

UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE
CENTRO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA E DE COMPUTAÇÃO

André Felipe Oliveira de Azevedo Dantas

**ALGORITMO MODIFICADO DE PSO MATRICIAL
APLICADO A IDENTIFICAÇÃO DE SISTEMAS COM
ANÁLISE DE CONVERGÊNCIA**

Natal-RN
2015

André Felipe Oliveira de Azevedo Dantas

**ALGORITMO MODIFICADO DE PSO MATRICIAL
APLICADO A IDENTIFICAÇÃO DE SISTEMAS COM
ANÁLISE DE CONVERGÊNCIA**

Defesa de Doutorado apresentada ao Programa de Pós-Graduação em Engenharia Elétrica e de Computação PPgEEC, da Universidade Federal do Rio Grande do Norte, como parte dos requisitos para a obtenção do título de Doutor em Ciências. Área de concentração: Automação e Sistemas

Orientador: Prof. D.Sc. André Laurindo Maitelli

Natal-RN

2015

André Felipe Oliveira de Azevedo Dantas

**ALGORITMO MODIFICADO DE PSO MATRICIAL
APLICADO A IDENTIFICAÇÃO DE SISTEMAS COM
ANÁLISE DE CONVERGÊNCIA**

Defesa de Doutorado apresentada ao Programa de Pós-Graduação em Engenharia Elétrica e de Computação PPgEEC, da Universidade Federal do Rio Grande do Norte, como parte dos requisitos para a obtenção do título de Doutor em Ciências. Área de concentração: Automação e Sistemas

Aprovada em:

Prof. D.Sc. André Laurindo Maitelli
Universidade Federal do Rio Grande do Norte - UFRN
Orientador

Prof. D.Sc. Fábio Meneghetti Ugulino de Araújo
Universidade Federal do Rio Grande do Norte - UFRN
Membro

Prof. Ph.D. Allan de Medeiros Martins
Universidade Federal do Rio Grande do Norte - UFRN
Membro

Prof. D.Sc. Danielle Simone da Silva Casillo
Universidade Federal Rural do Semi-Árido - UFERSA
Externo

Prof. D.Sc. Vicente Delgado Moreira
PETROBRÁS
Externo

A minha esposa Amanda Danielle

AGRADECIMENTOS

Agradeço primeiramente a Deus.

Agradeço aos meus pais, Antônio e Alina.

À minha esposa Amanda Danielle.

Aos meus amigos Fábio, Grace, Thiago, Vanessa, Frankswell, Tiago, Gabrielle. Principalmente a Leandro e Daniel que me ajudaram imensamente cooperando com este trabalho.

Agradeço aos professores: André Laurindo Maitelli, Allan de Medeiros Martins, Anderson Luiz de Oliveira Cavalcanti, Carlos Eduardo Trabuco Dórea e Danielle Simone da Silva Casillo por revisarem, sugerirem correções e melhorias para este trabalho.

Gostaria de mencionar ao Laboratório de Automação em Petróleo pela oportunidade de desenvolver este trabalho.

RESUMO

Recentemente diversas técnicas de computação evolucionárias têm sido utilizadas em áreas como estimação de parâmetros de processos dinâmicos lineares e não lineares ou até sujeitos a incertezas. Isso motiva a utilização de algoritmos como o otimizador por nuvem de partículas (PSO) nas referidas áreas do conhecimento. Porém, pouco se sabe sobre a convergência desse algoritmo e, principalmente, as análises e estudos realizados têm se concentrado em resultados experimentais. Por isso, é objetivo deste trabalho propor uma nova estrutura para o PSO que permita analisar melhor a convergência do algoritmo de forma analítica. Para isso, o PSO é reestruturado para assumir uma forma matricial e reformulado como um sistema linear por partes. As partes serão analisadas de forma separada e será proposta a inserção de um fator de esquecimento que garante que a parte mais significativa deste sistema possua autovalores dentro do círculo de raio unitário. Também será realizada a análise da convergência do algoritmo como um todo, utilizando um critério de convergência quase certa, aplicável a sistemas chaveados. Na sequência, serão realizados testes experimentais de maneira a verificar o comportamento dos autovalores após a inserção do fator de esquecimento. Posteriormente, os algoritmos de identificação de parâmetros tradicionais serão combinados com o PSO matricial, de maneira a tornar os resultados da identificação tão bons ou melhores que a identificação apenas com o PSO ou, apenas com os algoritmos tradicionais. Os resultados mostram a convergência das partículas em uma região delimitada e que as funções obtidas após a combinação do algoritmo PSO matricial com os algoritmos convencionais, apresentam maior generalização para o sistema apresentado. As conclusões a que se chega é que a hibridização, apesar de limitar a busca por uma partícula mais apta do PSO, permite um desempenho mínimo para o algoritmo e ainda possibilita melhorar o resultado obtido com os algoritmos tradicionais, permitindo a representação do sistema aproximado em quantidades maiores de frequências.

Palavras-chave: PSO, Análise de convergência, Identificação.

A MODIFIED MATRICIAL PSO ALGORITHM APPLIED TO SYSTEM IDENTIFICATION WITH CONVERGENCE ANALYSIS

ABSTRACT

Recently, several evolutionary computation techniques have been used in research areas such as parameter estimation of linear and nonlinear dynamic processes. This motivates the use of algorithms such as the particle swarm optimization (PSO) in the aforementioned fields of knowledge. However, little is known about the convergence of this algorithm and mainly the analyzes and studies have focused on experimental results. Therefore, the objective of this work is to propose a structure for the PSO that better analyze the convergence of the algorithm analytically. For this, the PSO is restructured to assume a matrix form, reformulated as a piecewise linear system. There was a convergence analysis of the algorithm as a whole, using an almost sure convergence criterion applicable to switched systems. Subsequently, traditional parameter identification algorithms were combined with the matricial PSO (MPSO), so as to make the identification results as good as or better than identifying only using the PSO or only the traditional algorithms. The obtained functions, after the identification, using the matricial PSO algorithm combined with the conventional identification algorithms, presented a better generalization and proper identification. The conclusions reached were that the hybridization permits a minimum performance and also contributes to improve the results obtained with the traditional algorithms, allowing the system representation in a higher range of frequencies.

Keywords: PSO, Convergence Analysis, Identification.

LISTA DE TABELAS

	<u>Pág.</u>
1 Erro quadrático Médio (MSE) e Erro no Diagrama de Bode (E_{Bode}).	60
2 Erro Quadrático Médio (MSE) e Erro do Diagrama de Bode (E_{Bode}).	61

LISTA DE ABREVIATURAS E SIGLAS

ACO	–	Ant Colony Optimization
AGA	–	Adaptive Genetic Algorithms
AR	–	Auto Regressive
ARX	–	Auto-Regressive with Exogenous Input
ARMAX	–	Auto-Regressive Moving Average with Exogenous Input
BJ	–	Box-Jenkins
FFO	–	Fruit Fly Optimization
FIR	–	Finite Impulse Response
GA	–	Genetic Algorithms
MA	–	Moving Average
MQ	–	Mínimos Quadrados
MQE	–	Mínimos Quadrados Estendido
MQR	–	Mínimos Quadrados Recursivo
MQRCB	–	Mínimos Quadrados Recursivo com Compensação de Polarização
NARX	–	Nonlinear Auto-Regressive with Exogenous Input
NARMAX	–	Non-Linear Auto-Regressive Moving Average with Exogenous Input
OE	–	Output Error
PSO	–	Particle Swarm Optimization
PSOM	–	Particle Swarm Optimization Matricial

SUMÁRIO

	<u>Pág.</u>
1 INTRODUÇÃO	12
1.1 Objetivos Gerais	13
1.2 Objetivos Específicos	13
1.3 Revisão Bibliográfica do Estado da Arte	14
1.4 Principais Contribuições	15
1.5 Organização do Texto	16
2 OTIMIZAÇÃO POR NUVEM DE PARTÍCULAS (PSO)	17
2.1 Origem e Descrição	17
2.2 Algoritmo Base e Principal Variação	18
2.3 PSO Matricial Simplificado	20
2.4 PSO Matricial - Versão sem simplificações	22
2.4.1 Análise de Estabilidade do Algoritmo	24
2.4.2 Análise de Convergência do Erro Quadrático Iterativo	32
2.4.3 PSOM - Híbrido	37
3 IDENTIFICAÇÃO DE SISTEMAS	38
3.1 Introdução	38
3.2 Projeto de Sinais de Excitação	41
3.2.1 Escolha e Coleta de Sinais	41
3.3 Modelos Polinomiais Lineares	42
3.3.1 ARX	43
3.4 Dificuldades Relacionadas à Identificação	44
3.4.1 Presença de Ruídos	44
3.4.2 Presença de Dinâmicas Não Modeladas	45
3.5 Estimação de Parâmetros	45
3.5.1 Estimador de Mínimos Quadrados - MQ	45
3.5.2 Estimador de Mínimos Quadrados Recursivo - MQR	47
3.5.3 Estimador de Mínimos Quadrados Estendido - MQE	48
3.5.4 Estimador de Mínimos Quadrados Recursivo com Compensação de Polari- zação - BCRLS	49
4 METODOLOGIA, RESULTADOS E DISCUSSÕES	54
4.1 Exemplo Simulado 1	54
4.1.0.1 Exemplo Simulado 2	58

5 CONCLUSÕES	62
REFERÊNCIAS	63

LISTA DE FIGURAS

	<u>Pág.</u>
1 Gráficos com regiões de estabilidade no R^4 com os parâmetros não apresentados iguais a 1	31
2 Procedimento básico para identificação. Fonte: Adaptado de Ljung (1987) . .	40
3 Diagrama de Blocos do Modelo ARX. Fonte: Isermann e Munchhof (2011) . .	43
4 Diagrama de blocos de um processo sujeito a ruído. Fonte: Adaptado de Isermann e Munchhof (2011)	44
5 Validação Comparando Saída Real e Estimada	56
6 Validação Comparando os Diagramas de Bode	57

1 INTRODUÇÃO

Segundo Nelles (2001) os modelos de sistemas reais podem ser úteis para a análise, fornecendo conhecimento sobre determinado sistema. Além disso, técnicas avançadas para projeto e síntese de controladores, otimização, supervisão, detecção de falhas, e diagnóstico de componentes são também baseadas em modelos de processos.

Para modelar um sistema é possível utilizar técnicas de identificação de sistemas, que é o campo da modelagem matemática de sistemas (processos) em que uma de suas etapas é a obtenção de parâmetros, que são obtidos a partir de testes ou dados experimentais (ZHU, 2001). Uma das importantes características das técnicas de identificação de sistemas, segundo Aguirre (2007), é que pouca ou nenhuma informação prévia sobre o sistema é necessária. Devido a isso, em muitos casos, será preferível usar técnicas de identificação de sistemas para obter o seu modelo.

Alguns modelos bastante utilizados na identificação de sistemas lineares são os modelos OE (*Output Error*), ARX (*Autoregressive with eXogenous Input*) e, em sistemas não lineares, o modelo NARX *Nonlinear Autoregressive with eXogenous Input*. Estes modelos são utilizados para representar sistemas os mais variados. Os parâmetros de tais modelos podem ser encontrados utilizando algoritmos de identificação BCRLS (*Bias Compensation Recursive Least Square*) (SÖDERSTRÖM et al., 2005; DING et al., 1999; MAHATA, 2007; ZHANG; CUI, 2011; DING; DING, 2011), MQ (*Mínimos Quadrados*) (AGUIRRE, 2007), MQE (*Mínimos Quadrados Estendido*) (ETTALEB et al., 1998; YAM; CHOW, 1997; WENXIAO, 2007; COHEN; TUNINSKY, 1995) e PSO (*Particle Swarm Optimization*) (HUYNH; DUNNINGAN, 2010; NAJJARZADEH; AYATOLLAHI, 2008; TANG et al., 2008; KHOSLA et al., 2007). Os três primeiros algoritmos, BCRLS, MQ e MQE, são métodos relativamente clássicos e o último, a PSO, é baseado em heurísticas que ajudam o algoritmo a buscar o melhor conjunto de parâmetros para os modelos.

As técnicas determinísticas dominaram as pesquisas voltadas à otimização por várias décadas. Porém, a incapacidade de encontrar os máximos e mínimos globais para certas funções, adicionada à complexidade dessas técnicas levaram ao estudo de métodos estocásticos e algoritmos heurísticos (EMARA; FATTAH, 2004). Dentre as técnicas de otimização observadas aquelas aplicações baseadas em PSO tem sido bastante estudadas (NAJJARZADEH; AYATOLLAHI, 2008), o que motiva o seu estudo, uso e aplicação na referida área do conhecimento (Identificação de Sistemas).

Desde a concepção da PSO, os pesquisadores desenvolveram melhorias principalmente na dinâmica das partículas, e no próprio algoritmo (BHATTACHARYA et al., 2008).

Estudos voltados para a melhoria dos algoritmos baseados na PSO podem ser encontrados nas seguintes referências [Zeng G. and Jiang \(2010\)](#) e [Sharifi et al. \(2012\)](#). Apesar de muitas evidências empíricas terem sido acumuladas, mostrando que a PSO funciona adequadamente, ou seja, é uma excelente técnica de otimização, até agora, poucas pesquisas são voltadas para a maneira como ele se comporta e a sua convergência, especificamente.

Portanto, devido a intensidade da pesquisa em torno da computação evolucionária, principalmente a PSO, aplicada à área de estimação de parâmetros ([Huynh e Dunnigan \(2010\)](#), [Najjarzadeh e Ayatollahi \(2008\)](#), [Tang et al. \(2008\)](#), [Khosla et al. \(2007\)](#), [Arruda et al. \(2011\)](#)) foi despertado o interesse em investigar tais aplicações, de maneira a garantir melhor desempenho do algoritmo. Para tanto, é apresentada neste trabalho uma análise do algoritmo PSO envolvendo uma mudança em sua estrutura de cálculo, visando permitir que a otimização realizada pela PSO retorne um resultado tão bom ou melhor que os algoritmos tradicionais para os casos analisados.

1.1 Objetivos Gerais

Dentro do contexto apresentado é objetivo geral deste trabalho propor uma nova estrutura de cálculo para a PSO, organizando-a no formato matricial como um sistema linear por partes e analisando a sua convergência, e aplicar à identificação de sistemas dinâmicos lineares simulados.

1.2 Objetivos Específicos

São objetivos específicos deste trabalho:

- Implementar os algoritmos de identificação utilizando MQ, MQE, BCRLS, PSO;
- Organizar o PSO nas formas matricial (PSOM) simplificada (pois serão realizadas simplificações no cálculo do algoritmo) e completa;
- Realizar uma análise de convergência das partículas da PSO matricial representando-a na forma de sistemas lineares por partes e analisando os autovalores das partes separadamente e do sistema completo (através do critério de convergência quase certa);
- Realizar uma análise de convergência do algoritmo para a solução otimizada utilizando o erro quadrático iterativo;
- Propor modificações ajustando a convergência das partículas da PSO matricial;
- Híbrido o algoritmo da PSOM com os MQE e BCRLS, visando obter desempenhos melhores quando os sistemas identificados estão sujeitos a dinâmicas não modeladas;

- Apresentar resultados e testes que mostrem a convergência do algoritmo da PSOM de maneira individual
- Apresentar resultados experimentais que comprovem o melhor desempenho em média da PSOM híbrido.

1.3 Revisão Bibliográfica do Estado da Arte

As técnicas de otimização determinísticas dominaram esta área por várias décadas. Porém, a incapacidade de encontrar o mínimo global de certas funções, somado à complexidade destas técnicas, motivaram o estudo de métodos estocásticos e algoritmos heurísticos (EMARA; FATTAH, 2004).

Recentemente, diversas técnicas de computação evolucionárias, tais como GA (Algoritmos genéticos), AGA (Algoritmos Genéticos Adaptativo), PSO, ACO (*Ant Colony Optimization*), têm emergido como técnicas de otimização aplicáveis a áreas como estimação de parâmetros (utilizando GA Huang et al. (2002), Bishop e Richards (1990), Kampisios et al. (2008), Chen et al. (2008), utilizando AGA Chen et al. (2008), Zhou e Cheng (2004), utilizando ACO Chen et al. (2008), utilizando PSO Huynh e Dunnigan (2010), Najjarzadeh e Ayatollahi (2008), Tang et al. (2008), Khosla et al. (2007)). Além dessas aplicações, podemos utilizar as técnicas de computação evolucionárias em áreas como identificação (ARRUDA et al., 2011), sintonia (DANTAS et al., 2012) e controle (SONG et al., 2007). Segundo Najjarzadeh e Ayatollahi (2008) sabe-se que as aplicações em GA são de difícil implementação devido à complexidade no código e por possuir baixa taxa de convergência em relação a algoritmos como a PSO. Isso motiva a utilização desse algoritmo (PSO) com o intuito de aplicá-lo nas referidas áreas do conhecimento.

Desde a concepção da PSO as pesquisas que o englobam desenvolvem principalmente melhorias nas dinâmicas das partículas e do algoritmo em si (BHATTACHARYA et al., 2008). Pesquisas sobre melhorias nos algoritmos baseados na PSO podem ser encontradas em Zhan et al. (2009), Ueno et al. (2005), Zeng G. end Jiang (2010), Lu et al. (2008), Sharifi et al. (2012), Chen e Li (2007). Apesar de muitas evidências empíricas acumuladas mostrarem que o PSO funciona, ou seja, é uma excelente ferramenta de otimização, até então, pouco se pesquisou sobre o seu funcionamento e convergência (CLERC; KENNEDY, 2002). Por isso, a análise de estabilidade da PSO vem despertando o interesse dos pesquisadores.

Alguns trabalhos na literatura buscaram analisar a convergência e estabilidade da PSO de várias maneiras. Dentre estas é possível citar a primeira análise de estabilidade da PSO proposta por Clerc e Kennedy (2002), a análise de estabilidade utilizando o conceito de estabilidade absoluta proposta por Kadiramanathan et al. (2006), a análise de convergência do PSO utilizando funções multi-objetivo proposta por Chakraborty et al. (2010) e a análise de estabilidade não linear variante no tempo do PSO proposta por Fan et al. (2010).

