

Modelagem de Sistemas Dinâmicos - Trabalho N^o3

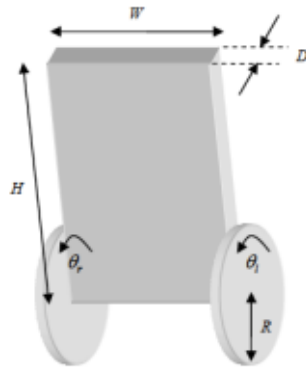
Leonardo Soares da Costa Tanaka - DRE: 121067652

Engenharia de Controle e Automação/UFRJ

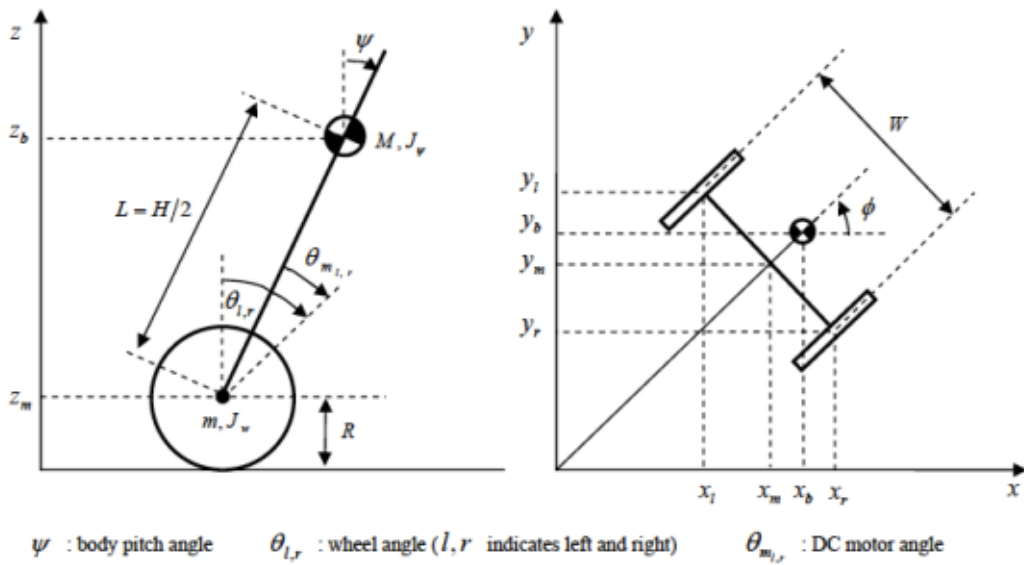
Rio de Janeiro, Brasil

Julho de 2023

Considerando um sistema de pêndulo invertido montado numa plataforma de 2 rodas:



A vista lateral e superior deste sistema com as variáveis associadas é mostrada na seguinte figura:



Os parâmetros físicos deste sistema são dados por:

- $g = 9.8\text{m/s}^2$: gravidade
- $m = 0.03\text{kg}$: peso da roda
- $R = 0.04\text{m}$: raio da roda
- $J_W = \frac{mR^2}{2}$: momento de inércia da roda
- $M = 0.6\text{kg}$: peso do corpo
- $W = 0.14\text{m}$: largura do corpo
- $D = 0.04\text{m}$: profundidade do corpo
- $H = 0.144\text{m}$: altura do corpo
- $L = \frac{H}{2}$: distância do centro de massa do corpo ao eixo da roda
- $J_\psi = \frac{ML^2}{3}\text{kgm}^2$: momento de inércia do corpo em pitch
- $J_\phi = \frac{M(W^2+D^2)}{12}\text{kgm}^2$: momento de inércia do corpo em rumo
- J_m : inércia do motor DC desprezível
- $R_m = 6.69\Omega$: resistência do motor DC
- L_m : indutância do motor desprezível
- $K_t = Ke = 0.4$: constante de torque (Nm/A) e constante de EMF (V s/rad)
- $f_m = 0.0022$: atrito entre o corpo e o motor
- $n = 1$: redução do motor

Considerando os graus de liberdade $\{\theta, \psi, \phi\}$ onde $\theta = (\theta_l + \theta_r)/2 \rightarrow \dot{\theta} = (\dot{\theta}_l + \dot{\theta}_r)/2$.

A posição cartesiana do carrinho (composto pelas 2 rodas), no sistema de coordenadas $\{x, y, x\}$, é dada por:

$$X_m = \begin{bmatrix} x_m \\ y_m \\ z_m \end{bmatrix} = \begin{bmatrix} x_m \\ y_m \\ R \end{bmatrix} \quad (1)$$

Considerando que as rodas do carrinho não escorregam, o carrinho é um sistema com restrições não-holonômicas, sendo que a velocidade do carrinho é dada por:

$$\dot{X}_m = R \begin{bmatrix} \cos(\phi) \\ \sin(\phi) \\ 0 \end{bmatrix} \dot{\theta} \quad (2)$$

A posição cartesiana das rodas (esquerda e direita) no sistema de coordenadas $\{x, y, z\}$, é dada por:

$$\begin{aligned} X_l &= \begin{bmatrix} x_l \\ y_l \\ z_l \end{bmatrix} = X_m + \frac{W}{2} \begin{bmatrix} -\sin(\phi) \\ \cos(\phi) \\ 0 \end{bmatrix} \\ X_r &= \begin{bmatrix} x_r \\ y_r \\ z_r \end{bmatrix} = X_m + \frac{W}{2} \begin{bmatrix} \sin(\phi) \\ -\cos(\phi) \\ 0 \end{bmatrix} \end{aligned} \quad (3)$$

A velocidade cartesiana das rodas é dada por:

$$\begin{aligned} \dot{X}_l &= \dot{X}_m + \frac{W}{2} \begin{bmatrix} -\cos(\phi) \\ -\sin(\phi) \\ 0 \end{bmatrix} \dot{\phi} \\ \dot{X}_r &= \dot{X}_m + \frac{W}{2} \begin{bmatrix} \cos(\phi) \\ \sin(\phi) \\ 0 \end{bmatrix} \dot{\phi} \end{aligned} \quad (4)$$

A posição cartesiana do corpo no sistema de coordenadas $\{x, y, z\}$, é dada por:

$$X_b = \begin{bmatrix} x_b \\ y_b \\ z_b \end{bmatrix} = X_m + L \begin{bmatrix} \cos(\phi)\sin(\psi) \\ \sin(\phi)\sin(\psi) \\ \cos(\psi) \end{bmatrix} \quad (5)$$

A velocidade cartesiana do corpo é dada por:

$$\dot{X}_b = \dot{X}_m + L \begin{bmatrix} \cos(\phi)\cos(\psi) \\ \sin(\phi)\cos(\psi) \\ -\sin(\psi) \end{bmatrix} \dot{\psi} + L \begin{bmatrix} -\sin(\phi)\sin(\psi) \\ \cos(\phi)\sin(\psi) \\ 0 \end{bmatrix} \dot{\phi} \quad (6)$$

Levando em consideração que a massa do motor DC é desprezível e utilizando a fórmula de energia cinética translacional de um único corpo, a energia cinética translacional é dada por:

$$K_{tra} = \frac{1}{2} m \|\vec{v}\|^2 \quad (7)$$

$$\begin{aligned}
T_1 &= \frac{1}{2}m\dot{X}_l^T\dot{X}_l + \frac{1}{2}m\dot{X}_r^T\dot{X}_r + \frac{1}{2}M\dot{X}_b^T\dot{X}_b = \\
&= \frac{1}{2}m \begin{bmatrix} R\cos(\phi)\dot{\theta} - \frac{W}{2}\cos(\phi)\dot{\phi} & R\sin(\phi)\dot{\theta} - \frac{W}{2}\sin(\phi)\dot{\phi} & 0 \end{bmatrix} \begin{bmatrix} R\cos(\phi)\dot{\theta} - \frac{W}{2}\cos(\phi)\dot{\phi} \\ R\sin(\phi)\dot{\theta} - \frac{W}{2}\sin(\phi)\dot{\phi} \\ 0 \end{bmatrix} + \\
&\quad \frac{1}{2}m \begin{bmatrix} R\cos(\phi)\dot{\theta} + \frac{W}{2}\cos(\phi)\dot{\phi} & R\sin(\phi)\dot{\theta} + \frac{W}{2}\sin(\phi)\dot{\phi} & 0 \end{bmatrix} \begin{bmatrix} R\cos(\phi)\dot{\theta} + \frac{W}{2}\cos(\phi)\dot{\phi} \\ R\sin(\phi)\dot{\theta} + \frac{W}{2}\sin(\phi)\dot{\phi} \\ 0 \end{bmatrix} + \\
&\quad \frac{1}{2}M \begin{bmatrix} R\cos(\phi)\dot{\theta} + L\cos(\phi)\cos(\psi)\dot{\psi} - L\sin(\phi)\sin(\psi)\dot{\phi} & R\sin(\phi)\dot{\theta} + L\sin(\phi)\cos(\psi)\dot{\psi} + L\cos(\phi)\sin(\psi)\dot{\phi} & -L\sin(\psi)\dot{\psi} \\ R\sin(\phi)\dot{\theta} + L\sin(\phi)\cos(\psi)\dot{\psi} + L\cos(\phi)\sin(\psi)\dot{\phi} & R\cos(\phi)\dot{\theta} + L\cos(\phi)\cos(\psi)\dot{\psi} - L\sin(\phi)\sin(\psi)\dot{\phi} & -L\sin(\psi)\dot{\psi} \\ -L\sin(\psi)\dot{\psi} & -L\sin(\psi)\dot{\psi} & L^2\sin^2(\psi)\dot{\psi}^2 \end{bmatrix} = \\
&= \frac{1}{2}m \left[R^2\cos(\phi)^2\dot{\theta}^2 - WR\cos(\phi)^2\dot{\theta}\dot{\phi} + \frac{W^2}{4}\cos(\phi)^2\dot{\phi}^2 + R^2\sin(\phi)^2\dot{\theta}^2 - WR\sin(\phi)^2\dot{\theta}\dot{\phi} + \frac{W^2}{4}\sin(\phi)^2\dot{\phi}^2 \right] + \\
&\quad \frac{1}{2}m \left[R^2\cos(\phi)^2\dot{\theta}^2 + WR\cos(\phi)^2\dot{\theta}\dot{\phi} + \frac{W^2}{4}\cos(\phi)^2\dot{\phi}^2 + R^2\sin(\phi)^2\dot{\theta}^2 + WR\sin(\phi)^2\dot{\theta}\dot{\phi} + \frac{W^2}{4}\sin(\phi)^2\dot{\phi}^2 \right] + \\
&\quad \frac{1}{2}M \left[R^2\cos(\phi)^2\dot{\theta}^2 + L^2\cos(\phi)^2\cos(\psi)^2\dot{\psi}^2 + L^2\sin(\phi)^2\sin(\psi)^2\dot{\phi}^2 + 2RL\cos(\phi)^2\cos(\psi)\dot{\theta}\dot{\psi} - 2L^2\sin(\phi)\cos(\phi)\sin(\psi)\dot{\phi}\dot{\psi} - 2RL\sin(\phi)\cos(\phi)\sin(\psi)\dot{\theta}\dot{\phi} \right] + \\
&\quad \frac{1}{2}M \left[R^2\sin(\phi)^2\dot{\theta}^2 + L^2\sin(\phi)^2\cos(\psi)^2\dot{\psi}^2 + L^2\cos(\phi)^2\sin(\psi)^2\dot{\phi}^2 + 2RL\sin(\phi)^2\cos(\psi)\dot{\theta}\dot{\psi} + 2L^2\sin(\phi)\cos(\phi)\sin(\psi)\dot{\phi}\dot{\psi} + 2RL\sin(\phi)\cos(\phi)\sin(\psi)\dot{\theta}\dot{\phi} \right] + \\
&\quad \frac{1}{2}M = L^2\sin(\psi)^2\dot{\psi}^2 = \\
&= \frac{m \left[R^2\dot{\theta}^2 - WR\dot{\theta}\dot{\phi} + \frac{W^2}{4}\dot{\phi}^2 + R^2\dot{\theta}^2 + WR\dot{\theta}\dot{\phi} + \frac{W^2}{4}\dot{\phi}^2 \right] + M \left[R^2\dot{\theta}^2 + L^2\cos(\psi)^2\dot{\psi}^2 + L^2\sin(\psi)^2\dot{\phi}^2 + 2RL\cos(\psi)\dot{\theta}\dot{\psi} + L^2\sin(\psi)^2\dot{\psi}^2 \right]}{2} = \\
&= \frac{m \left[2R^2\dot{\theta}^2 + \frac{W^2}{2}\dot{\phi}^2 \right] + M \left[R^2\dot{\theta}^2 + L^2\dot{\psi}^2 + L^2\sin(\psi)^2\dot{\phi}^2 + 2RL\cos(\psi)\dot{\theta}\dot{\psi} \right]}{2}
\end{aligned} \tag{8}$$

Utilizando a fórmula da energia cinética rotacional:

$$\begin{aligned}
K_{rot} &= \frac{1}{2}\vec{\omega} \cdot \mathbf{I} \cdot \vec{\omega} \\
T_2 &= \frac{J_\phi\dot{\phi}^2}{2} + \frac{J_\psi\dot{\psi}^2}{2} + \frac{J_w(\dot{\theta}_r^2 + \dot{\theta}_l^2)}{2} + \frac{J_m n^2(\dot{\theta}_l - \dot{\psi})^2}{2} + \frac{J_m n^2(\dot{\theta}_r - \dot{\psi})^2}{2}
\end{aligned} \tag{9}$$

Considerando que $\dot{\theta} = (\dot{\theta}_l + \dot{\theta}_r)/2$, temos que:

$$\begin{aligned}
\dot{\theta}_r^2 + \dot{\theta}_l^2 &= 2\dot{\theta}^2 + \frac{W^2}{2R^2}\dot{\phi}^2, \quad (\dot{\theta}_l - \dot{\psi})^2 + (\dot{\theta}_r - \dot{\psi})^2 = 2(\dot{\theta} - \dot{\psi})^2 + \frac{W^2}{2R^2}\dot{\phi}^2 \\
\dot{\phi} &= \frac{R}{W}(\dot{\theta}_r - \dot{\theta}_l)
\end{aligned} \tag{10}$$

Levando em consideração que J_m é desprezível e utilizando as fórmulas acima, então:

$$\begin{aligned}
T_2 &= J_w\dot{\theta}^2 + \frac{1}{2}J_\psi\dot{\psi}^2 + \frac{1}{2} \left(J_\phi + \frac{W^2}{2R^2}J_w + \frac{W^2}{2R^2}n^2J_m \right) \dot{\phi}^2 + n^2J_m(\dot{\theta} - \dot{\psi})^2 \\
T_2 &= J_w\dot{\theta}^2 + \frac{1}{2}J_\psi\dot{\psi}^2 + \frac{1}{2} \left(J_\phi + \frac{W^2}{2R^2}J_w \right) \dot{\phi}^2
\end{aligned} \tag{11}$$

A energia potencial do sistema pode ser descrito pela soma das energias potenciais dos centros de massa das rodas e do corpo, tal que:

$$U = mgz_l + mgz_r + mgz_b = 2mgR + Mg(R + L\cos(\psi)) \quad (12)$$

1 Derivação e obtenção das equações dinâmicas

Levando em consideração que o Lagrangiano do sistema pode ser descrito pela seguinte equação e os valores de T_1 , T_2 e U :

$$L = T_1 + T_2 - U \quad (13)$$

$$L = \frac{m[2R^2\dot{\theta}^2 + \frac{W^2}{2}\dot{\phi}^2] + M[R^2\dot{\theta}^2 + L^2\dot{\psi}^2 + L^2\sin(\psi)^2\dot{\phi}^2 + 2RL\cos(\psi)\dot{\theta}\dot{\psi}]}{2} + J_w\dot{\theta}^2 + \frac{1}{2}J_\psi\dot{\psi}^2 + \frac{1}{2}\left(J_\phi + \frac{W^2}{2R^2}J_w\right)\dot{\phi}^2 - 2mgR - Mg(R + L\cos(\psi))$$

Temos que:

$$\begin{aligned} \frac{\partial L}{\partial \theta} = 0, \quad \frac{\partial L}{\partial \psi} &= ML^2\sin(\psi)\cos(\psi)\dot{\phi}^2 - MRL\sin(\psi)\dot{\theta}\dot{\psi} + MgL\sin(\psi), \quad \frac{\partial L}{\partial \phi} = 0 \\ \frac{\partial L}{\partial \dot{\theta}} &= 2mR^2\dot{\theta} + MR^2\dot{\theta} + 2J_w\dot{\theta} + MRL\cos(\psi)\dot{\theta} \\ \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}}\right) &= 2mR^2\ddot{\theta} + MR^2\ddot{\theta} + 2J_w\ddot{\theta} - MRL\sin(\psi)\dot{\psi}^2 + MRL\cos(\psi)\ddot{\psi} \\ \frac{\partial L}{\partial \dot{\psi}} &= ML^2\dot{\psi} + MRL\cos(\psi)\dot{\theta} + j_\psi\dot{\psi} \\ \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\psi}}\right) &= ML^2\ddot{\psi} + j_\psi\ddot{\psi} - MRL\sin(\psi)\dot{\theta}\dot{\psi} + MRL\cos(\psi)\ddot{\theta} \\ \frac{\partial L}{\partial \dot{\phi}} &= m\frac{W^2}{2}\dot{\phi} + ML^2\sin(\psi)^2\dot{\phi} + \left(J_\phi + \frac{W^2}{2R^2}J_w\right)\dot{\phi} \\ \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\phi}}\right) &= m\frac{W^2}{2}\ddot{\phi} + ML^2\sin(\psi)^2\ddot{\phi} + 2ML^2\sin(\psi)\cos(\psi)\dot{\psi}\dot{\phi} + \left(J_\phi + \frac{W^2}{2R^2}J_w\right)\ddot{\phi} \end{aligned} \quad (14)$$

(a) Utilizando a formulação de Lagrange foi determinado a dinâmica entre as coordenadas generalizadas $\lambda = [\theta \quad \psi \quad \phi]^T$ e as forças (torques) generalizadas $F = [F_\theta \quad F_\psi \quad F_\phi]^T$.

