

Universidade Federal de Juiz de Fora
Programa de Mestrado em Modelagem Computacional

**MIDaS - Um sistema computacional baseado em
aplicações web para identificação modal de sistemas
dinâmicos deformáveis**

Por

Alexandre Abrahão Cury

JUIZ DE FORA, MG - BRASIL

MARÇO DE 2008

MIDAS - UM SISTEMA COMPUTACIONAL BASEADO EM
APLICAÇÕES WEB PARA IDENTIFICAÇÃO MODAL DE
SISTEMAS DINÂMICOS DEFORMÁVEIS

Alexandre Abrahão Cury

DISSERTAÇÃO SUBMETIDA AO CORPO DOCENTE DO UNIVERSIDADE
FEDERAL DE JUIZ DE FORA COMO PARTE DOS REQUISITOS NECES-
SÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS EM
MODELAGEM COMPUTACIONAL

Aprovada por:

Prof. Flávio de Souza Barbosa, D.Sc
(Orientador)

Prof. Carlos Cristiano Hasenclever Borges, D.Sc.

Prof. Ciro de Barros Barbosa, D.Sc.

Prof. Ronaldo Carvalho Battista, Ph.D.

Prof. Roque Luiz da Silva Pitangueira, D.Sc.

JUIZ DE FORA, MG - BRASIL
MARÇO DE 2008

Cury, Alexandre Abrahão

MIDaS - Um sistema computacional baseado em aplicações web para identificação modal de sistemas dinâmicos deformáveis / Alexandre Abrahão Cury; Orientador: Flávio de Souza Barbosa – 2008.

144 f.: il.

Dissertação (Mestrado em Modelagem Computacional) – Faculdade de Engenharia Civil, Universidade Federal de Juiz de Fora.

1. Dinâmica Estrutural. 2. Identificação Modal. 3. Aplicação Web. I. Barbosa, Flávio de Souza. II. Título.

CDU 642.042

**Nunca ande pelo caminho traçado,
pois ele conduz somente até onde os
outros foram.**

(Alexander Graham Bell.)

Aos meus pais.

Agradecimentos

A Deus, por iluminar meu caminho e estar sempre presente em todos momentos da minha vida.

Aos meus pais, pelo amor incondicional, carinho, dedicação e pela formação do meu caráter.

Ao professor, orientador e amigo Flávio de Souza Barbosa, pelo apoio e incentivo durante a iniciação científica e o mestrado. Sou muito grato por todos os ensinamentos passados.

Ao professor e amigo Carlos Cristiano Hasenclever Borges, pela orientação e companheirismo fundamentais demonstrados durante os três anos de iniciação científica e que serviram de alicerce para a realização deste trabalho.

Aos amigos de mestrado Brenno, Igor, Rafael, Samuel e, em especial, Carolina e Heder, pela valiosa ajuda nas lições de Java.

Ao meu primo David por toda a atenção, apoio e a quem também dedico este trabalho.

À Universidade Federal de Juiz de Fora pelo apoio financeiro recebido durante o período de abril de 2007 à abril de 2008.

Resumo da Dissertação apresentada ao MMC/UFJF como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

**MIDAS - UM SISTEMA COMPUTACIONAL BASEADO EM
APLICAÇÕES WEB PARA IDENTIFICAÇÃO MODAL DE
SISTEMAS DINÂMICOS DEFORMÁVEIS**

Alexandre Abrahão Cury

Março , 2008

Orientador: Flávio de Souza Barbosa, D.Sc

Com o aperfeiçoamento dos ensaios vibratórios e o aumento da capacidade de processamento dos computadores, nota-se o crescente desenvolvimento dos algoritmos de avaliação dinâmica, cujo principal objetivo é a determinação das características modais de sistemas dinâmicos deformáveis, em um processo denominado *identificação modal*.

A partir da identificação e análise das características modais, isto é, frequências naturais, taxas de amortecimento e modos de vibração, torna-se possível inferir sobre certas condições físicas de uma estrutura, como por exemplo o grau de integridade, o estado de danos e a margem de segurança.

Entretanto, o processo de identificação de sistemas dinâmicos deformáveis às vezes requer um custo computacional elevado. Mas, com a expansão da internet aliado à facilidade de seu acesso com velocidades cada vez maiores, o conceito de aplicação web surge como uma solução a ser empregada.

No presente trabalho apresenta-se uma aplicação web implementada em linguagem Java com o objetivo de realizar a identificação modal de sistemas deformáveis a partir do envio, através da internet, de dados obtidos de medições dinâmicas.

Para a implementação desta aplicação são utilizadas tecnologias e ferramentas de desenvolvimento web, como o JSP (Java Server Pages) e o JSF (Java Server

Faces), além de bibliotecas externas empregadas nos algoritmos de identificação.

No que tange ao processo de tratamento dos dados vibracionais dos sistemas deformáveis, são implementados dois métodos para a identificação modal no domínio do tempo: o Método do Decremento Aleatório e o Método de Ibrahim.

Finalmente, com o objetivo de avaliar a aplicação desenvolvida e os métodos de identificação implementados, três estudos de casos de estruturas ensaiadas são explorados.

Abstract of Dissertation presented to MMC/UFJF as a partial fulfillment of the requirements for the degree of Master of Sciences (M.Sc.)

**MIDAS - A COMPUTATIONAL SYSTEM BASED ON WEB
APPLICATIONS FOR THE MODAL IDENTIFICATION OF
DEFORMABLE STRUCTURAL SYSTEMS**

Alexandre Abrahão Cury

March, 2008

Advisor: Flávio de Souza Barbosa, D.Sc

The increasing development of algorithms for the determination of the modal characteristics of dynamic systems stems from a continuous improvement of vibration tests and computational processing capabilities.

By analyzing these modal parameters, i.e., natural vibration mode shapes and associated frequencies and damping ratios, it is possible to infer that some physical conditions of a structure, such as for example, the degree of integrity, the state of damage and safety margin.

However, the identification process of deformable dynamic systems demands sometimes high computational costs. Nevertheless, with the increasing expansion of higher transfer rate broad-band internet the concept of web application becomes a prevailing alternative.

This work presents a web application - MIDaS (Modal Identification of Deformable Structural Systems) - implemented in Java language capable to perform the modal identification of deformable systems via internet by using only data obtained through experimental dynamic measurements.

Available technologies and web development tools, such as JSP(JavaServer Pages) and JSF(JavaServer Faces) and external libraries for the identification algorithms were used to implement MIDaS.

The dynamic treatment of the experimental vibration data is accomplished by means of two time-domain modal identification methods: the Random Decrement Method and the Ibrahim Time Domain Method.

Finally, in order to evaluate the developed web application and the implemented identification methods, three cases of dynamically tested structures are explored.

Sumário

brazil

1	Introdução	1
1.1	Escopo do Texto	5
2	Introdução à Modelagem da Dinâmica de Sistemas Deformáveis	7
2.1	Sistemas com 1 Grau de Liberdade	7
2.1.1	Componentes de um Sistema Dinâmico Básico	7
2.1.2	Análise de Vibrações Livres	9
2.1.3	Análise de Vibrações Forçadas	11
2.2	Sistemas com vários Graus de Liberdade	12
2.2.1	Análise de Vibrações Livres	14
3	Identificação Modal de Sistemas Deformáveis	17
3.1	Introdução	17
3.2	Método do Decremento Aleatório	18
3.2.1	Princípios do Método do Decremento Aleatório	18
3.2.2	Definição das funções de decremento aleatório	20
3.2.3	Condição de desencadeamento	23
3.3	Método de Ibrahim	29
3.3.1	Identificação dos Parâmetros Modais	29
3.3.2	Obtenção dos parâmetros modais	32

4	Introdução às Aplicações Web em Java	38
4.1	A linguagem de programação Java	38
4.1.1	Alguns Conceitos sobre Java	41
4.1.2	Ambientes de Desenvolvimento Integrado - IDEs	43
4.1.3	Bibliotecas	43
4.2	Aplicações web	44
4.2.1	A Tecnologia JSP (JavaServer Pages)	45
4.2.2	A Tecnologia JSF (JavaServer Faces)	46
4.3	Servidores de Aplicação	47
4.3.1	Servidor Glassfish	47
4.3.2	Arquivo WAR	48
4.4	Exemplo de aplicação web	49
5	A Aplicação Web MIDaS	56
5.1	Introdução	56
5.2	Bibliotecas	64
5.3	Algoritmos Implementados	65
5.3.1	MIDaS-Ident	65
5.3.2	MIDaS-Trunc	66
5.3.3	MIDaS-Filter	67
5.3.4	MIDaS-Assembly	69
5.3.5	MIDaS-Visual	70
5.4	Arquivo de Parâmetros	71
5.4.1	<i>step1.txt</i>	71
5.4.2	<i>step2.txt</i>	75
5.5	Arquivo de dados	77
6	Exemplos de Aplicação	79
6.1	Viga bi-apoiada	80
6.1.1	Resultados	83

6.1.2	Tempos de análise	92
6.2	Ensaio dinâmico sobre ponte férrea	94
6.2.1	Resultados	94
6.2.2	Tempos de análise	96
6.3	Ensaio dinâmico sobre a ponte Z24	97
6.3.1	Resultados	98
6.3.2	Tempos de análise	101
7	Considerações Finais	103
	Referências Bibliográficas	106

Lista de Figuras

Figura

brazil		
1.1	Aplicações web do Google.	3
2.1	Sistema 1GL: a) componentes básicos. b) diagrama de corpo livre. .	8
2.2	Resposta temporal em vibrações livres com amortecimento sub- crítico para um sistema de 1GL. Extraído da referência [11].	11
2.3	Discretização de uma estrutura de viga arbitrária. Extraído da re- ferência [11].	12
3.1	Princípio do Método do Decremento Aleatório. Extraído de [27]. . .	19
3.2	Exemplo de estimativa das funções de Decremento Aleatório.	22
3.3	a)Pontos de desencadeamento; b)Funções de decremento utilizando diferentes números de pontos de desencadeamento. Extraído da referência [27].	27
3.4	Representação de um número complexo.	33
3.5	Exemplo de geração de pseudo-medidas do Método de Ibrahim. . . .	36
4.1	Compilador e interpretador Java.	39
4.2	Exemplo de aplicação web - <i>MSN Spaces</i>	44
4.3	Página inicial do servidor Glassfish.	48
4.4	Primeiro passo para a criação de uma aplicação web utilizando Net- Beans - Escolha do tipo de projeto.	50

4.5	Segundo passo para a criação de uma aplicação web utilizando NetBeans - Definição do nome e do servidor de aplicação utilizados no projeto.	50
4.6	Terceiro passo para a criação de uma aplicação web utilizando NetBeans - Escolha do <i>framework</i> a ser utilizado.	51
4.7	Interface do IDE NetBeans com recursos JSF.	51
4.8	Tela de edição do NetBeans.	53
4.9	Tela de exibição do navegador.	53
4.10	Primeira simulação do exemplo de aplicação web.	54
4.11	Segunda simulação do exemplo de aplicação web - Mensagem de erro.	54
4.12	Código dos métodos dos botões <i>Calcular</i> e <i>Apagar</i> do exemplo de aplicação web.	55
5.1	MIDaS - Página inicial.	58
5.2	MIDaS - Sobre a aplicação.	59
5.3	MIDaS - Página de cadastro de usuários.	59
5.4	Exemplo de um formulário de cadastro gerado pelo MIDaS.	60
5.5	MIDaS - Página de <i>login</i>	60
5.6	MIDaS - Análise MIDaS-Ident.	61
5.7	MIDaS - Resultados MIDaS-Ident / Análises MIDaS-Trunc, Filter, Assembly e Visual.	61
5.8	MIDaS - Página de resultados finais.	62
5.9	Fluxograma da aplicação web MIDaS.	63
5.10	Exemplo de arquivos de <i>log</i> da aplicação web MIDaS.	68
5.11	Exemplo de intervalo de confiança para uma amostra de frequências.	69
6.1	Viga de aço instrumentada com 2 acelerômetros.	80
6.2	Esquema de instrumentação da viga de aço com 15 posições.	81
6.3	Acelerações medidas nos pontos 3 e 1 da viga de aço ensaiada.	81
6.4	Histograma de frequências após a análise MIDaS-Ident.	84

6.5	Frequências naturais obtidas ao longo do sinal após a análise MIDaS-Trunc.	85
6.6	Taxas de amortecimento obtidas ao longo do sinal após a análise MIDaS-Trunc.	85
6.7	Histograma de frequências após a análise MIDaS-Trunc.	86
6.8	Histograma de amortecimentos após a análise MIDaS-Trunc.	86
6.9	Frequências naturais obtidas ao longo do sinal após a análise MIDaS-Filter.	87
6.10	Taxas de amortecimento obtidas ao longo do sinal após a análise MIDaS-Filter.	87
6.11	Histograma de frequências após a análise MIDaS-Filter.	88
6.12	Histograma de amortecimentos após a análise MIDaS-Filter.	88
6.13	Frequências naturais obtidas ao longo do sinal após a análise MIDaS-Assembly.	89
6.14	Taxas de amortecimento obtidas ao longo do sinal após a análise MIDaS-Assembly.	89
6.15	Histograma de frequências após a análise MIDaS-Assembly.	90
6.16	Histograma de amortecimentos após a análise MIDaS-Assembly.	90
6.17	Modos de vibração da viga bi-apoiada. Comparativo entre os resultados teórico e da aplicação MIDaS.	92
6.18	Esquema de instrumentação da ponte férrea - TGV.	95
6.19	Sinais adquiridos para nove pontos instrumentados na primeira série de medidas sobre a ponte férrea.	96
6.20	Características geométricas da ponte Z24.	97
6.21	Esquema de instrumentação da ponte Z24.	99
6.22	Sinais adquiridos para 33 pontos de medida instrumentados do primeiro conjunto de ensaios sobre a ponte Z24.	100
6.23	Histograma de frequências identificadas no ensaio da ponte Z24.	100

6.24 Primeiro modo de vibração identificado pelo MIDaS para a ponte Z24.	101
6.25 Primeiro modo de vibração da ponte Z24 identificado por Barbosa <i>et al</i> [3] e DeRoeck <i>et al</i> [41].	102

Lista de Tabelas

Tabela

brazil		
3.1	Condições de desencadeamento freqüentemente utilizadas.	26
4.1	Comparativo de tempo entre as linguagens C, Fortran e Java. . . .	40
6.1	Valores médios e desvios-padrões dos parâmetros modais identifica- dos pela aplicação MIDaS.	91
6.2	Comparativo entre freqüências teóricas e experimentais - Viga bi- apoiada.	91
6.3	Comparativo de tempo entre as aplicações - Viga bi-apoiada. Os tempos apresentados estão em segundos. (*) - Transmissão de da- dos. (**) - Tempo desconsiderando a transmissão de dados. (***) - Tempo considerando a transmissão de dados.	93
6.4	Valores médios e desvios-padrões dos parâmetros modais identifica- dos pela aplicação MIDaS - Ponte férrea.	95
6.5	Comparativo entre freqüências obtidas pelo MIDaS e pela referência [27] - Ponte férrea.	96
6.6	Comparativo de tempo entre as aplicações - Ponte férrea. Os tempos apresentados estão em segundos. (*) - Transmissão de dados. (**) - Tempo desconsiderando a transmissão de dados. (***) - Tempo considerando a transmissão de dados.	97
6.7	Valores médios e desvios-padrões dos parâmetros modais identifica- dos pela aplicação MIDaS - Ponte Z24.	99

6.8	Comparativo entre frequências obtidas pelo MIDaS e pelas referências [3] e [41] - Ponte Z24.	101
6.9	Comparativo de tempo entre as aplicações - Ponte Z24. Os tempos apresentados estão em segundos. (*) - Transmissão de dados. (**) - Tempo desconsiderando a transmissão de dados. (***) - Tempo considerando a transmissão de dados.	102

Glossário

Bytecodes : representação intermediária da linguagem Java compilada.

Download : cópia, para o computador do usuário, de arquivos através de um navegador de internet.

Eclipse : IDE para Java/C/C++ desenvolvida pela Sun

Framework : estrutura de suporte definida para que um outro projeto de *software* possa ser organizado e desenvolvido.

Glassfish : servidor de aplicação web.

HTML (HyperText Markup Language): linguagem de marcação para hipertexto com a qual as páginas web são feitas.

HTTP (Hypertext Transfer Protocol): protocolo para envio e recebimento de dados pela Internet.

IDE (Integrated Development Environments): ambiente de desenvolvimento integrado para implementação de códigos de uma linguagem.

J2EE : plataforma Java voltada para redes, internet, intranets e afins.

J2ME : plataforma de desenvolvimento para dispositivos móveis ou portáteis.

J2SE : plataforma de desenvolvimento voltada para computadores pessoais e servidores.

JDK (Java Development Kit): conjunto de programas para desenvolvimento de códigos em Java.

JRE (Java Runtime Environment): conjunto de bibliotecas necessárias para executar códigos Java.

JSF (JavaServer Faces): tecnologia utilizada para criação de aplicações web.

JSP (JavaServer Pages): tecnologia utilizada para criação de aplicações web.

JVM (Java Virtual Machine): máquina virtual para compilação dos *bytecodes*.

Link : texto ou imagem que permite o acesso imediato à outra parte de um sítio, bastando ser acionado pelo ponteiro do mouse.

MVC (Model-View-Controller): padrão de arquitetura de *software* utilizada para separar os dados (*Model*) da interface (*View*), de forma que as alterações feitas na interface não afetem a manipulação dos dados.

NetBeans : IDE para Java desenvolvida pela Sun.

Plug-in : programa ou recurso que adiciona uma funcionalidade a outro programa.

Scripts : códigos na linguagem de computador interpretada.

Servlets : classes Java escritas para serem executadas em um servidor.

Software : programa ou aplicativo de computador.

Sun : empresa desenvolvedora de tecnologias Java.

TCP/IP (Transmission Control Protocol/Internet Protocol): protocolo de rede usado na Internet;

Upload : cópia, a partir do computador do usuário, de arquivos através de um navegador de internet.

XML (Extensible Markup Language): linguagem de marcação extensível, utilizada para armazenamento de dados.

Capítulo 1

Introdução

A realização de ensaios de vibração em sistemas deformáveis é de grande importância prática na área das engenharias. Vários trabalhos foram publicados relatando a aplicação de diversos tipos de instrumentação e sistemas de aquisição de dados utilizados em ensaios e monitoração dinâmica realizados em estruturas de grande vulto, como a Ponte Rio-Niterói [1],[2] no Brasil, a Ponte Z24 [3] na Suíça e o viaduto de Millau [4] na França.

Algumas obras de arte são sensoreadas 24 horas por dia, 7 dias por semana, durante um certo intervalo de tempo, com o objetivo de fornecer dados extraídos de ensaios de vibração [5] que possam permitir a identificação de problemas estruturais, como presença de fissuras, vibração excessiva, etc, além de tornar possível uma avaliação mais aprofundada acerca de sua confiabilidade e vulnerabilidade.

Diante deste cenário e com o aumento da capacidade de processamento dos computadores, observa-se um crescente desenvolvimento de algoritmos de avaliação dinâmica, cujo principal objetivo é a determinação das características modais da estrutura, isto é, frequências naturais, taxas de amortecimento e modos de vibração do sistema, em um processo denominado *identificação modal*.

A partir da análise das características modais, é possível inferir sobre certas condições de integridade física de uma estrutura como, por exemplo, o estado de danos [6] e a margem de segurança. Entretanto, este processo geralmente requer um custo computacional elevado e, além disso, os programas disponíveis no mer-

cado para o tratamento e análise dos dados obtidos de ensaios de vibração são proprietários.

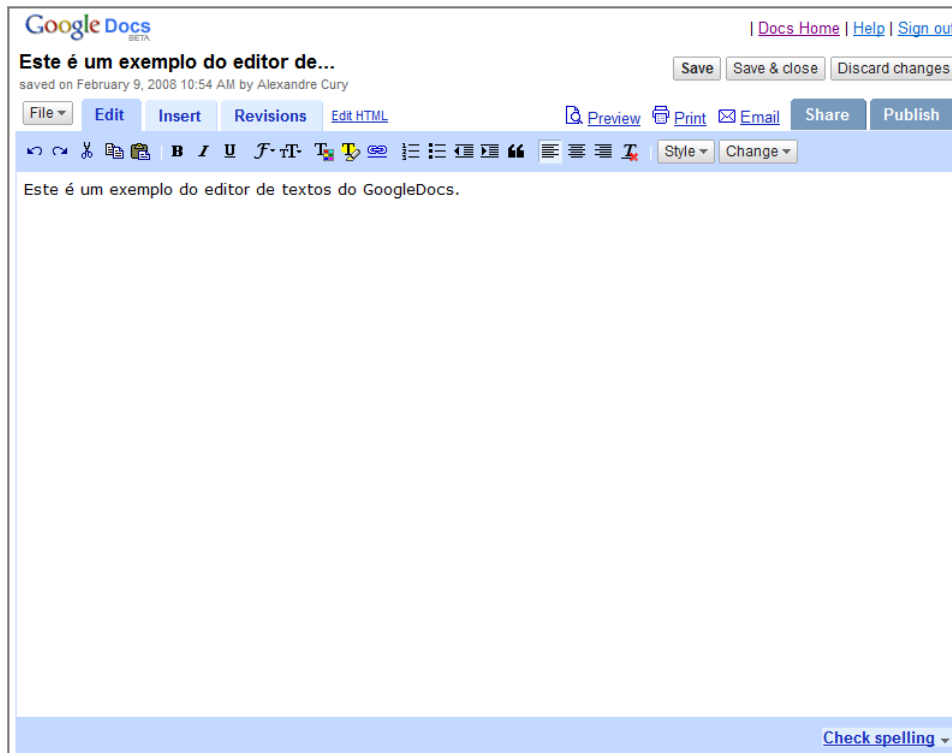
Dessa maneira, com a expansão e popularização da internet de alta velocidade a preços mais competitivos, as aplicações web surgem como uma solução a ser empregada. Estas aplicações são, de forma geral, sistemas de informática projetados para utilização através de um navegador - na internet ou em redes privadas - e consistem, normalmente, em um conjunto de programas que podem ser executados em um servidor de aplicação.

O desenvolvimento das aplicações web está relacionado, dentre outros fatores, à necessidade de simplificar a atualização e a manutenção destes sistemas e por permitir a utilização de aplicativos que não precisam ser instalados no computador do usuário. Atualmente, diversas empresas fornecem seus serviços diretamente na internet, como o BabelFish pertencente ao grupo Altavista [7] que oferece um tradutor de idiomas e até outros mais elaborados como o Google [8] que oferece desde editores de texto (Figura 1.1(a)) até ferramentas para apresentação em *slides* (Figura 1.1(b)).

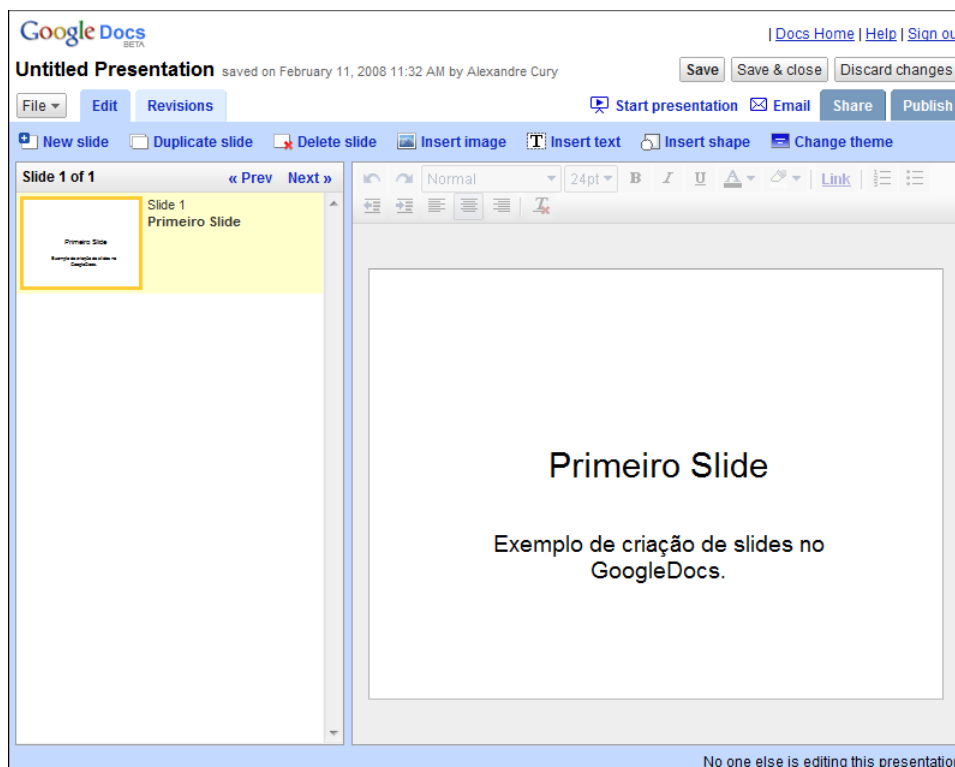
Aliado a isso, observa-se atualmente o desenvolvimento da tecnologia móvel, notadamente o padrão 3G (com alta velocidade de transmissão de dados) para aparelhos portáteis como *PDA*s¹, *palmtops*, celulares, etc. No dia-a-dia, isto representa a possibilidade de se conectar à internet diretamente de um aparelho móvel, permitindo ao usuário ler seu correio eletrônico, navegar em sítios virtuais e transferir grandes quantidades de informação em pouquíssimo tempo.

Trazendo este conceito para a área das engenharias, mais especificamente na dinâmica de sistemas deformáveis, esta tecnologia permite que, mesmo distante de um computador, o engenheiro pode receber os resultados referentes aos ensaios vibratórios que estiverem sendo realizados, em tempo real, considerando-se que a transmissão dos dados pode ser feita a partir do equipamento de aquisição dos sinais de forma rápida e eficiente.

¹ Personal Digital Assistants (Assistente Pessoal Digital): computador de dimensões reduzidas, dotado de grande capacidade computacional



(a) Exemplo do editor de textos web do Google.



(b) Exemplo do editor de *slides* web do Google.

Figura 1.1: Aplicações web do Google.

Existe na comunidade científica o projeto INSANE (Interactive Structural Analysis Environment) desenvolvido sob o conceito de *software* livre cujo um dos

objetivos é criar um sistema computacional para análise de modelos discretos de elementos finitos [9].

Este projeto foi inteiramente implementado em linguagem de programação Java e, atualmente, vem sendo desenvolvido no Departamento de Engenharia de Estruturas da Universidade Federal de Minas Gerais, sob a orientação do professor Roque Luiz da Silva Pitangueira. O INSANE é um aplicativo capaz de fornecer ao usuário uma interface gráfica para criação e edição de modelos de sistemas estruturais, além de ser uma ferramenta para solução de problemas modelados utilizando o Método dos Elementos Finitos.

Em meados de 2007, foi implementada por Câmara [10] um serviço web pertencente a este projeto e que tem por objetivo principal disponibilizar o INSANE através da internet, sem que haja a necessidade de o usuário instalar o aplicativo em seu computador pessoal.

Entretanto, na área de identificação dinâmica, o que se observa é a inexistência de aplicações deste tipo (busca realizada em Março 2008), isto é, que permitam ao usuário resolver problemas de identificação modal sem que haja necessidade de comprar ou instalar programas em seu computador. Diante deste cenário, o objetivo do presente trabalho é desenvolver uma aplicação web para identificação modal de sistemas dinâmicos deformáveis disponível de forma gratuita para qualquer usuário que tenha acesso à internet.

Esta aplicação, denominada MIDaS (Modal Identification of Deformable Systems - www.midas.ufjf.br), consiste em um conjunto de rotinas capazes de realizar a identificação modal de estruturas partindo-se de ensaios experimentais simples ou de maior grau de complexidade em que um grande número de sensores é usado em diversos conjuntos de séries de medidas.

Por se tratar de uma aplicação web, significa que o usuário não precisa se preocupar em possuir um computador dotado de grande capacidade de processamento, uma vez que todos os cálculos serão realizados em um servidor.

O MIDaS compreende algoritmos de identificação implementadas em lingua-

gem de programação Java. As análises realizadas são divididas em dois passos visando proporcionar uma maior aplicabilidade ao sistema. No primeiro passo da análise, o usuário, fornecendo apenas alguns dados de configuração e os arquivos de resultados de ensaios, consegue obter os valores das frequências naturais de vibração identificadas. No segundo passo, uma série de recursos tais como filtragem das frequências de interesse, definição de intervalo de confiança para os parâmetros identificados e a visualização dos modos próprios de vibração, são disponibilizados ao usuário. Entretanto, o segundo passo demanda dados complementares de configuração.

Ao final das análises, são disponibilizados ao usuário arquivos de resultados, sendo eles histogramas com as frequências naturais identificadas, gráficos com a variação das taxas de amortecimento, modos próprios de vibração, além de arquivos texto contendo as características modais e informações a respeito do tempo decorrido em cada análise e valores estatísticos dos parâmetros modais identificados.

1.1 Escopo do Texto

Este trabalho está organizado em 7 capítulos, incluindo o Capítulo 1 que dá uma visão geral sobre o trabalho desenvolvido.

No Capítulo 2 são expostos conceitos gerais a respeito da modelagem dinâmica de sistemas deformáveis, analisando-se as equações diferenciais ordinárias que regem o comportamento dinâmico desses sistemas em modelos de um ou mais graus de liberdade.

Em seguida, no Capítulo 3, são discutidos os dois métodos de identificação modal de sistemas deformáveis no domínio do tempo implementados na aplicação MIDaS: o Método do Decremento Aleatório e o Método de Ibrahim.

O Capítulo 4 tem o objetivo de descrever de forma clara e concisa algumas das tecnologias utilizadas para o desenvolvimento da aplicação web MIDaS. Para isso, são apresentados conceitos acerca da linguagem de programação utilizada -

Java - além de uma visão geral sobre servidores de aplicação, das tecnologias JSP (JavaServer Pages) e JSF (JavaServer Faces). Ao final deste capítulo, um exemplo simples de aplicação web é criado com o intuito de ilustrar as idéias discutidas.

No Capítulo 5 descreve-se a aplicação web desenvolvida para a identificação modal de sistemas dinâmicos deformáveis - o MIDaS. Neste capítulo são discutidos os algoritmos implementados, as bibliotecas utilizadas, além dos detalhes necessários para se utilizar a aplicação.

A potencialidade da aplicação MIDaS é apresentada no Capítulo 6 a partir de um breve estudo de três exemplos de tratamento de dados extraídos de casos de estruturas ensaiadas.

Conclusões, principais contribuições oferecidas por este trabalho, bem como algumas sugestões para futuros trabalhos de pesquisa envolvendo o tema discutido, são apresentadas no Capítulo 7.

Capítulo 2

Introdução à Modelagem da Dinâmica de Sistemas Deformáveis

O autor entende que uma boa compreensão dos métodos de identificação modal de sistemas deformáveis implementados neste trabalho passa também por uma boa compreensão dos conceitos acerca da modelagem computacional desses sistemas, razão pela qual este capítulo foi incorporado a esta dissertação.

Inicia-se este capítulo com a análise do problema clássico de 1 grau de liberdade (1GL) modelado por um sistema massa-mola-amortecedor. A seguir, esta análise é estendida para sistemas com vários graus de liberdade, que visam modelar estruturas com maior nível de complexidade.

Este capítulo resume os conceitos apresentados em alguns textos clássicos da literatura relativos à dinâmica de sistemas deformáveis. Para uma análise mais aprofundada, recomenda-se a leitura das referências [11], [12] e [13].

2.1 Sistemas com 1 Grau de Liberdade

Nesta seção serão apresentados breves conceitos a respeito de sistemas com 1 grau de liberdade.

2.1.1 Componentes de um Sistema Dinâmico Básico

De uma forma geral, as propriedades físicas essenciais de um sistema mecânico sujeito a uma fonte externa de excitação ou carregamento dinâmico são:

massa, propriedades elásticas e amortecimento ou perda de energia mecânica. No modelo mais simples de um sistema com 1 grau de liberdade, assume-se que cada uma destas propriedades esteja concentrada em um único elemento físico. As Figuras 2.1(a) e 2.1(b) mostram um esquema deste sistema.

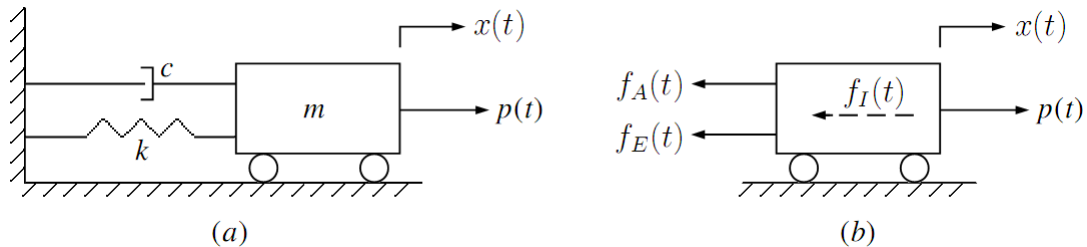


Figura 2.1: Sistema 1GL: a) componentes básicos. b) diagrama de corpo livre.

Considera-se que a massa m do sistema esteja contida em um bloco rígido que pode se mover somente em translação simples. Assim, a coordenada de deslocamento $x(t)$ define completamente sua posição. A resistência elástica ao deslocamento é devida a uma mola de rigidez k e peso desprezível, enquanto o mecanismo de perda de energia (ou amortecimento) é representado pelo amortecedor c . O carregamento dinâmico externo, variante no tempo, é representado por $p(t)$.

A equação de movimento de um sistema como o mostrado na Figura 2.1(a) é obtida diretamente a partir da expressão de equilíbrio de todas as forças atuantes no bloco:

$$f_I(t) + f_A(t) + f_E(t) = p(t), \quad (2.1)$$

onde $f_I(t)$ representa a força inercial, $f_A(t)$ é a força causada pelo amortecimento, $f_E(t)$ é a força elástica e $p(t)$, o carregamento.

De acordo com o princípio de d'Alembert, a força inercial é obtida pelo produto da massa pela aceleração:

$$f_I(t) = m\ddot{x}(t), \quad (2.2)$$

onde $\ddot{x}(t)$ representa a derivada segunda temporal de $x(t)$.

Assumindo um mecanismo de amortecimento viscoso, a força de amorteci-

mento pode ser representada como o produto da constante de amortecimento c pela velocidade:

$$f_A(t) = c\dot{x}(t), \quad (2.3)$$

onde $\dot{x}(t)$ representa a derivada primeira temporal de $x(t)$.

E, finalmente, considerando a força elástica como o produto da rigidez da mola pelo deslocamento, vem:

$$f_E(t) = kx(t). \quad (2.4)$$

Substituindo as Eq.(2.1) a Eq.(2.3) na Eq.(??), obtém-se a equação de movimento diferencial ordinária de segunda ordem para um sistema com um único grau de liberdade:

$$m\ddot{x}(t) + c\dot{x}(t) + kx(t) = p(t). \quad (2.5)$$

2.1.2 Análise de Vibrações Livres

Um sistema dinâmico se encontra em vibrações livres quando o carregamento aplicado $p(t)$ é nulo. Assim, tem-se da Eq.(2.4):

$$m\ddot{x}(t) + c\dot{x}(t) + kx(t) = 0. \quad (2.6)$$

A solução da Eq.(2.5) pode ser expressa como:

$$x(t) = Ce^{st}, \quad (2.7)$$

onde C é uma constante de integração arbitrária dependente das condições iniciais. Substituindo as derivadas primeira e segunda da Eq.(2.6) na Eq.(2.5) e fazendo $\omega = \sqrt{\frac{k}{m}}$, vem:

$$s^2 + \frac{c}{m}s + \omega^2 = 0. \quad (2.8)$$

A solução da Eq.(2.7) é dada por:

$$s_{1,2} = -\frac{c}{2m} \pm \sqrt{\left(\frac{c}{2m}\right)^2 - \omega^2}. \quad (2.9)$$

Para a maioria das estruturas reais, o radicando da Eq.(2.8) é negativo, representando um caso típico de sistemas dinâmicos com amortecimento sub-crítico. Desta forma, a solução da equação Eq.(2.5) com as condições iniciais de deslocamento $x(0)$ e velocidade $\dot{x}(0)$, se escreve como:

$$x(t) = \left[x(0) \cos \omega_D t + \left(\frac{\dot{x}(0) + x(0)\xi\omega}{\omega_D} \right) \sin \omega_D t \right] e^{-\xi\omega t}, \quad (2.10)$$

onde $\xi = \frac{c}{2m\omega}$ representa a taxa de amortecimento e ω_D é a frequência de vibração livre do sistema amortecido e é definida como $\omega_D = \omega\sqrt{1 - \xi^2}$.

A grandeza ω definida anteriormente como $\omega = \sqrt{\frac{k}{m}}$ é denominada frequência angular não-amortecida.

A relação entre a frequência angular não-amortecida ω e a frequência de movimento f é dada por:

$$f = \frac{\omega}{2\pi}, \quad (2.11)$$

então

$$T = \frac{1}{f} = \frac{2\pi}{\omega} \quad (2.12)$$

é o tempo necessário para se completar um ciclo de oscilação e é denominado período do movimento. Usualmente, para sistemas estruturais ou mecânicos, o período T é medido em segundos (s) e a frequência, em ciclos por segundo, comumente denominado Hertz (Hz).

A Eq.(2.9) pode ser escrita como:

$$x(t) = \hat{x} \sin(\omega_D t + \theta) e^{-\xi\omega t}, \quad (2.13)$$

onde a amplitude \hat{x} é dada por:

$$\hat{x} = \sqrt{[x(0)]^2 + \left[\frac{\dot{x}(0)}{\omega}\right]^2}, \quad (2.14)$$

e o ângulo de fase θ é escrito como:

$$\theta = \tan^{-1} \left[\frac{-\dot{x}(0)}{\omega x(0)} \right]. \quad (2.15)$$

A Figura 2.2 mostra a resposta temporal de um sistema com 1GL em vibrações livres com amortecimento sub-crítico.

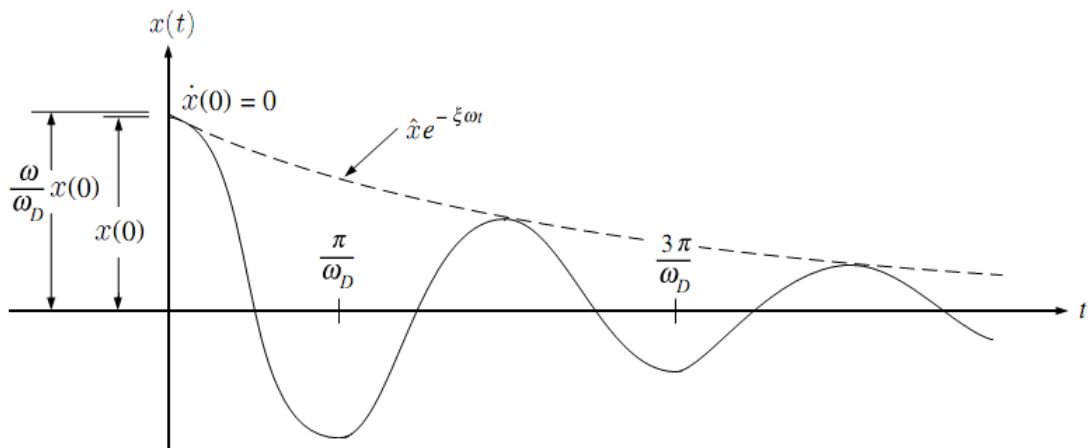


Figura 2.2: Resposta temporal em vibrações livres com amortecimento sub-crítico para um sistema de 1GL. Extraído da referência [11].

