

UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE ENGENHARIA MECÂNICA

Relatório Final
Trabalho de Conclusão de Curso II

Fusão de Sensores para Localização Espacial com Filtro
 \mathcal{H}_∞ em Sistema Markoviano

Autor: **Luiz Toledo**

Orientador: **Prof. Dr. André Ricardo Fioravanti**

Campinas, 15 de Novembro de 2015

UNIVERSIDADE ESTADUAL DE CAMPINAS

FACULDADE DE ENGENHARIA MECÂNICA

Relatório Final
Trabalho de Conclusão de Curso II

Fusão de Sensores para Localização Espacial com Filtro \mathcal{H}_∞ em Sistema Markoviano

Autor: **Luiz Toledo**

Orientador: **Prof. Dr. André Ricardo Fioravanti**

Curso: Engenharia de Controle e Automação

Trabalho de Conclusão de Curso, apresentado à Comissão de Graduação da Faculdade de Engenharia Mecânica, como requisito para a obtenção do título de Engenheiro de Automação e Controle.

Campinas, 2015
S.P. – Brasil

Agradecimentos

Este trabalho não poderia ser terminado sem a ajuda de diversas pessoas às quais presto minha homenagem:

Primeiramente agradeço a Deus por mais esse auxílio em minha vida.

Em especial, ao Prof. Dr. André Ricardo Fioravanti pelo auxílio na definição em conjunto do tema a ser abordado e pela literatura sugerida, permitindo a execução deste trabalho. Agradeço-o, também, pelo incentivo dado durante o processo.

À minha namorada Petiane, quem me auxiliou e me deu todo o suporte para a conclusão deste trabalho, acompanhando-me em diversas tardes de estudo.

Aos meus cunhados Eduardo e Daniel, enquanto que um me emprestou o celular, o outro me acompanhou por horas dirigindo o veículo, ambos componentes essenciais para a confecção deste trabalho.

Aos meus pais e padrastos por cederem o computador, seja em casa ou no trabalho, para que eu pudesse desenvolver o trabalho.

Aos meus amigos e demais familiares que me apoiaram ao longo do semestre.

	Resumo	3
	Lista de Figuras	4
	Lista de Tabelas	5
	Nomenclatura	5
	Abreviações e Terminologias	7
Capítulo 1	Introdução	8
Capítulo 2	Revisão Bibliográfica	10
2.1.	Acelerômetro	10
2.2.	Giroscópio	12
2.3.	Sistema de Posicionamento Global – GPS	15
2.4.	Geometria das Coordenadas Locais	17
2.5.	Sistemas Markovianos	20
2.6.	Filtro \mathcal{H}_∞	22
Capítulo 3	Procedimento Experimental	25
3.1.	Preparativos Iniciais	25
3.1.1.	Definição da Trajetória	25
3.1.2.	Identificação de Software de Aquisição	26
3.2.	Preparativos em Tempo de Execução	26
3.2.1.	Posicionamento do Sensor	26
3.2.2.	Parâmetros do Software	27
3.3.	Tratamento dos Dados Pós-Execução	28
3.3.1.	Aceleração Linear	28
3.3.2.	Velocidade Angular	29
3.3.3.	Coordenada GPS	30
3.4.	Definição do Sistema para Aplicação do Filtro	30
3.4.1.	Espaço de Estados	30
3.4.2.	Cadeia Markoviana	32
3.4.3.	Inserção da Cadeia Markoviana no Espaço de Estados	32
Capítulo 4	Resultados e Discussões	35
4.1.	Resultados dos Testes em Trajetória Real	35
4.1.1.	Validação dos Dados: GPS e Transformações	35
4.1.2.	Validação dos Dados: Acelerômetro e Giroscópio	36
4.2.	Simulação da Trajetória	37
4.3.	Filtragem da Trajetória Simulada	38
Capítulo 5	Conclusões	44
	Referências Bibliográficas	46
	Anexos	47

Resumo

Toledo, Luiz, *Fusão de Sensores para Localização Espacial com Filtro \mathcal{H}_∞ em Sistema Markoviano*, Faculdade de Engenharia Mecânica, Universidade Estadual de Campinas, Trabalho de Conclusão de Curso II, (2015).

É possível encontrar diversos sistemas eletrônicos no nosso cotidiano para os quais são necessárias previsões de valores em ambientes ruidosos. Para esse problema é possível utilizar o chamado filtro \mathcal{H}_∞ , um estimador de estados que minimiza o efeito do ruído nos sistemas e, através do seu equacionamento, é possível prever algumas variáveis do modelo em questão. Aliado ao estimador citado, utiliza-se aqui o conceito de saltos markovianos no equacionamento do sistema, definindo assim modos de operação. Esta inserção melhora ainda mais na precisão da previsão, tornando o sistema de medidas mais robusto.

Neste trabalho, buscamos realizar a fusão da informação de sensores, a saber, GPS, acelerômetro e giroscópio, atualmente presentes em telefones celulares, para realizar a localização espacial de um veículo, portanto, sistemas e dispositivos corriqueiros utilizados de maneira conjunta.

Um dos principais pontos que motiva esse trabalho é melhorar cada vez mais a precisão do posicionamento de um veículo utilizando um instrumento portátil, por exemplo, um telefone celular, minimizando efeitos de ruído, má precisão dos sensores e até mesmo eventuais perdas de sinal do GPS.

Palavras Chave: Previsão, filtragem robusta, filtro \mathcal{H}_∞ , sistema com saltos markovianos, GPS, acelerômetro, giroscópio, fusão de sensores.

Lista de Figuras

Figura 2.1. Exemplos de acelerômetro	10
Figura 2.2. Conceito de funcionamento do acelerômetro unidimensional de massa	11
Figura 2.3. (E): Caracterização dos elementos de um acelerômetro de capacitor; (D): Princípio de funcionamento do dispositivo	12
Figura 2.4. Circuito contendo um acelerômetro de capacitor com dimensões reduzidas	12
Figura 2.5. Exemplos de giroscópio	13
Figura 2.6. Conceito de funcionamento do giroscópio de massa unidimensional	13
Figura 2.7. Esquema de funcionamento do giroscópio capacitivo, utilizado em aparelhos de celular	14
Figura 2.8. (E) Circuito contendo um giroscópio de capacitor com dimensões reduzidas; (D) Variações em todas as dimensões	14
Figura 2.9. Segmentação do globo em fusos e zonas – em destaque a região de São Paulo - Brasil	15
Figura 2.10. Segmentação do globo em latitude e longitude	16
Figura 2.11. Placa de circuito impresso contendo um receptor de sinal GPS	17
Figura 2.12. Exemplos dos referenciais utilizados em Smartphones	18
Figura 2.13. Plano Tangente Local	18
Figura 2.14. Representação da elipse para mudança de coordenada	19
Figura 2.15. Cadeia de Markov com quatro estados	21
Figura 3.1. Diagrama simplificado do problema proposto	25
Figura 3.2. Trajetória selecionada para a execução do teste; em destaque, a presença de um túnel	25
Figura 3.3. Tela do aplicativo de celular “AndroSensor”	26
Figura 3.4. Direção de aceleração linear do celular coincidente com a direção de movimento do veículo	27
Figura 3.5. Representação gráfica da precisão do sinal do GPS	28
Figura 3.6. Decomposição dos vetores de aceleração ao longo de uma trajetória	29
Figura 4.1. Comparação da trajetória definida no plano de estudos com os dados obtidos	35
Figura 4.2. Comparação da trajetória definida no plano de estudos com os dados obtidos e com os dados calculados em PTL	36
Figura 4.3. Trajetória gerada em simulação	37
Figura 4.4. Acelerações para a trajetória gerada em simulação	38
Figura 4.5. Destaque das faixas de operação no traçado simulado	39
Figura 4.6. Comparação entre a trajetória gerada e a estimada na direção D	42
Figura 4.7. Comparação entre a trajetória gerada e a estimada na direção E	42
Figura 4.8. Comparação entre a trajetória gerada e a média estimada na direção D	43

Figura 4.9. Comparação entre a trajetória gerada e a média estimada na direção E	43
Figura A.1. Posição, nas direções D, E e N da trajetória simulada	47
Figura A.2. Velocidade, nas direções D, E e N, da trajetória simulada	47

Lista de Tabelas

Tabela 2.1. Características da Terra enquanto elipse	18
Tabela 3.1. Parâmetros coletados no teste	27
Tabela 3.2. Detalhe dos modos de operação	32
Tabela 4.1. Parâmetros coletados no teste	36

Nomenclatura

Letras Latinas

\mathbb{N}	Conjunto dos números Naturais	-
\mathbb{R}	Conjunto dos números Reais	-
x, y, z	Denominação dos eixos cartesianos	[m]
P	Com subscrito, Ponto qualquer no plano	-
P	Sem subscrito, Matriz de probabilidades	-
F	Força	[N]
d	Deslocamento	[m]
K	Constante de rigidez de uma mola	[N/m]
a	Com subscrito, Aceleração	[m/s ²]
a	Sem subscrito, Eixo maior da elipse	[m]
b	Eixo maior da elipse	[m]
e	Incerteza GPS e Acelerômetro	[m] e [m/s ²]
ex	Sem subscrito, Excentricidade da elipse	-
pl	Planicidade da elipse	-
m	Massa	[kg]
$X, X' e X''$	Vetor de posições	-
x	Vetor de estados	-
y	Vetor de medições	-
z	Vetor de predições	-

A, J, C, E	Matrizes de estado	-
F, G, L	Sistemas de equações de estado	-
T	Período	[s]
g	Aceleração da gravidade	[m/s ²]
s	Posição	[m]
v	Velocidade	[m/s]
J	Matriz Identidade	-
Z	Matriz de zeros	-
f	Frequência de atualização dos sensores	[hz]
ip_4	Ajuste das probabilidades	-
Q	Ruído de medição unidimensional	-

Letras Gregas

λ	Latitude	[°]
ϕ	Longitude	[°]
φ	Posição angular	[°]
ω	Velocidade angular	[°/s] ou [rad/s]

Superescritos

n	Dimensão do conjunto dos números Reais
T	Transposta (em vetores de posição)

Subscritos

N	Dimensão utilizado (em Conjunto dos números Naturais)
lin	Linear (em posição, velocidade ou aceleração)
$plano$	Indicando um ponto no plano (em ponto P)
UTM	Indicando o referencial utilizado (em ponto P)
$Dec - Deg$	Indicando o referencial utilizado (em ponto P)
s	Amostragem (em período)
PTL	Indicando o referencial utilizado (em ponto P)
acelerômetro	Indicar de qual sensor de fala (em frequência f)
GPS	Indicar de qual sensor de fala (em frequência f)

<i>f</i>	Fazer referência ao filtro (em vetores e matrizes de estado)
<i>real</i>	Fazendo referencia ao valor puro (em aceleração <i>a</i>)

Abreviações e Terminologia

Matlab	Linguagem de programação especializado em matrizes e matemática para o uso nas universidades e indústrias para modelagem matemática;
UTM	Universal Transversa Mercator
GPS	Do inglês: <i>Global Positioning System</i> é um sistema que utiliza Satélites para identificar uma determinada posição espacial;
PTL	Plano Tangente Local definido para as conversões de unidades e referenciais entre GPS e demais sensores;
MJLS	Do inglês: <i>Markov Jump Linear Systems</i> (trdução: Sistemas Lineares com Saltos Markovianos)
Dec-Deg	Do inglês: <i>Decimal Degrees</i> (tradução: Graus Decimais)

Capítulo 1

Introdução

Como consequência do desenvolvimento tecnológico atual, depara-se com a miniaturização de componentes eletrônicos, permitindo um acúmulo maior de funções em um sistema embarcado. Por exemplo, aparelhos celulares possuem diversas funções que são possíveis apenas por conta da presença de sensores instalados em seu *hardware*.

Uma das razões do tendente acúmulo de opções em um aparelho celular é o ganho de tempo e consolidação de diversos aparelhos dedicados em um. Isso ocorre à medida que o mundo se globaliza, o tempo se torna mais escasso e qualquer forma de economia de tempo é bem-vista. Mesmo o mais simples dos aparelhos celulares pode conter os mais diversos sensores como acelerômetros e GPS.

Aliado ao ganho de tempo, percebemos um uso extensivo atual dos celulares com a função de GPS. Aplicativos instalados conectam-se com a Internet e conseguem ter acesso ao mapa das ruas de uma determinada cidade. Com o auxílio do GPS nele presente, o aplicativo interpreta a posição de um veículo e, uma vez inserido o endereço de destino, é possível fugir do trânsito das grandes cidades para chegar com mais agilidade.

Apesar da presença de diversos minissistemas em um único aparelho, a tendência é que cada função não funcione com grande precisão, pois sistemas menos dedicados podem perder desempenho. Como dito anteriormente, o objeto de estudo deste Trabalho de Conclusão de Curso será a fusão de dois sensores presentes no aparelho, acelerômetro e GPS, a fim de realizar a localização espacial de um veículo. É sabido, porém, que um dos principais desafios será eliminar o ruído proveniente dos dados enviados por estes sensores.

Um método matemático presente na literatura, conhecido por filtro \mathcal{H}_∞ , é comumente usado para realizar a filtragem em modelos de estado, minimizando os efeitos do ruído de medição, com o intuito de prever os valores das variáveis deste modelo sem sua medição ou com medições ruidosas. Por exemplo, pode ser citada a publicação [9] que trata do mesmo tema, realizando a filtragem robusta de sistemas ditos markovianos. Ainda, os trabalhos [7] e [8] existentes na literatura abordam, adicionalmente, tratando da transmissão dos dados obtidos, ou seja, das medidas, através de redes. Já em [6], além da filtragem, a novidade abordada é o controle robusto do sistema em questão.

Comparado ao famoso filtro de Kalman, o filtro \mathcal{H}_∞ tende a ter uma resposta mais lenta, de menor desempenho, porém com grande capacidade de rejeição de ruídos e, portanto, maior robustez. Esta propriedade é a principal característica que buscamos em nosso sistema de fusão de dados e, desta forma, o filtro \mathcal{H}_∞ se torna nossa ferramenta de escolha.

Adicionalmente, outra obra existente na literatura merece ser lembrada por ser tão completa no que diz respeito à filtragem de processos markovianos, sendo ela [1].

