



*Universidade Estadual de Campinas
Faculdade de Engenharia Mecânica
Departamento de Projeto Mecânico
Laboratório de Dinâmica de Estruturas e Máquinas*



Guia Prático de Matlab

Capítulo III

Integradores e representações de sistemas

Docente Responsável:

Prof. Dr. Milton Dias Jr.

Autores:

Hugo Heidy Miyasato

Leonardo Gimenes

Vinicius Gabriel Segala Simionatto

Campinas, 2009

Seção 3.1 – Modelos Lineares e Não Lineares

– Modelagem Linear com um engrenamento

Considere o sistema torsional a seguir, composto de 4 inércias, 2 conjuntos de molas e amortecedores em paralelo.

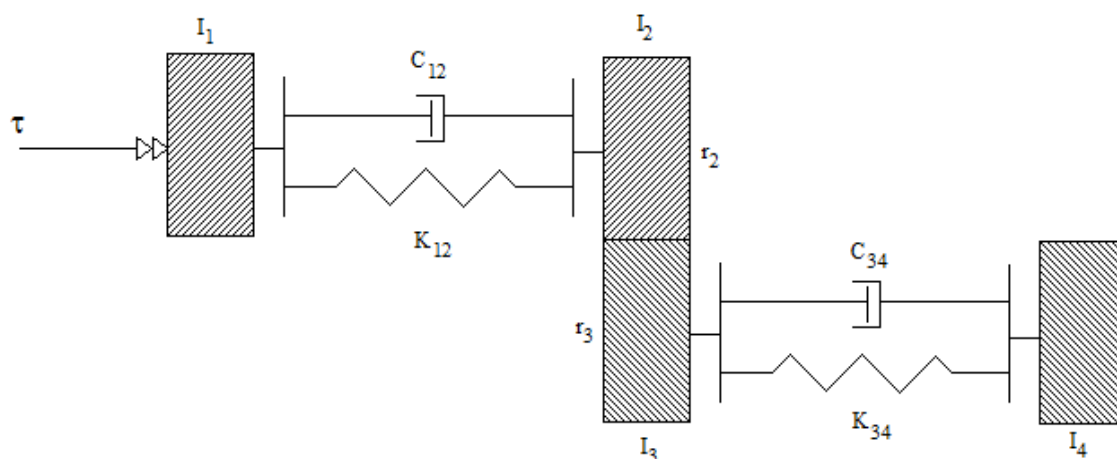


Figura 97: Modelo torsional simples, com um engrenamento ideal.

O primeiro passo para a resolução deste problema é identificar o número de graus de liberdade, e o número de restrições impostas pela configuração do problema.

Apesar de ter 4 discos (inércias), o problema apresenta apenas 3 graus de liberdade, pois entre as engrenagens 2 e 3 há uma restrição, à priori de velocidade. Ela ainda não pode ser considerada uma restrição de deslocamento, pois as engrenagens podem partir de referenciais de deslocamento distintos, e então, somente com os valores de deslocamento de cada variável, torna-se inviável extrair a relação de engrenamento.

De um modo mais simples, pode-se escrever que:

$$\dot{\theta}_2 r_2 = \dot{\theta}_3 r_3 \quad \text{ou} \quad \frac{\dot{\theta}_3}{\dot{\theta}_2} = \frac{r_2}{r_3} = \eta$$

Com η constante. Integrando esta equação, obtemos:

$$\theta_2 r_2 + C_2 = \theta_3 r_3 + C_3 \quad \text{ou} \quad \frac{\theta_3 r_3 + C_3}{\theta_2 r_2 + C_2} = \gamma$$

Com γ não constante e C_2 e C_3 constantes de integração. Assim, para que essa relação possa ser escrita em termos do deslocamento, a primeira hipótese que deve se fazer é a de que os referenciais de deslocamento são tais que $C_2 = C_3 = 0$.

Daí, podemos escrever a relação de engrenamento como:

$$\theta_2 r_2 = \theta_3 r_3 \quad \text{ou} \quad \frac{\theta_3}{\theta_2} = \frac{r_2}{r_3} = \eta$$

Além disso, as outras hipóteses a considerar são: o torque nas duas extremidades de uma mola linear e de um amortecedor linear têm a mesma intensidade, mesma direção e sentidos opostos (3ª lei de Newton).

O torque em uma mola torsional linear é dado por:

$$\tau = K(\theta_i - \theta_j)$$

O torque em um amortecedor torsional linear é dado por:

$$\tau = C(\dot{\theta}_i - \dot{\theta}_j)$$

Finalmente, para fins de montagem das equações de movimento do sistema, podemos imaginar uma situação onde:

$$\begin{cases} \theta_2 > \theta_1 & \dot{\theta}_2 > \dot{\theta}_1 \\ \theta_4 > \theta_3 & \dot{\theta}_4 > \dot{\theta}_3 \end{cases}$$

De onde, partindo das hipóteses sobre a 3ª lei de Newton, podemos extrair os seguintes diagramas de corpo livre (DCLs):

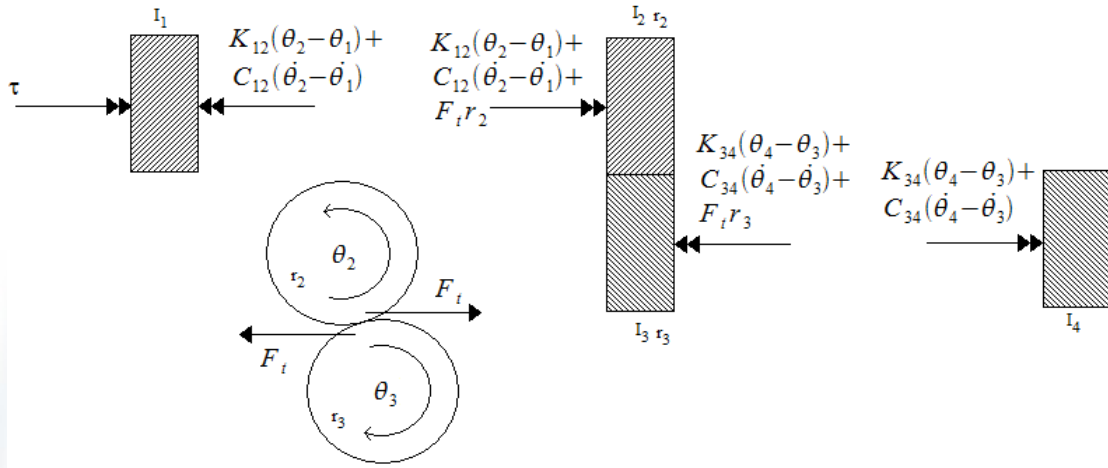


Figura 98: Diagrama de corpo livre do problema 1.

E finalmente montar as equações de movimento:

$$\begin{cases} I_1 \ddot{\theta}_1 = +K_{12}(\theta_2 - \theta_1) + C_{12}(\dot{\theta}_2 - \dot{\theta}_1) + \tau & (1) \\ I_2 \ddot{\theta}_2 = -K_{12}(\theta_2 - \theta_1) - C_{12}(\dot{\theta}_2 - \dot{\theta}_1) - F_t r_2 & (2) \\ I_3 \ddot{\theta}_3 = +K_{34}(\theta_4 - \theta_3) + C_{34}(\dot{\theta}_4 - \dot{\theta}_3) + F_t r_3 & (3) \\ I_4 \ddot{\theta}_4 = -K_{34}(\theta_4 - \theta_3) - C_{43}(\dot{\theta}_4 - \dot{\theta}_3) & (4) \end{cases}$$

Agora, o objetivo é, a partir das equações de movimento e da restrição de engrenamento, descobrir o valor da força de contato entre os dentes das engrenagens em função das variáveis de estado do sistema, e eliminar uma das equações de movimento, pois a variável de estado que seria calculada através da equação a ser removida, agora será calculada através da equação de restrição do sistema.

Dadas as hipóteses feitas no início do exercício, podemos assumir que em $t = 0$, $\theta_2 = \theta_3 = 0$. Assim, a condição de restrição é dada por

$$\theta_2 r_2 = \theta_3 r_3 \quad (5)$$

Considerando θ_2 como uma entrada e θ_3 como uma saída, a relação de transmissão η pode ser escrita como função de transferência. Assim:

$$\frac{\theta_3}{\theta_2} = \frac{r_2}{r_3} = \eta \quad (6)$$

Substituindo a equação (5) em (3), temos:

$$I_3 \ddot{\theta}_2 \frac{r_2}{r_3} + K_{34} \left(\theta_2 \frac{r_2}{r_3} - \theta_4 \right) + C_{34} \left(\dot{\theta}_2 \frac{r_2}{r_3} - \dot{\theta}_4 \right) = F_t r_3 \quad (7)$$

De onde pode-se descobrir o valor de F_t em função dos estados do sistema:

$$F_t = I_3 \ddot{\theta}_2 \frac{r_2}{r_3} + K_{34} \left(\theta_2 \frac{r_2}{r_3} - \frac{\theta_4}{r_3} \right) + C_{34} \left(\dot{\theta}_2 \frac{r_2}{r_3} - \frac{\dot{\theta}_4}{r_3} \right) \quad (8)$$

Substituindo (8) em (2):

$$I_2 \ddot{\theta}_2 + K_{12}(\theta_2 - \theta_1) + C_{12}(\dot{\theta}_2 - \dot{\theta}_1) = -I_3 \ddot{\theta}_2 \frac{r_2}{r_3} - K_{34} \left(\theta_2 \frac{r_2}{r_3} - \frac{\theta_4}{r_3} \right) - C_{34} \left(\dot{\theta}_2 \frac{r_2}{r_3} - \frac{\dot{\theta}_4}{r_3} \right) \quad (9)$$

Substituindo (6) em (9) e reorganizando os termos, temos:

$$(I_2 + I_3 \eta^2) \ddot{\theta}_2 + (C_{12} + C_{34} \eta^2) \dot{\theta}_2 + (K_{12} + K_{34} \eta^2) \theta_2 - K_{34} \eta \theta_4 - C_{34} \eta \dot{\theta}_4 - K_{12} \theta_1 - C_{12} \dot{\theta}_1 \quad (10)$$

Finalmente, a restrição (6) deve também ser substituída na equação (4), pois esta está em função de θ_3 , variável que deve ser calculada a partir da equação de restrição, e não mais de equações de movimento. Fazendo esta substituição, encontramos:

$$I_4 \ddot{\theta}_4 = -K_{34}(\theta_4 - \theta_2 \eta) - C_{43}(\dot{\theta}_4 - \dot{\theta}_2 \eta) \quad (11)$$

Assim, as equações (1), (10) e (11) formam um novo sistema de equações diferenciais ordinárias de movimento que já considera a restrição do engrenamento. Matricialmente, este sistema é escrito como:

$$\begin{bmatrix} I_1 & 0 & 0 \\ 0 & I_2 + I_3 \eta^2 & 0 \\ 0 & 0 & I_4 \end{bmatrix} \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \\ \ddot{\theta}_4 \end{bmatrix} + \begin{bmatrix} C_{12} & -C_{12} & 0 \\ -C_{12} & C_{12} + C_{34} \eta^2 & -C_{34} \eta \\ 0 & -C_{34} \eta & C_{34} \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_4 \end{bmatrix} + \begin{bmatrix} K_{12} & -K_{12} & 0 \\ -K_{12} & K_{12} + K_{34} \eta^2 & -K_{34} \eta \\ 0 & -K_{34} \eta & K_{34} \end{bmatrix} \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_4 \end{bmatrix} = \begin{bmatrix} \tau \\ 0 \\ 0 \end{bmatrix}$$

– Modelagem com não-linearidades

Considerando um modelo similar ao do exemplo 1, porém com uma não-linearidade de embreagem F_C entre as inércias 1 e 2, além de uma não-linearidade de folga de dente F_T entre as inércias 3 e 4:

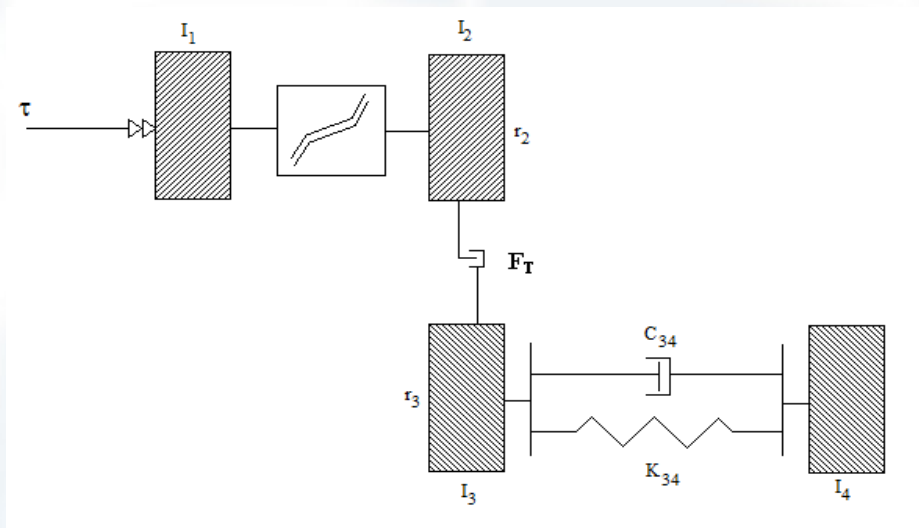


Figura 99: Modelo torsional não linear, com folga entre dentes e embreagem.

Os dispositivos de não linearidades considerados na modelagem trazem um nível de detalhamento ao sistema que o modelo linear não pode trazer, como folgas em engrenamentos, rigidez e amortecimento variáveis, amortecimento histerético, entre outros.

Bem como dispositivos lineares, estes também obedecem à lei da ação e reação. Assim sendo, a intensidade da força nas extremidades do dispositivo é a mesma e seu sentido é oposto.

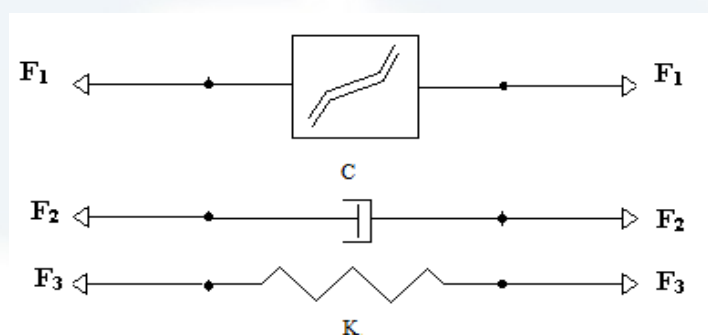


Figura 100: Efeitos práticos da 3ª lei de Newton.

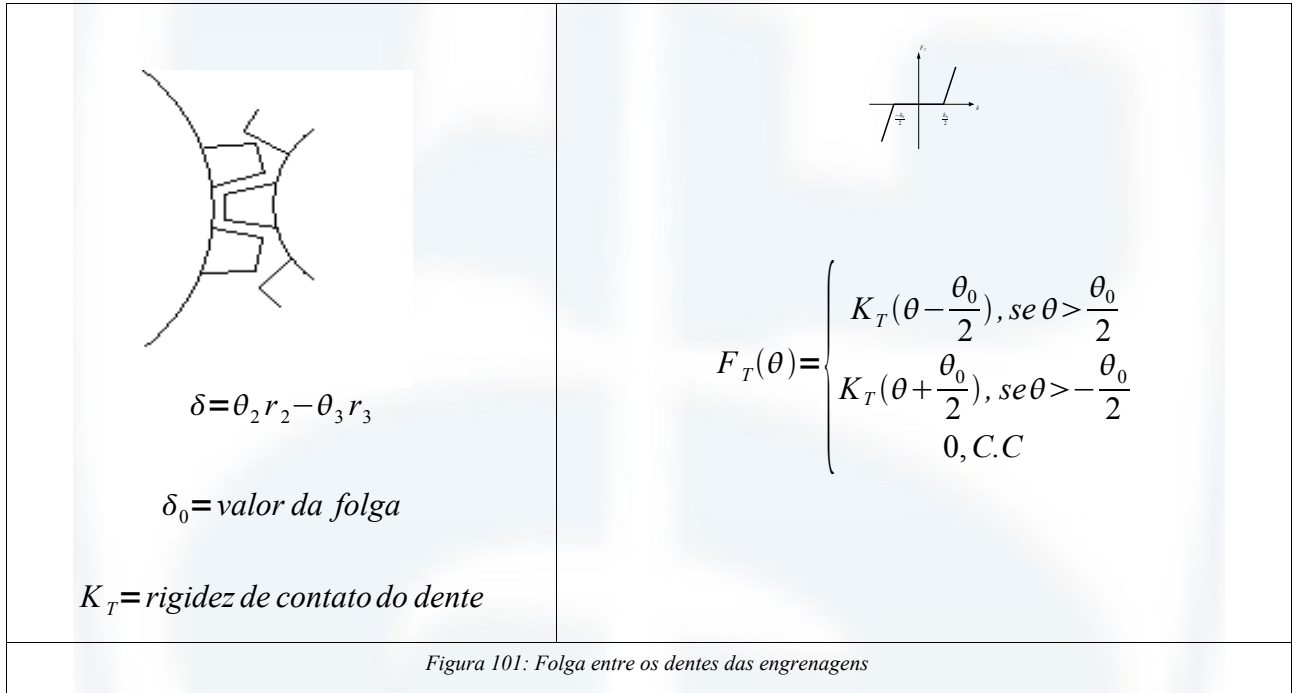
A partir desta hipótese, pode-se extrair as equações de movimento:

$$\begin{cases} I_1 \ddot{\theta}_1 = \tau - F_c(\theta_1, \theta_2) \\ I_2 \ddot{\theta}_2 = F_c(\theta_1, \theta_2) - F_T r_2 \\ I_3 \ddot{\theta}_3 = F_T r_3 - K_{34}(\theta_3 - \theta_4) - C_{34}(\dot{\theta}_3 - \dot{\theta}_4) \\ I_4 \ddot{\theta}_4 = K_{34}(\theta_3 - \theta_4) + C_{34}(\dot{\theta}_3 - \dot{\theta}_4) \end{cases}$$

Neste sistema, não há restrições. Assim, resta escrever as expressões das não-linearidades:

- *Modelagem de engrenamento com folga*

No caso do dente, a força de contato deve considerar também a folga. Assim, temos:



– *Rigidez Estratificada*

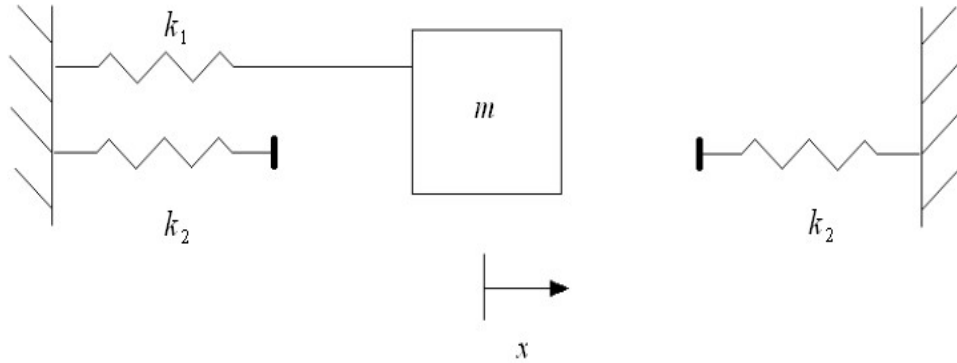


Figura 102: Rigidez estratificada

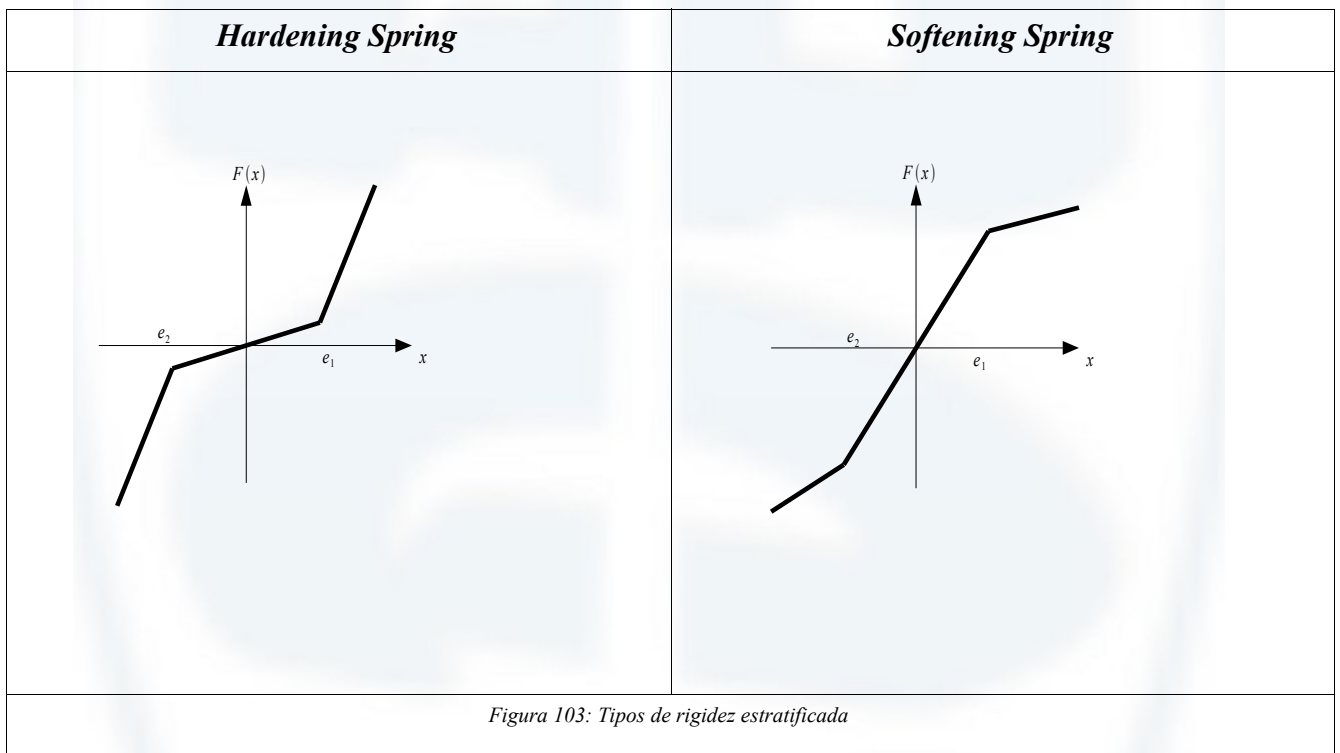


Figura 103: Tipos de rigidez estratificada

$$F(x) = \begin{cases} k_1 x, & \text{se } e_2 \leq x \leq e_1 \\ k_1 x + k_2(x - e_1) = (k_1 + k_2)x - k_2 e_1, & \text{se } x > e_1 \\ k_1 x + k_2(x - e_2) = (k_1 + k_2)x - k_2 e_2, & \text{se } x < e_2 \end{cases}$$

