

Identificação de um sistema de tanques acoplados usando redes LSTM

Kaique G. Machado * Wellington S. Santos * Daniel L. Martins *
Adrião D. D. Neto *

* *Laboratório de Instrumentação Inteligente Departamento de Engenharia de Computação e Automação Universidade Federal do Rio Grande do Norte Campus Universitário Lagoa Nova CEP 59078-970 Natal, Rio Grande do Norte, Brasil (E-mails: kaique.machado.099@ufrn.edu.br, wellington.santos.087@ufrn.edu.br, danlartin@dca.ufrn.br, adriao@dca.ufrn.br).*

Abstract: Dynamic systems models are essential to perform tests and determine behavior of real systems in different situations, but their determination is often complex. The present work presents the use of an LSTM neural network (*Long-Short Term Memory*) to identify a tank system. The results of the training of two models are presented, one with a single neural network and the other with a split neural network and the results demonstrate that for the problem addressed, the use of the model with a split neural network can meet the purpose of simulating the system of tanks with a low error (on the order of a few millimeters) between the real value and the one estimated by the model, both for the prediction and simulation approaches.

Resumo: Modelos de sistemas dinâmicos são essenciais para realizar testes e determinar comportamentos de sistemas reais em diversas situações, mas a sua determinação muitas vezes é complexa. O presente trabalho apresenta o uso de uma rede neural LSTM (*Long-Short Term Memory*) para identificar um sistema de tanques. São apresentados os resultados do treinamento de dois modelos, um com uma rede neural única e outro com uma rede neural dividida e os resultados demonstram que para o problema abordado, o uso do modelo com rede neural dividida consegue atender o propósito de simular o sistema de tanques com baixo erro (na ordem de poucos milímetros) entre o valor real e o estimado pelo modelo, tanto para a abordagem de predição quanto de simulação.

Keywords: Wireless Industrial Sensor Networks; Artificial neural networks; LSTM; Identified Systems; Dynamic Models; Coupled Tank System.

Palavras-chaves: Redes de Sensores Industriais Sem Fio; Redes Neurais Artificiais; LSTM; Sistemas Identificados; Modelos Dinâmicos; Sistema de Tanques Acoplados.

1. INTRODUÇÃO

A modelagem de sistemas é muito importante para prever o comportamento futuro de sistemas dinâmicos, para aplicações de controle baseados em modelos e para entender processos físicos, sendo encontrada em várias áreas da ciência (Gonzales e Yu, 2018).

Porém, a modelagem de sistemas é uma tarefa complexa, pois envolve conhecer as leis (físicas, químicas, e etc) que regem o sistema e para isso, utiliza-se a identificação. A identificação de um sistema visa, através de várias técnicas, construir um modelo que possa estimar um sistema físico (Wang, 2017).

Na teoria de controle de sistemas dinâmicos convencionais, há dois tipos de representação de sistemas: baseada em espaço de estados e baseada em entradas e saídas (Wang, 2017). No caso das representações baseadas em entradas e saídas, o problema da identificação é inferir a relação entre entradas passadas, com saídas futuras (Sjöberg et al., 1994; Forgione e Piga, 2021).

Os sistemas também podem ser divididos em lineares e não lineares. Muitos sistemas não lineares podem ser aproximados a sistemas lineares (com algumas restrições), mas existem sistemas inteligentes que podem ser utilizados para lidar com sistemas não lineares como sistemas *Fuzzy*, algoritmos genéticos, redes neurais (Kumar et al., 2019).

Um método conveniente é o uso de modelagem caixa-preta que somente utiliza os dados de entrada e saída do sistema, não se sabendo parâmetros que regem as leis físicas relacionadas ao sistema Sjöberg et al. (1994). As redes neurais são uma ferramenta de modelagem do tipo caixa-preta e, graças a sua característica de aproximador universal e aprendizagem automática, têm ganhado grande força em modelagem de sistemas (Gonzales e Yu, 2018). Comparado às abordagens tradicionais, as redes neurais artificiais, em especial as redes profundas, possuem algumas vantagens como extrapolação dos dados, adaptatividade, aproximação, trabalhar com alta dimensionalidade além de lidar com sistemas não lineares mais facilmente devido as suas próprias características não lineares (Sjöberg et al., 1994; Ogunmolu et al., 2016).

O treinamento de redes neurais utiliza basicamente o algoritmo *backpropagation* (retropropagação do erro) através da descida do gradiente. Uma rede de múltiplas camadas pode modelar sistemas contínuos, desde que a quantidade de neurônios utilizados seja grande o suficiente, mas isso leva ao problema do *overfitting* (sobreajuste em tradução literal), onde a rede “decora” o padrão a que foi apresentada no treinamento, diminuindo sua generalização para novos dados. Uma alternativa a esse problema é aumentar a quantidade de camadas ocultas, o que define as *deep neural networks*, ou rede profundas (Gonzales e Yu, 2018).

Por muitos anos, as redes neurais profundas tiveram seu desenvolvimento lento ou estagnado por dois problemas: poder computacional (para processar tantos parâmetros) e o esvaecimento do gradiente (durante a execução da retropropagação do erro) (Wang, 2017).

Com o advento de computadores mais potentes e o uso de placas de computação gráfica, que podem realizar processamento paralelo, o primeiro problema já foi resolvido. Já para o segundo problema, algumas soluções também foram desenvolvidas ao longo dos anos.

