

# CMC-12 - Relatório Lab 2 - Robô *Line Follower*

Aluno: Bruno Costa Alves Freire

17 de Maio de 2020

## 1 Projeto do Controlador Frontal

Para controlar a posição horizontal do robô, assumimos uma linearização do modelo e implementamos um controlador em malha fechada descrito pela equação  $\dot{x} + K_x x = K_x x_r$ , cujo comportamento será dado por

$$x(t) = x_r (1 - e^{-K_x t}).$$

Ajustamos o ganho  $K_x$  de modo a obter uma constante de tempo  $\tau = 0,15$  s. Simulando o controlador com condições iniciais nulas para uma posição de referência  $x_r = 1$  m, obtemos a curva do gráfico da figura 1, onde podemos ver que a simulação possui o comportamento esperado pela teoria.

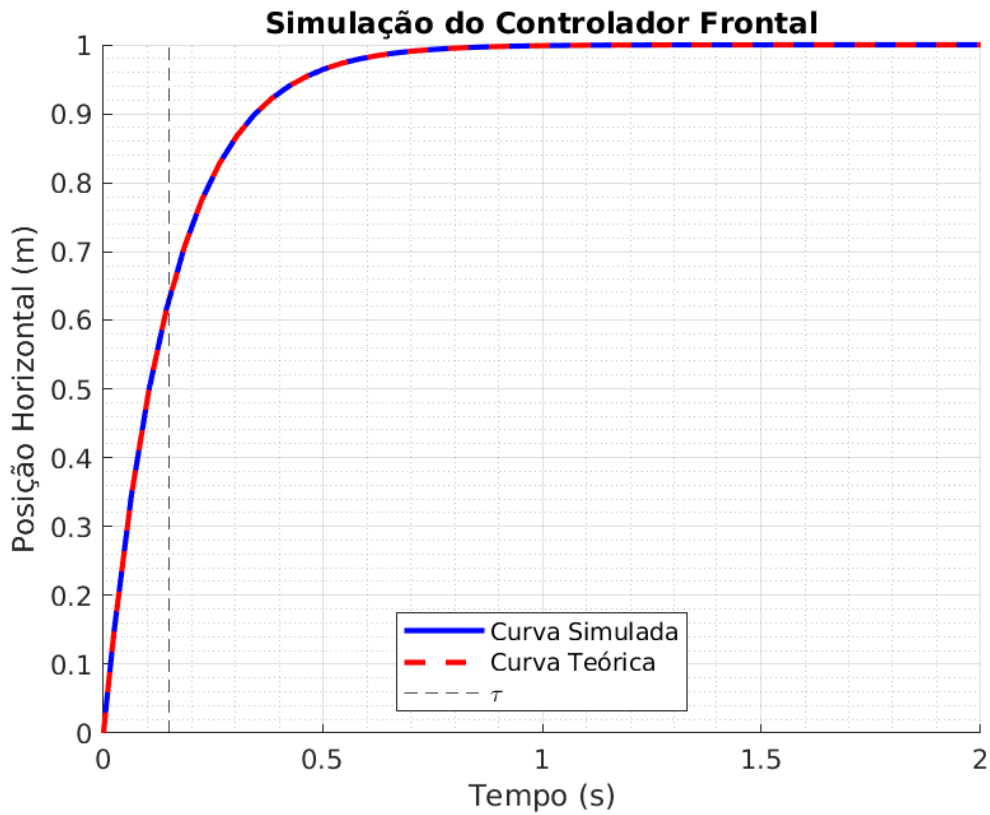


Figura 1: Simulação do controlador frontal vs. comportamento esperado pela teoria.

## 2 Projeto do Controlador Lateral

Para controlar o desvio vertical do robô em relação a uma linha de referência, implementamos o controlador lateral. Por se tratar de uma dinâmica não-linear, assumimos um modelo linearizado onde  $\sin \psi \approx \psi$ . Para comparar o funcionamento do modelo linearizado frente a uma estratégia de controle com malhas aninhadas para ângulo e posição, com o comportamento da dinâmica mais realista, na qual não assumimos a aproximação de  $\sin \psi$  e aplicamos saturação do ângulo de referência na malha de controle, plotamos os gráficos das figuras 2, 3 e 4.