– *Atrito e histerese*

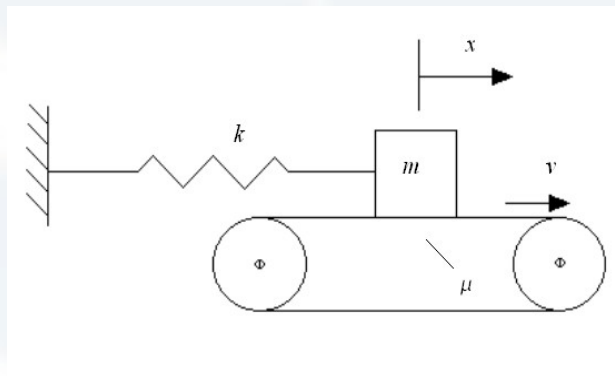


Figura 104: Modelo com atrito, para evidenciar o fenômeno de stick-slip

Em um modelagem mais simples, a direção da força de atrito dinâmico depende da velocidade relativa entre \dot{x} e v :

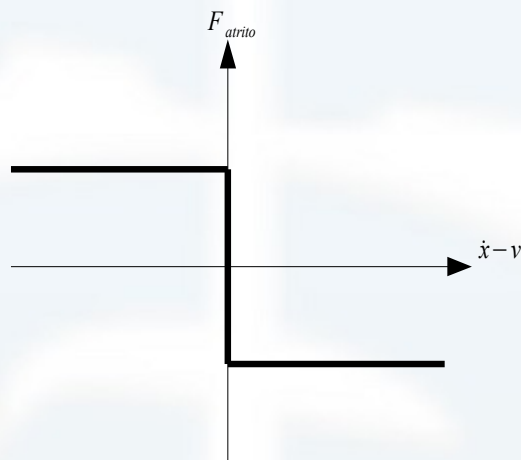
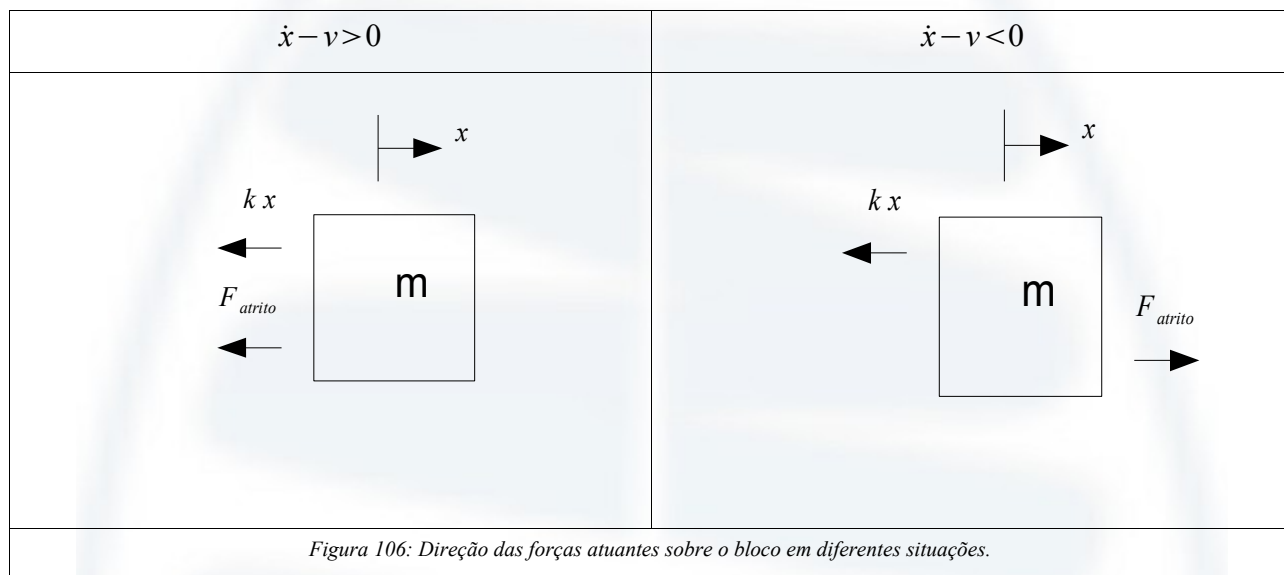


Figura 105: Força de atrito versus velocidade relativa

$$F_{\text{atrito}}(x) = \begin{cases} -F_{at}, & \text{se } \dot{x} - v > 0 \\ F_{at}, & \text{se } \dot{x} - v < 0 \end{cases}$$



Caso a velocidade relativa entre o bloco e a esteira seja nula, a força de atrito é equilibrada pela força da mola. Porém, devido às variações entre as velocidades relativas pode ocorrer o escorregamento entre as superfícies, originando o fenômeno vibratório conhecido como *stick-slip*.

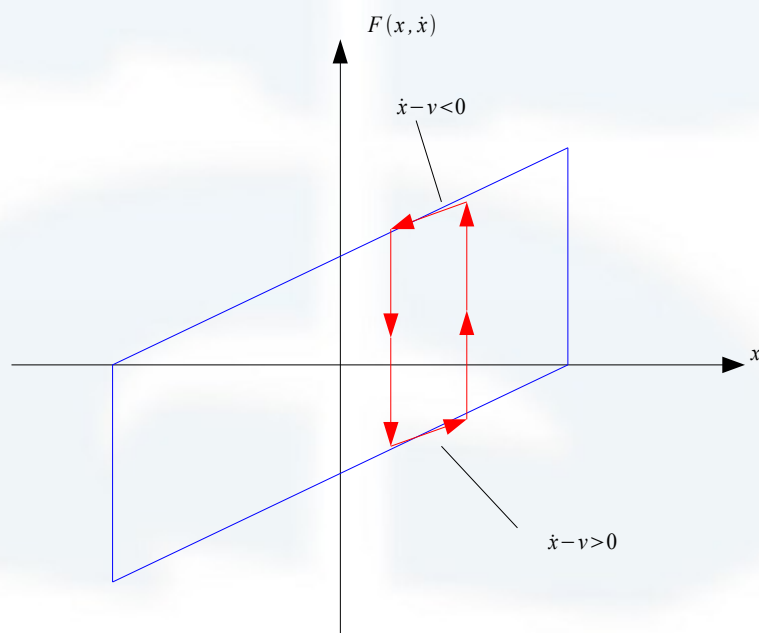


Figura 107: Ciclo de histerese causado pelas forças de atrito.

A não-linearidade da embreagem considera rigidez estratificada com histerese, ou seja, dependendo do sentido do movimento relativo, a rigidez é a mesma, mas o torque aplicado é diferente.

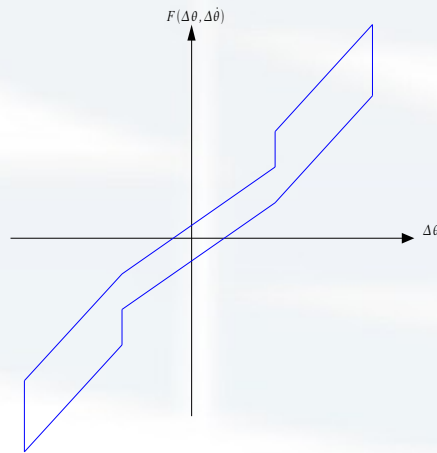


Figura 108: Combinação de stick-slip e rigidez estratificada na curva de força versus deslocamento relativo.

Finalmente, trata-se os torques causados pelas não-linearidades como excitações dependentes das variáveis de estado do sistema, ocasionando posições nulas nas matrizes de rigidez e amortecimento.

Equações de movimento na forma matricial:

$$\begin{pmatrix} I_1 & 0 & 0 & 0 \\ 0 & I_2 & 0 & 0 \\ 0 & 0 & I_3 & 0 \\ 0 & 0 & 0 & I_4 \end{pmatrix} \begin{pmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \\ \ddot{\theta}_3 \\ \ddot{\theta}_4 \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & C_{34} & -C_{34} \\ 0 & 0 & -C_{34} & C_{34} \end{pmatrix} \begin{pmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \\ \dot{\theta}_4 \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & K_{34} & -K_{34} \\ 0 & 0 & -K_{34} & K_{34} \end{pmatrix} \begin{pmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \end{pmatrix} = \begin{pmatrix} \tau \\ 0 \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} -F_C \\ F_C - F_T r_2 \\ F_T r_3 \\ 0 \end{pmatrix}$$

– Modelagem Linear com dois engrenamentos

O objetivo deste problema é fazer a modelagem linear de um trem de potência, considerando um veículo com tração traseira, transmissão, diferencial, eixo cardan e semi-eixos flexíveis e com amortecimento interno, embreagem linear (com rigidez constante e amortecimento viscoso) e inércia no volante de inércia, na transmissão, no diferencial, e nas rodas (incluindo a inércia do veículo). Além disso, serão considerados carregamentos externos como a resistência do ar e a inclinação da pista. O veículo não faz curvas neste percurso. Observe o esquema abaixo:

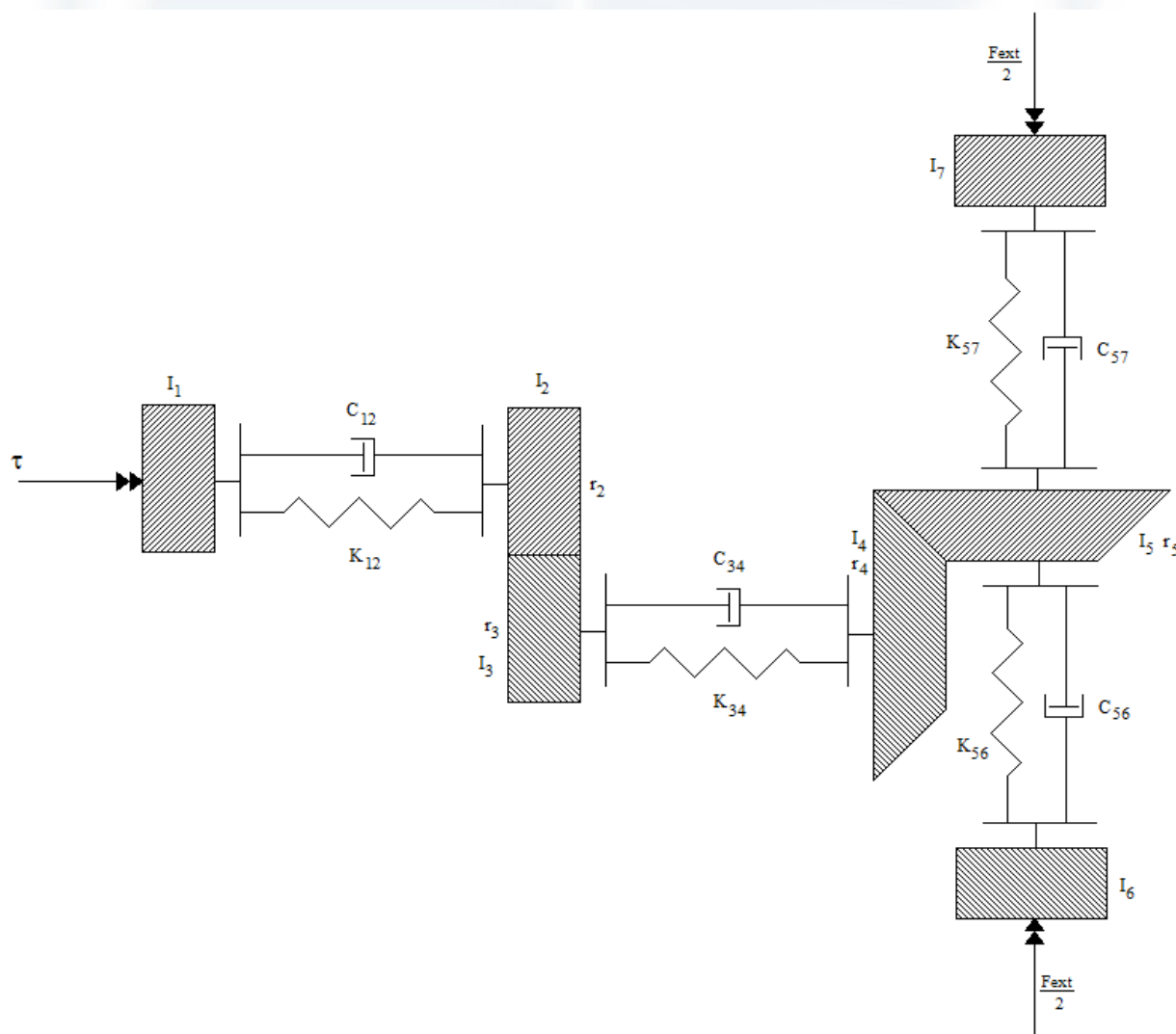


Figura 109: Modelo torsional linear considerando 2 engrenamentos ideais.

Considerar que o veículo não faz curvas, significa dizer que o momento resultante no veículo em torno de seu eixo vertical, que passa pelo CG é nulo. Sendo assim, a partir da análise estática simples, conclui-se que as forças propulsoras do veículo, em cada uma das rodas (traseiras) é igual.

Deste modo, ao considerar uma força externa agindo no veículo, pode-se dizer que metade desta força atuará em cada uma das duas rodas traseiras. Outra consideração importante a fazer é a de que

não ocorre deslizamento das rodas traseiras, com relação ao piso.

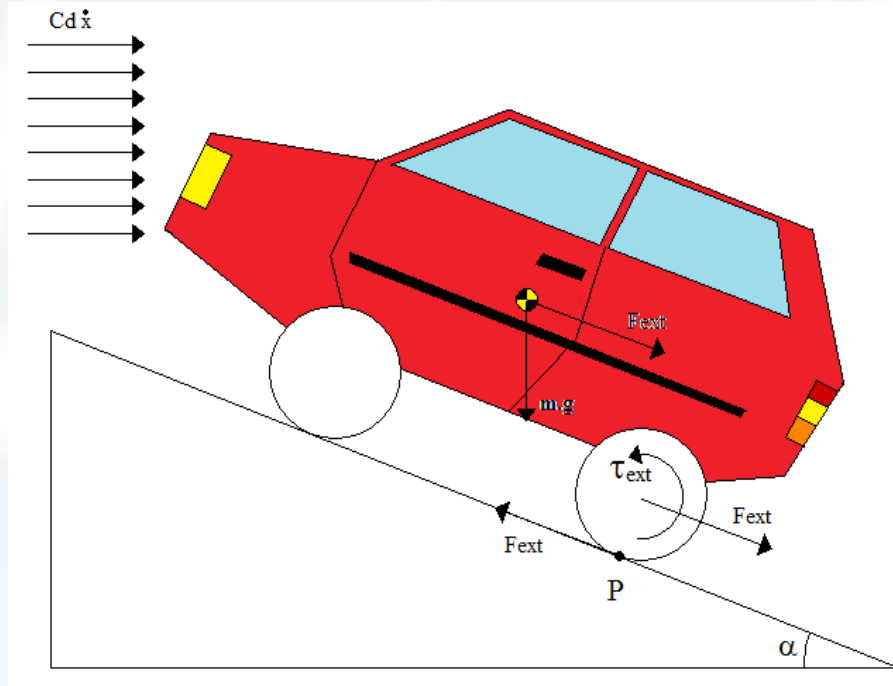


Figura 110: Veículo com carregamento externo e forças de reação no veículo e nas rodas.

Sendo assim, pode-se fazer a somatória dos momentos, em torno no ponto P, de modo que se encontre o torque causado no sistema, relativo às forças externas, o que resulta em:

$$\tau_{ext} = -\frac{F_{ext} r_R}{2} \quad (1)$$

Como forças externas, serão considerados a inclinação da pista e a resistência do ar. Para isso, considera-se o coeficiente de arraste aerodinâmico do veículo (C_d) e o ângulo de inclinação da pista α .

Assim, considerando x o deslocamento linear do veículo, pode-se escrever as forças externas como:

$$F_{ext} = C_d \dot{x} + m g \sin \alpha \quad (2)$$

Ainda sob as hipóteses de que as rodas não deslizam e que o veículo não faz curvas, é possível escrever a restrição de velocidade entre as rodas do veículo e o movimento translacional do mesmo, ou seja:

$$\dot{x} = r_R \dot{\theta}_6 = r_R \dot{\theta}_7 \quad (3)$$

Substituindo (2) e (3) em (1), obtemos, para cada roda, a expressão do torque devido às forças externas:

$$\tau_{ext6} = -\frac{C_d r_R^2}{2} \dot{\theta}_6 - \frac{m g r_R \sin \alpha}{2} \quad (4)$$

$$\tau_{ext7} = -\frac{C_d r_R^2}{2} \dot{\theta}_7 - \frac{m g r_R \sin \alpha}{2} \quad (5)$$

Para efeitos de montagem de equações de movimento, é suposta uma condição onde:

$$\begin{cases} \theta_2 > \theta_1 & \dot{\theta}_2 > \dot{\theta}_1 \\ \theta_4 > \theta_3 & \dot{\theta}_4 > \dot{\theta}_3 \\ \theta_6 > \theta_5 & \dot{\theta}_6 > \dot{\theta}_5 \\ \theta_7 > \theta_5 & \dot{\theta}_7 > \dot{\theta}_5 \end{cases}$$

Assim, escreve-se as seguintes equações de movimento:

$$I_1 \ddot{\theta}_1 = +K_{12}(\theta_2 - \theta_1) + C_{12}(\dot{\theta}_2 - \dot{\theta}_1) + \tau \quad (6)$$

$$I_2 \ddot{\theta}_2 = -K_{12}(\theta_2 - \theta_1) - C_{12}(\dot{\theta}_2 - \dot{\theta}_1) - F_{T23} r_2 \quad (7)$$

$$I_3 \ddot{\theta}_3 = +K_{34}(\theta_4 - \theta_3) + C_{34}(\dot{\theta}_4 - \dot{\theta}_3) + F_{T23} r_3 \quad (8)$$

$$I_4 \ddot{\theta}_4 = -K_{34}(\theta_4 - \theta_3) - C_{34}(\dot{\theta}_4 - \dot{\theta}_3) - F_{T45} r_4 \quad (9)$$

$$I_5 \ddot{\theta}_5 = +K_{56}(\theta_6 - \theta_5) + K_{57}(\theta_7 - \theta_5) + C_{56}(\dot{\theta}_6 - \dot{\theta}_5) + C_{57}(\dot{\theta}_7 - \dot{\theta}_5) + F_{T45} r_5 \quad (10)$$

$$I_6 \ddot{\theta}_6 = -K_{56}(\theta_6 - \theta_5) - C_{56}(\dot{\theta}_6 - \dot{\theta}_5) - \frac{C_d r_R^2}{2} \dot{\theta}_6 - \frac{m g r_R \sin \alpha}{2} \quad (11)$$

$$I_7 \ddot{\theta}_7 = -K_{57}(\theta_7 - \theta_5) - C_{57}(\dot{\theta}_7 - \dot{\theta}_5) - \frac{C_d r_R^2}{2} \dot{\theta}_7 - \frac{m g r_R \sin \alpha}{2} \quad (12)$$

Também utilizando as mesmas premissas do primeiro exercício, São escritas as restrições de engrenamento:

$$\theta_2 r_2 = \theta_3 r_3 \quad (13)$$

$$\theta_4 r_4 = \theta_5 r_5 \quad (14)$$

De onde se extrai as relações de transmissão:

$$\frac{\theta_3}{\theta_2} = \frac{r_2}{r_3} = \eta_{23} \quad (15)$$

$$\frac{\theta_5}{\theta_4} = \frac{r_4}{r_5} = \eta_{45} \quad (16)$$

Agora, o objetivo é, a partir das equações de movimento e da restrições de engrenamento, descobrir o valor das forças de contato entre os dentes das engrenagens em função das variáveis de estado do sistema, e eliminar duas das equações de movimento, pois as variáveis de estado que seriam calculadas através das equações a serem removidas, agora serão calculadas através das equações de restrição do sistema.

