

Antonio Cesar Urbano da Silva Filho

Análise Dinâmica e de Controle do Mini Drone Parrot Mambo Utilizando Matlab Simulink

Trabalho elaborado na disciplina Trabalho de Graduação II apresentado ao Centro de Engenharia, Modelagem e Ciências Sociais Aplicadas da Universidade Federal do ABC, no curso de Engenharia Aeroespacial.

Universidade Federal do ABC

Centro de Modelagem, Engenharia e Ciências Sociais Aplicadas

Orientador: Diego Paolo Ferruzzo Correa

São Bernardo do Campo - SP

2024

Sumário

1 Simulação e Controle

Este capítulo será dedicado a relacionar o que foi falado até o momento com o projeto disponibilizado pelo MATLAB e a Parrot. Dessa forma a simulação será executada para alguns cenários de interesse e serão gerados gráficos para análise de desempenho dos controladores atuais. Assim permitindo a proposta de melhorias onde for necessário.

1.1 Configuração do Modelo no MATLAB Simulink

Para configurar o modelo no MATLAB, existe uma documentação oficial no site da mathworks*adicionar citação*, que pode ser utilizada, mas além disso uma boa proposta é seguir o trabalho de LI, David, que apresenta um forma mais simples de configuração tanto para a simulação, quanto para testar o quadricóptero físico. Nesse trabalho o objetivo é trabalhar apenas com a simulação, então para iniciar a simulação deve-se rodar o comando *asbQuadcopterStart* no MATLAB, após instalar as dependências necessárias. É importante notar que a princípio as modificações que são feitas no projeto não têm nenhum intuito de melhorar a performance dos controladores ou o desempenho do modelo as configurações que serão citadas a seguir só tem o papel de deixar o modelo pronto para executar as simulações com as entradas que vamos inserir e facilitar a obtenção das saídas de forma que possamos coloca-las em gráficos e realizar a análise.

1.1.1 Configurando Entradas e Saídas

Uma trajetória de voo, pode ser completamente descrita, sabendo como as coordenadas em x, y e z e a guinada do quadricóptero se comportam durante o tempo de simulação, mas como já falado na seção de controle os ângulos de rolagem e arfagem são mais importantes do que as coordenadas em x e y, quando queremos controlar o quadricóptero. Assim, é interesse desse trabalho observar o comportamento dos seis graus de liberdade que existem na simulação, as coordenadas em x, y e z, bem como rolagem, arfagem e guinada.

1.1.1.1 Configuração das entradas

Esse modelo oferece quatro possíveis forma de entrada, Editor de Sinal, Joystick, Dados(em arquivo no formato .mat), planilha de dados(dados no formato .xlsx). Aqui como sugerido por David Li, vamos utilizar apenas editor de sinais para criar os cenários de simulação, após fazer algumas modificações necessárias para permitir a implementação correta no Simulink.

1.1.1.2 Configuração de Saída

Para as saídas, o modelo inclui estados estimados e valores de referência. No entanto, foram adicionados blocos extras, conforme descrito por Li [?], para incluir os valores de estado "verdadeiros" na simulação, permitindo uma comparação mais completa entre os valores estimados e simulados.

1.2 Simulação e Resultados

Nessa seção será apresentada a técnica utilizada para realizar as simulações, bem como os resultados obtidos na simulação.

1.2.1 Simulação

O projeto já tem dois cenários para simulação prontos dentro do bloco de edição de sinais, *Hover* (Voo pairado) e *Landing Search* (Voo buscando um local pré determinado para pouso). O cenário *Landing Search*, não é do nosso interesse no momento pois é um cenário mais complexo, onde o drone utiliza a câmera para procurar um ponto vermelho onde deveria pousar. Como nosso interesse é estudar e entender o desempenho dos controladores, a melhor forma de obter esses resultados é com cenários mais simples como *Hover*, *adicionar cenários que vão ter simulados*.

1.2.1.1 Criando cenários

Para criação ou edição de algum cenário existente, precisamos acessar o *Signal Editor*, abrir o bloco *Position/Attitude Reference*, e clicar no botão circulado em vermelho que abrirá o editor de sinais.