2.1.3 Análise de Vibrações Forçadas

A resposta temporal de um sistema dinâmico sob vibrações forçadas pode ser escrita como a soma de três parcelas:

- Resposta devida ao deslocamento inicial x_0 , definida como $x_{x_0}(t)$;
- Resposta devida a velocidade inicial \dot{x}_0 , definida como $x_{\dot{x}_0}(t)$;
- Resposta devida ao carregamento externo aplicado $p(t)$, definida como $x_{p(t)}(t)$.

Desta forma, tem-se que:

$$x(t) = x_{x_0}(t) + x_{\dot{x}_0}(t) + x_{p(t)}(t). \quad (2.16)$$

Caso a parcela referente ao carregamento $x_{p(t)}(t)$ seja igual a zero, recai-se nas equações de vibrações livres apresentadas na subseção anterior, sendo:

$$x_{x_0}(t) = x(0) \left(\cos \omega_D t + \frac{\xi \omega}{\omega_D} \sin \omega_D t \right) e^{-\xi \omega t} \quad e \quad (2.17)$$

$$x_{\dot{x}_0}(t) = \dot{x}(0) \left(\frac{\sin \omega_D t}{\omega_D} \right) e^{-\xi \omega t}. \quad (2.18)$$

2.2 Sistemas com vários Graus de Liberdade

As equações descritas na seção 2.1 podem ser generalizadas para sistemas dinâmicos com vários graus de liberdade. Seja a estrutura representada pela Figura 2.3.

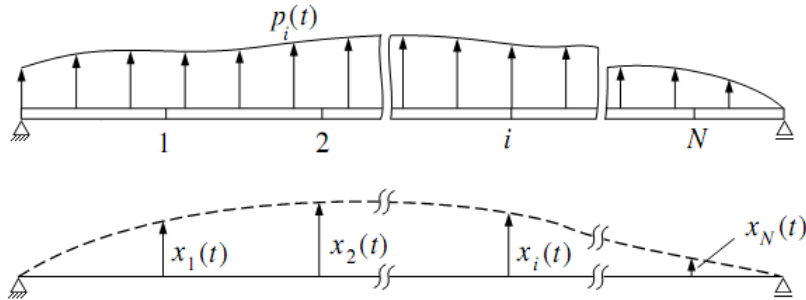


Figura 2.3: Discretização de uma estrutura de viga arbitrária. Extraído da referência [11].

As equações de movimento de sistemas com vários graus de liberdade podem ser escritas expressando-se, de maneira similar aos sistemas de 1 GL, o equilíbrio de forças associadas a cada um dos graus de liberdade. De forma geral, quatro tipos de forças atuarão em um ponto i : o carregamento externo aplicado $p_i(t)$ e as forças de inércia f_{Ii} , de amortecimento f_{Ai} e elástica f_{Ei} . Assim, para cada grau

de liberdade, as equações de equilíbrio se escrevem como:

$$\begin{aligned}
 f_{I1} + f_{A1} + f_{E1} &= p_1(t), \\
 f_{I2} + f_{A2} + f_{E2} &= p_2(t), \\
 &\vdots \\
 f_{IN} + f_{AN} + f_{EN} &= p_N(t).
 \end{aligned}
 \tag{2.19}$$

Escrevendo as Eqs.(2.18) em notação vetorial,

$$\mathbf{f}_I(t) + \mathbf{f}_A(t) + \mathbf{f}_E(t) = \mathbf{p}(t),
 \tag{2.20}$$

tem-se a representação equivalente à Eq.(??) para sistemas com vários graus de liberdade.

Cada força interna pode ser escrita de maneira mais conveniente utilizando-se um conjunto de coeficientes de influência. Considerando, por exemplo, a componente da força elástica atuante no ponto 1, fica claro que ela depende das componentes dos deslocamentos dos outros pontos da estrutura. Assim,

$$f_{E1} = k_{11}x_1 + k_{12}x_2 + \dots + k_{1N}x_N.
 \tag{2.21}$$

Similarmente, para o grau de liberdade x_2 ,

$$f_{E2} = k_{21}x_1 + k_{22}x_2 + \dots + k_{2N}x_N.
 \tag{2.22}$$

Generalizando,

$$f_{Ei} = k_{i1}x_1 + k_{i2}x_2 + \dots + k_{iN}x_N.
 \tag{2.23}$$

Nestas expressões assume-se que o sistema possui comportamento linear e que, portanto, o princípio da superposição de efeitos se aplica. Os coeficientes k_{ij}

são denominados coeficientes de influência de rigidez e são definidos como:

k_{ij} = força correspondente à coordenada i devido
ao deslocamento unitário da coordenada j .

Escrevendo estas expressões em forma matricial, vem:

$$\begin{Bmatrix} f_{E1} \\ f_{E2} \\ \vdots \\ f_{Ei} \\ \vdots \\ f_{EN} \end{Bmatrix} = \begin{bmatrix} k_{11} & k_{12} & \dots & k_{1i} & \dots & k_{1N} \\ k_{21} & k_{22} & \dots & k_{2i} & \dots & k_{2N} \\ \vdots & \vdots & \ddots & \vdots & \dots & \vdots \\ k_{i1} & k_{i2} & \dots & k_{ii} & \dots & k_{iN} \\ \vdots & \vdots & \dots & \vdots & \ddots & \vdots \\ k_{N1} & k_{N2} & \dots & k_{Ni} & \dots & k_{NN} \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ \vdots \\ x_i \\ \vdots \\ x_N \end{Bmatrix} \quad (2.24)$$

ou simplesmente,

$$\mathbf{f}_E = \mathbf{K}\mathbf{x}, \quad (2.25)$$

onde a matriz de coeficientes de rigidez \mathbf{K} é denominada matriz de rigidez do sistema e \mathbf{x} representa o vetor de deslocamentos.

Desenvolvendo de maneira similar para a força de amortecimento e a força inercial, tem-se:

$$\mathbf{f}_A = \mathbf{C}\dot{\mathbf{x}}, \quad (2.26)$$

onde a matriz de coeficientes de amortecimento \mathbf{C} é denominada matriz de amortecimento do sistema, e

$$\mathbf{f}_I = \mathbf{M}\ddot{\mathbf{x}}, \quad (2.27)$$

onde a matriz de coeficientes de massa \mathbf{M} é denominada matriz de massa do sistema.

Substituindo as Eqs.(2.24), (2.25) e (2.26) na Eq.(2.19), obtém-se a expressão completa de equilíbrio dinâmico, considerando todos os graus de liberdade do

sistema:

$$\mathbf{M}\ddot{\mathbf{x}} + \mathbf{C}\dot{\mathbf{x}} + \mathbf{K}\mathbf{x} = \mathbf{p}(t). \quad (2.28)$$

Assim, a Eq.(2.27) expressa as N equações diferenciais de movimento que definem a resposta de um sistema com vários graus de liberdade e é a generalização da Eq.(2.4).

2.2.1 Análise de Vibrações Livres

A equação de movimento para um sistema com vários graus de liberdade não-amortecido em vibrações livres pode ser escrita como:

$$\mathbf{M}\ddot{\mathbf{x}} + \mathbf{K}\mathbf{x} = \mathbf{0}, \quad (2.29)$$

onde $\mathbf{0}$ representa um vetor de zeros.

Assume-se que a solução da equação Eq.(2.28) seja do tipo:

$$\mathbf{x}(t) = \hat{\mathbf{x}} \sin(\omega t + \theta), \quad (2.30)$$

onde $\hat{\mathbf{x}}$ representa um vetor que caracteriza a forma de vibração do sistema e θ é o ângulo de fase. Substituindo a derivada segunda da Eq.(2.29) na Eq.(2.28), vem:

$$-\omega^2 \mathbf{M}\hat{\mathbf{x}} \sin(\omega t + \theta) + \mathbf{K}\hat{\mathbf{x}} \sin(\omega t + \theta) = \mathbf{0}. \quad (2.31)$$

A Eq.(2.30) pode ser simplificada:

$$[\mathbf{K} - \omega^2 \mathbf{M}]\hat{\mathbf{x}} = \mathbf{0}. \quad (2.32)$$

A Eq.(2.31) é um problema de autovalor generalizado, em que ω^2 representam os autovalores equivalentes ao quadrado das frequências de vibração livre e os vetores $\hat{\mathbf{x}}$ representam os correspondentes autovetores.

Para uma solução não-trivial da Eq.(2.31), deve-se ter:

$$\det[\mathbf{K} - \omega^2\mathbf{M}] = 0, \quad (2.33)$$

que é denominada equação de freqüências naturais de vibração e corresponde a uma equação algébrica de grau igual ao número de graus de liberdade do sistema. As N raízes $(\omega_1^2, \omega_2^2, \dots, \omega_N^2)$ representam os quadrados das freqüências naturais das N formas de vibração possíveis no sistema. Considera-se a primeira forma de vibração aquela associada à menor freqüência ω_i .

Por conveniência, os autovetores $\hat{\mathbf{x}}$ são normalizados dividindo-se todas componentes pela maior componente do vetor. O vetor adimensional resultante é denominado modo de vibração do sistema e é representado por ϕ . Para o n -ésimo modo de vibração, tem-se:

$$\phi_n = \begin{Bmatrix} \phi_{1n} \\ \phi_{2n} \\ \vdots \\ \phi_{Nn} \end{Bmatrix} \equiv \frac{1}{\hat{x}_{kn}} \begin{Bmatrix} 1 \\ \hat{x}_{2n} \\ \vdots \\ \hat{x}_{Nn} \end{Bmatrix}, \quad (2.34)$$

onde \hat{x}_{kn} é a maior componente de $\hat{\mathbf{x}}_n$, neste caso a primeira componente do vetor.

Desta forma, monta-se uma matriz quadrada contendo os N modos de vibração do sistema, representada por Φ . Assim,

$$\Phi = \begin{bmatrix} \phi_1 & \phi_2 & \dots & \phi_N \end{bmatrix} = \begin{bmatrix} \phi_{11} & \phi_{12} & \dots & \phi_{1N} \\ \phi_{21} & \phi_{22} & \dots & \phi_{2N} \\ \vdots & \vdots & \dots & \vdots \\ \phi_{N1} & \phi_{N2} & \dots & \phi_{NN} \end{bmatrix}. \quad (2.35)$$

Para um sistema amortecido ($\mathbf{C} \neq 0$), o problema de autovalor resultante fornece pólos s e modos de vibração ϕ_n compostos por pares de números complexos e conjugados. Em uma forma discreta, a solução da Eq.(2.27) para sistemas em

vibrações livres amortecidas pode ser escrita na forma [13]:

$$\mathbf{x}_n(t_j) = \sum_{r=1}^{2N} \phi_{nr} e^{s_r t_j}, \quad (2.36)$$

onde $\mathbf{x}_n(t_j)$ representa o vetor de deslocamentos na posição n e no tempo t_j , ϕ_{nr} é a n -ésima componente do r -ésimo autovetor e s_r é o r -ésimo par de raízes conjugadas do sistema, dado por:

$$s_r = a_r \pm I b_r = -\xi_r \omega_r \pm I \omega_r \sqrt{1 - \xi_r^2}, \quad (2.37)$$

onde $I = \sqrt{-1}$, ξ_r é a r -ésima taxa de amortecimento, ω_r é a r -ésima frequência natural de vibração amortecida correspondente ao r -ésimo modo próprio de vibração.

As Eqs.(2.35) e (2.36) serão importantes para a descrição dos métodos de identificação modal de sistemas dinâmicos deformáveis discutidos no capítulo a seguir.

Capítulo 3

Identificação Modal de Sistemas Deformáveis

3.1 Introdução

As medidas dinâmicas obtidas de ensaios estruturais são, de uma maneira geral, tomadas em função do tempo: acelerações, deslocamentos, deformações, etc. Este fato motiva o desenvolvimento de técnicas de identificação modal que trabalhem diretamente com estas medidas, eliminando assim a ocorrência de erros numéricos que possam ser introduzidos no problema quando se passa do domínio do tempo para outros domínios de análise.

Geralmente, os algoritmos de identificação no domínio do tempo têm como principal dificuldade a necessidade de uma resposta em vibrações livres do sistema estrutural e, na prática, esta necessidade inviabilizaria análises envolvendo estruturas nas quais, por um motivo qualquer, estas respostas não possam ser diretamente medidas. Porém, com o surgimento de técnicas que a partir da resposta da estrutura excitada aleatoriamente aproximam a resposta em vibrações livres de uma forma computacionalmente eficiente e com uma boa precisão, os métodos temporais ganham uma maior potencialidade.

Este capítulo tem como objetivo principal descrever duas técnicas que são amplamente aplicadas na identificação de sistemas dinâmicos deformáveis no domínio do tempo. Na seção a seguir, apresenta-se o Método do Decremento Aleatório

[14], [15], [16], [17], [18], [19] e, em seguida, o Método de Ibrahim [20], [21], [22] e [23].

3.2 Método do Decremento Aleatório

O método do Decremento Aleatório foi desenvolvido por Cole [24], [25], [26] nos anos 60 como uma alternativa ao algoritmo FFT (Fast Fourier Transform ou Transformada Rápida de Fourier, em português) na análise de respostas dinâmicas.

A idéia é estimar as chamadas funções de decremento aleatório, tomando-se a média de segmentos das medidas dinâmicas que obedecem a condições precisas. A partir da construção destas funções é então possível extrair os parâmetros modais da estrutura - frequências naturais, taxas de amortecimento e modos próprios de vibração - considerando-se, apenas, a resposta em vibrações livres do sistema. O método do decremento aleatório é bastante atrativo pois sua implementação computacional é simples e seu tempo de processamento é relativamente pequeno. Em certos casos, pode ser até 100 vezes mais rápido que o algoritmo FFT [19].

3.2.1 Princípios do Método do Decremento Aleatório

O princípio deste método está baseado no cálculo da média de segmentos da resposta de um sistema dinâmico, obtidos a partir da determinação de *pontos de desencadeamento*. Calculando os valores médios e utilizando condições iniciais específicas denominadas *condições de desencadeamento* para cada segmento, a média da resposta da estrutura se reduz à resposta em vibrações livres, causada por um deslocamento inicial, conforme explicado a seguir.

Considere a resposta aleatória $X(t)$ medida em um ponto arbitrário de uma determinada estrutura, sujeita à condição de desencadeamento T1 que define N_p pontos de desencadeamento, como mostra a Figura 3.1. De acordo com a Eq.(2.15), a resposta $X(t)$ é composta de três parcelas:

$$X(t) = X_{X_0}(t) + X_{\dot{X}_0}(t) + X_p(t), \quad (3.1)$$

onde $X_{X_0}(t)$ é a resposta produzida pelo deslocamento inicial no tempo $t = t_0$, $X_{\dot{X}_0}(t)$ é a resposta referente à velocidade inicial \dot{X}_0 no tempo $t = t_0$ e $X_p(t)$ é a parcela devida ao carregamento aleatório aplicado à estrutura entre os instantes t_0 e t . O objetivo é tomar médias destas respostas de forma que as parcelas relativas à velocidade inicial e a resposta forçada reduzam-se a zero.

A resposta temporal causada por uma excitação estacionária de média zero é, igualmente, um processo estacionário de média nula e, ao se tomar médias de vários segmentos temporais, a resposta devida a este carregamento também tenderá a zero.

Desta forma, se todos os segmentos começarem com a mesma condição inicial $X_{X_0}(t_0) = T1$, com alternância de velocidades positivas e negativas, a soma das respostas relativas à velocidade e à excitação aleatória se reduzirão a zero, ao contrário às relativas ao deslocamento inicial, que se manterão.

Portanto, pode-se dizer que com o aumento do número de pontos de desencadeamento, as funções de decremento aleatório tendem a se aproximar das respostas em vibrações livres de um sistema com um deslocamento inicial $X_{X_0}(t_0)$. A Figura 3.1 ilustra este princípio.

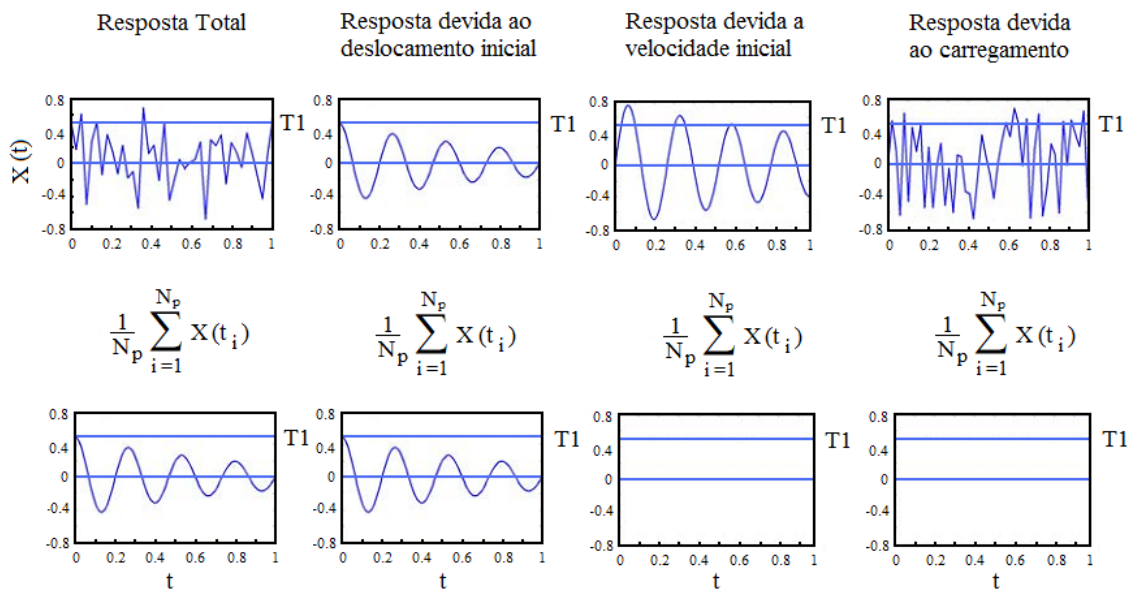


Figura 3.1: Princípio do Método do Decremento Aleatório. Extraído de [27].

Vandiver *et al* [28] mostraram que a função de decremento aleatório estimada

pelo deslocamento inicial é proporcional à função de correlação de um processo gaussiano estacionário. Este resultado é importante porque a proporcionalidade entre as funções de decremento aleatório e as funções de correlação tornou a técnica diretamente comparável às funções da densidade espectral consideradas pelo algoritmo FFT, dado o seu notável desempenho. A relação entre as funções de decremento aleatório e as funções de correlação foi estudada por Brincker *et al* [18] que introduziram a condição geral de desencadeamento e que será discutida nas seções posteriores.

3.2.2 Definição das funções de decremento aleatório

Partindo-se de dois processos (medições dinâmicas) estocásticos, estacionários e de média zero $X(t)$ e $Y(t)$, as funções de decremento aleatório D_{XX} , D_{YY} são definidas como:

$$\begin{aligned} D_{XX}(\tau) &= \mathbb{E} [X(t + \tau) | T_{X(t)}] \text{ e} \\ D_{YY}(\tau) &= \mathbb{E} [Y(t + \tau) | T_{Y(t)}], \end{aligned} \tag{3.2}$$

onde \mathbb{E} representa o valor médio condicional do processo, T é a condição de desencadeamento e τ é a variação de tempo.

O primeiro índice se refere ao processo a partir do qual a média é calculada e o segundo índice se refere ao processo em que a condição é satisfeita.

Similarmente, definem-se as funções cruzadas de decremento aleatório D_{XY} e D_{YX} [20], [21] como sendo:

$$\begin{aligned} D_{XY}(\tau) &= \mathbb{E} [X(t + \tau) | T_{Y(t)}] \text{ e} \\ D_{YX}(\tau) &= \mathbb{E} [Y(t + \tau) | T_{X(t)}]. \end{aligned} \tag{3.3}$$

A título de ilustração, considere um processo estocástico vetorial de dimensão 2×1 que descreve, por exemplo, a resposta de uma estrutura em dois pontos de medida:

$$\mathbf{x}(t) = [X(t) \quad Y(t)]^T. \tag{3.4}$$

Dessa forma, quatro funções de decremento aleatório podem ser definidas:

$$\begin{bmatrix} D_{XX}(\tau) & D_{XY}(\tau) \\ D_{YX}(\tau) & D_{YY}(\tau) \end{bmatrix} = \begin{bmatrix} \mathbb{E} [X(t + \tau)|T_{X(t)}] & \mathbb{E} [X(t + \tau)|T_{Y(t)}] \\ \mathbb{E} [Y(t + \tau)|T_{X(t)}] & \mathbb{E} [Y(t + \tau)|T_{Y(t)}] \end{bmatrix}, \quad (3.5)$$

onde cada coluna da Eq.(3.4) representa um conjunto de funções de decremento aleatório.

Com o objetivo de se estimar corretamente o valor médio condicional \mathbb{E} a partir desse conjunto de medidas, é necessário assumir que os processos estocásticos sejam não somente estacionários, mas também ergódicos, isto é, os seus históricos independem das condições iniciais.

Assim, as funções de decremento aleatório podem ser estimadas como:

$$\begin{aligned} \hat{D}_{XX}(\tau) &= \frac{1}{N_p} \sum_{i=1}^{N_p} X(t_i + \tau)|T_{X(t_i)}, \\ \hat{D}_{YY}(\tau) &= \frac{1}{N_p} \sum_{i=1}^{N_p} Y(t_i + \tau)|T_{Y(t_i)}, \\ \hat{D}_{XY}(\tau) &= \frac{1}{N_p} \sum_{i=1}^{N_p} X(t_i + \tau)|T_{Y(t_i)} \text{ e} \\ \hat{D}_{YX}(\tau) &= \frac{1}{N_p} \sum_{i=1}^{N_p} Y(t_i + \tau)|T_{X(t_i)}, \end{aligned} \quad (3.6)$$

onde N_p é o número de pontos temporais do processo que verifica a condição de desencadeamento.

Deve-se ressaltar que a convergência das estimativas mostradas pelas Eqs.(3.5) depende do número de pontos N_p utilizados na aproximação, isto é, da dimensão do histórico dos processos.

A Figura 3.2 mostra um exemplo simples para se estimar as funções $\hat{D}_{XX}(\tau)$ e $\hat{D}_{YX}(\tau)$ para dois processos $X(t)$ e $Y(t)$, dada a condição de desencadeamento $T_{X(t_1)} = T_{X(t_2)} = T1$. Apresenta-se, ainda, o conceito de janela de correlação que define o número de pontos utilizados para aproximar as funções de decremento aleatório como funções de correlação, que serão descritas posteriormente.

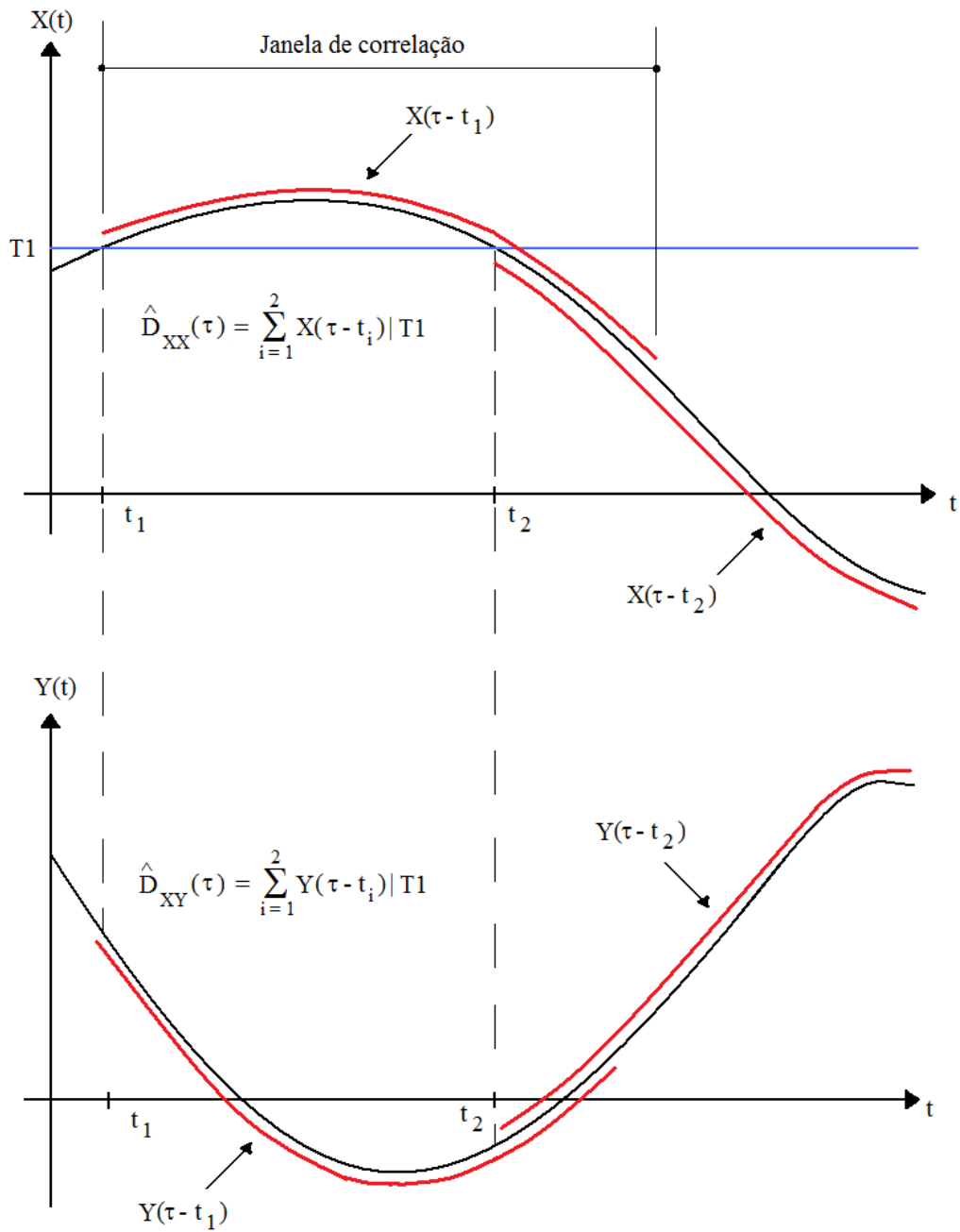


Figura 3.2: Exemplo de estimativa das funções de Decremento Aleatório.

3.2.3 Condição de desencadeamento

a) Condição geral teórica de desencadeamento

A condição geral teórica $T_{X(t)}^{G_T}$ para um processo estocástico $X(t)$ é definida como:

$$T_{X(t)}^{G_T} = \left\{ X(t) = a, \dot{X}(t) = b \right\}, \quad (3.7)$$

onde o índice G_T se refere à condição teórica de desencadeamento. Esta condição é difícil de ser manipulada em aplicações práticas, pois como a probabilidade do evento $T_{X(t)}^{G_T}$ é pequena, é necessário um grande número de amostras de $X(t)$ para que se possa selecionar um número satisfatório de pontos de desencadeamento.

Devido a esta limitação, torna-se necessário a utilização de condições menos exigentes. Porém, antes de defini-las, é preciso apresentar a relação existente entre as funções de correlação e as funções de decremento aleatório [14], uma vez que estas relações serão utilizadas na descrição das condições de desencadeamento.

b) Função de decremento aleatório e funções de correlação

Vários autores discutiram a relação entre as funções de decremento aleatório e as funções de correlação. Esta idéia foi apresentada inicialmente por Vandiver *et al* [28] e prosseguida por Brincker *et al*, 1990 [17], Brincker *et al*, 1991 [18] e Brincker *et al*, 1994 [15].

O princípio da relação entre estas funções está baseado na hipótese de que os processos estocásticos são gaussianos estacionários de média zero. Considerando esta hipótese e utilizando as condições gerais teóricas de desencadeamento $T_{X(t)}^{G_T}$, pode-se determinar as expressões que relacionam as funções de decremento aleatório às funções de correlação R e às suas derivadas temporais R' .

Sejam dois processos vetoriais \mathbf{x}_1 e \mathbf{x}_2 :

$$\mathbf{x}_1 = \left\{ \begin{array}{c} X(t + \tau) \\ Y(t + \tau) \end{array} \right\} \quad \text{e} \quad \mathbf{x}_2 = \left\{ \begin{array}{c} X(t) \\ \dot{X}(t) \end{array} \right\}, \quad (3.8)$$

onde $X(t)$ é considerado um processo gaussiano estacionário e, desta forma, $X(t)$

e $\dot{X}(t)$ são não-correlatos e independentes. As matrizes de correlação são dadas por:

$$\begin{aligned}
\mathbf{R}_{\mathbf{x}_1\mathbf{x}_1} &= \begin{bmatrix} R_{XX}(0) & R_{XY}(0) \\ R_{YX}(0) & R_{YY}(0) \end{bmatrix} = \begin{bmatrix} \sigma_X^2 & R_{XY}(0) \\ R_{YX}(0) & \sigma_Y^2 \end{bmatrix}, \\
\mathbf{R}_{\mathbf{x}_2\mathbf{x}_2} &= \begin{bmatrix} R_{XX}(0) & R_{X\dot{X}}(0) \\ R_{\dot{X}X}(0) & R_{\dot{X}\dot{X}}(0) \end{bmatrix} = \begin{bmatrix} \sigma_X^2 & 0 \\ 0 & \sigma_{\dot{X}}^2 \end{bmatrix}, \\
\mathbf{R}_{\mathbf{x}_1\mathbf{x}_2}(\tau) &= \begin{bmatrix} R_{XX}(\tau) & R_{X\dot{X}}(\tau) \\ R_{YX}(\tau) & R_{Y\dot{X}}(\tau) \end{bmatrix} = \begin{bmatrix} R_{XX}(\tau) & -R'_{XX}(\tau) \\ R_{YX}(\tau) & -R'_{YX}(\tau) \end{bmatrix},
\end{aligned} \tag{3.9}$$

onde R' é a derivada temporal de R e σ_X^2 é a variância de $X(t)$ e que é igual a $R_{XX}(0)$. Substituindo as Eqs.(3.8) na expressão que calcula a média condicional de dois vetores \mathbf{x}_1 e \mathbf{x}_2 de média nula [29], [30] e utilizando a condição geral teórica de desencadeamento definida pela Eq.(3.6), vem:

$$\mathbb{E}[\mathbf{x}_1|\mathbf{x}_2] = (\mathbf{R}_{\mathbf{x}_1\mathbf{x}_2}\mathbf{R}_{\mathbf{x}_2\mathbf{x}_2}^{-1})\mathbf{x}_2 = \begin{bmatrix} R_{XX}(\tau) & -R'_{XX}(\tau) \\ R_{YX}(\tau) & -R'_{YX}(\tau) \end{bmatrix} \begin{bmatrix} \sigma_X^{-2} & 0 \\ 0 & \sigma_{\dot{X}}^{-2} \end{bmatrix} \begin{Bmatrix} a \\ b \end{Bmatrix}. \tag{3.10}$$

Simplificando a Eq.(3.9) e utilizando a definição das funções de decremento aleatório mostrada nas Eqs.(3.1) e (3.2), obtém-se a relação entre estas funções e as funções de correlação:

$$\begin{aligned}
D_{XX}(\tau) &= \frac{R_{XX}(\tau)}{\sigma_X^2}a - \frac{R'_{XX}(\tau)}{\sigma_{\dot{X}}^2}b \quad \text{e} \\
D_{YX}(\tau) &= \frac{R_{YX}(\tau)}{\sigma_X^2}a - \frac{R'_{YX}(\tau)}{\sigma_{\dot{X}}^2}b.
\end{aligned} \tag{3.11}$$

O objetivo é, portanto, aproximar as funções de decremento aleatório como funções de correlação e aplicá-las nos algoritmos para extração dos parâmetros modais sobre estas funções.

c) Condição específica de desencadeamento

Utilizando a condição geral teórica de desencadeamento e a relação existente entre as funções de decremento aleatório e as funções de correlação, pode-se definir qualquer outra condição.

Considerando dois processos gaussianos estacionários e de média zero $X(t)$ e $Y(t)$, a condição geral aplicada de desencadeamento se define como:

$$T_{X(t)}^{G_A} = \left\{ a_1 \leq X(t) < a_2, \quad b_1 \leq \dot{X}(t) < b_2 \right\}, \quad (3.12)$$

onde o índice G_A se refere à condição geral aplicada de desencadeamento. Vale ressaltar, ainda, que Eq.(3.11) não tem nenhuma restrição sobre os limites de desencadeamento, $[a_1 \ a_2]$ e $[b_1 \ b_2]$. Assim, as funções de correlação e de decremento são dadas por [15]:

$$\begin{aligned} D_{XX}(\tau) &= \frac{R_{XX}(\tau)}{\sigma_X^2} \tilde{a} - \frac{R'_{XX}(\tau)}{\sigma_{\dot{X}}^2} \tilde{b} \quad e \\ D_{YX}(\tau) &= \frac{R_{YX}(\tau)}{\sigma_X^2} \tilde{a} - \frac{R'_{YX}(\tau)}{\sigma_{\dot{X}}^2} \tilde{b}, \end{aligned} \quad (3.13)$$

onde \tilde{a} e \tilde{b} são definidos como:

$$\tilde{a} = \frac{\int_{a_1}^{a_2} x p_X(x) dx}{\int_{a_1}^{a_2} p_X(x) dx} \quad e \quad \tilde{b} = \frac{\int_{b_1}^{b_2} \dot{x} p_{\dot{X}}(\dot{x}) d\dot{x}}{\int_{b_1}^{b_2} p_{\dot{X}}(\dot{x}) d\dot{x}}. \quad (3.14)$$

Desta forma, escolhendo-se os limites de desencadeamento a_1, a_2 e/ou b_1, b_2 de forma conveniente, as funções de decremento aleatório podem ser aproximadas às funções de correlação.

A Tabela 3.1 mostra algumas condições de desencadeamento freqüentemente utilizadas e que foram implementadas na aplicação MIDaS.

A seguir serão explicadas cada uma dessas condições.

1) *Passagem de Nível*

Esta é a condição mais freqüentemente utilizada pelo método do decremento

Tabela 3.1: Condições de desencadeamento frequentemente utilizadas.

Passagem de Nível	$T_{X(t)}^L = \{X(t) = a\}$
Extremo Local	$T_{X(t)}^E = \{a_1 \leq X(t) < a_2, \dot{X}(t) = 0\}$
Passagem de Zero	$T_{X(t)}^Z = \{X(t) = 0, \dot{X}(t) > 0\}$
Pontos Positivos	$T_{X(t)}^P = \{a_1 < X(t) \leq a_2\}$

aleatório. De acordo com esta condição, os pontos de desencadeamento são definidos por:

$$T_{X(t)}^L = \{X(t) = a\}. \quad (3.15)$$

A Eq.(3.14) pode ser escrita da mesma maneira que a expressão geral aplicada de desencadeamento:

$$T_{X(t)}^L = \{a \leq X(t) < a + \Delta a, -\infty \leq \dot{X}(t) < \infty\}, \Delta a \rightarrow 0. \quad (3.16)$$

Substituindo os limites da Eq.(3.15) nas Eqs.(3.13), vem:

$$\tilde{a} = \frac{\int_a^{a+\Delta a} x p_X(x) dx}{\int_a^{a+\Delta a} p_X(x) dx} = a \quad \text{e} \quad \tilde{b} = \frac{\int_{-\infty}^{\infty} \dot{x} p_{\dot{X}}(\dot{x}) d\dot{x}}{\int_{-\infty}^{\infty} p_{\dot{X}}(\dot{x}) d\dot{x}} = 0, \quad (3.17)$$

e substituindo as Eqs.(3.16) nas Eqs.(3.12), as funções de decremento aleatório serão dadas por:

$$D_{XX}(\tau) = \frac{R_{XX}(\tau)}{\sigma_X^2} a \quad \text{e} \quad (3.18)$$

$$D_{YX}(\tau) = \frac{R_{YX}(\tau)}{\sigma_X^2} a.$$

As expressões anteriores mostram que as funções de decremento aleatório são proporcionais às funções de correlação.

A Figura 3.3 mostra os processo de estimativa de D_{XX} com diferentes números de pontos de desencadeamento. Neste exemplo, considera-se $T_{X(t)}^L = 1,5 \sigma_X$. Percebe-se claramente que, aumentando o número de pontos de desencadeamento (de 1 até 12), a curva da função de decremento se estabiliza e converge para a resposta em vibrações livres.

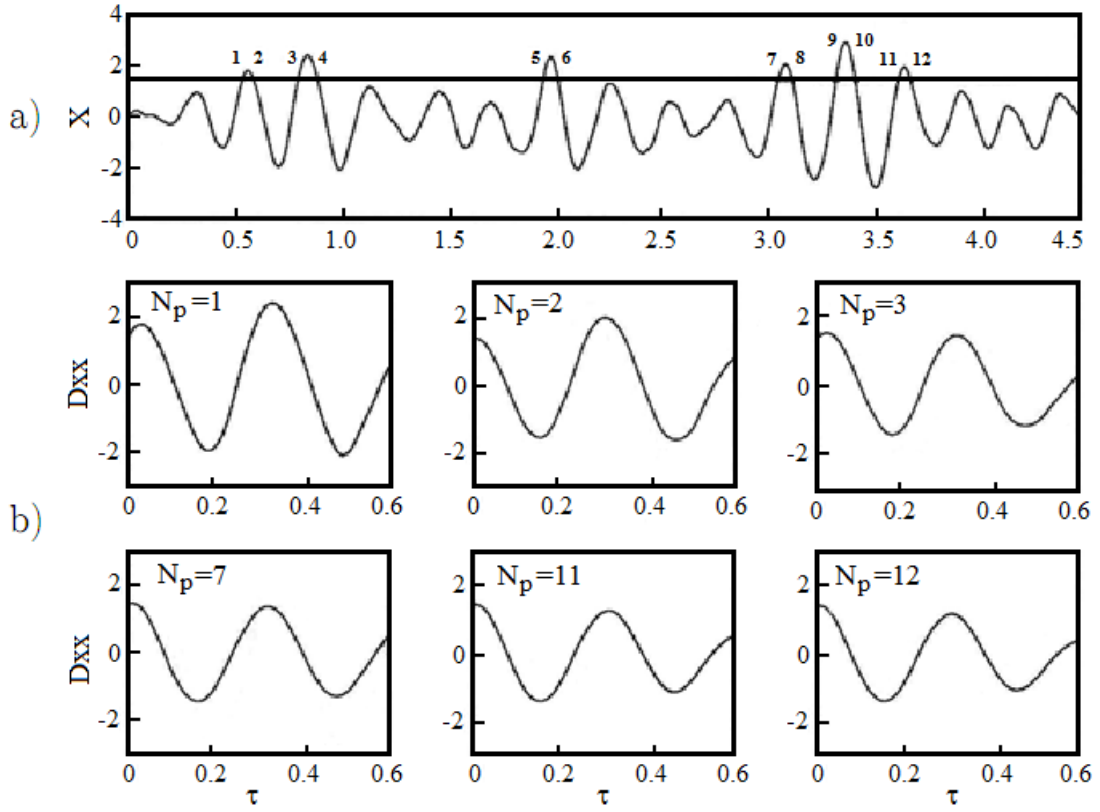


Figura 3.3: a) Pontos de desencadeamento; b) Funções de decremento utilizando diferentes números de pontos de desencadeamento. Extraído da referência [27].

