Algoritmo Genético e Rede Neural Artificial na Estimativa de Parâmetros em Oscilações Harmônicas

Fábio Dall Cortivo[1], **Leonardo Bacelar Lima Santos**[1]

[1]Programa de Doutorado em Computação Aplicada – CAP
Instituto Nacional de Pesquisas Espaciais – INPE

fabio.cortivo@lac.inpe.br,santoslbl@gmail.com

Abstract. *In this work we used a Genetic Algorithm and an Artificial Neural Network to estimate the damping (γ) and stiffness (k) coefficients of a spring-mass system. The inverse problem consists of to estimate, from a time series, the coefficients γ and k. The comparison between these two techniques is made on the results obtained in the estimation and on the computational time spent.*

Resumo. *Neste trabalho são utilizados um Algoritmo Genético (AG) e uma Rede Neural Artificial (RNA) do tipo Perceptron de Múltiplas Camadas para a determinar os coeficientes de amortecimento γ e a rigidez k de um sistema tipo massa-mola-amortecedor. O problema inverso consiste em determinar, a partir da série temporal de deslocamento, os coeficientes γ e k. A comparação entre as técnicas é feita com base nos resultados da estimativa dos parâmetros e custo computacional.*

Palavras-chave: *Algoritmos Genéticos, Redes Neurais Artificiais, Oscilações Harmônicas*

1. Introdução

Problemas de busca de solução ótima, frequentemente, tratam com conjuntos de cardinalidade muito alta (diversas soluções possíveis), inviabilizando processos exaustivos de procura. Para enfrentar tal questão, um caminho a seguir é a utilização de heurísticas [Bittencourt 2006]. Essas técnicas não garantem o resultado globalmente ótimo, mas, por usar informações do fenômeno durante a busca, reduzem significativamente o custo computacional, propiciando uma solução razoável a custo reduzido.

As heurísticas utilizadas neste trabalho são baseadas nas técnicas dos Algoritmos Genéticos (AGs) [Holland 1975] e das Redes Neurais Artificias (RNAs) [Haykin 2001], ambas advindas da área de Inteligência Artificial [Bittencourt 2006].

AGs são heurísticas bio-inspiradas que partem de um conjunto inicial de soluções e via operadores genéticos – seleção, cruzamento e mutação – vão atualizando o conjunto solução até que uma das soluções seja razoável, ou seja, apresente um erro menor que uma tolerância pré-estabelecida – ou um limite de iterações tenha sido alcançado. Essa técnica idealizada pelo americano John Henry Holland, representa uma classe particular de algoritmos evolutivos [Holland 1975].

RNAs são sistemais computacionais estruturados numa aproximação à computação baseada em ligações. Nós simples, chamados neurônios, são interligados para formar uma rede de nós. Os estudos pioneiros foram feitos pelo neurofisiologista Warren McCulloch, e o matemático Walter Pitts [McCulloch and Pitts 1943]. Nesse trabalho os autores fizeram uma analogia entre células nervosas vivas e o processamento eletrônico.

A técnica das RNAs é uma estratégia para solucionar problemas através da simulação do cérebro humano, inclusive em seu comportamento, assim sendo capaz de aprender, errar e fazer descobertas. Isto é, um modelo baseado na estrutura neuronal de organismos inteligentes e que adquirem experiência através do conhecimento. O trabalho de [Shiguemori et al. 2005] é um exemplo da aplicação de redes neurais artificiais para a resolução de um problema inverso em vibração.

O objetivo deste trabalho é fazer a estimativa simultânea dos parâmetros de rigidez e amortecimento de um sistema massa-mola-amortecedor e comparar os resultados obtidos pelas heurísticas (AG e RNA). Esse problema de estimação é encarado como um problema inverso [Campos Velho 2008], no qual os dados de entrada são os valores de uma série temporal de deslocamentos. Nessa comparação são avaliados itens como a qualidade das soluções e o tempo de processamento para determiná-las.

2. O Problema Abordado

A segunda lei de Newton para o problema abordado tem a forma

$$m\ddot{x}(t) + \gamma\dot{x}(t) + kx(t) = 0, \tag{1}$$

onde m, γ, e k são neste caso tomados constantes e definem a massa, o coeficiente de amortecimento e a constante de rigidez, respectivamente. As condições iniciais associadas ao problema, são dadas por

$$x(t_0) = a, \qquad \dot{x}(t_0) = b, \tag{2}$$

em que a e b são constantes.

