

PROJETO E SIMULAÇÃO DE UM SISTEMA DE ESTABILIZAÇÃO, ESTERÇO E WAYPOINT GUIDANCE PARA FOGUETES LANÇADOS DO SOLO

SHIMA, Icaro Ferri¹

TRAIANO, Denner²

RESUMO: Este trabalho tem como objetivo projetar um sistema de estabilização e esterço para foguetes lançados do solo em trajetória descrita por waypoint guidance. A partir do modelo Raytheon foi gerado um código em MatLab, que principalmente rege as características naturais do objeto e ambiente, em seguida um algoritmo em blocos foi confeccionado para funcionar como um computador de voo, recebendo os valores adotados no código e os transformando em operações para alcançar seus objetivos. O dispositivo obteve sucesso no trajeto se deslocando de um ponto a outro com demonstração das fases de ganho de altitude e descida suavizados, com grande contribuição da filtragem Kalman no controlador linear quadrático gaussiano e da escolha de coeficientes do Protocolo Integral Derivativo, sugerindo quais previsões ficam mais próximas do objetivo. Para esse tipo de sistema é de vital importância, afim de extrair máxima performance, um sistema computacional que possa executar previsões e filtragem de ruído na tomada de decisões.

PALAVRAS-CHAVE: Foguetes, Waypoint-Guidance, Estabilização.

ABSTRACT: This essay has the objective of projecting a stabilization and steering system for land launched rockets in a waypoint guidance trajectory. From the Raytheon model, a MatLab code was conceived, it regulates the natural characteristics of the object and the environment, in sequence a block-set algorithm was made to act as a flight computer, inputting values from the code and transforming them into operations to reach its objectives. The device had success in its flight path, moving from one point to the other demonstrating altitude gain and descending phases with smoothness, the Kalman Filter in the linear quadratic gaussian controller and the coefficient selection of the Integral Derivative Protocol, suggesting which predictions closely relate to the objective, contributed greatly for the results. In this kind of system is of vital importance for maximum performance a computational system capable of executing predictions and noise filtration for decision making.

KEYWORDS: Rockets, Waypoint-Guidance, Stabilization.

1 INTRODUÇÃO

Em conformidade ao avanço das tecnologias de propulsão, é comum que apesar da diminuição das dimensões dos sistemas a potência e eficiência aumentem de forma a substituir definitivamente as propostas anteriores, o espaço gerado pela redução de proporções pode ser explorado em diversas áreas, seja aumentando a quantidade de propulsores, de combustível ou sistemas de controle (SEIDL, KLEINER; 1999).

Na vertente da implementação de sistemas de controle, dois tipos são os principais utilizados na indústria hoje, o *Vectorial Thrust Control* (Controle de Impulso Vetorial)

¹ Graduando(a) de Engenharia Mecânica no Centro Universitário Campo Real. engm-icaroshima@camporeal.edu.br

² Graduado em Engenharia Mecânica. Mestrado em Engenharia Mecânica. Professor no Centro Universitário do Campo Real. prof_dennertraiano@camporeal.edu.br

que se beneficia de atuadores capazes de movimentar o bocal de exaustão do motor usando a alteração de direção do vetor de impulso para manobrar o dispositivo, e o sistema por superfícies de controle que utiliza de “fins” (barbatanas ou quilhas) móveis atuadas por servo motores que direcionam o fluxo do fluido em que o dispositivo está imerso para realizar manobras (JINGPING, YONGXI, XIAOBO, JING; 2018).

O VTC por um lado, tem sua atividade limitada pelo funcionamento do motor, podendo estar ativo somente durante a fase de impulso. Sua capacidade de esterço é limitada e quase que exclusivamente dependente do grau de liberdade da angulação do bocal e este por sua vez é limitado pela potência do motor. Dadas as proporções, motores com alta capacidade tendem a oferecer mais riscos em acoplamentos móveis. Tanto por falhas quanto perda de atitude por falta de controle, a limitação do ângulo não só restringe as capacidades de manobra mas também as de recuperação (SNOW; 1990).

Como contraproposta o sistema de superfícies de controle permite a manobrabilidade do dispositivo durante toda a sua fase de voo, sendo dependente apenas dos efeitos aerodinâmicos controlados pelas quilhas e que enquanto o dispositivo estiver em voo estes efeitos estarão sempre presentes. Manter um motor de alta capacidade em uma posição fixada possibilita a extração de performance máxima deste com menor risco, as superfícies de controle também oferecem uma gama maior de manobras, mais precisas e bruscas se comparadas ao VTC (KISABO, ADEBIMPE; 2019).

GNC (*Guidance, Navigation and Control*) é um campo de estudo na engenharia de transportes que visa combinar conceitos teóricos de orientação e coordenação, principalmente direcionamento, localização e posicionamento (DE CELIS, CARDOSO; 2019). Sistemas desenvolvidos a partir destes princípios utilizam de sistemas cíclicos compostos principalmente por um computador de voo. Sistemas efetores, corpo do dispositivo e sensoriamento, estes por sua vez trazem a análise de dados e utilizam os sistemas integrados para realizar as operações que possibilitarão a conclusão do objetivo (PELLACANI, GRAZIANO, SUATONI; 2019).

O computador de voo tem seu design voltado inicialmente a captação de resultados desejados (posição, velocidade, atitude), que deverão ser processados pelo algoritmo de orientação. Sendo ele responsável por converter estes em requisições aos algoritmos de controle e filtros, após processados no computador de voo esses dados serão enviados aos sistemas de efeito, nesse caso as superfícies de controle, que

tornarão esses dados em reações físicas, em prol da obtenção dos resultados desejados, o sistema de sensoriamento capta as informações (giroscópio, acelerômetros, barômetros, GPS) e repassa aos filtros de navegação novamente dentro do computador de voo. Para que este avalie novamente se alcançou o resultado esperado ou caso ainda não o tenha feito, quais ações tomar para que esteja cada vez mais próximo (SPEER, JACKSON, RAPHAEL; 2002).

Um dos principais fatores que levam a utilização do sistema de superfícies de controle é que o desenvolvimento desse tipo de dispositivo tem como objetivo o alcance e velocidade máximos. Sendo assim a necessidade de um motor de alta performance com um suprimento elevado de combustível são a combinação chave para alcançar estas determinações. Porém é de extrema importância que o conjunto periférico que auxilia no controle seja igualmente performático (JACKSON, 2010).

Este sistema compacto possibilita a utilização até mesmo de motores a jato e mais de um estágio para a fase *boost*, com a capacidade de manobrabilidade, simplificando alguns processos e tornando outros redundantes o que resulta no pleno funcionamento mesmo em condições severas (JACQUES, STROUBLE; 2010).

A eficiência elevada implícita, acarretada por uma fase de ganho de elevação e velocidade durante o primeiro estágio. Seguido por uma fase *glide* durante um período do trajeto sem o consumo de combustível apenas em aproveitamento da energia cinética e potencial; e por fim um estágio de aproximação do alvo com um segundo *booster* ativo. O qual permite manter o dispositivo com dimensões menores ou até utilizar o espaço em benefício de outras capacidades que não do combustível, reduzindo também a massa do objeto que impacta em sua dinâmica (KIBBEY; 2019).

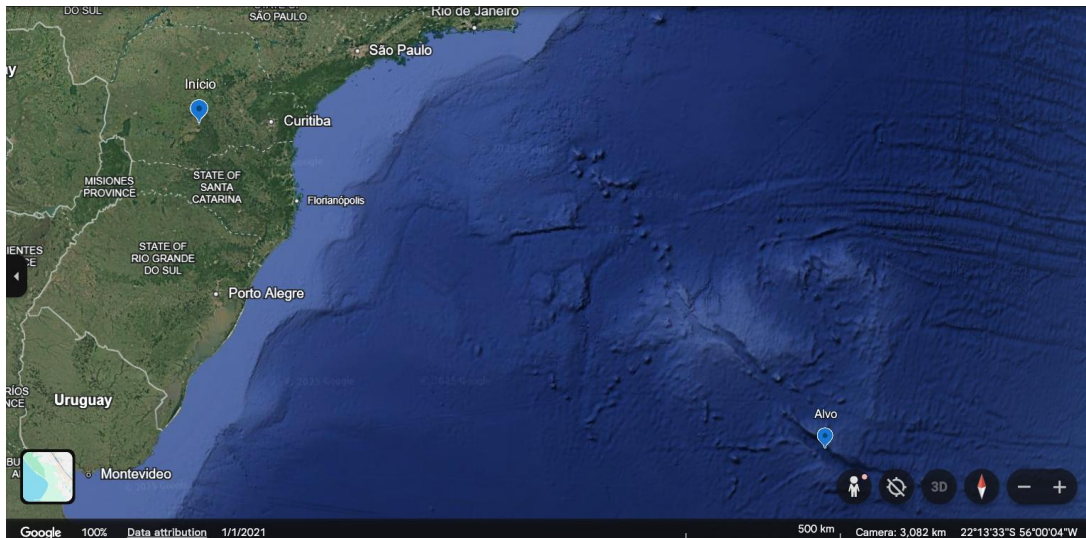
O sistema de estabilização e esterço por superfícies de controle é uma tecnologia que visa implementar um sistema atuante na maior parte do tempo de funcionamento do dispositivo. No desenvolvimento deste trabalho é realizado o projeto de um sistema de superfícies de controle e a realização de sua simulação computacional.

