

Instituto Tecnológico de Aeronáutica – ITA

Sistemas de Controle Contínuos e Discretos –

CMC-12

Laboratório 5 – Projeto de Autopiloto de Altitude de Avião

Professor: Marcos Ricardo Omena de Albuquerque Maximo

9 de maio de 2022

Observação: por questões de compatibilidade, este laboratório deve ser feito utilizando Simulink R2021b.

1 Introdução

Neste laboratório, projetar-se-á um autopiloto de altitude para um avião comercial Boeing 747 usando o lugar geométrico das raízes (LGR). A Figura 1 apresenta uma aeronave da família 747-8. O sistema de controle de um avião é complexo, de modo que envolve vários controladores, alguns deles para situações específicas de voo. Por conta disso, neste laboratório, foca-se apenas na dinâmica longitudinal da aeronave para projeto de um controlador de altitude. A abordagem aqui é semelhante à adotada em um exemplo do livro de Franklin, Powell e Emami-Naeini [2], porém fez-se algumas modificações com o objetivo de tornar o laboratório mais didático.



Figura 1: Boeing 747.

Um diagrama esquemático com a definição do sistema de coordenadas utilizado é mostrado na Figura 2. O sistema dinâmico linearizado do movimento de corpo rígido de um avião é um sistema MIMO (*multiple-input multiple-output*) com 8 estados. Para projeto através de técnicas clássicas, é comum dividir esse sistema em dois de 4 estados cada. Um deles modela o movimento longitudinal e outro o lateral. O sistema longitudinal

considera movimentos nas direções axial (X), vertical (Z) e de arfagem (θ, q). Já o sistema lateral envolve as direções de rolamento (ϕ, p) e de guinada (ψ, r). Embora na prática exista acoplamento entre os sistemas longitudinal e lateral, assume-se como aproximação de projeto que ele pode ser ignorado. Para controlar a aeronave, usa-se as chamadas superfícies aerodinâmicas, que são chamadas de aileron (rolamento), profundor (arfagem) e leme (guinada). Além disso, também é possível controlar o empuxo realizado pelo motor (propulsão).

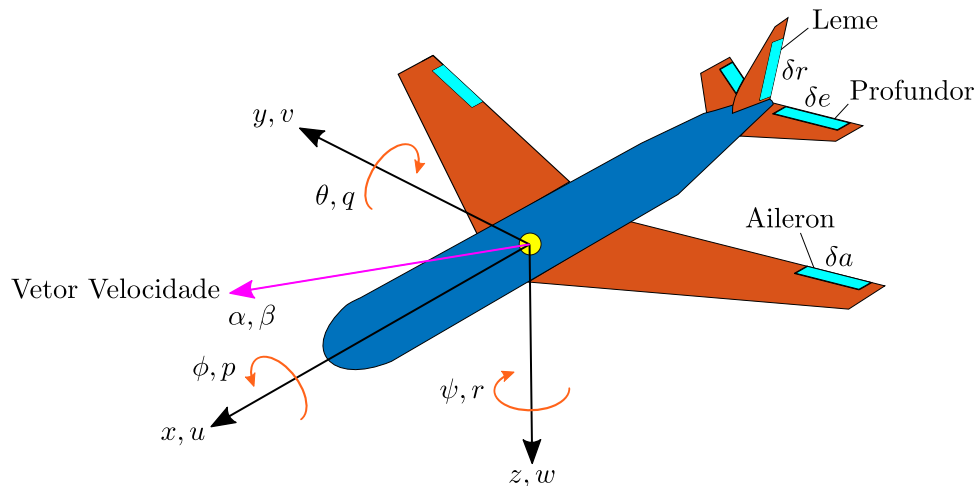


Figura 2: Sistema de coordenadas da aeronave.

