

Felipe Augusto Braga Viana

**Aumento da confiabilidade de Air Data Systems
através do uso de sensores de fluxo térmico e
votação de sinais baseada em detecção de
novidades**

Belo Horizonte - MG, Brasil

18/12/2018

Felipe Augusto Braga Viana

**Aumento da confiabilidade de Air Data Systems através
do uso de sensores de fluxo térmico e votação de sinais
baseada em detecção de novidades**

Dissertação de mestrado submetida à Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação em Engenharia Elétrica da Escola de Engenharia da Universidade Federal de Minas Gerais, como requisito para obtenção do grau de Mestre em Engenharia Elétrica

Universidade Federal de Minas Gerais – UFMG

Escola de Engenharia

Programa de Pós-Graduação em Engenharia Elétrica

Orientador: Frank Sill Torres

Belo Horizonte - MG, Brasil

18/12/2018

Felipe Augusto Braga Viana

Aumento da confiabilidade de Air Data Systems através do uso de sensores de fluxo térmico e votação de sinais baseada em detecção de novidades/ Felipe Augusto Braga Viana. – Belo Horizonte - MG, Brasil, 18/12/2018-

96 p. : il. (algumas color.) ; 30 cm.

Orientador: Frank Sill Torres

Dissertação (Mestrado) – Universidade Federal de Minas Gerais – UFMG
Escola de Engenharia

Programa de Pós-Graduação em Engenharia Elétrica, 18/12/2018.

1. Air Data System. 2. Robustez. 2. Confiabilidade. 3. Votação de sinais redundantes. 4. Sensor de fluxo térmico. 5. Detecção de novidade 6. Reconhecimento de padrões. 7. Aprendizado de máquina. I. Frank Sill Torres. II. Universidade Federal de Minas Gerais. III. Programa de Pós Graduação em Engenharia Elétrica. IV. Aumento da confiabilidade de Air Data Systems através do uso de sensores de fluxo térmico e votação de sinais baseada em detecção de novidades.

**"Aumento da Confiabilidade de Air Data Systems Através do
Uso de Sensores de Fluxo Térmico e Votação de Sinais
Baseada em Detecção de Novidades"**

Felipe Augusto Braga Viana

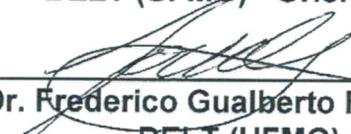
Dissertação de Mestrado submetida à Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação em Engenharia Elétrica da Escola de Engenharia da Universidade Federal de Minas Gerais, como requisito para obtenção do grau de Mestre em Engenharia Elétrica.

Aprovada em 18 de dezembro de 2018.

Por:



Prof. Dr. Frank Sill Torres
DELT (UFMG) - Orientador



Prof. Dr. Frederico Gualberto Ferreira Coelho
DELT (UFMG)



Dr. Celso Yukio Nakashima
Engineering & Technology Center (Embraer)

Dedico este trabalho à minha mãe Regina Márcia Braga Viana (in memorian) e ao meu pai Maurício Antônio Viana, que sempre me carregaram nos ombros para que eu pudesse alcançar tudo que quisesse, ao meu avô José Aniceto Viana (in memorian), fonte de inspiração de soluções criativas para problemas complexos, à minha melhor amiga e companheira Mariana Tormin, que acreditou em mim quando eu mesmo não conseguia, à minha irmã Marcela Carolina, que sempre me incentivou, e aos meus amigos, sempre fieis apesar da reclusão social.

Agradecimentos

Agradeço a UFMG e ao PPGEE, que são exemplos de ensino público de qualidade e me permitiram não só realizar esse trabalho, mas forneceram todos os degraus que subi em minha vida acadêmica desde o ensino médio realizado no colégio técnico e passando pela graduação na engenharia elétrica.

Ao meu orientador Frank Sill, que foi uma peça fundamental em todo o processo de desenvolvimento dessa dissertação e não deixou a distância continental atrapalhar um detalhe sequer de nossa comunicação.

À CAPES, que propicia o financiamento adequado ao desenvolvimento de atividades de pós graduação no PPGEE. O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001.

À Embraer, que incentivou o meu desenvolvimento pessoal e forneceu os dados necessários para essa pesquisa.

À toda sociedade brasileira, que muitas vezes sem saber financia o desenvolvimento de pesquisa de qualidade por todo o território nacional. Que o presente trabalho possa em algum momento contribuir com a sociedade e devolver esse enorme favor.

"Se consegui ver mais longe é porque estava nos ombros de gigantes"
(Isaac Newton)

Resumo

A medição de parâmetros e velocidade do ar em aplicações aeronáuticas tem alta criticidade pois falhas nos sistemas responsáveis (Air Data Systems - ADS) por tais parâmetros pode causar a morte de centenas de pessoas, como foi o caso do voo Air France 447 com 228 fatalidades.

Para suprir a demanda de segurança necessária para aplicações aeronáuticas o presente trabalho busca aumentar a confiabilidade de Air Data Systems através da exploração de duas vertentes: a utilização de sensores de fluxo térmico para medir velocidade do ar (airspeed) e o desenvolvimento de um novo votador de sinais redundantes para ser utilizado no ADS.

A utilização de sensores de fluxo térmico é explorada nesse trabalho através da integração do modelo de tais sensores em um modelo simplificado de ADS com redundância quádrupla seguida pela comparação de várias técnicas estabelecidas de votação simples. A nova proposta de votação de sinais presente aqui utiliza detecção de novidades baseada em reconhecimento de padrões e aprendizado de máquina e o desempenho desse votador proposto é comparado com um votador padrão em simulações de um ADS utilizando dados reais de entrada.

Os resultados indicam que para falhas características de sensores de fluxo térmico o ADS tem o melhor desempenho ao utilizar um algoritmo de votação de sinais simples baseado no cálculo da saída como média das entradas redundantes mais próximas que respeitam um intervalo definido.

Em relação ao votador proposto com detecção de novidades os resultados indicam que ele possui desempenho melhor que o votador padrão para casos em que a falha tem comportamento constante, crescente ou relacionado à derivada do valor medido. Para casos em que a falha tem comportamento proporcional ao valor medido o votador proposto tem desempenho inferior.

Os resultados obtidos são um passo em direção à melhoria da confiabilidade de Air Data Systems e mostram aplicabilidade de reconhecimento de padrões para identificação de falhas, além de abrir caminho para outras pesquisas relacionadas ao tema.

Palavras-chave: Air Data System. Air Data Computer. Votação. Robustez. Confiabilidade. Redundância. Aquisição de dados. Airspeed. Sensor de fluxo térmico. Detecção de novidade. Reconhecimento de padrões. Aprendizado de máquina.

Abstract

The measurement of air parameters and airspeed in aeronautical applications has a high criticality because failures in the Air Data Systems (ADS) for such parameters can cause the death of hundreds of people, as was the case of Air France flight 447 with 228 fatalities.

To meet this demand, the present work seeks to increase the reliability of Air Data Systems by exploring two aspects: the use of thermal flow sensors to measure airspeed and the development of a new redundant signal voter to be used in the ADS.

The use of thermal flow sensors is studied by integrating the model of such sensors into a simplified ADS model with 4-way redundancy and comparing several established simple voting techniques. The new signal voting proposal uses novelty detection based on pattern recognition and machine learning. The performance of the proposed voter is compared to a standard voter in simulations of an ADS using actual input data.

The results indicate that for thermal flow sensors's characteristic failures the ADS performs best when using a simple algorithm based on the calculation of the output as the average of the nearest redundant inputs that respect a defined interval.

In relation to the proposed voter with novelty detection the results indicate that it performs better than the standard voter for cases in which the failure is constant, increasing or related to the derivative of the measured value. For cases in which the failure has behavior proportional to the measured value the proposed voter has inferior performance.

The results obtained are a step toward improving the reliability of Air Data Systems and show applicability of pattern recognition for fault identification, as well as pave the way for other related research.

Keywords: Air Data System. Air Data Computer. Voting. Robustness. Reliability. Redundancy. Data Acquisition. Airspeed. Thermoflow sensor. Novelty detection. Pattern Recognition. Machine Learning.

Lista de ilustrações

Figura 1 – Representação de um ADS com 6 sensores inteligentes e um computador de processamento (UTC, 2015).	27
Figura 2 – Vista lateral de um tubo de <i>pitot</i> com as aberturas para medir pressão (WIKI, 2007). A abertura central mede a pressão total e as aberturas laterais medem a pressão estática.	29
Figura 3 – Componentes do sensor de fluxo térmico (MAKINWA; HUIJSING, 2001).	31
Figura 4 – Vista lateral do sensor de fluxo térmico (MAKINWA; HUIJSING, 2001).	33
Figura 5 – Comparação das votações simples aplicadas a quatro sinais.	35
Figura 6 – Representação em blocos da metodologia do trabalho.	40
Figura 7 – Representação em blocos da metodologia de integração do ADS.	41
Figura 8 – Modelo Simulink do sensor de fluxo térmico.	48
Figura 9 – Modelo Simulink de cálculo das entradas do sensor.	49
Figura 10 – Modelo Simulink com cálculo da entrada e sensor de fluxo conectados.	50
Figura 11 – Modelo Simulink de seleção de votadores.	51
Figura 12 – Modelo Simulink de seleção de saída dos votadores.	51
Figura 13 – Modelo Simulink de teste do votador padrão.	52
Figura 14 – Modelo Simulink do votador com detecção de novidades.	54
Figura 15 – Primeiro estágio interno do votador com detecção de novidades.	55
Figura 16 – Segundo estágio interno do votador com detecção de novidades.	55
Figura 17 – Terceiro estágio interno do votador com detecção de novidades.	56
Figura 18 – Modelo Simulink da comparação ente valor votado e referência.	57
Figura 19 – Modelo Simulink do Air Data System.	57
Figura 20 – Erro médio para diferentes quantidades de sensores com falha.	66
Figura 21 – Erro médio para diferentes tipos de entradas.	67
Figura 22 – Erro médio para diferentes tipos de falhas.	68
Figura 23 – Erro médio para diferentes frequências de entrada.	69
Figura 24 – Erro médio para todos os casos sem outliers	69
Figura 25 – Erro máximo para todos os casos sem outliers	70
Figura 26 – Curva de teste normal.	71
Figura 27 – Curva de teste com novidade.	71
Figura 28 – Erro médio para diferentes quantidades de falhas.	73
Figura 29 – Erro máximo para diferentes quantidades de falhas.	74
Figura 30 – Erro médio para diferentes tipos de falhas.	74
Figura 31 – Erro máximo para diferentes tipos de falhas.	75
Figura 32 – Erro médio para diferentes tipos de entradas.	76
Figura 33 – Erro máximo para diferentes tipos de entradas.	76

Figura 34 – Valor dos sensores e saídas do votador 1 para um caso de <i>split</i>	78
Figura 35 – Valor das saídas do votador 6 para um caso de <i>split</i>	79
Figura 36 – Valor dos sensores e saídas do votador 6 para um caso de erro proporcional.	80

Lista de tabelas

Tabela 1 – Parâmetros da utilização do votador padrão.	53
Tabela 2 – Parâmetros do ADS integrado.	58
Tabela 3 – Parâmetros da entrada para comparação de votadores simples	59
Tabela 4 – Parâmetros da falha para comparação de votadores simples	59
Tabela 5 – Parâmetros da entrada para comparação de votadores complexos	62
Tabela 6 – Parâmetros da falha para comparação de votadores complexos	62
Tabela 7 – Resultados da prova de conceito do detector de novidades.	71
Tabela 8 – Tempo de execução dos votadores complexos.	83

Lista de abreviaturas e siglas

ADS	Air Data System
TAS	True Air Speed
CAS	Calibrated Air Speed
ANAC	Agência Nacional de Aviação Civil
RBAC	Regulamento Brasileiro de Aviação Civil
BEA	Bureau d'Enquêtes et d'Analyses
ADC	Air Data Computer
CTI	Calculate Thermoflow Inputs
TF	Thermoflow

Lista de símbolos

Δ	Intervalo máximo para avaliar e calcular média da maioria
γ	Razão do calor específico do ar
q_e	Pressão estática da atmosfera
ρ	Densidade do ar
q_c	Pressão dinâmica de deslocamento de um corpo no ar
V_{TAS}	Velocidade do ar medida em um tubo de pitot
P_0	Pressão estática da atmosfera padrão ao nível do mar
ρ_0	Densidade padrão do ar ao nível do mar
ϕ	Direção do fluxo medida no sensor de fluxo térmico
U	Velocidade do fluxo medida no sensor de fluxo térmico
a_{ns} e a_{ew}	Diferenças devido a assimetria térmica
S_{ns} e S_{ew}	Fatores de proporcionalidade
ε_{ns} e ε_{ew}	Diferenças de fase em relação a referência
δP_{ns} e δP_{ew}	Diferenças entre as potências fornecidas para os aquecedores
α	Nível de significância
δ	Intervalo de mínima significância

Sumário

1	INTRODUÇÃO	23
2	REFERENCIAIS TEÓRICOS	27
2.1	Air Data Systems (ADS)	27
2.2	Tubo de pitot	28
2.3	Sensor de fluxo térmico	31
2.4	Técnicas de votação	33
2.5	Deteção de novidades utilizando reconhecimento de padrões	36
3	METODOLOGIA	39
3.1	Análise de técnicas de votação simples	40
3.2	Prova de conceito do detector de novidades	42
3.3	Desenvolvimento de votador complexo com deteção de novidades	43
3.4	Análise de votadores complexos	45
4	IMPLEMENTAÇÃO	47
4.1	Modelos	47
4.1.1	Sensor de fluxo térmico	47
4.1.2	Cálculo das entradas dos sensores	48
4.1.3	Sistema com quatro sensores de fluxo térmico	49
4.1.4	Votadores	50
4.1.4.1	Votador Padrão	52
4.1.4.2	Votador com detector de novidades	53
4.1.5	Comparação	56
4.1.6	ADS integrado	56
4.2	Execução	58
4.2.1	Análise de técnicas de votação simples	59
4.2.2	Prova de conceito do detector de novidades	60
4.2.3	Análise de votadores complexos	61
5	RESULTADOS	65
5.1	Análise de técnicas de votação simples	65
5.2	Prova de conceito do detector de novidades	71
5.3	Análise de votadores complexos	72
5.3.1	Pré análise dos resultados	72
5.3.2	Análise de duas falhas <i>split</i>	77

5.3.3	Análise do erro proporcional no detector de novidades	79
5.3.4	Análise estatística	81
5.3.5	Análise dos tempos de execução	82
6	CONCLUSÃO	85
	REFERÊNCIAS	89
	ANEXOS	91
	ANEXO A – DETALHES DA EXECUÇÃO	93
A.1	Prova de conceito do detector de novidades	93
A.2	Simulação do <i>ADS</i>	94
A.2.1	Técnicas de votação simples	95
A.2.2	Técnicas de votação complexa	95
A.2.3	Análise de resultados	96

1 Introdução

O movimento de uma aeronave é definido pelo conjunto de características do ar na região em que ela se encontra em conjunto com sua forma geométrica, posição e velocidade em relação ao fluxo. Com estas informações é possível calcular as forças e momentos aerodinâmicos gerados no avião em questão (NELSON, 1989). Medir essas propriedades é essencial e crítico para informar o piloto na tomada de decisões e para utilizar tais medidas como entradas de algoritmos de controle nos sistemas da aeronave. Como aviões podem carregar centenas de pessoas por milhares de quilômetros, falhas podem ser catastróficas e, portanto, essas medidas devem ser feitas considerando o alto grau de confiabilidade necessário para essa operação.

"Air Data Systems" (ADS) são sistemas responsáveis por medir e processar informação do fluxo de ar no entorno da aeronave, provendo essa informação para outros sistemas. Com esta integração entre os sistemas, malhas de controle complexas são construídas utilizando as medidas realizadas pelo ADS em um avião. O ADS mais comum utilizado em aeronaves comerciais tem como transdutor primário um tubo de *pitot*, responsável por medir pressão. Esse equipamento é descrito em detalhe na seção 2.2.

A principal desvantagem do tubo de *pitot* é a possibilidade do sensor congelar em situações específicas, invalidando completamente suas medidas. Isso aconteceu, por exemplo, no acidente do voo 447 da Air France em 2009 que resultou em 228 fatalidades. Nesse acidente as investigações concluíram que a aeronave caiu após inconsistências entre as medidas de airspeed, provavelmente devido ao congelamento dos tubos de *pitot*, o que fez o piloto automático se desativar. Após reações incorretas da tripulação a essa situação o resultado final foi a perda de sustentação da aeronave e sua queda (BEA, 2012). Esse é o acidente com mais mortes na história da Airfrance e do Airbus A330.

Outro exemplo interessante de acidente relacionado ao ADS é o voo 888T da Airways Germany, em que 7 pessoas faleceram. Apesar das investigações apontarem uma série de procedimentos incorretos, o sistema de votação da medida de ângulo de ataque da aeronave contribuiu para o resultado. O voo de teste de um Airbus A320 foi realizado após manutenção, na qual a aeronave foi lavada e água ficou alojada nos sensores de ângulo de ataque. Essa água congelou após a decolagem, fazendo com que dois dos três sensores fornecessem medidas incorretas. A votação do ângulo de ataque validou as medidas incorretas devido a estarem concordantes entre si e descartou os valores provenientes do único sensor correto. Após outras falhas de procedimento da tripulação o avião acabou caindo no mar mediterrâneo (BEA, 2008).

Devido à criticidade da medição de velocidade em uma aeronave e da existência da

possibilidade de falhas nos sistemas atuais, existe uma demanda para o estudo de maneiras de aumentar a confiabilidade destas medidas. Uma possibilidade é a utilização de sensores de velocidade com princípios de funcionamento distintos do tubo de *pitot*, dentre eles sensores de fluxo térmico. Este tipo de sensor vem ganhando atenção nos últimos anos (HAASL; STEMME, 2007).

Sensores de fluxo térmico tem como princípio de funcionamento a medição da velocidade de um fluido através da interação térmica entre esse fluido e o sensor. Apesar do princípio de funcionamento desse tipo de sensor ser conhecido, sua aplicação na aeronáutica não é usual. Desenvolvimentos recentes nesta área estão proporcionando sensores mais avançados, como feito por Makinwa e Huijsing (2001). A aplicação de tecnologias novas em aeronaves não é fácil, uma vez que a confiabilidade e robustez são essenciais nessa área e que soluções adequadas para fornecer o nível adequado de segurança precisam estar muito bem estabelecidas.

ADS utilizam de redundância para atingir o nível de segurança requerido pelas entidades certificadoras. Para tratar as medidas redundantes, algoritmos de votação e consolidação são utilizados. Assim, outra possibilidade para aumentar a confiabilidade e segurança desses sistemas é estudar melhorias nas técnicas de votação para consolidação das medidas dos sensores redundantes.

O objetivo deste trabalho é dar um passo em direção a melhoria da confiabilidade de "Air Data Systems" em aplicações aeronáuticas. Para atingir esse objetivo duas vertentes serão exploradas. A primeira é o estudo de como utilizar sensores de fluxo térmico com alta confiabilidade em um ADS. A segunda é o desenvolvimento e estudo de uma nova técnica de votação utilizando reconhecimento de padrões.

Para o estudo de confiabilidade de sensores de fluxo térmico, a exploração de como atingir esse objetivo passa pela integração do modelo de um sensor de fluxo térmico no modelo simplificado de um ADS com redundância, inserção de falhas típicas desse sensor nesse modelo e comparação do comportamento de diferentes técnicas de votação aplicadas a esse sistema. Comparando o desempenho das diferentes técnicas, é possível escolher a melhor para esse tipo de sistema e sensor.

A etapa de estudo das técnicas de votação envolve a análise de como técnicas de reconhecimento de padrões se comportam com dados reais típicos dessa aplicação e o desenvolvimento de um votador utilizando tal técnica, que em seguida tem seu desempenho comparado com uma técnica padrão da indústria.

O restante da dissertação é estruturada da seguinte maneira: O capítulo 2 introduz o estado da arte dos sistemas de medição de velocidade em aeronaves, sensores de fluxo térmico, tubos de *pitot*, técnicas de votação e detecção de novidade. Em seguida, o capítulo 3 descreve a metodologia para atingir o objetivo proposto. O capítulo 4 discute a

implementação da metodologia. O capítulo 5 mostra e analisa os resultados obtidos. Por fim, o capítulo 6 destaca as principais conclusões obtidas com este trabalho.

2 Referenciais teóricos

2.1 Air Data Systems (ADS)

Um ADS comercial típico pode ser encontrado descrito em [UTC \(2015\)](#) como um sistema que integra sensores e processamento de informações para gerar parâmetros críticos de "air data", incluindo pressão estática e dinâmica, velocidade, altitude, ângulo de ataque e deslizamento lateral ("sideslip"). A figura 1 mostra um ADS de seis sensores, cada um composto de um tubo de *pitot* e uma central de processamento. O tubo de *pitot* é o transdutor que transforma a grandeza física em sinais elétricos, que depois de digitalizados são enviados para serem consolidados pelo "Air Data Computer" (ADC). A redundância dos sensores é necessária para atingir o nível requerido de segurança para operação da aeronave. O processo de cálculo dos níveis de segurança e sua demonstração é regido por normas internacionais desenvolvidas por comitês especiais e documentos emitidos pelas autoridades de cada país ([SAE, 1996](#)) ([SAE, 2010](#)). Attingir estes níveis de segurança utilizando os processos adequados deve ser demonstrado para as autoridades regulamentadoras para que a aeronave obtenha permissão para circular naquele país ([ANAC, 2014](#)). O maior desafio em desenvolver um ADS está em conseguir projetar um sistema que é confiável, seguro, redundante, tolerante a falhas e integrado a outros sistemas, o que resulta em uma alta complexidade.

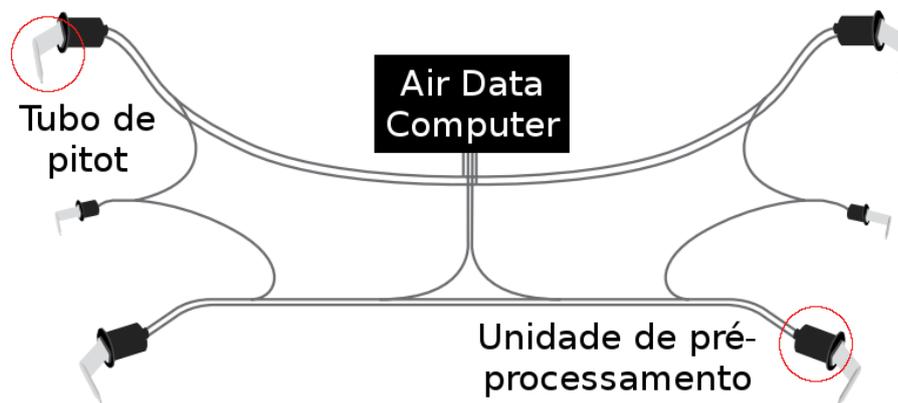


Figura 1 – Representação de um ADS com 6 sensores inteligentes e um computador de processamento ([UTC, 2015](#)).

Os diversos sinais produzidos por um ADS são tipicamente transmitidos pra outros sistemas do avião seguindo o protocolo de comunicação ARINC 429, que podem utilizar as informações para diferentes propósitos ([RADIO, 2004](#)). As informações sobre o ar fornecidas pelo ADS são algumas das mais importantes em uma aeronave. Até nos sistemas e aviões mais simples a velocidade (airspeed) é utilizada a todo momento pelo piloto em

todas as manobras críticas e navegabilidade. Em modelos mais complexos, o sistema de controle de voo pode utilizar os valores de airspeed em conjunto com outras informações de diferentes sistemas para identificar a proximidade da velocidade mínima de sustentação do avião (velocidade de stall) e emitir um alerta caso esteja muito próxima. O sistema de controle de voo também utiliza essa informação para calcular as cargas dinâmicas do avião e limitar e controlar suas manobras em voo (GRACEY, 1980).

Os requisitos de aeronavegabilidade necessários para permitir a operação de um avião são definidos pela autoridade de cada país, embora acordos de cooperação favoreçam o estabelecimento de um padrão bem semelhante entre os países. É o caso da norma do governo brasileiro para aviões de transporte denominada RBAC 25 (ANAC, 2014), que reedita um documento estrangeiro com as normas a serem seguidas.

A norma em questão não estabelece requisitos específicos para o ADS, mas para instrumentos, avisos, manobras e limites, que em sistemas modernos computadorizados dependem extensivamente dos dados provenientes do ADS. A definição dos requisitos que devem ser atingidos por um ADS em um modelo específico envolve o estudo cuidadoso da RBAC 25 e da integração dos sistemas nesse modelo, incluindo informações sobre a taxa de atualização dos dados e precisão necessários para atingir os objetivos específicos de cada sistema.