Para os objetivos principais teremos a concepção do código em MatLab dos valores numéricos a serem utilizados na regulamentação do sistema, estruturação do algoritmo em blocos dos sistemas embarcados e a avaliação do funcionamento do sistema de estabilização e esterço por superfícies de controle em uma simulação computacional que demonstra a atividade dos sistemas embarcados, a resposta dos efeitos aerodinâmicos sobre o dispositivo e a trajetória pré-estabelecida por *waypoint guidance*.

2 METODOLOGIA

Neste trabalho foi utilizado a demonstração de um dispositivo lançado de uma posição definida até que atinja um ponto alvo também pré-definido, como mostrado na Figura 1.

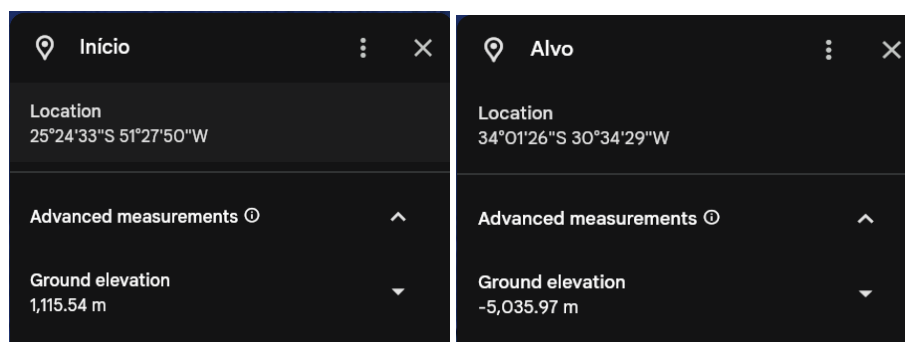
Figura 1 - Pontos de referência Google Maps



Fonte: o autor (2025).

Os pontos escolhidos, são arbitrários e requerem apenas que estejam nos padrões de latitude e longitude do planeta Terra. Descritos conforme seu posicionamento em graus, minutos e segundos se houverem, os quais são apresentados na Figura 2. A altura dos pontos também deve ser informada para que a trajetória identifique o desnível bem como obstáculos entre eles, serão utilizados os valores de 10 metros de elevação para o início e 0 metros para o alvo.

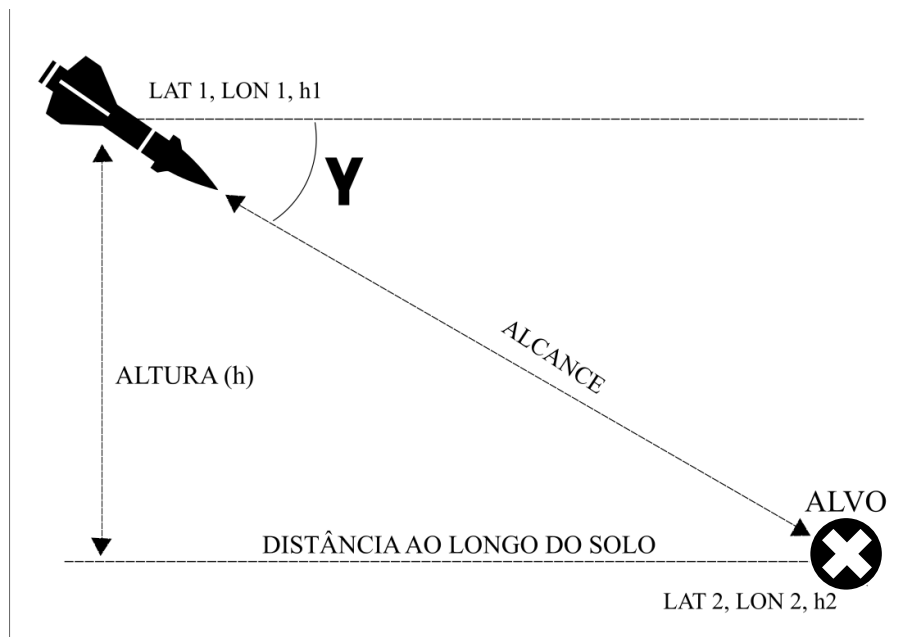
Figura 2 – Destaque dos parâmetros dos pontos



Fonte: o autor (2025).

O modelo matemático utilizado é baseado no modelo Raytheon (MRACEK, RIDGELY; 2005). Este utiliza de operações matriciais para definir o ângulo “ γ ” parametrizado pelo horizonte do dispositivo, Figura 3, e que principalmente dita sua atitude em relação ao solo, utilizando apenas um valor de entrada “ δ_p ”:

Figura 3 – Diagrama de aproximação



Fonte: o autor (2025).

Para o modelo, utiliza-se as equações:

$$X = [\alpha q]^T \quad (1)$$

$$Y = [Az q]^T \quad (2)$$

$$U = \delta_p \quad (3)$$

$$\dot{X} = AX + BU \quad (4)$$

$$Y = CX + DU \quad (5)$$

Sendo “ α ” o ângulo de ataque, “ Az ” o azimute (ângulo horizontal de direção), “ q ” a taxa de inclinação do dispositivo. “ A ”, “ B ”, “ C ” e “ D ”. São matrizes constantes definidas pelos cálculos das características do dispositivo em conformidade com o modelo.

Um sistema de controle linear-quadrático gaussiano foi responsável por definir as matrizes de estado e dados de entrada, auxiliado por um filtro Kalman que capta

resultados expressamente fora de parâmetro e gera feedback para evitar a execução de comandos prejudiciais.

Sobre o Controle Linear Quadrático Gaussiano (LQG) idêntico ao Regulador Linear Quadrático (LQR) com atuação do filtro Kalman, definimos um sistema com ruído: “n” estados, “m” entradas e “p” saídas. Neste utilizaremos $n = 2$, $m = 1$, $p = 2$.

$$\dot{X} = AX + BU + Gw \quad (6)$$

$$\dot{Y} = CX + DU + Hw + v \quad (7)$$

$$A = n \times n, B = n \times m, G = n \times n, w = n \times 1 \quad (8)$$

$$C = p \times n, D = p \times m, H = p \times n, v = p \times 1 \quad (9)$$

No LQR define-se as equações:

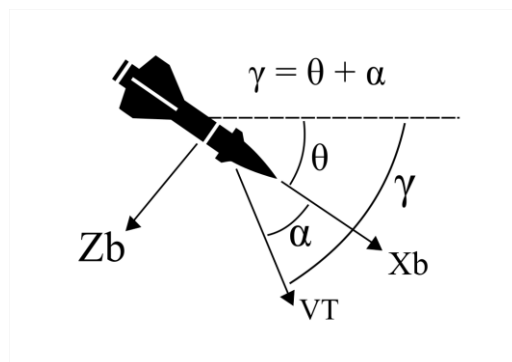
$$Q = n \times n, R = m \times m \quad (10)$$

Sendo estas as matrizes de estado e entrada que atribuem relevância aos valores, e para o filtro Kalman teremos:

$$\bar{Q} = n \times n, \bar{R} = m \times m \quad (11)$$

Que foram covariâncias de ruído, para o design realiza-se ganho do feedback no LQR com “K” e o ganho otimizado no filtro Kalman como “L”.

Figura 4 – Diagrama dos angulos e vetores do sistema



Fonte: o autor (2025).