Neste roteiro, faz-se uma introdução simplificada sobre as equações que descrevem a dinâmica de voo. Caso tenha interesse em se aprofundar no assunto, recomenda-se verificar a literatura de Mecânica de Voo [1, 4]. Considere a seguinte notação:

- m é a massa da aeronave.
- $[U \ V \ W]^T$ é a velocidade do centro de massa (CM) descrita no sistema de coordenadas do corpo.
- $\alpha = \arctan(W/U)$ é o ângulo de ataque.
- $\beta = \arctan(V/U)$ é o ângulo de derrapagem.
- $[p \ q \ r]^T$ contém as componentes da velocidade angular no sistema de coordenadas do corpo.
- $[X \ Y \ Z]^T$ são as forças aerodinâmicas em relação ao CM no sistema de coordenadas do corpo.
- $[L \ M \ N]^T$ são os torques aerodinâmicos em relação ao CM no sistema de coordenadas do corpo.
- g é a aceleração da gravidade.
- I_{ij} é a inércia relacionada aos eixos i e j .

- θ e ϕ são os ângulos de Euler de arfagem e rolamento da aeronave, respectivamente.
- T é o empuxo resultante da propulsão.
- κ é o ângulo entre o empuxo e o eixo do corpo.

Assumindo que o avião é simétrico em relação ao plano $x-z$, as equações não-lineares de dinâmica de corpo rígido escritas nos eixos do corpo da aeronave são

$$\begin{cases} m(\dot{U} + qW - rV) = X - mg \sin \theta + \kappa T \cos \theta, \\ m(\dot{V} + rU - pW) = Y + mg \cos \theta \sin \phi, \\ m(\dot{W} + pV - qU) = Z + mg \cos \theta \cos \phi - \kappa T \sin \theta, \end{cases} \quad (1)$$

$$\begin{cases} I_{xx}\dot{p} + I_{xz}\dot{r} + (I_{zz} - I_{yy})qr + I_{xz}qp = L, \\ I_{yy}\dot{q} + (I_{xx} - I_{zz})pr + I_{xz}(r^2 - p^2) = M, \\ I_{zz}\dot{r} + I_{xz}\dot{p} + (I_{yy} - I_{xx})qp - I_{xz}qr = N, \end{cases} \quad (2)$$

Um projeto de autopiloto de um veículo aéreo na prática considera linearizações em torno de diferentes pontos de equilíbrio, cada um relacionado a uma situação de voo. Entretanto, para simplificar, considera-se linearização das equações apenas numa situação de voo reto nivelado com velocidade constante. Nesse caso, tem-se

$$\dot{U} = \dot{V} = \dot{W} = \dot{p} = \dot{q} = \dot{r} = 0. \quad (3)$$

Seja $[U_0 \ V_0 \ W_0]^T$, $[\phi_0 \ \theta_0 \ \psi_0]^T$ e $[p_0 \ q_0 \ r_0]^T$ vetores de velocidade, ângulos de Euler e velocidades angulares no ponto de equilíbrio, respectivamente. Como assume-se que o avião não está rotacionando em nenhum eixo no equilíbrio, tem-se $p_0 = q_0 = r_0$. Como a aeronave está nivelada em rolamento, tem-se também $\phi_0 = 0$. Porém, é necessário que exista algum ângulo de ataque para que a asa forneça sustentação aerodinâmica para compensar a gravidade, logo $\theta_0 \neq 0$ e $W_0 \neq 0$. Ademais, g_0 é o valor da aceleração da gravidade. As velocidades no ponto de equilíbrio são dadas por

$$\begin{cases} U_0 = V_{ref} \cos \theta_0, \\ V_0 = 0, \\ W_0 = V_{ref} \sin \theta_0, \end{cases} \quad (4)$$

em que V_{ref} é o módulo da velocidade de referência. Denote

$$\begin{cases} u = U - U_0, \\ v = V - V_0, \\ w = W - W_0. \end{cases} \quad (5)$$

Assuma ainda que

$$\begin{aligned} (v^2, w^2) &<< u^2, \\ (\phi^2, \theta^2) &<< 1, \\ (p^2, q^2, r^2) &<< u^2, b^2, \end{aligned} \quad (6)$$

em que b é o envergadura de asa do avião (*wingspan*). Com isso, pode-se fazer linearização por análise de pequenos sinais, o que resulta nos sistemas lineares longitudinal e lateral.

