Relatório

Gabriel Renato

Julho 2020

1 Dinâmica física do sistema

Equações matemáticas que serão utilizadas como base para simular o comportamento do bicoptero.

$$a\ddot{\theta}_4(t) + b\dot{\theta}_4(t) + c\theta_4(t) = L_1 f_1(t) - L_2 f_2(t) \tag{1}$$

$$\ddot{f}_i(t) + p\dot{f}_i(t) + qf_i(t) = r(pwm_i(t)), \quad i = 1, 2 \tag{2}$$

As constantes presentes nas equações (1) e (2) são definidas na tabela 1.

Constante	Valor
a - Momento de inercia $(kg \cdot m^2)$	0.0187
b - Coeficiente de atrito dinâmico $(kg \cdot m^2/s)$	0.0495
c - Desalinhamento $(C_g - CIR)$ $(kg \cdot m^2/s^2)$	0.6925
L_1 - Comprimento do braço esquerdo (m)	0.225
L_2 - Comprimento do braço direito (m)	0.215
p - Coeficiente determinado experimentalmente	5.348
q - Coeficiente determinado experimentalmente	13.43
r - Coeficiente determinado experimentalmente	1.319

Table 1: Constantes presentes nas equações (1) e (2)

2 Controladores

Nessa seção serão descritos alguns controladores projetados para controlar a dinâmica modelada nas equações (1) e (2).

2.1 LQR

2.1.1 Modelo Matemático simplificado adotado para projeto

$$f_1(t) = -f_2(t) = f(t) \tag{3}$$

$$a\ddot{\theta}_4(t) + b\dot{\theta}_4(t) + c\theta_4(t) = (L_1 + L_2)f(t) \tag{4}$$

$$pwm(t) = \frac{q}{r}f(t) \tag{5}$$

Pode-se notar que foi considerada uma única entrada que aplicada positivamente no motor 1 e negativamente no motor 2.

2.1.2 Modelo no espaço de estados a tempo continuo

- Estados(n):
  - $x_1 = \theta_4(t)$: Posição angular
  - $x_2 = \dot{\theta}_4(t)$: Velocidade angular
- Variáveis manipuladas(p):
  - $u_1 = f(t)$: força de empuxo
- Variáveis controladas(q):
  - $y_1 = x_1$

$$
\begin{aligned}
\dot{\mathbf{x}}(t) &= \underbrace{\begin{bmatrix} 0 & 1 \\ -c/a & -b/a \end{bmatrix}}_{A} \mathbf{x}(t) + \underbrace{\begin{bmatrix} 0 \\ (L_1 + L_2)/a \end{bmatrix}}_{B} \mathbf{u}(t) \\
\mathbf{y}(t) &= \underbrace{\begin{bmatrix} 1 & 0 \end{bmatrix}}_{C} \mathbf{x}(t)
\end{aligned}
\tag{6}
$$

Onde:
$A \in R^{n\times n}$ - Matriz dos estados;
$B \in R^{n\times p}$ - Matriz das entradas;
$C \in R^{q\times n}$ - Matriz das saidas.

2.1.3 Modelo no espaço de estados a tempo discreto

$$
\begin{aligned}
\mathbf{x}(k+1) &= \Phi \mathbf{x}(k) + \Gamma \mathbf{u}(k) \\
\mathbf{y}(k) &= C\mathbf{x}(k)
\end{aligned}
\tag{7}
$$

Obtido discretizando (6) para um período de amostragem de Ts segundos.

2.1.4 Lei de controle

Adota-se um controle por realimentação de estados, isto é,

$$\mathbf{u}(k) = -K\mathbf{x}(k) \tag{8}$$

A matriz de ganhos (K) é obtida minimizando a seguinte função de custo:

$$\mathbf{J} = \sum_{k=0}^{\infty} (\mathbf{x}^T(k) Q \mathbf{x}(k) + \mathbf{u}^T(k) R \mathbf{u}(k)) \tag{9}$$

Em que:
$Q \in \mathbb{R}_*^{n\times n}$ - Matriz de peso simétrica dos estados ($Q = diag(\mu_1, \mu_2, ..., \mu_n)$);
$R \in \mathbb{R}_*^{p\times p}$ - Matriz de peso simétrica das entradas ($R = diag(\rho_1, \rho_2, ..., \rho_p)$).