Substituindo (13) em (8):

$$I_3 \frac{r_2}{r_3} \ddot{\theta}_2 + K_{34} \left(\theta_2 \frac{r_2}{r_3} - \theta_4 \right) + C_{34} \left(\dot{\theta}_2 \frac{r_2}{r_3} - \dot{\theta}_4 \right) = F_{T23} r_3 \quad (17)$$

Portanto:

$$F_{T23} = I_3 \frac{r_2}{r_3} \ddot{\theta}_2 + K_{34} \left(\theta_2 \frac{r_2}{r_3} - \frac{\theta_4}{r_3} \right) + C_{34} \left(\dot{\theta}_2 \frac{r_2}{r_3} - \frac{\dot{\theta}_4}{r_3} \right) \quad (18)$$

Substituindo (18) em (7) e reorganizando os termos:

$$\left(I_2 + I_3 \frac{r_2^2}{r_3^2} \right) \ddot{\theta}_2 + \left(C_{12} + C_{34} \frac{r_2^2}{r_3^2} \right) \dot{\theta}_2 + \left(K_{12} + K_{34} \frac{r_2^2}{r_3^2} \right) \theta_2 - K_{12} \theta_1 - C_{12} \dot{\theta}_1 - K_{34} \frac{r_2}{r_3} \theta_4 - C_{34} \frac{r_2}{r_3} \dot{\theta}_4 = 0 \quad (19)$$

Substituindo (15) em (19), temos:

$$\left(I_2 + I_3 \eta_{23}^2 \right) \ddot{\theta}_2 + \left(C_{12} + C_{34} \eta_{23}^2 \right) \dot{\theta}_2 + \left(K_{12} + K_{34} \eta_{23}^2 \right) \theta_2 - K_{12} \theta_1 - C_{12} \dot{\theta}_1 - K_{34} \eta_{23} \theta_4 - C_{34} \eta_{23} \dot{\theta}_4 = 0 \quad (20)$$

Exatamente o mesmo processo será feito para a restrição de engrenamento entre as engrenagens 4 e 5. Agora, substituindo (14) em (10):

$$I_5 \frac{r_4}{r_5} \ddot{\theta}_4 - K_{56} \left(\theta_6 - \theta_4 \frac{r_4}{r_5} \right) - K_{57} \left(\theta_7 - \theta_4 \frac{r_4}{r_5} \right) - C_{56} \left(\dot{\theta}_6 - \dot{\theta}_4 \frac{r_4}{r_5} \right) - C_{57} \left(\dot{\theta}_7 - \dot{\theta}_4 \frac{r_4}{r_5} \right) = F_{T45} r_5 \quad (21)$$

Portanto:

$$F_{T45} = I_5 \frac{r_4}{r_5^2} \ddot{\theta}_4 - K_{56} \left(\frac{\theta_6}{r_5} - \theta_4 \frac{r_4}{r_5^2} \right) - K_{57} \left(\frac{\theta_7}{r_5} - \theta_4 \frac{r_4}{r_5^2} \right) - C_{56} \left(\frac{\dot{\theta}_6}{r_5} - \dot{\theta}_4 \frac{r_4}{r_5^2} \right) - C_{57} \left(\frac{\dot{\theta}_7}{r_5} - \dot{\theta}_4 \frac{r_4}{r_5^2} \right) \quad (22)$$

Substituindo (22) e (13) em (9):

$$\begin{aligned} I_4 \ddot{\theta}_4 + K_{34} (\theta_4 - \theta_2 \frac{r_2}{r_3}) + C_{34} (\dot{\theta}_4 - \dot{\theta}_2 \frac{r_2}{r_3}) = & -I_5 \frac{r_4^2}{r_5^2} \ddot{\theta}_4 + K_{56} \left(\theta_6 \frac{r_4}{r_5} - \theta_4 \frac{r_4^2}{r_5^2} \right) + K_{57} \left(\theta_7 \frac{r_4}{r_5} - \theta_4 \frac{r_4^2}{r_5^2} \right) \dots \\ \dots + C_{56} \left(\dot{\theta}_6 \frac{r_4}{r_5} - \dot{\theta}_4 \frac{r_4^2}{r_5^2} \right) + C_{57} \left(\dot{\theta}_7 \frac{r_4}{r_5} - \dot{\theta}_4 \frac{r_4^2}{r_5^2} \right) \end{aligned} \quad (23)$$

Finalmente, substituindo (16) e (15) em (23), obtém-se:

$$\begin{aligned} (I_4 + I_5 \eta_{45}^2) \ddot{\theta}_4 + (C_{34} + C_{56} \eta_{45}^2 + C_{57} \eta_{45}^2) \dot{\theta}_4 + (K_{34} + K_{56} \eta_{45}^2 + K_{57} \eta_{45}^2) \theta_4 - K_{34} \eta_{23} \theta_2 \dots \\ \dots - K_{56} \eta_{45} \theta_6 - K_{57} \eta_{45} \theta_7 - C_{34} \eta_{23} \dot{\theta}_2 - C_{56} \eta_{45} \dot{\theta}_6 - C_{57} \eta_{45} \dot{\theta}_7 = 0 \end{aligned} \quad (24)$$

As equações (11) e (12) ainda estão em função de variáveis de estado do sistema que serão calculadas através das restrições de engrenamento. Portanto, estas variáveis devem ser substituídas por variáveis de estado do sistema que serão calculadas pelas equações de movimento. Sendo assim, substituindo (16) em (11) e (16) em (12), temos:

$$\begin{cases} I_6 \ddot{\theta}_6 = -K_{56} (\theta_6 - \theta_4 \eta_{45}) - C_{56} (\dot{\theta}_6 - \dot{\theta}_4 \eta_{45}) - \frac{C_d r_R^2}{2} \dot{\theta}_6 - \frac{m g r_R \sin \alpha}{2} & (25) \\ I_7 \ddot{\theta}_7 = -K_{57} (\theta_7 - \theta_4 \eta_{45}) - C_{57} (\dot{\theta}_7 - \dot{\theta}_4 \eta_{45}) - \frac{C_d r_R^2}{2} \dot{\theta}_7 - \frac{m g r_R \sin \alpha}{2} & (26) \end{cases}$$

Assim, com as equações (6), (20), (24), (25) e (26), pode-se montar um novo sistema de equações de movimento, que escrito na forma matricial pode ser explicitado como:

$$\begin{bmatrix} I_1 & 0 & 0 & 0 & 0 \\ 0 & I_2 + I_3 \eta_{23}^2 & 0 & 0 & 0 \\ 0 & 0 & I_4 + I_5 \eta_{45}^2 & 0 & 0 \\ 0 & 0 & 0 & I_6 & 0 \\ 0 & 0 & 0 & 0 & I_7 \end{bmatrix} \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \\ \ddot{\theta}_4 \\ \ddot{\theta}_6 \\ \ddot{\theta}_7 \end{bmatrix} + \dots$$

$$\dots \begin{bmatrix} C_{12} & -C_{12} & 0 & 0 & 0 \\ -C_{12} & C_{12} + C_{34} \eta_{23}^2 & -C_{34} \eta_{23} & 0 & 0 \\ 0 & -C_{34} \eta_{23} & C_{34} + C_{56} \eta_{45}^2 + C_{57} \eta_{45}^2 & -C_{56} \eta_{45} & -C_{57} \eta_{45} \\ 0 & 0 & -C_{56} \eta_{45} & C_{56} + \frac{C_d r_R^2}{2} & 0 \\ 0 & 0 & -C_{57} \eta_{45} & 0 & C_{57} + \frac{C_d r_R^2}{2} \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_4 \\ \dot{\theta}_6 \\ \dot{\theta}_7 \end{bmatrix} + \dots$$

$$\dots \begin{bmatrix} K_{12} & -K_{12} & 0 & 0 & 0 \\ -K_{12} & K_{12} + K_{34} \eta_{23}^2 & -K_{34} \eta_{23} & 0 & 0 \\ 0 & -K_{34} \eta_{23} & K_{34} + K_{56} \eta_{45}^2 + K_{57} \eta_{45}^2 & -K_{56} \eta_{45} & -K_{57} \eta_{45} \\ 0 & 0 & -K_{56} \eta_{45} & K_{56} + \frac{C_d r_R^2}{2} & 0 \\ 0 & 0 & -K_{57} \eta_{45} & 0 & K_{57} + \frac{C_d r_R^2}{2} \end{bmatrix} \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_4 \\ \theta_6 \\ \theta_7 \end{bmatrix} = \begin{bmatrix} \tau \\ 0 \\ 0 \\ -\frac{m g r_R}{2} \sin \alpha \\ -\frac{m g r_R}{2} \sin \alpha \end{bmatrix}$$

Seção 3.2 - Integradores ODE

Na maioria dos problemas que envolvem simulação de sistemas, sejam estes lineares ou não lineares, a resposta do sistema a uma excitação qualquer pode ser difícil de se encontrar analiticamente. Por meio de métodos numéricos, é possível descobrir uma resposta satisfatoriamente próxima à resposta analítica do sistema em questão. Por isso, os integradores do Matlab tem sido desenvolvidos extensivamente.

Nesta seção, serão tratados os integradores da família ODE.

Ordem dos integradores e formulação de estados

Neste ambiente, os integradores são todos de primeira ordem. Isto significa que eles resolverão problemas do tipo:

$$\{\dot{x}\} = [A]\{x\} + G(t, x) = H(t, x)$$

O que é a formulação típica de um sistema de equações diferenciais ordinárias de primeira ordem. Porém, é muito comum que os problemas estudados envolvam equações de ordens superiores, e por isso, precisaremos manipular estes sistemas, de forma que eles consigam representar o problema proposto, a partir de um sistema maior, de ordem um.

Esta manipulação é chamada de “Formulação de Estados”, e é importante ressaltar que existe mais de uma maneira de fazê-la.

Sistemas de segunda ordem

Os sistemas mais estudados na área de dinâmica e vibrações são os sistemas de segunda ordem. Deles fazem parte n equações diferenciais ordinárias de segunda ordem, que incluem as forças de inércia, rigidez e amortecimento. Estes sistemas podem ser escritos da seguinte forma:

$$[M]\{\ddot{x}\} + [C]\{\dot{x}\} + [K]\{x\} = F(t) + U(x, \dot{x})$$

Onde M , C e K são as matrizes de inércia, amortecimento e rigidez, respectivamente, F são as forças excitadoras externas, e U são as forças devido às não-linearidades do sistema.

Forma de estados 1

Especificamente, o vetor x pode ser definido como:

$$\{x\} = \begin{Bmatrix} x_1 \\ \vdots \\ x_n \end{Bmatrix} \quad \{\dot{x}\} = \begin{Bmatrix} \dot{x}_1 \\ \vdots \\ \dot{x}_n \end{Bmatrix} \quad \{\ddot{x}\} = \begin{Bmatrix} \ddot{x}_1 \\ \vdots \\ \ddot{x}_n \end{Bmatrix}$$

Com isso, podemos definir o vetor de variáveis de estado como:

$$\{y\} = \begin{Bmatrix} x \\ \dot{x} \end{Bmatrix} = \begin{Bmatrix} y_1 \\ \vdots \\ y_n \\ y_{n+1} \\ \vdots \\ y_{2n} \end{Bmatrix} = \begin{Bmatrix} x_1 \\ \vdots \\ x_n \\ \dot{x}_1 \\ \vdots \\ \dot{x}_n \end{Bmatrix}$$

E a derivada do vetor de estados fica da forma:

$$\{\dot{y}\} = \begin{Bmatrix} \dot{x} \\ \ddot{x} \end{Bmatrix} = \begin{Bmatrix} \dot{y}_1 \\ \vdots \\ \dot{y}_n \\ \dot{y}_{n+1} \\ \vdots \\ \dot{y}_{2n} \end{Bmatrix} = \begin{Bmatrix} \dot{x}_1 \\ \vdots \\ \dot{x}_n \\ \ddot{x}_1 \\ \vdots \\ \ddot{x}_n \end{Bmatrix}$$

De onde podemos extrair a relação:

$$\begin{Bmatrix} \dot{x}_1 \\ \vdots \\ \dot{x}_n \end{Bmatrix} = \begin{Bmatrix} \dot{y}_1 \\ \vdots \\ \dot{y}_n \end{Bmatrix} = \begin{Bmatrix} y_{n+1} \\ \vdots \\ y_{2n} \end{Bmatrix}$$

Já a equação de movimento pode ser reorganizada da seguinte forma:

$$\{\ddot{x}\} = -[M]^{-1}[C]\{\dot{x}\} - [M]^{-1}[K]\{x\} + [M]^{-1}F(t) + [M]^{-1}U(x, \dot{x})$$

Unindo a relação e a equação de movimento reorganizada, podemos escrever ambas em uma só equação matricial:

$$\begin{Bmatrix} \dot{x} \\ \ddot{x} \end{Bmatrix} = \begin{bmatrix} [0]_{n \times n} & [I]_{n \times n} \\ -[M]^{-1}[K] & -[M]^{-1}[C] \end{bmatrix} \begin{Bmatrix} x \\ \dot{x} \end{Bmatrix} + \begin{Bmatrix} \{0\}_{n \times 1} \\ [M]^{-1}(F(t) + U(x, \dot{x})) \end{Bmatrix}$$

Que também pode ser escrita na seguinte forma:

$$\{\dot{y}\} = [A]\{y\} + F(t, y) = H(t, y)$$

Onde:

$$\{y\} = \begin{Bmatrix} x \\ \dot{x} \end{Bmatrix}_{2n \times 1}$$

$$\{\dot{y}\} = \begin{Bmatrix} \dot{x} \\ \ddot{x} \end{Bmatrix}_{2n \times 1}$$

$$[A] = \begin{bmatrix} [0]_{n \times n} & [I]_{n \times n} \\ -[M]^{-1}[K] & -[M]^{-1}[C] \end{bmatrix}_{2n \times 2n}$$

$$F(t, y) = \begin{Bmatrix} [0]_{n \times 1} \\ [M]^{-1}(F(t) + U(y)) \end{Bmatrix}_{2n \times 1}$$

Que é a forma exata que o Matlab necessita para uso em integradores da família ODE.

Função para integração

Um dos parâmetros de entrada de um integrador é uma função que representará a derivada temporal do vetor de estados y , que aqui chamamos de $H(t, y)$. Como sugere o problema, esta função deve ter como parâmetro de saída um vetor que representa a derivada temporal do vetor y , e como entrada, o vetor y , que representa o estado do sistema no passo atual de integração, e a variável t , que representa o tempo atual de integração. Além destes parâmetros, outros parâmetros também podem ser passados como entrada para esta função. Porém, sua saída deve permanecer sempre a mesma.

IMPORTANTE: Na implementação da função, o primeiro e o segundo parâmetros de entrada devem ser **sempre** a variável t e o vetor y , respectivamente.

Exemplo

```
function dy = sistema1(t,y)

% Esta função representa um sistema de primeira ordem, onde o parâmetro de
% saída (dy) é a derivada do vetor de estados y, e neste exemplo simples,
% os parâmetros de entrada são somente o tempo t e o vetor de estados y.

% Parâmetros do sistema:
m = 10;
c = 1;
k = 4;

% Parâmetros da excitação:
w = 2*pi*3;
amp = 2;
```

```
% Forma de estados:
A = [ 0 1
      -k/m -c/m];

F = [0
      amp*sin(w*t)/m];

dy = A*y+F;
```

Lembre-se: Pelo fato de esta função se chamar *sistema1*, ela deve ser salva em um arquivo chamado *sistema.m*, na sua pasta de trabalho. Esta função pode ter qualquer nome que você julgar conveniente, desde que o nome do arquivo siga a mesma convenção.

Em alguns casos, desejamos fazer funções mais genéricas para o integrador, de forma que possamos alterar as características do sistema em estudo, e realizar novas integrações, sem fazer novas alterações nesta função. Isto pode ser feito passando as matrizes do sistema como argumentos de entrada desta função, e criando uma nova função, cujo parâmetro de entrada será o tempo, e o parâmetro de saída será um vetor que representa as forças externas (de excitação) do sistema naquele instante de tempo, em cada grau de liberdade.

Neste caso, se observarmos atentamente a forma de estados genérica do sistema, é fácil perceber que devemos passar como argumento de entrada a matriz de inércia já invertida. Assim, evitamos de calcular uma inversão de matriz a cada passo de integração, o que torna a integração muito mais veloz.

Exemplo

Função de integração:

```
function dy = sistema2(t,y,invM,C,K)

% Esta função representa um sistema de primeira ordem, onde o parâmetro de
% saída (dy) é a derivada do vetor de estados y, e neste exemplo, os
% parâmetros de entrada são o tempo t, o vetor de estados y, a inversa da
% matriz de inércia invM, e as matrizes de amortecimento e rigidez, C e K.

% Matrizes auxiliares:
I = eye(size(invM)); % Identidade n x n
Z = zeros(size(invM)); % Matriz nula n x n
Zv = zeros(size(invM,1),1); % Vetor nulo n x 1

% Forma de estados:
A = [ Z I
      -invM*K -invM*C];

F = [Zv
      invM*excit(t)];

dy = A*y+F;
```

Função auxiliar de excitação:

```
function s = excit(t)

% Parâmetros da excitação
amp = 2;
w = 2*pi*5;

% Forças excitadoras no instante t
s = [ amp*sin(w*t)
      0
      0];
```

IMPORTANTE: Cada função destas deve ser salva em arquivos diferentes, porém ambos na mesma pasta de trabalho.

Invocando um método integrador

A sintaxe a ser introduzida a seguir é a mesma para todos os integradores da família ODE que serão exibidos a diante. Por isso, a princípio os exemplos a seguir serão apresentados utilizando o método ode45, mas o nome deste método poderá sempre ser substituído pelo nome de outro dos métodos apresentados posteriormente.

Caso 1:

Neste caso, a função de integração só possui como parâmetros de entrada a variável t e o vetor y . Além disso, não estamos interessados em analisar o sinal de saída, portanto ele não deve estar amostrado com taxa de amostragem constante.

Assim, a chamada do integrador deve ser feita da seguinte maneira:

$$[T,Y] = \text{ode45}(@\text{sistema}, [t0\ tf], y0)$$

Onde *sistema1* é a função de integração, $t0$ é o instante inicial de tempo, tf é o instante final de tempo, e o $y0$ é um vetor coluna contendo todas as condições iniciais do sistema, na mesma ordem assumida na modelagem de estados.

A variável de saída T será o vetor de tempo relativo aos resultados da integração, e a matriz Y será a matriz dos resultados de integração. Cada coluna representará um grau de liberdade, na mesma ordem assumida para o vetor de variáveis de estado y .

IMPORTANTE: O vetor $y0$ será o vetor de estados no primeiro instante de integração, e por isso, os dados de condição inicial devem estar na mesma ordem adotada para o vetor y .