A rede LSTM (*Long-Short Term Memory*) é uma das redes recorrentes mais populares no campo de redes profundas e foi desenvolvida por Schmidhuber em 1997 (Wang, 2017). Ela consegue evitar o problema do esvaecimento do gradiente por utilizar uma estrutura interna com três portões: um portão de esquecimento, um de entrada e um de saída, onde através deles, pode-se controlar a passagem de estados passados dos neurônios durante o processo de aprendizagem (Wang, 2017; Gonzales e Yu, 2018).

Neste contexto, o presente trabalho propõe a utilização de uma rede neural LSTM para a identificação de um sistema de tanques acoplados. É assumido que não se sabe o modelo do sistema, trabalhando-o como uma caixa-preta.

A principal contribuição desse trabalho reside no estudo do problema proposto e da solução adotada, usando redes LSTM, para se chegar ao resultado obtido, um modelo com baixo erro predição.

1.1 Trabalhos Relacionados

O uso de redes neurais para a identificação de sistemas dinâmicos não é novo. Vários trabalhos já foram desenvolvidos usando diferentes tipos de redes.

Em Forgione e Piga (2021) os autores fazem uma nova proposta de redes neurais feita sob medida com critérios customizados para a aprendizagem de sistemas dinâmicos.

Já em Gonzales e Yu (2018), os autores propõem uma rede LSTM modificada (com o acréscimo de uma camada de rede neural comum) para resolver o problema (de não correlacionar bem as entradas com as saídas) das redes LSTM com sistemas não lineares.

Kumar et al. (2019) faz uma análise de três tipos de redes neurais (uma rede de múltiplas camadas *feedforward*, uma rede recorrente diagonal e uma rede não linear autoregressiva com entradas exógenas - NARX) para a identificação de sistemas não lineares.



Figura 1. Equipamento utilizado: o sistemas de tanques e os sensores de nível sem fio.

Sjöberg et al. (1994) realiza um trabalho de descrever como redes neurais podem ser utilizadas para identificar sistemas.

Já em de Menezes et al. (2015) é proposto o uso de modelos baseados em redes neurais (rede perceptron de multicamadas) para a simulação da precipitação na irrigação por aspersão.

Em Machado et al. (2010) foi proposto uma arquitetura multi-agente baseado em redes neurais onde um dos agentes utilize-se de um sistema identificado para detectar falhas no processo.

Filho et al. (2010) implementou um preditor baseado em redes neurais utilizando blocos funcionais do protocolo *Foundation Fieldbus*.

Já Bezerra et al. (2013) implementou um preditor baseado em rede NARX e um classificador utilizando lógica *Fuzzy* dentro de instrumentos que utilizavam protocolo *Foundation Fieldbus*.

Como pode ser visto pelos trabalhos citados, todos utilizaram de uma maneira ou outra redes neurais para gerarem modelos de predição e identificação de algum sistema físico, o que também é o objetivo deste trabalho.

2. MODELO DE TANQUES E CAPTURA DE DADOS

2.1 Modelo de Tanques

O sistema a ser identificado é um kit da Quanser (Figura 1) composto por 2 tanques conectados, tubos de conexão, 1 bomba e 1 reservatório. O funcionamento do sistema é iniciado no momento em que a bomba capta a água do reservatório e a leva até o tanque 1 (tanque superior) que por sua vez leva a água, através de um orifício até o tanque 2 (tanque inferior) por gravidade. Por fim, a água do tanque inferior é repassada para o reservatório. Os sensores de níveis sem fio (utilizando protocolos *WirelessHart* e *ISA100.11a*) realizam a medição (em milímetros de água) através da pressão exercida pela coluna de água no fundo de cada tanque.

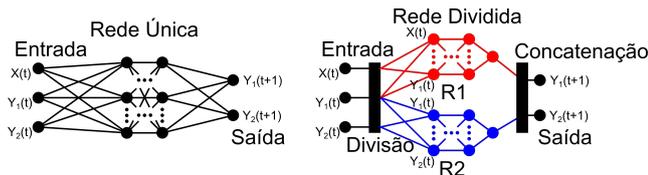


Figura 2. Redes neurais utilizadas no trabalho (sem os atrasos para simplificar).

2.2 Captura de Dados

A captura dos dados para as etapas de treino e teste da rede foi realizada com o microcontrolador *ESP32* e o *software Node-RED*. Os sensores sem fio transmitem os valores dos níveis a um *Gateway* que disponibiliza a informação para o ESP32, via o roteador sem fio do laboratório, utilizando o protocolo *Modbus*.

Além disso, o ESP32, também utilizando a rede sem fio, se comunica com o computador que roda o Node-Red via protocolo MQTT (*Message Queuing Telemetry Transport*). O código implementado no Node-RED envia a tensão desejada para o ESP que gera o sinal PWM (*Pulse Width Modulation*, ou modulação por largura de pulso) para a bomba do sistema de tanques, além de tratar a captura dos níveis e o salvamento das medições em um arquivo no formato CSV.