1.4 Principais Contribuições

Neste trabalho é proposto utilizar a PSO como técnica de otimização aplicada a identificação de sistemas. Atualmente, existem várias técnicas determinísticas capazes de realizar essa operação. Porém, elas falham quando a modelagem não é adequada, ou seja, quando existe no processo dinâmicas que não foram modeladas e ruídos de medição. Para aplicar a PSO na identificação de sistemas é necessário que o algoritmo tenha a capacidade de encontrar soluções com menores erros e mais genéricas que as encontradas pelos algoritmos tradicionais. Para que esse objetivo fosse alcançado, algumas análises e formulação precisaram ser realizadas. Nesse contexto, as principais contribuições deste trabalho são:

- A síntese do algoritmo da PSO no formato matricial (síntese simplificada e exata). Essa síntese permite realizar a análise da PSO, de convergência das partículas a partir da análise de seus autovalores;
- A síntese permite organizar o sistema como sendo linear por partes. Essa organização possibilita a utilização de um teorema para analisar a convergência das partículas utilizando o critério estatístico de “convergência quase certa”;
- Outra contribuição advinda da organização matricial e da síntese na forma de sistemas lineares por partes é a possibilidade de analisar a convergência do erro quadrático iterativo, que mostra as condições necessárias para que o erro quadrático de identificação da PSOM a cada iteração diminua;
- Outra contribuição deste trabalho é a hibridização da PSO com os algoritmos MQE (Mínimos Quadrados Estendido) e BCRLS (Bias Compensation Recursive Least Square), pois ela permite melhorar os resultados iniciais obtidos com a PSOM, apesar de diminuir a variabilidade da busca. Adicionalmente a isso, realizar essa hibridização permite que seja possível ter uma identificação tão boa quanto um algoritmo determinístico, e, em caso de uma modelagem não exata, diminuir o erro de identificação, utilizando a PSOM.

A análise de estabilidade apresentada neste trabalho se diferencia das demais, pois é formulada uma nova estrutura para o cálculo das novas posições, velocidades, melhores partículas locais e global. Esta formulação consiste em reorganizar o algoritmo para que estes cálculos sejam efetuados como uma multiplicação matricial. Desta maneira, é possível analisar o algoritmo da perspectiva de um sistema dinâmico linear variante no tempo. Neste caso, o sistema é formulado como um sistema linear por partes, cujas partes são analisadas separadamente, utilizando o conceito de autovalores, e conjuntamente, utilizando o conceito de convergência quase certa, cujo teorema é apresentado em [HANLON et al. \(2011\)](#).

1.5 Organização do Texto

O trabalho está organizado da seguinte maneira: No capítulo 2 é mostrada a origem da PSO. Também é apresentada uma revisão bibliográfica com várias aplicações da otimização a aplicada em diversas áreas, tais como a identificação, a sintonia e o controle. Além disso, no capítulo 2 serão apresentadas a síntese da PSOM (versão simplificada e versão completa), juntamente com os algoritmos, a análise de convergência das partículas da PSOM utilizando o critério dos autovalores e da convergência quase certa, e a análise da convergência do erro quadrático iterativo de otimização. Ainda no capítulo 2, será apresentada a estrutura que permitirá a inicialização das partículas, ou seja, será mostrado como uma possível hibridização com outros algoritmos poderia acontecer. No capítulo 3 é realizada uma revisão bibliográfica da estrutura principal de identificação e os principais algoritmos. Neste capítulo, serão apresentados os algoritmos MQ, MQE, BCRLS, suas principais características e problemas relacionados à identificação. No capítulo seguinte, 4, é formalizada a sequência de testes de identificação, quais os parâmetros inicializam os algoritmos e quais testes serão realizados. Além disso, são discutidos os resultados obtidos nesses testes, como eles comprovam a teoria apresentada e quais os problemas envolvidos na utilização dos algoritmos. Finalmente, no capítulo 5, são realizadas as considerações finais do trabalho, conclusões acerca dos testes do novo algoritmo de PSO e proposições para trabalhos futuros.

2 OTIMIZAÇÃO POR NUVEM DE PARTÍCULAS (PSO)

No presente capítulo serão apresentados o algoritmo PSO tradicional e sua principal variação. Em sequência, a PSO será reformulada na forma matricial (PSOM) simplificada e exata, ambos como um sistema linear por partes, em que as partes do algoritmo serão avaliadas tanto de maneira separada como conjunta. Em seguida, serão apresentados os algoritmos, ou seja, como utilizar as formulações matriciais. Posteriormente, será verificada a necessidade de uma modificação de maneira a manter os autovalores dentro do círculo de raio unitário. Adicionalmente às análises realizadas, será apresentada uma análise baseada no decrescimento do erro quadrático iterativo, visando mostrar condições necessárias para que ocorra a convergência do erro de otimização em direção à partícula correta. Após realizadas as modificações no algoritmo será proposta uma adequação na inicialização das partículas para que a PSOM possa ser hibridizado com algoritmos de identificação.

2.1 Origem e Descrição

Entre as décadas de 1970 e 1990, cientistas como [Heppner e Grenander \(1990\)](#), [Reynolds \(2007\)](#) e [Millonas \(1994\)](#) debruçavam-se sobre o estudo do comportamento psicológico e social de determinadas espécies. Tais pesquisadores estavam fascinados com as interações entre bandos de aves e cardumes de peixes em suas atividades cotidianas ([KENNEDY; EBERHART, 1995](#)). Tais grupos de animais, ao procurar por comida ou ao fugir de predadores, possuem um tipo de consciência coletiva. Cada elemento do grupo aprende com o comportamento dos seus demais componentes e tendo em vista esse comportamento, diversos pesquisadores buscaram sintetizar essas interações por intermédio de modelos matemáticos. Nesse contexto, o algoritmo PSO foi desenvolvido por [Kennedy e Eberhart \(1995\)](#), e foi posteriormente modificado por [Shi e Eberhart \(1998\)](#).

O PSO é similar a outras técnicas evolutivas, como por exemplo, algoritmo genético ([NAJJARZADEH; AYATOLLAHI, 2008](#)). Porém, na PSO, cada indivíduo ou partícula possui uma velocidade de adaptação (mudança de posição), que é alterada à medida que a partícula se move dentro do espaço de busca. Além disso, cada indivíduo tem memória sobre a melhor posição pela qual passou. Seu deslocamento é direcionado para a melhor posição já visitada e para a melhor partícula de uma vizinhança. O PSO é eficaz na otimização de problemas não contínuos em diversas áreas de conhecimento ([SHI; EBERHART, 1998](#)). Atualmente vem sendo aplicado a problemas de otimização, além dos já citados, em projetos de antenas ([ROBINSON et al., 1994](#)), planejamento de trajetórias de robôs ([SIERAKOWSKI C. A.; COELHO, 2005](#)) e sistemas de distribuição de energia elétrica ([FUKUYAMA; YOSHIDA,](#)

2001).

Existe uma teoria muito simples que está por trás do PSO. Nessa teoria, cada indivíduo de uma população possui sua própria experiência quantizada na forma de uma função objetivo. Para incorporar às partículas um comportamento social, cada indivíduo possui conhecimento sobre como seus vizinhos se comportam. Essas duas informações correspondem à aprendizagem individual (cognitiva) e à transmissão cultural (social), respectivamente. Portanto, cada decisão estará associado a seu desempenho passado e de alguns de seus vizinhos (SERAPIÃO, 2009).

Kennedy et al. (2001) utilizaram três princípios para resumir o processo de adaptação incorporada ao algoritmo de PSO:

- Avaliar - é a tendência de classificar se algo é positivo ou negativo, atrativo ou repulsivo, característica natural dos seres vivos;
- Comparar - é a capacidade de julgar algo utilizando outros indivíduos como referência comparativa;
- Imitar - é a habilidade que apenas os seres humanos e alguns pássaros são capazes de utilizar. A imitação incorporada ao algoritmo não é apenas fazer algo porque outro indivíduo reagiu de tal forma mas é a capacidade de entender o propósito ou o caminho percorrido e utilizá-lo quando apropriado.

Com isso é possível entender que, assim como em outras abordagens de inteligência coletiva, o algoritmo PSO incorpora capacidades a uma população de indivíduos de interagir entre si e com o meio ambiente. Com base nos conceitos de avaliação, comparação e imitação, esse algoritmo é capaz de, assim como os indivíduos são capazes de lidar com um número de possíveis situações que o ambiente lhes apresenta, resolver problemas de otimização em diversas áreas do conhecimento (SERAPIÃO, 2009).

2.2 Algoritmo Base e Principal Variação

Como já referido, os algoritmos de PSO empregam os princípios de avaliar, comparar e imitar (KENNEDY et al., 2001). Esses princípios são modelados na forma de aprendizagem individual (cognitiva) e a transmissão cultural (social), ideias importantes no processo de decisão. A transmissão cultural (P_G) conecta conceitualmente todos os membros de uma população entre si. Como consequência, o comportamento de cada partícula é influenciado pelo comportamento de todas as outras. A aprendizagem individual (P_B^i) cria uma vizinhança para cada indivíduo composta por ele próprio e seus vizinhos mais próximos. Ambas as métricas são respostas de uma função de avaliação J_{PSO} , também chamada de função objetivo ou de aptidão (*fitness*), que corresponde aos valores, capacidade, ou qualidade de cada indivíduo na solução do problema (KENNEDY et al., 2001).

Seguindo as ideias das transmissões cultural e individual, e aplicando-as a um modelo matemático, podemos entender que uma partícula $X^i(k)$, em um determinado passo k , irá se mover segundo uma função que relaciona a posição atual da partícula $X^i(k)$ no instante k e a velocidade $V^i(k+1)$ conforme mostrado na equação 2.1.

$$X^i(k+1) = X^i(k) + V^i(k+1) \quad (2.1)$$

Sendo que o índice i representa a i -ésima partícula ou indivíduo.

A velocidade $V^i(k+1)$, por sua vez, é uma função formada por sua experiência anterior $V^i(k)$, a distância entre a posição atual da partícula $X^i(k)$ e a melhor experiência (função de avaliação J_{PSO} com menor valor, para função de custo, ou maior valor, para função de maximização) que essa partícula possuiu $P_B^i(k)$ e a distância entre a posição atual da partícula $X^i(k)$ e a melhor experiência $P_G(k)$ que todas as partículas já possuíram em todos os k eventos passados. A relação descrita pode ser melhor entendida visualizando a equação 2.2.

$$V^i(k+1) = V^i(k) + \phi_1(P_B^i(k) - X^i(k)) + \phi_2(P_G(k) - X^i(k)) \quad (2.2)$$

Em que ϕ_1 e ϕ_2 são números aleatórios positivos com distribuição uniforme, geralmente denominados respectivamente como componentes “cognitivo” e “social”.

A partir dessas duas funções é possível descrever o algoritmo básico do PSO (SERAPIÃO, 2009) conforme mostrado:

Algoritmo 1 PSO - Algoritmo Base

- 1: Criar populações de agentes distribuídos uniformemente ao longo dos planos X (as i partículas de $X^i(0)$), V (as i partículas de $V^i(0)$) e P_B (as i partículas de $P_B^i(0)$);
 - 2: Avaliar a posição de cada partícula ($X^i(0)$) de acordo com a função (J_{PSO}) escolhida que se deseja minimizar ou maximizar;
 - 3: Se $J_{PSO}(X^i(k))$ for “melhor” que $J_{PSO}(P_B^i(k))$, atribuir a $P_B^i(k+1)$ o valor contido em $X^i(k)$, assim como a $J_{PSO}(P_B^i(k+1))$ o valor contido em $J_{PSO}(X^i(k))$;
 - 4: Encontrar utilizando $J_{PSO}(P_B^i(k))$ a “melhor” partícula colocando-a em $P_G(k)$ associando a sua avaliação a $J_{PSO}(P_G(k))$;
 - 5: Atualizar as i velocidades de acordo com a equação 2.2;
 - 6: Mover as partículas para novas posições segundo a equação 2.1
 - 7: Se o critério de avaliação não for atingido volte ao item 1. Caso contrário finalizar o algoritmo.
-

A maioria das modificações realizadas no algoritmo de PSO são variantes que, de alguma maneira, interferem direta ou indiretamente na lei de adaptação da velocidade das partículas. Como utilizado por Shi e Eberhart (1998) que insere um coeficiente de inércia, Zheng et al. (2012) que realiza uma mudança linear e não linear no coeficiente de inércia, Malik et al. (2007) que realiza atualizações no coeficiente de inércia utilizando funções

sigmoidais e [Tang et al. \(2008\)](#) que realiza uma busca local por uma melhor partícula global (P_G).

Como mostrado, muitas modificações dentre as que são realizadas na velocidade das partículas inserem coeficientes de inércia dos mais variados (estáticos e dinâmicos). Uma das modificações mais tradicionais é a desenvolvida por [Shi e Eberhart \(1998\)](#) que modificou a PSO original introduzindo um coeficiente de inércia (ω) na equação 2.2. A equação modificada pode ser visualizada na equação 2.3

$$V^i(k+1) = \omega V^i(k) + \phi_1(P_B^i(k) - X^i(k)) + \phi_2(P_G(k) - X^i(k)) \quad (2.3)$$

De acordo com testes numéricos, realizados por [Shi e Eberhart \(1998\)](#), [Shi e Eberhart \(1999\)](#), ao adotar o coeficiente ω a performance do PSO aumenta consideravelmente de tal forma que a junção das equações 2.3 e 2.1 é sempre considerada como o PSO padrão por muitos pesquisadores.

2.3 PSO Matricial Simplificado

Apesar de muitas evidências empíricas acumuladas mostrarem que o PSO é uma excelente ferramenta de otimização, até então, pouco se pesquisou sobre o seu funcionamento e convergência ([CLERC; KENNEDY, 2002](#)). Com base nessa motivação o algoritmo para a atualização das partículas, velocidade, melhor local e melhor global, é reorganizado em uma forma que a análise de convergência seja possível. A forma encontrada para a reescrita do algoritmo foi colocando-o na forma um sistema dinâmico discreto.

Para que seja possível organizar o PSO na forma matricial algumas definições são necessárias. Primeiramente é necessário reescrever a condição de atualização das melhores partículas locais conforme as equações 2.4 e 2.5.

$$\left\{ \begin{array}{l} Se(J_{PSO}(X^i(k)) < J_{PSO}(P_B^i(k))) \Rightarrow Verdadeiro = 1 \\ s_1^i(k) = \frac{-sign(J_{PSO}(X^i(k)) - J_{PSO}(P_B^i(k))) + 1}{2} \end{array} \right. \quad (2.4)$$

E,

$$\left\{ \begin{array}{l} Se(J_{PSO}(X^i(k)) < J_{PSO}(P_B^i(k))) \Rightarrow Falso = 0 \\ 1 - s_1^i(k) = \frac{sign(J_{PSO}(X^i(k)) - J_{PSO}(P_B^i(k))) + 1}{2} \end{array} \right. \quad (2.5)$$

Sendo que $J_{PSO}(L)$ é a função custo, que é uma função de L , e que são as melhores partículas locais, ou as melhores experiências locais das partículas. A função *sign*, ou sinal, é representada de acordo com a equação 2.6.

$$\text{sign}(x) = \begin{cases} 1, & \text{If } x \geq 0 \\ -1, & \text{If } x < 0 \end{cases} \quad (2.6)$$

Isso significa que podemos escrever as condições 2.4 e 2.5 na forma da equação 2.7.

$$P_B^i(k+1) = X^i(k)s_1^i(k) + P_B^i(k)(1 - s_1^i(k)) \quad (2.7)$$

Tendo reescrito a condição de atualização das melhores partículas locais, a segunda condição para construir a variação matricial do PSO é reescrever a condição de atualização da melhor partícula global. Para isso, foi buscada uma solução, em princípio baseada na simplificação desenvolvida por (MENDES et al., 2004). Na simplificação realizada pelos autores utiliza-se um conceito de média entre as melhores partículas locais e globais com o intuito de melhorar o algoritmo. Percebendo a possibilidade de utilização de tal conceito para a sintetização de uma relação semelhante a da equação 2.7, foi definido que a atualização da partícula global seria realizada a partir da média das partículas locais, conforme a equação 2.8.

$$P_m(k) = \frac{\sum_{i=1}^{n_I} P_B^i(k)}{n_I} \quad (2.8)$$

Sendo que n_I é o número de partículas, ou seja, número de indivíduos da população. Com isso, é possível a partir da condição 2.9 obter a equação 2.10:

$$\begin{cases} se(J_{PSO}(P_m(k)) < J_{PSO}(P_G(k))) \Rightarrow Verdadeiro = 1; & P_G(k+1) = P_m(k) \\ se(J_{PSO}(P_m(k)) < J_{PSO}(P_G(k))) \Rightarrow Falso = 0; & P_G(k+1) = P_G(k) \end{cases} \quad (2.9)$$

$$P_G(k+1) = P_m(k)s_2(k) + P_G(k)(1 - s_2(k)) \quad (2.10)$$

Definindo $T(k)$ como o vetor “aumentado”, ou vetor de partículas $1d$ (significando que $i = 1$) na forma da equação 2.11:

$$T(k) = \begin{bmatrix} X^i(k) \\ V^i(k) \\ P_B^i(k) \\ P_G(k) \end{bmatrix} \quad (2.11)$$

É possível, utilizando as equações 2.1, 2.3, 2.7, 2.8 e 2.10, e 2.11, escrever uma matriz “A” dada na equação 2.12, de forma a atualizar o vetor $T(k+1)$ a partir do vetor $T(k)$ conforme a equação 2.13.

$$A(k) = \begin{bmatrix} 1 - (\phi_1 + \phi_2) & \omega & \phi_1 & \phi_2 \\ -(\phi_1 + \phi_2) & \omega & \phi_1 & \phi_2 \\ s_1^i(k) & 0 & 1 - s_1^i(k) & 0 \\ 0 & 0 & \frac{s_2(k)}{n_I} & 1 - s_2(k) \end{bmatrix} \quad (2.12)$$

$$T(k+1) = A(k) T(k) \quad (2.13)$$

A solução otimizada com PSO, ou seja, a melhor partícula da iteração k pode ser obtida a partir da equação 2.14.

$$S = [0\ 0\ 0\ 1]T(k) = P_G(K) \quad (2.14)$$

Devido a simplicidade da equação 2.13 tanto o algoritmo do PSO se torna simplificado quanto é possível realizar a análise dos autovalores de forma a verificar a convergência das partículas em certas condições.

O algoritmo da PSOM, versão simplificada é descrito conforme mostrado:

Algoritmo 2 PSOM - Versão Simplificada

- 1: Criar populações de agentes distribuídos uniformemente ao longo dos planos X , V , e P_B ;
 - 2: Montar a matriz “A”, conforme equação 2.12;
 - 3: Atualizar as partículas conforme equação 2.13;
 - 4: Se o critério de avaliação não for atingido volte ao passo 2. Caso contrário finalizar o algoritmo.
-

2.4 PSO Matricial - Versão sem simplificações

A diferença básica entre a variação matricial exata e a variação simplificada está na forma como a melhor partícula global $P_G(k)$ é atualizada. Isso significa que a organização no formato matricial segue a mesma sequência das equações 2.4 até a 2.7 e na sequência será apresentada a síntese da equação que calcula a atualização de $P_G(k)$, desenvolvida neste trabalho.