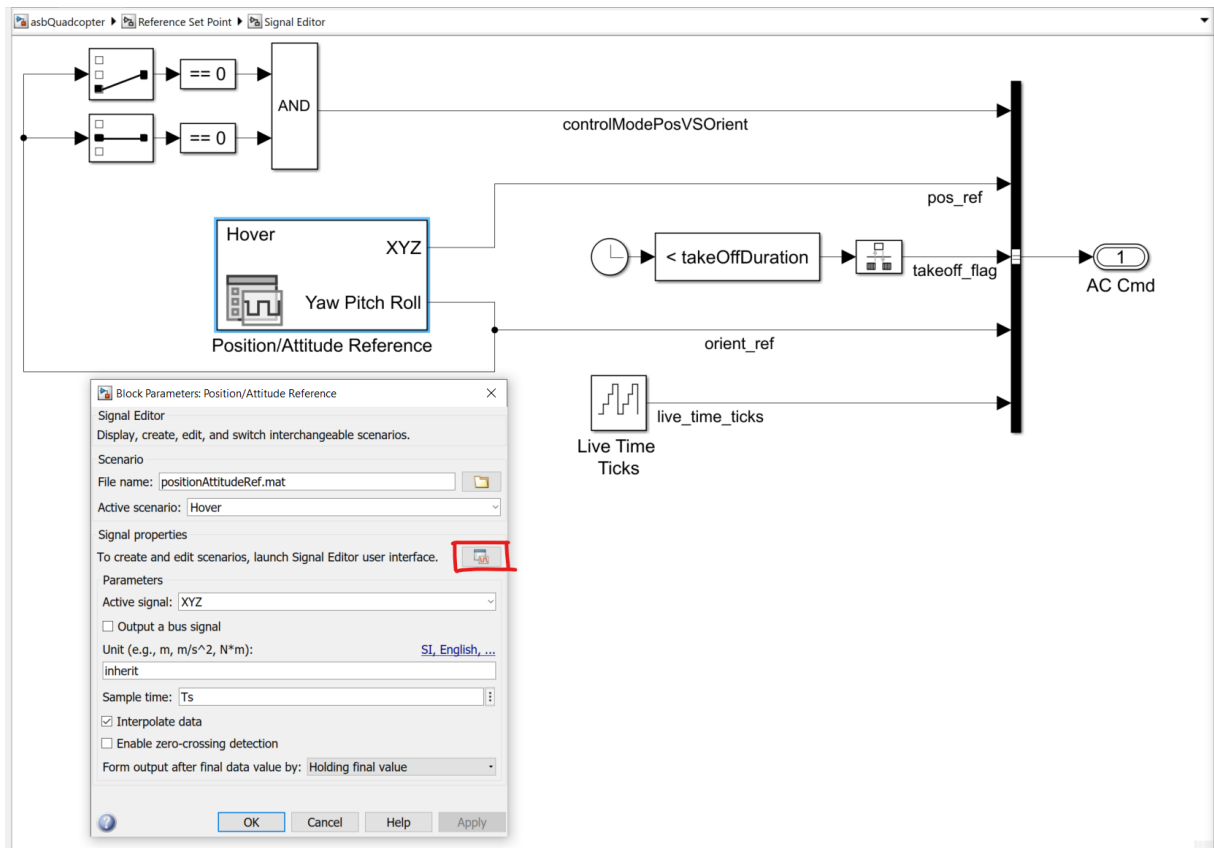


Figura 1 – Como abrir o Editor de Sinais

No editor de sinais, podemos inserir nossos dados de entrada, para os nossos seis graus de liberdade, e em qual momento da simulação o sinal será enviado.


Hover.XYZ			
			
TIME	DATA(1,1,:)	DATA(2,1,:)	DATA(3,1,:)
0	0	0	0
1	0	0	0
1	0	0	-2
2	0	0	-2
2	0	0	-2
100	0	0	-2

Figura 2 – Modificando cenários de Simulação X Y Z


Hover.Yaw Pitch Roll			
			
TIME	DATA(1,1,:)	DATA(2,1,:)	DATA(3,1,:)
0	0	0	0
100	0	0	0

Figura 3 – Modificando cenários de Simulação Guinada, Arfagem e Rolagem

Para executar os cenários de forma programática e pegar os resultados, utilizaremos um código retirado também da nossa principal referência nesse capítulo.

```

1     scenarios = {'TesteXY3'; 'LandingSearch'};
2     TFinals = [25;30];

