DANIEL IGOR MENDOZA QUIÑONES

ALGORITMO COLABORATIVO BASEADO EM FATORAÇÃO MULTIFRONTAL QR PARA ESTIMAÇÃO DE TRAJETÓRIA DE ALVOS COM REDES DE SENSORES SEM FIO

DANIEL IGOR MENDOZA QUIÑONES

ALGORITMO COLABORATIVO BASEADO EM FATORAÇÃO MULTIFRONTAL QR PARA ESTIMAÇÃO DE TRAJETÓRIA DE ALVOS COM REDES DE SENSORES SEM FIO

Tese apresentada à Escola Politécnica da Universidade de São Paulo para obtenção do título de doutor em Ciências

São Paulo 2013 DANIEL IGOR MENDOZA QUIÑONES

ALGORITMO COLABORATIVO BASEADO EM FATORAÇÃO MULTIFRONTAL QR PARA ESTIMAÇÃO DE TRAJETÓRIA DE ALVOS COM REDES DE SENSORES SEM FIO

Tese apresentada à Escola Politécnica da Universidade de São Paulo para obtenção do título de doutor em Ciências

Área de concentração: Engenharia Mecânica

Orientador: Jun Okamoto Jr.

São Paulo 2013 Este exemplar foi revisado e alterado em relação à versão original, sob responsabilidade única do autor e com a anuência de seu orientador.

São Paulo, de de 2013

Assinatura do autor

Assinatura do orientador

FICHA CATALOGRÁFICA

Quiñones, Daniel Igor Mendoza

Algoritmo colaborativo baseado em fatoração multifrontal QR para estimação de trajetória de alvos com redes de sensores sem fio / D. I. M. Quiñones -- ed.rev. -- São Paulo, 2013. 136 p.

Tese (Doutorado) - Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia Mecatrônica e de Sistemas Mecânicos.

1. Sensoriamento Remoto; Algoritmo; Wireless I. Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia Mecatrônica e de Sistemas Mecânicos II. t.

DEDICATÓRIA

A Manuel y Detti A mi huesitos ... por todo

AGRADECIMENTOS

Nestes mais de quatro anos de dedicação nesse doutorado, recebi o apoio e suporte de muitas pessoas as quais gostaria de agradecer.

A meus pais, que sempre estão do meu lado apesar da distância e dos longos tempos que demoro para comunicar-me com eles.

A Luciana por compartilhar parte de sua vida comigo e suas críticas aos textos preliminares que se converteram depois nessa tese.

A Jose e Rejane por compartilhar sua amizade e sua casa comigo.

A Angélica pelos divertidas discussões sobre diversos temas e suas revisões ortográficas.

A Zé Carlos por sua ajuda técnica.

Finalmente, a meu orientador Jun Okamoto Jr., por toda a ajuda, suporte financeiro e infra-estrutura disponível que me brindou nessa etapa de mina vida

RESUMO

As redes de sensores sem fio (RSSF) são uma tecnologia que ganhou muita importância nos últimos anos. Dentro das diversas aplicações para essas redes, o rastreamento de alvos é considerado essencial. Nessa aplicação, a RSSF deve determinar, de forma colaborativa, a trajetória de um ou mais alvos que se encontrem dentro de sua área de cobertura. O presente trabalho apresenta um algoritmo colaborativo baseado na fatoração multifrontal QR para estimação de trajetórias de alvos com RSSF. A solução proposta está inserida no âmbito da estimação por lotes, na qual os dados são coletados pelos sensores durante a aplicação e só no final é realizada a estimativa da trajetória do alvo. Uma vez coletados os dados, o problema pode ser modelado como um sistema de equações sobredeterminado Ax = b cuja característica principal é ser esparso. A solução desse sistema é dada mediante o método de mínimos quadrados, no qual o sistema é transformado num sistema triangular superior, que é solucionado mediante substituição inversa. A fatoração multifrontal QR é ideal neste contexto devido à natureza esparsa da matriz principal do sistema. A fatoração multifrontal QR utiliza um grafo denominado árvore de eliminação para dividir o processo de fatoração de uma matriz esparsa em fatorações densas de pequenas submatrizes denominadas matrizes frontais. Mapeando a árvore de eliminação na RSSF consegue-se que essas fatorações densas sejam executadas pelos nós sensoriais que detectaram o alvo durante seu trajeto pela rede. Dessa maneira, o algoritmo consegue realizar a fatoração da matriz principal do problema de forma colaborativa, dividindo essa tarefa em pequenas tarefas que os nós de sensoriais da rede possam realizar.

Palavras-chave: Sensoriamento Remoto; Algoritmo; Wireless.

ABSTRACT

Wireless Sensor Networks (WSN) is a technology that have gained a lot of importance in the last few years. From all the possible applications for WSN, target tracking is considered essential. In this application, the WSN has to determine, in a collaborative way, the trajectory of one or more targets that are within the sensing area of the network. The aim of this document is to present a collaborative algorithm based on multifrontal QR factorization for the solution of the target trajectory estimation problem with WSN. This algorithm uses a batch estimation approach, which assumes that all sensing data are available before the estimation of the target trajectory. If all the observations of the target trajectory is available, the problem can be modeled as an overdetermined system of equations Ax = b where A is sparse. This system of equations is solved by least squares method. The multifrontal QR factorization uses a tree graph called elimination tree to reorganize the overall factorization of a sparse matrix into a sequence of partial factorizations of dense smaller matrices named frontal matrices. By mapping the elimination tree into the WSN, the sensor nodes that observed the target can factorize the frontal matrices. In this manner, the WSN factorizes the matrix A in a collaborative way, dividing the work in small tasks that the sensor nodes could execute.

Keywords: Distributed estimation; Wireless sensor networks

LISTA DE ILUSTRAÇÕES

Figura 1 -	Processo de fatoração multifrontal	20
Figura 2 -	RSSF espalhada na cidade para rastreamento de gases químicos. Ex- traído de (ZHAO; GUIBAS, 2004)	23
Figura 3 -	Nós sensoriais de diversos tamanhos	25
Figura 4 -	Cenários do problema de rastreamento de alvos	33
Figura 5 -	Configurações dos sistemas de fusão de dados	38
Figura 6 -	Estruturas de rede das RSSF	39
Figura 7 -	Alternativas para o desenvolvimento de algoritmos de estimação (Adap- tado de (HALL, 2004))	42
Figura 8 -	Configurações das RSSF para realizar rastreamento de alvos mediante estimação sequencial. (Adaptado de (ZHAO; GUIBAS, 2004))	50
Figura 9 -	Mapeamento e localização simultânea distribuída. (Extraído de (DEL- LAERT; KIPP; KRAUTHAUSEN, 2005))	60
Figura 10 -	Representação gráfica de matrizes	65
Figura 11 -	Determinação da estrutura da matriz $A^T A$ para a matriz de exemplo	
E		00 67
Figura 12 -	Processo de eliminação de variaveis e <i>línea graph</i> de $A^{-}A^{-}$	60
Figura 13 -		70
Figura 14 -	Matriz A ordenada \dots	70 72
Figura 15 -	Fatoração multificantel QR da matriz de exemplo A	77
Figura 15 -	Fatoração multifrontal QR da matriz de exemplo A (continuação).	74
Figura 15 -	Fatoração multificantel QR da matriz de exemplo A (continuação).	75
Figura 15 -	Patoração multinontal QA da matriz de exemplo A (continuação)	70
Figura 17	Matriz H na qual o algoritmo do mínimo grau não ó ótimo	70
Figure 10	Poerdonação de columos utilizando ordenação nor dissocção	70
Figure 10	Exemple de utilização de ordenação por dissecção :	19
Figura 19 -	figura 16	80
Figura 20 -	Matriz tridiagonal na ordem natural, fator de cholesky e árvore de	
	eliminação \ldots \ldots \ldots \ldots \ldots	81
Figura 21 -	Matriz tridiagonal ordenada mediante o algoritmo de mínimo grau	81
Figura 22 -	Matriz tridiagonal ordenada mediante ordenação por dissecção	82

Figura 23 -	Rede bayesiana correspondente ao problema de estimação de trajetória
	com redes de sensores sem fio
Figura 24 -	Estrutura da matriz $A^T A$ para o problema de estimação de trajetórias 86
Figura 25 -	Cenário exemplo do problema de estimação de trajetória com uma rede
	de sensores sem fio
Figura 26 -	Árvore de eliminação para cenário estudado
Figura 27 -	Matriz principal do problema ordenada mediante o algoritmo de orde-
	nação por dissecção
Figura 28 -	Grupo de sensores para cada vértice da árvore $T(A')$ 91
Figura 29 -	Duas possíveis escolhas de sensores líderes de grupo que minimizam a
	comunicação da rede
Figura 30 -	Processo de coleta de dados dos líderes de grupo
Figura 31 -	Matrizes ampliadas correspondentes ao cenário estudado 95
Figura 32 -	Matriz ampliada $(A' \mid b)$ ordenada para visualizar as matrizes $A'[x_i]$ 96
Figura 33 -	Matrizes $A_{modelo}[x_i]$ e $A'_{modelo}[x_i]$ para o cenário estudado 97
Figura 34 -	Submatriz $K_{modelo}[x_i]$
Figura 35 -	Existência da matriz $ar{A}'_{modelo}[x_i]$
Figura 36 -	Fatoração multifrontal QR da matriz feita pelos sensores da rede no
	cenário estudado
Figura 36 -	Fatoração multifrontal QR da matriz feita pelos sensores da rede no
	cenário estudado (Continuação)
Figura 36 -	Fatoração multifrontal QR da matriz feita pelos sensores da rede no
	cenário estudado (Continuação)
Figura 36 -	Fatoração multifrontal QR da matriz feita pelos sensores da rede no
	cenário estudado (Continuação)
Figura 36 -	Fatoração multifrontal QR da matriz feita pelos sensores da rede no
	cenário estudado (Continuação)
Figura 36 -	Fatoração multifrontal QR da matriz feita pelos sensores da rede no
	cenário estudado (Continuação)
Figura 36 -	Fatoração multifrontal QR da matriz feita pelos sensores da rede no
	cenário estudado (Continuação)
Figura 37 -	Matriz R' solução da fatoração multifrontal QR da matriz A' 106
Figura 38 -	Exemplo de trajetória de alvo gerada pelo modelo dinâmico dos alvos . 109
Figura 39 -	Modelo de sensoriamento com cobertura probabilística
Figura 40 -	RSSF utilizada no simulador

Figura 41 -	Tabela de Observação de Estados para os primeiros 20 estados obser-
	vados pela RSSF
Figura 42 -	Matriz A que a RSSF deve fatorar $\ldots \ldots \ldots$
Figura 43 -	Matriz $A^{T}A$ da equação normal correspondente a simulação do problema 114
Figura 44 -	Matriz principal do problema reordenada
Figura 45 -	Árvore de eliminação correspondente à matriz A'
Figura 46 -	Trajetória real e observações dos nós sensoriais
Figura 47 -	Trajetória estimada mediante solução colaborativa
Figura 48 -	Trajetoria estimada mediante a solução centralizada
Figura 49 -	Trajetórias real, colaborativa e centralizada. Na figura vemos que as
	trajetórias estimadas centralizada e colaborativa encontram-se so-
	brepostas e são indistinguíveis devido ao fato de que o resultado da
	estimativa para ambas as soluçoes foram idênticas, como é mos-
	trado na tabela 2
Figura 50 -	Erro RMS (a) Solução colaborativa; (b) Solução centralizada $\ .\ .\ .$. 123
Figura 51 -	Processo de cálculo do tempo de fatoração da matriz principal do pro-
	blema

LISTA DE TABELAS

Tabela 1 -	Tabela de estados observados 8	39
Tabela 2 -	Comparação dos estados estimados com a solução colaborativa e cen- tralizada	22
Tabela 3 -	Comparação do tempo de fatoração das soluções colaborativa e cen- tralizada	25
Tabela 4 -	Quantidade de memória máxima e mínima para as matrizes frontais e	
	de atualização	26

LISTA DE ABREVIATURAS E SIGLAS

BLE	Best Linear Estimator
BLUE	Best Linear Unbiased Estimator
CHF	Channel Filter
CSIP	Collaborative Signal and Information Processing
CSIRO	Commonwealth Scientific and Industrial Research Organization
DSP	Digital Signal Processor
DWNA	Discrete White Noise Acceleration
EIF	Extended Information Filter
EKF	Extended Kalman Filter
RMS	Root Mean Square
full SLAM	full Simultaneous Localization and Mapping
WLS	Weigthed Least Squares
flops	floating point operations
GPS	Global Positioning System
IF	Information Filter
IP	Internet Protocol
KCF	Kalman-Consensus Filter
KF	Kalman Filter
MANET	Mobile ad-hoc Network
MAP	Maximum A Posteriori
MEMS	Microelectromechanical Systems
MMSE	Minimum Mean Square Error
MSE	Mean Square Error
OLS	Ordinary Least Squares
PDF	probability density function
RF	Radio Frequency
RSSF	Redes de Sensores sem Fio
UIF	Unscented Information Filter
UKF	Unscented Kalman Filter

SUMÁRIO

1 INTRODUÇÃO 1.1 MOTIVAÇÃO 1.2 SOLUÇÃO PROPOSTA: EST	15
, RAÇÃO MULTIFRONTAL	QR
2 REDES DE SENSORE	S SEM FIO 22
2.1 VANTAGENS DAS REDES D	E SENSORES SEM FIO
2.2 CARACTERÍSTICAS DAS RE	EDES DE SENSORES SEM FIO
2.3 DESAFIOS E TEMAS ABER	TOS A PESQUISA
3 RASTREAMENTO DE	E ALVOS COM REDES DE SENSO-
RES SEM FIO	
3.1 FUSÃO DE DADOS EM SIST	TEMAS MULTISENSORIAIS
3.2 ARQUITETURAS DE FUSÃO) DE DADOS E TOPOLOGIAS DE REDE DAS RSSF 35
3.3 MÉTODOS DE ESTIMAÇÃO	9
3.3.1 Metodologia do Process	amento
3.3.1.1 Estimação Sequencial	
3.3.1.2 Estimação por Lotes (Ba	tch Estimation)
3.4 INFERÊNCIA DISTRIBUÍDA	APLICADA AO PROBLEMA DE LOCALIZAÇÃO
E MAPEAMENTO COM	MÚLTIPLOS ROBÔS UTILIZANDO FATORAÇÃO
MULTIFRONTAL QR .	
4 FATORAÇÃO MULTIF	RONTAL <i>QR</i>
4.1 FATORAÇÃO SIMBÓLICA	
4.2 DETERMINAÇÃO DA ÁRVO	RE DE ELIMINAÇÃO
4.3 FATORAÇÃO NUMÉRICA	
4.4 PARALELISMO NA FATORA	ÇÃO
5 ALGORITMO COLAB	ORATIVO 83
5.1 CONSIDERAÇÕES PRELIMI	NARES
5.2 FORMULAÇÃO DO PROBLE	MA
5.3 CENÁRIO EXEMPLO DO PR	OBLEMA
5.4 COLETA DE DADOS	
5.5 FATORAÇÃO MULTIFRONT	AL QR
5.5.1 Determinação da árvore	de eliminação
5.5.2 Formação de grupos	

5.5.3 Fatoração numérica da matriz	92
5.5.3.1 Formação das matrizes frontais F_{x_i}	93
5.5.3.2 Fatoração da matriz A'	98
5.6 SOLUÇÃO DO SISTEMA UTILIZANDO SUBSTITUIÇÃO INVERSA 1	.05
6 SIMULAÇÕES E RESULTADOS	.07
6.1 MODELO DINÂMICO DOS ALVOS	.07
6.2 MODELO DOS NÓS SENSORIAIS	.09
6.3 CONFIGURAÇÃO DA REDE DE SENSORES SEM FIO	.11
6.4 SIMULAÇÃO DA ESTIMAÇÃO DE TRAJETÓRIAS COLABORATIVA COM RSSF	
BASEADO EM FATORAÇÃO MULTIFRONTAL QR	.11
6.4.1 Parâmetros das simulações	.11
6.4.2 Tabela de observação de estados	.13
6.4.3 Matriz principal A do problema de estimação de trajetórias 1	.13
6.4.4 Ordenação de colunas e árvore de eliminação	.13
6.4.5 Estimação distribuída da trajetória	.17
6.4.5.1 Estimação de trajetórias	.17
6.4.5.2 Erro quadrático médio	.23
6.4.6 Tempo na fatoração	.24
6.4.7 Mémoria utilizada	.25
7 CONCLUSÕES	.27
REFERÊNCIAS	.29

1 INTRODUÇÃO

Os avanços na miniaturização de sistemas microeletromecânicos (MEMS – *Microelectromechanical Systems*), na fabricação de semicondutores e nas tecnologias de comunicações e redes de dados tem possibilitado o surgimento de uma nova classe de sistemas denominados Redes de Sensores sem Fio (RSSF) (ZHAO; GUIBAS, 2004). Esses sistemas são constituídos por um grande número de pequenos módulos de hardware (denominados nós sensoriais) de baixo custo, dotados de um ou mais sensores, com capacidade de processamento e comunicação. Os nós são interconectados em rede e espalhados nos mais diversos ambientes para, colaborativamente, monitorar ou coletar dados de algum fenômeno de interesse.

O campo de aplicação das RSSF cresce continuamente. Alguns exemplos são: Monitoração ambiental (SZEWCZYK et al., 2004), sistemas de controle de tráfego inteligente (BATHULA et al., 2009), sistemas de vigilância militar (BOKAREVA et al., 2006), casas e edifícios inteligentes (FENG et al., 2008), monitoração subaquática (POMPILI; MELODIA; AKYILDIZ, 2006) e supervisão de linha de manufatura (ZHUANG; GOH; ZHANG, 2007), dentre outros. Estima-se que, no futuro, será possível monitorar o mundo com essas redes, o que modificará nossa forma de entendê-lo e interagir com ele (ZHAO; GUIBAS, 2004).

Apesar do grande interesse que tem despertado essa nova área, ainda existem diversos problemas a serem resolvidos ao desenvolver um sistema RSSF. Esses problemas aparecem devido às características especiais dos nós sensoriais (limitações em processamento, memória e bateria dos nós), às limitações no fornecimento de energia nas aplicações e às limitações de largura de banda na comunicação sem fio. Esses problemas demandam novos desenvolvimentos principalmente em duas áreas: tecnologias de rede e processamento de sinais e informação (MA, 2008).

Os desenvolvimentos de tecnologias de rede focam-se em problemas tais como estabelecimento automático da topologia da rede de sensores, controle e entrega de informação mediante o desenvolvimento de protocolos de roteamento, encriptação de dados, assim como administração do canal de comunicação de radiofrequência. Já as pesquisas relacionadas com processamento de sinais e informação colaborativo visam desenvolver técnicas que permitam processar, agregar, armazenar e/ou representar dados e informação em forma distribuída de maneira a utilizar melhor os limitados recursos da rede.

A contribuição proposta neste trabalho encontra-se inserida na área de processamento de sinais e informação colaborativa. O problema abordado é o rastreamento de alvos com uma rede de sensores sem fio distribuída. O rastreamento de objetos e eventos é considerado uma capacidade essencial de uma RSSF (MA, 2008; YICK; MUKHERJEE; GHOSAL, 2008; ZHAO; GUIBAS, 2004). O problema de rastreamento de alvos consiste em realizar a estimativa da

posição de um ou mais alvos móveis (veículos, pessoas, aviões) enquanto eles se encontram dentro do alcance de sensoriamento da rede.

A solução proposta nessa tese para o problema abordado é o desenvolvimento de um algoritmo colaborativo baseado na fatoração multifrontal QR, que permite o processamento distribuído entre os nós da rede de maneira a minimizar a carga de processamento entre os nós atingindo melhor eficiência em termos de energia, banda de comunicação, dentre outros. O objetivo da nossa solução é estimar a trajetória completa dos alvos que passaram dentro da área de cobertura da RSSF. Esse problema pode ser modelado como um grande sistema de equações superdeterminado. A solução desse sistema vem da teoria de mínimos quadrados que transforma o sistema de equações em um sistema triangular superior . A fatoração multifrontal QR é ideal neste contexto devido à natureza esparsa da matriz principal do sistema. A fatoração multifrontal QR utiliza um grafo denominado árvore de eliminação para dividir o processo de fatoração de uma matriz esparsa em fatorações densas de pequenas submatrizes denominadas matrizes frontais. Explorando a estrutura do problema e mapeando a árvore de eliminação na RSSF, essas fatorações densas podem ser executadas pelo conjunto de nós sensoriais que detectaram o alvo durante seu trajeto pela rede. Dessa maneira, o algoritmo aqui proposto consegue dividir o trabalho da fatoração da grande matriz principal do sistema de equações em pequenas tarefas que os nós sensoriais da rede possam realizar, procurando minimizar a quantidade de mensagens e de processamento local de cada nó.

Este documento está organizado da seguinte maneira: ainda nesse capítulo, na seção 1.1, apresentamos a motivação para os estudos realizados neste trabalho e na seção 1.2 descreve-se um resumo da solução proposta. No capítulo 2 apresenta-se uma introdução às RSSF. No capítulo 3 revisam-se os métodos utilizados para realizar rastreamento de alvos com RSSF. No capítulo 4 revisa-se o método de fatoração multifrontal QR. No capítulo 5 apresenta-se o algoritmo colaborativo para estimação de trajetórias baseado em fatoração multifrontal QR. No capítulo 6 apresentam-se as simulações e resultados da solução proposta. Finalmente, no capítulo 7 apresentam-se as conclusões e os trabalhos futuros a realizar.

1.1 MOTIVAÇÃO

Os nós sensoriais de uma RSSF, devido a seu pequeno tamanho e baixo custo, podem ser espalhados em grandes quantidades e em lugares de difícil acesso, formando grandes sistemas de sensoriamento distribuído. Porém, esses sistemas apresentam desafios e restrições diferentes aos encontrados em sistemas sensoriais distribuídos tradicionais. Os nós sensoriais das RSSF apresentam limitações que incluem: pouca capacidade de processamento, quantidade limitada de memória, largura de banda de comunicação reduzida e alcance de comunicação e sensoriamento limitados (TUBAISHAT; MADRIA, 2003). Além disso, em uma aplicação típica, cada nó sensorial deve trabalhar autonomamente e energizado mediante bateria interna. A falta de fonte de energia para alimentar os nós é um problema sério neste tipo de redes.

Por conta dessas restrições, se faz necessário desenvolver algoritmos que permitam aos nós sensoriais processarem a informação colaborativamente. Processamento de sinais e informação colaborativa (CSIP – *Collaborative Signal and Information Processing*) é o nome utilizado para designar os desenvolvimentos que visam a seleção ou agrupamento de sensores para realizar tarefas de processamento de informação (ZHAO; GUIBAS, 2004). Essas técnicas são necessárias, principalmente, devido aos seguintes fatores:

- Apesar dos nós sensoriais terem capacidade de sensoriamento, processamento e comunicação com outros nós, essas capacidades são limitadas. Isso faz necessário a colaboração entre os nós para realizar tarefas complexas que um só nó não poderia realizar.
- Na maioria das aplicações das RSSF existe a necessidade de processar grandes quantidades de informação. Em um sistema centralizado, essa informação seria coletada num servidor onde os dados seriam processados. Isso implica em transmitir todos esses dados através da rede até o centro de processamento. Para uma RSSF isso representa um gasto ineficiente da limitada largura de banda que possui. É sabido que, se cada nó sensorial de uma rede transmite dados ao mesmo tempo, a taxa de transferência por nó tende a zero na medida em que o número de nós da rede aumenta, independentemente da qualidade de roteamento ou da eficiência na transmissão de dados (ZHAO; GUIBAS, 2004). Processar os dados localmente ajuda a diminuir a quantidade de mensagens a serem transmitidas pela rede.
- Uma das maiores restrições dos nós sensoriais é a falta de energia. É bem conhecido o fato de que a transmissão de dados é a operação que consome mais energia, sendo que cada bit transmitido gasta a energia equivalente a 1000 instruções de processador (CULLER; ESTRIN; SRIVASTAVA, 2004). Portanto, transmitir os dados a longas distâncias limita a vida útil da rede. O processamento colaborativo eficiente entre os nós é necessário para aumentar o tempo de operação da rede.

Desenvolvimentos em processamento colaborativo de informação com RSSF tem sido utilizados em diversas aplicações. Uma das mais importantes é o rastreamento de alvos móveis porque expõe os temas mais importantes relacionados ao processamento de informação colaborativo (ZHAO; GUIBAS, 2004).

Nessa aplicação, o objetivo da rede de sensores é estimar o estado do alvo (posição, velocidade, direção) a cada momento, a partir das observações realizadas pelos sensores. Devido as restrições inerentes as RSSF, existe a necessidade de desenvolver algoritmos descentralizados para essa aplicação.

Diversas técnicas de rastreamento de alvos desenvolvidas para sistemas centralizados tem sido adaptadas à redes de sensores, dentre elas temos: estimação por máxima verossimilhança (SHENG; HU, 2005; ZHAO; NEHORAI, 2007), filtro de Kalman (KHAN; MOURA, 2008), métodos gráficos probabilísticos (SHI; TAN; ZHAO, 2009), filtro de partículas (SHENG; HU; RAMANATHAN, 2005). Ainda assim, existem diversos temas de estudo em aberto, como por exemplo: estimação do estado do alvo quando se tem incerteza da origem, propagação eficiente da estimativa entre os sensores da rede, formas de agrupamento dos sensores baseado na tarefa e recursos da rede, administração da comunicação entre os sensores, formas de compartilhamento da informação entre sensores, etc.

Dentre todos os algoritmos propostos para rastreamento com redes de sensores, o filtro de Kalman (BAR-SHALOM; KIRUBARAJAN; LI, 2002) é um dos mais utilizados. Esse algoritmo pode ser descentralizado eficientemente (DURRANT-WHYTE, 2000) e com ele consegue-se bons resultados de estimação.

Quando se utiliza o filtro de Kalman, o problema é modelado utilizando dois tipos de equações, que aparecem a cada intervalo de tempo (STRANG, 1986): a equação do modelo dinâmico

$$x_i = F_i x_{i-1} \tag{1}$$

e a equação de medição

$$z_k = H_k x_i \tag{2}$$

Se, durante o tempo que dure a aplicação, coletamos e combinamos essas equações num sistema único o resultado é um grande sistema de equações retangular superdeterminado do tipo Ax = b onde a principal característica é que a matriz principal A é esparsa (DELLAERT; KAESS, 2006).

A posibilidade de explorar a esparsidade antes mencionada assim como a necessidade de algoritmos descentralizados e colaborativos para RSSF servem de motivação para os estudos apresentados neste trabalho.

1.2 SOLUÇÃO PROPOSTA: ESTIMAÇÃO COLABORATIVA BASEADA NA FATORAÇÃO MULTIFRONTAL QR

A solução proposta neste trabalho explora a característica esparsa do sistema de equações Ax = b correspondente ao problema de rastreamento. A solução ao sistema sobredeterminado Ax = b é o melhor vetor \hat{x} , solução da seguinte expressão:

$$\hat{x} = \min \left\| Ax - b \right\|_2, \ A \in \mathbb{R}^{m \times n}, \ b \in \mathbb{R}^m \tag{3}$$

onde $\|\cdot\|_2$ denota norma Euclidiana do vetor. A expressão 3 é o problema de mínimos quadrados linear. A solução ao problema, o método de mínimos quadrados, foi desenvolvido por Gauss em 1795 e publicado no seu livro *Theoria Motus Corporum Coelestium* em 1809 (SORENSON, 1970). Gauss demonstrou que o vetor \hat{x} determinado mediante o método de mínimos quadrados é o melhor estimador não viciado, sem assumir que a distribuição dos erros aleatórios fosse gaussiana ou de qualquer tipo (BJORCK, 1996).

O método de mínimos quadrados utiliza eliminação de Gauss sobre a matriz principal A para transformá-la em uma matriz triangular superior, para depois determinar \hat{x} por substituição inversa. Uma das formas de realizar isso é mediante a decomposição ortogonal da matriz. Em outras palavras, fazer A = QR o que é conhecido como fatoração QR (GOLUB; LOAN, 1996), onde Q é uma matriz ortogonal e R é uma matriz triangular superior.

Devido ao fato de que A é esparsa, o problema pode ser solucionado de forma colaborativa. O método multifrontal pertence ao grupo de algoritmos conhecidos como métodos diretos (*Direct Methods*) porque trabalham diretamente na matriz A (STRANG, 1986). O método utiliza uma estrutura denominada árvore de eliminação para guiar a fatoração da matriz de forma paralela. A árvore de eliminação da matriz A é um grafo em forma de árvore, em que cada vértice corresponde a uma coluna da matriz (ou a uma variável do sistema de equações) e a estrutura do grafo mostra a relação que existe entre as colunas. Uma das propriedades mais importantes da árvore é que os vértices que pertencem a subárvores disjuntas podem ser eliminados em paralelo. Para realizar a fatoração em paralelo, o método multifrontal utiliza dois tipos de matrizes: a matriz frontal F_i e a matriz de atualização U_i . O método associa, a cada vértice da árvore, uma matriz F_i . O fluxo do cálculo vai das folhas até a raiz da árvore, a cada passo do cálculo realiza-se a fatoração QR da matriz F_i correspondente a cada vértice i, o que dá como resultado a linha r_{i*} da matriz triangular superior R e uma matriz de atualização U_i , que será utilizada pelo nó pai do vértice. Seguindo a propriedade da árvore, as matrizes frontais que se encontram em subárvores disjuntas podem ser fatoradas em paralelo. No nível seguinte da árvore, as matrizes frontais correspondentes aos vértices pais são formadas a partir



Figura 1 - Processo de fatoração multifrontal

de partes da matriz A combinadas com as matrizes U_k correspondentes aos filhos do vértice. O algoritmo continua até chegar à raiz da árvore. A figura 1 ilustra o processo de fatoração multifrontal para uma matriz A retangular.

