

UNIVERSIDADE FEDERAL DE ITAJUBÁ
INSTITUTO DE ENGENHARIA MECÂNICA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA MECÂNICA

DISSERTAÇÃO DE MESTRADO

**Aplicação de filtro de Kalman para um
problema inverso de localização e detecção de
dano em problema 2-D**

Autor: **Bruno André Duarte Braz Vieira**

Orientador: **Prof. Dr. Ariosto Bretanha Jorge**

Co-Orientador: **Prof. Dr. Sebastião Simões da Cunha Júnior**

Itajubá, Outubro de 2011

UNIVERSIDADE FEDERAL DE ITAJUBÁ
INSTITUTO DE ENGENHARIA MECÂNICA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA MECÂNICA

DISSERTAÇÃO DE MESTRADO

**Aplicação de filtro de Kalman para um
problema inverso de localização e detecção de
dano em problema 2-D**

Autor: Bruno André Duarte Braz Vieira

Orientador: Prof. Dr. Ariosto Bretanha Jorge

Co-Orientador: Prof. Dr. Sebastião Simões da Cunha Júnior

Curso: Mestrado em Engenharia Mecânica

Área de Concentração: Projeto e Fabricação

Dissertação submetida ao Programa de Pós-Graduação em Engenharia Mecânica como parte dos requisitos para obtenção do Título de Mestre em Engenharia Mecânica.

Itajubá, Outubro de 2011

M.G. – Brasil

Ficha catalográfica elaborada pela Biblioteca Mauá –
Bibliotecária Cristiane Carpinteiro- CRB_6/1702

V658a

Vieira, Bruno André Duarte Braz

Aplicação de filtro de Kalman para um problema inverso de localização e detecção de dano em problema 2 – D. / por Bruno André Duarte Braz Vieira. -- Itajubá (MG) : [s.n.], 2011.

115 p. : il.

Orientador : Prof. Dr. Ariosto Bretanha Jorge.

Coorientador : Prof. Dr. Sebastião Simões da Cunha Júnior.

Dissertação (Mestrado) – Universidade Federal de Itajubá.

1. Filtro de Kalman. 2. Detecção de danos. 3. Problema inverso. I. Jorge, Ariosto Bretanha, orient. II. Cunha Júnior, Sebastião Simões da, coorient. III. Universidade Federal de Itajubá. IV. Título.

UNIVERSIDADE FEDERAL DE ITAJUBÁ
INSTITUTO DE ENGENHARIA MECÂNICA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA MECÂNICA

DISSERTAÇÃO DE MESTRADO

**Aplicação de filtro de Kalman para um
problema inverso de localização e detecção de
dano em problema 2-D**

Autor: **Bruno André Duarte Braz Vieira**

Orientador: **Prof. Dr. Ariosto Bretanha Jorge**

Co-Orientador: **Prof. Dr. Sebastião Simões da Cunha Júnior**

Composição da Banca Examinadora:

Prof. Dr. Rogério José Marczak – DEMEC/UFRGS

Prof. Dr. André Garcia Chiarello – IEM/UNIFEI

Prof^a. Dr^a. Patricia da Silva Lopes – IEM/UNIFEI

Prof. Dr. Ariosto Bretanha Jorge, Orientador – IEM/UNIFEI

Prof. Dr. Sebastião Simões da Cunha Júnior, Co-orientador – IEM/UNIFEI

Dedicatória

*À minha esposa Ivana,
aos meus filhos
João Pedro e Luiza.*

Agradecimentos

Ao meu Orientador, Prof. Dr. Ariosto Bretanha Jorge, pela competência, paciência e amizade.

Aos amigos, do GEMEC – Grupo de Pesquisa em Mecânica Computacional e da pós-graduação pelos momentos de descontração e ajuda durante o programa de pós-graduação.

Ao Instituto de Engenharia Mecânica da UNIFEI, representado pelos seus dedicados Professores e Funcionários, pela oportunidade que me concedeu na realização deste trabalho, e aos amigos desse Instituto, pelo convívio profissional.

À FAPEMIG e CAPES, através do Programa de bolsas, pelo apoio financeiro.

*"Nenhum trabalho de qualidade pode ser feito sem concentração e auto-sacrifício,
esforço e dúvida." (Max Beerbohm)*

Resumo

VIEIRA, B. A. D. B. (2011), *Aplicação de filtro de Kalman para um problema inverso de localização e detecção de dano em problema 2-D*, Itajubá, 117 p. Dissertação (Mestrado em Projeto e Fabricação) – Instituto de Engenharia Mecânica, Universidade Federal de Itajubá.

Neste trabalho, um modelo de filtro de Kalman para um problema inverso de localização e identificação de parâmetros de dano em uma estrutura de 2-D é implementado. Alguns aspectos da aplicação da metodologia do filtro de Kalman para problemas inversos são discutidos, incluindo o modelo do filtro e seus parâmetros. O método inverso requer um modelo para o problema direto para obter a distribuição da quantidade de interesse em pontos de medição no domínio em função das variáveis de estado. Esta relação é escrita como um problema do valor limite, que pode ser modelada utilizando um método numérico apropriado. Neste trabalho, o método dos elementos de contorno é usado para este modelo de problema direto. Dois modelos para o problema direto são aplicados, o potencial e o elastostático. A formulação matemática e o procedimento para a obtenção do modelo numérico são descritos. Os resultados numéricos são apresentados para comparação entre a abordagem do filtro de Kalman e outras técnicas de identificação de parâmetros, tais como otimização e Redes Neurais Artificiais.

Palavras-chave

Filtro de Kalman, Detecção de Danos, Problema Inverso.

Abstract

VIEIRA, B. A. D. B. (2011), *A Kalman filter application for an inverse problem of localization and identification of damage on a 2-D problem*, Itajubá, 117 p. MSc. Dissertation – Mechanical Engineer Institute , Federal University of Itajubá.

In this work, a Kalman filter model for an inverse problem of localization and identification of damage parameters on a 2-D structure is implemented. Some aspects of the application of the Kalman filter methodology for inverse problems are discussed, including the filter model and its parameters. The inverse method requires a model for the direct problem of obtaining the distribution of the quantity of interest in the measurement points in the domain as a function of the state variables. This relationship is written as a boundary value problem, which can be modeled using an appropriate numerical method. In this work, the boundary element method is used for this direct problem model. Two models for the direct problem are implemented, for the potential and elastostatic problems. The mathematical formulation and the procedure for obtaining the numerical model are described. Numerical results are presented for comparison between the Kalman filter approach and other techniques for parameter identification, such as optimization and artificial neural networks.

Keywords

Kalman Filter, Damage Detection, Inverse Problem.

Sumário

SUMÁRIO.....	I
LISTA DE FIGURAS	III
SIMBOLOGIA.....	VII
LETRAS LATINAS.....	VII
LETRAS GREGAS	VIII
SOBRESCRITOS.....	VIII
SUBSCRITOS.....	VIII
SIGLAS E ABREVIATURAS.....	IX
CAPÍTULO 1	
INTRODUÇÃO.....	1
1.1 REVISÃO DA LITERATURA.....	3
1.2 OBJETIVO.....	4
1.3 HIPÓTESE.....	4
1.4 RELEVÂNCIA	5
CAPÍTULO 2	
MODELO DIRETO: MÉTODO DE ELEMENTOS DE CONTORNO.....	6
CAPÍTULO 3	
PROBLEMA INVERSO: FILTRO DE KALMAN.....	10
CAPÍTULO 4	
DESENVOLVIMENTO.....	20
CAPÍTULO 5	
ANÁLISE DOS RESULTADOS	28

CAPÍTULO 6**CONCLUSÃO..... 60****REFERÊNCIAS BIBLIOGRÁFICAS 62****APÊNDICE A****REVISÃO 65****APÊNDICE B****DISTRIBUIÇÃO T-STUDENT..... 71****APÊNDICE C****CÓDIGOS MODELO POTENCIAL..... 73****APÊNDICE D****CÓDIGOS MODELO ELASTOSTÁTICO 84**

Lista de Figuras

Figura 2.1 – Direções das informações dos nós dos contornos.....	8
Figura 3.1 – Ciclo do processo do Filtro de Kalman.....	14
Figura 3.2 – Algoritmo do filtro de Kalman discreto.....	15
Figura 3.3 – Algoritmo do filtro de Kalman estendido.....	19
Figura 4.1 – Modelo da placa para o modelo potencial. [Adaptado de Lopes (2010)].....	21
Figura 4.2 – Modelo potencial: (a) discretização da placa e posicionamento dos sensores; (b) discretização do furo.....	21
Figura 4.3 – Modelo da placa para o modelo elastostático.[Adaptado de Lopes (2010)].....	22
Figura 4.4 – Modelo elastostático: (a) discretização da placa e posicionamento dos sensores; (b) discretização do furo.....	23
Figura 4.5 – Posições relativas ao furo utilizadas para o calculo da matriz de sensibilidade. .	25
Figura 4.6 – Algoritmo do filtro de Kalman utilizado.....	26
Figura 5.1 – Região limite de identificação do filtro para sensores à 0,6 cm das bordas de uma placa de 6x6 cm.....	29
Figura 5.2 – Região limite de identificação do filtro para sensores à 0,06 cm das bordas de uma placa de 6x6 cm.....	30
Figura 5.3 – Estimação das componentes do estado pelo filtro de Kalman com $Q = 14Q_0$...	32
Figura 5.4 – Progresso da estimção do furo pelo filtro de Kalman com $Q = 14Q_0$	32
Figura 5.5 – Ampliação da estimção final do furo pelo filtro de Kalman com $Q = 14Q_0$	33
Figura 5.6 – Estimção das componentes do estado pelo filtro de Kalman com $Q = Q_0$	33
Figura 5.7 – Progresso da estimção do furo pelo filtro de Kalman com $Q = Q_0$	34
Figura 5.8 – Ampliação da estimção final do furo pelo filtro de Kalman com $Q = Q_0$	34
Figura 5.9 – Estimção das componentes do estado pelo filtro de Kalman com $Q = 4Q_0$	35
Figura 5.10 – Progresso da estimção do furo pelo filtro de Kalman com $Q = 4Q_0$	36
Figura 5.11 – Ampliação da estimção final do furo pelo filtro de Kalman com $Q = 4Q_0$	36
Figura 5.12 – Estimção das componentes do estado pelo filtro de Kalman com $Q = Q_0$ e $x = \{5,25; 4,40; 0,06\}^T$ [cm].....	37
Figura 5.13 – Progresso da estimção do furo pelo filtro de Kalman utilizando $Q = Q_0$ e $x = \{5,25; 4,40; 0,06\}^T$ [cm].....	38
Figura 5.14 – Ampliação da estimção final do furo pelo filtro de Kalman utilizando $Q = Q_0$ e $x = \{5,25; 4,40; 0,06\}^T$ [cm].....	38

Figura 5.15 – Estimação das componentes do estado pelo filtro de Kalman com $Q = 25Q_0$ e $x = \{5,25; 4,40; 0,06\}^T$ [cm]	39
Figura 5.16 – Progresso da estimação do furo pelo filtro de Kalman com $Q = 25Q_0$ e $x = \{5,25; 4,40; 0,06\}^T$ [cm]	40
Figura 5.17 – Problema potencial: resultados da ANN para 5 sensores para um furo nas posições: (a) (3,0;3,0) cm; (b) (1,0;1,0) cm; e, (c) (5,0;5,0) cm. [Retirado de Lopes (2010)]..	41
Figura 5.18 – Resultado do filtro de Kalman para o furo na posição (3,00;3,00) cm	42
Figura 5.19 – Resultado do filtro de Kalman para o furo na posição (1,00;1,00) cm	42
Figura 5.20 – Resultado do filtro de Kalman para o furo na posição (5,00;5,00) cm	43
Figura 5.21 – Resultados do GA, para o furo “real” e simulado para o potencial: (a) para um furo central com elitismo igual a 2 na posição (3,00;3,00) cm; (b) para um furo central com elitismo igual a 10 na posição (3,00;3,00) cm; (c) para um furo na posição (4,0;5,0) cm. As áreas ampliadas mostram a região do furo em detalhes. [Retirado de LOPES (2010)]	43
Figura 5.22 – Resultado do filtro de Kalman para o furo na posição (4,00;5,00) cm	44
Figura 5.23 – Resultado do filtro de Kalman para o furo na posição (3,00;3,00) cm e sensores à 0,06 cm das bordas.....	45
Figura 5.24 – Resultado do filtro de Kalman para o furo na posição (3,00;3,00) cm e sensores à 0,6 cm das bordas.....	46
Figura 5.25 – Resultado do filtro de Kalman para o furo na posição (3,00;3,00) cm e sensores à 1,2 cm das bordas.....	46
Figura 5.26 – Resultado do filtro de Kalman para o furo na posição (3,00;3,00) cm e sensores à 1,8 cm das bordas.....	47
Figura 5.27 – Resultado do filtro de Kalman para o furo na posição (3,00;3,00) cm e sensores à 2,4 cm das bordas.....	47
Figura 5.28 – Resultado do filtro de Kalman para o furo na posição (1,50;2,00) cm e sensores à 0,06 cm das bordas.....	49
Figura 5.29 – Resultado do filtro de Kalman para o furo na posição (1,50;2,00) cm e sensores à 0,6 cm das bordas.....	50
Figura 5.30 – Resultado do filtro de Kalman para o furo na posição (1,50;2,00) cm e sensores à 1,2 cm das bordas.....	50
Figura 5.31 – Resultado do filtro de Kalman para o furo na posição (1,50;2,00) cm e sensores à 1,8 cm das bordas.....	51
Figura 5.32 – Resultado do filtro de Kalman para o furo na posição (1,50;2,00) cm e sensores à 1,5 cm das bordas.....	52

Figura 5.33 – Resultado do filtro de Kalman para o furo na posição (1,50;2,00) <i>cm</i> e sensores à 1,3 <i>cm</i> das bordas.....	54
Figura 5.34 – Resultado do filtro de Kalman para o furo na posição (1,50;2,00) <i>cm</i> e sensores à 1,2 <i>cm</i> das bordas com a covariância alterada.....	55
Figura 5.35 – Resultado do filtro de Kalman para o furo na posição (1,50;2,00) <i>cm</i> e sensores à 1,3 <i>cm</i> das bordas com a covariância alterada.....	55
Figura 5.36 – Furo real e simulado para a tensão média: (a) para um furo central; (b) para um furo localizado em (2,0;2,0) <i>cm</i> ; (c) para um furo localizado em (5,0;3,0) <i>cm</i> . [Retirado de Lopes (2010)]	56
Figura 5.37 – Problema elastostático: resultados obtidos da ANN para um furo nas posições: (a) (3,0;3,0) <i>cm</i> ; (b) (1,0;1,0) <i>cm</i> ; e, (c) (5,0;5,0) <i>cm</i> . [Retirado de Lopes (2010)].....	57
Figura 5.38 – Resultado do filtro de Kalman para o furo (2,00;2,00;0,12) <i>cm</i> para diversas posições de sensores	58
Figura 5.39 – Zoom na região do furo (2,00;2,00;0,12) <i>cm</i>	59

Lista de Tabelas

Tabela 5.1 – Estimação do filtro de Kalman para um quarto da covariância Q_0	31
Tabela 5.2 – Estimação do filtro de Kalman para covariância Q_0	33
Tabela 5.3 – Estimação do filtro de Kalman para a covariância de $4Q_0$	35
Tabela 5.4 – Estimação do filtro de Kalman para covariância Q_0 e estado $x = \{5,25; 4,40; 0,06\}^T$ [cm].....	37
Tabela 5.5 – Estimação do filtro de Kalman para covariância $25Q_0$ e estado $x = \{5,25; 4,40; 0,06\}^T$ [cm].....	39
Tabela 5.6 – Valores comparativos de estimação entre o ANN e o KF.....	41
Tabela 5.7 – Resultados do filtro de Kalman para o furo na posição (3,00;3,00) cm para algumas posições de sensores.....	48
Tabela 5.8 – Intervalos de confiança de 95% para o furo (3,00;3,00;0,12) cm.....	48
Tabela 5.9 – Resultados do filtro de Kalman para o furo na posição (1,50;2,00) cm para algumas posições de sensores.....	52
Tabela 5.10 – Intervalos de confiança de 95% para o furo (1,50;2,00;0,12) cm.....	53
Tabela 5.11 – Resultados do filtro de Kalman para o furo na posição (1,50;2,00) cm para uma covariância Q igual a inicial	54
Tabela 5.12 – Resultados do filtro de Kalman para o furo na posição (1,50;2,00) cm para uma nova covariância Q	56
Tabela 5.13 – Resultados do filtro de Kalman para intervalos de confiança de 95%	57

Simbologia

Letras Latinas

A	<i>Matriz de correlação do estado do instante anterior com o estado no instante posterior</i>
B	<i>Matriz de correlação do controle do estado do instante anterior com o estado no instante posterior</i>
C	<i>Matriz de correlação do ruído do estado do instante anterior com o estado no instante posterior</i>
H	<i>Matriz de correlação do estado com a medida</i>
x	<i>Vetor de estado</i>
u	<i>Vetor de entradas de controle</i>
w	<i>Vetor de ruído do estado</i>
y	<i>Vetor de observação</i>
v	<i>Vetor de ruído da medida</i>
\mathbb{R}	<i>Conjunto dos números reais</i>
$E[]$	<i>Valor esperado</i>
Q	<i>Matriz de covariância do ruído do processo</i>
R	<i>Matriz de covariância do ruído da medida</i>
P	<i>Matriz de covariância do estado</i>
K	<i>Matriz de ganho de Kalman</i>
z	<i>Vetor de medidas</i>
I	<i>Matriz Identidade</i>
$f()$	<i>Função não linear que correlaciona o estado com ele mesmo</i>
$h()$	<i>Função não linear que correlaciona o estado com a medida</i>
V	<i>Matriz de correlação do ruído de medida com a medida</i>
e	<i>Erro residual</i>

Letras Gregas

δ	<i>Deslocamento</i>
ε	<i>Erro aleatório</i>
μ	<i>Média ou valor esperado</i>
σ	<i>Tensão normal</i>
τ	<i>Tensão cisalhante</i>
\varnothing	<i>Funcional de temperatura ou tensão</i>
Δ	<i>Variação do componente do estado</i>

Sobrescritos

T	<i>Transposta de uma matriz ou vetor</i>
$-$	<i>Medição predita</i>
$-$	<i>Valor médio</i>
$^{-1}$	<i>Inversa de uma matriz</i>
\sim	<i>Valor aproximado</i>
$\hat{\sim}$	<i>Valor a posteriori</i>

Subscritos

k	<i>Relativo k-ésima iteração</i>
-----	----------------------------------

<i>x</i>	<i>Direção do eixo x</i>
<i>y</i>	<i>Direção do eixo y</i>
<i>m</i>	<i>Valor médio</i>
<i>oct</i>	<i>Valor octaédrico</i>
<i>0</i>	<i>Valor inicial</i>

Siglas e Abreviaturas

<i>ANN</i>	<i>Rede Neural Artificial</i>
<i>ARX</i>	<i>Auto Regressive eXogenous</i>
<i>BEM</i>	<i>Método de Elementos de Contorno</i>
<i>CAE</i>	<i>Engenharia Auxiliada por Computador</i>
<i>COV</i>	<i>Coefficiente de Variação</i>
<i>EKF</i>	<i>Filtro de Kalman Estendido</i>
<i>FDI</i>	<i>Detecção e Identificação de Falhas</i>
<i>FEM</i>	<i>Método de Elementos Finitos</i>
<i>GA</i>	<i>Algoritmo Genético</i>
<i>IEM</i>	<i>Instituto de Engenharia Mecânica</i>
<i>KF</i>	<i>Filtro de Kalman</i>
<i>UKF</i>	<i>Filtro de Kalman Unscented</i>

Capítulo 1

INTRODUÇÃO

Nas últimas décadas, tem-se aumentado o número de artigos voltados para o desenvolvimento de técnicas de FDI (*Failure Detection and Identification*). A maioria destas técnicas está relacionada com análise por vibrações, utilização de algoritmos de redes neurais artificiais (ANN – *Artificial Neural Networks*), algoritmo genético (GA – *Genetic Algorithm*), filtro de Kalman (KF – *Kalman Filter*) e derivados, dentre outros.

Esse grande aumento deve-se ao fato de ser cada vez mais essencial que uma estrutura ou equipamento não falte por falha, gerando uma janela vazia (paralisação da produção, diminuição da produtividade, etc.) no caso de uma linha de produção ou acidente no caso de uma estrutura, este último podendo ser grave ou não dependendo da estrutura.

Ao se analisar um caso particular de uma estrutura, uma asa de aeronave, onde sua falha é grave, pode-se verificar a importância de uma manutenção constante, segura e correta. Baseado neste exemplo, que este trabalho foi concebido.

Toda aeronave passa por manutenções durante sua vida, onde a frequência e o tipo de manutenção dependem do tipo da aeronave e do ciclo (decolagem – aterrissagem) ou horas de voo. Mas alguns pontos da aeronave são de difícil acesso, senão impossíveis sem a danificação da estrutura. Com foco na monitoração destes pontos, imaginou-se a inserção de sensores durante a fabricação, que durante uma manutenção da mesma em um hangar, teria uma janela de acesso para a comunicação dos sensores, com algum equipamento de coleta de dados.

Estes dados fornecidos pelos sensores, com a análise feita pelo método presente em uma estação de coleta de dados, indicariam a presença ou não de algum dano na região monitorada da estrutura, caracterizando assim, um problema inverso de detecção de danos. O problema inverso de detecção de danos é caracterizado pela busca de um dano onde somente a informação dos efeitos desse dano sobre estrutura está disponível, portanto qualquer outra informação sobre o dano é ausente.

A modelagem numérica, para a determinação do dano, consiste de um problema direto e de um problema inverso. De acordo com Lopes (2010), no problema direto, para se obter as informações sobre a distribuição da quantidade de interesse em toda estrutura com dano, é necessário um modelo. Já no problema inverso, o modelo é necessário para a localização do dano na estrutura, para fornecer algumas informações, parciais, sobre algumas localizações específicas (localização dos sensores).

Pode-se utilizar como ferramentas, na modelagem numérica do problema direto, o método de elementos de contorno (BEM – *Boundary Element Methods*) ou o método de elementos finitos (FEM – *Finite Element Methods*). Durante o desenvolvimento deste trabalho, foi utilizado o BEM, para apenas duas formulações (potencial e elastostática). A utilização do BEM, proporciona uma comparação com o trabalho de Lopes (2010) mais adequada, pois o BEM também é utilizado como ferramenta no modelamento do problema direto.

Assim como em Lopes (2010), os dois modelos foram construídos para uma placa com um furo interno, onde foram consideradas algumas condições de contorno, como por exemplo, a distribuição de temperatura na superfície externa da placa para a formulação potencial, e as tensões para a formulação elastostática. No problema potencial, a condução de temperatura através dos furos, é considerada nula, ou seja, os furos são adiabáticos. Da mesma forma, as tensões em furos internos na placa são assumidas como nulas. Estes modelos serão posteriormente vistos com maiores detalhes. Também, como em Lopes (2010), o dano tratado aqui é representado por um furo circular, o que não pode ser definido como dano e sim uma descontinuidade geométrica, uma vez que este furo não traz concentração de tensões semelhante a que um dano traria.

Em um primeiro momento, a formulação BEM utilizada, para furos circulares é válida. Para um aprimoramento dos resultados, aproximando-se de uma trinca (caso real), deve-se utilizar uma elipse fina o suficiente. Portanto, a formulação elastostática do BEM utilizada

neste trabalho, deve migrar para um BEM especializado em mecânica da fratura, que segundo Lopes (2010), pode ser encontrado em Portela (1992) e Aliabadi (2002).

1.1 REVISÃO DA LITERATURA

Fazendo uma breve revisão, da detecção de danos, pode-se apontar alguns trabalhos que mostram o progresso e desenvolvimento do assunto.

Kalman (1960) formulou e resolveu o problema de Wiener para estado de um ponto de vista, gerando um tratamento muito geral para casos que apresentam dificuldades com outros métodos e mostrando também ser muito próximo de problemas da teoria de controle.

Swanson et al (1999), utilizam o filtro de Kalman (KF) para localizar a trinca em uma barra através do deslocamento da frequência modal e verossimilhança de quando a falha iminente está para ocorrer.

Simani et al (2000), em seu artigo utiliza o filtro de Kalman (KF) para a detecção de danos em turbinas a gás industriais com auxílio de sensores de entrada e saída, obtendo estimadores de estado, através de técnicas de identificação padrão baseadas em ARX (*Auto Regressive eXogenous*) ou em modelos de erros em variáveis, dependendo da origem do sinal do ruído.

Gomes (2004) compara o desempenho das técnicas, redes neurais artificiais (ANN) e análise modal, para a detecção de danos em estruturas do tipo pórticos (vigas), analisando a intensidade do dano e a presença de ruído em seus dados de alimentação.

Morais (2006), em sua dissertação de mestrado, utiliza observadores de estado do tipo filtro de Kalman (KF) e com entradas identificadas por meio de funções ortogonais (Fourier, Legendre e Chebyshev) e não identificadas por meio de um observador proporcional e integral, para o diagnóstico de falhas em sistemas mecânicos.

Senthil et al (2006), inseriram a lógica fuzzy no filtro de Kalman e comparam o desempenho com o filtro de Kalman estendido (EKF – *Extended Kalman Filter*), na medição de temperatura e concentração de um tanque de reator, analisando modelos de estado de espaço, que localmente eram lineares, mas pertenciam a sistemas não lineares.

Silva (2008) estuda a resposta do filtro de Kalman (KF) e seus derivados filtro de Kalman estendido (EKF) e o filtro de Kalman *unscented* (UKF – *Unscented Kalman Filter*) para verificar qual atende melhor a arquitetura proposta para a detecção de danos.

Lopes (2010) utilizou o método de elementos de contorno como ferramenta, combinado com dois diferentes tipos de ferramentas, algoritmo genético e rede neural artificial, para resolver um problema inverso de detecção de danos em estruturas.

1.2 OBJETIVO

Este trabalho tem como objetivo principal, buscar uma metodologia, para detecção de danos em estruturas, utilizando o filtro de Kalman. Conseqüentemente, tendo também como objetivo e contribuição, a comparação de métodos inversos para o mesmo problema, o Filtro de Kalman (KF) e Redes Neurais Artificiais (ANN), Algoritmo Genético (GA).

Um objetivo secundário deste trabalho é o fato dele ser uma indicação como trabalho inicial e caminho a tomar, para o desenvolvimento e aprimoramento de técnicas de detecção de danos, para aplicação na indústria.

1.3 HIPÓTESE

No desenvolvimento deste trabalho, foram feitas considerações com intuito de simplificação dos modelos e matemática utilizados, além de outras importantes que caracterizam o mesmo. As mais importantes e relevantes são:

- O dano aqui é tratado como um furo circular, ou seja, uma descontinuidade geométrica;
- O estado, coordenadas e raio do furo, não se alteram com o tempo (sistema invariante no tempo);
- Não há qualquer tipo de controle exercido sobre o modelo;

- O ruído presente é considerado de baixa amplitude e branco;
- As configurações do problema direto, dimensões e condições de contorno, para os modelos utilizados (potencial e elastostático) são de caráter acadêmico, para comparação de metodologias.

1.4 RELEVÂNCIA

O trabalho visa o aprimoramento de técnicas de detecção e localização de danos (descontinuidades geométricas) em estruturas mecânicas, para aplicação na indústria, no contexto de técnicas e sistemas de monitoramento da saúde da estrutura.

Uma vez que as técnicas existentes para a detecção e localização de danos, relacionadas ao filtro de Kalman (KF), trabalham especialmente com sistemas dinâmicos e, portanto fazem uso da análise de vibrações atuantes no sistema e na variação de propriedades do mesmo, o trabalho assume uma característica particular.

Logo, este trabalho traz uma alternativa para as técnicas já existentes de detecção de danos em estruturas, utilizando o filtro de Kalman como ferramenta, além do aprimoramento das mesmas.

Capítulo 2

MODELO DIRETO: MÉTODO DE ELEMENTOS DE CONTORNO

A formulação do problema inverso de detecção de danos é constituído de três problemas a serem modelados. O modelamento do problema direto, para levantamento dos parâmetros necessários, o modelamento do problema inverso, propriamente dito, para resolver o problema e o modelamento das incertezas de cada um dos problemas.

A modelagem correspondente ao problema direto pode ser feita através do método dos elementos finitos (FEM) ou através do método dos elementos de contorno (BEM). O BEM apesar de apresentar uma melhor precisão em comparação ao FEM, o que necessário em alguns problemas (problemas com concentração de tensão), não pode ser sempre aplicado, pois em alguns problemas não é possível obter a solução fundamental. Outro ponto positivo do BEM é de que se necessita somente discretizar a superfície e não o volume inteiro da peça [Lopes, 2010; Brebbia & Dominguez, 1992].

A modelagem do problema inverso é realizada com técnicas de otimização ou identificação de parâmetros. Neste trabalho, será utilizado apenas a técnica de identificação de parâmetros, no caso, o filtro de Kalman (KF) para a modelagem do problema inverso.

Neste capítulo, estão apresentados os conceitos do BEM para as formulações, potencial e elastostática utilizadas, que são necessárias para o entendimento do problema direto.

2.1 MÉTODO DE ELEMENTOS DE CONTORNO (BEM)

Como mencionado anteriormente, o BEM é uma alternativa melhor ao FEM nos casos em que uma melhor precisão é requerida, como nos casos de concentração de tensão onde se estende para o infinito [Brebbia & Dominguez, 1992]. Devido necessidade do FEM, em algumas aplicações da engenharia, em calcular integrais, definir e redefinir as malhas, os sistemas de engenharia auxiliados por computador (CAE – *Computer Aided Engineering*), que utilizam o método, podem se tornar lentos devido à complexidade destes cálculos.

O BEM fornece informações do domínio presentes no contorno, fazendo com que o problema seja descrito pelos acontecimentos sobre o contorno, o que reduz a dimensão do problema e simplifica numericamente o tratamento do mesmo. Logo para os modelos utilizados neste trabalho, são necessárias apenas as informações dos contornos da placa e do furo, para que se obtenha a informação de quaisquer pontos internos do domínio de interesse.

O BEM transforma a equação diferencial parcial de um domínio em um conjunto de equações integrais, onde são relacionadas as variáveis, conhecidas e desconhecidas, do contorno [Brebbia & Dominguez, 1992; Lopes, 2010].

2.1.1 Modelo potencial

No modelo potencial utilizado, a placa e o furo são divididos por elementos constantes, onde seus nós deverão conter informações nas direções x e y , sobre a temperatura T ou fluxo de calor q prescrito no mesmo. A quantidade de elementos que dividem a placa e o furo não necessita ser a mesma, logo esta quantidade fica como critério do usuário do método para uma definição, lembrando que quanto maior for a divisão melhor serão os resultados (mais precisos), mas com isso o custo computacional também aumenta.

Na figura a seguir, a placa esta dividida por 36 elementos constantes e o furo por oito elementos constantes. É possível verificar também pela Figura 2.1, que é somente necessário a informação contida no contorno da placa.

Com o fornecimento das informações necessárias, o BEM irá retornar apenas os valores de temperaturas para quaisquer pontos de interesse presentes no domínio da placa (esses pontos, em um caso real, seriam os sensores).

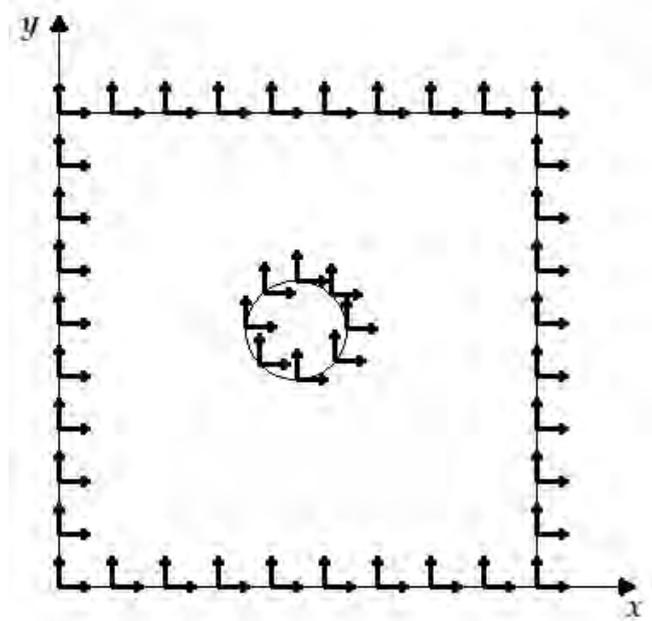


Figura 2.1 – Direções das informações dos nós dos contornos

2.1.2 Modelo elastostático

Assim como no modelo potencial, a placa e o furo são divididos por elementos constantes, onde seus nós deverão conter informações nas direções x e y , mas agora sobre a tensão, normal σ ou cisalhante τ , ou deslocamento δ prescrito no mesmo.