Este trabalho foca na medição de airspeed feita pelo ADS, e o requisito mais relevante para essa informação no RBAC 25 é a seção 25.1323 que especifica requisitos para o sistema de indicação de airspeed. São definidos requisitos de intervalos de medição, condições de calibração e o erro máximo, que é o mais útil para esse trabalho e definido em 3% ou 5 nós, o que for maior (ANAC, 2014).

É importante destacar que o airspeed medido é em relação a massa de ar na qual o avião se desloca e é justamente isso que o piloto precisa para manter o avião no ar da melhor maneira possível. No entanto, se essa massa de ar se desloca em relação ao solo, o piloto precisa derivar a velocidade do avião em relação ao solo utilizando outras medidas ou outros sistemas. Isso é relevante para a navegação e conseguir levar a aeronave de um ponto a outro, que também é um dos objetivos de sua operação (GRACEY, 1980).

2.2 Tubo de pitot

O tubo de *pitot* é o transdutor utilizado na maioria dos ADS para converter grandezas físicas em sinais elétricos. Ele é composto por dois tubos concêntricos mantidos a uma pequena distância da fuselagem do avião para evitar interferência dela no fluxo de ar. O tubo de dentro contém uma abertura alinhada com a direção do movimento e é responsável por medir a pressão total. A pressão total é o resultado da interação do corpo com ar ao se movimentar. O tubo externo contém aberturas perpendiculares ao movimento

e é responsável por medir a pressão estática, característica do ambiente. Na base dos tubos, transdutores convertem as pressões em sinais elétricos que então são digitalizados. A diferença entre a pressão total e pressão estática é chamada pressão dinâmica (ou pressão de impacto) e é utilizada para calcular a velocidade relativa entre o ar e o tubo de pitot. Essa velocidade é chamada de "airspeed" (NELSON, 1989). A figura 2 mostra um diagrama com as aberturas citadas para medir a pressão.



Figura 2 – Vista lateral de um tubo de *pitot* com as aberturas para medir pressão (WIKI, 2007). A abertura central mede a pressão total e as aberturas laterais medem a pressão estática.

Existem definições diferentes para os nomes e significados das velocidades que podem ser calculadas através das medidas de pressão do tubos de *pitot* dependendo de qual o contexto e aplicação em que o sensor está inserido. Neste trabalho vamos apresentar as definições feitas em Gracey (1980) e utilizadas pela NASA.

A "True Air Speed" (TAS) é definida como a velocidade real do avião em relação à massa de ar já corrigida para erros de medição. Seu cálculo considera efeitos de compressibilidade, densidade e pressão do fluido. As informações necessárias para esse cálculo são a pressão dinâmica, pressão estática e temperatura do ar. Portanto a TAS indica a velocidade real da aeronave independente da sua altitude, variações de temperatura e pressão atmosférica.

A "Calibrated Air Speed" (CAS) é a velocidade calculada utilizando somente a pressão dinâmica já corrigida para erros de medição. Seu cálculo considera efeitos de compressibilidade mas ignora variações devido a densidade e pressão do fluido. O efeito da densidade do ar e pressão estática são calibrados para o nível do mar em uma atmosfera padrão. Isso significa que a CAS indica a velocidade real da aeronave somente ao nível do mar em condições atmosféricas padrão.

Como exemplo da comparação entre TAS e CAS, imagine o cenário de um avião voando ao nível do mar em condições atmosféricas semelhantes à padrão. Nesse momento TAS e CAS tem o mesmo valor. Em seguida esse avião aumenta sua altitude em relação ao nível do mar e as condições atmosféricas se alteram. Tipicamente a pressão, temperatura e densidade do ar diminuem com a altitude. Nessa nova condição o valor de TAS é maior

do que o valor de CAS. TAS é uma medida da velocidade real do avião em relação ao ar, já CAS não mostra mais a velocidade real mas continua sendo proporcional à pressão sentida pelo avião ao se deslocar, o que também tem utilidade para o piloto.

O cálculo de TAS para valores subsônicos (menor do que a velocidade do som) é feito através da fórmula 2.1, proveniente da equação de Bernoulli para fluidos compressíveis. Nesta equação γ é a razão dos calores específicos do ar, q_e é a pressão estática, ρ é a densidade do ar e q_c é a pressão dinâmica. Para esta fórmula, o valor de γ é fixado em 1.4 como especificado em Nelson (1989).

$$V_{TAS} = \sqrt{\frac{2\gamma q_e}{(\gamma + 1)\rho} \left(\left(\frac{q_c + q_e}{q_e} \right)^{(\gamma-1)/\gamma} - 1 \right)} \quad (2.1)$$

A fórmula para CAS é a mesma, com a diferença de que os valores de p e ρ são fixos para os valores a nível do mar P_0 e ρ_0 .

O tubo de *pitot* é vastamente utilizado há décadas devido a ser simples, confiável e estar relacionado intimamente com a pressão do ar provocada pelo deslocamento do corpo no fluido, uma informação de extrema importância para o piloto. Isso significa que mesmo instrumentos simples de baixíssima complexidade, capazes de medir somente CAS, já são o suficiente para aviões pequenos ou antigos serem manobrados dentro dos requisitos necessários.

Devido às condições climáticas que uma aeronave enfrenta durante sua operação e a como os tubos de *pitot* estão dispostos na fuselagem, além de como são construídos, existem situações em que ele pode congelar. Nesse caso, o sensor não consegue medir a pressão externa, nem total e nem estática, fazendo com que suas medidas não correspondam a realidade. Em casos como esse o piloto deve perceber que a medida de velocidade está errada e ignorá-la, navegando através de outros instrumentos e pontos visuais. Fazer isso não é trivial e falhas desse tipo causaram acidentes recentes, como a já citada queda do Airfrance (BEA, 2012). Outra restrição do sensor é que ele deve estar alinhado com o fluxo do movimento, dentro de um desvio máximo permitido, para medir corretamente a velocidade. O tubo de *pitot* também interfere na aerodinâmica do avião e, portanto, a quantidade e local dos sensores deve ser cuidadosamente estudada em uma busca por redundância e performance.

Apesar das limitações e situações específicas em que pode falhar, o tubo de *pitot* tem um histórico sólido e confiável de utilização na aeronáutica. Mesmo com mais pontos positivos do que negativos esse instrumento não é perfeito e existe espaço para buscar melhorar a confiabilidade de medidas de airspeed, seja estudando sensores diferentes ou algoritmos mais robustos.

2.3 Sensor de fluxo térmico

Existe uma extensa linha de pesquisa em sensores de fluxo térmico, como podemos ver no estudo bibliográfico de [Haasl e Stemme \(2007\)](#). Todos se baseiam no princípio de medir a velocidade de um fluido usando a interação térmica entre este fluido e o sensor. Apesar do princípio comum, existem vários tipos de configuração, materiais e aplicações para sensores de fluxo térmico. Tomando como base o estudo de [Haasl e Stemme \(2007\)](#) é possível escolher um sensor que está próximo do cenário aeronáutico. Esse sensor é descrito nos artigos [Makinwa e Huijsing \(2000\)](#) e [Makinwa e Huijsing \(2001\)](#): foi desenvolvido para funcionar no ar, possui uma grande faixa de medida, capaz de medir velocidade e direção do fluxo, é recente, apresenta melhorias em relação a um sensor de fluxo térmico comercial e tem um modelo de funcionamento completo descrito em seus artigos.

O sensor de fluxo térmico descrito em [Makinwa e Huijsing \(2001\)](#) está mostrado na figura 3 e é composto por quatro sensores de temperatura (termopilhas) e quatro aquecedores (resistores) dispostos nas bordas de um chip de silício quadrado. Além disso, o componente contém um diodo no centro que mede a temperatura média do chip e a eletrônica responsável pela lógica de controle necessária para a operação do sensor.

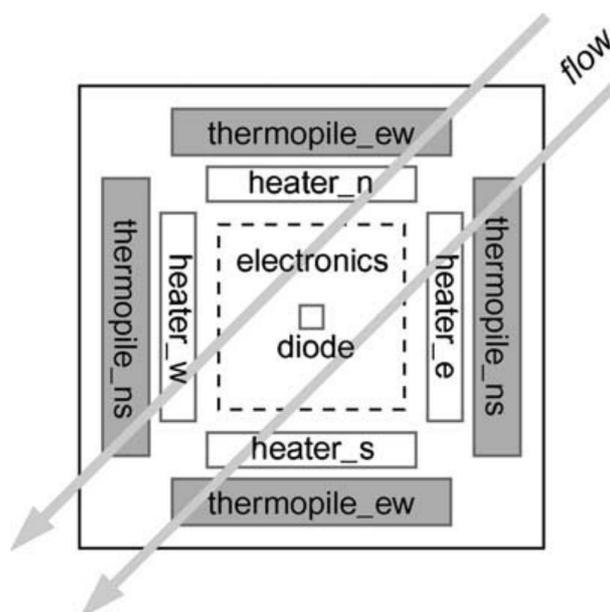


Figura 3 – Componentes do sensor de fluxo térmico ([MAKINWA; HUIJSING, 2001](#)).

O componente mostrado funciona em modo de diferença de temperatura média constante (Constant Average Temperatura Difference - CATD). Neste modo a temperatura medida pelo diodo central é comparada com uma temperatura de referência e os aquecedores são acionados para que a diferença de temperatura entre a referência e o medido pelo diodo seja constante.

Para realizar a medição da velocidade e direção do fluxo de ar que passa pelo sensor,

considera-se o efeito de resfriamento assimétrico em cada termopilha. Observando a figura 3, dado um fluxo horizontal da direita para a esquerda, as termopilhas "east-west" (ew) serão resfriadas de maneira assimétrica perdendo mais calor na extremidade da direita do que na extremidade da esquerda. Existem dois sensores "ew" espacialmente paralelos e ligados em série para que a sensibilidade da assimetria térmica seja maior. O mesmo efeito acontece quando há um fluxo vertical, em que o resfriamento assimétrico ocorre nos componentes "north-south" (ns).

Uma técnica de modulação térmica Sigma Delta é usada para manter o sensor operando em temperatura constante, o que significa que a assimetria térmica detectada pelas termopilhas funciona em malha fechada alimentando a escolha de qual aquecedor deve ser acionado de maneira a corrigir a diferença medida. Um comparador ligado à saída da termopilha cria uma série de pulsos que ligam ou desligam os aquecedores relacionados. Se o comparador detecta a extremidade "east" mais fria, então o aquecedor "e" é ligado, e vice versa.

Esta técnica de operação elimina a necessidade de um ADC para obter a medida do sistema. A saída do sensor de velocidade é o fluxo de bits que representa a potência dissipada por cada resistor. A informação necessária para calcular a velocidade é a diferença de potência entre os resistores opostos. Assim, a diferença entre a potência dos resistores "e" e "w" é chamada δP_{ew} e a diferença entre os resistores "n" e "s" de δP_{ns} . A velocidade do fluxo é proporcional à soma dos quadrados de δP_{ew} e δP_{ns} e a direção do fluxo é obtida considerando os valores de δP como os componentes ortogonais do vetor velocidade. Os valores obtidos em ensaio com este sensor têm um erro máximo de 4% para a velocidade e 4° para a direção do fluxo (MAKINWA; HUIJSING, 2001).

As equações (2.2), (2.3) e (2.4) que calculam a velocidade e direção do fluxo a partir de δP_{ew} e δP_{ns} são apresentadas em Makinwa e Huijsing (2000). Nessas equações: ϕ é a direção calculada do fluxo; U é a velocidade calculada do fluxo; a_{ns} e a_{ew} são diferenças devido a assimetria térmica; S_{ns} e S_{ew} são fatores de proporcionalidade; ε_{ns} e ε_{ew} são diferenças de fase em relação a referência, para orientação da direção; δP_{ns} e δP_{ew} são as diferenças entre as potências fornecidas para os aquecedores.

Como pode ser observado na figura 4, o sensor de fluxo térmico apresentado é protegido do contato direto com o fluxo de ar ao ser fixado em um fino disco liso de cerâmica que o separa do objeto de medição. Esta configuração é uma vantagem na aplicação aeronáutica pois o sensor pode ser colocado no local mais propício da fuselagem da aeronave sem interferir em sua aerodinâmica.

$$\phi = \tan^{-1} \left(\frac{v_{ns} \cos(\varepsilon_{ew}) - v_{ew} \sin(\varepsilon_{ns})}{v_{ns} \sin(\varepsilon_{ew}) + v_{ew} \cos(\varepsilon_{ns})} \right) \quad (2.2)$$

$$U = \frac{v_{ns}^2 + v_{ew}^2 - 2v_{ns}v_{ew} \sin(\varepsilon_{ns} - \varepsilon_{ew})}{\cos^2(\varepsilon_{ns} - \varepsilon_{ew})} \quad (2.3)$$

$$v_{ns} = \frac{\delta P_{ns} - a_{ns}}{S_{ns}} \quad v_{ew} = \frac{\delta P_{ew} - a_{ew}}{S_{ew}} \quad (2.4)$$

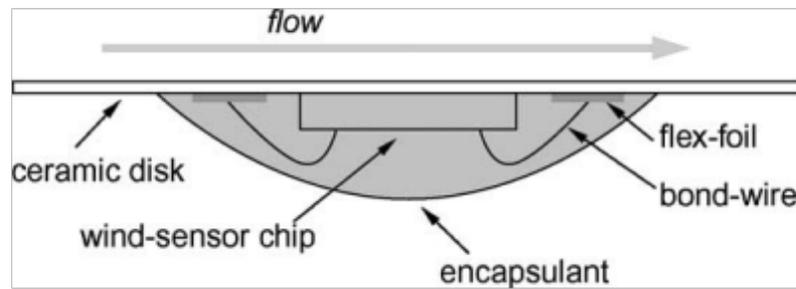


Figura 4 – Vista lateral do sensor de fluxo térmico (MAKINWA; HUIJSING, 2001).

2.4 Técnicas de votação

Técnicas de votação para consolidação de sinais redundantes são estudadas e aplicadas de maneira ampla, como pode ser observado em Askari, Nourani e Namazi (2011), em que várias técnicas são mencionadas.

Uma técnica é a realização da média simples dos sinais de entrada para compor a saída do votador. Esta é a maneira mais simples de diminuir a influência de um sinal com falhas no valor consolidado da medida, apesar de não remover sua influência. Outra técnica é o cálculo da saída como a mediana dos sinais de entrada. No caso de um número ímpar de sinais isso corresponde a escolher o sinal do meio para ser igual a saída. No caso de um número par de sinais, isso corresponde a fazer a média simples dos dois sinais do meio.

Além desses dois tipos, Askari, Nourani e Namazi (2011) propõem também a votação pela média da maioria, em que a diferença entre os sinais de entrada é comparada a um intervalo pré-definido Δ . Quando as diferenças são menores que Δ os sinais são considerados válidos e a saída é a média de todos que cumpriram o requisito. Se um sinal estiver muito diferente, a diferença dele para os outros será maior que o limiar e ele não será considerado no cálculo. Caso a distribuição dos valores dos sinais seja tal que a

diferença entre eles é menor que o limiar em mais do que um subconjunto de sinais, então aquele subconjunto que tiver a maioria dos sinais é escolhido para o cálculo. Por exemplo, para o caso de 4 sinais de entrada, pode acontecer dos sinais A, B e C cumprirem o critério do Δ e os sinais C e D também. Nesse caso o subconjunto ABC que contém a maioria dos sinais é escolhido.

O trabalho de [Alves e Torres \(2016\)](#) introduz uma variação da média da maioria, em que o valor de limiar não é utilizado. Nesta técnica, são escolhidos os dois sinais dentre todos os disponíveis que têm a menor diferença entre si, e a saída é a média deles.

As opções citadas até agora são técnicas de votação simples, em que a saída é resultado de um cálculo utilizando os valores instantâneos das entradas. Abaixo um resumo de todas elas:

$$Media = \frac{(Sinal_1 + Sinal_2 + Sinal_3 + Sinal_4)}{4} \quad (2.5)$$

$$Mediana = \frac{SinaisDoMeio(Sinal_1, Sinal_2, Sinal_3, Sinal_4)}{2} \quad (2.6)$$

$$Media da maioria = \frac{SomaNoIntervalo(Sinal_1, Sinal_2, Sinal_3, Sinal_4)}{QuantidadeNoIntervalo} \quad (2.7)$$

$$Media dos mais proximos = \frac{SomaMaisProximos(Sinal_1, Sinal_2, Sinal_3, Sinal_4)}{2} \quad (2.8)$$

Para demonstrar e comparar as quatro técnicas simples citadas a figura 5 abaixo demonstra qual seria o valor de saída da votação utilizando cada uma dessas técnicas para um sistema redundante com quatro entradas de valor fixo. As linhas cheias de diferentes cores representam as medidas de quatro entradas. As linhas tracejadas mostram o resultado para as técnicas citadas até agora. Hipoteticamente nesse caso o intervalo Δ da técnica de médias da maioria é 0,2.

É possível perceber que todas as técnicas obtêm como resultado um valor entre os dados de entrada, mas não é possível julgar qual delas tem um resultado "melhor" sem informações sobre o que está gerando os sinais. Se nenhum desses sinais tiver uma falha e essas variações estiverem dentro de flutuações normais para esse sistema teórico, todas as votações calculam sinais consolidados diferentes mas igualmente válidos.

O artigo de [Oliveira e Galvão \(2011\)](#) apresenta um algoritmo de votação mais complexo. Sua técnica se difere das outras por não ser uma fórmula simples de cálculo da saída, mas uma série de algoritmos de detecção de falhas que funcionam em paralelo,

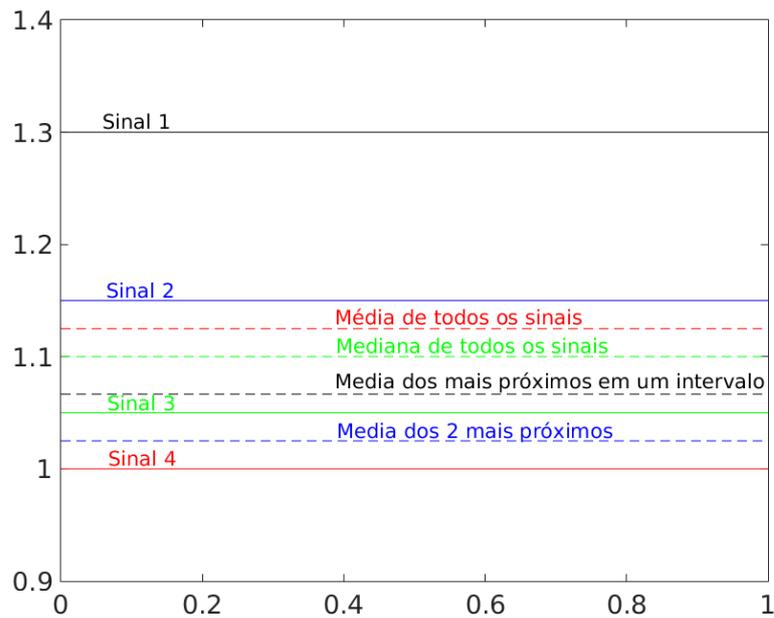


Figura 5 – Comparação das votações simples aplicadas a quatro sinais.

analisando constantemente os sinais de entrada. No caso de nenhuma falha ter sido detectada e todos os sinais de entrada serem considerados válidos, a saída é a mediana da entrada. A medida que um sinal é considerado com falha ele é removido, e a mediana é calculada com os outros sinais. As maneiras de detecção de falhas dos sinais são descritas abaixo:

- *Miscompare*: Se a diferença entre um sinal de entrada e valor votado da saída no ciclo anterior foi maior que um limiar pre-estabelecido ele é considerado com falha de *miscompare*.
- *Range*: Se o valor de um sinal de entrada estiver fora dos limites de operação pré-estabelecidos, ele é considerado com falha de "range".
- *Oscilação*: Se apenas um dos sinais estiver oscilando dentro de uma faixa de frequência preestabelecida e com uma amplitude maior do que um limiar, aquele sinal é considerado com falha de oscilação.
- *Split*: Se quatro sinais estiverem em uma configuração de dois pares com valores próximos dentro do par mas valores diferentes entre os pares, usando limiares preestabelecidos, eles são considerado em *split*. Nesse caso todos os sinais são considerados com falha.

Cada detecção de falha contém filtros para realizar análise dinâmica no tempo dos sinais e critérios de considerar um sinal válido de novo após uma falha ser detectada. Além disso a

saída é alterada de maneira amortecida a cada alteração no número de sinais considerados válidos. Por padrão, a técnica foi descrita para funcionar em um sistema de redundância de 4 entradas.

2.5 Detecção de novidades utilizando reconhecimento de padrões

Reconhecimento de padrões é uma técnica que busca identificar padrões e regularidades em dados de maneira automática. Reconhecimento de padrões está relacionado com inteligência artificial e aprendizado de máquina, mas os três não são sinônimos. Aprendizado de máquina é uma abordagem possível para implementar reconhecimento de padrões, que por sua vez é uma abordagem possível para implementar inteligência artificial (BISHOP, 2006). O reconhecimento de padrões é feito utilizando um conjunto de características ou atributos extraídos dos elementos sob análise para determinar sua categoria mais provável (ALVES; TORRES, 2016).

A detecção de novidades, por sua vez, é uma aplicação do reconhecimento de padrões, em que o objetivo é identificar padrões potencialmente diferentes do normal, o que pode indicar falhas ou defeitos. Para isso é necessário saber o que é o normal no cenário analisado e identificar quando dados se afastam desse padrão. A identificação da novidade pode se basear em diferentes princípios: probabilística, distância, domínio, reconstrução ou teoria da informação (ALVES; TORRES, 2016).

A seguir vamos detalhar a técnica utilizada no trabalho de Alves e Torres (2016). Em seu texto ele escolhe desenvolver um detector de novidades de classe única baseado em distância de mahalanobis entre modelos gaussianos multivariados, utilizando aprendizado de máquina para treinar seu detector sobre o que é o normal em sua aplicação.

A distância de mahalanobis é uma técnica de cálculo da distância entre duas amostras que mede a semelhança entre elas. Quanto menor a distância de mahalanobis, menor a diferença entre as amostras. O conceito é semelhante ao da distância euclidiana, mas a distância de mahalanobis é vastamente utilizada para detectar *outliers* em que os dados são modelados com variáveis aleatórias (TODESCHINI et al., 2013).

O modelo gaussiano multivariado é um modelo probabilístico utilizado para representar uma subpopulação. Na aplicação de Alves e Torres (2016) esse modelo é criado a partir dos dados de treinamento do detector de novidades e a distância de mahalanobis é calculada entre ele e a amostra que se deseja classificar.

O detector de novidades desenvolvido por Alves e Torres (2016) funciona da seguinte maneira. Primeiro o sistema é treinado utilizando como dados de entradas diversas curvas de aquecimento de um forno. Como resultado de treinamento são obtidos um modelo gaussiano multivariado que representa as amostras de treinamento e um valor limite.

Depois, durante a operação, o detector de novidades recebe curvas de aquecimento e calcula a distância de mahalanobis entre essa curva e o modelo gaussiano. Se essa distância for menor do que o limite, a curva é considerada normal. Se a distância for maior do que o limite, a curva é classificada como tendo novidades. O objetivo é funcionar identificando e sinalizando a ocorrência de uma falha, sem corrigir o resultado.

A técnica descrita foi escolhida para ser estudada porque é poderosa para encontrar *outliers* e o aumento da confiabilidade do ADS envolve, como já descrito na seção 2.4, detecção de falhas nos sinais redundantes. Assim, existe o potencial da aplicação da detecção de novidades para identificar tais falhas.

3 Metodologia

Como citado no capítulo 1 o objetivo deste trabalho é explorar como melhorar a confiabilidade de Air Data Systems, demanda que existe nos sistemas aeronáuticos atuais. Foram ainda citadas duas vertentes para atingir esse objetivo: estudar a integração de sensores de fluxo térmico em ADS e estudar novos algoritmos que melhorem a confiabilidade das medidas.

O estudo de como utilizar sensores de fluxo térmico com alta confiabilidade em um ADS é feito através da integração do modelo de um ADS contendo redundância com o modelo de tais sensores. Em seguida, defeitos específicos de sensores de fluxo térmico são inseridos no modelo e o comportamento do ADS é observado. Nesse cenário é possível analisar, portanto, como um ADS se comporta quando uma falha típica desse tipo de sensor ocorre. Como resultado desse estudo espera-se obter a descrição e a comparação do comportamento do ADS para diferentes tipos de votações típicas.

A exploração de um novo algoritmo de votação que possa melhorar o desempenho da consolidação de sinais em um ADS é inspirada pela técnica de detecção de novidades descrita na seção 2.5. A aplicação de uma técnica de reconhecimento de padrões em um sistema de controle aeronáutico é uma das grandes novidades desse trabalho. Para fazê-lo, primeiramente uma prova de conceito da técnica aplicada a dados reais aeronáuticos é realizada. Espera-se obter de resultado um indicativo sólido se a técnica serve para detectar novidades nesse tipo de dado ou não.