Neste trabalho apresentamos uma solução colaborativa para a determinação da trajetória de um alvo com redes de sensores sem fio baseada na fatoração multifrontal QR. Nessa solução a idéia central é o mapeamento da árvore de eliminação na RSSF. O resultado desse mapeamento é que, para cada vértice da árvore, vamos ter um grupo de nós sensoriais que realizaram a fatoração multifrontal QR. Em cada grupo se escolhe um nó líder que receberá as observações do estado i do alvo que cada nó sensorial do grupo realizou. Com essa informação, o nó líder do grupo formará a matriz F_i e fatorará essa matriz para determinar a linha $r_{i,*}$ e a matriz de atualização U_i . A linha $r_{i,*}$ será enviada ao sorvedouro e a matriz U_i ao líder do grupo correspondente ao pai do vértice i. Esse procedimento prosseguirá até o grupo do vértice raiz da árvore. Finalizada a fatoração, o sorvedouro determinará o valor do vetor de estados \hat{x} mediante substituição inversa.

O algoritmo proposto se ajusta bem à solução do problema de estimação de trajetórias com RSSF porque:

- A solução é distribuída e colaborativa. Distribuída porque a tarefa de fatoração da matriz principal do problema é distribuida entre os nós que observaram o alvo. Colaborativa porque para realizar a fatoração é necessário que os nós compartilhem entre eles seus resultados parciais. Essas características são muito importante para o desenvolvimento de aplicações para RSSF já que processar a informação dentro da rede consome menos energia do que transmitir os dados a um centro de fusão centralizado;
- Durante todo o processo de estimação em nenhum momento os nós sensoriais precisam enviar os dados observados ao sorvedouro. Os dados observados são processados localmente dentro dos grupos de nós sensoriais correspondentes aos nós da árvore de eliminação. A única informação que é transmitida através da rede, entre grupos ou ao sorvedouro, são fatorações parciais da matriz A. Isso contribui na redução do gasto de energia e no prolongamento da vida útil da rede;
- A estimativa de toda a trajetória dos alvos é calculada somente no final da aplicação ou a pedido do usuário final. Portanto, não é necessário a troca de mensagens entre os nós sensoriais durante toda a aplicação o que deixa a RSSF livre para realizar outras tarefas se for necessário.

2 REDES DE SENSORES SEM FIO

Originada a partir dos avanços nas áreas de comunicação sem fio, sistemas MEMS e semicondutores, às RSSF são consideradas atualmente uma das mais importantes tecnologias do século XXI (CHONG; KUMAR, 2003).

Uma RSSF é um conjunto de microdispositivos sensoriais autônomos que colaboram entre si para formar uma rede sensorial. Esses microdispositivos, denominados nós sensoriais, possuem capacidade de processamento em CPU local, pequena quantidade de memória, bateria interna, módulo de rádio (transreceptor) para comunicação com outros nós e 1 ou mais sensores de diferentes tipos como por exemplo: sensores de temperatura, intensidade de luz, umidade, pressão, sonares, câmeras.

Devido à autonomia, tamanho e baixo custo dos nós sensoriais, as RSSF podem ser utilizadas em quase qualquer entorno físico. Desta forma, pode-se criar pequenas redes com 10 a 100 nós para monitorar: ambientes urbanos (casas, prédios, estradas), ambientes industriais (fábricas, refinarias) ou ambientes de difícil acesso (vulcões, regiões subaquáticas) assim como grandes redes, literalmente espalhando milhares de nós (figura 2) sobre grandes extensões geográficas (cidades, estados, Amazônia).

O grande potencial desta tecnologia está no tipo de informação que ela provê. As RSSF conectam o usuário final diretamente à variável ou evento de interesse. A informação obtida é mais dinâmica e em tempo real do que as oferecidas por outros serviços de informação (Internet). Além disso, os dados coletados pela rede podem ser analisados, comparados, processados ou agregados localmente pelos nós sensoriais. Isso dá ao usuário final a possibilidade de realizar diferentes consultas sobre o mesmo evento, ao que a rede responde proporcionando-lhe informações mais detalhadas e adequadas às suas necessidades.

São muitas as aplicações em que se pode utilizar uma RSSF. Essas aplicações podem ser classificadas em 2 categorias: monitoração e rastreamento (YICK; MUKHERJEE; GHOSAL, 2008). Uma RSSF pode ser utilizada para monitorar espaços, objetos ou a interação entre objetos (CULLER; ESTRIN; SRIVASTAVA, 2004). Entre as aplicações de monitoração estão incluídas: monitoração ambiental; sistemas de vigilância militar; casas e edifícios inteligentes; monitoração subaquática; supervisão de linhas de manufatura; diagnóstico médico; etc. Entre as aplicações de rastreamento estão: sistemas de controle de tráfego inteligente; rastreamento de animais; sistemas logísticos inteligentes; etc.



Figura 2 - RSSF espalhada na cidade para rastreamento de gases químicos. Extraído de (ZHAO; GUIBAS, 2004)

2.1 VANTAGENS DAS REDES DE SENSORES SEM FIO

Uma RSSF é, principalmente, um sistema sensorial distribuído cujos componentes são autônomos. Trabalhar com esse tipo de sistema oferece diversas vantagens em comparação com um sistema centralizado de um único sensor:

- Maior cobertura: Uma RSSF distribuída na área do fenômeno de interesse incrementa a cobertura do sistema de monitoração (AGRE; CLARE, 2000). Além disso, quando a localização do fenômeno é desconhecida, utilizar múltiplos nós sensoriais permite ficar mais perto do fenômeno e superar obstáculos no ambiente que possam obstruir o sensoriamento do único sensor (ESTRIN et al., 2001).
- Robustez: Uma RSSF é muito mais robusta ante falhas de enlace ou de nós sensoriais devido à redundância obtida pela grande quantidade de nós utilizados nas aplicações (ZHAO; GUIBAS, 2004).
- Maior qualidade na informação obtida: Devido a maior quantidade de sensores, um sistema multisensorial distribuído consegue coletar maior quantidade de dados. Esses dados podem corresponder à observação de um evento desde diferentes posições e/ou, no caso de utilizar sensores de diferentes tipos, a diversos aspectos do evento. Uma RSSF

utiliza técnicas de fusão de dados ou processamento colaborativo de sinais para entregar ao usuário final informação mais detalhada, exata e de maior qualidade do que um sistema centralizado tradicional possa entregar (MARTINCIC; SCHWIEBERT, 2005).

 Escalabilidade: As soluções descentralizadas desenvolvidas para RSSF fazem que o sistema seja escalável devido ao fato de que eliminam as limitações impostas pelo gargalo computacional existente nos centros computacionais centralizados ou a falta de largura de banda de comunicação (DURRANT-WHYTE, 2000).

2.2 CARACTERÍSTICAS DAS REDES DE SENSORES SEM FIO

As RSSF tem características que as fazem diferentes de qualquer outro sistema de monitoração distribuído conhecido atualmente.

O paradigma das RSSF estabelece que os nós sensoriais são pequenos, com recursos limitados e muito baratos.

Em relação ao tamanho, uma RSSF pode ser implementada utilizando nós sensoriais de qualquer tamanho (figura 3). Porém, para aproveitar o potencial de aplicações que esses sistemas oferecem, considera-se que eles devam ser pequenos, podendo chegar a ser micro ou nanodispositivos. Com o grau de miniaturização atual, um exemplo de objetivo comercial já estabelecido é desenvolver um sistema sensorial completo, baseado em MEMS, com um volume de $1 mm^3$ (SOHRABY DANIEL MINOLI, 2007; CHONG; KUMAR, 2003).

O hardware dos nós sensoriais (também conhecidos como *motes*), esta constituído basicamente de 5 componentes: CPU; memória; sensores; bateria ou fonte de energia própria e módulo de rádio.

O nó sensorial conta com uma CPU para processar os dados obtidos pelos sensores. Já foram desenvolvidos nós sensoriais baseados em microprocessadores muito potentes, como o Intel ARM de 32 bits a 200 MHz de frequência de operação (WENTZLOFF et al., 2004), mas a tendência é reduzir o consumo de potência e o preço utilizando CPUs menos potentes e mais lentas. Atualmente é comum encontrar nós com microcontroladores de 8 bits e frequência de operação de 8 MHz, como o Mica2Dot e o IRIS da MEMSIC¹, e o Fleck da *Commonwealth Scientific and Industrial Research Organization* (CSIRO)². Além disso, estudos estão sendo realizados para desenvolver nós baseados em processador digital de sinal (DSP – *Digital Signal Processor*) (WANG; CHANDRAKASAN, 2002), que poderiam trabalhar em redes de sensores sem fio multimídia.

¹http://www.memsic.com/

²http://research.ict.csiro.au/



(a) Radar atmosférico do projeto CASA (JUNYENT et al., 2010)



(b) Telosb, nó sensorial da empresa MEMSIC. Ele é muito utilizado em laboratórios de pesquisa sobre RSSF com quase o comprimento de uma bateria AA.



(c) Mica2Dot, outro nó sensorial da MEMSIC. Ele é do tamanho de uma moeda de 25 centavos de dólar

Figura 3 - Nós sensoriais de diversos tamanhos

A memória dos nós sensoriais é limitada. Por exemplo, o TelosB da MEMSIC, comumente utilizado em *testbeds* de pesquisa, tem memória de programa flash de 48 KB, memória de dados flash serial de 1 MB, 10 KB de RAM e memória de configuração EEPROM de 16 KB. Alguns nós contam com memória extra não volátil para armazenar dados que poderiam ser coletados manualmente pelo usuário. Outros possuem interface para cartão de memória SD, porém essa memória consome muita potência o que limita seu uso nas aplicações.

Os sensores são os dispositivos com os quais o nó sensorial interage com o mundo. Atual-

mente, a maioria dos motes já vem com alguns sensores embutidos (temperatura, intensidade de luz, acelerômetros, microfones, etc.) Além disso, contam com uma ou mais interfaces digitais, portas seriais (RS232, USB, etc.) e/ou conversores analógico-digital. Essas interfaces lhes permitem conectar-se a vários tipos diferentes de sensores ao mesmo tempo, com saídas digitais tanto quanto analógicas, convertendo o nó sensorial em um pequeno centro de processamento multisensorial.

A longevidade da rede depende da vida útil dos seus nós sensoriais. Nas aplicações concebidas para RSSF, espera-se que a rede trabalhe em lugares onde exista pouca ou nenhuma infraestrutura, lugares de difícil acesso ou onde exista algum perigo para o usuário final. Adicionalmente, as aplicações podem requerer que a rede trabalhe durante semanas, meses ou anos e dependendo da quantidade de nós e do ambiente de trabalho da rede, a troca de baterias pode ser uma tarefa inviável. Nessas condições, a bateria dos nós sensoriais constitui o recurso mais importante do sistema.

O próprio nó sensorial é desenvolvido levando em consideração esse recurso. Os microcontroladores utilizados podem chegar a consumir cerca de 1 miliWatt em modo de operação e 1 microWatt em modo *standby*. ADCs tem sido desenvolvidos para ter um perfil de consumo similar ao da CPU (CULLER; ESTRIN; SRIVASTAVA, 2004). Memórias flash são utilizadas para reduzir o consumo de potência da CPU na manutenção dos dados na memória RAM. Adicionalmente, todos os circuitos do nó podem ser desligados e ligados só se for necessário. Alguns nós incluem um módulo de colheita de energia (*Energy Harvesting*), que permite obter energia do ambiente, seja fotovoltaica, vibração mecânica ou termoelétrica (NIYATO et al., 2007), para carregar a bateria ou produzir a potência de trabalho dos módulos do nó.

O módulo de rádio dos nós lhes permite estabelecer a rede de comunicação do sistema. Essa capacidade é uma das grandes vantagens das RSSF, já que permite sua utilização em qualquer ambiente; basta colocar os sensores e inicializar o sistema. Porém, a utilização de rádiofrequência (RF – *Radio Frequency*) para comunicação tem um custo. A largura de banda do canal é limitada e compartilhada por todos os nós. Além disso, é bem sabido que a comunicação sem fio é a operação que mais consome a energia do nó. A energia requerida para transmitir os dados incrementa rapidamente com a distância de transmissão, podendo aumentar ainda mais se existem obstruções e interferências no ambiente. Para distâncias curtas, 1 bit transmitido pode consumir a energia equivalente ao processamento de 1.000 a 10.000 instruções (ZHAO; GUIBAS, 2004). É por isso que é preferível processar os dados localmente ao invés de transmití-los a uma estação de processamento remoto.

Como consequência das características dos nós sensoriais, o próprio sistema de RSSF apresenta diferenças substanciais em relação a outros sistemas de redes de dados.

Com o avanço da tecnologia, espera-se que o custo e o tamanho dos nós sensoriais dimi-

nuam. O custo permitirá desenvolver sistemas de RSSF extremamente grandes em relação ao número de nós e facilmente expansíveis. Esses sistemas terão a capacidade de coletar grandes quantidades de informação, o que nos proporcionará melhor conhecimento do fenômeno monitorado. Além disso, essa informação será altamente redundante por causa da quantidade de nós do sistema, o que dará certa garantia em relação às falhas dos nós e/ou perdas de dados na rede.

Com a redução do tamanho dos nós, os sistemas de RSSF tornam-se cada vez mais penetrantes, expandindo nossa capacidade de monitorar lugares onde antes não se tinha acesso. Um exemplo disso são as pesquisas voltadas ao desenvolvimento de redes de sensores sem fio para monitorar o corpo humano (HANSON et al., 2009), denominadas *Body Area Sensor Networks*. Essa expansão traz consigo novos tipos de tráfego, derivados da coleta de informação efetuada diretamente do ambiente físico (ZHAO; GUIBAS, 2004).

Em uma aplicação, a limitação em energia e largura de banda do nó faz com que a transmissão de dados a longas distâncias seja proibitivo. Isso reduziria drasticamente a vida útil dos nós. Normalmente se diminui a distância de transmissão estabelecendo uma rede de múltiplos saltos (*multihop*). Por isso, e pelas características dos nós, é que alguns autores consideram as RSSF como uma caso particular de redes *Mobile ad-hoc Network* (MANET) (UMAR, 2004; MARGI et al., 2009).

As restrições energéticas e o fato de que os nós da rede estão sujeitos a falhas, seja por fatores próprios do ambiente de trabalho (ambientes perigosos onde alguns nós podem ser destruídos) ou por fatores humanos (acidentes, furtos) fazem que uma RSSF tenha uma topologia de rede altamente dinâmica. Adicionalmente, diferentemente das redes de dados tradicionais, os nós sensoriais podem ser multifuncionais. Isso devido a que, dependendo da aplicação, os nós podem exercer as funções de nó sensorial, roteador ou ambas; ao mesmo tempo ou em diferentes momentos durante a aplicação. Espera-se que o sistema seja suficientemente inteligente para adaptar-se a essas eventualidades.

Devido a todas essas características, faz-se necessário o desenvolvimento de novos algoritmos de processamento de informação e de protocolos de rede que otimizem o uso dos limitados recursos da RSSF e suportem suas necessidades nas aplicações.

2.3 DESAFIOS E TEMAS ABERTOS A PESQUISA

Do ponto de vista da aplicação, existem tarefas básicas que toda RSSF deve executar: estabelecimento e manutenção da rede; monitoração de eventos; processamento dos dados coletados; processamento das consultas feitas pelo usuário final e apresentação dos resultados. O desenvolvimento de algoritmos e protocolos de rede para a execução dessas tarefas apresentam desafios únicos como consequência das características da RSSF.

Estabelecimento e manutenção da rede

Para uma apropriada operação, os nós devem conhecer sua localização e quem são seus nós vizinhos. Em redes planejadas, a localização e a topologia da rede é determinada pelos desenvolvedores. Em uma rede espalhada aleatoriamente, os nós devem descobrir sua localização e estabelecer uma topologia de rede autonomamente.

A utilização de sistemas de posicionamento global (GPS – Global Positioning System) como sistema de auto-localização é uma solução simples mas em alguns casos irrealizável. O GPS é sensível a obstruções o que faz com que seja difícil sua utilização em aplicações *indoor*. Adicionalmente, em sistemas de RSSF com milhares de nós, pequenos e com bateria limitada, o custo de colocar um GPS em cada nó e o consumo de energia do GPS torna inviável sua utilização.

A topologia da rede depende da conectividade entre os nós. Uma vez que os nós se comunicam por meio de rádio, fatores como obstruções do ambiente, interferências, orientação das antenas, energia da bateria, etc. afetam essa conectividade. O problema é mais complicado se considerarmos que a rede pode ser expandida ou que os nós são propensos a falhas.

Para lidar com a característica dinâmica da topologia das RSSF os nós devem, inicialmente, construir a rede e após isso, monitorá-la periodicamente para adaptar-se a conectividade existente tomando em conta as necessidades da aplicação. Adicionalmente, se necessita estudar problemas tais como: qual é o tamanho da rede ou o número de enlaces e nós sensoriais necessários para atingir um certo grau predeterminado de redundância ou o controle da distância de comunicação entre os nós para administrar a energia de forma adequada (CHONG; KUMAR, 2003).

Monitoração de eventos

Toda RSSF tem como objetivo obter informação do ambiente. Essa informação pode ativar algum tipo de ação ou simplesmente ser apresentada ao usuário final. Dados os escassos recursos do sistema, especialmente a energia, não é conveniente manter todos os nós ativos durante todo o tempo que dure a aplicação. Além disso, se existir a necessidade de transmitir os dados coletados em uma central de processamento centralizada para serem agregados e analisados, haverá a possibilidade de inundar a rede com dados inúteis, reduzindo a vida útil do sistema (ZHAO; GUIBAS, 2004). Portanto, é recomendável ativar só os nós que sejam necessários para obter a informação desejada. Para permitir isso, os nós sensoriais são desenvolvidos para poderem desligar qualquer um de seus módulos ou ele mesmo entrar em modo *standby*. Porém, a desativação descontrolada dos nós, afeta a cobertura da rede, o que em aplicações como rastreamento de alvos, não é vantajoso. Então devem desenvolver-se algoritmos e protocolos que, em forma distribuída, realizem o controle da ativação/desativação ou ciclo de trabalho dos nós, utilizando só informação local dos sensores. Outro tema de pesquisa relacionado com o controle do ciclo de trabalho dos nós sensoriais é como sincronizar em tempo os nós para sua ativação e desativação. A sincronização de tempo nas RSSF é um problema importante já que, além de afetar o ciclo de trabalho dos nós, também afeta a forma como se integra e interpreta os dados e, inclusive, o encaminhamento da informação.

Processamento dos dados coletados

Já foi mencionado que comunicar os dados de longas distâncias até uma central de processamento centralizada representa um grande gasto de energia para os nós. Outra razão para não desejar processar os dados dessa maneira é que a transmissão dos dados utiliza grande parte da largura de banda do canal RF. Se, em uma RSSF de múltiplos saltos, cada nó tivesse que transmitir seus dados utilizando outro nó da rede, a capacidade de transmissão do canal RF seria proporcional a $\frac{1}{\sqrt{N}}$, sendo N o número de nós da rede. Isso indica que, para uma rede com grande quantidade de nós ou se a rede for expandida, a capacidade de transmissão diminui consideravelmente (ZHAO; GUIBAS, 2004).

Por essas razões, é preferível que os nós processem os dados na rede, colaborando entre si, antes de enviar os resultados ao usuário final. Processamento colaborativo de sinais e informação é uma nova área de pesquisa que estuda métodos distribuídos de fusão de dados distribuído (CHONG; KUMAR, 2003) e de seleção dos nós que devem participar do processamento (ZHAO; GUIBAS, 2004).

Ao desenvolver métodos de processamento colaborativo de sinais e informação deve-se levar em consideração o compromisso que existe entre precisão e utilização ótima de recursos. Quanto mais sensores colaborarem entre si para processar os dados, mais exata será a informação obtida; porém, isso requer maior comunicação entre os nós. A seleção dos nós participantes do processamento melhora a utilização de recursos, mas diminui a precisão do resultado. Igualmente, existem métodos de fusão de dados baseados em regras simples (os métodos mais comuns de agregação de dados utilizam regras como "máximo" e "média" (BOULIS; GANERIWAL; SRIVASTAVA, 2003)) que requerem maior troca de mensagens entre os nós, contudo, podem ser mais precisos do que métodos de fusão mais sofisticados, baseados em modelos, que podem utilizar menor quantidade de mensagens mas sua exatidão depende da precisão do modelo utilizado. Além disso, dada a natureza dinâmica da topologia da rede, as soluções desenvolvidas devem ser robustas às mudanças que a rede possa apresentar durante a aplicação.

Processamento das consultas feitas pelo usuário final e apresentação dos resultados

Toda RSSF deve prover uma interface com a qual o usuário final possa interagir para realizar consultas (*querys*) e receber as informações proporcionadas pela rede. As consultas poderiam ser sobre informações especificas como por exemplo: Qual é a sala que tem maior número de pessoas? ou a RSSF poderia estar pré-programada para efetuar alguma operação como, reconhecimento e rastreamento de objetos, na qual o usuário final teria que inicializar a aplicação indicando o tipo de objeto e o tempo de duração da tarefa.

Vista dessa forma, uma RSSF pode ser considerada como uma grande base de dados onde a informação é obtida diretamente do ambiente e se encontra distribuída entre os nós (YAO; GEHRKE, 2003).

Uma vez recebida a consulta, a rede deve atendê-la, obtendo a informação solicitada dos nós sensoriais e apresentá-la ao usuário final. O atendimento das consultas e apresentação de resultados está relacionado ao desenvolvimento de protocolos de roteamento e é um desafio devido as limitações da rede, à característica dinâmica da topologia da rede e devem, preferentemente, utilizar métodos de processamento colaborativo de sinais e informação.

Nas RSSF, ao contrário das redes de comunicação de dados convencionais, os protocolos baseados em *Internet Protocol* (IP) não são aplicáveis devido ao fato de ser impossível construir um esquema de endereçamento global para os nós (AKKAYA; YOUNIS, 2005). Além disso, os protocolos IP necessitam manter tabelas de roteamento que, dada a falta de confiabilidade dos enlaces da rede, sua atualização consumiria muito tempo, energia e memória (CHONG; KUMAR, 2003).

Ainda, nas RSSF é preferível determinar as rotas de encaminhamento da informação segundo os requerimentos da aplicação, levando em consideração que, na maioria delas, existe a necessidade de coletar a informação a partir de múltiplos nós sensoriais até o nó sorvedouro e que existe redundância de dados na rede que pode ser explorada para reduzir o gasto de energia (AKKAYA; YOUNIS, 2005).

Outra característica importante que os protocolos de roteamento para RSSF devem cumprir é que, do ponto de vista do sistema, o que realmente importa são os dados coletados pelos sensores. Isso tem motivado o desenvolvimento de protocolos de roteamento centrados em dados (*data-centric*), onde as rotas são criadas baseadas na lista de atributos da informação consultada. Finalmente, outros fatores que afetam os protocolos de roteamento são a latência na rede, a distribuição e funcionalidade dos nós e os métodos de fusão de dados utilizados.

Na área das RSSF existem diversos temas abertos a pesquisa que não foram abordados aqui devido ao escopo desse trabalho mas que certamente são de interesse da comunidade científica. Pode-se citar alguns: desenvolvimento de novos módulos de hardware para RSSF; cobertura, que visa desenvolver métodos que permitam assegurar que a rede consiga monitorar completamente um determinado espaço (GHOSH; DAS, 2006); desenvolvimento de sistemas operacionais para os nós sensoriais; atualização de software remotamente; desenvolvimento de métodos de segurança dos dados da rede (MARGI et al., 2009) e privacidade dos dados transmitidos pela rede (CULLER; ESTRIN; SRIVASTAVA, 2004).

3 RASTREAMENTO DE ALVOS COM REDES DE SENSORES SEM FIO

O Rastreamento de alvos é uma capacidade importante para as redes de sensores sem fio. Para Yick (YICK; MUKHERJEE; GHOSAL, 2008), rastreamento de alvos é umas das duas categorias em que se pode classificar as aplicações para RSSF. A outra categoria é monitoração. As aplicações de monitoração compreendem monitoração ambiental interna e externa, monitoração de pacientes em hospitais, monitoração de fenômenos sísmicos e de estruturas de edifícios, dentro outros. As aplicações de rastreamento compreendem rastreamento de objetos, animais, pessoas ou veículos em movimento.

Uma RSSF é ideal para rastrear objetos em movimento, devido a sua cobertura espacial e a multiplicidade no tipo de sensores que a rede pode ter (ZHAO; SHIN; REICH, 2002). Além disso, para Zhao (ZHAO; GUIBAS, 2004), o rastreamento expõe os principais problemas relacionados a uma das características fundamentais desses sistemas: a colaboração entre nós sensoriais. Nesse caso, colaboração refere-se à operação de selecionar e agrupar sensores para realizar tarefas de processamento de sinais e informação. Essa operação é denominada processamento de sinais e informação colaborativa (ZHAO; GUIBAS, 2004).

Rastreamento de alvos móveis é um problema clássico na engenharia (GUPTA; GUI; MOHA-PATRA, 2005). Em uma aplicação básica de rastreamento de alvos, pode-se ter um sistema de rastreamento constituído por 1 centro de processamento e 1 sensor monitorando a área de interesse. Uma vez detectado um objeto móvel, o sensor, que pode ser uma câmera ou um sonar, realiza observações da posição do alvo. Muitas vezes essas observações são realizadas de forma indireta como por exemplo, medindo o ângulo relativo à linha de vista do sensor com o alvo, como é mostrado na figura 4a.

Num cenário mais geral, pode-se ter um sistema com vários centros de processamento e cada um deles possuir 1 ou mais sensores. Esse conjunto (centro de processamento e sensores) será denominado a partir deste momento nó sensorial. Cada um desses nós sensoriais encontra-se espalhado em uma determinada área com o objetivo de rastrear 1 ou múltiplos alvos (figuras 4b e 4c). Os sensores podem ser de diferentes tipos, ter diferentes graus de precisão e podem observar diferentes atributos do alvo como por exemplo, sua posição, sua velocidade e sua direção. Esses sensores podem trabalhar de forma sincronizada, realizando observações ao mesmo tempo ou não sincronizada, realizando observações a intervalos irregulares.

O objetivo de um sistema de rastreamento é determinar a trajetória dos alvos que se encontram dentro de seu alcance de sensoriamento. Isso é feito utilizando as observações obtidas pelos sensores ao longo do tempo e o conhecimento prévio que se tem da dinâmica do(s) alvo(s).

Porém, o sistema deve enfrentar 3 problemas principais. Primeiro, o conhecimento da



(a) Cenário básico onde 1 sensor observa 1 alvo dentro de sua área de sensoriamento





(b) Cenário onde o alvo é observado por múltiplos sensores

(c) Cenário com múltiplos sensores e múltiplos alvos

Figura 4 - Cenários do problema de rastreamento de alvos

dinâmica do alvo, normalmente representado através de um modelo matemático, é impreciso (LI; JILKOV, 2003) já que não é possível incluir dentro de um modelo todos os fatores que podem afetar o movimento do alvo. Segundo, os sensores utilizados não são perfeitos e portanto, as observações obtidas por eles são ruidosas. Terceiro, com relação as observações,

existe mais um problema denominado origem das medições (*origin of the measurements*) ou associação de dados (*data association*) (BAR-SHALOM; LI, 1995), que se dá a partir da necessidade de determinar se a observação utilizada pelo algoritmo de rastreamento pertence ou não ao alvo de interesse. Essa condição existe quando o sistema trabalha na presença de:

- Obstáculos (clutter)
- Contramedidas (countermeasures)
- Falsos alarmes (false alarms)
- Múltiplos alvos

3.1 FUSÃO DE DADOS EM SISTEMAS MULTISENSORIAIS

Para tratar esses problemas, técnicas de fusão de dados têm sido muito utilizadas em sistemas multisensoriais desde longa data (SITTLER, 1964; CHONG; CHANG; MORI, 1986; BAR-SHALOM; LI, 1995; PAO, 1996). As técnicas de fusão de dados utilizam métodos probabilísticos para combinar as observações dos múltiplos sensores do sistema e assim, obter uma descrição mais robusta e completa do ambiente ou processo de interesse (DURRANT-WHYTE; HENDERSON, 2007). Em particular, para o caso do problema de rastreamento de alvos, a combinação inteligente das observações dos múltiplos sensores permite obter informações mais precisas sobre o alvo ou alvos de interesse (DASARATHY, 2000; SMITH; SINGH, 2006).

Nos sistemas multisensoriais tradicionais, geralmente é pressuposto que os nós sensoriais são plataformas com grande capacidade de processamento e que seus sensores podem detectar alvos a grandes distâncias. Se por um lado as RSSF são sistemas multisensoriais, por outro elas apresentam características especiais que as tornam diferentes dos sistemas multisensoriais tradicionais. As RSSF podem chegar a ter um número de nós sensoriais de 3 ou 4 vezes a ordem de magnitude do número de nós dos sistemas tradicionais. Além disso, os nós das RSSF apresentam capacidades limitadas de processamento, sensoriamento, comunicação e energia (GUPTA; GUI; MOHAPATRA, 2005).

Sendo sistemas multisensoriais, as RSSF podem beneficiar-se dos avanços obtidos no desenvolvimento das técnicas de fusão de dados. Mais do que isso, para Nakamura (NAKAMURA; LOUREIRO; FRERY, 2007), a utilização de técnicas de fusão de dados nas RSSF é fundamental. Além de serem bem conhecidas, essas técnicas permitem superar falhas dos sensores, limitações tecnológicas e problemas de cobertura espacial e temporal da rede. Adicionalmente, permitem garantir 3 propriedades importantes que devem estar presentes num sistema multisensorial e, portanto, em uma RSSF: cooperação, redundância e complementaridade.