Estas tensões, normal e cisalhante, não podem ser usadas como resposta do BEM no problema de detecção de danos, pois elas dependem do sistema de coordenadas utilizado, ou da direção normal do plano de corte que passa pelo ponto de interesse [Lopes, 2010]. Portanto é necessário usar invariantes de tensão, que podem ser a tensão normal média ou a tensão octaédrica.

A tensão média, para o caso bi-dimensional, que é a utilizada no trabalho de Lopes (2010), é definida por:

$$\sigma_m = \frac{\sigma_x + \sigma_y}{2} \quad (1)$$

Onde,

σ_x é a tensão normal na direção do eixo x ;

σ_y é a tensão normal na direção do eixo y ;

E a tensão octaédrica, também para o caso bi-dimensional, é definida por:

$$\tau_{oct} = \sqrt{\frac{2}{9} \left[(\sigma_x + \sigma_y)^2 - 3(\sigma_x \sigma_y - \tau_{xy}^2) \right]} \quad (2)$$

Logo, ao contrário da formulação do BEM para o problema potencial, o BEM na formulação elastostática fornece duas informações em um ponto, que são a tensão normal e a tensão octaédrica.

Capítulo 3

PROBLEMA INVERSO: FILTRO DE KALMAN

O filtro de Kalman (KF) foi desenvolvido por R. E. Kalman em 1960 para resolver o modelo de estado de espaço de Wiener, de forma recursiva. Aliando o filtro com os avanços da computação digital, a técnica teve forte aplicação em automação e navegação assistida [Welch & Bishop, 2001].

Muitos pesquisadores têm desenvolvido variações do filtro para diversos tipos de aplicação, seja para navegação, automação, indústria química, detecção de danos, além de outras. Mas todos tendo em comum, a identificação de parâmetros dos sistemas analisados.

O filtro de Kalman (KF), originalmente foi desenvolvido para sistemas lineares, discretos e que possuem ruído gaussiano branco, o que em sistemas reais não ocorre, mas que podem, em algumas regiões, serem aproximados para lineares, ampliando a sua aplicação.

Mas para que o filtro de Kalman (KF) seja capaz de estimar o estado, é necessário que o processo seja observável [Simon, 2006]. Neste capítulo estão apresentados os conceitos principais do filtro de Kalman (KF) e do filtro de Kalman estendido (EKF), que são os mais comumente utilizados.

3.1 CONTROLABILIDADE E OBSERVABILIDADE

A seguir serão fornecidas as definições de controlabilidade e de observabilidade para sistemas contínuos e discretos no tempo, além dos conceitos de estabilidade e detectabilidade. Estes conceitos foram inseridos por R. E. Kalman em seu trabalho [Simon, 2006], referente a sistemas lineares, e podem ser estendidos para sistemas não lineares, onde um modelo de filtro de Kalman estendido é obtido a partir de uma linearização do modelo não linear.

3.1.1 Controlabilidade

Um sistema linear, contínuo no tempo, é dito controlável se para qualquer estado inicial $x(0)$ e tempo final $t > 0$, existir um controle que transfere o estado para qualquer valor desejado no tempo t . Já um sistema linear, discreto no tempo, é dito controlável se para qualquer estado inicial x_0 e tempo final k , existir um controle que transfere o estado para qualquer valor desejado no tempo k .

A diferença nas definições para o tempo contínuo e o discreto, é que no contínuo a transferência é para qualquer tempo final e no discreto é para algum tempo final [Simon, 2006].

3.1.2 Observabilidade

Um sistema contínuo no tempo é dito observável se para qualquer estado inicial $x(0)$ e tempo final $t > 0$ o estado inicial $x(0)$ puder ser unicamente determinado pelo conhecimento de entradas $u(\tau)$ e saídas $y(\tau)$ para todo $\tau \in [0, t]$. Já um sistema linear, discreto no tempo, é dito observável se para qualquer estado inicial x_0 e tempo final k o estado inicial x_0 puder ser unicamente determinado pelo conhecimento de entradas u_i e saídas y_i para todo $i \in [0, k]$.

Novamente, a diferença nas definições para o tempo contínuo e o discreto, é que no contínuo a transferência é para qualquer tempo final e no discreto é para algum tempo final. Se o sistema é observável, o estado inicial pode ser determinado e todos os estados entre o inicial e o final também podem ser determinados [Simon, 2006].

3.1.3 Estabilidade e detectabilidade

Os conceitos de estabilidade e detectabilidade são relacionados a controlabilidade e a observabilidade, respectivamente. Eles também são relacionados com os modos do sistema. Para ver exemplos veja Simon (2006), página 43.

Se o sistema é controlável ou estável, então ele é estabilizado. Se ele é não controlável ou instável, então ele é estabilizado se seus modos não controláveis são estáveis.

Se o sistema é observável ou estável, então ele é detectável. Se ele é não observável ou instável, então ele é detectável se seus modos não observáveis são estáveis.

3.2 FILTRO DE KALMAN DISCRETO (KF)

O filtro de Kalman discreto (KF) é indicado para uso em sistemas dinâmicos de comportamento linear, e com ruídos Gaussianos brancos, ou seja, casos ideais. Este filtro é o filtro original de Kalman e que deu base para o desenvolvimento de seus derivados, o filtro de Kalman estendido (EKF) e o filtro de Kalman *unscented* (UKF), além de inúmeros outros filtros híbridos.

O desenvolvimento a seguir é adaptado de Welch & Bishop (2001). Um sistema dinâmico, variando linearmente no tempo, pode ser modelado por:

$$x_k = Ax_{k-1} + Bu_{k-1} + Cw_{k-1} \quad k \geq 1 \quad (3)$$

$$y_k = H_k x_k + v_k \quad (4)$$

Onde,

$x_k \in \mathbb{R}^n$ é o estado do sistema no instante k ;

A é a matriz quadrada que correlaciona o estado no instante $k - 1$ com o instante k ;

$u_{k-1} \in \mathbb{R}^m$ é o vetor de entradas de controle no instante $k - 1$;

B é a matriz quadrada que correlaciona o controle ao estado;

$w_{k-1} \in \mathbb{R}^n$ é vetor que transmite as fontes de erros do sistema;

$y_k \in \mathbb{R}^r$ é o vetor de observação (medidas);

H_k é a matriz de sensibilidade que correlaciona o estado com a medição;

$v_k \in \mathbb{R}^r$ é o vetor que transmite as fontes de erros da medição;

Sendo ainda, $\{w\}$ e $\{v\}$, ruídos Gaussianos, brancos, com média zero.

$$E[w_k] = E[v_k] = 0 \quad (5)$$

E a matriz de covariância conjunta,

$$E \left[\begin{pmatrix} w_k \\ v_k \end{pmatrix} \begin{pmatrix} w_k^T & v_k^T \end{pmatrix} \right] = \begin{bmatrix} Q_k & 0 \\ 0 & R_k \end{bmatrix} \quad (6)$$

Com,

- Q_k sendo a matriz de covariância do ruído do processo;
- R_k sendo a matriz de covariância do ruído da medição;

O estado inicial, x_0 , é um vetor aleatório Gaussiano com média definida por:

$$E[x_0] = \bar{x}_0 \quad (7)$$

E a matriz de covariância definida por:

$$E[(x_0 - \bar{x}_0)(x_0 - \bar{x}_0)^T] = P_0 \quad (8)$$

O filtro de Kalman é composto por duas fases, a fase de predição e a fase de correção. Após a estimativa inicial, o filtro de Kalman inicia um ciclo de atualização das duas fases até a convergência do estado, ou seja, o estado predito x_k corresponde ao estado que gerou a medição z_k . Este ciclo está exemplificado pela figura a seguir.

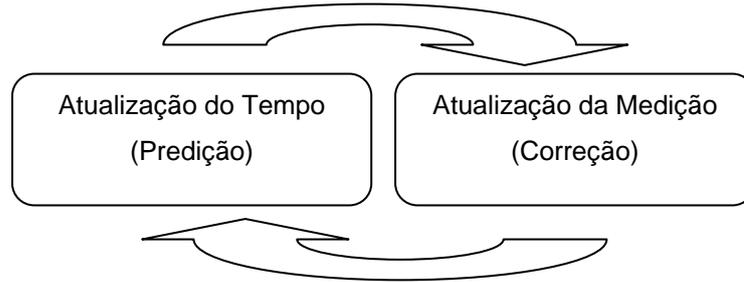


Figura 3.1 – Ciclo do processo do Filtro de Kalman

A fase de predição é composta pelas duas equações seguintes:

$$x_k^- = Ax_{k-1} + Bu_{k-1} + Cw_{k-1} \quad (9)$$

$$P_k^- = AP_{k-1}A^T + Q_k \quad (10)$$

Sendo a Equação (9) igual a (3) e o sinal sobrescrito de menos, no estado e na covariância do estado (x_k^- e P_k^-), é utilizado para definir o estado e a covariância do estado predito, respectivamente.

Já fase de correção é composta pelas três equações seguintes:

$$K_k = P_k^- H^T (HP_k^- H^T + R)^{-1} \quad (11)$$

$$x_k = x_k^- + K(z_k - Hx_k^-) \quad (12)$$

$$P_k = (I - K_k H)P_k^- \quad (13)$$

Estas equações estão presentes na figura seguinte onde um algoritmo do filtro de Kalman está representado. A Equação (12) justifica a origem probabilística do filtro. Essa justificativa tem a origem na probabilidade da estimativa *a priori* x_k^- condicionada para todas as medições anteriores z_k (Regra de *Bayes*). Mas primeiramente, basta apenas saber que o filtro de Kalman mantém os dois primeiros momentos da distribuição do estado (x_k e P_k) [Welch & Bishop, 2001]. A Equação (11) define o ganho de Kalman, que é responsável por dar o passo para o qual a estimação “caminha”. A figura a seguir traz um algoritmo geral do filtro de Kalman.

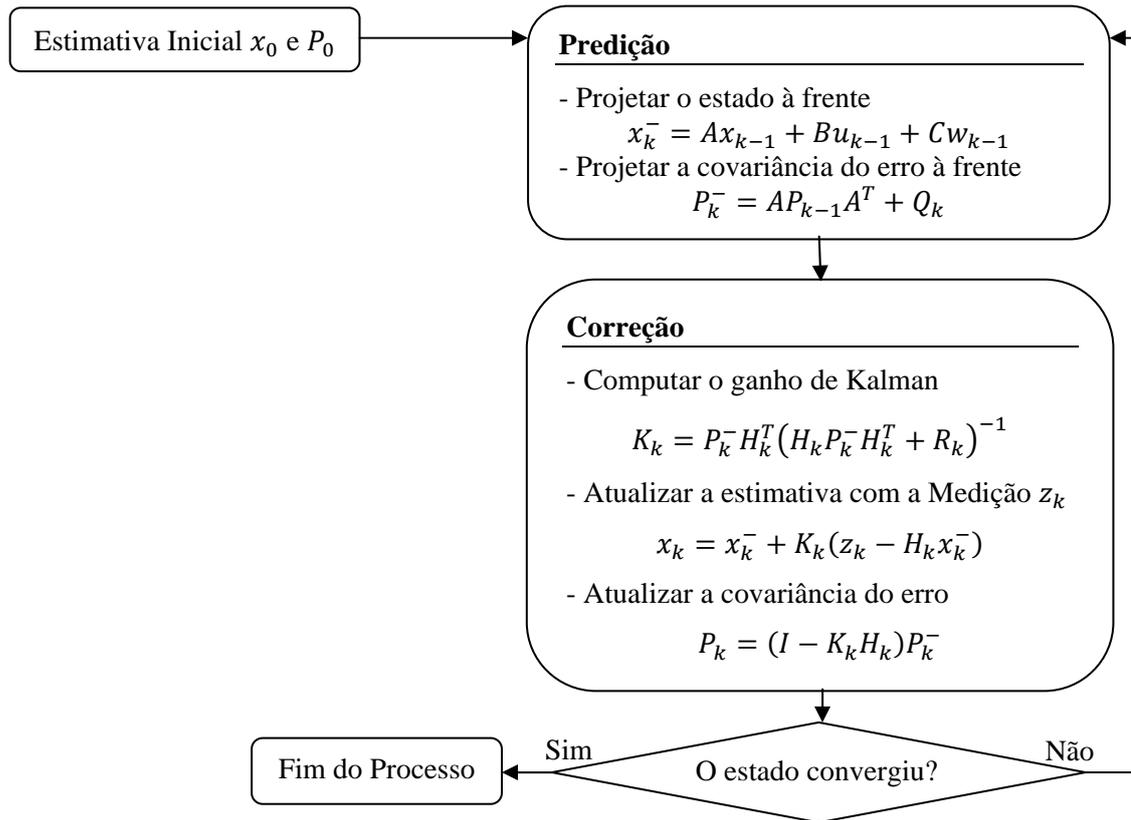


Figura 3.2 – Algoritmo do filtro de Kalman discreto

Na prática, durante a implementação do filtro, a covariância do erro da medição R é feita antes da operação do filtro. Esta medição é possível, porque de maneira geral, pode-se ser capaz de tomar algumas medidas de amostra *off-line*, ou seja, elas podem ser coletadas fora do sistema analisado. Essa definição da covariância anterior a operação do filtro, é feita a fim de determinar a variância do ruído de medição.

Já o ruído da medição da covariância do processo Q é mais difícil de ser determinada, pois não temos a capacidade de observar diretamente o processo a ser estimado. Em modelos de processo simples, pode-se obter resultados aceitáveis através da inserção de incertezas na escolha de Q . Neste caso o processo de medição teria de ser confiável.

De qualquer forma, se uma base racional estiver ou não disponível para a escolha dos parâmetros, o desempenho do filtro pode ser obtido através do ajuste de Q e R [Welch & Bishop, 2001]. Esse ajuste é feito *off-line* com outros filtros de Kalman em um sistema de identificação designado. Estes outros filtros de Kalman são desenvolvidos, unicamente, para obtenção destes parâmetros.

No final nota-se que, sob condições em que Q e R são constantes, P_k e K_k estabilizarão rapidamente e permanecerão constantes. Mas freqüentemente o erro de medição não permanece constante.

3.3 FILTRO DE KALMAN ESTENDIDO (EKF)

Como foi mostrado na Seção 3.1, o filtro de Kalman discreto (KF) trabalha com sistemas gerais governados por equações lineares. Quando o sistema é governado por equações não lineares e o filtro de Kalman lineariza a média e a covariância e ele é tratado como filtro de Kalman estendido (EKF). O Filtro de Kalman estendido foi proposto originalmente por Stanley Schmidt e sua idéia pode ser aplicada a problemas não lineares de navegação de espaçonaves [Simon, 2006].

Essa linearização é realizada através de derivadas parciais das equações do processo e das equações de medições, no caso delas se assemelharem a uma série de Taylor [Welch & Bishop, 2001].

Seja o sistema governado pela equação não linear:

$$x_k = f(x_{k-1}, u_{k-1}, w_{k-1}) \quad (14)$$

E as medições dadas por:

$$z_k = h(x_k, v_k) \quad (15)$$

Onde $f(\cdot)$ e $h(\cdot)$ são funções não lineares e, w_k e v_k representam novamente os ruídos do processo e medição.

Como os valores individuais na prática não estão disponíveis, aproximações sem o uso dos ruídos são utilizadas.

$$\tilde{x}_k = f(\hat{x}_{k-1}, u_{k-1}, 0) \quad (16)$$

$$\tilde{z}_k = h(\tilde{x}_k, 0) \quad (17)$$

Os sinais de ênfase “~” e “^” correspondem ao valor aproximado e a um valor *a posteriori* anterior a iteração k , respectivamente.

Logo, as equações governantes que irão linearizar as Equações (14) e (15), podem ser escritas como:

$$x_k \approx \tilde{x}_k + A(x_{k-1} - \hat{x}_{k-1}) + Cw_{k-1} \quad (18)$$

$$z_k \approx \tilde{z}_k + H(x_k - \tilde{x}_k) + Vv_k \quad (19)$$

Onde,

x_k e z_k são os vetores atuais de estado e medição,

\tilde{x}_k e \tilde{z}_k são os vetores aproximados do estado e da medição das Equações (14) e (15),

\hat{x}_k é uma estimativa *a posteriori* do estado na iteração k ,

As variáveis aleatórias w_k e v_k são os ruídos de processo e de medição respectivamente,

A é a matriz Jacobiana de derivadas parciais de f com relação a x ,

$$A_{[i,j]} = \frac{\partial f_{[i]}}{\partial x_{[j]}}(\hat{x}_{k-1}, u_{k-1}, 0)$$

C é a matriz Jacobiana de derivadas parciais de f com relação a w ,

$$C_{[i,j]} = \frac{\partial f_{[i]}}{\partial w_{[j]}}(\hat{x}_{k-1}, u_{k-1}, 0)$$

H é a matriz Jacobiana de derivadas parciais de h com relação a x ,

$$H_{[i,j]} = \frac{\partial h_{[i]}}{\partial x_{[j]}}(\tilde{x}_k, 0)$$

V é a matriz Jacobiana de derivadas parciais de h com relação a x ,

$$V_{[i,j]} = \frac{\partial h_{[i]}}{\partial v_{[j]}}(\tilde{x}_k, 0)$$

Agora, pode-se definir o erro predito (20) e a medição residual (21) como,

$$\tilde{e}_{x_k} \equiv x_k - \tilde{x}_k \quad (20)$$

$$\tilde{e}_{z_k} \equiv z_k - \tilde{z}_k \quad (21)$$

Deve-se lembrar que a informação de x_k em (18) não está disponível, mas deve-se ter a informação de z_k em (19). Logo com (20) e (21), é possível escrever as equações governantes para o erro de processo como,

$$\tilde{e}_{x_k} \approx A(x_{k-1} - \hat{x}_{k-1}) + w_k \quad (22)$$

$$\tilde{e}_{z_k} \approx H\tilde{e}_{x_k} + v_k \quad (23)$$

As variáveis w_k e v_k são variáveis aleatórias independentes com média zero e matrizes de covariância CQC^T e VRV^T , com Q e R semelhantes ao filtro de Kalman discreto (KF).

É possível notar que as Equações (22) e (23) são lineares e se assemelham as Equações (3) e (4).

Da mesma forma do que no filtro de Kalman discreto (KF), tem-se as duas fases, predição e correção que formaram o ciclo (Figura 3.1).

Sendo a fase de predição composta pelas equações,

$$\tilde{x}_k^- = f(\tilde{x}_{k-1}, u_{k-1}, 0) \quad (24)$$

$$P_k^- = A_k P_{k-1} A_k^T + C_k Q_{k-1} C_k^T \quad (25)$$

E a fase de correção pelas equações,

$$K_k = P_k^- H_k^T (H_k P_k^- H_k^T + V_k R_k V_k^T)^{-1} \quad (26)$$

$$\tilde{x}_k = \tilde{x}_k^- + K_k (z_k - h(\tilde{x}_k^-, 0)) \quad (27)$$

$$P_k = (I - K_k H_k) P_k^- \quad (28)$$

Se todas as medições entre as medições z_k e o estado através de $h(\cdot)$ não estão mapeadas uma a uma, o filtro diverge rapidamente e neste caso, o processo é não observável [Welch & Bishop, 2001]. A figura a seguir exemplifica o algoritmo do filtro de Kalman estendido.

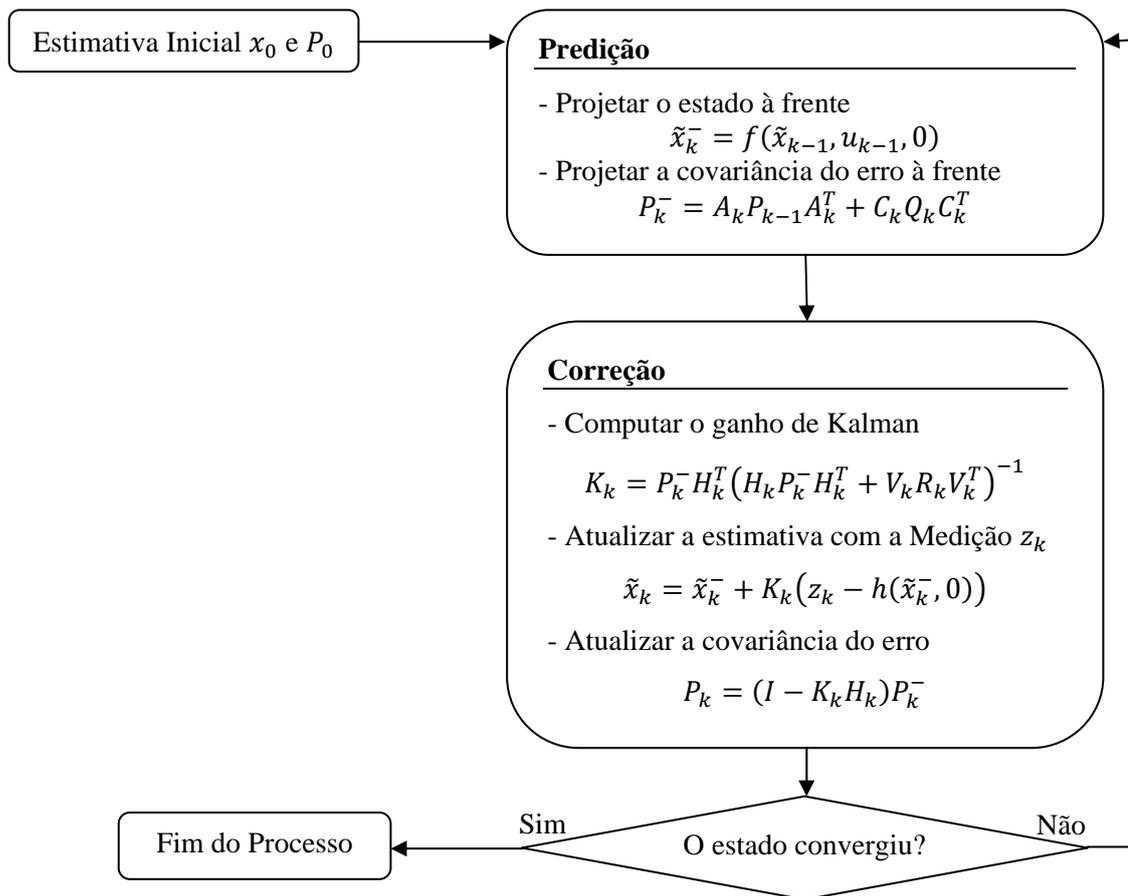


Figura 3.3 – Algoritmo do filtro de Kalman estendido

Capítulo 4

DESENVOLVIMENTO

Neste capítulo estão contidas as informações sobre os modelos e técnicas utilizadas para a execução do trabalho. Os modelos estão explicados com maiores detalhes, assim como pontos importantes da simulação que devem ser considerados.

Os modelos dos problemas potencial e elastostático foram adaptados de Brebbia & Dominguez (1992). Os códigos e a discretização implementados para esses problemas diretos são os mesmos adotados em Lopes (2010), para permitir a comparação dos resultados do filtro de Kalman com os diferentes métodos inversos empregados naquele trabalho.

4.1 MODELOS DIRETOS

O modelo potencial bi-dimensional, consiste da análise da distribuição de temperatura sobre a superfície de uma placa, quadrada, de seis centímetros de lado. É aplicado um diferencial de temperatura de $300\text{ }^{\circ}\text{C}$ entre duas arestas e sendo as outras duas arestas e o furo adiabáticos, ou seja, o fluxo de calor é nulo. A figura a seguir, exemplifica o modelo potencial da placa.

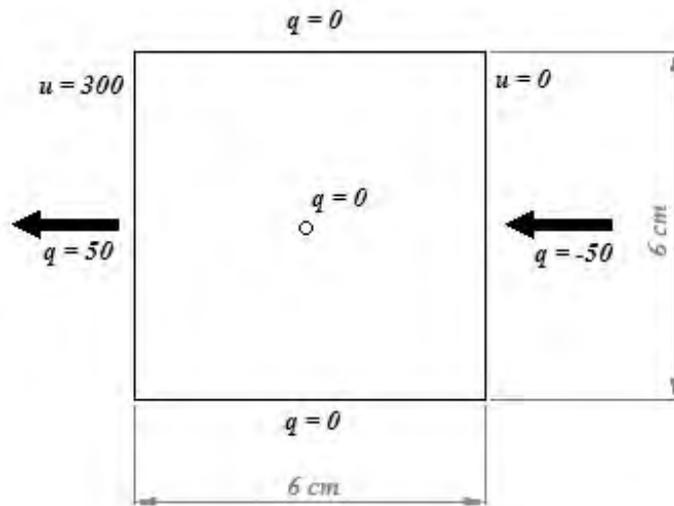


Figura 4.1 – Modelo da placa para o modelo potencial. [Adaptado de Lopes (2010)]

Para este modelo foi considerado um furo de $0,06\text{ cm}$ de raio e os sensores foram posicionados, inicialmente, a $0,6\text{ cm}$ de distância das bordas da placa, sendo em total de oito sensores. Este posicionamento de sensores foi adotado, imaginando-se um melhor aproveitamento aliado com a possibilidade de um fácil posicionamento real de termopares nas coordenadas.

Com relação ao modelo desenvolvido no BEM, a placa foi dividida em 12 elementos constantes e o furo em 24 elementos constantes, esta divisão pode ser observada na figura seguinte, assim como o posicionamento dos sensores.

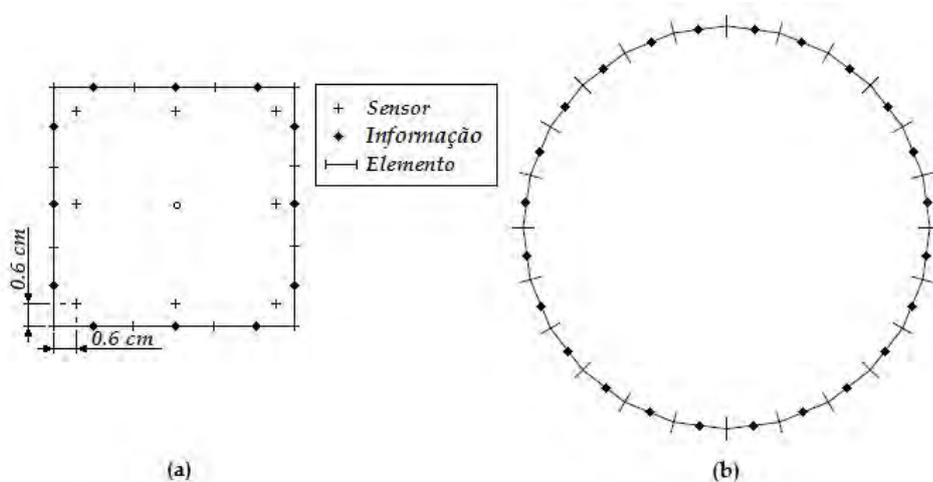


Figura 4.2 – Modelo potencial: (a) discretização da placa e posicionamento dos sensores; (b) discretização do furo.

Os códigos utilizados para o problema direto no modelo potencial encontram-se no Apêndice C deste trabalho.

Já o modelo elastostático, constitui-se de uma placa de mesmas dimensões, onde se é aplicada uma tensão de 1000 MPa tracionando a placa na direção do eixo das ordenadas e deixando os outros lados livres. Com relação ao furo, nele é considerado que há ausência de tensão. A figura seguinte exemplifica o modelo.

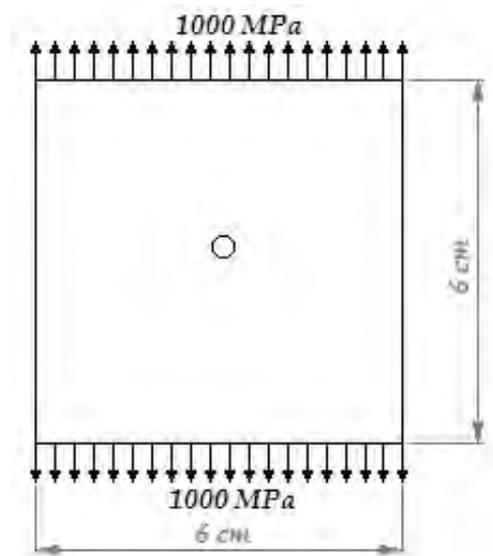


Figura 4.3 – Modelo da placa para o modelo elastostático.[Adaptado de Lopes (2010)]

Com relação ao material da placa, esta foi simulada considerando um módulo de cisalhamento de $94,5 \text{ GPa}$ e um coeficiente de Poisson para a deformação plana de $0,1$ [Brebbia & Dominguez, 1992; Lopes, 2010].

Neste modelo, o furo utilizado possui um raio de $0,12 \text{ cm}$ e os sensores estão também posicionados a $0,6 \text{ cm}$ de distância das bordas da placa, totalizando oito sensores também. Com relação ao número de elementos que este modelo foi dividido, a placa está dividida por 24 elementos constantes e o furo por 12 elementos constantes. Assim como na Figura 4.2, a figura seguinte exemplifica o posicionamento dos sensores a divisão pelos elementos na placa e no furo.

Deve-se chamar a atenção para o fato de se utilizar elementos constantes no modelo elastostático, pois este tipo de elemento não é o indicado para este modelo, mesmo com o uso de malhas grosseiras que auxiliam na estabilidade da solução do problema.

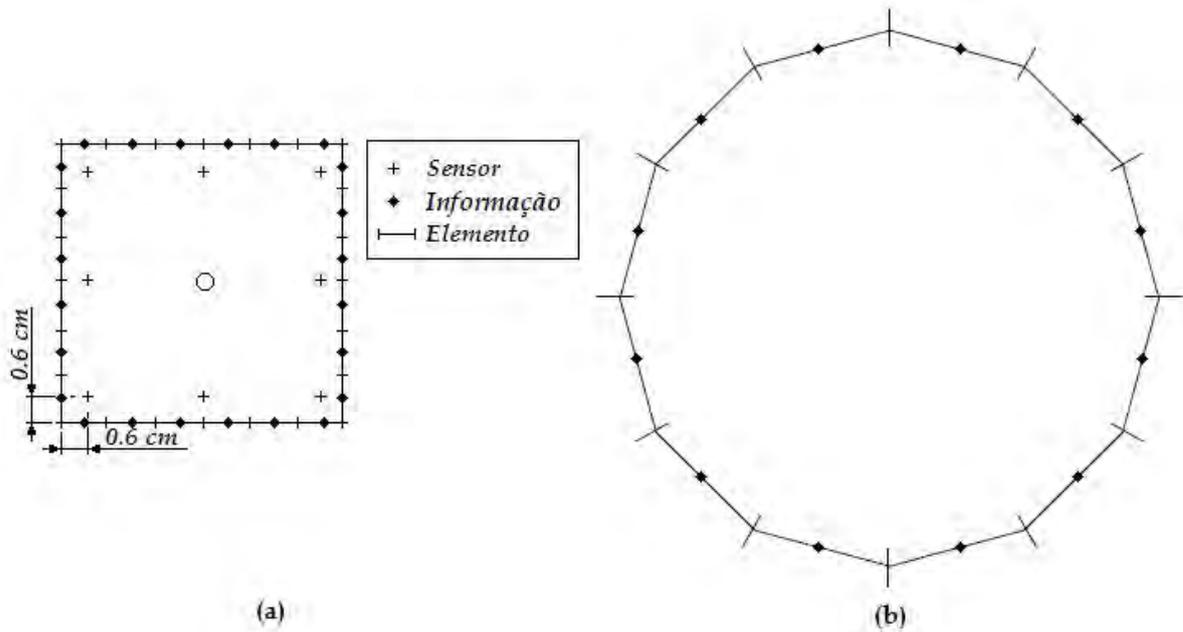


Figura 4.4 – Modelo elastostático: (a) discretização da placa e posicionamento dos sensores; (b) discretização do furo.

Os códigos utilizados para o problema direto no modelo elastostático, encontram-se no Apêndice D deste trabalho.