Esse problema de otimização não está no escopo deste relatório. Para isso utilizaremos a função **dlqr** do MATLAB pois quando passados como parâmetros as matrizes Φ, Γ, Q e R retorna a resposta para esse problema, que é a matriz de ganhos (K).

Como os sinais de entrada da dinâmica real da planta são dois sinais PWM utilizamos as equações (3) e (5) que descrevem uma relação simplificada entre força e PWM para converter o sinal, onde a verdadeira ação de controle aplicada é,

$$\mathbf{w}(k) = \begin{bmatrix} \frac{q}{r}\mathbf{u}(k) & -\frac{q}{r}\mathbf{u}(k) \end{bmatrix} \tag{10}$$

2.2 Alocação dinâmica dos polos de malha fechada

Os modelos adotados para projeto são os mesmos descritos para o LQR de 2.1.1 a 2.1.3

2.2.1 Lei de controle

Adota-se um controle por realimentação de estados, isto é,

$$\mathbf{u}(k) = -K\mathbf{x}(k) \tag{11}$$

A matriz de ganhos (K) é obtida de forma que os polos de MF sejam iguais aos desejados, isto é,

$$|zI_n - (\phi - \Gamma K)| = \underbrace{z^n + \alpha_1 z^{n-1} + \alpha_2 z^{n-2} + ... + \alpha_n}_{p.c.} = 0 \tag{12}$$

Sendo que o polinômio característico (p.c.) é obtido a partir da localização dos polos desejados de MF que por sua vez são escolhidos com base dos requisitos de projeto (Overshoot, settling time, rise time).
A matriz de ganhos (K) é obtida rapidamente utilizando a função **acker** do MATLAB pois quando passados como parâmetros as matrizes Φ, Γ e os polos desejados, retorna a matriz de ganhos (K) que satisfaz a equação (12).
A verdadeira ação de controle aplicada na planta é obtida conforme a equação (10).

2.3 Compensador LAG

2.3.1 Modelo Matemático simplificado adotado para projeto

Modelo descrito pelas equações 1 e 2, porem com a adição da seguinte simplificação:

$$pwm_1(t) = -pwm_2(t) = u(t) \tag{13}$$

2.3.2 Função de transferência a tempo continuo

O compensador foi projetado com base na resposta em frequência da função de transferência a tempo contínuo do modelo. A função de transferência é obtida aplicando transformada de Laplace as equações (1), (2) e (13) para condições iniciais nulas.

$$\Theta(s) = \underbrace{\frac{1}{as^2 + bs + c}}_{G(s)}(L_1F_1(s) - L_2F_2(s)) \tag{14}$$

$$F_i(s) = \underbrace{\frac{r}{s^2 + ps + q}}_{H(s)}(PWM_i(s)), \quad i = 1, 2 \tag{15}$$

$$PWM_1(s) = U(s), \quad PWM_2(s) = -U(s) \tag{16}$$

Substituindo adequadamente as equações (14) e (15) em (14) obtém-se a função de transferência de malha aberta de toda a planta.

$$\frac{\Theta(s)}{U(s)} = G(s)(L_1 + L_2)H(s) \tag{17}$$

2.3.3 Resposta em frequência

Diagrama de bode da função de transferência descrita em (17).

