

Estimação de falhas com observadores de entrada desconhecida e modelagem *fuzzy* granular ^{*}

Anderson Bento ^{*} Lucas Oliveira ^{*} Valter J. S. Leite ^{*}

^{*} Departamento de Engenharia Mecatrônica, CEFET-MG, Divinópolis
(e-mail: bentoavb@gmail.com, lqsoliveira@cefetmg.br, valter@ieee.org)

Abstract: The development of a hybrid method for fault estimation is proposed by sharing both model and process knowledge. The detection of process faults signal is performed from the residues generated by unknown input observers, while faults quantification comes from granular fuzzy modeling. The proposal is evaluated in a numerical example and applied to a nonlinear coupled tank physical process. The results suggest that the proposal can be used in systems of higher complexity and affected by measurement noise and uncertainties. The method could determine both the fault occurrence and the intensity of such faults.

Resumo: Neste trabalho é proposto o desenvolvimento de um método híbrido para estimação de falhas baseado tanto no modelo quanto no conhecimento sobre o processo. A detecção do sinal de falhas no processo é realizada a partir do resíduo obtido pelo uso de observadores de entrada desconhecida, enquanto a quantificação da falha é realizada por meio da modelagem *fuzzy* granular. A proposta é avaliada em um exemplo numérico e também aplicada em um processo físico de tanques acoplados com não-linearidade. Os resultados sugerem que a proposta pode ser usada em sistemas de maior porte, e sujeitos a ruídos e incertezas, conseguindo determinar tanto a ocorrência quanto a intensidade das falhas.

Keywords: Fault estimation; linear systems; observers; granular fuzzy modeling.

Palavras-chaves: Estimação de falhas; processos lineares; observadores; modelagem *fuzzy* granular.

1. INTRODUÇÃO

Diagnóstico de falhas consiste em um sistema de monitoramento aplicado à sistema de controle, a fim de detectar falhas, diagnosticar sua localização, importância e impacto para o processo controlado (Chen e Patton, 2012). Esse sistema normalmente desempenha três funções: detecção, primeira etapa do processo de detecção de falhas, que consiste na decisão binária da existência de falha; isolamento, etapa intermediária que consiste na identificação do componente do sistema que apresenta falha; e por fim, a estimação, que consiste na quantificação e dimensionamento da falha (Ding, 2008).

De modo geral, as técnicas de diagnóstico de falhas podem ser agrupadas em três grandes grupos: técnicas baseadas em modelo, baseadas em conhecimento do processo e baseadas em sinais (Gao et al., 2015). Diagnóstico de falhas baseado em modelo, é apresentado inicialmente por Beard (1971) e consiste em monitorar as diferenças entre as variáveis do processo real comparando-as com as variáveis do modelo do processo (Gao et al., 2015). Dentre as várias abordagens, é comum a aplicação de observadores de estados para obter informações do processo e a comparação dessas com a saída do modelo. Essa abordagem resulta em um sinal de resíduo (Ding, 2008) que é indicador de falhas. Por outro lado, técnicas baseadas em sinais

consistem na utilização de sinais medidos do processo em vez de modelos de entrada-saída, baseando-se na geração e análise de sintomas ou padrões de funcionamento do processo (Gao et al., 2015). Por fim, técnicas baseadas em conhecimento do processo consistem na aplicação de técnicas de aprendizado de máquina e inteligência artificial no procedimento de determinação do diagnóstico de falhas (Andonovski et al., 2019).

Nos últimos anos, algumas abordagens sobre diagnóstico de falhas foram introduzidas na literatura. Por exemplo, Chen e Zhang (1991) apresentaram um esquema de detecção de falhas, a partir da aplicação de observadores de entrada desconhecida para geração de resíduos, e a partir da análise desses resíduos detectar e isolar as falhas no sistema. Chen e Patton (1996), por outro lado, estenderam os resultados apresentados por Chen e Zhang (1991), realizando estimação e filtragem ótima de sistemas estocásticos. Recentemente, Junior et al. (2019) apresentaram uma abordagem de estimação de falhas para sistemas através do uso da técnica de modelagem *fuzzy* Takagi-Sugeno em tempo discreto aplicando observadores de entrada desconhecida.

Sob outra perspectiva, diversos trabalhos exploraram a aplicação de técnicas de aprendizado de máquina e inteligência artificial. Lemos et al. (2013), apresentaram uma técnica baseada em modelos *fuzzy* através da aplicação de um algoritmo de clusterização evolutiva participativa para identificar modos de operação do sistema. Já Ando-

^{*} Os autores agradecem ao CEFET-MG e FAPEMIG pelo apoio financeiro através do projeto PIBIC 10743/2019.

novski et al. (2019) propõem uma abordagem utilizando a modelagem *fuzzy* simplificada desenvolvida por Angelov e Yager (2011). Nessa abordagem, os autores aplicam a modelagem *fuzzy* para treinar uma base de regras e aplicá-la no processo de classificação de tipos de falhas em um sistema de aquecimento, ventilação e ar condicionado.

Neste trabalho é proposto o desenvolvimento de um método híbrido para estimação de falhas baseado em modelos e conhecimento do processo. Nessa abordagem, a detecção do sinal de falhas no processo é realizada a partir do resíduo obtido pelo uso de observadores de entrada desconhecida como é proposto em Chen e Patton (1996). Por sua vez, a estimação e avaliação da amplitude das falhas é realizada através do método proposto em Andonovski et al. (2019), aplicando modelagem *fuzzy* granular evolutiva. A metodologia proposta é submetida à busca de solução em um exemplo numérico e validada em um experimento realizado em um sistema de tanques acoplados. Os resultados são avaliados qualitativamente e quantificados levando em conta o erro de estimação das falhas, os quais sugerem que o método proposto estima a magnitude de falhas com relativa precisão.

