

Aula Prática 2: Linearização e Simulink

AB-722

Flávio Luiz Cardoso Ribeiro

<http://flavioluiz.github.io>

flaviocr@ita.br

Departamento de Mecânica do Voo
Divisão de Engenharia Aeronáutica e Aeroespacial
Instituto Tecnológico de Aeronáutica



2018

Objetivos da aula

- Linearização numérica das equações do movimento longitudinal;
- Implementação das equações do movimento no simulink com auxílio de s-function;
- Apresentação do uso do Flight Gear em ambiente simulink.

Linearização numérica

Obtivemos equações na forma:

$$\dot{X} = f(X, U)$$

Onde f é uma função não linear dos estados (X) e controles (U).
Desejamos linearizar o modelo para:

- Permitir a aplicação de técnicas de controle linear;
- Efetuar um estudo de estabilidade e da qualidade de vôo da aeronave.

Linearizando teremos:

$$\dot{X} = AX + BU$$

Linearização numérica

$$\dot{X} = \begin{bmatrix} \dot{V} \\ \dot{\gamma} \\ \dot{q} \\ \dot{\alpha} \\ \dot{H} \\ \dot{x} \end{bmatrix} = f(X, U) = \begin{bmatrix} f_V \\ f_\gamma \\ f_q \\ f_\alpha \\ f_H \\ f_x \end{bmatrix}$$

Linearizando por exemplo a primeira linha (em torno do equilíbrio), fica:

$$\dot{V} = f_V(X, U) = f_V(V, \gamma, q, \alpha, H, \pi, \delta p)$$

$$\dot{V} = f_{V_{eq}} + \frac{\partial f_V}{\partial V_{eq}} (V - V_{eq}) + \frac{\partial f_V}{\partial \gamma_{eq}} (\gamma - \gamma_{eq}) + \dots$$

E de maneira análoga para os demais estados:

$$\dot{\gamma} = f_{\gamma_{eq}} + \frac{\partial f_\gamma}{\partial V_{eq}} (V - V_{eq}) + \frac{\partial f_\gamma}{\partial \gamma_{eq}} (\gamma - \gamma_{eq}) + \dots$$

...

Linearização numérica

Incluindo todas as equações:

$$\begin{bmatrix} \frac{\partial f_V}{\partial V} & \frac{\partial f_V}{\partial \gamma} & \frac{\partial f_V}{\partial q} & \frac{\partial f_V}{\partial \alpha} & \frac{\partial f_V}{\partial H} \\ \frac{\partial V}{\partial f_\gamma} & \frac{\partial \gamma}{\partial f_\gamma} & \frac{\partial q}{\partial f_\gamma} & \frac{\partial \alpha}{\partial f_\gamma} & \frac{\partial H}{\partial f_\gamma} \\ \frac{\partial V}{\partial f_q} & \frac{\partial \gamma}{\partial f_q} & \frac{\partial q}{\partial f_q} & \frac{\partial \alpha}{\partial f_q} & \frac{\partial H}{\partial f_q} \\ \frac{\partial V}{\partial f_\alpha} & \frac{\partial \gamma}{\partial f_\alpha} & \frac{\partial q}{\partial f_\alpha} & \frac{\partial \alpha}{\partial f_\alpha} & \frac{\partial H}{\partial f_\alpha} \\ \frac{\partial V}{\partial f_H} & \frac{\partial \gamma}{\partial f_H} & \frac{\partial q}{\partial f_H} & \frac{\partial \alpha}{\partial f_H} & \frac{\partial H}{\partial f_H} \\ \frac{\partial V}{\partial V} & \frac{\partial \gamma}{\partial \gamma} & \frac{\partial q}{\partial q} & \frac{\partial \alpha}{\partial \alpha} & \frac{\partial H}{\partial H} \end{bmatrix} \begin{bmatrix} \dot{V} \\ \dot{\gamma} \\ \dot{q} \\ \dot{\alpha} \\ \dot{H} \end{bmatrix} = \begin{bmatrix} V - V_{eq} \\ \gamma - \gamma_{eq} \\ q - q_{eq} \\ \alpha - \alpha_{eq} \\ H - H_{eq} \end{bmatrix} + \begin{bmatrix} \frac{\partial f_V}{\partial \pi} & \frac{\partial f_V}{\partial \delta p} \\ \frac{\partial f_\gamma}{\partial \pi} & \frac{\partial f_\gamma}{\partial \delta p} \\ \frac{\partial f_q}{\partial \pi} & \frac{\partial f_q}{\partial \delta p} \\ \frac{\partial f_\alpha}{\partial \pi} & \frac{\partial f_\alpha}{\partial \delta p} \\ \frac{\partial f_H}{\partial \pi} & \frac{\partial f_H}{\partial \delta p} \\ \frac{\partial \pi}{\partial \pi} & \frac{\partial \delta p}{\partial \delta p} \end{bmatrix} \begin{bmatrix} \pi - \pi_{eq} \\ \delta p - \delta p_{eq} \end{bmatrix}$$

