MINISTÉRIO DA CIÊNCIA E TECNOLOGIA
INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS

INPE-14607-TDI/1187

ESTIMAÇÃO E CONTROLE APLICADOS A UM PROBLEMA DE RASTREAMENTO UTILIZANDO E COMPARANDO DOIS AMBIENTES INTEGRADOS DE MODELAGEM, IDENTIFICAÇÃO E SIMULAÇÃO

Maria Cecília Pereira de Faria

Dissertação de Mestrado do Curso de Pós-Graduação em Engenharia e Tecnologia Espaciais/Mecânica Espacial e Controle, orientada pelo Dr. Marcelo Lopes de Oliveira e Souza, aprovada em 20 de maio de 2005.

INPE
São José dos Campos
2007

681.5:629.78

Faria, M. C. P.

Estimação e controle aplicados a um problema de rastreamento utilizando e comparando dois ambientes integrados de modelagem, identificação e simulação / Maria Cecília Pereira de Faria. - São José dos Campos: INPE, 2005.

182p. ; – (INPE-14607-TDI/1187)

1. Controle ótimo. 2. Filtro de Kalman. 3. Estimação de estados 4. Simulação 5. Problema de rastreamento. I. Título.

Aprovado (a) pela Banca Examinadora em cumprimento ao requisito exigido para obtenção do Título de Mestre em

Engenharia e Tecnologia Espaciais/Mecânica Espacial e Controle

Dr. Roberto Vieira da Fonseca Lopes

Presidente / INPE / SJCampos - SP

Dr. Marcelo Lopes de Oliveira e Souza

Orientador(a) / INPE / SJCampos - SP

Dr. Gilberto da Cunha Trivelato

Membro da Banca / EMBRAER / SJCampos - SP

Dr. Waldemar de Castro Leite Filho

Membro da Banca / IAE/CTA / SJCampos - SP

Dr. Hassan Ahmad Sidaoui

Convidado(a) / MECTRON / SJCampos - SP

Aluno (a): Maria Cecília Pereira de Faria

São José dos Campos, 20 de maio de 2005

"Jamais considere seus estudos como uma obrigação, mas como uma oportunidade invejável (...) para aprender a conhecer a influência libertadora da beleza do reino do espírito, para seu próprio prazer pessoal e para proveito da comunidade à qual seu futuro trabalho pertencer."

Albert Einstein

À minha mãe,
CAROLINA

AGRADECIMENTOS

Ao Instituto Nacional de Pesquisas Espaciais (INPE), pela estrutura que pude utilizar e pelo curso em si, o que é a realização de um grande sonho.

Ao Dr. Marcelo Lopes de Oliveira e Souza pela orientação técnica e acadêmica, pelas oportunidades oferecidas, pelo seu apoio e incentivo.

Ao Eng. Pelson de Souza Pinto, pela co-orientação, pelo conhecimento transmitido, pelo apoio, incentivo, e pela sua paciência; e ao Dr. Hassan Ahmed Sidaoui, pela ajuda no aprendizado de MATLAB®.

Ao Dave Varvell, que, de longe, auxiliou no uso do MATRIXx®, com paciência e empenho. Sem sua ajuda a comparação de ambientes não poderia ser realizada.

Aos professores Dr. Roberto Vieira da Fonseca Lopes e Hélio Koiti Kuga, pela ajuda na parte de estimação e enorme paciência durante o andamento deste trabalho.

À Mectron EIC Ltda. pelo financiamento deste trabalho, pela oportunidade de convivência em uma empresa da área Aeroespacial, e aos colegas da Mectron que diretamente ou indiretamente apoiaram este trabalho.

Às secretárias da Divisão de Mecânica Espacial e Controle (DMC), que tornam trabalhos como este possíveis, em especial à Márcia Alvarenga, secretária da Pós Graduação da Engenharia e Tecnologia Espaciais/ Dinâmica Orbital e Controle (ETE/ CMC) durante o andamento deste trabalho.

Aos colegas de curso Gabriel, Carmen, Rolf, Renato, entre outros, que se tornaram meus grandes amigos durante o trabalho. Aos amigos que, de longe, deram força para o prosseguimento deste trabalho, em especial, Maria Francisca, que colaborou com os desenhos e Marcos, com traduções.

Aos colegas da Escola de Filosofia Humanista Excalibur, por terem me mostrado o sentido desta e outras realizações, em especial, ao Mestre Taboada e à profa Kátia.

À minha família, pelo apoio incondicional e braços sempre abertos: Mamãe, Laeticia, Tistu, Bola, Vovô e Vovó.

Finalmente, ao Leandro, pelo apoio, pela ajuda com a estimação e com o Latex, pela paciência e carinho, e também por ter colorido meus dias e os feito mais felizes.

RESUMO

O trabalho aqui apresentado teve como objetivo resolver um problema de rastreamento de trajetória, considerando um veículo aeroespacial rastreador que sofre influências não perfeitamente modeladas do ambiente, assim como medidas ruidosas dos seus sensores. Outro objetivo do trabalho foi utilizar dois ambientes similares de Modelagem, Identificação e Simulação, o MATLAB® e o MATRIXx® a fim de comparar suas funcionalidades. Assim, foram desenvolvidos três modelos de simulação, progressivamente mais realistas, que foram simulados nos dois ambientes citados. Um dos modelos operava em malha aberta, enquanto para dois deles foram projetadas leis de controle visando um rastreio de uma trajetória previamente definida. Em todos os modelos foram considerados sensores embarcados ruidosos. As medidas fornecidas por estes sensores foram utilizadas para se fazer uma estimativa não somente da posição e velocidade do veículo rastreador, como também da posição e velocidade de um segundo veículo que navega em mar aberto. Estes modelos foram simulados em dois ambientes integrados de Modelagem, Identificação e Simulação, assim como seus controladores e estimadores. Os controladores foram projetados utilizando a teoria de Controle Ótimo, enquanto o estimador utilizado foi o Filtro de *Kalman*. Foram simuladas as trajetórias desejadas, as trajetórias descritas pelo veículo controlado, sendo o controlador alimentado pelo estado exato do veículo e, finalmente, a trajetória descrita pelo veículo controlado, com o controlador alimentado pelas estimativas do estado real do veículo. Estas simulações ofereceram condições para se comparar os ambientes estudados sob o ponto de vista do usuário. Os resultados obtidos mostraram que o controlador projetado com a teoria de Controle Ótimo produziu um rastreamento eficiente, dentro das especificações determinadas, mesmo quando alimentado com estados estimados. As estimativas do estado do veículo rastreador foram sempre eficientes, embora, para casos mais realistas, as estimativas do estado do veículo rastreado não se mostrassem eficientes. O trabalho permitiu uma comparação entre os ambientes de Modelagem, Identificação e Simulação utilizados, sendo que um ou outro se faz mais adequado, dependendo da aplicação visada.

ESTIMATION AND CONTROL APPLIED TO A TRACKING PROBLEM USING AND COMPARING TWO INTEGRATED ENVIRONMENTS OF MODELLING, IDENTIFICATION AND SIMULATION

ABSTRACT

This paper's purpose is to solve a trajectory tracking problem. Here we considered a tracking aerospace vehicle which is under the influence of a non-perfectly modelled atmosphere. We also considered noisy measures taken by its sensors. Another purpose was to use two similar environments for Modelling, Identification and Simulation, MATLAB® and MATRIXx® to compare their functionalities. Thus, three simulation models were developed in both environments. These models were progressively more realistic. One of them used no feedback, while control laws were designed for the other ones, so that they would track a predefined trajectory. All models were supposed to have embedded noisy sensors. The measures taken by these sensors were used to estimate the position and velocity of the aerospace vehicle, but also position and velocity of a second vehicle, sailing at sea. These models, as well as their controllers and estimators, were simulated in two integrated Modelling, Identification and Simulation environments. The controllers were designed using the theory of Optimal Control. The estimator used was the *Kalman* Filter. We simulated the reference trajectory and the trajectory described by the vehicle, being the feedback of the controller both the exact state of the vehicle and its estimate. These simulations made it possible to compare the environments under the user's point of view. The obtained results show that the controller designed with Optimal Control Theory produced an effective tracking, considering the stipulated specifications. This tracking was effective even when the feedback of the controller was achieved by means of estimated states. The estimates of the state of the aerospace vehicle were always efficient, even though, for more realistic models, the estimates of the tracked vehicle were not efficient at all. This work provided ways to compare the environments of Modelling, Identification and Simulation used. Depending on the desired application, one can be more suitable than the other.

SUMÁRIO

LISTA DE FIGURAS

LISTA DE TABELAS

CAPÍTULO 1

INTRODUÇÃO

O problema de rastreamento de trajetórias (controle) é bastante tratado pela literatura do meio (Sobey e Suggs, 1963; Kirk, 1970; Garnell, 1980; Blakelock, 1991), embora o caso tratado aqui seja uma situação bastante específica e, ainda, aliada a um problema de estimação de estados.

Neste caso, tratamos um problema de rastreamento envolvendo um veículo aeroespacial seguindo uma trajetória pré-definida em direção a um segundo veículo que navega em mar aberto, enquanto estima seu estado (relacionado à sua posição e velocidade), e também a posição e velocidade do segundo. Analisamos aqui também a influência do efeito da estimação de estados na realimentação do controle. A abordagem do problema foi feita em etapas, considerando que a dinâmica do veículo rastreador envolve progressivos graus de realismo, embora a dinâmica do veículo rastreado tenha sido simulada sempre de acordo com o mesmo modelo.

Para se modelar um problema envolvendo veículos aeroespaciais, é necessário definir, inicialmente, uma representação do veículo ou veículos, além da interação entre eles. Os graus de realismo são acrescentados à medida que se nota que o modelo utilizado não corresponde à realidade com a fidelidade necessária, em um compromisso entre fidelidade e simplicidade.

A lei de controle a ser projetada deve levar o veículo aeroespacial para uma região próxima da trajetória desejada, dentro dos limites físicos impostos pelos atuadores. Ao projetar esta lei de controle deve-se considerar que ela deve ser robusta, para assegurar que o veículo será controlado mesmo sob condições iniciais desfavoráveis (como lançamento longe do ponto de operação, presença de rajadas de vento, entre outras); mas, ao mesmo tempo, deve-se considerar que o computador de bordo, responsável pelo seu processamento, tem capacidade de computação bastante limitada, como é usual neste tipo de veículos aeroespaciais.

Sabemos que na década de 80 os processadores usualmente embarcados eram processadores de ponto fixo, trabalhando com 8 bits, como por exemplo o 8031. Na década de 90, ainda usando processadores de ponto fixo de 8 bits, começou-se a usar

o 80196. Na atualidade, o que se faz é aliar DSP´s ao 80196, de forma que divide-se a atividade de processamento em várias tarefas, cada uma responsável por um tipo de tratamento de sinal, aliviando, assim, o processador principal. Um dos recursos que limita a operação destes componentes é o barramento, que é responsável pela troca de informações entre os processadores.

Assim, havendo hoje maior capacidade de processamento, pode-se pensar na possibilidade de projetar leis de controle que sejam menos simples, e que tragam melhor desempenho para o controlador do veículo. Apresenta-se neste trabalho, portanto, uma alternativa aos usuais de controladores PI, PD e PID normalmente ainda utilizadas no meio.

Pela própria natureza do problema, não se dispõem de dados físicos ou dinâmicos sobre o veículo rastreado, enquanto a modelagem do veículo aeroespacial também conta com imperfeições e simplificações, sendo também seu movimento influenciado por fenômenos não perfeitamente modelados. Sabemos, ainda, que as medidas disponíveis são feitas por sensores embarcados no veículo rasterador. As medidas de posição e velocidade são realizadas por uma plataforma de navegação inercial, que é responsável pelas medidas de posição e velocidade do centro de massa do veículo. Estes sensores estão sujeitos não apenas a erros de medida e ruídos intrínsecos aos dispositivos, mas também, no caso da plataforma inercial, a erros e arredondamentos devido às integrações que realiza. A medida do ângulo entre os dois veículos aeroespaciais, chamado ângulo da linha de visada, é feita por um autodiretor, que também conta com erros e ruídos em suas medidas. Há também um *bias* em suas medidas, cuja integração é crescente com o tempo.

Assim, uma lei de controle que fará com que o veículo rastreie a trajetória de referência baseando-se apenas nas medidas ou na propagação no tempo contará com imperfeições. A forma de se contornar este problema é filtrar de forma ponderada as duas, obtendo assim a melhor estimativa do estado desejado (Maybeck, 1979), e utilizar esta estimativa para a lei de controle e para a localização do veículo medido. Para esta filtragem será feito uso de teoria de estimação, através do algoritmo do filtro de *Kalman* (Maybeck, 1979; Stovall, 1997). Este filtro permite que, a partir de medidas realizadas e modelos matemáticos da evolução do sistema, uma estimativa ótima do seu estado seja realizada, desde que a representação do sistema tenha sido feita de forma linear.

Observe-se que foge do escopo deste trabalho a determinação da trajetória de referência, assim como a aproximação e encontro dos veículos (não se visa, neste caso, uma situação de *rendez-vous* entre os veículos).

Para se simular a evolução temporal do problema, assim como projetar e testar a lei de controle e o filtro de *Kalman* utilizados, emprega-se amplamente a ferramenta computacional. Esta etapa, que era feita até certo tempo atrás por programação procedural, é feita de forma facilitada atualmente pelo surgimento dos ambientes integrados de modelagem, identificação e simulação - MIS, cada vez mais utilizados em projetos de engenharia. Estes ambientes proporcionam a possibilidade de, em todas as etapas do projeto, otimizar, simular e testar vários subsistemas de forma amigável, intuitiva e confortável (Linkens, 1993; Davis, 1998; Ören, 2002; Steinhaus, 2002).

A escolha das ferramentas computacionais neste trabalho recai sobre os dois ambientes de MIS mais utilizados na área aeroespacial. Um deles, o MATLAB®/ *Simulink*, pertence à empresa *The MathWorks*; e o outro, o MATRIXx®/ *SystemBuild*, pertence à empresa *National Instruments*. Estes ambientes possuem várias das características exigidas para a modelagem de um problema físico, em particular, problemas que envolvem veículos aeroespaciais (Trivelato e Souza, 2001), embora estas características e funcionalidades não sejam igualmente distribuídas pelos ambientes.

Depois de feita a modelagem e a simulação do sistema em ambos os ambientes, uma comparação dos resultados oferecidos por ambos foi feita, sob o ponto de vista do usuário, inicialmente com critérios simples e gerais; e, posteriormente, mais complexos e específicos do problema estudado, estabelecidos durante o andamento do trabalho.

1.1 Objetivos

O objetivo deste trabalho é realizar o estudo de estimação e controle aplicados a um problema de rastreamento utilizando e comparando dois ambientes integrados de Modelagem, Identificação e Simulação - MIS. Nele será abordado o projeto da da lei de controle e do estimador de um veículo aeroespacial rastreando uma trajetória pré-definida enquanto estima a sua posição e velocidade, e as de um segundo veículo.

Para isto, alguns passos foram seguidos, a saber:

- Revisão da bibliografia disponível sobre o problema estudado e sobre os conceitos envolvidos;
- Formulação do problema de rastreamento como um problema de controle ótimo dentro das condições propostas, fazendo simplificações e considerações;
- Modelagem do veículo aeroespacial rastreador através de equações dinâmicas, aumentando os graus de realismo à medida que se fez necessário;
- Modelagem do veículo medido através de equações cinemáticas, uma vez que sua dinâmica é desconhecida, pela própria natureza do problema;
- Simulação das medidas de posição e velocidade do veículo aeroespacial, e do ângulo que este faz com o veículo medido, considerando ruídos e incertezas nas medidas, para que se pudesse proceder a filtragem dos dados a fim de se obter uma medida mais confiável;
- Filtragem das medidas simuladas através do filtro de *Kalman*, utilizando o modelo teórico impreciso, para a obtenção de uma estimativa do vetor de estados com valores próximos do valor exato;
- Utilização das estimativas obtidas com o Filtro de *Kalman* no controlador, o que permitirá uma manobra adequada para a situação de rastreamento, comparando este resultado com a situação ideal, onde o controlador seria alimentado com o próprio estado real;
- Simulação do problema, incluindo o estimador e o sistema de controle do veículo, em dois ambientes de MIS - MATLAB®/ *Simulink* e MATRIXx®/ *SystemBuild*;
- Análise das semelhanças e diferenças nas funcionalidades e apresentação de resultados de ambos os ambientes utilizados, comparação dos resultados oferecidos por ambos, segundo critérios previamente estabelecidos.

1.2 Motivações

As principais motivações para o desenvolvimento deste trabalho incluem: (*i*) Exploração de mais de um modelo, com diferentes graus de realismo, a fim de se verificar os benefícios e perdas na ponderação entre simples e realista na modelagem do problema; (*ii*) Projetar para cada um destes modelos uma lei de controle que permita uma situação de rastreamento de trajetória, utilizando a teoria de controle ótimo; (*iii*) Verificar o desempenho da lei de controle diante de uma situação mais realista, onde se consideram as incertezas da modelagem e ruídos dos sensores; (*iv*) Explorar a técnica de linearização de modelos, a fim de adequar as teorias lineares existentes às plantas utilizadas, a fim de constatar até onde esta aproximação é válida.

Por ser um problema realista, a variedade de modelos utilizados permitirá uma análise de quão fiel o modelo deve ser à realidade, considerando a simplicidade desejada para sua simulação, e até mesmo para sua implementação em um possível processador embarcado. Assim, pode-se chegar a um grau de realismo que, embora não seja o ideal, atende os requisitos do problema sendo ao mesmo tempo satisfatório e factível.

1.3 Justificativa do Trabalho

Ao lado das motivações acima citadas, dois aspectos deste trabalho são justificados, também, pelo interesse de uma indústria local da área Aeroespacial, a Mectron EIC Ltda., em dois aspectos do trabalho.

Um dos motivos do interesse da empresa é investigar leis de controle menos simples do que as utilizadas no meio até há um tempo atrás (PI, PD, PID), mas, ao mesmo tempo, mais robustas, uma vez que a capacidade de processamento dos computadores embarcados vem aumentando com o passar do tempo.

Além disso, há o interesse pela própria simulação dos sistemas, que é feito em dois ambientes de Modelagem, Identificação e Simulação - MIS. Os dois ambientes utilizados neste trabalho, o MATLAB® e o MATRIXx®, são similares e bastante conhecidos no meio aeroespacial. Porém, existem funcionalidades diferentes em ambos,

além de apresentarem algumas características bastante particulares. Portanto, existe o interesse da Mectron EIC Ltda. na comparação destes ambientes, visando uma análise que ajudaria uma possível escolha do que mais atende às suas necessidades, sob vários pontos de vista.

1.4 Organização deste Trabalho

Este trabalho está dividido em seis capítulos. No segundo Capítulo faz-se uma revisão da literatura, abordando os aspectos multidisciplinares deste trabalho, como modelagem, definição de referenciais, influência da atmosfera, controle, estimação de estados e os próprios ambientes em que o problema foi simulado. No terceiro Capítulo apresenta-se a metodologia do trabalho, desde a formulação do problema até sua modelagem matemática, de cada uma das partes envolvidas no trabalho. As simulações e os resultados estão apresentados no quarto Capítulo. No quinto Capítulo analisam-se os ambientes de MIS utilizados no trabalho, apresentando as conclusões e sugestões de trabalhos futuros no sexto Capítulo. Por fim, apresentam-se nos Apêndices a base teórica da linearização de sistemas, equações dinâmicas que regem o comportamento de um veículo aeroespacial e o Teorema Pi, citado no trabalho.

CAPÍTULO 2

REVISÃO DA LITERATURA E CONCEITOS BÁSICOS

O trabalho aqui apresentado possui um caráter multidisciplinar, envolvendo conceitos desde modelagem de sistemas até sua simulação e implementação em ambientes de Modelagem, Identificação e Simulação - MIS, considerando, ainda, alguns aspectos práticos relativos a uma possível aplicação.

Assim, assuntos como modelagem de sistemas, a dinâmica de veículos aeroespaciais (considerando efeitos da sua interação com a atmosfera no movimento), sistemas de controle (em particular, controle ótimo), estimação de estados (neste caso, o Filtro de *Kalman*) e simulação foram estudados para o desenvolvimento deste trabalho.

A teoria de controle aplicada neste trabalho visa fazer com que o veículo aeroespacial siga uma trajetória pré-definida, dada uma tolerância ao erro entre a trajetória desejada e a seguida. No problema específico aqui abordado não se busca uma situação de *rendez-vous*, admitindo, portanto, uma tolerância maior ao erro.

A principal referência na área de veículos aeroespaciais semelhantes ao veículo modelado neste trabalho (veículo voando na atmosfera, com trajetória definida, etc.) é Blakelock (1991), que teve sua primeira edição em 1965. O controle destes veículos também é abordado em trabalhos como o de Sobey e Suggs (1963), que tratam do controle de veículos aeroespaciais desde seus conceitos mais básicos, como controlabilidade do sistema, até os requisitos para aplicações e os elementos computacionais envolvidos. Garnell (1980) também faz uma explicação detalhada, tratando dos métodos de controle utilizados (tais como controle de rolamento, controle lateral, entre outros) assim como considera também efeitos aerodinâmicos, entre outros assuntos de interesse na área. Cheng e Gupta (1986) apresentam um modelo de um veículo aeroespacial desta classe, tratando de uma técnica de separação dos estados baseados nas suas constantes temporais. Este modelo, reduzido para duas dimensões, é um dos modelos simulados neste trabalho. O trabalho de Lin e Su (2000) analisa a teoria de controle inteligente em guiagem e controle, incluindo o uso de redes neurais e lógica *fuzzy*. Este trabalho lembra o fato de que um número grande de novas tecnologias de controle têm sido desenvolvidas visando este tipo de problemas, com o objetivo principal de se cumprir a missão apesar dos distúrbios ambientais. Estas

técnicas são baseadas, principalmente, na Teoria de Controle Clássico.

No Brasil trabalhos na área são desenvolvidos especialmente no Instituto Militar de Engenharia e no Instituto Tecnológico de Aeronáutica. Leite Filho (1982) trata da modelagem analógica de um veículo aeroespacial e sua pilotagem na mesma fase de cruzeiro, enquanto Carvalho (1989) aborda uma técnica de controle ótimo em tempo real, embora com um enfoque diferente do aqui apresentado: seu enfoque é em controle ótimo, em tempo real, porém utilizando o algoritmo de múltiplos tiros.

As técnicas utilizadas ao longo do desenvolvimento deste trabalho são apresentadas a seguir:

2.1 Modelagem

Para a avaliação de um sistema é necessário ter em mãos um modelo para tal sistema. Um modelo é uma representação, no caso deste trabalho, matemática (seja dinâmica ou cinemática), obtida através de teoria e/ou de experimentos. Tais modelos podem ter diferentes graus de realismo, que refletem um compromisso entre a simplicidade da representação e da sua fidelidade em relação ao sistema real (Johansson, 1993).

Modelos são necessários não somente para avaliar e projetar um sistema, mas também, em muitos casos, para se projetar uma lei de controle que force seu comportamento a seguir determinado padrão desejado.

Em um projeto de engenharia, há um compromisso entre realismo e simplicidade em um modelo de um sistema físico. O motivo pelo qual não se usa um modelo totalmente completo em um projeto destes deve-se, entre outras coisas, ao custo envolvido, à praticidade e conveniência de simulá-lo (computacional ou fisicamente), e, se for o caso, de embarcá-lo. Portanto, a aplicação do modelo delimita o realismo a ser considerado.

Johansson (1993) é uma referência padrão para a Modelagem de Sistemas e sua Identificação. Aborda não somente o papel que a modelagem tem na solução de problemas científicos, mas também variadas técnicas de modelagem, teóricas ou baseadas em dados experimentais, assim como identificação de modelos e sua validação.

O assunto também é tratado em Leigh (1980), onde se apresenta uma metodologia para a construção de modelos de vários tipos, inclusive técnicas de linearização. Outra referência para a aplicação específica deste trabalho é encontrada em Lipscombe (1980), que aborda a modelagem de sistemas aeroespaciais, seus sistemas de controle, filtros e forças aerodinâmicas.

Uma forma de se escolher uma determinada técnica de modelagem é visando o projeto de uma planta ou seu controle. Modelos que visam auxiliar o projeto de uma planta qualquer têm como características o fato de serem baseados em leis físicas, tão detalhadas quanto necessário, podendo ser dinâmicos ou não; e, dentro da realidade de uma indústria, economicamente viáveis de se analisar e reproduzir.

Modelos que auxiliam o projeto de uma lei de controle que governará a evolução de uma planta têm como característica a preferência por modelos dinâmicos que descrevam processos e operações envolvendo a planta a ser controlada, podendo ser simplificados ou não. Eles podem incluir determinadas variáveis que permitirão a análise econômica para se verificar a viabilidade de se implementar a lei projetada, além de poder contar com uma versão simplificada da planta que poderá auxiliar uma eventual utilização *on line* em partes do sistema de controle (Leigh, 1980).

Neste trabalho a modelagem que nos interessa é a modelagem feita através de equações matemáticas que se acredita que representem sistema. Geralmente a estrutura de tal sistema é amplamente conhecida no meio (Johansson, 1993). Neste caso, partindo-se de um modelo ideal e bastante simplificado, acrescentam-se níveis de fidelidade à medida que eles se façam necessários, para que, ao fim do processo, o modelo dê origem a uma estrutura física (passo este não incluído neste trabalho). Aliando-se às equações dinâmicas aqui utilizadas, contamos aqui com o auxílio de tabelas de valores relativos às variáveis utilizadas, que completam o modelo matemático considerado.

A descrição matemática de um sistema pode ser feita através de equações matemáticas, sejam estas algébricas ou dinâmicas. Estas equações são selecionadas de forma a dar origem a uma estrutura para o modelo matemático procurado. O formato destas equações determina se um sistema é linear ou não. Um sistema é dito linear se os efeitos das forças que atuam sobre ele se sobrepõem, ou seja, se a soma dos efeitos das forças equivale ao efeito das forças adicionadas (Takahashi *et*

al., 1972).

A resolução destas equações é feita através de técnicas algébricas ou numéricas. Como existem parâmetros difíceis de serem calculados (como coeficientes aerodinâmicos, que assumem diferentes valores dependendo das características físicas de cada veículo e das condições de vôo), estes podem ser determinados de forma empírica ou semi-empírica, e serão utilizados em conjunto com a modelagem teórica do problema. Deve-se também ter o cuidado de selecionar todas as equações que produzam efeitos de mesma grandeza no sistema. À medida que se faça necessário, acrescentam-se mais equações ao modelo, para deixá-lo mais fiel à realidade, ponderando-se, no entanto, o peso da complexidade que tal procedimento implica.

A modelagem de veículos aeroespaciais geralmente contém duas partes principais (Lipscombe, 1980), a saber: (*i*) geometria e cinemática tridimensionais e (*ii*) dinâmica de um corpo rígido e do fluido (ar) através do qual ele se move. Deve incluir a geometria do veículo, eixos de referência, dinâmica, efeitos de perturbações, entre outros efeitos.

As equações que descrevem a evolução temporal do sistema muitas vezes são não-lineares e/ou acopladas, o que dificulta bastante a solução das equações. Também, muitas das técnicas de modelagem, de projeto do sistema de controle, assim como da estimação de estados foram deduzidas para sistemas lineares, ainda que variantes no tempo. Uma das técnicas para a adequação do modelo à teoria, neste caso consiste na linearização do modelo. Leigh (1980) apresenta a linearização feita por série de *Taylor* em sistemas onde a não-linearidade é definida analiticamente. O desenvolvimento teórico da série de *Taylor* se encontra no Apêndice A.

2.2 Referenciais

Em um problema aerodinâmico, geralmente, para uma descrição correta do problema a ser estudado, é necessário especificar a posição e a orientação dos veículos em relação a um referencial inercial. Entre outros motivos, isto se dá porque alguns dos sensores realizam medidas no referencial do veículo, enquanto as equações de movimento são deduzidas em um referencial inercial, onde valem as leis de movimento de *Newton* e toda a formulação da Mecânica Clássica.

Consideraremos, neste trabalho, um referencial fixo à Terra como inercial. A justificativa para isto é a brevidade do vôo, e, conseqüentemente, do rastreamento e valores das grandezas envolvidas no trabalho. Para especificar a orientação dos veículos, utilizam-se os ângulos de *Euler*, que relacionam referenciais que giram um em relação ao outro. Para auxiliar a compreensão da disposição dos referenciais, observe-se a Figura 2.1, em que OX_E, OY_E, OZ_E é o referencial inercial e OX, OY, OZ é o referencial fixo no veículo.

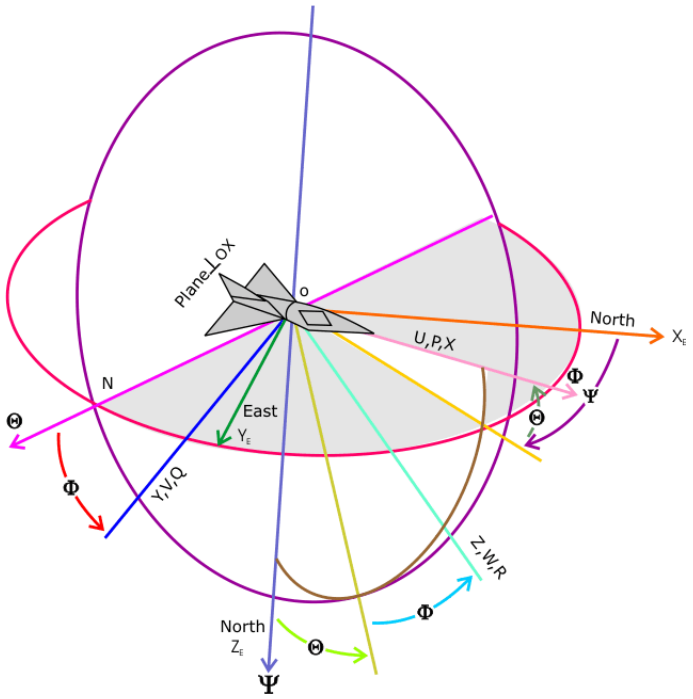


FIGURA 2.1 – Eixos fixos no referencial inercial (OX_E, OY_E, OZ_E) e no veículo (OX, OY, OZ).

FONTE: Blakelock (1991).

Sejam, na Figura 2.1, OX_E e OY_E os eixos que definem o plano horizontal, e seja OZ_E o eixo vertical, apontando para baixo. OX_E pode estar apontando para o Norte ou outra direção previamente definida. As grandezas indicadas na Figura 2.1 são:

- Ψ é o ângulo entre OX_E e a projeção do eixo OX no plano horizontal
- $\dot{\vec{\Psi}}$ é o vetor ao longo da direção OZ_E
- Θ é o ângulo entre o plano horizontal e o eixo OX medido no plano vertical

- $\dot{\vec{\Theta}}$ é o vetor ao longo de ON, a linha dos nodos (linha de interseção entre o plano de vôo do veículo e o plano orbital da Terra)
- Φ é o ângulo entre os eixos ON e OY medidos no plano OYZ, não necessariamente vertical
- $\dot{\vec{\Phi}}$ é o vetor ao longo da direção OX.

Assim, os ângulos Ψ, Θ, Φ especificam a atitude do veículo em relação à Terra. As direções positivas de tais ângulos estão indicadas na própria Figura 2.1.

Para se obter as componentes da velocidade angular no referencial do corpo, é necessário realizar a projeção de $\dot{\vec{\Psi}}$, $\dot{\vec{\Theta}}$, $\dot{\vec{\Phi}}$ nas direções OX, OY, OZ:

$$p = \dot{\Phi} - \dot{\Psi}\sin\Theta \tag{2.1}$$
$$q = \dot{\Theta}\cos\Phi + \dot{\Psi}\cos\Theta\sin\Phi \tag{2.2}$$
$$r = -\dot{\Theta}\sin\Phi + \dot{\Psi}\cos\Theta\cos\Phi \tag{2.3}$$

Para se obter Ψ, Θ e Φ, deve-se integrar $\dot{\Psi}$, $\dot{\Theta}$ e $\dot{\Phi}$ com as medidas de p, q, r e os valores de Ψ, Θ e Φ anteriores. Resolvendo Equações (2.1), (2.2), (2.3) para $\dot{\Psi}$, $\dot{\Theta}$, $\dot{\Phi}$ obtém-se:

$$\dot{\Theta} = q\cos\Phi - r\sin\Phi \tag{2.4}$$
$$\dot{\Phi} = p + q\sin\Phi\tan\Theta + r\cos\Phi\tan\Theta \tag{2.5}$$
$$\dot{\Psi} = q\frac{\sin\Phi}{\cos\Theta} + r\frac{\cos\Phi}{\cos\Theta} \tag{2.6}$$

Observe-se que os vetores $\dot{\vec{\Psi}}$, $\dot{\vec{\Theta}}$, $\dot{\vec{\Phi}}$ não são ortogonais entre si. Neste referencial do corpo onde as velocidades angulares foram acima deduzidas, têm-se as componentes da aceleração gravitacional, g, dadas por:

$$OX \quad : \quad -g \sin \Theta \tag{2.7}$$

$$OY \quad : \quad g \cos \Theta \sin \Phi \tag{2.8}$$

$$OZ \quad : \quad g \cos \Theta \cos \Phi \tag{2.9}$$

A abordagem dada aos modelos neste trabalho permite que todo o desenvolvimento seja feito no referencial inercial, dadas os estados definidos na sua dinâmica. Portanto, consideramos aqui apenas um referencial inercial, fixo à Terra.

2.3 Equações Dinâmicas

A dinâmica de veículos aeroespaciais é bastante abordada na literatura. Greensite (1970), Cornelisse *et al.* (1979), Blakelock (1991) e Stovall (1997) apresentam as equações que regem o movimento de um sólido, aplicáveis aos veículos aeroespaciais tratados neste trabalho. Stovall (1997) apresenta a abordagem mais simples entre os autores, sendo que os outros três fazem uma análise bastante completa.

As equações de movimento do veículo aeroespacial utilizado como modelo de simulação neste trabalho podem ser obtidas a partir da Segunda Lei de *Newton* (Cornelisse *et al.*, 1979; Blakelock, 1991), ou seja, as forças externas atuando no veículo devem ser iguais à taxa de variação no tempo do seu momento linear, e os torques externos se igualam à taxa de variação no tempo de seu momento angular, sendo que as grandezas e suas derivadas são definidas em relação ao referencial inercial. A dedução de tais equações podem ser encontradas no Apêndice B.

Nas equações dinâmicas consideram-se também os efeitos ambientais, tais como influências aerodinâmicas, ou seja, forças e torques que têm origem na interação do veículo com a camada atmosférica na qual ele se desloca. Possuem termos tais como viscosidade cinemática, sustentação, arrasto, que dependem de parâmetros do próprio veículo, assim como de propriedades do ar.

Também entram nas equações termos relativos ao controle, que neste caso, visa o rastreamento de uma trajetória pelo veículo aeroespacial. Para isto, parte- se do

princípio de minimização de um funcional. No caso específico deste rastreamento, a trajetória é definida por pontos que delimitam trechos os quais devem ser seguidos pelo veículo durante determinado intervalo de tempo. Assim, as duas trajetórias, a desejada e a real do veículo, devem se aproximar com o menor erro possível, respeitando, entretanto, a capacidade física do atuador responsável pela manobra. Não se exige neste caso que a distância mínima entre as trajetórias seja zero, e sim o mais próximas quanto seja possível.

As equações dinâmicas utilizadas neste trabalho são equações de movimento em um plano, obedecendo às Leis de *Newton*, sofrendo influência da gravidade, da atmosfera e do controle.

2.4 Atmosfera e Forças Aerodinâmicas

A influência da atmosfera sobre veículos aeroespaciais é tratada em Cornelisse *et al.* (1979) e Blakelock (1991), enquanto a dedução das forças, torques e coeficientes aerodinâmicos é feita em Kuethe e Chow (1950) e White (1981), referências estas que tratam da Análise Dimensional, particularmente aplicada à Aerodinâmica. Durand (1943), citado por Kuethe e Chow (1950), fornece suporte matemático à teoria da Análise Dimensional. Uma tabela com dados relevantes relacionados à atmosfera é fornecido pela National Aeronautics and Space Administration (1966).

Apesar de a massa da atmosfera não ser significativa ante a massa de toda a Terra, seus efeitos no deslocamento de veículos de trajetória baixa não são de forma alguma desprezíveis, devido às forças aerodinâmicas que surgem nestas circunstâncias.

No caso tratado neste trabalho, a região da atmosfera de interesse é a Troposfera, região mais baixa da atmosfera, que se extende por até 18 km sobre o Equador, e por até 8 km sobre os pólos (Cornelisse *et al.*, 1979).

A temperatura da Troposfera diminui à medida que se afasta da superfície da Terra (aquecida pelo Sol e principal fonte de calor para a Troposfera), assim como a sua densidade e pressão, que, na parte mais alta da Troposfera, chegam a valores que representam aproximadamente 30% e 22% dos valores assumidos na superfície da Terra (Cornelisse *et al.*, 1979).

Seu efeito sobre um veículo é bastante significativo, seja por fenômenos que podem ser previstos e considerados, como, por exemplo, o arrasto, até efeitos não previsíveis, tais como rajadas de vento e turbulências atmosféricas.

No primeiro caso, tais fenômenos podem e devem ser incluídos nas equações dinâmicas que ditam a trajetória do veículo, enquanto que no segundo caso, a dispersão da trajetória deve ser minimizada por um sistema de controle adequado.

As forças e momentos aerodinâmicos dependem, essencialmente, da velocidade do veículo em relação ao ar e da pressão, densidade e temperatura atmosféricas locais. Consideraremos aqui, a título de simplificação, que a atmosfera se mantém em repouso em relação à Terra. Esta simplificação não é crítica, uma vez que os sensores embarcados no veículo não são sensíveis a este movimento (Blakelock, 1991).

Parâmetros importantes para a análise da aerodinâmica do veículo que se segue são o número de *Mach*, M_a, o número de *Reynolds*, Re, e a pressão dinâmica, q_{din}, que serão definidos a seguir:

O número de *Mach*, M_a, representa uma comparação entre a velocidade do fluido (ar, neste caso) que se desloca pela superfície do veículo quando este se move na atmosfera e a velocidade do som. Quando igualado à unidade representa um fluxo sônico, enquanto que um número superior a um representa um fluxo supersônico, ou seja, a velocidade do fluido é maior do que a velocidade com que uma pequena perturbação na pressão se propaga pelo fluido (Cornelisse *et al.*, 1979). Sua definição matemática é:

$$M_a = \frac{V_T}{a} \tag{2.10}$$

Em que:

- M_a é o número de *Mach*
- V_T é a velocidade linear do ar na superfície do veículo
- a é a velocidade do som no ar

O número de *Reynolds*, *Re*, é considerado o parâmetro mais importante no estudo da Mecânica dos Fluidos (Cornelisse *et al.*, 1979). Representa a importância relativa das forças inerciais em relação às forças viscosas, sendo dado por:

$$Re = \frac{\rho V_T l}{\mu} \tag{2.11}$$

Em que:

- Re é o número de *Reynolds*
- ρ é a densidade do ar
- l é o comprimento característico do veículo
- μ é a viscosidade dinâmica do ar

A pressão dinâmica q_{din}, pressão relacionada com o movimento do ar (Kuethe e Chow, 1950), é definida como:

$$q_{din} = \frac{1}{2}\rho V_T^2 \tag{2.12}$$

Tais números podem ser obtidos através da Análise Dimensional (White, 1981). A Análise Dimensional consiste de um método para reduzir a complexidade e variedade de variáveis experimentais, agrupando estas variáveis em números adimensionais, cujo conjunto, e não as variáveis em separado, influenciam certos fenômenos.

O método consiste em se considerar que se uma equação matemática, em especial uma equação algébrica, representa fenômenos físicos, obrigatoriamente esta equação deve ser consistente dimensionalmente (Kuethe e Chow, 1950). Este fato ajuda a agrupar variáveis em combinações relevantes, chamadas números adimensionais, que sempre ocorrerão na descrição matemática do problema, fazendo com que as equações originais possam ser escritas em termos de tais números adimensionais.

A propriedade matemática que garante isto é apresentada pelo Teorema Pi, que será apenas enunciado neste trabalho no Apêndice C, sendo sua prova encontrada em Durand (1943), citado por Kuethe e Chow (1950).

Assim, supondo-se que uma força de módulo F experimentada por um corpo se deslocando em um fluido, como o ar, depende apenas da densidade do fluido, ρ, da velocidade linear do fluido sobre a superfície do corpo, V_T, do comprimento característico do veículo, l, da viscosidade do fluido, μ, e da velocidade do som, a, ou seja:

$$F = f\left(\rho, V_T, l, \mu, a\right) \tag{2.13}$$

Podemos escrever:

$$g = F - f\left(\rho, V_T, l, \mu, a\right) \tag{2.14}$$

$$g\left(F, \rho, V_T, l, \mu, a\right) = 0 \tag{2.15}$$

E, sabendo que esta equação contém seis variáveis mas apenas três de dimensões fundamentais, (ρ, V_T, l) constata-se que há três produtos Π, envolvendo as demais variáveis (F, μ, a). Escolhendo o grupo de variáveis K como constituído das variáveis ρ, l, V_T, temos os seguintes produtos Π:

$$\Pi_1 = f_1\left\{F, \rho, V_T, l\right\} \tag{2.16}$$

$$\Pi_2 = f_2\left\{\mu, \rho, V_T, l\right\} \tag{2.17}$$

$$\Pi_3 = f_3\left\{a, \rho, V_T, l\right\} \tag{2.18}$$

Sabendo que estas variáveis devem ser reunidas de forma a constituírem grupos adimensionais, chega-se a:

$$\Pi_1 = \frac{F}{\rho V_T^2 l^2} \tag{2.19}$$

$$\Pi_2 = \frac{\rho V_T l}{\mu} \tag{2.20}$$

$$\Pi_3 = \frac{V_T}{a} \tag{2.21}$$

em que podemos reconhecer Π_2 e Π_3 como sendo, respectivamente, o número de *Reynolds* e o número de *Mach*, enquanto Π_1 representa o coeficiente de força adimensional, C, que é uma função apenas do número de *Reynolds* (White, 1981). Logo:

$$F = \frac{1}{2} C \rho V_T^2 l^2 \tag{2.22}$$

Em que $C = C(Re, l)$, e l é influenciado diretamente pelo ângulo de ataque.

Sabendo-se que as grandezas μ, ρ, a, entre outras, variam grandemente não só com o local considerado, mas também com o tempo, na prática o que se faz é a estimativa de seus valores médios, utilizados em modelos de Atmosfera de Referência ou mesmo a Atmosfera Padrão (Cornelisse *et al.*, 1979). Neste trabalho só consideraremos valores tabelados para estas grandezas.

Para deduzir as expressões da força aerodinâmica serão feitas algumas simplificações, a saber: o vetor velocidade do veículo pertence a um dos planos de simetria do veículo; um dos planos de simetria coincide com o plano de trajetória do veículo. Chamaremos de ângulo de ataque ou incidência, representado por α, o ângulo do vetor velocidade linear do veículo para seu eixo longitudinal. Caso o vetor velocidade estivesse em outro plano de simetria, definiríamos um ângulo de derrapagem, representado por β. Estes nomes são padrão no estudo da aerodinâmica de veículos aeroespaciais (Cornelisse *et al.*, 1979). Para a definição das grandezas aqui mencionadas, observe a Figura 2.2.

Assumimos também que a força aerodinâmica total, $\vec{F}_a$, também se encontra no

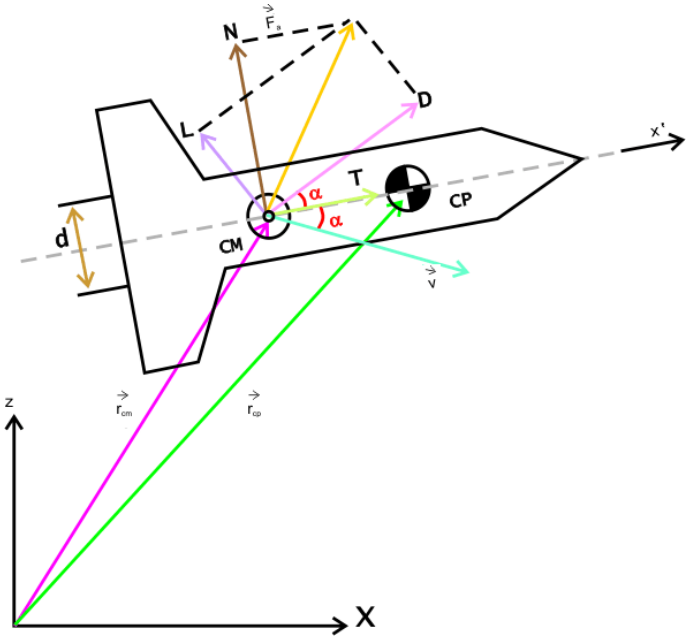


FIGURA 2.2 – Forças aerodinâmicas em um veículo, centros de pressão e massa em uma manobra plana.