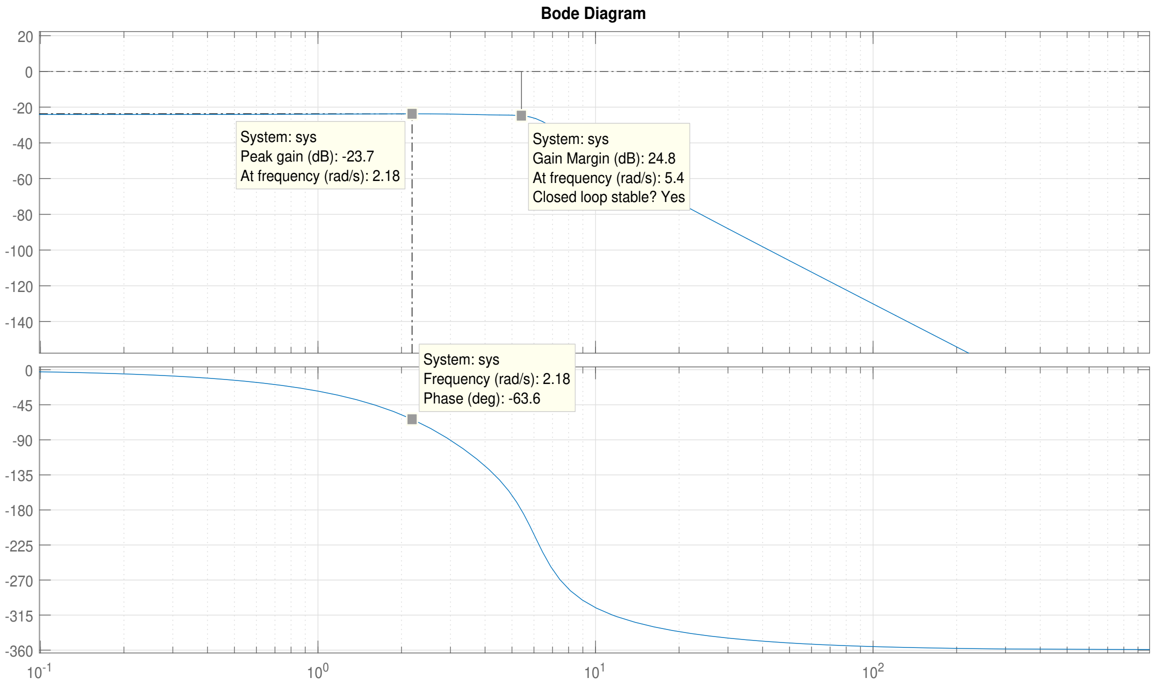

Figure 1: Diagrama de bode do sistema

Observa-se que o sistema possui uma constante de erro de posição de $20log(K_p) = -23.7dB$.

Para analisar se a emulação do compensador é possível a seguinte razão deve ser satisfeita.

$$\frac{w_s}{w_c} \geq 30 \tag{18}$$

Onde

w_s−Frequência de amostragem ($w_s = 2\pi/T_s$)

w_c−Frequência de corte

A frequência de amostragem é obtida para um período de amostragem (T_s) de 0.02 segundos. Logo,

$$\frac{w_s}{w_c} = \frac{2\pi}{0.02 \cdot w_c} \geq 30$$
$$w_c \leq 10.47rad/s$$

Ou seja, o controlador discreto obtido por emulação pode ser empregado seguramente desde que a frequência de corte seja menor que 10.47 rad/s para um período de amostragem de 0.02 segundos.

2.3.4 Lei de controle

A função de transferência do compensador LAG é,

$$C_{LAG}(s) = \frac{s+a}{s+b}, \quad a > b \tag{19}$$

Com base na resposta em frequência da figura 1, projetou-se um compensador do tipo LAG de forma que o erro de regime seja menor que 5%, ou seja, a constante de erro de posição deve ser maior que 19 ($25dB$). Com base nisso optou-se que o zero do LAG fosse pelo menos mil vezes maior que o polo isto é,

$$C_{LAG}(0) = \frac{a}{b} = 1000$$

Definindo arbitrariamente, a=7, tem-se que

$$\frac{7}{b} = 1000 \rightarrow b = 0.007$$

Posteriormente ajustou-se o ganho do sistema para obter uma margem de fase (PM) de aproximadamente 60deg ($\xi = 0.6$) que corresponde a um overshoot de aproximadamente 10%.

