



UNIVERSIDADE FEDERAL DO PARÁ
INSTITUTO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

Lucas de Carvalho Sodré

Controle Linear Quadrático Gaussiano de um Quadricóptero Baseado em um Filtro de Kalman Estendido com Variável Instrumental

DM 04/2024

Belém

2024

Lucas de Carvalho Sodré

**Controle Linear Quadrático Gaussiano de um
Quadricóptero Baseado em um Filtro de Kalman
Estendido com Variável Instrumental**

DM 04/2024

Dissertação submetida à banca examinadora do Programa de Pós-Graduação em Engenharia Elétrica da UFPA como requisito para a obtenção do Grau de Mestre em Engenharia Elétrica na área de Sistemas de Energia, na subárea Controle e Automação.

Universidade Federal do Pará

Orientador: Prof. Dr. Antonio da Silva Silveira

Belém

2024

**Dados Internacionais de Catalogação na Publicação (CIP) de acordo com ISBD
Sistema de Bibliotecas da Universidade Federal do Pará
Gerada automaticamente pelo módulo Ficat, mediante os dados fornecidos pelo(a) autor(a)**

D278c de Carvalho Sodré, Lucas.
Controle Linear Quadrático Gaussiano de um Quadricóptero
Baseado em um Filtro de Kalman Estendido com Variável
Instrumental / Lucas de Carvalho Sodré. — 2024.
85 f. : il. color.

Orientador(a): Prof. Dr. Antonio da Silva Silveira
Dissertação (Mestrado) - Universidade Federal do Pará,
Instituto de Tecnologia, Programa de Pós-Graduação em
Engenharia Elétrica, Belém, 2024.

1. Quadricóptero. 2. Múltiplas Entradas e Múltiplas
Saídas. 3. . Parâmetros Tendenciosos.. 4. Controle Linear
Quadrático. 5. Filtro de Kalman Estendido. I. Título.

CDD 003

Lucas de Carvalho Sodré

Controle Linear Quadrático Gaussiano de um Quadricóptero Baseado em um Filtro de Kalman Estendido com Variável Instrumental

Dissertação submetida à banca examinadora do Programa de Pós-Graduação em Engenharia Elétrica da UFPA como requisito para a obtenção do Grau de Mestre em Engenharia Elétrica na área de Sistemas de Energia, na subárea Controle e Automação.

Belém, 8 de fevereiro de 2024.

BANCA EXAMINADORA

Prof. Dr. Antonio da Silva Silveira - Orientador
PPGEE/UFPA

Prof. Dr. Roberto Célio Limão de Oliveira
PPGEE/UFPA

Prof. Dr. José Augusto Furtado Real
UFPA

Prof. Dr. Rodrigo Trentini Preuss
IFSC

Dedico a todos que debateram a ciência no mundo e o mundo na ciência.

Agradecimentos

Agradeço à Universidade Federal do Pará e todo o seu corpo docente pela oportunidade e suporte dados à minha formação e de milhares de pessoas. Em especial, agradeço ao Professor Antonio da Silva Silveira pela confiança, profissionalismo e dedicação.

Agradeço ao Laboratório de Controle e Sistemas e aos seus integrantes, pelas conversas e trocas de conhecimentos e bons cafés.

Agradeço à minha família pela compreensão, amor e apoio aos meus objetivos.

Agradeço ao meu Sol Julie, sendo a energia que me aquece e a esperança do amanhecer.

Agradeço a Deus por me proporcionar tudo isso e tudo o que virá.

*“Estendam as suas mãos para o que mais desejam e eu estarei com vocês! Vamos pegar o
ONE PIECE!”*

(Buggy, o Palhaço Estrela)

Resumo

Diante das transformações e fomentos de tecnologias e modernização em diversos âmbitos da sociedade, como o uso de Veículos Aéreos não Tripulados desempenhando inúmeras atividades automatizadas, faz-se necessário criar algoritmos com eficácia e segurança a fim de evitar perdas e danos em suas funções. Sistemas aéreos são, em sua maioria, sistemas de múltiplas entradas e múltiplas saídas, variantes no tempo, suscetíveis a perturbações e a ruídos de medição, se tornando um cenário desafiador na área de identificação de sistemas. Visto isso, é de extrema importância que no processo de identificação tais dinâmicas sejam consideradas. Logo, o objetivo desse trabalho é desenvolver um algoritmo capaz de estimar conjuntamente os estados e os parâmetros de sistemas, atenuando a interferência de ruídos de medição e perturbações externas no processo de identificação em tempo real. A partir desses princípios, estabeleceu-se a criação do algoritmo de estimação conjunta Filtro de Kalman Estendido com Variáveis Instrumentais. O algoritmo proposto, se destaca pelo comprometimento teórico com a minimização da interferência de dinâmicas que podem afetar a confiabilidade dos parâmetros calculados por métodos de identificação já consolidados na literatura, como Filtro de Kalman Estendido (FKE) e Mínimos Quadrados Recursivo (MQR). O método proposto foi testado para calcular o modelo linear estocástico do sistema de piloto automático do quadricóptero aéreo não tripulado modelo AR Drone 2.0 da Parrot, levando em consideração cenários em que o sinal dos sensores apresenta a relação sinal-ruído de 100, 50, 10. O seu desempenho foi comparado com a estimação de parâmetros MQR e FKE. Para avaliar as estimações de estado, utilizou-se o índice norma do desvio raiz-média-quadrado e, para avaliar os parâmetros, utilizou-se a distância Euclidiana entre os parâmetros reais e os parâmetros estimados. Por fim, os dados levantados pelos métodos foram utilizados para a sintonia do controlador Controle Linear Quadrático Gaussiano, dessa forma, permitindo comparar o impacto do método de identificação no comportamento em malha fechada do sistema aéreo. Para possibilitar a discussão e a comparação dos algoritmos de controle, foram aplicados os índices Integral do Erro ao Quadrado e Integral de Controle ao Quadrado para avaliar o desempenho de controle, a margem de ganho e a margem de fase para mensurar robustez do sistema.

Palavras-chave: Quadricóptero. Múltiplas Entradas e Múltiplas Saídas. Parâmetros Tendenciosos. Controle Linear Quadrático. Filtro de Kalman Estendido.

Abstract

Given the transformations and promotion of technologies and modernization in different areas of society, such as the use of Unmanned Aerial Vehicles performing numerous automated activities, it is necessary to create algorithms with efficiency and safety to avoid losses and damages in their functions. Aerial systems are, for the most part, multiple input and multiple output systems, time-varying, susceptible to disturbances and measurement noise, becoming a challenging scenario in the area of system identification. Given this, such dynamics must be considered in the identification process. Therefore, the objective of this work is to develop an algorithm capable of jointly estimating the states and parameters of systems, mitigating the interference of measurement noise and external disturbances in the real-time identification process. Based on these principles, the creation of the joint estimation algorithm Extended Kalman Filter with Instrumental Variables was established. The proposed algorithm stands out for its theoretical commitment to minimizing interference from dynamics that can affect the reliability of parameters calculated by identification methods already consolidated in the literature, such as Extended Kalman Filter (EKF) and Recursive Least Squares (RLS). The proposed method was tested to calculate the stochastic linear model of the autopilot system of the unmanned aerial quadcopter, Parrot's AR Drone 2.0 model, taking into account scenarios in which the sensor signal presents a signal-to-noise ratio of 100, 50, 10. Its performance was compared with RLS and EKF parameter estimation. To evaluate the state estimates, the root-mean-square deviation norm index was used and, to evaluate the parameters, the Euclidean distance between the real parameters and the estimated parameters was used. Finally, the data collected by the methods were used to tune the Gaussian Quadratic Linear Control controller, thus allowing comparison of the impact of the identification method on the closed-loop behavior of the aerial system. To enable discussion and comparison of control algorithms, the Squared Error Integral and Squared Control Integral indices were applied to evaluate the control performance, the gain margin and the phase margin to measure system robustness.

Keywords: Quadcopter. Multiple Inputs and Multiple Outputs. Biased Parameters. Linear Quadratic Gaussian. Extended Kalman Filter.

Lista de ilustrações

Figura 1.	Modelo ARMAX	25
Figura 2.	Diagrama FKEVI	38
Figura 3.	Diagrama FKEVI+FK	42
Figura 4.	Diagrama FKEVI+FK com Sistema MIMO	43
Figura 5.	Local de Testes	49
Figura 6.	AR.Drone 2.0 Voando	50
Figura 7.	Extração de Dados	52
Figura 8.	Orientação das dinâmicas do Parrot® AR Drone seguindo o sistema NED (North-East-Down)	52
Figura 9.	Sinal PRBS	56
Figura 10.	Parâmetros	58
Figura 11.	Ângulos	58
Figura 12.	Velocidades	59
Figura 13.	Altitude	59
Figura 14.	Ângulos com SNR 10	61
Figura 15.	Velocidades com SNR 10	62
Figura 16.	Altitude com SNR 10	62
Figura 17.	Parâmetro a_{41} no cenário de SNR 10	64
Figura 18.	LQG Ideal	67
Figura 19.	Curvas T e S	68
Figura 20.	Malha de Velocidade Frontal	69
Figura 21.	Malha de Velocidade Lateral	69
Figura 22.	Diagrama do LQG com o FKEVI+FK	70
Figura 23.	Diagrama do LQG com FKE	71
Figura 24.	Curvas S e T dos Controladores Adaptativos	71
Figura 25.	Malha de Velocidade Frontal	72
Figura 26.	Malha de Velocidade Lateral	72

Lista de tabelas

Tabela 1.	NRMSE Referencial	56
Tabela 2.	Variância da Perturbação	57
Tabela 3.	Parâmetros Estimados pelo FKE e FKEVI+FK	59
Tabela 4.	Ruídos Estimados pelo FKEVI+FK	60
Tabela 5.	Potência dos Ruídos	61
Tabela 6.	Índices NRMSE dos Experimentos de Identificação na Presença de Ruído	62
Tabela 7.	Parâmetros Estimados na Presença de Ruído.	63
Tabela 8.	A Distância Euclidiana na Presença de Ruído.	63
Tabela 9.	Estimações dos Ruídos pelo FKEVI+FK	65
Tabela 10.	Resultados Referência de Controle	69
Tabela 11.	Resultados do Controle Adaptativo	72

Lista de abreviaturas e siglas

ARMAX	Autorregressivo com Média Móvel e Entradas Exógenas
ARX	Autorregressivo com Entradas Exógenas
FKE	Filtro de Kalman Estendido
FKEVI	Filtro de Kalman Estendido com Variáveis Instrumentais
GMVSS	Controle de Variância Mínima Generalizado em Espaço de Estados
ISE	Integral do Erro Quadrático
ISU	Esforço de Controle Quadrático
KF	Filtro de Kalman
LQG	Controle Linear Quadrático Gaussiano
MIMO	Múltiplas Entradas e Múltiplas Saídas
MPC	Controle Preditivo baseado em Modelo
MQ	Mínimos Quadrados
MQR	Mínimos Quadrados Recursivo
MQRE	Mínimos Quadrados Recursivo Estendido
NRMSE	Média do Quadrado da Raiz do Erro Normalizado
PRBS	Sequência Binária Pseudoaleatório
SISO	Uma Entrada e Uma Saída
SNR	Relação Sinal Ruído
VANT	Veículos Aéreos não Tripulados
VI	Variáveis Instrumentais
WiFi™	Wireless Fidelit

Lista de símbolos

ρ	Ângulo de Rolagem
δ	Ângulo de Arfagem
v_l	Velocidade Lateral
v_f	Velocidade Frontal
h	Altura
r	Sinal de referência
u_ρ	Impulso de Rolagem
u_δ	Impulso de Arfagem
u_h	Impulso de Altura
k	Tempo Discreto
d	Atraso Puro de Tempo Discreto
T_s	Tempo de Amostragem
w	Perturbação
v	Ruído
e	Erro de Estimação

Sumário

1	INTRODUÇÃO	15
1.1	Motivação	15
1.2	Objetivos	18
1.2.1	Objetivo Principal	18
1.2.2	Objetivos Específicos	18
1.3	Revisão Bibliográfica	18
1.4	Contribuição Científica	21
1.5	Estrutura do Trabalho	21
2	FUNDAMENTOS TEÓRICOS	23
2.1	Métodos de Estimação	23
2.2	Modelo do Sistema Discreto	24
2.3	O Estimador de Mínimos Quadrados	25
2.4	O Estimador de Mínimos Quadrados Recursivo	28
2.5	O Estimador Recursivo Estendido de Mínimos Quadrados	30
2.6	Variáveis Instrumentais	31
2.7	Métodos de Estimação de Estados	33
2.7.1	Filtro de Kalman	33
2.7.2	Filtro de Kalman Estendido	36
2.8	Filtro de Kalman Estendido Instrumental	37
2.9	Regulador Linear Quadrático e Gaussiano	43
2.10	Índices de Análise de Desempenho e de Robustez	45
3	MATERIAIS E MÉTODOS	49
3.1	Ambiente Físico	49
3.2	Quadricóptero	50
3.3	PS-Drone	51
3.4	Modelo linear	52
4	EXPERIMENTOS DE IDENTIFICAÇÃO	55
4.1	Modelo Linear de Referência	55
4.2	Resultados da Identificação	60
5	RESULTADOS DE CONTROLE	66
6	CONCLUSÃO	74

REFERÊNCIAS	76
ANEXO A – PROGRAMA EM PYTHON 3 APLICADO PARA EXTRAIR E ENVIAR DADOS EXPERIMENTAIS DO AR.DRONE 2.0.	79
ANEXO B – CÁLCULO DA CURVA DE RESPOSTA EM FREQUÊN- CIA DE SISTEMAS LINEARES NO ESPAÇO DE ESTADOS	86

1 Introdução

A Teoria de Controle é uma área da matemática aplicada que visa a análise e o projeto de sistema de controle, dessa forma permitindo que o processo apresente o comportamento desejado. Os responsáveis por esses projetos devem se atentar às diversas dinâmicas as quais o processo está inserido, podendo este ser um sistema linear ou não linear, estável ou instável, variante ou invariante no tempo, de múltiplas entradas e múltiplas saídas (MIMO). Além disso, esses sistemas podem conter ruídos nos sensores e perturbações externas, sendo assim o projetista deve buscar métodos mais adequados às peculiaridades do processo.

Neste capítulo, são abordados os motivos e problemas envolvidos no trabalho, bem como os objetivos e a revisão literária necessária ao desenvolvimento da dissertação. Também foi criada uma breve abordagem dos principais resultados, contribuições e conclusões do estudo.

1.1 Motivação

A modelagem matemática é a área do conhecimento que estuda formas de construção e implementação de modelos matemáticos de sistemas experimentais, tendo três categorias principais de modelagem de sistemas: caixa preta, caixa cinza e caixa branca. Na modelagem caixa branca, todas as leis físicas e parâmetros do sistema são conhecidos; na modelagem caixa cinza, apenas as leis são conhecidas, porém, os parâmetros não; e na modelagem caixa preta, não se tem conhecimento dos parâmetros e das leis físicas que regem o sistema. Tais parâmetros estimados, nos diferentes processos de modelagem, tem o objetivo de compreender as dinâmicas dos sistemas, logo, a partir dos parâmetros é possível analisar o comportamento do processo (OGATA, 1996).

A partir do modelo, diversas técnicas de controle podem ser empregadas para moldar o comportamento do sistema, fazendo com que ele se comporte de maneira coerente com os objetivos estabelecidos pelo projeto de controle. Uma delas é o Regulador Linear Quadrático (LQR), no qual utiliza os parâmetros do sistema linear para encontrar a sintonia ótima do controlador, tendo como características a possibilidade de ter seus ganhos estáticos ou adaptativos e ser projetado tanto para sistemas de uma entrada e uma saída quanto para sistemas de múltiplas entradas e múltiplas saídas.

Visto isso, é crucial um aprofundamento teórico e prático nas áreas de identificação e controle de sistemas se o processo envolvido apresenta atributos desafiadores como não linearidade, parâmetros variantes no tempo, múltiplas entradas, múltiplas saídas, ruídos

nos sensores e perturbações externas, principais características de sistemas aeroespaciais. Recentemente, os sistemas aeroespaciais chamados de Veículos Aéreos Não Tripulados (VANTs) têm atraído um interesse crescente de pesquisadores devido às dinâmicas desafiantes (ELKHATEM; ENGIN, 2022), às incertezas, às perturbações e à rápida mudança a que estão sujeitos. Encontram-se vários exemplos de aplicações como vigilância militar e civil, logística, monitoramento de incêndios florestais, (CASBEER DEREK B. KINGSTON; MCLAIN, 2006), coleta de dados, aquisição de dados (MUNGUÍA; URZUA; GRAU, 2019), rede de comunicação alimentada sem fio (WANG et al., 2021) e assim por diante. O veículo quadricóptero é um dos mais adequados para o cumprimento dessas tarefas devido às suas características peculiares como tamanho reduzido, facilidade de decolagem e pouso em um espaço pequeno e menor complexidade mecânica e de hardware quando comparado a outros VANTs.

As técnicas de controle e identificação são usadas para controlar e modelar as dinâmicas do voo do veículo aéreo. Porém, as perturbações e ruídos dos sensores que afetam o VANT podem comprometer a fidelidade dos parâmetros estimados pelo algoritmo de identificação, tornando-os tendenciosos (polarizados) e conseqüentemente as estimações dos estados baseados em tais parâmetros são incertas. Em um cenário onde os estados e os parâmetros estimados estão comprometidos, os ganhos dos controladores calculados com base nesses dados resultam em um comportamento indesejado que muitas vezes podem ser desastrosos. Logo, é de extrema importância que algoritmos de estimação paramétrica e de estados atenuem a polarização, contribuindo para a robustez de sintonia do controle.

Em face desse cenário problemático, alguns questionamentos foram levantados: É possível estabelecer um algoritmo capaz de oferecer parâmetros e estados não polarizados por dinâmicas que interferem sistemas multivariáveis como o dos VANTs? E, caso seja possível, isso traria benefícios para a sintonia de controladores adaptativos? Será que essas técnicas combinadas podem oferecer robustez e desempenho melhorado ao projeto de sistema de controle em malha fechada do sistema de piloto automático de quadricópteros?

Para responder às questões, foram formuladas as seguintes hipóteses:

1. Uma vez que aeromodelos são sistemas MIMO e que estão inseridos em dinâmicas externas, como perturbações e ruídos de medição, e existem técnicas que reduzem as interferências no processo de estimação paramétrica e de estados, é razoável afirmar que utilizar técnicas não polarizantes pode melhorar a robustez nas estimativas das dinâmicas de sistemas MIMO.

- Hipótese: “O algoritmo de identificação não polarizante atenua a interferência de dinâmicas indesejadas nos dados estimados para VANTS”.

2. Em uma análise comparativa entre os parâmetros polarizados e os parâmetros não polarizados, estes descrevem melhor o comportamento de um processo do que aqueles.

Logo, utilizá-los pode sintonizar observadores de estado melhores que os demais. Portanto, algoritmos de estimação conjunta podem ser melhorados com a adição de métodos capazes de enfrentar dinâmicas que afetam negativamente o processo de estimação.

- Hipótese: “Projetar algoritmos de estimação conjunta com técnicas não polarizantes tem desempenho melhor dos que os que não são auxiliados, como o Filtro de Kalman Estendido.”

3. Em contraponto às técnicas tradicionais de controle adaptativo ao sistema de piloto automático do quadricóptero, o algoritmo de controle adaptativo proposto de características não polarizantes é mais robusto e dispõe de melhor desempenho.

-Hipótese 3: "O algoritmo proposto traz mais robustez e desempenho ao projeto de controle malha fechada de VANTs do tipo quadricóptero”.

Dadas as hipóteses, o que motiva este trabalho é a criação de um algoritmo capaz de fornecer dados que se aproximam do sistema real, evitando tendências que comprometam a fidelidade dos mesmos, dessa forma identificando as dinâmicas internas e externas do quadricóptero. A partir desses dados, sintonizar o Regulador Linear Quadrático que ofereça melhor desempenho e robustez.

A ideia do parágrafo acima expõe um dos principais “calcanhares de Aquiles” da Teoria de Controle: Como garantir que a solução de um controlador adaptativo não leve o sistema à instabilidade? Pois controladores adaptativos podem ser sintonizados com dados polarizados, resultando em sintonias de controle sub-ótimas, dessa forma tendo um comportamento em malha fechada indesejado, podendo até mesmo levar o processo à instabilidade. Porém, a aplicação de técnicas adaptativas pode trazer benefícios únicos, tais como: preservação dos critérios do projetista, maior economia no esforço de controle e menos erros entre a saída real e a saída desejada.

A fim de contribuir na solução do questionamento acima, este trabalho propõe uma abordagem que utiliza técnicas de identificação que diminuem a diferença entre os parâmetros reais e os calculados, trazendo maior confiança nas estimações de estado e na sintonia do controlador adaptativo. O algoritmo resultante dessa técnica é aplicado em um quadricóptero que tem como característica múltiplas entradas e múltiplas saídas, ruído nos sensores, é suscetível a perturbações externas e seu comportamento é variável no tempo, permitindo que a técnica proposta atenuie as adversidades e aumente os benefícios.

1.2 Objetivos

1.2.1 Objetivo Principal

Desenvolver técnicas de identificação de sistemas em tempo real que ofereça parâmetros não polarizados e estimativas precisas, e em posse desses dados, sintonizar um controlador adaptativo robusto para o sistema de piloto automático do quadricóptero AR.Drone 2.0, melhorando assim o desempenho qualitativo e quantitativo do processo em malha fechada.

1.2.2 Objetivos Específicos

- Enriquecer as experiências de aprendizagem fornecendo alternativas para que o pesquisador, estudante e/ou profissional compreenda e relacione os resultados obtidos e os conceitos vinculados à fundamentação teórica do experimento;
- Disponibilizar material desenvolvido para ministrar ensino sobre o uso do sistema comunicador em Python para Parrot® AR Drone 2.0;
- Explorar a ligação entre a identificação de sistemas e o controle adaptativo, que apresenta vários obstáculos e desafios.
- Permitir experiências com quadricópteros contribuindo para a formação de pessoal em nível regional e nacional;
- Fornecer modelos lineares para controle do Parrot® AR Drone 2.0, bem como introduzir métodos de identificação de sistemas como a técnica proposta;
- Investigar a robustez de controladores ótimos a partir de algoritmos adaptativos com técnicas de identificação não polarizante e associado ao LQG.

1.3 Revisão Bibliográfica

Por mais que este trabalho finalize com controle adaptativo, o foco principal está na identificação de sistemas MIMO afetados por ruídos e perturbações, sendo o Controle Linear Quadrático Gaussiano (LQG) adaptativo projetado com base nos resultados dessa identificação. Portanto a revisão tende a ser mais focada em estudos na extensa área de identificação de sistemas de VANTS. A identificação de sistemas quadricópteros, ou a técnica de obter modelos matemáticos desses sistemas físicos a partir dos dados medidos de entrada-saída, vem se desenvolvendo desde 1965 (SHRAIM; AWADA; YOUNESS, 2018). No entanto, veículos aéreos do tipo supracitado têm acoplamento não linear, múltiplas entradas e múltiplas saídas e tendem a apresentar características em voo que variam ao

longo do tempo, sendo assim um problema desafiador que compõe o estado-da-arte na área de controle e sistemas.