O restante deste texto é organizado com a seguinte estrutura. Na Seção 2 são apresentadas as técnicas e métodos necessários para desenvolvimento do trabalho. Na sequência, a abordagem de estimação de falhas proposta no trabalho é apresentada na Seção 3. Na Seção 4 são apresentados e avaliados os resultados obtidos com a simulação do problema numérico e o experimento realizado no sistema de tanques acoplados. Por fim, na Seção 5, apresenta-se as considerações finais relativas ao presente trabalho.

2. PRELIMINARES

Nessa seção é apresentada uma breve revisão sobre os métodos de detecção de falhas baseada em modelo, observador de entrada desconhecida e do algoritmo de modelagem *fuzzy* simplificada AnYa necessários para o desenvolvimento da solução proposta.

2.1 Detecção de falhas baseada em modelo

A detecção de falhas baseada em modelo consiste na simulação de um modelo em paralelo ao processo, e alimentá-lo com os mesmos sinais de entrada do sistema como é apresentado na Figura 1.

De acordo com o diagrama da Figura 1, o vetor de resíduos, $r(t)$, é definido pela diferença entre a saída do processo e a saída do modelo (Ding, 2008). Desse modo, é razoável assumir que os valores das variáveis de saída do modelo acompanham aquelas das saídas do processo durante uma operação sem falha, porém apresentam um desvio na ocorrência de falhas (Chen e Patton, 2012). Por sua vez, o sinal de resíduo é submetido ao bloco de avaliação do resíduo, onde será processado e avaliado, em ordem de identificar se há falha no processo. Nesse sentido, a decisão lógica pode ser obtida a partir da análise de um limiar, que definirá se há ou não falha no processo, de modo que seja respeitado a relação (Chen e Patton, 2012):

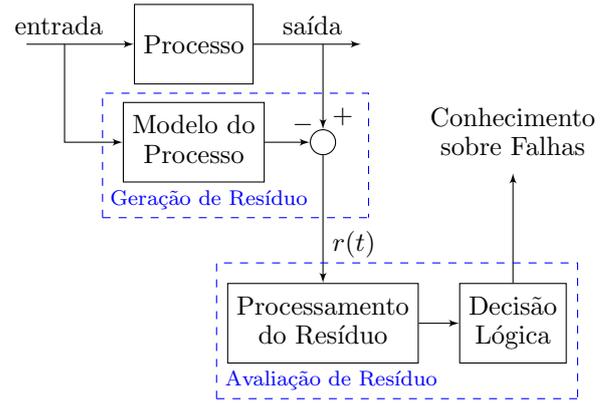


Figura 1. Detecção de falhas baseada em modelo, adaptado de Ding (2008).

$$\text{Conhecimento sobre Falhas} = \begin{cases} \text{processo livre de falhas} & \text{se } \|r(t)\| \leq \sigma, \\ \text{processo sujeito à falhas} & \text{se } \|r(t)\| > \sigma, \end{cases} \quad (1)$$

em que $\sigma > 0$ é um valor a ser definido pelo projetista, associado com o limiar de falhas do processo. Note que quanto mais ruidosas são as medições, maior o valor de σ .

2.2 Observador de entrada desconhecida

Considere o sistema linear discreto,

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) + Ed(k) + \zeta(k), \\ y(k) &= Cx(k) + \eta(k), \end{aligned} \quad (2)$$

em que $x(k) \in \mathbb{R}^n$ é o vetor de estados, $y(k) \in \mathbb{R}^m$ é o vetor de saídas, $u(k) \in \mathbb{R}^r$ é o vetor de entradas conhecidas, $d(k) \in \mathbb{R}^q$ é o vetor de entradas desconhecidas, $\zeta(k)$ e $\eta(k)$ são ruídos brancos com média zero e matrizes de covariância $Q \in \mathbb{R}^{n \times n}$ e $R \in \mathbb{R}^{m \times m}$ respectivamente. $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times r}$, $C \in \mathbb{R}^{m \times n}$ e $E \in \mathbb{R}^{n \times q}$ são matrizes do sistema. A estimação dos estados do sistema (2) se dá pelo uso do observador de entrada desconhecida (*unknown input observer* – UIO) definido em (Chen e Patton, 1996) como:

$$\begin{aligned} z(k+1) &= Fz(k) + TBu(k) + Ky(k), \\ \hat{x}(k) &= z(k) + Hy(k), \end{aligned} \quad (3)$$

em que $z(k) \in \mathbb{R}^n$ é o vetor de estados do observador, $\hat{x}(k) \in \mathbb{R}^n$ é o vetor de estados estimados e $F \in \mathbb{R}^{n \times n}$, $T \in \mathbb{R}^{n \times n}$, $K \in \mathbb{R}^{n \times m}$ e $H \in \mathbb{R}^{n \times m}$ são matrizes do observador, tal que (Chen e Patton, 2012):

$$E = HCE, \quad (4)$$

$$T = I - HC, \quad (5)$$

$$F = A_1 - LC, \quad (6)$$

$$K = L + FH, \quad (7)$$

em que $L \in \mathbb{R}^{n \times m}$ é o ganho do observador e $A_1 = TA$, tal que $A_1 \in \mathbb{R}^{n \times n}$. Note que só haverá solução para (4), se respeitado a condição: $\text{rank}(CE) = \text{rank}(E)$, o que resulta em

$$H = E(CE)^+, \quad (8)$$

em que $+$ é o operador pseudoinversa. Aplicando em (3) as relações apresentadas pelas equações (4)-(7), estabelece-se a relação matemática que determina a dinâmica do erro de estimação pelo observador, como:

$$e(k+1) = Fe(k) - L\eta(k) - H\eta(k+1) + T\zeta(k), \quad (9)$$

sendo $e(k) = x(k) - \hat{x}(k)$ o erro de estimação. Logo, uma vez que o par (C, A_1) seja observável, pode-se projetar o ganho L , tal que a matriz F torne-se estável, o que conseqüentemente resulta em $E\{e(k)\} \rightarrow 0$ quando $k \rightarrow \infty$, ou seja, o valor médio do erro de estimação tende a zero em regime permanente. Por outro lado, se o par (C, A_1) não é observável mas é detectável, ou seja, todos os estados não observáveis são estáveis, é necessário aplicar um procedimento de decomposição canônica da representação do sistema, para projetar o ganho do observador (Chen, 2013).