Exemplo:

```

clear all
close all
clc

t0 = 0;
tf = 100;

y0 = [0 % Condição inicial de deslocamento
      0]; % Condição inicial de velocidade

[T,Y] = ode45(@sistema1,[t0 tf],y0); % Integro a função sistema1

figure('numbertitle','off','color','w','name','Resultados')
subplot(2,1,1)
plot(T,Y(:,1))
xlabel('Tempo [s]')
ylabel('Deslocamento [m]')
title('Deslocamento')
subplot(2,1,2)
plot(T,Y(:,2))
xlabel('Tempo [s]')
ylabel('Velocidade [m/s]')
title('Velocidade')

```

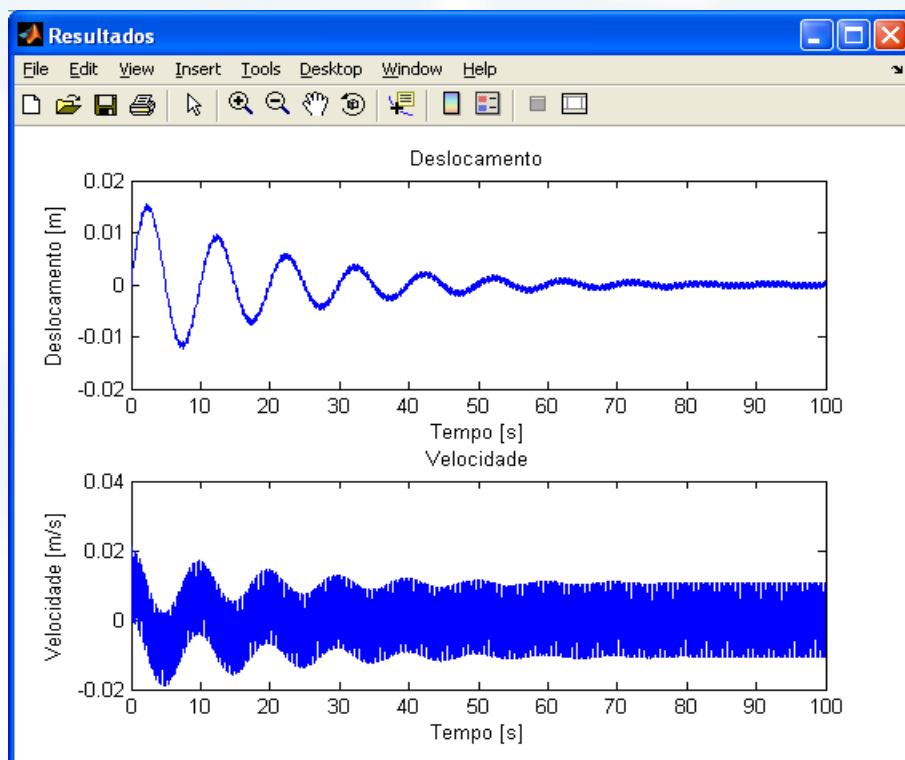


Figura 111: Resultado da integração da função sistema1

Nota: Fazendo a derivada do vetor de tempo final, podemos observar que a taxa de amostragem deste vetor é variável:

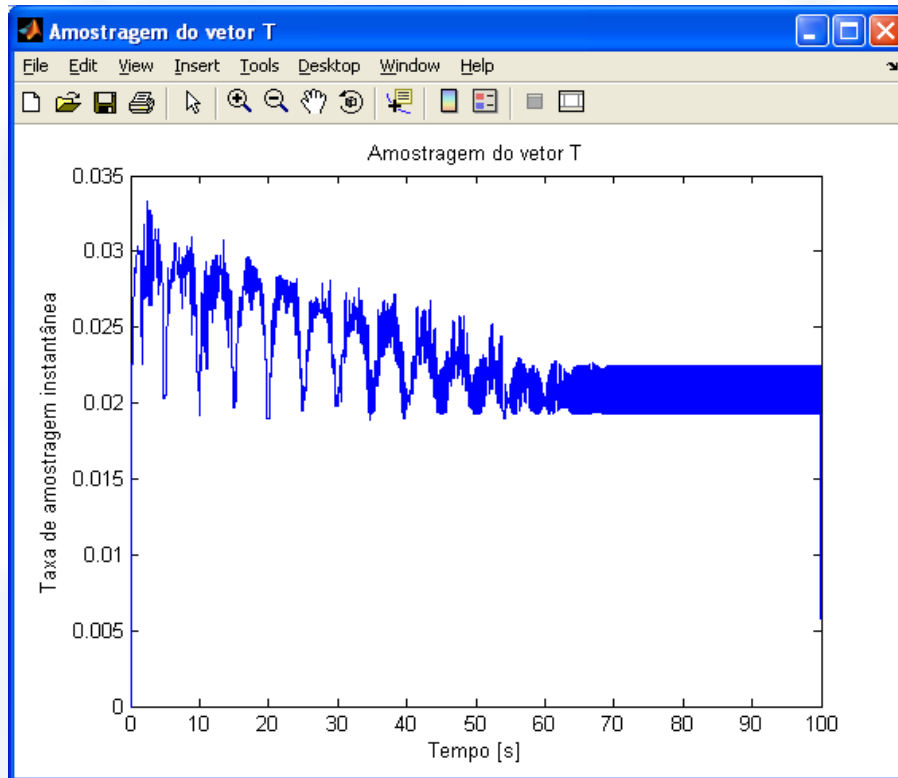


Figura 112: Taxa de amostragem variável do vetor de tempo

Caso 2:

Neste caso, a função de integração só possui como parâmetros de entrada a variável t e o vetor y . Porém, agora estamos interessados em analisar o sinal de saída, portanto ele deve estar amostrado com taxa de amostragem constante.

Assim, a chamada do integrador deve ser feita da seguinte maneira:

$$[T,Y] = \text{ode45}(@\text{sistema1}, ti, y0)$$

Onde ti agora representa o vetor de tempo desejado para os resultados da integração. Assim, vale notar que ao final da integração, o vetor T e o vetor ti serão iguais.

Lembre-se: Mesmo que tenha sido passado um vetor com taxa de amostragem contante para o integrador, se o método de integração for um método de passo variável, a integração continuará sendo feita com passo variável. Porém, após a integração, o Matlab realizará interpolações para fornecer a resposta final com a amostragem desejada.

Exemplo:

```

clear all
close all
clc

ti = 0:0.001:100;
y0 = [0 % Condição inicial de deslocamento
      0]; % Condição inicial de velocidade

[T,Y] = ode45(@sistema1,ti,y0); % Integro a função sistema1

figure('numbertitle','off','color','w','name','Resultados')
subplot(2,1,1)
plot(T,Y(:,1))
xlabel('Tempo [s]')
ylabel('Deslocamento [m]')
title('Deslocamento')
subplot(2,1,2)
plot(T,Y(:,2))
xlabel('Tempo [s]')
ylabel('Velocidade [m/s]')
title('Velocidade')

figure('numbertitle','off','color','w','name','Amostragem do vetor T')
plot(T,[0; diff(T)])
xlabel('Tempo [s]')
ylabel('Taxa de amostragem instantânea')
title('Amostragem do vetor T')

```

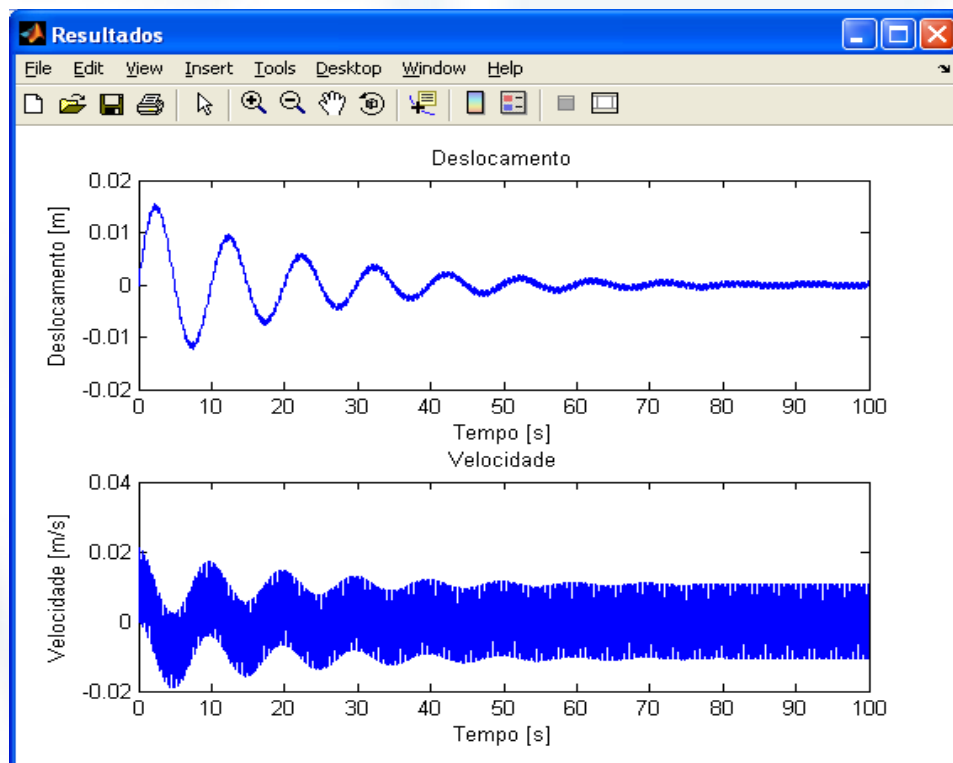


Figura 113: Resultado da integração da função sistema1

Nota: Fazendo a derivada do vetor de tempo final, podemos observar que a taxa de amostragem deste vetor é constante neste caso:

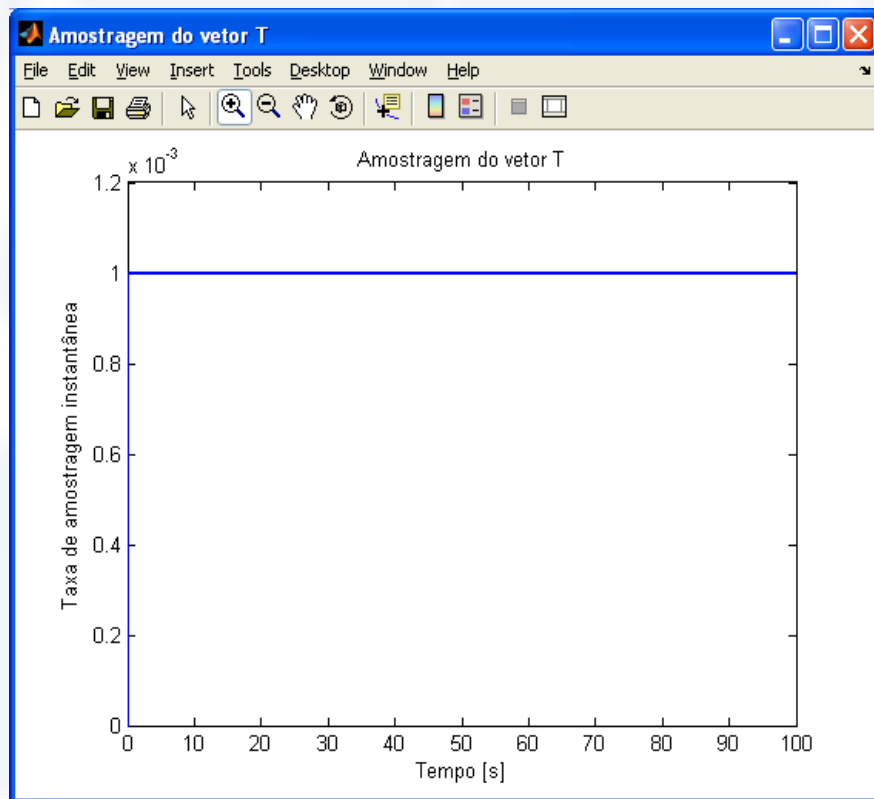


Figura 114: Taxa de amostragem constante do vetor de tempo.

Caso 3:

Neste caso, podemos ver um dos exemplos mais comuns do uso dos integradores ODE. A função de integração recebe mais parâmetros do que somente a variável t e o vetor de estados y . Além disso, os resultados terão taxa de amostragem constante, pois estamos interessados em, futuramente, processar estes dados.

Assim, a chamada do integrador deve se feita da seguinte maneira:

$$[T, Y] = \text{ode45}(@\text{sistema2}, ti, y0, [], p1, p2, \dots, pn)$$

Onde os parâmetros $p1$ a pn são parâmetros da função de integração do terceiro ao último na mesma ordem. Isto quer dizer que se a função de integração tiver a seguinte chamada:

$$\text{sistema2}(t, y, \text{invM}, C, K)$$

A chamada do integrador deverá ser feita desta maneira:

$$[T, Y] = \text{ode45}(@\text{sistema2}, ti, y0, [], \text{invM}, C, K)$$

Exemplo:

```

clear all
close all
clc

ti = 0:0.001:100; % Vetor de tempo com amostragem contante

y0 = [0 % Condições iniciais de deslocamento
      0
      0
      0 % Condições iniciais de velocidade
      0
      0];

% Matrizes do sistema
M = [2  0  0
     0  5  0
     0  0 20];

C = [ 1  -1  0
     -1  2  -1
      0  -1  3]* 0.5;

K = [ 1  -1  0
     -1  2  -1
      0  -1  3]*10;

[T,Y] = ode45(@sistema2,ti,y0,[],inv(M),C,K); % Integro a função sistema2

figure('numbertitle','off','color','w','name','Resultados GL1')
subplot(2,1,1)
plot(T,Y(:,1))
xlabel('Tempo [s]')
ylabel('Deslocamento [m]')
title('Deslocamento GL1')
subplot(2,1,2)
plot(T,Y(:,4))
xlabel('Tempo [s]')
ylabel('Velocidade [m/s]')
title('Velocidade GL1')

figure('numbertitle','off','color','w','name','Resultados GL2')
subplot(2,1,1)
plot(T,Y(:,2))
xlabel('Tempo [s]')
ylabel('Deslocamento [m]')
title('Deslocamento GL2')
subplot(2,1,2)
plot(T,Y(:,5))
xlabel('Tempo [s]')
ylabel('Velocidade [m/s]')
title('Velocidade GL2')

figure('numbertitle','off','color','w','name','Resultados GL3')
subplot(2,1,1)
plot(T,Y(:,3))
xlabel('Tempo [s]')
ylabel('Deslocamento [m]')

```

```
title('Deslocamento GL3')
subplot(2,1,2)
plot(T,Y(:,6))
xlabel('Tempo [s]')
ylabel('Velocidade [m/s]')
title('Velocidade GL3')
```

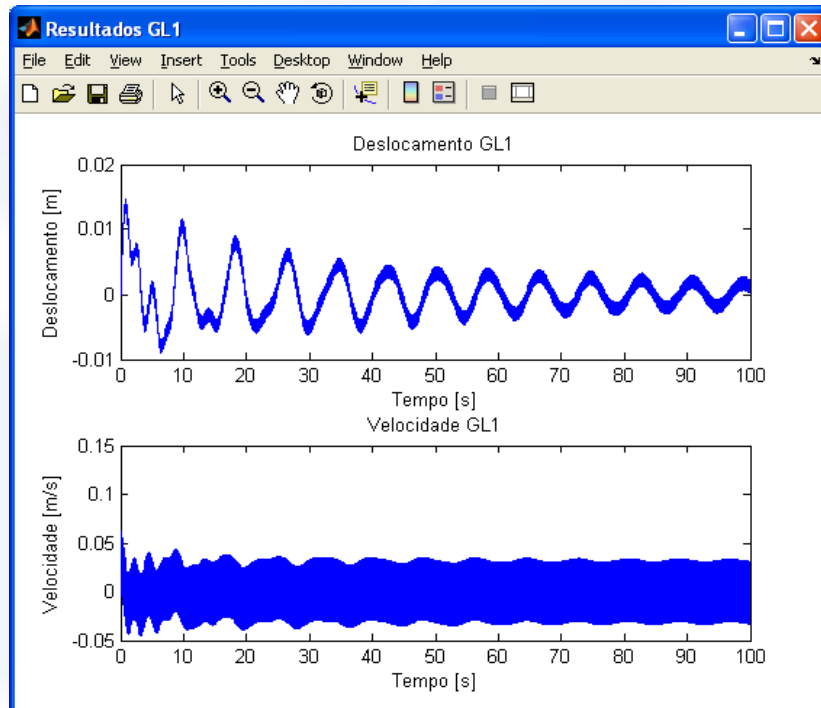


Figura 115: Resultado da integração do GL 1 da função sistema2

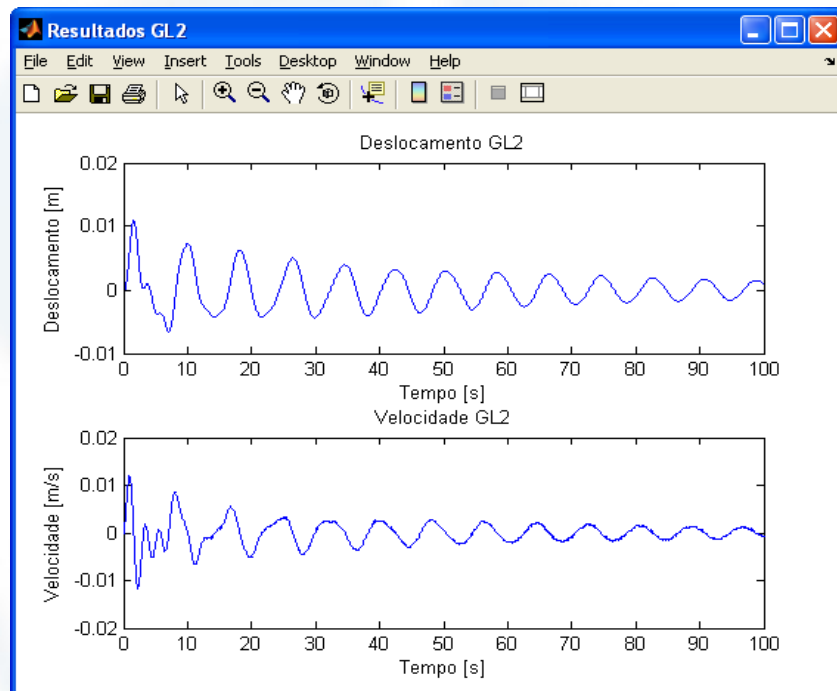


Figura 116: Resultado da integração do GL 2 da função sistema2.

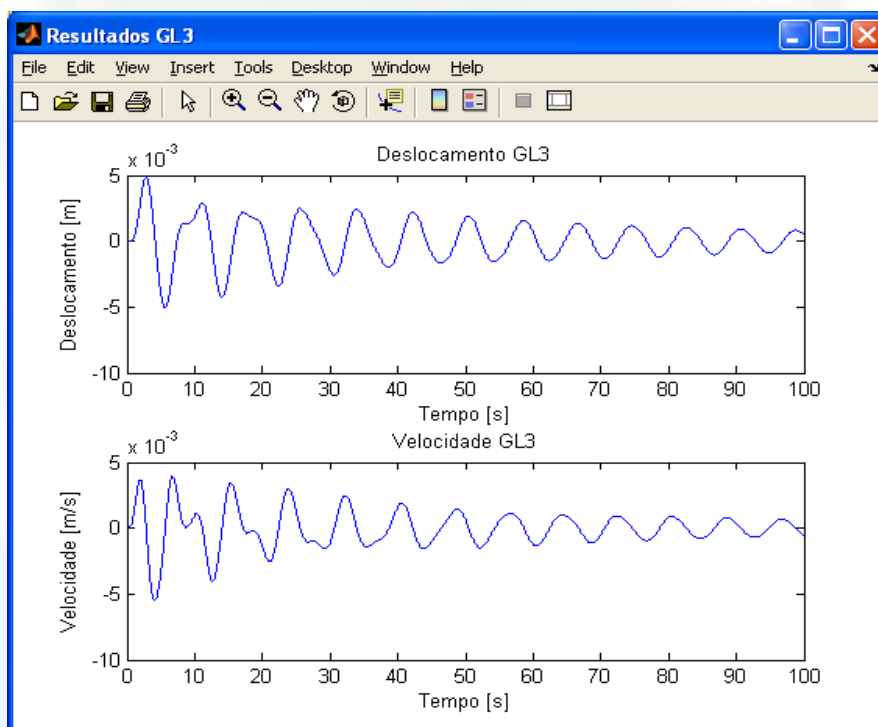


Figura 117: Resultado da integração do GL 3 da função sistema2.

Note que no quarto argumento da chamada do integrador há uma matriz vazia (`[]`). O quarto argumento é reservado para um objeto que identifica as opções relacionadas ao integrador. Passar como argumento esta matriz vazia, significa que as opções de integração serão as condições padrão do Matlab. Caso se deseje alterar estas condições, deve ser usada a função *odeset*.

Verifique no help do programa as utilidades da função *odeset*.

Como dito anteriormente, o Matlab dispõe de muitos outros integradores, diferentes do ode45. Cada um tem um propósito, e se encaixa melhor a cada tipo de problema a ser estudado. Veja na tabela abaixo, quais são os tipos de integradores disponíveis e qual é o seu uso.

Integrador	Tipo de problema	Precisão	Quando usar
ode45	Não rígido	Media	Na maioria das vezes. Deve ser o primeiro integrador a se tentar.
ode23	Não rígido	Baixa	Problemas com tolerância a erros maiores, ou problemas moderadamente rígidos.
ode113	Não rígido	Baixa a alta	Problemas com tolerância restrita a erros, ou para solução de problemas computacionalmente intensos.
ode15s	Rígido	Baixa a média	Alternativa à ode45 se o integrador está lento, pois o problema é rígido.
ode23s	Rígido	Baixa	Para problemas rígidos com maior tolerância a erros, e com matriz de massa constante.
ode23t	Moderadamente rígido	Baixa	Para problemas moderadamente rígidos, quando se deseja uma solução sem amortecimento numérico.
ode23tb	Rígido	Baixa	Para problemas rígidos com maior tolerância a erros.

