FTL066–Programação de Sistemas de Tempo Real
Laboratório 4 - Simulação de Sistemas 2 (Non-RT)

Prof. André Cavalcante
andrecavalcante@ufam.edu.br

Agosto de 2023

Um robô móvel, com acionamento diferencial, pode ser descrito pelo modelo no espaço de estados (1).

$$\begin{aligned} \dot{x}(t) &= \begin{bmatrix} \cos(x_3) & 0 \\ \sin(x_3) & 0 \\ 0 & 1 \end{bmatrix} u(t) \\ y(t) &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} x(t) + \begin{bmatrix} R\cos(x_3) \\ R\sin(x_3) \end{bmatrix} \end{aligned} \tag{1}$$

onde $x(t) = [x_1 x_2 x_3]^T = [x_c \ y_c \ \theta]^T$, sendo $(x_c, \ y_c)$ a posição do centro de massa do robô e θ a sua orientação. $u(t) = [v \, \omega]^T$ é a entrada do sistema, sendo v a velocidade linear e ω a velocidade angular do robô. A saída do sistema é $y(t)$, correspondendo a frente do robô cujo diâmetro $D = 2R = 0.6m$.

Para este sistema, tem-se:

$$\begin{aligned} \dot{y}(t) &= \begin{bmatrix} \cos(x_3)u_1 \\ \sin(x_3)u_1 \end{bmatrix} + \begin{bmatrix} -R\sin(x_3)\dot{x_3} \\ R\cos(x_3)\dot{x_3} \end{bmatrix} \\ &= \begin{bmatrix} \cos(x_3) & -R\sin(x_3) \\ \sin(x_3) & R\cos(x_3) \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \\ &= L(x)u(t) \end{aligned}$$

e portanto, como $L(x)$ é não singular, o sistema pode ser linearizado por realimentação fazendo-se

$$u = L^{-1}(x)v(t) \tag{2}$$

sendo $v(t)$ a nova entrada do sistema linearizado e desacoplado dado por

$$\dot{y}(t) = v(t) \tag{3}$$

Para controlar o sistema (3) será utilizado um controlador por modelo de refe-rência, com $v(t)$ dada por

$$v(t) = \begin{bmatrix} \dot{y}_{mx} + \alpha_1(y_{mx} - y1) \\ \dot{y}_{my} + \alpha_2(y_{my} - y2) \end{bmatrix} \quad (4)$$

sendo y_{mx} e y_{my} as saídas dos modelos de referência para a dinâmica do robô nas direções X e Y, respectivamente, dados por

$$G_{mx}(s) = \frac{\alpha_1}{s + \alpha_1}$$
$$G_{my}(s) = \frac{\alpha_2}{s + \alpha_2}$$

ou ainda, no domínio do tempo

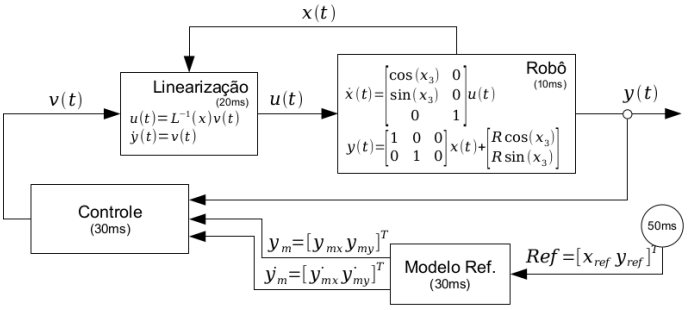
$$\dot{y}_{mx} = \alpha_1(x_{ref} - y_{mx})$$
$$\dot{y}_{my} = \alpha_2(y_{ref} - y_{my})$$

onde α_1 e α_2 são definidos empiricamente para melhor controle do robô. Use, como ponto de partida: $\alpha_1 = \alpha_2 = 3$.

A referência é dada por

$$x_{ref}(t) = \frac{5}{\pi}\cos(0.2\pi t)$$
$$y_{ref}(t) = \begin{cases} \frac{5}{\pi}\sin(0.2\pi t) & \text{, para } 0 <= t < 10 \\ -\frac{5}{\pi}\sin(0.2\pi t) & \text{, para } t >= 10 \end{cases}$$

Resumindo:



FAZER: um programa de simulação do robô em movimento.

1 Observações

1. O programa deve ser dividido em várias *threads*:
   (a) Simulação do robô em si, recebendo u e gerando x e y, com período de 30ms.
   (b) Malha de linearização por realimentação, gerando u a partir de x e v com período de 40ms.

(c) Malha de controle, gerando v a partir de y e das saídas dos modelos de referência y_{mx} e y_{my}, com período de 50ms.

(d) Simulação do modelo de referência na direção X, gerando y_{mx} a partir de x_{ref}, com período de 50ms.

(e) Simulação do modelo de referência na direção Y, gerando y_{my} a partir de y_{ref}, com período de 50ms.

(f) Geração da referência, x_{ref} e y_{ref}, com período de 120ms.

(g) Interface com o usuário e a gravação dos dados no disco.

2. As *threads* devem executar como tarefas normais do Linux.

3. A interface com o usuário deve apresentar na tela os valores do tempo, posição e orientação do robô e da valores da referência. Deve também permitir a troca dos valores de α_1 e α_2.

4. Devem ser construídos monitores para acesso as variáveis compartilhadas.

5. Faça o gráfico de $y(k)$ amostrado e x_{ref} e y_{ref} para um horizonte de simulação de 20s.

6. Faça uma tabela comparando os valores de média, variância, desvio padrão, valores máximos e mínimos dos tempos de computação e do respectivo *jitter* para o sistema sem carga e com carga para as diversas tarefas do sistema.

7. Faça uma análise formal da escalonabilidade do sistema.

8. O relatório deverá descrever a hierarquia de diretórios utilizada para estruturar o programa e explicar porque esta hierarquia facilita a reutilização do código gerado. Além disso, deverá apresentar uma análise crítica dos resultados dos experimentos.

2 Entrega

- Entrega conforme a data no Google Classroom.
- Fazer *updload* no Google Classroom de:
  - Relatório
    Deverá descrever a hierarquia de diretórios utilizada para estruturar o programa e explicar porque esta hierarquia facilita a reutilização do código gerado. Além disso, deverá apresentar uma análise crítica dos resultados dos experimentos.
  - A pasta de desenvolvimento compactada
    OBS.: o professor irá baixar em sua máquina a pasta, descompactá-la, fazer um `make` e executar.
- Não serão aceitos trabalhos iguais.