



UM MODELO ADAPTATIVO PARA DETECÇÃO E CLASSIFICAÇÃO DE FALHAS APLICADO NO TENNESSEE EASTMAN PROCESS

Laércio Ives Santos

Instituto Federal de Norte de Minas Gerais
Rua Dois, 300 - Village do Lago I - Montes Claros/MG
laercio.santos@ifnmg.edu.br

Reinaldo Martinez Palhares

Departamento de Engenharia Eletrônica, UFMG
Av. Antônio Carlos, 6627, Pampulha, Belo Horizonte, MG, Brasil
rpalhares@ufmg.br

Marcos Flávio Silveira Vasconcelos D'Angelo, João Batista Mendes

Universidade Estadual de Montes Claros
Departamento de Ciência da Computação, Av. Rui Braga, sn, Vila Mauriceia, Montes Claros,
Minas Gerais, 39401-089, Brasil
marcos.dangelo@unimontes.br, joao.mendes@unimontes.br

Tarcisio Francisco Batista Filho, Jéssica Flaviane Ferreira

Universidade Estadual de Montes Claros
Bacharelado em Engenharia de Sistemas, Av. Rui Braga, sn, Vila Mauriceia, Montes Claros,
Minas Gerais, 39401-089, Brasil
tarcisio.batista@unimontes.br, jessicaferreirall@gmail.com

RESUMO

Este trabalho apresenta um modelo adaptativo para detecção e classificação de falhas em sistemas dinâmicos baseado em duas etapas: na primeira etapa a abordagem proposta utiliza um método qualitativo para detecção de falhas e na segunda etapa propõe variações em algoritmos de agrupamento para a tarefa de classificação de falhas. Para verificar a eficiência da abordagem, a mesma foi aplicada no *Tennessee Eastman Process*.

PALAVRAS CHAVE. Detecção de Falhas, Classificação de Falhas, Agrupamento.

OA – Outras aplicações em PO

ABSTRACT

This paper presents an adaptive model for fault detection and classification in dynamic systems based on two stage: In the first stage the proposed model uses a qualitative method to detect faults and in the final stage, it proposes variations in clustering methods for the fault classification task. In order to verify the efficiency of the proposed model, experiments was performed using the *Tennessee Eastman Process*.

KEYWORDS. Fault Detection , Fault Classification, Clustering.

OA – Others Applications in PO



1. Introdução

A ocorrência de falhas em sistemas dinâmicos industriais tem levado pesquisadores a desenvolver tecnologias capazes de detectar e classificar falhas de forma segura. Uma boa detecção pode evitar perdas de materiais, desastres ambientais e até mesmo acidentes envolvendo seres humanos. Tal demanda deu origem a necessidade de sistemas com a capacidade de supervisão que integram os sistemas de FDI (*Fault Detection and Isolation*) que integram os processos e por isso, a detecção de falhas pode ser vista como uma ação preventiva.

Os três tipos principais de falhas em sistemas dinâmicos são: falhas abruptas, falhas incipientes e falhas intermitentes [Simani et al., 2003]. As falhas abruptas acontecem em um pequeno intervalo de tempo e resultam em grandes desvios nas condições normais do processo; enquanto as falhas incipientes acontecem de forma gradativa e podem demorar muito tempo para serem detectadas; já falhas intermitentes ocorrem repetidamente nos sistemas e são modeladas como uma combinação de impulsos com diferentes amplitudes em diferentes instantes de tempo.

Para resolver o problema de detecção e classificação de falhas em sistemas dinâmicos, várias abordagens são encontradas na literatura. Essas abordagens se dividem em *Abordagens Qualitativas* [Venkatasubramanian et al., 2003a] [Venkatasubramanian et al., 2003b] e *Abordagens Quantitativas* [Venkatasubramanian et al., 2003c]. Os métodos baseados em abordagens quantitativas se baseiam em analisar resíduos de sinais provenientes de modelos matemáticos com sinais reais [Li et al., 2015], isto é, comparam os valores instantâneos das variáveis de estado do processo, medidas por sensores, com valores calculados analiticamente a partir do modelo [Takahashi e Peres, 1999] [Kalman, 1960]. Já métodos baseados em abordagens qualitativas utilizam um conjunto de relações causais, um exemplo é o uso de uma base de conhecimento para extrair informações sobre o estado de operação do processo ou sistema [Fábrega et al., 2014] [Jiang et al., 2013] [D'Angelo e Costa Jr, 2001] [D'Angelo et al., 2016] [D'Angelo et al., 2014], outro exemplo são os modelos baseados em análise qualitativa de tendências [Maurya et al., 2007] [Lemos, 2007] [Baccarini et al., 2010] [D'Angelo et al., 2011a] [D'Angelo et al., 2016].