Redigindo o código no MatLab, apropria-se as matrizes iniciais provenientes do modelo Raytheon (MRACEK, RIDGELY; 2005). Adotando o vetor de velocidade “VT” como a representação do deslocamento em relação solo, “Xb” como a componente vetorial que dita a inclinação do dispositivo (θ), com base em seu horizonte e “ α ” mostra que o ângulo de ataque é formado entre o vetor resultante e a componente vetorial paralela ao dispositivo, fazendo com que atitude seja regulada por este, Figura 5.

Figura 5 – Matrizes com valores atribuídos

```
A = [-1.064 1.000; 290.26 0.00];
B = [-0.25; -331.40];
C = [-123.34 0.00; 0.00 1.00];
D = [-13.51; 0];

states = {'AoA', 'q'};
inputs = {'\delta_c'};
outputs = {'Az', 'q'};

sys = ss(A,B,C,D, 'statename', states, 'inputname', inputs, 'outputname', outputs);
```

Fonte: o autor (2025).

Na Figura 5 observa-se a inclusão dos “estados”, ângulo de ataque (‘AoA’, α), taxa de inclinação (q); “saídas” no azimute (Az) e “entradas”, ângulo (δ). Neste momento também é definido o sistema de matrizes que operam as Equações 1, 2, 3, 4 e 5.

Figura 6 – Declaração dos sistemas do Regulador Linear Quadrático

```
%TF
TFs = tf(sys);
TF = TFs(2,1);
disp(pole(TF));

%MATRIZES DE PESO LQR
Q = [0.1 0; 0 0.1];
R = 0.5;

%GANHO LQR
[K, S, e] = lqr(A,B,Q,R);
fprintf('eigenvalues of A-BK\n');
disp(eig(A-B*K));
fprintf('Feedback gain K');
disp(K);

%SISTEMA LOOP FECHADO
Ac1 = A-B*K;
Bc1 = B;

syscl = ss(Ac1, Bc1, C, D, 'statename', states, ...
'inputname', inputs, ...
'outputname', outputs);
```

Fonte: o autor (2025).

A Figura 6 mostra a confecção dos sistemas vinculados ao LQR e a configuração do dispositivo “K”. Mencionado acima, assim como a integração do formato matricial das Equações 10 e 11, configurando também o funcionamento do “loop” que irá corrigir a estabilidade do dispositivo conforme a operação de subtração das matrizes “A” e “B” com o uso do filtro, parametrizadas de acordo com as equações 8 e 9.

Figura 7 – Parâmetros de ruído no filtro Kalman

```
%TF - LOOP FECHADO
TF = tf(sysc1);
TFc = TF(2,1);

%DESIGN LQG FILTRO KALMAN
G = eye(2);
H = 0*eye(2);

%MATRIZES DE RUIDO KALMAN Q , R
Qbar = diag(0.00015*ones(1,2));
Rbar = diag(0.55*ones(1,2));

%DEFINICAO DO SISTEMA DE RUIDO
sys_n = ss(A , [B G] , C , [D H]);
[kest , L , P] = kalman(sys_n , Qbar , Rbar , 0);

%LOOP FECHADO DO OBSERVADOR DE GANHO KALMAN
Aob = A-L*C;

%DISPLAY DE EIGENVALUES DO OBSERVADOR
fprintf('Observer eigenvalues\n');
disp(eig(Aob));
```

Fonte: o autor (2025).

Como mostrado na Figura 7 define-se uma função de transferência do “loop” que converte sistemas dinâmicos estado-espço em funções de tempo, nesta parte também é necessário designar “L”, mencionado anteriormente, que otimizará os ganhos do filtro no sistema; é possível observar a inclusão das variáveis das Equações 6 e 7 neste momento.

A função computacional “eig” de “eigenvalues” calcula as raízes características de matrizes esparsas reais e simétricas, neste caso utilizado para obter um vetor associado a matriz “A” após o processamento do ganho.

Figura 8 – Parâmetros das constantes

```
%CONSTANTES DE RUÍDO DO TEMPO
dT1 = 0.75;
dT2 = 0.25;

%PARAMETROS DO MISSIL
R = 6371e3; %raio da Terra
Ve1 = 1021.08; %velocidade (m/s)
m2f = 3.2811; %meters to feet

%LOCALIZACAO DO PONTO ALVO
LAT_TARGET = -34.0126;
LON_TARGET = -30.3429;
ELEV_TARGET = 0;

%LOCALIZACAO DO PONTO INICIAL
LAT_INIT = -25.2433;
LON_INIT = -51.2750;
ELEV_INIT = 10;

%LOCALIZACAO DO OBSTACULO
LAT_OBS = -27.0950;
LON_OBS = -49.2374;
```

Fonte: o autor (2025).

Os valores extraídos dos pontos definidos no início integram o sistema. Como visto na Figura 8, aplicando constantes de posição e dados incorporados na telemetria de resposta ao ambiente do dispositivo. Por uma questão de leitura do programa é necessário realizar uma conversão de graus para radianos conforme a Figura 9.

Figura 9 – Conversor de graus para radianos

```
d2r = pi/180; %graus para radianos

%CONVERSOR RADIANS
l1 = LAT_INIT*d2r;
u1 = LON_INIT*d2r;
l2 = LAT_TARGET*d2r;
u2 = LON_TARGET*d2r;

d1 = l2-l1;
du = u2-u1;
```

Fonte: o autor (2025).

Então é possível formular as equações que regem o sistema dinâmico de trajeto até o alvo, a fórmula de Haversine, utiliza o conceito de “versenos” amplamente difundida em navegação para aquisição de valores do raio das cordas para distâncias sobre esferas ou elipses, para o alcance inicial temos o teorema de Pitágoras, o azimute nesse caso é o ângulo de guinada entre o objeto na posição inicial e no alvo e para o ângulo da trajetória de voo temos a tangente inversa da elevação pela distância entre os pontos, na Figura 10.

Figura 10 – Equações de trajetória

```

%FORMULA DE HAVERSINE
a = sin(d1/2)^2 + cos(l1)*cos(l2)*sin(du/2)^2;
c = 2*atan2(sqrt(a),sqrt(1-a));
d = R*c; %distancia horizontal em metros

%ALCANCE INICIAL (TEOREMA DE PITAGORAS)
r = sqrt(d^2+(ELEV_TARGET-ELEV_INIT)^2);

%AZIMUTE INICIAL
yaw_init = azimuth(LAT_INIT,LON_INIT,LAT_TARGET,LON_TARGET);
yaw = yaw_init*d2r;

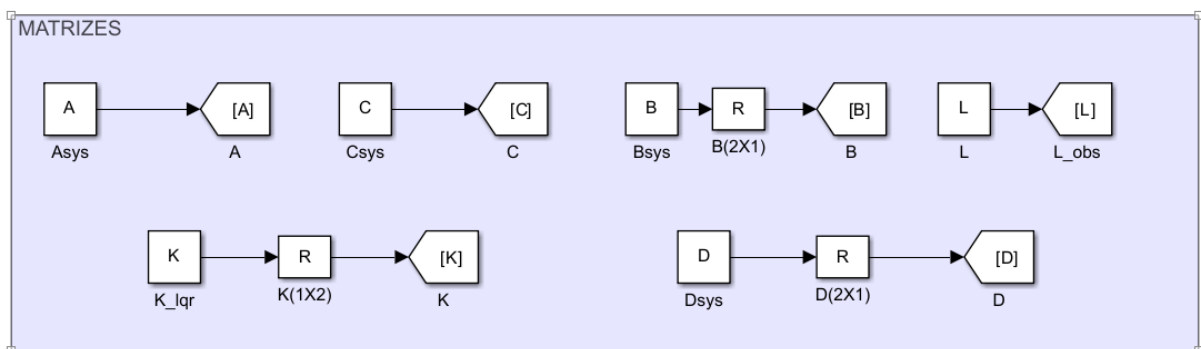
%ANGULO DO TRAJETO DE VOO INICIAL
dh = abs(ELEV_TARGET-ELEV_INIT);
FPA_INIT = atan(dh/d); %radiano

```

Fonte: o autor (2025).

Com o modelo matemático implementado em MatLab, na sequência é feita a montagem do algoritmo em blocos no Simulink. Este, permite utilizar o modelo matemático descrito em MatLab para regular uma simulação dos sistemas embarcados durante a trajetória.

