UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE ENGENHARIA DE CAMPINAS
DEPARTAMENTO DE MECÂNICA COMPUTACIONAL

Este exemplar corresponde à
Redação final defendida por
Pablo Siqueira Meirelles e aprovada
pela Comissão Julgadora
em 27/09/89

SIMULAÇÃO EXPERIMENTAL DE VIBRAÇÕES PARA TESTE DINÂMICO DE ESTRUTURAS COM NÃO LINEARIDADES

49/89

Pablo Siqueira Meirelles

Tese apresentada à Faculdade de Engenharia de Campinas - UNICAMP - como parte dos requisitos exigidos para a obtenção do título de Mestre em Engenharia Mecânica.

Campinas
- Setembro de 1989 -

UNIVERSIDADE ESTADUAL DE CAMPINAS

FACULDADE DE ENGENHARIA DE CAMPINAS

DEPARTAMENTO DE MECÂNICA COMPUTACIONAL

Tese de: MESTRADO

Título da tese: SIMULAÇÃO EXPERIMENTAL DE VIBRAÇÕES PARA TESTE DINÂMICO DE ESTRUTURAS COM NÃO LINEARIDADES.

Autor: Pablo Siqueira Meirelles

Orientador: José Roberto de França Arruda

Aprovado por

Prof. Dr. JOSÉ ROBERTO DE FRANÇA ARRUDA, presidente

Prof. Dr. AGENOR DE TOLEDO FLEURY

Prof. Dr. ANTONIO ARLINDO GUIDETTI PORTO

Campinas, 27 de Setembro de 1989.

A Lídia, Renata e Paula, que são o maior estímulo para tudo o que eu faço.

Aos meus pais, que perto ou longe sempre acreditaram em mim e me apoiaram em todos os meus empreendimentos.

AGRADECIMENTOS

Ao meu orientador, Prof. Dr. José Roberto de França Arruda, por ter viabilizado com o seu apoio a realização deste trabalho, e pelo exemplo de conduta profissional.

Ao Prof. Dr. Fernando Iguti, pelo auxílio prestado para o estudo dos métodos numéricos utilizados.

Ao Prof. Dr. Natanael Victor de Oliveira, pela presteza em fornecer todo o material e apoio ao seu alcance.

Ao colega e amigo Carlos Florian Heuberger, pelo socorro nas lutas contra os "mistérios" da computação.

Aos Departamentos de Projeto Mecânico e Mecânica Computacional, por terem fornecido toda a infraestrutura necessária para a realização deste trabalho.

À CAPES e CNPq, pelo imprescindível apoio financeiro.

A todos aqueles que, de uma forma ou de outra, contribuíram para a realização deste trabalho.

RESUMO

Este trabalho analisa as potencialidades e limitações de um método para a simulação de vibrações em estruturas. Com este propósito, foram efetuadas simulações em computador do comportamento dinâmico de estruturas, lineares ou não, e aplicou-se o método proposto para determinar as excitações que, se aplicadas na estrutura, provocariam respostas pré-determinadas na mesma. Estas experiências foram efetuadas para estruturas de 2 e 4 graus de liberdade, simuladas mediante um integrador que utiliza o método de Newmark, nas quais foram introduzidas não linearidades do tipo atrito de Coulomb, amortecimento proporcional ao quadrado da diferença de velocidade entre suas extremidades (amortecimento quadrático) e rigidez proporcional ao quadrado da variação da distância entre as suas extremidades (rigidez quadrática).

Para dar maior consistência aos resultados obtidos, foi montado um protótipo experimental simples, constituído por duas massas espaçadas por molas/amortecedores de borracha (não lineares), e no qual é possível alterar a influência das não-linearidades mudando o aperto de parafusos que fazem a ligação entre as duas peças.

Neste protótipo foram efetuadas várias experiências com o objetivo de mostrar a aplicabilidade do método na prática. As experiências efetuadas podem ser resumidas na obtenção dos sinais de excitação que conduzem a respostas semelhantes às obtidas no

protótipo quando este é submetido a diversas solicitações (batida, sequência de batidas, balançando aleatoriamente) e com diversas intensidades. Também foram alteradas as características de não linearidade do protótipo mudando o aperto dos parafusos de regulagem.

Os resultados obtidos evidenciam a viabilidade do método para a simulação experimental de vibrações em estruturas, lineares ou não, constituindo, portanto, uma importante ferramenta para a realização de testes. Sistemas baseados em princípios semelhantes a este são comercializados por empresas especializadas na área de testes de estruturas na forma de pacotes fechados de alto custo, acessíveis portanto apenas a empresas de grande porte. Este trabalho abre caminho para que o método possa ser utilizado por um número maior de usuários, se adaptando às necessidades e possibilidades de cada um.

Por outro lado, as experiências efetuadas mostraram, também, que o método não se aplica a todo e qualquer caso, podendo acontecer problemas para sistemas nos quais a participação dos efeitos não lineares é muito grande frente à participação da porção linear, assim como nos casos onde são requeridas amplitudes de sinal muito grandes, aumentando assim a influência relativa das não-linearidades.

As observações efetuadas permitiram adquirir significativa experiência com relação à metodologia que deve ser seguida, e precauções que devem ser tomadas, para que se obtenha sucesso na aplicação do método a um número maior de casos. Algumas destas precauções seríam: escolha adequada do tipo e amplitude dos sinais utilizados para identificação, seleção conveniente dos equi-

pamentos a serem utilizados e edição apropriada dos sinais desejados, a fim de que as conclusões extraídas dos resultados dos testes sejam compatíveis com as necessidades do usuário.

ABSTRACT

In this work, the possibilities and limitations of a method for the experimental simulation of vibrations on structures in the laboratory are investigated.

The method consists of determining the excitation forces that will make the structure vibrate in a predetermined way,which is usually the vibration measured "in situ" under normal or special operation conditions.

Numerically simulated data are used to investigate the method. The dynamic responses of linear and non-linear structures of 2 and 4 degrees of freedom are simulated. Different types of non-linear models are used, e. g. Coulomb damping, quadratic damping and quadratic stiffness.

Experimental data are also used to verify the possibilities of the practical implementation of the method. A simple structure consisting of two square steel plates connected through bolts and sandwiched with rubber is mounted on a shaker. The transient vibrations caused by manual shakes, are reproduced by synthetizing the appropriate signal for the shaker control.

The obtained results have shown that the method is feasible and can be used for the experimental simulation at the laboratory of the vibrations measured "in situ" on linear and non-linear structures.

The limitations of the method and the circumstances under which it might not work are also discussed based on the experimental and simulated results.

The results reported and discussed here should be useful information to someone in charge of implementing a vibration testing and simulation laboratory.

ÍNDICE

NOMENCLATURA

t - Variável tempo.

Δt - Incremento na variável tempo.

w - Variável frequência angular.

$\{x\}$ - Vetor coluna.

x_i - Elemento de índice i, do vetor $\{x\}$.

$[H(w)]$ - Matriz de resposta em frequência, MRF.

$H_{ij}(w)$ - Elemento da linha i, coluna j, da matriz $[H(w)]$.

$[H(w)]^t$ - Matriz transposta da matriz $[H(w)]$.

$[H(w)]^c$ - Matriz conjugada da matriz $[H(w)]$.

$[H(w)]^{-1}$ - Matriz inversa da matriz $[H(w)]$.

$S_{xx}(w)$ - Densidade espectral de potência, DEP, da variável $x(t)$.

$S_{xy}(w)$ - Densidade espectral cruzada das variãveis $x(t)$ e $y(t)$.

TRF - Transformada rápida de Fourier.

TRFI - Transformada rápida de Fourier inversa.

TFD - Transformada de Fourier discreta.

ITFD - Inversa da transformada de Fourier discreta.

$E[x]$ - Valor esperado da variável $x(t)$.

RPC - "Remote Parameter Control".

ITFC - "Iterative Transfer Function Compensation".

CESR - Correção das excitações por sensoreamento das respostas.

D/A - Digital-analógico.

A/D - Analógico-digital.

INTRODUÇÃO

Frequentemente as equipes de testes e/ou controle de qualidade se defrontam com o seguinte problema: é possível medir e registrar o nível de solicitações que alguns componentes sofrem quando uma estrutura está submetida a determinadas situações, mas não é geralmente possível medir diretamente quais são as excitações externas que conduzem a esses níveis de solicitações.

Um exemplo ilustrativo é o caso de um automóvel que trafega num determinado terreno, sob determinadas condições de carga, a uma determinada velocidade, efetuando determinadas manobras, etc.

É possível registrar as solicitações que acontecem em algumas peças ao longo do tempo (acelerações, deformações, deslocamentos, etc...) mas é muito difícil estabelecer quais foram os esforços externos (intensidade, direção e fases) atuantes sobre as rodas que originaram aquelas grandezas.

Em outras palavras, é possível medir o efeito, mas não a causa.

Mas existem hoje, e continuam sendo desenvolvidos, sistemas genericamente denominados "simuladores" que permitem, para muitos casos, determinar as solicitações que devem ser introduzidas nas estruturas em questão para que elas apresentem respostas semelhantes àquelas previamente observadas e registradas.

É fácil perceber as vantagens que podem ser obtidas com o uso de simuladores. Em [1] é salientado que o simulador de vários canais é um dos modernos recursos utilizados, na industria automobilística, para diminuir o tempo e o custo de desenvolvimento de novos produtos, representando uma alternativa para a simulação de situações com alto grau de fidelidade.

O autor sustenta, ainda em [1], que as simulações experimentais em laboratório têm suplantado os métodos clássicos de ensaios com sucesso, apresentando sobre aqueles as vantagens de permitir detetar e corrigir mais rapidamente os problemas.

Os métodos clássicos de ensaio aqui mencionados seríam, pelo menos no que diz respeito ao teste de autoveículos no Brasil, a realização de testes em estradas e em condições consideradas representativas do universo das solicitações que o veículo deverá enfrentar no desempenho das funções para as quais foi projetado. Outros métodos bastante utilizados, mas mais para componentes do que para estruturas completas, seriam os testes de laboratório utilizando atuadores que transmitam excitações harmônicas, excitações harmônicas variando amplitudes e frequências por blocos, excitações harmônicas com modulação em amplitude e frequências (varreduras) ou excitações aleatórias com densidades espectrais pré-definidas. Em [2] os autores fazem uma análise dos casos em que cada uma destas alternativas é aplicável, mas assim como em [1] e [3] se destaca a importância dos testes com excitações variáveis utilizando simuladores.

Em [1], [2] e [4] são mencionadas a redução de tempo e número de protótipos ensaiados até a validação de um novo projeto como sendo as principais vantagens obtidas com o uso de simulado-

res. Em [1] o autor atribui a estas virtudes o fato dos simuladores terem conseguido significativo desenvolvimento nas duas últimas décadas, desenvolvimento este impulsionado pela crescente competitividade das diversas montadoras e favorecido pelos progressos ocorridos nas técnicas de processamento digital dos sinais.

Conhecidas as entradas da estrutura, é possível efetuar os testes da mesma colocando atuadores em lugares apropriados, que a solicitem de forma similar àquela como ela é solicitada quando enfrenta as situações nas quais foram registradas as respostas.

Voltando ao exemplo do automóvel, o conhecimento dos esforços que agem nas rodas possibilita que os testes de comportamento e durabilidade da estrutura e os seus componentes no que diz respeito à vibração sejam efetuados dentro do laboratório, em condições rigorosamente controladas, com ambiente favorável para a instrumentação, onde os sensores ficam ligados a intrumentos estacionários (não embarcados), onde o acompanhamento dos testes pode ser facilmente efetuado por vários técnicos, e com alta repetitividade propiciada pelo melhor controle de fatores externos (mudanças climáticas, de condições de estrada, etc...)

Em [3] são descritas as dificuldades normalmente enfrentadas para definir as "entradas" que agem numa estrutura, entradas estas cujo conhecimento é de fundamental importância para que possam ser aplicados com sucesso métodos analíticos de estudo do desempenho de estruturas (por exemplo, elementos finitos). Estas entradas podem ser determinadas quando se utiliza um simulador, e por este motivo, em [1] e [3] é chamada a atenção para a vantagem de se associar o uso de simuladores a métodos analíticos.

Neste trabalho são analisadas algumas alternativas para a implementação de simuladores de estradas. Também são relatadas algumas tentativas já realizadas neste sentido, com maior ou menor sucesso.

Finalmente será proposto um algoritmo que pretende solucionar este problema, fazendo-se uma análise das suas possibilidades e limitações.

A referência [5] será frequêntemente citada neste trabalho por se tratar de um trabalho que aborda a maior parte dos tópicos concernentes a este problema.

O objetivo final deste trabalho é lançar a base para que possam ser desenvolvidos e empregados no Brasil sistemas de testes baseados neste princípio, que já é empregado por pelo menos duas grandes fábricas de equipamentos de testes: a MTS Sistems Corporation dos Estados Unidos da América do Norte [6], [7], [8], [9] e [10] e a Carl Schenck AG da República Federal da Alemanha [11], [12], [13], [14], [15], [16] e [17].

Para ser coerente com seus objetivos, esta tese terá um caráter tão prático e aplicado quanto possível.

CAPÍTULO I

ANÁLISE DO PROBLEMA

I.1) Alguns antecedentes.

O problema aqui tratado é o de como simular no laboratório uma determinada situação registrada, em termos de vibração.

Algumas tentativas neste sentido foram feitas pela indústria automobilística, com maior ou menor sucesso.

A seguir será feita uma breve resenha de algumas destas tentativas.

A resposta mais imediata a esta pergunta frequentemente conduz a tentar levantar o perfil da estrada por onde o veículo transita, e logo impor às rodas do carro um deslocamento vertical que siga esse perfil.

Uma possibilidade seria construir cames com o perfil da estrada desejado e rolá-los sob as rodas do carro. Possivelmente neste caso a maior dificuldade consista em escorar o carro sem interferir significativamente no seu comportamento.

Outra alternativa mais racional é colocar atuadores hidráulicos ou eletrodinâmicos sob as rodas do carro, controlados em deslocamento para fornecer o movimento desejado.

Algumas tentativas interessantes neste sentido foram realizadas pela General Motors Corporation, e são relatadas em [5].

O primeiro passo consiste em tentar levantar o perfil da estrada desejada. Para isso os engenheiros da GMC optaram primeiramente por construir um equipamento especialmente projetado para esse fim.

O levantador de perfil de estradas por eles construido e que é descrito na citada referência é constituido por uma pequena roda acoplada a um carro, com instrumentação adequada para gerar uma tensão DC proporcional ao deslocamento vertical da mesma quando transita em baixa velocidade pela estrada a ser simulada.

A tensão registrada era utilizada para gerar o sinal de comando fornecido a atuadores hidráulicos, controlados em deslocamento, instalados debaixo das rodas do veículo a ser testado, respeitando a defasagem dos sinais atuantes nas rodas anteriores e posteriores, que é função da velocidade do veículo.

A baixa correlação verificada entre a vibração medida no veículo quando transitando pelo caminho em questão e a obtida no mesmo veículo montado no simulador de estrada construido dessa forma foi atribuida fundamentalmente a três problemas:

1) Não foi levada em consideração a deformação dos pneus quando em contato com o perfil da estrada.

2) O seguidor do levantador de perfis e as rodas do carro não transitam exatamente pelo mesmo lugar, fenômeno este citado na referência [5] como "tracking error".

3) Não foi levada em consideração a rigidez do terreno, e portanto a sua própria deformação.

Provavelmente este último aspecto só seja significativo para veículos muito pesados ou terrenos não pavimentados (pouco rígidos).

Uma segunda tentativa foi feita instrumentando o carro para medir a força e aceleração nos eixos enquanto o veículo transitava a velocidade normal pela pista. Fazendo-se essas medições simultaneamente consegue-se eliminar o "tracking error".

Os valores de força e aceleração registrados foram fornecidos a um computador analógico no qual foi simulado o comportamento dos pneus. Desta forma foi obtido um resultado bastante mais satisfatório, especialmente para pequenas amplitudes, mas ainda longe das expectativas.

Um fator que sem dúvida deve ter contribuido significativamente para a obtenção ainda de uma baixa correlação entre a vibração obtida na pista e a obtida no simulador foi o fato do modelo utilizado para prever o comportamento dos pneus não ter levado em conta o comportamento não linear dos mesmos.

Um outro fato importante e que não foi levado em consideração é que a deformação do pneu é função não apenas dos esforços atuantes sobre ele, mas também do formato da superfície sob ele (arestas, etc...), mesmo assumindo que as caraterísticas do pneu sejam sempre as mesmas.

Uma dificuldade adicional desse método é a necessidade de se construir uma célula de carga especial para medir os esforços que as rodas transmitem aos eixos.

Uma análise da situação a esta altura levou as pessoas empenhadas neste trabalho a concluir que pouco progresso seria obtido sofisticando mais e mais o modelo, trabalhando sempre no mesmo

princípio.

Visto que a força que é transmitida das rodas aos eixos precisava ser medida para se aplicar o método descrito acima, é possível que um resultado melhor fosse obtido utilizando-se controle em força, e não em deslocamento, para os atuadores. Mesmo assim não seria obtido o resultado desejado, primeiro porque a célula está registrando inclusive as forças causadas pela inércia das rodas, e segundo, porque a resposta dos atuadores é função da frequência e da amplitude, e portanto a força obtida não é exatamente proporcional ao sinal de comando quando trabalhamos com um sinal rico em frequências. Este "desvio" de comportamento pode também ser corrigido utilizando-se o método que é estudado mais adiante.

Uma das experiências que podem ser extraídas desta tentativa é a necessidade, ou pelo menos conveniência, de se utilizar um procedimento no qual seja necessário conhecer apenas aquelas grandezas que podem ser mensuráveis com alguma precisão, como é o caso das acelerações, deformações, deslocamentos, etc..., que acontecem em alguns pontos do sistema que está sendo estudado.

De posse dessa informação, deve-se determinar quais são as "entradas" que devem ser fornecidas à estrutura para que ela se comporte da forma desejada.

O problema se resume então em como determinar essas "entradas", que neste trabalho serão frequentemente chamadas excitações externas.

I.2) Caso linear.

Quando estamos trabalhando com sistemas aos quais pode ser atribuído um comportamento linear sem cometer grandes erros, o problema pode ser resolvido de forma conceitualmente simples. Basta trabalhar com a matriz de resposta em frequência (MRF).

Seja por exemplo {f(t)} um vetor de funções do tempo que representa os esforços que solicitam uma estrutura linear. Cada coordenada do vetor corresponde à solicitação num ponto e/ou direção diferente.

Seja {x(t)} o vetor dos deslocamentos através do tempo correspondentes aos esforços {f(t)}. Cada coordenada de {x(t)} representa o deslocamento num ponto e/ou direção diferente.

Será utilizada a letra $\mathcal{F}$ para identificar a transformada de Fourier (Apêndice 1), e letras maiúsculas para identificar variáveis no domínio da frequência.

Portanto podemos dizer que:

$$\{X(w)\} = \mathcal{F}(\{x(t)\}) \qquad (I.1)$$

$$\{F(w)\} = \mathcal{F}(\{f(t)\}) \qquad (I.2)$$

onde w é a variável frequência angular e t é a variável tempo.

Valem também as relações inversas:

$$\{x(t)\} = \mathcal{F}^{-1}(\{X(w)\}) \qquad (I.3)$$

$$\{f(t)\} = \mathcal{F}^{-1}(\{F(w)\}) \qquad (I.4)$$

onde $\mathcal{F}^{-1}$ representa a função transformada de Fourier inversa (Apêndice 1).

É chamada de matriz das funções de resposta em frequência a matriz $[H(w)]$, que estabelece a relação existente entre os esforços $\{F(w)\}$ e os deslocamentos $\{X(w)\}$ para sistemas lineares, ou seja:

$$\{X(w)\} = [H(w)].\{F(w)\} \qquad (I.5)$$

A matriz $[H(w)]$ pode ser obtida experimentalmente para qualquer estrutura linear invariante no tempo, como veremos mais para a frente.

De posse da matriz de resposta em frequências $[H(w)]$ e do vetor dos deslocamentos desejados (medidos) $\{X(w)\}$ podemos calcular o vetor dos esforços $\{F(w)\}$ procurado (que conduzem aos deslocamentos $\{X(w)\}$) resolvendo a equação (I.5) para obter o vetor $\{F(w)\}$ (equação (I.6)).

$$\{F(w)\} = [H(w)]^{-1}.\{X(w)\} \qquad (I.6)$$

Uma vez obtido o vetor dos esforços no domínio das frequências $\{F(w)\}$ podemos obter o vetor dos esforços $\{f(t)\}$ no domínio do tempo aplicando a transformada de Fourier inversa $\mathcal{F}^{-1}$, como é mostrado na equação (I.4).

A equação (I.6) só é valida para sistemas com igual número de entradas e saidas, ou seja, para matrizes $[H(w)]$ quadradas

Para o sistema da equação (I.5) ser determinado é necessário

(não suficiente) que o número de incógnitas seja menor ou igual que o número de variáveis do vetor dos elementos conhecidos. Portanto é possível se trabalhar com sistemas com número de respostas (conhecidas) $X_i(w)$ maior do que o número de entradas (incógnitas) $F_j(w)$. Neste caso a matriz [H(w)] da equação (I.5) não é quadrada, e portanto não pode ser invertida, motivo pelo qual não é valida a equação (I.6). Por esse motivo deve-se utilizar outra forma para se obter a solução da equação (I.5). Um dos artifícios possíveis para este fim é o de se multiplicar ambos os lados da equaçao (I.5) pela matriz transposta da MRF, $[H(w)]^t$. As equações obtidas utilizando-se este procedimento são mostradas nas equações (I.7) a (I.9), onde (X) é o vetor conhecido, de dimensão m, (F) é o vetor que desejamos calcular, de dimensão n, e [H] é a matriz retangular de dimensões mxn, com m) n, também conhecida, que relaciona os dois. Portanto o sistema pode ser escrito:

$$(X)_{(mx1)} = [H]_{(mxn)} \cdot (F)_{(nx1)} \qquad (I.7)$$

Pré-multiplicando por $[H]^t$ dos dois lados da equação (I.7) obtem-se:

$$[H]^t_{(nxm)} \cdot (X)_{(mx1)} = [H]^t_{(nxm)} \cdot [H]_{(mxn)} \cdot (F)_{(nx1)} \qquad (I.8)$$

Pré-multiplicando dos dois lados da equação (I.8) pela matriz inversa da matriz quadrada obtida, isola-se o vetor das incognitas procurado (equação (I.9)).

$$(F) = ([H]^t_{(nxm)} \cdot [H]_{(mxn)})^{-1} \cdot [H]^t_{(nxm)} \cdot (X)_{(mx1)} \qquad (I.9)$$

O resultado expresso pela equação (I.9) é o que seria obtido aplicando o critério de mínimos quadrados para obter a solução ótima do sistema ([18], pg. 39, Estimador por Mínimos Quadrados).

Portanto deve-se utilizar a equação (I.6) para sistemas com igual número de entradas e saídas e a equação (I.9) para os casos com maior número de saídas do que de entradas.

Desta forma estaria resolvido o problema para os casos lineares. Bastaria conseguir que os atuadores transmitissem à estrutura, nos pontos adequados, os mesmos esforços (f(t)) calculados. Para isto bastaria fornecer aos atuadores um sinal de comando proporcional ao (f(t)) obtido, se for correto assumir que o comportamento dos atuadores é também linear e que o ganho dos mesmos é constante para todas as frequências da faixa de operação requerida. Já foi dito que frequentemente não é isso que acontece quando trabalhamos com faixas largas de amplitudes e frequências.

É por isto que geralmente faz-se necessário analizar o problema com mais detalhes mesmo quando se trata de estruturas lineares, onde a solução é matematicamente simples.

A MRF referida estabelece, na sua concepção mais usual, a relação existente no domínio das frequências entre os esforços que agem numa estrutura linear invariante no tempo, e os deslocamentos causados por esses esforços. Mas este conceito é extensível à relação entre outras grandezas que não sejam unicamente forças e deslocamentos, sempre que entre as grandezas envolvidas exista uma relação causa/efeito regida por um comportamento linear. Por exemplo, as entradas do sistema podem ser, além de forças, deslocamentos impostos em pontos da estrutura, tensão dos

sinais de comando enviados aos atuadores, etc... Já as respostas podem ser, além dos deslocamentos, acelerações, deformações em componentes da estrutura, etc...

Por causa desta ambiguidade as solicitações impostas a um sistema serão genericamente denominadas excitações, enquanto que, para os efeitos por elas ocasionados, utilizar-se-á a denominação genérica de respostas.

I.3) Caso não linear.

No caso linear a amplitude registrada numa determinada estação de medida para uma dada frequência é uma combinação linear das amplitudes de todas as excitações da mesma frequência (princípio da superposição).

Isso não acontece nos sistemas não lineares (e muitos sistemas devem ser tratados como tais), onde a amplitude da resposta obtida em qualquer estação para uma determinada frequência é o resultado da influência da excitação harmônica correspondente e das sub-harmônicas presentes nos sinais de excitação, e o resultado é diferente do que seria obtido por simples adição das respostas correspondentes a cada excitação independentemente. Ainda mais, mesmo que os sinais de excitação só contivessem uma componente em frequência, a resposta naquela frequência não poderia ser obtida por superposição linear dos efeitos de cada uma das entradas, e em geral aparecem respostas com componentes em frequência que são harmônicas de ordem superior da frequência de excitação.

É por isso que não se define uma função de resposta em frequência para sistemas não lineares, pois ela não seria única.

É para contornar este problema que se propõe um método que permita, após algum tratamento, estimar com uma precisão pré-estabelecida quais são as excitações que devem ser impostas em pontos escolhidos de uma estrutura não linear para que ela se comporte da forma desejada.

I.4) Método proposto.

O método proposto para contornar o problema mencionado consiste em, inicialmente, tratar a estrutura não linear como se ela fosse linear, estimando uma pseudo-matriz de resposta em frequência $[H'(w)]$. Por simplicidade, a pseudo MRF estimada será chamada apenas $[H(w)]$.

Utilizando esta matriz num processo semelhante ao descrito para sistemas lineares obteremos uma estimativa do sinal de excitação procurado. Esta estimativa carrega um erro decorrente de ter-se assumido que a estrutura tem um comportamento linear, e que é portanto função do grau de não linearidade da estrutura estudada.

Como será visto no exemplo 5 do capítulo IV, a influência das não linearidades pode ser minimizada utilizando um sinal de excitação adequado e com amplitudes convenientes para realizar a identificação do sistema. Na referência [19] também se alerta para este fato.

Seja por exemplo $(X(w))$ o vetor dos sinais desejados, em frequência, e $(F(w))$ o vetor das grandezas que devem ser fornecidas à estrutura para obter as respostas $(X(w))$.

Vale a relação:

$$[H(w)]^{-1}.(X(w)) = (F(w))+(dF(w)) = (F_1(w)) \quad (I.10)$$

onde $(dF(w))$ é o vetor erro de $(F(w))$ que aparece devido à não-linearidade da estrutura.

Excitando-se o sistema com o sinal $\{f_1(t)\}$, correspondente no tempo do sinal estimado em frequência $\{F_1(w)\}$, obteremos uma resposta que também carrega o erro decorrente de termos admitido a hipótese de sistema linear.

Vamos supor que a resposta obtida seja, em frequência, $\{X(w)+d_1X(w)\}$.

Utilizamos o erro medido para calcular uma correção da excitação anterior.

$$\{F_2(w)\} = \{F_1(w)\} - [H(w)]^{-1}.\{d_1X(w)\} \quad (I.11)$$

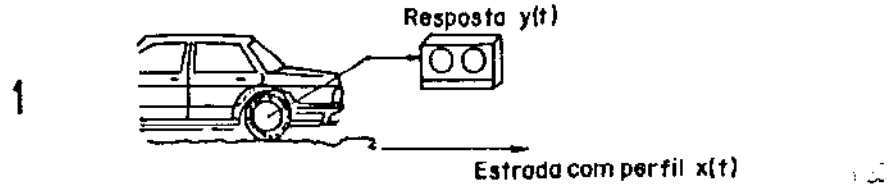
Com o novo sinal de excitação, $\{f_2(t)\}$, repetimos o processo anterior, o que nos levará a obter uma resposta mais próxima da desejada. Este procedimento é repetido num processo iterativo até obtermos um erro (diferença entre o sinal desejado e o obtido) considerado suficientemente pequeno para que se possa conviver com ele sem desvirtuar os resultados dos testes.

Este é basicamente o raciocínio utilizado pela MTS e Schenck, como é mostrado na figura I.1, inspirada em um esquema semelhante existente na referência [6].

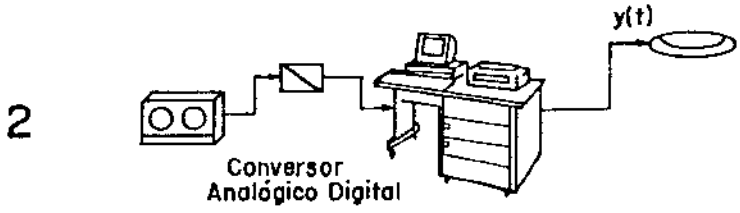
Os passos indicados na figura são:

1. Gravação das respostas da estrutura na situação que se deseja repetir no simulador.

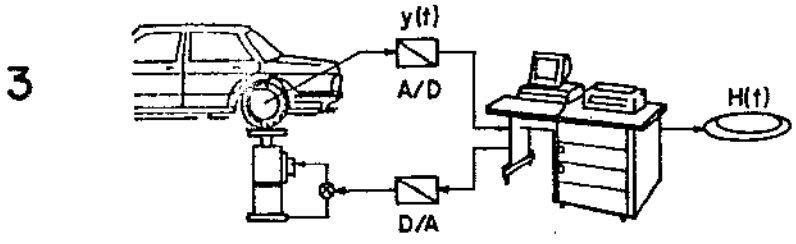
2. O sinal é digitalizado e armazenado na memória do computador.



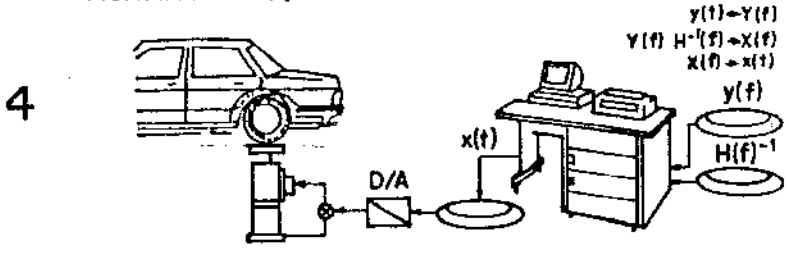
GRAVAR O SINAL EM CAMPO



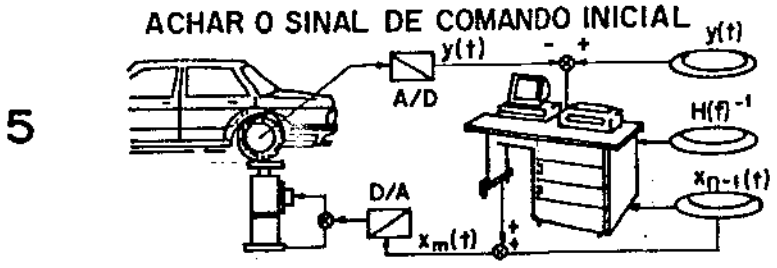
DIGITALIZAR O SINAL GRAVADO



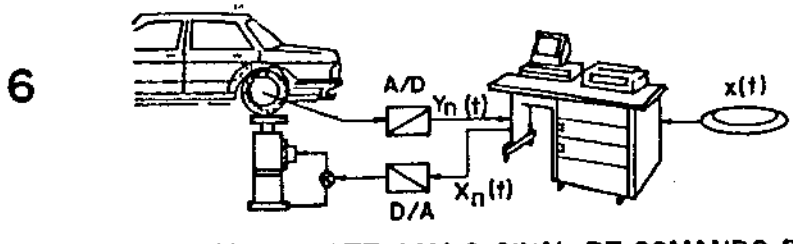
ACHAR A FUNÇÃO DE RESPOSTA EM FREQUÊNCIA



ACHAR O SINAL DE COMANDO INICIAL



AJUSTAR ITERATIVAMENTE O SINAL DE COMANDO



EFETUAR O TESTE COM O SINAL DE COMANDO RESULTANTE

Fig I.1: Visualização esquemática do método.

3. é feita a identificação do sistema (cálculo da matriz [H(w)]).

4. Com a resposta gravada e a matriz identificada calcula-se a primeira aproximação para o sinal de comando.

5. Mede-se o erro na resposta (diferença entre a resposta desejada e a obtida), e com o sinal de erro obtido calcula-se um novo sinal de comando. Este processo é repetido tantas vezes quantas forem necessárias para se obter uma resposta que seja tão semelhante com a desejada quanto se queira.

6. Efetua-se o teste com o sinal de comando calculado nos passos anteriores.

é inspirado nesta proposta que serão desenvolvidos os algoritmos e programas apresentados neste trabalho, mas não é esta a única proposta possível, como é mostrado no item seguinte.

Este procedimento para achar as entradas que vão dar as respostas desejadas numa estrutura não linear é conhecido como "Remote Parameter Control" (RPC) (nome utilizado pela MTS) [2], [3], [4] e [20], ou "Iterative Transfer Function Compensation" (ITFC) (nome dado pela Schenck) [14]. Neste trabalho, este processo será chamado "Correção das Excitações por Sensoreamento das Respostas" (CESR), por ser este um nome em português que faz alusão à idéia utilizada.

A MTS já comercializa a 3ª geração de simuladores para testes que operam de acordo com este princípio [10].

I.5) Outras alternativas.

O procedimento descrito sucintamente no item anterior sugere a possibilidade de se reproduzir com uma certa exatidão a situação em que uma determinada estrutura se encontrava quando foram efetuados registros de certas grandezas medidas em alguns pontos da estrutura.

Isto nos possibilita avaliar o desempenho da estrutura quando enfrenta aquela situação específica.

Mas em geral as solicitações a que as estruturas estão sujeitas carregam uma boa dose de aleatoriedade, e portanto os sinais deveriam receber um tratamento estatístico e não determinístico como acontece no caso anterior.

Neste caso, deveríamos tentar reproduzir as densidades espectrais de potência, DEP, dos sinais registrados, que pela própria definição (Apêndice 1) contém a informação estatística necessária.

Exemplificando novamente com o caso de veículos rodoviários, que é a área na qual se acha referências mais frequentes a este método, a aplicação da opção referida no item I.4 permite reproduzir o comportamento de um veículo numa determinada estrada, numa situação específica, e para um determinado registro.

Em compensação o método estatístico ao qual se faz referência neste item permite avaliar o comportamento da estrutura para um determinado "tipo" de estrada e de situações [5].

Não obstante isto, foi implementado neste trabalho o método do item anterior (tratando os sinais como se eles fossem determi-

nísticos), devido ao fato de que ele permite que o sinal registrado passe por uma série de tratamentos, genéricamente mencionados em algumas referências como "edição do sinal", que conferem ao método uma grande versatilidade e praticamente compensam as eventuais vantagens que o método das densidades espectrais de potência poderia apresentar.

A edição dos sinais pode variar desde a simples seleção de trechos mais representativos que são juntados num só bloco para serem reproduzidos, até o processamento estatístico prévio dos sinais desejados.

No entanto, apesar de não ter sido implementado o método das densidades espectrais, o procedimento para utiliza-lo é descrito a seguir.

Primeiro calcula-se a DEP das respostas obtidas quando a estrutura é submetida às excitações externas que se deseja simular. Para isto a excitação deve acontecer durante um intervalo de tempo suficientemente longo, para que possa ser identificado o "tipo" da resposta via DEP.

De acordo com a definição de DEP apresentada no apêndice 1, a matriz das DEP das respostas que desejamos obter é:

$$S_{XX}(w) = E[(X(w)).(X(w))^{ct}] \qquad (I.12)$$

onde o c indica complexo conjugado e o t matriz transposta.

Tambem é verdade que, pela definição de DEP:

$$S_{ff}(w) = E[(F(w)).(F(w))^{ct}] \qquad (I.13)$$

onde {F(w)} é o vetor das excitações externas (desconhecidas) atuantes na estrutura quando foi obtida a resposta {X(w)}.

Substituindo a equação (I.5) em (I.12) obtém-se:

$$S_{XX}(w) = E[[H(w)].\{F(w)\}.([H(w)].\{F(w)\})^{ct}] \quad (I.14)$$

onde o E indica esperança matemática (valor esperado) da expressão entre colchetes.

Os elementos da MRF são dependentes unicamente de w, e portanto pode ser tirada para fora dos colchetes que delimitam o alcance da esperança matemática. Isto não acontece com as excitações {F(w)} por causa da sua natureza aleatória.