Assim, este trabalho tem como objetivo propor um modelo capaz de detectar e classificar falhas, apresentando bons resultados no problema de detecção e classificação de falhas do *Tennessee Eastman Process* [Downs e Vogel, 1993]. Para tanto, a metodologia classifica falhas de forma adaptativa, visando maximizar a taxa de acertos e minimizar a quantidade de protótipos criados. A metodologia apresentada aqui se diferencia da abordagem apresentada em [Santos et al., 2015] com a inserção de três características: adição de um mecanismo de estagnação de partículas proposto no classificador de falhas, utilização de um método de geração da população inicial e utilização de um método qualitativo para detecção de falhas.

A parte restante deste trabalho é composta por mais 3 seções. Na seção 2 a abordagem proposta para classificação de falhas em sistemas dinâmicos é apresentada e, para fundamentá-la, é feita uma breve revisão sobre os algoritmos de agrupamento *Constructive Particle Swarm Clustering* e *Subtractive Clustering*. Na seção 3 a abordagem proposta é aplicada em um estudo de caso e os resultados obtidos são comparados com resultados de outras duas abordagens. Finalmente, na seção 4 são discutidas as considerações finais.

2. Modelo Utilizado Para Classificação de Falhas

Esta seção apresenta o Modelo Proposto para classificação de falhas. Nas subseções 2.1 e 2.2 serão feitas revisões dos métodos *Constructive Particle Swarm Clustering* e *Subtractive Clustering*, respectivamente e finalmente, na subseção 2.3 será apresentado o classificador de falhas proposto.

2.1. *Constructive Particle Swarm Clustering* (cPSC)

Em [Cohen e de Castro, 2006] é apresentado o algoritmo de agrupamento PSC (*Particle Swarm Clustering*). Inspirado no PSO, o PSC utiliza um termo de auto-organização [Kohonen, 1990] no mecanismo de movimentação de partículas, além dos termos de memória e cooperação.



No PSC cada padrão de entrada é apresentado ao enxame de partículas e a partícula com maior similaridade com o padrão, se move na direção dele. A movimentação da partícula é influenciada pelos temas Memória, Cooperação e de Auto-organização. Entretanto, assim como a maioria dos algoritmos de agrupamento baseado em partição, no PSC a quantidade (k) de grupos deve ser definido a priori.

Para atualizar a velocidade de uma partícula, usa:

$$V_v(t+1) = \omega V_v(t) + \varphi_1 \otimes (Pbest_{vj}(t) - X_v(t)) + \varphi_2 \otimes (Gbest_J(t) - X_v(t)) + \varphi_3 \otimes (Y_j - X_v(t)) \quad (1)$$

E para atualizar a posição de uma partícula (X), faz-se:

$$X_v(t+1) = X_v(t) + V_v(t+1) \quad (2)$$

Na expressão (1): ω é o termo de inércia utilizado para evitar que partículas com velocidades muito altas explodam e se desloquem para fora do universo de discurso. Os termos φ_1 , φ_2 e φ_3 são vetores de pesos aleatórios, gerados em uma distribuição uniforme no intervalo $[0,1]$, usados para ponderar os temas de: memória, cooperação e auto-organização, respectivamente, de cada partícula. O $Pbest$ representa a melhor (maior similaridade) posição da partícula em relação ao padrão de entrada, e o $Gbest$ representa a melhor posição de todas as partículas em relação ao padrão de entrada. Inicialmente, o $Pbest$ é definido como a posição inicial de cada partícula, a medida que a partícula se movimenta, a posição atual e o $Pbest$ daquela partícula são comparados e, caso a posição atual possua maior similaridade que o $Pbest$ em relação ao padrão de entrada, tal posição passa a ser o $Pbest$. Já o $Gbest$ é atualizado tomando sempre o melhor $Pbest$ para um determinado padrão de entrada.

Assim, baseado no PSC, o algoritmo *Constructive Particle Swarm Clustering* (cPSC) foi proposto, se diferenciando na capacidade de encontrar, de forma automática, o número de grupos, visto que no PSC essa informação era um parâmetro de entrada. Para resolver essa deficiência do algoritmo PSC, [Prior e de Castro, 2010] propõe o cPSC que, inspirado em [de Castro et al., 2003], adiciona os operadores de crescimento (clonagem) e poda de partículas, fazendo que ao final de sua execução exista um "número ótimo" de grupos.

O crescimento das partículas acontece da seguinte maneira: durante a execução do algoritmo é determinado o nível de concentração de cada partícula do enxame, como sendo, a quantidade de padrões de entrada que essa partícula representa. Se o nível de concentração de uma determinada partícula for maior que zero, essa partícula se torna candidata a ser clonada, e ainda, se a similaridade de uma partícula candidata com o padrão de entrada de maior similaridade com essa mesma partícula for maior que um limiar ε , então a partícula será clonada. Essa nova partícula (clone) será posicionada entre a partícula clonada e o padrão de entrada de maior similaridade. Já no processo de poda de partículas, é verificado o nível de concentração das partículas. Se uma partícula possui seu nível de concentração igual a zero em um número de iterações maior do que um limiar pré-definido, então esta partícula será eliminada do enxame.