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\lambda}}\right) - \frac{\partial L}{\partial \lambda} = F \quad (15)$$

Utilizando a fórmula acima e outras equações (14) para começar a montar a forma matricial:

$$\begin{aligned} [(2m + M)R^2 + J_w]\ddot{\theta} + MRL\cos(\psi)\ddot{\psi} - MRL\sin(\psi)\dot{\psi}^2 &= F_\theta \\ MRL\cos(\psi)\ddot{\theta} + (ML^2 + J_\psi)\ddot{\psi} - ML^2\sin(\psi)\cos(\psi)\dot{\phi}^2 - MgL\sin(\psi) &= F_\psi \\ \left[m\frac{W^2}{2} + ML^2\sin(\psi)^2 + J_\phi + \frac{W^2}{2R^2}J_w\right]\ddot{\phi} + 2ML^2\sin(\psi)\cos(\psi)\dot{\psi}\dot{\phi} &= F_\phi \end{aligned} \quad (16)$$

Esta dinâmica pode ser escrita em forma matricial como:

$$M(\lambda)\ddot{\lambda} + C(\lambda, \dot{\lambda}) + G(\lambda) = F \quad (17)$$

onde $M(\cdot) \in \mathbb{R}^{3 \times 3}$, $C(\cdot, \cdot) \in \mathbb{R}^3$, $G(\cdot) \in \mathbb{R}^3$.

Utilizando as equações (16) em forma matricial:

$$\begin{bmatrix} (2m + M)R^2 + 2J_w & MRL\cos(\psi) & 0 \\ MRL\cos(\psi) & ML^2 + J_\psi & 0 \\ 0 & 0 & m\frac{W^2}{2} + ML^2\sin(\psi)^2 + J_\phi + \frac{W^2}{2R^2}J_w \end{bmatrix} \ddot{\lambda} + \begin{bmatrix} -MRL\sin(\psi)\dot{\psi}^2 \\ -ML^2\sin(\psi)\cos(\psi)\dot{\psi}^2 \\ 2ML^2\sin(\psi)\cos(\psi)\dot{\psi}\dot{\phi} \end{bmatrix} + \begin{bmatrix} 0 \\ -MgL\sin(\psi) \\ 0 \end{bmatrix} = F \quad (18)$$

(b) Determinando a relação entre as forças generalizadas e os torques de cada um das rodas $\{Fl, Fr\}$:

$$F = \begin{bmatrix} F_\theta \\ F_\psi \\ F_\phi \end{bmatrix} = E \begin{bmatrix} F_l \\ F_r \end{bmatrix} \quad (19)$$

onde $E \in \mathbb{R}^{3 \times 2}$

Observando o sistema de pêndulo invertido e os graus de liberdade, é possível determinar as seguintes relações:

$$\begin{aligned} F_\theta &= F_l + F_r \\ F_\psi &= -F_l - F_r \\ F_\phi &= -\frac{W}{2R}F_l + \frac{W}{2R}F_r \end{aligned} \quad (20)$$

$$F = \begin{bmatrix} F_\theta \\ F_\psi \\ F_\phi \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ -1 & -1 \\ -\frac{W}{2R} & \frac{W}{2R} \end{bmatrix} \begin{bmatrix} F_l \\ F_r \end{bmatrix}$$

(c) Determinando a relação entre as forças (torques) generalizadas $\{F_\theta, F_\psi, F_\phi\}$ e a tensão elétrica aplicada em cada motor $\{Vl, Vr\}$ utilizando o modelo de um motor DC:

$$F = Q\dot{\lambda} + H \begin{bmatrix} V_l \\ V_r \end{bmatrix} \quad (21)$$

onde $Q \in \mathbb{R}^{3 \times 3}$ e $H \in \mathbb{R}^{3 \times 2}$.

Levando em consideração a Lei de Kirchhoff das Tensões, a Equação do Torque, a Lei do Motor e a Lei do Gerador. Além disso, o atrito entre o corpo e o motor e o indutância do motor DC L_m é desprezível:

$$\begin{aligned}
F_{l,r} &= nK_t \dot{i}_{l,r} + f_m(\dot{\psi} - \dot{\theta}_{l,r}) \\
V_{l,r} + K_e(\dot{\psi} - \dot{\theta}_{l,r}) &= L_m \dot{i}_{l,r} + R_m i_{l,r} \\
i_{l,r} &= \frac{V_{l,r} + K_e(\dot{\psi} - \dot{\theta}_{l,r})}{R_m} \\
F_{l,r} &= \left(\frac{nK_t K_e}{R_m} + f_m \right) (\dot{\psi} - \dot{\theta}_{l,r}) + \frac{nK_t V_{l,r}}{R_m}
\end{aligned} \tag{22}$$

Pegando a relação (20) entre as forças generalizadas e os torques de cada um das rodas $\{Fl, Fr\}$ e substituindo $\{Fl, Fr\}$ pelos valores obtidos (22):

$$\begin{aligned}
F &= \begin{bmatrix} 1 & 1 \\ -1 & -1 \\ -\frac{W}{2R} & \frac{W}{2R} \end{bmatrix} \begin{bmatrix} F_l \\ F_r \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ -1 & -1 \\ -\frac{W}{2R} & \frac{W}{2R} \end{bmatrix} \begin{bmatrix} \left(\frac{nK_t K_e}{R_m} + f_m \right) (\dot{\psi} - \dot{\theta}_l) + \frac{nK_t V_r}{R_m} \\ \left(\frac{nK_t K_e}{R_m} + f_m \right) (\dot{\psi} - \dot{\theta}_r) + \frac{nK_t V_r}{R_m} \end{bmatrix} \\
F &= \begin{bmatrix} -2 \left(\frac{nK_t K_e}{R_m} + f_m \right) & 2 \left(\frac{nK_t K_e}{R_m} + f_m \right) & 0 \\ 2 \left(\frac{nK_t K_e}{R_m} + f_m \right) & -2 \left(\frac{nK_t K_e}{R_m} + f_m \right) & 0 \\ 0 & 0 & -\frac{W^2}{2R^2} \left(\frac{nK_t K_e}{R_m} + f_m \right) \end{bmatrix} \begin{bmatrix} \dot{\theta} \\ \dot{\psi} \\ \dot{\phi} \end{bmatrix} + \begin{bmatrix} \frac{nK_t}{R_m} & \frac{nK_t}{R_m} \\ -\frac{nK_t}{R_m} & -\frac{nK_t}{R_m} \\ -\frac{W}{2R} \frac{nK_t}{R_m} & \frac{W}{2R} \frac{nK_t}{R_m} \end{bmatrix} \begin{bmatrix} V_l \\ V_r \end{bmatrix}
\end{aligned} \tag{23}$$

2 Derivação e obtenção das equações de estado

Definindo as variáveis de estado do sistema:

$$x_1 = \lambda = \begin{bmatrix} \theta \\ \psi \\ \phi \end{bmatrix}, \quad x_2 = \dot{\lambda} = \begin{bmatrix} \dot{\theta} \\ \dot{\psi} \\ \dot{\phi} \end{bmatrix} \tag{24}$$

Derivando as variáveis de estado do sistema e utilizando as representação (24) como base:

$$\dot{x}_1 = \dot{\lambda} = x_2 = \begin{bmatrix} \dot{\theta} \\ \dot{\psi} \\ \dot{\phi} \end{bmatrix} \tag{25}$$

Para calcular o $M(\lambda)^{-1}$, será necessário calcular o determinante:

$$\begin{aligned}
\det(M) &= \begin{vmatrix} (2m+M)R^2 + 2J_w & MRL\cos(\psi) & 0 \\ MRL\cos(\psi) & ML^2 + J_\psi & 0 \\ 0 & 0 & m\frac{W^2}{2} + ML\sin(\psi)^2 + J_\phi + \frac{W^2}{2R^2}J_w \end{vmatrix} = \\
&= [(2m+M)R^2 + 2J_w](ML^2 + J_\psi)(m\frac{W^2}{2} + ML\sin(\psi)^2 + J_\phi + \frac{W^2}{2R^2}J_w) - M^2R^2L^2\cos(\psi)^2(m\frac{W^2}{2} + ML\sin(\psi)^2 + J_\phi + \frac{W^2}{2R^2}J_w)
\end{aligned} \tag{26}$$

Além disso, é necessário calcular a matriz adjunta:

$$M_{adj} = \begin{bmatrix} (ML^2 + J_\psi)(m\frac{W^2}{2} + ML^2\sin(\psi)^2 + J_\phi + \frac{W^2}{2R^2}J_w) & -MRL\cos(\psi)(m\frac{W^2}{2} + ML^2\sin(\psi)^2 + J_\phi + \frac{W^2}{2R^2}J_w) & 0 \\ -MRL\cos(\psi)(m\frac{W^2}{2} + ML^2\sin(\psi)^2 + J_\phi + \frac{W^2}{2R^2}J_w) & [(2m+M)R^2 + 2J_w](m\frac{W^2}{2} + ML^2\sin(\psi)^2 + J_\phi + \frac{W^2}{2R^2}J_w) & 0 \\ 0 & 0 & [(2m+M)R^2 + 2J_w](ML^2 + J_\psi) - M^2R^2L^2\cos(\psi)^2 \end{bmatrix} \quad (27)$$