3. MODELO NEURAL, DADOS PARA TREINAMENTO, VALIDAÇÃO E TESTE

3.1 Modelo Neural

Para a execução do trabalho, foram criadas duas redes neurais do tipo LSTM (uma única e outra dividida) com 3 entradas (Tensão, X, Nível 1, Y1, e Nível 2, Y2) e 4 atrasos de tempo para cada entrada e 2 saídas (Nível 1 e Nível 2) do instante de tempo seguinte (como pode ser visto de forma simplificada na Figura 2. No caso das entradas, seus atrasos são referentes aos tempos t ($t-1$, $t-2$, $t-3$ e $t-4$) e no caso das saídas, são as previsões de tempo $t+1$. O motivo de se escolher a previsão de $t+1$ é que assim que o sistema captura os dados, ele tem todos os dados do tempo atual (t), tanto os níveis atuais quanto a tensão aplicada atualmente no sistema, então a previsão a partir deste estado é dos níveis que serão medidos na próxima amostragem ($t+1$).

A arquitetura da primeira rede (a rede única) é composta por 600 neurônios com uma camada LSTM e uma camada com função de ativação linear de duas saídas. Já a segunda rede treinada (a rede dividida) tem uma arquitetura de camadas internas diferente, composta por uma camada divisória, que divide as entradas em duas partes para alimentar duas redes LSTM com 10 neurônios cada, uma com as entradas de tensão e nível 1 (que faz a previsão do nível 1) e outra com as entradas de nível 1 e nível 2 (que faz a previsão do nível 2), ambas com saída linear. Por fim, uma última camada que concatena as redes para compor a estrutura LSTM explicada inicialmente, 3 entradas e 2 saídas.

O motivo da mudança entre a primeira e segunda rede treinada foi a análise do comportamento da primeira rede

que ao ser submetida ao laço que compõe a realimentação, que será explicada posteriormente, percebeu-se que havia um ruído no gráfico do nível 1 provocado pelos ruídos dos valores do nível 2. Com isso, foi constatado a necessidade de dividir o problema em duas redes LSTM, uma para cada nível.

3.2 Treinamento

O conjunto de dados de treinamento da rede única é constituído por 12 valores distintos de tensões que variam de 0.9 V a 1,9 V mais o valor de 0.0 V, com cada valor de tensão permanecendo por 5 minutos ativo, resultando em 18000 amostras como entrada da rede. A primeira rede foi treinada com as seguintes configurações: taxa de aprendizado de 10^{-3} , a métrica de perdas (*loss*) o erro médio quadrático (*MSE - Mean Squared Error*), o uso de 1000 épocas de treinamento com *batch* de 40 amostras e como métrica de validação a raiz quadrada do erro médio quadrático (*RMSE - Root Mean Squared Error*). Por fim, foi empregado uma divisão ao conjunto de dados em uma proporção de 70% para treino e 30% para validação.

A segunda rede foi treinada com as seguintes configurações: Um conjunto de treinamento de 24000 amostras, variando de 0.9 V a 1,9 V mais o valor de 0.0 V, cada valor de tensão permanecendo por 4 minutos ativo nas primeiras 12000 amostras e de 1 minuto nas 12000 seguintes. A taxa de aprendizado de 10^{-5} , a métrica de perdas utilizada foi o MSE, usando 20000 épocas de treinamento com *batch* de 32 amostras inicialmente (até 10000 épocas) e final de 8 amostras e como métrica de validação, o RMSE sobre um conjunto de validação independente de 720 amostras (3% do tamanho do conjunto de treinamento).

O motivo de conjuntos distintos para o treinamento das duas redes foi aumentar a variação da entrada da rede para verificar se haveria melhor captura de sua dinâmica. O primeiro conjunto possui um tempo mais alto (5 min) em cada tensão, e por isso, possui uma variação menor, dessa forma foi criado o segundo conjunto de treinamento.

3.3 Teste

O conjunto de dados de teste das redes é constituído de 10 valores de tensões distintas (exclui-se dos valores de tensão de 0.0 V e 1.9 V usados no treinamento) em que cada valor permanece 4 min em execução, totalizando 2400 amostras de teste para a rede.

4. RESULTADOS DO TREINAMENTO E DO TESTE

4.1 Rede única

Os resultados obtidos da rede única podem ser vistos na Tabela 1. A rede treinada conseguiu na métrica de perda (MSE) 0,4 sobre os dados de treinamento, o RMSE de 1,55 na validação e 2,02 no conjunto de teste. Na Figura 3 e na Figura 4, observamos a comparação entre previsão da saída da rede e valor real capturado dos níveis 1 e 2 utilizando o conjunto de teste (para o qual a rede não tinha sido apresentada na fase de treinamento). A rede treinada conseguiu capturar o comportamento do sistema acompanhando os regimes permanente e transitório.

Tabela 1. Resultados do treinamento e teste rede única.

