Matlab/Simulink como Ferramenta de Modelagem e Controle de Sistemas Dinâmicos: Uma Revisão

Rebecca Correa Recupero Janini, Arnaldo de Carvalho Junior

Programa de Pós-Graduação em Engenharia Mecânica (PPGEMEC), Universidade Santa Cecília (UNISANTA), Santos-SP, Brasil.

E-mail: rebecca.janini@icloud.com

Resumo: Este artigo apresenta uma revisão narrativa, seguindo o método PRISMA, sobre o uso do Matlab/Simulink na modelagem, simulação e controle de sistemas dinâmicos. Foram analisados estudos publicados entre 2006 e 2025, com ênfase em pesquisas de lógica paraconsistente aplicada a controle e identificação. Os resultados indicam que a ferramenta se consolidou como padrão acadêmico e industrial por oferecer prototipagem rápida, toolboxes dedicadas e integração com hardware via esquemas SIL e HIL. Apesar do custo de licenciamento frente a alternativas como o Scilab, sua robustez e a capacidade de interagir com simuladores e hardware externo a mantêm como referência no desenvolvimento e validação de algoritmos de controle.
Palavras-chave: Matlab; Simulink; Modelagem de Sistemas Dinâmicos; Engenharia de Controle; Identificação de Sistemas.

Matlab/simulink as a Tool for Modeling and Control of Dynamic Systems: A Review

Abstract: This paper presents a narrative review, following a PRISMA-method, on the use of Matlab/Simulink for modeling, simulation and control of dynamic systems. Studies published between 2006 and 2025 were analyzed, with emphasis on research applying paraconsistent logic to control and system identification. The results indicate that the tool has become an academic and industrial standard by offering rapid prototyping, dedicated toolboxes and hardware integration through software-in-the-loop (SIL) and hardware-in-the-loop (HIL) schemes. Despite its licensing cost when compared to alternatives such as Scilab, the robustness of its ecosystem and its ability to interact with external simulators and hardware keep it a reference for the development and validation of control algorithms.
Keywords: MATLAB; Simulink; Dynamic Systems Modeling; Control Engineering; System Identification.

Introdução

A modelagem e o controle de sistemas dinâmicos são fundamentais na engenharia contemporânea, com aplicações que vão do posicionamento de um pêndulo invertido a complexos sistemas aeroespaciais de controle de voo. Na perspectiva de controle, um sistema dinâmico pode ser descrito genericamente no espaço de estados conforme a Eq. (1):

$$\dot{x}(t) = f(x(t), u(t), t); \quad y(t) = h(x(t), u(t), t) \tag{1}$$

Em que $x(t)$ é o vetor de estados, $u(t)$ o vetor de entradas de controle, $y(t)$ o vetor de saídas

medidas, e f e h descrevem a dinâmica interna e a saída do sistema [2].

Historicamente, a obtenção, simulação e controle de sistemas descritos por essa formulação exigiam a implementação manual de algoritmos numéricos para resolver as equações diferenciais associadas, gerando lentidão, complexidade e erros de implementação.

O Matlab e o seu ambiente gráfico Simulink (MathWorks), transformou esse cenário ao oferecer, uma plataforma integrada para cálculo matricial, visualização e simulação por diagramas de blocos. Essa mudança de paradigma, de programar o solucionador numérico para definir a estrutura do sistema, ampliou o acesso mesmo para leigos em métodos numéricos, sem retirar dos usuários avançados o controle total sobre a linguagem de script, ajudando a situar a relevância dessas ferramentas computacionais [3].

Assim, a pergunta que orienta a pesquisa é: qual é a função do Matlab/Simulink no desenvolvimento de sistemas de controle e quais são suas vantagens e limitações frente a outras ferramentas computacionais? O objetivo geral é oferecer uma análise teórica sobre o uso do Matlab/Simulink na modelagem e no controle de sistemas dinâmicos. São objetivos específicos: (i) explicar a função da ferramenta em cada fase do ciclo de desenvolvimento de um sistema de controle; (ii) examinar aplicações recentes que unem controle clássico a técnicas de inteligência computacional; e (iii) avaliar criticamente suas limitações em relação a alternativas de código aberto. O estudo é relevante como fonte introdutória para pós-graduandos em engenharia que iniciam o uso dessas ferramentas em suas pesquisas.

