

010

CONSULTA
FD-3749
Ed.rev.

SÃO PAULO
2004

Dissertação apresentada à Escola
Politécnica da Universidade de São Paulo
para a obtenção do Título de Mestre em
Engenharia.

**APLICAÇÃO DO FILTRO DE KALMAN E DOS FILTROS DE
PARTÍCULAS À ESTIMAÇÃO DE TRAJETÓRIAS EM
NAVEGAÇÃO INERCIAL**

VICTOR AUGUSTO FERNANDES DE CAMPOS

VICTOR AUGUSTO FERNANDES DE CAMPOS

**APLICAÇÃO DO FILTRO DE KALMAN E DOS FILTROS DE
PARTÍCULAS À ESTIMAÇÃO DE TRAJETÓRIAS EM
NAVEGAÇÃO INERCIAL**

Dissertação apresentada à Escola
Politécnica da Universidade de São Paulo
para a obtenção do Título de Mestre em
Engenharia.

Área de Concentração:
Engenharia Mecatrônica

Orientador:

Prof. Dr. Newton Maruyama

SÃO PAULO

2004

“Eis que eu venho em breve, e minha retribuição está comigo, para pagar a cada um segundo as suas obras. Eu sou o Alfa e o Ômega, o Primeiro e o Último, o começo e o fim.”

Apocalipse 22, 12-13

Aos queridos Odival, Priscila, Renata e Solange, as
pessoas que são a razão da minha felicidade e que
me fortalecem nos momentos difíceis.

AGRADECIMENTOS

Ao meu pai, Odival, pelas valiosas orientações e ensinamentos dados durante a confecção desta dissertação.

Aos colegas do grupo de pesquisa, Douglas Santana, Prof. Newton Maruyama, Prof. Celso Furukawa, Prof. Júlio Adamowski, Paulo Peres e Luis Barbañi pelo auxílio em várias questões relacionadas ao trabalho em equipe. Ao Prof. Flávio Buiochi, pela revisão minuciosa do Capítulo 3. Aos Profs. Oswaldo Costa e Fábio Cozman pelas orientações dadas durante o exame de qualificação. Ao Prof. Marcelo Bruno pelas sugestões dadas durante a defesa desta dissertação.

Agradeço igualmente à Petrobrás na pessoa do Sr. Cláudio Camerini pelo fornecimento dos dados e fotos relativos ao teste do pig inercial, e ao Sr. Vinicius Lima pela aquisição de dados no teste do pig inercial.

Também desejo expressar meu agradecimento ao Metrô de São Paulo nas pessoas dos Srs. Lazarni, João Manuel e Francisco Valentim pela colaboração na realização do experimento metroviário, e ao Sr. Fábio Villas Boas pelo auxílio no segundo teste automotivo.

Agradeço à CAPES pelo financiamento fornecido a esta pesquisa.

RESUMO

A estimação de trajetórias em Navegação Inercial é um importante caso particular da estimação de estados na Teoria de Sistemas. A Navegação Inercial utiliza giroscópios (sensores que medem velocidade de rotação angular) e acelerômetros (sensores que medem aceleração) para manter estimativas de posição e velocidade do veículo no qual uma Central Inercial (conjunto de 3 acelerômetros e 3 giroscópios) está embarcada. Este veículo pode ser uma aeronave, um navio, um submarino, uma sonda ou um veículo terrestre. Os Sistemas de Navegação Inercial modernos normalmente utilizam o filtro de Kalman para calcular estimativas de trajetórias de veículos, o que constitui um sistema de fusão sensorial – nos filtros estocásticos há uma integração das medidas dadas pela Central Inercial com medidas de referência (que podem ser medidas de posição por um odômetro, ou medidas de velocidade dadas por um velocímetro, por exemplo). Uma aplicação em estudo recentemente é a implementação de Sistemas de Navegação Inercial através de filtros de partículas. Filtros de partículas são filtros preditivos destinados à estimação de estados em sistemas, sendo baseados no método de Monte Carlo sequencial. Estes filtros geram amostras (ou partículas) ponderadas para aproximar uma determinada função de probabilidade. Nosso objetivo é aplicar tanto o filtro de Kalman quanto os filtros de partículas à estimação de um veículo contendo uma Central Inercial, e posteriormente comparar o desempenho destes dois tipos de filtro. Neste contexto, uma aplicação particularmente importante para a indústria petrolífera é a estimação de trajetórias de um pig inercial. Pig é um cilindro dotado de sensores, sendo utilizado para inspecionar dutos na indústria petrolífera (um pig move-se no interior dos dutos impulsionado pelo óleo). Embarcando-se uma Central Inercial e um odômetro no pig, compõe-se um Sistema de Navegação Inercial de fusão sensorial, o qual utiliza os filtros previamente mencionados para estimar a trajetória descrita pelo pig. Uma utilidade da estimação de trajetória do pig é a reconstrução do mapeamento de dutos, que é de grande valia em trechos onde o mapeamento documentado é pobre ou pouco especificado. Este trabalho descreve a implementação de um Sistema de Navegação Inercial do tipo *strapdown* com uma Central Inercial de baixo custo para estimação de trajetórias de veículos. Este sistema realiza estimações utilizando o filtro de Kalman e os filtros de partículas, e são apresentados os resultados de vários tipos de testes: dois testes foram realizados num automóvel, um teste foi realizado no Metrô e o último e mais importante foi realizado usando-se um pig inercial. Analisando os resultados destes testes, podemos tirar conclusões a respeito do funcionamento e aspectos práticos de aplicação dos filtros supracitados.

ABSTRACT

The trajectory estimation in Inertial Navigation is an important case of the state estimation in the Systems Theory. The Inertial Navigation makes use of accelerometers and gyroscopes to estimate position and velocity of a vehicle in which an Inertial Measurements Unit (IMU) (cluster of 3 accelerometers and 3 gyroscopes) is embedded. This vehicle can be an airplane, a ship, a submarine or a land vehicle. The modern Inertial Navigation Systems are aided by a Kalman Filter, in order to estimate vehicles' trajectories – it creates a sensor fusion system. In stochastic filters, there is an integration of the IMU measurements with reference measurements (position given by an odometer, or velocities given by a velocimeter, for example). An important application that has been studied is the implementation of Inertial Navigation Systems using particle filters. Particle filters are predictive filters that can be applied to the states estimation in systems, being based on the sequential Monte Carlo method. These filters yield weighted samples to approximate a certain probability density function. Then, our aim is to apply the Kalman Filter and the particle filters to the position estimation of a vehicle carrying an IMU, and to compare the performance of these 2 types of filter. In this context, an important application for the petroleum industry is the trajectory estimation of an inertial pig. Pig is a cylinder composed by sensors, and it is used to inspect pipelines (a pig moves inside the pipelines pushed by the oil). By mounting an IMU and an odometer in the pig we get a sensor fusion Inertial Navigation System, that applies the filters to estimate the pig's trajectory. An utility of the trajectory estimation of the pig is the map reconstruction of the pipelines, that is valuable in passages where the documentation is poorly specified. This work describes the implementation of a Strapdown INS with a low-cost IMU to estimate vehicles' trajectories. This system uses the Kalman filter and the particle filters, and many experimental results are presented: two tests are automobillistic, one is done in the Metro and the most important one uses an inertial pig. By the analysis of these tests results we draw conclusions about the behaviour and the practical aspects of the application of these filters.

SUMÁRIO

1	1. INTRODUÇÃO
10	2. ESTIMADORES DE ESTADO
10	2.1. Introdução
12	2.2. O Filtro de Kalman
19	2.3. Os Filtros de Partículas
20	2.3.1. O Filtro Sequencial Importance Sampling (SIS)
23	2.3.1.1. O Problema da Degeração
24	2.3.1.2. Escolha Adequada da Função Densidade de Importância
26	2.3.1.3. Reamostragem (Resampling)
27	2.3.2. O Filtro Sampling Importance Resampling (SIR)
28	2.3.3. O Filtro Auxiliary Sampling Importance Resampling (ASIR)
30	2.3.4. O Filtro Metropolis Hastings Importance Resampling (MHIR)
30	2.3.4.1. O Método Markov Chain Monte Carlo (MCMC)
35	2.3.4.2. O Filtro MHIR
37	2.4. Exemplo de Aplicação
43	3. NAVEGAÇÃO INERCIAL E MODELAGEM DO PROBLEMA
43	3.1. Conceitos Básicos de Navegação Inercial
47	3.2. Sistemas de Coordenadas
50	3.3. Transformações de Coordenadas
53	3.4. Modelos de Erro de Aceleração
59	3.5. Influência dos Erros dos Sensores na Estimação de Posição
60	3.6. Alinhamento Inicial
61	3.7. Modelagem do Sistema no Espaço de Estados Discreto
71	3.8. Especificações Relevantes da Central Inercial
73	4. TESTES REALIZADOS, RESULTADOS EXPERIMENTAIS E ANÁLISE
74	4.1. Algoritmos Utilizados e Ajustes
74	4.1.1. Duplo Integrador

4.1.2. O Filtro de Kalman.....	74
4.1.3. O Filtro SIR.....	76
4.1.4. O Filtro ASIR.....	78
4.1.5. O Filtro MHIR.....	80
4.2. Descrição do Teste Automotivo.....	84
4.3. Resultados Experimentais do Teste Automotivo.....	85
4.3.1. Os Ruídos do Processo.....	89
4.3.2. Estimação de Posição Utilizando o Duplo Integrador.....	92
4.3.3. Estimação de Posição Utilizando o Filtro de Kalman.....	93
4.3.4. Estimação de Posição Utilizando o Filtro SIR.....	95
4.3.5. Estimação de Posição Usando o Filtro ASIR.....	97
4.3.6. Estimação de Posição Usando o Filtro MHIR.....	99
4.4. O Experimento Metroviário.....	99
4.5. Experimento do Pig Instrumentado.....	112
4.6. Experimento Automotivo II.....	125
5. CONCLUSÕES.....	132
6. REFERÊNCIAS BIBLIOGRÁFICAS.....	136

LISTA DE FIGURAS

Figura 2.1 – Gráfico das medições utilizadas para a estimação de estados nos filtros.....	38
Figura 2.2 – Histograma do ruído do processo para o sistema dado no exemplo.....	39
Figura 2.3 – Histograma do ruído de medição para o sistema dado no exemplo.....	39
Figura 2.4 – Gráfico da trajetória de estados estimada pelo filtro de Kalman.....	40
Figura 2.5 – Gráfico da trajetória de estados estimada pelo filtro de partículas SIR.....	41
Figura 3.1 – Strapdown system e Gimbal system.....	45
Figura 3.2 – Processamento dos dados de uma Central Inercial do tipo Strapdown.....	46
Figura 3.3 – Sistemas de Coordenadas ECI e ECEF.....	48
Figura 3.4 – Sistema de Coordenadas ENU.....	49
Figura 3.5 – Sistema de Coordenadas NED.....	49
Figura 3.6 – Sistema de Coordenadas RPY.....	49
Figura 3.7 – Diagrama da Mecanização de um Sistema de Navegação Inercial.....	57
Figura 3.8 – Diagrama Esquemático da Inicialização e Estimação de posição para um Sistema de Navegação Inercial utilizando o filtro de Kalman.....	71
Figura 4.1 – Esquema do duplo integrador.....	74
Figura 4.2 – Diagrama esquemático do Filtro de Kalman.....	75
Figura 4.3 – Gráfico da aceleração no eixo x para o primeiro teste automotivo.....	85
Figura 4.4 – Gráfico da aceleração no eixo y para o primeiro teste automotivo.....	86
Figura 4.5 – Gráfico da aceleração no eixo z para o primeiro teste automotivo.....	86
Figura 4.6 – Gráfico da velocidade de rotação (eixo z) para o primeiro teste.....	87
Figura 4.7 – Gráfico da referência de velocidade no eixo x para o primeiro teste.....	87
Figura 4.8 – Gráfico da referência de velocidade no eixo y para o primeiro teste.....	88
Figura 4.9 – Gráfico da referência de velocidade no eixo z para o primeiro teste.....	88

90	Figura 4.10 – Gráfico do ruído de aceleração do acelerômetro x
90	Figura 4.11 – Gráfico do ruído de aceleração do acelerômetro y
91	Figura 4.12 – Histograma do ruído de aceleração do acelerômetro x
91	Figura 4.13 – Histograma do ruído de aceleração do acelerômetro y
92	Figura 4.14 – Trajetória do automóvel no plano XY estimada pelo duplo integrador
93	Figura 4.15 – Trajetória do automóvel no plano XY estimada pelo filtro de Kalman
94	Figura 4.16 – Gráfico do erro de aceleração no eixo x estimado pelo filtro de Kalman
95	Figura 4.17 – Histograma do erro de aceleração (x) estimado pelo filtro de Kalman
96	Figura 4.18 – Gráfico de trajetórias obtidas pelo filtro de Kalman e pelo filtro SIR com ruído gaussiano
97	Figura 4.19 – Trajetória do automóvel no plano XY estimada pelo filtro SIR
98	Figura 4.20 – Trajetória do automóvel no plano XY estimada pelo filtro ASIR
100	Figura 4.21 – Gráfico da aceleração da composição metrovária no eixo x
101	Figura 4.22 – Gráfico da aceleração da composição metrovária no eixo y
101	Figura 4.23 – Gráfico da aceleração da composição metrovária no eixo z
102	Figura 4.24 – Gráfico da velocidade de rotação angular do Metrô em torno do eixo z
102	Figura 4.25 – Gráfico da velocidade de rotação angular do Metrô em torno do eixo x
103	Figura 4.26 – Gráfico da velocidade de rotação angular do Metrô em torno do eixo y
103	Figura 4.27 – Gráfico da medição de referência (velocidade) do Metrô (eixo x)
104	Figura 4.28 – Gráfico da medição de referência (velocidade) do Metrô (eixo y)
104	Figura 4.29 – Gráfico da medição de referência (velocidade) do Metrô (eixo z)
105	Figura 4.30 – Gráfico da trajetória metrovária (linha verde) estimada pelo filtro de Kalman
106	Figura 4.31 – Gráfico da evolução temporal da coordenada x do Metrô estimada através do filtro de Kalman

Figura 4.32 – Gráfico da evolução temporal da coordenada y do Metrô estimada através do filtro de Kalman.....	107
Figura 4.33 – Gráfico da trajetória metroviária (linha verde) estimada pelo filtro SIR.....	108
Figura 4.34 – Gráfico da evolução temporal da coordenada x do Metrô (filtro SIR).....	108
Figura 4.35 – Trajetórias da linha verde do Metrô de São Paulo estimadas pelo filtro SIR e pelo filtro de Kalman em escala proporcional.....	109
Figura 4.36 – Aspecto da instrumentação utilizada no experimento metroviário.....	110
Figura 4.37 – Composição do Metrô utilizada para realizar o experimento.....	110
Figura 4.38 – Painel do Metrô.....	111
Figura 4.39 – Notebook utilizado no experimento metroviário durante a coleta de dados no Metrô.....	111
Figura 4.40 – Gráfico da aceleração do pig no eixo x medida pela Central Inercial.....	114
Figura 4.41 – Gráfico da aceleração do pig no eixo y medida pela Central Inercial.....	115
Figura 4.42 – Gráfico da aceleração do pig no eixo z medida pela Central Inercial.....	115
Figura 4.43 – Gráfico da aceleração do pig (eixo x) no Sistema de coordenadas NED.....	116
Figura 4.44 – Gráfico da aceleração do pig (eixo y) no Sistema de coordenadas NED.....	116
Figura 4.45 – Gráfico da aceleração do pig (eixo z) no Sistema de coordenadas NED.....	117
Figura 4.46 – Gráfico da velocidade de rotação angular do pig em torno do eixo z.....	117
Figura 4.47 – Gráfico da velocidade do pig (eixo x) no sistema de coordenadas NED.....	118
Figura 4.48 – Gráfico da velocidade do pig (eixo y) no sistema de coordenadas NED.....	118

Figura 4.49 – Gráfico da velocidade do pig (eixo z) no sistema de coordenadas NED.....	119
Figura 4.50 – Gráfico da trajetória do pig reconstruída através do filtro de Kalman.....	120
Figura 4.51 – Montagem da Central Inercial no interior do pig instrumentado utilizado no experimento feito no duto de testes da Petrobras.....	122
Figura 4.52 – Aspecto da eletrônica embarcada utilizada no pig instrumentado para adquirir dados da Central Inercial e do odômetro.....	122
Figura 4.53 – Pig instrumentado.....	123
Figura 4.54 – Pig instrumentado (vista total) na bancada de um dos laboratórios do CENPES.....	123
Figura 4.55 – Pig sendo inserido na tubulação de testes da Petrobras pela equipe do Sr. Claudio Camerini.....	124
Figura 4.56 – Pig sendo inserido na tubulação de testes da Petrobras.....	124
Figura 4.57 – Pig sendo retirado da tubulação de testes da Petrobras.....	125
Figura 4.58 – Gráfico comparativo entre a rotação original e a rotação estimada pelo filtro de Kalman.....	126
Figura 4.59 – Gráfico da coordenada x do veículo estimada pelo filtro de Kalman.....	127
Figura 4.60 – Gráfico da coordenada y do veículo estimada pelo filtro de Kalman.....	127
Figura 4.61 – Gráfico comparativo entre a rotação original e a rotação estimada pelo filtro SIR.....	128
Figura 4.62 – Gráfico comparativo entre a evolução temporal da coordenada x do veículo estimada pelo filtro SIR e a evolução real da coordenada x.....	129
Figura 4.63 – Gráfico comparativo entre a evolução temporal da coordenada y do veículo estimada pelo filtro SIR e a evolução real da coordenada y.....	130
Figura 4.64 – Instrumentação utilizada na realização do segundo experimento automotivo.....	130
Figura 4.65 – Montagem da instrumentação no interior do automóvel.....	131

LISTA DE ABREVIATURAS

UMI	Unidade de Medidas Inercial
SNI	Sistema de Navegação Inercial
SIS	Sequential Importance Sampling
SIR	Sampling Importance Resampling
ASIR	Auxiliary Sampling Importance Resampling
MCMC	Markov Chain Monte Carlo
MHIR	Metropolis Hastings Importance Resampling
ECI	Earth Centered Inertial
ECEF	Earth Centered Earth Fixed
LTP	Local Tangent Plane
ENU	East North Up
NED	North East Down
RPY	Roll Pitch Yaw
MEMS	Micro Electro Mechanical Machines
FOG	Fiber Optic Gyro

LISTA DE SÍMBOLOS

x_k	Vector de estados de um sistema no instante de tempo k
y_k	Vector de medidas de referência de um sistema no instante k
u_k	Vector de entradas de um sistema no instante k
wr_k	Vector de ruídos de processo de um sistema no instante k
e_k	Vector de ruídos de medição de um sistema no instante k
$x_k^i(t)$	Amostra i do vector de estados do sistema no instante k
$p(x)$	Função densidade de probabilidades do vector x
$P(x)$	Função distribuição de probabilidades do vector x
$p(y/x)$	Função densidade de probabilidades condicional
N	Número total de amostras de um vector num dado instante de tempo
$w^k(i)$	Peso associado à amostra i do vector x no instante de tempo k
f_k	Função não-linear que associa o estado x_k ao estado x_{k-1}
g_k	Função não-linear que associa a entrada u_k ao estado x_{k+1}
h_k	Função não-linear que associa o estado x_k à medição de referência y_k
$p(x_0)$	Densidade de probabilidades a priori do estado x_k
$E(x_k)$	Valor esperado do vector aleatório x_k
m_k	Média do vector aleatório x no instante k
$\hat{x}_{k/k-1}$	Estimativa ótima do estado x_k dadas as medições y_1, \dots, y_{k-1}
K	Matriz de ganho do filtro de Kalman
P	Matriz de covariância do erro do filtro de Kalman
$E(x/y)$	Valor esperado condicional

$\text{Var}(x_k)$	Variancia do vetor aleatório x_k
\hat{A}	Linearização local da função não-linear $f(x)$
$E(f(x))$	Valor esperado de uma função do vetor aleatório x
$\delta(x)$	Função Impulso (delta de Dirac)
$q(x)$	Função densidade de importância do vetor x
N_{er}	Tamanho amostral efetivo
$\mu_k(i)$	Caracterização do estado x_k dada a amostra $x_{k-1}(i)$
$P(x_{k+1}/x_k)$	Núcleo de transição de uma cadeia de Markov
$\phi(\cdot)$	Distribuição estacionária de uma cadeia de Markov
$\bar{f}(x)$	Média ergódica da função $f(x)$
$\alpha(x,y)$	Probabilidade de aceitação de um candidato y numa cadeia de Markov
$I(\cdot)$	Função indicadora
u	Variável aleatória com distribuição uniforme
P_a	Probabilidade de aceitação de um candidato no algoritmo MHR
\hat{x}_k	Estimativa do estado x_k
C_X^Y	Matriz de rotação de coordenadas do sistema X para o sistema Y
T	Período de amostragem de um sistema
a	Vetor de acelerações de um veículo
ω	Vetor de velocidades de rotação de um veículo
p	Vetor de ângulos de rotação descritos por um veículo
Ω	Vetor de velocidades de rotação da Terra
λ	Longitude atual do veículo
ϕ	Latitude atual do veículo
R	Raio da Terra

G	Constante gravitacional universal
M_T	Massa da Terra
g	Aceleração da gravidade na superfície terrestre
b	Bias de um giroscópio ou de um acelerômetro
ea	Erro na medida de aceleração de um veículo
ev	Erro na estimativa da velocidade de um veículo
ep	Erro na estimativa da posição de um veículo
d	Vetor de posições de um veículo
v	Vetor de velocidades de um veículo
δa	Vetor de erros de medição de aceleração

1. Introdução

A estimação de estados é um dos mais antigos problemas na teoria dos sistemas. Como fazer para obtermos estados ocultos de um sistema, dadas as observações (ou medições) realizadas? Muitas técnicas foram desenvolvidas para tratar esse tipo de problema. As primeiras técnicas consideravam o sistema linear e sem ruídos associados - assim, seus estados poderiam ser determinados analiticamente, uma vez conhecida a dinâmica do sistema (obtida através de modelagem matemática). Outras técnicas mais refinadas consideravam ruídos associados à dinâmica do sistema - uma dessas técnicas é o filtro de Kalman (desenvolvido na década de 60), que ainda é utilizado nos dias de hoje para estimação de estados de sistemas. Apesar do filtro de Kalman apresentar um desempenho satisfatório na maioria das aplicações em que é utilizado, devemos nos certificar de que algumas hipóteses são satisfeitas para usá-lo. Essas hipóteses, se satisfeitas, garantem que o desempenho do filtro de Kalman na estimação de estados será ótimo. Entrementes, essas hipóteses são restritivas (por exemplo, o modelo do sistema deve ser linear, o ruído associado deve ser gaussiano, e na maioria dos sistemas reais elas não são satisfeitas. Dessa forma, o desempenho do filtro de Kalman fica degradado) apesar de ainda assim podermos utilizá-lo com sucesso), pois algumas aproximações devem ser feitas (o filtro de Kalman entendido, por exemplo, é uma aproximação do filtro de Kalman para sistemas não-lineares). Nestes casos, o uso dos filtros de partículas (que são filtros baseados na aplicação sequencial dos métodos de Monte Carlo [1]) pode ser mais

adequado, já que eles aproximam as densidades de probabilidades requeridas nos cálculos de estimação por um número finito de amostras (ou 'partículas').

O uso dos filtros de partículas na estimação de estados é relativamente recente (metade da década de 90), sendo que em aplicações reais eles ainda são muito pouco utilizados. Nesse sentido, nosso objetivo é aplicar tanto o filtro de Kalman quanto os filtros de partículas na estimação de posição de um veículo a partir dos dados fornecidos por uma Central Inercial, e posteriormente comparar o desempenho dos dois tipos de filtro. Para isso, faz-se necessário uma modelagem matemática adequada do funcionamento da Central Inercial (em qualquer problema de aplicação real, a adequação do modelo matemático é fundamental para que possamos tirar conclusões a respeito do funcionamento dos estimadores). Por ora, façamos uma breve revisão acerca dos conceitos básicos de Navegação Inercial.

Sensores inerciais são sensores utilizados para medir taxa de rotação e aceleração. Os sensores inerciais mais usados são os giroscópios e os acelerômetros. Os giroscópios são sensores que medem rotação - há giroscópios que medem velocidade de rotação (*rate gyroscopes*) e giroscópios que medem ângulo de rotação (*displacement gyroscopes*). Os acelerômetros são sensores que medem aceleração, como o próprio nome insinua. Entretanto, um acelerômetro não pode medir aceleração gravitacional, ou seja, um acelerômetro em queda livre não tem nenhuma entrada detectável. O eixo de entrada de um sensor inercial define qual componente do vetor ele deve medir (considerando o espaço tridimensional, temos 3 eixos: x, y e z). Sensores de eixos múltiplos (*multiaxis sensors*) podem medir mais de uma componente do vetor (x, y, z). [2]

A Navegação Inercial utiliza giroscópios e acelerômetros para estimar posição e velocidade do veículo no qual a Central Inercial está embarcada. Esse veículo pode ser uma aeronave, um avião, um navio, um submarino, uma sonda, um veículo terrestre ou até uma nave espacial.

Um Sistema de Navegação Inercial é formado pelas seguintes partes: [2]

Para calcularmos a posição de um veículo a partir das suas medidas de acelerações, precisamos integrar o sinal de aceleração 2 vezes, e este é um processo que acumula erro no tempo (pois integrar um sinal 2 vezes no tempo equivale a multiplicar o valor deste sinal pelo quadrado do tempo). Assim, a presença de ruídos no sinal medido de aceleração pode tornar o processo de dupla integração inviável, pois conforme o tempo passa o erro acumulado se torna extremamente elevado (como será visto no Capítulo 3). A filtragem do sinal (através de filtros estocásticos como o filtro de

principalmente nas acelerações medidas. Cada rotação realizada pelo veículo. Além disso, outras correções são requeridas, de rotação do veículo), de modo a corrigir a orientação dos eixos dos acelerômetros a transformações são feitas utilizando-se as leituras dos giroscópios (i.e., velocidades transformações de coordenadas nas acelerações medidas pelos acelerômetros. Estas

Desta forma, nos sistemas do tipo *strapdown* há a necessidade de realizarmos transformações de coordenadas nas acelerações medidas pelos acelerômetros. Estas transformações são feitas utilizando-se as leituras dos giroscópios (i.e., velocidades de rotação do veículo), de modo a corrigir a orientação dos eixos dos acelerômetros a cada rotação realizada pelo veículo. Além disso, outras correções são requeridas, principalmente nas acelerações medidas.

de movimento com 6 graus de liberdade para modelar o sistema [2].

Ha muitos tipos de arquitetura para Centrais Inerciais, cada um com uma característica de desempenho diferente, mas há 2 categorias básicas de arquitetura: *gimballed e strapdown*. Nos *gimballed systems* os acelerômetros são isolados de rotações externas do veículo - desse modo, podemos trabalhar com equações de movimento com 3 graus de liberdade. Já nos *strapdown systems* esses acelerômetros não são isolados de rotações externas, e assim é necessário que utilizemos equações

Computadores de Navegação (um ou mais), para efetuar os cálculos da aceleração gravitacional e fazer estimativas da posição do veículo a ser rastreado.

Kalman e os filtros de partículas), então, torna-se essencial para que consigamos estimar com sucesso a posição do veículo que carrega a Central Inercial.

As correções a serem feitas nas acelerações medidas podem ser determinísticas ou estocásticas. As correções determinísticas são fisicamente bem definidas, por exemplo: a aceleração da gravidade (que os acelerômetros não são capazes de medir), a aceleração centrífuga e a aceleração de Coriolis. Já as correções estocásticas, como o próprio nome insinua, são incertas e necessitam ser estimadas por filtros estocásticos, pois seu cálculo exato é inviável.

Nos filtros estocásticos, além das medidas dadas pelos acelerômetros e giroscópios, precisamos de medidas de referência (que pode ser uma medida de posição dada por um GPS, ou uma medida de velocidade do corpo dada por um velocímetro ou tacômetro no caso de um automóvel, por exemplo). Esta medida também terá um ruído associado, que será tratado pelo filtro, mas antes de utilizarmos uma medida de referência é necessário nos assegurarmos de que seu ruído não possui as mesmas características do ruído dos sensores de uma Central Inercial (que é cumulativo no tempo). Se isso acontecer, os filtros não funcionarão bem; caso contrário estes filtros estocásticos combinarão as diferentes características dos ruídos dos sensores da Central Inercial e do ruído das medidas de referência de modo a calcular uma estimativa precisa da posição do veículo considerado. Para uma utilização adequada do filtro de Kalman e dos filtros de partículas (ou seja, dos filtros estocásticos em geral), é necessário que façamos uma modelagem correta do sistema em questão. Neste modelo, as principais características do sistema devem ser definidas, e isto é crucial para o correto funcionamento da filtragem dos sinais.

O modelo de um sistema no espaço de estados discreto pode ser escrito da seguinte forma:

$$x_{k+1} = A \cdot x_k + B \cdot u_k + C \cdot w_k$$

$$y_k = H \cdot x_k + G \cdot e_k$$

(1.1)

Uma vez que este modelo foi definido, pode-se aplicar técnicas de estimação de estados a ele. Para a aplicação do filtro de Kalman, o sistema deve ser linear e os seus ruídos devem ser gaussianos – se estas hipóteses não forem satisfeitas (o que ocorre em muitos casos), pode-se ainda aplicar o filtro de Kalman estendido ao sistema. Contudo, o desempenho da estimação de estados pode ser degradado, já que o filtro de Kalman estendido não é um filtro ótimo, apenas uma aproximação – algumas vezes essa aproximação não é realizada com sucesso. Assim sendo, os filtros de partículas podem constituir uma melhor alternativa para a estimação de estados de sistemas não lineares e com ruídos não gaussianos, já que estes filtros geram estimativas de estado baseado em amostras do ruído real do processo (se este ruído for não-gaussiano, os filtros de amostras podem funcionar bem).

para a estimação de estados. Física do sistema e nos ruídos associados aos processos do sistema. O vetor de estados). Feito isso, devemos definir as matrizes da equação (1.1) baseando-se na entradas é formado pelas medidas de que dispomos para processar as estimativas dos entradas (normalmente, o vetor de estados é aquele que se quer estimar, e o vetor de primeiro passo é definirmos as componentes do vetor de estados e do vetor de de medições de referência a serem utilizadas. Quando modelamos um sistema, o primeira equação define a dinâmica do sistema, enquanto a segunda define o modelo h, que são funções não-lineares) são matrizes que definem a dinâmica do sistema. A referência (ou observações) e e é o vetor de erros de medição. A, B, C, G e H (ou F e Central Inercial), w é o vetor de ruídos do processo, y é o vetor de medições de o vetor de entradas (no caso, as acelerações e rotações angulares medidas pela , onde x é o vetor de estados do sistema (no caso, as posições a serem estimadas), n é

$$y_k = h(x_k) + G \cdot e_k$$

(1.2)

$$x_{k+1} = f(x_k, u_k) + C \cdot w_k$$

Ou, para o caso de o sistema ser não-linear:

Nesta função densidade de probabilidade gaussiana, temos que o vetor de médias é nulo e a matriz de covariâncias é a identidade, mas dependendo da aplicação esses parâmetros podem ser alterados. Temos ainda que n é a dimensão do vetor $y_k - H \cdot x_k$.

$$p(y_k / x_k) = \frac{\exp\left(-\frac{1}{2} \cdot (y_k - H \cdot x_k)^T \cdot (y_k - H \cdot x_k)\right)}{\sqrt{2\pi}^n} \quad (1.5)$$

Assume-se que x_k e e_k são independentes. A densidade $p_e(\cdot)$ deve ser escolhida de acordo com a aplicação, podendo ser inclusive uma gaussiana multivariável: [3]

$$p(y_k / x_k) = p_e(y_k - H \cdot x_k) \quad (1.4)$$

A função densidade de probabilidade $p(y_k / x_k)$ é definida de acordo com a equação (1.1) (ou seja, queremos saber a probabilidade de y_k sendo que x_k é conhecido, onde y_k é a medição de referência disponível no instante k):

$$w_k(t) = w_{k-1}(t) \cdot p(y_k / x_k(t)) \quad (1.3)$$

Após isso, deve-se associar um peso a essa amostra de x_k . Isso é feito segundo a equação abaixo (assim como o estado x_{k-1} deve ser conhecido através da inicialização do algoritmo, o peso w_{k-1} associado a ele também deve ser conhecido):

$$2. \text{ Feito isso, basta utilizar o modelo do sistema: } x_k(t) = A \cdot x_{k-1}(t) + B \cdot u_k + C \cdot w_k$$

1. Primeiramente, extraí-se uma amostra do ruído do processo, w_k

utilizando-se a equação (1.1):

(assume-se que x_{k-1} é conhecido, e queremos estimar x_k). Isso pode ser feito gerando uma amostra a partir da função densidade de probabilidade [3] $p(x_k / x_{k-1}(t))$ será visto com maiores detalhes no Capítulo 2). O primeiro passo do algoritmo é Para exemplificar o uso de um filtro de partículas, consideremos o filtro SIR (que

No Capítulo 2 serão apresentados alguns estimadores de estado, entre eles o filtro de Kalman, o filtro de Kalman estendido e os filtros de partículas (entre outros, o filtro SIR (*Sampling Importance Resampling*, ou Amostragem e Reamostragem por Importância), o filtro ASIR (*Auxiliary Sampling Importance Resampling*, ou Amostragem e Reamostragem por Importância Auxiliar) e o filtro MHIR (*Metropolis Hastings Importance Resampling*, ou Metropolis Hastings e Reamostragem por Importância)). No Capítulo 3 serão apresentados os conceitos de Navegação Inercial e a modelagem do problema, já deixando o sistema pronto para a aplicação dos filtros. No Capítulo 4 serão apresentados os resultados experimentais obtidos (foram realizados experimentos com a Central Inercial – um deles foi o teste automotivo, no

Escolhendo-se um valor adequado para N , faz-se a média acima convergir para o valor esperado do estado x_k condicionado nas medições disponíveis $\{y_1, \dots, y_k\}$ (de acordo com o método de Monte Carlo). Assim, os filtros de amostras consistem numa aplicação sequencial (ou seja, dinâmica) do método de Monte Carlo [1].

$$\hat{x}_k = \frac{1}{N} \sum_{i=1}^N x_k^i(t) \quad (1.7)$$

suas N amostras ponderadas:

Ao final do processo, pode-se calcular uma estimativa do estado x_k com base nas

relativamente baixo, concentrando-se apenas naquelas que têm um peso maior. A seguir, efetua-se uma reamostragem (a reamostragem será descrita em maiores detalhes no Capítulo 2). A ideia básica é eliminar as amostras que têm um peso

$$w_k^i(t) = \frac{\sum_{i=1}^N w_k^i(t)}{N} \quad (1.6)$$

Os passos descritos acima devem ser repetidos N vezes (onde N é o número de amostras do estado x_k que se quer obter), e assim obteremos N amostras do estado x_k , cada uma com um peso associado. Deve-se, então, normalizar estes pesos:

qual os dados fornecidos pela Central durante um percurso automobilístico foram coletados e processados pelos programas que executam os filtros de Kalman e de partículas, sendo obtidas estimativas da trajetória descrita pelo automóvel) e as análises e comparações entre os desempenhos dos diversos filtros na estimação de posição, bem como os ajustes necessários para executar estes filtros. Finalmente, no Capítulo 5 serão tiradas as conclusões sobre o trabalho realizado.

Um experimento particularmente importante para a indústria petrolífera que foi realizado neste trabalho foi o teste do pig inercial. Pig é um cilindro dotado de sensores utilizado para inspecionar dutos na indústria petrolífera. Um tipo de pig instrumentado é o pig ultra-sônico, que detecta falhas em dutos através da emissão e recepção de ondas de ultra-som (um pig move-se no interior dos dutos impulsionado pelo óleo). Os dados coletados pelos sensores ultra-sônicos são armazenados para processamento off-line. Para que se possa saber onde estas falhas foram detectadas, torna-se necessário estimar a posição do pig. Isto pode ser realizado embarcando-se uma Unidade de Medidas Inercial (UMI) no pig. Esta UMI coleta medidas de aceleração e velocidade de rotação angular do pig e armazena estes dados. Utilizando os dados fornecidos pela UMI e por um odômetro fixado no pig, compõe-se um Sistema de Navegação Inercial de fusão sensorial, o qual utiliza o Filtro de Kalman para estimar a trajetória descrita pelo pig. Uma outra utilidade da estimação de trajetória do pig é a reconstrução do mapeamento de dutos, que é de grande valia em trechos onde o mapeamento documentado é pobre ou pouco especificado.

Deve-se ainda atentar para o seguinte: o objetivo deste trabalho é calcular estimativas de posição de um veículo no qual uma Central Inercial está embarcada a partir das suas medidas (acelerações em três eixos ortogonais e velocidades de rotação angular em torno de três eixos ortogonais), utilizando-se o filtro de Kalman e os filtros de partículas. Uma vez que estas estimativas foram obtidas, podemos compará-las com as posições reais do veículo a fim de confrontar o desempenho destes dois tipos de filtros em aplicações relacionadas à Navegação Inercial.

Finalmente, os resultados experimentais obtidos sugerem que os filtros de partículas podem ter melhores desempenhos que o filtro de Kalman em aplicações relacionadas à Navegação Inercial nas quais o ruído do processo é não-gaussiano. Entretanto, estes testes também sugerem que os filtros de partículas são extremamente dependentes da aplicação em que são utilizados, sendo que sua sintonia é mais difícil que a do filtro de Kalman (o qual mostrou mais versatilidade).

2. Estimadores de Estado

2.1. INTRODUÇÃO

Neste capítulo, descreveremos dois métodos utilizados para estimar estados de um sistema: o filtro de Kalman e os filtros de partículas. O primeiro foi desenvolvido na década de 60, enquanto que os filtros de partículas (ou *particle filters*, como são conhecidos em inglês) foram desenvolvidos mais recentemente, em meados da década de 90. Eles funcionam como uma generalização do filtro de Kalman para casos em que temos sistemas não lineares e ruídos não gaussianos.

Seja o sistema abaixo:

$$x_{k+1} = f_k(x_k) + g_k(u_k) + C_k \cdot w_k \quad (2.1)$$

$$y_k = h_k(x_k) + G_k \cdot e_k \quad (2.2)$$

Neste modelo, x_k representa o vetor de estados do sistema, u_k representa o vetor de entradas do sistema (supõe-se que esse vetor seja conhecido a priori), w_k representa o vetor de ruídos associados ao sistema (forças externas não medidas, por exemplo), y_k representa o vetor de medições (ou observações) realizadas e e_k representa o vetor de erros de medição. As funções f_k , g_k e h_k podem ser lineares ou não-lineares, e C_k e G_k são matrizes associadas aos ruídos. A equação (2.1) modela o comportamento dinâmico do sistema, enquanto que a equação (2.2) modela as medições de referência do sistema.

Nossa meta pode ser definida como estimação recursiva da sequência de estados x_k a partir do conjunto de medições disponíveis até o instante k : $\{y_1, y_2, \dots, y_k\}$. Desta forma, o problema do rastreamento (estimação de posição) incorre na associação de incertezas ao valor que o estado x_k pode assumir (estimativa do estado). O que calcularemos através de estimativas é o valor mais provável a ser assumido pelo estado baseado nas medições $\{y_k\}$ - assim, a obtenção dessas estimativas está diretamente relacionada ao cálculo da função densidade de probabilidades $p(x_k / y_1, \dots, y_k)$. Assume-se que a densidade inicial $p(x_0 / y_0) = p(x_0)$ (densidade a priori) é conhecida. A princípio, a função densidade de probabilidade $p(x_k / y_1, \dots, y_k)$ pode ser obtida recursivamente em 2 etapas: predição e atualização. [1]

Supõe-se que a função densidade de probabilidade $p(x_{k-1} / y_1, \dots, y_{k-1})$ no instante $k - 1$ é conhecida. A etapa de predição consiste, então, na utilização do modelo do sistema dado pela equação (2.1) para obter a função densidade de probabilidades a priori do estado no instante k através da equação de Chapman - Kolmogorov [1]:

$$p(x_k / y_1, \dots, y_{k-1}) = \int p(x_k / x_{k-1}) p(x_{k-1} / y_1, \dots, y_{k-1}) dx_{k-1} \quad (2.3)$$

Note-se que na equação acima foi utilizado o fato de que $p(x_k / x_{k-1}, y_1, \dots, y_{k-1}) = p(x_k / x_{k-1})$, visto que o sistema definido pela equação (2.1) é um processo de Markov de primeira ordem. O modelo probabilístico da evolução temporal dos estados x_k ($p(x_k / x_{k-1})$) é definido pela equação (2.1) e pelas características estatísticas do ruído w_k .

No instante de tempo k , uma nova medição y_k é disponibilizada, e assim devemos usá-la para atualizar a função densidade de probabilidade a priori através da regra de Bayes (etapa de atualização) (considera-se que $p(y_k / x_k, y_1, \dots, y_{k-1}) = p(y_k / x_k)$):

$$(2.4) \quad p(x_k / y_1, \dots, y_k) = \frac{p(y_k / x_k) p(x_k / y_1, \dots, y_{k-1})}{p(y_k / y_1, \dots, y_{k-1})}$$

A função densidade de probabilidade no denominador da expressão acima é dada por:

$$(2.5) \quad p(y_k / y_1, \dots, y_{k-1}) = \int p(y_k / x_k) p(x_k / y_1, \dots, y_{k-1}) dx_k$$

Essa função densidade de probabilidade depende da função $p(y_k / x_k)$, definida pela equação de medição (2.2) e pelas características estatísticas de e_k . Na etapa de atualização, a medição y_k é usada para modificar a função densidade de probabilidade a priori, obtendo-se assim a função densidade de probabilidade requerida. [1]

As equações (2.3) e (2.4) constituem a base para a obtenção da solução Bayesiana ótima do problema acima mencionado. Entretanto, essa propagação recursiva das funções densidades de probabilidade não pode ser determinada analiticamente. O que existem são soluções para conjuntos restritos de casos, como o filtro de Kalman. Há também soluções aproximadas, que cobrem um conjunto mais amplo de situações, como o filtro de Kalman estendido e os filtros de partículas (ou *particle filters*). Estes filtros são apresentados a seguir.

2.2. O FILTRO DE KALMAN

O filtro de Kalman é um algoritmo recursivo para estimação de estados x_k de um sistema dinâmico a partir dos valores das medições $\{y_0, y_1, y_2, \dots, y_k\}$. Assume-se que as equações que regem o comportamento do sistema dinâmico cujos estados queremos estimar são as seguintes: [4]

$$x_{k+1} = A_k x_k + B_k u_k + C_k w_k \quad (2.6)$$

$$y_k = H_k x_k + G_k w_k \quad (2.7)$$

Temos que $\{w_k\}$ é um processo aleatório ruído branco [3] representado por um vetor de m componentes. É assumido que w_k tem matriz de covariância unitária (ou seja, $E(w_k w_k^T) = I_m$) e que a variável aleatória x_0 e o processo $\{w_k\}$ são descorrelacionados. Assume-se ainda que a média da variável x_0 (dada por m_0) e sua covariância (dada por P_0) são conhecidas. As matrizes A_k , B_k , C_k , H_k e G_k podem variar no tempo, sendo que a matriz G deve obedecer à condição: [4]

$$G_k^T G_k > 0$$

Se essa condição não for respeitada, existirão vetores λ tais que $\lambda^T G_k = 0$, e desse modo: [4]

$$\lambda^T y_k = \lambda^T H_k x_k + \lambda^T G_k w_k = \lambda^T H_k x_k$$

Ou seja, algumas combinações lineares de componentes de x_k poderão ser medidas *exatamente*. Assim, esta condição assegura que todas as medições y_k e combinações lineares de medições têm um ruído associado.

A sequência u_k representa as entradas do sistema (é um vetor de n componentes). Assumiremos que u_k é uma sequência determinística. Já a sequência $\{x_k\}$ é um processo estocástico [3], ou aleatório. Denotaremos por $\hat{x}_{k|j}$ o melhor estimador linear de x_k dados y_1, \dots, y_j , e por $x_{k|j}$ o erro ($x_k - \hat{x}_{k|j}$). A forma mais utilizada para esse estimador de estados é dada pelo estimador 'um passo à frente' (ou *one step ahead*): $\hat{x}_{k/k-1}$.

Para o sistema dinâmico descrito pelas equações (2.6) e (2.7), satisfeitas as suposições acima mencionadas, o estimador $\hat{x}_{k/k-1}$ obedece a seguinte equação recursiva: [4]

$$\hat{x}_{k+1/k} = A_k \cdot \hat{x}_{k/k-1} + B_k \cdot u_k + K_k \cdot [y_k - H_k \cdot \hat{x}_{k/k-1}] \quad (2.8)$$

A matriz de ganho K_k é dada por: [4]

$$K_k = [A_k \cdot P_k \cdot H_k^T + C_k \cdot G_k^T] \cdot [H_k \cdot P_k \cdot H_k^T + G_k \cdot G_k^T]^{-1} \quad (2.9)$$

E P_k é a matriz de covariância do erro: [4]

$$P_k = E[(x_k - \hat{x}_{k/k-1})(x_k - \hat{x}_{k/k-1})^T] \quad (2.10)$$

P_k satisfaz a equação recursiva de Riccati: [4]

$$P_{k+1} = A_k \cdot P_k \cdot A_k^T + C_k \cdot C_k^T - [A_k \cdot P_k \cdot H_k^T + C_k \cdot G_k^T] \cdot [H_k \cdot P_k \cdot H_k^T + G_k \cdot G_k^T]^{-1} \cdot [A_k \cdot P_k \cdot H_k^T + C_k \cdot G_k^T]^T \quad (2.11)$$

$$P_0 = P$$

Se assumirmos que as variáveis $(x_0, w_{T0}, w_{T1}, \dots)$ têm distribuição conjunta gaussiana [3], tal que $\{w_{Tj}\}$ é um ruído branco gaussiano, então:

$$\hat{x}_{k/k-1} = E[x_k / y_0, y_1, y_2, \dots, y_{k-1}] \quad (2.12)$$

Para implementar o filtro de Kalman, a sequência P_k deve ser calculada a partir da equação de Riccati, e a sequência correspondente de matrizes de ganho K_k deve ser calculada utilizando-se a equação (2.9). Todos esses cálculos podem ser feitos off-line, ou seja, antes que as medições sejam realizadas. Após isso, pode-se calcular

$x^{k/k-1}$ recursivamente usando-se a equação (2.8), conforme novas medições forem obtidas. O fato de que todos os coeficientes na equação (2.8) podem ser pré-calculados significa que precisamos realizar pouco processamento de dados on-line, e isso é muito importante quando os recursos computacionais são escassos. [4]

O CÁLCULO DE P_k

Para o sistema genérico descrito pelas equações (2.6) e (2.7), o filtro de Kalman é implementado através do cálculo off-line da sequência de matrizes de ganho K_k , o que depende do cálculo da sequência de covariâncias P_k - após isso, podemos utilizar a equação (2.8) e proceder ao cálculo das estimativas dos estados on-line. A princípio, o cálculo de P_k pode ser feito utilizando-se a equação de Riccati, mas este não é o modo mais adequado para isto, pois a equação (2.11) pode se tornar instável. Os 2 primeiros termos à direita na equação (2.11) são simétricos e positivos definidos, mas o último termo é subtraído dos outros 2 termos - desse modo, nada garante que P_{k+1} seja positiva definida - e se ela não o for, a equação de Riccati pode se tornar completamente instável. Um modo mais adequado de realizar o cálculo de P_k é através das equações abaixo: [4]

$$K_k = [A P_k H^T + C G^T] [H P_k H^T + G G^T]^{-1} \quad (2.13)$$

$$P_{k+1} = (A - K_k H) P_k (A - K_k H)^T + (C - K_k G) (C - K_k G)^T \quad (2.14)$$

Esse método é muito melhor, já que P_{k+1} é expressa como uma soma de termos positivos definidos.

SISTEMAS INVARIANTES NO TEMPO

Supõe-se agora que as matrizes A, B, C, G e H do modelo do sistema dado por (2.6) e (2.7) são invariantes no tempo (ou seja, não dependem de k). Neste caso, a equação de Riccati será expressa por: [4]

$$P = A.P.A^T + C.C^T - [A.P.H^T + C.G^T][H.P.H^T + G.G^T]^{-1}[A.P.H^T + C.G^T]^T \quad (2.15)$$

Se a matriz de covariâncias iniciais P_0 satisfizer esta equação então, a partir de (2.11), temos que $P_k = P_0$ para todo k , e o filtro de Kalman será invariante no tempo (pois K_k será uma matriz constante).

