

UNIVERSIDADE FEDERAL DO PARÁ
INSTITUTO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

**PROPOSTA DE UM FRAMEWORK PARA IDENTIFICAÇÃO DE SISTEMAS
DINÂMICOS MULTIVARIÁVEIS NÃO LINEARES**

EWERTON CRISTHIAN LIMA DE OLIVEIRA

DM 10/2020

UFPA/ITEC/PPGEE
Campus Universitário do Guamá
Belém – Pará – Brasil

2020

UNIVERSIDADE FEDERAL DO PARÁ
INSTITUTO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

**PROPOSTA DE UM FRAMEWORK PARA IDENTIFICAÇÃO DE SISTEMAS
DINÂMICOS MULTIVARIÁVEIS NÃO LINEARES**

EWERTON CRISTHIAN LIMA DE OLIVEIRA

DM 10/2020

UFPA/ITEC/PPGEE
Campus Universitário do Guamá
Belém – Pará – Brasil

2020

UNIVERSIDADE FEDERAL DO PARÁ
INSTITUTO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

EWERTON CRISTHIAN LIMA DE OLIVEIRA

**PROPOSTA DE UM FRAMEWORK PARA IDENTIFICAÇÃO DE SISTEMAS
DINÂMICOS MULTIVARIÁVEIS NÃO LINEARES**

Dissertação submetida à Banca
Examinadora do Programa de Pós-
Graduação em Engenharia Elétrica
da UFPA para a obtenção do Grau
de Mestre em Engenharia Elétrica
na Área de Computação Aplicada.

UFPA/ITEC/PPGEE
Campus Universitário do Guamá
Belém – Pará – Brasil

2020

Dados Internacionais de Catalogação na Publicação (CIP) de acordo com ISBD
Sistema de Bibliotecas da Universidade Federal do Pará
Gerada automaticamente pelo módulo Ficat, mediante os dados fornecidos pelo(a) autor(a)

O48p Oliveira, Ewerton Cristhian Lima de
Proposta de um framework para identificação de sistemas
dinâmicos multivariáveis não lineares / Ewerton Cristhian Lima de
Oliveira. — 2020.
85 f. : il. color.

Orientador(a): Prof^ª. Dra. Jasmine Priscyla Leite de Araujo
Dissertação (Mestrado) - Programa de Pós-Graduação em
Engenharia Elétrica, Instituto de Tecnologia, Universidade Federal
do Pará, Belém, 2020.

1. Identificação. 2. Sistemas Multivariáveis. 3. Framework.
4. PSO. 5. Mínimos Quadrados. I. Título.

CDD 629.8



UNIVERSIDADE FEDERAL DO PARÁ
INSTITUTO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

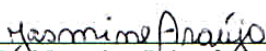
**“PROPOSTA DE UM FRAMEWORK PARA IDENTIFICAÇÃO DE
SISTEMAS DINÂMICOS MULTIVARIÁVEIS NÃO LINEARES”**

AUTOR: EWERTON CRISTHIAN LIMA DE OLIVEIRA

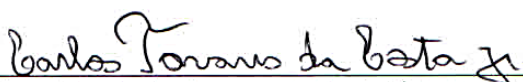
DISSERTAÇÃO DE MESTRADO SUBMETIDA À BANCA EXAMINADORA APROVADA PELO COLEGIADO DO PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA, SENDO JULGADA ADEQUADA PARA A OBTENÇÃO DO GRAU DE MESTRE EM ENGENHARIA ELÉTRICA NA ÁREA DE COMPUTAÇÃO APLICADA.

APROVADA EM: 27/02/2020

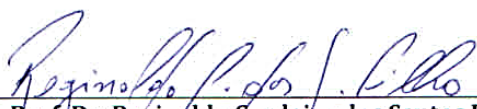
BANCA EXAMINADORA:



Prof.ª Dr.ª Jasmine Priscyla Leite de Araújo
(Orientadora – PPGEE/UFPA)



Prof. Dr. Carlos Tavares da Costa Júnior
(Avaliador Interno – PPGEE/UFPA)



Prof. Dr. Reginaldo Cordeiro dos Santos Filho
(Avaliador Externo ao Programa – ICEN/UFPA)

VISTO:

Prof.ª Dr.ª Maria Emília de Lima Tostes
(Coordenadora do PPGEE/ITEC/UFPA)

*“Maybe you could talk freely
Speak to me for once so truly
For once so truly
I could know
Just where you're going
Just where I'm standing
That would really be something”.*

(Loving)

DEDICATÓRIA

Dedico este trabalho a minha bisavó por ter sido um exemplo grandioso de como se deve viver a vida. Apesar dos poucos momentos que passei ao seu lado nos últimos anos, nunca esquecerei o amor, o carinho, a admiração e o grande orgulho que a Amelia tinha pelo seu primeiro bisneto. Infelizmente ela não estará aqui para ver eu concluir esse mestrado, mas o seu sorriso e seu amor sempre ficarão guardados em minha memória.

Obrigado bisavó, pois essa vitória também é sua!

AGRADECIMENTOS

Agradeço em primeiro lugar a Deus pela minha vida, pelo conhecimento e pelas oportunidades que ele me ofereceu durante toda a minha jornada.

Agradeço a toda minha família, em especial aos meus pais Josilea e Eliezer, a minha avó Marilea e aos meus irmãos Jadson e Sarah por sempre me ajudarem e por sempre acreditarem em mim, e por toda a ajuda que me deram para que esse sonho tenha se realizado, além de estarem comigo quando eu mais precisei.

Agradeço a minha orientadora, professora Dr^a Jasmine Araujo, por todo o conhecimento e orientações passadas a mim, as quais foram essenciais para o desenvolvimento deste trabalho. Agradeço grandemente pela oportunidade e confiança dadas a mim para o ingresso na pós-graduação e para o desenvolvimento desta dissertação. Agradeço toda a atenção que foi me dada durante toda essa trajetória.

Agradeço a todos os meus amigos da pós-graduação, da graduação e do Laboratório de Inteligência Computacional e Pesquisa Operacional (LINC) pelo acolhimento, pela amizade e ajuda em muitos momentos. Em especial aos amigos Lucas Vinicius, Sandio Maciel, Igor Falcão, Francisco Junior, Geandreson Costa, Arthur Cordovil, Igor Araujo, André Pereira, Tarcisio Pinheiro, Bruno Costa, Paulo Vitor, Luana Gonçalves, Jaqueline Oliveira, além de tantos outros.

Agradeço a todos os meus amigos, em especial ao Juan Ferreira Vidal, por ter se tornado um grande amigo e parceiro de pesquisa, e por me fornecer diversos auxílios, dicas, correções de trabalhos e ajudas que foram e são de grande importância para a minha formação acadêmica.

Agradeço aos professores Orlando Fonseca Silva e Manoel da Silva Filho por toda ajuda, apoio e credibilidade que me deram durante minha jornada acadêmica, e também por serem exemplos e inspirações para minha formação como pesquisador.

Agradeço ao professor Alex Castelo Branco Martins, a quem serei eternamente grato por ter me oferecido a oportunidade de ter um preparo de qualidade para ingressar nesta Universidade, além de ter sido um grande amigo nos momentos de dificuldade.

Agradeço a todos que de maneira direta ou indireta me ajudaram a conseguir concluir esta dissertação e contribuíram para que eu pudesse concluir mais esta etapa. Mesmo não podendo citar o nome de todos, quero deixar meu mais sincero obrigado.

Agradeço ao Programa de Pós-Graduação em Engenharia Elétrica (PPGEE) da UFPA pela oportunidade de aprendizagem e crescimento que foram dadas a mim. Agradeço a CAPES pelo apoio financeiro, que possibilitou que eu pudesse me dedicar para realizar mais este objetivo.

RESUMO

As técnicas de identificação de sistemas dinâmicos são algoritmos de extrema importância para a geração de modelos matemáticos e computacionais capazes de representar a dinâmica de sistemas e processos presentes em diversos âmbitos da sociedade, como: processos industriais; automóveis; produção de alimentos; veículos aeroespaciais; sistemas biológicos e etc. Identificar esses sistemas, que em geral possuem mais de uma variável de entrada e saída (sistemas multivariáveis) e também são não lineares, é de grande importância para a ciência e para a engenharia no que tange ao desenvolvimento de novas técnicas de controle, monitoramento de falhas e previsão de estados de operação desses mecanismos. Todavia, identificar sistemas MIMO (do inglês, *Multiple Input Multiple Output*) não lineares é uma tarefa complicada, tanto devido à dificuldade de se implementar os algoritmos clássicos para a resolução deste problema, quanto ao fato de que sistemas não lineares requerem modelos complexos para a representação de sua dinâmica de maneira satisfatória. Visando contribuir com a solução deste problema, este trabalho propõe um *framework* capaz de realizar tanto a identificação de sistemas dinâmicos MIMO não lineares no modelo *fuzzy* TSK multivariável, que representa de maneira simples o acoplamento das variáveis envolvidas na identificação, quanto a seleção do vetor regressor usado no modelo. Para a realização da parametrização do modelo *fuzzy* TSK multivariável, o *framework* proposto utiliza os algoritmos Mínimos Quadrados (MQ) e Otimização por Exame de Partículas (PSO do inglês, *Particle Swarm Optimization*), os quais são responsáveis por estimar as matrizes de parâmetros e o conjunto de desvio padrões das *Gaussianas* das entradas do modelo, respectivamente. A metodologia proposta é testada e comparado com uma RNA e o modelo de Hammerstein-Wiener (HW) na identificação de duas plantas industriais MIMO não lineares: Reator Contínuo de Tanque Agitado (CSTR); Secador Industrial. A comparação das três técnicas é feita com base nos índices de Erro Quadrático Médio (EQM) e *Variance Accounted For* (VAF), além da análise de resíduos entre os dados observados e estimados. Os resultados mostraram que o *framework* proposto obteve o melhor desempenho em 80% das estimações de saídas das duas plantas multivariadas com base nos dois índices, e também alcançou o melhor desempenho em 60% dos casos na análise residual da identificação das plantas.

PALAVRAS-CHAVES: Identificação, Não Linear, Sistemas Multivariáveis, *Framework*, *Fuzzy* TSK, PSO, Mínimos Quadrados.

ABSTRACT

The techniques of dynamic systems identification are algorithms of most importance for generating mathematical and computational models capable to represent the dynamic of systems and processes present in many fields of society, such as: industrial processes; automobiles; food production; aerospace vehicles; biological systems and etc. The identification of these systems, which generally have more than one variable of input and output (multivariable systems) and also are nonlinear, it is very important for science and engineering in relation to the development of new control techniques, fault monitoring and prediction of operating state of these mechanisms. Nonetheless, the identification of nonlinear MIMO (*Multiple Input Multiple Output*) systems is a hard task, as much due the difficulty of implementing the classic algorithms for solve this problem, as the fact that nonlinear systems require complex models for represent their dynamics in satisfactory way. In order to contribute with the solution of this problem, this work proposes a framework capable of performing as much the identification of nonlinear dynamic MIMO systems in multivariable fuzzy TSK model, which can represent in simple way the coupling among the variables involved in identification, as the selection of regressor vector used in model. To perform *fuzzy* TSK multivariable model parameterization, the proposed framework uses the algorithms Least Square (LS) and Particle Swarm Optimization (PSO), which are responsible to estimate the matrix of parameters and the set of standard deviation of the *Gaussians* in model inputs, respectively. The proposed methodology is tested and compared with RNA and a Hammerstein-Wiener (WH) model in identification of two nonlinear MIMO industrial plants: Continuous Stirred Tank Reactor (CSTR); Industrial Dryer. The comparison of these three techniques is made with base in indices of Mean Squared Error (*MSE*) and Variance Accounted For (*VAF*), further the analysis of residues between the observed and estimated data. The results show that the proposed framework got the best performance, based in the two indices, in 80% of outputs estimation of the two multivariable plants, and also reached the best performance in 60% of residual analysis of plants identification.

KEYWORDS: Identification, Nonlinear, Multivariable Systems, Framework, Fuzzy TSK, PSO, Least Squares.

SUMÁRIO

1	INTRODUÇÃO	1
1.1	JUSTIFICATIVA.....	2
1.2	REVISÃO BIBLIOGRÁFICA.....	2
1.3	OBJETIVOS.....	4
1.4	ESTRUTURA DO TRABALHO.....	4
2	IDENTIFICAÇÃO DE SISTEMAS MULTIVARIÁVEIS NÃO LINEARES	6
2.1	INTRODUÇÃO	6
2.2	TÉCNICAS CLÁSSICAS DE IDENTIFICAÇÃO DE SISTEMAS MIMO NÃO LINEARES	8
2.2.1	<i>Modelo de Hammerstein-Wiener</i>	9
2.2.2	<i>Redes Neurais Artificiais</i>	11
2.3	CONCLUSÃO	13
3	FRAMEWORK PARA IDENTIFICAÇÃO DE SISTEMAS MIMO NÃO LINEARES.....	14
3.1	ESTRUTURA GERAL DO <i>FRAMEWORK</i>	14
3.2	ALGORITMOS DA ETAPA DE IDENTIFICAÇÃO	17
3.2.1	<i>Sistema fuzzy TSK</i>	19
3.2.2	<i>Mínimos Quadrados</i>	25
3.2.3	<i>Particle Swarm Optimization (PSO)</i>	26
3.3	CONCLUSÃO	32
4	RESULTADOS	33
4.1	REATOR CONTÍNUO DE TANQUE AGITADO (<i>CONTINUOUS STIRRED TANK REACTOR</i>) ...	34
4.2	SECADOR INDUSTRIAL (<i>INDUSTRIAL DRYER</i>)	44
4.3	ANÁLISE GERAL DOS RESULTADOS	56
4.4	CONCLUSÃO	58
5	CONSIDERAÇÕES FINAIS.....	59
5.1	CONCLUSÃO	59
5.2	TRABALHOS FUTUROS	60
	REFERÊNCIAS BIBLIOGRÁFICAS	61
	APÊNDICE I.....	65
	APÊNDICE II	67

LISTA DE SÍMBOLOS

$u_m(k)$	m -ésima entrada MIMO
$y_n(k)$	n -ésima saída MIMO
m	Número de entradas no sistema MIMO
n	Número de saídas no sistema MIMO
k	Tempo discreto
$v_1(k)$	Entrada do bloco HW
$v_2(k)$	Saída do bloco HW
l	Termo final da função de transferência HW
N_1, N_2	Blocos do Hammerstein-Wiener SISO
$N_{n,m}, N_{2,n}$	Blocos do Hammerstein-Wiener MIMO
$G_s(q)$	Função de transferência do HW SISO
$G_{n,m}(q)$	Função de transferência do HW MIMO
i	Variável auxiliar da entrada no sistema
j	Variável auxiliar da saída no sistema
u_i	Entrada do neurônio
w_i	Peso sináptico da RNA
$\bar{b}$	Bias da RNA
τ_d	Atraso
$U_i(k - \tau_d)$	Entrada da RNA
$Y_j(k)$	Saída da RNA
N_s, N_{total}	Número de amostras total dos dados.
N_{id}, N_v, N_t	Dados de identificação, validação e teste
N_{sim}	Número de simulações da etapa de identificação
I_{mod}	Índice do contador de modelo
I_{total}	Total de modelos
$\phi(y_n, u_m, \tau_{du}, \tau_{dy})$	Vetor regressor
τ_{du}	Atrasos na entrada
τ_{dy}	Atrasos na saída
$\hat{\theta}$	Parâmetros das equações ARX
$\mu_m^{\rho_{\{1 \dots l_m\}}}$	Função de pertinência <i>fuzzy</i>
l_m	Total de conjuntos <i>fuzzy</i> por entrada
R	R -ésima regra <i>fuzzy</i>
S_R	Número total de regras <i>fuzzy</i>

d^R	Termo da função <i>fuzzy</i>
φ_1	Índices da matriz de parâmetros
$\bar{c}_{(1,2)m}$	Centro da gaussiana do <i>fuzzy</i>
$\sigma_{(1,2)m}$	Desvio padrão da gaussiana do <i>fuzzy</i>
$\vec{\sigma}_G$	Vetor de desvio padrão estimado pelo PSO
H^R	Matriz de parâmetros
$\xi_R(k)$	Fator de ponderação <i>fuzzy</i>
$\bar{Y}$	Matriz de medidas observadas da saída da planta
$\bar{X}$	Matriz de medidas com regressor
S_m	Número total de entradas da planta MIMO
S_n	Número total de saídas da planta MIMO
N_{itotal}	Número total de iterações do PSO
t	Iteração do PSO
q	Partícula do PSO
N_{part}	Número total de partículas
N_{var}	Dimensão do vetor de partículas do PSO
$\vec{x}_q(t)$	Posição das partículas do PSO
$\vec{v}_q(t)$	Velocidade das partículas do PSO
$\vec{P}_q(t)$	Memória cognitiva das partículas do PSO
$\vec{G}(t)$	Melhor partícula global do PSO
f_{obj}	Função objetivo do PSO
$\bar{w}$	Fator de inércia do PSO
ψ	Índice da melhor partícula do PSO
$r_1(t)$ e $r_2(t)$	Parâmetros aleatórios do PSO
$\hat{y}_n(k)$	Saída estimada da n -ésima saída planta MIMO
$\vec{U}(k)$	Vetor de entradas da planta MIMO
$\vec{Y}(k)$	Vetor de saídas da planta MIMO
$\vec{e}(k)$	Vetor de erro da planta MIMO
$\vec{q}_{NaOH}$	Vazão de hidróxido de sódio
$\vec{q}_{HAC}$	Vazão de Ácido Acético
$\vec{q}_{saida}$	Vazão de saída do CSTR

LISTA DE FIGURAS

Figura 2.1 - Sistema de quatro tanques acoplados.....	6
Figura 2.2 - Diagrama esquemático de um sistema multivariável.	7
Figura 2.3 - Diagrama de identificação de sistema multivariável.	7
Figura 2.4 - Modelo SISO de Hammerstein-Wiener.....	9
Figura 2.5 – Modelo MIMO de Hammerstein-Wiener.....	10
Figura 2.6 – Estrutura de uma MLP para sistema MIMO.	12
Figura 3.1 – Fluxograma do <i>Framework</i> proposto para identificar sistemas MIMO não lineares.....	14
Figura 3.2 – Etapa de identificação do <i>framework</i>	18
Figura 3.3 – Diagrama do sistema <i>fuzzy</i> TSK.	20
Figura 3.4 – Exemplo de conjuntos <i>fuzzy</i>	20
Figura 3.5 – Entrada do sistema <i>fuzzy</i> TSK com funções de pertinência <i>Gaussiana</i>	23
Figura 3.6 – Fluxograma do algoritmo PSO.....	27
Figura 4.1 – Diagrama do CSTR.....	35
Figura 4.2 – Dados de entrada das variáveis $q_{HAC}(k)$ e $q_{NaOH}(k)$	36
Figura 4.3 – Dados de saída das variáveis $pH(k)$ e $q_{saída}(k)$	36
Figura 4.4 – Saída y_1 (pH) do CSTR observada (preto), e estimada pelo HW (vermelho), pela RNA (verde) e pelo <i>framework</i> (azul).	40
Figura 4.5 – Saída y_2 ($q_{saída}$) do CSTR observada (preto), e estimada pelo HW (vermelho), pela RNA (verde) e pelo <i>framework</i> (azul).	40
Figura 4.6 – Erro da saída y_1 do CSTR com a saída estimada pelo HW (vermelho), pela RNA (verde) e pelo <i>framework</i> (azul).	42
Figura 4.7 – Erro da saída y_2 do CSTR com a saída estimada pelo HW (vermelho), pela RNA (verde) e pelo <i>framework</i> (azul).	42
Figura 4.8 – Diagrama de um secador industrial do tipo <i>Spray Dryer</i>	45
Figura 4.9 – Dados de entrada das variáveis $u_1(k)$, $u_2(k)$ e $u_3(k)$	46
Figura 4.10 – Dados de saída das variáveis $y_1(k)$, $y_2(k)$ e $y_3(k)$	46
Figura 4.11 – Saída y_1 (Temperatura do bulbo úmido) do tanque observada (preto), e estimada pelo HW (vermelho), pela RNA (verde) e pelo <i>framework</i> (azul).....	50
Figura 4.12 - Saída y_2 (Temperatura do bulbo seco) do tanque observada (preto), e estimada pelo HW (vermelho), pela RNA (verde) e pelo <i>framework</i> (azul).....	50
Figura 4.13 - Saída y_3 (Teor de umidade da matéria-prima) do tanque observada (preto), e estimada pelo HW (vermelho), pela RNA (verde) e pelo <i>framework</i> (azul).....	51
Figura 4.14 – Sinal de erro da saída y_1 do secador industrial com a saída estimada pelo HW (vermelho), pela RNA (verde) e pelo <i>framework</i> (azul).	53
Figura 4.15 – Sinal de erro da saída y_2 do secador industrial com a saída estimada pelo HW (vermelho), pela RNA (verde) e pelo <i>framework</i> (azul).	53
Figura 4.16 – Sinal de erro da saída y_3 do secador industrial com a saída estimada pelo HW (vermelho), pela RNA (verde) e pelo <i>framework</i> (azul).	54

LISTA DE TABELAS

Tabela 4.1 – Divisão percentual e temporal dos dados para o 1º estudo de caso.....	37
Tabela 4.2 – Parâmetros do PSO do <i>framework</i> no 1º estudo de caso.	37
Tabela 4.3 – Modelos de regressores do <i>framework</i> no 1º estudo de caso.....	37
Tabela 4.4 – Regras do modelo <i>fuzzy</i> TSK no 1º estudo de caso.	38
Tabela 4.5 – Parâmetros da RNA no 1º estudo de caso.	38
Tabela 4.6 – Parâmetros do modelo Hammerstein-Wiener no 1º estudo de caso.	38
Tabela 4.7 – Resultados do EQM_v e do $mEQM_v$ para os 3 modelos de regressores utilizados no <i>framework</i> na etapa de identificação no 1º estudo de caso.	39
Tabela 4.8 – Resultados do EQM_t e do VAF para os dados de teste no 1º estudo de caso.	41
Tabela 4.9 – Média dos desvios padrões das <i>Gaussianas</i> alcançados pelo PSO no 1º estudo de caso.	44
Tabela 4.10 – Divisão percentual e temporal dos dados para o 2º estudo de caso.....	47
Tabela 4.11 – Parâmetros do PSO do <i>framework</i> no 2º estudo de caso.	47
Tabela 4.12 – Modelos de regressores do <i>framework</i> no 2º estudo de caso.....	47
Tabela 4.13 – Regras do modelo <i>fuzzy</i> TSK no 2º estudo de caso.	48
Tabela 4.14 – Parâmetros do modelo Hammerstein-Wiener no 2º estudo de caso.	48
Tabela 4.15 – Parâmetros da RNA no 2º estudo de caso.	49
Tabela 4.16 – Resultados do EQM_v e do $mEQM_v$ para os 3 modelos de regressores utilizados no <i>framework</i> na etapa de identificação no 2º estudo de caso.	49
Tabela 4.17 – Resultados do EQM_t e do VAF para os dados de teste no 2º estudo de caso.	51
Tabela 4.18 – Média dos desvios padrões das <i>Gaussianas</i> alcançados pelo PSO no 2º estudo de caso.....	54
Tabela 4.19 – Sumário das técnicas com melhor desempenho em EQM_t e VAF por saída de cada planta.	56
Tabela 4.20 – Sumário da Média e Variância dos sinais de resíduo de cada técnica para cada saída.....	57

1 INTRODUÇÃO

Com o avanço da ciência e da tecnologia, processos e mecanismos cada vez mais complexos são criados com o intuito de realizar tarefas essenciais e que beneficiam as pessoas em diversos ramos do espectro produtivo da sociedade, como por exemplo: produção de alimentos; fabricação de polímeros; processos biotecnológicos; secagem de matéria-prima (Zhu; Backx, 1993); sistemas elétricos; sistemas mecânicos e hidráulicos (Coelho; Coelho, 2004); processos industriais (Cham et al., 2017).

O estudo da modelagem matemática destes processos é um assunto de grande relevância que vem ganhando espaço no meio científico, à medida que se deseja controlar, analisar e prever falhas nestes mecanismos (Aguirre, 2014). Todavia, uma das maiores dificuldades na modelagem matemática desta classe processos do mundo real recai não apenas na manipulação das equações físicas que regem seus modelos, mas também por serem não lineares (Paduart, et al., 2006).

A identificação de sistemas é uma alternativa viável a modelagem matemática, e que vem sendo utilizada e aprimorada nas últimas décadas com o intuito de representar sistemas por modelos que se aproximam da dinâmica do caso real, o qual possui diversas aplicações práticas, principalmente na indústria (Boutalis et al., 2017; Chiuso; Pillonetto, 2019).

Os sistemas multivariáveis, os quais representam a maioria dos processos industriais e algumas outras aplicações do mundo real, podem ser classificados em função do seu número de entradas, como: sistema com várias entradas e várias saídas ou MIMO; sistema com uma entrada e várias saídas ou SIMO (do inglês, *Single Input Multiple Output*); sistema com várias entradas e uma saída MISO (do inglês, *Multiple Input Single Output*) (Aguirre, 2014).

A identificação de sistemas multivariáveis e não lineares vem sendo estudada ao longo dos anos. Esta classe de sistemas requer algoritmos capazes de estimar modelos que possam representar a dinâmica não linear do processo, que por sua vez possam ser utilizados no projeto de controladores (Jafari et al., 2014).

Atualmente, existem algumas metodologias de identificação de sistemas não lineares que podem ser aplicadas a sistemas multivariáveis, como por exemplo a Série de Volterra, Redes Neurais Artificiais (RNA), modelos NARMAX (do inglês, *Nonlinear Autoregressive Moving Average with Exogenous Input*), modelos Hammerstein-Wiener (Zhu, 2001), modelos *Neuro-Fuzzy* (Nelles, 2001).

Entretanto, estruturas como RNA e *Neuro-Fuzzy* não fornecem equações de fácil manipulação e visualização, mas sim, uma estrutura fechada e interconectada através de funções e coeficientes de ponderação. O modelo de Hammerstein-Wiener, embora mais simples que os dois primeiros, necessita de adaptações para serem aplicadas em sistemas multivariáveis (Thomas; Allgower, 2010), aumentando assim o seu nível de complexidade de implementação e estimação paramétrica.

Outra característica importante é que as técnicas supracitadas, em sua forma canônica, não preveem a seleção de modelo, o que demanda o ajuste por métodos como Taxa de Redução de Erro, Critério de Informação de Akaike, etc. Todavia, para sistemas não lineares, tais métodos podem não ser tão eficientes (Aguirre, 2014).

Portanto, este trabalho propõem um *framework* (estrutura) que realize tanto a identificação de sistemas MIMO não lineares no modelo *fuzzy* TSK multivariável, utilizando os Mínimos Quadrados (MQ) e Otimização por Enxame de Partículas (PSO), quanto a seleção do vetor regressor utilizado no modelo estimado. Este *framework* será aplicado em dois estudos de casos reais de identificação, e os resultados são comparados com modelos de Hammerstein-Wiener (HW) e RNA.

1.1 Justificativa

Em virtude da necessidade de identificar sistemas multivariáveis não lineares utilizando um algoritmo que realize tanto a identificação paramétrica quanto a seleção de modelo para o qual a planta vai ser aproximada, cujo modelo identificado seja de fácil implementação computacional, manipulação e visualização, este trabalho aborda, a proposta de um *framework* capaz de realizar a identificação *off-line* de sistemas MIMO não lineares em modelo *fuzzy* TSK multivariável, além de realizar a seleção de modelo a ser identificado, visando a diminuição do erro de estimação do sistema.

1.2 Revisão bibliográfica

Alguns trabalhos sobre aplicação de sistemas *fuzzy* TSK e metaheurísticas na identificação de sistemas multivariáveis não lineares estão disponíveis na literatura.

Em Sanandaji, Salahshoor & Fatehi (2007) é proposto uma metodologia de identificação de sistemas MIMO não lineares no modelo *fuzzy* TSK, onde o Algoritmo Genético (AG) é usado para ajustar os parâmetros da função de pertinência. Esta metodologia é utilizada na identificação de uma coluna de destilação MIMO não linear,

e os resultados mostram que a proposta com AG possui maior acurácia que o modelo *fuzzy* TSK convencional, o qual foi ajustado por tentativa e erro.