O pseudocódigo do algoritmo cPSC é mostrado no algoritmo, 1.

Onde: S é o conjunto de partículas, $p.nc$ é o nível de concentração da partícula p .

2.2. Subtractive-Clustering

O método *Subtractive-Clustering* (SC), apresentado em [Chiu, 1996], busca um conjunto de centros que minimize a soma das distâncias de cada padrão ao centro mais próximo. Para isso, utiliza um mecanismo de subtração dos potenciais dos centros. Inicialmente, todos os padrões são tomados como possíveis centros e um valor (P_i) do potencial de cada centro é determinado utilizando a expressão:

$$P_i = \sum_{j=1}^{nc} e^{-\frac{4}{r_a^2} \|X_i - X_j\|^2} \quad (3)$$



Algoritmo 1: cPSC (Dados_Entrada, ε , ω)

```
1: S ← Gerar de forma aleatória o exame inicial;
2: while condição de para não for satisfeita do
3:   for cada partícula p ∈ S do
4:     p.nc ← ComputaNivelConcentração(p);
5:   end for
6:   for cada item  $Y_j \in \text{Dados\_Entrada}$  do
7:     Encontrar  $X_v \in S$  sendo  $X_v$  a mais similar partícula a  $Y_j$ ;
8:     Atualiza  $P_{best_{vj}}$  e  $G_{best_j}$ ;
9:     Atualiza  $V_v$  utilizando equação 1;
10:    Atualiza  $X_v$  utilizando equação 2;
11:  end for
12:  for cada partícula p ∈ S do
13:    if p.nc = 0 then
14:      Elimina_partícula(p);
15:    end if
16:  end for
17:  for cada partícula p ∈ S do
18:    if p.nc >  $\varepsilon$  then
19:      Clona_Partícula(p);
20:    end if
21:  end for
22:   $\omega \leftarrow \omega * 0.95$ ;
23: end while
```

Na expressão 3: nc a quantidade de centros iniciais (quantidade de padrões de treinamento) e r_a o raio do centro, quanto maior seu valor menor a quantidade de centros encontrados [Naresh et al., 2008].

Quando, os potenciais de todos os centros forem determinados, o centro com maior valor de P é selecionado para ser o primeiro centro (X_1^*). Em 3 percebe-se que centros em regiões mais densas têm maiores chances de serem escolhidos como primeiro centro.

A próxima etapa do agrupamento é diminuir os potenciais de todos os centros de acordo com:

$$P_i = P_i - P_1^* e^{-\frac{4}{r_b^2} \|X_i - X_1^*\|^2} \quad (4)$$

Sendo: P_1^* potencial do primeiro centro, X_i a posição do i -ésimo centro, X_1^* a posição do primeiro centro, r_b o raio entre o centro X_1 e X_i e geralmente se utiliza $1.25r_a$ [Chiu, 1996], f_1 uma fração predefinida do potencial de P_1 e quanto maior seu valor menor a quantidade de centros encontrada.

Nessa etapa, os potenciais de todos os centros, próximos do centro X_1 , diminuem em maior intensidade que de centros mais distantes. Com isso, as chances do centro X_1^* , ser selecionado em uma próxima iteração, passa a ser 0 e de centros próximos de X_1^* tende a 0.

Na versão utilizada neste trabalho, o SC separa os padrões de treinamento, de acordo com a classe, e para cada classe executa os passos a seguir:

1. Tome todos os dados de entrada como centros potenciais do grupo.
2. Para cada centro é calculado o potencial(P_i) utilizando (3);



3. Seleciona o centro com maior potencial como primeiro centro, $P_1^* = Max(P)$;
4. Recalcula P_i para todos os centros de acordo com (4);
5. O processo termina quando $P < f_1 P_1^*$ para todos os centros;

O problema desse método é que nem sempre os padrões, tomados como centros dos grupos, representam exatamente os centros dos grupos. Isso por que o centro selecionado tem a mesma coordenada de um dos dados de entrada e a escolha é influenciada por centros que estão em regiões mais concentradas.

A próxima subseção apresenta o modelo para classificação de falhas. Essa abordagem é baseada nos algoritmos SC e cPSC.

2.3. Classificador de Falhas

O objetivo principal deste trabalho, é apresentar um modelo para detecção e classificação de falhas que seja capaz de criar novos protótipos na tentativa de aumentar a taxa final de acertos de forma significativa sem, no entanto, aumentar significativamente a quantidade de classes. Para alcançar esse objetivo, o trabalho propõe modificações no algoritmo cPSC para maximizar a taxa de acerto e minimizar a quantidade de grupos, inserindo características supervisionadas, tendo em vista que o cPSC é um algoritmo de agrupamento (aprendizado não supervisionado), ou seja, o rótulo (classes) dos dados de entrada não são levados em consideração na formação dos grupos.