Em seguida um novo votador é implementado utilizando detecção de novidades e integrado ao mesmo ADS já criado para explorar a utilização de sensores de fluxo. Dessa vez o desempenho do ADS na presença de falhas é comparado entre o novo votador e um padrão da indústria. Como resultado é esperado que o desempenho dos dois votadores possa ser comparado. A esperança é que o desempenho do votador proposto seja superior, o que indicaria um avanço no aumento da confiabilidade desses sistemas. A figura 6 resume em um fluxo todos os passos descritos até agora da metodologia.

As próximas seções desse capítulo descrevem com mais detalhes essa metodologia. A seção 3.1 explica como é pensada a integração de sensores de fluxo no ADS, a seção 3.2 explica a prova de conceito da detecção de novidades, a seção 3.3 explica o processo de desenvolvimento e características desejadas do novo votador e por fim a seção 3.4 detalha a metodologia para comparar o desempenho do novo votador.

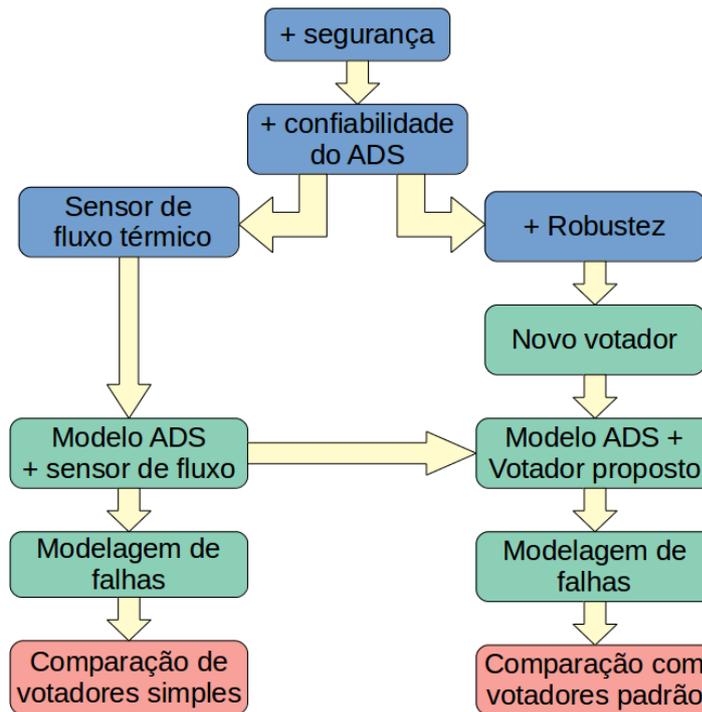


Figura 6 – Representação em blocos da metodologia do trabalho.

3.1 Análise de técnicas de votação simples

A primeira etapa do trabalho é analisar como técnicas de votação bem estabelecidas funcionam quando aplicadas a um ADS composto por sensores de velocidade de fluxo térmico. Devido a escolha de utilizar esse tipo de sensor, saber como uma falha interfere em seu funcionamento e como essa falha pode ser mitigada através de votação de sinais é relevante.

O cenário utilizado para a análise é um ADS com 4 sensores redundantes, escolhido por ser o estado da arte utilizado na indústria e no votador utilizado como padrão (OLIVEIRA; GALVÃO, 2011). Quatro técnicas de votação foram escolhidas para serem observadas: (1) Média dos sinais; (2) Mediana dos sinais; (3) Média da maioria e; (4) Média dos mais próximos; todas explicadas em detalhe do capítulo 2.4. Essas técnicas são as referências em votação de sinais de maneira simples, utilizando apenas o valor instantâneo dos sinais para calcular um valor consolidado. Essas técnicas não utilizam análises temporais ou outras técnicas de detecção de erro para selecionar os sinais a serem votados.

Um modelo do ADS contendo quatro sensores redundantes e a votação dos seus valores é criado com cada uma das técnicas descritas. Eles são simulados e comparados entre si para realização da análise. Esses modelos tem como entradas a velocidade e ângulo do fluxo de ar para qual queremos simular o sistema, parâmetros de funcionamento dos votadores e parâmetros de funcionamento dos sensores de fluxo térmico, incluindo as falhas

modeladas.

A simulação de um modelo é executada variando uma gama de parâmetros para observar o comportamento em diferentes cenários: forma do sinal de entrada; valor da entrada; frequência do sinal de entrada; tipo de relação da falha com a entrada; valor do erro associado à falha; localização e número de falhas.

O valor resultante da votação é a medida consolidada do ADS. Este valor é comparado com a entrada de referência do modelo para saber o erro e a performance do votador. É possível então comparar as performances das técnicas de votação e analisar como cada uma se comporta integrada ao ADS composto de sensores de fluxo térmico. A figura 7 mostra o método descrito até aqui para estudar a integração do sensor de fluxo térmico para um votador. Para comparar o desempenho de vários votadores esse cenário é repetido para cada um deles.

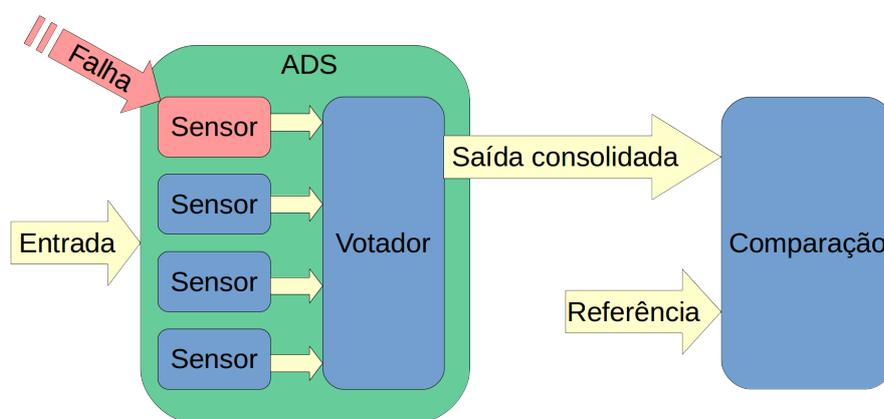


Figura 7 – Representação em blocos da metodologia de integração do ADS.

A realização das simulações do ADS respeitando os critérios colocados de análise resulta em uma comparação objetiva entre os votadores aplicados ao sensor de fluxo. As falhas inseridas são específicas desse tipo de sensor e como resultado é possível saber qual técnica de votação tem melhor resultado para aumento da sua confiabilidade e permitir sua aplicação em ambiente aeronáutico.

3.2 Prova de conceito do detector de novidades

A detecção de novidade foi escolhida para ser analisada e possivelmente utilizada em um novo método de votação devido aos bons resultados demonstrados onde foi proposta da maneira que a apresentamos aqui (ALVES; TORRES, 2016). Se tais resultados puderem ser atingidos em uma aplicação aeronáutica ela pode se provar uma técnica de detecção de erro tão ou mais poderosa do que as já utilizadas pelo votador padrão descrito por Oliveira e Galvão (2011).

Assim, a próxima etapa é o estudo de como o detector de novidades pode ser usado no contexto de melhoria do nível de confiabilidade de um ADS. Para fazer isso busca-se uma prova de conceito da técnica de Alves e Torres (2016) aplicada a dados reais de *airspeed*. Se a técnica funcionar bem para esses dados de maneira controlada, então o processo de implementar um novo votador pode ser iniciado.

Os dados de entrada desse estudo são curvas da velocidade de uma aeronave durante a sua decolagem. A etapa de decolagem foi escolhida por ser uma manobra muito frequente e característica de aeronaves. Existem diferentes perfis de curvas de decolagem que variam para cumprir os requisitos de aeronavegabilidade dependendo do modelo da aeronave, aeródromo no qual é feita a decolagem e configuração do avião naquele voo. Considerando condições ideais de manobra e um modelo fixo de veículo, suas manobras são feitas a otimizar seu funcionamento, o que torna o perfil da decolagem consistente o suficiente para ser associado a técnica de detecção de novidades e espera-se que o resultado dessa etapa mostre isso.

As curvas citadas são medidas reais de velocidade de um modelo de avião realizadas durante a etapa de ensaio em voo deste modelo. Medidas de dois protótipos diferentes do mesmo modelo foram usadas. Além das medidas de velocidade, o sinal lógico real de travamento do trem de pouso foi utilizado como marcador do final da decolagem para a obtenção das curvas. Esses dados foram cedidos pela Embraer S.A. para a realização deste trabalho com a condição da descaracterização dos valores para a proteção de informações sensíveis da empresa. Essa descaracterização muda os valores absolutos das medidas de maneira que características sensíveis dos aviões não possa ser extraída mas mantém a coerência entre os dados e os valores dentro de um intervalo representativo da realidade. Entendeu-se que isso não altera o trabalho ou as conclusões derivadas dele.

O método de treinamento utilizado para o detector de novidades consiste na sequência de passos descrita abaixo, utilizada em Alves e Torres (2016).

1. Separar uma curva do resto dos dados de treinamento.
2. Criar um modelo de mistura Gaussiana com todas as curvas, exceto a que foi separada.

3. Calcular a distância de Mahalanobis entre a curva que foi separada e o modelo Gaussiano.
4. Repetir os passos 1, 2 e 3 para cada uma das curvas do conjunto de treinamento, obtendo uma lista de distâncias de Mahalanobis, uma para cada curva.
5. Calcular um valor de Limite que seja maior do que 95% das distâncias de Mahalanobis calculadas.
6. Criar um modelo de mistura Gaussiana com todas as curvas de treinamento, sem exceção.

O resultado do treinamento é o modelo Gaussiano que contém todas as curvas de treinamento e o valor limite.

Dado uma nova curva, sua análise é feita calculando a distância de Mahalanobis entre essa curva e o modelo de mistura gaussiana resultante do treinamento e comparando-a com o valor limite obtido do treinamento. Se a distância é menor que o limite, a nova curva é classificada como sem novidades. Se a distância é maior que o limite, a nova curva é classificada como tendo novidades.

A validação cruzada é feita através de um k-fold com os dados disponíveis, detalhado na seção 4.2.2. O resultado esperado da validação que prova o funcionamento adequado do conceito é que o maior número possível de curvas de teste sejam classificadas adequadamente como COM ou SEM novidades. A média de classificação correta dos k-fold deve ser calculada pra esta análise.

3.3 Desenvolvimento de votador complexo com detecção de novidades

O objetivo seguinte é o desenvolvimento de uma técnica de votação complexa nova com o objetivo de aumentar a confiabilidade da consolidação de dados do ADS.

O estado da arte de técnicas semelhantes de votação complexa ([OLIVEIRA; GALVÃO, 2011](#)) utiliza um votador simples que calcula o sinal consolidado com base nos sinais de entrada que são considerados válidos e técnicas de detecção de erro paralelas que decidem se os sinais de entrada são ou não válidos ao longo do tempo. O votador desenvolvido segue a mesma linha, sendo que as escolhas principais passam a ser qual votador simples utilizar para consolidar o sinal e quais técnicas de detecção de erro incluir.

O votador simples escolhido para ser utilizado é o mesmo utilizado no votador padrão, de maneira que será possível observar o impacto da alteração nas técnicas de detecção de erro sobre a performance do votador desenvolvido. Para a parte de detecção

de erro a ideia é utilizar a detecção de novidades se a sua prova de conceito explicada em 3.2 funcionar.

A aplicação da detecção de erro utilizando a detecção de novidades tem mais elementos do que já explicado na prova de conceito. A lógica dessa técnica é analisar uma curva completa e calcular se ela tem ou não uma novidade em relação ao esperado. Por causa disso, só faz sentido executar o algoritmo quando todos os dados da curva estão disponíveis. Para garantir isso utiliza-se uma entrada de gatilho no votador, que executa a detecção de novidades quando há uma borda de subida no gatilho. Como entrada desse gatilho utiliza-se o sinal de trava do trem de pouso. Ele permanece em FALSO enquanto o trem de pouso está aberto e transita para VERDADEIRO assim que ele termina de ser recolhido, funcionando como claro indicativo de que a decolagem terminou.

Observando as curvas de decolagem disponíveis, foi percebido empiricamente que o intervalo de 60s captura bem a manobra completa. Dessa maneira, quando o sinal de gatilho é acionado, os últimos 60s de dados são considerados como sendo a curva a ser caracterizada.

Quando o número de sinais válidos se altera, pode ocorrer um degrau instantâneo no valor consolidado. Para evitar isso, escolheu-se utilizar uma lógica de amortecimento semelhante ao presente em [Oliveira e Galvão \(2011\)](#) para o sinal consolidado quando ocorrerem essas transições. O amortecimento promove uma interpolação linear entre os valores na transição durante um período definido, evitando o degrau.

Considerando as escolhas mostradas acima, o votador complexo baseado em detecção de novidades funciona da seguinte maneira:

1. Em cada ciclo o sinal consolidado é calculado utilizando uma técnica de votação simples aplicada a todos os sinais de entrada considerados válidos.
2. Os valores dos últimos 60s dos sinais de entrada são armazenados utilizando uma janela móvel.
3. Se acontecer uma borda de subida no sinal de gatilho significa que a decolagem acabou. O detector de novidades é executado para cada sinal de entrada usando os dados armazenados.
4. Caso alguma novidade seja encontrada em um sinal de entrada, aquele sinal é considerado inválido e removido da consolidação daquele momento em diante.
5. Sempre que o número de sinais válidos se alterar o amortecimento promove uma transição suave entre o valor consolidado anterior e atual.

Espera-se que esse votador complexo tenha ganhos de confiabilidade para o ADS já que a detecção de erro é individual para cada sinal, podendo funcionar para quaisquer

número de sinais com falha. As detecções de erro atuais baseiam-se nas relações dos sinais entre si e tem um limite duro de aplicabilidade. Em uma situação em que 3 sinais estão com falha e 1 está saudável, é esperado que o votador complexo desenvolvido tenha sucesso em identificá-los, o que seria mais raro no votador de [Oliveira e Galvão \(2011\)](#). Além disso é importante notar que é feita uma escolha ao definir o escopo desse votador proposto e utilizar somente uma técnica de detecção de erro (a detecção de novidades) apenas na manobra de decolagem.

3.4 Análise de votadores complexos

A última etapa do trabalho é a comparação do votador desenvolvido utilizando a detecção de novidades com um votador considerado estado da arte para consolidação de sinais ([OLIVEIRA; GALVÃO, 2011](#)).

Essa comparação é feita de maneira semelhante ao que já foi descrito em [3.1](#). Um sistema para cada votador é criado e simulado variando parâmetros de entrada. Depois, a medida consolidada do ADS é comparada com a entrada de referência do modelo para calcular o erro e a performance do votador.

Apesar das semelhanças com a comparação de votadores simples, a comparação desses dois votadores complexos exige mudanças na metodologia. Primeiramente que a detecção de novidades, como explicado nas seções [3.2](#) e [3.3](#), funciona depois de treinada para uma curva característica. Assim foram utilizadas as mesmas curvas de decolagem da prova de conceito do detector de novidades para as simulações nessa etapa.

Outra diferença em relação à comparação de votadores simples é que tanto o votador de referência quanto o desenvolvido neste trabalho têm como base o aumento da sua performance através de técnicas de detecção de erro temporais que trabalham em paralelo a uma votação simples. É necessário um tempo de análise do comportamento dos sensores para que os votadores complexos atuem no sentido de remover algum que por ventura possua uma falha. Especificamente para o detector de novidades, ele precisa de observar a decolagem completa para atuar. Assim, decidiu-se simular 60s de decolagem seguidos por mais 10s de operação, realizando a comparação da performance somente nos 10 segundos finais de operação.

4 Implementação

Para fazer a análise de técnicas de votação aplicadas a um sensor de fluxo térmico foi escolhido utilizar o ambiente de simulação Simulink, parte do Matlab. Este software permite utilizar modelos construídos com blocos matemáticos e lógicos e programação por script para realizar as simulações e analisar os resultados.

4.1 Modelos

4.1.1 Sensor de fluxo térmico

O modelo do sensor de fluxo térmico foi construído seguindo as instruções e fórmulas matemáticas do artigo de (MAKINWA; HUIJSING, 2000) e mostradas em 2.3. Dentre vários sensores listados na revisão feita por Haasl e Stemme (2007), esse modelo foi o escolhido por estar mais próximo da aplicação aeronáutica. Ele foi desenvolvido para operar no ar, tem grande faixa de operação, não interfere e nem precisa estar em contato direto com o fluxo, consegue medir velocidade e ângulo e é uma melhoria de um sensor comercial já disponível no mercado. Suas entradas são as diferenças de potências δP_{ns} , δP_{ew} , os desbalanceamentos térmicos nas termopilhas a_{ns} e a_{ew} , as constantes de proporcionalidade S_{ns} e S_{ew} e os parâmetros de diferenças de fase em relação a referência E_{ns} e E_{ew} . As saídas são os valores de velocidade e direção do fluxo, medidas pelo sensor. O bloco construído que representa essas informações pode ser observado na figura 8.

Não foram considerados no modelo o circuito eletrônico do sensor. O modelo matemático descrito em Makinwa e Huijsing (2000) e já descrito na seção 2.3, mais especificamente as fórmulas 2.2, 2.3 e 2.4, já levam em conta possíveis defeitos de fabricação e operação do sensor, sendo esses os parâmetros a_{ns} e a_{ew} representando a assimetria térmica na termopilha, os parâmetros S_{ns} e S_{ew} fatores de proporcionalidade e ε_{ns} e ε_{ew} diferenças de fase em relação a referência para orientação da direção.

Ao invés de estudar o efeito de todas as falhas causadas por valores incorretos desses parâmetros, foi decidido focar no efeito de um deles na operação do sensor. Dentre os parâmetros citados a assimetria térmica é escolhida por ser considerada representativa para esse tipo de sensor, pois uma alteração nela tem o mesmo efeito de uma medida errada de diferença de potência δP_{ns} ou δP_{ew} e afeta tanto o cálculo de velocidade quanto de ângulo.

Assim, os parâmetros a_{ns} e a_{ew} foram escolhidos para modelar o erro padrão característico e as falhas que queremos inserir nos sensores, e por consequência durante a simulação os parâmetros S_{ns} e S_{ew} são fixados em 1 e os parâmetros E_{ns} e E_{ew} fixados em

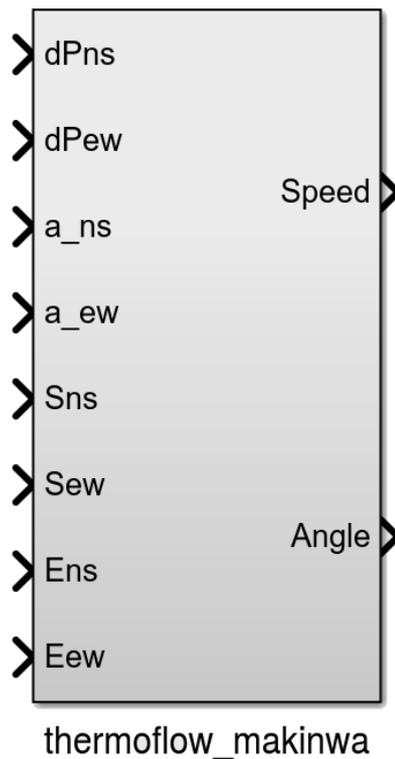


Figura 8 – Modelo Simulink do sensor de fluxo térmico.

0 para que não interfiram nos valores dos sensores.

A modelagem dos defeitos utilizando esses parâmetros permite que a simulação demonstre o efeito representativo de uma falha característica de sensores de fluxo térmico na operação integrada de um ADS.

4.1.2 Cálculo das entradas dos sensores

As simulações planejadas na seção 3 utilizam como entradas dados de velocidade, sendo eles gerados artificialmente como na seção 3.1, ou reais obtidos da Embraer como nas seções 3.2 e 3.4. A entrada dos sensores de fluxo, por outro lado, utiliza a diferença de potência. Observa-se então a necessidade da implementação de um modelo que realize essa adequação entre as grandezas.

Este bloco tem como entradas a velocidade (Speed) e ângulo (Angle) do fluxo de ar e um sinal de controle para a execução do bloco (Enable). Como saídas ele fornece os sinais de potência δP_{ns} e δP_{ew} e os sinais de desbalanceamento térmico a_{ns} e a_{ew} . A figura 9 mostra essa interface.

A conversão matemática das entradas nas saídas especificadas é a aplicação inversa das fórmulas do modelo do sensor de fluxo térmico de 4.1.1. A diferença é que outras funções são incluídas nesse bloco: modelagem do erro natural do sensor de fluxo medido

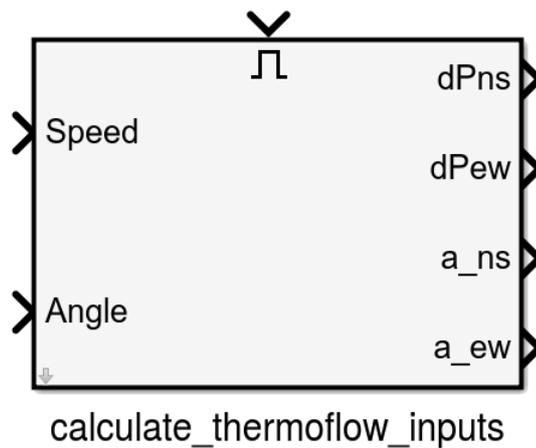


Figura 9 – Modelo Simulink de cálculo das entradas do sensor.

na prática e inserção de falhas nos sensores.

O erro natural do sensor é inserida diretamente nos sinais de δP através da soma do valor calculado com um valor aleatório que produz erro máximo de 4% na velocidade. Esse valor origina-se de medições experimentais em (MAKINWA; HUIJSING, 2001).

A possibilidade de inserir falhas é controlada através de parâmetros fixos decididos antes da simulação ser executada. Os tipos de falhas são Derivativa, Constante, Proporcional e Rampa, e serão descritos com mais detalhes na seção 4.2.1 que explica a execução. O modelo insere as falhas calculando os valores de saída de a_{ns} e a_{ew} levando em conta o tipo e a amplitude da falha escolhidas.

O sinal de enable do bloco foi inserido para que os valores só sejam realmente calculados e fornecidos quando ele estiver ativado. Isso é necessário pois há uma janela de tempo no início da simulação em que algumas inicializações são feitas e nesse período o cálculo dos valores é pausado.

4.1.3 Sistema com quatro sensores de fluxo térmico

O modelo de cálculo das entradas dos sensores 4.1.2 é conectado ao modelo do sensor de fluxo térmico 4.1.1. Essa junção permite utilizar como entrada velocidade do fluxo e modela por completo o comportamento do sensor de fluxo térmico, seu erro padrão e falhas inseridas. A figura 10 abaixo mostra essa configuração.

O Air Data System utilizado neste trabalho é composto por quatro sensores de fluxo térmico dispostos de maneira redundante. Dessa maneira, o modelo da figura 10 é utilizado quatro vezes para compor o ADS. A entrada de todos eles são os mesmos sinais de velocidade e ângulo, mas as saídas são bem diferentes, uma vez que os componentes aleatórios do erro padrão são inseridos. Além disso a quantidade de falhas no sistema também é um parâmetro da simulação e varia, de maneira que a existência ou não de

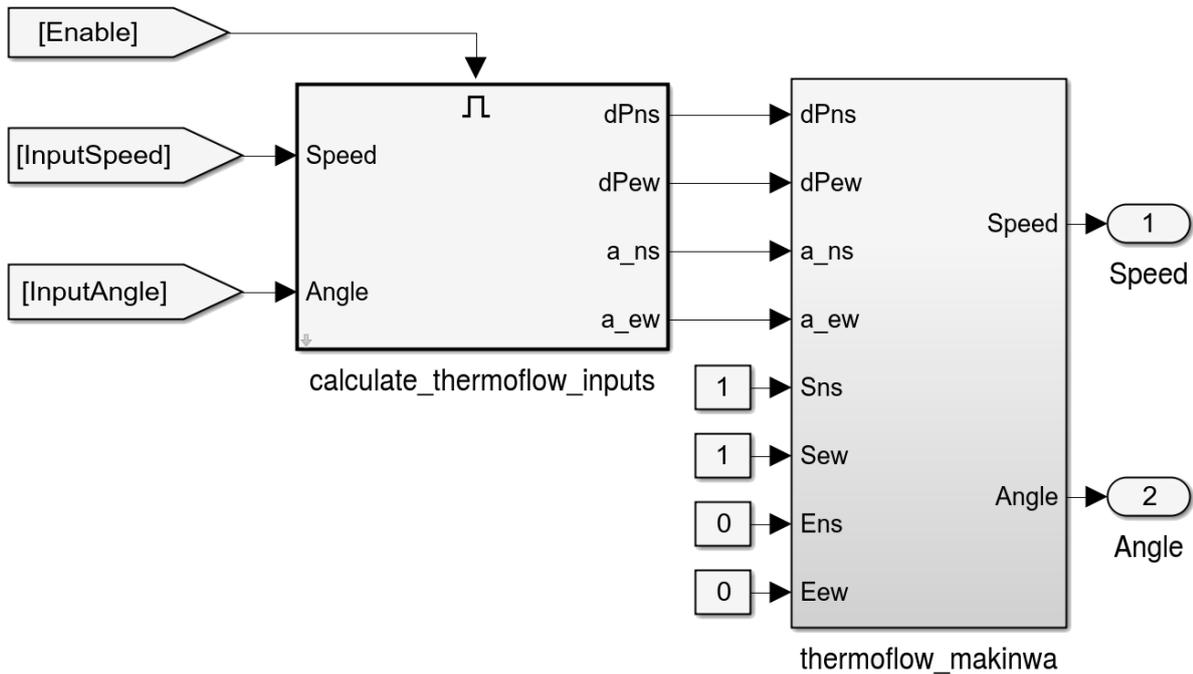


Figura 10 – Modelo Simulink com cálculo da entrada e sensor de fluxo conectados.

falhas em cada uma das quatro instâncias é controlada individualmente.