Resultado	Valor
<i>Train Loss (MSE)</i>	0,40
<i>Train Metric (RMSE)</i>	0,63
<i>Validation Loss (MSE)</i>	2,41
<i>Validation Metric (RMSE)</i>	1,55
<i>Test Metric (RMSE)</i>	2,02

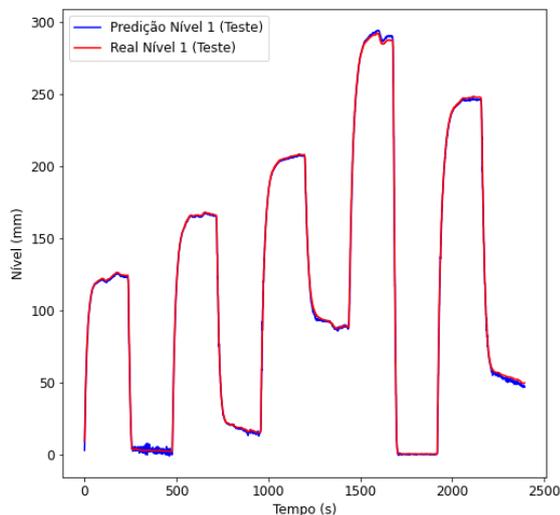


Figura 3. Saída da rede única (predição e real) do nível 1 com o conjunto de dados de teste.

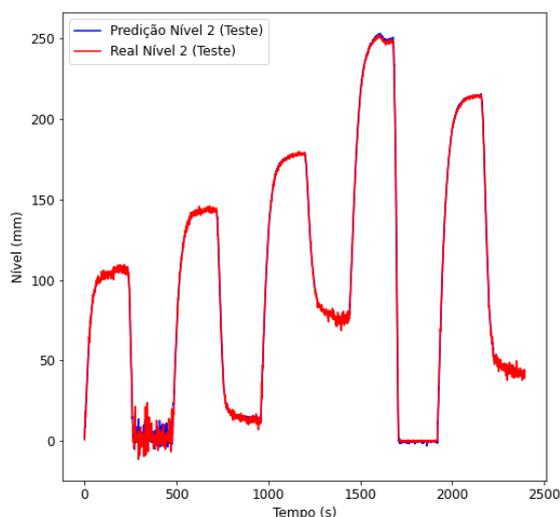


Figura 4. Saída da rede única (predição e real) do nível 2 com o conjunto de dados de teste.

As Figuras 5 e 6 exibem os gráficos de nível versus nível, respectivamente, para os níveis 1 e 2. Esse gráfico permite avaliar o quão longe ou perto os dados de predição estão dos dados capturados. Caso as predições sejam exatas, uma reta em 45° será observada. Quão pior for a previsão em relação ao dado real, mais distantes os valores estarão dessa reta.

Como é possível observar na Figura 5, ela se aproxima muito da reta de 45°, provando que ela conseguiu generalizar bem a predição sobre o conjunto de teste após treinamento. Já na Figura 6 é possível observar o mesmo

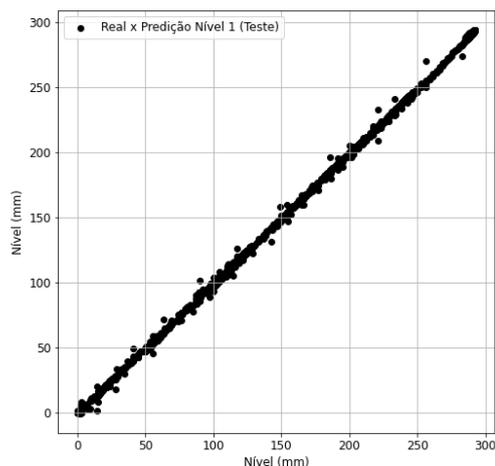


Figura 5. Saída da rede única (predição e real) do nível 1 com o conjunto de dados de teste.

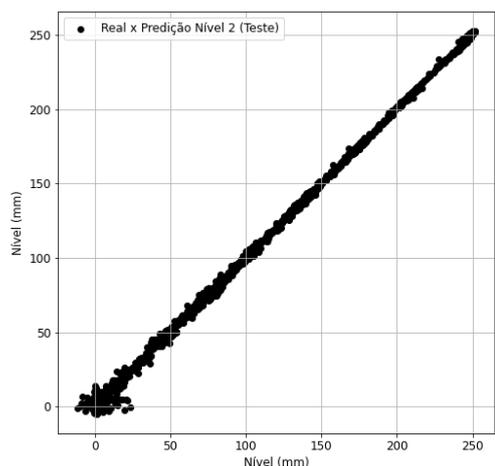


Figura 6. Saída da rede única (predição e real) do nível 1 com o conjunto de dados de teste.

comportamento, mas com a diferença nos valores próximos ao nível 0. Isso se deve principalmente ao ruído apresentado nos níveis mais baixos no tanque 2 (como pode ser visto na Figura 4 entre o tempo de 250 s e 500 s). Esse ruído surge por causa da forma como a água cai do primeiro para o segundo tanque, batendo diretamente no fundo e gerando ondulações e variações de pressão que são capturados pelo sensor de nível. Quando o nível está mais alto, essas ondulações diminuem pois a altura da queda da água diminui.

4.2 Rede Dividida

Em comparação com os resultados da rede única, os dados obtidos do treinamento da rede dividida (Tabela 2) foram inferiores, por exemplo, o valor da perda teve um aumento considerável, mas a métrica de validação (RMSE) teve uma leve redução. Contudo, a rede consegue generalizar

bem com os dados de teste, podendo ser percebido tanto no valor da métrica (inferior a da rede única) quanto nas Figuras 7 e 8.