A sequência a ser percorrida por esse trabalho, após contextualizar o tema relacionando-o com diversas obras disponíveis na Literatura no Capítulo 1, é apresentar no Capítulo 2 a teoria que envolve o tema. Versar-se-á desde a descrição e funcionamento dos componentes eletrônicos responsáveis por realizar as medidas (dados de entrada do sistema), passando pela definição do filtro \mathcal{H}_∞ propriamente dito, bem como de sistemas markovianos, até a modelização do sistema, fundindo as variáveis dos dois sensores. Uma vez pronto o modelo do sistema, o Capítulo 3 deste trabalho abordará os detalhes dos procedimentos para validação prática do tema central, mencionando os dispositivos usados e, inclusive, explicitando as matrizes de estado (aquelas que caracterizam um sistema em específico) para a aplicação dos conceitos teóricos sobre os dados medidos em campo. No Capítulo 4 se apresentarão os resultados obtidos em campo, bem como alternativas para testar o procedimento de estimação de estado que será proposto, baseado em simulações. Conclui-se, portanto, todos os resultados e métodos apresentados no Capítulo 5.

Capítulo 2

Revisão Bibliográfica

Neste Capítulo serão apresentados os principais elementos e conceitos que servirão de base nos próximos Capítulos que compõem este projeto, facilitando a compreensão e organização do mesmo. Ademais, facilitando a compreensão de algumas notações e definições matemáticas que virão na sequência, adotar-se-á como \mathbb{N}_N o conjunto dos N primeiros números naturais, \mathbb{R}^n como conjunto dos números reais de dimensão n , a tríade (x, y, z) sendo as coordenadas cartesianas retangulares de um ponto P_{plano} . Ainda, o mesmo ponto P_{plano} poderá ser representado por coordenadas polares, (λ, ϕ) , omitindo-se o raio que, neste caso, é o raio do planeta Terra. Mais adiante, x representará o vetor de estados do sistema em questão no tempo contínuo, \dot{x} como sendo sua derivada. Em tempo discreto, $x(k)$ é o equivalente ao x , e $x(k+1)$ ao vetor \dot{x} . Já $\tilde{x}(k+1)$ representa um sinal de erro. Representar-se-á por P (sem subscrito) a matriz de probabilidades.

2.1 Acelerômetro

Acelerômetros são dispositivos que podem funcionar a partir de diversos efeitos físicos para medir valores de aceleração de um determinado corpo. Eles possuem uma ampla faixa de valores de aceleração que são capazes de medir, logo, possuem por consequência uma gama de aplicações bastante elevada.

Esses dispositivos são usados principalmente em sistemas de posicionamento, sensores de inclinação, sistema esse que será abordado mais detalhadamente nos próximos Capítulos deste trabalho.

Existem muitos exemplos diferentes deste dispositivo que é largamente utilizado em aplicações industriais. Alguns exemplos podem ser vistos abaixo na através da Figura 2.1.



Figura 2.1. Exemplos de acelerômetro. Fonte: <http://pt.aliexpress.com/> em 05/09/15

Uma curiosidade sobre um acelerômetro é que ele é incapaz de medir a aceleração de um objeto em queda livre, por exemplo. A sensação de peso em um acelerômetro em queda livre no sol ou na lua é nula, embora a aceleração seja bastante diferente nesses casos.

O conceito de funcionamento de um acelerômetro pode ser comparado a um sistema massa-mola comumente aplicado na literatura. Através do esquema abaixo (Figura 2.2), pode ser vista uma massa de prova presa por duas molas, em cada uma de suas extremidades. Uma vez que a carcaça (seu referencial inercial) é submetida a uma aceleração, ocorre um deslocamento (d) desta massa (m).

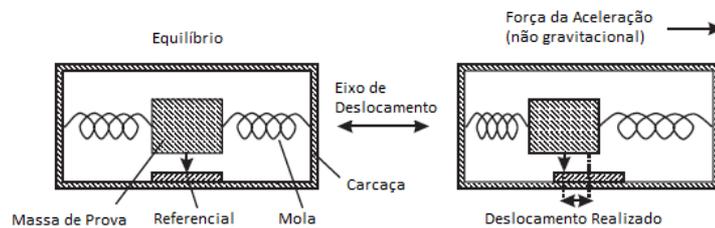


Figura 2.2. Conceito de funcionamento do acelerômetro unidimensional de massa.

Fonte: Adaptado de [4]

Uma vez conhecidas todas as características dos elementos utilizados no esquema (rigidez (K) das molas, massa exata da massa de prova, por exemplo), é possível saber a força (F) aplicada à mola e, conseqüentemente, a aceleração linear (a_{lin} , [m/s^2]) sofrida pelo conjunto. A equação abaixo mostra a relação entre os elementos:

$$a_{lin} = \frac{F}{m} \quad (1)$$

Onde:

$$F = \frac{K}{d} \quad (2)$$

Com a miniaturização dos componentes eletrônicos, é possível encontrar sensores como este, cujo princípio de funcionamento é a variação de potencial de um capacitor.



Figura 2.3. (E): Caracterização dos elementos de um acelerômetro de capacitor; (D): Princípio de funcionamento do dispositivo. Fonte: Adaptado de <http://www.decom.ufop.br/> em 05/09/15

Fazendo uso deste princípio de funcionamento, tornou-se possível embarcar o dispositivo para que pudesse ser utilizado, por exemplo, em aparelhos celulares (*smartphones*). Na Figura 2.4 abaixo é possível ver um circuito contendo um acelerômetro, possuindo apenas alguns milímetros de comprimento, largura e espessura.



Figura 2.4. Circuito contendo um acelerômetro de capacitor com dimensões reduzidas. Fonte: <https://multilogica-shop.com> em 05/09/15

No exemplo acima, é possível ver através da imagem a composição do dispositivo nos eixos x e y . Existe, porém, outras variantes deste sensor, sendo possível encontrá-lo na configuração 3D, ou seja, capaz de medir os valores de aceleração aos quais um aparelho de celular é submetido nas três dimensões (x , y e z). Os resultados preliminares deste trabalho contaram com dados obtidos com o auxílio de um sensor 3D, que serão abordados com mais detalhes no Capítulo 3.

2.2 Giroscópio

Giroscópios são dispositivos que medem a taxa de variação angular de um determinado objeto. Em outras palavras, este sensor é capaz de medir a velocidade angular (ω , [rad/s] ou [$^\circ/s$]) de um eixo específico ligado ao objeto em questão.

Assim como o acelerômetro, o giroscópio é largamente aplicado na indústria, desde a linha de produção de grandes usinas até a aviação civil, indicando a inclinação de um

avião, por exemplo. Na Figura 2.5 abaixo, são ilustradas algumas formas que este dispositivo se apresenta:



Figura 2.5. Exemplos de giroscópio. Fonte: <http://www.dreamstime.com/> e <http://www.rion-tech.net> em 19/09/15

Conceitualmente, a informação proveniente deste sensor pode ser fruto de um sistema massa-mola comum, de maneira análoga ao item 2.1 deste Capítulo. Na Figura 2.6 abaixo, mostra-se um esquema conceitual no qual uma massa é fixada por quatro molas. Quando existe um movimento relativo à carcaça, as molas se estendem/comprimem, fazendo como que o referencial móvel indicado na figura acuse um movimento com relação à carcaça (referencial fixo).

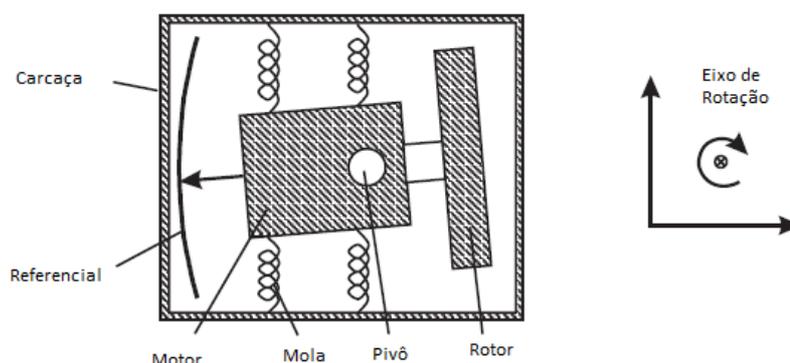


Figura 2.6. Conceito de funcionamento do giroscópio de massa unidimensional.

Fonte: Adaptado de [4]

Existem outros tipos mais precisos deste sensor diferentes deste de massa, como por exemplo giroscópios óticos, de vibração e de capacitor. Este último tipo acusa a velocidade angular de um determinado eixo através da variação de potencial elétrico, conforme a Figura 2.7 abaixo. Este sensor, sendo ele componente de um aparelho celular, foi utilizado durante os testes preliminares deste trabalho.

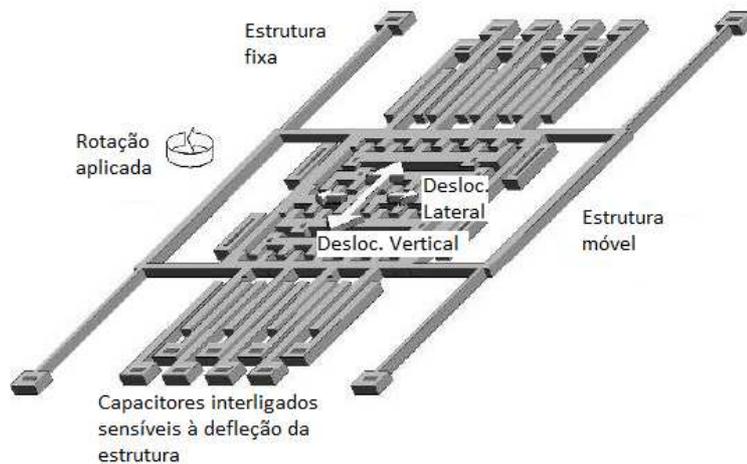


Figura 2.7. Esquema de funcionamento do giroscópio capacitivo, utilizado em aparelhos de celular. Fonte: Adaptado de <http://abertoatedemadrugada.com/> em 19/09/15

A miniaturização dos componentes eletrônicos é também presente neste caso. É possível ver na Figura 2.8 um circuito eletrônico contendo um giroscópio, que pode ser empregado como funcionalidade em um aparelho de celular, por exemplo. Mostra-se, inclusive, um esquema das variações angulares que este sensor é capaz de identificar em 3D.

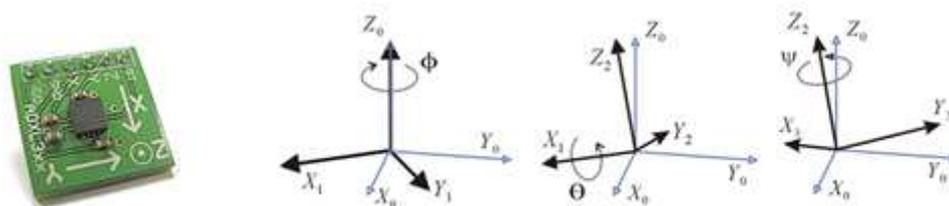


Figura 2.8. (E) Circuito contendo um giroscópio de capacitor com dimensões reduzidas; (D) Variações em todas as dimensões. Fonte: www.webtronico.com e www.wikiwand.com em 20/09/15

Outra aplicação que merece ser citada, fruto das dimensões reduzidas deste sensor, é a utilização do mesmo em jogos de videogame, tornando a experiência de jogo mais interativa e real.

2.3 Sistema de Posicionamento Global – GPS

O sistema de posicionamento global (do inglês, GPS¹) é um sistema no qual satélites orbitando o planeta Terra, realizando órbitas quase que circulares (26.500 km de raio), fornecem a um determinado aparelho apto a receber seu sinal, a informação sobre sua posição. As condições atmosféricas podem interferir na intensidade ou qualidade deste sinal, tornando-o mais fraco em dias nublados ou chuvosos. Ainda, para o bom funcionamento deste sistema, é necessário que o aparelho receptor não tenha obstáculos físicos com relação aos satélites. Por exemplo, em um veículo dentro de um túnel, a qualidade do sinal seria extremamente baixa (ou mesmo, em alguns casos, nula).

Por se tratar de um sistema de posicionamento, entende-se que é provida ao corpo móvel receptor uma informação de posição. Porém, este dado de posição pode ser informado sob diversos formatos e padrões mundiais. A grande diferença entre os padrões de posição são intrínsecos da geometria básica, uma vez que um ponto no espaço, de maneira simples, pode ser caracterizado por coordenadas esféricas e cartesianas. O sistema UTM² é um dos mais conhecidos e utilizados para referenciar um corpo no espaço geográfico de forma cartesiana e bidimensional. Ele foi concebido baseando-se na projeção cilíndrica, proposta em 1950 para abranger todas as longitudes. Divide-se o globo em fusos e zonas conforme mostrado na imagem abaixo (Figura 2.9).

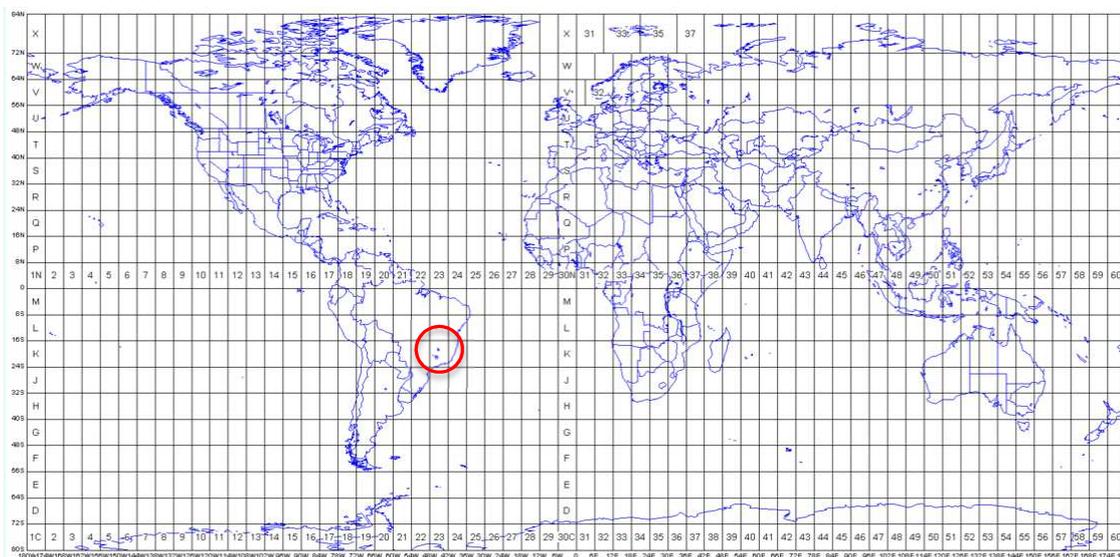


Figura 2.9. Segmentação do globo em fusos e zonas – em destaque a região de São Paulo - Brasil. Fonte: www.androidheadlines.com em 11/10/15

¹ GPS: *Global Positioning System*;

² UTM: *Universal Transversa Mercator*;

O formato desta notação pode ser visto através da Equação 3, que descreve um ponto P_{plano} pertencente à região de destaque no mapa em vermelho (São Paulo – Brasil):

$$P_{UTM} = 23K \ 338346mE \ 7396550mN \quad (3)$$

Detalhando a informação acima, o ponto P em questão está no quadrante indicado no mapa “23K”, a 338.346 metros da origem na direção Leste e 7.396.550 metros da origem na direção Norte.