Tendo reescrito a condição de atualização das melhores partículas locais (equações de 2.4 até 2.7), a segunda condição para construir a variação matricial do PSO é reescrever a condição de atualização da melhor partícula global. Para que isso seja possível serão definidas quatro novas variáveis conforme as equações 2.15, 2.16, 2.17, 2.18 que são:

$$E_{P_G}^i(k) = J_{PSO}(P_G) - J_{PSO}(P_B^i(k)) \quad (2.15)$$

$$ME_{P_G}^i(k) = \|J_{PSO}(P_B(k))\|_\infty - \|J_{PSO}(P_B^i(k))\|_\infty + \|J_{PSO}(P_B(k))\|_\infty \quad (2.16)$$

$$G_D^i(k) = \frac{[s(E_{P_G}^i(k)) + s(ME_{P_G}^i(k))] + [s(E_{P_G}^i(k)) s(ME_{P_G}^i(k))] + 1}{4} \quad (2.17)$$

$$s_{DG}(k) = \frac{s_3(\sum_{i=1}^N G_D^i(k)) + 1}{2} \sum_{i=1}^N G_D^i(k) + \frac{-s_3(\sum_{i=1}^N G_D^i(k)) + 1}{2} \quad (2.18)$$

Em que $s_3(x)$ é a função sinal, definida como:

$$s_3(x) = \begin{cases} 1, & \text{se } x > 0 \\ -1, & \text{se } x \leq 0 \end{cases} \quad (2.19)$$

e N é o número de partículas.

Utilizando as formulações apresentadas é possível obter a equação que atualiza as melhores partículas globais conforme 2.20.

$$P_G(k+1) = \frac{G_D^i(k)}{s_{DG}(k)} P_B^i(k) + (1 - \sum_{i=1}^N \frac{G_D^i(k)}{s_{DG}(k)}) P_G(k) \quad (2.20)$$

Utilizando as formulações é possível reescrever a matriz $A(k)$ para o PSOM exato na forma da equação 2.21.

$$A(k) = \begin{bmatrix} 1 - (\phi_1 + \phi_2) & \omega & \phi_1 & \phi_2 \\ -(\phi_1 + \phi_2) & \omega & \phi_1 & \phi_2 \\ s_1^i(k) & 0 & 1 - s_1^i(k) & 0 \\ 0 & 0 & \frac{G_D^i(k)}{s_{DG}(k)} & (1 - \sum_{i=1}^N \frac{G_D^i(k)}{s_{DG}(k)}) \end{bmatrix} \quad (2.21)$$

Com isso é possível organizar o sistema conforme apresentado na seção anterior nas equações 2.13, e 2.14 que retornam a solução otimizada do sistema.

O algoritmo da PSOM, versão sem simplificações é descrito conforme mostrado:

Algoritmo 3 PSOM - Versão não Simplificada

- 1: Criar populações de agentes distribuídos uniformemente ao longo dos planos X , V , e P_B ;
 - 2: Montar a matriz “A”, conforme equação 2.21;
 - 3: Atualizar as partículas conforme equação 2.13;
 - 4: Se o critério de avaliação não for atingido volte ao passo 2. Caso contrário finalizar o algoritmo.
-

2.4.1 Análise de Estabilidade do Algoritmo

Alguns trabalhos na literatura buscaram analisar a convergência e estabilidade do algoritmo PSO de várias maneiras. Dentre estas é possível citar a primeira análise de estabilidade do PSO proposta por Clerc e Kennedy (2002), a análise de estabilidade utilizando o conceito de estabilidade absoluta proposta por Kadiramanathan et al. (2006), a análise de convergência do PSO utilizando funções multi-objetivo proposta por Chakraborty et al. (2010) e a análise de estabilidade não linear variante no tempo do PSO proposta por Fan et al. (2010). Pode-se concluir, então, que a análise de estabilidade vem sendo profundamente estudada por vários autores.

Assim como foi analisada a estabilidade de algoritmos baseados em enxame de partículas pelos autores será realizada uma análise de estabilidade dividindo o algoritmo PSO, proposto, em sistemas lineares.

Nas ultimas duas décadas aumentou-se o interesse no estudo de sistemas híbridos. Esses sistemas são utilizados para reduzir a complexidade na modelagem de sistemas não lineares. Muitos deles podem ser modelados como um conjunto de sistemas lineares com chaves selecionadoras entre eles. Essa abordagem caracteriza uma classe de sistemas híbridos conhecidos como sistemas lineares por partes. Mesmo havendo progressos importantes na análise de estabilidade de sistemas lineares por partes, sua análise ainda é um grande desafio. Pois, a estabilidade de sistemas lineares por partes não pode ser inferida apenas pela estabilidade dos sistemas que o compõe, porque mesmo no caso em que todas as partes são estáveis a estrutura de chaveamento pode fazer com que o sistema se instabilize (BARIJOUGH; LEE, 2010). Como a estabilidade de um sistema linear por partes depende não só de todas as partes do sistema serem estáveis, mas também da estrutura de chaveamento, será proposta uma análise que leve em consideração ambos os acontecimentos, ou seja, cada parte em separado e o sistema como um todo.

Suponhamos que a equação 2.13 possa ser reescrita na forma da equação 2.22.

$$T(k + 1) = AT(k) \quad (2.22)$$

Ou seja, em princípio será assumindo que os seus parâmetros são constantes (lembrando que o sistema mostrado é aplicável tanto à estrutura do PSOM simplificado quanto

para o exato).

A partir do conceito de autovalores é possível provar que a estabilidade assintótica do ponto fixo do sistema linear depende do módulo dos autovalores de “A”, matriz da equação 2.22. Isto é, o ponto de equilíbrio é assintoticamente estável quando o módulo de todos os seus autovalores é menor que 1, para um sistema discreto. Em consequência, quando, pelo menos, o módulo de um autovalor é maior que 1 o sistema é instável (MONTEIRO, 2002). Isto significa que:

- A equação 2.22 é marginalmente estável se, e somente se, todos os autovalores de A têm magnitudes menores ou iguais a 1 e aqueles que tiverem magnitudes iguais a 1 forem raízes simples do polinômio mínimo de A;
- A equação 2.22 é assintoticamente estável se, e somente se, todos os autovalores de A têm magnitudes menores que 1;
- A equação 2.22 é instável quando pelo menos o módulo de um autovalor for maior que 1;

Para aplicar os conceitos apresentados é necessário supor que o sistema é linear e invariante no tempo. Dessa maneira, serão assumidas tais hipóteses de maneira a analisar parte do algoritmo.

Devido a $s_1^i(k)$ e $s_2(k)$ (onde $\sum_{i=1}^N \frac{G_D^i(k)}{s_{DG}(k)}$ é semelhante a $s_2(k)$) serem funções que variam em apenas dois valores é possível reestruturar o PSO matricial em 4 diferentes partes principais. Os valores que $s_1^i(k)$ e $s_2(k)$ podem assumir são:

- Para $s_1^i(k) = 0$ e $s_2(k) = 0$ temos:

$$A_1 = \begin{bmatrix} 1 - (\phi_1 + \phi_2) & \omega & \phi_1 & \phi_2 \\ -(\phi_1 + \phi_2) & \omega & \phi_1 & \phi_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.23)$$

- Para $s_1^i(k) = 1$ e $s_2(k) = 0$ temos:

$$A_2 = \begin{bmatrix} 1 - (\phi_1 + \phi_2) & \omega & \phi_1 & \phi_2 \\ -(\phi_1 + \phi_2) & \omega & \phi_1 & \phi_2 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.24)$$

- Para $s_1^i(k) = 0$ e $s_2(k) = 1$ temos:

$$A_3 = \begin{bmatrix} 1 - (\phi_1 + \phi_2) & \omega & \phi_1 & \phi_2 \\ -(\phi_1 + \phi_2) & \omega & \phi_1 & \phi_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad (2.25)$$

- Para $s_1^i(k) = 1$ e $s_2(k) = 1$ temos:

$$A_4 = \begin{bmatrix} 1 - (\phi_1 + \phi_2) & \omega & \phi_1 & \phi_2 \\ -(\phi_1 + \phi_2) & \omega & \phi_1 & \phi_2 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad (2.26)$$

Como o comportamento de $s_1^i(k)$ e $s_2(k)$ é equivalente a uma chave, ou seja, assume valores “0” ou “1”, as possibilidades formam o conjunto que calcula as novas partículas, o valor das melhores locais e a melhor partícula global para o algoritmo de PSO matricial. Com isso, é possível reorganizar o sistema da seguinte maneira:

$$T(k+1) = (\delta_1 A_1 + \dots + \delta_n A_n) T(k) \quad (2.27)$$

Sendo que:

- $\delta_i \in \{0, 1\}$, $\sum_{i=1}^n \delta_i = 1$ e $P(\delta_i = 1) = \frac{1}{k}$. Onde não necessariamente as probabilidades devem ser uniformemente distribuídas na forma $\frac{1}{k}$. Porém, isso simplifica a notação e é a distribuição mais comumente utilizada ([HANLON et al., 2011](#)).

- A_l corresponde à matriz A gerada com todas as possibilidades de $s_1^i(k)$ e $s_2(k)$.
Ou seja:

$$l = 1 \rightarrow s_1^i(k) = 0 \text{ e } s_2(k) = 0$$

$$l = 2 \rightarrow s_1^i(k) = 1 \text{ e } s_2(k) = 0$$

$$l = 3 \rightarrow s_1^i(k) = 0 \text{ e } s_2(k) = 1$$

$$l = 4 \rightarrow s_1^i(k) = 1 \text{ e } s_2(k) = 1$$

Como os sistemas apresentados nas equações 2.23, 2.24, 2.25, 2.26 estão na forma linear, é possível obter os autovalores de cada matriz, mostrados nas equações 2.28, 2.29, 2.30, 2.31.

- Para A_1 temos:

$$\begin{pmatrix} 1 & 1 \\ \frac{\omega}{2} - \frac{\phi_2}{2} - \frac{\phi_1}{2} - \frac{\sqrt{\phi_1^2 + 2\phi_1\phi_2 - 2\phi_1\omega - 2\phi_1 + \phi_2^2 - 2\phi_2\omega - 2\phi_2 + \omega^2 - 2\omega + 1}}{2} + \frac{1}{2} & \frac{\omega}{2} - \frac{\phi_2}{2} - \frac{\phi_1}{2} + \frac{\sqrt{\phi_1^2 + 2\phi_1\phi_2 - 2\phi_1\omega - 2\phi_1 + \phi_2^2 - 2\phi_2\omega - 2\phi_2 + \omega^2 - 2\omega + 1}}{2} + \frac{1}{2} \end{pmatrix} \quad (2.28)$$

- Para A_2 temos:

$$\begin{pmatrix} 0 & 1 \\ \frac{\omega}{2} - \frac{\phi_2}{2} - \frac{\phi_1}{2} - \frac{\sqrt{\phi_1^2 + 2\phi_1\phi_2 - 2\phi_1\omega + 2\phi_1 + \phi_2^2 - 2\phi_2\omega - 2\phi_2 + \omega^2 - 2\omega + 1}}{2} + \frac{1}{2} & \frac{\omega}{2} - \frac{\phi_2}{2} - \frac{\phi_1}{2} + \frac{\sqrt{\phi_1^2 + 2\phi_1\phi_2 - 2\phi_1\omega + 2\phi_1 + \phi_2^2 - 2\phi_2\omega - 2\phi_2 + \omega^2 - 2\omega + 1}}{2} + \frac{1}{2} \end{pmatrix} \quad (2.29)$$

- Para A_3 temos:

$$\begin{pmatrix} 0 & 1 \\ \frac{\omega}{2} - \frac{\phi_2}{2} - \frac{\phi_1}{2} - \frac{\sqrt{\phi_1^2 + 2\phi_1\phi_2 - 2\phi_1\omega - 2\phi_1 + \phi_2^2 - 2\phi_2\omega - 2\phi_2 + \omega^2 - 2\omega + 1}}{2} + \frac{1}{2} & \frac{\omega}{2} - \frac{\phi_2}{2} - \frac{\phi_1}{2} + \frac{\sqrt{\phi_1^2 + 2\phi_1\phi_2 - 2\phi_1\omega - 2\phi_1 + \phi_2^2 - 2\phi_2\omega - 2\phi_2 + \omega^2 - 2\omega + 1}}{2} + \frac{1}{2} \end{pmatrix} \quad (2.30)$$

- Para A_4 temos:

$$\begin{pmatrix} 0 \\ 1 \\ \frac{\omega}{2} - \frac{\phi_2}{2} - \frac{\phi_1}{2} - \frac{\sqrt{\phi_1^2 + 2\phi_1\phi_2 - 2\phi_1\omega + \phi_2^2 - 2\phi_2\omega - 4\phi_2 + \omega^2}}{2} \\ \frac{\omega}{2} - \frac{\phi_2}{2} - \frac{\phi_1}{2} + \frac{\sqrt{\phi_1^2 + 2\phi_1\phi_2 - 2\phi_1\omega + \phi_2^2 - 2\phi_2\omega - 4\phi_2 + \omega^2}}{2} \end{pmatrix} \quad (2.31)$$

É possível perceber que em todas as equações (2.28, 2.29, 2.30, 2.31), assumindo os parâmetros ω , ϕ_1 , e ϕ_2 constantes, existe pelo menos um autovalor em “1”. Isso significa que separadamente cada matriz é, no mínimo, marginalmente estável, podendo ser instável dependendo da escolha dos parâmetros ω , ϕ_1 , e ϕ_2 . Neste caso, não há convergência garantida analisando as partes lineares.

Para que seja possível reposicionar os autovalores das matrizes tanto para o caso simplificado quanto exato é necessário realizar uma modificação nas matrizes A. Para o caso específico do PSOM, será proposta uma modificação de maneira que seja possível posicionar os autovalores no interior do círculo de raio unitário (para que essa parte do sistema deixe de ser marginalmente estável e se torne assintoticamente estável), quando ocorrer $s_1^i = 1$ e $s_2 = 1$, momento onde P_B recebe o valor de X e P_G recebe o valor de P_B . A modificação proposta para satisfazer tal objetivo é inserir um fator de esquecimento (λ_{esq}) na equação de atualização da partícula. Dessa maneira, é possível reescrever a matriz “A” como apresentado nas equações 2.32 ou 2.33.

$$A_{esq1}(k) = \begin{bmatrix} \lambda_{esq} - (\phi_1 + \phi_2) & \omega & \phi_1 & \phi_2 \\ -(\phi_1 + \phi_2) & \omega & \phi_1 & \phi_2 \\ s_1^i & 0 & 1 - s_1^i & 0 \\ 0 & 0 & \frac{s_2}{n_I} & 1 - s_2 \end{bmatrix} \quad (2.32)$$

$$A_{esq2}(k) = \begin{bmatrix} \lambda_{esq} - (\phi_1 + \phi_2) & \omega & \phi_1 & \phi_2 \\ -(\phi_1 + \phi_2) & \omega & \phi_1 & \phi_2 \\ s_1^i(k) & 0 & 1 - s_1^i(k) & 0 \\ 0 & 0 & \frac{G_D^i(k)}{s_{DG}(k)} & (1 - \sum_{i=1}^N \frac{G_D^i(k)}{s_{DG}(k)}) \end{bmatrix} \quad (2.33)$$

Ao substituir as quatro possibilidades de $s_1^i(k)$ e $s_2(k)$, onde $\sum_{i=1}^N \frac{G_D^i(k)}{s_{DG}(k)}$ é semelhante a $s_2(k)$, é possível encontrar os autovalores das equações 2.34, 2.35, 2.36, 2.37.

Para $s_1^i(k) = 0$ e $s_2(k) = 0$ temos:

$$\begin{pmatrix} 1 \\ 1 \\ \frac{\lambda_{esq}}{2} - \frac{\phi_1}{2} - \frac{\phi_2}{2} + \frac{\omega}{2} - \frac{\sqrt{\lambda_{esq}^2 - 2\lambda_{esq}\phi_1 - 2\lambda_{esq}\phi_2 - 2\lambda_{esq}\omega + \phi_1^2 + 2\phi_1\phi_2 - 2\phi_1\omega + \phi_2^2 - 2\phi_2\omega + \omega^2}}{2} \\ \frac{\lambda_{esq}}{2} - \frac{\phi_1}{2} - \frac{\phi_2}{2} + \frac{\omega}{2} + \frac{\sqrt{\lambda_{esq}^2 - 2\lambda_{esq}\phi_1 - 2\lambda_{esq}\phi_2 - 2\lambda_{esq}\omega + \phi_1^2 + 2\phi_1\phi_2 - 2\phi_1\omega + \phi_2^2 - 2\phi_2\omega + \omega^2}}{2} \end{pmatrix} \quad (2.34)$$

Para $s_1^i(k) = 1$ e $s_2(k) = 0$ temos:

$$\begin{pmatrix} 0 \\ 1 \\ \frac{\lambda_{esq}}{2} - \frac{\phi_1}{2} - \frac{\phi_2}{2} + \frac{\omega}{2} - \frac{\sqrt{\lambda_{esq}^2 - 2\lambda_{esq}\phi_1 - 2\lambda_{esq}\phi_2 - 2\lambda_{esq}\omega + \phi_1^2 + 2\phi_1\phi_2 - 2\phi_1\omega + 4\phi_1 + \phi_2^2 - 2\phi_2\omega + \omega^2}}{2} \\ \frac{\lambda_{esq}}{2} - \frac{\phi_1}{2} - \frac{\phi_2}{2} + \frac{\omega}{2} + \frac{\sqrt{\lambda_{esq}^2 - 2\lambda_{esq}\phi_1 - 2\lambda_{esq}\phi_2 - 2\lambda_{esq}\omega + \phi_1^2 + 2\phi_1\phi_2 - 2\phi_1\omega + 4\phi_1 + \phi_2^2 - 2\phi_2\omega + \omega^2}}{2} \end{pmatrix} \quad (2.35)$$

Para $s_1^i(k) = 0$ e $s_2(k) = 1$ temos:

$$\begin{pmatrix} 0 \\ 1 \\ \frac{\lambda_{esq}}{2} - \frac{\phi_1}{2} - \frac{\phi_2}{2} + \frac{\omega}{2} - \frac{\sqrt{\lambda_{esq}^2 - 2\lambda_{esq}\phi_1 - 2\lambda_{esq}\phi_2 - 2\lambda_{esq}\omega + \phi_1^2 + 2\phi_1\phi_2 - 2\phi_1\omega + \phi_2^2 - 2\phi_2\omega + \omega^2}}{2} \\ \frac{\lambda_{esq}}{2} - \frac{\phi_1}{2} - \frac{\phi_2}{2} + \frac{\omega}{2} + \frac{\sqrt{\lambda_{esq}^2 - 2\lambda_{esq}\phi_1 - 2\lambda_{esq}\phi_2 - 2\lambda_{esq}\omega + \phi_1^2 + 2\phi_1\phi_2 - 2\phi_1\omega + \phi_2^2 - 2\phi_2\omega + \omega^2}}{2} \end{pmatrix} \quad (2.36)$$

Para $s_1^i(k) = 1$ e $s_2(k) = 1$ temos:

$$\begin{pmatrix} 0 \\ \frac{\lambda_{esq}}{3} + \frac{\omega}{3} - \frac{\phi_1}{3} - \frac{\phi_2}{3} + \sigma_1 + \frac{\sigma_2}{\sigma_1} \\ \frac{\lambda_{esq}}{3} + \frac{\omega}{3} - \frac{\phi_1}{3} - \frac{\phi_2}{3} - \frac{\sigma_1}{2} - \frac{\sigma_2}{2\sigma_1} - \frac{\sqrt{3}(\sigma_1 - \frac{\sigma_2}{\sigma_1})i}{2} \\ \frac{\lambda_{esq}}{3} + \frac{\omega}{3} - \frac{\phi_1}{3} - \frac{\phi_2}{3} - \frac{\sigma_1}{2} - \frac{\sigma_2}{2\sigma_1} + \frac{\sqrt{3}(\sigma_1 - \frac{\sigma_2}{\sigma_1})i}{2} \end{pmatrix} \quad (2.37)$$

Sendo:

$$\sigma_1 = \left(\begin{aligned} & \frac{\phi_2}{2} + \frac{(\lambda_{esq} + \omega - \phi_1 - \phi_2)^3}{27} + \dots \\ & + \frac{(\phi_1 - \lambda_{esq} + \omega)(\lambda_{esq} + \omega - \phi_1 - \phi_2)}{6} + \dots \end{aligned} \right)^{\frac{1}{3}} + \sqrt{\left(\frac{\phi_2}{2} + \frac{(\lambda_{esq} + \omega - \phi_1 - \phi_2)^3}{27} + \frac{(\phi_1 - \lambda_{esq} + \omega)(\lambda_{esq} + \omega - \phi_1 - \phi_2)}{6} \right)^2 - \sigma_2^3} \quad (2.38)$$

$$\sigma_2 = \frac{\phi_1}{3} + \frac{(\lambda_{esq} + \omega - \phi_1 - \phi_2)^2}{9} - \frac{\lambda_{esq}\omega}{3} \quad (2.39)$$

Como pode-se observar na equação 2.37, dependendo da forma como são escolhidos

os parâmetros, é possível encontrar valores constantes para as variáveis ϕ_1 , ϕ_2 , ω e λ_{esq} onde o módulo dos autovalores seja menor que “1”, e, para as outras possibilidades, sejam no máximo igual a “1”.