A equação (2.15) representa um intercâmbio entre 2 efeitos opostos: conforme mais medições y_k tornam-se disponíveis, aprendemos mais sobre x_k , pois acumulamos mais informação acerca dos estados do sistema; por outro lado, a posição de x_k pode se tornar mais incerta com o passar do tempo, conforme os estados forem se distanciando de sua posição inicial. A matriz P_0 que satisfaz a equação de Riccati representa o limiar no qual estes 2 efeitos se equilibram - e como a estimativa da covariância do erro P_k é constante nesse caso, temos que o grau de incerteza associado ao distanciamento da posição inicial x_0 também será constante. [4]

Quando falamos de sistemas invariantes no tempo, 3 questões a respeito da equação de Riccati são de fundamental importância: sob que condições a equação de Riccati terá solução? Pode haver mais de uma solução? Se partirmos de uma matriz inicial P_0 que não seja solução da equação de Riccati a sequência $P_0, P_1, P_2, \dots, P_k$ irá convergir para P (onde P é uma solução da equação de Riccati)? As respostas a estas questões estão relacionadas às propriedades de detectabilidade e estabilizabilidade (para revisão destes conceitos, ver [4]). Suponhamos, por exemplo, que um dos estados $x_k(t)$ seja completamente não-observável através das medições. Então, a melhor estimativa para $x_k(t)$ é a sua média, $E[x_k(t)]$, e o erro médio quadrático de estimação é $E[x_k(t) - E(x_k(t))]^2 = \text{Var}(x_k(t))$. Esta variância convergirá apenas se $x_k(t)$ for estável. [4]

Para encarminharmos de modo adequado as respostas às 3 questões acima, sejam as matrizes A e C definidas por:

$$A = A - C.G^T(G.G^T)^{-1}H$$

$$C = C [I - G^T (G G^T)^{-1} G]$$

(a) Se o par (H, A) for detectável, existe ao menos uma solução não-negativa para a equação algébrica de Riccati dada pela equação (2.15). [4]

(b) Se o par (A', C') for estabilizável, então a solução P para a equação algébrica de Riccati é única e P_k convergirá para P conforme $k \rightarrow \infty$, onde P_k é a sequência gerada pela equação (2.11) com matriz de covariâncias iniciais P_0 arbitrária. A matriz $(A - K H)$ será estável, onde K é a matriz de ganho correspondente a P , ou seja: [4]

$$K = [A P H^T + C G^T] [H P H^T + G G^T]^{-1}$$

Note-se que o filtro de Kalman da equação (2.8) pode ser escrito da seguinte forma:

$$\hat{x}_{k+1/k} = (A - K_k H) \hat{x}_{k/k-1} + B u_k + K_k y_k \quad (2.16)$$

Os resultados acima mostram que, sob as condições mencionadas, esse sistema (dado pela equação (2.16)) será invariante no tempo (ou seja, $K_k = K$, para todo k) se $P_0 = P$ (sendo P uma solução da equação algébrica de Riccati). Se $P(0) \neq P$, a sequência K_k tenderá ao valor estacionário K conforme $k \rightarrow \infty$. Assim, o filtro é quase invariante no tempo para valores de k elevados, e o filtro é estável, pois a matriz $(A - K H)$ em (2.16) é estável. A convergência para o estado estacionário é normalmente rápida, o que justifica a prática de se utilizar o filtro invariante no tempo mesmo quando $P_0 \neq P$. [4]

O FILTRO DE KALMAN ESTENDIDO

Em muitas situações práticas, as hipóteses necessárias para utilizarmos o filtro de Kalman não são satisfeitas. O caso mais comum é aquele em que temos um sistema não-linear, definido por equações do tipo:

$$x^{k+1} = f^k(x^k) + g^k(u^k) + C^k w^k \quad (2.17)$$

$$y^k = h^k(x^k) + G^k w^k \quad (2.18)$$

Neste modelo, temos que as funções $f^k(\cdot)$, $g^k(\cdot)$ e $h^k(\cdot)$ são não-lineares. Para realizarmos a estimação de estados deste tipo de sistema, precisamos fazer uma linearização local das funções não-lineares. O filtro de Kalman entendido está baseado neste tipo de aproximação, e suas equações são as seguintes:

$$\hat{x}^{k+1/k} = f^k(\hat{x}^{k/k-1}) + g^k(u^k) + K^k [y^k - h^k(\hat{x}^{k/k-1})] \quad (2.19)$$

$$K^k = [A^k P^k H^k + C^k G^k] [H^k P^k H^k + G^k G^k]^{-1} \quad (2.20)$$

$$P^{k+1} = (A^k - K^k H^k) P^k (A^k - K^k H^k)^T + (C^k - K^k G^k) (C^k - K^k G^k)^T \quad (2.21)$$

$$P^0 = P$$

Temos que as matrizes \hat{A}^k e H^k são linearizações locais das funções não-lineares: [1]

$$\hat{A}^k = \Delta f^k(x) \Big|_{x=\hat{x}^k/k} \quad (2.22)$$

$$H^k = \Delta h^k(x) \Big|_{x=\hat{x}^k/k} \quad (2.23)$$

Assim, o filtro de Kalman entendido utiliza o primeiro termo de uma expansão em séries de Taylor de uma função não-linear. Um filtro deste tipo que utilizasse outros termos da expansão em séries de Taylor adicionaria uma complexidade tal ao sistema que o seu uso ficaria muito restrito. [1]

O filtro de Kalman entendido (assim como o filtro de Kalman) tem sido usado em muitas aplicações com a finalidade de estimar estados de sistemas. No entanto, há a suposição de que as funções densidade de probabilidade envolvidas sejam gaussianas (por exemplo, $p(x_k / y_1, \dots, y_k)$ deve ser gaussiana). Se as funções densidades de probabilidade reais não forem gaussianas, podemos não conseguir aproximá-las por gaussianas adequadamente - nestes casos, o uso do filtro de Kalman entendido poderá não apresentar desempenho satisfatório. O uso dos filtros de partículas pode, desse modo, representar um ganho no que se refere ao desempenho obtido na estimação de estados desses sistemas. Os filtros de partículas serão descritos a seguir.

2.3. OS FILTROS DE PARTÍCULAS

Filtros de partículas são filtros preditivos baseados nos métodos de Monte Carlo. Estes filtros utilizam amostras (ou partículas) para aproximar uma determinada função densidade de probabilidade. Para utilizarmos o tradicional filtro de Kalman, precisamos nos certificar de que o ruído do sistema seja gaussiano e o modelo do sistema no espaço de estados seja linear, ou então o filtro não terá o desempenho ótimo. Estas hipóteses normalmente não valem para a maioria dos casos práticos, para os quais normalmente temos modelos não - lineares e ruídos não gaussianos. Assim, faz-se necessário o uso de filtros cujo desempenho não seja reduzido nesses casos, ou seja, filtros que possamos aplicar a sistemas em que o modelo não é linear e o ruído não é gaussiano. Os filtros de partículas servem a esse propósito, utilizando uma abordagem baseada nos métodos de Monte Carlo dinâmicos. Nesse sentido, a aplicação sequencial dos métodos de Monte Carlo gera um conjunto de amostras (ou partículas) que devem aproximar fidedignamente uma densidade de probabilidade que queremos obter.

A integração através do método de Monte Carlo visa calcular o valor de $E[f(x)]$ (valor esperado de uma determinada função da variável aleatória x), amostrando n valores de $p(x)$ - $X_i, i = 1, 2, \dots, n$. A seguinte aproximação é então realizada:

$$p(x_0, \dots, x_k / y_1, \dots, y_k) \approx \sum_{i=1}^N w_k^i(t) \delta(\{x_0, \dots, x_k\} - \{x_0^i(t), \dots, x_k^i(t)\}) \quad (2.24)$$

instante k pode ser aproximada por: [1]

que $\sum_{i=1}^N w_k^i(t) = 1$. Dessa forma, a função densidade de probabilidade a posteriori no x_k é o conjunto de todos os estados até o instante k . Os pesos são normalizados tal conjunto de pontos auxiliares com pesos associados $\{w_k^i(t) : i = 1, \dots, N\}$, e $\{x_0, \dots, \dots, N\}$ é um conjunto de amostras aleatórias que caracterizam a função densidade de probabilidade $p(x_0, x_1, \dots, x_k / y_1, \dots, y_k)$, onde $\{x_0^i(t), \dots, x_k^i(t) : i = 1, \dots, N\}$ é um De modo a esmiuçar os detalhes do algoritmo, seja $\{x_0^i(t), x_1^i(t), \dots, x_k^i(t) : i = 1, \dots, N\}$

apresentado pelo filtro SIS aproxima-se da estimativa Bayesiana ótima. [1] da função densidade de probabilidade a posteriori desejada, e o resultado amostras cresce, essa representação de Monte Carlo torna-se um modelo aproximado tendo como base essas amostras e seus respectivos pesos. Conforme o número de dessa forma, podemos calcular estimativas da função densidade de probabilidade por um conjunto de amostras aleatórias, tendo cada amostra um peso associado - Monte Carlo. A ideia central é representar uma função densidade de probabilidade para implementação de um filtro Bayesiano recursivo através da abordagem de maioria dos filtros de partículas desenvolvidos recentemente. Ele utiliza uma técnica O filtro SIS é um algoritmo baseado no método de Monte Carlo, e constitui a base da

2.3.1. O FILTRO SEQUENTIAL IMPORTANCE SAMPLING (SIS)

aumentarmos a quantidade de amostras (n). aproximação acima pode ser tão precisa quanto se queira, bastando para isso as amostras X_i são independentes, as leis dos grandes números asseguram que a Dessa forma, a estimativa do valor esperado de $f(x)$ é uma média amostral. Quando

$$E[f(x)] = \frac{1}{n} \sum_{i=1}^n f(X_i)$$

Temos, então, uma aproximação ponderada discreta para a função densidade de probabilidade a posteriori $p(x_0, \dots, x_k / y_1, \dots, y_k)$. Os pesos são escolhidos usando-se o princípio da Amostragem por importância, ou *Importance Sampling*. Este princípio é baseado no seguinte: suponha que $p(x)$ seja uma função densidade de probabilidades da qual é difícil extrairmos amostras, mas que podemos calcular $c(x)$ tal que $p(x) \propto c(x)$. Sejam ainda $\{x(i), i = 1, 2, \dots, N\}$ pontos amostrados de uma função $q(x)$. Esta função é chamada densidade de importância (ou *Importance Density*), e é escolhida de forma que possamos amostrar facilmente a partir dela. Assim, uma aproximação ponderada para a função densidade de probabilidade $p(x)$ será dada por: [1]

$$p(x) = \sum_{i=1}^N w(i) \delta(x - x(i)) \tag{2.25}$$

, onde $w(i) \propto \frac{c(x(i))}{q(x(i))}$ é o peso normalizado da i -ésima amostra.

Portanto, se as amostras $x_0(i), \dots, x_k(i)$ forem extraídas de uma função densidade de importância $q(x_0, \dots, x_k / y_1, \dots, y_k)$, os pesos serão: [1]

$$w_k(i) \propto \frac{p(x_0(i), \dots, x_k(i) / y_1, \dots, y_k)}{q(x_0(i), \dots, x_k(i) / y_1, \dots, y_k)} \tag{2.26}$$

No caso sequencial, a cada iteração temos amostras que são uma aproximação para $p(x_0, \dots, x_{k-1} / y_1, \dots, y_{k-1})$, e queremos obter uma aproximação para $p(x_0, \dots, x_k / y_1, \dots, y_k)$ a partir de um novo conjunto de amostras. Se escolhermos a função densidade de importância de tal forma que: [1]

$$q(x_0, \dots, x_k / y_1, \dots, y_k) = q(x_k / x_0, \dots, x_{k-1} / y_1, \dots, y_{k-1}, x_0, \dots, x_{k-1} / y_1, \dots, y_{k-1}) \tag{2.27}$$

, então podemos obter amostras $x_0(i), \dots, x_k(i)$ de $q(x_0, \dots, x_k / y_1, \dots, y_k)$ acrescentando o novo estado $x_k(i)$ amostrado de $q(x_k / x_0, \dots, x_{k-1} / y_1, \dots, y_{k-1}, x_0, \dots, x_{k-1})$ às amostras $x_0(i), \dots, x_{k-1}(i)$ de $q(x_0, \dots, x_{k-1} / y_1, \dots, y_{k-1})$. Para determinarmos a equação de

$$w^k(t) \propto w^{k-1}(t) \frac{d(y^k/x^k(t))d(x^k(t)/x^{k-1}(t))}{d(y^k/x^k(t))d(x^k(t)/x^{k-1}(t))} b(x^k(t)/x^{k-1}(t), y^k) \quad (2.30)$$

Além disso, se $q(x^k/x_0, \dots, x_{k-1}, y_1, \dots, y_k) = q(x^k/x_{k-1}, y_1, \dots, y_k)$, a função densidade de importância dependerá apenas de x_{k-1} e y_k . Este fato é muito útil quando apenas uma estimativa filtrada de $p(x^k/y_1, \dots, y_k)$ é requerida a cada instante de tempo k , como assumiremos deste ponto em diante. Nestes casos, apenas o valor de $x^k(t)$ necessita ser guardado, e não precisamos guardar a trilha $x^0(t), \dots, x_{k-1}(t)$, nem o histórico de observações y_1, \dots, y_{k-1} . Assim, a nova equação de atualização dos pesos será: [1]

$$w^k(t) \propto w^{k-1}(t) \frac{d(y^k/x^k(t))d(x^k(t)/x^{k-1}(t))}{d(y^k/x^k(t))d(x^k(t)/x^{k-1}(t))} b(x^k(t)/x^{k-1}(t), y_1, \dots, y_k) = w^k(t) \frac{d(y^k/x^k(t))d(x^k(t)/x^{k-1}(t))}{d(y^k/x^k(t))d(x^k(t)/x^{k-1}(t))} b(x^k(t)/x^{k-1}(t), y_1, \dots, y_k) \quad (2.29)$$

Fazendo a substituição das equações (2.27) e (2.28) na equação (2.26), temos a equação de atualização dos pesos assumindo a seguinte forma:

$$d(x_0, \dots, x_{k-1}/y_1, \dots, y_k) \propto d(y^k/x^k(t))d(x^k(t)/x^{k-1}(t))d(x_0, \dots, x_{k-1}/y_1, \dots, y_{k-1}) \quad (2.28)$$

$$= \frac{d(y^k/x^k(t))d(x^k(t)/x^{k-1}(t))}{d(y^k/x^k(t))d(x^k(t)/x^{k-1}(t))} d(x_0, \dots, x_{k-1}/y_1, \dots, y_{k-1}) \propto \frac{d(y^k/x^k(t))d(x^k(t)/x^{k-1}(t))}{d(y^k/x^k(t))d(x^k(t)/x^{k-1}(t))} d(x_0, \dots, x_{k-1}/y_1, \dots, y_{k-1}) = \frac{d(y^k/x^k(t))d(x^k(t)/x^{k-1}(t))}{d(y^k/x^k(t))d(x^k(t)/x^{k-1}(t))} d(x_0, \dots, x_{k-1}/y_1, \dots, y_{k-1})$$

atualização dos pesos, $p(x_0, \dots, x_{k-1}/y_1, \dots, y_k)$ e $p(y^k/x^k)$ deve ser expresso em função de [1]:

Portanto, a função densidade de probabilidades filtrada $p(x_k / y_1, \dots, y_k)$ será aproximada de acordo com a equação abaixo: [1]

$$p(x_k / y_1, \dots, y_k) \approx \sum_{i=1}^N w_k^i(t) \delta(x_k - x_k^i(t)) \quad (2.31)$$

Os pesos $w_k^i(t)$ da equação acima são aqueles definidos na equação (2.30). Pode ser mostrado que, quando N tende ao infinito, a aproximação dada pela equação (2.31) é válida.

Assim, o algoritmo SIS realiza uma propagação recursiva de pesos e pontos auxiliares. Esta propagação é feita sequencialmente, conforme cada nova medida é obtida.

2.3.1.1. O PROBLEMA DA DEGENERACÃO

Um problema comum que ocorre com o filtro SIS é o fenômeno da degeneração. Este fenômeno ocorre quando todas as amostras exceto uma têm um peso muito pequeno após algumas iterações do algoritmo SIS. A degeneração faz com que um grande esforço computacional seja feito para atualizar amostras que contribuem muito pouco para a aproximação de $p(x_k / y_1, \dots, y_k)$. Uma medida da degeneração do algoritmo SIS é dada pelo tamanho amostral efetivo (N_{ef}), introduzido em [5] e [6]. O tamanho amostral efetivo é dado pela expressão abaixo:

$$N_{ef} = \frac{1 + Var(w_k^*(t))}{N} \quad (2.32)$$

onde $w_k^*(t) = \frac{p(x_k(t) / y_1, \dots, y_k)}{b(x_k(t) / x_{k-1}(t), y_k)}$ é chamado 'peso verdadeiro',

O 'peso verdadeiro' não pode ser calculado exatamente, mas uma estimativa de N_{ef} pode ser obtida do seguinte modo: [1]

Esta escolha é ótima pois, para um dado $x_{k-1}(t)$, $y_k(t)$ assume o mesmo valor qualquer que seja a amostra extraída de $q(x_k / x_{k-1}(t), y_k)_{\text{ótima}}$. Deste modo, condicionado em $x_{k-1}(t)$, $\text{Var}(w_k^*(t)) = 0$.

$$w_k(t) \propto w_{k-1}(t) \cdot p(y_k / x_{k-1}(t)) = w_{k-1}(t) \cdot \int p(y_k / x_k) p(x_k / x_{k-1}(t)) dx_k \quad (2.35)$$

Substituindo a equação (2.34) na equação (2.30), temos: [1]

$$q(x_k / x_{k-1}(t), y_k) = p(x_k / x_{k-1}(t), y_k) = \frac{p(y_k / x_k, x_{k-1}(t)) p(x_k / x_{k-1}(t))}{p(y_k / x_{k-1}(t))} \quad (2.34)$$

Este método consiste na escolha da função densidade de importância $q(x_k / x_{k-1}, y_k)$ de modo tal que a variância $\text{Var}(w_k^*(t))$ seja minimizada - assim, N_{ef} será maximizado. A escolha ótima da função densidade de importância, que minimiza esta variância, é a seguinte, conforme foi mostrado em [7]:

2.3.1.2. ESCOLHA ADEQUADA DA FUNÇÃO DENSIDADE DE IMPORTÂNCIA

Podem-se perceber que $N_{\text{ef}} \leq N$, e valores de N_{ef} pequenos indicam a ocorrência de degeneração severa. Sendo o fenômeno da degeneração indesejável, devemos tentar reduzi-lo nos filtros de partículas. O primeiro modo de reduzirmos este efeito é escolher um valor elevado para N , o que é, em muitos casos, impraticável. O segundo modo consiste numa escolha adequada da função densidade de importância, e o terceiro modo consiste no uso da reamostragem (*resampling*). Os dois últimos métodos serão descritos a seguir.

, onde $w_k(t)$ é o peso normalizado obtido usando-se a equação (2.29).

$$N_{\text{ef}} = \frac{1}{\sum_{i=1}^N (w_k^i(t))^2} \quad (2.33)$$

Há 2 problemas na escolha ótima da função densidade de importância. O primeiro é que se torna necessário extrair amostras de $p(x_k / x_{k-1}(t), y_k)$, além de termos que calcular integrais usando-se o estado x_k . Em geral, fazer estas 2 coisas é complicado, mas há 2 casos nos quais o uso da função densidade de importância ótima é fátível. [1]

O primeiro caso é aquele em que x_k faz parte de um conjunto finito de estados. Neste caso, a integral da equação (2.35) torna-se uma soma e a amostragem a partir de $p(x_k / x_{k-1}(t), y_k)$ é possível. [1]

O segundo caso é aquele em que a função $p(x_k / x_{k-1}(t), y_k)$ é Gaussiana - assim, o cálculo analítico da equação (2.35) torna-se possível. Isso pode ocorrer se a dinâmica do sistema for não-linear e a equação de medidas do sistema for linear. Estas equações seriam dadas por: [1]

$$x_k = f_k(x_{k-1}) + w_{k-1}$$

(2.36)

$$y_k = C_k x_k + e_k$$

onde w_{k-1} é um ruído gaussiano com média nula e matriz de covariância Q_{k-1} , e e_k é um ruído gaussiano com média nula e matriz de covariância R_k . Temos ainda que f_k é uma função não-linear, C_k é a matriz de observação, e w_{k-1} e e_k são seqüências gaussianas independentes e identicamente distribuídas, com Q_{k-1} e R_k positivas definidas. Definindo: [1]

$$\begin{aligned} \Gamma_k^{-1} &= \bar{Q}_{k-1}^{-1} + C_k^T R_k^{-1} C_k \\ m_k &= \Gamma_k^{-1} (\bar{Q}_{k-1}^{-1} f_k(x_{k-1}) + C_k^T R_k^{-1} y_k) \end{aligned}$$

(2.37)

Temos que $p(x_k / x_{k-1}, y_k)$ será gaussiana com média m_k e matriz de covariância Γ_k , e $p(y_k / x_{k-1})$ será gaussiana com média $C_k f_k(x_{k-1})$ e matriz de covariância $Q_{k-1} + C_k R_k C_k^T$. [1]

Para muitos modelos, no entanto, estes cálculos analíticos não são possíveis. Podemos, ainda assim, calcular aproximações sub-ótimas para a função densidade de importância utilizando técnicas de linearização local [7]. Estas técnicas utilizam uma função densidade de importância que é uma aproximação gaussiana da função densidade de probabilidade $p(x_k / x_{k-1}, y_k)$.

Há também a possibilidade de escolhermos a função densidade de importância como sendo a função densidade de probabilidade a priori: [1]

$$q(x_k / x_{k-1}(t), y_k) = p(x_k / x_{k-1}(t)) \tag{2.38}$$

Substituindo a equação (2.38) na equação (2.30), temos:

$$w_k(t) \propto w_{k-1}(t) p(y_k / x_k(t)) \tag{2.39}$$

Esta é a escolha mais simples para a função densidade de importância, pois é facilmente implementável. [1]

2.3.1.3. REAMOSTRAGEM (RESAMPLING)

O segundo método usado para reduzir os efeitos da degeneração é a reamostragem. Podemos utilizá-la quando o tamanho amostral efetivo (N_{ef}) for menor que um certo limiar (N_T), o que é um indicador da ocorrência de degeneração. A ideia central da reamostragem é eliminar amostras que têm um peso muito pequeno, mantendo apenas as amostras que têm pesos grandes. Na reamostragem é gerado um novo conjunto $\{x_k^*(i), i = 1, 2, \dots, N\}$ através de uma amostragem (com reposição) realizada a partir da aproximação discreta de $p(x_k / y_1, \dots, y_k)$, repetida N vezes, e tal que $P(x_k^*(i) = x_k(i)) = w_k(i)$. Esta aproximação discreta é a seguinte: [1]

$$p(x_k / y_1, \dots, y_k) \approx \sum_{i=1}^N w_k(i) \delta(x_k - x_k^*(i)) \tag{2.40}$$

O resultado será uma amostra da função densidade de probabilidade discreta dada por (2.40) e, feito isso, os pesos são ajustados para valerem $1/N$.

Embora a reamostragem reduza os efeitos da degeneração, ela introduz outros problemas no filtro amostral. O principal problema é que as amostras com pesos maiores serão selecionadas estatisticamente muitas vezes. Isso leva a uma perda de diversidade amostral, já que a amostra resultante conterá muitos pontos repetidos. Este problema é conhecido como "empobrecimento amostral" (ou *sample impoverishment*), e é grave nos processos com baixo nível de ruído. Para casos em que o processo apresenta um nível de ruído muito baixo, todas as amostras convergirão para um único ponto após algumas iterações do algoritmo SIS. Pode-se evitar o empobrecimento amostral utilizando-se o método MCMC (*Markov Chain Monte Carlo*) [8], que será descrito mais adiante.

O algoritmo SIS constitui a base para a maioria dos filtros de partículas desenvolvidos recentemente. Os outros tipos de filtros de partículas apresentados na literatura podem ser entendidos como casos especiais do algoritmo genérico SIS. Estes outros filtros podem ser derivados a partir do filtro SIS escolhendo-se uma função densidade de importância adequada ou modificando-se a etapa de reamostragem. Apresentaremos, a seguir, outros 3 tipos de filtros de partículas: o filtro Sampling Importance Resampling, o filtro Auxiliary Sampling Importance Resampling e o filtro Metropolis Hastings Importance Resampling.

2.3.2. O FILTRO SAMPLING IMPORTANCE RESAMPLING (SIR)

O filtro Sampling Importance Resampling (SIR) [9] utiliza como base o método de Monte Carlo e pode ser aplicado a problemas de filtragem Bayesiana recursiva. Para utilizá-lo, devemos conhecer as funções f_k e g_k que descrevem a dinâmica do sistema considerado. Além disso, é necessário extrair amostras da função distribuição de probabilidades do ruído, w^{k-1} , e da função distribuição de probabilidades $p(x_k / x_{k-1})$. A função $p(y_k / x_k)$ também precisa ser conhecida, a menos de uma constante de proporcionalidade. O algoritmo SIR pode ser derivado a partir do algoritmo SIS

através de uma escolha adequada da função densidade de importância: $q(x_k / x_{k-1}(t), y_1, \dots, y_k) = p(x_k / x_{k-1}(t))$. Para completar, a reamostragem deve ser feita a cada iteração do algoritmo.

A escolha da função densidade de importância indicada acima requer a extração de amostras de $p(x_k / x_{k-1}(t))$. Uma amostra $x_k(t)$ extraída de $p(x_k / x_{k-1}(t))$ pode ser gerada do seguinte modo: primeiramente, extraímos uma amostra $w_{k-1}(t)$ da função de probabilidade de importância $p_{w_{k-1}}(w_{k-1})$; a seguir, fazemos $x_k(t) = f_k(x_{k-1}(t), w_{k-1}(t))$. Para esta escolha da função densidade de importância, os pesos associados às amostras serão dados por: [1]

$$w_k(t) \propto w_{k-1}(t) \cdot p(y_k / x_k(t)) \tag{2.41}$$

Como a reamostragem deve ser feita a cada iteração do algoritmo, temos que $w_{k-1}(t) = 1 / N$, $\forall i$, e deste modo:

$$w_k(t) \propto p(y_k / x_k(t)) \tag{2.42}$$

Os pesos dados pela equação acima devem ser normalizados antes da reamostragem. O principal problema apresentado pelo filtro SIR deve-se ao fato de a função densidade de importância ser independente da observação y_k . Assim, o espaço de estados é explorado sem o conhecimento das observações, e o filtro poderá ser ineficiente. Outrossim, como a reamostragem é feita a cada iteração do algoritmo, podemos ter uma rápida perda de diversidade amostral. Entretanto, o filtro SIR tem as vantagens de os pesos das amostras poderem ser facilmente calculados e de podermos amostrar com facilidade a partir da função densidade de importância. [1]

2.3.3. O FILTRO AUXILIARY SAMPLING IMPORTANCE RESAMPLING (ASIR)

O filtro Auxiliary Sampling Importance Resampling [10] é uma variação do filtro SIR. Ele pode ser obtido a partir do filtro genérico SIS através da escolha da função

densidade de importância $q(x_k, i / y_1, \dots, y_k, i(t), j = 1, \dots, N\}$, onde $i(t)$ é o índice da amostra no instante de tempo $k-1$.

Aplicando-se a regra de Bayes, pode-se obter o seguinte resultado: [1]

$$\begin{aligned}
 p(x_k, i / y_1, \dots, y_k) &\propto p(y_k / x_k) p(x_k, i / y_1, \dots, y_{k-1}) = \\
 &= p(y_k / x_k) p(x_k, i / y_1, \dots, y_{k-1}) p(i / y_1, \dots, y_{k-1}) = \\
 &= p(y_k / x_k) p(x_k) p(i / y_1, \dots, y_{k-1})
 \end{aligned}
 \tag{2.43}$$

O filtro ASIR obtém primeiramente uma amostra da função densidade de probabilidade conjunta $p(x_k, i / y_1, \dots, y_k)$, e então omite os índices i de (x_k, i) para gerar uma amostra $\{x_k(i), j = 1, \dots, N\}$ da função densidade de probabilidade marginal $p(x_k / y_1, \dots, y_k)$. A função densidade de importância utilizada para extrair a amostra $\{x_k(i), i(t), j = 1, \dots, N\}$ é definida de modo a satisfazer a equação abaixo: [1]

$$q(x_k, i / y_1, \dots, y_k) \propto p(y_k / H_k(i)) p(x_k / x_{k-1}(i)) w_{k-1}(i)
 \tag{2.44}$$

$\mu_k(i)$ é uma caracterização de x_k dado $x_{k-1}(i)$. Ela pode, portanto, ser a média: $\mu_k(i) = E(x_k / x_{k-1}(i))$, ou ainda uma amostra $\mu_k(i)$ extraída de $p(x_k / x_{k-1}(i))$. Fazendo:

$$q(x_k, i / y_1, \dots, y_k) = q(i / y_1, \dots, y_k) q(x_k / i, y_1, \dots, y_k)
 \tag{2.45}$$

E definindo:

$$q(x_k / i, y_1, \dots, y_k) = \underset{\Delta}{p}(x_k / x_{k-1}(i))
 \tag{2.46}$$

Utilizando as equações (2.44), (2.45) e (2.46), chegamos a: [1]

$$q(i / y_1, \dots, y_k) \propto p(y_k / H_k(i)) w_{k-1}(i)
 \tag{2.47}$$

A amostra $\{x_k(j), t(j), j = 1, \dots, N\}$ recebe então um peso proporcional à relação (2.43) / (2.44): [1]

$$w_k(j) \propto w_{k-1}(j) \frac{p(y_k/x_k(j), t(j) | y_1, \dots, y_k) q(x_k(j), t(j) | y_1, \dots, y_k)}{p(y_k/x_k(j)) p(x_k(j), t(j) | x_{k-1}(j))} = \frac{p(y_k/x_k(j), t(j) | H_k(j))}{p(y_k/x_k(j))} \quad (2.48)$$

Comparado ao filtro SIR, o filtro ASIR tem a vantagem de gerar pontos a partir da amostra no instante k-1 que estarão próximos do estado verdadeiro do sistema com maior probabilidade. O filtro ASIR pode ser entendido como o algoritmo SIR com uma reamostragem no instante de tempo anterior, baseada em algumas estimativas $(\mu^k(i))$ que caracterizam a função $p(x_k / x_{k-1}(t))$. Se o nível de ruído do processo for baixo, e desse modo $p(x_k / x_{k-1}(t))$ for bem definida por $\mu^k(i)$, o filtro ASIR funcionará melhor que o filtro SIR. Contudo, se o nível de ruído do processo for alto, $\mu^k(i)$ não caracterizará a função $p(x_k / x_{k-1}(t))$ de modo adequado; assim, o filtro ASIR fará a reamostragem baseada numa aproximação ruim de $p(x_k / x_{k-1}(t))$. Neste caso, o uso do algoritmo ASIR não é recomendável. [1]

2.3.4. O FILTRO METROPOLIS HASTINGS IMPORTANCE RESAMPLING (MHIR)

Antes de apresentarmos o filtro MHIR, que é um filtro baseado no método *Markov Chain Monte Carlo* (MCMC), faremos uma breve introdução do método MCMC.

2.3.4.1. O MÉTODO MARKOV CHAIN MONTE CARLO (MCMC)

INTEGRAÇÃO MONTE CARLO

A integração através do método de Monte Carlo visa calcular o valor de $E[f(x)]$ extraindo-se amostras de $p(x)$ - $\{x_k, k = 1, \dots, n\}$ - e realizando a seguinte aproximação: [8]

Um aspecto importante das cadeias de Markov é a influência que o estado x_0 tem sobre o estado x_k . Esta influência é caracterizada pela função densidade de probabilidades de x_k dado x_0 , ou seja, $P(x_k / x_0)$. Como não são dadas as variáveis $\{x_1, x_2, \dots, x_{k-1}\}$, x_k depende diretamente de x_0 . Sujeita a condições normais, a cadeia irá 'esquecer' gradativamente o seu estado inicial e, desta forma, $P(\cdot / x_0)$ convergirá para uma distribuição estacionária, que não dependerá de k nem de x_0 . Esta distribuição estacionária será chamada $\phi(\cdot)$. Assim, conforme k é incrementado, os

quais $P(\cdot / \cdot)$ não depende de k . [8]

Consideraremos aqui cadeias de Markov homogêneas no tempo, ou seja, cadeias nas Markov, e $P(\cdot / \cdot)$ é chamada de núcleo de transição (ou *transition kernel*) da cadeia. do histórico da cadeia $\{x_0, x_1, \dots, x_{k-1}\}$. Esta sequência é denominada cadeia de cadeia, ou seja, x_k . Em outras palavras, dado x_k , o próximo estado x_{k+1} não dependerá distribuição de probabilidades $P(x_{k+1} / x_k)$ que depende apenas do estado presente da a cada instante de tempo o estado seguinte x_{k+1} é amostrado a partir de uma função Suponha que tenhamos uma sequência de variáveis aleatórias, $\{x_0, x_1, \dots, x_n\}$, tal que

CADEIAS DE MARKOV

estacionária $p(x)$. [8]

a função $p(x)$ - por exemplo, uma cadeia de Markov que tenha distribuição ser independentes. $\{x_k\}$ pode ser gerado por qualquer processo que tenha como base pode ser uma função bastante complicada. No entanto, as amostras $\{x_k\}$ não precisam Em geral, extrair amostras independentes de $p(x)$ não é possível, uma vez que $p(x)$

para isso aumentarmos o tamanho da amostra (n). [8]

garantem que a aproximação acima pode ser tão precisa quanto se queira, bastando simples. Quando as amostras $\{x_k\}$ são independentes, as leis dos grandes números Deste modo, a média populacional de $f(x)$ é calculada como uma média amostral

$$E[f(x)] \approx \frac{1}{n} \sum_{k=1}^n f(x_k)$$

pontos amostrados $\{x_k\}$ serão cada vez mais similares a amostras dependentes de $\phi(\cdot)$. [8]

Deste modo, após um período inicial suficientemente longo (composto por m iterações), os pontos $\{x_k; k = m+1, \dots, n\}$ serão amostras dependentes de $\phi(\cdot)$. Podemos então utilizar a saída desta cadeia de Markov para estimar a expectativa $E[f(x)]$, onde x tem distribuição $\phi(\cdot)$. As amostras extraídas durante o período inicial não são utilizadas neste cálculo, de modo que o estimador obtido é o seguinte: [8]

$$\bar{f} = \frac{1}{n-m} \sum_{k=m+1}^n f(x_k) \quad (2.49)$$

Esta média é conhecida como média ergódica. O teorema ergódico garante a convergência desta média para a média requerida. [3]

O ALGORITMO METROPOLIS HASTINGS

A equação (2.49) mostra como uma cadeia de Markov pode ser utilizada para estimar $E[f(x)]$, onde a expectativa é tirada sobre sua distribuição estacionária, $\phi(\cdot)$. Entretanto, ainda precisamos descobrir como construir uma cadeia de Markov tal que sua distribuição estacionária $\phi(\cdot)$ seja a distribuição de interesse, $p(\cdot)$.

Para construir tal cadeia de Markov, utilizamos o algoritmo Metropolis - Hastings, inicialmente proposto por Metropolis [8]. No algoritmo Metropolis - Hastings, o estado seguinte x_{k+1} da cadeia de Markov é escolhido, a cada instante de tempo, do seguinte modo: primeiramente, amostramos um ponto candidato y de uma distribuição proposta $q(\cdot / x_k)$. A distribuição proposta pode ser, por exemplo, uma gaussiana multivariável de média x e matriz de covariância fixa. O ponto candidato y é então aceito com probabilidade $\alpha(x_k, y)$, onde: [8]

$$\alpha(x, y) = \min \left(1, \frac{p(y)q(x/y)}{p(x)q(y/x)} \right) \quad (2.50)$$

Se o ponto candidato for aceito, o estado seguinte passa a valer $x_{k+1} = y$. Caso o ponto candidato seja rejeitado, teremos $x_{k+1} = x_k$.

Dessa forma, o algoritmo Metropolis - Hastings pode ser escrito do seguinte modo

[8]:

Inicialização : $x_0 = \text{valor}$, $k = 0$,

Laço principal : for $k = 1 : \text{num}$,

{ Amostre um ponto y a partir de $q(\cdot / x_k)$;

Amostre uma variável aleatória u com distribuição uniforme no intervalo $[0, 1]$;

if ($u \leq \alpha(x_k, y)$) $x_{k+1} = y$;

else $x_{k+1} = x_k$;

$k = k + 1$;

}

A distribuição estacionária da cadeia de Markov gerada pelo algoritmo acima será $p(x)$, qualquer que seja a distribuição proposta $q(\cdot / \cdot)$. Isso pode ser visto usando-se o seguinte argumento: o núcleo de transição para o algoritmo Metropolis - Hastings é dado por: [8]

$$P(x_{k+1} / x_k) = q(x_{k+1} / x_k) \alpha(x_k, x_{k+1}) + I(x_{k+1} = x_k) \cdot [1 - \int q(y / x_k) \alpha(x_k, y) dy] \quad (2.51)$$

$I(\cdot)$ representa a função indicadora (se o argumento da função for verdadeiro, $I(\cdot) = 1$; caso contrário, $I(\cdot) = 0$). O primeiro termo em (2.51) resulta da aceitação do candidato $y = x_{k+1}$, enquanto o segundo termo resulta da rejeição de y , para todos os possíveis candidatos y . Utilizando o fato de que: [8]

$$p(x) q(x_{k+1} / x_k) \alpha(x_k, x_{k+1}) = p(x_{k+1}) q(x_k / x_{k+1}) \alpha(x_{k+1}, x_k) \quad (2.52)$$

, que se obtém a partir de (2.50), obtemos a equação de balanço:

Outra questão básica na implementação do algoritmo Metropolis - Hastings é como determinar o período inicial (ou *burn-in*) da cadeia de Markov (ou seja, determinar m , o número de iterações iniciais da cadeia que não serão usadas nas estimativas) e o seu ponto de parada (ou seja, o tamanho total da amostra, n). Um dos modos de determinar m (sugerido em [11]) é ajustar seu valor entre 1% e 2% do tamanho total da amostra (n). Para determinar n , um dos métodos consiste em gerar várias cadeias de Markov simultaneamente (cada uma tendo partido de um valor inicial diferente), e posteriormente comparar as estimativas de f (equação (2.49)) fornecidas por estas

Uma questão importante no que tange à implementação do algoritmo Metropolis - Hastings é como escolher a distribuição proposta $q(\cdot / \cdot)$. Um conjunto de formas canônicas possíveis para $q(\cdot / \cdot)$ é aquele em que a distribuição proposta é simétrica, ou seja, $q(x / y) = q(y / x)$ (algoritmo Metropolis). Se x for contínuo, $q(\cdot / x)$ pode ser uma gaussiana multivariável de média x e matriz de covariância constante, por exemplo. Um caso particular do algoritmo Metropolis é o algoritmo Metropolis do caminho aleatório (ou *random walk Metropolis*), no qual temos $q(y / x) = q(|x - y|)$. Um outro conjunto de formas canônicas possíveis para $q(\cdot / \cdot)$ é dado pelo amostrador independente (ou *independence sampler* [8]), no qual $q(y / x) = q(y)$.

O lado esquerdo da equação (2.54) é a distribuição marginal de x_{k+1} , assumindo que x_k foi extraído de $p(x)$. Então, a equação (2.54) sugere que se x_k for extraído da distribuição estacionária foi obtida, todas as amostras subsequentes pertencerão a esta distribuição. Isto prova que a distribuição estacionária da cadeia de Markov é $p(x)$. Restaria provar que $P(x_k / x_0)$ converge para a distribuição estacionária, o que não será feito aqui. Esta prova está detalhada em [8].

$$\int p(x_k) P(x_{k+1} / x_k) dx_k = \int p(x_{k+1}) P(x_k / x_{k+1}) dx_k \quad (2.54)$$

Integrando sobre x_k ambos os lados da equação (2.53), obtemos: [8]

$$p(x_k) P(x_{k+1} / x_k) = p(x_{k+1}) P(x_k / x_{k+1}) \quad (2.53)$$

cadeias. Se os valores de f estiverem relativamente discrepantes, n deve ser aumentado.

2.3.4.2. O FILTRO MHIR

O filtro MHIR [12] é um filtro de partículas que utiliza o algoritmo Metropolis - Hastings para gerar amostras dos estados do sistema ($x_k(i)$). Assume-se que no instante $k-1$ temos um conjunto de N partículas $\{x_{k-1}(1), x_{k-1}(2), \dots, x_{k-1}(N)\}$ que seguem a distribuição $p(x_{k-1} / y_1, \dots, y_{k-1})$. Para gerar uma amostra aproximada da função densidade de probabilidade $p(x_k / y_1, \dots, y_k)$, o seguinte algoritmo é utilizado:

[12]

Primeiro passo - Amostar um índice uniformemente a partir do conjunto $\{1, \dots, N\}$. Este índice será denominado i_j .

Lago principal - While ($j > N$),

{
 Amostar $x_k(i)$ a partir de uma iteração do algoritmo Metropolis - Hastings (a distribuição estacionária da cadeia de Markov deve ser $p(x_k / x_{k-1}(i_j), y_k)$);
 $j = j + 1$;
 Amostar um índice (i) uniformemente a partir do conjunto $\{1, \dots, N\}$;
 Amostar uma variável aleatória (u) com distribuição uniforme no intervalo

$$P_a = \min \left[1, \frac{p(x_k(i) / x_{k-1}(i), y_k)}{p(x_k(i_{j-1}) / x_{k-1}(i_{j-1}), y_k)} \right];$$

if ($u \leq P_a$) $i_j = i$;

else $i_j = i_{j-1}$;

end

}

A cada iteração do algoritmo, um par $(i_j, x_k(i_j))$ é gerado. Quando j tende ao infinito, a função densidade de probabilidade que rege o comportamento dos estados $x_k(i)$ aproxima-se do valor real de $p(x_k / y_1, \dots, y_k)$. Para gerar uma amostra aproximada

Contudo, o filtro MHIR pode ser muito eficiente nos casos em que a função de probabilidade de probabilidade $p(y_k / x_k)$ predomina sobre a função de probabilidade $p(x_k / y_k, \dots, y_k)$, a cada instante de tempo. É igualmente possível utilizar o filtro MHIR para combater um problema que ocorre com o filtro SIR, quando não se consegue uma boa aproximação para a função densidade de

$$p(x_k(t) / x_{k-1}(t), y_k) \text{ é mais alta [12].}$$

- Um agravante deste problema é que o filtro MHIR propõe novos índices de modo uniforme - um método mais adequado seria fazer com que fossem amostrados com maior frequência os índices i para os quais a probabilidade $p(x_k(t) / x_{k-1}(t), y_k)$ provavelmente terá um valor baixo para a maioria dos índices a . Como consequência, a probabilidade de aceitação P_a no algoritmo MHIR será baixa, e a cadeia de Markov passará longos períodos sem que o valor do seu índice i_j seja alterado.
- Se o nível de ruído do processo for baixo com relação à variância na amostra inicial, a cadeia de Markov irá demorar a explorar o conjunto de índices $\{1, \dots, N\}$. Assim, se os valores atuais do índice e do estado forem i e $x_k(t)$, respectivamente, e se $x_k(t)$ for uma amostra da função densidade de probabilidade $p(x_k / x_{k-1}(t), y_k)$, então a relação $p(x_k(t) / x_{k-1}(a), y_k) / p(x_k(t) / x_{k-1}(t), y_k)$ provavelmente terá um valor baixo para a maioria dos índices a . Como consequência, a probabilidade de aceitação P_a no algoritmo MHIR será baixa, e a cadeia de Markov passará longos períodos sem que o valor do seu índice i_j seja alterado.

seguintes fatos: [12]

Alguns problemas podem ocorrer com o filtro MHIR se o nível de ruído do processo for baixo, visto que a cadeia de Markov pode levar muito tempo para convergir. Assim, o período inicial (ou *burn-in*) da cadeia deverá ser muito longo, e teremos que utilizar um valor muito elevado para m . Estes problemas são devidos aos

a partir desta função densidade de probabilidade, os valores iniciais de $x_k(t)$ não devem ser utilizados (pois eles são gerados no período inicial, ou *burn-in*, da cadeia de Markov). Desta forma, os m valores iniciais de $x_k(t)$ não deverão ser utilizados nas estimativas (para um valor adequado de m , conforme visto na seção 2.3.4.1.) - os valores a serem utilizados serão os N seguintes. Estes valores constituirão a amostra aproximada da função densidade de probabilidade requerida. [12]

probabilidade $p(x_k / y_1, \dots, y_k)$ devido a erros diversos. Neste caso, ao invés de amostrar a partir da aproximação de $p(x_k / y_1, \dots, y_k)$ no instante k dada a aproximação de $p(x_{k-1} / y_1, \dots, y_{k-1})$ no instante $k-1$, o filtro MHIR poderia amostrar a partir de uma aproximação desta função densidade de probabilidade num instante de tempo posterior, τ . Isto foi proposto baseado no fato de que o efeito que o erro da aproximação de $p(x_{k-1} / y_1, \dots, y_{k-1})$ tem sobre o erro da aproximação de $p(x_\tau / y_1, \dots, y_\tau)$ diminui conforme τ aumenta [12].

2.4. EXEMPLO DE APLICAÇÃO

Para ilustrar o funcionamento do filtro de Kalman e dos filtros de partículas, apresentaremos o seguinte exemplo. Seja o sistema unidimensional abaixo:

$$x_{k+1} = \cos(x_k) + 3 \cdot n_k + 5 \cdot w_k$$

$$y_k = 2 \cdot x_k^2 + e_k$$

Neste sistema, a entrada n_k é um degrau unitário, w_k é um ruído aleatório uniformemente distribuído entre 0 e 0.1, e e_k é um ruído aleatório de medição com distribuição normal (média nula e covariância unitária). O período de amostragem utilizado neste sistema discreto foi de 0.1s, e foram utilizadas condições iniciais nulas (ou seja, $x_0 = 0$). Este sistema foi simulado no MATLAB, e foram coletados os valores das saídas y_k para posterior utilização nos filtros. As equações do filtro de Kalman entendido para este sistema podem ser escritas do seguinte modo, a cada iteração:

$$P_0 = I; x_0 = 0$$

$$A = -\sin(x_{k-1})$$

$$H = 4 \cdot x_{k-1}$$

$$K(i) = (A P(i) H) \cdot (H P(i) H + I)^{-1}$$

$$P(i+1) = (A - K(i)) \cdot P(i) \cdot (A - K(i)) \cdot H + 25 + K^2(i)$$

$$x_k = \cos(x_{k-1}) + 3 + K \cdot (y_k - 2 \cdot x_{k-1}^2)$$

A primeira etapa no conjunto de equações acima é a inicialização do filtro; após isso, temos as 2 equações de linearização das matrizes A e H . A seguir, temos a equação para o cálculo do ganho K e a equação de Riccati. Finalmente, a última equação representa a atualização recursiva dos estados x_k utilizando o ganho K calculado previamente e as medições ruidosas disponíveis, y_k .