- Cooperação: As técnicas de fusão de dados permitem compor uma visão completa de um evento de interesse a partir das visões parciais que possuem os nós sensoriais de uma determinada região de cobertura.
- Redundância: Uma RSSF é de natureza redundante devido a quantidade de nós sensoriais que pode ter. As técnicas de fusão de dados permitem obter informação mais precisa aproveitando a redundância da rede, porque permite a fusão das medições correlacionadas de sensores que observam o mesmo fenômeno.
- Complementaridade: Acontece quando se tem sensores de diferentes tipos observando uma região de interesse. As técnicas de fusão de dados permitem combinar dados obtidos a partir de sensores complementares (como por exemplo, sensores que medem ângulo e distância podem ser combinados para obter a posição de um alvo), de forma que a informação resultante permita realizar estimações que poderiam não ser possíveis de obter com as medições individuais.

No entanto, devido as limitações dos recursos de uma RSSF, a fusão de dados deve ser realizada em forma distribuída e colaborativa, o que permite administrar eficientemente a largura de banda e estender a vida útil da rede (NAKAMURA; LOUREIRO; FRERY, 2007; ZHAO; GUIBAS, 2004).

3.2 ARQUITETURAS DE FUSÃO DE DADOS E TOPOLOGIAS DE REDE DAS RSSF

Em sistemas multisensoriais a fusão de dados pode ser realizada em diferentes formas. Essas formas se caracterizam pelo menor ou maior uso da capacidade de processamento dos nós da rede. Os sistemas de fusão de dados podem ser agrupados em 3 categorias (MITCHELL, 2007), conforme ilustrado na figura 5:

1. Sistemas Centralizados: Em uma rede centralizada, existe um nó, normalmente denominado centro de fusão ou processador central, que coleta todos os dados obtidos pelos nós sensoriais da rede e realiza a fusão dos dados. Nessa configuração, pouco ou nenhum processamento de dados é realizado pelos nós da rede, com o qual o processador central tem o controle total do processamento e da interpretação da informação (DURRANT-WHYTE, 2001). Nas RSSF normalmente designa-se como processador central o nó sorvedouro que, além de realizar a fusão, também toma as decisões sobre a aplicação e envia as instruções ou tarefas que os nós sensoriais devem realizar. Em teoria, se todas as observações dos nós sensoriais se encontram no mesmo formato de representação (por exemplo, todas as
posições devem ser expressadas no mesmo sistema de coordenadas), esses sistemas tem o melhor desempenho já que o centro de processamento utiliza todas as medições da rede para realizar a fusão. No entanto, os sistemas centralizados apresentam diversas desvantagens quando aplicados a RSSF. Do ponto de vista da confiabilidade da rede, utilizar um centro de fusão centralizado não é uma solução robusta já que, se o centro de fusão falha, todo o sistema falha. Do ponto de vista da comunicação, centralizar as observações em um só nó causa congestionamento na rede de comunicação e, por conseguinte, atraso na recepção das mensagens enviadas ao centro de fusão. Do ponto de vista da modularidade, a adição de um novo nó sensorial pode mudar a topologia da rede de tal forma que seja necessário mudar a estrutura de controle e comunicação de toda a rede (JULIER; UHLMANN, 2001). Em relação as RSSF, a carga computacional imposta ao nó sorvedouro, que tem que processar todas as observações da rede, é alta o que causa gasto de energia adicional. Finalmente, o consumo da energia dos nós sensoriais nessa arquitetura é o menos eficiente por causa da potência consumida pelo módulo de rádio dos nós sensoriais ao transmitirem as mensagens, a qual é proporcional à distância de comunicação elevada a uma constante α , sendo $\alpha > 2$ (ZHAO; GUIBAS, 2004).

- 2. Sistemas Descentralizados: Em um sistema descentralizado não existe um centro de fusão central. Nesses sistemas os nós sensoriais não tem conhecimento da topologia da rede e cada nó realiza a fusão de dados localmente utilizando observações locais e informação comunicada pelos nós vizinhos. Os sistemas descentralizados são confiáveis já que a perda de subconjuntos de nós sensoriais ou de enlaces de comunicação não afeta, necessariamente, a funcionalidade do sistema. Adicionalmente, são flexíveis no sentido de que nós sensoriais podem ser adicionados ou retirados da rede realizando unicamente mudanças locais. O maior problema que essa configuração deve enfrentar é o efeito da informação redundante. Especificamente, pedaços de informação originadas a partir de múltiplas fontes não podem ser combinadas, se não são independentes ou se não se conhece o grau de correlação entre elas (JULIER; UHLMANN, 2001). Por exemplo: suponhamos que o nó A recebe um pedaço de informação do nó B, mas a topologia da rede é tal que B simplesmente retransmite essa informação para A. Se A realiza a fusão da informação que recebeu de B com a informação antiga que ele tem, assumindo que são informações independentes, então o nó A diminuirá a incerteza da informação erroneamente (MITCHELL, 2007).
- 3. Sistemas Hierárquicos: Uma estrutura de fusão hierárquica pode ser vista como uma mistura das topologias centralizada e descentralizada. Nos sistemas hierárquicos existem diferentes níveis de processamento, onde o nível superior é um nó central de fusão de

dados e o nível mais baixo é um conjunto de nós descentralizados. Nesses sistemas a informação flui do nível mais baixo até o centro de processamento central, passando por diversos níveis onde a informação é combinada e refinada. A principal vantagem que essa arquitetura apresenta em relação às redes centralizadas é que permite distribuir a carga computacional do nó de fusão central nos diversos níveis de processamento da rede. Como desvantagens temos que a rede ainda depende de um nó processador central e, portanto, como acontece com os sistemas centralizados, a confiabilidade, a comunicação e a modularidade são comprometidas. Adicionalmente, a incapacidade dos nós de comunicar a informação por outros meios que não seja a estrutura imposta na topologia hierárquica impede a possibilidade de criar alguma sinergia entre duas ou mais fontes de informação complementares desconectadas na estrutura. Finalmente, o desenvolvedor do sistema encontra-se restrito na combinação da informação ao utilizar uma estrutura hierárquica rígida .

Nas RSSF, a forma de realizar a fusão de dados está relacionada com a forma de comunicar os dados dentro da rede. A razão disso é que, devido à limitada energia que possuem os nós sensoriais, é desejável organizar a comunicação da rede de tal forma que permita explorar a capacidade computacional dos nós sensoriais para processar os dados dentro da rede, e assim reduzir o número de mensagens que seriam transmitidos pela rede e prolongar seu tempo de vida (NAKAMURA; LOUREIRO; FRERY, 2007).

A estrutura de rede das RSSF é estabelecida durante a etapa de incialização. Nessa etapa cada nó sensorial deve descobrir com quais nós pode comunicar-se diretamente e estabelecer a potência de rádio necessária para assegurar uma conectividade adequada. Logo, a rede deve organizar-se em uma estrutura de rede que lhe permita dar suporte à arquitetura de fusão de dados desejada para a aplicação.

Os nós sensoriais de uma RSSF são frequentemente organizados em grupos denominados *clusters* (ZHAO; GUIBAS, 2004). Isso permite utilizar uma estrutura de rede hierárquica (AL-KARAKI; KAMAL, 2004). Os *clusters* estão constituídos por um nó sensorial mais potente ou com capacidades especiais, denominado *cluster-head* e nós sensoriais de menor capacidade conectados ao *cluster-head*. Os membros do *cluster* de menor capacidade encarregam-se de realizar as tarefas de sensoriamento enquanto que os *cluster-head* realizam a fusão dos dados obtidos pelos membros do grupo e comunicam a informação ao sorvedouro (AL-KARAKI; KAMAL, 2004; ZHAO; GUIBAS, 2004). Existem redes hierárquicas estáticas, onde os atributos dos *clusters*, como tamanho, área de cobertura e membros do *cluster* são fixos, e redes hierárquicas dinâmicas, onde os nós sensoriais são idênticos e os *clusters* são formados dinamicamente, dependendo de algum evento de interesse (LI; ZHOU, 2010). Esse tipo de rede



(c) Sistema hierárquico

Figura 5 - Configurações dos sistemas de fusão de dados

é mostrada na figura 6a.

As redes hierárquicas permitem utilizar eficientemente os escassos recursos da rede, como frequência de transmissão, largura de banda e energia dos nós sensoriais. Além disso, permitem monitorar os nós sensoriais e identificar nós com mau funcionamento (ZHAO; GUIBAS, 2004). Porém os algoritmos para formação de *clusters* e seleção de *cluster-heads* são complexos e aumentam o número de mensagens trocados pelos nós sensoriais. Adicionalmente, durante



(b) RSSF plana

Figura 6 - Estruturas de rede das RSSF

a operação da rede os nós sensoriais selecionados como *cluster-heads* consomem muito mais energia do que os demais nós da rede e, portanto, devem ser trocados regularmente para não dividir a estrutura da rede. Finalmente, as redes hierárquicas não são muito escaláveis devido ao fato de que, com o aumento de nós sensoriais, aumenta também o número de *cluster-heads* e como consequência incrementa o número de mensagens para o estabelecimento da estrutura da rede (LEUSCHNER, 2005).

Outra estrutura de rede utilizada em RSSF é a denominada rede plana (*flat network*) (AL-KARAKI; KAMAL, 2004; NAKAMURA; LOUREIRO; FRERY, 2007), na qual cada nó sensorial da rede tem a mesma função (ver figura 6b). Nessas redes, cada nó sensorial só consegue se comunicar com seus nós vizinhos e, portanto, os dados são roteados mediante múltiplos saltos (*multihop*) até o sorvedouro. Logo, a fusão de dados deve ser realizada por cada nó sensorial de forma local ou pelos nós que participam do roteamento de dados (agregação de dados) (NAKAMURA; LOUREIRO; FRERY, 2007).

As redes planas permitem utilizar protocolos de roteamento simples e não é necessário utilizar algoritmos para formação de *clusters* e seleção de *cluster-heads*. Adicionalmente, as redes planas são escaláveis devido ao fato de que cada nó sensorial participa igualmente das tarefas de roteamento e, para serem incorporados na rede, só é preciso a informação de seus vizinhos diretos. Como desvantagem temos que se os nós encontram-se uniformemente distribuídos na rede e só existe um sorvedouro, os nós vizinhos do sorvedouro gastam sua energia antes do resto da rede. Isso acontece porque todo o tráfego da rede é roteado até o sorvedouro através do seus nós vizinhos (LEUSCHNER, 2005).

3.3 MÉTODOS DE ESTIMAÇÃO

Das diversas técnicas de fusão de dados utilizadas nas RSSF, a mais utilizada para abordar o problema de rastreamento de alvos é o método de estimação.

Estimação é o processo de inferir, utilizando as leis da probabilidade, o valor de uma quantidade de interesse a partir de observações indiretas, imprecisas e incertas (BAR-SHALOM; KIRUBARAJAN; LI, 2002). Para isso, o estimador deve determinar o vetor de estado (posição, velocidade, direção, dentre outros) que melhor se ajusta, em um sentido matemático definido, aos dados observados, os quais em geral são ruidosos devido a diversos fatores (HALL, 2004).

As bases da teoria de estimação datam do tempo de Karl Friedrick Gauss e seu desenvolvimento do método de mínimos quadrados (*least squares method*) em 1795 (SORENSON, 1970). Gauss inventou o método de mínimos quadrados para determinar órbitas de corpos celestes a partir de conjuntos de dados redundantes. Utilizando mínimos quadrados, determinou a órbita do asteróide Ceres em 1801 (BJORCK, 1996).

A contribuição de Gauss não foi só na invenção do método de mínimos quadrados. Também estabeleceu algumas idéias que são utilizadas até hoje na teoria de estimação moderna (SORENSON, 1970; HALL, 2004):

- Observabilidade: Quantas e que tipo de observações são necessárias para realizar a estimativa do vetor de estado;
- Modelo Dinâmico: E necessário ter um conhecimento aproximado do movimento do alvo ou da dinâmica do sistema. Quanto melhor o nosso conhecimento, melhor é o valor estimado;
- Valor inicial (a priori): Gauss percebeu o papel do valor inicial do vetor do estado na determinação da solução do problema;
- Tratamento probabilístico: Ao reconhecer a imprecisão das medições, Gauss estabeleceu

os princípios para uma interpretação probabilística deles. Além disso, reconheceu a necessidade de ter dados redundantes para reduzir a influência dos erros das observações na estimativa.

Desenvolvimentos posteriores notáveis na teoria de estimação são a interpretação probabilística do método de mínimos quadrados e a definição do método de máxima verosimilhança feitos por Fisher (FISHER, 1912), o desenvolvimento de Wiener (WIENER, 1964) e Kolmogorov (KOLMOGOROV, 1992) do método do mínimo erro quadrático médio linear, e o desenvolvimento do filtro de Kalman (KALMAN, 1960).

O desenvolvimento de um algoritmo de estimação requer que o desenvolvedor tome decisões sobre diversas questões. Para Hall (HALL, 2004), essas questões podem ser classificadas em 4 categorias :

- Modelos do sistema: Que modelos serão selecionados para definir o problema?, Que quantidades serão estimadas?, Que parâmetros são suficientes e necessários para descrever o estado do sistema?, Como será realizada a predição do vetor de estados?, Como estão relacionadas as observações com o vetor de estados?, Que pressupostos (se existem) pode-se fazer sobre as observações do sistema?
- Critério de otimização: Como será definido o critério para especificar o "melhor ajuste" dos dados observados? Em outras palavras, que equação será utilizada para especificar que o vetor de estados é o melhor ajuste para os dados observados?
- Metodologia da otimização: Uma vez selecionado o critério de otimização, que método será utilizado para determinar o vetor de estados que satisfaz esse critério?
- Metodologia do processamento dos dados: Como serão processadas as observações: de modo sequencial ou por lotes (*batch*)?

Na figura 7 apresentam-se as quatro categorias descritas acima e exemplos das alternativas dentro de cada uma delas.

Esse trabalho usa metodologia de processamento por lotes para resolver o problema de rastreamento de alvos, portanto, a seguir são descritas com maior detalhe as metodologias de processamento que existem para realizar a estimativa das variáveis.

3.3.1 Metodologia do Processamento

Do ponto de vista do processamento dos dados, a estimativa das variáveis pode ser realizada de 2 formas: sequencial ou por lotes (*batch*) (HALL, 2004). A seguir, descreveremos com



Figura 7 - Alternativas para o desenvolvimento de algoritmos de estimação (Adaptado de (HALL, 2004))

mais detalhe esses dois tipos de estimação. A teoria de estimação é uma teoria geral que é aplicada a diversos sistemas dinâmicos, porém devido ao fato de que a metodologia proposta nessa tese é aplicada ao rastreamento de alvos, na explicação a seguir, se não for especificado o contrário, assumiremos que o sistema observado é um alvo móvel.

3.3.1.1 Estimação Sequencial

Na estimação sequencial, também conhecida como estimação sequencial bayesiana (ZHAO; GUIBAS, 2004), o estimador tem como objetivo obter a função de densidade de probabilidade (PDF – *probability density function*) do vetor de estados dado todo o histórico de observações. Essa PDF representa toda a informação estatística disponível e pode-se dizer que é a solução completa ao problema (GORDON; SALMOND, 1998).

O algoritmo mais geral para realizar essa forma de estimação é o filtro de Bayes (THRUN; BURGARD; FOX, 2005). O filtro de Bayes estima a PDF do estado através da aplicação da regra de Bayes. A regra de Bayes estabelece a seguinte relação:

$$p(x \mid z) = \frac{p(z \mid x)p(x)}{\int p(z \mid x)p(x)dx} = \eta p(z \mid x)p(x)$$
(4)

onde, η é a constante de normalização, p(x) corresponde à distribuição *a priori* do estado, a qual sintetiza o conhecimento que se tem do estado x antes de incorporar as observações *z*, $p(x \mid z)$ corresponde à distribuição *a posteriori* do estado, também denominada crença e $p(z \mid x)$ corresponde à função de verossimilhança que descreve como o estado *x* causa a medição *z* do sensor.

Para realizar a estimação, normalmente efetua-se observações em intervalos de tempo regulares k. O estado no tempo k será representado mediante $x_k = x(k)$, que, dependendo do problema abordado, pode ser um vetor ou um valor escalar.

O histórico de medições será representado mediante o vetor $\overline{\mathbf{z}_k} = \{z_1, z_2, \dots, z_k\}$, onde $z_k = z(k)$ representa uma observação realizada no tempo k. As observações podem ser geradas por um único sensor ou um grupo de sensores. No caso de serem geradas por um grupo de s sensores, a observação do sensor s no tempo k será representada por $z_k^s = z^s(k)$, a coleção de todas as observações dos sensores no tempo k será representada mediante o vetor $\mathbf{z}_k^s = \{z_k^1, z_k^2, \dots, z_k^s\}$.

Na estimação de estados em sistemas dinâmicos, um pressuposto importante é que a sequência ou transição de estados x_0, x_1, \dots, x_k do sistema é um processo de Markov (BAR-SHALOM; KIRUBARAJAN; LI, 2002). Isso quer dizer que o estado x_k é completo e portanto, o conhecimento de estados passados ou medições não contém informação adicional que nos permita predizer o futuro com maior precisão (THRUN; BURGARD; FOX, 2005). Com esse pressuposto, as seguintes relações de independência condicional são satisfeitas:

• A função de verossimilhança é expressada por:

$$p\left(\underline{\mathbf{z}_{k}^{s}}\mid x_{k}\right) = \prod_{s=1,2\cdots,S} p\left(z_{k}^{s}\mid x_{k}\right)$$
(5)

• Condicionado ao estado x_k , o novo conjunto de observações \underline{z}_k^s é independente do histórico de medições passadas $\overline{z_{k-1}}$

$$p(z_k^s \mid x_k, \overline{z_{k-1}}) = p(z_k^s \mid x_k)$$
(6)

• Condicionado ao estado x_{k-1} , o estado x_k é independente do histórico de medições passadas $\overline{\mathbf{z}_{k-1}}$

$$p(x_k \mid x_{k-1}, \overline{\mathbf{z}_{k-1}}) = p(x_k \mid x_{k-1})$$
(7)

A cada instante k o estimador atualiza a crença $p(x_k | \overline{\mathbf{z}_k})$. Suponhamos que temos a crença calculada no tempo k - 1, $p(x_{k-1} | \overline{\mathbf{z}_{k-1}})$ e, no tempo k obtemos um novo conjunto de medições efetuado mediante s sensores, $\underline{\mathbf{z}_k^s}$. Utilizando a regra de Bayes e as equações (5), (6) e (7), a crença $p(x_k | \overline{\mathbf{z}_k})$ é calculada da seguinte forma:

$$p(x_{k} \mid \overline{\mathbf{z}_{k}}) = p\left(x_{k} \mid \underline{\mathbf{z}_{1}^{m}}, \underline{\mathbf{z}_{2}^{n}}, \cdots, \underline{\mathbf{z}_{k}^{s}}\right)$$

$$= \eta \left[p\left(\underline{\mathbf{z}_{k}^{s} \mid x_{k}\right) p\left(x_{k} \mid \overline{\mathbf{z}_{k-1}}\right)\right]$$

$$= \eta \left[\prod_{s=1,2,\cdots,S} p\left(z_{k}^{s} \mid x_{k}\right)\right] \int p\left(x_{k} \mid x_{k-1}, \overline{z_{k-1}}\right) p\left(x_{k-1} \mid \overline{\mathbf{z}_{k-1}}\right) dx_{k-1}$$

$$= \eta \left[\prod_{s=1,2,\cdots,S} p\left(z_{k}^{s} \mid x_{k}\right)\right] \int p\left(x_{k} \mid x_{k-1}\right) p\left(x_{k-1} \mid \overline{\mathbf{z}_{k-1}}\right) dx_{k-1}$$
(8)

Vemos na equação (8) que utilizando a regra de Bayes de forma recursiva pode-se calcular $p(x_k \mid \overline{\mathbf{z}_k})$, já que só depende da observação $\underline{\mathbf{z}_k^s}$ e da estimativa calculada anteriormente, $p(x_{k-1} \mid \overline{\mathbf{z}_{k-1}})$. Essa forma de estimação é denominada sequencial devido a que o cálculo do valor estimado é realizado observação traz observação durante o transcurso da aplicação (HALL, 2004).

Uma vez calculada a PDF, o valor estimado é dado por algum critério de otimização. Existem 2 critérios de otimização muito utilizados para variáveis aleatórias: o estimador de máximo *a posteriori* (MAP – *Maximum A Posteriori*) e o estimador de mínimo erro quadrático médio (MMSE – *Minimum Mean Square Error*).

O estimador MAP calcula a estimativa da variável mediante a maximização da distribuição *a posteriori*:

$$\hat{x}_{k}^{MAP} = \underset{x_{k}}{\operatorname{argmax}} p\left(x_{k} \mid \overline{\mathbf{z}_{k}}\right) \tag{9}$$

O estimador MMSE minimiza o valor esperado do erro ao quadrado condicionado às observações:

$$\hat{x}_{k}^{MMSE} = \operatorname{argmin}_{x_{k}} E[(\hat{x}_{k} - x_{k})^{2} \mid \overline{\mathbf{z}_{k}}]$$
$$= E[x_{k} \mid \overline{\mathbf{z}_{k}}]$$
(10)

No caso em que a distribuição *a priori* seja uma distribuição gaussiana, ambos os estimadores são equivalentes:

$$\hat{x}_{k}^{MAP} = \hat{x}_{k}^{MMSE} = E\left[x_{k} \mid \overline{\mathbf{z}_{k}}\right] = \int x_{k} p(x_{k} \mid \overline{\mathbf{z}_{k}}) dx_{k}$$
(11)

Finalmente, a qualidade do estimador é dada pelo erro quadrático médio condicional (*conditional* MSE – *Mean Square Error*), que produz como resultado a matriz de covariância:

$$\sum = \int (x_k - \hat{x}_k) (x_k - \hat{x}_k)^T p(x_k \mid \overline{\mathbf{z}_k}) dx$$
(12)

O filtro de Kalman

O filtro de Kalman (KF – *Kalman Filter*) é a variante do filtro de Bayes mais utilizada (FOX et al., 2003). Foi inventado por Swerling (1958) e Kalman (1960) como uma técnica de filtragem e predição em sistemas lineares gaussianos (THRUN; BURGARD; FOX, 2005).

O algoritmo de Kalman é uma forma especial do filtro bayesiano onde é pressuposto que a dinâmica do sistema observado e o modelo das observações são lineares e que a incerteza na transição de estados é um vetor aleatório gaussiano. Ele trabalha sobre o seguinte sistema dinâmico linear de tempo discreto:

$$x_k = F_{k-1}x_k + v_k \tag{13}$$

$$z_k = H_k x_k + w_k \tag{14}$$

onde (13) é a equação dinâmica do sistema ou processo ³ e (14) é a equação da observação. Nessas equações, k = 1, 2, ... representa o tempo, x_k é o vetor de estado a ser estimado, $F_k = F(k)$ e $H_k = H(k)$ são matrizes conhecidas e podem variar com o tempo e $v_k = v(k)$ e $w_k = w(k)$ são os ruídos do processo e da observação, respectivamente.

O pressuposto básico para a derivação do algoritmo é que v_k e w_k são variáveis aleatórias gaussianas com média zero, não correlacionadas no tempo

$$E\{v_k\} = E\{w_k\} = 0, \qquad \forall k$$
(15)

e com matrizes de covariância $Q_k = Q(k)$ e $R_k = R(k)$ (covariância do ruído da dinâmica do sistema e do ruído da observação, respectivamente) conhecidas

$$E\left\{v(i).v(j)^{T}\right\} = \delta_{ij}Q(i)$$
(16)

$$E\left\{w(i).w(j)^{T}\right\} = \delta_{ij}R(i)$$
(17)

Adicionalmente, geralmente é pressuposto que os ruídos do processo e da observação também não são correlacionados

$$E\left\{v_k.(w_k)^T\right\} = 0, \qquad \forall k \tag{18}$$

³Normalmente, a equação da dinâmica do processo é apresentada com um termo adicional relacionado ao vetor de entrada u_i do sistema, porém para modelar o problema de rastreamento esse termo é irrelevante e tem sido deixado de lado propositadamente.

O sistema definido por (13) e (14) se comporta como um processo de Gauss-Markov já que, devido à linearidade das equações, os estados e as observações futuras preservam a característica Gaussiana dos estados e observações predecessores (BAR-SHALOM; KIRUBARAJAN; LI, 2002).

Nas equações do algoritmo a seguinte notação será utilizada: x(k) representa a variável aleatória que queremos estimar, $\hat{x}(k \mid k)$ é o valor estimado da variável no tempo k, $\hat{x}(k \mid k-1)$ é a predição da variável ou o valor estimado da variável antes de incorporar a observação do tempo k, $P(k \mid k)$ é a matriz de covariância no tempo k e $P(k \mid k-1)$ é predição da matriz de covariância.

O filtro de Kalman é um algoritmo recursivo. A cada iteração, utilizando os valores obtidos na iteração anterior, $\hat{x}(k-1 | k-1)$ e P(k-1 | k-1), o algoritmo produz o valor estimado $\hat{x}(k | k)$, o qual minimiza a norma do erro, representado pela matriz de covariância associada à estimação $\hat{x}(k | k)$, definida por:

$$P(k \mid k) \triangleq E\left[[x(k) - \hat{x}(k \mid k)] [x(k) - \hat{x}(k \mid k)]^T \mid z(1), z(2), \cdots, z(k) \right]$$
(19)

portanto:

$$\hat{x}(k \mid k) = \underset{\hat{x}(k\mid k)}{\operatorname{argmin}} \{ P(k \mid k) \}$$

=
$$\underset{\hat{x}(k\mid k)}{\operatorname{argmin}} \{ E \left[[x(k) - \hat{x}(k \mid k)] [x(k) - \hat{x}(k \mid k)]^T \mid z(1), z(2), \cdots, z(k) \right] \}$$
(20)

A solução da equação (20) é o valor esperado do estado x(k) condicionado a todas as observações até o tempo k (média condicional):

$$\hat{x}(k \mid k) = E[x(k) \mid z(1), z(2), \cdots, z(k)]$$

O algoritmo começa com a estimativa inicial $\hat{x}(0 \mid 0)$ do estado x(0) e sua respectiva covariância $P(0 \mid 0)$. A cada observação realizada, o algoritmo realiza as duas seguintes etapas:

• Predição: A predição $\hat{x}(k \mid k-1)$ do estado no tempo k e sua covariância $P(k \mid k-1)$ são calculados mediante as seguintes fórmulas:

$$\hat{x}(k \mid k-1) = F(k)\hat{x}(k-1 \mid k-1)$$
(21)

$$P(k \mid k-1) = F(k)P(k-1 \mid k-1)F(k)^{T} + Q(k)$$
(22)

• Atualização: A predição $\hat{x}(k \mid k-1)$ e a matriz de covariância $P(k \mid k-1)$ são atualizadas utilizando a observação z(k), de acordo com as seguintes equações:

$$W(k) = P(k \mid k-1)H(k)^{T}[H(k)P(k \mid k-1)H(k)^{T} + R(k)]^{-1}$$
(23)

47

$$\hat{x}(k \mid k) = \hat{x}(k \mid k-1) + W(k)(z(k) - H(k)\hat{x}(k \mid k-1))$$
(24)

$$P(k \mid k) = (I - W(k)H(k))P(k \mid k - 1)(I - W(k)H(k))^{T} + W(k)R(k)W(k)^{T}$$
(25)

onde W(k) é o ganho do filtro. O valor resultante do algoritmo, $\hat{x}(k \mid k)$, corresponde ao estimador MMSE ou também denominado melhor estimador linear não viciado (BLUE – Best Linear Unbiased Estimator). Se os erros não fossem gaussianos e se só conhecêssemos os dois primeiros momentos da distribuição de probabilidade, o algoritmo do filtro de Kalman daria como resultado o melhor estimador linear (BLE – Best Linear Estimator) do estado.

No caso em que o estado seja observado por vários sensores, a cada sensor lhe corresponde uma equação de observação

$$z^{s}(k) = H^{s}(k)x(k) + w^{s}(k), \qquad s = 1, 2, \cdots, S$$
 (26)

e uma matriz de covariância $R^{s}(k)$. S corresponde ao número de sensores que observam o estado e os ruídos $w^{s}(k)$ são todos gaussianos com média zero, não correlacionados no tempo e não correlacionados entre sensores.

Essas equações podem ser combinadas, definindo o vetor de observação composto como

$$\mathbf{z}(k) \triangleq \left[z^1(k)^T, \cdots, z^S(k)^T \right]^T$$
(27)

e o modelo de observação composto como

$$\mathbf{H}(k) \triangleq \left[H^1(k)^T, \cdots, H^S(k)^T \right]^T$$
(28)

com o vetor de ruídos

$$\mathbf{w}(k) \triangleq \left[w^1(k)^T, \cdots, w^S(k)^T \right]^T$$
(29)

onde a partir da equação (17) temos

$$\mathbf{R}(k) = E\left\{\mathbf{w}(k).\mathbf{w}(k)^{T}\right\}$$
$$= E\left\{\left[w^{1}(k)^{T}, \cdots, w^{S}(k)^{T}\right]^{T}\left[w^{1}(k)^{T}, \cdots, w^{S}(k)^{T}\right]\right\}$$
$$= blockdiag\left\{R^{1}(k), \cdots, R^{S}(k)\right\}$$
(30)

Com esse conjunto de equações, a equação da observação (14) pode ser reescrita como

$$\mathbf{z}(k) = \mathbf{H}(k)x(k) + \mathbf{w}(k)$$
(31)

Com a equação (31), o cálculo da estimativa do estado pode ser realizado utilizando-se as equações do filtro de Kalman.