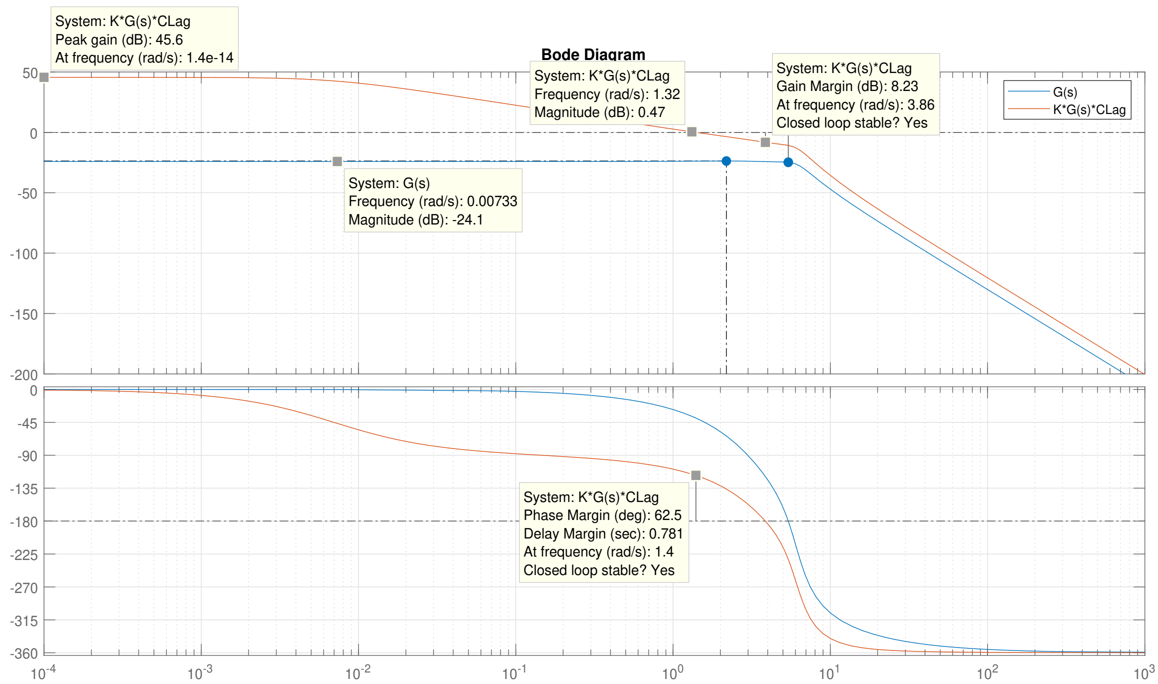

Figure 2: Diagrama de bode do sistema com o compensador

Ou seja o compensador final a tempo continuo é dado por:

$$C_{LAG}(s) = 3.0456\frac{s+7}{s+0.007} \tag{20}$$

A frequência de corte do sistema com compensador é 1.4 [rad/s] portanto a emulação é possível e o compensador final a tempo discreto emulado via Tustin é dado por:

$$C_{LAG}(z) = \frac{3.259z - 2.832}{z - 0.9999} \tag{21}$$

A lei de controle pode ser reescrita em uma equação a diferenças obtida aplicando transformada Z inversa no compensador discreto (21), o que resulta em,

$$\begin{aligned} \mathbf{u}(k) &= 0.9999 \cdot \mathbf{u}(k-1) + 3.258 \cdot \mathbf{e}(k) - 2.832 \cdot \mathbf{e}(k-1) \\ \mathbf{e}(k) &= referencia - \theta_4(k) \end{aligned} \tag{22}$$

Onde a verdadeira ação de controle aplicada é,

$$\mathbf{w}(k) = \begin{bmatrix} \mathbf{u}(k) & -\mathbf{u}(k) \end{bmatrix} \tag{23}$$

3 Simulações e sugestões para implementação

Todas as simulações foram feitas considerando ruido de medida de posição com dispersão de $\pm 0.025 rad$ e ruido de medida de velocidade com dispersão de $\pm 0.14 rad/s$, além disso também adotou-se um atraso de transporte de 0.01 segundos.

3.1 LQR

Para os parâmetros,

$$\begin{aligned} Q &= diag([1, 2]) \\ R &= 100 \end{aligned}$$

Obtém-se uma matriz de ganhos de,

$$K = \begin{bmatrix} -0.0222 \\ 0.0659 \end{bmatrix}$$

3.1.1 Implementação

Para introduzir referencia (r) adote,

$$xss = \begin{bmatrix} -0.0222 \\ 0.0659 \end{bmatrix} r, \quad uss = 1.5739 \cdot r$$

A implementação da lei de controle em c++

$$\begin{aligned} & uk = -K[0] \cdot (x1 - xss[0]) - K[1] \cdot (x2 - xss[1]) + uss; \\ & wk = \frac{13.43}{1.319} \cdot uk; //forca - pwm \\ & //saturacão \\ & wk = min(wLim[1], wk); \\ & wk = max(wLim[2], wk); \end{aligned}$$

Em que $x1$ e $x2$ são respectivamente, posição angular e velocidade angular de θ_4 lidos pelo sensor X e wk e $-wk$ são respectivamente os PWM's aplicados nos motores 1 e 2.