Contudo, a estabilidade dos sistemas lineares por partes não pode ser inferida apenas pela estabilidade dos sistemas que os compõe, porque mesmo no caso em que todas as partes são estáveis a estrutura de chaveamento pode fazer com que o sistema se instabilize (BARIJOUGH; LEE, 2010). Por isso, como a estabilidade de um sistema linear por partes depende não só de todas as partes do sistema serem estáveis, mas também da estrutura de chaveamento. A seguir, será proposta uma forma de analisar a estabilidade da estrutura da equação 2.27.

A teoria de sistemas chaveados tem recebido uma atenção especial na literatura de teoria de sistemas lineares nas últimas duas décadas, apresentando resultados sólidos no que concerne a estabilidade e estabilização desses sistemas. Um desses resultados, que pode ser visualizado em (HANLON et al., 2011), apresenta o seguinte teorema:

Teorema 1: O sistema 2.27 converge para zero quase certamente se e somente se todos os autovalores da matriz $\frac{\sum_{i=1}^n A_i}{n}$ se encontram no interior do ciclo de raio unitário.

Baseado no teorema apresentado a matriz da equação 2.32, substituídas com as 4 possibilidades, resultariam numa nova matriz \bar{A} , que pode ser visualizada na equação 2.40 cujos autovalores determinam a estabilidade quase certa das partículas.

$$\bar{A} = \sum_{l=1}^n \frac{A_l}{n} \quad (2.40)$$

Assim, é possível reescrever a matriz 2.40 na forma da equação 2.41.

$$\bar{A} = \begin{bmatrix} \lambda_{esq} - (c_1 + c_2) & \omega & c_1 & c_2 \\ -(c_1 + c_2) & \omega & c_1 & c_2 \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 \\ 0 & 0 & \frac{1}{2} & \frac{1}{2} \end{bmatrix} \quad (2.41)$$

Em que:

- $c_1 = \frac{\phi_{1in} + \phi_{1f}}{2}$;
- $\phi_{1in} = 0$;
- $c_2 = \frac{\phi_{2in} + \phi_{2f}}{2}$;
- $\phi_{2in} = 0$.

- ϕ_{in} e ϕ_{if} são os valores inicial e final dos intervalos onde as variáveis ϕ_1 e ϕ_2 estão distribuídas.

Com base na matriz da equação 2.41 é possível realizar uma simulação em busca de regiões estáveis, que são as possíveis escolhas de parâmetros que resultam em uma estabilidade quase certa. A figura 1 apresenta uma região formada com os parâmetros no R^4 (ϕ_1 , ϕ_2 , ω e λ_{esq}), porém, apresentada em pares, onde os parâmetros que não são apresentados em cada gráfico foram fixados iguais a 1.

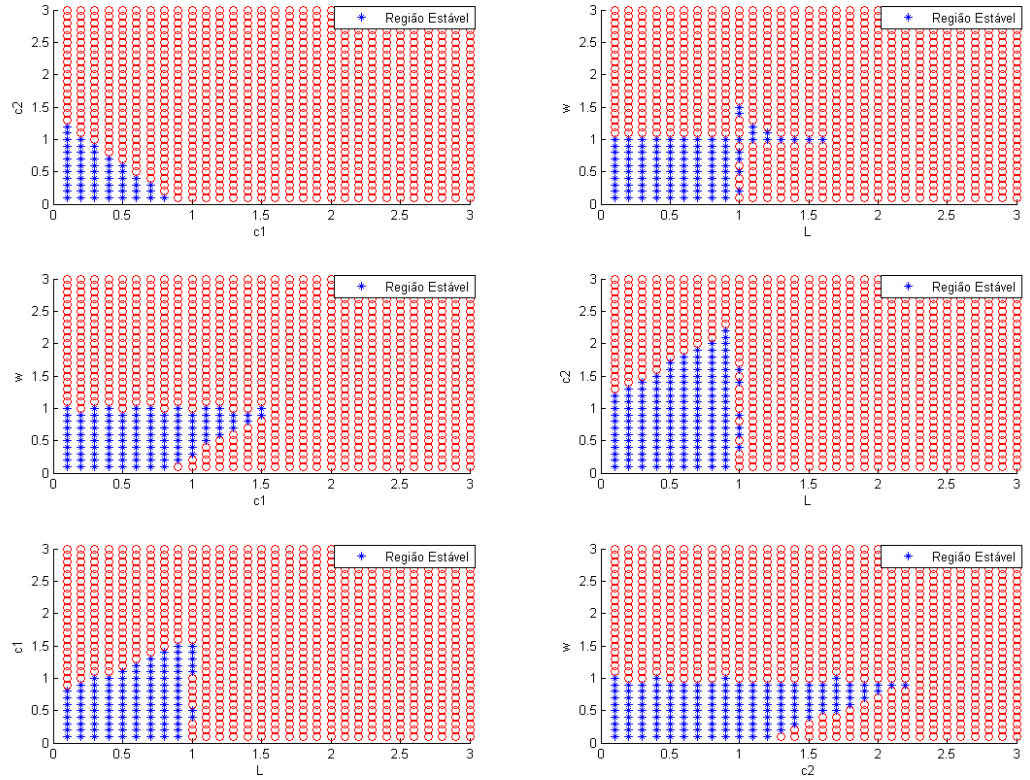


Figura 1 - Gráficos com regiões de estabilidade no R^4 com os parâmetros não apresentados iguais a 1

Portanto, é possível determinar uma região de estabilidade quase certa para o sistema linear por partes, apresentado na equação 2.27. Isso significa que a comutação entre as matrizes que geram esse sistema quase certamente resultará na convergência do estado $T(k)$, e, em consequência, na convergência das partículas (X), das velocidades (V), das melhores locais (P_B) e das melhores globais (P_G).

2.4.2 Análise de Convergência do Erro Quadrático Iterativo

O critério de estabilidade utilizado na seção 2.4.1 apenas garante que as partículas dentro do PSO tenderão a convergir para um valor finito, não significando que essa convergência acontecerá na direção da minimização ou maximização da função custo. Por isso, com o objetivo de analisar a convergência no sentido de otimizar a função custo, deve-se analisar se o erro diminui a cada iteração. Neste caso podemos utilizar o critério da inequação 2.42.

$$E(k)^2 - E(k-1)^2 < 0 \quad (2.42)$$

Em que:

- $E(k) = y - S A(k) T(k)$, isto é, $E(k) = y - S A(k) A(k-1) T(k-1)$;
- $E(k-1) = y - S A(k-1) T(k-1)$

Através da inequação 2.42 é possível perceber que sempre que a mesma é verdadeira o erro diminui, e, portanto, o sistema fica mais próximo da solução correta.

Para que seja possível perceber quais são as condições necessárias para que o erro diminua a inequação pode ser expandida. Conforme as inequações 2.43, 2.44, 2.45, 2.46, 2.47, 2.48, 2.49:

$$y^2 - 2y S A(k) A(k-1) T(k-1) + (S A(k) A(k-1) T(k-1))^2 - (y^2 - 2y S A(k-1) T(k-1) + (S A(k-1) T(k-1))^2) < 0 \quad (2.43)$$

$$-2y S A(k) A(k-1) T(k-1) + (S A(k) A(k-1) T(k-1))^2 + 2y S A(k-1) T(k-1) - (S A(k-1) T(k-1))^2 < 0 \quad (2.44)$$

$$-2y S A(k) A(k-1) T(k-1) + (S A(k) A(k-1) T(k-1) S A(k) A(k-1) T(k-1)) + 2y S A(k-1) T(k-1) - (S A(k-1) T(k-1) S A(k-1) T(k-1)) < 0 \quad (2.45)$$

$$(2y S I - 2y S A(k)) A(k-1) T(k-1) + (S A(k) A(k-1) T(k-1) S A(k) A(k-1) T(k-1)) - (S A(k-1) T(k-1) S A(k-1) T(k-1)) < 0 \quad (2.46)$$

$$2yS(I - A(k))A(k-1)T(k-1) + (SA(k)A(k-1)T(k-1)SA(k)A(k-1)T(k-1)) - (SA(k-1)T(k-1)SA(k-1)T(k-1)) < 0 \quad (2.47)$$

$$2yS(I - A(k))A(k-1)T(k-1) + ((SA(k)A(k-1)T(k-1)SA(k)) - (SA(k-1)T(k-1)S))A(k-1)T(k-1) < 0 \quad (2.48)$$

$$2yS(I - A(k))A(k-1)T(k-1) + S((A(k)A(k-1)T(k-1)SA(k)) - (A(k-1)T(k-1)S))A(k-1)T(k-1) < 0 \quad (2.49)$$

Ao analisar a inequação 2.49 conjuntamente com a forma como as matrizes $A(k)$ e $A(k-1)$ são montadas (tendo como base a equação 2.12), é possível perceber que existem $4 * 4$ possibilidades, 4 para $A(k)$ e 4 para $A(k-1)$. Dessa maneira temos:

- Para $s_1(k-1) = 0, s_2(k-1) = 0, s_1(k) = 0, s_2(k) = 0$;

$$2yS(I - A(k))A(k-1)T(k-1) + S((A(k)A(k-1)T(k-1)SA(k)) - (A(k-1)T(k-1)S))A(k-1)T(k-1) = 0 \quad (2.50)$$

- Para $s_1(k-1) = 0, s_2(k-1) = 0, s_1(k) = 0, s_2(k) = 1$;

$$\begin{aligned} & 2yS(I - A(k))A(k-1)T(k-1) + \\ & S((A(k)A(k-1)T(k-1)SA(k)) - \\ & (A(k-1)T(k-1)S))A(k-1)T(k-1) = \\ & -(G(k-1) - P(k-1))(G(k-1) + P(k-1) - 2y) \end{aligned} \quad (2.51)$$

- Para $s_1(k-1) = 0, s_2(k-1) = 0, s_1(k) = 1, s_2(k) = 0$;

$$2yS(I - A(k))A(k-1)T(k-1) + S((A(k)A(k-1)T(k-1)SA(k)) - (A(k-1)T(k-1)S))A(k-1)T(k-1) = 0 \quad (2.52)$$

- Para $s_1(k-1) = 0, s_2(k-1) = 0, s_1(k) = 1, s_2(k) = 1$;

$$\begin{aligned}
& 2yS(I - A(k))A(k-1)T(k-1) + \\
& S((A(k)A(k-1)T(k-1)SA(k)) - \\
& (A(k-1)T(k-1)S))A(k-1)T(k-1) = \\
& -(G(k-1) - P(k-1))(G(k-1) + P(k-1) - 2y)
\end{aligned} \tag{2.53}$$

- Para $s_1(k-1) = 0, s_2(k-1) = 1, s_1(k) = 0, s_2(k) = 0$;

$$\begin{aligned}
& 2yS(I - A(k))A(k-1)T(k-1) + S((A(k)A(k-1)T(k-1)SA(k)) - \\
& (A(k-1)T(k-1)S))A(k-1)T(k-1) = 0
\end{aligned} \tag{2.54}$$

- Para $s_1(k-1) = 0, s_2(k-1) = 1, s_1(k) = 0, s_2(k) = 1$;

$$\begin{aligned}
& 2yS(I - A(k))A(k-1)T(k-1) + S((A(k)A(k-1)T(k-1)SA(k)) - \\
& (A(k-1)T(k-1)S))A(k-1)T(k-1) = 0
\end{aligned} \tag{2.55}$$

- Para $s_1(k-1) = 0, s_2(k-1) = 1, s_1(k) = 1, s_2(k) = 0$;

$$\begin{aligned}
& 2yS(I - A(k))A(k-1)T(k-1) + S((A(k)A(k-1)T(k-1)SA(k)) - \\
& (A(k-1)T(k-1)S))A(k-1)T(k-1) = 0
\end{aligned} \tag{2.56}$$

- Para $s_1(k-1) = 0, s_2(k-1) = 1, s_1(k) = 1, s_2(k) = 1$;

$$\begin{aligned}
& 2yS(I - A(k))A(k-1)T(k-1) + S((A(k)A(k-1)T(k-1)SA(k)) - \\
& (A(k-1)T(k-1)S))A(k-1)T(k-1) = 0
\end{aligned} \tag{2.57}$$

- Para $s_1(k-1) = 1, s_2(k-1) = 0, s_1(k) = 0, s_2(k) = 0$;

$$\begin{aligned}
& 2yS(I - A(k))A(k-1)T(k-1) + S((A(k)A(k-1)T(k-1)SA(k)) - \\
& (A(k-1)T(k-1)S))A(k-1)T(k-1) = 0
\end{aligned} \tag{2.58}$$

- Para $s_1(k-1) = 1, s_2(k-1) = 0, s_1(k) = 0, s_2(k) = 1$;

$$\begin{aligned}
& 2yS(I - A(k))A(k-1)T(k-1) + \\
& S((A(k)A(k-1)T(k-1)SA(k)) - \\
& (A(k-1)T(k-1)S))A(k-1)T(k-1) = \\
& -(G(k-1) - X(k-1))(G(k-1) + X(k-1) - 2y)
\end{aligned} \tag{2.59}$$

- Para $s_1(k-1) = 1, s_2(k-1) = 0, s_1(k) = 1, s_2(k) = 0$;

$$\begin{aligned}
& 2yS(I - A(k))A(k-1)T(k-1) + S((A(k)A(k-1)T(k-1)SA(k)) - \\
& (A(k-1)T(k-1)S))A(k-1)T(k-1) = 0
\end{aligned} \tag{2.60}$$

- Para $s_1(k-1) = 1, s_2(k-1) = 0, s_1(k) = 1, s_2(k) = 1$;

$$\begin{aligned}
& 2yS(I - A(k))A(k-1)T(k-1) + \\
& S((A(k)A(k-1)T(k-1)SA(k)) - \\
& (A(k-1)T(k-1)S))A(k-1)T(k-1) = \\
& -(G(k-1) - X(k-1))(G(k-1) + X(k-1) - 2y)
\end{aligned} \tag{2.61}$$

- Para $s_1(k-1) = 1, s_2(k-1) = 1, s_1(k) = 0, s_2(k) = 0$;

$$\begin{aligned}
& 2yS(I - A(k))A(k-1)T(k-1) + S((A(k)A(k-1)T(k-1)SA(k)) - \\
& (A(k-1)T(k-1)S))A(k-1)T(k-1) = 0
\end{aligned} \tag{2.62}$$

- Para $s_1(k-1) = 1, s_2(k-1) = 1, s_1(k) = 0, s_2(k) = 1$;

$$\begin{aligned}
& 2yS(I - A(k))A(k-1)T(k-1) + \\
& S((A(k)A(k-1)T(k-1)SA(k)) - \\
& (A(k-1)T(k-1)S))A(k-1)T(k-1) = \\
& -(P(k-1) - X(k-1))(P(k-1) + X(k-1) - 2y)
\end{aligned} \tag{2.63}$$

- Para $s_1(k-1) = 1, s_2(k-1) = 1, s_1(k) = 1, s_2(k) = 0$;

$$\begin{aligned}
& 2yS(I - A(k))A(k-1)T(k-1) + S((A(k)A(k-1)T(k-1)SA(k)) - \\
& (A(k-1)T(k-1)S))A(k-1)T(k-1) = 0
\end{aligned} \tag{2.64}$$

- Para $s_1(k-1) = 1, s_2(k-1) = 1, s_1(k) = 1, s_2(k) = 1$;

$$\begin{aligned}
& 2yS(I - A(k))A(k-1)T(k-1) + \\
& S((A(k)A(k-1)T(k-1)SA(k)) - \\
& (A(k-1)T(k-1)S))A(k-1)T(k-1) = \\
& -(P(k-1) - X(k-1))(P(k-1) + X(k-1) - 2y)
\end{aligned} \tag{2.65}$$

Como pode ser percebido, existem apenas 3 possibilidades de ocorrer a diminuição do erro em direção ao zero, descritos pelas condições expressas nas inequações 2.66, 2.67 e 2.68.

$$-(G(k-1) - P(k-1))(G(k-1) + P(k-1) - 2y) < 0 \tag{2.66}$$

$$-(G(k-1) - X(k-1))(G(k-1) + X(k-1) - 2y) < 0 \tag{2.67}$$

$$-(P(k-1) - X(k-1))(P(k-1) + X(k-1) - 2y) < 0 \tag{2.68}$$

Isso significa que para que as inequações 2.66, 2.67 e 2.68 sejam satisfeitas é necessário que as condições 2.69, 2.70 e 2.71 aconteçam.

$$\left. \begin{array}{c} (G(k-1) - P(k-1)) > 0 \\ E \\ (G(k-1) + P(k-1) - 2y) > 0 \end{array} \right\} ou \left\{ \begin{array}{c} -(G(k-1) - P(k-1)) > 0 \\ E \\ -(G(k-1) + P(k-1) - 2y) > 0 \end{array} \right. \tag{2.69}$$

$$\left. \begin{array}{c} (G(k-1) - X(k-1)) > 0 \\ E \\ (G(k-1) + X(k-1) - 2y) > 0 \end{array} \right\} ou \left\{ \begin{array}{c} -(G(k-1) - X(k-1)) > 0 \\ E \\ -(G(k-1) + X(k-1) - 2y) > 0 \end{array} \right. \tag{2.70}$$

$$\left. \begin{array}{c} (P(k-1) - X(k-1)) > 0 \\ E \\ (P(k-1) + X(k-1) - 2y) > 0 \end{array} \right\} ou \left\{ \begin{array}{c} -(P(k-1) - X(k-1)) > 0 \\ E \\ -(P(k-1) + X(k-1) - 2y) > 0 \end{array} \right. \tag{2.71}$$

Algo que é possível perceber com isso é que a partir da forma como as partículas se movimentam é possível estimar quando as partículas estão abaixo do ponto de equilíbrio ou acima dele.