FONTE: Adaptada de Cornelisse *et al.* (1979).

plano da trajetória, e o ponto onde a linha de força de $\vec{F}_a$ encontra a linha de centro do veículo é chamado centro de pressão, *CP*. Se não há resultante de momento aerodinâmico no veículo, *CP* coincide com o centro de massa *CM*, que também assumimos aqui estar sobre a linha de centro do veículo. Caso não coincidam, haverá um momento aerodinâmico atuando no veículo, na direção do eixo y, chamado momento de arfagem, cuja expressão é dada por:

$$\vec{M'} = \vec{F}_a \times \Delta\vec{r} \tag{2.23}$$

em que:

$$\Delta\vec{r} = \vec{r}_{cm} - \vec{r}_{cp} \tag{2.24}$$

conforme a Figura 2.2.

Pode-se decompor a força aerodinâmica $\vec{F_a}$ em duas componentes: a sustentação $\vec{L}$ e o arrasto, $\vec{D}$, normal e paralela, respectivamente, à velocidade $\vec{V_T}$; ou também em uma força normal $\vec{N}$ e uma força tangencial, $\vec{T}$, normal e paralela, respectivamente, à linha de centro do veículo aeroespacial.

Momentos aerodinâmicos que venham a surgir nas direções z e x são chamados momentos de guinada e rolagem, respectivamente, sendo representados por $\vec{N'}$ e $\vec{L'}$, não indicados na Figura 2.2.

Para descrever fenômenos aerodinâmicos em função de coeficientes adimensionais, definimos uma área para referência, que geralmente é tomada como a área da seção reta do veículo (Cornelisse *et al.*, 1979). Assim, denominando o diâmetro da base do veículo d, tem-se a área da sua seção reta:

$$S = \frac{\pi}{4} d^2 \tag{2.25}$$

E, representando o volume como Sd, teremos as expressões para forças e momentos:

$$\begin{aligned} L &= C_L q_{din} S && (2.26)\\ D &= C_D q_{din} S && (2.27)\\ L' &= C_l q_{din} S d && (2.28)\\ M' &= C_m q_{din} S d && (2.29)\\ N' &= C_n q_{din} S d && (2.30) \end{aligned}$$

Assim definem-se sustentação, arrasto e momentos aerodinâmicos, além de se definirem de forma análoga as forças normal e tangencial, entre outras grandezas.

Em geral, para casos em três dimensões aparecem três coeficientes de força e três coeficientes de torque independentes. Já no caso bidimensional, onde se considera que as forças aerodinâmicas se encontram em um plano de simetria, há apenas dois coeficientes de força, C_L e C_D e um coeficiente de torque, C_m independentes. A

determinação destes coeficientes considera, principalmente os números de *Mach* e *Reynolds* envolvidos no problema.

Supondo que estas forças e momentos variam pouco em torno de um certo valor, pode-se realizar uma expansão das forças e momentos em torno do ponto de operação através da série de *Taylor*. Ignorando termos de ordem superior desta série chegamos ainda a outros coeficientes aerodinâmicos que são chamados derivadas de estabilidade. Assim, teremos a expansão de *Taylor*:

$$F = F_0 + F_\alpha \alpha + F_{\dot{\alpha}} \dot{\alpha} + \ldots \tag{2.31}$$

$$M' = M'_0 + M'_\alpha \alpha + M'_q q + M'_{\dot{\alpha}} \dot{\alpha} + \ldots \tag{2.32}$$

em que:

- F_0 é constante, sendo nula para veículos simétricos
- M'_0 é constante, sendo nulo para veículos simétricos
- $F_\alpha \equiv \dfrac{\partial F}{\partial \alpha}$, padrão seguido para as outras variáveis

É possível reescrever as equações 2.31 e 2.32 da forma:

$$F = \left(C_{F_0} + C_{F_\alpha} \alpha\right) q_{din} S + \left(C_{F_{\dot{\alpha}}} \dot{\alpha} + C_{F_q} q\right) q_{din} S \frac{d}{2V_T} + \ldots \tag{2.33}$$

$$M' = \left(C_{m_0} + C_{m_\alpha} \alpha\right) q_{din} S d + \left(C_{m_{\dot{\alpha}}} \dot{\alpha} + C_{m_q} q\right) q_{din} S \frac{d^2}{2V_T} + \ldots \tag{2.34}$$

O uso de um tempo adimensional faz com que o termo $\dfrac{d}{2V_T}$ surja, de forma que:

$$\dot{\alpha} = \frac{\partial \alpha}{\partial \left(\frac{2V_T t}{d} \right)} \tag{2.35}$$

Estes termos, tais como C_{F_α}, C_{m_q}, etc. são chamados derivadas de estabilidade. As derivadas de estabilidade relativas a amortecimentos não serão usadas neste trabalho, uma vez que seus valores são muito pequenos em relação aos outros parâmetros envolvidos no fenômeno para o veículo em questão.

A influência da atmosfera considerada aqui neste trabalho será o arrasto atmosférico. Para calcular sua influência, consideramos a altura na qual o veículo voa (que afetará a densidade do ar), e o coeficiente aerodinâmico C_D, que por sua vez depende do número de *Mach*.

2.5 Controle

O controle é o sistema da planta responsável por fazê-la evoluir no tempo conforme se deseja. Para isto, deve-se formular uma lei de controle, que é uma expressão para definir a evolução desejada para o sistema, ou a expressão que fará com que o sistema se comporte como se objetiva. Controle ótimo é a lei de controle que faz com que o sistema evolua conforme estabelecido pela lei, porém de forma a otimizar algum dos parâmetros envolvidos no problema, ou um conjunto deles, tais como tempo, trajetória, consumo de combustível, etc. No caso do controle ótimo que visa diminuir (ou otimizar) a diferença entre a trajetória seguida no espaço de estados por um veículo e a trajetória desejada, define-se o rastreamento de uma trajetória. No caso de a trajetória desejada estar na origem do espaço de estados, temos o caso particular de um regulador (Kirk, 1970).

Técnicas de Controle Clássico podem ser encontradas em Ogata (1997), que discute o projeto de controle sem, no entanto, enfatizar aspectos de estimação nem de otimização da lei de controle projetada. A teoria de controle voltada para o caso específico de veículos aeroespaciais pode ser encontrada nos livros de Greensite (1970), Cornelisse *et al.* (1979) e Blakelock (1991), que abordam inclusive alguns tipos de atuadores existentes de forma particular. Projetos de controle ótimo podem ser encontrados em Kirk (1970), onde são tratadas técnicas de otimização de sistemas,

voltadas para controle. É a referência mais didática encontrada na área de Cálculo Variacional e Controle Ótimo, tratando desde problemas de rastreamento, até situações de *rendez-vous*. Este assunto pode, ainda, ser encontrado em Takahashi *et al.* (1972) e Bryson Jr e Ho (1975), que abordam o contexto do controle ótimo, assim como os casos de rastreadores e reguladores em particular. Apesar de tratarem apenas de controles determinísticos, estas referências podem ser utilizadas para controle baseado em estimações de estado detalhadamente estudadas no livro de Maybeck (1979).

Sistemas de controle desempenham, nos dias atuais, um papel muito importante em projetos de engenharia, em particular, pilotagem, guiagem, controle de temperatura, entre outros subsistemas de veículos aeroespaciais. Podem ser basicamente de dois tipos: em malha aberta ou em malha fechada (Ogata, 1997).

Sistemas de controle em malha aberta são sistemas cujas saídas não afetam a ação do controle. Portanto, não há necessidade de se medir as saídas e muito menos de reenviá-las ao controlador. O controle corresponde a uma operação fixa, em função do tempo, que pode não produzir o resultado desejado caso as outras entradas sejam muito diferentes das entradas previstas, ou mesmo em caso de distúrbios. Por não ter realimentação, deve ser cuidadosamente calibrado para que o resultado do controle seja satisfatório. O controle em malha aberta deve ser utilizado quando se conhece com razoável certeza as outras variáveis de entrada da planta, assim como suas saídas, e na ausência de distúrbios, sejam eles internos ou externos. Em tais tipos de plantas o controle em malha aberta é satisfatório, proporcionando ainda um baixo custo para a planta controlada.

Sistemas de controle em malha fechada medem a diferença entre o estado real da planta e o estado desejado e produzem uma série de ações que, idealmente, diminuirão este erro através de processos interativos (Johansson, 1993); ou seja, ao contrário do sistema em malha aberta, se baseia em realimentações dos estados na lei de controle. O estado desejado pode conter não somente o estado medido, mas também suas derivadas e/ ou integrais, entre outros tipos de combinações.

Portanto, para se controlar um veículo aeroespacial, deve-se conhecer com razoável precisão seu estado real, uma vez que ele está sujeito a distúrbios internos e externos, nem sempre possíveis de serem modelados, além de ter entradas variáveis. O estado

real pode ser aproximado por modelos dinâmicos propagados, medidas fornecidas por sensores, ou, idealmente, uma ponderação entre os dois, feita através de um filtro (Estimação de Estados).

2.5.1 Controle Ótimo

No caso tratado neste trabalho, é desejável se formular uma lei de controle ótima, ou seja, determinar a lei de controle que satisfaça os vínculos presentes no sistema, extremando algum critério de desempenho. Este critério deverá refletir o desejo de se levar o estado do sistema próximo a um estado de referência, considerando os vínculos físicos envolvidos no problema.

Formular uma lei de controle ótimo é encontrar a melhor estratégia de controle, dados a planta a ser controlada, a função custo a ser extremada e os vínculos físicos. Geralmente esta lei depende da saída do sistema, representando, portanto, um sistema de controle em malha fechada.

A técnica para se determinar tal controle evoluiu do Cálculo Variacional, e, para aplicá-la, deve-se inicialmente formular o problema como um problema de Controle Ótimo, seguindo os passos a seguir (Takahashi *et al.*, 1972):

1) Fazer uma descrição matemática do processo ou planta a ser controlada, em forma de equações dinâmicas, $\forall t \in [t_0, t_f]$ como:

$$\begin{aligned} \dot{\underline{x}}(t) &= \underline{f}(\underline{x}(t), \underline{u}(t), t) \\ \underline{x}(t_0) &= \underline{x}_0 \end{aligned} \tag{2.36}$$

2) Determinar o estado inicial $\underline{x}(t_0)$ das equações dinâmicas que descrevem o sistema, assim como especificar o estado final $\underline{x}\,(t_f)$, em que t_f é o tempo final, não necessariamente especificado de antemão;

3) Definir o índice de desempenho J:

$$J = \int_{t_0}^{t_f} f_0\,(\underline{x}(t), \underline{u}(t), t)\,dt \tag{2.37}$$

em que:

- f_0 é uma função custo pré-determinada
- $\underline{x}(t)$ representa o vetor de estados do sistema, em função do tempo
- $\underline{u}(t)$ representa o vetor de controle atuando sobre o sistema, em função do tempo;

4) Determinar os vínculos físicos no vetor de estados $\underline{x}(t)$ e/ ou no vetor de controle $\underline{u}(t)$, ou seja, delimitar a região no Espaço de Estados onde o vetor $\underline{x}(t)$ pode estar, e também o esforço de controle e sua variação que podem ser despendidos.

No caso deste trabalho, o índice de desempenho deve refletir o erro entre a trajetória desejada e a trajetória real que o estado do veículo descreve, ou seja, tem-se aqui um problema de rastreamento.

O problema de rastreamento consiste em levar e manter o estado $\underline{x}(t)$ do sistema tão próximo quanto possível do estado desejado $\underline{r}(t)$ no intervalo $[t_0, t_f]$.

Para isto, o índice de desempenho quadrático deste problema seria:

$$J = \frac{1}{2}\int_{t_0}^{t_f} \left[\underline{x}\left(t\right) - \underline{r}\left(t\right)\right]^T Q_R\left(t\right)\left[\underline{x}\left(t\right) - \underline{r}\left(t\right)\right] dt \tag{2.38}$$

$$= \frac{1}{2}\int_{t_0}^{t_f} \| \underline{x}\left(t\right) - \underline{r}\left(t\right) \|^2_{Q_R(t)} dt \tag{2.39}$$

em que $Q_R\left(t\right)$ é uma matriz $n \times n$ positiva semi definida $\forall t \in [t_0, t_f]$. Os elementos da matriz $Q_R\left(t\right)$ são selecionados para definir um peso da importância relativa dos desvios nas diferentes componentes do vetor de estados e para normalizar os valores numéricos dos desvios. Por exemplo, se $Q_R\left(t\right)$ é uma matriz diagonal constante e q_{ii} é zero, isto indica que os desvios de x_i não influenciam o índice de desempenho.

Se o conjunto de controles admissíveis é limitado, *i.e.*, $|u_i\left(t\right)| < 1$, $i = 1, 2, \ldots, m$, então o funcional acima é um funcional razoável para o problema. Entretanto, para o caso de controles não limitados, a minimização daquele funcional resultará em controles com impulsos e suas derivadas. Para evitar a necessidade de se limitar os

valores admissíveis de controle, ou considerando a energia do controle conservada, podemos utilizar o funcional quadrático modificado, que considera as limitações no controle:

$$J = \frac{1}{2}\int_{t_0}^{t_f} \left\{ \| \underline{x}(t) - \underline{r}(t) \|^2_{Q_R(t)} + \left[\underline{u}(t)^T R_R \underline{u}(t) \right] \right\} dt \tag{2.40}$$

$$= \frac{1}{2}\int_{t_0}^{t_f} \left[\| \underline{x}(t) - \underline{r}(t) \|^2_{Q_R(t)} + \| \underline{u} \|^2_{R_R(t)} \right] dt \tag{2.41}$$

Sendo $R_R(t)$ uma matriz $m \times m$ positiva definida $\forall t \in [t_0, t_f]$ (Kirk, 1970).

Se a planta é linear, este funcional leva a um controlador ótimo mais fácil de se implementar. Pode ser especialmente importante que os estados sejam próximos ao valor desejado no tempo final. Neste caso, o funcional:

$$J = \frac{1}{2}\| \underline{x}(t_f) - \underline{r}(t_f) \|^2_{H_R} + \frac{1}{2}\int_{t_0}^{t_f} \left[\| \underline{x}(t) - \underline{r}(t) \|^2_{Q_R(t)} + \| \underline{u} \|^2_{R_R(t)} \right] dt \tag{2.42}$$

poderá ser utilizado. H_R é uma matriz real positiva semi definida $n \times n$.

Um caso especial do problema do rastreador é o regulador, em que o valor referência do vetor de estados é nulo, ou seja: $\underline{r}(t) = \underline{0}, \forall t \in [t_0, t_f]$. O regulador é um controlador em malha fechada que mantém um sistema estável dentro de uma região próxima à origem do Espaço de Estados sem, no entanto, dispender mais controle do que o sistema suporta (Bryson Jr e Ho, 1975). Para problemas envolvendo uma planta linear e um índice de desempenho quadrático, o controle ótimo pode ser encontrado como uma função linear variante no tempo dos estados do sistema. Sob certas condições esta lei de controle ótimo se torna invariante no tempo. Os resultados a seguir se devem aos trabalhos de *R. Kalman* (Kirk, 1970). Seja uma planta descrita pelas equações de estado lineares:

$$\dot{\underline{x}}(t) = A(t)\,\underline{x}(t) + B(t)\,\underline{u}(t) \tag{2.43}$$

Onde $\underline{x}(t_0) = \underline{x}_0$.

Na Equação 2.43 os coeficientes podem ser variantes no tempo. O funcional a ser minimizado é:

$$J = \frac{1}{2}\underline{x}^T(t_f)\, H_R \underline{x}(t_f) + \frac{1}{2}\int_{t_0}^{t_f} \left[\underline{x}^T(t)\, Q_R(t)\, \underline{x}(t) + \underline{u}^T(t)\, R_R(t)\, \underline{u}(t)\right] dt \quad (2.44)$$

O tempo final t_f é fixo, H_R e $Q_R(t)$ são matrizes reais positivas semi definidas, $R_R(t)$ é uma matriz real positiva definida. Assume-se que os estados e controles não são limitados, e $\underline{x}(t_f)$ é livre. O significado físico deste funcional é: *deseja-se manter o vetor de estados o mais próximo possível da origem sem um grande gasto de controle.* O Hamiltoniano é, $\forall t \in [t_0, t_f]$:

$$\begin{aligned} \mathscr{H}\left(\underline{x}(t), \underline{u}(t), \underline{p}(t), t\right) &= \frac{1}{2}\underline{x}^T(t)\, Q_R(t)\, \underline{x}(t) + \frac{1}{2}\underline{u}(t)\, R_R(t)\, \underline{u}(t) + \\ &\quad + \underline{p}^T(t)\, A(t)\, \underline{x}(t) + \underline{p}^T(t)\, B(t)\, \underline{u}(t) \end{aligned} \quad (2.45)$$

e as condições necessárias para otimalidade são, $\forall t \in [t_0, t_f]$:

$$\dot{\underline{x}}^*(t) = \frac{\partial \mathscr{H}}{\partial \underline{p}} = A(t)\underline{x}^*(t) + B(t)\underline{u}^*(t) \quad (2.46)$$

$$\dot{\underline{p}}^*(t) = -\frac{\partial \mathscr{H}}{\partial \underline{x}} = -Q_R(t)\underline{x}^*(t) - A^T(t)\underline{p}^*(t) \quad (2.47)$$

$$\underline{0} = \frac{\partial \mathscr{H}}{\partial \underline{u}} = R_R(t)\underline{u}^* + B^T(t)\underline{p}^*(t) \quad (2.48)$$

Com condições de contorno:

$$\underline{p}\,(t_f) \;=\; H_R \underline{x}\,(t_f) \tag{2.49}$$

ou seja, o controle ótimo fornece, $\forall t \in [t_0, t_f]$:

$$\underline{u}^*\,(t) = -R_R^{-1}\,(t)\,B^T\,(t)\,\underline{p}^*\,(t) \tag{2.50}$$

Observe-se que a existência de $R_R^{-1}(t)$ é garantida, por ser $R_R(t)$ uma matriz definida positiva.

R. Kalman provou que $p^*(t)$ é uma função linear dos estados do sistema (Kirk, 1970), ou seja, $\forall t \in [t_0, t_f]$:

$$p^*(t) \triangleq P_R(t)\underline{x}^*(t) \tag{2.51}$$

Em que $P_R(t)$ obedece a equação de *Riccati*:

$$\dot{P}_R(t) = -A^T(t)P_R(t) - P_R^T(t)A(t) - Q_R(t) + P_R(t)B(t)R_R^{-1}(t)B^T(t)P_R(t) \tag{2.52}$$

Com condição final $P_R(t_f) = H_R$.

Ou seja, o controle ótimo é dado por:

$$\underline{u}^*(t) \;=\; -R^{-1}(t)B^T(t)P_R(t)\underline{x}(t) \tag{2.53}$$

$$\;=\; -K(t)\underline{x}(t) \tag{2.54}$$

Em que $K(t) = R^{-1}(t)B^T(t)P_R(t)$.

Este controle pode ser aplicado continuamente ou discretamente na planta. No caso de aplicação discreta, escolhe-se uma taxa de amostragem na qual a matriz $K(t)$ será calculada.

No caso abordado neste trabalho, foi utilizado um regulador de erros, ou seja, espera-se levar o erro entre o estado desejado $\underline{r}(t)$ e o estado real $\underline{x}(t)$ a zero. Assim,

$$J = \frac{1}{2}\Delta\underline{x}^T\left(t_f\right)H_R\Delta\underline{x}\left(t_f\right) + \frac{1}{2}\int_0^{t_f}\left[\Delta\underline{x}^T\left(t\right)Q_R\left(t\right)\Delta\underline{x}\left(t\right) + \Delta\underline{u}^T\left(t\right)R_R\left(t\right)\Delta\underline{u}\left(t\right)\right]dt \tag{2.55}$$

O Hamiltoniano é, $\forall t \in [t_0, t_f]$:

$$\begin{aligned}\mathscr{H}\left(\Delta\underline{x}\left(t\right),\Delta\underline{u}\left(t\right),\underline{p}\left(t\right),t\right) &= \frac{1}{2}\Delta\underline{x}^T\left(t\right)Q_R\left(t\right)\Delta\underline{x}\left(t\right) + \frac{1}{2}\Delta\underline{u}\left(t\right)R_R\left(t\right)\Delta\underline{u}\left(t\right) + \\ &+\underline{p}^T\left(t\right)A\left(t\right)\Delta\underline{x}\left(t\right) + \underline{p}^T\left(t\right)B\left(t\right)\Delta\underline{u}\left(t\right)\end{aligned} \tag{2.56}$$

e as condições necessárias para otimalidade são:

$$\Delta\dot{\underline{x}}^*(t) = \frac{\partial\mathscr{H}}{\partial\underline{p}} = A(t)\Delta\underline{x}^*(t) + B(t)\Delta\underline{u}^*(t) \tag{2.57}$$

$$\dot{\underline{p}}^*(t) = -\frac{\partial\mathscr{H}}{\partial\Delta\underline{x}} = -Q_R(t)\Delta\underline{x}^*(t) - A^T(t)\underline{p}^*(t) \tag{2.58}$$

$$\underline{0} = \frac{\partial\mathscr{H}}{\partial\Delta\underline{u}} = R_R(t)\Delta\underline{u}^* + B^T(t)\underline{p}^*(t) \tag{2.59}$$

Com condições de contorno:

$$\underline{p}\,(t_f) = H_R \Delta \underline{x}\,(t_f) \tag{2.60}$$
$$\underline{x}(t_0) = \underline{x}_0 \tag{2.61}$$

ou seja, o controle ótimo fornece:

$$\Delta \underline{u}^*\,(t) = -R_R^{-1}\,(t)\,B^T\,(t)\,\underline{p}^*\,(t) \tag{2.62}$$

$$p^*(t) \triangleq P_R(t) \Delta \underline{x}^*(t) \tag{2.63}$$

Ou seja, o controle ótimo é dado por:

$$\Delta \underline{u}^*(t) = -R^{-1}(t) B^T(t) P_R(t) \Delta \underline{x}(t) \tag{2.64}$$
$$= -K(t) \Delta \underline{x}(t) \tag{2.65}$$

Em que $K(t) = R^{-1}(t) B^T(t) P_R(t)$.

Este incremento de controle, $\Delta \underline{u}(t)$ será somado ao controle de referência $\underline{u}_r(t)$.

2.6 Estimação de Estados

Maybeck (1979), Brown e Hwang (1997) e Welch e Bishop (2001) realizam uma abordagem completa da teoria da estimação. Nestas referências encontra-se desde a teoria axiomática de probabilidade, desenvolvendo os conceitos necessários para a compreensão de estimadores, até a dedução do filtro de *Kalman*. Também é apresentado nestas referências o contexto em que se insere a teoria da estimação, desenvolvendo o assunto desde modelos de sistemas determinísticos, passando pela inclusão de variáveis aleatórias até a dedução de estimadores e filtros. Maybeck (1979) apre-

senta, ainda, uma seção onde trata da aplicação do filtro de *Kalman* para sistemas de navegação inercial, amplamente aplicáveis neste trabalho. Uma abordagem superficial do filtro de *Kalman* também pode ser encontrada em Stovall (1997), enquanto que aplicações práticas foram bastante discutidas em Kuga (2003) e Lopes (2003).

A maneira como as variáveis do sistema estudado evoluem são de extrema importância para este trabalho. Isto pode ocorrer deterministicamente, ou seja, seguindo um padrão cuja evolução no tempo é conhecida, ou estocasticamente, onde o valor da variável em cada instante é dado de forma probabilística (Takahashi *et al.*, 1972). Neste último caso se incluem os casos de ruídos nos sensores e perturbações na dinâmica do sistema. Para se extrair deste sistema informações mais fiéis à realidade, deve-se proceder uma filtragem dos dados obtidos. Um filtro é um algoritmo que processa dados de forma a extrair deles informações relevantes. O Filtro de *Kalman* é o algoritmo recursivo ótimo para processamento de dados, no caso de evoluções lineares do sistema (Maybeck, 1979).

Usualmente, quando se projeta uma Lei de Controle para um sistema, seu desenvolvimento é feito assumindo que a planta a ser controlada é determinística. Assim, quando aplicado na planta real que sofre a influência de fenômenos não perfeitamente modelados ou mesmo não modelados e que conta ainda com sensores imperfeitos que fornecem medidas com ruídos e *bias*, o sistema de controle pode não funcionar exatamente da forma para a qual foi projetado.

Assim, para que a Lei de Controle funcione de forma apropriada, sua entrada deve ser a mais próxima possível do valor real do estado. Para tanto, o que se deve fazer é uma ponderação entre a informação prevista pelo modelo teórico e as informações ruidosas providas pelos sensores.

Neste trabalho, o algoritmo utilizado para tal ponderação é o Filtro de *Kalman*, que é o estimador ótimo para sistemas lineares. O Filtro de *Kalman* combina todas as medidas disponíveis com o conhecimento prévio da dinâmica do sistema e dos sensores de forma a minimizar o erro estatisticamente (Maybeck, 1979).

2.6.1 Filtro de Kalman

O filtro de *Kalman* pode englobar três casos distintos: dinâmica e medidas contínuas, dinâmica contínua e medidas discretas ou, ainda, dinâmica e medidas discretas. Neste trabalho abordaremos o caso onde a dinâmica é contínua e a medida é discreta.

Ainda, podemos classificar os possíveis tipos de filtro de *Kalman* como: filtro de *Kalman* propriamente dito, em que a dinâmica do sistema é linear, ainda que seja variante no tempo, e filtros estendido e linearizado de *Kalman*, onde a dinâmica do sistema não é linear, precisando, portanto, passar por uma linearização.

Trataremos aqui o caso do filtro de *Kalman* com dinâmica contínua e medidas discretas.

Para um sistema cuja dinâmica obedece a equação $\forall t \in [t_0, t_f]$:

$$\dot{\underline{x}} = A(t)\underline{x} + B(t)\underline{u} + G(t)\underline{\omega} \tag{2.66}$$

Com condição inicial $\underline{x}(t_0) = \underline{x}_0$.

E que tem medidas discretas, nos instantes $t_k = kT + t_0$, $\forall k \in \{0, 1, ..., N\}$, com $N = \dfrac{(t_f - t_0)}{T}$, onde T é o período de amostragem, indicadas pelo subscrito k:

$$\underline{z}_k = H_F\underline{x}_k + \underline{v}_k \tag{2.67}$$

Em que:

- $\underline{x}(t)$ é o estado a ser estimado
- $\underline{x}_0$ é o estado inicial, modelado como uma variável aleatória independente gaussiana $N(\underline{x}_0, P_{F0})$
- $A(t)$ é a matriz que descreve a dinâmica de $\underline{x}(t)$

- $B(t)$ é a matriz de ponderação do controle
- $\underline{u}(t)$ é o vetor de controle atuando na planta
- $G(t)$ é a matriz de adição do ruído dinâmico
- $\underline{z}_k$ é a medida do estado $\underline{x}$ realizada no instante t_k
- H_{Fk} é a matriz que relaciona as medidas $\underline{z}$ ao estado $\underline{x}$ no instante t_k
- $\underline{\omega}$ é o distúrbio sobre a dinâmica, modelada por um processo estocástico branco gaussiano $N(\underline{0}, Q_F)$
- $\underline{v}_k$ é o ruído sobre a medida no instante t_k, modelado por uma seqüência branca gaussiana $N(\underline{0}, R_{Fk})$

No caso de um sistema cuja dinâmica pode ser descrita de forma linear, as equações de propagação do filtro de *Kalman*, $\forall k \in \{0, 1, ..., N\}$, podem ser escritas da forma:

$$\overline{\underline{x}}_k = \phi_{k,k-1}\hat{\underline{x}}_{k-1} + \int_{t_{k-1}}^{t_k} \phi(t_k, \tau)B(\tau)\underline{u}(\tau)d\tau \tag{2.68}$$

$$\begin{aligned} \overline{P}_{Fk} &= \phi_{t_k,t_{k-1}}\hat{P}_F(t_{k-1})\phi^T_{t_k,t_{k-1}} + \\ &+ \int_{t_{k-1}}^{t_k} \phi_{t_k,\tau}G(\tau)Q_F(\tau))G(\tau)^T\phi(t_k,\tau)^T d\tau \end{aligned} \tag{2.69}$$

Em que:

- $\hat{\underline{x}}_0 = \underline{x}_0$ é a condição inicial utilizada para a propagação
- $\hat{P}_{F0} = P_{F0}$ é a covariância da condição inicial de $\hat{\underline{x}}_0$
- $\overline{\underline{x}}_k$ é a estimativa do estado $\underline{x}$ propagada para o instante t_k
- $\overline{P}_{Fk}$ representa a covariância do estado $\underline{x}$ propagada para o instante t_k
- $\phi_{k,k-1} = exp(\int_{t_{k-1}}^{t_k} A(\tau)d\tau)$ é a matriz de transição do estado do instante t_{k-1} ao estado t_k

- $Q_F(t)$ é a matriz de densidade espectral de potência do distúrbio $\underline{\omega}$, ou seja, $\underline{\omega} = N(\underline{0}, Q_F)$

As equações de atualização do estado são, $\forall k \in \{1, 2, ..., N\}$:

$$K_k = \overline{P}_{Fk} H_{Fk}^T \left(H_{Fk} \overline{P}_{Fk} H_{Fk}^T + R_{Fk} \right)^{-1} \quad (2.70)$$

$$\hat{P}_{Fk} = \left(I - K_k H_{Fk} \right) \overline{P}_{Fk} \quad (2.71)$$

$$\hat{x}_k = \underline{\overline{x}}_k + K_k \left(\underline{z}_k - H_{Fk} \underline{\overline{x}}_k \right) \quad (2.72)$$

Em que:

- R_{Fk} é a matriz de covariância dos erros das observações, ou seja, do ruído $\underline{v}_k$
- K_k é o ganho de *Kalman*
- H_{Fk} é a matriz que relaciona as medidas aos estados, no instante t_k
- I é a matriz Identidade
- O termo $(\underline{z}_k - H_{Fk} \underline{\overline{x}}_k)$ é chamado resíduo, ou seja, é a diferença entre a medida e o estado propagado

2.6.2 Filtro Estendido e Filtro Linearizado de Kalman

O Filtro de *Kalman* se aplica a casos de processos e medidas governados por equações lineares estocásticas. Para se aplicar a teoria a processos ou medidas não lineares estocásticas é utilizado o Filtro Estendido de *Kalman*, ou FEK (Welch e Bishop, 2001).

Utilizando princípios parecidos com os da série de *Taylor*, podemos linearizar as equações em torno da estimativa atual utilizando as derivadas parciais das funções do processo e da medida para computar as estimativas, ainda que em relações não lineares (Kuga, 2003).

O filtro estendido é uma abordagem diferente para se utilizar as mesmas equações lineares do filtro de *Kalman*. Neste caso, faz-se uma linearização da dinâmica em torno de uma trajetória de referência atualizada a cada processamento das medidas no instante correspondente (casos de modelos imprecisos ou simplificados). Uma abordagem parecida pode ser utilizada caso a trajetória de referência seja calculada de antemão (no caso de um modelo da dinâmica bastante preciso) ou trajetórias de referência. Este caso apresenta vantagens computacionais, pois as covariâncias e ganhos podem ser calculados *offline* e ser apenas consultados, deixando a CPU livre apenas para processar as medidas. É o caso mais adequado para situações de sistemas que operam em tempo real, embora nem sempre a dinâmica permita seu uso.

Assim, seja o sistema dinâmico governado pela equação $\forall t \in [t_0, t_f]$:

$$\dot{\underline{x}}(t) = \underline{f}(\underline{x}(t), \underline{u}(t), t) + G(t)\underline{\omega} \tag{2.73}$$

Com condições iniciais $\underline{x}(t_0) = \underline{x}_0$, $\underline{x}_0 = N(\underline{x}_0, P_{F0})$

Com medidas discretas dadas por, $\forall k \in \{0, 1, ..., N\}$:

$$\underline{z}_k = \underline{h}(\underline{x}_k) + \underline{v}_k \tag{2.74}$$

As equações de propagação, neste caso, são dadas por:

$$\dot{\overline{\underline{x}}}(t) = \underline{f}(\overline{\underline{x}}(t), \underline{u}(t), t) \tag{2.75}$$

$$\dot{\overline{P}}_F(t) = F(t)\overline{P}_F(t) + \overline{P}_F^T(t)F^T(t) + G(t)Q_F(t)G^T(t) \tag{2.76}$$

Em que:

- $\overline{\underline{x}} = \underline{x}_0$

- $\overline{P}_F(t_0) = P_{F0}$
- $F = \dfrac{\partial \underline{f}(\underline{x}(t), \underline{u}(t), t)}{\partial \underline{x}}\Big|_{\underline{x}(t)=\overline{\underline{x}}(t), \underline{u}(t)=\overline{\underline{u}}(t)}$

Ou, analogamente, podemos propagar a covariância de forma discreta. Neste caso, a Equação 2.76 se torna, $\forall k \in \{0, 1, ..., N\}$:

$$\overline{P}_{Fk} = \phi_{t_k, t_{k-1}} \hat{P}_F(t_{k-1}) \phi^T_{t_k, t_{k-1}} + \Gamma_k Q_F \Gamma^T_k \tag{2.77}$$

- $\phi_{k,k-1} = exp(\int_{t_{k-1}}^{t_k} A(\tau) d\tau)$ é a matriz de transição do estado do instante t_{k-1} ao estado t_k
- $\Gamma_k Q_F \Gamma^T_k = \int_{k-1}^{k} \phi_{\tau,k-1} G(\tau) Q_F(\tau) G^T(\tau) \phi^T_{\tau,k-1} d\tau$

2.7 Ambientes de Modelagem, Identificação e Simulação

Ören (2002) trata, superficialmente, do desenvolvimento da modelagem e simulação na ciência desde seus primórdios, definindo alguns dos requisitos essenciais para que a técnica seja confiável e utilizável. Davis (1998) também indica desenvolvimentos na área de simulação, embora com um enfoque diferente deste trabalho. Uma apresentação de ambientes de modelagem, identificação e simulação - MIS é feita por Linkens (1993), onde são mostradas várias ferramentas de *Computer-Aided Control System Design* (CAD) para sistemas de controle, incluindo requisitos e componentes de ambientes de MIS. Uma comparação de várias ferramentas de simulação é feita por Steinhaus (2002), onde o autor enfatiza principalmente a programação matemática, análise de dados e funcionalidades de simulação para um volume grande de dados. Trivelato e Souza (2001) fazem uma comparação inicial dos dois ambientes utilizados neste trabalho, MATLAB®/ Simulink e MATRIXx®/ SystemBuild para aplicações em modelagem e simulação de veículos aeroespaciais, porém, com uma visão mais geral do que a pretendida neste trabalho. Para o aprendizado destes ambientes foi utilizado o próprio material fornecido pelos fabricantes, The MathWorks (2005) e National Instruments, 2003. No caso do MATLAB®, ainda foi utilizado o material didático de Abdallah (2002), uma excelente referência para o aprendizado básico de MATLAB®.

A implementação destes algoritmos e as simulações dos sistemas estudados é feita numericamente em computadores, podendo ser feita em linguagem de programação procedural ou em ambientes de modelagem, identificação e simulação - MIS. Ambientes de MIS são ambientes para computação interativa (Linkens, 1993) que permitem que projetos de controle, que são por essência dependentes de computação numérica, sejam feitos mais rapidamente e de forma otimizada, além de proporcionar a possibilidade de uma programação visual.

Estes ambientes oferecem ferramentas mais poderosas para a modelagem e simulação de sistemas físicos com mais eficiência que a programação convencional e em um ambiente mais amigável e intuitivo, suprindo o esforço de modelagem e simulação numérica exigido. São utilizados na engenharia para projeto, simulação e implementação de sistemas de controle em várias aplicações. Seu uso melhora a atividade de projeto e acelera o desenvolvimento dos produtos utilizando lógica de controle em tempo real.

Ambientes de MIS, construídos em arquitetura orientada a objetos, foram evoluindo com o tempo, através da introdução de facilidades e funcionalidades, tanto para o momento da simulação, como para a análise posterior dos seus resultados, através de uma interface gráfica com o usuário (GUI) (The MathWorks, 2005, National Instruments, 2003).

Neste trabalho será feita a simulação de um problema de rastreamento utilizando dois ambientes integrados de MIS: o MATLAB®/ *Simulink* e o MATRIXx®/ *SystemBuild*, para análise dos ambientes e posterior comparação das funcionalidades e resultados apresentados por eles.

Estes ambientes de MIS, em particular os dois utilizados neste trabalho, foram desenvolvidos a partir de conceitos da álgebra linear como ferramenta para construção de suas funcionalidades, possuindo bibliotecas específicas que permitem uma computação de matrizes bastante eficiente (The MathWorks, 2005, National Instruments, 2003).

A escolha destes ambientes deve-se ao fato de um deles, o MATLAB®/ *Simulink*, ser bastante difundido no mundo todo, principalmente no meio acadêmico, enquanto que o outro, MATRIXx®/ *SystemBuild* é o ambiente padrão na indústria aeroespacial

norte - americana.

Além da própria estrutura dos ambientes, eles contam com módulos, como linguagem própria para computação técnica, módulo para simulação em ambiente gráfico, ferramentas para geração automática de código e documentação, módulos que interagem com outros *softwares* e com dispositivos de *hardware*, entre outros.

Estes módulos podem ser vistos nas Figuras 2.3 e 2.4, que apresentam as arquiteturas dos ambientes MATLAB®/ *Simulink* e MATRIXx®/ *SystemBuild*, respectivamente (The MathWorks, 2005; National Instruments, 2003):

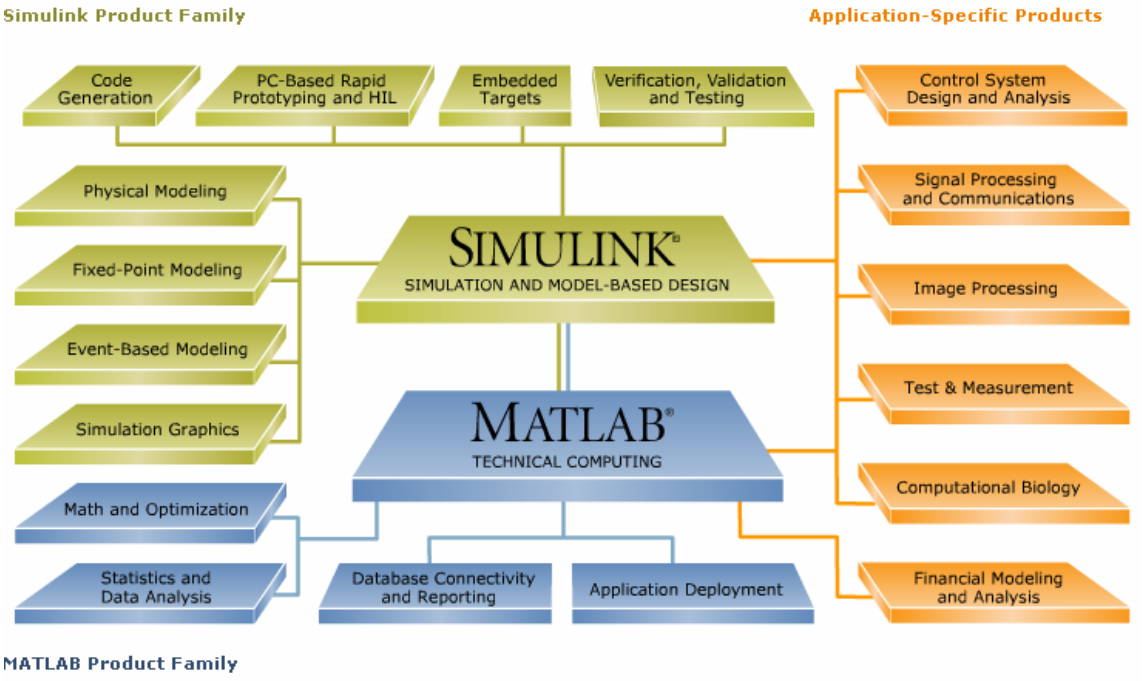


FIGURA 2.3 – Arquitetura da família MATLAB.
FONTE: The MathWorks (2005).

Entre suas aplicações típicas destacam-se Matemática e Computação, desenvolvimento de algoritmos, aquisição de dados, análise de dados, exploração e visualização, gráficos científicos e de engenharia, e outras ferramentas que juntas caracterizam uma modelagem e simulação de um sistema físico. Para isto, contam com bibliotecas de funções matemáticas, que são basicamente as coleções de algoritmos computacionais desde os mais simples (como somas e funções trigonométricas básicas) até funções mais complexas (como inversão de matrizes, autovalores, função de Bessel e FFT´s) (The MathWorks, 2005; National Instruments, 2003).

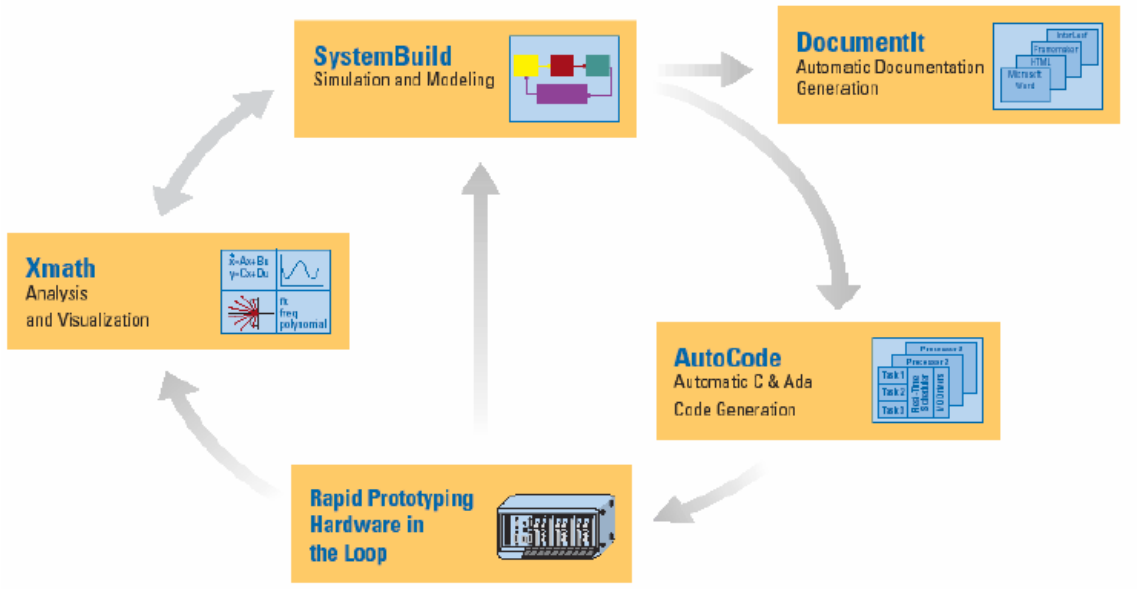


FIGURA 2.4 – Arquitetura da família MATRIXx.
FONTE: National Instruments (2003).

Estes ambientes são abertos e programáveis, suportando inserção de funções pelo usuário tanto em sua linguagem própria como em rotinas em FORTRAN, C e C++, permitindo a interação destas rotinas com o próprio ambiente.

Possuem em sua estrutura um simulador gráfico que permite que a descrição do problema seja feita graficamente, através de diagramas de blocos. Oferecem ferramentas para a realização de simulações de sistemas e análise do seu comportamento, além de proceder a avaliação do seu desempenho e realizar correções em projetos a partir de uma representação gráfica do sistema.

CAPÍTULO 3

METODOLOGIA

3.1 Formulação do Problema

Consideraremos neste trabalho um veículo rastreador $V1$ cuja trajetória cumpre três etapas bem definidas, a saber: 1) a fase de lançamento, quando ele se desprende do veículo lançador; 2) a fase de cruzeiro, onde $V1$ rastreia uma trajetória bem definida em direção a um veículo rastreado $V2$; e 3) a fase terminal, onde $V1$ rastreia o próprio $V2$, buscando um *rendez-vous* com ele. Aqui abordaremos a segunda fase, quando $V1$ não está mais sob ação das turbulências provocadas pelo veículo lançador, sua variação de massa é menos crítica, e sua trajetória de referência é definida por *way points*, e não pela trajetória de $V2$. Os chamados *way points* são pontos bem definidos, que indicam o caminho que deve ser seguido aproximadamente pelo veículo rastreador.

Os *way points* são definidos para que a trajetória na fase de cruzeiro se adapte aos acidentes geográficos locais ou atendam algum critério de otimalidade, tal como tempo de percurso ou consumo de combustível.

Durante a fase de cruzeiro, que é a fase de maior duração na trajetória do veículo, as estimativas da posição e velocidade de $V2$ são realizadas através da medida do ângulo da linha de visada, ainda que aquelas não sejam utilizadas para a lei de controle de $V1$.

Assim, esta lei de controle deverá ser projetada visando apenas o rastreamento da trajetória definida por pontos, segundo algum critério de otimalidade, que, neste caso de rastreamento, reflete a minimização do erro entre uma trajetória desejada e a trajetória descrita por $V1$ no espaço de estados.

Este problema, tanto na fase de cruzeiro quanto na fase terminal, envolve dois veículos: o veículo rastreador ($V1$) e o veículo medido ($V2$). $V1$ possui uma plataforma inercial que permite a medida de sua posição e velocidade inerciais, além de um autodiretor que mede o ângulo da linha de visada θ_v. Estas grandezas estão definidas na Figura 3.1. À medida que $V1$ mede indiretamente a posição de $V2$, ele

realiza as estimativas da posição e velocidade de $V2$ através do algoritmo do filtro de *Kalman*.