Como $([H].\{F\})^{ct} = \{F\}^{ct}.[H]^{ct}$ a equação (I.14) pode ser escrita:

$$S_{XX}(w) = [H(w)].E[\{F(w)\}.\{F(w)\}^{ct}].[H(w)]^{ct} \quad (I.15)$$

Mediante substituição da equação (I.13) na equação (I.15) percebe-se que esta é equivalente a:

$$S_{XX}(w) = [H(w)].S_{FF}(w).[H(w)]^{ct} \quad (I.16)$$

Pré-multiplicando por $[H(w)]^{-1}$ e pós-multiplicando por $([H(w)]^{ct})^{-1}$ dos dois lados da equação (I.16) conseguimos calcular a DEP das excitações, $S_{ff}(w)$, a partir do conhecimento das DEP das respostas, $S_{XX}(w)$, e da MRF [H(w)], como pode ser observado na equação (I.17).

$$S_{ff}(w) = [H(w)]^{-1} . S_{xx}(w) . ([H(w)]^{ct})^{-1} \qquad (I.17)$$

Para evitar ter que efetuar mais do que uma inversão de matrizes pode-se utilizar a propriedade da álgebra de matrizes que estabelece que $([H]^{ct})^{-1} = ([H]^{-1})^{ct}$.

Uma vez obtida a matriz DEP das excitações $S_{FF}(w)$ mediante a equação (I.17), resta agora gerar o sinal para os atuadores, ou seja, calcular um vetor $\{F(w)\}$ que tenha DEP igual à $S_{FF}(w)$ achada.

Seja $\{W(w)\}$ um vetor formado por N ruídos brancos independentes e com DEP = 1. Portanto $S_{WW}(w) = [I]$, sendo $[I]$ a matriz identidade de dimensões NxN. Procura-se uma matriz de transformação $[T(w)]$ de coeficientes constantes (funções apenas de w) tal que:

$$\{F(w)\} = [T(w)] . \{W(w)\} \qquad (I.18)$$

Calculando as DEP dos dois lados da equação (I.18) chega-se a:

$$S_{ff}(w) = [T(w)] . S_{ww}(w) . [T(w)]^{ct} \qquad (I.19)$$

Como já foi dito $S_{ww}(w) = [I]$ devido à forma como foram gerados os ruídos brancos do vetor $\{W(w)\}$, e portanto a equação (I.19) pode ser escrita simplesmente:

$$S_{ff}(w) = [T(w)] . [T(w)]^{ct} \qquad (I.20)$$

Como $S_{ff}(w)$ já foi calculado antes, a matriz $[T(w)]$ pode ser achada se for imposta a condição de que $[T(w)]$ seja triangular

com elementos reais na diagonal principal [5].

Uma vez achada a matriz de transformação [T(w)] pode-se calcular o vetor das excitações (F(w)) por substituição na equação (I.18), e chega-se ao vetor das excitações no domínio do tempo (f(t)) mediante a transformada de Fourier inversa da equação (I.4). Pode-se observar que o vetor das excitações (f(t)) determinado por este procedimento é uma realisação de um processo estocástico que foi amostrado e truncado com duração finita, e portanto é determinístico (periódico), apesar de ter sido obtido a partir das características estatísticas das respostas.

Efetuando-se o teste com o sinal de excitação obtido desta forma obtém-se uma DEP das respostas que pode diferir da desejada por causa das não linearidades do sistema, que foram desprezadas para efetuar o cálculo da MRF. Se isto acontecer utiliza-se a matriz da diferença entre a DEP desejada e a obtida na saída do sistema para calcular uma correção da excitação num processo idêntico ao anterior, e assim sucessivamente até chegar suficientemente perto do resultado desejado.

Pode-se observar que o vetor (W(w)) não é único com as condições impostas, e para diferentes vetores de ruídos brancos independentes e de DEP = 1 obtém-se diferentes vetores de excitação (F(w)), mas todos eles satisfazem a condição de ter DEP igual ao $S_{ff}(w)$ calculado antes.

Para aplicar este método, assim como também para o tratado no item anterior, é preciso previamente ter calculado a MRF do sistema, deixando as não-linearidades de lado, o que introduz na maior parte dos casos erros que devem ser minimizados mediante repetição do processo para corrigir os sinais de excitação.

I.6) Algumas opções para implementar o algoritmo do item I.4.

Para efetuar a correção do sinal de excitação no processo iterativo descrito no item I.4 existem vários caminhos, como é motrado na figura I.2. É que tanto a diferença entre o sinal desejado e o obtido como a adição da correção ao sinal de excitação podem ser feitas indistintamente no domínio do tempo ou da frequência.

Portanto, todos os caminhos mostrados são conceitualmente análogos. O programa implementado e apresentado neste trabalho segue o caminho identificado na figura I.2 com traço contínuo.

Para implementar eficientemente os algoritmos mencionados antes é necessário utilizar computação digital.

Para poder ser utilizado na prática, um sistema que funcione neste princípio deve consistir no mínimo de:

-unidade central de processamentos (CPU do computador) com os periféricos de armazenamento de dados e de comunicação, inclusive terminal gráfico e "plotter" ou impressora gráfica.

-unidade de aquisição de dados (conversor analógico/digital, CA/D).

-saída analógica (conversor digital/analógico, CD/A).

-atuadores (fornecem as "entradas", tambem chamadas aqui de excitações externas, para a estrutura que está sendo ensaiada).

-sensores (transdutores) para ler as "saídas", ou respostas, da estrutura.

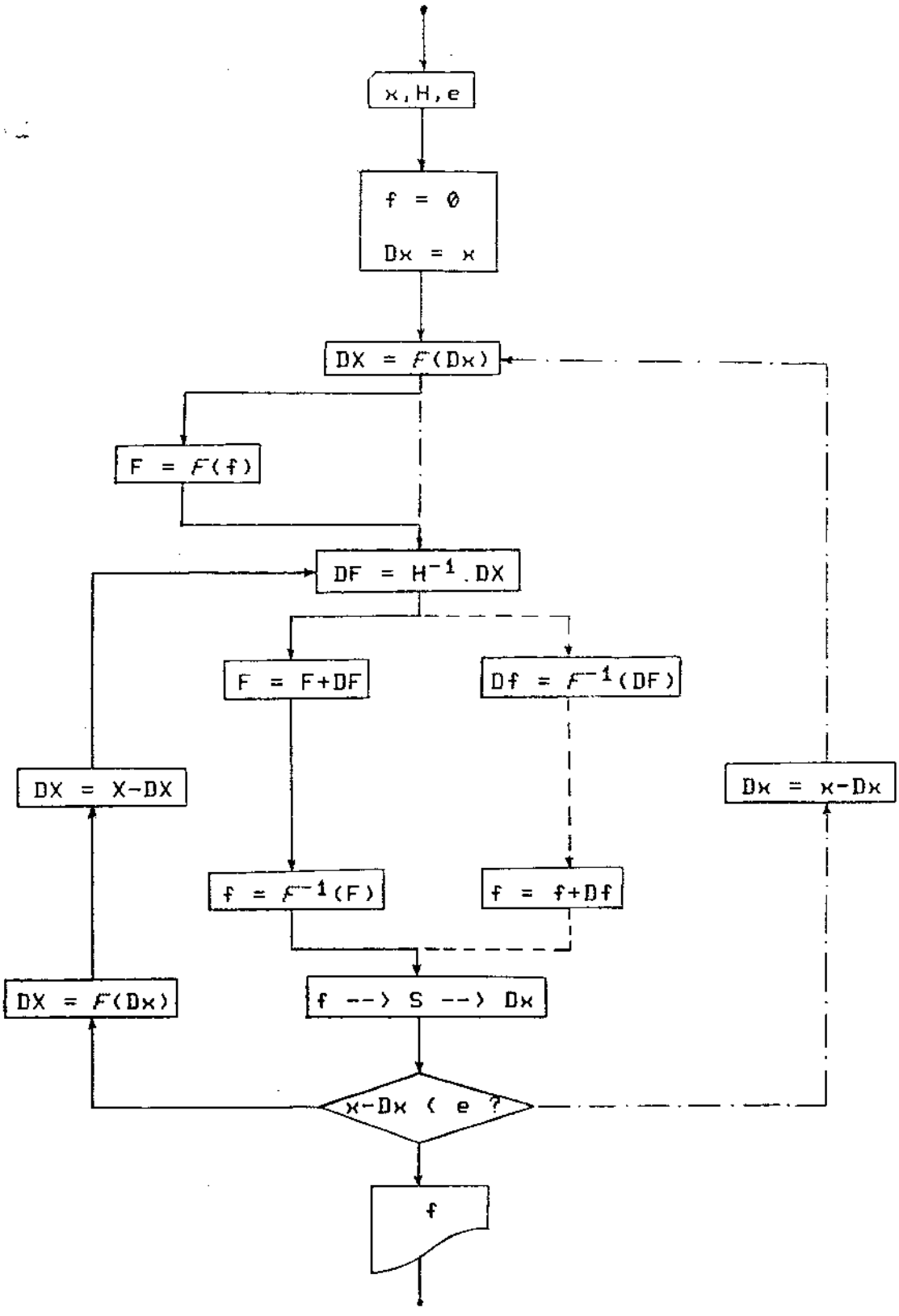


Fig. I.2: Diversas opções para montar o algoritmo que efetua a correção proposta no item I.4.

CAPÍTULO II

IDENTIFICAÇÃO DA ESTRUTURA

II.1) Introdução.

Todas as opções existentes para implementar os algoritmos mencionados no capítulo anterior passam por um processo de identificação da estrutura que queremos ensaiar.

A expressão identificação é definida por Zadeh (1962) como o processo de determinação, baseada na observação das entradas e saídas, de um sistema pertencente a uma determinada classe de sistemas, ao qual o sistema em estudo é equivalente [18].

Em particular é necessário neste trabalho determinar a Matriz de Resposta em Frequência (MRF), que estabelece uma correspondência biunívoca entre as entradas e as saídas de um sistema linear invariante no tempo, no dominio da frequência. A MRF pode também ser interpretada como sendo a transformada de Fourier da resposta do sistema ao impulso unitário [21], [22].

Cabe lembrar que se está trabalhando com estruturas em geral não lineares, às quais não pode ser associada de forma rigorosa uma matriz de resposta em frequência.

O artifício consiste, como já foi dito, em "esquecer" que o comportamento da estrutura está influenciado pela não linearidade

e calcular a função de resposta em frequência tal como se ela fosse linear. Ao contrário do que acontece quando se trabalha com sistemas com os quais assume-se que a hipótese de linearidade é satisfatória, agora se sabe que está sendo cometido um erro significativo no modelo matemático do sistema.

Deve-se lembrar que para diferentes processos de identificação obtém-se diversas MRF, por causa das não linearidades. Portanto deve-se utilizar um processo que leve a um resultado conveniente para a situação que se deseja estudar. Em [19] recomenda-se, por exemplo, que se efetue a identificação utilizando sinais de perturbação que ocasionem respostas de níveis próximos dos que se deseja reproduzir, minimizando assim as distorções ocasionadas pelas não linearidades. Para se verificar a validade desta afirmação foram efetuados alguns testes, cujos resultados serão apresentados no exemplo 5 do capítulo IV.

Existem muitas técnicas que permitem identificar experimentalmente uma estrutura, algumas delas bastante elaboradas e que devem conduzir a resultados mais precisos, tais como a análise modal experimental, e outras mais simples e de aplicação mais fácil mas igualmente satisfatória em muitos casos. Já que o algoritmo estudado tem por objetivo corrigir os erros cometidos na identificação do sistema, não parece muito sensato utilizar técnicas muito sofisticadas para esse fim. Não obstante isto, deve-se ter presente que uma identificação mais realista do sistema facilita a convergência para a solução procurada, e em alguns casos críticos pode possibilitar uma convergência que de outra forma seria impossível.

II.2) Método com excitação aleatória.

Na referência [5] é proposto o método descrito a seguir para calcular a MRF de uma estrutura, que pode ser visto também em [22] e [27].

Aplica-se à estrutura um conjunto de N solicitações aleatórias e independentes {f(t)}. Como consequência obtem-se na saída um conjunto de respostas {x(t)}, também de dimensão N.

Supondo que a estrutura tem um comportamento linear, vale a equação:

$$\{X(w)\} = [H(w)].\{F(w)\} \qquad (II.1)$$

onde [H(w)] é a matriz NxN que queremos achar.

Pós-multiplica-se ambos os lados da equação (II.1) por $\{F(w)\}^{ct}$. Em seguida acha-se os valores esperados dos dois lados da equação.

Assumindo que a matriz [H(w)] tem termos constantes para cada frequência pode-se coloca-la em evidência no lado direito da equação, que fica então:

$$E[\{X(w)\}.\{F(w)\}^{ct}] = [H(w)].E[\{F(w)\}.\{F(w)\}^{ct}] \qquad (II.2)$$

Lembrando a definição de Densidade Espectral de Potência (DEP, Apêndice 1) pode-se substituir $S_{xf}(w)$ e $S_{ff}(w)$ dos lados esquerdo e direito respectivamente da equação (II.2), que fica agora:

$$S_{xf}(w) = [H(w)].S_{ff}(w) \qquad (II.3)$$

Observe-se que pela própria definição de DEP $S_{ff}(w)$ e $S_{xf}(w)$ são matrizes NxN. Portanto se $S_{ff}(w)$ é não singular pode-se calcular $[H(w)]$ pós-multiplicando ambos os lados da equação (II.3) pela matriz inversa de $S_{ff}(w)$, resultando:

$$[H(w)] = S_{xf}(w).(S_{ff}(w))^{-1} \qquad (II.4)$$

A matriz $S_{ff}(w)$ é em geral não singular se os sinais de excitação F(w) não são todos completamente dependentes entre si.

Isto implica em que na aplicação prática deste método deve-se tomar algumas precauções para que não exista interação do sinal de um atuador em outro (eles devem, por exemplo, estar montados sobre bases individuais isoladas umas das outras).

Na referência [27] pode ser visto que se em lugar de multiplicar a equação (II.1) por $(F(w))^{ct}$ esta fosse multiplicada por $(X(w))^{ct}$ e se repetisse todo o procedimento anterior, se chegaria a:

$$[H(w)] = S_{xx}(w).(S_{xf}(w))^{-1} \qquad (II.5)$$

Portanto este é mais um procedimento que pode ser utilizado para o cálculo de $[H(w)]$.

As matrizes $[H(w)]$ identificadas por diferentes métodos podem ser comparadas para se verificar a validade da identificação.

II.3) Método com excitação periódica (utilizado).

Como foi empregado processamento digital dos sinais, cada elemento dos vetores de excitação ou resposta é por sua vez um vetor contendo um certo número, que será chamado NP, de valores correspondentes à amplitude dos sinais amostrados a intervalos iguais de tempo ou frequência, de acordo com o caso considerado. No caso de sinais no domínio da frequência esses valores são complexos.

Sendo assim, cada vetor no tempo contém NP valores de amplitude amostrados a intervalos de tempo constantes dt. Portanto cada vetor no tempo contém um registro discreto do que aconteceu com o sinal durante um tempo T = NP.dt.

Se o vetor contém um sinal de comando, ele será enviado ao atuador, após passar por um conversor D/A, de forma consecutiva e contínua, encadeando o final do bloco com o início dele próprio de modo a formar um trem contínuo e ininterrupto de sinal.

É óbvio que os sinais de comando assim gerados serão periódicos de período T, e portanto também o serão as respostas do sistema.

Sendo assim, não é necessário calcular o valor esperado dos sinais, já que uma vez passados eventuais transientes o valor esperado é o próprio valor contido em cada bloco.

Pode-se efetuar a identificação aplicando a excitação num só ponto por vez, mantendo iguais a zero os sinais dos outros n-1 atuadores restantes.

Suponha-se por exemplo que esteja sendo aplicada a excitação num determinado ponto j, e que os sinais de todos os outros atuadores sejam iguais a zero. A equação (II.1) se reduz a:

$$X_i(w) = H_{ij}(w).F_j(w) \quad i=1,..,n \qquad (II.6)$$

onde $X_i(w)$ e $F_j(w)$ são respectivamente os elementos i e j dos vetores $\{X(w)\}$ e $\{F(w)\}$ (lembrar que eles mesmos são por sua vez vetores de números complexos) e $H_{ij}(w)$ é o elemento (também vetor de números complexos) da linha i, coluna j da matriz $[H(w)]$.

Sendo (II.6) uma equação simples de números complexos pode-se isolar $H_{ij}(w)$, obtendo-se assim:

$$H_{ij}(w) = X_i(w)/F_j(w) \quad i=1,..,n \qquad (II.7)$$

Se $F_j(w)$ contém amplitudes diferentes de zero em toda a faixa de frequências de interesse, pode-se calcular toda a coluna j da matriz $[H(w)]$ mediante a equação (II.7).

Repetindo-se este processo para j variando de 1 até n calcula-se toda a matriz $[H(w)]$ para toda a faixa de frequências em estudo.

Uma vantagem adicional deste método é a facilidade com que ele se adapta ao processamento por computador.

A MRF identificada pode ser retangular, se se trabalha com números diferentes de atuadores e de sensores. Neste caso deve-se utilizar algum artifício para poder calcular o vetor das excitações $\{F(w)\}$ a partir do conhecimento do vetor das respostas $\{X(w)\}$ e da MRF $[H(w)]$, como já foi visto no item I.2 do capítulo

I. Tal como mencionado no citado item, deve-se utilizar a equação (I.6) para sistemas com igual número de excitações e respostas e a equação (I.9) para sistemas com maior número de respostas do que excitações.

Este é um dos recursos já existentes na última geração de aparelhos de testes deste tipo lançada pela MTS, podendo-se utilizar no máximo 20 atuadores e até 40 medições para monitoração das respostas [10].

Resumindo, o procedimento utilizado para calcular a MRF consta dos seguintes passos:

1. Joga-se num dos pontos de excitação j um sinal com amplitude no espectro diferente de zero em toda a faixa de frequências em que se deseja identificar a estrutura. As n-1 demais excitações devem permanecer iguais a zero.

2. Registra-se simultaneamente a excitação diferente de zero e as respostas em todos os m pontos de monitoração.

3. Passa-se ao domínio das frequências todos os sinais registrados.

4. Divide-se cada uma das respostas $X_i(w)$, com i variando de 1 até m, pela excitação não nula $F_j(w)$, em frequência. Os resultados desta divisão constituem toda a coluna j da MRF (equação II.7).

5. Repetindo os passos 1 até 4 para j=1,...,n tertemos calculado todos os elementos da MRF.

Por causa do processamento digital dos sinais, a matriz [H(w)] identificada é na verdade armazenada na forma de um vetor de matrizes $[H_i] = [H(w=i.dw)]$, i=0,..,MF

Considerável economia de memória poderia ser obtida ajustando-se (interpolando) curvas para cada elemento da matriz [H(w)]. Para achar as curvas que melhor se ajustam aos pontos obtidos deve-se utilizar um programa de ajuste de curvas, baseado em métodos de regressão, minimos quadrados, "splines", ou qualquer outro princípio utilizado para este fim [35]. Fazendo isto seria suficiente armazenar os parâmetros de cada curva ajustada, em lugar dos MF+1 valores discretos obtidos na identificação para cada uma delas. O preço a ser pago pela economia de memória obtida desta forma é o acréscimo de trabalho de programação e de tempo de processamento necessários para se efetuar o ajuste da curva sobre os pontos discretos calculados e para achar o valor da curva ajustada para cada ponto requerido pelo programa.

Ao se comparar as solicitações de memória para cada caso, efetuar-se-á o cálculo supondo que a MRF tem dimensões mxn, que o número de pontos de discretização da faixa de frequências analizada é MF e que sobre eles foram ajustados polinômios de ordem N.

Para o caso discreto, o número de variáveis a serem armazenadas é:

$$NVd = n.m.(MF+1) \qquad (II.8)$$

Já para o caso de se interpolar um polinômio sobre os pontos, o número de variáveis a serem armazenadas é:

$$NVi = n.m.(N+1) \qquad (II.9)$$

Para os casos rodados, admitindo que a interpolação de polinômios de 3º grau seja satisfatória, a relação NVd/NVi obtida seria 204/16, ou seja, utilizar-se-iam aproximadamente 13 vezes menos variáveis para armazenar a matriz [H(w)] ajustando polinômios de terceiro grau sobre os pontos da discretização. Não foi utilizado este recurso para economizar memória por causa do acréscimo no tempo de processamento que ele acarretaria, como já foi mencionado antes. Tambem não foi possível assumir a simetria da MRF [H(w)] e da sua inversa $[H(w)]^{-1}$, pois esta simetria desaparece quando se trabalha com sistemas não lineares. A simetria foi utilizada para economizar memória no armazenamento das matrizes de massa, amortecimento e rigidez dos sistemas simulados com o objetivo de se verificar o método de CESR.

II.4) Sinal de excitação escolhido para identificar o sistema.

O sinal utilizado para identificar o sistema de acordo com o método descrito no item II.3 deve reunir algumas características particulares para garantir um bom desempenho deste processo de identificação.

Uma característica essencial é que tenha amplitudes em frequência diferentes de zero para toda a faixa em que desejamos trabalhar.

É conveniente utilizar um sinal periódico para evitar eventuais problemas com transientes na resposta.

Deve-se tomar cuidado para não ultrapassar determinados níveis de amplitude que possam danificar a estrutura ou alterar o seu comportamento, e, por isso, é também conveniente trabalhar com sinais com baixo fator de pico. Este problema se torna mais crítico para sistemas não lineares, pois o efeito das não linearidades é minimizado quando trabalha-se com sinais de faixa de frequências e amplitudes próximas das que se deseja simular no teste. Isto pode ser verificado pelos resultados mostrados na tabela IV.2 do exemplo 5 do capítulo IV.

Neste trabalho foi escolhido para fazer a identificação do sistema o Sinal de Schroeder, que reúne a maioria das características desejadas de um sinal para este fim.

O algoritmo proposto por Schroeder para gerar o sinal referido calcula mediante minimização dos coeficientes de autocorrelação, quais são as fases adequadas dos componentes em frequência

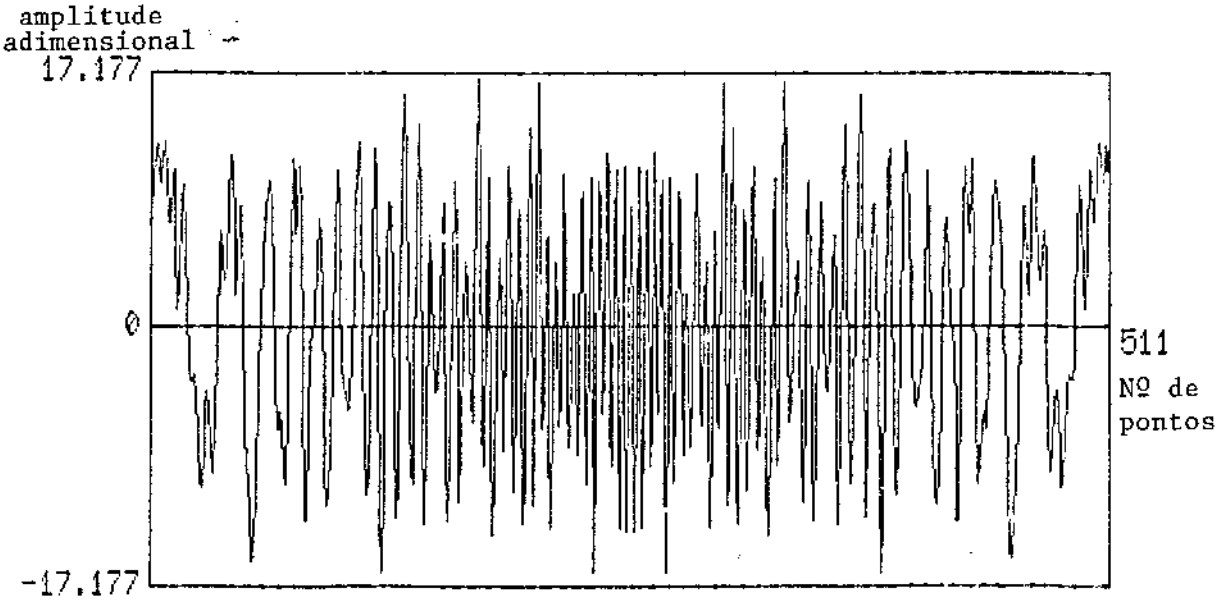


Fig II.1: Sinal de Schroeder no tempo.

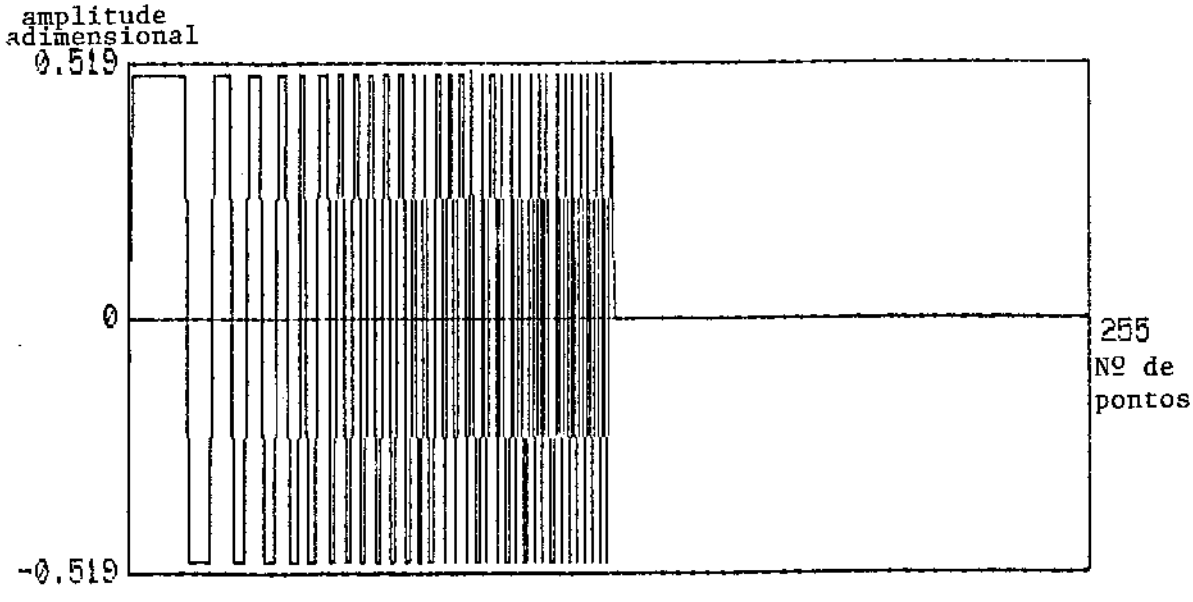


Fig II.2: Sinal de Schroeder em frequência.

para proporcionar ao sinal o menor fator de pico [23], [24].

O fator de pico é definido em [23] como a diferença entre os picos de amplitude máxima e mínima do sinal dividida pelo seu valor RMS.

A versão do gerador de sinal de Schroeder implementada neste trabalho contém algumas restrições que aumentam um pouco o fator de pico mas diminuem sensivelmente o tempo de geração do sinal. Estas restrições são: espectro plano de amplitude um para a parte real e zero para a parte imaginária, sinal simétrico, e fases variando unicamente entre zero e π, o que implica que a parte real do sinal em frequência vale sempre 1 ou -1 e a parte imaginária vale sempre zero.

O sinal gerado com estas condições é mostrado na figura II.1 no tempo e na figura II.2 em frequência.

No apêndice 2 é feita uma descrição mais detalhada do sinal de Schroeder.

CAPÍTULO III

DESCRIÇÃO DO PROGRAMA QUE EFETUA O PROCESSO DE CESR

III.1) Objetivo dos programas.

Foi implementado no computador um programa para efetuar o ajuste da resposta de um sistema de acordo com o algoritmo proposto.

Por razões de disponibilidade e rapidez de processamento foi escolhido para a implementação dos programas um micro-computador de 32 bits, que opera com sistema operacional do tipo UNIX.

A linguagem escolhida para a codificação dos programas foi "C" por vários motivos. Entre eles por ser uma linguagem estruturada, o que facilita a compreensão e organização dos programas, por ser uma linguagem de relativo baixo nível [25] e portanto bastante rápida, e por ter "um bom relacionamento" com o sistema operacional UNIX, que é escrito em C.

O objetivo deste programa foi o de verificar o funcionamento e eficiência do algoritmo proposto, e também o de verificar as suas limitações quando possível.

Devido ao computador não contar com interface de comunicação com instrumentos (conversores D/A e A/D) e também porque se dese-

java ter um maior conhecimento e controle sobre os fenômenos observados para poder detectar e corrigir eventuais falhas, optou-se por substituir a estrutura em estudo por uma sub-rotina que simulasse o seu comportamento dinâmico (integrador numérico). Apesar disso foram tomadas precauções para que o programa resultante fosse facilmente transportável para um caso real, ou experimental, sendo que basicamente as únicas alterações necessárias seriam a substituição do vetor de excitações externas fornecido ao integrador por saídas analógicas enviadas aos atuadores, e do vetor de respostas devolvido pelo integrador pela aquisição dos sinais obtidos em sensores montados na estrutura.

Além disso deve-se tomar alguns cuidados especiais, como por exemplo garantir que as aquisições das respostas sejam iniciadas sempre em sincronismo com os sinais de excitação. Se isto não for feito não se obtém convergência na aplicação do algoritmo iterativo.

Já no caso da estrutura ser simulada por um integrador não existe problema com o sincronismo, devido a que o processamento digital em bloco dos sinais nos garante automaticamente a manutenção das fases relativas.

III.2) O programa principal.

O programa principal efetua basicamente os passos descritos no item II.4. Ele cumpre a função de um gerenciador de subprogramas para que isto seja possível. A sua estrutura é esquemáticamente a seguinte:

1. Entrada de dados.

Lê os sinais desejados como resposta do sistema, hipoteticamente gravados em pontos da estrutura quando esta se encontrava na situação que se deseja reproduzir na simulação, e o sinal de excitação (Schroeder) que será utilizado na identificação do sistema. Lê tambem as matrizes Ms,C e K que descrevem a configuração do sistema em termos de massa, amortecimento e rigidez, sendo que isto não é feito no caso real quando em geral desconhecemos os parâmetros da estrutura.

Por tratarem-se de sistemas auto-adjuntos, as matrizes Ms, C e K são simétricas, e para economizar memória só é armazenada a metade superior dos seus coeficientes, e unicamente da banda com elementos diferentes de zero. O armazenamento é feito na forma de vetor, por colunas. Para efetuar a conversão das coordenadas da matriz para a coordenada correspondente do vetor é usada a função ICOORD.

2. Identificação do sistema.

Consiste de:

2.1 Chama o integrador para obter a resposta do sistema ao sinal de excitação aplicado num dos pontos da estrutura.

2.2 Efetua a transformada rápida de Fourier (TRF) do sinal de Schroeder e dos sinais obtidos.

2.3 Divide cada uma das respostas, em frequência, pela transformada do sinal de excitação (Schroeder).

2.4 Repete os pasos 2.1 até 2.3 para o sinal de Schroeder aplicado num outro ponto de excitação, até que estes tenham se esgotado. Monta a MRF.

3. Inverte a MRF.

4. Passa os sinais desejados para o dominio da frequência.

5. Faz que o sinal diferença entre o sinal desejado e o obtido seja igual ao sinal desejado (supõe sinal obtido = 0), e faz sinal de excitação igual 0.

6. Processo iterativo.

6.1 Multiplica a inversa da MRF pelo vetor dos sinais diferença entre o sinal desejado e o obtido, para obter a correção do sinal de excitação.

6.2 Soma o resultado de 6.1 ao sinal de excitação.

6.3 Efetua a transformada rápida de Fourier inversa (TRFI) dos sinais de excitação.

6.4 Chama o integrador para obter a resposta do sistema aos sinais de excitação.

6.5 Passa os sinais obtidos para o domínio das frequências mediante a TRF.

6.6 Compara o sinal obtido com o desejado. Se o resultado não for satisfatório atualiza o sinal da diferença entre o sinal desejado e o obtido (sinal de erro), e repete todo o item 6. Se o resultado for satisfatório continua em frente.

O sinal é considerado satisfatório se a somatória dos módulos do sinal de erro for menor do que 1% da somatória dos módulos do sinal desejado, em frequência.

7. Guarda os sinais de excitação corretos em arquivo, imprime mensagens de aviso, etc... e finaliza o programa.

A listagem do corpo principal do programa está no apêndice 3 com o nome SIM_TEST. O arquivo de dados e definições de variáveis requeridos pelo programa SIM_TEST está listado com o nome DADOS. Tambem estão no apêndice 3 as listagens dos sub-programas requeridos pelo programa principal descrito acima, antecedidas por uma breve descrição dos mesmos.

Por causa da dificuldade de alocação de memória para as variáveis, muitas delas precisaram ser utilizadas para mais do que uma função ao longo do programa, o que lamentavelmente prejudica a clareza e compreensão do mesmo.

Apesar do processamento ser conceitualmente idêntico, seria necessário efetuar algumas ligeiras modificações no programa para casos com números diferentes de excitações e respostas. As mudanças necessárias são unicamente a pré-multiplicação da MRF e do vetor das respostas pela matriz transposta da MRF, para desta forma se utilizar a equação (I.9) no lugar da (I.6) para o cálculo do vetor das excitações. Tal como está implementada, esta versão do programa somente prevê o uso da equação (I.6).

III.3) Alguns comentários adicionais sobre o programa principal.

Tal como foi implementado o programa, a correção do sinal efetuada no item 6.2 da seção anterior é baseada em 100% do sinal de correção obtido no item 6.1. No caso real (estrutura real, não simulada) este procedimento pode colocar em risco a integridade do protótipo, já que por influência das não linearidades podem ser obtidos sinais de excitação com amplitudes maiores do que as corretas. Por este motivo devem ser previstos alguns mecanismos de segurança para um caso real. Uma das alternativas possíveis é utilizar correções menores do que as obtidas pelo cálculo do item 6.1, o que deve retardar a convergência, mas confere segurança ao sistema. Na referência [4] é mencionado que cada correção é efetuada utilizando apenas 20% do sinal de correção obtido, enquanto que no sistema descrito na referência [12] já existe o recurso do usuario escolher a porcentagem do sinal de correção a ser efetivamente utilizado.

O uso de apenas uma porção do sinal de correção em cada iteração faz com que a convergência aconteça sempre por valores inferiores aos procurados, já que o sinal de excitação é assumido igual a zero no início do processo.

Além deste recurso devem existir, obviamente, outros mecanismos de segurança, por exemplo limitadores de amplitudes, comuns à maioria dos sistemas de testes dinâmicos.

No que diz respeito à aplicação do sistema de testes por simulação das condições de vibração, ou seja, utilizando CESR, o programa de controle descrito é geral, mas a implementação práti-

ca do sistema passa pela seleção de uma diversidade de opções de montagem. Numa primeira análise podem ser distinguidas opções de introdução das excitações e de origem das forças de reação.

Com relação à introdução das forças podem ser achadas nas referências ([2], [7], [8], [9], [10], [11], e [12]), no que diz respeito à aplicação na indústria automobilística, as seguintes opções:

- Excitação vertical em cada roda. Fácil de montar e desmontar, é realista por introduzir os esforços atraves dos pneus. A maior limitação desta técnica é que permite introduzir apenas excitações verticais.

- Excitação triaxial nas rodas. Requer a remoção das rodas e sustituição por um dispositivo especial. Tem a vantagem de permitir a simulação de esforços verticais, longitudinais e laterais.

- Excitação triaxial nas rodas mais direção e freio. é semelhante à anterior, mas um pouco mais completa. Permite simular com maior perfeição um número maior de situações.

- Excitação quadriaxial nas rodas. Permite aperfeiçoar a simulação de freadas [12].

Com relação às reações, nas citadas referências são mencionadas as opções:

- Reação inercial. As únicas forças de reação envolvidas são as provenientes da inércia das massas da estrutura. é a opção que permite simulações mais realistas. A maior deficiência consiste em não permitir simular corretamente curvas e freadas por limitação dos deslocamentos dos atuadores. Este problema pode ser parcialmente contornado utilizando compressão do sinal, que é um dos recursos de edição do sinal que será abordado novamente nas con-

clusões.

- Reação semi-inercial. Um dos eixos está ligado aos atuadores enquanto que o outro é preso num pivô. Permite a simulação de esforços no eixo ligado aos atuadores, inclusive freadas, mas introduz esforços fictícios no restante da estrutura.

- Carroceria semi-fixa. A carroceria é apoiada mediante molas especiais numa estrutura tipo gaiola. Permite simular todo tipo de esforços, mas tambem introduz solicitações fictícias em determinadas regiões do veículo.

