Rede Neural Adaptativa com Múltiplas Camadas Paralelas: Aplicação ao Monitoramento de Processos Industriais

Lucas Lima Carneiro * **Pedro Henrique Silva Coutinho** *,** **Nayron Morais Almeida** * **Carlos Henrique de Morais Bomfim** *** **Benjamin Rodrigues de Menezes** *** **Sergio Gregorio de Oliveira** **** **Reinaldo Martínez Palhares** *** **Walmir Matos Caminhas** ***

* *Programa de Pós-Graduação em Engenharia Elétrica – Universidade Federal de Minas Gerais – Av. Antônio Carlos 6627, 31270-901, Belo Horizonte, MG, Brasil (e-mail: llcarneiro@ufmg.br)*
** *Departamento de Engenharia Eletrônica e Telecomunicações, Universidade do Estado do Rio de Janeiro.*
*** *Departamento de Engenharia Eletrônica, Universidade Federal de Minas Gerais.*
**** *CENPES/PETROBRAS, RJ*

Abstract: This work proposes a novel neural network architecture based on Parallel Layer Perceptron. This new architecture aims to overcome some limitations of the aforementioned model. Thus, a neural network with multiple adjustable parallel layers capable of solving tasks with multiple outputs is achieved. The proposed algorithm is applied to monitor industrial processes in its adaptive autoencoder form, capable of iteratively updating its parameters from a data stream. The model is validated in two examples that show its effectiveness in input reconstruction and in providing a latent layer with significative features for event detection.

Resumo: Este trabalho propõe uma nova arquitetura de rede neural baseada na Rede *Perceptron* com Camadas Paralelas. Esta nova arquitetura tem como objetivo superar algumas limitações do modelo no qual é baseada. Desse modo, tem-se uma rede com múltiplas camadas paralelas ajustáveis e capaz de solucionar problemas com múltiplas saídas. O algoritmo proposto é aplicado ao monitoramento de processos industriais na sua variante *autoencoder* adaptativa, capaz de atualizar seus parâmetros iterativamente a partir de um fluxo de dados. O modelo é validado em dois exemplos que demonstram sua eficácia em reconstruir as variáveis e prover uma camada latente com características significativas para a detecção de eventos.

Keywords: Process modeling and identification; Fault detection and diagnosis; Adaptive autoencoder; Neural networks.

Palavras-chaves: Modelagem e identificação de processos; Detecção e diagnóstico de falhas; *Autoencoder* adaptativo; Redes neurais.

1. INTRODUÇÃO

O monitoramento de processos visa manter a operação regular de processos industriais, tornando-os mais seguros e confiáveis. Em um ambiente onde uma única falha pode levar a grandes perdas se não detectada precocemente, o monitoramento busca detectá-las e diagnosticá-las, isto é, saber identificar uma falha no momento que ela ocorre e quais variáveis estão associadas (Qian et al., 2022).

Devido à crescente complexidade dos processos industriais, modelos de monitoramento baseados em dados substituem aqueles que dependem de modelos fenomenológicos ou do conhecimento de um especialista. Contudo, estes modelos necessitam de características (variáveis) significativas do sistema para obterem uma representação adequada do processo. Neste contexto, ferramentas estatísticas como Análise de Componentes Principais (PCA, do inglês *Principal Component Analysis*) e Mínimos Quadrados Parciais (PLS, do inglês *Partial Least Squares*) conseguem mapear as variáveis originais em um espaço latente de dimensão reduzida que contém as principais informações do processo (De Ketelaere et al., 2015; Dong et al., 2015).

Apesar de eficientes, técnicas como PCA e PLS normalmente assumem que os dados obedecem a certas condições (distribuição gaussiana, correlação linear) que nem sempre são verdadeiras (Qian et al., 2022). Além disso, em um

⋆ Este trabalho foi realizado com o apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) por meio do Programa de Excelência Acadêmica (PROEX) (Processo Nº 88887.934884/2024-00) e Código de Financiamento 001; com apoio Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq); e financiado pela Petrobras (Termo de Cooperação Nº 0050.0123014.22.9).

ambiente de identificação *online*, elas necessitam de dados *a priori* e constantes retreinamentos. Assim, modelos de aprendizado de máquina se mostram mais adequados para a extração de características. O *autoencoder* é um algoritmo não supervisionado que ganhou popularidade em monitoramento de processos por suas vantagens em termos de robustez para representar características em sua camada latente (Vincent et al., 2008; Chopra & Yadav, 2015).