Caso $V1$ esteja impossibilitado de captar o sinal de $V2$, ele deve ser capaz de predizer a posição e a velocidade do centro de massa de $V2$ a partir dos dados de medidas anteriores (mesmo com incertezas e ruídos) ainda que não conheça a dinâmica de $V2$. O esquema do problema está mostrado na Figura 3.1:

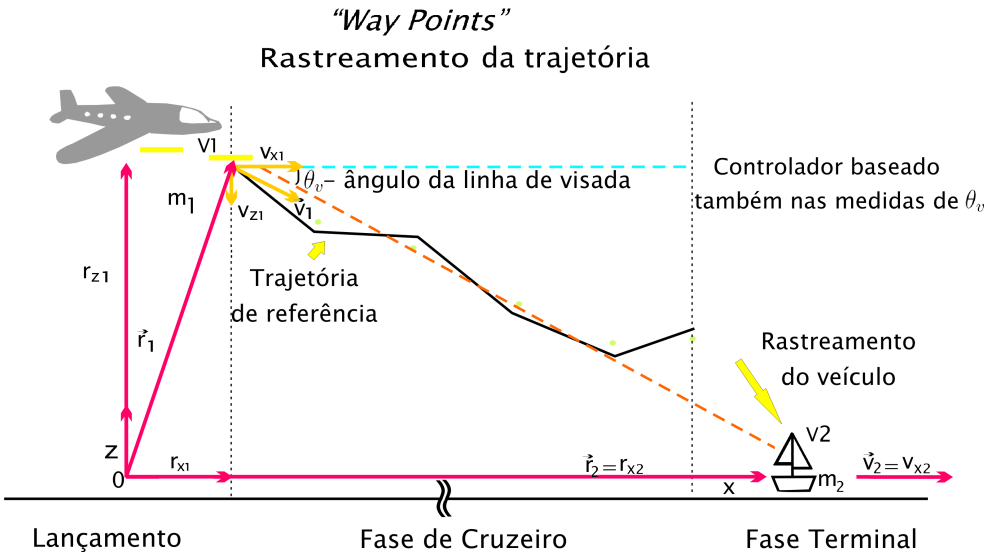


FIGURA 3.1 – Esquema do problema de rastreamento que definirá o controle do veículo $V1$.

3.2 Modelo Matemático do Problema

3.2.1 Dinâmica

Para a resolução do problema, assumem-se algumas hipóteses quanto à dinâmica dos veículos envolvidos: (i) os movimentos dos veículos se dão em um mesmo plano; (ii) o módulo da velocidade de $V2$ é muito menor que o módulo da velocidade de $V1$; (iii) a distância horizontal entre os veículos é tal que o referencial inercial pode ser fixado na Terra; (iv) as equações são dadas no referencial inercial. Os referenciais envolvidos no problema estão representados na Figura 3.2.

Assim, apresentam-se os critérios para se construir os modelos dos veículos $V1$ e $V2$ envolvidos no problema:

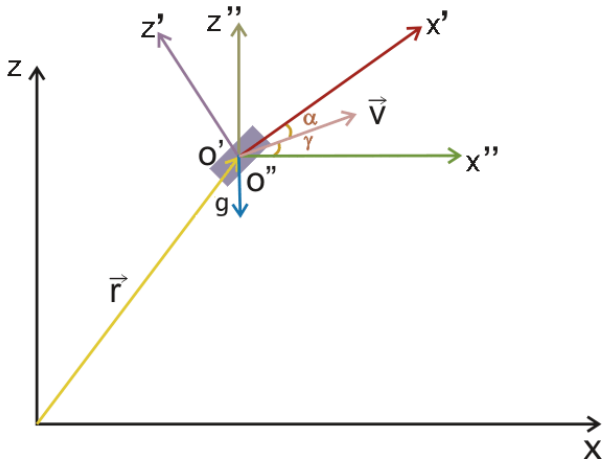


FIGURA 3.2 – Referenciais "inercial"e do corpo, fixo em $V1$.

Modelo de V1

$V1$ obedece a equações dinâmicas com diferentes graus de realismo, deduzidas a partir da Segunda Lei de *Newton*, sendo o vetor de estados de $V1$ dado por $\underline{x}_1(t)$ contendo termos que envolvem sua posição, $\vec{r}_1$ e velocidade, $\vec{v}_1$, anteriormente definidas na Figura 3.1 (seja diretamente por meios de suas componentes ou através de módulos e ângulos). Temos as equações gerais que descrevem sua evolução no tempo, $\forall t \in [t_0, t_f]$, com $t_0 = 0$, uma vez que o problema é invariante no tempo:

$$\dot{r}_{x1}(t) = v_{x1}(t) \tag{3.1}$$

$$\frac{d}{dt}(m_1(t)v_{x1}(t)) = \sum F_x(t) \tag{3.2}$$

$$\dot{r}_{z1}(t) = v_{z1}(t) \tag{3.3}$$

$$\frac{d}{dt}(m_1(t)v_{z1}(t)) = \sum F_z(t) \tag{3.4}$$

Em que $\sum F_x(t)$ e $\sum F_z(t)$ são os somatórios de forças nas direções x e z, respectivamente, considerando termos como força da gravidade, força de empuxo, controle, entre outras. A forma como estas variáveis serão definidas, e quais termos de força serão considerados, será diferente para cada um dos três modelos apresentados neste trabalho.

O veículo $V1$ entra na fase de cruzeiro com suas condições iniciais conhecidas, dadas por $\underline{x_1}(t_0) = \underline{x}_{1_0}$.

Enquanto $V1$ mede sua posição e velocidade, ele realiza medidas do ângulo da linha de visada θ_v.

As equações levam em conta propriedades físicas do próprio veículo $V1$ como área representativa S, diâmetro representativo d, massa m_1, entre outros parâmetros que possam influenciar sua dinâmica.

Modelo de V2

Pela própria natureza do problema, a dinâmica e a cinemática de $V2$ são desconhecidas. Portanto, na impossibilidade de modelá-lo por equações dinâmicas, sua modelagem será feita através de leis da Cinemática. Esta modelagem será feita com aceleração nula, havendo movimento retilíneo uniforme apenas na horizontal. Assim:

$$r_{x_2}(t) = r_{x_20} + v_{x_2}(t - t_0) \tag{3.5}$$
$$r_{z_2}(t) = 0 \tag{3.6}$$
$$v_{x_2}(t) = v_{x_20} \tag{3.7}$$
$$v_{z_2}(t) = 0 \tag{3.8}$$

Sendo que suas condições iniciais serão medidas indiretamente por $V1$, através do ângulo θ_v.

3.2.2 Ambiente

Atmosfera

Para a simulação do problema, é necessário considerar também o ambiente no qual os veículos se deslocam. Assim, os modelos do ambiente consideram:

- Densidade do ar: O modelo de densidade do ar ρ aqui considerado é seu decaimento exponencial em função da altura r_z do veículo (National Aeronautics and Space Administration, 1966). Assim,

$$\rho(r_z) = \rho_0 \exp\left(-\frac{r_z}{H_0}\right) \tag{3.9}$$

  Onde $\rho_0 = 1,752 kg/m^3$ e $H_0 = 6700m$.

- Velocidade do som no ar: O modelo de velocidade do som foi extrapolado segundo a Tabela 3.1 (National Aeronautics and Space Administration, 1966). É representado por um decaimento aproximadamente linear de $1m/s$ a cada $250m$.

TABELA 3.2 – Valores da velocidade do som para diferentes valores de altitude.

Altura (m)	v_{som} (m/s)	Altura (m)	v_{som} (m/s)	Altura (m)	v_{som} (m/s)
0	348.7	3500	336.0	7000	321.4
250	347.8	3750	334.9	7250	320.4
500	346.8	4000	333.9	7500	319.3
750	345.8	4250	332.9	7750	318.3
1000	344.8	4500	331.9	8000	317.2
1250	343.9	4750	330.8	8250	316.1
1500	342.9	5000	329.8	8500	315.1
1750	342.0	5250	328.8	8750	314.0
2000	341.0	5500	327.7	9000	312.9
2250	340.1	5750	326.7	9250	311.9
2500	340.1	6000	325.6	9500	310.8
2750	339.1	6250	324.6	9750	309.7
3000	338.0	6500	323.5	10000	308.6
3250	337.0	6750	322.5		

FONTE: National Aeronautics and Space Administration (1966)

- Pressão atmosférica: A influência da pressão atmosférica se dá, neste problema, principalmente nos coeficientes aerodinâmicos considerados. Assim, tendo os valores destes retirados da literatura, não é necessário considerar um modelo explícito para a pressão atmosférica.

Aceleração da Gravidade

Dadas as baixas alturas nas quais $V1$ se move, é suficiente modelar a aceleração da gravidade, g, como sendo constante, com o valor $9,8m/s^2$.

3.2.3 Linearização do Modelo

Considerando que nos modelos mais realistas $\underline{f}_1(\underline{x}(t), \underline{u}(t), t)$ não é linear, e considerando que grande parte da Teoria de Controle desenvolvida até nossos dias é baseada em plantas lineares, faz-se necessário se aproximar a planta não-linear (e muitas vezes variante no tempo) por uma linearizada, na tentativa de adequação do modelo à teoria. Assim, escolhendo instantes de tempo t_{lin} teremos, para cada momento, uma planta linearizada.

Seja $\underline{r}(t)$ uma linha que liga os *way points* por retas, e $\underline{x}_{1r}(t)$ a trajetória factível pelo veículo mais próxima possível de $\underline{r}(t)$. Para que o veículo seguisse uma trajetória real $\underline{x}_1(t)$ próxima de $\underline{x}_{1r}(t)$, ou seja, para que $\underline{x}_1(t) \approx \underline{x}_{1r}(t)$, teríamos a dinâmica da planta descrita por:

$$\dot{\underline{x}}_{1r}(t) = \underline{f}_1\left(\underline{x}_{1r}(t), \underline{u}_r(t), t\right) \tag{3.10}$$

Com condições iniciais dadas por $\underline{x}_{1r}(0) = \underline{x}_{1r_0}$.

Supondo que a trajetória real possa ser considerada como a soma de duas parcelas, sendo a primeira parcela constituída pela trajetória de referência e a segunda parcela sendo uma perturbação em torno desta trajetória, assumimos a trajetória real (perturbada) como, $\forall t \in [t_0, t_f]$:

$$\dot{\underline{x}}_1(t) = \underline{f}_1\left(\underline{x}_1(t), \underline{u}(t), t\right) \tag{3.11}$$

Com condições iniciais dadas por $\underline{x}_1(0) = \underline{x}_{1_0}$.

Temos que as diferenças entre estas duas trajetórias e seus controles são dadas por, $\forall t \in [t_0, t_f]$:

$$\Delta\underline{x}_1(t) \triangleq \underline{x}_1(t) - \underline{x}_{1r}(t) \tag{3.12}$$

$$\Delta\underline{u}(t) \triangleq \underline{u}(t) - \underline{u}_r(t) \tag{3.13}$$

Analogamente, a condição inicial desta nova variável $\Delta\underline{x}_1(t)$ é dada por $\Delta\underline{x}_1(0) \triangleq \underline{x}_{1_0} - \underline{x}_{1r_0}$.

Assim:

$$\Delta\dot{\underline{x}}_1(t) = \dot{\underline{x}}_1(t) - \dot{\underline{x}}_{1r}(t) \tag{3.14}$$

$$= \underline{f}_1\left(\underline{x}_1(t), \underline{u}(t), t\right) - \underline{f}_1\left(\underline{x}_{1r}(t), \underline{u}_r(t), t\right) \tag{3.15}$$

Se $\underline{f}_1\left(\underline{x}_1(t), \underline{u}(t), t\right) \in \mathbb{C}^2$, então podemos expandi-la em série de *Taylor* e escrever $\forall t \in [t_0, t_f]$:

$$\Delta\dot{\underline{x}}_1 = F_x(\underline{x}_{1r}(t), \underline{u}_r(t), t)\Delta\underline{x}(t) + F_u(\underline{x}_{1r}(t), \underline{u}_r(t), t)\Delta\underline{u} + O^2 \tag{3.16}$$

Onde:

- $F_x(\underline{x}_{1r}(t), \underline{u}_r(t), t) = \left.\dfrac{\partial \underline{f}_1\left(\underline{x}\left(t\right), \underline{u}\left(t\right), t\right)}{\partial \underline{x}}\right|_{\underline{x}(t)=\underline{x}_{1r}(t), \underline{u}(t)=\underline{u}_r(t)}$
- $F_u(\underline{x}_{1r}(t), \underline{u}_r(t), t) = \left.\dfrac{\partial \underline{f}_1\left(\underline{x}\left(t\right), \underline{u}\left(t\right), t\right)}{\partial \underline{u}}\right|_{\underline{x}(t)=\underline{x}_{1r}(t), \underline{u}(t)=\underline{u}_r(t)}$

A partir deste ponto, esta linearização será feita em torno de pontos previamente selecionados da linha dos *way points* $\underline{r}(t)$, considerando um controle que seria $\underline{u}_{wp}(t)$

pois, assume-se que o controle se encarrega de levar o veículo para a trajetória desejada, ou seja, consideramos que o veículo encontra-se perto desta região, isto é, $\underline{x}_1(t) \approx \underline{x}_{1r}(t) \approx \underline{r}(t)$ e $\underline{u}(t) \approx \underline{u}_r(t)$.

3.2.4 Agendamento de Ganhos

O problema tratado nos modelos 2 e 3 tem sua dinâmica variante com as condições de vôo, em razão das não linearidades presentes no processo. Porém, é possível mudar os parâmetros do controlador monitorando as condições de operação do veículo a ser controlado. O nome desta técnica é agendamento de ganhos (*"gain scheduling"*). Esta técnica utiliza um *feedback* especial: é um regulador cujos parâmetros dependem das condições de vôo do veículo. Esta técnica representa uma boa saída para compensar as variações nos parâmetros do sistema, assim como as não-linearidades presentes. Seu esquema é mostrado na Figura 3.3.

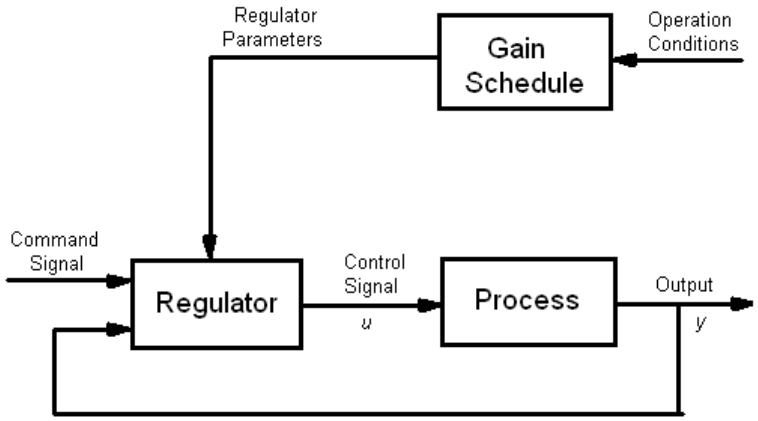


FIGURA 3.3 – Diagrama de blocos de um sistema cujas influências nos parâmetros são reduzidas através do agendamento de ganhos.
FONTE: Astrom e Wittenmark (1989).

No caso aqui apresentado, o parâmetro que fará o agendamento dos ganhos é o tempo, e as condições de operação utilizadas nos cálculos dos ganhos são dados da trajetória de referência, que deve ser seguida aproximadamente pelo veículo.

Um dos inconvenientes desta técnica é que a compensação por agendamento de ganhos trabalha em malha aberta, ou seja, não há *feedback* para compensar um erro no agendamento (Astrom e Wittenmark, 1989), além de se mostrar adequada em

casos onde a variação dos parâmetros não é muito rápida (Shamma e Athans, 1992). Porém, há a vantagem da possibilidade de uma mudança rápida nos parâmetros do regulador.

A idéia principal da técnica se divide em duas partes: primeira, projetar os controladores baseados em linearizações obtidas para diversas condições de vôo, e segunda, agendar estes ganhos para todas as condições de vôo possíveis (Shamma e Athans, 1992). Neste caso, não faremos a interpolação dos ganhos para condições intermediárias de vôo, e sim consideraremos o sistema constante por intervalos de tempo ($5s$ para o Modelo 2 e $2,5s$ para o Modelo 3). Esta técnica de considerar o sistema constante em um intervalo de tempo se chama congelamento de pólos.

3.2.5 Controlador

LQR

A partir dos valores dos estados de $V1$, será feito o projeto de um controlador que permita o rastreamento da trajetória a ser seguida. A teoria aqui utilizada, do Regulador Linear Quadrático, ou LQR, conforme indicado pelo próprio nome, se aplica a plantas lineares. Para isto, tomam-se as equações de estado do problema na forma linearizada, conforme equação 3.16, onde $\forall t \in [t_0, t_f]$:

$$\Delta\dot{\underline{x}}_1(t) = A(t)\Delta\underline{x}_1(t) + B(t)\Delta\underline{u}(t) \tag{3.17}$$

Em que $A(t) = F_x(\underline{r}(t), \underline{u}_{wp}(t), t)$ e $B(t) = F_u(\underline{r}(t), \underline{u}_{wp}(t), t)$, com $\Delta\underline{x}_1(t)$ e $\Delta\underline{u}(t)$ definidos pelas Equações 3.12 e 3.13 condições iniciais dadas por:

$$\Delta\underline{x}_1(0) = \Delta\underline{x}_{1_0} = \underline{x}_{10} - \underline{r}_0 \tag{3.18}$$

Considera-se, ainda, o índice de desempenho a ser minimizado, dado pela Equação 2.55:

$$J = \frac{1}{2}\|\Delta\underline{x}_1(t_f)\|_{H_R} + \frac{1}{2}\int_0^{t_f} \{\|\Delta\underline{x}_1(t)\|_{Q_R} + \|\Delta\underline{u}(t)\|_{R_R}\}\, dt \tag{3.19}$$

Onde:

- H_R é a matriz de pesos dos estados finais, sendo real, simétrica, positiva semi-definida
- Q_R é a matriz de pesos dos estados, sendo real, simétrica, positiva semidefinida, podendo também ser variante no tempo
- R_R é a matriz de pesos das variáveis de controle, sendo real, simétrica, positiva definida, podendo também ser variante no tempo

O sobrescrito $(^T)$ indica a transposição da matriz ou vetor.

A partir da otimização do funcional J determina-se a lei de controle ótima, $\Delta\underline{u}^*(t)$, que fará com que $V1$ siga a trajetória de referência com o menor erro possível, pelas Equações 2.45 a 2.54 . Esta lei de controle será aplicada na planta em conjunto com um controle de referência (em torno do qual a planta foi linearizada), com realimentação negativa, isto é, $\underline{u}(t) = \underline{u}_r(t) - \Delta\underline{u}(t)$, pois pelas Equações 3.12 e 3.13:

$$\underline{x}_1(t) = \underline{x}_{1r}(t) + \Delta\underline{x}_1(t) \tag{3.20}$$
$$\underline{u}(t) = \underline{u}_{wp}(t) + \Delta\underline{u}(t) \tag{3.21}$$

Porém, como $\underline{x}_{1r}$ e $\underline{u}_{wp}$ não são conhecidos, usaremos:

$$\underline{x}_1(t) = \underline{r}(t) + \Delta\underline{x}_1(t) \tag{3.22}$$
$$\underline{u}(t) = \underline{u}_r(t) + \Delta\underline{u}(t) \tag{3.23}$$

3.2.6 Estimador

Como as medidas do estado dos veículos não são exatas, e nem sempre a propagação usada para projetar o controle é fiel, sabemos que os valores dos estados fornecidos pelos sensores ou preditos pela propagação, que serviriam de entrada para o controlador não correspondem aos valores mais adequados. Para se extrair da propagação e das medidas informações mais confiáveis a respeito do verdadeiro estado da planta, faz-se uso de um estimador. Neste trabalho, o estimador utilizado é o Filtro de *Kalman*.

As variáveis de estado que se desejam estimar são $\begin{bmatrix} r_{x1} & r_{z1} & v_{x1} & v_{z1} & r_{x2} & v_{x2} \end{bmatrix}^T$, ou, em outra forma, $\begin{bmatrix} r_{x1} & r_{z1} & v_1 & \gamma & r_{x2} & v_{x2} \end{bmatrix}^T$, onde v_1 é o módulo da velocidade de $V1$ e γ é o ângulo que o vetor velocidade de $V1$ faz com a horizontal local. Na construção do modelo da planta e dos sensores, considera-se que todas as medidas, ou seja, r_{x1}, r_{z1}, v_{x1}, v_{z1} e θ_v estão sujeitas a ruídos dos sensores modelados por processos ou seqüências estocásticas brancas gaussianas de médias nulas e covariâncias conhecidas.

Uma visão geral do sistema (planta, sistema de controle e estimador) é mostrada na Figura 3.4.

As estimativas do estado do veículo medido serão feitas a partir de todas as medidas disponíveis da posição e da velocidade de $V1$ e do ângulo θ_v que este faz com a direção de $V2$. Caso se perdesse a habilidade de obtenção da medida do ângulo, o procedimento utilizaria sua capacidade de predição para estimar suas posições futuras.

Serão escolhidos para uso no Filtro valores para a matriz de covariância do estado inicial P_{F_0}, e para a matriz de covariância das perturbações na dinâmica, Q_F. O valor da matriz de covariância dos ruídos das observações, R_F é determinado pelas incertezas dos sensores. As medidas de posição e velocidade de $V1$ e as medidas de θ_v serão simuladas acrescidas de ruídos, para representar as incertezas envolvidas no processo de medição. A partir destes elementos, o filtro de *Kalman* deve propagar o estado de $V1$ e $V2$, além de suas respectivas covariâncias.

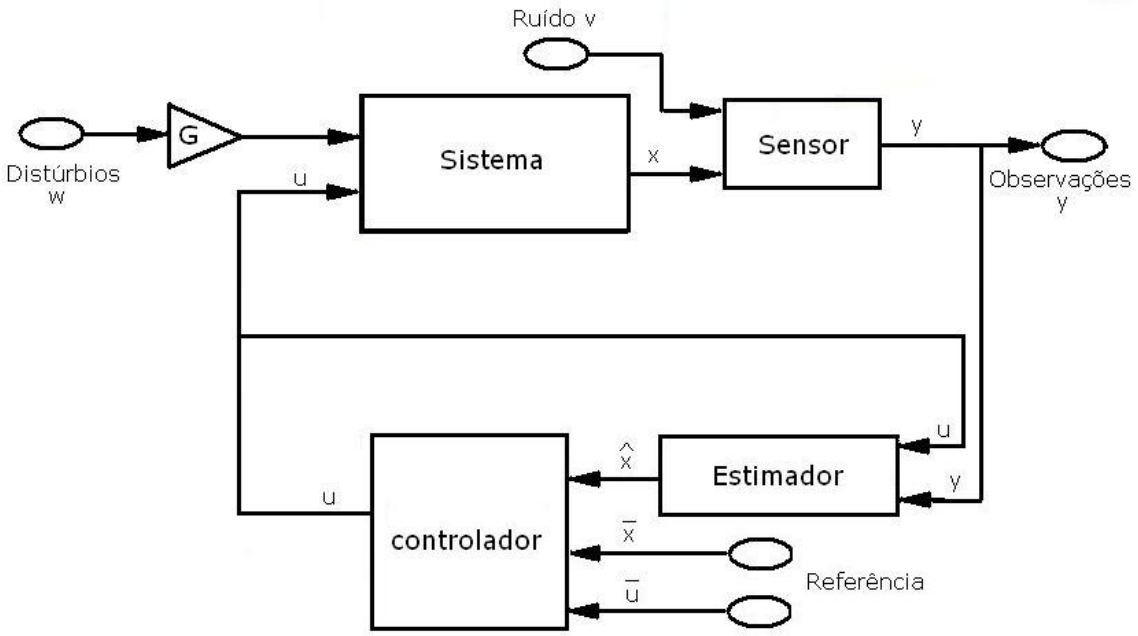


FIGURA 3.4 – Visão geral do sistema composto pelo veículo e seu sistema de controle.

Havendo mais medidas de θ_v, o filtro atualiza as medidas de $V2$. No caso de não haver mais medidas de θ_v, o filtro realizaria apenas uma propagação do estado. Depois de o filtro de *Kalman* estimar o estado de $V2$, os valores estimados serão comparados com os reais, simulados anteriormente. Caso a diferença seja menor que uma tolerância escolhida, aceita-se o desempenho do filtro. Caso contrário, escolhem-se outros valores de P_{k_0} e Q_k, a fim de melhorar os resultados obtidos.

3.2.7 Trajetória de Referência

Consideramos, neste problema, um veículo $V1$ que rastreia uma trajetória de referência $\underline{x}_{1r}(t)$ próxima à linha dos *way points* $\underline{r}(t)$, em direção ao veículo $V2$ com o qual se aspira, na fase terminal, realizar um *rendez-vous*. Foge do escopo deste trabalho a definição desta trajetória $\underline{x}_{1r}(t)$ ou mesmo da linha dos *way points*, o que, por si só já poderia ser formulada como um problema de controle ótimo (considerando, por exemplo, tempo de percurso ou economia de combustível como parâmetros para índices de desempenho).

Os chamados *way points* são pontos pelos quais, teoricamente, o veículo $V1$ deveria

passar. Na prática, admite-se que o veículo passe nos arredores dos pontos. Consideraremos, neste modelo, que a trajetória ótima $\underline{x}_{1r}(t)$ é uma trajetória admissível, passando próximo da linha que liga os *way points* $\underline{r}(t)$ de modo a permitir-nos usar $\underline{r}(t)$ em lugar de $\underline{x}_{1r}(t)$, por este não ser conhecido.

Para simplificar o procedimento, consideramos neste trabalho a linha dos *way points* constituída por linhas retas $\underline{r}(t)$. Isto porque $\underline{r}(t)$, que é uma trajetória não apenas na posição, mas em todo o espaço de estados, não é uma trajetória executável na prática, uma vez que as próprias leis físicas limitam uma mudança instantânea de velocidade. Porém, deseja-se passar próximo a esta trajetória, cumprindo assim um trajeto admissível fisicamente.

3.3 Simulação do Problema e Comparação das Implementações

Depois de projetado o controlador, constituído pelo estimador em conjunto com a lei de controle, os veículos serão simulados em ambos os ambientes de MIS, MATLAB®/ *Simulink* e MATRIXx®/ *SystemBuild*. Estas simulações devem incluir não somente o controlador, mas a dinâmica/ cinemática dos veículos, coeficientes aerodinâmicos envolvidos no movimento, além do ambiente no qual os veículos estão inseridos.

Depois de prontas as simulações, seus resultados serão analisados, e devem mostrar o mesmo desempenho nos dois ambientes. A forma de apresentar estes resultados será analisada, tanto na sua forma numérica quando na sua forma gráfica. Depois se procederá uma análise das funcionalidades oferecidas por cada um dos ambientes para permitir a construção do modelo e sua simulação.

CAPÍTULO 4

SIMULAÇÕES E RESULTADOS

Apresentam-se neste Capítulo os resultados da simulação do veículo aeroespacial, assim como sua lei de controle e seu estimador. Foram simulados três modelos distintos.

O primeiro modelo é bastante simples, sem controle. No entanto, conta com o estimador. Este modelo foi construído para desenvolver um primeiro contato com as ferramentas de simulação, assim como construir o primeiro filtro de *Kalman* utilizado no trabalho.

O segundo modelo se apresenta mais realista que o primeiro, apresentando sistema de controle, além do controlador, e também sofrendo influências do arrasto atmosférico, além de contar com um motor que fornece a força de empuxo. Seu controle, no entanto, não é executável na prática, uma vez que os sinais de controle estão representados no sistema inercial.

Um maior grau de realismo se apresenta no terceiro modelo, que conta com as mesmas características do modelo 2, considerando porém um controle factível. Considera também uma força de empuxo variável no tempo, massa variável, entre outras características mais realistas, representando assim melhor o veículo aeroespacial.

4.1 Modelo 1

4.1.1 Características do Modelo

Uma primeira abordagem do problema apresentado anteriormente foi feita enfatizando somente a estimação de estados, com um modelo bastante simplificado do veículo, que opera sob efeito apenas da ação da gravidade.

Foram considerados veículos puntuais, portanto sem movimento rotacional, e com área representativa desprezível. A atmosfera, portanto, não altera o movimento do veículo. Assim, os valores da densidade do ar e velocidade do som no ar foram desprezados. No entanto, há influência da ação da gravidade, e, uma vez que $V1$,

está sem controle, cai em conseqüência do seu efeito.

Uma vez que a planta é bastante simplificada, é representada de forma linear, de forma que não é necessário proceder a linearização do modelo.

As variáveis de estado a serem estimados são as posições e velocidades de $V1$ e $V2$, ou seja, $\underline{x} = \begin{bmatrix} r_{x_1} & r_{z_1} & v_{x_1} & v_{z_1} & r_{x_2} & v_{x_2} \end{bmatrix}^T$. As medidas disponíveis são de $\underline{z}_k = \begin{bmatrix} r_{x_1} & r_{z_1} & v_{x_1} & v_{z_1} & \theta_v \end{bmatrix}^T$. Todas estas medidas estão sujeitas a ruídos modelados por processos ou seqüências estocásticas brancas gaussianas de médias nulas e covariâncias conhecidas.

Foram feitas as simulações do problema nos ambientes de MIS MATLAB® e MATRIXx®, assim como a implementação do Filtro de *Kalman*.

4.1.2 Equações

Representa-se a dinâmica do sistema que, nesta abordagem, é bastante simplificada, linear com medidas não lineares, da forma, $\forall t \in [t_0, t_f]$:

$$\begin{aligned} \dot{\underline{x}}(t) &= A\underline{x}(t) + B\underline{u}(t) + G\underline{\omega}(t) && (4.1) \\ \underline{z}(t) &= \underline{z}(\underline{x}(t), t) + \underline{v}(t) && (4.2) \end{aligned}$$

Onde $\underline{x}(t)$ representa as variáveis de estado do sistema. Observa-se que a planta sofre ação de distúrbios $\underline{\omega}(t)$. Consideramos, também, que há uma incerteza de informação na posição inicial $\underline{x}_0$, e os sensores do veículo apresentam ruído, dado por $\underline{v}(t)$. As estatísticas da incerteza da posição inicial, dos distúrbios na planta e dos ruídos dos sensores são dadas por:

$$\begin{aligned} \underline{x}(0) &= N(\underline{x}_0, P_{F0}) \\ \underline{\omega}(t) &= N(\underline{0}, Q_F) \\ \underline{v}(t) &= N(\underline{0}, R_F) \end{aligned} \qquad (4.3)$$

Explicitamente, para $V1$:

$$\begin{aligned} \dot{r}_{x1}(t) &= v_{x1}(t) \\ \dot{r}_{z1}(t) &= v_{z1}(t) \\ \dot{v}_{x1}(t) &= 0 + \omega_{vx}(t) \\ \dot{v}_{z1}(t) &= g + \omega_{vz}(t) \end{aligned} \tag{4.4}$$

A dinâmica do veículo rastreado é dada pelas equações 3.8, aqui reproduzidas:

$$r_{x_2}(t) = r_{x_{20}} + v_{x_2}(t - t_0) \tag{4.5}$$

$$r_{z_2}(t) = 0 \tag{4.6}$$

$$v_{x_2}(t) = v_{x_{20}} \tag{4.7}$$

$$v_{z_2}(t) = 0 \tag{4.8}$$

Em que $r_{x_{20}}$ e v_{x_20} são condições iniciais, supostamente conhecidas. Assim, as matrizes A, B e G da dinâmica são dadas por:

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{4.9}$$

$$B = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}^T \tag{4.10}$$

$$G = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}^T \tag{4.11}$$

Com a aceleração da gravidade representada pelo "controle" u_z.

As medidas disponíveis são aquelas feitas pela plataforma inercial embarcada em $V1$ e pelo auto diretor, ou seja, medidas das componentes da posição e velocidade de $V1$ e do ângulo da linha de visada θ_v. Nesta abordagem, consideramos o ângulo θ_v aproximado pelo ângulo entre o vetor entre as posições de $V1$ e $V2$ e a horizontal. Assim, temos as equações de medidas:

$$\underline{z} = \begin{bmatrix} r_{x_1} \\ r_{z_1} \\ v_{x_1} \\ v_{z_1} \\ \theta_v \end{bmatrix} = \begin{bmatrix} r_{x_1} \\ r_{z_1} \\ v_{x_1} \\ v_{z_1} \\ \arctan\left(\dfrac{r_{z_1}}{r_{x_2} - r_{x_1}}\right) \end{bmatrix} + \underline{v}(t) \tag{4.12}$$

Portanto, o vetor $\underline{z}_k$ é representado pela linearização do vetor de medidas $\underline{z}(t)$ em relação aos estados, tomando como referência a estimativa imediatamente anterior ao instante da observação:

$$\underline{z_k} = \left.\frac{\partial \underline{z}}{\partial \underline{x}}\right|_{\underline{x}=\underline{x}_{k-1}} \tag{4.13}$$

4.1.3 Controle

Este veículo, como já dito anteriormente, opera sem controle, sofrendo apenas influência da ação da gravidade. Esta é modelada pelo "controle constante" $u_z = g$

4.1.4 Filtro de Kalman

Na fase de propagação do estado, utilizaram-se as equações de movimento dos veículos, considerando uma dinâmica contínua, com medidas discretas em $t_k = kT + t_0$, com $t_0 = 0$, $\forall k \in \{1, 2, ...N\}$, onde $N = \dfrac{t_f - t_0}{T}$.

Como a matriz A é contínua, podemos realizar a propagação do filtro de *Kalman* através da matriz de transição ϕ (Kuga, 2003) $\forall k \in \{1, 2, ...N\}$:

$$\overline{\underline{x}}_k = \phi_{k,k-1}\hat{\underline{x}}_{k-1} + \Gamma_B \underline{u}_{k-1} \tag{4.14}$$

$$\overline{P}_{Fk} = \phi_{k,k-1}\hat{P}_{Fk-1}\phi^T_{k,k-1} + \Gamma_{Gk} Q_F \Gamma^T_{Gk} \tag{4.15}$$

$$\overline{\underline{z}}_k = H_{Fk}\overline{\underline{x}}_k \tag{4.16}$$

Onde $\hat{\underline{x}}_0 = \underline{x}_0$, $\hat{P}_{F0} = P_{F0}$ a barra $(^-)$ representa a propagação para o instante k, enquanto H_{Fk} é a matriz que relaciona as observações aos estados no instante t_k, dada por:

$$H_{Fk} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ \dfrac{r_{y1}}{(r_{x2}-r_{x1})^2 + r_{y1}^2} & \dfrac{r_{x1}-r_{x2}}{(r_{x2}-r_{x1})^2 + r_{y1}^2} & 0 & 0 & -\dfrac{r_{y1}}{(r_{x2}-r_{x1})^2 + r_{y1}^2} & 0 \end{bmatrix} \tag{4.17}$$

Os valores de ϕ e Γ_i são dados por:

$$\phi = exp(AT) = \begin{bmatrix} 1 & 0 & T & 0 & 0 & 0 \\ 0 & 1 & 0 & T & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & T \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.18}$$

$$\Gamma_i = \int_0^T \exp(A\tau) i d\tau \tag{4.19}$$

Onde i é B ou G. Assim:

$$\Gamma_B = \begin{bmatrix} 0 & \frac{T^2}{2} & 0 & T & 0 & 0 \end{bmatrix} \tag{4.20}$$

$$\Gamma_G = \begin{bmatrix} 0 & 0 & 0 & T & 0 & 0 \\ 0 & 0 & T & 0 & 0 & 0 \end{bmatrix} \tag{4.21}$$

A atualização das estimativas feitas a partir das medidas é feita através das equações:

$$K_k = \overline{P}_{Fk} H_{Fk}^T \left(H_{Fk} \overline{P}_{Fk} H_{Fk}^T + R_{Fk} \right)^{-1} \tag{4.22}$$

$$\hat{P}_{Fk} = \left(I - K_k H_{Fk} \right) \overline{P}_{Fk} \tag{4.23}$$

$$\underline{\hat{x}}_k = \underline{\overline{x}}_k + K_k \left(z_k - H_{Fk} \underline{\overline{x}}_k \right) \tag{4.24}$$

Consideramos incertezas em todas os estados da propagação, que são caracterizadas através da matriz de incertezas da dinâmica, $\Gamma_k Q_F \Gamma_k^T$, que teve seu valor inicial ajustado para melhor desempenho do filtro.

As seguintes condições iniciais foram utilizadas:

- Estado inicial (posições e velocidades de $V1$, posição e velocidade horizontais de $V2$): $\underline{x}_0 = \begin{bmatrix} r_{x1_0} & r_{z1_0} & v_{x1_0} & v_{z1_0} & r_{x2_0} & v_{x2_0} \end{bmatrix}^T = \begin{bmatrix} 0 & 5000 & 150 & 5 & 8000 & 10 \end{bmatrix}^T$; em unidades do S.I.;
- Erro inicial na estimativa do estado: $\Delta \underline{x} = \underline{\hat{x}}(0) - \underline{x}(0) = \begin{bmatrix} 10 & 10 & 3 & 3 & 50 & 10 \end{bmatrix}^T$, em unidades do S.I.;
- Valor inicial da matriz de covariância do estado $\underline{x}$: $P_{Fk}(0) = P_{F0} = diag \begin{bmatrix} 5^2 & 5^2 & 2^2 & 3^2 & 10^2 & 5^2 \end{bmatrix}$
- Valor da matriz de densidade espectral de potência do distúrbio $\underline{\omega}(t)$ (processo branco): $\Gamma_k Q_F \Gamma_k^T = diag \begin{bmatrix} 0 & 0 & 0 & 1^2 & 0 & 0 \end{bmatrix}^T$ distúrbios agindo diretamente na aceleração do veículo $V1$)

- Valor da matriz de covariância do ruído $\underline{v}(t)$: $R_F = diag\left[\begin{array}{ccccc} 5^2 & 5^2 & 2^2 & 3^2 & (2e-3)^2 \end{array}\right]$

Estes valores iniciais foram escolhidos aleatoriamente. Simularam-se as posições e velocidades de $V1$ e $V2$ exatas (estado real) para uma comparação com os valores estimados pelo filtro, para verificar qual a sua influência no rastreio. Lembra-se que neste caso não há controle, e não consideramos nenhuma força de empuxo.

4.1.5 Simulações - MATLAB

Os resultados apresentados a seguir foram obtidos com $60s$ de simulação no ambiente de MIS MATLAB®, sendo os resíduos das medidas apresentados na Figura 4.1.

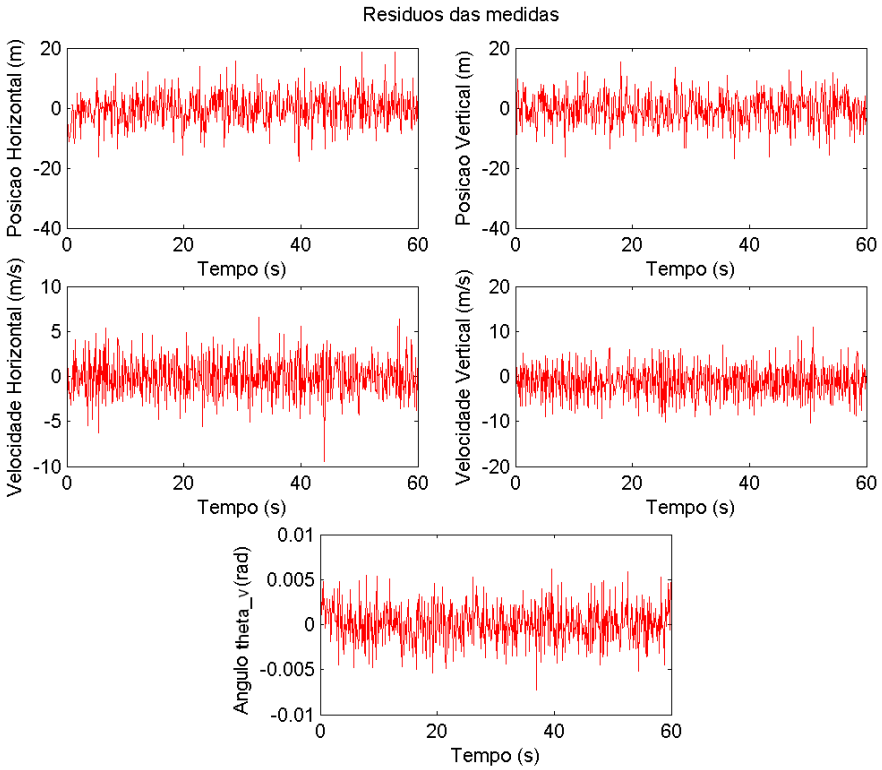


FIGURA 4.1 – Resíduos das medidas de posição e velocidade de V1 e θ_v em MATLAB.

O desempenho do filtro é mostrado nas Figuras 4.2 e 4.3.

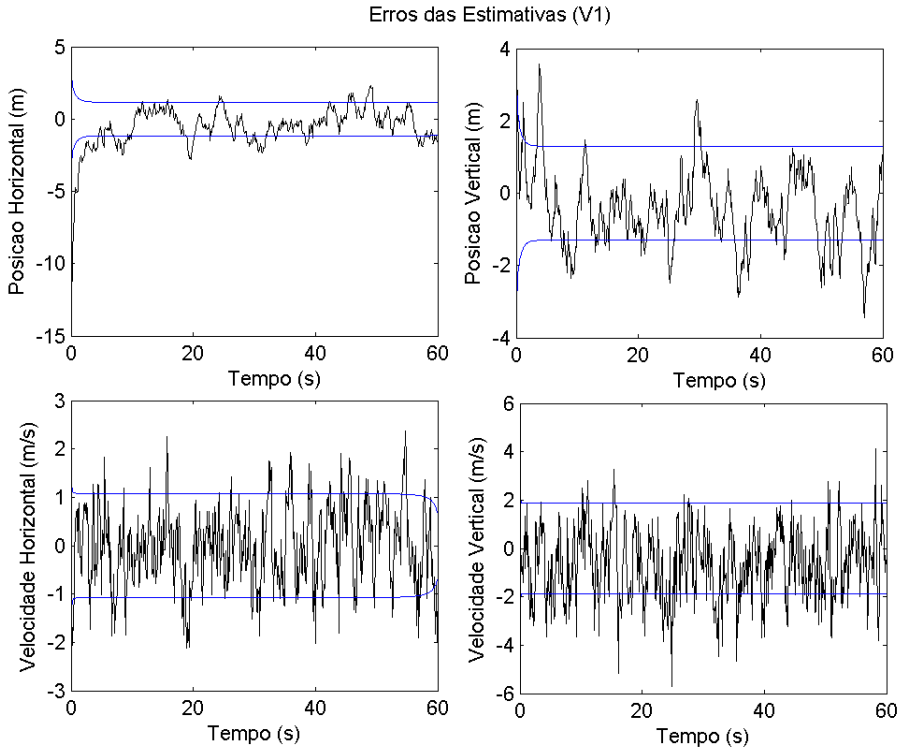


FIGURA 4.2 – Erro da estimação das medidas de $V1$ em MATLAB.

4.1.6 Simulações - MATRIXx

Os resultados apresentados a seguir foram obtidos com $60s$ de simulação no ambiente de MIS MATRIXx®, sendo os resíduos das medidas apresentados na Figura 4.4.

Obtivemos para os erros os resultados apresentados nas Figuras 4.5 e 4.6

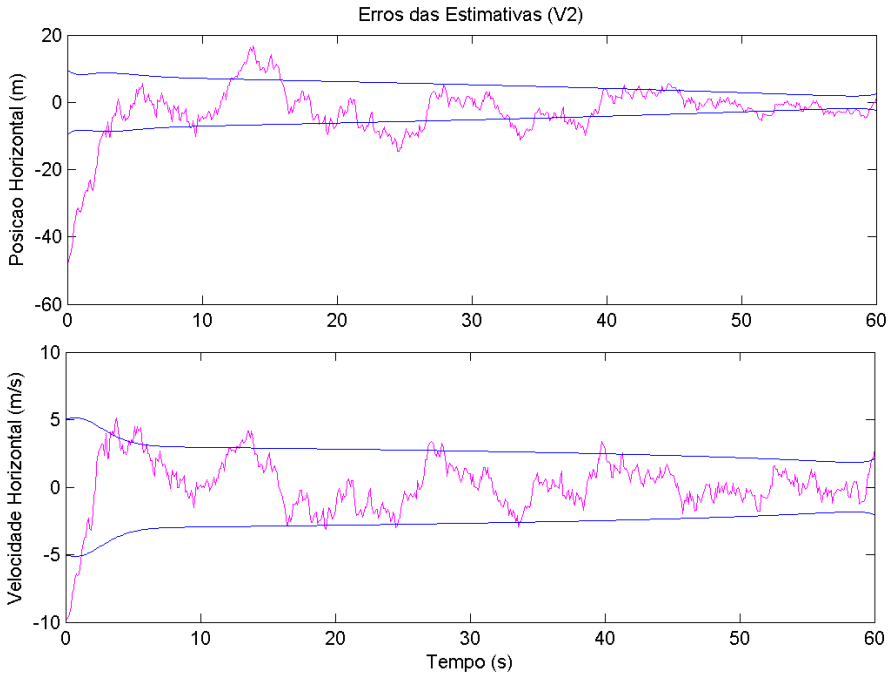


FIGURA 4.3 – Erro da estimação das medidas de $V2$ em MATLAB.