Por isso, técnicas de estimação paramétrica como Mínimos Quadrados (ILIFF, 1986), Gauss-Newton (CAMPBELL; BRUNKE, 2001) e Filtro de Kalman (JATEGAONKAR; PLAETSCHKE, 1989) têm seus parâmetros deteriorados quando as características não lineares não podem ser negligenciadas. Para controlar efetivamente tais veículos, torna-se necessário projetar e desenvolver controladores robustos e adaptativos e, assim, identificando o sistema em tempo real (KALLAPUR; ALI; ANAVATTI, 2007) e linearizando o sistema a cada amostra, fazendo com que o modelo linear estimado se aproxime do comportamento do quadricóptero. Dessa forma, mantendo as características de projeto no comportamento em malha fechada do sistema.

No artigo de Nogueira et al. (2023), o mesmo quadricóptero utilizado nessa dissertação é identificado pelo método caixa preta aplicando a técnica de Mínimos Quadrados (MQ), são levantados o modelo Autorregressivo com Entradas Exógenas (ARX) e o modelo Autorregressivo com Média Móvel e Entradas Exógenas (ARMAX), na qual o primeiro modelo considera a saída como uma função da entrada e erro de medição como erro na equação, o segundo considera a que a saída depende da entrada e de regressores do sinal erro da equação. Tal consideração feita pelo modelo ARMAX pode fazer com que o desempenho de identificação melhore consideravelmente em relação ao modelo ARX. Esse desempenho se justifica pelo fato de que o drone está exposto às mais diversas perturbações durante o voo, e é de extrema importância modelar essas dinâmicas. Outro fator importante desse trabalho é que a biblioteca utilizada entrega dados de saída do quadricóptero já filtrados, assim a técnica de Mínimos Quadrados oferece parâmetros não polarizados (ou pouco polarizados). Porém, VANTs, em geral, tem dinâmicas que variam com tempo, sendo necessário fazer uso de técnicas capazes de estimar parâmetros em tempo real.

No trabalho de Dhaybi e Daher (2019), o método de identificação em tempo real baseado no Mínimos Quadrados é aplicado para calcular os momentos inerciais de quadricópteros. O algoritmo utilizado chama-se Mínimos Quadrados Recursivo (MQR) e ele é encarregado de atualizar os parâmetros amostra por amostra, gerando o modelo linear do quadricóptero durante uma tarefa na qual se desloca uma carga. Porém, a identificação via MQR tem restrições, e se não respeitadas, o algoritmo gera parâmetros tendenciosos, assim comprometendo o modelo utilizado como base da técnica de controle. Tais restrições mencionadas serão debatidas no decorrer do trabalho. Além disso, o MQR oferece a saída estimada que não pondera os ruídos e as perturbações a que sistemas aéreos estão sujeitos.

Uma das técnicas capazes de solucionar esse problema é chamada de Filtro de Kalman Estendido (FKE), sendo considerada uma boa técnica para identificação do sistema de quadricópteros, pois lineariza o sistema amostra por amostra e estima parâmetros

e os estados do processo. Esta técnica visa atenuar a interferência as não linearidades do sistema a partir da derivação ponto a ponto da equação de estados estabelecida pelo método, permitindo assim que o parâmetro e o estado sejam estimados simultaneamente.

No trabalho de Munguía, Urzua e Grau (2019), o Filtro de Kalman Estendido foi aplicado no sistema de um quadricóptero, obtendo bons resultados nos quesitos de identificação do modelo, permitindo a filtragem das informações advindas de sensores e estimação paramétrica. Todavia, essa técnica de estimação conjunta de estados e parâmetros do quadricóptero necessita da variância da perturbação, do ruído de medição e da variabilidade dos parâmetros do sistema para ser projetada. Porém, em muitos casos, tais informações não estão à disposição do projetista, sendo necessário supor o comportamento de tais dinâmicas, resultando na estimação dos estados (e parâmetros) mal condicionada.

Outra técnica utilizada é o Filtro de Kalman Descentralizado (FKD). Esse método de identificação permite a estimação conjunta de parâmetros e estados do sistema e assim filtrando a saída dos sensores, similar ao método FKE, porém para estabelecer as matrizes de covariância necessárias para o algoritmo, exige-se que sejam efetuadas simulações muitos passos à frente do sistema no estado atual. Essa técnica foi aplicada para identificar o modelo do quadricóptero no artigo Svacha et al. (2020), onde se conhece as leis físicas do processo e as medidas inerciais do drone (caixa cinza). Em posse desses dados e o algoritmo de identificação FKD, estimaram-se os 14 parâmetros restantes do modelo, tendo um desempenho melhor que o FKE devido à robustez de identificação. Em contrapartida, o esforço computacional para calcular as matrizes necessárias pode ser considerado alto para serem calculadas a cada tempo de amostragem. Visto isso, algumas aplicações se restringem apenas em ambientes de simulação, como no trabalho de Kaba e Kiyak (2020), dessa forma o resultado não sendo calculado em tempo real, condição necessária para a aplicação do controle adaptativo.

Os conhecimentos debatidos nessa seção são a base do estado-da-arte na área de identificação conjunta em tempo real, obtendo os requisitos necessários para fusão com a área de controle. Essa fusão teve como resultado trabalhos como o de Chikasha e Dube (2017), onde o método de identificação MQR foi aplicado no projeto de controle preditivo baseado em modelo (MPC). O MQR foi usado para estimar os parâmetros amostra por amostra gerando o modelo atualizado que o controlador preditivo se baseava, tornando o sistema em malha fechada adaptativo. Porém, a intensidade dos ruídos de medição pode acentuar dinâmicas indesejadas em malha fechada. No trabalho de Lv et al. (2019) tais problemas são enfrentados, pois o FKE oferece os parâmetros estimados e as saídas filtradas para o algoritmo do MPC, dessa forma calculando os ganhos, a filtragem permite uma maior robustez e menor esforço de controle.

De fato, todos os trabalhos mencionados apresentam sólidas análises e resultados, alicerces para futuras aplicações na aeroespacial. Mas, uma pergunta intriga o autor dessa

dissertação: tais resoluções podem fornecer parâmetros não polarizados? Pois tanto o FKE e MQR são algoritmos estimadores recursivos ótimos (AGUIRRE, 2015), e tais características não garantem estimações paramétricas não tendenciosas.

Visto isso, nesse trabalho é estruturada e debatida a técnica proposta que, de maneira teórica, atenua a polarização, propiciando à área de identificação de sistemas um algoritmo capaz de estimar parâmetros e estado em tempo real que se aproximem das reais dinâmicas as quais o quadricóptero está inserido, e a partir desse algoritmo sintonizar um controle adaptativo que ofereça melhor robustez e desempenho.

1.4 Contribuição Científica

Nesta dissertação, o método de identificação proposto estima parâmetros, estados, intensidade do ruído, intensidade das perturbações de sistemas lineares auxiliado por técnicas não polarizantes e, em posse desses dados, é sintonizado um LQG adaptativo para controlar as dinâmicas de piloto automático de um quadricóptero. Dessa forma, contribuindo para a área de pesquisa de identificação de VANTs desenvolvendo uma técnica adequada às suas características dinâmicas.

1.5 Estrutura do Trabalho

O Capítulo 1 corresponde à introdução deste trabalho.

No Capítulo 2, compila-se o arcabouço teórico dos métodos de estimação utilizados, abordando métodos de estimação paramétrica, métodos não polarizados de estimação, métodos de estimação conjunta e, ao final, é construído o algoritmo de estimação conjunta não polarizante proposto. Após isso, são apresentados os controladores LQR e LQG, e por fim os índices que serão utilizados para avaliar os sistemas projetados.

No Capítulo 3 mostram-se os equipamentos e métodos utilizados para extrair e processar os dados dos experimentos. Debate-se a estrutura do quadrimotor modelo AR.Drone 2.0 da Parrot, como é feita a comunicação com o *firmware* do drone e o ambiente em que foram feitos os testes. Ao final, é apresentado o modelo linear do quadrimotor, o qual será feita a estimação conjunta dos estados e parâmetros.

No Capítulo 4 são feitos experimentos relacionados aos processos de identificação do quadricóptero. Com base nesses dados, são calculados os índices em que se baseia a avaliação do desempenho dos métodos testados. Ao final, são discutidas as características de cada método e estabelecido o modelo de referência.

No Capítulo 5 são definidos o LQG ideal e os controles adaptativos sintonizados com os parâmetros provenientes dos experimentos de identificação. A partir dos dados do

experimento de identificação feitos no capítulo anterior, são feitos experimentos simulados utilizando o LQG adaptativo auxiliado pelos métodos de estimação conjunta apresentados nesta dissertação. Os desempenhos de controle, assim como a robustez dos projetos de controle, são avaliados por índices e gráficos. Devido ao método de estimação conjunta proposto estar em fase de consolidação, optou-se pelo ambiente simulado.

Por fim, no Capítulo 6 as conclusões do trabalho são relatadas, abrindo margens à produção de trabalhos futuros correlacionados.

2 Fundamentos Teóricos

Na primeira parte desse capítulo, é feito um pequeno dicionário da área de identificação de sistema que facilita o entendimento das técnicas utilizadas para estimar os parâmetros e estados do sistema do quadricóptero. Mais adiante são apresentados os métodos já consolidados na área, como: Mínimos Quadrados, Mínimos Quadrados Recursivo, Mínimos Quadrados Recursivo Entendido, Variáveis Instrumentais e Filtro de Kalman Estendido. Por fim, é fundamentado o algoritmo de identificação proposto.

Após apresentados os métodos de identificação, os controladores LQR e LQG são abordados, explicando os seus fundamentos e aplicações. Ao final, são explicados os índices utilizados para avaliar o processo de identificação, o desempenho e robustez de controle.

2.1 Métodos de Estimação

O processo de identificação de sistemas obtém resultados melhores se as características dos processos forem contempladas pelo método de identificação aplicado (COELHO; COELHO, 2015). Visto isso, diversos fatores inerentes ao processo devem ser considerados durante o processo de escolha do tipo da identificação, sendo eles: atributos do sistema (linearidade, grau de não linearidade, atraso de transporte); as propriedades do modelo do sistema (complexidade); a seleção da medida do erro a ser minimizado; a presença de ruídos.

Dada a vastidão de algoritmos de identificação de sistema, é necessário construir um pequeno dicionário de termos da área, tornando a explicação de métodos utilizados mais fácil e compreensível. Logo, os métodos podem ser:

- Caixa branca: corresponde à modelagem fenomenológica da planta;
- Caixa cinza: quando o projetista tem algumas informações a respeito do comportamento da planta, exemplo: alguns parâmetros, leis físicas, etc.;
 - Este é o utilizado para estimar o modelo linear do drone utilizado.
- Caixa preta: quando o projetista tem apenas os sinais de entrada e saída da planta e não possui conhecimento do comportamento dela;
- Contínua: gera modelos no domínio contínuo;
- Determinística: levanta modelos em que as relações entre entrada e saída são as únicas bases de dados, sem considerar incertezas probabilísticas;

- Discreta: gera modelos no domínio discreto;
 - Como os dados provenientes da planta são todos discretos, então a identificação discreta será utilizada com o propósito de melhores modelos serem obtidos.
- Em malha direta: o projetista não inclui a retroalimentação na aquisição dos dados;
- Em malha fechada: o projetista usa a retroalimentação para a aquisição de dados.
- Estendida: levanta um modelo para descrever o comportamento da planta a estímulos determinísticos e estocásticos;
 - Esta dissertação utiliza esse método, pois dinâmicas estocásticas, se negligenciadas, podem comprometer o desempenho de identificação.
- Offline, não recursiva ou batelada: levanta um modelo para um conjunto de dados disponíveis;
- Online ou recursiva: capaz de atualizar o modelo da planta a cada iteração ou conjunto de iterações (épocas);
 - O drone foi estimado utilizando técnicas recursivas, pois o sistema é variável no tempo e o controle adaptativo utiliza tais técnicas.
- Ordinária: comum, clássica.

2.2 Modelo do Sistema Discreto

Os modelos estabelecem diversas relações do sistema. Tais relações promovem um direcionamento teórico do método de estimação. Considere o modelo escrito na equação (2.1).

$$A(q^{-1})y(k) = B(q^{-1})q^{-d}u(k) + C(q^{-1})w(k) \quad (2.1)$$

onde $y(k)$ é a saída do sistema, $u(k)$ é a entrada, $w(k)$ é a perturbação, d é o atraso de transporte, k é o operador do tempo discreto, q^{-1} é o operador do tempo discreto e $A(q^{-1})$, $B(q^{-1})$, $C(q^{-1})$ são polinômios discretos e n_a , n_b e n_c são seus graus respectivamente. Os polinômios estão representados nas equações abaixo.

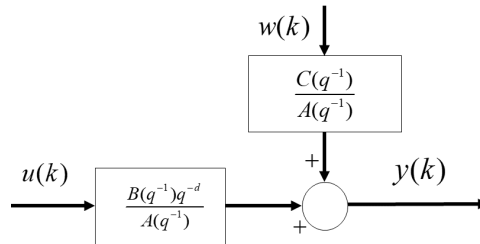
$$A(q^{-1})y(k) = 1 + a_1y(k-1) + \dots + a_{n_a}y(k-n_a) \quad (2.2)$$

$$B(q^{-1})u(k) = b_1u(k-d) + \dots + b_{n_b}u(k-n_b-d) \quad (2.3)$$

$$C(q^{-1})w(k) = 1 + c_1w(k-1) + \dots + w_{n_c}w(k-n_c) \quad (2.4)$$

Tal modelo é denominado como um sistema linear ARMAX. Sendo assim, os processos caracterizados com esse modelo têm perturbações Gaussianas, entradas exógenas e o seu estado atual está em função de estados anteriores (regressivo). O sistema do drone é estimado com base nesse modelo, no qual está representado na Figura 1.

Figura 1. Modelo ARMAX



Fonte: elaborado pelo autor do texto

É importante salientar que, para um sistema linear estabelecer casualidade, as ordens dos polinômios $B(q)$ e $C(q)$ devem ser menores ou iguais à ordem de $A(q)$, dessa forma possibilitando o controle do sistema e a estimação dos parâmetros dos regressores presentes na equação 2.5.

$$y(k) = -a_1y(k-1) - \dots - a_{n_a}y(k-n_a) + b_1u(k-d) + \dots \quad (2.5)$$

$$+ b_{n_b}u(k-n_b-d) + w(k) + c_1w(k-1) + \dots + c_{n_c}w(k-n_c)$$

Vale ressaltar que o polinômio descrito se trata de um sistema de uma entrada e uma saída (SISO), diferente do sistema MIMO do drone. Porém, é mais simples desenvolver os conceitos dos métodos de identificação em regressores SISO e, no decorrer da dissertação, apresentar como tais algoritmos são escritos nos modelos que melhor representam os sistemas multivariáveis, o sistema em espaço de estados.

2.3 O Estimador de Mínimos Quadrados

O Método de Mínimos Quadrados é um dos mais populares métodos de identificação devido às diversas áreas nas quais ele é aplicado. A origem do método está datada no ano de 1812 nos trabalhos de Gauss relacionados à trajetória dos astros. O resumo que melhor sintetiza a ideia básica (de certa maneira até poética) encontrado pelo autor desta dissertação está no seguinte trecho de Sorenson (1970):

"Se as observações astronômicas e outras grandezas nas quais se baseiam o cálculo de órbitas fossem absolutamente corretas, o resultado, ainda que obtido a partir de três ou quatro observações, também seria estritamente correto (na medida em que se supõem que o movimento ocorre exatamente de acordo com as leis de Kepler), e, portanto, o uso de outras medições seria útil apenas para confirmação, mas desnecessário para correção ou ajuste. Mas uma vez que todas as nossas medições e observações nada mais são do que aproximações da realidade, o mesmo vale para os resultados que se baseiam nelas, e o alvo mais elevado de todos os cálculos feitos relacionados a fenômenos concretos deve ser o de se aproximar, tanto quanto possível, da verdade. Isso, entretanto, não pode ser alcançado a não ser pela adequada combinação de observações adicionais àquelas estritamente necessárias à determinação das grandezas desconhecidas. Esse problema só pode ser resolvido apropriadamente quando um conhecimento aproximado da órbita já foi obtido, conhecimento esse que posteriormente deve ser ajustado de forma a satisfazer todas as observações da forma mais exata possível."

É importante ressaltar que a técnica de identificação por Mínimos Quadrados (MQ) tem o intuito de gerar parâmetros que melhor descrevam as observações contaminadas por incertezas como erros de medição, não linearidade, dinâmica não modelada, etc. Entre outras palavras, a diferença entre o cálculo utilizando os parâmetros estimados e as amostras deve ser mínima. Visto isso, a equação (2.6) aponta as variáveis que estão relacionadas na identificação via mínimos quadrados.

$$\phi^T(k) = [-y(k-1) \quad \cdots \quad -y(k-n_a) \quad u(k-d) \quad \cdots \quad u(k-n_b-d)] \quad (2.6)$$

$$y(k) = \phi^T(k)\hat{\theta}(k-1) + e(k) \quad (2.7)$$

onde ϕ é o vetor de regressores que tem a mesma dimensão do número de parâmetros estimados pelo método. Já o $\hat{\theta}$ é o vetor onde estão armazenados parâmetros estimados, $e(k)$ o erro de regressão e k o instante de tempo discreto da equação. Escrevendo a equação (2.6) para N medidas, resulta em uma forma matricial presente na equação (2.8).

$$Y = \Phi\hat{\theta} + \xi \quad (2.8)$$

$$\xi = Y - \Phi\hat{\theta} \quad (2.9)$$

onde $Y \in \mathbb{R}^N$ é o vetor de saídas e $\xi \in \mathbb{R}^N$ é o vetor de resíduos de estimação e Φ é a matriz

formada pelos vetores de regressores de cada amostra. O objetivo do MQ é encontrar o conjunto de parâmetros que reduz a magnitude do vetor de resíduo. Para isso, é projetada a equação custo com base no somatório dos erros quadráticos presente na equação (2.10).

$$J_{MQ}(\hat{\theta}) = \sum_{i=1}^N e(i)^2 = \xi^T \xi = \|\xi\|^2 \quad (2.10)$$

Substituindo ξ por $\xi = Y - \Phi\hat{\theta}$ temos.

$$J_{MQ} = (Y - \Phi\hat{\theta})^T (Y - \Phi\hat{\theta}) \quad (2.11)$$

$$J_{MQ} = Y^T Y - Y^T \Phi \hat{\theta} - \hat{\theta}^T \Phi^T Y + \hat{\theta}^T \Phi^T \Phi \hat{\theta} \quad (2.12)$$

De modo a minimizar a função custo J_{MQ} com respeito a $\hat{\theta}$, deriva-se a função em relação ao $\hat{\theta}$ e iguala-se a zero, dessa forma encontrando o ponto de inflexão da curva função custo resultando na equação onde se encontra o valor mínimo da função como demonstrado na equação (2.13).

$$\frac{\partial J_{MQ}}{\partial \hat{\theta}} = -(Y^T \Phi)^T - \Phi^T Y + (\Phi^T \Phi + \Phi^T \Phi) \hat{\theta} = 0 \quad (2.13)$$

Após isso, aplicam-se as operações matemáticas presentes em Aguirre (2015) resultando na equação (2.14).

$$\hat{\theta} = [\Phi^T \Phi]^{-1} \Phi^T Y \quad (2.14)$$

A solução de parâmetros estimados apresentada tem a menor magnitude de erro, logo se alinhando aos parâmetros reais do sistema se o processo atender os requisitos do método. Os mínimos quadrados têm diversas propriedades, porém a mais interessante que será mais debatida ao longo da dissertação é a de ortogonalidade, onde o valor de saída estimada pelos parâmetros, $\hat{Y} = \Phi\hat{\theta}$, não tem projeção no vetor de erro. Ou seja, o vetor de saída estimada não tem correlação com o vetor erro, isso pode ser demonstrado na equação abaixo.

$$\hat{Y}^T \xi = \hat{\theta}_{MQ}^T \Phi^T (Y - \Phi\hat{\theta}_{MQ}) \quad (2.15)$$

$$\hat{Y}^T \xi = \left\{ [\Phi^T X]^{-1} \Phi^T Y \right\}^T \Phi^T \left(Y - \left[\Phi [\Phi^T X]^{-1} \Phi^T Y \right] \right) \quad (2.16)$$

$$\hat{Y}^T \xi = Y^T \Phi [\Phi^T \Phi]^{-1} \Phi^T \left(Y - \Phi [\Phi^T X]^{-1} \Phi^T Y \right) \quad (2.17)$$

$$\hat{Y}^T \xi = Y^T X [\Phi^T \Phi]^{-1} Y - Y \Phi [\Phi^T \Phi]^{-1} [\Phi^T X] [\Phi^T X]^{-1} Y \quad (2.18)$$

$$\hat{Y}^T \xi = 0 \quad (2.19)$$

Pois $[\Phi^T \Phi]^{-1} [\Phi^T \Phi] = 1$. Logo o resultado comprova que os vetores \hat{Y} e ξ são ortogonais.