Uma topologia de rede neural promissora para o monitoramento de processos é a Rede *Perceptron* com Camadas Paralelas (PLP, do inglês *Parallel Layer Perceptron*) (Caminhas et al., 2003). Esta arquitetura mescla estruturas de redes neurais comuns com Sistema de Inferência Fuzzy baseado em Redes Adaptativas (ANFIS, do inglês *Adaptive-network-based Fuzzy Inference System*) (Jang, 1993) de forma a reduzir a complexidade computacional e favorecer a implementação em ambiente paralelizado. Contudo, restrições como apenas um neurônio na camada de saída e número fixo de camadas paralelas impedem o seu uso como *autoencoder*.

A fim de explorar uma nova topologia de *autoencoder* adequada ao problema de monitoramento de processos, este trabalho trata da generalização da rede PLP para uma estrutura de múltiplas saídas e número variável de camadas paralelas. As principais contribuições deste trabalho são listadas a seguir:

- Propor uma modificação do PLP para lidar com problemas com múltiplas saídas e que permita variar o número de camadas paralelas assim como a quantidade de neurônios em cada uma;
- Propor uma estrutura de *autoencoder* a partir da arquitetura PLP proposta para ser aplicada ao problema de monitoramento de processos, denominada MPLP-AE. Para aprimorar o aprendizado da rede, utiliza-se uma estratégia adaptativa para aprendizado em tempo real a partir de um fluxo de dados, que evita retreinamento da rede;
- Utilizar o MPLP-AE em conjunto com indicadores de monitoramento associados a limites de controle para identificar ocorrência de eventos, assim como índices de contribuição para correlacionar as variáveis de processo com a ocorrência do evento.

A organização do texto é como segue. A rede PLP padrão é apresentada na Seção 2. A nova arquitetura de rede PLP é proposta na Seção 3. Na Seção 4, formula-se o problema de monitoramento de processos por meio do algoritmo MPLP-AE proposto. Os resultados são apresentados e discutidos na Seção 5. Por fim, as principais conclusões são apresentadas na Seção 6.

2. REDE PLP COM UMA CAMADA PARALELA

A rede PLP, representada na Figura 1, consiste em uma rede neural com duas camadas paralelas, ambas com m *perceptrons*, em que sua saída é dada por (Caminhas et al., 2003):

$$y = \beta\left(\sum_{i=1}^{m} \gamma_i(a) \cdot \phi_i(b)\right), \tag{1}$$

onde

$$a = Px \quad \text{e} \quad b = Vx, \tag{2}$$

e ainda $\gamma_i(\cdot)$, $\phi_i(\cdot)$ e $\beta(\cdot)$ são funções de ativação, $P \in \mathbb{R}^{m\times n}$ e $V \in \mathbb{R}^{m\times n}$ são as matrizes de peso das camadas paralelas, e $x \in \mathbb{R}^n$ é a amostra fornecida como entrada. Definem-se ainda os vetores de funções de ativação $\gamma = (\gamma_1, \ldots, \gamma_m)$ e $\phi = (\phi_1, \ldots, \phi_m)$.

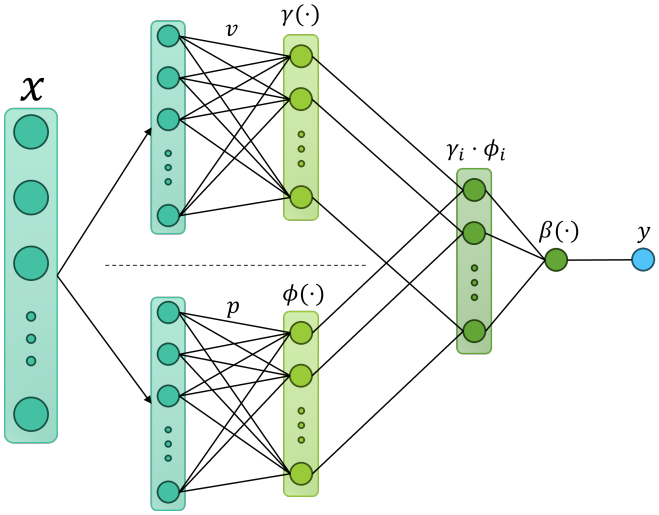


Figura 1. Arquitetura da Rede *Perceptron* com Camadas Paralelas.

Proposta como uma combinação das vantagens de redes neurais comuns e modelos ANFIS, a rede PLP busca ser capaz de implementar diferentes topologias minimizando o problema da complexidade computacional diretamente proporcional ao número de entradas. Além disso, sua arquitetura permite que ela seja facilmente implementada em máquinas ou *clusters* paralelos (Caminhas et al., 2003).