4.2 MODELO INVERSO

Para ambos os modelos, no ajuste dos parâmetros do filtro de Kalman, devem ser escolhidas a estimativa inicial e covariância do estado, as covariâncias dos ruídos de processo e medida, sendo que estas não irão variar durante as iterações do filtro.

Em relação aos modelos de filtro apresentados no Capítulo 3, um modelo foi desenvolvido neste trabalho na forma de um filtro híbrido, constituído por uma parte do KF e uma parte do EKF. Esta mistura das idéias do filtro é devido ao fato do estado ser estacionário (invariante no tempo), pois a relação do estado na iteração $k + 1$ com o estado na iteração k é linear, logo se tratando das idéias contidas no KF, já a relação do estado com a medida é não linear, referenciando-se assim as idéias do EKF.

Com isto e lembrando que o ruído possui valor médio igual a zero e não existe nenhum controle aplicado, a Equação (9) pode ser escrita como:

$$x_k^- = Ix_{k-1} \quad (29)$$

Sendo I é a matriz identidade. Já a Equação (10) pode ser escrita como:

$$P_k^- = IP_{k-1}I^T + Q \quad (30)$$

Portanto, as equações do filtro utilizadas no trabalho serão a (29) e (30) para a fase de predição e para a fase de correção serão as Equações (31), (32) e (33):

$$K_k = P_k^- H_k^T (H_k P_k^- H_k^T + I R I^T)^{-1} \quad (31)$$

$$x_k = x_k^- + K_k (z_k - h(x_k^-, 0)) \quad (32)$$

$$P_k = (I - K_k H_k) P_k^- \quad (33)$$

O termo $h(x_k^-, 0)$ da Equação (32) é fornecido através do BEM para o estado da iteração k , gerando assim um funcional que deve ser minimizado ($z_k - h(x_k^-, 0)$).

Segundo Simon (2006), a Equação (33) é mais utilizada na implementação devido ao menor custo computacional, mas é recomendado substituí-la pela seguinte para que se garanta uma simetria na matriz de covariância P_k .

$$P_k = (I - K_k H_k) P_k^- (I - K_k H_k)^T + K_k R K_k^T \quad (34)$$

A matriz de sensibilidade H_k , exemplificada pela equação seguinte, é uma matriz de derivadas, conforme mostrado na Seção 3.3, onde neste trabalho ela é calculada por diferenças finitas ao redor do estado na iteração k . Essas diferenças são feitas aplicando-se uma variação positiva e negativa, em cada um dos três componentes do estado (abscissa x , ordenada y e raio r do furo). Portanto esta variação também se torna um parâmetro de ajuste do filtro, que não se altera com as iterações:

$$H_k = \begin{bmatrix} \frac{(\phi_{x+\Delta x} - \phi_{x-\Delta x})_1}{2\Delta x} & \frac{(\phi_{y+\Delta y} - \phi_{y-\Delta y})_1}{2\Delta y} & \frac{(\phi_{r+\Delta r} - \phi_{r-\Delta r})_1}{2\Delta R} \\ \frac{(\phi_{x+\Delta x} - \phi_{x-\Delta x})_2}{2\Delta x} & \frac{(\phi_{y+\Delta y} - \phi_{y-\Delta y})_2}{2\Delta y} & \frac{(\phi_{r+\Delta r} - \phi_{r-\Delta r})_2}{2\Delta R} \\ \vdots & \vdots & \vdots \\ \frac{(\phi_{x+\Delta x} - \phi_{x-\Delta x})_i}{2\Delta x} & \frac{(\phi_{y+\Delta y} - \phi_{y-\Delta y})_i}{2\Delta y} & \frac{(\phi_{r+\Delta r} - \phi_{r-\Delta r})_i}{2\Delta R} \end{bmatrix}_k \quad (35)$$

Onde:

$(\phi_{x+\Delta x} - \phi_{x-\Delta x})_i$ é a diferença de temperatura ou tensão para a variação do centro do furo com relação ao deslocamento em x fornecida pelo i -ésimo sensor;

$(\phi_{y+\Delta y} - \phi_{y-\Delta y})_i$ é a diferença de temperatura ou tensão para a variação do centro do furo com relação ao deslocamento em y fornecida pelo i -ésimo sensor;

$(\phi_{r+\Delta r} - \phi_{r-\Delta r})_i$ é a diferença de temperatura ou tensão para a variação do furo com relação a variação do raio r , fornecida pelo i -ésimo sensor;

A figura seguinte exemplifica essa variação em cada um dos três componentes do estado, para uma dada a localização (x e y) e o tamanho (r).

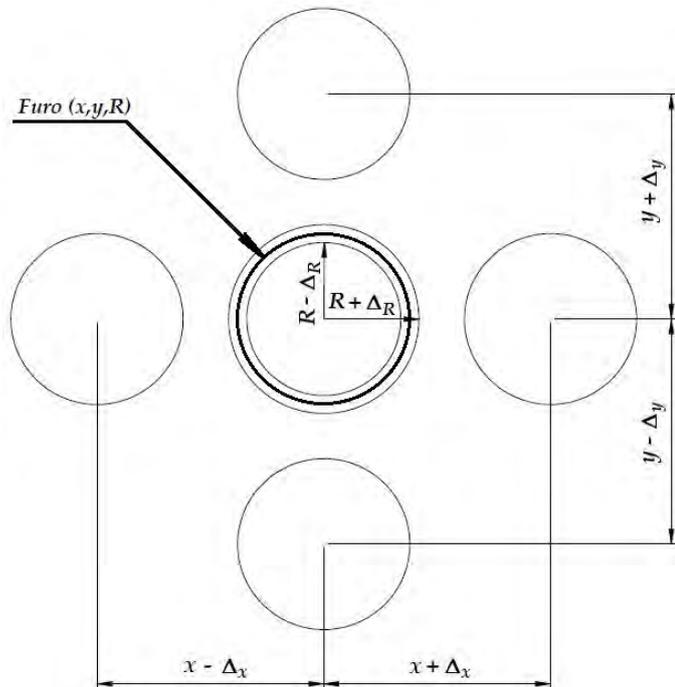


Figura 4.5 – Posições relativas ao furo utilizadas para o cálculo da matriz de sensibilidade.

Logo, pode-se escrever um algoritmo para o filtro de Kalman utilizado, conforme a figura seguinte mostra.

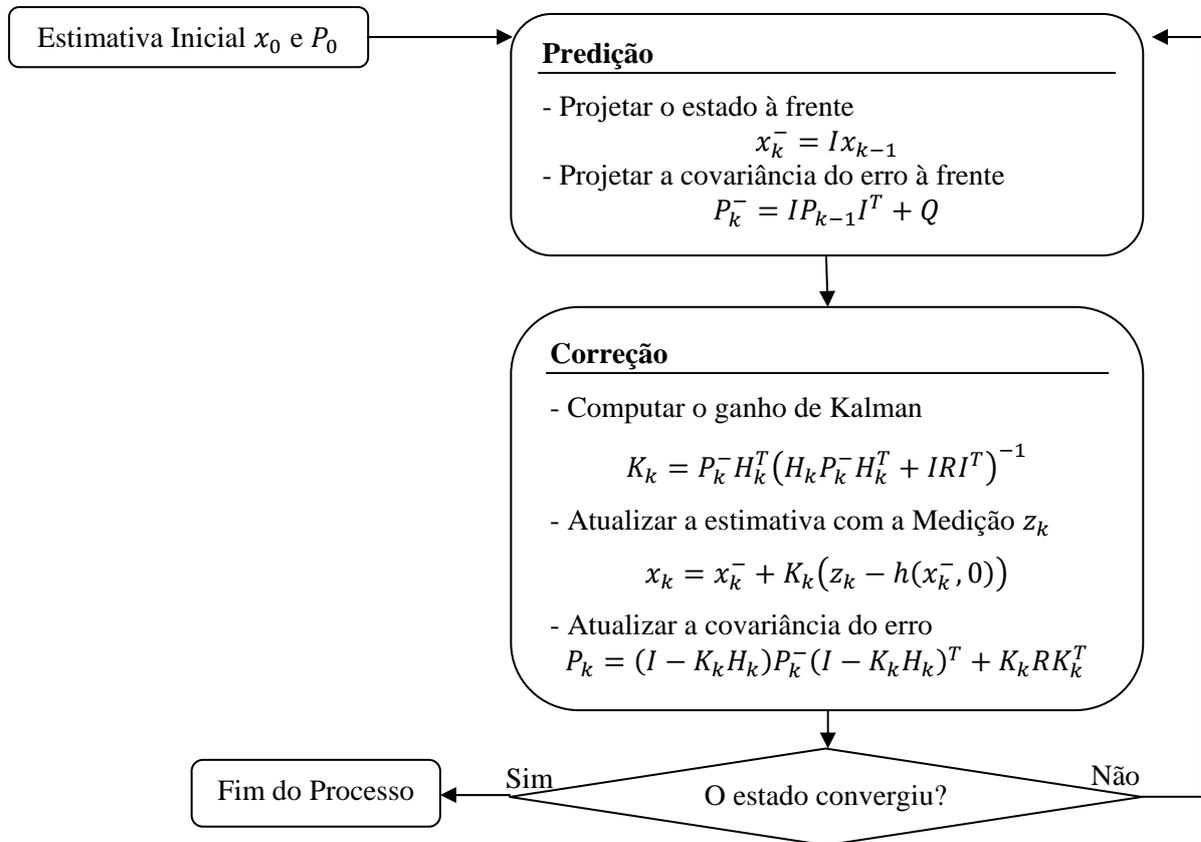


Figura 4.6 – Algoritmo do filtro de Kalman utilizado

Os códigos utilizados para o modelo inverso encontram-se nos Apêndices C e D deste trabalho, que se referem aos dois modelos diretos utilizados, potencial e elastostático, respectivamente.

Portanto até o momento, tem-se um total de oito parâmetros de ajuste, quatro sendo do modelo direto e quatro do modelo inverso. Os parâmetros do modelo direto são: a posição dos sensores, quantidade de sensores e a quantidade de elementos dos contornos, da placa e do furo. Já os parâmetros do modelo inverso são: as covariâncias do ruído da medida e do processo, a variação dos componentes do estado para o cálculo da matriz de sensibilidade e a tolerância considerada para convergência do resultado do filtro. Configurando desta forma, inúmeras possibilidades de combinação de parâmetros de ajuste para a modelagem computacional do problema. Neste trabalho foram adotadas algumas configurações para os

parâmetros dos modelos (direto e inverso), com o objetivo de comparar resultados com outros métodos inversos (GA e ANN).

A configuração dos parâmetros dos modelos (direto e inverso) pode ser otimizada através de técnicas de análise de sensibilidade, para levantar os parâmetros relevantes e as respectivas sensibilidades das medidas dos sensores às perturbações nos parâmetros do estado. Uma análise de sensibilidade completa, com todos os parâmetros envolvidos, fica como sugestão para trabalhos futuros.

Capítulo 5

ANÁLISE DOS RESULTADOS

Neste capítulo são apresentados os resultados obtidos pelo filtro de Kalman para o problema de detecção de danos. Inicialmente será feita a análise para o modelo potencial e posteriormente para o modelo elastostático.

5.1 MODELO POTENCIAL

Primeiramente foram ajustados todos os parâmetros mencionados no capítulo anterior, com algumas exceções, onde a maioria dos parâmetros é decidida por tentativa e erro. Logo após algumas tentativas os seguintes parâmetros foram adotados.

Covariância do ruído da medida:

$$R = \left(\frac{0,01^2 \cdot I_{8 \times 8}}{9} \right) [^{\circ}C^2]$$

Covariância do ruído do processo:

$$Q = \begin{bmatrix} 0,1^2 & 0 & 0 \\ 0 & 0,1^2 & 0 \\ 0 & 0 & 0,01^2 \end{bmatrix} [cm^2]$$

Variações dos componentes do estado para o cálculo da matriz de sensibilidade:

$$\Delta_x = \Delta_y = \Delta_r = 0,002 \text{ [cm]}$$

Também, foi adotado como estado inicial:

$$x_0 = \begin{Bmatrix} 3,00 \\ 3,00 \\ 0,001 \end{Bmatrix} \text{ [cm]}$$

A matriz de covariância do ruído da medida foi definida como sendo a precisão de um termopar de 0,1 °C, multiplicada pela identidade, pois um sensor não influi em outro, logo não são correlacionados.

Já a covariância do ruído do processo, onde também não são correlacionados, fornecendo assim somente valores na diagonal principal, sendo os valores de ruído de 10% e 1% do estado, para as coordenadas e tamanho do furo, respectivamente.

Com estes parâmetros, utilizando os sensores à uma distância de 0,6 cm das bordas e uma quantidade máxima de 50 iterações, o filtro de Kalman obteve resultados de estimação do estado com erros inferiores a 0,3% para qualquer um dos três componentes do estado, dentro de uma região que é ilustrada pela figura seguinte, cuja área é de 20,56 cm² e foi obtida por tentativa e erro para cada furo identificado.

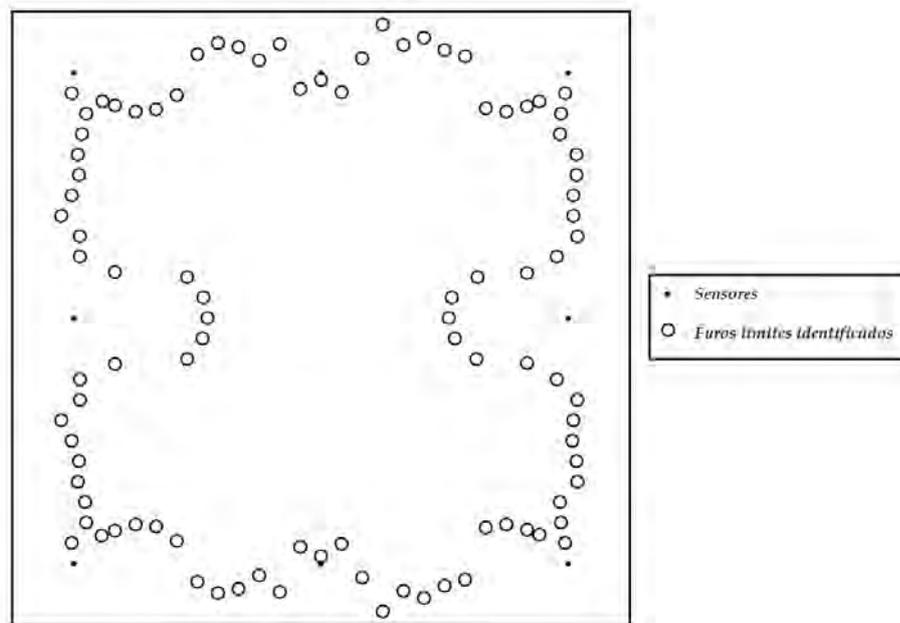


Figura 5.1 – Região limite de identificação do filtro para sensores à 0,6 cm das bordas de uma placa de 6x6 cm

Ao se posicionar o furo externamente a esta região, $0,1 \text{ mm}$ que seja o deslocamento, o erro referente à estimação do filtro salta para valores extremamente elevados. Essa configuração limite mostra que as condições de contorno, assim como os sensores interferem no resultado da estimação do filtro. Logo, alterou-se a posição dos sensores para uma distância de $0,06 \text{ cm}$ das bordas chegando à região limite ilustrada pela figura seguinte obtida da mesma maneira da Figura 5.1, com esta região possuindo uma área de $21,48 \text{ cm}^2$ e erros inferiores a $0,05\%$.

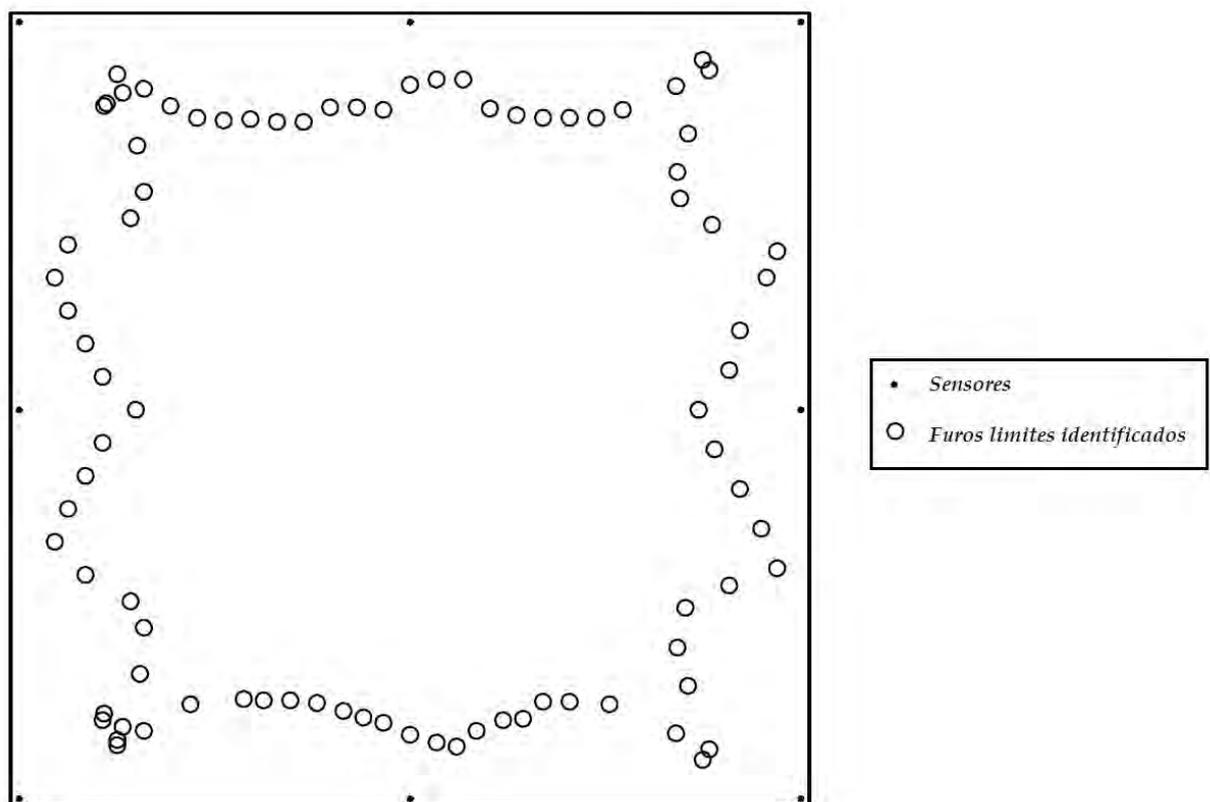


Figura 5.2 – Região limite de identificação do filtro para sensores à $0,06 \text{ cm}$ das bordas de uma placa de $6 \times 6 \text{ cm}$

Ao se analisar as Figuras 5.1 e 5.2, verifica-se que elas não são simétricas, apesar do problema ser simétrico. Esta não simetria pode ser justificada pela presença das incertezas (ruído) no modelamento tanto do estado quanto das medidas dos sensores.

Portanto, para este modelo, com estas configurações utilizadas, o posicionamento dos sensores, interfere no desempenho do filtro de Kalman. Para as duas configurações de posicionamento de sensores, houve um aumento de $4,47\%$ da área detectável pelo filtro de Kalman. Esta região interna é uma região onde o filtro localiza e identifica o furo, já a região

externa possui pontos que o filtro pode localizar e identificar, mas de forma aleatória, sendo alguns pontos isolados e outros formando sub-regiões pequenas.

Com relação a matriz de covariância do ruído do processo, a sua variação influencia na melhoria da precisão de alguns pontos, mas para outro ponto ela piora esta precisão. Além disso, a variação também interfere no número de iterações necessárias para que o filtro de Kalman encontre o resultado. Para exemplificar, considere os parâmetros utilizados anteriormente, com os sensores posicionados a 0,06 cm das bordas e uma covariância do ruído do processo Q_0 igual a:

$$Q_0 = \begin{bmatrix} 0,1^2 & 0 & 0 \\ 0 & 0,1^2 & 0 \\ 0 & 0 & 0,01^2 \end{bmatrix} [cm^2]$$

Utilizando os parâmetros fixos e variando apenas a matriz de covariância, multiplicada por fatores de um quarto, um e quatro, o filtro retornou os dados encontrados nas três tabelas a seguir, respectivamente.

Tabela 5.1 – Estimação do filtro de Kalman para um quarto da covariância Q_0

$Q = \frac{1}{4} Q_0$, Número de iterações = 29			
<i>Estado</i>	<i>Real (cm)</i>	<i>Estimado (cm)</i>	<i>Erro (%)</i>
<i>X</i>	<i>2,000000</i>	<i>2,001512</i>	<i>0,075530</i>
<i>Y</i>	<i>2,000000</i>	<i>2,001603</i>	<i>0,080110</i>
<i>R</i>	<i>0,060000</i>	<i>0,060060</i>	<i>0,099614</i>

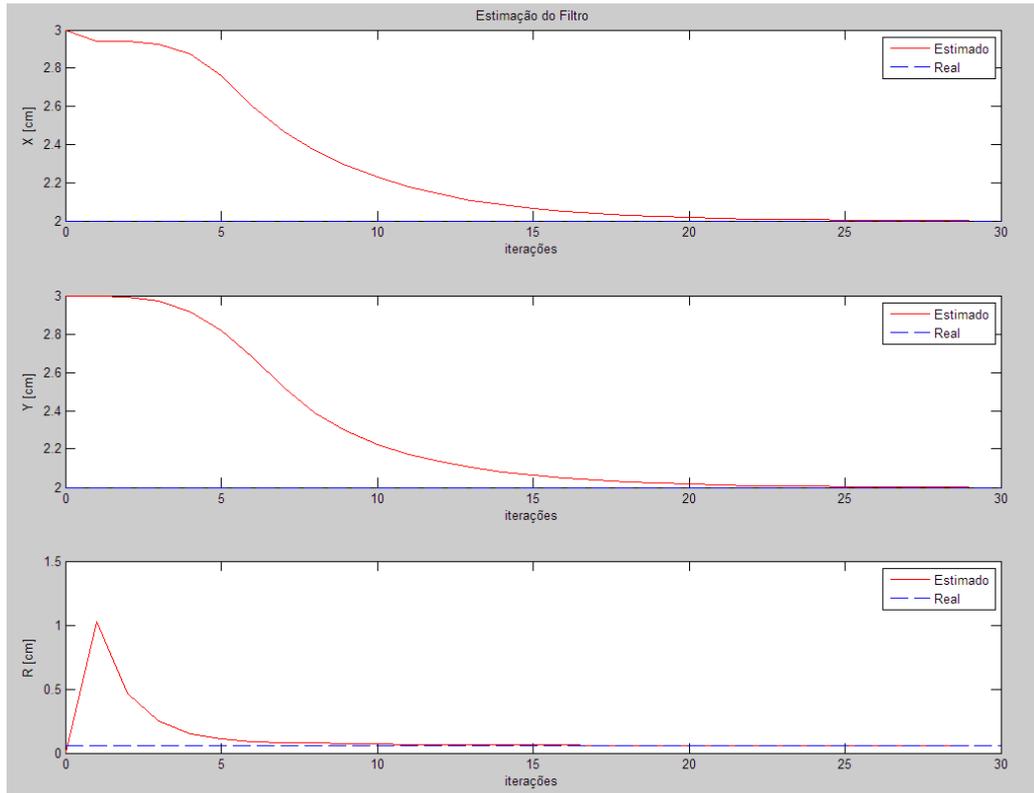


Figura 5.3 – Estimação das componentes do estado pelo filtro de Kalman com $Q = \frac{1}{4} Q_0$

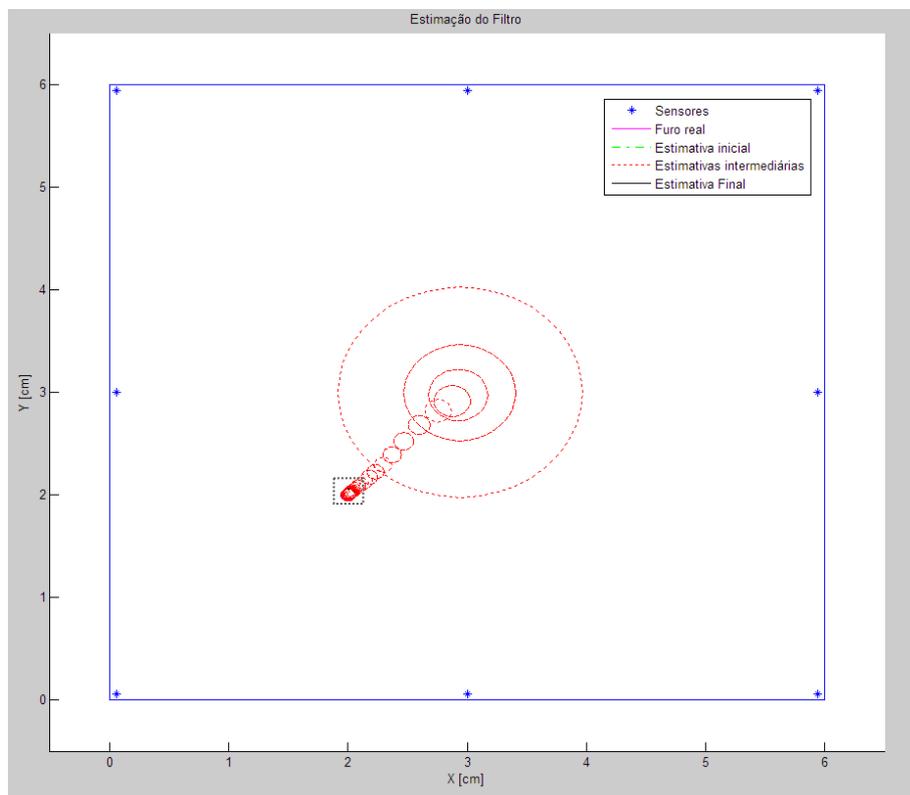


Figura 5.4 – Progresso da estimação do furo pelo filtro de Kalman com $Q = \frac{1}{4} Q_0$

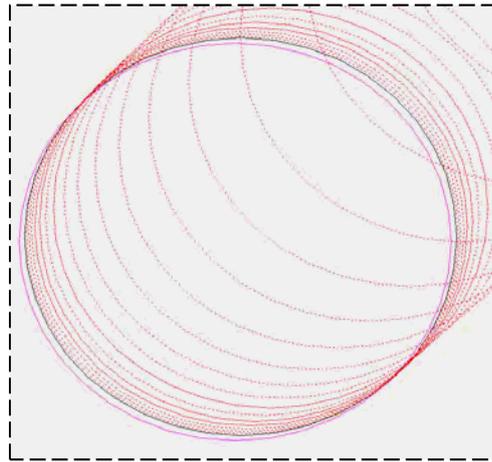


Figura 5.5 – Ampliação da estimativa final do furo pelo filtro de Kalman com $Q = \frac{1}{4} Q_0$

Tabela 5.2 – Estimativa do filtro de Kalman para covariância Q_0

$Q = Q_0$, Número de iterações = 19			
<i>Estado</i>	<i>Real (cm)</i>	<i>Estimado (cm)</i>	<i>Erro (%)</i>
<i>X</i>	<i>2,000000</i>	<i>2,000430</i>	<i>0,021515</i>
<i>Y</i>	<i>2,000000</i>	<i>2,000459</i>	<i>0,022933</i>
<i>R</i>	<i>0,060000</i>	<i>0,060017</i>	<i>0,028428</i>

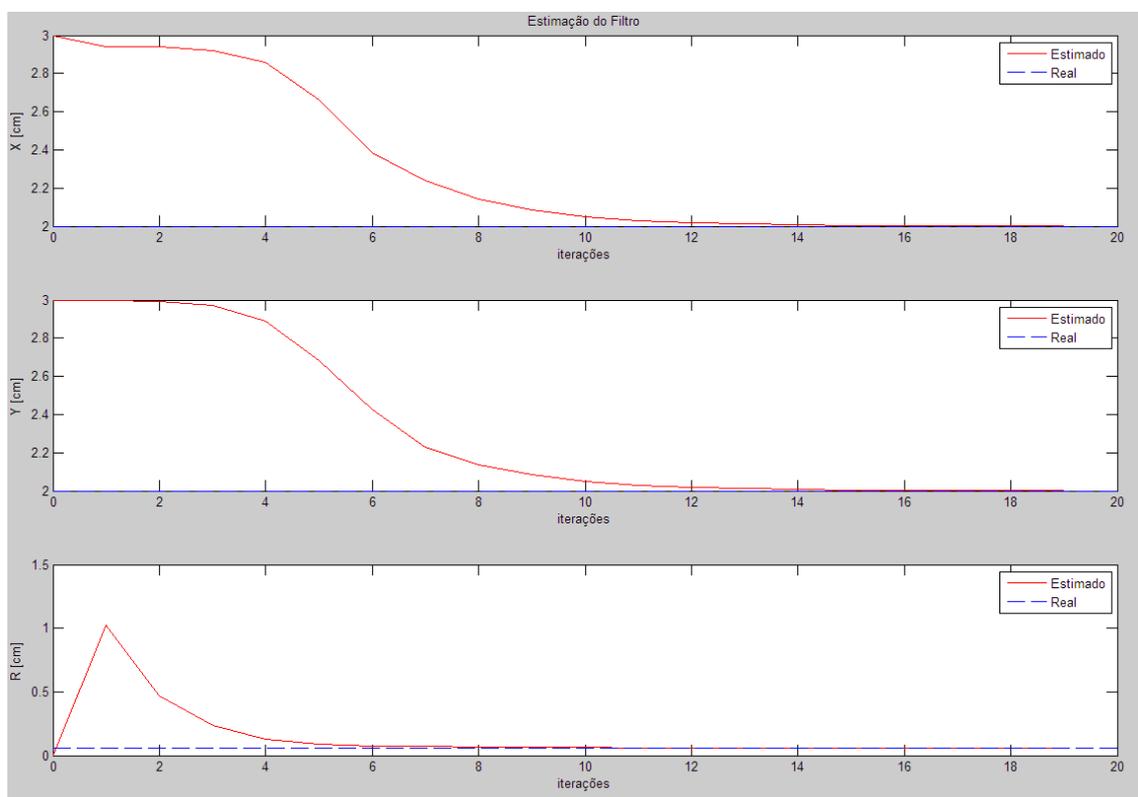


Figura 5.6 – Estimativa das componentes do estado pelo filtro de Kalman com $Q = Q_0$