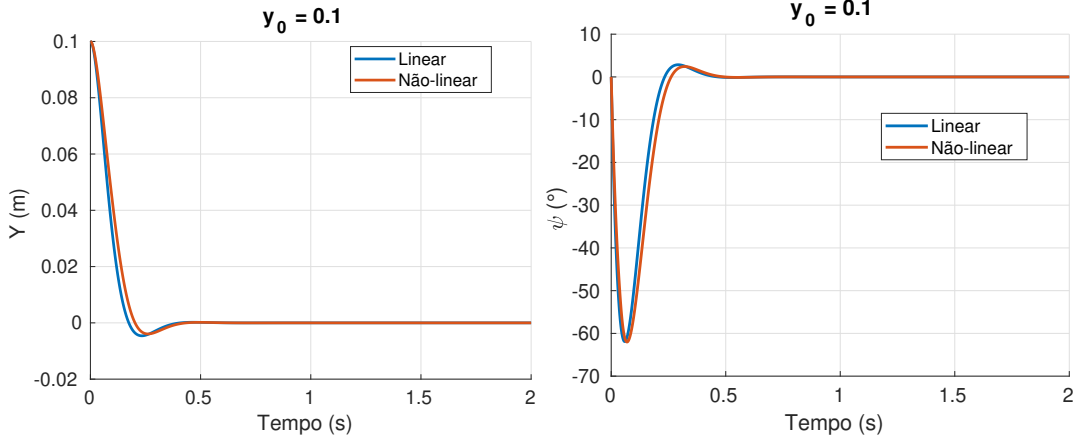


Figura 2: Simulação do controlador lateral do *line follower* com desvio inicial  $y_0 = 0.1$  m.

No modelo linearizado temos a dinâmica  $\ddot{y} + K_\psi \dot{y} + K_y K_\psi v y = K_y K_\psi v y_r$ , onde os ganhos foram escolhidos de modo que o sistema imitasse a dinâmica de  $\ddot{y} + 2\xi\omega_n \dot{y} + \omega_n^2 y = \omega_n^2 y_r$ , com  $\xi = 0.7$  e  $\omega_n = 6\pi$  rad/s. Definindo a altura de referência  $y_r = 0$ , temos uma equação homogênea, de modo que devemos esperar soluções do tipo

$$\begin{cases} \lambda = \omega_n(-\xi \pm i\sqrt{1-\xi^2}) \\ \dot{y}(0) = 0 \end{cases} \implies y(t) = y_0 e^{-\xi\omega_n t} \cos(\omega_d t) + \frac{\xi}{\sqrt{1-\xi^2}} e^{-\xi\omega_n t} \sin(\omega_d t),$$

onde  $\omega_d = \omega_n \sqrt{1-\xi^2}$ . Nas figuras 2 a 4 podemos observar comportamento condizente com esse nas curvas azuis, típicas de sistemas com amortecimento subcrítico. Na figura 2 notamos uma pequena disparidade do comportamento não-linear, que aparenta ser ligeiramente mais lento. Essa diferença é devida à aproximação usada na linearização.

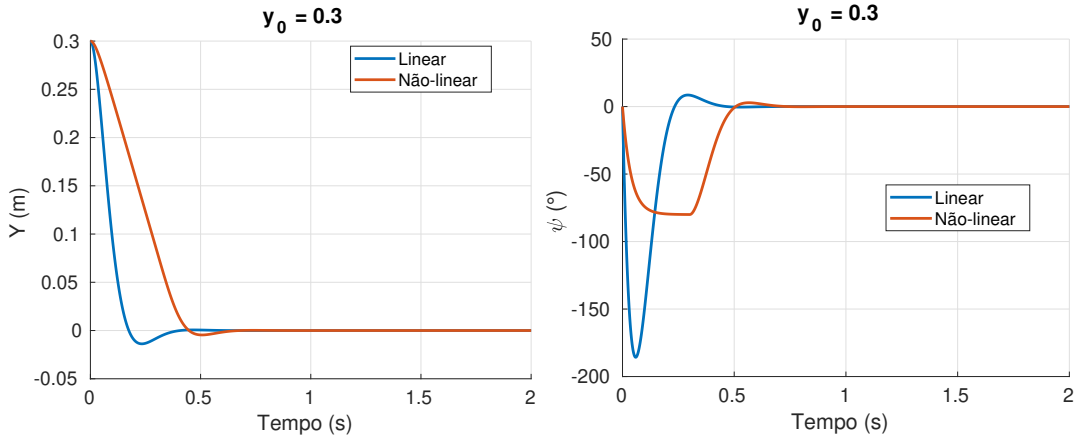


Figura 3: Simulação do controlador lateral do *line follower* com desvio inicial  $y_0 = 0.3$  m.

Na figura 3, já começamos a ver o mecanismo de saturação de ângulo de referência em ação, no gráfico direito, onde vemos que o ângulo  $\psi$  fica limitado próximo de  $-80^\circ$ . Essa saturação é necessária no modelo do controlador completo pois impede que o robô gire demais e acabe se descontrolando, mas aqui podemos notar que esta mesma saturação é responsável por retardar o robô em atingir a altura de referência.