2.3 Modelagem fuzzy simplificada AnYa

O algoritmo AnYa é um sistema *fuzzy* evolutivo simplificado desenvolvido por Angelov e Yager (2011) que se aplica aos antecedentes não paramétricos e regras *fuzzy* de classificação descritas pela regra:

$$\mathfrak{R}_i(k) : \text{SE } v(k) \sim X_i \text{ ENTÃO } v(k) \in \text{Classe}_i, \quad (10)$$

em que $\mathfrak{R}_i(k)$ é a i -ésima regra *fuzzy* no instante k , com $i = 1, \dots, c$, sendo c o número de regras, $v(k) \in \mathbb{R}^p$ é o vetor de dados de entrada, $X_i \in \mathbb{R}^p$ é um grupo ou nuvem de dados de entrada e o operador \sim representa que a entrada de dados está relacionada com o grupo X_i de acordo com o grau de pertinência ponderada,

$$\lambda_i(k) = \frac{\gamma_i(k)}{\sum_{j=1}^c \gamma_j(k)} \quad i = 1, 2, \dots, c, \quad (11)$$

em que $\gamma_i(k)$ é a densidade local do dado de entrada em relação ao grupo X_i calculada como

$$\gamma_i(k) = \frac{1 + \frac{M_i - 1}{M_i} \text{tr}(\Sigma_i)}{1 + (v(k) - \mu_i)^T (v(k) - \mu_i) + \frac{M_i - 1}{M_i} \text{tr}(\Sigma_i)}, \quad (12)$$

sendo μ_i o valor médio dos dados que fazem parte do grupo X_i , atualizado recursivamente com

$$\mu_i(k) = \frac{M_i - 1}{M_i} \mu_i(k - 1) + \frac{1}{M_i} v(k). \quad (13)$$

Desse modo, quando um novo dado é inserido no grupo, o número de dados nesse grupo, M_i , e a matriz de covariância, $\Sigma_i \in \mathbb{R}^{p \times p}$, dos dados presentes no grupo X_i , são atualizados pela relação:

$$M_i(k) = M_i(k - 1) + 1 \quad (14)$$

$$\Sigma_i(k) = \frac{1}{M_i - 1} \mathbf{S}_i(k),$$

com \mathbf{S}_i sendo a covariância não normalizada calculada recursivamente sempre que um novo dado é inserido com

$$\mathbf{S}_i(k) = \mathbf{S}_i(k - 1) + (v(k) - \mu_i(k))(v(k) - \mu_i(k))^T. \quad (15)$$

O mecanismo de evolução do algoritmo AnYa altera a base de regras (10) conforme é apresentado no Algoritmo 1. Desse modo, quando um novo dado de entrada $v(k)$ é recebido, calcula-se a densidade local desse elemento em relação a todos os grupos existentes com auxílio de (12). Na seqüência, verifica-se se a regra associada ao maior valor de densidade local, alcançou ou superou o limiar $\gamma_{\max} \in \mathbb{R}$. Em caso positivo a regra será ativada e o dado receberá a devida classificação proporcionada pela regra. Caso contrário, um novo grupo é adicionado à base de regras.

Algoritmo 1: Procedimento do sistema *fuzzy* AnYa.

Escolher γ_{\max} e inicializar a base de regras com v_1 ;

para $k = 2, 3, \dots$ **faça**

para $i = 1, 2, \dots, c$ **faça**

 | Calcular a densidade local $\gamma_i(k)$ com (12);

fim

se $\max_{j=1, \dots, c(k)} \{\gamma_j(k)\} < \gamma_{\max}$ **faça**

 | Criar um novo grupo e inicializar com $v(k)$;

se não

 | Adicionar o dado de entrada ao grupo com maior densidade local;

 | Atualizar valor médio com (13);

 | Atualizar a covariância com (14) e (15);

fim

fim

3. SOLUÇÃO DA PROPOSTA

O método híbrido de estimação de falhas proposto nesse trabalho aplica observadores de entrada desconhecida para geração de resíduos e modelagem *fuzzy* granular para avaliação de resíduos. Desse modo, na fase *offline*, o algoritmo de aprendizagem AnYa é aplicado para treinar uma base de regras *fuzzy* e na fase *online* essa base de regras é utilizada para estimar a magnitude de falha.

Nesse sentido, considere o sistema linear discreto sujeito à falha nos atuadores,

$$\begin{aligned} x(k+1) &= Ax(k) + Bu^f(k) + \zeta(k), \\ y(k) &= Cx(k) + \eta(k), \end{aligned} \quad (16)$$

em que $\zeta(k)$ e $\eta(k)$ são ruídos com média nula e matrizes de covariância Q e R . $x(k) \in \mathbb{R}^n$ é o vetor de estados, $y(k) \in \mathbb{R}^m$ é o vetor de saídas, $u^f(k) \in \mathbb{R}^m$ é o vetor de entradas com falha, tal que

$$u_j^f(k) = [1 + f_j(k)]u_j(k), \quad (17)$$

para $j = 1, 2, \dots, m$, em que $u(k) \in \mathbb{R}^m$ é o vetor de sinal de controle e $f(k) \in [-1, 1]^m$, tal que

$$f(k) = [f_1 \ f_2 \ \dots \ f_m]^T,$$

é o vetor de falha ao qual os atuadores estão sujeitos, tal que a falha $f_j(k)$ com $j = 1, \dots, m$ é a falha submetida ao atuador j . Desse modo, a fim de estimar a matriz de falha $f(k)$ é proposto o diagrama híbrido de detecção de falhas apresentado na Figura 2.