Material e Métodos

Trata-se de uma revisão bibliográfica narrativa, de abordagem qualitativa, inspirada nas recomendações PRISMA [4]. A busca foi realizada nas bases Google Scholar, IEEE Xplore, Scopus, Web of Science e MDPI, considerando publicações entre 2006 e 2025, em português ou inglês, com as palavras-chave em inglês: “Matlab”, “Simulink”, “dynamic systems”, “control”, “system identification” e “paraconsistent logic”. Priorizaram-se estudos alinhados à linha de pesquisa em lógica paraconsistente (LP) aplicada a controle e identificação, direcionando a seleção ao escopo tratado nas seções seguintes.

O processo de seleção seguiu cinco etapas: (1) identificação dos registros nas bases; (2) remoção de duplicatas; (3) triagem por título e idioma; (4) filtragem por relevância do periódico ou editora (IEEE, Elsevier, Springer e MDPI); e (5) leitura dos resumos e verificação de aderência ao propósito da revisão. Cada fonte incluída foi avaliada quanto à questão abordada, à função específica do Matlab/Simulink na metodologia e aos resultados relatados. Os

quantitativos de cada etapa são apresentados na Figura 1 e devem ser confirmados pela autora conforme a busca efetivamente executada.

Resultados

A Figura 1 sintetiza o fluxo de seleção dos artigos. Partindo dos registros identificados nas bases, as etapas sucessivas de filtragem conduziram ao conjunto final nesta revisão.

Figura 1. Fluxo de seleção dos estudos (funil PRISMA).



A Tabela 1 resume os estudos incluídos, indicando autoria, aplicação, área e um breve comentário sobre o papel do Matlab/Simulink em cada trabalho.

Tabela 1. Valores obtidos dos ensaios no laboratório.

Autor (ano)	Aplicação	Área	Comentário
Carvalho Jr. et al. (2021)	Identificação e controle do pêndulo de Furuta por rede neural paraconsistente	Controle inteligente	Simulação da planta não linear e treinamento do controlador em ambiente único
Carvalho et al. (2023)	Estimador de estados paraconsistente para o pêndulo de Furuta	Estimação de estados	Papel análogo a observador/filtro, implementado via blocos S-function
Pereira et al. (2024a)	Modelagem cinemática do manipulador Dobot Magician	Robótica	Uso do núcleo de álgebra matricial (transformações homogêneas 4×4)
Santos et al. (2024)	Sistema de segurança por reconhecimento visual do Dobot	Visão computacional	Processamento de imagem para parada segura do robô
Pereira et al. (2024b)	Filtro PAL2v vs. filtro de Kalman no sensor MPU6050	Processamento de sinais	Funções nativas facilitam a comparação de técnicas de filtragem
Horri; Pietraszko (2022)	Co-simulação de controle de voo com simuladores externos	Aeroespacial / co-simulação	Validação SIL/HIL via UDP, MAVLink e CAN
Ucchas et al. (2022)	Análise comparativa de controladores PI e PID em inversor PWM	Controle clássico	Referência de projeto PID em MATLAB/Simulink
Kozák (2014)	Estado da arte da engenharia de controle	Revisão / fundamentos	Contextualiza gerações de técnicas de controle
De Morais; De Souza (2025)	Otimização topológica de treliças em Scilab	Alternativa de código aberto	Base comparativa frente ao MATLAB
Annigeri (2006)	Análise estrutural de pórticos planos em Scilab	Alternativa de código aberto	Compatibilidade parcial de sintaxe com o MATLAB

No eixo de identificação e controle inteligente, Carvalho Jr. *et al.* (2021) empregaram o

Matlab para identificar e controlar um pêndulo invertido rotativo (pêndulo de Furuta), sistema não linear, subatuado e instável em malha aberta, por meio de uma rede neural paraconsistente (RNP), simulando a planta e treinando o controlador na mesma plataforma. Carvalho *et al.* (2023) propuseram um estimador de estados paraconsistente para a mesma planta, atuando como um observador de Luenberger ou a um filtro de Kalman. A implementação por blocos S-function, que permitem inserir código Matlab ou C no diagrama de blocos, viabiliza a prototipagem ágil de algoritmos de controle não convencionais.