4.1.4 Votadores

Modelos de todos os votadores descritos em 2.4 e do votador desenvolvido em 3.3 foram implementados para que pudessem ser utilizados nas simulações. Dentro do ADS esses votadores foram incluídos de maneira que o parâmetro *VoterSelection* pudesse ser usado para escolher qual votador utilizar em uma simulação específica. A figura 11 mostra essa implementação.

As entradas e saídas de todos os votadores foram padronizadas para que eles fossem intercambiáveis nas simulações. Todos possuem quatro entradas para os sinais redundantes medidos pelos sensores. Além disso todos possuem uma entrada de *Enable* que habilita ou não a execução do bloco. De Saída, eles possuem um sinal consolidado de velocidade *Speed*, resultado da votação e um sinal de *Validity* que indica se o valor votado deve ser considerado válido ou não. A seleção de qual votador será conectado ao sinais de saída está mostrado na figura 12.

Dependendo da necessidade as saídas dos votadores foram alteradas para que estivessem de acordo com esse padrão. Por exemplo, o votador padrão possui entradas de validade individuais para cada sensor. Como a simulação não leva em conta falhas anunciadas pelo próprio sensor essas entradas foram fixas em *True* para esse votador. Outro exemplo é que muitos dos votadores simples não possuem inteligência para considerar o

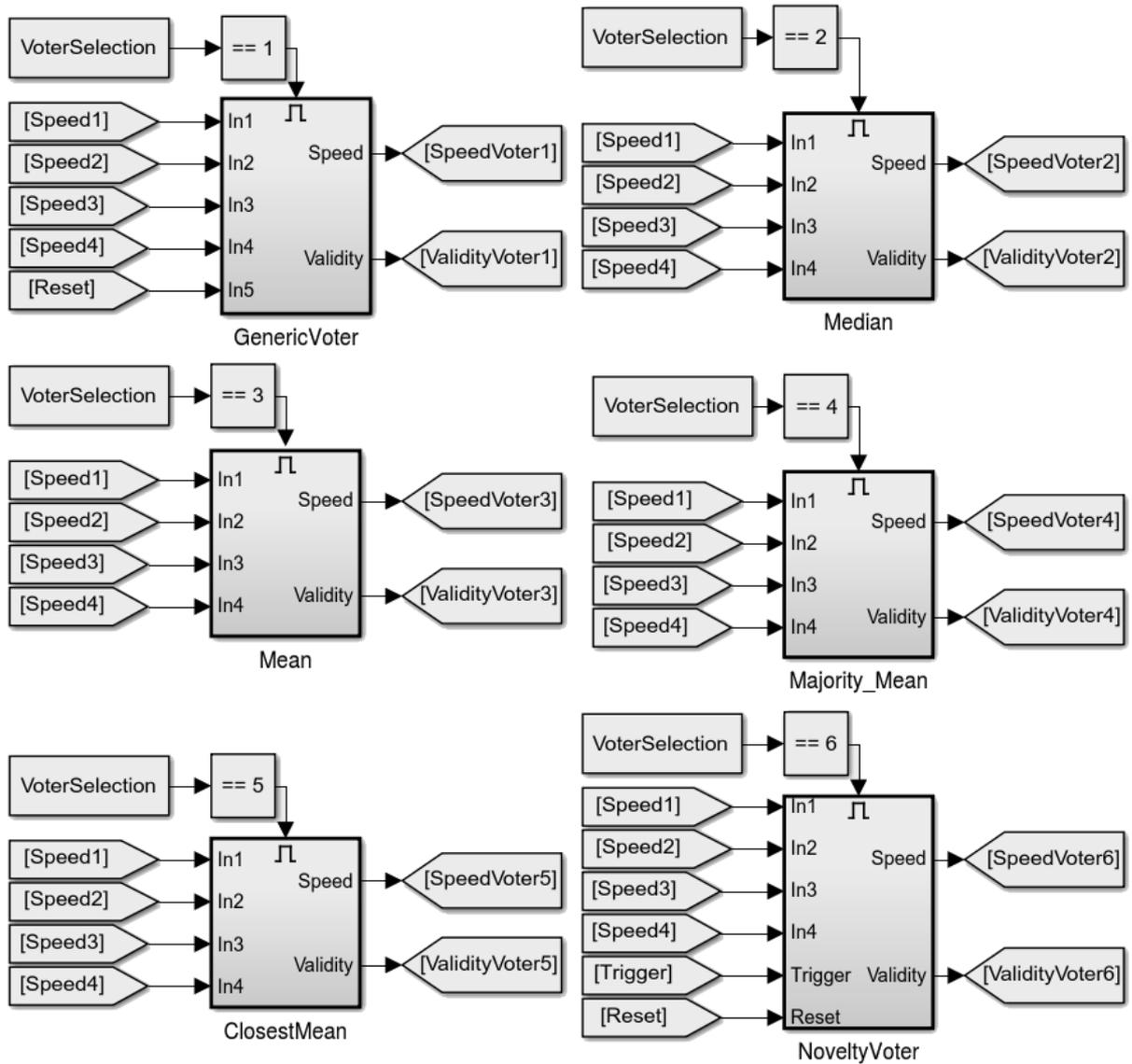


Figura 11 – Modelo Simulink de seleção de votadores.

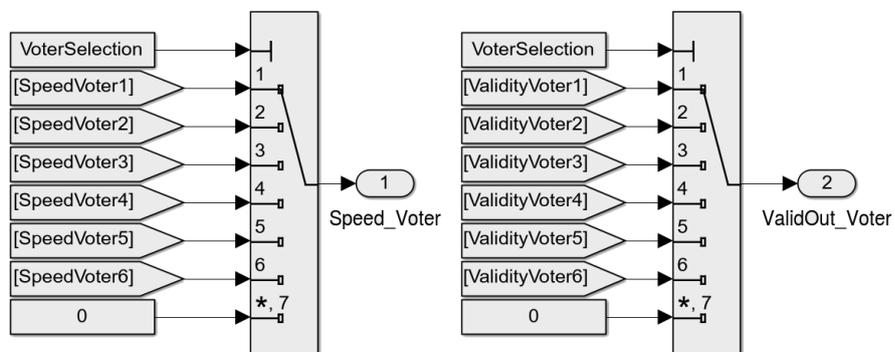


Figura 12 – Modelo Simulink de seleção de saída dos votadores.

valor votado válido ou não. Nesse caso o sinal de validade de saída foi fixo em *True* para esses votadores.

4.1.4.1 Votador Padrão

Devido à complexidade do votador padrão utilizado, proposto por [Oliveira e Galvão \(2011\)](#), foi utilizado um fluxo de implementação que envolveu o teste funcional isolado de cada macro-componente descrito no artigo e depois um teste integrado do votador, com inserção de sinais predefinidos para observar a operação do sistema, garantindo uma extensa verificação do modelo. Além disso, este votador possui uma série de parâmetros que modificam a sua operação e devem ser escolhidos para cada aplicação. O processo de implementação também envolveu a escolha mais apropriada de parâmetros para a aplicação do sensor de fluxo com suas características de operação. A figura 13 mostra o ambiente de teste do votador padrão.

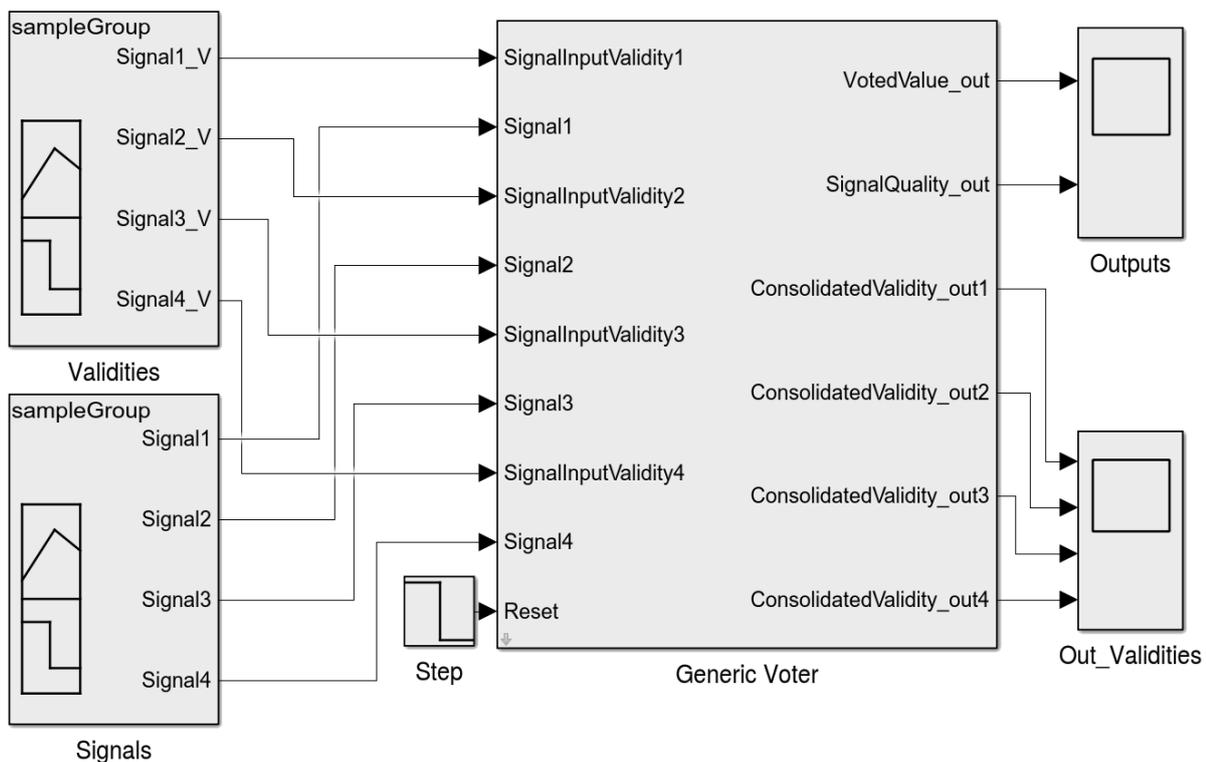


Figura 13 – Modelo Simulink de teste do votador padrão.

Ao final da implementação os parâmetros mostrados na tabela 1 foram escolhidos para serem usados na instância desse trabalho, adequados para a operação requerida e comparação com outros métodos.

Alguns dos parâmetros citados na tabela merecem uma explicação mais detalhada. O valor do erro padrão *TF_DefaultError* é escolhido como 4 devido a aplicação, que sabemos devido às curvas de decolagem, tem velocidades variando de 0 a até aproximadamente 100.

Parâmetro	Valor
TF_DefaultError	4
Flagged_DebounceWindowSize	10
Flagged_DebounceSamplesToTrue	7
Miss_Threshold	2*TF_DefaultError
Miss_TimeConstant	1
Miss_FilteredThresholdOFF	1*TF_DefaultError
Miss_FilteredThresholdON	2*TF_DefaultError
Range_SignalUpperLimit	10000
Range_SignalLowerLimit	-10000
Range_PersistentTime	1
Osc_LowFrequency	4Hz
Osc_HighFrequency	49Hz
Osc_Threshold	1
Osc_PersistentTimer	0.5
Split_MaxDiffToPair	TF_DefaultError
Split_MinDiffToSplit	2*TF_DefaultError
Split_PersistentTimer	1
Con_HealingTimer	1
Fader_FaderTimer	0.2
TS	0.01

Tabela 1 – Parâmetros da utilização do votador padrão.

Nesse caso o erro padrão aleatório do votador de 4% aplicado ao valor máximo esperado resulta no valor 4 para essa constante. Outros parâmetros são definidos em relação ao TF_DefaultError, como por exemplo Miss_Threshold, definido como 2 x TF_DefaultError. Esse valor é escolhido porque, como já é esperado que o valor do sensor varie dentro do erro padrão, só faz sentido detectar um erro de *Miscompare* se a diferença for maior. A mesma dependência existe para os parâmetros de *Miscompare* e *Split*.

A descrição completa de todos os componentes internos e lógicas do votador padrão pode ser encontrada em [Oliveira e Galvão \(2011\)](#). O modelo criado neste trabalho replicou as lógicas como descritas.

4.1.4.2 Votador com detector de novidades

A implementação do votador com detecção de novidades seguiu a lógica explicada na seção 3.3. Ele possui, portanto, quatro entradas para os sinais redundantes, uma entrada de *Trigger* que indica ao votador o momento em que a curva de interesse está completa e uma entrada de *Reset*. Suas saídas são o valor votado e a qualidade do sinal, sendo que essa última é um número inteiro que indica quantos sinais o votador considera válidos naquele momento. A figura 14 mostra a interface do bloco.

Os blocos internos do votador desenvolvido estão descritos separadamente nas três

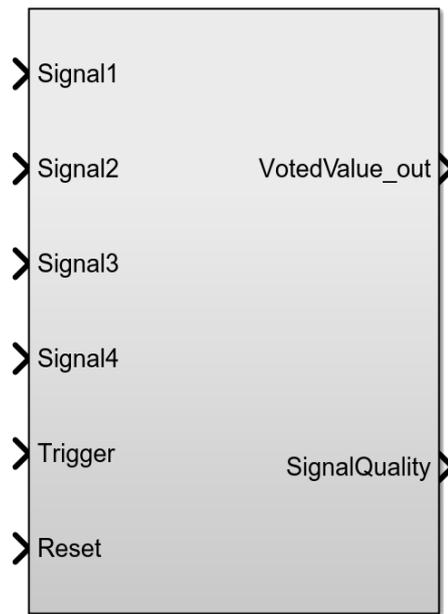


Figura 14 – Modelo Simulink do votador com detecção de novidades.

figuras 15, 16 e 17. O estágio de entrada do modelo está na primeira delas e conta com um filtro passa baixa para cada sinal recebido. A necessidade deste filtro foi identificada na primeira vez em que o votador foi integrado ao ADS. Nessa ocasião percebeu-se que o ruído causado pelo erro padrão aleatório de 4% do sensor de fluxo térmico era percebido sempre como uma novidade pelo detector, uma vez que os dados utilizados para treinamento originais não possuem tal oscilação. O filtro foi inserido e ajustado para suavizar a onda que é lida pelo detector de novidade e permitir uma comparação adequada com os dados de treinamento. É importante notar que o filtro não altera os sinais considerados para cálculo do valor votado, apenas para a detecção de novidades. Isso significa que o ruído inserido continua influenciando o cálculo da saída e que atrasos de fase devido aos filtros afetam somente o tempo em que a detecção de novidade ocorre e não o valor votado. Efeitos de temporização e atrasos são discutidos em detalhe na seção 5.3.5.

Logo após os filtros há um detector de novidade para cada sinal de entrada. Eles funcionam armazenando em janela móvel os valores dos últimos 60 segundos de execução. Apenas quando o sinal de *Trigger* na sua entrada tem uma borda de subida que o detector compara a curva armazenada com a curva esperada de uma decolagem padrão. Se há diferenças significativas, considerando os fatores já explicados em 3.2, a saída do bloco vai pra *True*, indicando que há novidade.

Em seguida há o estágio de cálculo do valor votado mostrado na figura 16, em que foi utilizado o método de cálculo através da mediana dos sinais de entrada. Note que o bloco recebe os valores dos sinais sem passar pelo filtro e as saídas dos detectores de novidades como validade dos sinais. Se uma novidade é detectada a validade daquele sinal

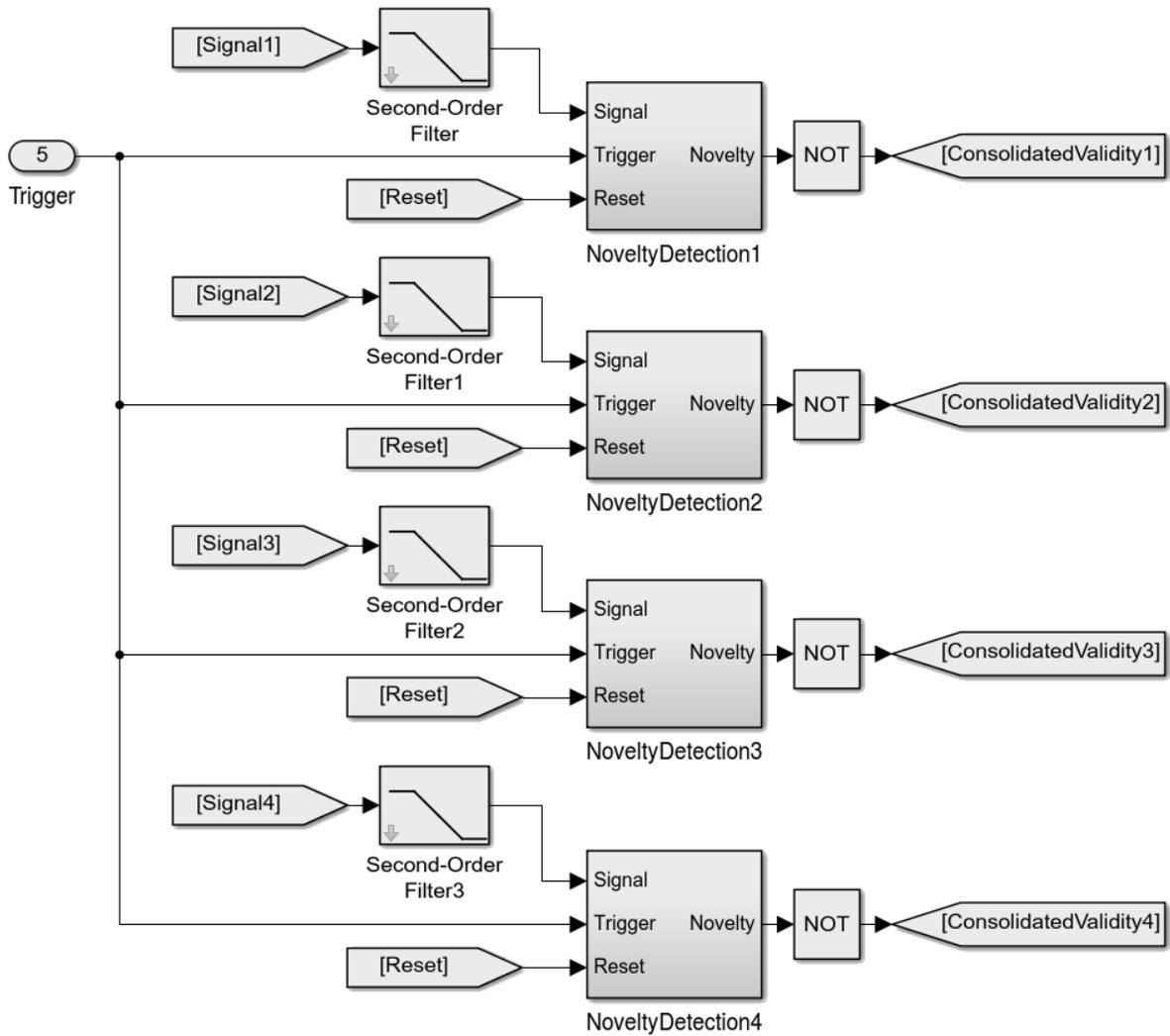


Figura 15 – Primeiro estágio interno do votador com detecção de novidades.

muda para *False* e ele passa a ser desconsiderado no cálculo do valor consolidado.

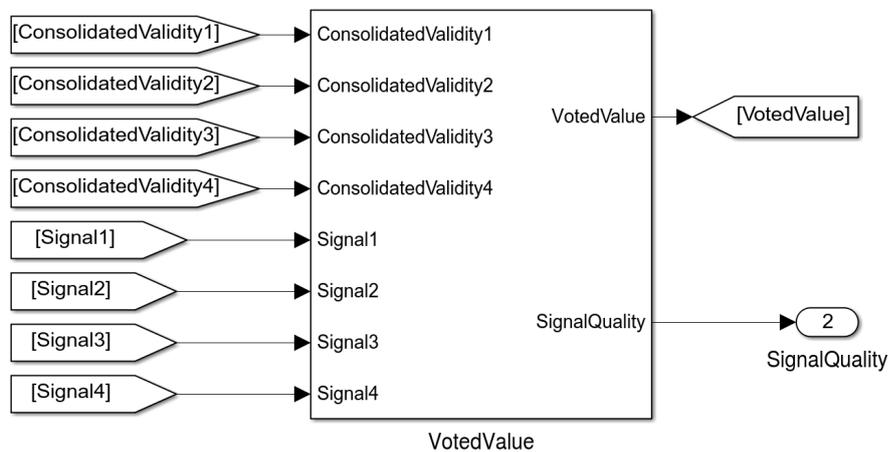


Figura 16 – Segundo estágio interno do votador com detecção de novidades.

Por último há o estágio de *Fader* na figura 17 que provoca uma transição suave no sinal de saída sempre que a quantidade de sinais válidos se altera.

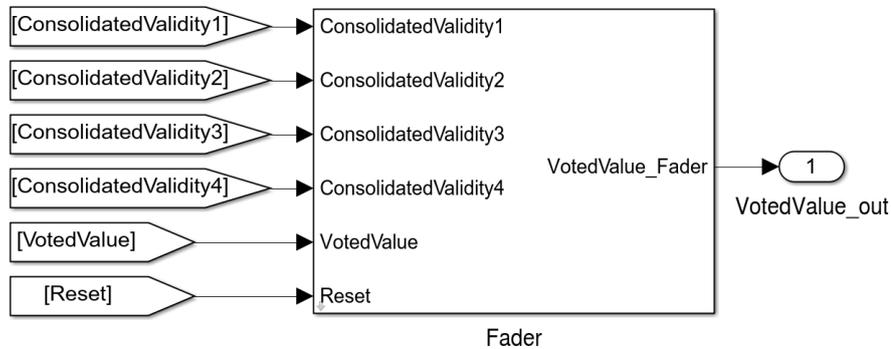


Figura 17 – Terceiro estágio interno do votador com detecção de novidades.

4.1.5 Comparação

A lógica de comparação é responsável por identificar a diferença entre o valor calculado pelo votador *Speed_Voter* e a referência *SpeedReference*, que para o ADS implementado é a própria entrada de velocidade do sistema. Na figura 18 pode-se ver que a diferença absoluta entre os dois, assim como a porcentagem que essa diferença representa, são nossos valores de interesse calculado. Um limite inferior de 0.1 é colocado para o cálculo de porcentagem para que valores pequenos de velocidade não provoquem uma distorção nos dados. A entrada de *Reset* funciona para interromper o cálculo quando está ativada (evitando dados incorretos). Por último, a entrada que contém a informação de validade da saída do votador *ValidOut_Voter* não é utilizada.

4.1.6 ADS integrado

A integração dos blocos descritos nas seções anteriores resulta no modelo completo do Air Data System, denominado *ADS_model*. Seu diagrama principal pode ser observado na figura 19.

Pode-se ver no modelo todos os estágios que já descrevemos. As entradas são a velocidade (*InputSpeed*) e ângulo (*InputAngle*) do fluxo, um sinal de *Reset* e o sinal *NLG_UL*, sigla para *NoseLandingGearUpLock*, responsável por indicar o momento em que a decolagem acaba e servir como acionamento para o detector de novidades.

O bloco *ThermoflowSensors* contém a configuração de quatro sensores de fluxo térmico redundantes descritos em 4.1.3. O bloco *Voters* contém todos os votadores, como explicado em 4.1.4. Por último, *Comparison* contém a lógica mostrada em 4.1.5.

