



Instituto Tecnológico de Aeronáutica

Relatório da disciplina CM-202: Planejamento e Controle para Robótica Móvel

Laboratório 3: Projeto de Servomotor de Velocidade

Guilherme Müller Bertolino

1. Introdução

Nesse laboratório, foi projetado um controlador PI discreto para um servomotor de velocidade, um conjunto formado por um motor elétrico, um sensor de velocidade, uma caixa de redução e um controlador. O diagrama de blocos da planta do servomotor é apresentada na Figura 1.

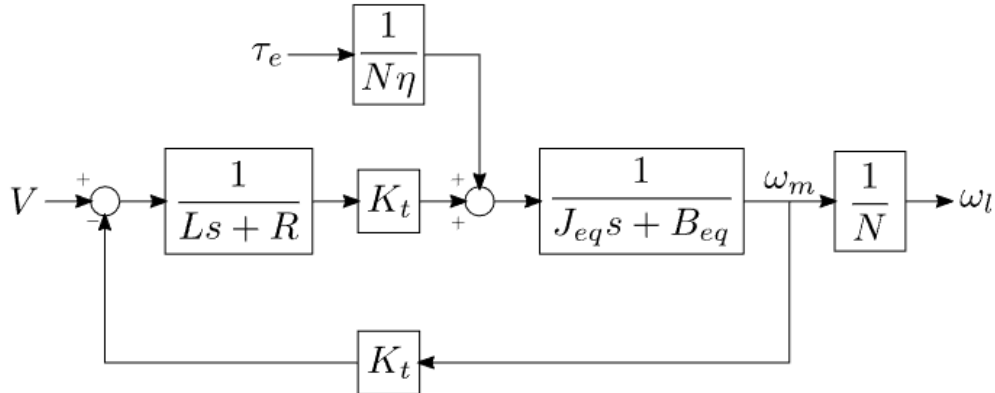


Figura 1: Diagrama de blocos da planta do servomotor

Onde V é a tensão aplicada entre os terminais do motor (entrada do sistema), ω_m é a velocidade de rotação do motor (saída do sistema), N é o fator de redução, J_{eq} é a inércia equivalente do sistema, B_{eq} é o coeficiente de atrito viscoso equivalente do sistema, K_t é a constante de torque, τ_e é um torque externo aplicado na carga (entrada de perturbação), L é a indutância do circuito do atuador e η é a eficiência de transmissão.

2. Metodologia

O primeiro passo para projetar o controlador foi identificar a função de transferência do diagrama de blocos da Figura 1, para isso, considerou-se que a dinâmica de corrente é muito mais rápida do que a dinâmica da mecânica, de modo que podemos desprezar o efeito do indutor ($L \approx 0$) e foi usada a fórmula de Mason:

$$G_m(s) = \frac{\Omega_m(s)}{V(s)} = \frac{(1/R) \cdot K_t \cdot 1/(J_{eq}s + B_{eq})}{1 + (1/R) \cdot K_t \cdot 1/(J_{eq}s + B_{eq})} =$$

$$= \frac{K_t}{J_{eq}Rs + B_{eq}R + K_t^2}$$

Para fazer o projeto em malha fechada do sistema, adicionou-se no diagrama de blocos apresentado a função de transferência do controlador PI e uma realimentação negativa da velocidade angular do motor, conforme o diagrama de blocos da Figura 2.

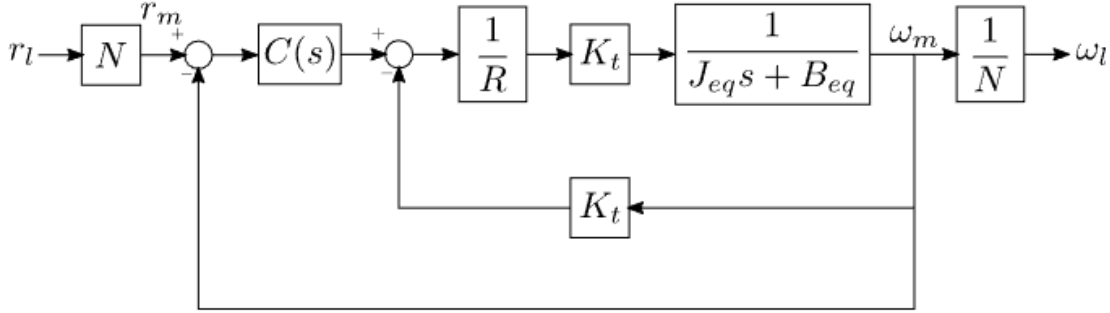


Figura 2: Diagrama de blocos do sistema em malha fechada

Neste diagrama de blocos, a função de transferência $C(s)$ é de um controlador PI: $C(s) = K_p + K_i/s$, para obter a função de transferência de malha fechada, foi feito o seguinte desenvolvimento:

$$G_f(s) = \frac{\Omega_m(s)}{R_m(s)} = \frac{C(s)G_m(s)}{1 + C(s)G_m(s)} =$$

$$= \frac{(K_p s + K_i)K_t/(J_{eq}R)}{s^2 + (B_{eq} + K_t^2 + K_p K_t)s/(J_{eq}R) + K_i K_t/(J_{eq}R)}$$

Como requisitos para o sistema, foram utilizados banda passante $\omega_b = 30Hz$ e margem de fase $PM = 50^\circ$. Para obter ω_n e ξ da função de transferência que geram os requisitos solicitados, pode-se usar as seguintes expressões:

$$\omega_b = \omega_n \sqrt{1 - 2\xi^2 + \sqrt{4\xi^4 - 4\xi^2 + 2}}$$

$$\xi = \frac{PM(^{\circ})}{100^{\circ}}$$

Dessa forma, desprezando o zero no numerador da função de transferência, pode-se encontrar os valores dos ganhos K_i e K_p pelas seguintes equações:

$$K_i = \omega_n^2 J_{eq} R / K_t \approx 0.0054$$

$$K_p = 2\xi \omega_n J_{eq} R / K_t - B_{eq} R / K_t - K_t \approx 1.065$$

Para refinar a implementação analítica proposta para o controlador, usa-se otimização com o algoritmo Nelder-Mead através da função `fminsearch` do MATLAB usando uma função de transferência mais próxima da real, considerando o atraso da discretização ($T/2$) e de leitura do encoder ($T/2$), conforme a Figura 3.

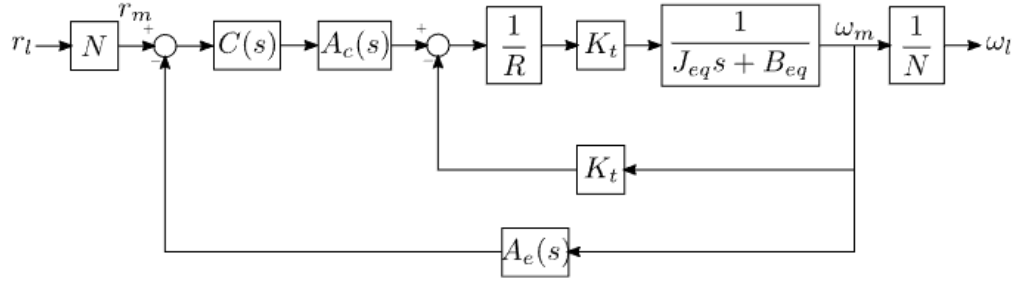


Figura 3: Diagrama de blocos do sistema considerando atrasos da discretização e do encoder

Os atrasos $A_c(s)$ e $A_e(s)$ foram modelados de acordo com a aproximação de Padé de segunda ordem:

$$A_c(s) = A_e(s) = e^{-sT/2} \approx \frac{s^2 - (6/\tau)s + 12/\tau^2}{s^2 + (6/\tau)s + 12/\tau^2}$$

O chute inicial para a otimização foram os ganhos obtidos analiticamente, e a função de custo usada foi a seguinte:

$$J(K_p, K_i) = (\omega_{b,req} - \omega_b(K_p, K_i))^2 + (PM_{req} - PM(K_p, K_i))^2$$

Depois da implementação, foram obtidos como ganhos $K_i \approx 0.2565$ e $K_p \approx 0.0039$. Para a implementação digital do controlador, foi usada a seguinte lei de controle discreta:

$$u[k] = u[k-1] + \left(K_p + \frac{K_i T}{2}\right) e[k] + \left(-K_p + \frac{K_i T}{2}\right) e[k-1]$$