Todavia, a rede PLP apresenta limitações que a impedem de ser facilmente empregada em problemas de monitoramento de processos industriais. Por exemplo, seu número fixo de camadas paralelas restringe a implementação de estruturas mais complexas; já seu número fixo de neurônios em cada camada limita a adaptabilidade e aumenta o número de parâmetros desnecessariamente, isto é, adicionar um neurônio na camada $\gamma(\cdot)$ implica adicionar um neurônio nas camadas $\phi(\cdot)$ e $\beta(\cdot)$; por fim, a saída da rede está restrita a $y \in \mathbb{R}^1$ o que impede sua aplicação em tarefas multivariáveis.

De modo a contornar essas limitações, este trabalho propõe a Rede com Múltiplas Camadas Paralelas (MPLP, do inglês *Multiple Parallel Layers Perceptron*) e, mais especificamente para o problema de monitoramento de processos, sua variante *autoencoder* aqui denominada MPLP-AE.

3. REDE PLP COM MÚLTIPLAS CAMADAS PARALELAS

A fim de tornar a rede PLP multivariável na saída, propõe-se a adição de pesos $W^{(y)}$ após a camada $\beta(\cdot)$. Desse modo, a arquitetura possibilita o redimensionamento da camada de saída. Ainda, ao remover o produto $\gamma_i(a)\cdot\phi_i(b)$, as camadas paralelas tornam-se independentes. Contudo, para transferir o conhecimento de cada uma à camada $\beta(\cdot)$, se faz necessária a adição de pesos $W^{(z)}$. Assim,

ao generalizar a rede para um número n_{pl} de camadas paralelas, obtém-se a seguinte estrutura:

$$z = \left(\varphi_1, \ldots, \varphi_{n_{pl}}\right), \tag{3}$$

$$y = W^{(y)} z, \tag{4}$$

onde

$$\varphi_i = \beta\left(W^{(z_i)} \gamma_i(a_i)\right), \quad a_i = W^{(x_i)} x, \tag{5}$$

os pesos $W^{(x_i)} \in \mathbb{R}^{m_i \times n}$ conectam a entrada às camadas γ_i, similar aos pesos P e V no PLP, com m_i sendo o número de neurônios na i-ésima camada paralela. A dimensão de $W^{(y)} \in \mathbb{R}^{n_y \times n_{pl}}$ depende do número de saídas n_y e tem-se $W^{(z_i)} \in \mathbb{R}^{1 \times m_i}$. Ressalta-se que, neste contexto, γ_i não representa componentes de um vetor, mas sim funções de ativação distintas.

A Figura 2 exibe a variante *autoencoder* da rede MPLP: uma camada que reconstrói a variável de entrada substitui a camada de saída. Nela, a camada $\beta(\cdot)$ recebe a denominação z, usualmente dada à camada latente de um *autoencoder*. Interpreta-se as camadas paralelas como o codificador, ou *encoder*, e a transformação de z em $\hat{x}$ como o decodificador, ou *decoder*.

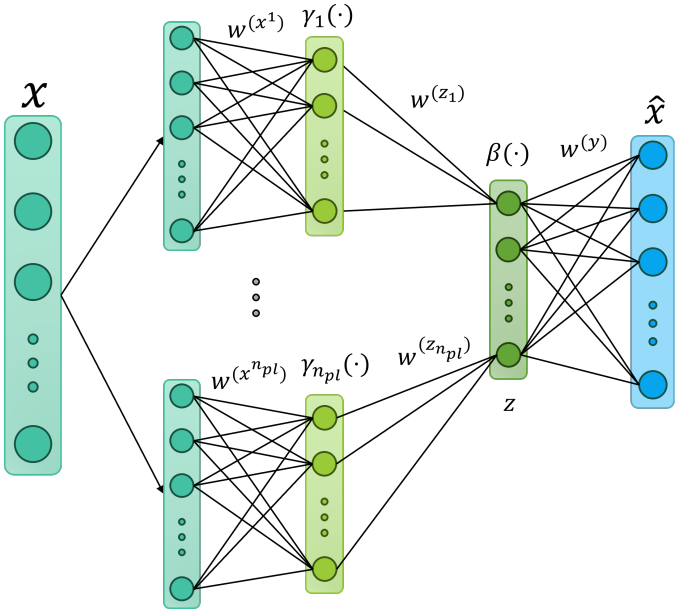


Figura 2. Arquitetura da variante *autoencoder* da Rede com Múltiplas Camadas Paralelas – MPLP-AE.