2) *Extremo Local*

Esta condição, ao contrário das demais, não é muito utilizada. Os pontos de desencadeamento são selecionados de forma que o processo esteja dentro do intervalo $[a_1 \ a_2]$ e sua derivada seja nula. Assim,

$$T_{X(t)}^E = \{a_1 \leq X(t) < a_2, \dot{X}(t) = 0\}, \quad 0 \leq a_1 < a_2 < \infty. \quad (3.19)$$

Em geral, a_1 e a_2 devem possuir o mesmo sinal. A Eq.(3.18) pode ser escrita da mesma maneira que a expressão geral aplicada de desencadeamento:

$$T_{X(t)}^E = \{a_1 \leq X(t) < a_2, 0 \leq \dot{X}(t) < 0 + \Delta b\}, \quad \Delta b \rightarrow 0. \quad (3.20)$$

Substituindo os limites da Eq.(3.19) nas Eqs.(3.13), os níveis de desencade-

amento são definidos como:

$$\tilde{a} = \frac{\int_{a_1}^{a_2} x p_X(x) dx}{\int_{a_1}^{a_2} p_X(x) dx} \quad \text{e} \quad \tilde{b} = \frac{\int_0^{0+\Delta b} \dot{x} p_{\dot{X}}(\dot{x}) d\dot{x}}{\int_0^{0+\Delta b} p_{\dot{X}}(\dot{x}) d\dot{x}} = 0. \quad (3.21)$$

Caso $[a_1 a_2] = [0 \infty]$ (ou alternativamente $[-\infty 0]$) seja escolhido como nível de desencadeamento, o número máximo de pontos de desencadeamento será dado por [18]:

$$\tilde{a} = \sqrt{\frac{2}{\pi}} \sigma_X. \quad (3.22)$$

3) *Passagem de Zero*

Nesta condição os pontos de desencadeamento são selecionados para o caso de cruzamento do processo com a linha zero, em uma inclinação (derivada) positiva:

$$T_{X(t)}^Z = \{X(t) = 0, \dot{X}(t) > 0\}. \quad (3.23)$$

A Eq.(3.22) pode ser escrita da mesma maneira que a expressão geral aplicada de desencadeamento:

$$T_{X(t)}^Z = \{0 \leq X(t) < 0 + \Delta a, 0 \leq \dot{X}(t) < \infty\}, \Delta a \rightarrow 0. \quad (3.24)$$

Substituindo os limites da Eq.(3.23) nas Eqs.(3.13), os níveis de desencadeamento são definidos como:

$$\tilde{a} = \frac{\int_0^{0+\Delta a} x p_X(x) dx}{\int_0^{0+\Delta a} p_X(x) dx} = 0 \quad \text{e} \quad \tilde{b} = \frac{\int_0^{\infty} \dot{x} p_{\dot{X}}(\dot{x}) d\dot{x}}{\int_0^{\infty} p_{\dot{X}}(\dot{x}) d\dot{x}} = \sqrt{\frac{2}{\pi}} \sigma_{\dot{X}}. \quad (3.25)$$

4) *Pontos Positivos*

A condição de desencadeamento por pontos positivos pode ser interpretada como uma generalização da condição de passagem de nível. De uma forma geral, ela é a mais adaptada entre as demais utilizadas. Os pontos de desencadeamento

são definidos para o caso de um processo delimitado por a_1 e a_2 :

$$T_{X(t)}^P = \{a_1 < X(t) \leq a_2\}. \quad (3.26)$$

Os limites devem possuir o mesmo sinal, geralmente positivos. A Eq.(3.25) pode ser escrita da mesma maneira que a expressão geral aplicada de desencadeamento:

$$T_{\dot{X}(t)}^P = \{a_1 < X(t) \leq a_2, -\infty \leq \dot{X}(t) < \infty\}, \quad a_2 > a_1 \geq 0. \quad (3.27)$$

Caso $a_1 \rightarrow a_2$, a condição de pontos positivos tenderá à condição de passagem de nível. Substituindo os limites da Eq.(3.26) nas Eqs.(3.13):

$$\tilde{a} = \frac{\int_{a_1}^{a_2} x p_X(x) dx}{\int_{a_1}^{a_2} p_X(x) dx} \quad \text{e} \quad \tilde{b} = \frac{\int_{-\infty}^{\infty} \dot{x} p_{\dot{X}}(\dot{x}) d\dot{x}}{\int_{-\infty}^{\infty} p_{\dot{X}}(\dot{x}) d\dot{x}} = 0, \quad (3.28)$$

e substituindo as Eqs.(3.27) nas Eqs.(3.12), as funções de decremento aleatório serão dadas por:

$$D_{XX}(\tau) = \frac{R_{XX}(\tau)}{\sigma_X^2} \tilde{a} \quad \text{e} \quad (3.29)$$

$$D_{YX}(\tau) = \frac{R_{YX}(\tau)}{\sigma_X^2} \tilde{a}.$$

Adotando os limites de desencadeamento $[a_1 \ a_2] = [0 \ \infty]$, é possível mostrar que o número de pontos de desencadeamento é máximo [18] e possui valor igual a:

$$\tilde{a} = \sqrt{\frac{2}{\pi}} \sigma_X. \quad (3.30)$$

Na seção seguinte apresenta-se o Método de Ibrahim que torna possível extrair as características modais da estrutura a partir da utilização das funções de correlação, que aproximam a resposta em vibrações livres do sistema.

3.3 Método de Ibrahim

3.3.1 Identificação dos Parâmetros Modais

Conforme visto nas seções anteriores, as funções de decaimento aleatório são proporcionais às funções de correlação que, por sua vez, são equivalentes às respostas em vibração livre do sistema. Desta forma, utilizando as funções de decaimento aleatório e aplicando o Método de Ibrahim, torna-se possível extrair as características modais da estrutura - frequências naturais, amortecimentos e modos próprios de vibração.

O Método de Ibrahim é uma técnica temporal de identificação modal baseada na utilização de duas matrizes deduzidas a partir da resposta transiente do sistema.

Com o objetivo de facilitar o entendimento do leitor, repetem-se aqui as Eqs.(2.35) e (2.36) mostradas na seção 2.2.1. A resposta de uma estrutura em vibrações livres com N graus de liberdade medida na posição n e no tempo t_j pode ser escrita como [13]:

$$\mathbf{x}_n(t_j) = \sum_{r=1}^{2N} \phi_{nr} e^{s_r t_j}, \quad (3.31)$$

onde ϕ_{nr} é a n -ésima componente do r -ésimo autovetor e s_r representa o r -ésimo par de raízes conjugadas do sistema, dado por:

$$s_r = a_r \pm Ib_r = -\xi_r \omega_r \pm I \omega_r \sqrt{1 - \xi_r^2}, \quad (3.32)$$

onde $I = \sqrt{-1}$, ξ_r é a r -ésima taxa de amortecimento, ω_r é a r -ésima frequência natural de vibração correspondente ao r -ésimo modo próprio de vibração.

Assumindo que a resposta do sistema seja tomada sob p pontos de medição em q intervalos de tempo, a Eq.(3.30) para um sistema de dimensão $p \times q$ é dada

por:

$$\begin{bmatrix} x_1(t_1) & x_1(t_2) & \dots & x_1(t_q) \\ x_2(t_1) & x_2(t_2) & \dots & x_2(t_q) \\ \vdots & \vdots & \ddots & \vdots \\ x_p(t_1) & x_p(t_2) & \dots & x_p(t_q) \end{bmatrix} = \begin{bmatrix} \phi_{11} & \phi_{12} & \dots & \phi_{1 \ 2N} \\ \phi_{21} & \phi_{22} & \dots & \phi_{2 \ 2N} \\ \vdots & \vdots & \ddots & \vdots \\ \phi_{p1} & \phi_{p2} & \dots & \phi_{p \ 2N} \end{bmatrix} \begin{bmatrix} e^{s_1 t_1} & e^{s_1 t_2} & \dots & e^{s_1 t_q} \\ e^{s_2 t_1} & e^{s_2 t_2} & \dots & e^{s_2 t_q} \\ \vdots & \vdots & \ddots & \vdots \\ e^{s_{2N} t_1} & e^{s_{2N} t_2} & \dots & e^{s_{2N} t_q} \end{bmatrix}$$

ou, de forma reduzida,

$$\mathbf{X}_{(p \times q)} = \mathbf{\Phi}_{(p \times 2N)} \mathbf{\Lambda}_{(2N \times q)}, \quad (3.33)$$

sendo \mathbf{X} a matriz de resposta (conhecida), $\mathbf{\Phi}$ a matriz de autovetores (desconhecida) e $\mathbf{\Lambda}$ a matriz de autovalores (desconhecida) nos instantes t_j (conhecidos).

Um segundo sistema é formado a partir da Eq.(3.30) defasada no tempo de um incremento Δt :

$$\mathbf{x}_n(t_j + \Delta t) = \sum_{r=1}^{2N} \phi_{nr} e^{s_r(t_j + \Delta t)} \quad (3.34)$$

ou

$$\hat{\mathbf{x}}_n(t_j) = \sum_{r=1}^{2N} \hat{\phi}_{nr} e^{s_r t_j}, \quad (3.35)$$

sendo

$$\hat{\mathbf{x}}_n(t_j) = \mathbf{x}_n(t_j + \Delta t) \quad \text{e} \quad (3.36)$$

$$\hat{\phi}_{nr} = \phi_{nr} e^{s_r \Delta t}, \quad (3.37)$$

isto é,

$$\hat{\mathbf{X}}_{(p \times q)} = \hat{\mathbf{\Phi}}_{(p \times 2N)} \mathbf{\Lambda}_{(2N \times q)}, \quad (3.38)$$

onde $\hat{\mathbf{X}}$ e $\hat{\mathbf{\Phi}}$ representam as matrizes de resposta e de autovetores defasadas no tempo, respectivamente.

Considerando $p = 2N$, é possível operar a inversão das matrizes $\mathbf{\Phi}$ e $\hat{\mathbf{\Phi}}$ das

Eqs.(3.32) e (3.37). Assim:

$$\mathbf{\Lambda}_{(2N \times q)} = \mathbf{\Phi}_{(2N \times 2N)}^{-1} \mathbf{X}_{(2m \times q)}, \quad e \quad (3.39)$$

$$\mathbf{\Lambda}_{(2N \times q)} = \hat{\mathbf{\Phi}}_{(2N \times 2N)}^{-1} \hat{\mathbf{X}}_{(2N \times q)}. \quad (3.40)$$

Igualando a Eq.(3.39) a (3.38), vem:

$$\mathbf{\Phi}^{-1} \mathbf{X} = \hat{\mathbf{\Phi}}^{-1} \hat{\mathbf{X}} \Rightarrow \mathbf{A} \mathbf{X} = \hat{\mathbf{X}}, \quad (3.41)$$

onde

$$\mathbf{A} = \hat{\mathbf{\Phi}} \mathbf{\Phi}^{-1}. \quad (3.42)$$

A matriz \mathbf{A} é denominada matriz de estado do sistema, uma vez que quando operada sobre a matriz \mathbf{X} no instante t , o sistema é levado ao estado $t + \Delta t$.

Pós-multiplicando a Eq.(3.40) por \mathbf{X}^T , vem:

$$\mathbf{A} \mathbf{X} \mathbf{X}^T = \hat{\mathbf{X}} \mathbf{X}^T. \quad (3.43)$$

Pós-multiplicando agora a Eq.(3.42) por $(\mathbf{X} \mathbf{X}^T)^{-1}$ tem-se:

$$\mathbf{A} = (\hat{\mathbf{X}} \mathbf{X}^T) (\mathbf{X} \mathbf{X}^T)^{-1}. \quad (3.44)$$

Desta forma, a matriz de estado \mathbf{A} fica definida apenas em função de medições, podendo ser, portanto, determinada.

Substituindo as Eqs.(3.32) e (3.37) na Eq.(3.40), tem-se:

$$\mathbf{A}(\mathbf{\Phi} \mathbf{\Lambda}) = \hat{\mathbf{\Phi}} \mathbf{\Lambda} \quad (3.45)$$

e considerando a relação da Eq.(3.36), vem:

$$\mathbf{A}(\mathbf{\Phi} \mathbf{\Lambda}) = \mathbf{\Phi} e^{s_r \Delta t} \mathbf{\Lambda} \Rightarrow (\mathbf{A} - e^{s_r \Delta t} \mathbf{I}) \mathbf{\Phi} = \mathbf{0}. \quad (3.46)$$

onde \mathbf{I} é a matriz identidade.

A Eq.(3.45) representa um problema de autovalor. Uma vez determinado os autovalores e autovetores da matriz \mathbf{A} , torna-se então possível determinar as frequências naturais, as taxas de amortecimento e os modos de vibração do sistema.

3.3.2 Obtenção dos parâmetros modais

A partir das Eq.(3.31) e Eq.(3.45), as frequências naturais e os coeficientes de amortecimento podem ser calculados. Considerando os autovalores já conhecidos da matriz \mathbf{A} escritos na forma:

$$e^{s_r \Delta t} = \beta_r + I\gamma_r, \quad (3.47)$$

onde I é a unidade imaginária e utilizando $s_r = a_r + Ib_r$, vem:

$$e^{(a_r + Ib_r)\Delta t} = \beta_r + I\gamma_r. \quad (3.48)$$

A partir da relação de Euler, tem-se que:

$$e^{I\alpha} = \cos \alpha + I \sin \alpha \quad (3.49)$$

e utilizando a representação no plano mostrada pela Figura 3.4, pode-se escrever as seguintes expressões:

$$\beta_r + I\gamma_r = \rho(\cos \alpha + I \sin \alpha) = \rho e^{I\alpha}, \quad (3.50)$$

onde $\rho = \sqrt{\beta_r^2 + \gamma_r^2}$ e $\alpha = \arctan\left(\frac{\gamma_r}{\beta_r}\right)$ são, respectivamente, o módulo e o ângulo de fase entre as componentes real e imaginária de um número complexo.

Substituindo a Eq.(??) na Eq.(3.47) com os valores de ρ e α , tem-se:

$$e^{(a_r + Ib_r)\Delta t} = \left(\sqrt{\beta_r^2 + \gamma_r^2}\right) e^{[I \arctan(\frac{\gamma_r}{\beta_r})]}. \quad (3.51)$$

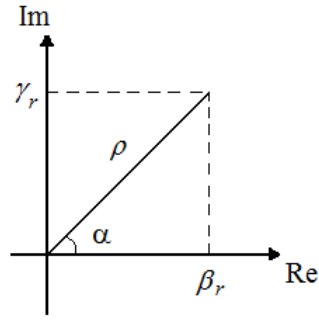


Figura 3.4: Representação de um número complexo.

Aplicando o logaritmo natural em ambos os membros da Eq.(3.49), vem:

$$(a_r + Ib_r)\Delta t = \ln \left(\sqrt{\beta_r^2 + \gamma_r^2} \right) + I \arctan \left(\frac{\gamma_r}{\beta_r} \right). \quad (3.52)$$

Igualando os termos reais e imaginários:

$$a_r = \frac{1}{2\Delta t} \ln (\beta_r^2 + \gamma_r^2), \quad \text{e} \quad (3.53)$$

$$b_r = +\frac{1}{\Delta t} \arctan \left(\frac{\gamma_r}{\beta_r} \right). \quad (3.54)$$

Uma vez determinados os coeficientes a_r e b_r , as freqüências naturais e as taxas de amortecimento são obtidas da Eq.(3.31):

$$\omega_r = \sqrt{a_r^2 + b_r^2}, \quad \text{e} \quad (3.55)$$

$$\xi_r = -\frac{a_r}{\omega_r}. \quad (3.56)$$

Desenvolvendo-se as Eqs.(3.46) até (3.54) para o conjugado $e^{s_r\Delta t} = \beta_r - I\gamma_r$, pode-se facilmente concluir que cada par de pólos s complexos e conjugados (autovalores da matriz \mathbf{A}) produzem freqüências naturais amortecidas e taxas de amortecimento idênticas.

Analisando-se as Eqs.(3.30) e (3.40) e considerando que a matriz de respostas \mathbf{X} descreve um sistema em vibrações livres, pode-se concluir que os autovetores de \mathbf{A} são os próprios modos de vibração ϕ_r do sistema dinâmico instrumentado.

Uma vez que estes modos são também representados por números complexos e conjugados, torna-se necessário desenvolver um procedimento para a obtenção de valores reais.

Os modos de vibração complexos (extraídos dos autovetores da matriz \mathbf{A}) podem ser escritos na forma:

$$\boldsymbol{\phi}_r = \mathbf{c}_r + I\mathbf{d}_r, \quad (3.57)$$

onde \mathbf{c}_r e \mathbf{d}_r são os vetores contendo as partes reais e imaginárias do modo $\boldsymbol{\phi}_r$, respectivamente.

O módulo da n -ésima componente do r -ésimo modo real ϕ_{nr}^* pode ser obtido da seguinte forma:

$$\phi_{nr}^* = \sqrt{c_{nr}^2 + d_{nr}^2}, \quad (3.58)$$

onde c_{nr} e d_{nr} representam a n -ésima componente do r -ésimo vetor contendo as partes reais e imaginárias do modo ϕ_{nr}^* , respectivamente.

Entretanto, o valor obtido na Eq.(3.56) deve ser multiplicado por uma constante η dada por:

$$\eta = \begin{cases} -1, & \text{se } \frac{\pi}{2} < \alpha < \frac{3\pi}{2}, \\ 1, & \text{caso contrário.} \end{cases} \quad (3.59)$$

onde $\alpha = \arctan\left(\frac{d_{nr}}{c_{nr}}\right)$.

Observa-se que para o modo conjugado $\boldsymbol{\phi}_r = \mathbf{c}_r - I\mathbf{d}_r$, obtém-se componentes reais de mesmo módulo porém simétricas àquelas obtidas para o modo $\boldsymbol{\phi}_r = \mathbf{c}_r + I\mathbf{d}_r$, representando, portanto, a mesma forma modal de vibração.

Uma dificuldade aparente da aplicação do Método de Ibrahim é o fato de serem necessários dois sinais de medidas para a determinação de uma frequência natural.

Esta dificuldade, porém, pode ser superada a partir da utilização de artifícios como a divisão ou a defasagem dos sinais criando-se as chamadas *pseudo-medidas*.

Seja, por exemplo, um sinal constituído de 100 valores temporais e que seja dividido em 4 partes (sendo 1 sinal e 3 pseudo-medidas extraídas do sinal), de 25

valores temporais cada uma. Neste caso, será possível determinar duas frequências naturais de vibração.

Para casos em que não se sabe exatamente o número de frequências presentes no sinal, recomenda-se uma super-estimativa do número de pseudo-medidas.

Outra estratégia utilizada para a criação de pseudo-medidas é a defasagem do sinal. Por exemplo, para o mesmo sinal com 100 medições temporais e considerando os sinais defasados de 10 valores no tempo, ter-se-ia:

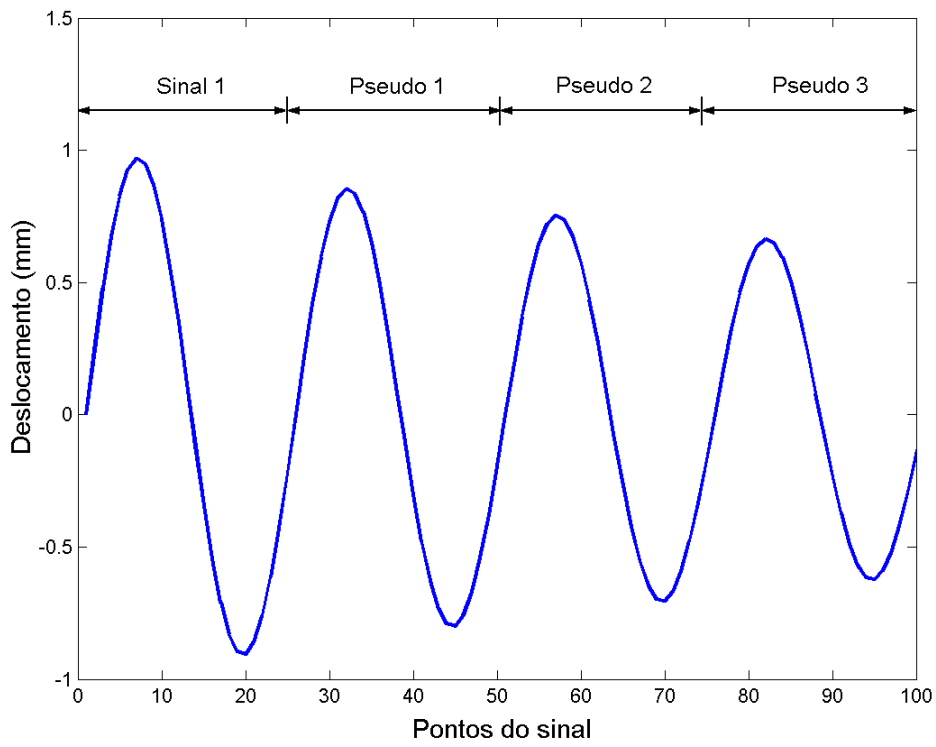
- Sinal 1: valores temporais de 1 a 70;
- Pseudo-medida 2: valores temporais de 11 a 80;
- Pseudo-medida 3: valores temporais de 21 a 90;
- Pseudo-medida 4: valores temporais de 31 a 100;

As Figuras 3.5(a) e 3.5(b) ilustram estes dois procedimentos.

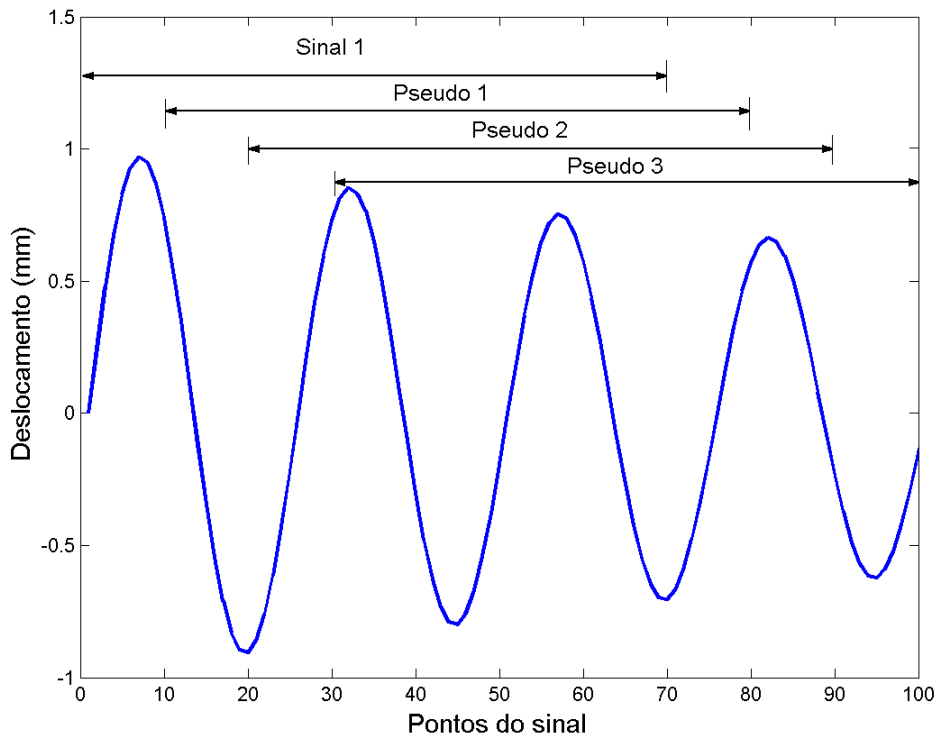
O uso de pseudo-medidas é muito importante para a correta identificação dos parâmetros modais da estrutura. Teoricamente, a análise com várias pseudo-medidas defasadas no tempo deveria conduzir à identificação das mesmas frequências naturais, taxas de amortecimento e modos de vibração. Entretanto, para o caso de medidas reais isto não ocorre, uma vez que as medições temporais não constituem um processo perfeitamente estacionário, levando a pequenas variações nos valores dos parâmetros modais.

A ocorrência de fatores externos como a presença de ruído nos sinais de medição, podem ocasionar o aparecimento de frequências espúrias, ou seja, que não compõem as frequências naturais da estrutura. No entanto, caso estes fatores produzam ruídos com características de estacionariedade e distribuição normal de média zero, o método do Decremento Aleatório é capaz de filtrar estas frequências associadas a esses fatores externos.

De uma forma geral, frequências espúrias devidas a erros numéricos podem ser identificadas durante a aplicação do método de Ibrahim observando-se as suas respectivas taxas de amortecimento, pois, normalmente, estas frequências estão



(a) Exemplo de divisão de um sinal.



(b) Exemplo de defasagem de um sinal.

Figura 3.5: Exemplo de geração de pseudo-medidas do Método de Ibrahim.

associadas a taxas de amortecimento superiores a 10% (valor recomendado pelo

autor e obtido após diversos testes numéricos).

Finalmente, deve-se observar que as componentes modais relativas às pseudo-medidas não fazem parte dos modos de vibração da estrutura.

Capítulo 4

Introdução às Aplicações Web em Java

4.1 A linguagem de programação Java

Java é uma linguagem de programação orientada a objetos que começou a ser desenvolvida na década de 90 pelo programador James Gosling, na empresa Sun Microsystems ®. Uma de suas principais características é a *portabilidade*, ou seja, é uma linguagem independente de plataforma. Este fato representa uma vantagem significativa, uma vez que o desenvolvedor de programas não precisa se preocupar em qual plataforma seu aplicativo irá ser executado, seja em um computador pessoal ou em um aparelho portátil, por exemplo.

A independência de plataforma é uma das razões pelas quais Java é interessante para internet, já que muitas pessoas a acessam através de arquiteturas distintas, sob sistemas operacionais (Linux, Windows, Solaris, etc.) diferentes. Esta capacidade está relacionada ao fato de programas Java não serem traduzidos para a linguagem de máquina como em outras linguagens estaticamente compiladas (C, C++, Fortran, etc.) mas sim para uma representação intermediária, denominada *bytecodes* que são interpretados pela máquina virtual Java (JVM - Java Virtual Machine), como mostra a Figura 4.1.

Considerada uma linguagem de baixo desempenho por alguns desenvolvedores, Java vem recebendo uma série de otimizações como a compilação especulativa, que aproveita o tempo ocioso do processador para pré-compilar os *bytecodes* para o código nativo da máquina, melhorando os tempos de resposta ao usuário.

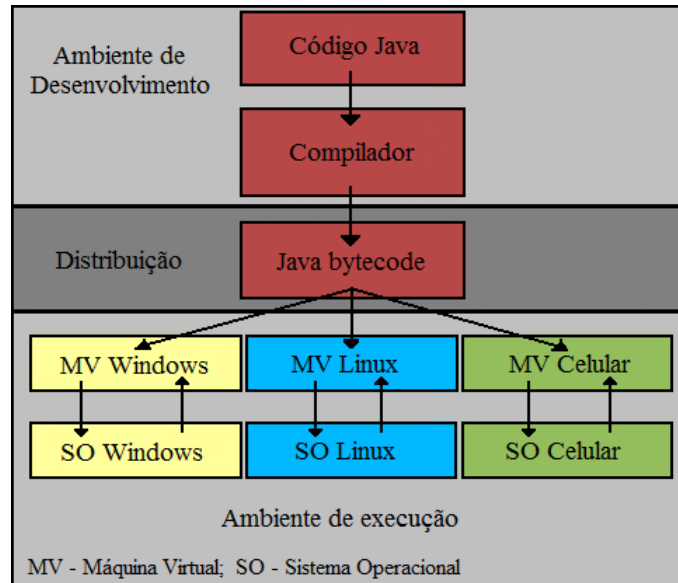


Figura 4.1: Compilador e interpretador Java.

Com o objetivo de comparar o desempenho das linguagens C, Fortran, Java e Matlab, foi feito um teste numérico simulando a multiplicação de dois vetores de dimensão [1000000] (1 milhão). Deve-se ressaltar que todas as linguagens foram compiladas utilizando a otimização máxima (o3). Estes testes foram executados em um computador com processador Pentium 4 3.0 GHz com 1GB de memória RAM, no sistema operacional Linux distribuição Kubuntu 7.0.

O compilador utilizado para o C foi o *gcc*, ao passo que para o Fortran utilizou-se o compilador *gfortran*, sendo ambos desenvolvidos e distribuídos pelo projeto GNU. Para o Java, utilizou-se o compilador presente no ambiente de desenvolvimento Eclipse, desenvolvido pela IBM®.

O programa foi executado 50 vezes para cada linguagem e os tempos médios, máximos e mínimos foram calculados. Os programas em C e Fortran foram executados diretamente da linha de comando, ao contrário do programa em Java que foi executado dentro do ambiente de desenvolvimento Eclipse, por possuir um compilador mais otimizado e de melhor desempenho quando comparado à execução direta em linha de comando, conforme apontaram testes iniciais realizados pelo autor. A Tabela 4.1 mostra os resultados obtidos.

A partir dos resultados obtidos, observa-se que o Java é, na média, 47% mais

Tabela 4.1: Comparativo de tempo entre as linguagens C, Fortran e Java.

Tempo	C	Fortran	Java	Matlab
Mínimo (ms)	37,00	24,00	41,00	91,00
Médio (ms)	40,74	31,76	46,74	101,10
Máximo (ms)	50,00	36,00	61,00	110,00

lento que o Fortran, 15% mais lento que o C e 54% mais rápido que o Matlab.

Cabe ressaltar que existem na comunidade científica máquinas virtuais Java com diferentes tipos de otimização e, dessa forma, as diferenças de desempenho observadas em relação ao C e ao Fortran poderiam ser distintas das obtidas e mostradas na Tabela 4.1.

Java apresentou um desempenho computacional competitivo, porém inferior ao apresentado pelo Fortran e C. Apesar disso, Java oferece alguns recursos que tornam esta linguagem atrativa, tais como:

- Recursos de rede - possui uma extensa biblioteca de rotinas que facilitam a inter-operabilidade com protocolos TCP/IP¹, HTTP² e FTP³;
- Segurança - capaz de executar programas via rede com restrições de execução;
- Versatilidade - possui um vasto conjunto de bibliotecas que abrangem vários tipos e níveis de aplicação;
- Desempenho - comparado a programas comerciais, como o Matlab, o Java possui melhor desempenho em algoritmos numéricos e se aproxima ao tempo gasto por linguagens estaticamente compiladas, como C.
- Gerenciamento de memória - desalocação de memória automática por processo de coletor de lixo (*garbage collector*);

¹ *Transmission Control Protocol* (Protocolo de Controle de Transmissão)/*Internet Protocol* (Protocolo de Internet): conjunto de protocolos de comunicação entre computadores em rede.

² *HyperText Transfer Protocol* (Protocolo de Transferência de Hipertexto): protocolo utilizado para transferência de dados pela internet.

³ *File Transfer Protocol* (Protocolo de Transferência de Arquivos): forma rápida e versátil para transferência de arquivos pela rede.

- Carga Dinâmica de Código - programas em Java são formados por uma coleção de classes armazenadas independentemente e que podem ser carregadas no momento de utilização, entre outras.

Atualmente, o uso do Java vem se disseminando principalmente na internet, onde está presente na maioria dos sítios de bancos, de compras *online*, sítios de pesquisa, entre outros.

Além de aplicações em computadores, diversos programas em Java vêm sendo desenvolvidos para dispositivos portáteis como celulares, *palmtops*, *PDA*s e *blackberries*. Hoje em dia, é possível executar desde aplicativos simples como jogos, agendas eletrônicas, calculadoras até mais robustos como navegadores, leitores de correio eletrônico, tradutores, etc.

A seguir serão apresentadas algumas considerações a respeito de conceitos sobre Java, bem como serão descritas tecnologias importantes para o desenvolvimento deste trabalho. Para maiores detalhes, recomenda-se a leitura da referência [31].

4.1.1 Alguns Conceitos sobre Java

a) *JDK - Java Development Kit*

O JDK (Java Development Kit) é um kit de desenvolvimento Java fornecido livremente pela Sun ®. É constituído por um conjunto de programas que engloba compilador (*javac*), interpretador (máquina virtual) e utilitários (*javadoc* - gerador automático de documentação, *jdb* - *debugger*, etc.), fornecendo um pacote de ferramentas básicas para o desenvolvimento de aplicações Java. Em geral, toda a funcionalidade do JDK já faz parte dos ambientes de desenvolvimento disponíveis (como o Eclipse, NetBeans, JBuilder, entre outros).

b) *JRE - Java Runtime Environment*

Enquanto o JDK é o ambiente voltado para os desenvolvedores, o Java Runtime Environment (JRE) é basicamente composto pela JVM e um conjunto de

bibliotecas necessárias para executar aplicações Java.

Além disso, o JRE contém o “Java Plug-in” que é um recurso capaz de permitir a execução de aplicações em Java em navegadores de internet sem suporte à linguagem ou que suportam, mas com alguma particularidade. É o caso de sítios de bancos, lojas, entre outros.

c) *Plataformas de Desenvolvimento*

Com o aumento do número de aplicações e, conseqüentemente, do número de bibliotecas padrões da linguagem, foi necessária a criação de três divisões na plataforma a partir da versão 2 da linguagem: J2SE, J2EE e J2ME.

J2SE - A J2SE (Java 2 Standard Edition) é a plataforma de desenvolvimento mais utilizada, uma vez que seu uso é voltado para computadores pessoais e servidores. Além disso, é considerada como a mais abrangente por servir, de forma geral, de base para as plataformas J2EE e J2ME.

J2EE - A J2EE (Java 2 Enterprise Edition) é a plataforma Java voltada para redes, internet, intranets e afins. Dessa maneira, ela contém bibliotecas especialmente desenvolvidas para o acesso a servidores, sistemas de correio eletrônico, banco de dados, etc. Por essas características, a J2EE foi desenvolvida para dar suporte a uma grande quantidade de usuários simultâneos e contém uma série de especificações, como as de acesso a banco de dados, criação de aplicações e serviços web, etc.

J2ME - A J2ME (Java 2 Micro Edition) é a plataforma de desenvolvimento para dispositivos móveis ou portáteis, como telefones celulares e palmtops. Ela contém configurações e bibliotecas específicas e, desde que estes dispositivos tenham uma JVM é possível, com poucas modificações, implementar os aplicativos em qualquer aparelho, como jogos e agendas eletrônicas.

Durante o desenvolvimento deste trabalho, foram utilizadas as plataformas J2SE para a implementação dos algoritmos (tendo o JDK e JRE como base) e a J2EE para as definições do servidor de aplicação web.

4.1.2 Ambientes de Desenvolvimento Integrado - IDEs

Os ambientes de desenvolvimento integrado ou simplesmente IDEs (Integrated Development Environments) são programas de computador que reúnem características e ferramentas de apoio ao desenvolvimento de códigos com o objetivo de facilitar e agilizar este processo.

Os IDEs utilizados para o desenvolvimento deste trabalho são o Eclipse e o NetBeans. O primeiro foi iniciado pela IBM® (empresa fabricante de programas e computadores) que desenvolveu a primeira versão e depois o disponibilizou como programa livre para a comunidade. Uma de suas características é o seu compilador otimizado capaz de compilar somente os trechos do código em que houve alteração, sem a necessidade de uma recompilação completa.

O segundo IDE, o NetBeans, foi desenvolvido pela Sun Microsystems® e é atualmente o mais bem integrado para criação e desenvolvimento de programas e aplicações multiplataforma [32]. Para a criação de aplicações web, por exemplo, a versão 6.0 Web & J2EE traz integrada ao pacote de instalação os servidores de aplicação Glassfish [33] e Tomcat (desenvolvido pela Apache) [34], além do kit Visual Web Server Faces [35] de desenvolvimento web que visa facilitar a manipulação de elementos necessários à construção de sítios virtuais.

Para o desenvolvimento deste trabalho, de acordo com as características de cada IDE descritos anteriormente, os algoritmos de identificação de sistemas dinâmicos foram implementados no Eclipse, por possuir melhor performance nos tempos de compilação e execução dos códigos. No entanto, para a criação da aplicação web, utilizou-se o NetBeans por já possuir as ferramentas para desenvolvimento web mais bem integradas e intuitivas.

4.1.3 Bibliotecas

Uma das vantagens já mencionadas para o desenvolvimento de programas em Java, é a sua extensa lista de bibliotecas com os mais diversos tipos e níveis de aplicação.

Bibliotecas são um conjunto de classes disponibilizadas no formato .jar (tipo de compactação de arquivos, como .zip, por exemplo) e que podem ser utilizadas como qualquer outra classe dentro do algoritmo. Entretanto, o desenvolvedor da biblioteca disponibiliza apenas os arquivos .class (que não podem ser lidos) como forma de proteger o código-fonte original.

Neste trabalho foram utilizadas quatro bibliotecas gratuitas para o desenvolvimento da aplicação. A discussão de cada uma delas será feita no Capítulo 5.

4.2 Aplicações web

Como brevemente descrito no Capítulo 1, aplicação web é o termo utilizado para designar, de forma geral, sistemas que são executados pela internet, tipicamente através de um navegador web oferecendo interação direta com o usuário.

A Figura 4.2 mostra um exemplo de aplicação web, o *MSN Spaces* desenvolvida pela Microsoft© [36] e disponibilizado para usuários do programa de bate-papo *MSN*. Nesta aplicação o usuário pode, além de acessar seu correio eletrônico, compartilhar fotos, vídeos, documentos, etc. com pessoas de sua lista de contatos.

Normalmente, a aplicação é composta por imagens, textos, entre outros arquivos estáticos, e uma biblioteca de classes (código de programação compilado) usada na execução do programa.

Com esta tecnologia, é possível o desenvolvimento de aplicativos que não precisam ser instalados no computador do usuário, tornando o processo de manutenção e atualização mais fáceis e menos intrusivo.