Tabela 2. Resultados do treinamento da rede dividida

Resultado	Valor
<i>Train Loss (MSE)</i>	1,40
<i>Train Metric (RMSE)</i>	1,18
<i>Validation Loss (MSE)</i>	1,80
<i>Validation Metric (RMSE)</i>	1,34
<i>Test Metric (RMSE)</i>	1,60

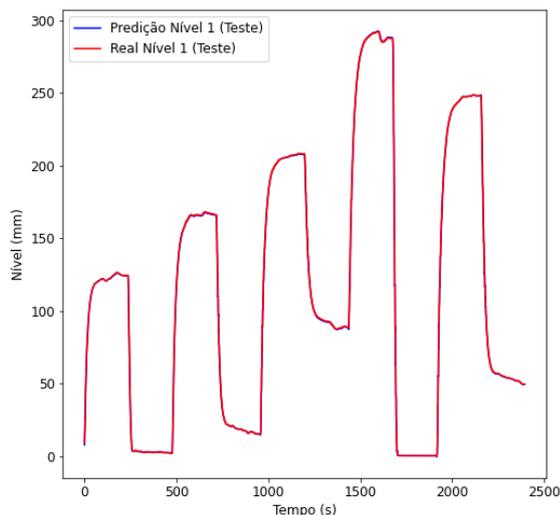


Figura 7. Saída da rede dividida (predição e real) do nível 1 com o conjunto de dados de teste.

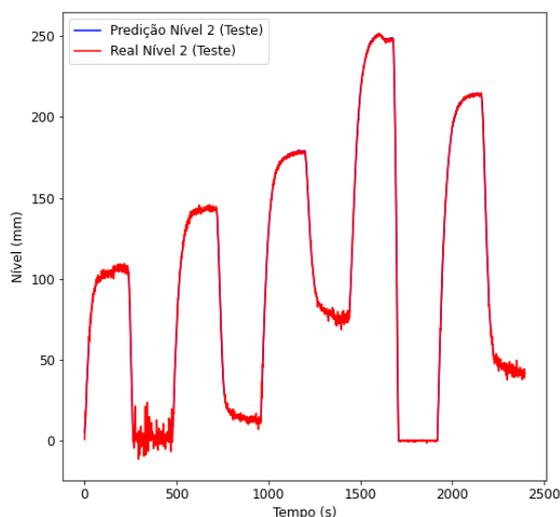


Figura 8. Saída da rede dividida (predição e real) do nível 2 com o conjunto de dados de teste.

As Figuras 9 e 10 exibem os gráficos de nível versus nível, respectivamente, para os níveis 1 e 2.

Observando ambas as Figuras 9 e 10 é possível constatar que a maioria dos valores ficam muito próximos a reta de 45°, sendo que na Figura 10 referente ao nível 2 é possível observar que os valores mais próximos do nível 0 ficam mais distantes da reta pelo mesmo motivo já explicado para a Figura 6

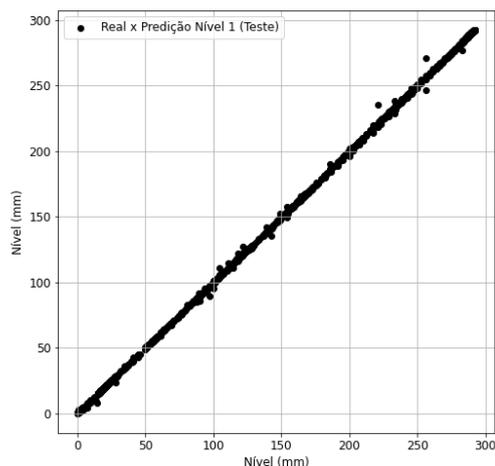


Figura 9. Gráfico capturado X predição da rede Dividida para o nível 1 sobre o conjunto de teste.

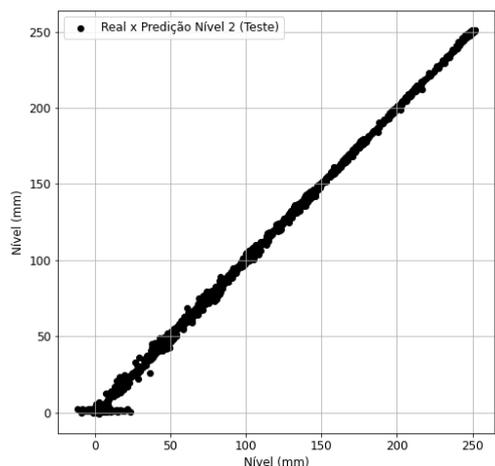


Figura 10. Gráfico capturado X predição da rede dividida para o nível 2 sobre o conjunto de teste.

5. MODELO REALIMENTADO

5.1 Modelo Neural

No modelo de simulação (Gonzales e Yu, 2018), as entradas, referentes aos dados capturados do sistema físico, são fornecidas pela saída do próprio modelo (as saídas previstas). O motivo de se utilizar o modelo dessa forma é que para o uso do modelo como simulador, você está tentando gerar dados de um sistema físico ao qual pode-se não ter acesso a medições (capturas) a todo momento.

Assim, se fez necessária a realimentação do modelo com suas próprias previsões. A execução desse sistema funciona em forma de *loop*, no qual é utilizado um laço de repetição para garantir essa realimentação. Ao fim de cada execução, a saída da rede é armazenada para ser utilizada como

entrada na próxima execução do laço (gerando um atraso de tempo na amostra). A Figura 11 exibe essa ideia.

