

Embarque de Rede Neural Recorrente Fenomenologicamente Informada como Analisador Virtual para Sistemas Dinâmicos

Silas Henrique Alves Araújo* Leonardo Silva de Souza*
Raony Maia Fontes* Márcio André Fernandes Martins*

* Escola Politécnica, Universidade Federal da Bahia, BA, (e-mail:
silasaraujo@ufba.br, lssouza@ufba.br, raony@ufba.br,
marciomartins@ufba.br)

Abstract: The use of Physics-Informed Recurrent Neural Networks (PIRNNs) offers a promising framework for modeling dynamic systems. They combine traditional machine learning with the system's phenomenological model by incorporating differential equations directly into the cost function. This work presents the deployment of a PIRNN on a low-cost microcontroller to operate as a virtual analyzer, tested in a software-in-the-loop environment for a system of cascading spherical tanks. The results demonstrate reductions of over three times in computation time and the ability to provide information in the absence of measurements. These findings suggest that PIRNNs can be an efficient solution for systems requiring real-time control and process analysis applications.

Resumo: O uso de Redes Neurais Recorrentes Fenomenologicamente Informadas (PIRNNs) tem se mostrado promissor na modelagem de sistemas dinâmicos. Elas combinam o aprendizado de máquina tradicional com o modelo fenomenológico do sistema incorporando equações diferenciais diretamente na função de custo. Este trabalho apresenta o embarque de uma PIRNN em um microcontrolador de baixo custo para operar como analisador virtual, testado em ambiente *software-in-the-loop* para um sistema de tanques esféricos em cascata. Os resultados demonstram reduções superiores a três vezes no tempo de cômputo e a capacidade de fornecer informações na ausência de medições. Esses achados sugerem que a PIRNN pode ser uma solução eficiente para sistemas que exigem controle em tempo real e aplicações de análise de processos.

Keywords: Physics-Informed Recurrent Neural Networks; Simulation of Nonlinear Systems; Low-Cost Microcontrollers; Virtual Analyzer; Software-in-the-loop.

Palavras-chave: Redes Neurais Recorrentes Fenomenologicamente Informadas; Simulação de Sistemas Não Lineares; Microcontroladores de Baixo Custo; Analisador Virtual; Software-in-the-loop.

1. SOBRE O AUTOR

Silas Henrique Alves Araújo cursa Engenharia de Controle e Automação de Processos na Universidade Federal da Bahia (UFBA). Tendo participado como diretor de *marketing* na OPTIMUS Jr., onde desenvolveu um novo site e ajudou a estruturar e manter a identidade visual da empresa. Atualmente, participa do projeto PRH 35.1, visando o desenvolvimento de estratégias para sistemas relacionados ao petróleo e gás usando técnicas de otimização e ciência de dados.

Possui formação técnica em Eletromecânica pelo Instituto Federal da Bahia (IFBA) — Campus Simões Filho, onde também atuou como monitor no projeto de Robótica Educacional, onde aplicou conceitos de programação e eletrônica em projetos com Arduino, além de ministrar um curso de robótica para os estudantes.

2. INTRODUÇÃO

A simulação e o controle de processos são fundamentais para a indústria, desempenhando um papel essencial na eficiência e segurança dos processos. Através da simulação, é possível prever o comportamento de sistemas complexos sem a necessidade de intervenções físicas, o que permite identificar problemas, testar soluções e otimizar processos antes comprometer recursos físicos. O controle, por sua vez, assegura que os sistemas operem nos parâmetros estabelecidos, ajustando variáveis em tempo real para manter a estabilidade e a produtividade.

O trabalho conjunto entre simulação e controle é particularmente bem exemplificado no controle preditivo baseado em modelo (*Model Predictive Control* — MPC), uma abordagem avançada que tem sido amplamente aplicada em indústrias de processos, como químicas e petroquímicas, desde o final da década de 1970 (García et al., 1989). Nesse contexto, o uso de um modelo preciso do sistema é fundamental, por permitir prever a resposta futura do

processo com base em seu estado atual e nas possíveis ações de controle.