```



```
3
4     in = Simulink.SimulationInput.empty(size(scenarios, 1),
5                                         0);
6     for i = 1 : size(scenarios, 1)
7         in(i) = Simulink.SimulationInput('asbQuadcopter')
8             ;
9         in(i) = in(i).setBlockParameter(['
10             flightControlSystem/Flight Control System/' ...
11             'landing logic/Position//Attitude
12             Reference'], ...
13             'ActiveScenario', scenarios{i});
14         in(i) = in(i).setVariable('TFinal', TFinals(i), '
15             Workspace', 'asbQuadcopter');
16     end
17
18     out = sim(in(1)); % simulate with the "Hover" scenario
19     for 10 seconds
```

Listing 1.1 – Configuração de Cenários no MATLAB Simulink

No código acima, as linhas 1 e 2 definem o cenário que pretendemos simular (previamente criado dentro do editor de sinais), e o tempo de simulação, respectivamente. O laço de repetição da linha 5 a 11, coloca o cenário no bloco Position/Attitude Reference que está dentro de *Landing logic*, que foi inserido após as modificações que seguimos na seção *configurações de entrada*. A linha 13, executa a simulação com o cenário escolhido e salva os resultados na variável *out*.

A variável *out*, contém todas as informações que são do nosso interesse, no seguinte formato:

```
1     t = out.posref.time;
2     xyzrpy = out.xyzrpy;
3     estim = out.estim.signals.values;
4     posref = out.posref.signals.values;
5     motor = out.motor.signals.values;
6     sensor = out.sensor.signals.values;
```

Listing 1.2 – Exemplo de extração de dados simulados

Com isso, agora podemos executar os cenários com facilidade e colocar os resultados de forma gráfica, para observarmos o comportamento dos controladores.

1.2.2 Resultados

Nesta seção, será realizada a análise dos gráficos gerados a partir das simulações realizadas no MATLAB Simulink. O objetivo principal é avaliar o desempenho dos controladores do quadricóptero em diferentes cenários, com foco no comportamento das variáveis de interesse, como a posição x , y , z e os ângulos de rolagem, arfagem e guinada.

Para interpretar os gráficos de desempenho, serão consideradas as seguintes métricas:

- **Tempo de estabilização:** Tempo necessário para que o sistema se estabilize após uma mudança de referência ou perturbação.
- **Overshoot (ultrapassagem):** A porcentagem que a resposta do sistema excede o valor de referência antes de se estabilizar.
- **Erro em regime permanente:** A diferença entre o valor final da variável controlada e o valor de referência após o sistema estabilizar.
- **Tempo de subida:** O tempo que o sistema leva para sair de 10% a 90% do valor final após uma mudança de referência.

Essas métricas permitirão avaliar a eficácia dos controladores em manter o quadricóptero estável e responsivo, principalmente no cenário de voo pairado (*Hover*), onde a precisão de posição e estabilidade são cruciais.

Os gráficos apresentam o comportamento das coordenadas do quadricóptero ao longo do tempo nos cenários de interesse montados. Ele compara três curvas: **Posição verdadeiro** (linha azul), que representa a altura real do quadricóptero; **Posição estimado** (linha laranja), que é a estimativa feita pelo sistema; e **Posição referência** (linha amarela), que é o valor de referência.

Nas curvas de atitude, é necessário elucidar melhor os termos utilizados. Assim, aqui é preciso lembrar que para o quadricóptero realizar movimentos horizontais, é necessário que aconteça uma arfagem ou rolagem, e no controlador de voo, temos um subsistema que converte os valores de referência de x e y , em comandos de rolagem e arfagem. Nos gráficos, a resultante desses valores são chamados de **comando de arfagem** e **comando de rolagem**.

1.2.2.1 Voo Pairado (Hover)

Neste cenário de voo pairado, o objetivo é que o drone suba até uma altura determinada e permaneça nessa altura até o fim da simulação. Além de analisar o desempenho do controlador de altitude, é interessante observar o comportamento dos controladores de posição nos eixos X e Y , além dos controladores de arfagem, guinada e rolagem.

Parâmetros da simulação:

- Tempo total: 25 segundos;
- Altura desejada: 2 metros;
- Posição em X e Y : 0.