A principal desvantagem desse tipo de simulação é que o erro da previsão é realimentado ao próprio modelo, o que pode acabar gerando uma previsão ainda mais errada da próxima amostra.

Na Figura 11 pode-se observar que os valores referentes a saída prevista pelo sistema se tornam os níveis do tanque 1 e 2 utilizados na próxima execução do modelo.

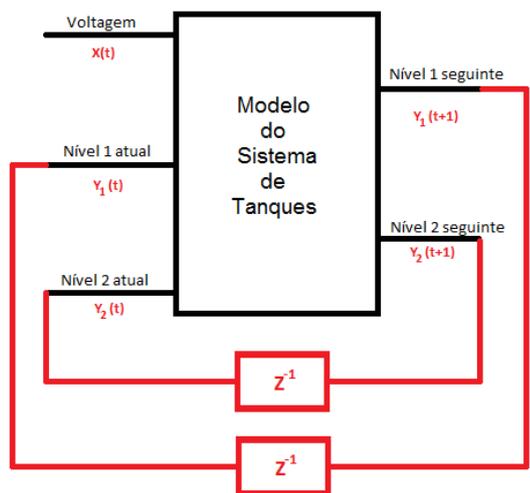


Figura 11. Sistema de realimentação dos tanques nível 1 e 2: Reutilização das saídas como entrada dos níveis.

6. RESULTADOS REDE REALIMENTADA

6.1 Gráficos Rede Única

Diferente do que ocorreu durante a fase de treinamento, no modelo realimentado a rede neural única apresentou um comportamento bem diferente, divergindo bastante dos valores esperados, como pode ser visto tanto nos valores de RMSE para ambos os níveis na Tabela 3 e os gráficos na Figura 12 para o nível 1 e na Figura 13 para o nível 2, que alcançaram dezenas de mm.

Tabela 3. Resultados da rede única no sistema realimentado

Resultado	Valor
Test Metric Lv1 (RMSE)	54,17
Test Metric Lv2 (RMSE)	45,62

As Figura 14 e 15 apresentam os gráficos dos dados capturados versus simulação. É possível ver o quão distante os dados ficaram da reta de 45°, o que indica o quão ruim ficou essa simulação, significando que a rede não conseguiu capturar tão bem a dinâmica do sistema (generalizar) como parecia na fase de treinamento.

Como esse modelo possuiu um erro de treinamento muito baixo, mas um erro relativamente maior para o conjunto de validação, isso pode indicar que ela entrou em *overfitting*. Como explicado anteriormente sobre a principal desvantagem da realimentação, por possivelmente estar em *overfitting* e o ruído do tanque 2 afetar também a previsão

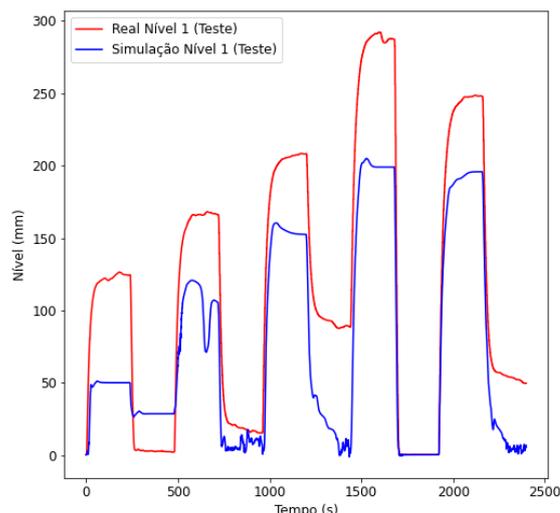


Figura 12. Saída da rede única (simulação e real) do nível 1.

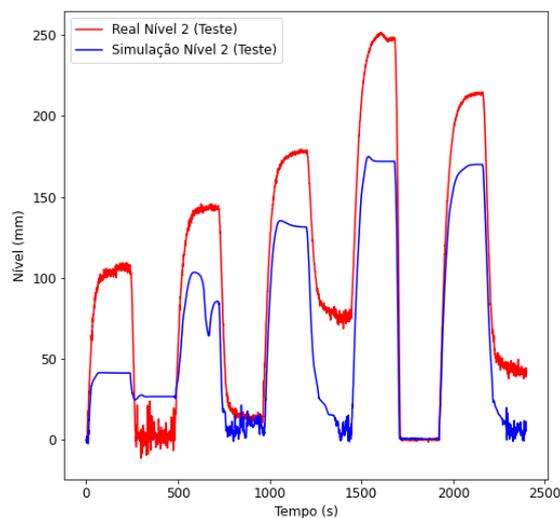


Figura 13. Saída da rede única (simulação e real) do nível 2.

do tanque 1, suas previsões conterão erros ainda maiores para valores de entradas diferentes daqueles treinados, gerando maiores erros de previsão.

6.2 Avaliação da Rede Dividida

Já para o modelo de rede dividida o seu nível de acerto se mostra bem superior, com erro na casa de poucos mm (como pode ser visto nos valores da Tabela 4 e nos gráficos das Figuras 16 e 17).

Tabela 4. Resultados da rede dividida no sistema realimentado

Resultado	Valor
Test Metric Lv1 (RMSE)	6,42
Test Metric Lv2 (RMSE)	5,42

Observando os gráficos dos dados capturados versus simulado (Figuras 18 e 19) é possível verificar que os resultados tendem a seguir a linha de 45°, o que indica a melhor generalização desse modelo, capturando bem a dinâmica do sistema.