Com o problema posto desta forma – equação diferencial ordinária de segunda ordem, unidimensional, linear, homogênea, a coeficientes constantes, é possível encontrar uma solução analítica pelo método da equação característica. Os detalhes para determinar a solução analítica (caso de raízes reais distintas ou repetidas ou raízes complexas) para o problema podem ser encontrados em [Boyce and DiPrima 2002].

3. Metodologia

O objetivo das técnicas utilizadas neste trabalho consiste em dada uma série temporal de deslocamentos, em que se conhece o valor da massa e das condições iniciais, determinar os coeficientes de amortecimento e rigidez. Isso caracteriza o que é chamado de problema inverso e, neste caso, trata-se de um problema inverso de estimativa de parâmetros [Neto and Neto 2005] no espaço de todas as combinações possíveis para os coeficientes γ e k definidos em um intervalo.

Os intervalos usados para esses parâmetros são $\gamma \in [1,\, 15]$ e $k \in [1,\, 25]$, para a massa adotou-se $m = 1$ e para a posição e velocidade inicial adotou-se $x(0) = 1$ e $\dot{x}(0) = 2$, respectivamente[1].

[1]Todas as grandezas serão aqui representadas adimensionalmente.

3.1. Algoritmos Genéticos

A questão central ao se trabalhar com AG é definir convenientemente o cromossomo e seus genes. Cromossomo é o elemento individual que compõe a população. Cada cromossomo deve estar relacionado a um e somente um elemento do espaço da busca. No nosso caso o cromossomo deve, portanto, ser o par coeficiente de amortecimento e rigidez, assim, cada cromossomo tem dois genes: o primeiro gene é o correspondente ao coeficiente de amortecimento, e o segundo à rigidez. Foi tomado cada um desses valores como um número real com precisão de duas casas decimais. Além disso, foi utilizada codificação binária, assim os genes serão escritos como sequências de zeros e uns, que serão convertidos a números reais quando no cálculo da função de aptidão: avaliação do quão boa é a solução proposta, ou seja, qual a distância entre o $x(t)$ experimental e o $x(t)$ correspondente aos valores estimados para os parâmetros.

O AG implementado trabalha respeitando a seguinte sequência de operações: $i)$ leitura do arquivo contendo os dados experimentais e que foram corrompidos com um ruído branco gaussiano multiplicativo de intensidade 5%; $ii)$ inicialização da população aleatoriamente; $iii)$ avaliação da população: se há algum individuo (cromossomo) cujo resíduo entre a série temporal (gerada por esse indivíduo) e os dados experimentais seja menor que uma tolerância pré-estabelecida, ou então, um número máximo de iterações foi alcançado; $iv)$ enquanto a resposta ao item anterior for negativa, aplica-se os operadores genéticos: seleção, cruzamento, mutação (consistiu em trocar os bits da cadeia de binários que representam cada cromossomo aleatoriamente seguindo a distribuição uniforme), e então, a população é atualizada e; $v)$ uma vez atingida a convergência, o cromossomo que representa o par de valores estimados tem seus genes transformados para base decimal e essa é a resposta do programa.

3.2. Perceptron de Múltiplas Camadas

A conexão de "vários" neurônios artificiais entre si compõe uma rede (além de outras) que é chamada de Perceptron de Múltiplas Camadas (MLP). Essa rede é constituída de uma camada de entrada, uma ou mais camadas ocultas, e uma camada de saída. Nesse modelo de rede, a informação é passada adiante camada por camada até atingir a camada de saída. A informação que chega em cada um dos neurônios (unidades de processamento) e que deverá ser processada por estes, é definida matematicamente por $v_i = \sum_j w_{ji}x_j + b_i$, em que x_j representa as informações de entrada, w_{ji} é o peso das conexões entre os neurônios (sinapses), e b_i é chamado de nível de bias, e pode ser interpretado como um valor que é ajustado a fim de "complementar" alguma atividade presente no neurônio biológico, mas não presente no neurônio artificial. Cada uma dessas unidades de processamento contém uma função de ativação, ϕ, que tem por finalidade restringir a amplitude do sinal de saída. Assim, o sinal que segue para o próximo neurônio (unidade de processamento) é definido como $y_i = \phi(v_i)$. Comumente são empregadas como funções de ativação: a função sigmoide, a função tangente hiperbólica e a função linear, definidas, respectivamente, por

$$\phi(v) = \frac{1}{1+e^{-av}}, \quad a > 0, \quad \phi(v) = \frac{1-e^{-av}}{1+e^{-av}}, \quad a > 0 \quad \text{e} \quad \phi(v) = av, \quad a \in \mathcal{R}^*.$$