Conforme pode-se verificar no diagrama da Figura 2, adota-se um banco de observadores como proposto por Chen e Patton (1996). Esse banco de observadores é composto por m observadores de entrada desconhecida, tal que,

$$\begin{aligned} z_j(k+1) &= F_j z_j(k) + T_j B_j u_j(k) + K_j y(k) \\ \text{UIO}_j: \hat{x}_j(k) &= z_j(k) + H_j y(k) \\ r_j(k) &= y(k) - C \hat{x}_j(k) \end{aligned} \quad (18)$$

para $i = 1, \dots, m$, em que $z_j(k)$ é o estado do observador j , $\hat{x}_j(k)$ é o estado estimado pelo observador j , B_j é a coluna j da matriz B , $u_j(k)$ é o sinal de controle do atuador i e $r_j(k)$ é o vetor de resíduos do observador j , que representa a compatibilidade do atuador j com o modelo. Além disso, a matriz de entradas desconhecidas E_j é a matriz B com exceção da coluna j .

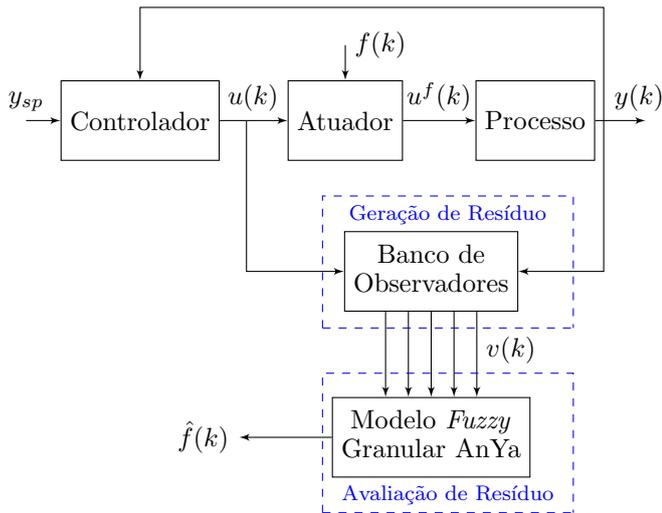


Figura 2. Diagrama do sistema de estimação de falhas proposto, em que $\hat{f}(k)$ é a estimação da falha.

Dessa forma, o vetor de resíduos é utilizado na fase *offline* para realizar um treinamento com o algoritmo AnYa, recebendo como entrada $v(k) \in \mathbb{R}^{m^2}$, tal que,

$$v(k) = \begin{bmatrix} \frac{1}{u_1(k)}(r_1(k)) \\ \frac{1}{u_2(k)}(r_2(k)) \\ \vdots \\ \frac{1}{u_m(k)}(r_m(k)) \end{bmatrix}. \quad (19)$$

Por outro lado, na fase *online*, $v(k)$ é aplicado para estimar o valor percentual de falha nos atuadores utilizando um procedimento de mudança de base no espaço \mathbb{R}^{m^2} .

3.1 Fase offline

Na fase *offline*, ou fase de treinamento, é criada uma base de regras *fuzzy* com $m + 1$ regras utilizando $m + 1$ conjuntos de dados com configurações de falhas diferentes: um conjunto obtido sem falhas nos atuadores (X_0) e m conjuntos obtidos com falhas críticas de cada um dos atuadores (X_1, \dots, X_m). Assim, cada um dos grupos X_1, \dots, X_m possui um vetor de parâmetros de falhas críticas, dado por:

$$\begin{aligned} \theta_1 &= [F_1 \ 0 \ \dots \ 0]^T, \\ \theta_2 &= [0 \ F_2 \ \dots \ 0]^T, \\ &\vdots \\ \theta_m &= [0 \ 0 \ \dots \ F_m]^T, \end{aligned} \quad (20)$$

em que θ_j é o vetor de parâmetros de falhas associado ao grupo X_j e F_j é o valor de falha aplicado ao atuador j durante a obtenção dos dados na criação do grupo X_j . Além disso, a cada grupo é associado um vetor

$$\mathbf{v}_j = \mu_j - \mu_0, \quad (21)$$

para $j = 1, 2, \dots, m$, tal que $\mathbf{v}_j \in \mathbb{R}^{m^2}$, em que $\mu_j \in \mathbb{R}^{m^2}$ é o valor médio dos dados do grupo, ou seja, um vetor que relaciona o valor médio do grupo que indica processo sem falha a cada um dos grupos indicadores de falha.

3.2 Fase online

Na fase *online* os vetores apresentados em (20) e (21) são aplicados para estimar a falha $f(k)$ como

$$\hat{f}(k) = \sum_{i=1}^m \alpha_i(k) \theta_i, \quad (22)$$

em que $\alpha(k) = [\alpha_1(k) \ \dots \ \alpha_m(k)]^T \in \mathbb{R}^m$ é o vetor de grau de ativação de cada nuvem de dados computado como

$$\alpha(k) = \mathbf{V}^+(v(k) - \mu_0), \quad (23)$$

em que $\mathbf{V}^+ = (\mathbf{V}^T \mathbf{V})^{-1} \mathbf{V}^T$ é a pseudoinversa de $\mathbf{V} = [\mathbf{v}_1 \ \dots \ \mathbf{v}_m] \in \mathbb{R}^{m^2 \times m}$. Ou seja, o vetor $v(k) - \mu_0 \in \mathbb{R}^{m^2}$ é reescrito na base formada pelos vetores $\mathbf{v}_1, \dots, \mathbf{v}_m$, obtendo-se o grau de ativação de cada vetor da base.