Entretanto, na prática, a indústria enfrenta uma série de desafios práticos para implementação de um sistema assim. Por exemplo, é preciso que a resposta da simulação seja rápida o suficiente para fornecer previsões em tempo real para o controlador tomar decisões eficazes a cada novo ciclo. Dependendo do modelo utilizado, isso exige alto poder computacional, o que pode ser um problema para sistemas embarcados.

Uma abordagem diferente para esses problemas é o uso de redes neurais artificiais (*Artificial Neural Networks* — ANNs). Elas são consideradas aproximadoras de funções universais (Csáji, 2001) e têm o potencial de serem mais rápidas que os métodos numéricos convencionais por terem uma arquitetura baseada em operações simples e paralelizáveis. Em adendo, as ANNs podem ser treinadas integrando o modelo matemático do problema diretamente na função de custo penalizando previsões que violam as relações físicas conhecidas. Essa metodologia, denominada Rede Neural Fenomenologicamente Informada (*Physics-Informed Neural Network* — PINN), é especialmente vantajosa para simular sistemas complexos e com poucos dados experimentais por combinar a flexibilidade das redes neurais com o rigor das equações diferenciais do problema físico (Raissi et al., 2019).

Uma variação dessa abordagem são as Redes Neurais Recorrentes Fenomenologicamente Informadas (*Physics-Informed Recurrent Neural Networks* — PIRNNs), que aprimoram a capacidade de capturar dependências temporais, sendo particularmente úteis para prever a dinâmica de sistemas que evoluem temporalmente ((Zheng et al., 2023)).

Essa pesquisa tem como objetivo explorar o uso de PIRNNs como uma alternativa aos métodos convencionais na simulação de sistemas dinâmicos. Especificamente, visa-se avaliar a redução no tempo de cômputo e a qualidade das previsões em comparação com os métodos numéricos tradicionais. Além disso, investiga-se a aplicabilidade dessas redes em sistemas embarcados, demonstrando a viabilidade de embarcar uma PIRNN em um microcontrolador de baixo custo.

3. METODOLOGIA

3.1 Descrição do Sistema e Modelagem matemática

O sistema utilizado neste trabalho, conforme descrito por Wildson (2024), é composto por dois tanques esféricos interligados, como mostrado na Figura 1. Ambos sujeitos à pressão atmosférica. A dinâmica do sistema inclui uma entrada de fluido para o tanque 1 e uma saída de fluido do tanque 2, cuja vazão depende diretamente do nível do líquido no segundo tanque. Para simplificar as equações, assume-se que o líquido é incompressível e possui massa específica constante.

A modelagem baseia-se no princípio de conservação de massa, onde a taxa de variação do volume de líquido em cada tanque é dada pela diferença entre a vazão de entrada q_{in} e de saída q_{out} , conforme a equação:

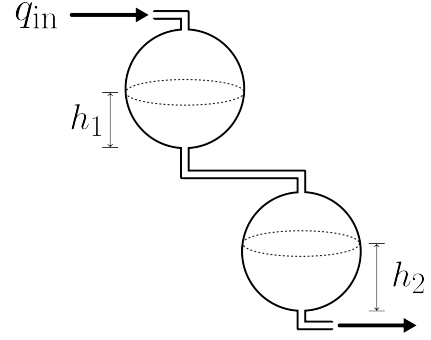


Figura 1. Representação esquemática do sistema de dois tanques esféricos.

$$\frac{dV(t)}{dt} = q_{in} - q_{out} \quad (1)$$

Para expressar a equação 1 em termos de nível de líquido, é necessário dividir ambos os lados pela área da seção transversal do tanque, que, no caso de um tanque esférico, é uma função não linear da altura do líquido.