2.4.3 PSOM - Híbrido

Após a PSO ser reescrito na forma matricial e realizada uma análise de convergências das partículas e da otimização, foi pensado em uma maneira de combinar a PSO com algoritmos de otimização determinísticos, quando a PSO é aplicada a processos com incertezas. O objetivo desta combinação é eliminar a possibilidade de uma inicialização ruim, que não permita a convergência das partículas para uma solução aceitável na PSO. Além disso, intenciona-se obter desempenhos melhores que os algoritmos tradicionais. Ou seja, associando algoritmos determinísticos com o PSO espera-se obter resultados tão bons ou melhores que os algoritmos determinísticos, eliminando a possibilidade de não encontrar um bom indivíduo devido à aleatoriedade do PSO.

Neste trabalho a hibridização consistirá em utilizar o resultado obtido com o uso de algoritmos determinísticos como base para a inicialização das melhores partículas locais (P_B) e melhor global do PSO (P_G) conforme apresentado nas equações 2.72 e 2.73.

$$P_B^i(0) = P_D + C \text{Rand}_{(1,n_{var})} P_D \quad (2.72)$$

$$P_G^0(0) = P_D \quad (2.73)$$

Onde P_D é o resultado obtido pelo algoritmo determinístico, C é uma constante que determina a região onde serão dispersadas as melhores partículas locais e Rand é um vetor de números aleatórios com distribuição uniforme no intervalo $[-1,1]$.

Nesses termos será possível aplicar o algoritmo de PSO para otimização de processos em que as técnicas tradicionais não alcançam o melhor desempenho, porém possuem desempenho suficiente para garantir resultados aceitáveis. Com isso será possível otimizar tal desempenho objetivando o melhor desempenho global dentro dos sistemas que serão utilizados.

3 IDENTIFICAÇÃO DE SISTEMAS

O algoritmo de otimização por nuvem de partículas pode ser aplicado nas mais variadas áreas, sempre o objetivo de encontrar a melhor solução de determinado processo. Uma das partes fundamentais deste trabalho é a aplicação do novo algoritmo de otimização, com a análise de convergência. A primeira área do conhecimento na qual serão realizadas análises e comparações é na área de identificação. O objetivo é realizar a modelagem de sistema através da minimização do erro de estimação, utilizando o algoritmo.

3.1 Introdução

Modelos de sistemas reais são de fundamental importância. Eles podem ser úteis para a análise como, por exemplo, para obter conhecimento sobre determinado sistema. A modelagem dos processos torna possível prever ou simular o seu comportamento. Além disso, técnicas avançadas para projeto de controladores, otimização, supervisão, detecção de falhas, e diagnóstico de componentes são também baseadas em modelos de processos (NELLES, 2001).

Modelos matemáticos podem ser desenvolvidos de duas maneiras, que podem ser combinadas. Uma maneira é desmembrando o sistema em outros subsistemas cujas propriedades são conhecidas. Isso basicamente significa confiar em conhecimentos como as leis da natureza e outras relações baseadas em trabalhos experimentais anteriores. Esses subsistemas, ao final, são agrupados matematicamente resultando em um modelo geral. Essa maneira de relacionar matematicamente um sistema é conhecida como modelagem e não necessariamente se utiliza de informações de dados coletados experimentalmente no sistema em questão. Nesse caso, o procedimento de modelar é praticamente dependente do sistema modelado e normalmente se limita a técnicas específicas na área de aplicação em questão. A outra maneira de obter um modelo, que descreve determinado sistema, é baseado em ensaios experimentais. Nesse caso, sinais de entrada e saída do sistema são armazenados e sujeitos a uma análise de dados com o objetivo de inferir o modelo. Isso consiste em identificar um sistema (LJUNG, 1987).

Identificação de sistemas ou identificação de processos é o campo da modelagem matemática de sistemas (processos) cujos parâmetros são obtidos a partir de testes ou dados experimentais. Em termos técnicos pode ser definido como a determinação de um modelo dentro de uma classe específica de sistemas baseada em dados de entrada e saída no qual o sistema excitado é equivalente (ZHU, 2001).

Uma das importantes características das técnicas de identificação de sistemas, se-

gundo [Aguirre \(2007\)](#), é que pouca ou nenhuma informação prévia é necessária. Devido a isso, em muitos casos será preferível usar técnicas de identificação de sistemas para modelar.

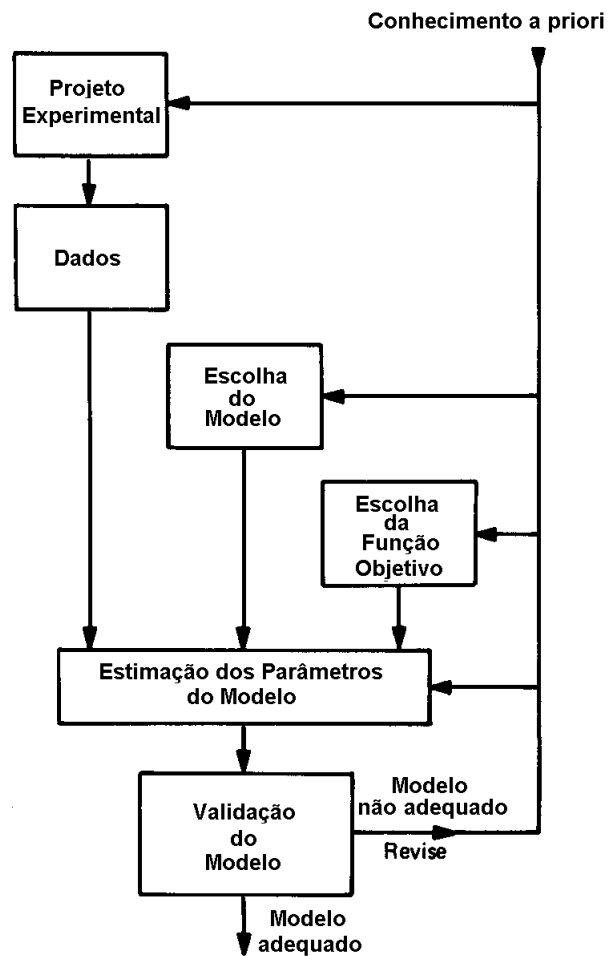
Segundo [Zhu \(2001\)](#) existem 4 entidades básicas envolvidas em identificação de sistemas que são:

- Entrada e Saída;
- Escolha do modelo;
- Estimação do modelo;
- Validação do modelo.

Os dados de entrada e saída são normalmente coletados a partir de um teste ou experimento de identificação, que é projetado de maneira que os dados medidos sejam suficientemente informativos, isto é, de maneira que seja extraído o máximo de informação possível sobre as propriedades do sistema de interesse ([ZHU, 2001](#)). Uma consideração que deve ser realizada, quando se trata da escolha do modelo, é que não existe uma única maneira de se modelar um sistema, e sim, uma família de modelos com características e desempenhos variados, que proporcionam representações aproximadas do processo a ser aprendido. Isto significa que decidir qual desses modelos melhor se adequa é um dos problemas não triviais a ser enfrentado ([AGUIRRE, 2007](#)).

Um conjunto de modelos candidatos pode ser obtido especificando propriedades comuns do sistema a ser estudado. É nesse estágio que o conhecimento *a priori* e a intuição devem ser combinados com as propriedades formais do sistema. Após obtidos os dados experimentais e escolhido o modelo que melhor se adequa ao sistema, os parâmetros desse modelo escolhido devem ser estimados. Para que seja possível essa estimação, um critério, comumente chamado de função objetivo, é especificado, e o valor dos parâmetros é encontrado, objetivando a minimização dessa função. Finalizada a etapa de estimação, o próximo passo é sua validação. Nesse estágio é avaliado se o modelo é suficientemente bom e se ele interpola os dados adequadamente ([ZHU, 2001](#)). A figura 2 mostra um diagrama representando a sequência de passos necessários para realizar a identificação de um processo.

Figura 2 - Procedimento básico para identificação. Fonte: Adaptado de Ljung (1987)



A figura 2 mostra o fluxo lógico e natural de um procedimento de identificação padrão. Primeiramente deve-se realizar, baseado no conhecimento do sistema (*a priori*), um projeto de sinais, cujo objetivo é obter a maior quantidade de informações possível do sistema. Esse passo faz parte do projeto experimental. Após, obtidos os dados do projeto experimental, esses dados, juntamente com o modelo escolhido, a partir do conhecimento *a priori*, servirão de base para a estimação de parâmetros, cujo objetivo é minimizar uma função custo (ou objetivo) que é uma relação do erro entre o modelo e os dados do processo. Finalmente, após estimados os parâmetros, o modelo deve ser avaliado. Caso ele não seja adequado, esse conhecimento poderá ser utilizado para a realização de outros projetos experimentais, ou levar à escolha de um outro modelo mais adequado, ou, ainda, levar à escolha de uma função objetivo mais apropriada (LJUNG, 1987).

3.2 Projeto de Sinais de Excitação

Ao fornecer a um sistema, que se deseja analisar sinais de entrada, é possível realizar um experimento de identificação. Esses sinais de teste devem ser aplicados quando os sinais operacionais (quando a planta está em operação) não são o suficiente para excitar as dinâmicas do sistema. Os sinais mais favoráveis para a excitação de sistemas segundo [Isermann e Munchhof \(2011\)](#) devem seguir alguns critérios tais como: devem ser simples de se reproduzir, com ou sem um gerador de sinais, possuir uma simples descrição matemática e suas propriedades correspondentes ao método de identificação, devem ser aplicáveis ao atuador do processo e ao processo como um todo, além de excitar as dinâmicas de interesse.

No projeto de sinais de excitação são elaboradas as entradas para o sistema em análise que, segundo [Aguirre \(2007\)](#), devem observar três aspectos:

- Onde excitar a planta;
- Que tipo de sinal utilizar de modo que estes sejam representativos o suficiente;
- Como amostrar os dados.

Com respeito a esses tópicos existem algumas diferenças entre modelos lineares e não lineares. Porém, a ideia principal listada é aplicável em ambos os casos ([AGUIRRE, 2007](#)). A ideia de projetos de sinais de excitação, neste trabalho, seguirá o seguinte fluxo: Primeiramente, haverá uma breve descrição de como escolher entre eles o sinal que melhor se adequa ao processo desejado e como realizar a coleta desses sinais. Em seguida, será abordada a escolha do período de amostragem do sistema. Nesse sentido, será apresentada uma metodologia para coleta de dados sobre-amostrados e uma solução para uma adequada re-amostragem.

3.2.1 Escolha e Coleta de Sinais

Os principais sinais utilizados em identificação são os sinais baseados em ruído. Sinais de teste baseados em ruído branco ou colorido podem ser utilizados para identificação e permitem obter a resposta ao impulso de um processo (sua função de transferência). Para o caso de um ruído branco, a resposta ao impulso é diretamente proporcional a sua função de correlação cruzada. Como a correlação cruzada de um sinal estacionário automaticamente separa o sinal desejado do ruído, é possível utilizar esses métodos mesmo na presença de grandes perturbações e relações sinal-ruído desfavoráveis. Nesse caso é apenas necessário que o tempo de medição dos dados seja longo o suficiente. O uso de ruídos naturais como sinal de teste é possível apenas sob certas circunstâncias. Por isso, na prática, é normalmente melhor utilizar sinais de excitações artificiais ([ISERMANN; MUNCHHOF, 2011](#)).

Os sinais pseudo aleatórios binários (PRBS) têm sido largamente utilizados devido a serem facilmente gerados computacionalmente, além de permitirem a identificação de respostas ao impulso diretamente. Porém, esse sinal não é recomendado para excitação de sistemas não lineares, pois, para que as não linearidades sejam identificadas, é necessário excitar mais níveis de sinal (ISERMANN; MUNCHHOF, 2011). No caso de sistemas não lineares, muitas vezes será necessário ou até desejável excitar várias amplitudes na faixa onde essa não linearidade é mais percebida com o objetivo de observar melhor as características estáticas e dinâmicas, ou seja, será interessante combinar as propriedades do PRBS em excitar várias frequências e ao mesmo tempo excitar várias amplitudes de sinal (AGUIRRE, 2007).

A seleção de uma sequência de sinais de entrada está associada diretamente à faixa de operação a ser excitada. Além disso, normalmente é interessante manter a excitação do sistema em baixas frequências, com o intuito de evitar excitar dinâmicas indesejadas para a identificação. Duas das estratégias mais utilizadas estão relacionadas nos itens a seguir (MELEIRO, 2002).

- Seja $v(t)$ um sinal correspondente a um ruído branco. O sinal de entrada utilizando o ruído pode ser definida como mostra a equação 3.1.

$$u(k) = v(k) \left(\text{int} \left[\frac{k-1}{N} \right] \right), k = 1, 2, \dots \quad (3.1)$$

Onde k é um índice correspondente ao tempo, N é o número máximo de amostras, o operador int é a parte inteira dos elementos no seu interior e u é o sinal resultante da operação.

- Uma extensão do método para geração de PRS pode ser obtida introduzindo-se uma variável aleatória adicional que determine quando o sinal de entrada deve ser alterado. A introdução desse conceito pode ser observada na equação 3.2.

$$u(k) = \begin{cases} u(k-1) & \text{com probabilidade } \alpha \\ v(k) & \text{com probabilidade } 1 - \alpha \end{cases} \quad (3.2)$$

3.3 Modelos Polinomiais Lineares

Após escolher o sinal a ser aplicado no sistema e o período de amostragem mais adequado (o que equivale ao item Entrada e Saída) o próximo passo é escolher o modelo.

Há diversas formas de representar modelos lineares. Uma das mais utilizadas é a forma de funções de transferência definida como a transformada da resposta ao impulso do sistema modelado, para condições iniciais nulas. Se a resposta ao impulso for contínua no tempo, então, utiliza-se a transformada de *Laplace*. Se a resposta ao impulso for discreta no

tempo, a respectiva função de transferência é, por definição, a transformada Z (AGUIRRE, 2007).

Nas representações a seguir definiremos $A(z^{-1})$, $B(z^{-1})$, $C(z^{-1})$ e $D(z^{-1})$ conforme as equações 3.3 e 3.4 respectivamente.

$$A(z^{-1}) = 1 + a_1 z^{-1} + a_2 z^{-2} + \dots + a_{n_a} z^{-n_a} \quad (3.3)$$

$$B(z^{-1}) = 1 + b_1 z^{-1} + b_2 z^{-2} + \dots + b_{n_b} z^{-n_b} \quad (3.4)$$

$$C(z^{-1}) = 1 + c_1 z^{-1} + c_2 z^{-2} + \dots + c_{n_c} z^{-n_c} \quad (3.5)$$

$$D(z^{-1}) = 1 + d_1 z^{-1} + d_2 z^{-2} + \dots + d_{n_d} z^{-n_d} \quad (3.6)$$

Isto significa que os polinômios $A(z^{-1})$, $B(z^{-1})$, $C(z^{-1})$ e $D(z^{-1})$ representarão sistemas discretos no tempo.

O modelo que representará o sistema utilizado neste trabalho será o ARX (*Autoregressive with eXogenous Input*).

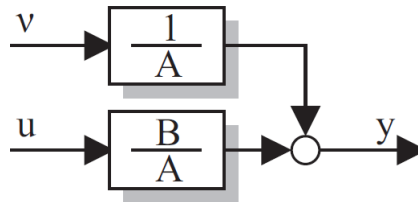
3.3.1 ARX

Sendo $A(z^{-1})$ e $B(z^{-1})$ polinômios arbitrários, o modelo ARX assume a forma da equação 3.7.

$$y(k) = \frac{B(z^{-1})}{A(z^{-1})} u(k) + \frac{1}{A(z^{-1})} v(k) \quad (3.7)$$

O modelo ARX (*AutoRegressive with eXogenous inputs*) apresentado no diagrama de blocos da figura 3, pode ser obtido a partir de um modelo polinomial linear mais geral de entradas e saídas resultando na equação 3.7 (AGUIRRE, 2007).

Figura 3 - Diagrama de Blocos do Modelo ARX. Fonte: Isermann e Munchhof (2011)



3.4 Dificuldades Relacionadas à Identificação

O conceito de identificabilidade é um dos problemas centrais em identificação. Esse conceito consiste em se analisar em quais condições o procedimento de identificação retornará uma solução única para os parâmetros do modelo escolhido, e, qual modelo resultante é igual, ou mais semelhante, ao sistema real. Isso envolve aspectos como que tipo de sinais de entrada são suficientemente informativos para distinguir se diferentes modelos são adequados ou não para descrever as dinâmicas do processo.

É possível que um método de estimação não retorne um modelo capaz de representar adequadamente um sistema. Uma das possíveis razões é a presença de perturbações externas, que podem inviabilizar a obtenção de um conjunto único de parâmetros que garantam uma representação fiel do sistema de interesse. Além disso, algumas abordagens podem não ser eficazes em problemas práticos, pois, estes podem conter ruído e/ou dinâmicas não modeladas (CAMPELLO et al., 2007).

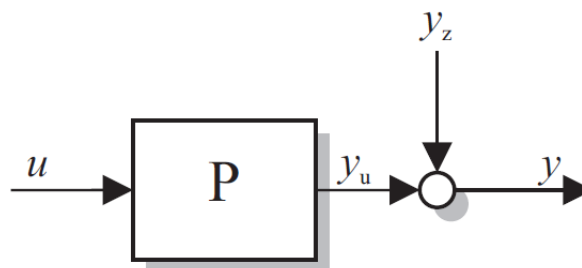
Segundo Rohrs et al. (1982) sinais coletados de plantas reais sempre conterão dinâmicas não modeladas de alta frequência e pequenos atrasos e, portanto, não existe um limite superior no número de zeros e polos de uma planta. Também, os sistemas reais sempre estarão sujeitos a perturbações aditivas, não medidas, presentes na saída, mesmo que muito pequenas. Ou seja, potencialmente um sistema pode ter a obtenção de um conjunto único de parâmetros que possam representar fielmente o sistema real escolhido.

