7b simulation with espss tool on
ariane 5 esc-a upper stage. doi:10.2514/6.2010-7047.
- Kheir, N., Åström, K., Auslander, D., Cheok, K.,
Franklin, G., Masten, M., and Rabins, M. (1996).
Control systems engineering education. *Automa-
tica*, 32(2). doi:https://doi.org/10.1016/0005-1098(96)
85546-4. URL [https://www.sciencedirect.com/
science/article/pii/0005109896855464](https://www.sciencedirect.com/science/article/pii/0005109896855464).
- Marinoni, G., Van't Land, H., Jensen, T., et al. (2020).
The impact of covid-19 on higher education around the
world. *IAU global survey report*, 23.
- Marra, R.M., Rodgers, K.A., Shen, D., and Bogue, B.
(2012). Leaving engineering: A multi-year single in-
stitution study. *Journal of Engineering Education*,
101(1). doi:https://doi.org/10.1002/j.2168-9830.2012.
tb00039.x. URL [https://onlinelibrary.wiley.com/
doi/abs/10.1002/j.2168-9830.2012.tb00039.x](https://onlinelibrary.wiley.com/doi/abs/10.1002/j.2168-9830.2012.tb00039.x).
- Moler, C. and Little, J. (2020). A history of matlab.
Proceedings of the ACM on Programming Languages,
4(HOPL).
- Oldaker, R. and Gilbert, J. (1995). An integrated package
for the simulation and practical evaluation of discrete
control systems running within the matlab environment.
In *IEE Colloquium on Applied Control Techniques Using
MATLAB*. doi:10.1049/ic:19950058.
- Otoni, A.L.C., Nepomuceno, E.G., and de Oliveira, M.S.
(2020). Development of a pedagogical graphical inter-
face for the reinforcement learning. *IEEE Latin America
Transactions*, 18(01).
- Ozili, P.K. and Arun, T. (2020). Spillover of covid-19:
impact on the global economy. *Available at SSRN*
3562570.
- Paula, M., Verly, A., and Ricco, R. (2016). Identificação
de sistemas: Uma abordagem didática por meio das
possibilidades de experimentação de uma planta de nível
de três tanques acoplados.
- Rumanová, L. and Ďuriš, V. (2016). Education of goni-
ometric functions with matlab software. In *2016 IEEE
10th International Conference on Application of Infor-
mation and Communication Technologies (AICT)*. doi:
10.1109/ICAICT.2016.7991804.
- Sharma, N. and Gobbert, M.K. (2010). A comparative
evaluation of matlab, octave, freemat, and scilab for
research and teaching. *UMBC Faculty Collection*.
- Wang, L. (2021). Construction and practice of computer
simulation course platform based on “new engineering”.
In *2021 International Conference on Education, Infor-
mation Management and Service Science (EIMSS)*. doi:
10.1109/EIMSS53851.2021.00098.