- Carroceria fixa. A carroceria é amarrada a uma estrutura rígida. Esta técnica é aplicavel apenas para testes dos mecanismos de suspensão.

A escolha da opção mais adequada aos testes que se deseja realizar, assim como a execução física das montages necessárias para os mesmos, são algumas das dificuldades que devem ser resolvidas pelo usuário para cada caso particular.

CAPÍTULO IV

RESULTADOS DA SIMULAÇÃO

IV.1) Introdução.

Vários exemplos foram rodados utilizando o programa descrito no capítulo anterior, para verificar a validade do algoritmo proposto. Cada exemplo se diferencia dos outros num destes quatro itens:

1) Pelo modelo da estrutura.

2) Pelo tipo de não linearidade adicionado ao modelo da estrutura.

3) Pelo valor da não linearidade.

4) Pela resposta que se deseja obter do sistema.

Para cada exemplo foi registrado o número de iterações que foram necessárias até a obtenção de uma resposta que tenha um erro menor do que 1% com relação à resposta desejada. Este é um critério para se medir a "dificuldade" de convergência. O critério para avaliar o erro existente que foi utilizado para a primeira estrutura simulada foi o de comparação dos valores médios quadráticos dos sinais desejados e dos sinais de erro, resultantes da diferença entre os sinais desejados e os obtidos. Já para

a segunda, em lugar do valor médio quadrático, foi utilizada a soma dos módulos das componentes do sinal em frequência dividida pelo número de pontos de discretização da faixa de frequências utilizada na análise dos sinais. Desta forma é possível economizar memória do computador, trabalhando com um número menor de variáveis. Este recurso será abordado novamente no item IV.3 deste capítulo. O erro foi considerado aceitável se o valor obtido para o sinal de erro for menor do que 1% do obtido para o sinal desejado, com qualquer um dos dois critérios.

Para todos os casos o número de iterações necessário variou de um no caso de sistemas lineares, para os quais não é necessário efetuar correções, até infinito (casos em que não houve convergência) para sistemas com não linearidades acima de um certo valor. Nos casos em que não houve convergência o erro mínimo aconteceu sempre antes de 20 iterações, e a partir daí o erro começou a aumentar, probablemente por causa do acúmulo de erros de operação. Por esse motivo, o número de iterações permitido foi limitado a no máximo 20, e se até a vigésima iteração não se obteve convergência assume-se que ela não é possível.

Foram equacionadas basicamente duas estruturas, ambas inspiradas em modelos muito simples de veículos.

IV.2) Resultados obtidos com o primeiro modelo.

A primeira estrutura modelada para simular o seu comportamento no programa foi inspirada num problema proposto em [21] (cap. 5, pag. 138) no qual é apresentado um modelo para o comportamento dinâmico de um automóvel.

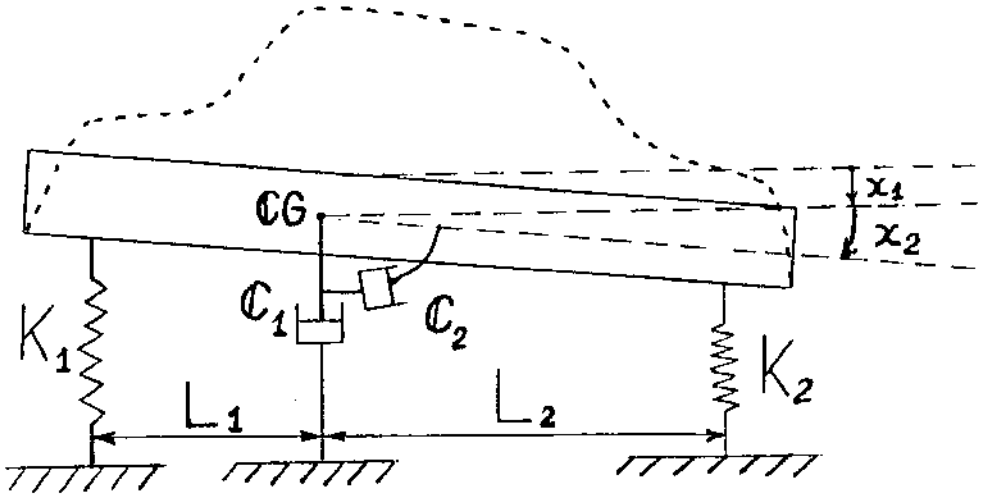


Fig IV.1: Esquema da estrutura do primeiro modelo.

A este modelo foi acrescentado amortecimento da forma que pode ser vista na figura IV.1. é obtido assim um sistema de dois graus de liberdade que é descrito pela equação (IV.1).

$$\begin{bmatrix} M & 0 \\ 0 & I \end{bmatrix} \begin{Bmatrix} \ddot{x}_1 \\ \ddot{x}_2 \end{Bmatrix} + \begin{bmatrix} C_1 & 0 \\ 0 & C_2 \end{bmatrix} \begin{Bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{Bmatrix} + \begin{bmatrix} K_1+K_2 & K_2L_2-K_1L_1 \\ K_2L_2-K_1L_1 & K_1L_1^2+K_2L_2^2 \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \end{Bmatrix} = \begin{Bmatrix} F_1 \\ F_2 \end{Bmatrix} \quad (IV.1)$$

Foram mantidos para os parâmetros de massa, momento de inércia e rigidez os valores atribuídos no problema mencionado. Entretanto, para os amortecimentos adicionados, foram arbitrados valores considerados razoáveis. Os valores utilizados foram:

$M = 1461,84$ kg

$I = 2176,25$ kg.m^2

$L_1 = 1,373$ m

$L_2 = 1,678$ m

$K_1 = 35016,40$ N/m

$K_2 = 37934,43$ N/m

$C_1 = 1000,00$ N/(m/s)

$C_2 = 1000,00$ N.m/(rad/s)

As frequências naturais não amortecidas do sistema obtido por substituição desses valores na equação IV.1 são:

$w_1 = 1,10$ Hz

$w_2 = 1,44$ Hz

Como pode ser observado o sistema é linear, e portanto a convergência deve ser imediata. A constatação deste fato permite verificar a correta identificação do sistema e demonstra que o nível de precisão no processamento dos sinais é adequado para a verificação do algoritmo de correção das excitações. Por esse motivo são mostrados a seguir os resultados obtidos em três exemplos rodados utilizando as equações lineares do sistema da figura IV.1.

Exemplo 1: Foi rodado o programa especificando que se deseja como resposta nas duas variáveis senóides com frequência de 2 Hz, que aparecem na figura IV.2. Como era de se esperar, os sinais de excitação corretos (também senóides com frequência de 2 Hz, figura IV.3) para se obter as respostas desejadas foram obtidos na primeira iteração, ou seja, sem necessidade de entrar no processo de correção dos sinais de excitação. Os sinais obtidos são mos-

trados na figura IV.4.

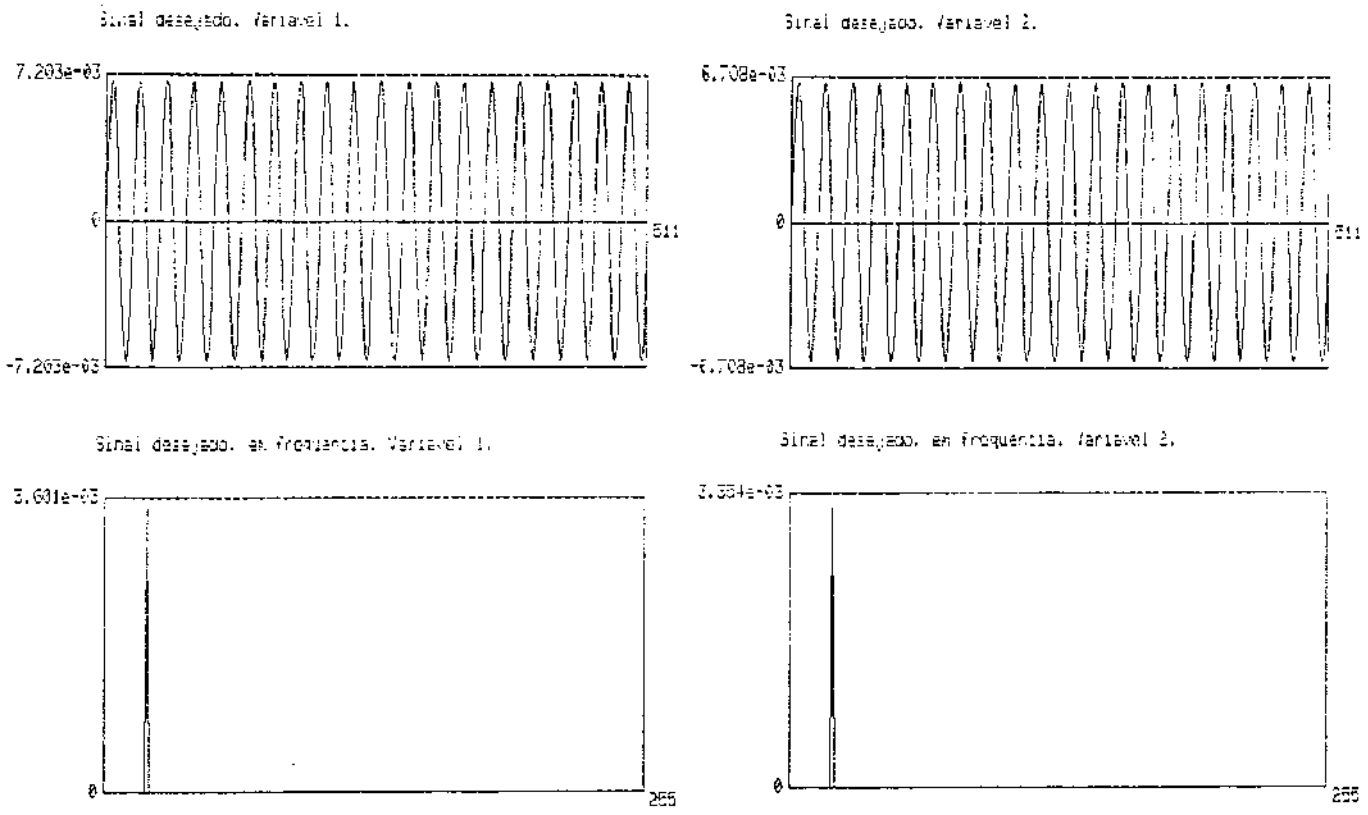


Fig. IV.2: Sinal desejado para o exemplo 1.

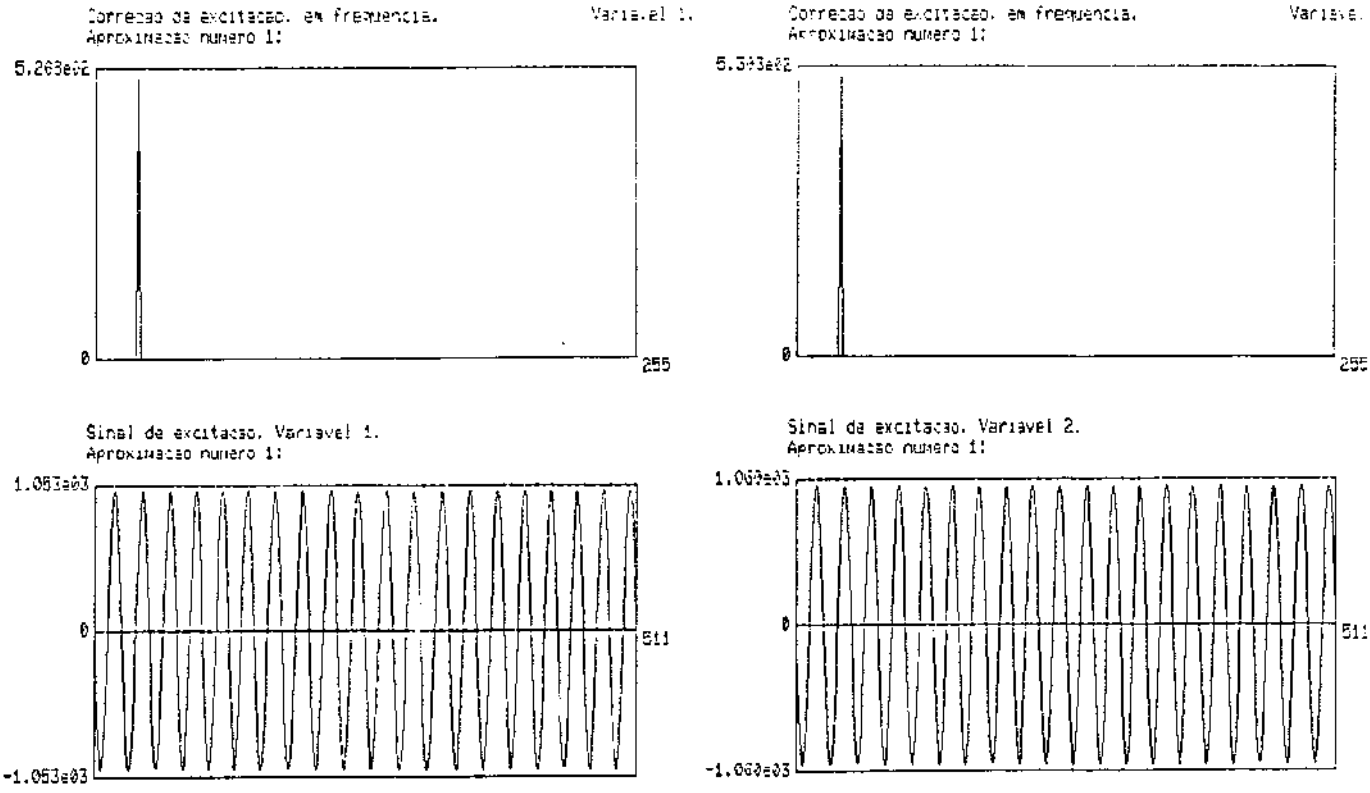


Fig. IV.3: Sinais de excitação obtidos na primeira iteração.

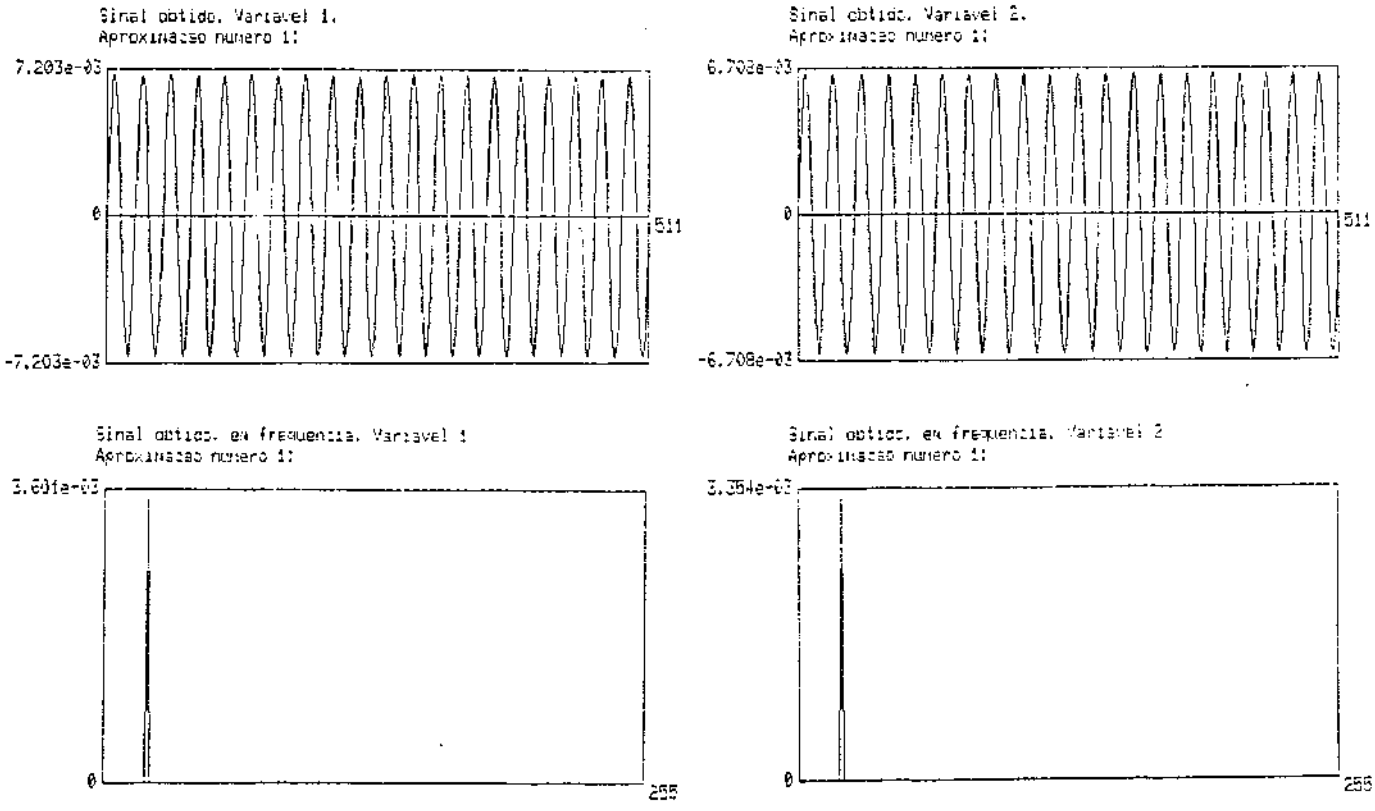


Fig. IV.4: Respostas obtidas na primeira iteração do exemplo 1.

Exemplo 2: Foi rodado o mesmo exemplo anterior, mas especificando-se para as respostas o sinal de Schroeder, figura IV.5. Observe-se que o sinal de excitação necessário neste caso, figura IV.6, será diferente, em amplitude e forma, do sinal de Schroeder desejado como resposta e utilizado para identificar o sistema.

Neste caso também houve convergência sem necessidade de se efetuar correção do sinal de excitação. As respostas obtidas são mostradas na figura IV.7.

Estes dois primeiros exemplos evidenciam o comportamento esperado para sistemas lineares, mas não mostram a eficiência do processo iterativo de correção do erro nos sinais de excitação. Por isso foi rodado mais um exemplo.

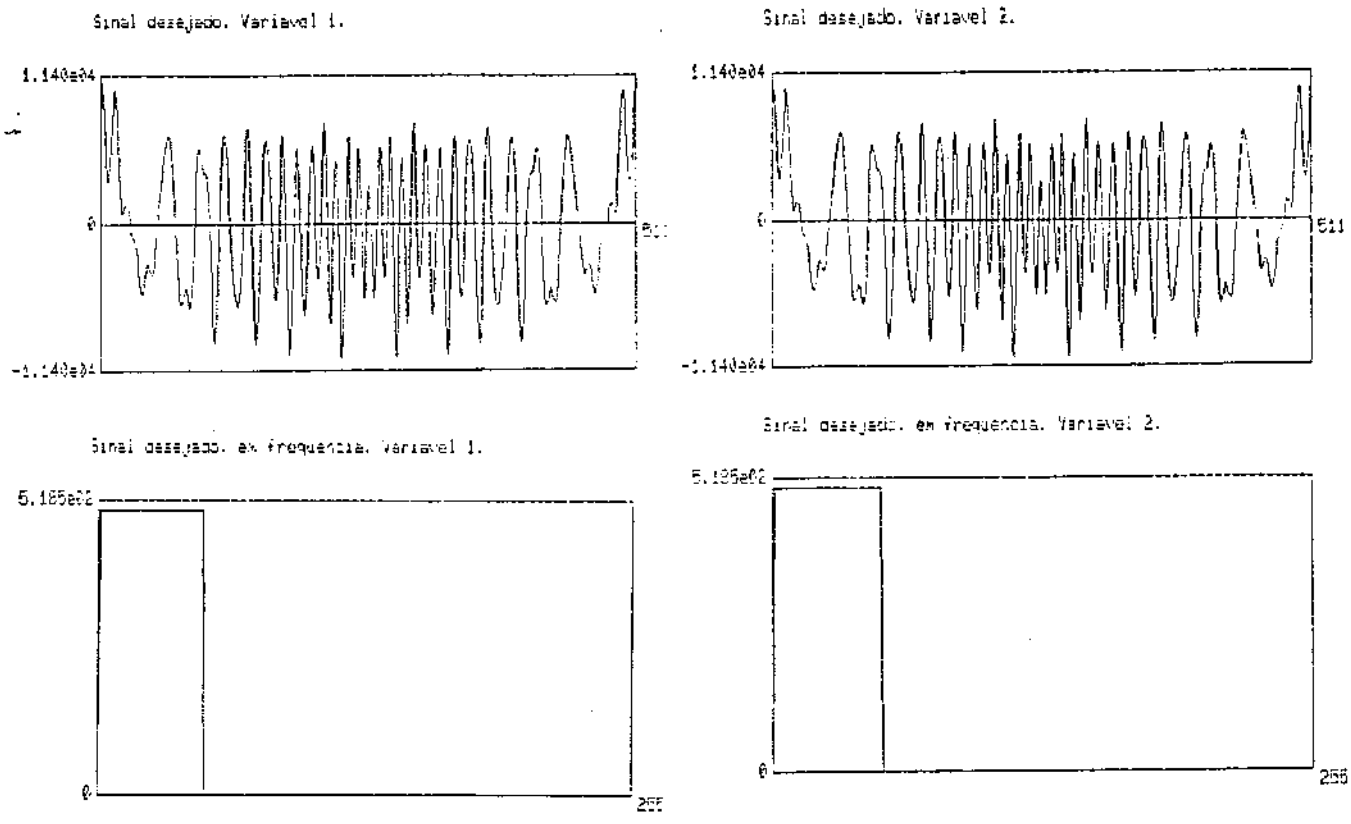


Fig.IV.5: Sinal desejado, exemplo 2.

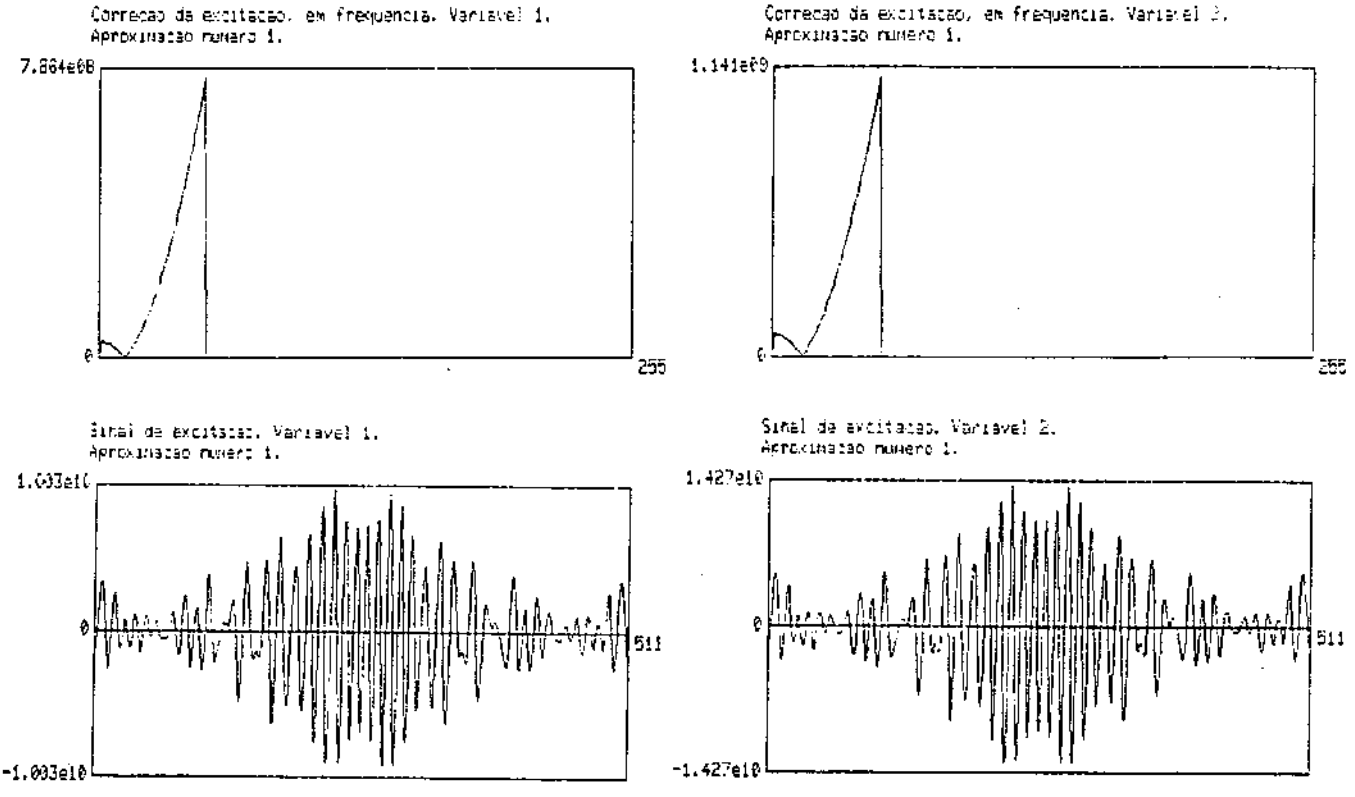


Fig. IV.6: Sinais de excitação calculados, exemplo 2.

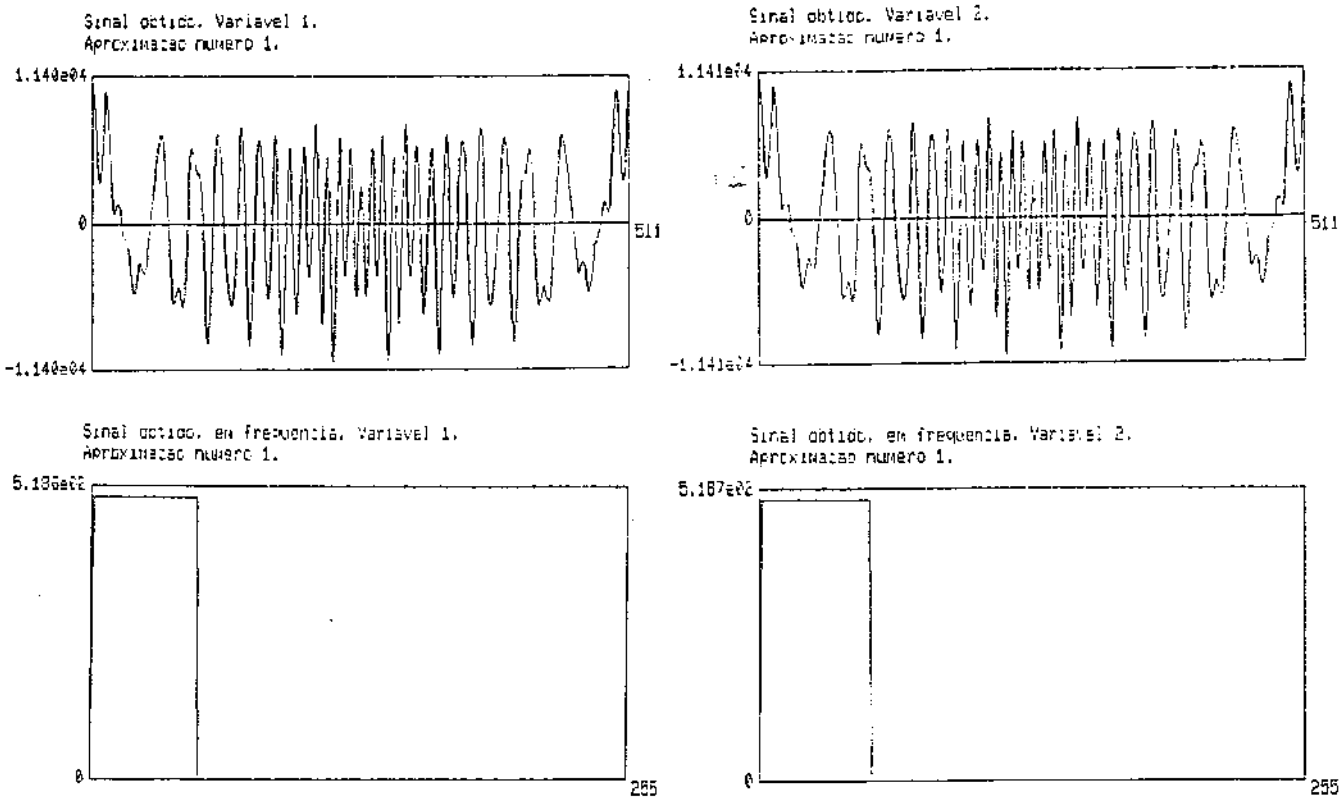


Fig IV.7: Respostas obtidas, exemplo 2.

Exemplo 3: Este exemplo é igual ao anterior, mas para forçar uma correção nos sinais de excitação foi deliberadamente introduzido um erro nos mesmos. Para esse fim os sinais de excitação calculados foram divididos por 2 antes de aplicá-los ao sistema para obter as respostas. Desta forma as respostas obtidas não foram evidentemente as desejadas, forçando a entrada em operação do mecanismo de correção das excitações.

Sendo correta a identificação do sistema linear, a primeira correção dos sinais de excitação deve ser a adequada para a obtenção do sinal desejado, o que foi de fato constatado obtendo-se a convergência das respostas para o valor desejado na segunda iteração.

Os sinais desejados para este exemplo são os mesmos do exemplo anterior, apresentados na figura IV.5. Os sinais de excitação

calculados em cada iteração são mostrados na figura IV.8, e as respostas correspondentes na figura IV.9.

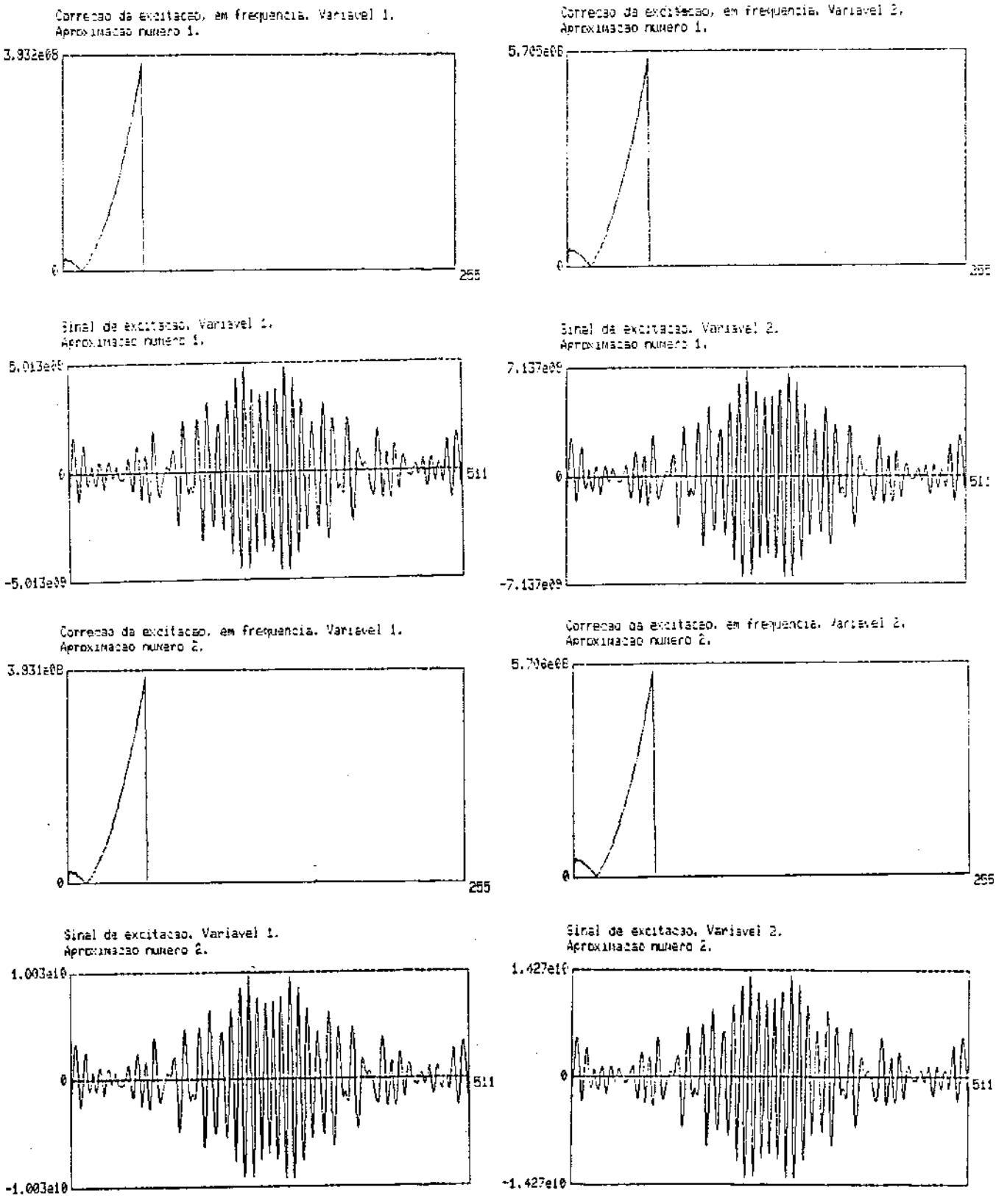


Fig. IV.8: Sinais de excitação calculados no exemplo 3.

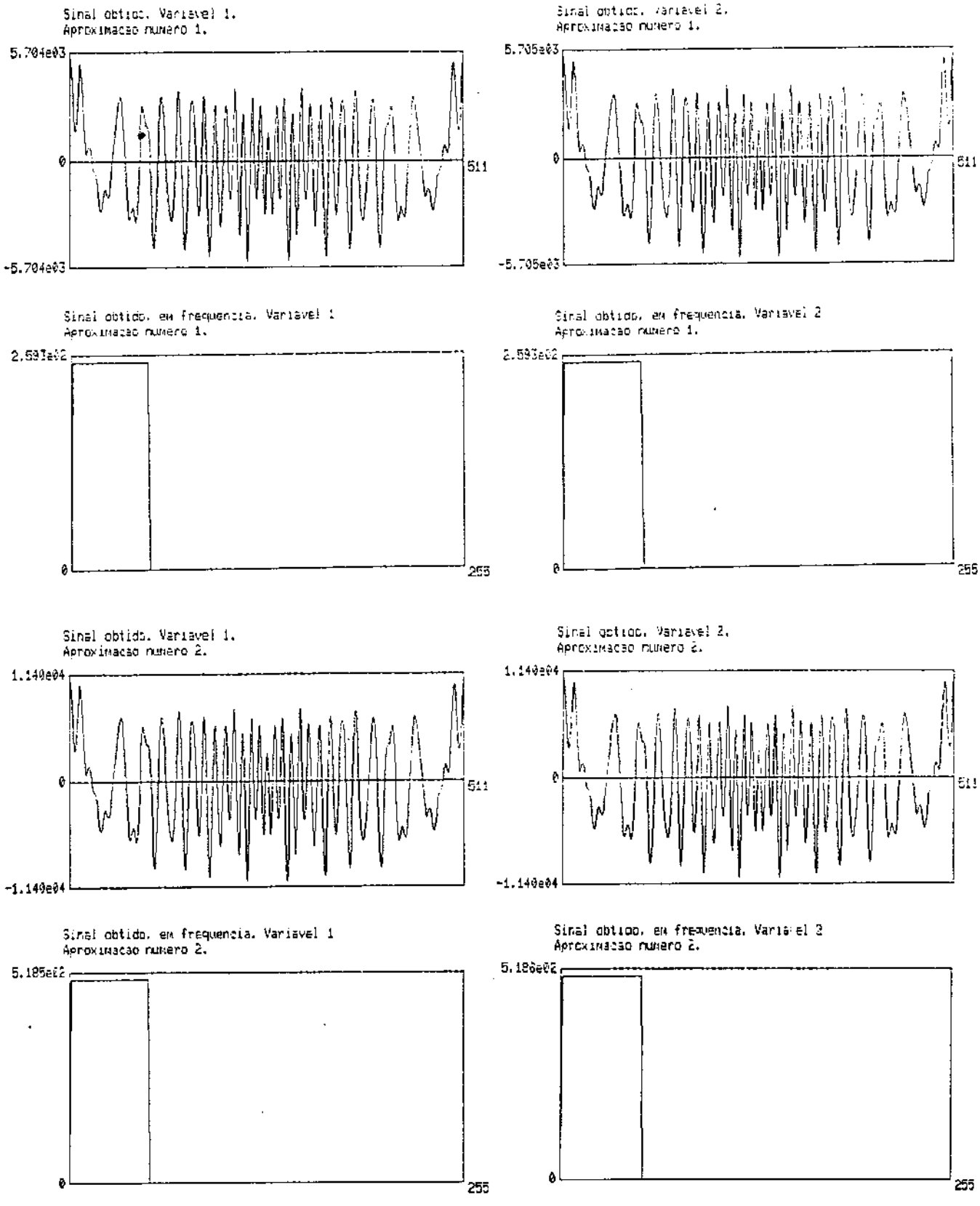


Fig. IV.9: Respostas obtidas no exemplo 3.