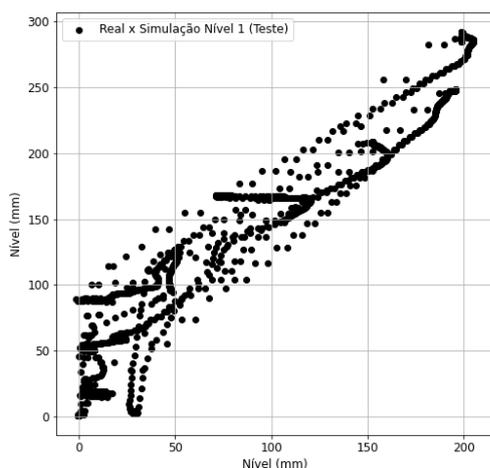


Figura 14. Gráfico capturado X simulação da rede única para o nível 1.

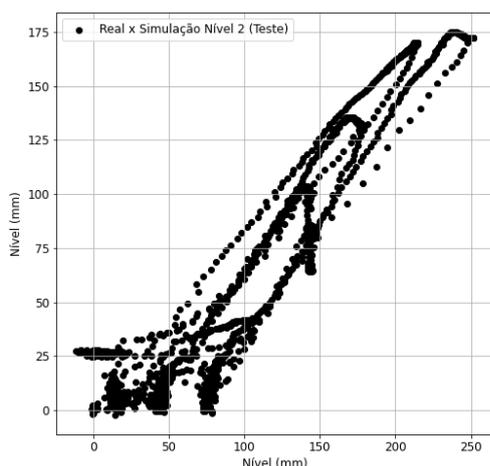


Figura 15. Gráfico capturado X simulação da rede única para o nível 2.

Obviamente, se comparados aos gráficos das Figuras 9 e 10, há uma perceptível piora com o afastamento dos valores da reta de 45° , mas isso já era esperado observando-se os gráficos das Figuras 16 e 17 e pelo que já foi explicado anteriormente sobre a desvantagem desse tipo de aplicação.

7. CONCLUSÃO

Como pode ser visto nos resultados, a identificação de um modelo através de uma rede neural pode apresentar bons resultados no treinamento (como modelo de predição), mas quando sua execução se dá como modelo de simulação, a rede pode apresentar resultados bem insatisfatórios.

O motivo disso reside no fato de que o erro de predição, que no modo de predição é pequeno, vai sendo realimentado e pode fazer com que no modo de simulação acabe por

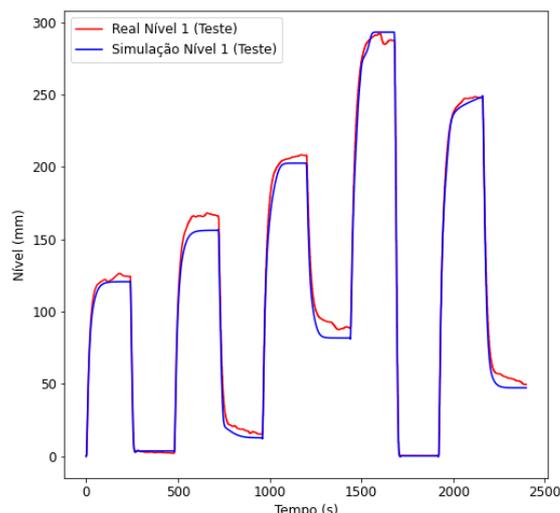


Figura 16. Saída da rede dividida (simulação e real) do nível 1.

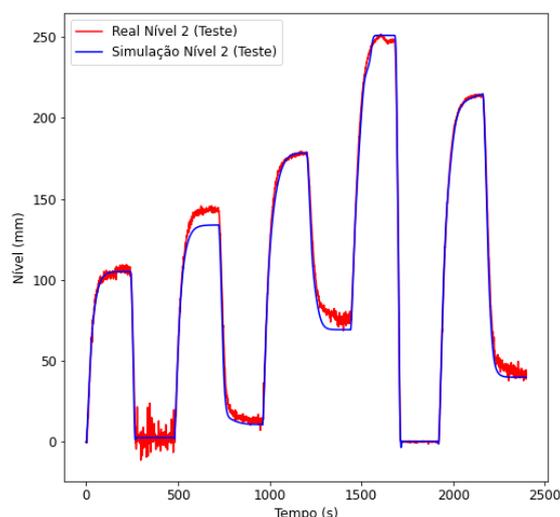


Figura 17. Saída da rede dividida (simulação e real) do nível 2.

divergir completamente do comportamento real do sistema modelado. Por isso, um bom treinamento é importante. Se o modelo consegue capturar bem a dinâmica do sistema, generalizando com os dados capturados, o erro residual pode não afetar tanto as predições e por conseguinte, no modo de simulação, afetar menos o comportamento do sistema identificado.

Neste trabalho, a escolha por dividir a rede internamente se mostrou a melhor solução para resolver o problema do ruído compartilhado (o ruído gerado no nível do tanque 2 afetando o nível do tanque 1). Isso prova que conhecer também algumas características do sistema permite determinar melhor a escolha do modelo a ser treinado. Uma outra estratégia para resolver o problema do ruído compartilhado poderia ser utilizar outras camadas com outras funções de ativação na rede neural para que ela mesma pudesse cancelar os efeitos do ruído.