4. MONITORAMENTO DE PROCESSOS VIA MPLP-AE

Em um *autoencoder*, uma amostra x é apresentada ao codificador (3) que a mapeia na camada latente z. Normalmente, a dimensão do espaço latente é menor do que o espaço original, de forma a comprimir a informação presente em x. No MPLP-AE isto é alcançado ao fazer $n_{pl} < n$. Para transformar esta informação codificada na reconstrução da amostra, fornece-se z ao decodificador (4).

O *autoencoder* é comumente utilizado como uma alternativa à métodos *offline* de redução de dimensionalidade, uma vez que sua camada latente é capaz de extrair caraterísticas representativas do conjunto de dados. Em sua forma adaptativa, seus parâmetros são atualizados a partir de um fluxo de dados, sem a necessidade de treino com dados *a priori* ou retreino após um dado período de tempo.

No problema de monitoramento de processos, novos eventos (falhas, mudanças de ponto de operação) são reconhecidos se alguma métrica $f(x)$ ultrapassa seu limite de controle CL. Além da identificação, é preciso associar as variáveis em x à ocorrência do evento. Para isso, definem-se índices de contribuição associados a cada variável $x_i \in x$.

Dois índices de monitoramento são utilizados neste trabalho. O primeiro é o SPE, definido da seguinte forma (Hallgrímsson et al., 2020):

$$SPE(k) = \|x(k) - E[\hat{x}(k)]\|^2, \tag{6}$$

onde k representa o instante de tempo ou amostra. Em contrapartida a sua definição original, aqui opta-se por substituir $\hat{x}$ por $E[\hat{x}]$, uma vez que o MPLP-AE é treinado para minimizar a norma da diferença entre x e $\hat{x}$.

Como limite de controle para o SPE, define-se (Hallgrímsson et al., 2020):

$$CL_{SPE}(k) = \frac{\sigma^2_{SPE}(k)}{2\mu_{SPE}(k)} \chi^2_{(2\mu^2_{SPE}(k)/\sigma^2_{SPE}(k), \alpha)}, \tag{7}$$

onde μ_{SPE} e σ^2_{SPE} são, respectivamente, a média e variância do índice SPE, calculados de forma iterativa; $\chi^2_{(\nu,\alpha)}$ é o valor-p da função de distribuição χ^2 com ν graus de liberdade e nível de significância de $1 - \alpha$.

O segundo índice utilizado, denominado Z^2, é definido da seguinte forma:

$$Z^2(k) = \|z(k) - \mu_z(k)\|^2. \tag{8}$$

Este índice é definido em termos da diferença entre a variável do espaço latente z e a sua média estimada de forma recursiva μ_z. Seu limite de controle é dado por:

$$CL_{Z^2}(k) = \frac{\sigma^2_{Z^2}(k)}{2\mu_{Z^2}(k)} \chi^2_{(2\mu^2_{Z^2}(k)/\sigma^2_{Z^2}(k), \alpha)}, \tag{9}$$

onde μ_{Z^2} e $\sigma^2_{Z^2}$ são, respectivamente, a média e variância de Z^2, calculados de forma iterativa.

Para avaliar a contribuição das variáveis de x sobre o evento identificado, utiliza-se o índice de contribuição definido como (Hallgrímsson et al., 2020):

$$SPE_i(k) = (x_i(k) - E[\hat{x}_i(k)])^2, \; i = 1, \ldots, n. \tag{10}$$

onde SPE_i indica a contribuição da i-ésima variável em x para a ocorrência do evento.

5. RESULTADOS DE SIMULAÇÃO

Nesta seção, o algoritmo proposto é aplicado ao problema de monitoramento de processos. Foram selecionadas dois conjuntos de dados: o primeiro, um conjunto denominado "Dados Sintéticos", consiste em degraus que simulam mudanças de ponto de operação do processo; o segundo, "Sim3Tanks", representa um processo com três tanques acoplados sujeito a falhas e mudanças de ponto de operação.

5.1 Exemplo 1: Dados Sintéticos

Neste exemplo, o conjunto de dados é composto por 2000 amostras e três variáveis. Durante as 500 primeiras amostras o processo está em um modo de operação em que todas

as 3 variáveis permanecem constantes no menor nível. Na amostra 501, as variáveis x_1 e x_2 simultaneamente sofrem uma mudança de ponto de operação. Quando a amostra 1001 é alcançada, a variável x_1 retorna ao primeiro modo de operação. Por fim, na amostra 1501, as variáveis x_1 e x_3 alteram o ponto de operação e o processo permanece assim até a amostra final.