Os elementos da MRF e da matriz inversa da MRF identificada para o sistema utilizado nos três exemplos anteriores são mostra-

dos nas figuras IV.10 e IV.11 respetivamente.

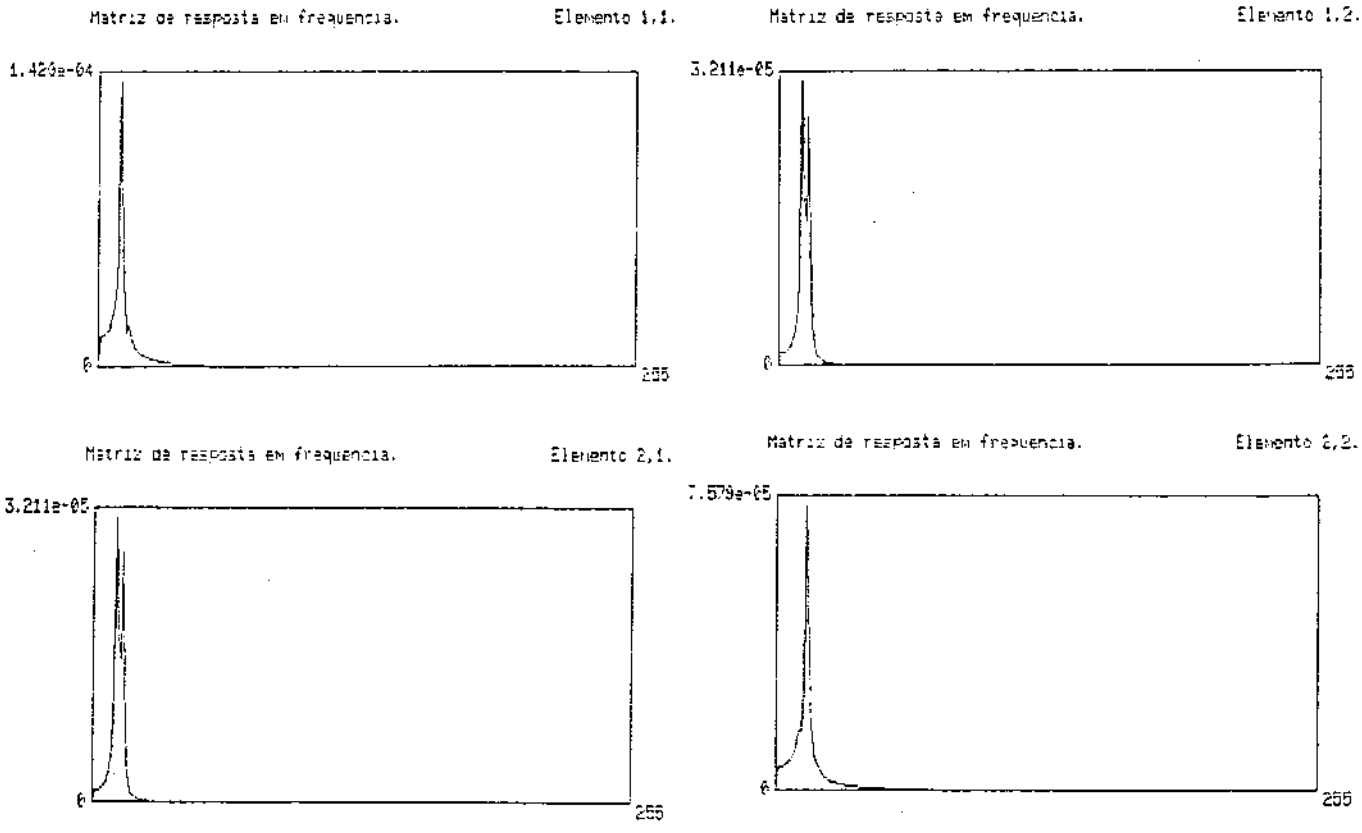


Fig. IV.10: MRF identificada para os exemplos 1, 2 e 3.

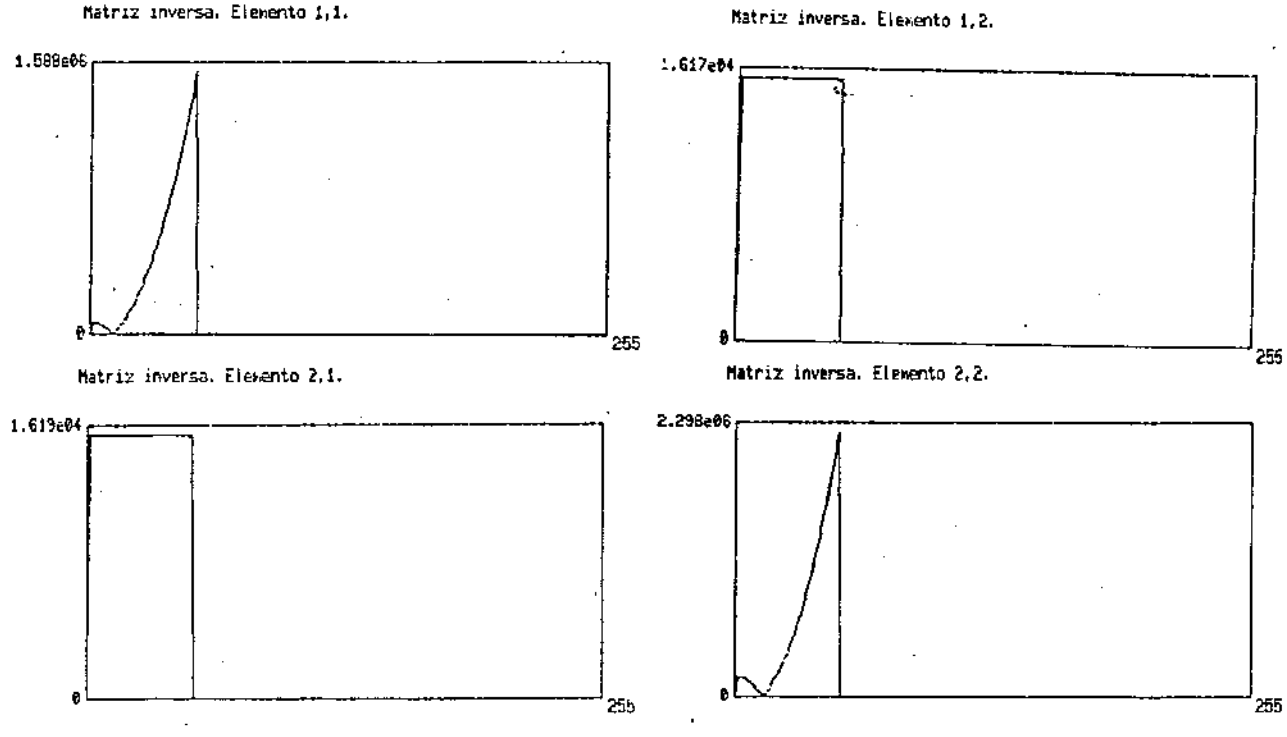


Fig. IV.11: IMRF identificada nos exemplos 1, 2 e 3.

Exemplo 4: o próximo passo é introduzir algum tipo de não linearidade no sistema. Neste exemplo foi simulada a existência de atrito seco. Foi supôsto que as superfícies em contato são perfeitamente planas e homogêneas, e portanto o atrito seco foi caraterizado pelo aparecimento de forças de amplitude constante e que se opõem sempre ao movimento. Empregando notação simplificada, a equação IV.1 fica, com a introdução do atrito seco, da forma mostrada na equação IV.2.

$$[M](\ddot{x})+[C](\dot{x})+[K](x)+a[I](\dot{x}/|\dot{x}|) = (F) \qquad (IV.2)$$

O atrito pode ser tratado como se fosse uma força externa, apesar de ser consequência do comportamento do próprio sistema, e portanto pode ser passado para o lado direito da equação, que fica então:

$$[M](\ddot{x})+[C](\dot{x})+[K](x) = (F)-a[I](\dot{x}/|\dot{x}|) \qquad (IV.3)$$

O atrito seco constitui uma não linearidade de comportamento bastante atípico por representar uma força que depende apenas da direção do deslocamento, tendo portanto pequena relação com o comportamento geral do sistema.

O parâmetro a da equação (IV.3) é um coeficiente de amplitude do atrito, e neste exemplo foi atribuido a ele o valor 1. Desta forma a amplitude das forças não lineares provenientes do atrito é da mesma magnitude das componentes harmônicas do sinal utilizado para a identificação.

Os elementos da MRF e da matriz inversa da MRF identificados neste caso são mostrados nas figuras IV.12 e IV.13 respectivamente. Como pode ser observado por simples comparação com as figuras IV.10 e IV.11 o efeito da não linearidade não foi, em absoluto, desprezível.

Os sinais desejados como respostas para este exemplo foram os mesmos dos exemplos 2 e 3.

Para este exemplo se obteve convergência após 6 iterações. A convergência foi assintótica como pode ser observado na tabela IV.1.

Na figura IV.14 é apresentada a sequência de espectros obtidos nas diferentes iterações do exemplo 4, sendo fácil observar a evolução evidenciada pelos resultados da tabela IV.1.

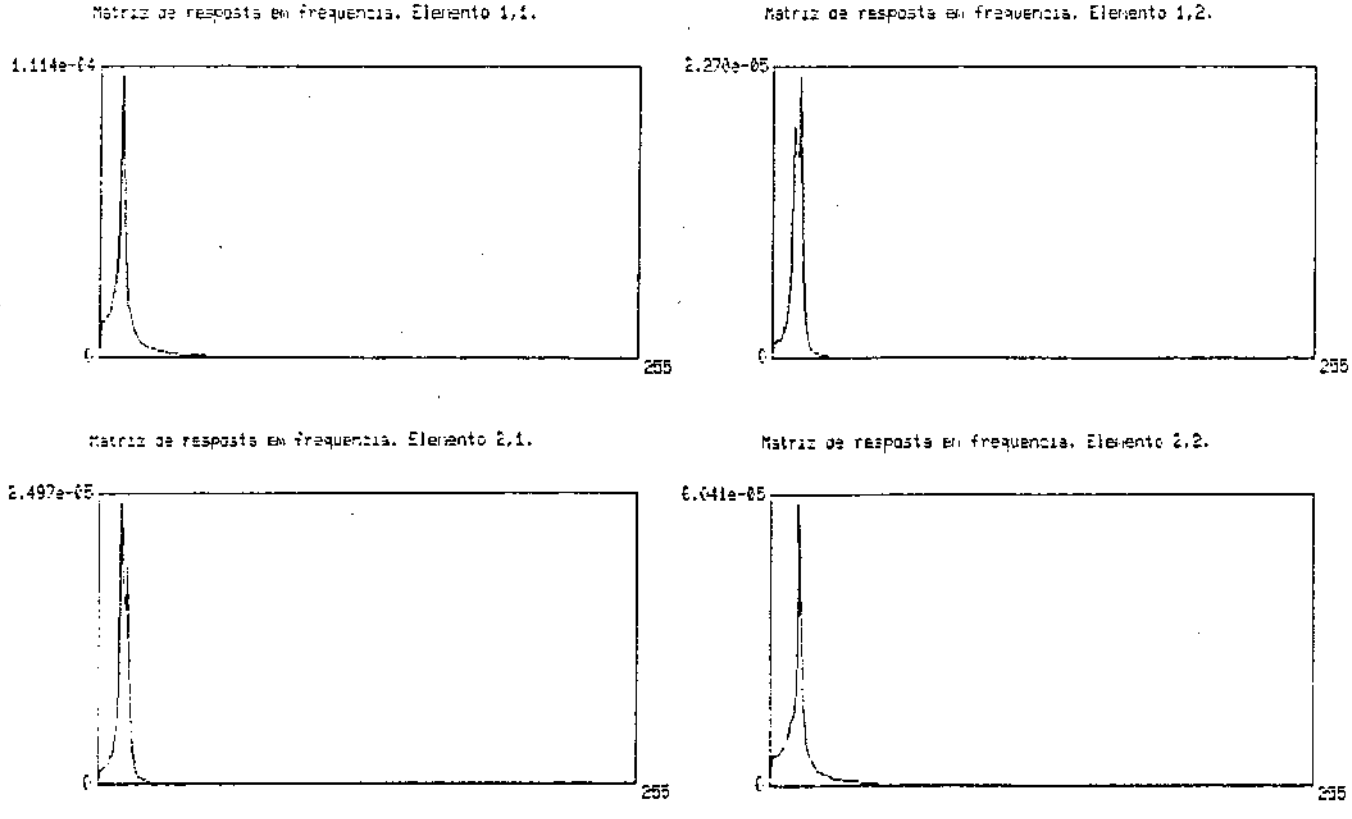


Fig. IV.12: MRF identificada no exemplo 4.

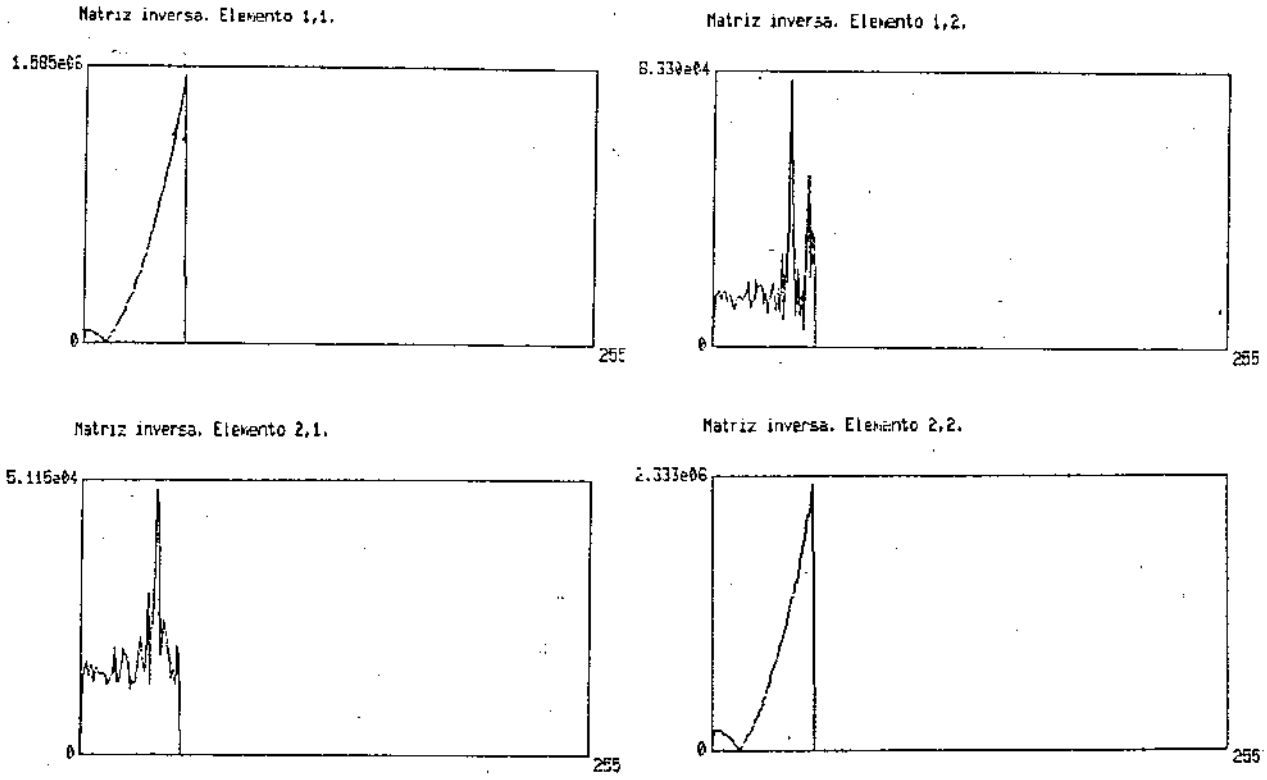


Fig. IV.13: IMRF identificada para o exemplo 4.

Tab. IV.1 - Erro ocorrido em cada iteração no exemplo 4.

Iteração	Erro na variável 1 (%)	Erro na variável 2 (%)
1	518,07	361,45
2	66,06	49,80
3	16,10	11,69
4	4,84	3,61
5	1,91	1,52
6	0,95	0,88

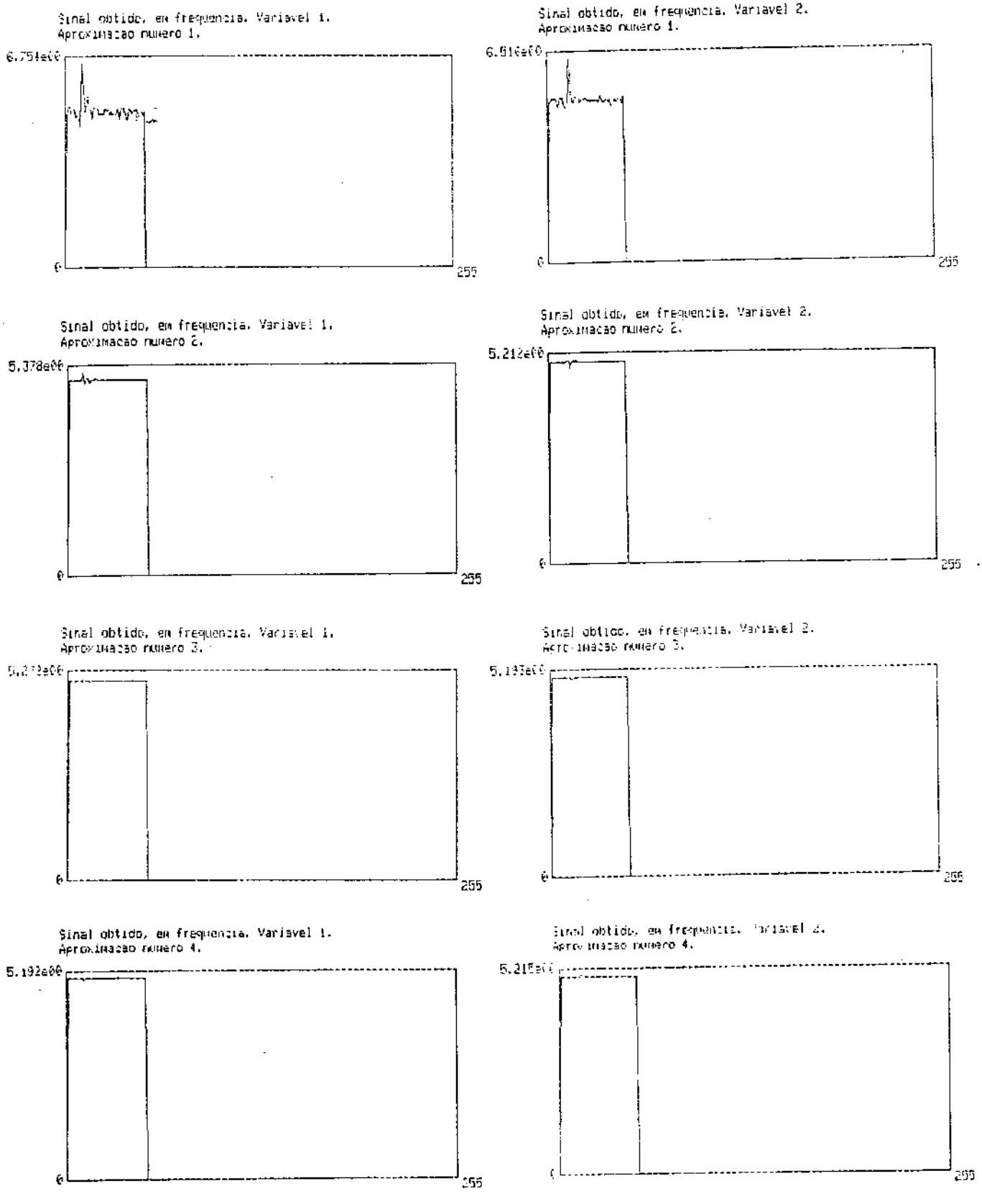


Fig. IV.14: Sequência de espectros das respostas obtidas em cada iteração do exemplo 4.

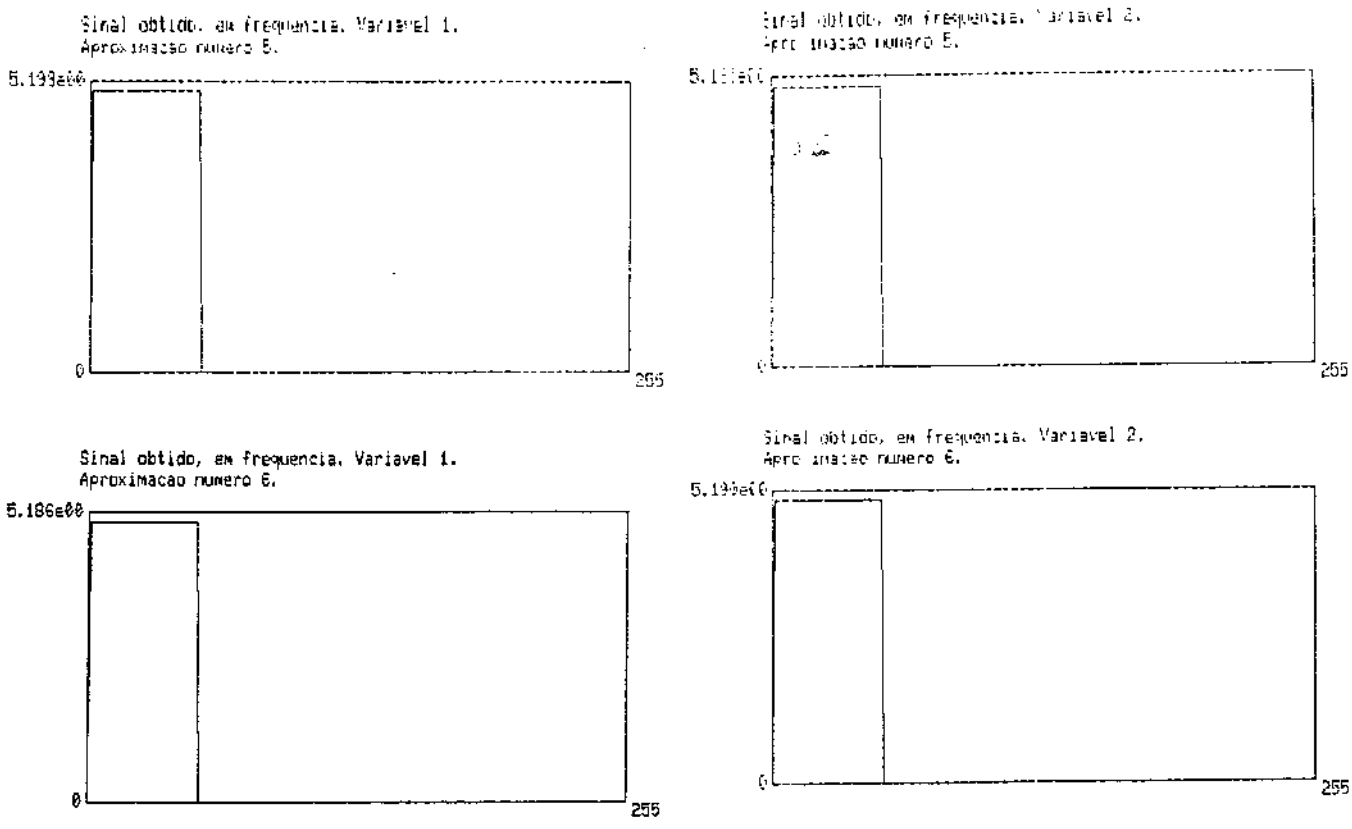


Fig. IV.14: Continuação.

Exemplo 5: Quando se efetua a identificação linear de um sistema não linear, é cometido um erro que se acentua à medida que se trabalha em condições mais distantes daquelas em que foi feita a identificação. Por exemplo, a rigidez identificada num sistema linear vale para qualquer deslocamento sempre que seja mantida a condição de linearidade. Para um sistema não linear no entanto a rigidez identificada é a rigidez correspondente àquela situação em que foi efetuada a identificação, e em geral varia para qualquer outra situação. Por este motivo é sempre aconselhável efetuar a identificação do sistema com sinais de excitação que tenham uma amplitude adequada para ocasionar nas respostas grandezas da ordem dos sinais que se deseja repetir nos testes. Desta forma a influência das não linearidades é minimizada.

Para verificar como a diferença das amplitudes dos sinais desejados e os obtidos na identificação dificulta a convergência, foi montado um exemplo onde é mudada a amplitude dos sinais desejados e mantidas constantes as amplitudes dos sinais utilizados para a identificação.

Para o primeiro caso o sinal desejado tem amplitudes da ordem das obtidas durante a identificação. Nos casos seguintes o sinal desejado é sucessivamente multiplicado por 10, de forma que a amplitude dos sinais desejados é aproximadamente 10 vezes maior que a amplitude obtida na identificação para o segundo caso, 100 vezes maior para o terceiro, e assim sucessivamente. A relação entre as amplitudes dos sinais desejados e os obtidos durante a identificação é expressa, na tabela IV.2, pelo parâmetro Rel.

Para todos os casos o sistema tem a mesma não linearidade, equivalente à adição de molas de rigidez quadrática atuando em paralelo com as molas convencionais da estrutura. É aqui chamada mola com rigidez quadrática qualquer dispositivo que fornece uma força proporcional ao quadrado do deslocamento relativo de suas extremidades a partir de uma posição de equilíbrio, e de sentido contrário.

A formulação do sistema é portanto a dada na equação (IV.4).

$$[M]\{\ddot{x}\}+[C]\{\dot{x}\}+[K]\{x\} = \{F\}-a[K]\{x.|x|\} \qquad (IV.4)$$

O parâmetro a estabelece a amplitude das forças não lineares.

Os resultados obtidos são mostrados na tabela IV.2, onde Rel é a relação aproximada das ordens de amplitude dos sinais deseja-

dos e os obtidos durante a identificação, e N é o número de iterações necessárias até a convergência.

Tab. IV.2 - Número de iterações (N) necessárias para a convergência, onde o parâmetro Rel mostra a relação entre as amplitudes dos sinais desejados e os obtidos durante o processo de identificação, no sistema 1 com rigidez quadrática.

Rel	10^0	10^1	10^2	10^3	10^4
N	2	3	3	4	>20

IV.3) Resultados obtidos com o segundo modelo.

O modelo anterior tem somente dois graus de liberdade e a forma como foi introduzido o amortecimento está muito distante da realidade de um veículo. Este fato não chega a prejudicar a análise feita dado que a validade do sistema independe do modelo, mesmo que as equações não representem nenhuma estrutura real, e porque o teste de veículos é apenas uma das aplicações possíveis para este sistema. Mas para dar maior consistência aos resultados extraídos foi modelada uma nova estrutura, esquematicamente mostrada na figura IV.15.

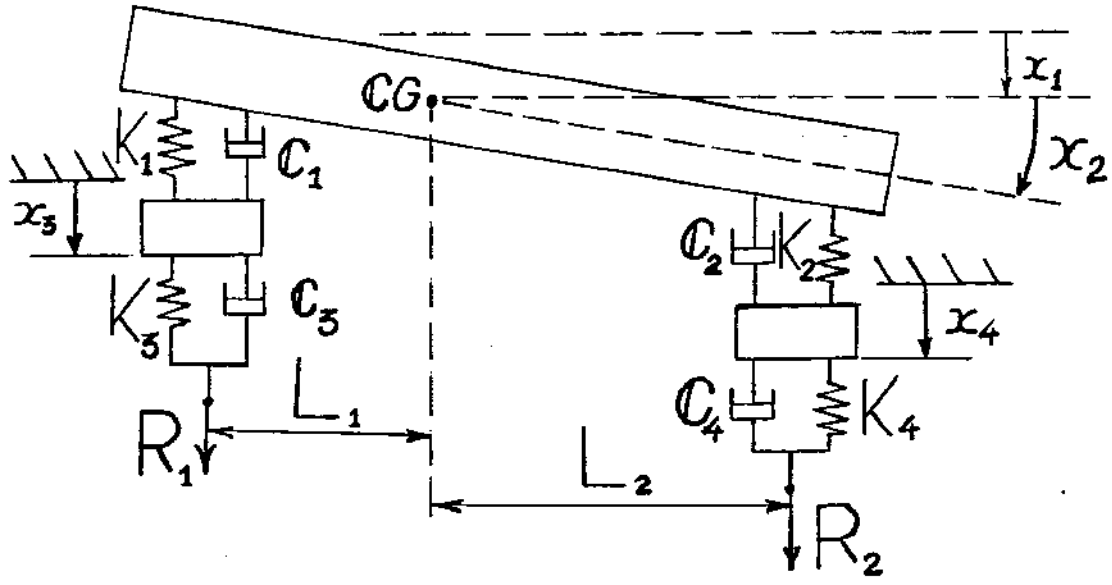


Fig. IV.15: Esquema da estrutura do segundo modelo.

O modelo matemático da estrutura é dado na equação (IV.5).

$$\begin{bmatrix} M & 0 & 0 & 0 \\ 0 & I & 0 & 0 \\ 0 & 0 & M_1 & 0 \\ 0 & 0 & 0 & M_2 \end{bmatrix} \begin{Bmatrix} \ddot{x}_1 \\ \ddot{x}_2 \\ \ddot{x}_3 \\ \ddot{x}_4 \end{Bmatrix} + \begin{bmatrix} 2(C_1+C_2) & 2(C_2L_2-C_1L_1) & -C_1 & -C_2 \\ 2(C_2L_2-C_1L_1) & 2(C_1L_1^2+C_2L_2^2) & C_1L_1 & -C_2L_2 \\ -C_1 & C_1L_1 & C_1+C_3 & 0 \\ -C_2 & -C_2L_2 & 0 & C_2+C_4 \end{bmatrix} \begin{Bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{Bmatrix} +$$

$$\begin{bmatrix} 2(K_1+K_2) & 2(K_2L_2-K_1L_1) & -K_1 & -K_2 \\ 2(K_2L_2-K_1L_1) & 2(K_1L_1^2+K_2L_2^2) & K_1L_1 & -K_2L_2 \\ -K_1 & K_1L_1 & K_1+K_3 & 0 \\ -K_2 & -K_2L_2 & 0 & K_2+K_4 \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \\ C_3R_1+K_3R_1 \\ C_4R_2+K_4R_2 \end{Bmatrix} \quad (IV.5)$$

Este sistema, como pode ser observado, é de quatro graus de liberdade.

Os valores atribuídos aos parâmetros deste modelo são os seguintes:

$M = 1461,84$ kg

$I = 2176,25$ kg.m^2

$L_1 = 1,373$ m

$L_2 = 1,678$ m

$K_1 = 35016,40$ N/m

$K_2 = 37934,43$ N/m

$C_1 = 7082,66$ N/m/s

$C_2 = 7371,90$ N/m/s

$M_1 = 10,00$ kg

$M_2 = 10,00$ kg

$K_3 = 350164,00$ N/m

$K_4 = 379344,30$ N/m

$C_3 = 708,30$ N/m/s

$C_4 = 737,20$ N/m/s

Devido à limitação de memória do computador foi necessário otimizar o uso das variáveis, e por esse motivo o erro não será mais medido pelo valor médio quadrático como foi feito até agora. Nos exemplos seguintes ele será medido pela somatória do módulo do sinal em frequência dividido pelo número de pontos da faixa de frequências analisada. Desta forma é possivel reutilizar variáveis ociosas neste ponto do programa, permitindo portanto alguma economia de memória.

As variáveis R_1 e R_2 representam os deslocamentos impostos nos pontos indicados na figura IV.15, e a equação (IV.5) leva em consideração os esforços que são transmitidos à estrutura como consequência desses deslocamentos. é assumido que estes são os únicos esforços externos agindo na estrutura, e por isso aparecem os dois zeros nas duas primeiras coordenadas do vetor dos termos independentes.

é razoável pensar que num caso real uma estrutura ensaiada tenha muitos mais graus de liberdade do que o número de pontos de excitação ou monitoração utilizados no ensaio. De forma análoga, para os próximos exemplos se tentará ajustar a resposta desejada em duas variáveis mediante o controle dos deslocamentos de dois outros pontos da estrutura. O fato do modelo ser na verdade de quatro graus de liberdade é desconhecido para o algoritmo que corrige as excitações, e é levado em consideracão unicamente pelo integrador, que obviamente calcula as respostas para todas as variáveis do sistema.

Exemplo 6: O primeiro exemplo rodado com este novo sistema não carrega qualquer tipo de não linearidade. O objetivo deste exem-

plo foi o de verificar a correta identificação do sistema linear e registrar a matriz inversa da MRF para que possa ser observado o efeito das não linearidades introduzidas nos próximos exemplos.

Como era de se esperar se obteve a resposta desejada já na primeira iteração, sem que houvesse necessidade de se efetuar correção no sinal de excitação calculado.

A MRF e a matriz inversa da MRF identificadas para o segundo sistema no caso linear são mostradas na figura IV.16 e IV.17 respectivamente.

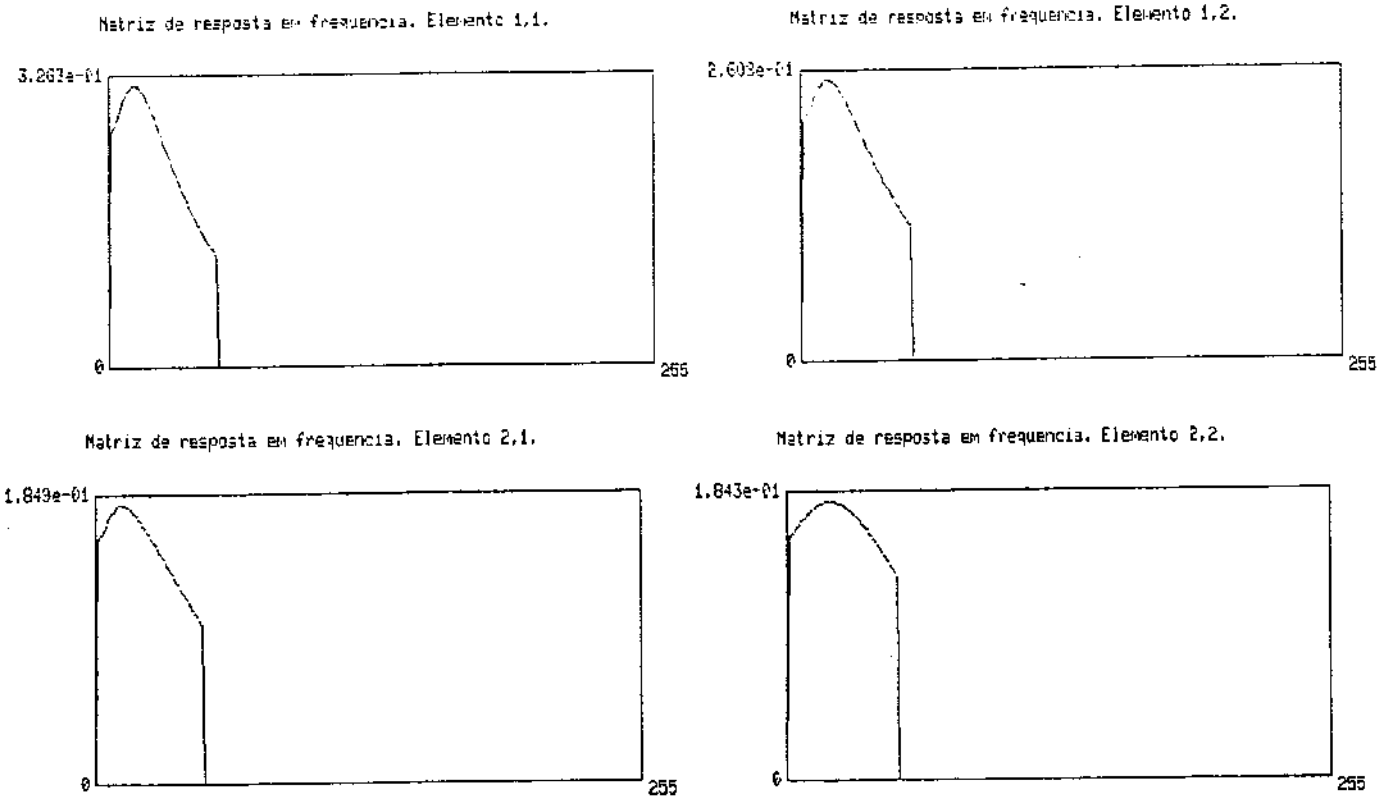


Fig. IV.16: MRF identificada no exemplo 6. Sistema 2, linear.