Abaixo, temos a trajetória de medições y_k utilizada nos filtros:

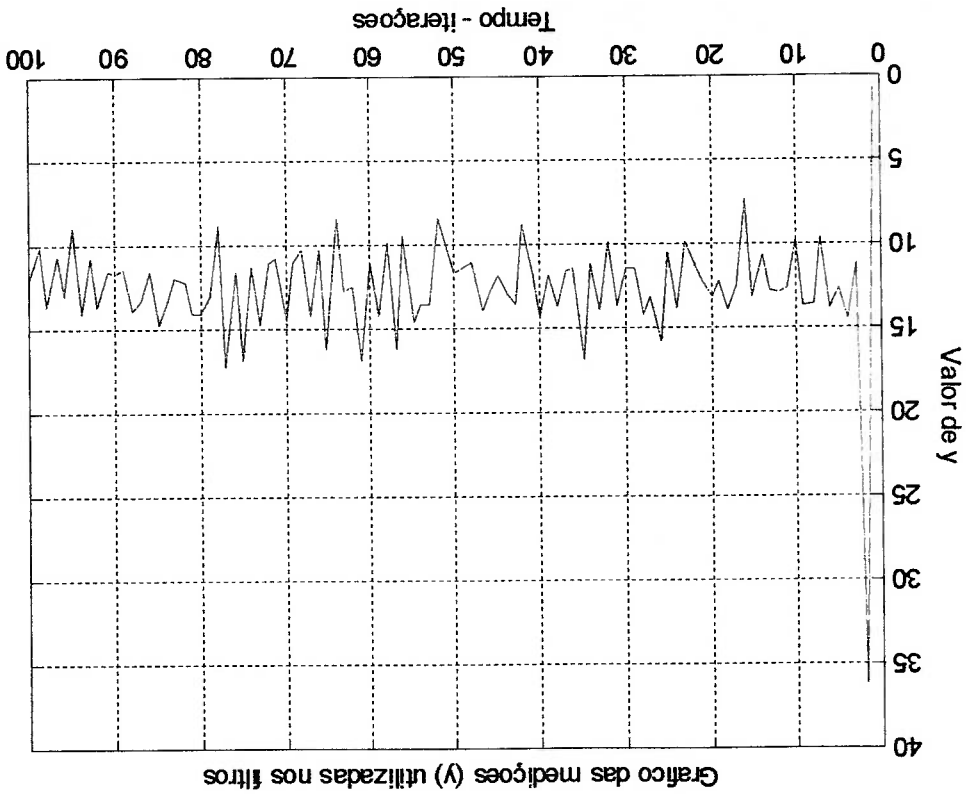


Figura 2.1 – Gráfico das medições utilizadas para a estimação de estados nos filtros

O ruído do processo, w_k , é um ruído não - gaussiano, conforme podemos observar na figura 2.2. Sua distribuição de probabilidades é uniforme, como podemos verificar através do histograma. Já o ruído de medição (e_k) é um ruído gaussiano, como podemos inferir a partir do histograma da figura 2.3. Simulando o filtro de Kalman estendido no MATLAB, obtivemos a trajetória de estados dada na figura 2.4 (a trajetória de estados estimada é a contínua, a trajetória de estados real é a tracejada).

Figura 2.3 – Histograma do ruído de medição para o sistema dado no exemplo

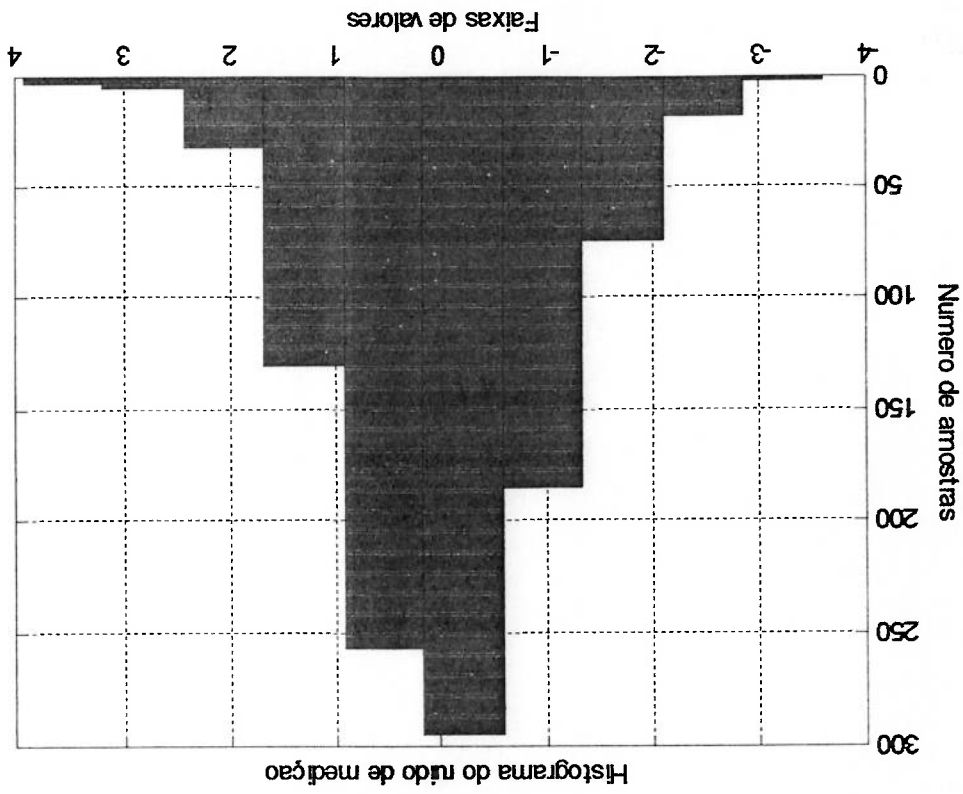
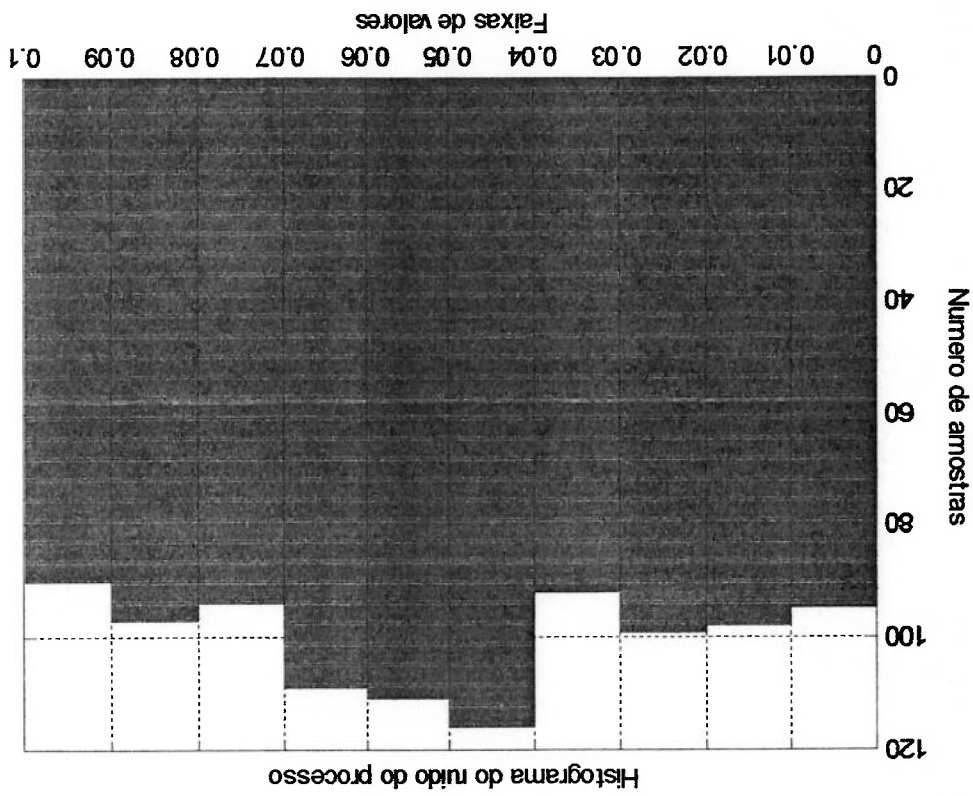


Figura 2.2 – Histograma do ruído do processo para o sistema dado no exemplo



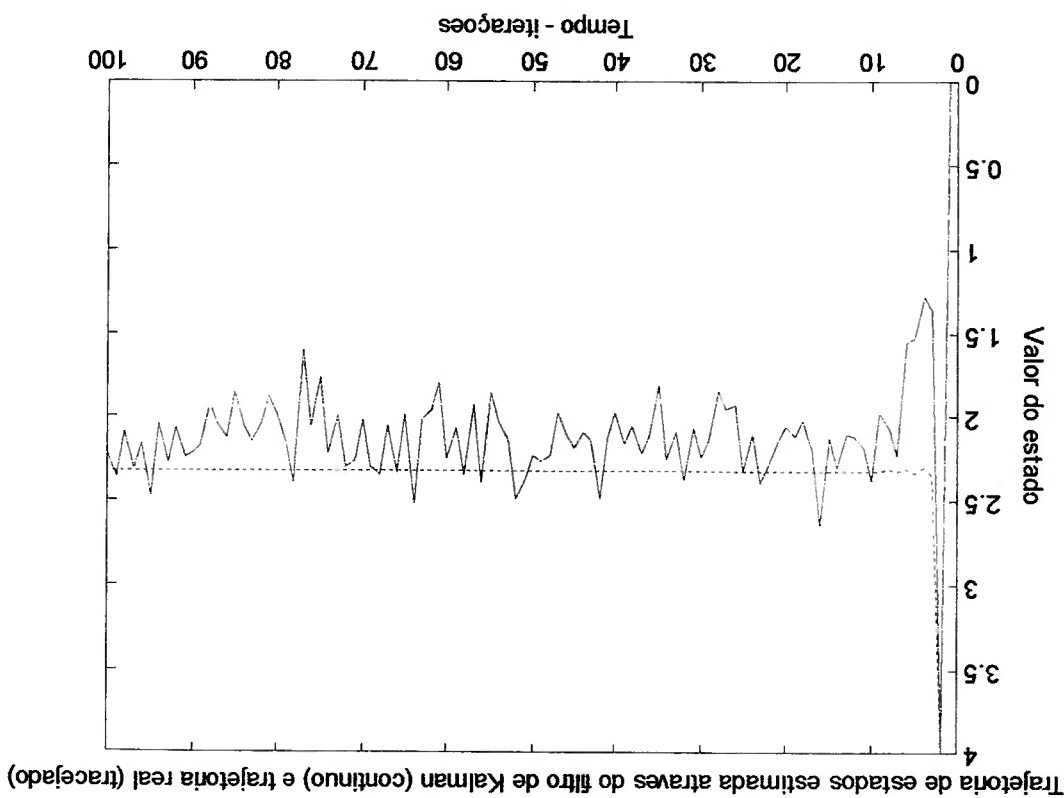


Figura 2.4 – Gráfico da trajetória de estados estimada pelo filtro de Kalman

Observando o gráfico acima, percebe-se que as trajetórias real e estimada estão relativamente próximas. Assim, conclui-se que o filtro de Kalman apresentou um desempenho satisfatório para este sistema, que é não-linear e apresenta ruído não-gaussiano. Contudo, o filtro de partículas SIR apresenta um ganho em relação ao filtro de Kalman estendido quanto ao erro de estimação para este sistema (devido ao fato de que os filtros de partículas são mais adequados para estimarmos estados de sistemas não-lineares e com ruídos não-gaussianos, conforme visto anteriormente). O algoritmo para sua execução é dado abaixo:

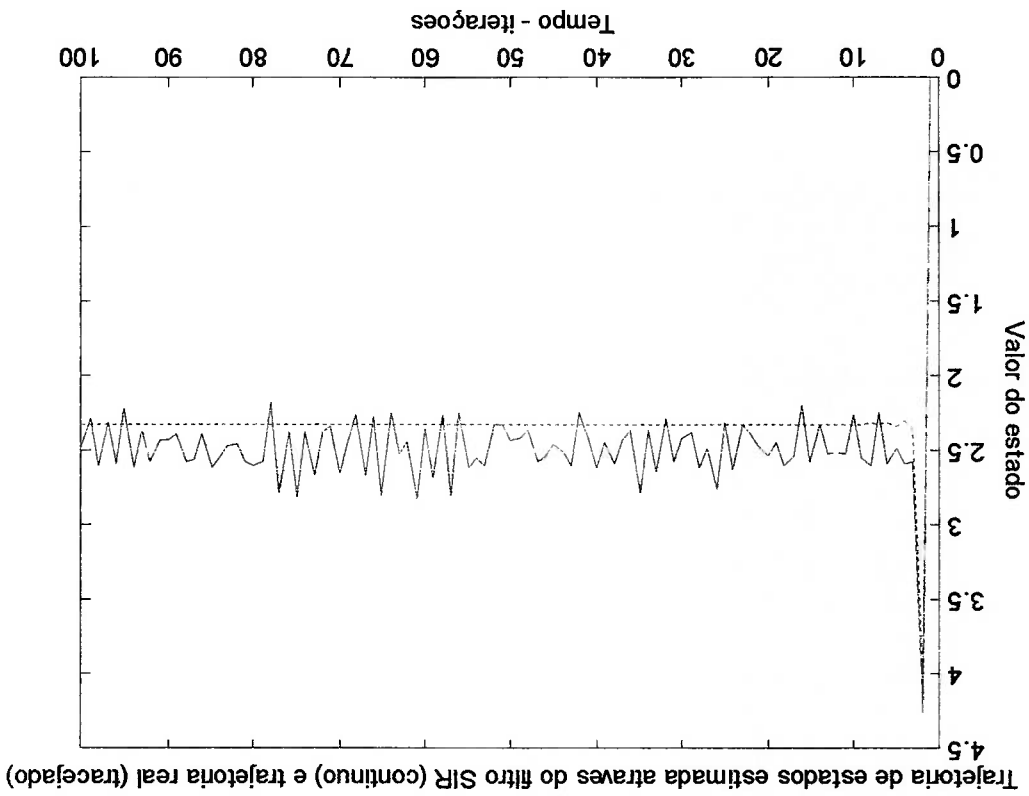
- Inicialização: $x_0(i) = 0, i = 1, 2, \dots, N$ ($N = 100$ amostras)
- Geração das amostras e seus pesos associados:

```

For (i = 1:N)
{
    wr(i) = 0.1 rand ; (amostragem do ruído)
     $x_k(i) = \cos(x_{k-1}(i)) + 3 + 5 \cdot wr(i)$  ; (geração das amostras do estado x)
     $peso_k(i) = \exp(-0.5 (y_k - 2 x_k(i))^2) / \sqrt{2 \cdot \pi}$  (cálculo dos pesos)
}
    
```

Pode-se observar que a trajetória de estados do sistema é estimada com eficácia pelo filtro SIR, apresentando um erro de estimação relativamente baixo.

Figura 2.5 – Gráfico da trajetória de estados estimada pelo filtro de partículas SIR



Executando este algoritmo no MATLAB, pode-se estimar a trajetória de estados x_k do sistema dado no exemplo supracitado. Abaixo temos um gráfico no qual estão plotadas a trajetória de estados estimada através do filtro SIR e a trajetória de estados real do sistema não-linear e não-gaussiano:

- Normalização dos pesos: $peso_k(t) = \frac{\sum_{i=1}^N peso_k(t)}{peso_k(t)}$
- Execução da reamostragem;
- Cálculo das estimativas do estado $x_k = \sum_{i=1}^N peso_k(t) \cdot x_k^i(t)$

A fim de compararmos os erros de estimação do filtro de Kalman estendido e do filtro de partículas SIR, utilizaremos o seguinte erro médio quadrático como medida de desempenho:

$$Erro = \sum_{k=1}^N (x_k - \hat{x}_k)^2$$

x_k é o estado verdadeiro do sistema, e \hat{x}_k é o estado estimado pelos filtros. Calculando os erros dos filtros com base nesta medida, temos o seguinte:

$$Erro_{kalman} = 60.15$$

$$Erro_{SIR} = 47.64$$

Assim, o erro do filtro de partículas SIR é menor que o erro do filtro de Kalman estendido, o que comprova o exposto anteriormente, ou seja, para sistemas não-lineares e com ruídos não-gaussianos o filtro de Kalman pode não aproximar os estados do sistema tão bem quanto os filtros de partículas (quando isso ocorre, o filtro de Kalman realiza seus cálculos com base em linearizações das funções não-lineares e aproximações gaussianas de ruídos não-gaussianos, o que em alguns casos pode não representar bem o comportamento do sistema). Já o filtro de partículas SIR, por trabalhar diretamente com o ruído real do processo e gerar amostras do estado a partir da equação não-linear original do sistema, apresenta um desempenho superior ao filtro de Kalman nestas situações.

3. Navegação

Inercial e

Modelagem do

Problema

Primeiramente, começaremos apresentando os conceitos básicos de Navegação Inercial e suas aplicações no problema de estimação de posição de um veículo contendo uma Central Inercial. Feito isso, apresentaremos a modelagem do problema no espaço de estados de modo a deixar o sistema pronto para a aplicação do filtro de Kalman e dos filtros de partículas.

3.1. CONCEITOS BÁSICOS DE NAVEGAÇÃO INERCIAL

Inércia é a tendência de um corpo em manter sua velocidade de translação constante se ele não sofrer ação de forças ou torques externos (Primeira lei do movimento, de Isaac Newton).

Um referencial inercial é um sistema de coordenadas no qual as leis do movimento de Newton são válidas. Referenciais inerciais não podem estar sofrendo rotação nem aceleração. [2]

Para medirmos aceleração e velocidade de rotação de um certo corpo, podemos dispor de sensores inerciais, que são de 2 tipos:

– Giroscópios, que medem rotação: giroscópios de velocidade medem velocidade de rotação e giroscópios de deslocamento medem ângulo de rotação.

– Acelerômetros, que medem aceleração. Contudo, acelerômetros não podem medir a aceleração da gravidade.

O eixo de entrada de um sensor inercial define a componente vetorial que ele mede (eixo x, y ou z). Sensores *multiaxis* medem mais de uma componente vetorial. [2]

A Navegação Inercial utiliza giroscópios e acelerômetros para manter estimativas da posição e da velocidade do veículo ou corpo no qual a Central Inercial está embarcada. Este corpo pode ser um carro, uma aeronave, um míssil, um avião, um navio, um robô submarino ou até um pig utilizado para inspeção de oleodutos.

Um Sistema de Navegação Inercial consiste do seguinte: [2]

- Uma Unidade de Medidas Inercial (UMI) ou Unidade de Referência Inercial (URI) (ou ainda Central Inercial) contendo uma combinação de sensores: 3 acelerômetros e 3 giroscópios. Estes sensores são montados numa mesma base para que eles mantenham a mesma orientação relativa.

- Computadores para processamento dos dados obtidos através da UMI. Estes computadores devem corrigir as medidas da UMI (como compensar a aceleração da gravidade, por exemplo) e integrar a aceleração resultante duas vezes a fim de estimar a posição do veículo no qual a UMI está embarcada.

Há 2 tipos principais de Sistemas de Navegação Inercial: os *gimballed systems* e os *strapdown systems*.

Um *gimbal* consiste de uma plataforma rígida (contendo 3 acelerômetros e 3 giroscópios) enrolada por 3 anéis de isolamento de rotação. Cada anel isola a plataforma da rotação em torno de um eixo (x, y ou z). Assim, com 3 anéis rotacionais podemos isolar a plataforma de rotações externas. Não abordaremos o processamento de dados para os *gimballed systems*, já que a Central Inercial utilizada na coleta de dados foi uma *strapdown*. [2]

Nos *strapdown systems*, temos 3 acelerômetros e 3 giroscópios montados sobre uma mesma base rígida. Os eixos dos acelerômetros devem ser ortogonais, bem como os eixos dos giroscópios, para que cada sensor meça o movimento com respeito a um eixo específico (x, y ou z). Além disso, os eixos dos giroscópios devem coincidir com os eixos dos acelerômetros para que seja possível realizar a transformação de coordenadas, como será visto mais adiante. Esta combinação de sensores é então fixada num veículo, sem anéis para isolamento de rotação. Após a coleta de dados, os computadores que realizarão o processamento de dados devem integrar as equações do movimento com 6 graus de liberdade (no caso dos *gimballed*, é necessário integrar as equações com apenas 3 graus de liberdade). [2]

Na figura abaixo, podemos observar um *strapdown* e um *gimballed system*.

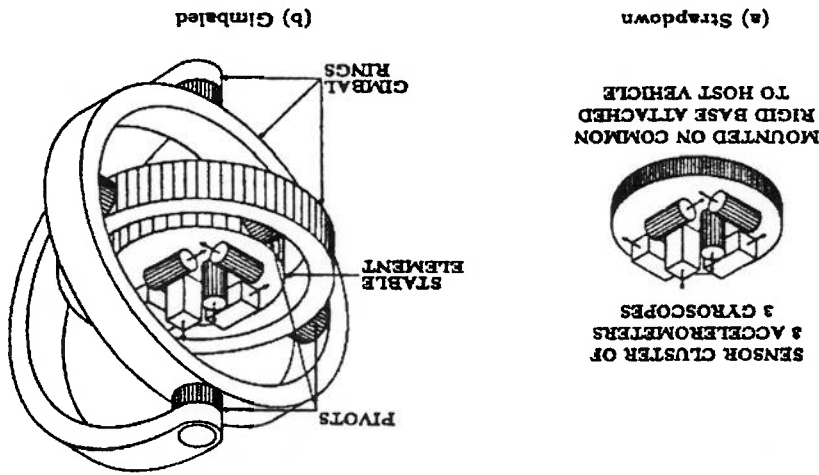


Figura 3.1 – *Strapdown system* (a) e *Gimballed system* (b)

As funções realizadas durante o processamento de dados de um *strapdown system* estão descritas no diagrama abaixo:

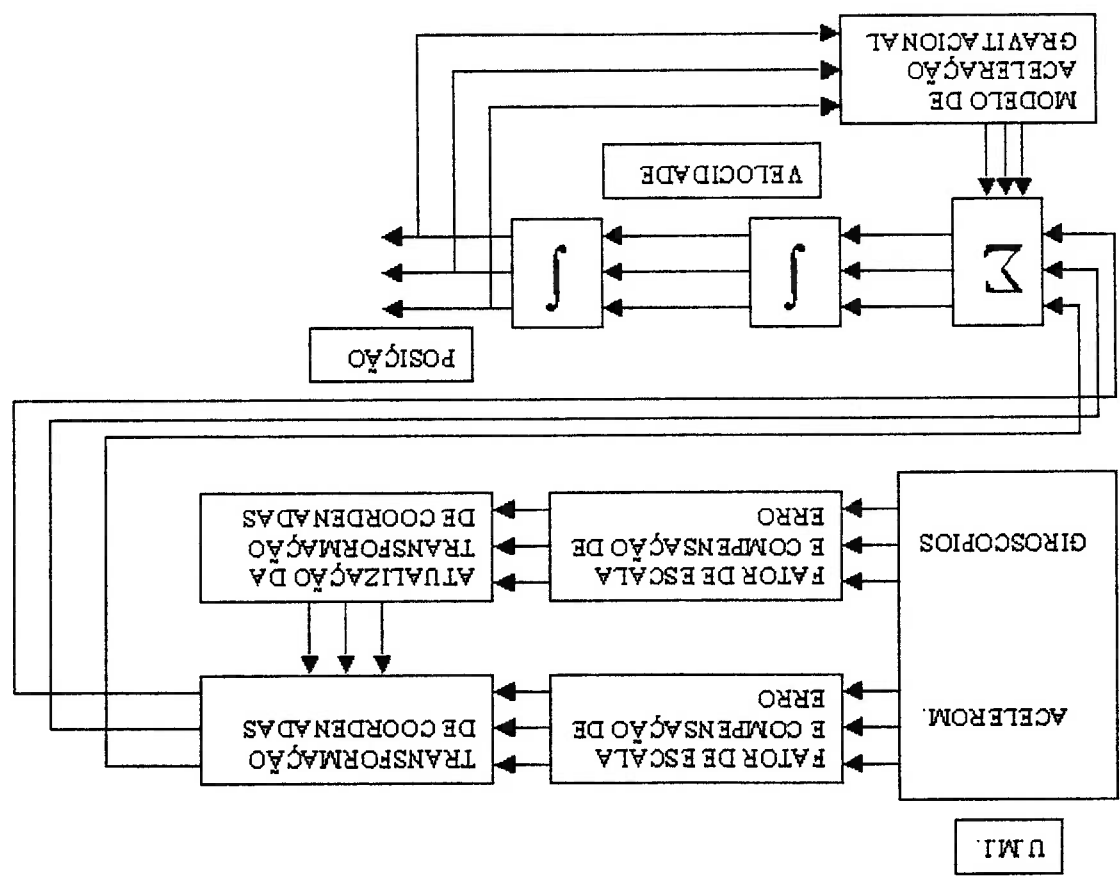


Figura 3.2 – Processamento dos dados de uma Central Inercial do tipo *strapdown*

Primeiramente, devemos aplicar um fator de escala nas saídas da Central Inercial, pois as medidas normalmente não são feitas nas unidades que nos interessam (os acelerômetros, por exemplo, medem força específica ao invés de aceleração – assim, deve-se multiplicar sua saída por um fator de escala para realizar a transformação adequada). Feito isso, é necessária uma compensação do erro, já que Centrais Inerciais tipicamente apresentam erros que podem ser eliminados deterministicamente (bias fixos, por exemplo).

O próximo passo consiste na transformação de coordenadas das acelerações, utilizando as medidas feitas pelos giroscópios. Isto é necessário porque uma Central

Inercial *strapdown* mede acelerações no sistema de coordenadas do veículo (sistema *roll-pitch-yaw*, ou *body fixed axis*), sendo que o sistema que nos interessa é o NED (*north-east-down*), que não se move junto com o veículo (é um sistema fixo cuja origem podemos escolher). Assim, fazendo a transformação de coordenadas teremos as acelerações no referencial adequado (o tratamento de dados para transformação de coordenadas será visto mais adiante).

Após isto, deve-se realizar uma correção importante nas acelerações, que é a aplicação do modelo gravitacional. Centrais Inerciais não medem aceleração gravitacional – portanto, esta correção deve ser feita para que o modelo fique adequado. Ao final desta etapa, podemos integrar o sinal obtido duas vezes para obter a posição do veículo no qual a Central Inercial está embarcada (segunda lei de Newton).

3.2. SISTEMAS DE COORDENADAS

Os Sistemas de Coordenadas são utilizados para referenciar a posição de um determinado corpo ou veículo. Assim, sempre que quisermos definir a posição de um objeto, devemos especificar um Sistema de Coordenadas. Esta posição será sempre relativa ao Sistema de Coordenadas especificado. Normalmente, um Sistema de Coordenadas consiste de 3 eixos ortogonais (x, y e z). Os Sistemas de Coordenadas mais utilizados são os seguintes: [2]

- Sistema ECI (*Earth Centered Inertial*) – sua origem está no centro de massa da Terra e seus eixos principais estão nas direções do Equinócio Vernal e do eixo de rotação da Terra (figura 3.3). Este sistema é inercial.

- Sistema ECEF (*Earth Centered Earth Fixed*) – sua origem está no centro de massa da Terra e seus eixos estão nas direções da interseção entre o meridiano de Greenwich e a linha do Equador e do eixo de rotação da Terra (figura 3.3). O Sistema ECEF não é inercial, pois seus eixos giram junto com a Terra.

- Sistemas LTP (*Local Tangent Plane*) – são Sistemas de Coordenadas locais, e representam a Terra como uma superfície plana em locais definidos. Estes sistemas incluem os seguintes:
 1. Sistema ENU (*East North Up*) – sua origem pode ser fixada em qualquer ponto da superfície terrestre. Os eixos estão na direção do Norte geográfico terrestre (eixo y), do leste geográfico terrestre (eixo x) e na direção perpendicular à superfície terrestre e apontando para cima (eixo z) (figura 3.4).
 2. Sistema NED (*North East Down*) – é similar ao Sistema ENU, com a diferença de que o eixo x está orientado para o norte geográfico da Terra, o eixo y está orientado para o leste geográfico da Terra e o eixo z aponta para baixo (figura 3.5).

- Sistemas Roll-Pitch-Yaw (RPY) – são sistemas com eixos fixos num veículo qualquer. Portanto, seus eixos movem-se junto com o veículo (figura 3.6). O eixo x deve estar na mesma direção do movimento do veículo, o eixo y deve ser ortogonal a x e apontar para o seu lado direito e o eixo z é definido de acordo com a regra da mão direita.

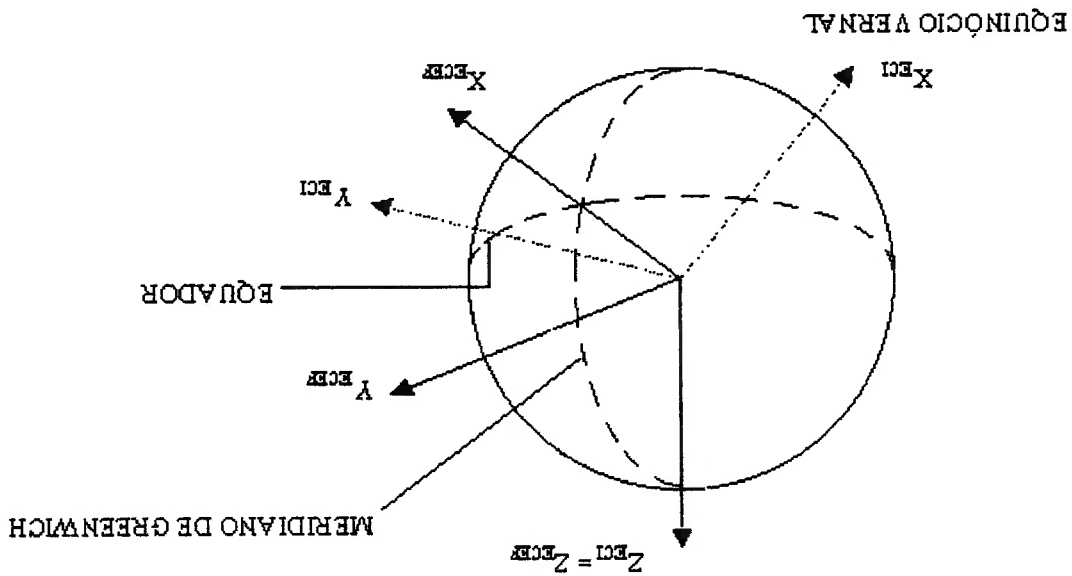


Figura 3.3 – Sistemas de Coordenadas ECI e ECEF

Figura 3.6 – Sistema de Coordenadas RPY

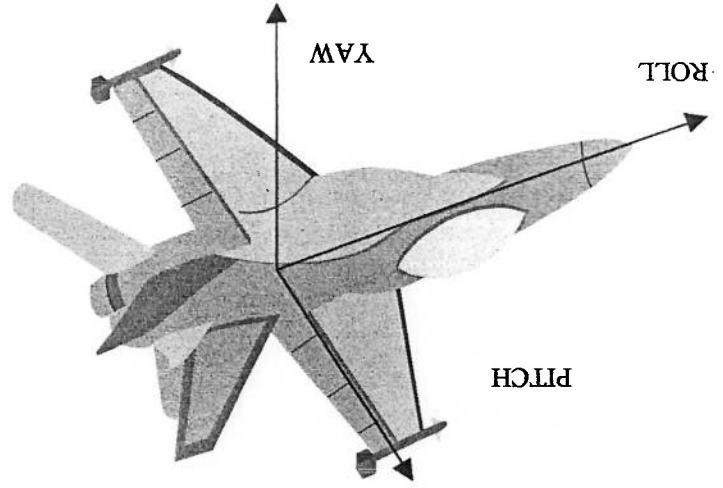


Figura 3.5 – Sistema de Coordenadas NED

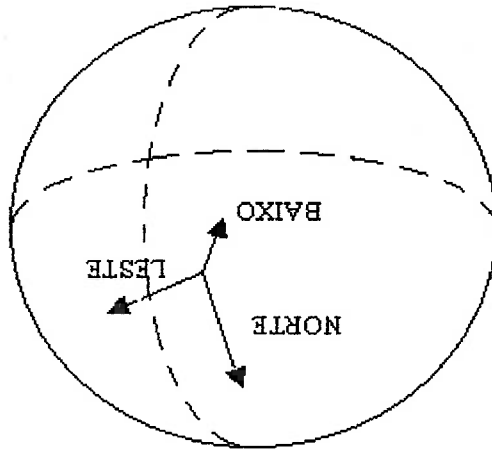
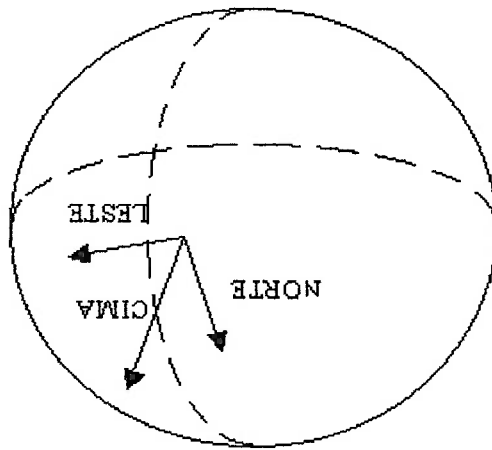


Figura 3.4 – Sistema de Coordenadas ENU



3.3. TRANSFORMAÇÕES DE COORDENADAS

Transformações de coordenadas são métodos utilizados para transformar um vetor representado em um certo Sistema de Coordenadas num vetor referenciado a um outro Sistema de Coordenadas. Estas transformações normalmente envolvem translações (para sistemas com origens diferentes) e rotações (para sistemas com eixos em direções diferentes). Como estamos trabalhando em 3 dimensões, a representação de um vetor qualquer (aceleração, velocidade, posição, etc.) será a seguinte:

$$a = \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix}$$

A matriz de transformação de coordenadas de um sistema x para um sistema y será denotada por C_x^y . No caso de uma Central Inercial do tipo *strapdown*, as componentes da aceleração medidas pelos 3 acelerômetros estão no sistema de coordenadas Roll-Pitch-Yaw (portanto, num sistema cujos eixos se movem junto com o veículo). Estas componentes da aceleração precisam ser transformadas para um sistema de coordenadas de navegação para que possamos integrar 2 vezes o sinal obtido (a fim de calcular velocidade e posição do veículo). Desta forma, seja $C_{veículo}^{navegação}$ a matriz de transformação de coordenadas do Sistema RPY (coordenadas fixas no veículo) para o Sistema de coordenadas de navegação (NED, por exemplo), e seja $a_{veículo}$ o vetor de acelerações do veículo no Sistema RPY. Temos o seguinte:

$$a_{navegação} = C_{veículo}^{navegação} a_{veículo} \quad (3.1)$$

Assim, $a_{navegação}$ será o vetor de acelerações do veículo no Sistema de coordenadas de navegação. Uma vez que este cálculo foi feito, devemos realizar algumas correções nas acelerações e integrar o sinal obtido 2 vezes para estimar velocidade e posição do veículo.

Há muitas maneiras de implementarmos transformações de coordenadas. O método mais comum de fazermos isto é através das matrizes de transformação dos ângulos de Euler (há outras formas, como Quaternions, que em algumas aplicações podem ser mais adequados; no entanto, o método utilizado aqui foi o dos ângulos de Euler pois este é mais fácil de depurar e implementar, além de ser apropriado ao problema em questão).

A orientação de um veículo em relação a um Sistema de coordenadas locais pode ser especificada através de ângulos de rotação em torno dos eixos *roll*, *pitch* e *yaw* deste veículo, sendo que na posição de partida o veículo deve estar com seus eixos alinhados com o Sistema NED. Estes ângulos de rotação são chamados ângulos de Euler, e é sempre necessário definirmos a ordem em que as rotações devem ser feitas (neste caso, a ordem será sempre *roll-pitch-yaw*). [2]

Portanto, os ângulos de Euler são utilizados para definir uma transformação de coordenadas em função de 3 rotações angulares, realizadas numa sequência específica em torno de 3 eixos ortogonais. A aplicação desta transformação faz o Sistema de coordenadas inicial coincidir com o final (pois é realizada uma rotação de eixos). Para transformarmos um vetor do Sistema RPY para o Sistema NED (esta será a transformação utilizada daqui em diante), a matriz de rotação é a seguinte: [2]

$$C_{RPY}^{NED} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(roll) & \sin(roll) \\ 0 & -\sin(roll) & \cos(roll) \end{bmatrix} \cdot \begin{bmatrix} \cos(pitch) & 0 & \sin(pitch) \\ 0 & 1 & 0 \\ -\sin(pitch) & 0 & \cos(pitch) \end{bmatrix} \quad (3.2)$$

A primeira matriz acima é a matriz de rotação em torno do eixo *roll*. A segunda é a matriz de rotação em torno do eixo *pitch* e a terceira é a matriz de rotação em torno do eixo *yaw* (ver figura 3.6).

$$P^k = \begin{bmatrix} roll^k \\ pitch^k \\ yaw^k \end{bmatrix}$$

angulares, devemos fazer $P = -\omega_k \cdot T$. [2]

girando no sentido de $-\omega$. Portanto, para calcularmos o vetor de incrementos coordenadas de navegação vistas de um observador fixo no veículo parecem estar fixo no veículo. Se o vetor de velocidades angulares medidas pelos giro ω , as para as coordenadas de navegação (NED, por exemplo) se vistas por um observador estar girando ao redor da Terra no sentido oposto à rotação terrestre. O mesmo ocorre Contudo, para um observador fixo na Terra (que está sofrendo rotação), o Sol parece cada um localizado entre 2 amostragens consecutivas dos giroscópios.

estaremos definindo uma série de incrementos angulares descritos pelo veículo (P^k), T é o intervalo de amostragem utilizado no processo. Fazendo este cálculo,

$$P^k = \omega_k \cdot T \quad (3.3)$$

transformamos este vetor de velocidades angulares num vetor de ângulos, faremos:

angulares. A cada instante de tempo, os giroscópios geram um vetor ω . Para ω é um vetor de rotações medidas pelos giroscópios, composto por 3 velocidades

$$\omega = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}$$

giroscópios para estimarmos estes ângulos. Supondo que tenhamos:

rotação), e não ângulos. Faz-se necessário, portanto, integrar o sinal obtido pelos Inercial. No entanto, os giroscópios medem velocidades angulares (ou taxas de ângulos devem ser determinados a partir das leituras dos giroscópios da Central em torno do eixo $pitch$ e yaw é o ângulo de rotação em torno do eixo yaw . Estes Assim, $roll$ é o ângulo de rotação em torno do eixo $roll$, $pitch$ é o ângulo de rotação

3.4. MODELOS DE ERRO DE ACELERAÇÃO

Os acelerômetros, assim como qualquer outro tipo de sensor, apresentam erros típicos de medida (o que será dito aqui pode ser entendido também para os giroscópios). Os mais comuns são os chamados bias (ou vícios), que são um desvio na medida. Os bias podem ser constantes ou variáveis. Quando eles são constantes, sua compensação é simples, bastando descontarmos das medidas dos sensores o seu valor. Na prática, entretanto, muitas vezes o bias do sensor é variável (ele funciona como um ruído aleatório). Assim, compensar as leituras dos acelerômetros destes tipos de erro é muito complicado, sendo que a filtragem do sinal é o modo mais usual de eliminarmos estes erros. O filtro de Kalman e os filtros de partículas, já vistos anteriormente, trabalham neste sentido (eliminação de ruídos aleatórios de medição).

Além do bias, os acelerômetros também apresentam fatores de escala (isto se deve ao fato de o acelerômetro medir força específica, e não aceleração – deve ser feita uma mudança de escala através de uma constante, portanto), e normalmente eles apresentam variações aleatórias (chamadas *scale factor instability*). Outro tipo de incerteza presente nas medidas dos acelerômetros é o fator de segunda ordem (chamado *g-squared*), cujo ruído é proporcional ao quadrado da aceleração medida. Ainda pode aparecer outro tipo de erro, chamado *cross axis*, que é proporcional ao produto de duas componentes da aceleração, medidas por dois acelerômetros distintos. Apesar de estes erros estarem sempre presentes nas medidas dos acelerômetros, eles não são tão relevantes quanto os bias, e em alguns casos não é necessário compensá-los. [2]

O modelo de erro para a aceleração medida por um acelerômetro, então, será o seguinte:

$$a_{real} = a_{medida} - C_0 - C_1 \cdot a_{medida} - C_2 \cdot a_{medida}^2 - C_3 \cdot a_{medida,x} \cdot a_{medida,y} \quad (3.4)$$

Os erros de medida dos acelerômetros apresentados acima são aleatórios. Normalmente eles são eliminados utilizando-se filtragem estocástica. Entretanto, acelerômetros e giroscópios fornecem medidas que apresentam também erros determinísticos, que podem ser calculados com base nas leis da Física e que devem ser compensados nas medidas obtidas. Os erros determinísticos a serem eliminados são a aceleração da gravidade, a aceleração centrífuga, a aceleração de Coriolis e a velocidade de rotação da Terra (para os giroscópios). [13]

O efeito de Coriolis consiste numa aceleração apresentada por um corpo que se move num referencial girante. Neste caso, teremos a Central Inercial dentro de um veículo movendo-se a uma velocidade instantânea v (a cada instante de tempo, ou seja, a cada amostra coletada) num referencial que sofre rotação (a Terra, que gira a uma velocidade angular Ω). Assim, a aceleração de Coriolis pode ser modelada do seguinte modo (\otimes denota produto vetorial): [2]

$$a^{Coriolis} = 2 \cdot \Omega \otimes v = 2 \cdot \begin{bmatrix} \Omega_1 \\ \Omega_2 \\ \Omega_3 \end{bmatrix} \otimes \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = 2 \cdot \begin{bmatrix} \Omega_1 v_2 - \Omega_2 v_1 \\ \Omega_2 v_3 - \Omega_3 v_2 \\ \Omega_3 v_1 - \Omega_1 v_3 \end{bmatrix} \quad (3.5)$$

O vetor Ω , no Sistema de coordenadas ECEF, é o seguinte (a Terra gira em torno do eixo z, no Sistema ECEF):

$$\Omega = \begin{bmatrix} 0 \\ 0 \\ 0.00007292 \end{bmatrix} \text{ rad/s}$$

O vetor v , no entanto, que será calculado a cada iteração integrando-se a aceleração do veículo medida pelos acelerômetros e transformada através dos ângulos de Euler, estará no Sistema NED. Desta forma, precisaremos transformar o vetor Ω para o Sistema NED a fim de podermos processar o cálculo da aceleração de Coriolis. Esta transformação de coordenadas é realizada através da seguinte matriz: [14]

R é a distância entre o centro da Terra e o centro de massa do veículo (normalmente isso equivale ao raio da Terra). Assim, podemos calcular a aceleração centrífuga no referencial ECEF. Feito isso, deve-se proceder uma transformação de coordenadas desta aceleração do Sistema ECEF para o Sistema NED (matriz C_{ECEF}^{NED} , já vista anteriormente) para que possamos, finalmente, subtrair o vetor de aceleração centrífuga do vetor de aceleração medido pelos acelerômetros e transformado para o Sistema NED.

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} R \cdot \cos(\lambda) \cdot \cos(\phi) \\ -R \cdot \sin(\lambda) \cdot \cos(\phi) \\ -R \cdot \sin(\phi) \end{bmatrix} \quad (3.8)$$

Ω é o vetor de velocidade de rotação da Terra (Sistema ECEF) e r_v é a posição do veículo no Sistema ECEF. Podemos calcular esta posição da seguinte forma:

$$a^{centrifuga} = -\Omega \otimes \Omega \otimes r_v \quad (3.7)$$

Outra correção determinística que precisa ser feita é a aplicação do modelo gravitacional terrestre, que consiste na aceleração da gravidade e na aceleração centrífuga. A aceleração centrífuga é causada pela diferença de posição entre o veículo e o eixo de rotação da Terra e pela velocidade de rotação terrestre. Sua fórmula é a seguinte: [13]

ϕ é a latitude do veículo, e λ é a sua longitude. Com isto, poderemos calcular a aceleração de Coriolis e subtrai-la, a cada iteração, do vetor de acelerações medidas pelos acelerômetros e transformadas para o Sistema NED.

$$C_{ECEF}^{NED} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ -1 & 0 & 0 \end{bmatrix} \begin{bmatrix} \cos(\phi) & 0 & \cos(\phi) \\ 0 & 1 & 0 \\ -\sin(\phi) & 0 & \cos(\phi) \end{bmatrix} \begin{bmatrix} \cos(\lambda) & \sin(\lambda) & 0 \\ -\sin(\lambda) & \cos(\lambda) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.6)$$

A aceleração da gravidade terrestre também deve ser descontada das leituras dos acelerômetros, já que eles não são capazes de medir esta aceleração (um acelerômetro em queda livre apresenta leitura 0). A aceleração da gravidade apresentada por um corpo no planeta Terra pode ser descrita pela seguinte equação:

$$g = \frac{GM_T}{R^2} \approx 9.80665 m/s^2 \quad (3.9)$$

G é a constante gravitacional, M_T é a massa da Terra e R é a distância entre o centro da Terra e o centro de massa do veículo. Nos experimentos realizados, como o veículo não teve a sua altitude muito alterada em relação à superfície terrestre (os testes com a Central Inercial foram feitos para aplicações terrestres), podemos efetuar a aproximação acima (onde $R \approx$ Raio da Terra). No Sistema NED, o vetor de aceleração da gravidade é o seguinte:

$$g = \begin{bmatrix} 9.80665 \\ 0 \\ 0 \end{bmatrix} m/s^2$$

Este vetor de aceleração da gravidade deve ser somado às leituras dos 3 acelerômetros transformadas para o Sistema NED. Assim, o modelo utilizado para fazer as correções determinísticas nas leituras dos acelerômetros pode ser equacionado do seguinte modo:

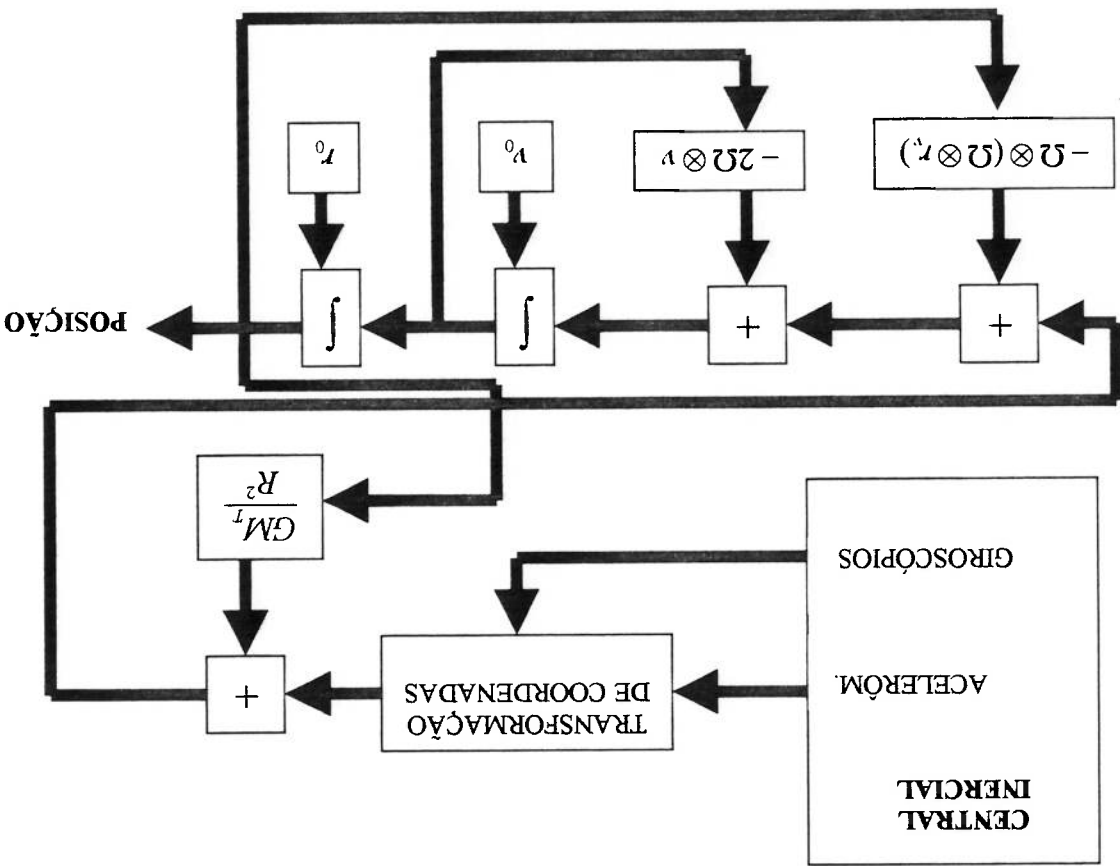
$$a_v = C_{RPY}^{NED} a_m - (C_{ECFF}^{NED} \cdot 2 \cdot \Omega) \otimes v + g - C_{ECFF}^{NED} \cdot (\Omega \otimes \Omega \otimes r_v) \quad (3.10)$$

a_v é a aceleração corrigida do veículo; o primeiro termo na equação acima é o vetor de aceleração medido pelos acelerômetros e transformado para o Sistema NED, o segundo termo é a aceleração de Coriolis, o terceiro termo é a aceleração da gravidade e o quarto termo é a aceleração centrífuga. A equação acima é a equação Universal da Navegação Inercial. [13]

A razão de termos de descontar todos estes erros de aceleração é a seguinte: quando integramos 2 vezes o sinal de aceleração para obter posição, estamos na verdade multiplicando a aceleração pelo quadrado do tempo. Assim, integrar duas vezes um erro no período de 1 hora equivale a multiplicar este erro por aproximadamente 13 milhões de segundos ao quadrado, e se não corrigirmos o erro de aceleração centrífuga, por exemplo, ao final de 1 hora teremos um erro acumulado de posição superior a 3 quilômetros. Estes erros, portanto, dependem diretamente do tempo e podem causar desvios grandes nas estimativas se não forem devidamente corrigidos.