Utilizando o determinante e a matriz adjunta para encontrar $M(\lambda)^{-1}$:

$$M(\lambda)^{-1} = \frac{M_{adj}}{\det(M)} = \begin{bmatrix} \frac{ML^2 + J_\psi}{[(2m+M)R^2 + 2J_w](ML^2 + J_\psi) - M^2R^2L^2\cos(\psi)^2} & -\frac{MRL\cos(\psi)}{[(2m+M)R^2 + 2J_w](ML^2 + J_\psi) - M^2R^2L^2\cos(\psi)^2} & 0 \\ -\frac{MRL\cos(\psi)}{[(2m+M)R^2 + 2J_w](ML^2 + J_\psi) - M^2R^2L^2\cos(\psi)^2} & \frac{(2m+M)R^2 + 2J_w}{[(2m+M)R^2 + 2J_w](ML^2 + J_\psi) - M^2R^2L^2\cos(\psi)^2} & 0 \\ 0 & 0 & \frac{1}{m\frac{W^2}{2} + ML\sin(\psi)^2 + J_\phi + \frac{W^2}{2R^2}J_w} \end{bmatrix} \quad (28)$$

Com isso, tenho tudo que é preciso para preencher a seguinte equação:

$$\ddot{x}_2 = \ddot{\lambda} = M(\lambda)^{-1} \left[-C(\lambda, \dot{\lambda}) - G(\lambda) + F(\dot{\lambda}, V) \right] \quad (29)$$

3 Determinação dos pontos de equilíbrio do sistema

Considerando $V_l = V_r = 0$, é possível determinar os pontos de equilíbrio de forma trivial. Já que um pêndulo invertido montado numa plataforma de 2 rodas só estará em equilíbrio sem a diferença de potencial nos motores, quando estiver em pé ou de cabeça para baixo verticalmente e sem nenhuma velocidade angular em nenhuma das direções. Logo:

$$\bar{\psi} = \{0, \pm\pi, \pm2\pi, \dots, \pm n\pi\}, n \in \mathbb{N}$$

$$\bar{x}_2 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad (30)$$

Então, o $\bar{\theta}$ e o $\bar{\phi}$ podem ser quaisquer valores porque não afetam no equilíbrio do sistema.

4 Simulação do sistema não linear com $V_l = V_r = 0$

Podemos simular numericamente o sistema através da função ode45 no matlab ou de uma S-functions com um bloco do mesmo nome no Simulink

(a) Função ode45 (Matlab):

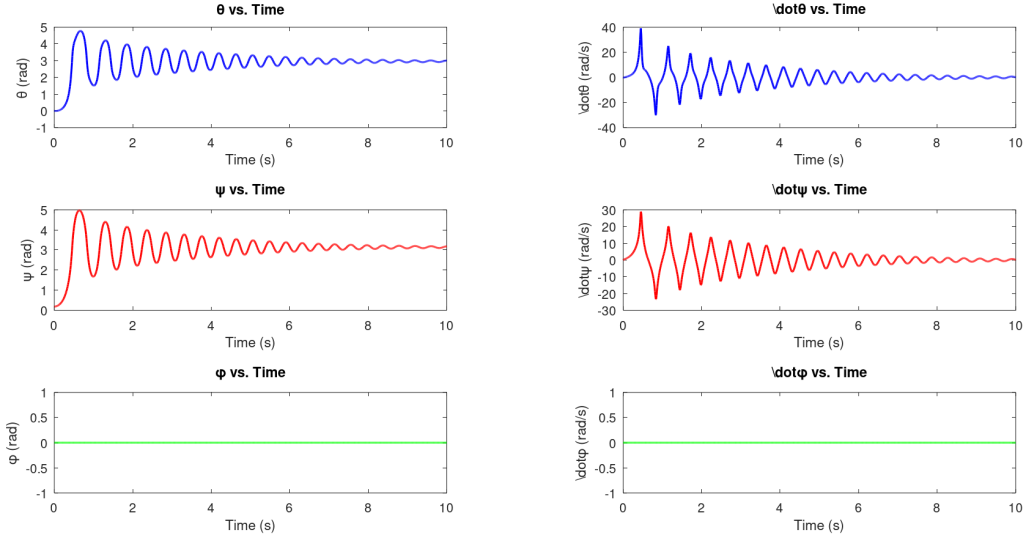


Figura 1: Simulação com condições iniciais $x_0 = [0; 0.18; 0; 0; 0; 0]$

Observando o gráfico é possível observar o comportamento de pêndulo invertido do sistema, que não tem variação em ϕ e $\dot{\phi}$ ao ser colocado zero nas condições iniciais. Porém, as outras variáveis de estado ligadas a posição tendem a π e as variáveis de estado ligadas a velocidade tendem a 0.

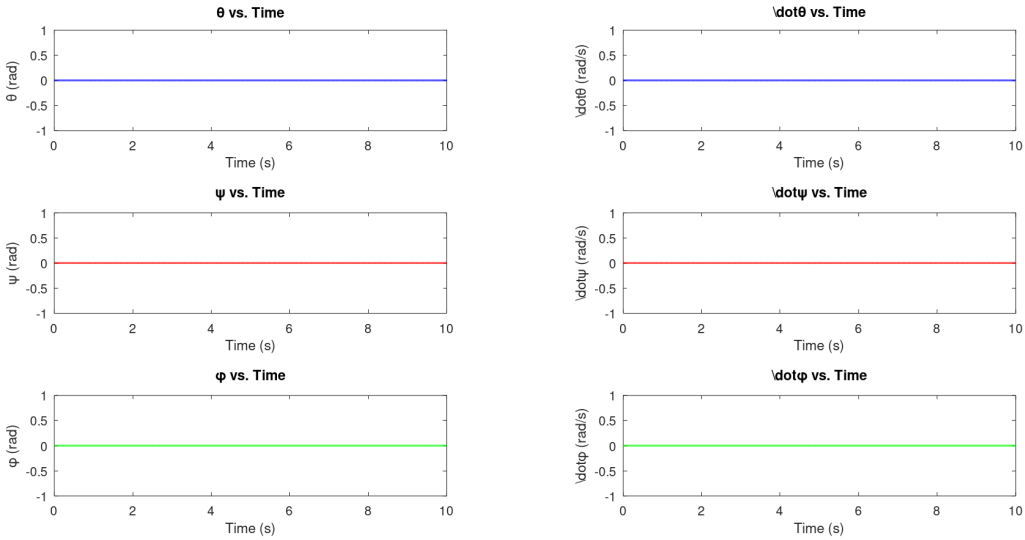


Figura 2: Simulação com condições iniciais $x_0 = [0; 0; 0; 0; 0; 0]$

Ao colocar as condições iniciais em um dos pontos de equilíbrio, o sistema não tem variação das suas variáveis de estado o que acaba comprovando um dos casos propostos na parte 3 do relatório.

(b) S-functions (Simulink)

Não foi possível implementar a S-function, porque o Simulink que está disponível atualmente r2023a não consegue rodar o template que foi enviado. Então, vou mostrar o máximo que consegui chegar de desenvolvimento. Começando pelo penduloinvertido.m :

Listing 1: penduloinvertido.m

```

1  function [sys,x0,str,ts] = penduloinvertido(t,x,flag)
2      switch flag
3          case 0
4              str = [] ;
5              ts = [0 0] ;
6              s = simsizes ;
7              s.NumContStates = 6;
8              s.NumDiscStates = 0 ;
9              s.NumOutputs = 6 ;
10             s.NumInputs = 2 ;
11             s.DirFeedthrough = 0 ;
12             s.NumSampleTimes = 1 ;
13             sys = simsizes(s) ;
14             x0 = [0 0 0 0 0 0] ;
15         case 1
16             sys = penduloinvertido_f(t,x) ;
17         case 3
18             sys = x;
19         case {2 4 9}
20             sys = [];
21         otherwise
22             error(['unhandled flag =',num2str(flag)]) ;
23     end
24 end

```

Além disso a função que estaria na S-function:

Listing 2: penduloinvertido_f.m

```

1  function dx = penduloinvertido_f(t, x)
2      % Variaveis de estado
3      theta = x(1);
4      psi = x(2);
5      phi = x(3);
6      % Derivadas das variaveis de estado
7      theta_dot = x(4);
8      psi_dot = x(5);
9      phi_dot = x(6);
10     % Parametros do sistema
11     g = 9.8; % gravidade