Para que seja possível contornar tais problemas é necessário estudar os efeitos de cada um desses problemas buscando minimizá-los.

3.4.1 Presença de Ruídos

Os sinais de saída de vários processos não contém apenas a resposta ao sinal de teste (sinal desenvolvido no projeto experimental), mas também neles pode conter ruído (RUSCIO, 1995) como mostrado no diagrama 4.

Figura 4 - Diagrama de blocos de um processo sujeito a ruído. Fonte: Adaptado de Isermann e Munchhof (2011)



Esse ruído interfere no sinal de saída por algumas razões particulares. O ruído pode

ser causado por perturbações externas atuando no processo ou por perturbações internas localizadas dentro dos limites do processo. Identificar um processo utilizando um sinal de teste só é normalmente possível se o ruído for de pequena amplitude comparado ao sinal projetado e se o ruído possuir média constante. Caso um ruído de propriedades desconhecidas esteja atuando no sistema é praticamente impossível obter qualquer modelo útil resultante de uma identificação em um tempo relativamente curto de aquisição de dados. Nesse caso, é necessário esperar um período de tempo no qual a média do sinal de ruído seja percebida constante ou buscar métodos mais avançados de identificação (ISERMANN; MUNCHHOF, 2011).

3.4.2 Presença de Dinâmicas Não Modeladas

Os modelos nominais, ou simplificados, não representam as plantas com total fidelidade, ou por diferenças nas medições do comportamento do sinal de saída do sistema na etapa de modelagem, ou por simplificações no modelo que não comportam comportamentos não lineares nem variância no tempo. É importante distinguir incertezas nos valores dos parâmetros do modelo principalmente quando o modelo é utilizado para controle. As incertezas agregadas aos parâmetros do modelo podem ser características de alta frequência, efeitos de componentes parasitas, e a classificação de tais características permitiram uma série de trabalhos associados a robustez de controladores baseados em modelos (OLIVEIRA, 2002).

3.5 Estimação de Parâmetros

Com o objetivo de prever as respostas futuras de um processo dinâmico ou manipular suas saídas para seguir trajetórias desejadas, um modelo matemático que descreve as dinâmicas desse sistema é necessário. A estimação de parâmetros pode ser realizada utilizando vários métodos tais como mínimos quadrados ((AGUIRRE, 2007)), mínimos quadrados recursivo (Lindoff e Holst (1999)), mínimos quadrados estendido (Ettaleb et al. (1998), Yam e Chow (1997), Wenxiao (2007), Cohen e Tuninsky (1995)), mínimos quadrados com compensação de polarização (Söderström et al. (2005), Ding et al. (1999), Mahata (2007), Zhang e Cui (2011) e Ding e Ding (2011)). Um importante aspecto das técnicas que utilizam mínimos quadrados é o tratamento apropriado de dados discrepantes (COHEN; TUNINSKY, 1995). Esses algoritmos de estimação de parâmetros em sua essência resolvem o problema de minimização do erro quadrático. Eles serão mostrados nas seguintes subseções.

3.5.1 Estimador de Mínimos Quadrados - MQ

O método dos mínimos quadrados foi desenvolvido por Gauss em 1795 e é um método largamente utilizado na solução de problemas de otimização linear. Quando esse método

é aplicado em identificação de sistemas o seu objetivo é encontrar a saída do modelo que melhor se aproxime dos dados coletados da saída do processo minimizando a função de custo da soma do erro quadrático médio (erro entre a saída real e a estimada) (NELLES, 2001). Nesta seção será descrito o algoritmo de mínimos quadrados utilizando modelos ARX entendendo que é possível expandir tal modelo para as formas ARMAX, OE, NARX, NARMAX.

A partir do mostrado na figura 3 e na equação 3.7 e utilizando as equações 3.3, e 3.4 é possível definir o modelo ARX em sua forma de regressores conforme a equação 3.8.

$$y(k) = \varphi^T(k)\theta + e(k) \quad (3.8)$$

Onde:

$$\varphi(k) = [-y(k-1), -y(k-2), \dots, -y(k-n_a), u(k-1), u(k-2), \dots, u(k-n_b)]^T \quad (3.9)$$

E:

$$\theta = [a_1, a_2, \dots, a_{n_a}, b_1, b_2, \dots, b_{n_b}]^T \quad (3.10)$$

A partir da equação 3.8 definimos:

$$\Psi = \begin{bmatrix} \varphi^T(1) \\ \varphi^T(2) \\ \vdots \\ \varphi^T(N) \end{bmatrix} \quad (3.11)$$

$$Y = \begin{bmatrix} y(1) \\ y(2) \\ \vdots \\ y(N) \end{bmatrix} \quad (3.12)$$

$$\Xi = \begin{bmatrix} e(1) \\ e(2) \\ \vdots \\ e(N) \end{bmatrix} \quad (3.13)$$

Onde N é o número de amostras coletadas.

Com isso é possível representar a equação 3.8 na forma matricial da equação 3.14:

$$Y = \Psi\theta + \Xi \quad (3.14)$$

A partir da equação 3.14 é possível obter a solução de mínimos quadrados dada pela equação 3.15.

$$\hat{\theta}_{MQ} = (\Psi^T \Psi)^{-1} \Psi^T Y \quad (3.15)$$

Na equação 3.15 a expressão apresentada é chamada de pseudo-inversa da matriz Ψ . Se Ψ for uma matriz de posto completo, o que implica que pelo menos ela possui tantas medidas quanto parâmetros a encontrar, a matriz pode ser invertida e a solução de mínimos quadrados encontrada.

3.5.2 Estimador de Mínimos Quadrados Recursivo - MQR

Identificação de parâmetros em sistemas estocásticos constantemente fazem uso de técnicas de estimação de parâmetros recursivas. Isso acontece principalmente quando o sistema que se deseja estudar é não linear ou variante no tempo, ou, ainda, quando a estimação *off-line* demanda muito tempo de processamento computacional ou memória. A solução de mínimos quadrados recursiva é um algoritmo genérico. Suas propriedades já são bem consolidadas quando utilizado em sistemas variantes no tempo e nesse caso se utiliza uma penalização ou fator de esquecimento. (LINDOFF; HOLST, 1999).

O estimador de mínimos quadrados analisa as propriedades de uma variável aleatória θ utilizando equações a diferenças do processo (COELHO; COELHO, 2004). As propriedades mais importantes nesse método são que o estimador é não polarizado, ou seja, os parâmetros estimados convergem para os verdadeiros quando o número de iterações aumenta, a precisão das estimativas é estabelecida pelo valor inicial da matriz de covariância e pelo fato de o resíduo ser branco com média nula (LJUNG; SÖDERSTRÖM, 1983).

Os autores Coelho e Coelho (2004) descrevem um algoritmo simples para estimação em mínimos quadrados descrito a seguir:

- Mede-se a saída e entrada do sistema;
- Atualiza-se o vetor de medidas $\phi(t-1)$;
- Calcula-se o erro de previsão $\xi(t) = y(t) - \theta\phi(t-1)$;
- Calcula-se o ganho do estimador utilizando-se da equação 3.16;

$$K(t) = \frac{\phi(t-1)p(t)}{\lambda + p^T(t)\phi p(t)} \quad (3.16)$$

- Calcula-se o vetor de parâmetros estimados θ como em 3.17;

$$\hat{\theta}(t+1) = \hat{\theta}(t) + K(t)\xi(t) \quad (3.17)$$

- Calcula-se a matriz de covariância como em 3.18;

$$\phi(t) = \frac{1}{\lambda}\phi(t-1) - K(t)[\phi(t-1)p(t)]^T \quad (3.18)$$

3.5.3 Estimador de Mínimos Quadrados Estendido - MQE

O algoritmo para estimar o modelo do processo utilizando o MQE pode ser visualizado a seguir.

Algoritmo 4 Estimador de Mínimos Quadrados Estendido

- 1: Montar a matriz de regressores Ψ como na equação 3.11, e determinar a solução de mínimos quadrados mostrada na equação 3.15.
- 2: Calcular o vetor de resíduos Ξ como na equação 3.19.

$$\Xi = Y - \Psi\hat{\theta}_{MQ} \quad (3.19)$$

- 3: Utilizando o vetor de resíduos calculado, utilizá-lo para montar a matriz estendida de regressores mostrada na equação 3.20.

$$\Psi_{MQE} = \begin{bmatrix} \varphi_{MQE}(1) \\ \varphi_{MQE}(2) \\ \vdots \\ \varphi_{MQE}(N) \end{bmatrix}^T \quad (3.20)$$

Onde:

$$\varphi_{MQE}(k) = [-y(k-1), -y(k-2), \dots, -y(k-n_a), u(k-1), u(k-2), \dots, u(k-n_b), e(k-1), e(k-2), \dots, e(k-n_c)]^T \quad (3.21)$$

- 4: Estimar os parâmetros $\hat{\theta}_{MQE}$ a partir da equação 3.22.

$$\hat{\theta}_{MQE} = (\Psi_{MQE}^T \Psi_{MQE})^{-1} \Psi_{MQE}^T Y \quad (3.22)$$

Onde os parâmetros de θ_{MQE} são correspondentes aos polinômios $A(z^{-1})$, $B(z^{-1})$ e $C(z^{-1})$ conforme a equação 3.23.

$$\theta_{MQE} = [a_1, a_2, \dots, a_{n_a}, b_1, b_2, \dots, b_{n_b}, c_1, c_2, \dots, c_{n_c}]^T \quad (3.23)$$

- 5: Recalcular o vetor de resíduos utilizando a equação 3.24

$$\Xi = Y - \Psi\hat{\theta}_{MQE} \quad (3.24)$$

- 6: Volte para 3 e repita até que um critério de convergência (em relação ao resíduo) seja satisfeito.
-

3.5.4 Estimador de Mínimos Quadrados Recursivo com Compensação de Polarização - BCRLS

Em aplicações reais como em controle de processos, as saídas observadas estão sujeitas a perturbações no processo e erros de observação os quais, consequentemente, podem resultar em ruídos coloridos (DING; DING, 2011). Como analisado por Lindoff e Holst

(1999), e Aguirre (2007) algoritmos que utilizam mínimos quadrados recursivo e mínimos quadrados puro são sensíveis a ruídos e resultam em parâmetros polarizados. Por isso, métodos que visam eliminar essa polarização foram estudados por Söderström et al. (2005), Ding et al. (1999), Mahata (2007), Zhang e Cui (2011) e Ding e Ding (2011).

Sabe-se que os métodos de correção e eliminação de polarização são maneiras efetivas de obter parâmetros não polarizados em sistemas estocásticos (sujeitos a incertezas variantes no tempo). Consequentemente, essas técnicas são de grande importância quando utilizadas em sistemas práticos (ZHANG; CUI, 2011).

Dentre os métodos listados na bibliografia será descrito neste trabalho o método mais recente, que trabalha com algoritmos recursivos de compensação de polarização encontrado em Ding e Ding (2011). Esse método foi escolhido devido a atualização dos parâmetros ser iterativa. Isso faz com que os algoritmos recursivos tenham vantagem em relação aos métodos não recursivos quando aplicados na prática.

Consideremos polinômios $A(z^{-1})$, $B(z^{-1})$ e $D(z^{-1})$ das equações 3.3, 3.4, 3.6:

Utilizando-os na forma de um sistema descrito por um modelo OE com ruído colorido temos:

$$y(t) = \frac{B(z^{-1})}{A(z^{-1})}u(t) + e(t) \quad (3.25)$$

Onde:

- $u(t)$ é a sequência de entradas;
- $y(t)$ é a sequência de saídas;
- $e(t)$ é a saída do resíduo colorido com média zero e variância desconhecida.

A variável $e(t)$ pode ser descrita como um processo de média móvel da seguinte maneira:

$$e(t) = D(z^{-1})v(t) \quad (3.26)$$

Onde $v(t)$ é um ruído branco com média zero e variância desconhecida σ^2

Além das variáveis descrita seja o modelo do ruído colorido dado por:

$$w(t) = A(z^{-1})e(t) = A(z^{-1})D(z^{-1})v(t) \quad (3.27)$$

Então, a partir das equações 3.25, 3.26 e 3.27 é possível obter as formas regressivas do modelo da saída $y(t)$ do processo, do modelo do ruído branco $e(t)$ e do modelo do ruído colorido $w(t)$, respectivamente, como mostrado nas equações 3.28, 3.29 e 3.30.

$$y(k) = \varphi^T(k)\theta + w(k) \quad (3.28)$$

$$e(k) = \varphi_n^T(k)\theta_n + v(k) \quad (3.29)$$

$$w(k) = \psi^T(k)\theta + e(k) \quad (3.30)$$

Onde os vetores de parâmetro do sistema são dados por:

$$\theta = [a_1, a_2, \dots, a_{n_a}, b_1, b_2, \dots, b_{n_b}]^T \quad (3.31)$$

$$\theta_n = [d_1, d_2, \dots, d_{n_d}]^T \quad (3.32)$$

E os vetores de informação são:

$$\varphi(k) = [-y(k-1), -y(k-2), \dots, -y(k-n_a), u(k-1), u(k-2), \dots, u(k-n_b)]^T \quad (3.33)$$

$$\psi(k) = [e(k-1), e(k-2), \dots, e(k-n_a), \vec{0}_{1 \times n_b}]^T \quad (3.34)$$

$$\varphi_n(k) = [v(k-1), v(k-2), \dots, v(k-n_d)]^T \quad (3.35)$$

A partir dessas informações podemos sintetizar as equações necessárias para executar o algoritmo de compensação de polarização.

A estimação recursiva não polarizada, como mostrado por [Ding e Ding \(2011\)](#), é sintetizado no seguinte algoritmo:

Algoritmo 5 Estimador de Mínimos Quadrados Recursivo com Compensação de Polarização - Parte 1

- 1: Coletar os dados de entrada e saída e escolher os valores de n_a , n_b e n_d .
- 2: Inicializar a matriz de covariância $P(0) = p_0 I$, os valores $p_0 = 10^6$, $\hat{\theta}_{LS}(0) = \hat{\theta}_c(0) = \frac{\vec{1}}{p_0}$ e o valor inicial do cálculo da função objetivo $J(0) = 0$.
- 3: Montar o vetor a partir dos dados de medida formando a equação 3.33.
- 4: Calcular o valor da função objetivo como na equação 3.36 .

$$J(k) = J(k-1) + \frac{[y(k) - \varphi^T(k)\hat{\theta}_{LS}(k-1)]^2}{1 + \varphi^T(k)P(k-1)\varphi(k)} \quad (3.36)$$

- 5: Calcular $P(k)$ e $\hat{\theta}_{LS}(k)$ utilizando as equações 3.37 e 3.38 respectivamente.

$$P(k) = \frac{1}{\lambda} \left[P(k-1) - \frac{P(k-1)\varphi(k)\varphi^T(k)P(k-1)}{\lambda + \varphi^T(k)P(k-1)\varphi(k)} \right] \quad (3.37)$$

$$\hat{\theta}_{LS}(k) = \hat{\theta}_{LS}(k-1) + P(k)\varphi(k)[y(k) - \varphi^T(k)\hat{\theta}_{LS}(k-1)] \quad (3.38)$$

- 6: Montar os vetores de ruído branco estimado $\varphi_n(k)$ e ruído colorido estimado segundo as relações 3.39 e 3.40 respectivamente.

$$\varphi_n(k) = [\hat{v}(k-1), \hat{v}(k-2), \dots, \hat{v}(k-n_d)]^T \quad (3.39)$$

$$\psi(k) = [\hat{e}(k-1), e(k-2), \dots, \hat{e}(k-n_a), \vec{0}_{1 \times n_b}]^T \quad (3.40)$$

- 7: Computar os valores estimados para o ruído branco e colorido dado nas equações 3.41, 3.42 respectivamente.

$$\hat{v}(k) = \hat{e}(k) - \varphi_n^T(k)\hat{\theta}_n \quad (3.41)$$

$$\hat{e}(k) = y(k) - \varphi^T(k)\hat{\theta}_{LS}(k) - \hat{\psi}^T(k)\hat{\theta}_{LS}(k) \quad (3.42)$$

- 8: Estimar os parâmetros do ruído utilizando as equações 3.43 e 3.44.

$$P_n(k) = \frac{1}{\lambda_n} \left[P_n(k-1) - \frac{P_n(k-1)\hat{\varphi}_n(k)\hat{\varphi}_n^T(k)P_n(k-1)}{\lambda_n + \hat{\varphi}_n^T(k)P_n(k-1)\hat{\varphi}_n(k)} \right] \quad (3.43)$$

$$\hat{\theta}_n(k) = \hat{\theta}_n(k-1) + P_n(k)\hat{\varphi}_n(k)[\hat{e}(k) - \hat{\varphi}_n^T(k)\hat{\theta}_n(k-1)] \quad (3.44)$$

Algoritmo 6 Estimador de Mínimos Quadrados Recursivo com Compensação de Polarização - Parte 2

9: Usando 3.45, 3.46, 3.36, 3.47, 3.48 e $\hat{\theta}_n(k)$, computar $\hat{\sigma}^2(k)$ usando 3.49

$$\frac{\hat{r}_i}{\hat{\sigma}^2} = \hat{\theta}_n^T(i : n_d) \begin{bmatrix} 1 \\ \hat{\theta}_n^T(i : n_d - i) \end{bmatrix}, i = 1, 2, \dots, n_d \quad (3.45)$$

$$\hat{M}(k) = \begin{bmatrix} \begin{bmatrix} 1 + \hat{\theta}_n^T \hat{\theta}_n & \frac{\hat{r}_1}{\hat{\sigma}^2} & \dots & \frac{\hat{r}_{n_a-1}}{\hat{\sigma}^2} \\ \frac{\hat{r}_1}{\hat{\sigma}^2} & 1 + \hat{\theta}_n^T \hat{\theta}_n & \dots & \frac{\hat{r}_{n_a-2}}{\hat{\sigma}^2} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\hat{r}_{n_a-1}}{\hat{\sigma}^2} & \frac{\hat{r}_{n_a-2}}{\hat{\sigma}^2} & \dots & 1 + \hat{\theta}_n^T \hat{\theta}_n \end{bmatrix} & 0_{(n_a \times n_b)} \\ 0_{(n_b \times n_a)} & 0_{(n_b \times n_b)} \end{bmatrix} \quad (3.46)$$

$$\hat{\zeta} = [\frac{\hat{r}_1}{\hat{\sigma}^2}, \frac{\hat{r}_2}{\hat{\sigma}^2}, \dots, \frac{\hat{r}_{n_a}}{\hat{\sigma}^2}, \vec{0}_{(1 \times n_b)}]^T \quad (3.47)$$

$$\hat{\eta}(k) = \hat{\theta}_c^T(k) \hat{M}(k) \hat{\theta}_{LS}(k) + 1 + \hat{\theta}_n^T(k) \hat{\theta}_n(k) + \hat{\zeta}^T(k) [\hat{\theta}_c(k) + \hat{\theta}_{LS}(k)] \quad (3.48)$$

$$\hat{\sigma}^2(k) = \frac{\frac{1}{k} J(k)}{\hat{\eta}(k)} \quad (3.49)$$

10: Então, a estimativa da matriz de autocorrelação do ruído pode ser realizada sendo R mostrada na equação 3.50

$$\hat{R}(k) = \hat{\sigma}^2(k) \hat{M}(k) \quad (3.50)$$

11: E, com isso, $\hat{\gamma}(k)$ pode ser obtido como na equação 3.51 e finalmente calculado $\hat{\theta}_c(k)$ utilizando a equação 3.52

$$\hat{\gamma}(k) = \hat{\sigma}^2(k) \hat{\zeta}(k) \quad (3.51)$$

$$\hat{\theta}_c(k) = \hat{\theta}_{LS}(k) + kP(k) [\hat{R}(k) \hat{\theta}_c(k-1) + \hat{\gamma}(k)] \quad (3.52)$$

12: Caso K seja maior que a quantidade de iterações desejadas, finalizar a estimação, sendo o vetor final de estimativas não polarizadas $\hat{\theta}_c(k)$. Em caso contrário, incrementar k e retornar a 3.