[13]

Figura 3.7 – Diagrama da Mecanização de um Sistema de Navegação Inercial



A mecanização básica para implementação de um Sistema de Navegação Inercial, portanto, é a seguinte:

Para corrigir os erros dos giroscópios, é necessário apenas descontarmos de suas medidas a velocidade angular de rotação da Terra. Isto pode ser feito do seguinte modo:

No Sistema NED, a velocidade angular de rotação da Terra pode ser escrita da seguinte forma:

$$\Omega_{NED} = C_{BCEF}^{NED} \Omega_{BCEF} \quad (3.11)$$

Para que este vetor possa ser subtraído das leituras dos giroscópios, elas também devem ser transformadas para o Sistema NED:

$$\omega_{NED} = C_{RPT}^{NED} \omega_{RPT} \leftarrow \omega_{CORRIGIDO} = \omega_{NED} - \Omega_{NED} \quad (3.12)$$

Os passos para a determinação da posição de um veículo a partir dos dados fornecidos pela sua Central Inercial, desta forma, são os seguintes: [2]

1. Aplicação do fator de escala – Devemos multiplicar a saída dos acelerômetros e giroscópios por constantes (fatores de escala), que são relações entre unidade de aceleração e unidade de saída da Central Inercial (por exemplo, metros por segundo ao quadrado por volt), para que obtenhamos aceleração (ou rotação angular).

2. Deve-se realizar a transformação de coordenadas das acelerações medidas, para que elas fiquem relacionadas a um referencial adequado. Esta transformação é realizada com base nas medidas dos giroscópios.

3. Os acelerômetros e giroscópios têm erros de medida, a saber: bias (desvios de medida, constantes ou variáveis), variações aleatórias do fator de escala, desalinhamentos desconhecidos entre eixos dos acelerômetros e giroscópios (em condições ideais, estes eixos devem ser ortogonais) e ruído aleatório nas saídas

dos sensores. Estes ruídos normalmente são eliminados utilizando-se filtragem do sinal (por exemplo, filtro de Kalman).

4. Acelerações gravitacionais e de Coriolis devem ser modeladas e descontadas das leituras dos acelerômetros para obtermos a aceleração real do veículo (acelerômetros não medem aceleração da gravidade).

5. Deve-se integrar o sinal da aceleração real do veículo para se obter sua velocidade. Este processo de integração requer condições iniciais de velocidade, ou seja, é preciso que saibamos a velocidade inicial do veículo.

6. Deve-se integrar o sinal da velocidade do veículo para se obter sua posição. Este processo também requer condições iniciais, o que significa que precisamos conhecer a posição inicial do veículo.

3.5. INFLUÊNCIA DOS ERROS DOS SENSORES NA ESTIMAÇÃO DE POSIÇÃO

No caso dos giroscópios, é gerado um sinal proporcional à velocidade angular do veículo somada a ruídos e desvios (bias). Para curtos intervalos de tempo, os desvios podem ser considerados constantes. Isto pode ser descrito pela equação abaixo: [15]

$$\theta^m = \int (\dot{\theta} + b + w) dt = \theta + b t + \int w dt \quad (3.13)$$

Assim, integrando-se o sinal obtido pelos giroscópios, temos o ângulo de rotação do veículo adicionado a 2 termos: um *random walk* devido ao ruído w (ruído do sensor, ou bias variável) e outro termo que aumenta linearmente com o tempo t e é proporcional ao bias do giro (b , que é um bias constante). O erro causado pelo bias dos giroscópios irá gerar uma avaliação incorreta da orientação dos eixos dos acelerômetros x e y , acoplando uma componente da aceleração do eixo x no eixo y . Uma aceleração constante a_x acoplada no eixo y irá induzir erros na aceleração,

Para que possamos processar os cálculos de posição para um Sistema de Navegação Inercial, é necessário que ele esteja devidamente alinhado no instante inicial da coleta de dados. Para isto, é preciso que um de seus eixos aponte para o Norte geográfico terrestre e sua plataforma (ou seja, o plano definido pelos eixos que

3.6. ALINHAMENTO INICIAL

Kalman e os filtros de partículas. Todos estes erros, se não forem devidamente compensados, introduzirão um desvio no cálculo da posição que comprometerá por completo as estimativas, visto que eles crescerão proporcionalmente ao quadrado e ao cubo do tempo. Os erros aleatórios serão compensados utilizando-se 2 métodos de cancelamento de ruído: o filtro de

$$e_{v^x} = b^x t$$

$$e_{p^x} = \frac{1}{2} b^x t^2$$

(3.15)

Desta forma, o bias constante dos giroscópios pode ocasionar erros no cálculo da posição que são proporcionais a t^3 . Já o bias dos acelerômetros pode ocasionar erros na estimativa da posição proporcionais ao quadrado do tempo: [15]

$$e_{a^y} = a^y b^y t$$

$$e_{v^y} = \frac{1}{2} a^y b^y t^2$$

$$e_{p^y} = \frac{1}{6} a^y b^y t^3$$

(3.14)

pequenos, estes erros podem ser calculados da seguinte forma: [15]

velocidade e posição na componente y. Assumindo que os ângulos envolvidos são

apontam para o Norte e para o Leste da Terra) esteja nivelada (assim, um dos 3 eixos deve estar perpendicular à superfície terrestre).

Nos testes realizados, foi utilizado o auto – alinhamento: o nivelamento é feito rotacionando-se a plataforma em torno de seus eixos horizontais até que os acelerômetros horizontais indiquem leitura zero e o acelerômetro vertical indique leitura –1g. O alinhamento do eixo x para o Norte geográfico terrestre (também conhecido por *gyrocompassing*) é feito rotacionando-se a plataforma nivelada em torno do eixo vertical até que um dos giroscópios não indique leitura da velocidade de rotação da Terra. [13]

3.7. MODELAGEM DO SISTEMA NO ESPAÇO DE ESTADOS DISCRETO

O modelo utilizado para estimarmos a posição de um veículo a partir dos dados fornecidos por uma Central Inercial embarcada neste veículo (ou seja, acelerações em 3 eixos ortogonais e rotações angulares em torno de 3 eixos) será descrito no espaço de estados. Isto facilitará a posterior implementação da estimação recursiva através dos filtros (filtro de Kalman e filtro de partículas), que são aplicados a modelos no espaço de estados. Assim, nosso Sistema será modelado do seguinte modo:

$$x_{k+1} = A \cdot x_k + B \cdot u_k + C \cdot w_k$$

(3.16)

$$y_k = H \cdot x_k + G \cdot e_k$$

Neste modelo, x_k é o vetor de estados do Sistema, u_k é o vetor de entradas do Sistema, y_k é o vetor de medidas de referência (ou saídas), w_k é um vetor de ruídos aleatórios do processo e e_k é um vetor de erros de medição. A, B, C são as matrizes que definem a dinâmica do processo e G e H são as matrizes que definem a equação de medição.

Neste caso, podemos utilizar as leis do movimento de Newton para obter posição a partir de aceleração (p_i é a posição do veículo no instante t_i , v_i é a velocidade do veículo no instante t_i e a_i é a aceleração do veículo no instante t_i): [16]

$$(3.17) \quad \left\{ \begin{array}{l} a_i = \frac{dp_i}{dt} \\ v_i = \frac{dv_i}{dt} \end{array} \right. \Leftrightarrow p_i = p_0 + v_0 T + \frac{1}{2} a_i T^2$$

Se o período de amostragem do Sistema for igual a T , teremos o seguinte modelo discreto entre 2 medições consecutivas (essa aproximação é válida apenas para intervalos de amostragem T muito pequenos - assim, entre 2 amostragens consecutivas, a aceleração medida pelos acelerômetros é considerada constante, e a equação abaixo pode ser utilizada): [16]

$$(3.18) \quad p_{k+1} = p_k + v_k T + \frac{1}{2} a_k T^2$$

As medidas a que temos acesso, no caso da Central Inercial, são as acelerações nas direções x , y e z (neste caso, teremos as acelerações já transformadas para o referencial NED). Assim, a_k deverá ser o nosso vetor de entradas. Além disso, será considerado o erro de medida das acelerações, de modo que: [16]

$$(3.19) \quad a_{real,k} = a_k + \delta a_k$$

a_k é a aceleração medida e δa_k é o erro de medição. Este erro de medição fará parte do vetor de estados (pois queremos estimá-lo), resultando no seguinte modelo matricial discreto:

Para determinarmos a posição de um veículo com uma Central Inercial, basta utilizarmos o modelo acima (feitas as devidas transformações e correções, de acordo com o esquema da figura 3.7), utilizando como entradas as acelerações e inicializando o Sistema de Forma adequada. No entanto, uma simples simulação do modelo acima irá gerar erros de posição muito altos, após alguns poucos minutos

amostragem utilizado no processo.

Neste modelo, o vetor de estados é formado por 9 elementos: x , y e z são as 3 coordenadas de posição do veículo, v_x , v_y e v_z são as 3 componentes da velocidade do veículo e δa_x , δa_y e δa_z são as componentes do erro de medida de aceleração. O vetor de entradas do Sistema é formado por 3 elementos, que são as 3 componentes da aceleração do veículo fornecidas pela Central Inercial (estas são as acelerações já transformadas para o referencial NED). O vetor w_r , composto por 9 elementos independentes, é um vetor de ruídos aleatórios do processo. T é o período de

$$\begin{bmatrix} x^{k+1} \\ y^{k+1} \\ z^{k+1} \\ v_x^{k+1} \\ v_y^{k+1} \\ v_z^{k+1} \\ \delta a_x^{k+1} \\ \delta a_y^{k+1} \\ \delta a_z^{k+1} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x^k \\ y^k \\ z^k \\ v_x^k \\ v_y^k \\ v_z^k \\ \delta a_x^k \\ \delta a_y^k \\ \delta a_z^k \end{bmatrix} + \begin{bmatrix} 0.5T^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.5T^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.5T^2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.5T^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.5T^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.5T^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.5T^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.5T^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.5T^2 \end{bmatrix} \begin{bmatrix} a_x^k \\ a_y^k \\ a_z^k \end{bmatrix} + 10 \begin{bmatrix} w_r^k & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ w_r^k & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ w_r^k & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ w_r^k & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ w_r^k & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ w_r^k & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ w_r^k & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ w_r^k & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ w_r^k & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ w_r^k & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.20)$$

(vide seção 3.5), e isso se deve à presença de ruídos aleatórios nas medidas fornecidas pelos acelerômetros e giroscópios. Sendo este ruído imprevisível, é muito complicado compensá-lo através de métodos determinísticos. O modo mais adequado de o fazermos é utilizando filtros estocásticos, como o filtro de Kalman. Para utilizarmos estes filtros, devemos ter uma referência do vetor de estados (a referência pode ser posição ou velocidade) – estimar posição baseando-se apenas nas medidas da Central Inercial é inviável, já que os erros de medição desta Central são considerados elevados (ver seção 3.8). Neste caso, a referência será a velocidade do veículo, fornecida por um velocímetro (a referência pode apresentar erro, o que já está considerado no modelo) ou sua posição, fornecida indiretamente por um odômetro. O importante é que o ruído do processo (w) não tenha o mesmo perfil do ruído da referência (e) – se eles tiverem o mesmo perfil, o filtro de Kalman não funcionará de modo adequado. Como o ruído dos sensores da Central Inercial e o ruído de um velocímetro não apresentam o mesmo perfil (o ruído da Central é cumulativo no tempo, enquanto que o ruído do velocímetro normalmente não é), o filtro de Kalman (ou os filtros de partículas) consegue combinar essas características de modo a estimar a posição do veículo com erros relativamente pequenos (da ordem de 1 a 2%).

O modelo de medição de referência esta descrito abaixo:

$$\begin{bmatrix} x_k \\ y_k \\ z_k \\ vx_k \\ vy_k \\ vz_k \\ \delta ax_k \\ \delta ay_k \\ \delta az_k \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} + 0.05 \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} e_1^k \\ e_2^k \\ e_3^k \end{bmatrix} \tag{3.21}$$

Neste modelo, e é um vetor de erros das medidas de referência, e y é o vetor das medidas de referência (no caso, velocidades decompostas em 3 direções).

MODELO PARA INICIALIZAÇÃO DO SISTEMA

O equacionamento que será utilizado para inicializar o sistema deve considerar modelos de erro mais sofisticados. De modo a modelar erros de acelerômetros e giroscópios, considerando ainda a dinâmica do processo, sejam as equações abaixo, que especificam as relações físicas entre acelerômetros e giroscópios em apenas 1 eixo: [47]

$$\Delta \ddot{x} = a - g \phi$$

$$\dot{\phi} = \frac{1}{R} \Delta \dot{x} + \varepsilon$$

É considerada a seguinte nomenclatura:

- Δx - Erro de posição
- $\Delta \dot{x}$ - Erro de velocidade
- $\Delta \ddot{x}$ - Erro de aceleração
- ϕ - Erro de nivelamento da plataforma
- a - Ruído do acelerômetro
- ε - Ruído do giroscópio

Na primeira das equações (3.22), o erro de aceleração é devido a uma combinação de ruído do acelerômetro e uma componente da aceleração gravitacional que o acelerômetro mede devido ao erro de nivelamento da plataforma (a plataforma é considerada nivelada, mas na verdade ela está inclinada de um ângulo ϕ em relação à linha horizontal, que é muito pequeno, mas não pode ser anulado – assim, ao invés de a aceleração da gravidade ser medida exclusivamente pelo acelerômetro fixado no eixo z (eixo vertical), ela passa a ter componentes também nos eixos x e y). A segunda das equações (3.22) modela a taxa de variação do erro de nivelamento da plataforma, que é resultante do ruído do giroscópio somado a um erro de velocidade linear que, quando projetado sobre a superfície terrestre (que apresenta uma

curvatura), transforma-se num erro de velocidade angular. O erro do acelerômetro, que integrado uma vez origina um erro de velocidade, causa um aumento no erro de nivelamento da plataforma (segunda das equações 3.22). Este aumento no erro de nivelamento da plataforma causará um aumento na componente da aceleração da gravidade medida pelo acelerômetro horizontal (primeira das equações 3.22) – este erro, segundo a equação (3.22), cancelará em parte o ruído do acelerômetro. O fenômeno resultante da combinação destas 2 fontes de erro é conhecido como Oscilação de Schuller, e é responsável pela estabilidade dos erros associados aos eixos horizontais.

Quando equacionamos os erros de sensores de um Sistema de Navegação Inercial em 3 eixos a partir do equacionamento acima (apenas para 1 eixo), surgem termos adicionais, derivados das interações entre os sensores. As equações que modelam estes erros para um SNI em 3 eixos são dadas abaixo (ω é a velocidade angular de rotação da plataforma):

$$\Delta \ddot{x} = a_x - g \cdot \phi_y$$

$$\Delta \ddot{y} = a_y - g \cdot (-\phi_x)$$

$$\Delta \ddot{z} = a_z$$

(3.23)

$$-\dot{\phi}_x = \frac{1}{R} \Delta \dot{y} + \omega_y \cdot \phi_z + \varepsilon_x$$

$$\dot{\phi}_y = \frac{1}{R} \Delta \dot{x} + \omega_x \cdot \phi_z + \varepsilon_y$$

$$\dot{\phi}_z = \varepsilon_z$$

Os modelos de erro dos sensores associados aos eixos x e y (sensores horizontais) estão sujeitos à oscilação de Schuller, e isto garante a estabilidade destes erros. Já o canal vertical (sensor associado ao eixo z) não se beneficia deste fenômeno, e portanto seu erro é instável no tempo.

Um modelo dinâmico de 9 estados será utilizado para inicializar o sistema (ou seja, este modelo será utilizado para estimarmos o erro inicial do sistema – erros de nivelamento da plataforma, acelerômetro e giroscópio; uma vez que este erro inicial foi estimado, podemos subtrair-lo do vetor de acelerações medidas pelos acelerômetros), com o auxílio do filtro de Kalman. As variáveis do vetor de estados serão as seguintes:

- x_1 – Erro de posição no eixo x
- x_2 – Erro de posição no eixo y
- x_3 – Erro de posição no eixo z
- x_4 – Erro de velocidade no eixo x
- x_5 – Erro de velocidade no eixo y
- x_6 – Erro de velocidade no eixo z
- x_7 – Ângulo de inclinação da plataforma em relação ao eixo x
- x_8 – Ângulo de inclinação da plataforma em relação ao eixo y
- x_9 – Diferença angular entre o eixo x e o Norte geográfico terrestre

Tendo como base o conjunto de equações (3.23), pode-se escrever a equação matricial para inicialização do Sistema de Navegação Inercial:

$$(3.24) \quad \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \\ x_9 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \\ x_9 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \\ a_x \\ a_y \\ a_z \\ \varepsilon_x \\ \varepsilon_y \\ \varepsilon_z \end{bmatrix}$$

Escrevendo de forma mais simplificada, $\dot{x} = A_{INS} \cdot x + w$.

A equação matricial (3.24) pode ser transformada num modelo discreto, bastando para isso utilizarmos uma aproximação dada por $\Phi_{INS} \approx e^{A^T T} + A^T T$, para períodos de amostragem T muito pequenos:

$$x = A_{INS} \cdot x + w \rightarrow x_{k+1} = \Phi_{INS} \cdot x_k + w R_k \quad (3.25)$$

$$\Phi_{INS} \approx \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

(3.26)

Os erros associados aos acelerômetros e giroscópios ($a_x, a_y, a_z, \epsilon_x, \epsilon_y, \epsilon_z$) serão modelados utilizando-se a teoria de Processos Estocásticos [3] (pois estes erros são variáveis aleatórias). O modelo mais adequado é um processo de Gauss Markov de primeira ordem, que representará cada erro de sensor do Sistema de Navegação Inercial. Como o valor exato destas variáveis é desconhecido, elas serão incluídas no vetor de estados, que deste modo terá 6 estados adicionais – 3 deles serão os erros dos acelerômetros e outros 3 serão os erros dos giroscópios. O processo de Gauss Markov que representará os erros dos sensores é o seguinte:

$$a_{k+1} = \beta \cdot a_k + w R_k \quad (3.27)$$

No modelo acima, $w R_k$ é o processo aleatório ruído branco Gaussiano e $\beta = e^{-T/\tau}$, com τ sendo a constante de tempo da função de autocorrelação exponencial que rege o processo. Adicionando os estados supracitados ao modelo descrito pela equação (3.24), teremos o seguinte modelo:

Desta forma, o modelo dado pelas equações (3.26), (3.28) e (3.29) pode ser utilizado para inicializar o Sistema de Navegação Inercial. Contudo, para isto precisamos de medidas de referência para integrá-las com este modelo através do filtro de Kalman e estimarmos seus estados iniciais (do mesmo modo como foi mencionado

$$CO = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (3.29)$$

A matriz CO do modelo acima é uma matriz de conexões entre os estados adicionais de erros dos sensores (x_{10}, \dots, x_{15}) e os outros estados (x_1, \dots, x_9):

- x_{10} – erro associado ao acelerômetro x
- x_{11} – erro associado ao acelerômetro y
- x_{12} – erro associado ao acelerômetro z
- x_{13} – erro associado ao giroscópio x
- x_{14} – erro associado ao giroscópio y
- x_{15} – erro associado ao giroscópio z

Os estados adicionais são os seguintes:

$$\begin{bmatrix} x_{15} \\ x_{14} \\ x_{13} \\ x_{12} \\ x_{11} \\ x_{10} \\ x_{1-9} \end{bmatrix}^{k+1} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_{15} \\ x_{14} \\ x_{13} \\ x_{12} \\ x_{11} \\ x_{10} \\ x_{1-9} \end{bmatrix}^k + \begin{bmatrix} w_{r_{15}} \\ w_{r_{14}} \\ w_{r_{13}} \\ w_{r_{12}} \\ w_{r_{11}} \\ w_{r_{10}} \\ w_{r_{1-9}} \end{bmatrix}^k \quad (3.28)$$

anteriormente). Para inicialização de um SNI, o veículo deve permanecer estacionário (movendo-se com velocidade constante, por exemplo) e a plataforma deve estar nivelada. As medidas de referência a serem utilizadas serão a velocidade lida de um velocímetro automotivo (que deve ser transformada para o referencial NED) e os ângulos de inclinação da plataforma da Central Inercial (eles podem ser obtidos utilizando-se o software Gyroview, adquirido juntamente com a Central Inercial – este software possui aplicativos que fornecem a inclinação da Central Inercial em relação aos seus eixos x e y, sendo que estes ângulos podem ser mostrados juntamente com outros dados da Central Inercial). Assim, o modelo de medição de referência será o seguinte:

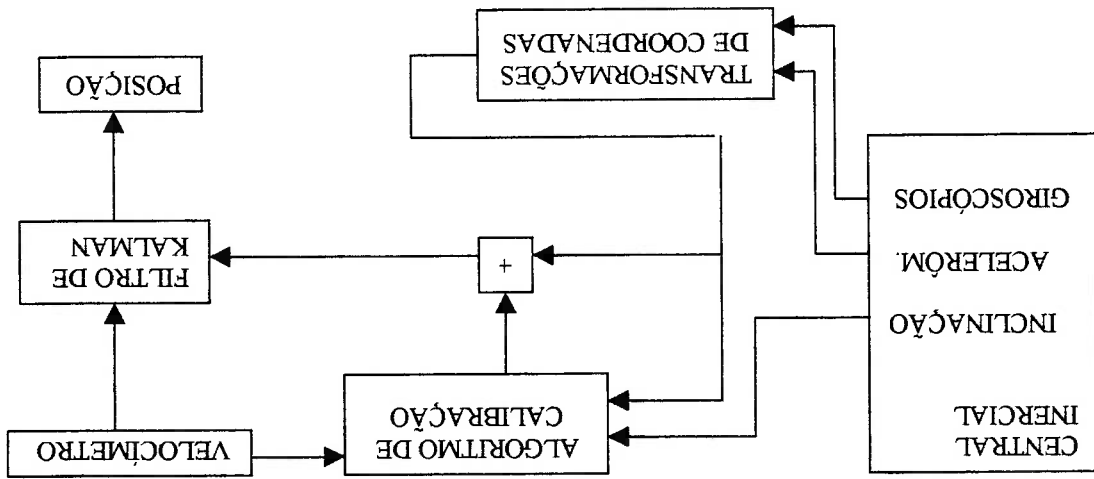
$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \\ x_9 \\ x_{10} \\ x_{11} \\ x_{12} \\ x_{13} \\ x_{14} \\ x_{15} \end{bmatrix} + \begin{bmatrix} e_1 \\ e_2 \\ e_3 \\ e_4 \\ e_5 \end{bmatrix} \quad (3.30)$$

Os erros de velocidade (y_1, y_2 e y_3) são calculados da seguinte forma: o sinal obtido através dos acelerômetros é corrigido, transformado para o referencial adequado e integrado uma vez (deste modo será obtida a velocidade instantânea do veículo). A velocidade assim obtida é então subtraída da velocidade lida no velocímetro do automóvel (já transformada para o referencial NED) para se calcular o erro de velocidade do SNI.

A Central Inercial utilizada foi o modelo VG 700 AA – 202 da Crossbow. Esta Central foi projetada principalmente para a realização de testes automotivos. Ela utiliza 3 acelerômetros do tipo MEMS (micro electro mechanical machines) e 3 giroscópios de fibra ótica (FOG – fiber optic gyro). Os acelerômetros são micro dispositivos de silício que utilizam capacitância diferencial para medir acelerações. Além disso, a Central possui um módulo de Processamento Digital de Sinais para compensação de erros determinísticos. Também foi utilizado o software Gyroview, adquirido juntamente com a Central Inercial. Este software é responsável pela aquisição de dados a partir da Central Inercial (a captura de dados é feita através de uma saída serial – RS 232) e pelo tratamento inicial destes dados, além de realizar

3.8. ESPECIFICAÇÕES RELEVANTES DA CENTRAL INERCIAL

Figura 3.8 – Diagrama esquemático da inicialização e estimação de posição para um Sistema de Navegação Inercial utilizando o filtro de Kalman.



Tendo este modelo, deve-se então utilizar o filtro de Kalman para estimar os erros iniciais de aceleração, rotação angular e inclinação da plataforma e descontá-los posteriormente do vetor de acelerações e rotações angulares dados pelos sensores, de acordo com a figura abaixo:

uma calibração dos sensores da Central. As especificações relevantes da Central são as seguintes: [17]

Taxa máxima de amostragem – em torno de 100 Hz

Bias dos giroscópios – máximo de 0.03 graus / segundo

Resolução dos giroscópios – máximo de 0.025 graus / segundo

Bias dos acelerômetros – máximo de 0.085 m / s²

Resolução dos acelerômetros – máximo de 0.01 m / s²

Temperatura de operação – -40 a 71 graus Celsius

Voltagem de alimentação – 10 a 30 VDC

Formato para captura de dados – RS 232

Dimensões físicas do equipamento – 12.7 x 15.24 x 10.16 cm

Conforme foi dito no capítulo 3, devemos integrar as medições da Central Inercial com medições de referência (velocidade, posição, etc.) a fim de reduzir o erro na estimação de posição. O modo mais adequado de integrar estas medidas é através do filtro de Kalman e dos filtros de partículas. Neste capítulo, serão apresentados os algoritmos básicos para implementação destes filtros (filtro de Kalman e filtros de partículas), bem como os ajustes necessários para a execução deles (os algoritmos foram executados em ambiente MATLAB). Também serão descritos os testes realizados com a Central Inercial, e os resultados obtidos serão então apresentados e discutidos. A principal questão é: Quais seriam as vantagens e desvantagens dos filtros de partículas em relação ao tradicional filtro de Kalman em aplicações relacionadas à Navegação Inercial ?

4. Testes Realizados, Resultados Experimentais e Análise

4.1. ALGORITMOS UTILIZADOS E AJUSTES

Serão apresentados cinco algoritmos distintos para estimação de posição: o duplo integrador, o filtro de Kalman, o filtro ASIR e o filtro MHIR.

4.1.1. DUPLO INTEGRADOR

O duplo integrador é a forma mais simples de estimarmos a posição de um veículo a partir dos seus dados de aceleração. O algoritmo consiste em integrarmos 2 vezes o sinal de aceleração (com as devidas correções e transformações para o referencial adequado) para obtermos posição (um requisito importante é que saibamos as condições iniciais de posição e velocidade). Devido ao fato de o duplo integrador não tratar os ruídos aleatórios presentes nas medidas da Central Inercial, ele não constitui um bom algoritmo para estimação de posição (além disso, ele não utiliza medidas de referência, apenas os dados fornecidos pela Central – assim, após alguns minutos o erro de posição obtido é da ordem de centenas de metros, dependendo da aplicação).

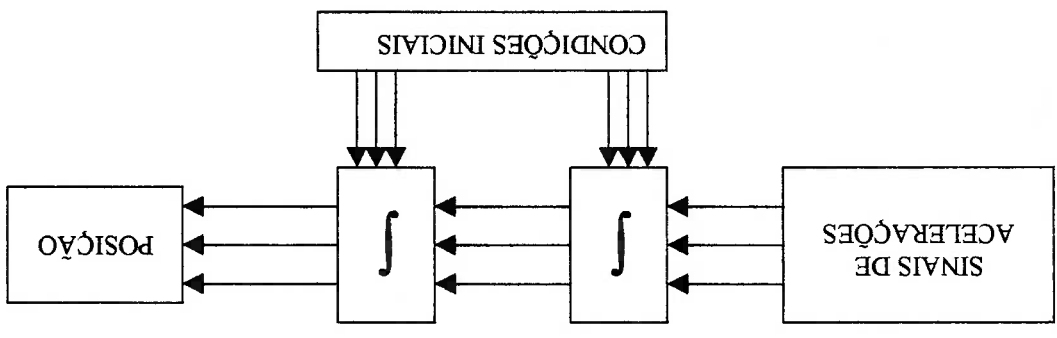


Figura 4.1 – Esquema do duplo integrador.

4.1.2. O FILTRO DE KALMAN

O algoritmo do filtro de Kalman utilizado pode ser derivado a partir daquele descrito no Capítulo 2 (equações 2.8, 2.13 e 2.14). Neste caso, temos os ruídos w_k e e_k independentes, diferentemente do que foi considerado no Capítulo 2; contudo, a derivação do algoritmo utilizado aqui a partir daquele dado anteriormente é bastante simples (as matrizes $C.G^T$ e $G.C^T$ anulam-se). O modelo do sistema é descrito no

Capítulo 3. No caso do filtro de Kalman, a questão chave é definir a matriz de covariância do erro (matriz P). Dependendo da aplicação e do modelo do sistema, a matriz P da equação de Riccati pode não convergir, e então modos alternativos de implementarmos o filtro de Kalman se fazem necessários. Nos testes realizados, não houve problemas quanto à convergência da matriz P, e assim não foi necessário utilizarmos outras formas de implementação do filtro de Kalman. No entanto, como uma das principais causas de instabilidade numérica da equação de Riccati é a assimetria da matriz P, foi utilizada uma técnica de simetrização desta matriz [2]:

$$P = 0.5 (P + P^T)$$

A inicialização da matriz P é outra questão que pode afetar a convergência da equação de Riccati. Neste caso, utilizamos $P_0 = I$ (matriz identidade), mas testes realizados com outras matrizes P_0 sugeriram que nesta aplicação a equação de Riccati converge independentemente da matriz P_0 escolhida. O diagrama esquemático do filtro de Kalman é mostrado abaixo:

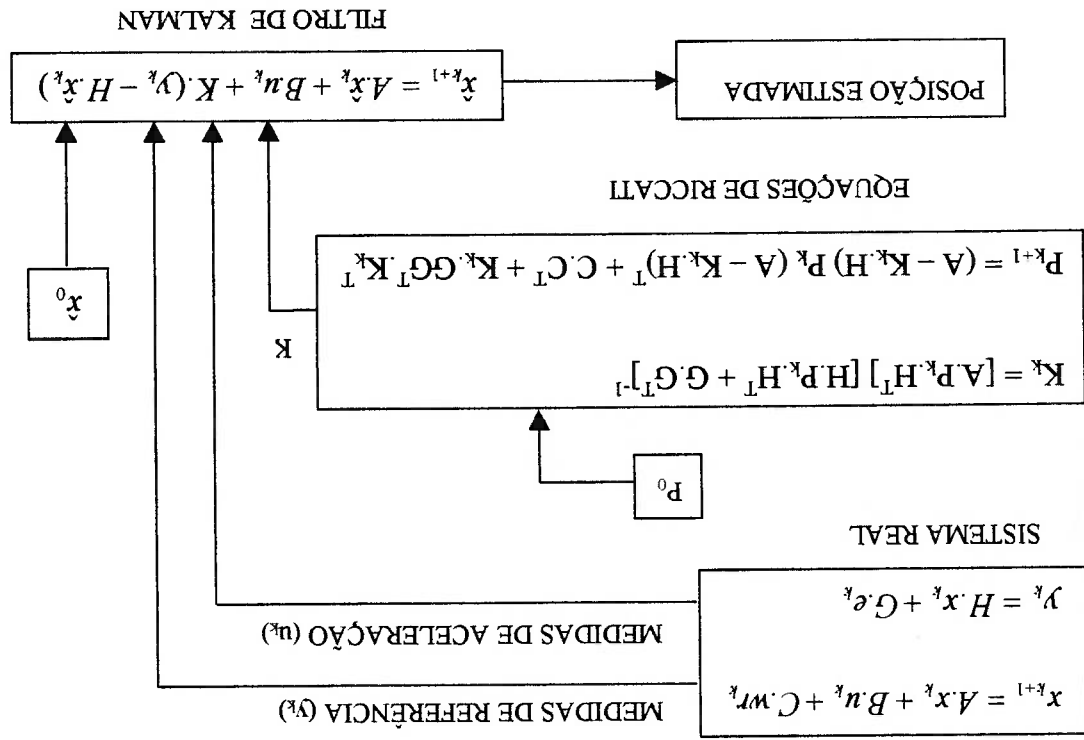


Figura 4.2 – Diagrama esquemático do Filtro de Kalman.

4.1.3. O FILTRO SIR

Conforme visto no Capítulo 2 (seção 2.3.2), para calcularmos os pesos associados às amostras do filtro SIR precisamos conhecer a função densidade de probabilidade $p(y^k / x^k)$. Devido à incerteza na equação do modelo de medição, esta função densidade de probabilidade pode ser escrita do seguinte modo [16]:

$$y^k = H \cdot x^k + G \cdot e^k \Rightarrow e^k = y^k - H \cdot x^k \Rightarrow p(y^k / x^k) = p(e^k) = p(y^k - H \cdot x^k) \quad (4.1)$$

Na equação acima, fizemos a suposição que a matriz G é uma matriz identidade. Assim, tendo y^k e x^k podemos calcular $p(y^k / x^k)$ desde que se conheça a função densidade de probabilidade do erro e^k (pois $p(y^k / x^k) = p(e^k)$). Logicamente, esta função depende sempre da aplicação e das medições que estão sendo utilizadas, mas a princípio consideraremos que ela é uma Gaussiana multivariável: [3]

$$p(e^k) = \frac{1}{\sqrt{2\pi}^n} \exp\left(-\frac{1}{2} \cdot (e^k)^T \cdot e^k\right)$$

(4.2)

Nesta Gaussiana, o vetor de médias é nulo (considera-se que as componentes do vetor de erros e^k têm média zero) e a matriz de covariâncias é a identidade (ou seja, as componentes do vetor e^k não são correlacionadas entre si e têm variância unitária). Temos ainda que n é a dimensão do vetor e^k . Estas simplificações nem sempre são válidas, e em alguns casos podemos ter que ajustar adequadamente o vetor de médias e a matriz de covariâncias desta função Gaussiana.

Para o cálculo das estimativas de estado (x^k) a partir das N amostras geradas pelo filtro ($x^k(i)$, $i = 1, \dots, N$), faremos uma média ponderada [16]:

$$x^k = \sum_{i=1}^N w^k(i) \cdot x^k(i) \quad (4.3)$$

$w_k(t)$ são os pesos associados às amostras $x_k(t)$. Deste modo, podemos escrever o

algoritmo do filtro SIR:

- Inicializar $x_k(t)$ e $w_k(t)$
- For ($i = 1 : N$)
 - { Extrair uma amostra do ruído w_{k-1} ;
 - Calcular as amostras do estado atual do sistema:

$$x_k(i) = A \cdot x_{k-1}(i) + B \cdot u_{k-1} + C \cdot w_{k-1}$$
 - Calcular os pesos associados às amostras:

$$w_k(i) = w_{k-1}(i) \cdot \exp(-0.5 (y_k - H \cdot x_k(i))^T (y_k - H \cdot x_k(i))) / \sqrt{(2\pi)^3}$$
 - Calcular a soma dos pesos: $\text{soma} = \sum_{i=1:N} w_k(i)$
 - Normalizar os pesos:
 - For ($i = 1 : N$)
 - { $w_k(i) = w_k(i) / \text{soma}$ }
 - Calcular o número efetivo de amostras:

$$N_{\text{ef}} = 1 / (\sum_{i=1:N} w_k(i)^2)$$
 - Realizar a reamostragem (se o número efetivo de amostras for baixo):
 - If ($N_{\text{ef}} > \text{Limiar}$)
 - { Executar o algoritmo de reamostragem }
 - Calcular a estimativa do estado atual do sistema:

$$x_k = \sum_{i=1:N} w_k(i) \cdot x_k(i)$$

O número de amostras (N) utilizado foi 100. Este número pode ser considerado baixo para algumas aplicações, mas funcionou bem nos testes automáticos realizados. Além disso, um número de amostras muito elevado pode tornar a execução do filtro lenta, impossibilitando sua aplicação em tempo real, por exemplo, ou encarecendo o custo do sistema (os gastos computacionais seriam elevados, e o ganho adicional de desempenho seria comparativamente pequeno).

Para os filtros SIR e ASIR, o algoritmo de reamostragem utilizado é dado abaixo [1]:

- Inicializar c : $c_1 = 0$;
- Construir o vetor c :
- Construir o vetor c :

$$\text{For } (i = 2 : N) \quad \{ c_i = c_{i-1} + w_k(i) \}$$
- Início da reamostragem: $i = 1$;
- Amostar u_1 a partir de uma distribuição uniforme no intervalo $[1, N]$
- Varredura do vetor c :

For $(j = 1 : N)$

{ $u_j = u_1 + (1 / N) \cdot (j - 1)$

While ($u_j > c_i$ e $i < N$)

{ $i = i + 1$ }

$x_k(i) = x_k(i)$;

$w_k(i) = 1 / N$;

$i(i) = i$;

Este algoritmo é chamado de reamostragem sistemática [1]. Há outros tipos de algoritmos de reamostragem (ver [52], por exemplo), mas aquele que mostrou maior simplicidade e adequação ao problema tratado foi o algoritmo dado acima.

4.1.4. O FILTRO ASIR

A caracterização μ_k do estado x_k dado o estado x_{k-1} foi escolhida como sendo uma amostra da função densidade de probabilidade $p(x_k / x_{k-1})$. Os pesos associados às amostras devem ser calculados de acordo com a equação (2.48). Nesta equação, escolhemos as funções densidade de probabilidade $p(y_k / x_k)$ e $p(y_k / \mu_k)$ como sendo Gaussianas multivariáveis, com vetor de médias nulo e matriz de covariância unitária (nos testes realizados, esta hipótese simplificada funcionou bem), conforme visto na seção 4.1.3.

O algoritmo do filtro ASIR é o seguinte:

- Inicializar $x_k(i)$ e $w_k(i)$;
- For $(i = 1 : N)$
 - { Extrair uma amostra do ruído w_{k-1} ;
 - Calcular as amostras da caracterização do estado do sistema:

$$\mu_k(i) = A \cdot x_{k-1}(i) + B \cdot u_{k-1} + C \cdot w_{k-1}$$
 - Calcular os pesos associados a essas amostras:

$$w_k(i) = w_{k-1}(i) \cdot \exp(-0.5 \cdot (y_k - H \cdot \mu_k(i))^T (y_k - H \cdot \mu_k(i))) / \sqrt{(2\pi)^3}$$
 - Calcular a soma dos pesos: $\text{soma} = \sum_{j=1:N} w_k(j)$
 - Normalizar os pesos:
 - For $(i = 1 : N)$
 - { $w_k(i) = w_k(i) / \text{soma}$ }
 - Executar o algoritmo de reamostragem
 - For $(j = 1 : N)$
 - { Extrair uma amostra do ruído w_{k-1} ;
 - Calcular as amostras do estado x_k :

$$x_k(j) = A \cdot x_{k-1}(j) + B \cdot u_{k-1} + C \cdot w_{k-1}$$
 - Calcular os pesos associados a essas amostras:

$$w_k(j) = \frac{\exp(-0.5 \cdot (y_k - H \cdot x_k(j))^T (y_k - H \cdot x_k(j)))}{\exp(-0.5 \cdot (y_k - H \cdot \mu_k(j))^T (y_k - H \cdot \mu_k(j)))}$$
 - Calcular a soma dos pesos: $\text{soma} = \sum_{j=1:N} w_k(j)$
 - Normalizar os pesos:
 - For $(j = 1 : N)$
 - { $w_k(j) = w_k(j) / \text{soma}$ }
 - Calcular a estimativa do estado atual do sistema:

$$x_k = \sum_{j=1:N} w_k(j) \cdot x_k(j)$$

No algoritmo acima, os índices $i(j)$ são obtidos através do algoritmo de reamostragem (ver o algoritmo na seção 4.1.3.).

4.1.5. O FILTRO MHIR

A função distribuição proposta $q(\cdot)$ do algoritmo Metropolis Hastings escolhida foi uma função Gaussiana com média $Ax_k + Bu_k$. Nesta função, fazemos $q(y/x) = q(y)$ (como no algoritmo *independence sampler*, visto na seção 2.3.4.1). Desta forma, temos que a função $\alpha(x,y)$ torna-se:

$$\alpha(x,y) = \min \left(1, \frac{p(y)q(x)}{p(x)q(y)} \right), \text{ com } q(x) = \frac{\sqrt{|CC^T|_9}}{\exp(0.5(x - Ax_k - Bu_k)^T (CC^T)^{-1} (x - Ax_k - Bu_k))} \quad (4.4)$$

Nesta equação, $p(x)$ é a distribuição estacionária da cadeia de Markov. No algoritmo MHIR, temos que esta distribuição equivale a $p(x_k / x_{k-1}, y_k)$. Considera-se que esta função seja Gaussiana, e então se pode utilizar a equação (2.37) para calcular valores que a função assume. Assim, tem-se a seguinte função Gaussiana:

$$p(x_k / x_{k-1}, y_k) = \frac{\sqrt{|Cov|}}{\exp(0.5(x_k - m_k)^T (Cov)^{-1} (x_k - m_k))}$$

$$Cov = ((CC^T)^{-1} + H^T (GG^T)^{-1} H)^{-1} \quad (4.5)$$

$$m_k = Cov^{-1} (A x_{k-1} + H^T (GG^T)^{-1} y_k)$$

Além disso, precisamos definir a função $p(x_k, y_k / x_{k-1})$, necessária para o cálculo da variável β do algoritmo MHIR. Aplicando a regra de Bayes, temos:

$$p(x_k, y_k / x_{k-1}) = p(y_k / x_k, x_{k-1}) p(x_k / x_{k-1}) = p(y_k / x_k) p(x_k / x_{k-1}) \quad (4.6)$$

Conforme visto na seção 4.1.3, a função densidade de probabilidades $p(y_k / x_k)$ é obtida através da equação do sistema no espaço de estados discreto, assumindo a seguinte forma:

$$y_k = H x_k + G e_k \Rightarrow e_k = y_k - H x_k \Rightarrow p(y_k / x_k) = p(e_k) = p(y_k - H x_k) \quad (4.7)$$

Considera-se que a função $p(e_k)$ seja uma Gaussiana multivariável, e então:

$$(4.8) \quad p(e_k) = \frac{1}{\sqrt{2\pi}^n} \exp\left(-\frac{1}{2}(e_k^T \cdot e_k)\right)$$

De modo similar pode-se obter a função densidade de probabilidades $p(x_k/x_{k-1})$:

$$(4.9) \quad x_k = A \cdot x_{k-1} + B \cdot u_{k-1} + C \cdot w_{k-1} \Rightarrow w_{k-1} = x_k - A \cdot x_{k-1} - B \cdot u_{k-1} \rightarrow p(x_k/x_{k-1}) = p(w_{k-1}) = p(x_k - A \cdot x_{k-1} - B \cdot u_{k-1})$$

A função $p(w_{k-1})$ também é aproximada por uma Gaussiana, e sua expressão é igual à (4.8). Desta forma, utilizando as equações (4.6), (4.7), (4.8) e (4.9), chegamos a:

$$(4.10) \quad p(x_k, y_k/x_{k-1}) = \frac{\sqrt{(2\pi)^3 \cdot (2\pi)^9}}{\exp[-0.5 \cdot ((y_k - H \cdot x_k)^T (y_k - H \cdot x_k) + (x_k - A \cdot x_{k-1} - B \cdot u_{k-1})^T (x_k - A \cdot x_{k-1} - B \cdot u_{k-1}))]}$$

Então, pode-se escrever o algoritmo MHIR do seguinte modo:

- Amostar um índice i uniformemente a partir do conjunto $\{1, \dots, N\}$;
- Inicialização da Cadeia de Markov: $x_k(1) = \text{num}$; $j = 1$;
- While ($j < N$),

{ Algoritmo Metropolis Hastings :

Extrair uma amostra do ruído do processo w_{k-1} ;

$$\text{am}_k(j) = A \cdot x_{k-1}(j) + B \cdot u_{k-1} + C \cdot w_{k-1} ;$$

Amostar uma variável aleatória uniforme u $[0, 1]$;

If ($u \leq \alpha(x_k(j-1), \text{am}_k(j))$) - calcular α utilizando as equações (4.4) e (4.5),

a distribuição estacionária da cadeia de Markov deve ser $p(x_k/x_{k-1}(j), y_k)$

{ $x_k(j) = \text{am}_k(j)$;

Else { $x_k(j) = x_k(j-1)$; }

Ainda, neste algoritmo é utilizada a função densidade de probabilidade $p(y^k / x^k)$ para calcular os pesos associados às amostras geradas pelo algoritmo Metropolis Hastings – ver equações (4.7) e (4.8). O algoritmo de reamostragem é o mesmo visto anteriormente (seção 4.1.3). Observar que as estimativas de estado são calculadas baseando-se nas amostras obtidas após a iteração 100 (ou seja, após o período de *burn-in* da Cadeia de Markov), para que as amostras utilizadas sejam obtidas após a convergência da Cadeia de Markov. Este número pode não ser o mais adequado; contudo, se para este tipo de aplicação a Cadeia não convergir neste número de iterações, utilizar um número mais alto tornaria a execução do filtro MHIR muito mais lenta se comparada aos outros filtros de partículas. Assim, para efeito de

Pode-se utilizar ainda uma variante do algoritmo MHIR, cuja distribuição de probabilidade estacionária da Cadeia de Markov é $p(x^k / x^{k-1})$ e no qual é utilizada a reamostragem por importância. Para o cálculo de α no algoritmo Metropolis Hastings (função densidade de probabilidade estacionária da Cadeia de Markov), pode-se utilizar uma Gaussiana como aproximação para a função real, $p(x^k / x^{k-1})$ – ver equações (4.8) e (4.9).

- Calcular a estimativa do estado x_k :
$$\frac{\sum_{j=20}^N x^k(j)}{N-20}$$
- If ($u_2 \leq \beta$)
 - { $i_j = i$; }
 - Else { $i_j = i_{j-1}$; }
- $\beta = \min \left(1, \frac{p(x^k(j-1), y^k / x^{k-1}(l))}{p(x^k(j-1), y^k / x^{k-1}(l))} \right)$ - calcular as probabilidades de acordo com a equação (4.10);
- Amostragem uma variável aleatória uniforme $u_2[0, 1]$;
- Amostragem um índice (l) uniformemente a partir do conjunto $\{1, \dots, N\}$;
- $j = j + 1$;

comparação entre os vários filtros de partículas, é necessário que utilizemos condições semelhantes de execução para todos os filtros.

- Inicialização da Cadeia de Markov: $x_k(1)=num$, $w_k(1)=1$;

- For $j = 2 : N$,

{ Algoritmo Metropolis Hastings :

Extrair uma amostra do ruído do processo w_{t-k-1} ;

Calcular um ponto candidato para a Cadeia de Markov:

$$am_k(j) = A \cdot x_{k-1}(j) + B \cdot u_{k-1} + C \cdot w_{k-1};$$

Amostrar uma variável aleatória uniforme u $[0, 1]$;

Calcular a probabilidade de aceitação do ponto candidato:

$$\alpha(x_k(j) | j-1, am_k(j)) = \min(1, \frac{\exp(-0.5(am_k(j) - A \cdot x_{k-1}(j) - B \cdot u_{k-1})^T (am_k(j) - A \cdot x_{k-1}(j) - B \cdot u_{k-1}))}{\exp(-0.5(x_k(j) - 1) - A \cdot x_{k-1}(j) - B \cdot u_{k-1})^T (CC^T)^T (am_k(j) - A \cdot x_{k-1}(j) - B \cdot u_{k-1}))})$$

Calcular o próximo estado da Cadeia de Markov:

If $(u \leq \alpha(x_k(j)-1), am_k(j)))$

{ $x_k(j) = am_k(j)$;

Else { $x_k(j) = x_k(j-1)$; }

Calcular o peso associado ao estado:

$$w_k(j) = w_{k-1}(j) \cdot \exp(-0.5 (y_k - H \cdot x_k(j))^T (y_k - H \cdot x_k(j))) / \sqrt{(2\pi)^3}$$

- Calcular a soma dos pesos: soma = $\sum_{i=1}^{100:N} w_k(i)$

Normalizar os pesos:

For $(i = 1 : N)$

{ $w_k(i) = w_k(i) / \text{soma}$ }

- Calcular o número efetivo de amostras:

$$N_{ef} = 1 / (\sum_{i=1}^{100:N} w_k(i)^2)$$

- Realizar a reamostragem (se o número efetivo de amostras for baixo):

If $(N_{ef} < \text{Limiar})$

{ Executar o algoritmo de reamostragem }

- Calcular a estimativa do estado atual do sistema:

$$x_k = \sum_{i=1}^{100:N} w_k(i) \cdot x_k(i)$$

4.2. DESCRIÇÃO DO TESTE AUTOMOTIVO

Foram realizados 4 testes: 2 automotivos, 1 metroviário e o teste do pig inercial. Nesta seção será descrito o procedimento para a realização dos testes automotivos. O teste metroviário será descrito na seção 4.4, e o teste do pig será descrito na seção 4.5. Nos testes automotivos, a Central Inercial foi fixada no chão de um carro. Este carro descreveu então um percurso em circuito fechado na Cidade Universitária (USP – cidade de São Paulo), enquanto a Central Inercial coletava dados de aceleração e rotação angular de seus sensores através do software GyrovieW (da Crossbow, adquirido juntamente com a Central). O software realiza um pré-processamento dos dados coletados pelos sensores, compensa erros determinísticos e aplica fatores de escala, fornecendo os dados na unidade adequada, além de realizar uma calibração nos sensores. O GyrovieW também grava um log dos dados coletados em formato .txt, o que facilita sua posterior leitura no MATLAB. A taxa de amostragem utilizada foi de 50 amostras por segundo. O procedimento correto para a realização de um teste automotivo é o seguinte:

1. Primeiramente, deve-se montar a Central Inercial VG700AA próxima ao centro de gravidade do carro e conectá-la à porta serial (RS-232) de um notebook. A alimentação da Central pode ser feita através da bateria do automóvel (12 V).