Linearização numérica

Para calcular a primeira coluna numericamente, podemos:

$$\begin{bmatrix} \frac{\partial f_V}{\partial f_\gamma} \\ \frac{\partial f_q}{\partial f_\alpha} \\ \frac{\partial f_H}{\partial V} \end{bmatrix} = \frac{f(V_{eq} + \Delta, \gamma_{eq}, q_{eq}, \alpha_{eq}, H_{eq}, \pi_{eq}, \delta p_{eq}) - f(V_{eq} - \Delta, \gamma_{eq}, q_{eq}, \alpha_{eq}, H_{eq}, \pi_{eq}, \delta p_{eq})}{2\Delta}$$

Desde que Δ seja suficientemente pequeno.
E assim sucessivamente para as demais colunas.

S-function

Na aula passada, escrevemos uma função com as equações do movimento longitudinal da aeronave. Exemplo:

$$\dot{X}_p = \text{dinamica}(t, X, U, \text{params})$$

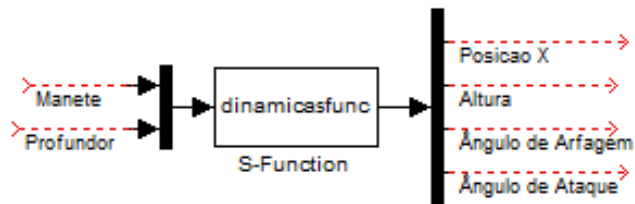
A simulação foi feita através da função ODE45:

$$[t, x] = \text{ode45}(@\text{dinamica}, [0 \ 40], X0, [], U, \text{params});$$

A utilização de s-functions permite a implementação da mesma dinâmica (escrita em M-FILE, C, C++ ou Fortran), em ambiente simulink.

Ver tutorial: <http://www.chem.mtu.edu/tbco/cm416/MatlabTutorialPart5.pdf>

S-function



S-function

Estrutura:

```
function [sys,x0,str,ts] = dinamicasfunc(t,x,U,flag, params, xinicial)
    switch flag,
        case 0
            [sys,x0,str,ts]=inicializa;
        case 1
            sys = dinamicalong(t,x,U, params);
        case 3
            sys = saidas(t,x,U);
        case { 2, 4, 9 }
            sys = [];
        otherwise
            error(['Unhandled flag = ',num2str(flag)]);
    end
end
```

S-function

```
function [sys,x0,str,ts] = inicializa(xinicial)
    sizes = simsizes;
    sizes.NumContStates = 6;
    sizes.NumDiscStates = 0;
    sizes.NumOutputs = 4;
    sizes.NumInputs = 2;
    sizes.DirFeedthrough = 0;
    sizes.NumSampleTimes = 1;
    sys = simsizes(sizes);

    x0 = xinicial;

    str = [];

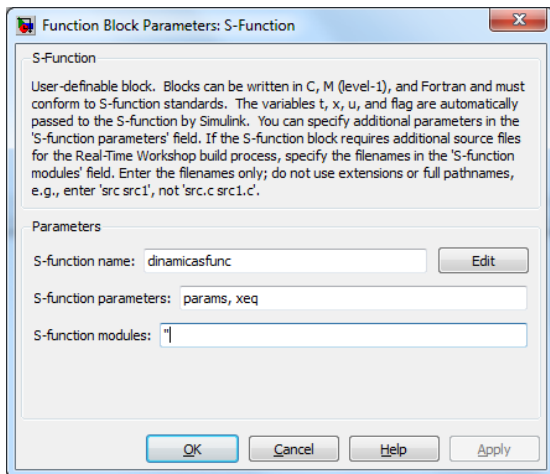
    ts = [0 0];

end
```

S-function

```
function sys = saidas(~,x,~)
    posicaoX = x(6);
    H = x(4);
    gamma = x(2);
    alpha = x(5);
    theta = gamma + alpha;
    sys = [posicaoX, -H, gamma + alpha, alpha*180/pi];
end
```

S-function



O Simulink permite interface com a saída gráfica do simulador FlightGear (a partir da versão 2.0 do Aerospace Blockset - Matlab R14SP2+ em diante, incluindo 2006a e mais recentes).