Exemplo 7: Ao sistema linear anterior foi adicionada não-linearidade proveniente da existência de atrito seco no deslocamento de todas as variáveis. O artifício utilizado para introduzir o atrito seco nas equações é o mesmo utilizado com esse fim no pri-

meiro sistema, no exemplo 3, e que é mostrado na equação (IV.3).

O valor do atrito foi sendo incrementado até não haver mais convergência no processo.

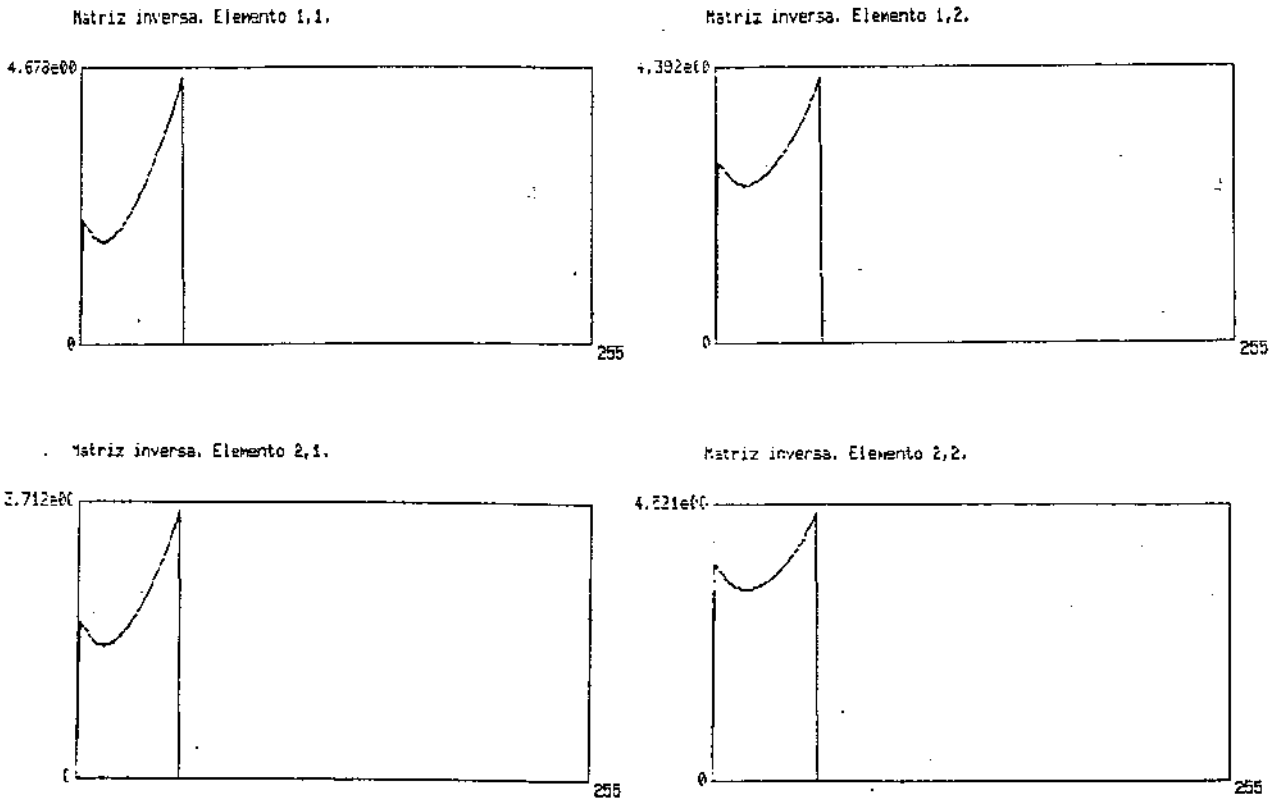


Fig. IV.17: IMRF identificada no exemplo 6. Sistema 2, linear.

Na tabela IV.3 são apresentados os valores obtidos para o erro na primeira aproximação, quantas correções (iterações) foram necessarias até a convergência, e qual foi o erro ainda existente nas variáveis após a convergência. O erro na primeira iteração é apresentado por representar um bom critério para a avaliação da influência da não linearidade. É apresentado o erro unicamente para as duas variáveis escolhidas como pontos de monitoração, nas quais se deseja obter uma resposta específica.

Tab. IV.3 - Número de iterações necessário para a convergência em função do erro na primeira aproximação.

Sistema com atrito seco.

Erro na 1ª aproximação		Nº de iterações até a converg.	Erro remanescente	
Var. 1 (%)	Var. 2 (%)		Var.1 (%)	Var. 2 (%)
2,73	2,19	2	0,60	0,06
12,87	11,12	2	0,60	0,80
257,03	224,90	5	0,64	0,07
666,67	556,24	5	0,56	0,06
1405,62	1094,38	) 20	----	----

Exemplo 8: Foi repetido o mesmo procedimento do exemplo anterior, mas agora a não linearidade simulada é correspondente à existência de rigidez quadrática (além da convencional) nas molas da estrutura (ou como foi exposto no exemplo 5, à existência de molas de rigidez quadrática agindo paralelamente as convencionais). O equacionamento correspondente é dado utilizando notação simplificada na equação (IV.6).

$$[M]\{\ddot{x}\}+[C]\{\dot{x}\}+[K]\{x\}+a[K]\{x.|x|\} = \{F\} \qquad (IV.6)$$

Utilizando o artifício de jogar as forças não lineares do lado direito da equação obtém-se:

$$[M]\{\ddot{x}\}+[C]\{\dot{x}\}+[K]\{x\} = \{F\}-a[K]\{x.|x|\} \qquad (IV.7)$$

Tab. IV.4 - Número de iterações necessários para a convergência em função do erro na primeira aproximação. Sistema com rigidez quadrática.

Erro na 1ª aproximação		Nº de iterações até a converg.	Erro remanescente	
Var. 1 (%)	Var. 2 (%)		VAR. 1 (%)	Var. 2 (%)
54,22	54,02	2	0,75	0,33
107,63	107,43	3	0,60	0,07
514,06	514,06	4	0,54	0,14
977,91	975,90	5	0,73	0,24
1791,16	1789,16	7	0,76	0,31
3654,62	3654,62	> 20	----	----

O parâmetro a permite controlar a grandeza das forças não lineares, e foi incrementado até que não se obteve convergência. Os resultados obtidos neste caso estão na tabela IV.4.

Exemplo 9: O mesmo procedimento descrito para o sistema com rigidez quadrática é utilizado agora para o mesmo sistema, mas em lugar de rigidez quadrática a não linearidade simulada é do tipo amortecimento quadrático. Chama-se amortecimento quadrático a qualquer dispositivo que introduz na estrutura forças com intensidade proporcional ao quadrado da velocidade relativa das suas extremidades, mas sempre de sentido contrário. O caso estudado supõe a existência de amortecedores de comportamento quadrático agindo em paralelo com os convencionais, ou o que é a mesma coisa, supõe que os amortecedores existentes têm comportamento quadrático superposto com o normal. O modelo matemático da estrutura

neste caso é o da equação (IV.8).

$$[M]\{\ddot{x}\}+[C]\{\dot{x}\}+[K]\{x\} = \{F\}-a[C]\{\dot{x}.|\dot{x}|\} \qquad (IV.8)$$

Tab. IV.5 - Número de iterações necessárias até a convergência em função do erro cometido na primeira iteração.

Amortecimento quadrático.

Erro na 1ª aproximação		Nº de iterações até a converg.	Erro remanescente	
Var. 1 (%)	Var. 2 (%)		Var. 1 (%)	Var. 2 (%)
175,70	167,47	3	0,68	0,12
763,05	726,91	5	0,59	0,10
447,79E3	287,15E3	> 20	----	----

Como no caso anterior, o parâmetro a permite controlar a intensidade das forças não lineares. Variando o valor de a obteve-se os resultados da tabela IV.5.

Exemplo 10: Finalmente foram rodados dois casos onde são combinados os três tipos de não linearidade descritos antes (atrito seco, rigidez quadrática e amortecimento quadrático) simultaneamente. Os dois casos se diferenciam pela amplitude das forças não lineares em cada um deles. Os resultados obtidos podem ser vistos na tabela IV.6. A matriz inversa da MRF identificada para o segundo caso deste exemplo pode ser vista na figura IV.18.

Este último exemplo mostra o bom desempenho do algoritmo mesmo para sistemas um pouco mais complexos, mas obviamente estes resultados não podem ser generalizados para qualquer caso.

É claro que um estudo exaustivo de sistemas não lineares é inviável por este método dada a infinidade de casos diferentes possíveis de serem encontrados na prática.

Tab. IV.6 - Número de iterações necessárias até a convergência em função do erro cometido na primeira iteração.

Atrito seco, rigidez quadrática e amortecimento quadrático.

Erro na 1ª aproximação		Nº de iterações até a converg.	Erro remanescente	
Var. 1 (%)	Var. 2 (%)		Var 1 (%)	Var. 2 (%)
502,00	471,89	5	0,90	0,10
1439,76	1451,81	6	0,86	0,34

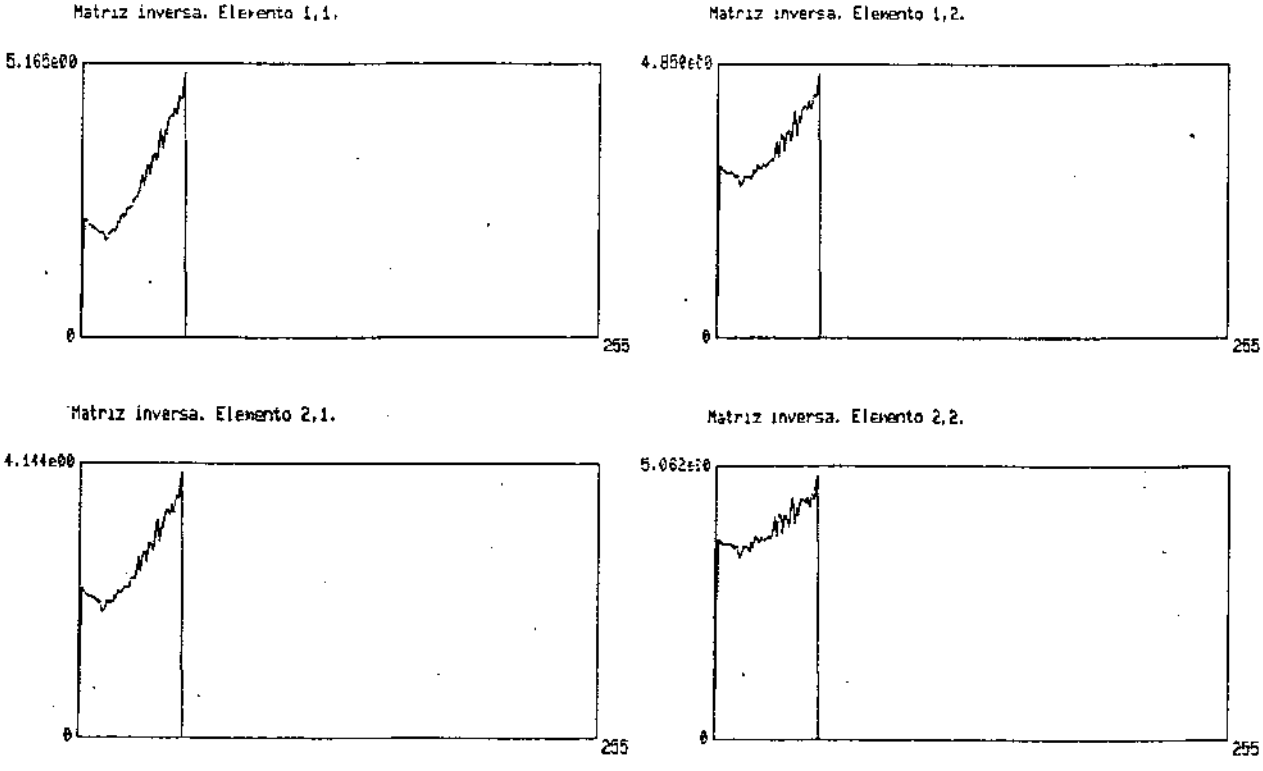


Fig. IV.18: Elementos da IMRF identificada no exemplo 10.

É também virtualmente impossível fazer um estudo teórico suficientemente genérico para que possa ser extendido a todos ou pelo menos grande parte dos casos possíveis de sistemas não lineares, principalmente se levado em consideração que estes são de equacionamento difícil, quando não impossível. É bem provável que a grande maioria dos sistemas achados na prática contenham mais não linearidades do que as que possam ser imaginadas, no que se refere ao gênero. Por sorte, em muitos deles, no entanto, a contribuição quantitativa dos efeitos não lineares é suficientemente pequena para que possa ser desprezada, facilitando consideravelmente o seu tratamento, mas não é para esses sistemas que interessa utilizar um procedimento de testes como o descrito aqui.

Devido à impossibilidade de se varrer a vastidão de casos possíveis chegou-se à conclusão de que não seria possível enriquecer consideravelmente as conclusões que já podem ser extraídas das experiências realizadas, mediante a simples realização de novas experiências com casos particulares. Em lugar disso, achou-se mais ilustrativo realizar alguma experiência simples em algum protótipo experimental.

CAPÍTULO V

MONTAGEM UTILIZADA PARA O ESTUDO EXPERIMENTAL

V.1) Introdução:

Para dar maior consistência aos resultados obtidos na simulação, foi montado um protótipo experimental.

O objetivo das experiências realizadas foi o de detectar alguns problemas que podem surgir na utilização prática do algoritmo, e que não são facilmente detectáveis na simulação. Alguns destes problemas, previsíveis *a priori*, seriam: escolha de instrumentação inadequada, "ruídos" nos sinais medidos, dificuldades na sincronização das aquisições (para evitar erros de defasagem entre os sinais correspondentes a duas aquisições diferentes).

Outro objetivo foi o de observar o desempenho do algoritmo frente à diversidade de fenômenos não lineares que podem estar influenciando as respostas, mesmo em algumas estruturas simples.

As experiências montadas limitam-se às modestas possibilidades dos recursos disponíveis. A seguir, faz-se uma descrição dos componentes do sistema utilizado para a simulação.

V.2) Estrutura:

A estrutura utilizada para as experiências, consiste de uma base de metal, sobre a qual foi colado um material flexível, formado pela superposição de dois tipos diferentes de espuma de borracha. Sobre a borracha foi colada mais uma peça de metal, que atua como massa, comprimindo-a. Existe ainda mais uma massa, também metálica, unida à anterior por meio de parafusos, e espaçada dela por pequenas peças de borracha, que agem como molas de comportamento não linear. Esta última massa pode ser removida, ou ainda a rigidez dos elementos de borracha pode ser alterada mudando o aperto dos parafusos, o que interfere no comportamento não linear da estrutura. Este recurso será novamente abordado na descrição das experiências realizadas.

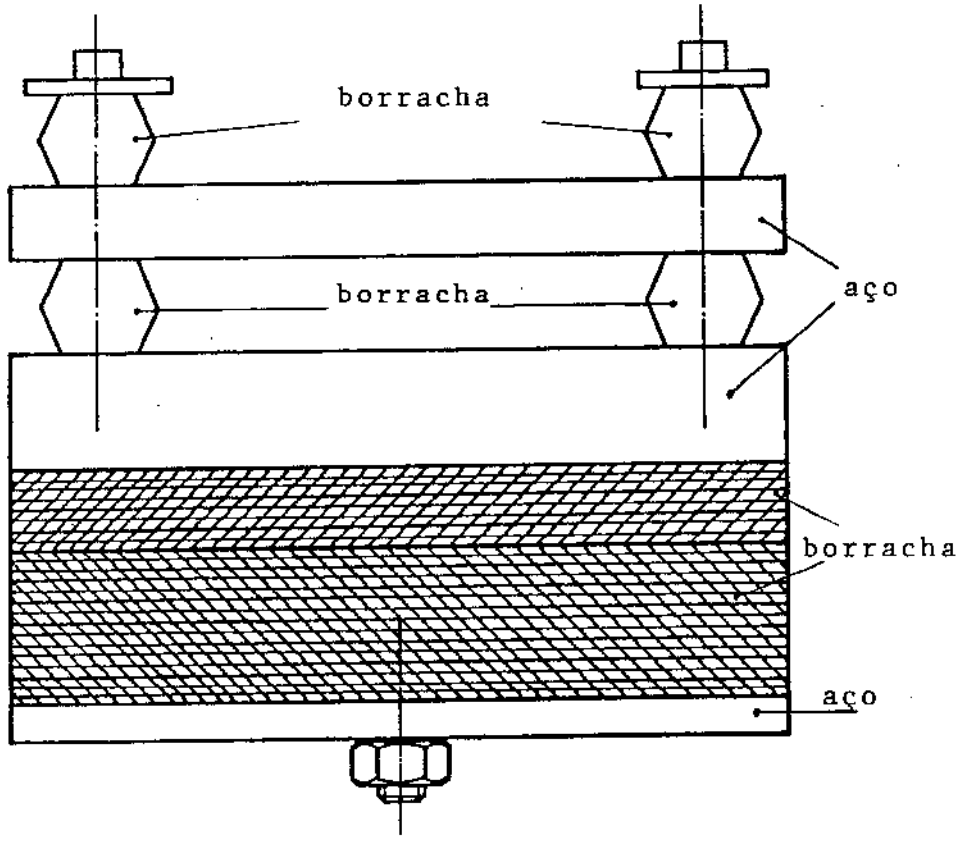


Fig. V.1) Esquema da estrutura utilizada nas experiências.

A representação esquemática da estrutura descrita pode ser observada na figura V.1, e a sua fotografia está na figura V.2.

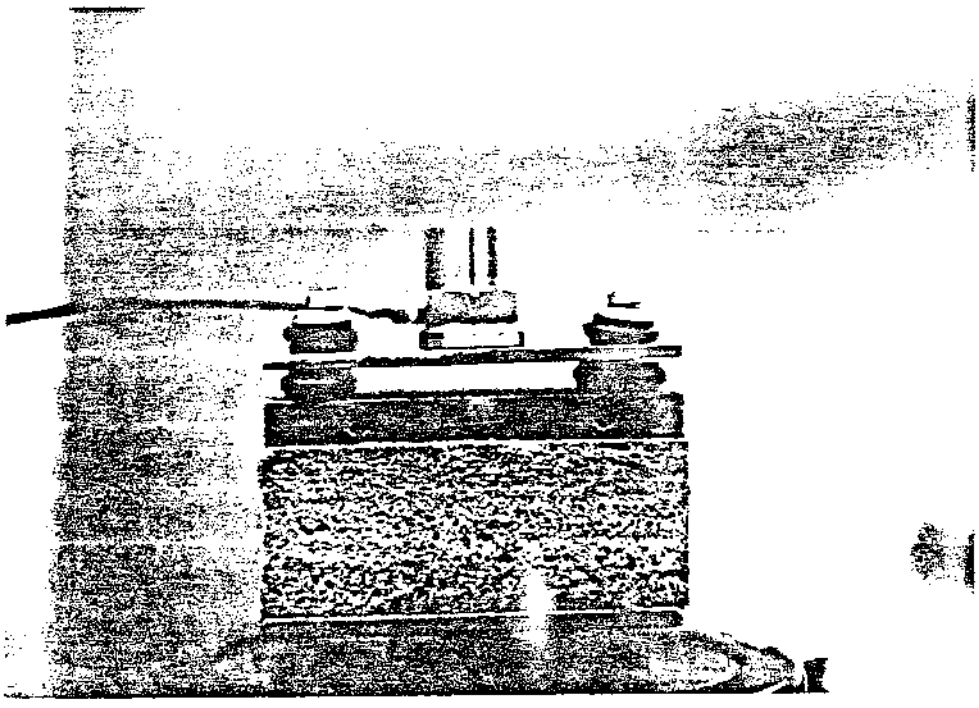


Fig. V.2) Fotografia da estrutura utilizada nas experiências

V.3) Computador:

Uma característica essencial com que um computador deve contar para a implementação do sistema de simulação, é a de possuir interfaces para comunicação com instrumentação. Mais especificamente, conversores A/D e D/A (aquisição de dados e saida analógica). Por este motivo as experiências foram efetuadas utilizando um Analisador de Fourier, da marca Hewlett Packard, modelo 5451C, único sistema com as características citadas disponível atualmente na Área de Engenharia Mecânica da UNICAMP.

O citado sistema dispõe de apenas um conversor D/A, (o que limitou as experiências a apenas um sinal de excitação), quatro canais de aquisição de dados (conversores A/D), filtros anti

"aliasing" (passa baixa), e uma infinidade de recursos que o tornam particularmente apto para o processamento e análise de sinais. Dentre os recursos disponíveis, foram de particular interesse para este trabalho, a TRF, direta e inversa, e a operação matemática em bloco entre sinais armazenados em registros discretos diferentes, tanto no domínio do tempo como em frequência, recursos estes intrínsecos ao equipamento, o que reduz o tempo de processamento.

O computador conta ainda com osciloscópio, onde são exibidos os sinais que estão sendo processados, terminal gráfico e traçador de gráficos ("plotter").

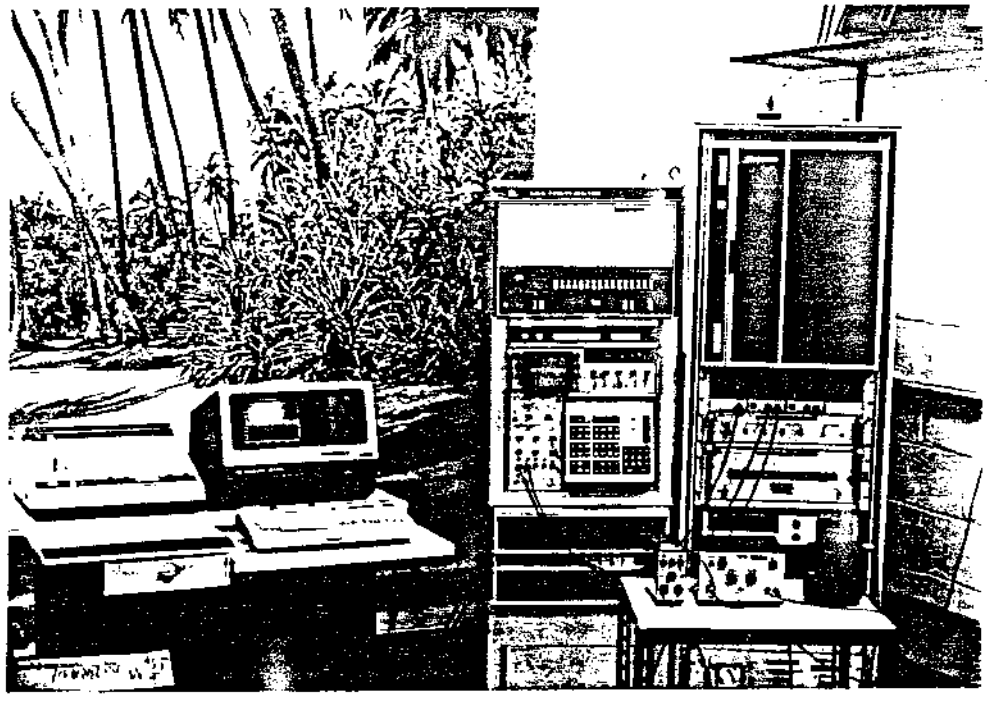


Fig. V.3) Fotografia do sistema utilizado para as experiências.

Uma fotografia do sistema pode ser observada na figura V.3.

O computador utiliza uma linguagem de programação própria, e por esse motivo foi necessário codificar uma versão simplificada (restrita a apenas um grau de liberdade) do programa de cálculo

da excitação correta, assim como do programa que gera o sinal de Schroeder. Uma listagem destes programas está no final do apêndice 3.

V.4) Atuador:

Os atuadores mais utilizados no ensaio dinâmico de estruturas são de acionamento eletrodinâmico ou hidráulico. Para esta experiência foi utilizado um atuador eletrodinâmico, mais indicado para sistemas de pequeno porte.

O atuador utilizado foi um de marca Bruel & Kjaer, modelo 4809, que pode ser utilizado para uma faixa de frequências que vai de 4Hz até 20kHz, e que fornece esforços de até 45N. O atuador é alimentado por um amplificador de potência, que é por sua vez controlado pelo sinal proveniente do computador, via conversor D/A. O amplificador de potência utilizado foi um Bruel & Kjaer modelo 2706.

V.5) Sensor.

Poderia ser utilizado, para ler as respostas do sistema, qualquer sensor que medisse alguma grandeza que mudasse na estrutura com o transcurso do tempo (forças, deformações, acelerações, etc...), como consequência das solicitações provenientes dos atuadores. Para estas experiências foi utilizado um acelerômetro, devido à facilidade de montagem e leitura do sinal. O acelerômetro utilizado foi um Bruel & Kjaer modelo 4381, com sensibilidade de 99,6 pc/g. O sinal proveniente do acelerômetro deve passar por

um condicionador de sinais. Neste caso foi utilizado um de marca Bruel & Kjaer, modelo 2635. Antes de ser enviado ao computador, o sinal deve passar por um filtro passa baixa (anti "aliasing") para evitar que o processo de discretização introduza erros no processamento do sinal, o que aconteceria se este contivesse componentes de frequências superiores à frequência de Nyquist, $1/2\Delta t$.

À primeira vista, o algoritmo pode dar a impressão de que são necessários também sensores para medir as "entradas" que agem na estrutura, já que o princípio se baseia na correção das excitações. Por exemplo, poderiam ser medidos os esforços que os atuadores transmitem à estrutura instalando uma célula de carga entre ambos. Isto não é, porém, necessário, já que o sinal de entrada a ser corrigido pode ser o proveniente do conversor D/A (e neste caso a identificação do sistema inclui o comportamento da estrutura e dos atuadores), ou o registro discreto que deu origem ao sinal analógico, que está na memória do computador (e neste caso a identificação do sistema inclui o comportamento da estrutura, dos atuadores e dos conversores D/A).

V.6) Descrição geral do sistema.

O computador efetua o processamento dos sinais, calculando o sinal de comando adequado, que é enviado, via conversor D/A, ao amplificador de potência que alimenta o atuador. As respostas obtidas no acelerômetro são enviadas, após passarem por condicionadores e filtros, ao conversor A/D do computador, para que seja efetuado novo processamento. O esquema das ligações pode ser visto na figura V.4.

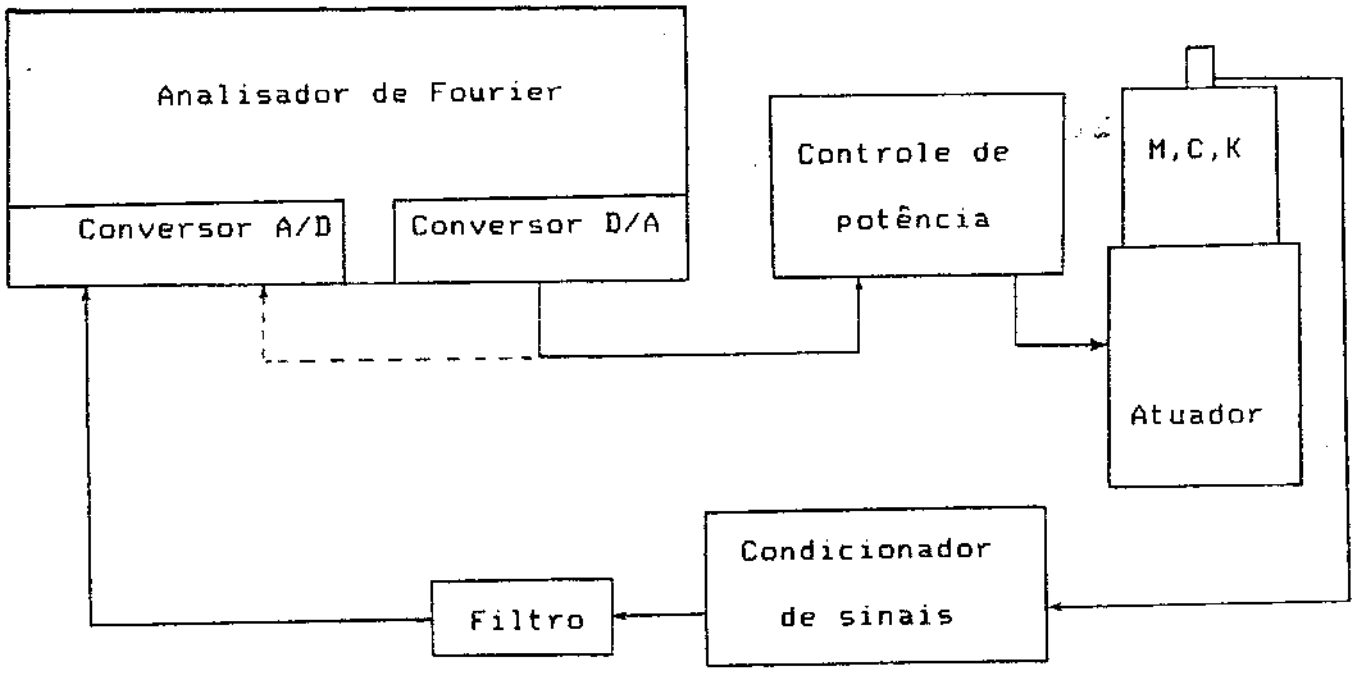


Fig. V.4) Esquema das ligações entre os instrumentos.

Dentre as diversas comparações realizadas durante as experiências, uma foi a de observar o desempenho do programa, ora adquirindo o sinal da saída analógica simultaneamente com a resposta da estrutura, ora utilizando o seu registro na memória do computador, para ser utilizado como sinal de excitação. Por este motivo a ligação entre os conversores A/D e D/A está pontilhada no esquema da figura V.4. Nos exemplos dos quais são apresentados os resultados no capítulo seguinte foi utilizada a segunda opção, que mostrou ser mais eficiente.

CAPÍTULO VI

RESULTADOS EXPERIMENTAIS

VI.1) Sincronização das aquisições:

Para obter convergência na execução do algoritmo é necessário que as aquisições sejam efetuadas sempre com a mesma fase relativa ao sinal de comando. Isto não é imediato na prática, já que devemos esperar que o sinal de resposta do sistema esteja estabilizado (livre de transientes) para se efetuar as aquisições. Num sistema especialmente adaptado às necessidades do algoritmo isto seria imediatamente resolvido mediante a geração de um pulso de período igual ao do sinal de comando, e que gerado sempre simultaneamente com este, indique o início (qualquer que seja, desde que seja sempre o mesmo) de um período do sinal de comando. Desta forma, uma vez estabilizado o sistema, seria dada a ordem para efetuar as aquisições, mas estas só começariam quando o pulso (sinal de gatilhamento) indicasse que se está no início de um período.

Este recurso não pode ser utilizado neste equipamento, dado que ele conta com apenas um conversor D/A, e além disto, por causa de uma limitação de ordem técnica, o computador não aceita,

para gatilhamento, um sinal que ele próprio gera.

Para contornar este problema, optou-se por efetuar um número sempre igual de aquisições após o início da excitação, para dar tempo à estabilização do sistema. Somente a última aquisição foi considerada válida para o processamento. Desta forma deveria transcorrer um lapso de tempo sempre igual entre o início da excitação e o início da aquisição da resposta a ser utilizada no processamento, o que garantiria uma defasagem constante entre excitações e respostas, satisfazendo assim as exigências do algoritmo. Porém, o tempo necessário para efetuar o mesmo número de aquisições mostrou ligeiras alterações de um evento para outro, possivelmemte por causa dos processos internos do computador. Estas diferenças de tempo ocasionaram variações entre as defasagens obtidas entre excitações e respostas para diferentes aquisições. Estas diferenças de fase mostraram sér pouco significativas para frequências mais baixas, mas ocasionaram alguns transtornos para frequências superiores a 30 Hz. Por esse motivo a faixa de análise foi limitada, para os exemplos apresentados, a no máximo 25 Hz, o que constitui provavelmente a mais séria limitação das experiências realizadas.

Devido ao atuador não conseguir operar em frequências inferiores a 4 Hz, a faixa de frequências de análise foi limitada entre 5 e 25 Hz.

VI.2) Opções para o sinal de referência (excitação).

Foram identificadas três opções para o sinal de excitação a ser utilizado para o cálculo da MRF. O sinal proveniente de sensores colocados entre os atuadores e a estrutura (por exemplo, células de carga), o sinal de comando dos atuadores, provenientes dos conversores D/A, e os sinais digitalizados e armazenados na memória do computador que deram origem aos sinais de comando, via conversor D/A.

A primeira opção foi descartada *a priori* por vários motivos, alguns dos quais são enumerados a seguir.

- Exige maior número de sensores, complicando assim a instrumentação.

- Exige medição de um número maior de parâmetros, aumentando os erros aleatórios de medida. Por exemplo, o ruído nos sinais.

- O sistema que a MRF identifica neste caso inclui apenas a estrutura e os sensores, não percebendo o comportamento dos atuadores e dos conversores D/A.

A segunda opção permite simplificar a instrumentação e aparentemente resolve o problema da defasagem entre excitação e resposta na hora de se efetuar a identificação, já que ambos os sinais são adquiridos simultaneamente. Ela é eficiente, com efeito, para a identificação do sistema, mas cria dificuldades quando começa o processo iterativo de correção dos sinais. Isto porque o processo de transformação do sinal discretizado em sinal analógico envolve um atraso de fases, que não é identificado pela MRF obtida. Mas a correção é efetuada sobre o sinal discreto armazenado na memória e não sobre o sinal analógico. Por isso, a cada

correção comete-se um erro sistemático de fases. Além disso, uma vez iniciado o processo de correção do sinal, seria necessário utilizar um sinal de gatilhamento, como no caso anterior, já que a aquisição simultânea de excitação e resposta é válida somente na fase de identificação.

Uma última crítica a esta opção é que a identificação do sistema não avalia o comportamento do conversor D/A.

Devido às deficiências apontadas para as duas opções anteriores, optou-se pela terceira delas, embora tenham sido feitas algumas tentativas com a segunda. Portanto, a MRF identificada estabelece a relação existente entre o sinal proveniente dos sensores que monitoram as respostas, adquiridas através dos conversores A/D, e o sinal de excitação, que está armazenado na memória do computador, discretizado.

VI.3) Experiências efetuadas.

Foram realizadas várias experiências, onde tentou-se mudar as caraterísticas da estrutura e do sinal desejado como resposta.

Para mudar as características da estrutura contou-se com a ajuda dos parafusos, que permitem remover a peça superior, colocá-la com os parafusos apertados (neste caso a rigidez das borrachas aumenta, diminuindo os deslocamentos, e consequentemente também a influência das não linearidades), com os parafusos mais frouxos, ou ainda totalmente soltos, deixando uma folga apreciável entre as peças, o que aumenta, evidentemente, as não linearidades.

Com relação ao sinal desejado como resposta do sistema às excitações, tentou-se variar o máximo possível. Pediu-se ao sistema que reproduzisse uma pancada (figura VI.1), várias pancadas consecutivas (figura VI.2), ou o sinal obtido balançando a estrutura aleatoriamente (figura VI.3). Variaram-se também as amplitudes dos sinais desejados, dentro dos limites possíveis para a instrumentação utilizada.

Para todos os casos rodados com o sistema na sua configuração final, e respeitando as limitações de amplitude e faixa de frequências mencionadas, obteve-se convergência para o sinal desejado. Na tabela VI.1 são apresentados os resultados obtidos em alguns dos exemplos rodados. Nas figuras VI.4, VI.5 e VI.6 são mostrados alguns gráficos dos sinais em um caso típico, que pode ser considerado representativo do acontecido nos outros casos.