3.1.2 Resultados

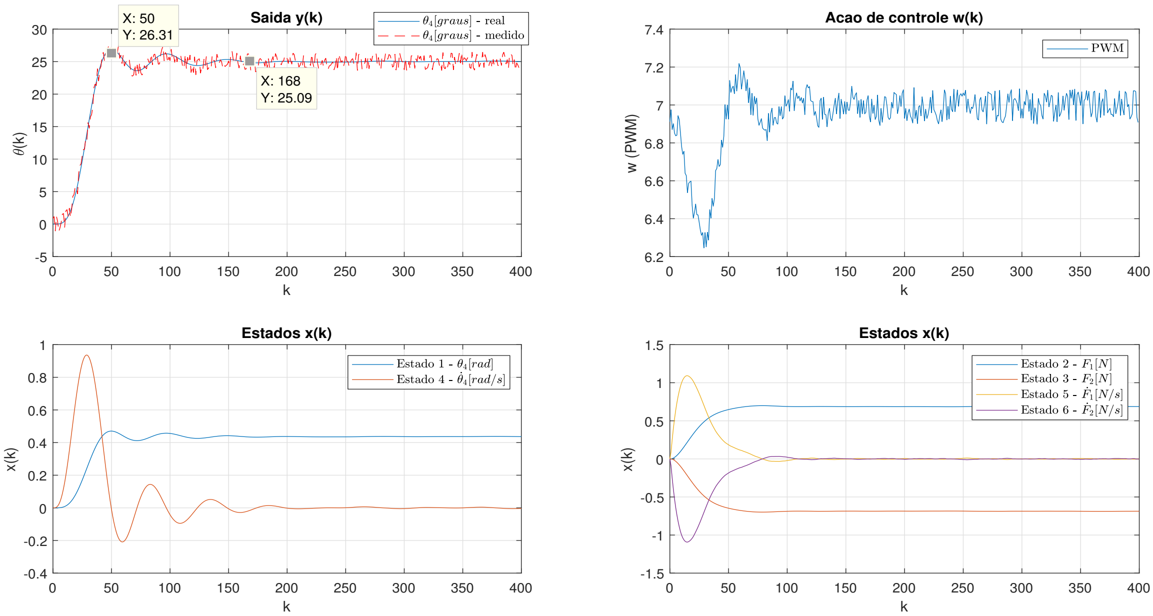

Figure 3: Resultado para referência em 25°.