2.4 O Estimador de Mínimos Quadrados Recursivo

Como visto na seção anterior, a técnica de estimação paramétrica mínimos quadrados é capaz de estimar parâmetros do sistema utilizando um conjunto de entradas e saídas. Porém, é necessário que esse conjunto seja armazenado e calculado em batelada. Tal configuração faz com que a estimação paramétrica seja mais demorada devido ao aumento das matrizes, e assim impossibilitando a estimação em tempo real. Já a versão recursiva do método utiliza o mesmo critério de minimização do erro de estimação, porém sem necessitar que o conjunto de saída e entradas seja armazenado em matrizes longas e estimando os parâmetros amostra por amostra. Para isso, é necessário encontrar o K_{MQR} da equação (2.20) que pondera o erro de estimação, convergindo os parâmetros estimados para os parâmetros reais do sistema. Importante destacar que o tempo discreto, k , pode aparecer tanto entre parênteses como subscrito nas variáveis temporais. Tal dualidade é necessária de modo a reduzir o tamanho de algumas equações.

$$\hat{\theta}_k = \hat{\theta}_{k-1} + K_{MQR} (y_k - \phi_k^T \hat{\theta}_{k-1}) \quad (2.20)$$

Para possibilitar a estimação dos parâmetros em tempo real, é necessário reescrever a estimação por mínimos quadrados em batelada para amostra por amostra, para isso são feitas as seguintes relações:

$$\hat{\theta}_{MQ}(k) = [\Phi^T \Phi]^{-1} \Phi^T Y = \left[\sum_{i=0}^k \phi(i) \phi^T(i) \right]^{-1} \left[\sum_{i=0}^k \phi(i) y(i) \right] \quad (2.21)$$

Outra adaptação necessária é a matriz de covariância paramétrica, P_k , calculada no instante k que é atualizada amostra por amostra:

$$P_k = \left[\sum_{i=0}^k \phi(i) \phi^T(i) \right]^{-1} \quad (2.22)$$

$$P_k^{-1} = \left[\sum_{i=0}^{k-1} \phi(i) \phi^T(i) \right] + \phi_k \phi_k^T \quad (2.23)$$

$$P_k^{-1} = P_{k-1}^{-1} + \phi_k \phi_k^T \quad (2.24)$$

Calculando o $\hat{\theta}$ no instante $k - 1$ temos:

$$\left[\sum_{i=0}^{k-1} \phi(i) \phi^T(i) \right] \hat{\theta}_{MQ}(k-1) = \left[\sum_{i=0}^{k-1} \phi(i) y(i) \right] \quad (2.25)$$

Substituindo a parte esquerda pela forma compacta de equação 2.22 resulta na simplificação $P_{k-1}^{-1} \hat{\theta}_{MQ_{k-1}}$. Essa modificação permite a alteração da equação, resultando nas equações:

$$\hat{\theta}_{MQ} = \hat{\theta}_{MQR_k} = P_k \left[P_{k-1}^{-1} \hat{\theta}_{MQR_{k-1}} + \phi_k y_k \right] \quad (2.26)$$

$$\hat{\theta}_{MQR_k} = \hat{\theta}_{MQR_{k-1}} + P_k \phi_k \left[y_k - \phi_k^T \hat{\theta}_{MQR_{k-1}} \right] \quad (2.27)$$

$$\hat{\theta}_{MQR}(k) = \hat{\theta}_{MQR}(k-1) + K_{MQR}(k) \left[y(k) - \phi(k)^T \hat{\theta}_{MQR}(k-1) \right] \quad (2.28)$$

O destaque dado por $\hat{\theta}_{MQ} = \hat{\theta}_{MQR}(k)$, pretende ressaltar que apesar do método de calcular os parâmetros tenha se modificado, o MQR mantém as mesmas características do MQ sendo uma delas a ortogonalidade. Nota-se que o valor de K_{MQR} no método de MQR é determinado pelo valor de P_k , contudo, é inadequado utilizar o método presente na equação, pois requer uma inversão de matriz a cada laço do algoritmo em que o MQR for aplicado. Visto isso, é mais vantajoso o uso do Lemma de inversão matricial apresentado em Aguirre (2015), gerando a equação abaixo:

$$P_k = P_{k-1} - P_{k-1} \phi_k \left(\phi_k^T P_{k-1} \phi_k + 1 \right)^{-1} \phi_k^T \hat{\theta}_{k-1} \quad (2.29)$$

Essa forma possibilita reescrever K_k em relação a P_{k-1} e calcular o P_k para as futuras iterações do algoritmo, resultando nos seguintes conjunto de equações:

$$K_{MQR_k} = \frac{P_{k-1} \phi_k}{\phi_k^T P_{k-1} \phi_k + 1} \quad (2.30)$$

$$\hat{\theta}_{MQR_k} = \hat{\theta}_{MQR_{k-1}} + K_{MQR_k} \left[y_k - \phi_k^T \hat{\theta}_{MQR_{k-1}} \right] \quad (2.31)$$

$$P_k = P_{k-1} - K_k \phi_k^T P_{k-1} \quad (2.32)$$

Com base nestas equações, o algoritmo MQR possibilita a estimação dos valores paramétricos amostra por amostra, atualizando o ganho do estimador a cada inovação, sendo assim uma versão *online* do MQ, conservando as suas propriedades. Importante notar que o algoritmo necessita de um valor inicial para a matriz covariância, P_0 , dessa forma nos processos em que os reais parâmetros do sistema são completamente desconhecidos, assume-se que os parâmetros estimados iniciais são iguais a zero e é aconselhável que uma matriz inicial assuma o valor $P_0 > 10^7$ (COELHO; COELHO, 2015). Essa matriz de covariância inicial de grande magnitude será minimizada durante o laço de repetição do algoritmo, aumentando o nível de conhecimento dos parâmetros reais do sistema.

2.5 O Estimador Recursivo Estendido de Mínimos Quadrados

Nota-se que os algoritmos estimadores citados identificam os parâmetros relacionando as saídas com entradas do sistema, assumindo que qualquer perturbação não tem regressão no processo. Porém, em sistemas do modelo ARMAX a perturbação tem regressores, logo, um ruído colorido. Se tal dinâmica não for modelada pela técnica de identificação, a fidelidade dos parâmetros estimados pelo algoritmo é comprometida. Visto isso, é adicionado a parcela do erro de estimação no vetor de regressores do MQR para modelar essa dinâmica, tornando-o Mínimos Quadrados Recursivo Estendido (MQRE). Essa alteração no vetor de regressões não afeta o funcionamento tradicional do MQR, sendo assim o método MQRE é estruturado pelas seguintes equações:

$$y(k) = [y(k-1) \cdots y(k-n_a) \ u(k-d) \cdots u(k-n_u) \ \hat{e}(k-1) \cdots \hat{e}(k-n_c)] \quad (2.33)$$

$$K_k = \frac{P_{k-1}\phi(k)}{\phi^T(k)P_{k-1}\phi(k) + 1} \quad (2.34)$$

$$\hat{\theta}_{MQRE}(k) = \hat{\theta}_{MQRE}(k-1) + K_k [y(k) - \phi(k)^T \hat{\theta}_{MQRE}(k-1)] \quad (2.35)$$

$$P_k = P_{k-1} - K_k \phi^T(k) P_{k-1} \quad (2.36)$$

$$\hat{e}(k) = y(k) - \phi^T(k) \hat{\theta}(k) \quad (2.37)$$

Como supracitado, o objetivo dos mínimos quadrados é a diminuição do erro de estimação, logo, ao agregar o erro estimado ao vetor de regressões, pode-se estimar valores que modelam a perturbação, atenuando conseqüentemente a polarização. Assim, o comportamento do polinômio formado pelo erro de estimação se assemelha com as perturbações reais do sistema, sendo assim:

$$\hat{C}(q)e(k) \approx C(q)w(k) \quad (2.38)$$

2.6 Variáveis Instrumentais

Os algoritmos da família MQ têm como princípio a minimização de uma função definida como o somatório do quadrado do erro entre a medição e a estimação com base em um conjunto de dados. Como visto, a álgebra linear é a ferramenta base do estimador MQ, tendo boas propriedades de robustez a erros de regressão. Tal técnica "acredita" que todo o erro entre a medição do sensor e a saída estimada pelos regressores modelo é resultado de perturbações que afetam a planta, porém todos os sensores, em maior ou menor grau, têm uma parcela de ruído em sua medição e tal parcela não descreve a grandeza do sistema que o sensor está medindo. Logo, o sinal ruidoso proveniente do sensor afeta o processo de estimação de algoritmos da família MQ tornando os parâmetros estimados tendenciosos, e o seu grau de polarização está estritamente ligado com a Relação Sinal Ruído (SNR) do sistema (AGUIRRE, 2015).

Visto isto, a polarização dos parâmetros afeta diretamente as estimações de saídas, assim como os projetos de controle que usam tais parâmetros para definir as ordens de controle. Portanto, foram desenvolvidos estimadores não polarizados que visam aumentar a confiabilidade dos parâmetros. Um dos mais usuais é o método de Variáveis Instrumentais (VI), este aplica variáveis complementares que auxiliam na estimação paramétrica, permitindo que, por um curto período, tal polarização exista, mas com o aumento de número de observações, a polarização diminua a tal ponto que seja irrelevante. Abaixo segue a equação geral de métodos que utilizam as saídas medidas do sistema para estimar os seus parâmetros:

$$\hat{\theta} = AY = BCY \quad (2.39)$$

O método de estimação por VI decompõe a matriz A , representado na equação (2.39), em duas que vão auxiliar no cálculo dos parâmetros, $A = BC$. Essas duas matrizes são função da mesma variável instrumental $B = f(z)$ e $C = g(z)$, sendo z um vetor de variáveis instrumentais. Escolhendo z de forma que dependa dos dados do sistema, a matriz B e C podem ser definidas como na equação (2.40), onde Z é a matriz formada pelas variáveis instrumentais.

$$\hat{\theta}_{VI} = [Z^T \Phi]^{-1} Z^T Y \quad (2.40)$$

$$B = [Z^T \Phi]^{-1}; C = Z^T \quad (2.41)$$

Note que para o caso MQ as matrizes B e C são respectivamente $B = [\Phi^T \Phi]^{-1}$ e $C = \Phi^T$, logo pode-se afirmar que o método de variáveis instrumentais não é ortogonal, pois, como definido na equação (2.19), apenas a configuração do MQ garante ortogonalidade com o vetor de erros de estimação. Porém, utilizar apenas os regressores como base para calcular o $\hat{\theta}_{MQ}$ resultaria em parâmetros tendenciosos, então o estimador VI evita a polarização fazendo uso de variáveis instrumentais correlacionadas com os dados do sistema afetado por ruídos de medição. Essas variáveis instrumentais utilizadas no processo de identificação são selecionadas pelo projetista.

A matriz Z é uma variável instrumental que pode ser formada pelos mais diversos vetores regressores estimados do sistema (z), como os representados nas equações (2.42) e (2.43), onde \hat{y} é a saída estimada por outro método de identificação, muitas vezes o próprio MQ estimando a saída com dados polarizados, e a segunda equação é um vetor de variáveis instrumentais com apenas as entradas do sistema. A escolha afeta no grau de polarização dos parâmetros estimados pela técnica de VI tornando a convergência paramétrica mais rápida ou mais lenta. A escolha de qual vetor de regressores estrutura matriz Z afeta o grau de polarização dos parâmetros estimados pela técnica de VI tornando a convergência paramétrica mais rápida ou mais lenta.

$$z = [\hat{y}(k-1) \quad \cdots \quad \hat{y}(k-n_y) \quad u(k-d) \quad \cdots \quad u(k-n_{nu}-d)] \quad (2.42)$$

$$z = [u(k-d) \quad \cdots \quad u(k-n_{nu}-n_y-d)] \quad (2.43)$$

Assim como o estimador MQ, o estimador VI tem a sua versão recursiva que mantém as propriedades do estimador VI em batelada, sendo necessário adequar a matriz de variância inicial, M_0 , conforme o grau de conhecimento das dinâmicas da planta a ser identificada. Abaixo seguem as equações do estimador VI recusivo:

$$M_k = M_{k-1} - M_{k-1} z_k [1 + \phi_k^T M_{k-1} z_k]^{-1} \phi_k^T M_{k-1} \quad (2.44)$$

$$K_k = M_k z_k \quad (2.45)$$

$$\hat{\theta}_{VI}(k) = \hat{\theta}_{VI}(k-1) + K_k (y(k) - \phi_k^T \hat{\theta}_{VI}(k-1)) \quad (2.46)$$

A versão recursiva de VI foi utilizada pelo proposto método, sendo responsável por atenuar os efeitos da polarização dos parâmetros estimados com base em leitura dos sensores dos quadrimotores afetados por ruídos de medição, dessa forma, oferecendo maior confiabilidade nos dados (parâmetros e estados) resultantes do algoritmo projetado.

2.7 Métodos de Estimação de Estados

O Filtro de Kalman Estendido é um método bem discutido na literatura e tem como característica estimar tanto os parâmetros quanto os estados do sistema, tendo como base as equações do Filtro de Kalman. Logo, é necessário primeiro conhecer os princípios do Filtro Kalman para poder compreender o funcionamento do Filtro de Kalman Estendido.

2.7.1 Filtro de Kalman

O filtro dessa seção tem como base a minimização da variância do erro de estimação, tópico já debatido em todos os métodos de identificação citados nessa dissertação, porém ao invés de estimar parâmetros, o Filtro de Kalman (FK) minimiza o erro de estimação de estado de sistemas lineares, já que o filtro utiliza os parâmetros do modelo para amenizar a influência dos ruídos de dados provenientes de sensores, dessa maneira estimando os estados do sistema.

Nesta dissertação, o FK utiliza o seguinte modelo discreto em espaço de estados:

$$\begin{aligned}x(k) &= A_{k-1}x(k-1) + B_{k-1}u(k-d) + G_{k-1}w(k-1) \\y(k) &= Cx(k) + v_k\end{aligned}\tag{2.47}$$

onde x é o vetor de estado do sistema, u das entradas, A é a matriz Jacobiana $n \times n$ em relação ao estado sendo n o número de estados do sistema, B é a matriz Jacobiana $n \times n_u$ em relação às entradas sendo n_u o número de entradas do sistema, G é a matriz Jacobiana $n \times n$ em relação às perturbações, C é matriz Jacobiana $n_y \times n$ sendo n_y o número de saídas do sistema que relaciona os estados com a saída da planta, $w(k-1)$ é a variável aleatória independente que representa a perturbação do sistema e $v(k)$ é a variável aleatória independente que representa o ruído do sensor. Nota-se que o sistema tem parte regressiva da saída, entradas exógenas e perturbações Gaussianas, sendo assim um modelo ARMAX.

Em posse desse modelo, o FK tem o objetivo de estimar os estados do sistema de maneira ótima utilizando os dados de entrada, u , e os dados ruidosos dos sensores, y . Esse método é aplicado em diversas áreas, como na filtragem de sinais de exames eletrocardiográficos (HESAR; MOHEBBI, 2020), sistemas aeroespaciais (AFONIN et al., 2020), até mesmo em previsões de demanda turística (ALAMINOS; SALAS, 2022).

O FK tem como base o trabalho de Kalman (1960), desenvolvido na área de filtragem digital e um ano depois expandindo para sistemas contínuos. Este método no domínio digital é dividido em duas etapas, a etapa de propagação e a etapa de assimilação. A etapa de propagação se refere à transição de um estado para o outro sem a informação

dos dados provenientes dos sensores. Para destacar isso, o índice “−” na parte superior da variável refere-se à etapa de propagação e o índice “+” à etapa de assimilação. Como dito, o problema a ser resolvido é a minimização da variância do estado estimado para o estado real, para isso temos as seguintes equações:

$$\tilde{x}_k^- = \hat{x}_k^- - x_k \quad (2.48)$$

$$\hat{x}_k^- = A_{k-1}\hat{x}_{k-1}^+ + B_{k-1}u(k-d) \quad (2.49)$$

onde \tilde{x} é diferença entre o estado estimado, \hat{x} , presente na equação (2.49) e o estado do sistema, x . Calculando a matriz covariância do estado estimado, P_{kf_k} , temos seguinte equação:

$$\text{cov}[\hat{x}_k^-] = E \left[\left(\hat{x}_k^- - E[\hat{x}_k^-] \right) \left(\hat{x}_k^- - E[\hat{x}_k^-] \right)^T \right] = P_{kf_k}^- \quad (2.50)$$

onde E é o operador de esperança matemática. Desenvolvendo a equação (2.50), chega-se à solução representada pela equação (2.51), onde Q é a matriz de covariância do ruído de perturbação, $E[ww^T]$.

$$P_{kf_k}^- = A_{k-1}P_{kf_{k-1}}^+ A_{k-1}^T + G_{k-1}Q_{k-1}G_{k-1}^T \quad (2.51)$$

Com esse resultado, chega-se à solução da etapa de propagação do filtro de Kalman, que recebe esse nome devido à propagação de um estado para o outro. Nota-se que nessa etapa a variação do estado estimado não depende da leitura dos sensores, portanto é necessário encontrar uma relação que pondere as saídas ruidosas aos estados estimados. Essa etapa é conhecida como assimilação, pois utiliza os dados ruidosos do sistema a fim de estimar estados otimizados, minimizando a matriz P_{kf} amostra por amostra. A equação a seguir representa a relação proposta pelo método:

$$\hat{x}_k^+ = \hat{x}_k^- + L_k \left[y_k - C_k \hat{x}_k^- \right] \quad (2.52)$$

Logo, o objetivo é encontrar o conjunto de valores que faz com que o ganho L seja ótimo. Para isso, utiliza-se a variância do ruído de medição, $E[vv^T]$, representada por R . Portanto, o ganho L pondera a relação entre a saída estimada da etapa de propagação e a saída ruidosa. Para definir o ganho, é determinada a matriz de covariância na etapa de assimilação de forma parecida com a etapa de propagação. O resultado está presente na equação (2.53).

$$P_{kfk}^+ = (I - L_k C_k) P_{kfk}^- (I - L_k C_k)^T + L_k R L_k^T \quad (2.53)$$

Em posse dessa equação, pode-se encontrar o valor do ganho L que minimiza a matriz P_{kfk}^+ , dessa forma encontrando o ganho ótimo do sistema que atualiza os estados estimados pelo método KF. Para isso, o traço da matriz ($Tr(\cdot)$) P_{kfk}^+ é derivado em relação à L e a equação resultante é igualada a zero, assim encontrando os conjuntos de ganhos ótimos do filtro.

$$\frac{\partial Tr(P_{kfk}^+)}{\partial L} = 0 = -2(I - LC) P_{kfk}^- C^T + 2LR \quad (2.54)$$

Isolando L , temos a seguinte equação:

$$L = P_{kfk}^- C^T [C P_{kfk}^- C^T + R]^{-1} \quad (2.55)$$

Substituindo o L na equação e simplificando as equações, têm-se os seguintes conjuntos de equação que fundamentam o algoritmo do Filtro de Kalman.

$$FK \begin{cases} \hat{x}_k^+ = A_{k-1} \hat{x}_{k-1}^+ + B_{k-1} u_{k-1} + A_{k-1} \left([y_k - C \hat{x}_{k-1}^+] \right) \\ L_k = P_k^- C^T [C P_k^- C^T + R]^{-1} \\ P_k^+ = A_{k-1} P_{k-1}^+ A_{k-1}^T - A_{k-1} L_k C P_{k-1}^+ A_{k-1}^T + G_{k-1} Q G_{k-1}^T \end{cases} \quad (2.56)$$

Dessa forma, retirando as variáveis da etapa de propagação e evidenciando o algoritmo em função dos dados da etapa de assimilação anterior, \hat{x}_{k-1}^+ .

Como visto em outros métodos de minimização da matriz covariância, o valor inicial da matriz deve ser alto ou conhecido *a priori* e a atualização do ganho L proporciona a redução da magnitude da matriz até encontrar o conjunto de ganhos que otimiza a relação dos estados estimados e os estados reais. O desenvolvimento das fórmulas do FK é retratado para evidenciar que o algoritmo é um estimador recursivo ótimo, pois, se os estados do sistema forem os parâmetros do modelo como ilustrado na equação (2.57) onde a variável aleatória, $v(k)$, tem variância igual a 1, o desenvolvimento matemático desse sistema recairá nas mesmas equações do MQR.

$$\begin{aligned} \theta_k &= \theta_{k-1} \\ y_k &= \phi_k^T \theta_k + v_k \end{aligned} \quad (2.57)$$

Tal semelhança permite expandir a estimativa MQR para sistemas MIMO utilizando as matrizes de regressões apresentada na equação (2.58), onde Φ_k^T , é a matriz de regressores

formada por cada vetor de regressão de cada estado do sistema, $\phi_{x_n}^T(k)$, como representado a seguir:

$$\Phi_k^T = \begin{bmatrix} \phi_{x_1}^T(k) & 0 & 0 & 0 \\ 0 & \phi_{x_2}^T(k) & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & \phi_{x_n}^T(k) \end{bmatrix} \quad (2.58)$$

Após isso, estimam-se os parâmetros do sistema MIMO aplicando o filtro de Kalman no sistema em espaço de estado, dado por:

$$\begin{aligned} \theta_k &= \theta_{k-1} \\ x_k &= \Phi_k^T \theta_k + v_k \end{aligned} \quad (2.59)$$

Porém, tal estrutura inviabiliza o KF estimar os estados do sistema, pois em sua forma canônica não permite a estimação conjunta (parâmetros e estados). Visto isso, foi desenvolvido o algoritmo de Filtro de Kalman Estendido capaz de evitar o problema da não linearidade do sistema e estimar parâmetros e estados.

2.7.2 Filtro de Kalman Estendido

Como visto na seção anterior, o cálculo do FK não permite a estimação dos estados e dos parâmetros que representam a dinâmica do processo em que ele é aplicado. Tal configuração resultaria em uma estimação de estados com base em parâmetros que ainda não foram calculados, logo, utilizar uma técnica de identificação conjunta em sistemas resultaria em um algoritmo que usa como base um sistema não linear representado abaixo:

$$\begin{aligned} x_k &= f(x_{k-1}, u_{k-d}, w_{k-1}) \\ y_k &= h(x_k) + v_k \end{aligned} \quad (2.60)$$

onde os estados do modelo, x_k , estão em função da correlação dos parâmetros, estados, entradas e perturbações da planta no instante $k - 1$, e a saída está em função de $h(\cdot)$. Para estimar de maneira ótimas os estados e os parâmetros, Jazwinski (1970) desenvolveu uma solução que tem como base a linearização analítica em torno do estado atual implementando os primeiros termos da expansão de Taylor, desenvolvendo o filtro de Kalman a partir desse ponto, essa linearização é feita amostra por amostra derivando a função $f(\cdot)$ e $h(\cdot)$. Em resumo, o FKE é o próprio FK implementado utilizando as Jacobianas de $f(\cdot)$ e $h(\cdot)$ em torno dos estados atuais do sistema (AGUIRRE, 2015).

Sabendo que $f(\cdot)$ e $h(\cdot)$ são funções vetoriais, pois $x(k)$ e $y(k)$ são vetores, as Jacobianas são definidas da seguinte forma:

$$Df(x) = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \cdots & \frac{\partial f_n}{\partial x_n} \end{bmatrix}, Df(w) = \begin{bmatrix} \frac{\partial f_1}{\partial w_1} & \cdots & \frac{\partial f_1}{\partial w_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial w_1} & \cdots & \frac{\partial f_n}{\partial w_n} \end{bmatrix}, Dh(x) = \begin{bmatrix} \frac{\partial h_1}{\partial x_1} & \cdots & \frac{\partial h_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial h_{ny}}{\partial x_1} & \cdots & \frac{\partial h_{ny}}{\partial x_n} \end{bmatrix}. \quad (2.61)$$

Em posse das matrizes Jacobianas, é formulado o conjunto de equações que estimam tanto os parâmetros como os estados amostra por amostra.

$$FKE \begin{cases} \hat{x}_k^- = f(\hat{x}_{k-1}^+, u_{k-d}) \\ P_k^- = Df(\hat{x}_{k-1}^+)P_{k-1}^+ Df(\hat{x}_{k-1}^+)^T + Df(w)QDf(w)^T \\ L_k = P_k^- Dh(x)^T [Dh(x)P_k^- Dh(x)^T + R] \\ \hat{x}_k^+ = \hat{x}_k^- + L_k [y_k - C\hat{x}_k^-] \\ P_k^+ = P_k^- - L_k Dh(x)P_k^- \end{cases} \quad (2.62)$$

Nota-se que tanto as funções $f(\cdot)$ e $h(\cdot)$ não são lineares, porém as Jacobinas de tais funções assumem o lugar de A e C do FK viabilizando o algoritmo. Porém essa adaptação tem o custo de dificultar a determinação os valores da matriz de covariância dos estados, pois a variância do parâmetro e dos estado se correlacionam no mesmo instante, além de que é necessário conhecer a matriz R e a matriz Q antes de formular o algoritmo. Visto isso, nesta dissertação é descrita outra abordagem que utiliza as definições de variáveis instrumentais combinadas à estimação ótima de estados proposta pelo FK.