Como o sistema trata de dois tanques interligados, a vazão de saída do tanque 1 torna-se a vazão de entrada do tanque 2. Assim, a dinâmica de um tanque afeta diretamente o outro. Com isso, obtemos o sistema de equações diferenciais:

$$\begin{cases} \frac{dh_1(t)}{dt} = \frac{q_{in} - \alpha_1 s_1 \sqrt{2gh_1}}{\pi(2Rh_1 - h_1^2)} \\ \frac{dh_2(t)}{dt} = \frac{\alpha_1 s_1 \sqrt{2gh_1} - \alpha_2 s_2 \sqrt{2gh_2}}{\pi(2Rh_2 - h_2^2)} \end{cases} \quad (2)$$

que descrevem a variação temporal do nível de líquido em cada tanque.

O valor e o significado de cada um dos parâmetros e constantes utilizados na modelagem, como área da seção transversal da tubulação e coeficientes relacionados à geometria dos tanques, são descritos com seus valores na Tabela 1.

Tabela 1. Constantes do sistema de equações 2.

Símbolo	Descrição	Valor
α_1	Fator de correção 1	0.56
α_2	Fator de correção 2	0.30
s_1	Área da seção de saída do tanque 1	0.50 cm^2
s_2	Área da seção de saída do tanque 2	0.50 cm^2
g	Aceleração da gravidade	980.665 cm/s^2
R	Raio dos tanques	14.85 cm

3.2 Desenvolvimento e Estrutura da Rede Neural

As ANNs são modelos computacionais inspirados no funcionamento do cérebro humano. Elas consistem em camadas compostas de neurônios artificiais, elementos de processamento capazes de receber entradas e produzir uma saída com base em funções matemáticas. Cada neurônio realiza operações simples sobre suas entradas e as transmite para os neurônios da camada seguinte, permitindo a construção de representações mais complexas à medida que os dados passam pelas camadas da rede neural (Mu and Sun, 2022). Elas são amplamente utilizadas na indústria química por sua capacidade de processar dados

complexos e não-lineares comuns a essa indústria, tudo isso sendo consideradas modelos do tipo “caixa preta”, onde as relações matemáticas entre variáveis não precisam ser conhecidas de antemão. Isso é especialmente útil quando não se tem um bom modelo matemático para o sistema, por ser possível construir modelos baseados unicamente em dados (Wang and Chen, 2022).

Entretanto, os modelos puramente baseados em dados apresentam limitações consideráveis. Para que esses modelos sejam eficazes, eles demandam um grande volume de dados representativos e abrangentes, algo nem sempre viável. Além disso, eles carecem de capacidade para extrapolar além das condições observadas nos dados de treinamento. Outro ponto crítico é que esses modelos podem gerar previsões inconsistentes com o conhecimento científico existente, pois não consideram as leis físicas que governam o sistema (Karniadakis et al., 2021). Então, para superar essas limitações e aproveitar o conhecimento físico disponível, surgiram as PINNs, introduzidas por Raissi et al. em 2017. Elas funcionam de forma parecida com as redes neurais artificiais tradicionais, com a diferença de que, além do custo associado aos dados, a função de custo também considera o erro relacionado ao problema físico, ou seja, a rede neural é penalizada ao diferir das relações físicas já conhecidas.

Da mesma maneira, as PIRNNs foram propostas como uma extensão das PINNs, combinando sua capacidade de incorporar conhecimento físico com a habilidade das redes neurais recorrentes (*Recurrent Neural Networks* — RNNs) de processar dados sequenciais e capturar dependências temporais eficientemente (Zheng et al., 2023).