```

```

12     m = 0.03; % peso da roda
13     R = 0.04; % raio da roda
14     Jw = m * R^2 / 2; % momento de inercia da roda
15     M = 0.6; % peso do corpo
16     W = 0.14; % largura do corpo
17     D = 0.04; % profundidade do corpo
18     H = 0.144; % altura do corpo
19     L = H / 2; % distancia do centro de massa do corpo ao eixo da roda
20     Jphi = M * L^2 / 3; % momento de inercia do corpo em pitch
21     Jpsi = M * (W^2 + D^2) / 12; % momento de inercia do corpo em rumo
22     Rm = 6.69; % resistencia do motor DC
23     Kt = 0.4; % constante de torque (Nm/A)
24     Ke = 0.4; % e constante de EMF (V s/rad)
25     fm = 0.0022; % atrito entre o corpo e o motor
26     n = 1; % reducao do motor
27     % Matriz de inercia inversa
28     M_inv = inv([
29         R^2 * (2 * m + M) + 2 * Jw, M * R * L * cos(psi), 0;
30         M * R * L * cos(psi), M * L^2 + Jpsi, 0;
31         0, 0, m * (W^2 / 2) + M * L * sin(psi)^2 + Jphi + (W^2 / (2 * R^2)) * Jw
32     ]);
33     % Matriz G
34     G = [
35         0;
36         -M * g * L * sin(psi);
37         0];
38     % Matriz C
39     C = [
40         -M * R * L * sin(psi) * psi_dot^2;
41         -M * L^2 * sin(psi) * cos(psi) * phi_dot^2;
42         M * L^2 * sin(2 * psi) * psi_dot * phi_dot
43     ];
44     % Matriz Q
45     Q = [
46         -2 * ((n * Kt * Ke / Rm) + fm), 2 * ((n * Kt * Ke / Rm) + fm), 0;
47         2 * ((n * Kt * Ke / Rm) + fm), -2 * ((n * Kt * Ke / Rm) + fm), 0;
48         0, 0, -W^2 / (2 * R^2) * ((n * Kt * Ke / Rm) + fm)
49     ];
50     F = Q*[theta_dot; psi_dot; phi_dot]
51     % Vetor de entrada
52     V = [0; 0]; % [V_l; V_r]
53     % Sistema de equacoes diferenciais
54     dx = zeros(6, 1);
55     dx(1:3) = [theta_dot; psi_dot; phi_dot];
56     dx(4:6) = M_inv * (F - G - C);
57     end

```

Ademais, montei no simulink como funcionaria o sistema não-linear. Tendo uma entrada com amplitude 0 e seis saídas que seriam os estados do sistema que poderiam ser visualizadas pelos scopes.

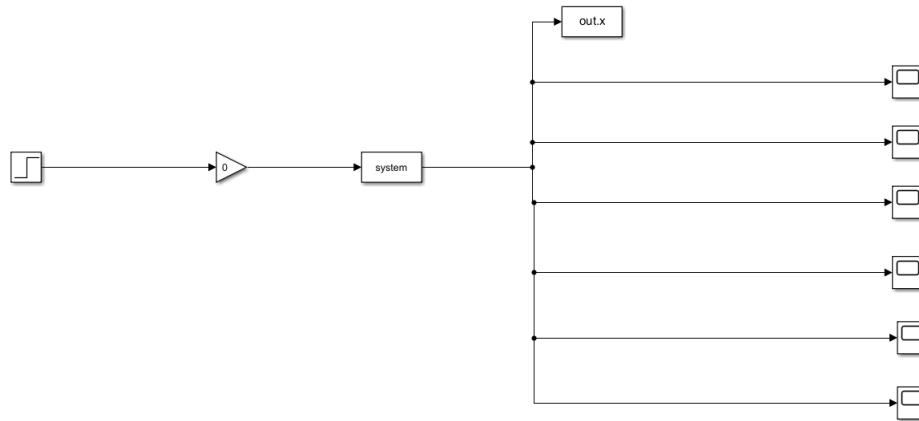


Figura 3: Simulink

(c) `scipy.integrate.solve_ivp` (Python):

Foi obtido um resultado muito similar ao da função `ode45` do Matlab:

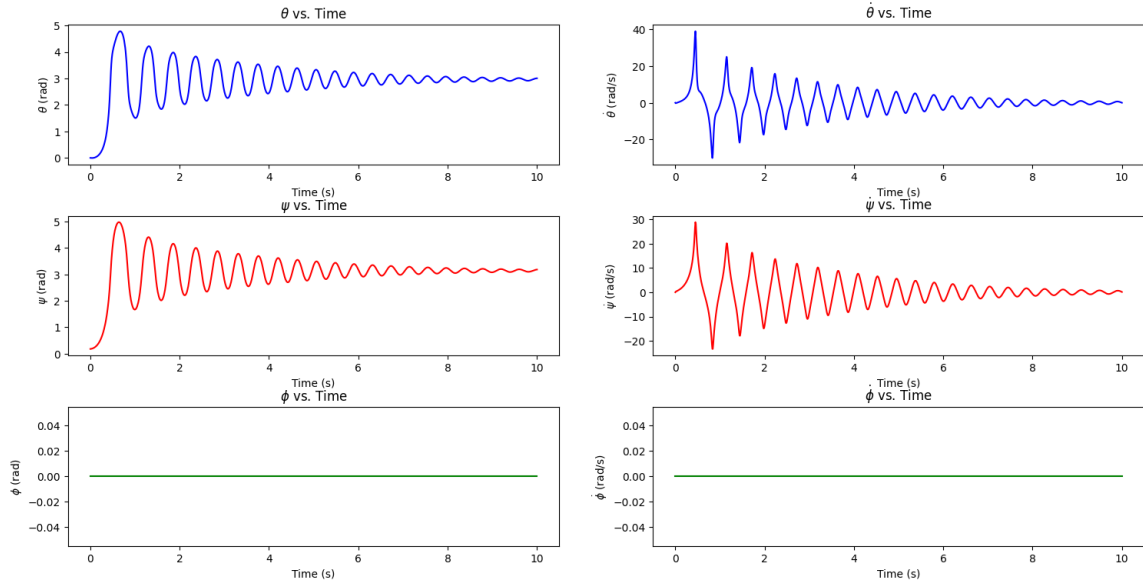


Figura 4: Simulação com condições iniciais $x_0 = [0; 0.18; 0; 0; 0; 0]$

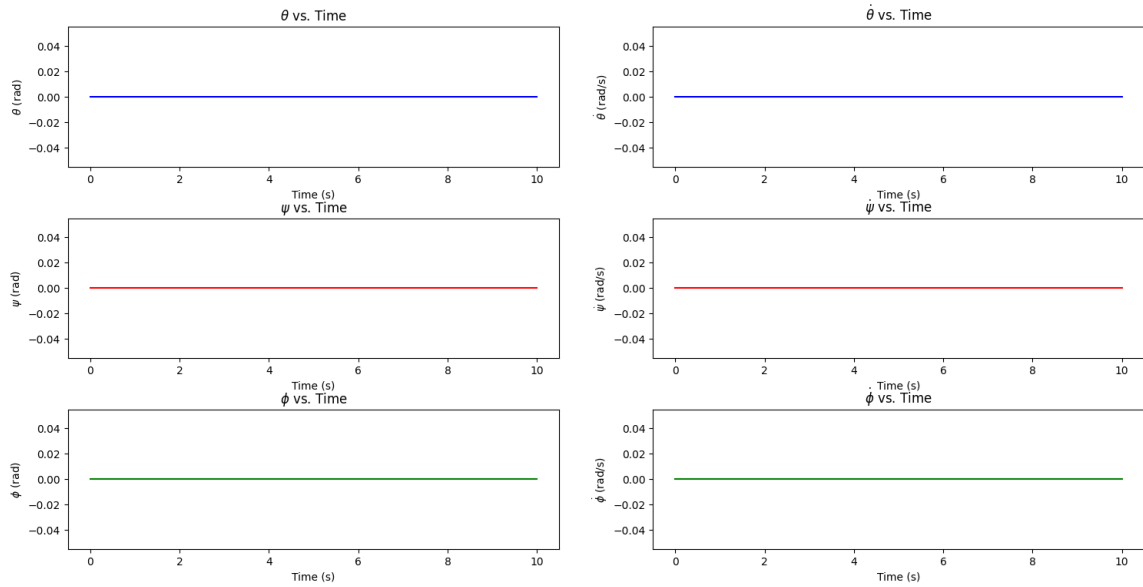


Figura 5: Simulação com condições iniciais $x_0 = [0; 0; 0; 0; 0; 0]$

5 Linearização do sistema com $\psi = 0$

O sistema será linearizado em relação ao ponto de equilíbrio quando $\psi = 0$. Logo, teremos as seguintes aproximações:

$$\begin{aligned}
 \tilde{\theta} &= \theta - \bar{\theta}, \tilde{\psi} = \psi - \bar{\psi}, \tilde{\phi} = \phi - \bar{\phi} \\
 \dot{\tilde{\theta}} &= \dot{\theta} - \dot{\bar{\theta}}, \dot{\tilde{\psi}} = \dot{\psi} - \dot{\bar{\psi}}, \dot{\tilde{\phi}} = \dot{\phi} - \dot{\bar{\phi}} \\
 \sin\psi &= \tilde{\psi} \\
 \cos\psi &= 1
 \end{aligned} \tag{31}$$

Assim, obtemos a linearização do sistema:

$$\begin{aligned}
 M^{-1} &= \begin{bmatrix} \frac{ML^2 + J_\psi}{[(2m+M)R^2 + 2J_w](ML^2 + J_\psi) - M^2 R^2 L^2} & -\frac{MRL}{[(2m+M)R^2 + 2J_w](ML^2 + J_\psi) - M^2 R^2 L^2} & 0 \\ -\frac{MRL}{[(2m+M)R^2 + 2J_w](ML^2 + J_\psi) - M^2 R^2 L^2} & \frac{(2m+M)R^2 + 2J_w}{[(2m+M)R^2 + 2J_w](ML^2 + J_\psi) - M^2 R^2 L^2} & 0 \\ 0 & 0 & \frac{1}{m\frac{W^2}{2} + \frac{W^2}{2R^2} J_w} \end{bmatrix} \\
 C &= \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \\
 G(\lambda) &= \begin{bmatrix} 0 \\ -MgL\tilde{\psi} \\ 0 \end{bmatrix} \\
 \dot{\tilde{x}}_1 &= \begin{bmatrix} \dot{\tilde{\theta}} \\ \dot{\tilde{\psi}} \\ \dot{\tilde{\phi}} \end{bmatrix} \\
 \dot{\tilde{x}}_2 &= M^{-1} \left[-G(\tilde{\lambda}) + F(\tilde{\lambda}, \tilde{V}) \right]
 \end{aligned} \tag{32}$$