As variáveis foram normalizadas entre 0 e 1. As configurações do MPLP-AE foram: um neurônio com ativação sigmoide e um com ativação gaussiana [1], ou seja, duas camadas paralelas; treinado por *backpropagation* com taxa de aprendizado de 0,05. O nível de significância de ambos os índices de monitoramento foi escolhido como 0,95.

O processo foi simulado com o MPLP-AE e os índices de monitoramento SPE e Z^2 foram utilizados para a identificação de eventos, seus valores e dos seus devidos limites de controle são apresentados nas Figuras 3 e 4, respectivamente.

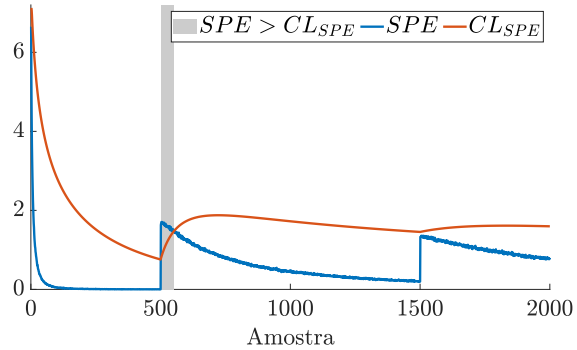


Figura 3. Exemplo 1 – Valores do índice SPE e seu limite de controle CL_{SPE} em cada amostra. Eventos detectados por este índice estão destacados pelas regiões em cinza.

O SPE foi capaz de detectar a primeira mudança no ponto de operação que ocorre na amostra 501. Contudo, ele não consegue perceber a mudança na amostra 1001. Isto é notável pois além de não ultrapassar seu limite de controle, o valor do índice permanece em decréscimo, comportamento similar a um transitório. O evento na amostra 1501 também não é detectado pelo índice, porém vê-se que seu valor sofre uma variação brusca no exato momento do evento.

Por sua vez, o Z^2 conseguiu detectar os três eventos, ainda que com algumas ressalvas. Os eventos nas amostras 501 e 1501 foram detectados assim que ocorreram. Porém, o evento em 1001 foi detectado tardiamente apenas na amostra 1041. O valor do índice ainda sofreu alteração na amostra 1001 mas não foi suficiente para ultrapassar seu limite de controle. Nota-se que ambos os índice têm dificuldade em detectar o segundo evento, no qual apenas uma variável sofre alteração.

Visto que os três eventos foram detectados, o SPE_i foi utilizado para verificar quais variáveis mais contribuíram

[1] Em ambos os exemplos desta seção, a função de ativação gaussiana foi utilizada com $c = 1$ e $\sigma = 0$, onde c é seu centro e σ seu desvio padrão.

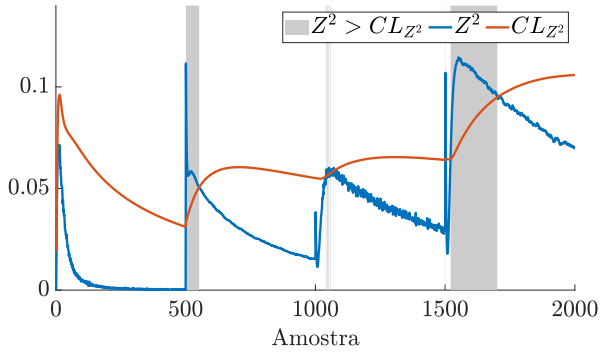
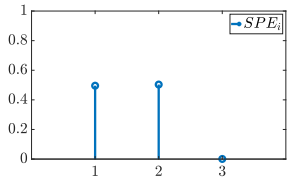
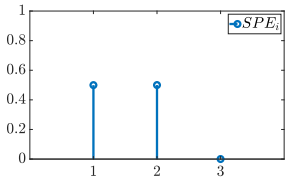


Figura 4. Exemplo 1 – Valores do índice Z^2 e seu limite de controle CL_{Z^2} em cada amostra. Eventos detectados por este índice estão destacados pelas regiões em cinza.

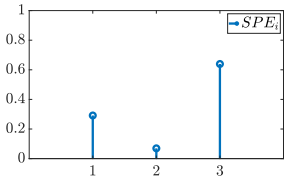
para a ocorrência de cada evento. A Figura 5 exibe as contribuições normalizadas.



(a) Contribuições das variáveis na amostra 501.



(b) Contribuições das variáveis na amostra 1041.



(c) Contribuições das variáveis na amostra 1501.