4. RESULTADOS

A fim de avaliar o desempenho e comportamento da abordagem híbrida de estimação de falhas proposta neste trabalho, submete-se a mesma a duas avaliações distintas. No primeiro teste de avaliação, busca-se encontrar a solução para um exemplo numérico. Já na segunda parte da avaliação, submete-se a topologia a um experimento prático em um sistema de tanques acoplados, o qual será condicionado a falhas nos atuadores.

4.1 Exemplo Numérico

Considere o sistema:

$$\begin{aligned} x(k+1) &= \underbrace{\begin{bmatrix} 0,5 & -0,1 & 0,2 \\ -0,1 & 0,5 & 0,1 \\ -0,5 & 0,3 & 0,5 \end{bmatrix}}_A x(k) + \underbrace{\begin{bmatrix} 1 & 0 \\ 0,1 & 1 \\ 0 & 0,1 \end{bmatrix}}_B u^f(k) + \zeta(k) \\ y(k) &= \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}}_C x(k) + \eta(k) \end{aligned} \quad (24)$$

em que $x(k) \in \mathbb{R}^3$, $y(k) \in \mathbb{R}^2$, $u^f(k) \in \mathbb{R}^2$ são os vetores de estados, saídas e entradas do sistema, respectivamente. $\zeta(k)$ e $\eta(k)$ são ruídos brancos com variância $S^2 = 0,05$. Esse sistema apresenta um controlador por realimentação de estados com ação integral, tal que a lei de controle é dada por:

$$u(k) = \mathbf{K} [x(k)^T \ x_a(k)^T]^T \quad (25)$$

em que $\mathbf{K} = [K_p \ K_i]$ é o ganho de realimentação de estado e $x_a(k)$ é o estado do integrador. O ganho \mathbf{K} foi projetado para alocar os autovalores da malha fechada em 0,55, 0,56, 0,57, 0,58 e 0,59 (valores arbitrados), sendo obtido:

$$K_p = \begin{bmatrix} -0,3001 & 0,0634 & -0,1986 \\ 0,1002 & -0,3481 & -0,0808 \end{bmatrix} \text{ e } K_i = \begin{bmatrix} 0,1628 & 0,0134 \\ -0,0053 & 0,1768 \end{bmatrix}.$$

A dinâmica de $x_a(k)$ é dada por:

$$x_a(k+1) = x_a(k) + (y_{sp} - y(k)), \quad (26)$$

com $y_{sp} = [10 \ 10]^T$ a referência para a saída do sistema. Os resíduos são obtidos a partir do projeto de dois observadores de entrada desconhecida, (18), com alocação de autovalores em 0,5, 0,53 e 0,56, resultando em ganhos:

$$L_1 = \begin{bmatrix} -0,06999 & -0,09975 \\ 0,00001 & -0,53 \\ -0,4865 & 0,24997 \end{bmatrix} \text{ e } L_2 = \begin{bmatrix} -0,51547 & -0,04749 \\ -0,1453 & -0,05473 \\ -0,50004 & 0,29999 \end{bmatrix},$$

que, por meio das equações (4)-(8), resulta em:

$$F_1 = \begin{bmatrix} 0,56999 & -0,00025 & 0,2 \\ -0,00001 & 0,53 & 0 \\ -0,0035 & 0,00003 & 0,49 \end{bmatrix}, T_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & -0,1 & 1 \end{bmatrix},$$

$$K_1 = \begin{bmatrix} -0,06999 & -0,08 \\ 0,00001 & 0 \\ -0,4865 & 0,299 \end{bmatrix}, H_1 = \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 0 & 0,1 \end{bmatrix},$$

$$F_2 = \begin{bmatrix} 0,5303 & -0,0029 & -0,00792 \\ -0,0032 & 0,55967 & 0,07921 \\ 0,00004 & 0 & 0,5 \end{bmatrix}, T_2 = \begin{bmatrix} 0,009 & -0,099 & 0 \\ -0,099 & 0,99 & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

$$K_2 = \begin{bmatrix} 0,0093 & 0,00498 \\ -0,09307 & -0,0495 \\ -0,5 & 0,3 \end{bmatrix}, H_2 = \begin{bmatrix} 0,99 & 0,099 \\ 0,099 & 0,0099 \\ 0 & 0 \end{bmatrix}.$$

O algoritmo AnYa recebe como entrada $v(k) \in \mathbb{R}^4$, tal que

$$v(k) = \begin{bmatrix} \frac{1}{u_1(k)}(r_1(k)) \\ \frac{1}{u_2(k)}(r_2(k)) \end{bmatrix}, \quad (27)$$

em que $r_1(k), r_2(k) \in \mathbb{R}^2$ e é configurado com o limiar $\gamma_{\max} = 0,70$. Durante a fase de treinamento (*offline*), são definidas duas falhas críticas possíveis para os atuadores, tal que: $\theta_1 = [-0,8 \ 0]^T$ e $\theta_2 = [0 \ -0,8]^T$. Desse modo, inicia-se a fase de treinamento, a fim de definir a base de regras, $X = [X_0 \ X_1 \ X_2]$. Note que o grupo X_0 é o grupo formado na ausência de falha, ou seja, $\theta_0 = [0 \ 0]^T$. Na sequência, o algoritmo é submetido as falhas definidas por:

$$f(k) = \begin{bmatrix} f_1 \\ f_2 \end{bmatrix} = \begin{cases} \theta_0, & \text{se } 0 < k < 500, \\ \theta_1, & \text{se } 500 < k < 1000, \\ \theta_2, & \text{se } 1000 < k < 1500, \end{cases} \quad (28)$$

e é obtido o resultado ilustrado na Figura 3, em que

$$\mathbf{v}_1 = \begin{bmatrix} -1,8323496 \\ 0,0000209 \\ 0,00071688 \\ -0,00716456 \end{bmatrix} \text{ e } \mathbf{v}_2 = \begin{bmatrix} -0,00906316 \\ 0,00000024 \\ 0,18220237 \\ -1,82084203 \end{bmatrix}$$

são os vetores obtidos na fase de treinamento.