2. Ligar a Central Inercial através do GyrovieW e deixá-la parada por 10 minutos. O motor do carro deve estar desligado, já que ele pode causar vibrações de alta frequência, interferindo assim na calibração correta da Central.

3. Zerar os sensores da Central (através do GyrovieW), ainda com o motor desligado (procedimento de calibração).

4. Ligar o motor do carro.

5. Começar a coleta de dados através do GyrovieW. Iniciar a realização do percurso automotivo.

Ao final do percurso, o carro foi parado no mesmo local de onde havia partido. Desta forma, para verificarmos a consistência das trajetórias reconstruídas pelos estimadores basta analisarmos o formato desta trajetória e se o seu ponto final está próximo do ponto de partida. Nos testes realizados, a velocidade do carro foi mantida constante (no primeiro teste automotivo ela foi mantida em 20 km/h, e no segundo teste automotivo ela foi mantida em 30 km/h). Assim, as medidas de referência utilizadas nos filtros para estimar a posição do veículo foram as suas velocidades (no referencial RPY), sempre constantes.

4.3. RESULTADOS EXPERIMENTAIS DO TESTE AUTOMOTIVO

No primeiro teste automotivo (neste teste o carro realizou uma trajetória em circuito fechado, ou seja, foi dada 1 volta no estacionamento da Escola Politécnica) foram coletadas as seguintes acelerações (já transformadas para o referencial NED), rotações angulares e velocidades de referência:

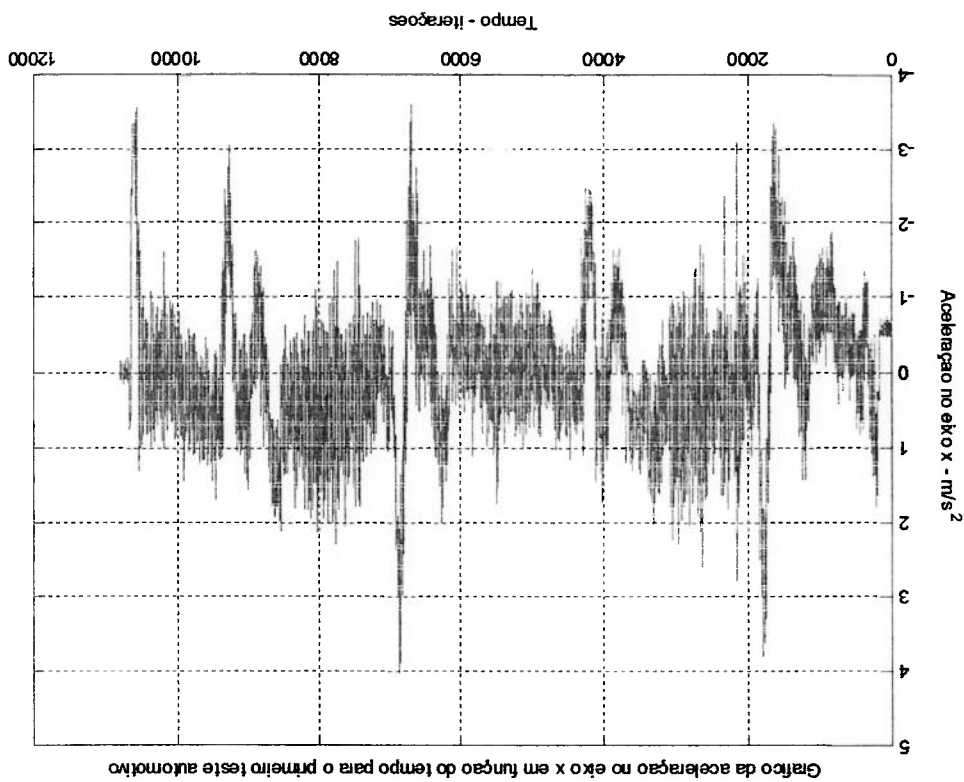


Figura 4.3 – Gráfico da aceleração no eixo x para o primeiro teste automotivo.

Figura 4.5 – Gráfico da aceleração no eixo z para o primeiro teste automotivo.

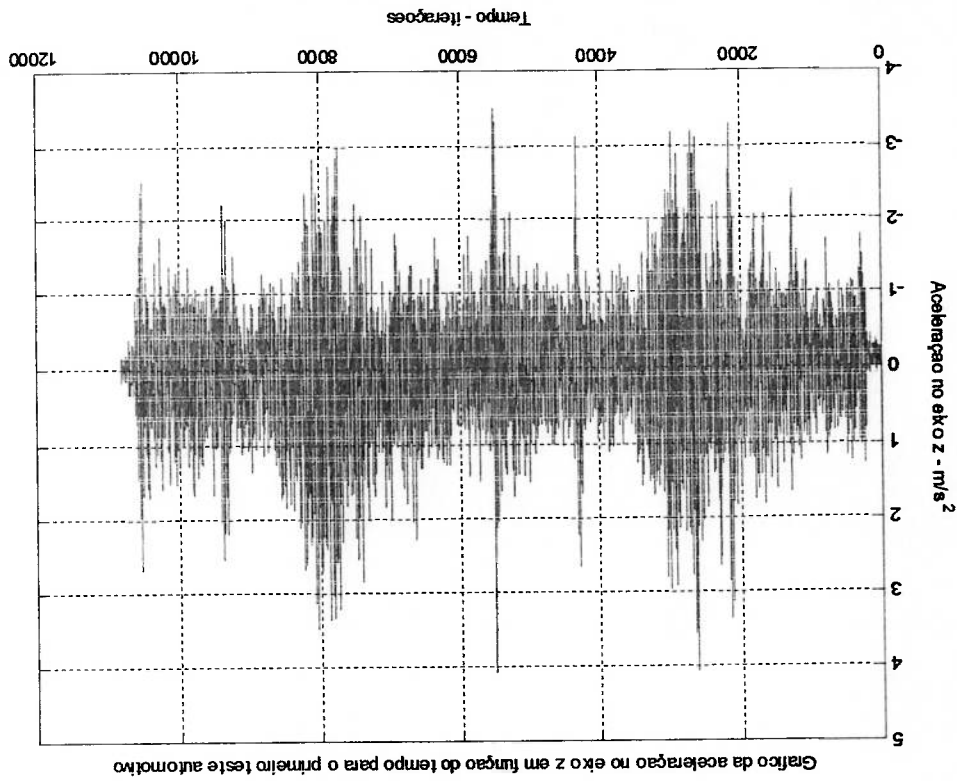


Figura 4.4 – Gráfico da aceleração no eixo y para o primeiro teste automotivo.

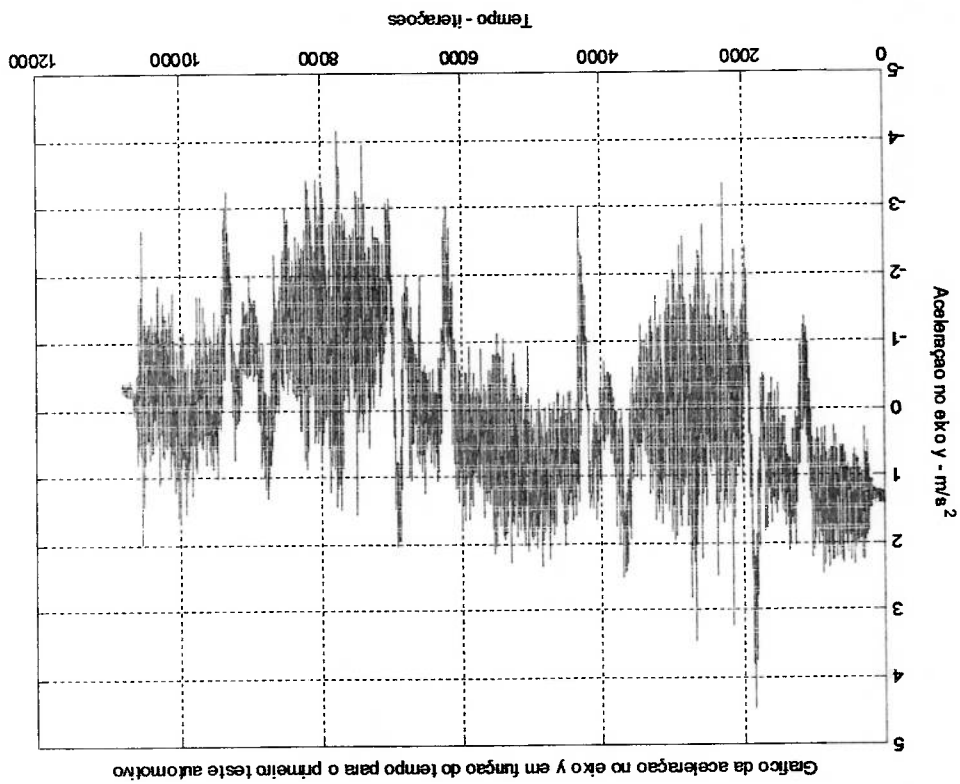


Figura 4.7 – Gráfico da referência de velocidade de velocidade no eixo x para o primeiro teste.

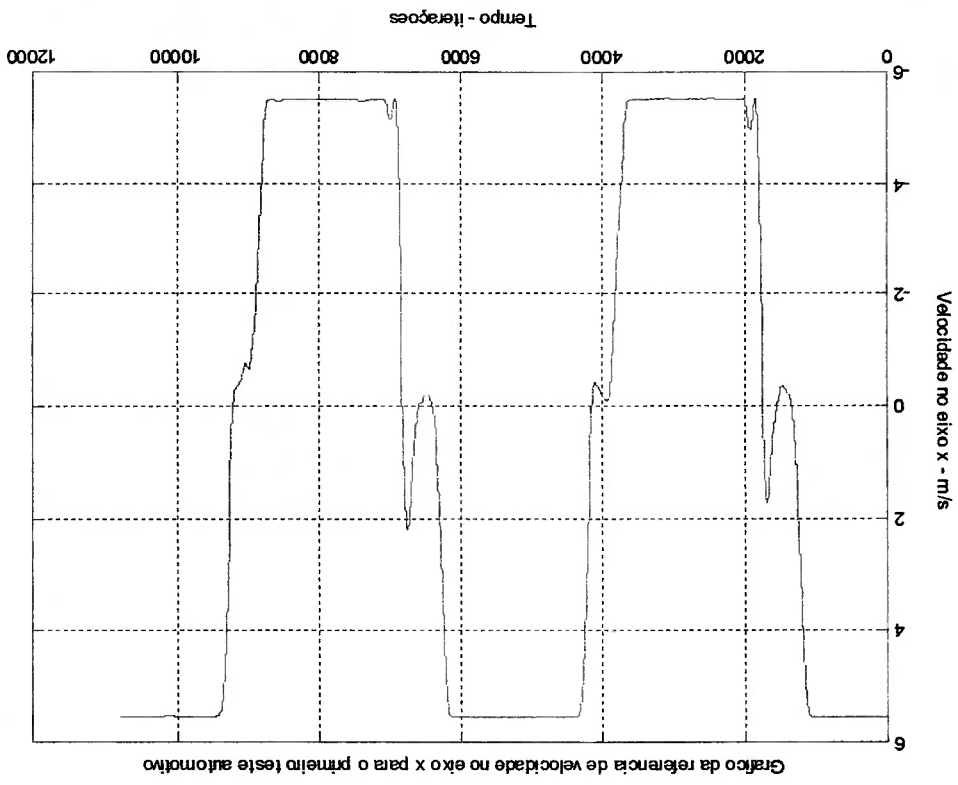


Figura 4.6 – Gráfico da velocidade de rotação (eixo z) para o primeiro teste.

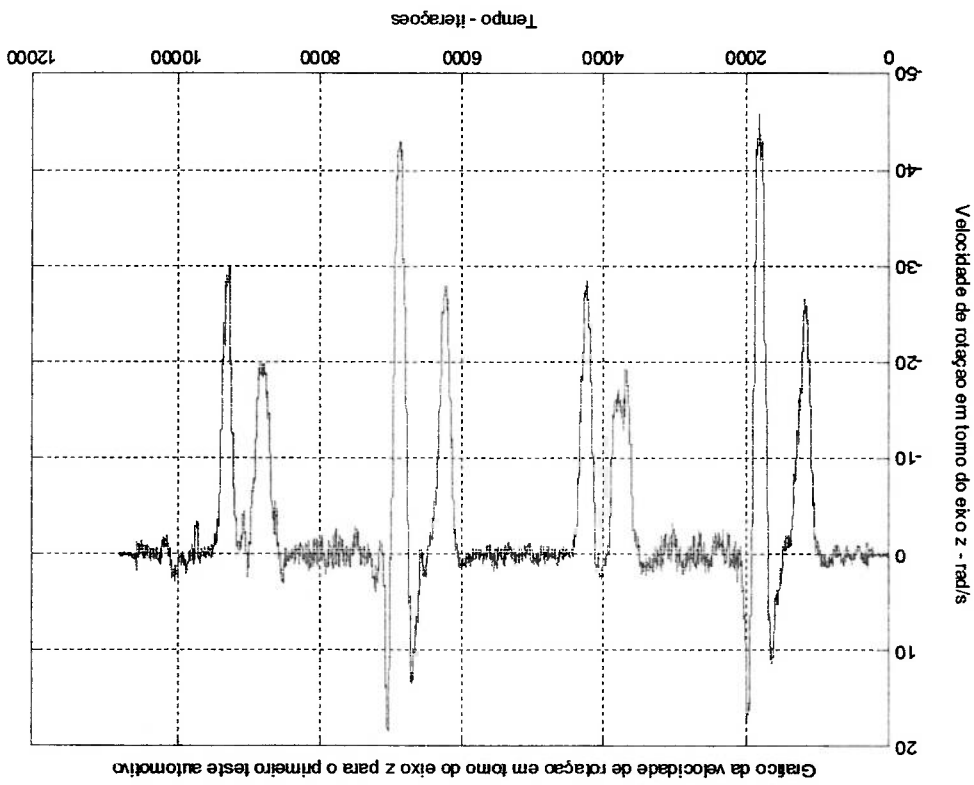


Figura 4.9 – Gráfico da referência de velocidade de velocidade no eixo z para o primeiro teste.

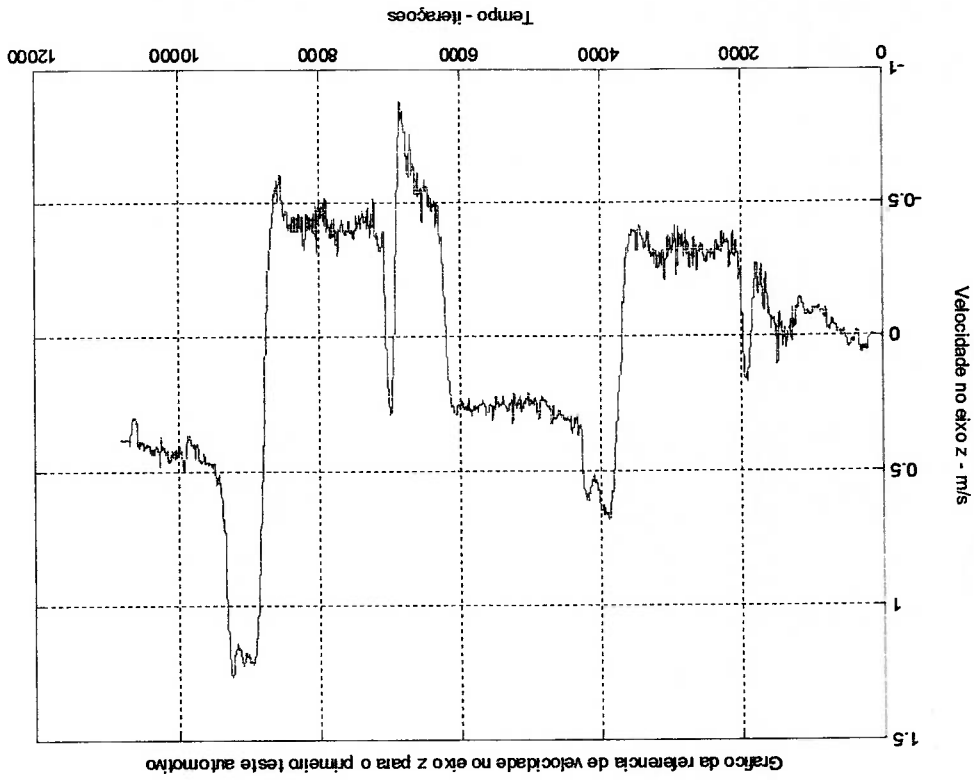
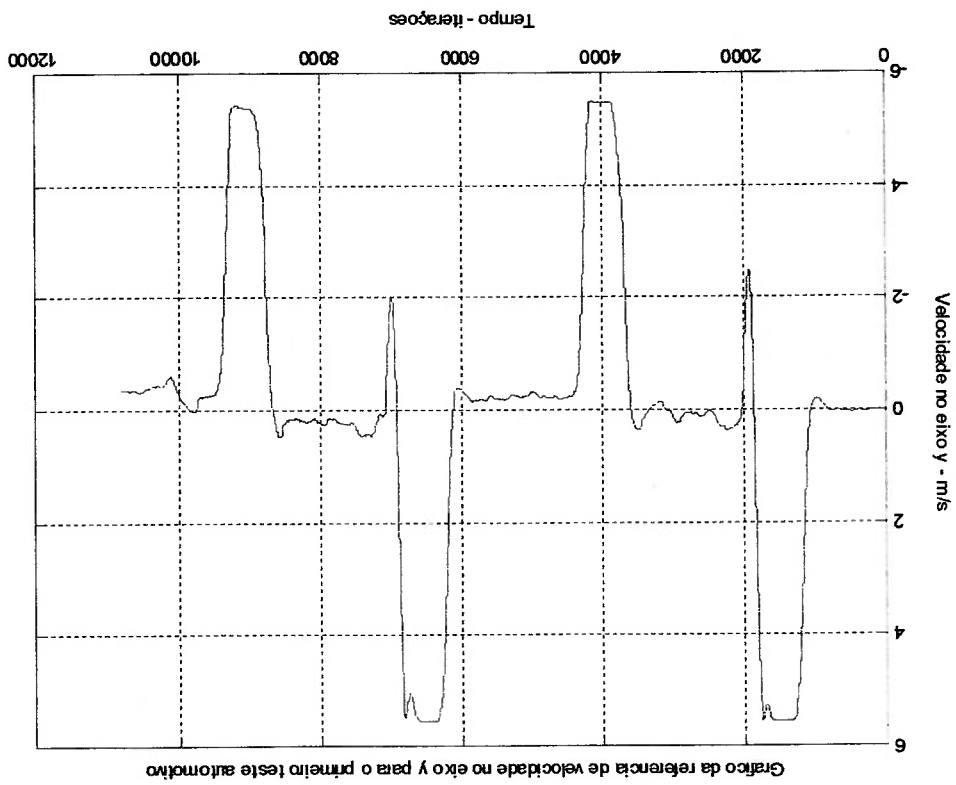


Figura 4.8 – Gráfico da referência de velocidade de velocidade no eixo y para o primeiro teste.



Conforme podemos observar nos gráficos acima, o carro movimentou-se quase que exclusivamente no plano XY (foram realizados apenas alguns pequenos movimentos em outros planos) – assim, as leituras dos acelerômetros x e y descrevem o movimento do carro, enquanto que a leitura do acelerômetro z apresenta apenas ruídos de medição. A leitura do giroscópio z descreve, deste modo, as curvas realizadas pelo carro (o carro praticamente não realizou rotações em torno dos eixos x e y). Pode-se reparar ainda que as leituras dos acelerômetros apresentam alto nível de ruído (os acelerômetros utilizados são considerados *low cost*, de acordo com as suas especificações de ruído e bias).

4.3.1. OS RUIDOS DO PROCESSO

O ruído de medição dos acelerômetros utilizados não é Gaussiano. Para verificarmos isto, basta coletarmos uma amostra deste ruído da seguinte forma: deixamos a Central Inercial parada sobre uma mesa de desempenho nivelada (para que os acelerômetros x e y não meçam componentes da aceleração da gravidade) e coletamos medidas dos 3 acelerômetros. Com as devidas correções, as leituras deveriam indicar zero (pois a Central está parada). O que foi medido desta forma será, então, ruído de medição dos acelerômetros. Uma amostra deste tipo foi coletada durante aproximadamente 3 minutos (a uma taxa de 50 amostras por segundo), e as figuras 4.10 e 4.11 mostram os ruídos dos acelerômetros x e y plotados no tempo. Para verificarmos que eles não são Gaussianos, basta plotarmos um histograma de cada um deles, conforme apresentado nas figuras 4.12 e 4.13 – fica claro, assim, que as distribuições destes ruídos não são Normais.

sendo assim, a estimação de posição utilizando o filtro de Kalman pode não ser a melhor alternativa, já que este filtro trata todos os ruídos do processo como se fossem Gaussianos. Uma melhor opção seria utilizar os filtros de partículas, nos quais poderíamos utilizar diretamente as amostras de ruído coletadas e transformados para o referencial adequado (ver algoritmos nas seções 4.1.3 e 4.1.4). Utilizando este ruído real do processo, os filtros de partículas poderiam fornecer melhores resultados no que diz respeito à estimação de posição se comparados com o filtro de Kalman.

Figura 4.11 – Gráfico do ruído de aceleração do acelerômetro y.

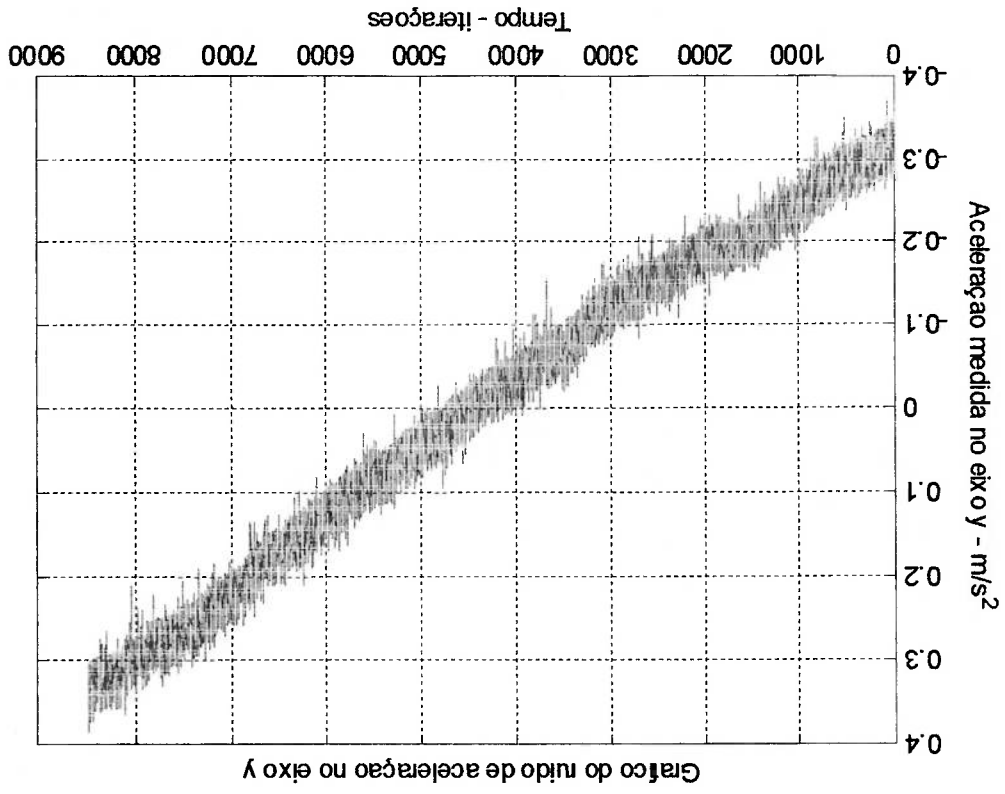


Figura 4.10 – Gráfico do ruído de aceleração do acelerômetro x.

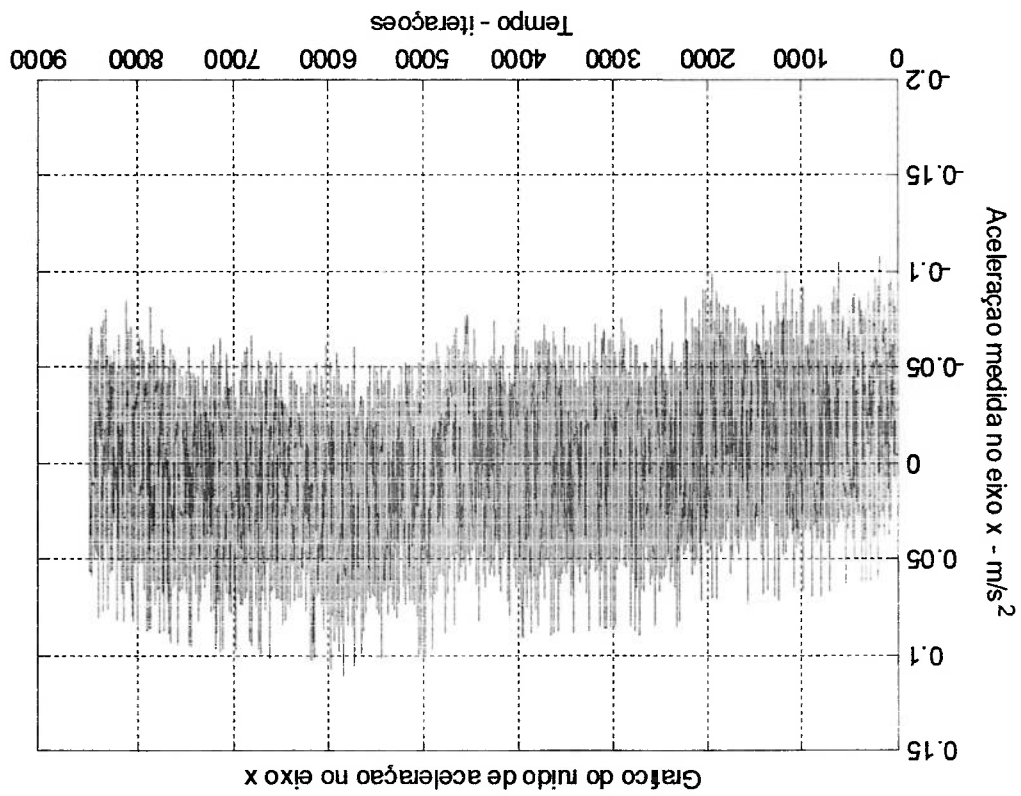


Figura 4.13 – Histograma do ruído de aceleração do acelerômetro y.

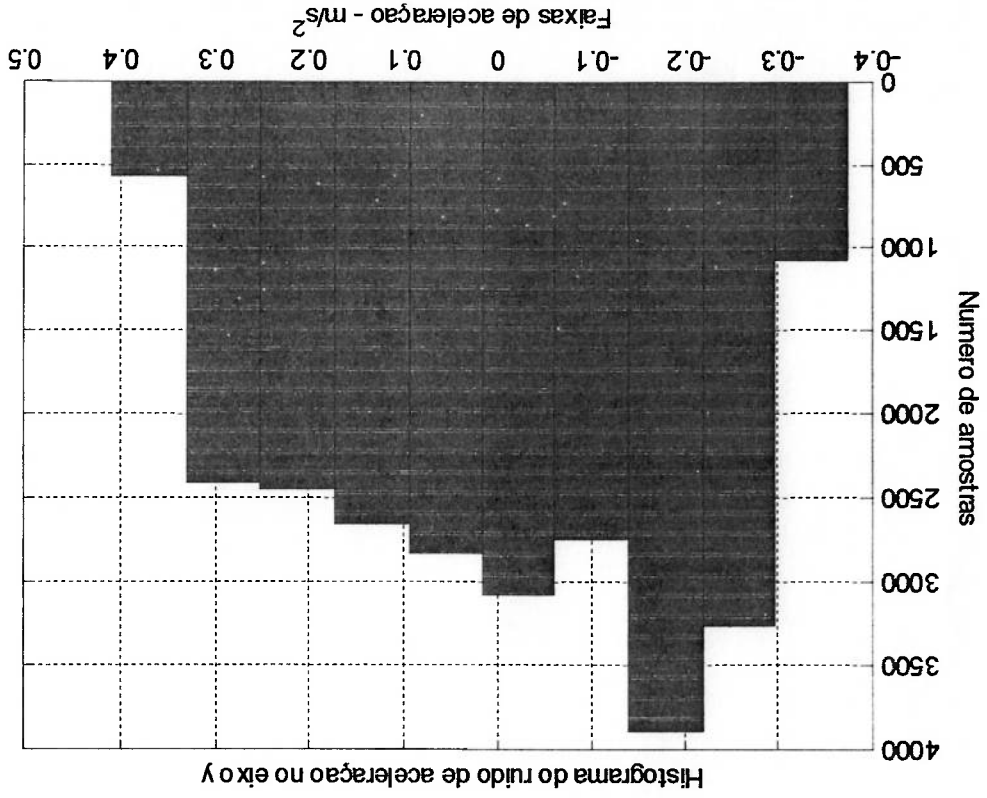
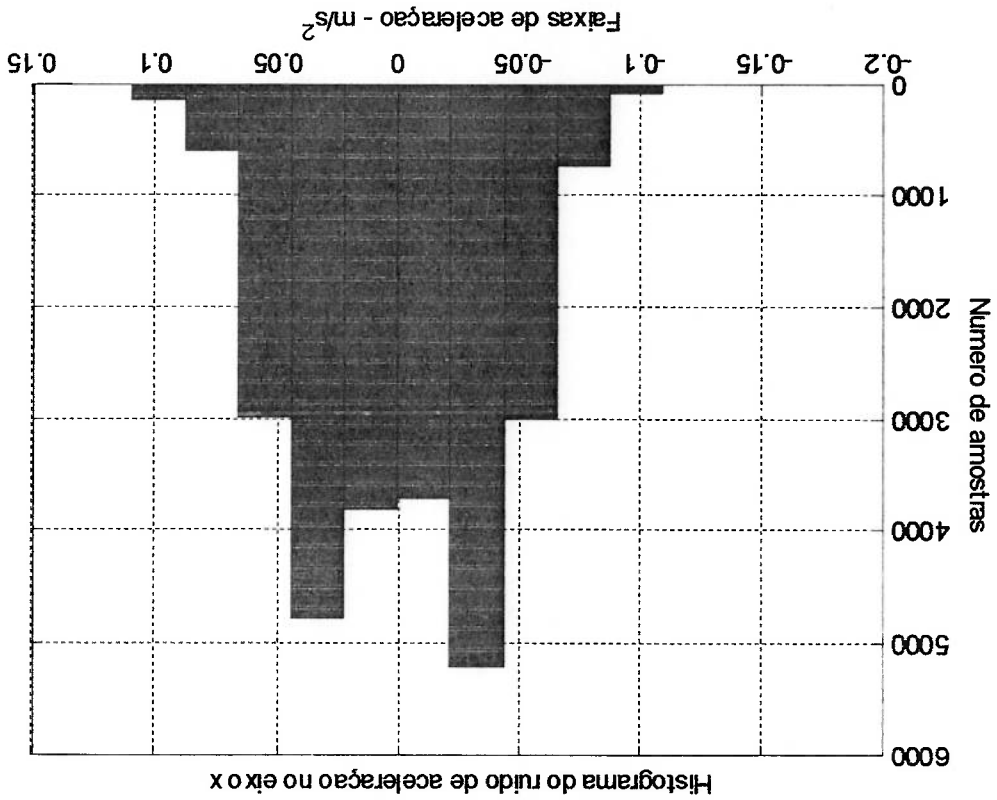


Figura 4.12 – Histograma do ruído de aceleração do acelerômetro x.



4.3.2. ESTIMAÇÃO DE POSIÇÃO UTILIZANDO O DUPLO INTEGRADOR

Utilizando o duplo integrador (seção 4.1.1), foi estimada a seguinte trajetória descrita pelo automóvel (no plano XY):

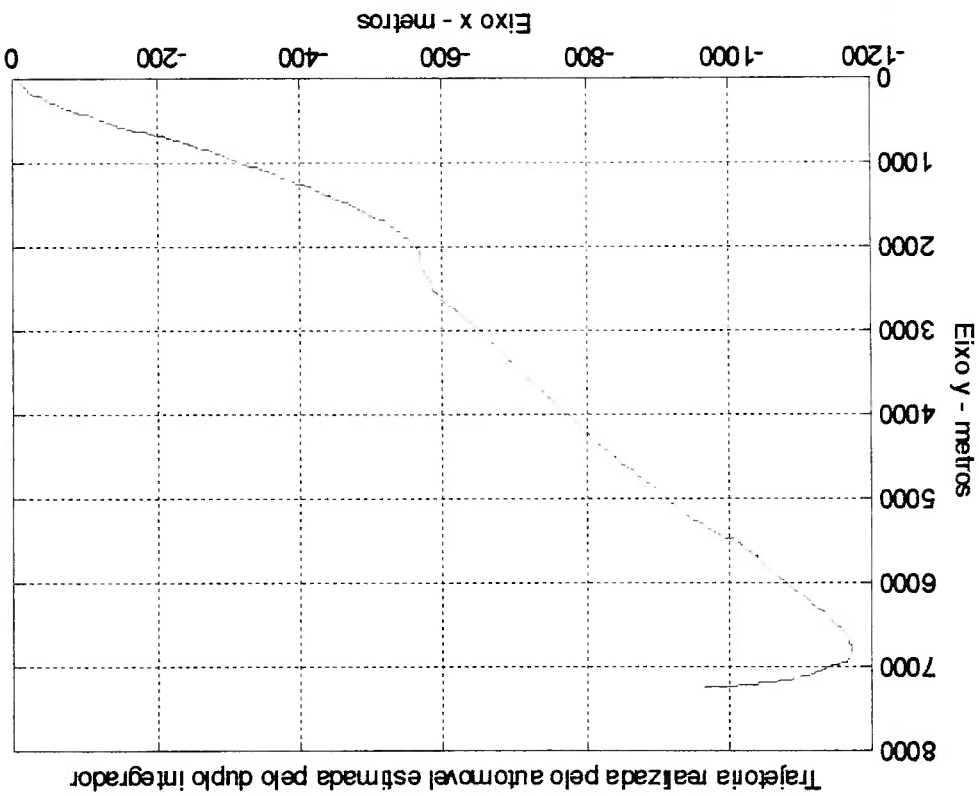


Figura 4.14 – Trajetória do automóvel no plano XY estimada pelo duplo integrador.

Pode-se perceber que o erro de estimação de posição é muito elevado, uma vez que o ponto final da trajetória automática deveria coincidir com o ponto de partida. Os erros de posição obtidos no ponto final da trajetória foram os seguintes:

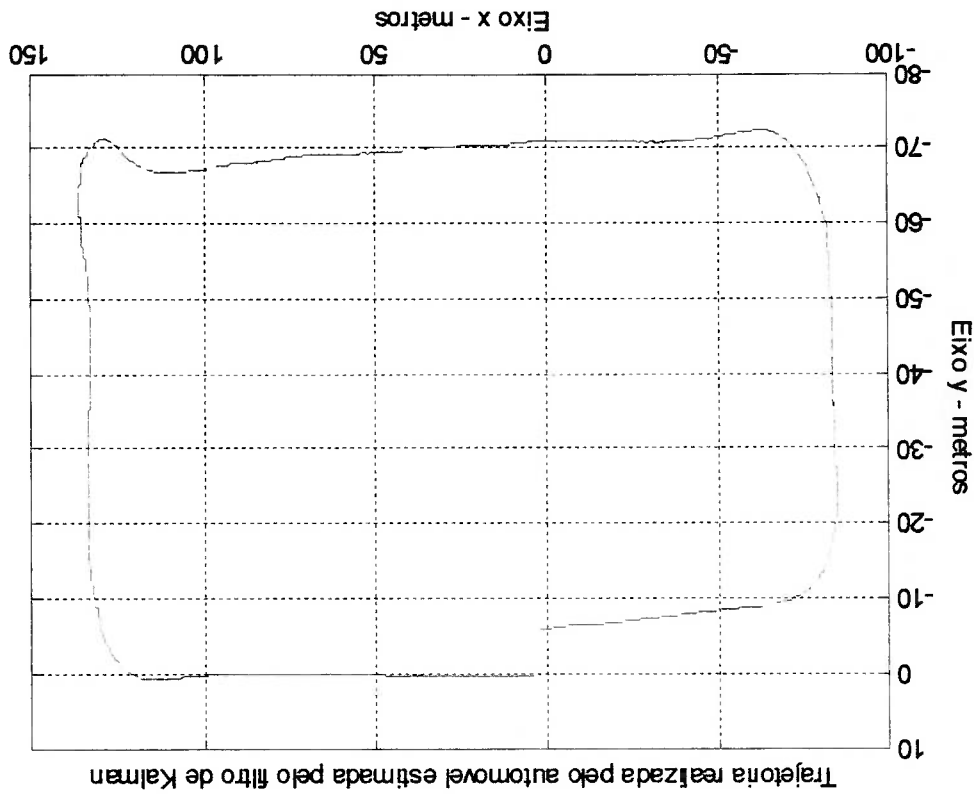
Erro x – 960,5 m
 Erro y – 7253,2 m
 Erro z – 803,8 m

Como o teste durou aproximadamente 4 minutos e o espaço total percorrido foi de 1300 m, estes erros são da ordem de 70% para o eixo x e de 550% para o eixo y. Isso

faz com que seja inviável utilizarmos as medidas dos sensores da Central Inercial sem uma referência (como é feito no duplo integrador). Assim, a utilização dos filtros de Kalman e de partículas é essencial neste caso.

4.3.3. ESTIMAÇÃO DE POSIÇÃO UTILIZANDO O FILTRO DE KALMAN

A medida de referência utilizada foi a velocidade do carro (que foi medida através do velocímetro). Como o velocímetro mede a velocidade do carro no sistema RPY, é necessário que façamos uma transformação de coordenadas para o sistema NED. As velocidades dos 3 eixos no referencial NED estão apresentadas nas figuras 4.7, 4.8 e 4.9. Executando o programa que implementa o filtro de Kalman no MATLAB, obtivemos a seguinte trajetória:



Pode-se observar que a trajetória estimada é extremamente compatível com a trajetória realizada pelo veículo, e os erros de posição foram os seguintes:

Erro x - 3.09 m
 Erro y - 4.09 m
 Erro z - 2.52 m

Estes erros são da ordem de 0.74% para o eixo y e 0.55% para o eixo x. Assim, o ganho de desempenho obtido em relação ao duplo integrador é inquestionável. O erro de aceleração no eixo x (que é um ruído de medição, ou ainda *bias*) estimado pelo filtro de Kalman pode ser apreciado na figura 4.16. Plotando-se um histograma deste ruído estimado, percebe-se que ele é aproximadamente Gaussiano (o que faz sentido, uma vez que o filtro de Kalman considera que todos os ruídos do processo são Gaussianos).

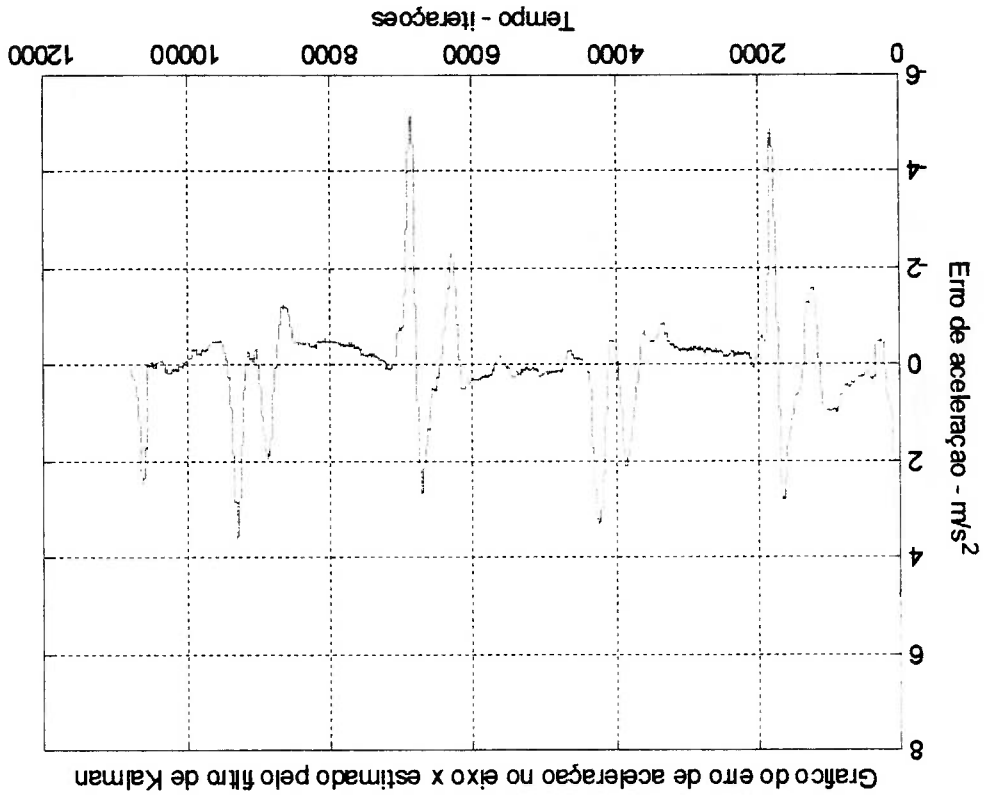


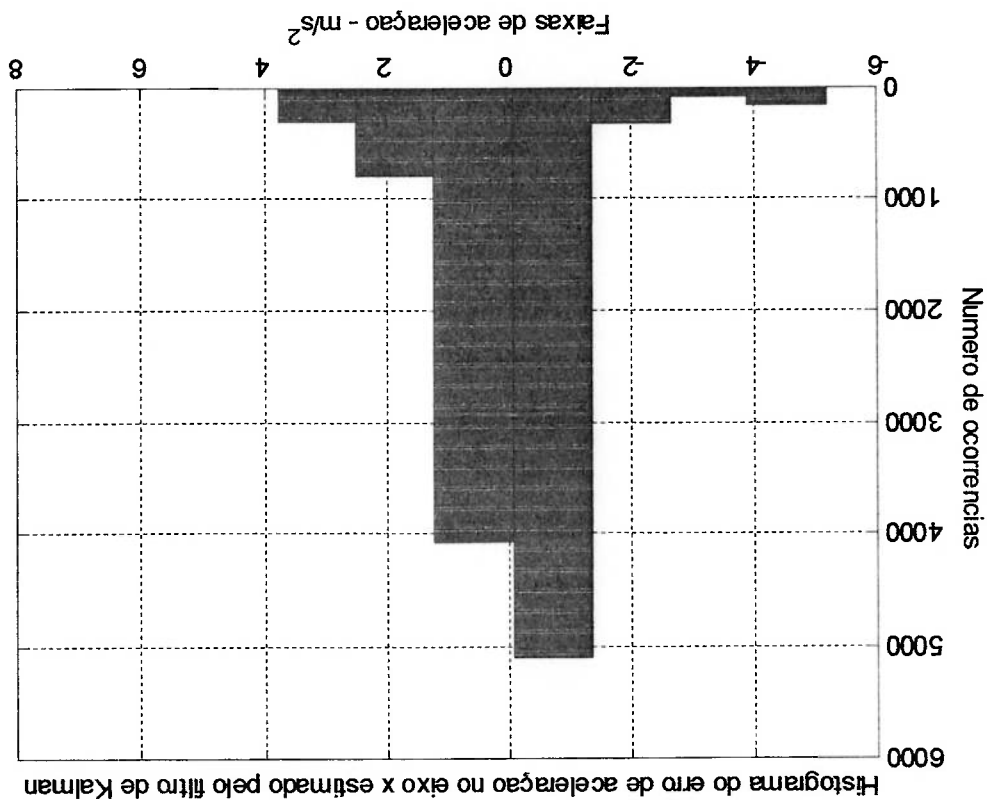
Figura 4.16 - Gráfico do erro de aceleração no eixo x estimado pelo filtro de

Kalman.

A medida de referência utilizada foi a mesma do filtro de Kalman (velocidade do automóvel lida no velocímetro), e foram utilizadas 100 partículas a cada iteração do algoritmo SIR. Antes de analisarmos o desempenho do filtro SIR, façamos a validação do programa utilizado para executá-lo. Para isso, será feita uma comparação entre os resultados do filtro SIR e os resultados do filtro de Kalman do seguinte modo: foi visto no capítulo 2 que os filtros de partículas são uma generalização do filtro de Kalman para os casos em que temos ruídos não - gaussianos e sistemas não - lineares. Como o modelo do nosso sistema no espaço de estados é linear, se amostrarmos um ruído Gaussiano no filtro SIR ele deve fornecer o mesmo resultado do filtro de Kalman (que funciona para casos em que o sistema é linear e o ruído é Gaussiano). Plotando as trajetórias obtidas pelo filtro de Kalman e pelo filtro SIR com ruído Gaussiano, temos o seguinte resultado (figura 4.18):

4.3.4. ESTIMAÇÃO DE POSIÇÃO UTILIZANDO O FILTRO SIR

Figura 4.17 – Histograma do erro de aceleração (x) estimado pelo filtro de Kalman.