Identificação de sistemas MIMO também é abordada em Bouzaida, Sakly, & M'Sahli (2012), onde um sistema de inferência *Neuro-fuzzy* com modelo TSK é aplicado nessa classe de sistema. A adaptação dos parâmetros do modelo TSK é realizada com o algoritmo *Shuffled Frog Leaping* (SFLA). O resultado deste trabalho compara o desempenho do uso SFLA em relação ao PSO e o AG na parametrização do modelo TSK na identificação de duas plantas MIMO, cujo modelo parametrizado com o SFLA apresentou acurácia maior comparada às demais técnicas com base no índice *RMSE*.

Em Millan & Barrios (2016) é realizado um estudo sobre modelagem *fuzzy* utilizando a estrutura TSK, o qual é ajustado com o algoritmo de clusterização Gustafson-Kessel. O algoritmo proposto é aplicado na identificação de um sistema MIMO, cujas variáveis analisadas são: fluxo; nível; temperatura. Os resultados mostram que o modelo *fuzzy* TSK proposto atinge resultados melhores com base nos índices de desempenho *RMSE* e *VAF* quando comparados com modelos paramétricos ARX, ARMAX e OE.

No trabalho de Paula (2016) é abordado a identificação de sistemas MIMO não lineares com modelos de blocos interconectados de Hammerstein-Wiener, onde o bloco dinâmico linear é estimado pelo algoritmo MOESP de identificação de subespaço. A técnica proposta é a um sistema de tanques acoplados e a um reator químico, ambos simulados.

Em Parikh (2016) um novo conceito de *framework* para identificação parcial de sistemas multivariáveis é abordado com o intuito de melhorar o desempenho de identificação de sistemas em malha aberta e fechada. Esta metodologia aborda a identificação de sistemas MIMO lineares com base na divisão das entradas da planta em três categorias: malha fechada crítica; malha aberta; malha fechada instável. O resultado deste trabalho mostra a aplicação do *framework* em uma planta MIMO de quatro tanques.

Identificação de sistemas não lineares com modelo multivariável de espaço de estados de Hammerstein, onde o bloco não linear é aproximado de um sistema de inferência *fuzzy* TSK é abordado no trabalho de Santos & Serra (2018). Os resultados mostram a identificação *on-line* de um sistema dinâmico MIMO com não linearidade complexa combinada, e também em um processo de evaporação de três entradas e três saídas.

No trabalho de Adánez, Al-Hadithi, & Jiménez (2019) é proposta uma metodologia de identificação de sistemas multivariáveis na qual o algoritmo genético é

utilizado para realizar o ajuste de funções de pertinência multidimensionais, e o sistema *fuzzy* TS é usado para modelar e identificar o sistema não linear. A presente metodologia é aplicada à um sistema de tanque de acoplado, e os resultados mostram que a metodologia proposta alcançou menor erro de identificação comparada a identificação com TS tradicional.

1.3 Objetivos

O principal objetivo deste trabalho é propor um *framework* capaz tanto de fazer identificação de sistemas MIMO não lineares no modelo *fuzzy* TSK multivariável, quanto realizar a seleção de modelo com o menor erro de estimação. A proposta é dividida em objetivos específicos, os quais são descritos abaixo.

- Realizar uma descrição geral sobre o *framework*, levando em consideração todas as suas etapas e características;
- Descrever a interligação dos Mínimos Quadrados e do PSO dentro do *framework*, e como estes realizam identificação de um sistema MIMO não linear no modelo *fuzzy* TSK multivariável;
- Aplicar o *framework* proposto em dois estudos de caso: Reator Contínuo de Tanque Agitado (do inglês, *Continuous Stirred Tank Reactor*); Secador Industrial (do inglês, *Industrial Dryer*);
- Comparar os resultados alcançados pelo *framework* com os obtidos pelo modelo de Hammerstein-Wiener e pela RNA, na identificação das plantas com os dados de teste, utilizando os índices Erro Quadrático Médio (*EQM*) e *VAF* (do inglês, *Variance Accounted For*), além da análise de resíduos;

1.4 Estrutura do trabalho

A estrutura deste trabalho está dividida em 5 capítulos. O presente capítulo trata da introdução do trabalho, onde são abordados o contexto do problema, a justificativa e a motivação, assim como os objetivos do trabalho. Os demais capítulos são descritos a seguir.

Capítulo 2: Apresenta-se neste capítulo um estudo da arte sobre identificação de sistemas MIMO não lineares, abordando de forma breve alguns dos algoritmos mais convencionais para se resolver este problema.

Capítulo 3: Neste capítulo é apresentado as características gerais do *framework* proposto, onde todas as suas etapas são descritas. Também é abordado neste capítulo, de forma detalhada, os algoritmos MQ e PSO, e o modelo *fuzzy* TSK multivariável. Também é abordada a conexão destes para realizar a identificação de sistema de sistemas MIMO.

Capítulo 4: São apresentados os resultados da aplicação do *framework* em duas plantas multivariadas reais, as quais são brevemente explicadas. Neste capítulo também é apresentada a metodologia de avaliação de desempenho aplicada para comparar os resultados do *framework* com o modelo de Hammerstein-Wiener e a RNA.

Capítulo 5: Neste capítulo são feitas as conclusões dos resultados alcançados no trabalho e as considerações finais, também são citadas as propostas de trabalhos futuros.

2 IDENTIFICAÇÃO DE SISTEMAS MULTIVARIÁVEIS NÃO LINEARES

2.1 Introdução

Sistemas multivariáveis, conforme previamente mencionado, podem ser classificados de acordo com o número de entrada e saídas presentes e analisadas no sistema, cujas classificações podem ser: sistema com uma entrada e várias saídas (SIMO); sistema com várias entradas e uma saída (MISO); com múltiplas entradas e múltiplas saídas (MIMO).

Os sistemas MIMO constituem uma boa parcela dos processos encontrados na vida real (Hanafy et al., 2014). A Figura 2.1 mostra um exemplo de um conjunto de quatro tanques de água interligados (Johansson, 2000), onde as variáveis u_1 e u_2 representam o sinal de corrente que controla o fluxo de água em duas válvulas diferentes, e as variáveis de interesse a serem controladas são os níveis de água y_1 e y_2 no tanque 1 e 2 respectivamente.

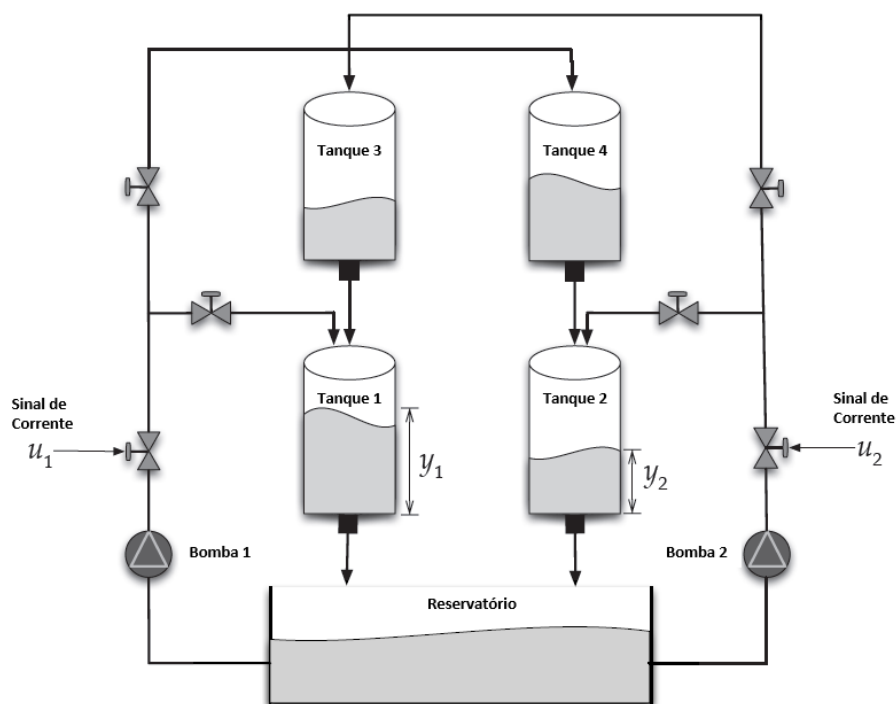


Figura 2.1 - Sistema de quatro tanques acoplados.

Fonte: Adaptado de Tangirala (2015).

O sistema descrito na Figura 2.1 é um exemplo clássico de sistema multivariável e não linear, e assim como este, existem diversos outros sistemas com aplicações diretas no ramo industrial e comercial.

As equações que modelam a dinâmica do sistema de quatro tanques por meio de equações diferenciais não lineares já são bem descritas na literatura, todavia, há muitas outras aplicações de interesse comercial que são bem mais complexas e difíceis de se modelar matematicamente, o que torna o uso das técnicas de identificação uma alternativa bastante viável para quem deseja levantar um modelo para o processo, de forma a controlá-lo e supervisioná-lo.

Do ponto de vista da identificação, sistemas MIMO podem ser resumidos conforme o diagrama da Figura 2.2. Nele estão presentes as variáveis de entrada $u_m(k) \in \mathbb{R}^m$ e as de saída $y_n(k) \in \mathbb{R}^n$, onde $k \in \mathbb{Z}$ representa o tempo discreto, m é a m -ésima entrada e n é a n -ésima saída.



Figura 2.2 - Diagrama esquemático de um sistema multivariável.

Identificar esta classe de sistema consiste primeiramente em definir um modelo matemático não linear e multivariável para representá-lo, e o passo seguinte consiste em utilizar um algoritmo de identificação para estimar os parâmetros do modelo previamente selecionado. A Figura 2.3 ilustra de maneira genérica o processo de identificação para sistemas multivariáveis.

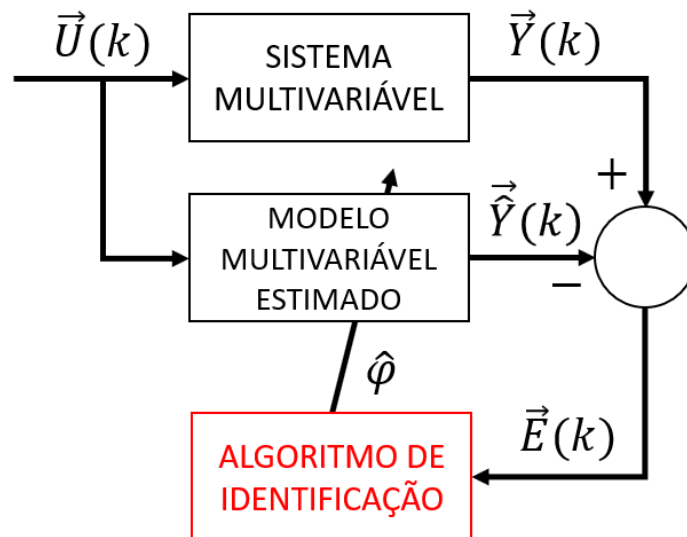


Figura 2.3 - Diagrama de identificação de sistema multivariável.

As variáveis $\vec{U}(k) \in \mathbb{R}^m$ e $\vec{Y}(k) \in \mathbb{R}^n$ representam respectivamente o vetor com todas as entradas e saídas observadas na planta, e $\vec{\hat{Y}}(k) \in \mathbb{R}^n$ representa o vetor com todas as saídas estimadas com o modelo multivariável, e a variável $\vec{E}(k) \in \mathbb{R}^n$ é o vetor dos sinais de erro entre as saídas observadas e as estimadas.

Portanto, a técnica de identificação tem como objetivo final estimar os parâmetros de um modelo que relacione as variáveis de entrada e saída de uma planta, de forma a representar a dinâmica desta. A seguir serão apresentadas algumas técnicas clássicas de identificação de sistemas multivariáveis e não lineares.

2.2 Técnicas clássicas de identificação de sistemas MIMO não lineares

Existem algumas abordagens clássicas para identificação de sistemas não lineares e que podem ser estendidos para sistemas multivariáveis MIMO. Cada abordagem possui um modelo e sua respectiva técnica de estimação paramétrica (Boutalis, et al., 2014). Abaixo são listados alguns exemplos de modelos que podem ser utilizados para representar sistemas multivariáveis.

- Modelos de Hammerstein-Wiener;
- Séries de Volterra;
- Redes Neurais Artificiais;
- Redes *Neuro-Fuzzy*;

Séries de Volterra, Redes Neurais Artificiais (RNA) e modelos NARMAX (do inglês, *Nonlinear Autoregressive Moving Average Model with Exogenous Input*) são exemplos de modelos muito usados para estimar uma planta não linear, com o intuito de representar sua dinâmica (Zhu & Backx, 1993). Entretanto, identificar sistemas multivariáveis com esses modelos apresenta algumas dificuldades, como: considerável custo computacional para modelos com muitas variáveis; dificuldade manipulações matemáticas com os modelos identificados; grande quantidade de parâmetros a serem estimados para plantas pouco lineares, como é o caso da série de Volterra (Aguirre, 2014).

Outra vertente na identificação desta classe de sistemas é o uso de modelos de blocos interconectados (multimodelos), tendo como exemplos: modelo de Hammerstein, modelo de Wiener, modelo de Hammerstein-Wiener, Realimentação Estática. Estes modelos possuem a vantagem de serem de baixo custo de identificação e de testes computacionais, todavia, apresentam deficiência na descrição global de sistemas (Paula;

M. Thomas, 2016). Por outro lado, o uso de redes *Neuro-Fuzzy* ANFIS, na estratégia multimodelos, possui boa eficiência na identificação de sistemas multivariáveis, porém, demandam alto custo computacional (Rodrigues, 2010; Azevedo, 2019).

Nas seções seguintes serão descritos detalhadamente os métodos de Hammerstein-Wiener e Redes Neurais Artificiais para identificação de sistemas MIMO não lineares, uma vez que tais estratégias são bastante utilizadas na resolução deste problema.

2.2.1 Modelo de Hammerstein-Wiener

Modelo de Hammerstein e de Wiener são estruturas não lineares simples de blocos interconectados. Estes modelos são largamente usados na identificação de sistemas multivariáveis na forma multimodelos (Yan et al., 2013; Paula, 2016). Ao longo dos anos, vários métodos têm sido propostos para a identificação da estrutura do modelo HW (M. Thomas; F. Allgower, 2010).

O modelo Hammerstein-Wiener é uma união entre o modelo de Hammerstein proposto por Narendra e Gallman (1966) e a teoria de Wiener, o qual se caracteriza pela conexão em cascata de uma função não linear estática (N_1) seguida de um sistema dinâmico linear ($G_s(q)$) e seguido por uma saída estática não linear (N_2), conforme a Figura 2.4, onde $u(k)$ e $v_1(k)$ são a entrada e a saída da função estática N_1 respectivamente, e $v_2(k)$ e $y(k)$ são respectivamente a entrada e a saída da função estática não linear N_2 .

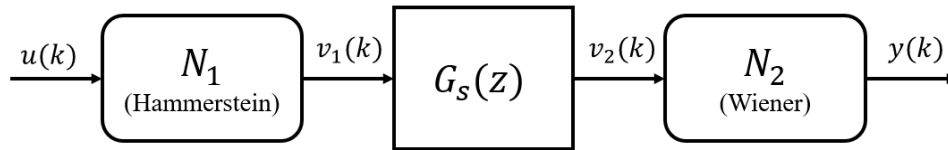


Figura 2.4 - Modelo SISO de Hammerstein-Wiener.

No diagrama da figura acima, os termos N_1 e N_2 correspondem a uma classe de funções estáticas não lineares como por exemplo: polinomial; descontínua; *piecewise* linear; sigmoide; etc. O termo $G_s(z)$ é uma função de transferência discreta, a qual pode ser descrita conforme a Equação 2.1, e que também pode ser escrita na forma ARX (do inglês, *Auto-Regressive with Exogenous Input*), onde b_l e a_l são os coeficientes da função de transferência, e o coeficiente q é o operador de atraso.

$$G_s(z) = \frac{B(z)}{A(z)} = \frac{b_0 + b_1 z^{-1} + \dots + b_l z^{-l}}{1 - a_1 z^{-1} - \dots - a_l z^{-l}} \quad (2.1)$$

Entretanto, para sistemas multivariáveis, o modelo de HW apresenta algumas modificações para o acoplamento das variáveis de entrada e saída do processo, conforme é mostrado no diagrama de blocos da Figura 2.5.

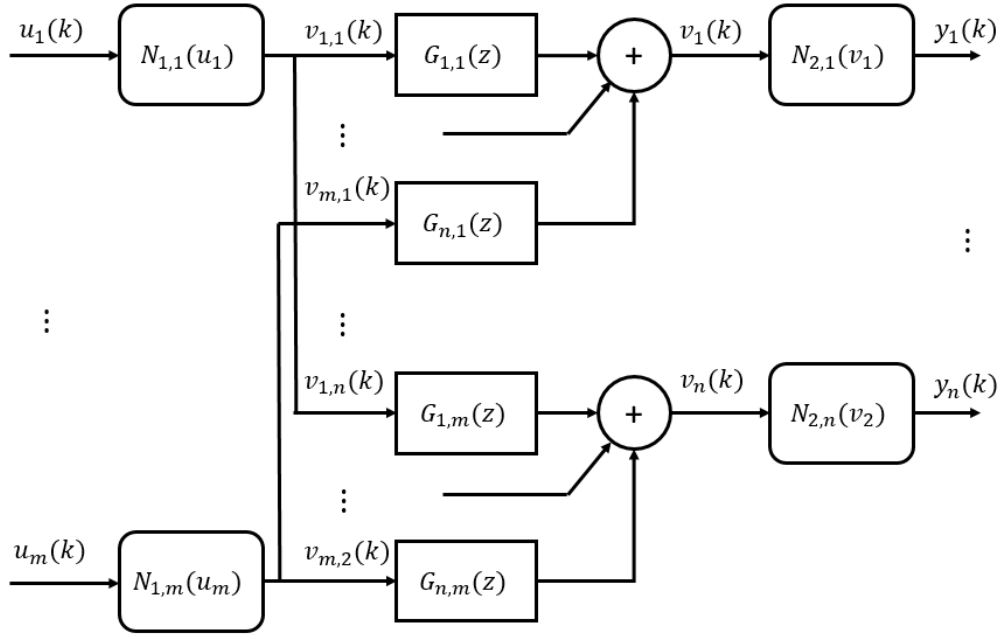


Figura 2.5 – Modelo MIMO de Hammerstein-Wiener.

Conforme pode ser visto na figura acima, o modelo MIMO de Hammerstein-Wiener pode ser composto por n modelos MISO, de modo que cada saída apenas dependa do acoplamento das m entradas e dos valores regressivos de sua respectiva saída.

Existem diversas estratégias para a realização da identificação desta classe de modelo multivariável, como por exemplo: Identificação de Subespaços (Goethals et al, 2005); Identificação Recursiva de Subespaços (Lovera et al., 2000); Identificação de Modelos de Bloco Estruturados (Goethals et al, 2005); etc.

Um dos métodos de estimação dos parâmetros do modelo MIMO de Hammerstein-Wiener consiste em isolar a equação paramétrica de cada parte MISO (Zhu, 2001). A Equação 2.2 mostra o modelo MISO isolado na forma de função custo, onde $\{i \in \mathbb{Z}/ 1 \leq i \leq m\}$ e $\{j \in \mathbb{Z}/ 1 \leq j \leq n\}$ são variáveis auxiliares para indicar a m -ésima entrada e a n -ésima saída respectivamente.

$$V_{ARX}(j) = \sum_{k=1}^{N_s} \left\{ \sum_{i=1}^m \{A_{j,i}(z)N_{2,j}^{-1}[y_j(k)] - B_{j,i}(z)N_{j,i}[u_i(k)]\} \right\}^2 \quad (2.2)$$

As etapas para estimação dos parâmetros $A_{j,i}(z)$, $B_{j,i}(z)$, $N_{2,j}^{-1}[y_j(k)]$ e $N_{j,i}[u_i(k)]$ são enumeradas abaixo:

- 1) Primeiramente, define-se $j = 1$;
- 2) Define-se $N_{2,j}^{-1}[y_j(k)] = y_j(k)$ e $N_{j,i}[u_i(k)] = u_i(k)$, e utiliza-se Mínimos Quadrados para a estimação de $A_{j,i}(z)$ e $B_{j,i}(z)$;
- 3) Calcula-se $N_{j,i}[u_i(k)]$ para valores fixados de $N_{2,j}^{-1}[y_j(k)] = y_j(k)$, $A_{j,i}(z)$ e $B_{j,i}(z)$, minimizando a Equação 2.2 com os devidos valores fixados;
- 4) Calcula-se $N_{2,j}^{-1}[y_j(k)]$ para valores fixados de $N_{j,i}[u_i(k)] = u_i(k)$, $A_{j,i}(z)$ e $B_{j,i}(z)$, minimizando a Equação 2.2 com os devidos valores fixados;
- 5) Volte à etapa 1) fazendo $j = j + 1$;

Conforme observado no algoritmo acima, o sistema MIMO de Hammerstein-Wiener pode ser decomposto em blocos MISO, cujo modelo é identificado utilizando-se o algoritmo dos Mínimos Quadrados para a estimação da parte dinâmica linear ARX de cada acoplamento HW. Entretanto, a estimação dos parâmetros das funções não lineares depende de algoritmos de busca e otimização não linear como: Levenberg-Marquardt (Gavin, 2019); *Support Vector Machine* (SVM) (M. Thomas, F. Allgower); etc.

2.2.2 Redes Neurais Artificiais

A rede neural artificial (RNA) são modelos matemáticos bioinspirados na estrutura de neurônios do cérebro humano. Em outras palavras, a RNA é uma máquina projetada para modelar a maneira como o cérebro realiza uma tarefa particular ou função de interesse (Haykin, 2001). Essa estrutura também é utilizada na representação não linear de sistemas (Aguirre, 2014).

As RNAs são compostas por camadas de neurônios artificiais interconectados. A Equação 2.3 mostra o modelo de um neurônio artificial com m entradas, onde $u_i \in \mathbb{R}$ representa as entradas da rede, $w_i \in \mathbb{R}$ são os pesos sinápticos, $\bar{b} \in \mathbb{R}$ é o *bias* e h_a uma função de ativação, onde $\{i \in \mathbb{Z} / 1 \leq i \leq m\}$.

$$y = h_a \left(\sum_{i=1}^m w_i u_i + \bar{b} \right) \quad (2.3)$$

Existem diversas funções de ativação que podem ser usadas na rede neural, como por exemplo: linear; log-sigmoide; tangente hiperbólica sigmoide; saturada; degrau; etc (Hagan, et al, 2014).

Para identificação de sistemas multivariáveis não lineares, um exemplo de formulação matemática da RNA na estrutura de uma MLP (do inglês, *Multilayer Perceptron*), com uma camada escondida e uma camada de saída, pode ser vista na Equação 2.4, onde $U_i(k - \tau_d)$ e $Y_j(k)$ são respectivamente a entrada e a saída da RNA, e $\tau_d \in \mathbb{Z}$ é o atraso do sinal, e i e j são variáveis auxiliares das entradas e saídas da rede.

$$\vec{Y}_j(k) = h_{2,s} \left(\sum_{j=1}^n w_j h_{1,j} \left(\sum_{i=1}^m w_i \vec{U}_i(k - \tau_d) + \bar{b}_{1,i} \right) + \bar{b}_{2,j} \right) \quad (2.4)$$

A Figura 2.6 ilustra a estrutura de uma rede MLP para sistema MIMO, a qual é descrita na equação acima, onde os pesos sinápticos foram suprimidos para melhorar a visualização da RNA.

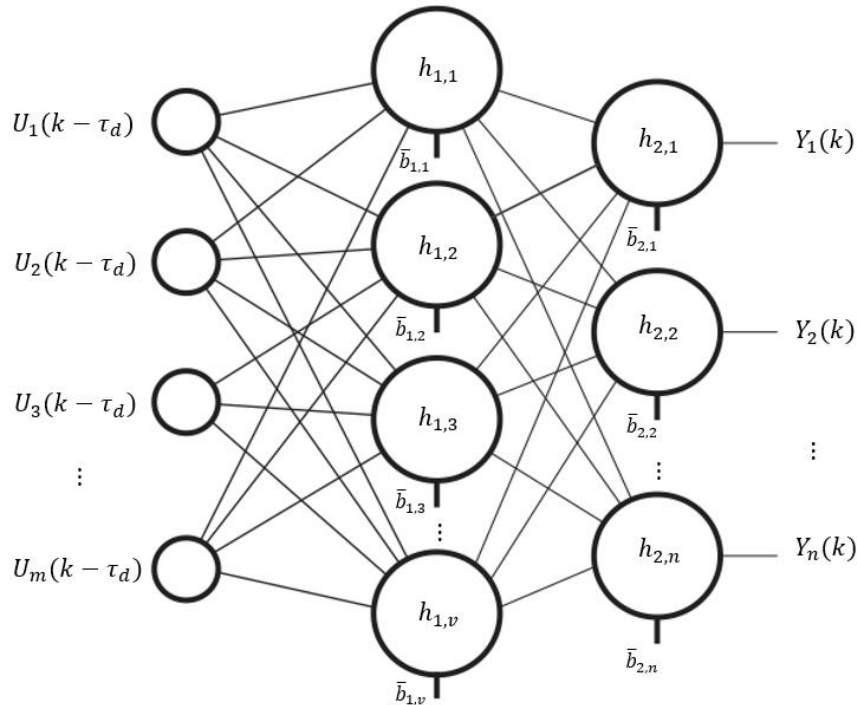


Figura 2.6 – Estrutura de uma MLP para sistema MIMO.

Na prática, as entradas $U_i(k - \tau_d)$ da rede para a representação de sistemas MIMO são constituídos pelas m entradas defasadas da planta e pelos valores regressivos

das suas saídas. Para o caso especial onde as funções de ativação da rede são lineares, a MLP equivale-se a várias estruturas NARX (do inglês, *Nonlinear Auto-Regressive with Exogenous Input*) acopladas.

Com relação a estimação dos valores dos pesos sinápticos e dos *bias* da MLP, existem diversos algoritmos de estimação não linear, tais como: retropropagação do erro (do inglês *backpropagation*) (Haykin, 2001); método de newton; gradiente descendente (Hagan et al., 2014); Levenberg-Marquardt (Fu et al., 2015); etc. Estes algoritmos são técnicas determinísticas de busca e otimização, que tem por finalidade parametrizar a rede neural de modo a reduzir o erro de estimação.

Após as fases de treinamento (etapa onde a rede é parametrizada com o uso de algum algoritmo de treinamento) e validação (etapa onde se avalia a capacidade da rede generalizar o modelo a partir de um conjunto de dados), a RNA é capaz de representar totalmente ou parcialmente o modelo de um sistema dinâmico MIMO não linear.

2.3 Conclusão

Neste capítulo foram apresentados alguns conceitos básicos sobre modelos e técnicas de identificação clássicas de sistemas MIMO não lineares, com ênfase nas características dos modelos de Hammerstein-Wiener e nas Redes Neurais Artificiais, assim como também são citados alguns algoritmos de estimação paramétrica destes modelos. No próximo capítulo será apresentado o *framework* proposto para identificação de sistemas multivariável não lineares.

3 FRAMEWORK PARA IDENTIFICAÇÃO DE SISTEMAS MIMO NÃO LINEARES

Neste capítulo é descrita a estrutura do *framework* proposto, abordando cada uma de suas etapas, e como este é utilizado na identificação de sistemas MIMO não lineares. Também são descritos com detalhes os algoritmos Mínimos Quadrados e PSO, além do modelo *fuzzy* TSK multivariável, que também são utilizados nesta estrutura.

3.1 Estrutura Geral do *Framework*

Um *framework* consiste em uma estrutura básica que contém ferramentas, guias e subsistemas, conectando vários mecanismos mais adequado para auxiliar no desenvolvimento de soluções (Negash et al, 2016). Para o caso da identificação de sistemas dinâmicos, o *framework* proposto, Figura 3.1, uni alguns algoritmos e condições lógicas tanto para parametrizar um modelo *fuzzy* TSK multivariável, o qual representa o sistema MIMO não linear.