No eixo de robótica e processamento de sinais, Pereira et al. (2024a) modelaram a cinemática do manipulador Dobot Magician, obtendo as transformações homogêneas entre articulações pela convenção de Denavit-Hartenberg, aproveitando o núcleo de álgebra matricial do Matlab. Santos et al. (2024) desenvolveram, para o mesmo manipulador, um sistema de segurança por reconhecimento visual baseado em inteligência artificial, capaz de interromper o movimento diante da presença humana. Já Pereira et al. (2024b) compararam um filtro paraconsistente a um filtro de Kalman no processamento de dados de um acelerômetro/giroscópio MPU6050, tarefa comum em navegação inercial de baixo custo.

No eixo de co-simulação e validação, Horri e Pietraszko (2022) apresentaram um tutorial sobre co-simulação entre o Simulink e três simuladores de voo (X-Plane, FlightGear e VFTE), mostrando como sinais de atuadores calculados no Simulink são transmitidos em tempo real por protocolos de rede (UDP, MAVLink e CAN) a um simulador externo, que devolve os estados do sistema físico simulado. Essa estrutura viabiliza a validação de controladores robustos, como um controlador H-infinito com observador de Luenberger no controle lateral de um mini-VANT, em ambiente realista antes dos testes em *hardware*.

Discussão

O ciclo de desenvolvimento de um sistema de controle envolve quatro etapas principais: modelagem matemática da planta, projeto do controlador, simulação em malha fechada e validação em hardware. O Matlab dá suporte a quase todas por meio de toolboxes dedicadas. O Control System Toolbox permite representar sistemas no domínio da frequência, com funções de transferência $G(s) = Y(s)/U(s)$, e no espaço de estados, além de projetar controladores clássicos (PID, alocação de polos, *root locus*) e modernos (H-infinito, LQR/LQG). O System Identification Toolbox estima modelos a partir de dados de entrada e saída, sendo útil quando o modelo físico é difícil de derivar analiticamente. O Simulink representa graficamente a estrutura do sistema e resolve o conjunto de equações por

solucionadores numéricos configuráveis (Runge-Kutta de passo fixo ou variável, ou métodos implícitos para sistemas rígidos), ocultando a complexidade da integração numérica sem impedir o ajuste fino de tolerância e passo [1]. Um exemplo recorrente é o controle PID aplicado a uma planta de função de transferência $G(s)$, descrito pela Eq. (2):

$$C(s) = Kp + Ki/s + Kd \cdot s; \quad Y(s) = [C(s) \cdot G(s) / (1 + C(s) \cdot G(s))] \cdot R(s) \tag{2}$$

Embora conceitualmente simples, esse controlador reproduz o modelo empregado em situações reais mais complexas, como o ajuste dos ganhos *Kp*, *Ki* e *Kd* de um PID sobre uma planta de terceira ordem, normalmente realizado no Simulink pela facilidade de reconfiguração dos parâmetros e de visualização imediata da resposta ao degrau, sem exigir a solução analítica da equação diferencial de malha fechada a cada ajuste [2], [10]. A Tabela 2 sintetiza os papéis do Matlab/Simulink por etapa do ciclo, conforme os estudos revisados.

Tabela 2. Papéis do Matlab/Simulink por etapa do ciclo de desenvolvimento de controle.

Etapa do ciclo	Papel do MATLAB/Simulink	Exemplo na literatura revisada
Modelagem / Identificação	Álgebra matricial, System Identification Toolbox, blocos S-function	Carvalho et al. (2023); Pereira et al. (2024a)
Projeto de controlador	Control System Toolbox, ajuste de ganhos PID / H-infinito	Horri; Pietraszko (2022); Ucchas et al. (2022)
Simulação em malha fechada	Simulink (diagramas de blocos, solucionadores numéricos)	Carvalho Jr. et al. (2021)
Processamento de sinais / sensores	Filtros recursivos, visão computacional	Pereira et al. (2024b); Santos et al. (2024)
Validação SIL / HIL	Interfaces UDP / MAVLink / CAN com simuladores e hardware	Horri; Pietraszko (2022)