Para reduzir efeitos de *windup*, causados pela saturação da tensão da bateria, foi implementado um *anti-windup*:

$$u[k] = \begin{cases} u[k], & -V_{\max} \leq u[k] < V_{\max}, \\ -V_{\max}, & u[k] \leq -V_{\max}, \\ V_{\max}, & u[k] > V_{\max}. \end{cases}$$

3. Resultados

Avaliando o controlador analítico sem o atraso causado por discretização e leitura do encoder, obteve-se banda passante de $37.46Hz$ e os diagramas de Bode e resposta a degrau unitário a seguir:

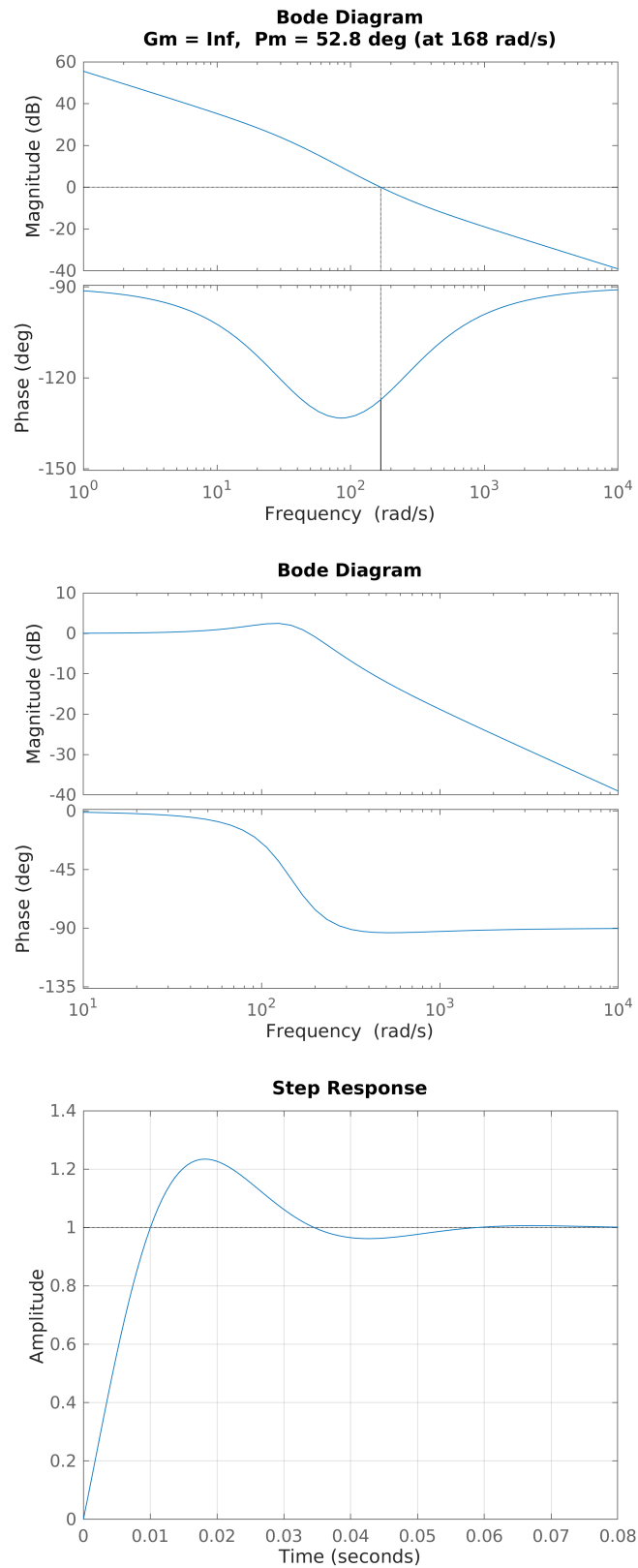


Figura 4: Margem de fase, diagrama de Bode e resposta a degrau obtidos no controlador analítico sem atrasos

Observe que no caso sem atrasos, o controlador atendeu razoavelmente bem os requisitos de

projeto, esses ganhos também foram testados para o caso com os atrasos, obtendo os gráficos a seguir:

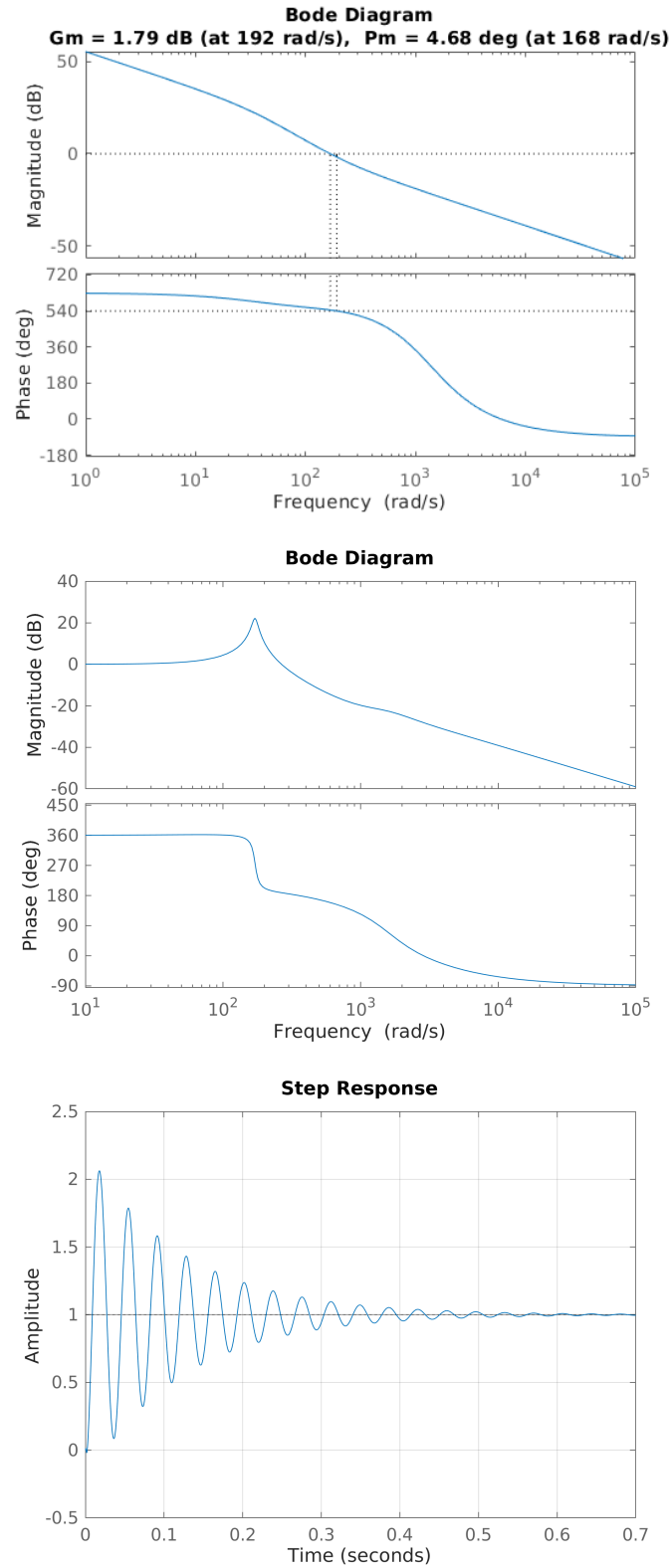


Figura 5: Margem de fase, diagrama de Bode e resposta a degrau obtidos no controlador analítico considerando atrasos

A banda passante obtida foi de $48.00Hz$, e a margem de fase de 4.68° , de modo que nesse caso, esses ganhos não atenderam os requisitos do projeto. Veja também que a resposta a degrau foi muito oscilatória e demorou muito para se estabilizar.

Fazendo os testes com o controlador otimizado, obteve-se banda passante de $29.56Hz$ e os diagramas de Bode e resposta a degrau a seguir:

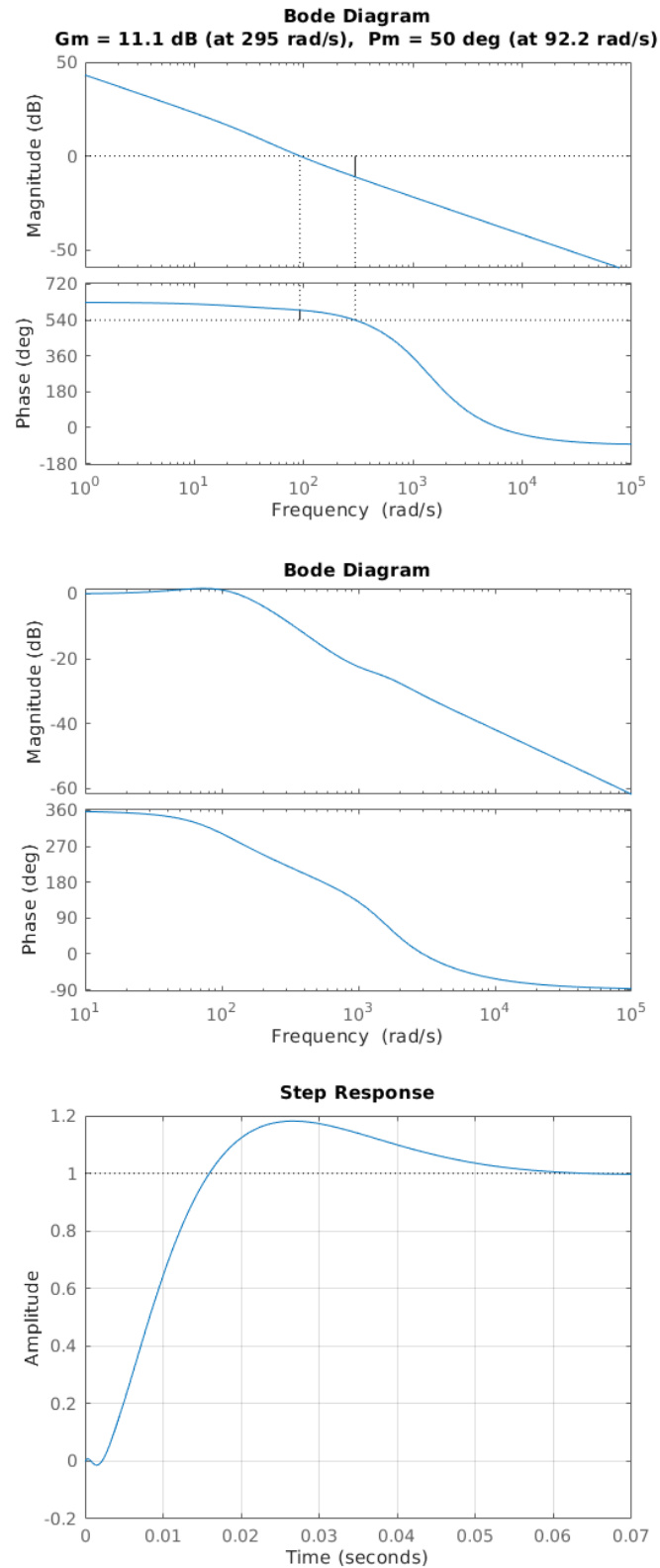


Figura 6: Margem de fase, diagrama de Bode e resposta a degrau obtidos no controlador otimizado considerando atrasos

Observe que nesse caso, o controlador atendeu perfeitamente aos requisitos de projeto.

Testando o controlador com o *anti-windup* implementado e fazendo testes para degraus de 40,

70 e 100rad/s , foram obtidos os gráficos das Figuras 7, 8, 9 e 10.