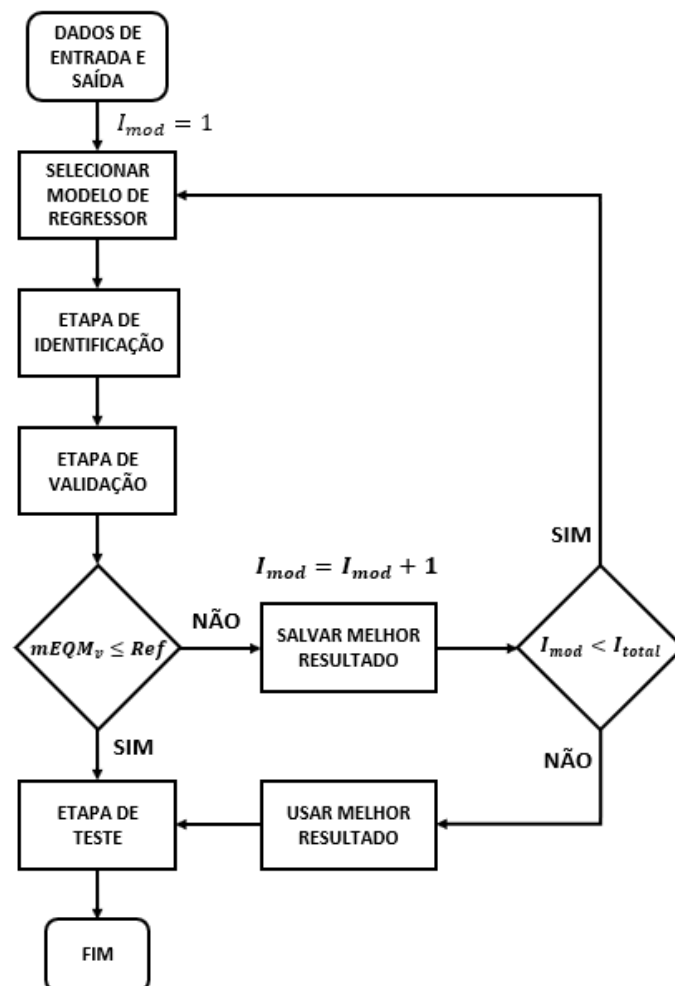


Figura 3.1 – Fluxograma do *Framework* proposto para identificar sistemas MIMO não lineares.

Além de realizar a identificação do sistema MIMO, o *framework* também seleciona o melhor vetor regressor para ser usado juntamente com o modelo *fuzzy* TSK multivariável que represente o sistema com o menor erro possível. A seguir são descritas as etapas do fluxograma acima.

- **Dados de Entrada e Saída:** Esta etapa consiste basicamente em dividir as N_{total} amostras do número total de entradas e saídas analisadas no sistema MIMO que se deseja identificar. Os dados totais de amostra são subdivididos em: dados de identificação (N_{id}); dados de validação (N_v); e dados de teste (N_t). Cada uma destas amostras será utilizada em etapas específicas do *framework*.

- **Selecionar Modelo de Regressor:** O regressor é um vetor $\phi(n, m, \tau_{du}, \tau_{dy}) \in \mathbb{R}^{1 \times (n\tau_{du} + m\tau_{dy})}$ utilizado na etapa de identificação, especificamente no cálculo dos Mínimos Quadrados (MQ), e que também é usado na estrutura do modelo *fuzzy* TSK final. O regressor possui uma estrutura pré-definida utilizando sinais defasados de entrada e saída da planta, onde $\tau_{du} \in \mathbb{Z}$ e $\tau_{dy} \in \mathbb{Z}$ representam o número de atrasos do sinal das m entradas e das n saídas, respectivamente.

A Equação 3.1 exemplifica o regressor de um sistema MIMO de duas entradas e duas saídas, com $\tau_{du} = 1$ e $\tau_{dy} = 1$.

$$\phi(2,2,1,1)^T = \begin{bmatrix} y_1(k-1) \\ y_2(k-1) \\ u_1(k-1) \\ u_2(k-1) \end{bmatrix} \quad (3.1)$$

A ordem, o formato, número de atrasos por variáveis de entrada e saída, assim como a quantidade de vetores regressores usados para a seleção, são definidos pelo usuário antes da execução do *framework*. Em resumo, esta etapa apenas vai selecionar um regressor de forma consecutiva para ser usado na etapa seguinte de identificação.

Na primeira iteração do *framework*, um dos regressores é selecionado para compor o modelo *fuzzy* TSK multivariável na etapa de identificação e validação. Caso este modelo com o primeiro vetor regressor não passe na etapa de validação, o próximo regressor é selecionado para ser usado nas mesmas etapas, e assim sucessivamente, ou seja, os regressores são selecionados de forma consecutiva e em ordem pré-definida pelo usuário.

- **Etapa de Identificação:** Esta pode ser considerada a etapa mais importante do *framework*, pois é nela que ocorre o processo de identificação de sistema multivariável não linear utilizando o modelo *fuzzy* TSK multivariável. Os algoritmos MQ e PSO usados para estimar respectivamente a matriz de parâmetros $\hat{H} \in \mathbb{R}^{(n) \times (n\tau_{du} + m\tau_{dy})}$ e o vetor de desvio padrão $\vec{\sigma}_G \in \mathbb{R}^{N_{var}}$ das *Gaussianas* das entradas do modelo multivariável identificado.

Como o PSO é um algoritmo de busca e otimização estocástica, o que não garante que o vetor $\vec{\sigma}_G$ seja estimado sempre com o mesmo valor para as mesmas condições iniciais, esta etapa é executada N_{sim} vezes para cada regresso selecionado. E no final de todas as simulações, o modelo estimado nesta etapa, e utilizado nas etapas seguintes, constitui-se da média dos parâmetros estimados pelos dois algoritmos (Oliveira et al., 2019). Os critérios de para o PSO é definido nesta metodologia com sendo o número de iterações e um valor de referência para o índice $mEQM_i$, explicado no Tópico 3.2.

Para a execução da identificação, apenas uma parcela das amostras dos dados das entradas e saídas da planta é utilizada, pois o restante será utilizado nas etapas de validação e teste. O Tópico 3.2 explicará com mais detalhes a etapa de identificação.

- **Etapa de Validação:** A validação consiste em verificar se o modelo identificado alcançou um nível desejável em relação à algum critério de desempenho (Aguirre, 2014). Para este *framework*, o critério estipulado utiliza os N_v dados de validação (saídas observadas) juntamente com os dados estimados (saídas estimadas) pelo modelo identificado para calcular o EQM_v da etapa de validação, conforme a Equação 3.2, e a Equação 3.3 mostra o $mEQM_v$ (média dos EQM_v de todas as saídas), onde $y_j(k)$ são os dados observados e $\hat{y}_j(k)$ os estimados da j -ésima saída.

$$EQM_v = \sum_{k=N_{id}+1}^{N_{id}+N_v} \frac{[y_j(k) - \hat{y}_j(k)]^2}{N_v} \quad (3.2)$$

$$mEQM_v = \frac{1}{n} \sum_{j=1}^n (EQM_v) \quad (3.3)$$

Se o valor de $mEQM_v$ for menor ou igual ao valor de referência pré-definido pelo usuário, então o *framework* executará a fase de teste com o modelo de regressor atualmente utilizado. Caso contrário, o modelo com o melhor resultado (menor $mEQM_v$

já alcançado) será salvo, e o ciclo se repetirá até que este critério seja alcançado ou o índice I_{mod} , que indica a contagem dos modelos de regressores testados, seja maior que o número total de modelos I_{Total} .

Em resumo, na etapa de validação, se o modelo de regressor passar no critério de validação ou o número total de modelos testados já tenha sido ultrapassado, o *framework* executará a fase de teste com o regressor que alcançou menor $mEQM_v$.

- **Etapas de Teste:** Esta é a última etapa do *framework*, a qual consiste em testar as N_t amostras remanescentes com o modelo identificado. Nesta etapa é possível ver se o modelo estimado é capaz de representar a dinâmica do sistema real em um ponto de operação diferente dos dados de identificação e validação, assim como também pode-se aplicar métricas para avaliar o erro de estimação, conforme será visto no Capítulo 4.

As etapas descritas acima ilustram de maneira geral as etapas e o fluxo de informação dentro *framework* proposto neste trabalho. O tópico a seguir descreverá de maneira mais detalhada a etapa de identificação, a qual envolve a interconexão de diferentes algoritmos.

3.2 Algoritmos da Etapa de Identificação

A etapa de identificação do *framework* é a mais complexa devido ao fato desta conter dois algoritmos diferentes como os MQ e o PSO para estimar parâmetros, além de utilizar o modelo *fuzzy* TSK multivariável. Cada um dos algoritmos possui uma função específica nesta etapa, todavia, o processo de identificação da planta MIMO depende da atuação deles de forma conjunta. A Figura 3.2 mostra o fluxograma geral desta etapa, destacando a relação de cada algoritmo.

As variáveis $\vec{U}(k) \in \mathbb{R}^m$ e $\vec{Y}(k) \in \mathbb{R}^n$ representam respectivamente o vetor com todas as entradas e saídas observadas na planta MIMO, e $\vec{\hat{Y}}(k) \in \mathbb{R}^n$ representa o vetor com todas as saídas estimadas com o modelo *fuzzy* TSK multivariável, e a variável $\vec{E}(k) \in \mathbb{R}^n$ é o vetor dos sinais de erro entre as saídas observadas e as estimadas.

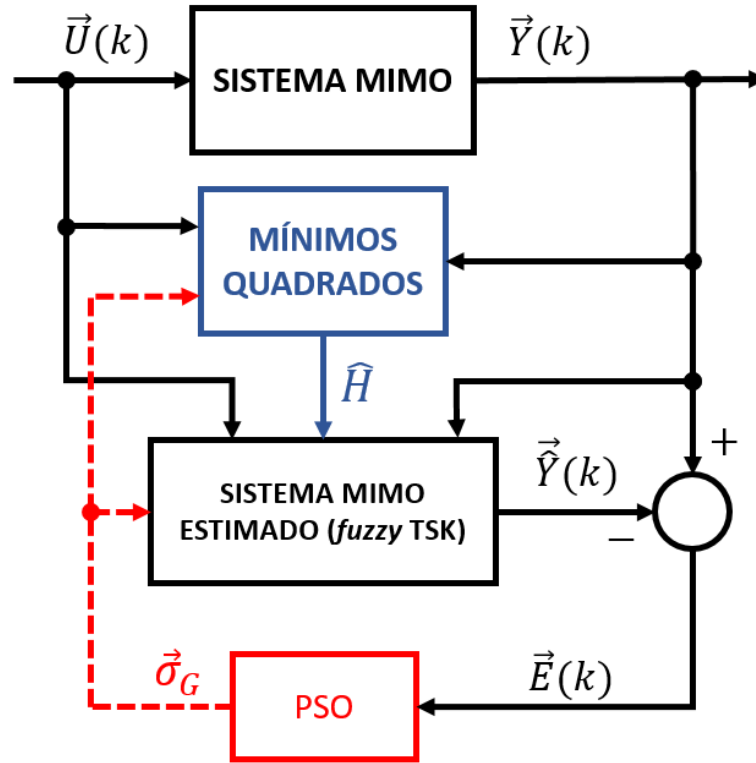


Figura 3.2 – Etapa de identificação do *framework*.

A Figura 3.2 ilustra a etapa responsável propriamente por identificar a planta MIMO não linear, e para que isso ocorra, são necessárias duas informações. A primeira consiste na matriz estimada de parâmetros $\hat{H}$, cujos parâmetros irão compor o modelo *fuzzy* TSK multivariável. Esta matriz é estimada pelos Mínimos Quadrados, o qual depende basicamente das informações de entrada e saída da planta real, e também depende do valor das funções de pertinência *fuzzy* das entradas.

A segunda informação consiste em estimar o vetor de desvio padrão $\vec{\sigma}_G$ por meio do PSO. Este vetor é constituído pelo conjunto de desvios padrões, onde cada desvio é utilizado no conjunto de funções de pertinência em sua respectiva entrada do modelo *fuzzy* TSK multivariável. O PSO, em seu processo de busca e otimização, realiza a minimização do valor da média do $mEQM_i$ entre as saídas e estimadas da planta utilizando apenas as amostras de dados de identificação, conforme a Equação 3.4.

$$mEQM_i = \frac{1}{n} \sum_{j=1}^n \left(\sum_{k=1}^{N_{id}} \frac{[y_j(k) - \hat{y}_j(k)]^2}{N_{id}} \right) \quad (3.4)$$

Portanto, o modelo *fuzzy* TSK identificado que representa o sistema MIMO não linear real depende da ação conjunta do MQ e do PSO para estimar os parâmetros do modelo, utilizando apenas informações da própria planta.

É importante destacar que a etapa de identificação do *framework* é simulada N_{sim} vezes para as mesmas N_{id} amostras de dados, uma vez que o PSO, que é um algoritmo de otimização estocástica, não garante que o mesmo sempre alcançará o mesmo resultado na estimação dos desvio padrões em cada simulação. Portanto, o modelo *fuzzy* TSK multivariável estimado nesta etapa, e que será usado nas etapas seguintes (validação e teste) é formado pelo valor médio de $\hat{H}$ e $\vec{\sigma}_G$ estimados dentre todas as execuções da etapa de identificação.

Os tópicos abaixo descrevem de maneira mais detalhada cada um dos algoritmos pertencentes à etapa de identificação.

3.2.1 Sistema *fuzzy* TSK

a) Características Gerais do sistema *fuzzy* TSK

A lógica *fuzzy* é um sistema inteligente que usa o conhecimento especialista com base em regras ‘SE-ENTÃO’, as quais relacionam de maneira lógica e matemática funções de pertinências contínuas (Passino; Yurkovich, 1998). A lógica *fuzzy* é usada principalmente para modelar de maneira inteligente sistemas não lineares e complexos com base no conhecimento de um especialista. Ao longo dos anos, diversas aplicações com sistema *fuzzy* vem sendo usadas em diversos ramos da ciência e da engenharia, como: tomada de decisão; projeto de controle; identificação de sistemas; clusterização; processamento de sinais; etc (Wang, 1997; Lilly, 2010).

Mamdani e Takagi-Sugeno-Kang (TSK) são os dois principais tipos de máquina de inferência *fuzzy*, os quais possuem estruturas diferentes, principalmente porque o TSK apresenta como saída funções ponderadas ao invés do estágio de defuzzificação que é visto na máquina de Mamdani. A saída do TSK basicamente é uma média ponderada entre funções ou constantes e as próprias funções de pertinência do conjunto *fuzzy*.

A Figura 3.3 ilustra o diagrama de um sistema *fuzzy* TSK, onde $u \in \mathbb{R}^m$ e $y^{crisp} \in \mathbb{R}$ são respectivamente a entrada e a saída do sistema.

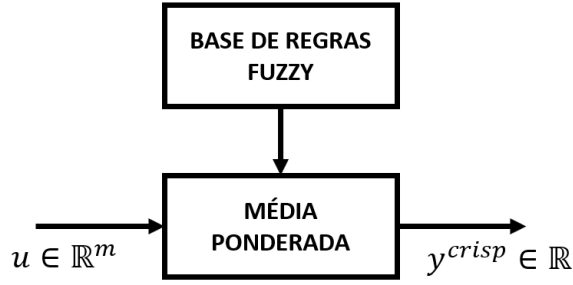


Figura 3.3 – Diagrama do sistema *fuzzy* TSK.

Os sistemas *fuzzy* são baseados em regras de associações de conjuntos, os quais se associam à valores numéricos das funções de pertinência das entradas do sistema, ou seja, cada entrada pode possuir diversos conjuntos. Cada um desses conjuntos está associado à uma função de pertinência $\mu_m^{\rho_{\{1...l_m\}}}(u(k)) \in \mathbb{R}$, a qual pode ser do formato triangular, trapezoidal, *Gaussiana*, sigmoide, etc (Lilly, 2010).

Considere a R -ésima regra de um sistema *fuzzy* TSK, conforme mostrado abaixo, onde f_R é a função ou constante relacionada com a saída y^R .

$$R: \text{SE } u_1 \text{ é } A_1^{\rho_{l_{m1}}} \text{ e } u_2 \text{ é } A_2^{\rho_{l_{m2}}} \text{ e ... e } u_m \text{ é } A_m^{\rho_{l_{mm}}}, \\ \text{ENTÃO } y^R = f_R(\bullet)$$

Os conjuntos de cada entrada do sistema *fuzzy* são representados por $A_m^{\rho_{\{1...l_m\}}}$, onde $\rho_{\{1...l_m\}}$ é o índice que designa os diferentes conjuntos A para a m -ésima entrada, assim como as funções de pertinência μ associadas a cada conjunto, e l_m é o total de conjuntos por entrada. A Figura 3.4 exemplifica a entrada com três conjuntos *fuzzy*, onde $A_1^{\rho_1}$ e $A_1^{\rho_3}$ correspondem a funções de pertinência sigmoides e $A_1^{\rho_2}$ é uma função *Gaussiana*.

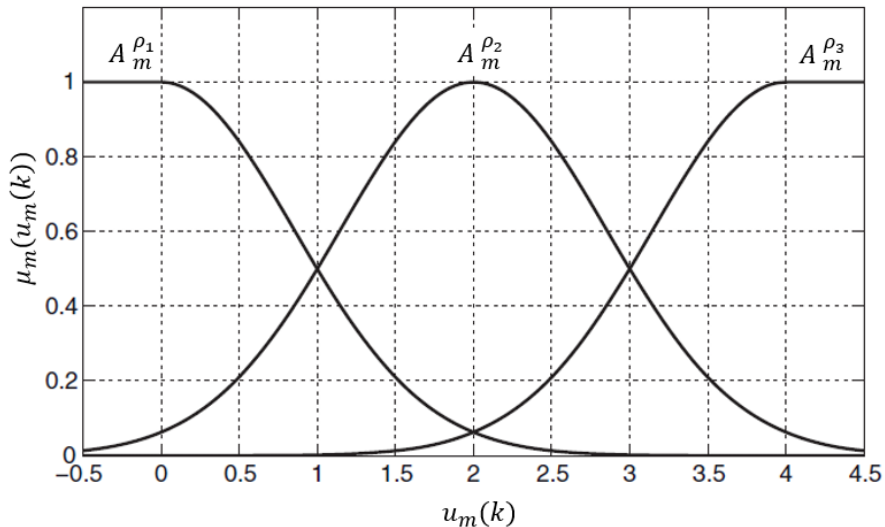


Figura 3.4 – Exemplo de conjuntos *fuzzy*.

O valor de saída y^{crisp} do sistema *fuzzy* TSK é uma ponderação da função (ou constante) de saída de cada regra com o valor da associação das funções de pertinência das entradas, Equação 3.5. Onde $\mu_R(u_1, \dots, u_m)$ corresponde ao valor da associação das funções de pertinência de cada R -ésima regra com o uso de um operador norma S ou norma T (Wang, 1997).

$$y^{crisp} = \frac{\sum_{R=1}^{S_R} y^R \mu_R(u_1, \dots, u_m)}{\sum_{R=1}^{S_R} \mu_R(u_1, \dots, u_m)} \quad (3.5)$$

Onde S_R representa o número total de regras possíveis que podem ser utilizadas no sistema *fuzzy* TSK, cujo valor pode ser calculado conforme a Equação 3.6. Entretanto, a quantidade e a forma das regras utilizadas podem ser ajustadas pelo especialista.

$$S_R = 2^m \quad (3.6)$$

Na seção seguinte será abordado o modelo matemático *fuzzy* para sistema MIMO proposto neste trabalho, o qual é utilizado na etapa de identificação.

b) Modelo *fuzzy* TSK para Sistema MIMO

A estrutura geral do modelo *fuzzy* TSK para sistemas multivariáveis proposto como metodologia deste trabalho na etapa de identificação pode ser visto nas Equações 3.7 a 3.9.

$$\begin{bmatrix} \hat{y}_1(k) \\ \vdots \\ \hat{y}_n(k) \end{bmatrix} = \sum_{R=1}^{S_R} d^R \xi_R(k) \quad (3.7)$$

$$d^R = H^R \phi(n, m, \tau_{du}, \tau_{dy})^T \quad (3.8)$$

$$\xi_R(k) = \frac{\mu_R(\tilde{u}_1(k), \dots, \tilde{u}_m(k))}{\sum_{R=1}^{S_R} \mu_R(\tilde{u}_1(k), \dots, \tilde{u}_m(k))} \quad (3.9)$$

O modelo *fuzzy* TSK para sistema MIMO é sintetizado na Equação 3.7, a qual mostra a relação entre as n saídas $\hat{y}_n(k)$ do modelo estimado em função de parâmetros do termo d^R e da ponderação $\xi_R(k) \in \mathbb{R}$ das funções de pertinência, onde os termos $\tilde{u}_1(k), \dots, \tilde{u}_m(k)$ são os valores das m entradas da planta normalizadas entre 0 e 1.

Portanto, percebe-se que o modelo multivariável proposto consiste em ponderações dos termos d^R com os termos $\xi_R(k)$ em função da quantidade de regras do sistema *fuzzy*.

O termo d^R é calculado com o produto entre a matriz de parâmetros H^R e o regressor $\phi(n, m, \tau_{du}, \tau_{dy})$ selecionado pelo *framework*. Este produto está em função da R -ésima regra, ou seja, para cada regra existe uma matriz de parâmetros diferente, a qual multiplica com o vetor regressor. As Equações 3.10 e 3.11 mostram respectivamente a estrutura genérica da matriz de parâmetros e do vetor regressor.

$$H^R = \begin{bmatrix} a_{1\varphi_1}^R & \cdots & a_{1\varphi_f}^R & b_{1\varphi_1}^R & \cdots & b_{1\varphi_f}^R \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ a_{n\varphi_1}^R & \cdots & a_{n\varphi_f}^R & b_{n\varphi_1}^R & \cdots & b_{n\varphi_f}^R \end{bmatrix} \quad (3.10)$$

$$\phi(n, m, \tau_{du}, \tau_{dy})^T = \begin{bmatrix} y_1(k-1) \\ \vdots \\ y_1(k-\tau_{dy}) \\ \vdots \\ y_n(k-1) \\ \vdots \\ y_n(k-\tau_{dy}) \\ u_1(k-1) \\ \vdots \\ u_1(k-\tau_{du}) \\ \vdots \\ u_m(k-1) \\ \vdots \\ u_m(k-\tau_{du}) \end{bmatrix} \quad (3.11)$$

A Equação 3.10 mostra uma matriz que está relacionada à R -ésima regra *fuzzy*, e que também é dividida em duas partes. A primeira parte é composta por parâmetros $a_{n\varphi_f}^i$, os quais estão associados aos termos regressores de saída da planta, e a segunda parte é formada por parâmetros $b_{n\varphi_f}^i$, os quais estão associados às entradas. Os índices $\varphi_1, \dots, \varphi_f$ são utilizados para indicar com qual termo do vetor regressores cada parâmetro de H^R está associado.

O vetor de regressor mostrado na Equação 3.11 é uma maneira generalizada de propor a relação de cada saída da planta com os valores regressivos das saídas e das entradas em função de um número pré-definido de atrasos. Para o *framework* proposto, a estrutura deste vetor pode variar conforme a escolha do usuário.

O fator $\xi_R(k)$ da Equação 3.9 são relações ponderadas das funções de pertinência, onde a relação *fuzzy* da R -ésima regra é dividida pela soma das relações de todas as regras. O termo $\mu_R(\tilde{u}_1(k), \dots, \tilde{u}_m(k))$ consiste em uma operação *fuzzy*, que para este trabalho é utilizado a norma T do valor das funções de pertinência, a qual utiliza os dados normalizados das entradas da planta MIMO, conforme mostrado na Equação 3.12.

$$\mu_R(\tilde{u}_1(k), \dots, \tilde{u}_m(k)) = \mu_1^{p_{(1,2)}}(\tilde{u}_1(k)) \dots \mu_m^{p_{(1,2)}}(\tilde{u}_m(k)) \quad (3.12)$$

Conforme a equação acima, a norma T consiste na multiplicação das funções de pertinência dos conjuntos *fuzzy* associados por cada regra (Wang, 1997). Cada uma dessas funções corresponde a um conjunto que está associado a uma entrada do sistema MIMO analisado. O índice $p_{(1,2)}$ na equação acima é utilizado para diferenciar as funções utilizadas em cada entrada *fuzzy*, e também pode ser decomposto em p_1 e p_2 .

A função de pertinência utilizada no modelo *fuzzy* TSK multivariável deste trabalho é a *Gaussiana*, a qual pode assumir valores reais entre 0 e 1, conforme visto nas relações da Equação 3.13 e na Figura 3.5.

$$\mu_m^{p_{(1,2)}}(\tilde{u}_m(k)) = e^{-\frac{1}{2} \left(\frac{\tilde{u}_m(k) - \bar{c}_{(1,2)m}}{\sigma_{(1,2)m}} \right)^2} \quad (3.13)$$

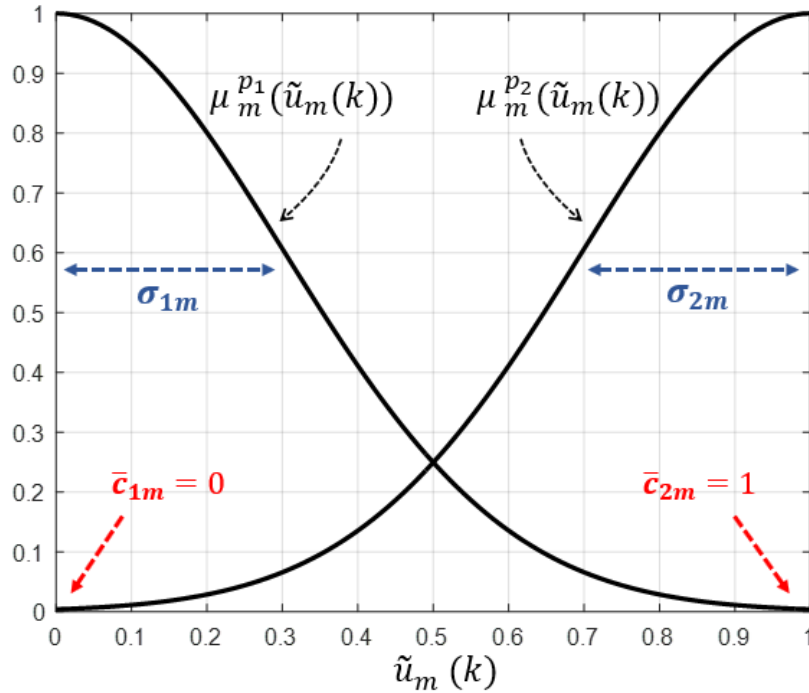


Figura 3.5 – Entrada do sistema *fuzzy* TSK com funções de pertinência *Gaussiana*.

Para o *framework* proposto, cada m -ésima entrada possui apenas duas *Gaussianas* que possuem limites entre 0 e 1 no domínio e no contradomínio. Cada função possui um desvio padrão, σ_{1m} ou σ_{2m} , os quais variam de entrada para entrada e de função para função na mesma entrada. Estes desvios padrões são estimados usando o PSO. As *Gaussianas* de cada entrada possuem valores de média $\bar{c}_{1m}$ ou $\bar{c}_{2m}$ fixadas em 0 e 1 respectivamente, conforme visto na Figura 3.5.