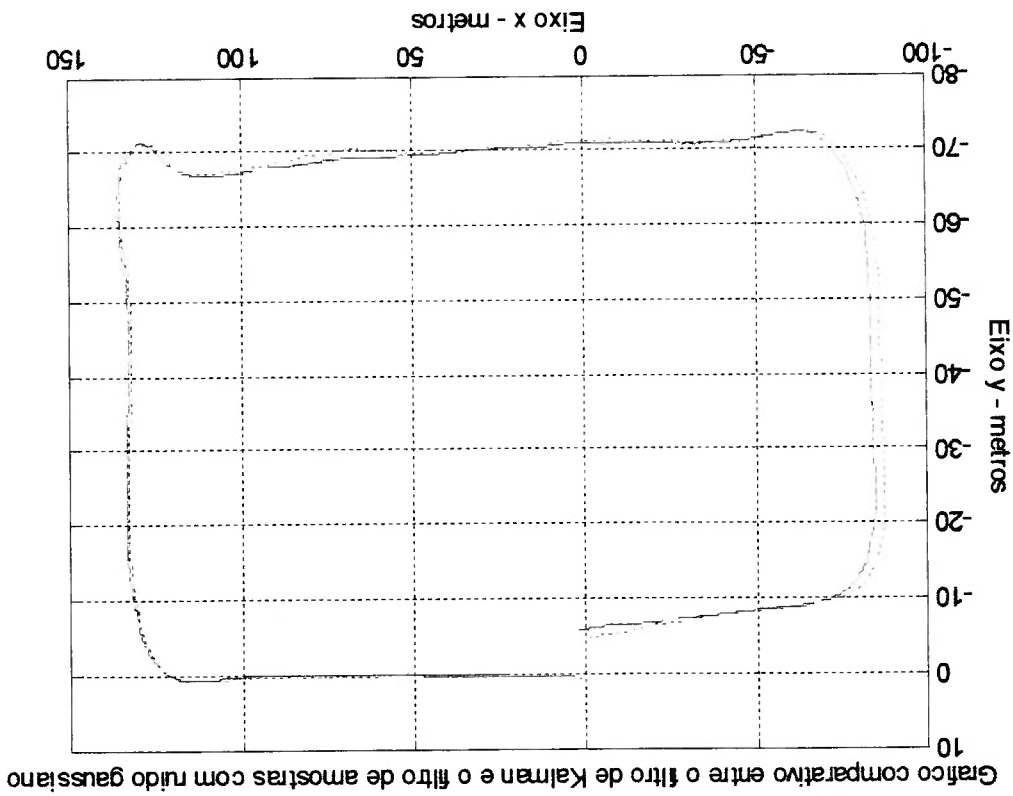


Estes erros são da ordem de 0.18% para o eixo x e 0.1% para o eixo y. Indubitavelmente, o desempenho do filtro SIR é superior ao do filtro de Kalman (já que o ponto final da trajetória automotiva deve coincidir com o ponto inicial). Foram

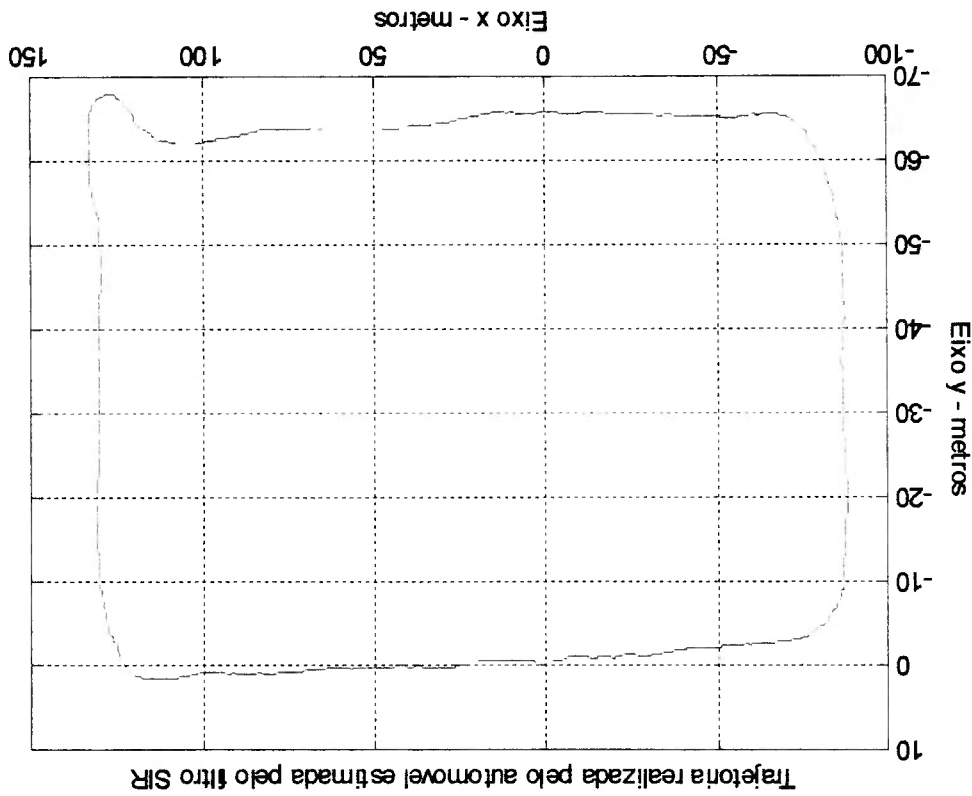
Erro x - 1 m
Erro y - 0.57 m
Erro z - 6.6 m

Podemos verificar que as trajetórias estimadas pelo filtro de Kalman e pelo filtro SIR são quase idênticas, o que valida o programa utilizado para executarmos o filtro SIR. Passemos, então, à análise do desempenho do filtro SIR com amostras do ruído real do processo (obtido de acordo com o descrito na seção 4.3.1). A trajetória obtida desta forma está apresentada na figura 4.19. Ela é compatível com a trajetória realizada pelo automóvel, e os erros de posição são os seguintes:

Figura 4.18 - Gráfico de trajetórias obtidas pelo filtro de Kalman (contínuo) e pelo filtro SIR com ruído Gaussiano (tracejado).



obtidos erros relativamente baixos, e isso provavelmente se deve ao fato de termos utilizado o ruído real do processo no filtro SIR, enquanto que no filtro de Kalman o ruído é considerado como sendo Gaussiano (o que não ocorre na prática). Uma desvantagem do filtro SIR com relação ao filtro de Kalman é que ele demora mais tempo para ser executado em ambiente MATLAB, o que poderia dificultar o seu uso em aplicações on-line (se utilizarmos um número de amostras adequado à aplicação).



4.3.5. ESTIMAÇÃO DE POSIÇÃO USANDO O FILTRO ASIR

A medida de referência utilizada neste caso é a mesma dos 2 filtros anteriores. A trajetória obtida utilizando-se o filtro ASIR é apresentada na figura 4.20, e ela também é compatível com o percurso realizado pelo automóvel. Assim como no filtro SIR, no filtro ASIR também são utilizadas amostras do ruído real do processo. Os erros de posição obtidos foram:

Erro x – 4.96 m
 Erro y – 5.32 m
 Erro z – 4.59 m

Estes erros foram superiores aos erros apresentados pelo filtro de Kalman, e podemos concluir que o filtro ASIR não funcionou bem neste caso. A razão pode ser a seguinte: conforme visto no capítulo 2, se o ruído do processo for relativamente alto (que é o caso), $\mu_k(i)$ não será uma caracterização adequada para a função densidade de probabilidades $p(x_k / x_{k-1}(i))$ – desta forma, o filtro ASIR fará a reamostragem baseado numa aproximação ruim de $p(x_k / x_{k-1}(i))$, e então o desempenho deste filtro não será bom. Portanto, devido ao nível de ruído deste processo, não é recomendável utilizarmos o filtro ASIR (um resultado melhor até poderia ser obtido, mas para isso seria necessário utilizarmos um número de amostras muito elevado no filtro, o que tornaria a execução do programa extremamente demorada).

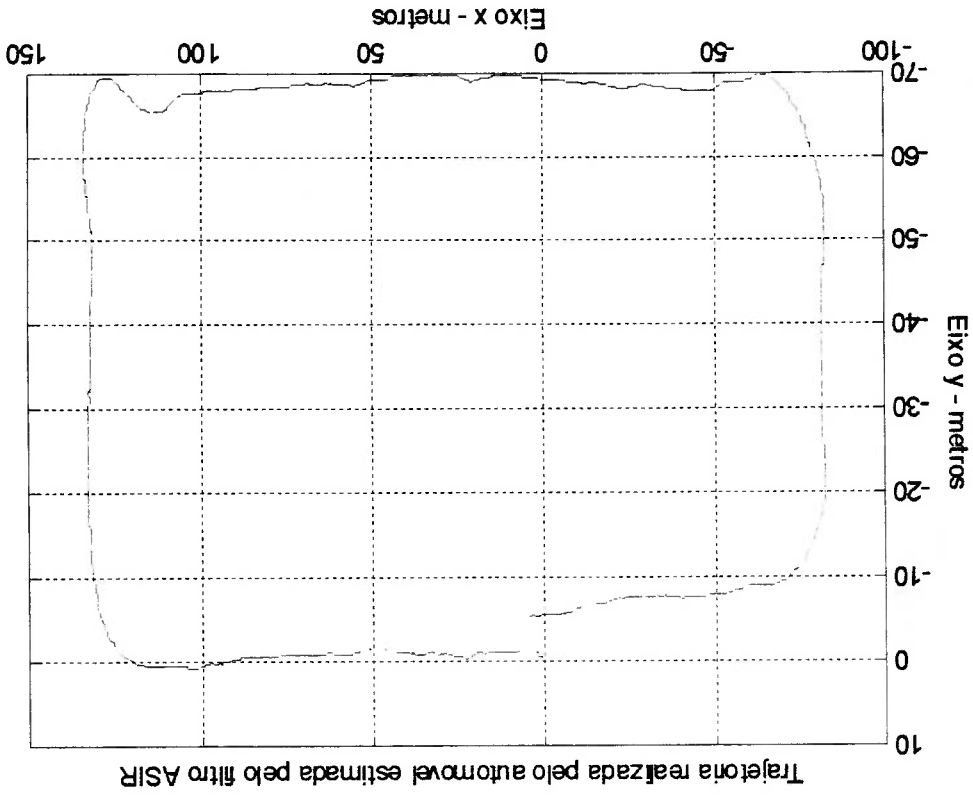


Figura 4.20 – Trajetória do automóvel no plano XY estimada pelo filtro ASIR.

O procedimento utilizado para executar o experimento no Metrô foi semelhante ao do teste automotivo (seção 4.2). A Central Inercial foi fixada numa caixa de madeira ao lado de baterias de 12 V para alimentação da mesma; também foi levado um notebook para realizar a aquisição de dados da Central Inercial e sua calibração. O notebook foi montado sobre a caixa de madeira - esta montagem pode ser apreciada na figura (4.36). Após a inicialização (similar ao procedimento para o teste automotivo) e alinhamento do sistema (conforme visto no capítulo 3), esta estrutura foi colocada no interior de uma das cabines da composição do Metrô que circula

4.4. O EXPERIMENTO METROVIÁRIO

desempenho x velocidade de execução.

filtro de Kalman. A decisão a ser tomada envolveria então o seguinte confronto: amostras utilizado, a execução do filtro SIR pode ser muito mais demorada que a do aplicação nos filtro de partículas). Dependendo da aplicação e do número de Kalman em ambiente MATLAB (utilizando-se um número de amostras adequado à opções por utilizar este filtro: sua execução é mais demorada que a do filtro de No entanto, em aplicações on-line teríamos de considerar outro fator antes de características do sistema (ruído relativamente alto e não Gaussiano) foi o filtro SIR. Conclui-se, deste modo, que o filtro que apresentou o melhor desempenho dadas as utilizadas nos algoritmos do filtro MHIR teria de ser muito elevado).

demorada se comparado aos outros filtros descritos anteriormente (o valor de N aplicações em Navegação Inercial), já que sua execução seria extremamente MHIR mostrou-se totalmente inadequado para este tipo de caso (ou seja, para Markov, o que seria inviável do ponto de vista computacional. Deste modo, o filtro teríamos que utilizar um número de amostras muito elevado para as cadeias de valor utilizado para N foi 500). Para que a convergência fosse obtida, provavelmente Metropolis Hastings para nenhum dos dois filtros MHIR descritos na seção 4.1.5 (o Não foi possível obter a convergência das cadeias de Markov geradas pelo algoritmo

4.3.6. ESTIMAÇÃO DE POSIÇÃO USANDO O FILTRO MHIR

entre as estações Ana Rosa e Vila Madalena, e foram coletadas medidas de aceleração e velocidade de rotação angular durante 1 ciclo de ida e volta completo desta composição do Metrô. Também foram coletadas medidas de referência (velocidades instantâneas do Metrô), da seguinte forma: de 10 em 10 segundos foi anotada a velocidade instantânea dada pelo velocímetro do Metrô (ver figura 4.38). A medição das velocidades foi sincronizada com as medições da Central Inercial (ou seja, a marcação de tempo destas medições foi iniciada ao mesmo tempo), e para que obtivéssemos o mesmo número de amostras para cada tipo de medição foi realizada uma interpolação linear das velocidades lidas no velocímetro do Metrô. As velocidades assim obtidas devem passar por uma transformação de coordenadas (do sistema RPY para o sistema NED) para que elas possam ser utilizadas como medidas de referência nos filtros de estimação. A taxa de amostragem utilizada neste experimento foi de 50 amostras por segundo. Os gráficos das medidas obtidas no experimento metrológico podem ser apreciados a seguir.

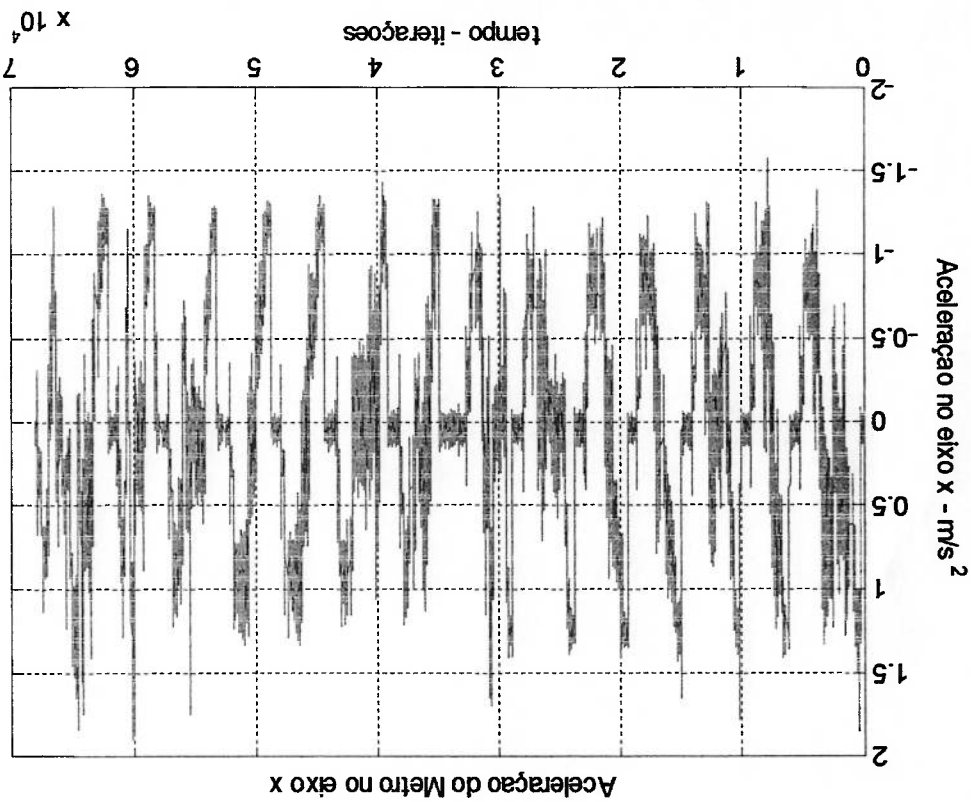


Figura 4.21 – Gráfico da aceleração da composição metrológica no eixo x (sist. RPY)

Figura 4.23 – Gráfico da aceleração da composição metrorviária no eixo z (sist. RPY)

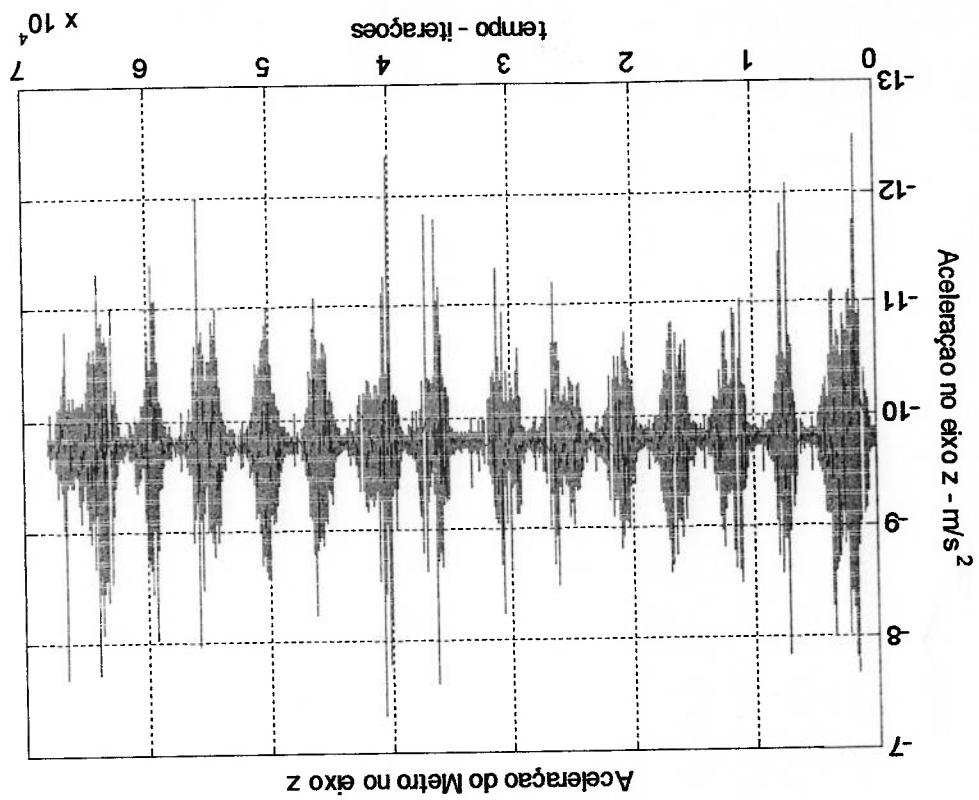


Figura 4.22 – Gráfico da aceleração da composição metrorviária no eixo y (sist. RPY)

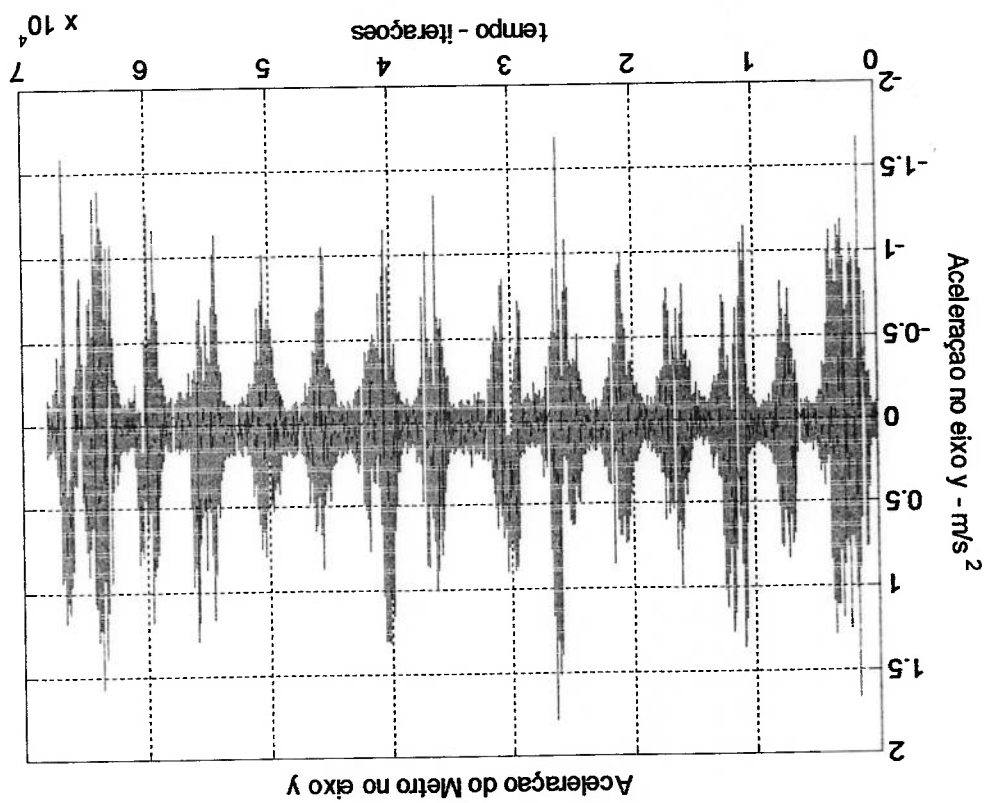


Figura 4.25 – Gráfico da velocidade de rotação angular do Metrô em torno do eixo x.

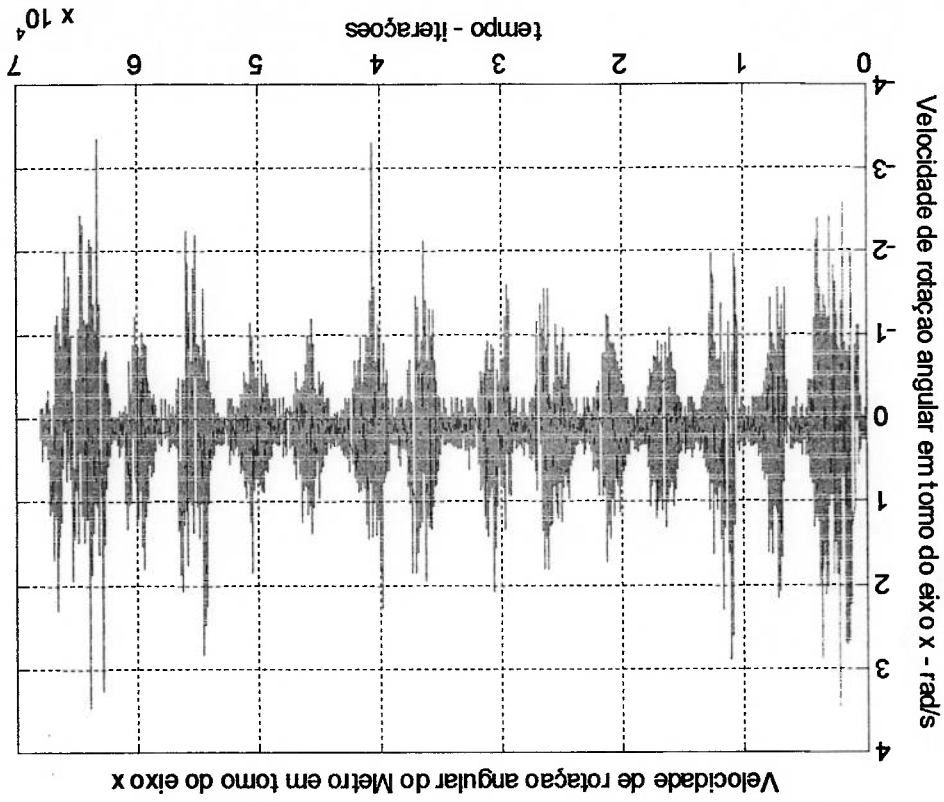


Figura 4.24 – Gráfico da velocidade de rotação angular do Metrô em torno do eixo z.

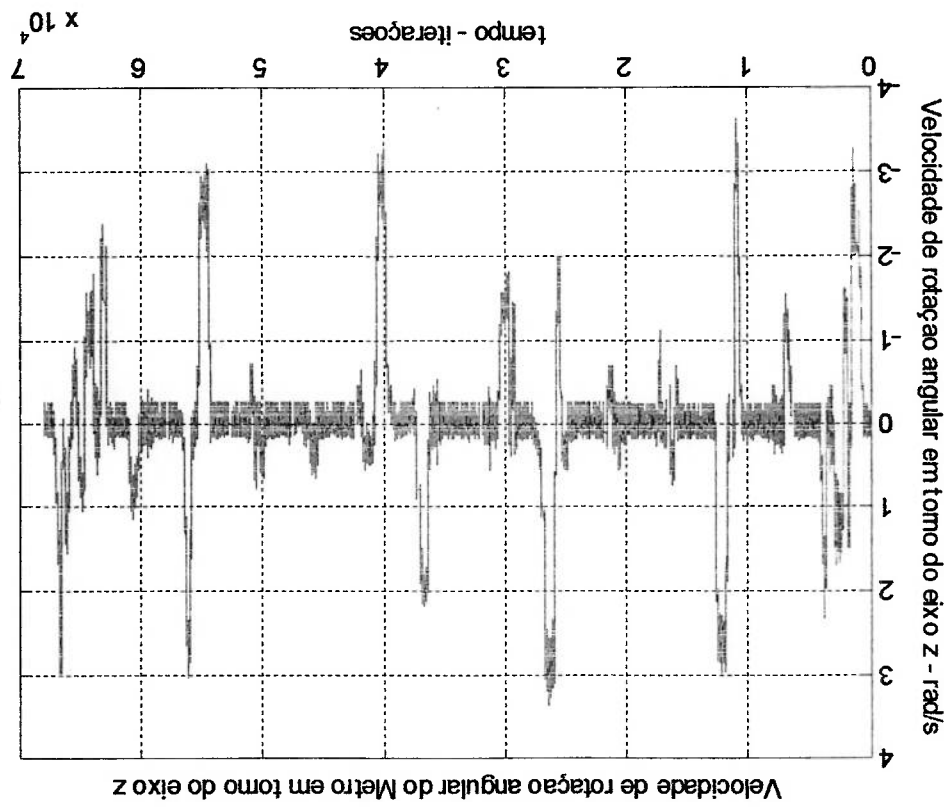


Figura 4.27 – Gráfico da medição de referência (velocidade) do Metrô no eixo x.

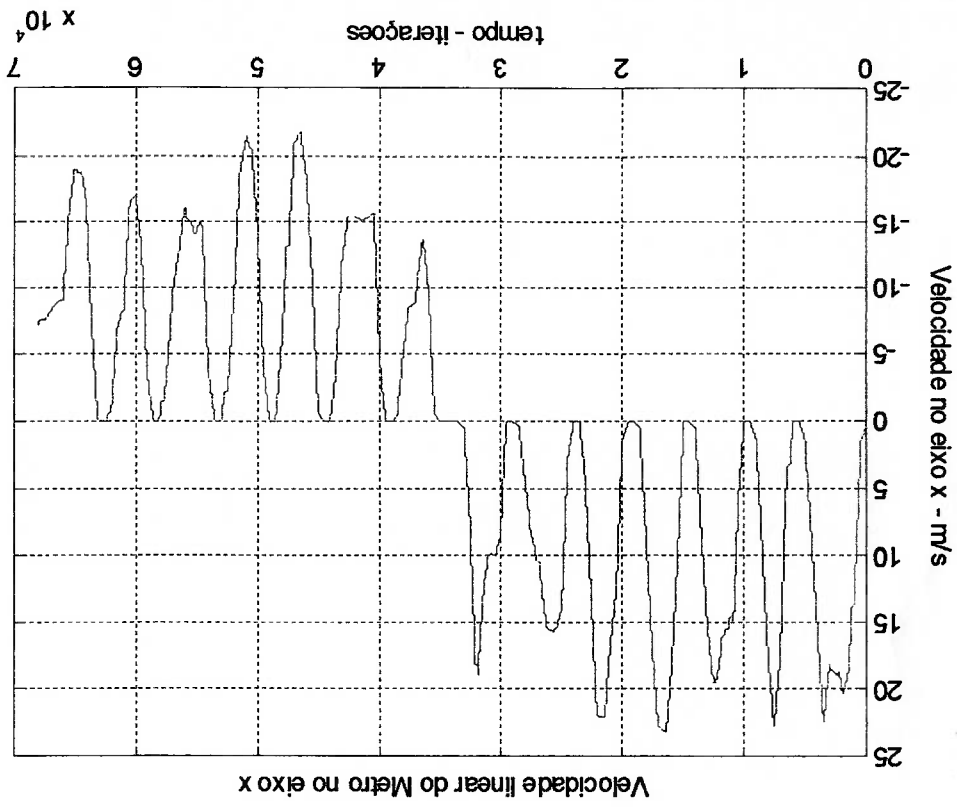


Figura 4.26 – Gráfico da velocidade de rotação angular do Metrô em torno do eixo y.

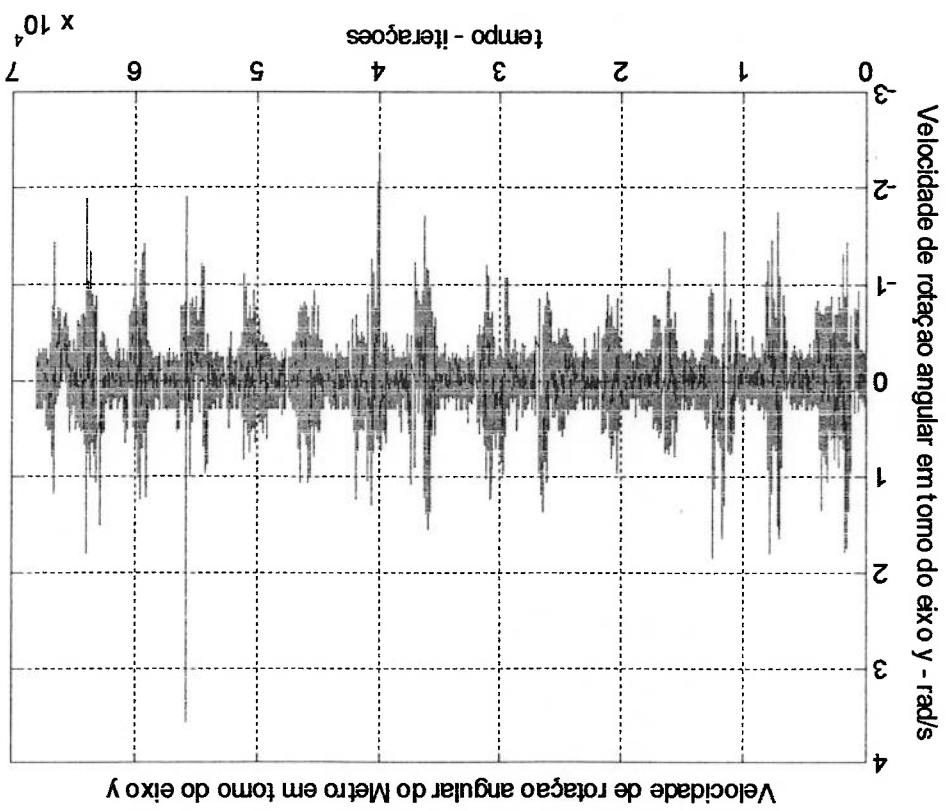


Figura 4.29 – Gráfico da medição de referência (velocidade) do Metrô no eixo z.

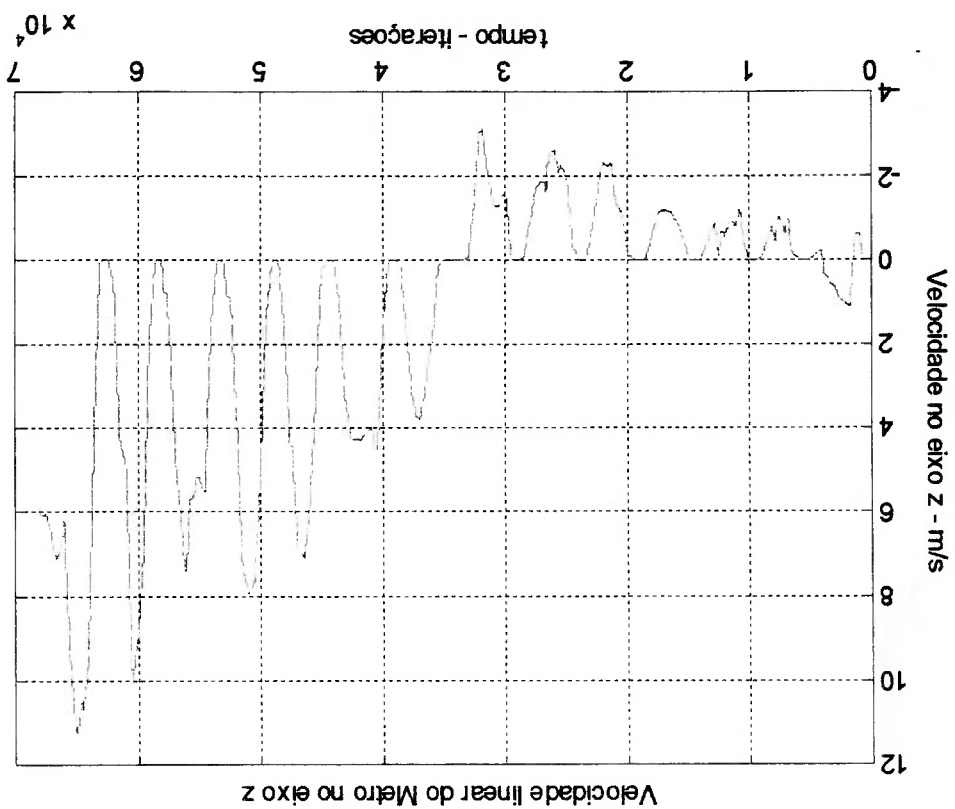
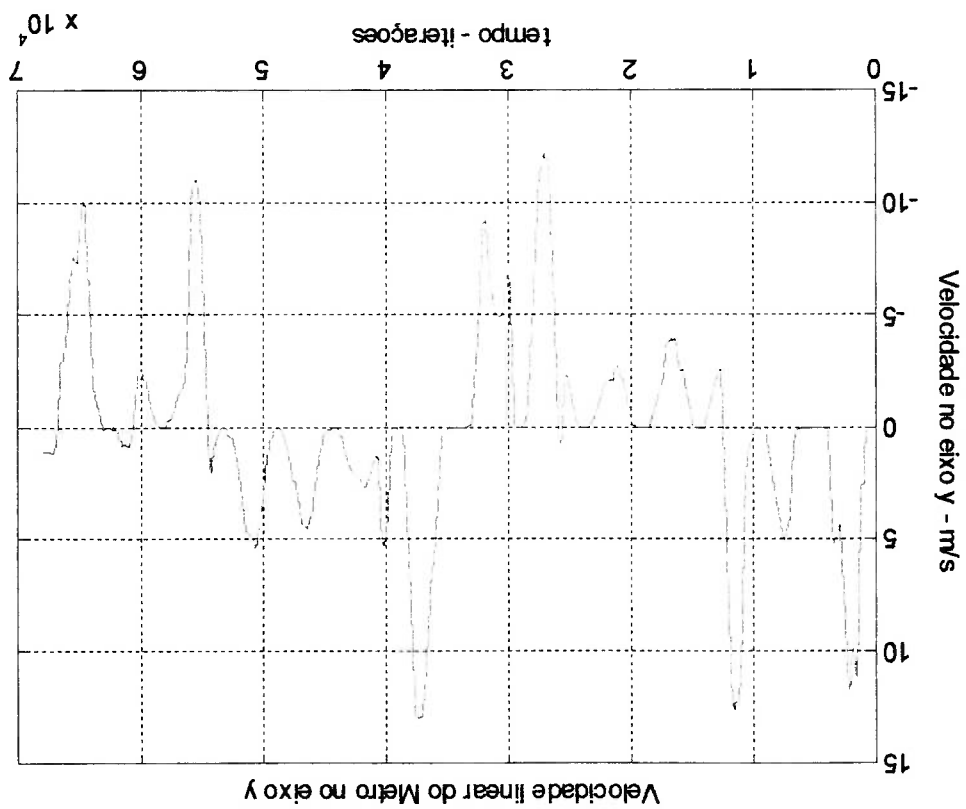


Figura 4.28 – Gráfico da medição de referência (velocidade) do Metrô no eixo y.



Para estimarmos a trajetória descrita pelo Metrô, foi utilizado o modelo dado pelas equações (3.20) e (3.21). A trajetória estimada pelo filtro de Kalman (ver figura 4.2) é a seguinte:

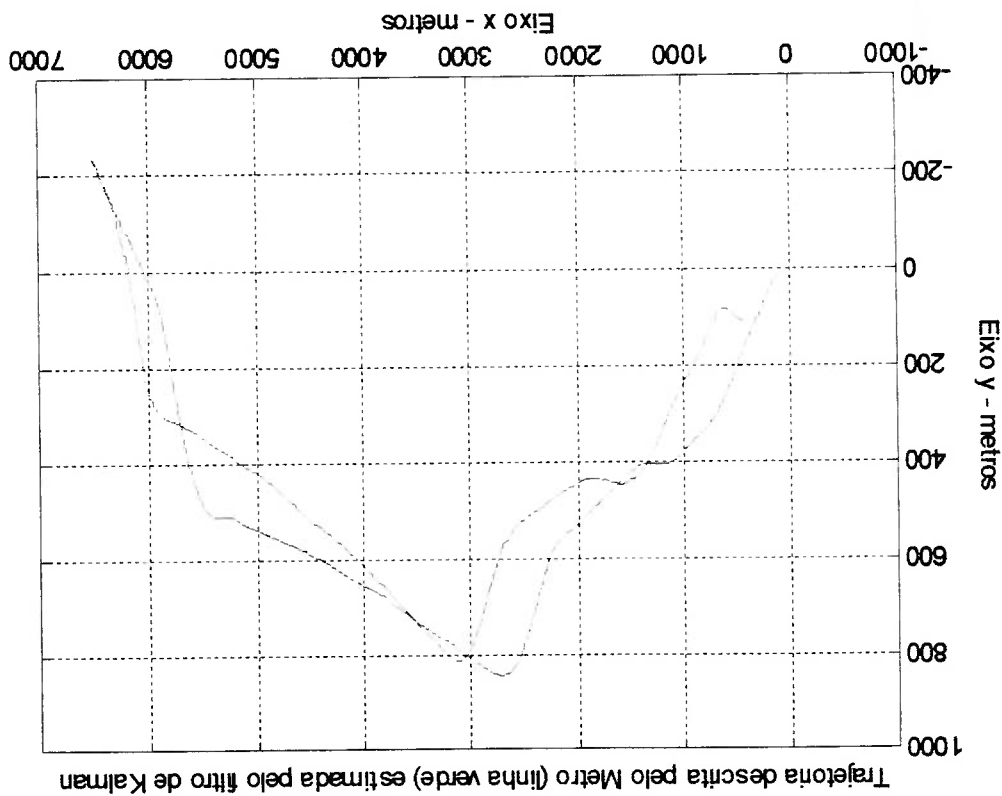


Figura 4.30 – Gráfico da trajetória metroriária (linha verde) estimada pelo filtro de Kalman.

Os erros de estimação ao final do percurso foram (considerando que o Metrô volta exatamente para o mesmo ponto de onde havia partido quando realiza um percurso de ida e volta numa determinada linha):

Erro x – -430 m

Erro y – -108,77 m

Percentualmente, estes erros correspondem a 3,3% para o eixo x e 3% para o eixo y.

Assim, o desempenho do filtro de Kalman foi satisfatório para este experimento.

Os gráficos das posições da composição do Metrô nas coordenadas x e y são apresentados abaixo. Neles, pode-se perceber claramente as transições entre 2 estações do Metrô (o caminho percorrido foi Ana Rosa - Paraiso - Brigadeiro - Trianon Masp - Consolação - Clinicas - Sumaré - Vila Madalena - Sumaré - Clinicas - Consolação - Trianon Masp - Brigadeiro - Paraiso - Ana Rosa).

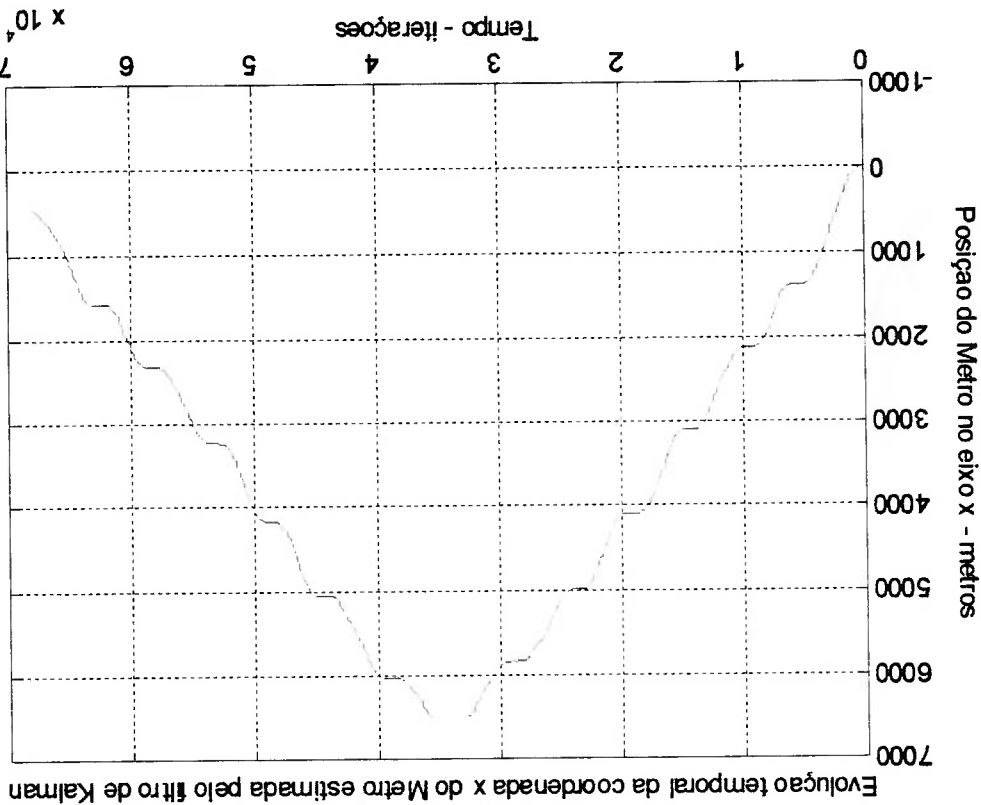


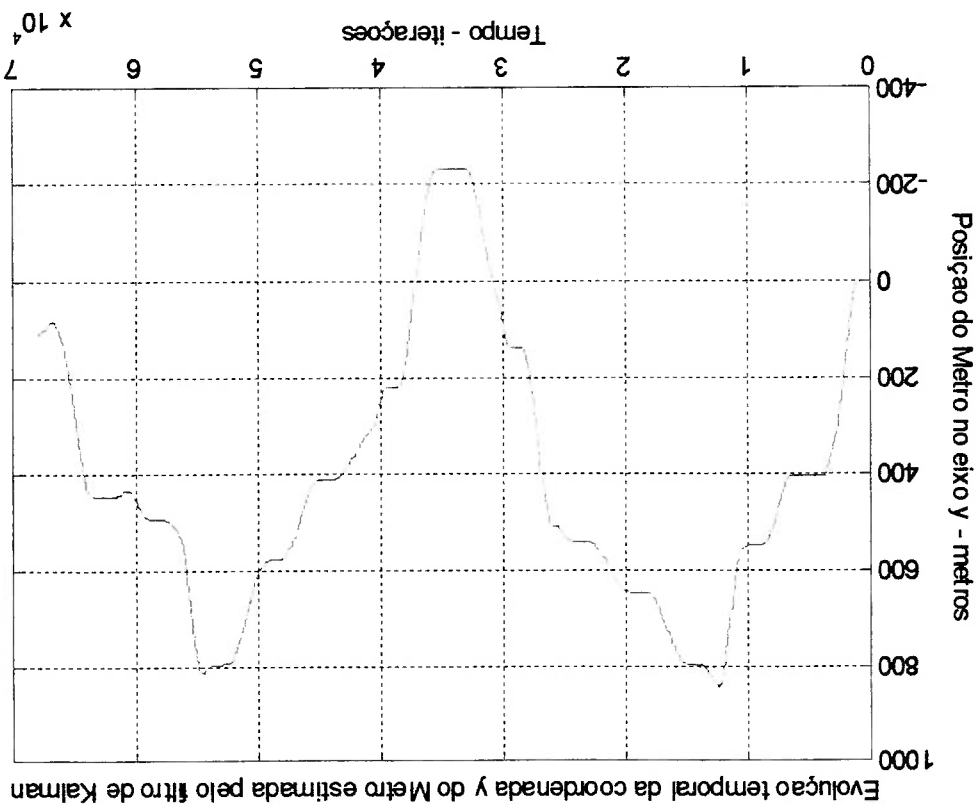
Figura 4.31 - Gráfico da evolução temporal da coordenada x do Metrô estimada através do filtro de Kalman.

Percentualmente estes erros correspondem a 2.11% para o eixo x e 0.12% para o eixo y. Estes erros foram menores que os erros apresentados pelo filtro de Kalman, e a razão disto provavelmente é que o filtro SIR utiliza nas suas estimativas amostras do ruído real do processo (que não é Gaussiano), enquanto o filtro de Kalman supõe que este ruído seja Gaussiano, conforme foi explicado na seção 4.3.4.

Erro x - 275.39
 Erro y - 5.32

A estimação da trajetória metroviária através do filtro SIR foi feita utilizando-se o modelo do sistema dado pelas equações (3.20) e (3.21). O algoritmo implementado no MATLAB é aquele que foi descrito na seção 4.1.3. O número de amostras utilizadas no filtro foi 500. A trajetória estimada é dada na figura 4.33. Os erros de estimação ao final do percurso foram:

Figura 4.32 - Gráfico da evolução temporal da coordenada y do Metrô estimada através do filtro de Kalman.



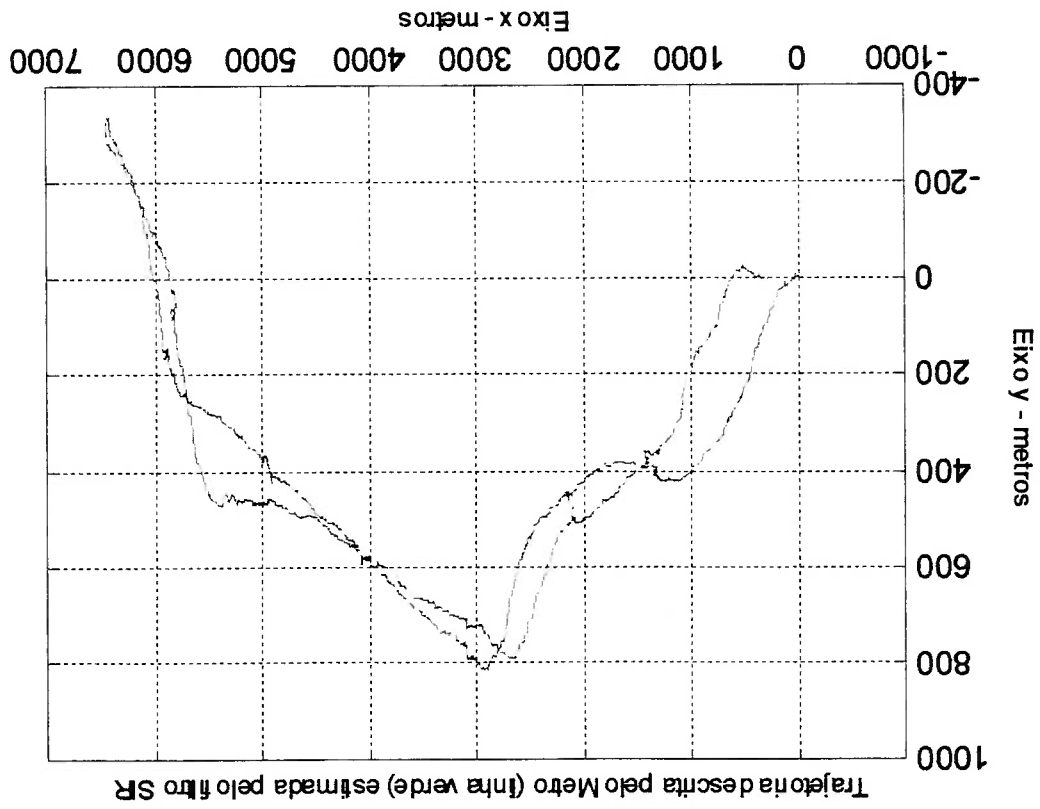


Figura 4.33 – Gráfico da trajetória metrorviária (linha verde) estimada pelo filtro SIR.

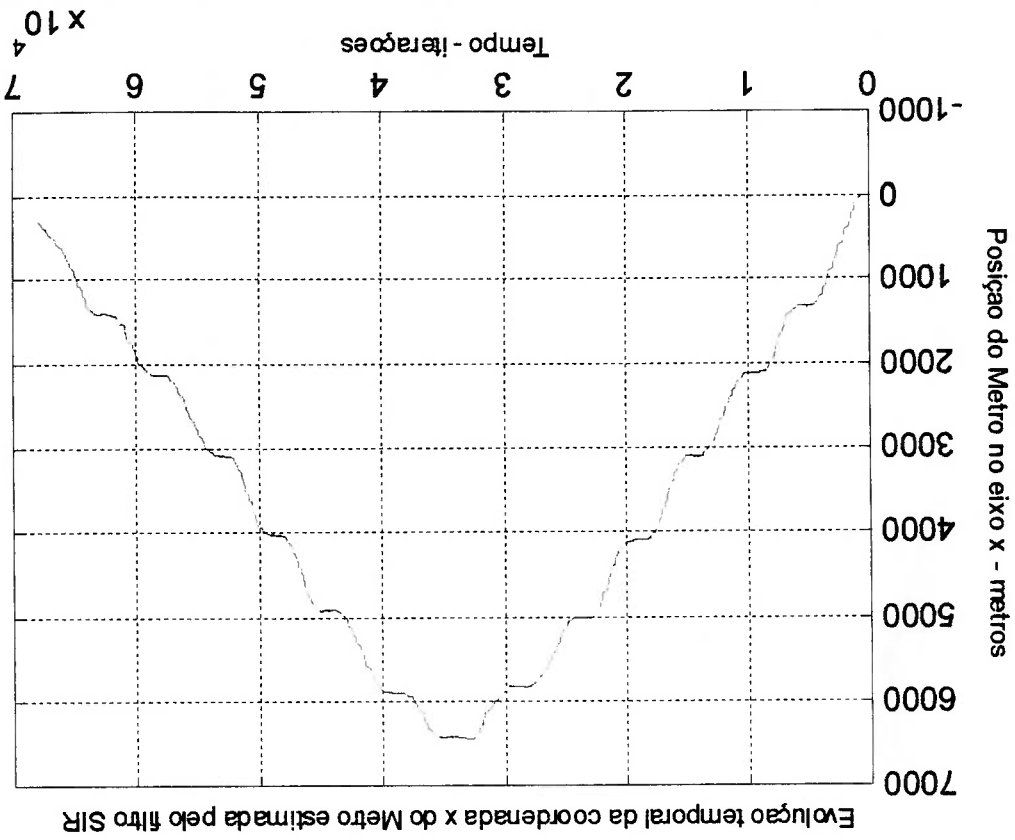
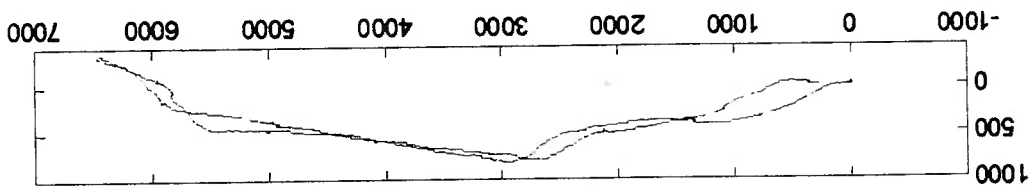


Figura 4.34 – Gráfico da evolução temporal da coordenada x do Metro (filtro SIR).