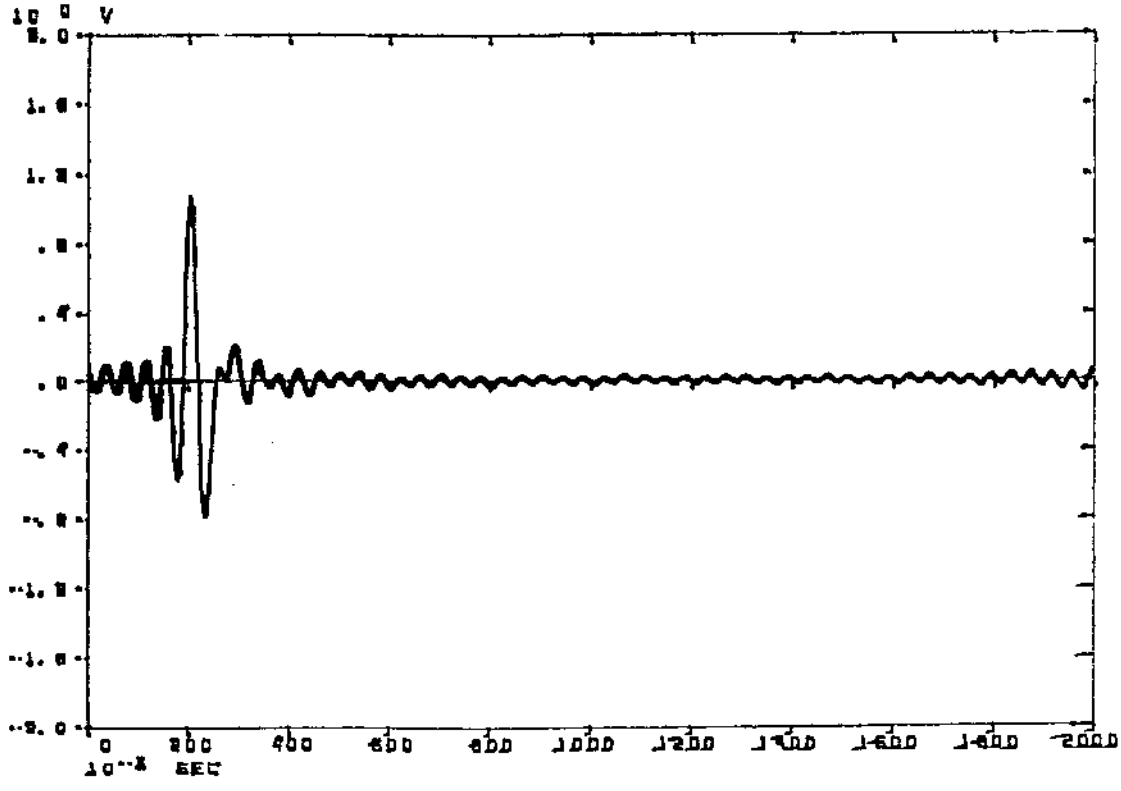


Fig. VI.1) Exemplo de resposta da estrutura a uma pancada.

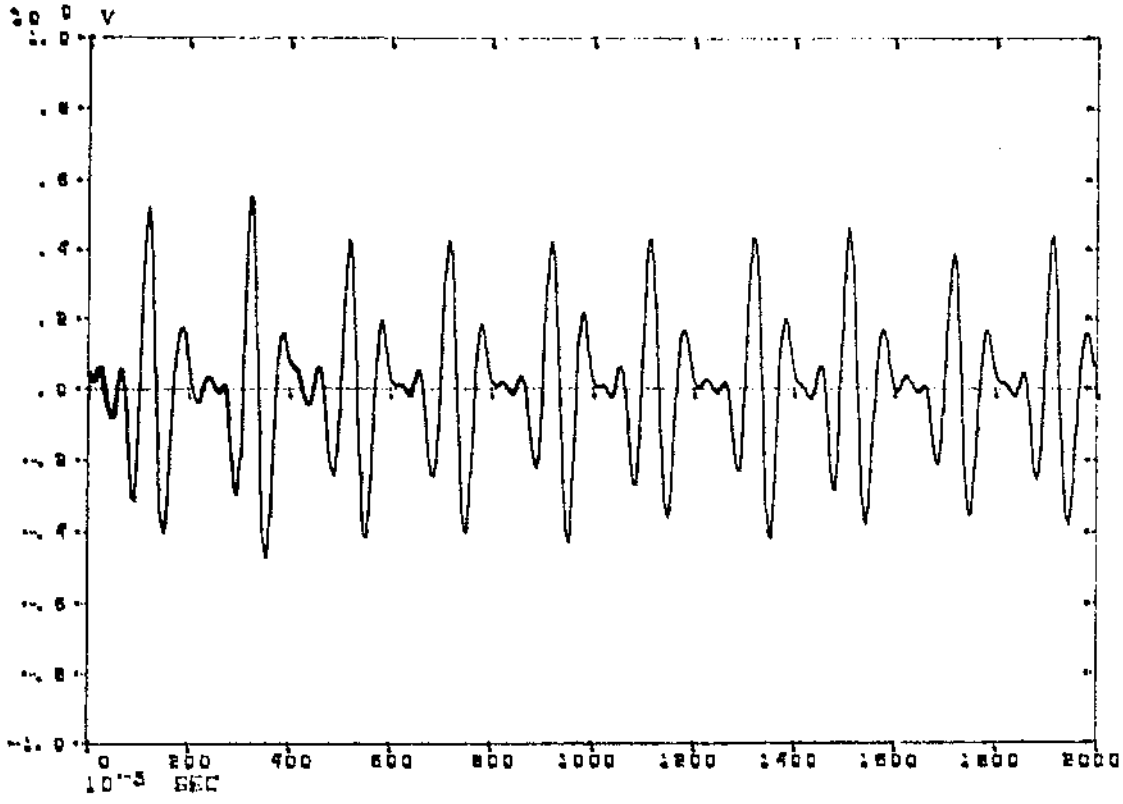


Fig. VI.2) Resposta a várias pancadas consecutivas.

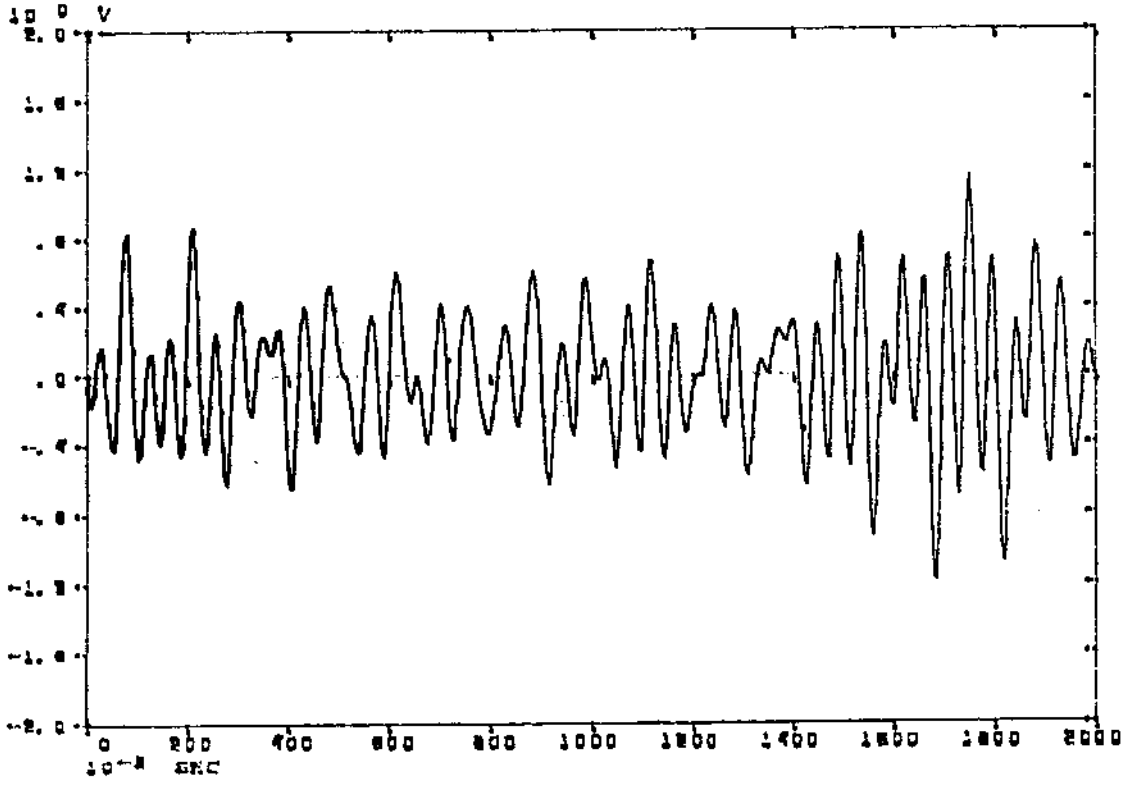


Fig. VI.3) Resposta da estrutura a uma excitação aleatória.

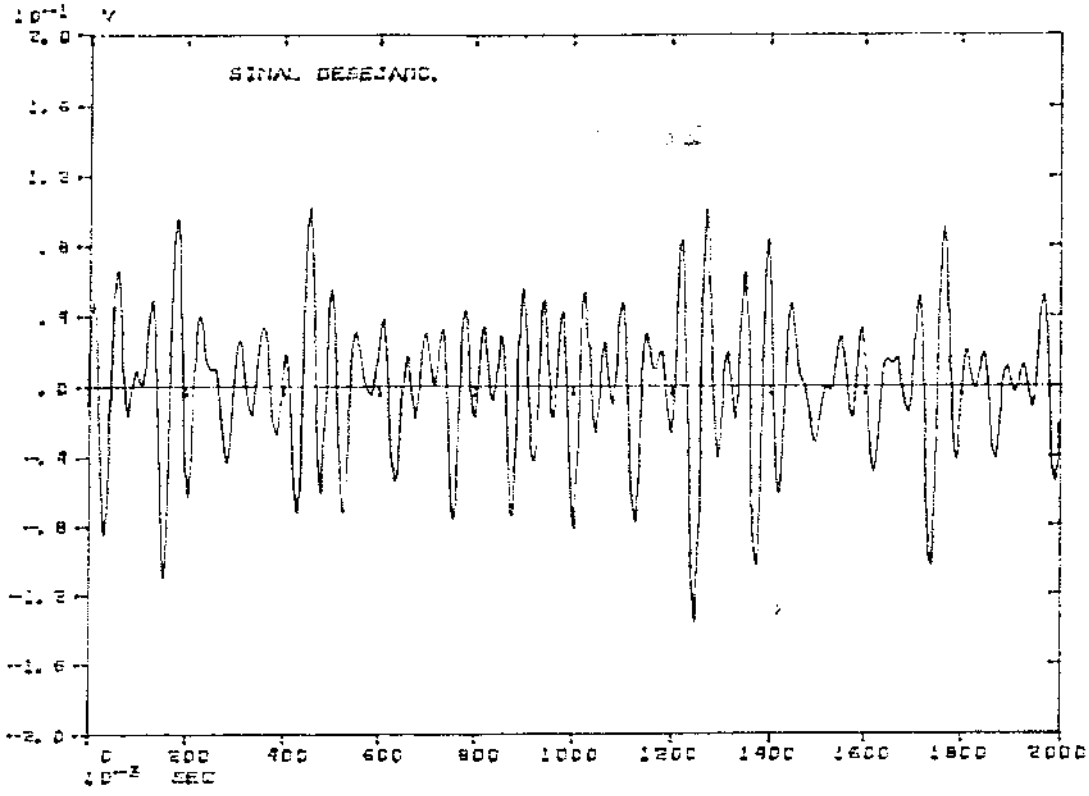
Nos exemplos 1,2,3,4,5,6,9 e 10 o sinal desejado como resposta do sistema, foi o registrado pelo acelerômetro, batendo com a mão em cima da estrutura. Nos exemplos 7,8,11 e 12 o sinal desejado foi o obtido no acelerômetro balançando aleatoriamente a estrutura.

Os parafusos que seguram a parte superior da estrutura estavam apertados nos exemplos 1 e 2, menos apertados nos exemplos 3 e 4, sem aperto nos exemplos 5,6,7 e 8, e com folga nos exemplos 9, 10, 11 e 12.

Os exemplos que não se diferenciam pela situação da estrutura (aperto dos parafusos) nem pelo tipo do sinal desejado como resposta, se diferenciam pela amplitude do sinal desejado, como pode ser observado pelo valor do erro admitido em cada caso, que é sempre igual a 1% do valor médio do quadrado das amplitude do sinal.

Tab. VI.1) Alguns resultados experimentais, onde pode ser observado o erro admitido e o obtido em cada iteração, em valor absoluto e em % do valor médio do quadrado do sinal desejado.

Ex.	Erro adm.	1ª it.	2ª it.	3ª it.	4ª it.	5ª it.	6ª it
1	18	139	14				
	1%	7,7%	0,8%				
2	22	567	193	20			
	1%	25,8%	8,8%	0,9%			
3	16	119	13				
	1%	7,4%	0,8%				
4	27	252	50	6			
	1%	9,3%	1,8%	0,2%			
5	39	33					
	1%	0,8%					
6	57	968	265	42			
	1%	16,9%	4,6%	0,7%			
7	38	43	141	131	129	11	
	1%	1,1%	3,7%	3,4%	3,4%	0,3%	
8	57	458	299	7			
	1%	8,0%	5,2%	0,1%			
9	49	634	161	30			
	1%	12,9%	3,3%	0,6%			
10	74	3095	1549	677	228	90	4[illegible]
	1%	41,8%	20,9%	9,1%	3,1%	1,2%	0,5%
11	37	305	43	4			
	1%	8,2%	1,2%	0,1%			
12	98	1556	731	28			
	1%	15,9%	7,5%	0,3%			



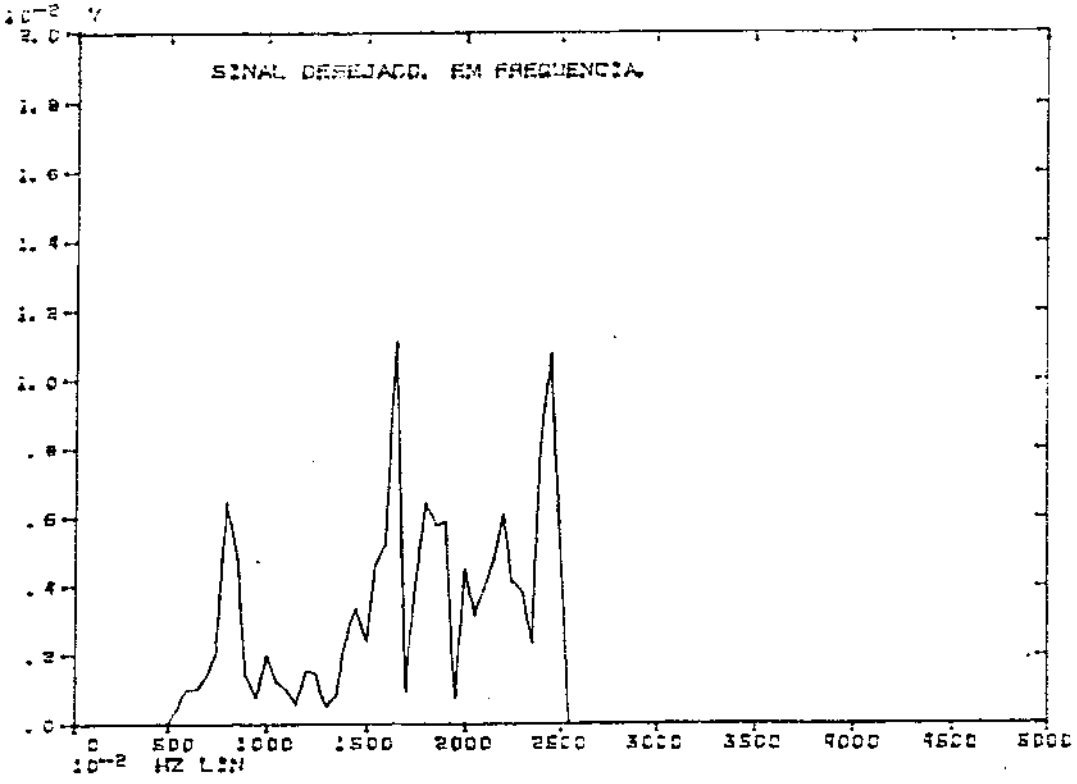
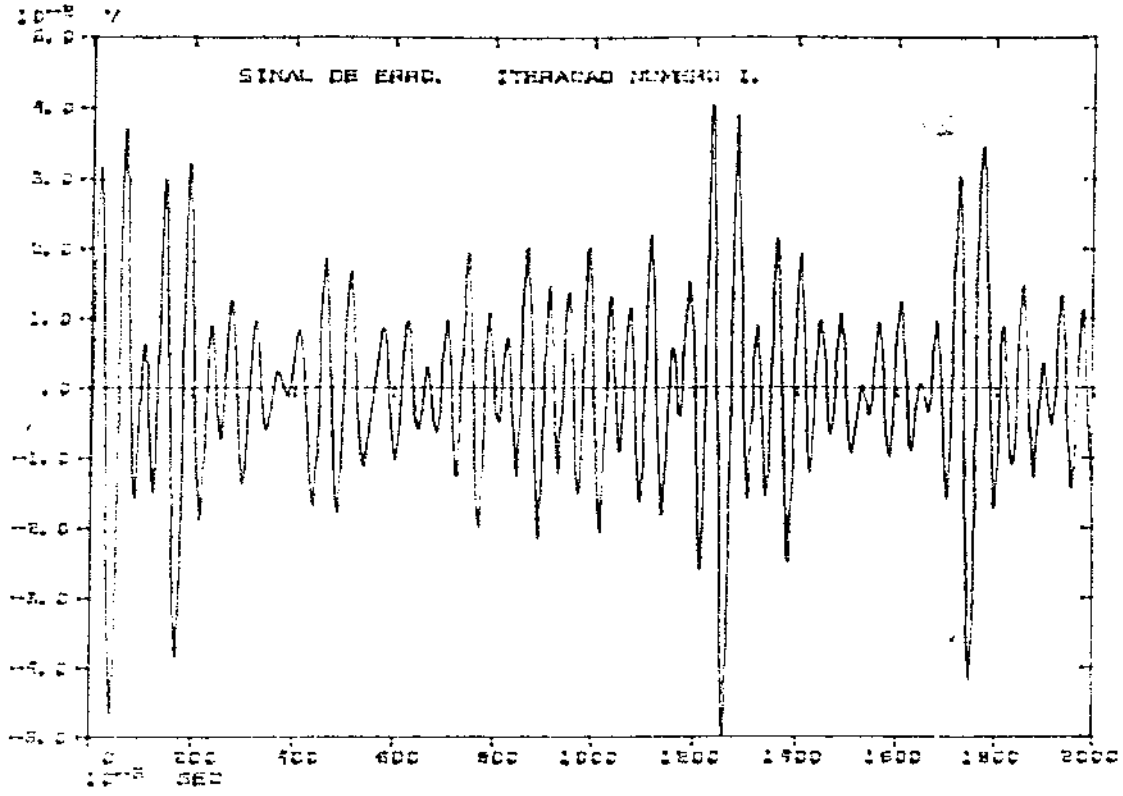


Fig VI.4) Exemplo de sinal desejado, no tempo e em frequência.



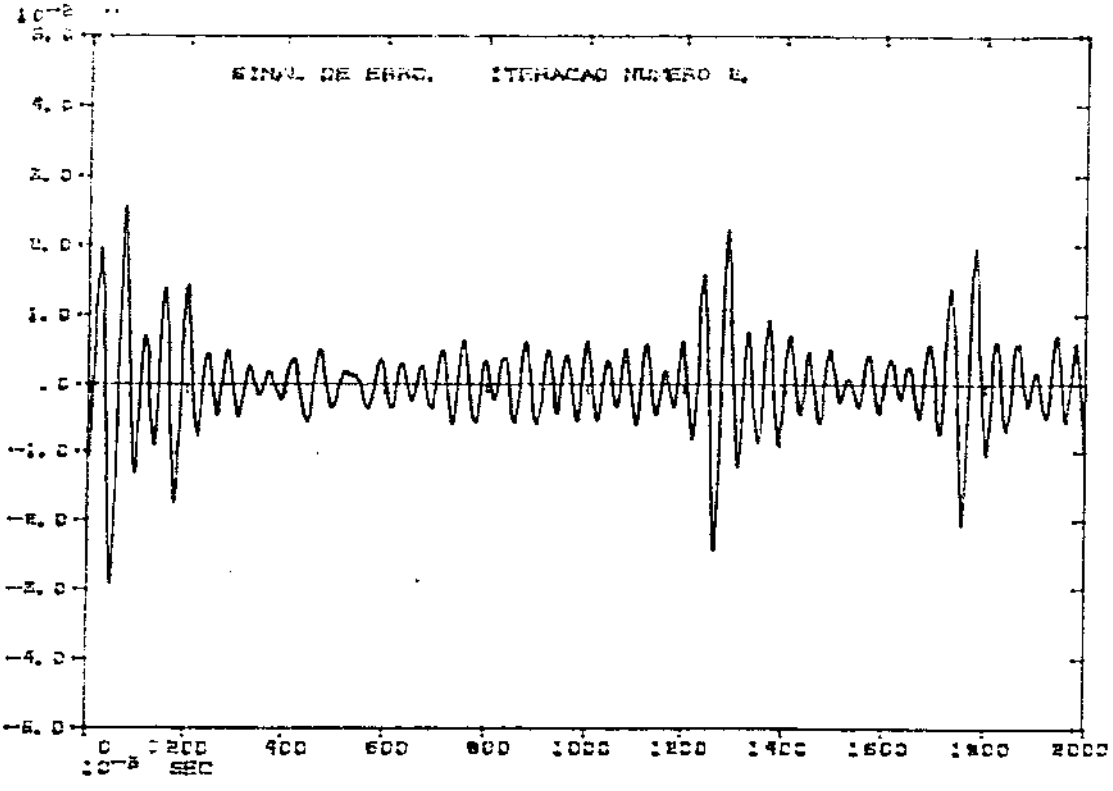


Fig. VI.5) Evolução do sinal de erro no exemplo da figura VI.4.

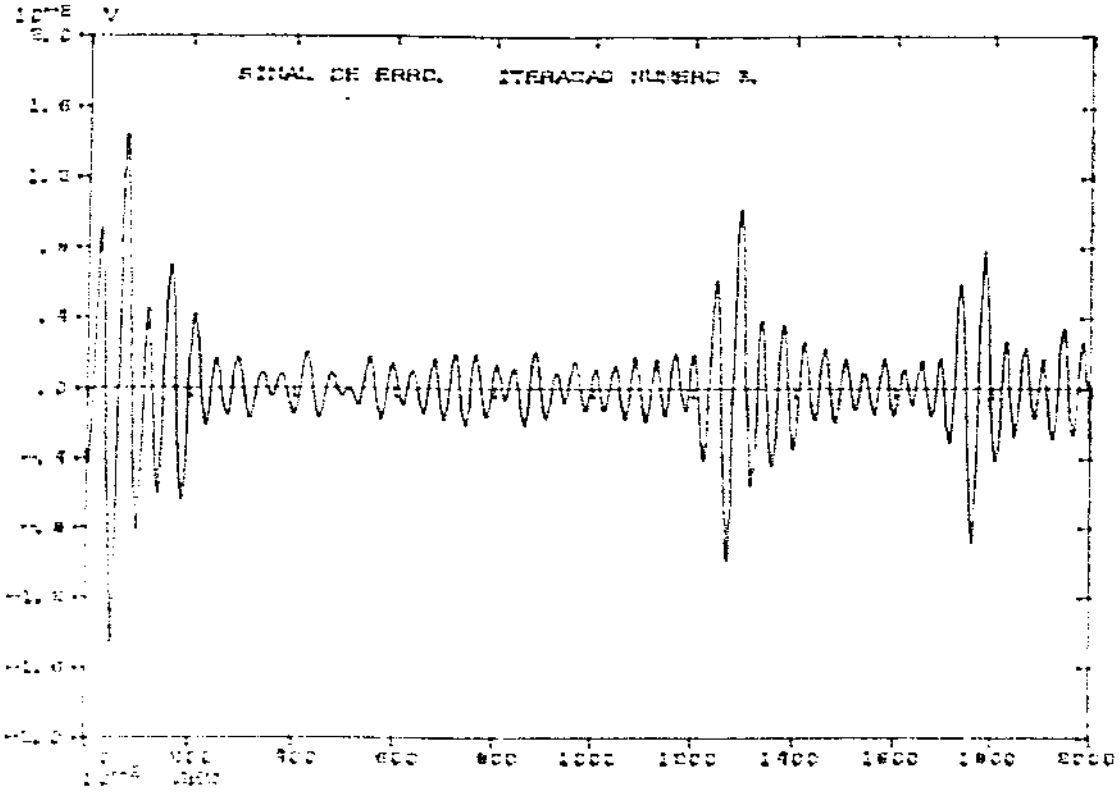


fig. VI.5) Continuação.

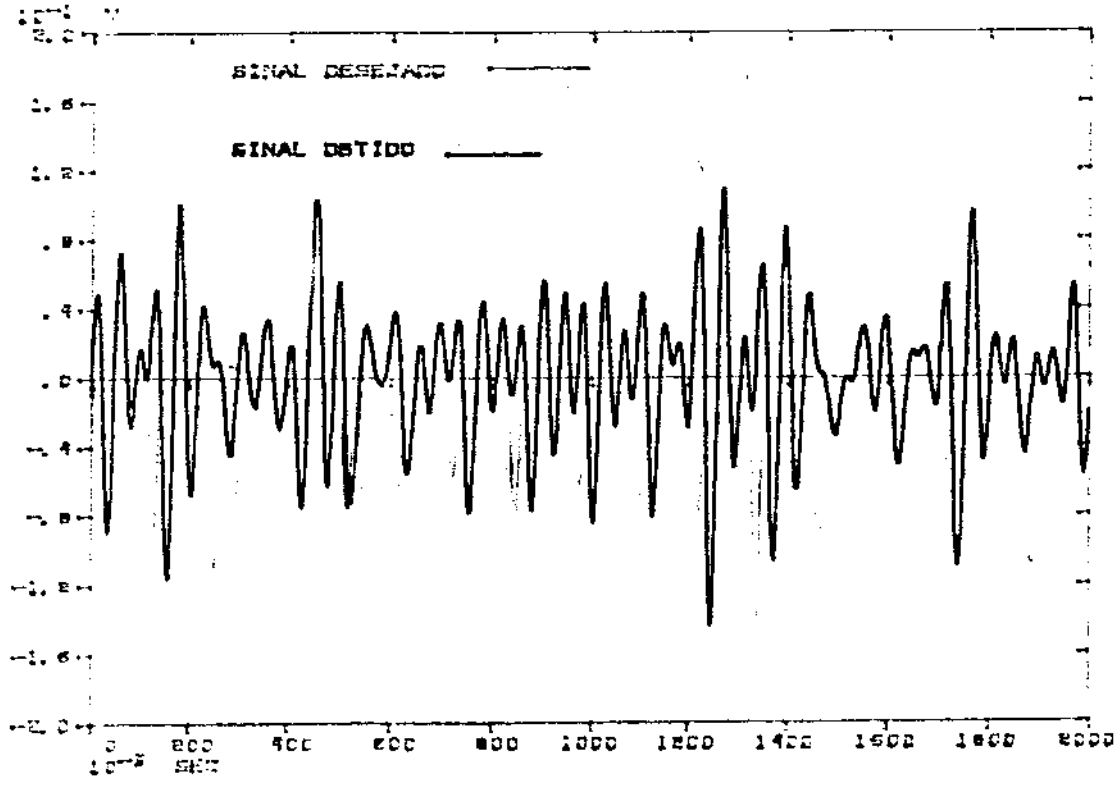


Fig. VI.6) Superposição do sinal desejado com o sinal obtido após 3 iterações.

CAPÍTULO VII

CONCLUSÕES

VII.1) Introdução.

O desenvolvimento deste trabalho representa um avanço significativo no sentido da implementação de um sistema de testes (ou de simulação de determinadas situações) baseado no princípio chamado CESR neste trabalho. Além de mostrar algumas caraterísticas, limitações e possíveis aplicações para estes sistemas, este trabalho mostra algumas exigências de "hardware", assim como também alguns procedimentos que devem ser seguidos, para que se obtenha sucesso no uso do sistema.

Apesar das limitações de ordem técnica enfrentadas na realização das experiências, os resultados obtidos forneceram importante subsídio com vias à implementação deste sistema em equipamentos com recursos para suprir as suas necessidades específicas, e que possibilitem assim o aproveitamento prático do sistema na simulação de respostas em estruturas de interesse real.

O método de simulação por CESR tem diversas aplicações. Algumas delas são:

- testes de estruturas de veículos (rodoviários, ferro-

viários, aeronáuticos e espaciais).

- testes de estruturas em geral.
- testes de partes de veículos ou outras estruturas.
- simulação de ondas do mar.

Algumas conclusões merecem especial atenção pela sua importância para o sucesso e bom aproveitamento do método, e por este motivo serão sumarizadas a seguir.

VII.2) Relativo aos equipamentos.

No capítulo I, ítem I.6, são enumerados os componentes essenciais a um equipamento de testes que funcione baseado no princípio do CESR. Para efetuar os testes, o operador do sistema deve levar em consideração algumas características destes equipamentos. Por exemplo, deve verificar que a faixa de frequências e as amplitudes dos esforços e dos deslocamentos solicitados dos atuadores não excedam as possibilidades dos mesmos. Algumas situações que não podem ser simuladas devido às limitações físicas do sistema (por exemplo, as acelerações transversais ou longitudinais registradas num automóvel durante um "slalom" ou uma freada forte, respectivamente, que exigiriam grandes deslocamentos para serem simuladas, nos casos em que as únicas reações envolvidas são as forças originadas pela inércia das massas da estrutura) requerem um tratamento prévio do sinal para que possam ser aproximadamente reproduzidas. Estes procedimentos formam parte da edição do sinal.

Um aspecto importante é que devem ser previstos recursos do equipamento que garantam a sincronização das aquisições, para

evitar defasagens que prejudiquem o desempenho do sistema.

É aconselhável que o "sistema" a ser identificado para a execução dos testes não se limite apenas à estrutura a ser ensaiada. A identificação deve incluir também o comportamento de atuadores, sensores, conversores e qualquer outro componente do sistema que possa interferir na relação entre sinal de comando e resposta.

Normalmente as máquinas universais de ensaios dinâmicos contam com portas de entrada de sinais analógicos, que são utilizados como referência para as grandezas monitoradas nos atuadores. Portanto o sistema de testes baseado no princípio do CESR pode ser constituído por uma destas máquinas ligada a um computador, no qual seriam processados os sinais e calculados os sinais analógicos a serem enviados à máquina de testes. Neste caso, deve se analisar cuidadosamente se deve ser mantido o controle próprio da máquina de testes. Este controle geralmente tem constante de tempo bastante longa, para garantir a estabilidade, e por esse motivo consegue acompanhar apenas variações graduais dos sinais. É possível que existam problemas com sinais de comando que mudam constantemente de frequências e amplitudes. Por outro lado, o processo CESR é ele próprio um sistema de controle, do tipo "off-line".

Deve-se tomar precauções para que os níveis de tensão dos sinais de comando sejam compatíveis com as exigências e limitações dos atuadores, criando sistemas de segurança que protejam o sistema e a estrutura a ser ensaiada.

VII.3) Relativo ao processamento dos sinais.

Um dos recursos que confere grande versatilidade ao sistema é a possibilidade de se efetuar a edição dos sinais registrados em campo antes de dar início aos testes. A edição dos sinais permite criar sinais que representem estatisticamente um conjunto de registros obtidos em medições diferentes ou até em condições diferentes. Também permite reproduzir aproximadamente situações irreprodutíveis sem a ajuda de algum artifício, como é o caso do "slalom" ou da freada de um automóvel. Nestes dois casos específicos, o que pode ser feito é uma compressão dos sinais no tempo, ou seja, criar um sinal que tenha os mesmos níveis de amplitude (por exemplo, na aceleração transversal ou longitudinal) mas agindo por um intervalo de tempo consideravelmente menor, valores estes possíveis de serem atingidos com o equipamento disponível. Este recurso é descrito em [30].

Também podem ser escolhidos os trechos mais representativos de um sinal para serem reunidos num mesmo bloco, descartando os trechos carentes de interesse. Desta forma seria criado um sinal fictício que contém aqueles fenômenos que é interessante repetir, sem necessidade de se trabalhar com blocos que contenham períodos de tempo muito longos, e que exigem portanto um número de pontos de discretização muito alto.

Uma precaução que deve ser tomada, na edição dos sinais, é a de se evitar o aparecimento de descontinuidades nos sinais, por exemplo entre o início e o fim de cada bloco.

APÊNDICE 1

ALGUMAS FERRAMENTAS PARA ANÁLISE DE SINAIS

A1.I) ANÁLISE DE FOURIER.

De acordo com a teoria de Fourier, é possível expressar qualquer função do tempo x(t), periódica de período T, que satisfaz as condições de Dirichlet e tal que a expressão $\int_0^\infty |x(t)|dt$ exista (finito), como a soma infinita de uma série trigonométrica [32], que, utilizando notação exponencial, é mostrada na equação:

$$x(t) = \sum_{k=-\infty}^{\infty} X_k . e^{i2\pi kt/T} \qquad (A1.1)$$

Os coeficientes X_k da série mostrada na equação (A1.1) são chamados coeficientes de Euler-Fourier, e representam a amplitude das componentes de frequência k.1/T (Hz) presentes no sinal x(t). Os coeficientes X_k podem ser calculados utilizando a equação:

$$X_k = \frac{1}{T} \int_{-T/2}^{T/2} x(t) . e^{-i2\pi kt/T} \qquad (A1.2)$$

É possivel estender o conceito para sinais não periódicos (incluidos os transitórios), estipulando que eles têm período infinito. Neste caso chega-se às expressões:

$$x(t) = \int_{-\infty}^{\infty} X(f).e^{i2\pi ft}df \qquad (A1.3)$$

$$X(f) = \int_{-\infty}^{\infty} x(t).e^{-i2\pi ft}dt \qquad (A1.4)$$

A expressão (A1.4) é chamada transformada de Fourier do sinal x(t), enquanto que a expressão (A1.3) é a transformada de Fourier inversa.

Para aproveitar as vantagens oferecidas pelo processamento digital de sinais mediante o uso de computadores, define-se a transformada de Fourier discreta, direta e inversa (TFD e TFDI), que podem ser obtidas utilizando as expressões (A1.5) e (A1.6) repectivamente.

$$X_k = \frac{1}{N}\sum_{r=0}^{N-1} x_r e^{-i2\pi kr/N} \qquad k = 0,..,(N-1) \qquad (A1.5)$$

$$x_r = \sum_{k=0}^{N-1} X_k e^{i2\pi kr/N} \qquad r = 0,..,(N-1) \qquad (A1.6)$$

N é o número de pontos em que foi discretizado um período T de tempo, x_r é a amplitude do sinal no tempo no instante $t = r.\Delta t = r.T/N$, e X_k é a amplitude da harmônica de frequência $f = k.\Delta f = k.1/T$ presente no sinal x(t).

O processamento digital dos sinais requer alguns cuidados, caso contrário ele conduz a resultados erroneos. Algumas das precauções que devem ser tomadas são:

- evitar a presença, no sinal, de componentes de frequência superior a $1/(2.\Delta t)$ (frequência de Nyquist). Se isto acontecer, ocorrerão distorções no espectro por "aliasing", também chamado rebatimento, devido à discretização inadequada. Para evitar este problema utilizam-se filtros passa-baixa com frequência de corte conveniente.

- discretizar sempre um número inteiro de períodos do sinal x(t). Caso contrário acontece erro de "leakage" [31], e tanto as amplitudes como as frequências obtidas diferirão da verdadeira composição espectral de x(t).

O processamento da TFD requer um número de operações matemáticas muito grande, e consequentemente, um tempo de processamento longo. Para minimizar este problema existem algoritmos que calculam a TFD e TFDI, utilizando para isso um número consideravelmente menor de operações, e consumindo portanto menos tempo. Estes algoritmos são conhecidos pela denominação genérica de transformada de Fourier rápida, direta e inversa (TRF e TRFI respectivamente). Para este trabalho foi utilizado o algoritmo de Cooley e Tukey, também chamado algoritmo da borboleta. Mais detalhes sobre este algoritmo podem ser achados em [22].

A1.II) COVARIANÇA, CORRELAÇÃO E COEFICIENTE DE CORRELAÇÃO.

Sejam dois sinais obtidos de processos aleatórios estacionários, x(t) e y(t). A função de covariança de x(t) e y(t) é definida de acordo com a expressão (A1.7).

$$C_{xy}(\tau) = E[(x(t) - u_x).(y(t+\tau) - u_y)] = R_{xy}(\tau) - u_x u_y \quad (A1.7)$$

As variáveis u_x e u_y são os valores esperados (valores médios) de x(t) e y(t) respectivamente.

$R_{xy}(\tau)$ é definida pela expressão:

$$R_{xy}(\tau) = \lim_{T\to\infty} \frac{1}{T}\int_0^T x(t).y(t+\tau)dt \qquad (A1.8)$$

Se x(t) e y(t) são diferentes, $R_{xy}(\tau)$ é chamada correlação cruzada das variáveis x(t) e y(t). Se x(t) = y(t), $R_{xx}(\tau)$ é chamada função de autocorrelação de x(t).