Redes do tipo MLPs requerem um treinamento do tipo supervisionado, ou seja, com a presença de um supervisor o qual que tem o conhecimento do ambiente que a rede deverá

aprender. Esse ambiente é representado por um conjunto de entradas e saídas. Durante a fase de aprendizagem, o supervisor apresenta uma determinada entrada à rede, que por sua vez processa a informação apresentada e devolve uma saída. A saída calculada pela rede é comparada com a saída desejada, e com isso é gerado um sinal de erro, o qual é propagado para trás (retropropagação) através do algoritmo de treinamento. Durante a retropropagação do erro, o algoritmo de treinamento, baseado em alguma regra, faz uma correção nas variáveis livres da rede de modo a corrigir os valores dos seus parâmetros livres, visando minimizar o erro cometido. Dessa forma, espera-se que a rede aprenda a fazer a correta associação entre os padrões de entrada e os padrões de saída.

3.2.1. Geração dos Padrões, Treino e Validação da Rede

Os padrões necessários para treinar e validar a rede foram obtidos computacionalmente a partir da solução do problema definido na Seção 2, para as combinações inteiras de γ e k, como mostrado na Tabela 1.

Tabela 1. Conjunto de valores adotados para a geração dos padrões

Grupo	γ	k
A	1	$1, 2, 3, \ldots, 24, 25$
B	2	$1, 2, 3, \ldots, 24, 25$
$\vdots$	$\vdots$	$\vdots$
N	14	$1, 2, 3, \ldots, 24, 25$
O	15	$1, 2, 3, \ldots, 24, 25$

O número total de combinações/padrões possíveis são de 375, assim, tem-se um total de 375 séries temporais geradas a partir da solução da Eq. (1). Para a variável tempo, adotou-se o intervalo $t \in [0,\ 30]$ segundos, com medidas de um em um segundo, gerando assim, vetores de séries temporais com 31 posições. Esses 375 padrões foram embaralhados, e então, divididos em dois grupos e em cada grupo foi adicionado 2.5% e 5% de ruído (a fim de simular possíveis erros de medição). Posteriormente, do total de 375, escolheu-se aleatoriamente 80% desses padrões para formar o conjunto de treinamento (aprendizagem), 10% para formar o conjunto de validação (38 padrões) e os outros 10% para o conjunto de teste (37 padrões).

O treinamento da rede foi formulado como um problema de otimização definido como sendo a soma das diferenças quadráticas entre os parâmetros estimados pela rede e os dos padrões de treinamento e para o qual se buscou um valor mínimo, ou seja,

$$\mathcal{J}(\mathbb{E}_t, \mathbb{W}, \mathbb{B}) = \min \sum_{j=1}^{T} \sum_{i=1}^{M} \left|\left|O_{t_{ij}} - O_{n_{ij}}\right|\right|_2^2, \tag{3}$$

em que $\mathbb{E}_t$, $\mathbb{W}$ e $\mathbb{B}$ são matrizes que contém os padrões de entrada para o treinamento, os pesos e os bias, respectivamente. Já M e T representam, respectivamente, o número total de entradas de cada padrão de treinamento e o total de padrões do conjunto de treinamento. Por fim, $O_{t_{ij}}$ representa cada entrada da matriz que contém o conjunto de padrões

de saída para o treinamento, e $O_{n_{ij}}$ representa cada entrada da matriz de valores calculados pela rede, durante o treinamento. Outros detalhes acerca dessa formulação podem ser encontrados em [Dall Cortivo et al. 2012b, Dall Cortivo et al. 2012a].

3.3. Estudos de caso numérico

Foi analisado um conjunto de 75 pares (k,γ) para o AG e para a RNA (considerando os padrões de validação e teste) escolhidos aleatoriamente dentro da faixa de estudo, com o objetivo de ver a robustez da recuperação de parâmetros tendo como base diversos valores destes, e promovendo uma análise estatística dos resultados.

Além dos resultados da estimação, foram registrados os tempos de processamento para computação das soluções, afim de confrontar os esforços computacionais de cada método: AG e RNA.