A análise revela que o Matlab/Simulink funciona como núcleo de um amplo ecossistema, integrando *hardware* físico (Pixhawk, MPU6050, manipuladores robóticos), simuladores externos especializados (X-Plane, FlightGear, VFTE) e algoritmos de inteligência computacional desenvolvidos pelo pesquisador. Essa capacidade de integração, mais do que qualquer função isolada, é o principal diferencial da ferramenta frente a alternativas que, embora eficientes em uma etapa específica, são menos versáteis quando o projeto precisa transitar entre modelagem, controle inteligente, processamento de sinais e validação em *hardware*. Nos trabalhos de LP, o Matlab é usado não apenas para simular a planta e o controlador de referência, mas como ambiente de desenvolvimento do próprio algoritmo proposto, o que reforça seu papel como plataforma de prototipagem para pesquisa.

Destaca-se a possibilidade de implementar rapidamente uma nova abordagem de controle, sem reescrever toda a estrutura de simulação numérica, como uma das principais

razões da permanência do Matlab. Mesmo com a difusão de linguagens de uso geral como o Python, que por padrão não dispõem de um ambiente de simulação por diagramas de blocos tão maduro quanto o Simulink, o Matlab mantém vantagem nesse aspecto.

A principal limitação é o custo de licenciamento, que pode restringir instituições e pesquisadores independentes e estimula o uso de alternativas de código aberto, como o Scilab. Contudo, as toolboxes especializadas do Matlab, como o "UAV" e o "Deep Learning", não têm contrapartes diretas no Scilab, o que limita o uso em projetos complexos. Soma-se a isso a dependência de versões mínimas do *software* para o uso pleno de certas toolboxes, aspecto relevante para a manutenção de projetos a longo prazo [1].

Conclusões

Esse estudo confirmou que o Matlab/Simulink continua sendo uma das principais ferramentas para modelagem, simulação e controle de sistemas dinâmicos. Os artigos analisados abrangem desde o controle de pêndulos invertidos por RNPs até a modelagem cinemática de manipuladores e o processamento de sinais inerciais, evidenciando a adaptabilidade da ferramenta a diferentes sistemas e fases de projeto. O custo da ferramenta ainda é relevante. Assim, para trabalhos futuros, sugere-se um estudo comparativo prático entre Matlab/Simulink e o Scilab/Xcos sobre um mesmo problema de controle.

Referências

1. Horri, N.; Pietraszko, M. A Tutorial and Review on Flight Control Co-Simulation Using Matlab/Simulink and Flight Simulators. Automation, v. 3, n. 3, p. 486-510, 2022.
2. OGATA, K. Engenharia de Controle Moderno. 5. ed. São Paulo, Pearson Prentice Hall, 2010.
3. Kozák, Š. State-of-the-art in control engineering. Journal of Electrical Systems and Information Technology, v. 1, n. 1, p. 1-9, 2014.
4. Page, M. J. et al. The PRISMA 2020 statement: an updated guideline for reporting systematic reviews. BMJ, v. 372, n. 71, 2021.
5. Carvalho Jr., A. de et al. Rotary Inverted Pendulum Identification for Control by Paraconsistent Neural Network. IEEE Access, 2021.
6. Carvalho, A. et al. Paraconsistent State Estimator for a Furuta Pendulum Control. SN Computer Science, v. 4, n. 29, 2023.
7. Pereira, A. R. et al. Dobot Manipulator Kinematics Modeling with MATLAB. Anais do Workshop de Micro-ondas (ADW), I(VIII), out. 2024a.
8. Santos, L. B. et al. Image Recognition Using Artificial Intelligence for a Safety System of the Dobot Magician. Anais do Workshop de Micro-ondas (ADW), I(VIII), out. 2024.

9. Pereira, A. R. et al. Comparison of PAL2v Filter and Kalman Filter Used in MPU6050 Accelerometer Gyroscope Sensor. Anais do Workshop de Micro-ondas (ADW), I(VIII), out. 2024b. DOI: 10.5281/zenodo.15459963.
10. Ucchas, M. T. W. et al. Performance and Comparative Analysis of PI and PID Controller-based Single Phase PWM Inverter Using MATLAB Simulink for Variable Voltage. In: 2022 Sec. Int. Conf. on Advances in Electrical, Computing, Communication and Sustainable Technologies (ICAECT). IEEE, 2022. p. 1-6.