Como trabalhos futuros, serão testados novas configurações de redes para evitar o problema do ruído compartilhado, além de que esse modelo servirá de base como

REFERÊNCIAS

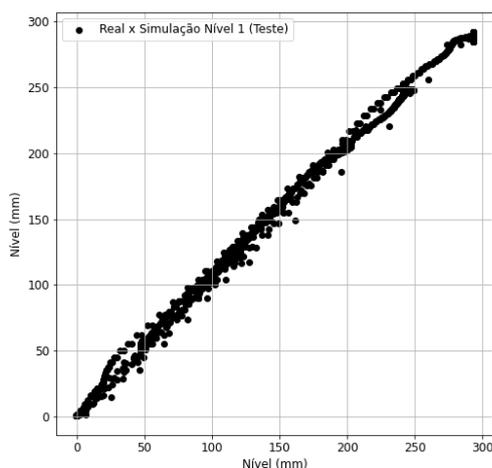


Figura 18. Gráfico capturado X simulação da rede dividida para o nível 1.

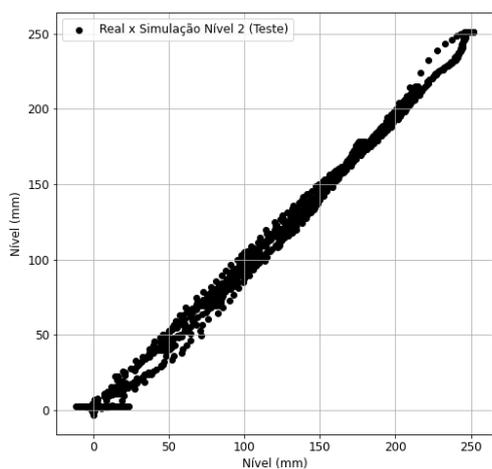


Figura 19. Gráfico capturado X simulação da rede dividida para o nível 2.

simulador para o treinamento de um controlador baseado em *Deep Q-Learning*. Como o agente deverá testar inúmeras ações para determinar as melhores para controlar o sistema, o uso de um modelo digital torna possível a aceleração do treinamento bem como evita condições perigosas no sistema de tanques real.

Além desse trabalho, também está em vista o uso desse mesmo modelo no treinamento de um sistema de detecção de falhas, servindo de base para determinar comportamentos anômalos do sistema de tanques real.

AGRADECIMENTOS

Os autores agradecem à UFRN pela infraestrutura de suporte ao desenvolvimento deste trabalho e pelo apoio financeiro através de bolsa de pesquisa.

- Bezerra, L.G.S., Martins, D.L., Neto, A.D.D., e de Melo, J.D. (2013). Detecção e diagnóstico de falhas em sensores de uma rede industrial *Foundation Fieldbus* utilizando sistemas inteligentes. *Anais XI Simpósio Brasileiro de Automação Inteligente*.
- de Menezes, P.L., Azevedo, C.A.V., Eyng, E., Neto, J.D., e Lima, V.L.A. (2015). Artificial neural network model for simulation of water distribution in sprinkle irrigation. *Revista Brasileira de Engenharia Agrícola e Ambiental*, 19(9), 817–822. doi:10.1590/1807-1929/agriambi.v19n9p817-822.
- Filho, A.M.P.P., Lopes, K.R., da Silva, V.L.C.M., Martins, D.L., Neto, A.D.D., de Melo, J.D., e Guedes, L.A. (2010). Implementação de um preditor neural no ambiente industrial *Foundation Fieldbus*. *Anais do XVIII Congresso Brasileiro de Automática*.
- Forgione, M. e Piga, D. (2021). Continuous-time system identification with neural networks: Model structures and fitting criteria. *European Journal of Control*, 59, 69–81.
- Gonzales, J. e Yu, W. (2018). Non-linear system modeling using lstm neural networks. *IFAC-PapersOnLine*, 51(13), 485–489.
- Kumar, R., Srivastava, S., Gupta, J.R.P., e Mohindru, A. (2019). Comparative study of neural networks for dynamic nonlinear systems identification. *Soft Comput*, 23, 101–114. doi:10.1007/s00500-018-3235-5.
- Machado, V., Neto, A.D.D., e de Melo, J.D. (2010). A neural network multiagent architecture applied to industrial networks for dynamic allocation of control strategies using standard function blocks. *IEEE Transactions on Industrial Electronics*, 57(5), 1823–1834. doi:10.1109/TIE.2009.2031196.
- Ogunmolu, O., Gu, X., Jiang, S., e Gans, N. (2016). Non-linear systems identification using deep dynamic neural networks. doi:10.48550/ARXIV.1610.01439. URL <https://arxiv.org/abs/1610.01439>.
- Sjöberg, J., Hjalmarsson, H., e Ljung, L. (1994). Neural networks in system identification. *IFAC Proceedings Volumes*, 27(8), 359–382. doi:10.1016/S1474-6670(17)47737-8.
- Wang, Y. (2017). A new concept using lstm neural networks for dynamic system identification. *American Control Conference (ACC)*, 5324–5329. doi:10.23919/ACC.2017.7963782.