Quando uma partícula concentra uma quantidade razoável de padrões de entrada e tem uma baixa taxa de acertos, essa partícula tende a diminuir a taxa de acerto final do algoritmo. Dessa forma, a clonagem da partícula cria a possibilidade de, a soma da taxa de acerto da partícula clone somada à taxa de acerto da partícula clonada aumentar. Considerando tal fato, a primeira modificação foi realizada no critério de crescimento das partículas e na forma como a clonagem é feita. Como critérios de crescimento foram utilizados o nível de concentração e a taxa de acerto de uma determinada partícula, visto que os dados, aqui tratados, eram rotulados. Se uma determinada partícula possui nível de concentração maior que um limiar ε_1 e taxa de acerto menor que γ_1 então a partícula será clonada. A partícula clonada gera uma nova partícula e essa sofre uma pequena alteração na direção de uma de suas variáveis [Santos et al., 2015]. Na segunda modificação, foi introduzido o conceito de estagnação de partículas, que consiste em não movimentar partículas com taxas de acertos e concentrações consideradas boas. Portanto, caso uma partícula possua nível de concentração maior que um limiar ε_2 e taxa de acerto maior que um limiar γ_2 tal partícula não será movimentada naquela iteração do algoritmo. Esse mecanismo foi introduzido porque, no decorrer da apresentação dos padrões às partículas, as regiões com grandes concentração de padrões tendem a atrair mais partículas, e se nessas regiões os padrões pertencem a muitas classes, vai diminuir a taxa de acerto das partículas atraídas e conseqüentemente a taxa final do algoritmo. Por fim, foi utilizado um mecanismo de geração do exame inicial com o método *Subtractive-Clustering* apresentado em 2.2, em que os centros gerados pelo SC são utilizados para compor o exame inicial de partículas no algoritmo cPSC-Modificado.

A Figura 1 ilustra o processo de clonagem de partículas. A Figura (a) ilustra a situação em que uma partícula possui alta taxa de concentração e baixa taxa de acerto, a Figura (b) ilustra o momento que a clonagem acontece. Na Figura (a) existem 3 classes e apenas duas partículas, por ela percebe-se que a partícula 2 concentra uma quantidade de padrões maior que a partícula 1. Porém, a partícula 1 concentra padrões de apenas uma classe (classe 1) e possui uma taxa de acerto maior que a partícula 2 que concentra padrões de duas classes (classes 2 e 3). Dessa forma, a partícula 2 deve ser clonada. Quando acontece a clonagem, a partícula clone passa a concentrar parte dos padrões que eram concentrados pela partícula 2 (clonada) e isso pode aumentar a taxa de acerto final do algoritmo. E ainda, pode melhorar a disposição das partículas em relação aos padrões, fazendo a partícula clonada ir para o centro da classe 2.

O pseudocódigo do algoritmo cPSC_Modificado é mostrado no algoritmo, 2.