– Waitbar

Alguns processos de integração podem ser muito demorados. Quando usamos integradores do tipo ODE, o Matlab não dá sinais indicando o progresso da integração. Assim, podemos ficar em dúvida se a integração está em andamento ou se ocorreu algum erro.

Para indicar o andamento do processo de integração, podemos utilizar a ferramenta *Waitbar*, que exibirá uma barra indicando o progresso dos cálculos.

Para criar uma waitbar a sintaxe é:

```
h = waitbar( init, 'message')
```

Onde *init* é o estado inicial da waitbar (0 é vazia e 1 é cheia), e *message* é a mensagem que será exibida junto à barra. A variável *h* é chamada de handle da waitbar, e será utilizada para atualizarmos o valor da barra de progresso.

Para a atualização da waitbar, a sintaxe é:

```
waitbar(p,h)
```

Onde *p* é o novo valor da barra de progresso, e *h* é o handle da waitbar.

Exemplo:

Neste exemplo, iremos integrar a função *sistema2* e mostrar seu progresso na waitbar. Porém, a atualização da waitbar será feita dentro desta função. Para isso, ela deve permitir mais 2 argumentos de entrada: o handle da waitbar, *h*, e o tempo final de integração, para que possamos atualizar a waitbar.

Assim, a função *sistema2* ficará da seguinte forma:

```
function dy = sistema2(t,y,invM,C,K,Tf,h)

% Esta função representa um sistema de primeira ordem, onde o parâmetro de
% saída (dy) é a derivada do vetor de estados y, e neste exemplo, os
% parâmetros de entrada são o tempo t, o vetor de estados y, a inversa da
% matriz de inércia invM, e as matrizes de amortecimento e rigidez, C e K.

% Matrizes auxiliares:
I = eye(size(invM)); % Identidade n x n
Z = zeros(size(invM)); % Matriz nula n x n
Zv = zeros(size(invM,1),1); % Vetor nulo n x 1

% Forma de estados:
A = [      Z      I
     -invM*K -invM*C];

F = [Zv
     invM*excit(t)];

dy = A*y+F;

waitbar(t/Tf,h); % Atualizo o valor da waitbar
```

E o script para a integração ficará da seguinte forma:

```
clear all
close all
clc

Tf = 100; % Tempo final de integração
ti = 0:0.001:Tf; % Vetor de tempo com amostragem contante

y0 = [0 % Condições iniciais de deslocamento
      0
      0
      0 % Condições iniciais de velocidade
      0
      0];

% Matrizes do sistema
M = [2  0  0
     0  5  0
     0  0 20];

C = [ 1  -1  0
     -1  2  -1
      0  -1  3]* 0.5;

K = [ 1  -1  0
     -1  2  -1
      0  -1  3]*10;

h = waitbar(0,'Aguarde...'); % Crio a waitbar
```

```
[T,Y] = ode45(@sistema2,ti,y0,[],inv(M),C,K,Tf,h); % Integro a função sistema2

close(h); % Fecho a waitbar

figure('numbertitle','off','color','w','name','Resultados GL1')
subplot(2,1,1)
plot(T,Y(:,1))
xlabel('Tempo [s]')
ylabel('Deslocamento [m]')
title('Deslocamento GL1')
subplot(2,1,2)
plot(T,Y(:,4))
xlabel('Tempo [s]')
ylabel('Velocidade [m/s]')
title('Velocidade GL1')

figure('numbertitle','off','color','w','name','Resultados GL2')
subplot(2,1,1)
plot(T,Y(:,2))
xlabel('Tempo [s]')
ylabel('Deslocamento [m]')
title('Deslocamento GL2')
subplot(2,1,2)
plot(T,Y(:,5))
xlabel('Tempo [s]')
ylabel('Velocidade [m/s]')
title('Velocidade GL2')

figure('numbertitle','off','color','w','name','Resultados GL3')
subplot(2,1,1)
plot(T,Y(:,3))
xlabel('Tempo [s]')
ylabel('Deslocamento [m]')
title('Deslocamento GL3')
subplot(2,1,2)
plot(T,Y(:,6))
xlabel('Tempo [s]')
ylabel('Velocidade [m/s]')
title('Velocidade GL3')
```

Durante a Integração, esta será a *waitbar* exibida:

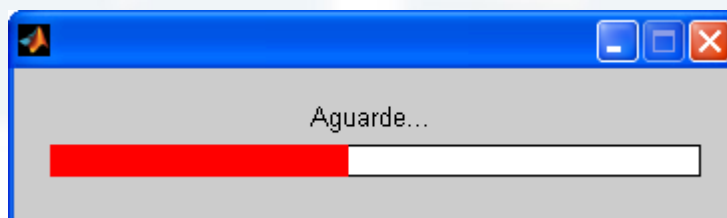


Figura 118: Waitbar

E ao final, serão obtidos os seguintes resultados:

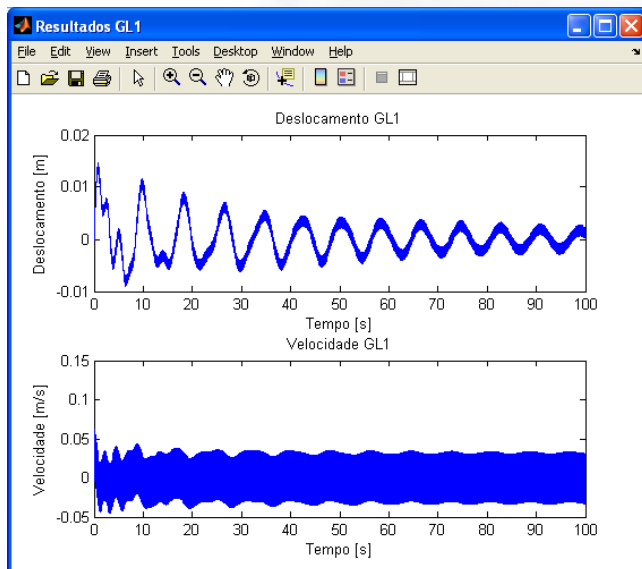


Figura 119: Resultado para o GL 1.

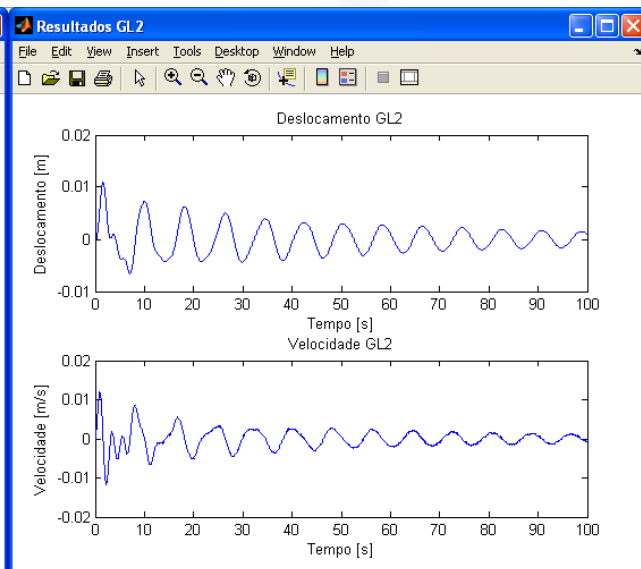


Figura 120: Resultado para o GL 2.

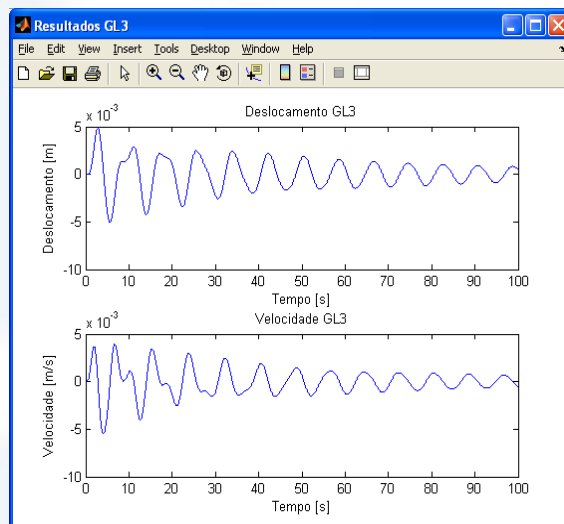


Figura 121: Resultado para o GL 3.

Seção 3.3 – Forma de estados e Transfer Function

Quando trabalhamos com sistemas LTI (Lineares e invariantes no tempo), é muito comum utilizarmos a transformada de Laplace e funções de transferência.

No Matlab, isso pode ser feito através da função *tf*. Na verdade, esta função define qual será a variável de Laplace (normalmente *s*) para a definição das funções de transferência.

Com a definição desta variável, será possível escrever funções, matrizes de transferência, e sistemas na forma de estados padrão do Matlab.

Além disso, neste ambiente há muitas funções que aceitam como entrada funções de transferência, e traçam diagramas de Bode, Nyquist, Margens de estabilização, mapa de polos e zeros, e outras funcionalidades para análise de plantas em geral.

Definindo uma planta:

Da teoria, temos que para um sistema de 1 GL da forma:

$$m \ddot{x}(t) + c \dot{x}(t) + k x(t) = f(t)$$

Aplicando a transformada de Laplace em ambos os membros desta equação, considerando todas as condições iniciais homogêneas, temos:

$$m X(s)s^2 + c X(s)s + K X(s) = F(s)$$

Neste caso, $X(s)$ é chamado de saída do sistema, e $F(s)$ é chamado de entrada. Pela definição, dividindo a saída do sistema por sua entrada, temos a função de transferência do sistema, chamada de $H(s)$.

$$H(s) = \frac{X(s)}{F(s)} = \frac{1}{m s^2 + c s + k}$$

Também por definição, o número de zeros de uma função de transferência deve ser menor ou igual ao número de polos da mesma, caso contrário, a causalidade do sistema não será respeitada.

No Matlab, para criarmos funções de transferência, ou *transfer functions*, devemos primeiramente definir a variável de Laplace (*s*). Para isso devemos usar a função *tf*. Com o uso desta função, podemos atribuir a qualquer variável, o valor *s* que representa o operador de Laplace. Depois disso, qualquer operação executada com esta variável resultará em uma função de transferência em termos de *s*.

Exemplo:

```
clear all
close all
clc

a = tf('s');
H = 1/(a+1);

x = tf('s');
```

```
T = 1/(x+1);
s = tf('s');
Y = 1/(s+1);

disp('***** Valor de H *****')
H
disp('***** Valor de T *****')
T
disp('***** Valor de Y *****')
Y
```

Command Window:

```
***** Valor de H *****
```

```
Transfer function:
```

```
1
-----
s + 1
```

```
***** Valor de T *****
```

```
Transfer function:
```

```
1
-----
s + 1
```

```
***** Valor de Y *****
```

```
Transfer function:
```

```
1
-----
s + 1
```

```
>>
```

E para a análise destas funções de transferência, que representam sistemas, o Matlab possui algumas funções para a exibição de diagramas que auxiliam na compreensão de seu comportamento em frequência. Os diagramas mais frequentes são:

- *Diagrama de Bode*
- *Diagrama de Nyquist*
- *Diagrama de Nichols*
- *Diagrama de Valores Singulares*
- *Diagrama do Lugar das Raízes*
- *Mapa de Polos e Zeros*
- *Diagrama de margens de estabilidade*

Alguns destes diagramas, por poderem tratar também de sistemas de múltiplas entradas e múltiplas saídas, necessitam que o sistema esteja representado não por meio de funções ou matrizes de transferência, mas por meio da forma de estados padrão do Matlab. Portanto, antes de mostrar os diagramas, iremos apresentar uma maneira de se criar um sistema nesta representação.

Forma de estados padrão

Primeiro, vamos supor a existência de um sistema linear, invariante no tempo, causal, observável, e com N graus de liberdade. Sua equação de movimento pode ser escrita na seguinte forma:

$$[M]\{\ddot{z}\} + [C]\{\dot{z}\} + [K]\{z\} = \{F(t)\}$$

Como já visto no capítulo anterior, definindo o vetor de estados x , a forma de estados padrão do sistema é:

$$\begin{Bmatrix} \dot{z} \\ \ddot{z} \end{Bmatrix} = \begin{bmatrix} [0]_{n \times n} & [I]_{n \times n} \\ -[M]^{-1}[K] & -[M]^{-1}[C] \end{bmatrix} \begin{Bmatrix} z \\ \dot{z} \end{Bmatrix} + \begin{Bmatrix} [0]_{n \times 1} \\ [M]^{-1}\{F(t)\}_{n \times 1} \end{Bmatrix}$$

Também representada como:

$$\{\dot{x}\} = [A]\{x\} + \bar{F}(t) = H(t, x)$$

Onde:

$$\{x\} = \begin{Bmatrix} z \\ \dot{z} \end{Bmatrix}_{2n \times 1}$$

$$\{\dot{x}\} = \begin{Bmatrix} \dot{z} \\ \ddot{z} \end{Bmatrix}_{2n \times 1}$$

$$[A] = \begin{bmatrix} [0]_{n \times n} & [I]_{n \times n} \\ -[M]^{-1}[K] & -[M]^{-1}[C] \end{bmatrix}_{2n \times 2n}$$

$$\bar{F}(t) = \begin{Bmatrix} [0]_{n \times 1} \\ [M]^{-1}\{F(t)\}_{n \times 1} \end{Bmatrix}_{2n \times 1}$$

O vetor $\bar{F}(t)$ também pode ser representado da seguinte maneira:

$$\bar{F}(t) = \begin{Bmatrix} [0]_{n \times 1} \\ [M]^{-1}\{F(t)\}_{n \times 1} \end{Bmatrix}_{2n \times 1} = \begin{bmatrix} [0]_{n \times n} \\ [M]^{-1}_{n \times n} \end{bmatrix}_{2n \times n} \begin{Bmatrix} F(t) \end{Bmatrix}_{n \times 1} = [B]_{2n \times n} \{F(t)\}_{n \times 1}$$

Assim, o sistema pode ser representado da seguinte maneira:

$$\{\dot{x}\} = [A]_{2n \times 2n} \{x\} + [B]_{2n \times n} \{F(t)\}$$

E a saída y (resultado de interesse), pode ser calculada em função dos estados e das forças externas, como uma combinação linear destes. Portanto, por meio de duas matrizes (C e D), podemos calcular as saídas do sistema. Considerando que o sistema tenha m saídas, y pode ser representado da

seguinte maneira:

$$y_{m \times 1} = [C]_{m \times n} \{x\} + [D]_{m \times n} \{F(t)\}$$

Finalmente, temos o sistema representado por meio das matrizes A, B, C e D:

$$\begin{cases} \dot{x} = [A]x + [B]F(t) \\ y = [C]x + [D]F(t) \end{cases}$$

Caso se tenha estas 4 matrizes, para criar um objeto no espaço de estados que representa este sistema, basta utilizar o comando `ss`.

Exemplo:

```
clear all
close all
clc

% Matrizes M, C e K do sistema

M = [1 0 0
      0 2 0
      0 0 3];

Cd = [ 0.1 -0.1 0
       -0.1 0.2 -0.1
        0 -0.1 0.2];

K = [ 10 -10 0
      -10 20 -10
        0 -10 20];

Z = zeros(size(M)); %Matriz de zeros n x n
I = eye(size(M)); %Matriz de identidade n x n

A = [ Z I
      -M\K -M\Cd]; % Matriz A da forma de estados padrão

B = [Z
      M\I]; % Matriz B da forma de estados padrão

C = eye(size(A)); % Matriz C da forma de estados padrão.

D = zeros(size(A,1),size(B,2)); % Matriz D da forma de estados padrão.

% Neste caso, as saídas do sistema são os próprios estados.

sisFE = ss(A,B,C,D)
```

Command Window:

```
a =
      x1      x2      x3      x4      x5      x6
```

```

x1      0      0      0      1      0      0
x2      0      0      0      0      1      0
x3      0      0      0      0      0      1
x4     -10     10      0     -0.1     0.1     0
x5       5     -10      5     0.05    -0.1     0.05
x6       0      3.333   -6.667      0    0.03333   -0.06667

b =
      u1      u2      u3
x1       0       0       0
x2       0       0       0
x3       0       0       0
x4       1       0       0
x5       0      0.5       0
x6       0       0    0.3333

c =
      x1      x2      x3      x4      x5      x6
y1      1      0      0      0      0      0
y2      0      1      0      0      0      0
y3      0      0      1      0      0      0
y4      0      0      0      1      0      0
y5      0      0      0      0      1      0
y6      0      0      0      0      0      1

d =
      u1      u2      u3
y1      0      0      0
y2      0      0      0
y3      0      0      0
y4      0      0      0
y5      0      0      0
y6      0      0      0

Continuous-time model.
>>

```

Nota: As matrizes A e B são definidas com base nas matrizes do sistema, enquanto as matrizes C e D são definidas arbitrariamente

Agora já é possível demonstrar todos os diagramas.

– Diagrama de Bode - *bode*

O diagrama de bode exibe a função de resposta em frequência do sistema, tanto em amplitude (dB) quanto em fase (graus). Esta função pode ser aplicada a funções de transferência simples, ou a sistemas na forma de estados.

Exemplos:

Aplicação da função para uma função de transferência simples:

```
clear all
close all
clc

s = tf('s'); % Definição do operador de laplace

H = 1/(s^2+0.2*s+15); % Criação da função de transferência

bode(H) % Diagrama de bode
```

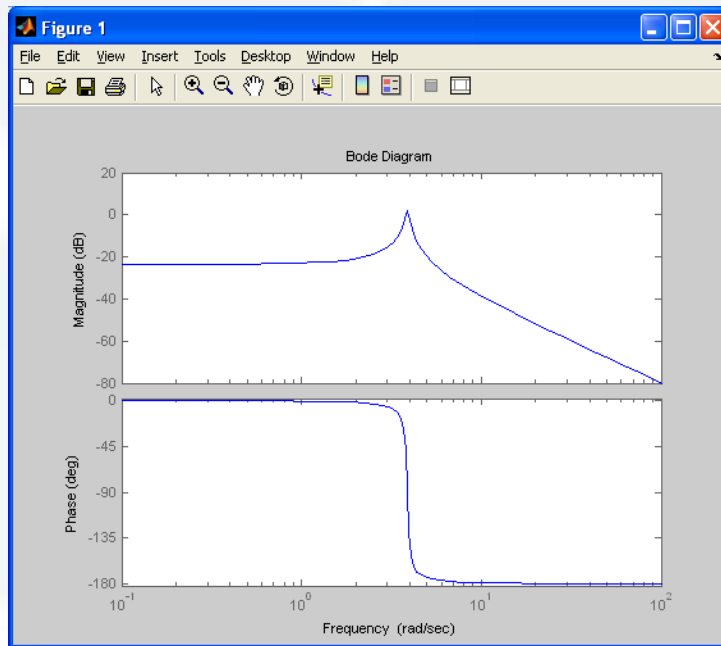


Figura 122: Diagrama de bode de uma função de transferência.