Os seguintes parâmetros foram utilizados no AG: crossover de ponto único, seleção por roleta, elitismo de 5%, probabilidade de mutação 15% e o número de cromossomos igual a 100. A função objetivo utilizada foi a mesma utilizada pela RNA, ou seja, dada pela Eq. (3), a probabilidade de cruzamento foi de 100%, ou seja, sempre há cruzamento e critério de parada conforme mencionado anteriormente. Além disso, foram usados 200 pontos amostrais de deslocamento, indo do $t = 0.01$ a $t = 2$, em passos de $\Delta_t = 0.01$. O AG foi implementado na linguagem C++, sob o paradigma da Orientação a Objetos. O código foi baseado no Código Livre POGA [Santos and Miranda 2009] disponível para download em `http://sourceforge.net/projects/thepoga/`. O código foi compilado pelo compilador ICC (compilador INTEL para C++) e executado em ambiente GNU/Linux.

Já para a RNA foi utilizado uma camada de entrada com 31 neurônios (número de pontos amostrais no tempo), uma camada oculta com 31 neurônios, os quais fizeram uso da função tangente hiperbólica como função de ativação, já nos dois neurônios da camada de saída foi utilizado a função sigmoide como função de ativação. Os valores para as variáveis da rede foram escolhidos inicialmente de modo aleatório e atualizados para um "valor ótimo" durante o treinamento. A rede foi implementada em FORTRAN90 e o processo de otimização utilizado foi o quasi-Newton implementado na biblioteca NAG [NAG 1995]. A rede e a biblioteca foram compiladas com compilador INTEL IFORT em ambiente GNU/Linux.

A máquina usada para execução dos programas foi: Notebook Dell Vostro 1520 com processador Intel® Core™ 2 Duo P8600 2.40GHz L2 3Mb, 4Gb de memória RAM, sistema operacional GNU/Linux 64 bits (Ubuntu 11.04), compilador Intel® Fortran Composer XE. A compilação da rede e da biblioteca foi feita com a opção -O3.

4. Resultados

Na Figura 1 são exibidos o valor estimado e o real da rigidez, k, para amostras de experimentos executados via AG (a) e usando a RNA (b). Percebe-se que, na maioria dos casos, o valor estimado é bastante próximo ao valor real; e reordenando as amostras de acordo com o valor da rigidez (Figura 1(c) e 1(d)), pode-se notar que é mais comum superestimação para valores intermediários de rigidez e subestimação para os valores de rigidez mais elevados, tanto para a RNA quanto para o AG – sendo que a RNA apresenta alguns casos de subestimação para sistemas pouco rígidos. A Figura 2 traz resultados

análogos aos apresentados na Figura 1 só que com respeito à estimação do coeficiente de amortecimento.

A Tabela 2 congrega uma descrição estatística dos erros de estimação em $x(t)$ – distância euclideana entre $x(t)$ real e o $x(t)$ proporcionado pelos valores estimados para os parâmetros – para o AG e para as RNA.

Tabela 2. Descrição estatística dos resultados

	Resíduo Médio	Desvio Padrão	Resíduo Mínimo	Resíduo Máximo
AG	0.035	0.005	0.029	0.061
RNA	0.373	1.218	0.031	9.618

O tempo médio para processamento das soluções via AG foi de 0.88 segundos. Para as RNA o processo de treinamento demandou 540 segundos, mas uma vez treinada a RNA dá sua resposta para cada amostra da estimação de parâmetros em menos de 1μ segundo.

O número de chamadas à função fitness (resolução do problema direto) no AG foi de 1100 (20 iterações avaliando metade da população, mais a avaliação da população inicial completa): tais avaliações representam a maior demanda computacional de cada iteração de um AG. Por outro lado, o fato da RNA não precisar resolver o problema direto, pois faz o uso da solução deste (padrões de entrada), o tempo de computação é bastante inferior - todavia, para treinar a RNA é preciso um alto número de padrões já conhecidos do problema: relação $(k,\, \gamma) -> x(t)$, obtidos, por exemplo, através da resolução do problema direto.

5. Conclusões

O confronto entre as duas técnicas, AG e RNA, permite, para os estudos de caso aqui apresentados, constatar que o AG apresenta valores de resíduo médio uma ordem e resíduo máximo duas ordens de grandeza inferiores aos da RNA, enquanto que a grande vantagem da RNA é sua baixa demanda de tempo de processamento, uma vez treinada, chegando a precisar para o treinamento de apenas metade do tempo que o AG leva para ser executado.

Problemas inversos, fatalmente, não apresentam solução única [Campos Velho 2008]. Em aplicações práticas, casos reais, não se conhece o valor real dos parâmetros, pois este é justo o alvo da estimação, logo a avaliação da qualidade de uma solução inversa deve ser dada pela diferença, distância, entre o dado experimental e o obtido quando usando no problema direto os valores estimados para os parâmetros. Ambos os métodos mostraram-se robustos frente a presença de ruído no dado de entrada - neste dado experimental.