No caso da PIRNN utilizada neste trabalho, as entradas são compostas por dois valores anteriores dos estados do sistema (h_1 , h_2) e da vazão de entrada q_{in} , enquanto a saída da rede corresponde aos próximos valores dos níveis dos tanques. Dependendo da aplicação, a entrada pode ser composta por um valor anterior e um atual ($t-1$ e t) para a rede retornar o valor futuro ($t+1$), ou, como foi utilizado neste trabalho, dois pontos anteriores ($t-2$ e $t-1$) para retornar o valor presente (t). A arquitetura da rede é composta por uma camada recorrente do tipo *Elman* (Elman, 1990) seguida por camadas lineares totalmente conectadas. Para treiná-la, a função de custo adotada é composta pela soma do custo associado as equações diferenciais (L_{EDOs}) e o custo relacionado aos dados de treinamento (L_{data}). Os dados utilizados foram obtidos por meio de simulações computacionais com métodos numéricos tradicionais, uma vez que a física do problema está bem definida e permite gerar dados confiáveis para a modelagem. No total, a função de custo é definida como:

$$L = \omega_1 \cdot L_{EDOs} + \omega_2 \cdot L_{data}, \quad (3)$$

onde

$$L_{EDOs} = \frac{1}{N} \sum_{i=1}^N \left(\frac{dy_{prev,i}}{dt} - f(y_{prev,i}) \right)^2 \quad (4)$$

$$L_{data} = \frac{1}{N} \sum_{i=1}^N (y_{data,i} - y_{prev,i})^2$$

Sendo os pesos ω_1 e ω_2 coeficientes ajustáveis que determinam a importância relativa de cada termo na função de custo, y_{data} os dados obtidos pela simulação, y_{prev} os

valores de saída previstos pela rede neural e f o sistema de equações diferenciais 2. A derivada em relação ao tempo foi calculada por meio do método das diferenças finitas com 3 pontos, considerando os dois pontos fornecidos para a rede e o ponto previsto pela rede.

O treinamento da rede foi realizado utilizando o otimizador *Adam* no framework PyTorch (Kingma and Ba, 2017; Ansel et al., 2024). Para otimizar a inferência em tempo de execução, o modelo treinado foi convertido para o formato ONNX (*Open Neural Network Exchange*), e os testes de velocidade foram realizados com o framework *ONNX Runtime* (ONNX Runtime developers, 2021).

3.3 Implementação no Arduino

Criado em 2005, o Arduino é uma plataforma prototipagem amplamente utilizada por programadores devido ao seu baixo custo e facilidade para desenvolver. A plataforma conta com uma série de placas diferentes, cada uma com características específicas, para atender a diversas necessidades e aplicações. Entre as placas mais populares, destaca-se o Arduino UNO, uma das primeiras e mais acessíveis opções disponíveis e que foi a escolhida para esse trabalho (Hughes, 2016).

No entanto, o Arduino é bem limitado em termos de capacidade de memória e processamento. O modelo UNO, possui 32 kilobytes de memória flash e apenas 2 kilobytes de memória RAM, o que dificulta bastante a tarefa de implementar algoritmos mais complexos, como redes neurais que possuem muitas camadas. Nesse trabalho, para garantir que o código fosse compatível com as restrições do hardware, foram feitas otimizações para reduzir o uso de memória e armazenamento. Dentre elas, vale a pena destacar o uso da memória flash para armazenamento dos pesos e bias da rede neural por meio da diretiva *PROGMEM* (Margolis et al., 2020). Isso é possível, pois esses valores são constantes, uma vez que foram definidos na etapa de treinamento da rede.

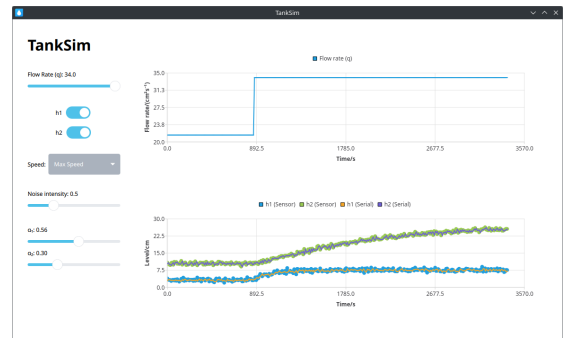


Figura 2. Captura de tela do TankSim.