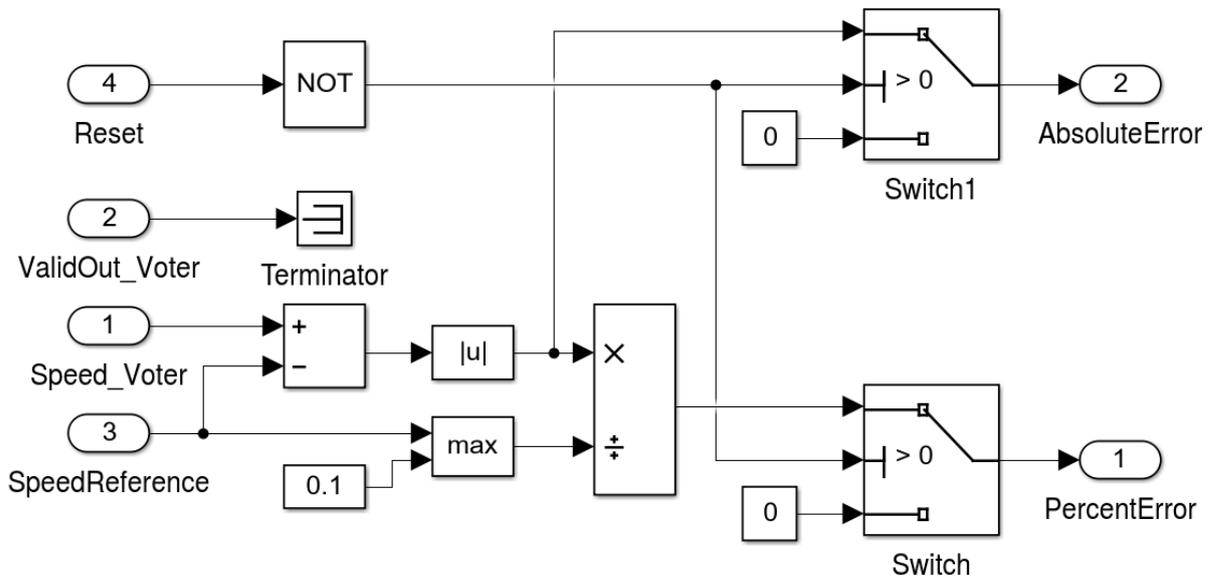


Figura 18 – Modelo Simulink da comparação entre valor votado e referência.

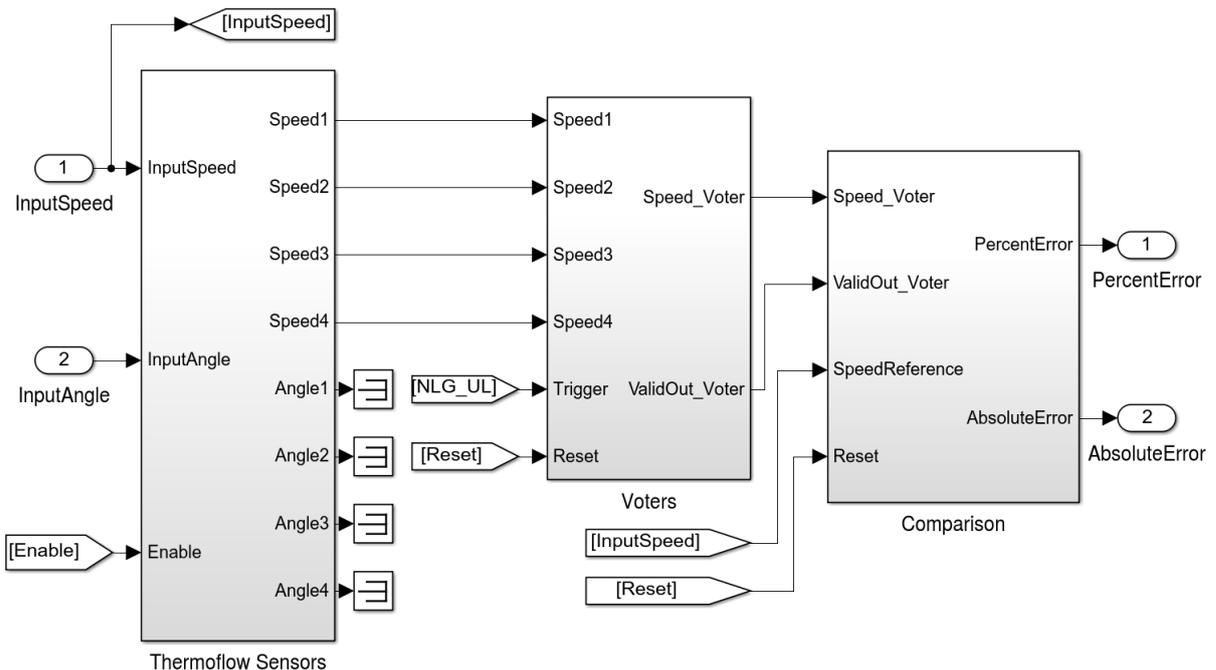


Figura 19 – Modelo Simulink do Air Data System.

As saídas do modelo são as quatro medidas de velocidade dos sensores *Speed1..4*, a velocidade consolidada do votador escolhido *SpeedVoter*, a validade dessa informação, *ValidityVoter*, além dos erros absolutos e percentuais *AbsoluteError* e *PercentError*.

Além dos elementos visíveis na figura, o modelo integrado do ADS também utiliza vários parâmetros fixos, definidos antes da simulação, sendo que alguns já foram citados

nas seções anteriores. A lista completa deles e o seu valor está na tabela 2.

Parâmetro	Valor padrão	Uso
TS	0.01	Taxa de execução da simulação
EndTime	-	Tempo total da simulação
VoterSelection	-	Votador a ser utilizado
TF_DefaultError	4	Erro padrão na saída
TF_DefaultErrorPercent	0.04	Erro padrão na saída (%)
CTI_InputDeviationPercent_ns	0.0198	Erro padrão no componente ns
CTI_InputDeviationPercent_ew	0.0198	Erro padrão no componente ew
CTI_ErrorType_ns[1...4]	-	Tipo de erro ns nos sensores 1 a 4
CTI_ErrorAmp_ns[1...4]	-	Amplitude do erro ns nos sensores 1 a 4
CTI_ErrorType_ew[1...4]	-	Tipo de erro nos ew sensores 1 a 4
CTI_ErrorAmp_ew[1...4]	-	Amplitude do erro ns nos sensores 1 a 4
CTI_StartSeed[1...4]	-	Semente aleatória do erro padrão

Tabela 2 – Parâmetros do ADS integrado.

Os parâmetros sem valor na tabela não possuem um valor padrão e precisam ser definidos para cada simulação. A sigla *CTI* de alguns parâmetros indica que eles são usados pelo bloco que calcula a entrada do sensor de fluxo (*Calculate Thermoflow Inputs*). Já a sigla *TF* representa parâmetros do sensor de fluxo (*Thermo Flow*) mas que são utilizadas em diversas partes do modelo em que são relevantes. Os outros parâmetros dizem respeito a simulação como um todo.

O parâmetro `TF_DefaultError` dessa tabela é o mesmo que já foi mostrado na tabela 1. Ele juntamente com o parâmetro `TF_DefaultErrorPercent`, `CTI_InputDeviationPercent_ns` e `CTI_InputDeviationPercent_ew` são maneiras diferentes de representar a mesma coisa, o erro padrão do sensor de fluxo definido em 4% em sua caracterização (MAKINWA; HUIJSING, 2001). `TF_DefaultError` vale 4 considerando a máxima operação dos sistema, `TF_DefaultErrorPercent` representa o valor percentual e `CTI_InputDeviationPercent_(ns|ew)` são os valores do erro em cada componente ortogonal do sensor para que no final o erro seja de 4%.

4.2 Execução

Após a criação dos modelos é necessário criar um arcabouço que seja responsável por sua simulação e definir valores de entradas para a execução seguindo os moldes estabelecidos pela metodologia. Nas seções seguintes serão explicadas as decisões e implementações escolhidas para realizar isso e obter os dados de saída necessários.

Para detalhes sobre como executar os scripts e indicação de quais são os arquivos de entrada e saída das etapas, consulte o anexo A.

4.2.1 Análise de técnicas de votação simples

A comparação entre as técnicas de votação simples envolve simular os votadores 2, 3, 4 e 5 implementados. Um script se encarrega de realizar essa simulação em vários cenários diferentes, variando parâmetros que alteram a entrada do modelo e a falha inserida no sensor de fluxo térmico. A tabela 3 lista as possíveis alternativas de três parâmetros relacionados à entrada e a tabela 4 lista as possíveis alternativas de três parâmetros relacionados à falha inserida. O scrip simula todas as combinações possíveis entre esses parâmetros. Dado o número de possibilidades para cada parâmetro e excluindo a variação de frequência para tipos de entrada constante e rampa é possível obter 1008 configurações distintas.

Tipo da entrada	Valor da entrada	Frequência(Hz) da entrada
Constante	1	0,1
Rampa	10	1,0
Senoidal	100	5,0
Rotacional	-	10,0

Tabela 3 – Parâmetros da entrada para comparação de votadores simples

Tipo da falha	Valor do erro	Quantidade de falhas
Derivativo	0,1	1
Proporcional	1	2
Constante	10	3
Rampa	-	2 (split)

Tabela 4 – Parâmetros da falha para comparação de votadores simples

A escolha desses parâmetros foi feita para observar um universo de cenários bem expandido dentro do que é realista. O objetivo não é definir a relação entre o desempenho e a variação de nenhum dos parâmetros, mas analisar a comparação entre os votadores no geral, tendo um universo diversificado de situações para utilizar.

Os tipos de entrada escolhidas tentam representar situações realistas. O tipo fixo é simplesmente um valor que não se altera no tempo. A entrada em rampa começa em zero e cresce linearmente, terminando com duas vezes o valor definido para a entrada. A entrada senoidal é uma senoide de amplitude igual ao valor definido para a entrada e variando na frequência escolhida para a entrada, deslocada no eixo Y para não ter valor negativo. A entrada rotacional é mais peculiar pois representa a situação em que um fluxo de velocidade constante igual ao valor escolhido muda a direção com que passa pelo sensor. Isso significa que essa entrada tem valores diferentes para δP_{ns} e δP_{ew} variando no tempo de maneira senoidal e cossenoidal. Como cada direção é um componente ortogonal da velocidade do fluxo, a amplitude da velocidade é constante.

Os valores reais esperados para a operação desse sensor está em torno de 100m/s, portanto esse parâmetro é variado de 1m/s até 100m/s. A frequência de operação típica de sensores de velocidade vai 10Hz até 100Hz, portanto escolheu-se variá-la de um valor bem baixo 0,1Hz até o valor de 10Hz.

As falhas são inseridas no modelo do sensor através da alteração do valor do seu parâmetro a_{ns} . O tipo de falha constante muda o parâmetro para o valor escolhido por toda a simulação. O tipo de falha em rampa altera a_{ns} crescendo linearmente de 0 até 2 vezes do valor especificado do erro. No caso proporcional, o parâmetro de desbalanceamento térmico a_{ns} é igual a entrada do sensor multiplicada pelo valor escolhido da falha. No caso derivativo, a mesma multiplicação, mas considerando a derivada da entrada.

Os valores dos erros associados às falhas, representados por variações em a_{ns} , foram escolhidos empiricamente ao observar como a saída do sensor de fluxo é alterada para uma dada relação entre a_{ns} e a entrada do sistema.

Em relação a quantidade das falhas, foram escolhidos os cenários com 1, 2 e 3 falhas distribuídas em posições e com sinais aleatórios. Outro caso especial é o de duas falhas mas com sinais iguais. Isso provoca a criação de dois pares de sinais que concordam entre si, chamado *split*. Como o votador padrão tem uma lógica específica para identificar esse tipo de falha ela foi incluída na comparação.

Cada uma das 1008 combinações possíveis dos parâmetros corresponde a uma amostra do experimento. Cada amostra, por sua vez, é uma média de 20 simulações fixas em uma combinação de parâmetros. Isso é feito para que uma resposta adequada da simulação seja obtida considerando os aspectos pseudo aleatórios inseridos no modelo. Ao realizar a média de 20 simulações, por exemplo, a posição de uma falha se torna irrelevante pois ela foi colocada em um lugar aleatório para cada uma das simulações, mas o efeito de uma falha no desempenho se mantém e é medida. Cada execução simula 10s de operação. Esse valor foi escolhido por representar um intervalo de tempo realístico em que uma variação errônea na velocidade deve ser percebida.

4.2.2 Prova de conceito do detector de novidades

Os dados de entrada utilizados são cedidos pela Embraer S.A. e correspondem a valores descaracterizados de medidas reais. Dentre os dados de entrada, além da medida de velocidade há também o sinal NLG_UP ("Nose Landing Gear Up Lock") que foi escolhida para representar o fim da manobra de decolagem da aeronave. Dessa maneira, ao coletar as curvas de decolagem, utiliza-se os 60s anteriores à borda de subida do sinal NLG_UP. O intervalo de 60 segundos foi escolhido empiricamente por ser suficiente para conter a manobra de interesse completa.

A primeira etapa é justamente a separação e classificação das curvas utilizando

esse critério em duas categorias. Essa classificação é facilitada por um script que mostra ao usuário a curva de interesse e pede a ele uma classificação manual daquela curva.

Os dois tipos de decolagem em que as curvas foram separadas são o tipo 1, uma decolagem mais suave em que a velocidade da aeronave se estabiliza em um patamar antes do fechamento do trem de pouso e o tipo 2, uma manobra mais aguda em que o objetivo é alcançar maior altitude mais rápido e portanto a velocidade não se estabiliza antes do fechamento do trem de pouso. A utilização de uma ou outra manobra na decolagem depende do piloto, aeródromo ou ensaio sendo realizado.

Os dois tipos de curvas têm semelhanças mas também diferenças significativas. Treinar um sistema sem essa separação poderia significar deixar o detector relaxado de mais, ou seja, queremos identificar falhas que se manifestem alterando a curva menos do que a diferença entre tipos de decolagem. Após a separação foi decidido também utilizar somente as decolagens tipo 1 no restante do trabalho, ignorando as curvas do tipo 2. As implicações dessa definição de escopo do trabalho são discutidas com mais detalhes na seção 5.

Em seguida, com as curvas já selecionadas e classificadas, é realizada a validação do método de detecção de novidade. Isso ocorre através de um k-fold com k igual a 9 em um conjunto de 117 curvas. Isso significa que o processo de treinamento e teste foi repetido 13 vezes, separando as curvas em um conjunto de treinamento com 108 curvas e um conjunto de teste com 9 curvas, de maneira que todas as curvas estão presentes no grupo de teste exatamente uma vez. Essas curvas de teste são originais dos dados de decolagem, e portanto, sem novidades. Por causa disso, em cada repetição também são incluídas no conjunto de teste 9 curvas com novidades, criadas a partir da modificação das curvas sem novidades.

Esse processo resulta em um total de 234 classificações de curvas feitas pelo detector de novidades. Como saída da execução obtêm-se um log de execução e o valor consolidado de quantas vezes a detecção de novidade foi testada, e dessas vezes quantas eram acertos, falsos positivos ou falsos negativos.

4.2.3 Análise de votadores complexos

A comparação das técnicas de votação complexas é feita de maneira semelhante ao já descrito para as técnicas simples, com diferenças que serão detalhas a seguir.

Primeiramente, a detecção de novidades desenvolvida aqui funciona comparando uma curva de decolagem de entrada com um sistema treinado para observar esse tipo de manobra. Então para essa simulação será utilizada como entrada do modelo uma curva formada pela junção de uma decolagem seguida por outra forma de onda de um tipo especificado.

Das 117 curvas de decolagem tipo 1, 97 são escolhidas aleatoriamente para realizar o treinamento do detector de novidades. 20 curvas são reservadas para serem utilizadas na simulação do sistema. A escolha por utilizar 20 curvas de decolagem para teste se baseia no número de simulações por amostra que são realizadas, exatamente 20, como já descrito anteriormente. Dessa maneira cada uma das 20 curvas de teste é utilizada uma vez para gerar cada amostra do experimento. É importante destacar que a etapa de análise de votadores complexos descrita aqui é um experimento independente da prova de conceito da seção anterior 4.2.2, o que significa que a separação das curvas segue um caminho próprio.

Por último, devido ao custo computacional elevado da detecção de novidades, a variação de valores dos parâmetros e conseqüentemente a quantidade total de simulações foi reduzida. Os parâmetros variados são: forma do sinal de entrada; valor da entrada; tipo de relação da falha com a entrada; valor do erro associado à falha; localização e número de falhas. Os valores escolhidos estão mais próximos dos valores nominais esperados durante a operação após a decolagem, totalizam 192 combinações. A tabela 5 lista as alternativas dos parâmetros relacionados à entrada do sistema e a tabela 6 lista as alternativas dos parâmetros relacionados à falha inserida.

Tipo da entrada	Valor da entrada	Frequência(Hz) da entrada
Rampa	1	0,2
Senoidal	10	-
Rotacional	-	-
-	-	-

Tabela 5 – Parâmetros da entrada para comparação de votadores complexos

Tipo da falha	Valor do erro	Quantidade de falhas
Derivativo	1	1
Proporcional	5	2
Constante	-	3
Rampa	-	2 (split)

Tabela 6 – Parâmetros da falha para comparação de votadores complexos

A entrada do tipo constante foi removida por não ser tão diferente assim da entrada em rampa, que é bem representativa da realidade. O valor de entrada 100 também é desnecessário porque nessa simulação esse valor é adicionado à curva de decolagem que já termina em um valor representativo da realidade. A frequência foi fixada em um valor também representativo para a aplicação. Os valores das falhas foram alterados de 3 para 2, escolhendo valores intermediários dos valores antigos.

Por último, o tempo total de execução da simulação foi escolhido para ser 70 segundos. 60 são devido a curva de decolagem, que tem essa duração. Após ela terminar a simulação roda mais 10 segundos com o tipo de entrada escolhido. Esses valores de entrada

após a decolagem usam como referência o último valor dessa curva, ou seja, uma entrada em rampa começa no valor inicial igual a último valor da decolagem e cresce. Uma entrada senoidal oscila em torno do último valor da decolagem, etc.

5 Resultados

Como foi discutido ao longo desse trabalho, o seu objetivo principal é aumentar a confiabilidade de Air Data Systems. Em outras palavras, busca-se fazer com que a diferença entre a medida de airspeed consolidada de um sistema e o valor real de tal grandeza seja o menor possível para um leque amplo de situações.

A meta desse capítulo é verificar através de dados e comparações objetivas se o objetivo geral do trabalho foi atingido e em qual grau, discutindo suas particularidades, pontos positivos e negativos.

A metodologia proposta na seção 3 engloba várias etapas para verificar se as técnicas e ideias propostas terão sucesso nesse objetivo. Nas etapas seguintes os resultados das simulações e testes propostos serão apresentados e discutidos. Serão mostrados os resultados da integração do sensor de fluxo térmico no ADS e a comparação das técnicas de votação simples nesse modelo na seção 5.1. Em seguida a aplicação da detecção de novidades a dados reais de aviação será discutida na seção 5.2. Por fim o votador proposto nesse trabalho é comparado a um votador padrão na seção 5.3, sendo essa a maior contribuição desse texto. Para detalhes sobre a execução, arquivos e scripts, consulte o anexo A.

5.1 Análise de técnicas de votação simples

Nesta seção de análise dos votadores simples a seguinte notação é adotada nos gráficos e texto para se referir a cada um dos quatro votadores:

- Votador 2: Mediana dos sinais de entrada
- Votador 3: Média dos sinais de entrada
- Votador 4: Média da maioria dos sinais próximos de entrada
- Votador 5: Média dos dois sinais mais próximos de entrada

Para analisar o desempenho de cada uma das técnicas escolheu-se analisar principalmente o erro percentual médio de cada uma. O erro absoluto iria diluir muito o efeito para valores pequenos em comparação com valores grandes e indicar um resultado errôneo. A saturação inferior com valor 0.1 no denominador inserida no modelo de comparação descrito em 4.1.5 evita o oposto, que variações pequenas no valor absoluto para cenários de baixa velocidade gerem um erro percentual extremamente grande.

Para a análise o erro médio é comparado para os cenários criados com a variação dos parâmetros descritos na tabela ???. A figura 20 mostra o erro percentual médio dos votadores para diferentes quantidades de sensores com falhas a uma frequência constante de 0,1Hz. Há a separação também entre 2 falhas aleatórias e 2 falhas de mesmo sinal, formando um *split*, como explicado na seção 4.2.1. Nesses gráficos o erro está representado em escala logarítmica. Observando essa figura é possível notar que os votadores 2, 4 e 5 têm comportamentos parecidos e boa performance para apenas um sensor com falha, com valores médios de erro de até 3% para os votadores 2 e 5 e igual a 4,5% para o votador 4. Já o votador 3 tem um desempenho bem pior, com erros atingindo 711%. O votador 3 é o único que sempre utiliza o valor de todas as entradas para o cálculo da saída e isso se reflete no seu desempenho. Ele consegue diminuir a influência de uma falha na saída mas não remover completamente. Observando o comportamento para duas falhas aleatórias e duas falhas em *split* pode-se perceber que o votador 2 piora significativamente, atingindo um erro em torno de 1300% nesses casos. Já o votador 3 piora ainda mais e atinge erros de até 1417%. O votador 4 tem o melhor desempenho para esses casos, com erro de 4,8% para duas falhas aleatórias e 11% para o caso de *split*. O votador 5, por sua vez, apesar de se mostrar pior do que o votador 4, tem valores ainda bem melhores do que os outros, 67% e 118% respectivamente. Para 3 falhas todos os votadores tem desempenhos muito ruins, na casa de 2800%. Esse é o cenário mais difícil pois não há nenhuma maneira de saber qual é o único sensor saudável do sistema.

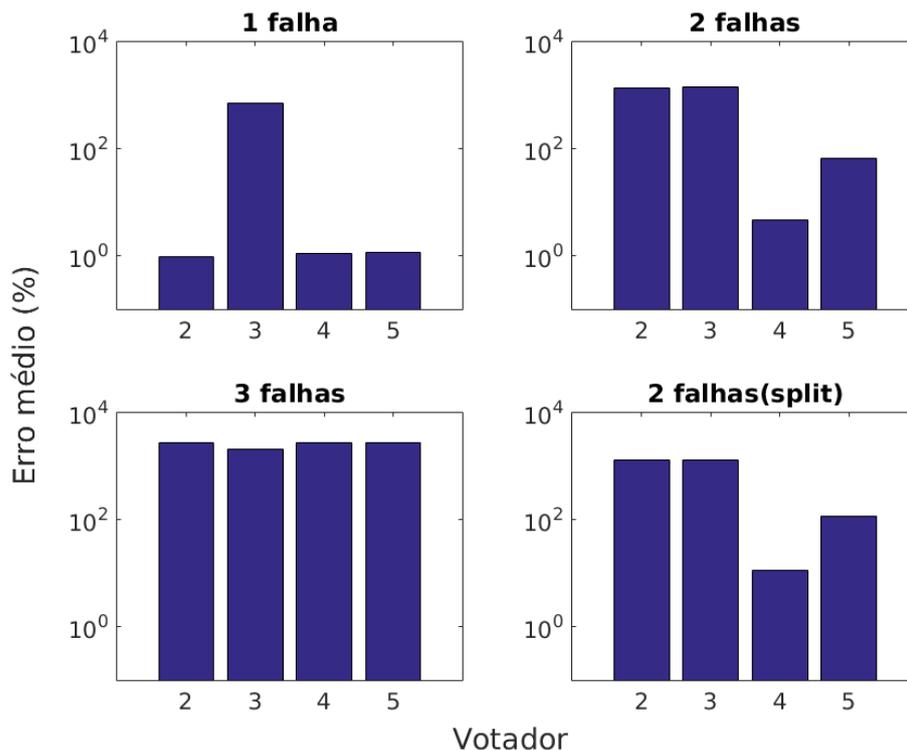


Figura 20 – Erro médio para diferentes quantidades de sensores com falha.

A próxima etapa é analisar como os erros dos votadores se comportam ao variarmos o tipo de entrada do sistema para uma frequência fixa de 0,1Hz. A figura 21 mostra os valores de erro médio para os diferentes tipos de entrada. Pode-se ver ao analisar esses gráficos que os votadores 4 e 5 têm desempenho próximo e são consistentemente melhores do que os votadores 2 e 3 para todos os cenários. É possível ver também que o comportamento de todos os votadores é melhor para os casos de entrada constante e rotacional (que também possui módulo constante) e piores para os casos de entrada em rampa e oscilatória. O melhor votador é o 4, que teve seu menor erro médio para entrada constante (284%) e seu maior para oscilatória (1300%).