Em se tratando de coordenadas esféricas, o mais usual é o formato *Dec-Deg*³ (graus decimais). Neste sistema, existem algumas variações nas quais se considera ora a Terra uma esfera aproximada, ora uma elipse. A localização, todavia, é fornecida através de dois ângulos (latitude e longitude), mostrado no exemplo abaixo para o mesmo ponto P indicado no mapa da Figura 2.9.

$$P_{Dec-Deg} = (\lambda, \phi) = (-23,5332990^\circ, -46,5835390^\circ) \quad (4)$$

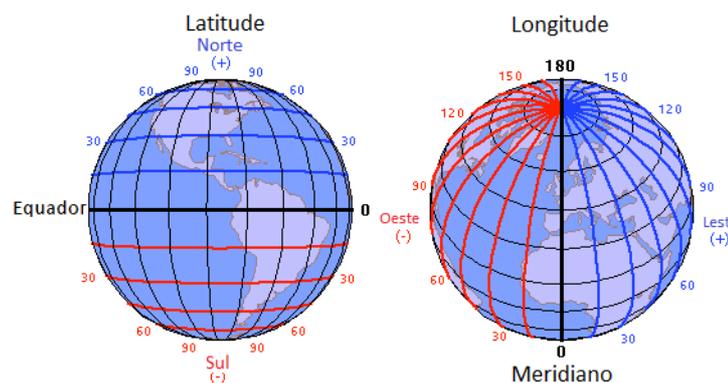


Figura 2.10. Segmentação do globo em latitude e longitude. Fonte: www.snipview.com em 11/10/15

A conversão numérica entre os sistemas envolve, via de regra, geometria básica, sendo possível encontrar na literatura uma série de programas e códigos já prontos. Tudo depende da finalidade que se deseja empregar as coordenadas. Além destes, é possível converter os padrões supracitados para a coordenada cartesiana 3D pura e simples, fornecendo ao mesmo ponto P_{plano} as componentes (x, y, z) . Mais detalhes desta conversão serão abordados neste mesmo Capítulo, item 2.4.

³ Dec-Deg: Do inglês, Decimal Degrees;

De maneira análoga ao que foi visto sobre acelerômetros e giroscópios, o receptor de sinal de GPS teve suas dimensões reduzidas com o crescente desenvolvimento da eletrônica, podendo ser embarcado em uma vasta gama de aparelhos e aplicações. Encontra-se este dispositivo atualmente em veículos, em aparelhos celulares, aviões, navios, etc. Ou seja, é um quesito fundamental quando se tem a necessidade de rastrear um objeto ou auxiliá-lo a cumprir uma trajetória. Na imagem a seguir (Figura 2.11) pode-se vê-lo enquanto componente de uma placa de circuito impresso presente em *smartphones*. Neste trabalho foi utilizado um componente semelhante para fazer aquisição nas etapas preliminares, sendo ele integrante de um aparelho de celular.



Figura 2.11. Placa de circuito impresso contendo um receptor de sinal GPS. Fonte: www.androidheadlines.com em 11/10/15

2.4 Geometria das Coordenadas Locais

Quando se trabalha com informações provenientes de diferentes tipos de sensores e sistemas, é imprescindível que se coloquem todos os dados no mesmo referencial. Neste trabalho, é feita a fusão de dados do acelerômetro, giroscópio e GPS para melhor estimar a localização espacial de um dado objeto.

Um artifício utilizado é converter as coordenadas de GPS (latitude e longitude) em cartesianas, sendo transformadas e aproximadas em um plano tangente na superfície terrestre. Isto é possível, pois as trajetórias que foram simuladas e que serão expostas na sequência são pequenas com relação às dimensões da Terra, de modo que o raio de curvatura da terra, sendo ela uma elipse ou esfera, não influencia nos cálculos.

Nesta etapa, considera-se que os dados obtidos pelo acelerômetro e giroscópio estão em um referencial local (plano cartesiano composto pela tríade $(x, y$ e $z)$), cujos eixos são explicitados a seguir.

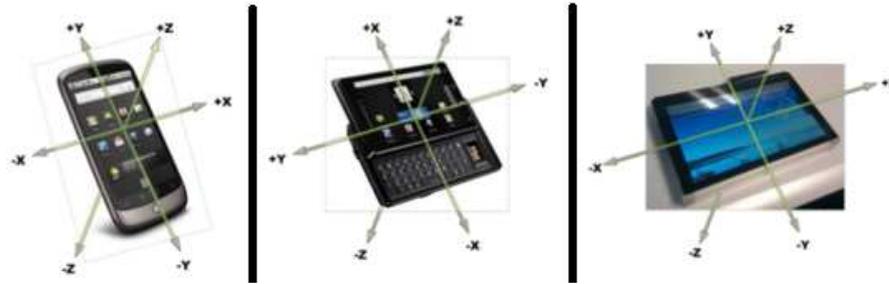


Figura 2.12. Exemplos dos referenciais utilizados em Smartphones. Fonte: Adaptado de <http://www.decom.ufop.br> em 24/10/15

Para fundir os dados destes sensores com os do GPS, que neste caso estão no formato apontado pela Equação 4 ($P_{plano} = (latitude, longitude) = (\lambda, \phi)$) é necessário convertê-lo ao mesmo referencial, que será chamado de PTL⁴ (Figura 2.13).

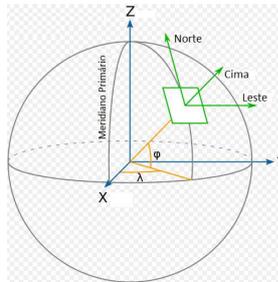


Figura 2.13. Plano Tangente Local. Fonte: Adaptado de <http://www.decom.ufop.br> em 24/10/15

Dentro do conceito de PTL, assume-se para os devidos fins que a Terra é um elipsoide, cujas dimensões e características podem ser vistas na Tabela 2.1 abaixo.

Tabela 2.1. Características da Terra enquanto elipse. Fonte: [3]

Constante	Descrição	Valor	Unidade
a	Eixo Maior	6.378.137,0000	[m]
b	Eixo Menor	6.356.752,3142	[m]
ex	Excentricidade	Equação 5	adimensional
pl	Planicidade	Equação 6	adimensional

Seguem abaixo as Equações 5 e 6 destacadas na Tabela 2.1:

⁴ PTL: Plano Tangente Local;

$$pl = \frac{a - b}{a} \quad (5)$$

$$ex = \sqrt{pl(2 - pl)} \quad (6)$$

A representação da elipse e suas constantes (Figura 2.14) permite a manipulação geométrica de mudança de coordenadas. A distância $N(\lambda)$ da superfície terrestre ao eixo z ao longo da linha normal da elipsoide é calculada pela Equação 7.

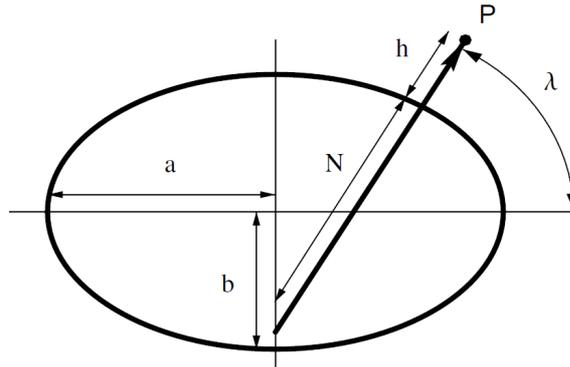


Figura 2.14. Representação da elipse para mudança de coordenada. Fonte: [3]

$$N(\lambda) = \frac{a}{\sqrt{1 - ex^2 \text{sen}^2(\lambda)}} \quad (7)$$

Dessa forma, é possível escrever as coordenadas cartesianas da seguinte forma:

$$X = \begin{bmatrix} x = (h + N)\cos\lambda \cos\phi \\ y = (h + N)\cos\lambda \text{sen}\phi \\ z = (h + (1 - ex^2)N)\text{sen}\lambda \end{bmatrix} \quad (8)$$

Sendo $X_0 = (x_0, y_0, z_0)^T$ a origem do PTL, transforma-se a origem do centro da Terra à origem do PTL subtraindo uma da outra, criando uma nova coordenada transladada X' paralela ao plano $X = (x, y, z)^T$ original.

$$X' = \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} - \begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix} \quad (9)$$

Em seguida, define-se uma nova coordenada $X'' = (x'', y'', z'')^T$ rotacionando de ϕ o sistema em torno do eixo z' , da forma:

$$X'' = \begin{bmatrix} x'' \\ y'' \\ z'' \end{bmatrix} = \begin{bmatrix} -\text{sen}\phi & \text{cos}\phi & 0 \\ \text{cos}\phi & \text{sen}\phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} \quad (10)$$

Finalmente, $X_{PTL} = (x_{PTL}, y_{PTL}, z_{PTL})^T$ é criado o sistema através de uma nova rotação de λ em torno do eixo x' com a Equação 11 abaixo:

$$X_{PTL} = \begin{bmatrix} x_{PTL} \\ y_{PTL} \\ z_{PTL} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -\text{sen}\lambda & \text{cos}\lambda \\ 0 & \text{cos}\lambda & \text{sen}\lambda \end{bmatrix} \cdot \begin{bmatrix} x'' \\ y'' \\ z'' \end{bmatrix} \quad (11)$$

O vetor X_{PTL} , portanto, está no mesmo PTL que os dados obtidos neste trabalho através do celular, tanto para aceleração quanto para velocidade angular, que serão explicitados nos tópicos seguintes.

2.5 Sistemas Markovianos

Muitos trabalhos foram publicados acerca de sistemas markovianos, ou sob a denominação de sistemas lineares sujeitos a saltos markovianos ou ainda simplesmente MJLS⁵. Pode ser citado como exemplo [9], que trata de MJLS em sistemas discretos, semelhante ao que será realizado neste trabalho.

Um sistema markoviano nada mais é do que um sistema que apresenta mudanças repentinas na sua estrutura ou em seus parâmetros. Em aplicações físicas, sistemas com essa característica modelam problemas onde, por exemplo, a estrutura da planta sofre mudanças abruptas e/ou aleatórias. Isto pode ser decorrente de eventuais falhas estruturais.

A fim de ilustrar melhor um sistema dito markoviano, ou seja, sujeito a saltos de probabilidade entre estados, considera-se um sistema que possa apresentar mais de um modo de operação, cada um possuindo suas próprias equações que o descrevem, em particular equações estas diferenciais (tempo contínuo) ou a diferenças (tempo discreto) lineares e invariantes no tempo. O sistema, então, irá alterar seu modo de operação de

⁵ Do inglês: *Markov Jump Linear Systems*

acordo com uma cadeia de Markov, isto é, a probabilidade dele passar de um modo de operação a outro depende apenas do seu estado atual. Para fins de definição, considera-se ainda que sejam conhecidas todas estas probabilidades de transição. Na prática, a probabilidade pode ser calculada em muitos casos, uma vez que os efeitos são conhecidos. A Figura 2.15 abaixo apresenta um sistema com quatro modos, onde os círculos representam os diferentes modos de operação e os arcos com setas são as respectivas probabilidades de transição.

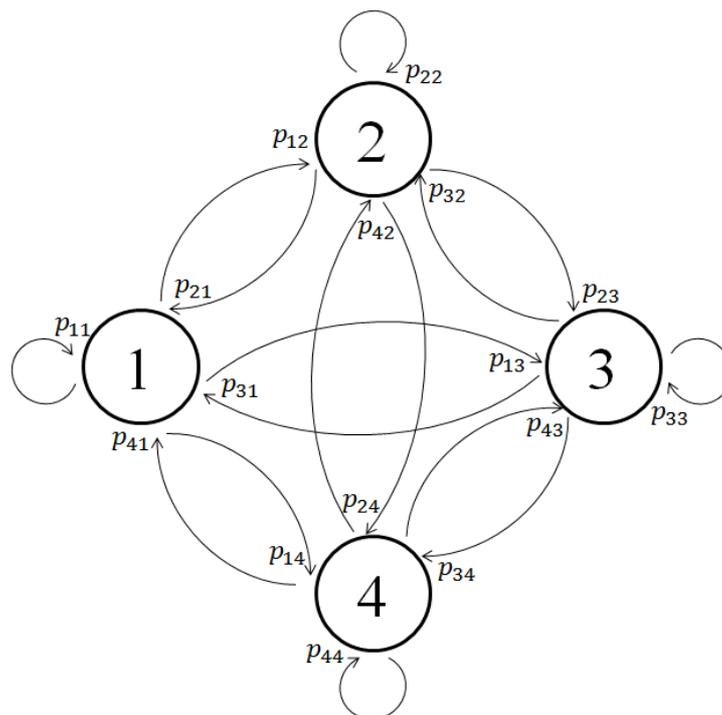


Figura 2.15. Cadeia de Markov com quatro estados. Fonte: Autoria própria

Os sistemas markovianos não terão sua teoria aprofundada por não fazer parte do objeto principal deste trabalho. Em consequência disso, o conceito de estabilidade, muito comum na literatura, não será explicitado. Mais informações da literatura podem ser encontradas em [5] ou até mesmo em [1] com maior riqueza de detalhes. Apresentar-se-á apenas algumas definições.