Vale a pena registrar que o uso da solução analítica não é recorrente na literatura, apesar de ser uma abordagem válida em alguns casos de problemas físicos reais quando sujeitos a condições especiais de controle. Comparar os métodos de problemas inversos usando um problema direto com solução analítica pode colaborar para um melhor entendimento do resultado comparativo, evitando possíveis efeitos dos métodos numéricos de aproximação para resolução do problema direto.

Dentre os próximos passos da investigação à qual reporta este artigo está a hibridização das técnicas: usar um AG no processo de treinamento da RNA, ou até mesmo considerar cada indivíduo da população do AG como uma RNA com uma dada arquitetura,

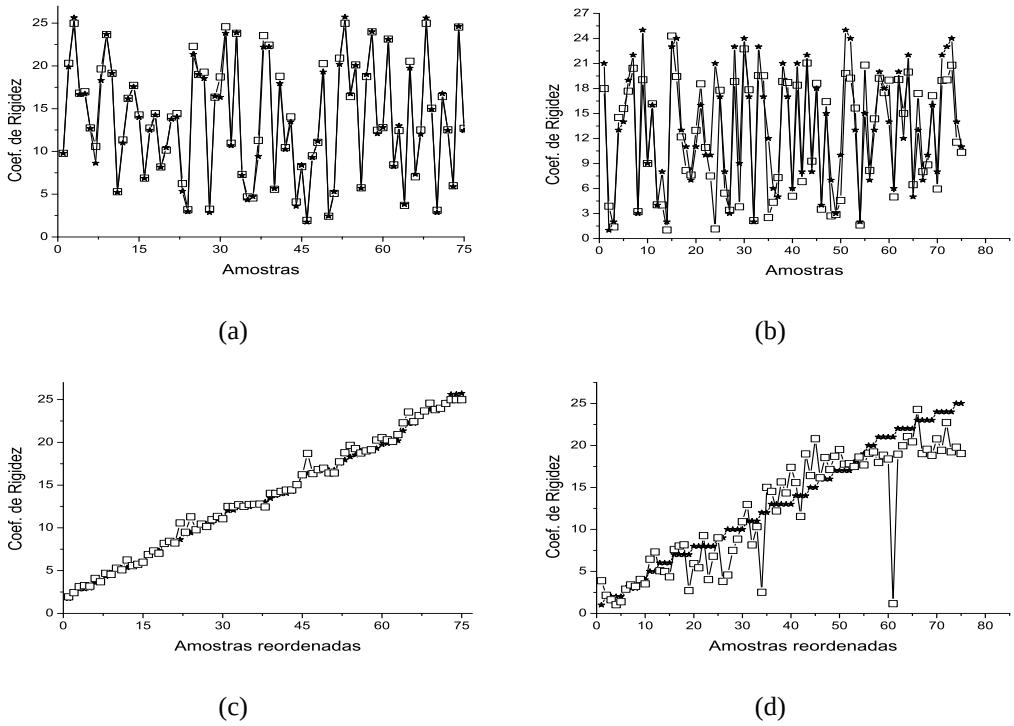

Figura 1. Resultados da estimação do parâmetro coeficiente de rigidez (k) via AG: com 75 amostras aleatoriamente escolhidas (figura a) e amostras ordenadas em ordem crescente do valor do parâmetro real (c); e via método das RNA com 75 amostras aleatoriamente escolhidas (figura b) e amostras ordenadas em ordem crescente do valor do parâmetro real (d). Em todos os casos o valor estimado é representado com quadrados vazados e os reais com estrelas preenchidas.