Figura 5. Exemplo 1 – Contribuições das variáveis nas amostras em que foram detectados eventos.

As variáveis que contribuíram para o primeiro evento foram identificadas corretamente pelo SPE_i como x_1 e x_2. No segundo evento, apesar do índice Z^2 conseguir detectá-lo, o SPE_i não identifica as variáveis corretamente. Isto pode ter ocorrido porque este índice de contribuição é definido analogamente ao índice de monitoramento SPE que por sua vez ainda apresentava comportamento similar a um transitório após a ocorrência do primeiro evento, assim identificando as mesmas variáveis. Já no terceiro evento, as variáveis são identificadas corretamente como x_1 e x_3, isto ocorre pois, apesar do SPE não ultrapassar seu limite de controle, o evento é "sentido" pelo índice, como é possível ver na amostra 1501 da Figura 3.

As variáveis de processo e suas reconstruções pelo MPLP-AE são exibidas na Figura 6, onde as áreas em cinza mostram quando um evento foi identificado por quaisquer índices.

As reconstruções se mostram adequadas e capazes de reduzir parcialmente o ruído presente nas variáveis originais. É perceptível que após a mudança no ponto de operação

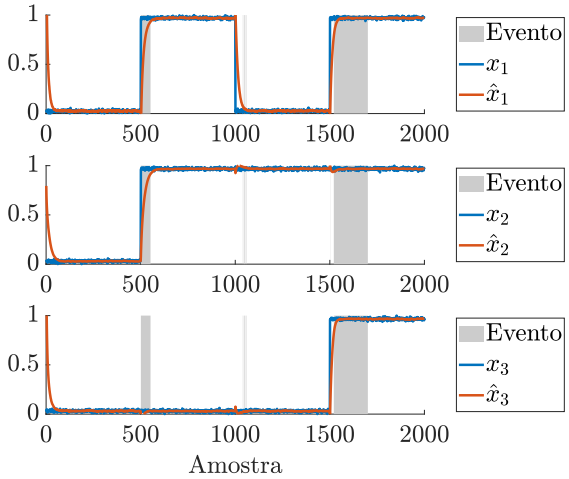


Figura 6. Exemplo 1 – Variáveis do processo e suas respectivas reconstruções. Eventos estão destacados pelas regiões em cinza.

de uma variável, as demais apresentam uma pequena variação. Também é notável que o evento 2 foi detectado pelo índice apenas após o transitório da variável reconstruída.

5.2 Exemplo 2: Sim3Tanks – Processo de controle de três tanques acoplados

Neste exemplo, o problema abordado consiste em um sistema de controle de um processo de três tanques acoplados. Os dados foram coletados a partir do Sim3Tanks (Farias et al., 2018). Ilustrado na Figura 7, ele representa um sistema de comportamento híbrido e não linear sujeito a diferentes tipos de perturbações, falhas e ruídos.

Fonte: Farias et al. (2018).

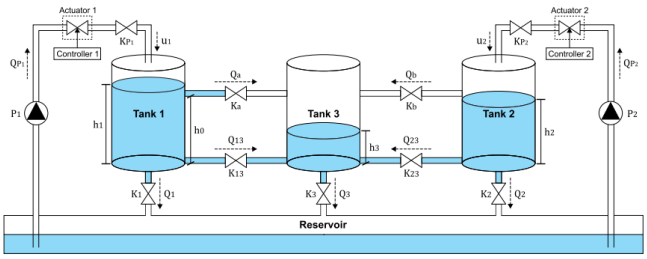


Figura 7. Ilustração do processo de três tanques acoplados.

Este sistema foi simulado considerando-se que as válvulas K_{p1}, K_{p2}, K_{13}, K_{23}, e K_3 estão abertas e as demais fechadas. O sistema de controle em malha fechada da vazão Q_3 está sujeito a uma mudança de ponto de operação de 120 cm^3/s para 40 cm^3/s no instante $t = 300$ s. Porém, isto ocorre durante um vazamento no Tanque 1 que se iniciou no instante $t = 200$ s. [2]

Assim, o conjunto de dados é formado por 8 variáveis: os níveis dos três tanques, $x_1 = h_1$, $x_2 = h_2$, $x_3 = h_3$; e as vazões $x_4 = u_1$, $x_5 = u_2$, $x_6 = Q_{13}$, $x_7 = Q_{23}$, $x_8 = Q_3$, todas sujeitas a ruídos brancos de medição com desvio padrão 0,3 para os sensores de nível e desvio padrão de 1,5 para os sensores de fluxo. O tempo total de simulação foi de $t = 500$ s que, com o tempo de amostragem de 0,1 s, totalizou 5000 amostras. As variáveis foram normalizadas entre 0 e 1 e os parâmetros referentes ao MPLP-AE e aos índices de monitoramento foram definidos como no Exemplo 1.