O filtro de Kalman trabalha sobre sistemas lineares. Quando o sistema e o modelo das observações são não lineares deve-se utilizar métodos para linearizar as equações, o que dá lugar às variantes do filtro. Duas das variantes mais conhecidas são o filtro de Kalman estendido (EKF – *Extended Kalman Filter*) (CHEESEMAN; SMITH, 1986; MOUTARLIER; CHATILA, 1990), o qual utiliza linearização por expansão de Taylor e o filtro de Kalman *unscented* (UKF – *Unscented Kalman Filter*) (JULIER; UHLMANN, 1997), o qual realiza uma linearização estocástica através do uso de um processo de regressão linear estatística ponderada (THRUN; BURGARD; FOX, 2005).

Uma variante do filtro de Kalman importante para sistemas multisensoriais de fusão de dados é o filtro de informação (IF – Information Filter) (THRUN; BURGARD; FOX, 2005). Esse filtro é matematicamente equivalente ao filtro de Kalman salvo que, ao invés de calcular o estado $\hat{x}(k \mid k)$ e a covariância $P(k \mid k)$, ele calcula o vetor de informação $\hat{y}(k \mid k)$ e a matriz de informação $Y(k \mid k)$ (DURRANT-WHYTE; HENDERSON, 2007). Essas variáveis estão relacionadas com o vetor de estado e a matriz de convariância mediante as seguintes equações:

$$\hat{y}(k \mid k) = P(k \mid k)^{-1} \hat{x}(k \mid k)$$
 (32)

$$Y(k \mid k) = P(k \mid k)^{-1}$$
(33)

A distribuição gaussiana parametrizada mediante o vetor de informação e a matriz de informação é denominada parametrização canônica (THRUN; BURGARD; FOX, 2005).

O filtro de informação segue as mesmas etapas de predição e atualização do filtro de Kalman: Predição:

$$Y(k \mid k-1) = (F(k)Y(k-1 \mid k-1)^{-1}F(k)^{T} + Q(k))^{-1}$$
(34)

$$\hat{y}(k \mid k-1) = Y(k \mid k-1)(F(k)Y(k-1 \mid k-1)^{-1}\hat{y}(k-1 \mid k-1))$$
(35)

Atualização:

$$Y(k \mid k) = H(k)^T R(k)^{-1} H(k) + Y(k \mid k - 1)$$
(36)

$$\hat{y}(k \mid k) = H(k)^T R(k)^{-1} z(k) + \hat{y}(k \mid k-1)$$
(37)

A maior vantagem do filtro de informação sobre o filtro de Kalman em sistemas de fusão de dados é a simplicidade da etapa de atualização. Para um sistema com n sensores a etapa

de atualização do IF é exatamente a somatória das contribuições de informação de todos os sensores (DURRANT-WHYTE; HENDERSON, 2007):

$$\hat{y}(k \mid k) = \hat{y}(k \mid k-1) + \sum_{i=1}^{n} H_i(k) R_i(k)^{-1} z_i(k)$$
(38)

$$Y(k \mid k) = Y(k \mid k-1) + \sum_{i=1}^{n} H_i R_i(k)^{-1} H_i(k)^T$$
(39)

Essa propriedade tem sido explorada na solução do problema de estimação descentralizada por Rao e Sheen (1993) e também no desenvolvimento de filtro de canal (CHF – *Channel Filter*) por Durrant-Whyte (2000).

Uma das desvantagens substanciais do filtro de informação é a codificação de modelos não lineares, especialmente para a etapa de predição (DURRANT-WHYTE; HENDERSON, 2007). Ainda assim, como acontece com o filtro de Kalman, o filtro de informação pode ser estendido para o caso não linear, existindo o filtro de informação estendido (EIF – *Extended Information Filter*) (THRUN; BURGARD; FOX, 2005) e o filtro de informação *unscented* (UIF – *Unscented Information Filter*) (LEE, 2008).

Estimação sequencial distribuída aplicada ao rastreamento de alvos com RSSF

Para Zhao e Guibas (2004), o rastreamento de alvos com RSSF utilizando estimação sequencial pode ser implementado de 3 formas: a crença pode ser armazenada num nó sensorial fixo, em uma sequência de nós realizando *hand-off* s sucessivos ou num subconjunto de nós concorrentes.

Na figura 8 mostra as 3 formas de estimação sequencial. Nela os círculos representam nós sensoriais e os círculos pretos representam nós que guardam informações sobre o estado do objeto. As flechas cinzas ou linhas são caminhos de comunicação entre nós vizinhos. As flechas pretas estreitas representam *hand-offs* entre nós sensoriais. As flechas grossas representam a trajetória que segue o objeto dentro do campo de sensores.

A primeira forma corresponde a uma configuração de fusão de dados centralizada, na qual a cada tempo t um nó sensorial fixo é selecionado como processador central e recebe todas as observações realizadas pelos nós da rede, para depois, realizar a estimação (ver figura 8a). Essa configuração é a mais simples de implementar e a que dá o melhor resultado na estimação. Porém é a de menor eficiência energética devido ao grande número de mensagens na rede durante a transmissão das observações.

Normalmente essa configuração é implementada para fins de comparação (LI; ZHOU, 2010).



(a) Configuração centralizada. Um nó fixo realiza a estimação.



(b) Configuração clusterizada. Nós líderes são selecionados de acordo com o movimento do

objeto.



(c)Todos os nós sensoriais armazenam e realizam a estimativa da posição do alvo

Figura 8 - Configurações das RSSF para realizar rastreamento de alvos mediante estimação sequencial. (Adaptado de (ZHAO; GUIBAS, 2004))

A segunda forma corresponde a uma rede hierárquica clusterizada (figura 8b). Nesse caso, a rede seleciona um nó sensorial líder inicial o qual coleta as observações dos nós que observaram o alvo e realiza a estimação. Na medida que o alvo se movimenta pela rede, o nó inicial pode já não ser desejável, devido ao fato de que pode ficar cada vez mais longe do alvo e dos nós que realizam as observações. Nesse momento um novo líder é eleito pela rede o qual se encarregará de continuar a realizar a estimativa estado do alvo. Para isso, o novo líder recebe a crença do líder anterior, coleta as observações e atualiza a crença. Essa configuração corresponde a utilizar um esquema de clusterização dinâmico na RSSF. O rastreamento também pode ser realizado em uma rede de *clusters* fixos, na qual o *cluster-head* estima a posição do alvo enquanto ele esteja no alcance de sensoriamento do cluster. Quando o alvo está saindo do alcance do cluster, o cluster-head transmite a crença ao cluster-head do grupo de nós sensoriais que observa o alvo nesse momento. Essa operação é denominada hand-off. A vantagem dessa configuração é que mantém a comunicação localizada entre nós sensoriais que se encontram na vizinhança geográfica do alvo, o que permite reduzir o consumo de largura de banda e incrementar a vida útil da rede. Contudo, um dos desafios mais importantes na implementação dessa configuração é a definição de um critério efetivo para formar clusters e escolher clusterheads dentro da rede (KUMARAWADU et al., 2008; RAMESH; SOMASUNDARAM, 2011). Adicionalmente, em um esquema de seleção dinâmica de um líder para realizar a estimação, o estimador pode sofrer de falta de robustez em caso de atraso na seleção do nó líder.

Na terceira forma, a crença é armazenada de forma distribuída nos diversos nós sensoriais da rede (figura 8c). Essa configuração é atrativa do ponto de vista da robustez. A estimativa a partir dos dados observados é realizada em cada nó sensorial, fazendo com que a comunicação seja localizada. O maior desafio dessa configuração é estimar eficientemente propriedades globais sobre os alvos a partir de informações parciais e locais, e manter a consistência da estimação nos diversos nós sensoriais. Além disso, a informação do estado deve ser organizada de tal forma que permita ao usuário da RSSF realizar consultas sobre a informação que a rede possui. Um exemplo desse caso é apresentado por Olfati-Saber em (OLFATI-SABER; SANDELL, 2008). Nesse trabalho Olfati-Saber apresenta uma solução ao problema de rastreamento de alvos manobráveis utilizando uma versão do filtro Kalman-Consensus (KCF Kalman-Consensus Filter) (OLFATI-SABER, 2007) baseada em passagem de mensagens (message-passing). O KCF é uma rede plana (flat network) de estimadores locais, denominados microfiltros de Kalman, embutidos nos nós sensoriais. Esses microfiltros interagem localmente até chegar a um consenso (*consensus*) sobre o valor estimado do vetor de estados. Dessa forma a crença é distribuída na RSSF, já que todos os nós sensoriais obtém aproximadamente o mesmo valor da variável estimada. Para realizar o rastreamento, a RSSF é organizada em uma estrutura hierárquica na qual, no nível baixo a rede de microfiltros realiza a estimativa

da localização do alvo, enquanto que um centro de fusão de alto nível seleciona aleatoriamente um conjunto de nós sensoriais (microfiltros) para combinar seus valores estimados mediante uma função de agregação.

3.3.1.2 Estimação por Lotes (Batch Estimation)

Na metodologia de estimação por lotes todas as observações são coletadas antes de realizar a estimativa. Essa forma de estimação é comumente utilizada em problemas de ajuste de curvas ou modelos. Além disso, dado que é necessário coletar todas as observações, esse tipo de estimação é utilizado em problemas nos quais o tempo não é uma variável crítica. Por exemplo, na estimação da trajetória de um asteróide é necessário coletar observações durante vários dias, para depois, realizar a estimativa (HALL, 2004). Para ilustrar esse tipo de estimação, nesta sessão apresentamos a formulação do problema de estimação desde um ponto de vista algebraico. Mais adiante, na seção 5.2 apresentaremos uma formulação mais geral, utilizando redes bayesianas. Porem, antes de apresentar o método de estimação por lotes, vamos revisar a teoria relacionada com o problema de mínimos quadrados e sua solução mediante eliminação de Gauss, já que essa teoria é utilizada para realizar a estimação por lotes.

O problema de mínimos quadrados (Least Squares)

A teoria de mínimos quadrados, desenvolvida por Gauss no ano 1795 (SORENSON, 1970), é um dos temas de estudo mais importante da Álgebra Linear. Ela serve para dar solução a sistemas de equações lineares que não tem solução exata. Nesta sessão descreveremos as idéias principais por trás da solução de sistemas lineares sobredeterminados utilizando mínimos quadrados.

Consideremos o problema de determinar o vetor $x \in \mathbb{R}^n$ que da solução ao sistema de equações:

$$Ax = b \tag{40}$$

onde a matriz de dados $A \in \mathbb{R}^{m \times n}$ e o vetor de observações $b \in \mathbb{R}^m$ são conhecidos e $m \ge n$.

Quando um sistema de equações tem mais equações do que incógnitas o sistema é denominado sistema sobredeterminado. Normalmente os sistemas sobredeterminados não tem solução exata. Por esse motivo, a solução consiste em minimizar $||Ax - b||_p$ para uma norma p determinada. Quando p = 2 o problema é denominado problema de mínimos quadrados ordinarios (OLS – Ordinary Least Squares) (GOLUB; LOAN, 1996) e consiste em determinar:

$$\hat{x} = \min_{x} \|Ax - b\|_{2} = \min_{x} \|Ax - b\|^{2}$$
(41)

O método mais utilizado para solucionar o problema OLS é o método das equações normais (GOLUB; LOAN, 1996). Se as colunas de A são linearmente independentes, demonstra-se que existe uma solução única ao problema de mínimos quadrados e que é dada pelo vetor \hat{x} que dá solução ao sistema de equações normais:

$$A^T A x = A^T b \tag{42}$$

onde a matriz $A^T A$ tem como característica ser simétrica positiva definida.

Soluciona-se o sistema (42) mediante eliminação de Gauss (STRANG, 1986). Sendo $A^T A$ simétrica, utiliza-se fatoração de Cholesky (GOLUB; LOAN, 1996) para decompor a matriz em:

$$A^T A \triangleq R^T R \tag{43}$$

sendo $R \in \mathbb{R}^{n imes n}$ uma matriz triangular superior denominada triângulo de Cholesky.

Dessa maneira podemos determinar o vetor \hat{x} solucionando dois sistemas: primeiro $R^T y = A^T b$ mediante substituição direta (*forward substitution*) e finalmente $R\hat{x} = y$ mediante substituição inversa (*back substitution*).

Para matrizes densas, a fatoração de Cholesky precisa $n^3/3$ operações de ponto flutuante (flops – *floating point operations*) (DELLAERT; KAESS, 2006) enquanto que o algoritmo completo para solucionar o sistema mediante o método de equações normais precisa de $(m+n/3) n^2$ (GOLUB; LOAN, 1996).

Uma outra forma de solucionar o problema OLS é mediante ortogonalização. Esse método é mais precisso e numericamente mais estável do que o método de equações normais (DEL-LAERT; KAESS, 2006; STRANG, 1986) e consiste em calcular a fatoração QR da matriz A:

$$A = Q \begin{bmatrix} R\\0 \end{bmatrix}$$
(44)

onde $Q \in \mathbb{R}^{m \times n}$ é uma matriz ortogonal e $R \in \mathbb{R}^{n \times n}$ uma matriz triangular superior com elementos positivos na diagonal, que corresponde ao fator de Cholesky (equação (43)) (BJORCK, 1996).

A decomposição ortogonal (44) utilizada no sistema (40) soluciona o problema sem ter que

calcular a matriz $A^T A$:

$$Ax = b$$

$$QRx = b$$

$$Q^{T}QRx = Q^{T}b$$

$$\begin{bmatrix} R \\ 0 \end{bmatrix} x = Q^{T}b$$
(45)

O método preferido para realizar a fatoração QR é utilizar transformações de Householder (DELLAERT; KAESS, 2006). O método consiste em realizar a eliminação de Gauss coluna por coluna, da esquerda para a direita, multiplicando a matriz A, pela esquerda, pelas matrizes de reflexão de Householder H_j correspondentes a cada coluna. Após n iterações A estará completamente fatorada:

$$H_n \dots H_2 H_1 A = Q^T A = \begin{bmatrix} R \\ 0 \end{bmatrix}$$
(46)

Na prática, a matriz ortogonal Q não é calculada. Ao invés disso, $Q^T b$ é determinado realizando a transformação de Householder sobre a matriz aumentada [A | b]. Devido ao fato de que Q é ortogonal, temos que:

$$\|Ax - b\|^{2} = \|Q^{T}Ax - Q^{T}b\|^{2}$$
$$= \left\| \begin{bmatrix} R \\ 0 \end{bmatrix} x - \begin{bmatrix} c \\ d \end{bmatrix} \right\|^{2}$$
$$= \|Rx - c\|^{2} + \|d\|^{2}$$
(47)

Portanto, utilizando (47) temos que

$$\hat{x} = \underset{x}{\operatorname{argmin}} \|Ax - b\|$$

$$\hat{x} = \underset{x}{\operatorname{argmin}} \{\|Rx - c\|^{2} + \|d\|^{2}\}$$

$$\hat{x} = \underset{x}{\operatorname{argmin}} \|Rx - c\|^{2}$$
(48)

onde fica claro que o termo $||d||^2$ é o mínimo residual (*minimum residual*) (GOLUB; LOAN, 1996; DELLAERT; KAESS, 2006) e a solução \hat{x} é dada pela solução do sistema

$$Rx = c \tag{49}$$

por substituição inversa.

O trabalho do algoritmo para a solução do problema de mínimos quadrados é dominado pelo trabalho realizado para fatorar a matriz A e requer $2n^2 (m-n/3)$ flops (GOLUB; LOAN, 1996).

Nessa forma de estimação, o sistema observado é o mesmo utilizado na estimação sequencial e se cumprem os mesmos pressupostos. Portanto, a cada instante de tempo k vamos ter a equação do modelo da dinâmica do sistema e a equação de observação em cada sensor que observa o estado do sistema:

$$x_k = F_{k-1}x_{k-1} + v_k \tag{50}$$

$$z_k^s = H_k^s x_k + w_k^s \tag{51}$$

onde s = 1, 2, ..., S identifica os nós sensoriais e cada equação é associada às matrizes de covariância Q_k e R_k^s , respectivamente. Essas equações podem ser combinadas em um sistema de equações único (STRANG, 1986). Por exemplo, assumindo que temos um sensor observando (s = 1 mas é omitido por conveniência) o sistema e que $x_0 = \hat{x}_0$ e sua matriz de covariância P_0 são conhecidas, o sistema de equações para os tempos k = 1, 2, 3, 4 teria a seguinte forma:

$$x_{0} = \hat{x}_{0}$$
$$-F_{0}x_{0} + x_{1} = 0$$
$$H_{1}x_{1} = z_{1}$$
$$-F_{1}x_{1} + x_{2} = 0$$
$$H_{2}x_{2} = z_{2}$$
$$-F_{2}x_{2} + x_{3} = 0$$
$$H_{3}x_{3} = z_{3}$$
$$-F_{3}x_{3} + x_{4} = 0$$
$$H_{4}x_{4} = z_{4}$$

Expressando esse sistema em forma matricial teríamos:

$$\begin{bmatrix} I & & & & \\ -F_0 & I & & & \\ & H_1 & & & \\ & -F_1 & I & & \\ & & H_2 & & \\ & & -F_2 & I & & \\ & & & H_3 & & \\ & & & -F_3 & I & & \\ & & & & H_4 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} \hat{x}_0 \\ 0 \\ z_1 \\ 0 \\ z_2 \\ 0 \\ z_3 \\ 0 \\ z_4 \end{bmatrix}$$
(52)

No caso de existirem múltiplos sensores observando o processo, o sistema de equações é composto de forma similar. Supondo que tivéssemos $s = \{1, 2, 3, 4\}$ sensores observando o processo a cada tempo k = 1, 2, 3, 4 e que esses sensores realizaram M = 6 observações do estado de acordo com o seguinte padrão: para k = 1 o estado foi observado pelos sensores $s = \{1\}$; para k = 2, $s = \{1, 2\}$; para k = 3, $s = \{3, 4\}$ e para k = 4, $s = \{4\}$; a cada um desses sensores lhe corresponderia uma equação de observação :

$$k = 1; s = 1 \rightarrow z_1^1 = H_1^1 x_1$$

$$k = 2; s = 1 \rightarrow z_2^1 = H_2^1 x_2$$

$$k = 2; s = 2 \rightarrow z_2^2 = H_2^2 x_2$$

$$k = 3; s = 3 \rightarrow z_3^3 = H_3^3 x_3$$

$$k = 3; s = 4 \rightarrow z_3^4 = H_3^4 x_3$$

$$k = 4; s = 4 \rightarrow z_4^4 = H_4^4 x_4$$

Com essas observações, o sistema de equações a ser resolvido, em forma de equação matricial seria:

$$\begin{bmatrix} I \\ -F_0 & I \\ & H_1^1 \\ & -F_1 & I \\ & H_2^2 \\ & -F_2 & I \\ & & H_3^3 \\ & & & H_3^4 \\ & & & -F_3 & I \\ & & & & -F_3 & I \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & & \\ & & & & & & & & \\ & & & & & & & & \\ & & & & & & & & \\ & & & & & & & & \\ & & & & & & & & \\ & & & & & & & & \\ & & & & & & & & \\ & & & & & & & & \\ & & & & & & & & \\ & & & & & & & & \\ & & & & & & & & \\ & & & & & & & & \\ & & & & & & & & \\ & & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & \\ & & & & & & \\ & & & & & & \\ & & & & & & \\ & & & & & & \\ & & & & & & \\ & & & & & & \\ & & & & & & \\ & & & & & & \\ & & & & & & \\ & & & & & & \\ & & & & & & \\ & & & & & & \\ & & & & & & \\ & & & & & & \\ & & & & & & \\ & & & & & & \\ & & & & & & \\ & & & & & \\ & & & & & & \\ & & & & & & \\ & & & & \\ & &$$

As equações matriciais (52) e (53) correspondem a um sistema de equações sobredeterminado da forma Ax = b, onde $A \in \mathbb{R}^{m \times n}$ com m > n. Tanto o método de estimação sequencial como o de

estimação por lotes tem como objetivo a solução desse sistema. O método de estimação sequencial soluciona o sistema de forma recursiva. O método de estimação por lotes o soluciona utilizando mínimos quadrados (STRANG, 1986).

O critério de otimização utilizado pelo método de mínimos quadrados é a minimização do quadrado da magnitude do vetor de erros e = Ax - b portanto, o valor estimado corresponde a:

$$\hat{x} = \underset{x}{\operatorname{argmin}} \|e\|^{2} = \underset{x}{\operatorname{argmin}} \left\{ (Ax - b)^{T} (Ax - b) \right\}$$
(54)

Na formulação do sistema Ax = b apresentado tem-se pressuposto que o modelo e as medições são igualmente confiáveis, o que não é realista. As equações do modelo e as equações das medições não são exatas. Além disso, não é razoável pressupor que os erros v(k) e w(k) tem o mesmo tamanho ou que são medidos nas mesmas unidades de medida (STRANG, 1986). Suponhamos que temos m medições que não são igualmente confiáveis. Se b_1 é mais confiável do que b_2 , é razoável tentar diminuir e_1 mais do que e_2 . A forma de fazer isso é atribuir pesos aos componentes do vetor de erro e minimizar $||We||^2 = w_1^2 e_1^2 + w_2^2 e_2^2 + \cdots$. Esse problema é denominado problema de mínimos quadrados ponderados (WLS – Weigthed Least Squares) (STRANG, 1986).

Utilizando a matriz de pesos W, o critério de estimação do vetor de estado é :

$$\hat{x} = \underset{x}{\operatorname{argmin}} \|We\|^{2} = \operatorname{argmin}_{x} \left\{ [W(Ax-b)]^{T} [W(Ax-b)] \right\} \\ = \operatorname{argmin}_{x} \left\{ (Ax-b)^{T} W^{T} W(Ax-b) \right\}$$
(55)

Segundo a teoria de WLS, a matriz $W^T W$ corresponde à inversa da matriz de covariância (STRANG, 1986). Substituindo $W^T W$ pela inversa da matriz de covariância de todo o sistema, Σ^{-1} , na equação (55): :

$$\hat{x} = \underset{x}{\operatorname{argmin}} \left\{ (Ax - b)^T \Sigma^{-1} (Ax - b) \right\}$$
(56)

Na equação (56) pode-se eliminar o termo correspondente a Σ^{-1} e incluí-lo dentro do sistema Ax - b da seguinte forma:

$$(Ax - b)^{T} \Sigma^{-1} (Ax - b) = \left[\Sigma^{-\frac{T}{2}} (Ax - b) \right]^{T} \left[\Sigma^{-\frac{T}{2}} (Ax - b) \right]$$
$$= \left[\Sigma^{-\frac{T}{2}} Ax - \Sigma^{-\frac{T}{2}} b \right]^{T} \left[\Sigma^{-\frac{T}{2}} Ax - \Sigma^{-\frac{T}{2}} b \right]$$
$$= \left[A'x - b' \right]^{T} \left[A'x - b' \right]$$
(57)

Com essa troca de variável, a solução ao problema WLS equivale à solução por OLS do novo sistema A'x = b'.

Na prática, para reformular o sistema (53) como um sistema de WLS basta pré-multiplicar pela esquerda as matrizes F_k e I da equação (50) com $Q_k^{-\frac{T}{2}}$ e a matriz H_k^s e o vetor z_k^s com $R_k^{s^{-\frac{T}{2}}}$. Assim, a cada tempo k o sistema de equações será formado com as seguintes equações:

$$-Q_{k}^{-\frac{T}{2}}F_{k-1}x_{k-1} + Q_{k}^{-\frac{T}{2}}x_{k} = 0$$

$$R_{k}^{s^{-\frac{T}{2}}}H_{k}^{s}x_{k} = R_{k}^{s^{-\frac{T}{2}}}z_{k}^{s}$$
(58)

Com esse resultado, o sistema de multiplos sensores da equação (53) teria a seguinte forma (pressupondo Q(k) = Q e R(k) = R constantes:

$$\begin{bmatrix} P_{0}^{-\frac{T}{2}} & & & & \\ -Q_{1}^{-\frac{T}{2}}F_{0} & Q_{1}^{-\frac{T}{2}} & & & \\ & R_{1}^{1-\frac{T}{2}}H_{1}^{1} & & & \\ & -Q_{2}^{-\frac{T}{2}}F_{1} & Q_{2}^{-\frac{T}{2}} & & \\ & & R_{2}^{1-\frac{T}{2}}H_{2}^{1} & & \\ & & & R_{2}^{2-\frac{T}{2}}H_{2}^{2} & & \\ & & & & R_{3}^{2-\frac{T}{2}}H_{2}^{2} & & \\ & & & & & R_{3}^{2-\frac{T}{2}}F_{2} & Q_{3}^{-\frac{T}{2}} & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & & \\ & & & & & & & & \\ & & & & & & & & \\ & & & & & & & & \\ & & & & & & & & \\ & & & & & & & & \\ & & & & & & & & \\ & & & & & & & & \\ & & & & & & & & \\ & & & & & & & & \\ & & & & & & & & \\ & & & & & & & & \\ & & & & & & & & \\ & & & & & & & & \\ & & & & & & & & \\ & & & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & \\ & & & & & & \\ & & & & & & \\ & & & & & & \\ & & & & & & & \\$$

A solução do sistema A'x = b' mediante OLS da como resultado o vetor $\hat{x}(k)$ que contém o valor estimado de toda a trajetória observada do alvo.

Em relação à qualidade da estimativa, ela segue sendo representada mediante a matriz de covariância de todo o sistema, definida como:

$$\Omega = (A'^T A')^{-1} = (R'^T R')^{-1}$$
(60)

onde os elementos da diagonal são as matrizes de covariâncias das componentes do vetor $\hat{x}(k)$...

Como pode-se observar na equação (60), a matriz de covariância do sistema depende da matriz A' e não das observações obtidas pelos sensores. Portanto, a matriz de covariância Ω pode ser calculada antes ou depois de realizada a estimativa.

Assim como no caso da estimação sequencial, o método de estimação por lotes também pode solucionar problemas não lineares. Para isso, as equações (50) e (51) são linearizadas mediante o método de expansão por expansão de Taylor (THRUN; BURGARD; FOX, 2005). Após a linearização, a formação do sistema de equações Ax = b procede da mesma forma mostrada. A solução do sistema é realizada mediante o metodo de Newton (DENNIS; SCHNABEL, 1996; STRANG, 1986).

3.4 INFERÊNCIA DISTRIBUÍDA APLICADA AO PROBLEMA DE LOCALIZAÇÃO E MAPEA-MENTO COM MÚLTIPLOS ROBÔS UTILIZANDO FATORAÇÃO MULTIFRONTAL *QR*

Na área das redes de sensores sem fio, mais dedicação tem sido dada para desenvolver algoritmos de estimação sequencial distribuído do que para desenvolver algoritmos usando técnicas de estimação por lotes. Porém, em outras áreas da ciência, o método de estimação por lotes tem sido muito estudado. Exemplo disso é a utilização dessa técnica para resolver problemas complexos e de grande escala tais como: *Bundle Adjustment* na área de fotogrametria (BROWN, 1976; GRANSHAW, 1980); *Structure from Motion* na área de visão computacional (FAUGERAS, 1993; SZELISKI; KANG, 1994; HARTLEY; ZISSERMAN, 2000) e Mapeamento e Localização Simultânea total (full SLAM – *full Simultaneous Localization and Mapping*) na área da robótica (DELLAERT; KAESS, 2006).

Particularmente o trabalho apresentado por Dellaert em (DELLAERT; KIPP; KRAUTHAUSEN, 2005) tem inspirado a proposta apresentada nessa tese. Em (DELLAERT; KIPP; KRAUTHAUSEN, 2005), o autor aborda o problema de *full* SLAM com um sistema de múltiplos robôs. Na robótica, denomina-se SLAM ao problema de determinar a localização do robô enquanto, simultaneamente, se realiza o mapeamento do ambiente onde o robô se encontra navegando. *Full* SLAM refere-se ao problema de estimar a trajetória completa do robô e todas as características do ambiente (todo o mapa) simultaneamente. No trabalho apresentado, o objetivo é ainda maior, já que o interesse é determinar o mapa completo e as trajetórias de cada um dos robôs. Esse é um problema de inferência de grande escala devido ao grande número de variáveis desconhecidas e, normalmente, é realizado de forma centralizada com um computador de grande capacidade de processamento.

Na robótica, é normal representar o problema de SLAM mediante algum modelo gráfico como Redes Bayesianas (*Bayesian Networks*), Campos Aleatórios de Markov (*Markov Random Fields*) ou Grafo de Fatores (*Factor Graphs*). Na figura 9 se mostra o problema abordado, representado como um campo aleatório de Markov. Nessa figura, cada quadrado representa uma característica do ambiente ou marco (*landmark*), os círculos coloridos representam a trajetória de 3 robôs navegando pelo ambiente e as arestas mostram os marcos que cada robô observou durante seu trajeto pelo ambiente.

A distribuição de probabilidade conjunta de todo o problema é:

$$\prod_{r=1}^{R} \left\{ p(x_{r0}) \prod_{i=1}^{M} p\left(x_{ri} \mid x_{ri-1}, u_{ri}\right) \right\} \prod_{k=1}^{K} p\left(z_k \mid x_{ri_k}, l_{j_k}\right)$$
(61)

onde x_{ri} representa o estado do robô r no tempo i com $r \in 1 ... R$ e $i \in 0 ... M$, l_j representa os marcos com $j \in 1 ... N$, e z_k representa as medições com $k \in 1 ... K$. Adicionalmente, $p(x_{r0})$ é a distribuição a priori do estado inicial do robô r, $p(x_{ri} | x_{ri-1}, u_{ri})$ é a distribuição do modelo do movimento dos robôs e $p(z_k | x_{ri_k}, l_{j_k})$ é a distribuição do modelo das medições feitas por cada robô.