4 METODOLOGIA, RESULTADOS E DISCUSSÕES

Nas seções seguintes serão apresentados dois testes simulados, um deles é o modelo apresentado em [Arruda et al. \(2011\)](#), o outro utiliza um sistema de segunda ordem no qual é inserido o efeito de uma dinâmica não modelada. O objetivo dessas seções é realizar um comparativo entre as técnicas apresentadas na revisão e mostrar o desempenho da técnica proposta.

4.1 Exemplo Simulado 1

Nesta subseção é apresentada a sequência de passos para identificar sistemas lineares utilizando os algoritmos de identificação MQ, MQR, MQE, BCRLS, a PSO convencional e a PSOM híbrida. A planta escolhida é similar à encontrada em [Arruda et al. \(2011\)](#) com a ausência do atraso de transporte. O sistema linear pode ser visualizado na equação 4.1, em que $y(k)$ e $u(k)$ são as saída e entrada do sistema, respectivamente.

$$\begin{aligned}
 y(k) = & 2,14y(k-1) - 1,3049y(k-2) \\
 & -0,0963y(k-3) + 0,3467y(k-4) - 0,1097y(k-5) \\
 & +0,0105y(k-6) + 0,01u(k-1) + 0,0074u(k-2) \\
 & +0,000924u(k-3) - 0,00001762u(k-4)
 \end{aligned} \tag{4.1}$$

Como mostrado em [Arruda et al. \(2011\)](#), os algoritmos determinísticos podem falhar se o modelo utilizado não é capaz de representar o sistema (ou planta), ou quando se deseja realizar uma aproximação de ordem inferior a do modelo. Por isso, nesse exemplo simulado, para identificar o sistema será utilizado um modelo de ordem reduzida. A partir desse modelo de ordem reduzida, será realizada uma comparação da performance dos algoritmos, utilizando o erro médio quadrático (MSE) obtido a partir da diferença entre os sinais de saída dos modelos real e estimado, e o erro médio quadrático obtido a partir do diagrama de Bode (soma de todas as magnitudes e fases em diferentes frequências) dos sistemas real e estimado. As equações utilizadas para realizar estes cálculos podem ser observadas nas equações 4.2, and 4.3.

$$MSE = \frac{\sum_{i=1}^{N_1} (y(i) - y_{estimado}(i))^2}{N_1} \tag{4.2}$$

Em que:

- y é a saída obtida a partir da simulação do sistema da equação 4.1;

- $y_{estimada}$ é a saída obtida a partir da simulação do sistema com os parâmetros identificados pelos algoritmos propostos. Esses parâmetros são quatro valores que aproximam uma função de segunda ordem no domínio discreto. O objetivo é aproximar a função 4.1 por um sistema de segunda ordem no domínio discreto;
- MSE é o erro médio quadrático;
- N_1 é o tamanho do vetor de saída (y).

$$E_{Bode} = \frac{\sum_{i=1}^{N_2} \left(\left| \frac{Bode_{abs} - RBode_{abs}}{Bode_{abs}} \right| + \left| \frac{Bode_{phase} - RBode_{phase}}{Bode_{phase}} \right| \right)^2}{N_2} \quad (4.3)$$

Em que:

- $Bode_{abs}$ e $RBode_{abs}$ são as magnitudes real e estimada do sistema (em dB) obtidas em frequências determinadas;
- $Bode_{phase}$ e $RBode_{phase}$ são as fases real e estimada do sistema obtidas em frequências determinadas;
- E_{Bode} é o erro médio quadrático obtido ao utilizar os dados do diagrama de Bode na equação 4.3;
- N_2 é a quantidade de dados do diagrama de Bode que foram coletados.

Vários testes foram realizados com o objetivo de realizar comparações entre os algoritmos. Nesses testes, os algoritmos foram inicializados de acordo com os seguintes parâmetros:

- A matriz de covariância dos algoritmos recursivos (BCRLS e MQR) foi inicializada como uma matriz identidade vezes 10^{11} ;
- A função custo utilizada pelos de otimização com a PSO é a apresentada na equação 4.2, em outras palavras, $J_{PSO} = MSE$, uma função unimodal;
- Os algoritmos que utilizam a PSO tradicional foram inicializados com os valores $\phi_1 = 0,3$, $\phi_2 = 0,7$, $\omega = 0,4$, 80 partículas, 1000 iterações e foram estabelecidos os limites no intervalo de busca dos parâmetros entre os valores $[-3,3]$;
- Para os algoritmos da PSOM (PSOMMQ e PSOMBCRLS), além de serem inicializados com os mesmos parâmetros da PSO tradicional, também foi utilizado o fator de esquecimento $\lambda_f = 0,99$ e $C = 0,1$, para a inicialização em uma região próxima aos algoritmos tradicionais (MQ e BCRLS);

- Foram realizados, com o objetivo de obter resultados consistentes, 80 repetições utilizando os algoritmos heurísticos avaliados. Em sequência, foram coletadas a média, variância e as melhores performances obtidas dentro das 80 repetições.

A escolha dos parâmetros apresentada permite que os autovalores da equação 2.37 sejam posicionados no interior do círculo de raio unitário fazendo com que o maior autovalor absoluto da matriz \bar{A} seja 0,9985. Isto significa que existe uma garantia de convergência quase certa para a PSO de acordo com o teorema apresentado. Os resultados são podem ser visualizados na tabela 1 e nas figuras 5 e 6.

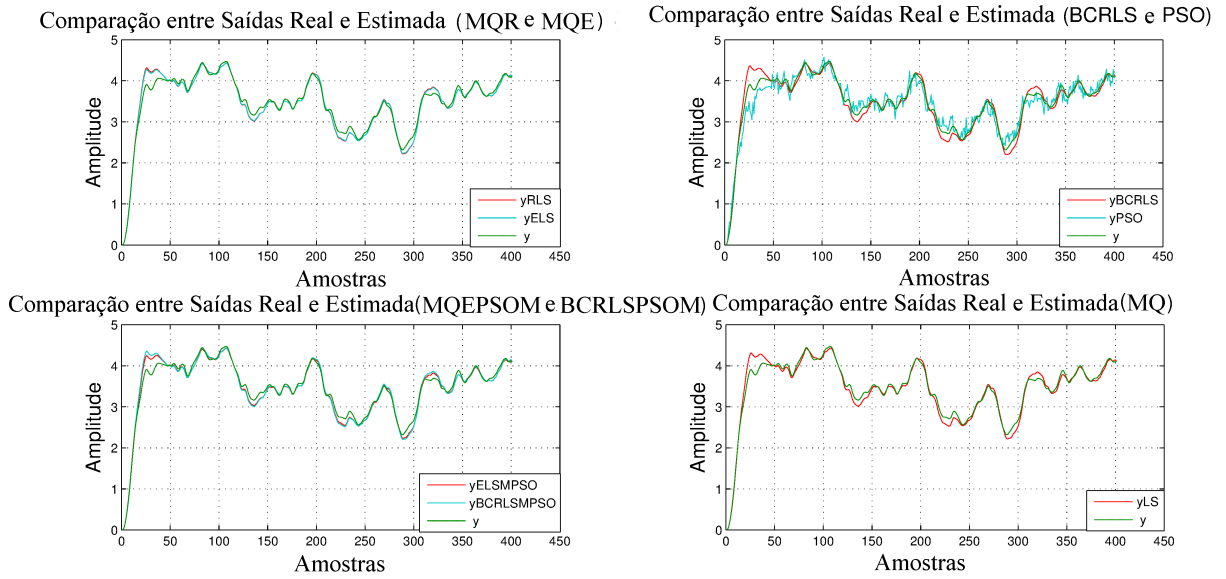


Figura 5 - Validação Comparando Saída Real e Estimada

Na tabela 1 (ao final do capítulo) foram apresentados seis índices para avaliar a performance dos algoritmos de estimação. O primeiro e o quarto índices o MSE_{av} e o $E_{Bode_{av}}$, são a média de todos os resultados obtidos a partir das 80 repetições (de obtenção de resultados, visto que os algoritmos heurísticos possuem uma variação nos resultados finais) utilizando a saída do sistema e os dados do diagrama de Bode. Os segundo e quinto índices, o MSE_{var} e o $E_{Bode_{var}}$, são as variâncias dos resultados obtidos utilizando a saída do sistema e os dados do diagrama de Bode. Os terceiro e último índices de avaliação, o $MSE_{min}(\%)$ e o $E_{Bode_{min}}(\%)$, foram obtidos utilizando a equação 4.4.

$$Result_{min}(\%) = 100 - \frac{100 \times |MaxResult - Result_{min}|}{MaxResult} \quad (4.4)$$

Em que, a variável $Result_{min}$ é o menor valor obtido do algoritmo em questão (com o objetivo de normalizá-lo e colocá-lo em porcentagem), e $MaxResult$ é o maior resultado obtido de todos os algoritmos (ou um valor máximo de referência) e o resultado pode ser

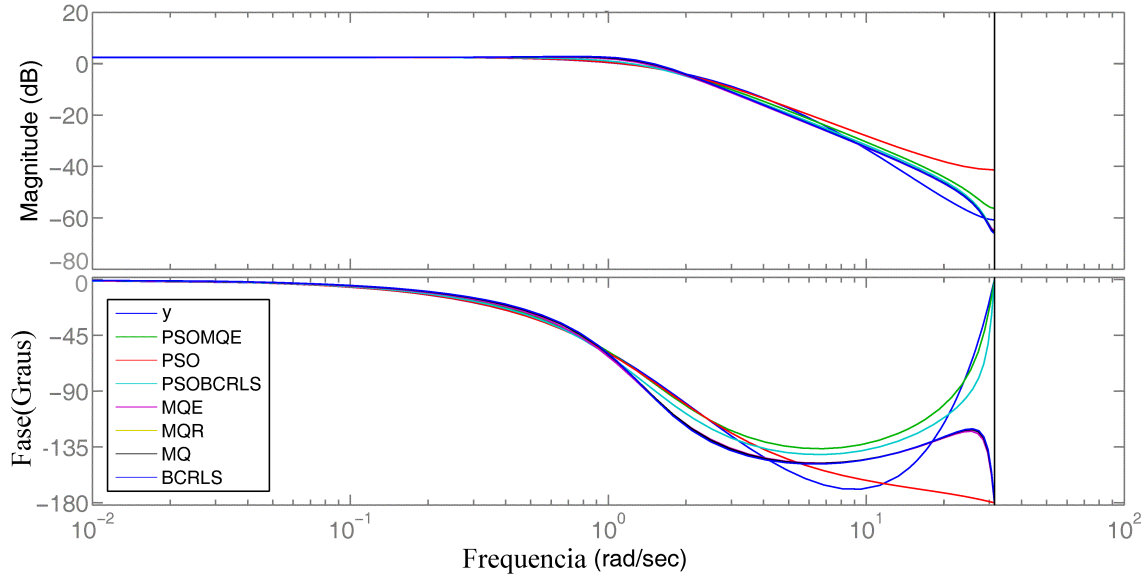


Figura 6 - Validação Comparando os Diagramas de Bode

obtido utilizando os critérios MSE or $Bode$.

Os resultados mostram que, com a configuração de parâmetros utilizada, os valores de erro MSE_{av} e o E_{Bodeav} são bem próximos. Por outro lado, quando se analisa os critérios MSE_{min} e $E_{Bodemin}$ é possível perceber uma diferença significativa entre os algoritmos híbridos, baseados na $PSOM$ e os outros algoritmos avaliados. Como esperado, todos os erros da PSO tradicional (MSE_{av} , E_{Bodeav} , MSE_{min} e $E_{Bodemin}$) foram maiores que os outros algoritmos avaliados. Isso aconteceu devido a uma escolha proposital dos parâmetros ϕ_1 , ϕ_2 e w de maneira a resultar em uma performance ruim e mostrar a eficiência da hibridização (no caso, manter os resultados em uma faixa aceitável, com desempenho tão bom ou melhor que os algoritmos determinísticos).

Quando se analisa a variância, é possível visualizar claramente que os algoritmos baseados na $PSOM$ híbrida resulta valores menores em comparação aos outros algoritmos heurísticos. Isso ocorre devido tanto a inicialização da $PSOM$ ser realizada utilizando os algoritmos tradicionais quanto devido à $PSOM$ propriamente, e a tendência do algoritmo em procurar em uma região próxima à partícula global, partículas que são inicializadas ao redor do resultado obtido pelos algoritmos tradicionais. Isso permite, também, que o algoritmo conserve a generalidade, representando o sistema em uma faixa razoável de frequências, observado com o diagrama de Bode, cujo erro permanece o mesmo (no diagrama de Bode) enquanto o MSE diminui. Como resultado, a resposta obtida nas diversas frequências são bem próximas e representa bem o sistema real.

4.1.0.1 Exemplo Simulado 2

Nesse segundo exemplo, foram realizados testes utilizando uma planta de segunda ordem sujeita a dinâmica não modelada de segunda ordem. Esse tipo de dinâmica foi utilizada em Rohrs et al. (1982) para mostrar que quando alguns sistemas de controle adaptativos são sujeitos a essas dinâmicas, eles podem se tornar instáveis, devido a identificação inadequada dos parâmetros do processo. Portanto, esses são aspectos importantes de serem analisados. A equação 4.5 pode ser visualizada na forma de função de transferência.

$$G(s) = \frac{2}{(s+1)(s+2)} \frac{229}{s^2+30s+229} \quad (4.5)$$

Ao escolher um período de amostragem igual a 0,1 é possível obter a função de transferência na forma discreta, como pode ser visualizado na equação 4.6.

$$\begin{aligned} y(k) = & 2,161 y(k-1) - 1,544 y(k-2) + 0,4098 y(k-3) \\ & - 0,03688 y(k-4) + 0,001024 u(k-1) + 0,006192 u(k-2) \\ & + 0,003207 u(k-3) + 0,0001416 u(k-4) \end{aligned} \quad (4.6)$$

De maneira similar ao realizado na seção 4.1, nesta seção foram realizados testes de maneira a realizar comparações entre os algoritmos apresentados. Nesses testes, os algoritmos foram inicializados de maneira similar ao utilizado na seção 4.1. Os únicos parâmetros que foram modificados para realizar o segundo teste foram os parâmetros relacionados a PSO. Os parâmetros que foram modificados são apresentados a seguir:

- Os algoritmos que utilizam como base a PSO tradicional foram inicializados com os seguintes valores: $\phi_1 = 0,3$, $\phi_2 = 2,7$, $\omega = 0,4$, 80 partículas, 1000 iterações e foi estabelecido uma limitação no espaço de busca dos parâmetros entre os valores $[-3, 3]$;
- Para os algoritmos na forma matricial (PSOMQE e o BCRLSMPSO), além dos valores que foram apresentados na inicialização dos algoritmos baseados na PSO, foi utilizado o fator de esquecimento $\lambda_f = 0,99$ e $C = 0,2$;
- Os teste que foram realizados, de maneira a obter os resultados, foram repetidos 80 vezes utilizando cada uma das heurísticas implementadas. Após a obtenção dos 80 resultados, estes foram utilizados para obter a variância, média e a melhor performance encontradas durante os testes.

A escolha dos parâmetros não permitem que os autovalores da equação 2.37 permaneçam no interior do círculo de raio unitário, ou seja, que todos os autovalores sejam menores ou iguais a 1. Porém, o valor do módulo do maior autovalor da matriz \bar{A} permanece no interior do círculo. Isto significa que, existe uma garantia de convergência quase

certa com a PSO, de acordo com o teorema apresentado anteriormente, pois, o maior autovalor em módulo 0,9991. Os resultados podem ser visualizados na tabela 2 ao final do capítulo.

Nesse segundo exemplo, os parâmetros escolhidos da *PSO* e da *MPSO* foram modificados, porque nos últimos parâmetros escolhidos não foram percebidas mudanças ou melhorias nos resultados finais obtidos com as heurísticas quando comparados com os algoritmos tradicionais. Portanto, neste exemplo, pode-se verificar que o fato de alguma das partes não possuir autovalores no interior do círculo de raio unitário não é determinante na estabilidade do sistema, o que deixa claro, experimentalmente a estabilidade das partículas quando a média dos autovalores está no interior do círculo de raio unitário. Em outras palavras, uma parte do sistema estar fora do círculo não significa que o sistema por completo é instável, porque os autovalores da equação 2.41 são menores que 1 em módulo. Portanto, os resultados experimentais mostram que as partículas permanecem em uma região finita. Porém, comparando os resultados com os obtidos no primeiro exemplo é possível perceber uma pequena melhoria.

Tabela 1 - Erro quadrático Médio (MSE) e Erro no Diagrama de Bode (E_{Bode}).