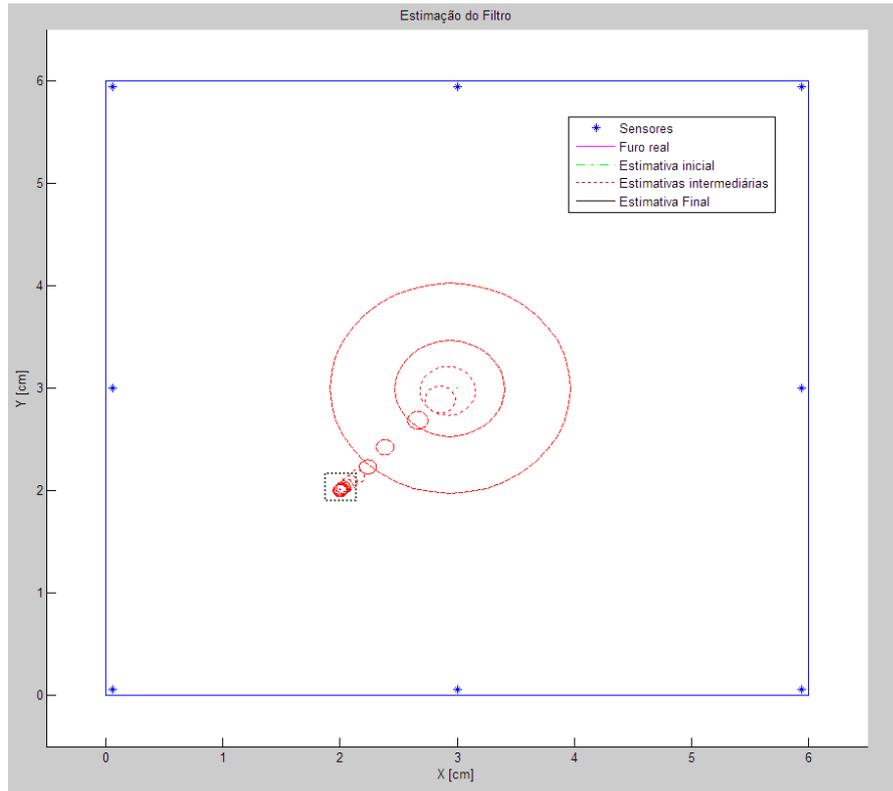


Figura 5.7 – Progresso da estimação do furo pelo filtro de Kalman com $Q = Q_0$

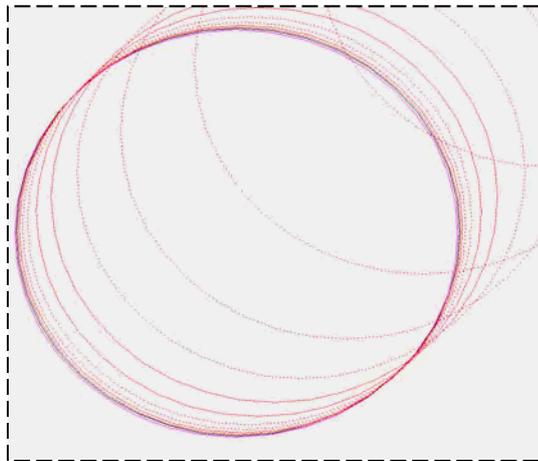
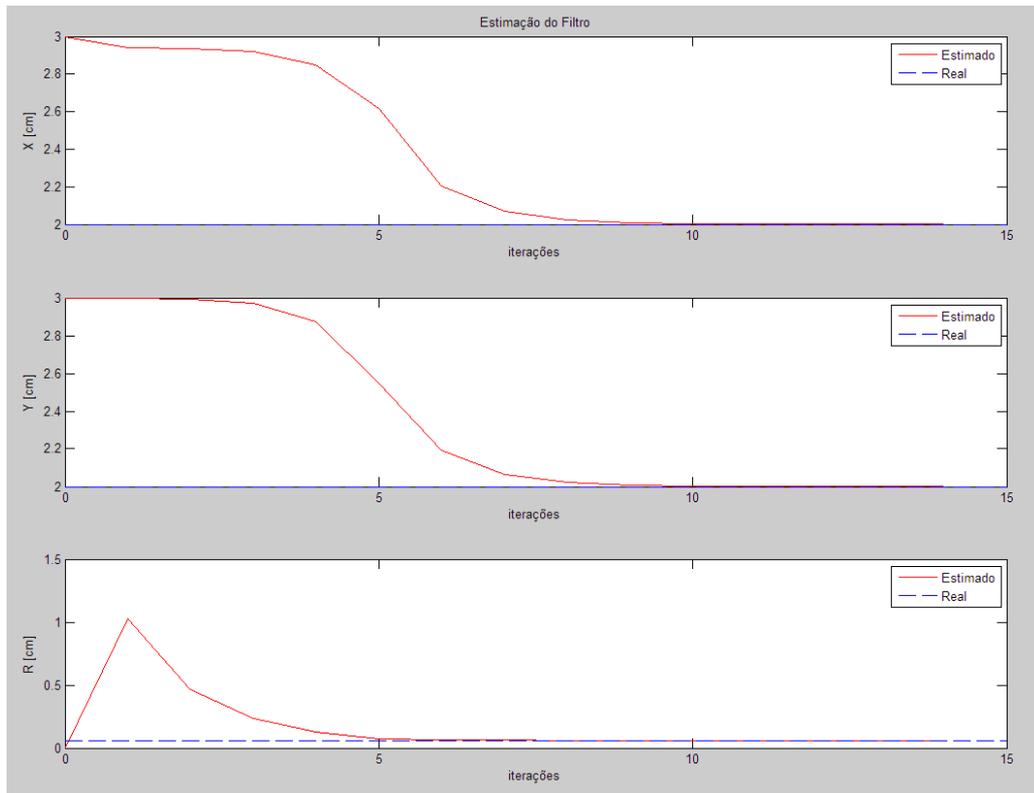


Figura 5.8 – Ampliação da estimação final do furo pelo filtro de Kalman com $Q = Q_0$

Tabela 5.3 – Estimação do filtro de Kalman para a covariância de $4Q_0$

$Q = 4Q_0$, Número de iterações = 14			
<i>Estado</i>	<i>Real (cm)</i>	<i>Estimado (cm)</i>	<i>Erro (%)</i>
<i>X</i>	<i>2,000000</i>	<i>2,000050</i>	<i>0,002518</i>
<i>Y</i>	<i>2,000000</i>	<i>2,000054</i>	<i>0,002690</i>
<i>R</i>	<i>0,060000</i>	<i>0,060002</i>	<i>0,003330</i>

Figura 5.9 – Estimação das componentes do estado pelo filtro de Kalman com $Q = 4Q_0$

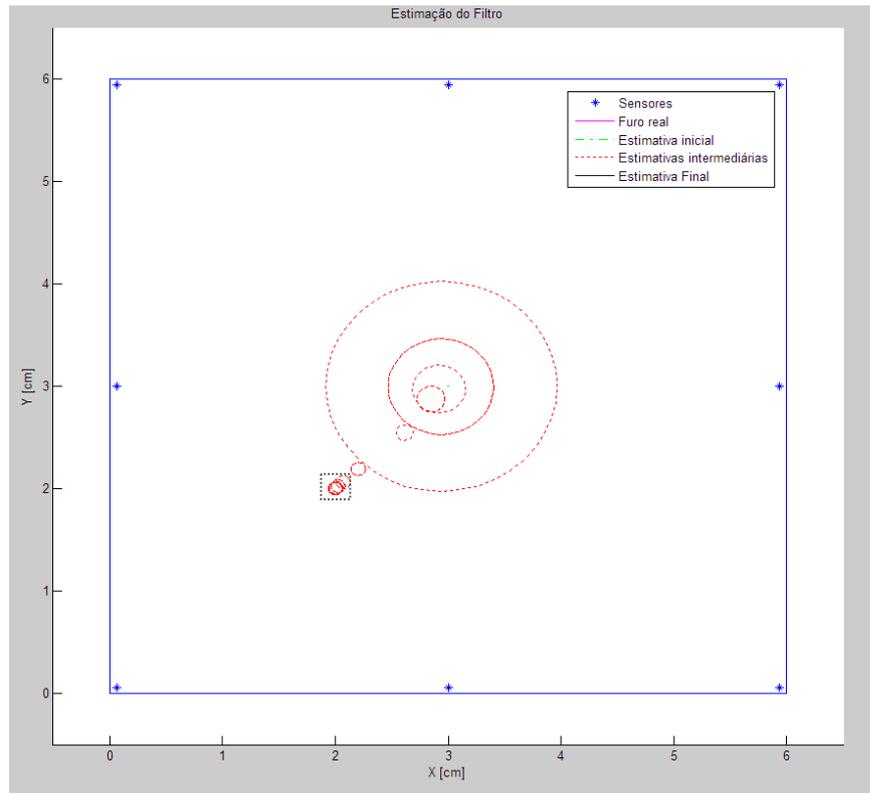


Figura 5.10 – Progresso da estimação do furo pelo filtro de Kalman com $Q = 4Q_0$

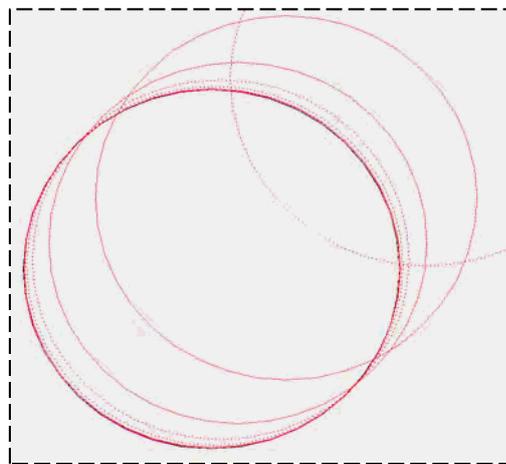


Figura 5.11 – Ampliação da estimação final do furo pelo filtro de Kalman com $Q = 4Q_0$

Com estes três casos apenas, pode-se chegar a uma conclusão errada de que quanto maior for o fator multiplicativo, mais rápida e mais precisa será a estimação do filtro. Pois, como foi dito anteriormente, esse aumento na covariância pode favorecer alguns estados e ao mesmo tempo prejudicar outros, como por exemplo, o estado:

$$x = \begin{Bmatrix} 5,25 \\ 4,40 \\ 0,06 \end{Bmatrix} [cm]$$

Que é um estado limite identificado pelo filtro, que esta presente na Figura 5.2.

Tabela 5.4 – Estimação do filtro de Kalman para covariância Q_0 e estado $x = \{5,25; 4,40; 0,06\}^T [cm]$

$Q = Q_0$, Número de iterações = 25			
<i>Estado</i>	<i>Real (cm)</i>	<i>Estimado (cm)</i>	<i>Erro (%)</i>
<i>X</i>	<i>5,250000</i>	<i>5,250375</i>	<i>0,007141</i>
<i>Y</i>	<i>4,400000</i>	<i>4,399445</i>	<i>0,012619</i>
<i>R</i>	<i>0,060000</i>	<i>0,060013</i>	<i>0,022348</i>

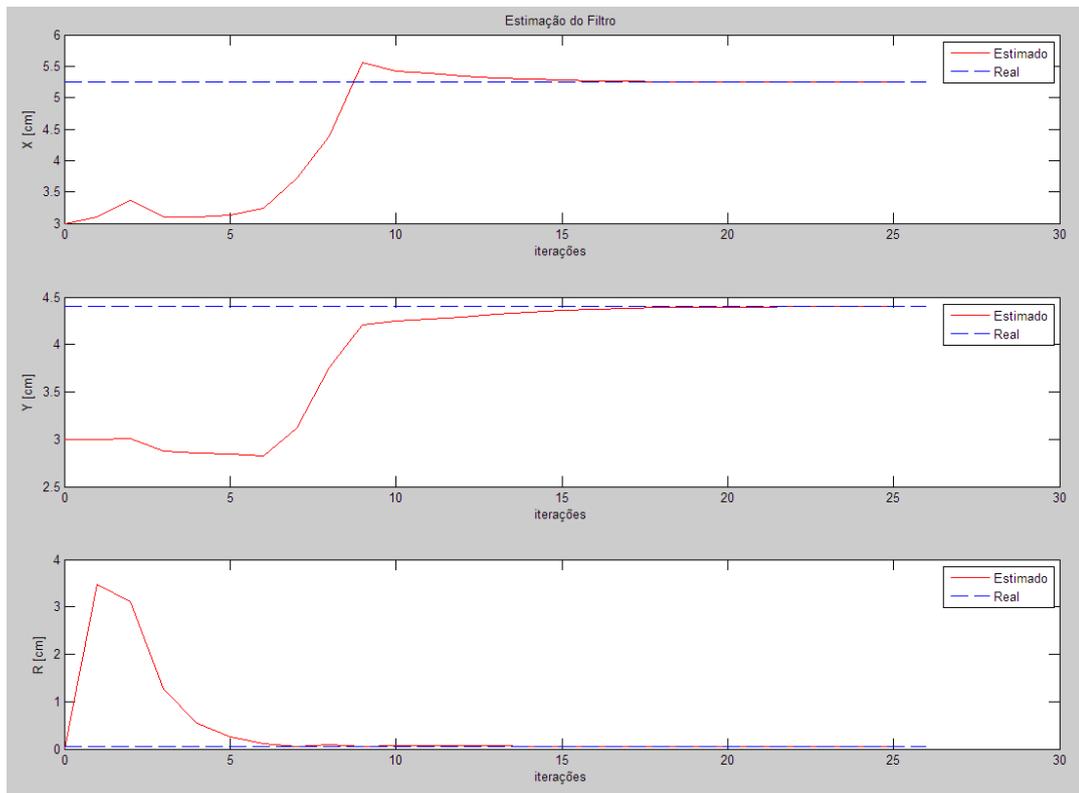


Figura 5.12 – Estimação das componentes do estado pelo filtro de Kalman com $Q = Q_0$ e $x = \{5,25; 4,40; 0,06\}^T [cm]$

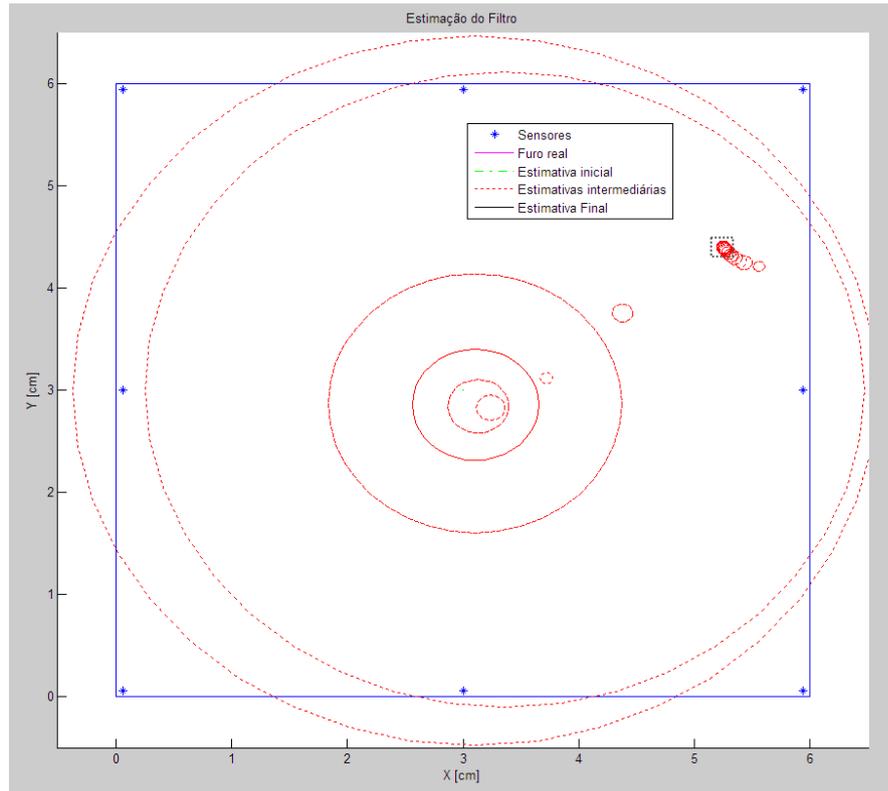


Figura 5.13 – Progresso da estimação do furo pelo filtro de Kalman utilizando $Q = Q_0$ e $x = \{5,25; 4,40; 0,06\}^T$ [cm]

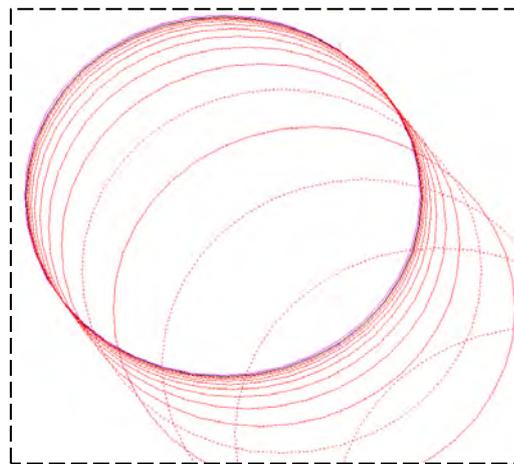


Figura 5.14 – Ampliação da estimação final do furo pelo filtro de Kalman utilizando $Q = Q_0$ e $x = \{5,25; 4,40; 0,06\}^T$ [cm]

Tabela 5.5 – Estimação do filtro de Kalman para covariância $25Q_0$ e estado $x = \{5,25; 4,40; 0,06\}^T$ [cm]

$Q = 25Q_0$, Número de iterações = 50			
<i>Estado</i>	<i>Real (cm)</i>	<i>Estimado (cm)</i>	<i>Erro (%)</i>
<i>X</i>	<i>5,250000</i>	<i>23,895929</i>	<i>78,029731</i>
<i>Y</i>	<i>4,400000</i>	<i>1,995332</i>	<i>120,514711</i>
<i>R</i>	<i>0,060000</i>	<i>9,225784</i>	<i>99,349649</i>

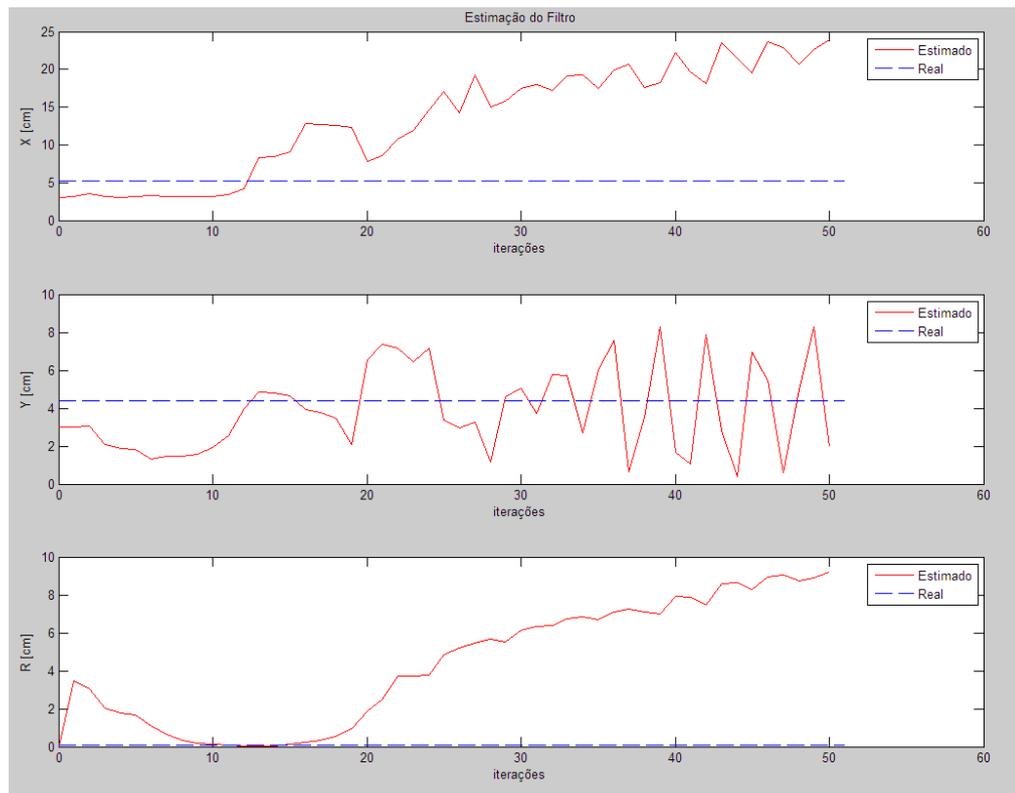


Figura 5.15 – Estimação das componentes do estado pelo filtro de Kalman com $Q = 25Q_0$ e $x = \{5,25; 4,40; 0,06\}^T$ [cm]

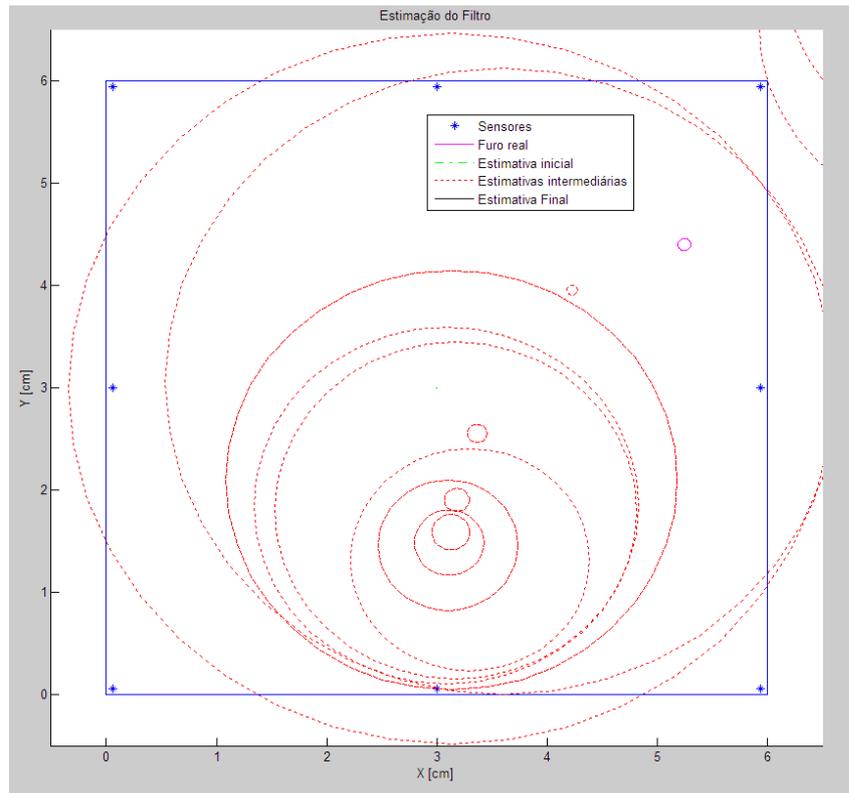


Figura 5.16 – Progresso da estimação do furo pelo filtro de Kalman com $Q = 25Q_0$ e $x = \{5,25; 4,40; 0,06\}^T [cm]$

Como foi possível observar, para um fator multiplicativo de 25 vezes, estados antes identificados pelo filtro, deixam de ser identificados devido a divergência da estimação, atingindo erros muito grandes.

Ao se comparar o resultado do filtro de Kalman, com os resultados do Algoritmo Genético e da Rede Neural Artificial, encontrado em Lopes (2010), para uma diferença de estados, anterior e atual, menor que 1×10^{-6} e número máximo de iterações do filtro de 75, o filtro de Kalman mostra uma repetitividade no resultado final da estimação, muito melhor que os outros dois métodos. Esta repetitividade é devido ao fato do filtro começar sempre de um mesmo ponto inicial e não necessitar do aprendizado da ANN e nem das mutações do GA. A tabela a seguir e as seis figuras seguintes mostram essas diferenças nos resultados.

Tabela 5.6 – Valores comparativos de estimação entre o ANN e o KF

<i>Estado</i>	<i>Real (cm)</i>	<i>ANN (9 sensores)</i>		<i>ANN (25 sensores)</i>		<i>KF* (8 sensores)</i>	
		<i>Estimado (cm)</i>	<i>Erro %</i>	<i>Estimado (cm)</i>	<i>Erro %</i>	<i>Estimado (cm)</i>	<i>Erro %</i>
X	3,00	2,9998	0,0067	3,0035	0,1165	3,0000	0**
Y	3,00	2,9973	0,0901	3,0003	0,0100	3,0000	0**
R	0,10	0,1002	0,1996	0,0992	0,8065	0,1000	0**
X	4,00	2,4224	65,1255	3,4568	15,7140	4,0000	0**
Y	2,00	0,4355	359,2423	0,5676	252,3608	2,0000	0**
R	0,10	0,9774	89,7688	0,0994	0,6036	0,1000	0**

* Estado inicial $x_0 = \{3,00; 3,00; 0,0001\}^T$ [cm]

** Valores aproximados, pois a diferença do estado real para o estado estimado é muito menor do que a tolerância de 1×10^{-6}

As Figuras 5.18, 5.19, 5.20 e 5.22, que são referentes ao KF, utilizam a tolerância de 1×10^{-6} e número máximo de iterações igual a 75 e os seguintes parâmetros:

$$x_0 = \begin{Bmatrix} 3,00 \\ 3,00 \\ 0,0001 \end{Bmatrix} [cm], \quad Q_0 = \begin{bmatrix} 0,1^2 & 0 & 0 \\ 0 & 0,1^2 & 0 \\ 0 & 0 & 0,01^2 \end{bmatrix} [cm^2], \quad R = \left(\frac{0,01^2 \cdot I_{8 \times 8}}{9} \right) [^\circ C^2] \quad e$$

$$\Delta_x = \Delta_y = \Delta_r = 0,002 [cm]$$

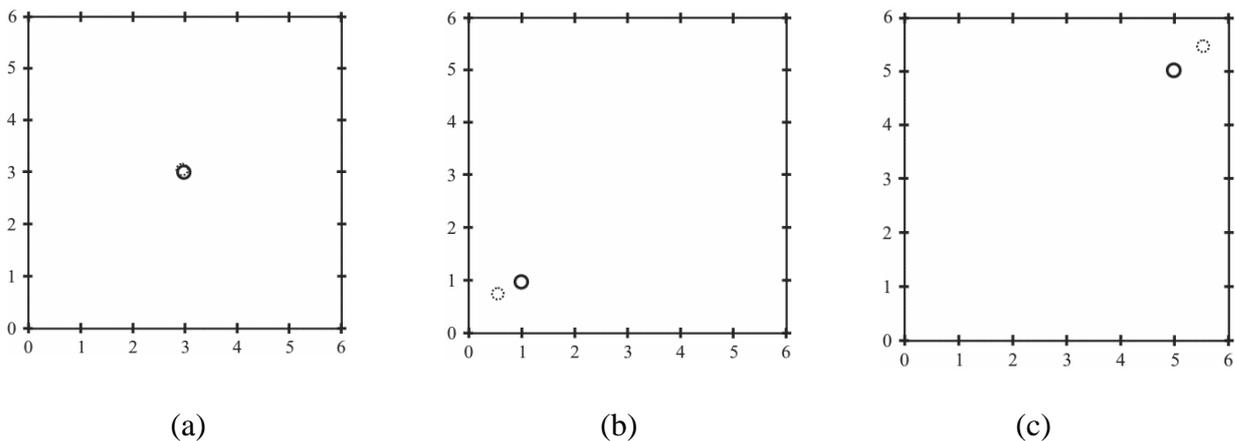


Figura 5.17 – Problema potencial: resultados da ANN para 5 sensores para um furo nas posições: (a) (3,0;3,0) cm; (b) (1,0;1,0) cm; e, (c) (5,0;5,0) cm. [Retirado de Lopes (2010)]

A Figura 5.17 ilustra apenas o resultado final das iterações do ANN e não como o método “caminha” até a estimação final.

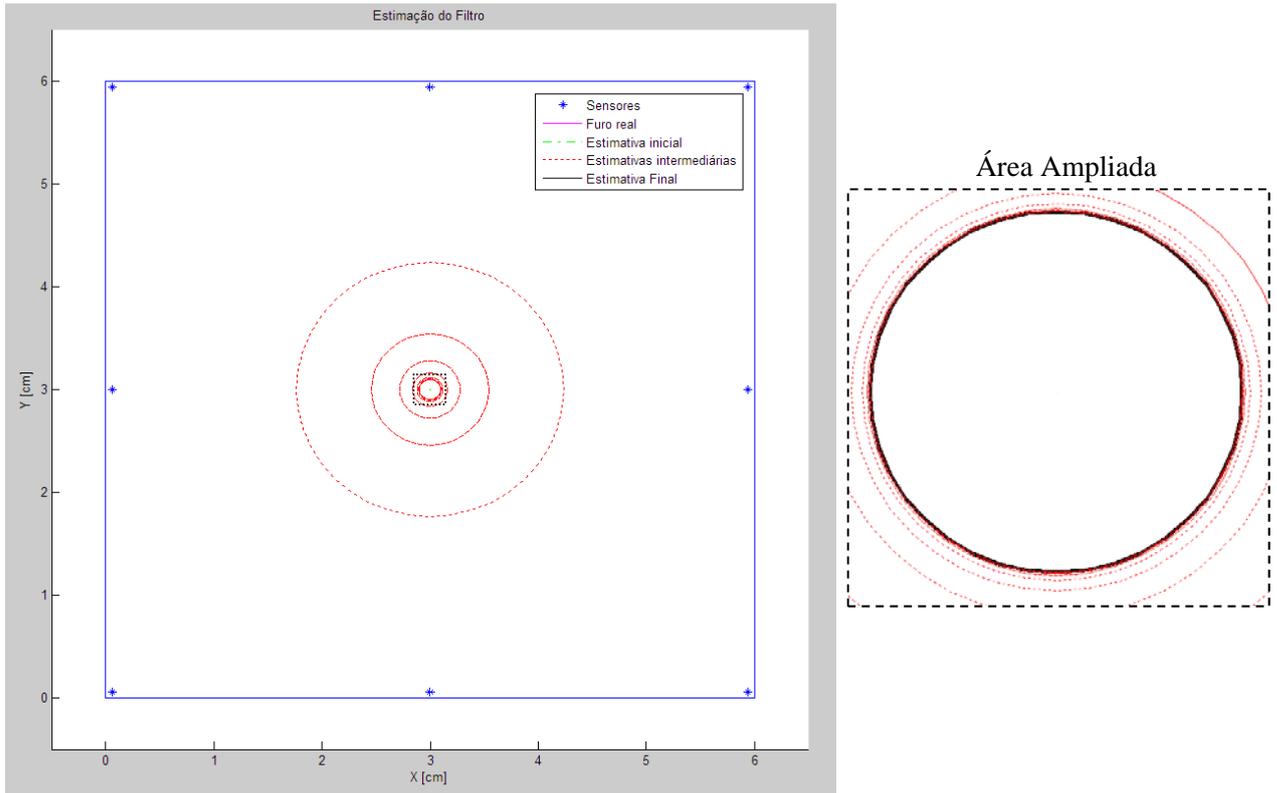


Figura 5.18 – Resultado do filtro de Kalman para o furo na posição (3,00;3,00) *cm*

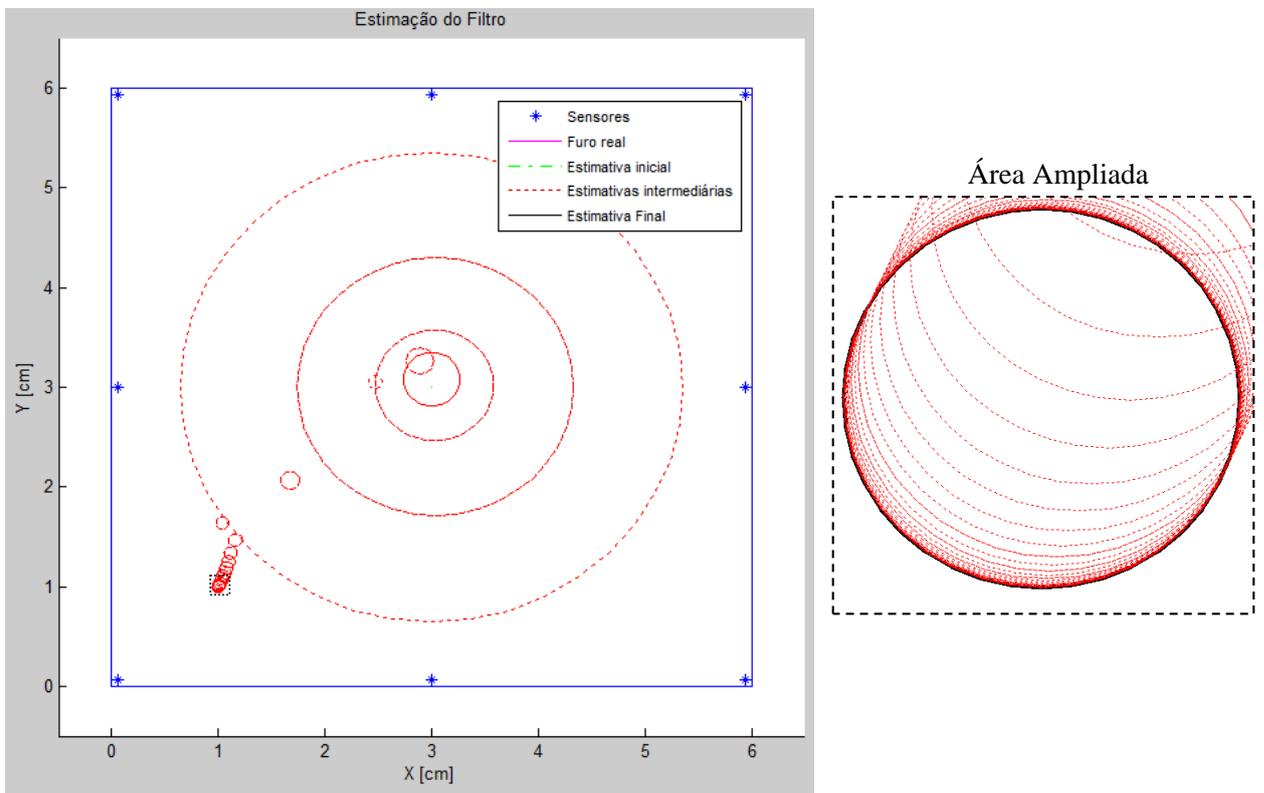


Figura 5.19 – Resultado do filtro de Kalman para o furo na posição (1,00;1,00) *cm*