Para o desenvolvimento deste trabalho, foram utilizadas como base duas tecnologias para a criação da aplicação web [35]: o JavaServer Pages (JSP) e o JavaServer Faces (JSF). Uma breve descrição de cada uma delas será feita nas seções seguintes.

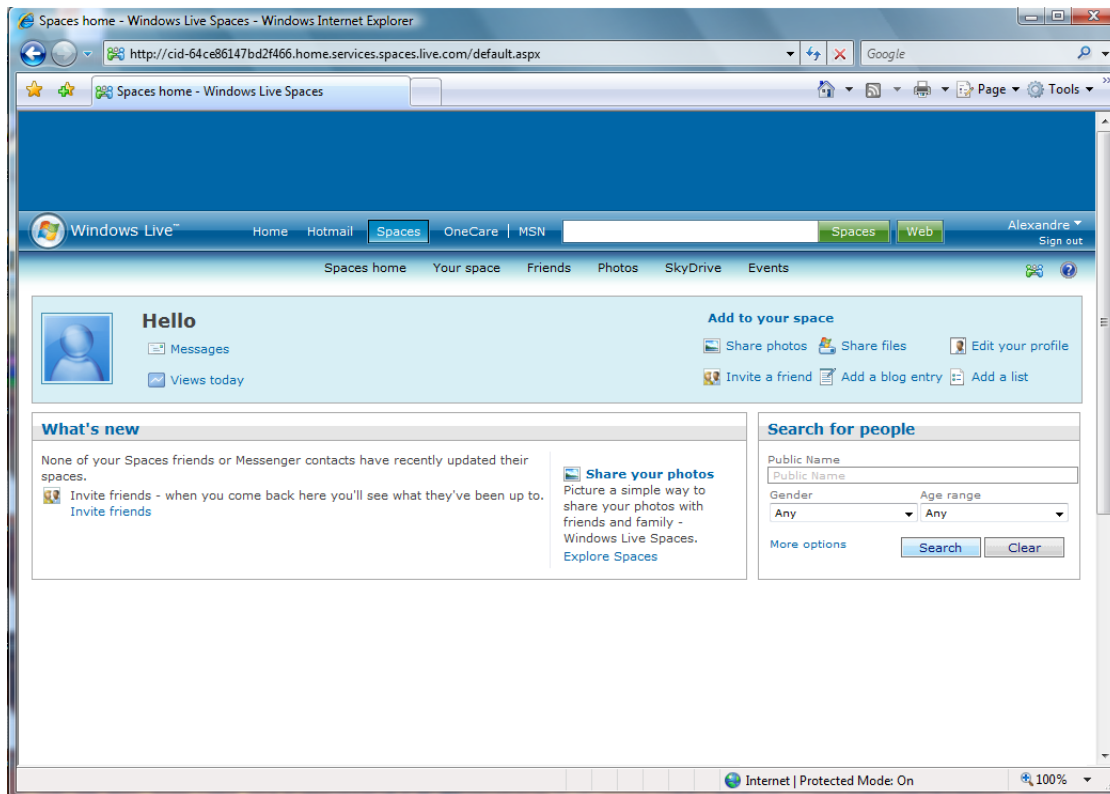


Figura 4.2: Exemplo de aplicação web - *MSN Spaces*.

4.2.1 A Tecnologia JSP (JavaServer Pages)

Em uma definição simples, pode-se dizer que JSP é uma tecnologia orientada para a criação de páginas web dinâmicas⁴ utilizando linguagem de programação Java. Com ela, é possível criar aplicações web que possam ser executadas em vários servidores, sob múltiplas plataformas.

As páginas JSP são arquivos texto representados pela extensão “.jsp”, basicamente compostas de código HTML⁵ /XML⁶ e de etiquetas especiais (*tag's*), utilizadas para programar *scripts*⁷ de servidor em linguagem Java.

Quando uma página JSP é solicitada por um usuário e processada pelo servi-

⁴ Páginas que implementam alguma funcionalidade ou interatividade ausente nas páginas estáticas. Dentre os recursos utilizados destaca-se, de forma geral, o uso de bancos de dados e armazenamento de dados específicos para a aplicação desenvolvida.

⁵ *HyperText Markup Language* (Linguagem de Marcação de Hipertexto): linguagem com a qual se definem as páginas web. Basicamente trata-se de um conjunto de etiquetas (tags) que servem para definir a forma na qual se apresentará o texto e outros elementos da página.

⁶ *EXtensible Markup Language* (Linguagem extensível de formatação): linguagem definida como o formato universal para dados estruturados na web (tabelas, desenhos, parâmetros de configuração, etc.).

⁷ Códigos na linguagem de computador interpretada.

dor HTTP, a parte HTML da página é transmitida. No entanto, a parte de código só é executada no momento em que a solicitação é recebida e o conteúdo dinâmico gerado por este código é unido à página, antes de ser enviada para o usuário [32].

Em outras palavras, esta tecnologia permite separar a programação lógica (parte dinâmica) da programação visual (parte estática), facilitando o desenvolvimento de aplicações mais robustas e de forma independente [37].

Neste trabalho, porém, o JSP não foi diretamente utilizado para a implementação da aplicação web. Por possuir uma interface mais direta e intuitiva, empregou-se o kit Visual Web Faces disponibilizado no IDE NetBeans em conjunto com a tecnologia JavaServer Faces, que será abordada na seção seguinte.

4.2.2 A Tecnologia JSF (JavaServer Faces)

JSF é uma tecnologia que incorpora elementos de um *framework*⁸ MVC⁹ para web e de um modelo de interfaces gráficas baseado em eventos [32].

Deve-se destacar que integrado ao ambiente de desenvolvimento NetBeans na versão Web, existe um kit de desenvolvimento denominado Visual Web Server Faces com o objetivo de permitir ao usuário com conhecimentos básicos de JSP e JSF criar aplicações web de forma rápida e eficiente, uma vez que os códigos que servem como pano de fundo para a aplicação são gerados automaticamente.

No JSF o controle é composto por um *servlet*¹⁰ denominado FacesServlet, por arquivos de configuração e por um conjunto de manipuladores de ações e observadores de eventos. O FacesServlet é responsável por receber requisições da web, redirecioná-las para o modelo e então remeter uma resposta. Os arquivos de configuração são responsáveis por realizar associações e mapeamentos de ações e pela definição de regras de navegação. Os manipuladores de eventos são responsáveis por receber os dados vindos da camada de visualização, acessar o modelo, e então

⁸ Estrutura de suporte definida para que um outro projeto de *software* possa ser organizado e desenvolvido.

⁹ *Model-View-Controller* (Modelo-Visualização-Controle): padrão de arquitetura de *software* utilizada para separar os dados (*Model*) da interface (*View*), de forma que as alterações feitas na interface não afetem a manipulação dos dados.

¹⁰ Programas destinados para execução no servidor.

devolver o resultado para o FacesServlet [38].

4.3 Servidores de Aplicação

Servidores de aplicação são programas que fornecem a infra-estrutura de serviços necessária para o funcionamento de aplicações web e que são executados em servidores acessados por clientes através de uma conexão local ou através da internet.

Em geral, os servidores de aplicação rodam em vários sistemas operacionais, como Solaris, Linux e Windows, o que permite que seja possível o desenvolvimento em uma plataforma e sua utilização em outra.

Existem diversos servidores de aplicação disponíveis para Java, como Glassfish, Tomcat, JBoss [39], entre outros. Para o desenvolvimento da aplicação web deste trabalho, foi adotado o servidor Glassfish por possuir uma forte integração com o ambiente de desenvolvimento NetBeans, na versão 6.0 Web & J2EE.

Além disso, o projeto Glassfish que atualmente está na versão 2.01 e é um dos poucos projetos de servidor de aplicação que possuem suporte total às novas especificações web Java e certificação de compatibilidade com o padrão Java EE 5.0 [33].

Na seção seguinte serão mostrados, brevemente, alguns detalhes e a configuração deste servidor.

4.3.1 Servidor Glassfish

Glassfish [33] é um servidor de aplicação de código livre (*open-source*) desenvolvido pela Sun® e distribuído juntamente com o IDE NetBeans 6.0. É, de forma geral, um programa capaz de dar suporte ao desenvolvimento e a execução de aplicações web criadas segundo os padrões da plataforma Java.

Como configuração padrão, o servidor Glassfish está programado para “escutar” nas portas 8080, 8181 e 4848. Isto significa que para acessar uma aplicação em um determinado computador que possua este servidor instalado, é necessário

saber, além do IP da máquina, em qual porta a aplicação está sendo executada.

Para uma aplicação web, no entanto, deve-se utilizar a porta “80” que é o padrão utilizado para transferência HTTP.

A Figura 4.3 mostra a página inicial do servidor, após iniciá-lo.

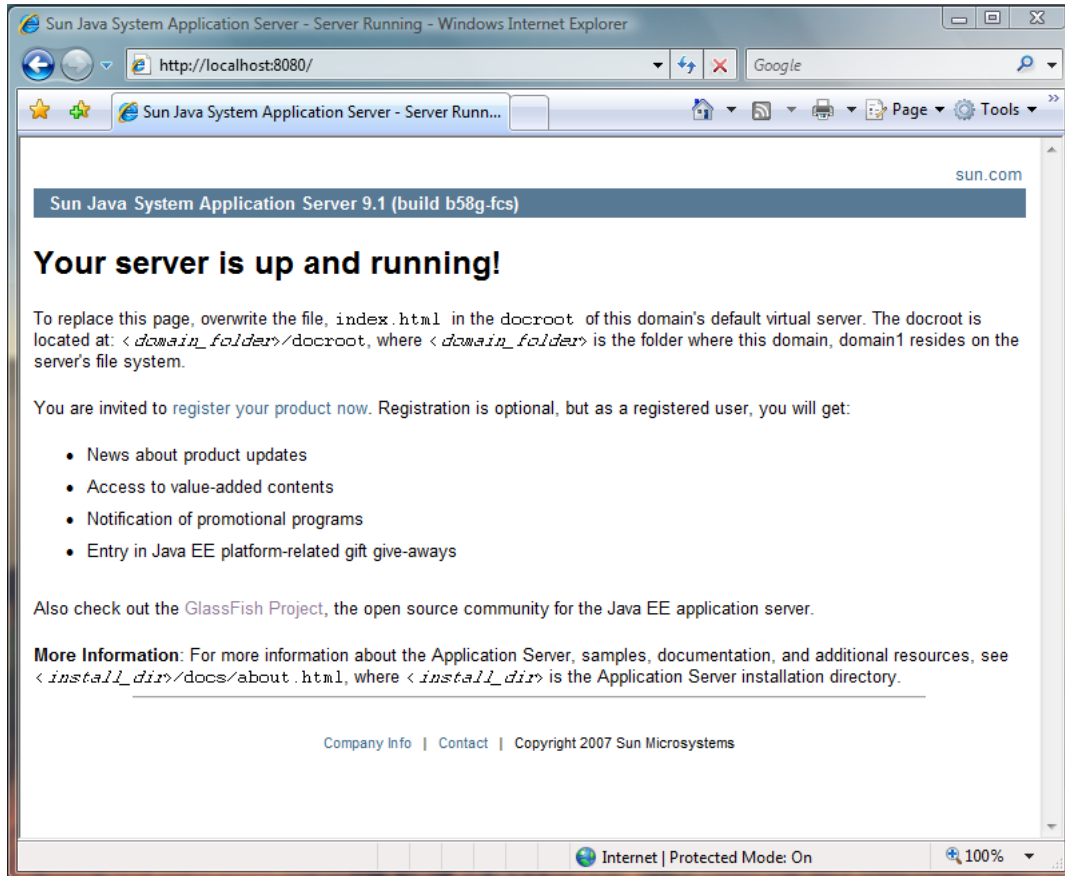


Figura 4.3: Página inicial do servidor Glassfish.

Na barra de endereços, *localhost* significa que a aplicação está sendo acessada da máquina local. Caso contrário, aparecerá o IP da máquina em que o servidor Glassfish estiver sendo executado.

Cabe ressaltar que ao se iniciar o servidor, o JRE é automaticamente carregado para a memória, tornando possível a execução das aplicações.

4.3.2 Arquivo WAR

O arquivo WAR (Web ARchive) compreende um conjunto de elementos que são armazenados em um só arquivo para ser instalado em um servidor de aplicação.

De forma geral, estes elementos podem ser [10]:

- *Servlets*: programas destinados para execução no servidor;
- Páginas JSP (Java Server Pages): geração de conteúdo dinâmico em páginas web;
- Arquivos estáticos: figuras, documentos HTML, entre outros;
- Bibliotecas: uso de conjunto de classes externas importadas no projeto por arquivos .jar;
- Arquivo de configuração (obrigatório): é o descritor da aplicação que consiste em um arquivo XML com o nome “web.xml”. Este arquivo especifica parâmetros de inicialização, regras de segurança e mapeamento de URL’s para *servlets*, entre outras informações.

No IDE NetBeans a criação do arquivo WAR é feita automaticamente, após a recompilação do projeto. De posse deste arquivo, é necessário copiá-lo para a pasta do servidor de aplicação para que a instalação seja concluída e, a aplicação web, esteja pronta para ser utilizada.

4.4 Exemplo de aplicação web

Com o objetivo de ilustrar os conceitos descritos nas seções anteriores, apresenta-se nesta seção um exemplo simples de uma aplicação web utilizando NetBeans e o Visual Web JSF.

O primeiro passo é a criação de um projeto do tipo “Web Application”, definindo um nome (neste caso ExemploWebApp), o servidor de aplicação (a escolha pode ser feita entre os servidores Apache Tomcat e Glassfish, ambos instalados com a versão Web & Java EE do NetBeans) e o *framework* a ser utilizado - Visual Web JavaServer Faces. As Figuras 4.4, 4.5 e 4.6 mostram cada etapa de configuração.

É válido destacar que o próprio ambiente de desenvolvimento se encarrega de criar os códigos que servem como pano de fundo e de suporte para a aplicação.

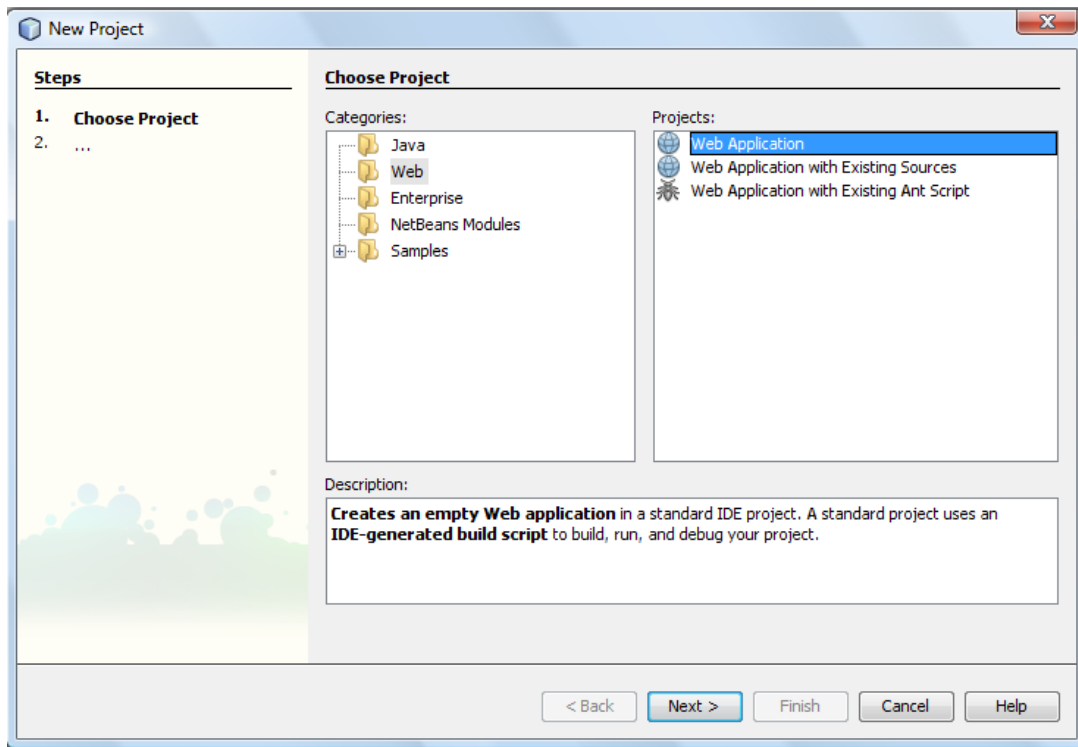


Figura 4.4: Primeiro passo para a criação de uma aplicação web utilizando NetBeans - Escolha do tipo de projeto.

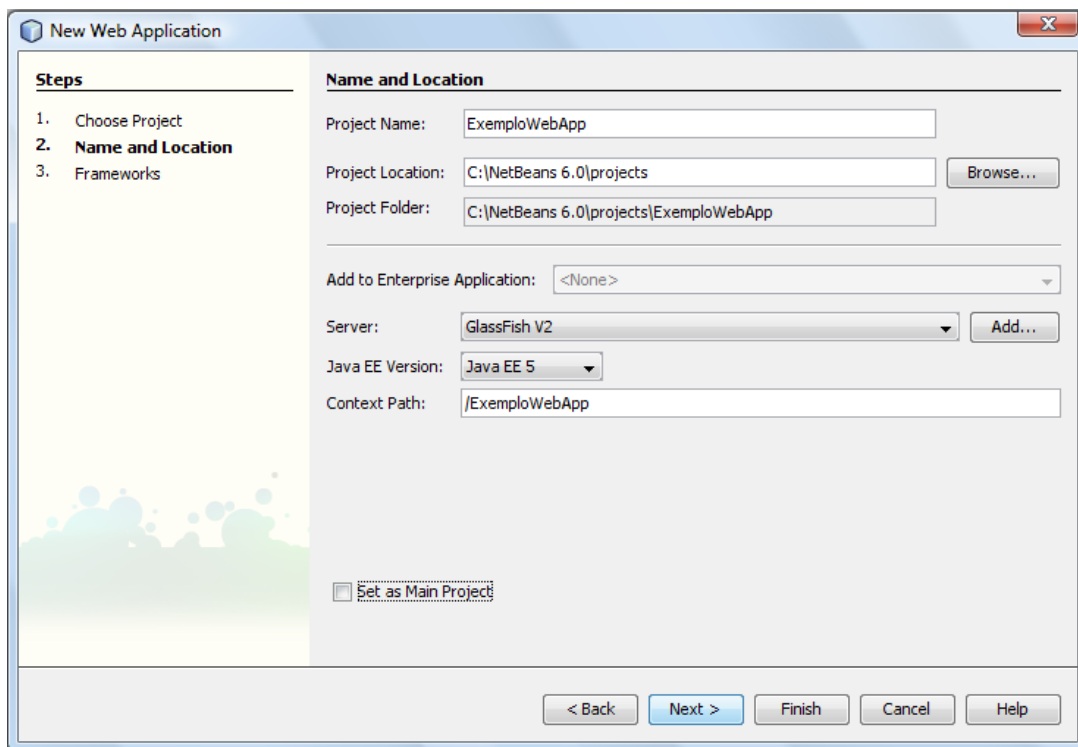


Figura 4.5: Segundo passo para a criação de uma aplicação web utilizando NetBeans - Definição do nome e do servidor de aplicação utilizados no projeto.

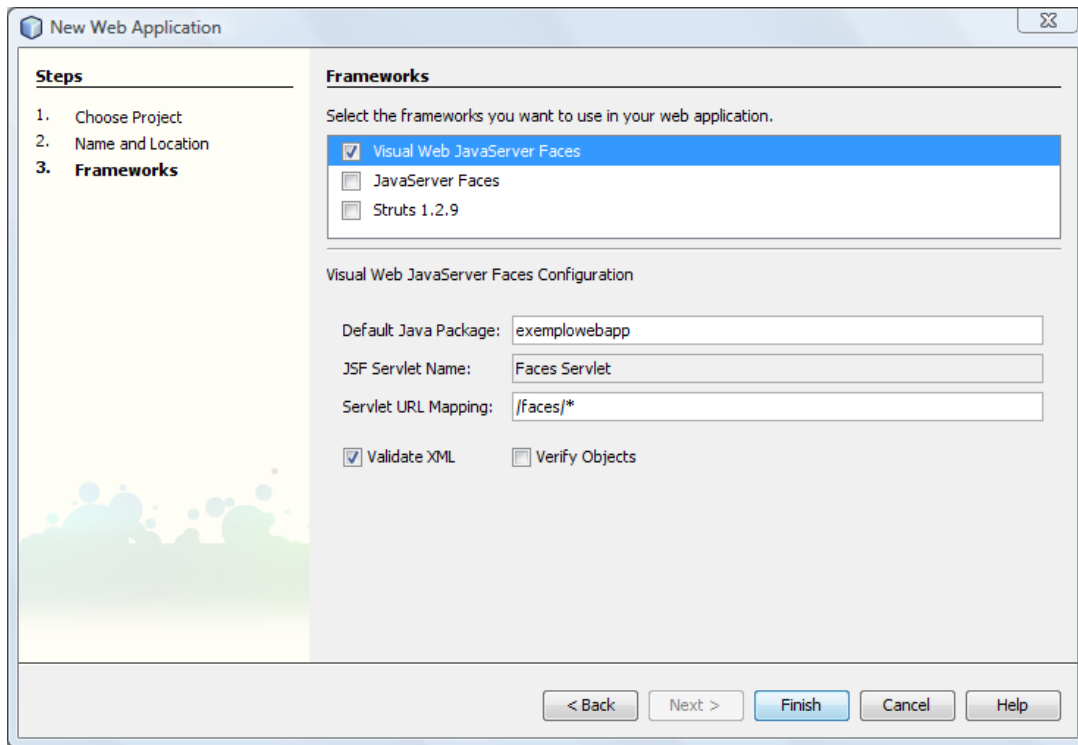


Figura 4.6: Terceiro passo para a criação de uma aplicação web utilizando NetBeans - Escolha do *framework* a ser utilizado.

Desta forma, o usuário deve se preocupar em apenas implementar seu próprio código-fonte em Java e criar a interface que a aplicação possuirá.

A Figura 4.7 mostra a interface do IDE para desenvolvimento de aplicações web. À esquerda, encontra-se o navegador do projeto que mostra todos os arquivos-fonte presentes (*Source Packages* → arquivos .java) e as bibliotecas utilizadas (*Libraries*).

No diretório “Web Pages” são armazenados os arquivos referentes às páginas criadas (.jsp) e os sub-diretórios de configuração, dentre eles o “WEB-INF” que contém o arquivo descritor da aplicação “web.xml” (para definição de filtros, itens de segurança, tempo de sessão, etc), e o arquivo “faces-config.xml” que fornece um mapeamento da aplicação indicando o conteúdo de cada página (textos, *labels*, *hyperlinks*, etc) e a maneira como estas páginas se inter-relacionam.

Neste diretório também são armazenados os arquivos “.class” gerados na compilação e as bibliotecas utilizadas (arquivos “.jar”). Existe, ainda, o sub-diretório “resources” que contém os arquivos de imagens estáticas utilizados (como fundos

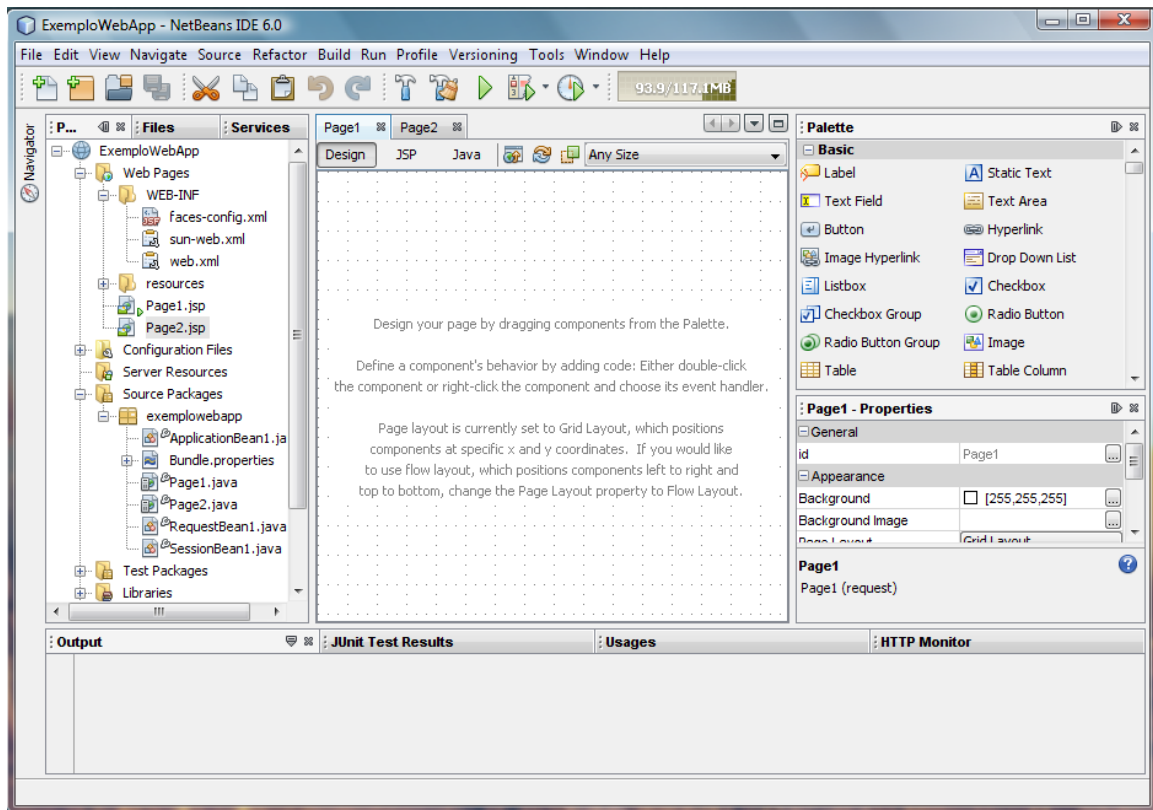


Figura 4.7: Interface do IDE NetBeans com recursos JSF.

de página, por exemplo) e outras configurações específicas como o arquivo “.css”, que guarda as folhas de estilo utilizadas.

De posse de todos estes dados cria-se, então, o arquivo “.war” que será instalado no servidor de aplicação.

Na barra de ferramentas à direita da Figura 4.7 são mostrados os componentes padrão existentes no kit Visual Web JSF (*Pallette*). Através de comandos do tipo clicar-arrastar, o usuário é capaz de montar sua aplicação e configurar a ação de cada um destes componentes, seja através da janela *Properties* ou seja através da guia Java. Deve-se ressaltar que, para esta última opção, a ação do componente deverá ser criada através de instruções, isto é, a partir de linhas de comando.

Na aplicação de exemplo mostrada nesta seção o usuário deve entrar com dois números reais quaisquer e, como resultado, serão mostrados a soma, o produto e a divisão destes números. Para a operação de divisão, entretanto, será feito um tratamento de erro com o denominador pois, caso seja nulo, uma mensagem de

erro será mostrada ao usuário.

As Figuras 4.8 e 4.9 ilustram, respectivamente, as interfaces de edição do IDE utilizando diversos componentes JSF e a tela de visualização do navegador, isto é, aquela com a qual o usuário irá interagir.

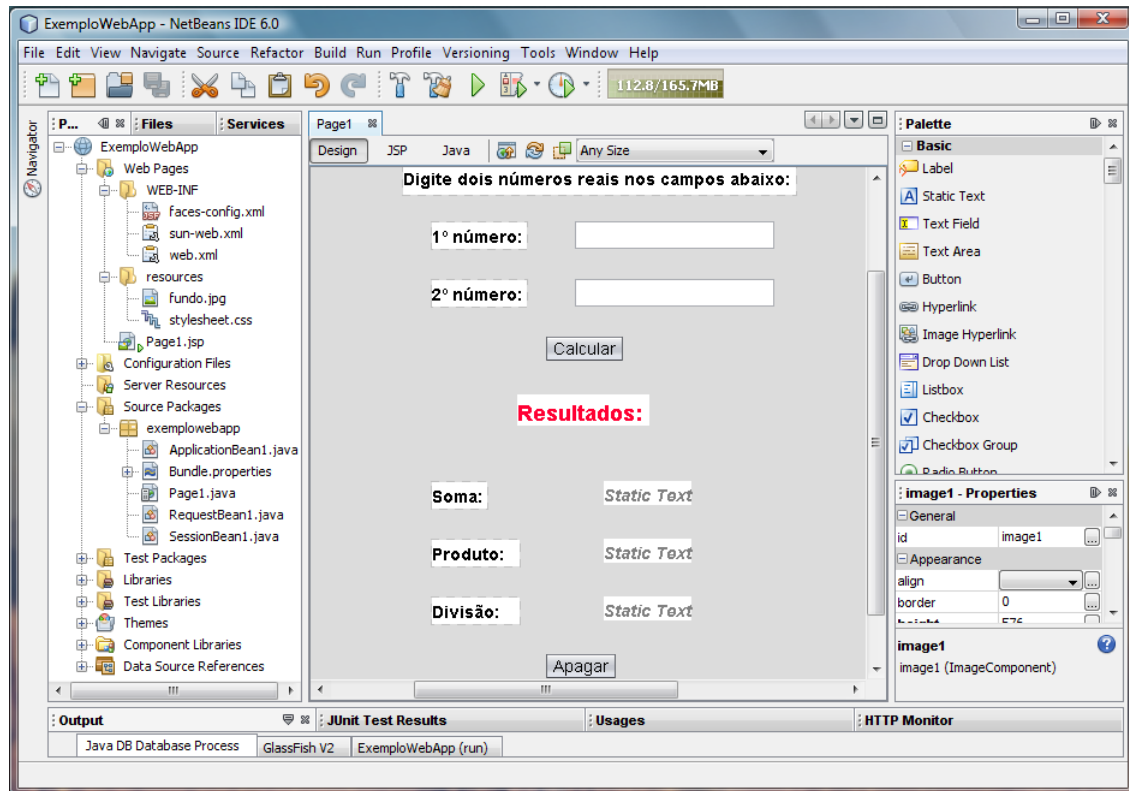


Figura 4.8: Tela de edição do NetBeans.

As instruções devem ser implementadas na aba “Java”. Neste caso, como a ação (somar, multiplicar e dividir) é feita após o clique no botão “Calcular”, o código será inserido no método referente a ele. Similarmente, para o botão “Apagar”, serão implementadas as instruções para limpar os campos utilizados.

As Figuras 4.10, 4.11 e 4.12 mostram duas simulações da aplicação e os códigos utilizados nos métodos “Calcular” e “Apagar”, respectivamente.

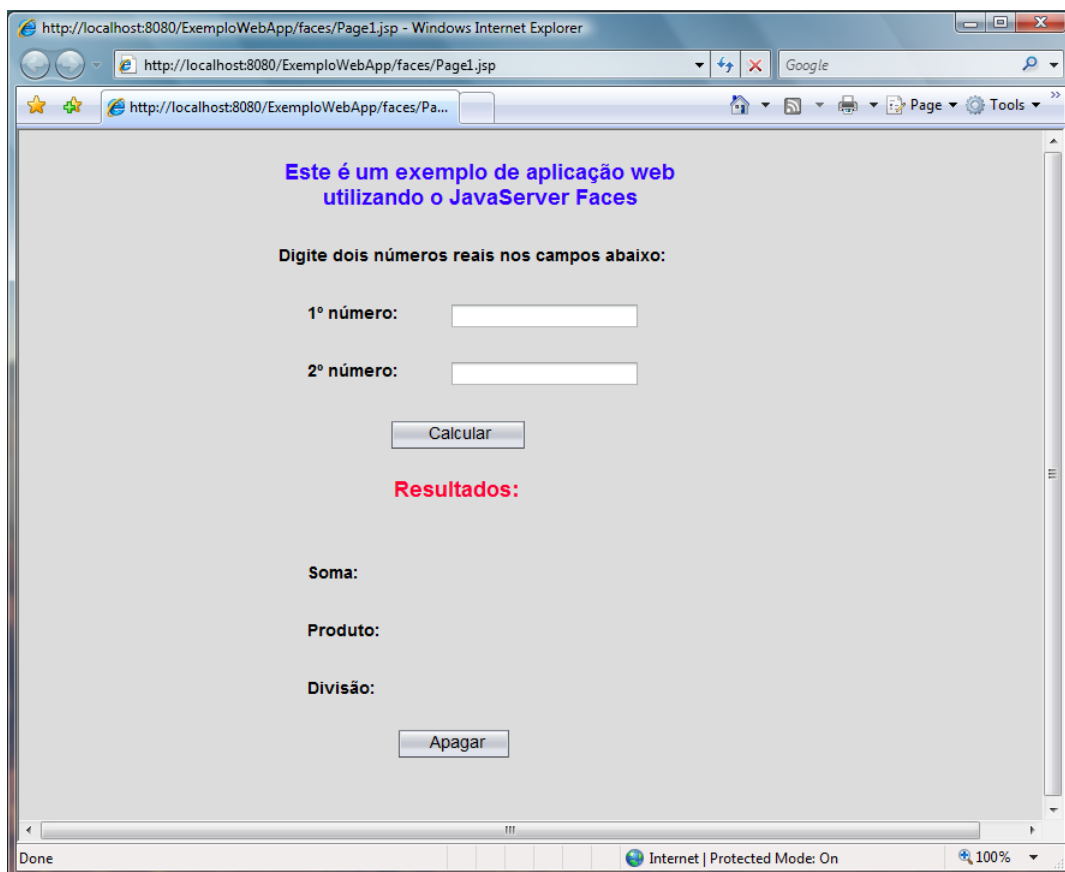


Figura 4.9: Tela de exibição do navegador.

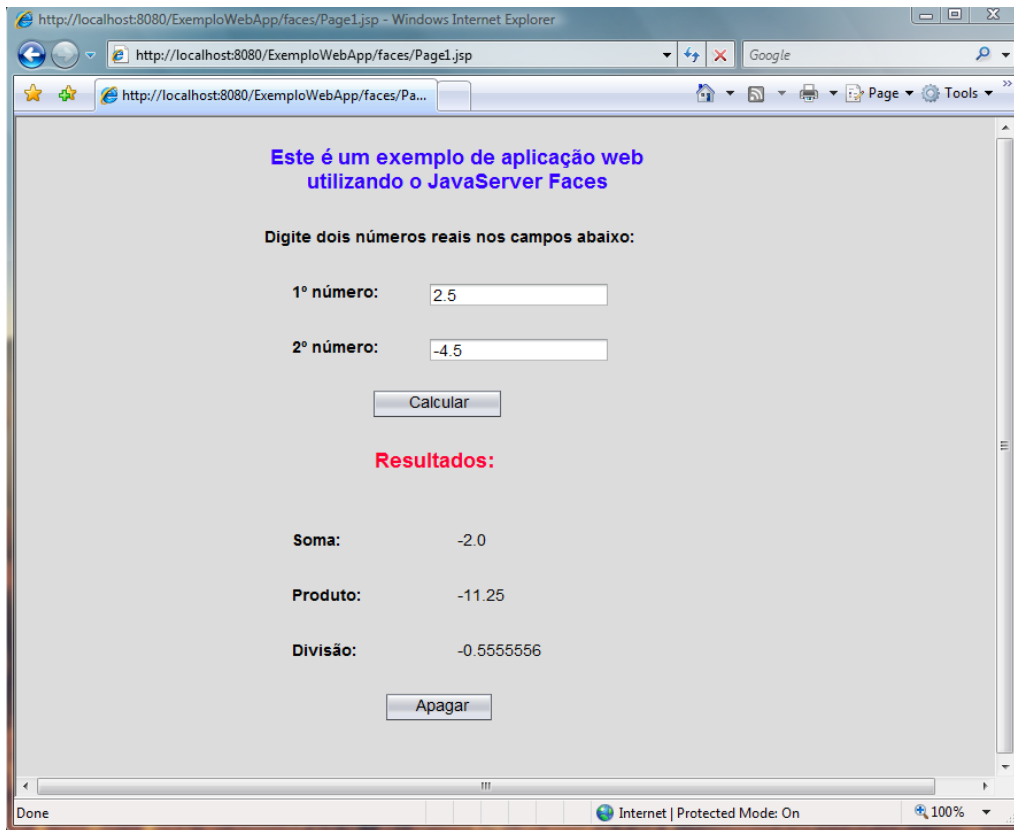


Figura 4.10: Primeira simulação do exemplo de aplicação web.

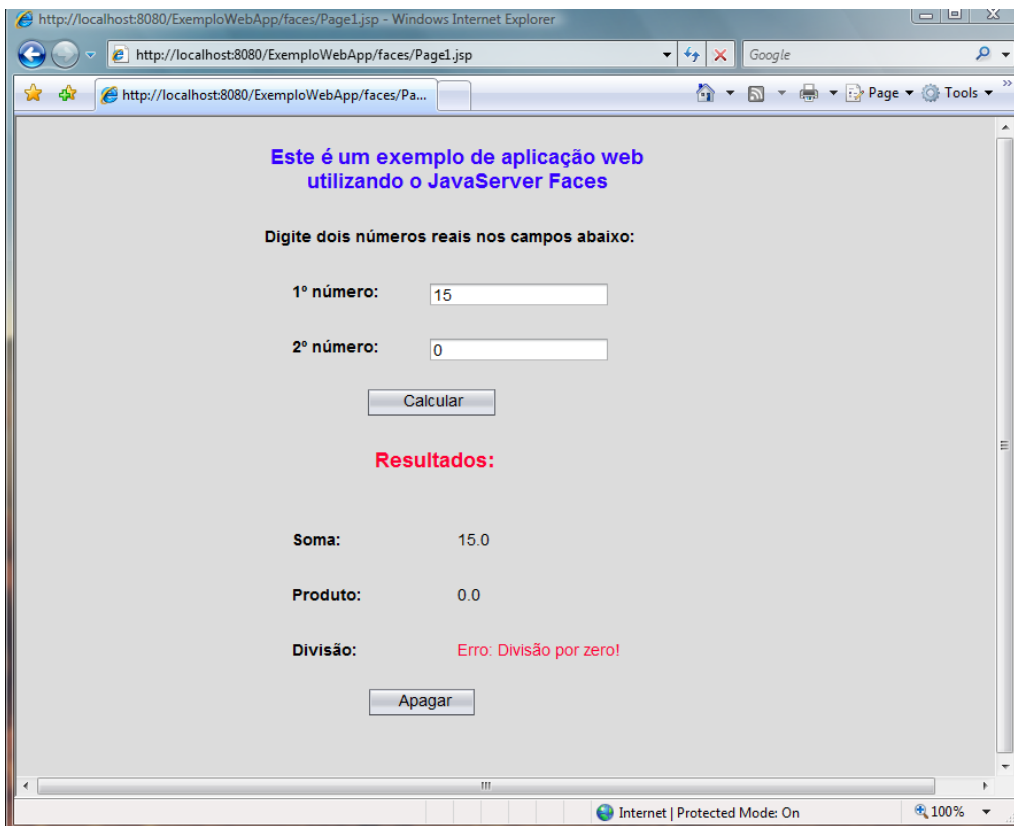


Figura 4.11: Segunda simulação do exemplo de aplicação web - Mensagem de erro.

```

public String Calcular () {

    float n1 = Float.parseFloat(textField1.getText().toString());
    float n2 = Float.parseFloat(textField2.getText().toString());

    float soma = n1 + n2;
    float prod = n1 * n2;

    float divi;

    if (n2==0){
        staticText3.setText("Erro: Divisão por zero!");
    }
    else{
        divi = n1/n2;
        staticText3.setText(divi);
    }

    staticText1.setText(soma);
    staticText2.setText(prod);

    return null;
}

public String Apagar () {

    textField1.setText("");
    textField2.setText("");
    staticText1.setText("");
    staticText2.setText("");
    staticText3.setText("");

    return "case1";
}

```

Figura 4.12: Código dos métodos dos botões *Calcular* e *Apagar* do exemplo de aplicação web.