Por concisão, mostra-se apenas o sistema longitudinal, o qual é dado por

$$\begin{bmatrix} \dot{u} \\ \dot{w} \\ \dot{q} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} X_u & X_w & -W_0 & -g_0 \cos \theta_0 \\ Z_u & Z_w & U_0 & -g_0 \sin \theta_0 \\ M_u & M_w & M_q & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} u \\ w \\ q \\ \theta \end{bmatrix} + \begin{bmatrix} X_{\delta e} \\ Z_{\delta e} \\ M_{\delta e} \\ 0 \end{bmatrix} \delta e \quad (7)$$

em que:

- δe é o comando de profundor (variação em torno do valor de equilíbrio).
- X_u , X_w e $X_{\delta e}$ representam derivadas parciais de X em relação a u , w e δe , respectivamente.
- Z_u , Z_w e $Z_{\delta e}$ representam derivadas parciais de Z em relação a u , w e δe , respectivamente.
- M_u , M_w , M_q e $M_{\delta e}$ representam derivadas parciais de M em relação a u , w , q e δe , respectivamente.

Para incluir a altitude na dinâmica, deve-se considerar ainda

$$\dot{h} = V_{ref} \sin \theta - w \cos \theta, \quad (8)$$

de modo que a equação linearizada fica

$$\dot{h} = V_{ref} \theta - w. \quad (9)$$

Linearizando um Boeing 747 em voo horizontal com uma velocidade nominal $U_0 = 830 \text{ ft/s} = 253 \text{ m/s}$ a $20.000 \text{ ft} = 6.096 \text{ m}$ com uma massa de $637.000 \text{ lb} = 288.938 \text{ kg}$, o sistema longitudinal fica

$$\begin{bmatrix} \dot{u} \\ \dot{w} \\ \dot{q} \\ \dot{\theta} \\ \dot{h} \end{bmatrix} = \begin{bmatrix} -0,00643 & 0,0263 & 0 & -32,2 & 0 \\ -0,0941 & -0,624 & 820 & 0 & 0 \\ -0,000222 & -0,00153 & -0,668 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & -1 & 0 & 830 & 0 \end{bmatrix} \begin{bmatrix} u \\ w \\ q \\ \theta \\ h \end{bmatrix} + \begin{bmatrix} 0 \\ -32,7 \\ -2,08 \\ 0 \\ 0 \end{bmatrix} \delta e, \quad (10)$$

de modo que a função de transferência de δe para h é

$$\frac{h(s)}{\delta e(s)} = \frac{32,7(s + 0,0045)(s + 5,645)(s - 5,61)}{s(s + 0,003 + 0,0098j)(s + 0,003 - 0,0098j)(s + 0,6463 + 1,1211j)(s + 0,6463 - 1,1211j)}. \quad (11)$$

Em mecânica de voo, os polos mais lentos (em $-0,003 \pm 0,0098j$) são associados ao chamado **modo fugóide**, enquanto os polos de menor período (em $-0,6463 \pm 1,1211j$) representam o **modo de período curto**. Perceba que a dinâmica em questão é de quinta ordem, com dois pares de polos complexo-conjugados mal amortecidos e um zero no semiplano direito (fase não-mínima), de modo que projetar um autopiloto de altitude não é uma tarefa tão trivial. Destaca-se que as unidades do sistema apresentado em (11) são

- u e w em ft/s .
- q em $crad/s$.
- θ em $crad/s$.
- h em ft .
- δe em $crad$.

Apesar das unidades não estarem em SI, o que é um tanto incômodo, não há necessidade de fazer nenhuma modificação nos métodos de projeto para se adaptar a essas unidades. Como a unidade de tempo usada é segundo (s), os conceitos relacionados a frequência continuam válidos.