Supondo o sistema linear contínuo descrito pelas equações abaixo:

$$\mathbb{F} : \begin{cases} \dot{x} = Ax + Jw \\ z = C_z x + E_z w \\ y = C_y x + E_y w \end{cases} \quad (12)$$

Este sistema contínuo \mathbb{F} é discretizado, obtendo o sistema representado por \mathbb{G} . Por hipótese, o sistema linear discreto abaixo está sujeito a saltos markovianos, onde $x(k) \in \mathbb{R}^n$ é o estado do sistema, $w(k) \in \mathbb{R}^m$ representa perturbações externas, $z(k) \in \mathbb{R}^r$ é a saída a ser estimada, $y(k) \in \mathbb{R}^q$ a saída medida e $\theta_k = \theta(k)$ é uma variável aleatória tal que $\{\theta(k); k \in \mathbb{N}\}$ caracteriza uma cadeia de Markov.

$$\mathbb{G}: \begin{cases} x(k+1) = A(\theta_k)x(k) + J(\theta_k)w(k) \\ z(k) = C_z(\theta_k)x(k) + E_z(\theta_k)w(k) \\ y(k) = C_Y(\theta_k)x(k) + E_Y(\theta_k)w(k) \end{cases} \quad (13)$$

Todas as matrizes de estado representadas no sistema \mathbb{G} acima dependem também de uma cadeia de Markov, que assume valores em um conjunto finito $\mathbb{N}_N = \{1, \dots, N\}$, onde N é o número de modos de operação do sistema e a matriz de probabilidades de transição $P \in \mathbb{M}^N$ é dada por $P = [p_{ij}]$ com:

$$p_{ij} = \text{Prob}(\theta_{k+1} = j \mid \theta_k = i) \quad (14)$$

Tem-se desta forma que as restrições: $p_{ij} \geq 0$ e $\sum_{j=1}^N p_{ij} = 1$ para todo $(i, j) \in \mathbb{N}_N \times \mathbb{N}_N$ devem ser respeitadas.

Na literatura podem ser encontrados diversos exemplos de modelos que realizam a fusão de sensores para melhorar a qualidade da informação. Ainda, diversos sistemas de filtragem são igualmente disponíveis para o mesmo tipo de problema, fazendo uso de Filtro de Kalman, ou até mesmo filtragens mais robustas como \mathcal{H}_∞ . Porém, aplicar o conceito de sistemas sujeitos a saltos markovianos na fusão de sensores é um diferencial neste estudo. A probabilidade de transição entre os modos do sistema melhora a qualidade do valor da variável estimada.

2.6 Filtro \mathcal{H}_∞

O problema tratado neste trabalho concerne na fusão de sensores, aplicando um filtro no sistema markoviano, como mencionado no item anterior, a fim de melhor estimar a posição de um determinado corpo no espaço. Optou-se aqui pelo filtro \mathcal{H}_∞ por ser robusto, sendo capaz de lidar com as incertezas paramétricas que este caso apresenta, limitando o erro intrínseco da estimação. Conforme já versado no Capítulo 1, este filtro é frequentemente utilizado na literatura, como por exemplo, em [5], [6] ou [7].

Antes de adentrar ao equacionamento do filtro \mathcal{H}_∞ , é prudente passar brevemente pelo conceito da norma \mathcal{H}_∞ . Por hipótese, tem-se que os parâmetros da cadeia de Markov sejam mensuráveis. Tomando por base o sistema discreto \mathbb{G} descrito pela Equação 13, o objetivo desta norma é estabelecer um limitante para a influência da perturbação w na saída z do sistema. Define-se, assim, a norma para sistemas lineares sujeitos a saltos markovianos, assumindo que \mathbb{G} seja estável:

$$\|\mathbb{G}\|_\infty^2 = \sup_{0 \neq w \in \mathcal{L}_2^m, \theta_0 \in \mathbb{N}_N} \frac{\|z\|_2^2}{\|w\|_2^2} \quad (15)$$

A Equação acima pode ser lida de uma maneira mais simples, como por exemplo, busca-se encontrar qual pior saída possível dentre todas as combinações de z e w . Assim, se esta saída for limitada, todas as outras saídas também o serão. Existem algumas premissas que permitem a utilização da norma em questão, mas pelo mesmo motivo citado anteriormente, não se entrará em detalhes teóricos da norma, uma vez que é tida neste tema enquanto ferramenta. As condições de aplicabilidade da norma podem ser vistas nas supracitadas referências bibliográficas.

Com relação ao filtro propriamente dito, a construção da sua base teórica inicia-se pela seguinte definição, associada ao sistema apresentado pela Equação 13:

$$\mathbb{L}: \begin{cases} x_f(k+1) = A_f(\theta_k)x_f(k) + B_f(\theta_k)y(k) \\ z_f(k) = C_f(\theta_k)x_f(k) + D_f(\theta_k)y(k) \end{cases} \quad (16)$$

No sistema acima, $x_f(k) \in \mathbb{R}^n$, $x_f(0) = 0$ e as matrizes A_f , B_f , C_f e D_f possuem dimensões compatíveis. O objetivo é determinar estas matrizes de tal forma que se minimize a norma \mathcal{H}_∞ do erro de estimação. Quando o filtro \mathbb{L} acima descrito é ligado ao sistema apresentado pela Equação 13, o erro de estimação produzido é tido por:

$$\mathbb{E}: \begin{cases} \tilde{x}(k+1) = \tilde{A}(\theta_k)\tilde{x}(k) + \tilde{J}(\theta_k)w(k) \\ e(k) = \tilde{C}(\theta_k)\tilde{x}(k) + \tilde{E}(\theta_k)w(k) \end{cases} \quad (17)$$

Finalmente, a equação que precisa ser resolvida pode ser escrita como:

$$\min_{A_f, B_f, C_f, D_f} \|\mathbb{E}\|_\infty^2 \quad (18)$$

Através de transformações de similaridade, este problema pode ser transformado em um problema de otimização convexo e resolvido de forma ótima através de desigualdades matriciais lineares, ou LMIs. Estas transformações podem ser vistas em [5] e o código MATLAB que implementa estas desigualdades e resolve o problema proposto se encontra em anexo a este relatório (Anexos 2, 3 e 4).

Capítulo 3

Procedimento Experimental

Aqui serão apresentadas, na sequência, as etapas que foram seguidas para a realização de um teste prático. Evocando novamente a essência do trabalho, deseja-se criar um método de estimação, fundindo dados de diferentes sensores, fornecendo uma informação de posição, algo semelhante a um GPS virtual (aqui denotado por GPS* na Figura 3.1).

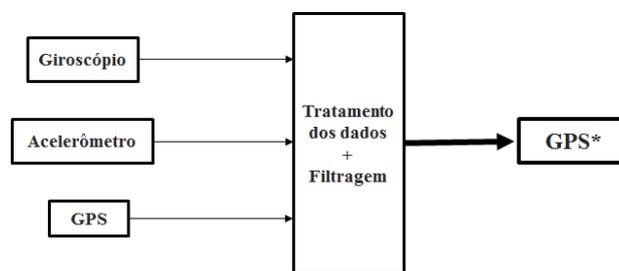


Figura 3.1. Diagrama simplificado do problema proposto. Fonte: Autoria própria

3.1 Preparativos Iniciais

3.1.1 Definição da Trajetória

Buscou-se aqui, então, definir uma trajetória que oferecesse uma barreira contra o GPS, fazendo com que houvesse ausência momentânea de sinal, inserindo a necessidade da utilização de outros sensores, que não apenas o GPS. Em termos estruturais, a presença de um túnel no percurso atenderia à necessidade acima exposta. Selecionou-se, assim, um trajeto na região do centro da Cidade de São Paulo, no bairro da Consolação, de 2,8 km, com um túnel, cujo trecho é destacado em vermelho na Figura 3.2 abaixo.



Figura 3.2. Trajetória selecionada para a execução do teste; em destaque, a presença de um túnel. Fonte: Google Maps em 31/10/15

3.1.2 Identificação de Software de Aquisição

Com o objetivo de simplificar a aquisição dos dados, conforme já foi exposto precedentemente, utilizou-se um aparelho de celular, pela facilidade de o mesmo já reunir os sensores em questão, a saber: acelerômetro, giroscópio e GPS. O desafio foi encontrar um aplicativo de celular (Software) que gravasse os dados dos sensores em uma tabela para um tratamento posterior, em um período de amostragem constante. Dos diversos programas gratuitos e disponíveis, como por exemplo, “*Sensor Track*” e “*Sensor Log*”, o programa que melhor se comportou durante os testes preliminares foi o “*AndroSensor*”. Abaixo se ilustra sua tela de funcionamento.



Figura 3.3. Tela do aplicativo de celular “AndroSensor”. Fonte: Autoria própria

3.2 Preparativos em Tempo de Execução

3.2.1 Posicionamento do Sensor

O posicionamento adequado dos sensores, conseqüentemente do celular, foi uma etapa importante que antecedeu o início dos testes. Neste quesito, o desafio foi encontrar o melhor lugar para fixá-lo, evitando que, durante a trajetória, houvesse movimento relativo dentro do veículo. Isso poderia prejudicar a qualidade das informações coletadas no experimento. Empiricamente, observou-se que locais mais baixos no veículo apresentavam menos vibração. Assim, optou-se por fixar o aparelho sobre o assento do passageiro com um elástico de borracha, fazendo com que o celular acompanhasse o veículo durante o movimento, sem escorregamentos.

Para fins de notação, o celular foi disposto no veículo de modo que a componente da aceleração $a_{lin,y}$ esteja sempre solidária ao eixo de movimento, conforme Figura 3.4.



Figura 3.4. Direção de aceleração $a_{lin,y}$ do celular coincidente com a direção de movimento do veículo. Fonte: Adaptado de [2]

3.2.2 Parâmetros do Software

Apesar de o aplicativo coletar dados de todos os sensores presentes no celular (luminosidade, ruído, bateria, etc.), o interesse alinhado com o tema era coletar apenas as informações consolidadas na Tabela 3.1 abaixo:

Tabela 3.1. Parâmetros coletados no teste. Fonte: Aplicativo “AndroSensor”

Variável	GPS (Lat.e Long.)	GPS (Precisão)	Vel Angular (x, y e z)	Acel. Linear (x, y e z)
Unidade	[°] (graus)	[m] (metros)	[°/s] (graus por segundo)	[m/s ²] (metros por segundo ao quadrado)

Na tabela acima, pode-se ver a informação sobre a precisão do GPS que também é fornecida pelo aplicativo. Ela se torna importante uma vez que o sistema em questão realiza algo análogo à média ponderada de forma inteiramente dinâmica entre os três sensores, para cada ponto. O período de amostragem selecionado foi de $T_s = 0,5s$, sendo este o menor disponível no aplicativo.

Na definição do percurso de testes buscou-se o extremo, ou seja, um ambiente no qual o GPS perderia o sinal completamente. Entretanto, mesmo em trechos sem túnel, a força do sinal do GPS sofre variações, podendo ser oscilações do receptor do celular ou até mesmo a presença de nuvens no céu. A variação da força do sinal é representada analiticamente pela precisão, em metros, destacada na Figura 3.5.

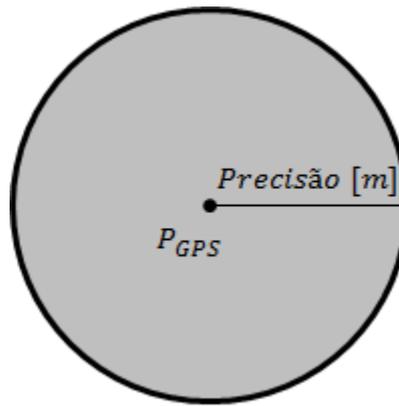


Figura 3.5. Representação gráfica da precisão do sinal do GPS. Fonte: Autoria própria

O círculo acima indica uma região de incerteza, ou seja, dependendo da intensidade do sinal do GPS, esta é a possível região que a coordenada fornecida pode se encontrar. Quanto menor for raio do círculo centrado no ponto P_{GPS} (portanto, quanto menor o valor numérico precisão) mais forte é esse sinal, constatando-se uma área mais restrita. Se o sinal estiver fraco, a área de possibilidades na vizinhança do ponto P_{GPS} será maior.

A ideia mencionada na introdução do Capítulo 3 sintetiza basicamente a criação de um GPS virtual, mais robusto e menos susceptível à falha de sinal do GPS ou até mesmo mau funcionamento momentâneo dos outros dois sensores, como pode ser visto, dependendo da sua precisão instantânea.

3.3 Tratamento dos Dados Pós-Execução

3.3.1 Aceleração Linear

Para a tratativa dos valores obtidos de aceleração linear tridimensional através do acelerômetro, é necessário evocar alguns conceitos de cinemática básica. Dessa forma, o dado de posição é obtido a partir dos dados coletados de aceleração sob os quais o celular estava submetido dentro do veículo ao curso da trajetória.

Assume-se aqui, portanto, que o celular é solidário ao veículo, ou seja, a aceleração linear do aparelho equivale à aceleração linear $a_{lin_{x,y,z}}$ (aceleração linear nas direções x , y e z) do próprio veículo, e que não há escorregamento do veículo em relação à pista e do celular em relação ao veículo. Ademais, conforme Figura 3.6, a componente a_{lin_y} acompanha o sentido do movimento. Na Figura 3.6 abaixo se esboça um esquema no qual é possível ver a evolução ao longo do percurso, exemplificando com três instantes

diferentes, a decomposição dos vetores de aceleração para uma trajetória fictícia, análogo ao que foi elaborado na prática.

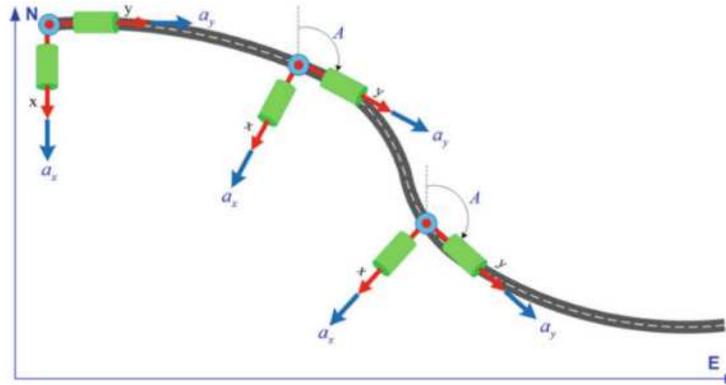


Figura 3.6. Decomposição dos vetores de aceleração ao longo de uma trajetória.