Abaixo temos as trajetórias metroviárias estimadas pelos filtros de Kalman e SIR na escala proporcional:

Filtro SIR:



Filtro de Kalman:

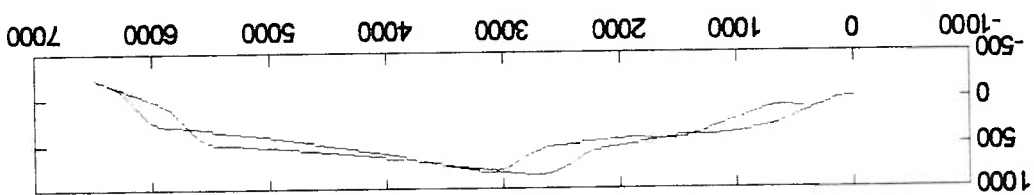


Figura 4.35 – Trajetórias da linha verde do Metrô de São Paulo estimadas pelo filtro SIR e pelo filtro de Kalman em escala proporcional.

Pode-se perceber que as trajetórias são extremamente semelhantes, e ambas são compatíveis com o mapeamento da linha verde do Metrô de São Paulo.

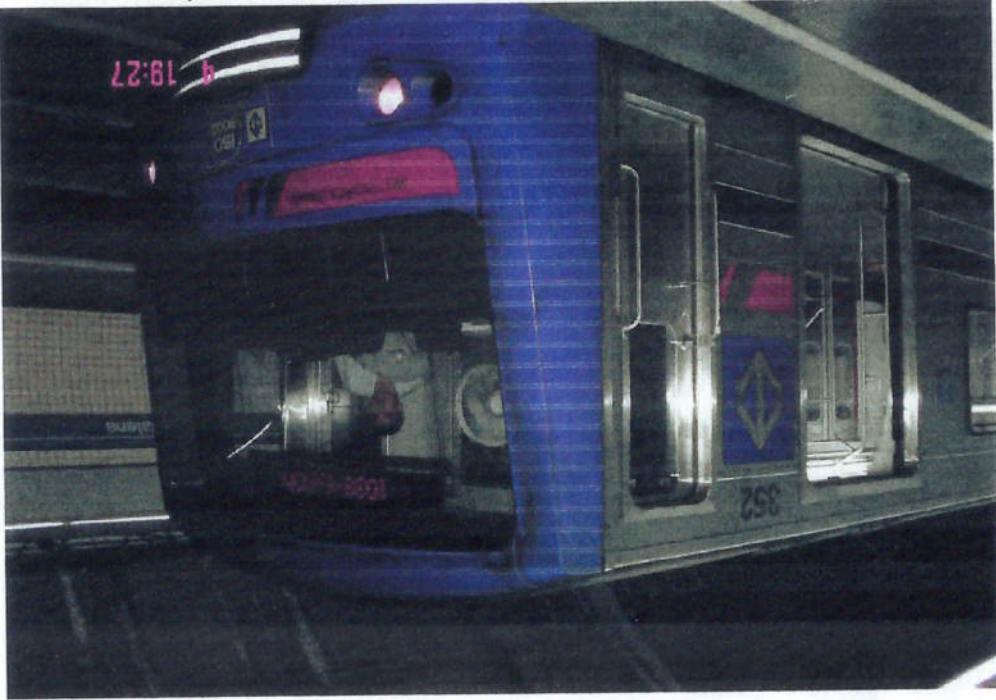


Figura 4.37 – Composição do metrô utilizada para realizar o experimento.

Figura 4.36 – Aspecto da instrumentação utilizada no experimento metroviário. No notebook está instalado o software Gyroview, como se pode observar na tela, e na caixa montada abaixo do notebook estão as baterias utilizadas para alimentação do sistema e a Central Inercial strapdown.

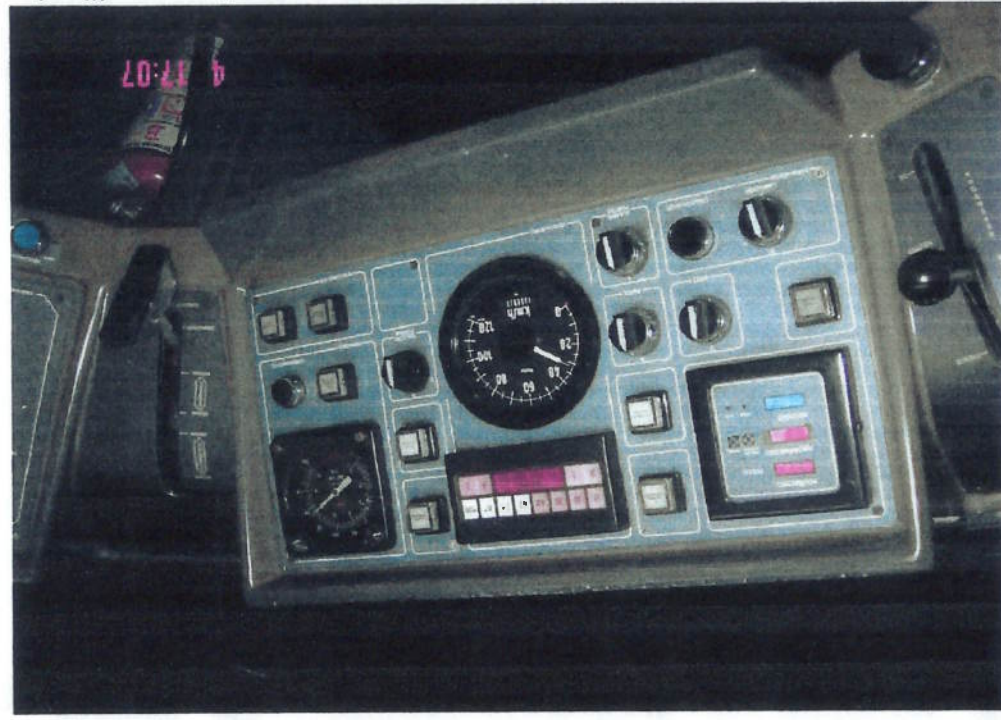


O sistema está montado no interior da cabine do Metrô. Pode-se observar, em sua tela, o navegador do software GyroView. O sistema esta montado no interior da cabine do Metrô.

Figura 4.39 – Notebook utilizado no experimento metroviário durante a coleta de dados no Metrô. Pode-se observar, em sua tela, o navegador do software GyroView.



Figura 4.38 – Painel do Metrô. No centro dele está o velocímetro utilizado para coletar as medidas de referência (velocidades instantâneas, de 10 em 10 segundos, da composição).



O experimento do pig foi conduzido do seguinte modo: foi embarcada uma Central Inercial de baixo custo num PIG instrumentado (este pig possui um odômetro digital, que emite 1 pulso a cada volta completa de sua roda – esta roda está sempre em contato com a parede interna dos dutos). Para fazer a aquisição dos dados foi utilizado um microprocessador PC 104 (ele realiza a aquisição e o armazenamento dos dados da Central Inercial e do odômetro). Este pig foi colocado dentro de um circuito fechado de oleodutos, onde ele descreveu uma trajetória impulsionado pelo óleo pressurizado no interior destes dutos – assim, a trajetória percorrida pelo pig corresponde ao desenho do circuito de dutos de testes. O pig realizou uma volta completa no loop de testes da Petrobras (Reduque, Rio de Janeiro), e os dados obtidos foram posteriormente processados por um Sistema de Navegação Inercial

Pig instrumentado é um encapsulamento de sensores e dispositivos eletrônicos embarcados utilizados para inspecionar dutos na indústria petrolífera. Um tipo de pig instrumentado é o pig ultra-sônico, que detecta espessuras de oleodutos alteradas por corrosão através da emissão e recepção de ondas de ultra-som (um pig move-se no interior dos dutos impulsionado pelo óleo – ele não possui propulsão própria) [39]. Os dados coletados pelos sensores de ultra-som são armazenados para processamento off-line. Para que se possa então saber onde estas falhas de dutos foram detectadas, faz-se necessária uma estimativa de posição do pig (os dados de posição e de ultra-som devem ser combinados na mesma base de tempo para que se possa saber a posição exata do pig nos dutos onde a falha foi detectada). Isto pode ser realizado embarcando-se uma Unidade de Medidas Inercial (UMI) no pig. Esta UMI coleta medidas de aceleração e rotação angular do pig e armazena estes dados. Utilizando os dados fornecidos pela UMI e por um odômetro fixado no pig, compõe-se um Sistema de Navegação Inercial de fusão sensorial, que através de estimadores como o filtro de Kalman ou os filtros de partículas pode estimar a trajetória descrita pelo pig. Uma outra utilidade da estimativa da trajetória do pig é a reconstrução do mapeamento de dutos, que é de grande valia para a indústria petrolífera em trechos onde o mapeamento documentado é pobre, pouco especificado ou ausente.

4.5. EXPERIMENTO DO PIG INSTRUMENTADO

auxiliado por um filtro de estimação (os dados utilizados foram base de tempo, acelerações nos 3 eixos ortogonais da plataforma, rotações angulares nos mesmos 3 eixos e distâncias medidas pelo odômetro). O duto de testes é aproximadamente retangular, com 150 metros de perímetro. As fotos apresentadas a seguir auxiliam no entendimento de como o experimento do pig inercial foi realizado.

Os sinais da Central Inercial (acelerações medidas em 3 eixos ortogonais e velocidades de rotação angular medidas nos mesmos 3 eixos) passaram por um pré-processamento. Este pré-processamento foi feito passando-se os sinais da UMI por um filtro passa baixas de Butterworth de segunda ordem, com frequência de corte igual a 5 Hz.

Para que os dados do odômetro pudessem ser integrados com as medidas da UMI, eles foram transformados em velocidades instantâneas. Isto foi feito do seguinte modo: calculou-se uma velocidade média para os pontos dos trechos nos quais a leitura do odômetro foi constante (o odômetro é digital, e mede distâncias cumulativas de 12.7 em 12.7 cm). Esta velocidade foi calculada de acordo com a expressão abaixo:

$$vel_{MÉDIA} = \frac{T_{FINAL} - T_{INICIAL}}{0.127} \quad (4.11)$$

Na equação acima, $T_{INICIAL}$ e T_{FINAL} são os instantes de tempo inicial e final do trecho no qual a leitura do odômetro foi constante, respectivamente. Desta forma, pode-se obter velocidades instantâneas do pig no sistema de coordenadas RPY (as velocidades assim obtidas correspondem ao eixo do movimento do pig, que é o eixo x), e transformá-las para o sistema de coordenadas NED:

$$Vel_k = \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} = C_{RPY}^{NED} \cdot \begin{bmatrix} vel_k \\ 0 \\ 0 \end{bmatrix} \quad (4.12)$$

Na equação (4.12), vel_k é a velocidade do pig no instante k , obtida de acordo com a equação (4.11); C_{RPY}^{NED} é a matriz de transformação de coordenadas do sistema RPY para o sistema NED (ver Capítulo 3). Vel_k é o vetor com as 3 componentes da velocidade do pig, no sistema de coordenadas NED. Este vetor, calculado a cada instante de tempo k , será utilizado como medida de referência no modelo para estimação de posição do pig inercial (o modelo utilizado é aquele dado pelas equações (3.20) e (3.21)).

Após o pig inercial ter realizado 1 volta completa no loop de testes da Petrobras, foram coletados os dados de aceleração, velocidade de rotação angular e posição através da Central Inercial e do odômetro. Estes dados são apresentados graficamente abaixo (já foram calculadas as velocidades instantâneas do pig através dos dados de posição fornecidos pelo odômetro, conforme o procedimento descrito acima):

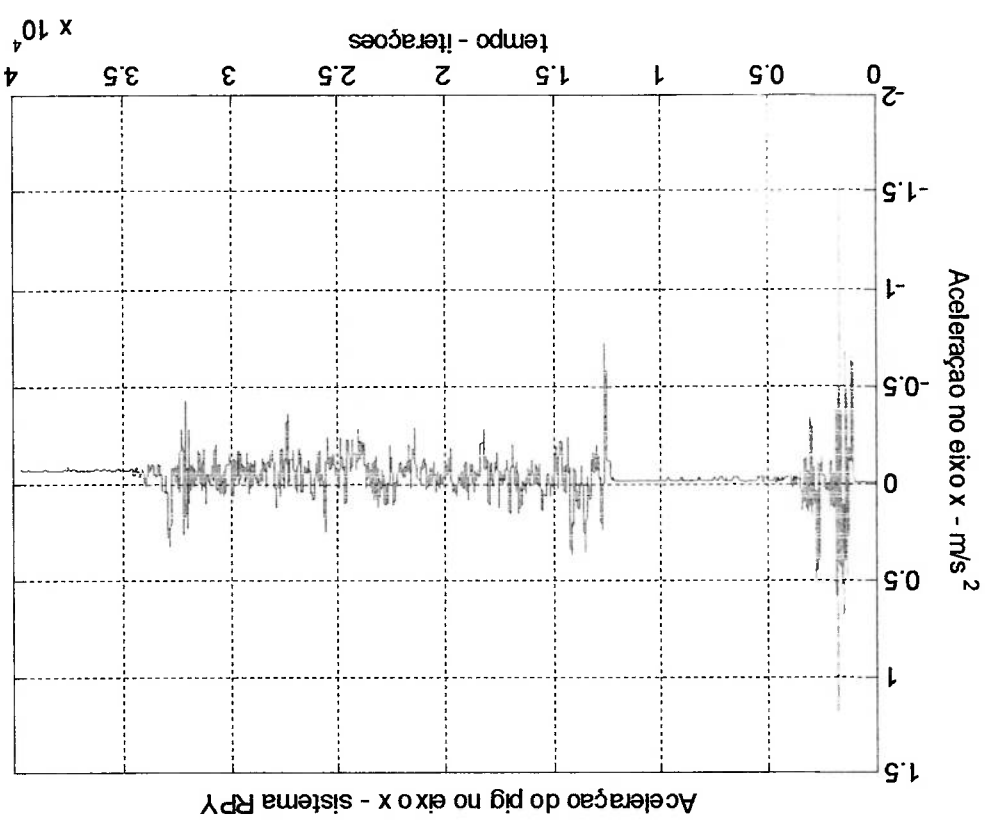


Figura 4.40 – Gráfico da aceleração do pig no eixo x medida pela Central Inercial.

Figura 4.42 – Gráfico da aceleração do pig no eixo z medida pela Central Inercial.

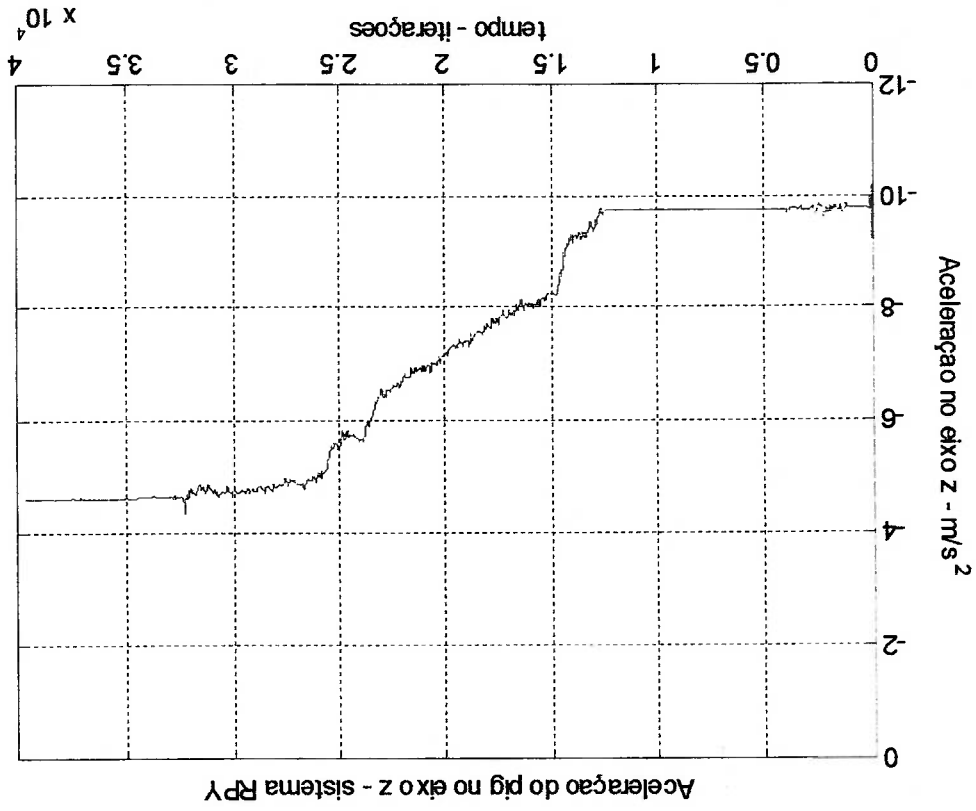


Figura 4.41 – Gráfico da aceleração do pig no eixo y medida pela Central Inercial.



Figura 4.44 – Gráfico da aceleração do pig (eixo y) no sistema de coordenadas NED.

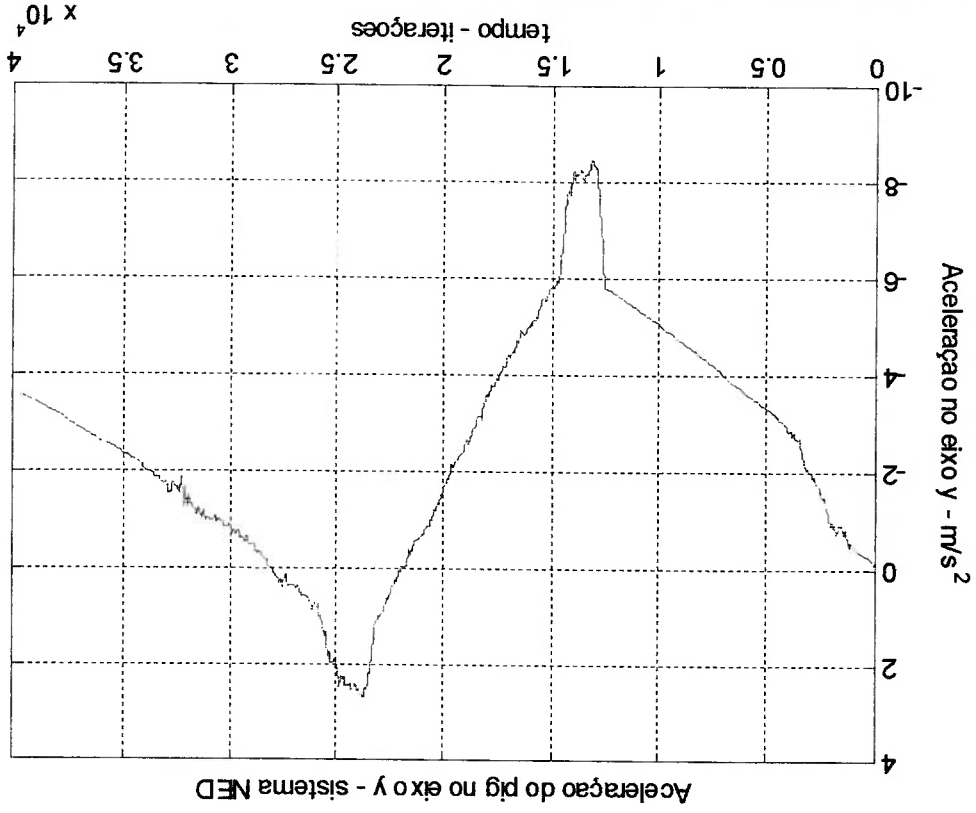


Figura 4.43 – Gráfico da aceleração do pig (eixo x) no sistema de coordenadas NED.

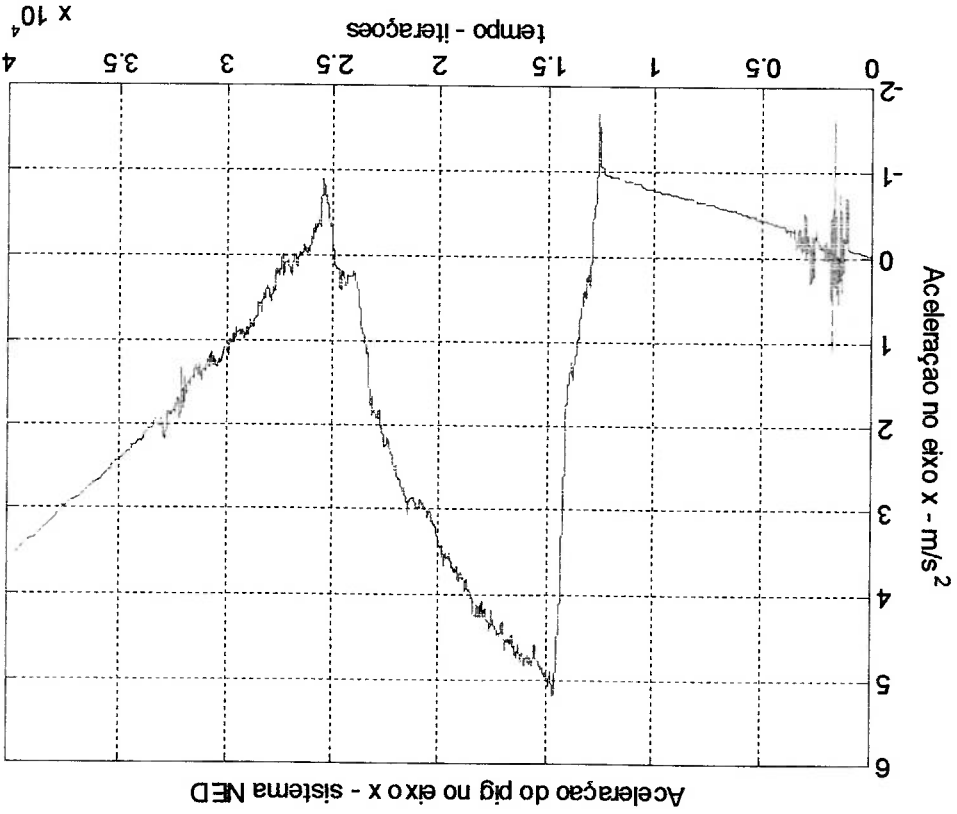


Figura 4.46 – Gráfico da velocidade de rotação angular do pig em torno do eixo z.

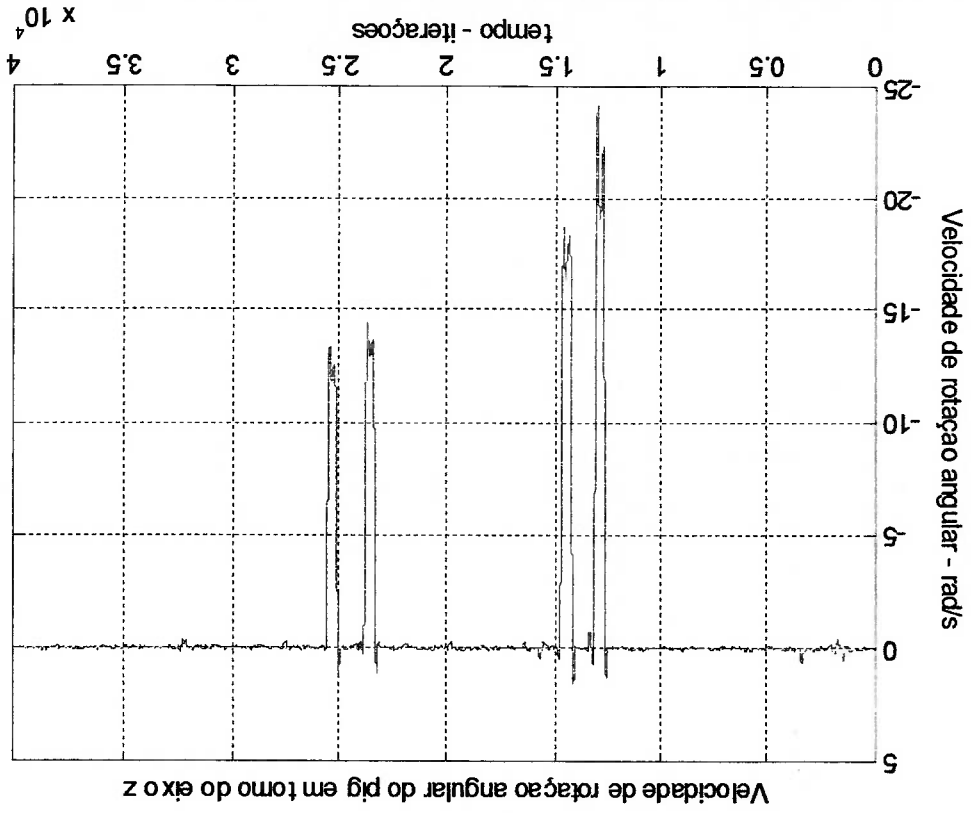


Figura 4.45 – Gráfico da aceleração do pig (eixo z) no sistema de coordenadas NED.

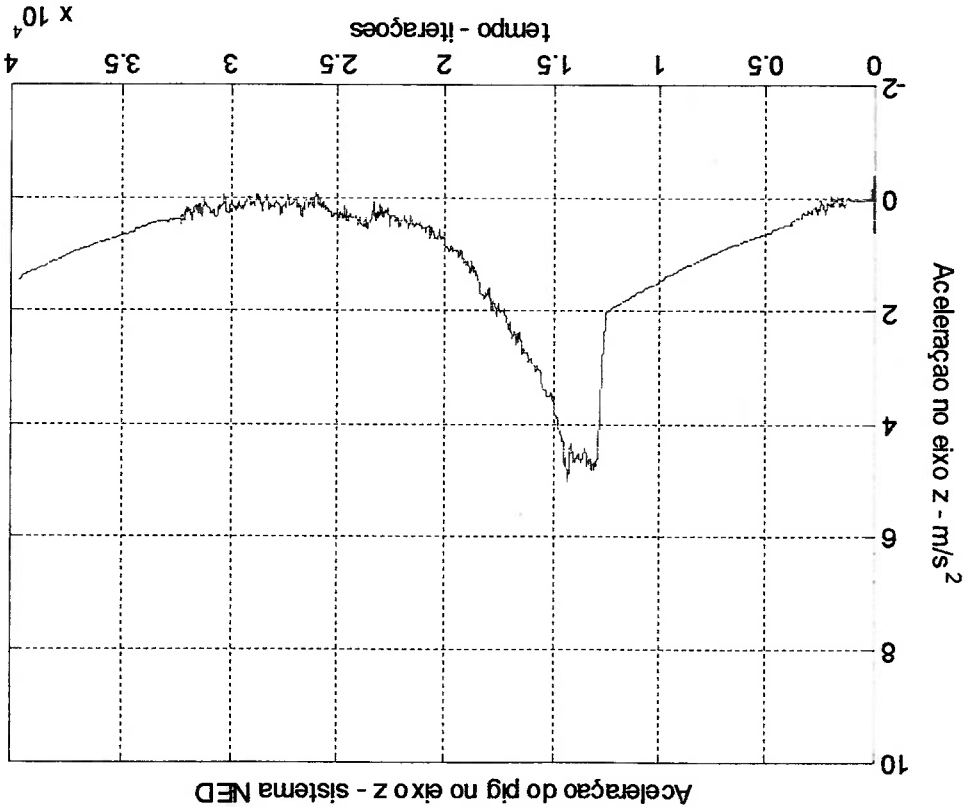


Figura 4.48 – Gráfico da velocidade do pig (eixo y) no sistema de coordenadas NED.

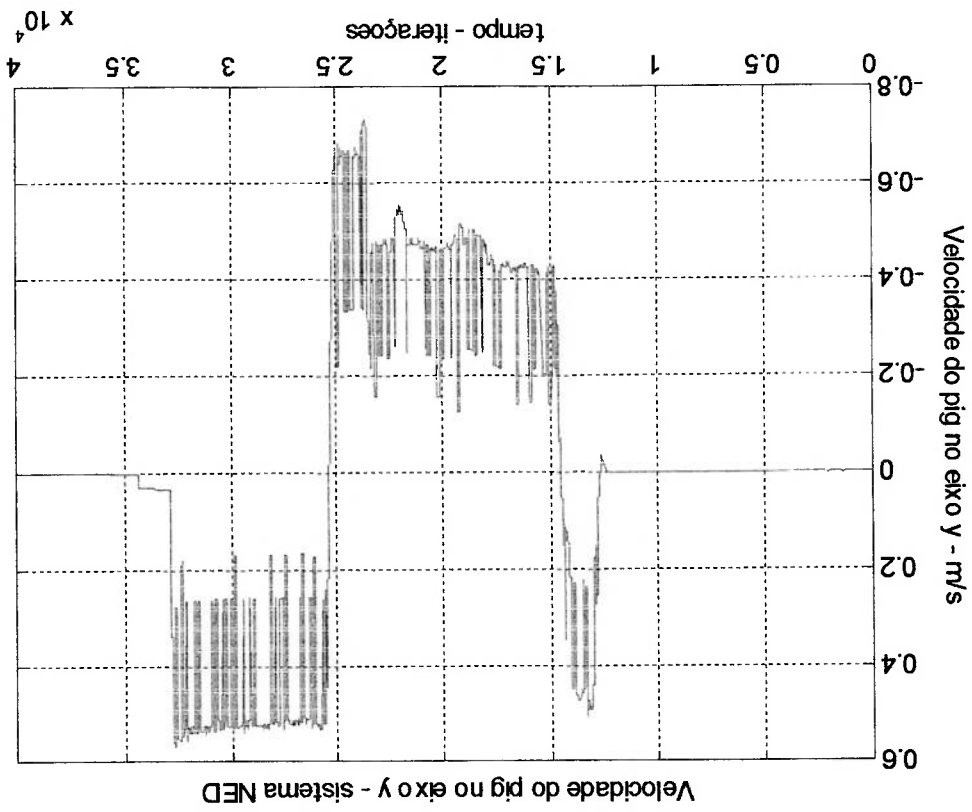


Figura 4.47 – Gráfico da velocidade do pig (eixo x) no sistema de coordenadas NED.

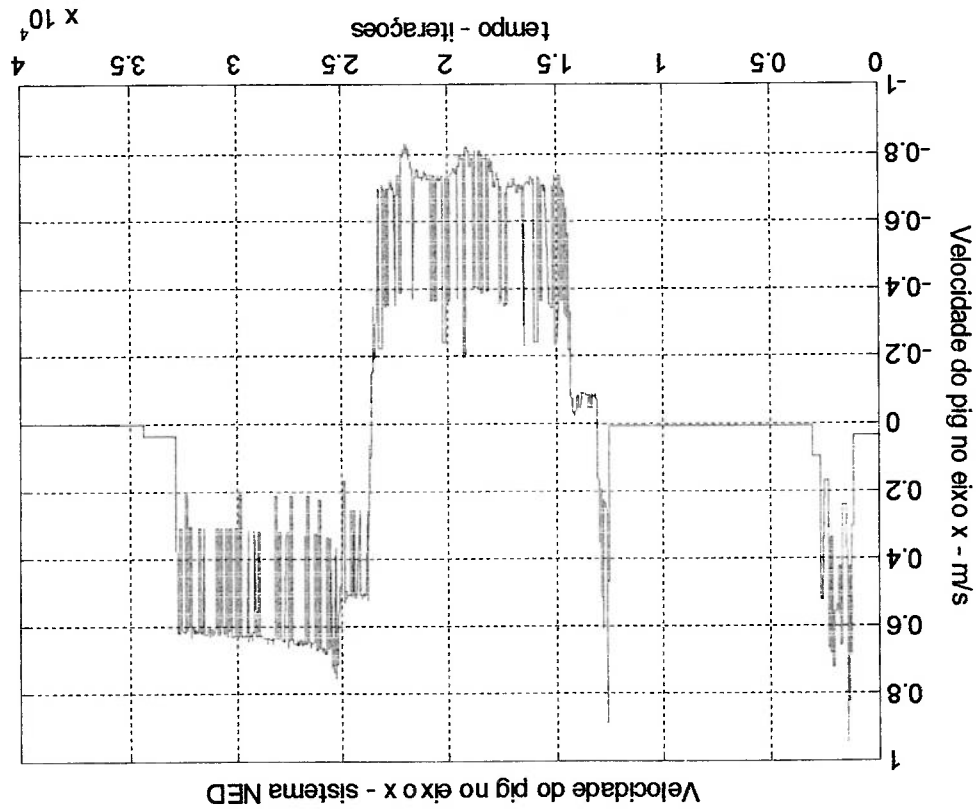
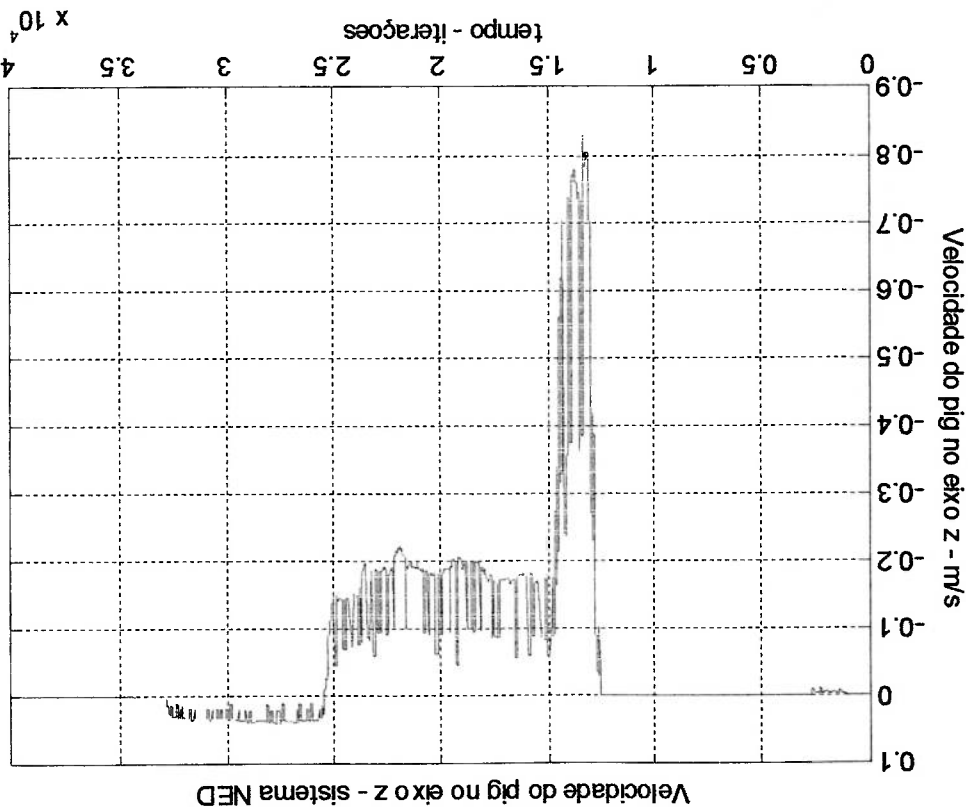


Figura 4.49 – Gráfico da velocidade do pig (eixo z) no sistema de coordenadas NED.



A duração do experimento foi de aproximadamente 7 minutos, e a taxa de amostragem utilizada foi de 100 amostras por segundo.

Utilizando o modelo dado pelas equações (3.20) e (3.21), fez-se uma integração das medidas descritas acima através do filtro de Kalman (ver figura 4.2). Este sistema foi então implementado no MATLAB, e obteve-se a trajetória descrita pelo pig, dada na figura 4.50. O percurso reconstruído é muito próximo do original (o circuito do loop de testes é fechado, sendo aproximadamente retangular), e os erros obtidos ao final do percurso foram de 1.9 m para o eixo x e 1.56 m para o eixo y. Em termos percentuais, estes erros equivalem a:

$$\text{Erro}_x = 1.73\%$$

$$\text{Erro}_y = 1.48\%$$

Estes erros são baixos, o que comprova que o filtro de Kalman é um excelente estimador de estados para aplicações relacionadas à Navegação Inercial.

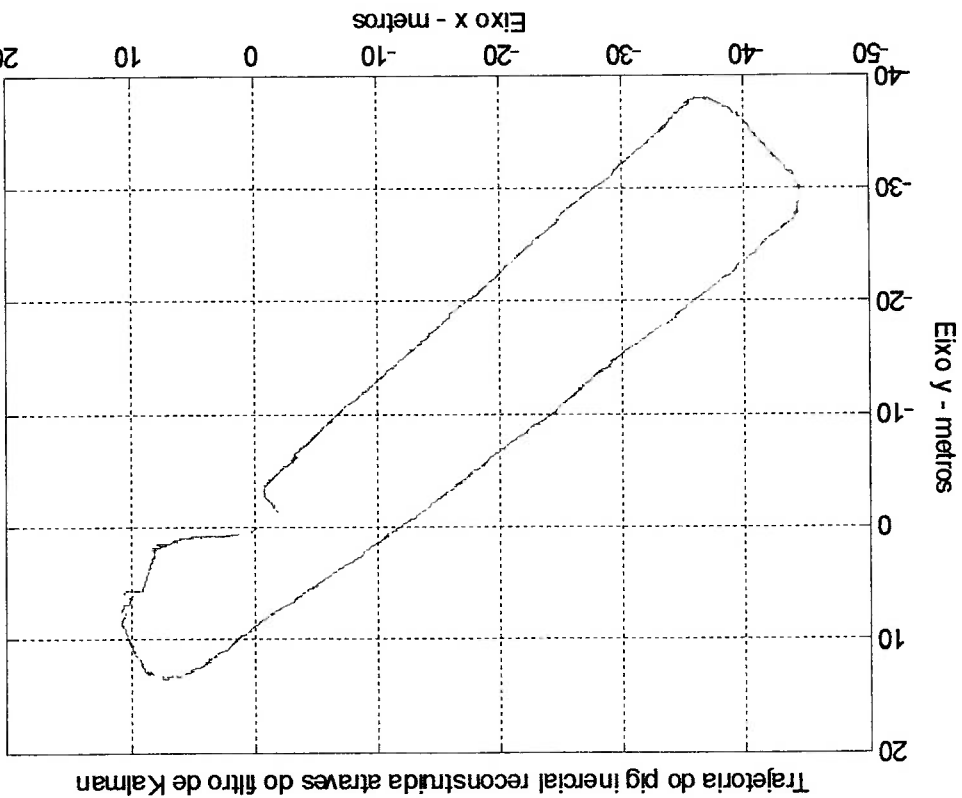


Figura 4.50 – Gráfico da trajetória do pig reconstruída através do filtro de Kalman.

O modelo dado pelas equações (3.20) e (3.21) também foi utilizado para estimação de posição do pig inercial através do filtro SIR. O algoritmo do filtro SIR é aquele que foi definido na seção 4.1.3, com $N = 100$ amostras. Este filtro foi executado em ambiente MATLAB, e não se conseguiu obter uma trajetória compatível com a trajetória realizada pelo pig no duto de testes da Petrobras. Isto pode ter sido ocasionado pelas seguintes razões:

1. Número de amostras inadequado – nesta aplicação, o filtro SIR poderia requerer um número de amostras muito elevado;
2. Erros no modelo do sistema – as matrizes C e G do modelo do sistema descrito no capítulo 3 não necessitaram de ajustes muito específicos para os testes automotivo e metroviário, mas no caso do pig inercial poderíamos ter que operar um ajuste específico destas matrizes para o correto funcionamento do filtro SIR;

Assim, pode-se depreender deste experimento que os filtros de partículas são extremamente dependentes da aplicação em que eles estão sendo utilizados, o que não ocorre com o filtro de Kalman. Para utilizarmos o filtro de Kalman, o modelo do sistema não necessita ser muito preciso para que o desempenho do filtro seja satisfatório, o que não ocorre com o filtro SIR em determinadas aplicações (Navegação Inercial em veículos que se locomovem a baixas velocidades, como o pig inercial). Além disso, para utilizarmos os filtros de partículas precisamos calcular algumas funções densidade de probabilidade, o que pode representar uma dificuldade em certas aplicações (no caso da Navegação Inercial, vimos que a aproximação de algumas funções densidade de probabilidade por Gaussianas foi satisfatória no experimento automotivo, mas no caso do pig inercial ela não funcionou bem). O uso dos filtros de partículas também está vinculado ao nível de ruído do processo – se o nível de ruído for alto, deve-se utilizar o filtro SIR, e se o nível de ruído for baixo, é preferível utilizar o filtro ASIR. Uma questão em aberto no que diz respeito ao uso dos filtros de partículas é como determinar o número de partículas otimizado a ser utilizado numa aplicação (um número de partículas muito elevado pode tornar a execução do filtro muito lenta, enquanto que um número baixo de partículas pode fazer com que o filtro não alcance a convergência).

3. Função Gaussiana inadequada para aproximar a função densidade de probabilidade $p(y/x)$ – No caso do pig inercial, uma função Gaussiana (como aquela que foi utilizada no filtro SIR) pode não ser uma boa aproximação para a função $p(y/x)$, e isto pode ter prejudicado o funcionamento do filtro SIR.

Figura 4.52 – Aspecto da eletrônica embarcada utilizada no pig instrumentado para aquisição dados da Central Inercial e do odômetro.

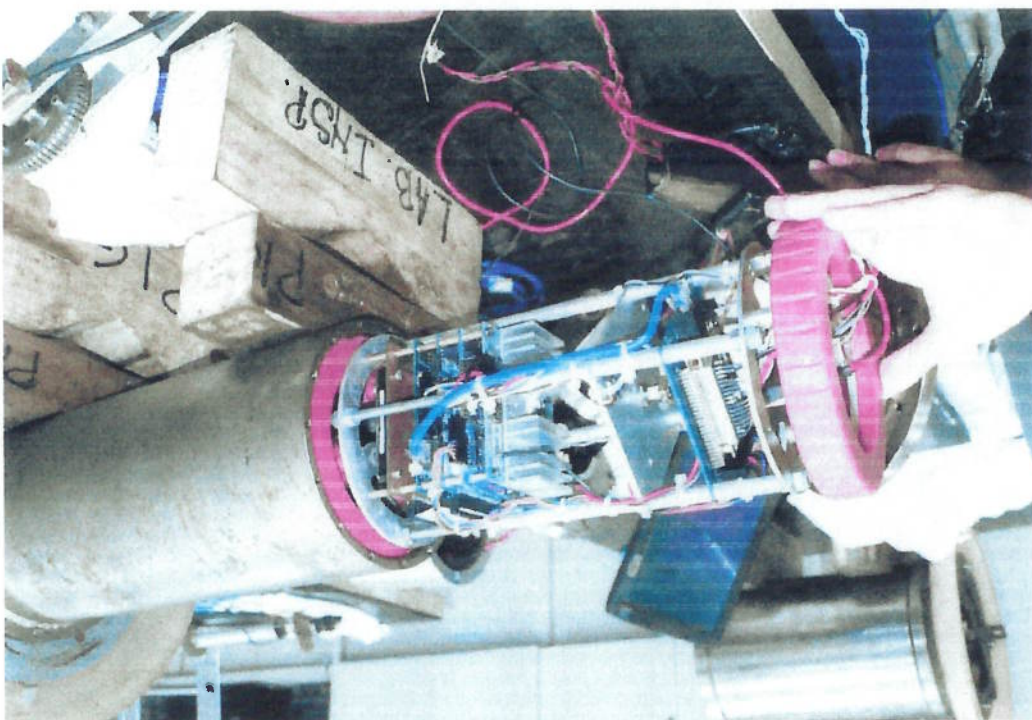


Figura 4.51 – Montagem da Central Inercial no interior do pig instrumentado utilizado no experimento feito no duto de testes da Petróbras (Rio de Janeiro).

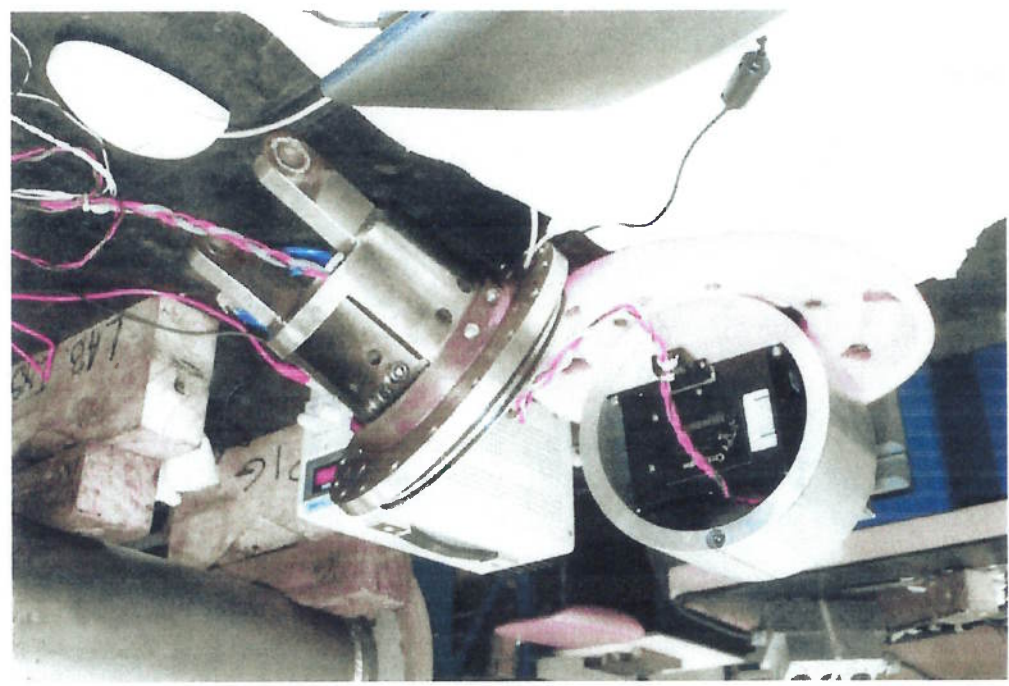
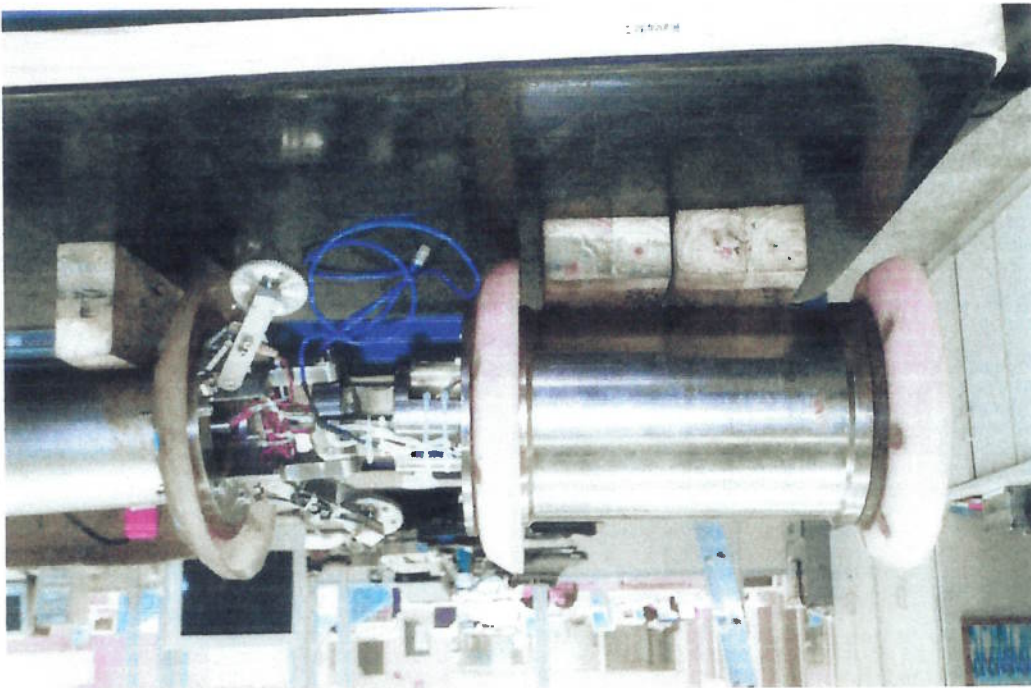


Figura 4.54 – Pig instrumentado (vista total) na bancada de um dos laboratórios do CENPES (Centro de Pesquisas da Petrobrás – Rio de Janeiro).