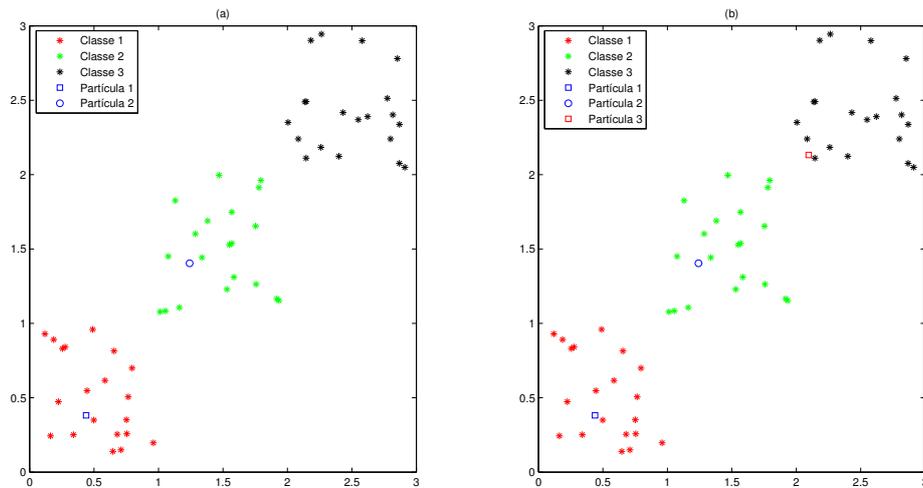


Figura 1: Ilustração do processo de clonagem de partículas

Algoritmo 2: cPSC_Modificado (Dados_Entrada, ε_1 , γ_1 , ε_2 , γ_2 , ω)

```

S ← subtractive-Clustering(Dados_Entrada);
2: while condição de para não satisfeita do
    for cada partícula p ∈ S do
4:     p.NC ← ComputaNivelConcentração(p);
        p.tAcerto ← ComputaTaxaAcerto(p);
6:     end for
    for cada item  $Y_j \in \text{Dados\_Entrada}$  do
8:     Encontrar  $X_v \in S$  sendo  $X_v$  a mais similar partícula a  $Y_j$ ;
        Atualiza  $Pbest_{vj}$  e  $Gbest_j$ ;
10:    if  $p.nc < \varepsilon_2$  OU  $p.tAcerto < \gamma_2$  then
        Atualiza  $V_v$  utilizando equação 1;
12:    Atualiza  $X_v$  utilizando equação 2;
        end if
14:    end for
    for cada partícula p ∈ S do
16:    if  $p.nc = 0$  then
        Elimina_partícula(p);
18:    end if
    end for
20:    for cada partícula p ∈ S do
        if  $p.nc > \varepsilon_1$  E  $p.tAcerto < \gamma_1$  then
22:        Clona_Partícula(p);
        end if
24:    end for
     $\omega \leftarrow \omega * 0.95$ ;
26: end while

```

Onde: $p.nc$ é o nível de concentração da partícula p , $p.tAcerto$ é taxa de acerto da partícula p .



O nível de concentração de uma partícula é obtido sendo a razão entre a quantidade de padrões que essa partícula concentra (a partícula é a mais similar ao padrão dentre todas as partículas) e a quantidade total de padrões. Esse valor é utilizado também na obtenção da taxa de acerto. Essa taxa é obtida sendo a razão entre a quantidade de padrões pertencentes à classe majoritária, dentre todas as classes existentes nos padrões que a partícula concentra, e a quantidade total de padrões concentrados pela partícula.

Os passos necessários para calcular o nível de concentração (rotina *ComputaNivelConcentracao(p)*) das partículas são:

- Faça iguais a 0 os níveis de concentração de todas as partículas;
- Para cada Dado de Entrada, Faça:
 1. Encontre a partícula (X_v) de maior similaridade com o padrão;
 2. Incremente em 1 o nível de concentração de $X_v.NC \leftarrow X_v.NC + 1$;
 3. Marque o padrão como concentrado em X_v ;

Os passos necessários para calcular a taxa de acerto das partículas, (rotina *ComputaTaxaAcerto(p)*) são:

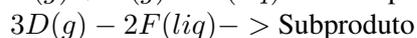
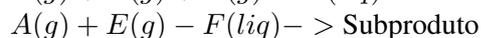
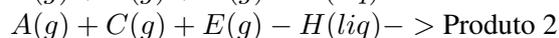
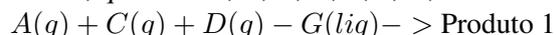
- Para cada Partícula (p_j) do enxame, Faça:
 1. Usando todos os padrões (N_j) concentrados em (p_j) obtenha a classe majoritária;
 2. encontre a quantidade de padrões concentrados em (p_j) que pertencem à classe majoritária (N_{maxC_j});
 3. Calcule a taxa de acerto da partícula, como: $p.tAcerto = \frac{N_{maxC_j}}{N_j}$

3. Aplicação no *Tennessee Eastman Process*

Nessa seção será apresentado o Sistema de Controle *Tennessee Eastman* bem como os resultados da aplicação do modelo para detecção e classificação de falhas no mesmo.

3.1. O *Tennessee Eastman Process*

O modelo descreve um processo petroquímico no qual são produzidos 2 produtos e quatro reagentes. Também estão presentes um inerte e um subproduto perfazendo um total de 8 componentes, que são: A, B, C, D, E, F, G, H. Com as seguintes reações:



Todas as reações são irreversíveis, exotérmicas e aproximadamente de primeira ordem em relação à concentração dos reagentes. Além disso, a produção de G exige maior sensibilidade a temperatura por ter uma energia de ativação mais elevada.

O processo é composto pelo reator, o condensador de produto, um separador de vapor líquido, um compressor de recirculação e um limpador de produto. Essas são as cinco maiores unidades de operação do processo. Resumidamente o processo acontece da seguinte forma:

- No reator os reagentes gasosos formam produtos líquidos;
- Na fase gasosa as reações dissolvidas na fase líquida são catalizadas por um catalizador não volátil;
- O separador de vapor líquido é usado para remover o calor da reação;



- Componentes não condensados são levados de volta ao reator por meio do compressor centrífugo;
- Componentes condensados vão para o Stripper afim de remover o restante dos reagentes;
- O inerte e o subproduto são removidos pelo separador de vapor líquido;