Fonte: [2]

É importante ressaltar que a aceleração linear medida desconta os efeitos da gravidade na direção do eixo z . Dessa maneira, o valor aferido de a_{lin_z} representa a variação da aceleração sofrida pelo veículo com relação à aceleração da gravidade (g):

$$a_{lin_z} = a_{real_z} - g \quad (19)$$

A Equação 20 mostra a relação cinemática de forma genérica para se obter, a cada instante, $s_{lin_{x,y,z}} = (x, y, z)$, adotando-se este como o referencial em PTL.

$$s_{lin_{x,y,z}} = \int v_{lin_{x,y,z}} dt = \int \left(\int a_{lin_{x,y,z}} dt \right) dt \quad (20)$$

3.3.2 Velocidade Angular

Analogamente, segue-se aqui o mesmo conceito previamente discutido no item 3.3.1 para a velocidade angular $\omega_{x,y,z}$. Da cinemática, a posição angular pode ser calculada conforme a Equação 21:

$$\varphi_{x,y,z} = \int \omega_{x,y,z} dt \quad (21)$$

A cada instante, a equação acima fornecerá a posição angular, isto é, a direção do movimento que o veículo possui ponto a ponto.

3.3.3 Coordenada GPS

Para que fosse possível cruzar os dados do acelerômetro e giroscópio (obtidos no PTL) com as informações provenientes do GPS, tornou-se necessário o tratamento dos dados deste último. Como já explicitado na Tabela 3.1, o aplicativo coleta a informação de posição no formato exatamente igual, e não por acaso, à Equação 4 ($P_{Dec-Deg} = (\lambda, \phi)$). A matemática de conversão de dados no formato (latitude, longitude) ao PTL também foi explicitada pelas Equações 8 a 11 no Capítulo 2.

Em suma, obtém-se para cada ponto aferido, sua posição equivalente no PTL no formato $P_{PTL} = (x_{PTL}, y_{PTL}, z_{PTL})$.

3.4 Definição do Sistema para Aplicação do Filtro

3.4.1 Espaço de Estados

Considere o modelo de um sistema contínuo (equação de estado \mathbb{F}) descrito pela Equação 1. Para este problema em questão, definiu-se o vetor de estados x e sua derivada \dot{x} como é mostrado a seguir:

$$x = \begin{bmatrix} Pos N \\ Pos E \\ Pos D \\ Vel N \\ Vel E \\ Vel D \\ 0 \\ 0 \\ 0 \end{bmatrix}; \dot{x} = \begin{bmatrix} Vel N \\ Vel E \\ Vel D \\ Acc N \\ Acc E \\ Acc D \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (22)$$

Onde $PosN, E, D$ ⁶ representam as posições cartesianas x, y, z . De modo semelhante, Vel indica a velocidade nas respectivas direções, assim como o termo Acc representa a aceleração no vetor \dot{x} .

Faz-se uma ressalva neste caso, pois na discretização deste sistema, adotar-se-á o vetor x de estados possuindo um bloco de armazenamento da informação do GPS, fazendo o ajuste da seguinte forma:

⁶ A notação N, E e D vem do inglês: *North, East* e *Down*, cuja tradução é Norte, Leste e Baixo, respectivamente;

$$x = \begin{bmatrix} Pos N \\ Pos E \\ Pos D \\ Vel N \\ Vel E \\ Vel D \\ GPS N \\ GPS E \\ GPS D \end{bmatrix} \quad (23)$$

Isto se faz necessário, pois a frequência de atualização do GPS é diferente do acelerômetro. Dessa forma, esse ajuste no caso discreto serve para armazenar a informação anterior do GPS até que ele tenha seu valor atualizado pela medição.

O vetor w , sendo ele o agrupamento dos fatores externos ao sistema, foi definido como:

$$w = \begin{bmatrix} Acc N \\ Acc E \\ Acc D \\ \delta Pos N \\ \delta Pos E \\ \delta Pos D \\ \delta Acc N \\ \delta Acc E \\ \delta Acc D \end{bmatrix} \quad (24)$$

Neste, δPos indica o ruído oriundo da oscilação do sinal do GPS, nas três direções. Já δAcc indica o ruído presente nas medidas de aceleração. De posse desses vetores acima descritos, construir-se-ão as matrizes de estado A , J , C_y , C_z , E_y e E_z . Por questões de notação, $J_{n \times n}$ é a matriz Identidade de dimensões $n \times n$ e $Z_{n \times m}$ é uma matriz de zeros de dimensões $n \times m$, para $n, m \in \mathbb{N}$.

$$A = \begin{bmatrix} Z_{3 \times 3} & J_{3 \times 3} & Z_{3 \times 3} \\ Z_{3 \times 3} & Z_{3 \times 3} & Z_{3 \times 3} \\ Z_{3 \times 3} & Z_{3 \times 3} & Z_{3 \times 3} \end{bmatrix}; \quad J = \begin{bmatrix} Z_{3 \times 3} & Z_{3 \times 3} & Z_{3 \times 3} \\ J_{3 \times 3} & Z_{3 \times 3} & Z_{3 \times 3} \\ Z_{3 \times 3} & Z_{3 \times 3} & Z_{3 \times 3} \end{bmatrix} \quad (25)$$

$$C_y = \begin{bmatrix} J_{3 \times 3} & Z_{3 \times 3} & Z_{3 \times 3} \\ Z_{3 \times 3} & Z_{3 \times 3} & Z_{3 \times 3} \end{bmatrix}; \quad C_z = [J_{3 \times 3} \quad Z_{3 \times 3} \quad Z_{3 \times 3}] \quad (26)$$

$$E_y = \begin{bmatrix} Z_{3 \times 3} & J_{3 \times 3} & Z_{3 \times 3} \\ J_{3 \times 3} & Z_{3 \times 3} & J_{3 \times 3} \end{bmatrix}; \quad E_z = [Z_{3 \times 3} \quad Z_{3 \times 3} \quad Z_{3 \times 3}] \quad (27)$$

Assim como feito com o vetor de estados x , é necessário discretizar o sistema para realizar as simulações. Porém, como se fará uso do sistema utilizando cadeia markoviana

de probabilidades, este passo será executado apenas ao final do projeto, utilizando o software de simulação Matlab.

3.4.2 Cadeia Markoviana

Para este tema em específico, definiram-se quatro modos de operação, assim como fora mencionado anteriormente. Segue na tabela abaixo o detalhamento dos modos escolhidos:

Tabela 3.2. *Detalhe dos modos de operação.* Fonte: Autoria própria

Modos de Operação	Descrição
1	Sinal do GPS forte; Atualiza-se a aferição do GPS;
2	Sinal do GPS médio; Atualiza-se a aferição do GPS;
3	Sinal do GPS fraco; Atualiza-se a aferição do GPS;
4	Período sem atualização do GPS - intervalo de medidas;

Tendo detalhado os modos com a Tabela 3.2 e explicado o conceito markoviano nos Capítulos precedentes, é possível agora apresentar a matriz de probabilidades P que fará parte do sistema durante a simulação.

As variáveis que aparecem na última linha serão determinadas levando em consideração o percurso. Por exemplo, p_1 pode ser lida da seguinte maneira: após um intervalo de medida, a probabilidade de estarmos em uma região detalhada no primeiro modo é p_1 e assim sucessivamente. Já o termo ip_4 é construído de tal sorte que a P respeite as regras de construção já apresentadas, sendo que $ip_4 = 1 - p_4$. O termo $p_4 = 98/99$, por sua vez, está relacionado com a diferença entre as frequências de atualização do GPS e acelerômetro, uma vez que diz respeito ao modo intermediário entre duas medições do GPS. Considerou-se a atualização deste cem vezes menor que do acelerômetro. Em $f_{acelerômetro} = 100f_{GPS}$. Os valores respectivos de cada modo serão determinados e explicados durante a simulação do sistema. Já

$$P = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ p_1 \cdot ip_4 & p_2 \cdot ip_4 & p_3 \cdot ip_4 & p_4 \end{bmatrix} \quad (28)$$

3.4.3 Inserção da Cadeia Markoviana no Espaço de Estados

O conceito de saltos markovianos aplicados ao sistema basicamente origina a aparição de quatro sistemas de estado em um só. Dentro das equações que criam o filtro

\mathcal{H}_∞ , os respectivos modos são selecionados de acordo com a matriz de probabilidades P representada acima. Ou seja, de modo prático, as matrizes de estado do sistema ganham um índice i cujo índice representará o respectivo modo (Equação 29).

$$\mathbb{M}: \begin{cases} \dot{x} = A_i x + J_i w \\ y = C_{y_i} x + E_{y_i} w \\ z = C_{z_i} x + E_{z_i} w \end{cases} \quad (29)$$

Com $i \in \mathbb{N} \mid i = \{1, 2, 3, 4\}$ devido aos quatro modos de operação. Segue abaixo o detalhamento das matrizes:

$$A_1 = A_2 = A_3 = A_4 = \begin{bmatrix} \mathcal{Z}_{3 \times 3} & J_{3 \times 3} & \mathcal{Z}_{3 \times 3} \\ \mathcal{Z}_{3 \times 3} & \mathcal{Z}_{3 \times 3} & \mathcal{Z}_{3 \times 3} \\ \mathcal{Z}_{3 \times 3} & \mathcal{Z}_{3 \times 3} & \mathcal{Z}_{3 \times 3} \end{bmatrix} \quad (30)$$

$$J_1 = J_2 = J_3 = J_4 = \begin{bmatrix} \mathcal{Z}_{3 \times 3} & \mathcal{Z}_{3 \times 3} & J_{3 \times 3} \\ J_{3 \times 3} & \mathcal{Z}_{3 \times 3} & J_{3 \times 3} \\ \mathcal{Z}_{3 \times 3} & \mathcal{Z}_{3 \times 3} & \mathcal{Z}_{3 \times 3} \end{bmatrix} \quad (31)$$

$$E_{y_1} = \begin{bmatrix} \mathcal{Z}_{3 \times 3} & e_1 * J_{3 \times 3} & \mathcal{Z}_{3 \times 3} \\ J_{3 \times 3} & \mathcal{Z}_{3 \times 3} & e_a * J_{3 \times 3} \end{bmatrix} \quad (32)$$

$$E_{y_2} = \begin{bmatrix} \mathcal{Z}_{3 \times 3} & e_2 * J_{3 \times 3} & \mathcal{Z}_{3 \times 3} \\ J_{3 \times 3} & \mathcal{Z}_{3 \times 3} & e_a * J_{3 \times 3} \end{bmatrix} \quad (33)$$

$$E_{y_3} = \begin{bmatrix} \mathcal{Z}_{3 \times 3} & e_3 * J_{3 \times 3} & \mathcal{Z}_{3 \times 3} \\ J_{3 \times 3} & \mathcal{Z}_{3 \times 3} & e_a * J_{3 \times 3} \end{bmatrix} \quad (34)$$

$$E_{y_4} = \begin{bmatrix} \mathcal{Z}_{3 \times 3} & \mathcal{Z}_{3 \times 3} & \mathcal{Z}_{3 \times 3} \\ J_{3 \times 3} & \mathcal{Z}_{3 \times 3} & e_a * J_{3 \times 3} \end{bmatrix} \quad (35)$$

O fator multiplicador (parâmetro e_n), $n \in \{1, 2, 3\}$, corresponde à força do sinal do GPS que caracteriza o modo n . Já o fator e_a representa a precisão do acelerômetro.

$$E_{z_1} = E_{z_2} = E_{z_3} = E_{z_4} = [\mathcal{Z}_{3 \times 3} \quad \mathcal{Z}_{3 \times 3} \quad \mathcal{Z}_{3 \times 3}] \quad (36)$$

$$C_{y_1} = C_{y_2} = C_{y_3} = \begin{bmatrix} J_{3x3} & Z_{3x3} & Z_{3x3} \\ Z_{3x3} & Z_{3x3} & Z_{3x3} \end{bmatrix} \quad (37)$$

$$C_{y_4} = \begin{bmatrix} J_{3x3} & Z_{3x3} & Z_{3x3} \\ Z_{3x3} & Z_{3x3} & J_{3x3} \end{bmatrix} \quad (38)$$

$$C_{z_1} = C_{z_2} = C_{z_3} = C_{z_4} = [J_{3x3} \quad Z_{3x3} \quad Z_{3x3}] \quad (39)$$

Ao final de todo o equacionamento acima explicitado, necessita-se efetuar todos os cálculos para avaliar os resultados preliminares. Em seguida, aplicar-se-á o filtro \mathcal{H}_∞ para prever a posição deste veículo ao longo da rota com a fusão dos sensores em questão.

Capítulo 4

Resultados e Discussões

É mostrado neste Capítulo os resultados obtidos com a simulação real e, de posse dos mesmos, quais foram as escolhas feitas para o desenrolar deste trabalho.

4.1 Resultados dos Testes em Trajetória Real

4.1.1 Validação dos Dados: GPS e Transformações

Com o objetivo de validar os dados obtidos durante a primeira trajetória (bairro Consolação), com o auxílio do Google Maps, plotou-se em um mapa as coordenadas de GPS (ainda no formato $P = (\textit{latitude}, \textit{longitude}) = (\lambda, \phi)$) para comparar com o trajeto definido em estudo. Na Figura 4.1 é possível visualizar a comparação de ambas, validando os dados adquiridos:



Figura 4.1. Comparação da trajetória definida no plano de estudos com os dados obtidos. Fonte: Autoria própria

O trecho indicado com o círculo vermelho, no qual é possível constatar a ausência de pontos do GPS, corresponde exatamente ao túnel proposto ao percurso, comprovando a teoria inicial de que a fusão de diferentes sensores evitaria, até certo ponto, a dependência

exclusiva de um ou outro. Neste exemplo, o GPS fica completamente sem sinal e o motorista poderia se perder no percurso.