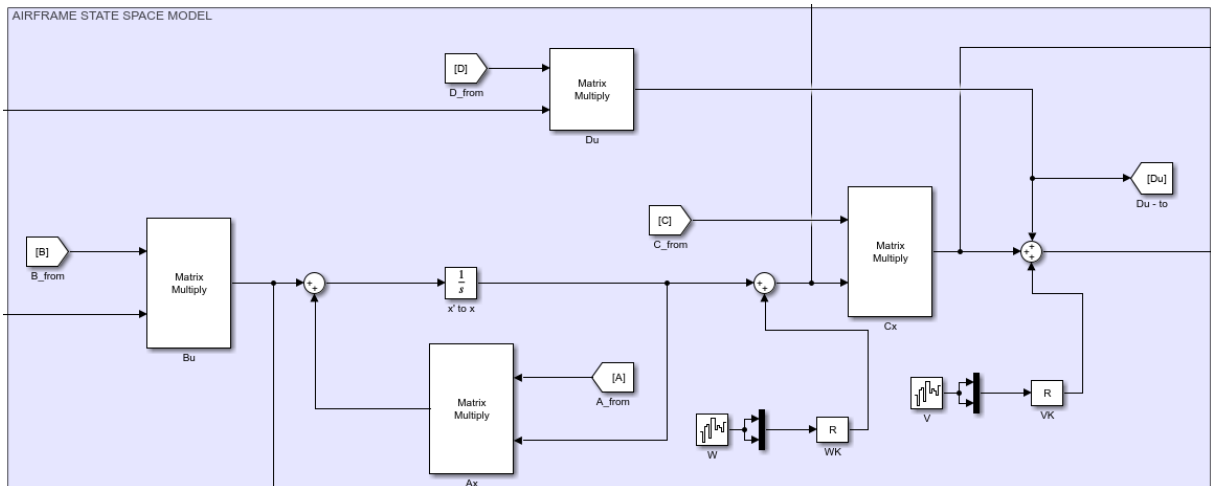
Figura 11 – Blocos das matrizes de entrada



Fonte: o autor (2025).

Na Figura 11, as matrizes são feitas com blocos de constantes e as assimétricas possuem um bloco de formatação que determina as linhas e colunas, seguindo para um bloco do tipo “Go-To” que permite que as matrizes sejam chamadas em outras partes do algoritmo para atribuir valores.

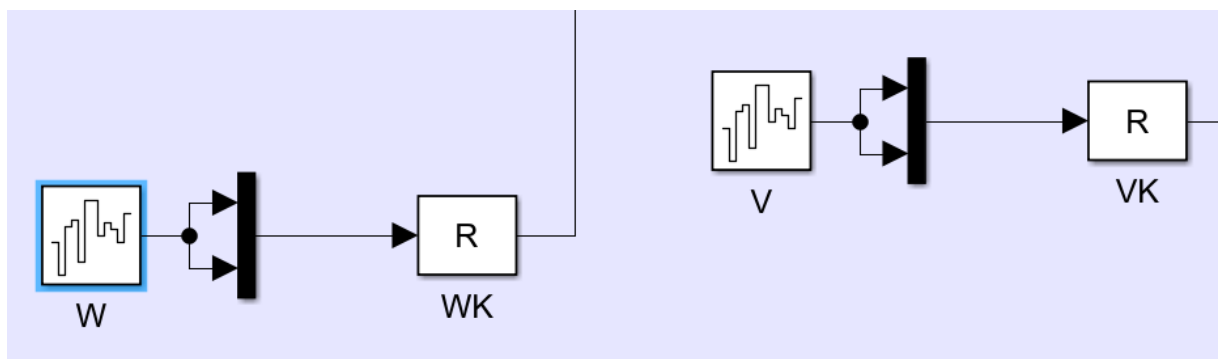
Figura 12 – Modelo Espaço-Estado da fuselagem



Fonte: o autor (2025).

O modelo físico do dispositivo é representado pelas operações das matrizes, Figura 12, definidas pelos parâmetros de design e aspectos de performance deste. Como descrito pelo modelo, blocos do tipo “Matrix Multiply” que são os responsáveis pela multiplicação ou divisão das matrizes em conjuntura com o bloco “Sum” que realiza somas ou subtrações.

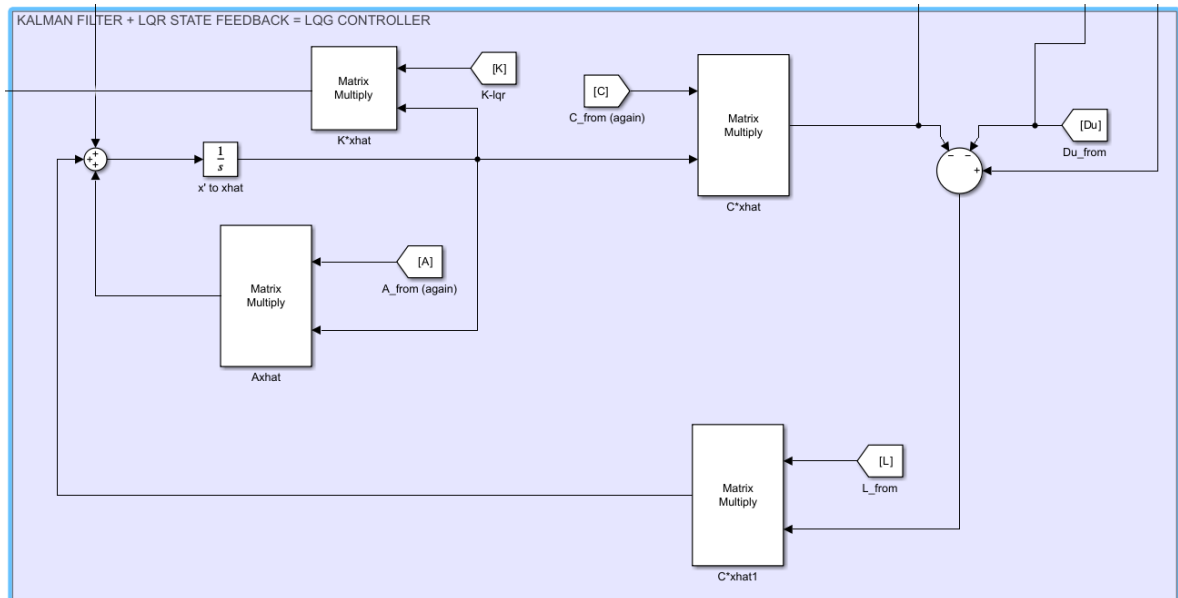
Figura 13 – Limitadores de banda dos fatores “w” e “v”



Fonte: o autor (2025).

Na Figura 13, é possível observar os blocos “Band Limited White Noise”, responsáveis por reter os parâmetros de ruído no sistema, funcionando conforme Equações 6 e 7.

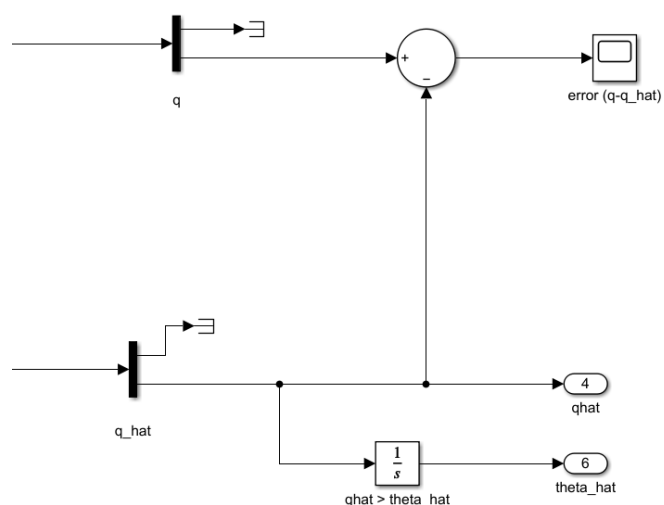
Figura 14 – Módulo de controle Linear Quadrático Gaussiano



Fonte: o autor (2025).

O filtro Kalman, Figura 14, realiza a captação dos dados de posição inicial e atual e aciona as operações matriciais incluindo os produtos dos ganhos em “K” e “L”.