Considerando x_{r0} conhecida e as distribuições do modelo do movimento dos robôs e das observações gaussianas, o problema representado pela equação (61) pode ser reformulado como um problema de



Figura 9 - Mapeamento e localização simultânea distribuída. (Extraído de (DELLAERT; KIPP; KRAUTHAUSEN, 2005))

mínimos quadrados ordinários da forma (DELLAERT; KIPP; KRAUTHAUSEN, 2005; DELLAERT; KAESS, 2006; KAESS; RANGANATHAN; DELLAERT, 2007):

$$\underset{V}{\operatorname{argmin}} \|AX - b\|_{\Sigma}^{2} \tag{62}$$

onde X é o vetor que contém as trajetórias completas dos robôs e o mapa, A é a matriz principal do problema e tem como característica ser esparsa e \sum a matriz de covariância dos erros de todo o sistema.

O algoritmo distribuído apresentado é baseado no método de fatoração multifrontal QR. A solução da equação (62) é feita pelo método de eliminação de Gauss, o que implica a fatoração da matriz A. O método multifrontal explora a esparsidade da matriz A para fatorá-la mediante uma sequência de fatorações densas de submatrizes da matriz. Essa sequência depende da ordem em que as variáveis do sistema são eliminadas e é representada mediante um grafo do tipo árvore, que na álgebra linear é denominado árvore de eliminação (LIU, 1990). Porém, existem diversas variantes da árvore de eliminação básica para fatoração de matrizes esparsas. No trabalho apresentado por Dellaert utilizase uma estrutura denominada *clique tree* (POTHEN; SUN, 1992), também conhecida como *junction tree* na área da Inteligência Artificial (DELLAERT; KIPP; KRAUTHAUSEN, 2005). Uma propriedade importante do *clique tree* (e da árvore de eliminação e suas variantes) é que as variáveis que se encontram em subárvores disjuntas dentro do grafo podem ser eliminadas de forma paralela. Na solução, o *clique tree* é mapeado na topologia de rede dos robôs. Dessa forma, utilizando o *clique tree* como guia, os robôs realizam a fatoração da matriz R. Logo determina-se a solução do problema

mediante substituição inversa.

Vemos no trabalho apresentado em Dellaert, Kipp e Krauthausen (2005) que um problema esparso pode ser abordado mediante técnicas de estimação por lotes e solucionado em paralelo utilizando fatoração multifrontal QR. O problema de rastreamento de alvos também é um problema esparso, porém se diferencia do problema de SLAM em que:

- 1. Os sensores são fixos em contraste com a mobilidade dos robôs;
- 2. O sistema observado é um alvo móvel, enquanto que os marcos observados no SLAM são fixos;
- 3. No problema de rastreamento não se conhece o sinal de controle que governa a mobilidade do alvo.

Nos próximos capítulos dessa tese mostraremos nossa solução para o problema de rastreamento de alvos com uma RSSF, primeiro explicando como realizar a fatoração multifrontal QR e depois detalhando a nossa proposta de algortimo distribuído baseado nesse metodo de fatoração.

4 FATORAÇÃO MULTIFRONTAL QR

Neste capítulo descreveremos o método de fatoração multifrontal QR. Decidimos utilizar fatoração multifrontal QR para a solução da estimação da trajetoria de alvos com RSSF mediante estimação por lotes devido a que:

- Por ser uma decomposição ortogonal, é extremamente estável e pode ser utilizada em matrizes mal condicionadas (*ill-conditioned*) (STRANG, 1986);
- 2. O método trabalha diretamente na matriz A, o que evita a necessidade da formação de $A^T A$ e $A^T b$ (GOLUB; LOAN, 1996; STRANG, 1986);
- O método permite paralelizar a fatoração da matriz A (BJORCK, 1996; DELLAERT; KAESS, 2006).

A seguir apresentamos o algoritmo básico para realizar a fatoração multifrontal QR utilizado no desenvolvimento de nossa soluçao de estimação de trajetórias distribuído. Ele esta baseado no algoritmo apresentado por Matstoms (1994). Para maior detalhe sobre o metodo multifrontal e outros aspectos relacionados com o algoritmo pode-se consultar as seguintes referências: (LIU, 1992), (GEORGE, 1998), (LIU, 1990),

O método multifrontal foi apresentado em Duff e Reid (1983). Inicialmente foi desenvolvido para dar solução a sistemas lineares simétricos e, posteriormente, para sistemas não simétricos. O primeiro algoritmo para fatoração multifrontal QR foi desenvolvido por Liu (1986b). Nesse trabalho utilizase rotações de Givens (GOLUB; LOAN, 1996) para realizar a decomposição ortogonal. George e Liu (1987) apresentam uma versão modificada do algoritmo de Liu que utiliza transformações de Householder.

O método multifrontal de Duff e Reid explora a estrutura de uma matriz esparsa quadrada para organizar sua decomposição *LU* como uma sequência de fatorações de submatrizes densas. Consiste em 3 passos:

- Fatoração simbólica: Fatoriza-se simbolicamente a matriz A para determinar a estrutura de L^T. A partir dessa estrutura determina-se a relação entre as variáveis do sistema, o que é representado mediante um grafo do tipo árvore, denominado árvore de eliminação.
- 2. Fatoração numérica: Guiada pela árvore de eliminação, determina-se a fatoração LU mediante a fatoração de submatrizes densas de A associadas a cada nó da árvore.
- 3. Solução do sistema: Soluciona-se o sistema utilizando os fatores LU calculados no passo 2.

O método de fatoraçao multifrontal QR baseia-se nas mesmas idéias que o método multifrontal LU, porém, por ser aplicado à solução de sistemas sobredeterminados, ele não realiza a fatoração simbólica da matriz A original. Ao invés disso, realiza a fatoração simbólica da matriz A^TA (AMESTOY; DUFF; PUGLISI, 1996). O método utiliza a observação feita por George e Heath (1980) onde o fator

R determinado na fatoração QR da matriz A de um sistema sobredeterminado (equação (40)) é o mesmo que o fator de Cholesky da matriz $A^T A$ da equação normal (42). Essa relação faz-se evidente se substituirmos (44) em $A^T A$:

$$A^{T}A = \begin{pmatrix} R^{T} & 0 \end{pmatrix} Q^{T}Q \begin{pmatrix} R \\ 0 \end{pmatrix} = R^{T}R$$
(63)

Daqui identificamos que $R = L^T$ é o fator de Cholesky.

Para ilustrar os cálculos feitos em cada passo do método multifrontal QR utilizaremos a matriz (64) como exemplo de matriz esparsa:

Nessa matriz, o símbolo \times representa um valor diferente de zero e os espaços em branco representam zeros. Além disso, assume-se que:

- 1. A matriz A cumpre a propriedade de Hall forte (*strong Hall property*). Essa propriedade assegura a correta predição da estrutura de R (BJORCK, 1996).
- 2. Durante a fatoração simbólica, toda vez que duas quantidades sejam adicionadas ou subtraídas não haverá cancelamento de termos .
- 3. Na fatoração de matrizes esparsas, uma das maiores preocupações é a minimização de preenchimento (*fill-in*) no fator de Cholesky R. A redução do preenchimento é obtida mediante a ordenação de colunas da matriz A. Diversos métodos heurísticos tem sido desenvolvidos para diminuir o preenchimento, entre os mais populares temos o algoritmo de mínimo grau (*Minimum Degree*) (TINNEY; WALKER, 1967) e a ordenação por dissecção (*Nested Dissection*) (GE-ORGE, 1973). Ao longo da explicação assumiremos que a matriz A tenha sido ordenada antes de proceder com a fatoração multifrontal.

A fatoração multifrontal QR de uma matriz retangular A consiste nos seguintes passos:

- 1. Fatoração simbólica: Realiza a fatoração de Cholesky simbólica de $A^T A$ analisando a estrutura da matriz A. O resultado é a estrutura do fator de Cholesky R;
- Determinação da árvore de eliminação: A partir da estrutura de A^TA determina-se a relação entre as variáveis do sistema, o que é representado mediante um grafo do tipo árvore, denominado árvore de eliminação;
- 3. Fatoração numérica: A árvore de eliminação indica a forma de realizar a fatoração QR mediante a fatoração de submatrizes densas associadas a cada nó da árvore.

Esses passos serão detalhados a seguir.

4.1 FATORAÇÃO SIMBÓLICA

O objetivo da fatoração simbólica é predizer a estrutura do fator de Cholesky R, sem necessidade de calcular $A^T A$.

O primeiro passo é determinar a estrutura da matriz $A^T A$ a partir de A. Em geral, o produto de duas matrizes AB pode ser expressado como a somatória do produto externo (*outer product*) de colunas e linhas. Dessa forma, $A^T A$ pode ser expressada como (BJORCK, 1996):

$$A^T A = \sum_{i=1}^m a_i a_i^T \tag{65}$$

onde a_i^T é a i-ésima linha de A.

A equação (65) indica que $A^T A$ é o resultado da soma de matrizes simétricas de característica r = 1. Portanto, considerando que não existe cancelamento de termos, temos que a estrutura de $A^T A$ é a soma direta das estruturas das matrizes $a_i a_i^T$, $i = 1, 2, \dots, m$.

A estrutura de matrizes esparsas normalmente pode ser representada mediante grafos. Um grafo consiste em um conjunto de vértices $X = \{x_1, x_2, \dots, x_n\}$ e um conjunto de arestas E. Dois vértices $x_i e x_j$ podem ser conectados mediante uma aresta não direcionada, $x_i - x_j$ ou uma aresta direcionada, $x_i \rightarrow x_j$. Quando se utilizam arestas não direcionadas, o grafo é denominado grafo não direcionado e o conjunto de arestas E consiste em um conjunto de pares não ordenados (x_i, x_j) . Similarmente, quando se utilizam arestas direcionadas, o grafo é denominado grafo direcionado e E consiste em pares ordenados (x_i, x_j) , tal que (x_i, x_j) representa $x_i \rightarrow x_j$. Nesse caso, $(x_i, x_j) \neq (x_j, x_i)$ devido ao fato de que $x_i \rightarrow x_j \neq x_j \rightarrow x_i$.

A estrutura de uma matriz esparsa simétrica é representada mediante um grafo não direcionado. Por exemplo, na figura 10 mostra-se a matriz simétrica $M \in \mathbb{R}^{n \times n}$ e seu grafo associado. O símbolo × representa um elemento diferente de zero. O grafo de M, G(M) = (X, E), consiste no conjunto de vértices $X = \{1, 2, ..., n\}$ e o conjunto de arestas $(x_i, x_j) \in E$ se e só se $a_{ij} = a_{ji} \neq 0$.



(a) Estrutura de uma matriz simétrica M.



(b) Grafo da estrutura da matriz M

Figura 10 - Representação gráfica de matrizes

Se utilizamos a representação gráfica das estruturas das matrizes $a_i a_i^T$, a partir da equação (65) vemos que $G(A^T A)$, a estrutura de $A^T A$, é a união direta dos grafos $G(a_i a_i^T)$ sem considerar as repetições dos vértices (BJORCK, 1996).

Como exemplo disso, vemos na figura 11 a determinação de $G(A^TA)$ para a matriz de exemplo A (equação (64)). Na figura 11a podemos notar que cada linha de A gera um subgrafo não direcionado totalmente conectado. Esse tipo de grafo é conhecido como *clique* e corresponde a uma submatriz simétrica completa de A^TA . O grafo $G(A^TA)$ (figura 11b) é a união dos cliques formados a partir das linhas de A. A partir do grafo $G(A^TA)$ temos que a matriz A^TA tem a estrutura mostrada na figura 11c.

A estrutura do fator de Cholesky R é determinada a partir do grafo $G(A^T A)$ realizando uma versão gráfica do algoritmo de eliminação de Gauss. Esse algoritmo foi apresentado pela primeira vez por Parter (1961) . A partir de $G(A^T A)$, o algoritmo gera uma sequência de grafos não direcionados, denominados grafos de eliminação, resultantes da eliminação dos vértices. Inicia-se tomando $G_0 = G(A^T A)$. Logo forma-se G_i a partir de G_{i-1} fazendo:

- 1. Eliminação do vértice i e das arestas que o conectam com seus vizinhos;
- 2. Formação de um clique com os vizinhos do vértice *i*.

Após a eliminação de todos os vértices de $G(A^TA)$, forma-se o grafo $G_F(A^TA)$ que é a união de $G(A^TA)$ com as arestas adicionadas no segundo passo do algoritmo. O grafo $G_F(A^TA)$ é denominado grafo preenchido (*filled graph*) de A^TA . Na figura 12 podemos observar o processo de eliminação de



(a) Grafos $G(a_i a_i^T)$ para cada fila de A.



(b) Grafo $G(A^T A)$



(c) Estrutura da matriz $A^T A$

Figura 11 - Determinação da estrutura da matriz $A^T A$ para a matriz de exemplo A.

variáveis e o grafo preenchido $G_F(A^T A)$.

O grafo $G_F(A^T A)$ contém a estrutura de fator de Cholesky R,

$$G(R+R^T) \subseteq G_F(A^T A)$$

porém, sobre a condição de não cancelamento, essa relação é de igualdade.



Figura 12 - Processo de eliminação de variáveis e filled graph de $A^T A$

As novas arestas no grafo $G(A^TA)$, determinadas a partir do processo de eliminação, representam valores diferentes de zero na estrutura da matriz de R. Esses elementos são denominados preenchimento (*fill-in*) e são indesejáveis porque reduzem a esparsidade da matriz R. Uma ordenação adequada de colunas reduz o preenchimento e pode ser alcançada utilizando algoritmos bem conhecidos como por exemplo o algoritmo de mínimo grau ou a ordenação por dissecção.

A estrutura do fator de Cholesky R para a matriz A é:

onde o símbolo o representa preenchimento na matriz.

Outro algoritmo para determinar a estrutura de R foi apresentado por Gilbert, Moler e Schreiber (1992). Ele trabalha diretamente na matriz A. A idéia principal é realizar a eliminação de Gauss diretamente sobre os cliques formados a partir das linhas da matriz A. Toda vez que um vértice i é eliminado, os cliques que contém o vértice i desaparecem e um novo clique é formado com todos os vértices restantes. Dessa forma determinam-se os preenchimentos da fatoração de $A^T A$ sem formar $G(A^T A)$.

4.2 DETERMINAÇÃO DA ÁRVORE DE ELIMINAÇÃO

A árvore de eliminação é uma estrutura importante para a fatoração de matrizes esparsas (LIU, 1990). A árvore é a guia do processo de fatoração. Ela indica como formar as denominadas matrizes frontais. Além disso, sua estrutura mostra a forma de explorar paralelismo na fatoração da matriz A.

A definição da árvore de eliminação para uma matriz simétrica é a seguinte:

Dada uma matriz $A \in \mathbb{R}^{m \times n}$ com $m \ge n$, tal que $A^T A \in \mathbb{R}^{n \times n}$ é simétrica positiva definida. $T(A^T A)$, árvore de eliminação da matriz $A^T A$, é um grafo do tipo árvore que consiste em n vértices, cada um unicamente identificado por um inteiro j = 1, 2, ..., n que corresponde à ordenação de colunas da matriz, na qual o vértice j é pai do vértice i se e só se

$$p = \min\{j > i \mid r_{ij} \neq 0\}$$

$$\tag{67}$$

sendo r_{ij} um elemento diferente de zero no fator de Cholesky R de $A^T A$.

Essa equação indica que j é pai de i se o primeiro elemento diferente de zero da i-ésima linha de R encontra-se na coluna j. A propriedade forte de Hall e a correta predição de R garantem que a árvore de eliminação seja correta (MATSTOMS, 1994).

Na figura 13 observa-se o processo de determinação da árvore de eliminação para $A^T A$.



Figura 13 - Árvore de eliminação de $A^T A$

A árvore de eliminação mostra a relação das dependências das linhas do fator de Cholesky R e pode ser usada como guia do processo de substituição inversa. Além disso, também indica como explorar o paralelismo na fatoração da matriz. O seguinte teorema, extraído de Bjorck (1996), que é a base da fatoração multifrontal, descreve essa propriedade : **Teorema 1.** Seja T[j] a subárvore com raiz no vértice j. As colunas k e j podem ser eliminadas independentemente uma da outra se $k \notin T[j]$

A partir desse teorema temos que se T[i] e T[j] são duas subárvores desconexas na árvore de eliminação e se $s \in T[i]$ e $t \in T[j]$, então as colunas s e t podem ser eliminadas paralelamente. Além do paralelismo, a árvore de eliminação indica a ordem de eliminação de colunas. Para eliminar a *i*-ésima coluna todos seus descendentes na árvore devem ser eliminados.

4.3 FATORAÇÃO NUMÉRICA

Este é o último passo da fatoração multifrontal QR. Para descrever a fatoração numérica, primeiro faremos as seguintes definições:

- 1. j é um vértice na árvore de eliminação assim como é a j-ésima coluna da matriz A;
- A[j] é a matriz formada com as linhas de A que tem o primeiro elemento diferente de zero na coluna j.

Além dessas definições, utilizaremos o operador *extended matrix merge-notation* ($\leftarrow A \rightarrow$) apresentado em (MATSTOMS, 1994). Esse operador é baseado no operador *extend-add operator* (LIU, 1992). Ele permite coletar matrizes de diferentes índices de colunas em uma única matriz. Para isso, adicionam-se colunas zero para fazer a matriz consistente. O conjunto de índices de colunas da matriz resultante é a união dos conjuntos de índices de coluna das matrizes coletadas

O seguinte exemplo ilustra a utilização da notação ($\leftarrow A \rightarrow$). Sejam as matrizes A e B definidas como:

$$A = \begin{bmatrix} a_{11} & 0 & a_{16} \\ a_{21} & a_{23} & 0 \end{bmatrix} \quad B = \begin{bmatrix} b_{13} & b_{14} \\ 0 & b_{24} \\ b_{33} & b_{44} \end{bmatrix}$$

com conjunto de índices de coluna $j_A = \{1, 3, 6\}$ e $j_B = \{3, 4\}$ respectivamente. Então, a matriz $M = \begin{bmatrix} \leftarrow A \rightarrow \\ \leftarrow B \rightarrow \end{bmatrix}$ é formada da seguinte forma:

$$\leftarrow A \rightarrow = \begin{bmatrix} a_{11} & 0 & 0 & a_{16} \\ a_{21} & a_{23} & 0 & 0 \end{bmatrix}$$
$$\leftarrow B \rightarrow = \begin{bmatrix} 0 & b_{13} & b_{14} & 0 \\ 0 & 0 & b_{24} & 0 \\ 0 & b_{33} & b_{44} & 0 \end{bmatrix}$$

$$M = \begin{bmatrix} \leftarrow A \\ \leftarrow B \\ \leftarrow B \\ \end{bmatrix} = \begin{bmatrix} a_{11} & 0 & 0 & a_{16} \\ a_{21} & a_{23} & 0 & 0 \\ 0 & b_{13} & b_{14} & 0 \\ 0 & 0 & b_{24} & 0 \\ 0 & b_{33} & b_{44} & 0 \end{bmatrix}$$

O conjunto de índices de coluna da matriz M é $j_M = j_A \cup j_B = \{1, 3, 4, 6\}.$

A fatoração numérica da matriz A utiliza a árvore de eliminação como guia para fatorar a matriz. A cada vértice da árvore de eliminação é associada uma matriz denominada matriz frontal. A fatoração sistemática das matrizes frontais dá como resultado o fator de Cholesky R. Por isso o nome de método multifrontal.

Para facilitar a explicação da formação das matrizes frontais, podemos ordenar a matriz A da seguinte forma:

$$A = \begin{bmatrix} A[1] \\ \vdots \\ A[n] \end{bmatrix}$$
(68)

com A[j] a submatriz formada pelas linhas de A que tem seu primeiro elemento diferente de zero na coluna j. Isso devido a que o fator de Cholesky R é invariante com a ordem das linhas. Na figura 14 mostra-se a matriz A ordenada segundo (68).



Figura 14 - Matriz A ordenada

O método inicia pelas folhas da árvore de eliminação $T(A^TA)$. Seja j uma folha da árvore. Formase a matriz frontal F_j tomando a matriz A[j] e eliminando as colunas zero; chamaremos a essa matriz $\bar{A}[j]$. No caso das folhas da árvore, A[j] é a única matriz que contribui com a formação da matriz F_j . Logo, realiza-se a fatoração QR da matriz frontal, o que resulta em $F_j = Q_j R_j$. A primeira linha de R_j corresponde à j-ésima linha do fator de Cholesky R. A matriz remanescente de R_j (R_j sem a primeira linha e coluna) é denominada matriz de atualização U_j e será utilizada pelo pai do vértice j para a formação da sua matriz frontal.

Por exemplo, a formação da matriz frontal F_1 a partir de A[1] é:

$$A[1] = \begin{bmatrix} a_{11} & 0 & a_{13} & 0 & 0 & 0 \\ a_{21} & 0 & 0 & 0 & a_{26} & 0 \end{bmatrix} \bar{A}[1] = \begin{bmatrix} a_{11} & a_{13} & 0 \\ a_{21} & 0 & a_{26} \end{bmatrix} \implies F_1 = \begin{bmatrix} \leftarrow \bar{A}[1] \rightarrow \end{bmatrix} = \bar{A}[1]$$

Realizando a fatoração QR dessa matriz temos:

$$F_1 = Q_1 R_1 \implies R_1 = \begin{bmatrix} r_{11} & r_{13} & 0 \\ 0 & u_{13} & u_{16} \end{bmatrix} \implies U_1 = \begin{bmatrix} u_{13} & u_{16} \end{bmatrix}$$

onde $(r_{11} r_{13})$ são os elementos das colunas 3 e 5 da primeira linha do fator de Cholesky R e U_1 a matriz de atualização do vértice 1, que será utilizada pelo pai do nó 1 (nó 3).

Agora, seja j um vértice dentro da árvore de eliminação. Forma-se a matriz frontal F_j :

- 1. Formando $\bar{A}[j]$.
- 2. Coletando, em uma matriz única, $\bar{A}[i]$ e as matrizes de atualização de todos os filhos de j, U_{cn} , utilizando a seguinte fórmula:

$$F_{j} = \begin{bmatrix} \leftarrow \bar{A}[j] \rightarrow \\ \leftarrow U_{c1} \rightarrow \\ \vdots \\ \leftarrow U_{cn} \rightarrow \end{bmatrix}$$

Após isso, realiza-se a fatoração QR de F_i . A primeira linha de R_i corresponderá à *i*-ésima linha de R e o restante à matriz de atualização U_i , que é armazenada para ser utilizada pelo pai do vértice *i*.

Por exemplo, a formação da matriz F_3 seria:

$$\bar{A}[3] = \begin{bmatrix} a_{63} & a_{66} & 0\\ a_{73} & a_{76} & 0\\ a_{83} & 0 & 0\\ a_{93} & 0 & a_{97} \end{bmatrix}$$
$$U_{c1} = U_1 = \begin{bmatrix} u_{13} & u_{16} \end{bmatrix}$$
$$F_3 = \begin{bmatrix} \leftarrow \bar{A}[i] \rightarrow \\ \leftarrow U_1 \rightarrow \end{bmatrix} = \begin{bmatrix} a_{63} & a_{66} & 0\\ a_{73} & a_{76} & 0\\ a_{83} & 0 & 0\\ a_{93} & 0 & a_{97}\\ u_{13} & u_{16} & 0 \end{bmatrix}$$
Fatorizando F_3 teríamos:

$$F_{3} = Q_{3}R_{3} \implies R_{3} = \begin{bmatrix} r_{33} & r_{35} & r_{36} & r_{37} \\ 0 & u_{15} & u_{16} & u_{17} \\ 0 & 0 & u_{26} & u_{27} \\ 0 & 0 & 0 & u_{37} \\ 0 & 0 & 0 & 0 \end{bmatrix} \implies U_{3} = \begin{bmatrix} u_{15} & u_{16} & u_{17} \\ u_{26} & u_{27} \\ u_{37} \end{bmatrix}$$

Na figura 15 mostra-se o processo completo de fatoração multifrontal QR da matriz ${\sf A}$



(a) Matriz de exemplo A e árvore de eliminação



(b) Inicio da fatoração multifrontal QR. Eliminação da coluna 1
 Figura 15 - Fatoração multifrontal QR da matriz de exemplo A.



(c) Eliminação da coluna 3

Figura 15 - Fatoração multifrontal QR da matriz de exemplo A (continuação).



(d) Eliminação das colunas 4, 5, 6. Observa-se que elas podem ser eliminadas em paralelo Figura 15 - Fatoração multifrontal QR da matriz de exemplo A (continuação).



(e) Eliminação dos das colunas 6 e 7 e matriz R final

Figura 15 - Fatoração multifrontal QR da matriz de exemplo A (continuação).

4.4 PARALELISMO NA FATORAÇÃO

Vimos na sessão anterior que o método de fatoração multifrontal permite fatorar uma matriz em forma paralela e que a árvore de eliminação mostra a forma em que essa fatoração deve acontecer. Outra forma de explorar paralelismo, em outra fase da fatoração, é a reordenação de colunas da matriz. O principal objetivo da reordenação de colunas é reduzir o preenchimento no fator de Cholesky *R* e com isso, reduzir trabalho e uso de memória nos processadores. Porém, é bem conhecido que, reordenar a matriz com esse objetivo nem sempre produz uma ordem de colunas que explore eficientemente o paralelismo. Isso pode ser observado claramente na ordenação de colunas de uma matriz tridiagonal. A seguir revisaremos os conceitos básicos das principais heurísticas utilizadas para reordenar as colunas de uma matriz segundo Heath, Ng e Peyton (1991) . Após isso, aplicaremos essas heurísticas na ordenação de um sistema tridiagonal e observaremos o tipo de parelelismo que se consegue. Mais adiante veremos que o estudo da reordenação utilizando essas heurísticas para o sistema tridiagonal será importante para o desenvolvimento do algoritmo colaborativo para redes de sensores distribuídas. Para uma revisão mais detalhada deste tema pode-se consultar (DUFF; VORST, 1999; DEMMEL; HEATH; VORST, 1992).

Para explorar paralelismo em qualquer problema computacional, a tarefa deve ser decomposta em subtarefas que possam ser executadas independentemente. Utilizamos o termo granularidade para denominar o número e tamanho das subtarefas. Uma tarefa dividida em poucas subtarefas de grande

tamanho tem menor granularidade do que se for dividida em muitas subtarefas pequenas.



(a) Aplicando a eliminação de Gauss na matriz M na ordem original obtém-se como resultado 7 preenchimentos.



(b) Aplicando o algoritmo de eliminação na matriz M reordenada mediante o algoritmo de mínimo grau (M'). Nesse caso não ocorre preenchimento

Figura 16 - Reordenação de colunas pelo algoritmo de Mínimo Grau

Segundo Liu (1986a), na fatoração de Cholesky para matrizes esparsas, a partir da árvore de eliminação distingue-se 3 níveis de granularidade que podem ser utilizados para obter paralelismo:

- Paralelismo de granularidade fina, em que cada tarefa é uma única operação de ponto flutuante (flop), tal como multiplicar-adicionar;
- Paralelismo de granularidade média, em que cada tarefa é uma única modificação de uma coluna da matriz;
- 3. Paralelismo de granularidade grossa, em que cada tarefa implica na execução da eliminação de todas as colunas em um subárvore da árvore de eliminação.

A partir dessa definição, conclui-se que o método de fatoração multifrontal permite explorar paralelismo de granularidade média e grossa. Média, porque colunas que pertencem a subárvores disjuntas podem ser eliminadas independentemente e grossa, porque subárvores disjuntas podem ser eliminadas em paralelo.

Além do paralelismo conseguido mediante método multifrontal, pode-se incrementar o paralelismo mediante a reordenação das colunas da matriz. A reordenação das colunas de uma matriz é um problema de natureza combinatória muito difícil de ser resolvido (NP-completo). Por isso, utilizam-se heurísticas para desenvolver algoritmos que permitam limitar o preenchimento durante o processo de fatoração. Os algoritmos heurísticos de maior sucesso dentro dessa área são o algoritmo de Mínimo Grau (*Minimum Degree*) e a ordenação por dissecção (*Nested Dissection*). Eles são os mais utilizados e são a base de diversos algoritmos de ordenação de colunas desenvolvidos atualmente.

O algoritmo de mínimo grau foi desenvolvido por Tinney e Walker (1967). O algoritmo elimina, a cada passo da eliminação de Gauss gráfica, o vértice que tenha menor grau no grafo de eliminação atual. Entende-se vértice de menor grau o vértice que tem menor número de vértices adjacentes no grafo. Um exemplo é mostrado na figura 16. Se a matriz M for fatorada na ordem original, teríamos 7 preenchimentos no fator R (figura 16a), enquanto que, reordenada mediante o algoritmo, não ocorre preenchimento (figura 16b).

Essa simples heurística gera uma ordem de colunas razoavelmente boa e é muito eficiente na maioria dos problemas. Porém, a desvantagem do algoritmo é que a qualidade da ordem resultante é sensível à forma de selecionar vértices quando esses tem o mesmo grau. Na figura 17 mostra-se um exemplo, tomado de (BJORCK, 1996), de uma matriz na qual o algoritmo não consegue uma ordem ótima das colunas. Vemos que os vértices 1, 2, 3, 5, 7, 8, 9 tem mínimo grau, entretanto se escolhermos eliminar o vértice 5 primeiro, produz-se preenchimento na posição (4, 6).



Figura 17 - Matriz H na qual o algoritmo de mínimo grau não é ótimo



(a) Estrutura de blocos resultante da aplicação do algoritmo para M1 (um nível dissecção).