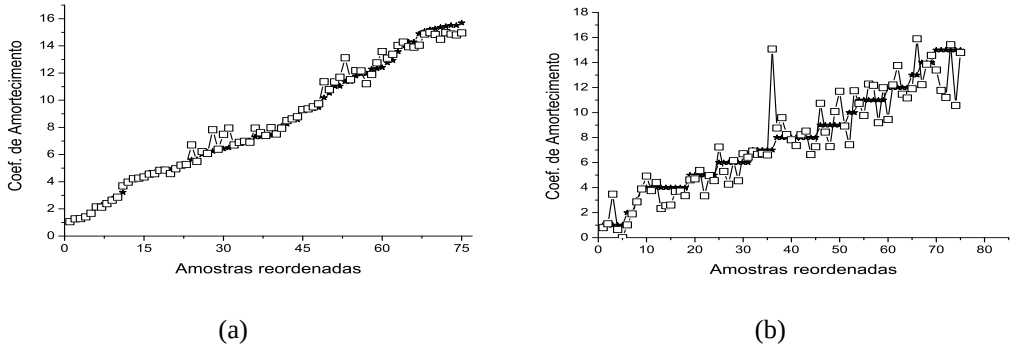

Figura 2. Resultados da estimação do parâmetro coeficiente de amortecimento (γ) com 75 amostras ordenadas em ordem crescente do valor do parâmetro real, via AG (a) e via método das RNA (b). Em ambos os casos o valor estimado é representado com quadrados vazados e os reais com estrelas preenchidas.

para aplicação em problemas nos quais as metodologias isoladamente já foram aplicadas, como o problema de reconstrução de perfil de temperatura em fenômenos de propagação do calor [Campos Velho 2008] e o problema de identificação de danos estruturais [Santos et al. 2011, Santos 2011, Chiwiacowsky 2005], cujo fenômeno físico central é exatamente as oscilações harmônicas.

Referências

[Bittencourt 2006] Bittencourt, G. (2006). *Inteligência artificial*. Editora da UFSC, Florianópolis.

[Boyce and DiPrima 2002] Boyce, W. E. and DiPrima, R. C. (2002). *Equações diferenciais elementares e problemas de valores de contorno*. LTC, Rio De Janeiro, 7ª edition.

[Campos Velho 2008] Campos Velho, H. F. (2008). *Problemas Inversos em Pesquisa Espacial*. Belém, PA, Brasil. Mini-curso.

[Chiwiacowsky 2005] Chiwiacowsky, L. D. (2005). *Método Variacional e Algoritmo Genético em Identificação de Danos Estruturais*. Tese (doutorado em computação aplicada), Instituto Nacional de Pesquisas Espaciais (INPE), São José dos Campos.

[Dall Cortivo et al. 2012a] Dall Cortivo, F., Chalhoub, E. S., and Campos Velho, H. F. (2012a). A committee of MLP with adaptive slope parameter trained by the quasi-Newton method to solve problems in hydrologic optics. In *International Joint Conference on Neural Networks*, pages 1–8, Brisbane, QLD, Australia.

[Dall Cortivo et al. 2012b] Dall Cortivo, F., Chalhoub, E. S., and Campos Velho, H. F. (2012b). Comparison of two learning strategies for a supervised neural network. In *Proceedings of the First International Symposium on Uncertainty Quantification and Stochastic Modelling*, pages 366–380, São Sebastião, SP, Brazil.

[Haykin 2001] Haykin, S. (2001). *Redes neurais*. Bookman, Porto Alegre.

[Holland 1975] Holland, J. H. (1975). *Adaptation in natural and artificial systems*. MIT Press.

[McCulloch and Pitts 1943] McCulloch, W. and Pitts, W. H. (1943). A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 5:115–133.

[NAG 1995] NAG (1995). *Fortran Library Manual*. The Numerical Algorithms Group (NAG), Oxford, UK.

[Neto and Neto 2005] Neto, A. J. S. and Neto, F. D. M. (2005). *Problemas Inversos: Conceitos Fundamentais e Aplicações*. EdUERJ, Rio de Janeiro.

[Santos 2011] Santos, L. B. L. (2011). *Abordagem hierárquica ao método híbrido de estimação de danos em estruturas aeroespacias*. Dissertação (mestrado em computação aplicada), Instituto Nacional de Pesquisas Espaciais (INPE), São José dos Campos.

[Santos et al. 2011] Santos, L. B. L., Chiwiacowsky, L. D., and Campos Velho, H. F. (2011). Análise de robustez do método híbrido de estimação de dano estrutural. *TEMA–Tendências em Matemática Aplicada e Computacional*, 12(3):245–252.

[Santos and Miranda 2009] Santos, L. B. L. and Miranda, J. G. V. (2009). Poga :: Parameter's optimization by genetic algorithm, http://sourceforge.net/projects/thepoga/files/.

[Shiguemori et al. 2005] Shiguemori, E. H., Chiwiacowsky, L. D., Campos Velho, H. F., and SILVA, J. D. S. (2005). An inverse vibration problem solved by an artificial neural network. *TEMA– Tendências em Matemática Aplicada e Computacional*, 6(1):163–175.