6 Simulação do sistema linearizado com $V_l = V_r = 0$

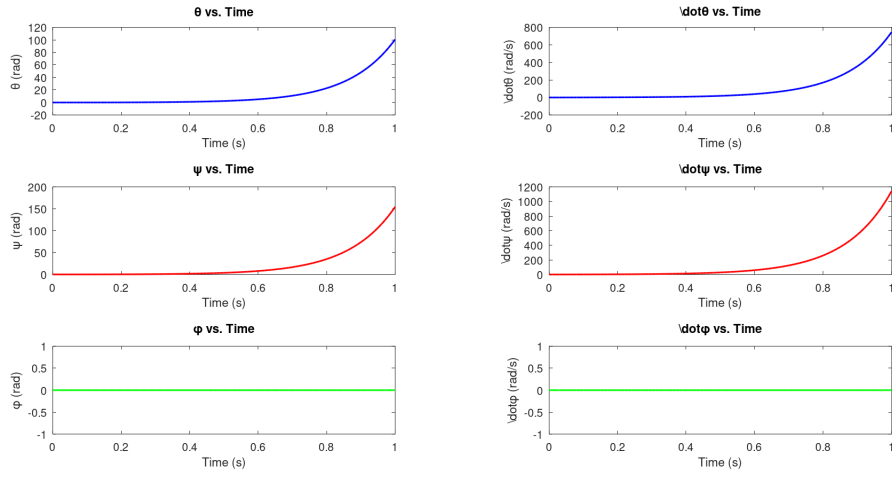


Figura 6: Simulação com condições iniciais $x_0 = [0; 0.18; 0; 0; 0; 0]$

Observando o gráfico fica bem claro que a linearização não é muito conveniente após passar de $\psi > \pi$. É possível ver a gigantesca de diferença entre o sistema não-linear e o sistema linearizado.

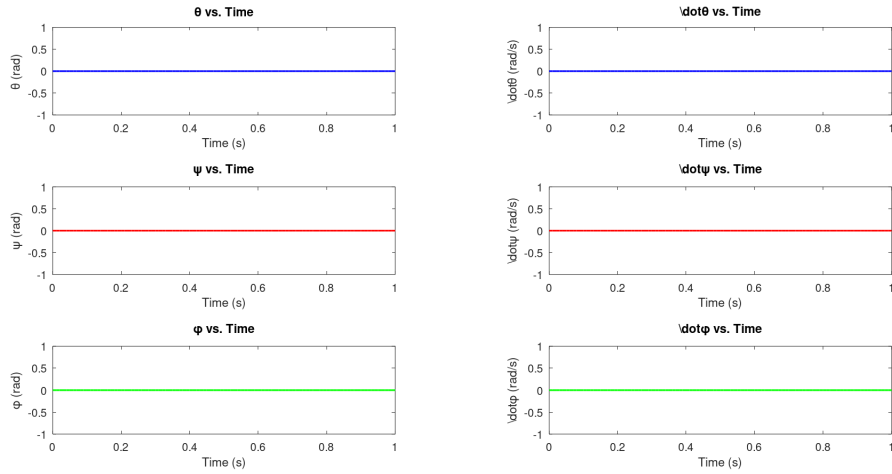


Figura 7: Simulação com condições iniciais $x_0 = [0; 0; 0; 0; 0; 0]$

Já no caso de colocar as condições iniciais $x_0 = [0; 0; 0; 0; 0; 0]$, o comportamento do sistema se mantém no ponto de equilíbrio.

7 Controle realimentado

O sistema considerado é instável em $\psi = 0$. Desta forma é projetado o seguinte controle realimentado:

$$u = -Kx + Nr \quad (33)$$

onde $u^T = [V_l \ V_r]$, $x^T = [\theta \ \psi \ \phi \ \dot{\theta} \ \dot{\psi} \ \dot{\phi}]$, r é uma entrada de referências para comandar sistema para frente, $N^T = [-0.7071 \ 0.7071]$ e

$$K = \begin{bmatrix} -0.0224 & -25.4867 & -0.7071 & -1.0362 & -2.2530 & -0.0076 \\ -0.0224 & -25.4867 & 0.7071 & -1.0362 & -2.2530 & 0.0076 \end{bmatrix} \quad (34)$$

Foi simulado a resposta do sistema não linear com este controle realimentado, considerando diferentes condições iniciais.

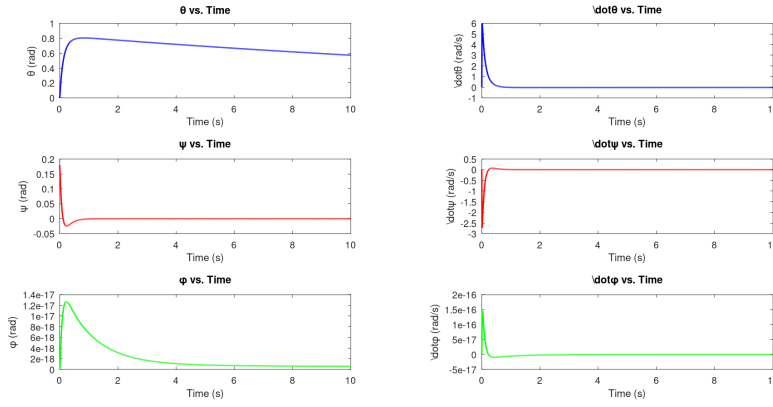


Figura 8: Simulação com condições iniciais $x_0 = [0; 0.18; 0; 0; 0; 0]$

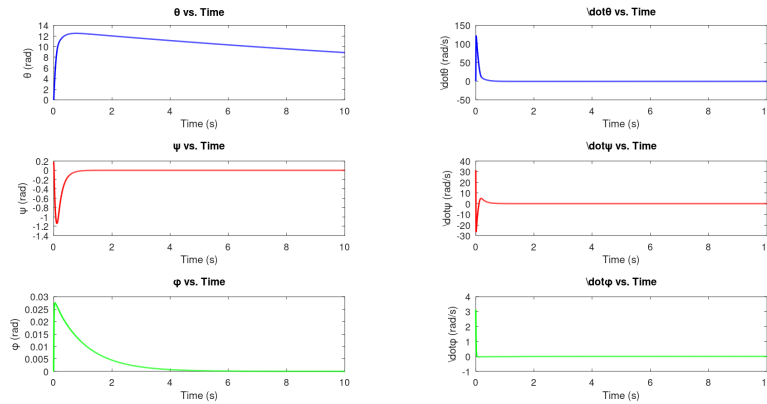


Figura 9: Simulação com condições iniciais $x_0 = [0; 0.18; 0; 0; 31.4; 3.14]$

É possível observar que o controle realimentado tem como objetivo equilibrar o sistema de uma forma que se torne próximo da estabilidade.

A Código: Simulação do sistema não-linear

Listing 3: Código da simulação do sistema não-linear com ode45

```

1 function pendulum_simulation()
2     % Initial conditions
3     x0 = [0; 0.18; 0; 0; 0; 0]; % [theta; psi; phi; dot{theta}; dot{psi}; dot{phi}]
4     % Time span for simulation
5     tspan = 0:0.01:10; % From 0 to 10 seconds with a time step of 0.01 seconds
6     % Solve the differential equations using ode45 solver
7     [t, x] = ode45(@penduloinvertido_f, tspan, x0);
8     % Extract the states for plotting
9     theta = x(:, 1);
10    psi = x(:, 2);
11    phi = x(:, 3);
12    dot_theta = x(:, 4);
13    dot_psi = x(:, 5);
14    dot_phi = x(:, 6);
15    % Plot the results
16    figure;
17    subplot(3, 2, 1);
18    plot(t, theta, 'b', 'LineWidth', 1.5);
19    xlabel('Time (s)');
20    ylabel('\theta (rad)');
21    title('\theta vs. Time');
22    subplot(3, 2, 3);
23    plot(t, psi, 'r', 'LineWidth', 1.5);
24    xlabel('Time (s)');
25    ylabel('\psi (rad)');
26    title('\psi vs. Time');
27    subplot(3, 2, 5);
28    plot(t, phi, 'g', 'LineWidth', 1.5);
29    xlabel('Time (s)');
30    ylabel('\phi (rad)');
31    title('\phi vs. Time');
32    subplot(3, 2, 2);
33    plot(t, dot_theta, 'b', 'LineWidth', 1.5);
34    xlabel('Time (s)');
35    ylabel('\dot{\theta} (rad/s)');
36    title('\dot{\theta} vs. Time');
37    subplot(3, 2, 4);
38    plot(t, dot_psi, 'r', 'LineWidth', 1.5);
39    xlabel('Time (s)');
40    ylabel('\dot{\psi} (rad/s)');
41    title('\dot{\psi} vs. Time');
42    subplot(3, 2, 6);