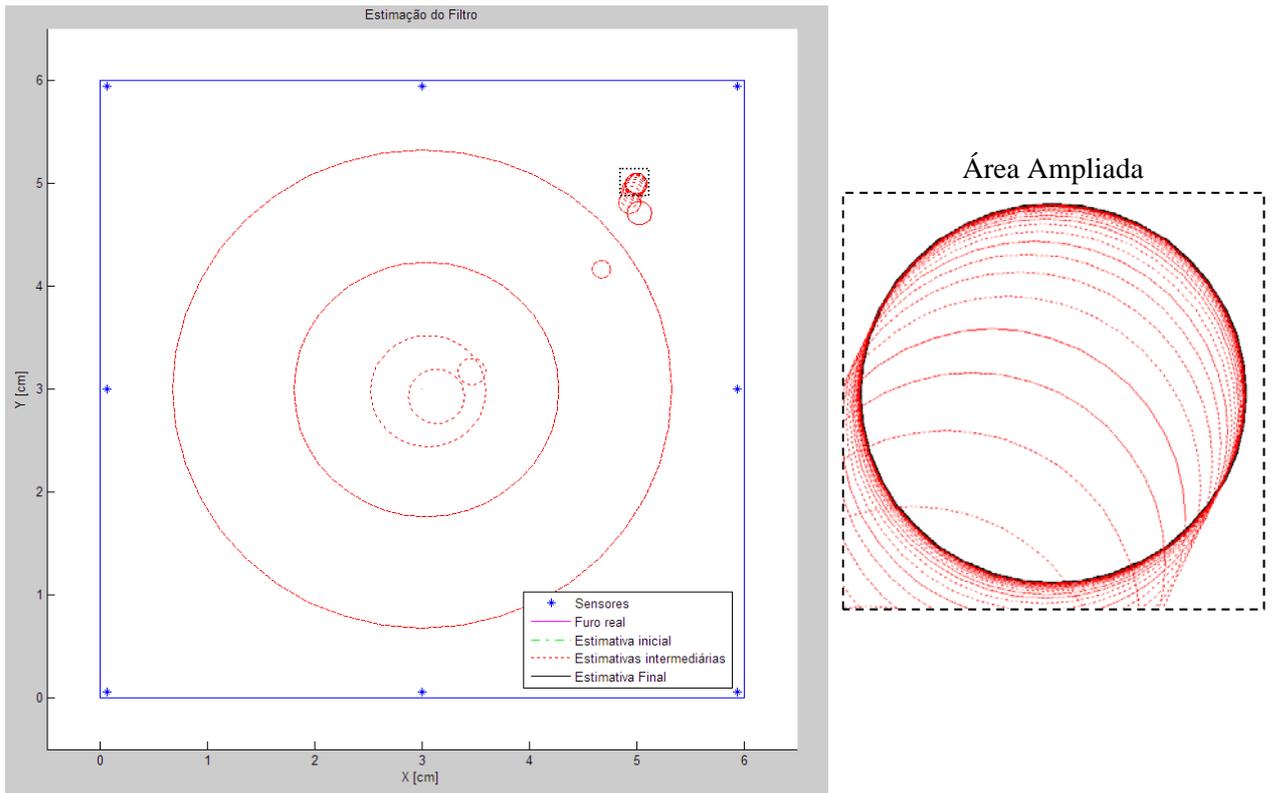


Figura 5.20 – Resultado do filtro de Kalman para o furo na posição $(5,00;5,00)$ cm

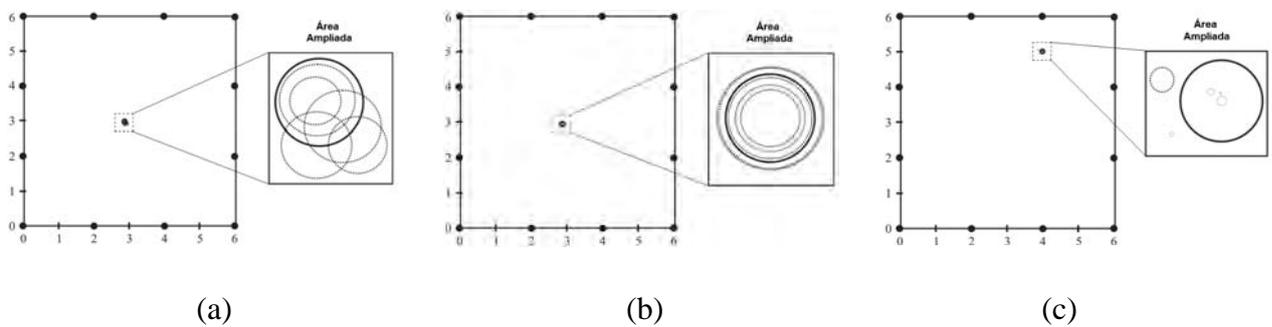


Figura 5.21 – Resultados do GA, para o furo “real” e simulado para o potencial: (a) para um furo central com elitismo igual a 2 na posição $(3,00;3,00)$ cm; (b) para um furo central com elitismo igual a 10 na posição $(3,00;3,00)$ cm; (c) para um furo na posição $(4,0;5,0)$ cm. As áreas ampliadas mostram a região do furo em detalhes. [Retirado de LOPES (2010)]

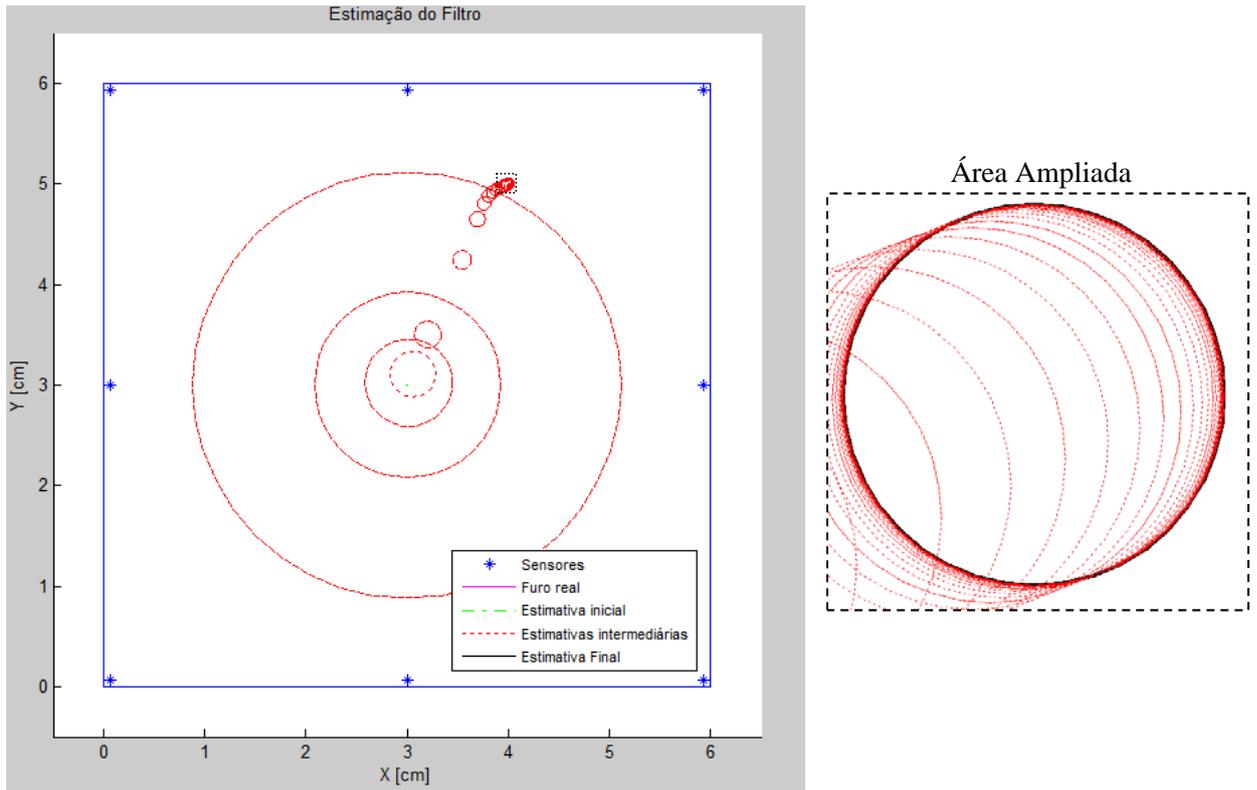


Figura 5.22 – Resultado do filtro de Kalman para o furo na posição $(4,00;5,00)$ cm

Logo, considerando estes resultados do modelo potencial, o filtro de Kalman se mostrou comparativamente eficiente em relação aos métodos de algoritmo genético e de redes neurais artificiais. Mas deve-se ressaltar a necessidade de se ajustar os parâmetros iniciais para que a estimação do filtro seja eficiente, pois como foi mostrado existem diversos fatores que influenciam no resultado do filtro como, por exemplo, o posicionamento dos sensores, a covariância do ruído do processo e outros não mencionados no trabalho.

5.2 MODELO ELASTOSTÁTICO

Assim como no modelo potencial, tentou-se encontrar uma região de sensibilidade do filtro para diversos parâmetros de ajuste como mostrado pelas Figuras 5.1 e 5.2. Mas ao contrário do que foi encontrado para o outro modelo, não foi possível a identificação de uma região de confiabilidade do filtro. Como possíveis causas para esta não identificação pode-se citar o uso de elementos constantes e a discretização grosseira do contorno. O uso de elementos de contorno de ordem mais alta, como elementos lineares ou quadráticos e o

refinamento da malha do contorno ficam como sugestão para trabalhos futuros. A inclusão destes parâmetros de discretização do modelo direto em uma futura análise de sensibilidade para identificação de parâmetros, pode contribuir para a otimização do desempenho do filtro.

Logo, como alternativa para este modelo, é proposto um deslocamento dos sensores, de forma que eles se aproximem do centro da placa, realizando assim algumas leituras e detectando uma região provável de presença do furo.

Para exemplificar a proposta, as figuras e tabelas a seguir, mostram os resultados do filtro para algumas posições de sensores, utilizando como parâmetros:

$$x_0 = \begin{Bmatrix} 3,00 \\ 3,00 \\ 0,01 \end{Bmatrix} [cm], Q = \left(\begin{bmatrix} 0,01^2 & 0 & 0 \\ 0 & 0,03^2 & 0 \\ 0 & 0 & 0,001^2 \end{bmatrix} \cdot 10^6 \right) [cm^2],$$

$$R = (\sigma_{z_k}^2 \cdot I_{8 \times 8}) [MPa^2], \Delta_x = \Delta_y = 0,0035 [cm] \text{ e } \Delta_r = 0,002 [cm]$$

Também foi considerado uma tolerância para conversão de 1×10^{-3} e número máximo de iterações igual a 75. Na covariância do ruído da medida é utilizado o desvio padrão da medida (σ_{z_k}) vezes a matriz identidade ($I_{8 \times 8}$).

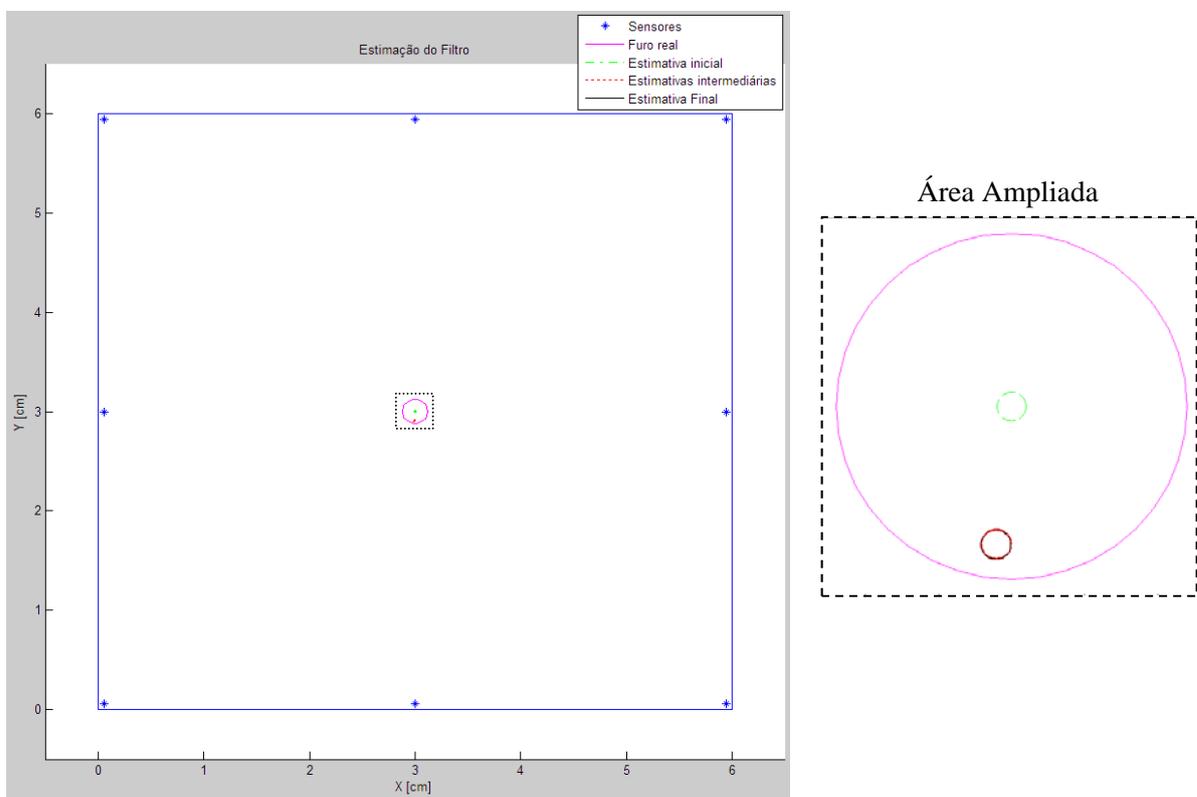


Figura 5.23 – Resultado do filtro de Kalman para o furo na posição (3,00;3,00) cm e sensores à 0,06 cm das bordas

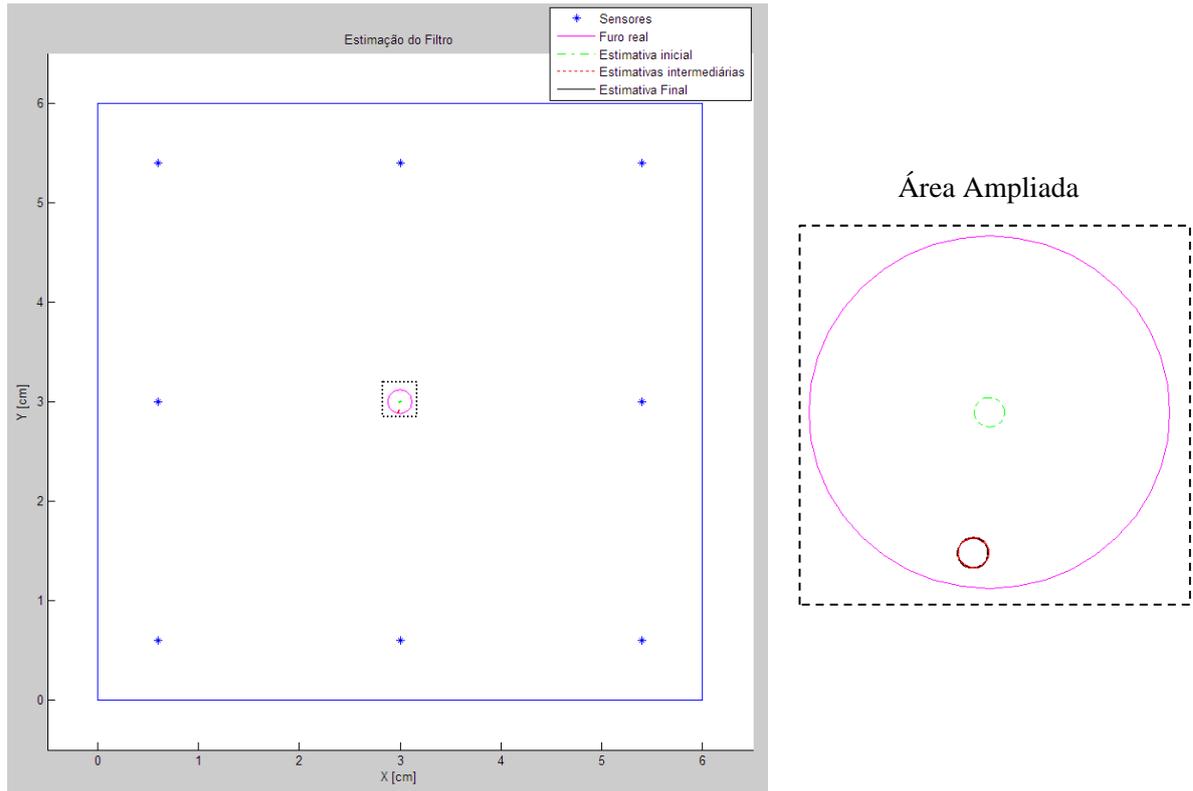


Figura 5.24 – Resultado do filtro de Kalman para o furo na posição $(3,00;3,00)$ *cm* e sensores à $0,6$ *cm* das bordas

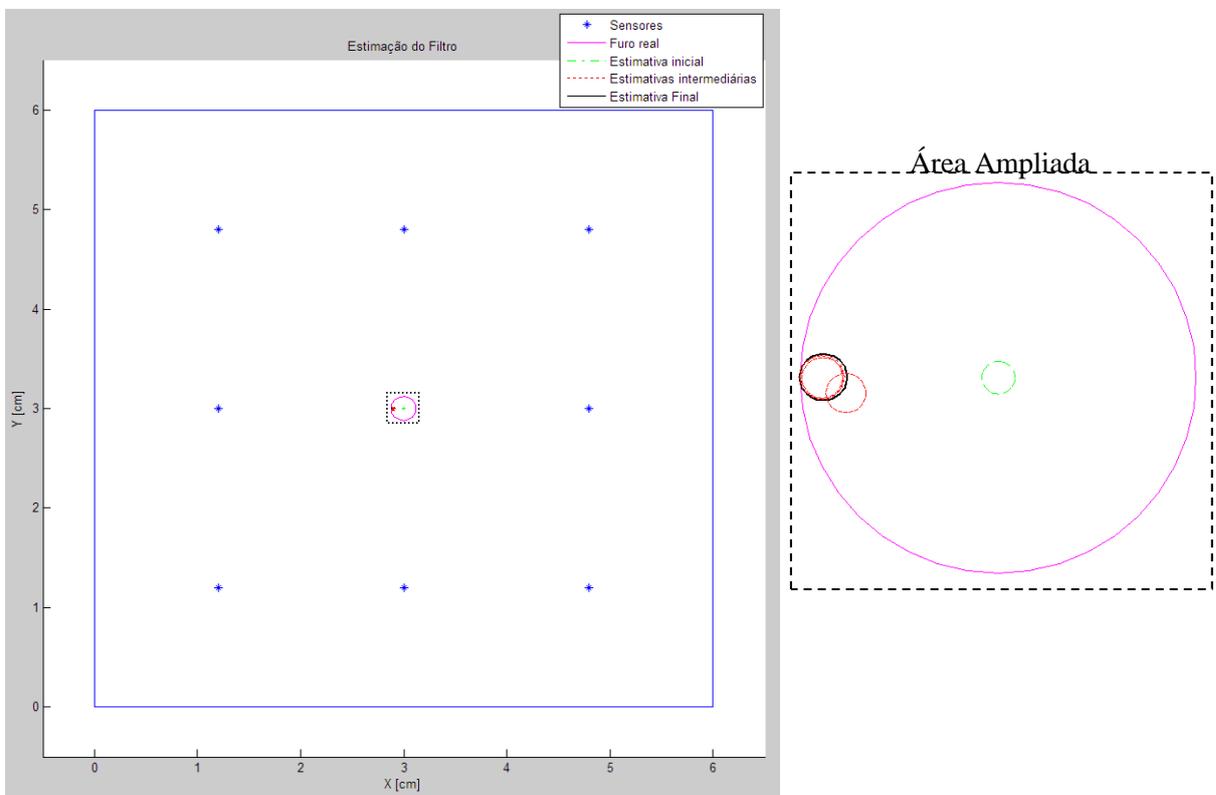


Figura 5.25 – Resultado do filtro de Kalman para o furo na posição $(3,00;3,00)$ *cm* e sensores à $1,2$ *cm* das bordas

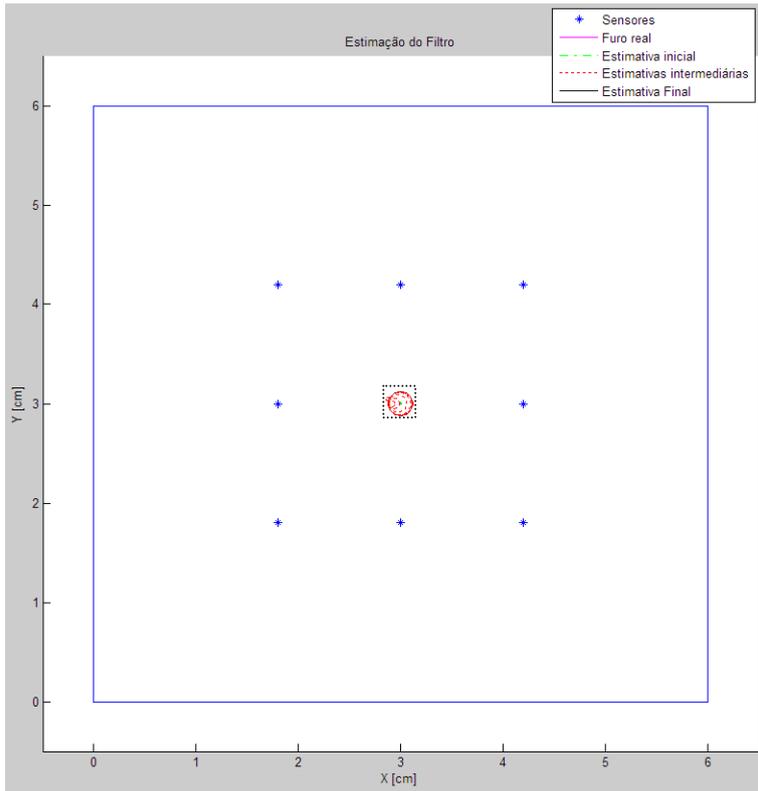


Figura 5.26 – Resultado do filtro de Kalman para o furo na posição (3,00;3,00) *cm* e sensores à 1,8 *cm* das bordas

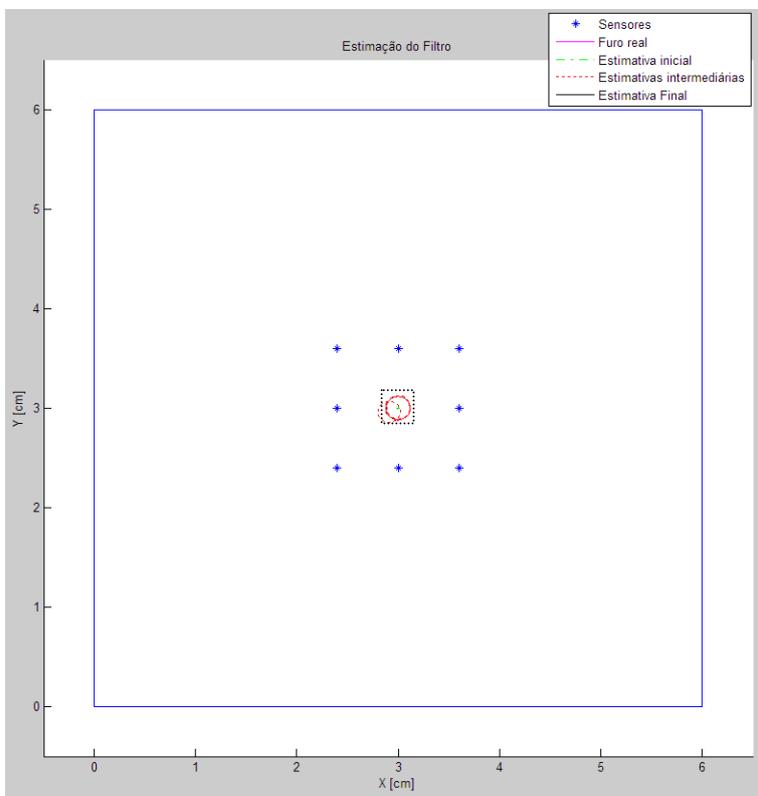
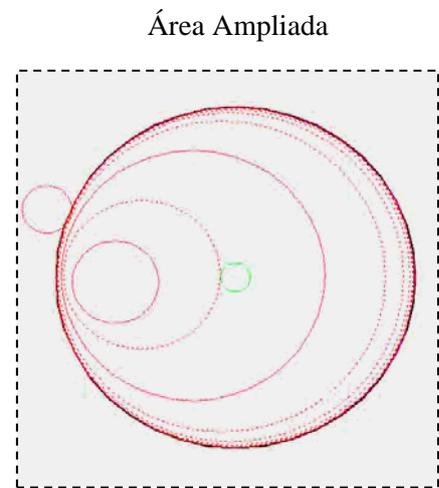


Figura 5.27 – Resultado do filtro de Kalman para o furo na posição (3,00;3,00) *cm* e sensores à 2,4 *cm* das bordas

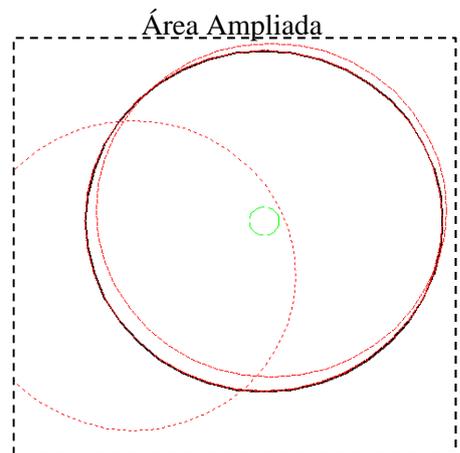


Tabela 5.7 – Resultados do filtro de Kalman para o furo na posição (3,00;3,00) *cm* para algumas posições de sensores

<i>Distância dos sensores em relação às bordas (cm)</i>		0,06		0,6		1,2		1,8		2,4	
<i>Área do furo em relação à área coberta pelos sensores (%)</i>		0,00131		0,00196		0,00349		0,00785		0,03142	
<i>Iterações Necessárias</i>		4		4		5		11		6	
<i>Estado</i>	<i>Real (cm)</i>	<i>Estimado (cm)</i>	<i>Erro (%)</i>								
<i>X</i>	3,00	2,989	0,367	2,989	0,367	2,893	3,567	3,000	0	3,000	0
<i>Y</i>	3,00	2,904	3,2	2,904	3,2	3,000	0	3,000	0	3,000	0
<i>Raio</i>	0,12	0,010	91,67	0,010	91,67	0,014	88,33	0,120	0	0,120	0

Agora, se for utilizada uma distribuição *t-student* para um intervalo de confiança de 95% para quatro, três e dois graus de liberdade, que são referentes aos cinco, quatro e três últimos resultados da Tabela 5.7 respectivamente, pois o grau de liberdade é definido como sendo o número de amostras menos um, pode se obter os seguintes intervalos para as componentes do estado, presentes na tabela seguinte.

Tabela 5.8 – Intervalos de confiança de 95% para o furo (3,00;3,00;0,12) *cm*

<i>Grau de Liberdade</i>	4	3	2
<i>Estado</i>	-	-	-
<i>X</i>	2,974 ± 0,056	2,971 ± 0,082	2,964 ± 0,154
<i>Y</i>	2,962 ± 0,065	2,976 ± 0,076	3,000 ± 0,000
<i>R</i>	0,055 ± 0,074	0,066 ± 0,100	0,085 ± 0,152

Deve-se chamar a atenção, na Tabela 5.8, para o fato de que o erro presente no intervalo de confiança para o raio do furo estar na mesma ordem de grandeza ou até maior que o valor médio. Isto é devido a presença dos valores de 0,010 e 0,014 *cm* que fazem com que a

variância das amostras seja elevada, além de diminuir o valor médio, ocasionando este nível de grandeza no erro.

Para o caso do furo central, essa proposta convergiu para o estado real logo é preciso avaliá-la para outros casos.

Utilizando os mesmos parâmetros, para um furo na posição $(1,50;2,00)$ cm, obtemos os resultados mostrados pelas cinco figuras e duas tabelas seguintes.