Árvore de eliminacão



(b) Estrutura de blocos resultante para M2 (2 níveis de dissecção)

Figura 18 - Reordenação de colunas utilizando ordenação por dissecção

O algoritmo de ordenação por dissecção, desenvolvido por George (1973) baseia-se no conceito "dividir e conquistar".

A idéia básica do algoritmo é a seguinte: seja uma matriz M com grafo associado G(M), seja S o conjunto de vértices do grafo G(M) (denominados separadores), os quais, ao serem removidos junto com suas respectivas arestas, dividem G(M) pelo menos em dois subgrafos. Se a matriz é reordenada de forma que as variáveis em cada parte sejam numeradas de forma contígua, e as variáveis em S colocadas no final, a matriz resultante terá uma forma diagonal em blocos com borda (*bordered block diagonal*). Essa idéia pode ser aplicada recursivamente, dividindo o grafo em subgrafos cada vez

menores. Na figura 18 mostramos o efeito do algoritmo nas matrizes M1 e M2. Na figura 18a o grafo G(M1) foi divido em 2 (dissecção de um nível), enquanto que, em 18b vemos que G(M2) foi dividido em 4 (dissecção em dois níveis). Nelas mostra-se também a estrutura em blocos que gera o algoritmo e a árvore de eliminação resultante.

A caraterística mais importante da ordenação por dissecção é que a eliminação de um vértice (coluna da matriz) dentro de um bloco não gera preenchimento nos outros blocos. Se existe preenchimento, esse aparecerá nos respectivos blocos ou nas bordas. Isso pode ser observado na figura 19 que é um exemplo da ordenação resultante utilizando o algoritmo na matriz M da figura 16.



Figura 19 - Exemplo da utilização de ordenação por dissecção na matriz M da figura 16

A eficiência do algoritmo em limitar o preenchimento depende do tamanho dos separadores usados para dividir o grafo. Para problemas muito regulares, como por exemplo problemas de diferenças finitas em 2 dimensões ou para problemas resolvidos pelo método de elementos finitos, a ordenação por dissecção dá bons resultados. Para problemas muito irregulares com menos conectividade local, o algoritmo é menos efetivo. Não obstante, a ordenação por dissecção é um método importante por suas propriedades de otimização de preenchimento, o que serve de referência teórica para avaliar a qualidade de outros algoritmos de ordenação.

Apesar do exposto, ordenar as colunas de uma matriz com o intuito de minimizar o preenchimento não provê sempre uma ordenação adequada para explorar paralelismo na fatoração. A forma mais fácil de ilustrar isso é observar o efeito de aplicar os algoritmos de mínimo grau e ordenação por dissecção a uma matriz tridiagonal.

Denominaremos ordem natural da matriz a ordem que preserva sua estrutura tridiagonal (figura 20). Fatorar uma matriz tridiagonal em sua ordem natural não gera preenchimento. A árvore de eliminação dessa ordenação é uma cadeia e não oferece alguma possibilidade de paralelismo.



Figura 20 - Matriz tridiagonal na ordem natural, fator de cholesky e árvore de eliminação

Se utilizamos o algoritmo de mínimo grau é muito provável que a ordem resultante seja idêntica à ordem natural ou uma ordem equivalente da qual a árvore de eliminação continue sendo uma cadeia. Ainda assim, existem algumas ordenações geradas por esse algoritmo que permite conseguir paralelismo. Na figura 21 pode observar-se uma dessas ordenações. A ordem resultante não gera preenchimento e a árvore de eliminação tem 2 subárvores.



Figura 21 - Matriz tridiagonal ordenada mediante o algoritmo de mínimo grau

O algoritmo de ordenação por dissecção tem melhor resultado quando se trata de paralelismo. A figura 22 mostra a matriz ordenada segundo esse algoritmo. Utilizando ordenação por dissecção vemos que a árvore de eliminação resultante permite explorar maior paralelismo do que as outras ordenações.

A altura da árvore de eliminação é aproximadamente $log_2(n)$, sendo n o número de colunas. Essa altura é muito menor do que a altura n - 1 da árvore da matriz na ordem natural. Entretanto, há um custo por esse paralelismo: fatorar a matriz produz preenchimento e portanto maior trabalho na fatoração. No entanto, o tempo utilizado para fatorar a matriz diminui, sendo idealmente $O(\log n)$ enquanto que o tempo para fatorar a matriz na ordem natural é O(n).

Esses exemplos ilustram o fato, já mencionado, de que ordenar a matriz com o intuito de minimizar o preenchimento não é o mais apropriado quando a intenção é explorar paralelismo. Porém, ainda não se tem desenvolvido métricas adequadas para mesurar a qualidade das ordenações para fatoração em paralelo.

O algoritmo colaborativo proposto neste trabalho visa explorar paralelismo ao fatorar a matriz principal do sistema. Veremos mais adiante que, coincidentemente, a matriz principal do problema abordado é tridiagonal. É por isso que na nossa solução utilizaremos o algoritmo de ordenação por dissecção para explorar o máximo possível o paralelismo na fatoração.



Figura 22 - Matriz tridiagonal ordenada mediante ordenação por dissecção

5 ALGORITMO COLABORATIVO

O problema abordado neste trabalho é a estimação da trajetória de um alvo móvel dentro de uma rede de sensores sem fio. O algoritmo proposto soluciona o problema de forma *offline*. Isso quer dizer que a estimação é realizada após todos os dados terem sido coletados.

A solução consta de 3 etapas:

- Coleta de dados;
- Fatoração multifrontal QR;
- Solução do sistema utilizando substituição inversa.

Em resumo, a rede de sensores estimará a trajetória do alvo, solucionando o sistema de equações Ax = b, onde x é o vetor das posições do alvo durante sua trajetória dentro da rede de sensores. Na etapa de coleta de dados, a rede observará a posição do alvo enquanto este se encontre dentro do seu alcance de sensoriamento. Uma vez terminada a coleta de dados, a rede inicia a estimação da trajetória. Para isso a rede realiza a fatoração multifrontal QR da matriz principal A da seguinte forma: Primeiro o nó sorvedouro utilizará ordenação por dissecção para ordenar as colunas da matriz A e obter uma árvore de eliminação que permita efetuar a fatoração em paralelo. Obtida a árvore de eliminação, o nó sorvedouro determinará, para cada vértice da árvore, qual é o conjunto de sensores que tem a informação necessária para formar as matrizes frontais de cada vértice. Dentro desses grupos se escolherá um nó líder que deve formar a matriz frontal F_i utilizando os dados dos membros de seu grupo e fatorá-la utilizando decomposição QR. O resultado dessa fatoração é a matriz $\begin{bmatrix} r_{i,*} \\ U_i \end{bmatrix}$, da qual a primeira linha do resultado $(r_{i,*})$ será enviada ao sorvedouro e a matriz U_i será propagada de filhos para pai até chegar à raiz da árvore. Uma vez terminada a fatoração da matriz A, o sorvedouro estima a trajetória do alvo solucionando o sistema de equações $Rx = Q^T b$, cuja solução é direta mediante substituição inversa.

5.1 CONSIDERAÇÕES PRELIMINARES

Na explicação a seguir se pressupõe o seguinte:

- Os movimentos do alvo se limitam ao plano de 2 dimensões;
- A posição dos sensores da rede de sensores sem fio é conhecida antes de iniciar a aplicação. A
 posição pode ser obtida mediante planejamento prévio à aplicação e posicionamento dos nós por
 operadores humanos ou porque os nós sensoriais possuem GPS ou porque a rede utilizou um
 algoritmo de localização antes da aplicação;
- A rede de sensores encontra-se sincronizada antes da aplicação;

- Os nós sensoriais podem obter a posição global do alvo dentro do alcance da rede;
- Na prática, as equações do modelo dinâmico e das observações são não lineares. Na explicação assume-se que o sistema tenha sido linearizado com algum método apropriado antes de dar-lhe solução. Além disso, já que o problema é solucionado utilizando o método de mínimos quadrados, assume-se que as equações estão devidamente ponderadas utilizando as matrizes de covariância segundo o explicado na sessão 3.3.1.2;
- Na teoria de fatoração de matrizes esparsas utiliza-se o símbolo j para designar o índice de coluna de uma matriz retangular. Porém, para facilitar o entendimento da solução apresentada neste trabalho se utilizará o símbolo x_i tanto para designar os índices de coluna da matriz principal A assim como os estados observados do alvo. Isso porque Ax̂ = b é um sistema de equações e a cada coluna de A lhe corresponde uma das variáveis de estado observadas x_i.

5.2 FORMULAÇÃO DO PROBLEMA

Nesta sessão apresentamos a formulação do problema de estimação de trajetorias com redes de sensores sem fio. A formulação apresentada foi adaptada para redes de sensores sem fio a partir da formulação apresentada por Dellaert em (DELLAERT; KAESS, 2006).

Para formular o problema utilizaremos redes bayesianas (PEARL, 1988). As redes bayesianas são representações gráficas, eficientes e compactas, das relações de independência condicional das variáveis do problema.

A rede bayesiana correspondente ao problema de estimação de trajetorias é mostrada na figura 23. Nessa figura, alem da rede bayesiana, mostramos tambem os nós sensoriais da RSSF, representados com circulos vermelhos. Dentro dos sensores temos colocado os nós da rede bayesiana correspondentes as observações que eles realizam, para mostrar que as observações do alvo encontram-se espalhadas nos sensores da rede.

A distribuição de probabilidade conjunta que representa a rede bayesiana da figura 23 é:

$$P(X,Z) = P(x_0) \prod_{k=1}^{K} P(x_k \mid x_{k-1}) \prod_{m=1}^{M} P(z_{m_k} \mid x_k)$$
(69)

onde $x_k = x(k)$ representa o estado do alvo no tempo k, $X = \{x_k\}$, com k = 1, 2, ..., K, é o vetor que contem todos os de estados do alvo até o tempo K, $z_{m_k} = z_m(k)$ representa a medição m do estado do alvo no tempo k, $Z = \{z_{m_k}\}$, com m = 1, 2, ..., M, é o vetor de todos as medições realizadas pelos sensores que observaram o alvo durante a aplicação, $P(x_0)$ é a distribuição a priori do estado inicial x_0 , $P(x_k \mid x_{k-1})$ é a distribuição do modelo de movimento do alvo e $P(z_{m_k} \mid x_k)$ corresponde a função de verossimilhança das observações dos sensores.

Como é usual na literatura de estimação, consideramos que as distribuições são gaussianas. As equações que descrevem o movimento do alvo e as observações são as mesmas descritas na seção 3.3.1.1 para o filtro de Kalman, que aqui reescrevemos por conveniência:



Figura 23 - Rede bayesiana correspondente ao problema de estimação de trajetória com redes de sensores sem fio

$$x_k = F x_{k-1} + v_k \tag{70}$$

$$z_{m_k} = Hx_k + w_{m_k} \tag{71}$$

sendo $v_k = v(k)$ e $w_{m_k} = w_m(k)$ são variáveis aleatórias gaussianas com média zero, não correlacionadas no tempo e matrizes de covariância conhecidas $Q_k = Q(k)$ e $R_k = R(k)$ respectivamente.

Utilizando as equações 70 e 71, as distribuições de probabilidade $P(x_k | x_{k-1})$ e $P(z_{m_k} | x_k)$ ficam definidas como:

$$P(x_{k} \mid x_{k-1}) \propto exp - \frac{1}{2} \left\| Q_{k}^{-\frac{T}{2}} (Fx_{k-1} - x_{k}) \right\|^{2}$$
(72)

$$P(z_{m_k} \mid x_k) \propto exp - \frac{1}{2} \left\| R_k^{-\frac{T}{2}} (Hx_k - z_{m_k}) \right\|_{R_k}^2$$
(73)

No problema estamos interessados em determinar a trajetória inteira do alvo dentro da rede de sensores. Para estimar a trajetória utilizamos o estimador Máximo a Posteriori (Maximum A Posteriori – MAP) para toda a trajetória $X = \{x_k\}$ dadas todas as medições dos sensores $Z = \{z_{m_k}\}$.

O estimador MAP \hat{X} é obtido minimizando o log negativo da probabilidade conjunta P(X, Z):

$$\hat{X} \triangleq \operatorname{argmax}_{X} P(X|Z) \\
= \operatorname{argmax}_{X} P(X,Z) \\
= \operatorname{argmin}_{X} - \log P(X,Z)$$
(74)

Utilizando (72) e (73) em (74) temos que:

$$\hat{X} \triangleq argmin\left\{\sum_{k=1}^{K} \left\| Q_k^{-\frac{T}{2}} (Fx_{k-1} - x_k) \right\|^2 + \sum_{m=1}^{M} \left\| R_k^{-\frac{T}{2}} (Hx_k - z_{m_k}) \right\|^2 \right\}$$
(75)

Na equação (75), a distribuição a priori $P(x_0)$ não aparece porque pressupomos que x_0 é conhecida e, por tanto, $P(x_0)$ é constante.

(a) Grafos $G(a_i a_i^T)$ para cada linha da matriz A

(c) Estrutura da matriz $A^T A$

Figura 24 - Estrutura da matriz $A^T A$ para o problema de estimação de trajetórias

Vemos tambem nessa equação, que o problema corresponde a um problema de mínimos quadrados. Para visualizar melhor isso, devemos lembrar o critério da solução do problema de mínimos quadrados ponderados (vide seção (3.3.1.2)) que é:

$$\hat{x} = \operatorname{argmin} \|We\|^2 \tag{76}$$

Associando os termos, vemos que:

- 1. $(Fx_{k-1} x_k)$ corresponde ao erro do processo;
- 2. $(Hx_k z_{m_k})$ corresponde ao erro das observações;
- 3. $Q_k^{-\frac{T}{2}}$ e $R_k^{-\frac{T}{2}}$ são as matrizes de ponderação dos erros;

Portanto, a equação (75) diz que a trajetoria \hat{X} é o resultado da minimização da somatoria do quadrado dos erros de todo sistema, o que claramente corresponde à definição de um problema de mínimos quadrados ponderados.

Como vimos na seção 3.3.1.2, o problema representado pela equação 75 pode ser reformulado como um problema de mínimos quadrados ordinários, premultiplicando pela esquerda as matrizes de ponderação pelas matrizes F e H e juntando tudo num sistema de equações único da forma Ax = b. Um exemplo desse sistema de equações obtido com esta formulação e da estrutura da matriz A resultante é mostrado na equação (59) da seção 3.3.1.2.

Outra caraterística importante do sistema de equações final Ax = b é a estrutura da matriz $A^T A$ do sistema de equações normais. Essa estrutura corresponde a uma matriz tridiagonal em blocos, já que as componentes de x são os vetores de estado correspondentes à trajetória do alvo. A estrutura de $A^T A$ pode ser obtida determinando o grafo $G(A^T A)$ utilizando o método mostrado na seção 4.1. Isso é mostrado na figura 24 onde se determina a estrutura de $A^T A$ para a matriz A da equação (59) da seção 3.3.1.2. Também nessa estrutura podemos verficar a condição de Markov. O estado x_i (coluna i) esta relacionado com o estado x_{i-1} (coluna i-1) e com o estado x_{i+1} (coluna i+1), que é condição necessária para o funcionamento do filtro de Kalman.

5.3 CENÁRIO EXEMPLO DO PROBLEMA

Para facilitar a explicação de cada etapa da solução será utilizado como exemplo o cenário mostrado na figura 25. Na figura 25a se observa uma rede de sensores sem fio e um alvo dentro do alcance dos sensores. A rede tem observado 7 estados do alvo no momento de estimar a trajetória. Na figura 25b observa-se a estrutura da matriz principal do problema, matriz *A*. Nessa matriz, as primeiras 7 linhas correspondem às equações do modelo dinâmico do alvo. As linhas restantes correspondem às observações dos sensores.

5.4 COLETA DE DADOS

Durante a coleta de dados, os nós atuam como se fossem sensores binários. A cada tempo de amostragem t_i , os nós que detectarem o alvo enviarão 1 bit ao sorvedouro. Com essa informação o sorvedouro criará uma tabela, que denominaremos *Tabela de Estados Observados*. Nessa tabela o sorvedouro tem a informação dos grupos de sensores que observaram cada um dos estados do alvo.

Para o cenário estudado, a tabela 1 mostra os estados observados pelos 4 sensores.



(b) Matriz principal resultante



5.5 FATORAÇÃO MULTIFRONTAL QR

Uma vez terminada a coleta de dados, inicia-se a estimação da trajetória do alvo. O problema é modelado como um grande sistema Ax = b. Para solucionar o problema a rede realizará a fatoração

	x_1	x_2	x_3	x_4	x_5	x_6	x_7
S_1	1	1	1				
S_2		1	1	1	1		
S_3				1	1		
S_4					1	1	1

Tabela 1 - Tabela de estados observados

da matriz A utilizando o método multifrontal QR. A fatoração será feita em 3 pasos:

- Determinação da árvore de eliminação;
- Formação de grupos de sensores para fatoração multifrontal QR;
- Fatoração numérica da matriz.

As duas primeiras etapas são realizadas pelo sorvedouro, enquanto que a fatoração é realizada pelos nós sensoriais.

5.5.1 Determinação da árvore de eliminação

A partir do estudo do problema, a estrutura da matriz A é conhecida. Também conhecemos $G(A^TA)$, grafo da matriz principal da equação normal do sistema $A^TA\hat{x} = A^Tb$, e a partir desse grafo, a árvore de eliminação da matriz principal, T(A). Vimos na seção 4.4 que com a ordem de colunas original T(A) não provê paralelismo na fatoração da matriz e que, para atingir paralelismo, deve-se reordenar as colunas da matriz A. Consequentemente, na solução utilizaremos o algoritmo de ordenação por dissecção para reordenar as colunas da matriz, porque para matrizes tridiagonais (matriz A^TA da equação normal) a árvore de eliminação resultante da reordenação provê um bom grau de paralelismo na fatoração.

Portanto, nesta etapa o sorvedouro determinará uma nova ordem de colunas para a matriz Autilizando ordenação por dissecção. Como resultado disso o novo sistema a ser solucionado é $A'\hat{x}' = b$ onde a matriz A' corresponde à matriz A reordenada e tem como árvore de eliminação T(A'), e o vetor \hat{x}' é o vetor da estimação \hat{x} também reordenado segundo a nova ordem de colunas. Além disso, durante o processo de determinação da árvore, a estrutura da matriz R da decomposição A' = QRé obtida. Logo, o sorvedouro reservará espaço na sua memória para essa matriz.

Cabe notar aqui que, para determinar T(A') não é necessário construir a matriz simbólica de A'. O único conhecimento necessário é o grafo $G(A^TA)$.

Na figura 26 vemos a árvore de eliminação resultante para o exemplo estudado.



Figura 26 - Árvore de eliminação para cenário estudado



Figura 27 - Matriz principal do problema ordenada mediante o algoritmo de ordenação por dissecção

Essa árvore corresponde a ordenação de colunas $\{1, 3, 5, 7, 2, 6, 4\}$, portanto, a matriz A' tem a estrutura mostrada na figura 27.

5.5.2 Formação de grupos

Uma vez determinada T(A'), o sorvedouro designará, para cada vértice da árvore de eliminação, um grupo de nós sensoriais que se encarregará de formar as matrizes frontais durante o processo de fatoração. Já que cada vértice de T(A') representa um estado observado do alvo, o grupo de sensores é formado por todos os nós que observaram o estado correspondente ao vértice da árvore. Essa informação é obtida durante a etapa de coleta de dados e está contida na Tabela de Estados Observados (tabela 1). Na figura 28 observa-se os grupos determinados para o cenário estudado.



Figura 28 - Grupo de sensores para cada vértice da árvore T(A')

Porém, os dados para formar as matrizes frontais estão distribuídos nos sensores do grupo. Portanto, o sorvedouro designará um nó líder em cada grupo, que se encarregará de coletar os dados observados pelos outros sensores do grupo e formar as matrizes frontais de cada vértice. São os nós líderes do grupo os que realizarão a fatoração multifrontal *QR*, seguindo como guia a árvore de eliminação.

Nessa tese não tratamos o tema da seleção do líder dos grupos. Para execução do algoritmo qualquer nó do grupo de sensores pode ser o líder que realize a fatoração. Existem diversos estudos na área das RSSF que abordam o problema de seleção de nó líder e que visam desenvolver protocolos distribuídos para a seleção dos líderes de grupo (ZHAO; GUIBAS, 2004). Porém, no caso do algoritmo apresentado, a escolha pode ser feita pelo sorvedouro da RSSF, tomando como critério de escolha a minimização da comunicação entre os nós na hora da fatoração. Para isso, podemos observar que dentro de cada grupo de nós sensoriais não é possível minimizar a comunicação devido a que, para qualquer nó escolhido como líder, os outros nós do grupo devem comunicar-se com ele para enviar seus dados. No entanto, a comunicação entre líderes de grupo pode sim ser otimizada. Para isso procura-se, entre todas as possíveis combinações de líderes de grupo, qual é a que utiliza menor número de mensagens desde as folhas até o vértice raiz da árvore. No caso de encontrar combinações com o mesmo número de mensagens, o desempate pode ser feito escolhendo a última combinação encontrada. O resultado desse processo seria enviado aos nós da rede para a execução da fatoração multifrontal QR da matriz.

A figura 29 mostra as duas possíveis combinações que requerem menor comunicação entre sensores para o cenário de exemplo. Na explicação a seguir é pressuposto que se esta trabalhando com a combinação mostrada na figura 29b.



Figura 29 - Duas possíveis escolhas de sensores líderes de grupo que minimizam a comunicação da rede

5.5.3 Fatoração numérica da matriz

Após determinar a nova ordem das colunas, a árvore de eliminação T(A') e os grupos de sensores, o sorvedouro envia essa informação aos nós líderes para que possam agrupar-se e iniciar o processo de fatoração numérica da matriz.

Para realizar a fatoração multifrontal QR da matriz A', cada nó líder de grupo deve formar a matriz frontal F_{x_i} correspondente a seu vértice em T(A') e fatorá-la utilizando decomposição QR.

Além disso, para a solução do problema, a RSSF deve calcular tambem o vetor $Q^T b$. Já que o vetor b também encontra-se espalhado nos sensores da rede, os nós líderes tambem podem realizar essa tarefa. Para isso, basta adicionar o vetor b como última coluna da matriz A'. Dessa forma, a matriz que RSSF fatorizara é a matriz aumentada (A' | b). (GOLUB; LOAN, 1996; DELLAERT; KAESS, 2006).

Portanto, no que resta desse capítulo, para facilitar as explicações, nas equações em que nos referimos à matriz A' estará subentendido que a matriz que a rede esta fatorando é a matriz aumentada (A' | b). Consequentemente, nas equações e explicações que se referem à formação das matrizes frontais F_{x_i} , tambem estará subentendido que se está trabalhando com a matriz aumentada $(F_{x_i} | b(F_{x_i}))$, sendo $b(F_{x_i})$ as contribuições do vetor b à matriz F_{x_i} . Já nas figuras que explicam o método, estamos incluindo o vetor b para mostrar que ele também está sendo processado.

Coleta de Dados	Matriz Obtida				
$x_1 \rightarrow \{S_1\}$	$\begin{pmatrix} x_1 & b \\ O_{x_1}^{S_1}[t_1] & b_8 \end{pmatrix}$				
$\begin{array}{c c} x_3 \rightarrow \{S_2\} \\ \begin{pmatrix} O_{x_3}^{S}[t_3] & b_{10} \end{pmatrix} \\ & S_1 \end{array}$	$ \begin{array}{ccc} x_{3} & b \\ O_{x_{1}}^{S_{1}}[t_{3}] & b_{10} \\ O_{x_{1}}^{S_{2}}[t_{3}] & b_{12} \end{array} $				
$ \begin{array}{c} x_5 \rightarrow \{S_4\} \\ \begin{pmatrix} O_{x_5}^{S_2}[t_5] & b_{14} \end{pmatrix} \\ S_2 \\ S_3 \end{array} \begin{pmatrix} O_{x_6}^{S_4}[t_5] & b_{16} \end{pmatrix} $	$\begin{array}{ccc} x_5 & b \\ O_{x_5}^{S_2}[t_5] & b_{14} \\ O_{x_5}^{S_3}[t_5] & b_{16} \\ O_{x_6}^{S_4}[t_5] & b_{17} \end{array}$				
$x_7 \rightarrow \{S_4\}$	$ \begin{pmatrix} x_7 & b \\ O_{x_7}^{S_4}[t_7] & b_{19} \end{pmatrix} $				
$\begin{array}{c c} x_2 \rightarrow \{S_2\} \\ \begin{pmatrix} O_{x_2}^{S_1}[t_2] & b_9 \end{pmatrix} \\ S_1 \end{array}$	$\begin{array}{ccc} x_{2} & b \\ \left(O_{x_{1}}^{S_{1}}[t_{2}] & b_{9} \\ O_{x_{2}}^{S_{2}}[t_{2}] & b_{11} \end{array} \right)$				
$x_6 \rightarrow \{S_4\}$					
$\begin{array}{c c} \hline x_4 \rightarrow \{S_2\} \\ \hline (O_{x_4}^{S_4}[t_4] & b_{15}) \\ \hline S_3 \end{array}$	$\begin{array}{ccc} x_4 & b \\ O_{x_4}^{S_2}[t_4] & b_{13} \\ O_{x_4}^{S_1}[t_4] & b_{15} \end{array}$				

Vértice → {Nó líder de grupo}

Figura 30 - Processo de coleta de dados dos líderes de grupo

Como primeiro passo para formar as matrizes F_{x_i} os nós líderes de grupo vão coletar as observações obtidas pelos nós do seu grupo. Na figura 30 observa-se as submatrizes obtidas pelos nós líderes após coletar as observações feitas pelos membros de seus grupos para o cenário estudado.

Após a coleta, o nó deve determinar qual é a contribuição das equações do modelo dinâmico para as matrizes F_{x_i} .

Se o problema fosse solucionado em uma estação de processamento centralizada, as matrizes F_{x_i} seriam formadas, primeiro ordenando as linhas da matriz A' para determinar as submatrizes $A'[x_i]$ e logo, juntando em uma matriz as submatrizes $\bar{A}'[x_i]$ (que é a mesma submatriz $A'[x_i]$ com as colunas zero eliminadas) com as matrizes de atualização U_{x_k} dos filhos do vértice x_i (se existissem), mediante o operador *extended matrix merge-notation* ($\leftarrow A \rightarrow$) (vide capítulo 4). Nas figuras 31a e 31b observa-se as matrizes aumentadas $(A \mid b)$ e $(A' \mid b)$ para o cenário estudado. Nelas vemos as linhas correspondentes ao modelo dinâmico do sistema e as observações do sensores da rede.

Já na figura 32 observa-se a matriz (A' | b) ordenada para visualizar as submatrizes $A'[x_i]$ para o cenário estudado, onde $C_{x_i}[t_i]$ representa o coeficiente de x_i no tempo t_i das equações do modelo dinâmico e $O_{x_i}^{S_s}[t_i]$ o coeficiente da observação de x_i no tempo t_i feita pelo sensor S_s .

Para construir a matriz F_{x_i} , cada nó líder deve calcular a submatriz $\bar{A}'[x_i]$. Essa submatriz é obtida a partir de $A'[x_i]$. Na figura 32 observamos que algumas submatrizes $A'[x_i]$ estão constituídas por uma contribuição do modelo dinâmico e uma contribuição das observações de x_i . Outras estão constituídas só com a contribuição das observações de x_i . Essa característica sugere uma forma de determinar $\bar{A}'[x_i]$:

- Cada nó líder de grupo pode simular o modelo dinâmico do sistema para extrair sua contribuição à submatriz A'[x_i] e depois adicionar as linhas correspondentes às observações do estado x_i obtidas a partir do seu grupo de nós sensoriais;
- No caso de não existir contribuições do modelo, A'[xi] é formada só com as contribuições das observações de xi.

Isso é possível porque o modelo dinâmico é um conhecimento prévio à aplicação e pode ser simulado em qualquer momento pelos nós líderes de grupo. Já as observações dos estados x_i acontecem durante a aplicação e são coletadas pelos líderes dos grupos antes de iniciar a fatoração.

Seja a submatriz $A'_{modelo}[x_i]$ a contribuição das equações do modelo dinâmico à submatriz $A'[x_i]$ e $\bar{A}'_{modelo}[x_i]$ a matriz $A'_{modelo}[x_i]$ com as colunas zero eliminadas. Na figura 33 mostram-se as matrizes A_{modelo} e A'_{modelo} para o cenário estudado, ordenadas para visualizar suas submatrizes $A_{modelo}[x_i]$ e $A'_{modelo}[x_i]$ respectivamente.

O tamanho de A_{modelo} depende do número de estados observados durante toda a aplicação, portanto pode ser muito grande e não seria conveniente para um nó sensorial ter que trabalhar com ela. Porém, não é necessário formar toda a matriz A_{modelo} para que os líderes de grupo possam determinar as contribuições do modelo dinâmico às submatrizes $\bar{A}'[x_i]$.

Observa-se na figura 33 que antes da ordenação existem as submatrizes $A_{modelo}[x_i]$ para as colunas $x_i = x_1, x_2, x_3, x_4, x_5, x_6$. Porém, depois da ordenação, na matriz $(A'_{modelo} \mid b)$ só existem as submatrizes $A'_{modelo}[x_i]$ para $x_i = x_1, x_3, x_5, x_7$. Portanto, não basta ordenar $A_{modelo}[x_i]$ para obter $A'_{modelo}[x_i]$ porque, por definição, as submatrizes $A_{modelo}[x_i]$ e $A'_{modelo}[x_i]$ são formadas pelas linhas que têm o primeiro elemento diferente de zero na coluna x_i e ao modificar a posição da coluna x_i , seus coeficientes mudam de posição podendo ficar primeiros, no meio ou últimos em relação aos demais coeficientes da linha, o que determina a existência ou não da submatriz.