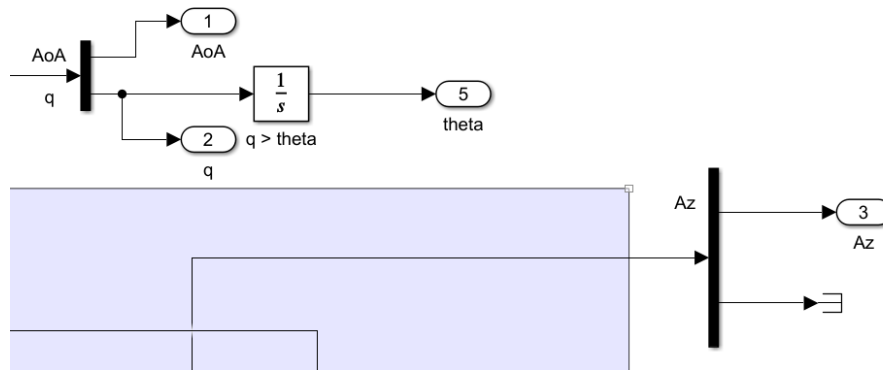
Figura 15 – Valores de saída dos ângulos “q” e “θ” no módulo principal



Fonte: o autor (2025).

Apresentados na Figura 15, os ângulos da taxa de inclinação e “ θ ”, o ângulo de inclinação em relação ao horizonte. Um bloco do tipo “scope” apresenta a plotagem do gráfico de erro entre a taxa de inclinação sem processamento e “ \hat{q} ”, seu correspondente pós processado pelo filtro.

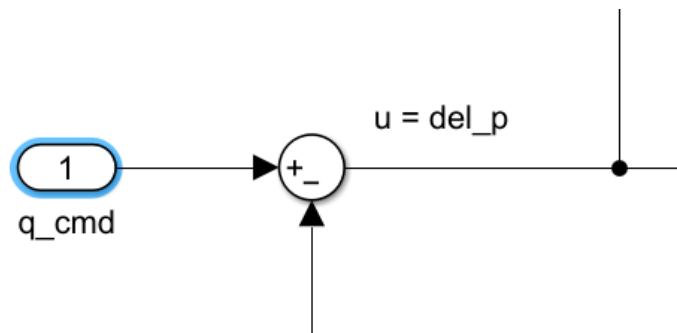
Figura 16 – “Block-set” de “Outport” dos ângulos iniciais



Fonte: o autor (2025).

Para definir as saídas iniciais da estrutura principal, utiliza-se blocos do tipo “outport” que provisiona valores de saída para subsistemas, e sob pretexto de condição pode conter o valor que recebeu por último após seu processamento ou manter o valor inicial. Um bloco do tipo “integrator” realiza a integração em tempo contínuo de um determinado valor de entrada, nesse caso, quando a taxa de inclinação é superior a “ θ ”, sua integral é “ θ ”, apontados na Figura 16.

Figura 17 – “Inport” da taxa de inclinação e soma ao filtro

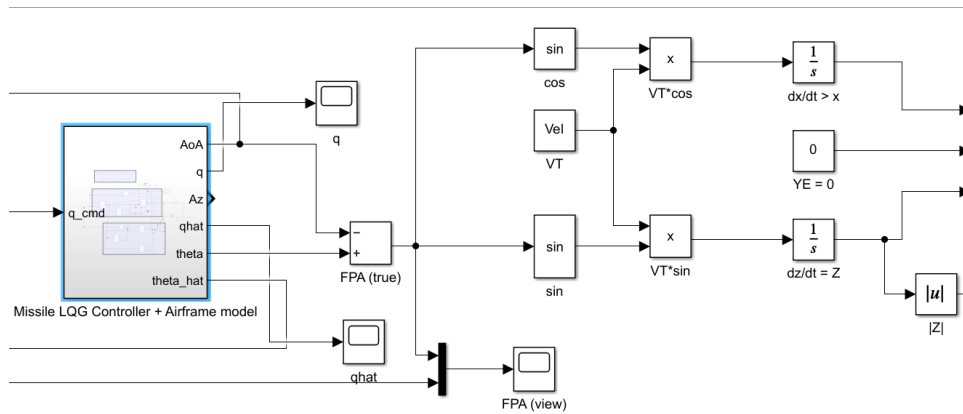


Fonte: o autor (2025).

Regulando o fechamento do “loop” temos o bloco “inport” da taxa de inclinação fornecida ao sistema, Figura 17, atrelada a um bloco “sum” em operação de subtração

ao valor correspondente processado pelo filtro Kalman, essa sequência nos fornece em teoria o ângulo “ δ_p ”.

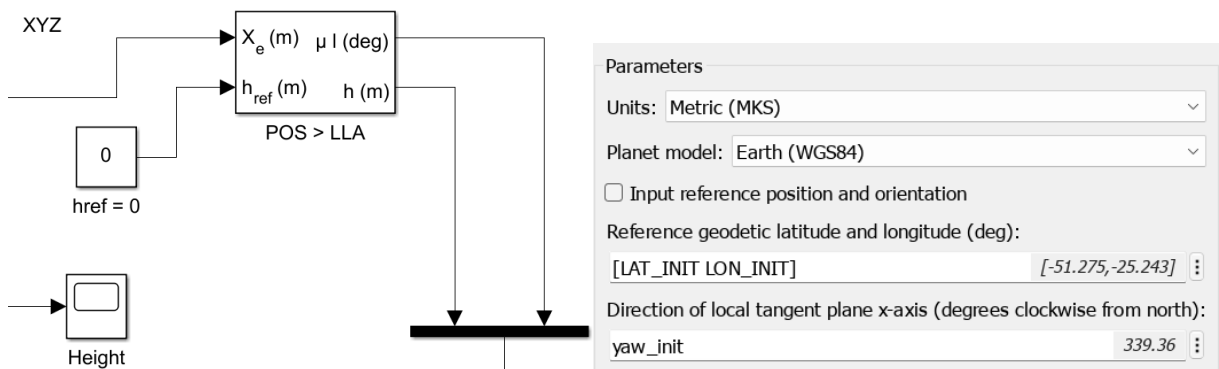
Figura 18 – Sistemas vetoriais



Fonte: o autor (2025).

A partir dos valores de saída da estrutura principal, é possível realizar as operações vetoriais do sistema, Figura 18, para FPA (Flight Path Angle) ou ângulo da trajetória de voo, equivalente a “ γ ”, é apresentado nesse caso em uma operação de subtração entre “AoA” e “ θ ”. Este valor é a saída principal da estrutura com filtro operante, é utilizado nesse sistema para obter os vetores “ X_b ” e “ Z_b ”, que são processados em sequência por um bloco “integrator”, mas de forma derivativa, obtendo a posição em relação a latitude e elevação. Também podemos observar um bloco do tipo “Abs” que nada mais é do que a aplicação do módulo em um valor, representando que factualmente o módulo de “ Z ” é a altitude do objeto em decorrência do tempo.

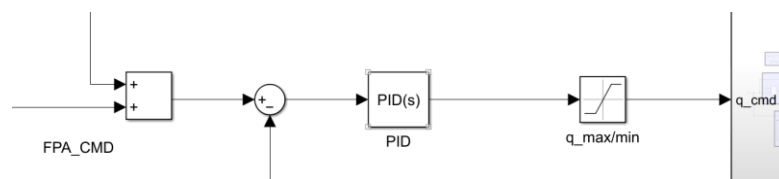
Figura 19 – Módulo “Flat Earth”



Fonte: o autor (2025).

O bloco “Flat Earth” contido especificamente no “add-on” do “Aerospace Toolbox & Blockset”. Como pode ser visto na Figura 19, estima longitude, latitude e altitude a partir de uma posição planejada do planeta Terra. Por posição planejada utiliza-se o contexto teórico de um plano tangencial local, para os dados do bloco foram empregadas as unidades no sistema métrico e a modelagem convencional do planeta Terra. Os pontos referenciais da geodésica são a latitude e longitude iniciais, e a direção do plano tangencial local no eixo “x”, o ângulo da guinada inicial.

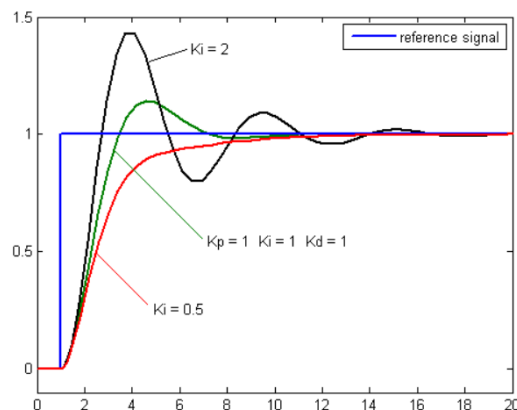
Figura 20 – FPA inicial, PID e saturação “q”



Fonte: o autor (2025).