Capítulo 5

A Aplicação Web MIDaS

5.1 Introdução

Atualmente (em busca realizada em Março de 2008), na área de identificação dinâmica de sistemas deformáveis, não existem aplicações web como a desenvolvida por este trabalho e que estejam disponíveis ao público para livre acesso. O que se observa, no entanto, é a existência de diversos algoritmos implementados em linguagens de programação distintas e que são restritos a solução de problemas específicos. Ou ainda, o uso de programas comerciais que resolvem apenas parte de um determinado problema.

Diante deste cenário, a aplicação MIDaS (Modal Identification of Deformable Systems) foi desenvolvida baseada nos algoritmos implementados em Matlab por Barbosa [40] com o objetivo de disponibilizar, através da internet, um conjunto de rotinas capazes de realizar a identificação modal de sistemas deformáveis sem a necessidade de instalação de qualquer tipo de programa (exceção: navegador web - *Firefox*, *Internet Explorer*, etc), conferindo à aplicação características de portabilidade. Por se tratar de uma aplicação web, isto significa que o usuário não precisa se preocupar em possuir um computador dotado de grande capacidade de processamento, uma vez que todos os cálculos serão realizados em um servidor e somente os resultados serão disponibilizados. Uma outra vantagem é o fato de permitir o compartilhamento de aplicativos em âmbito mundial.

A aplicação MIDaS foi totalmente desenvolvida utilizando linguagem Java.

Para a sua correta utilização, o usuário deverá carregar (através de um *upload*) três arquivos, sendo dois deles de definição de parâmetros - no formato *.txt* - e o outro contendo os arquivos de dados, isto é, os sinais resultantes de medições dinâmicas (contidos no formato compactado *.zip*). A descrição detalhada de cada um deles será feita nas seções seguintes.

O processo de identificação é dividido em dois passos. No primeiro, que é obrigatório, será executada a identificação dos parâmetros modais - frequências naturais, taxas de amortecimento e amplitudes das componentes dos modos de vibração. No segundo passo, estes parâmetros passarão por processos de truncamento, filtragem e, finalmente, serão organizados para um pós-processamento caso o usuário julgue necessário. Para cada passo existe um arquivo de parâmetros próprio, ao contrário do arquivo de dados, que é comum para ambos. Ao final de todas as análises, é possível visualizar os modos próprios de vibração, a partir das componentes identificadas.

Atualmente, a aplicação MIDaS está sendo executada em um servidor pertencente ao NUMEC (Núcleo de Pesquisa em Métodos Computacionais em Engenharia) da Universidade Federal de Juiz de Fora e pode ser acessada no endereço:

`www.midas.ufjf.br`

A Figura 5.1 mostra a página inicial da aplicação. Nesta página são descritas algumas funcionalidades do MIDaS, sendo possível fazer o *download* de um tutorial contendo informações detalhadas sobre como utilizar a aplicação, além de um exemplo completo para análise. Além disso, o usuário tem acesso às outras páginas através de *links* disponibilizados nas barras laterais.

A Figura 5.2 apresenta a página contendo breves informações a respeito do trabalho que concerne a aplicação. Nesta página pode-se fazer o *download* desta dissertação como mais uma fonte de consulta para o entendimento e a utilização do MIDaS.

A Figura 5.3 mostra a página criada para o cadastro de usuários (através do *link* "Register"). Para se ter acesso ao MIDaS é necessário preencher um formulá-

rio contendo além de informações pessoais (nome, profissão, instituição), algumas informações a respeito do interesse do usuário pela aplicação. Finalmente, deve-se escolher um nome de usuário (*login*) e uma senha para ter acesso à página principal. De posse de todas essas informações, o MIDaS gera um formulário no formato “.pdf” contendo os dados fornecidos e o disponibiliza para que o próprio usuário faça o *download*. A Figura 5.4 mostra um exemplo de um formulário gerado pela aplicação que permite ao desenvolvedor obter informações a respeito do tipo de usuário e interesse com relação à aplicação MIDaS.

Após cadastrado no MIDaS e de posse de um nome de usuário e senha, é possível utilizar a aplicação MIDaS fazendo um *login*, como mostra a Figura 5.5. Caso o usuário encontre problemas para acessar a aplicação, basta “clique” no *link* “Contact” na barra lateral direita e uma mensagem de correio eletrônico poderá ser preenchida e enviada ao autor.

A Figura 5.6 apresenta a página do primeiro passo da aplicação MIDaS: o MIDaS-Ident. Nesta página, pode-se fazer o *download* do tutorial mencionado anteriormente, além de um exemplo completo de um ensaio dinâmico. Caso o usuário já conheça a aplicação, procede-se com o *upload* dos arquivos de parâmetros e de dados referentes ao primeiro passo da análise.

Após o término da análise MIDaS-Ident, é disponibilizado ao usuário os arquivos contendo os primeiros resultados, como mostra a Figura 5.7. Nesta página o usuário pode, se julgar necessário, continuar com o segundo passo da aplicação MIDaS: MIDaS-Trunc, MIDaS-Filter, MIDaS-Assembly e MIDaS-Visual, através do *upload* dos arquivos de parâmetros e de dados referentes a estas análises.

Depois de terminado o segundo passo do processo de identificação, apresenta-se a página final da aplicação contendo os resultados de todas as análises realizadas, como mostra a Figura 5.8.

Finalmente, a Figura 5.9 apresenta um fluxograma contendo os principais procedimentos da aplicação web MIDaS.

Nas seções seguintes serão descritos em detalhes os algoritmos implementados

nesta aplicação.

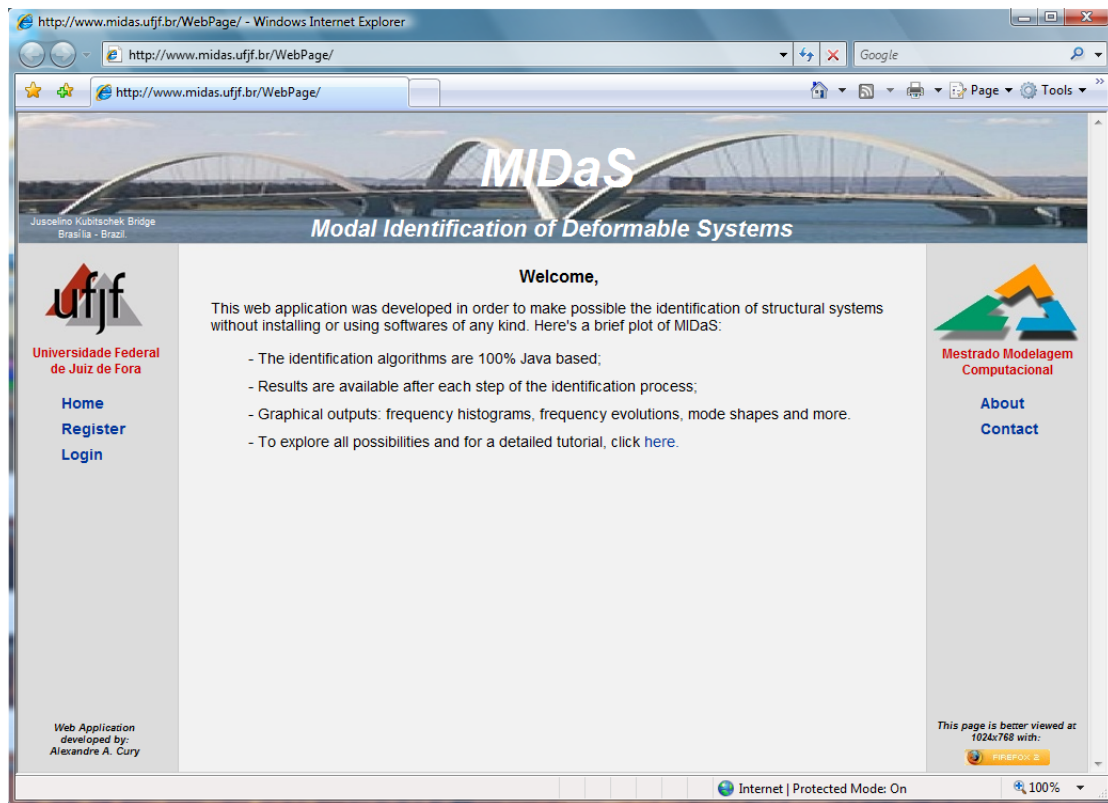


Figura 5.1: MIDaS - Página inicial.

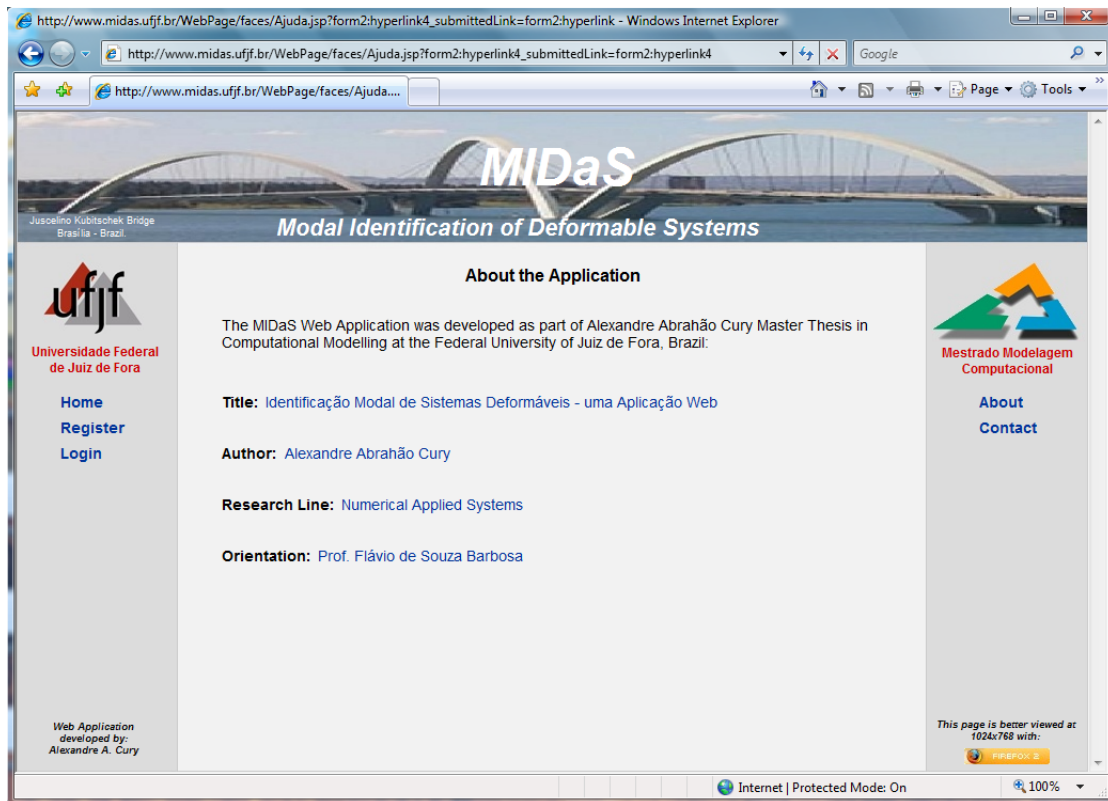


Figura 5.2: MIDA S - Sobre a aplicação.

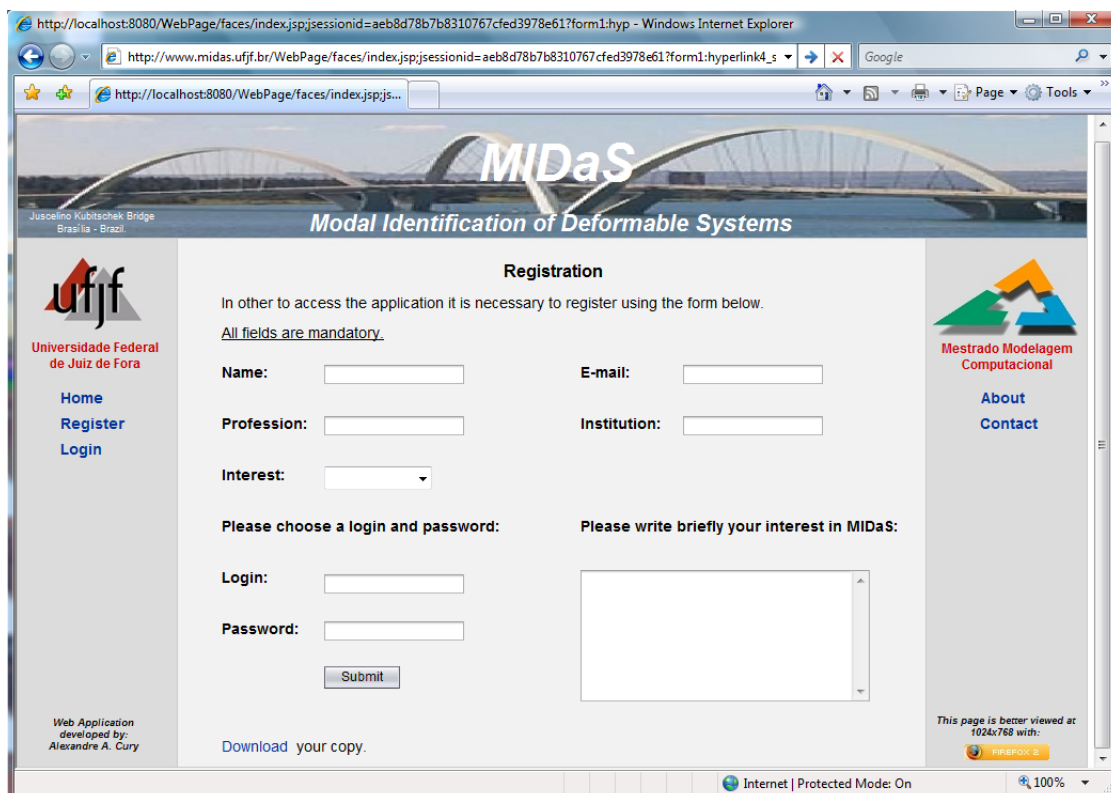


Figura 5.3: MIDA S - Página de cadastro de usuários.

MIDaS Registration Sheet

Name: Alexandre

E-mail: alexandre.cury@engenharia.ufjf.br

Profession: Estudante

Institution: UFJF

Interest: educational

Login: alexandre

Password: teste

Message: Escreva aqui o seu interesse pela aplicação MIDaS.

Date/Time sent: 16-Mar-08 15:45:29 BRT

Figura 5.4: Exemplo de um formulário de cadastro gerado pelo MIDaS.

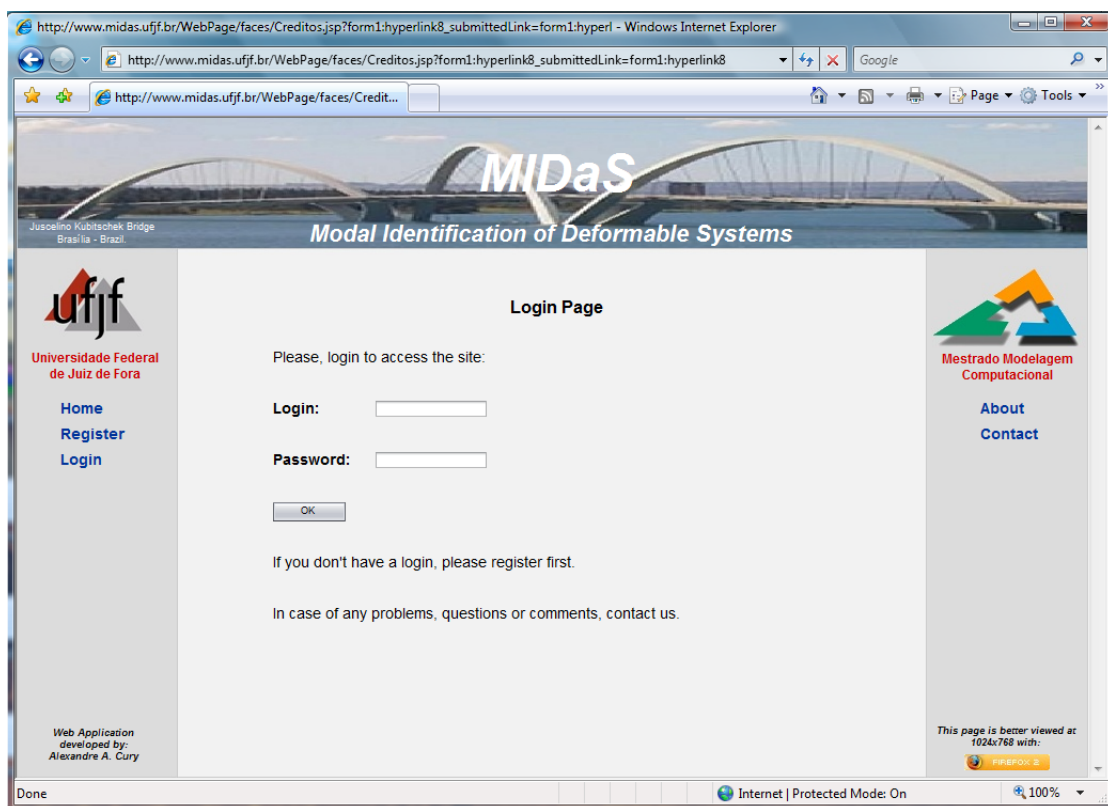


Figura 5.5: MIDaS - Página de *login*.

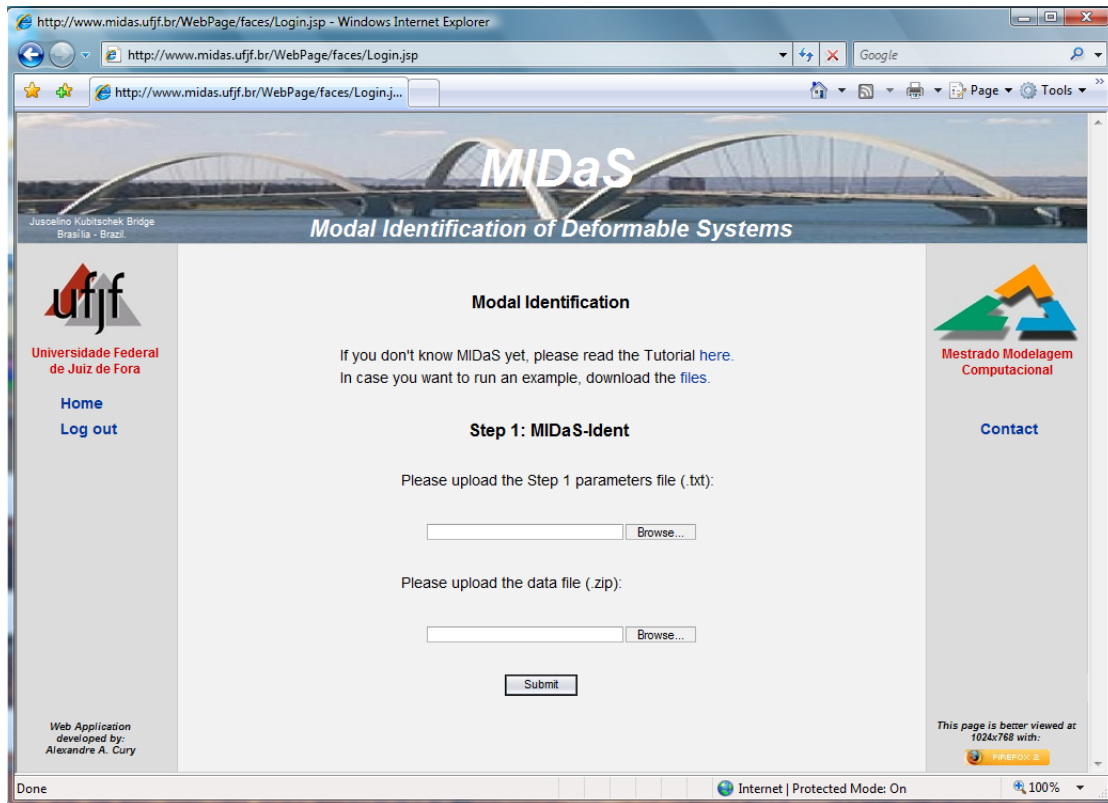


Figura 5.6: MIDaS - Análise MIDaS-Ident.



Figura 5.7: MIDaS - Resultados MIDaS-Ident / Análises MIDaS-Trunc, Filter, Assembly e Visual.

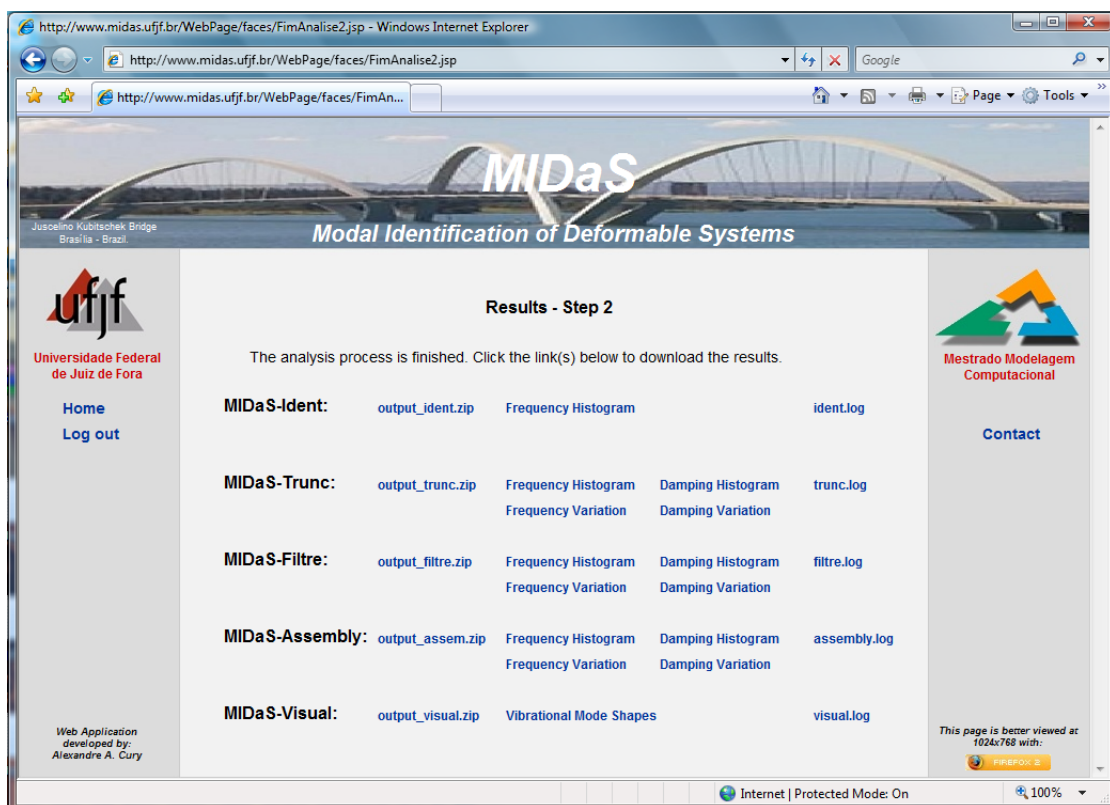


Figura 5.8: MIDaS - Página de resultados finais.

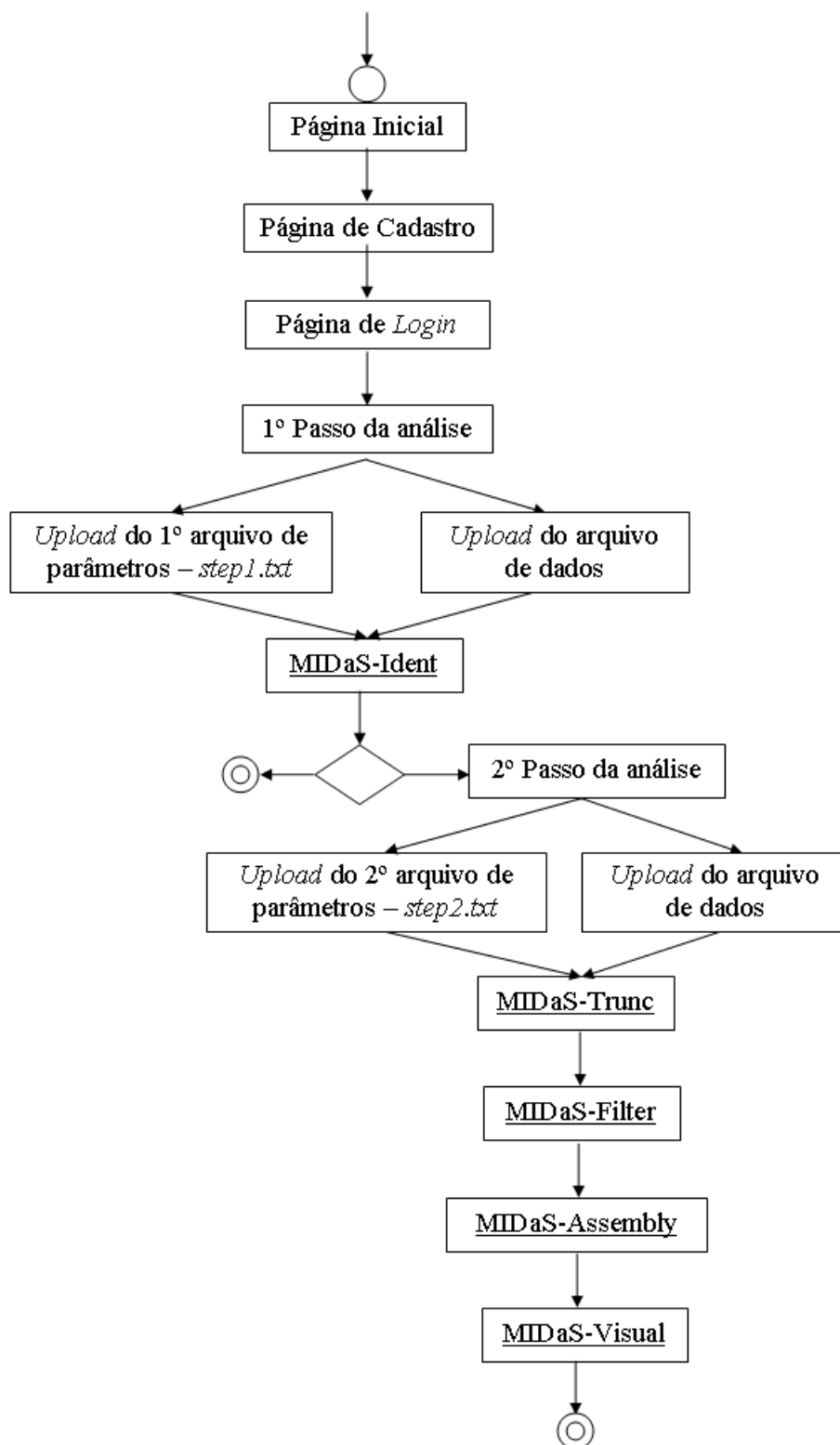


Figura 5.9: Fluxograma da aplicação web MIDaS.

5.2 Bibliotecas

Como descrito no capítulo anterior, as bibliotecas são um conjunto de classes disponibilizadas em formato .jar e que podem ser facilmente integradas aos códigos em processo de implementação.

Para o desenvolvimento deste trabalho e na implementação dos algoritmos foram utilizadas 4 bibliotecas gratuitas:

- A biblioteca JAMA - Java Matrix Package: (<http://math.nist.gov/javanumerics/jama>), foi desenvolvida pelo National Institute of Standards and Technology e pela Universidade of Maryland. Esta biblioteca é utilizada nos algoritmos de identificação para manipulação de matrizes (soma, multiplicação, fatoração, etc) e para a resolução do problema de autovalor (cálculo de autovalores e autovetores). Atualmente (Março 2008), a versão disponível desta biblioteca é a 1.0.2, possui código aberto e está em constante atualização.
- JMatIO - <http://sourceforge.net/projects/jmatio>, foi desenvolvida por Wojciech Gradkowski e outros. Ela é composta por um conjunto de classes que possibilitam, entre outros recursos, a leitura de arquivos gravados pelo programa Matlab no formato .mat. Ressalta-se que este programa é amplamente utilizado por engenheiros e o uso desta biblioteca é importante, uma vez que os resultados de ensaios dinâmicos são freqüentemente gravados neste formato. Com isso, a possibilidade de aceitar esse tipo de arquivo se torna um atrativo para usuários Matlab, uma vez que o MIDaS representará uma alternativa gratuita e eficiente. A versão atual (Março 2008) é 0.2 update 1 e não é de código-livre.
- A biblioteca JFreeChart - <http://www.jfree.org/>, fundada por David Gilbert e desenvolvida por sua equipe é utilizada para criação e visualização dos gráficos de saída referentes aos algoritmos de identificação. Dentre eles, são impressos histogramas de freqüência e amortecimento, modos de

vibração da estrutura, entre outros. Assim como as outras bibliotecas, esta também possui atualização constante e está na versão 1.0.9 (Março 2008). Não possui código aberto.

- Finalmente, a biblioteca iText - <http://www.lowagie.com/iText>, desenvolvida por Bruno Lowagie e Paulo Soares. As classes do iText são utilizadas para criação de arquivos no formato .pdf, como na geração de formulários de cadastro de usuários da aplicação desenvolvida. A versão atual é 2.0.8 e possui código aberto.

5.3 Algoritmos Implementados

A aplicação MIDaS é constituída de 5 algoritmos, assim denominados: MIDaS-Ident, MIDaS-Trunc, MIDaS-Filter, MIDaS-Assembly e MIDaS-Visual. Cada um deles possui parâmetros de entrada específicos e arquivos de saída disponibilizados separadamente. A seguir, descreve-se cada um destes algoritmos.

5.3.1 MIDaS-Ident

Esta é a principal rotina da aplicação MIDaS e deve ser obrigatoriamente utilizada.

O algoritmo MIDaS-Ident é capaz de realizar a identificação modal de sistemas no domínio do tempo, utilizando os métodos do Decremento Aleatório e de Ibrahim, descritos do Capítulo 3. Neste caso, os arquivos contendo os sinais provenientes das medições dinâmicas (acelerações, deslocamentos, deformações, etc.) são analisados simultaneamente, a partir dos arquivos de dados fornecidos pelo usuário.

- *Arquivos de entrada:*

Os parâmetros de entrada deste algoritmo estão contidos em um arquivo específico, denominado “step1.txt”. Além deste, existe o arquivo de dados (compactado no formato .zip) que também deverá ser submetido para iniciar a análise. A descrição e a formatação desses arquivos serão discutidas na seção 5.4.

- *Arquivos de saída:*

Os resultados da identificação (frequências naturais, taxas de amortecimento e amplitudes das componentes dos modos de vibração) são gravados em arquivos de texto e disponibilizados para *download* no formato compactado, denominado “output_ident.zip”. Com estes resultados, o usuário pode utilizá-los em um programa externo ou, se preferir, continuar a análise com os outros algoritmos da aplicação MIDaS.

Além deste arquivo, é possível visualizar um histograma com as frequências identificadas. Este histograma é importante para a continuação do processo de identificação porque é através de sua análise que o usuário irá definir as frequências a serem efetivamente analisadas.

Finalmente, disponibiliza-se também um arquivo contendo informações importantes acerca da análise efetuada, denominado arquivo de *log*. Neste arquivo, gravado no formato texto, podem ser consultados a ocorrência ou não de erros durante o processo, bem como o tempo decorrido para a transferência de arquivos (do *upload* dos arquivos de parâmetros e de dados), o tempo de cálculo efetivo de identificação e o tempo de escrita dos dados de saída e para a impressão dos gráficos.

5.3.2 MIDaS-Trunc

Esta é a segunda rotina da aplicação MIDaS a ser executada.

Após a análise do histograma de frequências obtido como resultado da rotina Ident, o usuário é capaz de definir o número de frequências que continuarão na análise e a banda de truncamento de cada uma delas, isto é, a definição dos limites inferior e superior de corte. Como consequência, haverá também o truncamento das respectivas taxas de amortecimento e dos modos próprios de vibração.

- *Arquivos de entrada:*

Os parâmetros iniciais deste algoritmo estão contidos no segundo arquivo de entrada, denominado “step2.txt”. Além deste, existe o arquivo de dados (compac-

tado no formato .zip) que também deverá ser submetido para iniciar a análise. A descrição do arquivo “step2.txt” também será discutida na seção seguinte.

- *Arquivos de saída:*

Os resultados deste algoritmo são as frequências naturais, taxas de amortecimento e amplitudes das componentes dos modos de vibração truncados. Tem-se ainda, alguns vetores de ponteiros que são utilizados internamente nas próximas rotinas. Estes resultados também são gravados em arquivos de texto e disponibilizados para *download* no formato compactado, denominado “output_trunc.zip”. Com estes resultados, o usuário pode utilizá-los em um programa externo ou, se preferir, continuar a análise com os outros algoritmos da aplicação MIDaS.

Além deste arquivo é possível visualizar um histograma com as frequências e taxas de amortecimento truncadas, além de gráficos com a evolução destes parâmetros modais.

Da mesma forma que na rotina anterior, disponibiliza-se um arquivo de *log* com informações sobre a ocorrência ou não de erros durante o processo de truncamento, os tempos decorridos em cada etapa e tabelas contendo as médias e desvios-padrões das frequências e amortecimentos truncados. A Figura 5.10 mostra um típico arquivo de *log* da aplicação MIDaS.

5.3.3 MIDaS-Filter

Esta é a terceira rotina da aplicação MIDaS a ser executada.

Após o truncamento, esta rotina tem o objetivo de realizar o processo de filtragem dos parâmetros modais, de forma a definir o intervalo de confiança das frequências naturais ou das taxas de amortecimento.

A opção do tipo de filtragem e o número de desvios-padrões a ser utilizado para a determinação do intervalo de confiança dos dados filtrados, são definidos pelo próprio usuário através do segundo arquivo de entrada, “step2.txt”.

Os intervalos de confiança são usados para indicar a confiabilidade de uma estimativa. A título de ilustração, seja um conjunto amostral de frequências natu-

```

log_trunc - WordPad
File Edit View Insert Format Help
Universidade Federal de Juiz de Fora
Programa de Mestrado em Modelagem Computacional
Web Application developed by Alexandre A. Cury

MIDA-S - Modal Identification of Deformable Systems

MIDA-S-Trunc analysis was finished successfully.

No errors occurred.

Number of frequencies: 2

Mean values (Hz):

    6.7891
    26.136

Standard Deviations (Hz):

    0.0311
    0.0880

Damping Ratios: 2

Mean Values (%):

    0.0158
    0.0129

Standard Deviations (%):

    0.0051
    0.0146

Total time to transfer data: 0 second
Total time of the analysis: 1 second
Total time of file manipulation: 3 seconds

Finished at: Sun Mar 16 10:22:38 BRT 2008

Thank you for using MIDA-S.

For Help, press F1

```

Figura 5.10: Exemplo de arquivos de *log* da aplicação web MIDA-S.

rais identificadas, com média $\bar{f} = 3,90\text{Hz}$ e desvio-padrão igual a $\sigma_f = 0,025\text{Hz}$. Considerando uma distribuição de probabilidade próxima à Normal (Gaussiana), define-se a região que contém aproximadamente 95% dessas amostras como sendo:

$$\text{IC} [\bar{f}; 95\%] = [\bar{f} - 2 \times \sigma_f; \bar{f} + 2 \times \sigma_f] = [3,85; 3,95],$$

onde IC representa o intervalo de confiança para a média \bar{f} e o valor $2 \times \sigma_f$ se refere à probabilidade de aproximadamente 95% das amostras pertencerem ao intervalo

de confiança.

A Figura 5.11 mostra um histograma de freqüências que ilustra este exemplo.

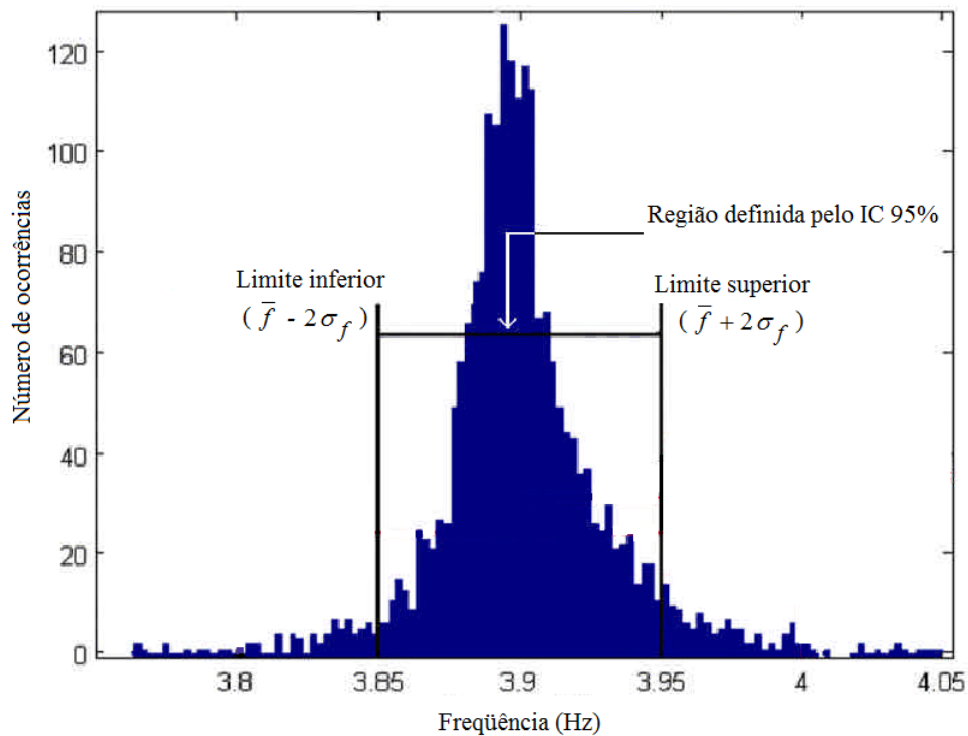


Figura 5.11: Exemplo de intervalo de confiança para uma amostra de freqüências.