Em seguida, realizou-se a matemática das coordenadas locais explicitada no Capítulo 2, quarto tópic, colocando os pontos do GPS em um PTL para que fosse possível comparar com os outros sensores. Faz-se nova comparação através da Figura 4.2 abaixo e constata-se consistência nos cálculos feitos:

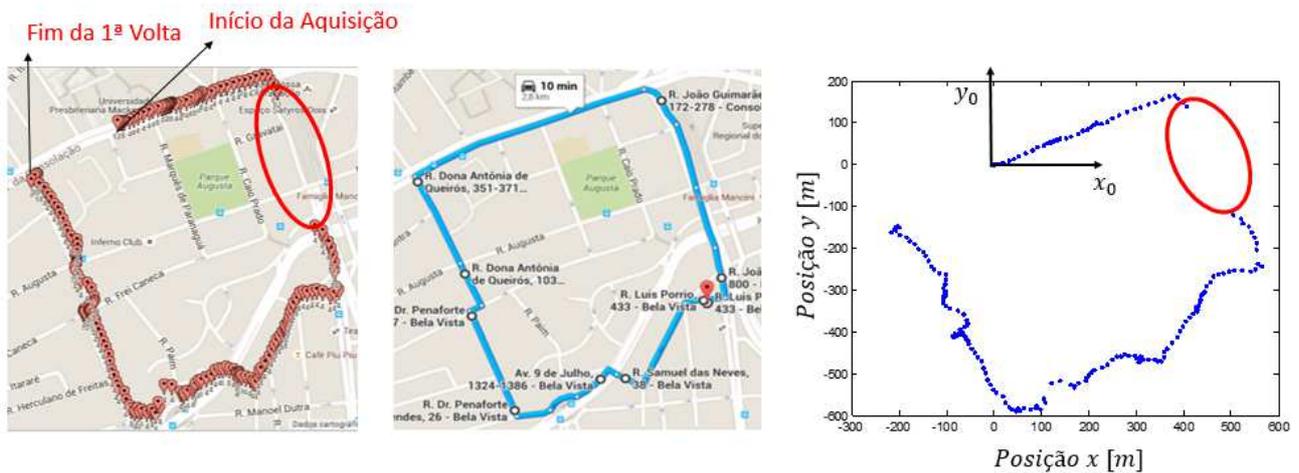


Figura 4.2. Comparação da trajetória definida no plano de estudos com os dados obtidos e com os dados calculados em PTL. Fonte: Autoria própria

4.1.2 Validação dos Dados: Acelerômetro e Giroscópio

Na validação das informações provenientes dos sensores acelerômetro e giroscópio, constatou-se que não correspondiam à realidade. Por exemplo, na tabela abaixo mostra-se os primeiros pontos para os quais calculou-se a posição angular:

Tabela 4.1. Parâmetros coletados no teste. Fonte: Aplicativo “AndroSensor”

Ponto 1	Ponto 2	Ponto 3	Ponto 4	Ponto 5	Ponto 6	Ponto 7	Ponto 8
-0,3037°	-0,4297°	0,1203°	-0,2464°	-0,2464°	0,3037°	-0,1833°	0,1833°

É fácil perceber que as posições angulares acima não são consistentes com o percurso realizado. Na terceira parte da Figura 4.2 (da direita para esquerda) na qual é ilustrada a origem (x_0 e y_0), o movimento se inicia com aproximadamente 30° de inclinação e em nenhum momento ocorrem ângulos negativos no percurso. Já para a aceleração, nos primeiros cálculos realizados obteve-se um erro próximo de uma centena de metros, inviabilizando por completo estes dados coletados, dificultando desnecessariamente a validação da qualidade do filtro.

Este erro pode ter diversas origens, valendo citar a alta sensibilidade ou baixa precisão dos sensores do celular, más condições da via culminando em uma vibração excessiva no interior do veículo ou até mesmo erros do algoritmo iterativo de integração numérica realizadas no Matlab.

4.2 Simulação da Trajetória

Em razão do curto período de tempo que se dispõe para o encerramento deste Trabalho de Conclusão de Curso II, optou-se por não insistir na obtenção de dados confiáveis e simular o sistema com dados reais e práticos. Apesar deste contratempo, foi gerado um programa que criou uma trajetória fictícia, bem como os valores de aceleração ao longo da mesma. O objetivo é de validar em teoria a fusão dos sensores, mas agora com dados simulados. Aqui, não se recuperou os dados de velocidade angular, deixando de lado a simulação de um giroscópio. O formato deste percurso pode ser visto na Figura 4.3 abaixo.

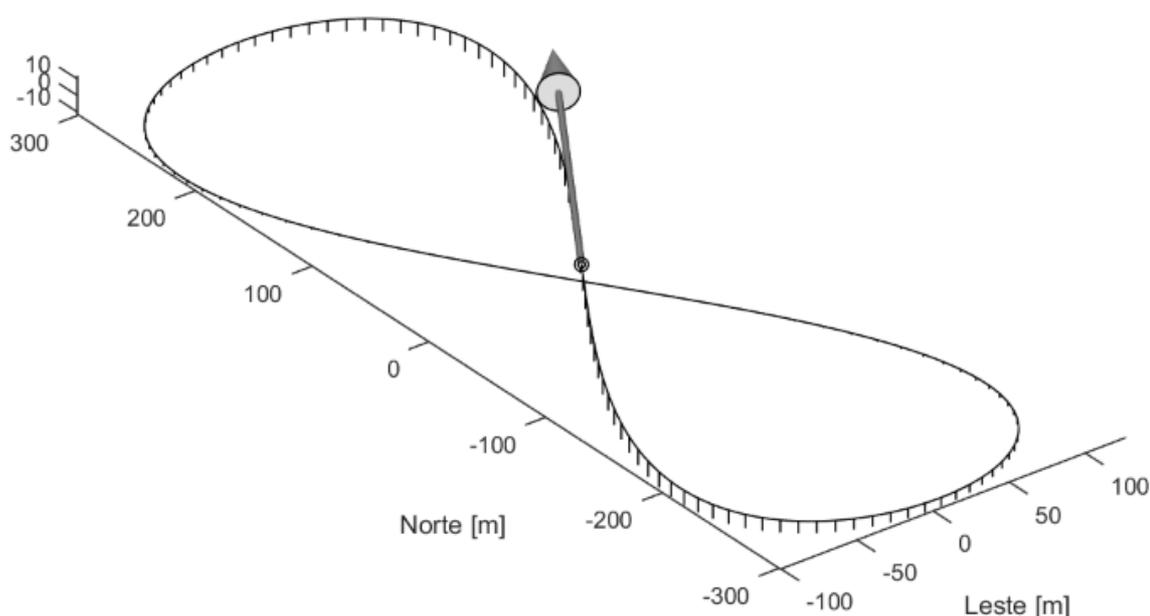


Figura 4.3. *Trajetoária gerada em simulação.* Fonte: Autoria própria

O traçado acima em formato de oito que foi simulado possui 1500 metros de extensão e desnível de 10 metros. Já a Figura 4.4 abaixo caracteriza as acelerações nas três direções (aqui chamadas de *N*, *E* e *D* apenas para simplificar a notação).

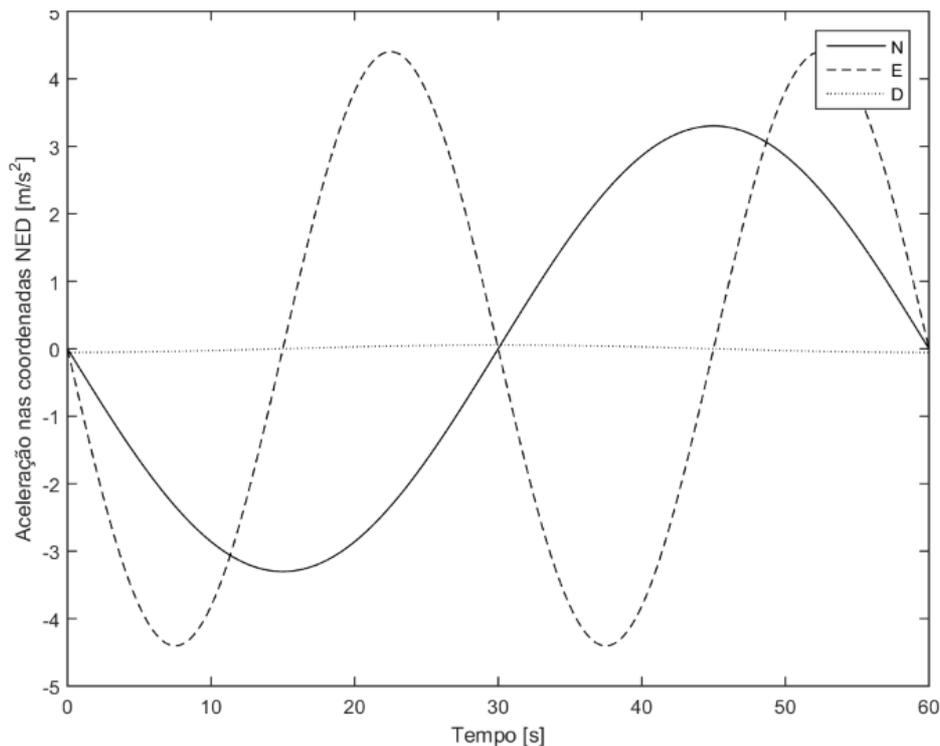


Figura 4.4. Acelerações para a trajetória gerada em simulação. Fonte: Autoria própria

Os gráficos que ilustram as posições assim como as velocidades nas três direções (N, E, D) seguirão em forma de Anexo (Anexo 1).

De posse de todos os dados acima, torna-se possível concentrar os esforços na aplicação do filtro (estimador de estados) propriamente dita.

4.3 Filtragem da Trajetória Simulada

É importante iniciar o tópico lembrando e quantificando alguns parâmetros mencionados em Capítulos anteriores. Com o intuito de definir os trechos de cada modo de operação da cadeia markoviana, foram determinadas as probabilidades (p_1 a p_3), conforme abaixo:

$$p_1 = 85,67\%; p_2 = 9,87\%; p_3 = 4,47\%; p_4 = 98,99\% \quad (40)$$

A razão de p_4 ser maior que as outras se dá por conta de a quarta faixa de operação aparecer em todos os trechos do circuito. É como se, ao longo do tempo, supondo 100 amostras dentro de 1 segundo, o modo p_4 aparecesse 99 vezes em contrapartida de qualquer um dos outros modos que apareceria apenas 1 vez.

Dessa forma, esses valores representam numericamente a razão de tempo em que o veículo se encontra no percurso em uma região com baixo sinal do GPS (semelhante ao túnel da trajetória real), exceto o quarto modo, conforme já explicado anteriormente. Na Figura 4.5 abaixo se destaca aproximadamente no traçado as regiões de operação.

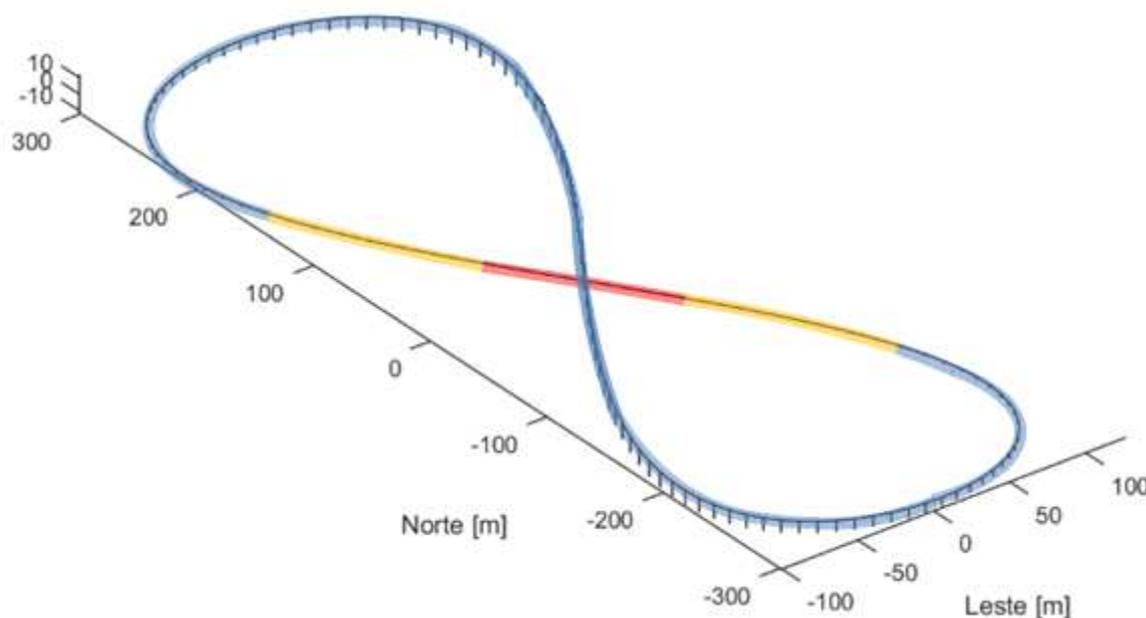


Figura 4.5. Destaque das faixas de operação no traçado simulado. Fonte: Autoria própria

Em azul, é representada a primeira região de operação, com bom sinal do satélite; em amarelo, representa-se a segunda região, com sinal mediano; por fim, a cor vermelha corresponde à terceira região, com pouco sinal, passando por baixo da elevação no trajeto (semelhante à um viaduto).

Em se tratando das matrizes do sistema, os índices e_1 , e_2 e e_3 que aparecem nas Equações 32, 33 e 34, respectivamente, nada mais são do que os graus de incerteza do GPS. Quantitativamente, determinou-se na simulação que uma boa precisão corresponde a uma vizinhança dentro de um círculo de raio 7 metros; a precisão média, ou seja, força média de sinal corresponde a um raio de 15 metros; finalmente, uma má precisão equivale a um círculo teórico de 30 metros. Segue abaixo na Equação 41 os valores de maneira resumida.

$$e_1 = 7; e_2 = 15; e_3 = 30; \quad (41)$$

O índice e_a , por sua vez, representando a incerteza da aceleração, possuiu seu valor determinado com base na literatura dos acelerômetros, isto é, relacionado à imprecisão do aparelho. Dessa forma, o valor escolhido pode ser visto na Equação 42 abaixo.

$$e_a = 0,003 \quad (42)$$

Nos códigos que seguem como Anexo (Anexos 2 e 3) pode-se notar a presença de um ruído branco gaussiano. Justifica-se essa aparição uma vez que se deseja simular uma condição real de funcionamento. Embora existam maneiras práticas de minimizá-los, ruídos de medições são inerentes a quase todos os processos físicos existentes, interferindo na determinação nas matrizes do filtro em questão. Denota-se por Q , portanto, este ruído unidimensional de medição. Em razão deste fato, o vetor de fatores externos acima denominado w é reescrito com a seguinte análise física:

$$w = \begin{bmatrix} Acc\ N \\ Acc\ E \\ Acc\ D \\ \delta Pos\ N \\ \delta Pos\ E \\ \delta Pos\ D \\ \delta Acc\ N \\ \delta Acc\ E \\ \delta Acc\ D \end{bmatrix} = \begin{bmatrix} \text{Dados do Simulador} \\ \text{Dados do Simulador} \\ \text{Dados do Simulador} \\ Q\ (\text{ruído}) \end{bmatrix} \quad (43)$$

No que diz respeito ao filtro \mathcal{H}_∞ propriamente dito, o código implementado em Matlab (Anexo 4), após executado em conjunto com o sistema discreto representado pela Equação 13, gera as matrizes de estado do filtro A_f , B_f , C_f e D_f , em acordo com o sistema da Equação 16, levando em conta os parâmetros citados acima, as estatísticas do ruído de medida, bem como a matriz P de saltos markovianos. O que se deseja obter é o vetor de posições estimadas z_f , em todas as direções.