Na sequência de blocos que formam a parte inicial dos dados que são admitidos na estrutura principal, Figura 20, declara-se o FPA inicial em uma operação de subtração ao ângulo “ θ ” em decorrência do tempo. Seguindo para o Protocolo Integral-Derivativo (PID), este dispositivo tem como função avaliar coeficientes para obter resultados que se assemelham em resultado final a uma curva base. Mas o comportamento anterior a estabilidade pode ser em diversos níveis, distinto da forma inicial, Figura 21, no bloco “saturation” podem ser apontados os limites de entrada de sinal para o máximo (0,5) e mínimo (-0,5) dos valores de saturação e tratando como ganho as linearidades, correspondendo a valores muito próximos do desejado.

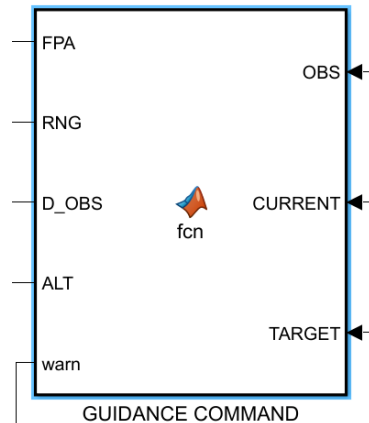
Figura 21 – Comportamento do PID



Fonte: KORKMAZ (2007).

Em sequência, é configurado o bloco de comando do sistema de orientação, Figura 22. A partir de um bloco do tipo “MATLAB Function” que pode receber programação própria para operar com valores gerados dentro dos subsistemas do algoritmo.

Figura 22 – Bloco de comando do sistema de orientação



Fonte: o autor (2025).

Dentro do bloco do sistema de comando definimos as entradas a partir dos valores gerados pelos sistemas anteriores no termo “CURRENT”, sendo alimentado por valores atualizados em decorrer do tempo, em conjunto com a posição do obstáculo (OBS) e do alvo (TARGET).

Figura 23 – Declaração das localizações

```

1  function [FPA,RNG,D_OBS,ALT,warn] = fcn(OBS , CURRENT , TARGET)
2
3  R = 6371e3;
4  d2r = pi/180;
5
6  %LOCALIZACAO DO OBSTACULO
7  LAT_OBS = OBS(1);
8  LON_OBS = OBS(2);
9
10 %LIMITE DO OBSTACULO (ZONA)
11 thres = 2000;
12
13 %LOCALIZACAO DO ALVO
14 LAT_TARGET = TARGET(1);
15 LON_TARGET = TARGET(2);
16 ELEV_TARGET = TARGET(3);
17
18 %LOCALIZACAO ATUAL
19 ELEV_CUR = CURRENT(1);
20 LAT_CUR = CURRENT(2);
21 LON_CUR = CURRENT(3);

```

Fonte: o autor (2025).

Os valores declarados no início da programação, Figura 23, são recebidos a partir dos blocos dos sistemas anteriores e por isso adotam o formato de equivalência.

Figura 24 – Cálculo da distância até o alvo

```
23 %DISTNCIA ATE O ALVO
24 l1 = LAT_CUR*d2r;
25 u1 = LON_CUR*d2r;
26 l2 = LAT_TARGET*d2r;
27 u2 = LON_TARGET*d2r;
28 dh = abs(ELEV_TARGET-ELEV_CUR);
29
30 d1 = l2-l1;
31 du = u2-u1;
32
33 a = sin(d1/2)^2 + cos(l1)*cos(l2)*sin(du/2)^2;
34 c = 2*atan2(sqrt(a),sqrt(1-a));
35 d = R*c; %Distancia horizontal em metros
```

Fonte: o autor (2025).

Para o cálculo da distância progressiva até o alvo, visto na Figura 24. Utiliza-se, a diferença entre o ponto atual do dispositivo em decorrência do tempo e a localização do alvo, em seguida aplicada na fórmula Haversine para obter a distância horizontal.

Figura 25 – Cálculo da distância do obstáculo

```
37 %%DISTANCIA DO OBSTACULO
38 l3 = LAT_OBS*d2r;
39 u3 = LON_OBS*d2r;
40
41 d1 = l3-l1;
42 du = u3-u1;
43
44 a = sin(d1/2)^2 + cos(l1)*cos(l3)*sin(du/2)^2;
45 c = 2*atan2(sqrt(a),sqrt(1-a));
46 d_obs = R*c; %Distancia horizontal em metros
```

Fonte: o autor (2025).

Na Figura 25, observa-se as mesmas operações do cálculo de distância anterior, apenas com o objetivo diferente, a distância do obstáculo auxilia referenciando o ponto de tomada de decisões e antecipação da alteração de trajetória para evitar a localidade e sua abrangência.

Figura 26 – Alcance do alvo

```
47  
48 %%ALCANCE ATUAL (DO ALVO) - ALCANCE > DISTANCIA  
49  
50 range = sqrt(d^2+dh^2);
```

Fonte: o autor (2025).

Define-se o alcance, Figura 26, em um teorema de Pitágoras entre a distância vertical e horizontal, para que seja possível medir a proximidade do dispositivo em relação ao alvo em dois eixos.

Figura 27 – Programação do ponto de ajuste de rota de voo

```
51  
52 %%CALCULA O PONTO DE AJUSTE DA ROTA DE VOO COMANDADA BASEADO NO d_obs,  
53 if abs(d_obs)>=thres  
54     w=0;  
55     FPA_CMD = atan(dh/d);  
56 else  
57     %QUANDO NA ZONA DO OBSTACULO, DISPARAR ALARME  
58     w=1;  
59     FPA_CMD = 0;  
60 end
```

Fonte: o autor (2025).

Para calcular o ajuste de rota de voo, Figura 27, é confeccionado um sistema do tipo “if-else” que no caso de o módulo da distância do objeto ser maior ou igual ao limite de abrangência pré-estabelecido. O valor do alarme de parada (w) é de zero e o ângulo do trajeto de voo será dado pelo inverso da tangente da distância horizontal dividida pela distância vertical. Mas se a distância do obstáculo for menor que o limite de abrangência, o alarme de parada recebe o valor 1, equivalente a “true”, e o ângulo da trajetória de voo é zero.

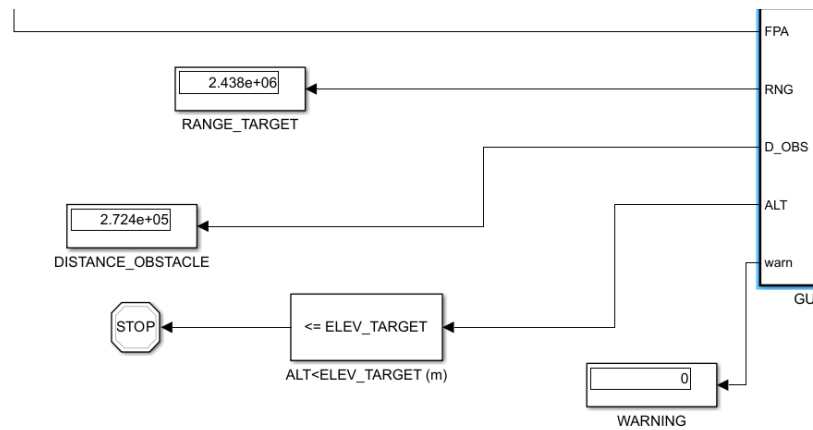
Figura 28 – Declaração das variáveis de saída

```
62 %%VARIABLES DE SAIDA  
63 FPA = FPA_CMD;  
64 RNG = range;  
65 D_OBS = d_obs;  
66 warn = w;  
67 ALT = ELEV_CUR;
```

Fonte: o autor (2025).

Na finalização da programação do módulo de comando de orientação, são declaradas as variáveis de saída do sistema. *Range* como o alcance, “d_obs” a distância do obstáculo, “warn” o alerta e “ALT” recebendo a atualização da elevação, conforme Figuras 28 e 22.