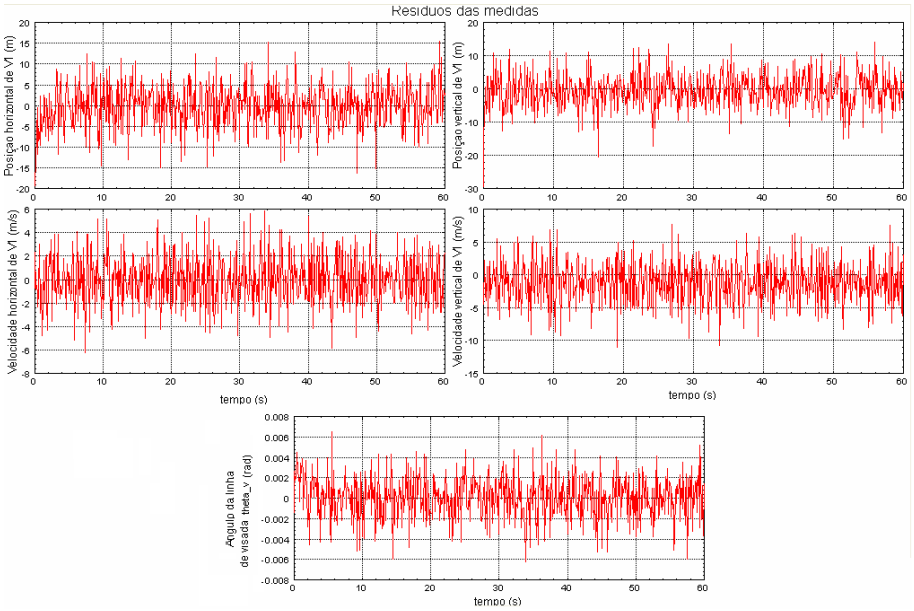


FIGURA 4.4 – Resíduos das medidas de posição e velocidade de V1 e θ_v em MATRIXx.

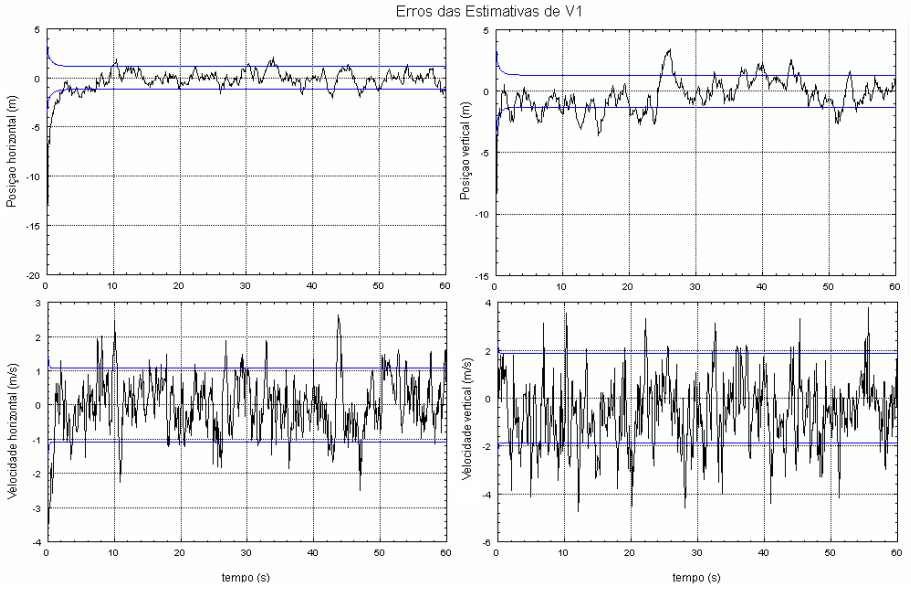


FIGURA 4.5 – Erro da estimação das medidas de $V1$ em MATRIXx.

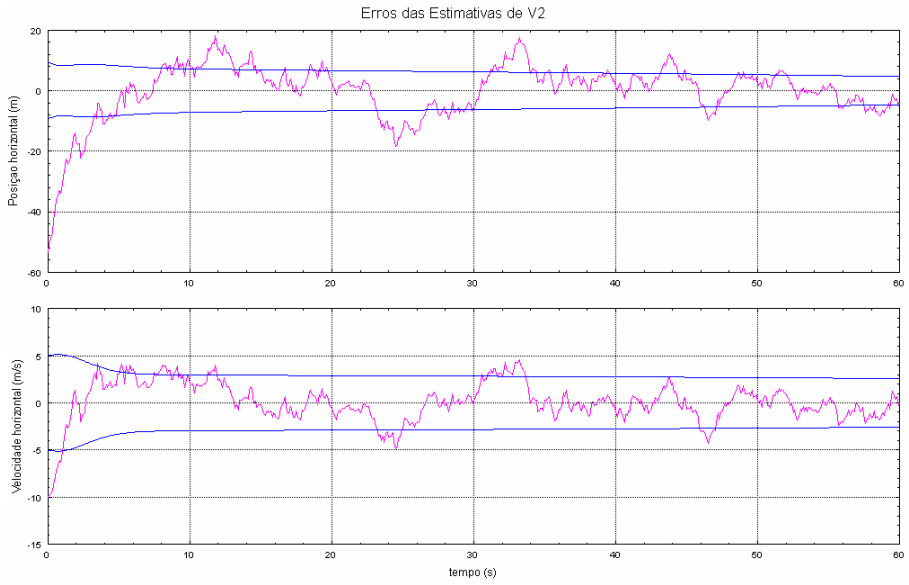


FIGURA 4.6 – Erro da estimação das medidas de $V2$ em MATRIXx.

4.1.7 Análise de Resultados

Filtragem de dados

Os mesmos resultados numéricos foram obtidos em ambos os ambientes de MIS.

Pode-se observar que os resíduos nas medidas se distribuem em torno do zero, com valores positivos e negativos, o que está de acordo com a previsão teórica. O desvio padrão deste resíduo, como esperado, é da ordem do desvio padrão no erro da medida.

Observa-se, ainda, que o filtro faz com que os valores de estado convirjam para os valores corretos, embora se tenha começado a simulação com erros nas estimativas iniciais. Embora a covariância esteja diminuindo, no intervalo de interesse ela não assume valores tão pequenos que prejudiquem o desempenho do filtro. Caso o intervalo estudado fosse maior, a covariância deveria se estabilizar. Caso continuasse diminuindo, pela falta de controlabilidade estocástica, chegaria um ponto onde o filtro divergiria. Como o tempo foi insuficiente para a divergência do filtro, percebe-se que o desempenho do filtro na estimação do estado de $V2$ foi eficiente, embora não houvesse nenhuma informação sobre sua dinâmica, além de melhorar as medidas feitas pela plataforma inercial de $V1$.

Conclui-se que, para esta primeira abordagem, o filtro desenvolvido foi adequado, produzindo estimativas de estado bem próximas do estado real. O filtro de *Kalman* apresentou resíduos consistentes com a formulação do problema; estimativas de erro próximas da realidade; tempo de convergência favorável.

4.2 Modelo 2

4.2.1 Características do Modelo

A abordagem deste novo modelo apresenta alguns graus de realismo adicionais em relação ao Modelo 1. Aumentando os graus de realismo na simulação, pode-se ter uma maior confiança nos resultados apresentados pelo controlador e estimador. O Modelo 2 é não linear, contém estados acoplados, e é controlado, com o objetivo de rastrear uma trajetória pré-definida. O tempo de simulação também é mais realista, considerando a fase de cruzeiro, que dura aproximadamente $90s$.

Ao contrário do Modelo 1, o veículo aqui modelado sofre influência não somente da aceleração da gravidade, mas também o arrasto atmosférico, provocado pela interação do veículo com a atmosfera, uma vez que sua área representativa não é desprezível. Assim, características do ambiente tiveram que ser incluídas no modelo, como a densidade do ar e a velocidade do som, além dos efeitos provocados pelo controle, já mencionado.

Este modelo, como dito anteriormente, não é linear, com estados acoplados entre si. Para se projetar uma lei de controle segundo a teoria do Regulador Linear Quadrático, deve-se fazer uma linearização do sistema. Para este modelo, tal linearização foi feita a cada $5s$, com o uso da teoria da Série de *Taylor* (Apêndice A). Tal intervalo de tempo foi escolhido pensando na eficiência do controle, mas considerando também os dados a serem armazenados no computador de bordo e seu acesso, que contam no tempo de processamento do controlador.

Observe-se que o controle aqui projetado não pode ser implementado na prática. Isto se dá pelo fato de, pelo modelo, os controles x e z seriam aplicados no referencial inercial. Seria possível sim, aplicá-lo nas direções x' e z' do referencial preso ao veículo. No entanto, como uma simplificação para este segundo modelo, consideraremos o controle aplicado nas direções x e z inerciais.

As variáveis de estado a serem estimados são as posições e velocidades de $V1$ e $V2$, ou seja, $\underline{x} = \begin{bmatrix} r_{x_1} & r_{z_1} & v_{x_1} & v_{z_1} & r_{x_2} & v_{x_2} \end{bmatrix}^T$. As medidas disponíveis são de $\underline{z}_k = \begin{bmatrix} r_{x_1} & r_{z_1} & v_{x_1} & v_{z_1} & \theta_v \end{bmatrix}^T$. Todas estas medidas estão sujeitas a ruídos mo-

delados por distribuição gaussiana de média nula e covariância conhecida.

O referencial "inercial"(denotado por xOz) está fixo na Terra, com sua origem na direção do ponto onde começa a fase de cruzeiro. Há um referencial fixo no corpo (denotado por $x'Oz'$), porém, como as equações estão dadas no referencial inercial, e suas medidas são em relação ao referencial inercial, não há necessidade de uma tranformação de referenciais. Estes referenciais podem ser conferidos na Figura 4.7.

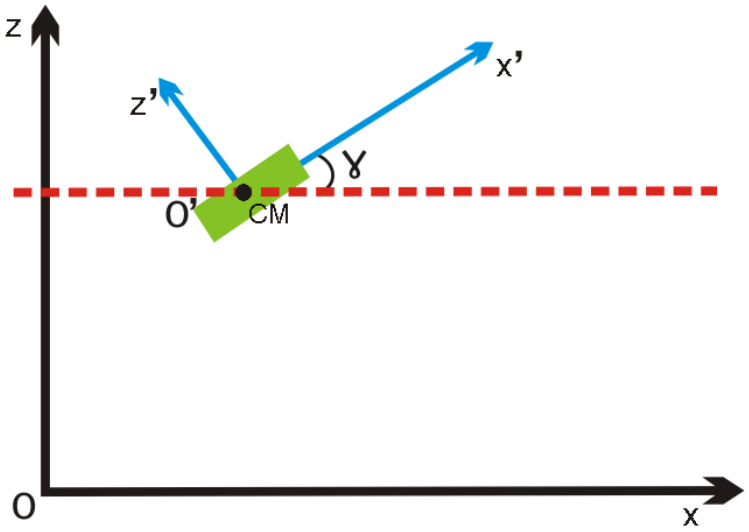


FIGURA 4.7 – Referenciais inercial e do corpo.

Foram feitas as simulações do problema nos ambientes de MIS MATLAB® e MATRIXx®, assim como a implementação do Filtro de *Kalman*.

4.2.2 Equações

A dinâmica do sistema, neste caso, ainda que simplificada, não é linear, e tem seus estados acoplados. Assim, uma representação geral do sistema é dada por:

$$\dot{\underline{x}}(t) = \underline{f}(\underline{x}(t), \underline{u}(t), t) \tag{4.25}$$

$$\underline{z}(t) = \underline{h}(\underline{x}(t), t) + \underline{v}(t) \tag{4.26}$$

Onde os ruídos nos sensores são modelados da mesma forma que nas Equações 4.3.

A dinâmica de $V1$ é explicitada nas equações descritas a seguir:

$$\dot{r}_x(t) = v_x \tag{4.27}$$

$$\dot{r}_z(t) = v_z \tag{4.28}$$

$$\dot{v}_x(t) = \left[F_E\frac{v_x}{v} - \frac{\rho S C_D}{2} v v_x + u_x\right]\frac{1}{m} \tag{4.29}$$

$$\dot{v}_z(t) = \left[F_E\frac{v_z}{v} - \frac{\rho S C_D}{2} v v_z - mg + u_z\right]\frac{1}{m} \tag{4.30}$$

Onde:

- $F_E = 1500N$
- $\rho = \rho_0 \exp\left(\frac{r_{z1}}{H_0}\right)$, com $\rho_0 = 1,752kg/m^3$ e $H_0 = 6700m$
- $S = 0,04m^2$
- $m = 300kg$
- $g = 9,8m/s^2$
- $v = \sqrt{v_x^2 + v_z^2}$
- $M_a = \frac{v}{v_{som}}$

O valor do coeficiente aerodinâmico C_D foi extrapolado de acordo com a Tabela 4.1.

Observa-se que para este modelo, uma simplificação é feita: a força de empuxo F_E está sempre na direção do vetor velocidade e a sustentação está sempre na vertical, direcionada para cima, contrabalançando a força peso. Assim, na situação de equilíbro, quando a velocidade do veículo é tal que o módulo da força de arrasto se iguala ao módulo da força de empuxo, e a sustentação equilibra a força peso, a

TABELA 4.7 – Valores do coeficiente aerodinâmico C_D para valores diferentes de *Mach*.

Mach	C_D	Mach	C_D
0	0.98	0.88	1.04
0.12	0.98	1.04	1.21
0.27	0.97	1.2	1.22
0.41	0.96	1.35	1.21
0.56	0.96	1.47	1.19
0.72	0.96	2.51	1.17

resultante de forças sobre o veículo é nula. Portanto, seu movimento será influenciado basicamente pela força de controle que será aplicada no veículo.

A dinâmica do veículo rastreado $V2$ é a mesma dinâmica modelada anteriormente (Equações 4.8).

A trajetória de referência a ser seguida por $V1$ é dada por:

$$r_{x_r}(t) = \begin{cases} 200t + 1000; & 0 \leq t < 20 \\ 250t; & 20 \leq t \leq 90 \end{cases} \tag{4.31}$$

$$r_{z_r}(t) = \begin{cases} 50t + 3000; & 0 \leq t \leq 30 \\ -70t + 6600; & 30 < t \leq 60 \\ 2400; & 60 < t \leq 90 \end{cases} \tag{4.32}$$

$$v_{x_r}(t) = \begin{cases} 200; & 0 \leq t \leq 20 \\ 10t; & 20 < t \leq 25 \\ 250; & 25 < t \leq 90 \end{cases} \tag{4.33}$$

$$v_{z_r}(t) = \begin{cases} 50; & 0 \leq t \leq 30 \\ -24t + 770; & 30 < t \leq 35 \\ -70; & 35 < t \leq 60 \\ 14t - 910; & 60 < t \leq 65 \\ 0; & 65 < t \leq 90 \end{cases} \tag{4.34}$$

Estas trajetórias são consideradas no controlador com amostras tomadas a cada $5s$,

não de forma contínua.

4.2.3 Controle

Apresentamos aqui os requisitos da missão (rastreamento de trajetória). Estes requisitos serão o parâmetro pra se avaliar o desempenho do sistema de controle. Assim, o objetivo de se controlar $V1$ é fazer com que ele siga uma trajetória pré-definida. Deseja-se um controle que, em no máximo $30s$ de trajetória, alcance uma posição (horizontal e vertical) com menos de 10% de erro em relação à trajetória de referência, além de ter um *overshooting* menor que 15%. Deseja-se, igualmente, que o esforço de controle, em módulo, não exceda $4m_1g$.

Para se utilizar a teoria do Regulador Linear Quadrático, o modelo da planta deve estar representado de forma linear, ainda que variante no tempo. Como o modelo descrito anteriormente é não-linear, procede-se uma linearização dele, baseado na teoria da Série de Taylor (Apêndice A). Esta linearização foi feita a cada $5s$, com os estados definidos pela trajetória de referência, ou seja, assume-se que o controle se encarregará de levar o veículo às proximidades da trajetória desejada, e o controle de referência será aquele que contrabalanceará a força peso.

Assim teremos uma planta linearizada da forma:

$$\Delta\underline{\dot{x}} = A\Delta\underline{x} + B\Delta\underline{u} \tag{4.35}$$

Em que:

$$A = F_x = \frac{\partial \underline{f}}{\partial \underline{x}}\Big|_{\underline{x}=\underline{r}} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & \frac{\partial \dot{v}_x}{\partial r_y}\Big|_{r_z=r_{zr}} & \frac{\partial \dot{v}_x}{\partial v_x}\Big|_{v_x=v_{xr}} & \frac{\partial \dot{v}_x}{\partial v_y}\Big|_{v_z=v_{zr}} & 0 & 0 \\ 0 & \frac{\partial \dot{v}_y}{\partial r_y}\Big|_{r_z=r_{zr}} & \frac{\partial \dot{v}_y}{\partial v_x}\Big|_{v_x=v_{xr}} & \frac{\partial \dot{v}_y}{\partial v_y}\Big|_{v_z=v_{zr}} & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{4.36}$$

Ou seja:

$$\frac{\partial \dot{v}_x}{\partial r_z}\Big|_{r_z=r_{zr}} = -\frac{1}{m}\frac{\rho S C_D}{2H_0} v v_x \tag{4.37}$$

$$\frac{\partial \dot{v}_x}{\partial v_x}\Big|_{v_x=v_{xr}} = \frac{1}{m}\left[F_E\left(\frac{1}{v}+\frac{v_x^2}{v^3}\right) - \frac{\rho S C_D}{2}\left(\frac{v_x^2}{v}+v\right)\right] \tag{4.38}$$

$$\frac{\partial \dot{v}_x}{\partial v_z}\Big|_{v_z=v_{zr}} = \frac{1}{m}\left[-F_E\left(\frac{v_x v_z}{v^3}\right) - \frac{\rho S C_D}{2}\left(\frac{v_x v_z}{v}\right)\right] \tag{4.39}$$

$$\frac{\partial \dot{v}_y}{\partial r_z}\Big|_{r_z=r_{zr}} = -\frac{1}{m}\frac{\rho S C_D}{2H_0} v v_z \tag{4.40}$$

$$\frac{\partial \dot{v}_y}{\partial v_x}\Big|_{v_x=v_{xr}} = \frac{1}{m}\left[-F_E\left(\frac{v_z v_x}{v^3}\right) - \frac{\rho S C_D}{2}\left(\frac{v_z v_x}{v}\right)\right] \tag{4.41}$$

$$\frac{\partial \dot{v}_y}{\partial v_z}\Big|_{v_z=v_{zr}} = \frac{1}{m}\left[F_E\left(\frac{1}{v}+\frac{v_z^2}{v^3}\right) - \frac{\rho S C_D}{2}\left(\frac{v_z^2}{v}+v\right)\right] \tag{4.42}$$

E, também, sabendo que:

$$B = F_u = \frac{\partial \underline{f}}{\partial \underline{u}}\Big|_{\underline{u}=\underline{u}_r} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \frac{\partial \dot{v}_x}{\partial u_x}\Big|_{u_x=u_{xr}} & 0 \\ 0 & \frac{\partial \dot{v}_z}{\partial u_z}\Big|_{u_z=u_{zr}} \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \tag{4.43}$$

Teremos:

$$\left.\frac{\partial \dot{v}_x}{\partial u_x}\right|_{u_x=u_{xr}} = \frac{1}{m} \tag{4.44}$$

$$\left.\frac{\partial \dot{v}_y}{\partial u_y}\right|_{u_y=u_{yr}} = \frac{1}{m} \tag{4.45}$$

De posse das equações linearizadas, o cálculo do controlador ótimo foi feito utilizando rotinas do MATLAB® e do MATRIXx®, e então este controlador foi inserido na planta. Os valores das matrizes Q_R (matriz de peso dos estados) e R_R (matriz de peso dos controles) utilizadas para o rastreamento foram:

$$Q_R = diag\begin{bmatrix} 10 & 10 & 1 & 1 \end{bmatrix} \tag{4.46}$$

$$R_R = diag\begin{bmatrix} 0.1 & 0.1 \end{bmatrix} \tag{4.47}$$

Portanto, o sistema linearizado fica, em diagrama de blocos, conforme Figura 4.8:

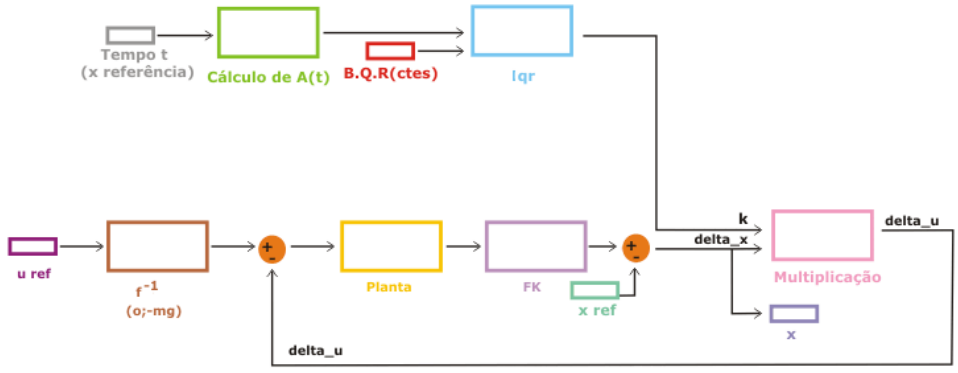


FIGURA 4.8 – Sistema controlado, com planta linearizada.

O sistema em diagrama de blocos feito em *Simulink* (MATLAB®) é apresentado na Figura 4.9, enquanto os ganhos do controlador variam no tempo conforme a Figura 4.10.

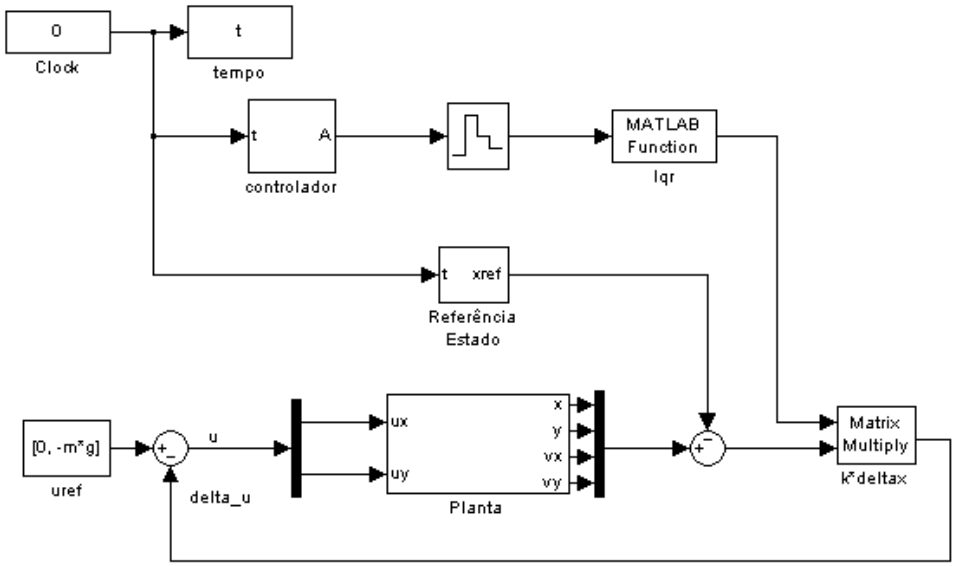


FIGURA 4.9 – Diagrama de Blocos de $V1$ em *Simulink* (MATLAB).

O sistema em diagrama de blocos feito em *SystemBuild* (MATRIXx®) é apresentado na Figura 4.11, enquanto os ganhos do controlador variam no tempo conforme a Figura 4.12.

4.2.4 Filtro de Kalman

Na fase de propagação do estado, utilizaram-se as equações de movimento dos veículos, considerando uma dinâmica contínua, com medidas discretas.

Aproximamos a dinâmica do veículo como uma dinâmica não variante no tempo, por trechos. A forma de fazer uma simulação com esta consideração estará ligada ao ambiente de MIS utilizado. Como incertezas na modelagem, supusemos desconhecimento dos parâmetros que influenciam a variação da densidade do ar. Assim, a densidade do ar modelada no filtro é dada por:

$$\rho = \rho_0 \exp\left(\frac{r_{z1}}{H_0}\right) \tag{4.48}$$

Em que $\rho_0 = 2$ e $H_0 = 7000$. Ou seja, tal desconhecimento da planta faz as vezes de um "ruído"na dinâmica.

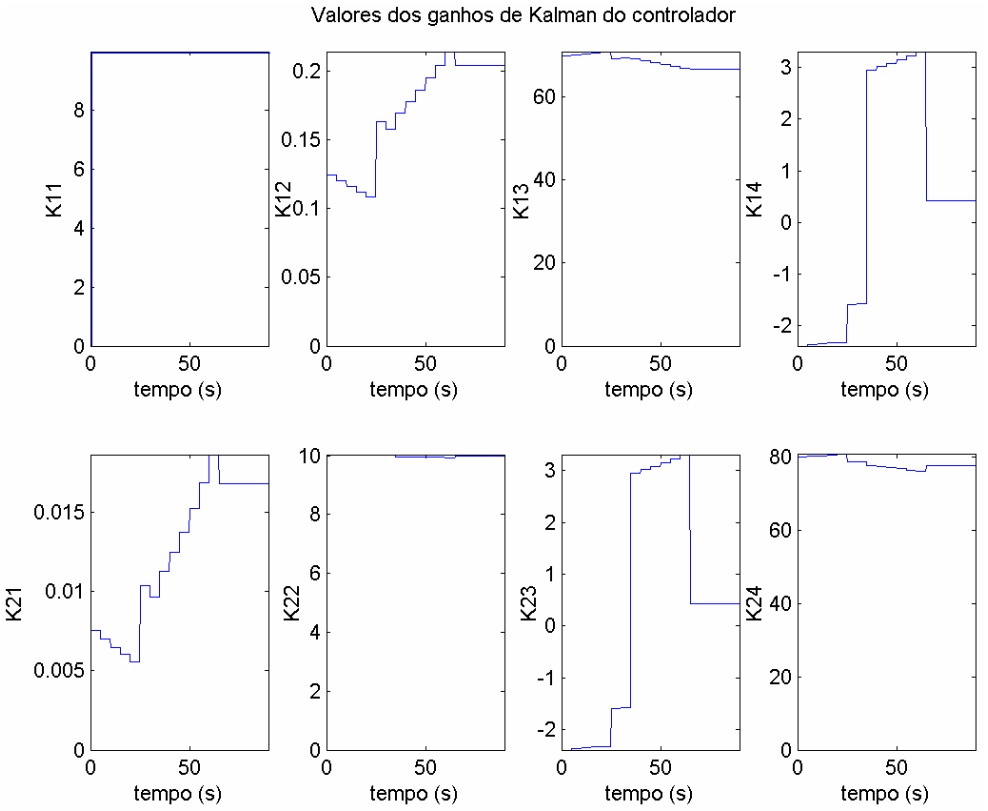


FIGURA 4.10 – Ganhos do controlador para o Modelo 2 - MATLAB.

Representando a dinâmica do sistema como equações lineares variantes no tempo pode-se realizar a propagação do estado através da matriz de transição ϕ, considerando, no entanto, a atuação do controle na planta. Consideramos incertezas em todas os estados da propagação, que são inseridas através da covariância do processo branco, Q_F, que teve seu valor inicial ajustado para melhor desempenho do filtro.

Assim, a fase de propagação do filtro é feita através das equações:

$$\overline{x}_k = \phi\hat{x} + \int_{t_{k-1}}^{t_k} \phi(t_k,\tau)B(\tau)\underline{u}(\tau)d\tau \tag{4.49}$$

$$\overline{P}_{Fk} = \phi\hat{P}_{Fk}\phi^T + \Gamma_{Gk}Q_F\Gamma_{Gk}^T \tag{4.50}$$

E a fase de atualização é dada pelas equações discretas:

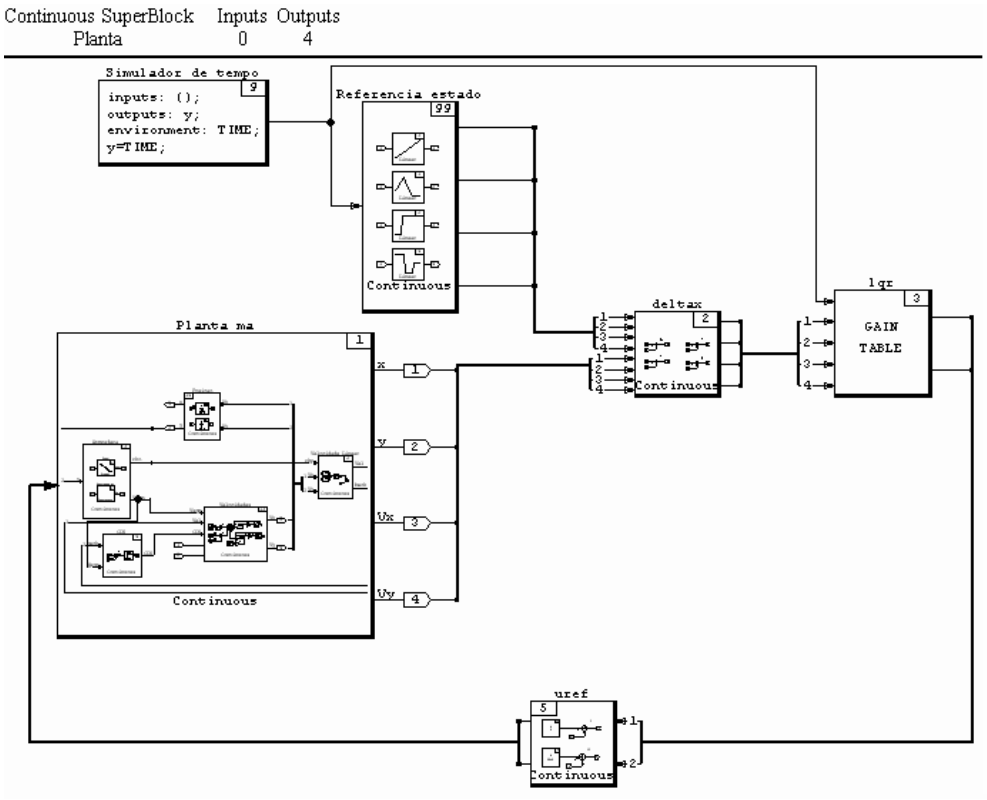


FIGURA 4.11 – Diagrama de Blocos de $V1$ em *SystemBuild* (MATRIXx).

$$K_k = \overline{P}_{Fk} H_{Fk}^T \left(H_{Fk} \overline{P}_{Fk} H_{Fk}^T + R_F \right)^{-1} \tag{4.51}$$

$$\hat{P}_{Fk} = \left(I - K_k H_{Fk} \right) \overline{P}_{Fk} \tag{4.52}$$

$$\underline{\hat{x}}_k = \underline{\overline{x}}_k + K_k \left(z_k - H_{Fk} \underline{\overline{x}}_k \right) \tag{4.53}$$

Onde, novamente, H_{Fk} é a matriz que relaciona as medidas aos estados no instante t_k. Neste caso, $H_{Fk} \equiv \dfrac{\partial \underline{h}}{\partial \underline{x}}$, em que $\underline{h}$ é a função vetorial que relaciona as medidas aos estados.

As medidas disponíveis são aquelas feitas pela plataforma inercial embarcada em $V1$ e pelo to diretor, ou seja, medidas das componentes da posição e velocidade de $V1$ e do ângulo da linha de visada θ_v. Nesta abordagem, consideramos o ângulo θ_v aproximado pelo ângulo entre as posições de $V1$ e $V2$. Assim, temos as equações de medidas:

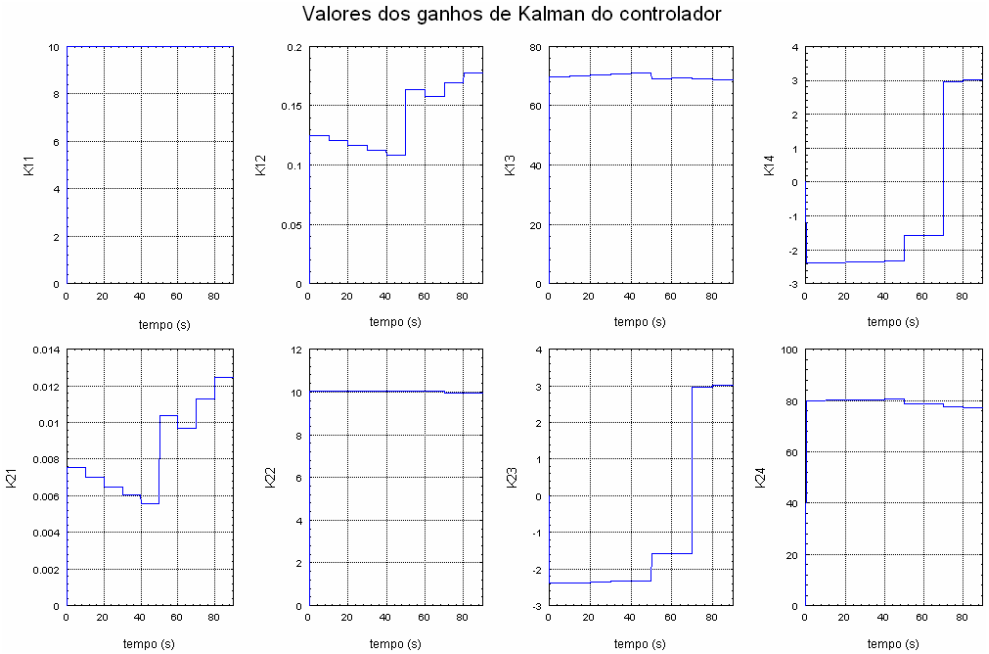


FIGURA 4.12 – Ganhos do controlador para o Modelo 2 - MATRIXx.

$$\underline{z}_k = \begin{bmatrix} r_{x_1} \\ r_{z_1} \\ v_{x_1} \\ v_{z_1} \\ \theta_v \end{bmatrix} = \begin{bmatrix} r_{x_1} \\ r_{z_1} \\ v_{x_1} \\ v_{z_1} \\ \arctan\left(\dfrac{r_{z_1}}{r_{x_2} - r_{x_1}}\right) \end{bmatrix} + \underline{v}(t) \qquad (4.54)$$

As seguintes condições iniciais foram utilizadas:

- Estado inicial (posições e velocidades de $V1$, posição e velocidade horizontais de $V2$): $\underline{x} = \begin{bmatrix} r_{x10} & r_{z10} & v_{x10} & v_{z10} & r_{x20} & v_{x20} \end{bmatrix}^T = \begin{bmatrix} 0 & 5000 & 150 & 15 & 25000 & 15 \end{bmatrix}^T$; em unidades do S.I.;
- Erro inicial na estimativa do estado: $\Delta\underline{x} = \underline{\hat{x}}(0) - \underline{x}(0) = \begin{bmatrix} 10 & 10 & 3 & 3 & 50 & 10 \end{bmatrix}^T$, em unidades do S.I.;
- Valor inicial da matriz de covariância do estado $\underline{x}$: $P_{Fk}(0) = P_{k0} = diag \begin{bmatrix} 5^2 & 5^2 & 2^2 & 3^2 & 10^2 & 5^2 \end{bmatrix}$

- Valor da matriz de densidade espectral de potência do distúrbio $\underline{\omega}(t)$: $\Gamma_k Q_F \Gamma_k^T = diag \left[\begin{array}{cccccc} 0.2^2 & 0.2^2 & 0.7^2 & 1.5^2 & 0.2^2 & 0.5^2 \end{array} \right]$
- Valor da matriz de covariância do ruído $\underline{v}(t)$: $R_F = diag \left[\begin{array}{ccccc} 5^2 & 5^2 & 2^2 & 3^2 & (2e-3)^2 \end{array} \right]$

Simularam-se as posições e velocidades de $V1$ e $V2$ exatas (estado real) para uma comparação com os valores estimados pelo filtro.

4.2.5 Simulações - MATLAB

Os resultados apresentados a seguir foram obtidos com $90s$ de simulação no ambiente de MIS MATLAB®, sendo os resíduos das medidas apresentados na Figura 4.13

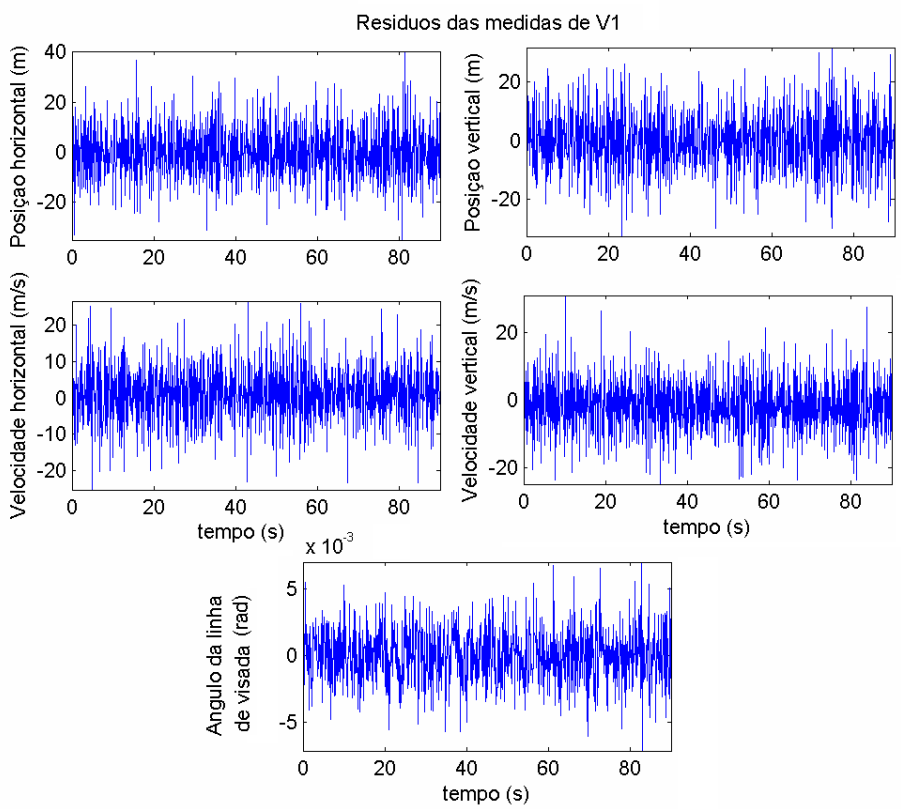


FIGURA 4.13 – Resíduos das medidas de V1.

Os erros das estimativas dos estados de $V1$ e $V2$ são apresentados nas Figuras 4.14 e 4.15.

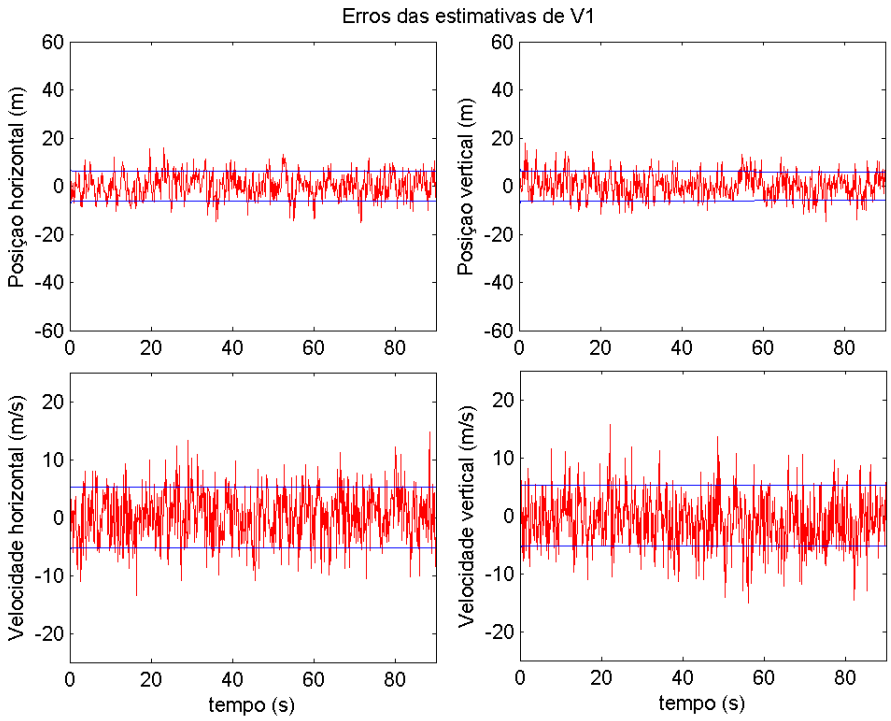


FIGURA 4.14 – Erros de estimação dos estados de V1.

Os estados estimados de $V1$ e $V2$ podem ser conferidos nas Figuras 4.16 e 4.17.

O fator de carga utilizado para tais manobras foi limitado em um máximo de $4g$ e pode ser visto na Figura 4.18.

4.2.6 Simulações - MATRIXx

Os resultados apresentados a seguir foram obtidos com $90s$ de simulação no ambiente de MIS MATRIXx®, sendo os resíduos das medidas apresentados na Figura 4.19

Os erros das estimativas dos estados de $V1$ e $V2$ são apresentados nas Figuras 4.20 e 4.21.

Os estados estimados de $V1$ e $V2$ podem ser conferidos nas Figuras 4.22 e 4.23.

O controle utilizado para tais manobras pode ser conferido na Figura 4.24.

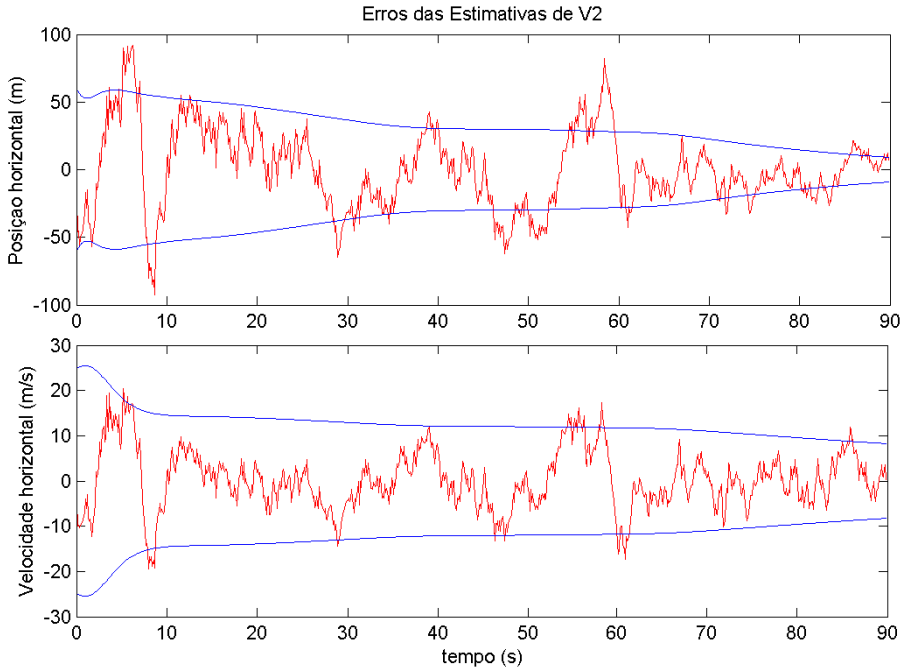


FIGURA 4.15 – Erros de estimação dos estados de V2.

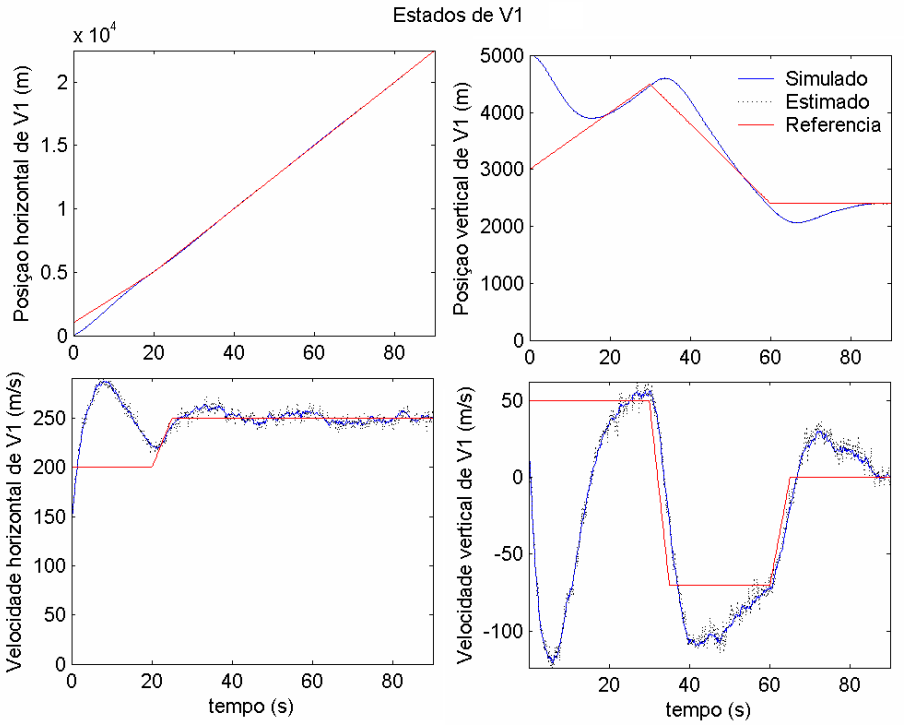


FIGURA 4.16 – Estados de V1, referência, com realimentação com estados reais, e com realimentação com estados estimados.