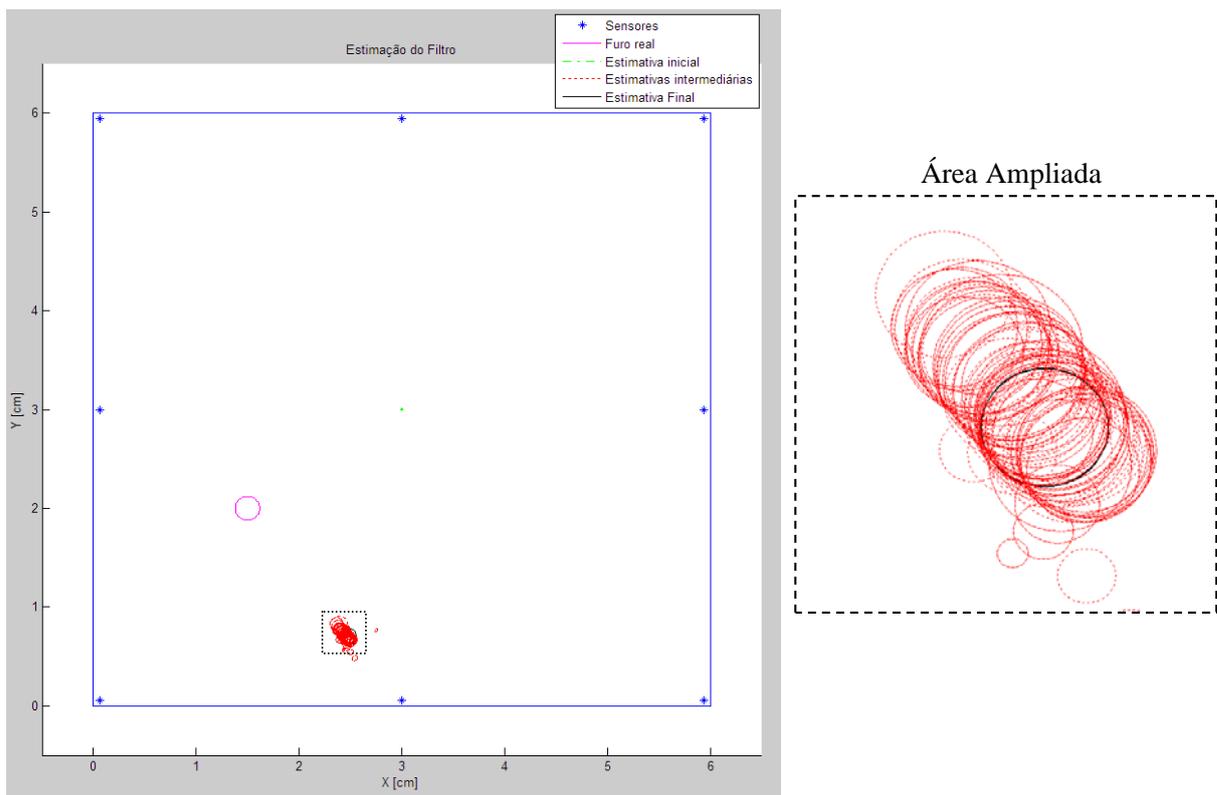


Figura 5.28 – Resultado do filtro de Kalman para o furo na posição $(1,50;2,00)$ cm e sensores à $0,06$ cm das bordas

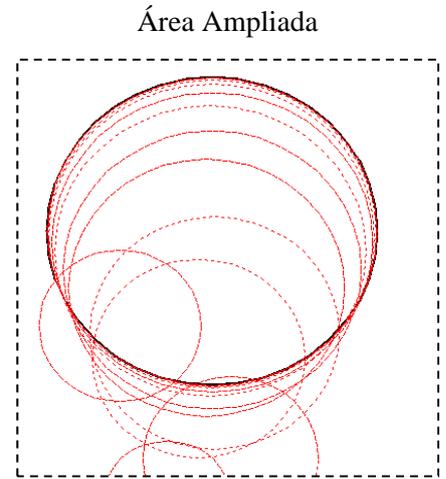
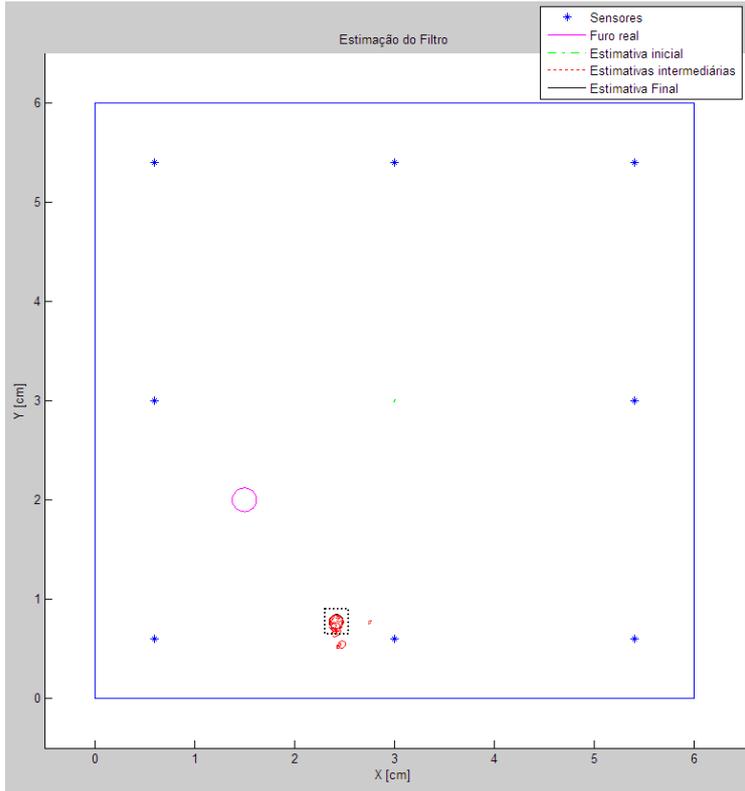


Figura 5.29 – Resultado do filtro de Kalman para o furo na posição (1,50;2,00) *cm* e sensores à 0,6 *cm* das bordas

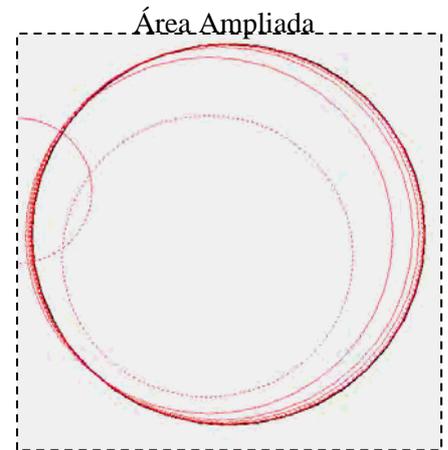
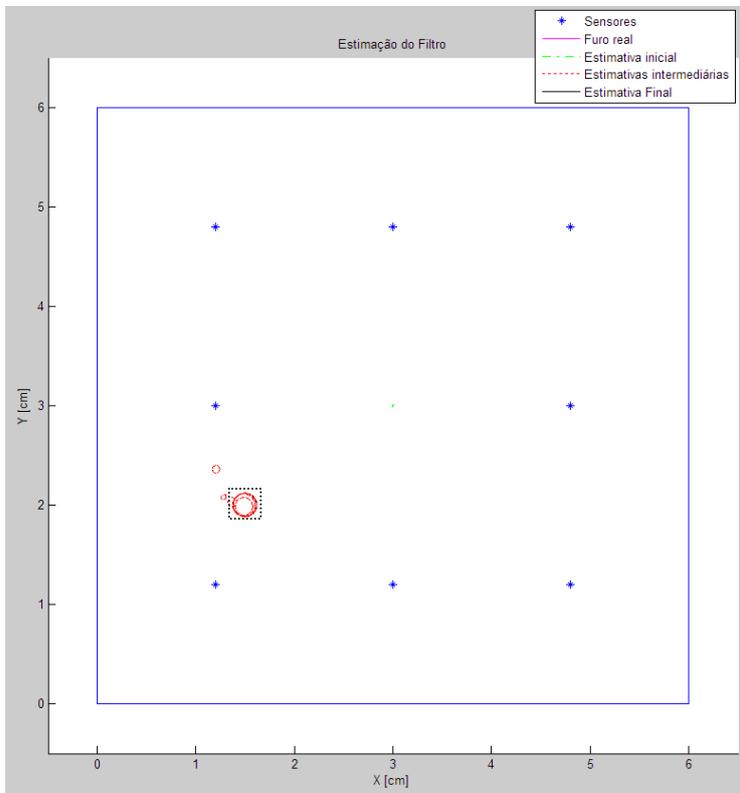


Figura 5.30 – Resultado do filtro de Kalman para o furo na posição (1,50;2,00) *cm* e sensores à 1,2 *cm* das bordas

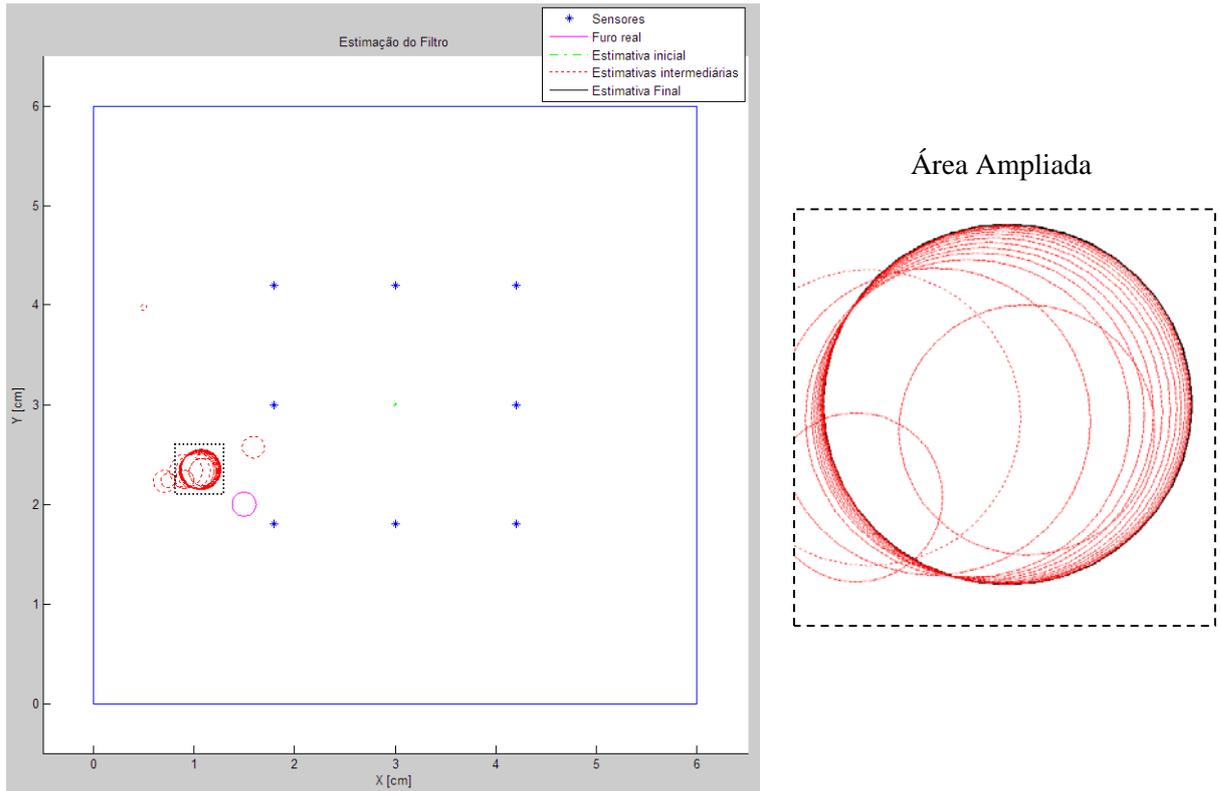


Figura 5.31 – Resultado do filtro de Kalman para o furo na posição (1,50;2,00) *cm* e sensores à 1,8 *cm* das bordas

Ao analisar a Figura 5.31, pode-se verificar que a estimativa encontrada para os sensores posicionados à 1,8 *cm* das bordas teve um erro maior do que a estimativa para os sensores posicionados à 1,2 *cm* das bordas.

Ao se utilizar uma variação menor da posição dos sensores, de forma que estes fiquem posicionados à 1,5 *cm* de distância das bordas, o filtro volta a estimar um resultado correto para o estado, o que é exemplificado pela figura seguinte.

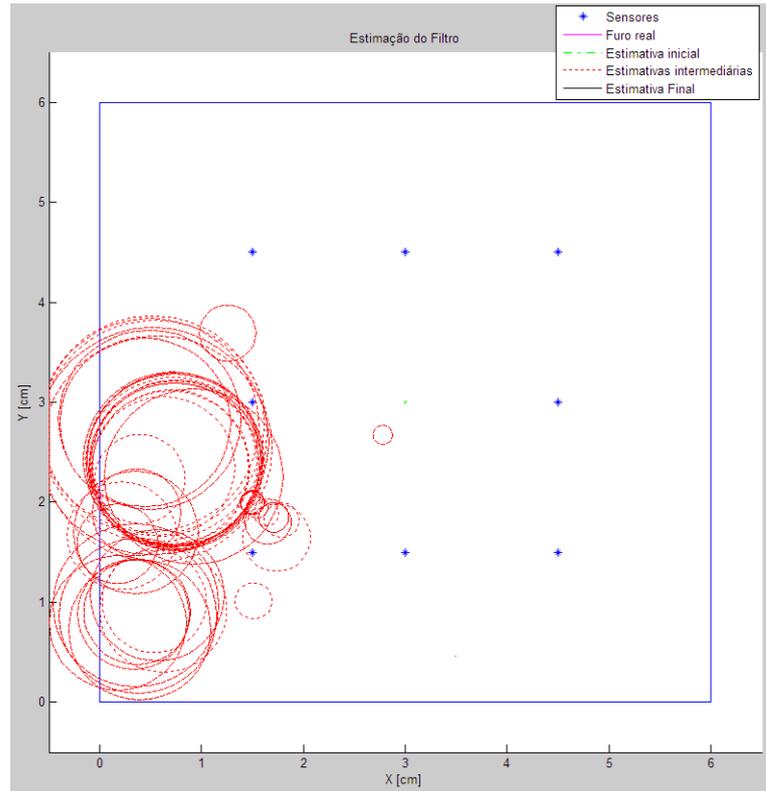


Figura 5.32 – Resultado do filtro de Kalman para o furo na posição (1,50;2,00) cm e sensores à 1,5 cm das bordas

Tabela 5.9 – Resultados do filtro de Kalman para o furo na posição (1,50;2,00) cm para algumas posições de sensores

<i>Distância dos sensores em relação às bordas (cm)</i>		<i>0,06</i>		<i>0,6</i>		<i>1,2</i>		<i>1,5</i>		<i>1,8</i>	
<i>Área do furo em relação à área coberta pelos sensores (%)</i>		<i>0,00131</i>		<i>0,00196</i>		<i>0,00349</i>		<i>0,00503</i>		<i>0,00785</i>	
<i>Iterações Necessárias</i>		<i>75</i>		<i>18</i>		<i>11</i>		<i>52</i>		<i>21</i>	
<i>Estado</i>	<i>Real (cm)</i>	<i>Estimado (cm)</i>	<i>Erro (%)</i>								
<i>X</i>	<i>1,50</i>	<i>2,479</i>	<i>65,27</i>	<i>1,500</i>	<i>0</i>	<i>1,500</i>	<i>0</i>	<i>1,500</i>	<i>0</i>	<i>1,071</i>	<i>28,6</i>
<i>Y</i>	<i>2,00</i>	<i>0,703</i>	<i>64,85</i>	<i>0,774</i>	<i>61,3</i>	<i>2,000</i>	<i>0</i>	<i>2,000</i>	<i>0</i>	<i>2,350</i>	<i>17,5</i>
<i>Raio</i>	<i>0,12</i>	<i>0,081</i>	<i>32,5</i>	<i>0,070</i>	<i>41,67</i>	<i>0,120</i>	<i>0</i>	<i>0,120</i>	<i>0</i>	<i>0,199</i>	<i>65,83</i>

Agora, aplicando novamente a distribuição *t-student* para um intervalo de confiança de 95% mas agora apenas para três e dois graus de liberdade, descartando o resultado para os sensores à 1,8 cm das bordas, tem-se os intervalos presentes na tabela seguinte.

Tabela 5.10 – Intervalos de confiança de 95% para o furo (1,50;2,00;0,12) cm

<i>Grau de Liberdade</i>	3	2
<i>Estado</i>	-	-
<i>X</i>	1,745 ± 0,779	1,500 ± 0,000
<i>Y</i>	1,369 ± 1,769	1,591 ± 1,759
<i>R</i>	0,098 ± 0,042	0,103 ± 0,072

Na Tabela 5.10, também se deve chamar a atenção para o erro presente nos intervalos de confiança estar na mesma ou maior ordem de grandeza do valor médio, que por sua vez é devido à variância alta das estimativas do filtro. Logo para que este intervalo de confiança tenha valores mais aceitáveis, é necessária a redução da variação da distância de posicionamento dos sensores.

Durante a execução deste modelo, foi testada a influência da covariância sobre a estimativa e notou-se que a mesma possui uma relação com o posicionamento dos sensores, além da posição real do furo. Logo, foi testado para o furo não central (1,50;2,00;0,12) cm com sensores posicionados à 1,2 e 1,3 cm (esta última devido estar entre duas posições de sensores resultaram em estimativas corretas).

As Figuras 5.30 e 5.33 mostram o resultado do filtro para os sensores posicionados à 1,2 e 1,3 cm, respectivamente, para uma matriz de covariância do ruído do processo igual a:

$$Q = \left(\begin{bmatrix} 0,01^2 & 0 & 0 \\ 0 & 0,03^2 & 0 \\ 0 & 0 & 0,001^2 \end{bmatrix} \cdot 10^6 \right) [cm^2]$$

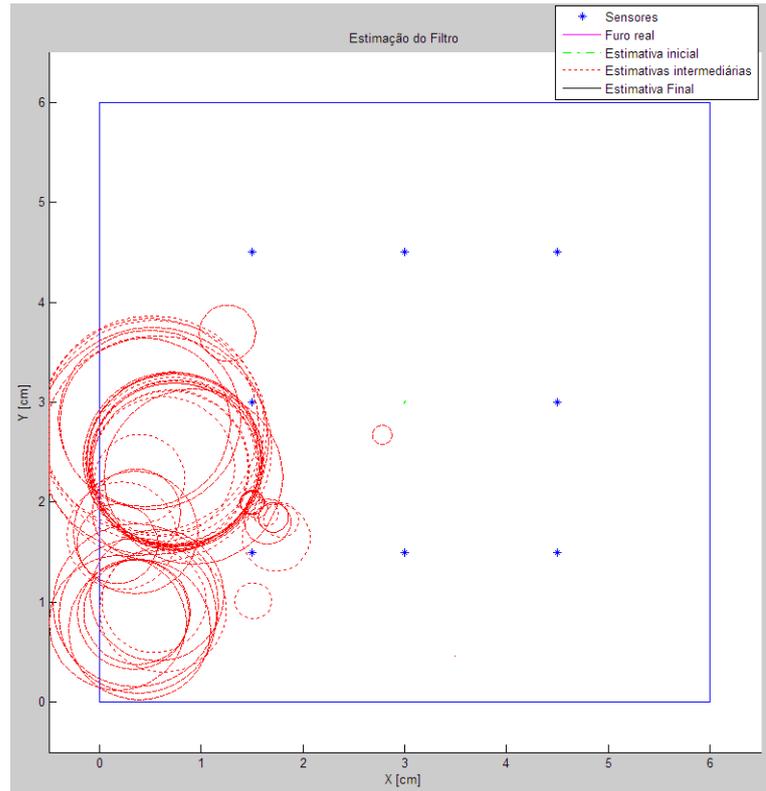


Figura 5.33 – Resultado do filtro de Kalman para o furo na posição (1,50;2,00) cm e sensores à 1,3 cm das bordas

Tabela 5.11 – Resultados do filtro de Kalman para o furo na posição (1,50;2,00) cm para uma covariância Q igual a inicial

<i>Distância dos sensores em relação às bordas (cm)</i>		<i>1,2</i>		<i>1,3</i>	
<i>Iterações Necessárias</i>		<i>11</i>		<i>9</i>	
<i>Estado</i>	<i>Real (cm)</i>	<i>Estimado (cm)</i>	<i>Erro (%)</i>	<i>Estimado (cm)</i>	<i>Erro (%)</i>
<i>X</i>	<i>1,50</i>	<i>1,49988</i>	<i>0,00792</i>	<i>1,49994</i>	<i>0,00399</i>
<i>Y</i>	<i>2,00</i>	<i>2,00003</i>	<i>0,00169</i>	<i>1,99998</i>	<i>0,00122</i>
<i>Raio</i>	<i>0,12</i>	<i>0,11997</i>	<i>0,02899</i>	<i>0,11470</i>	<i>0,11470</i>

Agora se a matriz de covariância Q for substituída por uma nova matriz de covariância igual a:

$$Q = \left(\begin{bmatrix} 0,01^2 & 0 & 0 \\ 0 & 0,03^2 & 0 \\ 0 & 0 & 0,01^2 \end{bmatrix} \cdot 10^6 \right) [cm^2]$$

Obtém-se os resultados mostrados pelas duas figuras seguintes.

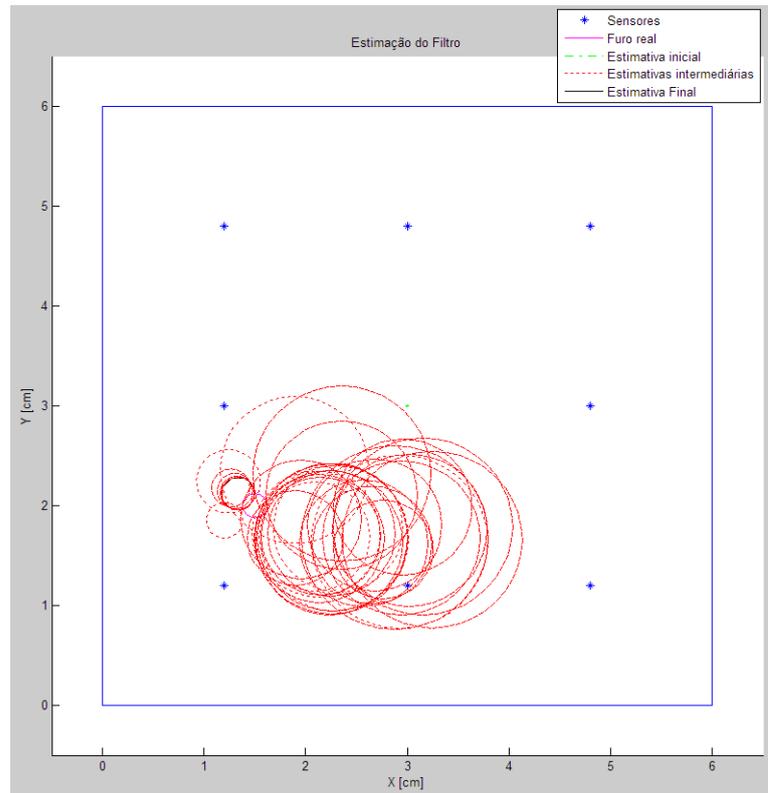


Figura 5.34 – Resultado do filtro de Kalman para o furo na posição $(1,50;2,00)$ cm e sensores à 1,2 cm das bordas com a covariância alterada

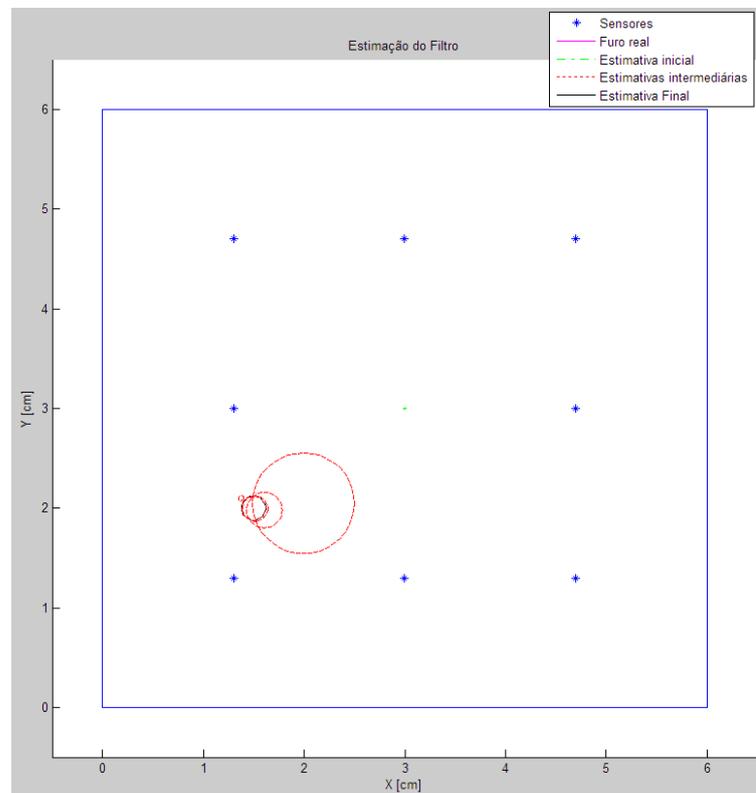


Figura 5.35 – Resultado do filtro de Kalman para o furo na posição $(1,50;2,00)$ cm e sensores à 1,3 cm das bordas com a covariância alterada

Tabela 5.12 – Resultados do filtro de Kalman para o furo na posição (1,50;2,00) cm para uma nova covariância Q

<i>Distância dos sensores em relação às bordas (cm)</i>		<i>1,2</i>		<i>1,3</i>	
<i>Iterações Necessárias</i>		<i>36</i>		<i>8</i>	
<i>Estado</i>	<i>Real (cm)</i>	<i>Estimado (cm)</i>	<i>Erro (%)</i>	<i>Estimado (cm)</i>	<i>Erro (%)</i>
<i>X</i>	<i>1,50</i>	<i>1,33609</i>	<i>10,92734</i>	<i>1,50000</i>	<i>0</i>
<i>Y</i>	<i>2,00</i>	<i>2,11913</i>	<i>5,95646</i>	<i>2,00000</i>	<i>0</i>
<i>Raio</i>	<i>0,12</i>	<i>0,16125</i>	<i>34,37674</i>	<i>0,12000</i>	<i>0</i>

Com estes resultados é possível observar que a variação de um único elemento interfere no resultado obtido pelo filtro. Ao se comparar os resultados presentes nas Tabelas 5.11 e 5.12, verifica-se que a alteração da covariância Q prejudicou o resultado para os sensores posicionados a 1,2 cm, mas melhorou para 1,3 cm. Como no momento este parâmetro é fixo e não se tem uma relação para que o mesmo se atualize automaticamente com o problema, o filtro não possui uma robustez desejada.

A figura seguinte traz o resultado encontrado por Lopes (2010) para o GA, para três posições de furos (3,00;3,00) cm, (2,00;2,00) cm e (5,00;3,00) cm.

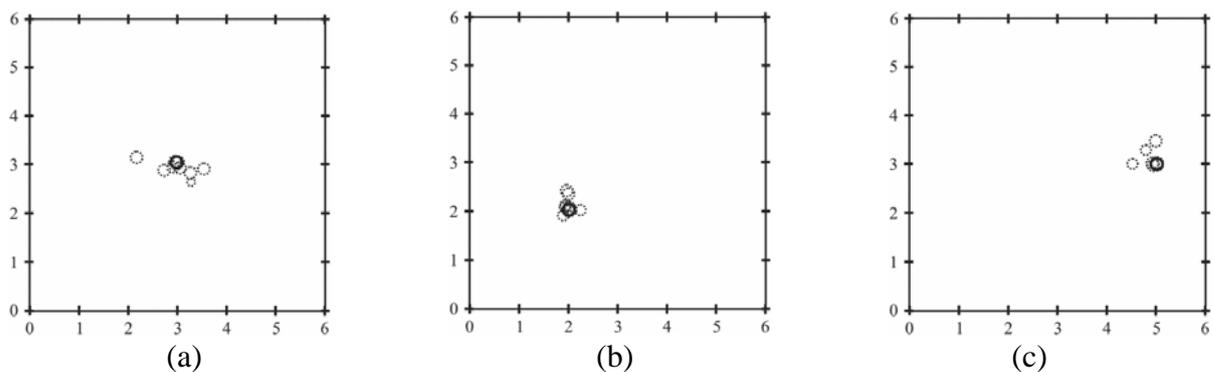


Figura 5.36 – Furo real e simulado para a tensão média: (a) para um furo central; (b) para um furo localizado em (2,0;2,0) cm; (c) para um furo localizado em (5,0;3,0) cm. [Retirado de Lopes (2010)]

Já a figura seguinte traz o resultado encontrado por Lopes (2010) para a ANN, para três posições de furos (3,00;3,00) cm, (1,00;1,00) cm e (5,00;5,00) cm.

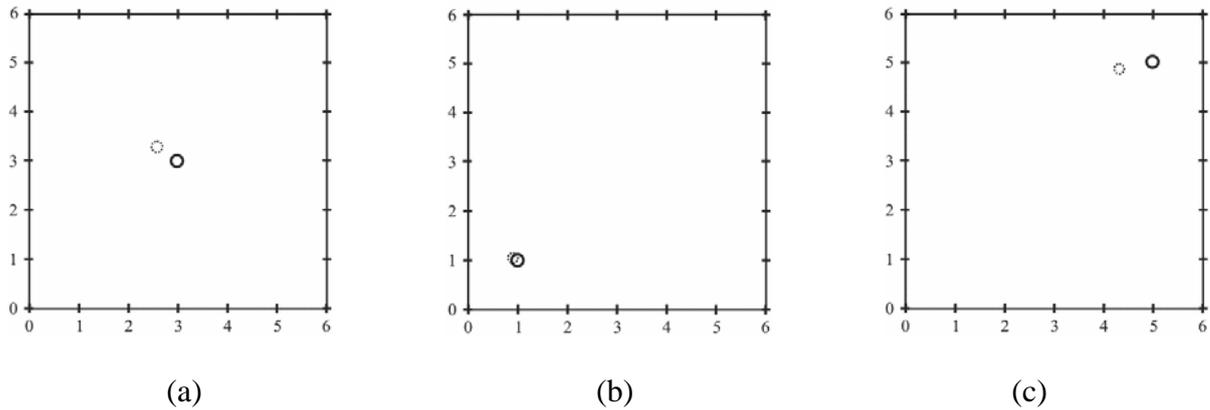


Figura 5.37 – Problema elastostático: resultados obtidos da ANN para um furo nas posições: (a) (3,0;3,0) *cm*; (b) (1,0;1,0) *cm*; e, (c) (5,0;5,0) *cm*. [Retirado de Lopes (2010)]

Como comparação, para os mesmos furos, o FK com a metodologia proposta de aproximação, onde a posição inicial dos sensores é 0,06 *cm* de distância das bordas, passando para 0,6 *cm* em seguida e assim variando de 0,05 *cm* até não houver uma variação significativa do resultado para três posições seguidas dos sensores. É utilizado um limite de 75 iterações e uma tolerância para conversão de 1×10^{-3} e não 1×10^{-6} como no trabalho de Lopes (2010), pois neste problema esta tolerância não interfere no resultado final, apenas no número de iterações necessárias. Também é utilizada uma distribuição *t-student* para um intervalo de confiança de 95% e com dois graus de liberdade.

Tabela 5.13 – Resultados do filtro de Kalman para intervalos de confiança de 95%

<i>Furo Real (cm)</i>	<i>(3,00;3,00;0,12)</i>
<i>Estado</i>	-
<i>X</i>	$2,998 \pm 0,001$
<i>Y</i>	$3,000 \pm 0,000$
<i>R</i>	$0,118 \pm 0,001$

Note que na Tabela 5.13, não estão presentes os intervalos de confiança para os furos (1,00;1,00;0,12), (2,00;2,00;0,12), (5,00;3,00;0,12) e (5,00;5,00;0,12) *cm*. Esta ausência é devida ao fato do método não ter convertido as iterações para um furo, mas a não convergência não significa que não exista posições para os sensores onde o filtro de Kalman encontre um resultado correto. As duas figuras seguintes exemplificam esta afirmação, para o furo (2,00;2,00;0,12) *cm*, onde estão presentes os resultados encontrados pelo filtro com a variação da posição dos sensores.

Logo, apesar do método não acusar uma convergência de resultados, de maneira consecutiva, ao se analisar todos os resultados encontrados pode-se identificar uma região com certa concentração dos resultados estimados, assim como mostra a figura seguinte. Mas esta região provável de conter o furo possui uma exatidão menor do que o resultado encontrado pelo método quando o mesmo converge.

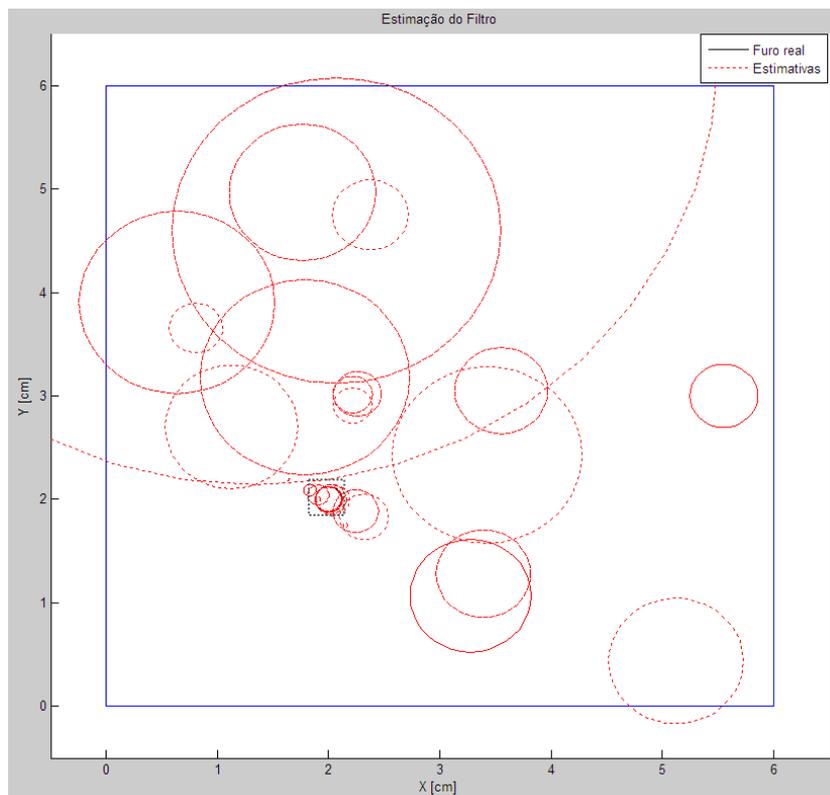


Figura 5.38 – Resultado do filtro de Kalman para o furo (2,00;2,00;0,12) cm para diversas posições de sensores

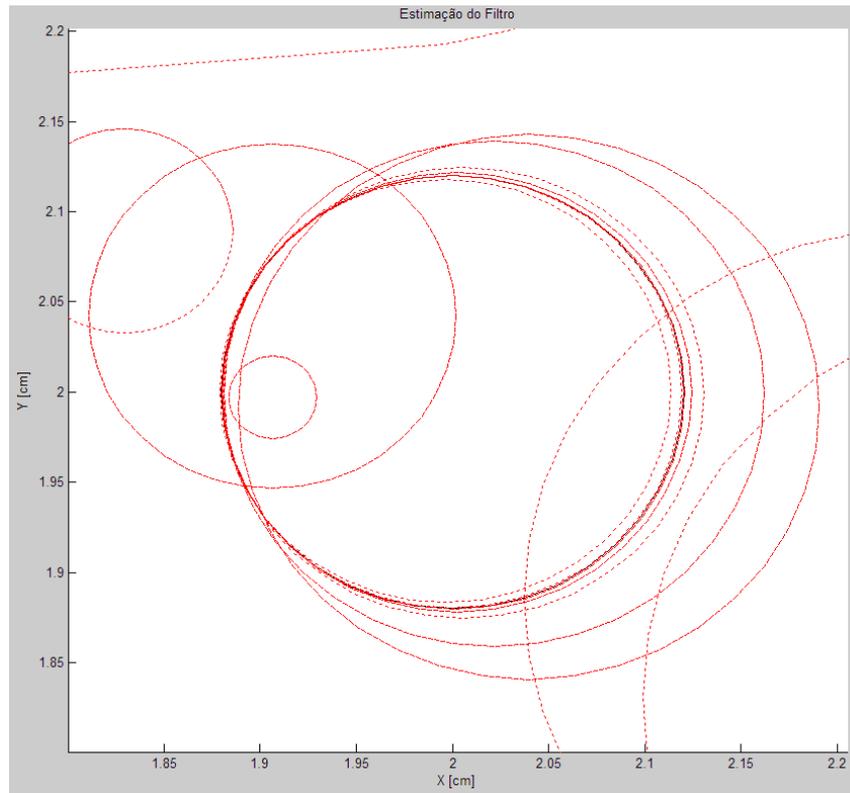


Figura 5.39 – Zoom na região do furo (2,00;2,00;0,12) cm

Logo, considerando estes resultados do modelo elastostático, o filtro de Kalman não se mostrou tão eficiente quando comparado ao seu desempenho no modelo potencial, apesar de ser possível a localização e identificação do furo. Este desempenho inferior do filtro pode ser devido a configuração do modelo direto utilizada, sendo que o uso de um BEM especializado em mecânica da fratura, ou outro tipo de elemento poderia, talvez melhorar o desempenho.