Aplicação da função para um sistema na forma de estados:

```
clear all
close all
clc

% Matrizes M, C e K do sistema

M = [1 0
     0 2];

Cd = [ 0.1 -0.1
      -0.1  0.2 ];

K = [ 10 -10
     -10  20 ];

Z = zeros(size(M)); %Matriz de zeros n x n
I = eye(size(M)); %Matriz de identidade n x n

A = [ Z      I
     -M\K -M\Cd]; % Matriz A da forma de estados padrão
```

```

B = [Z
     M\I]; % Matriz B da forma de estados padrão

C = eye(size(A)); % Matriz C da forma de estados padrão.

D = zeros(size(A,1),size(B,2)); % Matriz D da forma de estados padrão.

% Neste caso, as saídas do sistema são os próprios estados.

sisFE = ss(A,B,C,D);

bode(sisFE)

```

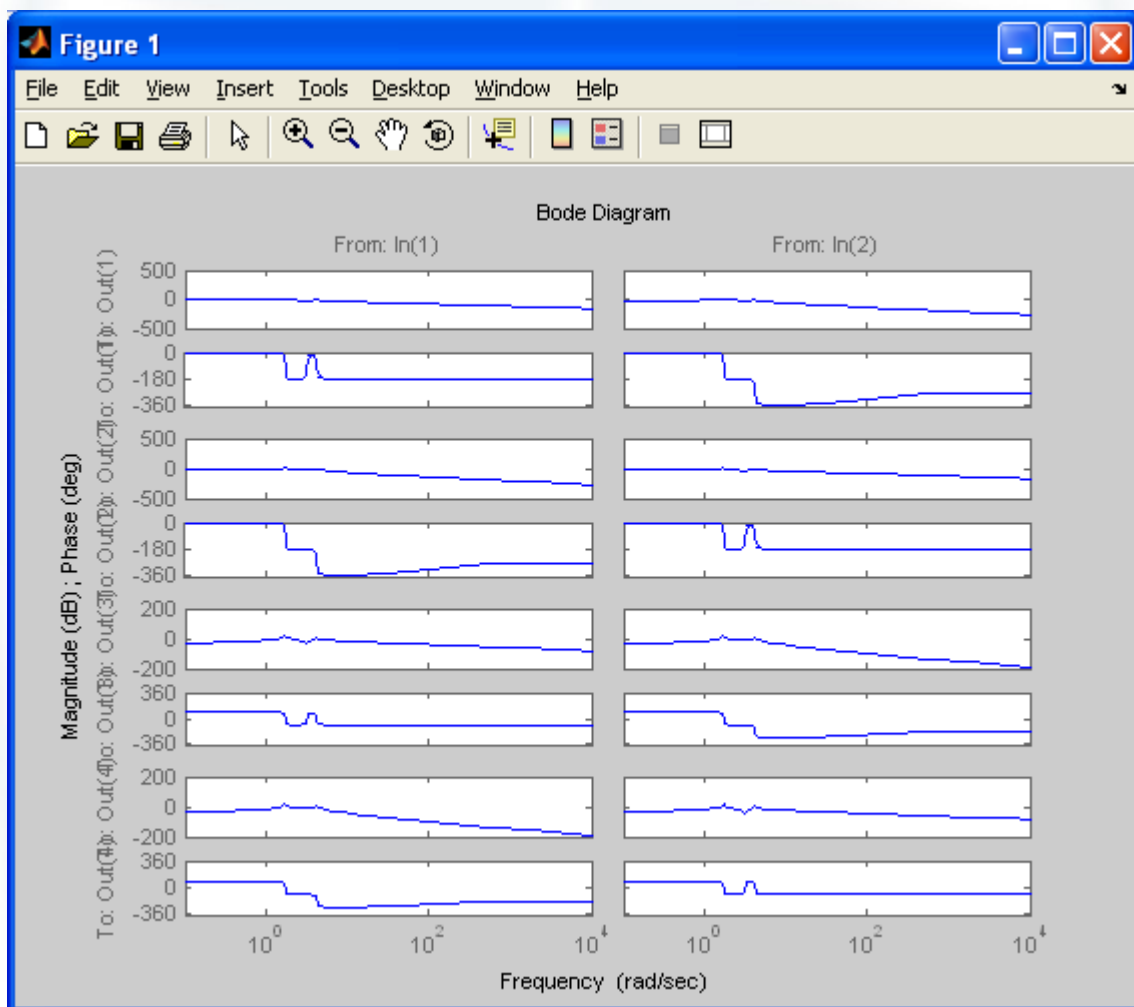


Figura 123: Diagrama de bode de um sistema MIMO.

– Diagrama de Nyquist - *nyquist*

O diagrama de Nyquist é uma representação gráfica onde a função de resposta em frequência do sistema é plotada em um sistema Argand-Gauss (eixo real – imaginário). Esta função pode ser aplicada a funções de transferência simples, ou a sistemas na forma de estados.

Exemplos:

Aplicação a uma função de transferência simples:

```
clear all
close all
clc

s = tf('s'); % Definição do operador de laplace

H = 1/(s^2+0.2*s+15); % Criação da função de transferência

nyquist(H) % Diagrama de Nyquist
```

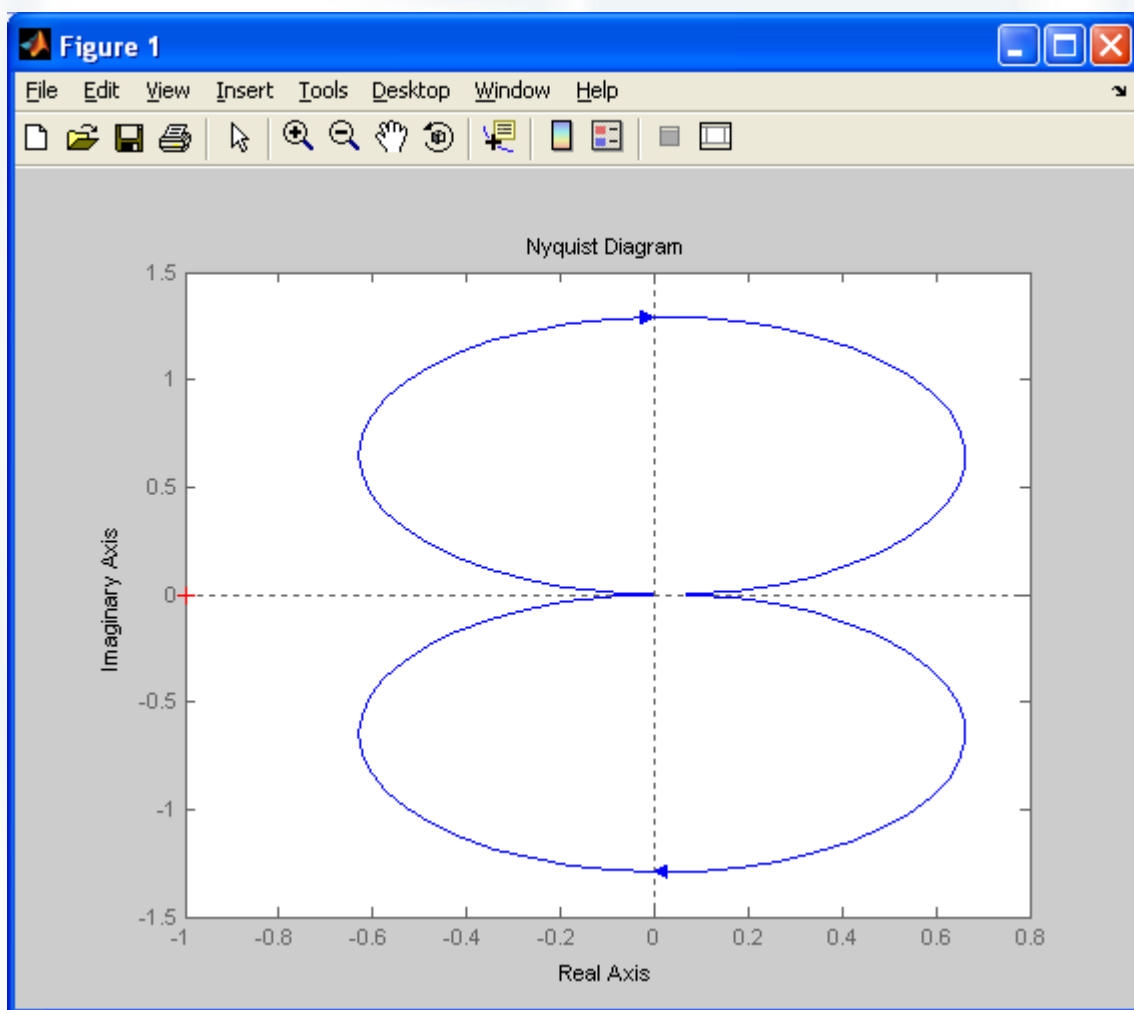


Figura 124: Diagrama de Nyquist de uma função de transferência

Aplicação a um sistema na forma de estados:

```
clear all
close all
clc

% Matrizes M, C e K do sistema

M = [1 0
     0 2];

Cd = [ 0.1 -0.1
      -0.1  0.2 ];

K = [ 10 -10
     -10  20 ];

Z = zeros(size(M)); %Matriz de zeros n x n
I = eye(size(M)); %Matriz de identidade n x n

A = [ Z      I
     -M\K -M\Cd]; % Matriz A da forma de estados padrão

B = [Z
     M\I]; % Matriz B da forma de estados padrão

C = eye(size(A)); % Matriz C da forma de estados padrão.

D = zeros(size(A,1),size(B,2)); % Matriz D da forma de estados padrão.

% Neste caso, as saídas do sistema são os próprios estados.

sisFE = ss(A,B,C,D);

nyquist(sisFE)
```

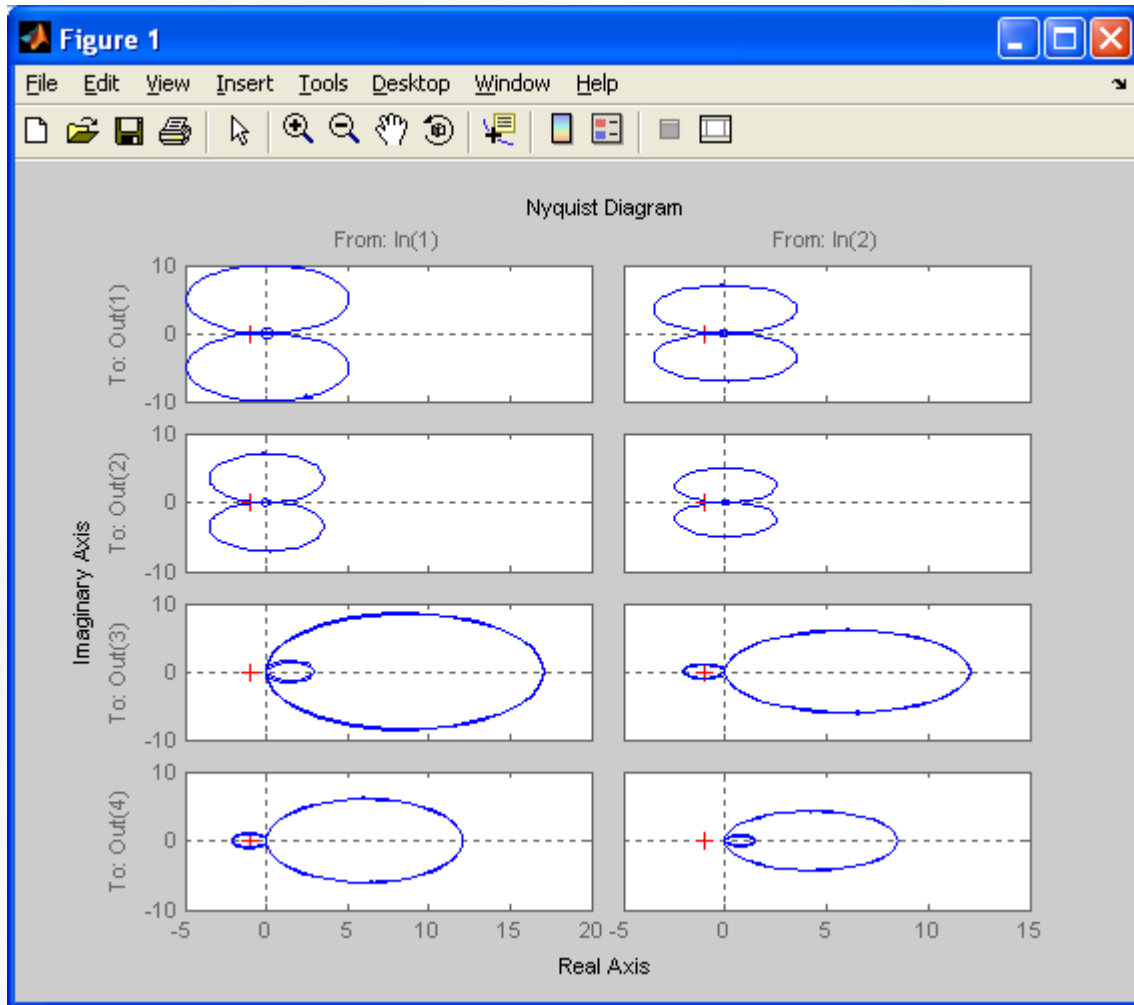


Figura 125: Diagramas de Nyquist para um sistema MIMO.

– Diagrama de Nichols - *nichols*

No diagrama de Nichols, a FRF do sistema é plotada em um sistema onde a ordenada é o ganho da planta em malha aberta, em dB, e a abscissa é a fase, em graus. Esta função pode ser aplicada a funções de transferência simples, ou a sistemas na forma de estados.

Exemplos:

Aplicação a uma função de transferência simples:

```
clear all
close all
clc

s = tf('s'); % Definição do operador de laplace

H = 1/(s^2+0.2*s+15); % Criação da função de transferência

nichols(H) % Diagrama de Nichols
```

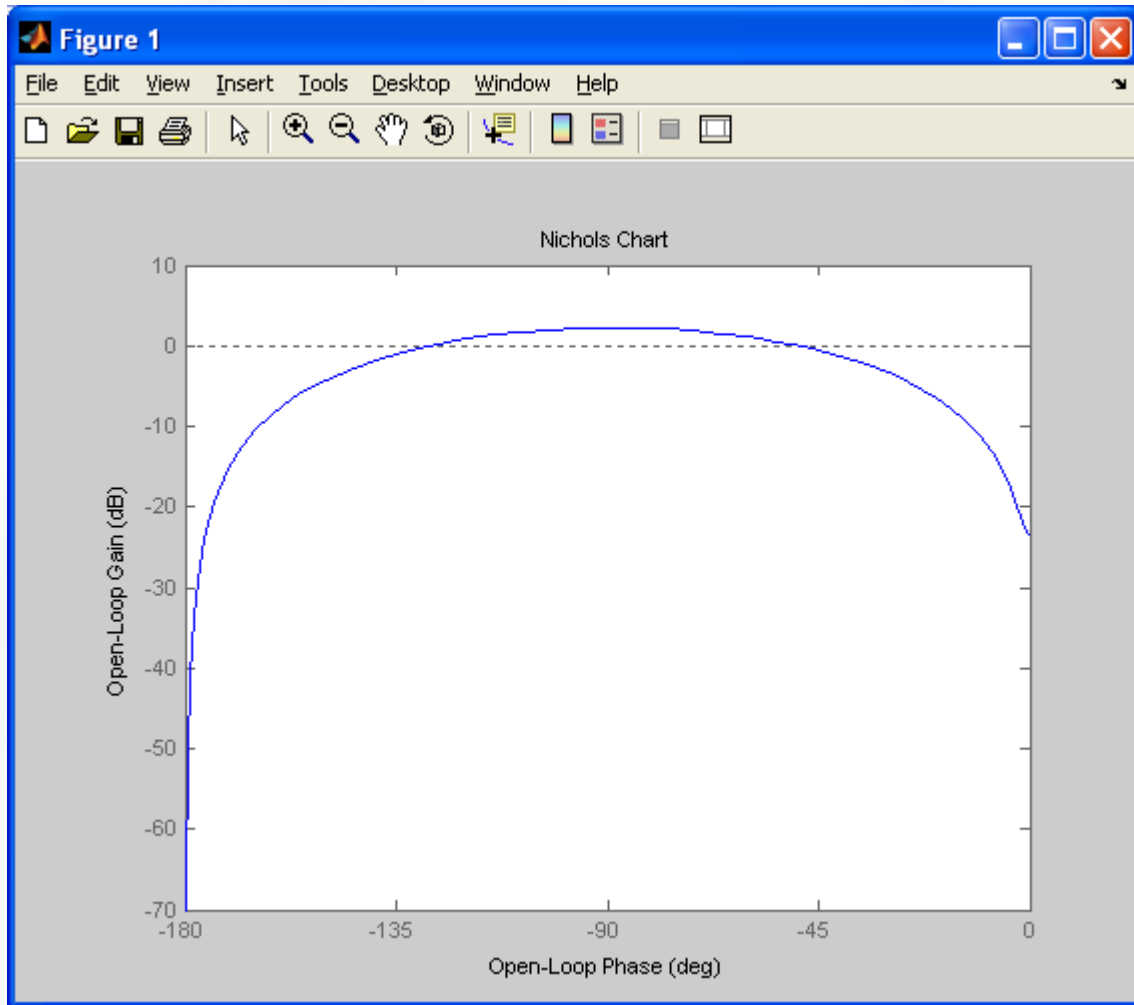


Figura 126: Diagrama de Nichols para uma função de transferência

Aplicação a um sistema na forma de estados:

```
clear all
close all
clc

% Matrizes M, C e K do sistema

M = [ 1  0
      0  2];

Cd = [ 0.1 -0.1
       -0.1  0.2 ];

K = [ 10 -10
      -10  20 ];

Z = zeros(size(M)); %Matriz de zeros n x n
I = eye(size(M)); %Matriz de identidade n x n

A = [ Z      I
      -M\K  -M\Cd]; % Matriz A da forma de estados padrão
```

```

B = [Z
      M\I]; % Matriz B da forma de estados padrão

C = eye(size(A)); % Matriz C da forma de estados padrão.

D = zeros(size(A,1),size(B,2)); % Matriz D da forma de estados padrão.

% Neste caso, as saídas do sistema são os próprios estados.

sisFE = ss(A,B,C,D);

nichols(sisFE)

```

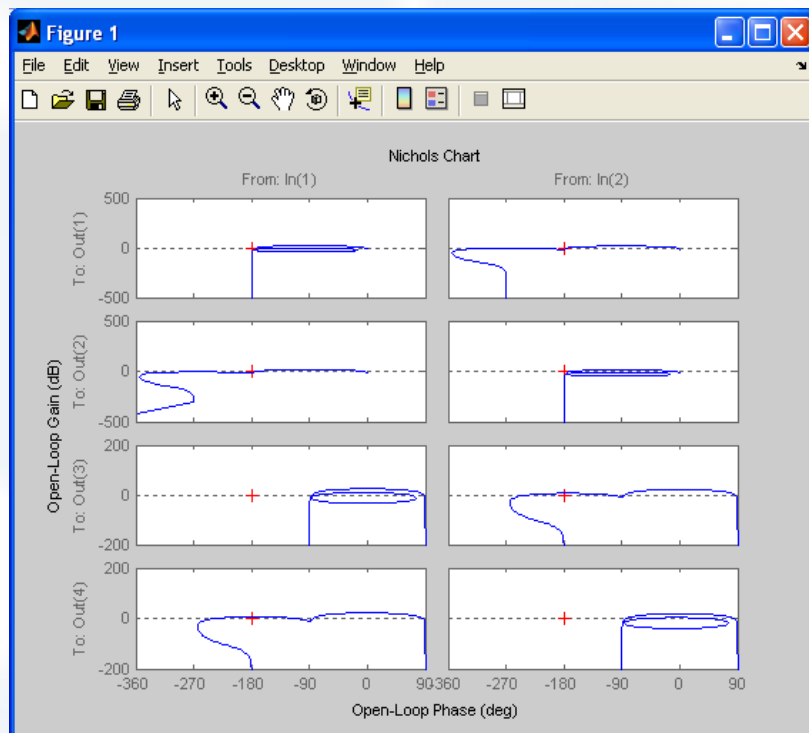


Figura 127: Diagramas de Nichols de um sistema MIMO.

– Diagrama de Valores Singulares - *sigma*

São exibidas neste diagrama as curvas de valores singulares do sistema. Esta função pode ser aplicada a funções de transferência simples, ou a sistemas na forma de estados.