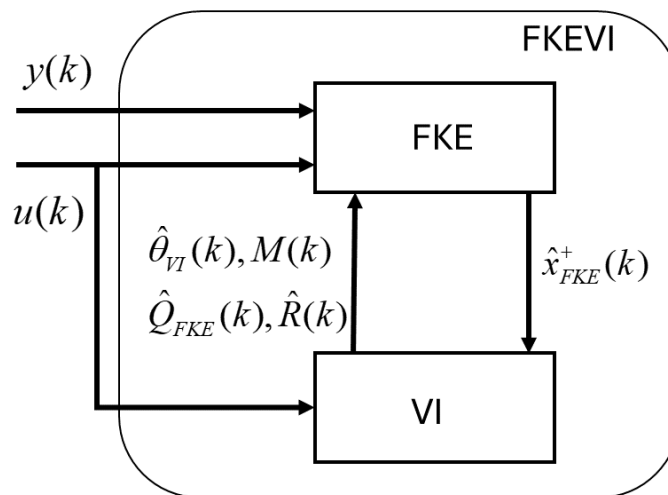
2.8 Filtro de Kalman Estendido Instrumental

Nessa dissertação, propõe-se outra abordagem na área de identificação de sistemas, que utilize as técnicas já apresentadas e as sintetize de forma que cada técnica possa contribuir com os seus atributos. Como já discutido, tanto o FK como o MQR são estimadores ótimos, logo, conclui-se que a versão estendida do filtro (FKE) também seja ótima no ponto onde foi linearizado, tendo como característica a mínima variância do erro de estimação de estado. Porém, em um projeto onde se aplica um filtro, entende-se que os dados estão corrompidos por ruídos, logo, utilizar o FKE calculando os parâmetros por métodos de minimização da variância tende a resultar em parâmetros polarizados, como discutido na seção 2.6. Diante disso, este trabalho utiliza o método de variáveis instrumentais para atenuar a polarização dos parâmetros estimados pelo FKE usando como base os dados previamente filtrados pelo estimador conjunto. Em contrapartida, o

estimador VI envia parâmetros menos polarizados, estabelecendo uma relação mutualista a qual é dado o nome de Filtro de Kalman Estendido com Variável Instrumental (FKEVI).

Na Figura 2, segue o diagrama que apresenta relações matemáticas que serão explicadas posteriormente.

Figura 2. Diagrama FKEVI



Fonte: elaborado pelo autor do texto

Note que a técnica mostrada no diagrama ocorre amostra por amostra, logo os algoritmos que compõem o método proposto estão na versão sua recursiva. O FKE nesse trabalho é projetado para o sistema MIMO, logo é necessário adaptar o estimador VI para as características do sistema, já que na maioria dos materiais didáticos o método VI é projetado para o sistema de uma entrada e uma saída. Com base na sessão anterior, pode-se criar a matriz de regressores estimados, como mostra a equação (2.63).

$$\Pi_k = \begin{bmatrix} z_{x_1}^T(k) & 0 & 0 & 0 \\ 0 & z_{x_2}^T(k) & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & z_{x_n}^T(k) \end{bmatrix} \quad (2.63)$$

onde $z_{x_1}^T(k)$ é o vetor de regressores estimados de cada estado pelo FKE, dessa forma, oferecendo dados previamente filtrados. Essa configuração permite que a matriz Π_k seja a variável instrumental do método VI recursivo. Logo, o algoritmo do estimador VI aplicado neste trabalho tem a seguinte configuração:

$$\Phi_k^T = \begin{bmatrix} \phi_{x_1}^T(k) & 0 & 0 & 0 \\ 0 & \phi_{x_2}^T(k) & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & \phi_{x_n}^T(k) \end{bmatrix} \quad (2.64)$$

$$M_k = M_{k-1} - M_{k-1}\Pi_k \left[I_{n \times n} + \Phi_k^T M_{k-1} \Pi_k \right]^{-1} \Phi_k^T M_{k-1} \quad (2.65)$$

$$K_k = M_k \Pi_k \quad (2.66)$$

$$\hat{\theta}_{VI}(k) = \hat{\theta}_{VI}(k-1) + K_k \left[y(k) - \Phi_k^T \hat{\theta}_{VI}(k-1) \right] \quad (2.67)$$

Feitas as adaptações, o primeiro passo para aplicação do FKEVI se refere à estrutura do filtro que deve ser adaptada conforme o grau de confiabilidade daquele parâmetro. No FKE tradicional, a matriz de covariância do próprio filtro estabelece esse grau e, ao longo do experimento, ele segue minimizando a matriz de covariância dos estados (estados e parâmetros) a cada amostra. Tal conversão na presença de ruído estabelece parâmetros polarizados, resultando em estimativas de estado tendenciosas que irão suprimir as reais dinâmicas do processo, comprometendo os dados enviados para o estimador VI. A alternativa encontrada foi inserir no algoritmo FKE os graus de incertezas paramétricas do próprio estimador VI, dessa forma fazendo com que ambos os tempos de convergência dos algoritmos sejam os mesmos.

Logo, é adicionada a diagonal da matriz covariância do estimador VI, M_k , na parte da diagonal da matriz Q do algoritmo FKE que se referem aos estados paramétricos do processo. Dessa maneira, ambos os algoritmos convergem em conjunto. Essa operação está representada na equação (2.68).

$$Q_{FKE}^+(k-1) = \text{diag} \left(\left[Q_{FK}^+(k-1), M_{VI}(k-1) \right] \right) \quad (2.68)$$

onde Q_{FKE} é a matriz do Filtro Kalman estendido e o Q_{FK} é a matriz do filtro de Kalman apenas com matriz de covariância dos ruídos de perturbação.

As matrizes Q e a R do FKE podem ser estabelecidas conforme o grau de conhecimento do projetista sobre a planta, porém nesse trabalho tais informações não estão disponíveis para o FKEVI, sendo necessárias serem levantadas pelo processo de estimação conjunta proposta com o auxílio do estimador VI recursivo. Em posse dos estados estimados pelo FKE (estados do sistema e parâmetros), o estimador VI é aplicado, resultando

em parâmetros menos polarizados, auxiliando na estimação dos ruídos de medição e perturbações do sistema, como segue:

$$\hat{w}(k) = \hat{x}_{EKF}^+(k) - \Omega_k^T \hat{\theta}_{VI}(k) \quad (2.69)$$

$$\hat{v}(k) = y(k) - (\Pi_k^T \hat{\theta}_{VI}(k) + \hat{w}(k)) \quad (2.70)$$

$$Cov[\hat{w}] = \hat{Q}_{FK} \quad (2.71)$$

$$Cov[\hat{v}] = \hat{R}_{FK} \quad (2.72)$$

onde Ω_k^T é matriz de regressores estimados, pelo próprio estimador VI, como apresentado na equação (2.73).

$$\Omega_k^T = \begin{bmatrix} h_{x_1}^T(k) & 0 & 0 & 0 \\ 0 & h_{x_1}^T(k) & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & h_{x_n}^T(k) \end{bmatrix} \quad (2.73)$$

$$h_{x_1}^T(k) = [\hat{x}_{VI_1}(k-1) \cdots \hat{x}_{VI_n}(k-1) \ u_1(k-d) \ \cdots \ u_{nu}(k-d) \ \hat{w}_{x_1}(k-1)] \quad (2.74)$$

$$\hat{x}_{VI}(k) = \Pi_k^T \hat{\theta}_{VI}(k) \quad (2.75)$$

Dessa forma se estrutura o método de estimação dos parâmetros do sistema, mas vale ressaltar que tal método é focado em oferecer parâmetros estimados e as matrizes R e Q estimadas do processo, pois o estimador VI não é um observador de estados e FKE faz estimações com no grau de confiabilidade dos parâmetros expressas pela matriz de covariância do estimador VI. Logo, esses dois núcleos são responsáveis por estabelecer parâmetros determinísticos, como a matriz A , B e G , do sistema em espaço de estados e as dinâmicas estocásticas do processo, como a variância do ruído e a variância da perturbação.

Visto isso, é sintonizado um filtro de Kalman com base nesses dados estimados, assim oferecendo um observador ótimo que filtra o ruído dos sensores. Logo, a técnica proposta nesse trabalho é fundamentada em três núcleos principais.

O primeiro núcleo é o FKE, que estima os parâmetros e estados do sistema, porém, a estimação é tendenciosa devido à ausência de conhecimento das perturbações e ruídos do sistema. Logo, esse núcleo é caracterizado por estabelecer a filtragem/estimação bruta do sistema levando em consideração todas as incertezas do algoritmo principal, como o não conhecimento dos parâmetros e das variâncias dos ruídos e das perturbações que afetam o processo. Abaixo seguem as equações dessa etapa:

$$FKE \left\{ \begin{array}{l} \hat{x}_k^- = f(\hat{x}_{k-1}^+, u_{k-d}) \\ P_k^- = Df(\hat{x}_{k-1}^+)P_{k-1}^+Df(\hat{x}_{k-1}^+)^T + Df(w)\hat{Q}_{FKE}(k-1)Df(w)^T \\ L_k = P_k^- Dh(x)^T [Dh(x)P_k^- Dh(x)^T + \hat{R}_{FK}(k-1)] \\ \hat{x}_k^+ = \hat{x}_k^- + L_k [y_k - C\hat{x}_k^-] \\ P_k^+ = P_k^- - L_k Dh(x)P_k^- \end{array} \right. \quad (2.76)$$

Em posse desses dados provenientes da estimação bruta, o método de VI utiliza os estados estimados pelo FKE para estabelecer a matriz instrumental que auxilia na minimização da interferência dos ruídos nos parâmetros do sistema. Essa etapa estabiliza os resultados da filtragem bruta, oferecendo dados mais confiáveis para o terceiro e último núcleo do algoritmo proposto. Abaixo seguem as equações que compõem o segundo núcleo da técnica aplicada:

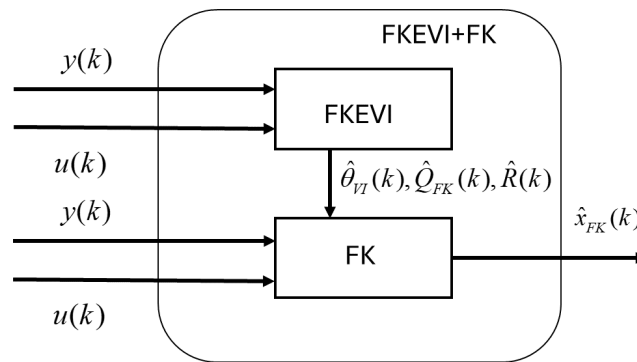
$$VI \left\{ \begin{array}{l} M_k = M_{k-1} - M_{k-1}\Pi_k [I_{n \times n} + \Phi_k^T M_{k-1}\Pi_k]^{-1} \Phi_k^T M_{k-1} \\ K_k = M_k \Pi_k \\ \hat{\theta}_{VI}(k) = \hat{\theta}_{VI}(k-1) + K_k (y(k) - \Phi_k^T \hat{\theta}_{VI}(k-1)) \\ \hat{w}(k) = \hat{x}_{EKF}^+(k) - \Omega_k^T \hat{\theta}_{VI}(k) \\ \hat{v}(k) = y(k) - (\Pi_k^T \hat{\theta}_{VI}(k) + \hat{w}(k)) \\ Cov[\hat{w}] = \hat{Q}_{FK}(k) \\ Cov[\hat{v}] = \hat{R}_{FK}(k) \end{array} \right. \quad (2.77)$$

Nota-se que nessa etapa, além dos parâmetros, é calculada variância dos ruídos e das perturbações que afetam o sistema MIMO. Dessa forma, permitindo a filtragem/estimação fina dos estados do sistema que usa como base resultados provenientes da etapa de estabilização promovida pelo método de VI. Essa estimação é chamada de fina, pois é calculada com dados atualizados provenientes da relação mutualística do FKEVI e não leva em conta as incertezas paramétricas que a etapa de filtragem bruta (FKE) é feita. Abaixo seguem as equações que compõem essa etapa do processo:

$$FK \left\{ \begin{array}{l} \hat{x}_{FK}^-(k) = A_{k-1}\hat{x}_{FK}^+(k-1) + B_{k-1}u_{k-1} \\ P_k^- = A_{k-1}P_{k-1}^+A_{k-1}^T + G_{k-1}\hat{Q}_{FK}(k)G_{k-1}^T \\ L_k = P_k^-C^T[CP_k^-C^T + \hat{R}_{FK}(k)]^{-1} \\ \hat{x}_{FK}^+(k) = \hat{x}_{FK}^-(k) + L_k[y_k - C\hat{x}_k^-] \\ P_k^+ = P_k^- - L_kCP_k^- \end{array} \right. \quad (2.78)$$

Devido à combinação dos algoritimos, o método proposto é chamado FKEVI+FK e o diagrama das suas conexões está exposto na Figura 3.

Figura 3. Diagrama FKEVI+FK



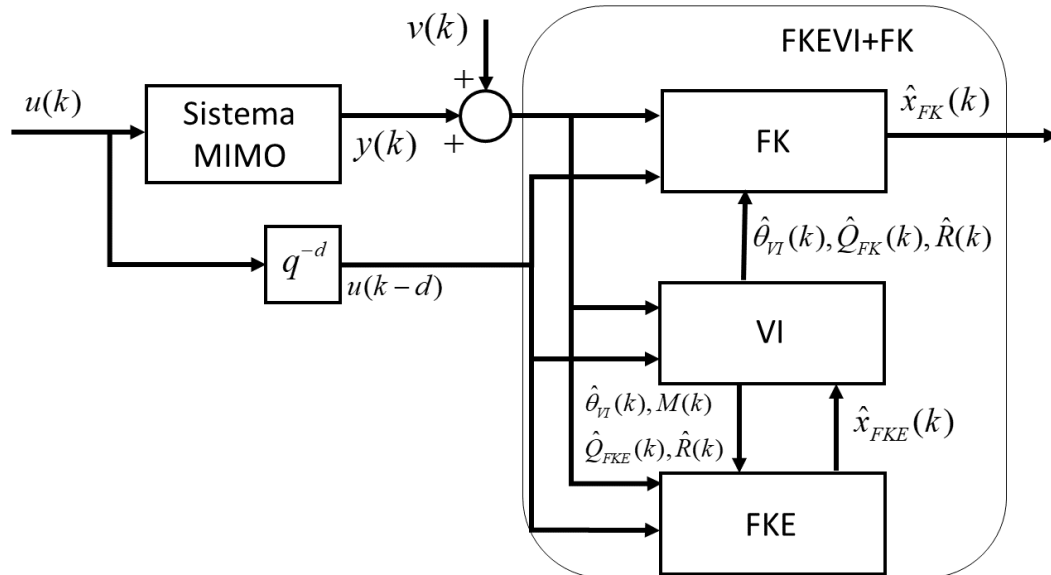
Fonte: elaborado pelo autor do texto

Dessa forma, o algoritimo FKEVI+FK filtra dados e estima parâmetros recursivamente. Vale destacar que tal algoritmo não é descentralizado (cada algoritmo trabalha independente do outro), tampouco centralizado (onde um algoritmo dita comportamento do outro em uma relação mestre e escravo) como é estabelecido em muitos trabalhos de estimação conjunta (XU et al., 2020). O termo mais adequado seria um algoritmo autogestionado devido à globalização dos dados, pois todos os núcleos algorítmicos (FKE, VI e FK) recebem os dados brutos (sem filtragem), e a cooperação pelo compartilhamento de dados do processo entre eles. Assim, ambos convergem à solução do problema.

Sendo assim, o método proposto pode ser sintonizado ao sistema MIMO sendo capaz de estimar conjuntamente parâmetros, estados e variância do ruído e variância das perturbações que afetam a planta, dessa forma, calculando tanto a parte determinística quanto a parte estocástica do sistema em espaço de estados. Na Figura 4 mostra o diagrama em blocos que representa as interações entre os sistemas e os algoritmos que compõem a técnica projetada nessa seção, onde no bloco FKE o algoritmo executa os cálculos presentes

na equação (2.76), no bloco VI são feitos os cálculos presentes na equação (2.77) e no bloco FK é aplicado o filtro Kalman presente na equação (2.78).

Figura 4. Diagrama FKEVI+FK com Sistema MIMO



Fonte: elaborado pelo autor do texto

2.9 Regulador Linear Quadrático e Gaussiano

O LQR é um regulador ótimo que pode ser projetado tanto para sistemas MIMO como para SISO (OGATA, 1996). O objetivo do LQR é encontrar um conjunto de ganhos que permita uma trajetória otimizada dos estados do sistema até o seu ponto de equilíbrio. Na equação (2.79) está o modelo em espaço de estados discreto, o qual o LQR está sujeito:

$$\begin{aligned} x(k) &= Ax(k-1) + Bu(k-d) \\ y(k) &= Cx(k) \end{aligned} \quad (2.79)$$

Com base no sistema linear, o algoritmo LQR minimiza a função custo presente na equação (2.80), onde J é a função de custo, Q_c é uma matriz definida (semi-) positiva e R_c é definida positiva. Essas matrizes determinam a sintonia do controlador ótimo.

$$J = \sum_{k=0}^{\infty} (x_k^T Q_c x_k + u_k^T R_c u_k) \quad (2.80)$$

Em posse da função custo busca-se o valor de K_c que otimiza a trajetória dos estados como descrito na equação:

$$u(k) = -K_c(x(k)) \quad (2.81)$$

Onde K_c é calculado da seguinte forma:

$$K_c = (R_c + B^T P_{c_k} B)^{-1} (B^T P_{c_k} A) \quad (2.82)$$

E P_{c_k} é encontrado iterativamente no tempo discreto pela equação dinâmica de Riccati:

$$P_{c_k} = A^T P_{c_{k-1}} A - A^T P_{c_{k-1}} B (R_c + B^T P_{c_{k-1}} B)^{-1} (B^T P_{c_{k-1}} A) + Q_c \quad (2.83)$$

Dessa forma, o algoritmo do LQR calcula o valor dos ganhos e das ordens de controle que otimizam a trajetória do estado para o ponto de equilíbrio do sistema. Esses valores se modificam a partir dos coeficientes das matrizes Q_c e R_c ; por exemplo, selecionar coeficientes grandes para a matriz R_c significa que o projetista aplica uma penalidade maior ao esforço de controle para otimizar a função de custo, o que também é conhecido como uma solução de controle de baixo custo, em contraste, selecionar coeficientes pequenos significa tentar estabilizar o sistema com uma estratégia de controle cara. Da mesma forma, se o projetista escolher coeficientes grandes para a matriz Q_c , isso significa tentar estabilizar o sistema com o mínimo de mudanças possíveis nos estados, já uma matriz Q_c com coeficientes pequenos implica menos preocupação com as mudanças nos estados.

O objetivo de controle neste trabalho na aplicação do LQR é o seguimento de referência da velocidade frontal, velocidade lateral. Para garantir o erro estacionário zero no seguimento de referência, o controlador deve ser projetado para o modelo estendido do sistema, controlando os estados e suas variações, como abordado no trabalho de Silveira et al. (2020), que está representado na equação (2.84), e ter a lei de controle de realimentação do estado baseada na equação (2.86) onde $r(k)$ é o vetor de referência do estado.

$$x_a(k) = \begin{bmatrix} y(k) \\ \Delta x(k) \end{bmatrix} = \overbrace{\begin{bmatrix} I & CA \\ \mathbf{0} & A \end{bmatrix}}^{A_a} \begin{bmatrix} y(k-1) \\ \Delta x(k-1) \end{bmatrix} + \overbrace{\begin{bmatrix} CB \\ B \end{bmatrix}}^{B_a} \Delta u(k-d) \quad (2.84)$$

$$y(k) = \overbrace{\begin{bmatrix} C_c & \mathbf{0} \end{bmatrix}}^{C_a} x_a(k) \quad (2.85)$$

$$\Delta u(k) = -K(x_a(k) - r(k)) \quad (2.86)$$

O operador de diferença discreto é representado por $\Delta = 1 - q^{-1}$, na qual o q^{-1} é o operador de deslocamento para trás do tempo discreto, $\Delta u(k)$ é o incremento de controle, C_c é matriz que correlaciona os estados com as saídas controladas pelo LQR e todas as variáveis como índice inferior a se trata de matrizes do espaço de estados com a sua versão aumentada. Logo, a ordem de controle está descrita na equação (2.87) é o sinal de controle a ser aplicado à entrada do sistema, que pode ser facilmente implementado em um computador digital para obter controle incremental.

$$u(k) = u(k - 1) + \Delta u(k) \quad (2.87)$$

Com base no sistema aumentado, tem-se as matrizes de ponderação aumentadas, Q_c e R_c , sendo necessário propor valores para os novos estados criados. O método de escolha de tais valores será abordado mais adiante da dissertação. Porém, o LQR para o cálculo da ordem de controle necessita dos valores dos estados do sistema, uma variável a que alguns casos não se têm acesso, sendo necessário estimar os estados. Ao utilizar o estimador Filtro de Kalman para estimar essas variáveis, o LQR se torna o LQG, onde o comando de controle é calculado usando ganhos ótimos com base em estimações ótimas como o descrito na equação (2.88).

$$\Delta u(k) = -K_c (\hat{x}_a^+(k) - r(k)) \quad (2.88)$$

Logo, o comando de controle é afetado não só pelos ganhos do sistema, mas também pelas estimativas feitas pelo FK. Logo, ao avaliar o projeto de controle, deve-se levar todas as dinâmicas do projeto de controle (Controlador, Planta e Observador) para ter maior previsibilidade da planta do sistema e robustez.

2.10 Índices de Análise de Desempenho e de Robustez

Os índices são ótimos métodos de avaliar o desempenho de algoritmos de identificação e controle. Para avaliar os algoritmos de estimação de estados, foi utilizado índice desvio raiz-média-quadrado normalizado (NRMSE) O NRMSE serve para agregar as magnitudes dos erros nas previsões para vários pontos de dados em uma única medida. O NRMSE é uma medida de precisão para comparar erros de previsão de diferentes modelos para um determinado conjunto de dados (HANNA; HEINOLD, 1985). Esse índice é definido pela equação mostrada a seguir:

$$NRMSE = \frac{\sum_k^N (\hat{x}_n^+(k) - x_n(k))^2}{\sum_k^N x_n^2(k)} \quad (2.89)$$

cada estado do sistema é comparado com a sua estimação. No caso desse trabalho, a biblioteca que permite a comunicação por WiFi com o drone fornece as informações dos sensores que leem estados que são usados nessa dissertação, dessa forma, possibilitando a construção do índice. Ao analisar a equação, quanto mais próximo de 1, mais o algoritmo de estimação se aproximou dos dados reais do sistema, possibilitando uma análise comparativa entre os métodos utilizados nesse trabalho.

Porém, tal método permite apenas avaliar o desempenho de estimação dos estados. Para comparar o desempenho da estimação paramétrica, utilizou-se o somatório das distâncias euclidianas dos parâmetros estimados para o parâmetro real do sistema, dessa forma avaliando o grau de polarização dos parâmetros. O método está representado na equação (2.90).

$$D_{\theta} = \sqrt{(\theta - \hat{\theta})^2} \quad (2.90)$$

Para avaliar o desempenho do sistema de controle neste trabalho, os critérios utilizados foram os índices integral do erro ao quadrado (ISE) e esforço de controle quadrático (ISU). Os índices de desempenho são calculados da seguinte maneira:

$$ISE = \frac{1}{N} \sum_{k=0}^N (y(k) - r(k))^2 \quad (2.91)$$

$$ISU = \frac{1}{N} \sum_{k=0}^N u(k)^2 \quad (2.92)$$

O ISE expressa o quanto se desviou a saída real para a saída desejada e o ISU o quanto foi gasto de energia para controlar aquele sistema, logo, quanto maiores tais índices, pior é o desempenho de controle (ZIANE et al., 2022). Esses índices são fundamentais na avaliação do sistema de controle e de identificação do algoritmo proposto, pois o controlador é sintonizado com base no modelo levantado pelo processo de identificação, sendo assim quanto mais os parâmetros estimados se aproximam do valor real melhor será o desempenho de controle.