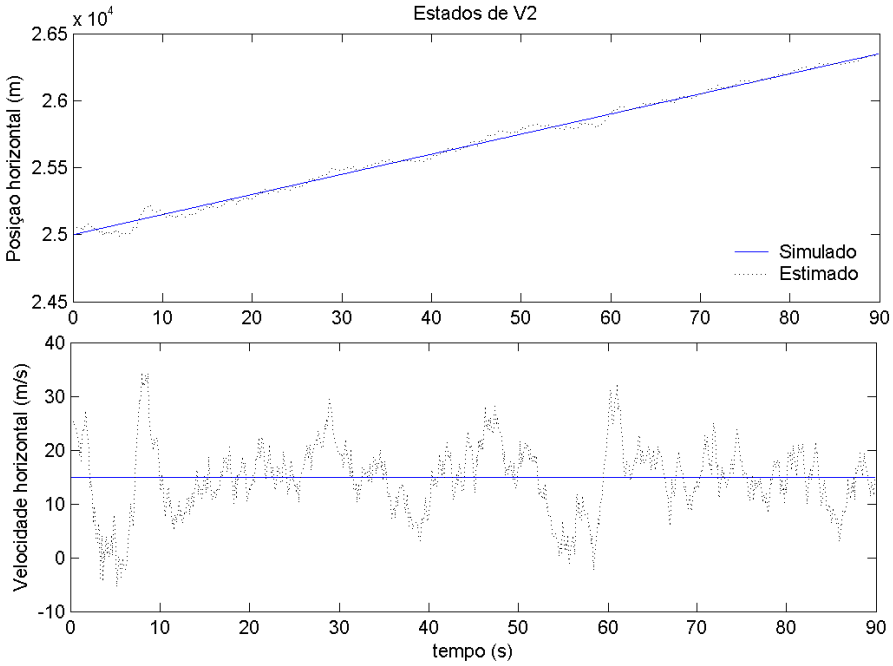


FIGURA 4.17 – Estados de V2, real e estimado.

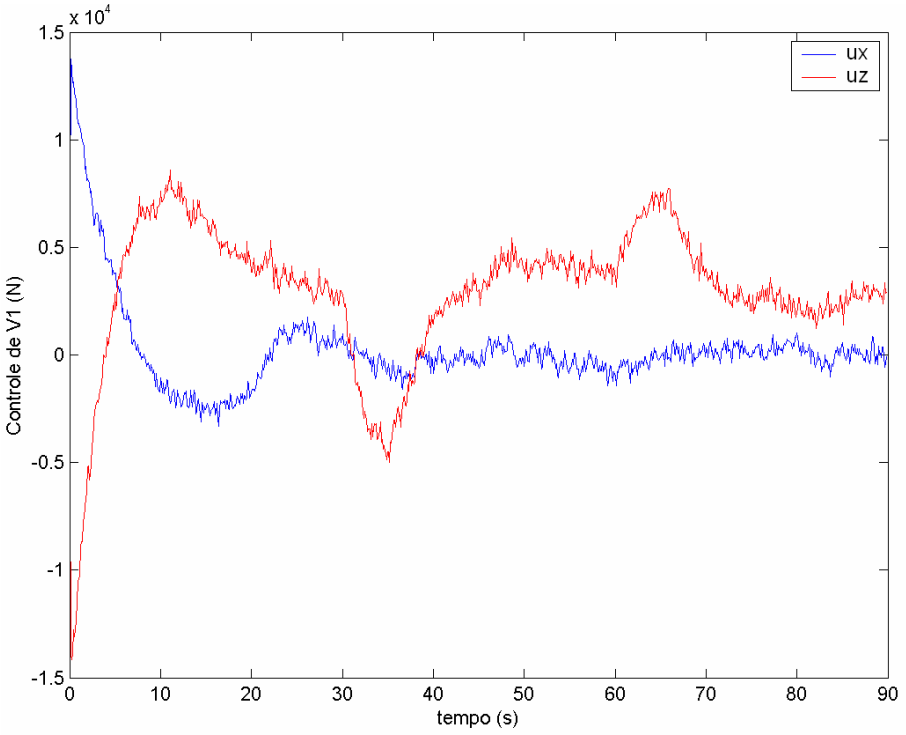


FIGURA 4.18 – Controle empregado pelo atuador para o rastreamento da trajetória, nas direções x e z.

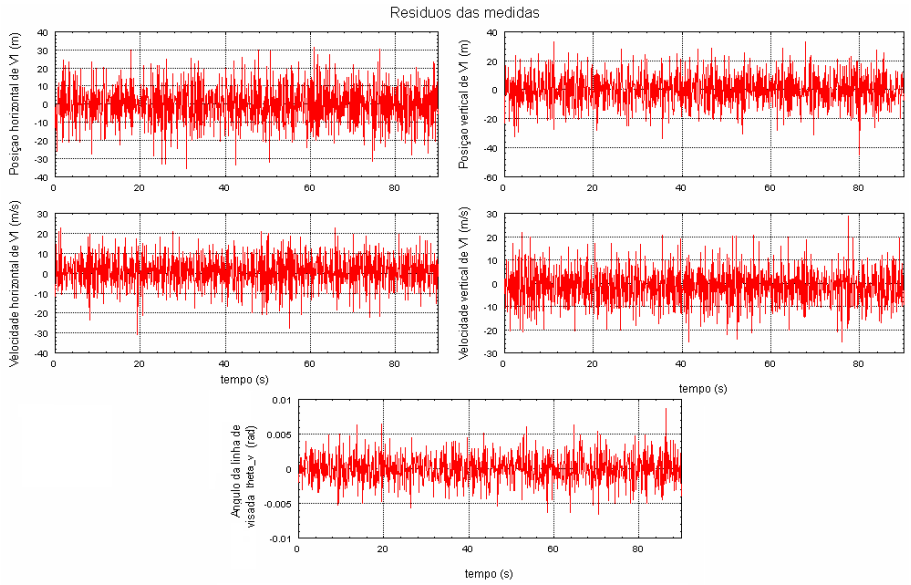


FIGURA 4.19 – Resíduos das medidas de V1.

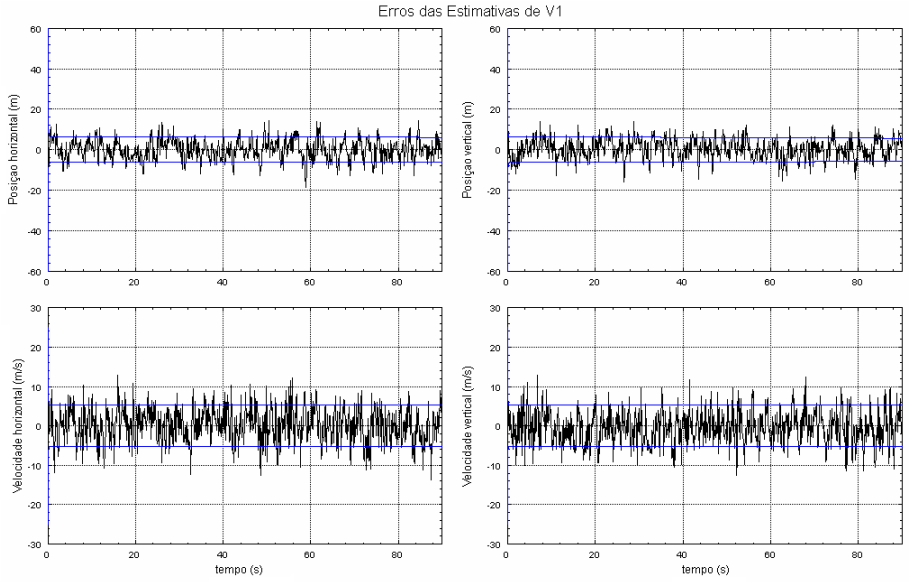


FIGURA 4.20 – Erros de estimação dos estados de V1.

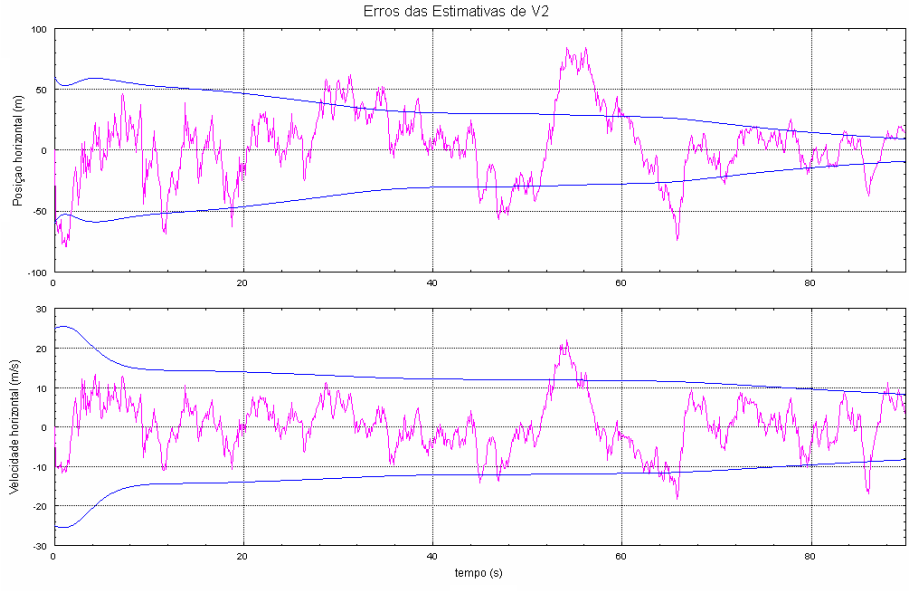


FIGURA 4.21 – Erros de estimação dos estados de V2.

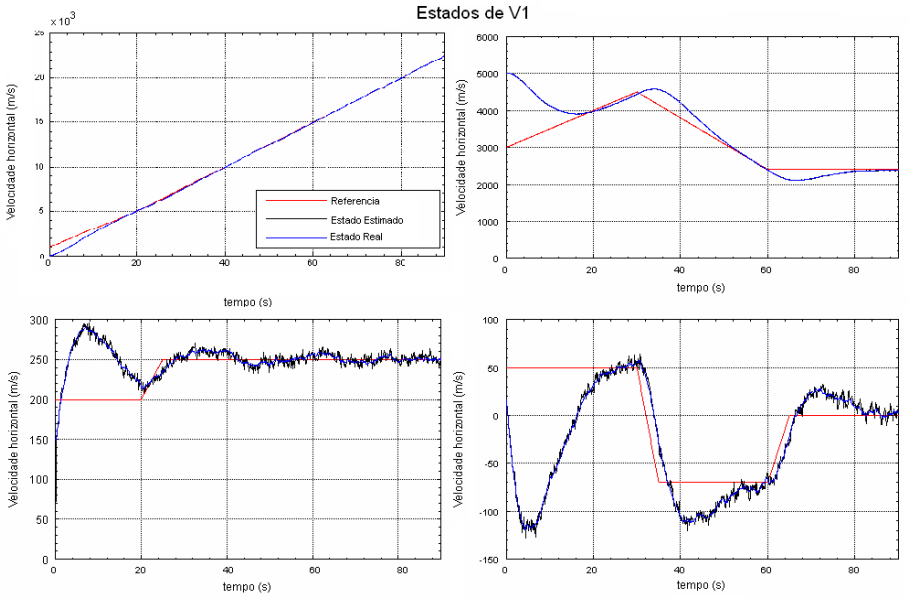


FIGURA 4.22 – Estados de V1, referência, com realimentação com estados reais, e com realimentação com estados estimados.

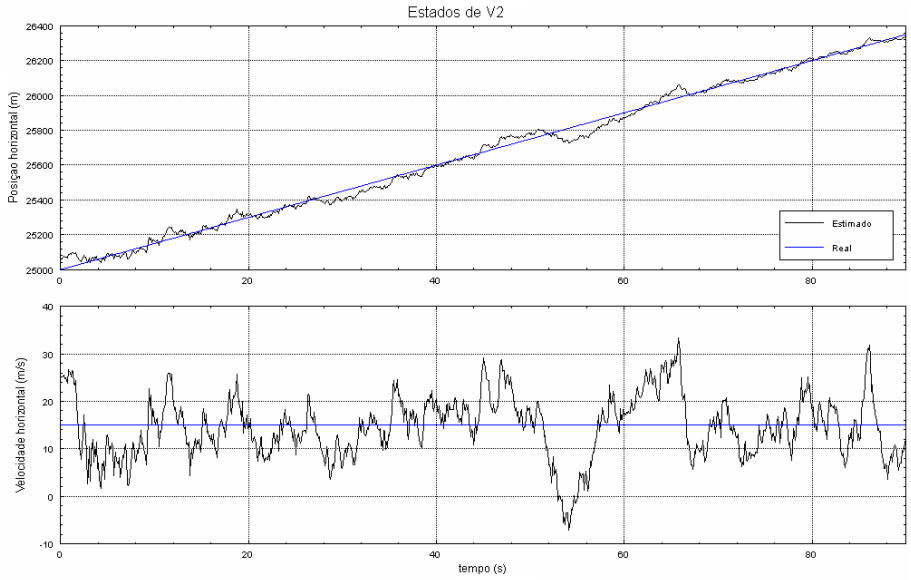


FIGURA 4.23 – Estados de V2, real e estimado.

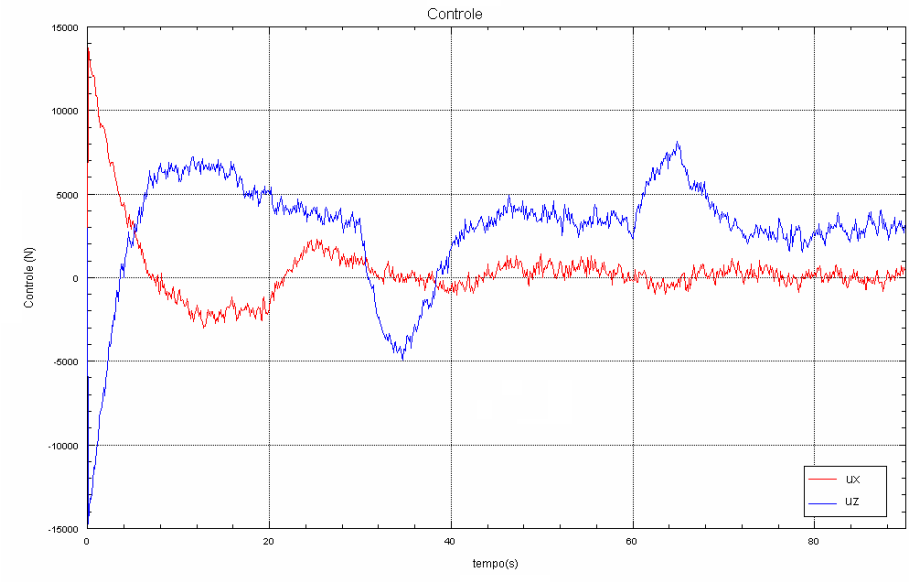


FIGURA 4.24 – Controle empregado pelo atuador para o rastreamento da trajetória, nas direções x e z.

4.2.7 Análise de Resultados

Controle

A Lei de Controle fez com que o veículo $V1$ seguisse a trajetória desejada dentro das especificações apresentadas. O máximo de fator de carga especificado foi respeitado. Para tanto, não fizemos uma saturação do controle no limite especificado, e sim foi feita uma ponderação dos pesos nas matrizes do controlador, que foram ajustados manualmente até um valor que respeitasse as especificações. Observa-se que, quando a trajetória de posição vertical desejada se aproxima de uma reta, o controle na direção z é suficiente pra contrabalançar o peso.

Apesar de o controle ter sido feito adequadamente, observamos que os dois ambientes fornecem valores diferentes para os ganhos de *Kalman* do controlador. Tal diferença se dá pelo uso das rotinas prontas do regulador linear quadrático dos ambientes. Valores iguais das matrizes do sistema e matrizes de peso foram fornecidas, sendo, no entanto, os resultados dos dois ambientes mantidos diferentes. Apesar disto, os valores do ganho são da mesma ordem de grandeza, fazendo com que o controle projetado por ambos os ambientes seja eficiente.

Filtragem de dados

Pode-se observar que os resíduos nas medidas se distribuem em torno do zero, com valores positivos e negativos, como era de se esperar. O desvio padrão deste resíduo, como esperado, é da ordem do desvio padrão no erro da medida.

Pode-se observar que o filtro faz com que os valores de estado convirjam para o valor correto, embora se tenha começado a simulação com erros nas estimativas iniciais. A covariância tende a se estabilizar e o erro no estado tende a ficar dentro da faixa estabelecida pela covariância. Com isto, percebe-se que o desempenho do filtro na estimação do estado de $V2$ foi eficiente, embora não houvesse nenhuma informação sobre sua dinâmica, além de melhorar as medidas feitas pela plataforma inercial de $V1$.

Conclui-se que, para esta primeira abordagem, o filtro desenvolvido foi adequado, produzindo estimativas de estado bem próximas do estado real. O filtro de *Kalman* apresentou resíduos consistentes com a formulação do problema; estimativas de erro próximas da realidade; tempo de convergência favorável.

4.3 Modelo 3

4.3.1 Características do Modelo

O modelo aqui apresentado é desenvolvido em Cheng e Gupta (1986). Seu modelo, apresentado em 3 dimensões, foi aqui adaptado para seguir uma trajetória no plano. Seus valores de massa, área representativa, empuxo, entre outros, foram adaptados para representar casos mais realistas. É um modelo não linear, também com estados acoplados. O tempo de simulação é novamente $90s$, tempo este compatível com a realidade de tempo de cruzeiro.

Como o Modelo 2, este modelo apresenta interação com a atmosfera, representada pelo termo relativo ao arrasto que aparece nas equações. As variáveis de estado foram mudadas, de forma que o seu controle é implementável por ser definido em um referencial fixo no corpo, onde estão presos os atuadores.

Novamente é feita uma linearização do modelo, com o auxílio da Série de *Taylor* (Apêndice A), para que a Teoria do LQR seja aplicável.

4.3.2 Equações

Embora este modelo seja mais realista que os modelos apresentados anteriormente, ele ainda apresenta algumas hipóteses simplificadoras. O movimento ainda se dá em um único plano, que é o plano entre o veículo aeroespacial e o veículo rastreado. A força de empuxo se dá na direção do vetor velocidade. O controle é feito através do ângulo que o veículo faz com a direção horizontal. O único efeito atmosférico considerado é o arrasto.

A dinâmica de $V1$ obedece às seguintes equações:

$$\dot{r}_{x1}(t) = V_1(t)\cos\gamma(t) \tag{4.55}$$

$$\dot{r}_{z1}(t) = V_1(t)\sin\gamma(t) \tag{4.56}$$

$$\dot{V}_1(t) = \frac{F_E(t)}{m(t)} - \frac{\rho S C_D V_1(t)^2}{2m(t)} - g\sin\gamma(t) \tag{4.57}$$

$$\dot{\gamma}(t) = \frac{-g\cos\gamma(t)}{V_1(t)} + \frac{u}{m(t)V_1(t)} \tag{4.58}$$

Em que:

- $S = 0,04m^2$
- $g = 9,8m/s^2$
- $M_a = \dfrac{V}{v_{som}}$
- $\rho = \rho_0 \exp\left(\dfrac{r_{z1}}{H_0}\right)$, com $\rho_0 = 1,752kg/m^3$ e $H_0 = 6700m$

Os valores do empuxo, $F_E(t)$, massa, $m(t)$, e coeficiente aerodinãmico, C_D, foram extrapolados de acordo com as Tabelas 4.2, 4.3 e 4.4.

A dinâmica do veículo rastreado $V2$ é a mesma dinâmica simulada anteriormente (Equações 4.8).

A trajetória de referência a ser seguida por $V1$ é dada por:

Tabela 4.24 – Valores da força de empuxo do motor de aceleração em função do tempo.

t (s)	$F_E(N)$	m_a (kg)	t (s)	$F_E(N)$	m_a (kg)	t (s)	$F_E(N)$	m_a (kg)
0	104.52	13.961	1.9	9468.9	4.4143	3.9	540.35	0.21166
0.1	5415.3	14.205	2	9114.1	4.0129	4	437.12	0.14848
0.2	12224	13.804	2.1	8765.8	3.4913	4.1	347.61	0.14255
0.3	11854	13.494	2.2	8409.3	3.0968	4.2	284.9	0.12457
0.4	11844	12.845	2.3	8053.2	2.5793	4.3	230.05	0.074148
0.5	11828	12.081	2.4	7526.4	2.1981	4.4	191.84	0.046017
0.6	11844	11.561	2.6	6614.6	1.7179	4.5	155.69	0.061756
0.7	11833	10.932	2.7	5156.4	1.4407	4.6	127.38	0.046195
0.8	11826	10.289	2.8	3832.2	1.1165	4.7	103.61	0.027513
0.9	11817	9.7389	2.9	3117.9	0.91451	4.8	84.813	0.10397
1	11780	9.2036	3	2645.1	0.77081	4.9	68.251	0.13053
1.1	11760	8.5505	3.1	2266.7	0.67417	5.1	56.552	0.07614
1.2	11730	7.8892	3.2	1932	0.59574	5.2	0.12576	0.050367
1.3	11695	7.2777	3.3	1656.5	0.60684	200	0.066126	0.03553
1.4	11560	6.7276	3.4	1402.9	0.56661			
1.5	11175	6.2955	3.5	1186.2	0.4475			
1.6	10721	5.7696	3.6	989.28	0.3505			
1.7	10284	5.2558	3.7	821.06	0.32356			
1.8	9852	4.8786	3.8	671.52	0.25473			

$$r_{x_r}(t) = \begin{cases} 315; & 0 \leq t < 17,5 \\ 247t + 1185; & 17,5 \leq t \leq 90 \end{cases} \tag{4.59}$$

$$r_{z_r}(t) = \begin{cases} 3000; & 0 \leq t \leq 10 \\ 50t + 2500; & 10 < t \leq 20 \\ 3500; & 20 < t \leq 60 \\ -100t + 9500; & 60 < t \leq 75 \\ 2000; & 75 < t \leq 90 \end{cases} \tag{4.60}$$

$$V_r(t) = \begin{cases} -5t + 350; & 0 \leq t \leq 20 \\ 250; & 20 \leq t \leq 90 \end{cases} \tag{4.61}$$

$$\gamma_r(t) = \begin{cases} 0; & 0 \leq t \leq 10 \\ \dfrac{1}{300}t + \dfrac{4}{30}; & 12,5 < t \leq 20 \\ 0; & 22,5 < t \leq 60 \\ -0,4; & 60 < t \leq 77,5 \\ 0; & 77,5 < t \leq 90 \end{cases} \tag{4.62}$$

Tabela 4.24 – Valores da força de empuxo do motor de cruzeiro em função do tempo.

t (s)	$F_E(N)$	m_c (Kg)
0	1500	70
90	1500	0
90.1	0	0
200	0	0

Tabela 4.24 – Valores do coeficiente aerodinâmico C_D para valores diferentes de *Mach*.

Mach	C_D	Mach	C_D
0	0.9	0.8	0.96
0.1	0.893	1	1.01
0.2	0.876	1.2	1.06
0.4	0.866	1.3	1.11
0.5	0.878	1.4	1.13
0.7	0.91	2.5	1.15

Novamente observa-se que esta trajetória não é computada continuamente no controlador, e sim linearizada por partes em intervalos de 2,5 segundos. Tal valor de intervalo de tempo foi diminuído em relação ao utilizado no modelo 2. A razão para isto é aumentar a eficiência do rastreio, já que os computadores de bordo utilizados atualmente suportam tal processamento em tempo factível.

4.3.3 Controle

Assim como no caso do Modelo 2, o objetivo de se controlar $V1$ é fazer com que ele siga uma trajetória pré-definida. Deseja-se um controle que, em, no máximo $30s$ de trajetória alcance uma posição (horizontal e vertical) com menos de 10% de erro em relação à trajetória de referência, além de ter um *overshooting* menor que 15%. Deseja-se, igualmente, que o esforço de controle, em módulo, não exceda $4m_1g$. Neste caso o controle foi saturado neste valor, e não ajustado para ser menor que ele.

Como feito para o Modelo 2, procedemos uma linearização do modelo, a fim de poder utilizar a teoria do Regulador Linear Quadrático. A planta fica linearizada,

embora variante no tempo, com a aplicação da teoria da Série de Taylor (Apêndice A). A linearização foi feita aqui em intervalos menores de tempo, de 2,5s, a fim de deixar a planta linearizada mais próxima do estado real. Os valores utilizados para a linearização são tirados da trajetória de referência, e o controle de referência será aquele que contrabalanceará a força peso.

Assim, a planta linearizada tomará a forma:

$$\Delta\dot{\underline{x}} = A\Delta\underline{x} + B\Delta\underline{u} \tag{4.63}$$

Assim, teremos:

$$A = F_x = \Big|_{\underline{x}=\underline{r}} = \begin{bmatrix} 0 & 0 & \frac{\partial \dot{r}_{x1}}{\partial V_1}\Big|_{V_1=V_{1r}} & \frac{\partial \dot{r}_{x1}}{\partial \gamma}\Big|_{\gamma=\gamma_r} \\ 0 & 0 & \frac{\partial \dot{r}_{y1}}{\partial V_1}\Big|_{V_1=V_{1r}} & \frac{\partial \dot{r}_{y1}}{\partial \gamma}\Big|_{\gamma=\gamma_r} \\ 0 & \frac{\partial \dot{V}_1}{\partial r_{z1}}\Big|_{r_{z1}=r_{z1r}} & \frac{\partial \dot{V}_1}{\partial V_1}\Big|_{V_1=V_{1r}} & \frac{\partial V_1}{\partial \gamma}\Big|_{\gamma=\gamma_r} \\ 0 & 0 & \frac{\partial \dot{\gamma}}{\partial V_1}\Big|_{V_1=V_{1r}} & \frac{\partial \dot{\gamma}}{\partial \gamma}\Big|_{\gamma=\gamma_r} \end{bmatrix} \tag{4.64}$$

E também:

$$B = F_u\Big|_{u=u_r} = \begin{bmatrix} 0 & 0 & 0 & \frac{1}{mV_1}\Big|_{u=u_r} \end{bmatrix}^T \tag{4.65}$$

De posse das equações linearizadas, o cálculo do controlador ótimo foi feito utilizando rotinas prontas do MATLAB® e MATRIXx®, e este controlador então foi inserido na planta. Para proceder o cálculo do controlador, as matrizes Q_R e R_R utilizadas foram:

$$Q_R = diag[\ 10 \quad 10 \quad 1 \quad 1 \quad 0 \quad 0\ ] \tag{4.66}$$

$$R_R = 0.1 \tag{4.67}$$

Como o controlador utilizado só considera os estados de $V1$, a controlabilidade estocástica do sistema não é prejudicada pelos zeros presentes na matriz Q_R.

O diagrama de blocos que representa o sistema em MATLAB®/Simulink é apresentado na Figura 4.25.

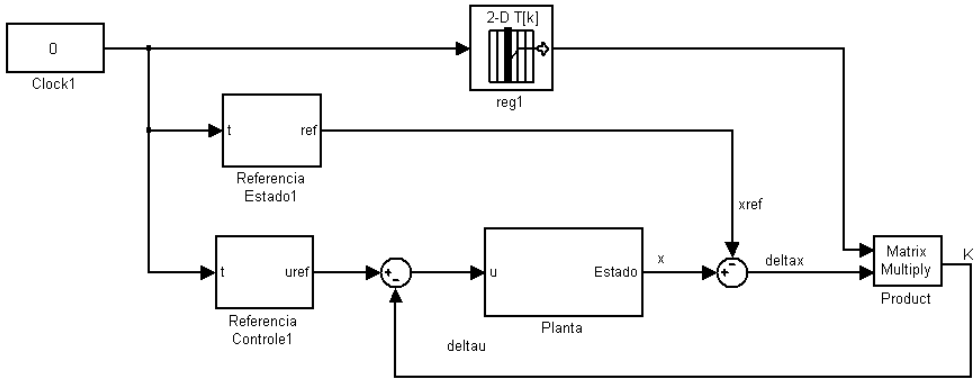


FIGURA 4.25 – Diagrama de blocos do Modelo 3 controlado, em MATLAB.

E os ganhos de Kalman do controlador podem ser vistos na Figura 4.26.

O diagrama de blocos que representa o sistema em MATRIXx®/SystemBuild é apresentado na Figura 4.27.

E os ganhos de Kalman do controlador em MATRIXx®podem ser vistos na Figura 4.28.

4.3.4 Filtro de Kalman

Consideraremos aqui uma dinâmica contínua, com medidas discretas. Neste caso, não aproximamos a dinâmica do veículo por uma dinâmica linear, quando se poderia utilizar a matriz de transição ϕ. A propagação foi feita resolvendo-se a equação dinâmica.

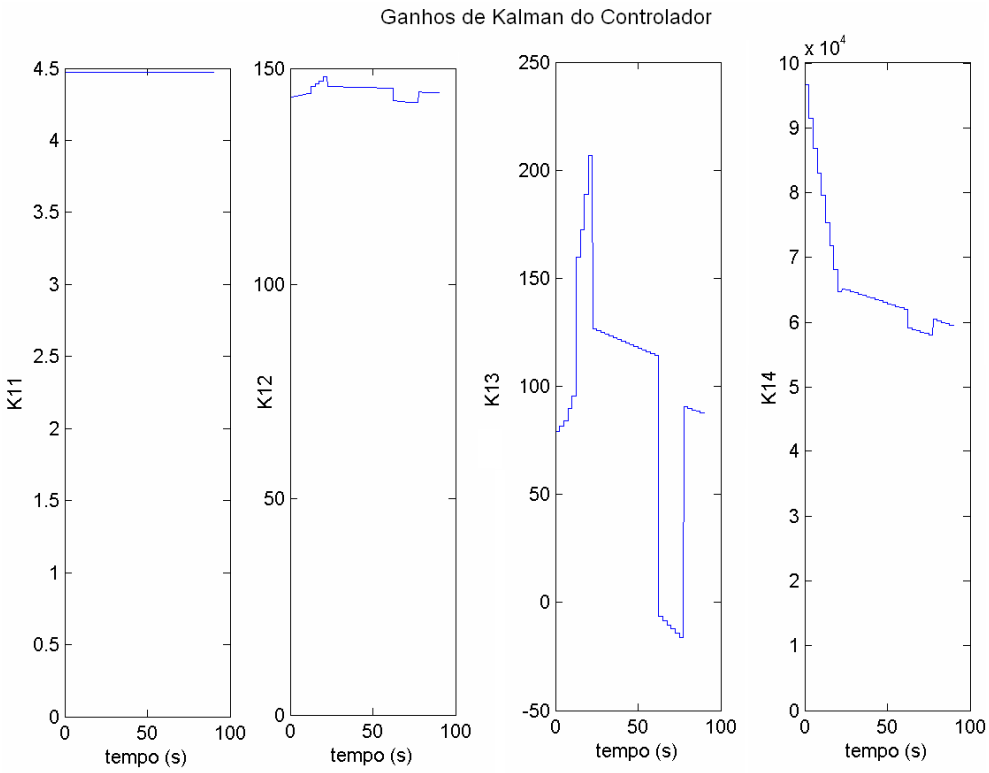


FIGURA 4.26 – Ganhos de Kalman do controlador do Modelo 3 em MATLAB.

Consideramos as incertezas novamente devidas a uma imprecisão do modelo da densidade do ar (Equação 4.70).

Assim, a fase de propagação do filtro é realizada com as seguintes equações:

$$\dot{\underline{x}} = \underline{f}_1(\underline{x}(t), \underline{u}(t), t) \tag{4.68}$$

$$\dot{\overline{P}}_F = F\overline{P}_F^T + \overline{P}_F F^T + Q_F \tag{4.69}$$

Onde $F \triangleq \dfrac{\partial \underline{f}_1(\underline{x}(t), \underline{u}(t), t)}{\partial \underline{x}}$, e as equações de atualização são dadas pelas expressões discretas (4.51, 4.52, 4.53).

Como incertezas na modelagem, supusemos desconhecimento dos parâmetros que influenciam a variação da densidade do ar. Assim, a densidade do ar modelada no filtro é dada por:

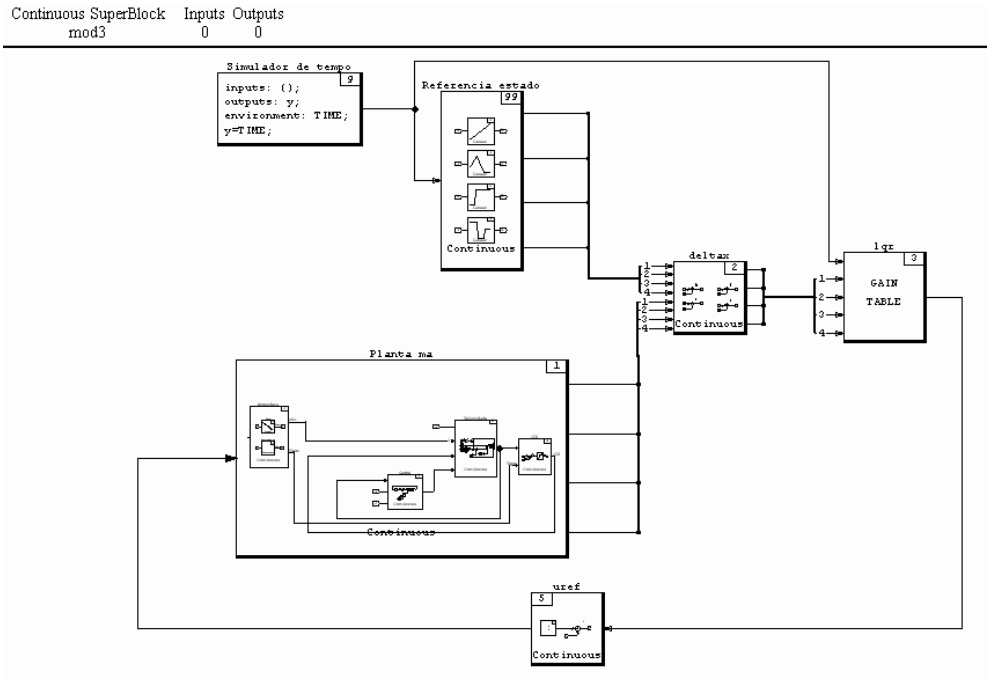


FIGURA 4.27 – Diagrama de blocos do Modelo 3, controlado em MATRIXx.

$$\rho = \rho_0 \exp\left(\frac{r_{z1}}{H_0}\right) \tag{4.70}$$

Em que $\rho_0 = 2$ e $H_0 = 7000$. Ou seja, tal desconhecimento da planta faz as vezes de um "ruído"na dinâmica.

As medidas disponíveis são aquelas feitas pela plataforma inercial embarcada em $V1$ e pelo auto diretor, ou seja, medidas das componentes da posição e velocidade de $V1$ e do ângulo da linha de visada θ_v. Nesta abordagem, consideramos o ângulo θ_v como sua definição real, ou seja, o ângulo entre a direção para onde o veículo $V1$ aponta (neste caso, coincide com seu vetor velocidade) e o veículo $V2$. Assim, temos as equações de medidas:

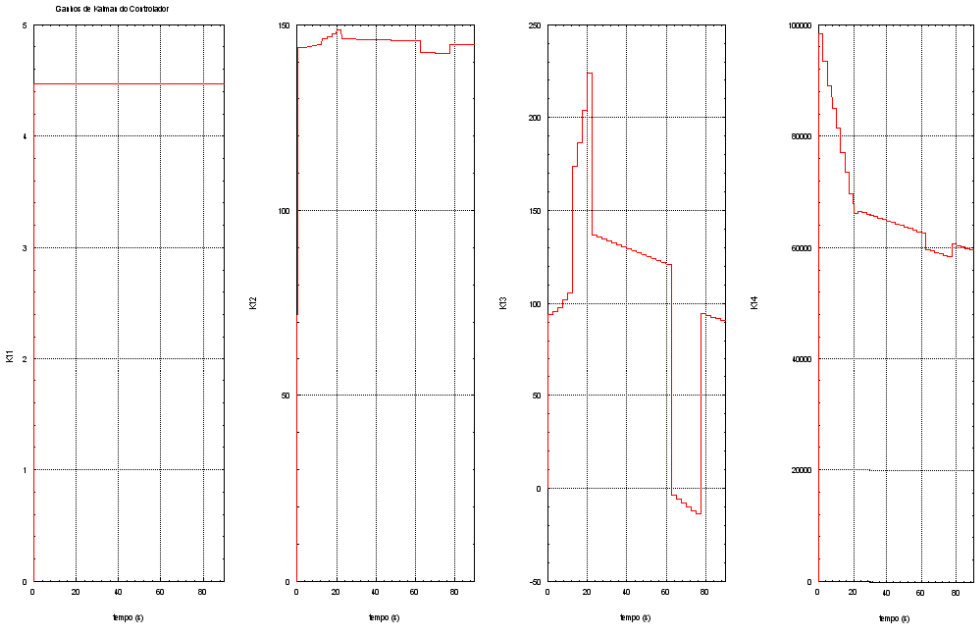


FIGURA 4.28 – Ganhos de Kalman do controlador do Modelo 3 em MATRIXx.

$$\underline{z}_k = \begin{bmatrix} r_{x_1} \\ r_{z_1} \\ v_{x_1} \\ v_{z_1} \\ \theta_v \end{bmatrix} = \begin{bmatrix} r_{x_1} \\ r_{z_1} \\ V_1 \cos\gamma \\ V_1 \sin\gamma \\ \arctan\left(\dfrac{r_{z_1}}{r_{x_2} - r_{x_1}}\right) + \gamma \end{bmatrix} + \underline{v}_k \qquad (4.71)$$

As seguintes condições iniciais foram utilizadas:

- Estado inicial (posições e velocidades de $V1$, posição e velocidade horizontais de $V2$): $\underline{x} = \begin{bmatrix} r_{x10} & r_{z10} & V_{10} & \gamma_0 & r_{x20} & v_{x20} \end{bmatrix}^T = \begin{bmatrix} 0 & 2500 & 250 & 0 & 35000 & 15 \end{bmatrix}^T$; em unidades do S.I.;
- Erro inicial na estimativa do estado: $\Delta\underline{x} = \begin{bmatrix} 9.8 & 14 & 1{,}2 & -0,0015 & 39 & 13 \end{bmatrix}^T$, em unidades do S.I.;
- Valor inicial da matriz de covariância: $P_{F0} = diag \begin{bmatrix} 30^2 & 30^2 & 3^2 & 0,03^2 & 30^2 & 3^2 \end{bmatrix}$
- Valor inicial da matriz de densidade espectral de potência do distúrbio:

$$Q_F = diag \left[\begin{array}{cccccc} 0 & 0 & 3^2 & (5e-2)^2 & 0 & 3^2 \end{array} \right]$$

Simularam-se as posições e velocidades de $V1$ e $V2$ exatas (estado real) para uma comparação com os valores estimados pelo filtro.

4.3.5 Simulações - MATLAB

Os resultados apresentados a seguir foram obtidos com 90s de simulação no ambiente de MIS MATLAB®, sendo os resíduos das medidas apresentados na Figura 4.29.

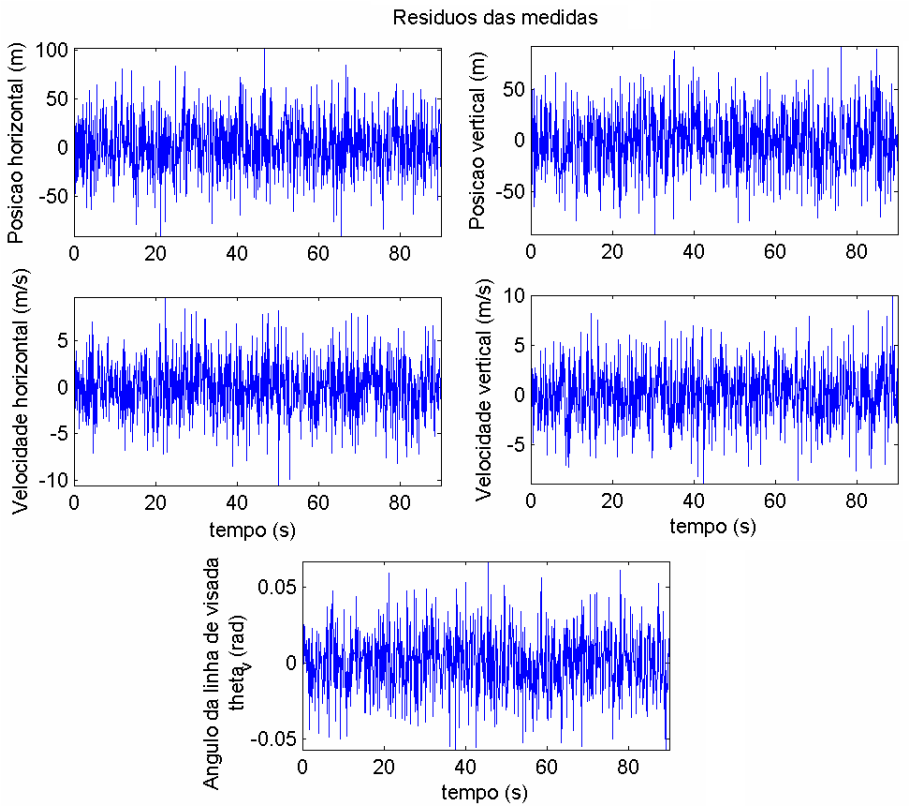


FIGURA 4.29 – Resíduos das medidas de V1.

Os erros das estimativas dos estados de $V1$ e $V2$ são apresentados nas Figuras 4.30 e 4.31.

Os estados estimados de $V1$ e $V2$ podem ser conferidos nas Figuras 4.32 e 4.33.

O controle utilizado para tais manobras foi limitado em um máximo de $4m_1g$, e pode ser visto na Figura 4.34.

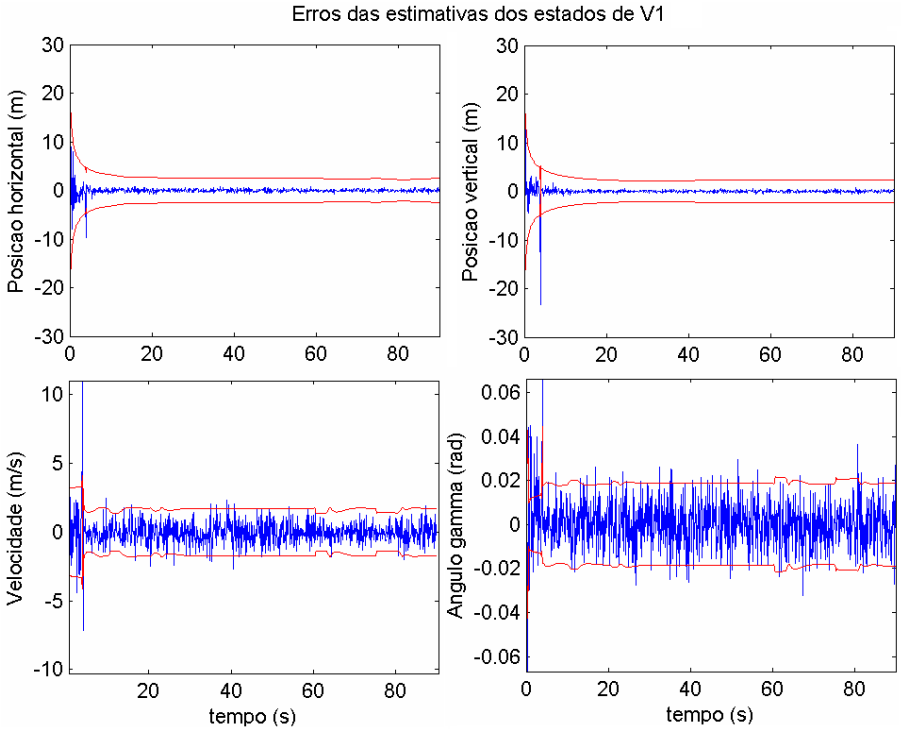


FIGURA 4.30 – Erros de estimação dos estados de V1.

4.3.6 Simulações - MATRIXx

Os resultados apresentados a seguir foram obtidos com 90s de simulação no ambiente de MIS MATRIXx®, sendo os resíduos das medidas apresentados na Figura 4.35

Os erros das estimativas dos estados de $V1$ e $V2$ são apresentados nas Figuras 4.36 e 4.37.

Os estados estimados de $V1$ e $V2$ podem ser conferidos nas Figuras 4.38 e 4.39.

O controle utilizado para tais manobras foi limitado em um máximo de $4m_1g$, e pode ser visto na Figura 4.40.