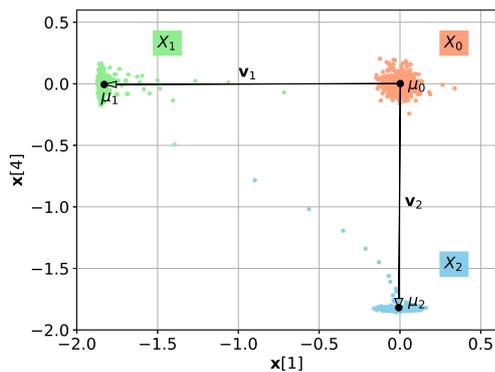


Figura 3. Ilustração no \mathbb{R}^2 do treinamento do exemplo numérico, em que $v[1]$ e $v[4]$ são a primeira e quarta coordenadas da entrada de dados $v \in \mathbb{R}^4$, X_0 é o grupo obtido para o processo sem falha, X_1 e X_2 são os grupos obtidos para o processo com falha e μ_0, μ_1 e μ_2 são os valores médios dos grupos X_0, X_1 e X_2 .

Por outro lado, na fase *online*, a estimação de falha é

$$\hat{f}(k) = \begin{bmatrix} \hat{f}_1 \\ \hat{f}_2 \end{bmatrix} = \alpha_1(k) \begin{bmatrix} -0,8 \\ 0 \end{bmatrix} + \alpha_2(k) \begin{bmatrix} 0 \\ -0,8 \end{bmatrix}, \quad (29)$$

em que $\alpha(k) = [\alpha_1(k) \ \alpha_2(k)]^T$ é calculado com (23) aplicando o vetor $\mu_0 = [0 \ 0 \ 0 \ 0]^T$ e a matriz $\mathbf{V} = [\mathbf{v}_1 \ \mathbf{v}_2]$,

tal que $\mathbf{V} \in \mathbb{R}^{4 \times 2}$. Os dados da Figura 3 também ilustram que o método híbrido proposto é capaz de detectar a falha com rapidez, mesmo na presença de ruídos. Assim, o resultado da estimação de falha realizada na fase *online*, conforme apresentado na Figura 4, é obtido aplicando uma sequência de falhas

$$f(k) = \begin{bmatrix} f_1 \\ f_2 \end{bmatrix} = \begin{cases} [0 \ 0]^T, & \text{se } 0 < k < 500, \\ [0,5 \ 0,5]^T, & \text{se } 500 < k < 1000, \\ [0 \ 0,75]^T, & \text{se } 1000 < k < 1500, \\ [0,75 \ 0]^T, & \text{se } 1500 < k < 2000, \end{cases} \quad (30)$$

em que f_1 e f_2 são as falhas nos atuadores 1 e 2 do sistema, respectivamente.

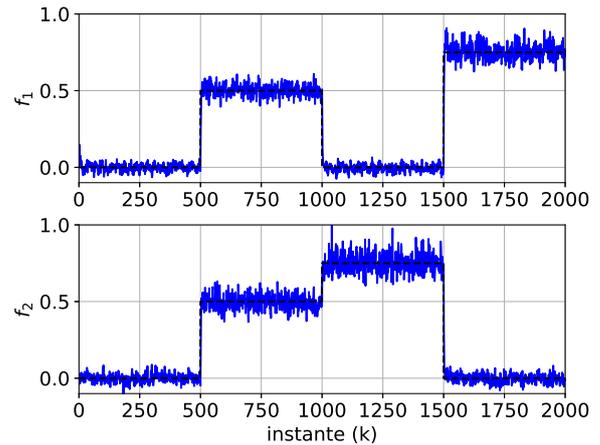


Figura 4. Estimação de falha pelo método proposto, sendo: magnitude da falha (**linha preta**) e valor estimado da falha (**linha azul**).

Uma análise qualitativa da estimação apresentada na Figura 4 mostra que a estimação possui média próxima da magnitude da falha imposta ao sistema, porém com certa variância ocasionada pelo ruído branco. Assim, para avaliar quantitativamente o erro de estimação, a média e o desvio padrão em cada degrau da sequência de falhas são calculados e apresentados na Tabela 1, em que $e_1 = \hat{f}_1 - f_1$ e $e_2 = \hat{f}_2 - f_2$ são os erros de estimação e $t_0(k)$ e $t_1(k)$ são o tempo inicial e final de cada degrau.

Tabela 1. Erro de estimação de cada degrau da sequência de falhas na validação numérica.

$t_0(k)$	$t_1(k)$	e_1		e_2	
		Média	Desvio Padrão	Média	Desvio Padrão
0	500	0,0004	0,0444	0,0009	0,043
500	1000	0,0004	0,0698	-0,0035	0,0755
1000	1500	-0,0015	0,048	0,0026	0,0894
1500	2000	0,0008	0,0913	0,0017	0,0583

De acordo com os dados da Tabela 1, pode-se verificar que a média do erro de estimação é próxima de zero ($\approx 10^{-3}$), o que condiz com o observado no gráfico da Figura 4. Verifica-se que o desvio padrão dos erros é menor que 0,1, sendo da ordem de 50% do desvio padrão do ruído de medição usado nas simulações.

4.2 Sistema de Tanques Acoplados - Teste Experimental

O sistema de tanques acoplados utilizado para validação do método proposto é apresentado na Figura 5a. Este

é composto por quatro tanques com capacidade de 200l cada, equipados com sensor de pressão *26PCBFA6D* para medição da coluna de líquido, dois reservatórios acoplados com 400l de capacidade cada e duas bombas hidráulicas trifásicas com potência de 2 CV, comandadas por dois inversores WEG CFW09 (Oliveira et al., 2020). O controlador do processo opera em uma Raspberry Pi 3 conectada ao controlador lógico programável (CLP) Simatic S7-300, conectados via protocolo Ethernet a uma interface gráfica baseada em Python (Sousa et al., 2018). Para validar a abordagem proposta adota-se a configuração não linear apresentada na Figura 5b, em que h_2, h_3 e $h_4 \in [0, 70]$ cm são as alturas da coluna de líquido nos tanques 2, 3 e 4 e $B1$ e $B2$ são as duas bombas hidráulicas do processo.