(b) Matriz $(A' \mid b)$

Figura 31 - Matrizes ampliadas correspondentes ao cenário estudado

Para determinar se existe a submatriz $A'_{modelo}[x_i]$ devemos observar qual é a posição dos elementos da coluna x_i em relação aos outros coeficientes das linhas após a ordenação. Esses outros coeficientes

	x_{I}	<i>x</i> ₃	x_{5}	<i>x</i> ₇	x_{2}	x_{6}	$x_{_{\mathcal{A}}}$	b	
Modelo —	$C_{x_1}[t_1]$ $C_{x_1}[t_2]$				$C_{x_2}[t_2]$			b_1 b_2	$A'[x_i]$
Observações	$O_{x_1}^{S_1}[t_1]$							b_8	
Modelo —		$C_{x_3}[t_3] \\ C_{x_3}[t_4]$			$C_{x_2}[t_3]$		$C_{x_4}[t_4]$	b_3 b_4	415 1
Observações —		$O_{x_3}^{S_1}[t_3] \ O_{x_3}^{S_2}[t_3]$						b ₁₀ b ₁₂	$A^{T}[x_{3}]$
Modelo —			$C_{x_5}[t_5]$ $C_{x_5}[t_6]$			$C_{x_{6}}[t_{6}]$	$C_{x_4}[t_5]$	b_5 b_6	
(A' b) =			$O_{x_5}^{S_2}[t_5]$					b ₁₄	$A'[x_5]$
Observações —			$O_{x_5}[t_5] O_{x_5}^{S_4}[t_5]$					_{b16} b ₁₇	
Modelo —	_			$C_{x_7}[t_7]$		$C_{x_6}[t_7]$		b_7	4/[m]
Observações —				$O_{x_7}^{S_4}[t_7]$				b ₁₉	$\begin{bmatrix} A & x_6 \end{bmatrix}$
Observações —					$O_{x_2}^{S_1}[t_2] \\ O_{x_2}^{S_2}[t_2]$			b ₉ b ₁₁	$A'[x_{7}]$
Observações	-					$O_{x_6}^{S_4}[t_6]$		b ₁₈	$A'[x_6]$
Observações —							$O_{x_4}^{S_2}[t_4] O_{x_4}^{S_3}[t_4]$	b ₁₃ b ₁₅	$A'[x_4]$

Figura 32 - Matriz ampliada $(A' \mid b)$ ordenada para visualizar as matrizes $A'[x_i]$

são os correspondentes as variáveis (colunas) com as quais x_i esta relacionada, já que, seja qual for a ordenação de colunas utilizada, a relação entre as variáveis do sistema não muda. Logo, os coeficientes com os quais está relacionado x_i são os coeficientes das colunas $x_{i-1} e x_{i+1}$. Por exemplo, o estado x_5 sempre vai estar relacionado com os estados $x_4 e x_6$ devido à condição de Markov. Vemos na figura 33 que para $A'_{modelo}[x_5]$, os coeficientes existentes na submatriz são $C_{x_4}[t_5]$, $C_{x_5}[t_5]$, $C_{x_5}[t_6]$, $C_{x_6}[t_6]$. Esses coeficientes correspondem às equações do modelo onde participa o estado x_5 .

A partir dessas observações podemos concluir que, para determinar $\bar{A}'_{modelo}[x_i]$, o nó líder somente deve simular o modelo dinâmico e formar a matriz $(A_{modelo} | b)$ até a equação de predição do estado x_{i+1} . Se eliminamos, da matriz resultante, as linhas que não têm coeficiente em x_i e as colunas zero, o resultado é a matriz $K_{modelo}[x_i]$ mostrada na figura 34. $K_{modelo}[x_i]$ é uma pequena submatriz de dimensão 2×4 que contém os coeficientes relacionados com x_i e a informação necessária para determinar $\bar{A}'_{modelo}[x_i]$.

Após se obter $K_{modelo}[x_i]$, o nó líder deve ordenar as colunas dessa matriz segundo a ordem de colunas determinada pelo sorvedouro. O resultado da ordenação é a matriz $K'_{modelo}[x_i]$, na qual, a coluna x_i pode estar colocada em 3 posições diferentes. Isso se observa na figura 35. Nessa figura



Figura 33 - Matrizes $A_{modelo}[x_i]$ e $A'_{modelo}[x_i]$ para o cenário estudado

$$K_{modelo}[x_i] = \begin{pmatrix} x_{i-1} & x_i & x_{i+1} & b \\ C_{x_{i-1}}[t_i] & C_{x_i}[t_i] & \mathbf{0} & b_{x_i} \\ \mathbf{0} & C_{x_i}[t_{i+1}] & C_{x_{i+1}}[t_{i+1}] & b_{x_{i+1}} \end{pmatrix}$$

Figura 34 - Submatriz $K_{modelo}[x_i]$

vemos que, das 3 posições possíveis para a coluna x_i , só para 2 delas existe a submatriz $\bar{A}'_{modelo}[x_i]$. A regra é a seguinte: se a coluna x_i recebe o maior índice de coluna em relação as colunas x_{i-1} e x_{i+1} , então $\bar{A}'_{modelo}[x_i]$ não existe.

Como consequência, depois da determinação de $K_{modelo}[x_i]$, o nó líder pode fazer a verificação da existência de $\bar{A}'_{modelo}[x_i]$. Se a verificação for positiva, proceder a calculá-la. Caso contrário continuar com o processo de construção da matriz $\bar{A}'[x_i]$.

Após obtida $\bar{A}'_{modelo}[x_i]$, a matriz $\bar{A}'[x_i]$ é o resultado de adicionar à $\bar{A}'_{modelo}[x_i]$ as linhas correspondentes as observações de x_i ampliadas com sua correspondente coluna b, como se observa na figura 32.

Uma vez obtida a matriz $\bar{A}'[x_i]$, a matriz F_{x_i} pode ser construída utilizando a seguinte fórmula:

$$\begin{split} & \begin{bmatrix} x_{i} & x_{i-l} & x_{i+l} & b \\ C_{x_{i}}[t_{i}] & C_{x_{i-1}}[t_{i}] & 0 & b_{x_{i}} \\ C_{x_{i}}[t_{i+1}] & 0 & C_{x_{i+1}}[t_{i+1}] & b_{x_{i+1}} \\ \end{bmatrix} & = \begin{bmatrix} x_{i-l} & x_{i} & x_{i+l} & b \\ C_{x_{i-1}}[t_{i}] & C_{x_{i}}[t_{i}] & 0 & b_{x_{i}} \\ 0 & C_{x_{i}}[t_{i+1}] & C_{x_{i+1}}[t_{i+1}] & b_{x_{i+1}} \\ \end{bmatrix} & = \begin{bmatrix} x_{i-l} & x_{i+l} & x_{i+l} & b \\ C_{x_{i}}[t_{i+1}] & C_{x_{i+1}}[t_{i+1}] & b_{x_{i+1}} \\ \end{bmatrix} & = \begin{bmatrix} x_{i-l} & x_{i+l} & x_{i} & b \\ C_{x_{i-1}}[t_{i}] & 0 & C_{x_{i}}[t_{i}] & b_{x_{i}} \\ 0 & C_{x_{i+1}}[t_{i+1}] & C_{x_{i}}[t_{i+1}] & b_{x_{i+1}} \\ \end{bmatrix} & \tilde{A}'_{modelo}[x_{i}] \\ & \tilde{A}'_{modelo}[x_{i}] \end{split}$$

Figura 35 - Existência da matriz $A'_{modelo}[x_i]$

$$F_{x_i} = \begin{bmatrix} \leftarrow \bar{A}' [x_i] \rightarrow \\ \leftarrow U_{filho1_{x_i}} \rightarrow \\ \leftarrow U_{filho2_{x_i}} \rightarrow \\ \vdots \end{bmatrix}$$
(77)

5.5.3.2 Fatoração da matriz A'

O processo de fatoração da matriz segue os passos explicados na capítulo 4. O processo inicia pelas folhas. A cada nó líder lhe corresponde formar sua matriz frontal e realizar a decomposição $F_{x_i} = QR_{x_i}$. Do fator R_{x_i} , a primeira linha é enviada ao sorvedouro e a matriz de atualização U_{x_i} é enviada ao nó líder do grupo do vértice pai de x_i na árvore de eliminação. Esse processo continua até chegar ao vértice raiz da árvore, onde acaba a fatoração.

Já que estamos trabalhando com a matriz aumentada $(F_{x_i} \mid b(F_{x_i}))$, a primeira linha de R_{x_i} contém os valores correspondentes à linha x_i do fator R e do vetor $Q^T b$. Aqui devemos lembrar que o fator R é matriz quadrada (existem igual número de linhas e colunas) e, por ser ele o fator de Cholesky, à coluna x_i lhe corresponde a linha x_i .

Dessa forma, após o processo de fatoração, o sorvedouro obtém o sistema $R\hat{x}' = d$ e pode estimar a trajetória do alvo mediante substituição inversa.

O processo completo de fatoração multifrontal QR executado pela rede de sensores para o cenário estudado é mostrado na figura 36. No cenário, os sensores S_1 , S_2 , e S_4 são os nós líderes de grupo que devem realizar a fatoração multifrontal QR seguindo a árvore de eliminação T(A'). A árvore consta de três níveis e portanto a fatoração é feita em 3 fases. A primeira fase inicia nas folhas de T(A') e consiste na eliminação, em paralelo, dos estados correspondentes ao nível 1 da árvore (x_1, x_3, x_5, x_7). A segunda fase é a eliminação dos estados do nível 2 (x_2 e x_6), também eliminados em forma paralela. A última fase é a eliminação do estado do nível 3 da árvore, o nó raíz (x_4), com o qual finaliza a fatoração. Para facilitar o entendimento do processo de fatoração, na figura 36 observa-se também os índices de coluna das matrizes antes e depois de serem ordenadas.



(a) Nível 1 - Eliminação da coluna correspondente ao estado x_1



(b) Nível 1 - Eliminação da coluna correspondente ao estado x_3



(c) Nível 1 - Eliminação da coluna correspondente ao estado x_5



(d) Nível 1 - Eliminação da coluna correspondente ao estado x_7



(e) Nível 2 - Eliminação da coluna correspondente ao estado x_2



(f) Nível 2 - Eliminação da coluna correspondente ao estado x_6



(g) Nível 3 - Eliminação da coluna correspondente ao estado x_4

Figura 36 - Fatoração multifrontal QR da matriz feita pelos sensores da rede no cenário estudado (Continuação)

5.6 SOLUÇÃO DO SISTEMA UTILIZANDO SUBSTITUIÇÃO INVERSA

O resultado da fatoração da matriz A' é a matriz R. A estrutura de R foi determinada pelo sorvedouro na etapa de determinação da árvore de eliminação T(A'). O nó sorvedouro foi preenchendo os elementos da matriz R na medida em que os nós líderes de grupo terminavam de realizar a fatoração multifrontal. A estimação é realizada solucionando o sistema $R\hat{x}' = d$ mediante substituição inversa. O vetor obtido, \hat{x}' , será reordenado na sequência original para obter a trajetória seguida pelo alvo.

Para o caso estudado, na figura 37 observa-se o sistema que o sorvedouro deve solucionar para estimar a trajetória do alvo. Após realizar a substituição inversa, o resultado é o vetor $\hat{x}' = (x_1, x_3, x_5, x_7, x_2, x_6, x_4)$. O sorvedouro deve reordenar o vetor na sequência original para obter a trajetória estimada do alvo.

$$R \hat{x}' = d = Q^{T} b$$

$$\begin{vmatrix} r_{(1,1)} & & r_{(1,5)} & & \\ r_{(2,2)} & & r_{(2,5)} & & r_{(2,7)} \\ & & r_{(3,3)} & & r_{(3,6)} & r_{(3,7)} \\ & & r_{(4,4)} & & r_{(4,6)} \\ & & & r_{(5,5)} & & r_{(5,7)} \\ & & & r_{(6,6)} & r_{(6,7)} \\ & & & & r_{(7,7)} \end{vmatrix} \begin{vmatrix} x_{1} \\ x_{3} \\ x_{5} \\ x_{7} \\ x_{6} \\ x_{6} \\ x_{7} \\ x_{6} \\ x_{6} \\ x_{7} \\$$

Figura 37 - Matriz R^\prime solução da fatoração multifrontal QR da matriz A^\prime

6 SIMULAÇÕES E RESULTADOS

Testamos, mediante diversas simulações, o estimador apresentado neste trabalho. A eficiência e desempenho do algoritmo colaborativo proposto tem sido comparada com a solução centralizada ao problema de estimação de trajetórias com RSSF. Além disso, realizamos simulações para verificar o tempo que a RSSF gastaria para realizar a fatoração multifrontal QR e simulações para verificar a quantidade de memória que ocupariam as matrizes frontais e as matrizes de atualização durante o processo de fatoração. Neste capítulo apresentamos as simulações e os resultados obtidos. As simulações foram realizadas utilizando o software Mathematica em um computador Macintosh com 2 processadores de 4 núcleos cada um a 2.8 GHz, com 16 GB de memória e sistema operacional Mac OS X.

6.1 MODELO DINÂMICO DOS ALVOS

Para realizar as simulações, temos utilzado um modelo dinâmico linear. Esse modelo é fácil de trabalhar e nos permite demonstrar claramente o processo de distribuição do cálculo da estimativa de trajetórias com RSSF.

O modelo utilizado corresponde ao movimento de uma partícula com velocidade aproximadamente constante e é denominado *Discrete White Noise Acceleration* (DWNA) ou *piecewise constant white acceleration*, devido a considerarmos a aceleração constante durante os intervalos de amostragem (BAR-SHALOM; KIRUBARAJAN; LI, 2002). Esse modelo é muito utilizado em problemas de rastreamento de alvos, por ser simples, linear e admitir uma solução fácil para problemas com múltiplos alvos (DURRANT-WHYTE, 2001).

O modelo linear de tempo contínuo para uma partícula com movimento de velocidade aproximadamente constante sobre uma linha (eixo x) é dado pela seguinte equação:

$$\begin{bmatrix} \dot{x}(t) \\ \ddot{x}(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x(t) \\ \dot{x}(t) \end{bmatrix} + \begin{bmatrix} 0 \\ v(t) \end{bmatrix}$$
(78)

onde o vetor de estado para esse sistema consta de duas variáveis, a posição x(t) e a velocidade $\dot{x}(t)$.

A matriz de transição de estado para esse modelo, no intervalo de amostragem ΔT é dada por:

$$F = \begin{bmatrix} 1 & \Delta T \\ 0 & 1 \end{bmatrix}$$
(79)

Por ser uma partícula de velocidade constante, em teoria não existe aceleração. Porém, na prática a velocidade tem pequenas mudanças. Isso é modelado mediante um ruído branco de tempo continuo $\tilde{v}(t)$ com média zero :

$$E[\tilde{v}(t)] = 0 \tag{80}$$
e variância dada por:

$$E[v(t)v(\tau)] = \sigma_v^2(t)\delta(t-\tau)$$
(81)

Com esses pressupostos, o modelo equivalente de tempo discreto do ruído do processo é dado por:

$$\mathbf{v}(k) = \int_0^{\Delta T} \begin{bmatrix} \tau \\ 1 \end{bmatrix} v(k\Delta T + \tau)d\tau = \begin{bmatrix} \frac{\Delta T^2}{2} \\ \Delta T \end{bmatrix} v(k)$$
(82)

Portanto, a equação de estado em tempo discreto é a seguinte:

$$\begin{bmatrix} x(k+1) \\ \dot{x}(k+1) \end{bmatrix} = \begin{bmatrix} 1 & \Delta T \\ 0 & 1 \end{bmatrix} x(k) + \begin{bmatrix} \frac{\Delta T^2}{2} \\ \Delta T \end{bmatrix} v(k)$$
(83)

$$x(k+1) = Fx(k) + \Gamma v(k)$$
(84)

onde o ruído do processo, v(k), corresponde a uma sequência de variáveis aleatórias escalares com média zero e variância σ_v^2 .

A matriz de covariância do ruído do processo é determinada por:

$$Q(k) = E\left[\Gamma v(k)v(k)\Gamma^{T}\right] = \begin{bmatrix} \frac{\Delta T^{4}}{4} & \frac{\Delta T^{3}}{2} \\ \frac{\Delta T^{3}}{2} & \Delta T^{2} \end{bmatrix} \sigma_{v}^{2}$$
(85)

A dimensão física de v(k) é de $[distancia]/[tempo]^2$ o que corresponde a aceleração.

Se observações da posição da partícula são feitas pelos sensores da rede a cada tempo de amostragem, o modelo das observações e sua correspondente matriz de convariância têm as seguintes equações:

$$z(k) = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x(k) \\ \dot{x}(k) \end{bmatrix} + w(k)$$
(86)

$$R(k) = E[w^2(k)] = \sigma_z^2$$
 (87)

O modelo descrito pode ser estendido para duas ou três dimensões. No caso de duas dimensões (plano x-y), as equações seriam as seguintes:

$$\begin{bmatrix} x(k+1) \\ \dot{x}(k+1) \\ y(k+1) \\ \dot{y}(k+1) \end{bmatrix} = \begin{bmatrix} 1 & \Delta T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \Delta T \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x(k) \\ \dot{x}(k) \\ y(k) \\ \dot{y}(k) \end{bmatrix} + \begin{bmatrix} \frac{\Delta T^2}{2} & 0 \\ \Delta T & 0 \\ 0 & \frac{\Delta T^2}{2} \\ 0 & \Delta T \end{bmatrix} \begin{bmatrix} v_x(k) \\ v_y(k) \end{bmatrix}$$
(88)
$$\begin{bmatrix} z_x(k) \\ z_y(k) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x(k) \\ \dot{x}(k) \\ y(k) \\ \dot{y}(k) \end{bmatrix} + \begin{bmatrix} w_x(k) \\ w_y(k) \end{bmatrix}$$
(89)

$$Q(k) = \begin{bmatrix} Q_x & 0 \\ 0 & Q_y \end{bmatrix} = \begin{bmatrix} \frac{\Delta T^4}{4} \sigma_{v_x}^2 & \frac{\Delta T^3}{2} \sigma_{v_x}^2 & 0 & 0 \\ \frac{\Delta T^3}{2} \sigma_{v_x}^2 & \Delta T^2 \sigma_{v_x}^2 & 0 & 0 \\ 0 & 0 & \frac{\Delta T^4}{4} \sigma_{v_y}^2 & \frac{\Delta T^3}{2} \sigma_{v_y}^2 \\ 0 & 0 & \frac{\Delta T^3}{2} \sigma_{v_y}^2 & \Delta T^2 \sigma_{v_y}^2 \end{bmatrix}$$
(90)
$$R(k) = \begin{bmatrix} \sigma_{z_x}^2 & 0 \\ 0 & \sigma_{z_y}^2 \end{bmatrix}$$
(91)

Nas simulações temos utilizado o modelo em duas dimensões (plano x-y) descrito pelas equações 88, 89, 90, 91. Na figura 38 mostra-se em exemplo de trajetória gerada por esse modelo.



Figura 38 - Exemplo de trajetória de alvo gerada pelo modelo dinâmico dos alvos

6.2 MODELO DOS NÓS SENSORIAIS

Na simulação temos como hipótese que todos os nós sensoriais são homogêneos. A cobertura do sensor é modelada mediante um modelo probabilístico para detecção do alvo extraído de (ZOU; CHAKRABARTY, 2004).



(b) Probabilidade de detecção do modelo, para $R1=2\,m,\,R2=0.5\,m$ diferentes valores de λ e β

Figura 39 - Modelo de sensoriamento com cobertura probabilística

A probabilidade de detecção do alvo do sensor s_i é dada pela seguinte equação:

$$P(c_{xy}(s_i)) = \begin{cases} 0, & se \ R1 \le d(s_i, alvo) \\ e^{-\lambda a^{\beta}}, & se \ R2 < d(s_i, alvo) < R1 \\ 1, & se \ R2 \ge d(s_i, alvo) \end{cases}$$
(92)

Nessa equação, $d(s_i, alvo)$ é a distância euclidiana entre o sensor s_i e o alvo. Se o alvo se encontra fora do raio de cobertura do sensor R1, o sensor não consegue detectar o alvo. Se o alvo se encontra a uma distância menor do que R2, o sensor consegue detetá-lo com probabilidade 1. Se o alvo se encontra no intervalo $R2 < d(s_i, alvo) < R1$, a probabilidade diminui exponencialmente com a distância entre R2 e R1, com parâmetros $a = d(s_i, alvo) - R2$, $\lambda \in \beta$. Diferentes valores dos parâmetros $\lambda \in \beta$ geram diferentes comportamentos de detecção, que podem ser vistos como característicos de vários tipos de sensores físicos (ZOU; CHAKRABARTY, 2004).

Na figura 39 pode-se ver o modelo de cobertura do sensor e a probabilidade de detecção do sensor para diferentes valores de λ e β . Nas simulações, os parâmetros utilizados foram $\lambda = 0.5$ e $\beta = 1$

6.3 CONFIGURAÇÃO DA REDE DE SENSORES SEM FIO

A RSSF simulada se encontra espalhada em uma área quadrada de 40 m de lado. Os nós sensoriais encontram-se posicionados uniformemente dentro desse espaço, formando uma grade em que os sensores encontram-se distanciados 2 m entre si. Isso é mostrado na figura 40a.

A topologia da rede depende do alcance de comunicação dos sensores. Nas simulações assumimos que o alcance de comunicação, o qual denominaremos r_c , é suficientemente grande para permitir que, ante qualquer distribuição da posição dos nós, se um alvo encontra-se na intersecção do alcance de sensoriamento de dois ou mais sensores, esses sensores consigam comunicar-se entre si. A relação que garante essa comunicação é $r_c \ge 2R1$. Na figura 40b mostra-se a topologia de comunicação da RSSF no caso em que $r_c = 2R1$.

A forma como os nós sensoriais encontram-se posicionados na figura 40a, só é utilizada para fins da simulação, já que o algoritmo colaborativo proposto é generico e não depende de uma estrutura específica para seu funcionamento.

6.4 SIMULAÇÃO DA ESTIMAÇÃO DE TRAJETÓRIAS COLABORATIVA COM RSSF BASEADO EM FATORAÇÃO MULTIFRONTAL QR

Extensas simulações foram realizadas para testar o algoritmo colaborativo proposto nessa tese. Além disso, a eficiência e precisão do algoritmo proposto foram testadas comparando o algoritmo com a solução centralizada do problema, na qual todas as observações dos sensores são transmitidas ao sorvedouro para realizar a estimação.

6.4.1 Parâmetros das simulações

Os seguintes parâmetros foram utilizados durante as simulações:

- Ambiente: Quadrado de 40 m. (por lado)
- Tempo de simulação: 200 segundos





Figura 40 - RSSF utilizada no simulador

- Tempo de amostragem: 1 segundo
- Número de alvos: 1
- Número de nós sensoriais: 361
- Parâmetros da cobertura dos sensores: R1 = 2 m, R2 = 0.5 m, $\lambda = 0.5$ e $\beta = 1$.
- Ponto inicial da trajetória dos alvos: Posição aleatória próxima ao centro do ambiente

Adicionalmente, nas simulações, é pressuposto que o problema de associação de dados foi resolvido. Em outras palavras, os nós sensoriais conseguem identificar a que alvo pertencem suas observações.

6.4.2 Tabela de observação de estados

A Tabela de Observação de Estados é obtida pelo sorvedouro segundo o explicado na seção 5.4. A figura 41 mostra parte de uma tabela, obtida durante uma das simulações.

sensorId	x 1	x 2	x 3	x 4	x 5	хб	x 7	x 8	x9	x10	x11	x12	x13	x14	x15	x16	x17	x18	x19	x20
181	0	1	0	1	1	1	0	0	1	1	1	1	1	1	0	1	1	1	1	1
199	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
200	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
218	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
219	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
237	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
238	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
239	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
257	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
258	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
259	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
277	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
278	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
279	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
297	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
298	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
299	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
317	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
318	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figura 41 - Tabela de Observação de Estados para os primeiros 20 estados observados pela RSSF

6.4.3 Matriz principal A do problema de estimação de trajetórias

Na figura 42 é mostrada a matriz principal do sistema de equações Ax = b de uma das simulações realizadas. Essa é a matriz que é formada na solução centralizada do problema e que, no algoritmo distribuído, os nós sensoriais fatoram colaborativamente. Nessa matriz, os elementos não nulos (nnz - number of nonzeros) estão representados em cores e os zeros em branco.

Como se observa, essa matriz é extremamente esparsa, com 1.337.530 zeros e 3.270 elementos não nulos.

Na figura 43 se mostra a matriz $A^T A$ do sistema de equações normais. Como pode ser observado, sua estrutura corresponde a uma matriz tridiagonal em blocos.

6.4.4 Ordenação de colunas e árvore de eliminação

A determinação da nova ordem de colunas é realizada utilizando ordenação por dissecção. O programa Mathematica não possui uma função específica para determinar a ordenação, portanto foi escrito um algoritmo para realizar essa ordenação utilizando as funções próprias do Mathematica.



Figura 43 - Matriz A^TA da equação normal correspondente a simulação do problema

Como resultado da ordenação, a nova matriz A^\prime , a qual os nós da RSSF devem fatorar, é mostrada

na figura 44.

Na figura 45 se mostra a árvore de eliminação correspondente à matriz A'. O algoritmo para determinar a árvore de eliminação foi implementado em Mathematica segundo George (1998).



Figura 44 - Matriz principal do problema reordenada



Figura 45 - Árvore de eliminação correspondente à matriz A^\prime

6.4.5 Estimação distribuída da trajetória

A estimação da trajetória do alvo é realizada segundo o algoritmo apresentado no capítulo 5. Os resultados da estimação de trajetória distribuída têm sido comparados com a estimativa realizada de forma centralizada. Além disso, para testar o desempenho do estimador, temos comparado o erro quadrático medio (RMS – *Root Mean Square*) das posições estimadas mediante as soluções distribuída e centralizada, utilizando o método de Monte Carlo para 200 simulações.

6.4.5.1 Estimação de trajetórias

O simulador, utilizando como guia a árvore de eliminação, realiza a fatoração multifrontal QR formando cada uma das matrizes frontais associadas aos nós da árvore. Uma vez obtido o fator de Cholesky R, a estimativa é obtida mediante substituição inversa. A fatoração das matrizes frontais é realizada utilizando a função QRDecomposition do Mathematica. Essa função utiliza transformações de Householder para realizar a decomposição ortogonal.

A solução centralizada do problema de estimação de trajetórias foi realizada formando a matriz Ae o vetor b correspondente a todo o problema. A solução é obtida a partir do sistema de equações Ax = b sem reordenar as colunas da matriz A (já que não é necessário) e utilizando fatoração QR e substituição inversa. A funções do Mathematica utilizadas foram *QRDecomposition* para a fatoração QR e *LinearSolve* utilizando R e $Q^{-1}b$ para realizar a substituição inversa.

A figura 46 mostra a trajetória real do alvo. Nessa figura se observam também os nós sensorias (pontos pretos) e seu alcance de sensoriamento.

As estimações obtidas com nosso algoritmo colaborativo e com a solução centralizada são mostradas nas figuras 47 e 48, respectivamente.

Na figura 49 mostram-se as trajetórias real, estimada colaborativamente e estimada de forma centralizada na mesma figura. Observa-se nessa figura que não é possivel diferenciar as trajetórias geradas pela solução colaborativa e a solução centralizada. A razão disso é que as estimativas obtidas nas duas soluções são idênticas.

Na tabela 2 mostra-se as posições estimadas com a solução colaborativa e a solução centralizada para os primeiros e últimos 10 estados. Os resultados são idênticos porque, utilizando a fatoração multifrontal QR, estamos solucionando o mesmo problema, sem aproximações ou minimizações, que o estimador centralizado, porém de forma distribuída.



Figura 46 - Trajetória real e observações dos nós sensoriais



(b) Detalhe das trajetórias real e estimada junto com as observações feitas pelos sensores

Figura 47 - Trajetória estimada mediante solução colaborativa



(b) Detalhe das trajetórias real e estimada junto com as observações feitas pelos sensores

Figura 48 - Trajetoria estimada mediante a solução centralizada



Figura 49 - Trajetórias real, colaborativa e centralizada. Na figura vemos que as trajetórias estimadas centralizada e colaborativa encontram-se sobrepostas e são indistinguíveis devido ao fato de que o resultado da estimativa para ambas as soluçoes foram idênticas, como é mostrado na tabela 2

	Solução	Colaborativa	Solução Centralizada			
Estado	Pos x	Pos y	Pos x	Pos y		
1	20,336	21,690	20,336	21,690		
2	20,317	21,686	20,317	21,686		
3	20,307	21,667	20,307	21,667		
4	20,303	21,668	20,303	21,668		
5	20,274	21,675	20,274	21,675		
6	20,204	21,690	20,204	21,690		
7	20,132	21,708	20,132	21,708		
8	20,077	21,721	20,077	21,721		
9	20,035	21,732	20,035	21,732		
10	20,001	21,747	20,001	21,747		
:	:		:	:		
191	14,212	26,968	14,2115	26,968		
192	14,128	27,007	14,1281	27,007		
193	14,063	27,039	14,0627	27,039		
194	14,000	27,058	13,9995	27,058		
195	13,932	27,059	13,9315	27,059		
196	13,858	27,060	13,8582	27,060		
197	13,800	27,047	13,7998	27,047		
198	13,735	27,043	13,735	27,043		
199	13,669	27,064	13,6691	27,064		
200	13,621	27,106	13,6205	27,106		

Tabela 2 - Comparação dos estados estimados com a solução colaborativa e centralizada

6.4.5.2 Erro quadrático médio

Para verificar a eficiência do estimador, temos calculado e comparado o erro RMS das posições estimadas do alvo mediante ambas as soluções para 200 simulações Monte Carlo.