A Equação 3.14 é um exemplo compactado de um modelo *fuzzy* TSK multivariável com duas entradas e duas saídas, e a Equação 3.15 mostra a forma estendida do mesmo modelo para $\tau_{du} = 1$ e $\tau_{dy} = 1$, onde H^1 a H^4 são divisões da matriz $\hat{H}$, conforme mostrado na Equação 3.19.

$$\begin{bmatrix} \hat{y}_1(k) \\ \hat{y}_2(k) \end{bmatrix} = H^1 \phi^T \xi_1(k) + H^2 \phi^T \xi_2(k) + H^3 \phi^T \xi_3(k) + H^4 \phi^T \xi_4(k) \quad (3.14)$$

$$\begin{aligned} \begin{bmatrix} \hat{y}_1(k) \\ \hat{y}_2(k) \end{bmatrix} &= \begin{bmatrix} a_{1\varphi_1}^1 & a_{1\varphi_2}^1 & b_{1\varphi_1}^1 & b_{1\varphi_2}^1 \\ a_{2\varphi_1}^1 & a_{2\varphi_2}^1 & b_{2\varphi_1}^1 & b_{2\varphi_2}^1 \end{bmatrix} \begin{bmatrix} y_1(k-1) \\ y_2(k-1) \\ u_1(k-1) \\ u_2(k-1) \end{bmatrix} \xi_1(k) + \\ &\quad \begin{bmatrix} a_{1\varphi_1}^2 & a_{1\varphi_2}^2 & b_{1\varphi_1}^2 & b_{1\varphi_2}^2 \\ a_{2\varphi_1}^2 & a_{2\varphi_2}^2 & b_{2\varphi_1}^2 & b_{2\varphi_2}^2 \end{bmatrix} \begin{bmatrix} y_1(k-1) \\ y_2(k-1) \\ u_1(k-1) \\ u_2(k-1) \end{bmatrix} \xi_2(k) + \\ &\quad \begin{bmatrix} a_{1\varphi_1}^3 & a_{1\varphi_2}^3 & b_{1\varphi_1}^3 & b_{1\varphi_2}^3 \\ a_{2\varphi_1}^3 & a_{2\varphi_2}^3 & b_{2\varphi_1}^3 & b_{2\varphi_2}^3 \end{bmatrix} \begin{bmatrix} y_1(k-1) \\ y_2(k-1) \\ u_1(k-1) \\ u_2(k-1) \end{bmatrix} \xi_3(k) + \\ &\quad \begin{bmatrix} a_{1\varphi_1}^4 & a_{1\varphi_2}^4 & b_{1\varphi_1}^4 & b_{1\varphi_2}^4 \\ a_{2\varphi_1}^4 & a_{2\varphi_2}^4 & b_{2\varphi_1}^4 & b_{2\varphi_2}^4 \end{bmatrix} \begin{bmatrix} y_1(k-1) \\ y_2(k-1) \\ u_1(k-1) \\ u_2(k-1) \end{bmatrix} \xi_4(k) \end{aligned} \quad (3.15)$$

Portanto, a identificação de um sistema MIMO não linear no modelo proposto acima depende tanto da estimação dos parâmetros das matrizes H^R , quanto da estimação do desvio padrão das *Gaussianas* das funções de pertinência do fator $\xi_R(k)$.

O tópico a seguir mostra como as matrizes de parâmetros são estimadas utilizando o algoritmo dos Mínimos Quadrados.

3.2.2 Mínimos Quadrados

O estimador de Mínimos Quadrados foi formulado no século XVIII por Friedrich Gauss para prever a trajetória de planetas e cometas a partir das observações realizadas (Coelho; Coelho, 2004). A principal ideia do MQ é minimizar o quadrado da distância *Euclidiana* entre o valor observado e o valor estimado pelo modelo (Tangirala, 2015).

Neste trabalho, o MQ é usado para estimar os parâmetros das matrizes H^R do modelo *fuzzy* TSK multivariável, conforme pode ser visto na Equação 3.16.

$$\hat{H} = \bar{Y}\bar{X}^T(\bar{X}\bar{X}^T)^{-1} \quad (3.16)$$

O termo $\bar{Y} \in \mathbb{R}^{(n) \times (N_{id})}$, Equação 3.17, representa a matriz contendo as N_{id} amostras dos dados das medidas observadas de todas as saídas da planta MIMO que se deseja identificar, e $\bar{X} \in \mathbb{R}^{(S_R) \times ((n \cdot \tau_{dy} + m \cdot \tau_{du}) \cdot N_{id})}$ é a matriz de regressores que depende tanto do vetor regressor $\phi(k)$ (este é o mesmo vetor da Equação 3.11, porém, é utilizado de forma simplificada na matriz $\bar{X}$ apenas como função da k -ésima medida), quanto do fator $\xi_R(k)$ de cada regra, conforme a Equação 3.18.

$$\bar{Y} = \begin{bmatrix} y_1(1) & y_1(2) & \cdots & y_1(N_{id}) \\ y_2(1) & y_2(2) & \cdots & y_2(N_{id}) \\ \vdots & \vdots & \ddots & \vdots \\ y_n(1) & y_n(2) & \cdots & y_n(N_{id}) \end{bmatrix} \quad (3.17)$$

$$\bar{X} = \begin{bmatrix} \phi^T(1)\xi_1(1) & \phi^T(2)\xi_1(2) & \cdots & \phi^T(N_{id})\xi_1(N_{id}) \\ \phi^T(1)\xi_2(1) & \phi^T(2)\xi_2(2) & \cdots & \phi^T(N_{id})\xi_2(N_{id}) \\ \vdots & \vdots & \ddots & \vdots \\ \phi^T(1)\xi_R(1) & \phi^T(2)\xi_R(2) & \cdots & \phi^T(N_{id})\xi_R(N_{id}) \end{bmatrix} \quad (3.18)$$

A variável $\hat{H} \in \mathbb{R}^{(n) \times (n \cdot \tau_{du} + m \cdot \tau_{dy})}$ representa o conjunto de matrizes H^R estimadas pelo MQ, conforme mostrado na Equação 3.19. Esta matriz é utilizada no modelo *fuzzy* TSK do sistema MIMO identificado pelo *framework*, conforme também pode ser constatado na Figura 3.2.

$$\hat{H} = [H^1 \quad H^2 \quad \cdots \quad H^R] \quad (3.19)$$

Conforme pode ser visto neste tópico, os Mínimos Quadrados são utilizados na etapa de identificação com o propósito de estimar os parâmetros das matrizes do modelo *fuzzy* TSK multivariável. O próximo tópico irá abordar com mais detalhes o algoritmo

PSO, e como este é usado para estimar o desvio padrão de cada *Gaussiana* presente no modelo utilizado.

3.2.3 *Particle Swarm Optimization (PSO)*

Neste tópico são descritas as características gerais do algoritmo PSO utilizado na estimação do desvio padrão das funções de pertinência do modelo *fuzzy* TSK multivariável. Também será explicado algumas características específicas do PSO utilizadas como metodologia no *framework*.

Também é importante ressaltar que o PSO foi escolhido devido a sua facilidade de implementação e eficiência na resolução de problemas complexos e não lineares. Todavia, qualquer outra metaheurística ou algoritmo determinístico de otimização não linear pode ser utilizado nesta etapa do *framework*.

a) **Características Gerais do PSO**

A computação natural é um ramo de conhecimento aplicável em diversos campos de estudo como: matemática; estatística; ciência da computação; biologia; química etc. Esta área possui o aspecto de se inspirar nas características de sistemas naturais, como por exemplo o processo de sobrevivência. Estas características são abstraídas em forma de algoritmos e modelos matemáticos que são largamente utilizados na resolução de problemas de busca e otimização (Brabazon et al., 2015).

A inteligência coletiva emerge do comportamento cooperativo de um grupo social de organismos vivos em sua interação com seu ecossistema, realizando tarefas que individualmente são muito difíceis de serem executadas, mas que podem ser executadas com maior facilidade em grupo de indivíduos (Sun et al., 2012).

Inteligências de enxame foi uma expressão introduzida por Gerardo Beni e Jing Wang no contexto de sistemas robóticos celulares (Beni; Wang, 1993). Essa categoria de inteligência computacional demonstra o padrão de comportamento global de entidades em coletivo, as quais não são observados individualmente, pois o comportamento é guiado pelo compartilhamento do conhecimento entre os indivíduos do bando.

Uma das inteligências de enxame mais conhecidas é o algoritmo PSO, também traduzido como Otimização por Enxame de Partículas, o qual foi introduzido nos trabalhos de Eberhart e Kennedy em 1995, cujo fluxograma é mostrado na Figura 3.6.

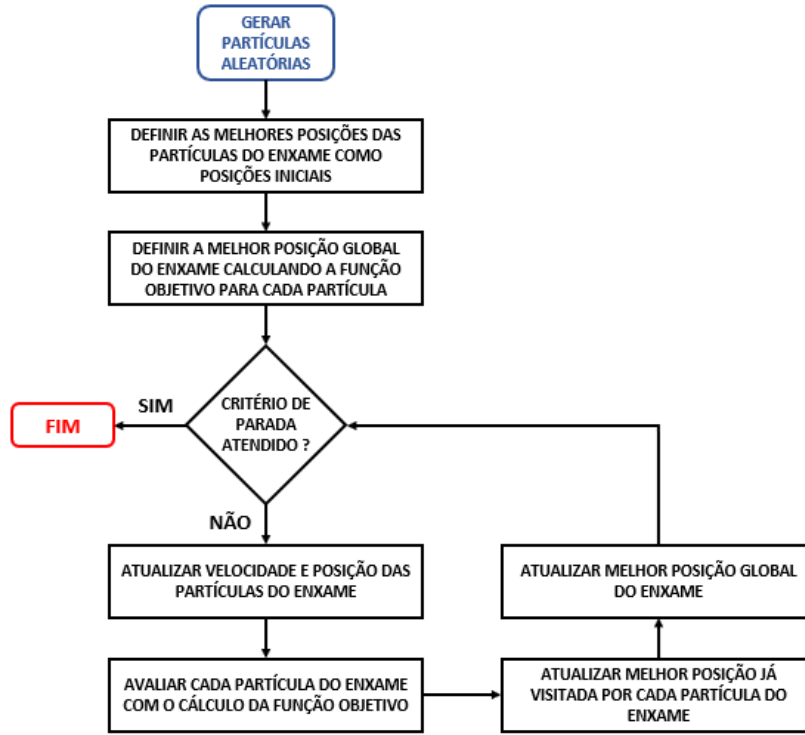


Figura 3.6 – Fluxograma do algoritmo PSO.

Este algoritmo de otimização trata-se de uma inspiração no comportamento de bandos de pássaros, os quais são representados por um conjunto de partículas, onde a posição de cada uma representa uma possível solução para um problema a ser resolvido, e cada partícula possui uma velocidade associada, ambas as variáveis são atualizadas com auxílio de informações compartilhadas pelo enxame (Kennedy; Eberhart, 1995).

No PSO, cada partícula do enxame é formada por um vetor posição ($\vec{x}_q(t)$), Equação 3.20, que representa uma solução possível para o problema; um vetor velocidade ($\vec{v}_q(t)$), Equação 3.21, que pondera com qual velocidade uma partícula se move no espaço de busca; um vetor de memória cognitiva da partícula ($\vec{P}_q(t)$), Equação 3.22, o qual indica a melhor posição individual já ocupada por determinada partícula.

$$\vec{x}_q(t) = [x_q^1(t) \quad x_q^2(t) \quad \dots \quad x_q^{N_{var}}(t)] \quad (3.20)$$

$$\vec{v}_q(t) = [v_q^1(t) \quad v_q^2(t) \quad \dots \quad v_q^{N_{var}}(t)] \quad (3.21)$$

$$\vec{P}_q(t) = [P_q^1(t) \quad P_q^2(t) \quad \dots \quad P_q^{N_{var}}(t)] \quad (3.22)$$

Onde q é a q -ésima partícula do enxame $\{q \in \mathbb{Z} / 1 \leq i \leq N_{part}\}$, N_{part} é a quantidade total de partículas inicializadas no enxame, N_{var} é o número total de parâmetros de cada partícula e t representa a t -ésima iteração do algoritmo.

Antes da execução do PSO, os seguintes parâmetros são definidos: quantidade total de partículas (N_{part}); dimensão das partículas (N_{var}); os limites do espaço de busca das partículas; a função objetivo que será utilizada para avaliar cada partícula; o número total de iterações do algoritmo (N_{total}), assim como qualquer outro critério de parada para o algoritmo.

A primeira etapa da execução do PSO consiste em gerar as partículas com dimensão pré-definidas. Os vetores posição e velocidade são inicializados com valores aleatórios e limitados respectivamente pelos intervalos $[X_{min}^a; X_{max}^a] \in \mathbb{R}^{N_{var}}$ e $[V_{min}^a; V_{max}^a] \in \mathbb{R}^{N_{var}}$, onde a é a a -ésima dimensão da partícula $\{a \in \mathbb{Z} / 1 \leq a \leq N_{var}\}$. Os valores de $\vec{P}_q(t)$ são inicializados com os mesmos valores iniciais gerados para os vetores posição. Geralmente, utiliza-se distribuição aleatória uniforme para os valores iniciais dos dois vetores citados.

Outra variável importante neste algoritmo é o vetor da melhor posição global ($\vec{G}(t)$), onde este vetor indica qual a partícula do enxame possui a melhor posição na t -ésima iteração, conforme pode ser visto na Equação 3.23.

$$\vec{G}(t) = [G^1(t) \quad G^2(t) \quad \dots \quad G^{N_{var}}(t)] \quad (3.23)$$

O vetor $\vec{G}(t)$ é definido em função dos vetores $\vec{P}_q(t)$, ou seja, a partícula com melhor memória cognitiva (partícula mais bem avaliada) na t -ésima se torna a melhor posição global. Entende-se por melhor global, aquela posição alcançada por uma partícula do enxame que atinge o melhor valor para a função objetivo (f_{obj}) em uma determinada iteração do algoritmo.

A avaliação de uma partícula ocorre por meio da função objetivo, a qual é utilizada tanto para representar o modelo matemático do problema a ser resolvido, assim como também é utilizada para avaliar o quão bom cada partícula é na resolução de um dado problema na t -ésima iteração.

Após a inicialização do enxame, a avaliação de cada partícula e a determinação da melhor posição global do enxame, o critério de parada do algoritmo é avaliado, se o critério for atendido então o PSO finaliza sua execução e a resposta encontrada para o problema é dada pelo vetor $\vec{G}(t)$, caso contrário, o algoritmo segue para as etapas de atualização dos vetores posição e velocidade de cada partícula conforme as Equações 3.24 e 3.25 respectivamente.

$$\vec{v}_q(t+1) = \bar{w}\vec{v}_q(t) + c_1 r_1(t)[\vec{P}_q(t) - \vec{x}_q(t)] + c_2 r_2(t)[\vec{G}(t) - \vec{x}_q(t)] \quad (3.24)$$

$$\vec{x}_q(t+1) = \vec{x}_q(t) + \vec{v}_q(t+1) \quad (3.25)$$

Os coeficientes c_1 e c_2 são chamados de aceleração das partículas, mas também são conhecidos como fatores cognitivo e social respectivamente. O parâmetro $\bar{w}$ é conhecido como fator de inércia, o qual só foi acrescentado ao PSO canônico depois dos trabalhos de Shi e Eberhart no artigo *A Modified Particle Swarm Optimization* de 1998 (Shi; Eberhart, 1998). Os coeficientes $r_1(t)$ e $r_2(t)$ são parâmetros inicializados aleatoriamente com distribuição uniforme entre 0 e 1 para cada partícula, onde em cada iteração são gerados novos valores aleatórios.

Após a etapa de atualização das posições e velocidades das partículas, avalia-se estes dois vetores, de todas as partículas do enxame, com o intuito de garantir se estão dentro de seus respectivos limites de busca para todos os parâmetros. Caso o limite seja extrapolado, deve-se fazer as correções conforme as Equações 3.26 e 3.27.

$$\vec{x}_q(t+1) = \begin{cases} X_{max}^a; & \text{se } \vec{x}_q(t+1) > X_{max}^a \\ X_{min}^a; & \text{se } \vec{x}_q(t+1) < X_{min}^a \end{cases} \quad (3.26)$$

$$\vec{v}_q(t+1) = \begin{cases} V_{max}^a; & \text{se } \vec{v}_q(t+1) > V_{max}^a \\ V_{min}^a; & \text{se } \vec{v}_q(t+1) < V_{min}^a \end{cases} \quad (3.27)$$

Em seguida, o valor da função objetivo de cada partícula atualizada é calculado, e assim, o vetor de memória cognitiva de cada uma também é atualizado utilizando-se uma comparação entre o valor de $f_{obj}(\vec{x}_q(t+1))$ e o de $f_{obj}(\vec{P}_q(t))$, conforme a Equação 3.28. Esta atualização depende se a busca é pelo mínimo ou pelo máximo da função objetivo, devendo apenas inverter os sinais $>$ e $\leq$ se necessário for.

$$\vec{P}_q(t+1) = \begin{cases} \vec{x}_q(t+1); & \text{se } f_{obj}(\vec{x}_q(t+1)) > f_{obj}(\vec{P}_q(t)) \\ \vec{P}_q(t); & \text{se } f_{obj}(\vec{x}_q(t+1)) \leq f_{obj}(\vec{P}_q(t)) \end{cases} \quad (3.28)$$

Posteriormente a atualização das memórias cognitivas, avalia-se qual dentre todas estas possui o melhor valor para a função objetivo (menor valor para problema de minimização, e maior valor para problema de maximização), e a partícula com melhor valor é declarada como a melhor posição global atualizada $\vec{G}(t+1)$, conforme a Equação

3.29. Esta posição é considerada a resposta encontrada pelo PSO na t -ésima iteração para a resolução do problema abordado.

$$\vec{G}(t+1) = \vec{P}_\psi(t+1) = \begin{cases} \psi = \min(f_{obj}(\vec{P}(t+1))); & \text{se Minimização} \\ \psi = \max(f_{obj}(P(t+1))); & \text{se Maximização} \end{cases} \quad (3.29)$$

Após atualizar a melhor posição global do enxame, o PSO testa o seu critério de parada com o valor da função objetivo de $\vec{G}(t+1)$. Se o critério for alcançado então o algoritmo para, se não, os procedimentos desde a Equação 3.23 até a 3.28 são realizados novamente em *loop* até algum critério de parada ser alcançado.

b) Características do PSO Aplicado à Identificação de Sistemas MIMO

As etapas descritas acima, com base no fluxograma da Figura 3.6 e nas equações mostradas, são características gerais para o PSO canônico. Entretanto, a função objetivo, a estrutura e o espaço de busca das partículas utilizados neste algoritmo na etapa de identificação do *framework* são atributos importantes e que serão mais detalhados a seguir.

- **Função Objetivo:** A função objetivo (f_{obj}), conforme mencionado, é uma equação que representa matematicamente o problema a ser tratado pelo PSO, cujo valor retornado é usado para ponderar a eficiência de cada partícula em cada iteração.

Para a etapa de identificação de sistemas MIMO no *framework* é proposto como função objetivo a média do Erro Quadrático Médio ($mEQM_i$) do sinal observado de cada n -ésima saída da planta ($y_n(k)$), e seu respectivo sinal estimado ($\hat{y}_n(k)$), utilizando-se os dados de identificação, conforme a Equação 3.30, a qual coincide com a Equação 3.4.

$$f_{obj} = \frac{1}{n} \sum_{j=1}^n \left(\sum_{k=1}^{N_{id}} \frac{[y_j(k) - \hat{y}_j(k)]^2}{N_{id}} \right) \quad (3.30)$$

A Figura 3.7 abaixo ilustra o fluxograma de como a função objetivo do PSO utiliza informações do vetor de erro dos sinais de saída da planta original e do modelo estimado ($\vec{e}(k) \in \mathbb{R}$), para realizar a busca do desvio padrão de cada *Gaussiana*. No *framework*, o PSO busca minimizar o valor de f_{obj} de cada q -ésima partícula em cada t -ésima iteração, para que assim os valores ótimos ou sub ótimos dos desvios padrões sejam alcançados.

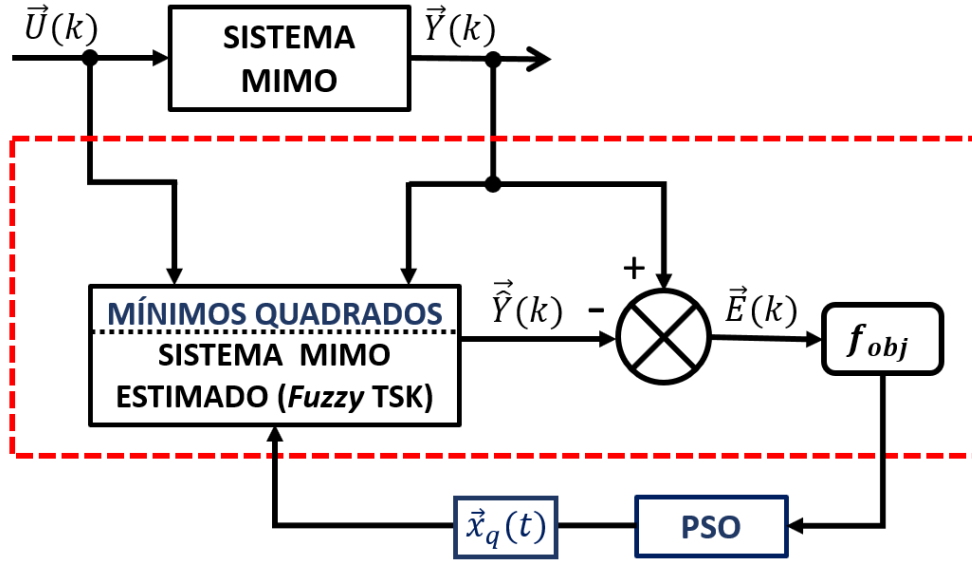


Figura 3.7 – Fluxograma da função objetivo do PSO.

• **Estrutura e Espaço de Busca das Partículas:** O vetor posição e seu espaço de busca no PSO usado na etapa de identificação possuem características peculiares ao problema, como por exemplo a estrutura de $\vec{x}_q(t)$, conforme pode ser visto na Equação 3.31.

$$\vec{x}_q(t) = [\sigma_{11} \quad \sigma_{21} \quad \cdots \quad \sigma_{1m} \quad \sigma_{2m}] \quad (3.31)$$

Cada parâmetro vetor posição corresponde ao desvio padrão de cada *Gaussiana* usada como função de pertinência em cada m -ésima entrada do modelo *fuzzy TSK* multivariável, e cada entrada possui duas funções de pertinência, conforme estabelecido para o *framework*, portanto, o vetor posição possui dimensão $\mathbb{R}^{2m}$.

A melhor posição global alcançada pelo PSO quando este atinge algum critério de parada ($\vec{G}(final)$), mostra o vetor $\vec{\sigma}_G \in \mathbb{R}^{N_{var}}$ dos valores ótimos de desvio padrão das *Gaussianas* do modelo *fuzzy TSK* multivariável da planta identificada, conforme a Equação 3.32.

$$\vec{\sigma}_G = \vec{G}(final) \quad (3.32)$$

A Figura 3.2 do tópico 3.2 mostra como este vetor é utilizado na fase de identificação, embora seja importante ressaltar que ele é o resultado final alcançado pelo PSO. Os valores dos desvios padrões do vetor $\vec{\sigma}_G$ e da matriz $\hat{H}$ também são utilizados no modelo *fuzzy TSK* multivariável na fase de validação, e na etapa de teste do *framework* caso estes sejam relacionados ao modelo selecionado.

O espaço de busca da posição das partículas está compreendido no intervalo $[0; 1]$, Equação 3.33, ou seja, $X_{min}^a = 0$ e $X_{max}^a = 1$, uma vez que este são os limites de excursão dos sinais de entrada do sistema *fuzzy*. Os limites de $\vec{P}_q(t)$ coincidem com o do vetor posição, e os do vetor velocidade são definidos em um intervalo menor que do vetor posição.

$$0 \leq \vec{x}_q(t) \leq 1 \quad (3.33)$$

3.3 Conclusão

Neste capítulo foram apresentadas todas as características do *framework* proposto e como este é utilizado no processo de identificação de sistemas MIMO não lineares. Também foram explicados com detalhes os algoritmos Mínimos Quadrados e PSO, além do modelo *fuzzy* TSK multivariável.

No próximo capítulo serão mostrados os resultados da aplicação do *framework* na identificação de duas plantas multivariadas, cujos resultados serão comparados aos alcançados por uma RNA e com o modelo de Hammerstein-Wiener.

4 RESULTADOS

Neste capítulo são mostrados os resultados da aplicação do *framework* proposto, uma Rede Neural Artificial com arquitetura MLP, e o modelo de Hammerstein-Wiener para a identificação *off-line* e caixa-preta (ausência de conhecimentos detalhados da dinâmica e do modelo matemático do sistema) de duas plantas MIMO não lineares: Reator Contínuo de Tanque Agitado (CSTR); Secador Industrial (*Industrial Dryer*). As amostras de dados utilizados nas duas plantas foram retiradas da base de dados DaISy (*Database for Identification Systems*).

Para análise dos resultados, é importante destacar que não houve interferência externa para condicionamento dos sinais de entrada com ruído branco ou sinal do tipo PRBS (do inglês, *Pseudo Random Binary Sequence*) para que fosse feita a identificação, ou mesmo feita pré-filtragem dos sinais observados e excitação das plantas para que atingissem outros pontos de operação.

Os ajustes dos parâmetros do PSO no *framework*, da RNA e do modelo HW foram feitos heurísticamente com base em testes preliminares, e a melhor configuração de cada técnica para cada estudo de caso foi utilizada nos resultados. Todas as simulações deste trabalho foram feitas no *software* MATLAB como ambiente de programação em um computador Intel(R) Core(TM) i5 (4 GB de memória RAM).

A etapa de identificação do *framework* foi executada 30 vezes ($N_{sim} = 30$), e o modelo *fuzzy* TSK multivariável utilizado nas etapas de validação e teste é constituído pela média da matriz de parâmetros H^R e do vetor de desvio padrão $\vec{\sigma}_G$ estimados dentre todas as simulações.

A avaliação de desempenho das três técnicas de identificação nos resultados é feita tanto por meio de métricas de *EQM* e *VAF*, divididas em Métrica de Validação e Métricas de Teste, quanto pela análise do sinal de resíduo (erro absoluto) nos sinais de saída das plantas identificadas.

- **Métrica de Validação**

A métrica de validação, conforme previamente explicado no tópico 3.1 deste trabalho, é aplicada apenas ao *framework* em sua fase de validação com o objetivo de mensurar qual vetor regressor gerou menor erro de estimação, o qual será usado na etapa seguinte de teste. O índice de desempenho utilizado como métrica de validação é o *EQM*,

e o $mEQM_v$, mostrados respectivamente nas Equações 3.2 e 3.3. O limite máximo do $mEQM_v$ definido para o *framework* na identificação das duas plantas é $1,0 \times 10^{-3}$.

- **Métricas de Teste**

O desempenho do *framework* é comparado com a RNA e o modelo HW utilizando os N_t dados de teste por meio dos índices EQM_t e VAF , Equações 4.1 e 4.2 respectivamente. Onde $\vec{Y}_j$ e $\vec{\hat{Y}}_j$ são o vetor de medidas observadas e estimadas em cada j -ésima saída da planta respectivamente. O índice VAF indica o quão um sinal se aproxima do outro em termos percentuais, ou seja, quanto mais esse indicador se aproxima de 100%, maior é a próximo o sinal estimado é do sinal real (Millan and Barrios, 2016).

$$EQM_t = \sum_{k=N_{id}+N_v+1}^{N_{id}+N_v+N_t} \frac{[y_j(k) - \hat{y}_j(k)]^2}{N_t} \quad (4.1)$$

$$VAF = \left(1 - \frac{var(\vec{Y}_j - \vec{\hat{Y}}_j)}{var(\vec{Y}_j)} \right) \times 100\% \quad (4.2)$$

A seguir serão descritos os dois estudos de caso abordados, assim como o resultado alcançado em cada um pela aplicação das técnicas de identificação citadas.

4.1 Reator Contínuo de Tanque Agitado (*Continuous Stirred Tank Reactor*)

O Reator Contínuo de Tanque Agitado (CSTR), é um mecanismo utilizado para realizar a mistura controlada de reagentes químicos principalmente em fase líquida, para se obter no final um produto desejável (Fogler, 2009).

No misturador, um ou mais reagentes fluidos são introduzidos dentro de um reator equipado com pás rotativas, as quais ficam rotacionando para realizar a mistura dos reagentes enquanto o produto final flui por outro compartimento (Ahmed et al., 2016). A Figura 4.1 ilustra um esquemático de um CSTR usado como planta multivariada a ser identificada.

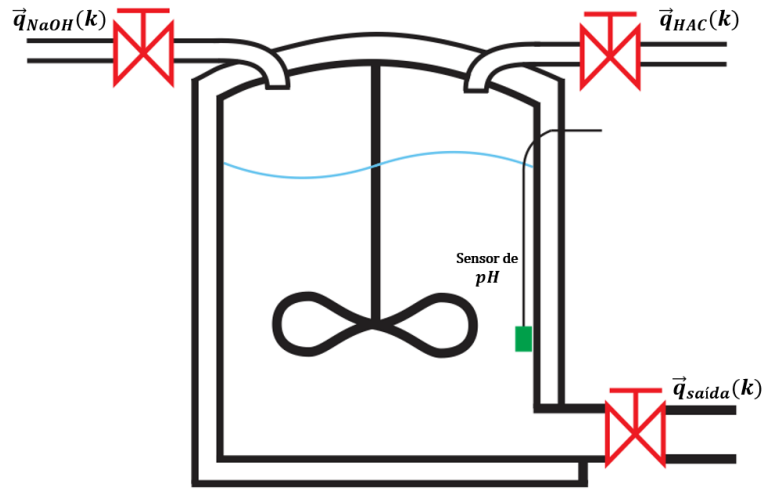
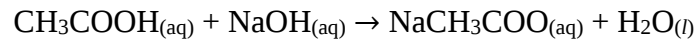


Figura 4.1 – Diagrama do CSTR.