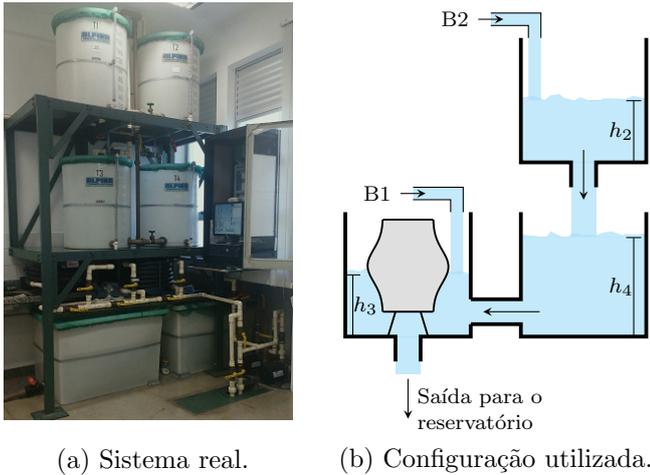


Figura 5. Sistema de tanques interconectados.

Nesse processo, busca-se regular a coluna de líquido nos tanques quando as bombas são configuradas para operar com 50% da potência nominal. Para essa configuração, os tanques apresentam comportamento dinâmico descrito por:

$$x(k+1) = \underbrace{\begin{bmatrix} 0,997 & 0 & 0 \\ 0 & 0,94 & 0,034 \\ 0,003 & 0,011 & 0,989 \end{bmatrix}}_A x(k) + \underbrace{\begin{bmatrix} 0 & 0,005 \\ 0,018 & 0 \\ 0 & 0 \end{bmatrix}}_B u^f(k) \quad (31)$$

$$y(k) = \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}}_C x(k) + \eta(k)$$

em que $x(k) = [h_2 \ h_3 \ h_4]^T$, $y(k) = [h_2 \ h_3]^T$, $u^f(k) = [u_{B1} \ u_{B2}]^T \in [0, 100]$ %.

O procedimento de estimação de falhas proposto nesse texto, conforme apresentado na Seção 3, foi aplicado no sistema real de tanques interconectados. Os resíduos são obtidos a partir do projeto de dois observadores de entrada desconhecida, (18), com ganhos:

$$L_1 = \begin{bmatrix} -0,98019 & -0,000003 \\ -0,00004 & -0,03098 \\ 0,00265 & 0,01036 \end{bmatrix} \text{ e } L_2 = \begin{bmatrix} 0,02689 & -0 \\ 0 & -0,98019 \\ 1 & 1 \end{bmatrix},$$

e com auxílio das equações (4)-(8). Na fase *offline*, o treinamento da base de regras é realizado experimentalmente com uma sequência de falhas

$$f(k) = \begin{bmatrix} f_1 \\ f_2 \end{bmatrix} = \begin{cases} \begin{bmatrix} 0 & 0 \end{bmatrix}^T, & \text{se } 0 < k < 1000, \\ \begin{bmatrix} 0,4 & 0 \end{bmatrix}^T, & \text{se } 1000 < k < 2000, \\ \begin{bmatrix} 0 & 0,4 \end{bmatrix}^T, & \text{se } 2000 < k < 3000, \end{cases} \quad (32)$$

em que f_1 é a falha aplicada na bomba B1 e f_2 é a falha aplicada na bomba B2, e é obtido como resultado os vetores

$$v_1 = \begin{bmatrix} 0,00007219 \\ 0,42594653 \\ -0,00012948 \\ 0 \end{bmatrix} \text{ e } v_2 = \begin{bmatrix} -0,00000072 \\ -0,00920185 \\ 0,27620096 \\ 0 \end{bmatrix}.$$

A fase *online*, por sua vez, é realizada experimentalmente com uma determinada sequência de falhas e a falha é estimada por

$$\hat{f}(k) = \begin{bmatrix} \hat{f}_1 \\ \hat{f}_2 \end{bmatrix} = \alpha_1(k) \begin{bmatrix} 0,4 \\ 0 \end{bmatrix} + \alpha_2(k) \begin{bmatrix} 0 \\ 0,4 \end{bmatrix}, \quad (33)$$

em que $\alpha(k) = [\alpha_1(k) \ \alpha_2(k)]^T$ é calculado com (23) aplicando o vetor

$$\mu_0 = [0,00007 \ 0,40881 \ 0,19733 \ 0]^T$$

e a matriz $V = [v_1 \ v_2]$, tal que $V \in \mathbb{R}^{4 \times 2}$. O resultado da estimação aplicando (33) é apresentado na Figura 6.

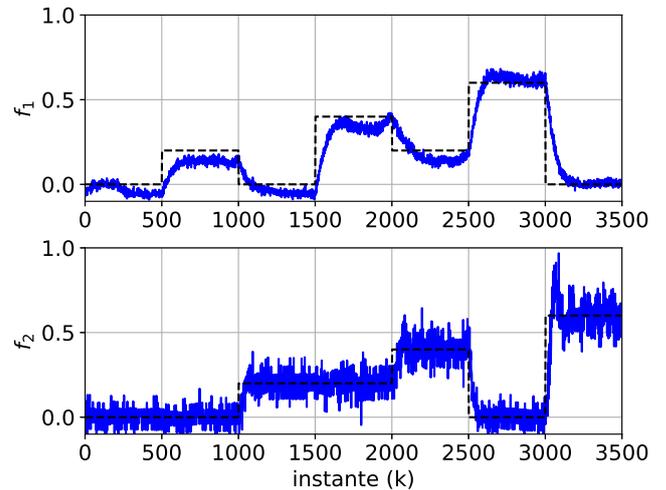


Figura 6. Estimação de falha do exemplo experimental: magnitude da falha (linha preta) e estimação da falha (linha azul).