Para realizar a comunicação com o Arduino, foi desenvolvido o *software TankSim* (Figura 2), responsável por simular o funcionamento do sistema, enviar os dados simulados para o Arduino e receber os valores previstos pela rede neural, além de plotar os gráficos comparando os dados simulados com os previstos. A interface conta ainda com um controle deslizante que permite o usuário alterar a vazão de entrada q_{in} , dois interruptores para suspender o envio dos dados simulados do nível dos tanques para o Arduino e outro controle deslizante para controlar o

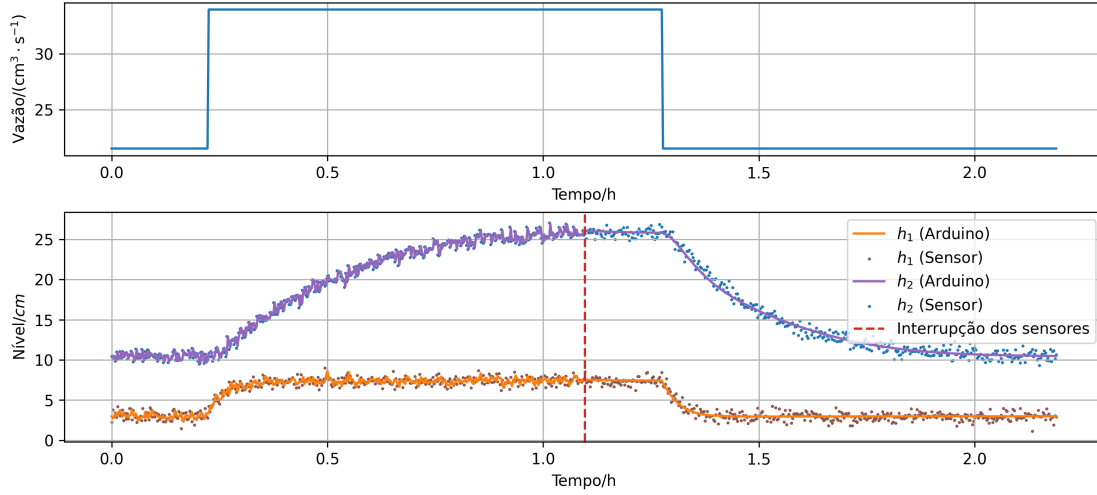


Figura 3. Comparação entre as leituras dos sensores, obtidas a partir dos níveis simulados com adição de ruído, e os níveis previstos pela PIRNN embarcada após uma perturbação do tipo degrau na vazão de entrada. A linha tracejada vermelha vertical indica o instante em que os valores dos sensores deixam de ser enviados ao Arduino, fazendo com que a PIRNN passe a se retroalimentar.

nível de ruído. A interface foi programada utilizando o framework Qt (<https://www.qt.io/>), que oferece uma ampla gama de componentes e ferramentas para a criação de interfaces.

No total, este sistema configura-se como um esquema *software-in-the-loop* (SIL) (Chen et al., 2008). Nele, o *TankSim* simula o envio de dados dos sensores de uma planta para o Arduino, que, por sua vez, responde com os dados previstos pela rede neural, ou seja, os prováveis valores para o nível de líquido em cada tanque no próximo intervalo de tempo. Quando os dados dos sensores não estão disponíveis, simulando uma falha ou interrupção manual por meio dos interruptores da interface, o Arduino entra em um modo de retroalimentação. Nesse modo, ele utiliza os valores previstos pela rede neural como entrada para gerar a previsão do próximo valor, mantendo a operação do sistema mesmo com os sensores indisponíveis.

4. RESULTADOS E DISCUSSÃO

Os resultados obtidos demonstram ser possível embarcar PIRNNs em plataformas de baixo custo com eficiência e robustez. Essa compatibilidade torna essa abordagem particularmente atrativa para aplicações industriais e acadêmicas que possuem restrições orçamentárias, permitindo a implementação de soluções de controle e simulação em larga escala sem a necessidade de hardware especializado ou de alto custo. Outro fator positivo foi o espaço ocupado pelo programa: verificou-se que o programa completo, que inclui tanto o modelo quanto o código responsável pela comunicação serial com o computador, utilizou apenas 68,4% da memória flash e 13,2% da memória RAM do Arduino. Esses números indicam que existem recursos disponíveis para implementação de modelos mais complexos ou execução de tarefas adicionais, caso necessário.