```

```

43     plot(t, dot_phi, 'g', 'LineWidth', 1.5);
44     xlabel('Time (s)');
45     ylabel('\dot{\phi} (rad/s)');
46     title('\dot{\phi} vs. Time');
47 end
48
49 function dx = penduloinvertido_f(t, x)
50     % Variaveis de estado
51     theta = x(1);
52     psi = x(2);
53     phi = x(3);
54     % Derivadas das variaveis de estado
55     theta_dot = x(4);
56     psi_dot = x(5);
57     phi_dot = x(6);
58     % Parametros do sistema
59     g = 9.8; % gravidade
60     m = 0.03; % peso da roda
61     R = 0.04; % raio da roda
62     Jw = m * R^2 / 2; % momento de inercia da roda
63     M = 0.6; % peso do corpo
64     W = 0.14; % largura do corpo
65     D = 0.04; % profundidade do corpo
66     H = 0.144; % altura do corpo
67     L = H / 2; % distancia do centro de massa do corpo ao eixo da roda
68     Jphi = M * L^2 / 3; % momento de inercia do corpo em pitch
69     Jpsi = M * (W^2 + D^2) / 12; % momento de inercia do corpo em rumo
70     Rm = 6.69; % resistencia do motor DC
71     Kt = 0.4; % constante de torque (Nm/A)
72     Ke = 0.4; % e constante de EMF (V s/rad)
73     fm = 0.0022; % atrito entre o corpo e o motor
74     n = 1; % reducao do motor
75     % Matriz de inercia inversa
76     M_inv = inv([
77         R^2 * (2 * m + M) + 2 * Jw, M * R * L * cos(psi), 0;
78         M * R * L * cos(psi), M * L^2 + Jpsi, 0;
79         0, 0, m * (W^2 / 2) + M * L * sin(psi)^2 + Jphi + (W^2 / (2 * R^2)) * Jw
80     ]);
81     % Matriz G
82     G = [
83         0;
84         -M * g * L * sin(psi);
85         0
86     ];
87     % Matriz C
88     C = [

```

```

89     -M * R * L * sin(psi) * psi_dot^2;
90     -M * L^2 * sin(psi) * cos(psi) * phi_dot^2;
91     M * L^2 * sin(2 * psi) * psi_dot * phi_dot
92 ];
93 % Matriz Q
94 Q = [
95     -2 * ((n * Kt * Ke / Rm) + fm), 2 * ((n * Kt * Ke / Rm) + fm), 0;
96     2 * ((n * Kt * Ke / Rm) + fm), -2 * ((n * Kt * Ke / Rm) + fm), 0;
97     0, 0, -W^2 / (2 * R^2) * ((n * Kt * Ke / Rm) + fm)
98 ];
99 F = Q*[theta_dot; psi_dot; phi_dot]
100 % Vetor de entrada
101 V = [0; 0]; % [V_l; V_r]
102 % Sistema de equacoes diferenciais
103 dx = zeros(6, 1);
104 dx(1:3) = [theta_dot; psi_dot; phi_dot];
105 dx(4:6) = M_inv * (F - G - C);
106 end
107
108 pendulum_simulation();

```

B Código: Simulação do sistema não-linear em Python

```

import numpy as np
import matplotlib.pyplot as plt
from scipy.integrate import solve_ivp

def pendulum_simulation():
    # Initial conditions
    x0 = [0, 0, 0, 0, 0, 0] # [theta; psi; phi; dot{theta}; dot{psi}; dot{phi}]

    # Time span for simulation
    tspan = np.arange(0, 10.01, 0.01) # From 0 to 10 seconds with a time step of 0.01 seconds

    # Solve the differential equations using solve_ivp
    sol = solve_ivp(penduloinvertido_f, [tspan[0], tspan[-1]], x0, t_eval=tspan)

    # Extract the states for plotting
    t = sol.t

    theta, psi, phi, dot_theta, dot_psi, dot_phi = sol.y

    # Plot the results
    fig, axs = plt.subplots(3, 2, figsize=(12, 10))
    axs[0, 0].plot(t, theta, 'b', linewidth=1.5)
    axs[0, 0].set_xlabel('Time (s)')

```

```

    axs[0, 0].set_ylabel(r'$\theta$ (rad)')
    axs[0, 0].set_title(r'$\theta$ vs. Time')
    axs[1, 0].plot(t, psi, 'r', linewidth=1.5)
    axs[1, 0].set_xlabel('Time (s)')
    axs[1, 0].set_ylabel(r'$\psi$ (rad)')
    axs[1, 0].set_title(r'$\psi$ vs. Time')
    axs[2, 0].plot(t, phi, 'g', linewidth=1.5)
    axs[2, 0].set_xlabel('Time (s)')
    axs[2, 0].set_ylabel(r'$\phi$ (rad)')
    axs[2, 0].set_title(r'$\phi$ vs. Time')
    axs[0, 1].plot(t, dot_theta, 'b', linewidth=1.5)
    axs[0, 1].set_xlabel('Time (s)')
    axs[0, 1].set_ylabel(r'$\dot{\theta}$ (rad/s)')
    axs[0, 1].set_title(r'$\dot{\theta}$ vs. Time')
    axs[1, 1].plot(t, dot_psi, 'r', linewidth=1.5)
    axs[1, 1].set_xlabel('Time (s)')
    axs[1, 1].set_ylabel(r'$\dot{\psi}$ (rad/s)')
    axs[1, 1].set_title(r'$\dot{\psi}$ vs. Time')
    axs[2, 1].plot(t, dot_phi, 'g', linewidth=1.5)
    axs[2, 1].set_xlabel('Time (s)')
    axs[2, 1].set_ylabel(r'$\dot{\phi}$ (rad/s)')
    axs[2, 1].set_title(r'$\dot{\phi}$ vs. Time')
    plt.tight_layout()
    plt.show()

def penduloinvertido_f(t, x):
    # Unpack the state variables
    theta, psi, phi, theta_dot, psi_dot, phi_dot = x

    # Parameters of the system
    g = 9.8
    m = 0.03
    R = 0.04
    Jw = m * R ** 2 / 2
    M = 0.6
    W = 0.14
    D = 0.04
    H = 0.144

```

```

L = H / 2
Jphi = M * L ** 2 / 3
Jpsi = M * (W ** 2 + D ** 2) / 12
Rm = 6.69
Kt = 0.4
Ke = 0.4
fm = 0.0022
n = 1
# Matrices
M_inv = np.linalg.inv(np.array([
    [R ** 2 * (2 * m + M) + 2 * Jw, M * R * L * np.cos(psi), 0],
    [M * R * L * np.cos(psi), M * L ** 2 + Jpsi, 0],
    [0, 0, m * (W ** 2 / 2) + Jphi + (W ** 2 / (2 * R ** 2)) * Jw]
]))
G = np.array([0, -M * g * L * np.sin(psi), 0])
C = np.array([
    -M * R * L * np.sin(psi) * psi_dot ** 2,
    -M * L ** 2 * np.sin(psi) * np.cos(psi) * phi_dot ** 2,
    M * L ** 2 * np.sin(2 * psi) * psi_dot * phi_dot
])
Q = np.array([
    [-2 * ((n * Kt * Ke / Rm) + fm), 2 * ((n * Kt * Ke / Rm) + fm), 0],
    [2 * ((n * Kt * Ke / Rm) + fm), -2 * ((n * Kt * Ke / Rm) + fm), 0],
    [0, 0, -W ** 2 / (2 * R ** 2) * ((n * Kt * Ke / Rm) + fm)]
])
F = Q @ np.array([theta_dot, psi_dot, phi_dot])
# Input vector
V = np.array([0, 0]) # [V_l, V_r]
# System of differential equations
dxdt = np.zeros_like(x)
dxdt[0:3] = [theta_dot, psi_dot, phi_dot]
dxdt[3:6] = M_inv @ (F - G - C)
return dxdt

# Call the simulation function to run the simulation and plot the results
pendulum_simulation()

```

C Código: Simulação do sistema linearizado

Listing 4: Código da simulação do sistema com ode45

```

1  function pendulum_simulation()
2      % Initial conditions
3      x0 = [0; 0.18; 0; 0; 0; 0]; % [theta; psi; phi; dot{theta}; dot{psi}; dot{phi}]
4      % Time span for simulation
5      tspan = 0:0.01:0.4; % From 0 to 10 seconds with a time step of 0.01 seconds
6      % Solve the differential equations using ode45 solver
7      [t, x] = ode45(@penduloinvertido_f, tspan, x0);
8      % Extract the states for plotting
9      theta = x(:, 1);
10     psi = x(:, 2);
11     phi = x(:, 3);
12     dot_theta = x(:, 4);
13     dot_psi = x(:, 5);
14     dot_phi = x(:, 6);
15     % Plot the results
16     figure;
17     subplot(3, 2, 1);
18     plot(t, theta, 'b', 'LineWidth', 1.5);
19     xlabel('Time (s)');
20     ylabel('\theta (rad)');
21     title('Pendulum Motion: \theta vs. Time');
22     subplot(3, 2, 3);
23     plot(t, psi, 'r', 'LineWidth', 1.5);
24     xlabel('Time (s)');
25     ylabel('\psi (rad)');
26     title('Pendulum Motion: \psi vs. Time');
27     subplot(3, 2, 5);
28     plot(t, phi, 'g', 'LineWidth', 1.5);
29     xlabel('Time (s)');
30     ylabel('\phi (rad)');
31     title('Pendulum Motion: \phi vs. Time');
32     subplot(3, 2, 2);
33     plot(t, dot_theta, 'b', 'LineWidth', 1.5);
34     xlabel('Time (s)');
35     ylabel('\dot{\theta} (rad/s)');
36     title('Pendulum Motion: \dot{\theta} vs. Time');
37     subplot(3, 2, 4);
38     plot(t, dot_psi, 'r', 'LineWidth', 1.5);
39     xlabel('Time (s)');
40     ylabel('\dot{\psi} (rad/s)');
41     title('Pendulum Motion: \dot{\psi} vs. Time');
42     subplot(3, 2, 6);