Para avaliar robustez do algoritmo de controle adaptativo, são utilizadas a margem de ganho, GM_{db} e a margem de fase, PM° , do sistema em malha fechada. A margem de ganho estabelece quanto se pode elevar o ganho de malha até que o sistema se torne instável. E a margem de fase estabelece quanto atraso a malha de controle suporta até que se torne instável. Dessa forma, quanto maiores tais índices, mais robusto o sistema é

variação paramétrica e perturbações. Neste trabalho, para calcular tais índices, utiliza-se das funções S e T do sistema, onde S é a função de sensibilidade do sistema MIMO controlado em malha fechada e T é a função de sensibilidade complementar. Visto isso, a margem de ganho e a margem de fase são calculadas da seguinte maneira:

$$T + S = 1 \quad (2.93)$$

$$m_t = \underbrace{\max}_{\omega} |T(e^{-j\omega T_s})| \quad (2.94)$$

$$m_s = \underbrace{\max}_{\omega} |S(e^{-j\omega T_s})| \quad (2.95)$$

$$GM_{db} = 20 \log_{10} \left\{ \min \left[\left(\frac{m_s}{m_s - 1} \right), \left(1 + \frac{1}{m_t} \right) \right] \right\}, \quad (2.96)$$

$$PM^o = \left(\frac{180^\circ}{\pi} \right) \min \left[2 \operatorname{sen}^{-1} \left(\frac{1}{2m_s} \right), 2 \operatorname{sen}^{-1} \left(\frac{1}{2m_t} \right) \right] \quad (2.97)$$

onde e é a constante de Euler, ω é a frequência em radianos e j é a unidade imaginária e T_s é o tempo de amostragem do sistema discreto. A função $T(\cdot)$ engloba todo o processo no qual o sistema está inserido, logo, em casos de sistemas controlados e necessário avaliar o impacto do sinal de referência e do controlador no comportamento frequencial do processo, sabendo que os sistemas LQR e LQG são retroalimentados, para controlar o sistema expandido deve considerar a interferência do controlador no modelo em espaço de estado. A interferência está apresentada na equação (2.98).

$$x_a(k+1) = A_a x_a(k) + B_a \Delta u(k-d) = A_a x_a(k) - B_a K_c (x_a(k) - r(k)) \quad (2.98)$$

Transformando a equação em uma representação canônica no espaço de estado, obtém-se o seguinte sistema:

$$x_a(k+1) = \overbrace{(A_a - B_a K_c)}^{A_c} x_a(k) + \overbrace{B_a K_c}^{B_c} r(k) \quad (2.99)$$

onde A_c e B_c são a representação canônica de A e B do sistema MIMO em malha fechada, possibilitando a análise da dinâmica do sistema controlado. Ao adicionar observadores de estado, o sistema em malha fechada também será afetado pelo efeito de tal dinâmica, logo é necessário agregar o efeito do estimador na representação em espaço de estados, compreendendo o sistema em sua totalidade. Na equação (2.100), mostra-se o sistema estimador de estado.

$$\hat{x}_a(k+1) = A_{est}\hat{x}_a(k) + B_{est}\Delta u(k-d) + L(y(k) - \hat{y}(k)) \quad (2.100)$$

Nota-se que o sistema tem duas entradas, sendo uma as entradas do sistema e a outra a leitura dos sensores. Pela substituição da lei de controle na equação do estimador de estado obtém-se:

$$\hat{x}_a(k+1) = (A_{est} - LC - BK_c)\hat{x}_a(k) + B_{est}K_cr(k) + LCx(k) \quad (2.101)$$

onde A_{est} e B_{est} são os parâmetros estimados pelo método de identificação utilizado. Sabendo que o controlador utiliza as estimativas de estado, se tem o seguinte resultado:

$$x_a(k+1) = A_ax_a(k) - B_aK_c\hat{x}_a(k) + B_aK_cr(k) \quad (2.102)$$

Em posse das duas representações em espaço de estado da equação (2.96) e da equação (2.95), é possível representar na forma canônica assumindo o vetor de estado como um compensador dinâmico completo, como se mostra a seguir:

$$x_c(k+1) = \begin{bmatrix} x(k+1) \\ \hat{x}(k+1) \end{bmatrix} = \begin{bmatrix} A & -BK \\ LC_o & A_{est} - LC_o - B_{est} \end{bmatrix} \begin{bmatrix} x(k) \\ \hat{x}(k) \end{bmatrix} + \begin{bmatrix} B_{est}K \\ B_{est}K \end{bmatrix} r(k) \quad (2.103)$$

$$y(k) = \begin{bmatrix} C_c & 0 \end{bmatrix} \begin{bmatrix} x(k) \\ \hat{x}(k) \end{bmatrix} \quad (2.104)$$

A representação do compensador dinâmico demonstra o grau de importância da área de identificação de sistemas no projeto de controle, pois os métodos de identificação paramétrica e de estados devem se aproximar ao máximo dos reais parâmetros do sistema, para que o LQG projete ganhos ótimos com base nos parâmetros e o FK estime de maneira ótima para que o sistema em malha fecha tenha o melhor desempenho possível.

3 Materiais e Métodos

Nesse capítulo, serão debatidos os meios que auxiliaram a aplicação dos algoritmos de estimação e de controle. A primeira seção apresenta o local onde foram feitos os testes, logo após é descrito o quadricóptero e suas características. A terceira seção aborda a biblioteca em Python que permitiu o envio e recebimento de dados e, ao final, se estabelece o modelo linear a ser identificado.

3.1 Ambiente Físico

O ambiente de testes experimentais para a identificação do modelo do AR Drone 2.0 deve ser amplo o suficiente para tornar as colisões mais difíceis de acontecerem, com boa iluminação (pois as velocidades translacionais do drone são estimadas via *pixel flow* fazendo uso de uma câmera localizada na parte de baixo do drone, apontada para o solo) e isolado de interferências externas (*indoor*) como a do vento, por exemplo. Por isso, escolheu-se o Laboratório de Sistemas Elétricos de Potência como Geração Distribuída da UFPA como local adequado para a efetivação dos testes. No momento do voo experimental, o laboratório encontrava-se apenas com a estrutura predial, sem circulação de pessoas e objetos. A Figura 5 exibe a foto do local de realização dos voos experimentais.

Figura 5. Local de Testes



Fonte: elaborado pelo autor do texto

3.2 Quadricóptero

O algoritmo de identificação proposto foi aplicado no VANT do tipo quadricóptero modelo AR.Drone 2.0 da Parrot. Esse modelo é uma versão melhorada de seu antecessor (AR.Drone) e conta com câmera frontal e vertical, bateria de lítio de 3 células (11,1 CV), 1500 mAh, interface USB e WiFi™ 802.11n. Sua estrutura comporta quatro motores acoplados a hélices igualmente espaçadas do eixo central, sendo motores coreless de 14.5 W, sistema de processamento e transmissão de dados conta com processador ARM Cortex A8 32 bits de 1 GHz, sensores de navegação inercial – dentre eles, acelerômetro, girômetro, ultrassom (para pequenas altitudes) e barômetro (para maiores altitudes) (PARROT, 2021). A Figura 6 mostra o drone durante o seu voo.

Figura 6. AR.Drone 2.0 Voando



Fonte: elaborado pelo autor do texto

Tal configuração é bastante convidativa ao intuito da pesquisa, pois os sensores de navegação inercial propiciam dados que facilitam o processo de estimação de variáveis e, além disso, o aeromodelo está equipado com um dos sistemas *wireless* mais populares no mundo atual, o WiFi™ (MCFARLAND; WONG, 2003). Essa comunicação permite a troca de informações entre a unidade de comando e o drone em um espaço de tempo curto. A biblioteca desenvolvida em Python utilizada nesse trabalho se debruça nas configurações de processamento de dados e envio dos mesmos.

O AR Drone possui quatro rotores dispostos em formato de “x”, isto é, para que ele execute variações no ângulo de rolagem, sem que o ângulo de arfagem seja afetado, todos os quatro rotores devem ser acionados de forma que dois rotores aumentem as rotações e dois rotores diminuam as rotações. É importante deixar claro esse fator, pois

existem drones onde os rotores estão dispostos em formato de “+” o que pode dificultar o entendimento das dinâmicas.

Também é necessária a utilização de um computador munido de comunicação WiFi™ e, de preferência, com reduzidos atrasos de processamento para não inviabilizar os testes, devido à grande flutuação de atraso de transporte. O computador utilizado para fazer a comunicação e extração de dados para o trabalho foi o ideapad S145 da Lenovo, equipado com AMD Ryzen 5 8 GB de memória RAM, com sistema operacional Ubuntu 22.12.

3.3 PS-Drone

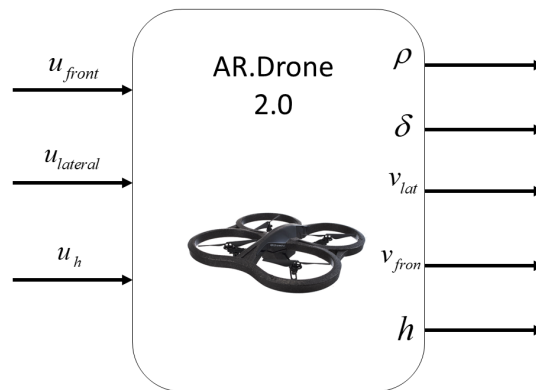
Criada por Graff (2014), a biblioteca PS-Drone é uma Interface de Programação de Aplicação completa, escrita em Python para AR.Drone 2.0 da Parrot. O pacote de informações dos sensores e dos comandos de voo tem tempo de envio médio de 1/15 segundos, podendo ser mais rápidos, porém na configuração padrão são enviados os dados filtrados pelo próprio *firmware* do quadricóptero. Logo, o tempo de envio e recepção dos dados é considerado o tempo de amostragem do sistema discreto do quadricóptero, T_s .

A partir dessa biblioteca, se tem acesso ao conjunto completo de possibilidades do AR.Drone 2.0, incluindo Sensor-Data (também conhecido como NavData) e os comandos de voo. Porém, a sua versão estava escrita para a versão antiga do Python 2, sendo necessário atualizá-la para a versão 3 do Python, possibilitando o voo experimental.

As variáveis que serão utilizados para a construção do modelo são os sensores (saídas) altitude (h) em metros, ângulo de arfagem (δ) em radianos, ângulo de rolagem (ρ) em radianos, velocidade frontal (v_f) e lateral (v_l) em metros por segundo. As entradas são e os comandos são: impulso frontal u_f , impulso lateral u_l e impulso de altitude u_h . É importante ressaltar que esses impulsos são adimensionais e limitados em $[-1.0; 1.0]$, e que a nomenclatura impulso é devido à biblioteca acessar apenas comandos de entrada de alto nível do quadricóptero. Os comandos de baixo nível, como a corrente elétrica que passa pelos motores, estão a cargo do próprio *firmware*, pois seria contraproducente enviar comandos de corrente elétrica aos motores com um atraso de 1/15s da comunicação. Logo, o comando impulso é entrada que executa manobras já padronizadas do *firmware*, permitindo ao usuário o controle apenas do movimento espacial do drone (dinâmica mais lenta) e o *firmware* é responsável por controlar a dinâmica mecatrônica do drone (dinâmica mais rápida). A Figura 6 mostra o bloco de comunicação.

A biblioteca se destaca por ter comandos que são muito importantes durante o experimento, como o de decolagem, pouso e pairagem que podem ser acionadas pelo prompt de comando, dessa forma possibilita que o pesquisador consiga interromper ou inicializar o experimento com praticidade e segurança.

Figura 7. Extração de Dados

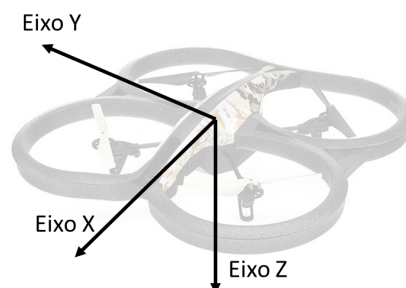


Fonte: elaborado pelo autor do texto

3.4 Modelo linear

As dinâmicas internas dos quadricópteros podem ser visualizadas na Figura 8, onde os seis graus de liberdade de movimentação ficam explícitos. De acordo com Stevens, Lewis e Johnson (2015), a rotação em torno do eixo X, do eixo Y e do eixo Z correspondem, respectivamente, aos ângulos de Rolagem, de Arfagem e de Guinada do Drone, tais ângulos são chamados de Ângulos de Euler. O giro desses ângulos no sentido horário resulta em valores de velocidades angulares positivas (regra da mão direita, onde o polegar segue o sentido dos eixos e os dedos restantes indicam o sentido de rotação). Consequentemente, para o giro em sentido oposto, as velocidades são negativas.

Figura 8. Orientação das dinâmicas do Parrot® AR Drone seguindo o sistema NED (North-East-Down)



Fonte: elaborado pelo autor do texto

Neste trabalho, são identificados os seis estados do sistema em espaço de estados,

sendo eles os ângulos entorno dos eixos X e Y (ângulo de rolagem e de arfagem), a taxa de deslocamento nesses eixos (velocidade frontal e lateral) e a altitude em relação ao solo, já os controladores adaptativos projetados nessa dissertação controlaram as velocidades lateral e frontal do quadricópteros. Portanto, as localizações no plano XY não serão identificadas, focando-se nos dados filtrados provenientes da fusão sensorial dos sensores inerciais (acelerômetro e girômetro), ultrassônicos e câmeras gerados pelo *firmware* do AR.Drone 2.0.

Em posse das dinâmicas que cerceiam o processo a ser identificado, é necessário definir o modelo linear que melhor englobe tais dinâmicas. Utilizaram-se duas abordagens principais. A primeira foi identificação feita no trabalho de Silveira et al. (2020). Nesse trabalho, o tipo de identificação foi caixa cinza e o modelo proposto foi sistema estocástico linear em tempo discreto MIMO. Optou-se por esse modelo de sistema em espaço de estados, pois os voos dos aeromodelos possuem perturbações não gaussianas, sendo necessário descrever os parâmetros de tais dinâmicas. O modelo MIMO possui três estados a mais do que o modelo identificado nessa dissertação, sendo eles o ângulo de guinada e a sua velocidade angular e a velocidade no eixo Z, dessa forma explorando todos os graus de liberdade que o drone oferece.

Porém, a identificação caixa cinza gera, segundo o conjunto de entradas e saídas dessa dissertação, 65 parâmetros, logo, o método de variáveis instrumentais demoraria muito para convergir, já que tal método é a estimação com base em outra estimação. Visto isso, essa dissertação apoia-se no trabalho de Santana, Brandão e Sarcinelli-Filho (2016), no qual é utilizado o método de identificação caixa cinza para estimar os parâmetros do modelo linear. O trabalho é feito no controle de alto nível do quadricóptero Bebop 2 com condições de comunicação muito parecidas com as do AR. Drone.

Nesse trabalho, são identificadas apenas as dinâmicas de altitude e atitude (movimento lateral e frontal), ignorando a malha de orientação (ângulo de guinada) e considera-se que os comandos impulso do Bebop 2 afetam apenas os respectivos eixos de ação (o impulso lateral afeta a velocidade lateral e assim sucessivamente), não levando em consideração o ângulo de inclinação para efetuar tais tarefas. Logo, as dinâmicas do drone nos eixos longitudinais e laterais são apresentadas na equação abaixo:

$$V_{eixo}(k) = AV_{eixo}(k-1) + Bu(k-d) \quad (3.1)$$

onde V_{eixo} é a velocidade em um eixo X ou Y. Nota-se que as velocidades estão em função apenas dos regressores delas e das entradas. Porém, tal simplificação é feita a partir da condição de que o ângulo feito para cada manobra seja menor que 5 graus, para manter as relações de linearidade do eixo com a aceleração resultante do comando de impulso.

Visto isso, o algoritmo proposto por essa dissertação optou pela relação impulso com a inclinação da manobra padronizada que quadricóptero realiza, sabendo que tal inclinação reflete na aceleração direção lateral e frontal do sistema. Dessa forma, gerando o seguinte modelo ARMAX:

$$x(k) = \begin{bmatrix} \rho(k) \\ \delta(k) \\ v_f(k) \\ v_l(k) \\ h(k) \end{bmatrix} = \begin{bmatrix} a_{11} & 0 & 0 & 0 & 0 \\ 0 & a_{22} & 0 & 0 & 0 \\ 0 & a_{32} & a_{33} & 0 & 0 \\ a_{41} & 0 & 0 & a_{44} & 0 \\ 0 & 0 & 0 & 0 & a_{55} \end{bmatrix} \begin{bmatrix} \rho(k-1) \\ \delta(k-1) \\ v_f(k-1) \\ v_l(k-1) \\ h(k-1) \end{bmatrix} + \begin{bmatrix} b_{11} & 0 & 0 \\ 0 & b_{22} & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & b_{53} \end{bmatrix} \begin{bmatrix} u_\rho(k-d) \\ u_\delta(k-d) \\ u_h(k-d) \end{bmatrix} \quad (3.2)$$

$$+ \begin{bmatrix} g_{11} & 0 & 0 & 0 & 0 \\ 0 & g_{22} & 0 & 0 & 0 \\ 0 & 0 & g_{33} & 0 & 0 \\ 0 & 0 & 0 & g_{44} & 0 \\ 0 & 0 & 0 & 0 & g_{55} \end{bmatrix} \begin{bmatrix} e_\rho(k-1) \\ e_\delta(k-1) \\ e_{v_f}(k-1) \\ e_{v_l}(k-1) \\ e_h(k-1) \end{bmatrix}$$

$$y(k) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} x(k) \quad (3.3)$$

Logo, diminuí o número de parâmetros que os estimadores devem calcular, pois os impulsos frontal e lateral são substituídos por impulsos na arfagem e na rolagem do quadricóptero. Essas entradas resultam na inclinação que permite o deslocamento no eixo X e Y do drone, por fim, o deslocamento em Z em função apenas da entrada impulso de altitude. Tais relações lineares compõem a parte determinística do modelo em espaço de estados.

Porém, os quadricópteros estão suscetíveis a dinâmicas externas ao processo, visto isso, adicionou-se a parte estocástica do sistema que modela o comportamento das perturbações que o afetam, gerando o modelo discreto em espaço de estado presente na equação (3.2).

4 Experimentos de Identificação

Nesse capítulo, são relatados os resultados dos experimentos de identificação das dinâmicas do quadricóptero. A primeira parte apresenta o modelo referência, na qual será base da avaliação dos desempenhos de identificação e de controle. Em seguida, são apresentados dados provenientes das aplicações dos métodos já consolidados (MQRE, FKE) e do método proposto (FKEVI+FK) debatendo o impacto da nova abordagem.

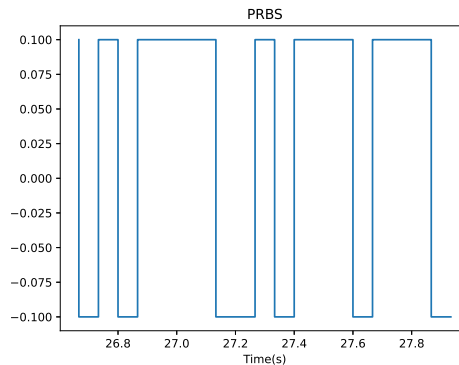
4.1 Modelo Linear de Referência

Em posse do modelo linear proposto, estimaram-se os parâmetros que serão referência para a avaliação dos métodos de identificação e controle utilizados no trabalho. O método utilizado para estimar os parâmetros foi MQRE. Isso se justifica, pois os dados recebidos pelo Nav-Data já foram filtrados pelo *firmware* do drone, logo diminuindo a interferência de ruídos de medição. Porém, serão utilizados os outros métodos de estimação visando evidenciar como cada método pode calcular, com o mesmo conjunto de dados, parâmetros diferentes.

Para estimar parâmetros, é muito importante utilizar padrões de entradas que percorrem os mais diferentes níveis de frequência, dessa forma, explorando grande parte das dinâmicas do processo, resultando em modelos que descrevem com maior precisão o processo (COELHO; COELHO, 2015). Visto isso, o padrão de entrada utilizado foram sinais gerados pelo algoritmo *Pseudorandom Binary Sequence* (PRBS). Esse algoritmo gera sinais binários que apresentam uma distribuição normal varrendo todo o espectro de frequência. Porém, utilizar um sinal aleatório (ou pseudoaleatório) nos comandos de quadricópteros pode oferecer um risco e dano às pessoas e objetos ao redor. Logo, junto ao PRBS foi adicionado sinal do tipo degrau enviado pelo próprio notebook durante o experimento, pois a biblioteca PS-Drone permite o envio de comandos via teclado. Esse sinal de degraus enviados pelo prompt de comando do sistema operacional Ubuntu da Linux permite que o usuário evite colisões do drone nos momentos de risco eminente durante o experimento, tendo pouca influência no conjunto de sinais de entrada enviados para o quadricóptero.

O risco que o conjunto aleatoriedade dos dados de entradas enviados para o quadricóptero justifica a escolha do ambiente de teste selecionado, pois atualmente o lugar não está em funcionamento, logo diminuindo o risco de acidentes que possam causar ferimentos em pessoas e danos ao patrimônio público. A Figura 9 mostra como o sinal pseudoaleatório PRBS é formado.

Figura 9. Sinal PRBS



Fonte: elaborado pelo autor do texto

Outro ponto importante é o atraso do sistema, para isso foi considerado o atraso de comunicação e o atraso de acionamento, resultando no atraso de 2 tempos discretos, $d = 2$. No experimento, para levantar os parâmetros referenciais, foi feita a manobra padronizada de decolagem do drone com a bateria cheia. Após isso, é inserido o sinal PRBS que se limita -0.1 a 0.1 acompanhados de sinais do tipo degraus de -0.2 a 0.2 . Esse experimento durou entorno de 10 minutos até o descarregamento total da bateria do quadricóptero e o algoritmo em Python 3 utilizado para extração e envio de dados para o quadricóptero AR.Drone 2.0 está exposto no Anexo A.

O método de identificação MQRE aplicado nos dados de entrada e saída do quadricóptero apresentou o índice NRMSE presente na Tabela 1. O cálculo do índice desconsiderou os dois minutos iniciais do experimento, pois são considerados o tempo de convergência dos parâmetros.