Nota-se que a detecção da ocorrência de falha é bastante rápida (quase instantânea). Já a determinação da intensidade das falhas tem uma dinâmica que é dirigida pelos autovalores alocados nos observadores UIO_1 e UIO_2 . Além disso, a média e o desvio padrão do erro de estimação em cada degrau da sequência de falhas é calculado e apresentado na Tabela 2, em que e_1 é o erro de estimação da falha f_1 , e_2 é o erro de estimação da falha f_2 e $t_0(k)$ e $t_1(k)$ são o tempo inicial e final de cada degrau.

É possível observar na Figura 6 e na Tabela 2 que, diferente da validação numérica, o erro de estimação não teve média próxima de zero. Isso aconteceu no caso experimental devido às perturbações e não linearidades não consideradas no modelo linear utilizado para projetar os observadores e treinar o modelo *fuzzy* granular. Ainda assim, os valores obtidos são compatíveis com a amplitude do ruído de medição verificado no processo. A existência de diferenças entre

Tabela 2. Erro de estimação de cada degrau da sequência de falhas na validação experimental.

$t_0(k)$	$t_1(k)$	e_1		e_2	
		Média	Desvio Padrão	Média	Desvio Padrão
0	500	-0,0715	0,0424	-0,0017	0,047
500	1000	-0,0797	0,0591	-0,0015	0,0422
1000	1500	-0,0361	0,0635	-0,0248	0,073
1500	2000	-0,1156	0,1183	-0,0011	0,0539
2000	2500	-0,0053	0,0902	-0,0217	0,0817
2500	3000	-0,1042	0,0993	0,0394	0,1132
3000	3500	0,1201	0,1747	-0,0398	0,1785

o processo real e o modelo linear usado para representá-lo sugere o potencial do método híbrido proposto para ser aplicado em sistemas mais complexos, nos quais há significativa diferenças entre modelo e processo.

5. CONCLUSÃO

Foi apresentada uma nova abordagem híbrida para detecção e estimação de falhas em atuadores de processos. Essa abordagem foi desenvolvida associando as técnicas de observadores de entrada desconhecida e a modelagem *fuzzy* granular. A partir da criação de uma base de regras, a estrutura proposta consegue monitorar em tempo real o comportamento do processo, detectando a ocorrência e estimando a intensidade de tais falhas de atuadores. A solução proposta foi avaliada por meio de dois experimentos, um numérico e outro experimental, obtendo-se resultados que sugere sua aplicação em sistemas de maior porte. Foram feitas avaliações qualitativas e quantitativas das informações geradas, permitindo concluir que a proposta identifica e estima falhas em velocidade compatível com os sistemas experimentados. Além disso, a precisão da estimativa da intensidade de falha é compatível com o nível de ruído existente nas medições. Espera-se prosseguir as investigações dessa proposta frente a erros mais significativos de modelagem e também para tratar sistemas com parâmetros variantes no tempo.

REFERÊNCIAS

- Andonovski, G., Blažič, S., e Škrjanc, I. (2019). Evolving fuzzy model for fault detection and fault identification of dynamic processes. In *Predictive Maintenance in Dynamic Systems*, 269–285. Springer.
- Angelov, P. e Yager, R. (2011). Simplified fuzzy rule-based systems using non-parametric antecedents and relative data density. In *2011 IEEE Workshop on Evolving and Adaptive Intelligent Systems (EAIS)*, 62–69. IEEE.
- Beard, R.V. (1971). *Failure accommodation in linear systems through self-reorganization*. Ph.D. thesis, Massachusetts Institute of Technology.
- Chen, C.T. (2013). *Linear system theory and design*, volume 4. Oxford university press New York.
- Chen, J. e Patton, R.J. (1996). Optimal filtering and robust fault diagnosis of stochastic systems with unknown disturbances. *IEE Proceedings-Control Theory and Applications*, 143(1), 31–36.
- Chen, J. e Patton, R.J. (2012). *Robust model-based fault diagnosis for dynamic systems*, volume 3. Springer Science & Business Media.
- Chen, J. e Zhang, H. (1991). Robust detection of faulty actuators via unknown input observers. *International Journal of Systems Science*, 22(10), 1829–1839.
- Ding, S.X. (2008). *Model-based fault diagnosis techniques: design schemes, algorithms, and tools*. Springer Science & Business Media.
- Gao, Z., Cecati, C., e Ding, S.X. (2015). A survey of fault diagnosis and fault-tolerant techniques—part i: Fault diagnosis with model-based and signal-based approaches. *IEEE Transactions on Industrial Electronics*, 62(6), 3757–3767.
- Junior, E.Q.C., Maitelli, A.L., Lopes, K.R., Andrade, A.C.C., e De Lima, B.F.A. (2019). Observador de entrada desconhecida robusto aplicado à estimação de falhas em atuadores em sistemas fuzzy takagi-sugeno no tempo discreto. *Anais da Sociedade Brasileira de Automática*, 1(1).
- Lemos, A., Caminhas, W., e Gomide, F. (2013). Adaptive fault detection and diagnosis using an evolving fuzzy classifier. *Information sciences*, 220, 64–85.
- Oliveira, L., Bento, A., Leite, V.J., e Gomide, F. (2020). Comparisons of robust methods on feedback linearization through experimental tests. *IFAC-PapersOnLine*, 53(2), 7983–7988. 21th IFAC World Congress.
- Sousa, A., Leite, V.J., e Rubio Scola, I. (2018). Affordable control platform with mpc application. *Studies in Informatics and Control*, 27(3), 265–274.