Define-se o coeficiente de correlação $\rho(\tau)$ de acordo com a expressão (A1.9) [18].

$$\rho_{xy}^2(\tau) = \frac{C_{xy}^2(\tau)}{C_{xx}(0)C_{yy}(0)} = \frac{(R_{xy}(\tau) - u_x u_y)^2}{(R_{xx}(0)-u_x^2)(R_{yy}(0)-u_y^2)} \qquad (A1.9)$$

O coeficiente de correlação $\rho_{xy}(\tau)$ estabelece a relação causa - efeito, linear, de dois sinais x(t) e y(t).

Para qualquer valor de τ vale: $0 \leq \rho_{xy}^2(\tau) \leq 1$.

Se $\rho_{xy}^2(\tau) = 0$, os sinais x(t) e y(t) são completamente independentes. Se por outro lado $\rho_{xy}^2(\tau) = 1$, x(t) é uma consequência unicamente de y(t) (ou vice-versa), num processo linear.

O coeficiente de correlação pode ser utilizado para saber se um sinal x(t) periódico, de período T, está estabilizado (livre da influência de transientes). Em tal caso $\rho_{xx}(T)$ deve ter valor numérico muito próximo da unidade.

A1.III) Densidade espectral de potência (DEP).

A DEP fornece uma relação da quantidade de energia contida em cada faixa de frequências presente num determinado sinal.

Conceitualmente, a DEP de um sinal x(t) seria obtido da seguinte maneira: o sinal x(t) seria filtrado num filtro passa banda com frequência central (variável) f e largura de banda Δf. Calcula-se então o valor médio quadrático do sinal emergente do filtro, $x(f,\Delta f,t)$, e divide-se o resultado por Δf. Efetuando-se o limite para $\Delta f \to 0$ da expressão resultante, obtém-se a quantidade de energia contida na frequência f. A expressão resultante é mostrada na equação (A1.10)

$$G_{xx}(f) = \lim_{\Delta f \to 0} \frac{1}{\Delta f . T} \int_0^T x^2(f,\Delta f,t)dt \qquad (A1.10)$$

$G_{xx}(f)$ é o espectro de potência de x(t), e é definido para $f \geq 0$.

Define-se a DEP simétrica, $S_{xx}(f)$, de modo que ela exista para qualquer f, seja simétrica com relação à origem f = 0, e que para $f \geq 0$, valha:

$$S_{xx}(f) = G_{xx}(f)/2 \qquad f \geq 0 \qquad (A1.11)$$

A extensão da definição para frequências negativas não tem significado físico, e é apenas um recurso matemático. Porém, esta extensão não introduz nenhuma alteração conceitual, já que a quantidade de energia contida em cada frequência permanece inalterada.

A variável DEP $S_{xx}(f)$ assim definida é frequentemente chamada "auto - espectro", por envolver apenas uma variável x(t). Um procedimento semelhante pode ser aplicado a duas variáveis x(t) e y(t), obtendo-se a DEP cruzada, $S_{xy}(f)$, das duas variáveis [35].

Na prática, para se obter a DEP, pode-se utilizar o processo de filtragem descrito, mas é mais usual utilizar outros procedimentos. Pode ser visto em [29] que a DEP pode ser obtida mediante a transformada de Fourier da função de correlação $R_{xy}(\tau)$, ou ainda a partir das transformadas de Fourier dos sinais x(t) e y(t).

Extendendo o conceito para sistemas com várias entradas e várias saídas, definem-se as matrizes de DEP:

$$S_{xx}(f) = \lim_{T\to\infty} T.E[(X(f)).(X(f))^{ct}] \qquad (A1.12)$$

$$S_{xy}(f) = \lim_{T\to\infty} T.E[(X(f)).(Y(f))^{ct}] \qquad (A1.13)$$

onde ct denota o conjugado transposto.

Na prática, calcula-se uma aproximação de $S_{xx}(f)$ e $S_{xy}(f)$, que será chamada $\underline{S}_{xx}(f)$ e $\underline{S}_{xy}(f)$, utilizando-se um tempo de análise suficientemente longo para que tenha validade estatística. Para este fim efetuam-se r aquisições consecutivas de duracão T [29]. A expressão resultante pode ser vista nas equações (A1.14) e (A1.15).

$$\underline{S}_{xx}(f=k/T) = \frac{T}{r}\sum_{k=1}^{r}(X_k).(X_k)^{ct} = T.E[(X_k).(X_k)^{ct}] \qquad (A1.14)$$

$$\underline{S}_{xy}(f=k/T) = \frac{T}{r}\sum_{k=1}^{r}(X_k).(Y_k)^{ct} = T.E[(X_k).(Y_k)^{ct}] \qquad (A1.15)$$

APÊNDICE 2

SINAL DE SCHROEDER

Para inúmeras aplicações é interessante contar com um sinal que possua uma DEP pré-estabelecida e que tenha baixo fator de pico. O fator de pico de um sinal é definido em [24] como a diferença das amplitudes máximas e mínimas do sinal, dividida pelo seu valor médio quadrático.

Schroeder propõe em [24] um método para obter um sinal com estas características, mediante o cálculo dos ângulos de fase mais adequados para todas as componentes do sinal.

Seja r(t) um sinal periódico de período T e com expansão em série de Fourier com número N, finito, de termos não nulos.

$$r(t) = \sum_{k=1}^{N} (p_k/2)^{1/2} \cdot \cos(2\pi k t/T + a_k) \qquad (A2.1)$$

p_k é a contribuição relativa da k<u>ésima</u> harmônica na DEP do sinal (A somatória de todos os p_k, com k = 1,..,N, é igual a 1). a_k é o ângulo de fase da k<u>ésima</u> componente. Dados os p_k, devemos calcular os a_k correspondentes que minimizem o fator de pico do sinal.

Considere-se o sinal s(t), modulado em frequência, que é dado na equação (A2.2).

$$s(t) = \cos(B(t)) \qquad (A2.2)$$

onde B(t) obedece à relação:

$$B(t) = \int_0^t \frac{2\pi k z}{T} dz \qquad (A2.3)$$

O instante t_k em que a frequência instantânea muda de sinal é dado por:

$$t_k = T\sum_{l=1}^{k} p_l \qquad k=0,\ldots,N \qquad (A2.4)$$

Ou seja, o tempo em que o sinal s(t) tem frequência instantânea k/T é $t_k - t_{k-1}$, e é proporcional à contribuição relativa da kª harmônica na DEP do sinal.

A frequência instantânea de s(t) no n<u>ésimo</u> intervalo de tempo é dada por:

$$B(t) = a_n + (2\pi n t)/T \qquad t_{n-1} < t < t_n \qquad (A2.5)$$

Para garantir a continuidade de B(t) em $t = t_{n-1}$, chega-se a:

$$a_n = a_1 - 2\pi\sum_{l=1}^{n-1} (n-1)p_l \qquad n=1,\ldots,N \qquad (A2.6)$$

que é a fórmula procurada para o cálculo das fases a_n; a_1 é arbitrária, e pode ser escolhida igual a zero por comodidade.

No caso de sinal com espectro plano tem-se $p_n = 1/N$, e a equação (A2.6) se reduz a:

$$a_n = a_1 - \pi n^2/N \qquad (A2.7)$$

Se os ângulos de fase são restritos apenas aos valores 0 ou π, e neste caso o sinal obtido no tempo é simétrico, a equação (A2.7) fica:

$$a_n = \pi[n^2/2N] \qquad (A2.8)$$

O sinal no tempo pode ser obtido diretamente neste último caso (espectro plano unitário e fases valendo unicamente 0 ou π), mediante a equação (A2.9).

$$s(t) = \sum_{n=1}^{N} S_n . \cos(nt/T) \qquad (A2.9)$$

S_n assume unicamente os valores 1 ou -1, e é calculado mediante a equação (A2.10).

$$S_n = 1 - 2[n^2/(2N)]_{mod2} \qquad (A2.10)$$

onde o operador mod2 significa "resto da divisão inteira por 2", da porção entre colchetes.

O aproveitamento das fórmulas de Schroeder foi inicialmente proposto por Burrows [32], com a finalidade de efetuar identificação de mancais. Posteriormente Oliveira [33] propôs o uso dos sinais assim obtidos em problemas de identificação de parâmetros de sistemas mecânicos em geral.

APÊNDICE 3

DESCRIÇÃO E LISTAGEM DOS PROGRAMAS

A3.1) Arquivo DADOS de definições.

Este arquivo contém as definições das constantes, variáveis e estruturas de variáveis a serem utilizadas no programa de simulação das condições de vibração, assim como também os "include" (ver ref. [25]) de funções de biblioteca que são requeridos pelo programa principal e os sub-programas.

O arquivo DADOS é carregado junto com o programa principal e os sub-programas durante a compilação, e por tanto o seu conteúdo so pode ser atualizado efetuando uma nova compilação. Neste sentido ele não se comporta como um arquivo de dados convencional, cujo conteúdo é lido sempre que o programa é executado.

```
/*  Arquivo com os dados requeridos pelo programa SIM_TLSI.                */

#include <math.h>              /* Carrega as funcoes matematicas.          */
#include <stdio.h>             /* Carrega as funcoes de entrada/saida.     */
#define T 10.0                 /* Tempo de cada bloco.                     */
#define N 4                    /* Numero de graus de liberdade do sist..   */
#define Ne 2                   /* Numero de pontos de excitacao.           */
/* #define Nr 4                   Numero de respostas, se != Ne.           */
#define NP 512                 /* Numero de pontos de cada vetor.          */
#define P2 9                   /* Potencia de 2: (NP=2**P2).               */
#define MF NP/10               /* Maxima frequencia de interesse.          */
#define PI 3.1415926535        /* Numero pi.                               */
#define Pc 1                   /* Porcentagem de erro admitido.            */
#define LB 4                   /* Largura da banda.                        */
#define MV LB*(2*N-LB+1)/2     /* Numero de elem. da matriz comprimida.    */

int COORD();

double MODULO();

struct Comp{
    double Re;
    double Im;
    } MUL_COMP(),DIV_COMP();

struct Vet{
    double V[NP];
    }SCHROEDER();

struct Col{
    double C[N];
    } MMAT_VET(),FNAO_LIN(),SOLVE();

struct Vet_Comp{
    struct Comp VC[NP];
    }TRF(),TRFI();

struct Col_Comp{
    struct Comp CC[Ne];
    }MMV_COMP();

struct Mat_Comp{
    struct Comp MC[Ne][Ne];
    }INV_MATC();

FILE *Ap,*fopen();
```

A3.2) Programa principal:

O programa principal, de nome SIM_TEST, mereceu uma descrição mais detalhada no capítulo III. Aqui é apresentada a listagem do mesmo.

```
/* Programa SIM_TEST, para achar as entradas de um sistema nao li-
   near que proporcionaram as saidas que se desele.                  */

#include "DADOS"

main()
{
     int i,j,k,Ruim,b;
     char st[60],st1[60];
     double Teste,TOL[Ne],Ms[MV],G[MV],K[MV];
     struct Col cO,cD;
     struct Col_Comp Aux;
     struct Vet SE[Ne],SO[N];
     struct Vet_Comp DS[Ne],S[Ne];
     struct Mat_Comp MT[MF+1];

     printf("Desenha graficos?\n Nao (0)\n Na tela (1)\n\
 Na tela e na impressora (2)\n");
     scanf("%d",&b);

/* Para ler as matrizes do sistema. */

     printf("\nNome do arquivo que contem as matrizes do sistema = ");
     gets(st);
     Ap=fopen(st,"r");
     for(j=0;j<MV;j++) fscanf(Ap,"%G%G%G\n",&Ms[j],&G[j],&K[j]);
     fclose(Ap);
     printf("\n Matrizes do sistema.\n");
     for(i=0;i<MV;i++) printf("M(%d)=%G   G(%d)=%G   K(%d)=%G\n",
          i,Ms[i],i,G[i],i,K[i]);

/* Para identificar o sistema. */

   /* Obs.: A dimensao da matriz de resposta em frequencia e igual
         ao numero Ne de entradas do sistema, que e tambem o numero
         de variaveis monitoradas.                                   */

/*   printf("\n\nIdentificando o sistema.\n\n");    */
     for(i=0;i<Ne;i++){
          for(j=0;j<Ne;j++){
               MT[0].MC[i][j].Re=0.0;
               MT[0].MC[i][j].Im=0.0;
          }
          for(j=0;j<NP;j++) SE[i].V[j]=0.0;
     }

    /* Para ler o sinal de Schroeder. */

     printf("Lendo o sinal de Schroeder.\n");
     i=sprintf(st,"S%3d-%2d",NP,MF);
     Ap=fopen(st,"r");
     for(i=0;i<NP;i++) fscanf(Ap,"%G",&SE[0].V[i]);
     fclose(Ap);
     GRAF(SE[0],NP,"Sinal de Schroeder."," ",0);
     for(i=0;i<NP;i++){
          DS[0].VC[i].Re=SE[0].V[i];
          DS[0].VC[i].Im=0.0;
     }
     DS[0]=TRF(DS[0],NP,P2);
```

```
    if( b){
        for(j=0,j<NP,j++) SO[0].V[j]=MODULO(DS[0].VC[j]),
        GRAF(SO[0],NP/2,"Sinal de Schroeder, em frequencia.","  ",0);
    }
    Ruim=1,
    for(i=0,i<Ne,i++){
        for(j=0,j<N;j++){
            uO.C[j]=0.0,
            vO.C[j]=0.0;
        }
        SIMULA(N,NP,LB,Ruim,0.25,0.5,T/NP,Ms,C,K,uO,vO,SE,SO),
        for(j=0;j<Ne;j++){
            for(k=0,k<NP;k++){
                S[0],VC[k].Re=SO[j].V[k],
                S[0],VC[k].Im=0.0,
            }
            S[0]=TRF(S[0],NP,P2),
            for(k=1;k<=MF;k++) /* Os limites atuam como filtro. */
                MT[k].MC[j][i]=DIV_COMP(&S[0].VC[k],&DS[0].VC[k]),
        }
        for(j=0,j<NP,j++){
            if(i < Ne-1) SE[i+1].V[j]=SE[i].V[j];
            SE[i].V[j]=0.0,
        }
        Ruim=0,
    }
    if( b ) for(i=0,i<Ne,i++)
        for(j=0,j<Ne,j++){
            for(k=0,k<NP/2,k++){
                if(k <= MF) SO[0].V[k]=MODULO(MT[k].MC[i][j]),
                else SO[0].V[k]=0.0;
            }
            k=sprintf(st1,"Matriz de resposta em frequencia.\
 Elemento %d,%d.",i+1,j+1);
            GRAF(SO[0],NP/2,st1,"  ",b),
    }

/* Para inverter a matriz de resposta em frequencia. */

    for(i=1;i<=MF,i++) MT[i]=INV_MATC(Ne,MT[i]);
    if( b) for(i=0;i<Ne;i++)
        for(j=0;j<Ne;j++){
            for(k=0;k<NP/2;k++){
                if(k <= MF) SO[0].V[k]=MODULO(MT[k].MC[i][j]);
                else SO[0].V[k]=0.0;
            }
            k=sprintf(st1,"Matriz inversa. Elemento %d,%d.",i+1,j+1);
            GRAF(SO[0],NP/2,st1,"  ",b);
    }

/* Para ler o sinal de campo. */

    printf("Nome do arquivo que contem os sinais desejados =\n")
    gets(st);
    Ap=fopen(st,"r");
    for(i=0,i<Ne,i++)
        for(j=0;j<NP,j++) fscanf(Ap,"%G",&S[i].VC[j].Re),
    fclose(Ap),
    for(i=0;i<Ne,i++){
        for(j=0,j<NP;j++){
```

```
            S[i].VC[j].Im=0.0;
            SO[0].V[j]=S[i].VC[j].Re;
        }
        j=sprintf(st1,"Sinal desejado. Variavel %d.",i+1);
        GRAF(SO[0],NP,st1," ",b);

    /* Para passar para frequencia. */

        S[i]=TRF(S[i],NP,P2);

    /* Para filtrar o sinal. */

        S[i].VC[0].Re=0.0;      /* Nao interessa o nivel DC. */
        S[i].VC[0].Im=0.0;
        for(j=MF+1;j<NP-MF;j++){
            S[i].VC[j].Re=0.0;
            S[i].VC[j].Im=0.0;
        }
        if( b ){
            for(j=0;j<NP/2;j++) SO[0].V[j]=MODULO(S[i].VC[j]);
            j=sprintf(st1,"Sinal desejado, em frequencia. Variavel %d.",
                i+1);
            GRAF(SO[0],NP/2,st1," ",b);
        }

    /* Para inicializar DS. */

        DS[i]=S[i];

    /* Para calcular a tolerancia no erro. */

        TOL[i]=0.0;
        for(j=1;j<=MF;j++) TOL[i]+=MODULO(S[i].VC[j]);
        TOL[i]=TOL[i]/MF*Pc/100;
    }
    fclose(Ap);

/* Processo iterativo para achar a excitacao correta. */

    k=0;
    printf("\n\nProcesso iterativo.\n");
    do{
        k+=1;
        i=sprintf(st1,"Aproximacao numero %d.",k);
        printf("\n%s\n",st1);
        if( b ) for(i=0;i<Ne;i++){
            for(j=0;j<NP/2;j++) SO[0].V[j]=MODULO(DS[i].VC[j]);
            j=sprintf(st,"Erro da resposta, em frequencia. Variavel %d."
                ,i+1);
            GRAF(SO[0],NP/2,st,st1,b);
        }
        for(j=1;j<=MF;j++){
            for(i=0;i<Ne;i++) Aux.CC[i]=DS[i].VC[j];
            Aux=MMV_COMP(Ne,Ne,MT[j],Aux);
            for(i=0;i<Ne;i++){
                DS[i].VC[j]=Aux.CC[i];
                DS[i].VC[NP-j].Re=DS[i].VC[j].Re;
                DS[i].VC[NP-j].Im= -DS[i].VC[j].Im;
            }
        }
```

```
        for(i=0;i<Ne,i++){
            if( b ) for(j=0,j<NP,j++) SO[0].V[j]=MODULO(DS[i].VC[j]),
            j=sprintf(st,"Correcao da excitacao, em frequencia.\
 Variavel %d.",i+1),
            GRAF(SO[0],NP/2,st,st1,b);
            DS[i]=TRF(DS[i],NP,P2);
            for(j=0;j<NP;j++) SE[i].V[j]+=DS[i].VC[j].Re;
            j=sprintf(st,"Sinal de excitacao. Variavel %d.",i+1);
            GRAF(SE[i],NP,st,st1,b),
        }
        SIMULA(N,NP,LB,0,0.25,0.5,T/NP,Ms,C,K,u0,v0,SE,SO);
        if( b ) for(i=0;i<N;i++){
            j=sprintf(st,"Sinal obtido. Variavel %d.",i+1),
            GRAF(SO[i],NP,st,st1,b),
        }

/* Para filtrar o sinal. */

        for(i=0;i<Ne;i++){
            for(j=0,j<NP,j++){
                DS[i].VC[j].Re=SO[i].V[j],
                DS[i].VC[j].Im=0.0,
            }
            DS[i]=TRF(DS[i],NP,P2),
            for(j=MF+1;j<NP-MF;j++){
                DS[i].VC[j].Re=0.0;
                DS[i].VC[j].Im=0.0,
            }
            if( b ) for(j=0;j<NP/2,j++){
                SO[i].V[j]=MODULO(DS[i].VC[j]);
                j=sprintf(st,"Sinal obtido, em frequencia.\
 Variavel %d.",i+1);
                GRAF(SO[i],NP/2,st,st1,b),
            }
        }

/* Calculo do erro nas respostas. */

        Ruim=0;
        for(i=0;i<Ne;i++){
            Teste=0.0;
            for(j=1;j<=MF;j++){
                DS[i].VC[j].Re=S[i].VC[j].Re-DS[i].VC[j].Re;
                DS[i].VC[j].Im=S[i].VC[j].Im-DS[i].VC[j].Im;
                Teste+=MODULO(DS[i].VC[j]),
            }
            Teste/=MF;
            printf("\n Var. %d Erro tolerado = %G",i+1,TOL[i]);
            printf("  Erro =  %G\n",Teste);
            if(Teste > TOL[i]) Ruim=1;
        }
    } while(Ruim),
    unlink("Sin_Com"),
    Ap=fopen("Sin_Com","a"),
    for(i=0;i<Ne;i++){
        for(j=0;j<NP,j++) fprintf(Ap," %G",SE[i].V[j]);
        j=sprintf(st,"Sinal de excitacao correto. Variavel %d.",i+1);
        GRAF(SE[i],NP,st," ",b);
    }
    fclose(Ap);
```

```
    printf("\n\nOs sinais de excitacao estao no arquivo Sin_Com.\n\n ");
    printf("                    ** FIM **\n");
}
```

A3.3) Inversão de matrizes.

No algoritmo proposto é utilizada a equação (I.11), onde aparece a IMRF, $[H(w)]^{-1}$, porém é impossivel identificar diretamente esta matriz se o sistema for de mais do que um grau de liberdade. Por isto é identificada a MRF e deve-se seguir um dos caminhos seguintes para calcular (F(w)): inverter a MRF e utilizar a equação (I.11) ou resolver o sistema de equações (I.5), onde (F(w)) é o vetor das incognitas e (X(w)) é o vetor dos elementos conhecidos.

O caminho mais usual é o segundo, mas neste caso particular foram feitas as seguintes considerações:

1º A matriz [H(w)] é de coeficientes complexos.

2º O sistema deve ser resolvido várias vezes durante a execução do programa, e uma vez invertida a matriz cada resolução significa apenas uma multiplicação de uma matriz por um vetor, o que representa economia de tempo. Deve ser lembrado que o que está sendo chamado vetor ou matriz são na verdade vetores de vetores e vetores de matrizes, por causa do processamento digital dos sinais, já que devem ser armazenados os valores correspondentes das variáveis para cada valor discreto que é assumido pela variável independente (w_i = i.dw i=1,..,n).

A primeira consideração não chega a ser um problema, sendo que existem diversos algoritmos para resolução de sistemas de equações que se adaptam a sistemas de números complexos. Inclusive a sub-rotina SOLVE utilizada no integrador, que se baseia no

método de Choleski clássico, e é portanto aplicável apenas a sistemas reais, poderia ser utilizada para sistemas complexos mediante um artifício, como será visto em A3.8, mas isso implica em duplicar a dimensão do problema.

Por isso optou-se por inverter de fato a matriz [H(w)].

Para tal fim foi implementada a sub-rotina INV_MATC que é baseada no método de redução por pivotamento máximo.

Para testar este algoritmo foram invertidas várias matrizes. Em seguida se efetuou o produto de cada matriz pela matriz inversa obtida. Obviamente o resultado deste produto deveria ser a matriz identidade, a menos dos erros numéricos cometidos no processamento. Seja a_{ij} um elemento genérico da matriz obtida da multiplicação de uma matriz pela inversa calculada utilizando-se esta sub-rotina. Para todos os casos testados verificou-se que:

$|a_{ij}| < 10^{-7}$ para todo i diferente de j, e

$1,0-10^{-7} < |a_{ii}| < 1,0+10^{-7}$ para os elementos da diagonal principal.

Portanto a precisão obtida na inversão das matrizes foi considerada satisfatória para a aplicação a que se destina.

```
/*  Funcao para inverter uma matriz m,complexa, de dimensao NxN.
    A chamada deve ter a forma: MI=INV_MATC(N,MA);
    MA e MI devem ter estructura Mat_Comp, definida em DADOS.    */

struct Mat_Comp INV_MATC(n,ma)

int n;
struct Mat_Comp ma;

{
    int i,j,k;
    float modulo,pivo;
    struct Comp c1,c2;
    struct Mat_Comp minv;

    for(i=0;i<n;i++){
        for(j=0;j<n;j++){
            minv.MC[i][j].Re=0.0;
            minv.MC[i][j].Im=0.0;
        }
        minv.MC[i][i].Re=1.0;
    }
    for(i=0;i<n;i++){
        k=i;
        pivo=0.0;
        for(j=i;j<n;j++){
            modulo=sqrt(ma.MC[j][i].Re*ma.MC[j][i].Re+
            ma.MC[j][i].Im*ma.MC[j][i].Im);
            if(modulo > pivo){
                pivo=modulo;
                k=j;
            }
        }
        if(k != i)
        for(j=0;j<n;j++){
            c1=ma.MC[i][j];
            ma.MC[i][j]=ma.MC[k][j];
            ma.MC[k][j]=c1;
            c1=minv.MC[i][j];
            minv.MC[i][j]=minv.MC[k][j];
            minv.MC[k][j]=c1;
        }
        c1=ma.MC[i][i];
        for(j=0;j<n;j++){
            ma.MC[i][j]=DIV_COMP(&ma.MC[i][j],&c1);
            minv.MC[i][j]=DIV_COMP(&minv.MC[i][j],&c1);
        }
        for(k=0;k<i;k++){
            c1=ma.MC[k][i];
            for(j=0;j<n;j++){
                c2=MUL_COMP(&c1,&ma.MC[i][j]);
                ma.MC[k][j].Re-=c2.Re;
                ma.MC[k][j].Im-=c2.Im;
                c2=MUL_COMP(&c1,&minv.MC[i][j]);
                minv.MC[k][j].Re-=c2.Re;
                minv.MC[k][j].Im-=c2.Im;
            }
        }
        for(k=i+1;k<n;k++){
```

```
            c1=ma.MC[k][i];
            for(j=0;j<n;j++){
                c2=MUL_COMP(&c1,&ma.MC[i][j]);
                ma.MC[k][j].Re-=c2.Re;
                ma.MC[k][j].Im-=c2.Im;
                c2=MUL_COMP(&c1,&minv.MC[i][j]);
                minv.MC[k][j].Re-=c2.Re;
                minv.MC[k][j].Im-=c2.Im;
            }
        }
    }
    return minv;
}
```

A3.4) Integrador.

Para obter a resposta do sistema foi implementado um integrador identificado com o nome SIMULA.

Após analisadas diversas alternativas, foi escolhido o método de integração de Newmark, que é um integrador implícito direto no domínio do tempo, por tratar-se de um integrador adequado para a obtenção da resposta de sistemas não lineares, e por ser incondicionalmente estável quando respeitadas certas condições.

O algoritmo utilizado para implementar este integrador é basicamente o apresentado na tabela 8.4 da página 323 da referência [26].

Para garantir a estabilidade foram utilizados os parâmetros alfa e delta respectivamente iguais a 0,25 e 0,50, tal como sugerido em [26], que também aconselha que se utilize um passo de integração dt igual a 1/10 do menor período contido no sinal de excitação. O menor período corresponde à maior frequência. Como a frequência de Nyquist é igual a 1/(2.dt) a condição é satisfeita se é garantido que a maior frequência contida no sinal é menor do que 1/5 da frequência de Nyquist, o que justifica a faixa de frequências escolhida para a geração do sinal de Schroeder e para o processamento dos sinais.

O programa SIMULA utiliza uma sub-rotina chamada INSIMULA que inicializa o integrador, calculando os parâmetros de integração e a matriz de rigidez efetiva, que por sua vez é triangularizada pela chamada da sub-rotina TRIANG para poder resolver o sis-

tema resultante mediante o método de Choleski utilizando a sub-rotina SOLVE descrita no item A3.8)

Os valores calculados em INSIMULA não são perdidos uma vez finalizado o processo de integração, de forma que o integrador só precisa ser inicializado na primeira vez em que é chamado, a menos que mude o sistema a ser simulado.

Para os casos de sistemas não lineares é utilizada uma sub-rotina que calcula as forças devidas às não linearidades do sistema. Estas forças são tratadas como se fossem forças externas ao sistema, e portanto adicionadas às excitações do mesmo, do lado direito da equação. Esta sub-rotina é chamada FNAO_LIN e assume aspectos diferentes para cada tipo de não linearidade simulado.

Para os casos em que as excitações externas são fornecidas na forma de deslocamentos impostos em alguns pontos e não na forma das forças que agem nos mesmos, é utilizada a sub-rotina DESL_FOR que calcula os esforços correspondentes aos deslocamentos dados.

Para os efeitos deste trabalho só interessa a resposta de regime do sistema, e portanto deve ser utilizado para interromper o integrador um critério que identifique se os sinais já estão estabilizados. Considera-se estabilizado o sinal se para todas as variáveis monitoradas a resposta obtida no último período T é semelhante à obtida no período anterior. Existem vários critérios de semelhança que poderíam ser utilizados. Talvez os mais exigentes e conclusivos sejam a comparação ponto a ponto ou a medida da coerência de um sinal entre o último período calculado e o anterior. Estes métodos têm como desvantagem o fato de ser necessá-

rio manter ativos na memória do computador dois blocos consecutivos de sinal, o que aumenta muito a solicitação de memória, já bastante crítica. Também não é viável guardar esses resultados temporariamente em arquivo devido ao tempo consideravelmente alto de acesso aos mesmos.

Um outro critério possível para verificar a estabilidade do sinal é o de se comparar o valor do primeiro ponto de um bloco com o do primeiro ponto do bloco seguinte da mesma variável. Já que o bloco deve conter um período do sinal esses valores devem ser iguais depois que o sinal está estabilizado. Este critério não foi adotado pois, principalmente para sinais muito ricos, existe o risco de esta coincidência de valores ser apenas casual.

O critério de comparação adotado foi o dos valores médios quadráticos dos sinais, ou seja, um sinal foi considerado estabilizado se o valor médio quadrático do período atual e do período anterior diferem em menos de 1%. Se isto acontecer para todas as variáveis as respostas são enviadas ao programa principal e o integrador é desativado.

Este critério de semelhança utilizado evidentemente não é ótimo, mas ele se mostrou satisfatório nos exemplos rodados, e deve-se lembrar que no caso de aplicação prática do método esta verificação não é necessária.

```
/*  Sub-rotina para obter a resposta de um sistema nao linear de va-
  rios graus de liberdade, sob a acao de forcas externas, ou de deslo-
  camentos impostos em alguns pontos.
    E utilizado o algoritmo de Newmark.
    As variaveis utilizadas sao:

    n - Ordem do sistema.
    np - Numero de pontos por bloco.
    lb - Largura da banda de armazenamento.
    par =0 -> Matriz de rigides efetiva ja foi montada.
        =1 ->   "     "      "       "     deve ser montada.
    al - Parametro alfa.(al)>=0.25*(0.5+del)**2. Geralmente al=0.25)
    del - Parametro delta.(del)>=0.5. Geralmente del=0.5)
    dt - Passo de integracao, em segundos.
    Ms - Matriz de massa, armazenada em vetor, em banda.
    C - Matriz de amortecimento  "     "    "      "    "   .
    K - Matriz de rigidez        "     "    "      "    "   .
    u0 - Posicao inicial.
    v0 - Velocidade inicial.
    R - Vetor dos sinais de forca externa, ou dos desloc. impostos.
    U - Vetor dos sinais de resposta obtidos.

  O numero de graus de liberdade do sistema, N, deve ser definido exter-
namente, assim como o numero de pontos dos vetores que contem os sinais,
NP.
*/

SIMULA(n,np,lb,par,al,del,dt,Ms,C,K,u0,v0,R,U)

int n,np,lb,par;
double al,del,dt,Ms[],C[],K[];
struct Vet R[],U[];
struct Col u0,v0;

{
    int i,j,it,l;
    double t;
    static double a[8],d[MV],mat[MV];
    struct Col ra,re,rd,v1,a0,a1,FNAO_LIN(),SOLVE(),DESL_FOR();

/*  printf("\nCalculando a resposta do sistema.\n");  */

 /* Inicializar os deslocamentos. */

    for(i=0;i<n;i++) U[i].V[0]=u0.C[i];

 /* Velocidades iniciais =v0. */

 /* Inicializar as aceleracoes. */

    if( par ) TRIANG(n,lb,Ms,mat);
    ra=MMAT_VET(n,lb,C,v0);
    re=MMAT_VET(n,lb,K,u0);
    rd=DESL_FOR(0,np,dt,R);
    for(i=0;i<n;i++) ra.C[i]=rd.C[i]-re.C[i]-ra.C[i];
    a0=SOLVE(n,lb,mat,ra);

 /* Inicializar o integrador se par=1. */
```

```
    a[6]=NEWMK_A(n,lb,dt,a,d1,a1,del,Ms,C,K);

 /* Processo iterativo. */

    it=0;
    for(i=0;i<n;i++) ra.C[i]=0.0;
    do{
        it+=1;
/*      printf("\nIteracao numero %d\n\n",it);  */
        for(i=0;i<np;i++){

 /* Calculo da carga efetiva em t+dt. */

            for(j=0;j<n;j++){
                a1.C[j]=a[0]*U[j].V[i]+a[2]*v0.C[j]+a[3]*a0.C[j];
                v1.C[j]=a[1]*U[j].V[i]+a[4]*v0.C[j]+a[5]*a0.C[j];
            }
            a1=MMAT_VET(n,lb,Ms,a1);
            v1=MMAT_VET(n,lb,C,v1);
            rd=FNAO_LIN(n,lb,u0,v0,C,K); /* Forcas nao lineares. */
            re=DESL_FOR(i,np,dt,R);  /* Forca corresp. ao desl. R. */
            for(j=0;j<n;j++) re.C[j]+=rd.C[j]+v1.C[j]+a1.C[j];

 /* Resolucao para o deslocamento em t+dt. */

            v1=SOLVE(n,lb,d1,re);
            u0=v1;
            if( i<np-1 ) for(j=0;j<n;j++) U[j].V[i+1]=v1.C[j];

 /* Calcular velocidades e aceleracoes em t+dt. */

            for(j=0;j<n;j++){
                a1.C[j]=v1.C[j]-U[j].V[i];
                a1.C[j]=a[0]*a1.C[j]-a[2]*v0.C[j]-a[3]*a0.C[j];
                v0.C[j]=v0.C[j]+a[6]*a0.C[j]+a[7]*a1.C[j];
                a0.C[j]=a1.C[j];
            }
        }
        for(i=0;i<n;i++) GRAF(U[i],np,"Resposta da iteracao."," ",1);

 /* Teste da convergencia. */

        i=0;
        for(i=0;i<n;i++){
            t=0.0;
            for(j=0;j<np;j++) t+=U[i].V[j]*U[i].V[j];
            t=sqrt(t)/np;
            printf("Variavel %d :\n Erro admitido =%G",i+1,1*t/100);
            printf("    Erro =%G\n",fabs(t-ra.C[i]));
            if(fabs(t-ra.C[i]) > 1*t/100) i=1;
            ra.C[i]=t;
        }

 /* Calculo do deslocamento inicial para a proxima iteracao. */

        if(i) for(i=0;i<n;i++) U[i].V[0]=v1.C[i];
    }
    while(i);
    for(i=0;i<n;i++) GRAF(U[i],np,"Resposta do sistema."," ",1);
}
```

A3.5) Sub-rotina INSIMULA para inicializar o integrador:

O sub-programa INSIMULA calcula os parâmetros e monta as matrizes requeridos pelo integrador SIMULA. As suas variáveis são declaradas estáticas, de forma que esta sub-rotina só precisa ser ativada para a primeira integração de uma estrutura ou quando a estrutura simulada sofre alguma alteração.

```
/* Sub-rutina auxiliar de SIMULA.
   Calcula os parametros que permanecem estaticos para o sistema.  */

INSIMULA(n,nb,dt,a,dl,al,del,Ms,C,K)

    int n,nb;
    double a[],dl[];
    double dt,al,del,Ms[],C[],K[];
{
    int i,j,ij;
    double ke[MV];

/*  Calculo das constantes. */

    a[2]=1.0/al/dt;
    a[1]=del*a[2];
    a[0]=a[2]/dt;
    a[3]=0.5/al-1.0;
    a[4]=del/al-1.0;
    a[5]=(del/al-2.0)*dt/2.0;
    a[6]=(1.0-del)*dt;
    a[7]=del*dt;

/* Calculo da matriz de rigides efectiva. */

    for(i=0;i<=n-nb;i++)
        for(j=i;j<i+nb;j++){
            ij=COORD(i,j,nb);
            ke[ij]=K[ij]+a[0]*Ms[ij]+a[1]*C[ij];
    }
    for(i=n-nb+1;i<n;i++)
        for(j=i;j<n;j++){
            ij=COORD(i,j,nb);
            ke[ij]=K[ij]+a[0]*Ms[ij]+a[1]*C[ij];
    }

/* Triangularizar a matriz de rigides efectiva. */

    TRIANG(n,nb,ke,dl);
}
```

A3.6) Sub-rotina FNAO_LIN para calcular o valor das forças não lineares.