2 Tarefas

2.1 Projeto de Controlador de Arfagem

Dada a função de transferência apresentada em (11), projetar um controlador de altitude diretamente é muito complicado. Assim, usa-se uma estratégia já adotada diversas vezes durante o curso: uso de malhas aninhadas, com uma malha interna de arfagem (θ) rápida, de modo a facilitar o projeto da malha externa de altitude (h), como mostra a Figura 3. Perceba que usar realimentação (*feedback*) de ângulo e velocidade angular de arfagem para melhorar o amortecimento de uma malha de altitude é uma estratégia muito usada em controle de sistemas aeroespaciais [2, 3].

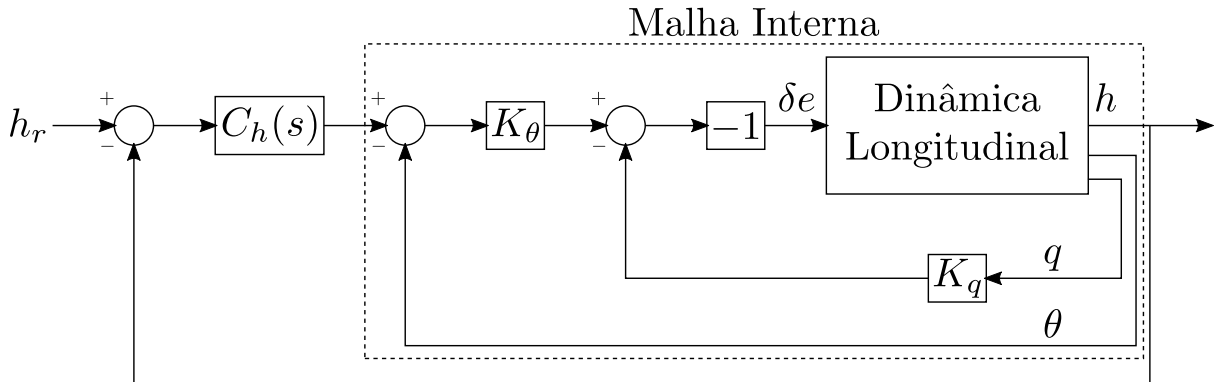


Figura 3: Autopiloto longitudinal com malhas aninhadas.

A malha de controle de uma aeronave de grande porte costuma contar com medidas de θ , q e h para realimentação. A medida de q é determinada por um girômetro presente em uma unidade de medida inercial (UMI), enquanto h é obtido através de um altímetro barométrico. Há dispositivos para medição direta de θ chamados giroscópios, porém são caros e pesados. Por conta disso, veículos aéreos modernos costumam estimar θ através da fusão de informações dos demais sensores.