A equação para calcular o erro RMS para N simulações Monte Carlo é a seguinte:

$$RMS(pos) = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (\tilde{x}_i^2 + \tilde{y}_i^2)}$$
(93)

onde $\tilde{x}_i = (x_{i \, real} - x_{i \, estimada})$ e $\tilde{y}_i = (y_{i \, real} - y_{i \, estimada})$. Na figura 50 se observam os resultados para a solução colaborativa e centralizada.



Figura 50 - Erro RMS (a) Solução colaborativa; (b) Solução centralizada

Nessa figura observa-se que o erro RMS de ambas as soluções encontram-se na mesma faixa, entre $0,025 \, \text{e} \, 0,041$. O valor médio para o erro RMS para ambas as soluções são idênticos, sendo o da solução colaborativa de 0,02805 e o da solução centralizada 0,02783. Esse resultado corrobora que, como visto nas provas de estimação da trajetória, os dois estimadores são equivalentes.

6.4.6 Tempo na fatoração

Além dos testes de eficiência do estimador, temos realizado testes para estimar o tempo que demorariam os nós da RSSF para calcular a fatoração multifrontal QR.

Para a solução colaborativa, a figura 51 ilustra como temos calculado o tempo que demora a RSSF para realizar a fatoração.



T Fatoração = t_{max} nível 1 + t_{max} nível 2 + ... + t_{max} nível 8

Figura 51 - Processo de cálculo do tempo de fatoração da matriz principal do problema

Desconsiderando o tempo de transmissão das matrizes de atualização, podemos considerar que o tempo de fatoração da matriz é o tempo que demora a RSSF em formar e fatorar as matrizes frontais até chegar na raiz da árvore de eliminação. Como a fatoração realizada pela RSSF é calculada em paralelo, para calcular o tempo da fatoração, determinamos, para cada nível da árvore, o tempo gasto pelos nós em formar e fatorar as matrizes frontais (a árvore consta de 8 níveis para 200 estados observados, vide figura 45). Logo, o tempo total da fatoração corresponde à somatória dos tempos máximos obtidos em cada nível.

Para a solução centralizada determinou-se o tempo que demora o simulador em realizar a fatoração QR da matriz principal A. Os tempos são calculados utilizando a função *Timing* do Mathematica, a qual calcula o tempo utilizado pela CPU para obter os resultados.

O teste realizado executa 50 simulações Monte Carlo para a solução colaborativa e centralizada e em cada uma delas determina o tempo gasto na fatoração. Finalmente, o tempo de fatoração corresponde à media dos tempos das 50 simulações. Devido a que as matrizes frontais crescem em tamanho com o número de observações por estado, temos realizado o teste para diferentes valores do alcance de sensoriamento dos nós sensoriais, o que aumenta o número de observações por estado . Os resultados mostram-se na tabela 3.

	lempo de fatoração (ms)					
Alcance (m)	Colaborativa	Centralizada				
2	7,221	845,641				
5	9,589	1.485,960				
10	10,790	1.762,760				
15	11,236	1.928,310				

Tabela 3 - Comparação do tempo de fatoração das soluções colaborativa e centralizada

Observa-se nos resultados que executar a fatoração de forma colaborativa diminui consideravelmente o tempo de cálculo da estimativa da trajetória do alvo.

6.4.7 Mémoria utilizada

Foram realizados testes para verificar a quantidade de memória necessária para armazenar as matrizes frontais e as matrizes de atualização durante a fatoração multifrontal. Os testes consitiram em realizar 50 simulações Monte Carlo da estimação da trajetória de um alvo e, para cada uma das simulações, determinar a memória máxima e mínima necessaria para alocar as matrizes frontais e de atualização.

Para determinar a memória utilizada, calculamos o número de elementos das matrizes frontais e de atualização para todos os nós da árvore de eliminação. Aqui consideramos que cada elemento das matrizes normalmente é representado, no mircrocontrolador dos nós sensoriais, como um número de ponto flutuante (*floating point*) de 32 bits (4 *bytes*) (REICHENBACH et al., 2006), portanto a memória ocupada por cada matriz será calculada multiplicando o número de elementos da matriz por 32 bits.

Da mesma forma que para o teste de tempo de fatoração, o teste de memória foi realizado para diferentes alcances de sensoriamento dos sensores, o que incrementa o tamanho das matrizes frontais. Os resultados são mostrados na tabela 4, onde apresentamos o tamanho máximo e mínimo de memória para as matrizes frontais e de atualização.

Vemos nos resultados que as matrizes frontais ocupam maior espaço de memória que as matrizes de atualização. Porém os valores de memória se mantêm pequenos apesar de aumentar o número de observações por estado dos sensores da rede. Além disso, considerando que, durante o processo de formação e fatoração das matrizes frontais, um nó sensorial pode ter, no mesmo instante de tempo, as matrizes de atualização enviadas por seus filhos e a matriz frontal final, a quantidade de memória

	Memória Ma	atriz Frontal	Memória Matriz Atualização			
Alcance (m)	Máxima (KiB)	Mínima (KiB)	Máxima (KiB)	Mínima (KiB)		
2	1,625	0,210	0,422	0,078		
5	2,437	0,352	0,422	0,156		
10	3,047	0,211	0,422	0,078		
15	2,945	0,352	0,422	0,156		

Tabela 4 - Quantidade de memória máxima e mínima para as matrizes frontais e de atualização

máxima que ocupariam essas matrizes seria de 3,469 KiB (vemos na árvore de eliminação da figura 45 que cada nó tem como máximo 2 nós filhos) o que se encontra dentro dos tamanhos de memória de dados encontrados nos nós sensoriais para RSSF existentes no mercado (HEALY; NEWE; LEWIS, 2008).

7 CONCLUSÕES

Esta tese tratou do problema de estimação de trajetórias com RSSF. Esse problema normalmente é abordado como um problema de rastreamento de alvos, onde a RSSF estima sequencialmente a posição do alvo. O problema de rastreamento de alvos com RSSF é considerado um dos mais importantes na área, já que se considera que essa funcionalidade é essencial da RSSF (MA, 2008; YICK; MUKHERJEE; GHOSAL, 2008; ZHAO; GUIBAS, 2004).

Por outro lado, a solução proposta e validada nesta tese aborda o problema de forma diferente. A solução utiliza o método de estimação por lotes, onde coletam-se as observações do alvo durante toda a aplicação antes de realizar a estimação. Esta forma de realizar a estimação tem sido pouco explorada na área de RSSF, mas já tem sido muito utilizada em áreas como fotogrametria, visão computacional e robótica.

Na solução utiliza-se o método de estimação por lotes para modelar o problema como um sistema de equações sobredeterminado do tipo Ax = b, o qual é solucionado utilizando mínimos quadrados ponderados. A base do método de mínimos quadrados é a fatoração da matriz A e, devido ao fato de A ser esparsa, ela pode ser fatorada utilizando o método de fatoração multifrontal QR. Utilizar o método de fatoração multifrontal QR permite realizar a fatoração de forma colaborativa e distribuída já que o trabalho da fatoração da matriz A é distribuído entre os nós sensoriais da RSSF. Até onde sabemos, esta é a primeira vez que se apresenta uma solução desse tipo na área de RSSF, já que os trabalhos revisados na literatura mostram que existe preferência para abordar esse problema utilizando estimação sequencial.

Diversas simulações foram realizadas para validar a solução proposta. Essas simulações produziram bons resultados que validam a efetividade da solução proposta.

A conclusão mais importante refere-se à distribuição exata, na RSSF, do estimador centralizado . Esse resultado é consequência da característica esparsa da matriz principal do problema e à efetiva divisão do trabalho de fatoração conseguida com o método multifrontal QR.

Além disso, se comprovou que o tempo de fatoração gasto pelo método distribuído proposto pode chegar a ser muito menor do que o tempo gasto pelo método centralizado. Em todos os testes realizados, sem considerar o tempo de comunicação entre os nós sensoriais da RSSF, a solução distribuída consegue fatorar a matriz principal A em um tempo 99% menor do que a solução centralizada.

Finalmente, em relação à memoria de dados que os nós sensoriais utilizariam com a solução proposta, vimos que as matrizes frontais e de atualização se mantêm pequenas apesar do aumento das observações sendo que, como máximo, o nó sensorial requer 4 KiB de memória para a solução poder ser implementada.

Mesmo assim, uma implementação em uma RSSF real poderá resultar em um análise mais detalhada do algoritmo e em melhoras do algoritmo apresentado.

Como trabalho futuro, a solução proposta pode ser estendida para realizar a estimação em forma incremental. Na solução incremental a estimação por lotes seria realizada em intervalos de tempo

predeterminados. Para isso, a cada intervalo de tempo é necessário determinar uma ordem de colunas que dê como resultado uma árvore de eliminação no qual o nó raiz da árvore de eliminação do intervalo de tempo anterior esteja no nível das folhas da nova árvore de eliminação. Dessa forma, o nó raiz da árvore de eliminação do intervalo de tempo anterior teria toda a informação necessária para formar a matriz frontal da nova árvore sem necessidade de repetir a fatoração anterior. Testes preliminares apontam que é possível determinar essa ordem de colunas.

REFERÊNCIAS

AGRE, J.; CLARE, L. An integrated architecture for cooperative sensing networks. **Computer**, v. 33, n. 5, p. 106 – 108, May 2000.

AKKAYA, K.; YOUNIS, M. A survey on routing protocols for wireless sensor networks. Ad Hoc Networks, v. 3, n. 3, p. 325 – 349, 2005.

AL-KARAKI, J.; KAMAL, A. Routing techniques in wireless sensor networks: a survey. Wireless Communications, IEEE, v. 11, n. 6, p. 6 – 28, Dec. 2004.

AMESTOY, P.; DUFF, I.; PUGLISI, C. Multifrontal qr factorization in a multiprocessor environment. Int. Journal of Num. Linear Alg. and Appl., v. 3, p. 275–300, 1996.

BAR-SHALOM, Y.; KIRUBARAJAN, T.; LI, X.-R. Estimation with Applications to Tracking and Navigation. New York, NY, USA: John Wiley & Sons, Inc., 2002.

BAR-SHALOM, Y.; LI, X.-R. Multitarget-Multisensor Tracking: Principles and Techniques. [S.I.]: Artech House, 1995.

BATHULA, M. et al. A sensor network system for measuring traffic in short-term construction work zones. DCOSS '09: Proceedings of the 5th IEEE International Conference on Distributed Computing in Sensor Systems. p. 216–230, 2009.

BJORCK, A. Numerical methods for least squares problems. Philadelphia: SIAM, 1996.

BOKAREVA, T. et al. Wireless sensor networks for battlefield surveillance. Brisbane, Queensland, Australia, 2006.

BOULIS, A.; GANERIWAL, S.; SRIVASTAVA, M. Aggregation in sensor networks: an energyaccuracy trade-off. Sensor Network Protocols and Applications, 2003. Proceedings of the First IEEE. 2003 IEEE International Workshop on. May 2003.

BROWN, D. C. The bundle adjustment - progress and prospects. Int. Archives Photogrammetry, v. 21, n. 3, 1976.

CHEESEMAN, P.; SMITH, P. On the representation and estimation of spatial uncertainty. International Journal of Robotics, v. 5, p. 56–68, 1986.

CHONG, C.-Y.; CHANG, K.-C.; MORI, S. Distributed tracking in distributed sensor networks. **American Control Conference**, **1986**. p. 1863–1868, Jun. 1986.

CHONG, C.-Y.; KUMAR, S. P. Sensor networks: evolution, opportunities, and challenges. **Proceedings of the IEEE**, v. 91, n. 8, p. 1247 – 1256, Aug. 2003.

CULLER, D.; ESTRIN, D.; SRIVASTAVA, M. Guest editors' introduction: overview of sensor networks. **Computer**, v. 37, n. 8, p. 41 – 49, Aug. 2004.

DASARATHY, B. More the merrier...or is it? sensor suite augmentation benefits assessment. Information Fusion, 2000. FUSION 2000. Proceedings of the Third International Conference on. v. 2, July 2000.

DELLAERT, F.; KAESS, M. Square root sam: simultaneous localization and mapping via square root information smoothing. **International Journal of Robotics Research**, Thousand Oaks, CA, USA, Sage Publications, Inc., v. 25, n. 12, p. 1181–1203, Dec 2006. DOI: http://dx.doi.org/10.1177/0278364906072768.

DELLAERT, F.; KIPP, A.; KRAUTHAUSEN, P. A multifrontal qr factorization approach to distributed inference applied to multi-robot localization and mapping. **in Proceedings of the American Association for Artificial Intelligence 9-13 July**. p. 1261–1266, 2005.

DEMMEL, J. W.; HEATH, M. T.; VORST, H. A. van der. Parallel Numerical Linear Algebra. [S.l.: s.n.], 1992.

DENNIS, J. E.; SCHNABEL, R. B. Numerical Methods for Unconstrained Optimization and Nonlinear Equations. [S.I.]: SIAM, 1996.

DUFF, I. S.; REID, J. K. The multifrontal solution of indefinite sparse symmetric linear equations. **ACM Trans. Math. Softw.**, New York, NY, USA, ACM, v. 9, n. 3, p. 302–325, 1983. DOI: http://doi.acm.org/10.1145/356044.356047.

DUFF, I. S.; VORST, H. A. van der. Developments and trends in the parallel solution of linear systems. **Parallel Computing**, v. 25, n. 13-14, p. 1931–1970, 1999.

DURRANT-WHYTE, H. A Beginners Guide to Decentralised Data Fusion. The University of Sydney NSW 2006 Australia: [s.n.], July 2000.

DURRANT-WHYTE, H. **Multi Sensor Data Fusion**. [S.I.: s.n.], jan. 2001. Course Notes for the KC-3 Multisensor Data Fusion - Australian Center For Field Robotics - The University of Sydney.

DURRANT-WHYTE, H.; HENDERSON, T. C. Multisensor data fusion. In: Handbook of Robotics. Handbook of Robotics. [S.I.]: Springer, 2007. p. 585 – 610.

ESTRIN, D. et al. Instrumenting the world with wireless sensor networks. **Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)**. p. 2033 – 2036, 2001.

FAUGERAS, O. Three-Dimensionl Computer Vision: A geomatric Viewpoint. [S.I.]: The MIT Press, 1993.

FENG, M.-W. et al. Wireless sensor network and sensor fusion technology for ubiquitous smart living space applications (invited paper). **Proc. Second International Symposium on Universal Communication ISUC '08**. p. 295–302, 2008. DOI: 10.1109/ISUC.2008.87.

FISHER, R. A. On an absolute criterion for fitting frequency curves. Messenger of Mathematics, v. 41, p. 155–160, 1912.

FOX, D. et al. Bayesian filtering for location estimation. **Pervasive Computing, IEEE**, v. 2, n. 3, p. 24 – 33, July-Sept. 2003.

GEORGE, A. Nested dissection of a regular finite element mesh. **SIAM Journal of Nume**rical Analisys, v. 10, n. 2, p. 345–363, 1973.

GEORGE, A. On finding and analizing the structure of the cholesky factor. In: Algorithms for Large Scale Linear Algebraic Systems: Applications in Science and Engineering. Algorithms for Large Scale Linear Algebraic Systems: Applications in Science and Engineering. [S.I.]: Kluwer Academic Publishers, 1998.

GEORGE, A.; HEATH, M. T. Solution of sparse linear least squares problems using givens rotations. Linear Algebra and its Applications, v. 34, p. 69–83, 1980.

GEORGE, A.; LIU, J. W. Householder reflections versus givens rotations in sparse orthogonal decomposition. Linear Algebra and its Applications, v. 88-89, p. 223–238, 1987.

GHOSH, A.; DAS, S. K. Coverage and connectivity issues in wireless sensor networks. In: Mobile, Wireless and Sensor Networks: Technology, Applications and Future Directions. Mobile, Wireless and Sensor Networks: Technology, Applications and Future Directions. [S.I.]: John Wiley & Sons, 2006. cap. 9, p. 221–256.

GILBERT, J. R.; MOLER, C.; SCHREIBER, R. Sparse matrices in matlab: design and implementation. **SIAM Journal on Matrix Analysis and Applications**, 1992.

GOLUB, G. H.; LOAN, C. F. V. Matrix Computations. 3. ed. [S.I.]: The Johns Hopkins University Press, 1996.

GORDON, N.; SALMOND, D. Aspects of target tracking: problems and techniques. Proc. IEE Colloquium on Target Tracking and Data Fusion (Digest No. 1998/282). p. 1–6, 1998.

GRANSHAW, S. I. Bundle adjustment methods in engineering photogrammetry. **The Pho-togrammetric Record**, Blackwell Publishing Ltd, v. 10, n. 56, p. 181–207, 1980.

GUPTA, A.; GUI, C.; MOHAPATRA, P. Moblie target tracking using sensor networks. In: Mobile, Wireless and Sensor Networks: Technology, Applications and Future Directions. Mobile, Wireless and Sensor Networks: Technology, Applications and Future Directions. [S.I.]: Wiley-Interscience, 2005. HALL, D. L. Estimation and kalman filters. In: **Distributed Sensor Networks**. **Distributed Sensor Networks**. [S.I.]: Chapman and Hall/CRC, 2004.

HANSON, M. A. et al. Body area sensor networks: challenges and opportunities. **Computer**, v. 42, n. 1, p. 58–65, 2009.

HARTLEY, R.; ZISSERMAN, A. Multiple View Geometry in Computer Vision. [S.I.]: Cambridge University Press, 2000.

HEALY, M.; NEWE, T.; LEWIS, E. Wireless sensor node hardware: a review. **Sensors, 2008 IEEE**. p. 621 – 624, 2008.

HEATH, M. T.; NG, E.; PEYTON, B. W. Parallel algorithms for sparse linear systems. **SIAM Review**, v. 33, n. 3, p. pp. 420–460, Sep. 1991.

JULIER, S.; UHLMANN, J. A new extension of the kalman filter to nonlinear systems. Int. Symp. Aerospace/Defense Sensing, Simul. and Controls. 1997.

JULIER, S.; UHLMANN, J. K. General decentralized data fusion with covariance intersection (ci). In: Handbook of Multisensor Data Fusion. Handbook of Multisensor Data Fusion. [S.I.]: CRC Press, 2001. cap. 12, p. 12–1 – 12–15.

JUNYENT, F. et al. The casa integrated project 1 networked radar system. Journal of Atmospheric and Oceanic Technology, p. 61–78, 2010.

KAESS, M.; RANGANATHAN, A.; DELLAERT, F. Isam: fast incremental smoothing and mapping with efficient data association. **Proc. IEEE International Conference on Ro-botics and Automation**. p. 1670–1677, April 2007. DOI: 10.1109/ROBOT.2007.363563.

KALMAN, R. E. A new approach to linear filtering and prediction problems. **Transactions** of the ASME–Journal of Basic Engineering, v. 82, n. Series D, p. 35 – 45, 1960.

KHAN, U. A.; MOURA, J. M. F. Distributing the kalman filter for large-scale systems. **IEEE Journal of Signal Processing**, v. 56, n. 10, p. 4919–4935, out. 2008. DOI: 10.1109/TSP.2008.927480.

KOLMOGOROV, A. N. Interpolation and extrapolation of stationary random sequences. In: Selected Works of A. N. Kolmogorov. Selected Works of A. N. Kolmogorov. [S.I.]: Springer-Verlag, 1992. (Mathematics and its Applications, II).

KUMARAWADU, P. et al. Algorithms for node clustering in wireless sensor networks: a survey. Information and Automation for Sustainability, 2008. ICIAFS 2008. 4th International Conference on. p. 295 – 300, Dec. 2008.

LEE, D.-J. Nonlinear estimation and multiple sensor fusion using unscented information filtering. **Signal Processing Letters, IEEE**, v. 15, p. 861 – 864, 2008. LEUSCHNER, C. J. The design of a simple energy efficient routing protocol to improve wireless sensor network lifetime. 2005. Dissertação (Mestrado) — University of Pretoria.

LI, J.; ZHOU, Y. Target tracking in wireless sensor networks. In: Wireless Sensor Networks: Application-Centric Design. Wireless Sensor Networks: Application-Centric Design. [S.I.]: InTech, Dec. 2010.

LI, X. R.; JILKOV, V. p. Survey of maneuvering target tracking. part i: dynamic models. **IEEE Trans. on Aerospace and Electronic Systems**, v. 39, n. 4, p. 1333–1363, October 2003.

LIU, J. W. H. Computational models and task scheduling for parallel sparse cholesky factorization. **Parallel Computing**, v. 3, n. 4, p. 327–342, 1986.

LIU, J. W. H. On general row merging schemes for sparse givens transformations. **SIAM** Journal on Scientific and Statistical Computing, v. 7, n. 4, p. 1190–1211, Oct. 1986.

LIU, J. W. H. The role of elimination trees in sparse factorization. **SIAM Journal on Matrix Analysis and Applications**, v. 11, n. 1, p. 134–172, 1990.

LIU, J. W. H. The multifrontal method for sparse matrix solution: theory and practice. **SIAM Rev.**, Philadelphia, PA, USA, Society for Industrial and Applied Mathematics, v. 34, n. 1, p. 82–109, 1992. DOI: http://dx.doi.org/10.1137/1034004.

MA, H. Collaborative information processing techniques for target tracking in wireless sensor networks. 2008. Tese (Doutorado) — University of Adelaide, School of Electrical and Electronic Engineering.

MARGI, C. B. et al. Segurança em redes de sensores sem . In: Minicursos: SBSEG 2009 / IX Simpósio Brasileiro de Segurança da Informação e de Sistemas Computacionais. Minicursos: SBSEG 2009 / IX Simpósio Brasileiro de Segurança da Informação e de Sistemas Computacionais. 1. ed. [S.I.]: Sociedade Brasileira da Computação, 2009. cap. CH4.

MARTINCIC, F.; SCHWIEBERT, L. Introduction to wireless sensor networking. In: Handbook of Sensor Networks : Algorithms and Architectures. Handbook of Sensor Networks : Algorithms and Architectures. [S.I.]: John Wiley & Sons, Inc., 2005. cap. 1.

MATSTOMS, P. Sparse qr factorization in matlab. **ACM Trans. Math. Softw.**, New York, NY, USA, ACM, v. 20, n. 1, p. 136–159, 1994. DOI: http://doi.acm.org/10.1145/174603.174408.

MITCHELL, H. B. Multi-Sensor Data Fusion: An Introduction. [S.I.]: Springer, 2007.

MOUTARLIER, P.; CHATILA, R. An experimental system for incremental environment modelling by an autonomous mobile robot. The First International Symposium on Experimental Robotics I. p. 327–346, 1990.

NAKAMURA, E. F.; LOUREIRO, A. A. F.; FRERY, A. C. Information fusion for wireless sensor networks: methods, models, and classifications. **ACM Computing Surveys**, v. 39, n. 3, 2007.

NIYATO, D. et al. Wireless sensor networks with energy harvesting technologies: a gametheoretic approach to optimal energy management. Wireless Communications, IEEE, v. 14, n. 4, p. 90–96, Aug 2007.

OLFATI-SABER, R. Distributed kalman filtering for sensor networks. **Decision and Control**, **2007 46th IEEE Conference on**. p. 5492–5498, Dec 2007.

OLFATI-SABER, R.; SANDELL, N. Distributed tracking in sensor networks with limited sensing range. **American Control Conference**, **2008**. p. 3157–3162, June 2008.

PAO, L. Y. A measurement reconstruction approach for distributed multisensor fusion. Journal of Guidance, Control, and Dynamics, v. 19, n. 4, p. 842–847, 1996.

PARTER, S. The use of linear graphs in gauss elimination. **SIAM Review**, v. 3, n. 2, p. 119–130, April 1961.

PEARL, J. Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. [S.I.]: Morgan Kaufmann Publishers Inc., 1988.

POMPILI, D.; MELODIA, T.; AKYILDIZ, I. F. Deployment analysis in underwater acoustic wireless sensor networks. WUWNet '06: Proceedings of the 1st ACM international workshop on Underwater networks. p. 48–55, 2006. DOI: http://doi.acm.org/10.1145/1161039.1161050.

POTHEN, A.; SUN, C. Distributed multifrontal factorization using clique trees. **Proceedings** of the Fifth SIAM Conference on Parallel Processing for Scientific Computing. p. 34–40, 1992.

RAMESH, K.; SOMASUNDARAM, D. K. A comparative study of clusterhead selection algorithms in wireless sensor networks. International Journal of Computer Science and Engineering Survey (IJCSES), v. 2, n. 4, 2011.

RAO, H. D.-W.; SHEEN, J. A fully decentralized multi-sensor system for tracking and surveillance. The International Journal of Robotics Research, v. 12, n. 1, p. 20 –44, 1993.

REICHENBACH, F. et al. A distributed linear least squares method for precise localization with low complexity in wireless sensor networks. Proceedings of the Second International Conference on Distributed Computing in Sensor Systems, DCOSS 2006. 2006.

SHENG, X.; HU, Y.-H. Maximum likelihood multiple-source localization using acoustic energy measurements with wireless sensor networks. **Signal Processing, IEEE Transactions on**, v. 53, n. 1, p. 44–53, Jan 2005.

SHENG, X.; HU, Y.-H.; RAMANATHAN, P. Distributed particle filter with gmm approximation for multiple targets localization and tracking in wireless sensor network. Information Processing in Sensor Networks, 2005. IPSN 2005. Fourth International Symposium on. p. 181–188, April 2005.

SHI, L.; TAN, J.; ZHAO, Z. Target tracking in sensor networks using statistical graphical models. **ROBIO '09: Proceedings of the 2008 IEEE International Conference on Robotics and Biomimetics**. p. 2050–2055, 2009. DOI: http://dx.doi.org/10.1109/ROBIO.2009.4913317.

SITTLER, R. W. An optimal data association problem in surveillance theory. **Military Electronics, IEEE Transactions on**, v. 8, n. 2, p. 125 – 139, April 1964.

SMITH, D.; SINGH, S. Approaches to multisensor data fusion in target tracking: a survey. **IEEE Transactions on knowledge and data engineering**, v. 18, p. 1696–1710, 2006.

SOHRABY DANIEL MINOLI, T. Z. Wireless Sensor Networks: Technology, protocols and applications. [S.I.]: Wiley, 2007.

SORENSON, H. W. Least-squares estimation: from gauss to kalman. **IEEE Spectrum**, v. 7, p. 63–68, July 1970.

STRANG, G. Introduction to Applied Mathematics. [S.I.]: Wellesley-Cambridge, 1986.

SWERLING, P. A proposed stagewise differential correction procedure for satellite tracking and prediction. [S.l.: s.n.], 1958.

SZELISKI, R.; KANG, S. B. Recovering 3d shape and motion from image streams using nonlinear least squares. Journal of Visual Comunication and Image Representation, v. 5, n. 1, 1994.

SZEWCZYK, R. et al. An analysis of a large scale habitat monitoring application. In Proceedings of the Second ACM Conference on Embedded Networked Sensor Systems (SenSys. p. 214–226, 2004.

THRUN, S.; BURGARD, W.; FOX, D. **Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)**. [S.I.]: The MIT Press, 2005.

TINNEY, W.; WALKER, J. Direct solutions of sparse network equations by optimally ordered triangular factorization. **Proceedings of the IEEE**, v. 55, n. 11, p. 1801–1809, nov. 1967.

TUBAISHAT, M.; MADRIA, S. Sensor networks: an overview. v. 22, n. 2, p. 20-23, 2003.

UMAR, A. Emerging wireless networks: uwb, fso, manet, and flash ofdm. In: MOBILE Computing and Wireless Communications. [S.I.]: NGE Solutions, 2004. cap. 10.

WANG, A.; CHANDRAKASAN, A. Energy-efficient dsps for wireless sensor networks. **Signal Processing Magazine**, **IEEE**, v. 19, n. 4, p. 68 – 78, Jul 2002.

WENTZLOFF, D. et al. Design considerations for next generation wireless power-aware microsensor nodes. VLSI Design, 2004. Proceedings. 17th International Conference on. p. 361 – 367, 2004.

WIENER, N. Extrapolation, Interpolation and Smoothing of Stationary Time Series. [S.I.]: The MIT Press, 1964.

YAO, Y.; GEHRKE, J. Query processing in sensor networks. Proc. 1st Biennal Conf. Innovative Data Systems Research (CIDR 2003). 2003.

YICK, J.; MUKHERJEE, B.; GHOSAL, D. Wireless sensor network survey. **Computer Networks**, v. 52, n. 12, p. 2292 – 2330, 2008. DOI: DOI: 10.1016/j.comnet.2008.04.002.

ZHAO, F.; GUIBAS, L. Wireless Sensor Networks: An Information Processing Approach. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2004.

ZHAO, F.; SHIN, J.; REICH, J. Information-driven dynamic sensor collaboration. **Signal Processing Magazine, IEEE**, v. 19, n. 2, p. 61–72, Mar 2002.

ZHAO, T.; NEHORAI, A. Information-driven distributed maximum likelihood estimation based on gauss-newton method in wireless sensor networks. **Signal Processing, IEEE Transactions on**, v. 55, n. 9, p. 4669–4682, Sept. 2007.

ZHUANG, L. Q.; GOH, K. M.; ZHANG, J. B. The wireless sensor networks for factory automation: issues and challenges. **Proc. ETFA Emerging Technologies and Factory Automation IEEE Conference on**. p. 141–148, 2007. DOI: 10.1109/EFTA.2007.4416764.

ZOU, Y.; CHAKRABARTY, K. Sensor deployment and target localization in distributed sensor networks. **ACM Transaction on Embedded Computing Systems**, v. 3, n. 1, p. 61 – 91, Feb 2004.