Fig.

Figura 4.53 – Pig instrumentado. Pode-se observar 2 odômetros, um de cada lado do



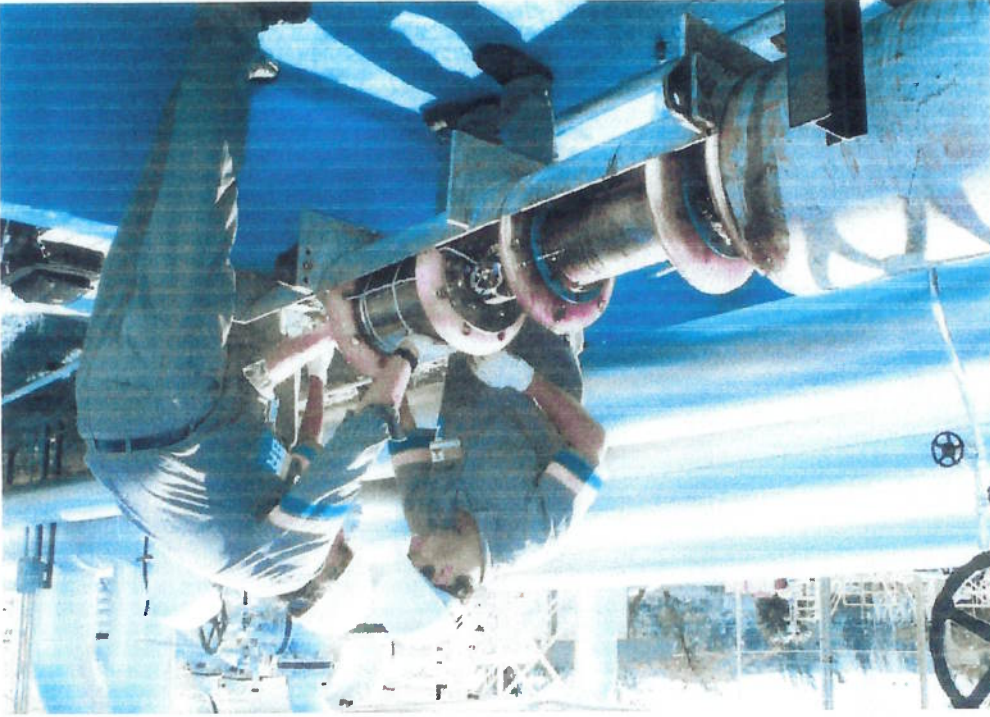


Figura 4.56 – Pig sendo inserido na tubulação de testes da Petrobrás. Ao fundo, observa-se a tubulação que compõe o circuito de dutos de teste da Petrobrás.

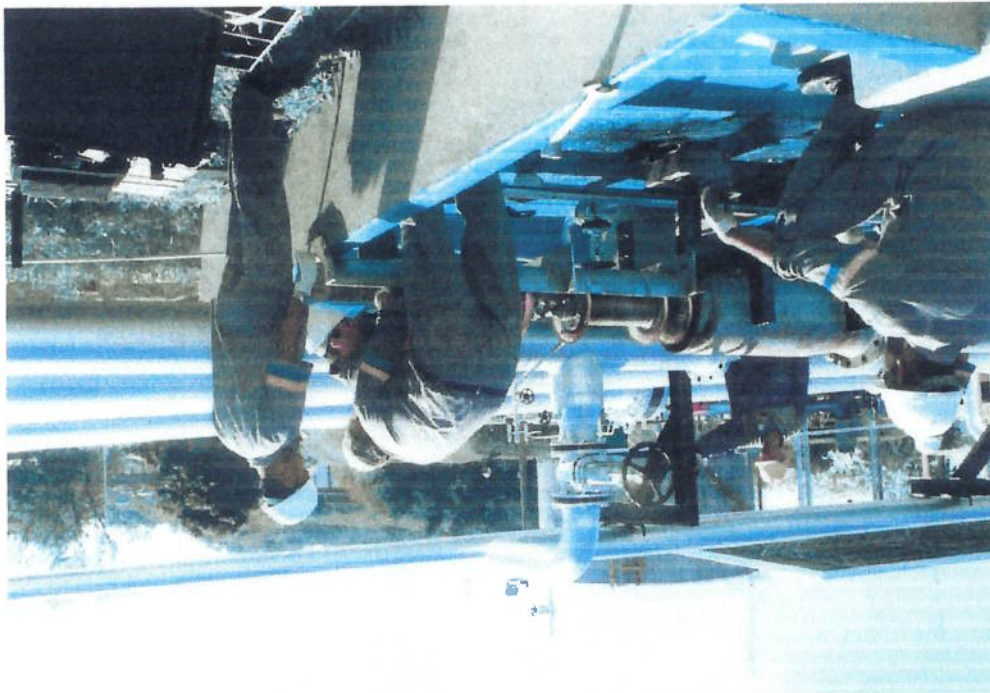


Figura 4.55 – Pig sendo inserido na tubulação de testes da Petrobrás (Duque de Caxias – RJ) pela equipe do Sr. Cláudio Camerini.

(4.13)

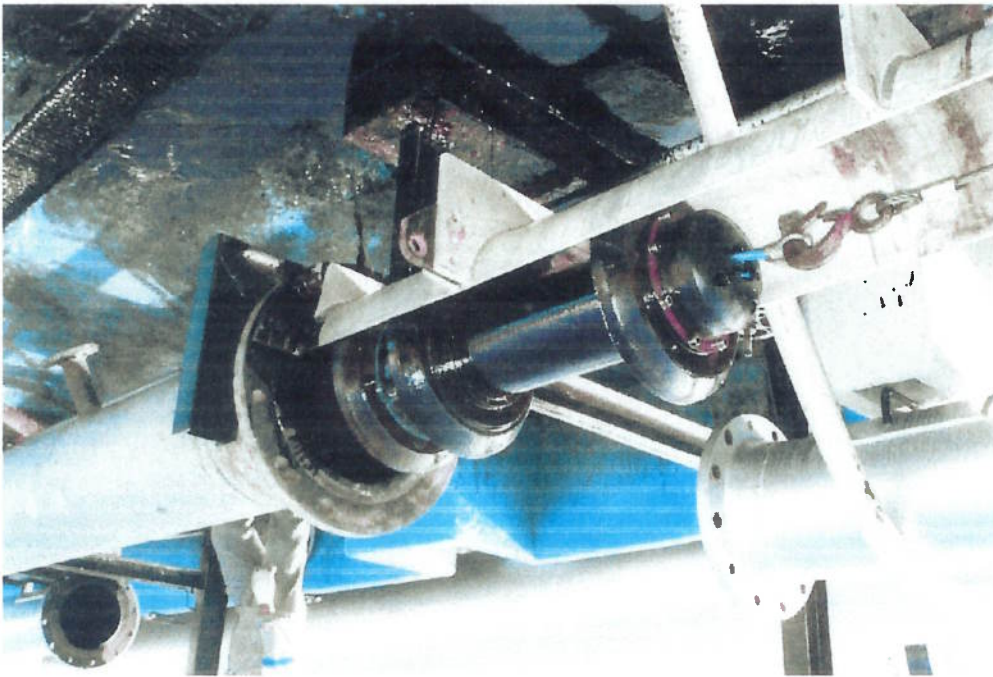
$$Erro = \sum_{k=1}^N (x_k - \hat{x}_k)^2$$

seguinte:

O procedimento para a realização do segundo experimento automotivo é aquele que foi descrito na seção 4.2. Neste teste, o carro manteve-se a uma velocidade constante de 30 km/h (esta velocidade, transformada para o sistema NED, foi utilizada como medição de referência nos filtros). O percurso realizado foi aproximadamente circular, sendo descrito em uma das rotatórias da Cidade Universitária. A montagem utilizada para a realização deste experimento pode ser apreciada nas figuras 4.64 e 4.65. A taxa de amostragem utilizada foi de 50 amostras por segundo. A trajetória estimada com o auxílio do filtro de Kalman e a trajetória real do automóvel estão dadas na figura 4.58. Pode-se perceber que elas são muito próximas. Os erros de estimação serão, neste caso, calculados sobre todo o percurso (e não somente sobre o ponto final da trajetória, como foi feito nos outros testes). A equação utilizada será a

4.6. EXPERIMENTO AUTOMOTIVO II

Figura 4.57 – Pig sendo retirado da tubulação de testes da Petrobras.



Na equação acima, x_k é o estado verdadeiro do sistema e \hat{x}_k é o estado estimado pelo filtro de Kalman ou pelo filtro SIR. Utilizando-se esta equação, pode-se calcular os erros de estimação para o filtro de Kalman:

$$\text{Erro } x - 25837$$

$$\text{Erro } y - 46173$$

No ponto final da trajetória, os erros são de 0.12m para o eixo x e 6.25m para o eixo y. Os erros percentuais, então, são de 0.075% para o eixo x e 3.9% para o eixo y, o que é satisfatório.

Trajétoria rotatória estimada pelo filtro de Kalman (contínua) e trajetória real (tracejada)

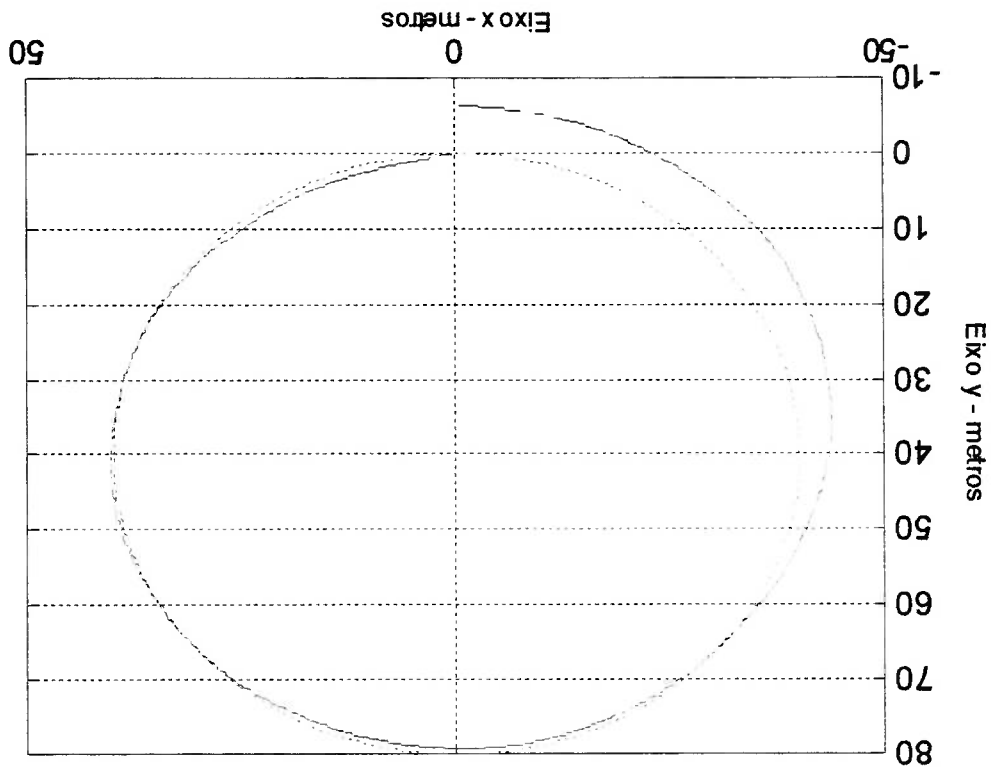


Figura 4.58 – Gráfico comparativo entre a trajetória original (linha tracejada) e a trajetória estimada pelo filtro de Kalman (linha contínua).

Figura 4.60 – Gráfico da coordenada y do veículo estimada pelo filtro de Kalman.

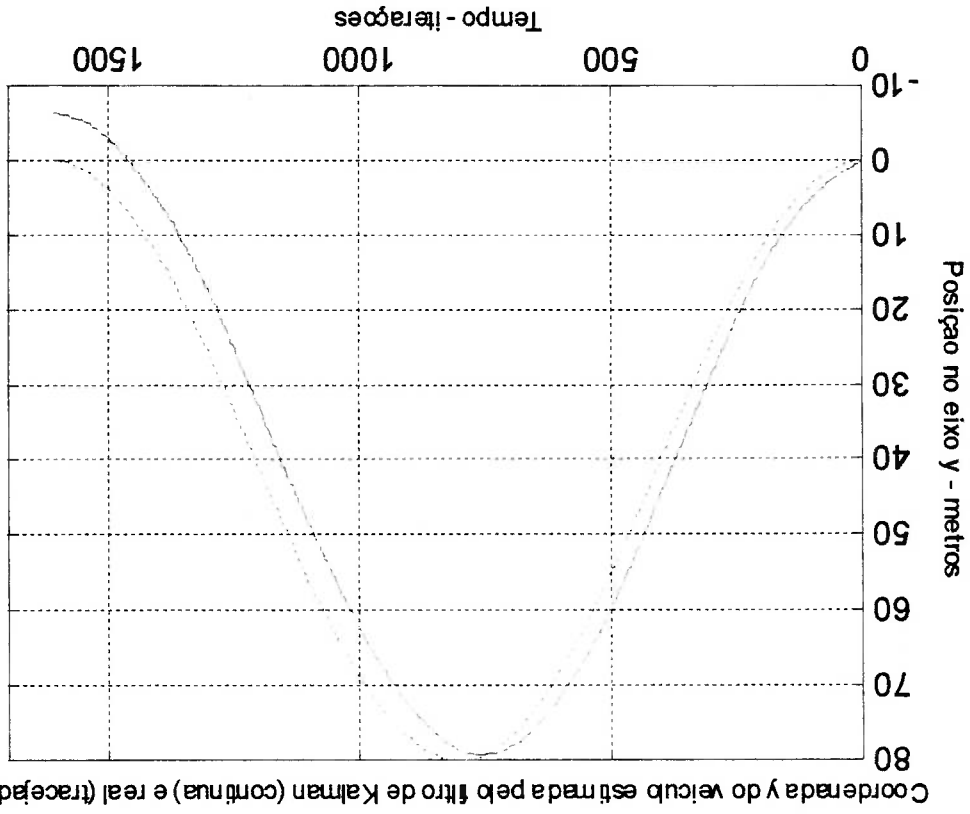
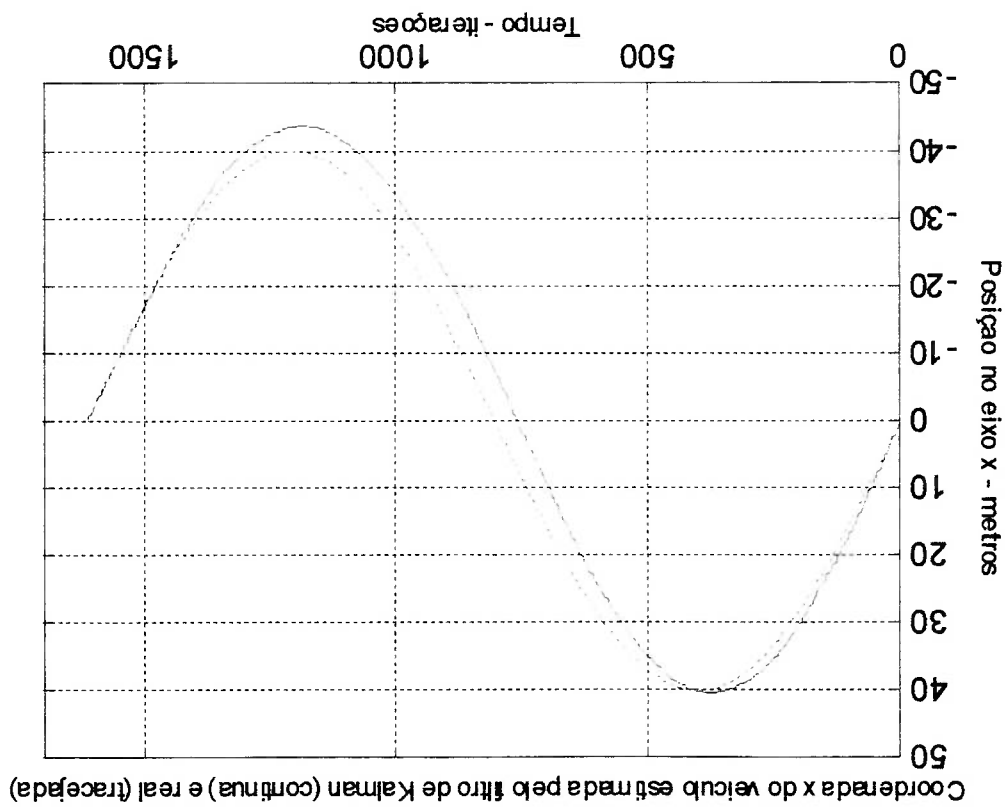


Figura 4.59 – Gráfico da coordenada x do veículo estimada pelo filtro de Kalman.



O filtro SIR foi executado utilizando-se 450 amostras (este foi um número otimizado – com menos amostras o filtro não funcionou bem neste teste). Também foi necessário um ajuste da matriz C do modelo do sistema para este caso: $C = 2 \cdot I_{90}$. Desta forma, sincronizar o filtro SIR não foi uma tarefa simples no segundo teste automotivo. A trajetória estimada por este filtro é apresentada na figura 4.61, na qual também está plotada a trajetória original realizada pelo veículo. Nota-se que as 2 trajetórias estão muito próximas.

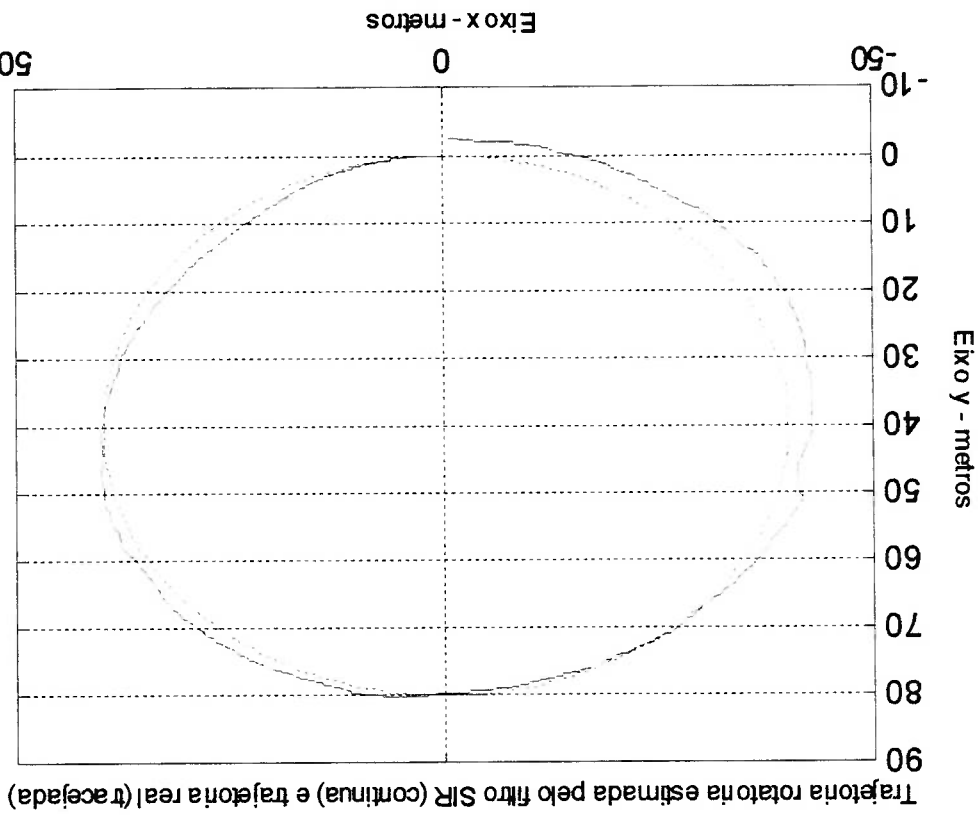


Figura 4.61 – Gráfico comparativo entre a trajetória original (linha tracejada) e a rotatória estimada pelo filtro SIR (linha contínua).

Os erros de estimação para este filtro, considerando todo o percurso, são (equação (4.13)):

$$\text{Erro } x - 14335$$

$$\text{Erro } y - 15282$$

Ambos os erros acima são inferiores aos erros apresentados pelo filtro de Kalman, e isto provavelmente ocorreu devido ao fato de o sistema ser não-gaussiano e o filtro SIR utilizar nas suas estimativas amostras do ruído real do processo.

Os erros no ponto final da trajetória são os seguintes:

Erro x – 0,25m

Erro y – 2,7m

Estes erros correspondem a 0,16% para o eixo x e 1,69% para o eixo y, o que é satisfatório para Sistemas de Navegação Inercial que empregam Centrais Inerciais de baixo custo.

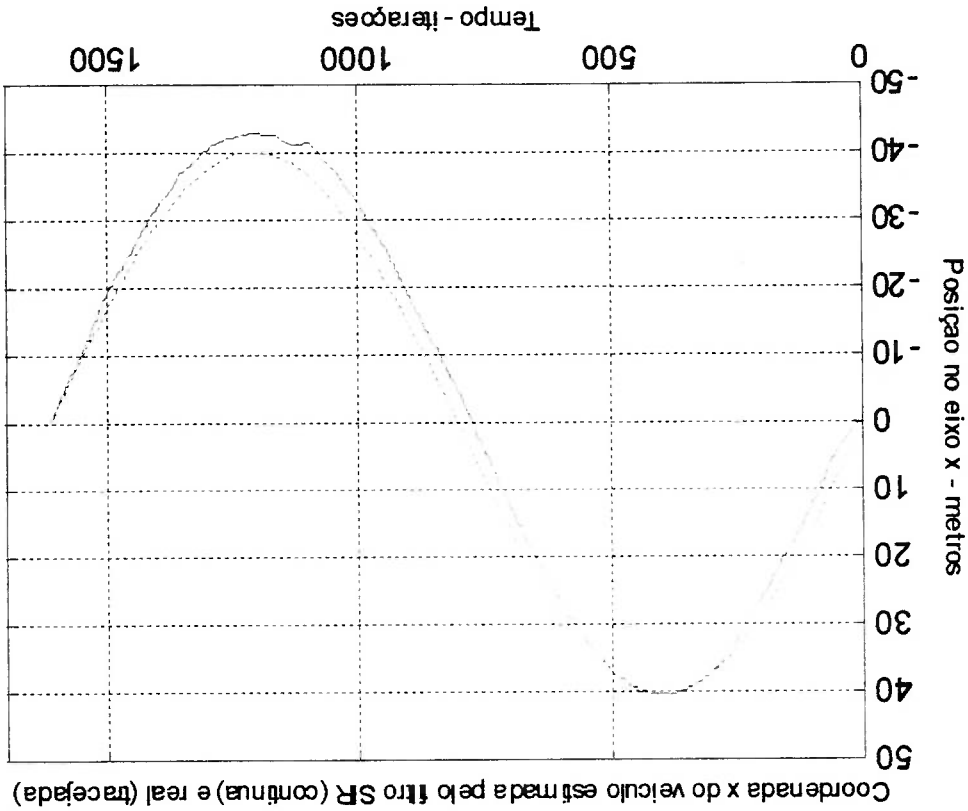
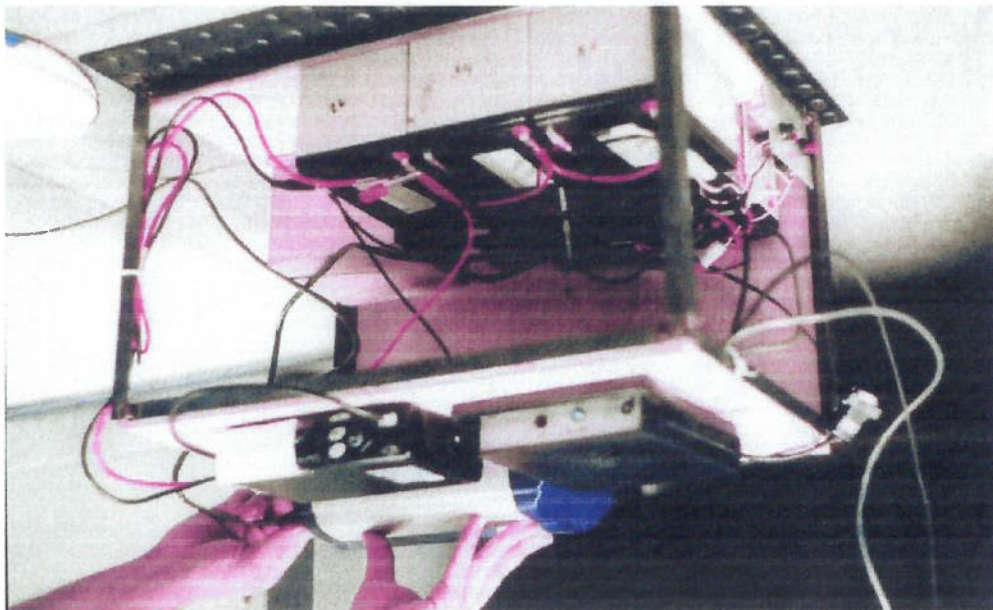


Figura 4.62 – Gráfico comparativo entre a evolução temporal da coordenada x do veículo estimada pelo filtro SIR (linha contínua) e a evolução real da coordenada x do veículo (linha tracejada).

Figura 4.64 – Instrumentação utilizada na realização do segundo experimento automotivo. No interior da caixa pode-se observar a Central Inercial (cor amarelo (ouro)), ao lado das baterias usadas para a alimentação da mesma.



(linha tracejada).

Figura 4.63 – Gráfico comparativo entre a evolução temporal da coordenada y do veículo estimada pelo filtro SIR (linha contínua) e a evolução real da coordenada y

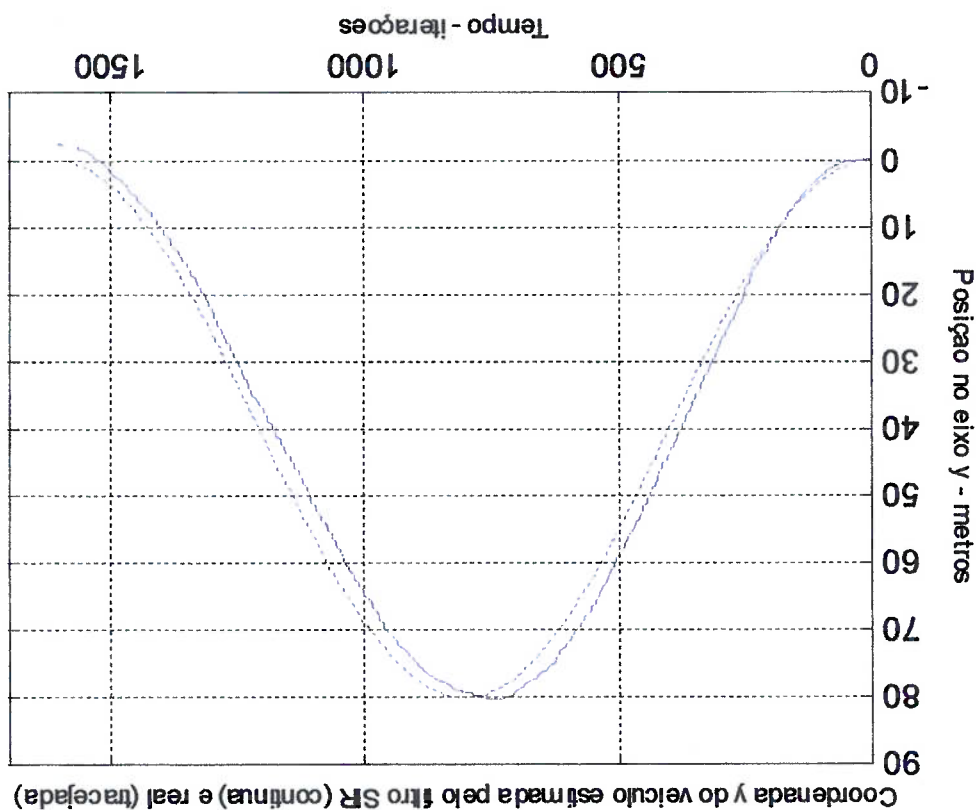
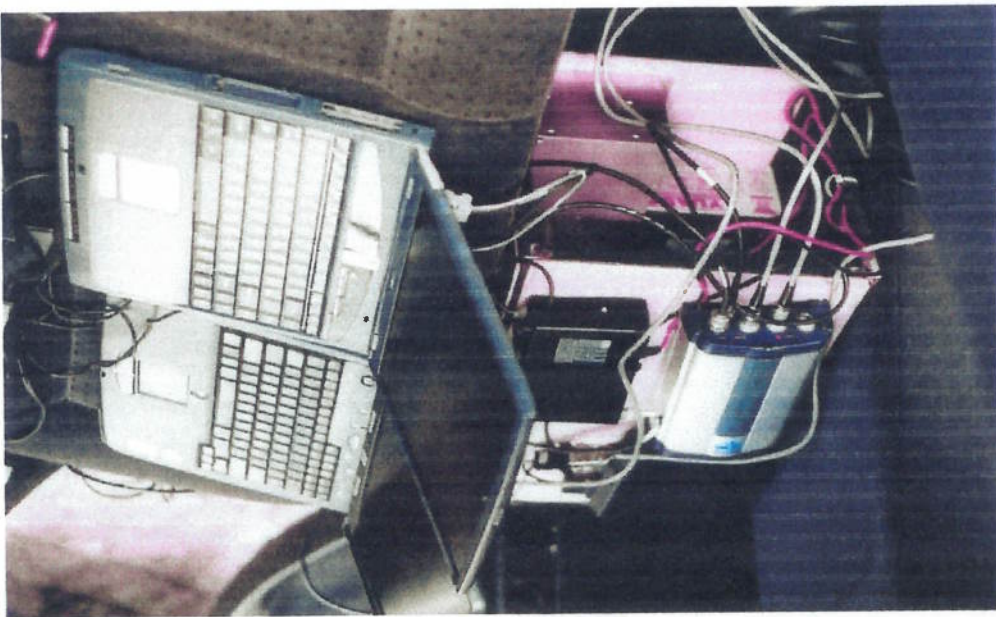


Figura 4.65 – Montagem da instrumentação no interior do automóvel. Os notebooks no banco traseiro foram utilizados para a aquisição dos dados da Central Inercial (que foi feita através do software Gyroview).



5. Conclusões

Os filtros de partículas funcionam como uma generalização do filtro de Kalman, sendo portanto aplicáveis a uma amplitude de modelos no espaço de estados maior que este filtro tradicional. O desempenho do filtro de Kalman deixa de ser ótimo quando aplicado a sistemas não-lineares e/ou que tenham ruídos não-gaussianos associados. Assim sendo, o uso dos filtros de partículas pode fornecer um melhor desempenho nestes casos, pois podem utilizar nas estimativas modelos não-lineares e amostras de ruídos não-gaussianos. Contudo, a aplicação dos filtros de partículas a um modelo qualquer requer o conhecimento de algumas funções densidade de probabilidades (por exemplo, $p(y/x)$). Em algumas aplicações, o cálculo desta função pode ser difícil. Assim, 2 questões em aberto para a implementação dos filtros de partículas são a escolha adequada do valor de N (número total de amostras a serem utilizadas) e como calcular a função densidade de probabilidades $p(y/x)$. Devido a estas complicações, os filtros de partículas são mais dependentes da aplicação que o filtro de Kalman. Um dos fatores que podem alterar o desempenho dos filtros de partículas é o nível de ruído do processo (para utilizarmos estes filtros, devemos ter algum conhecimento acerca do ruído do processo) – dependendo dele, a escolha do filtro de partículas a ser utilizado pode variar (filtro SIR ou ASIR).

A estimação de posição através dos filtros de Kalman e dos filtros de partículas se mostrou eficaz quando determinamos a posição de um automóvel no qual estava embarcada uma Central Inercial. Foram utilizadas para isto as medidas da própria Central (acelerações em 3 eixos e velocidades de rotação angular em torno de 3 eixos) e também medidas de referência (no caso, velocidades fornecidas pelo velocímetro do automóvel). Para Centrais Inerciais de baixo custo (que é o caso da Central utilizada), não é possível estimar a posição de um veículo sem medidas de referência (ou seja, unicamente com as medidas fornecidas pela Central), já que as medidas de aceleração e rotação da Central têm um ruído relativamente alto associado a elas. Os erros percentuais apresentados pelos filtros mostraram-se satisfatórios, o que indica que a aplicação dos filtros de Kalman e de partículas em Navegação Inercial foi bem sucedida.

Comparativamente, o filtro SIR mostrou melhor desempenho no teste automático que os filtros de Kalman e ASIR. Isto se deve ao fato de este filtro utilizar em suas estimativas amostras do ruído original do processo (amostras deste ruído foram obtidas coletando-se medidas de uma Central Inercial parada sobre uma mesa de desempenho nivelada), que não é Gaussiano (ver Capítulo 4). Como o filtro de Kalman supõe que todos os ruídos associados ao processo são Gaussianos, ele não funcionou tão bem quanto o filtro de partículas SIR. Já o filtro ASIR funciona melhor quando aplicado a processos com baixos níveis de ruído (o que não é o caso da Central Inercial), e assim seu desempenho não foi tão bom quando o do filtro SIR (este último funciona melhor para processos com níveis mais altos de ruído associado às medidas). No entanto, se o filtro SIR for escolhido para executar estimações de posição em detrimento do filtro de Kalman, deve-se considerar seu tempo de execução em aplicações on-line (para $N = 500$ amostras, o tempo de execução do filtro SIR é, em média, 18 vezes maior que o tempo de execução do filtro de Kalman em ambiente MATLAB). O tempo de execução de um algoritmo SIR em ambiente MATLAB é maior que o tempo de execução de um filtro de Kalman. Neste sentido, um número de amostras muito elevado nos filtros de partículas pode tornar sua aplicação on-line inviável (um número de amostras elevado pode ser necessário para a convergência do filtro em algumas aplicações). O

mais adequado é que sejam considerados 2 aspectos na escolha de um filtro de estimação para aplicações em Navegação Inercial: tempo de execução x desempenho.

Deve-se considerar, ainda, que a adequação do modelo é de fundamental importância para o correto funcionamento dos filtros de estimação. No caso da Navegação Inercial, tivemos que realizar uma série de correções determinísticas e rotações de coordenadas nas acelerações antes de as utilizarmos para estimar a posição do automóvel. Uma modelagem adequada do problema no espaço de estados pode levar ao sucesso total da aplicação dos filtros de estimação; uma modelagem incorreta, por outro lado, pode fazer com que cheguemos a conclusões errôneas acerca do desempenho destes filtros.

A aplicação do filtro MHIR não foi bem sucedida no teste automotivo, e isto provavelmente foi devido ao fato de as cadeias de Markov geradas pelo algoritmo Metropolis Hastings não terem convergido. Para que a convergência ocorresse, provavelmente teríamos que utilizar um número de elementos muito elevado para estas cadeias, o que tornaria a execução do filtro MHIR muito demorada se comparado aos outros filtros. Assim, este filtro não se mostrou adequado à aplicação em problemas de Navegação Inercial.

No teste do pig inercial (que representa uma aplicação extremamente útil para a indústria petrolífera), o filtro de Kalman funcionou bem e se mostrou totalmente adequado, enquanto que o filtro SIR não alcançou a convergência. Isto pode ser explicado pelas seguintes causas:

1. Número de amostras inadequado – nesta aplicação, o filtro SIR poderia requerer um número de amostras muito elevado para convergir;
2. Erros no modelo do sistema – as matrizes C e G do modelo do sistema descrito no capítulo 3 não necessitaram de ajustes muito específicos para os testes automotivo e metroviário, mas no caso do pig inercial poderíamos ter quer

Assim, pode-se depreender deste experimento que os filtros de partículas são extremamente dependentes da aplicação em que eles estão sendo utilizados, o que não ocorre com o filtro de Kalman (este último mostrou-se muito versátil). Para utilizarmos o filtro de Kalman, o modelo do sistema não necessita ser muito preciso para que o desempenho do filtro seja satisfatório, o que não ocorre com o filtro SIR em determinadas aplicações (Navegação Inercial em veículos que se locomovem a baixas velocidades, como o pig inercial). Além disso, para utilizarmos os filtros de partículas precisamos calcular algumas funções densidade de probabilidade, o que pode representar uma dificuldade em certas aplicações (no caso da Navegação Inercial, vimos que a aproximação de algumas funções densidade de probabilidade por Gaussianas foi satisfatória no experimento automotivo, mas no caso do pig inercial ela não funcionou bem). O uso dos filtros de partículas também está vinculado ao nível de ruído do processo – se o nível de ruído for alto, deve-se utilizar o filtro SIR, e se o nível de ruído for baixo, é preferível utilizar o filtro ASIR. Uma questão em aberto no que diz respeito ao uso dos filtros de partículas é como determinar o número de partículas otimizado a ser utilizado numa aplicação (um número de partículas muito elevado pode tornar a execução do filtro muito lenta, enquanto que um número baixo de partículas pode fazer com que o filtro não alcance a convergência).

3. Função Gaussiana inadequada para aproximar a função densidade de probabilidade $p(y/x)$ – No caso do pig inercial, uma função Gaussiana (como aquela que foi utilizada no filtro SIR) pode não ser uma boa aproximação para a função $p(y/x)$, e isto pode ter prejudicado o funcionamento do filtro SIR;
4. Finalmente, o ruído apresentado pelos sensores da Central Inercial neste teste foi mais alto que nos outros testes, o que pode ter degradado o desempenho do filtro SIR.

6. Referências Bibliográficas

- [1] ARULAMPALAM, S. et al. A Tutorial on Particle Filters for On-line Non-linear/Non-Gaussian Bayesian Tracking. *IEEE Transactions on Signal Processing*, vol 50, no. 2, pp. 174-188, Fevereiro de 2002.
- [2] GREWAL, M. S.; WEILL, L. R.; ANDREWS, A. P. *Global Positioning Systems, Inertial Navigation and Integration*. John Wiley & Sons, 2001.
- [3] PAPANOLIS, A.; PILLAI, S. U. *Probability, random variables and stochastic processes*, 4. Ed. New York: Mc Graw Hill, 2001.
- [4] DAVIS, M. H. A.; VINTER, R. B. *Stochastic Modelling and Control*. Chapman and Hall, 1985.
- [5] BERGMAN, N. *Recursive Bayesian Estimation: Navigation and Tracking Applications*. 1999. PhD Thesis – Linköping University. Sweden, 1999.

- [6] LIU, J. S.; CHEN, R. Sequential Monte Carlo Methods for Dynamical Systems. *Journal of the American Statistical Association*, vol 93, pp. 1032-1044, 1998.
- [7] DOUCET, A. Sequential Monte Carlo Methods for Bayesian Filtering. University of Cambridge, UK, 1998 (technical report).
- [8] GILKS, W. R.; RICHARDSON, S.; SPIEGELHALTER, D. *Markov Chain Monte Carlo in Practice*. London: Chapman and Hall, 1996.
- [9] GORDON, N.; SALMOND, D.; SMITH, A. F. M. Novel Approach to Non-linear and Non-Gaussian Bayesian State Estimation. *IEEE Proceedings*, vol 140, pp. 107-113, 1993.
- [10] PITT, M.; SHEPHARD, N. Filtering via Simulation: Auxiliary Particle Filters. *Journal of the American Statistical Association*, vol 94, pp. 590-599, 1999.
- [11] GEYER, C. J. Practical Markov Chain Monte Carlo. *Statistical Sci*, vol 7, pp. 473-511, 1992.
- [12] FEARNHEAD, P. Sequential Monte Carlo Methods in Filter Theory. 1998. PhD Thesis – Merton College, University of Oxford. UK, 1998.
- [13] KELLY, A. *Introduction to Mobile Robots, Position Estimation 4: Inertial Navigation Systems*. Outono de 1996.
- [14] STOVALL, S. H. *Basic Inertial Navigation*. California: Naval Air Warfare Center Weapons Division, September 1997 (report number NAWCWPNS TM 8128).
- [15] NEBOT, E.; DURRANT-WHYTE, H. Initial Calibration and Alignment of Low Cost Inertial Navigation Units for Land Vehicle Applications. *Journal of Robotics Systems*, v. 16, no. 2, pp. 81-92, Fevereiro de 1999.

- [16] GUSTAFSSON, F. et al. Particle Filters for Positioning, Navigation and Tracking. *IEEE Transactions on Signal Processing*, Special Issue on Monte Carlo Methods for Statistical Signal Processing, pp. 1-13, 2001.
- [17] CROSSBOW. *VG700AA User's Manual*. Setembro de 2002 (Document 7430-0074-01).
- [18] DOUCET, A.; GORDON, N.; DE FREITAS, J. F. G. *Sequential Monte Carlo Methods in Practice*. New York: Springer Verlag, 2001.
- [19] PEARL, J. *Probabilistic Reasoning in Intelligent Systems : Networks of Plausible Inference*. San Mateo: Morgan-Kaufmann, 1988.
- [20] CORMEN, T.; LEISERSON, C.; RIVEST, R. *Introduction to Algorithms*. Boston: Mc Graw-Hill, 1990.
- [21] VAN TREES, H. *Detection, Estimation and Modulation Theory - Part I*. John Wiley and Sons, 1968.
- [22] ROSS, S. *Introduction to Probability Models*. Academic Press, 1972.
- [23] GARCIA, C. *Modelagem e Simulação de Processos Industriais e Sistemas Eletromecânicos*. São Paulo: EDUSP, 1997.
- [24] MONTEIRO, L. H. A. *Sistemas Dinâmicos*. São Paulo: Livraria da Física Editora, 2002.
- [25] COWELL, R. G.; DAWID, A. P.; LAURITZEN, S. L.; SPIEGELHALTER, D. J. *Probabilistic Networks and Expert Systems*. New York: Springer-Verlag, 1999.
- [26] CHATFIELD, A. B. *Fundamentals of High Accuracy Inertial Navigation*. The American Institute of Aeronautics and Astronautics, vol. 174, 1997.

- [27] LAWRENCE, A. *Modern Inertial Technology - Navigation, Guidance and Control*. Segunda edição. New York: Springer-Verlag, 1998.
- [28] HOLMGREN, R. A. *A First Course in Discrete Dynamical Systems*. New York: Springer-Verlag, 1994.
- [29] VIK, B.; FOSSEN, T. I. *A Nonlinear Observer for Integration of GPS and Inertial Navigation Systems*. Norwegian University of Science and Technology, Maio de 2001.
- [30] GHAHRAMANI, Z. *An Introduction to Hidden Markov Models and Bayesian Networks*. *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 15, pp. 9-42, 2001.
- [31] JENSEN, F. V. *Bayesian Networks and Decision Graphs*. New York: Springer-Verlag, 2001.
- [32] PAVLOVIC, V. I.; FREY, B. J.; HUANG, T. S. *Time Series Classification Using Mixed State Dynamic Bayesian Networks*. *IEEE Transactions on Knowledge and Data Engineering*, vol. , pp. 609-615, 1999.
- [33] RUSSELL, S.; NORVIG, P. *Artificial Intelligence - A Modern Approach*. Prentice-Hall, 1995.
- [34] ACKOFF, R. L.; SASIENI, M. W. *Pesquisa Operacional (versão traduzida)*. São Paulo: EDUSP, 1971.
- [35] GEMAN, D.; GEMAN, S. *Stochastic Relaxation, Gibbs Distributions and the Bayesian restoration of Images*. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 6, pp. 721-741, 1984.

- [36] KALMAN, R.; BUCY, R. New Results in linear filtering and prediction theory. *Journal of Basic Engineering*, Transactions ASME Series D, vol 83, pp. 95-108, 1961.
- [37] METROPOLIS, N. et al. Equations of state calculations by fast computing machines. *The Journal of Chemical Physics*, vol. 21, pp. 1087-1091, 1953.
- [38] DOUCET, A. et al. Rao-Blackwellised Particle Filtering for Dynamic Bayesian Networks. In: **PROCEEDINGS OF THE CONFERENCE ON UNCERTAINTY IN ARTIFICIAL INTELLIGENCE**, Stanford, 2000.
- [39] FURUKAWA, C. M.; ADAMOWSKI, J. C.; CAMERINI, C. S. Ultrasonic Pig for Oil Pipeline Inspection. In: **IEEE INDUSTRY APPLICATIONS SOCIETY, III INDUSCON**: anais. São Paulo: IEEE Sul Brasil, Escola Politécnica da USP, 1998.
- [40] BACHMANN, E. R. et al. Rigid Body Dynamics, Inertial Reference Frames, and Graphics Coordinate Systems: A Resolution of Conflicting Conventions and Terminology. For submission to **IEEE Symposium on Computational Intelligence in Robotics and Automation (CIRA 2003)**, Abril de 2002.
- [41] LJUNG, L. **System Identification: Theory for the User**. Segunda edição. Englewood Cliffs, Prentice-Hall, 1999.
- [42] SODERSTRÖM, T.; STOICA, P. **System Identification**. U.K.: Hermel Hempstead, Prentice-Hall International, 1989.
- [43] CRAIG, J. J. **Introduction to Robotics - Mechanics and Control**. Segunda edição. Addison-Wesley, 1989.

- [44] GREWAL, M. S.; HENDERSON, V. D.; MIYASAKO, R. S. Application of Kalman Filtering to the Calibration and Alignment of Inertial Navigation Systems. *IEEE Transactions on Automatic Control*, vol. 36, no. 1, Janeiro de 1991.
- [45] TITTERTON, D. H.; WESTON, J. L. *Strapdown Inertial Navigation Technology*. London: IEE Radar, Sonar, Navigation and Avionics Series 5, 1997.
- [46] ROGERS, R. M. *Applied Mathematics in Integrated Navigation Systems*. Virginia, USA: AIAA Education Series, 2000.
- [47] BROWN, R. G.; HWANG, P. Y. C. *Introduction to Random Signals and Applied Kalman Filtering*. Terceira edição. John Wiley and Sons, Inc., 1997.
- [48] KAILATH, T. *Linear Systems*. New Jersey, USA: Prentice-Hall, Inc., Englewood Cliffs, 1980.
- [49] MATLAB versão 6.5.0.180913a Release 13. The Mathworks, Inc., Junho de 2002.
- [50] SANTANA, D. D. S.; CAMPOS, V. A. F.; FURUKAWA, C. M.; MARUYAMA, N. Estimação de Trajetórias utilizando Sistema de Navegação Inercial *Strapdown*. In: *XV Congresso Brasileiro de Automática: anais*, in press. A ser realizado em Gramado, Setembro de 2004.
- [51] CAMPOS, V. A. F.; SANTANA, D. D. S.; FURUKAWA, C. M.; MARUYAMA, N. Filtros de Partículas aplicados à estimação de trajetórias. In: *XV Congresso Brasileiro de Automática: anais*, in press. A ser realizado em Gramado, Setembro de 2004.
- [52] BOLIC, M.; DJURIC, P. M.; HONG, S. New Resampling Algorithms for Particle Filters. *IEEE*, 2003.

- [53] Escola Politécnica da USP, Diretrizes para apresentação de dissertações e teses. Agosto de 2001.
- [54] CAMPOS, V. A. F.; SANTANA, D. D. S.; FURUKAWA, C. M.; MARUYAMA, N. Trajectory Estimation of a Pig Using a Strapdown Low Cost Inertial Measurement Unit and Odometry. In: VI INDUSCON: anais, in press. A ser realizado em Joinville, Outubro de 2004.