Esta sub-rotina é um exemplo de função utilizada para calcular o valor das forças provenientes de não-linearidades simuladas no sistema. O resultado é fornecido ao integrador SIMULA para cada passo de integração.

```
/* Este e um exemplo da sub-rotina utilizada para calcular as forcas
   ocaclonadas pelas nao linearidades presentes num sistema.        */

struct Col FNAO_LIN(n,lb,d,v,C,K)

int n,lb;
double C[],K[];
struct Col d,v;

{
     int i;
     double a[Nr];
     struct Col f,f1,f2,MMAT_VET();

     for(i=0;i<n;i++) a[i]=100.0;
     for(i=0;i<n;i++){
          if( v.C[i] != 0 ) f.C[i]= -a[i]*v.C[i]/fabs(v.C[i]);
          else f.C[i]=0.0;
          f1.C[i]= -d.C[i]*fabs(d.C[i])/100.0;
          f2.C[i]= -v.C[i]*fabs(v.C[i])/1000;
     }
     f1=MMAT_VET(n,lb,K,f1);
     f2=MMAT_VET(n,lb,C,f2);
     for(i=0;i<n;i++) f.C[i]+=f1.C[i]+f2.C[i];
     return f;
}
```

A3.7) Sub-rotina DESL_FOR, que calcula as forças correspondentes a deslocamentos impostos.

Esta sub-rotina é um exemplo de algoritmo utilizado para determinar as forças que agem numa estrutura como consequência de deslocamentos impostos em determinados pontos da mesma. Ela deve auxiliar o integrador naqueles casos em que as perturbações externas são especificadas em termos de deslocamentos e não de forças.

```
/* Exemplo da sub-rotina utulizada para calcular as forcas correspon-
dentes ao deslocamento imposto a alguns pontos de um sistema.      */

struct Col DESL_FOR(i,np,dt,R)

int i,np;
float dt;
struct Vet R[];

{
    float DERIV();
    struct Col A;

    A.C[0]=0.0;
    A.C[1]=0.0;
    A.C[2]=DERIV(i,np,dt,R[0])*708.30+R[0].V[]*350164.00;
    A.C[3]=DERIV(i,np,dt,R[1])*737.20+R[1].V[]*379344.30;
    return A;
}
```

A3.8) Resolução de sistemas.

O integrador utilizado requer que sejam resolvidos sistemas de equações reais, simétricas, positivas definidas, de coeficientes constantes. Foi utilizado o método de Choleski, que se baseia na decomposição triangular da matriz do sistema [28].

Esquematicamente o método de Choleski é mostrado nas equações seguintes, onde (x) é o vetor das incognitas, (f) é o vetor dos termos conhecidos, (y) é um vetor auxiliar, [A] é uma matriz quadrada real, simétrica, positiva definida, e [L] uma matriz real triangular. O sistema a ser resolvido é:

$$[A].(x) = (f) \qquad (A3.1)$$

é sempre possível achar uma matriz [L] com as caraterísticas descritas tal que:

$$[L].[L]^t = [A] \qquad (A3.2)$$

Portanto pode-se calcular (x) mediante os dois passos a seguir:

$$[L].(y) = (f) \qquad (A3.3)$$

$$[L]^t.(x) = (y) \qquad (A3.4)$$

Os sistemas das equações (A3.3) e (A3.4) são facilmente resolvidos por substituição, uma vez que [L] é uma matriz triangular.

Este método é utilizado pelo sub-programa SOLVE, que por sua vez se auxilia com a sub-rotina LUP. A sub-rotina SOLVE recebe a matriz [L] previamente calculada pelo sub-programa TRIANG. De todas as matrizes só é guardada a banda que contém elementos diferentes de zero da metade superior, armazenando os coeficientes em vetor por colunas, aproveitando assim a simetria para economizar memória.

Este método pode ser extendido para sistemas de números complexos utilizando o seguinte artifício: um sistema

$$[A].(x) = (f) \qquad (A3.5)$$

de números complexos é equivalente ao sistema

$$\begin{bmatrix} Re\ A & -Im\ A \\ Im\ A & Re\ A \end{bmatrix} \cdot \begin{Bmatrix} Re(x) \\ Im(x) \end{Bmatrix} = \begin{Bmatrix} Re(f) \\ Im(f) \end{Bmatrix} \qquad (A3.6)$$

onde todos os coeficientes são reais. Porém, como pode ser observado a matriz do sistema resultante, que será chamada [B], não é simétrica. Para simetrizá-la deve-se pré-multiplicar dos dois lados da equação (A3.6) pela transposta da matriz resultante, ou seja que o sistema real equivalente a ser resolvido é:

$$[B]^t.[B].\begin{Bmatrix} Re(x) \\ \\ Im(x) \end{Bmatrix} = [B]^t.\begin{Bmatrix} Re(f) \\ \\ Im(f) \end{Bmatrix} \qquad (A3.7)$$

onde todos os coeficientes são reais e a matriz do sistema é simétrica.

Como pode ser observado, a dimensão do sistema é duplicada.

A continuação é listada a sub-rotina TRIANG que efetua a decomposição da equação (A3.2), e logo a seguir a sub-rotina SOLVE junto com a sua auxiliar LUP, que resolvem o sistema de acordo com as equações (A3.3) e (A3.4).

```
/* Sub-rutina para triangularizar uma matriz ke simetrica, positiva
definida, armazenada em vetor, utilizando o metodo de choleski.
   O resultado retorna no vetor dl.                                  */

TRIANG(n,nb,ke,dl)

int n,nb;
double ke[],dl[];

{
    int i,j,ij,jj,k,ini;

    for(i=0;i<n;i++){
        if(i<nb) ini=0;
        else ini=i-nb+1;
        for(j=ini;j<=i;j++){
            ij=COORD(i,j,nb);
            dl[ij]=0.0;
            for(k=ini;k<j;k++)
                dl[ij]+=dl[COORD(i,k,nb)]*dl[COORD(j,k,nb)];
            if(i==j) dl[ij]=sqrt(ke[ij]-dl[ij]);
            else dl[ij]=(ke[ij]-dl[ij])/dl[COORD(j,j,nb)];
        }
    }
}
```

```
/* Sub-rotina para resolver um sistema de equacoes lineares com a
 matriz ja na forma triangular, armazenada em vetor.
  A chamada deve ser da forma: x=SOLVE(n,nb,dl,f);
 onde n e a dimensao da matriz, nb e a largura da faixa na qual
 existem elementos diferentes de zero, dl e o vetor que contem os
 elementos da matriz triangular, e f e o vetor das variaveis indepen-
 dentes, e x e o vetor que recebe o valor calculado para as incogni-
 tas. x e f devem ter a estrutura Col definida em DADOS.          */

struct Col SOLVE(n,nb,dl,f)

int n,nb;
double dl[];
struct Col f;

{
     int i,j,i_f;
     double LUP();
     struct Col x,y;

     for(i=0;i<n;i++){
          if(i<nb) i_f=0;
          else i_f=i-nb+1;
          y.C[i]=LUP(i,y,i,nb,f,dl,i_f,i-1);
     }
     for(i=n-1,i>=0;i--){
          if(i+nb>=n) i_f=n-1;
          else i_f=i+nb-1;
          x.C[i]=LUP(i,x,i-n+1,nb,y,dl,i+1,i_f);
     }
     return x;
}

/* Funcao auxiliar de SOLVE. */

double LUP(i,x,j,nb,f,dl,in,fin)

int i,j,nb,in,fin;
double dl[];
struct Col x,f;

{
     int l;

     x.C[i]=0.0;
     if(j=0) x.C[i]=f.C[i]/dl[COORD(i,i,nb)];
     else{
          for(l=in;l<=fin;l++) x.C[i]+=x.C[l]*dl[COORD(l,i,nb)];
          x.C[i]=(f.C[i]-x.C[i])/dl[COORD(i,i,nb)];
     }
     return x.C[i];
}
```

A3.9) Gerador do sinal de Schroeder.

Para identificação das estruturas é utilizado o sinal de Schroeder, que como já foi dito é um sinal periódico de baixo fator de pico [23], [24].

Na versão implementada neste trabalho foi imposto que o espectro do sinal gerado seja plano com parte imaginária zero, e que as fases das componentes da parte real do sinal em frequência tenham unicamente os valores zero ou π.

Para gerar o sinal com essas caraterísticas foi implementada a sub-rotina SCHROEDER, que gera o sinal no domínio do tempo.

A amplitude das componentes em frequência é sempre um para uma faixa de frequências que é selecionada na sub-rotina. Fora desta faixa a amplitude é sempre zero.

Devido ao tempo necessário para gerar o sinal de Schroeder ser muito longo a geração não é feita durante a execução do programa SIM_TEST. Ao contrário, ela é feita antes e o resultado é armazenado em um arquivo que é lido na hora de efetuar a identificação do sistema.

```
/*  Sub-rutina para gerar o sinal de Schroeder com n pontos no tempo,
correspondente a um espectro plano unitario, com fases variando entre
0 e Pi, e faixa de frequencias extendida de df ate m*df.
  A resolucao em frequencia df depende do periodo de tempo T contido
 no vetor a ser gerado: df = 1/T = 1?(n*df).

  A chamada deve ser da forma:

                        S=SCHROEDER(n,m);

A variavel S deve ter a estrutura Vet definida no arquivo DADOS.     */

struct Vet SCHROEDER(n,m)

int n,m;

{
    struct Vet S;
    int i,f;
    long int j;

    printf("Gerando Schroeder. Tenha paciencia.\n Gerado ponto :");
    for(i=0;i<n;i++){
        S.V[i]=0.0;
        for(j=1;j<=m;j++){              /* Obs.: m < n/2   */
            f=1-2*((j*j/(2*m))%2);
            S.V[i]+=f*cos(2*Pi*j*i/n);
        }
        printf("%4d\b\b\b\b",i);
    }
    printf("\n");
    return S;
}
```

A3.10) Transformada rápida de Fourier direta e inversa.

É utilizado o algoritmo de Cooley e Tukey para a TRF e TRFI. A lógica para a implementação do mesmo foi extraída de [22].

A sub-rotina que passa os sinais do domínio do tempo para o domínio das frequências é chamada TRF, e a que realiza a operação inversa é chamada TRFI. Ambas utilizam as sub-rotinas auxiliares REORDENA e COOL_TUCK para efetuar a transformação. Apenas os parâmetros de chamada são diferentes para o caso direto e inverso.

A sub-rotina REORDENA faz o rearranjo dos elementos do vetor que contém o sinal a ser transformado, e logo a sub-rotina COOL_TUCK efetua a transformação sobre o vetor já reordenado.

Por não ser feita a correção necessária nas amplitudes dos sinais no domínio das frequências, estas aparecem divididas por dois.

Uma limitação do algoritmo de Cooley e Tukey é que ele somente efetua a transformação sobre vetores que contém um número NP de pontos igual a alguma potência inteira de dois, ou seja, para $NP = 2^n$, com n inteiro. Por este motivo o número NP de pontos de discretização utilizado para todos os vetores que contém os sinais respeita esta propriedade.

O algoritmo de Cooley e Tukey é longamente conhecido e dispensa portanto maiores apresentações. As fórmulas utilizadas no mesmo podem ser vistas no apêndice 1 ou em [22].

```
/* Sub-rotinas para efetuar Transformada Rapida de Fourier,
   direta e inversa, mediante o algoritmo de Cooley-Tuckey. */

/* Transformada direta.
    A chamada deve ser da forma:  B=TRF(A,NP,P);
   onde A e B devem ter estrutura Vet_Comp como a definida no ar-
   quivo DADOS, NP e o numero de pontos do vetor a ser transformado,
   e P e outro inteiro tal que NP = 2^P.                              */

struct Vet_Comp TRF(A,N_Pts,Pot_2)

int N_Pts,Pot_2;
struct Vet_Comp A;

{
    struct Vet_Comp REORDENA(),COOL_TUCK();

    A=REORDENA(A,N_Pts,-1);
    A=COOL_TUCK(A,N_Pts,Pot_2,-1);
    return A;
}

/*  Transformada inversa.
    A chamada deve ser da forma: B=TRFI(A,NP,P);
   onde todas as variaveis sao as mesmas descritas para a
   transformada direta.                                          */

struct Vet_Comp TRFI(A,N_Pts,Pot_2)

int N_Pts,Pot_2;
struct Vet_Comp A;

{
    struct Vet_Comp REORDENA(),COOL_TUCK();
    A=REORDENA(A,N_Pts,1);
    A=COOL_TUCK(A,N_Pts,Pot_2,1);
    return A;
}

/* Para reordenar os vetores. */

struct Vet_Comp REORDENA(A,N_Pts,P)

int  N_Pts,P;
struct Vet_Comp A;

{
    int i,j,k,npd2,npm1;
    struct Comp t;

    if(P==-1)for(i=0;i<N_Pts;i++){
        A.VC[i].Re/=N_Pts;
        A.VC[i].Im/=N_Pts;
    }
    npd2=N_Pts/2;
    npm1=N_Pts-1;
    j=0;
    for(j=0;j<npm1;j++){
        if(j<i){
```

```
                t=A.VC[i];
                A.VC[i]=A.VC[j];
                A.VC[j]=t;
            }
            k=npd2;
            while(k<=i){
                i-=k;
                k/=2;
            }
            i+=k;
        }
        return A;
}

/* Algoritmo "da borboleta". */

struct Vet_Comp COOL_TUCK(A,N_Pts,Pot_2,P)

int N_Pts,Pot_2,P;
struct Vet_Comp A;

{
        int i,j,k,l,m,me,ik;
        struct Comp u,t,w;

        for(m=1;m<=Pot_2;m++){
            u.Re=1.0;
            u.Im=0.0;
            me=1;
            for(i=1;i<=m;i++)me*=2;
            k=me/2;
            w.Re=cos(Pi/k);
            w.Im=P*sin(Pi/k);
            for(j=0;j<k;j++){
                for(l=j;l<N_Pts;l+=me){
                    ik=l+k;
                    t=MUL_COMP(&A.VC[ik],&u);
                    A.VC[ik].Re=A.VC[l].Re-t.Re;
                    A.VC[ik].Im=A.VC[l].Im-t.Im;
                    A.VC[l].Re+=t.Re;
                    A.VC[l].Im+=t.Im;
                }
            u=MUL_COMP(&u,&w);
            }
        }
        return A;
}
```

A3.11) Programa gráfico.

Para acompanhar o processamento dos sinais e interpretar e documentar os resultados obtidos é de fundamental importância a apresentação gráfica dos sinais. A soluçaõ deste problema não foi imediata por causa da falta de recursos gráficos do computador utilizado. Por este motivo foi necessário ligar um microcomputador do tipo PC funcionando como terminal do computador utilizado para fazer os programas, para obter as saídas gráficas através dele.

Para que o microcomputador se comporte como um terminal do computador maior é executado nele o programa ST240, que simula no computador PC um terminal de vídeo do tipo VT240.

Para a elaboração do programa de gráficos foram utilizados os comandos do ST240.

O programa implementado, identificado com o nome GRAF, foi direcionado para uso específico no programa SIM_TEST, e tem por isso algumas restrições e caraterísticas especiais. Algumas delas são:

- Somente desenha curvas com os valores de amplitude discretizados em vetores com número de pontos igual a alguma potência inteira de dois ($NP = 2^n$).
- Ajusta automaticamente as amplitudes para otimizar o aproveitamento da tela, ajustando ao mesmo tempo as escalas para que sejam lidos os valores corretos.

- Calcula a relação NP/Nº de pontos da tela ou Nº de pontos da tela/NP, dependendo de quem for o maior, para obter a melhor resolução possível.

- Se o vetor não contiver pontos com amplitudes negativas, desenha somente a metade superior (por exemplo quando se deseja desenhar a amplitude do sinal em frequência em coordenadas polares).

- Pode-se escolher o periférico de saída (tela ou impressora) mediante seleção dos parâmetros de chamada da sub-rotina.

Além disso o programa GRAF só pode ser utilizado através de um microcomputador que tenha condições de interpretar o programa ST240.

```
/* Sub-rotina para tracar o grafico de uma curva discretizada num
 vetor x, armazenado numa estructura tipo Vet ,contendo n pontos
igualmente espacados no eixo das abscisas. O numero de pontos n
deve ser uma potencia entera de 2. As variaveis st e st1 sao ca-
deias de caracteres, e servem para colocar titulo ao desenho. A
variavel a define se o desenho e feito no video (a=1) ou no papel
(a=2). Se a=0 nao faz o desenho.
  Esta sub-rotina so funciona em terminais do padrao vt240.     */

GRAF(x,n,st,st1,a)

int n,a;
char st[],st1[];
struct Vet x;

{
    int i,j,k,l,b=1;
    double max=0.0;

    if( a ){
        for(i=0;i<n;i++){
            if(max < fabs(x.V[i])) max=fabs(x.V[i]);
            if(x.V[i] < 0) b=0;
        }
        b*=140;
        if(max > 0.0){
            printf("\033H \033J");
            printf("\n\033< \033P1p");
            printf("S(A[0,0][799,479])");
            printf("S(E)");
            printf("P[610,%d]",220+b);
            if(n >= 512){
                k=n/512;
                l=1;
                j=511;
            }
            else{
                k=1;
                l=512/n;
                j=l*(n-1);
            }
        printf("V[-%d][,-%d][+%d][,+280][-%d][,-%d]",j,140+b,j,j,140-b);
            printf("P[,%d]",220+b-(int)(x.V[0]/max*135.0));
            for(i=k;i<n;i+=k)
          printf("V[+%d,%d]",l,220+b-(int)(x.V[i]/max*(135+b/14*13.5)));
            printf("P[615,%d]",220+b);
            printf("T'%d'",n-1);
            printf("P[%d,%d]",610-(j+15),210+b);
            printf("T'0'");
            printf("P[-90,70]");
            printf("T'%10.3E'",max*(140+b)/(135+b/14*13.5));
            if( !b ){
                printf("P[-90,350]");
                printf("T'%10.3E'",-max*14.0/13.5);
            }
            printf("P[90,20]");
            printf("T'%s'",st);
            printf("P[90,40]");
            printf("T'%s'",st1);
```

```
            if( a == 2 ) printf("S(H)");
            else getchar();
            printf("\033\\");
            if( a == 2 ) printf("\033[5i\n\n\n\n\033[4i");
            printf("\033c \n");
        }
        else printf("Nao desenho vetor nulo\n");
    }
}
```

A3.12) Tradutor de coordenadas.

A simetria das matrizes envolvidas é aproveitada aos efeitos de se economizar memória no armazenamento dos coeficientes das mesmas. Para tal fim, os elementos das matrizes são armazenados na forma de um vetor, somente os elementos da metade superior ou inferior. Para estabelecer a correspondência entre os elementos do vetor e os elementos da matriz correspondente é utilizada a sub-rotina COORD, que efetua a transformação das coordendas da matriz para as do vetor.

```
/* Funcao para calcular as coordenadas para guardar uma matriz em forma
   de vetor.
    A chamada deve ser da forma:
                                        ij=COORD(i,j,nb);
   onde ij e o inteiro que recebe a coordenada do vetor, i e j sao as
   coordenadas da matriz e nb e a largura da banda com elementos !=0.  */

int COORD(i,j,nb)

int i,j,nb.

{
    int ij;

    if(i-j >= nb || j-i >= nb)
        printf("As coordenadas (i,j) nao pertencem a banda\n");
    if(i<=j) if(j<nb) ij=j*(j+1)/2+i;
        else ij=nb*(nb+1)/2+(j+1)*(nb-1)-nb*nb+i;
    else if(i<nb) ij=i*(i+1)/2+j;
        else ij=nb*(nb+1)/2+(i+1)*(nb-1)-nb*nb+j;
    return ij;
}
```

A3.13) Funções auxiliares.

Para servir de apoio aos programas descritos antes foi necessário implementar algumas funções auxiliares, parte das quais são funções normais de biblioteca em algumas linguagens, mas que não estavam disponíveis na versão utilizada de C.

Segue uma breve descrição de algumas delas.

A3.13.1) Produto de números complexos.

A função MUL_COMP calcula o produto de dois números complexos, que devem estar armazenados em variáveis que tenham uma estrutura complexa como a definida no programa descrito.

A3.13.2) Divisão de números complexos.

Idem à anterior, só que para a divisão entre números complexos, a função DIV_COMP imprime uma mensagem de advertência se o módulo do denominador for zero, mas não detém o programa para facilitar a detecção de erros de lógica.

A3.13.3) Módulo de números complexos.

A função MODULO calcula o módulo de um número complexo, definido como a raiz quadrada da soma dos quadrados das partes real e imaginária do complexo, ou seja

$$|C| = ((\mathrm{Re}\ C)^2 + (\mathrm{Im}\ C)^2)^{1/2} \qquad (A3.8)$$

A3.13.4) Derivada de um vetor.

A função DERIV calcula a derivada numérica de uma variável discretizada e armazenada num vetor. O método utilizado é o das diferenças centrais, que para intervalos de discretização constantes é equivalente a aproximar a derivada num ponto pela corda extendida do ponto imediatamente anterior ao imediatamente seguinte, ou seja:

$$dx_i/dt = ((x_i - x_{i-1})/dt + (x_{i+1} - x_i)/dt)/2 = (x_{i+1} - x_{i-1})/2dt \qquad (A3.9)$$

A3.13.5) Multiplicação de matriz simétrica por vetor.

A função MMAT_VET serve para efetuar o produto de uma matriz quadrada simétrica de coeficientes reais armazenados na forma de vetor, somente a banda superior com elementos diferentes de zero, por um vetor de elementos também reais.

A3.13.6) Multiplicação de matriz por vetor, ambos complexos.

A função MMV_COMP calcula o produto de uma matriz de elementos complexos e dimensões mxn quaisquer por um vetor de elementos complexos e dimensão n.

```
/* Funcao para multiplicar dois numeros complexos.
   O resultado retorna na variavel complexa P.
   A chamada deve ter a forma:

      P=MUL_COMP(&A,&B);                                    */

struct Comp MUL_COMP(m,n)

struct Comp *m,*n;

{
    struct Comp P;

    P.Re=m->Re*n->Re-m->Im*n->Im;
    P.Im=m->Re*n->Im+n->Re*m->Im;
    return P;
}

/* Funcao para dividir numeros complexos.
   O resultado retorna na variavel complexa Q.
   A chamada deve ter o formato:

      Q=DIV_COMP(&A,&B);                                    */

struct Comp DIV_COMP(n,d)

struct Comp *n,*d;
{
    struct Comp Q;
    double fator;

    fator=d->Re*d->Re+d->Im*d->Im;
    if(fator)
    {
        Q.Re=(n->Re*d->Re+n->Im*d->Im)/fator;
        Q.Im=(n->Im*d->Re-n->Re*d->Im)/fator;
    }
    else
    {
        Q.Re=0.0;
        Q.Im=0.0;
        printf("** Atencao: divissao por zero = 0.0 **\n");
    }
    return Q;
}

/* Funcao para calcular o modulo de um numero complexo.
   A chamada deve ser:
            b=MODULO(a);
    onde b e uma variavel real e a uma estructura tipo Comp.   */

double MODULO(a)

struct Comp a;

{
    double b;

    b=sqrt(a.Re*a.Re+a.Im*a.Im);
```

```
    return b;
}

/* Sub-rotina para calcular a derivada de um vetor.  */

float DERIV(i,np,dt,R)

int i,np;
float dt;
struct Vet R;

{
    float D;

    if( i=0 ) D=(R.V[1]-R.V[np-1])/(2.0*dt);
    else{
        if( i=np-1 ) D=(R.V[0]-R.V[np-2])/(2.0*dt);
        else D=(R.V[i+1]-R.V[i-1])/(2.0*dt);
    }
    return D;
}

/* Sub-rotina para multiplicar uma matriz real quadrada, armazenada
 em banda, de dimensoes nxn, por um vetor. */

struct Col MMAT_VET(n,lb,A,B)

int n,lb;
double A[MV];
struct Col B;

{
    int i,j,ini,fin;
    struct Col C;

    for(i=0;i<n;i++){
        C.C[i]=0.0;
        if(i < lb) ini=0;
        else ini=i-lb+1;
        if(i+lb <= n)fin=i+lb;
        else fin=n;
        for(j=ini;j<fin;j++) C.C[i]+=A[COORD(i,j,lb)]*B.C[j];
    }
    return C;
}

/* Sub-rotina para multiplicar 1 matriz A complexa por 1
  vetor B complexo, armazenando o resultado em outro ve-
  tor analogo C.
   A chamada deve ser da forma "C=MMV_COMP(n,m,A,B)," onde
  A tem uma estrutura tipo Mat_Comp, e B e C tem estrutu-
  ra tipo Col_Comp, definidas externamente. As dimensoes
  da matriz A sao n x m.                                  */

struct Col_Comp MMV_COMP(n,m,A,B)

int n,m;

struct Mat_Comp A;
struct Col_Comp B;  /* Verificar a definicao em cada caso. */
```

```
{
    int i,j;
    struct Col_Comp C;

    for(i=0;i<n;i++){
        C.CC[i].Re=0.0;
        C.CC[i].Im=0.0;
        for(j=0;j<m;j++){
            A.MC[i][j]=MUL_COMP(&A.MC[i][j],&B.CC[j]);
            C.CC[i].Re+=A.MC[i][j].Re;
            C.CC[i].Im+=A.MC[i][j].Im;
        }
    }
    return C;
}
```

A3.14) Programa de correção das excitações implementado no Analisador de Fourier.

Para verificar o bom funcionamento do programa num protótipo experimental, foi implementada uma versão simplificada do mesmo (restrita a apenas um grau de liberdade) num Analisador de Fourier Hewllett Packard modelo 5451C. Este programa foi codificado utilizando a linguagem de programação própria do citado equipamento.

Para identificar a estrutura é utilizado o sinal de Schroeder, a exemplo do que acontecia nas simulações, e portanto foi necessário montar um programa para gerar este sinal também no Analisador de Fourier.

A listagem destes dois programas é apresentada a seguir.

No final está a listagem das mensagens utilizadas nestes dois programas.

```
/*
     PROGRAMA PARA VERIFICAÇÃO DO ALGORITMO DE CORREÇÃO DAS EXCI-
TAÇÕES NO PROTÓTIPO MONTADO, UTILIZANDO O ANALISADOR DE FOURIER.
*/

     L
     Y     5838     8
     Y     W     1     1
     J     0     4     1
     :     7     2
     F     7
     CL    7     0     10
     F     7
     Y     100   0     -2
     Y     5838  8
     B     7
     L     2
     RA
     H     2     2     0
     B
     X(    7
     F     0     1
     :     1
     Y     BS    10
     Y     A-    13    10D
     Y     :     11    10D   2
     Y     :     12    10D   4
     CL    0     51    11D
     X)    6
     Y     W     2     1
     Y     W     3     1
     D
     RA
     X(    1
     F
     CL    0     0     10
     CL    0     51    11D
     F
     X)    8
     Y     W     4     1
     D
     *
     $
     :     0     100
     Y     X(    2001     0     13D
     Y     W     5     1
     Y     P     2001
     X(    8
     F
     *     6
     F
     X)    7
     L     3
     B     7
```

```
L     4
RA
#     4     2     0
B
F     1
CL    1     0     10
CL    1     51    11D
F     1
X(    8
A-    1
X)    2
*
$
Y     X(    2000      0     13D
Y     W     6     1
Y     P     2000
Y     IF    2000      2001D     8     -1
X(    2
F
*     6
F
A+    7
X)    7
#     3     20    0
Y     W     7     1
D     7
```

```
/*
     PROGRAMA  PARA GERAR O SINAL DE SCHROEDER COM FAIXA DE  FRE-
QUÊNCIAS DE 0 A Fmax/2, ESPECTRO PLANO UNITÁRIO E FASES VALENDO 0
OU Pi, NO ANALISADOR DE FOURIER.
*/

     L    0
     Y    5838     7
     Y    W     1     1
     CL   7
     F    7
     Y    BS    5
     Y    :     10    5D    4
     Y    _     1     1
     Y    A-    11    10D   1
     L    1
     Y    *     2002      1D      1D
     Y    :     2000      2002D   5D
     L    2
     Y    IF    2000      2       2     -2
     Y    A-    2000      2002D   2
     J    2     -1
     Y    IF    2000      1       2     1
     Y    _     12     1
     J    3     -1
     Y    _     12    -1
     L    3
     Y    X)    12    7     1D
     Y    A+    1     1D    1
     #    1     11D   0
     F    7
     Y    X(    2001   7    0
     :    7     2001D
     (
```

```
/*
     MENSAGENS DO PROGRAMA DE CORREÇÃO DAS EXCITAÇÕES.
*/

01
PROGRAMA  PARA VERIFICAÇÃO EXPERIMENTAL DO ALGORITMO DE  CORREÇÃO
DAS EXCITAÇÕES.

02
A MATRIZ DE RESPOSTA EM FREQUÊNCIA ESTÁ NO BLOCO 6.

03
ENTRAR COM O SINAL DESEJADO.

04
O  SINAL DESEJADO ESTÁ NO BLOCO 8. APERTE "CONTINUE" PARA INICIAR
O PROCESSO ITERATIVO.

05
ERRO ADMITIDO =

06
ERRO NESTA ITERAÇÃO =

07
NÃO CONVERGIU EM 20 ITERAÇÕES.



/*
     MENSAGEM DO PROGRAMA QUE GERA O SINAL DE SCHROEDER.
*/

01
PROGRAMA PARA GERAR O SINAL DE SCHROEDER.
ESPECTRO PLANO UNITÁRIO.
FAIXA DE FREQUÊNCIAS: 0 A Fmax/2.
FASES VALENDO 0 OU Pi.
```

REFERÊNCIAS

[1] Landgraf, R. W.
Durability by Design - An Overview.
S. A. E. paper 871937.

[2] Nolan, S. A. and Linden, N. A.
Integrating Simulation Technology into Automotive Design Evaluation and Validation Processes.
S. A. E. paper 871941.

[3] Devlukia, J. and Davies, J.
Experiment and Analytical Techniques for Assessing the Durability of Automotive Structures

[4] Petersen, J. and Weisberger, G.
Strain simulation on the complete vehicle... The AUDI road simulator.
MTS paper.

[5] Cryer, B. W., Nawrocky, P. E. and Lund, R. A.
A Road Simulation System for Heavy Duty Vehicles.
S.A.E. paper 760361.

[6] The Automated Fatigue Testing Laboratory...
Publicação MTS 300002-26/701.00-02-386, 1986.

[7] Remote Parameter Control.
Publicação MTS 300002-30/700.02-01-587, 1986.

[8] MTS Idea Book for Vehicle Testing.
Publicação MTS 300002-07/116.01-06-686.

[9] Test Site Controller.
Publicação MTS 300002-25/116.12-02-484.

[10] New Remote Parameter Control III.
Publicaçãso MTS 500000-32/700.05-01-288, 1988.

[11] The Hidropuls System.
Publicação Schenck P 2062e.

[12] Computer Based Testing.
Publicação Schenck P 2062e.

[13] Werkstoffprufung, Bauteilprufung und Betriebslastensimulation mit Schenck-Synthese aus Kreativitat und Erfahrung.
Publicação Schenck P 2100.

[14] What Does ITFC (Iterative Transfer Function Compensation) Offer the Test Engeneer?
Publicação Schenck P50e.

[15] Fatigue Life Test of a VW caravelle Using a Hidropuls Road Simulator.
Publicação Schenck PTV-VW H.M. 86e.

[16] Craig, J. B.
Multiple Excitation Fatigue Testing of a Vehicle Axle With ITFC.
Paper presented at the SEE-conference "Digital Techniques in Fatigue".
London, 28 to 30th of March 1983.

[17] Craig, J. B.
ITFC - How it Works and Where To Use It
Publicação Scheck.

[18] Eykhoff, P.
System Identification.
John Wiley & Sons Ltd., 1974.

[19] Styles, D. J. and Dodds, C. J.
Simulation of Random Environments for Structural Dynamics Testing.
Experimental Mechanics, November 1976, pp. 416-424.

[20] Klinger, F.
Digitale Regelung von Mehrkomponenten-Betriebsfestingkeits-prufstanden fur die Automobilindustrie.
ATZ Automobiltechnische Zeitschrift 82(1980)9, pp. 469-474.

[21] Thomson, W. T.
Teoria da Vibração.
Editora Interciencia, 1978.

[22] Newland, D. E.
Random Vibrations and Spectral Analysis, second edition.
Longman Group Limited, 1984.

[23] Arruda, J. R. F.
Nota técnica sobre a obtenção da Função de Resposta em Frequência, Departamento de Projeto Mecânico, UNICAMP, 1987.

[24] Schroeder, M. R.
Synthesis of Low-Peak-Factor Signals and Binary Sequences With Low Autocorrelation.
IEEE Transactions on Information Theory, january 1970, pp 85-89.

[25] Kernigan, B. W. e Ritchie, D. M.
A linguagem de programação C.
Editora Campus Ltda., 1986.

[26] Bathe, K. J. and Wilson, E. L.
Numerical Methods in Finite Element Analysis.
Prentice-Hall Inc., 1976.

[27] Ewins, D. J.
Modal Testing: Theory and Pratice.
John Wiley & Sons Inc., 1985.

[28] Westlake, J. R.
A Handbook of Numerical Matrix Inversion and Solution of Linear Equations.
John Wiley & Sons Inc., 1968.

[29] Bendat, J. S. and Piersol, A. G.
Engineering Applications of Correlation and Spectral Analysis.
John Wiley & Sons, 1980.

[30] Arruda, J. R. F. e Godoy, E.
A Peak Classification Technique in Digital Spectral Analysis.
Proceedings of the 7th IMAC, Las Vegas, USA, 1989.
pp. 1582-1586.

[31] Arruda, J. R. F.
Introdução à análise de sinais.
Apostila do curso de vibrações aleatórias, UNICAMP, 1984.

[32] Burrows, C. R.
An Appraisal of Schroeder-Phased Harmonic Signals for Bearing Identification.
ASME paper 80-WA/DSC-34.

[33] Oliveira, N. V.
Identificação de parâmetros de sistemas mecânicos com aplicação a mancais.
Tese de Doutoramento, UNICAMP, 1988.

[34] Beck, J. V.
Parameter Estimation in Engeneering and Science.
John Wiley & Sons, 1976.

[35] Bendat, J. S. and Piersol, A. G.
Analysis and Measurement Procedures
Second edition.
John Wiley & Sons, 1986.

BIBLIOGRAFÍA

Spangler, E. B. and Kelly, W. J.
GMR Road Profiler - A Method for Measuring Road Profiles.
Highway Research Record Nº 121, pp 27-54, 1965.

Borgman, L. E.
Ocean Wave Simulation for Engineering Design.
Journal of Waterways and Harbours Div. Proc. Am. Soc. Civil Eng. WW4, pp 557-582, 1969.

Camp, J. D.
Random Load Fatigue Test on Automotive Components and Structures.
ASTM Special Tech. Publication 476 (Advanced Testing Techniques) pp 46-58, 1970.

Whittemore, A. P.
A Technique for Measuring "Efective" Road Profiles.
SAE paper 720094.

Cripe, R. A.
Making a Road Simulator Simulate.
SAE paper 720095.

Dodds, C. J. and Robson, J. D.
The Description of Road Surface Roughness.
Journal of Sound and Vibration, 1973, v. 31(2), pp 175-183.

Dodds, C. J.
The Laboratory Simulation of Vehicle Service Stress.
ASME paper 73-DET-24 pp 391-398, 1974.

Zell, K
Overview of Advanced Road Simulation Techniques
SAE paper 740946.

Norman, J. and Craig, J.
Service Load Simulation Testing of Complex Structures.
Chart. Mech. Eng. Vol. 27 Nº 3, 3/80, pp 47-48.

Petersen, J and Weissberger, G
The Conception, Description and Aplication of a New Vehicle Endurance Test Sistem at AUDI NSU.
SAE paper 820942.

Dutkievicz, I and Nates, R.
Design and Construction of a Rolling-Road Dynamometer for Road Simulation.
South Africa Mechanical Engineering, Vol. 35 Nº 3, pp76-79, 1985.

Shinkle, G. A.
Acelerated Vibration Durability Testing: A Practical Approach.
Automotive Engineering Vol. 93 06/1985, pp 58-64.

Harrison, R. F. and Hammond, J. K.
Analysis of Nonstationary Response of Vehhicles With Multiple wheels.
Journal of Dynamics Systems, Meassurement and Control, Vol. 108/69, 03/1986.

Tustin, W.
The Future of Random Vibration Screening and Testing in Automotive Engineering.
S. A. E. paper 870984.

Santika, P., Adnyana, D. N. and Partowiyatmo, A.
Laboratory Testing Facilities to support the Development of the Automotive Industry.
S.A.E. paper 871251.