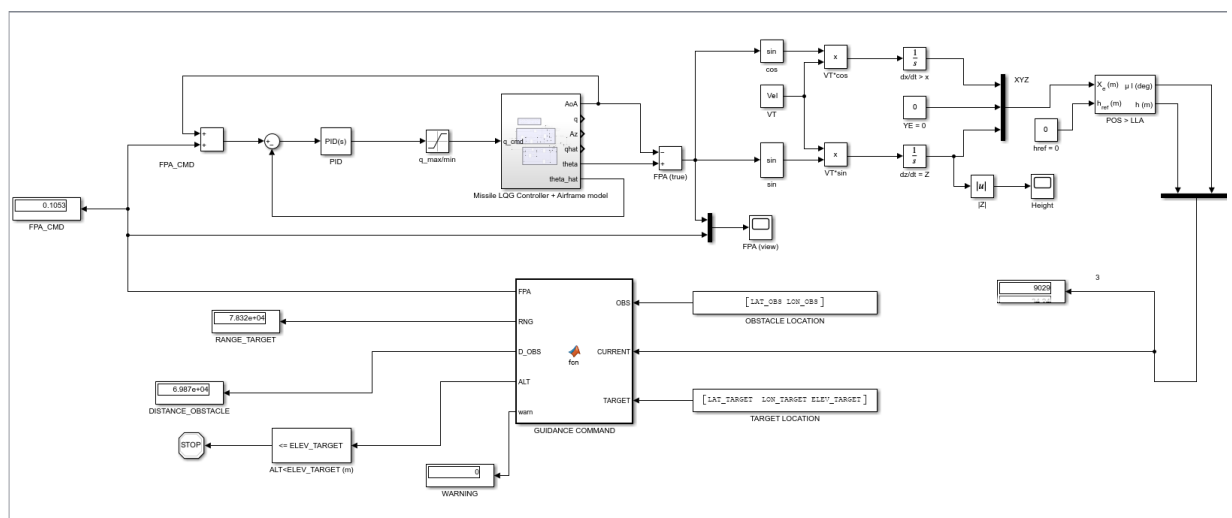
Figura 29 – Saída de valores do comando de orientação



Fonte: o autor (2025).

Os valores indicados pelos blocos “scope” na saída do sistema de comando, Figura 29. Tem como papel apresentar os valores no decorrer da simulação, permitindo avaliar o funcionamento das interações do dispositivo.

Figura 30 – Módulo completo de simulação



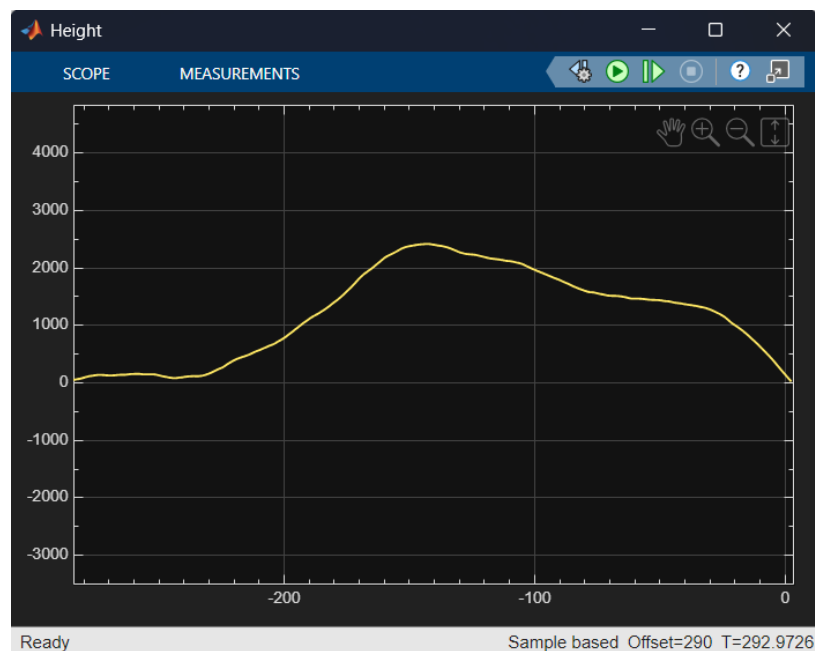
Fonte: o autor (2025)

A Figura 30 apresenta a versão final do modelo para simulação, com os blocos determinantes das variáveis e constantes de entrada que regem o sistema físico. Velocidades, altura, posições, protocolo integral-derivativo e referencial de aspectos terrestres, incluídos nesta etapa.

3 RESULTADOS

Para iniciar a simulação do sistema é feita a compilação e execução do código em MatLab para a aquisição dos dados numéricos que irão suprir os sistemas do algoritmo no Simulink, na sequência o processo de compilação e execução também é realizado neste segundo. Concluída a execução podemos extrair gráficos plotados a partir dos blocos do tipo “scope” para análise.

Figura 31 – Gráfico de altitude



Fonte: o autor (2025).

Na Figura 31 é possível observar a “decolagem” do dispositivo, a sequência de ganho de altitude para o voo e para evitar o obstáculo, completando o trajeto atingindo o alvo a nível do solo novamente. Nota-se o desempenho do funcionamento do grupo do sistema na ausência de pontos abruptos de subida ou descida, mantendo o dispositivo em uma rota suavizada.

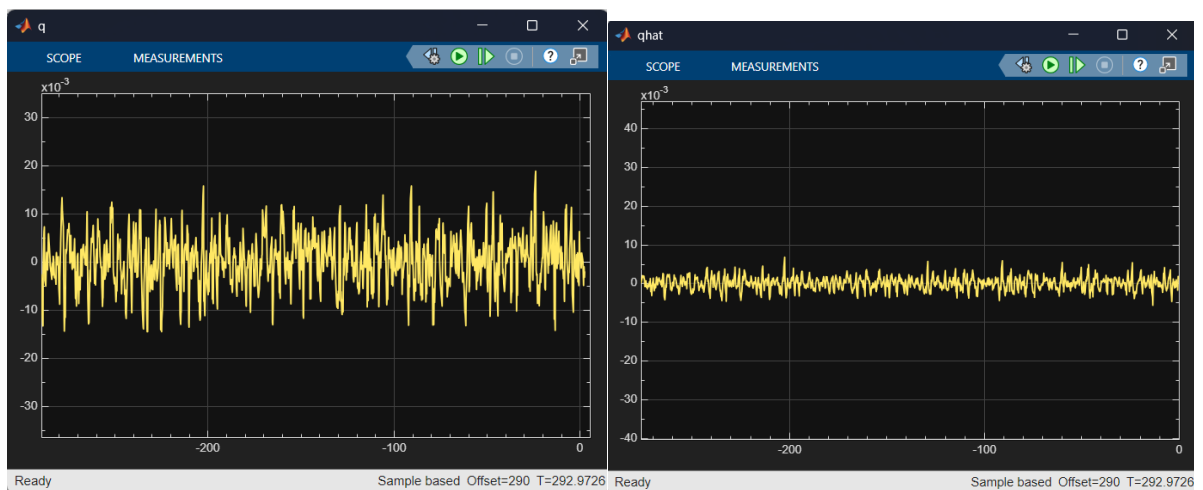
Figura 32 – Gráfico do ângulo da trajetória de voo



Fonte: o autor (2025).

A Figura 32, mostra as variações no ângulo da trajetória de voo, para o de cor amarela temos o gráfico do FPA extraído diretamente do módulo principal, e no de cor azul temos o gráfico obtido a partir do FPA do comando de trajetória. Apontando a abundância de ruído mesmo em um sistema dessa natureza e a sutileza das alterações do sistema de comando.

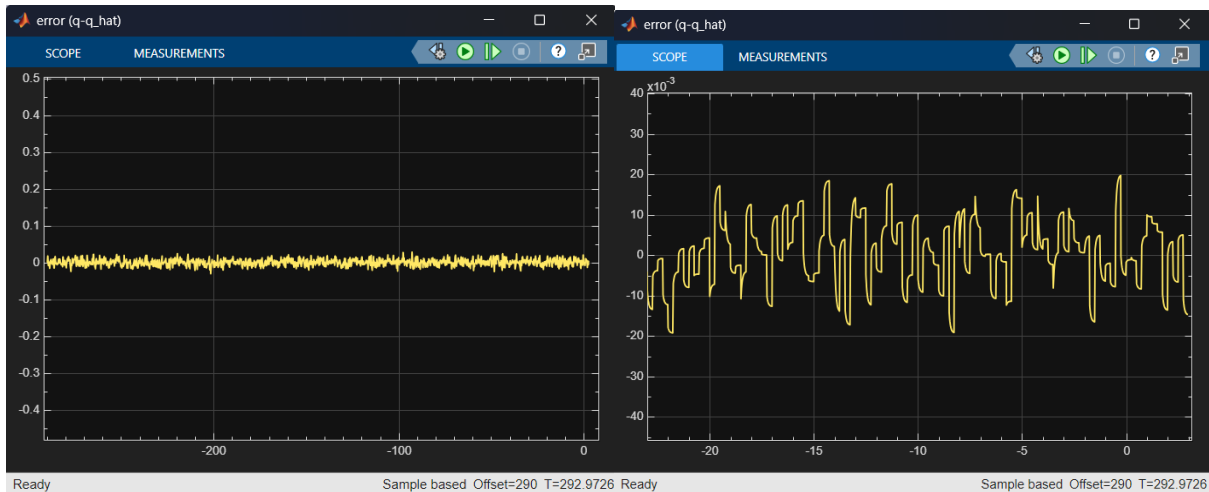
Figura 33 – Gráficos da taxa de inclinação



Fonte: o autor (2025).