- A partir do código base entregue no arquivo `obterPlantaLongitudinal.m`, defina um sistema dinâmico em representação de espaço de estados de acordo com (10). Considere que o vetor de saídas é $\mathbf{y} = [q \ \theta \ h]^T$. Use a função `ss` do MATLAB para definir o sistema dinâmico a partir das matrizes do espaço de estados.
- Com uso da função `step` do MATLAB, mostre que uma deflexão positiva do profundor gera valores negativos de q , θ e h , o que é o contrário do que usualmente tem-se trabalhado, já que isso gera um ganho negativo ao longo da malha. Inclua o gráfico no relatório.
- Para compensar o efeito discutido no item anterior e evitar trabalhar com ganhos negativos, considere $G_q(s) = -q(s)/\delta e(s)$. **Apresente essa função de transferência no seu relatório.** A propósito, esse é o motivo de existir um bloco com ganho -1 no diagrama de blocos da Figura 3. **Dica:** use a função `ss2tf` do MATLAB para converter de espaço de estados para função de transferência.
- Mostre que um controlador P+V (com a topologia mostrada na Figura 3) aloca os polos de malha fechada da malha de atitude (entenda atitude como um sinônimo de orientação) na mesma posição que um controlador PD desde que $K_{p\theta} = K_\theta$ e $K_{d\theta} = K_q$. **Dica:** considere $G_q(s) = N_q(s)/D_q(s)$.
- Como q é medido diretamente por um sensor, é preferível usar o controlador P+V na malha de atitude. Com isso, tem-se dois ganhos (K_θ e K_q) para ajustar, mas o LGR descreve a influência apenas de um ganho proporcional. Por outro lado, através a observação do item anterior, pode-se equivalentemente verificar a influência de K_q como se fosse um zero a ser adicionado ao LGR. Com isso, use a ferramenta `rltool` do MATLAB para projetar o controlador P+V de atitude. Sobre o projeto, destaca-se:
 - No caso de um autopiloto de avião com passageiros, como é o caso, deve-se tomar o cuidado de não projetar uma resposta agressiva demais. Para uma resposta agradável, recomenda-se $\xi \approx 0,5$ e ω_n menor que 1 rad/s [2] para a dinâmica de altitude.
 - Tomando $\omega_n \approx 0,7 \text{ rad/s}$ para a altitude, adota-se $\omega_n = 4 \text{ rad/s}$ e $\xi = 0,6$ como requisitos para a malha interna de arfagem (muito mais rápida e bem amortecida).
 - Adicione o zero referente ao K_q à esquerda dos polos complexo-conjugados do modo fugóide, de modo a “puxá-los” mais para a esquerda no LGR. Para adicioná-lo, clique com o botão direito em cima do gráfico e vá até **Add Pole or Zero->Real Zero**. Então, clique na posição do eixo real onde deseja posicionar o zero. Você pode arrastar o zero e ver as alterações no LGR em tempo real, o que facilita manipular os ramos do LGR para que passem pelos pontos desejados.
 - Adicione os requisitos no plano s . Clique com o botão direito em cima do gráfico e vá até **Design requirements->New...**
 - Finalize o projeto com uso do `rltool`. Mova o zero e os polos de malha fechada no plano s de forma intuitiva até atender aproximadamente aos requisitos.

Dica: mova primeiramente o zero até que os ramos do LGR passem pelos pontos desejados de projeto e então mova os polos de malha fechada até que eles estejam posicionados nas posições desejadas.

- Ignore a resposta ao degrau unitário mostrada pelo `rltool` nesse caso, pois o `rltool` está considerando o zero adicional do sistema com PD que não existe no sistema com P+V.
- Com o ganho necessário para posicionar os polos relacionados ao modo fugóide na posição desejada, os polos relativos ao modo de período curto deverão estar muito próximos dos zeros do sistema. Com isso, o efeito do período curto é praticamente cancelado.
- Para obter os ganhos projetados, clique com o botão direito em cima do gráfico e vá até **Edit Compensator...**
- Apresente no relatório um *print screen* do `rltool` na situação do projeto.
- Compare em dois gráficos separados o sistema em malha fechada projetado (com controlador P+V) com o em malha aberta em termos de resposta ao degrau unitário. Discuta a melhoria em termos de rapidez e amortecimento. Inclua o gráfico no relatório.

2.2 Projeto de Controlador de Altitude

Projete o controlador de altitude usando o controlador de arfagem como malha interna:

- Adote um compensador PD de acordo com $C_h(s) = K_{ph} + K_{dh}s$.
- Para obter a função de transferência $h(s)/\theta_r(s)$, encontre a dinâmica do sistema considerando a presença da malha interna em representação de espaço de estados na forma

$$\dot{\mathbf{x}} = \mathbf{A}_\theta \mathbf{x} + \mathbf{B}_\theta \theta_r, \quad (12)$$

$$h = [0 \ 0 \ 0 \ 0 \ 1] \mathbf{x}, \quad (13)$$

em que $\mathbf{x} = [u \ w \ q \ \theta \ h]^T$, e \mathbf{A}_θ e \mathbf{B}_θ são as matrizes de estado e de entrada, respectivamente. Então, converta o espaço de estados para função de transferência com `ss2tf`. Apresente a função de transferência $h(s)/\theta_r(s)$ no relatório. **Dica:** para obter (12), considere (10) e substitua:

$$\delta_e = (-1) \times [K_\theta (\theta_r - \theta) - K_q q], \quad (14)$$

em que o -1 está presente devido ao bloco com ganho -1 existente na Figura 3.