Exemplos:

Aplicação a uma função de transferência simples:

```

clear all
close all
clc

s = tf('s'); % Definição do operador de laplace

```

```
H = 1/(s^2+0.2*s+15); % Criação da função de transferência
sigma(H)
```

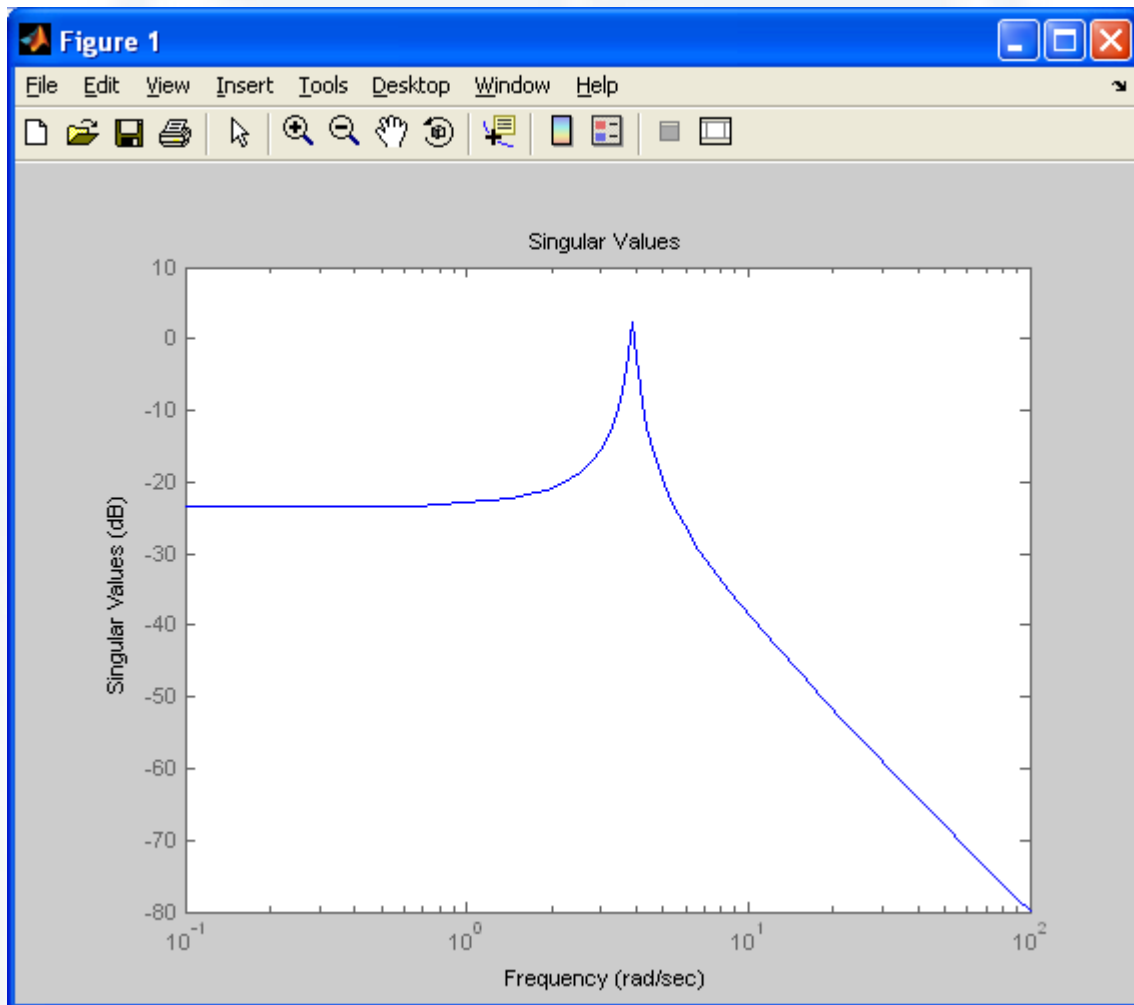


Figura 128: Diagrama de valores singulares para uma função de transferência

Aplicação a um sistema na forma de estados:

```
clear all
close all
clc

% Matrizes M, C e K do sistema

M = [1 0
     0 2];

Cd = [ 0.1 -0.1
      -0.1  0.2 ];

K = [ 10 -10
     -10  20 ];
```

```

Z = zeros(size(M)); %Matriz de zeros n x n
I = eye(size(M)); %Matriz de identidade n x n

A = [ Z      I
      -M\K  -M\Cd]; % Matriz A da forma de estados padrão

B = [ Z
      M\I]; % Matriz B da forma de estados padrão

C = eye(size(A)); % Matriz C da forma de estados padrão.

D = zeros(size(A,1),size(B,2)); % Matriz D da forma de estados padrão.

% Neste caso, as saídas do sistema são os próprios estados.

sisFE = ss(A,B,C,D);

sigma(sisFE)

```

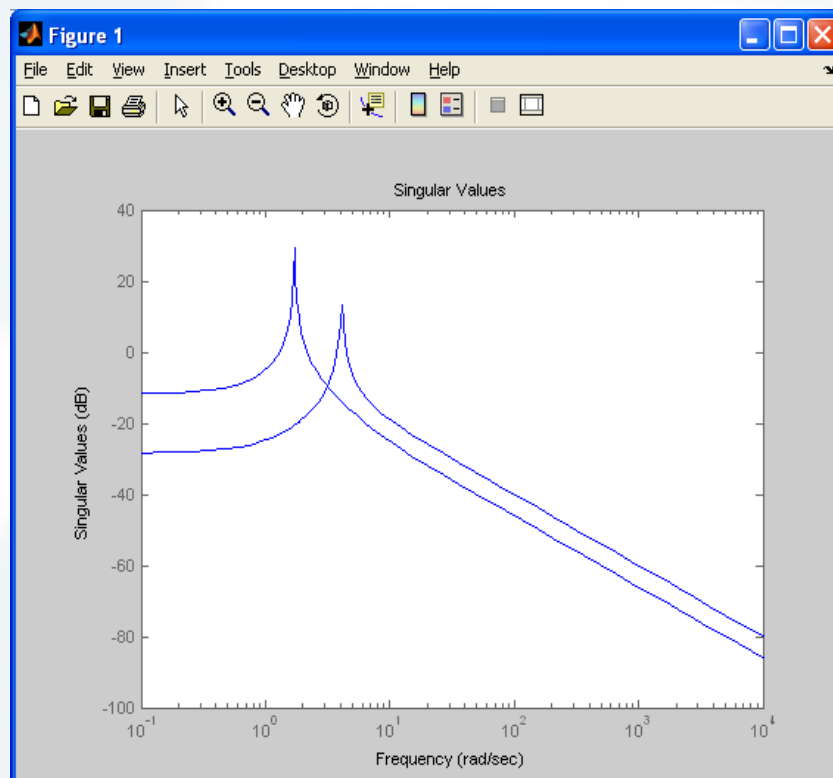


Figura 129: Diagramas de valores singulares para um sistema MIMO.

– Diagrama do Lugar das Raízes – *rlocus*

Exibe o diagrama de lugar das raízes do sistema. Esta função pode ser aplicada apenas a funções de transferência simples, ou a sistemas na forma de estados, desde que ele seja um sistema SISO (Single Input Single Output).

Exemplo:

Aplicação a uma função de transferência simples:

```
clear all
close all
clc

s = tf('s'); % Definição do operador de laplace
H = 1/(s^2+0.2*s+15); % Criação da função de transferência
rlocus(H)
```

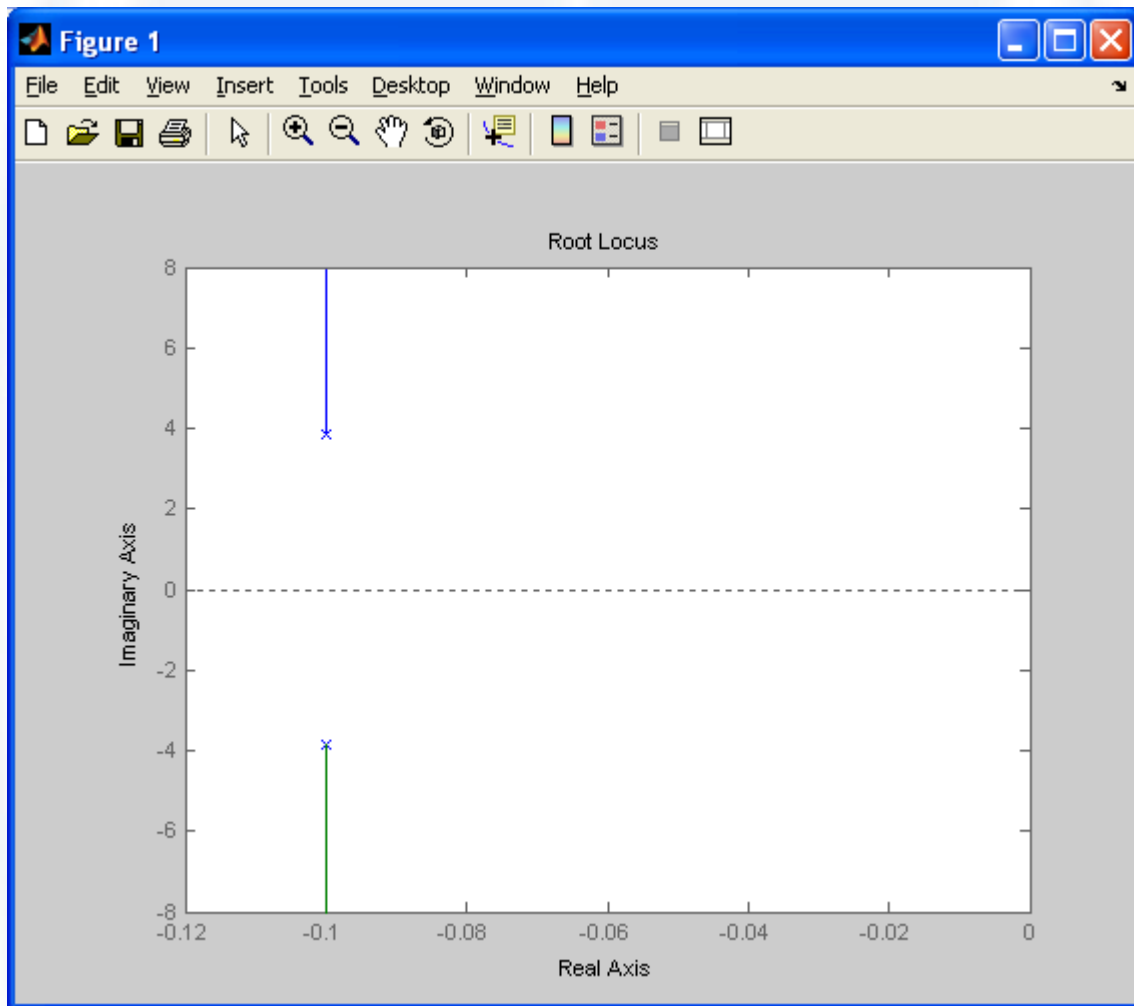


Figura 130: Diagrama de Lugares das raízes para uma função de transferência

– Mapa de Polos e Zeros - *pzmap*

Mostra em um plano Argand-Gauss (Real-Imaginário) os polos e os zeros do sistema. Os polos são representados por “x” e os zeros por “o”. Esta função pode ser aplicada a funções de transferência simples, ou a sistemas na forma de estados.

Exemplos:

Aplicação a uma função de transferência simples:


```
clear all
close all
clc

s = tf('s'); % Definição do operador de laplace

H = (s-2)/(s^2+0.2*s+15); % Criação da função de transferência

pzmap(H)
```

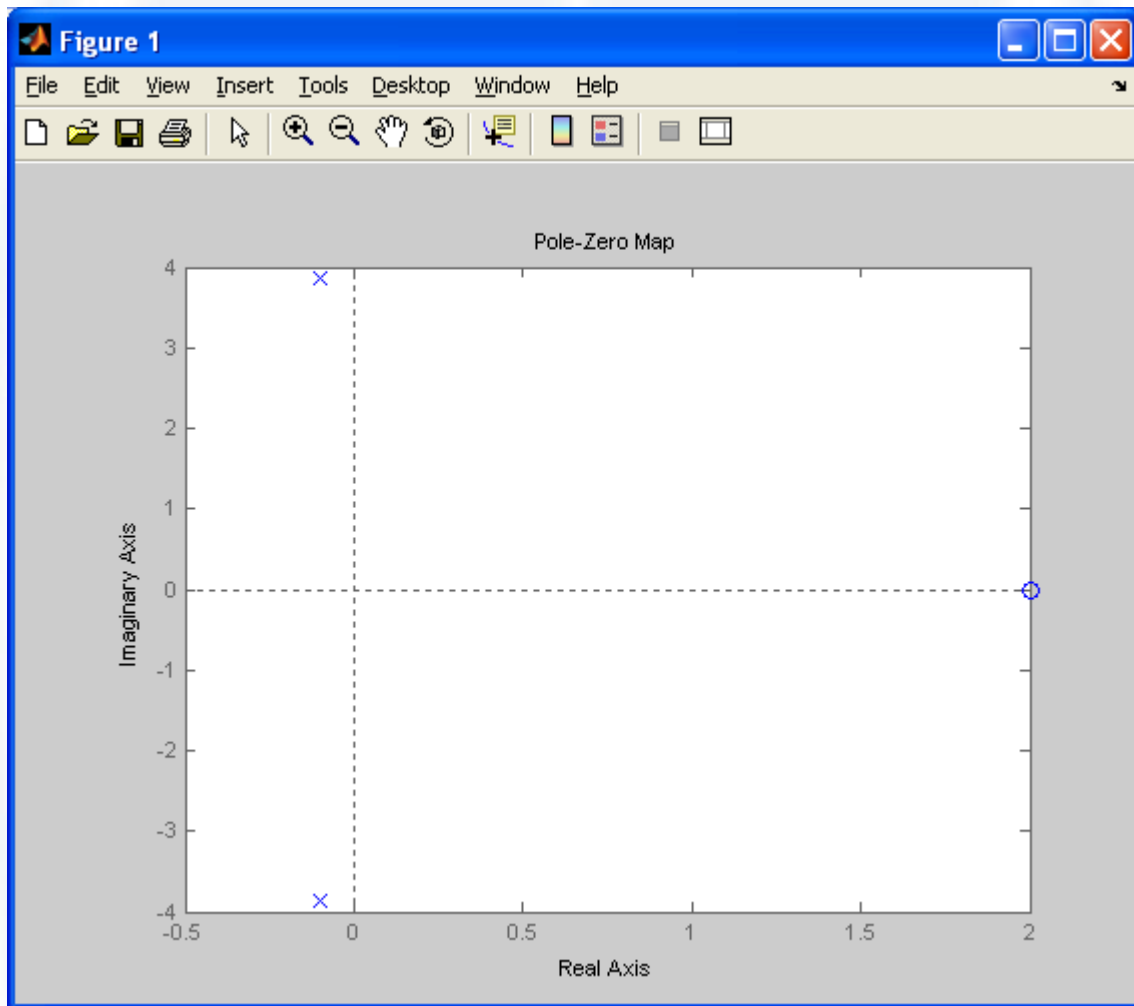


Figura 131: Mapa de polos e zeros para uma função de transferência.

Aplicação a um sistema na forma de estados:

```
clear all
close all
clc

% Matrizes M, C e K do sistema

M = [1 0
     0 2];
```

```

Cd = [ 0.1 -0.1
      -0.1  0.2 ];

K = [ 10 -10
      -10  20 ];

Z = zeros(size(M)); %Matriz de zeros n x n
I = eye(size(M)); %Matriz de identidade n x n

A = [ Z      I
      -M\K -M\Cd]; % Matriz A da forma de estados padrão

B = [Z
      M\I]; % Matriz B da forma de estados padrão

C = eye(size(A)); % Matriz C da forma de estados padrão.

D = zeros(size(A,1),size(B,2)); % Matriz D da forma de estados padrão.

% Neste caso, as saídas do sistema são os próprios estados.

sisFE = ss(A,B,C,D);

pzmap(sisFE)

```

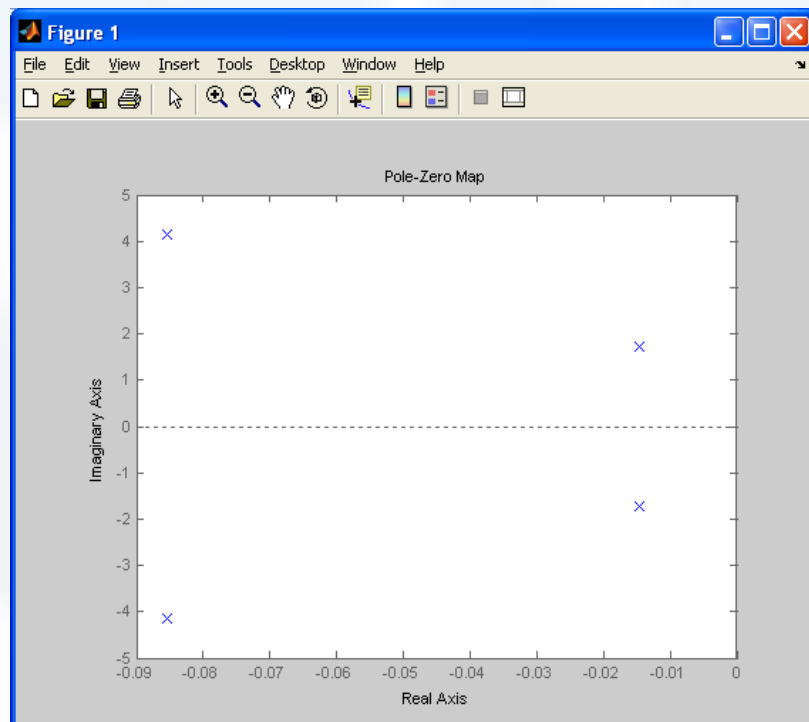


Figura 132: Mapa de polos e zeros para um sistema MIMO.

– Diagrama de margens de estabilidade – *margin*

É similar ao diagrama de bode, mas exibe também as margens de estabilidade (de ganho e fase) do sistema. Esta função pode ser aplicada apenas a funções de transferência simples, ou a sistemas na

forma de estados, desde que ele seja um sistema SISO (Single Input Single Output).

Exemplo:

Aplicação a uma função de transferência simples:

```
clear all
close all
clc

s = tf('s'); % Definição do operador de laplace

H = (s-2)/(s^2+0.2*s+15); % Criação da função de transferência

margin(H)
```

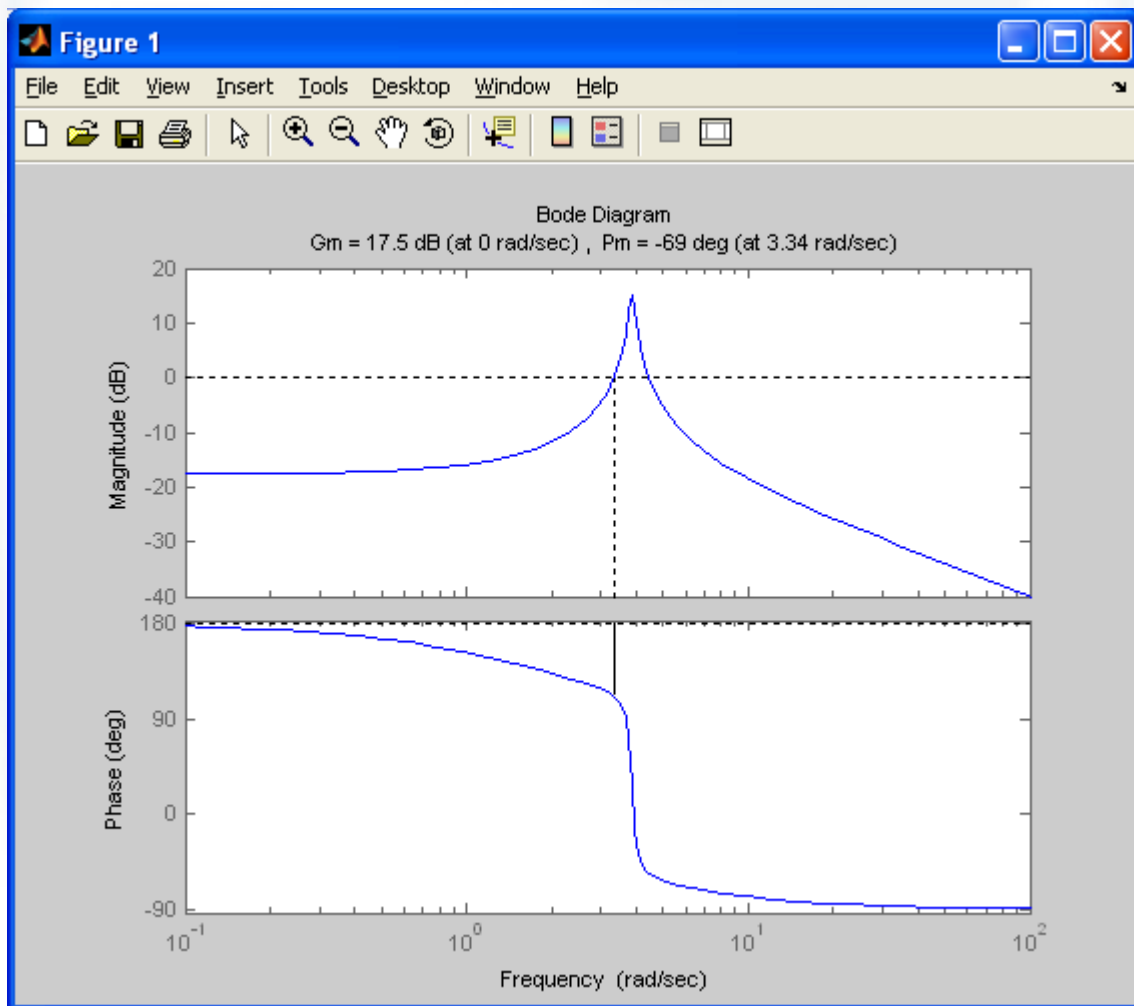


Figura 133: Diagrama de margens de estabilidade para uma função de transferência.

Seção 3.4 – Integradores para forma de estados

As formas padrão de representação de sistemas lineares são as funções ou matrizes de transferência e a forma de estados. Quando os sistemas em estudo são representados por meio da forma de estados, há integradores específicos para a análise destes, de forma que não se precise levantar as matrizes do sistema e utilizar os integradores ODE da maneira convencional.

Lembre-se: Para este tipo de representação, o sistema deve ser linear e invariante no tempo, o que é uma desvantagem em relação aos integradores ODE, pois neles pode-se tratar também sistemas não lineares.

Há quatro tipos de integradores que se pode utilizar para obter as respostas temporais dos sistemas: initial, impulse, step e lsim.