Para este estudo de caso, foram utilizados dados reais de um processo de mistura do hidróxido de sódio (NaOH) com ácido acético (CH₃COOH) o qual é representado aqui pela sigla (HAC). A reação de neutralização dos reagentes é mostrada abaixo.



As variáveis de entrada do reator são $\vec{q}_{\text{NaOH}}(k)$ e $\vec{q}_{\text{HAC}}(k)$, as quais representam a vazão da base e do ácido respectivamente em litros. E as variáveis de saída são a vazão da mistura ($\vec{q}_{\text{saída}}(k)$) em litros, e o potencial hidrogeniônico da solução ($\overrightarrow{pH}(k)$). O volume de solução do processo é considerado constante a 1100 litros e a concentração de HAC e NaOH são respectivamente 0,0032 mol/L e 0,05 mol/L. Portanto, este processo pode ser considerado um sistema MIMO 2x2 (duas entradas e duas saídas).

A dinâmica da concentração do cálculo da vazão de saída do CSTR é abordada nos trabalhos de Ahmed et al. (2016) e Antonelli e Astolfi (2003), e a dinâmica da estimação do pH nesses tipos de sistemas é abordado no trabalho de McAvoy, Hsu, e Lowenthal, (1972). Todavia, algumas informações como a constante de equilíbrio do ácido acético e da água, assim como as constantes de velocidade da reação direta e inversa, as quais são necessárias para a modelagem matemática das saídas do sistema, não são conhecidas para este estudo de caso.

As Figuras 4.2 e 4.3 abaixo mostram respectivamente as amostras totais dos dados de entrada e saída coletadas da planta do tanque misturador em 308,5 minutos de operação, cuja taxa de amostragem é $T_s = 10$ s.

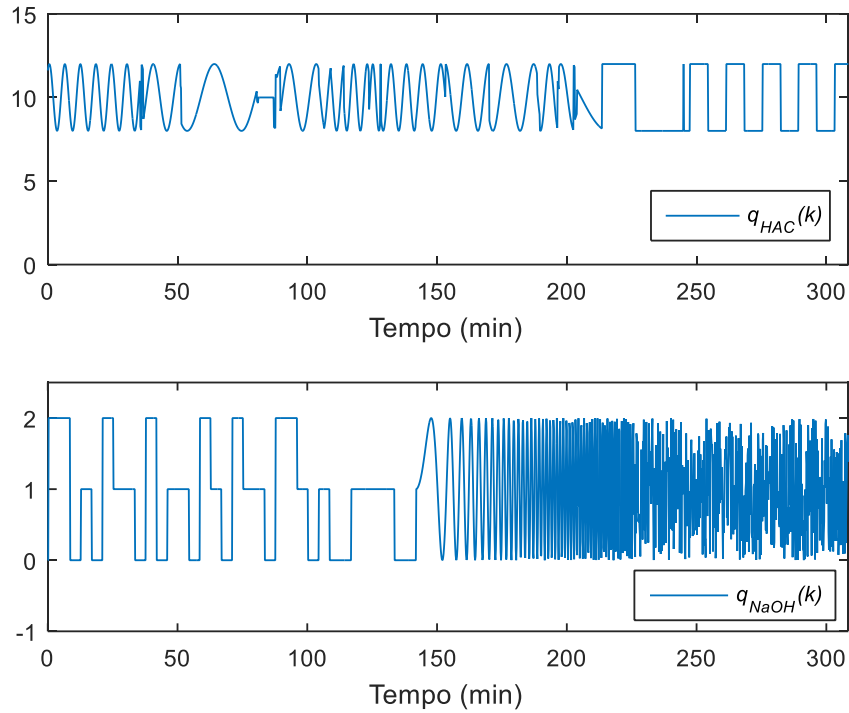


Figura 4.2 – Dados de entrada das variáveis $\vec{q}_{HAC}(k)$ e $\vec{q}_{NaOH}(k)$.

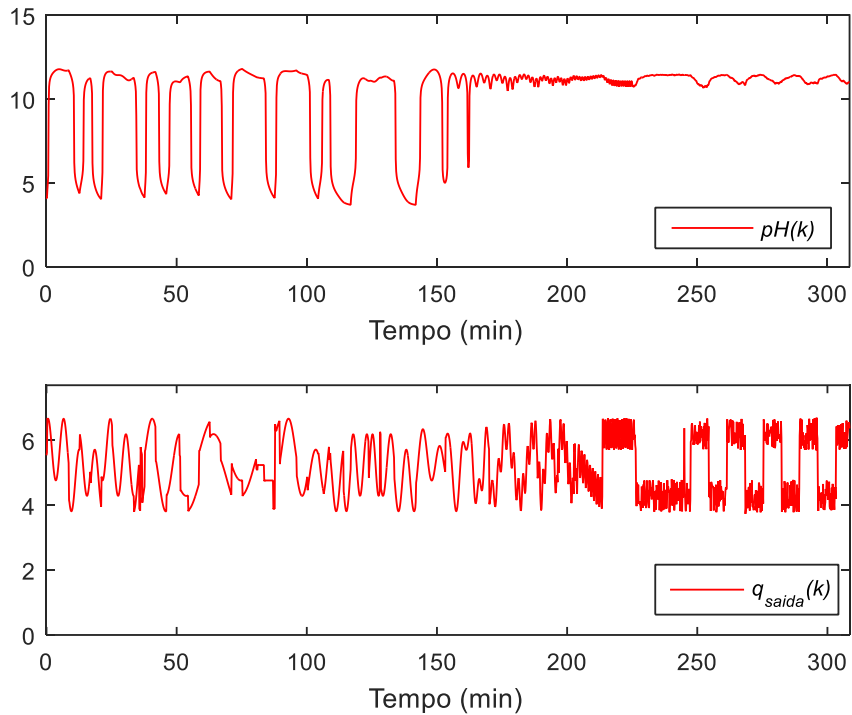


Figura 4.3 – Dados de saída das variáveis $\vec{pH}(k)$ e $\vec{q}_{saida}(k)$.

Primeiramente, deve-se levar em consideração na análise dos resultados que os dados utilizados para a etapa de identificação (treinamento no caso da RNA) mostram que a planta do CSTR possui uma dinâmica nos dados de identificação bem diferente da dinâmica dos dados de teste, o que torna este problema ainda mais complexo do que se

fosse apenas estimar uma planta multivariada não linear com uma dinâmica de operação com pouca variação.

A Tabela 4.1 mostra a divisão percentual e temporal dos dados do reator utilizados em cada técnica, com base na ordem temporal progressiva de execução da planta. A divisão percentual das amostras foi baseada na metodologia de Silva et al. (2018).

Tabela 4.1 – Divisão percentual e temporal dos dados para o 1º estudo de caso.

	Identificação/Treino 0 – 215,83 (min)	Validação 216 – 262,16 (min)	Teste 262,33 – 308,5 (min)
Framework	70%	15%	15%
RNA	70%	15%	15%
HW	70%	-	15%

As Tabelas 4.2 a 4.4 mostram respectivamente parametrização do PSO usado no *framework* conforme proposto em Oliveira et al. (2019), os modelos de vetor regressor utilizados nessa planta MIMO 2x2 e as regras do sistema *fuzzy* TSK multivariável.

Tabela 4.2 – Parâmetros do PSO do *framework* no 1º estudo de caso.

N_{part}	N_{var}	$\bar{w}$	c_1	c_2	X_{min}	X_{max}	V_{min}	V_{max}	N_{itotal}	$mEQM_i$
25	4,0	0,8	2,0	2,0	0,0	1,0	-0,5	0,5	100	10^{-3}

Tabela 4.3 – Modelos de regressores do *framework* no 1º estudo de caso.

	Regressor
Regressor 1 $\tau_{dy} = 1 \ \tau_{du} = 1$	$[y_1(k-1) \ y_2(k-1) \ u_1(k-1) \ u_2(k-1)]$
Regressor 2 $\tau_{dy} = 2 \ \tau_{du} = 2$	$[y_1(k-1) \ y_1(k-2) \ y_2(k-1) \ y_2(k-2) \ \dots$ $u_1(k-1) \ u_1(k-2) \ u_2(k-1) \ u_2(k-2)]$
Regressor 3 $\tau_{dy} = 3 \ \tau_{du} = 3$	$[y_1(k-1) \ y_1(k-2) \ y_1(k-3) \ \dots$ $y_2(k-1) \ y_2(k-2) \ y_2(k-3) \ \dots$ $u_1(k-1) \ u_1(k-2) \ u_1(k-3) \ \dots$ $u_2(k-1) \ u_2(k-2) \ u_2(k-3)]$

Tabela 4.4 – Regras do modelo *fuzzy* TSK no 1º estudo de caso.

Regras	Preposição SE - ENTÃO
$R = 1$	SE $\tilde{u}_1(k)$ é $A_1^{\rho_1}$ e $\tilde{u}_2(k)$ é $A_2^{\rho_1}$ ENTÃO $\hat{Y}(k) = d^1 \xi_1(k)$
$R = 2$	SE $\tilde{u}_1(k)$ é $A_1^{\rho_1}$ e $\tilde{u}_2(k)$ é $A_2^{\rho_2}$ ENTÃO $\hat{Y}(k) = d^2 \xi_2(k)$
$R = 3$	SE $\tilde{u}_1(k)$ é $A_1^{\rho_2}$ e $\tilde{u}_2(k)$ é $A_2^{\rho_1}$ ENTÃO $\hat{Y}(k) = d^3 \xi_3(k)$
$R = 4$	SE $\tilde{u}_1(k)$ é $A_1^{\rho_2}$ e $\tilde{u}_2(k)$ é $A_2^{\rho_2}$ ENTÃO $\hat{Y}(k) = d^4 \xi_4(k)$

As Tabelas 4.5 e 4.6 mostram as configurações paramétricas da RNA e do modelo HW respectivamente. Ambas as configurações foram ajustadas de maneira heurística.

Tabela 4.5 – Parâmetros da RNA no 1º estudo de caso.

Função da Camada Escondida	2 Camadas Escondidas Tangente Sigmoid (20 Neurônios) Tangente Sigmoid (20 Neurônios)
Função da Camada de Saída	Linear (2 Neurônios)
Algoritmo de Treinamento	Levenberg-Marquardt
Entradas da RNA	$[y_1(k-1) \ y_2(k-1) \ u_1(k-1) \ u_2(k-1)]$
Saídas da RNA	$[y_1(k) \ y_2(k)]$

Tabela 4.6 – Parâmetros do modelo Hammerstein-Wiener no 1º estudo de caso.

Parte Não Linear (Estática)	Funções Estáticas da Entrada	Função Network Sigmoid (30 parâmetros)
	Funções Estáticas da Saída	Função Network Sigmoid (30 parâmetros)
	Algoritmo de Estimação	Levenberg-Marquardt
	Número Máximo de Iterações	30
Parte Linear (Dinâmica)	Funções de Transferência (FT) da Saída 1	Número de FT: 2 Estrutura por FT: 2 polos; 2 zeros; 2 atrasos
	Funções de Transferência (FT) da Saída 2	Número de FT: 2 Estrutura por FT: 2 polos; 2 zeros; 2 atrasos
	Algoritmo de Estimação	Mínimos Quadrados

Nas tabelas e figuras acima, as variáveis $\vec{q}_{HAC}$, $\vec{q}_{NaOH}$, pH e $\vec{q}_{saída}$ da planta do reator foram substituídas respectivamente por u_1 , u_2 , y_1 , y_2 para efeitos de simplificação. A Tabela 4.7 mostra o desempenho em termos do EQM_v alcançado pelo *framework* na etapa de validação em relação a cada saída isolada e da média entre elas ($mEQM_v$), utilizando os três modelos de regressores.

Tabela 4.7 – Resultados do EQM_v e do $mEQM_v$ para os 3 modelos de regressores utilizados no *framework* na etapa de identificação no 1º estudo de caso.

<i>Framework</i>	Regressor 1	Regressor 2	Regressor 3
$EQM_v(y_1)$	0,0425	0,0444	0,2501
$EQM_v(y_2)$	0,1665	0,2422	0,3136
$mEQM_v(y_1, y_2)$	0,1045	0,1433	0,2832

Na etapa de validação, o modelo 1 de regressor alcançou o melhor desempenho neste estudo de caso, em termos de EQM_v por saída e na média entre as saídas. Nesta etapa, pode-se observar uma diferença razoável entre o erro de estimação entre o regressor 1 e o 3, o que pode ser explicado pelo erro gerado na aplicação dos Mínimos Quadrados com os dados de entrada gerados por cada um destes vetores.

Uma possível explicação para as diferenças encontradas no erro de estimação com cada vetor regressor recai no impacto direto que cada um gera sobre o cálculo dos parâmetros de cada matriz de parâmetros H^R , uma vez que quanto menor o tamanho do regressor, menor será a dimensão da matriz $\bar{X}$ que é utilizado nos MQ, o que “garante” que a operação $(\bar{X}\bar{X}^T)^{-1}$ não gere números com valores muito elevados, o que ocasiona erros demasiados na estimação paramétrica da parte linear do modelo. Ou seja, devido a menor dimensionalidade gerada pelo regressor 1, menor foi o erro de estimação das matrizes do modelo *fuzzy* TSK multivariável para este estudo de caso.

O modelo *fuzzy* com o regressor 1, o qual alcançou o melhor desempenho na etapa de validação, é utilizado na fase de teste para avaliação de desempenho do *framework* com as demais técnicas. As Figuras 4.4 e 4.5 mostram respectivamente os resultados, com os dados de teste, das saídas y_1 (pH) e y_2 ($\vec{q}_{saída}$) da planta do CSTR identificada.

Conforme observado nas Figuras 4.4 e 4.5 que, para $k = 1$, a saída estimada pelo *framework* é sempre igual a zero, devido aos regressores utilizados possuírem valor diferente de zero apenas para $k \geq 2$. Enquanto que para as demais técnicas o valor inicialização dos seus respectivos algoritmos de identificação.

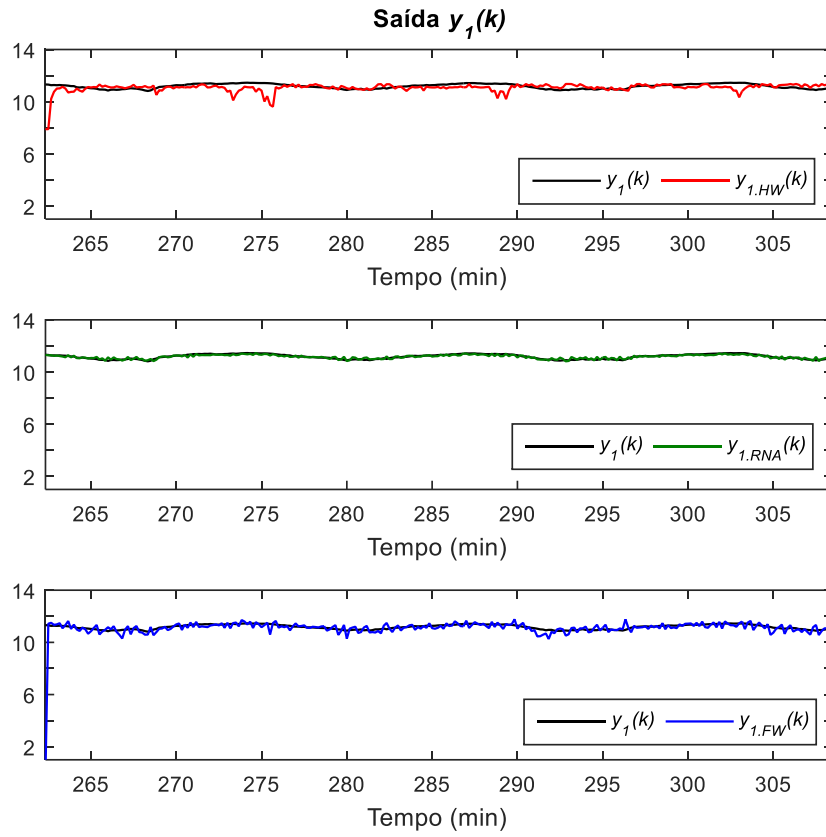


Figura 4.4 – Saída y_1 (pH) do CSTR observada (preto), e estimada pelo HW (vermelho), pela RNA (verde) e pelo *framework* (azul).

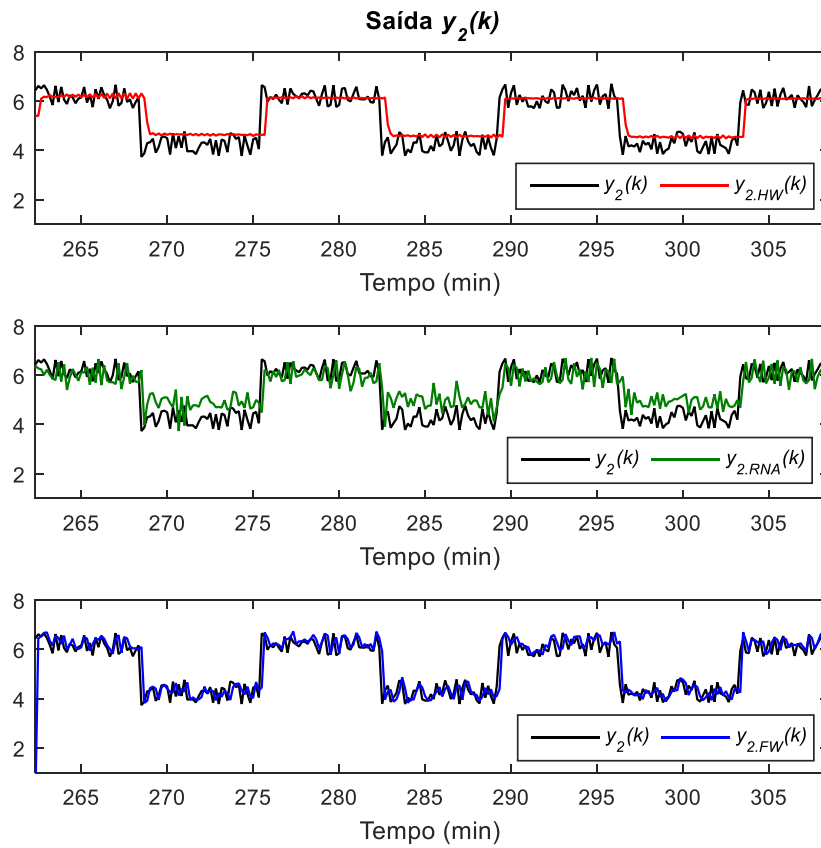


Figura 4.5 – Saída y_2 ($q_{saída}$) do CSTR observada (preto), e estimada pelo HW (vermelho), pela RNA (verde) e pelo *framework* (azul).

A Tabela 4.8 exibe os resultados do EQM_t e do VAF alcançados pelas técnicas de identificação utilizando os dados de teste da planta do reator.

Tabela 4.8 – Resultados do EQM_t e do VAF para os dados de teste no 1º estudo de caso.

Critério	Técnica		
	<i>Framework</i> (modelo 1)	RNA	HW
$EQM_t(y_1)$	0,0522	0,0081	0,1626
$VAF(y_1)$	-54,86 %	75,08 %	-361,32 %
$EQM_t(y_2)$	0,1807	0,4750	0,3111
$VAF(y_2)$	82,58 %	57,01 %	71,03 %

Os resultados acima mostram que o *framework* alcançou o segundo melhor resultado na estimação da saída y_1 . Isso demonstra que a RNA treinada foi capaz representar a dinâmica da planta de maneira mais eficiente, mesmo com pontos de operação distintos entre dados de identificação e teste, o que pode ser comprovado tanto pelo seu baixo valor de EQM_t , assim como pela maior taxa de VAF . Entretanto, o *framework* ainda foi capaz de estimar a saída y_1 com baixo erro e com desvio aceitável em relação ao sinal original, o que não compromete tanto a estimação do pH da solução no tanque.

Em relação a saída y_2 , o *framework* alcançou o melhor desempenho em relação aos dois índices, o que demonstra que o mesmo foi capaz de estimar a vazão de saída do tanque com uma acurácia maior que as outras duas técnicas, o que pode ser constatado na Figura 4.5, onde o modelo HW mostrou um resultado linearizado da estimação, e a RNA obteve erro significativo nos pontos de vazão de saída próximos a 4 litros.

As Figuras 4.6 e 4.7 mostram respectivamente o sinal de erro absoluto (resíduos) das saídas y_1 e y_2 em relação para cada uma das três técnicas de identificação. Uma análise importante que pode ser inferida é que a RNA teve o sinal de erro absoluto (gráfico verde) com menor variância, o que teve reflexo no seu valor de EQM_t , enquanto o *framework* proposto teve o sinal de erro (gráfico azul) com a segunda menor variância para esta saída.

Por outro lado, a Figura 4.7 mostra que a variância do erro absoluto do *framework* para a estimação da vazão do tanque foi a menor entre as técnicas. Também se pode observar que para o HW e para a RNA houveram pontos de operação desta planta em que a média do sinal de erro está um pouco distante de zero, como nos intervalos aproximados de 275 a 283 minutos e de 289 a 296 minutos. Este desvio da média em relação ao zero

reflete no próprio sinal de estimação desta saída por estas duas técnicas, pois exatamente nestes intervalos houve um erro maior de estimação.

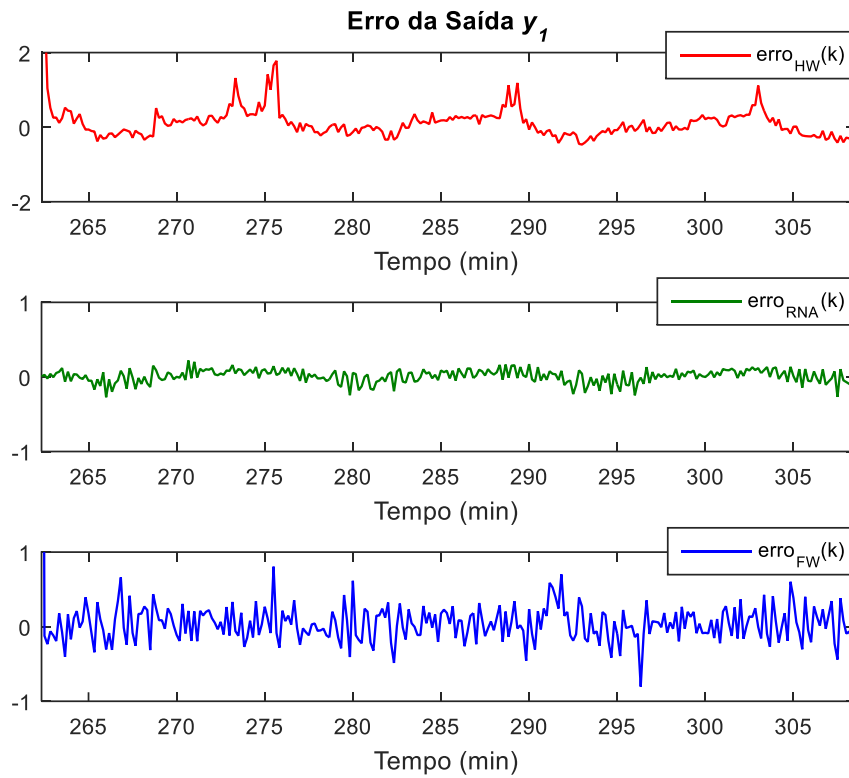


Figura 4.6 – Erro da saída y_1 do CSTR com a saída estimada pelo HW (vermelho), pela RNA (verde) e pelo *framework* (azul).

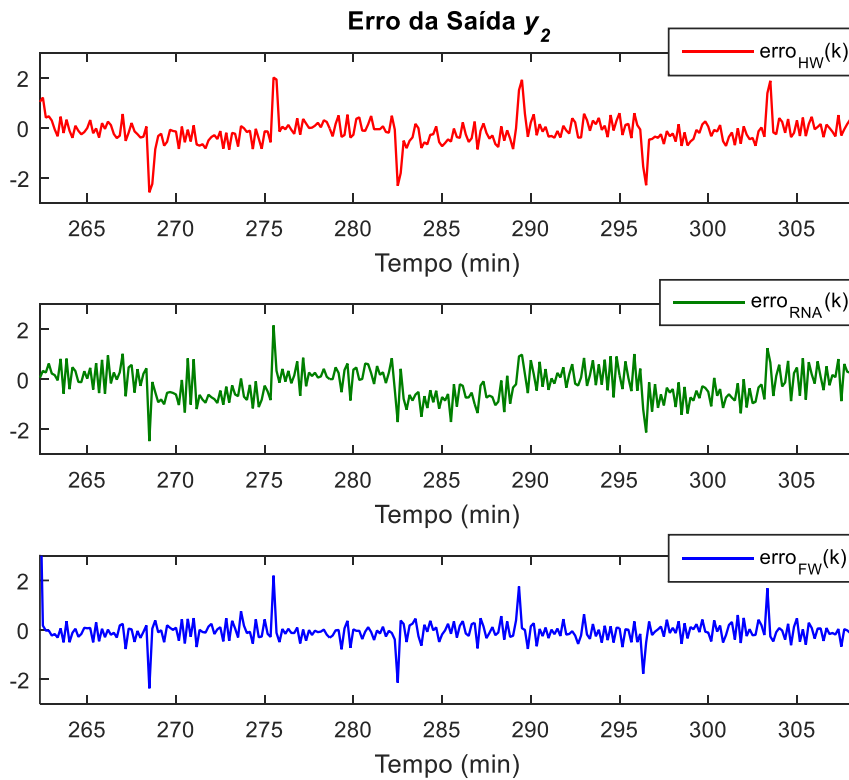


Figura 4.7 – Erro da saída y_2 do CSTR com a saída estimada pelo HW (vermelho), pela RNA (verde) e pelo *framework* (azul).

Em uma visão geral dos resultados, pode-se inferir algumas questões da estimação do pH , o qual possui uma dinâmica variante no tempo. Uma vez que os dados de treino possuem informações de maior variação entre uma mistura básica ou ácida, o *framework*, com o regressor 1 não conseguiu agregar informações suficientes para representar a planta com uma dinâmica de pH mais estável, conforme o é mostrado nos dados de teste. Por outro lado, a RNA com a arquitetura proposta foi capaz de identificar e simular este sistema em pontos distintos de operação.

Em relação a vazão de saída da mistura dos reagentes, por se tratar de uma variável que depende diretamente das vazões de entrada em um processo isovolumétrico, não há grandes variações da dinâmica desta variável de saída. Logo, o modelo *fuzzy* TSK estimado pelo *framework* foi capaz de estimar esta saída utilizando um regressor de pequena dimensão, o que denota a alta correlação da saída com as duas entradas com baixo grau de não linearidade.