- Com uso da ferramenta `rltool`, projete o sistema para que tenha polos dominantes com $\omega_n = 0,7 \text{ rad/s}$ e $\xi = 0,5$. Os demais polos devem ter $\xi > 0,5$. Além disso, pede-se tempo de subida de 0 a 100% tal que $3 \text{ s} \leq t_r|_0^{100\%} \leq 4 \text{ s}$ e sobressinal $M_p < 0,2$.
- Assim como no projeto do controlador de atitude, use o zero adicional para “puxar” os polos mais lentos para a esquerda.

- Para os requisitos no domínio do tempo, aproveite o gráfico de resposta ao degrau unitário traçado pelo `rltool`. Caso o gráfico não esteja aberto, solicite-o em **New Plot->New Step**.
- Caso precise alterar os eixos do gráfico (para dar *zoom* em alguma parte específica), clique com o botão direito em cima do gráfico e vá em **Properties....** Então, vá na aba **Limits**.
- Apresente um *print screen* do `rltool` na situação do projeto e a resposta ao degrau unitário do sistema em malha fechada no relatório para comprovar atendimento aos requisitos.

2.3 Simulação do Autopiloto Longitudinal

O Simulink não permite implementar uma função de transferência de um derivativo puro, pois resulta em um sistema impróprio. Conforme foi discutido em sala, numa implementação prática, passa-se a saída do termo derivativo por um filtro passa-baixas, de modo que a função de transferência do compensador PD fica

$$C(s) = K_p + K_d s \left(\frac{a}{s+a} \right) = \frac{(K_p + K_d a)s + K_p a}{s+a} = K \left(\frac{s+z}{s+p} \right), \quad (15)$$

em que

$$\begin{cases} K = K_p + K_d a, \\ z = K_p a / (K_p + K_d a), \\ p = a. \end{cases} \quad (16)$$

Escolhe-se a com frequência alta o suficiente para não perturbar o comportamento do sistema, mas baixa o suficiente para fornecer uma filtragem de ruído adequada. Um compensador no formato $K(s+z)/(s+p)$ com $p > z$ também é chamado de compensador *lead* na literatura. Com isso, solicita-se:

- Implemente os parâmetros do autopiloto longitudinal encontrados nas tarefas anteriores na função `obterControladorLongitudinal`. Escolha $a = 10z$ para que a influência do polo no compensador *lead* seja baixa.
- Conclua a implementação do modelo de simulação da dinâmica longitudinal em `boeing747_longitudinal.slx`. Use o diagrama da Figura 3 como base. O modelo Simulink simula ruído de medida no barômetro (sensor de altitude), de modo que a altura medida é $h_m = h + \nu$, em que ν é um ruído com distribuição gaussiana de média nula e variância σ_h^2 , amostrado a uma taxa de 100 Hz , ou seja, a cada $0,01 \text{ s}$, o modelo amostra um novo número aleatório $\nu \sim \mathcal{N}(0, \sigma_h^2)$. Veja a seção de dicas caso ainda tenha dificuldades.
- Use a função `avaliarAutopilotoLongitudinal`, que simula a resposta do autopiloto longitudinal com diferentes valores da variância σ_h^2 .
- Inclua os gráficos gerados no relatório e discuta o que se observa qualitativamente. Perceba que, mesmo com uma medida de altura muito ruidosa, a altura real do avião é razoavelmente bem comportada. Isso acontece porque sistemas dinâmicos atuam como filtros.