– Initial

Calcula a resposta do sistema a condições iniciais. A sintaxe do comando é:

$$initial(sys,y0)$$

Onde *sys* é o sistema representado na forma de estados, e *y0* é o vetor de condições iniciais. As condições iniciais devem estar na mesma ordem adotada para a criação do vetor de estados. Assim, utilizando a convenção já apresentada, os *n* primeiros estados são os estados de velocidade, e os *n* finais os de deslocamento.

Exemplo:

```
clear all
close all
clc

% Matrizes M, C e K do sistema

M = [1 0
     0 2];

Cd = [ 0.1 -0.1
      -0.1  0.2 ];

K = [ 10 -10
     -10  20 ];

Z = zeros(size(M)); %Matriz de zeros n x n
I = eye(size(M)); %Matriz de identidade n x n

A = [ Z      I
     -M\K -M\Cd]; % Matriz A da forma de estados padrão

B = [Z
     M\I]; % Matriz B da forma de estados padrão

C = eye(size(A)); % Matriz C da forma de estados padrão.
```

```

D = zeros(size(A,1),size(B,2)); % Matriz D da forma de estados padrão.
% Neste caso, as saídas do sistema são os próprios estados.
sisFE = ss(A,B,C,D)
initial(sisFE,[1 2 10 5])

```

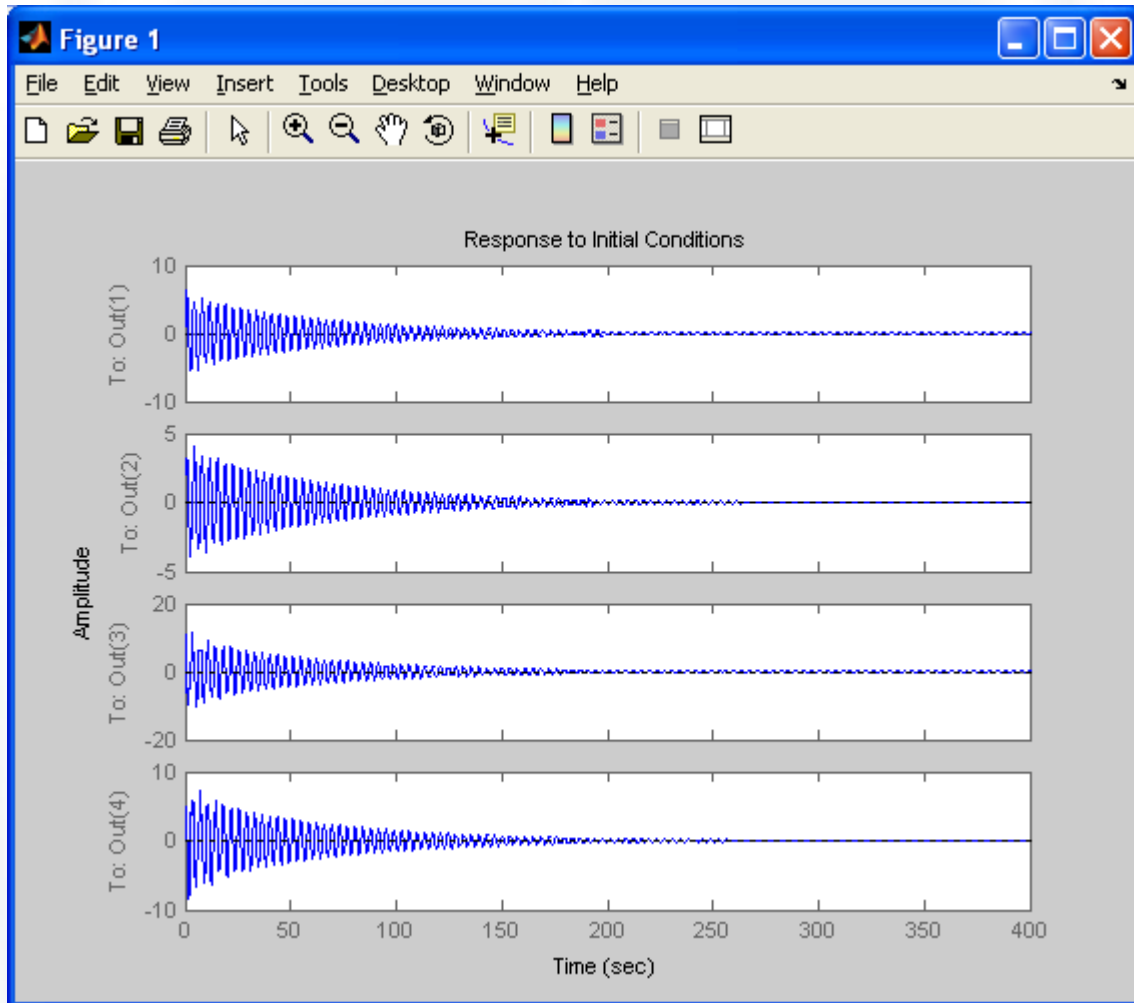


Figura 134: Resposta de um sistema a condições iniciais

– Step

Calcula a resposta temporal do sistema a um degrau unitário aplicado em apenas uma das entradas do sistema por vez, enquanto as outras permanecem sem excitação. Cada coluna, na janela de gráficos representa todas as saídas do sistema, quando o degrau unitário foi aplicado apenas em uma das entradas, ou seja, a primeira coluna mostra todas as saídas resultantes da aplicação de um degrau unitário na primeira entrada apenas, e assim por diante. A sintaxe deste comando é:

```
step(sys)
```

Onde sys é o sistema representado na forma de estados.

Exemplo:

```
clear all
close all
clc

% Matrizes M, C e K do sistema

M = [1 0
     0 2];

Cd = [ 0.1 -0.1
      -0.1  0.2 ];

K = [ 10 -10
     -10  20 ];

Z = zeros(size(M)); %Matriz de zeros n x n
I = eye(size(M)); %Matriz de identidade n x n

A = [ Z      I
     -M\K -M\Cd]; % Matriz A da forma de estados padrão

B = [Z
     M\I]; % Matriz B da forma de estados padrão

C = eye(size(A)); % Matriz C da forma de estados padrão.

D = zeros(size(A,1),size(B,2)); % Matriz D da forma de estados padrão.

% Neste caso, as saídas do sistema são os próprios estados.

sisFE = ss(A,B,C,D)

step(sisFE)
```

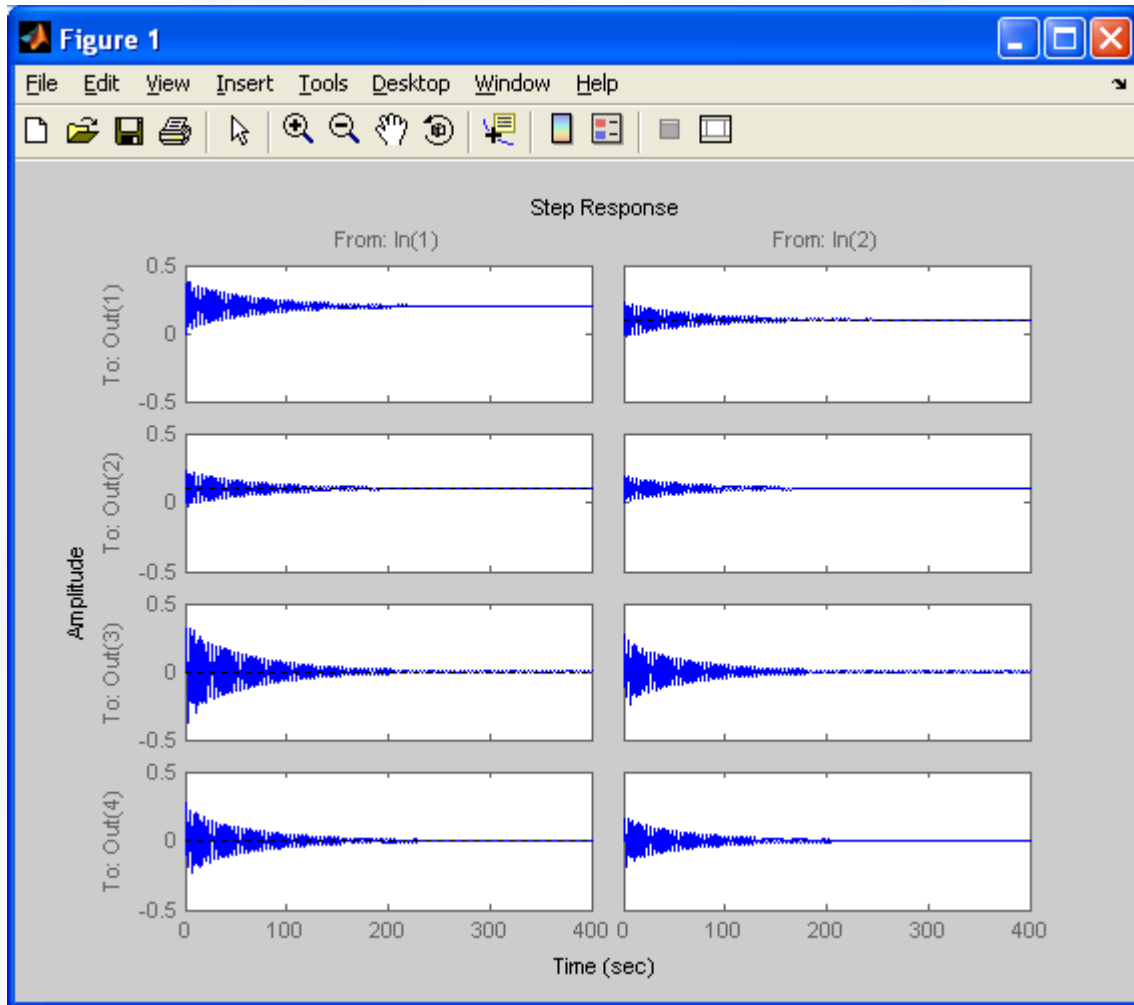


Figura 135: Resposta de um sistema a o degrau unitário.

– Impulse

Calcula a resposta temporal do sistema a um impulso unitário (delta de Dirac), da mesma forma que a função *step*. A sintaxe deste comando é:

impulse(sys)

Onde *sys* é o sistema representado na forma de estados.

Exemplo:

```
clear all
close all
clc

% Matrizes M, C e K do sistema

M = [1 0
     0 2];
```

```
Cd = [ 0.1 -0.1  
      -0.1  0.2 ];  
  
K = [ 10 -10  
      -10  20 ];  
  
Z = zeros(size(M)); %Matriz de zeros n x n  
I = eye(size(M)); %Matriz de identidade n x n  
  
A = [ Z      I  
      -M\K -M\Cd]; % Matriz A da forma de estados padrão  
  
B = [Z  
      M\I]; % Matriz B da forma de estados padrão  
  
C = eye(size(A)); % Matriz C da forma de estados padrão.  
  
D = zeros(size(A,1),size(B,2)); % Matriz D da forma de estados padrão.  
  
% Neste caso, as saídas do sistema são os próprios estados.  
  
sisFE = ss(A,B,C,D)  
  
impulse(sisFE)
```

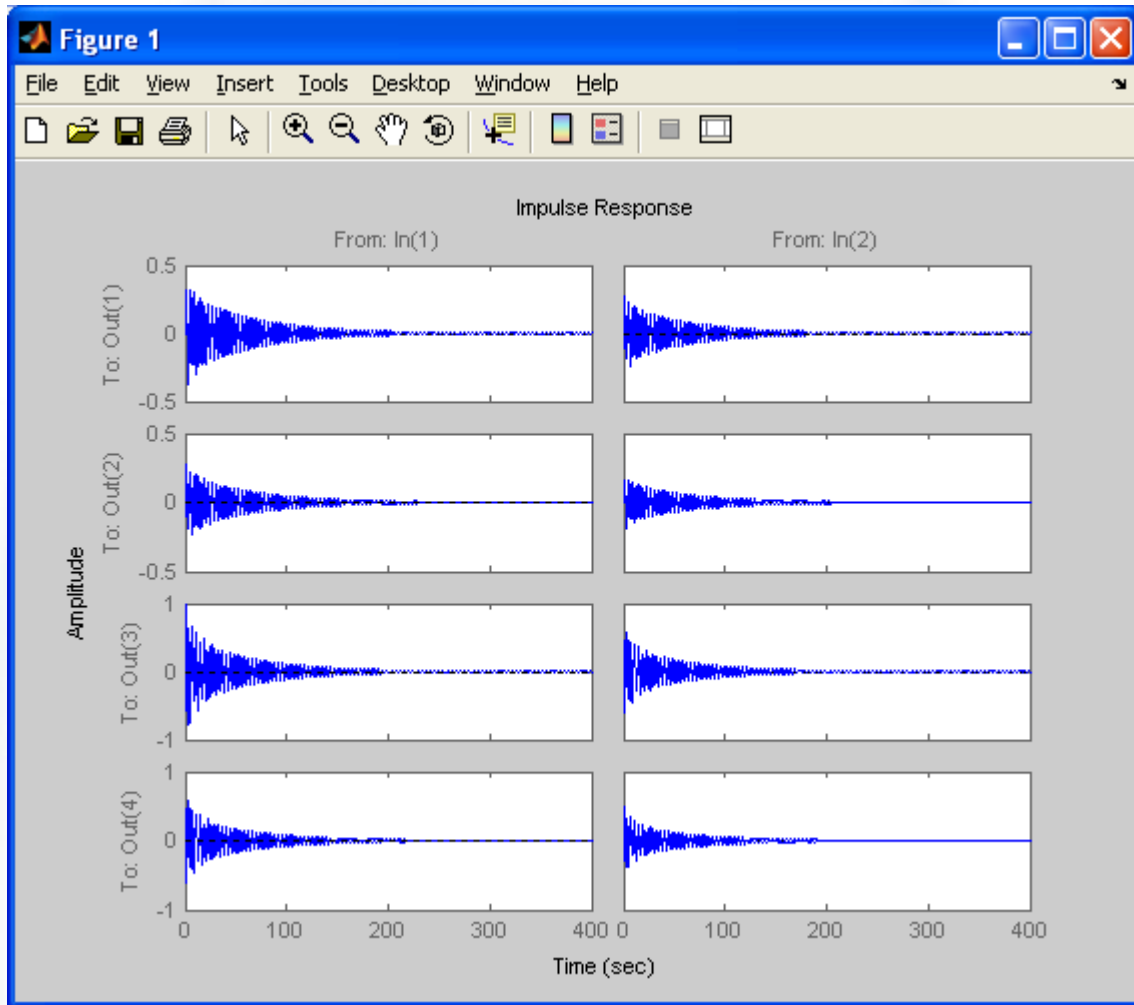



Figura 136: Resposta de um sistema a um impulso unitário.

– Lsim

Calcula a resposta temporal do sistema a uma força excitadora arbitrária. Esta função também aceita sistemas representados na forma de matriz de transferência. Porém, o uso desta função em sistemas representados na forma de estados permite o uso de mais recursos. Este comando apresenta mais de uma sintaxe.

Para apenas plotar a resposta a uma condição de excitação arbitrária devemos utilizar:

$lsim(sys,u,t)$

Onde *sys* é o sistema na forma de estados, *u* é a matriz de forças excitadoras, e *t* é o vetor de tempo desejado para as respostas, com taxa de amostragem constante.

Exemplo:

```

clear all
close all
clc

% Matrizes M, C e K do sistema

M = [0.1  0
      0   0.2];

Cd = [ 0.1 -0.1
       -0.1  0.2 ];

K = [ 10 -10
      -10  20 ];

Z = zeros(size(M)); %Matriz de zeros n x n
I = eye(size(M)); %Matriz de identidade n x n

A = [ Z      I
      -M\K -M\Cd]; % Matriz A da forma de estados padrão

B = [Z
      M\I]; % Matriz B da forma de estados padrão

C = eye(size(A)); % Matriz C da forma de estados padrão.

D = zeros(size(A,1),size(B,2)); % Matriz D da forma de estados padrão.

% Neste caso, as saídas do sistema são os próprios estados.
% O sistema tem 2 entradas (as forças excitadoras) e 4 saídas (os estados).

sisFE = ss(A,B,C,D); % Crio o sistema na forma de estados padrão.

t = 0:0.001:10; % Crio o vetor de tempo com taxa de amostragem constante.

u = [30*sin(5*2*pi*t).' 10*sin(15*2*pi*t).']; % Crio o vetor de entrada

lsim(sisFE,u,t)

```

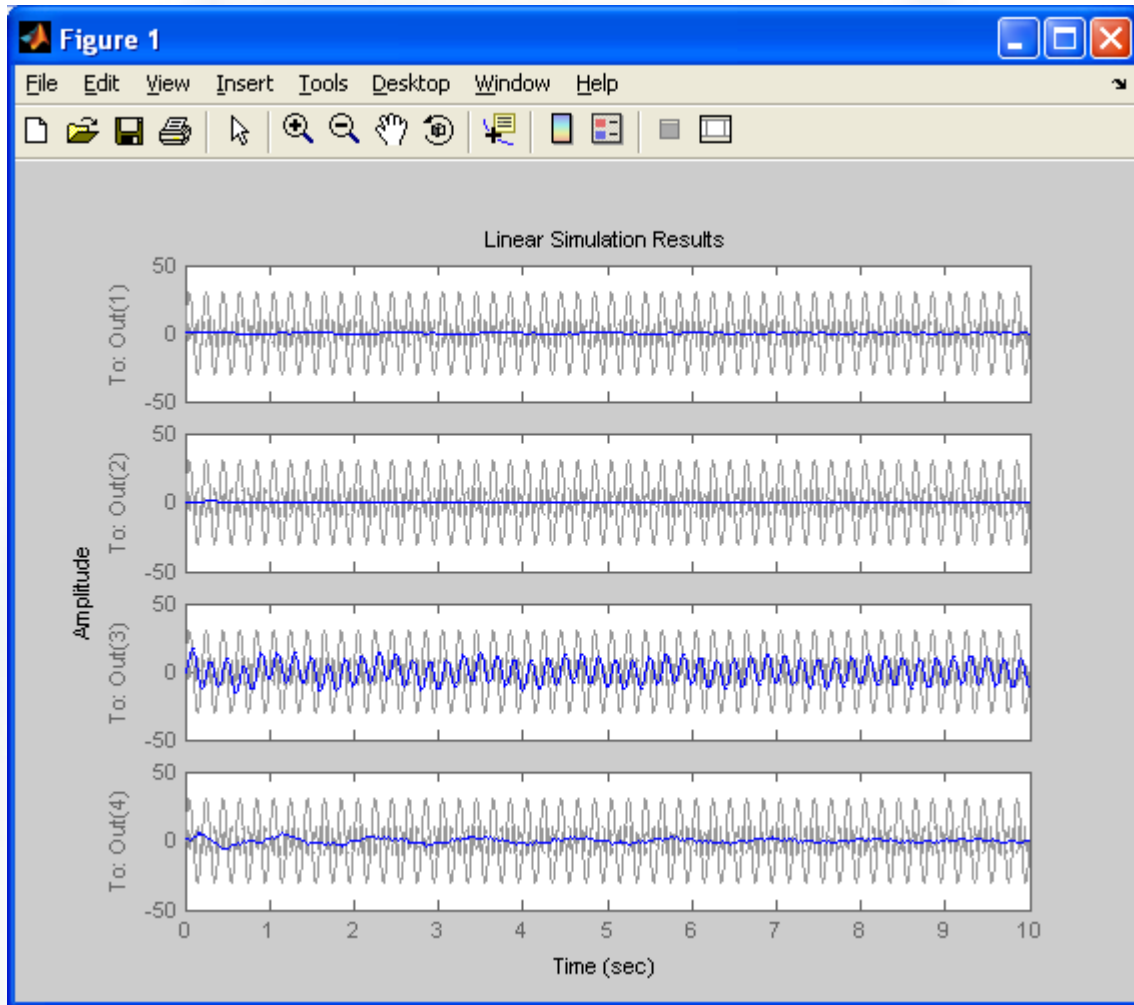


Figura 137: Resposta de um sistema a forças excitadoras arbitrárias.

Em cinza são plotadas as forças excitadoras, e em azul, as respostas. As forças excitadoras estão na mesma ordem adotada para a criação do vetor de excitação $F(t)$ utilizado nas equações de movimento.

Para guardar em um vetor a resposta a uma condição de excitação arbitrária devemos utilizar:

$$[y,t,x] = \text{lsim}(\text{sys},u,t)$$

Onde sys é o sistema na forma de estados, u é a matriz de forças excitadoras, t é o vetor de tempo desejado para as respostas, com taxa de amostragem constante, y é a resposta do sistema, e x é a trajetória dos estados.

Em ambos os casos da lsim , pode-se adicionar um quarto argumento, $y0$, de condições iniciais. Assim, é possível fazer as simulações não só com forças excitadoras arbitrárias, mas também com condições iniciais arbitrárias.