No *Tennessee Eastman* existem 12 variáveis manipuladas e 40 variáveis medidas. O processo contém 21 falhas no total, sendo 15 falhas conhecidas e 6 desconhecidas. A tabela 1 apresenta as 21 falhas. Para esse trabalho foram utilizados dados obtidos em <http://web.mit.edu/~braatzgroup>, onde podem ser encontrados 42 *data set* contendo dados para treinamento e testes. Cada *data set* contém dados de treinamento ou teste correspondente a uma das falhas do processo apresentadas na Tabela 1, e ainda 2 *data set* contendo dados para treinamento e teste da operação normal do sistema. No total, foram utilizados 480 (24 horas de simulação) situações de treinamento e 960 (48 horas de simulação) de testes, em que, cada *data set* foi amostrado em um intervalo de 3 minutos com as falhas sendo introduzidas após oito horas de simulação para o conjunto de testes e após uma hora para o conjunto de treinamento.

Tabela 1: Falhas no Processo *Tennessee Eastman*

Descrição
Taxa de alimentação A/C, Constante composição B (Corrente 4)
Composição B, Taxa constante A/C (Corrente 4)
Temperatura de alimentação D (Corrente 2)
Temperatura de entrada da água do refrigerador do reator
Temperatura de entrada da água do refrigerador do condensador
Perda de alimentação A (Corrente 1)
Perda de pressão de cabeçalho C / Disponibilidade Reduzida (Corrente 4)
Composição de alimentação A, B, C (Corrente 4)
Temperatura de alimentação D (Corrente 2)
Temperatura de alimentação C (Corrente 4)
Temperatura de entrada da água do refrigerador do reator
Temperatura de entrada da água do refrigerador do condensador
Cinética da Reação
Válvula de água de refrigeração do reator
Válvula de água de refrigeração do condensador
Desconhecida

3.2. Resultados

Essa subseção apresenta os resultados da aplicação do sistema de classificação de falhas no Tennessee Eastman e comparados os resultados de 2 classificadores clássicos apresentados em

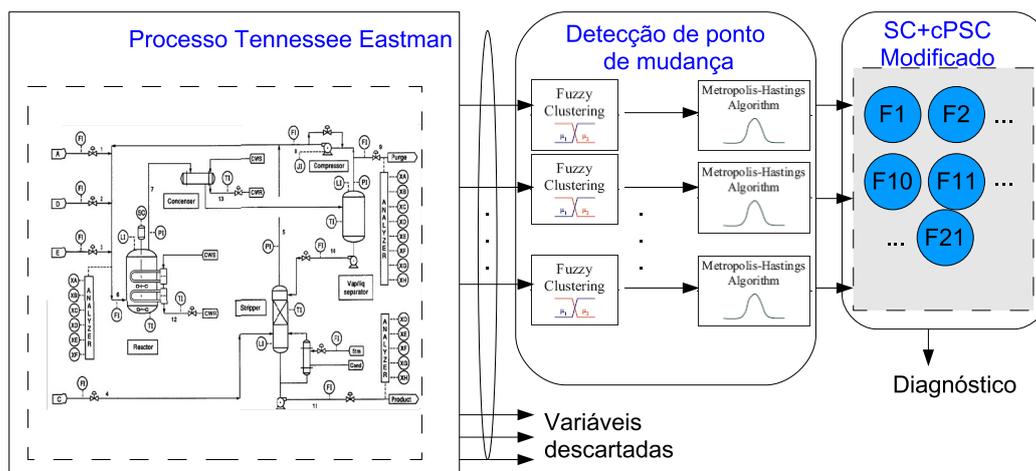


Figura 2: Esquema de Detecção e Classificação de Falhas no Tennessee Eastman

[Jing e Hou, 2015].

No modelo proposto nesse trabalho, a etapa de detecção de falhas utilizou um modelo baseado em Análise Qualitativa de Tendências (QTA) proposto em [D'Angelo et al., 2011b]. Tal modelo, faz uso de duas etapas para detectar falhas. A primeira é baseada na teoria dos conjuntos fuzzy e a segunda em uma abordagem bayesiana, para detecção de ponto de mudança em séries temporais, isto é, a detecção de ponto de mudança dará suporte a hipótese de ocorrência de falha. Caso exista tal hipótese, o classificador apresentado na subseção 2.3 será responsável por dizer qual falha aconteceu. O esquema geral do modelo de detecção e classificação de falhas no Tennessee Eastman é mostrado na Figura 2. Pela Figura, ainda pode-se notar que algumas variáveis do processo foram descartadas. Isso aconteceu por um processo de redução da dimensionalidade que aconteceu em 2 momentos. No primeiro momento, foi utilizada uma matriz de correlação obtida a partir das 52 variáveis do processo. As variáveis com correlação maior que 0,6 foram descartadas. E ao final do primeiro momento as variáveis 15 variáveis foram descartadas, restando apenas 37 variáveis. No segundo momento, 7 variáveis foram descartadas por não apresentarem qualquer indicação de falha pelo abordagem Fuzzy/Bayesiana, restando apenas 30 variáveis que foram utilizadas para treinar e testar o classificador.