3 Instruções

- A primeira etapa do processo de correção consistirá em submeter as funções implementadas a vários casos de teste de forma automatizada. Assim, os cabeçalhos das funções devem ser seguidos **rigorosamente**. Arquivos `.m` com os nomes destas funções e os cabeçalhos já implementados foram fornecidos juntamente com este roteiro. Dê preferência a implementar seu laboratório a partir destes arquivos `.m` fornecidos para evitar erros.
- A entrega da solução desse laboratório consiste de arquivos de código (MATLAB e Simulink) e de um relatório (em `.pdf`), que devem ser submetidos no Google Classroom.
- Compacte todos os arquivos a serem submetidos em um único `.zip` (use obrigatoriamente `.zip`, e **não** outra tecnologia de compactação de arquivos) e anexe esse `.zip` no Google Classroom. Para o `.zip`, use o padrão de nome `<login_ga>_labX.zip`. Por exemplo, se seu login é `marcos.maximo` e você está entregando o laboratório 1, o nome do arquivo deve ser `marcos.maximo_lab1.zip`. **Não** crie subpastas, deixe todos os arquivos na “raiz” do `.zip`.
- O relatório deve ser sucinto, preocupe-se apenas em incluir discussões e entregáveis solicitados no roteiro. Pede-se apenas um cuidado mínimo na elaboração do relatório: responder adequadamente as perguntas, incluir figuras diretamente no relatório (ao invés de deixar como arquivos separados), figuras de boa qualidade, colocar nomes nos eixos dos gráficos, colocar legenda para diferenciar curvas num mesmo gráfico etc.
- **Não** é permitido o uso de funções ou comandos prontos do MATLAB que realizem toda a funcionalidade atribuída a uma certa função cuja implementação foi solicitada. Entretanto, o uso destas funções para verificação das implementações realizadas é encorajado. Em caso de dúvida, consulte o professor.
- A criação de *scripts* e funções auxiliares no MATLAB para execução de experimentos e geração de gráficos é fortemente recomendada para facilitar seu trabalho. Porém, não há necessidade de entregar código auxiliares.
- **Não** há necessidade de copiar e colar o código no seu relatório, dado que você também submeterá os arquivos de código. Também **não** há necessidade de explicar sua implementação no relatório, a **não** ser que o roteiro tenha solicitado explicitamente. Porém, organizar e comentar o código é muito salutar, pois ele será o foco da correção.

4 Dicas

- Nos modelos de simulação entregues, todos os blocos e os parâmetros de simulação já estão adequadamente configurados.

- Além dos blocos Simulink conhecidos de laboratórios anteriores, neste laboratório, usa-se:
 - **State Space**: simula um sistema dinâmico escrito em representação de espaço de estados.
 - **Random Number**: amostra número aleatórios de acordo com uma distribuição gaussiana $\mathcal{N}(\mu, \sigma^2)$ e tempo de amostragem T .
 - **Demux**: separa um sinal “vetorial” do Simulink em suas componentes. A ordem dos sinais na saída do *demux* é a mesma em que eles estão no vetor. Sim, lembra o *demux* de EEA-21 :).
- Para colocar tempo de amostragem em blocos Simulink, deve-se editar a propriedade **Sample Time** desses. Não se preocupe, pois todos os blocos já foram configurados.
- Para usar a saída do bloco **To Workspace** no formato **Structure with Time**:
`plot(out.x.time, out.x.signals.values)`
- Caso queira chamar o Simulink dentro do MATLAB, use `out = sim('arquivo.slx')`. A saída do Simulink ficará na variável `out` nesse caso.

Referências

- [1] John H. Blakelock. *Automatic Control of Aircraft and Missiles*. Wiley, 1991.
- [2] Gene F. Franklin, J. Davis Powell, and Abbas F. Emami-Naeini. *Feedback Control of Dynamic Systems*. Pearson, 2015.
- [3] Paul B. Jackson. Overview of missile flight control systems. Technical report, John Hopkins APL, 2010.
- [4] Arthur E. Bryson Jr. *Control of Spacecraft and Aircraft*. Princeton University Press, 1994.