Capítulo 6

CONCLUSÃO

Neste trabalho, um problema inverso de identificação de parâmetros para a detecção de danos foi resolvido utilizando filtro de Kalman como ferramenta. Para tal, foi desenvolvido um modelo híbrido, onde o estado é descrito por um modelo linear e as medidas são descritas por um modelo não linear, obtidos a partir de resultados numéricos de um modelo de BEM do problema direto. O desempenho deste modelo de problema inverso baseado em filtro de Kalman foi comparado com outros dois métodos inversos, GA e ANN, para dois modelos diferentes de problemas diretos.

Os dois modelos de problemas diretos, formulação potencial e formulação elastostática, são desenvolvidos para o BEM considerando um problema de fluxo de calor e um problema de análise de tensões, respectivamente, onde a informação de pontos internos (sensores) de uma placa é fornecida para o problema de detecção de danos. Em ambas as formulações, as informações que são fornecidas, temperatura e tensão média, independem do sistema de coordenadas. Para a formulação potencial foi mostrado que para certo posicionamento dos sensores, existe uma região da placa para a qual o filtro consegue localizar e identificar o furo com boa precisão. Já para a formulação elastostática, foi mostrado que é necessária uma variação da localização dos pontos internos para que a localização e identificação do furo fossem possíveis.

O problema inverso de detecção de danos, utilizando o filtro de Kalman, se mostrou muito eficiente desde que seus parâmetros estejam ajustados para o problema em questão.

Esta necessidade de adequação de parâmetros ao problema faz com que o método não seja robusto a ponto de se alterar o problema, ou o modelo, mantendo-se as características e parâmetros do filtro de Kalman utilizados em um problema inicial. Para uma melhor adequação destes parâmetros, pode-se realizar uma análise de sensibilidade dos mesmos, a fim de se obter uma relação de interferência sobre o método por parte de cada parâmetro.

A análise e comparação dos resultados entre o filtro de Kalman e os encontrados em Lopes (2010), mostram uma melhor precisão das componentes do estado, principalmente no modelo potencial, por parte do filtro de Kalman e também uma similaridade entre os resultados dos métodos, considerando a particularidade de cada método. Esta similaridade de resultados mostra que os métodos podem ser utilizados em paralelo, como critérios de decisão para a identificação de um dano, verificando e garantido uma confiança maior no resultado.

Como sugestão para trabalhos futuros podem ser citados:

- A análise da sensibilidade dos parâmetros de ajuste sobre o resultado do filtro de Kalman, principalmente a localização e quantidade mínima de sensores;
- Uso de modelagem acústica para o problema de detecção de danos;
- Estudo com dois ou mais danos na estrutura;
- Modelamento de danos de geometrias diversas, como trincas, por exemplo;
- Uso do método dos elementos finitos (FEM) ao invés do método dos elementos de contorno (BEM) para a formulação do problema direto;
- Estudo da influência dos erros da discretização do problema direto sobre os resultados da identificação.
- Utilizar modelos com materiais não homogêneos ou anisotrópicos;
- Estudo de um modelo de estrutura tri-dimensional;
- Comparar modelos numéricos com modelos experimentais;
- Realizar análises visando o prognóstico e não o diagnóstico de falhas.

REFERÊNCIAS BIBLIOGRÁFICAS

- ALIABADI, M. H. (2002).** “The boundary element method”, v. 2: applications in solids and structures, Wiley.
- BREBBIA, C. A., DOMINGUEZ, J. (1992).** “Boundary elements – an introductory course”, 2ª Ed. McGraw-Hill. 314 p.
- GOMES, H. M. (2004).** “Redes Neurais Artificiais para Detecção de Danos”. *Bariloche, Argentina. Mecânica Computacional*. Vol. XXIII, pp. 3351-3367. Disponível em: <<http://www.amcaonline.org.ar/ojs/index.php/mc/article/viewFile/445/428>>. Acesso em 31 mar. 2010.
- HALDAR, A., MAHADEVAN, S.** “Probability, Reability and Statistical Methods in Engineering Desing”. *John Wiley & Sons, Inc.* p.304.
- KALMAN, R. E. (1960).** “A new approach to linear filtering and prediction problems”, *Transactions of the ASME – Journal of Basic Engineering*. pp. 35–45. Disponível em: <<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.129.6247&rep=rep1&type=pdf>>. Acesso em 18 mai. 2010.
- LOPES, P. S. (2007).** “Detecção de danos em estruturas por meio de técnicas de redes neurais artificiais e algoritmos genéticos”. 106 p. Dissertação (Mestrado em Engenharia Mecânica) – Universidade federal de Itajubá, Itajubá.
- LOPES, P. S. (2010).** “Modelagem de Problema Inverso de Detecção de Danos por Técnicas de Identificação de Parâmetros e de Otimização”. 135 p. Tese (Doutorado em Engenharia Mecânica) – Universidade Federal de Itajubá, Itajubá.

- MATHWORKS (2011).** “MATLAB[®] Getting Started Guide R2011b”. 276p. The MathWorks, Inc. Set. 2011.
- MONTGOMERY, D. C, RUNGER, G. C. (2003).** “Estatística Aplicada e Probabilidade para Engenheiros”. 2^a ed. Rio de Janeiro. LTC.
- MORAIS, T. B. (2006).** “Diagnóstico de Falhas via Observadores de Estado com Excitações Desconhecidas, Identificadas via Funções Ortogonais”. 116 p. Dissertação (Mestrado em Engenharia Mecânica) – Universidade Estadual Paulista, Faculdade de Engenharia de Ilha Solteira. Disponível em: <http://www.dem.feis.unesp.br/ppgem/teses_dissertacoes/2006/dissertacao_tsmorais.pdf>. Acesso em 25 mar. 2010.
- PORTELA, A. (1992).** “Dual Boundary Element Incremental Analysis of Crack Growth”, PhD Dissertation, Wessex Institute of Technology, Damage Tolerance Division, Southampton.
- SENTHIL, R., JANARTHANAN, K., PRAKASH, J. (2006).** “Nonlinear State Estimation Using Fuzzy Kalman Filter”. 11 p. Disponível em: <<http://pubs.acs.org/doi/abs/10.1021/ie0601753>>. Acesso em 04 abr. 2011.
- SILVA, D. R. C. (2008).** “Sistema de Detecção e Isolamento de Falhas em Sistemas Dinâmicos Baseado em Identificação Paramétrica”. 115 p. Tese (Doutorado em Engenharia Elétrica) – Universidade Federal do Rio Grande do Norte, Natal. Disponível em: <ftp://ftp.ufrn.br/pub/biblioteca/ext/bdtd/DiegoRCS_tese.pdf>. Acesso em 25 mar. 2010.
- SIMANI, S., FANTUZZI, C., BEGHELLI S. (2000).** “Diagnosis Techniques for Sensor Faults of Industrial Processes”. *IEEE Transactions on Control Systems Technology*. vol. 8. No. 5. pp. 848-855. Disponível em: <<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=865858>>. Acesso em 08 abr. 2010.
- SIMON, D. (2006).** “Optimal State Estimation: Kalman, H_∞ and Nonlinear Approaches”. A *John Wiley & Sons, Inc., Publication*. 526 p.
- SWANSON, D. C., SPENCER, J. M., ARZOUMANIAN S. H. (1999).** “Prognostic Modelling of Crack Growth in a Tensioned Steel Band”. *Mechanical Systems and*

Signal Processing (2000). 14(5). pp. 789-803. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0888327000913244>>. Acesso em 08 abr. 2010.

WELCH, G. & BISHOP, G. (2001). “An introduction to the Kalman filter”. In: SIGGRAPH, course 8, 2001. Los Angeles. Disponível em: <http://www.cs.unc.edu/~tracker/media/pdf/SIGGRAPH2001_CoursePack_08.pdf>. Acesso em 31 mar 2010.

Apêndice A

REVISÃO

Este apêndice é dedicado a fazer uma breve revisão, de conceitos de probabilidade, que são a base do filtro de Kalman e é adaptado de Montgomery & Runger (2003).

A.1 PROBABILIDADE

Nós temos, na maioria, a noção de “acontecimento” aleatório, ou a probabilidade de ocorrer um dado evento em um determinado espaço amostral. A probabilidade de que um evento A venha ocorrer é definido como:

$$p(A) = \frac{\text{Resultados possíveis favorecendo } A}{\text{Número total de resultados possíveis}} \quad (36)$$

A probabilidade de um resultado favorecer A e B é dado por:

$$p(A \cup B) = p(A) + p(B) \quad (37)$$

Se dois resultados são independentes (um não afeta o outro), então a probabilidade de ambos ocorrerem é o produto das probabilidades do eventos individuais:

$$p(A \cap B) = p(A)p(B) \quad (38)$$

Como por exemplo, a probabilidade de um dado de seis lados, cair com a face “um” voltada para cima é de $1/6$, a probabilidade de dois dados, lançados juntos, caírem ambos com a face “um” para cima é de $1/36$.

Se tivermos a probabilidade de um resultado A dado a ocorrência de um resultado B , é chamado de probabilidade condicional de A dado B :

$$p(A|B) = \frac{p(A \cap B)}{p(B)} \quad (39)$$

A.2 VARIÁVEIS ALEATÓRIAS

A variável aleatória é em sua essência, uma função que mapeia todos os pontos de um espaço amostral para números reais.

No caso de uma variável aleatória contínua, a probabilidade de um evento A ser único e discreto é na verdade 0 ($p(A) = 0$). Logo, a função que representa a probabilidade de uma variável aleatória em um intervalo é chamada de função de distribuição cumulativa:

$$F_X(x) = p(-\infty, x] \quad (40)$$

As propriedades importantes da função de densidade cumulativa são:

1. $F_X(x) \rightarrow 0$ assim como $x \rightarrow -\infty$
2. $F_X(x) \rightarrow 1$ assim como $x \rightarrow +\infty$
3. $F_X(x)$ é uma função não decrescente de x

Ao invés de utilizar a função de distribuição cumulativa, comumente utiliza-se a função densidade de probabilidade:

$$f_X(x) = \frac{d}{dx} F_X(x) \quad (41)$$

Que tem as seguintes propriedades:

1. $f_X(x)$ é uma função não negativa

$$2. \int_{-\infty}^{\infty} f_X(x) dx = 1$$

Logo, a probabilidade para um intervalo $[a, b]$ pode ser definida como:

$$p_X[a, b] = \int_a^b f_X(x) dx \quad (42)$$

A.3 MÉDIA E VARIÂNCIA

Nós estamos acostumados com a noção de média de uma seqüência de números. A média de N amostras de uma variável aleatória discreta X é dada por:

$$\bar{X} = \frac{X_1 + X_2 + \dots + X_N}{N} \quad (43)$$

Quando rastreamos um sinal contínuo é comum pensarmos em termos de amostras infinitas, da mesma forma, se tivermos uma amostra infinita de uma variável aleatória para cada uma de n resultados x_1, \dots, x_n . Logo para uma variável aleatória discreta, podemos aproximar para uma média ponderada de probabilidade:

$$\bar{X} \approx \frac{(p_1 N)x_1 + (p_2 N)x_2 + \dots + (p_n N)x_n}{N} \quad (44)$$

Isto leva a definição de valor esperado para uma amostra de uma variável aleatória discreta:

$$\text{valor esperado de } X = E(X) = \sum_{i=1}^n p_i x_i \quad (45)$$

No caso de uma variável aleatória contínua, o valor esperado torna-se:

$$\text{valor esperado de } X = E(X) = \int_{-\infty}^{\infty} x f_X(x) dx \quad (46)$$

Onde ambas, podem ser aplicadas a funções de uma variável aleatória X :

$$E(g(X)) = \sum_{i=1}^n p_i g(x_i) \quad (47)$$

E

$$E(g(X)) = \int_{-\infty}^{\infty} g(x) f_X(x) dx \quad (48)$$

O valor esperado de variáveis aleatórias também é conhecido como primeiro momento estatístico. Se fizermos $g(X) = X^k$ para obtermos o k-ésimo momento estatístico:

$$E(X^k) = \int_{-\infty}^{\infty} x^k f_X(x) dx \quad (49)$$

Que para o interesse do assunto deste trabalho, é o segundo momento estatístico;

$$E(X^2) = \int_{-\infty}^{\infty} x^2 f_X(x) dx \quad (50)$$

Fazendo $g(X) = X - E(X)$, nós teremos a variância do sinal sobre a média:

$$\text{Variância de } X = E[(X - E(X))^2] = E(X^2) - E(X)^2 \quad (51)$$

Onde a raiz quadrada da variância é conhecida como desvio padrão:

$$\sigma_X = \sqrt{\text{Variância de } X} \quad (52)$$

A.4 DISTRIBUIÇÃO DE GAUSS OU NORMAL

A distribuição de Gauss ou Normal é uma distribuição de probabilidade especial bem conhecida no modelamento de sistemas aleatórios. Fazendo algumas considerações, podemos mostrar que a soma de variáveis aleatórias com uma distribuição tende a uma distribuição normal. Que é conhecido como teorema do limite central.

Seja um processo aleatório X com distribuição normal e média μ e variância σ^2 ($X \sim N(\mu, \sigma^2)$) que tem a função densidade de probabilidade:

$$f_X(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2} \frac{(x-\mu)^2}{\sigma^2}} \quad (-\infty < x < \infty) \quad (53)$$

Qualquer função linear ($Y = aX + b$) de um processo aleatório com distribuição normal ($X \sim N(\mu, \sigma^2)$) é também de distribuição normal.

$$Y \sim N(a\mu + b, a^2\sigma^2) \quad (54)$$

Logo, a função densidade de probabilidade de Y :

$$f_Y(y) = \frac{1}{\sqrt{2\pi a^2 \sigma^2}} e^{-\frac{1}{2} \frac{(y-(a\mu+b))^2}{a^2 \sigma^2}} \quad (55)$$

Se nós temos dois processos (X_1, X_2) independentes, então:

$$X_1 + X_2 \sim N(\mu_1 + \mu_2, \sigma_1^2 + \sigma_2^2) \quad (56)$$

Com a função densidade de probabilidade:

$$f_X(x_1 + x_2) = \frac{1}{\sqrt{2\pi(\sigma_1^2 + \sigma_2^2)}} e^{-\frac{1}{2} \frac{(x-(\mu_1 + \mu_2))^2}{(\sigma_1^2 + \sigma_2^2)}} \quad (57)$$

A.5 PROBABILIDADE CONDICIONAL E INDEPENDÊNCIA CONTÍNUA

Para serem estatisticamente independentes, duas variáveis aleatórias X e Y , devem ter sua probabilidade conjunta $f_{XY}(x, y)$ igual ao produto de suas probabilidades.

$$f_{XY}(x, y) = f_X(x)f_Y(y) \quad (58)$$

Complementando, a regra de Bayes especifica a densidade de probabilidade da variável aleatória X dada à variável aleatória Y :

$$f_{X|Y}(x) = \frac{f_{Y|X}(y)f_X(x)}{f_Y(y)} \quad (59)$$

Dado um processo discreto X e um contínuo Y , a função massa de probabilidade discreta para X condicionada em $Y = y$, é dada por:

$$p_X(x|Y = y) = \frac{f_Y(y|X=x)p_X(x)}{\sum_z f_Y(y|X=z)p_X(z)} \quad (60)$$

Apêndice B

DISTRIBUIÇÃO T-STUDENT

Este apêndice mostra o *teste t* e a tabela contendo os valores de t .

B.1 TESTE T E INTERVALO DE CONFIANÇA

O *teste t*, mede a probabilidade da média de uma amostra apresentar o valor apresentado \bar{x} dada a média da população μ_0 . Para o teste, calculamos a estatística t com a Equação (61):

$$t = \frac{\bar{x} - \mu_0}{\frac{s}{\sqrt{n}}} \quad (61)$$

Onde:

\bar{x} é a média da amostra;

μ_0 é o valor fixo usado para comparação com a média da amostra;

s é o desvio padrão amostral;

n é o tamanho da amostra.

Livros de estatística trazem tabelas que contém o valor de t , em função do número de graus de liberdade ($\nu = n - 1$) e níveis de confiança (α). A Tabela B1 traz alguns valores.

Tabela B1 – Tabela de CDF da distribuição t-student. [Adaptado de Haldar & Mahadevan (2000)]

$\nu \backslash \alpha$	0,600	0,750	0,800	0,900	0,950	0,975	0,990	0,995	0,999
1	0,325	1,000	1,376	3,078	6,314	12,706	31,821	63,656	318,289
2	0,289	0,816	1,061	1,886	2,920	4,303	6,965	9,925	22,328
3	0,277	0,765	0,978	1,638	2,353	3,182	4,541	5,841	10,214
4	0,271	0,741	0,941	1,533	2,132	2,776	3,747	4,604	7,173
5	0,267	0,727	0,920	1,476	2,015	2,571	3,365	4,032	5,894
6	0,265	0,718	0,906	1,440	1,943	2,447	3,143	3,707	5,208
7	0,263	0,711	0,896	1,415	1,895	2,365	2,998	3,499	4,785
8	0,262	0,706	0,889	1,397	1,860	2,306	2,896	3,355	4,501
9	0,261	0,703	0,883	1,383	1,833	2,262	2,821	3,250	4,297
10	0,260	0,700	0,879	1,372	1,812	2,228	2,764	3,169	4,144
11	0,260	0,697	0,876	1,363	1,796	2,201	2,718	3,106	4,025
12	0,259	0,695	0,873	1,356	1,782	2,179	2,681	3,055	3,930
13	0,259	0,694	0,870	1,350	1,771	2,160	2,650	3,012	3,852
14	0,258	0,692	0,868	1,345	1,761	2,145	2,624	2,977	3,787
15	0,258	0,691	0,866	1,341	1,753	2,131	2,602	2,947	3,733
20	0,257	0,687	0,860	1,325	1,725	2,086	2,528	2,845	3,552
25	0,256	0,684	0,856	1,316	1,708	2,060	2,485	2,787	3,450
40	0,255	0,681	0,851	1,303	1,684	2,021	2,423	2,704	3,307
60	0,254	0,679	0,848	1,296	1,671	2,000	2,390	2,660	3,232
150	0,254	0,676	0,844	1,287	1,655	1,976	2,351	2,609	3,145
∞	0,253	0,674	0,842	1,282	1,645	1,960	2,326	2,576	3,090

Logo, para se calcular o intervalo de confiança deve-se usar a equação (62):

$$I.C._{\alpha/2} = \left(\bar{x} \pm t \frac{s}{\sqrt{n}} \right) \quad (62)$$

Como exemplo, se quisermos encontrar um intervalo de confiança de 95% para uma amostra com quatro graus de liberdade, devemos utilizar um valor de t igual a 2,132 caso a distribuição tenha uma cauda ou 2,776 quando for bi-caudal.

Apêndice C

CÓDIGOS MODELO POTENCIAL

Este apêndice mostra os códigos utilizados na formulação do modelo potencial, implementados em no software MATLAB® [MATHWORKS (2011)].

C.1 ROTINA POT_FKE

```
% modelo potencial - rotina principal e variáveis de entrada
%-----
%inicialização
clear all %apaga todos dados pré-existentes
close all %fecha todas janelas pré abertas
clc %limpa a tela de comando
warning off %desliga todas mensagens de aviso
tic %inicia a contagem de tempo
%-----
%Posição do furo em uma placa de 6x6 [cm]:
C=[3,3,0.06]; %C=[X,Y,R]: X - abscissa do centro do furo [cm]
                %           Y - ordenada do centro do furo [cm]
                %           R - raio do furo [cm]

zk=bem(C(3),C(1),C(2)); % Leitura das medidas zk através da rotina bem
m=size(zk); % capta o número de sensores utilizados, através do tamanho do
                %vetor zk
%-----
% estimativa inicial e parâmetros
x0=[3;3;0.001];
X=[x0];

%Covariância do ruído da medida
R=((0.01*eye(m(1)))/3)^2;
%Covariância do ruído do estado
Q=(( [0.1,0,0;0,0.1,0;0,0,0.01] )^2);

tol=1E-6; %tolerância
```

```

it_max=75; %número máximo de iterações

%-----
%rotina do Filtro de Kalman
[xk,it,X,P]=FKE(x0,R,Q,zk,X,tol,it_max);

%-----
Resultado(xk,it,C,X); %imprime os resultados através da rotina Resultado

toc %termina a contagem de tempo
%-----
%fim da rotina

```

C.2 ROTINA FKE

%Função que contém o algoritmo do filtro de Kalman, que retorna o estado xk
%o número de iterações e a matriz contendo todos os estados calculados pelo
%filtro durante as iterações.

```

%-----
function [xk,it,X,Pk]=FKE(x0,R,Q,zk,X,tol,it_max)

P0=eye(3); %covariância inicial do processo.
xk=x0;
Pk=P0;
it=0; %contador inicial de iterações
rep=0; %contador inicial de repetições da tolerância

while (rep<3)&&(it<it_max)

    xk_p=xk;
    Pk_p=Pk+Q;
    Pk_p=0.5*(Pk_p+Pk_p'); %Garante a simetria da matriz.
    zk_p=bem(xk_p(3,1),xk_p(1,1),xk_p(2,1));
    Hk=df(xk); %calcula da matriz de sensibilidade
    Kk=Pk_p*Hk'/(Hk*Pk_p*Hk'+R); %cálculo do ganho de Kalman
    xk=xk_p+Kk*(zk-zk_p);%atualização do estado
    Pk=(eye(3)-Kk*Hk)*Pk_p*(eye(3)-Kk*Hk)'+Kk*R*Kk';%atualização da
                                                %covariância do
                                                %processo (garante ser
                                                %simetrica positiva
                                                %definida(Simon, D.,
                                                %p.129))

    xk=xk.*sign(xk); %garante valores positivos do estado
    dif=max(abs(xk-xk_p)); %critério de convergência para análise
    if dif<=tol
        rep=rep+1;%incremento do contador de repetições da tolerância
    else
        rep=0;
    end
    it=it+1;%incremento do contador de iterações
    X=[X xk];%atualização da matriz que contém os estados

end
%-----
%fim da rotina

```

C.3 ROTINA RESULTADO

```

%Rotina que retorna o resultado do filtro, imprimindo a ultima iteração na
%tela e plotando todo processo de estimacao
%-----
function Resultado(xk,it,C,X)

p=('%');
fprintf(' \n')
fprintf('          Furo real | Furo estimado | erro\n')
fprintf('X      = %f | %f | %f %s\n',C(1),xk(1,1),(xk(1,1)/C(1)-1)*100,p)
fprintf('Y      = %f | %f | %f %s\n',C(2),xk(2,1),(xk(2,1)/C(2)-1)*100,p)
fprintf('Raio = %f | %f | %f %s\n',C(3),xk(3,1),(xk(3,1)/C(3)-1)*100,p)

fprintf('\n O número de iterações necessárias foram de %d\n',it);
m=size(X);
t=0:m(2)-1;
figure()
subplot(3,1,1)
plot(t,X(1,:), 'r', [0 m(2)], [C(1) C(1)], '--b')
xlabel('iterações')
ylabel('X [cm]')
legend('Estimado', 'Real')
title('Estimação do Filtro')
subplot(3,1,2)
plot(t,X(2,:), 'r', [0 m(2)], [C(2) C(2)], '--b')
xlabel('iterações')
ylabel('Y [cm]')
legend('Estimado', 'Real')
subplot(3,1,3)
plot(t,X(3,:), 'r', [0 m(2)], [C(3) C(3)], '--b')
xlabel('iterações')
ylabel('R [cm]')
legend('Estimado', 'Real')

x=[0 6 6 0 0];
y=[0 0 6 6 0];
n=-2:0.05:2;
figure()
title('Estimação do Filtro')
hold on
xlabel('X [cm]')
ylabel('Y [cm]')
axis([-0.5 6.5 -0.5 6.5])
plot(0.06,0.06, '*b')
plot(C(3)*cos(n.*pi)+C(1),C(3)*sin(n.*pi)+C(2), 'm')
plot(X(3,1)*cos(n.*pi)+X(1,1),X(3,1)*sin(n.*pi)+X(2,1), '-.g')
plot(X(3,2)*cos(n.*pi)+X(1,2),X(3,2)*sin(n.*pi)+X(2,2), ':r')
plot(X(3,m(2))*cos(n.*pi)+X(1,m(2)),X(3,m(2))*sin(n.*pi)+X(2,m(2)), 'k')
legend('Sensores', 'Furo real', 'Estimativa inicial', 'Estimativas
intermediárias', 'Estimativa Final')
plot(x,y)
plot(0.06,3, '*b', 0.06,5.94, '*b', 3,5.94, '*b', ...
      5.94,5.94, '*b', 5.94,3, '*b', 5.94,0.06, '*b', 3,0.06, '*b')
for i=3:m(2)-1
    plot(X(3,i)*cos(n.*pi)+X(1,i),X(3,i)*sin(n.*pi)+X(2,i), ':r')
end
%-----
%fim da rotina

```

C.4 ROTINNA DF

```
%rotina que calcula a matriz de sensibilidade
%-----
function H=df(xk)

deltax=0.002;
deltay=0.002;
deltar=0.002;

Sar = bem(xk(3,1)-deltar,xk(1,1),xk(2,1));
Sdr = bem(xk(3,1)+deltar,xk(1,1),xk(2,1));

Sax = bem(xk(3,1),xk(1,1)-deltax,xk(2,1));
Sdx = bem(xk(3,1),xk(1,1)+deltax,xk(2,1));

Say = bem(xk(3,1),xk(1,1),xk(2,1)-deltay);
Sdy = bem(xk(3,1),xk(1,1),xk(2,1)+deltay);

H1=(Sdx-Sax)/(2*deltax);
H2=(Sdy-Say)/(2*deltay);
H3=(Sdr-Sar)/(2*deltar);

H=[H1 H2 H3];
%-----
%fim da rotina
```

C.5 ROTINA BEM

```
% -----VARIAVEIS -----
%
% npi: numero de pontos internos
% xi: coordenadas dos pontos internos
% x: coordenadas dos nos
% kode: 0 -> potencial prescrito;
%       1 -> fluxo prescrito
% fi: condições iniciais de contorno
% n_placa: numero de nos da placa
% n_furo: numero de nos do furo
% r: raio do furo
% ixc: abscissa do centro do circulo
% iyc: ordenada do centro do circulo
%
%-----
function [fipi] = bem(r,ixc,jyc)

format short
n_furo = 24;

%DETERMINACAO DA POSICAO DO FURO (ixc,jyc)
x = determ_furo(n_furo,r,ixc,jyc);

[n,npi,x,kode,fi,xi,n_placa] = inputpc(x);
[pc,fi,dfi,xi,fipi] = poconbe_v5(n,npi,x,kode,fi,xi,n_placa);
%-----
%fim da rotina
```

C.6 ROTINA DETERM_FURO

```

%determina a posicao do furo e dos nos
%-----
function [x] = determ_furo(n,r,xc,yc)

xaux=zeros(1,n+1);
yaux=zeros(1,n+1);
x=zeros(n+1,2);
for i = 1:n+1
    %no sentido horário: DOMINIO ABERTO (FURO)
    xaux(i) = xc - r*cos(-2*pi*(i-1)/n);
    yaux(i) = yc - r*sin(-2*pi*(i-1)/n);
    x(i,1) = xaux(i);
    x(i,2) = yaux(i);
end
%-----
%fim da rotina

```

C.7 ROTINA INPUTPC

```

%-----SUBROTINA INPUTPC -----
%
% DADOS DE ENTRADA
%
%----- VARIAVEIS -----
%
% npi: numero de pontos internos
% xi: coordenadas dos pontos internos
% x_placa: coordenadas dos nos
% x: coordenadas do furo
% kode: 0 -> potencial prescrito;
%       1 -> fluxo prescrito
% fi: condições iniciais de contorno
% n: numero de nos (elementos) totais (placa + furo)
% n_placa: numero de nos (elementos) da placa
%
%-----

function [n,npi,x,kode,fi,xi,n_placa]=inputpc(x)

n_placa = 12;

%coordenadas da placa
%em cm:
x_placa=[0 0
          2 0
          4 0
          6 0
          6 2
          6 4
          6 6
          4 6
          2 6
          0 6
          0 4
          0 2
          0 0]; %para fechar

```



```
% reordenar os vetores de potencial e fluxo para impressao
[fi dfi]=reorder(fi,dfi,kode,n);
```

```
% calculo do potencial nos pontos internos
fipi=interp(n,npix,xi,dfi,fi);
```

```
%-----
```

```
%fim da rotina
```

C.9 ROTINA GHMATPC

```
%-----SUBROTINA GHMATPC-----
```

```
function [g,h]=ghmatpc(n,x)
```

```
pc=zeros(n,2);
```

```
h=zeros(n,n);
```

```
g=zeros(n,n);
```

```
epi=0; % epi=0 não é cálculo de ponto interno
```

```
for i=1:n % varredura dos nós
```

```
    pc(i,:)=0.5*(x(i,:)+x(i+1,:)); % ponto de colocação
```

```
    for j=1:n % varredura dos elementos
```

```
        if i == j
```

```
            [haux gaux]=locinpc(i,j,pc(i,:),x(j,:),x(j+1,:));%integração analítica
```

```
        else
```

```
            [haux gaux]=extinpc(i,j,pc(i,:),x(j,:),x(j+1,:),epi);%integração numérica
```

```
        end
```

```
        h(i,j)=haux;
```

```
        g(i,j)=gaux;
```

```
    end
```

```
end
```

```
%-----
```

```
%fim da rotina
```

C.10 ROTINA LOCINPC

```
%-----SUBROTINA LOCINPC -----
```

```
%
```

```
%Calcula os elementos da matriz G, correspondendo as integrais ao longo
```

```
%dos elementos que incluem a singularidade
```

```
%
```

```
%-----
```

```
function [h,g]=locinpc(i,j,pc,x1,x2)
```

```
L=norm(x2-x1); % comprimento do elemento
```

```
h=0.5;
```

```
g=L/(2*pi)*(1+log(2/L));
```

```
%-----
```

```
%fim da rotina
```

C.11 ROTINA EXTINPC

```

%----- SUBROTINA EXTINPC -----
%
% Retorna as integrais numericas do ponto de colocação pc até o elemento
% formado pelos pontos x1 e x2
%
%----- VARIAVEIS -----
%
% ng: numero pontos de Gauss
% w: pesos dos pontos de Gauss
% e: coordenadas dos pontos de Gauss
% epi: flag para ponto interno
% pc: ponto de colocação
%
%-----

function [h,g]=extinpc(i,j,pc,x1,x2,epi)
ng=4; % nr de pontos de integração
[w e]=gauss(ng);
L=norm(x2-x1); % comprimento do elemento
xc=zeros(ng,2);
r=zeros(1,ng);
for z=1:ng
    xc(z,:)=e(1,z).*(x1-x2)./2 +(x1+x2)./2; %mapeam.conforme [0 L] -> [-1 1]
    r(1,z)=norm(xc(z,:)-pc); %distância ponto de colocação ponto de integração
end

D=dist(pc,x1,x2); % distância do ponto de colocação ao elemento

if (i==j) && (epi~=1) %epi=1 flag para ponto interior
    h=0.5;
    g=L/(2*pi)*(1+log(2/L));
else
    Haux=-1./(2*pi*r.^2)*D.*w*L/2;
    Gaux=1./(2*pi).*log(1./r).*w*L/2;
    h=sum(Haux);
    g=sum(Gaux);
end

%-----
%fim da rotina

```

C.12 ROTINA GAUSS

```

%----- SUBROTINA GAUSS -----
%

function [w,e]=gauss(ng)

x1=-1;
x2=1;
m=(ng+1)/2;
xm=0.5*(x2+x1);
x1=0.5*(x2-x1);
ep=3.0*10^(-14);

for i=1:m
    z=cos(pi*(i-0.25)/(ng+0.5));

```

```

p1=1;
p2=0;
for j=1:ng
    p3=p2;
    p2=p1;
    p1=((2*j-1)*z*p2-(j-1)*p3)/j;
end
pp=ng*(z*p1-p2)/(z*z-1);
z1=z;
z=z1-p1/pp;
while abs(z-z1)>ep
    p1=1;
    p2=0;
    for j=1:ng
        p3=p2;
        p2=p1;
        p1=((2*j-1)*z*p2-(j-1)*p3)/j;
    end
    pp=ng*(z*p1-p2)/(z*z-1);
    z1=z;
    z=z1-p1/pp;
end
e(i)=xm-x1*z;
e(ng+1-i)=xm+x1*z;
w(i)=2*x1/((1-z^2)*pp^2);
w(ng+1-i)=w(i);
end

%-----
%fim da rotina

```

C.13 ROTINA DIST

```

%-----SUBROTINA DIST -----
%
% Calcula a distância do ponto xp ate a reta formada por x1 e x2
%
%-----VARIAVEIS-----
%
% xp: coordenadas do ponto de colocação
% x1: primeiro nó do elemento a ser varrido
% x2: segundo nó do elemento a ser varrido
%
%-----

function d=dist(xp,x1,x2)

xp=[xp 0];
x2=[x2 0];
x1=[x1 0];
a=(xp-x2);
b=(x1-x2)/norm(x1-x2);
d=norm(cross(a,b));
sig=sum(cross((x1-xp),(x2-xp)));
if sig < 0
    d=-d;
end

%-----
%fim da rotina

```

C.14 ROTINA APLYBC

```

%----- SUBROTINA APLYBC -----
%
%Reordena as colunas do sistema de acordo com as condicoes de contorno
%e forma a matriz coeficiente armazenada em G e o lado direito em DFI
%
%-----VARIAVEIS-----
%
% h: matriz quadrada (n,n)
% g: matriz retangular (n,2*n)
%
%-----

function [g,dfi]=aplybc(g,h,kode,fi,n)
for j=1:n
    if kode(j) ~= 0
        for i=1:n
            ch=g(i,j);
            g(i,j)=-h(i,j);
            h(i,j)=-ch;
        end
    end
end
dfi=h*fi';
dfi=dfi';
%-----
%fim da rotina

```

C.15 ROTINA REORDER

```

%----- SUBROTINA REORDER-----
%
%Armazena o potencial (fi) e o fluxo (dfi) nos respectivos vetores
%
%-----VARIAVEIS-----
%
% fi: potencial do nó
% dfi: fluxo antes no nó
%
%-----

function [fi,dfi]=reorder(fi,dfi,kode,n)
for i=1:n
    if kode(i) ~= 0
        ch=fi(i);
        fi(i)=dfi(i);
        dfi(i)=ch;
    end
end
%-----
%fim da rotina

```

C.16 ROTINA INTERPC

```

%-----SUBROTINA INTERPC-----
%
% Calcula os valores de potencial nos pontos internos
%
%-----VARIABLES-----
%
% fipi: potencial no ponto interno
% xi: coordenadas dos pontos internos
%
%-----
function fipi=interpc(n,npix,xi,dfi,fi)
clear g h;
h=zeros(npix,n);
g=zeros(npix,n);
epi=1; % epi=1 calculo de ponto interno
for i=1:npix
    for j=1:n
        [haux gaux]=extinpc(i,j,xi(i,:),x(j,:),x(j+1,:),epi); % integração numérica
        h(i,j)=haux;
        g(i,j)=gaux;
    end
end
% solução para pontos internos (Paris & Cannas pag 77, eq 3.4.18)
fipi=g*dfi-h*fi';

%-----
%fim da rotina

```

Apêndice D

CÓDIGOS MODELO ELASTOSTÁTICO

Este apêndice mostra os códigos utilizados na formulação do modelo elastostático, implementados no software MATLAB® [MATHWORKS (2011)].