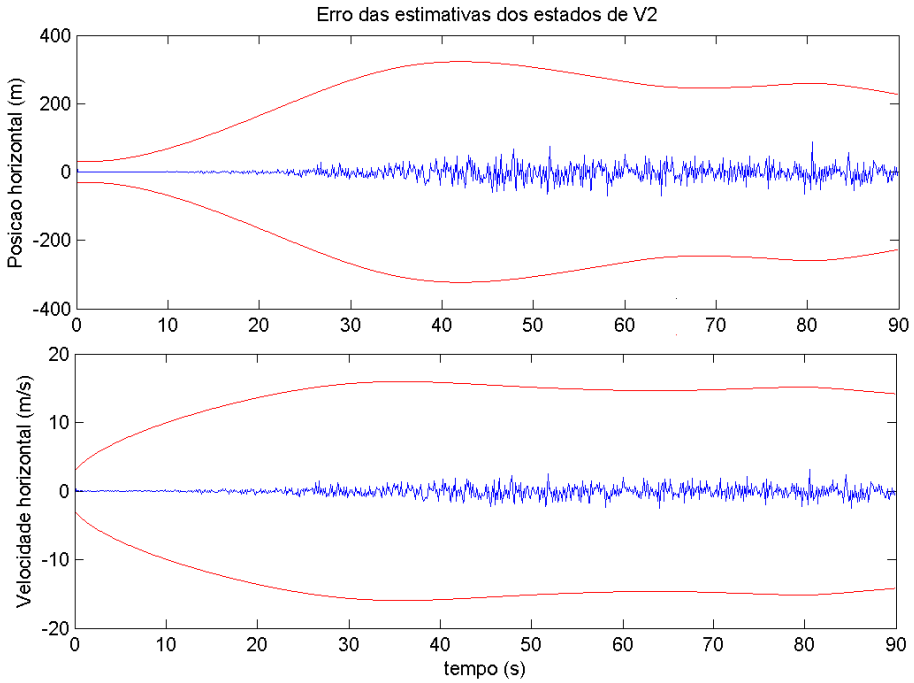


FIGURA 4.31 – Erros de estimação dos estados de V2.

4.3.7 Análise de Resultados

Modelo

O Modelo 3 apresenta um maior grau de realismo em relação ao Modelo 2. Com isto, fica mais evidente a limitação do controlador e do estimador para uma situação mais realista. Porém, conforme análise a seguir, o controlador se mostra eficiente e, embora o estimador não seja adequado para a estimação de estados do veículo rastreado, ainda assim é eficiente na estimação de estados do próprio veículo aeroespacial.

Controle

Para fazer a lei de controle de um veículo com mais graus de realismo, como o modelo 3, faz-se necessária forçar uma saturação do controle. Assim, as matrizes Q_R e R_R puderam ser ajustadas, por meio de tentativa e erro, de forma a fazer com que a resposta do controle fosse mais rápida, uma vez que não se permitiu passar do valor máximo de controle suportado pelo atuador (fator de carga de $4g$). A lei de controle foi eficiente, considerando principalmente que os principais estados a serem

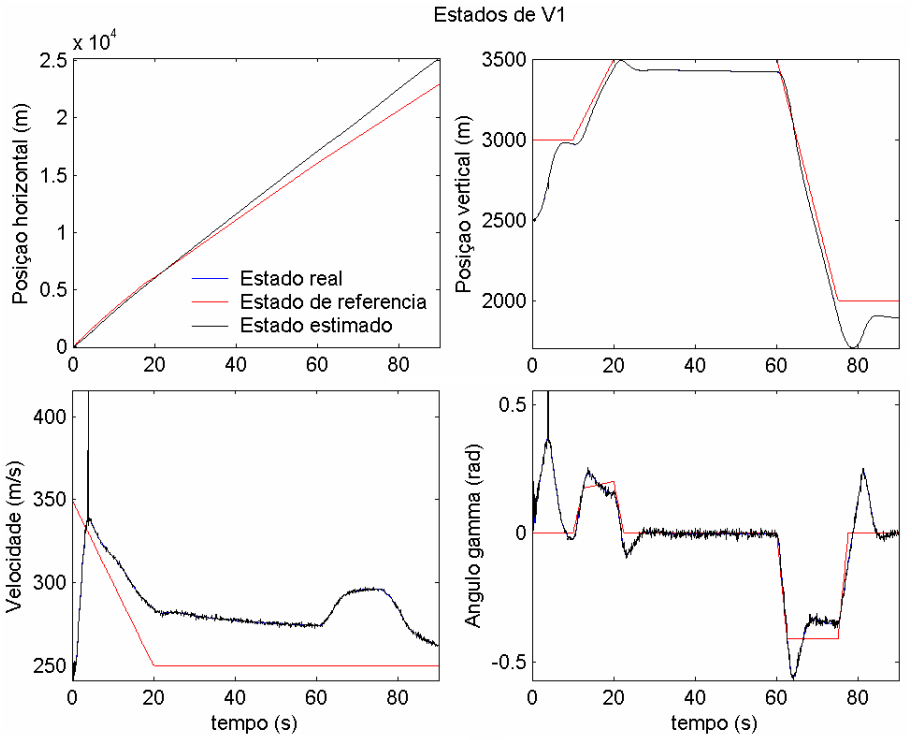


FIGURA 4.32 – Estados de V1, referência, com realimentação com estados reais, e com realimentação com estados estimados.

rastreados eram os estados relacionados à posição.

Estimação de Estados

Observa-se para este modelo mais completo, com um ângulo da linha de visada θ_v definido com mais rigor, que a estimação de estados do veículo rastreado $V2$ não foi adequada porque a covariância aumentou. O valor da covariância é de uma ordem de grandeza maior que o erro do estimador. Porém, as variáveis de estado do próprio veículo aeroespacial $V1$ foram estimadas com eficiência. Observa-se, neste caso, que a covariância dos estados V_1 e γ não foram monotonicamente decrescentes, e sim oscilaram em torno de valores aceitáveis da covariância. Isto se dá devido à natureza acoplada deste modelo, característica esta que era menos acentuada no Modelo 2.

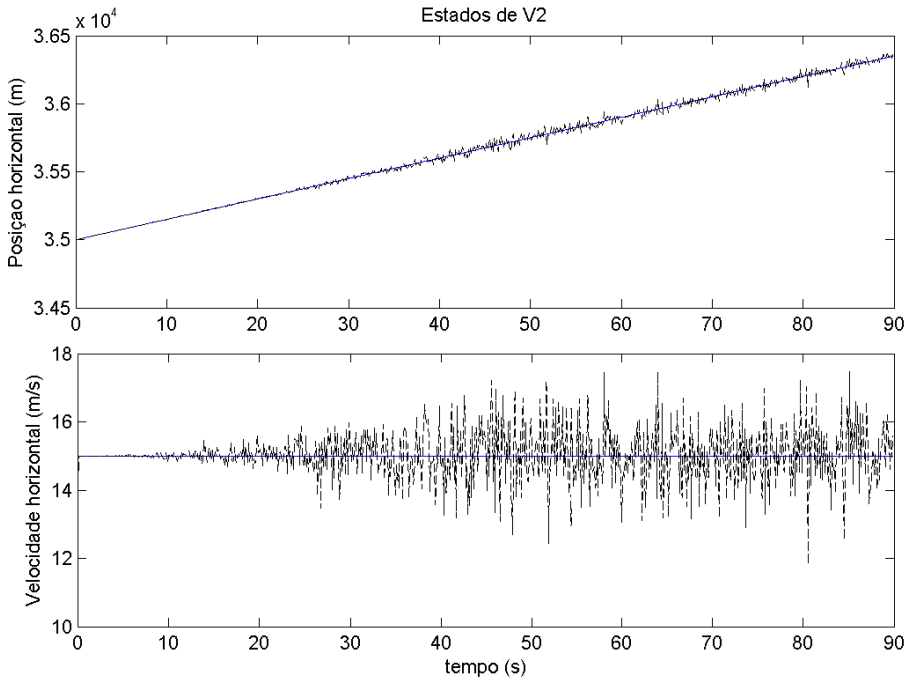


FIGURA 4.33 – Estados de V2, real e estimado.

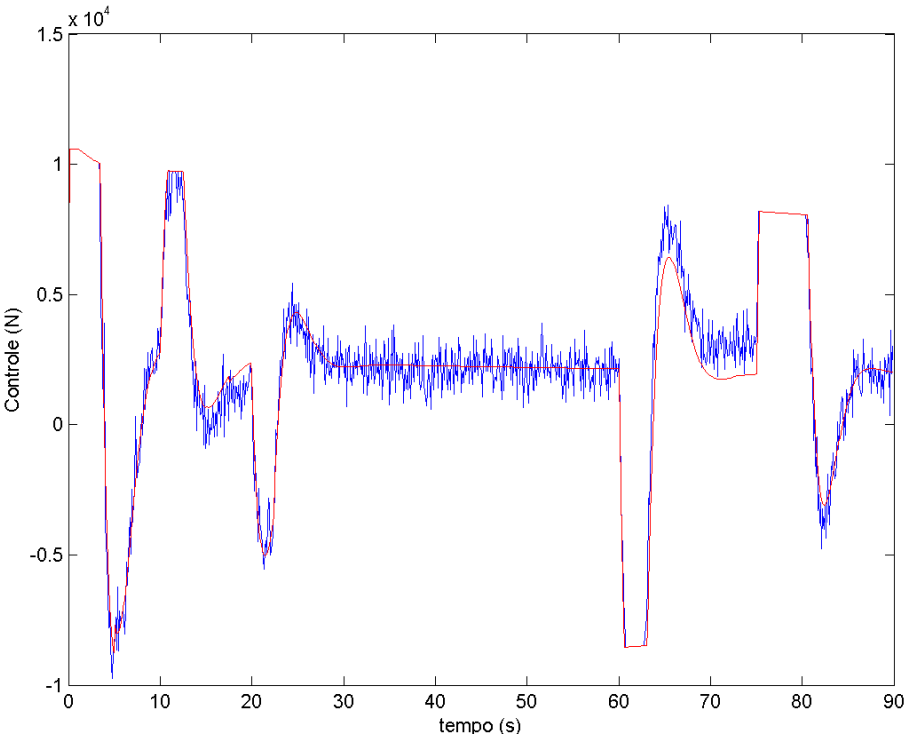


FIGURA 4.34 – Controle empregado pelo atuador para o rastreamento da trajetória.

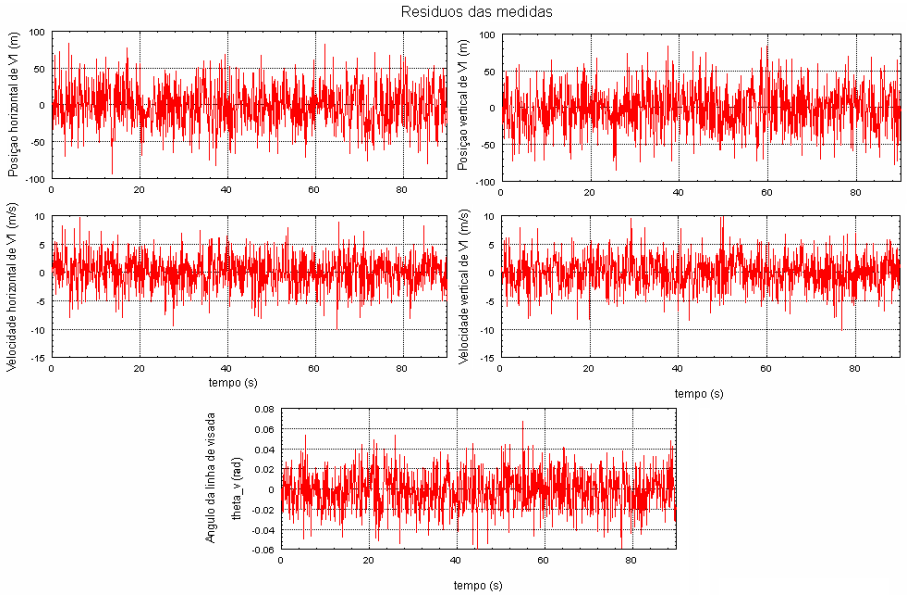


FIGURA 4.35 – Resíduos das medidas de V1.

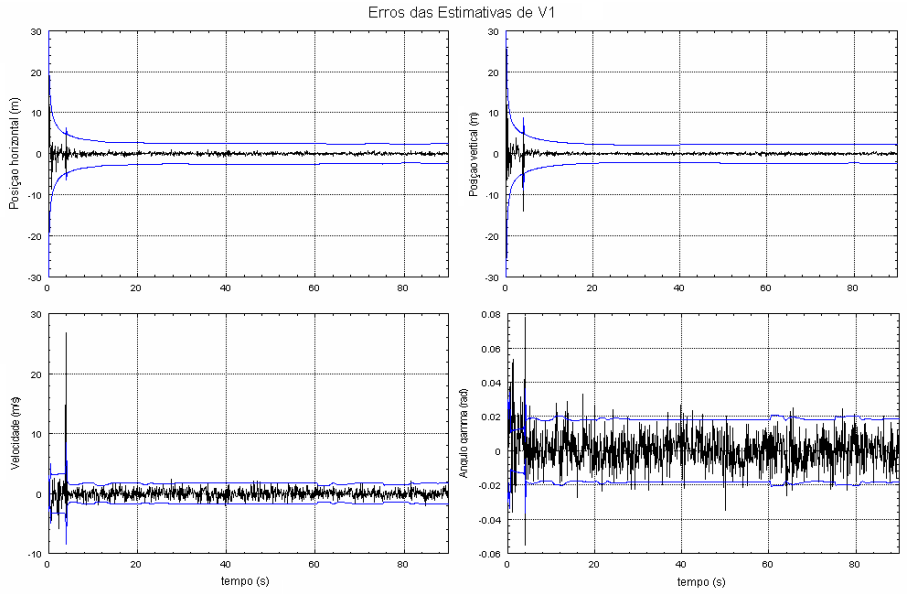


FIGURA 4.36 – Erros de estimação dos estados de V1.

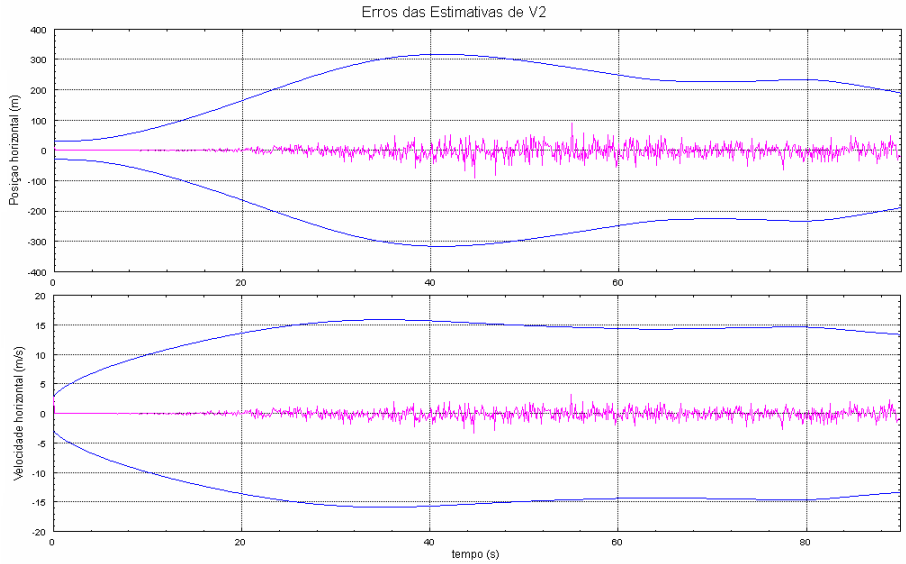


FIGURA 4.37 – Erros de estimação dos estados de V2.

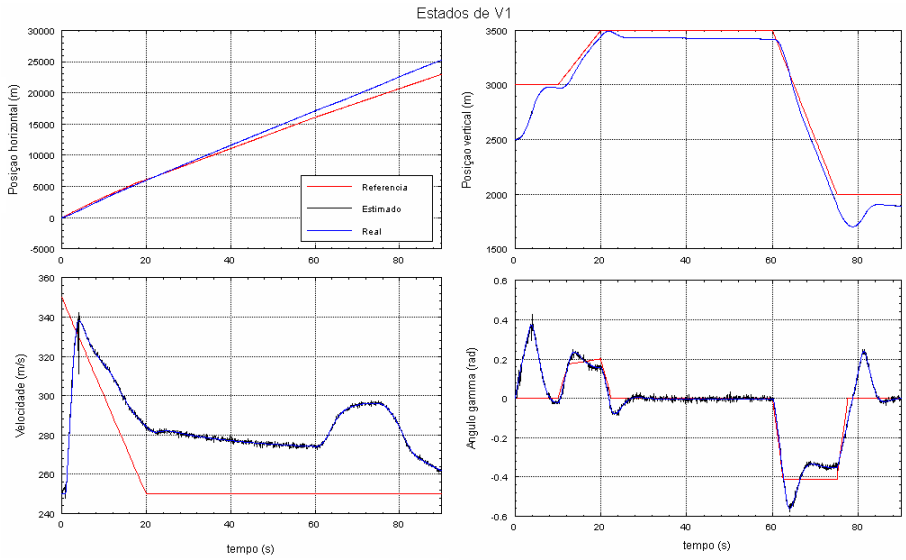


FIGURA 4.38 – Estados de V1, referência, com realimentação com estados reais, e com realimentação com estados estimados.

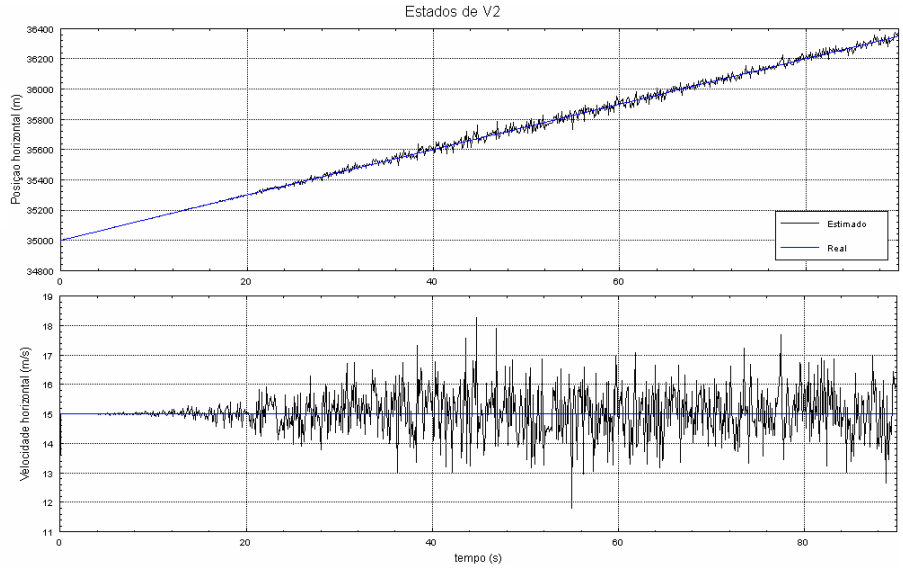


FIGURA 4.39 – Estados de V2, real e estimado.

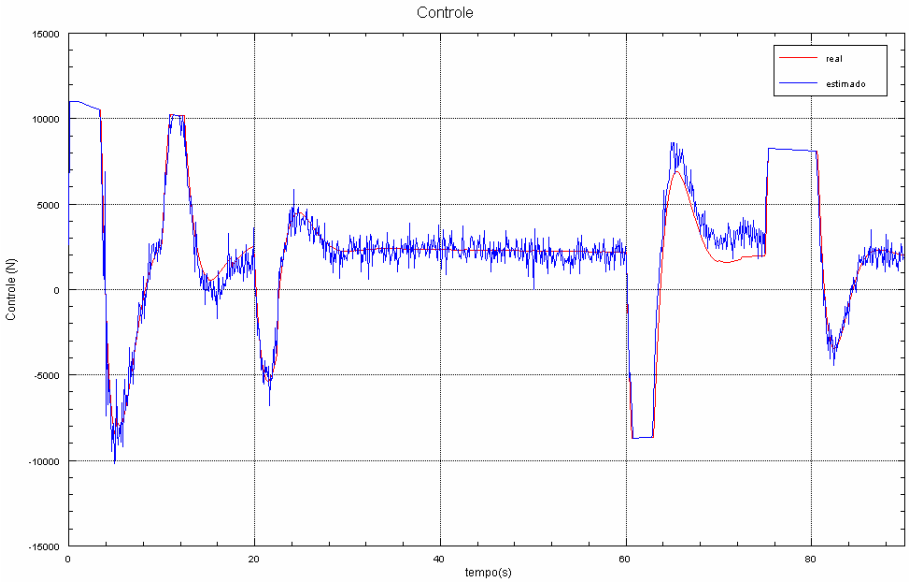


FIGURA 4.40 – Controle empregado pelo atuador para o rastreamento da trajetória.

CAPÍTULO 5
COMPARAÇÃO DOS AMBIENTES DE SIMULAÇÃO

Apresentamos aqui uma comparação dos ambientes de MIS utilizados neste trabalho. Como critérios de comparação escolhemos:

- Apresentação do ambiente
- Ambientes gráficos
- Tempo de processamento
- Facilidade para um novo usuário se adaptar ao ambiente
- Ferramentas específicas para engenharia

Tais critérios foram escolhidos por sua importância para uma aplicação semelhante à utilizada neste trabalho. A apresentação do ambiente é importante por ser a interface do usuário com o *software*, ou seja, a forma como as ferramentas estão dispostas no ambiente, e quais funcionalidades têm acesso mais fácil. Como a engenharia atualmente se utiliza bastante de recursos visuais, tais como diagramas de bloco, fluxo de sinais, etc, outro critério que se faz importante é o ambiente gráfico do *software*, assim como suas ferramentas disponíveis. Também analisamos o tempo de processamento para os modelos desenvolvidos neste trabalho. O tempo de processamento se faz importante porque é importante saber, no momento da simulação, se o tempo de processamento é um tempo real ou se é mais rápido ou lento que o processamento do computador de bordo. Outro fator que se mostra importante é a facilidade para um novo usuário se adaptar ao uso do ambiente, ou seja, quanto o ambiente é intuitivo, de fácil uso, e quanto o sistema de *help* é bem estruturado. Outro critério analisado aqui é a disponibilidade de ferramentas específicas para a engenharia, em particular para o problema aqui tratado.

5.1 Apresentação

O MATLAB® se inicia com a janela do módulo *MATLAB*, que é o módulo de interação com o usuário através de linhas de comando em linguagem *MATLAB*. Este módulo é constituído de:

- Janela de comandos e registro
- Histórico de comandos
- Diretório atual
- Janela de variáveis

A Figura 5.1 mostra os itens descritos anteriormente.

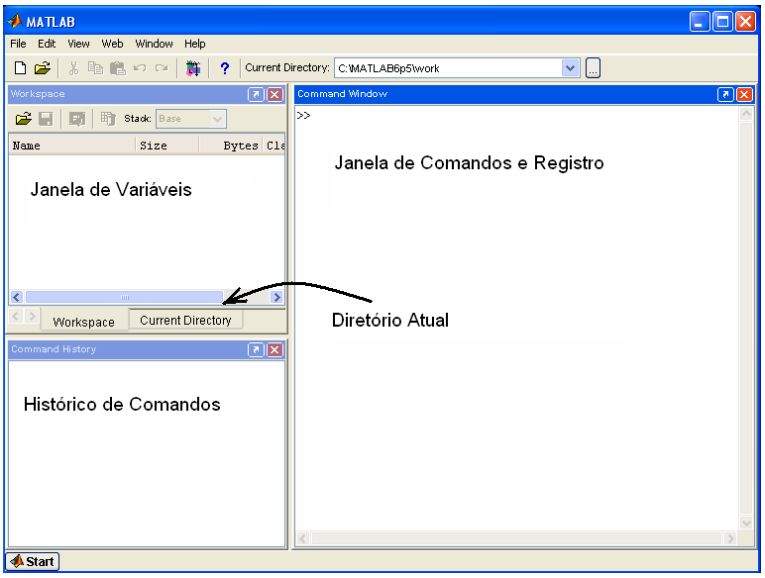


FIGURA 5.1 – Apresentação do ambiente MATLAB/ *Simulink*.

O módulo equivalente do MATRIXx® é o *Xmath*, que faz a interação do usuário com o ambiente através de linhas de comando, em linguagem *MathScript*, que será apresentada no próximo item. Sua janela inicial apresenta:

- Janela de comandos
- Janela de registros

A Figura 5.2 mostra os itens descritos anteriormente.

Embora o MATRIXx® não tenha as janelas do *MATLAB*, tem basicamente as mesmas funcionalidades, apenas acessadas de forma diferente:

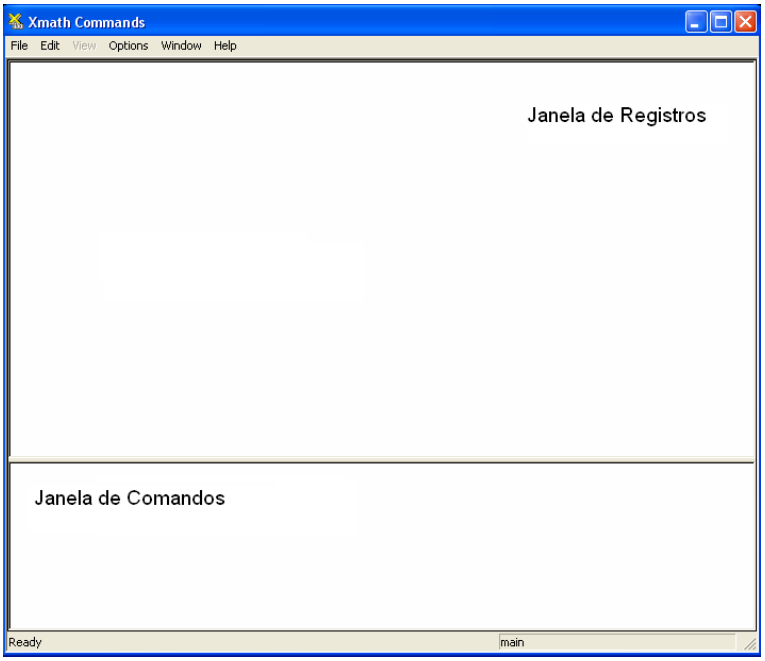


FIGURA 5.2 – Apresentação do ambiente MATRIXx/ *SystemBuild*.

- O histórico de comandos no MATRIXx® é recuperado através das teclas "Ctrl+↑". O histórico de comandos do MATLAB® também recupera comandos através de atalhos de teclas, com "↑". A diferença entre eles é que o MATLAB® recupera comandos de várias sessões anteriores, enquanto o MATRIXx® recupera comandos apenas da sessão atual.

- O diretório corrente do MATLAB® é aquele por onde se acessa mais facilmente arquivos do MATLAB®. As funções definidas pelos usuários somente são reconhecidas caso estejam no diretório corrente, ou através do comando "`path`". O diretório corrente do MATRIXx® é definido através do menu, ou então por linha de comando ("`set directory`"). Este diretório tem a função, basicamente, de facilitar encontrar um arquivo que se sabe estar em determinado diretório. Para se encontrar as funções definidas pelos usuários, define-se um caminho, através do menu ou de linha de comando ("`set path`"), que formará uma hierarquia de diretórios onde a função poderá ser achada, ou seja, a função não precisa estar, necessariamente, no mesmo diretório do arquivo que se deseja executar. O diretório corrente e o(s) caminho(s) especificado(s) podem ser facilmente recuperados através de comandos.

- A janela de variáveis do MATLAB® permite que se veja quais são as variáveis

reconhecidas pelo MATLAB® e suas propriedades, e permite que, clicando-se em cima do ícone de qualquer uma delas, abra-se uma janela de edição da variável, onde ela pode facilmente ser modificada. Tal funcionalidade encontra-se no MATRIXx® através de comando ("`who`"), que imprime na tela todas as variáveis reconhecidas e também suas propriedades. Não oferece janela de edição de variáveis, embora por linha de comandos seja possível modificar qualquer elemento das variáveis.

A apresentação do MATLAB® para a aplicação deste trabalho se revelou mais prática, haver mais ferramentas na tela principal, sendo mais fácil acessar dados e pastas de interesse. A tela principal do MATLAB® possui todas as ferramentas disponíveis na tela principal do MATRIXx®, com algumas adicionais. No entanto, a *National Instruments* revela planos para mudança da tela principal do MATRIXx® deixando à vista algumas funcionalidades importantes, como o acesso às variáveis.

5.2 Ambientes Gráficos

Como projetos de engenharia são comumente representados por diagramas de blocos, os ambientes MATLAB® e MATRIXx® contam com módulos que permitem a simulação da planta, assim como seu sistema de controle e estimação, de forma gráfica.

O ambiente gráfico do MATLAB® é o *Simulink*, que pode ou não trabalhar em conjunto com o *StateFlow*. O *Simulink* representa o sistema em diagrama de blocos, enquanto o *StateFlow* o faz por diagrama de fluxo de estados. Ambos os ambientes gráficos são a base para a estrutura que poderá gerar automaticamente o código em C, com o auxílio do *Real Time Workshop*, ou de uma documentação automática, com o *Simulink Report Generator*. A apresentação do *Simulink* pode ser vista na Figura 5.3, onde podemos ver o espaço para se montar os diagramas de blocos e as bibliotecas de blocos, que contêm não somente os blocos próprios do MATLAB® como também blocos acrescentados por outros módulos (*toolboxes*). Dentro de cada bloco pode-se determinar valores utilizados na simulação ou mesmo atribuir a eles uma variável, que será lida do módulo *MATLAB*. No caso de o mesmo bloco ser utilizado por várias partes do sistema, este poderá ser copiado e colado, tendo atribuídos a ele valores diferentes de variáveis, caso se faça necessário.

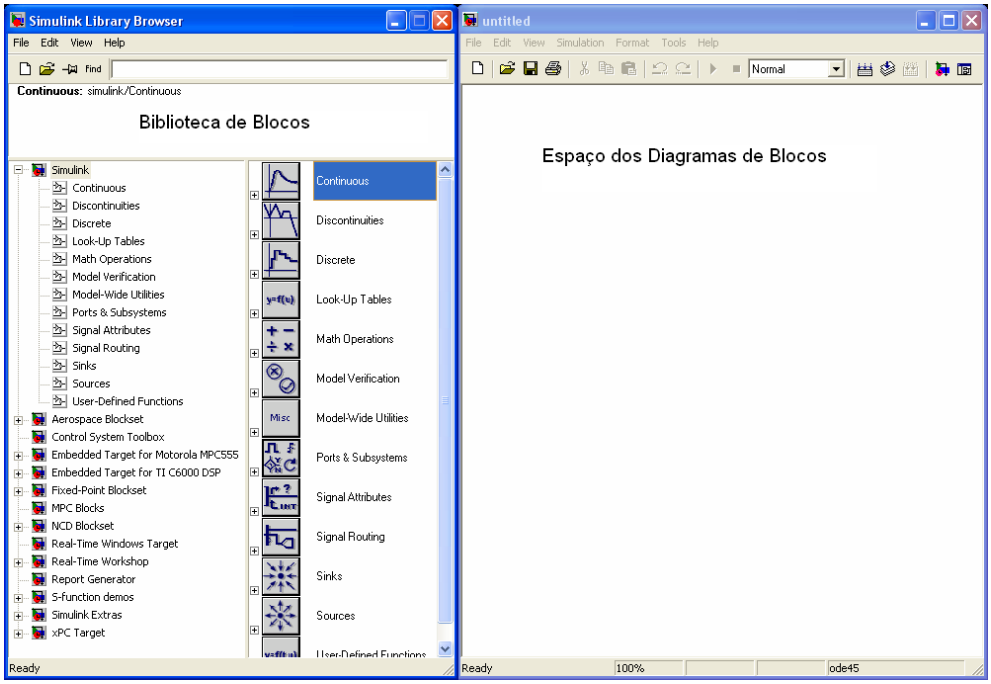


FIGURA 5.3 – Apresentação do módulo *Simulink*.

Analogamente, o MATRIXx® conta com o *SystemBuild*, que representa o sistema em diagrama de blocos. Assim como o MATLAB®, os blocos gerados no *SystemBuild* são a base utilizada pelo *Autocode* para gerar automaticamente o código em C ou Ada, além da documentação automática do projeto, feita através do *DocumentIt*. Sua apresentação pode ser verificada na Figura 5.4, onde podemos ver o espaço onde os blocos são montados, a biblioteca de blocos (chamada "Pallete") e uma janela onde os blocos são organizados. Existe uma hierarquia de blocos e superblocos no MATRIXx®. Estes blocos são organizados em ordem alfabética na janela de organização, onde pode-se verificar as propriedades de cada bloco, e utilizar um determinado bloco várias vezes em um mesmo sistema, sem ter que repeti-lo, apenas inserindo o bloco já pronto dentro de outro. Assim como no MATLAB®, a simulação pode ler um valor numérico inserido, ou pode ter variáveis atribuídas a parâmetros da simulação, armazenados no *Xmath*, ou mesmo valores numéricos.

Ao contrário do MATLAB®, o MATRIXx® pode ter estes dois valores atribuídos, procurando, assim o valor da variável no *Xmath* e, não o encontrando, utilizar o valor numérico inserido. No caso de um mesmo bloco ser utilizado várias vezes no mesmo sistema, porém com variáveis diferentes, pode-se utilizar o recurso chamado "`variable scope`", que define uma partição onde o valor deverá ser lido. Esta

partição é criada no *Xmath* através do comando "`new partition`"e acessado através de "`set partition`". A simulação busca o valor a ser usado na partição especificada e, caso a variável não esteja lá, ele busca na partição do bloco hierarquicamente superior, até que a encontre ou use o valor numérico atribuído no bloco.

Cada arquivo carrega não somente um sistema, mas tantos quantos forem salvos, ainda que de forma independente. Há possibilidade de carregar uma planta e só simular parte dela, ou, ainda, ter vários diagramas dentro de um arquivo e escolher somente um para fazer a simulação.

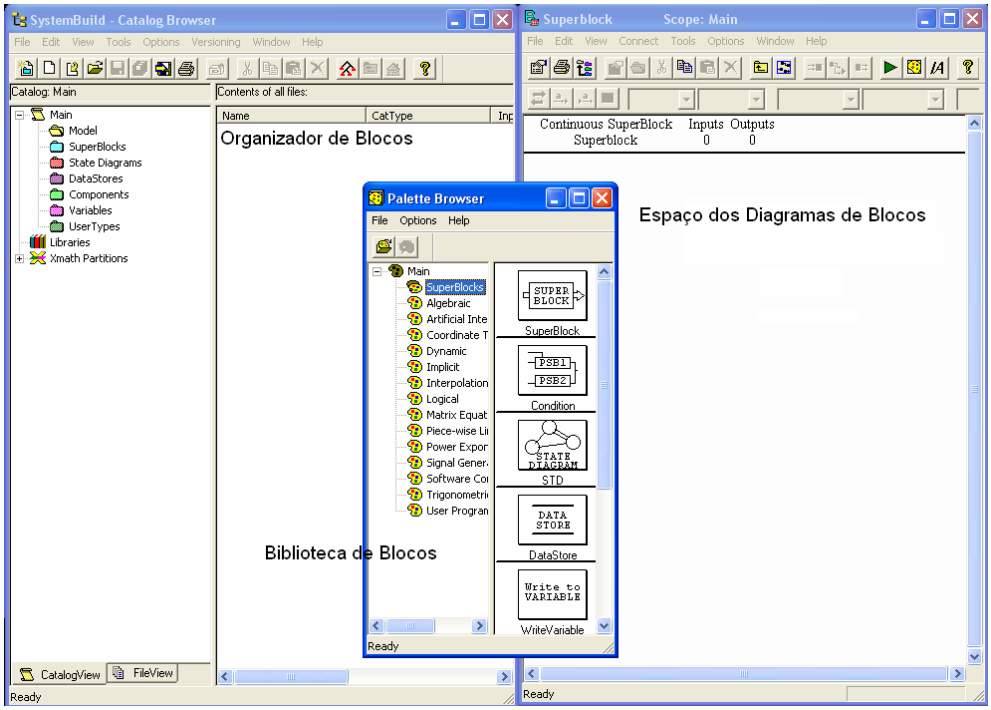


FIGURA 5.4 – Apresentação do módulo *SystemBuild*.

Há um módulo no MATLAB®, chamado *SB2SL* que foi criado com o objetivo de "traduzir"um diagrama de blocos do *SystemBuild* para o *Simulink*, porém este módulo não faz parte da licença disponível para o INPE e, portanto sua eficiência não foi analisada. O MATRIXx®não conta com um módulo que faça a operação inversa, ou seja, que transforme um arquivo do *Simulink* em arquivo do *SystemBuild*

Entre os ambientes gráficos do MATLAB® e do MATRIXx®, o *SystemBuild* se mostrou mais adequado para este trabalho. Embora a forma de montar os diagramas de blocos seja mais simples no *Simulink* do que no *SystemBuild*, a organização e

acesso aos blocos no *SystemBuild* ajuda o usuário a montar o diagrama de blocos, seja do bloco de menor hierarquia para o de maior hierarquia ou vice versa. A janela de simulação também oferece mais opções para a simulação que devem ser colocadas em blocos separados no *Simulink*, como a exportação dos vetores de saída para a área de trabalho e plotagem de gráficos.

5.3 Tempo de processamento

O tempo de processamento dos ambientes é feito, em ambos os casos, através de comandos que marcam o tempo de relógio no começo e no final da simulação a ser rodada, e depois se procedendo a subtração dos tempos para se obter o intervalo entre o começo e final da simulação. Tal comando para o MATLAB® é o "`cputime`", enquanto para o MATRIXx® o comando equivalente é o "`clock({cpu})`".

Para o modelo 2 o MATLAB® gastou 17,675s, enquanto o MATRIXx® gastou 14,641s. Para o Modelo 3 a diferença de tempos entre eles foi discrepante: enquanto o MATLAB® demorou 253,2640s, ou 4'13"264 milésimos, o MATRIXx® levou 646,48s, ou 10'46"48 milésimos. Tal diferença se dá por uma diferença nos algoritmos de controle ótimo. O MATRIXx® dá uma mensagem ao usuário para fazer o cálculo das matrizes de ganho ("*Residual is large - Riccati*") que é impressa na tela todas as vezes que tal condição acontece. O algoritmo do MATLAB® não dá este tipo de erro, portanto, não gasta tempo imprimindo tal informação na tela.

5.4 Facilidades para um novo usuário

Cada um destes ambientes possui uma apresentação e distribuição das ferramentas que são mais ou menos intuitivas, quando um usuário começa a trabalhar com os ambientes. Apresentamos nesta seção características de ambos os ambientes que são diferentes, embora possuam as mesmas funções.

5.4.1 Arquivos

Os ambientes MATLAB® e MATRIXx® utilizam arquivos para executar uma determinada seqüência de comandos, permitir que o usuário crie novas funções, ou

mesmo criar outros objetos, entre várias tarefas. Os principais tipos de arquivos que o MATLAB® usa são:

- .m: arquivo que pode definir uma seqüência de comandos que será executada como se estivesse sendo digitada na minha de comandos, ou uma função definida pelo usuário ou original do MATLAB®, ou ainda de alguma *toolbox*. Este arquivo deve ser escrito em linguagem MATLAB®, e é conhecido como *M-file*.
- .mat: arquivo que armazena variáveis a serem carregadas. Cria as variáveis na área de trabalho.
- .fig: arquivo que salva as figuras (gráficos) gerados pelo MATLAB® em formato próprio
- .mdl: arquivo do *Simulink*, ou seja, arquivos gráficos que representam diagramas de blocos
- .cdr: arquivos do *Stateflow*, ou seja, arquivos que representam diagramas de fluxo de estados
- .rtw: arquivo criado pelo *Real Time Workshop*, gerador de código do MATLAB®
- .c, .h: arquivos criados pelo *Real Time Workshop*, gerador de código do MATLAB®

Enquanto os arquivos equivalentes do MATRIXx® são:

- .ms: arquivo que contém uma seqüência de comandos a ser seguida pelo *Xmath*, como se tivessem sido digitados na janela de comandos. Arquivo escrito em linguagem *MathScript*.
- .msf: arquivo que define uma função criada pelo usuário. Arquivo escrito em linguagem *MathScript*. Seu executável tem extensão .xf.
- .mso: arquivo utilizado para definir um objeto. Arquivo escrito em linguagem *MathScript*. Objetos são um tipo especial de arquivos, que reúne uma coleção de definições de funções, comandos ou mesmo novos operadores.

- .msc: arquivo que permite o usuário criar comandos para o MATRIXx®. Seu executável tem extensão .xc.
- .dat: arquivos do *SystemBuild*, ou seja, arquivos gráficos que representam diagramas de blocos.
- .c, .ada: arquivos criados pelo AutoCode, gerador de código do MATRIXx®.

5.4.2 Editores

O MATLAB® conta com um editor para seus programas e funções criadas pelos usuários, que já contém em si um depurador com todos os elementos necessários para seu uso. O editor é de fácil utilização, bastante intuitivo, oferecendo ferramentas, tais como possibilidade de se comentar um determinado trecho do programa, interação com o módulo *MATLAB*. Os erros encontrados no depurador são explicitados na janela de comandos do próprio *MATLAB*.

O MATRIXx® não conta com um editor próprio, podendo, porém, ter seus programas e funções editados em qualquer editor de texto. Se no momento de rodar o programa ou fazer uso das funções um problema é detectado, uma janela de depuração é aberta, onde os erros são apontados, com a possibilidade de se corrigir, salvar e rodar novamente o programa através desta janela. O depurador do *Xmath* cria uma partição onde as variáveis para depuração são criadas, e então, ao fim da sessão de depuração, esta partição é apagada, e o usuário continua utilizando a partição de seu programa. Conta com um recurso chamado "`set watch`"que avisa ao usuário quando o valor de determinada variável é mudado. Como seu editor não tem linhas numeradas, os pontos são marcados através de grifos das variáveis que se deseja observar.

5.4.3 Apresentação de resultados

Neste trabalho a apresentação dos resultados foi analisada, principalmente, de forma gráfica. Constata-se aqui uma grande diferença entre eles não somente na forma de plotar os gráficos, mas também de editá-los.

O MATLAB® permite que vários gráficos sejam plotados lado a lado com simples

comandos, já no formato desejado para apresentação final. Depois de prontos os gráficos, permite que partes dele sejam analisadas através de uma ferramenta de lupa, além de permitir que o gráfico seja salvo em formato próprio, ou em formatos de figuras difundidos comumente, tais como .jpg, .emf, .bmp, .png, .eps, entre vários outros. Observe-se que nem sempre a exportação para outros formatos fica da forma apresentada na figura.

O MATLAB® trata gráficos como objetos dentro do ambiente. Assim, os comandos para se plotar os gráficos obedecem ao mesmo padrão de outros objetos. A edição de legendas, títulos e *labels* é feira posteriormente, através da seleção do texto a ser editado. O *zoom* dos gráficos é feito através da plotagem de outro gráfico, especificando o intervalo de interesse. Permite que as figuras sejam salvas da mesma forma que se salvam outros tipos de dados em um arquivo. A exportação de figuras é feita em formato .ps, ou utilizando comando próprio ("`hardcopy`") para ser exportado para outros formatos, a saber: .eps, .hpgl, .pict e .emf.

5.4.4 Autocoerência

Embora ambos os ambientes tenham funcionalidades diferentes, são abertos e intuitivos. Entretanto, embora o MATLAB® permita que um leigo utilize sua estrutura, o MATRIXx® exige que se tenha um conhecimento prévio, não somente do próprio ambiente, mas também de fundamentos de engenharia. Assim, embora um usuário leigo consiga construir um sistema em MATLAB®/ *Simulink*, os problemas virão à tona no momento de se simular a dinâmica do sistema. O MATRIXx® tem uma estrutura tal que um leigo não consegue apenas "montar"um diagrama de blocos sem sentido físico, uma vez que o MATRIXx®/ *SystemBuild* exige que o diagrama de blocos tenha coerência ao ser montado. Uma facilidade oferecida pelo *SystemBuild* é que os nomes dos blocos são propagados, facilitando a montagem dos blocos, evitando que erros sejam cometidos por distração.

5.4.5 Ajuda

A ajuda do MATLAB® é bastante fácil e intuitiva, sendo que o usuário consegue acessar o tema de seu interesse por vários caminhos.

Para se obter ajuda no MATRIXx®, é necessário ter um conhecimento prévio do ambiente ou da função na qual se tem dúvida. O sistema de ajuda se apresenta em tópicos dentro de cada um dos módulos do ambiente (*Xmath, SystemBuild, AutoCode, DocumentIt, etc.*) ou por ordem alfabética dos comandos.

Para se fazer este trabalho, foi necessário um aprendizado de ambos os ambientes. Houve uma tentativa de se aprender ambos os ambientes ao mesmo tempo. Porém, observou-se que, dedicando o mesmo tempo ao aprendizado de ambos, o progresso no ambiente MATLAB® foi muito maior que o ambiente MATRIXx®. Depois de um tempo estudando apenas o MATLAB®, quando havia um domínio básico deste ambiente, o estudo do MATRIXx® foi retomado. Então, o progresso neste ambiente foi bastante rápido, até que o domínio de ambos fosse comparável. Portanto, enquanto o MATLAB® se faz mais intuitivo para um usuário leigo em termos de ambientes de MIS, o aprendizado do MATRIXx® é muito mais rápido quando já se domina algum outro ambiente. Sua estrutura permite que o usuário cometa menos erros quando o uso do ambiente é dominado.