Nos testes de robustez, foram simulados cenários com ruído nas medições e com ausência completa de medições. A Figura 3 apresenta os resultados obtidos nesses testes. Observa-se que, no caso de medições ruidosas, o modelo

capta as variações impostas pelo ruído, mas com um leve amortecimento, sugerindo uma capacidade de filtragem parcial das perturbações. Quando as medições são interrompidas completamente, o modelo passa a se retroalimentar. Nesse cenário, o modelo continua seguindo com precisão o comportamento esperado dos tanques, mesmo sem a entrada de dados dos sensores.

Nos testes de desempenho, a PIRNN demonstrou uma significativa vantagem em termos de velocidade quando comparada com métodos numéricos tradicionais. A comparação foi feita com os métodos RK, CVODES e IDAS da biblioteca *CasADi*, e os métodos RK45, RK23 e LSODA da biblioteca *SciPy*, todos executados no mesmo computador. Para avaliar o desempenho de cada método, foram realizadas 100 medições do tempo necessário para simular aproximadamente 33 horas de operação, considerando intensas perturbações na vazão de entrada. A Figura 4 apresenta o boxplot dos tempos de execução dos métodos avaliados, evidenciando a superioridade da PIRNN em termos de tempo de processamento.

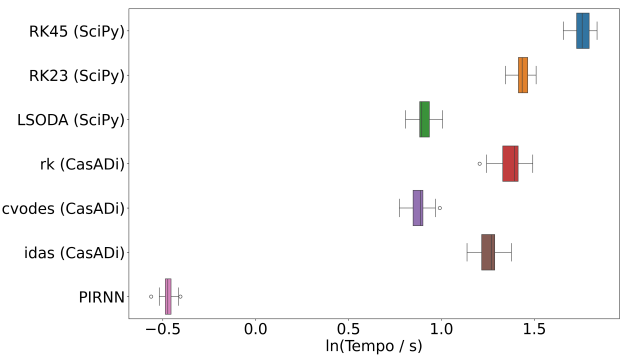


Figura 4. Boxplot dos tempos de execução em escala de logaritmo natural dos métodos avaliados.

Esses resultados mostram que o uso de PIRNNs embarcadas pode ser uma alternativa viável aos métodos tradicionais na simulação e controle de sistemas. A arquitetura proposta mostrou-se capaz de fornecer respostas rápida-

mente, o que é essencial para aplicações de controle em tempo real. Além disso, os testes de robustez demonstraram que a PIRNN manteve a qualidade das previsões mesmo em cenários adversos, como a presença de ruídos ou ausência de medições.

5. CONSIDERAÇÕES FINAIS

Neste trabalho, foi demonstrada a utilização de uma PIRNN embarcada em um Arduino UNO para simular um sistema dinâmico formado por dois tanques esféricos em cascata. Os resultados mostraram que a PIRNN é uma alternativa viável aos métodos numéricos tradicionais e apresenta desempenho superior em relação ao tempo de simulação, sem grandes prejuízos a fidelidade das previsões. Isso a torna adequada para o controle em tempo real, sendo aplicável em automação industrial, monitoramento de processos e como analisador virtual. Sua compatibilidade com plataformas de baixo custo, como o Arduino, torna-a acessível e robusta, mantendo boa desempenho mesmo com falhas ou ruídos nas medições.