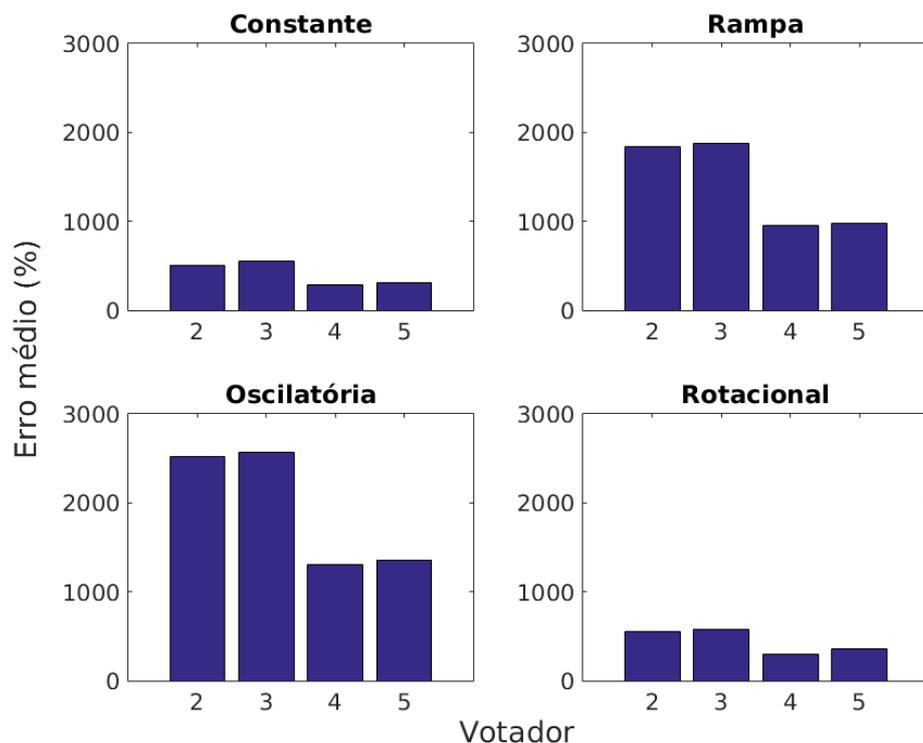


Figura 21 – Erro médio para diferentes tipos de entradas.

O próximo gráfico mostrado na figura 22 contém os dados para variações dos tipos de erro em uma frequência fixa de 0,1Hz. A variação desse fator causa um erro maior para todos os votadores para os casos derivativos e rampa mas não interfere na relação entre os votadores, de maneira que os votadores 4 e 5 são sempre os melhores, com o 4 ainda tendo uma pequena vantagem em relação ao 5 em todos os casos. O pior erro médio chega ao valor de 1875% para o votador 3 no erro em rampa e o melhor fica com o votador 4 no erro proporcional, com valor de 387%.

Alterar a frequência de entrada também não muda o desempenho relativo dos votadores. Os votadores 4 e 5 têm consistentemente melhor desempenho, e o 4 ainda possui uma leve vantagem. Isso pode ser observado na figura 23 que mostra os erros médios de tipos de entradas que variam no tempo (oscilatória e rotacional) para diferentes frequências.

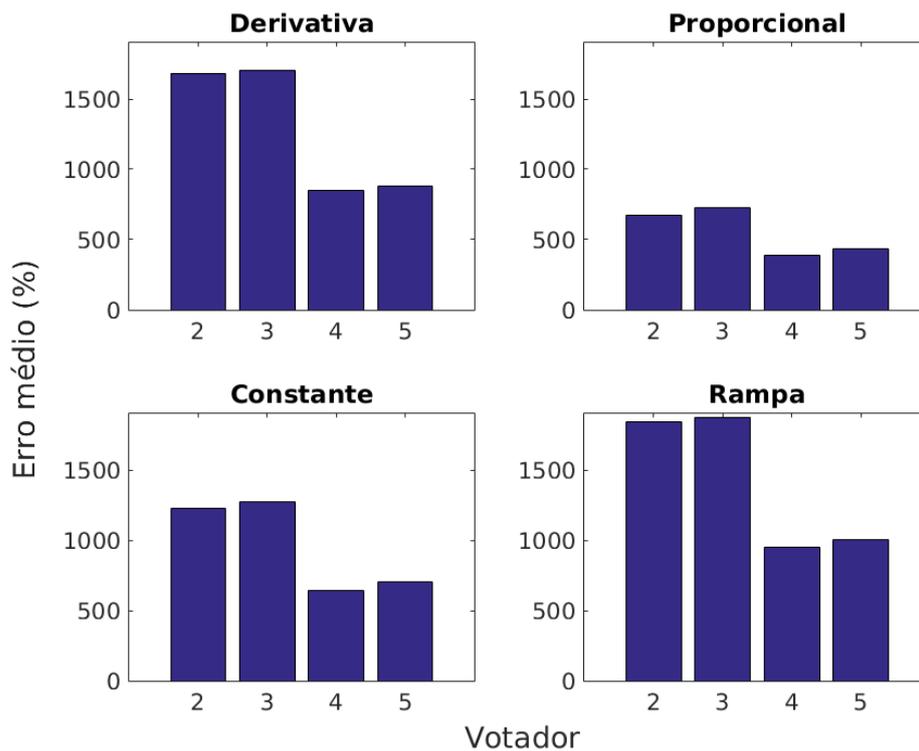


Figura 22 – Erro médio para diferentes tipos de falhas.

Pode-se observar também nesse gráfico que o aumento da frequência provoca um aumento grande no erro médio para todas as técnicas, com pico em 5Hz, e diminuindo um pouco para 10Hz. A diferença chega a ser uma ordem de grandeza com a mudança de 0,1Hz para 1Hz e de mais uma ordem de grandeza ao passar para 5Hz.

Segundo as análises anteriores podemos concluir que a quantidade de falhas é o único fator que altera o desempenho relativo dos votadores. Principalmente ao observarmos os desempenhos para 1 falha, em que apenas o votador 3 tem um desempenho ruim, e para 3 falhas, em que todos os votadores têm desempenho ruim. Analisando o tipo de entrada, tipo de falha e frequência, apesar de alguns casos se mostrarem piores do que outros no geral, a relação entre os votadores permanece: os votadores 4 e 5 têm desempenho melhor, sendo o 4 consistentemente o melhor de todos.

Uma análise geral com todos os casos é feita abaixo. A diferença é que os pontos extremos ("outliers") de desempenho de cada votador são removidos ao considerar somente o quantil inferior de 95% dos pontos. Poucos pontos de valores extremamente altos interferem no cálculo do erro médio e máximo para os votadores. Desconsiderá-los deixa nossa análise mais perto da realidade de operação.

As figuras 24 e 25 mostram os valores médios e máximos dos erros dos votadores, respectivamente. O votador 4 (média da maioria) possui o melhor desempenho dos votadores simples com um erro médio geral de 135,76% e um erro máximo de 1,79E4%. Em segundo

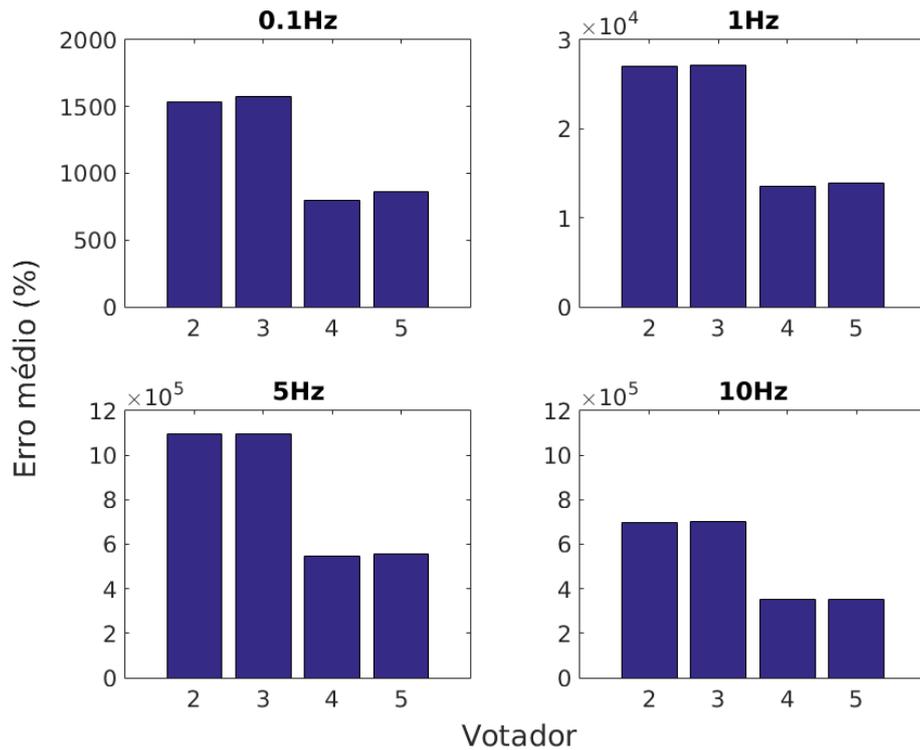


Figura 23 – Erro médio para diferentes frequências de entrada.

lugar o votador 5 (média dos mais próximos) tem um desempenho mais que duas vezes pior, com erro médio de 400% e máximo de $2,69E5\%$. Em seguida aparece o votador 2 (mediana) com valores de erro médio de 1778,4% e máximo de $4,93E5\%$. Em último lugar o votador 3 (média) tem um erro médio mais de 20 vezes maior que o votador 4, na casa dos 3097%. Seu erro máximo chega a $1,22E6\%$.

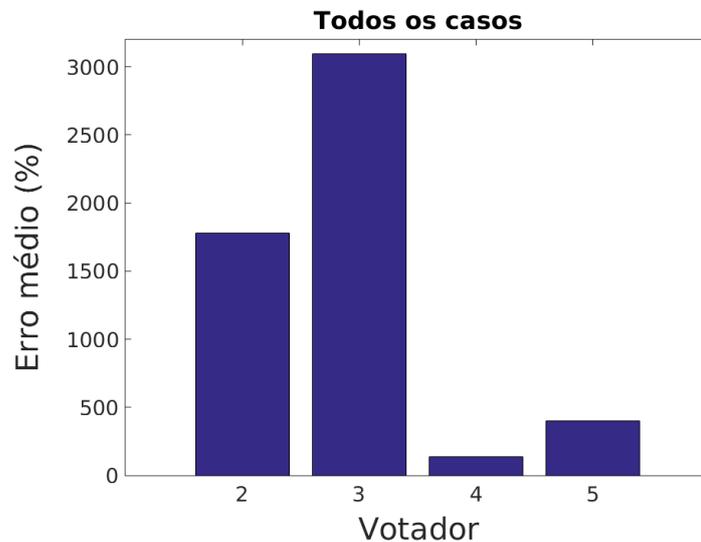


Figura 24 – Erro médio para todos os casos sem outliers

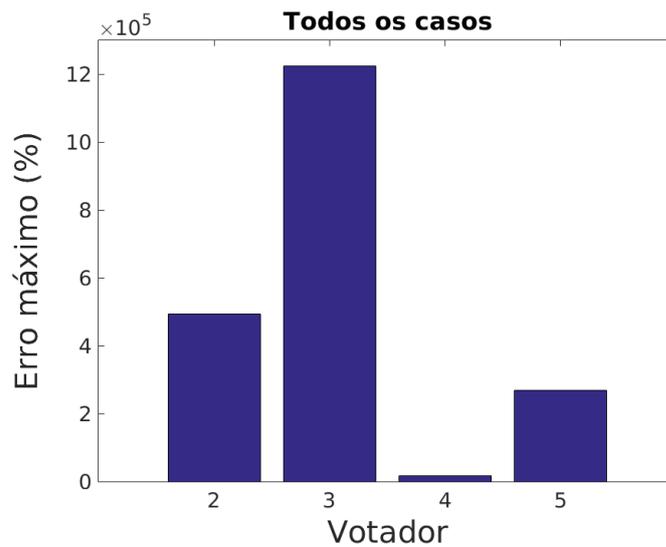


Figura 25 – Erro máximo para todos os casos sem outliers

A conclusão de que o votador que calcula a saída como a média da maioria dos sinais de entrada é o melhor para os casos testados é diferente dos resultados obtidos em [Viana e Torres \(2018\)](#), em que o votador que utiliza a mediana foi considerado o melhor. Existem diferenças substanciais da metodologia que explicam essas diferenças.

A primeira grande diferença é que o artigo em questão escolheu os valores das entradas dos sensores de maneira que elas estivessem distribuídas em um intervalo de $\pm 4\%$ da referência, mas cada entrada tem uma posição fixa nesse intervalo, de maneira que as falhas simuladas nos sensores precisam ter sua posição especificada e essa posição interfere nos resultados obtidos. As simulações realizadas no trabalho atual tomaram o cuidado de distribuir o valor do erro padrão do sensor e a posição da falha inserida de maneira aleatória.

Outra diferença considerável é que dos 4 cenários considerados no artigo, 2 são referentes a somente um sensor com falha. Observando os resultados mostrados acima é possível ver que para uma falha o desempenho do votador 2 (2,17%) é realmente melhor do que o votador 4 (4,53%), embora os valores sejam próximos. Outro cenário observado no artigo é o de 2 falhas opostas, em que os dois sensores mais distantes em módulo da referência falham com erros de sinais opostos. Esse cenário é uma situação específica para duas falhas em que o votador 2 ataca com mais precisão e tem desempenho melhor.

Analisando essas diferenças é possível ver que os resultados do artigo [Viana e Torres \(2018\)](#) estão limitados devido à metodologia e escopo escolhidos para serem utilizados. O trabalho atual amplia o escopo ao utilizar elementos aleatórios nas simulações que tornam irrelevante as localizações e comportamentos específicos das falhas. Além disso o cenário com 3 falhas também é trazido para análise. Podemos concluir que o trabalho atual apresenta um cenário mais completo e bem estabelecido de comparação entre os

métodos de votação, e sob essa ótica o votador 4 que calcula a média da maioria dos sinais tem melhor desempenho.

5.2 Prova de conceito do detector de novidades

Seguindo os detalhes do k-fold explicados na seção 4.2.2 foram realizadas 13 bateladas de teste, cada uma com 18 curvas (9 originais e 9 modificadas para ter novidades). No total são 234 classificações utilizando o detector de novidades. Abaixo pode-se observar na figura 26 uma curva de teste original obtida dos dados e na figura 27 uma curva modificada pra conter novidades. As modificações para inserir novidades foram arbitrárias e diferentes para cada curva, mas as mesmas em cada uma das 13 bateladas. A figura 27 contém um degrau inserido para ser a novidade. Outras novidades inseridas incluem degraus em outras posições e amplitudes e soma da curva com formas de ondas arbitrárias, como a senoide. Os resultados obtidos da execução de todas as classificações estão sintetizados na tabela 7.

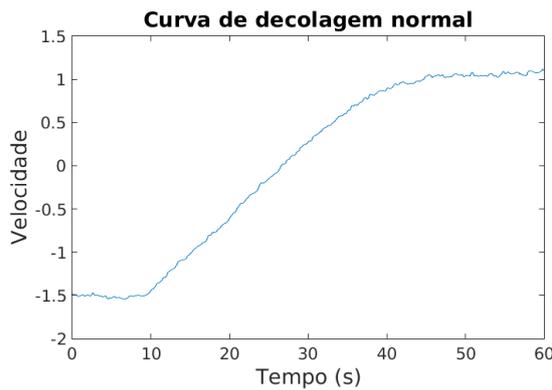


Figura 26 – Curva de teste normal.

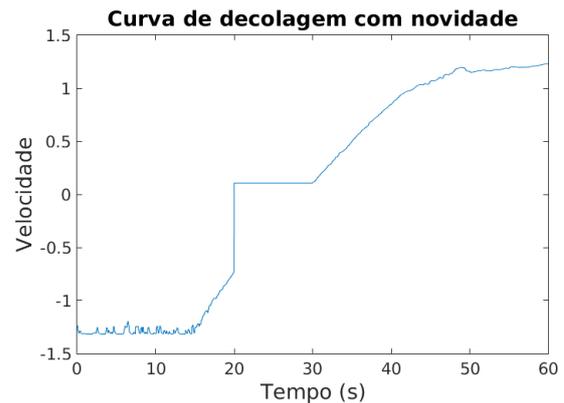


Figura 27 – Curva de teste com novidade.

Número de classificações totais:	234
Número de classificações corretas:	230 (98,29%)
Número de falsos positivos:	4
Número de falsos negativos:	0

Tabela 7 – Resultados da prova de conceito do detector de novidades.

Com esses dados conclui-se que o detector de novidades funciona bem para o caso proposto, com uma taxa de acerto de 98% para detectar novidades. Os falsos positivos encontrados mostram que em alguns casos o detector acusou uma novidade em uma curva que originalmente deveria ser considerada normal. Isso significa que o limite escolhido para a detecção de novidades poderia ser relaxado. O fato de que nenhum falso negativo ser detectado é importante, pois mostra que nenhuma vez ocorreu o caso de uma curva com

novidades ser classificada como normal. Para aplicações de alta confiabilidade como a que estamos estudando esse fator é muito importante. A alteração do limite para diminuir os falsos positivos deve levar isso em conta e ser cuidadosamente ajustado caso uma aplicação futura tente melhorar a performance do algoritmo.

5.3 Análise de votadores complexos

A seção atual apresenta os resultados obtidos das simulações realizadas com os votadores complexos, um deles proposto nesse trabalho e outro selecionado como padrão da indústria.

O votador proposto traz como novidade para o ambiente aeronáutico a utilização de reconhecimento de padrões baseado em aprendizado de máquina para identificar falhas em sensores de velocidade e com isso melhorar a confiabilidade de Air Data Systems, que atualmente não utilizam tal técnica em sua composição. O votador é treinado com dados típicos de decolagem e então durante sua operação analisa se os dados lidos pelos sensores batem com o aprendido durante o treinamento. Se um desvio significativo é detectado o sensor é marcado como falhado e isolado, melhorando a qualidade da medida do ADS.

Espera-se que a utilização desse novo votador melhore o desempenho dos sistema de medição de airspeed na presença de sensores com falhas. A comparação entre o erro do votador proposto e o erro do votador padrão em um mesmo cenários de falha irá dar um indicativo sólido do desempenho dos dois e será possível avaliar se a técnica proposta traz ou não ganhos.

5.3.1 Pré análise dos resultados

A primeira etapa da análise dos resultados das simulações dos votadores complexos é semelhante ao realizado para os votadores simples. O objetivo é analisar se existem indícios da influência da variação dos parâmetros nos resultados obtidos. Será analisado de maneira isolada se a variação da quantidade de falhas, tipo de erro e tipo de entrada alteram o desempenho relativo dos votadores.

O detector de novidades funciona após a decolagem ter terminado, analisando os 60s de dados anteriores a esse momento. O votador padrão também precisa de um certo período, que varia dependendo de sua configuração e da técnica de detecção de erro, para detectar uma falha. Dessa maneira faz sentido restringir a comparação do desempenho dos dois votadores complexos ao período posterior a esse instante, nesse caso o período será restrito aos 10s após o término da decolagem. Para efeitos de nomenclatura, o votador padrão também poderá ser chamado de votador 1 e o votador com detecção de novidades de votador 6.

A primeira análise é como o desempenho dos votadores é influenciado pela quantidade de falhas presentes no sistema. A figura 28 mostra o erro médio para essa situação. É possível ver que o votador 6 é melhor para os casos de 1 e 2 falhas, incluindo *split*, e o votador 1 é melhor para 3 falhas. De maneira semelhante para ambos, o caso com 1 falha possui erros bem menores do que os outros. O votador 1 tem seu melhor desempenho para 1 falha, com erro médio de 3,86% e pior para 2 falhas *split*, com erro médio de 210,42%. Já o votador 6 tem seu melhor desempenho também para 1 falha, com erro de 1,02% e pior para 3 falhas, 154,04%. A figura 29 mostra o erro máximo para os mesmos cenários. Os valores são diferentes mas a comparação entre os votadores é a mesma.

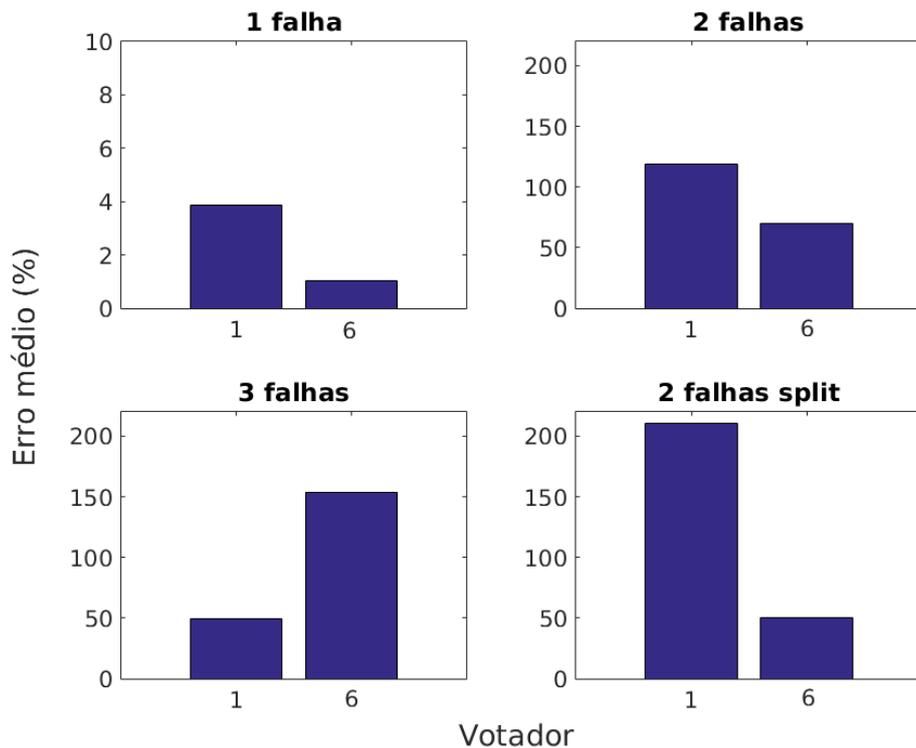


Figura 28 – Erro médio para diferentes quantidades de falhas.

Com esse comportamento observado em mente nota-se que o caso com 1 falha é facilmente contornado por ambos os votadores, independente de outros fatores como tipo ou amplitude. Isso muda bastante para maior número de falhas.

Em seguida é feita uma análise de como o tipo de erro influencia no desempenho dos votadores. A figura 30 mostra o erro médio e a figura 31 o erro máximo para diferentes tipos de relação entre o erro no sensor e a entrada. É possível ver que o votador 6 teve um desempenho melhor para os casos de erro derivativo, constante e rampa, com o menor erro médio de 1,5% para o caso derivativo. O votador 1 foi melhor para o erro proporcional, com valor de 29%. Os piores valores de erro foram 268% para o votador 1 com o erro derivativo e 253% para o votador 6 com o erro proporcional. Os erros máximos tem comportamentos semelhantes em relação a qual votador tem melhor desempenho em cada caso.

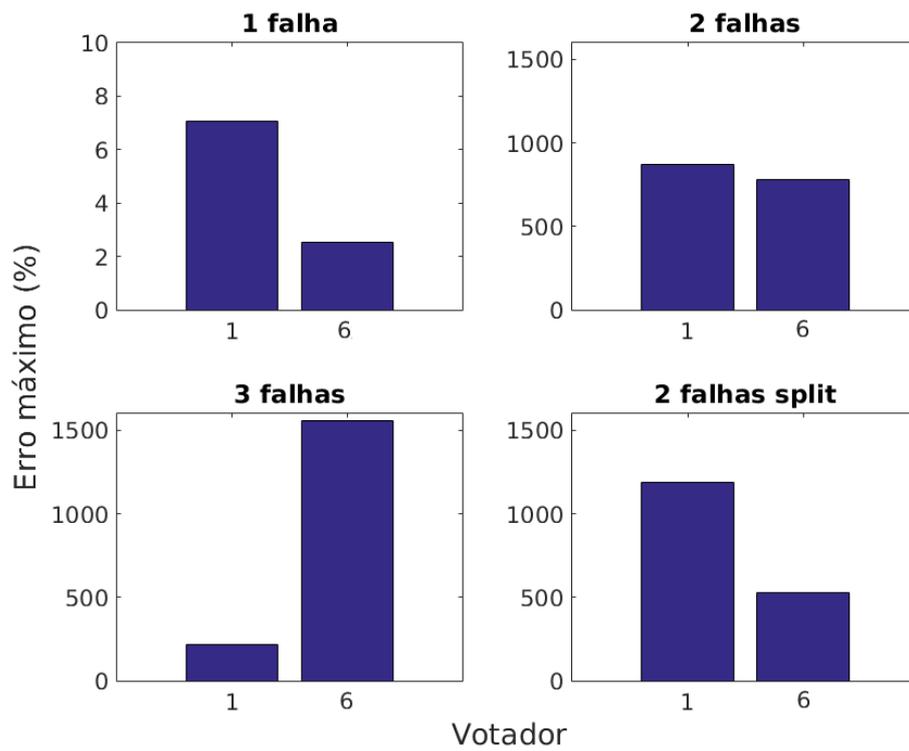


Figura 29 – Erro máximo para diferentes quantidades de falhas.