- *Arquivos de entrada:*

Os arquivos de entrada são os mesmos utilizados na rotina MIDaS - Trunc.

- *Arquivos de saída:*

Os resultados deste algoritmo são as freqüências naturais, taxas de amortecimento e amplitudes das componentes dos modos de vibração filtrados, além de alguns vetores de ponteiros necessários nas próximas rotinas.

Estes resultados são gravados no arquivo “output_filter.zip”, sendo possível visualizar um histograma com as freqüências e taxas de amortecimento truncadas, além de gráficos com a evolução destes parâmetros modais. Similarmente às rotinas anteriores, disponibiliza-se um arquivo de *log* contendo, dentre outras informações, as médias e desvios-padrões das freqüências e amortecimentos filtrados.

5.3.4 MIDaS-Assembly

Esta é a quarta rotina da aplicação MIDaS a ser executada.

Esta rotina tem o objetivo de organizar as frequências naturais e as taxas de amortecimento após o processo de filtragem. Além disso, os modos próprios são organizados utilizando-se a matriz de ponteiros fornecida pelo usuário no arquivo de parâmetros “step2.txt”.

Após esta rotina, considera-se que o processo de identificação modal está totalmente completo.

- *Arquivos de entrada:*

Os arquivos de entrada são os mesmos utilizados na rotina MIDaS - Trunc.

- *Arquivos de saída:*

Os resultados deste algoritmo são gravados no arquivo “output_assembly.zip”, contendo as frequências naturais, taxas de amortecimento e modos de vibração após a organização dos dados.

Similarmente às análises anteriores, disponibiliza-se, além das saídas gráficas o arquivo de *log* com informações a respeito do processo.

5.3.5 MIDaS-Visual

Esta é a quinta e última rotina da aplicação MIDaS a ser executada.

O objetivo desta rotina é preparar e gerar os gráficos com os modos próprios de vibração obtidos do algoritmo anterior - MIDaS-Assembly - para sua visualização, isto é, para a saída gráfica.

- *Arquivos de entrada:*

Os arquivos de entrada são os mesmos utilizados na rotina MIDaS - Trunc.

- *Arquivos de saída:*

Os resultados deste algoritmo são os modos de vibração prontos para impressão, gravados no arquivo “output_visual.zip”. Com estes resultados, o usuário também pode traçar os modos em um programa externo, se preferir.

Também é disponibilizado o arquivo de *log* no formato de texto.

5.4 Arquivo de Parâmetros

Como mencionado nas seções anteriores, para a utilização da aplicação MIDaS o usuário deve submeter dois arquivos de parâmetros específicos para cada passo de análise.

Para o primeiro passo, referente ao MIDaS-Ident, é necessário o arquivo “step1.txt” e, para o segundo passo, referente às análises MIDaS-Trunc, Filter, Assembly e Visual, o usuário deve carregar o arquivo “step2.txt”.

Para ambos os arquivos é muito importante que a formatação seja obedecida, pois, caso contrário, ocorrerá erro de execução. A formatação deverá obedecer, portanto, às seguintes instruções:

- O espaçamento entre os números da mesma linha deve ser de uma tabulação simples;
- Não deve existir nenhuma linha vazia entre os parâmetros;
- O números decimais devem ser representados por pontos (.) e não vírgulas.
Ex.: 0.25.

5.4.1 *step1.txt*

Este arquivo é referente à análise MIDaS-Ident e contém os seguintes parâmetros:

- a) **Vfi** - Arquivos que serão utilizados na análise. Os arquivos são lidos de acordo com a ordem alfabética dos computadores¹. Exemplo: de 5 arquivos submetidos no arquivo de dados (compactado no formato “dados.zip”), deseja-se analisar o 1º, 3º e 4º arquivos:

Vfi

1 3 4

¹ Um exemplo de ordenação de números no computador: sejam os arquivos step1.txt, 2.m, 10a.txt, 22b.dat, 3c.m e dado.zip. A ordem será 10a.txt, 22b.dat, 2.m, 3c.m, dado.zip, step1.txt

b) **ext** - Extensão dos arquivos de dados que podem ser utilizados:

ext = 1 → extensão .mat (arquivos Matlab)

ext = 2 → extensão .dat (arquivos texto)

c) **tfic** - Tipo dos arquivos de dados. Pode-se escolher entre 6 tipos:

tfic = 1 → dados do tipo texto prontos. Este caso se refere às situações em que o sinal adquirido possui média igual a zero para todas as vias de medida.

tfic = 2 → dados do tipo texto em que é necessária a translação do sinal em relação à média. Este caso se aplica para sinais que possuem média diferente de zero e ocorre em situações em que existem ruídos com valor constante em todo o sinal.

tfic = 3 → dados do tipo texto em que é necessária a translação do sinal em relação à média e também é necessário fazer o sinal rotacionar em relação ao eixo das abscissas. Este caso ocorre quando da presença de ruído linear.

tfic = 4 → dados do tipo .mat (idem ao caso 1).

tfic = 5 → dados do tipo .mat (idem ao caso 2).

tfic = 6 → dados do tipo .mat (idem ao caso 3).

d) **MV_m** - Tipos de medidas para a análise. Entre os arquivos escolhidos em

V_{fi} com a extensão **ext**, escolhe-se as medições que farão parte da análise.

Esta variável é uma matriz que possui o número de linhas igual ao número de arquivos lidos e que contém os tipos de medições úteis a cada arquivo.

Existem 2 particularidades a respeito das 2 primeiras colunas:

- 1^a coluna: Referência para normalização dos modos próprios.

- 2^a coluna: Base para o cálculo das funções de correlação. Esta coluna indica a série de medidas para qual serão definidas as condições de desencadeamento do método do Decremento Aleatório. Os modos são formados sem esta medida.

Exemplo: $\mathbf{MVm} = \begin{bmatrix} 2 & |2| & 1 & 5 \\ 2 & |3| & 6 & 8 \\ 2 & |4| & 7 & 5 \end{bmatrix}$. A coluna entre “|” não participa da formação dos modos.

e) **ref** - As medições utilizadas para a normalização dos modos próprios (1ª coluna da matriz \mathbf{MVm}) podem ou não fazer parte dos modos de vibração que se desejam obter.

ref = 1 → 1ª coluna faz parte do vetor próprio.

ref = 2 → 1ª coluna não faz parte do vetor próprio.

f) **varint** - Variáveis de identificação inteiras. O vetor **varint** consiste em três parâmetros: *nf*, *deltati* e *deltatf*. Onde:

nf → Número de frequências próprias desejadas. Geralmente, nos problemas de engenharia, estes valores são definidos entre 5 e 15.

deltati → Parâmetro de valor mínimo para as defasagens das pseudo-medidas do Método de Ibrahim. (deve ser > 1).

deltatf → Parâmetro de valor máximo para as defasagens das pseudo-medidas do Método de Ibrahim. (geralmente < 50).

Entrada:

varint

nf *deltati* *deltatf*

g) **varreal** - Variáveis de identificação reais. O vetor **varreal** consiste em quatro parâmetros: *pnpf*, *p1*, *p2* e *deltat*. Onde:

pnpf → Razão entre o número de pontos da janela de correlação e o número total de pontos do sinal. Geralmente o valor de pontos a serem utilizados para o cálculo das correlações é definido entre 500 e 1000, pois valores superiores acarretam perdas de desempenho nos cálculos e valores inferiores causam imprecisões no processo de identificação.

p1 e *p2* → Parâmetros para determinação dos pontos de desencadeamento (*a1* e *a2*) do método do Decremento Aleatório descritos na Tabela 3.1.

- Passagem de nível: $a1 = p1 \times \sigma_X$; $a2 = 0$;
- Extremo Local: $a1 = p1 \times \sigma_X$; $a2 = p2 \times \sigma_X$;
- Passagem de Zero: $a1 = 0$; $a2 = 0$;
- Pontos Positivos: $a1 = p1 \times \sigma_X$; $a2 = p2 \times \sigma_X$;

onde σ_X representa o desvio-padrão das medidas.

deltat → Tempo de amostragem dos sinais.

Entrada:

varreal

pnpf p1 p2 deltat

h) flag1 - Tipo de condições de desencadeamento do método do Decremento Aleatório.

flag1 = 1 → Passagem de Nível.

flag1 = 2 → Extremo Local.

flag1 = 3 → Passagem de Zero.

flag1 = 4 → Ponto Positivo.

i) flag2 - Tipo de pseudo-medidas do Método de Ibrahim.

flag2 = 1 → Divisão dos Sinais.

flag2 = 2 → Defasagem dos Sinais.

j) nd - Variável para aumentar artificialmente, se necessário, o número de pontos dos sinais medidos através de interpolação linear.

nd = 1 → Sem alteração.

nd = 2 → Dobro do número de pontos.

nd = 3 → Triplo do número de pontos.

nd = 4 → quádruplo do número de pontos.

nd = 6 → Sêxtuplo do número de pontos.

k) limxi - Variável utilizada para determinar a faixa de eliminação de frequências espúrias. De uma forma geral, as frequências associadas às taxas de amortecimento elevadas não são relativas à estrutura. Um valor razoável

para o limite superior da taxa de amortecimento é de 10% (0,01).

limxi

limite_inferior *limite_superior*

5.4.2 *step2.txt*

Este arquivo é referente às análises MIDaS-Trunc, Filter, Assembly e Visual e, para tal, alguns parâmetros devem ser definidos:

Utilizados na rotina MIDaS-Trunc:

a) **limfreq** - Definição dos intervalos de limite para cada frequência identificada pelo MIDaS-Ident. Com isso, previne-se cálculos com frequências indesejadas. Ex.: Considerando-se 2 frequências,

limfreq

freq1_inf *freq2_inf*

freq1_sup *freq2_sup*

Utilizados na rotina MIDaS-Filter:

a) **tipofilt** - Este vetor consiste em dois parâmetros: *tf* e *ic*. Onde:

tf → Tipo de filtragem:

tf = 1 → Filtragem via frequências naturais.

tf = 2 → Filtragem via taxas de amortecimento.

ic → Valor utilizado para a determinação do intervalo de confiança dos parâmetros modais filtrados. Caso *ic* = 2, o considera-se que aproximadamente 95% dos parâmetros filtrados esteja no intervalo de confiança definido.

Entrada:

tipofilt

tf *dp*

Utilizados na rotina MIDaS-Assembly:

a) **td** - Vetor de descrição do arquivo. Este vetor possui o número de posições igual ao número de arquivos de análise, sendo que cada número i é relativo à i -ésima série de medidas analisada.

Exemplo: **td** = [1 2 1 2 2 3]

1 → 1^o ensaio da 1^a série de medidas.

2 → 1^o ensaio da 2^a série de medidas.

1 → 2^o ensaio da 1^a série de medidas.

2 → 2^o ensaio da 2^a série de medidas.

2 → 3^o ensaio da 2^a série de medidas.

3 → 1^o ensaio da 3^a série de medidas.

Utilizados na rotina MIDaS-Visual:

a) **visu** - Variáveis de visualização inteiras. Este vetor consiste em dois parâmetros: dim e $npontos$. Onde:

dim → Número de modos próprios a serem visualizados.

$npontos$ → Número de pontos observados do problema (pontos em que as medições são feitas) + Número de pontos suplementares para traçar o modo próprio (caso seja necessário definir pontos com condições de contorno - deslocamento zero - nos extremos da estrutura). Entrada:

visu

dim $npontos$

b) **ap** - Matriz de ponteiros. As dimensões desta variável são: Número de séries de medidas × Número máximo de sinais por série de medida. Os elementos desta matriz indicam a posição e a direção das medidas para o modelo de visualização.

Exemplo: Para uma matriz **MV_m** e um vetor **td**:

$$\begin{bmatrix} 2 & |2| & 1 & 5 & 6 & 7 & 8 & 11 \\ 2 & |3| & 6 & 8 & 9 & 12 & 10 & 0 \\ 2 & |4| & 7 & 5 & 6 & 3 & 11 & 0 \\ 2 & |4| & 7 & 5 & 6 & 3 & 0 & 0 \\ 2 & |4| & 7 & 5 & 6 & 3 & 0 & 0 \\ 2 & |4| & 7 & 5 & 6 & 3 & 11 & 0 \\ 2 & |4| & 7 & 5 & 6 & 3 & 11 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 4 \\ 3 \\ 3 \end{bmatrix} \begin{matrix} \text{(série 1)} \\ \text{(série 2)} \\ \text{(série 3)} \\ \text{(série 4)} \\ \text{(série 4)} \\ \text{(série 3)} \\ \text{(série 3)} \end{matrix}$$

e considerando a variável $\mathbf{ref} = 1$, isto é, a coluna 1 faz parte do vetor próprio, a matriz \mathbf{ap} será:

$$\mathbf{ap} = \begin{bmatrix} 1.1 & 1.2 & 1.3 & 2.1 & 8.2 & 2.3 & 3.1 \\ 0 & 4.1 & 5.2 & 4.3 & 5.1 & 4.2 & 0 \\ 0 & 6.1 & 6.3 & 7.3 & 8.1 & 2.2 & 0 \\ 0 & 9.1 & 9.2 & 9.3 & 10.1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix} \begin{matrix} \text{(série 1)} \\ \text{(série 2)} \\ \text{(série 3)} \\ \text{(série 4)} \end{matrix}$$

- Para deslocamentos na direção X, deve-se usar o dígito 1 após o “.”;
- Para deslocamentos na direção Y, deve-se usar o dígito 2 após o “.”;
- Para deslocamentos na direção Z, deve-se usar o dígito 3 após o “.”.

De forma geral, $N.n$ significa que existe um deslocamento N na direção n .

5.5 Arquivo de dados

Como descrito anteriormente, os arquivos de dados devem ser carregados no formato compactado .zip. Este arquivos, no formato descompactado, podem possuir as extensões .mat (Matlab) ou .dat (texto).

Assim como os arquivos de parâmetros, os números devem conter espaçamento de uma tabulação simples e os decimais devem ser representados por ponto (.) e não vírgulas. A título de ilustração, seja um sinal com 10 pontos de medição e 3 vias de medidas:

```
4.7975  203.1235  28.3171
3.8491  194.5166  27.0300
```

2.8450	185.9097	24.0267
2.1756	176.8724	22.7395
1.3946	168.6958	21.4524
0.3905	158.7978	19.3071
-0.3347	150.6213	17.1619
-1.1715	143.3054	14.1586
-1.9525	133.8377	12.0133
-2.6219	123.9398	10.7262
-3.3471	116.6239	8.1519

Ressalta-se, novamente, que no site da aplicação (www.midas.ufjf.br) o usuário poderá fazer o *download* de um tutorial contendo informações detalhadas sobre como utilizar o MIDaS, além de um exemplo completo para análise. Este exemplo, aliás, será o primeiro a ser exposto no capítulo a seguir, contendo os resultados.

Capítulo 6

Exemplos de Aplicação

O presente capítulo tem o objetivo de mostrar os resultados obtidos de alguns exemplos de estruturas ensaiadas, utilizando a aplicação web MIDaS. É importante destacar que o principal intuito dos exemplos de aplicação não é o de esgotar todas as possíveis análises sobre as características dinâmicas das estruturas avaliadas, mas sim o de mostrar, de forma geral, as potencialidades da aplicação e o modo como a mesma deve ser utilizada.

O primeiro exemplo se refere a uma viga de aço bi-apoiada, instrumentada e ensaiada pelo autor no Laboratório de Resistência dos Materiais da Universidade Federal de Juiz de Fora.

Em seguida, apresentam-se os resultados de um ensaio dinâmico feito sobre um viaduto por onde passam trens de alta velocidade TGV, na França. Os dados deste ensaio foram gentilmente cedidos pelo pesquisador Christian Cremona, do LCPC (Laboratoire Central des Ponts et Chaussées), de Paris, dentro de um acordo de cooperação firmado entre a Universidade Federal de Juiz de Fora e o LCPC.

O terceiro e último exemplo de aplicação refere-se ao ensaio realizado sobre a ponte Z24 que liga as cidades de Koppigen e Utzenstorf, na Suíça, cujos dados também foram cedidos pelo LCPC.

6.1 Viga bi-apoiada

O primeiro exemplo de aplicação do MIDaS é uma viga de aço bi-apoiada instrumentada e ensada no Laboratório de Resistência dos Materiais da Universidade Federal de Juiz de Fora, como mostra a Figura 6.1.

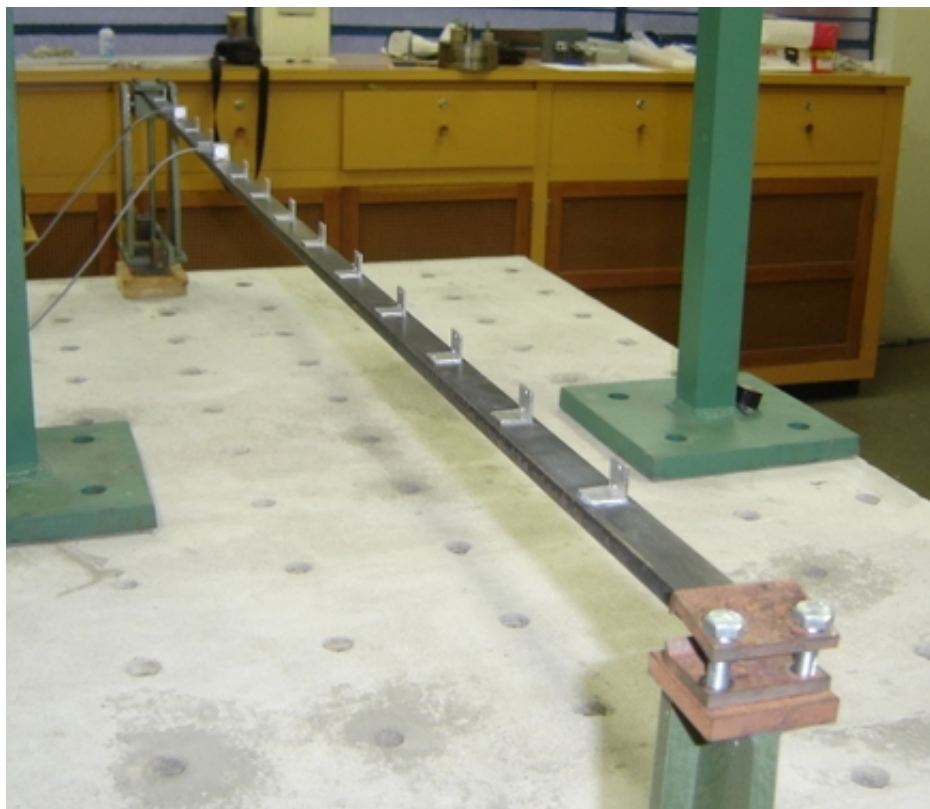


Figura 6.1: Viga de aço instrumentada com 2 acelerômetros.

As principais características físicas e geométricas desta viga são descritas abaixo:

- Dimensões: 1,80 x 0,0254 x 0,009 m
- Massa específica: 7.850 Kg/m³
- Módulo de elasticidade: 210 GPa.

A viga foi instrumentada utilizando dois acelerômetros da marca Kyowa, ambos do modelo AS-2GA e capazes de medir acelerações entre $-19,61$ a $19,61$ m/s² para aquisição dos sinais da estrutura. Um deles permaneceu fixo na posição

número 3 (de referência), enquanto o outro alternou entre as posições de número 1 a 15, com exceção à posição 3. Além disso, todas as posições são igualmente espaçadas, como indica o esquema de instrumentação da Figura 6.2. Para cada posição do acelerômetro “móvel” eram realizadas 5 aquisições com o objetivo de atenuar o efeito de fatores externos sobre o resultado final da identificação. A Figura 6.2 também apresenta o ponto de carregamento da estrutura, excitada através de impactos com as mãos.

Para a aquisição dos sinais da estrutura utilizou-se uma taxa de aquisição de 500Hz e um filtro passa-baixa em 200Hz. O equipamento utilizado foi desenvolvido pela empresa Lynx, modelo ADS2000, juntamente com o programa AqDados disponibilizado pela mesma.

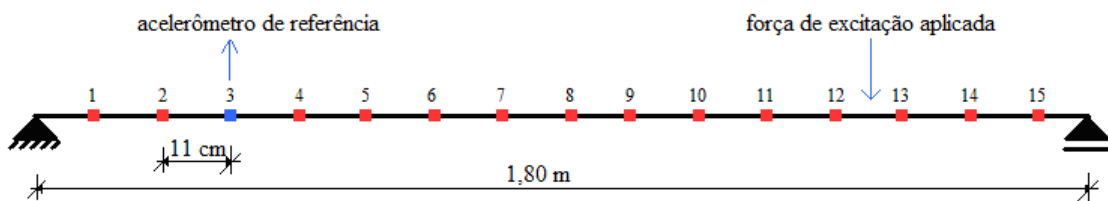


Figura 6.2: Esquema de instrumentação da viga de aço com 15 posições.

A Figura 6.3 apresenta as acelerações medidas para os pontos 3 e 1 obtidas durante os ensaios experimentais.

Os sinais aquisitados da estrutura (medições dinâmicas) foram gravados em arquivos do tipo .mat, gerando, ao final, 70 arquivos - 14 posições possíveis \times 5 repetições.

Os parâmetros utilizados para esta análise são descritos a seguir:

Utilizados no MIDaS-Ident:

a) $\mathbf{Vfi} = [1:70]$

b) $\mathbf{ext} = 1$

c) $\mathbf{tfic} = 5$

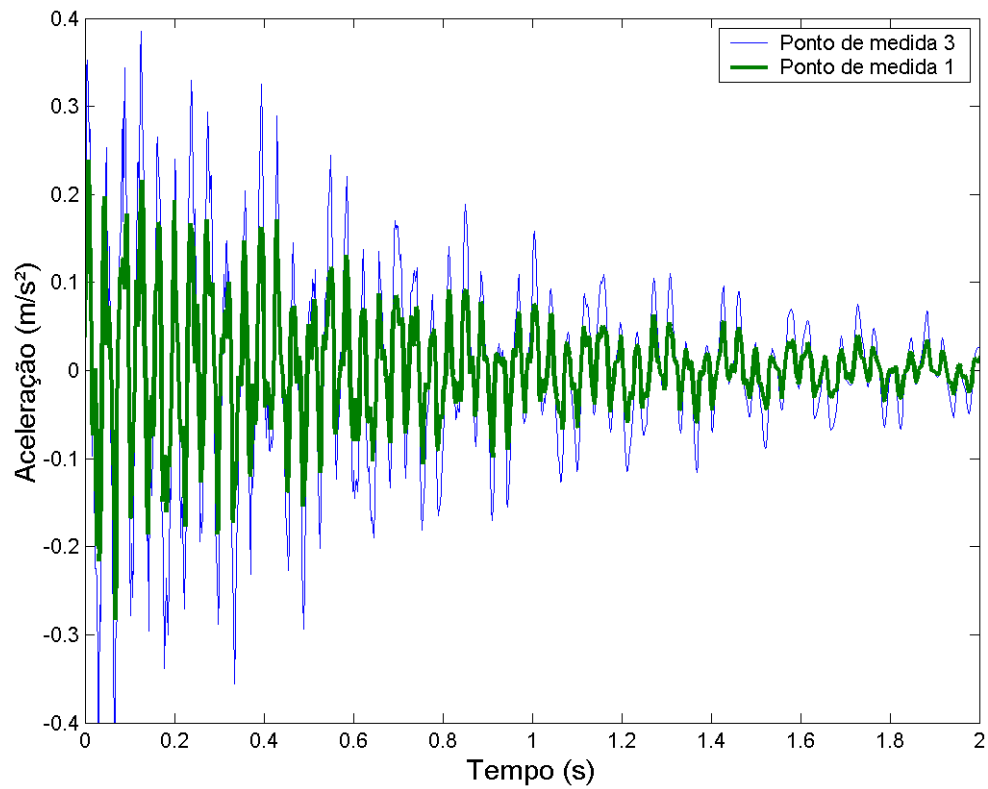


Figura 6.3: Acelerações medidas nos pontos 3 e 1 da viga de aço ensaiada.

$$\text{d) } \mathbf{MVm} = \begin{bmatrix} 1 & 1 & 2 \\ \vdots & \vdots & \vdots \\ 1 & 1 & 2 \end{bmatrix}_{(70 \times 3)}$$

$$\text{e) } \text{ref} = 1$$

$$\text{f) } \text{varint} = [10 \quad 2 \quad 40]$$

$$\text{g) } \text{varreal} = [0.15 \quad 1 \quad 100000 \quad 0.002]$$

$$\text{h) } \text{flag1} = 4$$

$$\text{i) } \text{flag2} = 2$$

$$\text{j) } \text{nd} = 1$$

$$\text{k) } \text{limxi} = [0 \quad 0.10]$$

Utilizados no MIDaS-Trunc:

$$\text{a) } \mathbf{limfreq} = \begin{bmatrix} 6.70 & 25.80 & 56.70 \\ 6.90 & 26.80 & 57.30 \end{bmatrix} \text{ (foram analisadas somente as 3 primeiras frequências).}$$

Utilizados na rotina MIDaS-Filter:

$$\text{a) } \mathbf{tipofilt} = [1 \quad 2]$$

Utilizados na rotina MIDaS-Assembly:

a) **td** - Como mencionado anteriormente, foram executados 5 testes para cada série de medidas. Desta forma, o vetor **td** possui 70 posições, contendo os números de 1 a 14 (*setups* do ensaio) com cada um deles repetindo 5 vezes.

Utilizados na rotina MIDaS-Visual:

a) **visu** = [3 17] Neste caso, somente os três primeiros modos de vibração serão visualizados. Como existem dois pontos que definem condições de contorno nos extremos da estrutura, o parâmetro **npontos** foi definido como 15 (número de pontos observados) + 2 (número de pontos extremos).

$$\text{b) } \mathbf{ap} = \begin{bmatrix} 3.3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1.3 & 2.3 & 4.3 & 5.3 & 6.3 & 7.3 & 8.3 & 9.3 & 10.3 & 11.3 & 12.3 & 13.3 & 14.3 & 15.3 \end{bmatrix}^T$$

6.1.1 Resultados

Como descrito na seção 5.3, cada algoritmo de identificação possui saídas específicas. Para este primeiro exemplo, após a análise MIDaS-Ident, obtém-se o histograma de frequências mostrado pela Figura 6.4.

Percebe-se claramente a identificação das cinco primeiras frequências naturais, em especial as três primeiras que possuem número de ocorrências superior às demais.

Optou-se por continuar com as demais análises utilizando apenas as frequências na faixa dos 6, 26 e 57 Hz.

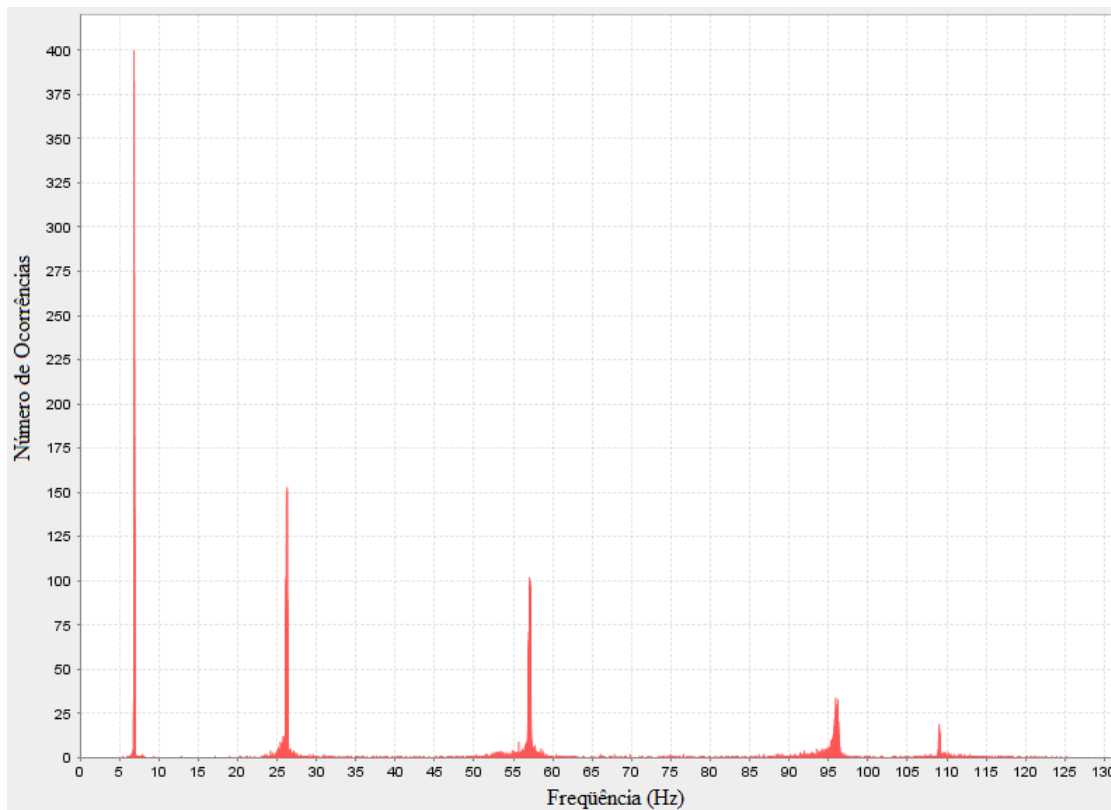


Figura 6.4: Histograma de frequências após a análise MIDaS-Ident.

Após a análise MIDaS-Trunc e com a definição dos limites de truncamento descritos anteriormente, obtém-se os gráficos das três primeiras frequências naturais e taxas de amortecimento ao longo do sinal, bem como os seus respectivos histogramas, como mostram as Figuras 6.5, 6.6, 6.7 e 6.8, respectivamente.

Prosseguindo com a rotina MIDaS-Filter após definir a filtragem via frequências naturais e utilizando um intervalo de confiança de 95%, tem-se os resultados mostrados pelas Figuras 6.9, 6.10, 6.11 e 6.12:

Para a penúltima rotina, MIDaS-Assembly, obtém-se os resultados finais para as frequências naturais e taxas de amortecimento mostrados na Tabela 6.1. As Figuras 6.13, 6.14, 6.15 e 6.16 apresentam os gráficos contendo as frequências naturais e taxas de amortecimento ao longo do sinal e os histogramas destes parâmetros após a análise MIDaS-Assembly.

A Tabela 6.2 compara os resultados obtidos pela aplicação MIDaS com os

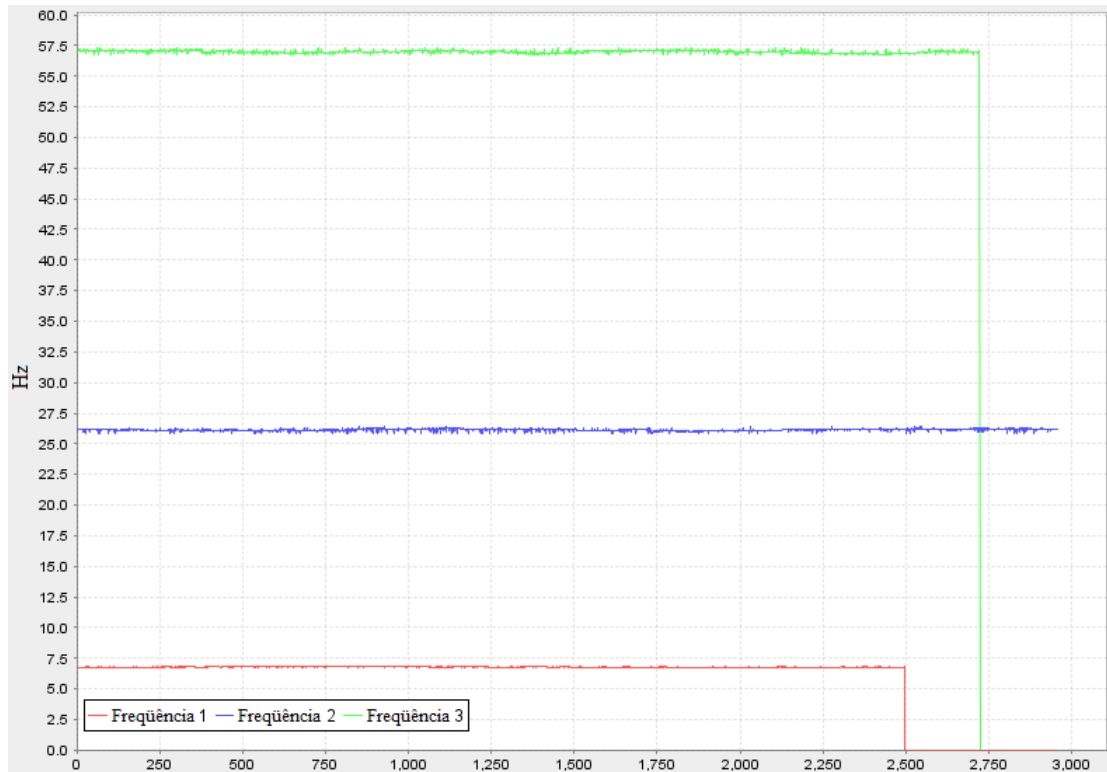


Figura 6.5: Frequências naturais obtidas ao longo do sinal após a análise MIDaS-Trunc.

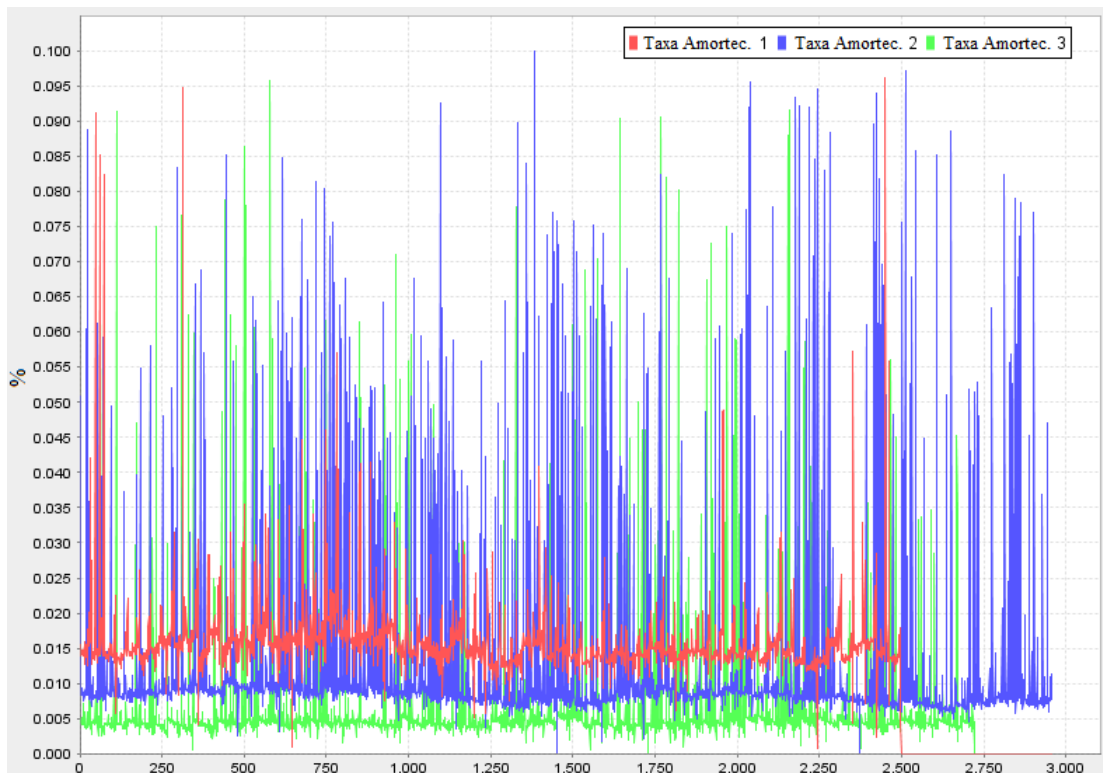


Figura 6.6: Taxas de amortecimento obtidas ao longo do sinal após a análise MIDaS-Trunc.

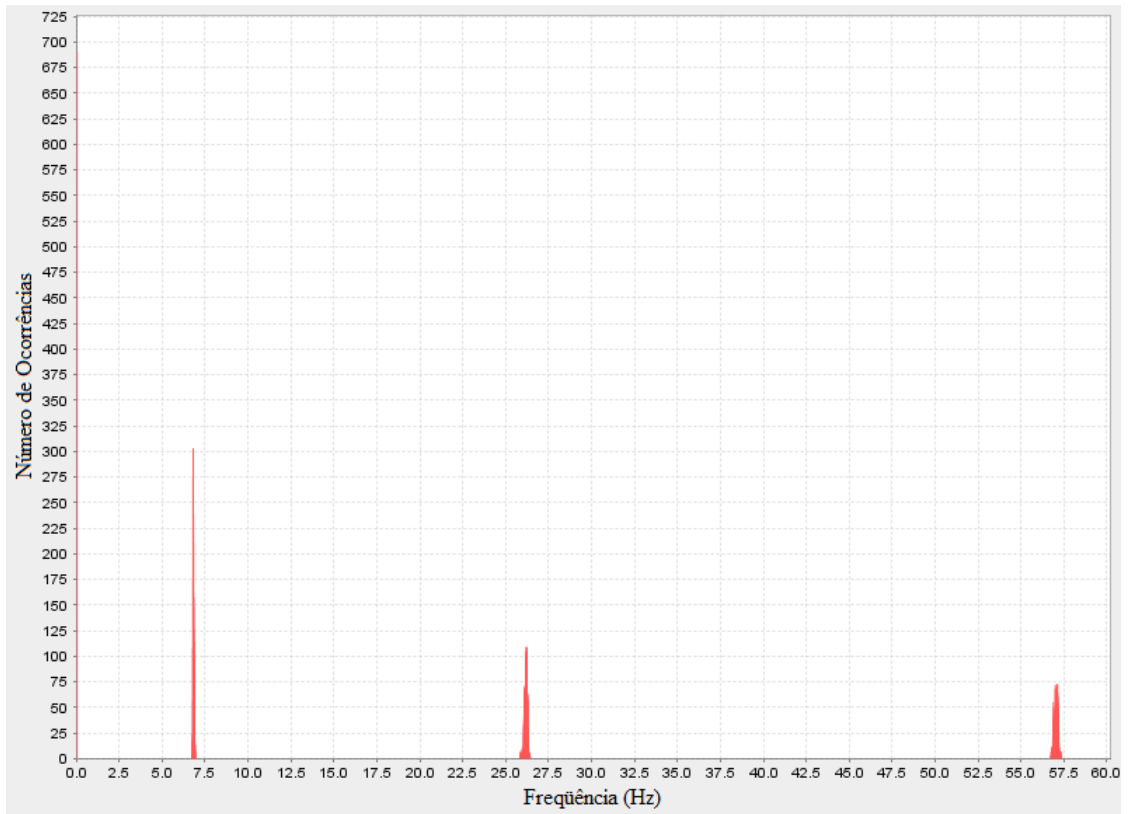


Figura 6.7: Histograma de frequências após a análise MIDaS-Trunc.