5.5 Ferramentas para engenharia

Para a utilização específica desde trabalho, faz-se importante uma análise das ferramentas existentes que facilitam seu uso para engenharia. Aqui analisamos algumas delas, tendo em vista que o MATLAB® foi criado a partir de um projeto que visava criar ferramentas para a resolução de problemas matemáticos comuns em várias áreas do conhecimento. Com seu uso constante em projetos de engenharia, voltou-se para esta área, porém crescendo em outras direções, como matemática financeira. Esta origem é refletida na presenças das *toolboxes*, que são programas escritos em linguagem *MATLAB*, que complementam as funções do MATLAB® em várias áreas do conhecimento. Enquanto isso, o MATRIXx® foi construído visando a comunidade de engenharia, mais especificamente controle. Portanto, ele já foi estruturado de forma a atender este público, reconhecendo objetos de engenharia como sistemas lineares, e priorizando o reconhecimento e computação destes elementos aos outros tipos de objetos encontrados no sistema.

5.5.1 Linguagem

Ambos os ambientes, MATLAB® e MATRIXx®, têm seu código fonte escrito em linguagem C.

O MATLAB® contém em sua estrutura uma linguagem de programação chamada também *MATLAB*. É uma linguagem procedural, embora de nível maior do que linguagens convencionais, como C e C++. Funciona como uma seqüência de comandos digitados na própria tela do MATLAB®, e podem ser salvos como *M-files*, já descritos anteriormente. Também é utilizada para definir novas funções no MATLAB®.

Assim como o MATLAB®, o MATRIXx® tem em sua estrututa uma linguagem de programação própria, chamada *MathScript*. É uma linguagem parecida com a linguagem *MATLAB*, funcionando também como comandos digitados na janela de comandos. Utilizada para definir novos comandos, novas funções e novos objetos, além de utilizada nos diagramas de blocos.

5.5.2 Funções

Ambos os ambientes, MATLAB® e MATRIXx® possuem funções pré-programadas nas suas bibliotecas, envolvendo rotinas matemáticas mais simples, desde cálculos algébricos, até rotinas para cálculos de equações diferenciais e rotinas para lidar com sistemas de engenharia.

Tem-se, no MATLAB®, além das funções que podem ser acrescentadas pelo usuário em FORTRAN, C e C++, conta-se ainda com um grande número de *toolboxes*, que são "anexos"do MATLAB®, construídos por terceiros, que definem funções e blocos de áreas específicas do conhecimento. As funções em FORTRAN, C e C++ podem ser agregadas em diagramas de blocos em formatos de *"S-functions"*, que são funções com formato especificado e regras de construção, podendo ser em uma das linguagens já citadas ou em linguagem *MATLAB*, que depois podem ser embutidas em blocos do *Simulink*. As funções criadas pelos usuários podem ser utilizadas nos diagramas de blocos através do bloco "*MATLAB functions*".

O MATRIXx® não possui a possibilidade de se agregarem *toolboxes* a ele, embora

suporte funções criadas pelo usuário também em FORTRAN, C e C++, inclusive dentro dos blocos. Há um bloco especial para isto, que suporta rotinas inteiras de programa nas linguagens anteriormente citadas. Há, também, possibilidade de se embutir nos diagramas de blocos funções inteiras em linguagem *MathScript*, através do bloco "*Block Script*".

Importante observar que estes blocos, em nenhum dos dois ambientes, é codificado ao se gerar automaticamente um código.

5.5.3 Tensores

O MATLAB® reconhece tensores, ou seja, matrizes n-dimensionais que armazenam, por exemplo, matrizes a serem utilizadas em cada instante de tempo. Há, porém, uma dificuldade em plotar estes dados, o que pode ser contornado pelo usuário, plotando vários gráficos com determinados elementos em particular.

No MATRIXx® tal armazenamento é feito através das *PDM*'s, ou Matrizes Dependentes de Parâmetros, que armazenam em uma estrutura não numérica qualquer tipo de dados, entre eles, matrizes. Assim, criam-se tensores como no MATLAB®, apenas com uma forma diferente de acessá-los. As *PDM*'s apresentam a vantagem de armazenar não somente dados numéricos, mas também objetos, tais como sistemas lineares ou funções de transferência.

5.5.4 Discretização de Malha

O MATLAB® permite, no seu módulo *Simulink*, que a discretização de uma malha contínua seja feita através da inclusão de um bloco de *Sample & Hold*, que deve ter sua constante de tempo ajustada para a menor constante de tempo envolvida no sistema.

No MATRIXx® tal discretização é feita automaticamente, e o *SystemBuild* ajusta automaticamente a constante de tempo, de forma que o usuário somente seleciona a opção de discretização, ficando o ambiente por conta dos ajustes.

No caso das ferramentas para engenharia, temos uma maior opção de ferramentas no

MATLAB®. Isto se dá pelo fato de o MATLAB® permitir integração das *toolboxes* à sua estrutura principal. Porém, como o MATRIXx® tem sua estrutura criada especificamente para a engenharia, seus algoritmos são mais robustos do que os apresentados no MATLAB®. Cabe ao usuário, portanto, escolher entre um e outro, dependendo da aplicação que ele deseja.

CAPÍTULO 6

COMPARAÇÃO ESPECÍFICA DOS AMBIENTES DE MIS

Neste Capítulo são apresentadas diferenças e semelhanças de ambos os ambientes, MATLAB® e MATRIXx®, constatadas na simulação de cada um dos três modelos apresentados.

6.1 Modelo 1

MATLAB® e MATRIXx® têm sua estrutura de matrizes e vetores muito parecidas. A única diferença é na separação de elementos da mesma linha, que no caso do MATLAB® é feito com espaços e no caso do MATRIXx® é feito com vírgulas.

O MATLAB® conta com 3 comandos que servem para "limpar"a área de trabalho: "`clear all`", que apaga todas as variáveis, "`close all`", que fecha todos os gráficos abertos, e "`clc`", que limpa a janela de comandos. O MATRIXx® conta com as mesmas funcionalidades, utilizadas de forma diferente. O comando "`delete *.*`"apaga todas as variáveis criadas. Porém, no caso de não haverem variáveis, ele indica erro. A limpeza das janelas de comando e histórico pode ser feita através de opções do menu, "`clear command`"e "`clear log`", enquanto o gráfico deve ser fechado da forma convencional.

A chamada de outros arquivos pode ser feita de forma bastante diferente. Enquanto no MATLAB® basta digitar o nome do arquivo, desde que ele esteja no diretório corrente, no MATRIXx® é necessário um comando ("`execute file`"), acompanhado do endereço do arquivo e seu nome. Porém, ambos possuem o recurso no menu, permitindo que a chamada seja feita de forma gráfica.

As variáveis aleatórias também dispõem de operadores diferentes nos dois ambientes, embora produzam o mesmo resultado. Enquanto o MATLAB® possui dois comandos para se gerar variáveis aleatórias ("`rand`"e "`randn`"para distribuições uniforme e normal, respectivamente), o MATRIXx® possui apenas um, sendo o tipo de distribuição especificado pelo usuário, através do comando "`set distribution`". Os comandos do MATLAB® sem especificação de tamanho da variável são interpretados como escalares, enquanto o comando do MATRIXx® exige uma especificação

da dimensão da variável.

A forma de se plotar gráficos também é bastante diferente. O MATRIXx® trata seus gráficos como objetos, enquanto o MATLAB® dá um tratamento diferenciado. Assim, quando se salvam os dados da área de trabalho do MATRIXx®, os gráficos são salvos juntos. No MATLAB®, apenas as variáveis são salvas, podendo os gráficos serem salvos no formato *.fig* ou exportados como figuras.

Ao se plotar mais de um gráfico no MATLAB®, ele substitui a última janela de gráfico pelo atual, ou, no caso de um novo gráfico ("`figure`"), abre mais uma janela. O MATRIXx® atribui um nome de variável àquele gráfico, exibindo-o quando a variável é chamada. Enquanto no MATLAB® pode-se abrir mais de uma janela de gráficos ao mesmo tempo, posicionando-as no monitor de forma a comparar os gráficos (mantendo, no entanto, apenas uma janela ativa), o MATRIXx® permite que apenas um gráfico seja aberto de cada vez.

Desejando-se plotar gráficos lado a lado, ambos os ambientes têm sua funcionalidade específica. O MATLAB® o faz através do comando "`subplot`", enquanto o MATRIXx® posiciona o gráfico com os comandos "`row`"e "`column`", colocados nas especificações da plotagem.

Quanto aos ambientes gráficos, foram encontradas as mesmas funções, apenas em formatos diferentes.

Os blocos do MATRIXx®/*SystemBuild* são muito mais expressivos do que os blocos do MATLAB®/*Simulink*. A visualização dos bloco pode ser de cinco tipos (simples, especial, do usuário, alternativo e padrão), fornecendo a possibilidade de se enxergar através dos Superblocos (blocos que contém outros blocos), fazendo com que o usuário já saiba quais os blocos e superblocos que estão dentro. Além disso, informa no próprio bloco se ele é contínuo ou discreto. Os blocos do MATLAB® são apenas retângulos simples, com nomes de saídas e entradas. Exemplos de ambos os tipos de blocos podem ser verificados nas Figuras 6.1 e 6.2.

Além disso, o MATRIXx®/*SystemBuild* conecta linhas que saem de um mesmo bloco para outro bloco em comum em uma única linha, deixando o diagrama de blocos mais limpo.

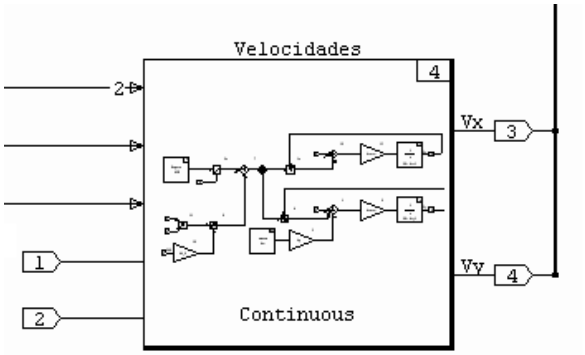


FIGURA 6.1 – Exemplo de bloco do MATRIXx.

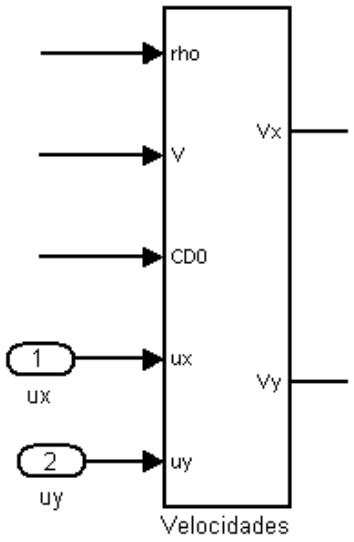


FIGURA 6.2 – Exemplo de bloco do MATLAB.

O simulador de tempo do MATLAB® é um bloco pronto, que pode apenas ser inserido no diagrama de blocos. O MATRIXx® utiliza um bloco para comandos, o *"BlockScript"* onde o algoritmo de simulação de tempo (muito simples) é feito. Estes blocos podem ser vistos nas Figuras 6.3 e 6.4, respectivamente.

O vetor de tempo de simulação tem seu princípio e seu final determinados no MATLAB®, sendo os valores intermediários dependentes do tipo de integrador que se utiliza. No MATRIXx®, o vetor de tempo é determinado pelo usuário, e inserido como um dado da simulação. Os resultados no MATLAB® podem ser enviados para a área de trabalho através de bloco próprio, e um osciloscópio utilizado para se verificar o andamento da simulação.

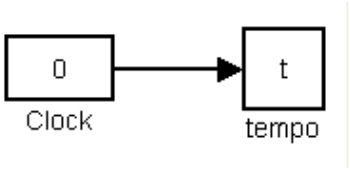


FIGURA 6.3 – Simulador de tempo do MATLAB.

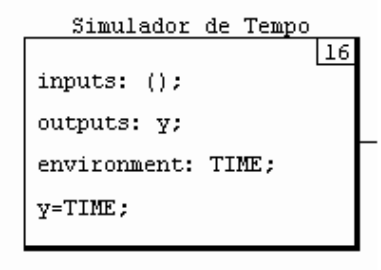


FIGURA 6.4 – Simulador de tempo do MATRIXx.

No MATRIXx® existe uma funcionalidade na janela de simulação que permite que os dados da simulação sejam exportados para a área de trabalho, na forma de PDM´s, e há uma opção para se plotar os gráficos de resultados de todas as saídas assim que a simulação termina.

6.2 Modelo 2

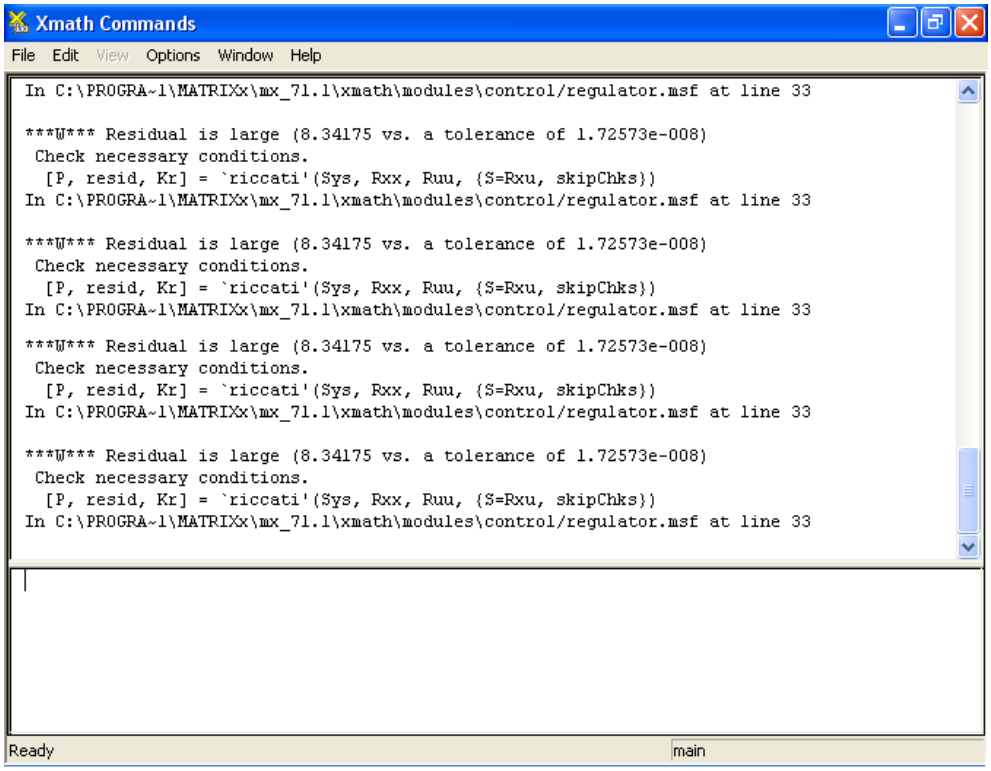
No Modelo 2 foi utilizada pela primeira vez a funcionalidade das PDM´s, as matrizes dependentes de parâmetros do MATRIXx®. Embora o MATLAB® não conte com a ferramenta das PDM´s, em todas as situações testadas durante este trabalho, tais matrizes foram substituídas com eficiência pelos tensores, disponíveis no MATLAB® e não disponíveis no MATRIXx®. No caso da necessidade de se utilizar algum dado inserido em uma PDM, há um comando próprio("`makematrix`") que transforma o dado em um objeto numérico como os outros.

Outro comando explorado neste modelo foi o comando utilizado para arredondamentos. Ambos os ambientes possuem exatamente as mesmas opções de arredondamento. O MATLAB® possui um comando para arredondar para o número mais próximo ("`round`"), arredondar para cima ("`ceil`") ou para baixo ("`floor`"). O MATRIXx® possui apenas um comando para todas as opções ("`round`"), cujo parâmetro padrão é arredondar para o mais próximo ("`nearest`"), podendo também realizar o arredon-

damento para cima ("`up`") ou para baixo ("`down`"), parâmetros estes especificados nas opções do comando.

Nos ambientes gráficos as implementações foram feitas de forma diferente. No MATLAB® */Simulink* foi utilizado o bloco chamado "MATLAB Function", que permite que uma função do MATLAB® seja utilizada dentro de um bloco. Há um bloco equivalente no MATRIXx® *SystemBuild*, chamado "MathScript Block", que permite que as funções do MATRIXx® sejam utilizadas em blocos. Porém, nenhum destes dois blocos é codificável pelo gerador automático de código do respectivo ambiente, embora isto não fique muito claro no MATLAB®. O MATRIXx® deixa bem claro que tal bloco não é codificável. Assim, na hora de montar o diagrama de blocos no MATLAB®, foi utilizado um bloco "MATLAB Function"contendo a função "`lqr`", enquanto que no MATRIXx® os valores de ganhos do regulador foram calculados no *Xmath* e armazenados em um bloco chamado "*Gain Scheduler*", que, como o próprio nome diz, escolhe a tabela de ganhos a ser utilizada para cada instante de tempo.

Os próprios comandos para se calcular os ganhos do Regulador Linear Quadrático são, em sua essência, diferentes. O comando do MATLAB® para se calcularem os ganhos do controlador regulador linear quadrático é o "`lqr`", que é calculado a partir das matrizes *A, B, Q, R* fornecidas, onde A e B são as matrizes do sistema linear e Q e R são as matrizes de peso, respectivamente, do estado e do controle. O comando equivamente do MATRIXx® é o "`regulator`", que trabalha com dados mais completos. A entrada para esta função é um sistema dinâmico, além das matrizes Q e R. Este sistema linear pode ser definido de várias formas, entre elas, matrizes *A, B, C, D* de um sistema linear, função de transferência ou um objeto que caracterize um sistema linear. Há possibilidade, ainda, de, no sistema dinâmico, se especificar as condições iniciais de tal sistema. Assim, o MATRIXx® considera todo o sistema a ser controlado, e não somente as matrizes da dinâmica. Isto se deve à natureza do MATRIXx®, que é intrinsecamente voltada para controle. Daí as diferenças de ganhos de alguns componentes da matriz de ganhos dos controladores. Esta diferença faz com que os algoritmos de cálculo dos ganhos seja diferente, de forma que o cálculo que é feito no MATLAB® normalmente, no MATRIXx® aparecem alguns *warnings*, alertando quanto a possíveis erros numéricos devidos ao sistema a ser controlado, conforme a Figura 6.5.



```
In C:\PROGRA~1\MATRIXx\mx_71.1\xmath\modules\control/regulator.msf at line 33

***W*** Residual is large (8.34175 vs. a tolerance of 1.72573e-008)
 Check necessary conditions.
  [P, resid, Kr] = `riccati'(Sys, Rxx, Ruu, {S=Rxu, skipChks})
In C:\PROGRA~1\MATRIXx\mx_71.1\xmath\modules\control/regulator.msf at line 33

***W*** Residual is large (8.34175 vs. a tolerance of 1.72573e-008)
 Check necessary conditions.
  [P, resid, Kr] = `riccati'(Sys, Rxx, Ruu, {S=Rxu, skipChks})
In C:\PROGRA~1\MATRIXx\mx_71.1\xmath\modules\control/regulator.msf at line 33

***W*** Residual is large (8.34175 vs. a tolerance of 1.72573e-008)
 Check necessary conditions.
  [P, resid, Kr] = `riccati'(Sys, Rxx, Ruu, {S=Rxu, skipChks})
In C:\PROGRA~1\MATRIXx\mx_71.1\xmath\modules\control/regulator.msf at line 33

***W*** Residual is large (8.34175 vs. a tolerance of 1.72573e-008)
 Check necessary conditions.
  [P, resid, Kr] = `riccati'(Sys, Rxx, Ruu, {S=Rxu, skipChks})
In C:\PROGRA~1\MATRIXx\mx_71.1\xmath\modules\control/regulator.msf at line 33
```

FIGURA 6.5 – Mensagem de aviso do MATRIXx.

O *debugger* do MATLAB® apresentou-se bastante eficaz. No caso de erro, o MATLAB® apresenta em sua tela principal, na janela de comandos, o arquivo que contém o erro, assim como a linha onde tal erro se encontra. Clicando-se neste aviso, o usuário já é remetido diretamente ao ponto de erro, já de posse da origem do erro, para consertá-lo, conforme mostrado na Figura 6.6.

O *debugger* do MATRIXx® é de uso muito mais difícil. Ele abre sua janela com o código, e com um grifo no comando errado, conforme Figura 6.7.

Mas, como é o caso ilustrado, algumas vezes o *debugger* do MATRIXx® dá um alerta errado sobre a origem do problema. Neste caso, o MATRIXx® reporta uma incompatibilidade entre os tamanhos das variáveis a serem plotadas. Porém, copiando-se o trecho do código para a janela de comandos, os gráficos são plotados normalmente. Assim, percebe-se que o erro apontado não é a verdadeira causa do MATRIXx® parar de rodar o programa. Este erro deve-se ao uso da opção de plotagem “`keep`”(equivalente ao comando “`hold`”do MATLAB®), que serve para se plotar mais de uma curva no mesmo gráfico. Este comando, de alguma forma,

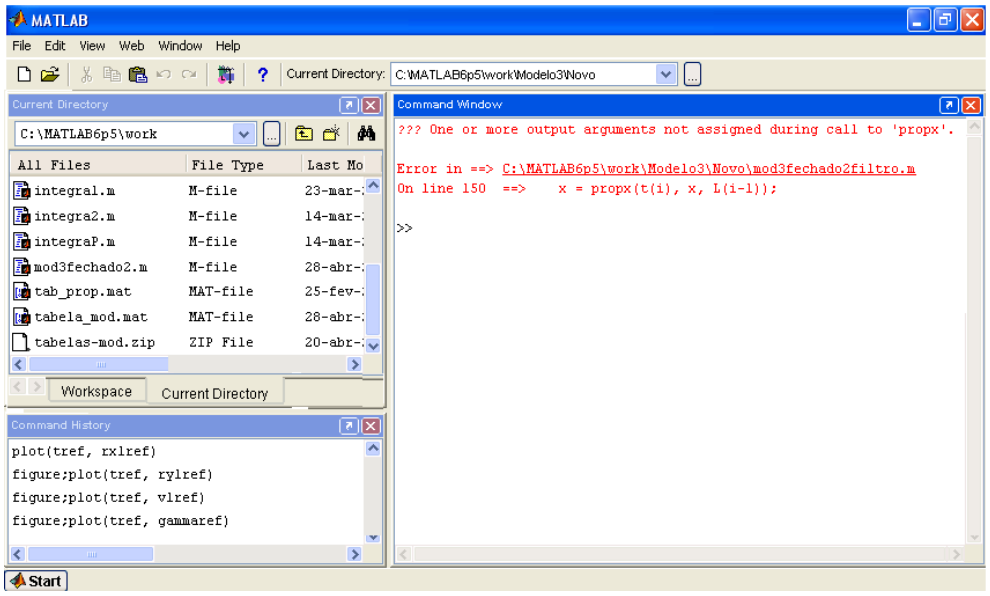


FIGURA 6.6 – Mensagem de Erro do MATLAB.

não se comporta bem quando associado a comandos para plotar vários gráficos na mesma janela ("`row`", "`column`") dentro de um arquivo .ms, .msf e outros, sendo que este *bug* não acontece quando os comandos são digitados interativamente na janela de comandos.

6.3 Modelo 3

No Modelo 3 foi o primeiro lugar onde a questão da partição se fez notar. Foi definida uma função $dx = f(x)$. Esta função depende de parâmetros a serem carregados. No MATLAB®, ao se carregar um arquivos de dados dentro de uma função, estas variáveis são carregadas em uma partição acessível apenas à função, ou seja, é uma variável local. No MATRIXx®, ao se carregar um arquivo de dados, ainda que dentro de uma função, colocam-se estes dados na partição principal, a menos que outra partição seja especificada. Ou seja, para acessar estes dados, faz-se necessário referenciá-los como "`main.dados`". Porém, esta característica apresenta a vantagem de permitir, em algumas situações, que um dado definido fora da função seja utilizado nos cálculos, ainda que não seja argumento da própria função.

Na definição de novas funções também surgem algumas diferenças entre os ambientes. O MATLAB® aceita, na definição da função, expressões como a atualização de

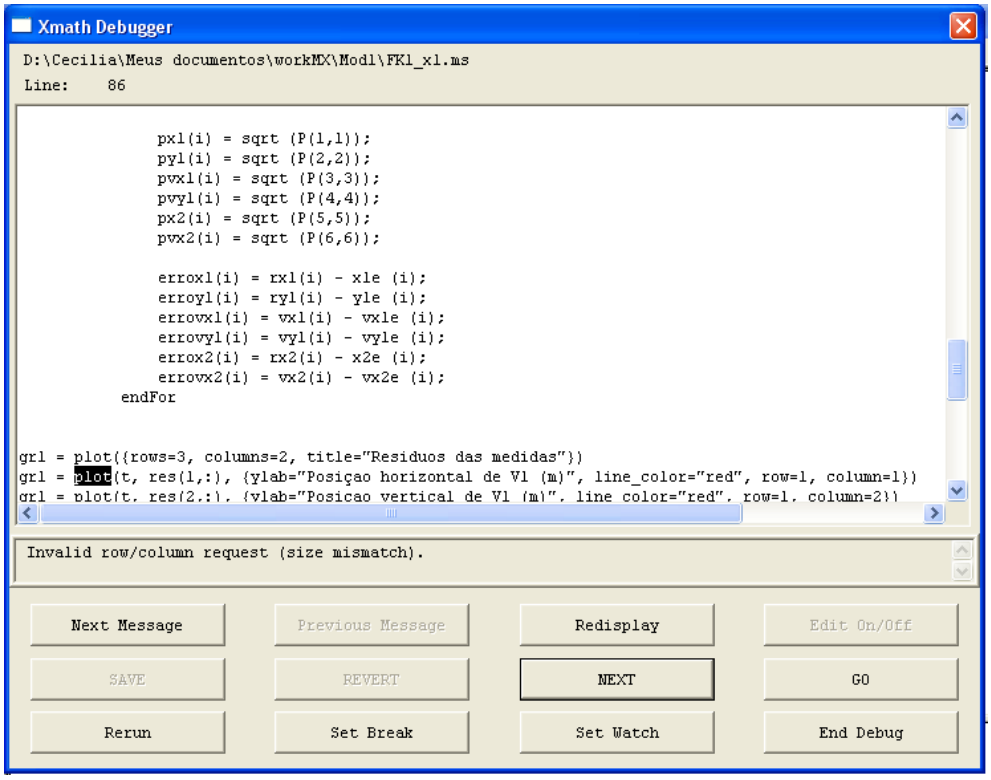


FIGURA 6.7 – Debugger do MATRIXx.

uma variável, como, por exemplo, $x = f(x)$. O MATRIXx® não aceita na definição de uma função que uma mesma variável seja a variável calculada e, ao mesmo tempo, argumento da função.

Outra questão diferente é a questão da impressão no ambiente gráfico. A impressão no MATLAB®/ *Simulink* é feita abrindo-se o bloco que se deseja imprimir. Assim, quando se abre a janela para impressão, o usuário pode escolher se quer imprimir somente o bloco aberto, o bloco aberto e superiores, o bloco aberto e inferiores ou todo o sistema. No MATRIXx® *SystemBuild* o bloco a ser impresso, e somente ele, é selecionado na janela de organização de blocos e se manda imprimir. Nenhum bloco interno será impresso. As janelas de impressão do MATLAB®/ *Simulink* e do MATRIXx® *SystemBuild* são mostradas, respectivamente, nas Figuras 6.8 e 6.9.

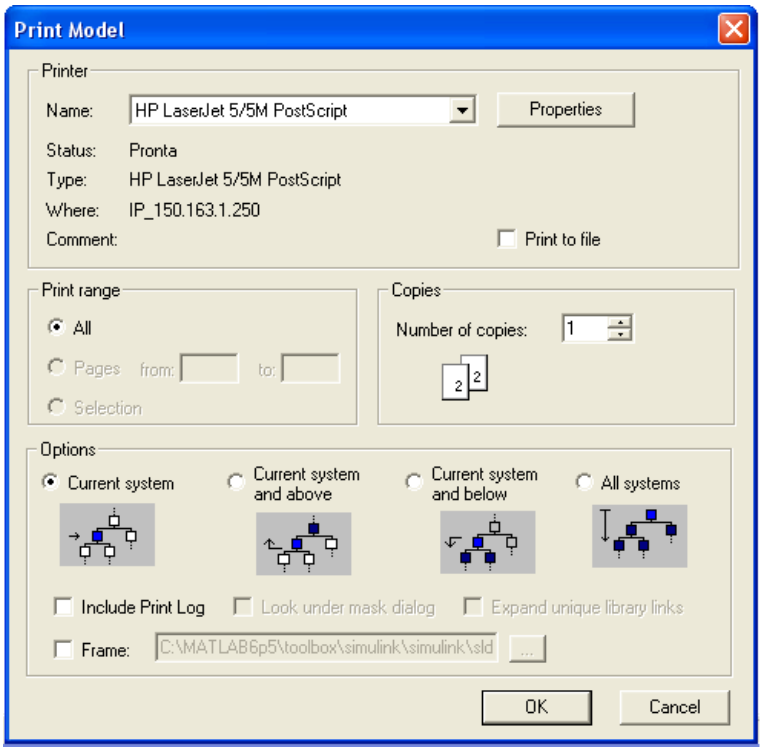


FIGURA 6.8 – Janela de impressão do MATLAB.

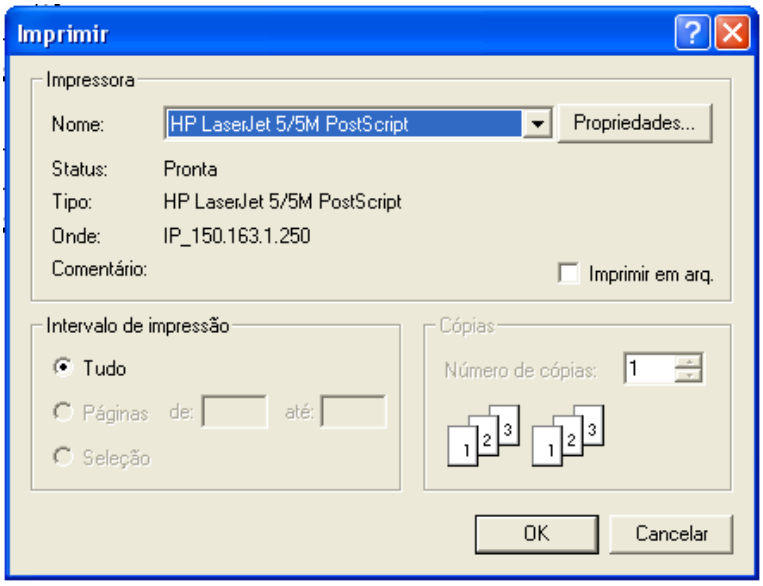


FIGURA 6.9 – Janela de impressão do MATRIXx.

CAPÍTULO 7

COMPARAÇÃO ESPECÍFICA

Neste Capítulo são apresentadas diferenças e semelhanças de ambos os ambientes, MATLAB® e MATRIXx®, constatadas na simulação de cada um dos três modelos apresentados.

7.1 Modelo 1

MATLAB® e MATRIXx® têm sua estrutura de matrizes e vetores muito parecidas. A única diferença é na separação de elementos da mesma linha, que no caso do MATLAB® é feito com espaços e no caso do MATRIXx® é feito com vírgulas.

O MATLAB® conta com 3 comandos que servem para "limpar"a área de trabalho: *"clear all"*, que apaga todas as variáveis, *"close all"*, que fecha todos os gráficos abertos, e *"clc"*, que limpa a janela de comandos. O MATRIXx® conta com as mesmas funcionalidades, utilizadas de forma diferente. O comando *"delete *.**" apaga todas as variáveis criadas. Porém, no caso de não haverem variáveis, ele indica erro. A limpeza das janelas de comando e histórico pode ser feita através de opções do menu, *"clear command"* e *"clear log"*, enquanto o gráfico deve ser fechado da forma convencional.

A chamada de outros arquivos pode ser feita de forma bastante diferente. Enquanto no MATLAB® basta digitar o nome do arquivo, desde que ele esteja no diretório corrente, no MATRIXx® é necessário um comando ("*execute file*"), acompanhado do endereço do arquivo e seu nome. Porém, ambos possuem o recurso no menu, permitindo que a chamada seja feita de forma gráfica.

As variáveis randômicas também dispõe de operadores diferentes nos dois ambientes, embora produzam o mesmo resultado. Enquanto o MATLAB® possui dois comandos para se gerar variáveis randômicas ("*rand*"e "*randn*"para distribuições uniforme e normal, respectivamente), o MATRIXx® possui apenas um, sendo o tipo de distribuição especificado pelo usuário, através do comando "*set distribution*". Os comandos do MATLAB® sem especificação de tamanho da variável são interpretados como escalares, enquanto o comando do MATRIXx® exige uma especificação

da dimensão da variável.

A forma de se plotar gráficos também é bastante diferente. O MATRIXx® trata seus gráficos como objetos, enquanto o MATLAB® dá um tratamento diferenciado. Assim, quando se salvam os dados da área de trabalho do MATRIXx®, os gráficos são salvos junto. No MATLAB®, apenas as variáveis são salvas, podendo os gráficos serem salvos no formato *.fig* ou exportados como figuras.

Ao se plotar mais de um gráfico no MATLAB®, ele substitui a última janela de gráfico pelo atual, ou, no caso de um novo gráfico (*"figure"*), abre mais uma janela. O MATRIXx® atribui um nome de variável àquele gráfico, exibindo-o quando a variável é chamada. Enquanto no MATLAB® pode-se abrir mais de uma janela de gráficos ao mesmo tempo, posicionando-as no monitor de forma a comparar os gráficos (mantendo, no entanto, apenas uma janela ativa), o MATRIXx® permite que apenas um gráfico seja aberto de cada vez.

Desejando-se plotar gráficos lado a lado, ambos os ambientes têm sua funcionalidade específica. O MATLAB® o faz através do comando *"subplot"*, enquanto o MATRIXx® posiciona o gráfico com os comandos *"row"* e *"column"*, colocados nas especificações da plotagem.

Quanto aos ambientes gráficos, foram encontradas as mesmas funções, apenas em formatos diferentes.

Os blocos do MATRIXx®/*SystemBuild* são muito mais expressivos do que os blocos do MATLAB®/*Simulink*. A visualização dos bloco pode ser de cinco tipos (simples, especial, do usuário, alternativo e padrão), fornecendo a possibilidade de se enxergar através dos Superblocos (blocos que contém outros blocos), fazendo com que o usuário já saiba quais os blocos e superblocos que estão dentro. Além disso, informa no próprio bloco se ele é contínuo ou discreto. Os blocos do MATLAB® são apenas retângulos simples, com nomes de saídas e entradas. Exemplos de ambos os tipos de blocos podem ser verificados nas Figuras 7.1 e 7.2.

Além disso, o MATRIXx®/*SystemBuild* conecta linhas que saem de um mesmo bloco para outro bloco em comum em uma única linha, deixando o diagrama de blocos mais limpo.

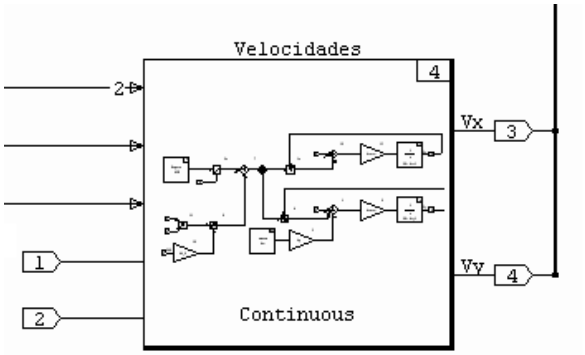


FIGURA 7.1 – Exemplo de bloco do MATRIXx.

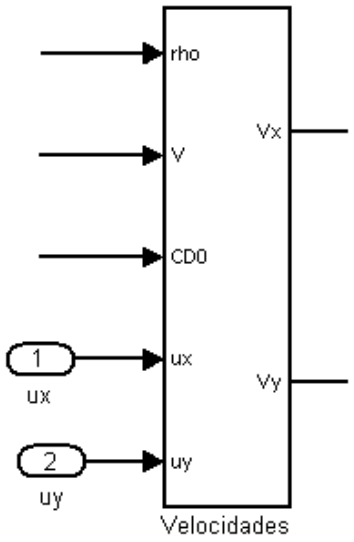


FIGURA 7.2 – Exemplo de bloco do MATLAB.

O simulador de tempo do MATLAB® é um bloco pronto, que pode apenas ser inserido no diagrama de blocos. O MATRIXx® utiliza um bloco para comandos, o *"BlockScript"* onde o algoritmo de simulação de tempo (muito simples) é feito. Estes blocos podem ser vistos nas Figuras 7.3 e 7.4, respectivamente.

O vetor de tempo de simulação tem seu princípio e seu final determinados no MATLAB®, sendo os valores intermediários dependentes do tipo de integrador que se utiliza. No MATRIXx®, o vetor de tempo é determinado pelo usuário, e inserido como um dado da simulação. Os resultados no MATLAB® podem ser enviados para a área de trabalho através de bloco próprio, e um osciloscópio utilizado para se verificar o andamento da simulação.

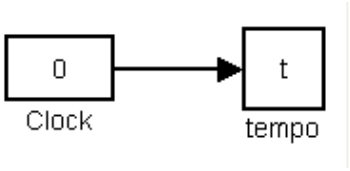


FIGURA 7.3 – Simulador de tempo do MATLAB.

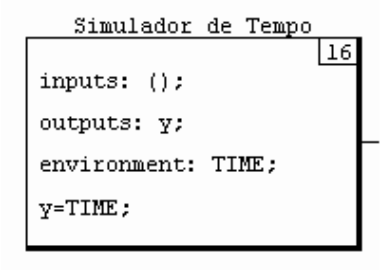


FIGURA 7.4 – Simulador de tempo do MATRIXx.

No MATRIXx® existe uma funcionalidade na janela de simulação que permite que os dados da simulação sejam exportados para a área de trabalho, na forma de PDM´s, e há uma opção para se plotar os gráficos de resultados de todas as saídas assim que a simulação termina.

7.2 Modelo 2

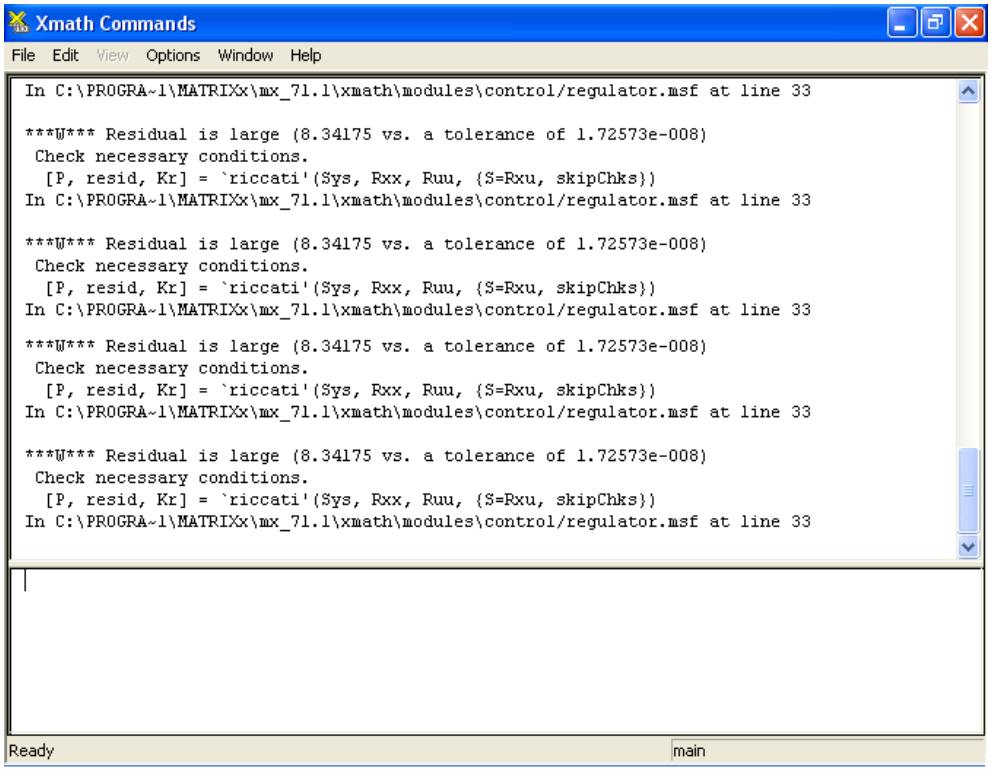
No Modelo 2 foi utilizada pela primeira vez a funcionalidade das PDM´s, as matrizes dependentes de parâmetros do MATRIXx®. Embora o MATLAB® não conte com a ferramenta das PDM´s, em todas as situações testadas durante este trabalho, tais matrizes foram substituídas com eficiência pelos tensores, disponíveis no MATLAB® e não disponíveis no MATRIXx®. No caso da necessidade de se utilizar algum dado inserido em uma PDM, há um comando próprio("*makematrix*") que transforma o dado em um objeto numérico como os outros.

Outro comando explorado neste modelo foi o comando utilizado para arredondamentos. Ambos os ambientes possuem exatamente as mesmas opções de arredondamento. O MATLAB® possui um comando para arredondar para o número mais próximo ("*round*"), arredondar para cima ("*ceil*") ou para baixo ("*floor*"). O MATRIXx® possui apenas um comando para todas as opções ("*round*"), cujo parâmetro padrão é arredondar para o mais próximo ("*nearest*"), podendo também realizar o arredon-

damento para cima ("*up*") ou para baixo ("*down*"), parâmetros estes especificados nas opções do comando.

Nos ambientes gráficos as implementações foram feitas de forma diferente. No MATLAB® */Simulink* foi utilizado o bloco chamado "MATLAB Function", que permite que uma função do MATLAB® seja utilizada dentro de um bloco. Há um bloco equivalente no MATRIXx® *SystemBuild*, chamado "MathScript Block", que permite que as funções do MATRIXx® sejam utilizadas em blocos. Porém, nenhum destes dois blocos é codificável pelo gerador automático de código do respectivo ambiente, embora isto não fique muito claro no MATLAB®. O MATRIXx® deixa bem claro que tal bloco não é codificável. Assim, na hora de montar o diagrama de blocos no MATLAB®, foi utilizado um bloco "MATLAB Function"contendo a função *"lqr"*, enquanto que no MATRIXx® os valores de ganhos do regulador foram calculados no *Xmath* e armazenados em um bloco chamado "*Gain Scheduler*", que, como o próprio nome diz, escolhe a tabela de ganhos a ser utilizada para cada instante de tempo.

Os próprios comandos para se calcular os ganhos do Regulador Linear Quadrático são, em sua essência, diferentes. O comando do MATLAB® para se calcularem os ganhos do controlador regulador linear quadrático é o *"lqr"*, que é calculado a partir das matrizes *A, B, Q, R* fornecidas, onde A e B são as matrizes do sistema linear e Q e R são as matrizes de peso, respectivamente, do estado e do controle. O comando equivamente do MATRIXx® é o *"regulator"*, que trabalha com dados mais completos. A entrada para esta função é um sistema dinâmico, além das matrizes Q e R. Este sistema linear pode ser definido de várias formas, entre elas, matrizes *A, B, C, D* de um sistema linear, função de transferência ou um objeto que caracterize um sistema linear. Há possibilidade, ainda, de, no sistema dinâmico, se especificar as condições iniciais de tal sistema. Assim, o MATRIXx® considera todo o sistema a ser controlado, e não somente as matrizes da dinâmica. Isto se deve à natureza do MATRIXx®, que é intrinsecamente voltada para controle. Daí as diferenças de ganhos de alguns componentes da matriz de ganhos dos controladores. Esta diferença faz com que os algoritmos de cálculo dos ganhos seja diferente, de forma que o cálculo que é feito no MATLAB® normalmente, no MATRIXx® aparecem alguns *warnings*, alertando quanto a possíveis erros numéricos devidos ao sistema a ser controlados, conforme Figura 7.5.



```
In C:\PROGRA~1\MATRIXx\mx_71.1\xmath\modules\control/regulator.msf at line 33

***W*** Residual is large (8.34175 vs. a tolerance of 1.72573e-008)
 Check necessary conditions.
  [P, resid, Kr] = `riccati'(Sys, Rxx, Ruu, {S=Rxu, skipChks})
In C:\PROGRA~1\MATRIXx\mx_71.1\xmath\modules\control/regulator.msf at line 33

***W*** Residual is large (8.34175 vs. a tolerance of 1.72573e-008)
 Check necessary conditions.
  [P, resid, Kr] = `riccati'(Sys, Rxx, Ruu, {S=Rxu, skipChks})
In C:\PROGRA~1\MATRIXx\mx_71.1\xmath\modules\control/regulator.msf at line 33

***W*** Residual is large (8.34175 vs. a tolerance of 1.72573e-008)
 Check necessary conditions.
  [P, resid, Kr] = `riccati'(Sys, Rxx, Ruu, {S=Rxu, skipChks})
In C:\PROGRA~1\MATRIXx\mx_71.1\xmath\modules\control/regulator.msf at line 33

***W*** Residual is large (8.34175 vs. a tolerance of 1.72573e-008)
 Check necessary conditions.
  [P, resid, Kr] = `riccati'(Sys, Rxx, Ruu, {S=Rxu, skipChks})
In C:\PROGRA~1\MATRIXx\mx_71.1\xmath\modules\control/regulator.msf at line 33
```

FIGURA 7.5 – Mensagem de aviso do MATRIXx.