Algoritmos	BCRLS	MQE (PSOM)	BCRLS (PSOM)	MQE	MQ	MQR	PSO
MSE_{av}	$4,809 \times 10^{-3}$	$3,039 \times 10^{-3}$	$4,147 \times 10^{-3}$	$3,400 \times 10^{-3}$	$3,900 \times 10^{-3}$	$3,900 \times 10^{-3}$	$2,062 \times 10^{-2}$
MSE_{var}	—	$2,178 \times 10^{-7}$	$7,389 \times 10^{-7}$	—	—	—	$1,456 \times 10^{-5}$
MSE_{min} (%)	11,682	5,380	5,357	8,259	9,474	9,474	21,996
$E_{Bode_{av}}$ (%)	45,856	40,012	43,629	41,787	43,602	43,601	111,020
$E_{Bode_{var}}$ (%)	—	$5,260 \times 10^0$	$8,251 \times 10^0$	—	—	—	$7,247 \times 10^3$
$E_{Bode_{min}}$ (%)	5,571	4,201	4,612	5,077	5,298	5,298	9,334

Tabela 2 - Erro Quadrático Médio (MSE) e Erro do Diagrama de Bode (E_{Bode}).

Algoritmos	BCRLS	MQE (PSOM)	BCRLS (PSOM)	MQE	MQ	MQR	PSO
MSE_{av}	$1,106 \times 10^{-3}$	$8,534 \times 10^{-4}$	$1,100 \times 10^{-3}$	$8,396 \times 10^{-4}$	$9,761 \times 10^{-4}$	$9,764 \times 10^{-4}$	$5,804 \times 10^{-1}$
MSE_{var}	—	$2,665 \times 10^{-9}$	$1,162 \times 10^{-9}$	—	—	—	$1,138 \times 10^0$
MSE_{min} (%)	0,028	0,021	0,026	0,021	0,025	0,025	0,780
$E_{Bode_{av}}$ (%)	91,457	91,628	92,941	91,719	92,062	92,069	462,560
$E_{Bode_{var}}$ (%)	—	5,894	$1,518 \times 10^2$	—	—	—	$2,003 \times 10^6$
$E_{Bode_{min}}$ (%)	0,817	0,717	0,736	0,819	0,822	0,822	1,727

5 CONCLUSÕES

Este trabalho teve como objetivo principal utilizar a PSO como técnica de otimização aplicada a identificação de sistemas em duas plantas simuladas. Para que esse objetivo fosse alcançado, algumas análises e formulação precisaram ser realizadas, tais como a análise de convergência das partículas da PSO, a análise das condições necessárias para que a PSO tenha uma convergência em direção ao valor correto de identificação e a definição de uma inicialização para as partículas a partir de algoritmos determinísticos.

As análises apresentadas neste trabalho se diferenciaram das demais, principalmente no rearranjo da estrutura da PSO e na análise do erro no sentido quadrático. Além disso, foram aplicados teoremas de sistemas dinâmicos lineares por partes, que se mostraram coerentes com os resultados práticos.

Além disso, neste trabalho, foi apresentada uma revisão bibliográfica que relaciona os algoritmos heurísticos à área de identificação. Em conformidade com a revisão bibliográfica os algoritmos apresentados foram implementados. Esses algoritmos são o MQ, MQR, MQE, BCRLS, PSO tradicional, PSOM (híbrido com o MQE), o PSOM (híbrido com o BCRLS).

Além das implementações, foram realizados testes para analisar o desempenho dos algoritmos relacionados à identificação. Os resultados mostram que o sinal de teste aplicado é capaz de excitar o sistema, e obter resultados com baixo erro entre a saída real e estimada. Também, foi possível observar que quando o sistema é aproximado por um de segunda ordem o desempenho dos PSO híbridos são superiores, ou seja, possuem a melhor representação quando utilizado o diagrama de Bode, mesmo que o PSO tradicional estivesse minimizando a mesma função de custo.

Como trabalhos futuros, é sugerido que sejam realizados testes experimentais para aplicar os conceitos em um processo prático. Além disso, segundo essa linha, o algoritmo da PSO pode ser mais investigado para realização de sintonia de controladores e controle preditivo de processos (com aplicações experimentais).

REFERÊNCIAS

- AGUIRRE, L. A. **Introdução à Identificação de Sistemas**: Técnicas lineares e não-lineares aplicadas a sistemas reais. Belo Horizonte: UFMG, 2007. [12](#), [39](#), [41](#), [42](#), [43](#), [45](#), [50](#)
- ARRUDA, L. V. R.; ISFER, L. A. D.; LOPES, H. S. Identificação de parâmetros por enxame de partículas através do compromisso entre variância e polarização. In: **X SBAI - Simpósio Brasileiro de Automação Inteligente**. São João del-Rei - MG - Brasil: SBAI, 2011. X. [13](#), [14](#), [54](#)
- BARIJOUGH, S. M.; LEE, J. W. On stability characterization of discrete-time piecewise linear systems. In: **American Control Conference**. Marriott Waterfront, Baltimore, MD, USA: [s.n.], 2010. p. 916–921. [24](#), [30](#)
- BHATTACHARYA, S.; KONAR, A.; NAGAR, A. A lyapunov-based extension to pso dynamics for continuous function optimization. In: **Second UKSIM - European Symposium on Computer Modeling and Simulation**. Liverpool, England: IEEE, 2008. p. 28–33. [12](#), [14](#)
- BISHOP, R. R.; RICHARDS, G. G. Identifying induction machine parameters using a genetic optimization algorithm. In: **Southeastcon 1990**. Piscataway, NJ: IEEE, 1990. v. 2, p. 476–479. [14](#)
- CAMPELLO, R. J. G. B.; OLIVEIRA, G. H. C.; AMARAL, W. C. Identificação e controle de processos via desenvolvimentos em séries ortonormais. parte a: identificação. **Controle & Automação**, v. 18, 2007. [44](#)
- CHAKRABORTY, P.; DAS, S.; ABRAHAM, A.; SNASEL, V.; ROY, G. On convergence of multi-objective particle swarm optimizers. In: **IEEE Congress on Evolutionary Computation**. Barcelona, Spain: [s.n.], 2010. p. 1–8. [14](#), [24](#)
- CHEN, X.; LI, Y. A modified pso structure resulting in high exploration ability with convergence guaranteed. **IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS**, IEEE, v. 37, p. 1271–1289, 2007. [14](#)
- CHEN, Z.; ZHONG, Y.; LI, J. Parameter identification of induction motors using ant colony optimization. In: **Evolutionary Computation, 2008**. Hong Kong, China: IEEE, 2008. p. 1611–1616. [14](#)
- CLERC, M.; KENNEDY, J. The particle swarm-explosion, stability, and convergence in a multidimensional complex space. **IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION**, IEEE, v. 6, p. 58–73, 2002. [14](#), [20](#), [24](#)
- COELHO, A. A. R.; COELHO, L. S. **Identificação de Sistemas dinâmicos Lineares**. Florianópolis: UFSC, 2004. [47](#)
- COHEN, E. R.; TUNINSKY, V. S. The extended-least-squares treatment of correlated data. **IEEE TRANSACTIONS ON INSTRUMENTATION AND MEASUREMENT**, IEEE, v. 44, p. 475–478, 1995. [12](#), [45](#)

DANTAS, A. F. d. A.; LINHARES, L. L. d. S.; PINTO, J. E. M. G.; ARAÚJO, F. M. U. d.; MAITELLI, A. L. Evaluation and optimization of three-phase separator vessel controlled by pi controls applying particle swarm optimization. In: **10th Portuguese Conference on Automatic Control - CONTROLO'12**. Funchal, Madeira Island: CONTROLO'12, 2012. 14

DING, F.; CHEN, T.; QIU, L. Bias compensation based recursive least-squares identification algorithm for miso systems. **IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS**, IEEE, v. 53, p. 349–353, 1999. 12, 45, 50

DING, J.; DING, F. Bias compensation-based parameter estimation for output error moving average systems. **INTERNATIONAL JOURNAL OF ADAPTIVE CONTROL AND SIGNAL PROCESSING**, Wiley Online Library, v. 25, p. 1100–1111, 2011. 12, 45, 49, 50, 51

EMARA, H. M.; FATTAH, H. A. A. Continuous swarm optimization technique with stability analysis. In: **American Control Conference**. Boston, Massachusetts: IEEE, 2004. p. 2811–2817. 12, 14

ETTALEB, L.; DUMONT, G. A.; DAVIES, M. S. An extended off-line least-squares method for parameter identification and time delay estimation. In: **Conference on Decision & Control**. Tampa, Florida USA: IEEE, 1998. v. 37th, p. 3423–3428. 12, 45

FAN, W.; CUI, Z.; CHEN, Y.; TAN, Y. Nonlinear time-varying stability analysis of particle swarm optimization. In: **International Conference on Computational Aspects of Social Networks**. Taiyuan, China: [s.n.], 2010. p. 3–6. 14, 24

FUKUYAMA, Y.; YOSHIDA, H. A particle swarm optimization for reactive power and voltage control in electric power systems. In: **Proceedings of the Congress on Evolutionary Computation**. Seoul, Korea: [s.n.], 2001. v. 1, p. 87–93. 18

HANLON, B.; WANG, N.; EGERSTEDT, M.; MARTIN, C. Switched linear systems: Stability and the convergence of random products. **COMMUNICATIONS IN INFORMATION AND SYSTEMS**, International Press, v. 11, p. 327–342, 2011. 15, 26, 30

HEPPNER, F.; GRENDER, U. A stochastic nonlinear model for coordinated bird flocks. In: S. Krasner, Ed., *The Ubiquity of Chaos*. AAAS Publications, Washington, DC, USA, 1990. 17

HUANG, K. S.; WU, Q. H.; TURNER, D. R. Effective identification of induction motor parameters based on fewer measurements. **IEEE TRANSACTIONS ON ENERGY CONVERSION**, IEEE, v. 17, p. 55–60, 2002. 14

HUYNH, D. C.; DUNNIGAN, M. W. Parameter estimation of an induction machine using advanced particle swarm optimization algorithms. **IET Electric Power Applications**, IEEE, v. 4, p. 748–760, 2010. 12, 13, 14

ISERMANN, R.; MUNCHHOF, M. **Identification of Dynamic Systems: An introduction with applications**. Berlin: Springer, 2011. 11, 41, 42, 43, 44, 45

KADIRKAMANATHAN, V.; SELVARAJAH, K.; FLEMING, P. J. Stability analysis of the particle dynamics in particle swarm optimizer. **IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION**, IEEE, v. 10, p. 245–255, 2006. [14](#), [24](#)

KAMPISIOS, K.; ZANCHETTA, P.; GERADA, C.; TRENTIN, A. Identification of induction machine electrical parameters using genetic algorithms optimization. In: **Industry Applications Society Annual Meeting**. Edmonton, Alberta, Canadá: IEEE, 2008. v. 8, p. 1–7. [14](#)

KENNEDY, J.; EBERHART, R. C. Particle swarm optimization. In: **Proceedings Of the IEEE International Conference in Neural Networks**. Western Australia: IEEE, 1995. p. 1942–1948. [17](#)

KENNEDY, J. F.; EBERHART, R. C.; SHI, Y. **Swarm Intelligence**. United States of America: Academic Press, 2001. [18](#)

KHOSLA, A.; KUMAR, S.; GHOSH, K. A comparison of computational efforts between particle swarm optimization and genetic algorithm for identification of fuzzy models. In: **NAFIPS**. San Diego, California, USA: IEEE, 2007. p. 245–250. [12](#), [13](#), [14](#)

LINDOFF, B.; HOLST, J. Convergence analysis of the rls identification algorithm with exponential forgetting in stationary arx-structures. **INTERNATIONAL JOURNAL OF ADAPTIVE CONTROL AND SIGNAL PROCESSING**, Wiley Online Library, v. 13, p. 1–22, 1999. [45](#), [47](#), [50](#)

LJUNG, L. **System identification: Theory for the user**. Upper Saddle River, NJ: Prentice Hall, 1987. [11](#), [38](#), [40](#)

LJUNG, L.; SÖDERSTRÖM, T. **Theory and Practice of Recursive Identification**. Massachusetts, USA: MIT press, 1983. [47](#)

LU, L.; LUO, Q.; LIU, J. y.; LONG, C. An improved particle swarm optimization algorithm. In: **International Conference on Granular Computing**. Hangzhou, China: IEEE, 2008. p. 486–490. [14](#)

MAHATA, K. An improved bias-compensation approach for errors-in-variables model identification. **Automatica**, ScienceDirect, v. 8, p. 1339–1354, 2007. [12](#), [45](#), [50](#)

MALIK, R. F.; RAHMAN, T. A.; HASHIM, S. Z. M.; NGAH, R. New particle swarm optimizer with sigmoid increasing inertia weight. **The International Journal of Computer Science and Security (IJCSS)**, v. 1, p. 43–52, 2007. [19](#)

MELEIRO, L. A. C. **Projeto e Aplicação de Controladores Baseados em Modelos Lineares, Neurais e Nebulosos**. 125 p. Tese (Engenharia Química) — Universidade Estadual de Campinas, Campinas, 2002. [42](#)

MENDES, R.; KENNEDY, J.; NEVES, J. The fully informed particle swarm: Simpler, maybe better. **IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION**, IEEE, v. 8, p. 204–210, 2004. [21](#)

MILLONAS, M. M. Swarms, phase transitions, and collective intelligence. In: **C. G. Langton, Ed., Artificial Life III**. MA: Addison Wesley, 1994. [17](#)

- MONTEIRO, L. H. A. **Sistemas Dinâmicos**. Brasil: Editora Livraria da Física, 2002. 25
- NAJJARZADEH, M.; AYATOLLAHI, A. A comparison between genetic algorithm and pso for linear phase fir digital filter design. In: **ICSP**. Beijing, China: IEEE, 2008. v. 9, p. 2134–2137. 12, 13, 14, 17
- NELLES, O. **Nonlinear system identification**: From classical approaches to neural networks and fuzzy models. Verlag Berlin Heidelberg New York: Springer, 2001. 12, 38, 46
- OLIVEIRA, J. B. **Estabilidade e Robustez de um Controlador Adaptativo Indireto por Modelo de Referência e Estrutura Variável**. 171 p. Tese (Engenharia Elétrica) — Universidade Federal do Rio Grande do Norte, Natal, 2002. 45
- REYNOLDS, C. W. Flocks, herds and schools: a distributed behavioral model. In: **Computer Graphic**. [S.l.: s.n.], 2007. v. 21, p. 25–34. 17
- ROBINSON, J.; SINTON, S.; RAHMAT-SAMII, Y. Particle swarm, genetic algorithm, and their hybrids: optimization of a profiled corrugated horn antenna. In: **Antennas and Propagation Society International Symposium**. Texas, USA: IEEE, 1994. v. 1, p. 314–317. 17
- ROHRS, C. E.; VALAVANI, L.; ATHANS, M.; STEIN, G. Robustness of adaptive control algorithms in the presence of unmodeled dynam. In: **Conference on Decision and Control**. [S.l.]: IEEE, 1982. v. 21, p. 3–11. 44, 58
- RUSCIO, D. D. **SUBSPACE SYSTEM IDENTIFICATION**: Theory and applications. Porsgrunn, Norway: Telemark Institute of Technology, 1995. 44
- SÖDERSTRÖM, T.; HONG, M.; ZHENG, W. X. Convergence properties of bias-eliminating algorithms for errors-in-variables identification. **INTERNATIONAL JOURNAL OF ADAPTIVE CONTROL AND SIGNAL PROCESSING**, Wiley Online Library, v. 19, p. 703–722, 2005. 12, 45, 50
- SERAPIÃO, A. B. S. Fundamentos de otimização por inteligência de enxames: Uma visão geral. **Revista Controle & Automação**, v. 20, n. 3, p. 271–304, 2009. 18, 19
- SHARIFI, A.; NOROOZI, V.; BASHIRI, M.; HASHEMI, A. B.; MEYBODI, M. R. Two phased cellular pso: A new collaborative cellular algorithm for optimization in dynamic environments. In: **World Congress on Computational Intelligence**. Brisbane, Australia: IEEE, 2012. p. 1–8. 13, 14
- SHI, Y.; EBERHART, R. A modified particle swarm optimizer. In: **Proceedings of IEEE International Conference on Evolutionary Computation**. Anchorage, Alaska: IEEE, 1998. v. 2, p. 66–73. 17, 19, 20
- _____. Empirical study of particle swarm optimization. In: **Proceedings of IEEE International Conference on Evolutionary Computation**. EDS Indianapolis Technol. Center, Carmel, IN: IEEE, 1999. v. 3, p. 1945–1950. 20

- SIERAKOWSKI C. A.; COELHO, L. S. Study of two swarm intelligence techniques for path planning of mobile robots. In: **IFAC World Congress**. Prague, República Checa: [s.n.], 2005. 17
- SONG, Y.; CHEN, Z.; YUAN, Z. New chaotic pso-based neural network predictive control for nonlinear process. **IEEE TRANSACTIONS ON NEURAL NETWORKS**, v. 18, n. 2, p. 595–600, 2007. 14
- TANG, H.; ZHANG, W.; FAN, C.; XUE, S. Parameter estimation using a clpso strategy. In: **CEC**. Washington, DC, USA: IEEE, 2008. p. 70–74. 12, 13, 14, 20
- UENO, G.; YASUDA, K.; IWASAKI, N. Robust adaptive particle swarm optimization. In: **International Conference on Systems, Man and Cybernetics**. Hawaii, USA: IEEE, 2005. v. 4, p. 3915–3920. 14
- WENXIAO, Z. Identification for hammerstein systems using extended least squares algorithm. In: **Chinese Control Conference**. Zhangjiajie, Hunan, China: IEEE, 2007. v. 26th, p. 241–245. 12, 45
- YAM, J. Y. F.; CHOW, T. W. S. Extended least squares based algorithm for training feedforward networks. **IEEE Transactions on Neural Networks**, IEEE, v. 8, p. 806–810, 1997. 12, 45
- ZENG G. END JIANG, Y. A modified pso algorithm with line search. In: **International Conference on Computational Intelligence and Software Engineering (CiSE)**. Wuhan, China: IEEE, 2010. p. 1–4. 13, 14
- ZHAN, Z. H.; ZHANG, J.; LI, Y.; CHUNG, H. S. H. Adaptive particle swarm optimization. **IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS**, IEEE, v. 39, p. 1362–1381, 2009. 14
- ZHANG, Y.; CUI, G. Bias compensation methods for stochastic systems with colored noise. **Applied Mathematical Modelling**, ELSERVIER, v. 35, p. 1709–1716, 2011. 12, 45, 50
- ZHENG, J.; ZHANG, G.; ZUO, H. Hybrid linear and nonlinear weight particle swarm optimization algorithm. In: **International Conference on Machine Learning and Cybernetics**. Xian, China: IEEE, 2012. v. 1, p. 1237–1241. 19
- ZHOU, X.; CHENG, H. The induction motor parameter estimation through an adaptive genetic algorithm. In: **Universities Power Engineering Conference**. Hong Kong, China: IEEE, 2004. v. 39, p. 494–498. 14
- ZHU, Y. **Multivariable System Identification for Process Control**. Eindhoven: Elsevier Science & Technology Books, 2001. 12, 38, 39