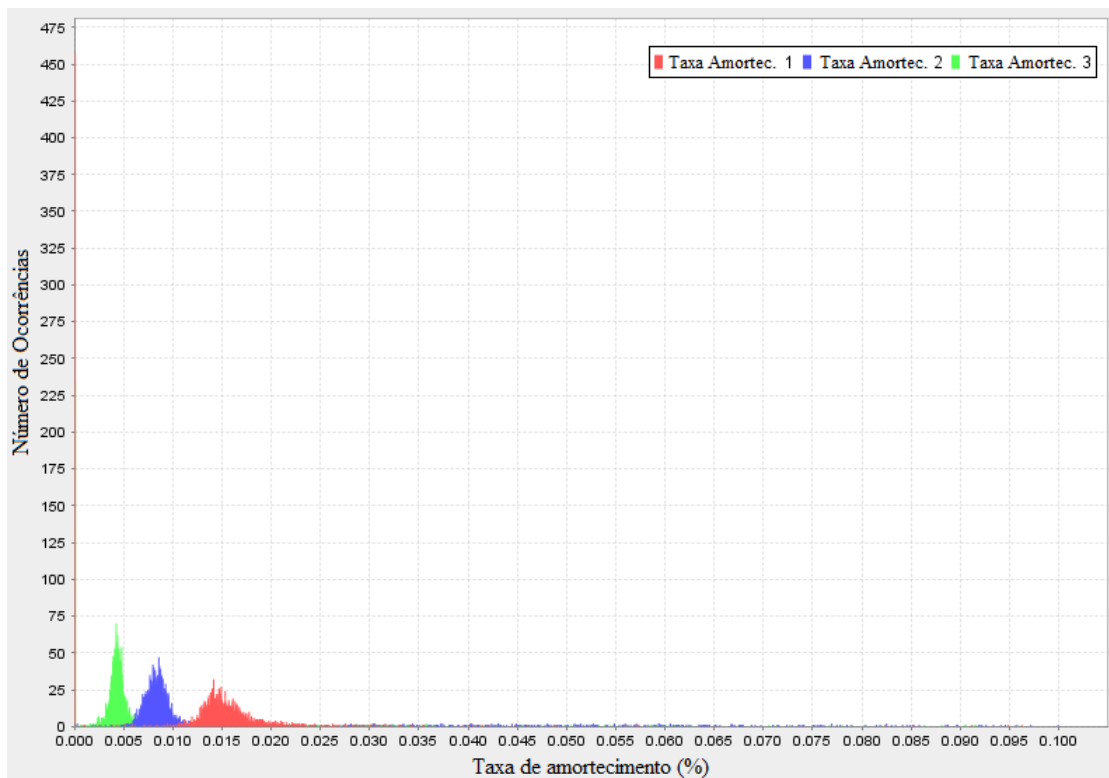


Figura 6.8: Histograma de amortecimentos após a análise MIDaS-Trunc.

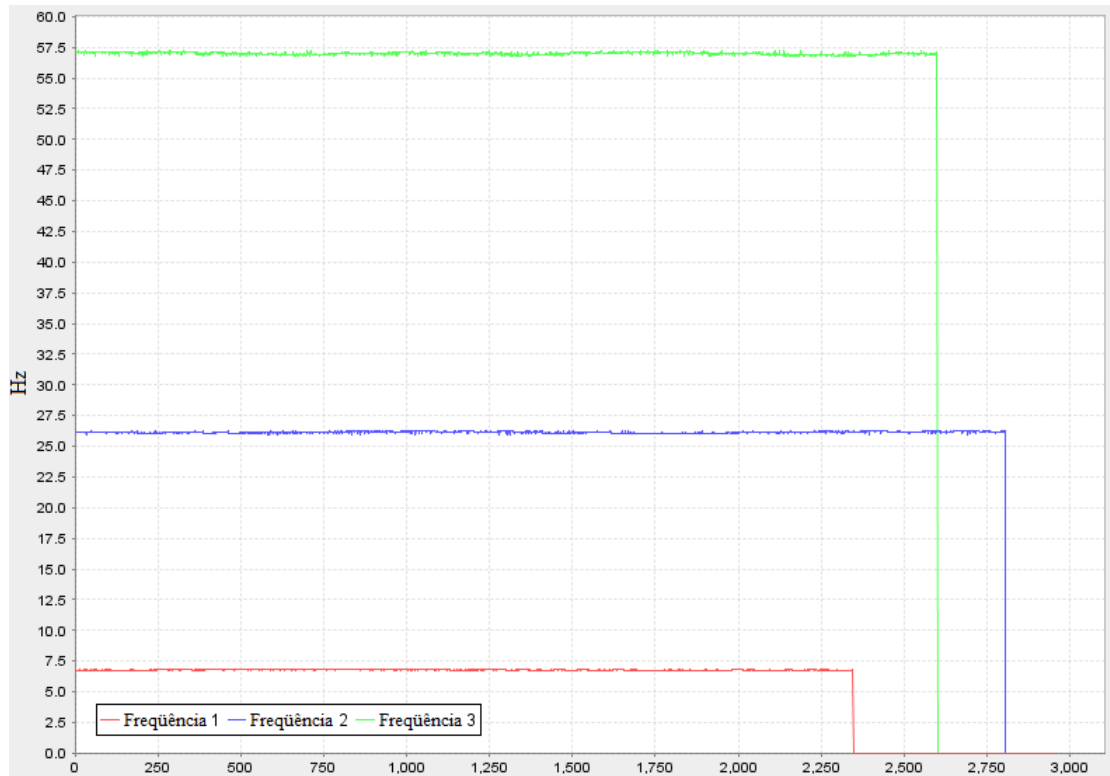


Figura 6.9: Frequências naturais obtidas ao longo do sinal após a análise MIDaS-Filter.

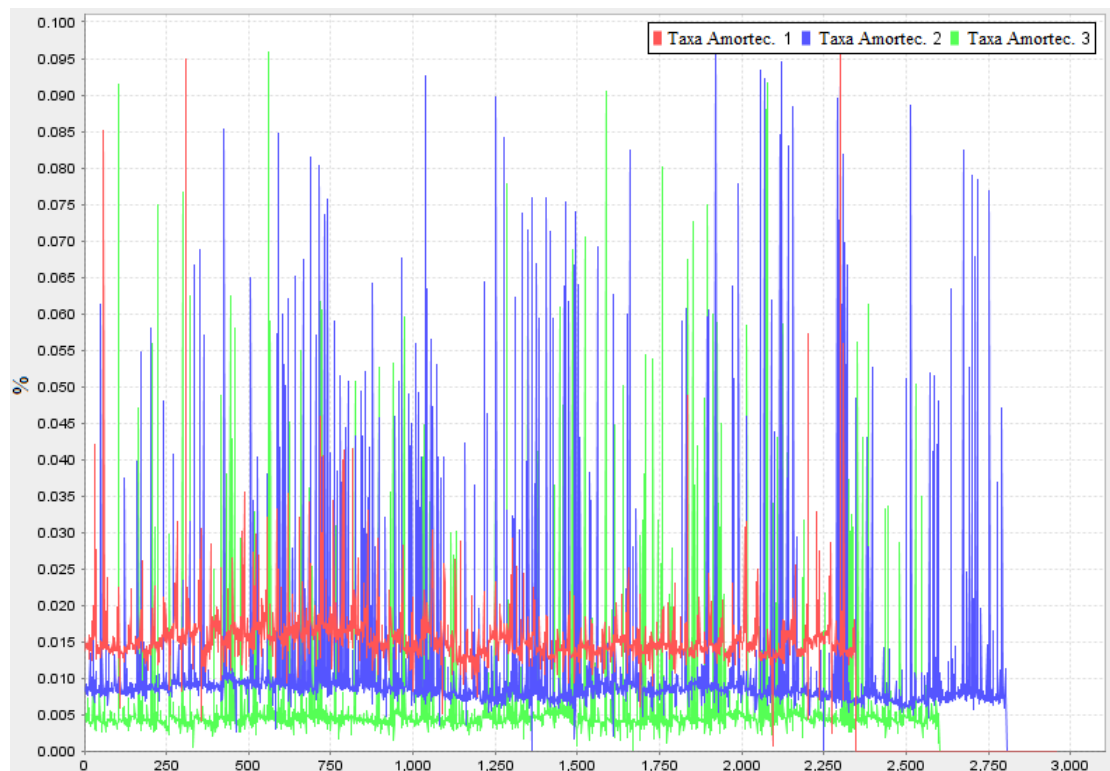


Figura 6.10: Taxas de amortecimento obtidas ao longo do sinal após a análise MIDaS-Filter.

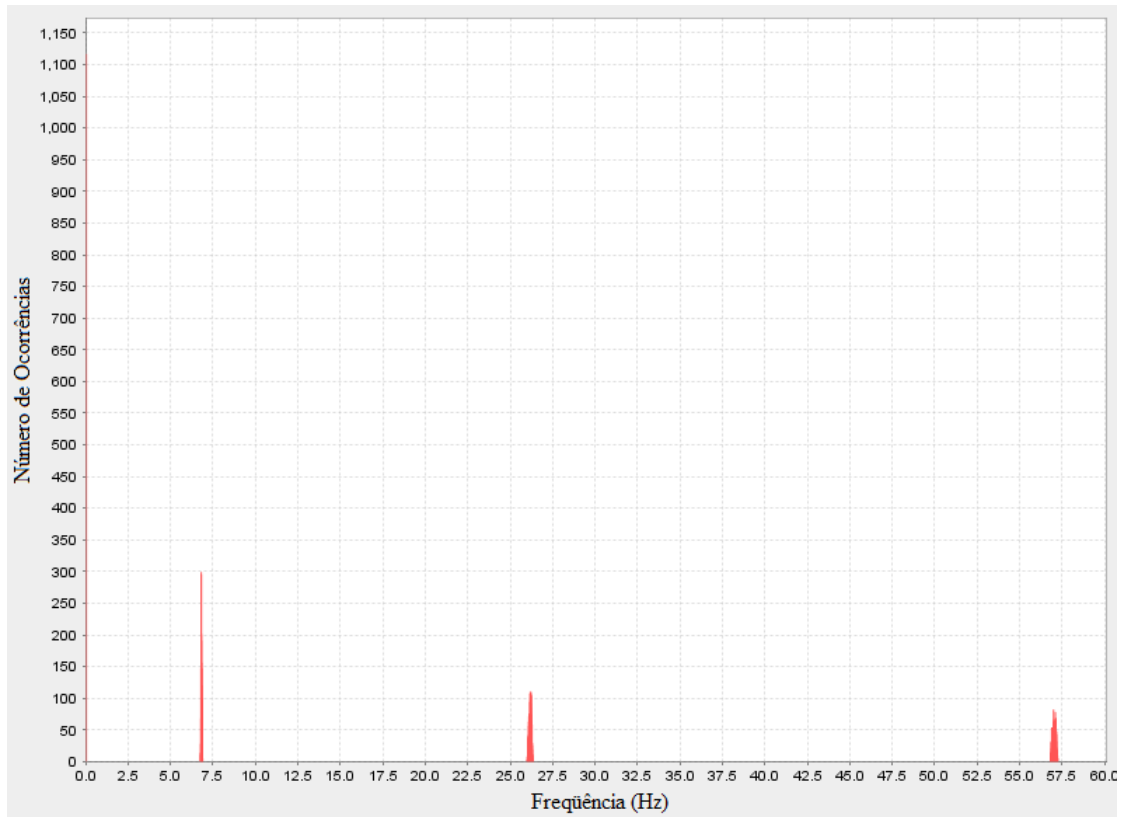


Figura 6.11: Histograma de frequências após a análise MIDaS-Filter.

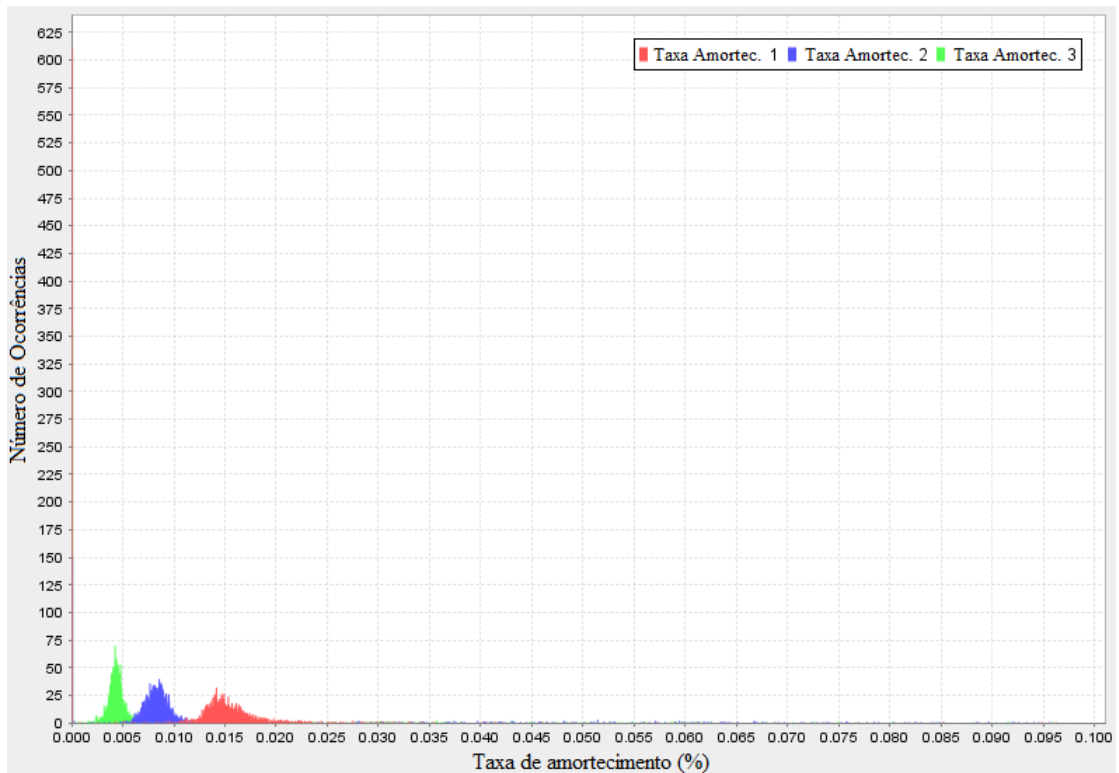


Figura 6.12: Histograma de amortecimentos após a análise MIDaS-Filter.

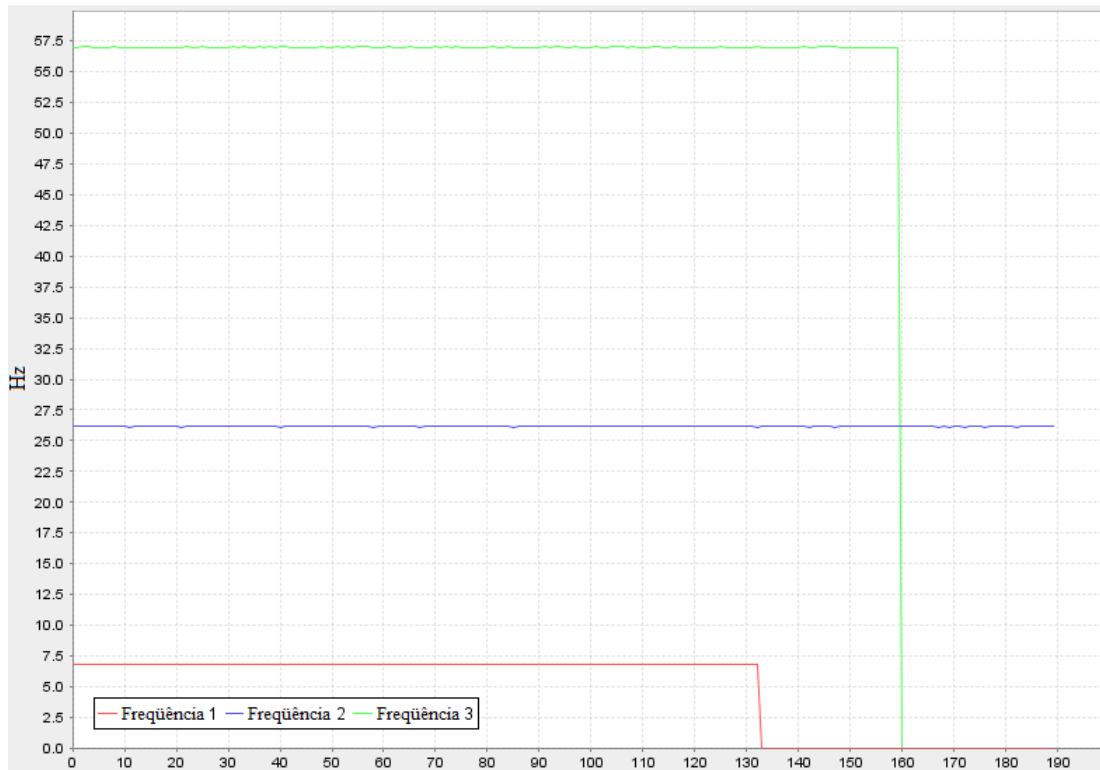


Figura 6.13: Frequências naturais obtidas ao longo do sinal após a análise MIDaS-Assembly.

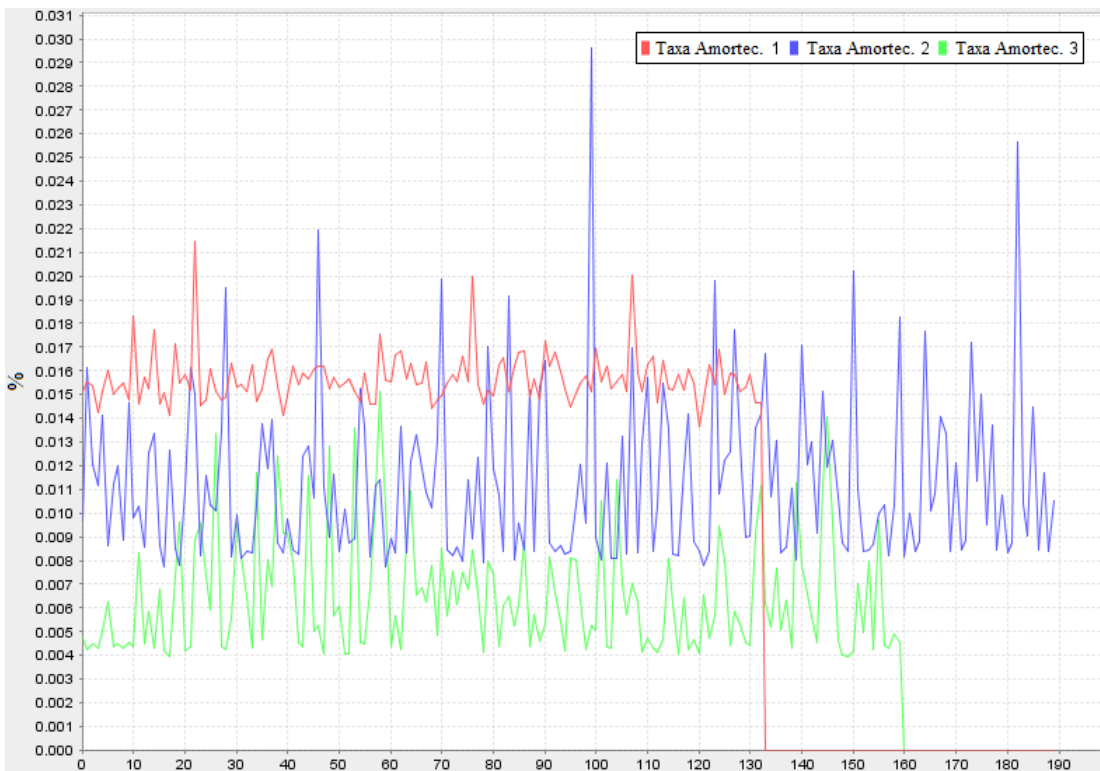


Figura 6.14: Taxas de amortecimento obtidas ao longo do sinal após a análise MIDaS-Assembly.

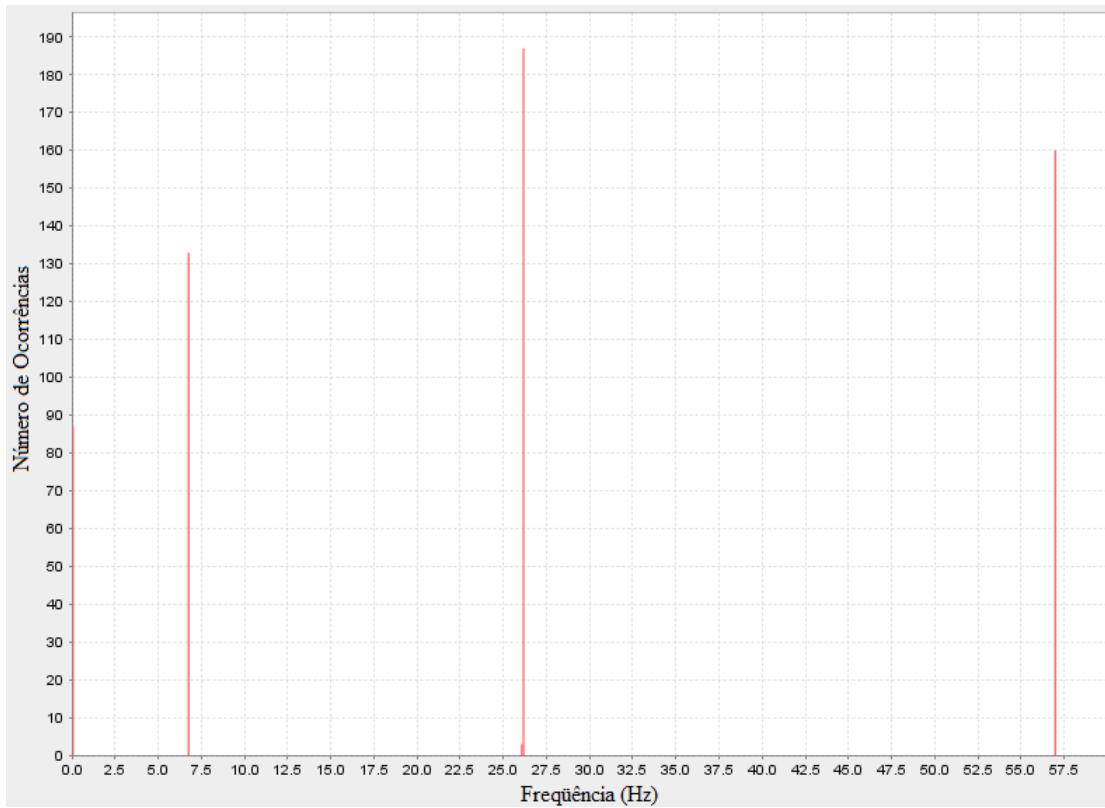


Figura 6.15: Histograma de frequências após a análise MIDaS-Assembly.

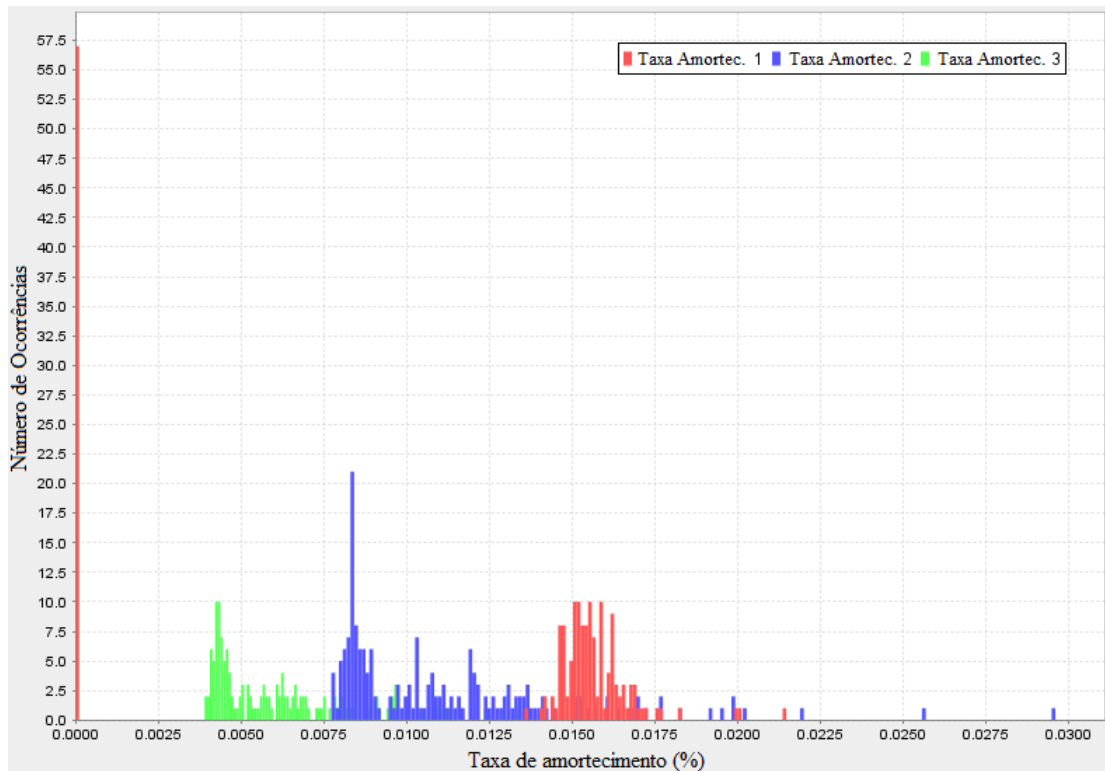


Figura 6.16: Histograma de amortecimentos após a análise MIDaS-Assembly.

Tabela 6.1: Valores médios e desvios-padrões dos parâmetros modais identificados pela aplicação MIDaS.

i	Frequências naturais		Taxas de Amortecimento	
	Valor médio (Hz)	Desvio-padrão (Hz)	Valor médio (%)	Desvio-padrão (%)
1	6,79	0,01	0,0156	0,0010
2	26,14	0,01	0,0112	0,0035
3	56,99	0,02	0,0064	0,0024

valores teóricos calculados pela Eq.(6.1):

$$\omega_i = \frac{i^2\pi}{2} \sqrt{\frac{EI}{\rho SL^4}}, \quad i = 1, 2, \dots \quad \text{em Hz}, \quad (6.1)$$

onde E é o módulo de elasticidade longitudinal ($E_{\text{aço}} = 210\text{GPa}$), I é o momento de inércia da seção e é igual a $1,54 \times 10^{-9}\text{m}^4$, ρ é a massa específica ($\rho_{\text{aço}} = 7850\text{Kg/m}^3$), $S = 2.29 \times 10^{-4}\text{m}^2$ representa a área da seção e L é o comprimento da barra, igual a 1,80m.

Tabela 6.2: Comparativo entre frequências teóricas e experimentais - Viga bi-apoiada.

Frequência	Teórica (Hz)	Experimental (Hz)	Diferença (%)
1	6,51	6,79	4,18
2	26,06	26,14	3,22
3	58,63	56,99	2,88

A partir da análise da Tabela 6.2, nota-se que os valores médios experimentais de frequências naturais obtidos com o MIDaS foram bastante próximos dos valores teóricos calculados.

Os modos próprios de vibração também possuem solução analítica e são dados por:

$$\phi_i = \sin\left(\frac{i\pi x}{L}\right) \quad i = 1, 2, 3, \dots \quad (6.2)$$

Utilizando o arquivo de saída “output_visual.zip” (gerado após a análise MIDaS-Visual) que contém os modos de vibração identificados, é possível imprimir no programa Matlab os modos próprios experimentais e teóricos como mostra a Figura 6.17.

Nota-se que os resultados obtidos pelo MIDaS para os modos de vibração foram, também, relativamente próximos aos teóricos.

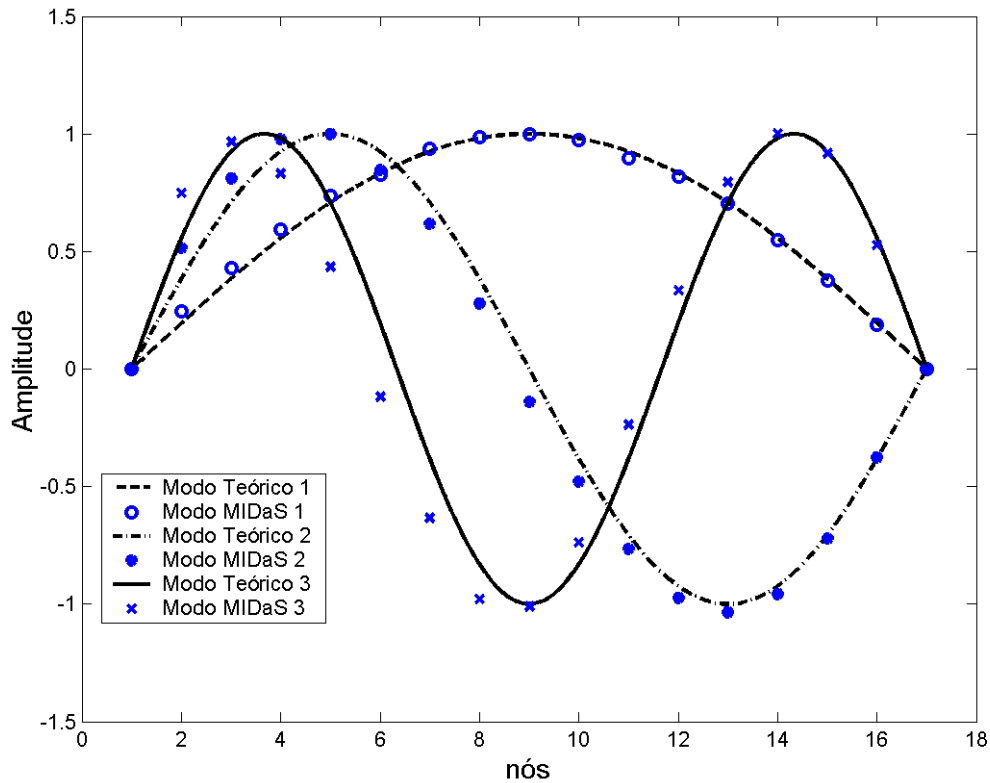


Figura 6.17: Modos de vibração da viga bi-apoiada. Comparativo entre os resultados teórico e da aplicação MIDaS.

No Apêndice A deste trabalho são disponibilizados os arquivos de parâmetros utilizados nesta análise.

6.1.2 Tempos de análise

A aplicação MIDaS possui tomadas de tempo internas e que são disponibilizadas ao usuário no arquivo de *log*, após cada rotina.

Com o objetivo de comparar o desempenho entre a linguagem Java e o programa Matlab, as mesmas rotinas foram executadas e seus tempos calculados. Para ambos os casos, os testes foram realizados em um computador com processador Pentium 4 de 3.0 GHz com 1 GB de memória RAM, rodando no sistema operacional Windows NT.

As tomadas de tempo foram considerados da seguinte forma:

- T0: Tempo estimado decorrido para a transferência dos arquivos de parâmetros e de dados, considerando conexão banda larga 1Mbps (125kB/s);
- T1: Tempo decorrido durante a leitura e descompactação dos arquivos de parâmetros e de dados;
- T2: Tempo decorrido durante o processo de identificação realizado;
- T3: Tempo decorrido para impressão dos gráficos e para escrita dos arquivos de saída.

O tempo T0 será estimado apenas para a transferência do arquivo de dados, desconsiderando o tempo necessário para a transmissão do arquivo de parâmetros, uma vez que este último possui um tamanho bastante inferior comparado ao arquivo de dados.

Nas tomadas de tempo no Matlab somente o tempo T2 foi considerado, já que as ações dos tempos T0, T1 e T3 ocorrem exclusivamente na aplicação MIDaS.

O tempo T1 ocorre somente nas análises MIDaS-Ident e MIDaS-Trunc, pois são estas rotinas que descompactam os arquivos de dados referentes a cada passo da análise.

A Tabela 6.3 mostra os tempos decorridos para três situações durante a análise representada neste exemplo. A primeira delas se refere ao tempo necessário para executar as rotinas do MIDaS no computador local; a segunda, o tempo decorrido para as análises do MIDaS via internet e, a terceira, utilizando o programa Matlab (desenvolvido por Barbosa [40]), no computador local.

O arquivo de dados deste exemplo possui aproximadamente 1,94MB.

No que se refere à análise dos tempos de execução, observou-se que o MIDaS obteve um desempenho muito superior à plataforma Matlab, sendo que a aplicação web apresentou uma melhor performance em relação à sua execução local, quando se desconsidera o tempo de transmissão de dados. Este último resultado se deve,

Tabela 6.3: Comparativo de tempo entre as aplicações - Viga bi-apoiada. Os tempos apresentados estão em segundos. (*) - Transmissão de dados. (**) - Tempo desconsiderando a transmissão de dados. (***) - Tempo considerando a transmissão de dados.

	MIDaS (local)			MIDaS (internet)			Matlab (local)		
T. dados(*)	-			T0 = 17			-		
<i>Análise</i>	T1	T2	T3	T1	T2	T3	T1	T2	T3
Ident	0,31	15,03	0,17	0,21	10,12	0,14	-	80,22	-
Trunc	0,30	1,40	2,23	0,23	1,10	3,20	-	2,56	-
Filter	-	0,26	2,11	-	0,21	3,18	-	0,47	-
Assembly	-	0,45	0,89	-	0,33	1,09	-	5,83	-
Visual	-	0,01	0,06	-	0,01	0,10	-	0,02	-
Tempo Total	23,21			19,92(**) 36,92(***)			89,10		

possivelmente, ao fato de que durante a execução local mais recursos computacionais, como memória RAM, processamento, etc, são dispendidos com ambiente de desenvolvimento (Eclipse), diminuindo o desempenho local.

O tempo de transmissão de dados, que neste exemplo foi de 17s, pode ser consideravelmente reduzido utilizando conexões mais rápidas.

6.2 Ensaios dinâmicos sobre ponte férrea

Os ensaios dinâmicos referentes a este exemplo de aplicação foram realizados sobre a ponte férrea que liga as cidades de Paris e Lyon, na França, por onde passam trens de alta velocidade, denominados TGV. Nestes ensaios foram utilizados 8 acelerômetros, sendo que um deles (o de número 7) permitia a aquisição dos pontos de medida em duas direções, totalizando 9 sinais de medidas.

O acelerômetro 2 foi utilizado como o de referência e foram realizadas 12 séries de medidas no total.

As Figuras 6.18 e 6.19 mostram o esquema de aquisição e os 9 sinais adquiridos na primeira série de medidas, respectivamente.

O sinal apresentado pela Figura 6.19 representa dois momentos do ensaio: o primeiro, descrito pelo trecho A1, mostra o momento em que ocorre a passagem do trem e o segundo, mostrado pelo trecho A2, representa a resposta da estrutura em vibrações livres, isto é, após a excitação causada pela passagem do trem. Vi-

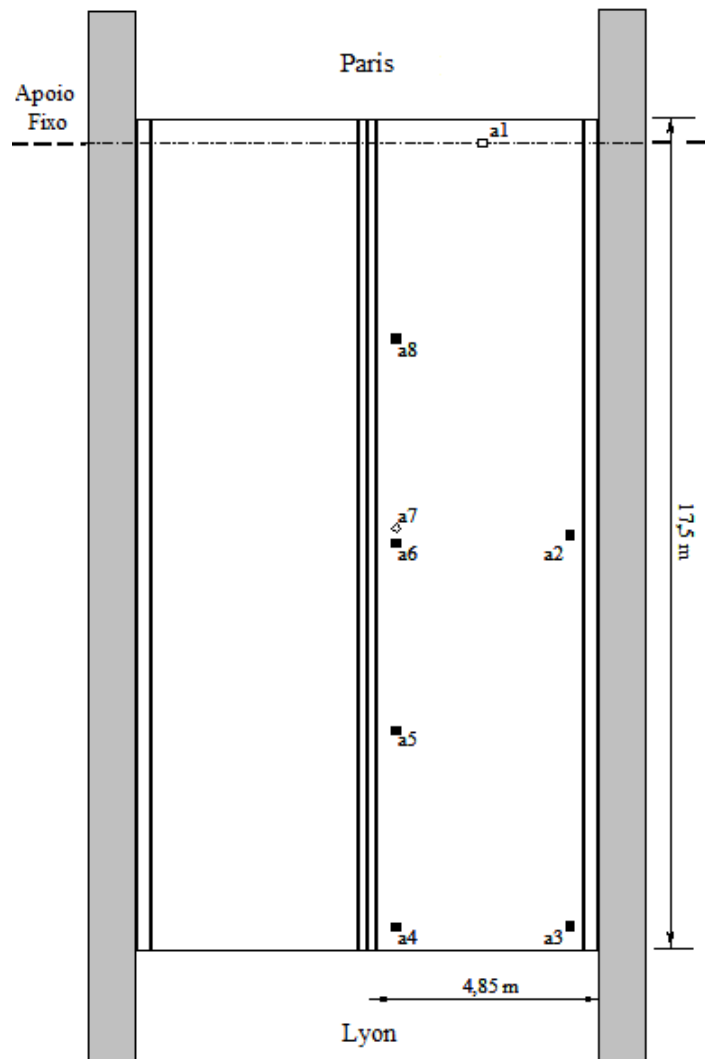


Figura 6.18: Esquema de instrumentação da ponte férrea - TGV.

sando a identificação dos parâmetros modais da estrutura isolada, os resultados apresentados neste exemplo são relativos ao trecho A2.

6.2.1 Resultados

Os parâmetros utilizados para a configuração da aplicação MIDaS para este exemplo são mostrados no Apêndice A deste trabalho.

Após as análises MIDaS-Ident, MIDaS-Trunc, MIDaS-Filter e MIDaS-Assembly, têm-se os valores para as três primeiras frequências naturais e taxas de amortecimento, como mostra a Tabela 6.4.