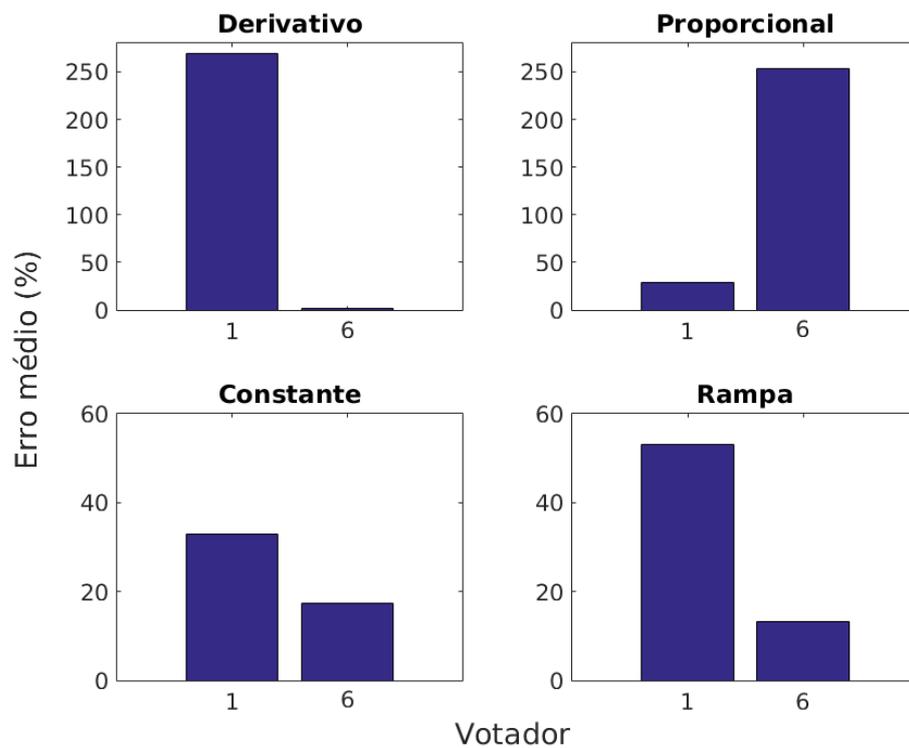


Figura 30 – Erro médio para diferentes tipos de falhas.

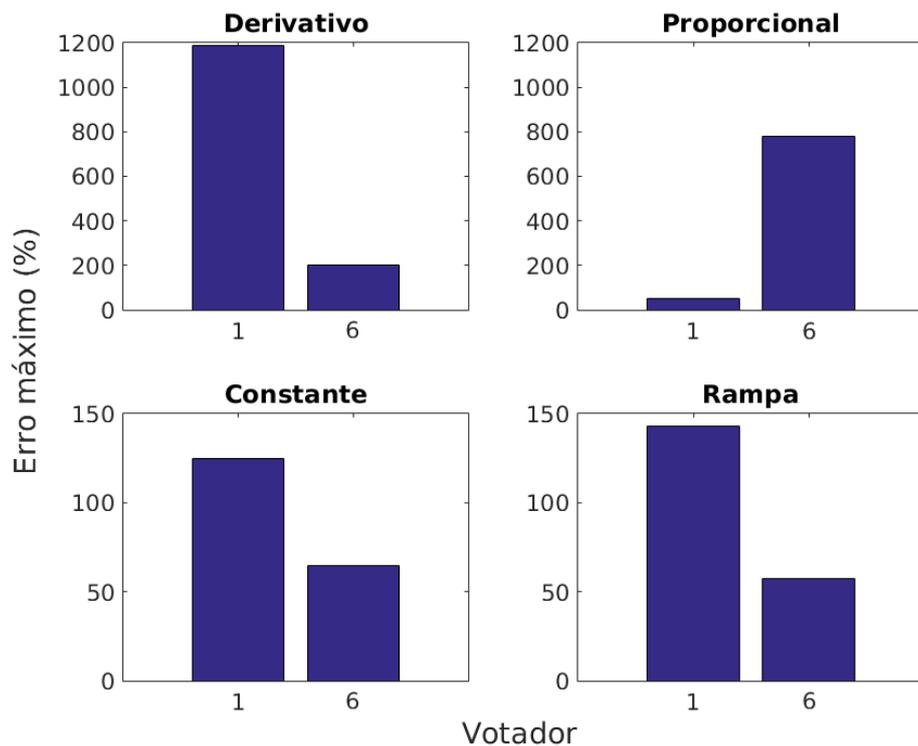


Figura 31 – Erro máximo para diferentes tipos de falhas.

É importante destacar que a mudança de comportamento dos votadores ao alterar o tipo de falha foi muito aguda, maior inclusive do que observado ao alterar a quantidade de falhas. Especialmente a disparidade entre os comportamentos considerando o erro proporcional e o erro derivativo. Isso é um indício forte de que o comportamento da falha no sensor altera significativamente a capacidade do votador de detectá-la. Especificamente pode-se dizer que o votador 6 é particularmente ruim em detectar falhas do tipo proporcional e que o votador 1 é igualmente ruim para falhas derivativas.

Analisando como o tipo de entrada influencia o desempenho dos votadores as figuras 32 referente ao erro médio e 33 referente ao erro máximo são geradas. Existe uma leve diferença no comportamento, com o votador 6 sendo melhor para entradas em rampa e oscilatórias, e pior para entradas rotacionais. Ainda sim os valores de erro são próximos (entre 50% e 120%) nos votadores para esses casos, o que leva a conclusão de que o tipo de entrada não é um fator relevante para o desempenho da votação aplicada a esse tipo de sensor.

Considerando as análises variando número de falhas, tipo de falha e tipo de entrada percebe-se que existem indícios de que o votador proposto nesse trabalho é melhor em alguns casos em comparação com o padrão. Além disso, as diferenças observadas ao variar o tipo de erro são muito grandes, indicando que esse fator tem um grande papel na performance dos votadores. A quantidade de falhas também influencia o desempenho, mas

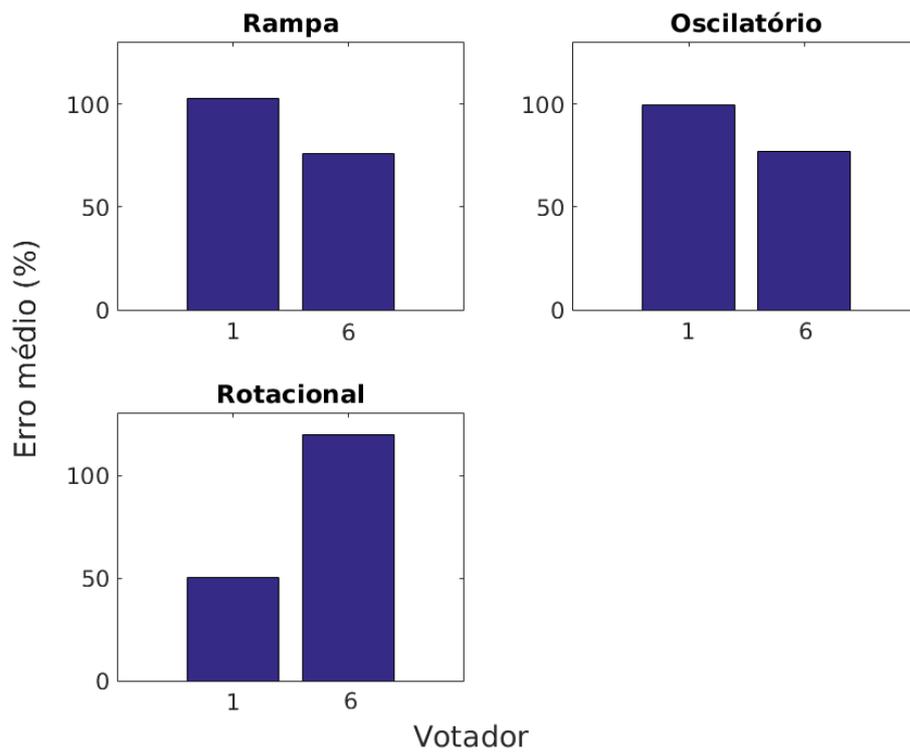


Figura 32 – Erro médio para diferentes tipos de entradas.

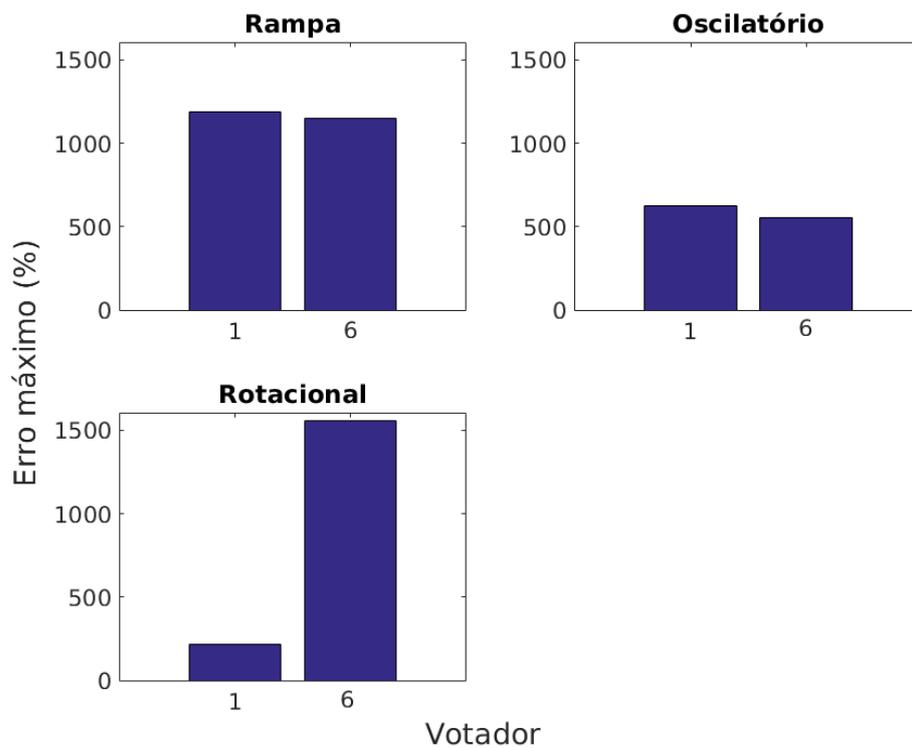


Figura 33 – Erro máximo para diferentes tipos de entradas.

de maneira menos aguda por causa de dois fatores: primeiro que no caso de uma falha os dois votadores têm desempenhos muito parecidos e muito bons e segundo que o caso com duas falhas em *split* é especial pois o erro não conta toda a história do desempenho do votador 1, comportamento que é analisado na seção 5.3.2. Por último, o tipo de entrada tem pouca influência no desempenho e é possível raciocinar que o motivo para isso está na identificação de falhas dos votadores complexos. Tal identificação é realizada primariamente durante o período de decolagem, sendo que o tipo de entrada interfere nos dados somente após esse período.

A primeira vista pode parecer estranho o valor alto dos erros mostrados já que a prova de conceito da detecção de novidades obteve mais de 98% de classificações corretas, mas existem elementos distintos entre os dois experimentos. Na prova de conceito as novidades inseridas foram arbitrárias. Na simulação do ADS para a comparação dos votadores complexos, os erros de medição dos sensores são consequência das falhas inseridas em seus modelos e portanto bem diferentes das novidades da prova de conceito. Isso significa que temos erros representativos da operação do sensor e que o desempenho do detector nessa situação é diferente. Erros grandes significam novidades não detectadas, e o motivo pode variar. O caso identificado de desempenho ruim para erro proporcional é discutido em detalhes na seção 5.3.3.

5.3.2 Análise de duas falhas *split*

O comportamento do votador 1 para duas falhas em *split* merece ser estudado em detalhe. Esse tipo de falha ocorre quando os quatro sensores fornecem medidas que se organizam de maneira semelhante a dois pares. Em cada par duas medidas estão próximas e concordam entre si. Entre os diferentes pares os valores são diferentes e distantes. Essa situação é uma das mais difíceis para o votador, pois é virtualmente impossível saber qual dos pares está correto tendo como informação apenas os valores medidos naquele momento.

Devido a extrema dificuldade em saber qual é o valor correto nessa situação o design do votador 1 tem uma técnica de detecção de erro exclusiva para identificar essa condição de separação em dois pares. Mesmo observando e detectando o comportamento ainda não é possível saber qual o par correto. Por isso, nessa situação, o votador 1 tem por design o comportamento de congelar o valor de saída no último valor válido e sinalizar que o seu valor votado não é mais confiável.

O funcionamento do votador 1 nessa condição pode ser observado com detalhes nos gráficos da figura 34. O primeiro gráfico mostra os quatro valores lidos dos sensores. Na condição separada para análise dois dos sensores possuem falhas de mesma direção, causando a separação das medidas em dois pares, o que pode ser observado. O segundo gráfico mostra a saída do votador, que é congelada perto dos 18s, quando ele identifica

que há uma condição de *split*. Em vermelho no mesmo gráfico o erro entre o valor votado e a referência aumenta a medida que o tempo passa porque os valores dos sensores estão aumentando e o valor consolidado do votador permanece estático. No terceiro gráfico é possível ver a saída do votador que sinaliza a validade do seu valor votado. No instante em que ele identifica o *split* e congela a saída a validade vai de True para False, informando que o valor não é mais confiável.

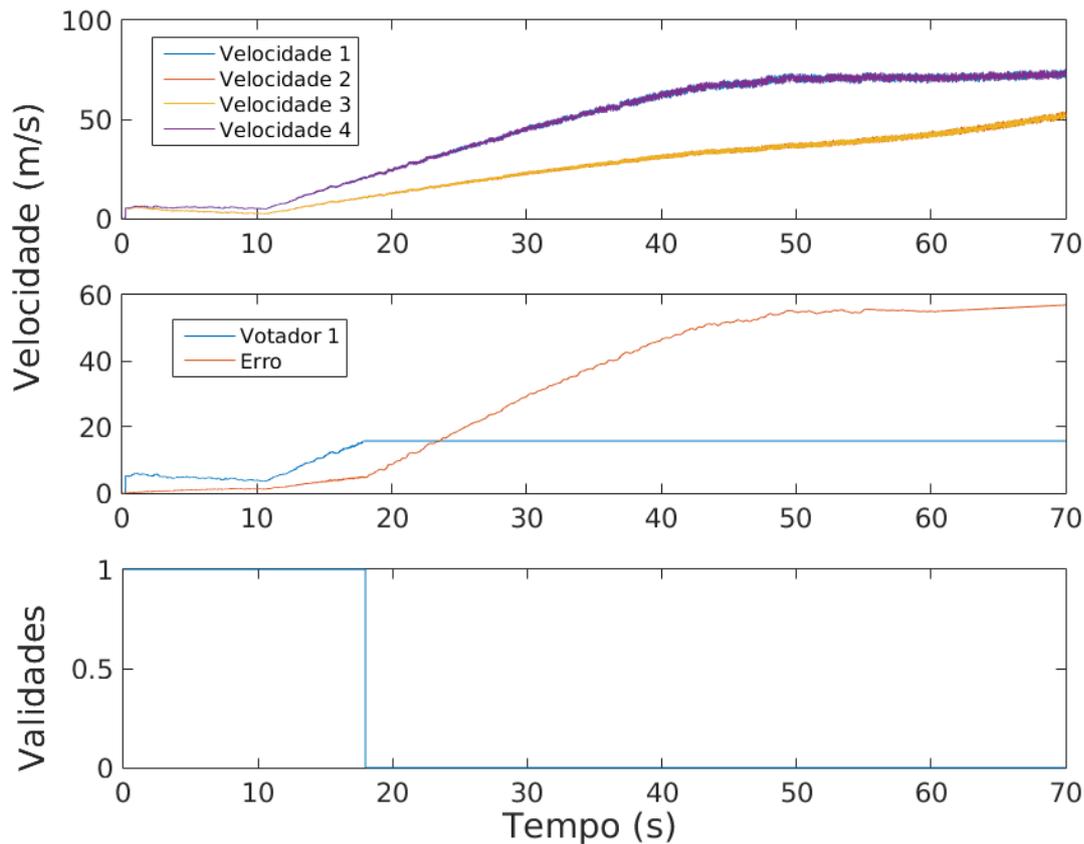


Figura 34 – Valor dos sensores e saídas do votador 1 para um caso de *split*.

Esse comportamento do votador 1 causa os resultados observados na seção 5.3.1 em que ele possui um erro alto para duas falhas em *split*. Apesar da sinalização da falha ocorrer o fato de que o valor votado continua errado permanece e isso aparece nos resultados.

A comparação com o votador 6 é inevitável e a figura 35 mostra suas saídas para o mesmo caso em que o votador 1 foi analisado. A detecção de erro do votador proposto funciona independentemente para cada sinal, comparando a curva obtida em cada sensor com a esperada para uma decolagem e detecta uma novidade se houver falha naquele sensor, removendo-o do cálculo do valor votado. O caso de *split* é sentido pelo votador 6 como duas novidades em dois sensores independentemente, como pode ser observado no segundo gráfico da figura, em que dois detectores de novidade são ativados no tempo

de 60s. Os valores dos sensores em que as novidades são detectadas são removidos e o valor consolidado é calculado com base nos sensores sem falhas. O primeiro gráfico da figura mostra exatamente isso acontecendo, com o valor de saída sendo reajustado e o erro caindo quando a novidade é detectada.

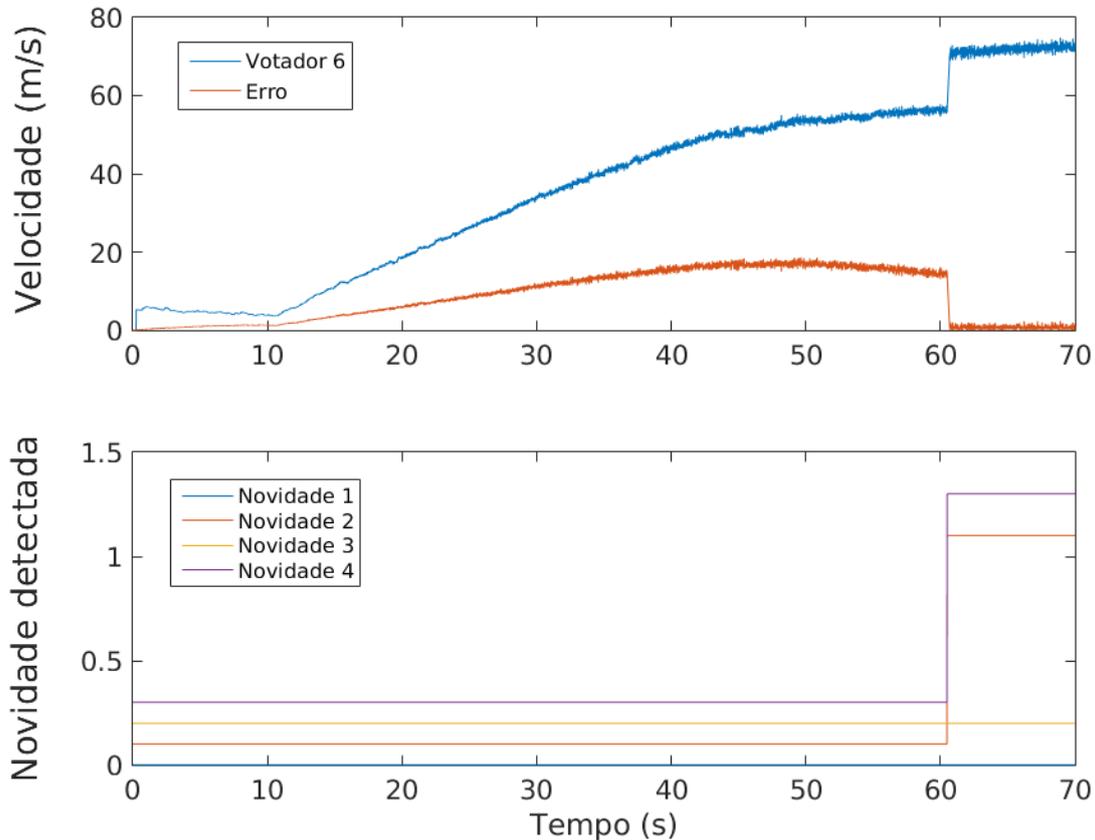


Figura 35 – Valor das saídas do votador 6 para um caso de *split*.

Relembrando a análise feita na seção 5.3.1, o erro médio para *split* na figura 28 é um indicativo de que o votador 6 tem um desempenho melhor do que o votador 1 para o caso de *split*, ou seja, ele consegue identificar pelo menos em alguns casos os sensores com falha nessa situação.

5.3.3 Análise do erro proporcional no detector de novidades

Devido ao desempenho ruim do votador 6 para o caso de erro proporcional, uma investigação mais detalhada sobre seu comportamento é valiosa. A figura 36 mostra o cenário de falhas proporcionais de grande amplitude inseridas em 3 dos 4 sensores.

Analisando o segundo gráfico é possível ver que o valor votado está acompanhando os sensores com falha e o erro em relação a referência se torna muito grande. No terceiro

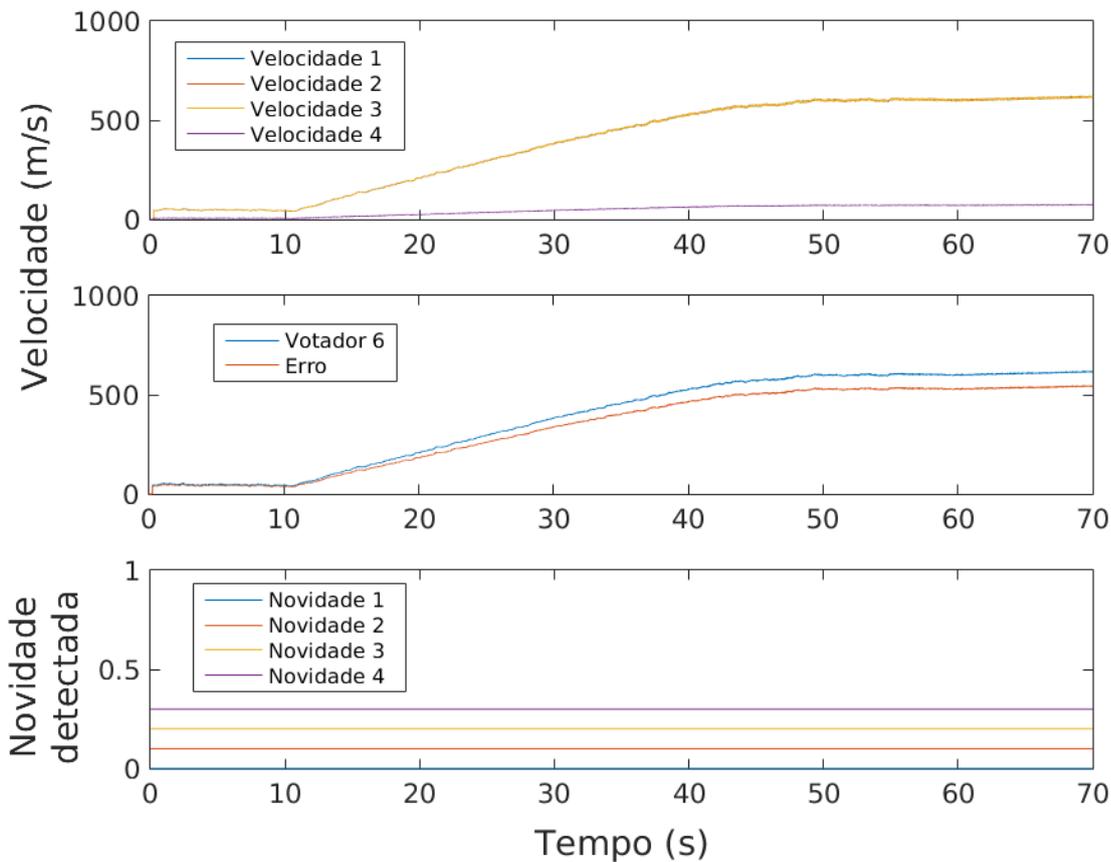


Figura 36 – Valor dos sensores e saídas do votador 6 para um caso de erro proporcional.

gráfico fica claro o porquê: apesar do valor do erro ser muito grande durante a etapa de decolagem, nenhuma novidade é detectada pelo votador 6. Isso está dentro do esperado após ver os resultados desse votador para erros desse tipo.

O motivo da detecção de novidades não funcionar para esse caso é que uma das etapas incluídas no seu processo é a normalização da curva de entrada. Dessa maneira é possível dizer que a detecção de novidades está muito mais interessada na forma da curva do que em seus valores absolutos. A vantagem é que decolagens que têm diferentes valores finais de velocidade mas o mesmo perfil são consideradas adequadas, ao mesmo tempo como efeito colateral um erro proporcional em relação ao valor da entrada gera uma curva que está diferente do esperado mas que não é identificada como tal. Pode-se concluir que efetivamente esse tipo de falha não é adequadamente identificado e corrigido pelo detector de novidades.

5.3.4 Análise estatística

Nesta etapa o objetivo é responder a seguinte pergunta: "O votador 6 (proposta) tem um desempenho melhor do que o votador 1 (padrão)?" Para fazer isso será levado em conta a análise feita na seção 5.3.1.