1) Coordenada Z

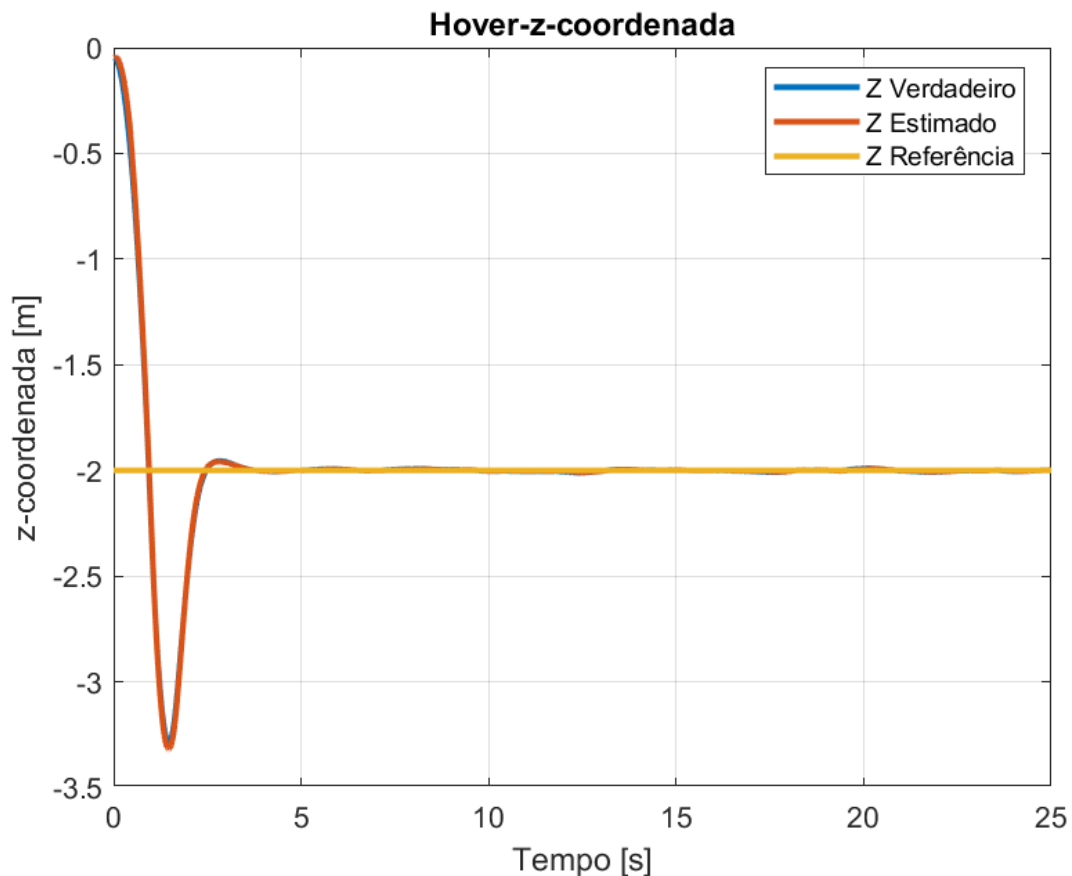


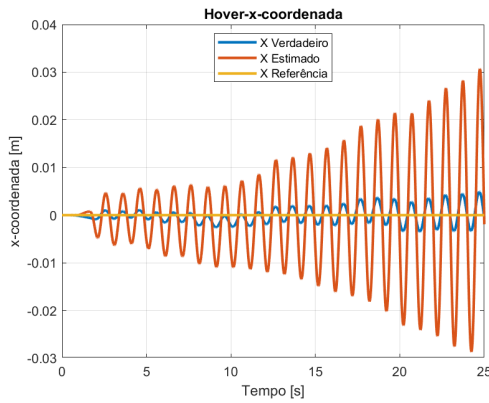
Figura 4 – Coordenada Z para o Cenário de Voo Pairado

Analisando o gráfico, podemos observar que o tempo de estabilização é rápido, por volta dos 3 segundos, após o qual o sistema atinge a estabilidade próximo ao valor de referência. Um pequeno overshoot é visível no início, mas é bem controlado, e o erro em regime permanente é mínimo, indicando um bom desempenho do controlador para manter a altitude.