A Tabela 6.5 compara os resultados obtidos pela aplicação MIDaS com os

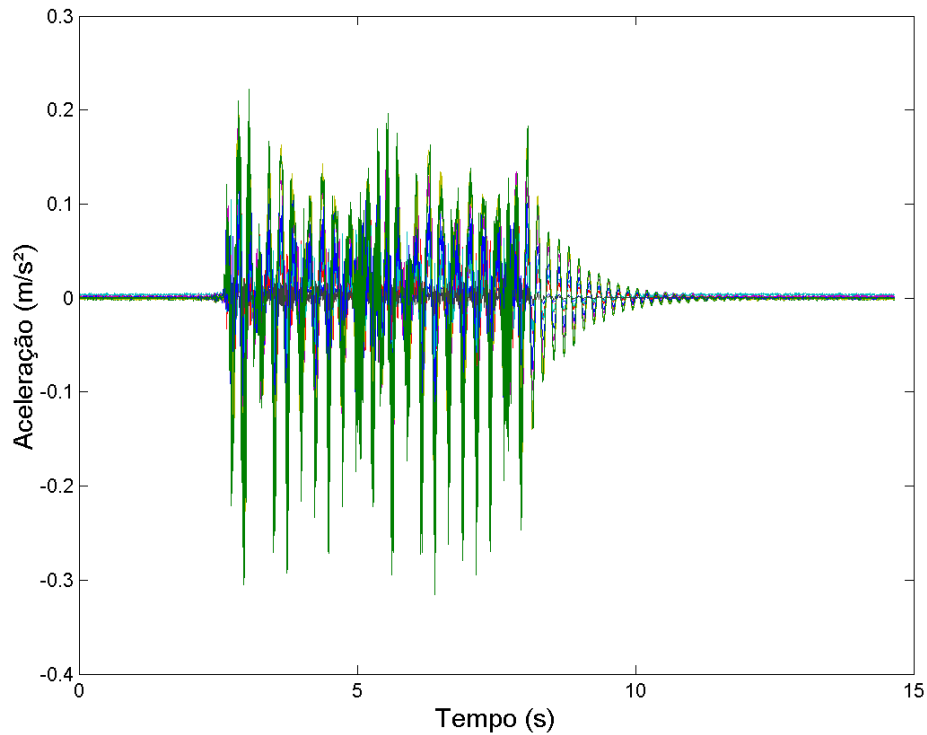


Figura 6.19: Sinais adquiridos para nove pontos instrumentados na primeira série de medidas sobre a ponte férrea.

Tabela 6.4: Valores médios e desvios-padrões dos parâmetros modais identificados pela aplicação MIDaS - Ponte férrea.

i	Frequências naturais		Taxas de Amortecimento	
	Valor médio (Hz)	Desvio-padrão (Hz)	Valor médio (%)	Desvio-padrão (%)
1	5,40	0,01	0,0479	0,0009
2	12,65	0,08	0,0273	0,0094
3	17,52	0,19	0,0398	0,0039

presentes na referência [27]. A partir da análise desta tabela, percebe-se que os valores obtidos pelo MIDaS são muito próximos aos identificados pela referência citada.

Tabela 6.5: Comparativo entre frequências obtidas pelo MIDaS e pela referência [27] - Ponte férrea.

Frequência	MIDaS (Hz)	Referência [27] (Hz)	Diferença (%)
1	5,40	5,43	0,50
2	12,65	12,75	0,80
3	17,52	17,22	1,70

6.2.2 Tempos de análise

Similarmente ao exemplo de aplicação anterior, foram tomados os tempos de análise como mostra a Tabela 6.6.

Tabela 6.6: Comparativo de tempo entre as aplicações - Ponte férrea. Os tempos apresentados estão em segundos. (*) - Transmissão de dados. (**) - Tempo desconsiderando a transmissão de dados. (***) - Tempo considerando a transmissão de dados.

	MIDaS (local)			MIDaS (internet)			Matlab (local)		
T. dados(*)	-			T0 = 200			-		
Análise	T1	T2	T3	T1	T2	T3	T1	T2	T3
Ident	1,64	23,74	0,50	1,55	20,86	0,42	-	100,12	-
Trunc	1,57	0,56	1,95	1,46	0,48	2,08	-	0,81	-
Filter	-	0,04	1,88	-	0,03	1,97	-	0,08	-
Assembly	-	0,13	1,01	-	0,10	0,94	-	0,28	-
Visual	-	0,01	0,07	-	0,02	0,12	-	0,03	-
Tempo Total	33,10			30,03(**) 230,03(***)			101,32		

O arquivo de dados deste exemplo possui aproximadamente 22,00MB.

No que se refere à análise dos tempos de execução, mais uma vez observou-se que o MIDaS obteve um desempenho muito superior à plataforma Matlab e que a aplicação web apresentou uma performance ligeiramente melhor em relação à sua execução local, desconsiderando-se os tempos de transmissão de dados.

6.3 Ensaios dinâmicos sobre a ponte Z24

A ponte Z24 era uma obra de arte que ligava as cidades de Utzenstorf e Koppigen na Suíça. Ela foi construída entre os anos de 1961 e 1963 e se situava sobre a auto-estrada entre Berna a Zurique (auto-estrada A1). As principais características geométricas da ponte são mostradas na Figura 6.20.

Esta ponte foi destruída para dar lugar a uma outra (ponte A36) construída paralelamente à Z24. Porém, antes da sua destruição, vários ensaios dinâmicos foram realizados com o objetivo de realizar a identificação dos parâmetros modais da estrutura e, ainda, validar técnicas de detecção de danos impostos à ponte [41]. Estes experimentos foram feitos dentro de um projeto europeu denominado

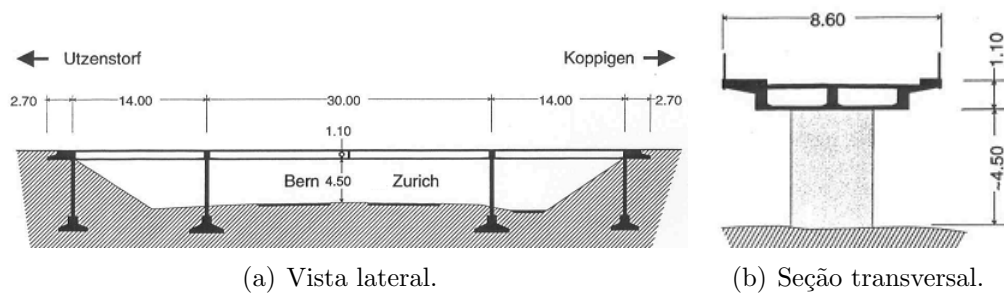


Figura 6.20: Características geométricas da ponte Z24.

SIMCES (System Identification to Monitor Civil Engineering) [42]. A Figura 6.21 mostra o plano de instrumentação da ponte.

Para a execução dos ensaios dinâmicos, foram utilizados um total de 154 pontos de medidas com acelerômetros, sendo 35 pontos com acelerômetros medindo três direções, 27 pontos com acelerômetros medindo duas direções e 92 pontos com acelerômetros medindo uma direção.

As acelerações foram tomadas em 9 grupos de medições (*setups*), guardando-se sempre 3 pontos de medidas de referência em comum: R1, R2 e R3 (ver Figura 6.21).

A ponte foi ensaiada 17 vezes, sendo que em cada um destes ensaios a estrutura possuía um estado de danificação diferente. A excitação foi aplicada na ponte de duas formas: excitação ambiente AVT (Ambient Vibration Test - Teste vibratório com solicitação ambiente) e excitação imposta por excitadores de forma aleatória FVT (Forced Vibration Test - Teste vibratório com solicitação forçada aleatória).

No presente trabalho apresentam-se os resultados obtidos para a solicitação FVT utilizando apenas as acelerações verticais medidas da linha de captores 99 a 143 (ver Figura 6.21) do primeiro conjunto de ensaios. A Figura 6.22 mostra uma resposta típica para a excitação FVT aplicada à estrutura.

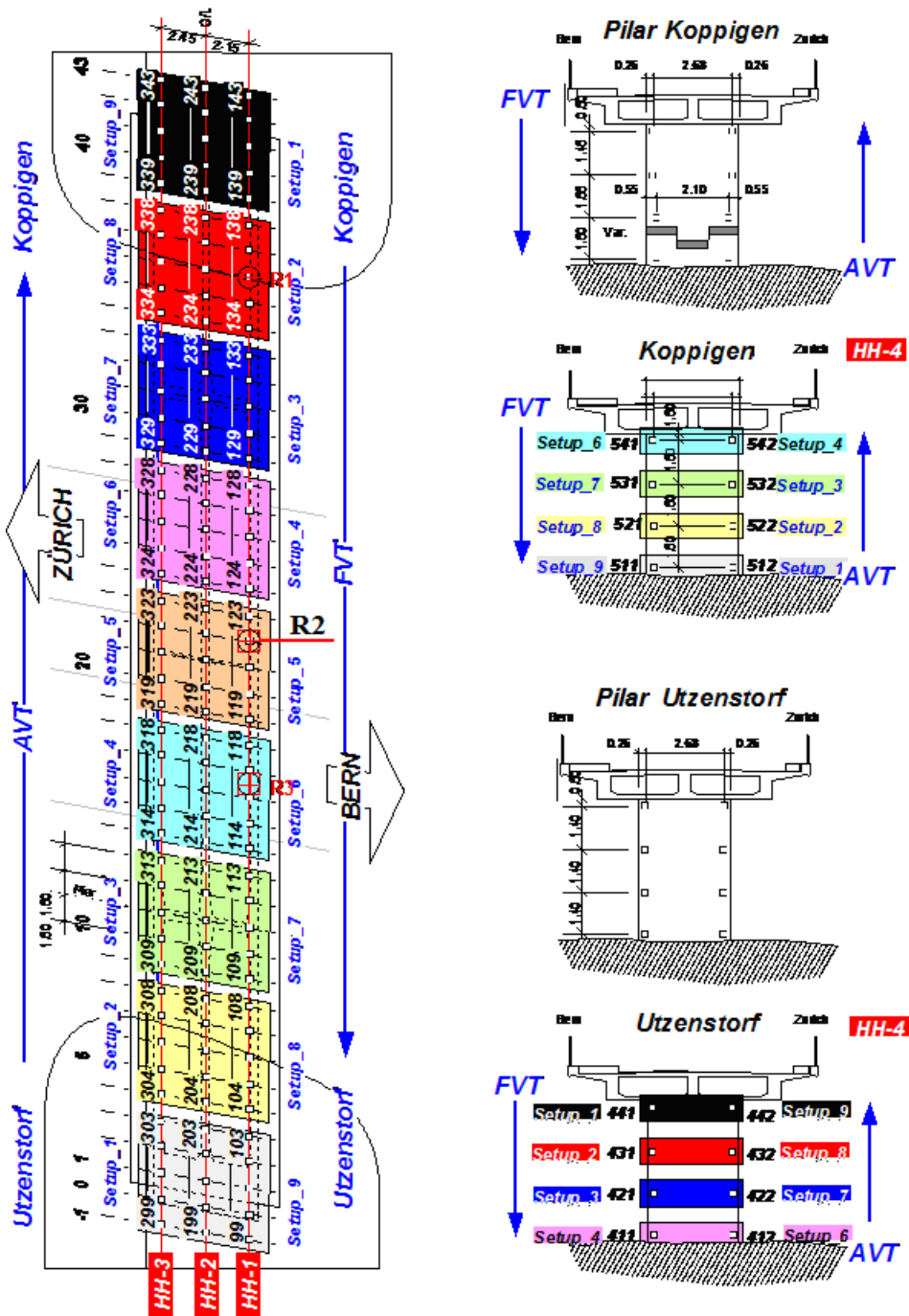


Figura 6.21: Esquema de instrumentação da ponte Z24.

6.3.1 Resultados

Os parâmetros utilizados para a configuração da aplicação MIDaS para este exemplo são mostrados no Apêndice A deste trabalho.

Foi utilizado como referência o acelerômetro R2, com o objetivo de priorizar a identificação dos parâmetros com maior amplitude no vão central da ponte. A

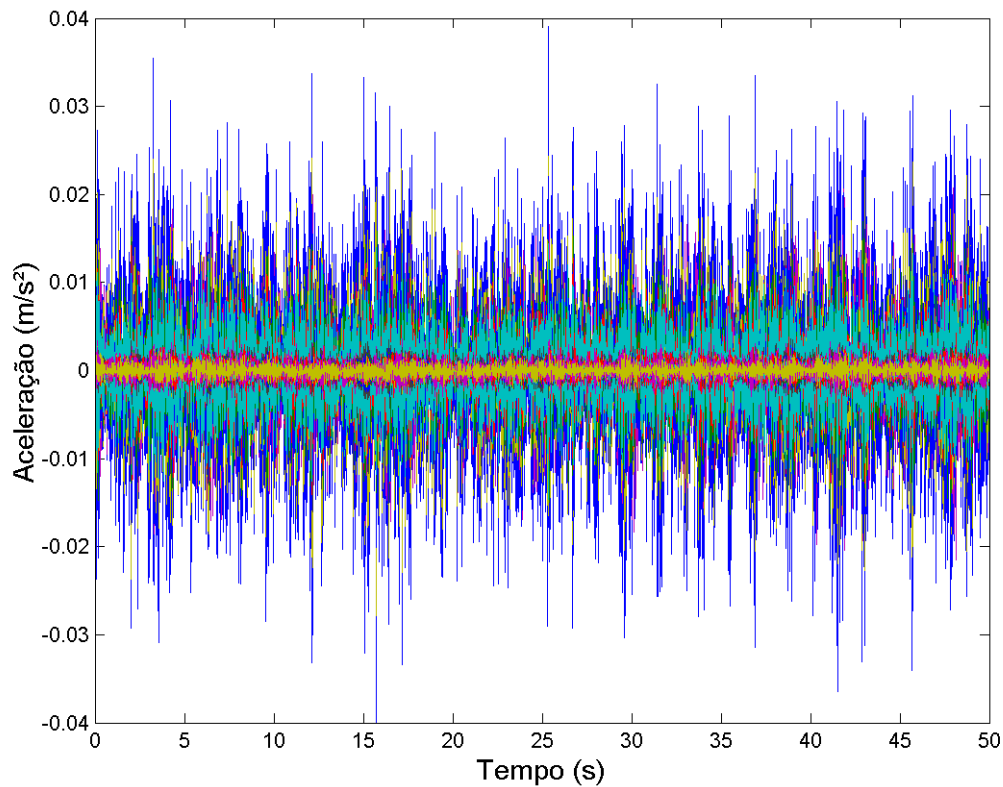


Figura 6.22: Sinais adquiridos para 33 pontos de medida instrumentados do primeiro conjunto de ensaios sobre a ponte Z24.

Figura 6.23 mostra o histograma de frequências obtido após a análise MIDaS-Ident.

Após a análise do histograma de frequências, prossegue-se com os algoritmos MIDaS-Trunc, MIDaS-Filter e MIDaS-Assembly e monta-se a Tabela 6.7 com os valores médios e desvios-padrões para as quatro primeiras frequências naturais identificadas e as respectivas taxas de amortecimento.

Tabela 6.7: Valores médios e desvios-padrões dos parâmetros modais identificados pela aplicação MIDaS - Ponte Z24.

i	Frequências naturais		Taxas de Amortecimento	
	Valor médio (Hz)	Desvio-padrão (Hz)	Valor médio (%)	Desvio-padrão (%)
1	3,91	0,01	0,0448	0.0087
2	9,93	0,03	0,0479	0.0067
3	10,65	0,05	0,0550	0,0072
4	12,77	0,17	0,0505	0,0046

A Tabela 6.8 compara os resultados obtidos para as frequências naturais pelo MIDaS-Assembly com os presentes nas referências [3] e [41].

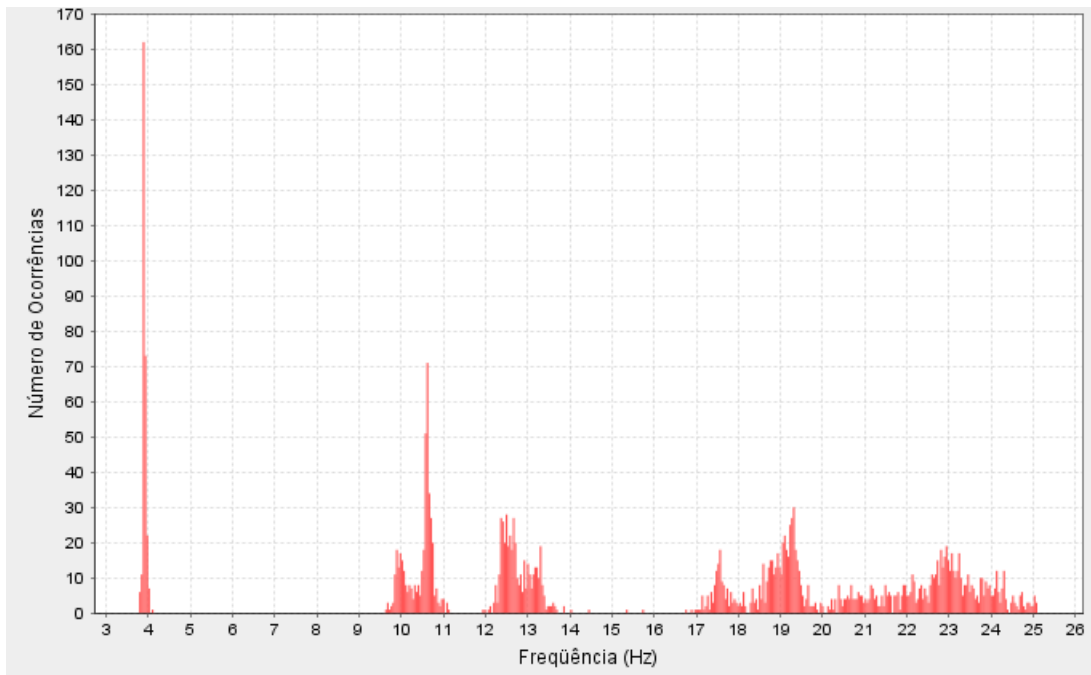


Figura 6.23: Histograma de frequências identificadas no ensaio da ponte Z24.

Tabela 6.8: Comparativo entre frequências obtidas pelo MIDaS e pelas referências [3] e [41] - Ponte Z24.

Frequência	MIDaS (Hz)	Referência [3] (Hz)	Referência [41] (Hz)
1	3,91	3.90	3.92
2	9,93	9.90	9.93
3	10,65	10.63	10.52
4	12,77	12.52	12.69

A partir da análise da Tabela 6.8 nota-se que os valores identificados pelo MIDaS são muito próximos aos obtidos pelas referências citadas, sendo que a maior diferença foi de 2,0% para a referência [3] (na 4ª frequência) e de 1,2% para a referência [41] (na 3ª frequência).

A Figura 6.24 apresenta o primeiro modo de vibração da estrutura utilizando a linha de captadores descrita anteriormente.

As Figuras 6.25(a) e 6.25(b) mostram o primeiro modo de vibração identificado nas referências [3] e [41], respectivamente. Novamente, para este exemplo, os resultados obtidos pela aplicação MIDaS quando comparados às estas referências foram muito próximos.

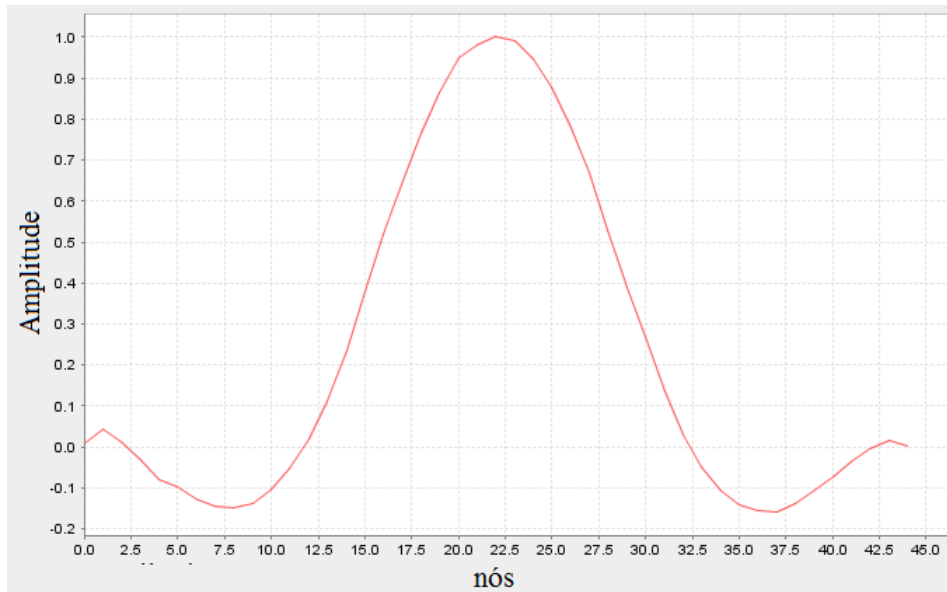
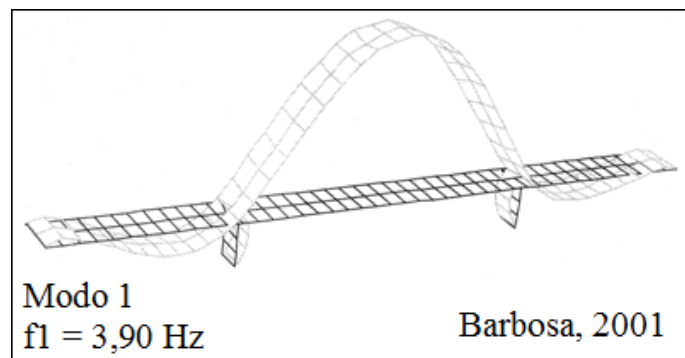
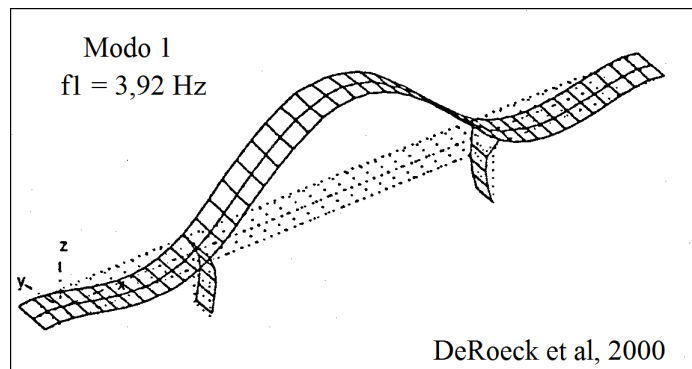


Figura 6.24: Primeiro modo de vibração identificado pelo MIDaS para a ponte Z24.



(a) Primeiro modo de vibração da ponte Z24 identificado por Barbosa *et al*[3].



(b) Primeiro modo de vibração da ponte Z24 identificado por DeRoeck *et al*[41].

Figura 6.25: Primeiro modo de vibração da ponte Z24 identificado por Barbosa *et al*[3] e DeRoeck *et al*[41].

6.3.2 Tempos de análise

Da mesma forma apresentada nos exemplos anteriores, a Tabela 6.9 mostra os tempos de análise decorridos.

Tabela 6.9: Comparativo de tempo entre as aplicações - Ponte Z24. Os tempos apresentados estão em segundos. (*) - Transmissão de dados. (**) - Tempo desconsiderando a transmissão de dados. (***) - Tempo considerando a transmissão de dados.

	MIDaS (local)			MIDaS (internet)			Matlab (local)		
T. dados(*)	-			T0 = 1020			-		
<i>Análise</i>	T1	T2	T3	T1	T2	T3	T1	T2	T3
Ident	2,13	33,77	0,71	2,01	31,05	0,66	-	175,02	-
Trunc	2,07	0,35	3,37	1,99	0,31	3,19	-	0,63	-
Filter	-	0,04	2,58	-	0,04	2,44	-	0,09	-
Assembly	-	0,04	1,94	-	0,03	2,06	-	0,10	-
Visual	-	0,02	0,10	-	0,01	0,08	-	0,03	-
Tempo Total	47,12			43,87(**) 1063,87(***)			175,87		

O arquivo de dados deste exemplo possui aproximadamente 124,00MB.

Novamente, no que se refere à análise dos tempos de execução, observou-se que o MIDaS obteve, mais uma vez, um desempenho muito superior à plataforma Matlab e uma melhor performance em relação à sua execução local, quando desconsiderado o tempo de transmissão de dados.

Capítulo 7

Considerações Finais

O presente trabalho teve como objetivo o desenvolvimento de uma aplicação web, o MIDaS, que permite a identificação modal de sistemas dinâmicos deformáveis através da internet. Nesse sentido, o usuário não precisa se preocupar em instalar ou comprar programas, uma vez que toda a análise é feita em um servidor e somente os dados de entrada e arquivos de saída são transferidos de uma máquina para a outra. Essa aplicação tem como característica ser gratuita e que, até então (Março de 2008), não existia na comunidade.

No campo da engenharia de *software*, destacam-se como contribuições deste trabalho características como: cooperação em âmbito global, por estar disponível na internet; e a facilidade da reutilização dos algoritmos, uma vez que foram implementados em linguagem orientada a objetos.

A aplicação MIDaS foi inteiramente desenvolvida utilizando a linguagem de programação Java. Todas as classes, métodos, bem como a aplicação web em si foram implementadas nos ambientes de desenvolvimento Eclipse e NetBeans. Para tal, foram utilizadas bibliotecas com o intuito de possibilitar a manipulação de arquivos, impressão de gráficos de resultados, além de outros pacotes numéricos como o de operações com matrizes, por exemplo.

A linguagem Java se mostrou eficiente em diversos aspectos. O primeiro a se considerar é a facilidade de integração entre as diversas bibliotecas e a possibilidade de se criar um próprio pacote com as classes desenvolvidas para a aplicação MIDaS

como fonte de utilização para outros usuários. O segundo aspecto se refere ao desempenho: apesar de não ter apresentado uma performance tão rápida quanto a do Fortran e considerando-se que o Java não é uma linguagem estaticamente compilada, isto é, não é diretamente interpretada pela máquina, pode-se dizer que os tempos de processamento decorridos durante as análises de identificação foram relativamente baixos. Em comparação aos tempos obtidos no programa Matlab, a performance do MIDaS foi muito superior.

Com o objetivo de se mostrar as potencialidades da aplicação MIDaS, três exemplos de ensaios experimentais foram utilizados. No primeiro deles, referente a uma viga de aço bi-apoiada ensaiada no Laboratório de Resistência dos Materiais, na Universidade Federal de Juiz de Fora, foram identificados os cinco primeiros parâmetros modais, sendo que três deles foram analisados. Os resultados obtidos foram próximos aos valores analíticos calculados.

No segundo exemplo foram analisados os ensaios realizados sobre a ponte férrea por onde passam os TGVs na França. Similarmente ao exemplo anterior, as três primeiras frequências naturais e respectivas taxas de amortecimento foram identificadas. Para ambos os parâmetros, os valores obtidos pelo MIDaS foram satisfatórios quando comparados à referência [27].

No terceiro e último exemplo de aplicação foi feita uma breve análise de identificação dos parâmetros modais a partir do ensaios realizados sobre a ponte Z24, na Suíça. A aplicação MIDaS foi capaz de identificar as quatro primeiras frequências naturais e taxas de amortecimento da estrutura. Além disso, o primeiro modo de vibração foi traçado e comparado às referências [3] e [41]. Para todos os parâmetros modais, os valores obtidos pelo MIDaS foram próximos aos identificados pelas referências citadas.

Entende-se que os obstáculos relativos aos tempos de transmissão de dados podem ser superados à medida em que se confirme a tendência moderna de se aumentar, cada vez mais, a velocidade de conexão de computadores, *palm-tops*, celulares, etc. à internet.

Dentre as dificuldades encontradas para o desenvolvimento deste trabalho, destacam-se a escassez de bibliotecas que dêem suporte para manipulação de matrizes de forma eficiente e que contemplem as mais diversas operações de álgebra linear; dificuldades inerentes à implementação dos algoritmos em linguagem orientada a objetos; e o domínio das tecnologias de suporte ao desenvolvimento de aplicações web.

Dentre as sugestões para trabalhos futuros, destacam-se:

- Aprimoramento dos algoritmos de identificação, com a implementação de novas técnicas não somente no domínio do tempo, mas também no domínio da frequência;
- Aperfeiçoamento dos tratamentos de erro na aplicação web, como filtros de arquivos, mensagens de erro em caso de formatação não suportada, entre outros;
- Tratamento da acessibilidade para viabilizar o uso de aparelhos portáteis (*PDA*s, *blackberries*, etc);
- Utilização de processamento distribuído para aprimorar a performance na solução dos algoritmos de identificação modal;
- Criação de um banco de dados interno capaz de salvar informações e resultados de diversos usuários da aplicação;
- Utilização de formatos de auto-descrição como XML, por exemplo, para definição dos arquivos de parâmetros;
- Criação de um serviço web, tornando disponível à comunidade alguns dos métodos de identificação modal implementados;
- Utilização de novas tecnologias para criação de interfaces ricas com o objetivo de incluir novas funcionalidades e melhorar a interação com o usuário, permitindo impressão de gráficos em tempo real, por exemplo.

Referências Bibliográficas

- [1] Battista R.C., Pfeil M.S. “Reduction of Vortex-Induced Oscillations of Rio-Niterói Bridge by Dynamic Control Devices”. *Journal of Wind Engineering and Industrial Aerodynamics*, 84(3): pp. 273–288, 2000.
- [2] Battista R.C. “Structural Monitoring of the World Largest Span Steel Box-Girder Bridge”. *Proceedings of the EUROLYN (European Dynamics)*, 2005.
- [3] Barbosa F.S. “Identification Modale du Pont Z24”. Technical Report. *Laboratoire National des Ponts et Chaussées/France*, 2001.
- [4] Gautier Y., Cremona C., Moretti O. “Experimental Modal Analysis of the Millau Bridge”. *Proceedings of the EVACES (Experimental Vibration Analysis for Civil Engineering Structures)*, 2005.
- [5] Cremona C., Barbosa F.S., Alvandi A. “Modal Identification under Ambient Excitation: Application to Bridge Monitoring”. *Revue Mécanique et Industries*, (4), pp. 259–271, 2003.
- [6] Cury A.A., Barbosa F.S., Borges C.C.H, Cremona C. “A Hybrid Technique of Damage Assessment Using Experimental Data and Computational Modeling.”. *Proceedings of the EVACES (Experimental Vibration Analysis for Civil Engineering Structures)*, 2007.
- [7] Altavista. “Tradutor Babelfish: <http://babelfish.altavista.com>”. Março/2008.
- [8] Google Inc. “Portal GoogleDocs: <http://www.google.com/docs>”. Março/2008.

- [9] Fonseca F.T., Pitagueira R.L.S. “Um Programa Gráfico Interativo para Modelos Estruturais de Barras”. Proceedings of the XV CILAMCE - Iberian Latin American Congress on Computational Methods in Engineering, 2004.
- [10] Câmara L.S. “Um Serviço Web para o Método dos Elementos Finitos”. Universidade Federal de Minas Gerais. Dissertação de Mestrado, 2007.
- [11] Clough R.W., Penzien J. “Dynamics of Structures”. 3rd Edition. Computers and Structures Inc., Berkeley, EUA, 2003.
- [12] Groehs A.G. “Mecânica Vibratória”. Editora Unisinos, São Leopoldo, Brasil, 2001.
- [13] Ewins D.J. “Modal Testing: Theory, practice and application”. Research Studies Press, 2000.
- [14] Asmussen J.C., Brinker R. “Estimation of Correlation Functions by Random Decrement”. Proceedings of the International Conference on Noise and Vibration Engineering, 1996.
- [15] Asmussen J.C., Brinker R., Ibrahim S.R. “Application of Vector Triggering Random Decrement”. Proceedings of the XV International Modal Analysis Conference, 1997.
- [16] Asmussen J.C. “Modal Analysis Based on the Random Decrement Technique - Application to Civil Engineering Structures”. Ph.D. Thesis, Aalborg University/Danemark, 1998.
- [17] Brincker R., Jensen J.L., Krenk S. “Spectral Estimation By The Random Decrement Technique”. Proceedings of the 9th International Conference on Experimental Mechanics, 1990.
- [18] Brincker R., Jensen J.L., Krenk S. “Estimation of Correlation Functions by the Random Decrement Technique”. Proceedings of the IMAC 9, pp. 14–18, 1991.

- [19] Brincker R., Jensen J.L., Krenk S. “Identification of Dynamical Properties from Correlation Function Estimates”. Proceedings of the IMAC 10, pp. 610–615, 1994.
- [20] Ibrahim S.R. “Random Decrement Technique for Modal Identification of Structures”. Journal of Spacecraft and Rockets, 14(11): pp. 696–700, 1977.
- [21] Ibrahim S.R. “The use of Random Decrement Technique for Identification of Structural Modes of Vibration”. AIAA Paper, 77: pp. 1–9, 1977.
- [22] Ibrahim S.R., Mikulcik E.C. “A Method for the Direct Identification of Vibration Parameters from the Free Responses”. Journal of Sound and Vibration, 113(1): pp. 47–57, 1987.
- [23] Ibrahim S.R. “An Upper Hessenberg Sparse Matrix Algorithm For Modal Identification On Microcomputers”. Journal of Sound and Vibration, pp. 183–198, 1977.
- [24] Cole H.A. “On-The-Time Analysis Of Random Vibrations”. AIAA Paper, pp. 68–288, 1968.
- [25] Cole H.A. “Failure Detection Of A Space Shuttle Wing By Random Decrement”. NASA, 1971.
- [26] Cole H.A. “On-Line Failure Detection And Damping Measurements Of Aerospace Structures By Random Decrement Signature”. NASA, 1973. CR-2205.
- [27] Alvandi A. “Contribution A L’Utilisation Pratique De L’Evaluation Dynamique Pour La Detection D’Endommagements Dans Les Ponts”. Ph.D. Thesis, Laboratoire National des Ponts et Chaussées/França, 2002.
- [28] Vandiver J.K., Campbell R.B., Cook M.F., Dunwoody A.B. “A Mathematical Basis For The Random Decrement Vibration Signature Analysis Technique”. Journal of Mechanical Design, 104: pp. 307–313, 1982.

- [29] Melsa J.L., Sage A.P. “An Introduction to Probability and Stochastic Processes”. Prentice-Hall, Inc., Englewood Cliffs, 1973.
- [30] Sonderstrom T., Stoica P. “System Identification”. Prentice-Hall International, Inc., Englewood Cliffs, 1989.
- [31] Deitel P.J., Deitel H.M. “Java: Como Programar”. Prentice-Hall, Inc., Englewood Cliffs, 2005.
- [32] Gonçalves E. “Desenvolvendo Aplicações Web com NetBeans IDE 5.5”. Ciência Moderna, São Paulo, 2007.
- [33] Servidor de aplicação Glassfish. “Sítio na internet: <http://glassfish.java.net>”. Março/2008.
- [34] Servidor de aplicação Tomcat. “Sítio na internet: <http://tomcat.apache.org>”. Março/2008.
- [35] Sun Microsystems. “Tutorial: Getting Started with Visual Web JSF Application Development”. Março/2008.
- [36] Microsoft. “Portal MSN Spaces: <http://spaces.live.com>”. Março/2008.
- [37] Ferreira A.D. “Comparativo entre as Tecnologias JSP (JavaServer Pages) e ASP (Active Server Pages)”. Trabalho de Final de Curso, UFRGS, 2002.
- [38] Java Magazine. “Navegação Estática e Dinâmica com JSF”. Outubro/2007.
- [39] Servidor de aplicação JBoss. “Sítio na internet: <http://www.jboss.org>”. Março/2008.
- [40] Barbosa F.S. “Identification Modale de Structures sous Sollicitation Ambiante”. Technical report. Laboratoire National des Ponts et Chaussées/France, 2001.

- [41] De Roeck G., Maeck J., Peeters B. “Damage Detection on a Prestressed Concrete Bridge using Dynamic System Identification”. Technical Report, 2000.
- [42] Deliverable A3. “Long-Term Monitoring and Bridge Tests”. Contract BRITE EURAM BRPR-CT96-0277. Technical report, EMPA, 1999.

Apêndice A

Neste apêndice são apresentados os arquivos de parâmetros “step1.txt” e “step2.txt” para os exemplos de aplicação descritos no Capítulo 6.

Arquivos de Parâmetros - Viga bi-apoiada

a) step1.txt

VFi

```
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26
27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49
50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70
```

ext

1

tfic

5

MVm

1 1 2

1 1 2

1 1 2

1 1 2

1 1 2

1 1 2

1 1 2

1 1 2

1 1 2

1 1 2
1 1 2
1 1 2
ref
1
varint
10 2 40
varreal
0.15 1 100000 0.002
flag1
4
flag2
2
nd
1
limxi
0 0.1

b) step2.txt

limfreq
6.7 25.8 56.7
6.9 26.4 57.3
tipofilt
1 2
td
1 1 1 1 1 2 2 2 2 2 3 3 3 3 3 4 4 4 4 4 5 5 5 5 5 6 6 6 6 6 7 7 7 7 7
8 8 8 8 8 9 9 9 9 9 10 10 10 10 10 11 11 11 11 11 12 12 12 12 12 13
13 13 13 13 14 14 14 14 14
visu
3 17

ap

3.3 10.3

0 11.3

0 12.3

0 13.3

0 14.3

0 15.3

0 1.3

0 2.3

0 4.3

0 5.3

0 6.3

0 7.3

0 8.3

0 9.3

Arquivos de Parâmetros - Ponte Férrea TGV

a) step1.txt

VFi

1 2 3 4 5 6 7 8 9 10 11 12

ext

2

tfic

5

MVm

2 2 4 5 6 8

2 2 4 5 6 8

2 2 4 5 6 8

2 2 4 5 6 8

2 2 4 5 6 8

2 2 4 5 6 8

2 2 4 5 6 8

2 2 4 5 6 8

2 2 4 5 6 8

2 2 4 5 6 8

2 2 4 5 6 8

2 2 4 5 6 8

ref

2

varint

15 2 51

varreal

0.016 1 100000 0.000244

flag1

4

flag2

2

nd

1

limxi

0 0.1

b) step2.txt

limfreq

5.0 11.4 17.0

5.6 13.0 19.0

tipofilt

1 2

td

1 2 3 4 5 6 7 8 9 10 11 12

visu

3 5

ap

4.3 5.3 6.3 8.3

Arquivos de Parâmetros - Ponte Z24

a) step1.txt

VFi

1 2 3 4 5 6 7 8 9

ext

1

tfic

4

MVm

2 2 6 7 8 9 10

2 2 6 7 8 9 10

2 2 6 7 8 9 10

2 2 6 7 8 9 10

2 2 6 7 8 9 10

2 2 6 7 8 9 10

2 2 6 7 8 9 10

2 2 6 7 8 9 10

2 2 6 7 8 9 10

ref

2

varint

20 2 51

varreal

0.012 1 100000 0.01

flag1

1

flag2

2

nd

1

limxi

0 0.1

b) step2.txt

limfreq

3.6 9.6 10.4 12.0

4.2 10.3 11.2 13.6

tipofilt

1 3

td

1 2 3 4 5 6 7 8 9

visu

4 45

ap

41.3 42.3 43.3 44.3 45.3

36.3 37.3 38.3 39.3 40.3

31.3 32.3 33.3 34.3 35.3

26.3 27.3 28.3 29.3 30.3

21.3 22.3 23.3 24.3 25.3

16.3 17.3 18.3 19.3 20.3

11.3 12.3 13.3 14.3 15.3

6.3 7.3 8.3 9.3 10.3

1.3 2.3 3.3 4.3 5.3