Tabela 1. NRMSE Referencial

Estado	NRMSE
ρ	0.989285
δ	0.989031
v_f	0.994554
v_l	0.99841
h	0.99992

Na equação abaixo apresentam-se os parâmetros em que o modelo de referência é baseado. Esses parâmetros são os últimos estimados pelo método de identificação.

$$\begin{aligned}
\begin{bmatrix} \rho(k) \\ \delta(k) \\ v_f(k) \\ v_l(k) \\ h(k) \end{bmatrix} &= \begin{bmatrix} 0.869 & 0 & 0 & 0 & 0 \\ 0 & 0.882 & 0 & 0 & 0 \\ 0 & -0.716 & 0.975 & 0 & 0 \\ 0.601 & 0 & 0 & 0.966 & 0 \\ 0 & 0 & 0 & 0 & 1.0013 \end{bmatrix} \begin{bmatrix} \rho(k-1) \\ \delta(k-1) \\ v_f(k-1) \\ v_l(k-1) \\ h(k-1) \end{bmatrix} \\
&+ \begin{bmatrix} 0.058 & 0 & 0 \\ 0 & -0.046 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0.029 \end{bmatrix} \begin{bmatrix} u(k-d) \\ u_\delta(k-d) \\ u_h(k-d) \end{bmatrix} \\
&+ \begin{bmatrix} 0.55 & 0 & 0 & 0 & 0 \\ 0 & 0.64 & 0 & 0 & 0 \\ 0 & 0 & 0.197 & 0 & 0 \\ 0 & 0 & 0 & 0.216 & 0 \\ 0 & 0 & 0 & 0 & 0.43 \end{bmatrix} \begin{bmatrix} e_\rho(k-1) \\ e_\delta(k-1) \\ e_{v_f}(k-1) \\ e_{v_l}(k-1) \\ e_h(k-1) \end{bmatrix}
\end{aligned} \tag{4.1}$$

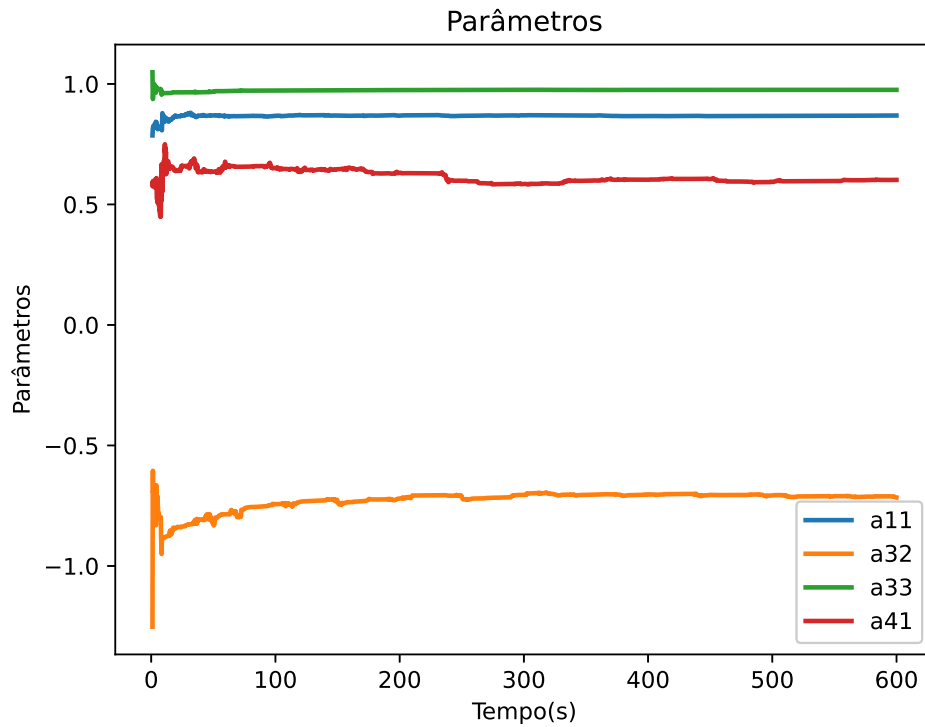
Em posse do modelo do quadricóptero, estabeleceram-se os erros de estimação e os regressores desse erro que são considerados semelhantes as perturbações estocásticas do sistema conforme a seção 2.5. Seus valores estão representados na Tabela 2. Nota-se que a potência dos ruídos de processo são maiores nos estados de velocidade do sistema (velocidade frontal e velocidade lateral), pois são estados que mais estão expostos às dinâmicas ambientais de voo que perturbam o sistema (como ventos, temperatura do ar e etc.). Logo, é de extrema importância que a parte estocástica da perturbação seja estimada, pois se trata de uma dinâmica inerente de aeromodelos.

Tabela 2. Variância da Perturbação

Estados	$E[ww^T]$
ρ	0.00035
δ	0.00031
v_l	0.00157
v_f	0.00199
h	0.0002

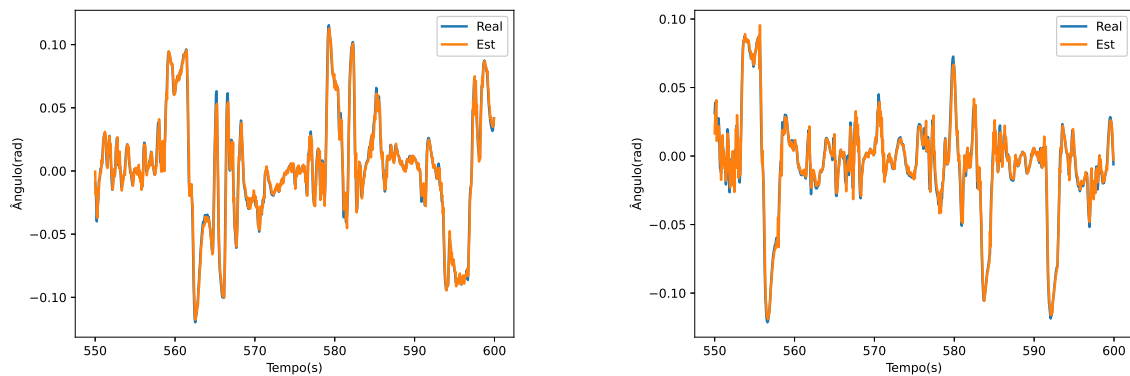
A técnica MQRE permite que sejam levantados os parâmetros não só ao fim do experimento (como o MQ), mas durante todo o processo de estimação, possibilitando analisar a evolução dos parâmetros e como eles convergem ao longo do tempo. A convergência está representada na Figura 10, note que, após alguns instantes, os parâmetros se mantêm estáveis até o fim do experimento, isso se justifica devido ao ambiente de teste *indoor* minimizar as perturbações que podem afetar bruscamente o sistema.

Figura 10. Parâmetros



Fonte: elaborado pelo autor do texto

Somando todos os índices NRMSE obtidos em cada estado do sistema, o modelo estimado obteve NRMSE geral em torno de 0,99, superando em todos os estados a marca 0,8 na qual Coelho e Coelho (2015) considera aceitável. Logo, os parâmetros descrevem com precisão o comportamento do voo do quadricóptero, como demonstrado nas Figuras a seguir:



(a) Arfagem

(b) Rolagem

Figura 11. Ângulos

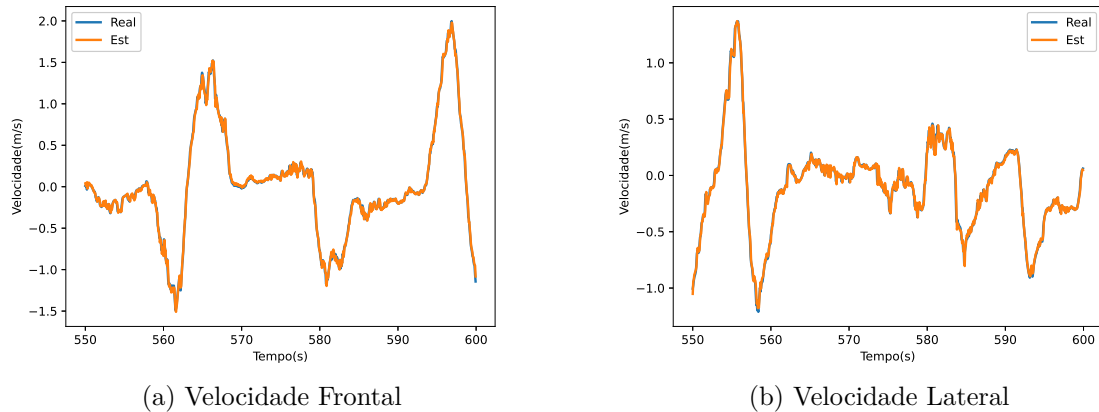
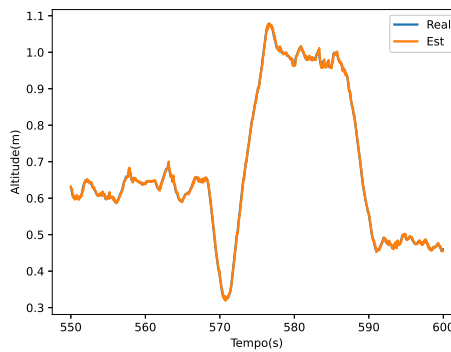


Figura 12. Velocidades

Figura 13. Altitude



Fonte: elaborado pelo autor do texto

Definido o sistema linear referência da dissertação, aplicaram-se o FKEVI+FK e o FKE para identificar as dinâmicas do processo com os mesmos dados de entrada e saída. Para a aplicação do FKE foi informado a matriz da variância da perturbação estabelecido pelo modelo referencial, Q_r , e estabeleceu-se a matriz de ruído com pouca potência de ruído, $R_{FKE} = 0.001 \times I_{n \times n}$, já para o método FKEVI+FK não foi informado as matrizes deixando a cargo do próprio algoritmo calcular as intensidades dos ruídos, das perturbações e os parâmetros do quadricóptero. O resultado do desempenho deste experimento está exposto na Tabela 3, onde a sigla FKEVI+KF é abreviada para FKEVI.

Tabela 3. Parâmetros Estimados pelo FKE e FKEVI+FK

	a_{11}	a_{22}	a_{32}	a_{33}	a_{41}	a_{44}	a_{55}	b_{11}	b_{22}	b_{53}
FKE	0.85	0.85	-0.73	0.97	0.61	0.96	1.00	0.066	-0.055	0.029
FKEVI	0.85	0.86	-0.75	0.97	0.61	0.96	0.99	0.063	-0.052	0.032

Percebe-se que os parâmetros do FKE se aproximam dos parâmetros do FKEVI+FK, logo a técnica não polarizante (estimador VI) e o observador KF não distorcem o processo de estimação, fazendo com que FKE e o método proposto tenham comportamento parecido. Outro ponto importante é o FKEVI+FK identifica a presença de ruído nos dados, principalmente nos sensores de velocidade no eixo X e Y . A potência dos ruídos estimados está presente na Tabela abaixo:

Tabela 4. Ruídos Estimados pelo FKEVI+FK

Sensor	$E[\hat{v}\hat{v}^T]$
ρ	0.0001
δ	0.00008
v_f	0.00403
v_l	0.00209
h	0.00002

Porém a possível polarização dos parâmetros resultantes da interferência do ruído no método MQRE que definiu o modelo referência do trabalho será descartada, pois além da potência do ruído de medição constatado pelo FKEVI+KF ser de pequena magnitude, o método proposto está em fase de consolidação impossibilitando essa afirmação.

4.2 Resultados da Identificação

Em posse dos dados do experimento, testaram-se três estimadores paramétricos: os já conhecidos na literatura MQRE e FKE e o método proposto FKEVI+FK . Nesses testes, foram adicionados ao sinal do primeiro experimento 3 tipos de ruídos de medição, gerando 3 conjuntos de dados diferentes. Os conjuntos têm a relação sinal ruído de (SNR) de 10, 50 e 100 respectivamente. A relação está descrita na equação 4.2, onde P_{signal} é a potência do sinal advindo do sensor (saída) e P_{noise} é a potência do ruído gaussiano adicionado ao sinal dos sensores nesse experimento, $v(k)$. A variância dos ruídos nos sensores está presente na Tabela a seguir:

$$SNR = \frac{P_{signal}}{P_{noise}} \quad (4.2)$$

Esses ruídos Gaussianos são adicionados aos dados de saída do primeiro experimento. Sendo assim, os estimadores processaram dados ruidosos para calcular parâmetros em um processo MIMO com perturbações estocásticas e ruídos para três experimentos com tipos de intensidade de ruído diferentes. O intuito de adicionar esses ruídos nos dados de saída do quadricóptero é para avaliar o impacto dessa dinâmica nesses três cenários (SNR 10, 50, 100) no processo de estimação dos algoritmos utilizados, dessa forma, possibilitando a comparação dos desempenhos.

Tabela 5. Potência dos Ruídos

SNR	$E[vv^T]$				
	ρ	δ	v_f	v_l	h
10	0.00414857	0.00385873	0.0611042	0.0442151	0.0385279
50	0.000829713	0.000771746	0.0122208	0.00884303	0.00770558
100	0.000414857	0.000385873	0.00611042	0.00442151	0.00385279

Por mais que em alguns trabalhos o FKE tenha uma versão adaptativa (XIONG et al., 2012), onde as matrizes R e Q do FKE são calculados pelo próprio algoritmo, nesse trabalho, foi informado as variâncias do ruído de medição e das perturbações para o algoritmo, permitindo a análise do desempenho no ambiente ao qual ele foi teorizado. Já as configurações do MQRE, como já debatido, são uma solução particular de Kalman, logo a saída estimada também é filtrada, podendo ser comparada com os demais estimadores de estado (FKE e FKEVI).

Outra ressalva importante é que a avaliação dos parâmetros levantados pelos métodos está focada nos elementos das matrizes A_{est} e B_{est} , pois, segundo Aguirre (2015), a matriz G_{est} estimada por diferentes métodos de identificação tende a variar tanto os parâmetros como as variâncias dos resíduos, pois se trata da parte estocástica do modelo. Logo, se a parte determinística do modelo converge para parâmetros reais, a parte estocástica do sistema apresenta um comportamento semelhante à estimada pelo método de identificação. Outro ponto é que o LQG é sintonizado com a parte não estocástica do sistema, assim apenas a parte determinística tem efeito no projeto do controlador. Segue o gráfico e a tabela de desempenho de cada saída estimada com a referente relação sinal ruído:

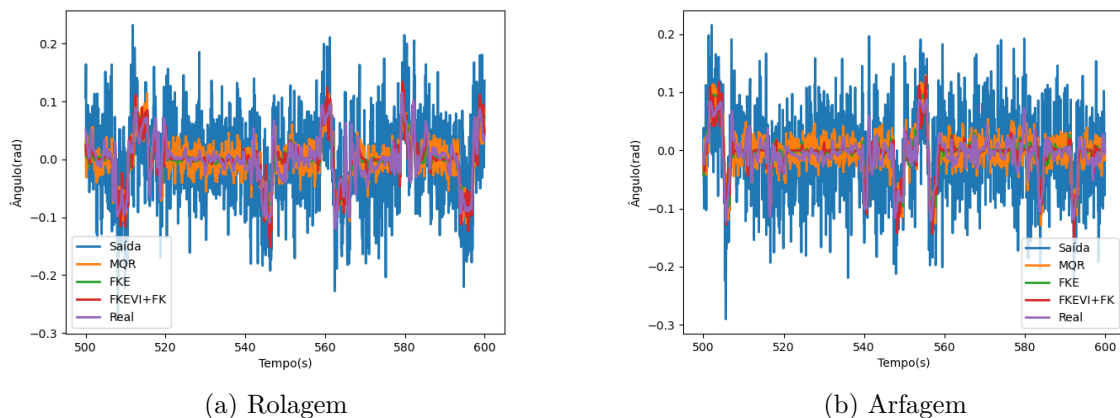


Figura 14. Ângulos com SNR 10

Todos os métodos convergiram, em certo grau, com a saída real do sistema promovendo a filtragem dos dados de saídas afetados pelos ruídos, e que, à medida que o ruído

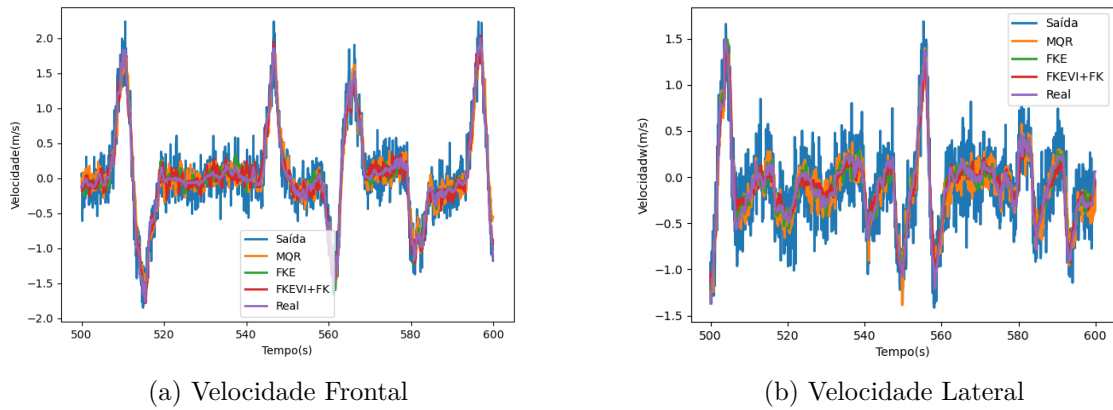


Figura 15. Velocidades com SNR 10

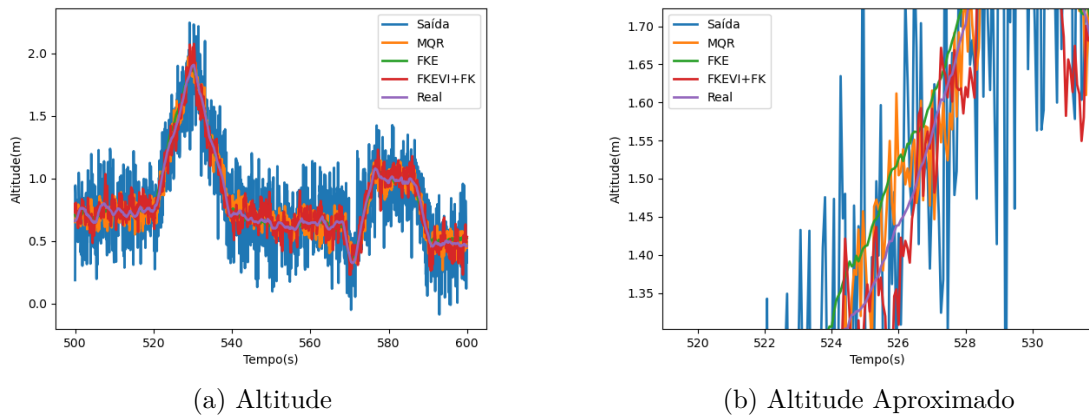


Figura 16. Altitude com SNR 10

Tabela 6. Índices NRMSE dos Experimentos de Identificação na Presença de Ruído

	SNR	ρ	δ	v_f	v_l	h
MQRE	10	0.829656	0.803918	0.975699	0.947542	0.972193
	50	0.861188	0.841475	0.988919	0.970234	0.980534
	100	0.869031	0.85573	0.990932	0.974926	0.982044
FKE	10	0.8873	0.860018	0.964193	0.973995	0.99319
	50	0.937721	0.847192	0.992972	0.990399	0.995893
	100	0.952045	0.936542	0.997073	0.993426	0.997211
FKEVI	10	0.883595	0.886686	0.987798	0.967583	0.967225
	50	0.937746	0.927651	0.994313	0.983158	0.99295
	100	0.953615	0.950858	0.995464	0.987242	0.996753

tem menor intensidade, as saídas estimadas obtêm índices melhores. Um ponto importante é que a filtragem pelo MQRE deixa explícito como o método de estimação e o modelo escolhido impõem limitações que não permitem agregar todos os dados, resultando na

filtragem de dinâmicas incapazes de serem assimiladas pelo modelo. Nota-se que o FKE, durante o experimento, apresenta um resultado satisfatório com bons índices NRMSE, porém no cenário com maior intensidade de ruído ($SNR = 10$) foi preciso repetir a simulação para que os parâmetros convergissem. Tal comportamento é relatado no trabalho Abreu, Oliveira e Neto (2020) onde é abordado o comportamento variável do FKE em sistemas ruidosos.

Por fim, FKEVI+FK apresentou os melhores resultados de estimação dos estados sem que as matrizes R e Q fossem informadas, dessa forma, tendo que quantificá-las com os parâmetros para oferecer as bases para a estimação de FK. Segue a Tabela que apresenta os parâmetros calculados por cada técnica nos três diferentes cenários:

Tabela 7. Parâmetros Estimados na Presença de Ruído.

	SNR	a_{11}	a_{22}	a_{32}	a_{33}	a_{41}	a_{44}	a_{55}	b_{11}	b_{22}	b_{53}
MQRE	10	0.82	0.82	-0.39	0.982	0.319	0.973	0.97	0.083	-0.064	0.4
	50	0.83	0.84	-0.55	0.98	0.46	0.97	0.99	0.078	-0.063	0.38
	100	0.84	0.85	-0.64	0.977	0.51	0.968	0.99	0.07	-0.6	0.39
FKE	10	0.85	0.73	-1.03	0.963	0.75	0.95	0.99	0.63	-0.1	0.45
	50	0.72	0.73	-0.92	0.967	0.67	0.96	0.99	0.124	-0.1	0.44
	100	0.74	0.74	-0.86	0.969	0.67	0.961	0.99	0.11	-0.09	0.47
FKEVI	10	0.85	0.83	-0.84	0.97	0.64	0.96	0.99	0.061	-0.064	0.38
	50	0.84	0.86	-0.79	0.97	0.63	0.966	0.99	0.066	-0.051	0.40
	100	0.85	0.85	-0.76	0.974	0.609	0.967	0.99	0.6	-0.056	0.36

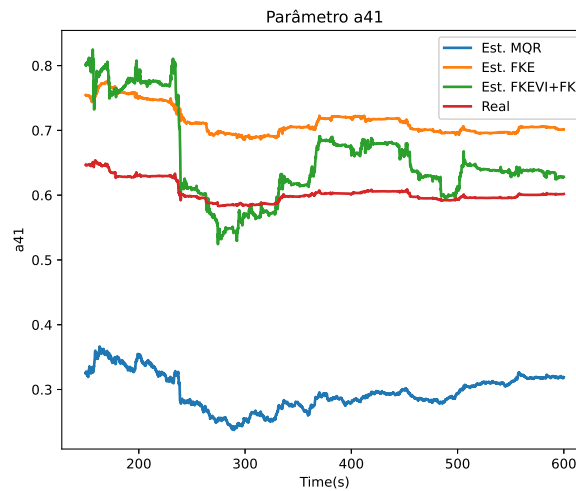
Tabela 8. A Distância Euclidiana na Presença de Ruído.

	SNR	D_θ
MQRE	10	0.781
	50	0.431
	100	0.26
FKE	10	0.81
	50	0.731
	100	0.646
FKEVI	10	0.28
	50	0.19
	100	0.144

O conjunto dos dados de desempenho de estimação mostra que o FKE oferece uma melhor estimação dos estados em relação ao MQRE, em contrapartida, o MQRE oferece estimativas paramétricas mais próximas dos reais parâmetros, e que tais características

ficam menos destoantes quando se diminui a potência do ruído. Já o FKEVI+FK se diferencia dos demais por estimar melhor tanto os estados como os parâmetros na maioria dos cenários, devido ao seu comprometimento teórico em diminuir a interferência de ruído no seu processo de estimação. Esse comprometimento se torna mais alarmante quando se observa a Figura 17 na qual representa as estimações dos parâmetros a_{11} no decorrer do experimento, em que os dados de saída estão afetados pelo ruído com SNR 10.