O modelo *fuzzy* TSK multivariável 2x2 do CSTR identificado pelo *framework*, utilizando o modelo 1 de vetor regressor e a média da matriz de parâmetros estimada pelo MQ, é mostrada nas Equações 4.3 a 4.5. A Tabela 4.9 mostra a média dos desvios padrões estimados pelo PSO. As estruturas da RNA e do modelo HW, além do gráfico da evolução média da função do PSO neste estudo de caso são mostradas no apêndice I.

$$\begin{bmatrix} \hat{y}_1(k) \\ \hat{y}_2(k) \end{bmatrix} = \begin{bmatrix} 1,0231 & 0,5232 & -0,2774 & -0,1605 \\ 0,0305 & 0,5153 & 0,1909 & 0,0396 \end{bmatrix} \phi^T \xi_1(k) + \begin{bmatrix} 0,4210 & 1,0881 & 0,2708 & 0,0663 \\ -0,064 & 1,6207 & -0,1494 & -0,2195 \end{bmatrix} \phi^T \xi_2(k) + \begin{bmatrix} 0,8891 & 1,1308 & -0,520 & 0,2840 \\ -0,0182 & 1,0057 & 0,0225 & -0,0372 \end{bmatrix} \phi^T \xi_3(k) + \begin{bmatrix} 0,7042 & -1,8605 & 1,1225 & -0,3168 \\ 0,1380 & -1,9888 & 1,4547 & 0,3074 \end{bmatrix} \phi^T \xi_4(k) \quad (4.3)$$

$$\phi = [y_1(k-1) \quad y_2(k-1) \quad u_1(k-1) \quad u_2(k-1)]$$

$$\xi_1(k) = \frac{\mu_1^{\rho_1}(\tilde{u}_1(k))\mu_2^{\rho_1}(\tilde{u}_2(k))}{\sum_{R=1}^4 \mu_R(\tilde{u}_1(k), \tilde{u}_2(k))} \quad \xi_2(k) = \frac{\mu_1^{\rho_1}(\tilde{u}_1(k))\mu_2^{\rho_2}(\tilde{u}_2(k))}{\sum_{R=1}^4 \mu_R(\tilde{u}_1(k), \tilde{u}_2(k))} \quad (4.4)$$

$$\xi_3(k) = \frac{\mu_1^{\rho_2}(\tilde{u}_1(k))\mu_2^{\rho_1}(\tilde{u}_2(k))}{\sum_{R=1}^4 \mu_R(\tilde{u}_1(k), \tilde{u}_2(k))} \quad \xi_4(k) = \frac{\mu_1^{\rho_2}(\tilde{u}_1(k))\mu_2^{\rho_2}(\tilde{u}_2(k))}{\sum_{R=1}^4 \mu_R(\tilde{u}_1(k), \tilde{u}_2(k))}$$

$$\mu_1^{\rho_1}(\tilde{u}_1(k)) = e^{-\frac{1}{2}\left(\frac{\tilde{u}_1(k)}{\sigma_{1m}}\right)^2} \quad \mu_1^{\rho_2}(\tilde{u}_1(k)) = e^{-\frac{1}{2}\left(\frac{\tilde{u}_1(k)-1}{\sigma_{2m}}\right)^2}$$

$$\mu_2^{\rho_1}(\tilde{u}_2(k)) = e^{-\frac{1}{2}\left(\frac{\tilde{u}_2(k)}{\sigma_{3m}}\right)^2} \quad \mu_2^{\rho_2}(\tilde{u}_2(k)) = e^{-\frac{1}{2}\left(\frac{\tilde{u}_2(k)-1}{\sigma_{4m}}\right)^2} \quad (4.5)$$

Tabela 4.9 – Média dos desvios padrões das *Gaussianas* alcançados pelo PSO no 1º estudo de caso.

σ_{1m}	σ_{2m}	σ_{3m}	σ_{4m}
0,9595	0,6280	0,9625	0,2723

Portanto, os resultados acima mostram que o *framework* proposto, com o modelo de regressor 1, alcançou resultados satisfatórios para as métricas de avaliação e para a análise de resíduos, uma vez que o modelo *fuzzy* TSK MIMO 2x2 identificado consegue descrever a dinâmica do CSTR de maneira satisfatória para diferentes pontos de operação.

4.2 Secador Industrial (*Industrial Dryer*)

Secadores industriais são dispositivos utilizados em processos produtivos que possuem o objetivo de remover substâncias voláteis (umidade) de produtos sólidos utilizando princípios da termodinâmica, uma vez que o processo de secagem possui aplicações na indústria alimentícia, química, farmacêutica, etc. A secagem é uma das operações unitárias mais complexas e menos entendida, devido à dificuldade de modelar matematicamente todos os fenômenos físicos envolvidos no processo. (Pontes, 2014)

Atualmente, existem diversos modelos de secadores industriais, onde cada um possui uma arquitetura e uma dinâmica de secagem diferente, como por exemplo: *Batch Dryers*; *Rotary Dryers*; *Spray Dryers*; *Convoyer Dryers*; *Flash Dryers*; etc (Mujumdar, 2014). Para este trabalho, foram utilizados dados reais coletados de um secador industrial do tipo *Spray Dryer*, cujo esquemático é mostrado na Figura 4.8.

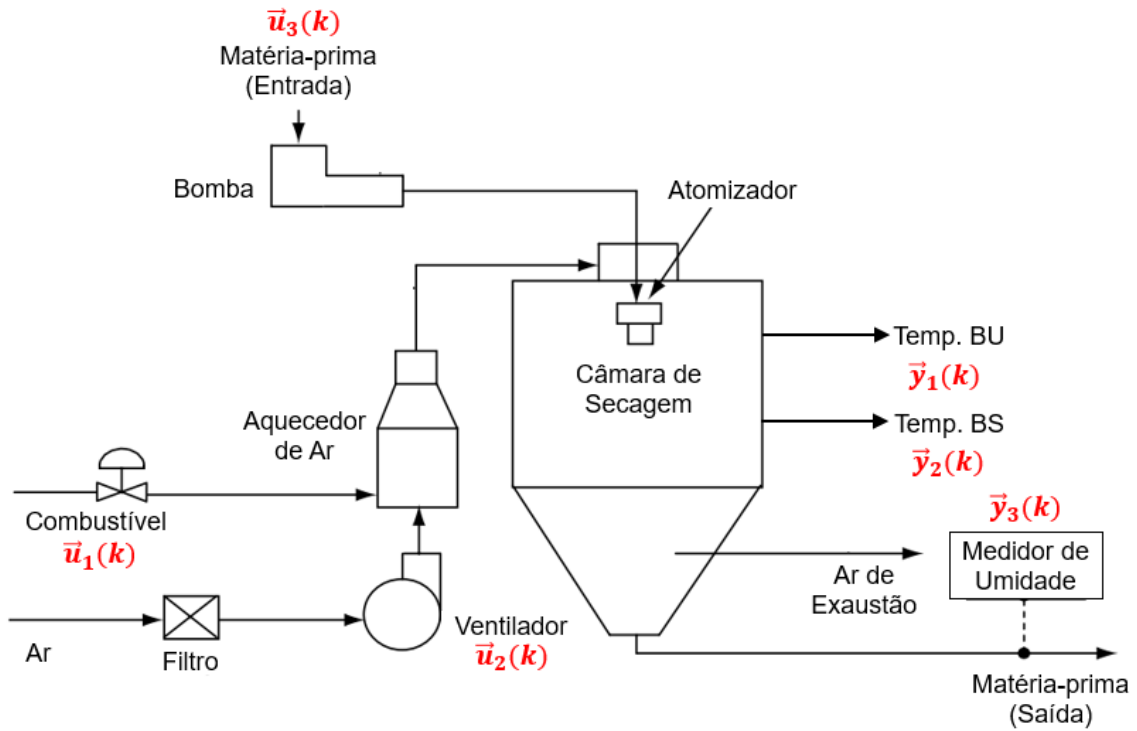


Figura 4.8 – Diagrama de um secador industrial do tipo *Spray Dryer*.
Fonte: Adaptado de Mujumdar (2014)

A figura acima ilustra de maneira genérica o mecanismo de um *Spray Dryer* utilizado neste estudo de caso, cuja planta possui três variáveis de entradas e três de saída, caracterizando-a como um sistema MIMO 3x3. As variáveis de entrada utilizadas para a identificação do secador são: Taxa do fluxo de combustível ($\vec{u}_1(k)$); Velocidade do ventilador do exaustor de gás quente ($\vec{u}_2(k)$); Taxa do fluxo de matéria-prima ($\vec{u}_3(k)$). E as variáveis de saída utilizadas são: Temperatura do bulbo úmido ($\vec{y}_1(k)$); Temperatura do bulbo seco ($\vec{y}_2(k)$); Teor de umidade da matéria-prima ($\vec{y}_3(k)$).

As unidades de medida das variáveis utilizadas para identificação desta planta não são informadas pelo provedor da base de dados, assim como também informações sobre os sensores utilizados ou dados do fabricante deste secador industrial, o que torna este um problema de identificação caixa-preta.

As Figuras 4.9 e 4.10 mostram respectivamente as amostras totais dos dados de entrada e saída coletados da planta do secador industrial em 77,83 minutos de operação, cuja taxa de amostragem é $T_s = 10$ s. E a Tabela 4.10 mostra a divisão percentual e temporal dos dados do secador industrial utilizados em cada etapa de cada técnica de identificação. Os dados foram divididos conforme a ordem temporal progressiva de execução da planta, e a divisão percentual das amostras é baseada na metodologia proposta em Silva et al. (2018).

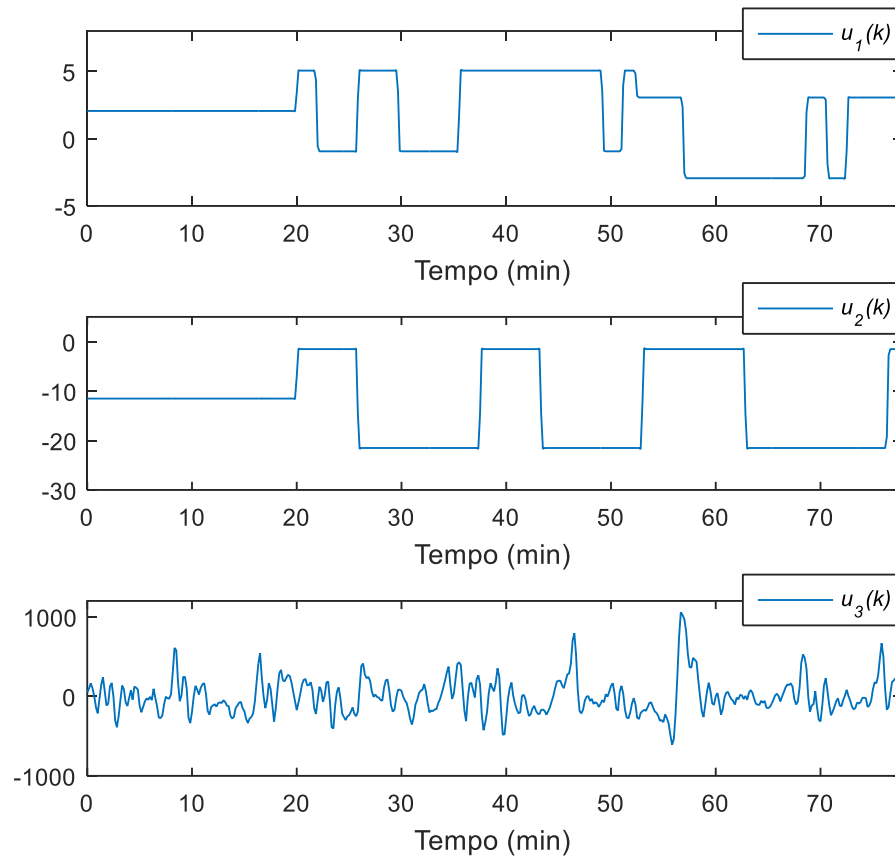


Figura 4.9 – Dados de entrada das variáveis $\bar{u}_1(k)$, $\bar{u}_2(k)$ e $\bar{u}_3(k)$.

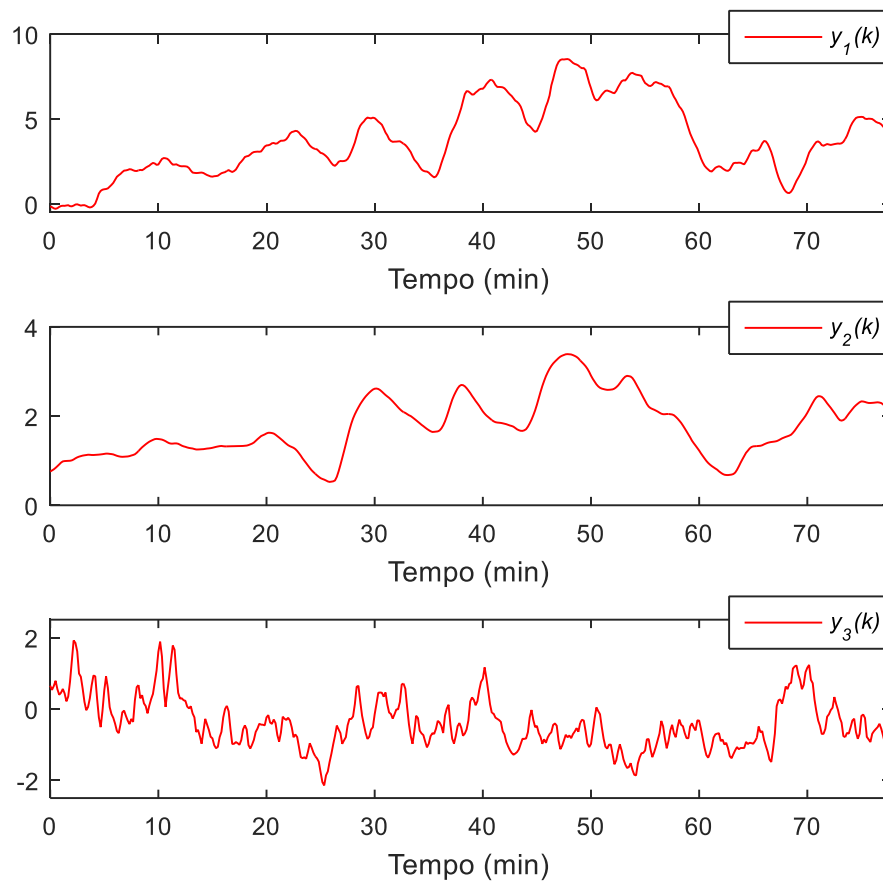


Figura 4.10 – Dados de saída das variáveis $\bar{y}_1(k)$, $\bar{y}_2(k)$ e $\bar{y}_3(k)$.

Tabela 4.10 – Divisão percentual e temporal dos dados para o 2º estudo de caso.

	Identificação/Treino 0 – 54,5 (min)	Validação 54,66 – 66,16 (min)	Teste 66,33 – 77,83 (min)
Framework	70%	15%	15%
RNA	70%	15%	15%
HW	70%	-	15%

Conforme a divisão estabelecida na tabela acima, observa-se que os dados de identificação são diferentes dos dados de teste, principalmente para a dinâmica dos sinais de saída da planta do secador, uma vez que este apresenta variações nos pontos de operação durante todo o tempo observado.

As Tabelas 4.11 a 4.13 mostram respectivamente a parametrização do PSO usado no *framework* conforme proposto em Oliveira et al. (2019), os modelos de vetor regressor utilizados nessa planta MIMO 3x3 e as regras do sistema *fuzzy* TSK multivariável.

Tabela 4.11 – Parâmetros do PSO do *framework* no 2º estudo de caso.

N_{part}	N_{var}	$\bar{w}$	c_1	c_2	X_{min}	X_{max}	V_{min}	V_{max}	N_{itotal}	$mEQM_i$
40	6,0	0,9	1,8	1,8	0,0	1,0	-0,5	0,5	100	10^{-3}

Tabela 4.12 – Modelos de regressores do *framework* no 2º estudo de caso.

	Regressor
Modelo 1 $\tau_{dy} = 1 \ \tau_{du} = 1$	$[y_1(k-1) \ y_2(k-1) \ y_3(k-1) \ \dots$ $u_1(k-1) \ u_2(k-1) \ u_3(k-1)]$
Modelo 2 $\tau_{dy} = 2 \ \tau_{du} = 2$	$[y_1(k-1) \ y_1(k-2) \ y_2(k-1) \ y_2(k-2) \ \dots$ $y_3(k-1) \ y_3(k-2) \ u_1(k-1) \ u_1(k-2) \ \dots$ $u_2(k-1) \ u_2(k-2) \ u_3(k-1) \ u_3(k-2)]$
Modelo 3 $\tau_{dy} = 3 \ \tau_{du} = 3$	$[y_1(k-1) \ y_1(k-2) \ y_1(k-3) \ \dots$ $y_2(k-1) \ y_2(k-2) \ y_2(k-3) \ \dots$ $y_3(k-1) \ y_3(k-2) \ y_3(k-3) \ \dots$ $u_1(k-1) \ u_1(k-2) \ u_1(k-3) \ \dots$ $u_2(k-1) \ u_2(k-2) \ u_2(k-3) \ \dots$ $u_3(k-1) \ u_3(k-2) \ u_3(k-3)]$

Tabela 4.13 – Regras do modelo *fuzzy* TSK no 2º estudo de caso.

Regras	Preposição SE - ENTÃO
$R = 1$	SE $\tilde{u}_1(k)$ é $A_1^{\rho_1}$ e $\tilde{u}_2(k)$ é $A_2^{\rho_1}$ e $\tilde{u}_3(k)$ é $A_3^{\rho_1}$ ENTÃO $\hat{Y}(k) = d^1 \xi_1(k)$
$R = 2$	SE $\tilde{u}_1(k)$ é $A_1^{\rho_1}$ e $\tilde{u}_2(k)$ é $A_2^{\rho_1}$ e $\tilde{u}_3(k)$ é $A_3^{\rho_2}$ ENTÃO $\hat{Y}(k) = d^2 \xi_2(k)$
$R = 3$	SE $\tilde{u}_1(k)$ é $A_1^{\rho_1}$ e $\tilde{u}_2(k)$ é $A_2^{\rho_2}$ e $\tilde{u}_3(k)$ é $A_3^{\rho_1}$ ENTÃO $\hat{Y}(k) = d^3 \xi_3(k)$
$R = 4$	SE $\tilde{u}_1(k)$ é $A_1^{\rho_1}$ e $\tilde{u}_2(k)$ é $A_2^{\rho_2}$ e $\tilde{u}_3(k)$ é $A_3^{\rho_2}$ ENTÃO $\hat{Y}(k) = d^4 \xi_4(k)$
$R = 5$	SE $\tilde{u}_1(k)$ é $A_1^{\rho_2}$ e $\tilde{u}_2(k)$ é $A_2^{\rho_1}$ e $\tilde{u}_3(k)$ é $A_3^{\rho_1}$ ENTÃO $\hat{Y}(k) = d^5 \xi_5(k)$
$R = 6$	SE $\tilde{u}_1(k)$ é $A_1^{\rho_2}$ e $\tilde{u}_2(k)$ é $A_2^{\rho_1}$ e $\tilde{u}_3(k)$ é $A_3^{\rho_2}$ ENTÃO $\hat{Y}(k) = d^6 \xi_6(k)$
$R = 7$	SE $\tilde{u}_1(k)$ é $A_1^{\rho_2}$ e $\tilde{u}_2(k)$ é $A_2^{\rho_2}$ e $\tilde{u}_3(k)$ é $A_3^{\rho_1}$ ENTÃO $\hat{Y}(k) = d^7 \xi_7(k)$
$R = 8$	SE $\tilde{u}_1(k)$ é $A_1^{\rho_2}$ e $\tilde{u}_2(k)$ é $A_2^{\rho_2}$ e $\tilde{u}_3(k)$ é $A_3^{\rho_2}$ ENTÃO $\hat{Y}(k) = d^8 \xi_8(k)$

As Tabelas 4.14 e 4.15 mostram as configurações paramétricas do modelo HW e da RNA respectivamente para este estudo de caso. Ambas as configurações foram ajustadas de maneira heurística.

Tabela 4.14 – Parâmetros do modelo Hammerstein-Wiener no 2º estudo de caso.

Parte Não Linear (Estática)	Funções Estáticas da Entrada	Função Network Sigmoid (60 parâmetros)
	Funções Estáticas da Saída	Função Network Sigmoid (6 parâmetros)
	Algoritmo de Estimação	Levenberg-Marquardt
	Número Máximo de Iterações	30
Parte Linear (Dinâmica)	Funções de Transferência (FT) da Saída 1	Número de FT: 3 Estrutura por FT: 4 polos; 3 zeros; 4 atrasos
	Funções de Transferência (FT) da Saída 2	Número de FT: 3 Estrutura por FT: 4 polos; 3 zeros; 4 atrasos
	Funções de Transferência (FT) da Saída 3	Número de FT: 3 Estrutura por FT: 4 polos; 3 zeros; 4 atrasos
	Algoritmo de Estimação	Mínimos Quadrados

Tabela 4.15 – Parâmetros da RNA no 2º estudo de caso.

Função da Camada Escondida	2 Camadas Escondidas Tangente Sigmoid (40 Neurônios) Tangente Sigmoid (40 Neurônios)
Função da Camada de Saída	Linear (3 Neurônios)
Algoritmo de Treinamento	Levenberg-Marquardt
Entradas da RNA	$\begin{bmatrix} y_1(k-1) & y_1(k-2) & y_2(k-1) & y_2(k-2) & \dots \\ y_3(k-1) & y_3(k-2) & u_1(k-1) & u_1(k-2) & \dots \\ u_2(k-1) & u_2(k-2) & u_3(k-1) & u_3(k-2) \end{bmatrix}$
Saídas da RNA	$[y_1(k) \ y_2(k) \ y_3(k)]$

A Tabela 4.16 a seguir mostra o desempenho do *framework* na etapa de validação utilizando cada um dos três modelos de vetor regressor.

Tabela 4.16 – Resultados do EQM_v e do $mEQM_v$ para os 3 modelos de regressores utilizados no *framework* na etapa de identificação no 2º estudo de caso.

Framework	Modelo 1	Modelo 2	Modelo 3
$EQM_v(y_1)$	0,0660	0,3114	1,0189
$EQM_v(y_2)$	0,0049	0,0083	0,0227
$EQM_v(y_3)$	1,9740	5,3359	15,3901
$mEQM_v(y_1, y_2, y_3)$	0,6817	1,8852	5,4773

Para este estudo de caso, o *framework* alcançou melhor desempenho utilizando o vetor regressor 1, uma vez que os menores valores de EQM_v foram alcançados. O resultado reflete que o regressor mais simples obteve melhor êxito na composição do modelo *fuzzy* TSK multivariável para a representação da dinâmica secador industrial.

Uma possível explicação para o regressor 1 ter alcançado o melhor desempenho dentre todos está na sua estrutura matemática. Assim como no caso anterior, este regressor agrega menos informações, tornando a matriz de regressores menor, o que tem impacto direto na diminuição do erro de estimação da matriz de parâmetros com o MQ.

O modelo *fuzzy* TSK multivariável estimado com o regressor 1 é utilizado na fase de teste, pois obteve o melhor resultado na etapa de validação. As Figuras 4.11 a 4.13 mostram respectivamente os resultados das saídas y_1 , y_2 e y_3 alcançados na identificação da planta do secador industrial com os dados de teste.

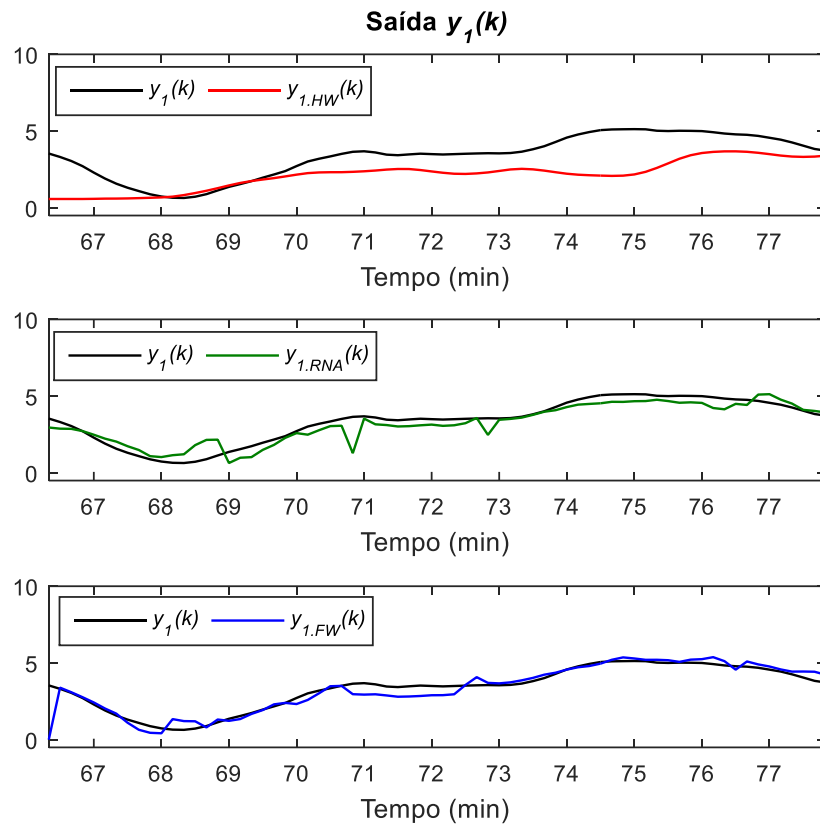


Figura 4.11 – Saída y_1 (Temperatura do bulbo úmido) do tanque observada (preto), e estimada pelo HW (vermelho), pela RNA (verde) e pelo *framework* (azul).

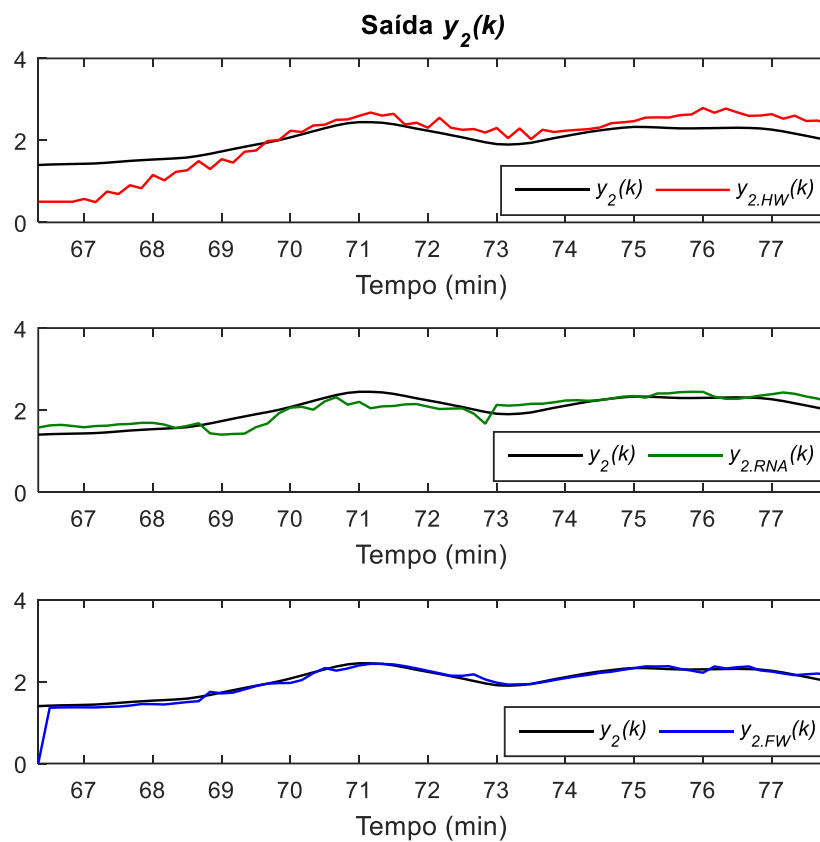


Figura 4.12 - Saída y_2 (Temperatura do bulbo seco) do tanque observada (preto), e estimada pelo HW (vermelho), pela RNA (verde) e pelo *framework* (azul).

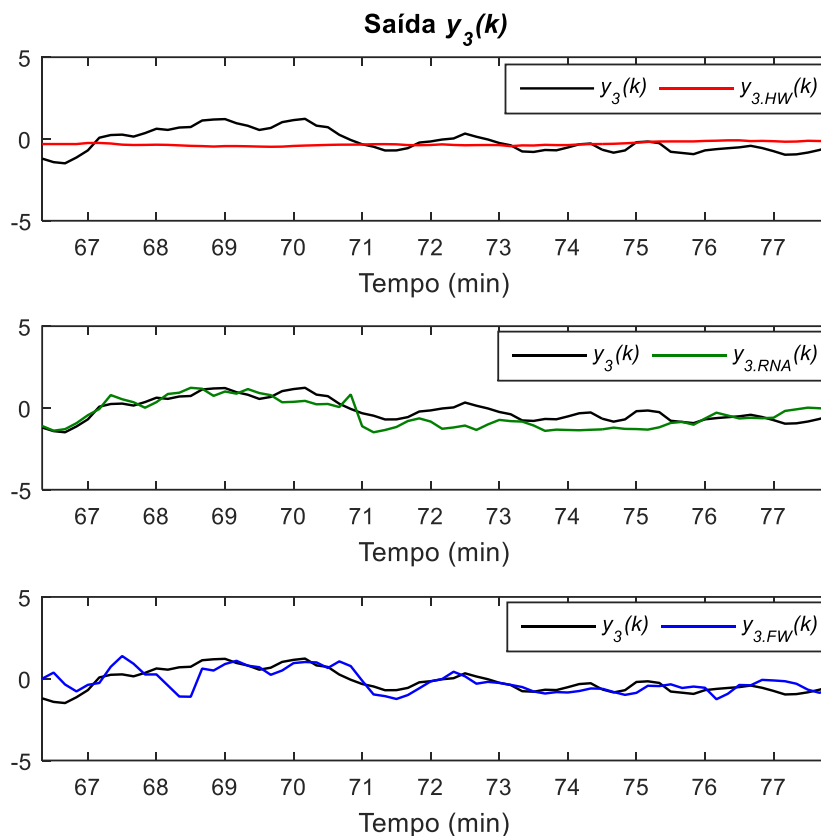


Figura 4.13 - Saída y_3 (Teor de umidade da matéria-prima) do tanque observada (preto), e estimada pelo HW (vermelho), pela RNA (verde) e pelo *framework* (azul).