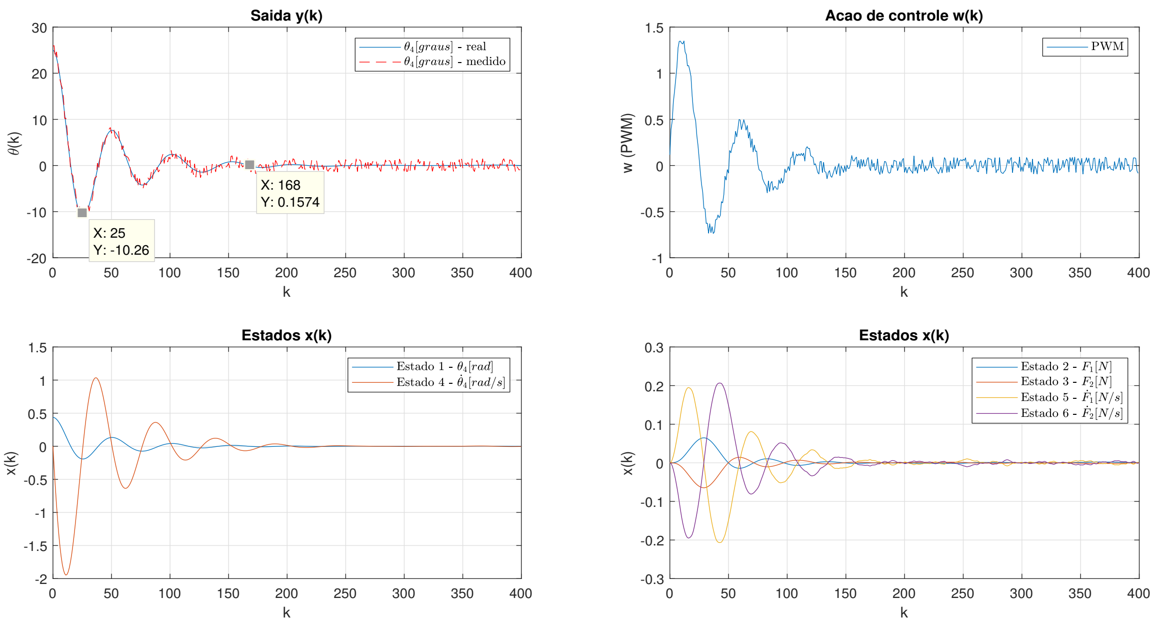

Figure 4: Resultado para referência em 0° e condição inicial de 25°.

3.2 Alocação de polos em MF

Para os parâmetros,

$$polos = [0.92 \quad 0.93]$$

Obtém-se uma matriz de ganhos de,

$$K = \begin{bmatrix} -0.9622 \\ 0.1935 \end{bmatrix}$$

3.2.1 Implementação

Para introduzir referencia (r) adote,

$$xss = \begin{bmatrix} -0.0222 \\ 0.0659 \end{bmatrix} r, \quad uss = 1.5739 \cdot r$$

A implementação da lei de controle em c++

$$\begin{aligned}
&uk = -K[0] \cdot (x1 - xss[0]) - K[1] \cdot (x2 - xss[1]) + uss; \\
&wk = \frac{13.43}{1.319} \cdot uk; //forca - pwm \\
&//saturac\tilde{a}o \\
&wk = min(wLim[1], wk); \\
&wk = max(wLim[2], wk);
\end{aligned}$$

Em que $x1$ e $x2$ são respectivamente, posição angular e velocidade angular de θ_4 lidos pelo sensor X e wk e $-wk$ são respectivamente os PWM's aplicados nos motores 1 e 2.

3.2.2 Resultados

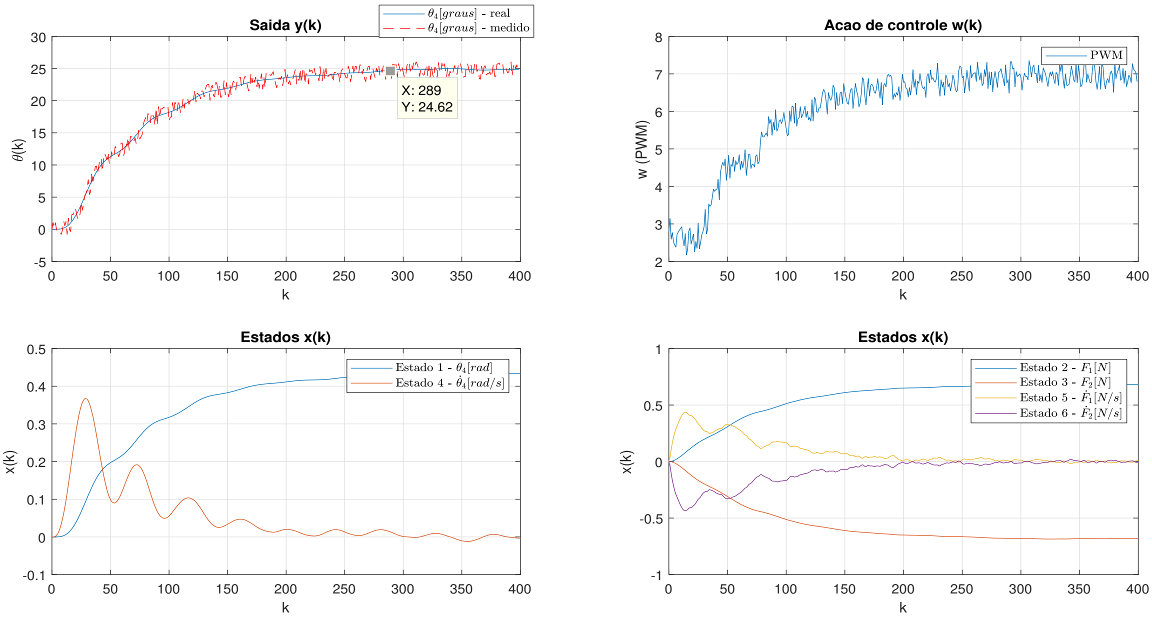

Figure 5: Resultado para referencia em 25°.