Contudo, em sistemas dinâmicos reais, é necessário considerar a degradação dos parâmetros do modelo ao longo do tempo, causada pelo desgaste ou envelhecimento dos componentes físicos. Essa degradação pode comprometer o desempenho da PIRNN, tornando necessário o retreinamento periódico do modelo para garantir sua acurácia. Nesse contexto, estudos futuros podem explorar técnicas de aprendizado por reforço, visando a adaptação em tempo real da PIRNN às mudanças dos parâmetros do sistema sem a necessidade de intervenções manuais.

Por fim, pesquisas adicionais podem investigar a implementação de modelos mais complexos, visando expandir a aplicabilidade das PIRNNs em sistemas dinâmicos descritos por equações diferenciais parciais.

AGRADECIMENTOS

Os autores gostariam de agradecer o suporte financeiro da ANP no âmbito do PRH 35.1.

REFERÊNCIAS

- Ansel, J., Yang, E., He, H., Gimelshein, N., Jain, A., Voznesensky, M., Bao, B., Bell, P., Berard, D., Burovski, E., Chauhan, G., Chourdia, A., Constable, W., Desmaison, A., DeVito, Z., Ellison, E., Feng, W., Gong, J., Gschwind, M., Hirsh, B., Huang, S., Kalambarkar, K., Kirsch, L., Lazos, M., Lezcano, M., Liang, Y., Liang, J., Lu, Y., Luk, C. K., Maher, B., Pan, Y., Puhrsch, C., Reso, M., Saroufim, M., Siraichi, M. Y., Suk, H., Zhang, S., Suo, M., Tillet, P., Zhao, X., Wang, E., Zhou, K., Zou, R., Wang, X., Mathews, A., Wen, W., Chanan, G., Wu, P., Chintala, S., 2024. Pytorch 2: Faster machine learning through dynamic python bytecode transformation and graph compilation. In: Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2. ASPLOS '24. Association for Computing Machinery, New York, NY, USA, p. 929–947.
- Chen, X., Salem, M., Das, T., Chen, X., 2008. Real time software-in-the-loop simulation for control performance validation. *SIMULATION* 84 (8-9), 457–471.
- Csáji, B., 06 2001. Approximation with artificial neural networks. Ph.D. thesis, Eötvös Loránd University.
- Elman, J. L., 1990. Finding structure in time. *Cognitive Science* 14 (2), 179–211.
- García, C. E., Prett, D. M., Morari, M., 1989. Model predictive control: Theory and practice—a survey. *Automatica* 25 (3), 335–348.
- Hughes, J. M., May 2016. Arduino: A Technical Reference. “O’Reilly Media, Inc.”.
- Karniadakis, G. E., Kevrekidis, I. G., Lu, L., Perdikaris, P., Wang, S., Yang, L., Jun. 2021. Physics-informed machine learning. *Nature Reviews Physics* 3 (6).
- Kingma, D. P., Ba, J., 2017. Adam: A method for stochastic optimization.
- Margolis, M., Jepson, B., Weldin, N. R., Apr. 2020. Arduino Cookbook. “O’Reilly Media, Inc.”.
- Mu, Y., Sun, L., Jan. 2022. Catalyst optimization design based on artificial neural network. *Asian Journal of Research in Computer Science*.
- ONNX Runtime developers, 2021. Onnx runtime. <https://onnxruntime.ai/>, version: 1.20.0.
- Raissi, M., Perdikaris, P., Karniadakis, G. E., 2017. Physics informed deep learning (part i): Data-driven solutions of nonlinear partial differential equations. *arXiv preprint arXiv:1711.10561*.
- Raissi, M., Perdikaris, P., Karniadakis, G. E., 2019. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics* 378, 686–707.
- Wang, H., Chen, Y., Jun. 2022. Application of artificial neural networks in chemical process control. *Asian Journal of Research in Computer Science* 14 (1), 22–37.
- Wildson, 2024. Tcc - trabalho de conclusão de curso, trabalho não publicado.
- Zheng, Y., Hu, C., Wang, X., Wu, Z., 2023. Physics-informed recurrent neural network modeling for predictive control of nonlinear processes. *Journal of Process Control* 128, 103005.