É importante destacar nas Figuras 4.11 a 4.13 que para $k = 1$, a saída estimada pelo *framework* é sempre igual a zero, devido aos regressores utilizados possuírem valor diferente de zero apenas para $k \geq 2$. Enquanto que para as demais técnicas o valor inicialização dos seus respectivos algoritmos de identificação.

A Tabela 4.17 exibe os resultados do EQM_t e do VAF alcançados pelas técnicas de identificação utilizando os dados de teste da planta.

Tabela 4.17 – Resultados do EQM_t e do VAF para os dados de teste no 2º estudo de caso.

Critério	Técnica		
	<i>Framework</i> (Regressor 1)	RNA	HW
$EQM_t(y_1)$	0,1207	0,2949	2,1931
$VAF(y_1)$	93,51 %	85,65 %	58,84 %
$EQM_t(y_2)$	0,0041	0,0342	0,1506
$VAF(y_2)$	96,12 %	66,29 %	-48,00 %
$EQM_t(y_3)$	0,3308	0,3983	0,6033
$VAF(y_3)$	29,90 %	29,63 %	-22,84 %

Conforme os resultados da Tabela 4.17 e das Figuras 4.11 a 4.13, pode-se observar que para as saídas y_1 e y_2 o *framework* alcançou um desempenho em termos de EQM_t e VAF superior as demais técnicas, enquanto que para a terceira saída, o seu desempenho também foi melhor que as outras duas técnicas, embora próximo ao da RNA. E com base nos gráficos dos sinais de saída, pode-se observar também que, de maneira geral, o *framework* consegue estimar as temperaturas com valores bem próximos aos valores reais, enquanto que o teor de umidade da matéria-prima, cujos dados possuem muita não linearidade, foi estimado com valores relativamente próximos.

As Figuras 4.14 a 4.16 mostram os resíduos gerados por cada técnica entre o sinal observado e o estimado de cada saída. E com base nesses gráficos, pode-se inferir que resíduo originado pela identificação do secador industrial com o *framework* para as saídas y_1 e y_2 foi mais próxima ao valor zero quando comparada com a RNA e o HW, o que demonstra que a técnica proposta obteve um bom desempenho e baixo erro entre o sinal original e o estimado para a representação da dinâmica da planta durante o intervalo de operação dos dados de teste.

Entretanto, para a saída y_3 , nota-se que as três técnicas de identificação geraram resíduos razoáveis para os dados de teste. O *framework* gerou menos resíduos a partir dos 73 minutos, e a RNA comportou-se melhor antes deste tempo, entretanto, o modelo de Hammerstein-Wiener gerou erros significativos ao ponto de linearizar esta saída.

Os resíduos são importantes para avaliar o erro de estimação do secador atuando em diversos pontos de operação durante o período de análise. E conforme os gráficos das Figuras 4.14 a 4.16 mostram, o *framework* alcançou desempenho melhor na estimação das duas temperaturas, enquanto a umidade não foi muito bem estimada.

Uma possível explicação para essa divergência pode ser encontrada nas próprias física que rege essas variáveis. Pois, enquanto as temperaturas dos bulbos do secador evoluem lentamente ao longo do tempo, a umidade apresenta variações que ocorrem em intervalos de tempo menores, o que contribui para a dificuldade de se estimar um modelo para a terceira saída da planta, uma vez que plantas com dinâmicas rápidas demandam estruturas mais complexas de modelos para serem aproximados.

Portanto, a análise dos resíduos indica que ainda existe uma necessidade de se utilizar um vetor regressor com uma estrutura mais robusta, de modo que este consiga rastrear o teor de umidade com valor mais aproximado do real.

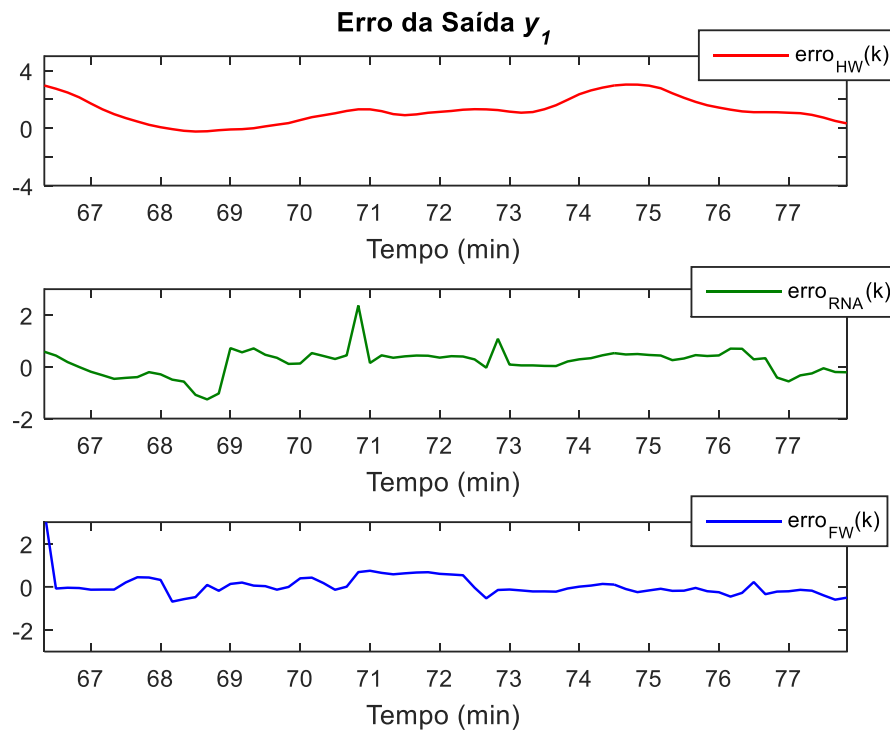


Figura 4.14 – Sinal de erro da saída y_1 do secador industrial com a saída estimada pelo HW (vermelho), pela RNA (verde) e pelo *framework* (azul).

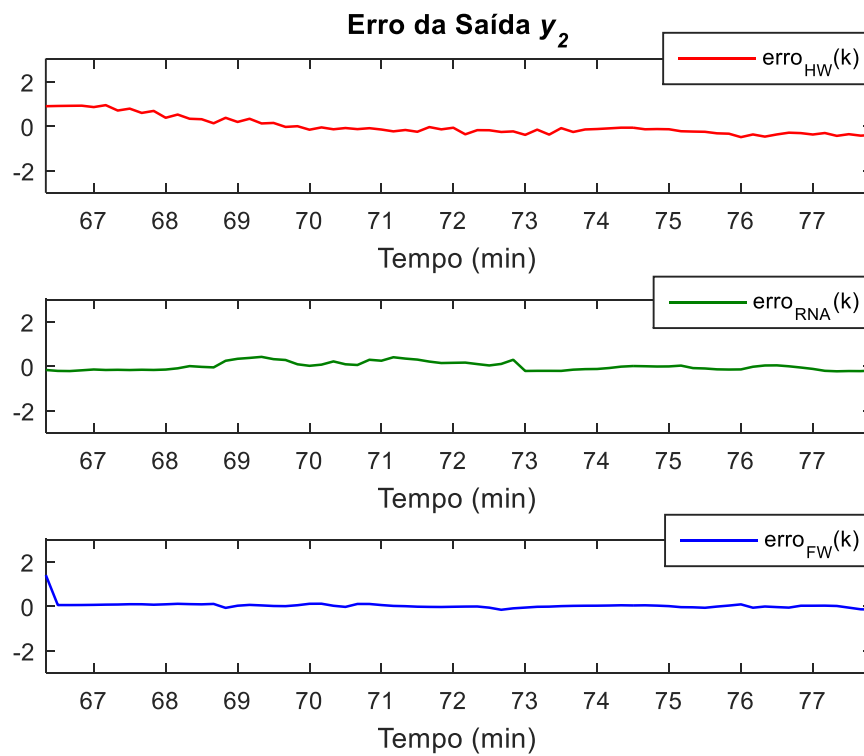


Figura 4.15 – Sinal de erro da saída y_2 do secador industrial com a saída estimada pelo HW (vermelho), pela RNA (verde) e pelo *framework* (azul).

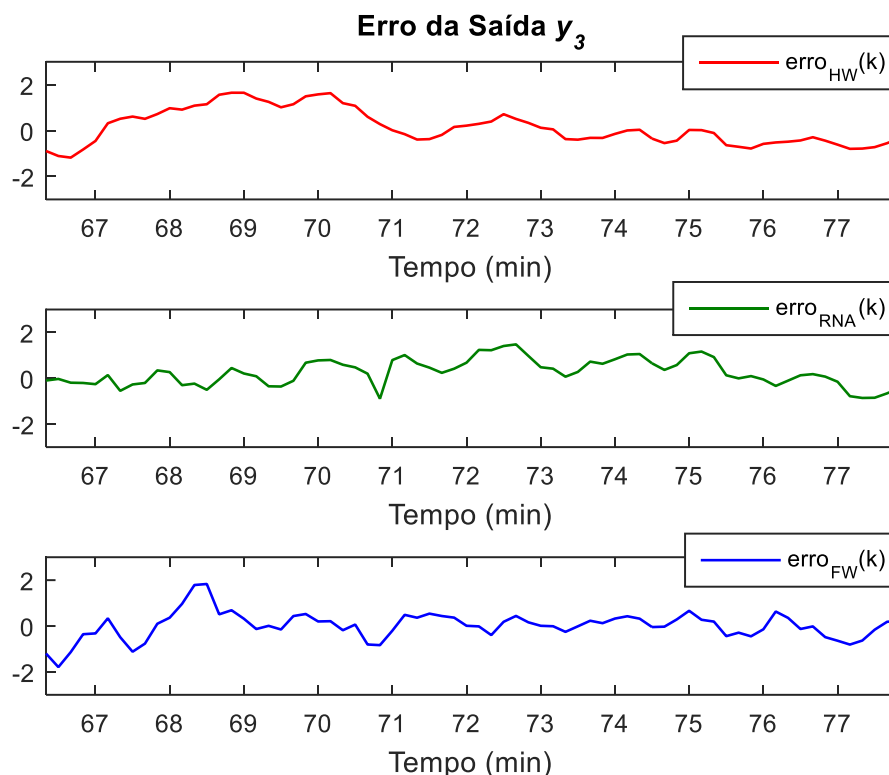


Figura 4.16 – Sinal de erro da saída y_3 do secador industrial com a saída estimada pelo HW (vermelho), pela RNA (verde) e pelo *framework* (azul).

A equação do modelo *fuzzy* TSK multivariável 3x3 do secador industrial identificado pelo *framework*, utilizando o modelo 1 de vetor regressor e a média da matriz de parâmetros estimada pelo MQ, é mostrada nas Equações 4.6 a 4.8. A Tabela 4.18 mostra a média dos desvios padrões estimados pelo PSO. As estruturas da RNA e do modelo HW, além do gráfico da evolução da média da função objetivo do PSO são mostradas no apêndice II.

Tabela 4.18 – Média dos desvios padrões das *Gaussianas* alcançados pelo PSO no 2º estudo de caso.

σ_{1m}	σ_{2m}	σ_{3m}	σ_{4m}	σ_{5m}	σ_{6m}
0,3332	0,4219	0,3870	0,7997	0,3520	0,2787

$$\begin{aligned}
\begin{bmatrix} \hat{y}_1(k) \\ \hat{y}_2(k) \\ \hat{y}_3(k) \end{bmatrix} &= \begin{bmatrix} 1,2531 & -0,5555 & -0,1082 & 0,2017 & 0,0096 & 0,0006 \\ -0,0172 & 1,1437 & -0,0450 & -0,0101 & 0,0145 & -0,0001 \\ 2,0348 & -5,7494 & 2,5046 & -0,3342 & -0,3323 & 0,0045 \end{bmatrix} \phi^T \xi_1(k) \\
&+ \begin{bmatrix} 1,3059 & -1,3922 & 1,1666 & 0,0039 & -0,1355 & -0,0007 \\ 0,0673 & 0,6677 & 0,0390 & 0,0848 & -0,0275 & 0,0001 \\ -3,3048 & 11,7415 & -2,4628 & 0,0945 & 0,8267 & -0,0013 \end{bmatrix} \phi^T \xi_2(k) \\
&+ \begin{bmatrix} 1,2174 & -0,8837 & 0,1716 & 0,0203 & -0,0614 & -0,0001 \\ 0,0386 & 0,7974 & 0,0215 & 0,0167 & -0,0115 & 0 \\ -1,3864 & 3,6612 & -0,6305 & 0,3624 & 0,2384 & 0,0009 \end{bmatrix} \phi^T \xi_3(k) \\
&+ \begin{bmatrix} 0,2122 & 2,8250 & -0,3439 & -0,0554 & 0,1999 & -0,0001 \\ -0,0537 & 1,2150 & 0,0391 & -0,0134 & 0,0218 & 0,0001 \\ 2,6252 & -7,9734 & 4,2087 & -0,4636 & -0,6285 & -0,0061 \end{bmatrix} \phi^T \xi_4(k) \\
&+ \begin{bmatrix} 0,5923 & 0,9560 & -0,8556 & -0,2300 & -0,0277 & 0,0002 \\ -0,0067 & 0,9112 & -0,0428 & -0,0378 & -0,0155 & 0,0002 \\ -0,1929 & 0,1964 & 0,5421 & 0,1245 & 0,0349 & -0,0015 \end{bmatrix} \phi^T \xi_5(k) \\
&+ \begin{bmatrix} 0,7794 & 0,6757 & 0,5537 & 0,2860 & 0,0651 & 0,0011 \\ -0,0751 & 1,2776 & 0,1347 & 0,0723 & 0,0149 & -0,0003 \\ 0,1489 & -1,2532 & 1,4303 & 0,1768 & -0,1061 & 0,0005 \end{bmatrix} \phi^T \xi_6(k) \\
&+ \begin{bmatrix} 0,8107 & 0,6672 & 0,2195 & 0,0083 & 0,0439 & 0,0001 \\ -0,0418 & 1,1310 & -0,0143 & 0,0017 & 0,0035 & 0 \\ 0,0755 & -0,0777 & 1,3818 & -0,0138 & -0,0167 & 0,0001 \end{bmatrix} \phi^T \xi_7(k) \\
&+ \begin{bmatrix} 1,4027 & -1,0870 & -0,2723 & -0,0344 & -0,0580 & -0,0005 \\ 0,0775 & 0,6922 & -0,0495 & 0,0066 & -0,0164 & 0 \\ -0,2540 & 1,1238 & 0,1301 & -0,1840 & 0,0750 & 0,0009 \end{bmatrix} \phi^T \xi_8(k)
\end{aligned} \tag{4.6}$$

$$\phi = [y_1(k-1) \quad y_2(k-1) \quad y_3(k-1) \quad u_1(k-1) \quad u_2(k-1) \quad u_3(k-1)]$$

$$\begin{aligned}
\xi_1(k) &= \frac{\mu_1^{\rho_1}(\tilde{u}_1(k))\mu_2^{\rho_1}(\tilde{u}_2(k))\mu_3^{\rho_1}(\tilde{u}_3(k))}{\sum_{R=1}^8 \mu_R(\tilde{u}_1(k), \tilde{u}_2(k), \tilde{u}_3(k))} \quad \xi_2(k) = \frac{\mu_1^{\rho_1}(\tilde{u}_1(k))\mu_2^{\rho_1}(\tilde{u}_2(k))\mu_3^{\rho_2}(\tilde{u}_3(k))}{\sum_{R=1}^8 \mu_R(\tilde{u}_1(k), \tilde{u}_2(k), \tilde{u}_3(k))} \\
\xi_3(k) &= \frac{\mu_1^{\rho_1}(\tilde{u}_1(k))\mu_2^{\rho_2}(\tilde{u}_2(k))\mu_3^{\rho_1}(\tilde{u}_3(k))}{\sum_{R=1}^8 \mu_R(\tilde{u}_1(k), \tilde{u}_2(k), \tilde{u}_3(k))} \quad \xi_4(k) = \frac{\mu_1^{\rho_1}(\tilde{u}_1(k))\mu_2^{\rho_2}(\tilde{u}_2(k))\mu_3^{\rho_2}(\tilde{u}_3(k))}{\sum_{R=1}^8 \mu_R(\tilde{u}_1(k), \tilde{u}_2(k), \tilde{u}_3(k))}
\end{aligned} \tag{4.7}$$

$$\begin{aligned}
\xi_5(k) &= \frac{\mu_1^{\rho_2}(\tilde{u}_1(k))\mu_2^{\rho_1}(\tilde{u}_2(k))\mu_3^{\rho_1}(\tilde{u}_3(k))}{\sum_{R=1}^8 \mu_R(\tilde{u}_1(k), \tilde{u}_2(k), \tilde{u}_3(k))} & \xi_6(k) &= \frac{\mu_1^{\rho_2}(\tilde{u}_1(k))\mu_2^{\rho_1}(\tilde{u}_2(k))\mu_3^{\rho_2}(\tilde{u}_3(k))}{\sum_{R=1}^8 \mu_R(\tilde{u}_1(k), \tilde{u}_2(k), \tilde{u}_3(k))} \\
\xi_7(k) &= \frac{\mu_1^{\rho_2}(\tilde{u}_1(k))\mu_2^{\rho_2}(\tilde{u}_2(k))\mu_3^{\rho_1}(\tilde{u}_3(k))}{\sum_{R=1}^8 \mu_R(\tilde{u}_1(k), \tilde{u}_2(k), \tilde{u}_3(k))} & \xi_8(k) &= \frac{\mu_1^{\rho_2}(\tilde{u}_1(k))\mu_2^{\rho_2}(\tilde{u}_2(k))\mu_3^{\rho_2}(\tilde{u}_3(k))}{\sum_{R=1}^8 \mu_R(\tilde{u}_1(k), \tilde{u}_2(k), \tilde{u}_3(k))} \\
\mu_1^{\rho_1}(\tilde{u}_1(k)) &= e^{-\frac{1}{2}\left(\frac{\tilde{u}_1(k)}{\sigma_{1m}}\right)^2} & \mu_1^{\rho_2}(\tilde{u}_1(k)) &= e^{-\frac{1}{2}\left(\frac{\tilde{u}_1(k)-1}{\sigma_{2m}}\right)^2} \\
\mu_2^{\rho_1}(\tilde{u}_2(k)) &= e^{-\frac{1}{2}\left(\frac{\tilde{u}_2(k)}{\sigma_{3m}}\right)^2} & \mu_2^{\rho_2}(\tilde{u}_2(k)) &= e^{-\frac{1}{2}\left(\frac{\tilde{u}_2(k)-1}{\sigma_{4m}}\right)^2} \\
\mu_3^{\rho_1}(\tilde{u}_3(k)) &= e^{-\frac{1}{2}\left(\frac{\tilde{u}_3(k)}{\sigma_{5m}}\right)^2} & \mu_3^{\rho_2}(\tilde{u}_3(k)) &= e^{-\frac{1}{2}\left(\frac{\tilde{u}_3(k)-1}{\sigma_{6m}}\right)^2}
\end{aligned} \tag{4.8}$$

Portanto, conforme os resultados mostrados acima, o *framework* com o modelo 1 de vetor regressor alcançou um desempenho satisfatório para a identificação da planta do secador industrial no modelo *fuzzy* TSK MIMO 3x3, tanto com base nas métricas de avaliação, quanto na análise dos resíduos, uma vez que o resultado mostra que o modelo identificado consegue descrever a dinâmica desta planta de maneira satisfatória para vários pontos de operação.

4.3 Análise Geral dos Resultados

Em uma análise geral dos resultados, a Tabela 4.19 sumariza o desempenho comparado do *framework* proposto com a RNA e o modelo HW no processo de identificação das duas plantas multivariadas analisadas, destacando qual técnica alcançou melhor resultado nos índices EQM_t e VAF por saída de cada planta identificada.

Tabela 4.19 – Sumário das técnicas com melhor desempenho em EQM_t e VAF por saída de cada planta.

<i>Planta</i>	CSTR	Secador Industrial
$EQM_t(y_1)$	RNA	<i>Framework</i>
$VAF(y_1)$	RNA	<i>Framework</i>
$EQM_t(y_2)$	<i>Framework</i>	<i>Framework</i>
$VAF(y_2)$	<i>Framework</i>	<i>Framework</i>
$EQM_t(y_3)$	-	<i>Framework</i>
$VAF(y_3)$	-	<i>Framework</i>

Com base nos resultados da tabela acima, considerando-se uma análise conjunta de EQM_t e VAF , pode-se inferir que no total 5 saídas analisadas na etapa de teste em todo o resultado deste trabalho, o *framework* obteve o melhor desempenho em 80% dos casos (saída y_2 da planta do reator e todas as saídas do secador industrial) e a RNA teve melhor desempenho em 20% dos casos (saída y_1 da planta do reator), enquanto o modelo Hammerstein-Wiener não atingiu o melhor desempenho em nem dos casos analisados.

Para a análise geral dos resíduos, a Tabela 4.20 mostra o valor da média e variância do sinal de resíduo gerado pelas técnicas em cada saída das plantas identificadas.

Tabela 4.20 – Sumário da **Média** e **Variância** dos sinais de resíduo de cada técnica para cada saída.

Resíduo	CSTR			Secador Industrial		
	<i>Framework</i>	RNA	HW	<i>Framework</i>	RNA	HW
Média – y_1	0,0783	0,0101	0,1192	0,0588	0,1729	1,2198
Var. – y_1	0,5086	0,0082	0,1907	0,2986	0,2695	0,8106
Média – y_2	-0,0487	-0,238	-0,1347	0,0311	-0,0017	-0,0032
Var. – y_2	0,3279	0,2968	0,4333	0,0315	0,0347	0,1622
Média – y_3	-	-	-	0,0163	0,2545	2,3233
Var. – y_3	-	-	-	0,3514	0,3327	0,0929

Para que o modelo identificado possa representar bem a dinâmica de uma planta, o ideal é que a média do resíduo seja igual ou muito próxima de zero e a variância também seja próxima a zero. Portanto, pela análise conjunta da relação média –variância, mostrado na tabela acima, nota-se que o *framework* obteve os melhores resultados em 60% dos resultados totais (número de saídas analisadas em todo o resultado), em segundo lugar fica a RNA com 40% seguido do modelo HW com 0%.

É importante destacar que os resultados da tabela acima são meramente quantitativos e calculados para toda a janela temporal de dados de teste. Todavia, em alguns casos, conforme os gráficos dos resíduos nos tópicos anteriores mostraram, cada técnica conseguiu bom desempenho em determinados pontos de operação da planta, enquanto em outros pontos a identificação foi precária. Isso denota a necessidade de se fazer uma análise de resíduos tanto gráfica quanto numérica, pois cada planta analisada, dependendo da necessidade e nível de segurança, pode ter sua margem de erro para identificação.

Com base em todos os resultados abordados, observa-se que o *framework* alcançou a maioria dos melhores resultados com base nas métricas de avaliação de desempenho

propostas, ou seja, ele consegue identificar e representar sistemas com esses níveis de dificuldade de maneira satisfatória.

4.4 Conclusão

Nesse capítulo foram apresentados os resultados da aplicação do *framework* proposto, juntamente com a RNA e o modelo Hammerstein-Wiener, na identificação de duas plantas MIMO não linear: Reator Contínuo de Tanque Agitado (CSTR); Secador Industrial. Também foram abordadas as análises comparativas dos resultados de cada técnica com base em métricas de avaliação de desempenho e análise de resíduo. Os resultados deste capítulo mostram que *framework* foi capaz de identificar cada uma das plantas e representar sua dinâmica de operação de maneira satisfatória.

No próximo capítulo será feita uma análise geral os resultados deste trabalho, assim como a também serão citadas propostas de possíveis melhorias na técnica proposta, além de trabalhos futuros.

5 CONSIDERAÇÕES FINAIS

5.1 Conclusão

Neste trabalho foi apresentada a proposta de *framework* para identificação de sistema multivariáveis não lineares, onde toda a sua estrutura, equações, etapas e metodologia de uso no processo de identificação foram detalhados. Também foi explicado a matemática e a lógica do modelo *fuzzy* TSK multivariável, o qual representa a estrutura final do sistema identificado pelo *framework*, assim como também foi descrito os algoritmos PSO e Mínimos Quadrados, e como estes interagem no processo de identificação e composição do modelo final.

Também foi abordada a importância, a aplicação e base teórico sobre identificação de sistemas multivariáveis, onde as Redes Neurais Artificiais (MLP) e modelos de Hammerstein-Wiener foram detalhadas mais profundamente, uma vez que estas são duas das técnicas clássicas mais usadas na resolução deste problema.

Para avaliar o *framework* proposto, o mesmo foi comparado com a RNA e com o modelo HW na identificação *off-line* e caixa-preta de duas plantas multivariadas e não lineares: Reator Contínuo de Tanque Agitado (CSTR) com 2 entradas e 2 saídas; Secador Industrial com 3 entradas e 3 saídas. Para cada planta, as três técnicas foram comparadas com os mesmos dados de testes para dois índices de desempenho EQM_t e VAF com o intuito de mensurar o erro de estimação e o quanto a saída estimada se aproxima da real. Também foi realizada a análise do resíduo gerado por cada algoritmo de identificação com o objetivo de indicar qual a técnica consegue estimar a o modelo que mais consiga representar a dinâmica do sistema real.

Dos resultados obtidos neste trabalho é possível verificar que o *framework* obteve o melhor desempenho em 80% das estimações de saídas das plantas com base nos índices EQM_t e VAF , e também alcançou o melhor desempenho em 60% dos casos na análise residual da identificação das duas plantas multivariadas. Por tanto, com base em todos os resultados abordados, pode-se concluir que o *framework* pode ser utilizado como técnica de identificação de sistemas multivariáveis, uma vez que os resultados mostram a sua boa eficiência na representação da dinâmica de sistemas que possuem várias entradas e saídas, assim como acentuada não linearidade e diferenças no ponto de operação.

5.2 Trabalhos Futuros

Existem alguns pontos não abordados neste trabalho que são interessantes de serem investigados. O primeiro seria a aplicação do *framework* proposto na identificação de plantas com mais de três entradas e saídas, uma vez que com este estudo de caso, seria possível saber as limitações da proposta para a resolução deste problema utilizando apenas um modelo *fuzzy* TSK multivariável, pois assim, seria possível saber a partir de qual configuração (quantidade) de entradas e saídas seria mais viável trabalhar com identificação em sistemas multi modelos.

O estudo da modelagem do resíduo e acoplamento deste no modelo final da planta identificada também é ponto interessante a ser pesquisado, pois para as saídas com dinâmica bastante não linear, talvez essa seja uma boa estratégia para a diminuição do erro de estimação.

Outro trabalho importante seria a análise estatística do desempenho do *framework* em relação ao ajuste de cada parâmetro do PSO (ou outra técnica metaheurística ou clássica de otimização escolhida) e do modelo *fuzzy* TSK no processo de identificação de plantas multivariadas, de modo a avaliar como a técnica proposta neste trabalho se comporta em relação à acurácia e precisão da estimação paramétrica, além do tempo de processamento gasto para se obter a planta identificada após a execução do *framework*.

A avaliação do custo computacional do *framework*, envolvendo o tempo de processamento de cada uma de suas etapas, também pode ser considerada uma análise interessante de ser realizada em trabalhos futuros.