```

```

43     plot(t, dot_phi, 'g', 'LineWidth', 1.5);
44     xlabel('Time (s)');
45     ylabel('\dot{\phi} (rad/s)');
46     title('Pendulum Motion: \dot{\phi} vs. Time');
47 end
48 function dx = penduloinvertido_f(t, x)
49     % Variaveis de estado
50     theta = x(1);
51     psi = x(2);
52     phi = x(3);
53     % Derivadas das variaveis de estado
54     theta_dot = x(4);
55     psi_dot = x(5);
56     phi_dot = x(6);
57     % Parametros do sistema
58     g = 9.8; % gravidade
59     m = 0.03; % peso da roda
60     R = 0.04; % raio da roda
61     Jw = m * R^2 / 2; % momento de inercia da roda
62     M = 0.6; % peso do corpo
63     W = 0.14; % largura do corpo
64     D = 0.04; % profundidade do corpo
65     H = 0.144; % altura do corpo
66     L = H / 2; % distancia do centro de massa do corpo ao eixo da roda
67     Jphi = M * L^2 / 3; % momento de inercia do corpo em pitch
68     Jpsi = M * (W^2 + D^2) / 12; % momento de inercia do corpo em rumo
69     Rm = 6.69; % resistencia do motor DC
70     Kt = 0.4; % constante de torque (Nm/A)
71     Ke = 0.4; % e constante de EMF (V s/rad)
72     fm = 0.0022; % atrito entre o corpo e o motor
73     n = 1; % reducao do motor
74     % Matriz de inercia inversa
75     M_inv = inv([
76         R^2 * (2 * m + M) + 2 * Jw, M * R * L, 0;
77         M * R * L, M * L^2 + Jpsi, 0;
78         0, 0, m * (W^2 / 2) + M * L * sin(psi)^2 + Jphi + Jphi + (W^2 / (2 * R^2)) * Jw
79     ]);
80     % Matriz G
81     G = [
82         0;
83         -M * g * L * psi;
84         0
85     ];
86     % Matriz C
87     C = [
88         0;

```

```

89         0;
90         0
91     ];
92     % Matriz Q
93     Q = [
94         -2 * ((n * Kt * Ke / Rm) + fm), 2 * ((n * Kt * Ke / Rm) + fm), 0;
95         2 * ((n * Kt * Ke / Rm) + fm), -2 * ((n * Kt * Ke / Rm) + fm), 0;
96         0, 0, -W^2 / (2 * R^2) * ((n * Kt * Ke / Rm) + fm)
97     ];
98     F = Q*[theta_dot; psi_dot; phi_dot]
99     % Vetor de entrada
100    V = [0; 0]; % [V_l; V_r]
101    % Sistema de equacoes diferenciais
102    dx = zeros(6, 1);
103    dx(1:3) = [theta_dot; psi_dot; phi_dot];
104    dx(4:6) = M_inv * (F - G - C);
105 end
106 pendulum_simulation();

```

D Código: Simulação do sistema com controle realimentado

Listing 5: Código da simulação do sistema com ode45

```

1 function pendulum_simulation()
2     % Initial conditions
3     x0 = [0; 0.18; 0; 0; 0; 0]; % [theta; psi; phi; dot{theta}; dot{psi}; dot{phi}]
4     % Time span for simulation
5     tspan = 0:0.01:10; % From 0 to 10 seconds with a time step of 0.01 seconds
6     % Solve the differential equations using ode45 solver
7     [t, x] = ode45(@penduloinvertido_f, tspan, x0);
8     % Extract the states for plotting
9     theta = x(:, 1);
10    psi = x(:, 2);
11    phi = x(:, 3);
12    dot_theta = x(:, 4);
13    dot_psi = x(:, 5);
14    dot_phi = x(:, 6);
15    % Plot the results
16    figure;
17    subplot(3, 2, 1);
18    plot(t, theta, 'b', 'LineWidth', 1.5);
19    xlabel('Time (s)');
20    ylabel('\theta (rad)');
21    title('Pendulum Motion: \theta vs. Time');
22    subplot(3, 2, 3);

```



```

23     plot(t, psi, 'r', 'LineWidth', 1.5);
24     xlabel('Time (s)');
25     ylabel('\psi (rad)');
26     title('Pendulum Motion: \psi vs. Time');
27     subplot(3, 2, 5);
28     plot(t, phi, 'g', 'LineWidth', 1.5);
29     xlabel('Time (s)');
30     ylabel('\phi (rad)');
31     title('Pendulum Motion: \phi vs. Time');
32     subplot(3, 2, 2);
33     plot(t, dot_theta, 'b', 'LineWidth', 1.5);
34     xlabel('Time (s)');
35     ylabel('\dot{\theta} (rad/s)');
36     title('Pendulum Motion: \dot{\theta} vs. Time');
37     subplot(3, 2, 4);
38     plot(t, dot_psi, 'r', 'LineWidth', 1.5);
39     xlabel('Time (s)');
40     ylabel('\dot{\psi} (rad/s)');
41     title('Pendulum Motion: \dot{\psi} vs. Time');
42     subplot(3, 2, 6);
43     plot(t, dot_phi, 'g', 'LineWidth', 1.5);
44     xlabel('Time (s)');
45     ylabel('\dot{\phi} (rad/s)');
46     title('Pendulum Motion: \dot{\phi} vs. Time');
47 end
48 function dx = penduloinvertido_f(t, x)
49     % Variaveis de estado
50     theta = x(1);
51     psi = x(2);
52     phi = x(3);
53     % Derivadas das variaveis de estado
54     theta_dot = x(4);
55     psi_dot = x(5);
56     phi_dot = x(6);
57     % Parametros do sistema
58     g = 9.8; % gravidade
59     m = 0.03; % peso da roda
60     R = 0.04; % raio da roda
61     Jw = m * R^2 / 2; % momento de inercia da roda
62     M = 0.6; % peso do corpo
63     W = 0.14; % largura do corpo
64     D = 0.04; % profundidade do corpo
65     H = 0.144; % altura do corpo
66     L = H / 2; % distancia do centro de massa do corpo ao eixo da roda
67     Jphi = M * L^2 / 3; % momento de inercia do corpo em pitch
68     Jpsi = M * (W^2 + D^2) / 12; % momento de inercia do corpo em rumo

```

```

69 Rm = 6.69; % resistencia do motor DC
70 Kt = 0.4; % constante de torque (Nm/A)
71 Ke = 0.4; % e constante de EMF (V s/rad)
72 fm = 0.0022; % atrito entre o corpo e o motor
73 n = 1; % reducao do motor
74 % Matriz de inercia inversa
75 M_inv = inv([
76     R^2 * (2 * m + M) + 2 * Jw, M * R * L * cos(psi), 0;
77     M * R * L * cos(psi), M * L^2 + Jpsi, 0;
78     0, 0, m * (W^2 / 2) + Jphi + (W^2 / (2 * R^2)) * Jw
79 ]);
80 % Matriz G
81 G = [
82     0;
83     -M * g * L * sin(psi);
84     0
85 ];
86 % Matriz C
87 C = [
88     -M * R * L * sin(psi) * psi_dot^2;
89     -M * L^2 * sin(psi) * cos(psi) * phi_dot^2;
90     M * L^2 * sin(2 * psi) * psi_dot * phi_dot
91 ];
92 % Matriz Q
93 Q = [
94     -2 * ((n * Kt * Ke / Rm) + fm), 2 * ((n * Kt * Ke / Rm) + fm), 0;
95     2 * ((n * Kt * Ke / Rm) + fm), -2 * ((n * Kt * Ke / Rm) + fm), 0;
96     0, 0, -W^2 / (2 * R^2) * ((n * Kt * Ke / Rm) + fm)
97 ];
98 % Matriz H
99 H = [
100     n * Kt / Rm, n * Kt / Rm;
101     -n * Kt / Rm, -n * Kt / Rm;
102     -W / (2 * R) * n * Kt / Rm, W / (2 * R) * n * Kt / Rm
103 ];
104 % Matriz K
105 K = [-0.0224, -25.4867, -0.7071, -1.0362, -2.2530, -0.0076;
106     -0.0224, -25.4867, 0.7071, -1.0362, -2.2530, 0.0076];
107 F = Q*[theta_dot; psi_dot; phi_dot] - H * K * [theta;psi;phi;theta_dot;psi_dot;phi_dot];
108 % Sistema de equacoes diferenciais
109 dx = zeros(6, 1);
110 dx(1:3) = [theta_dot; psi_dot; phi_dot];
111 dx(4:6) = M_inv * (F - G - C);
112 end
113 pendulum_simulation();

```