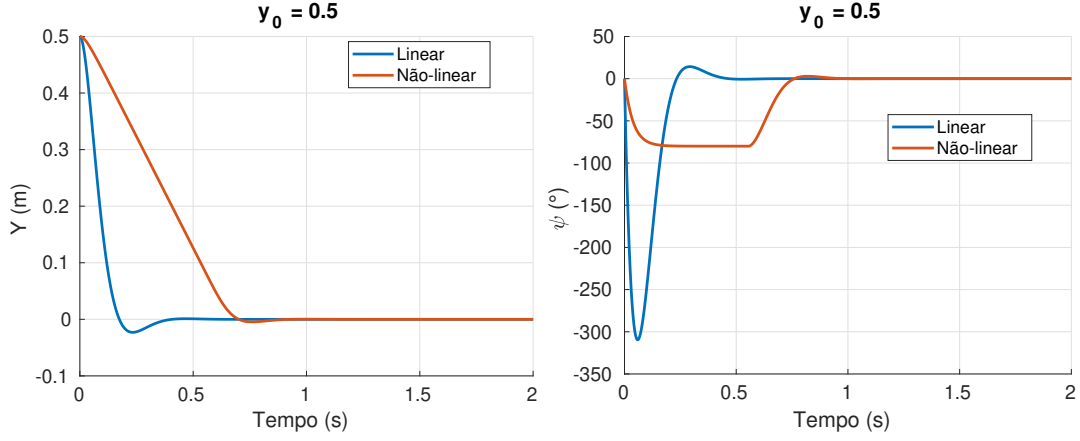


Figura 4: Simulação do controlador lateral do *line follower* com desvio inicial  $y_0 = 0.5$  m.

Por fim, na figura 4, vemos mais acentuado o efeito da saturação de ângulo na velocidade de resposta do sistema, em comparação com o modelo linearizado, uma vez que temos um desvio inicial maior da posição de referência. Outro ponto a ser observado é que, ao mesmo tempo em que reduz a velocidade de resposta do sistema, a saturação angular também reduz o *overshoot* da posição vertical.

### 3 Projeto do Controlador Completo

Finalmente, no projeto do controlador completo, iremos integrar as malhas de controle frontal e lateral. A dinâmica real (não-linear) será simulada com a estratégia de controle de ganho proporcional no eixo horizontal, e com realimentação de posição e de ângulo para o eixo vertical.

Além disso, para contornar problemas da dinâmica não-linear, implementamos a saturação do ângulo de referência, bem como da velocidade comunicada ao robô. Para lidar com o acoplamento do controlador frontal com o lateral, aplicamos um escalonamento de velocidade para atualizar o ganho em  $y$  dinamicamente.

O resultado da implementação do controlador completo pode ser visto nos gráficos da figura 5, onde temos as trajetórias do robô para várias linhas de referência, em que o robô deve progredir para frente e para trás. Podemos dizer que o controlador funcionou de acordo com o esperado.

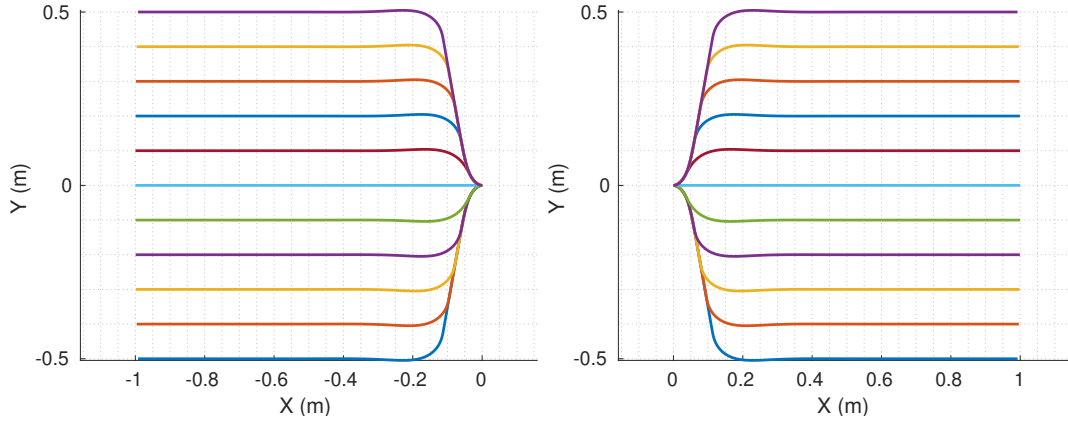


Figura 5: Simulação do controlador completo do *follow line* para diferentes retas de referência, com o robô indo para frente (direita) e para trás (esquerda).

Para observar mais detalhadamente como se dá o movimento do robô, foi gerado um vídeo simulando o robô da liga VSS percorrendo a trajetória para frente com  $y_r = 0.5$  m, no arquivo `followline.avi`.