Exemplifica-se aqui a maneira com que o sistema \mathbb{L} constrói o vetor de posições estimadas: Considerando o modo três (sinal fraco do GPS), os elementos das matrizes C_f e D_f que formam o primeiro elemento do vetor z_f , são mostrados a seguir.

$$z_f(1, k) = \begin{bmatrix} Pos N^*(1, k) \\ \vdots \end{bmatrix} = \begin{bmatrix} 0,7468 & 0 & 0 & Z_{1 \times 6} \\ 0 & 0,7468 & 0 & Z_{1 \times 6} \\ 0 & 0 & 0,7468 & Z_{1 \times 6} \end{bmatrix} \cdot \begin{bmatrix} x_f(1, k) \\ x_f(2, k) \\ \vdots \end{bmatrix} + \begin{bmatrix} 0,2532 & 0 & 0 & Z_{1 \times 6} \\ 0 & 0,2532 & 0 & Z_{1 \times 6} \\ 0 & 0 & 0,2532 & Z_{1 \times 6} \end{bmatrix} \cdot \begin{bmatrix} y(1, k) \\ y(2, k) \\ \vdots \end{bmatrix} \quad (44)$$

Acima, $Pos N^*(k)$ representa o ponto k do vetor de posição $Pos N$, cujo significado físico é a predição da posição na direção N (GPS virtual). O mesmo raciocínio se dará para as demais componentes. Os itens $x_f(i)$ e $y(i)$ correspondem a i – ésima posição dos vetores x_f e y , respectivamente.

$$z_f(1) = 0,7468 \cdot x_f(1) + 0,2532 \cdot y(1) \quad (45)$$

A matemática simples desta composição (Equação 45 acima) possui por trás um conceito importante sobre o mecanismo de filtragem. Isso significa na realidade que, para o terceiro modo (tido como exemplo), a posição estimada levará em conta 74,68% da estimativa do estado anterior e apenas 25,32% do dado medido. Na prática, esta diferenciação faz todo sentido, pois neste modo, definiu-se através da matriz de probabilidades e das matrizes de estado que o sinal do GPS seria fraco, portanto, uma informação pouco precisa. Sendo assim, o estimador de estados pondera esta composição, dando mais peso à variável mais confiável, que neste caso é a informação sobre o estado anterior. Pontos específicos em outros modos de operação seguirão a mesma lógica, isto é, em regiões com maior certeza sobre o valor medido, este dado possuirá um peso maior durante a estimação.

Uma vez realizada a simulação completa do problema, foi possível gerar o vetor z_f predizendo as posições, contemplando as informações dos sensores GPS e acelerômetro. Nas Figuras 4.6 e 4.7 abaixo é possível ver, para uma simulação, como o valor estimado (linha vermelha) se aproxima do valor real (linha azul - nas direções N e E), comprovando a eficácia do filtro e do método implementado. Os valores em D (equivalente ao eixo z) não são abordados, pois se focou em localizar primeiramente o veículo no plano $2D$, uma vez que desvios em altitude não possuem grande influência na definição da trajetória.

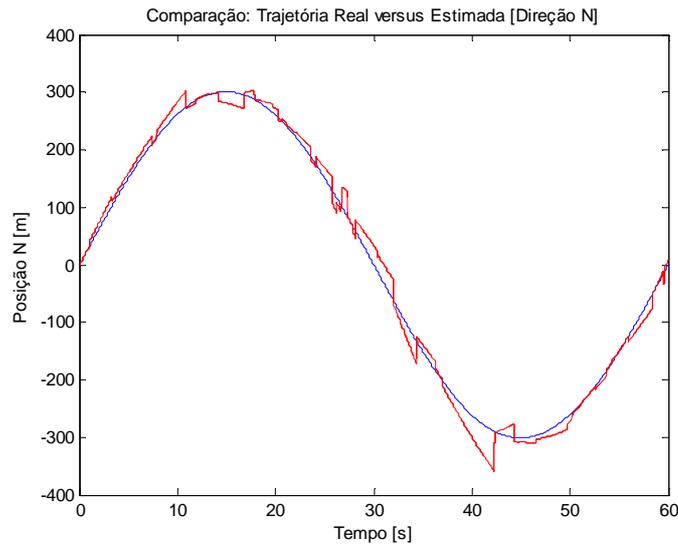


Figura 4.6. *Compara o entre a trajet ria gerada e a estimada na direc o N.* Fonte:

Autoria pr pria

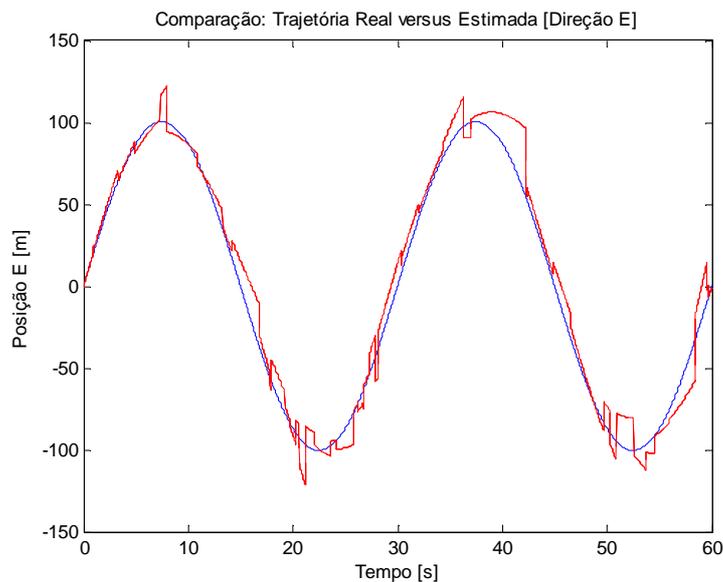


Figura 4.7. *Compara o entre a trajet ria gerada e a estimada na direc o E.* Fonte: Autoria

pr pria

Apesar da boa estima o do resultado desejado, nota-se ainda ru do nas curvas estimadas acima. Isso ocorre principalmente por conta do ru do inserido no sistema. Este valor depende de cada execu o, o que pode ser pouco significativo para indicar a qualidade do filtro proposto. Assim sendo, uma forma mais significativa para verificar a qualidade do filtro projetado indica em realizar uma simula o de monte carlo, ou seja, simulando uma s rie de vezes (aproximadamente 2.000 vezes neste caso) e extraindo o valor m dio dos vetores de posi o estimados. Em pontilhado,   poss vel ver, nos limites inferior e superior,

a média acrescida ou descontada de um desvio padrão, respectivamente. Nota-se, portanto, que as curvas quase se sobrepõem, indicando que a predição foi feita com sucesso.

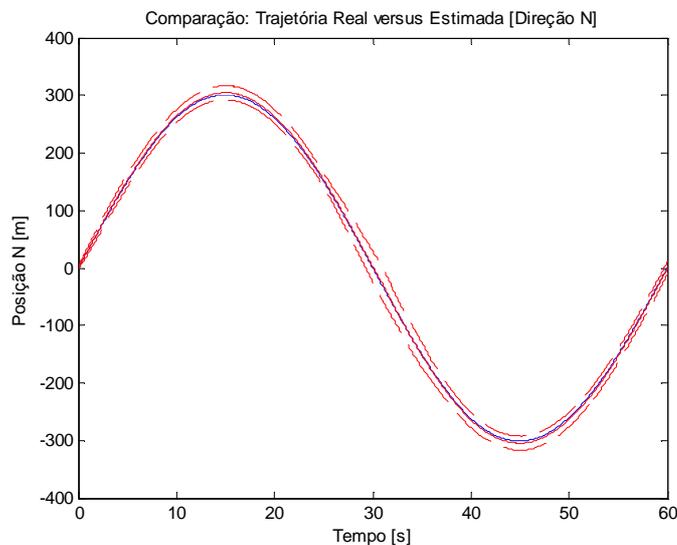


Figura 4.8. Comparação entre a trajetória gerada e a média estimada na direção N.

Fonte: Autoria própria

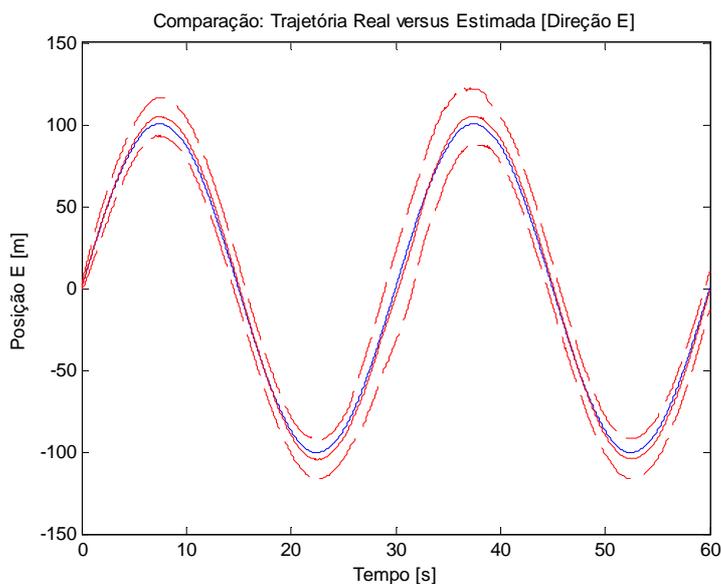


Figura 4.9. Comparação entre a trajetória gerada e a média estimada na direção E.

Fonte: Autoria própria

Outro ponto que merece ser destacado é que o erro de posição, na maioria dos pontos, é menor que aproximadamente 15 metros. Quando comparado ao comprimento do percurso e aos momentos em que não se dispunha de um GPS informando a localização, além do fato de que a própria precisão do GPS chega em 30 metros no terceiro modo de operação, trata-se de um erro mínimo, atestando a robustez do método.

Capítulo 5

Conclusões

Durante o percurso teórico e prático que levou à produção deste Trabalho de Conclusão de Curso II, algumas dificuldades foram encontradas em diversas etapas. Foi possível concluir que os sensores (acelerômetro e giroscópio) internos de um aparelho celular (*smartphone*) utilizados possuíam reduzida precisão, apesar de o sinal do GPS se comportar como o esperado, representando fidedignamente a trajetória seguida. Além disso, a transformação matemática de referenciais também se comportou conforme o esperado, representado em um plano (x, y) a trajetória escolhida.

Além do sinal extremamente ruidoso e pouco preciso, o emprego destes sensores ainda carrega mais erros devido à integração numérica inerente ao processo. Os dados do giroscópio, para se tornarem medidas de posição, necessitam ser integrados computacionalmente uma vez, processo este que acaba amplificando suas incertezas, produzindo valores pouco confiáveis. Já no caso do acelerômetro, se faz necessária duas integrações seguidas, amplificando ainda mais os erros de medição. Mesmo repetindo o trajeto por diversas vezes, até mesmo trocando de trajeto, não foi possível realizar uma coleta de dados confiável.

Com relação aos resultados obtidos após tornar este um sistema sujeito a saltos de probabilidade markovianos amplamente utilizado na literatura e aplicar o método de filtragem nos dados teóricos, é possível concluir que esta se mostrou extremamente robusta. Permitiu-se prever com precisão a posição espacial em uma trajetória específica. Mesmo com a presença de incertezas inerentes aos sensores (inseridas na simulação) além da inserção de um ruído de medição, o valor estimado se aproximou do valor real, como pôde ser visto nas Figuras 4.6 a 4.9. Ou seja, mesmo na região de baixíssimo sinal de satélite que se localiza entre os 22 e 32 segundos de simulação, a curva estimada acompanhou com precisão a trajetória real. Conclui-se, portanto, que a utilização de outros sensores e a aplicação do método produziram uma medida de posição necessariamente mais precisa que apenas a utilização do GPS em si, criando uma espécie de GPS virtual.

Dessa forma, nos aparelhos celulares, a fusão de sensores sozinha ainda não é suficiente para uma estimativa precisa do posicionamento de um veículo em condições de

dificuldade de obtenção do sinal GPS. Esta realização de certa forma justifica a implementação de outros algoritmos por aplicativos do tipo Google Maps, que se valem dos mapas locais contendo dados sobre o posicionamento preciso das vias, de forma a auxiliar a correção dos dados brutos obtidos pelos sensores.

Algumas sugestões que podem ser dadas para a continuidade do trabalho, como por exemplo, realizar a mesma simulação teórica utilizando valores simulados de giroscópio ou até mesmo insistir na obtenção de dados práticos para uma trajetória real em um ambiente mais controlado e/ou com equipamentos mais robustos. Demonstrando-se na prática o sucesso esperado para o método de fusão de sensores, sugere-se ainda que os códigos sejam otimizados, possibilitando embarcá-los em um celular, por exemplo, através da criação de um aplicativo. Assim, eventualmente, poderá se testar na prática com diferentes tipos de celular, cujos fabricantes melhoram sua capacidade de processamento e qualidade de seus componentes de maneira contínua, a criação de uma ferramenta poderosa e menos suscetível às intempéries e oscilações de componentes eletrônicos, como a criação do GPS virtual.

Bibliografia

- [1] O. L. V. Costa, M. D. Fragoso e R. Marques, *Discrete-Time Markov Jump Linear*, Londres: Springer-Verlag, 2000.
- [2] A. Noureldin, T. B. Kamarat e J. Georgy, *Fundamentals of Inertial Navigation, Satellite-based Positioning and their Integration*, Berlim: Springer-Verlag, 2013.
- [3] Portland State Aerospace Society, *Conversion of Geodetic coordinates to the Local Tangent Plane*, Portland, 2007.
- [4] P. D. Groves, *Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems*, London: Artech House, 2008.
- [5] A. R. Fioravanti, *Filtragem de sistemas discretos com parâmetros sujeitos a saltos markovianos*, Campinas, 2007.
- [6] A. P. C. Gonçalves, A. R. Fioravanti e J. C. Geromel, “ H_∞ robust and networked control of discrete-time MJLS through LMIs,” *Elsevier*, vol. 349, pp. 2171-2181, 2012.
- [7] A. P. C. Gonçalves, A. R. Fioravanti e J. C. Geromel, “Markov jump linear systems and filtering through network transmitted measurements,” *Elsevier*, vol. 90, nº Signal Processing, pp. 2842-2850, 2010.
- [8] A. P. C. Gonçalves, A. R. Fioravanti e J. C. Geromel, “Filtragem de Medidas Transmitidas por Rede Usando Sistemas com Saltos Markovianos a Tempo Discreto,” em *CBA - Congresso Brasileiro de Automática*, 2010.
- [9] A. P. C. Gonçalves, A. R. Fioravanti e J. C. Geromel, “ H_∞ Filtering of Discrete-Time Markov Jump Linear Systems Through Linear Matrix Inequalities,” *IEEE TRANSACTIONS ON AUTOMATIC CONTROL*, vol. 54, pp. 1347-1351, 2009.