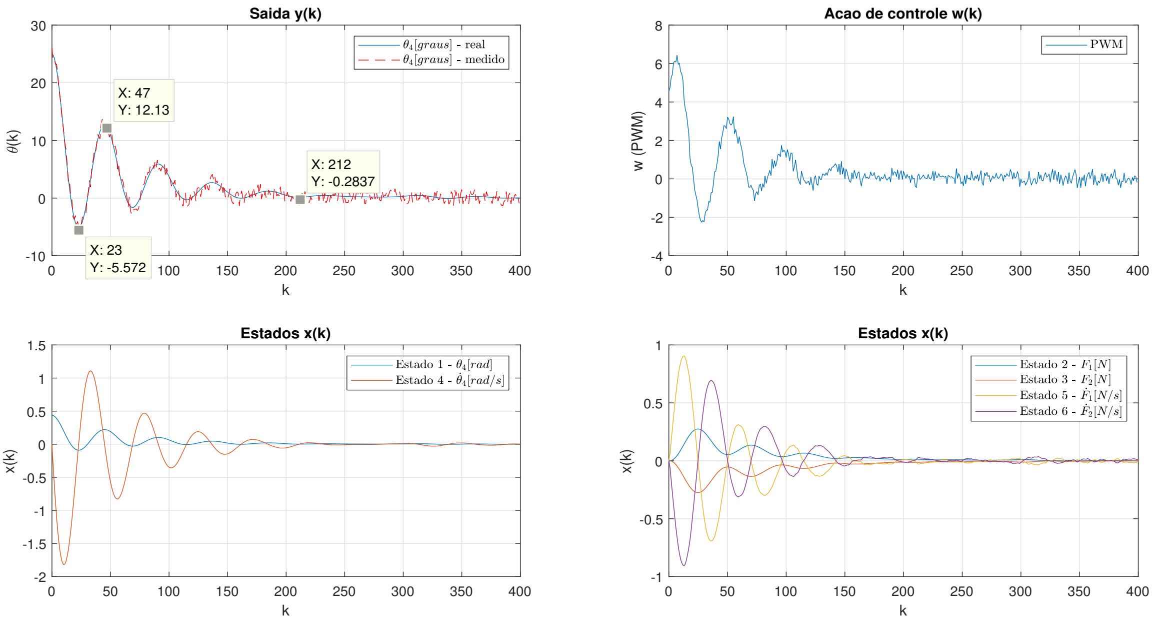

Figure 6: Resultado para referencia em 0° e condição inicial em 25°.

3.3 Compensador LAG

3.3.1 Implementação

A implementação da lei de controle em c++

$$
\begin{aligned}
&ekm1 = ek;\\
&ek = r - x1;\\
&wkm1 = wk;\\
&wk = 0.9999 \cdot wkm1 + 3.259 \cdot ek - 2.832 \cdot ekm1;\\
&//saturação\\
&wk = min(wLim[1], wk);\\
&wk = max(wLim[2], wk);
\end{aligned}
$$

Em que $x1$ é a posição angular de θ_4 lido pelo sensor X e wk e $-wk$ são respectivamente os PWM's aplicados nos motores 1 e 2. Note que nessa estrategia de controle não é necessária a conversão de força para PWM.