Comparando os gráficos para as taxas de inclinação “q” e “qhat”, Figura 33, também é possível notar a diferença exercida pela filtragem de ruído, apesar de possuírem vários pontos abruptos a suavização da curva em “qhat” mostra os efeitos da predição e atribuição da relevância do filtro Kalman.

Figura 34 – Gráfico de erro da taxa de inclinação



Fonte: o autor (2025).

Na Figura 34, é possível analisar a gráfico da divergência entre “q” e “qhat”, mostrando o quão menor a correção deve ser, para que ao invés da progressão dos resultados em “q” sem processamento, seja possível traduzir para curvas com melhores índices de resposta ao ambiente, como em “qhat”.

4 CONSIDERAÇÕES FINAIS

O advento das tecnologias de reutilização de foguetes está transformando a engenharia de transportes e diversas áreas relacionadas. Porém é um campo com produção escassa de material acadêmico e, portanto, de difícil acesso civil. A difusão deste tópico em caráter de pesquisa e projetos auxilia na ampliação do repositório nas universidades e pode contribuir com os avanços dessas tecnologias.

Neste trabalho a principal abordagem é a relevância de um sistema preditivo na captação de dados e processamento de informações para tomada de decisões. Este aspecto dos sistemas embarcados auxilia principalmente na performance do dispositivo sem a necessidade de captações de dados complexas. A utilização de uma quantidade reduzida de variáveis se torna uma quantidade menor de operações,

acompanhando os processos de “down-sizing” mas também requer que os conjuntos equivalentes ou que superem em performance.

Como é possível observar a partir dos resultados, este sistema comprova sua eficiência mesmo que com parâmetros rudimentares, e uma quantidade ínfima de variáveis para um sistema dinâmico, a tomada de decisões nas correções e ganhos de filtro assimilam um formato quase natural de reações evitando perdas energéticas e assegurando resultados bastante próximos do ideal. É possível, então, concluir que sistemas de loop com atualização e predição são essenciais em um projeto de dispositivo auto esterçante e estabilizado, determinando resultados otimizados, unindo exigências teóricas e captações reais.

5 REFERÊNCIAS BIBLIOGRÁFICAS

ABBOTT, Terence S. **A Trajectory Algorithm to Support En Route and Terminal Area Self-Spacing Concepts: Fourth Revision**. Nasa Technical Reports Server. Virginia, USA, 2018.

DE CELIS, Raúl; CARDOSO, Luis. **Adaptive navigation, guidance and control techniques applied to ballistic projectiles and rockets**. Ballistics IntechOpen, Chapter 1. London, United Kingdom, 2019.

DESHPANDE, Vinayak. **Guidance, Navigation and Control System Design**. <https://www.vinayakd.com/articles/guidance-navigation-and-control-fundamentals>. Acesso em 18 de Março de 2025.

EISENBERG, Yael. **Adaptive Control Law for PID**. Nasa Technical Reports Server. Florida, USA, 2019.

EVERETT, Jason M. **A Generalized Guidance Approach to In-Space Solid-Propellant Vehicle Maneuvers**. 43rd Annual AAS Guidance, Navigation & Control Conference, Nasa Technical Reports Server. Colorado, USA, 2020.

HIRSHORN, Steven R. **Expanded Guidance for NASA Systems Engineering. Volume 1: Systems Engineering Practices**. Nasa Technical Reports Server. Washington D.C., USA, 2016.

JACKSON, Paul B. **Overview of Missile Flight Control Systems**. JOHNS HOPKINS APL TECHNICAL DIGEST, VOLUME 29, NUMBER 1. Maryland, USA, 2010.

JACQUES, David R.; STROUBLE, Dennis D. **A-10 Thunderbolt II (Warthog) SYSTEMS ENGINEERING CASE STUDY**. Air Force Center for Systems Engineering (AFIT/SY), Air Force Institute of Technology. Dayton, Ohio, USA, 2010.

JINGPING, Shi; YONGXI, Lu; XIAOBO, Qu; JING, Shi. **A Coordinated Control Method of Thrust Vector and Aerodynamic Surfaces Based on Control Allocation Technology**. 18th International Conference on Control, Automation and Systems (ICCAS), PyeongChang, Korea (South), 2018.

KARPENKO, Mark; HALVERSON, Julie K.; BESSER, Rebecca L. **Waypoint Following Dynamics of a Quaternion Error Feedback Attitude Control System**. Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering Vol. 236 issue 2, Professional Engineering Publishing (Institution of Mechanical Engineers) / SAGE Publications. Maryland, USA, 2022.

KIBBEY, Tim. **Small Launch Vehicle Sizing Analysis with Solid Rocket Examples**. Nasa Technical Reports Server. Ohio, USA, 2019.

KISABO, Aliyu B.; ADEBIMPE, Aliyu F. **State-space modeling of a rocket for optimal control system design**. Ballistics IntechOpen, Chapter 2. London, United Kingdom, 2019.

LEWIS, Emily K. **Piloted Full-Motion Simulation with Simulink®**. AIAA SciTech 2019, Nasa Technical Reports Server. California, USA, 2019.

MEDAGODA, Eran D. B.; GIBBENS, Peter W. **Synthetic-Waypoint Guidance Algorithm for Following a Desired Flight Trajectory**. Journal of guidance, control and dynamics Vol. 33 no. 2. Sydney, Australia, 2010.

MRACEK, Curtis P.; RIDGELY, Brett D. **Missile Longitudinal Autopilots: Connections Between Optimal Control and Classical Topologies**. Raytheon Missile Systems. Tucson, Arizona, USA, 2005.

PANFEROV, Alexander; NEBYLOV, Alexander; BRODSKY, Sergey. **Mathematical Models of Complex Flexible Missile and Software for Control System Design and Simulation**. International Institute for Advanced Aerospace Technologies of State University of Aerospace Instrumentation. Saint Petersburg, Russia, 2009.

PELLACANI, A.; GRAZIANO M.; SUATONI, M. **Design, Development, Validation and Verification of GNC Technologies**. 8th European Conference for Aeronautics and Space Sciences (EUCASS). Madrid, Spain, 2019.

RANAJIT, Das; SIRISHA, Ch. V.; KUMAR, Choudhary Manoj. **Homing Guidance Design Challenges for Tactical Missile**. Scientist, Defence Research and Development Laboratory, IFAC Papers Vol. 51 Issue 1. Hyderabad, India, 2018.

SEIDL, Michael; KLEINER, Brian. **Aircraft Engineering and Aerospace Technology: An International Journal vol. 71 issue 6**. USA, 1999. 546–549p.

SMITH, Kelly M. **Predictive Lateral Logic for Numerical Entry Guidance Algorithms**. AAS/AIAA Space Flight Mechanics Meeting, Nasa Technical Reports Server. California, USA, 2016.

SNOW, Barton H. **Thrust Vectoring Control Concepts and Issues**. JOURNAL OF AEROSPACE Vol. 99 Section 1, Part 2, SAE International. Evendale, Ohio, USA, 1990. 1488-1499p.

SPEER, Dave; JACKSON, George; RAPHAEL, David. **Flight computer design for the Space Technology 5 (ST-5) mission**. IEEE Aerospace Conference. Montana, USA, 2002.

YANG, Yaguang; ZHOU, Zhiqiang. **Spacecraft Dynamics Should be Considered in Kalman Filter Attitude Estimation**. AAS/AIAA Space Flight Mechanics Meeting, Nasa Technical Reports Server. California, USA, 2016.