O sistema foi simulado com o MPLP-AE e valores dos índices de monitoramento SPE e Z^2 assim como seus devidos limites de controle são apresentados nas Figuras 8 e 9, respectivamente.

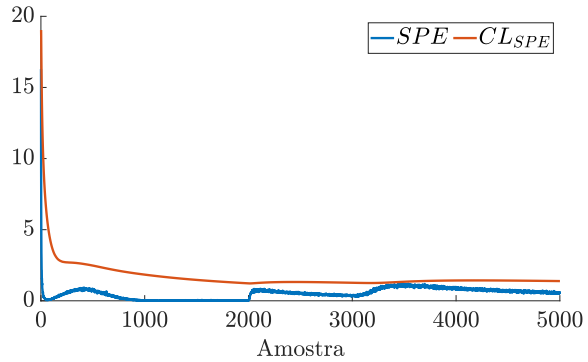


Figura 8. Exemplo 2 – Valores do índice SPE e seu limite de controle CL_{SPE} em cada amostra. Eventos detectados por este índice estão destacados pelas regiões em cinza.

O índice SPE não foi capaz de detectar nem o vazamento nem a mudança no ponto de operação. Apesar de haver variação de seu valor em amostras próximas aos eventos, este não ultrapassa seu limite de controle.

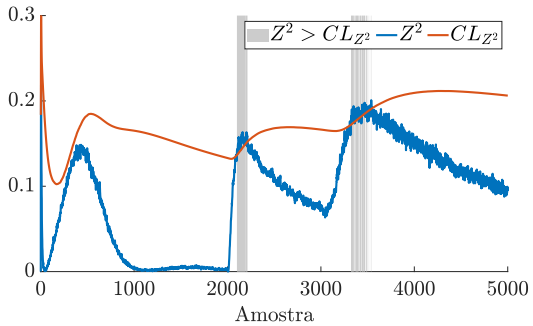


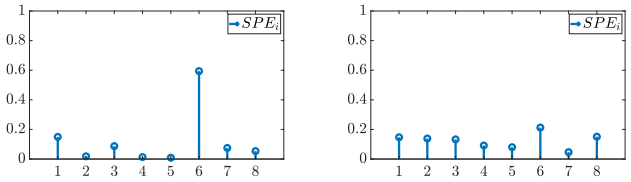
Figura 9. Exemplo 2 – Valores do índice Z^2 e seu limite de controle CL_{Z^2} em cada amostra. Eventos detectados por este índice estão destacados pelas regiões em cinza.

A Figura 9 indica que o índice Z^2 identificou os dois eventos. Novamente, percebe-se uma identificação tardia uma vez que o evento na amostra 2000 foi identificado apenas na 2091 e o evento na amostra 3000 percebido na apenas na 3310. Porém, a partir das reconstruções apresentadas na Figura 11, fica claro que os eventos foram detectados quando o sistema se encontrava próximo ao regime permanente (novo estado do sistema após o evento). Isto, em conjunto com os resultados da Seção 5.1, indica que a detecção tardia decorre do tempo necessário à transição para o novo estado e não da estrutura do modelo.

[2] Para simular o vazamento, tomou-se o sinal $f_7 = 0{,}8$ após o instante $t = 200$ s.

Ademais, é notável que os tanques levam aproximadamente 1000 amostras para atingir o regime permanente após o início da simulação. Observa-se que ambos os índices sofrem alterações decorridas deste transitório. Mas, como esperado, os índices não o identificam como novo evento.

Dado que os dois eventos foram detectados, verificou-se quais variáveis mais contribuíram para a ocorrência de cada evento por meio do SPE_i. A Figura 10 exibe as contribuições normalizadas.



(a) Contribuições das variáveis na amostra 2091.

(b) Contribuições das variáveis na amostra 3310.

Figura 10. Exemplo 2 – Contribuições das variáveis nas amostras em que foram detectados eventos.

As variáveis x_6 e x_1 foram identificadas como as que mais contribuíram para o primeiro evento: a altura da coluna de água (x_1) e a vazão do Tanque 1 para o Tanque 3 (x_6) foram comprometidas em aproximadamente 54% e 87%, respectivamente. Sendo ambas variáveis relacionadas ao Tanque 1, esta escolha faz sentido uma vez que foi neste que o vazamento ocorreu.