D.1 ROTINA ELASTO_MAIN

```
%Modelo Elastostatico - rotina principal e algumas variaveis de entrada
%-----
%inicialização
clear all %apaga todos dados pré-existentes
close all %fecha todas janelas pré abertas
clc %limpa a tela de comando
warning off %desliga todas mensagens de aviso
tic %inicia a contagem de tempo

%-----
%Posição do furo em uma placa de 6x6 [cm]:

C=[3,3,0.12]; %C=[X,Y,R]: X - abscissa do centro do furo [cm]
                %           Y - ordenada do centro do furo [cm]
                %           R - raio do furo [cm]

X=[];
%posição dos sensores inicial
xi = [0.06 0.06
      0.06 3
      0.06 5.94
      3 5.94
      5.94 5.94
      5.94 3
      5.94 0.06
      3 0.06];

xk0=rot_inicial(C,xi); %retorna a estimativa do filtro de Kalman
xka=xk0;
```

```

X=[X xka];
rep=0; %iniciando o contador de repeticoes
tol=1e-3; %tolerancia para conversao
Xt=xka;

for dist_sens=0:0.05:1.8 %variação da posição dos sensores
    xi=[0.6+dist_sens 0.6+dist_sens
        0.6+dist_sens 3
        0.6+dist_sens 5.4-dist_sens
        3 5.4-dist_sens
        5.4-dist_sens 5.4-dist_sens
        5.4-dist_sens 3
        5.4-dist_sens 0.6+dist_sens
        3 0.6+dist_sens];

    xk=rot_inicial(C,xi);

    dif=max(abs(xk-xka)); %calcula a diferenca entre a estimativa para a
                        %posicao anterior de sensores e a atual
    Xt=[Xt xk];

    if dif<=tol %verificação de convergência
        X=[X xk];
        rep=rep+1;
    else
        clear X %limpa da memória a variavel X
        X=[xk];
        xka=xk;
        rep=0;
    end
    if rep==2
        Resultado(C,xi,X,Xt) %impressão dos resultados
        break %interrompe o programa, imprimindo na tela os resultados
    end
end
Resultado(C,xi,X,Xt) %impressão dos resultados
toc %finaliza a contagem do tempo
%-----
%fim da rotina

```

D.2 ROTINA ROT_INICIAL

```

%Rotina que insere alguns parametros do problema e chama a rotina do filtro para
%o calculo do estado
%-----
function xk=rot_inicial(C,xi)

L=1e3; %Carga aplicada [MPa]
zk=DOE_bem_elasticity(C(3),C(1),C(2),L,xi); %Leitura das medidas zk e fornecimento
                        %das posições dos sensores
m=size(zk); % capta o número de sensores utilizados, através do tamanho do
            %vetor zk

%-----
%estimativa inicial e parâmetros
x0=[3;3;0.01]; %x0=[X,Y,R]: X - abscissa do centro do furo [cm]
                %           Y - ordenada do centro do furo [cm]
                %           R - raio do furo [cm]

```

```

%Covariância do ruído da medida
R=(std(zk)*eye(m(1)))^2;
%Covariância do ruído do estado
Q=(( [0.01,0,0;0,0.03,0;0,0,0.001]*1e3)^2);
tol=1E-3; %tolerância

it_max=75; %número máximo de iterações

%-----
%rotina do Filtro de Kalman
xk=FKE(x0,R,Q,zk,tol,it_max,L,xi);

%-----
%fim da rotina

```

D.3 ROTINA RESULTADO

```

%rotina que retorna e imprime o resultado da estimação na tela
%-----
function Resultado(C,xi,X,Xt)

m=size(X);
x=[0 6 6 0 0];
y=[0 0 6 6 0];

n=-2:0.05:2;

figure(1)%figura referente ao(s) ultimo(s) resultado(s) do filtro,
          %convergencia (três ultimos) ou não da estimativa(menos que três).
title('Estimação do Filtro')
hold on
xlabel('X [cm]')
ylabel('Y [cm]')
axis([-0.5 6.5 -0.5 6.5])
plot(xi(1,1),xi(1,2),'*b')
plot(C(3)*cos(n.*pi)+C(1),C(3)*sin(n.*pi)+C(2),'k')
plot(X(3,1)*cos(n.*pi)+X(1,1),X(3,1)*sin(n.*pi)+X(2,1),'r')
legend('Sensores','Furo real','Estimativas')
plot(x,y)
plot(xi(2,1),xi(2,2),'*b',xi(3,1),xi(3,2),'*b',xi(4,1),xi(4,2),'*b',...
      xi(5,1),xi(5,2),'*b',xi(6,1),xi(6,2),'*b',xi(7,1),xi(7,2),'*b',...
      xi(8,1),xi(8,2),'*b')
for i=2:m(2)
    plot(X(3,i)*cos(n.*pi)+X(1,i),X(3,i)*sin(n.*pi)+X(2,i),'r')
end
m1=size(Xt);

figure(2)%figura referente a todos resultados do filtro.
title('Estimação do Filtro')
hold on
xlabel('X [cm]')
ylabel('Y [cm]')
axis([-0.5 6.5 -0.5 6.5])
plot(xi(1,1),xi(1,2),'*b')
plot(C(3)*cos(n.*pi)+C(1),C(3)*sin(n.*pi)+C(2),'k')
plot(Xt(3,1)*cos(n.*pi)+Xt(1,1),Xt(3,1)*sin(n.*pi)+Xt(2,1),'r')
legend('Sensores','Furo real','Estimativas')
plot(x,y)
plot(xi(2,1),xi(2,2),'*b',xi(3,1),xi(3,2),'*b',xi(4,1),xi(4,2),'*b',...

```

```

        xi(5,1),xi(5,2),'*b',xi(6,1),xi(6,2),'*b',xi(7,1),xi(7,2),'*b',...
        xi(8,1),xi(8,2),'*b')
    for i=2:m1(2)
        plot(Xt(3,i)*cos(n.*pi)+Xt(1,i),Xt(3,i)*sin(n.*pi)+Xt(2,i),'r')
    end

%-----
%fim da rotina

```

D.4 ROTINA FKE

%Função que contém o algoritmo do filtro de Kalman, que retorna o estado x_k
 %o número de iterações e a matriz contendo todos os estados calculados pelo
 %filtro durante as iterações.

```

%-----
function xk=FKE(x0,R,Q,zk,tol,it_max,L,xi)

P0=eye(3); %covariância inicial do processo.
xk=x0;
Pk=P0;

it=0; %contador inicial de iterações
rep=0; %contador inicial de repetições da tolerância

while (rep<3)&&(it<it_max)

    xk_p=xk;
    Pk_p=Pk+Q;
    Pk_p=0.5*(Pk_p+Pk_p'); %Garante a simetria da matriz.
    zk_p=DOE_bem_elasticity(xk_p(3,1),xk_p(1,1),xk_p(2,1),L,xi);
    Hk=df(xk,L,xi); %cálculo da matriz de sensibilidade
    Kk=Pk_p*Hk'/(Hk*Pk_p*Hk'+R); %cálculo do ganho de Kalman
    %atualização do estado
    xk=xk_p+Kk*(zk-zk_p);
    Pk=(eye(3)-Kk*Hk)*Pk_p*(eye(3)-Kk*Hk)'+Kk*R*Kk';%atualização da covariância do
                                                %processo (garante ser
                                                %simetrica positiva
                                                %definida(Simon, D., p.129))

    xk=xk.*sign(xk); %garante valores positivos
    dif=max(abs(xk-xk_p)); %critério de convergência para análise
    if dif<=tol
        rep=rep+1;%incremento do contador de repetições da tolerância
    else
        rep=0;
    end
    it=it+1;%incremento do contador de iterações
end

%-----
%fim da rotina

```

D.5 ROTINA DF

```

%rotina que calcula a matriz de sensibilidade (H) em cada iteração, por
%diferencas finitas
%-----
function H=df(xk,L,xi)

%incrementos
deltax=0.0035;
deltay=0.0035;
deltar=0.002;

Sar = DOE_bem_elasticity(xk(3,1)-deltar,xk(1,1),xk(2,1),L,xi);
Sdr = DOE_bem_elasticity(xk(3,1)+deltar,xk(1,1),xk(2,1),L,xi);

Sax = DOE_bem_elasticity(xk(3,1),xk(1,1)-deltax,xk(2,1),L,xi);
Sdx = DOE_bem_elasticity(xk(3,1),xk(1,1)+deltax,xk(2,1),L,xi);

Say = DOE_bem_elasticity(xk(3,1),xk(1,1),xk(2,1)-deltay,L,xi);
Sdy = DOE_bem_elasticity(xk(3,1),xk(1,1),xk(2,1)+deltay,L,xi);

H1=(Sdx-Sax)/(2*deltax);
H2=(Sdy-Say)/(2*deltay);
H3=(Sdr-Sar)/(2*deltar);

H=[H1 H2 H3];

%-----
%fim da rotina

```

D.6 ROTINA DOE_BEM_ELASTICITY

```

% -----VARIABLES-----
%
% npi: numero de pontos internos
% xi: coordenadas dos pontos internos
% x: coordenadas dos nos
% kode: 0->deslocamento prescrito;
%       1->tração prescrita
% fi: condições iniciais de contorno nas direções x1 e x2
% n_placa: numero de nos da placa
% n_furo: numero de nos do furo
% r: raio do furo
% ixc: abscissa do centro do circulo
% iyc: ordenada do centro do circulo
% b: base da placa
% h: altura da placa
%
%-----
function sigma_m = DOE_bem_elasticity(r,ixc,jyc,carga,xi)

n_furo = 12;

x_furo = determ_furo(n_furo,r,ixc,jyc);

%COM FURO
[n,npi,x,kode,fi,xi,ge,xnu,n_placa]=DOE_inputec_PlacaFuro(x_furo,carga,xi);

```

```
[pc,fi,dfi,ss,ds] = DOE_elconbe_v2(n,npix,kode,fi,xi,ge,xnu,n_placa);

%-----
%tensão média para placa com furo (2D) - SIGMA X e Y
%COM FURO
sigma_m=zeros(npix,1);
for i=1:npix
    sigma_m(i,1) = (ss(3*i-2) + ss(3*i))/2;
end

%-----
%fim da rotina
```

D.7 ROTINA DETERM_FURO

```
%rotina que determina a posicao do furo e dos nos
%-----

function [x] = determ_furo(n,r,xc,yc)

for i = 1:n+1

    %no sentido horário: DOMINIO ABERTO (FURO)
    xaux(i) = xc - r*cos(-2*pi*(i-1)/n);
    yaux(i) = yc - r*sin(-2*pi*(i-1)/n);
    x(i,1) = xaux(i);
    x(i,2) = yaux(i);
end

%-----
%fim da rotina
```

D.8 ROTINA DOE_INPUTEC_PLACAFURO

```
% contem os dados para a solucao do problema
%
% -----VARIABLES-----
%
% n: numero de nós
% npix: numero de pontos internos
% xi: coordenadas dos pontos internos
% x: coordenadas dos nos
% kode: 0->deslocamento prescrito;
%       1->tração prescrita
% fi: condições iniciais de contorno nas direções x1 e x2
% ge: shear modulus
% xnu: coeficiente de Poisson para deformação plana (plane strain); use um
% coeficiente de Poisson fictício xnu=v/(1+v) para tensão plana (plane stress)
% v sendo o coeficiente de Poisson (Poisson's ratio)
%
%-----

function [n,npix,x,kode,fi,xi,ge,xnu,n_placa]=DOE_inputec_PlacaFuro(x_furo,carga,xi)
```

```

%coordenadas da placa
%-----
%24 elementos [cm]
n_placa = 24;
i = 1;
for x1 = 0:1:6
    x_placa(i,:) = [x1 0];
    i = i+1;
end
for y1 = 1:1:6
    x_placa(i,:) = [6 y1];
    i = i+1;
end
for x2 = 5:-1:0
    x_placa(i,:) = [x2 6];
    i = i+1;
end
for y2 = 5:-1:0
    x_placa(i,:) = [0 y2];
    i = i+1;
end

%-----
x = [x_placa
     x_furo(:,1) x_furo(:,2)];
n = length(x);
%coordenadas dos pontos internos
%-----
npi = length(xi(:,1)); %numero de pontos internos

%parametros da placa
ge = 94500;
xnu = 0.1;

%condicoes iniciais de contorno
%obs.: colocar 1 pto a mais na placa e no furo para fechar o contorno
%-----
%24 elementos (24+24+2=50+furo)
kode = [1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 ...
        1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 ...
        1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1];
%valores das variáveis conhecidas nas direções 'x' e 'y'
fi = [0 -carga 0 -carga 0 -carga 0 -carga 0 -carga 0 -carga ...
      0 0 0 0 0 0 0 0 0 0 carga 0 carga 0 carga 0 carga 0 carga ...
      0 0 0 0 0 0 0 0 0 0 -carga ...
      0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0];

%-----
%fim da rotina

```

D.9 ROTINA DOE_ELCONBE_V2

```

%-----PROGRAMA ELCONBE-----
%
% Potential Problems using Constant Boundary Elements
% Brebbia & Dominguez 2nd Ed. - Cap 4.5
%(adaptado)
%
%-----VARIÁVEIS-----
%
% fi: vector where the prescribed values of boundary conditions are stored.

```

```

%      Each element is associated with a value of KODE.
% dfi: Right and side vector in the global system. After solution in
%      contains the values of the unknowns.
% ds: Values of displacements at internal points (2 displacements per
%      point).
% ss: Values of stresses at internal points (3 stresses per point).
%
%-----
function [pc,fi,dfi,ss,ds] = DOE_elconbe_v2(n,npix,x,kode,fi,xi,ge,xnu,n_placa)

format long

x(n+1,:) = x(1,:); % "fechando o contorno"

% Calculo das matrizes g e h
[G H pc] = ghmatec(n,x,ge,xnu);

% aplicando as condições de contorno
[G dfi] = aplybc(G,H,kode,fi,n);

% solução do sistema
dfi = G\dfi';

[fi dfi] = reorder(fi,dfi,kode,n);

% solução para pontos interiores
[ds ss] = interec(n,npix,xi,x,fi,dfi,ge,xnu);

%-----
%fim da rotina

```

D.10 ROTINA GHMATEC

```

%-----SUBROTINA GHMATEC-----
%
%-----VARIAVEIS-----
%
% Matrizes g e h
%
%-----
function [G,H,pc]=ghmatec(n,x,ge,xnu)

H=zeros(2*n);
G=zeros(2*n);
pc=zeros(n,2);
for i=1:n, % varredura dos nós
    pc(i,:)=0.5*(x(i,:)+x(i+1,:)); % ponto de colocação
    for j=1:n, % varredura dos elementos elemento
        if i == j
            [g11 g12 g22]=locinec(x(j,:),x(j+1,:),ge,xnu);%integração analítica
            H((2*i-1),(2*j-1))=0.5;
            H((2*i-1),(2*j))=0;
            H((2*i),(2*j-1))=0;
            H((2*i),(2*j))=0.5;
            G((2*i-1),(2*j-1))=g11;
            G((2*i-1),(2*j))=g12;
            G((2*i),(2*j-1))=g12;
            G((2*i),(2*j))=g22;
        else
            [g11 g12 g22 h11 h12 h21 h22]=extinec(pc(i,:),x(j,:),x(j+1,:),ge,xnu);

```

```

%integração numérica
    H((2*i-1),(2*j-1))=h11;
    H((2*i-1),(2*j))=h12;
    H((2*i),(2*j-1))=h21;
    H((2*i),(2*j))=h22;
    G((2*i-1),(2*j-1))=g11;
    G((2*i-1),(2*j))=g12;
    G((2*i),(2*j-1))=g12;
    G((2*i),(2*j))=g22;
end
end
end
%-----
%fim da rotina

```

D.11 ROTINA LOCINEC

```

%-----SUBROTINA LOCINEC-----
%
% Computes the elements of matrix G that relate an element with itself
%
%-----VARIAVEIS-----
%
% L: comprimento do elemento
%
%-----

function [g11,g12,g22]=locinec(x1,x2,ge,xnu)
L=norm(x2-x1);
de=8*pi*ge*(1-xnu);
g11=L*((3-4*xnu)*(1-log(L/2))+(x2(1,1)-x1(1,1))^2/(L^2))/de;
g22=L*((3-4*xnu)*(1-log(L/2))+(x2(1,2)-x1(1,2))^2/(L^2))/de;
g12=(x2(1,1)-x1(1,1))*(x2(1,2)-x1(1,2))/(L*de);

%-----
%fim da rotina

```

D.12 ROTINA APLYBC

```

%-----SUBROTINA APLYBC-----
%
% reorder columns of the system according to boundary conditions
% and form the coeficient matrix stored in G and rigth hand side in DFI
%
%-----VARIAVEIS-----
%
% h: matriz quadrada (2*n,2*n)
% g: matriz quadrada (2*n,2*n)
%
%-----

function [G,dfi]=aplybc(G,H,kode,fi,n)
for j=1:(2*n),
    if kode(j) ~= 0
        for i=1:(2*n),

```

```

        ch=G(i,j);
        G(i,j)=-H(i,j);
        H(i,j)=-ch;
    end
end
end
dfi=H*fi';
dfi=dfi';

%-----
%fim da rotina

```

D.13 ROTINA REORDER

```

%-----SUBROTINA REORDER-----
%
% stores potential (fi) and flux (dfi) in the respective vectors
%
%-----

function [fi,dfi]=reorder(fi,dfi,kode,n)
for i=1:2*n,
    if kode(i) ~= 0
        ch=fi(i);
        fi(i)=dfi(i);
        dfi(i)=ch;
    end
end
end

%-----
%fim da rotina

```

D.14 ROTINA INTEREC

```

%-----SUBROTINA INTEREC-----
%
% This subroutines comput the values of the stress and displacement
% components at internal points
%
%-----

function [ds,ss]=interec(n,npixi,x,fi,dfi,ge,xnu)
ds=zeros(2*npixi,1);
ss=zeros(3*npixi,1);
for i=1:npixi,
    for j=1:n,
        [g11 g12 g22 h11 h12 h21 h22]=extinec(xi(i,:),x(j,:),x(j+1,:),ge,xnu); %
        integração numérica
        ds(2*i-1)=ds(2*i-1)+dfi(2*j-1)*g11+dfi(2*j)*g12-fi(2*j-1)*h11-fi(2*j)*h12;
        ds(2*i)=ds(2*i)+dfi(2*j-1)*g12+dfi(2*j)*g22-fi(2*j-1)*h21-fi(2*j)*h22;
        [d111 d211 d112 d212 d122 d222 s111 s211 s112 s212 s122
s222]=sigmaec(xi(i,:),x(j,:),x(j+1,:),ge,xnu);
        ss(3*i-2)=ss(3*i-2)+dfi(2*j-1)*d111+dfi(2*j)*d211-fi(2*j-1)*s111-
fi(2*j)*s211;
        ss(3*i-1)=ss(3*i-1)+dfi(2*j-1)*d112+dfi(2*j)*d212-fi(2*j-1)*s112-
fi(2*j)*s212;
    end
end

```

```

        ss(3*i)=ss(3*i)+dfi(2*j-1)*d122+dfi(2*j)*d222-fi(2*j-1)*s122-fi(2*j)*s222;
    end
end

%-----
%fim da rotina

```

D.15 ROTINA EXTINEC

```

% -----SUBROTINA EXTINEC-----
%
% Computes the G and H matrices coefficients that relate a collocation
% point with a different element using gauss quadrature.
%
% -----VARIABLES-----
%
% ng: numero pontos de Gauss
% w: pesos dos pontos de Gauss
% e: coordenadas dos pontos de Gauss
% pc: ponto de colocação
% r: distância do ponto de colocação ao ponto de integração
% L: comprimento do elemento
% D: distância do ponto de colocação ao elemento
% rd: derivada de r
% eta1,eta2: componente do normal unitario do elemento
%
%-----

function [g11,g12,g22,h11,h12,h21,h22]=extinec(pc,x1,x2,ge,xnu)
ng=4;
L=norm(x2-x1);
eta1=(x2(1,2)-x1(1,2))/L;
eta2=(x1(1,1)-x2(1,1))/L;
[w e]=gauss(ng);
xc=zeros(ng,2);
rd=zeros(ng,2);
r=zeros(1,ng);
for z=1:ng,
    xc(z,:)=e(1,z).*(x2-x1)./2+(x1+x2)./2;%mapeamento conforme [0 L]->[-1 1]
    rd(z,:)=(xc(z,:)-pc);
    r(1,z)=norm(xc(z,:)-pc);
end
aux=[r
     r];
rd=rd'./aux;
D=dist(pc,x1,x2);
de=8*pi*(1-xnu);
gaux11=((3-4*xnu).*(log(1./r))+rd(1,:).^2).*w.*L/(2*de*ge);
gaux12=rd(1,:).*rd(2,:).*w.*L/(2*de*ge);
gaux22=((3-4*xnu).*(log(1./r))+rd(2,:).^2).*w.*L/(2*de*ge);
haux11=D*((1-2*xnu)+2*rd(1,:).^2)./(r.^2*de).*w.*L;
haux12=(D*2*rd(1,:).*rd(2,:)./r+(1-2*xnu).*(eta1.*rd(2,:)-
eta2.*rd(1,:))).*w.*L./(r*de);
haux21=(D*2*rd(1,:).*rd(2,:)./r+(1-2*xnu).*(eta2.*rd(1,:)-
eta1.*rd(2,:))).*w.*L./(r*de);
haux22=D*((1-2*xnu)+2*rd(2,:).^2).*w.*L./(r.^2*de);
g11=sum(gaux11);
g12=sum(gaux12);
g22=sum(gaux22);
h11=-sum(haux11);
h12=-sum(haux12);

```

```
h21=-sum(haux21);
h22=-sum(haux22);
```

```
%-----
%fim da rotina
```

D.16 ROTINA GAUSS

```
function [w,e]=gauss(ng)
```

```
x1=-1;
x2=1;
m=(ng+1)/2;
xm=0.5*(x2+x1);
x1=0.5*(x2-x1);
ep=3.0*10^(-14);
for i=1:m
    z=cos(pi*(i-0.25)/(ng+0.5));
    p1=1;
    p2=0;
    for j=1:ng
        p3=p2;
        p2=p1;
        p1=((2*j-1)*z*p2-(j-1)*p3)/j;
    end
    pp=ng*(z*p1-p2)/(z*z-1);
    z1=z;
    z=z1-p1/pp;
    while abs(z-z1)>ep
        p1=1;
        p2=0;
        for j=1:ng
            p3=p2;
            p2=p1;
            p1=((2*j-1)*z*p2-(j-1)*p3)/j;
        end
        pp=ng*(z*p1-p2)/(z*z-1);
        z1=z;
        z=z1-p1/pp;
    end
    e(i)=xm-x1*z;
    e(ng+1-i)=xm+x1*z;
    w(i)=2*x1/((1-z^2)*pp^2);
    w(ng+1-i)=w(i);
end
```

```
%-----
%fim da rotina
```

D.17 ROTINA DIST

```

%-----SUBROTINA DIST-----
%
% x=dist(x1,x2,x3) calcula a distância do ponto xp
% até a reta formada por x1 e x2
%
%-----VARIAVEIS-----
%
% xp: coordenadas do ponto de colocação
% x1: primeiro nó do elemento a ser varrido
% x2: segundo nó do elemento a ser varrido
%
%-----

function d=dist(xp,x1,x2)
xp=[xp 0];
x2=[x2 0];
x1=[x1 0];
a=(xp-x2);
b=(x1-x2)/norm(x1-x2);
d=norm(cross(a,b));
sig=sum(cross((x1-xp),(x2-xp)));
if sig < 0
    d=-d;
end

%-----
%fim da rotina

```

D.18 ROTINA SIGMAEC

```

% -----SUBROTINA SIGMAEC-----
%
% Computes the values of the S and D matrices using gauss quadratures in
% order to compute the stresses at any internal point.
%
% -----VARIAVEIS-----
%
% ng: numero pontos de Gauss
% w: pesos dos pontos de Gauss
% e: coordenadas dos pontos de Gauss
% pc: ponto de colocação
%
%-----

function
[d111,d211,d112,d212,d122,d222,s111,s211,s112,s212,s122,s222]=sigmaec(pc,x1,x2,ge,x
nu)

ng=4; % ng de pontos de integração
L=norm(x2-x1); % comprimento do elemento
eta1=(x2(1,2)-x1(1,2))/L;
eta2=(x1(1,1)-x2(1,1))/L;
[w e]=gauss(ng);
for z=1:ng,
    xc(z,:)=e(1,z).*(x2-x1)./2 +(x1+x2)./2; % mapeamento conforme [0 L] -> [-1 1]
    rd(z,:)=(xc(z,)-pc);
    r(1,z)=norm(xc(z,)-pc); % distância ponto de colocação ponto de integração

```

```

end
aux=[r
      r];
rd=rd'./aux;
D=dist(pc,x1,x2); % distância do ponto de colocação ao elemento
de=8*pi*(1-xnu);
fa=1-4*xnu;
al=1-2*xnu;
daux111=(al*rd(1,:)+2*rd(1,:).^3).*w*L./(de*r);
daux211=(2*rd(1,:).^2.*rd(2,)-al*rd(2,)).*w*L./(de*r);
daux112=(al*rd(2,:)+2*rd(1,:).^2.*rd(2,)).*w*L./(de*r);
daux212=(al*rd(1,:)+2*rd(2,:).^2.*rd(1,)).*w*L./(de*r);
daux122=(2*rd(2,:).^2.*rd(1,)-al*rd(1,)).*w*L./(de*r);
daux222=(al*rd(2,:)+2*rd(2,:).^3).*w*L./(de*r);
saux111=(2*D./r.*(al*rd(1,)+xnu*2*rd(1,))-
4*rd(1,:).^3)+4*xnu*etal*rd(1,:).^2+al*(2*etal*rd(1,:).^2+
fa*etal)*2*ge.*w*L./(de*r.^2);
saux211=(2*D./r.*(al*rd(2,)-
4*rd(1,:).^2.*rd(2,))+4*xnu*etal*rd(1,).*rd(2,)+al*2*eta2*rd(1,:).^2-
fa*eta2)*2*ge.*w*L./(de*r.^2);
saux112=(2*D./r.*(xnu*rd(2,)-
4*rd(1,:).^2.*rd(2,))+2*xnu*(etal*rd(2,).*rd(1,)+eta2*rd(1,:).^2)+al*(2*etal*rd(
1,).*rd(2,)+eta2))*2*ge.*w*L./(de*r.^2);
saux212=(2*D./r.*(xnu*rd(1,)-
4*rd(2,:).^2.*rd(1,))+2*xnu*(etal*rd(2,).^2+eta2*rd(1,).*rd(2,))+al*(2*eta2*rd(
1,).*rd(2,)+etal))*2*ge.*w*L./(de*r.^2);
saux122=(2*D./r.*(al*rd(1,)-
4*rd(2,).^2.*rd(1,))+4*xnu*eta2*rd(1,).*rd(2,)+al*2*etal*rd(2,).^2-
fa*etal)*2*ge.*w*L./(de*r.^2);
saux222=(2*D./r.*(al*rd(2,)+xnu*2*rd(2,)-
4*rd(2,).^3)+4*xnu*eta2*rd(2,).^2+al*(2*eta2*rd(2,).^2+2*eta2)-
fa*eta2)*2*ge.*w*L./(de*r.^2);

d111=sum(daux111);
d211=sum(daux211);
d112=sum(daux112);
d212=sum(daux212);
d122=sum(daux122);
d222=sum(daux222);
s111=sum(saux111);
s211=sum(saux211);
s112=sum(saux112);
s212=sum(saux212);
s122=sum(saux122);
s222=sum(saux222);

% -----
%fim da rotina

```