Em [Jing e Hou, 2015] as técnicas *support vector machine* (SVM) e *principal component analysis* (PCA) são aplicadas no problema de classificação de falhas no *Tennessee Eastman* e os resultados da taxa de acerto por classe são apresentados na tabela 2. No total a abordagem proposta obteve uma taxa de acerto de 89,45% e encontrou 69 partículas.

Pelos resultados, obtidos pela abordagem proposta, quando comparados com os resultados obtidos por PCA (Tabela 2), percebe-se que, das 21 classes, em apenas duas delas a abordagem proposta obteve um resultado pior e nas demais percebe-se uma ampla vantagem do modelo proposto. Já, quando comparados com SVM em nenhuma das classes a abordagem proposta obteve resultados piores. Ainda é importante destacar que o modelo proposto obteve uma taxa de acerto geral (com 21 classes) de 89,45%, enquanto que o PCA e o SVM obtiveram 67,21% e 41,64%, respectivamente.

4. Considerações Finais

Este trabalho teve como objetivo principal propor uma abordagem para detecção e classificação de falhas em sistemas dinâmicos de forma adaptativa, ou seja, a metodologia proposta pode criar mais de um protótipo para representar uma mesma classe durante o processo de treinamento na tentativa de aumentar a taxa de acertos final do classificador. Porém, a quantidade de protótipos cri-



Tabela 2: Comparação das taxas de acertos (em %) do Modelo Apresentado com PCA e SVM

Desc	cPSC_M	PCA	SVM	Desc	cPSC_M	PCA	SVM
Falha 1	100	87.19	87.19	Falha 2	100	87.5	85.83
Falha 3	77.5	18.33	15.21	Falha 4	67	72.71	49.48
Falha 5	100	4.06	57.60	Falha 6	97.5	90.21	78.85
Falha 7	96.5	89.69	88.85	Falha 8	100	85	32.19
Falha 9	25.79	20.21	12.81	Falha 10	99.5	76.15	22.60
Falha 11	88	65.42	11.88	Falha 12	99.5	85.83	50.21
Falha 13	100	69.06	21.46	Falha 14	58.5	86.56	54.90
Falha 15	100	23.02	18.85	Falha 16	98	69.48	12.81
Falha 17	95	74.48	48.02	Falha 18	99.5	59.90	32.19
Falha 19	99.49	84.06	46.25	Falha 20	78.39	77.50	38.96
Falha 21	95	85	8.23				

ados não deve ser significativamente grande devido ao custo computacional. A abordagem proposta busca maximizar a taxa de acertos e minimizar o número de protótipos gerados.

Para isso, inicialmente este trabalho propôs três alterações no método de agrupamento cPSC com o objetivo de inserir características supervisionadas ao mesmo, visto que tal algoritmo não leva em consideração o rótulo dos dados na formação dos grupos por ser um algoritmo de aprendizagem não supervisionada. Assim, a contribuição desse trabalho foi propor uma metodologia capaz de classificar falhas de forma adaptativa, tendo em vista que nem sempre um único método de detecção e/ou classificação de falhas consegue obter bons resultados para um determinado problema. Dessa forma, a metodologia apresentada é mais um opção para tratar problemas de FDI. Analisando os resultados da metodologia apresentada na classificação de falhas quando aplicada no *Tennessee Eastman Process*, percebe-se que a mesma é promissora para detecção e classificação de falhas na classe de sistemas dinâmicos estudada.

Futuramente pretende-se aplicar e avaliar o comportamento e a eficiência dessa metodologia em outros problemas de FDI nos quais as classes estejam bastante sobrepostas.

Referências

- Baccarini, L. M. R., de Menezes, B. R., e Caminhas, W. M. (2010). Fault induction dynamic model, suitable for computer simulation: Simulation results and experimental validation. *Mechanical Systems and Signal Processing*, 24(1):300–311.
- Chiu, S. (1996). Method and software for extracting fuzzy classification rules by subtractive clustering. In *Fuzzy Information Processing Society, 1996. NAFIPS., 1996 Biennial Conference of the North American*, p. 461–465. IEEE.
- Cohen, S. e de Castro, L. N. (2006). Data clustering with particle swarms. In *Evolutionary Computation, 2006. CEC 2006. IEEE Congress on*, p. 1792–1798. IEEE.
- D’Angelo, M. F., Palhares, R. M., Maia, R. D., Mendes, J. B., Ekel, P. Y., Cangussu, C. K., e Aguiar, L. A. (2016). Monitoring the stator current in induction machines for possible fault detection: A fuzzy/bayesian approach for the problem of time series multiple change point detection. *Pesquisa Operacional*, 36(2):301–320.
- D’Angelo, M. F., Palhares, R. M., Takahashi, R. H., e Loschi, R. H. (2011a). Fuzzy/bayesian change point detection approach to incipient fault detection. *IET control theory & applications*, 5(4):539–551.