O *debugger* do MATLAB® apresentou-se bastante eficaz. No caso de erro, o MATLAB® apresenta em sua tela principal, na janela de comandos, o arquivo que contém o erro, assim como a linha onde tal erro se encontra. Clicando-se neste aviso, o usuário já é remetido diretamente ao ponto de erro, já de posse da origem do erro, para consertá-lo, conforme mostrado na Figura 7.6.

O *debugger* do MATRIXx® é de uso muito mais difícil. Ele abre sua janela com o código, e com um grifo no comando errado, conforme Figura 7.7.

Mas, como é caso ilustrado, algumas vezes o *debugger* do MATRIXx® dá um alerta errado sobre a origem do problema. Neste caso, o MATRIXx® reporta uma incompatibilidade entre os tamanhos das variáveis a serem plotadas. Porém, copiando-se o trecho do código para a janela de comandos, os gráficos são plotados normalmente. Assim, percebe-se que o erro apontado não é a verdadeira causa do MATRIXx® parar de rodar o programa. Este erro deve-se ao uso da opção de plotagem *"keep"* (equivalente ao comando *"hold"* do MATLAB®), que serve para se plotar mais de uma curva no mesmo gráfico. Este comando, de alguma forma, não se comporta bem

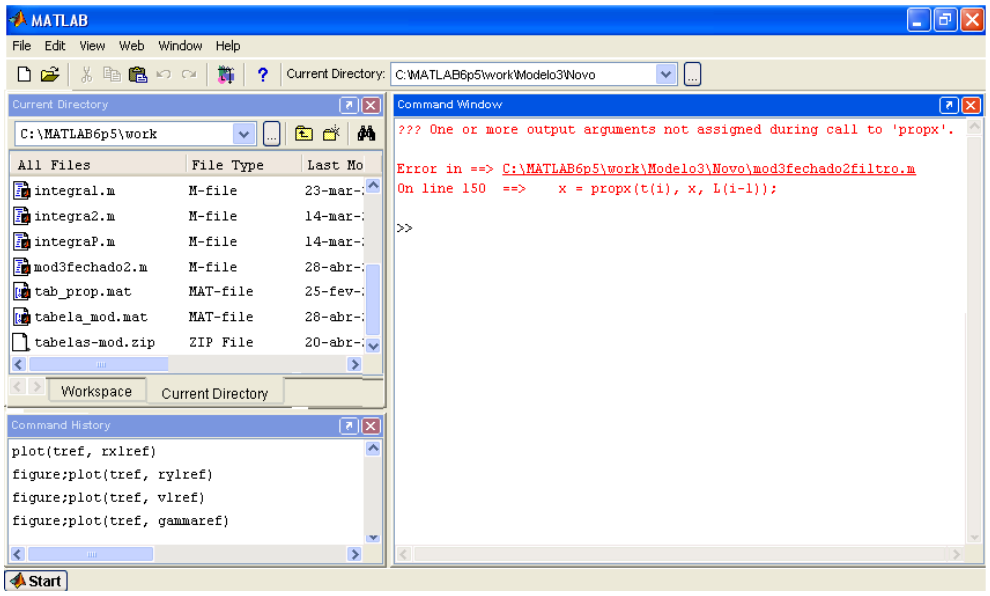


FIGURA 7.6 – Mensagem de Erro do MATLAB.

quando associado a comandos para plotar vários gráficos na mesma janela (*"row"*, *"column"*) dentro de um arquivo .ms, .msf e outros, sendo que este *bug* não acontece quando os comandos são digitados interativamente na janela de comandos.

7.3 Modelo 3

No Modelo 3 foi o primeiro lugar onde a questão da partição se fez notar. Foi definida uma função $dx = f(x)$. Esta função depende de parâmetros a serem carregados. No MATLAB®, ao se carregar um arquivos de dados dentro de uma função, estas variáveis são carregadas em uma partição acessível apenas à função, ou seja, é uma variável local. No MATRIXx®, ao se carregar um arquivo de dados, ainda que dentro de uma função, colocam-se estes dados na partição principal, a menos que outra partição seja especificada. Ou seja, para acessar estes dados, faz-se necessário referenciá-los como "*main.dados*". Porém, esta característica apresenta a vantagem de permitir, em algumas situações, que um dado definido fora da função seja utilizado nos cálculos, ainda que não seja argumento da própria função.

Na definição de novas funções também surgem algumas diferenças entre os ambientes. O MATLAB® aceita, na definição da função, expressões como a atualização de uma variável, como, por exemplo, $x = f(x)$. O MATRIXx® não aceita na definição

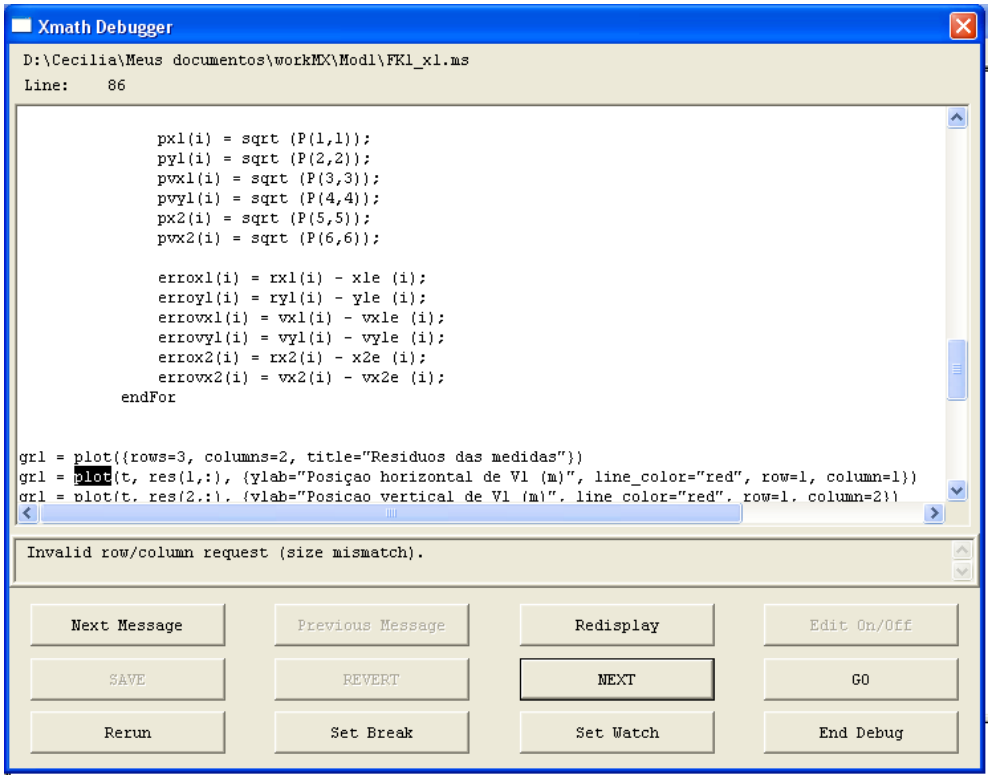


FIGURA 7.7 – Debugger do MATRIXx.

de uma função que uma mesma variável seja a variável calculada e, ao mesmo tempo, argumento da função.

Outra questão diferente é a questão da impressão no ambiente gráfico. A impressão no MATLAB®/ *Simulink* é feita abrindo-se o bloco que se deseja imprimir. Assim, quando abre-se a janela para impressão, o usuário pode escolher se quer imprimir somente o bloco aberto, o bloco aberto e superiores, o bloco aberto e inferiores ou todo o sistema. No MATRIXx® *SystemBuild* o bloco a ser impresso, e somente ele, é selecionado na janela de organização de blocos e se manda imprimir. Nenhum bloco interno será impresso. As janelas de impressão do MATLAB®/ *Simulink* e do MATRIXx® *SystemBuild* são mostradas, respectivamente, mas Figuras 7.8 e 7.9.

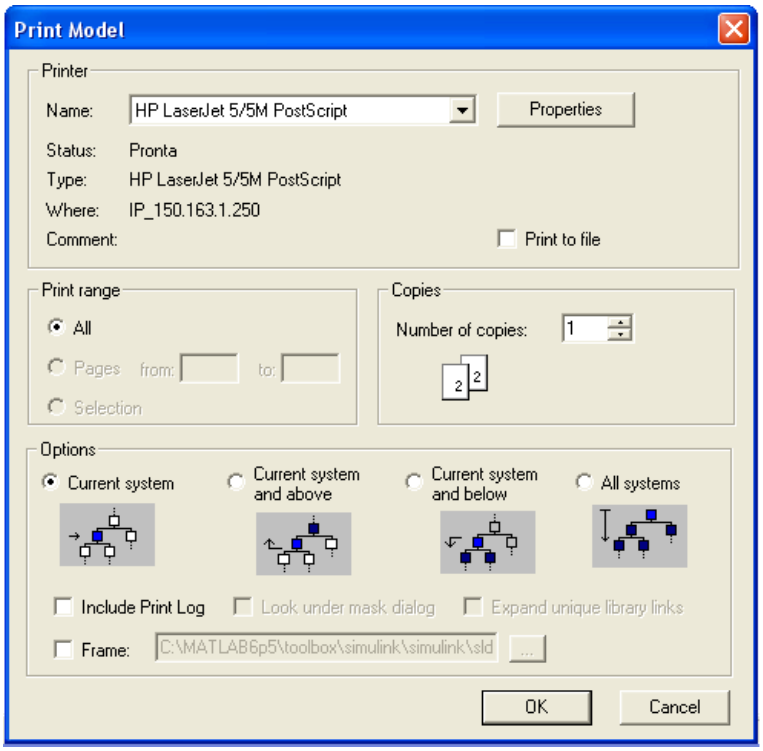


FIGURA 7.8 – Janela de impressão do MATLAB.

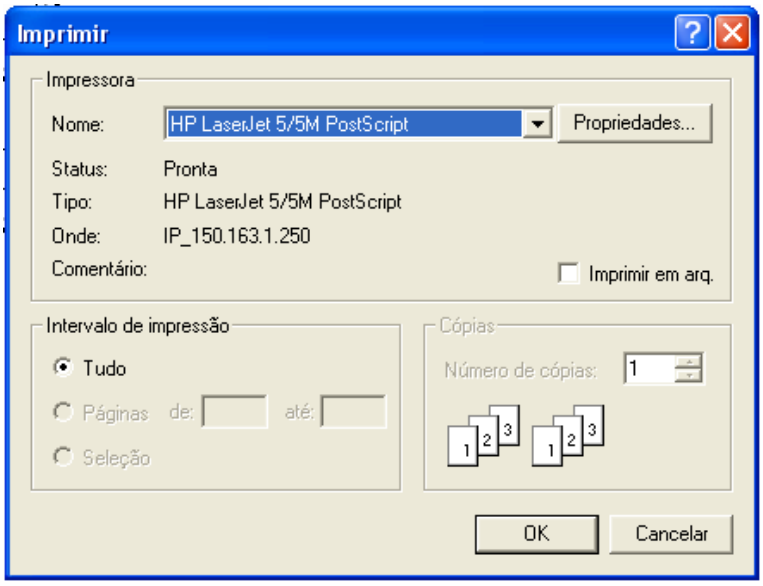


FIGURA 7.9 – Janela de impressão do MATRIXx.

CAPÍTULO 8

CONCLUSÕES, RECOMENDAÇÕES E SUGESTÕES

Conclui-se com este trabalho que a evolução de modelos pode revelar limitações de técnicas que, para um modelo mais simples, mostram-se eficientes. Daí a necessidade de trabalhar com modelos progressivamente mais realistas. O modelo 1 não considerava nenhum efeito além da gravidade. A atmosfera era desconsiderada, assim como o empuxo do motor e a variação de massa. No modelo 2 consideramos um modelo mais realista. Como todos os modelos apresentados neste trabalho, as equações de estado podem ser deduzidas a partir da Segunda Lei de *Newton*. Considerou-se o efeito do arrasto atmosférico, inclusive assumindo um modelo para a densidade atmosférica. Para esta consideração, foi necessário definir a área da seção reta do veículo, embora seu movimento rotacional fosse ignorado nas equações. Também há nas equações um termo relativo ao empuxo do motor, assumido constante durante todo o tempo de vôo, assim como a massa. Observa-se que é considerado no modelo que a força de empuxo oferecida pelo motor está sempre na direção do vetor velocidade. Porém, o controle aplicado é dado no referencial inercial. Tal referencial não é adequado para o cálculo do controle, uma vez que o atuador responsável pela manobra está fixo ao referencial do corpo. Faz-se então necessário um modelo que considere um controle aplicado no referencial do corpo, para que seja factível. Assim é o modelo 3 que conta com um controle atuando no ângulo que o vetor velocidade faz com a horizontal. Consideramos aí, além da força de empuxo do motor (desta vez considerada variável com o tempo), o consumo de massa à medida que passa o tempo. Porém, ainda há a simplificação do empuxo do motor estar na direção do vetor velocidade. No entanto, é um modelo realista o suficiente para representar uma situação de vôo na fase de cruzeiro.

Como dito anteriormente, o modelo 1 não conta com um sistema de controle. O modelo 2 conta com um controlador que gera dois sinais de controle, que deveriam ser aplicados nas direções x e z inerciais. Embora tal controle não seja implementável na prática, os sinais gerados pelo controlador projetado com o auxílio da Teoria de Controle Ótimo fizeram com que o veículo seguisse a trajetória de referência dentro das especificações determinadas, ainda que lançado em uma posição diferente da posição de referência. Priorizamos neste trabalho o rastreamento de posição, embora considerássemos no controle também um rastreamento de velocidade. Observamos

para o modelo 2 que os ganhos do controlador calculados com o MATLAB® ou MATRIXx® foram diferentes. Tais ganhos, no entanto, eram da mesma ordem de grandeza, fazendo, assim, que o controle fosse eficiente com ambos os valores de matrizes de ganho. Tal diferença se dá por implementações próprias de cada ambiente para a solução do problema. Entradas de matrizes de mesmo valor na função de cada ambiente geram diferentes valores de matrizes de ganho. As diferenças de implementação não puderam ser observadas, uma vez que embora ambos os ambientes se definam como ambientes de "código aberto", o acesso ao código não é permitido para estas funções. Observa-se também neste modelo 2 que quando a trajetória de referência vertical permanece constante no tempo, o valor do sinal do controle tende a ser aquele que balanceia a força peso. O controlador projetado para o modelo 3 atuava somente em uma variável de estado, γ, o ângulo que o vetor velocidade do veículo faz com a horizontal. Este controle é implementável por ser calculado em um referencial preso ao veículo. Novamente foi priorizado o rastreamento da posição. Porém, como o controle é mais uma das formas de acoplar os movimentos nas direções x e z, o rastreio na direção x acabou divergindo da trajetória nominal, nos últimos $30s$. Neste caso a matriz de ganhos do controlador foi igual para ambos os ambientes de MIS. Como para um modelo mais elaborado pode-se considerar uma saturação do controle, determinado pelo fator de carga do veículo, o rastreamento pôde se tornar mais eficiente, a partir de escolhas das matrizes para cálculo do controlador que fizessem uma resposta mais rápida no controle.

O estimador feito para o modelo 1 produziu boas estimativas tanto para o veículo rastreador como para o veículo rastreado. Os resíduos das medidas disponíveis tiveram média nula, se mantendo em sua maioria em valores na faixa definida por 3σ, onde σ é a margem de erro dos sensores. Para o veículo rastreador a convergência da covariância se estabilizou logo nos primeiros passos do filtro, se mantendo praticamente constante até o final do tempo de simulação. Para o veículo rastreado a covariância tendeu a diminuir. Se o rastreamento fosse simulado por mais tempo, provavelmente a covariância diminuiria até destruir a estabilidade do filtro, dada a ausência de controlabilidade estocástica do sistema. No entanto, para o período simulado, os erros das estimativas se mantiveram dentro do limite definido pela covariância. Para o modelo 2 as estimativas ainda foram eficientes para ambos os veículos. Assim como no modelo 1, os resíduos das medidas se comportaram conforme previsão teórica. A propagação da covariância foi feita através de uma matriz de transição Φ constante por trechos. Para as estimativas do veículo rastreador, no-

vamente o valor da covariância convergiu nos primeiros passos, ficando o valor do erro dentro da faixa definida por ela. A covariância para as estimativas do veículo rastreado também convergiu, embora mais lentamente, porém de modo satisfatório. O erro das estimativas mais uma vez ficou dentro da faixa definida pela covariância. Para o modelo 3 mais uma vez os resíduos das medidas tiveram o comportamento previsto pela teoria, de média nula e ficando dentro da faixa definida por 3σ. A propagação do filtro foi feita através da integração das equações de estado. As estimativas de posição do veículo rastreador tiveram suas covariâncias descrescentes até se estabilizar, com os erros das estimativas permanecendo dentro da faixa definida por elas. Para a velocidade e o ângulo que esta faz com a horizontal, a covariância descresceu, embora oscilasse dentro de um valor tolerável. Tal variação se dá em razão dos estados totalmente acoplados considerados no modelo. A maior diferença, no entanto, se mostrou nas estimativas do estado do veículo rastreado. Não somente a covariância aumentou e permaneceu em um valor alto, como o erro da estimativa foi também aumentando gradativamente. No entanto concluímos que para este modelo o filtro de *Kalman* ainda é eficiente para a estimativa de estado do próprio veículo rastreador.

Ambos os ambientes se mostraram eficientes como ferramentas para a simulação do problema, apresentando recursos parecidos disponíveis ao usuário. Embora ambos os ambientes sejam definidos pelos seus fabricantes como ambientes de "código aberto", no caso de funções mais elaboradas, como aquelas que calculam o ganho para um controlador ótimo, tal código não está disponível. É possível, sim, abrir os arquivos onde estas funções estão descritas. Porém estas funções utilizam comandos cujos arquivos não são encontrados ou recorrem a outros arquivos que mais uma vez não se pode localizar. Apesar da diferença de implementação para a referida função e dos diferentes resultados, ambos as matrizes de ganho calculadas por eles produzem um rastreamento eficaz. Os ambientes, embora possuam apresentações diferentes, apresentam basicamente os mesmos recursos distribuídos de forma diferente na sua estrutura. Porém, como as ferramentas são utilizadas de forma diferente e oferecem algumas diferenças menores na sua estrutura, o usuário pode se identificar melhor com um ou outro, podendo recair também no objetivo do seu trabalho a escolha de um ou outro ambiente de MIS.

Como trabalhos futuros recomenda-se aumentar os graus de realismo dos modelos a fim de se verificar até onde a técnica de congelamento de pólos é adequada, e até

onde os controladores e estimadores são robustos. Também pode-se aplicar a teoria do Regulador Quadrático não-linear, pouco explorada em nosso meio, tanto para se verificar a melhora nos resultados é significativa para os modelos linearizados, quanto para ser utilizado em casos onde a linearização não se aplica.

Sugere-se, também, em trabalhos futuros, que a avaliação do código automaticamente gerado pelos ambientes seja feita mais cuidadosamente, assim como a geração automática de documentação, uma vez que estas duas etapas de um trabalho de engenharia consomem muito tempo.

REFERÊNCIAS BIBLIOGRÁFICAS

Abdallah, Y. M. **Curso básico de MATLAB® R13**. São José dos Campos: INPE, 2002. Em publicação.

Astrom, K. J.; Wittenmark, B. **Adaptive control**. 2. ed. Massachusetts: Addison-Wesley, 1989.

Blakelock, J. H. **Automatic control of aircraft and missiles**. 2. ed. New York: John Wiley & Sons, 1991.

Brown, R. G.; Hwang, P. Y. C. **Introduction to random signals and applied Kalman filtering**. 3. ed. New York: John Wiley & Sons, 1997.

Bryson Jr, A. E.; Ho, Y.-C. **Applied optimal control**. Washington, D.C., USA: John Wiley & Sons, 1975.

Carvalho, C. A. de P. **Estudo do algoritmo de múltiplos tiros para controle ótimo em tempo real de veículos aeroespaciais**. 1989. 64 p. Dissertação (Mestrado em Engenharia Eletrônica na Área de Sistemas de Controle) — Instituto de Tecnologia e Aeronáutica, São José dos Campos, 1989.

Cheng, V. H. L.; Gupta, N. K. Advanced midcourse guidance air-to-air missiles. **Journal of Guidance**, v. 9, p. 135 – 142, 1986.

Cornelisse, J. W.; Schöyer, H. F. R.; Wakker, K. F. **Rocket propulsion and spaceflight dynamics**. London, England: Pitman, 1979.

Davis, C. New developments in modelling and simulation for modelling submarine operations. In: Submarine Science & Technology Conference. **Proceedings...** Stirling, Australia: DSTO, 1998.

Durand, W. F. Mathematical aids. In: Durand, W. F. (ed.). **Aerodynamic theory**. Pasadena, CA, USA: Durand Reprinting Comittee, 1943.

Garnell, P. **Guided weapon control systems**. Great Britain: Pergamon Press, 1980.

Greensite, J. P. **Analysis and design of space vehicle flight control systems-Control Theory**. 2. ed. New York, USA: Spartan Books, 1970.

Johansson, R. **System modeling & identification**. New Jersey, USA: Prentice Hall, 1993.

Kirk, D. E. **Optimal control theory:** an introduction. New Jersey, USA: Prentice- Hall, 1970.

Kuethe, A. M.; Chow, C.-Y. **Foundations of aerodynamics:** bases of aerodynamic design. New York, USA: John Wiley & Sons, 1950.

Kuga, H. K. **Sobre a utilização prática de técnicas de estimação**. 2003. Notas de aula.

Leigh, J. R. Modelling principles and simulation. In: Nicholson, H. (ed.). **Modelling of Dynamical Systems**. London, England: The Institution of Electrical Engineers, 1980. v. 1, p. 1 – 24.

Leite Filho, W. de C. **Pilotagem de mísseis táticos na fase de cruzeiro**. 1982. 94 p. Dissertação (Mestrado em Ciências) — Instituto Militar de Engenharia, Rio de Janeiro, 1982.

Lin, C.-L.; Su, H.-W. Intelligent control theory in guidance and control system design: an overview. **Proceedings of National Sciences**, China, v. 24, p. 15 – 30, 2000.

Linkens, D. **CAD for control systems**. Sheffield, England: Marcel Dekker, 1993.

Lipscombe, J. Aerospace systems. In: Nicholson, H. (ed.). **Modelling of dynamical systems**. London, England: The Institution of Electrical Engineers, 1980. v. 1, p. 117 – 166.

Lopes, R. V. da F. **Otimização de sistemas dinâmicos II**. 2003. Notas de aula.

Maybeck, P. S. **Stochastic models, estimation and control**. New York: Academic Press, 1979.

National Aeronautics and Space Administration. **US standard atmosphere**. Washington, USA: NASA, 1966.

National Instruments. **MATRIXx product family user´s manual**. Austin, TX, USA: National Instruments, 2003.

Ogata, K. **Modern control engineering**. 3. ed. New York, USA: Prentice-Hall, 1997.

Ören, T. I. Future of modelling and simulation: some development areas. In: Proceedings of the 2002 Summer Computer Simulation Conference. **Proceedings...** Ottawa, Canada, 2002.

Shamma, J. S.; Athans, M. Gain scheduling: potential hazards and possible remedies. **IEEE Control Systems**, v. 12, p. 101 – 107, 1992.

Sobey, A. J.; Suggs, A. M. **Control of aircraft and missile powerplants**. New York, USA: John Wiley & Sons, 1963.

Steinhaus, S. Comparison of mathematical programs for data analysis. München, Germany, 2002. Disponível em: <http://www.scientificweb.de/ncrunch/>.

Stovall, S. H. **Basic inertial navigation**. Naval Air Warfare Center Weapons Division, Sept. 1997.

Takahashi, Y.; Rabins, M. J.; Auslander, D. M. **Control and dynamic systems**. 2. ed. Massachusetts, USA: Addison-Wesley, 1972.

The MathWorks, I. **MATLAB product family manuals**. Natick, MA, USA: The MathWorks, 2005.

Trivelato, G. C.; Souza, M. L. O. **Comparando MATRIXx e MATLAB para modelagem e simulação de veículos aeroespaciais**. São José dos Campos: INPE, 2001. (INPE-9658-PRE/5266).

Welch, G.; Bishop, G. An introduction to the Kalman filter. In: International Conference on Computer Graphics and Interative Techniques, 25., 12-17 August 2001. **Proceedings...** Los Angeles: SIGGRAPH, 2001.

White, F. M. **Fluid mechanics**. New York, USA: McGraw Hill, 1981.

APÊNDICE A

SÉRIE DE TAYLOR

Para funções de apenas uma variável, seja $x = f(u)$. Expandindo esta função em torno de um ponto de interesse u_0 em série de *Taylor*, tem-se:

$$x = f(u_0) + \frac{df}{du}\Big|_{u_0} (u - u_0) + \mathscr{O}^2 \tag{A.1}$$

Se $x_0 = f(u_0)$:

$$\begin{aligned} x_0 + x - x_0 &= f(u_0) + \frac{df}{du}\Big|_{u_0} (u - u_0) + \mathscr{O}^2 && \text{(A.2)} \\ x - x_0 &= \frac{df}{du}\Big|_{u_0} (u - u_0) + \mathscr{O}^2 && \text{(A.3)} \end{aligned}$$

Ou seja:

$$\begin{aligned} \Delta x &= \frac{df}{du}\Big|_{u_0} \Delta u + \mathscr{O}^2 && \text{(A.4)} \\ \delta x &= \frac{df}{dx}\Big|_{u_0} \Delta u && \text{(A.5)} \\ \Delta u &= \delta u && \text{(A.6)} \\ \Delta(x) &= \delta x + \mathscr{O}^2 && \text{(A.7)} \end{aligned}$$

Onde $(\Delta u, \Delta x)$ é a perturbação em torno do ponto de linearização (u_0, x_0).

Caso f seja uma função de múltiplas variáveis, ainda assim usa-se a expansão em série de *Taylor*. Para isto, seja:

$$x = f(u_1, u_2, \ldots, u_n) \tag{A.8}$$

A expansão em série de *Taylor* fornece:

$$x = f\left(u_1^o, \ldots, u_n^o\right) + \frac{\partial f}{\partial u_1}\Big|_{u_0}\left(u_1 - u_1^o\right) + \cdots + \frac{\partial f}{\partial u_n}\Big|_{u_0}\left(u_n - u_n^o\right) + \mathscr{O}^2 \tag{A.9}$$

Assim, a perturbação da função em torno do ponto de referência $(u_1^o, \ldots, u_1^n)$é dada por:

$$\Delta x \approx \delta x = \frac{\partial f}{\partial u_1}\Big|_{u_0}\delta u_1 + \ldots + \frac{\partial f}{\partial u_n}\Big|_{u_0}\delta u_n \tag{A.10}$$

Onde o subscrito u nas equações acima se refere ao vetor das variáveis independentes: $u_0 = (u_1^0, \ldots, u_n^0)$

Observe-se que são considerados apenas termos de primeira ordem. Assim, para se determinar o grau de aproximação que está sendo feito, deve-se analisar a ordem de grandeza dos termos de ordem superior que estão sendo desprezados (Leigh, 1980).

APÊNDICE B

EQUAÇÕES DINÂMICAS

Para deduzir estas equações, algumas hipóteses serão assumidas, a saber:

- Os eixos OX e OZ estão sobre o plano de simetria do veículo, o que resulta em produtos de inércia I_{xy} e I_{yz} nulos. Entretanto, o eixo OX não deve ser um dos eixos principais de inércia, para que I_{zx} seja não nulo, a fim de se abordar um caso geral nas deduções;
- Considera-se que o veículo está em equilíbrio até que um controle ou uma perturbação sejam adicionados;
- Forças de equilíbrio são a força da gravidade, o empuxo, a força de arrasto e a sustentação;
- Torques de equilíbrio são os resultantes da sustentação e do arrasto gerados pelas várias partes do veículo e do empuxo;
- O veículo está, a princípio, em vôo não acelerado e os distúrbios em geral são provocados pela deflexão das superfícies de controle ou alterações nas condições atmosféricas;
- A massa do veículo permanece aproximadamente constante durante qualquer análise dinâmica feita na fase de cruzeiro tratada neste trabalho;
- O veículo é considerado um corpo rígido (Blakelock, 1991);
- O referencial inercial pode ser fixado na Terra e, a menos que previamente especificado, sua atmosfera permanece em repouso em relação a ela (esta simplificação pode ser feita devido a limitações dos sensores, que não medem a velocidade angular da Terra e acelerações provocadas por ela);
- O empuxo é alinhado com OZ;
- O campo gravitacional é constante.

Assim, têm-se a Segunda Lei de *Newton* da translação e rotação, respectivamente:

$$\sum \vec{F} = \frac{d}{dt}\left(m\vec{V}_T\right) \quad \text{(B.1)}$$

$$\sum \vec{\tau} = \frac{d}{dt}\vec{L} \quad \text{(B.2)}$$

Onde:

- $\vec{F}$ é a forças atuando no veículo
- m é a massa do veículo
- $\vec{V}_T$ é a velocidade do centro de massa do veículo
- $\vec{\tau}$ é o torque atuando no veículo
- $\vec{L}$ é o momento angular do veículo

Conforme ressaltado anteriormente, as grandezas e suas derivadas são definidas em relação ao referencial inercial. Os somatórios de força e torque são dados por:

$$\sum \vec{F} = \sum \vec{F}_0 + \sum \Delta\vec{F} \quad \text{(B.3)}$$

$$\sum \vec{\tau} = \sum \vec{\tau}_0 + \sum \Delta\vec{\tau} \quad \text{(B.4)}$$

Onde $\sum \vec{F}_0$ e $\sum \vec{\tau}_0$ são a soma das forças e torques de equilíbrio, nulos por hipótese. Assim, têm- se:

$$\sum \Delta\vec{F} = \frac{d}{dt}\left(m\vec{V}_T\right) \quad \text{(B.5)}$$

$$\sum \Delta\vec{\tau} = \frac{d}{dt}\vec{L} \quad \text{(B.6)}$$

Considerando o movimento do centro de massa do veículo em relação à Terra, a Equação (B.5) pode ser escrita como:

$$\sum \Delta \vec{F} = \frac{dm}{dt}\vec{V_T} + m\frac{d\vec{V_T}}{dt} \tag{B.7}$$

E, como a massa é considerada constante, obtém-se:

$$\sum \Delta \vec{F} = m\frac{d\vec{V_T}}{dt} \tag{B.8}$$

Onde a derivada acima indicada é feita em relação ao referencial da Terra. Esta derivada pode ser reescrita em relação a um referencial do veículo supondo-o girando em relação ao referencial da Terra, como:

$$\frac{d\vec{V_T}}{dt} = \hat{1}_{V_T}\frac{d\vec{V_T}}{dt} + \vec{\omega} \times \vec{V_T} \tag{B.9}$$

Na expressão acima:

- $\hat{1}_{V_T}\frac{d\vec{V_T}}{dt}$ é a mudança na velocidade do centro de massa do veículo medida no referencial do veículo
- $\vec{\omega}$ é a velocidade angular total do veículo em relação à Terra

Escrevendo $\vec{V_T}$ e $\vec{\omega}$ em termos de suas componentes:

$$\vec{V_T} = u\hat{i} + v\hat{j} + w\hat{k} \tag{B.10}$$

$$\vec{\omega} = p\hat{i} + q\hat{j} + r\hat{k} \tag{B.11}$$

Onde os vetores unitários $\hat{i}$, $\hat{j}$, $\hat{k}$ apontam na direção dos eixos OX, OY, OZ, respectivamente, do veículo. Assim,

$$\hat{1}_{V_T}\frac{d\vec{V_T}}{dt} = \dot{u}\hat{i} + \dot{v}\hat{j} + \dot{w}\hat{k} \tag{B.12}$$

$$\vec{\omega} \times \vec{V_T} = (wq - vr)\,\hat{i} + (ur - wp)\,\hat{j} + (vp - uq)\,\hat{k} \tag{B.13}$$

Enquanto $\sum \Delta\vec{F}$ pode ser escrito na forma:

$$\sum \Delta\vec{F} = \hat{i}\sum \Delta F_x + \hat{j}\sum \Delta F_y + \hat{k}\sum \Delta F_z \tag{B.14}$$

Assim, combinando as equações (B.12), (B.13), (B.14), obtém-se:

$$\sum \Delta\vec{F}_x = m\,(\dot{u} + wq - vr) \tag{B.15}$$

$$\sum \Delta\vec{F}_y = m\,(\dot{v} + ur - wp) \tag{B.16}$$

$$\sum \Delta\vec{F}_z = m\,(\dot{w} + vp - uq) \tag{B.17}$$

Estas são as equações do movimento do centro de massa do veículo, ou seja, seu movimento translacional, em relação ao referencial da Terra, escritas no referencial do veículo.

Para desenvolver as equações do movimento em torno do centro de massa, obtém-se primeiro uma expressão para o momento angular $\vec{L}$. Por definição, o momento angular é o momento do momento linear de um corpo girante. O momento de um elemento de massa dm devido a uma velocidade angular $\vec{\omega}$ será igual ao produto de sua velocidade tangencial em torno do centro de rotação e do elemento de massa dm. A velocidade tangencial pode ser expressa como:

$$\vec{V}_{tan} = \vec{\omega} \times \vec{r} \tag{B.18}$$

Onde $\vec{r}$ é o vetor posição que vai do centro de rotação até o elemento de massa dm. Assim, o elemento de momento linear correspondente a este elemento de massa é:

$$d\left(m\vec{V}_{tan}\right) = (\vec{\omega} \times \vec{r})\, dm \tag{B.19}$$

E o elemento de momento angular é dado pela expressão a seguir:

$$d\vec{L} = \vec{r} \times (\vec{\omega} \times \vec{r})\, dm \tag{B.20}$$

Integrando a expressão acima, obtém-se o momento angular total procurado:

$$\vec{L} = \int \vec{r} \times (\vec{\omega} \times \vec{r})\, dm \tag{B.21}$$

Sendo o vetor posição $\vec{r}$ dado por:

$$\vec{r} = x\hat{i} + y\hat{j} + z\hat{k} \tag{B.22}$$

De forma que o produto vetorial da equação (B.20) resulta:

$$\begin{aligned} \vec{r} \times (\vec{\omega} \times \vec{r}) \;=\; & \left[(y^2 + z^2)\, p - xyq - xzr\right] \hat{i} + \\ & + \left[(z^2 + x^2)\, q - yzr - xyp\right] \hat{j} + \left[(x^2 + y^2)\, r - xzp - yzq\right] \hat{k} \end{aligned} \tag{B.23}$$

O que fornece:

$$\begin{aligned}\vec{L} &= \hat{i}\int\left[(y^2+z^2)\,p - xyq - xzr\right]dm + \\ &\quad + \int\hat{j}\left[(z^2+x^2)\,q - yzr - xyp\right]dm + \int\hat{k}\left[(x^2+y^2)\,r - xzp - yzq\right]dm\end{aligned} \tag{B.24}$$

Sabendo-se que $\int(y^2+z^2)\,dm$ é definida como o momento de inércia I_x, e que $\int(xy)\,dm$ é definida como o produto de inércia I_{xy}, e sabendo-se que o análogo se aplica às outras expressões, e sabendo-se que, por hipótese, $I_{xy} = I_{yz} = 0$, tem-se que as componentes do momento angular $\vec{L}$ são:

$$L_x = I_x p - I_{xz} r \tag{B.25}$$
$$L_y = I_y q \tag{B.26}$$
$$L_z = I_z r - I_{xz} p \tag{B.27}$$

De (B.6), deve-se calcular a taxa de variação temporal do momento angular $\vec{L}$, que varia em módulo e direção. Assim:

$$\sum\Delta\vec{\tau} = \hat{1}_L\frac{dL}{dt} + \vec{\omega}\times\vec{L} \tag{B.28}$$

Onde $\hat{1}_L\dfrac{dL}{dt}$ tem componentes:

$$\frac{dL_x}{dt} = \dot{p}I_x - \dot{r}I_{xz} \tag{B.29}$$
$$\frac{dL_y}{dt} = \dot{q}I_y \tag{B.30}$$
$$\frac{dL_z}{dt} = \dot{r}I_z - \dot{p}I_{xz} \tag{B.31}$$

E o produto vetorial indicado em (B.28) resulta:

$$\vec{\omega} \times \vec{L} = (qL_z - rL_y)\,\hat{i} + (rL_x - pL_z)\,\hat{j} + (pL_y - qL_x)\,\hat{k} \tag{B.32}$$

Escrevendo $\sum \Delta\vec{\tau}$ como:

$$\sum \vec{\Delta}\tau = \hat{i}\sum \Delta\mathscr{L} + \hat{j}\sum \Delta\mathscr{M} + \hat{k}\sum \Delta\mathscr{N} \tag{B.33}$$

Resulta:

$$\sum \Delta\mathscr{L} = \dot{p}I_x - \dot{r}I_{xz} + qr\,(I_z - I_y) - pqI_{xz} \tag{B.34}$$

$$\sum \Delta\mathscr{M} = \dot{q}I_y - pr\,(I_x - I_z) - \left(p^2 - r^2\right) I_{xz} \tag{B.35}$$

$$\sum \Delta\mathscr{N} = \dot{r}I_z - \dot{p}I_{xz} + pq\,(I_y - I_x) - qrI_{xz} \tag{B.36}$$

Estas são as equações do movimento de rotação do veículo em relação ao referencial da Terra, escritas no referencial do veículo. Somando-se às equações (B.15), (B.16), (B.17), (B.34), (B.35) e (B.36) as forças e torques aerodinâmicos, além dos termos relativos ao controle, tem-se uma descrição completa do movimento do veículo.

APÊNDICE C

TEOREMA PI

Teorema Pi 1*Seja uma função qualquer de N variáveis:*

$$f\left(P_1, P_2, \ldots, P_N\right) = 0 \tag{C.1}$$

Esta função pode ser expressa em termos de $(N-K)$ produtos Π de forma:

$$\tilde{f}\left\{\Pi_1, \Pi_2, \ldots, \Pi_{N-K}\right\} = 0 \tag{C.2}$$

Onde cada produto Π é uma combinação adimensional de grupos de K variáveis independentes umas das outras, arbitrariamente escolhidas. Isto leva a:

$$\begin{aligned}
\Pi_1 &= f_1\left\{P_1, P_2, \ldots, P_K, P_{K+1}\right\} && \text{(C.3)}\\
\Pi_2 &= f_2\left\{P_1, P_2, \ldots, P_K, P_{K+2}\right\} && \text{(C.4)}\\
&\vdots && \text{(C.5)}\\
\Pi_{N-K} &= f_{N-K}\left\{P_1, P_2, \ldots, P_K, P_{K+N}\right\} && \text{(C.6)}
\end{aligned}$$

onde K é o número de dimensões fundamentais necessárias para descrever as variáveis P.

APÊNDICE D

VARIAÇÃO TEMPORAL DA MATRIZ A

Apresentam-se aqui gráficos relativos à variação temporal da matriz A, dos modelos 2 e 3.

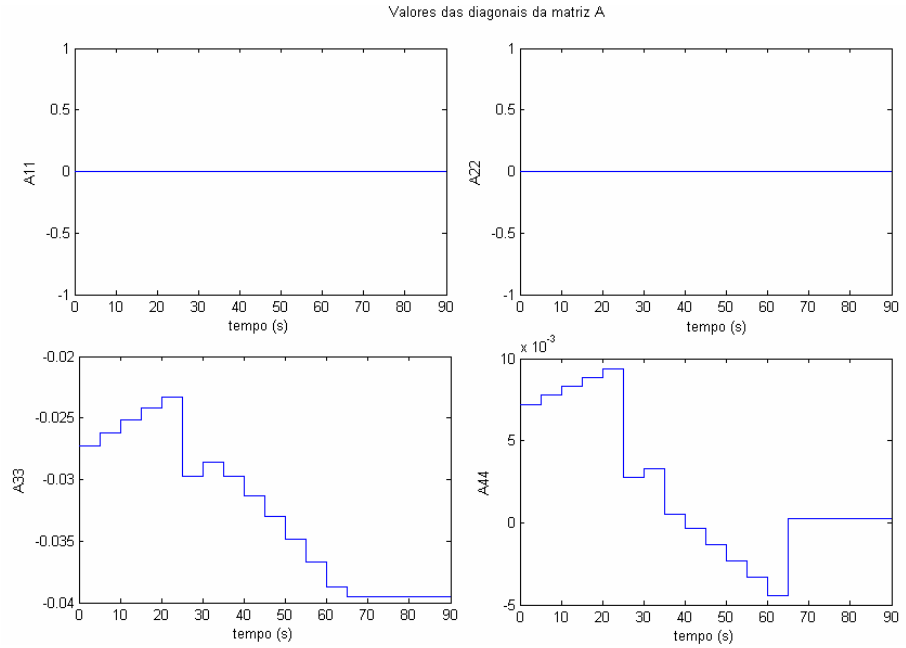


FIGURA D.1 – Variação temporal das diagonais da matriz A para o Modelo 2.

A seguir apresentamos os gráficos relativos às componentes da matriz B do Modelo 3:

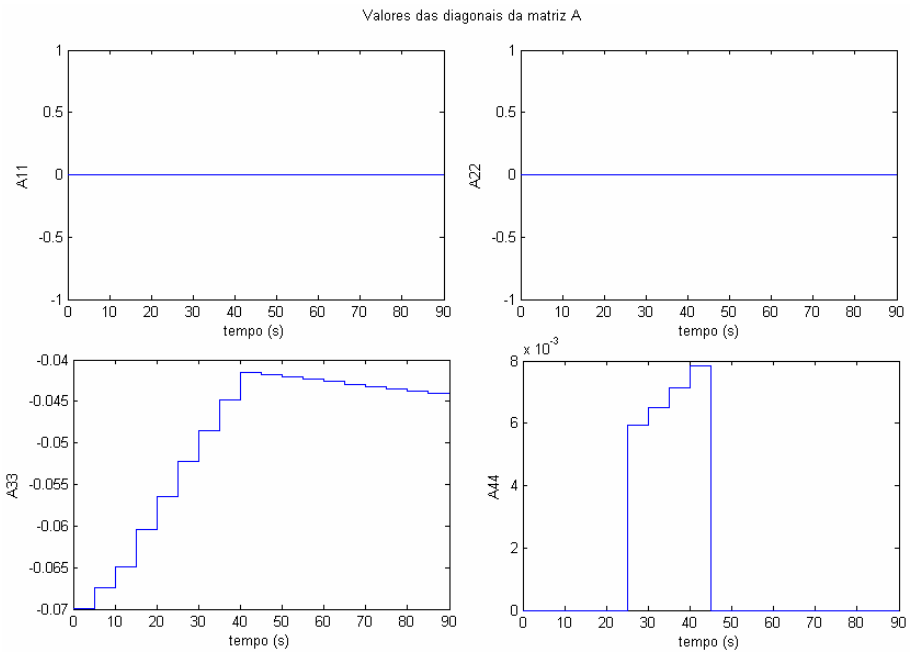


FIGURA D.2 – Variação temporal das diagonais da matriz A para o Modelo 3.

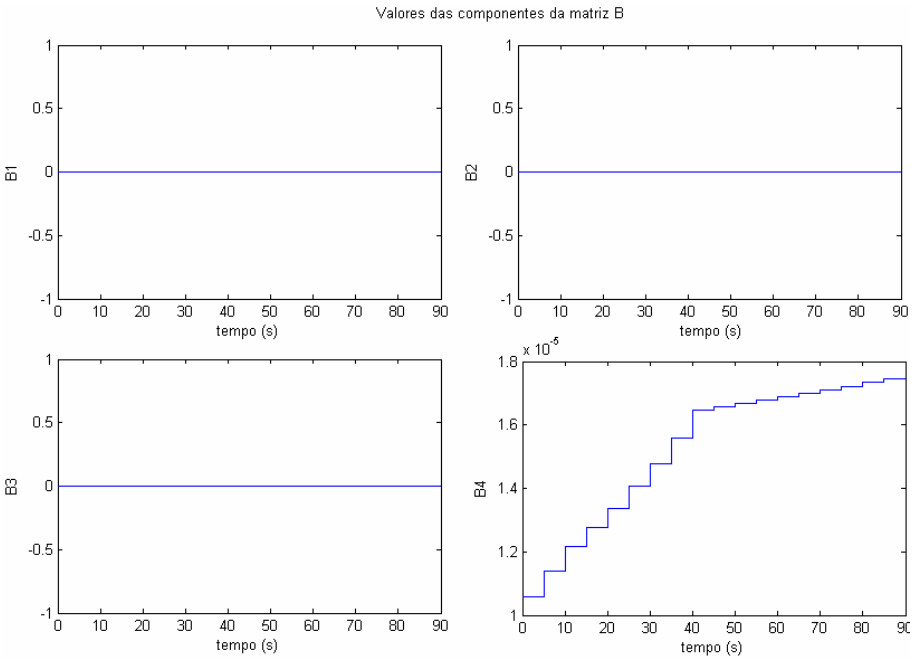


FIGURA D.3 – Variação temporal das componentes da matriz B para o Modelo 3.