Como há indícios de que o tipo de erro tem uma grande influência no desempenho dos votadores, a pergunta sobre qual votador tem melhor desempenho é separada por esse fator e a análise é feita separadamente para cada um dos tipos de erro. Além disso, há indícios de que o votador 6 é melhor para erros do tipo derivativo, constante e rampa, e que o votador 1 é melhor para erro do tipo proporcional. Também é levado em conta que o erro padrão para o sensor de fluxo térmico é 4%, como já exposto anteriormente, o que significa que só faz sentido dizer que um ou outro votador é melhor se a diferença entre eles for maior do que esse valor, considerado a mínima significância. Assim, as seguintes hipóteses são definidas:

Erro derivativo, constante e rampa

- H_0 (hipótese nula): O erro médio do votador 1 não é maior do que o erro médio do votador 6 em pelo menos 0,04 (representando 4%).
- H_1 (hipótese alternativa): O erro médio do votador 1 é maior do que o erro médio do votador 6 em pelo menos 0,04 (representando 4%).

Erro proporcional

- H_0 (hipótese nula): O erro médio do votador 6 não é maior do que o erro médio do votador 1 em pelo menos 0,04 (representando 4%).
- H_1 (hipótese alternativa): O erro médio do votador 6 é maior do que o erro médio do votador 1 em pelo menos 0,04 (representando 4%).

Considerando a separação das 192 amostras em 4 grupos por tipo de erro, cada subgrupo tem 48 amostras. Os testes estatísticos são realizados em cada subgrupo separadamente. Devido a um número de amostras (N) maior do que 30, segundo o teorema do limite central, é possível considerar verdadeira a premissa de que as médias das amostras segue uma distribuição normal mesmo que as amostras não tenham essa distribuição. Isso permite a utilização de testes paramétricos para a comparação (CAMPELO, 2018).

Outra informação de muita relevância é que as amostras são pareadas, o que significa que para uma configuração específica de execução (entrada, erro, etc) existem duas amostras que estão relacionadas, uma do votador 6 e uma do votador 1.

Considerando esses elementos expostos foram realizados t-test para testar as hipóteses apresentadas (CAMPELO, 2018). Os dados de entrada do t-test são as diferenças

dos erros dos votadores, considerando o pareamento. O nível de significância escolhido é α igual a 0,05 e o intervalo de mínima significância é δ igual a 0,04. O número de amostras (N) é 48 para cada teste. Devido às hipóteses considerarem um valor ser maior que outro o teste é feito somente para um dos lados (unilateral), observando a cauda da direita.

Os testes foram executados no matlab com as entradas e parâmetros descritos. Para maiores detalhes sobre o script utilizado o anexo A explica com mais detalhes essa etapa. Os resultados obtidos estão listados abaixo.

Erro derivativo

Há evidências suficientes para rejeitar a hipótese nula H_0 e aceitar a hipótese alternativa H_1 de que o erro médio do votador 1 é maior do que o erro médio do votador 6 em pelo menos 0,04 (M=2,67, SD=3,37). Valor $t(47)=5,41$, $p<0.001$. Intervalo de confiança (1,86; inf).

Erro constante

Há evidências suficientes para rejeitar a hipótese nula H_0 e aceitar a hipótese alternativa H_1 de que o erro médio do votador 1 é maior do que o erro médio do votador 6 em pelo menos 0,04 (M=0,22, SD=0,28). Valor $t(47)=4,38$, $p<0.001$. Intervalo de confiança (0,15; inf).

Erro rampa

Há evidências suficientes para rejeitar a hipótese nula H_0 e aceitar a hipótese alternativa H_1 de que o erro médio do votador 1 é maior do que o erro médio do votador 6 em pelo menos 0,04 (M=0,14, SD=0,24). Valor $t(47)=3,00$, $p<0.001$. Intervalo de confiança (0,09; inf).

Erro proporcional

Há evidências suficientes para rejeitar a hipótese nula H_0 e aceitar a hipótese alternativa H_1 de que o erro médio do votador 6 é maior do que o erro médio do votador 1 em pelo menos 0,04 (M=1,96, SD=3,33). Valor $t(47)=3,99$, $p<0.001$. Intervalo de confiança (1,15; inf).

Principalmente devido ao valor $p < 0.001$ em todos os testes, os resultados sugerem fortemente que o votador 6 proposto nesse trabalho tem um desempenho melhor para erros do tipo derivativo, constante ou rampa em comparação ao votador 1 padrão. Também sugere de maneira enfática que o votador 1 padrão tem um desempenho melhor em comparação ao votador 6 para erros do tipo proporcional.

5.3.5 Análise dos tempos de execução

Os tempos de execução das simulações foi registrado para os dois votadores complexos. Na tabela 8 é possível ver os valores totais e por simulação, considerando que foram realizadas 20 simulações para cada uma das amostras colhidas.

	Votador 1 (padrão)	Votador 6 (proposto)
Tempo de execução total:	16337s	88064s
Tempo médio de execução por simulação:	3,4s	18,35s

Tabela 8 – Tempo de execução dos votadores complexos.

Observa-se que o votador 6 demora 5,4 vezes mais tempo para executar em comparação com o votador 1. Esse resultado, no entanto, não tem muita significância prática. A implementação do votador 6 foi realizada em cima de dois pontos cruciais que tiram a validade da comparação dos tempos de execução. A primeira é que os resultados do treinamento do reconhecimento de padrões foi armazenado em um arquivo em disco que é acessado toda vez que a detecção de novidades é executada. A segunda é que a função de cálculo da distância de mahalanobis não pôde ser compilada junto com o modelo para a execução e o simulink precisa chamar o matlab para a executar externamente.

Esses dois pontos que nada têm a ver com o algoritmo de detecção de novidades têm um impacto muito grande em tempo de execução e impede uma comparação adequada em relação às velocidades dos votadores. Um primeiro passo em um trabalho futuro seria resolver esses pontos e realizar uma comparação de performance temporal adequada.

Ainda sim é possível afirmar que o tempo de execução gasto pelo votador 6 está adequado para implementação em um sistema real. É possível fazer essa afirmação considerando como o pior caso 5,4 vezes mais lento que o padrão. O diferencial dessa técnica é que ela não é executada em todos os ciclos mas somente quando ocorre uma borda de subida na sua entrada de gatilho. Em uma situação real isso ocorreria uma vez após a decolagem e um longo período se passaria até a próxima execução. Seria possível, portanto, executar a detecção de novidades gradualmente através de alguns ciclos após o gatilho ocorrer e ter o resultado com atraso. Para a aplicação em questão isso seria completamente aceitável e provavelmente indistinguível de uma execução mais rápida. As alterações necessárias para que a execução possa funcionar dessa maneira também é uma ideia de trabalho futuro.

6 Conclusão

Existe uma demanda de aumento da confiabilidade de Air Data Systems devido a grande criticidade da aplicação desses sistemas na aeronáutica. O objetivo deste trabalho é explorar como fazer isso através de dois caminhos, estudando a integração de sensores de fluxo térmico em um ADS e propondo um novo algoritmo de votação de sinais redundantes que utiliza reconhecimento de padrões para identificar falhas.

Os resultados mostram que a integração de quatro sensores de fluxo térmico redundantes em um ADS permite que o sistema atinja um grau de tolerância a falhas bom para 1 e 2 sensores com defeito ao analisarmos técnicas de votação simples. Para o caso de 3 falhas, o desempenho do ADS é ruim. A técnica de votação que calcula o valor consolidado dos sensores redundantes como a médias dos sensores mais próximos em um intervalo Δ (ASKARI; NOURANI; NAMAZI, 2011) tem o melhor desempenho dentre as votações comparadas e é a responsável pelo grau citado de confiabilidade. O erro médio para esse votador nos casos de 1 e 2 falhas ficou próximo de 4,5%, um pouco acima do valor de 3% definido pelo requisito aeronáutico. A simulação de defeitos específicos de sensores de fluxo térmico permite obter uma resposta significativa dos efeitos e comportamentos do sistema para essa categoria de sensores. É possível ver que devido às características desse sensor ele ainda não está maduro para atender os requisitos aeronáuticos desse tipo de medida atuando isoladamente, embora esteja perto. Ele pode ser utilizado em conjunto com tubos de *pitot* padrão para aumentar a confiabilidade do sistema como um todo ou ainda ser fisicamente melhorado para diminuir seu erro padrão de 4%.

O desenvolvimento de uma nova técnica de votação complexa passa pela prova de conceito da detecção de novidades (ALVES; TORRES, 2016) em dados reais de decolagem de aviões. Esse estudo nos permite concluir que a técnica é aplicável para o cenário proposto, com uma taxa alta de classificação correta (98,29%) e nenhum falso negativo.

O votador complexo desenvolvido utiliza a detecção de novidades durante a decolagem para identificar sensores com falhas e os removerem do cálculo do valor consolidado de airspeed. Seu desempenho integrado em um ADS com sensores de fluxo térmico é comparado ao desempenho de outro votador complexo (OLIVEIRA; GALVÃO, 2011) padrão para esse sistema. Conclui-se dessa comparação que para esse tipo de sensor e falhas características o votador proposto tem desempenho superior para erros que são derivativos, constantes ou crescentes em relação à entrada e o votador padrão tem desempenho superior para erros que são proporcionais à entrada. O custo computacional dos dois votadores complexos foi analisado apenas superficialmente, sendo que o votador proposto tem vários pontos óbvios de melhoria. Nas simulações desse trabalho ele se mostrou 5,4 vezes mais

lento que o padrão.

Esse resultado para o votador proposto é muito positivo pois indica que há espaço para melhorar a confiabilidade dos ADS atuais e que técnicas de reconhecimento de padrões são uma alternativa viável para tal. O fato de que ele não é superior em todos os cenários em comparação com o votador padrão é um reflexo de como esse tipo de votador complexo funciona: utilizando várias técnicas de detecção de erros em paralelo para aumentar as chances de identificar quais sensores estão saudáveis e quais apresentam defeitos. A técnica de reconhecimento de padrões é diferente do empregado comumente e pode muito bem ser utilizada em conjunto com outras técnicas já estabelecidas em um mesmo votador para atingir níveis de confiabilidade maiores ainda do que o demonstrado neste trabalho. Pode-se concluir que o objetivo deste trabalho de caminhar em direção ao aumento da confiabilidade de *Air Data Systems* foi atingido com êxito.

Os trabalhos futuros que podem ser inspirados daqui são muitos. Primeiramente como acabamos de citar, o desenvolvimento de um votador que contenha os pontos positivos do votador proposto nesse trabalho junto com os pontos positivos do votador padrão utilizado para comparação tem potencial de atingir níveis mais altos ainda de confiabilidade para ADS.

Outro ponto que pode ser mais explorado é a utilização do votador simples de média da maioria, que teve o melhor desempenho dentre os votadores simples, como parte de um votador complexo. Os dois votadores complexos analisados nesse trabalho tinham como votação básica interna a mediana das entradas saudáveis.

A implementação do votador proposto em um sistema embutido real, com as devidas otimizações em relação ao tempo de execução e estudo da maneira mais eficiente de salvar o resultado do treinamento da detecção de novidades é um próximo passo desafiador e interessante. Existem pontos simples que podem ser alterados para diminuir seu custo computacional, mas uma alternativa ligeiramente mais complexa para garantir sua aplicabilidade é fatiar sua execução em vários ciclos do sistema. Como a detecção de novidades é executada uma vez por decolagem no cenário proposto, não há impacto que ela demore vários ciclos para concluir, desde que seja o mais preciso possível.

O votador proposto neste trabalho foi treinado para detectar falhas em um tipo de decolagem padrão, mas na prática a curva de velocidade durante a decolagem está sujeita a variações dependendo de decisões do piloto e características do aeródromo. Um estudo de muita relevância para essa aplicação é analisar quais e quantas categorias de decolagem podem ser definidas para serem utilizadas como treinamento do detector de novidades.

Por último, a técnica de detecção de novidades baseada em reconhecimento de padrões se provou útil e poderosa na aplicação desse trabalho, o que estimula a busca por outros sistemas nos quais ela também possa ser utilizada, principalmente na detecção de

falhas em sistemas com redundância.

Referências

- ALVES, E. M. A.; TORRES, F. S. Aumento de confiabilidade de sistemas embutidos usando redundância e algoritmos de decisões baseados em reconhecimento de padrões. *XXI Congresso Brasileiro de Automática - CBA2016*, 2016. Citado 4 vezes nas páginas 34, 36, 42 e 85.
- ANAC. *Regulamento Brasileiro da Aviação Civil 25 - RBAC 25*. [S.l.]: Agência Nacional de Aviação Civil, 2014. Citado 2 vezes nas páginas 27 e 28.
- ASKARI, S.; NOURANI, M.; NAMAZI, A. Fault-tolerant a/d converter using analogue voting. *IET Circuits, Devices & Systems*, 2011. Citado 2 vezes nas páginas 33 e 85.
- BEA. *Accident on 27 November 2008 off the coast of Canet-Plage (66) to the Airbus A320-232 registered D-AXLA operated by XL Airways Germany*. [S.l.]: Bureau d'Enquetes et d'Analyses, 2008. Citado na página 23.
- BEA. *Final Report On the accident on 1st June 2009 to the Airbus A330-203*. [S.l.]: Bureau d'Enquetes et d'Analyses, 2012. Citado 2 vezes nas páginas 23 e 30.
- BISHOP, C. M. *Pattern Recognition and Machine Learning*. [S.l.]: Springer, 2006. Citado na página 36.
- CAMPELO, F. *Lecture Notes on Design and Analysis of Experiments*. 2018. <<https://github.com/fcampelo/Design-and-Analysis-of-Experiments>>. Version 2.12. Creative Commons BY-NC-SA 4.0. Citado na página 81.
- GRACEY, W. *Measurement of Aircraft Speed and Altitude*. [S.l.]: NASA Reference Publication 1046, 1980. Citado 2 vezes nas páginas 28 e 29.
- HAASL, S.; STEMME, G. *Comprehensive Microsystems, Chapter 2.07: Flow Sensors*. [S.l.]: Elsevier Science, 2007. Citado 3 vezes nas páginas 24, 31 e 47.
- MAKINWA, K. A. A.; HUIJSING, J. H. A wind-sensor interface using thermal sigma delta modulation techniques. *Sensors and Actuators A 92 (2001) 280-285*, 2000. Citado 3 vezes nas páginas 31, 32 e 47.
- MAKINWA, K. A. A.; HUIJSING, J. H. A smart wind sensor using thermal sigma-delta modulation techniques. *Sensors and Actuators A 97-98 (2002) 15-20*, 2001. Citado 7 vezes nas páginas 13, 24, 31, 32, 33, 49 e 58.
- NELSON, R. C. *Flight Stability and Automatic Control*. 1. ed. [S.l.]: McGraw-Hill Book Company, 1989. Citado 3 vezes nas páginas 23, 29 e 30.
- OLIVEIRA, L. C. R. de; GALVÃO, R. K. H. A four-signal voting algorithm for aircraft redundant sensors. *21st Brazilian Congress of Mechanical Engineering - COBEM 2011*, 2011. Citado 9 vezes nas páginas 34, 40, 42, 43, 44, 45, 52, 53 e 85.
- RADIO, I. A. *ARINC SPECIFICATION 429 PART 1-17*. [S.l.]: AERONAUTICAL RADIO, INC, 2004. Citado na página 27.

- SAE. *Guidelines and Methods for Conducting the Safety Assessment Process on Civil Airborne Systems and Equipment*. [S.l.]: SAE International, 1996. Citado na página 27.
- SAE. *Guidelines for Development of Civil Aircraft and Systems*. [S.l.]: SAE International, 2010. Citado na página 27.
- TODESCHINI, R. et al. Locally centred mahalanobis distance: A new distance measure with salient features towards outlier detection. *Analytica Chimica Acta*, 2013. Citado na página 36.
- UTC. *SmartProbe® Air Data Systems*. [S.l.]: UTC Aerospace Systems, 2015. Citado 2 vezes nas páginas 13 e 27.
- VIANA, F. A. B.; TORRES, F. S. Reliable integration of thermal flow sensors into air data systems. *VIII Brazilian Symposium on Computing Systems Engineering*, 2018. Citado na página 70.
- WIKI. *Pitot tube types*. [S.l.]: Wikimedia Commons, 2007. Citado 2 vezes nas páginas 13 e 29.

Anexos

ANEXO A – Detalhes da execução

Esta seção descreve os scripts e passos realizados para efetivamente executar a metodologia proposta no capítulo 3 de metodologia e analisar os dados obtidos, utilizando o modelo do ADS explicado na seção anterior 4.1. A prova de conceito do detector de novidades não utiliza o modelo do ADS, mas dados de entrada que são explicados.

A explicação de como a execução é feita cita os arquivos e pastas utilizados durante o desenvolvimento. A pasta raiz com todos os artefatos desenvolvidos é denominada "Models". Nessa pasta se encontra um arquivo "READ-ME.txt" que cita a organização dos artefatos e modelos e pode servir como referencia.

A.1 Prova de conceito do detector de novidades

As entradas utilizadas para a prova de conceito estão em "Models/model_02/inputs/Aircraft_data". A pasta "raw" contém os dados cedidos pela Embraer para utilização, já descaracterizados dos valores originais. Existem dados de dois modelos de aviões, 002 e 003, e cada arquivo .mat dessas pastas é o dado de um voo diferente. Um voo pode conter uma ou mais decolagens, que é a manobra de interesse escolhida para ser usada.

Os script dessa etapa estão em "Models/model_02/02_Novelty_Detection". O primeiro a ser executado é "generate_curves_from_aircraft_data.m" que utiliza os dados em "raw" como entrada para gerar tabelas com as curvas de interesse. O script em questão lê os arquivos de dados procurando por bordas de subida do sinal NLG_UP e mostra cada curva de velocidade nos 60 segundos anteriores a essa transição para o usuário, que classifica manualmente as curvas mostradas entre dois tipos de decolagem bem sucedidas ou como erradas. O resultado da execução do script é uma tabela com todas as curvas analisadas e a classificação de cada uma delas e está armazenado na mesma pasta de inputs, no diretório "selected_curves".

O próximo passo é a execução do script "kfold_validation_NoveltyDetection.m". Ele usa a tabela de curvas classificadas e seleciona somente o tipo de decolagem 1. Em seguida realiza um k-fold com k igual a 9 em um conjunto de 117 curvas. Adicionalmente às 9 curvas sem novidades selecionadas do conjunto de dados original, para cada batelada de validação o scrip inclui 9 curvas com novidades, sendo essas versões modificadas das curvas sem novidades do teste.

O treinamento do detector de novidades para ser utilizado no votador complexo que utiliza essa técnica é feita pelo script "train_NoveltyDetection.m". Esta etapa utiliza 97 curvas escolhidas aleatoriamente para treinamento e reserva 20 curvas para a etapa de

execução, feita posteriormente ao simular o modelo do ADS.

As saídas da prova de conceito estão na pasta "Models/model_02/results/NoveltyTrain". O resultado do k-fold executado está no arquivo "kfold_validation_log.txt" e o resultado do treinamento nos arquivos "NoveltyTrainResult_type1takeoff.mat". Esse arquivo .mat contém os parâmetros de saída do treinamento e as curvas separadas entre treinamento e teste para fácil utilização.

Dentro da pasta de scripts há também o arquivo "RunNoveltyDetection.m", criado para executar a detecção de novidades sob uma curva de entrada utilizando como base o resultado do treinamento. Na etapa posterior de simulação do ADS a detecção de novidade é executada chamando esse script.

A.2 Simulação do ADS

Os arquivos de simulação do modelo ADS estão em "Models/model_02/01_ADS_Model". O modelo principal que contém todas as lógicas descritas em 4.1 é ADS_model.slx. Os blocos referenciados por esse modelo podem ser bibliotecas do Simulink, estando salvas em outros arquivos ou pastas. Portanto todas as outras subpastas presentes dentro de "Models" devem ser preservadas e adicionados no path do Matlab para o funcionamento adequado.

O arquivo "default_ADS_options.m" contém a inicialização de todos os parâmetros necessários para a simulação do ADS em uma estrutura "Options" que deve ser passada para a função que simula o modelo. Para utilizar um valor diferente do padrão para a simulação basta alterá-lo individualmente.

O script "execute_ADS_simulation.m" é o responsável por preparar todo o ambiente e simular o modelo do ADS. Ele não só simula o modelo, mas realiza todo o processo de variar parâmetros de entrada, armazenar os resultados e logs de execução para a obtenção de todos os dados de um votador. Esse script deve ser chamado uma vez para cada votador (simples ou complexo) que se deseja obter dados de desempenho. O fluxo de tarefas executadas pelo script está mostrado abaixo.

- Ler a estrutura "Options" de entrada. A estrutura contém todos os parâmetros necessários para a simulação, incluindo em forma de lista os parâmetros que devem ser variados para obtenção dos dados de um votador (tipo, valor e frequência da entrada, tipo, valor e quantidade de falhas).
- Registrar o log de execução ".txt" com todos os passos do script, incluindo tempo de execução. Para possibilidade de reprodutibilidade a semente do gerador pseudo aleatório é registrada.

- Para cada combinação única de parâmetros variáveis, executar a simulação do ADS 20 vezes. Isso é feito chamando a função "single_model_run" do próprio arquivo 20 vezes. Em seguida fazer a média simples dos valores de saída das 20 simulações através da função "calc_consolidated_simOut".
- Armazenar o resultado da simulação para cada uma das combinações únicas de parâmetros em um arquivo ".mat"

Após execução a função presente no arquivo "get_ADS_result.m" consegue extrair e consolidar os dados obtidos. Passando para ela uma estrutura contendo os resultados do passo anterior e a opção desejada é possível: obter os dados temporais de uma saída; fazer a média de uma saída para um range definido de combinações de parâmetros; plotar uma saída. Essas opções são exploradas na etapa de análise dos dados.

O arquivo "ADS_simulations_calls_and_analysis.m" contém trechos de códigos utilizados para realizar as simulações propostas, além de trechos utilizados para debug e testes. Um trecho desse arquivo mostra a maneira mais simples de executar uma simulação para um votador:

```
Options = default_ADS_options();  
Options.VoterSelection = 2;  
execute_ADS_simulation(Options);
```

A.2.1 Técnicas de votação simples

Para a coleta de dados dos votadores simples a chamada presente no arquivo "ADS_simulations_calls_and_analysis.m" é utilizada. Como é possível observar lá, são feitas quatro chamadas para a função "execute_ADS_simulation.m" uma vez para cada votador simples.

Os resultados estão localizados na pasta "Models/model_02/results/<data>_ADS_model_full_execution". Em diferentes pastas, dependendo da data, pode-se encontrar os resultados para os votadores simples, denominados Votador 2, Votador 3, Votador 4 e Votador 5.

A.2.2 Técnicas de votação complexa

No arquivo "ADS_simulations_calls_and_analysis.m" há uma seção dedicada para a chamada da execuções das simulações para comparar os votadores complexos. Há uma diferença para esse tipo de votador, porque como explicado na seção 3.4, eles utilizam como entrada curvas de velocidade reais de uma decolagem.

A diferença na preparação da execução consiste em abrir o arquivo que contém o resultado do treinamento da detecção de novidades e selecionar as curvas separadas para teste desse arquivo. As opções de simulação são alteradas para receberem essas curvas como entrada e as utilizarem na execução do modelo. Além disso o tempo da simulação é alterado para 70 segundos e os parâmetros que devem ser variados são alterados para ficarem de acordo com o requerido para os votadores complexos.

Com a opção adequada selecionada, a mesma função "execute_ADS_simulation.m" pode ser chamada e utilizar as curvas de decolagem como entradas. Como essas curvas tem 60s de duração, o tempo após o término da curva é preenchido com um valor constante igual ao último valor da curva adicionado ao tipo de entrada (ex: oscilatória) escolhida para aquela execução.

Os resultados estão no mesmo local dos votadores simples, sendo que os votadores complexos são aqueles de número 1 (votador padrão) e 6 (votador com detector de novidades).

A.2.3 Análise de resultados

Os scripts de análise de resultado estão em "Models/model_02/results". Eles utilizam como entrada os arquivos de resultados gerados nas etapas anteriores. Os scripts estão divididos entre análise dos votadores simples em "SimpleVotersAnalysis.m" e análise dos votadores complexos em "ComplexVotersAnalysis.m".

Ambos os script funcionam de maneira semelhante, carregando os resultados das simulações e em seguida utilizando a função "get_ADS_result.m" para obter os sinais de interesse e gerar os gráficos escolhidos. Essa função foi desenvolvida para ajudar a obtenção e geração dos gráficos com menor esforço, sendo necessário passar para ela como entrada o intervalo de interesse dos parâmetros, o nome do sinal e a operação desejada.

Todos os gráficos mostrados na seção 5 são gerados por esses scripts. Além disso o script de análise dos votadores complexos também realiza os testes de hipótese que são mostrados para essas técnicas ao apresentar os resultados. A seção inicial de cada um deles contém algumas variáveis de controle que podem ser alterada manualmente para controlar com refinamento qual seção do script será executada. Isso é útil para o caso de ser necessário executar somente uma parte da análise ao invés dela inteira, como por exemplo executar somente os testes de hipótese sem re-gerar os gráficos.