Figura 17. Parâmetro a_{41} no cenário de SNR 10



Fonte: elaborado pelo autor do texto

Nota-se que as técnicas baseadas na variância mínima de suas estimações (FKE, MQRE) estagnam com o decorrer do experimento, em estimações paramétricas mais distantes do valor referencial que o algoritmo proposto. Tal dinâmica ocorre nos outros parâmetros do modelo, explicitando o porquê do método proposto oferecer parâmetros menos polarizados. Isso é possível, pois a comunicação estabelecida pelo compartilhamento da matriz de variância, M_k , do estimador VI com FKE faz com que o processo de filtragem acompanhe o processo de estimação paramétrica, fazendo com que a convergência do FKE do FKEVI+FK seja mais lenta e criteriosa que o FKE tradicional. Nota-se que os modelos levantados pelo FKEVI+FK em todos os cenários passaram no teste de observabilidade e controlabilidade de sistema MIMO e que as matrizes de variância do ruído de medição estimada pelo algoritmo proposto, obteve valores próximos dos valores dos ruídos adicionados às leituras dos sensores como mostra a tabela.

Tabela 9. Estimacões dos Ruídos pelo FKEVI+FK

SNR	100		50		10	
Variância	$E[\hat{v}\hat{v}^T]$	$E[vv^T]$	$E[\hat{v}\hat{v}^T]$	$E[vv^T]$	$E[\hat{v}\hat{v}^T]$	$E[vv^T]$
ρ	0.0006	0.0004	0.0011	0.0008	0.0045	0.0041
δ	0.0005	0.0003	0.001	0.0007	0.00452	0.0038
v_f	0.0128	0.0061	0.02	0.012	0.075	0.0611
v_l	0.0124	0.0044	0.019	0.0088	0.062	0.044
e	0.0045	0.0038	0.0088	0.0077	0.044	0.038

5 Resultados de Controle

A partir dos modelos levantados pelos algoritmos de identificação, são produzidas simulações de controle adaptativo utilizando o modelo de referência para assumir o papel do quadricóptero. Tal modificação é necessária, pois o algoritmo proposto está no processo de desenvolvimento e consolidação, além de que os ruídos empregados nos sensores são de alta magnitude, possibilitando que o controle em malha fechada apresente um comportamento perigoso, ocasionando perdas materiais e até ferimentos em pessoas. Abaixo segue o sistema linear expandido pelo qual o LQG foi sintonizado. Note que há diferença na matriz C , pois o C_c é uma matriz que correlaciona os estados com a saída desejada do sistema, y_d , que são velocidade lateral, velocidade frontal. Já C_o é referente aos estados que são filtrados, y_f .

$$\begin{bmatrix} y_f(k) \\ \Delta x(k) \end{bmatrix} = \begin{bmatrix} 1 & C_f A \\ 0 & A \end{bmatrix} \begin{bmatrix} y(k-1) \\ \Delta x(k-1) \end{bmatrix} + \begin{bmatrix} C_f B \\ B \end{bmatrix} \Delta u(k-d) \quad (5.1)$$

$$y_d(k) = \begin{bmatrix} C_c & 0 \end{bmatrix} \begin{bmatrix} y_f(k) \\ \Delta x(k) \end{bmatrix} \quad (5.2)$$

$$(5.3)$$

Os LQGs adaptativos projetados nesse experimento têm o objetivo de controlar o sistema com base nos parâmetros estimados no experimento anterior. Os valores dos parâmetros utilizados são os do experimento com a $SNR = 10$ e o modelo referência é afetado pela mesma intensidade de ruídos em que os parâmetros foram estimados. Uma das características pelas quais se optou aplicar o LQG nesse trabalho é o fato de ser uma técnica de otimização que tira a responsabilidade projetista de oferecer os ganhos do controlador. Porém, o ponto sensível da sintonia LQG é estabelecer os valores das matrizes Q_c e R_c , pois em muitos casos os parâmetros e as matrizes de covariância do ruído de medição e de perturbações, R e Q , já são conhecidas ou se têm uma boa estimativa, e a partir desses dados se projeta o FK otimizado.

Visto isso, para estabelecer as matrizes de ponderação de controle do LQG utilizou-se o método Bryce. De acordo com este método, as matrizes de ponderação são matrizes diagonais cujos elementos são expressos como os inversos dos quadrados dos valores máximos aceitáveis da variável de estado e da variável de controle de entrada respectivamente (ELKHATEM; ENGIN, 2022). Os elementos diagonais das matrizes Q_c e R_c podem ser escritos como:

$$Q_{ii} = \frac{1}{(\text{valor máximo aceitável de } X_i)^2} \quad (5.4)$$

$$R_{ii} = \frac{1}{(\text{valor máximo aceitável de } u_i)^2} \quad (5.5)$$

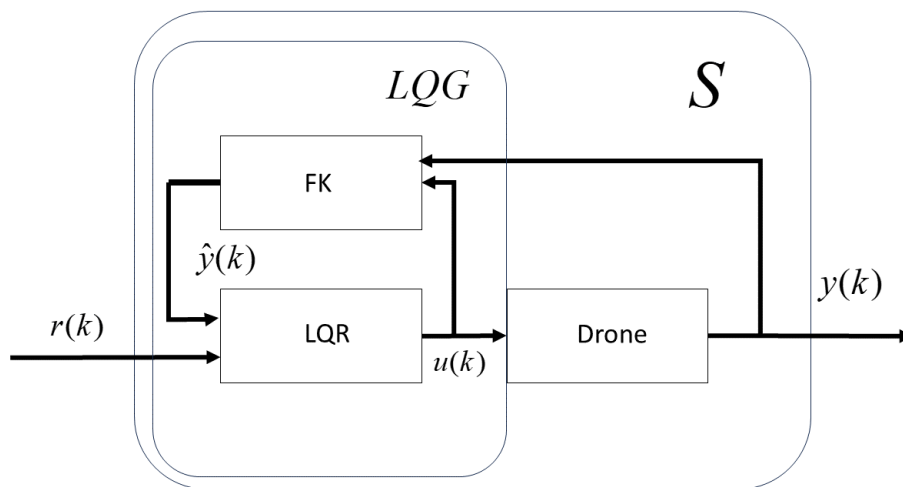
Onde i são elementos diagonais das matrizes. Visto isso, os estados que mais se têm interesse em estabilizar são os de variação dos estados, Δx , tendo os pesos de 10^5 , já os estados que devem seguir o sinal referência, obtendo valores de 100 e para os estados restantes 1, tendo os valores mostrados na equação (5.6). Já a matriz de ponderação de entrada optou-se por usar valores próximos da área de saturação dos acionadores do sistema $([-1,1])$ como presente na equação (5.7).

$$Q_c = \text{diagonal} (1 \ 1 \ 1000 \ 1000 \ 1000 \ 10^5 \ 10^5 \ 10^5 \ 10^5 \ 10^5) \quad (5.6)$$

$$R_c = \text{diagonal} (0,25 \ 0,25 \ 0,25) \quad (5.7)$$

A partir das matrizes de ponderação, o LQG referência é projetado conforme a seção 2.8. O LQG referência é sintonizado usando o modelo de referência com a matriz de covariância de perturbações e com a matriz de covariância dos ruídos que afetam o sensor do sistema simulado do quadricóptero. O ganho K_c ótimo faz com que se estabeleça o comando de controle do sistema em malha fechada que está representado Figura 18.

Figura 18. LQG Ideal



Fonte: elaborado pelo autor do texto

Onde S representa o sistema em malha fechada. A função S permite analisar o comportamento do sistema em todo o espectro frequencial e extrair os índices de robustez do sistema controlado. Porém, é contraproducente analisar a resposta frequência do sistema em cada estado individualmente como se faz em sistemas SISO aplicando o diagrama de Bode, pois os estados do sistema se entrecruzam. Visto isso, o trabalho de (DOYLE, 1982) estabelece que para avaliar o processo é necessário descrevê-lo em valores singulares que exprimem o comportamento da planta MIMO naquela frequência determinada.

Em posse da função S do sistema em malha fechada, são calculadas as curvas de ganho dB e de fase do sistema controlado pelo LQG e do quadricóptero em malha aberta, tal resultado só foi possível devido à implementação das equações na linguagem Python 3, pois até o momento da escrita dessa dissertação, não foi encontrado uma função que execute a tarefa de calcular os valores singulares de um sistema linear MIMO discreto. O trecho de algoritmo está disponível para replicação no Anexo B.

Em posse dos dados de reposta em frequência dos sistemas em malha fechada e aberta, é estruturada a Figura 19, onde a linha azul representa as linhas do sistema e a vermelha as curvas de dinâmicas externas, S . Percebe-se que a planta em malha aberta tem altos ganhos nas zonas de baixa frequência. Porém, ao adicionar o LQG com o sistema expandido à planta, as baixas frequências se aproximam do zero dB . Dessa forma, o processo segue a referência ao degrau sem erro estacionário.

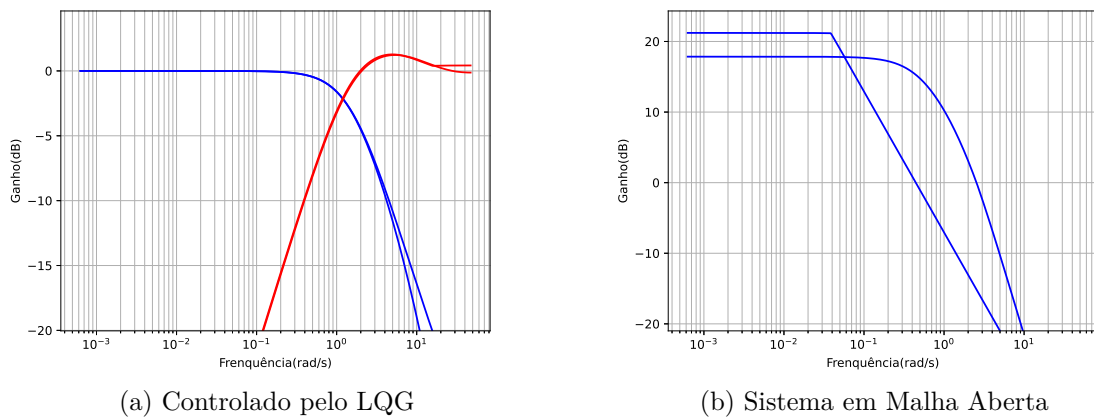


Figura 19. Curvas T e S

Como o problema proposto é o segmento de referência do tipo degrau, simula-se a resposta ao degrau unitário no sinal de entrada do sistema em malha fechada com o intuito de obter os índices desempenho ISE e ISU algoritmo. Nesse experimento, os ruídos de medição que afetam o quadricóptero têm as variações equivalentes às calculadas com a $SNR = 10$ do experimento de identificação. Importante ressaltar que os resultados desse experimento são referência para a análise, pois o LQG projetado com os parâmetros referências do sistema apresenta a sintonia “mais ótima” do LQG. Por mais que tal citação

pareça absurda, na realidade não é, pois, se o método para identificar o modelo se distancia do real modelo, o LQG calculado com o modelo defasado não é o melhor “ótimo possível”.

A tabela abaixo mostra o desempenho individual de cada saída desejada, das entradas, a margem de ganho e a margem de fase do sistema MIMO do quadricóptero.

Tabela 10. Resultados Referência de Controle

	ISE_{v_l}	ISE_{v_f}	ISU_{u_ρ}	ISU_{u_δ}	GM_{db}	PM°
LQG Referência	0.026	0.023	0.062	0.065	6,206	51.01

O índice de desempenho das saídas desejadas (velocidade lateral e frontal) apresentou valores ISE próximos, pois o ganho de ponderação dado foi o mesmo para os estados que correspondem às saídas do processo, assim como foi dada a mesma ponderação para as entradas. As figuras abaixo mostram como o LQG filtra com o FK e os comandos de controle utilizam esse estado estimado para diminuir o esforço de controle.

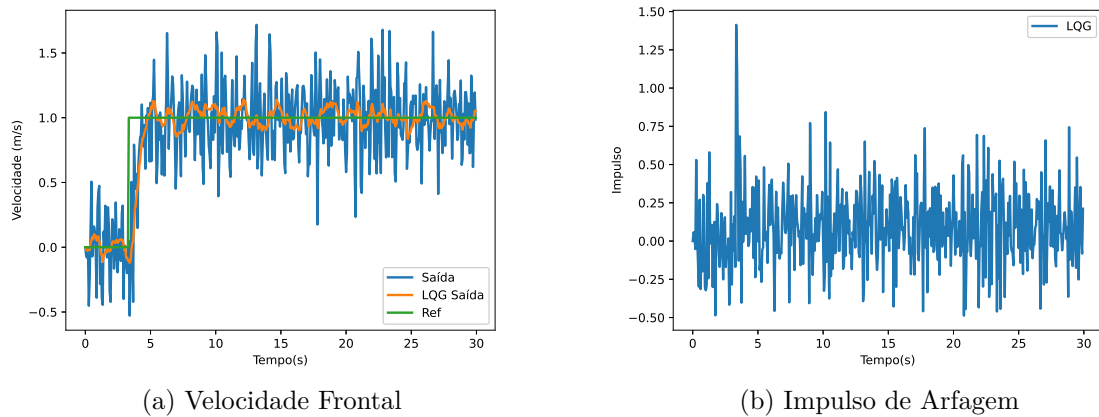


Figura 20. Malha de Velocidade Frontal

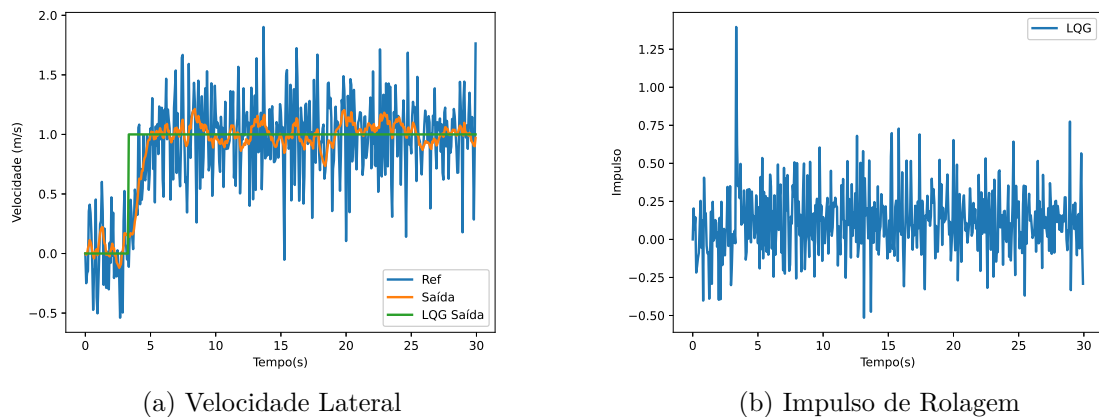
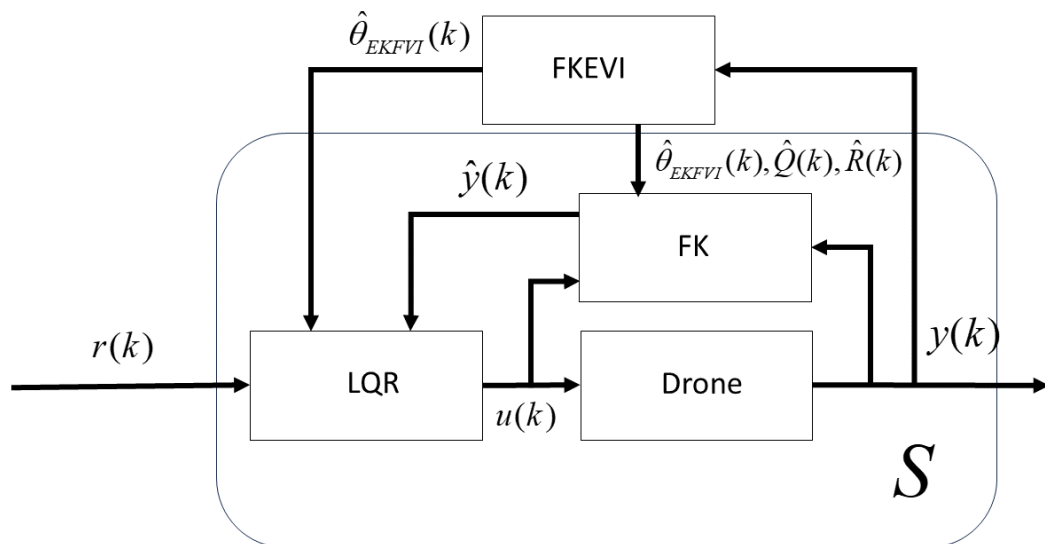


Figura 21. Malha de Velocidade Lateral

Após isso, foram feitas as simulações do processo em malha fechada utilizando os métodos de identificações desenvolvidos nessa dissertação associados ao LQG adaptativo. Os métodos utilizaram os parâmetros estimados na seção resultados estimação e, a partir desses dados, são desenvolvidos os controladores ótimos adaptativos. Porém, a planta a qual esses controladores são sintonizados é o modelo referencial, permitindo a avaliação do quanto a técnica de controle é penalizada, tanto por desempenho quanto por robustez, conforme o método de identificação utilizado. Nessa parte do trabalho, experimentam-se o FKE e o FKEVI+FK pois o algoritimo MQRE não possui observadores de estado, logo impedindo a sua aplicação no controlador LQG.

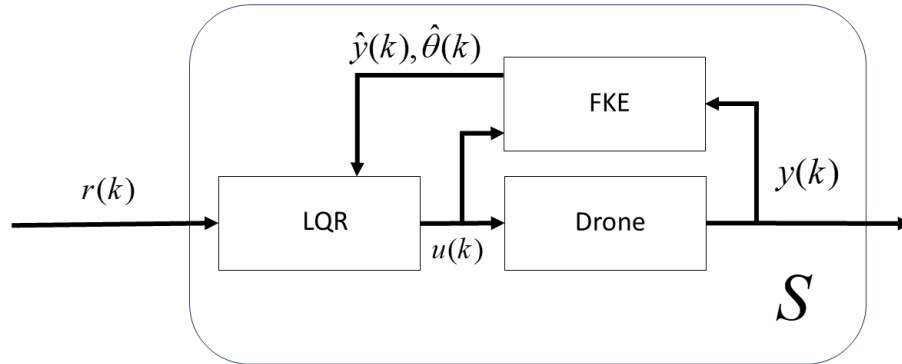
A adição de observadores, como o FKEVI+FK e FKE, faz-se necessário expandir a análise do sistema em malha fechada para englobar a dinâmica do controlador com a dinâmica do observador (filtro) em conjunto com o processo, pois as variáveis de estado estimadas pelos filtros ótimos são enviadas para o controlador LQG. Tal relação é exposta na equação (2.103). Logo, cada um dos estimadores conjuntos tem uma função em malha fechada diferente, onde os ganhos do observador e do controlador estão baseados no resultado da estimação conjunta dos métodos. Com base nisso, foi levantada a função S em cada um dos cenários (FKE e FKEVI+FK). Abaixo segue os diagramas da função S :

Figura 22. Diagrama do LQG com o FKEVI+FK



Fonte: elaborado pelo autor do texto

Figura 23. Diagrama do LQG com FKE



Fonte: elaborado pelo autor do texto

Em posse da função S dos LQGs adaptativos, levantou-se as curvas de resposta em frequência real dos sistemas MIMOs controlados. O termo real serve para destacar que a parte do estado estimado do compensador presente na equação (2.103) e os ganhos do observador e do controlador foram calculados pelos dados provenientes da estimação conjunta, e a parte superior da equação onde estão os estados reais do compensador, são os parâmetros do modelo referência.

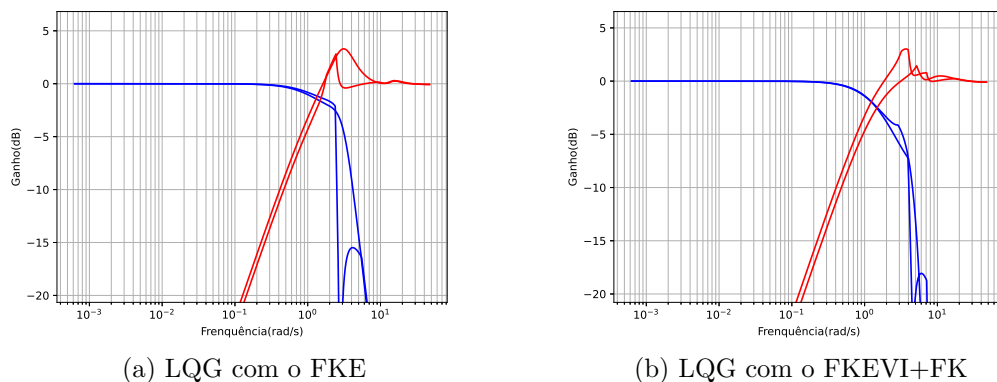


Figura 24. Curvas S e T dos Controladores Adaptativos

Por fim, é feito teste de desempenho dos controles adaptativos em malha fechada no seguimento de referência, onde o sistema inicia no seu ponto de equilíbrio e o sinal de entrada é o degrau unitário nas mesmas condições do experimento feito para estabelecer os índices de desempenho do LQG ideal.

Abaixo segue a Tabela dos índices de controle utilizados nessa dissertação:

Tabela 11. Resultados do Controle Adaptativo

	ISE_{v_l}	ISE_{v_f}	ISU_{u_p}	ISU_{u_s}	GM_{db}	PM°
LQG Referência	0.026	0.023	0.062	0.065	6,206	51.01
LQG FKE	0.033	0.027	1.6	1.04	6,0	39
LQG FKEVI+FK	0.0266	0.0261	0.102	0.28	6,1	42,3

Note que o desempenho do algoritmo proposto se aproxima muito dos reais índices do LQG referencial, logo se aproximando da solução ideal do projeto de controle. Porém, o LQG associado ao FKE teve os índices de esforço de controle muito prejudicados devido ao cálculo dos ganhos do controlador utilizar dados mais polarizados, tal dinâmica é indesejada em projetos de controle reais, pois ocasiona no desgaste do acionador do sistema e aumento da energia enviada para controlar o sistema. Logo, se aplicado no sistema de piloto automático do quadricóptero, a bateria descarregaria de maneira mais rápida, comprometendo a execução da tarefa a qual o drone foi submetido. Os desempenhos dos controladores adaptativos podem ser visualizados nas Figuras abaixo:

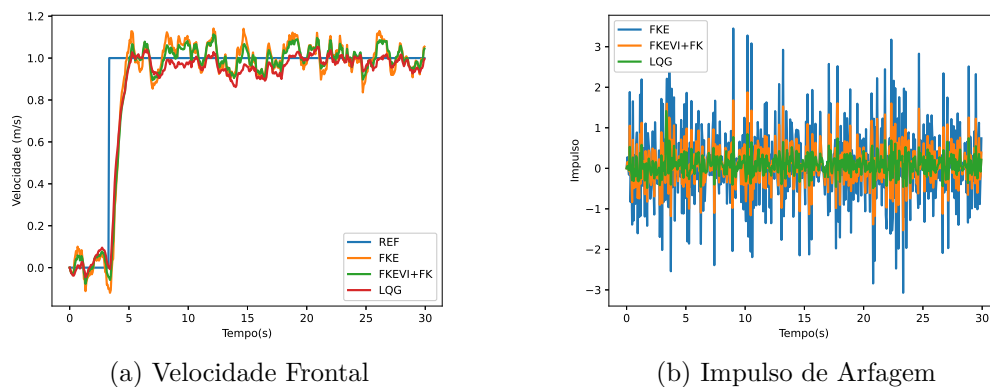


Figura 25. Malha de Velocidade Frontal

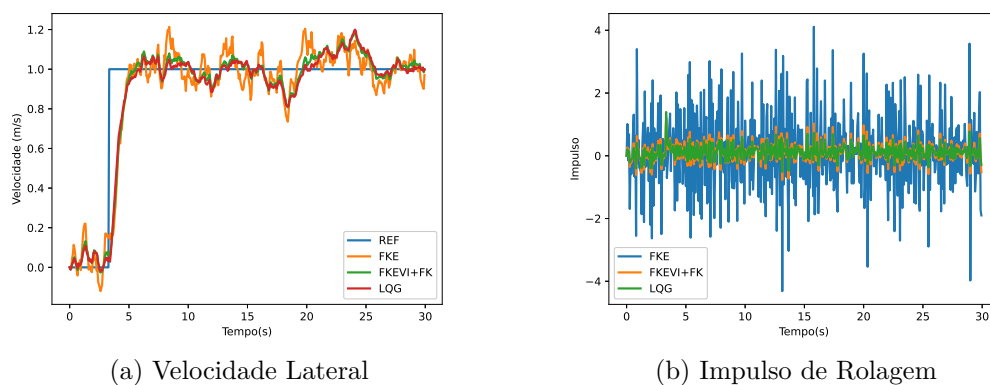


Figura 26. Malha de Velocidade Lateral

Por mais que na análise de robustez do sistema a margem de fase e margem de ganho real do sistema controlado pelo LQG ideal fossem próximas dos controladores adaptativos projetados nesse experimento, os desempenhos apresentam comportamentos muito distintos, com o LQG adaptativo associado ao FKE tendo o custo de esforço muito maior e obtendo um desempenho pior no seguimento de referência que o LQG adaptativo associado ao FKEVI+KF evidenciando a importância e o impacto das técnicas de estimação em tempo real no comportamento em malha fechada.