Download do flightgear: <http://www.flightgear.org/Downloads/>

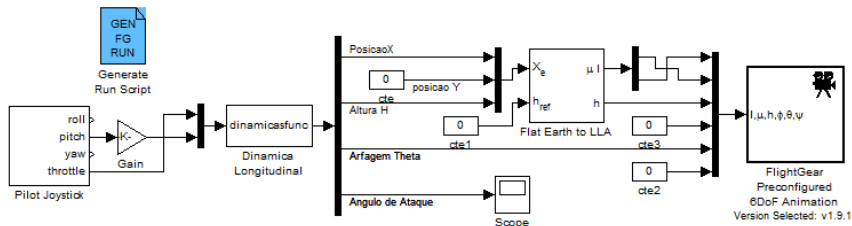
Blocos de interesse no Simulink (Aerospace Blockset -> Animation):

- **Generate Run Script** Gera arquivo .bat executável, usado para iniciar o FlightGear (obs.: incompatível com a versão mais recente do FlightGear, deve-se fazer algumas modificações no .bat - ver exemplo);
- **FlightGear Preconfigured 6DoF Animation:** Comunica ao FlightGear a posição e atitude da aeronave: possui como variáveis de entrada a latitude, longitude, altitude e ângulos representando a atitude da aeronave: ϕ , θ , ψ .

Na nossa dinâmica, utilizamos coordenadas cartesianas, com a consideração de Terra plana. Para converter em coordenadas geodésicas, usar o bloco: **Flat Earth to LLA**.

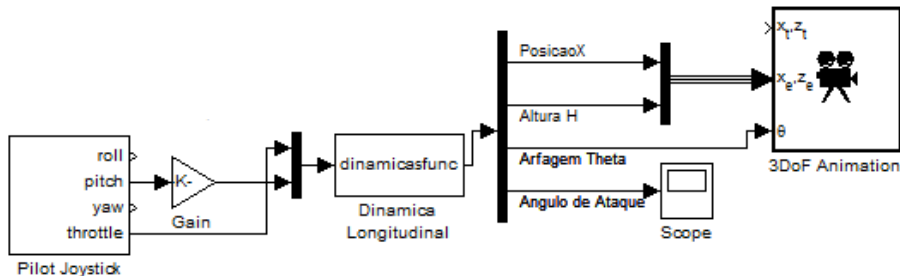
Outro bloco de interesse: **Pilot Joystick**. Permite a utilização de Joystick para entradas durante a simulação no simulink.

Exemplo que utiliza o Joystick, a s-function apresentada anteriormente, e o FlightGear como saída gráfica para a simulação:



Outra sugestão de saída gráfica. O Aerospace Toolbox possui uma saída gráfica mais simples: **3DoF Animation** e **6DoF Animation**. A vantagem é que não precisa instalar nenhum software adicional (e exige pouco do computador).

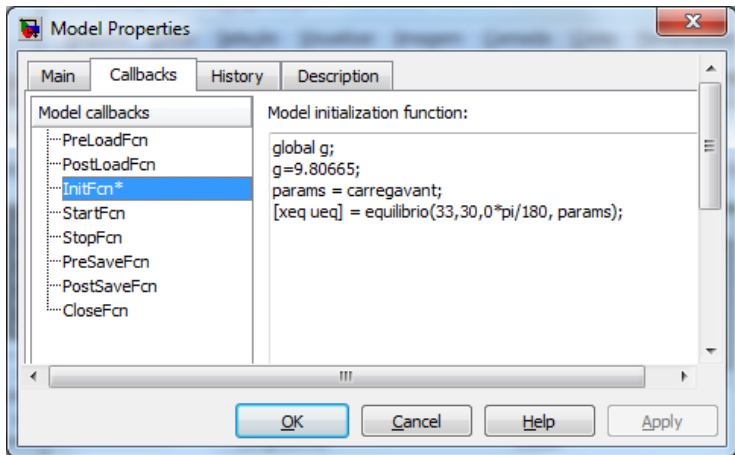
Exemplo:



Simulink - sugestões

Não esquecer de inicializar dados da aeronave. Sugestão: gravar dados da aeronave antes de começar a simulação.

Exemplo (ir em File-> Model Properties):



As variáveis aí carregadas poderão ser utilizadas no Simulink.

Tarefas para entregar

- Linearizar as equações do movimento (encontrar A e B) e obter os auto-valores da matriz A ($\text{eig}(A)$);
- Comparar a resposta do sistema linearizado com o não-linear para uma perturbação inicial de algum dos estados;
- Integrar dinâmica desenvolvida na aula passada ao Simulink através de S-function.

A animação no FlightGear é opcional.
Ver arquivos de exemplo.