No segundo evento, x_6 e x_8 foram as variáveis com maior contribuição. Entretanto, observa-se valores também elevados para x_1, x_2 e x_3, e um pouco menor mas ainda significante para x_4 e x_5. Este comportamento pode ser explicado por um esforço conjunto do sistema de controle em manter a vazão do Tanque 3 na referência, ou seja, não só a vazão x_6 e a variável controlada x_8 sofreram mudança, mas variações nas alturas das colunas de água (x_1, x_2 e x_3) e nos esforços de controle (x_4 e x_5) foram essenciais para a mudança do ponto de operação.

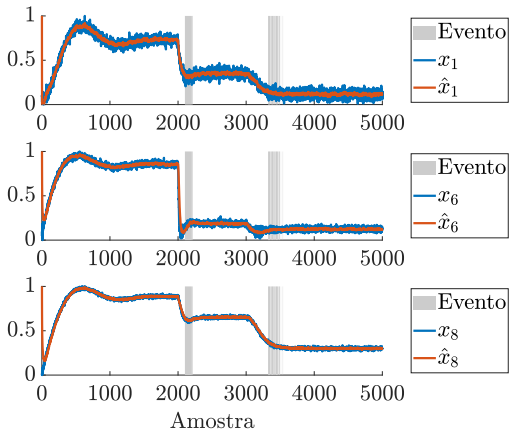


Figura 11. Exemplo 2 – Variáveis que apresentaram as maiores contribuições para os eventos detectados e suas respectivas reconstruções. Eventos estão destacados pelas regiões em cinza.

As variáveis que mais contribuíram para os eventos e suas reconstruções pelo MPLP-AE são exibidas na Figura 11, onde as áreas em cinza mostram quando um evento foi identificado por quaisquer índices. Mais uma vez, as reconstruções se mostraram adequadas e com redução dos ruídos de forma a permitir a identificação dos eventos pelos índices.

6. CONCLUSÃO

Este trabalho apresentou uma nova arquitetura de rede neural capaz de executar tarefas de forma paralela. O modelo proposto foi apresentado na forma de *autoencoder* adaptativo e aplicado ao problema de monitoramento de processos em dois exemplos por simulações com dados sintéticos, um destes que simulava um processo de controle que apresenta falha e mudança de ponto de operação. A rede proposta, em conjunto com os índices de monitoramento, foi capaz de identificar a ocorrência de todos os eventos programados e, com uma exceção, as variáveis que mais contribuíram para os eventos. Assim, os resultados sugerem eficiência da rede em comprimir informações significativas na camada latente e validam seu uso em monitoramento de processos.

REFERÊNCIAS

Caminhas, W.M., Vieira, D.A., & Vasconcelos, J.A. (2003). Parallel layer perceptron. *Neurocomputing*, 55(3-4), 771–778.

Chopra, P. & Yadav, S.K. (2015). Fault detection and classification by unsupervised feature extraction and dimensionality reduction. *Complex & intelligent systems*, 1, 25–33.

De Ketelaere, B., Hubert, M., & Schmitt, E. (2015). Overview of PCA-based statistical process-monitoring methods for time-dependent, high-dimensional data. *Journal of Quality Technology*, 47(4), 318–335.

Dong, J., Zhang, K., Huang, Y., Li, G., & Peng, K. (2015). Adaptive total PLS based quality-relevant process monitoring with application to the tennessee eastman process. *Neurocomputing*, 154, 77–85.

Farias, A.O., Queiroz, G.A.C., Bessa, I.V., Medeiros, R.L.P., Cordeiro, L.C., & Palhares, R.M. (2018). Sim3tanks: a benchmark model simulator for process control and monitoring. *IEEE Access*, 6, 62234–62254.

Hallgrímsson, Á.D., Niemann, H.H., & Lind, M. (2020). Improved process diagnosis using fault contribution plots from sparse autoencoders. *IFAC-PapersOnLine*, 53(2), 730–737.

Jang, J.S. (1993). ANFIS: adaptive-network-based fuzzy inference system. *IEEE transactions on systems, man, and cybernetics*, 23(3), 665–685.

Qian, J., Song, Z., Yao, Y., Zhu, Z., & Zhang, X. (2022). A review on autoencoder based representation learning for fault detection and diagnosis in industrial processes. *Chemometrics and Intelligent Laboratory Systems*, 104711.

Vincent, P., Larochelle, H., Bengio, Y., & Manzagol, P.A. (2008). Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, 1096–1103.