3.3.2 Resultados

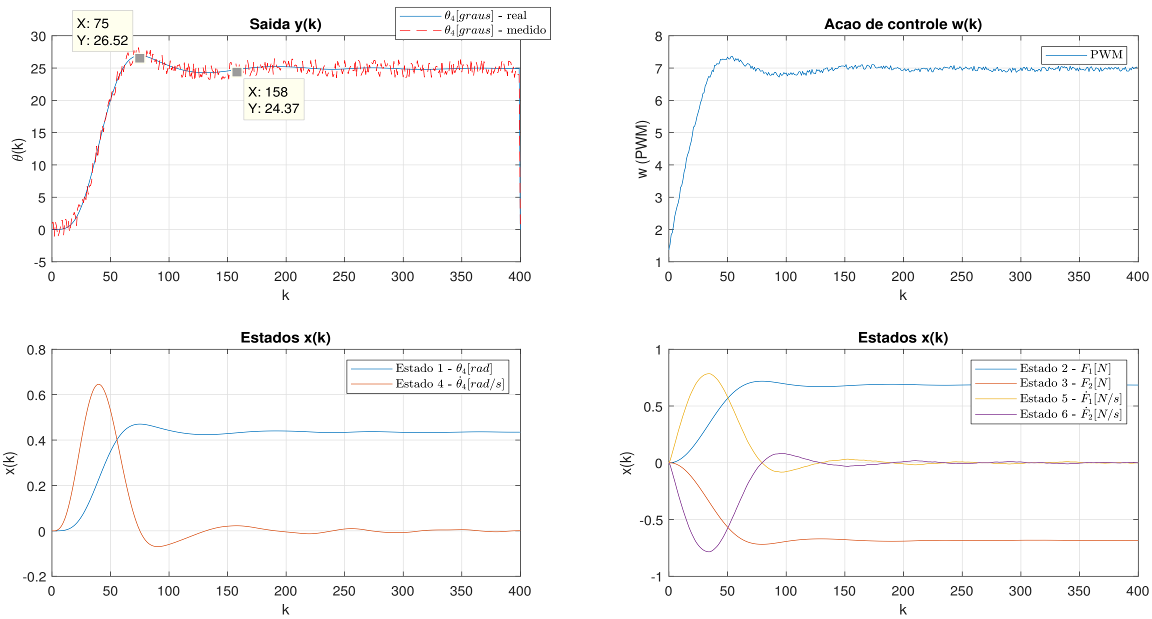

Figure 7: Resultado para referencia em 25°.

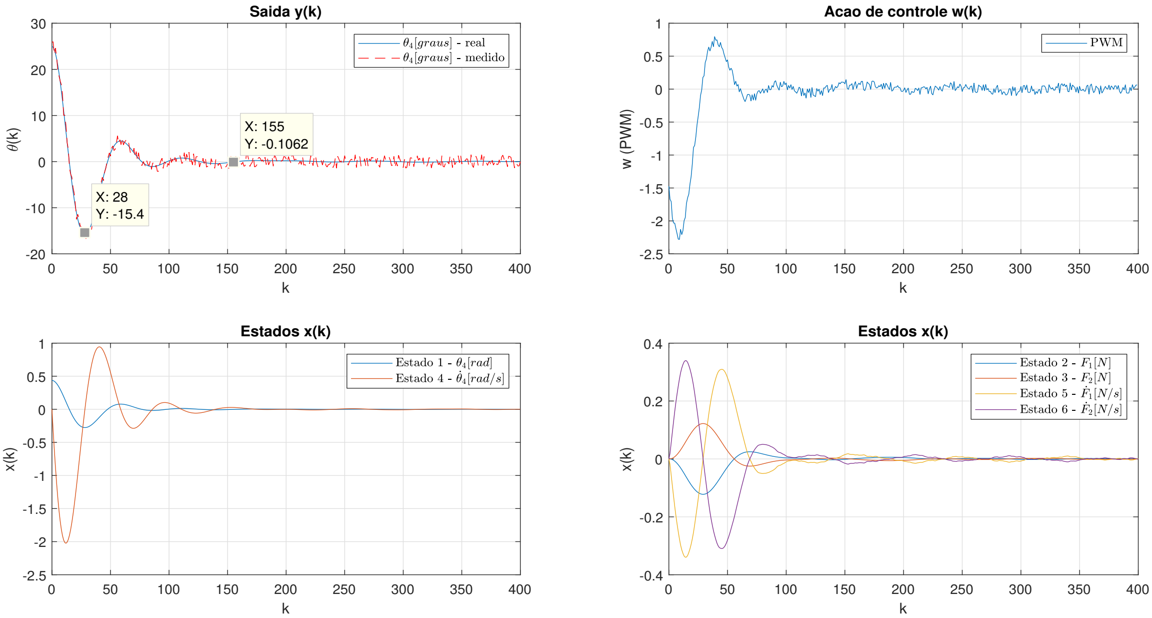

Figure 8: Resultado para referencia em 0° e condição inicial em 25°