6 Conclusão

A principal relevância do trabalho está no impacto das técnicas de estimação paramétrica e de estados na sintonia de controladores adaptativos, pois através dos dados obtidos, nota-se a necessidade de viabilizar técnicas de estimação mais seguras. Atento a esse fato desenvolve-se a técnica de estimação conjunta em tempo real FKEVI+FK. Tal técnica resultou em melhora na estimação de parâmetros menos polarizados nas mais diversas intensidades de ruído. A partir desses dados se estabeleceram filtros (observadores) otimizados que conseguiram atenuar e estimar as potências dos ruídos que os afetam.

A técnica desenvolvida se tornou base para o desenvolvimento de controladores ótimos adaptativos, onde foram feitos testes de robustez e desempenho, obtendo resultados melhores que métodos consagrados na literatura (FKE). Todos os testes mencionados usaram dados provenientes do voo real do quadricóptero AR.Drone 2.0, impondo o cenário real no processamento das técnicas mencionadas na dissertação. Logo, as hipóteses 1 e 2 são desenvolvidas e comprovadas nos experimentos realizados, comprovando o caráter não polarizante do FKEVI+FK sendo assim o ponto de partida para a aplicação em outros processos.

Importante analisar que a base do sucesso da técnica de identificação proposta não está apenas em algoritmos isolados como o FKE, estimador VI e o FK, e sim, na cooperação mútua dos três. Essa cooperação tem como princípio o compartilhamento de informações internas, como o estimador VI compartilhando da matriz de variação, M_k , para os demais filtros (FKE, FK) fazendo com que ambos os algoritmos convirjam juntos e todos os núcleos recebam os dados brutos de saída do sistema. O triunfo dessa cooperação confirmou a terceira hipótese, tendo como resultado o desenvolvimento do LQG adaptativo associado ao FKEVI+KF.

Para trabalhos futuros, pretende-se utilizar simulações Monte Carlos para avaliar o desempenho do FKEVI+KF nos mais diversos cenários de ruídos e condições iniciais, estabelecendo de maneira mais clara as suas limitações e qualidade. Outra ideia de trabalho futuro é desenvolver a cooperação presente no algoritmo de estimação conjunta proposto para abranger o controlador. Dessa forma, o controlador será sintonizado e calculará a sua ordem de controle não só usando os parâmetros e estados estimados, mas também as incertezas dessas variáveis, moldando o seu comportamento em malha fechada conforme a confiança nos dados estimados pelo processo de identificação paramétrica, assim expandindo a cooperação presente no estimador conjunto para o sistema de controle. Outra aplicação está no uso de controladores que sejam sintonizados utilizando a parte estocástica do sistema, como o GMVSS, dessa forma, extraindo uma maior funcionalidade do modelo

ARMAX levantado.

Referências

- ABREU, J. A. P. de; OLIVEIRA, R. C. L. de; NETO, J. V. da F. Rocket tracking impact point prediction using α - β , standard kalman, extended, kalman, and unscented kalman filters: a comparative analysis. *Research, Society and Development*, Grupo de Pesquisa Metodologias em Ensino e Aprendizagem em Ciências de . . . , v. 9, n. 3, p. 20, 2020. Citado na página 63.
- AFONIN, A. A.; MIKHAYLIN, D. A.; SULAKOV, A. S.; MOSKALEV, A. P. The adaptive kalman filter in aircraft control and navigation systems. In: IEEE. *2020 2nd International Conference on Control Systems, Mathematical Modeling, Automation and Energy Efficiency (SUMMA)*. [S.l.], 2020. p. 121–124. Citado na página 33.
- AGUIRRE, L. A. Introdução à identificação de sistemas: técnicas lineares e não lineares: teoria e aplicação. *Editora UFMG*, v. 1, n. 3, p. 4, 2015. Citado 6 vezes nas páginas 21, 27, 29, 31, 36 e 61.
- ALAMINOS, D.; SALAS, M. B. Tourism stock prices, systemic risk and tourism growth: A kalman filter with prior update dsge-var model. In: SPRINGER. *International Conference on Artificial Intelligence and Soft Computing*. [S.l.], 2022. p. 167–181. Citado na página 33.
- CAMPBELL, M.; BRUNKE, S. Nonlinear estimation of aircraft models for on-line control customization. In: *2001 IEEE Aerospace Conference Proceedings (Cat. No.01TH8542)*. [S.l.: s.n.], 2001. v. 2, p. 2/621–2/628 vol.2. Citado na página 19.
- CASBEER DEREK B. KINGSTON, R. W. B. D. W.; MCLAIN, T. W. Cooperative forest fire surveillance using a team of small unmanned air vehicles. *International Journal of Systems Science*, Taylor Francis, v. 37, n. 6, p. 351–360, 2006. Disponível em: <<https://doi.org/10.1080/00207720500438480>>. Citado na página 16.
- CHIKASHA, P. N.; DUBE, C. Adaptive model predictive control of a quadrotor. *IFAC-PapersOnLine*, Elsevier, v. 50, n. 2, p. 157–162, 2017. Citado na página 20.
- COELHO, A. A. R.; COELHO, L. dos S. *Identificação de sistemas dinâmicos lineares*. [S.l.: s.n.], 2015. Citado 4 vezes nas páginas 23, 30, 55 e 58.
- DHAYBI, M.; DAHER, N. Real-time estimation of the inertia tensor elements of a quadcopter hover platform. In: IEEE. *2019 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*. [S.l.], 2019. p. 1347–1352. Citado na página 19.
- DOYLE, J. Analysis of feedback systems with structured uncertainties. In: INSTITUTION OF ELECTRICAL ENGINEERS. *IEE Proceedings D Control Theory and Applications*. [S.l.], 1982. v. 129, n. 6, p. 242–250. Citado na página 68.
- ELKHATEM, A. S.; ENGIN, S. N. Robust lqr and lqr-pi control strategies based on adaptive weighting matrix selection for a uav position and attitude tracking control. *Alexandria Engineering Journal*, Elsevier, v. 61, n. 8, p. 6275–6292, 2022. Citado 2 vezes nas páginas 16 e 66.

- GRAFF, J. *The ps-drone-api: Programming a parrot ar. Drone 2.0 with python-the easy way*. 2014. Citado na página 51.
- HANNA, S. R.; HEINOLD, D. W. *Development and application of a simple method for evaluating air quality models*. [S.l.]: American Petroleum Institute, 1985. Citado na página 45.
- HESAR, H. D.; MOHEBBI, M. An adaptive kalman filter bank for ecg denoising. *IEEE journal of biomedical and health informatics*, IEEE, v. 25, n. 1, p. 13–21, 2020. Citado na página 33.
- ILIFF, K. Parameter estimation for flight vehicles. *Journal of Guidance, Control and Dynamics*, p. 609—622, 1986. Citado na página 19.
- JATEGAONKAR, R.; PLAETSCHKE, E. Algorithms for aircraft parameter estimation accounting for process and measurement noise. *Journal of Aircraft*, v. 26, p. 360–372, 1989. Citado na página 19.
- JAZWINSKI, A. H. *Stochastic processes and filtering theory*. [S.l.]: American Press, 1970. Citado na página 36.
- KABA, A.; KIYAK, E. Optimizing a kalman filter with an evolutionary algorithm for nonlinear quadrotor attitude dynamics. *Journal of Computational Science*, Elsevier, v. 39, p. 101051, 2020. Citado na página 20.
- KALLAPUR, A. G.; ALI, S. S.; ANAVATTI, S. G. *Application of Extended Kalman Filter Towards UAV Identification*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007. 199–207 p. Citado na página 19.
- KALMAN, R. E. A new approach to linear filtering and prediction problems. 1960. Citado na página 33.
- LV, T.; YANG, Y.; CHAI, L.; LI, Q. EKF-based model predictive control for a quadrotor subject to wind disturbances and measurement noise. In: IEEE. *2019 WRC Symposium on Advanced Robotics and Automation (WRC SARA)*. [S.l.], 2019. p. 260–265. Citado na página 20.
- MCFARLAND, B.; WONG, M. The family dynamics of 802.11: The 802.11 family of standards is helping to move wireless lans into promising new territory. *Queue*, ACM New York, NY, USA, v. 1, n. 3, p. 28–38, 2003. Citado na página 50.
- MUNGUÍA, R.; URZUA, S.; GRAU, A. EKF-based parameter identification of multi-rotor unmanned aerial vehicles models. *Sensors*, MDPI, v. 19, n. 19, p. 4174, 2019. Citado 2 vezes nas páginas 16 e 20.
- NOGUEIRA, C. E.; SILVEIRA, A. S.; NOGUEIRA, P. V.; SODRE, L. d. C.; PARA, P. do; PARA, P. Federal do. A linear identification method applied to a quadcopter drone for stochastic control algorithms. 2023. Citado na página 19.
- OGATA, K. *Sistemas de control en tiempo discreto*. [S.l.]: Pearson educación, 1996. Citado 2 vezes nas páginas 15 e 43.
- PARROT. *Camera Drones and Applications / Official Parrot site*. 2021. URL <https://www.parrot.com/en/>. Citado na página 50.

- SANTANA, L. V.; BRANDÃO, A. S.; SARCINELLI-FILHO, M. Navigation and cooperative control using the ar. drone quadrotor. *Journal of Intelligent & Robotic Systems*, Springer, v. 84, p. 327–350, 2016. Citado na página 53.
- SHRAIM, H.; AWADA, A.; YOUNESS, R. A survey on quadrotors: Configurations, modeling and identification, control, collision avoidance, fault diagnosis and tolerant control. *IEEE Aerospace and Electronic Systems Magazine*, IEEE, v. 33, n. 7, p. 14–33, 2018. Citado na página 18.
- SILVEIRA, A.; SILVA, A.; COELHO, A.; REAL, J.; SILVA, O. Design and real-time implementation of a wireless autopilot using multivariable predictive generalized minimum variance control in the state-space. *Aerospace Science and Technology*, Elsevier, v. 105, p. 106053, 2020. Citado 2 vezes nas páginas 44 e 53.
- SORENSEN, H. W. Least-squares estimation: from gauss to kalman. *IEEE spectrum*, IEEE, v. 7, n. 7, p. 63–68, 1970. Citado na página 25.
- STEVENS, B. L.; LEWIS, F. L.; JOHNSON, E. N. *Aircraft control and simulation: dynamics, controls design, and autonomous systems*. [S.l.]: John Wiley & Sons, 2015. Citado na página 52.
- SVACHA, J.; PAULOS, J.; LOIANNO, G.; KUMAR, V. Imu-based inertia estimation for a quadrotor using newton-euler dynamics. *IEEE Robotics and Automation Letters*, IEEE, v. 5, n. 3, p. 3861–3867, 2020. Citado na página 20.
- WANG, Z.; WEN, M.; DANG, S.; YU, L.; WANG, Y. Trajectory design and resource allocation for uav energy minimization in a rotary-wing uav-enabled wpcn. *Alexandria Engineering Journal*, v. 60, n. 1, p. 1787–1796, 2021. ISSN 1110-0168. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S1110016820306098>>. Citado na página 16.
- XIONG, R.; HE, H.; SUN, F.; ZHAO, K. Evaluation on state of charge estimation of batteries with adaptive extended kalman filter by experiment approach. *IEEE Transactions on Vehicular Technology*, IEEE, v. 62, n. 1, p. 108–117, 2012. Citado na página 61.
- XU, Y.; HU, M.; ZHOU, A.; LI, Y.; LI, S.; FU, C.; GONG, C. State of charge estimation for lithium-ion batteries based on adaptive dual kalman filter. *Applied Mathematical Modelling*, Elsevier, v. 77, p. 1255–1272, 2020. Citado na página 42.
- ZIANE, M. A.; PERA, M.-C.; JOIN, C.; BENNE, M.; CHABRIAT, J.-P.; STEINER, N. Y.; DAMOUR, C. On-line implementation of model free controller for oxygen stoichiometry and pressure difference control of polymer electrolyte fuel cell. *International Journal of Hydrogen Energy*, Elsevier, v. 47, n. 90, p. 38311–38326, 2022. Citado na página 46.

ANEXO A . Programa em Python 3 Aplicado para Extrair e Enviar Dados Experimentais do AR.Drone 2.0.

```

1 import time, sys
2 import ps_drone3 as ps_drone
3 import math
4 import numpy as np
5 import prbs
6
7 #####MAIN_Program#####
8
9 def modeloNavDControl():
10     drone = ps_drone.Drone()
11     #####MAIN_Program#####
12
13     Ts=1/15
14
15     #####VARIaveis_Temporais#####
16     Ts=1/15
17     T=1000
18     N=round(T/Ts)
19
20
21     #####Variaveis do experimento#####
22     X=(np.zeros(8*N).reshape((8,N)))
23     U=(np.zeros(4*N).reshape((4,N)))
24
25     #i=prbs(0.2,10,N,Ts)
26
27     U[0,:]=prbs(0.1,9,N,Ts)
28     U[1,:]=prbs(0.1,10,N,Ts)
29     U[2,:]=prbs(0.1,11,N,Ts)
30     U[3,:]=0*prbs(0.1,7,N,Ts)
31
32     d=1
33
34     Ex_time=False
35
36     end=False                #Initials the PS-Drone-API
37
38     iniciar(drone)           # rotina de inicia o

```



```

39
40
41     NDC=drone.NavDataCount
42
43
44     while (not end and not Ex_time):
45         #while (not end or for k in range(d ,N ,1))
46             for k in range(d ,N ,1):
47                 while drone.NavDataCount==NDC: time.sleep(Ts) #Wait for
NavData
48
49                 NDC=drone.NavDataCount
50
51
52                 statesDrone=[drone.NavData["demo"][2][1], drone.NavData["demo
"] [2] [0], drone.NavData["demo"][2][2], drone.NavData["demo"][4][0],
drone.NavData["demo"][4][1], drone.NavData["demo"][2][2]/Ts, drone.
NavData["demo"][3], drone.NavData["demo"][4][2]]
53
54                 #print(statesDrone[6])
55
56                 X[:,k]=np.array(Conversion(statesDrone),float)
57                 X[5,k]=(X[4,k]-X[4,k-1])/Ts
58
59                 #drone.hover()
60
61                 key = drone.getKey()
62
63                 if key == " ":
64                     break
65
66                 elif key == "0": drone.hover()
67
68                 elif key == "w":
69                     drone.move(0.0,0.2,0.0,0.0)
70                     U[1,k]=0.2
71                     U[0,k]=0.0
72                     U[2,k]=0.0
73                     U[3,k]=0.0
74                 elif key == "s":
75                     drone.move(0.0,-0.2,0.0,0.0)
76                     U[1,k]=-0.2
77                     U[0,k]=0.0
78                     U[2,k]=0.0
79                     U[3,k]=0.0
80
81                 elif key == "a":

```

```
82         drone.move(-0.2,0.0,0.0,0.0)
83         U[0,k]=-0.2
84         U[2,k]=0.0
85         U[1,k]=0.0
86         U[3,k]=0.0
87
88     elif key == "d":
89         drone.move(0.2,0.0,0.0,0.0)
90         U[0,k]=0.2
91         U[2,k]=0.0
92         U[1,k]=0.0
93         U[3,k]=0.0
94
95     elif key == "q":
96         drone.move(0.0,0.0,0.0,-0.2)
97         U[2,k]=0.0
98         U[0,k]=0.0
99         U[1,k]=0.0
100        U[3,k]=-0.2
101
102     elif key == "e":
103         drone.move(0.0,0.0,0.0,0.2)
104         U[3,k]=0.2
105         U[0,k]=0.0
106         U[1,k]=0.0
107         U[2,k]=0.0
108
109     elif key == "8":
110         drone.move(0.0,0.0,0.2,0.0)
111         U[3,k]=0.0
112         U[0,k]=0.0
113         U[1,k]=0.0
114         U[2,k]=0.2
115
116
117     elif key == "2":
118         drone.move(0.0,0.0,-0.2,0.0)
119         U[3,k]=0.0
120         U[0,k]=0.0
121         U[1,k]=0.0
122         U[2,k]=-0.2
123
124     elif key == "":
125         drone.move(float(U[0,k]),float(U[1,k]),float(U[2,k]),
126 float(U[3,k]))
127
```

```

128
129         #U[:,k]=[0.0,0.0,0.0,0.0,]
130
131
132
133
134         Ex_time=True
135         print("saiu")
136
137
138     print("Acabou o Experimento")
139
140     drone.land()
141
142
143     np.save('StatesR',X)
144     np.save('ControlR',U)
145
146     np.savetxt('StatesR.txt',X)
147     np.savetxt('ControlR.txt',U)
148
149     t = np.arange(0,N*Ts,Ts)
150     np.save('tempoR',t)
151     np.savetxt('tempoR.txt',t)
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166 def iniciar(drone):
167     drone.startup()                                #Connects to the drone and
168     starts subprocesses
169
170     drone.reset()                                  #Sets drones LEDs to
171     green when red
172     while (drone.getBattery()[0]==-1): time.sleep(0.1) #Reset completed

```

```

?
173
174     print("Battery:"+str(drone.getBattery()[0])+"% "+str(drone.
175     getBattery()[1]))
176     if drone.getBattery()[1]=="empty": sys.exit()
177
178
179     drone.useDemoMode(False)           #Give me everything...fast
180     drone.getNDpackage(["demo","altitude"]) #Packets, which shall
181     time.sleep(0.5)                     #Give it some time to fully
182     awake
183
184     drone.trim()
185
186     time.sleep(1)
187     drone.takeoff()
188     time.sleep(0.5)
189     drone.hover()
190     time.sleep(2)
191     drone.mtrim()
192     time.sleep(2)
193
194
195
196 def Conversion(statesDrone):
197     statesDrone=np.array(statesDrone)
198     ##### Angle Conversion (degree to rad) #####
199     statesAngle=math.pi/180*np.array(statesDrone[0:3],float)
200
201     ##### Velocity Conversion (mm/s to m/s) #####
202     statesVelocity=0.001*np.array([statesDrone[3],statesDrone[4],
203     statesDrone[7]],float)
204
205     ##### Distance Conversion (mm to m)#####
206     altitude=0.01*statesDrone[6]
207
208     ##### Angle speed (d/s to rad/s)#####
209     phi_speed=(math.pi/180)*statesDrone[5]
210
211     States=[statesAngle[0],statesAngle[1],statesVelocity[0],
212     statesVelocity[1],statesAngle[2],phi_speed,altitude,statesVelocity
213     [2]]
214
215     return States

```

```

213
214
215
216 def prbs(K,N,nit,Ts):
217     fmin = 1/((2**N-1)*Ts)
218     fmax=1/(3*Ts)
219     print(fmin)
220     print(fmax)
221     import numpy as np
222
223     U=np.ones(N,int)
224     if N==2 :
225         pos=np.array([1, 2],int)
226     elif N==3:
227         pos=np.array([2, 3],int)
228     elif N==4:
229         pos=np.array([3, 4],int)
230     elif N==5:
231         pos=np.array([3, 5],int)
232     elif N==6:
233         pos=np.array([5, 6],int)
234     elif N==7:
235         pos=np.array([4, 7],int)
236     elif N==8:
237         pos=np.array([2, 3,4,8],int)
238     elif N==9:
239         pos=np.array([5, 9],int)
240     elif N==10:
241         pos=np.array([7, 10],int)
242     elif N==11:
243         pos=np.array([9, 11],int)
244     else:
245         print("numero de celulas invalido")
246     pos[0]=pos[0]-1
247     pos[1]=pos[1]-1
248
249     prbsi=np.zeros(nit,float)
250     for i in range(0,nit,1):
251         aux=np.array([U[pos[0]]^U[pos[1]]],int)
252         #U=U[aux,U[0:N-1]]
253         U=np.concatenate((aux,U[0:N-1]),axis=0)
254
255         if U[N-1]==1:
256             prbsi[i]=K
257         else:
258             prbsi[i]=-K
259     return prbsi

```

```
260
261
262
263
264 if (__name__ == "__main__"):
265     modeloNavDControl()
```

ANEXO B . Cálculo da Curva de Resposta em Frequência de Sistemas Lineares no Espaço de Estados

```

1 import numpy as np
2 import matplotlib.pyplot as plt
3 import control.matlab as control
4
5
6 A=A_R #Matriz A do Sistema
7 B=B_R #Matriz B do Sistema
8 C=C_R #Matriz C do Sistema
9
10 N=50000 # Numero de frequencias
11
12
13 Ts=1/15 # Tempo de amostragem
14
15 Zer=np.zeros((3,3,N-1),float)
16 oneee=np.ones((1,N-1),float)
17 Zer[0,0,:]=oneee
18 Zer[1,1,:]=oneee
19 Zer[2,2,:]=oneee
20
21 SS=control.StateSpace(A,B,C,0,Ts)
22 w=2*np.pi*np.linspace(0.0001,(1/Ts)/2,N)
23 w1=w[0:N-1]
24
25 m, p, o = SS.freqresp(w1)
26 sjw = (m*np.exp(1j*p)).transpose(2, 0, 1)
27 sv = np.linalg.svd(sjw, compute_uv=False)
28
29 #plt.semilogx(w1, 20*np.log10(sv[:, 0]), label=r'\sigma_1(S_1)',c='b')
30 plt.semilogx(w1, 20*np.log10(sv[:, 1]), label=r'\sigma_2(S_1)',c='b')
31 plt.semilogx(w1, 20*np.log10(sv[:, 2]), label=r'\sigma_2(S_1)',c='b')
32 plt.grid(True, which="both", ls="--")
33 plt.xlabel("Frequencia (rad/s)")
34 plt.ylabel("Ganho (dB)")
35 plt.show()

```