Anexos

Anexo 1: Gráficos de posição e velocidade da Trajetória Simulada

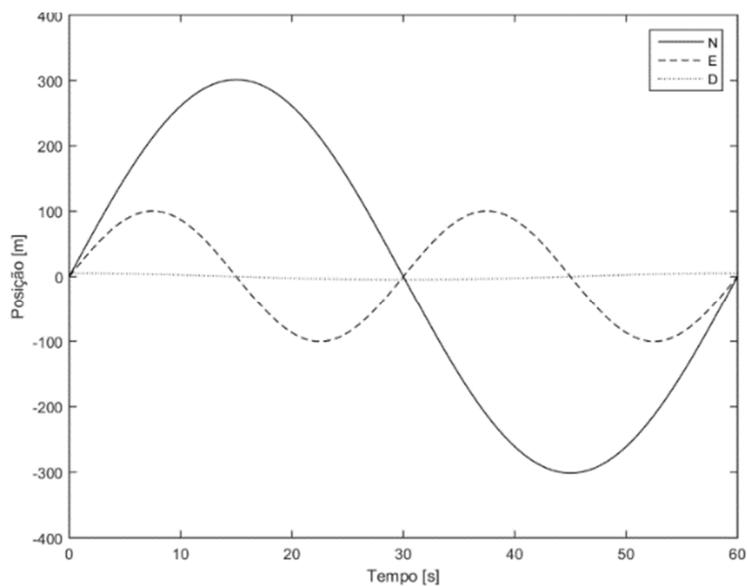


Figura A.1. Posição, nas direções D, E e N, da trajetória simulada. Fonte: Autoria própria

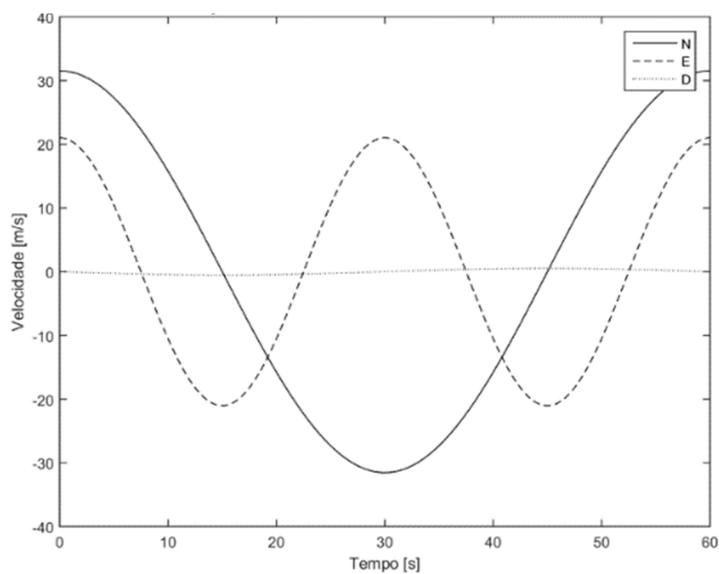


Figura A.2. Velocidade, nas direções D, E e N, da trajetória simulada. Fonte: Autoria própria

Anexo 2: Código MATLAB: Definição das matrizes de estado

```
clear all; close all; clc

I = eye(3);
Z = zeros(3);

%% Construção da matriz de probabilidades
prob = 98/99;
iprob = 1 - prob;
P = [0 0 0 1; 0 0 0 1; 0 0 0 1; (2570/3000)*iprob (296/3000)*iprob (134/3000)*iprob prob];

%% Parametros de incerteza
eld = 7; % Boa precisão do GPS (raio de 7 metros de incerteza)
e2d = 15; % Precisão mediana do GPS (raio de 15 metros de incerteza)
e3d = 30; % Má precisão do GPS (raio de 30 metros de incerteza)
ea = 0.003; % Incerteza do acelerômetro

%% Construção das matrizes de Estado
% x = [xN xE xD vN vE vD xaM xaE xaD]'
% w = [aN aE aD dxN dxE dxD daN daE daD]'

Ac = zeros(9,9,4); Ad = Ac;
Ac(:,:,1) = [Z I Z; Z Z Z; Z Z Z];
Ac(:,:,2) = [Z I Z; Z Z Z; Z Z Z];
Ac(:,:,3) = [Z I Z; Z Z Z; Z Z Z];
Ac(:,:,4) = [Z I Z; Z Z Z; Z Z Z];

Czc = zeros(3,9,4); Czd = Czc;
Czc(:,:,1) = [I Z Z];
Czc(:,:,2) = [I Z Z];
Czc(:,:,3) = [I Z Z];
Czc(:,:,4) = [I Z Z];

Cyc = zeros(6,9,4); Cyd = Cyc;
Cyc(:,:,1) = [I Z Z; Z Z Z];
Cyc(:,:,2) = [I Z Z; Z Z Z];
Cyc(:,:,3) = [I Z Z; Z Z Z];
Cyc(:,:,4) = [Z Z I; Z Z Z];

Jc = zeros(9,9,4); Jd = Jc;
Jc(:,:,1) = [Z Z Z; I Z Z; Z Z Z];
Jc(:,:,2) = [Z Z Z; I Z Z; Z Z Z];
Jc(:,:,3) = [Z Z Z; I Z Z; Z Z Z];
Jc(:,:,4) = [Z Z Z; I Z Z; Z Z Z];

Ezc = zeros(3,9,4); Ezd = Ezc;
Ezc(:,:,1) = [Z Z Z];
Ezc(:,:,2) = [Z Z Z];
Ezc(:,:,3) = [Z Z Z];
Ezc(:,:,4) = [Z Z Z];

Eyc = zeros(6,9,4); Eyd = Eyc;
Eyc(:,:,1) = [Z eld*I Z; I Z ea*I];
Eyc(:,:,2) = [Z e2d*I Z; I Z ea*I];
Eyc(:,:,3) = [Z e3d*I Z; I Z ea*I];
Eyc(:,:,4) = [Z Z Z; I Z ea*I];

%% Período de amostragem;
Ts = 1e-2;

%% Construção do o espaço de estados markoviano
for i = 1:4
    Sc = ss(Ac(:,:,i),Jc(:,:,i),[Cyc(:,:,i);Czc(:,:,i)],[Eyc(:,:,i);Ezc(:,:,i)]);
    Sd = c2d(Sc,Ts);
    Ad(:,:,i) = Sd.A;
    Ad(7:end,:,i) = zeros(3,9);
    Jd(:,:,i) = Sd.B;
    Cyd(:,:,i) = Sd.C(1:6,:);
    Czd(:,:,i) = Sd.C(7:end,:);
    Eyd(:,:,i) = Sd.D(1:6,:);
    Ezd(:,:,i) = Sd.D(7:end,:);
end

%% Ajuste para a inserção da repetição dos valores medidos do GPS, em tempo discreto
Ad(7:end,:,1) = [I Z Z];
```

```

Jd(7:end,:,1) = [Z e1d*I Z];
Ad(7:end,:,2) = [I Z Z];
Jd(7:end,:,2) = [Z e2d*I Z];

Ad(7:end,:,3) = [I Z Z];
Jd(7:end,:,3) = [Z e3d*I Z];

Ad(7:end,:,4) = [Z Z I];
Jd(7:end,:,4) = [Z Z Z];

```

Anexo 3: Código MATLAB: Algoritmo do filtro \mathcal{H}_∞ para sistemas markovianos

```
function [filtro, norma, var] = markov_hinf_dfilter(P,A,J,Cz,Ez,Cy,Ey)
```

```

n = size(A,1); % Numero de estados
N = size(P,1); % Numero de modos
r = size(Cy,1);
p = size(Cz,1);

setlmis([]);

gamma = lmivar(1,[1 0]);

X = zeros(1,N);
F = zeros(1,N);
K = zeros(1,N);

for i = 1:N
    X(i) = lmivar(1,[n 1]);
    F(i) = lmivar(2,[n r]);
    K(i) = lmivar(2,[p r]);
end

ct = 1;
for i = 1:N
    lmiterm([-ct 1 1 X(i)],1,1);
    lmiterm([-ct 2 2 gamma],1,1);
    lmiterm([-ct 3 1 F(i)],1,Cy(:, :, i));
    lmiterm([-ct 3 2 F(i)],1,Ey(:, :, i));
    lmiterm([-ct 4 1 0],Cz(:, :, i));
    lmiterm([-ct 4 1 K(i)],-1,Cy(:, :, i));
    lmiterm([-ct 4 2 0],Ez(:, :, i));
    lmiterm([-ct 4 2 K(i)],-1,Ey(:, :, i));
    lmiterm([-ct 4 4 0],1);
    for j = 1:N
        lmiterm([-ct 3 1 X(j)],P(i,j),A(:, :, i));
        lmiterm([-ct 3 2 X(j)],P(i,j),J(:, :, i));
        lmiterm([-ct 3 3 X(j)],P(i,j),1);
    end
    ct = ct + 1;
end

lmisys = getlmis;
options = [1e-7,2000,0,200,1];

c = zeros(decnbr(lmisys),1);
c( decinfo(lmisys,gamma) ) = 1;

[copt,xopt] = mincx(lmisys,c,options);

if ~isempty(xopt)

    var.X = zeros(n,n,N);
    var.F = zeros(n,r,N);
    var.K = zeros(p,r,N);
    var.Xp = zeros(n,n,N);

    for i = 1:N
        var.X(:, :, i) = dec2mat(lmisys,xopt,X(i));
        var.F(:, :, i) = dec2mat(lmisys,xopt,F(i));
        var.K(:, :, i) = dec2mat(lmisys,xopt,K(i));
        for ii = 1:N
            var.Xp(:, :, ii) = var.X(:, :, ii) + P(ii,i)*var.X(:, :, i);

```

```

        end
    end

    var.gamma = dec2mat(lmisys,xopt,gamma);
    norma = sqrt(copt);
    filtro.A = zeros(n,n,N);
    filtro.B = zeros(n,r,N);
    filtro.C = zeros(p,n,N);
    filtro.D = zeros(p,r,N);

    for i = 1:N
        filtro.A(:,:,i) = A(:,:,i) + var.Xp(:,:,i)\var.F(:,:,i)*Cy(:,:,i);
        filtro.B(:,:,i) = -var.Xp(:,:,i)\var.F(:,:,i);
        filtro.C(:,:,i) = Cz(:,:,i) - var.K(:,:,i)*Cy(:,:,i);
        filtro.D(:,:,i) = var.K(:,:,i);
    end
else
    var = [];
    norma = inf;
    filtro = [];
end
end
end

```

Anexo 4: Código MATLAB: Rotina que explicita os modos de operação, fazendo chamada à simulação

```

clear all; close all; clc

%% Carrega-se o Espaço de Estados construído em "markov_gps"
markov_gps;

%% Chamada a função do filtro propriamente dita
[Filtro, norma, var] = markov_hinf_dfiler(P,Ad,Jd,Czd,Ezd,Cyd,Eyd);

%% Carrega-se os dados da trajetória
load('dados.mat');

%% Definição da quantidade de iterações para se obter um valor médio
Niter = 2000;
zf = zeros(3,6001,Niter);

%% Construção do vetor de posições estimadas
for iter = 1:Niter
    %% Criação do vetor de fatores externos, bem como a inserção do ruído de medidas
    w = [AccN; AccE; AccD; randn(6,6001)];

    x0 = [PosN(1) PosE(1) PosD(1) VelN(1) VelE(1) VelD(1) PosN(1) PosE(1) PosD(1)]';
    xf0 = x0;

    x = x0;
    xf = xf0;
    dado_gps = [PosN(1) PosE(1) PosD(1)]';

    for k = 2:6001
        dado = rand(1);
        if(dado < 98/99)
            m = 4;
        else
            if ( (k < 2570) || (k > 3432) )
                m = 1;
            elseif ( (k < 2866) || (k > 3136) )
                m = 2;
            else
                m = 3;
            end
        end
    end

    if( m < 4 )
        dado_gps = [PosN(k); PosE(k); PosD(k)] + Eyd(1:3,4:6,m)*w(4:6,k);
    end

    x = [PosN(k); PosE(k); PosD(k); VelN(k); VelE(k); VelD(k); dado_gps];
end

```

```

y = Cyd(:, :, m)*x + Eyd(:, :, m)*w(:, k);

zf(:, k, iter) = Filtro.C(:, :, m)*xf + Filtro.D(:, :, m)*y;
xf = Filtro.A(:, :, m)*xf + Filtro.B(:, :, m)*y;

end

end

%% Obtenção dos gráficos
zfm = mean(zf, 3);
zfs = std(zf, [], 3);
figure
plot(Time, PosN); hold on; plot(Time, zfm(1, :), 'red'); hold on;
title('Comparação: Trajetória Real versus Estimada [Direção N]'); hold on;
xlabel('Tempo [s]'); hold on;
ylabel('Posição N [m]');
hold on;
plot(Time, zfm(1, :) + zfs(1, :), 'r--');
hold on;
plot(Time, zfm(1, :) - zfs(1, :), 'r--');

figure
plot(Time, PosE); hold on; plot(Time, zfm(2, :), 'red'); hold on;
title('Comparação: Trajetória Real versus Estimada [Direção E]'); hold on;
xlabel('Tempo [s]'); hold on;
ylabel('Posição E [m]');
hold on;
plot(Time, zfm(2, :) + zfs(2, :), 'r--');
hold on;
plot(Time, zfm(2, :) - zfs(2, :), 'r--');

figure
plot(Time, PosD); hold on; plot(Time, zfm(3, :), 'red'); hold on;
title('Comparação: Trajetória Real versus Estimada [Direção D]'); hold on;
xlabel('Tempo [s]'); hold on;
ylabel('Posição D [m]');
hold on;
plot(Time, zfm(3, :) + zfs(3, :), 'r--');
hold on;
plot(Time, zfm(3, :) - zfs(3, :), 'r--');

```