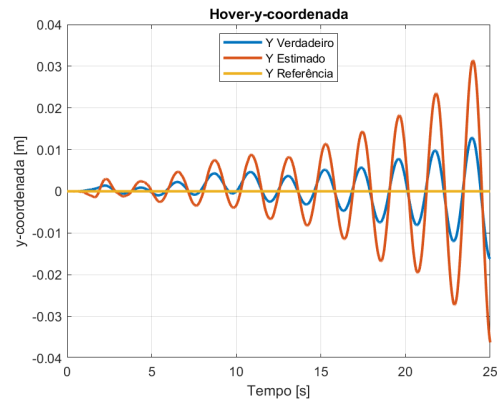
2) Coordenadas X e Y

Abaixo estão os gráficos das coordenadas X e Y do quadricóptero ao longo do tempo:

No dois gráficos, podemos observar oscilações crescentes em torno do valor de referência com o tempo. No gráfico do eixo Y , vemos que o valor verdadeiro aumenta ao



(a) Coordenada X para o Cenário de Voo Pairado



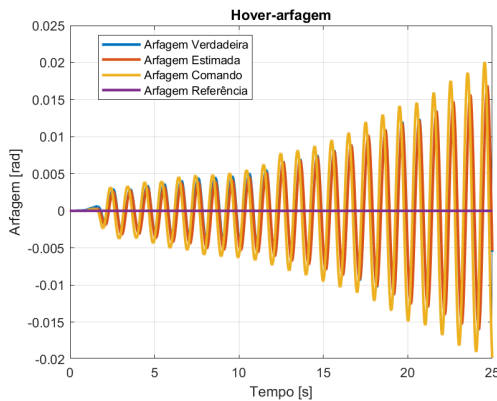
(b) Coordenada Y para o Cenário de Voo Pairado

Figura 5 – Coordenadas X e Y para o Cenário de Voo Pairado

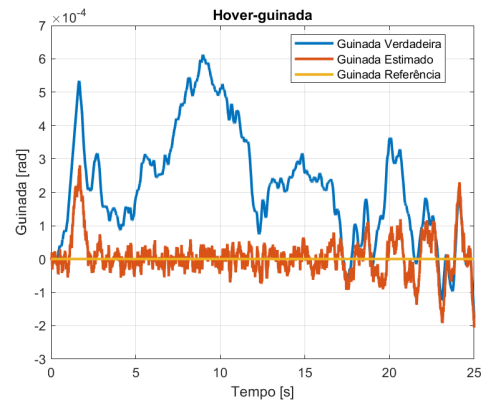
longo do tempo proporcionalmente com os valores estimados, na curva de X , apesar das altas oscilações dos valores estimados, o valor verdadeiro consegue se manter próximo ao valor de referência.

3) Ângulos de Rolagem, Arfagem e Guinada

A seguir estão os gráficos para os ângulos de rotação do quadricóptero: rolamento (roll), arfagem (pitch) e guinada (yaw).



(a) Arfagem para o Cenário de Voo Pairado



(b) Guinada para o Cenário de Voo Pairado

Figura 6 – Arfagem e Guinada para o Cenário de Voo Pairado

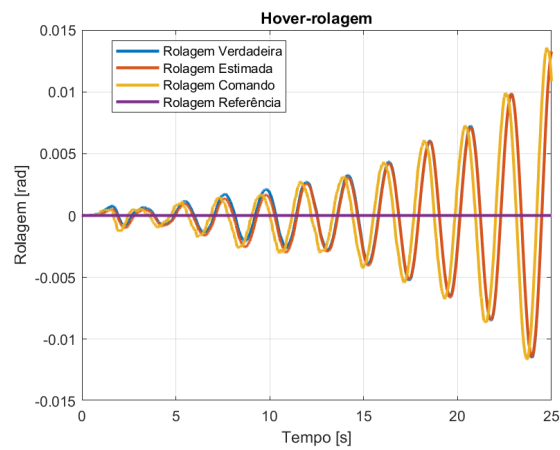


Figura 7 – Rolagem para o Cenário de Voo Pairado

Nos gráficos, vemos que os ângulos de arfagem e rolamento apresentam oscilações crescentes ao longo do tempo, o que inclusive concorda com o comportamento que observamos nos eixos X e Y , enquanto o ângulo de guinada permanece relativamente estável. Essas oscilações indicam uma possível falta de amortecimento no controlador, especialmente nos eixos de arfagem e rolamento.

Sendo assim, uma boa estratégia para iniciarmos nossas melhorias seria ajustar os parâmetros dos controladores de arfagem e rolagem, bem como os controladores da posição X e Y .