- D'Angelo, M. F., Palhares, R. M., Takahashi, R. H., Loschi, R. H., Baccarini, L. M., e Caminhas, W. M. (2011b). Incipient fault detection in induction machine stator-winding using a fuzzy-bayesian change point detection approach. *Applied Soft Computing*, 11(1):179–192.
- D'Angelo, M. F. e Costa Jr, P. P. (2001). Detection of shorted turns in the field winding of turbogenerators using the neural network mlp. In *Systems, Man, and Cybernetics, 2001 IEEE International Conference on*, volume 3, p. 1930–1935. IEEE.
- de Castro, L. N., Zuben, F. J. V., e de Deus Jr, G. (2003). The construction of a boolean competitive neural network using ideas from immunology. *Neurocomputing*, 50:51–85.
- Downs, J. J. e Vogel, E. F. (1993). A plant-wide industrial process control problem. *Computers & chemical engineering*, 17(3):245–255.
- D'Angelo, M. F., Palhares, R. M., Camargos Filho, M. C., Maia, R. D., Mendes, J. B., e Ekel, P. Y. (2016). A new fault classification approach applied to tennessee eastman benchmark process. *Applied Soft Computing*, 49:676–686.
- D'Angelo, M. F., Palhares, R. M., Cosme, L. B., Aguiar, L. A., Fonseca, F. S., e Caminhas, W. M. (2014). Fault detection in dynamic systems by a fuzzy/bayesian network formulation. *Applied Soft Computing*, 21:647–653.
- Fábrega, A. J. S., Caro, J. M. B., Herrera, P. J. A., e Gasca, R. M. (2014). Data-driven bounded-error fault detection. *International Journal of Adaptive Control and Signal Processing*, 28(12): 1299–1324.
- Jiang, Q., Yan, X., e Zhao, W. (2013). Fault detection and diagnosis in chemical processes using sensitive principal component analysis. *Industrial & Engineering Chemistry Research*, 52(4): 1635–1644.
- Jing, C. e Hou, J. (2015). Svm and pca based fault classification approaches for complicated industrial process. *Neurocomputing*.
- Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Journal of Fluids Engineering*, 82(1):35–45.
- Kohonen, T. (1990). The self-organizing map. *Proceedings of the IEEE*, 78(9):1464–1480.
- Lemos, A. P. (2007). *Proposta de um algoritmo genérico de detecção de novidades em séries temporais utilizando modelos de previsão*. PhD thesis, Master's thesis, Universidade Federal de Minas Gerais, Belo Horizonte, Brasil.
- Li, X., Liu, H. H. T., e Jiang, B. (2015). Parametrization of optimal fault detection filters. *Automatica*, 56:70–77.
- Maurya, M. R., Rengaswamy, R., e Venkatasubramanian, V. (2007). Fault diagnosis using dynamic trend analysis: A review and recent developments. *Engineering applications of artificial intelligence*, 20(2):133–146.
- Naresh, R., Sharma, V., e Vashisth, M. (2008). An integrated neural fuzzy approach for fault diagnosis of transformers. *Power Delivery, IEEE Transactions on*, 23(4):2017–2024.
- Prior, A. K. F. e de Castro, L. N. (2010). cpsc: Um algoritmo de enxame construtivo para agrupamento de dados. In *Anais do XVIII Congresso Brasileiro de Automática*.



- Santos, L. I., D'Angelo, M. F., Costa, L. M., e Palhares, R. M. (2015). Uma abordagem adaptativa para classificação de falhas em sistemas dinâmicos. In *Anais do XVII Simpósio Brasileiro de Automação Inteligente*, Natal-RN.
- Simani, S., Fantuzzi, C., e Patton, R. J. (2003). *Model-Based Fault Diagnosis In Dynamic Systems Using Identification Techniques*. Springer.
- Takahashi, R. e Peres, P. (1999). Unknown input observers for uncertain systems: a unifying approach. *European Journal of Control*, 5(2):261–275.
- Venkatasubramanian, V., Rengaswamy, R., e Kavuri, S. N. (2003a). A review of process fault detection and diagnosis – part II: Qualitative models and search strategies. *Computers and Chemical Engineering*, 27(3):313–326.
- Venkatasubramanian, V., Rengaswamy, R., Kavuri, S. N., e Yin, K. (2003b). A review of process fault detection and diagnosis – part III: Process history based methods. *Computers and Chemical Engineering*, 27(3):327–346.
- Venkatasubramanian, V., Rengaswamy, R., Yin, K., e Kavuri, S. N. (2003c). A review of process fault detection and diagnosis – part I: Quantitative model-based methods. *Computers and Chemical Engineering*, 27(3):293–311.