REFERÊNCIAS BIBLIOGRÁFICAS

- Adánez, José Miguel, et al. “Multidimensional Membership Functions in T–S Fuzzy Models for Modelling and Identification of Nonlinear Multivariable Systems Using Genetic Algorithms.” *Applied Soft Computing*, vol. 75, Elsevier B.V., Feb. 2019, pp. 607–15, doi:10.1016/j.asoc.2018.11.034.
- Aguirre, Luis Antonio. *INTRODUÇÃO À IDENTIFICAÇÃO DE SISTEMAS Técnicas Lineares e Não Lineares: Teoria e Aplicação*. 4th ed., Editora UFMG, 2014.
- Ahmed, Ahmed Babiker, et al. *Control System Design for Continuous Stirred Tank Reactor Using Matlab Simulink*. no. July, Nile Valley University, 2016.
- Antonelli, Rita, and Alessandro Astolfi. “Continuous Stirred Tank Reactors: Easy to Stabilise?” *Automatica*, vol. 39, no. 10, Oct. 2003, pp. 1817–27, doi:10.1016/S0005-1098(03)00177-8.
- Azevedo, Ana Zuila Castro Calandrini de. *APLICAÇÃO DE REDE NEURO-FUZZY ANFIS NA IDENTIFICAÇÃO DE SISTEMAS DINÂMICOS MULTIVARIADOS NÃO LINEARES*. UNIVERSIDADE FEDERAL DO PARÁ, 2019, doi:10.1037/0033-2909.126.1.78.
- Beni, Gerardo, and Jing Wang. “Swarm Intelligence in Cellular Robotic Systems.” *Robots and Biological Systems: Towards a New Bionics?*, Springer Berlin Heidelberg, 1993, pp. 703–12, doi:10.1007/978-3-642-58069-7_38.
- Boutalis, Yiannis, et al. *System Identification and Adaptive Control*. Springer International Publishing, 2014, doi:10.1007/978-3-319-06364-5.
- Bouzaida, Sana, et al. *Identification of MIMO Systems Using Neuro-Fuzzy Models with a Shuffled Frog Leaping Algorithm*. Vol. 6, no. 5, 2012, pp. 1120–25.
- Brabazon, A., et al. *Natural Computing Algorithms*. 2015.
- Cham, Chin Leei, et al. “Identification of a Multivariable Nonlinear and Time-Varying Mist Reactor System.” *Control Engineering Practice*, vol. 63, no. May 2016, Elsevier Ltd, June 2017, pp. 13–23, doi:10.1016/j.conengprac.2017.03.010.
- Chiuso, A., and G. Pillonetto. “System Identification: A Machine Learning Perspective.” *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 2, no. 1, May 2019, pp. 281–304, doi:10.1146/annurev-control-053018-023744.
- Coelho, Antônio augusto Rodrigues, and Leandro dos Santos Coelho. *Identificação de Sistemas Dinâmicos Lineares*. Editora da UFSC, 2004.
- Oliveira, E. C., et al. “Quadrotor Black-Box System Identification Using

- Metaheuristics.” *DINAME2019*, ABCM, 2019, doi:10.26678/ABCM.DINAME2019.DIN2019-0078.
- Y. Zhu. “Multivariable System Identification For Process Control.” *Multivariable System Identification For Process Control*, no. October, 2000, doi:10.1016/b978-0-08-043985-3.x5000-5.
- Fogler, H. Scott. *Elementos de Engenharia Das Reações Químicas*. 4th ed., LTC, 2009.
- Fu, Xingang, et al. “Training Recurrent Neural Networks With the Levenberg–Marquardt Algorithm for Optimal Control of a Grid-Connected Converter.” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 9, Sept. 2015, pp. 1900–12, doi:10.1109/TNNLS.2014.2361267.
- Gavin, Henri P. *The Levenberg-Marquardt Algorithm for Nonlinear Least Squares Curve-Fitting Problems*. 2019, pp. 1–19, <http://people.duke.edu/~hpgavin/ce281/lm.pdf>.
- Goethals, Ivan, et al. “Subspace Intersection Identification of Hammerstein-Wiener Systems.” *Proceedings of the 44th IEEE Conference on Decision and Control, and the European Control Conference, CDC-ECC '05*, vol. 2005, 2005, pp. 7108–13, doi:10.1109/CDC.2005.1583307.
- Hagan, Martin T., et al. *Neural Network Design*. 2014, doi:10.1007/BF00738424.
- Hanafy, Tharwat O. S., et al. *Identification of Uncertain Nonlinear MIMO Spacecraft Systems Using Coactive Neuro Fuzzy Inference System (CANFIS)*. Vol. 3, no. 2, 2014.
- Haykin, Simon. *Redes Neurais: Princípios e Práticas*. 2001.
- Jafari, Masoumeh, et al. “Identification of Multivariable Nonlinear Systems in the Presence of Colored Noises Using Iterative Hierarchical Least Squares Algorithm.” *ISA Transactions*, vol. 53, no. 4, Elsevier, 2014, pp. 1243–52, doi:10.1016/j.isatra.2013.12.034.
- Johansson, K. H. “The Quadruple-Tank Process: A Multivariable Laboratory Process with an Adjustable Zero.” *IEEE Transactions on Control Systems Technology*, vol. 8, no. 3, May 2000, pp. 456–65, doi:10.1109/87.845876.
- Kennedy, J., and R. Eberhart. “Particle Swarm Optimization.” *Proceedings of ICNN'95 - International Conference on Neural Networks*, vol. 4, IEEE, 1995, doi:10.1109/ICNN.1995.488968.
- Lilly, John H. “Fuzzy Control and Identification.” *Fuzzy Control and Identification*, 2010, doi:10.1002/9780470874240.

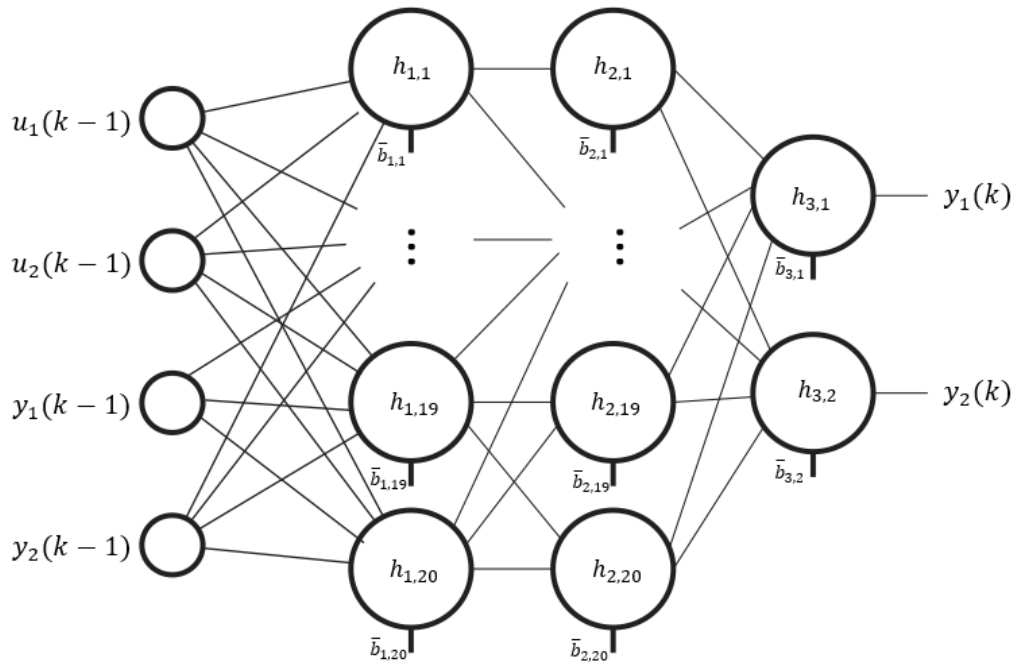
- Lovera, Marco, et al. "Recursive Subspace Identification of Linear and Non-Linear Wiener State-Space Models." *Automatica*, vol. 36, no. 11, Nov. 2000, pp. 1639–50, doi:10.1016/S0005-1098(00)00103-5.
- M. Thomas, F. Allgower, M. Morari. *Block-Oriented Nonlinear System Identification*. Vol. 404, 2010, doi:10.1007/978-1-84996-513-2.
- McAvoy, Thomas J., et al. "Dynamics of PH in Controlled Stirred Tank Reactor." *Industrial and Engineering Chemistry Process Design and Development*, vol. 11, no. 1, 1972, pp. 68–70, doi:10.1021/i260041a013.
- Millan, H. A., and E. Barrios. "Identification of a Nonlinear MIMO System Using Fuzzy and Parametric Models." *2016 IEEE Colombian Conference on Robotics and Automation (CCRA)*, no. 6, IEEE, 2016, pp. 1–6, doi:10.1109/CCRA.2016.7811410.
- Mujumdar, Arun S. "Handbook of Industrial Drying." *Industry*, 2014.
- Narendra, K., and P. Gallman. "An Iterative Method for the Identification of Nonlinear Systems Using a Hammerstein Model." *IEEE Transactions on Automatic Control*, vol. 11, no. 3, July 1966, pp. 546–50, doi:10.1109/TAC.1966.1098387.
- Negash, Berihun M., et al. "Conceptual Framework for Using System Identification in Reservoir Production Forecasting." *Procedia Engineering*, vol. 148, The Author(s), 2016, pp. 878–86, doi:10.1016/j.proeng.2016.06.479.
- Nelles, Oliver. "Nonlinear System Identification: From Classical Approaches to Neural Networks and Fuzzy Models." *Book*, 2001, doi:10.1007/978-3-662-04323-3.
- Paduart, J., et al. "NONLINEAR STATE SPACE MODELLING OF MULTIVARIABLE SYSTEMS." *IFAC Proceedings Volumes*, vol. 39, no. 1, IFAC, 2006, pp. 565–69, doi:10.3182/20060329-3-AU-2901.00086.
- Parikh, Nishant. "Partial Identification Framework for the Identification of Multi-Input Multi- Partial Identification." *International Scientific Journal on Science Engineering & Technology*, vol. 19, no. September, 2016.
- Passino, Kevin, and Stephen Yurkovich. "Fuzzy Control." *Fuzzy Control*, 1998.
- Paula, Marcus Vinicius de. *Identificação de Modelos de Hammerstein e Wiener Para Sistemas Não Lineares Multivariáveis via Métodos de Subespaços*. UFMG, 2016.
- Pontes, Tatiana Gomes de. *AVALIAÇÃO DO SISTEMA DE SECAGEM DO AÇÚCAR DA USINA MONTE ALEGRE*. Universidade Federal da Paraíba, 2014.
- Rodrigues, Marconi Câmara. *Identificação Fuzzy-Multimodelos Para Sistemas Não Lineares*. 2010.

- Sanandaji, Borhan Molazem, et al. "Multivariable GA-Based Identification of TS Fuzzy Models: MIMO Distillation Column Model Case Study." *2007 IEEE International Fuzzy Systems Conference*, IEEE, 2007, pp. 1–6, doi:10.1109/FUZZY.2007.4295567.
- Santos, Jessica A., and Ginalber L. O. Serra. "Recursive Identification Approach of Multivariable Nonlinear Dynamic Systems Based on Evolving Fuzzy Hammerstein Models." *2018 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, vol. 2018-July, IEEE, 2018, pp. 1–8, doi:10.1109/FUZZ-IEEE.2018.8491606.
- Shi, Y., and R. Eberhart. "A Modified Particle Swarm Optimizer." *1998 IEEE International Conference on Evolutionary Computation Proceedings. IEEE World Congress on Computational Intelligence (Cat. No.98TH8360)*, IEEE, 1998, pp. 69–73, doi:10.1109/ICEC.1998.699146.
- Silva, Arilson Galdino da, et al. "Modelo de Previsão Hidrológica Utilizando Redes Neurais Artificiais: Um Estudo de Caso Na Bacia Do Rio Xingu – Altamira-PA." *Revista Brasileira de Computação Aplicada*, vol. 10, no. 3, Oct. 2018, pp. 55–62, doi:10.5335/rbca.v10i3.8779.
- Sun, J., et al. "Particle Swarm Optimization: Classical and Quantum Perspectives." *CRC Press, Taylor & Francis Group LLC*, 2012, doi:10.1017/CBO9781107415324.004.
- Tangirala, Arun K. "Principles of System Identification: Theory and Practice." *Principles of System Identification: Theory and Practice*, 2015, doi:10.1201/9781315222509.
- Wang, Li-Xin. *A COURSE IN FUZZY SYSTEMS AND CONTROL*. Prentice Hall, 1997, <http://portal.acm.org/citation.cfm?id=248374&dl=>.
- Yan, Jun, et al. "Nonlinear State Space Modeling and System Identification for Electrohydraulic Control." *Mathematical Problems in Engineering*, vol. 2013, 2013, pp. 1–9, doi:10.1155/2013/973903.
- Zhu, Yucai. "Multivariable System Identification For Process Control." *Multivariable System Identification For Process Control*, no. October, Elsevier, 2001, doi:10.1016/B978-0-08-043985-3.X5000-5.
- Zhu, Yucai, and Ton Backx. *Identification of Multivariable Industrial Processes*. Vol. 404, Springer London, 1993, doi:10.1007/978-1-4471-2058-2.

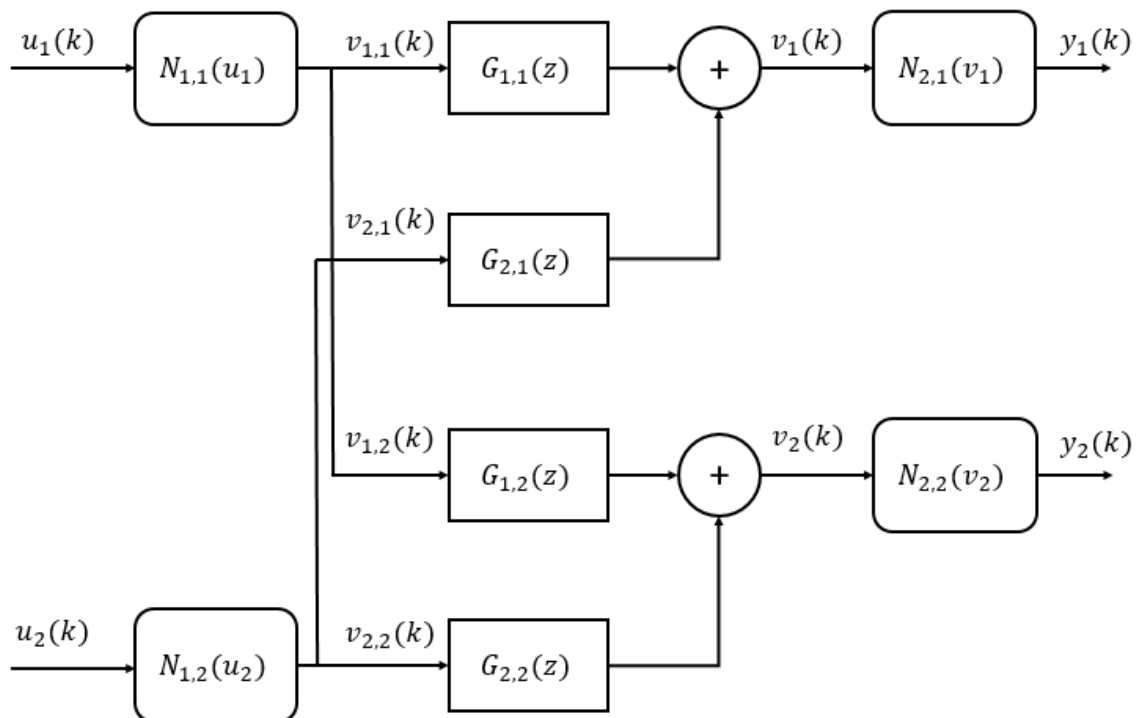
APÊNDICE I

Estrutura da RNA e do modelo Hammerstein-Wiener do 1º estudo de caso (Reator Contínuo de Tanque Agitado - CSTR).

- Estrutura RNA para o CSTR



- Estrutura modelo de Hammerstein-Wiener para o CSTR



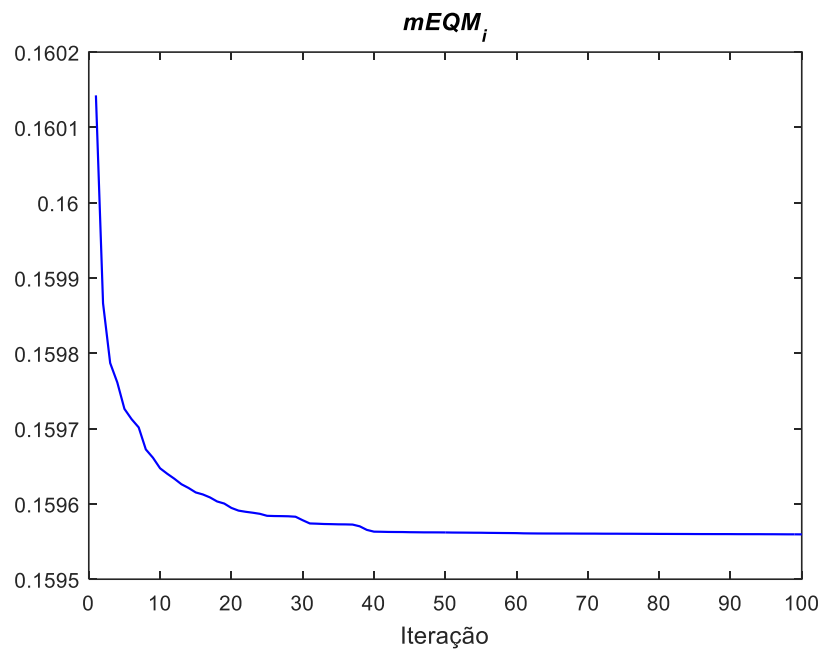
$$G_{1,1}(z) = \frac{z^{-2} - 0,95z^{-3}}{1 - 0,4922z^{-1} - 0,5078z^{-2}}$$

$$G_{1,2}(z) = \frac{z^{-2} + 0,8905z^{-3}}{1 - 0,026z^{-1} - 0,7665z^{-2}}$$

$$G_{2,1}(z) = \frac{2,306z^{-2} + z^{-3}}{1 + 0,2312z^{-1} - 0,0575z^{-2}}$$

$$G_{2,2}(z) = \frac{0,0504z^{-2} - 0,0629z^{-3}}{1 - 0,0957z^{-1} - 0,9z^{-2}}$$

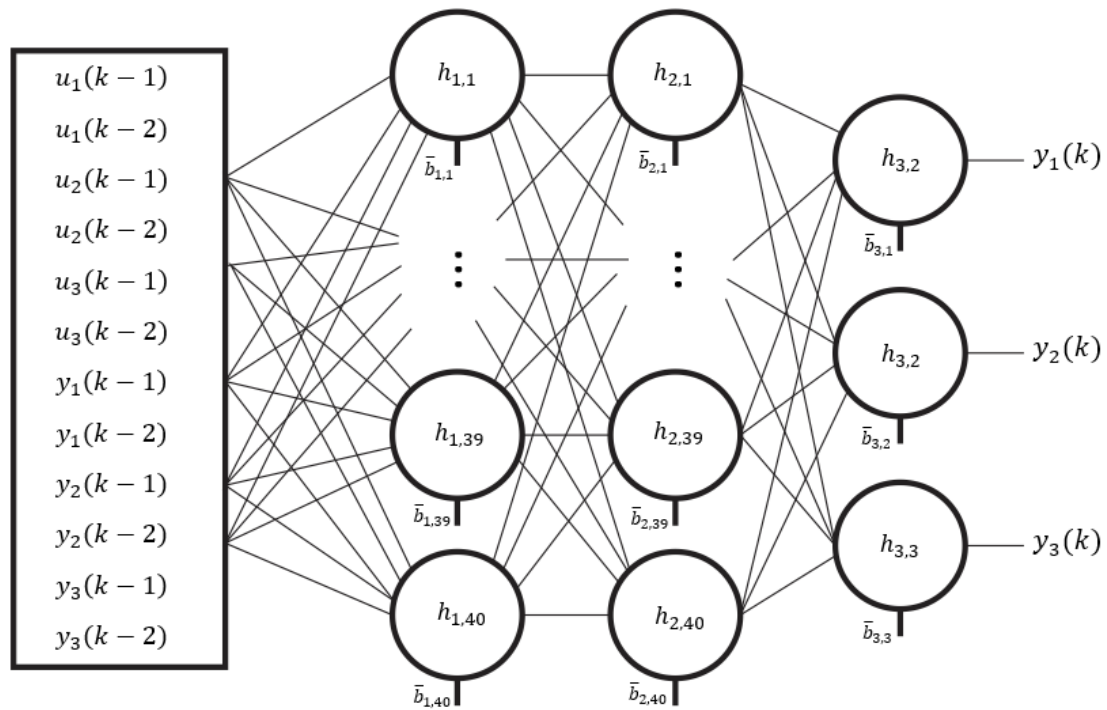
- Gráfico da função média da função objetivo do PSO para as 30 simulações da etapa de identificação da planta do CSTR.



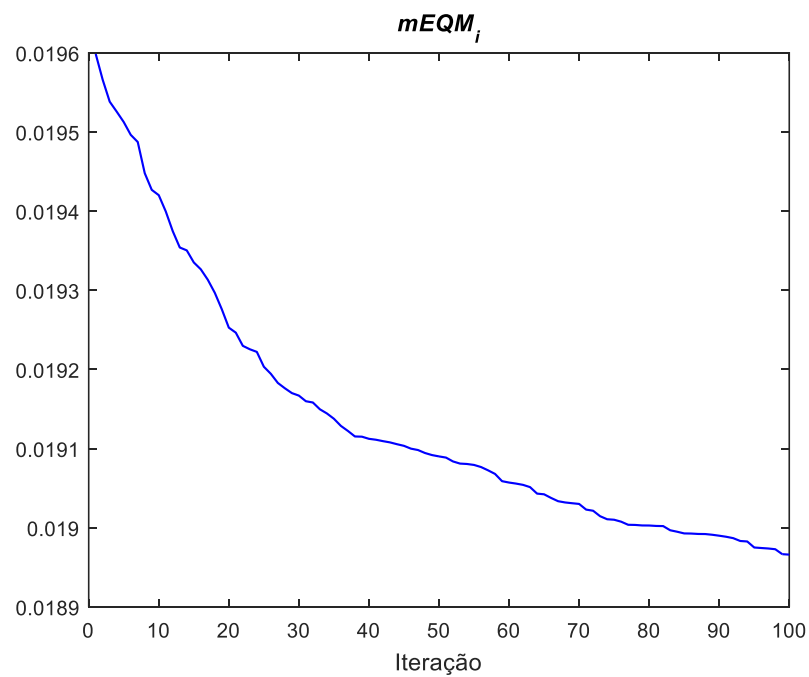
APÊNDICE II

Estrutura da RNA e do modelo Hammerstein-Wiener do 2º estudo de caso (Secador Industrial).

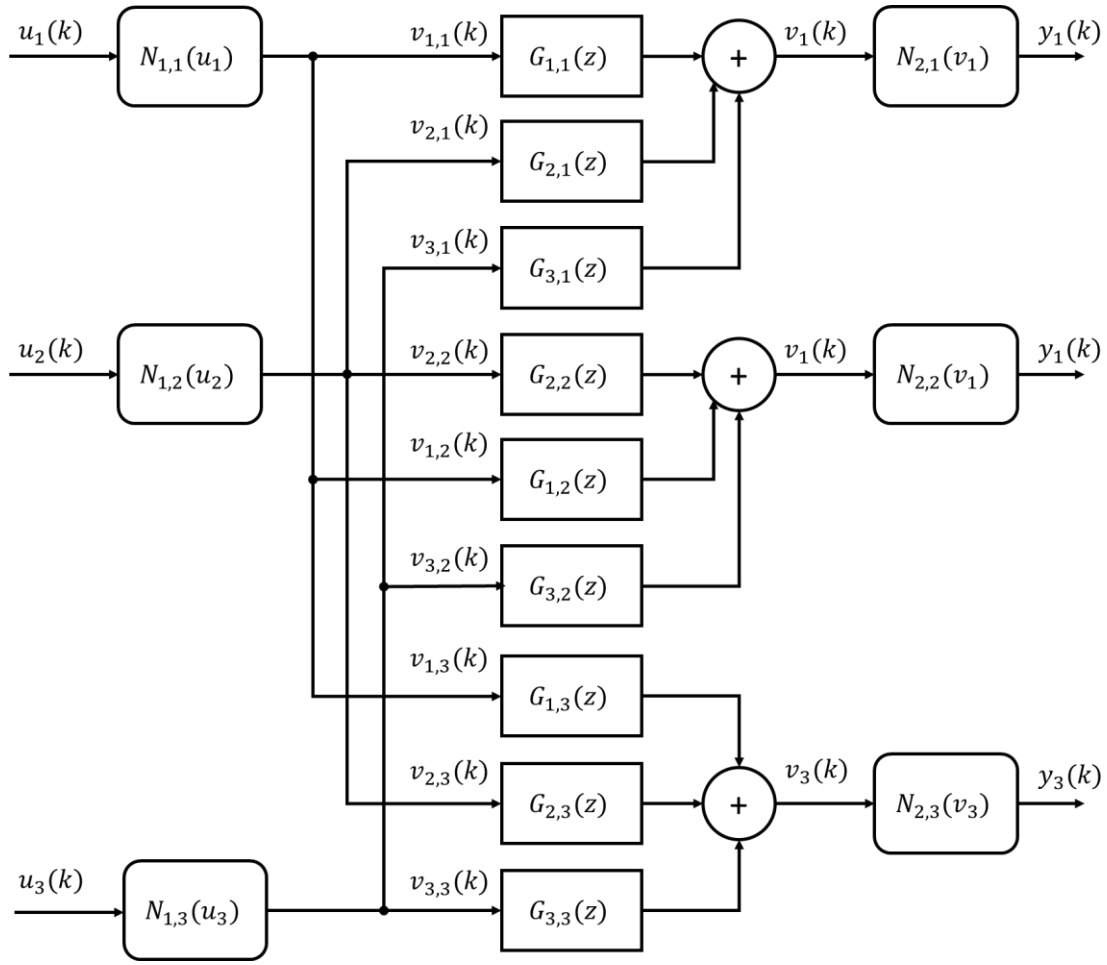
- Estrutura RNA para Secador Industrial



- Gráfico da função média da função objetivo do PSO para as 30 simulações da etapa de identificação da planta do Secador Industrial.



- Estrutura modelo de Hammerstein-Wiener para Secador Industrial



$$G_{1,1}(z) = \frac{-0,1496z^{-4} - 0,4873z^{-5} + z^{-6}}{1 - 2,235z^{-1} + 1,426z^{-2} - 0,01z^{-3} - 0,1787z^{-4}}$$

$$G_{2,1}(z) = \frac{-0,8042z^{-4} + 0,1928z^{-5} + z^{-6}}{1 - 1,118z^{-1} - 0,6464z^{-2} + 1,122z^{-3} - 0,3502z^{-4}}$$

$$G_{3,1}(z) = \frac{0,1156z^{-4} + z^{-5} - 0,0324z^{-6}}{1 - 3,055z^{-1} + 3,477z^{-2} - 1,745z^{-3} + 0,3229z^{-4}}$$

$$G_{1,2}(z) = \frac{-0,5035z^{-4} + z^{-5} - 0,4969z^{-6}}{1 - 2,271z^{-1} + 0,9053z^{-2} + 1,007z^{-3} - 0,6414z^{-4}}$$

$$G_{2,2}(z) = \frac{-0,0053z^{-4} + 0,0054z^{-5} - 0,0007z^{-6}}{1 - 1,347z^{-1} - 0,51z^{-2} + 1,225z^{-3} - 0,3575z^{-4}}$$

$$G_{3,2}(z) = \frac{(7,177z^{-4} - 9,141z^{-5} + 2,704z^{-6}) \cdot 10^{-5}}{1 - 1,956z^{-1} + 0,0718z^{-2} + 1,805z^{-3} - 0,9213z^{-4}}$$

$$G_{1,3}(z) = \frac{-0,9203z^{-4} + z^{-5} - 0,0822z^{-6}}{1 - 1,713z^{-1} + 0,3105z^{-2} + 0,547z^{-3} - 0,1398z^{-4}}$$

$$G_{2,3}(z) = \frac{-0,0188 z^{-4} + 0,0364 z^{-5} - 0,0174 z^{-6}}{1 - 2,575 z^{-1} + 2,83 z^{-2} - 1,918 z^{-3} + 0,664 z^{-4}}$$

$$G_{3,3}(z) = \frac{0,0001 z^{-4} - 0,0002 z^{-5} + 0,0001 z^{-6}}{1 - 2,655 z^{-1} + 3,102 z^{-2} - 1,869 z^{-3} + 0,4399 z^{-4}}$$