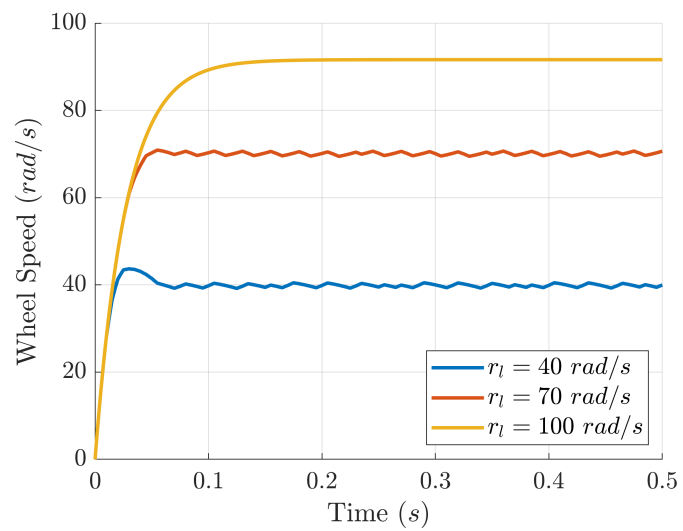


Figura 7: Velocidade de rotação

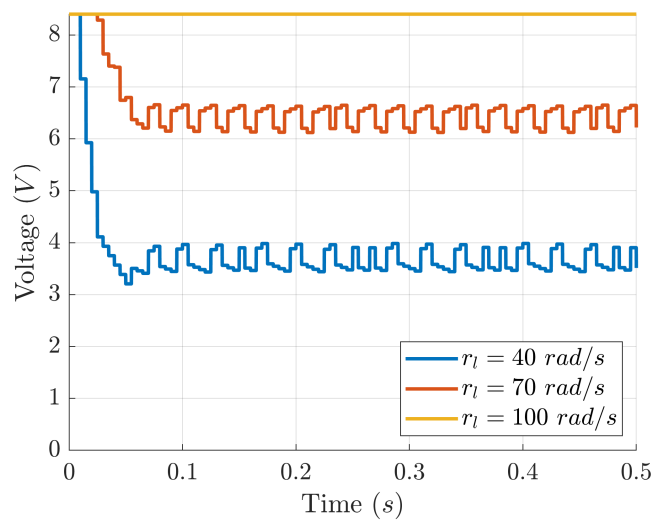


Figura 8: Tensão aplicada no servo

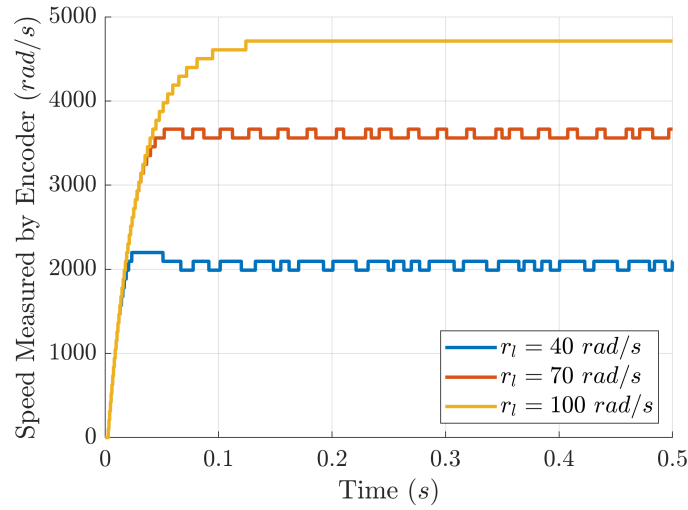


Figura 9: Leitura do encoder do motor

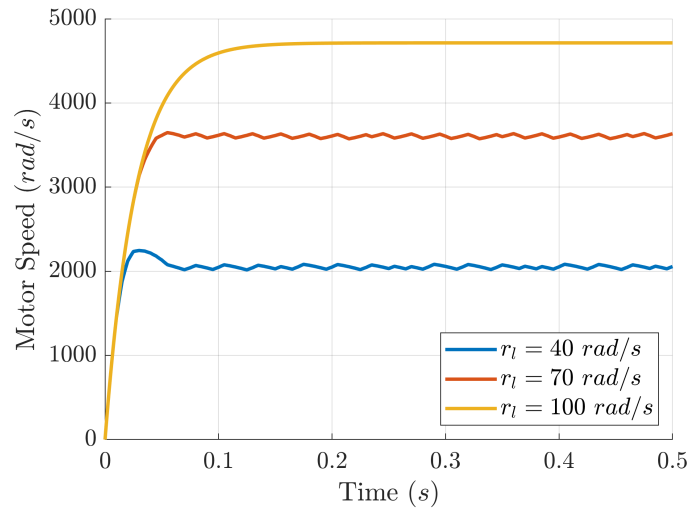


Figura 10: Velocidade do motor do servo

Veja que o servo seguiu bem a referência, tendo *overshoot* e tempos de estabilização aceitáveis. Observe que no caso de degrau de 100 rad/s , o esforço de controle (tensão da bateria) foi máximo o tempo todo, visto que a referência é maior do que o que o sistema consegue alcançar com a tensão de bateria disponível. Assim, ele simplesmente sobe a velocidade até aproximadamente 90 rad/s e se mantém lá até o fim do ensaio, nunca sendo capaz de atingir a referência.

Por fim, foi feito o teste do *anti-windup*, obtendo os resultados da Figura 11.

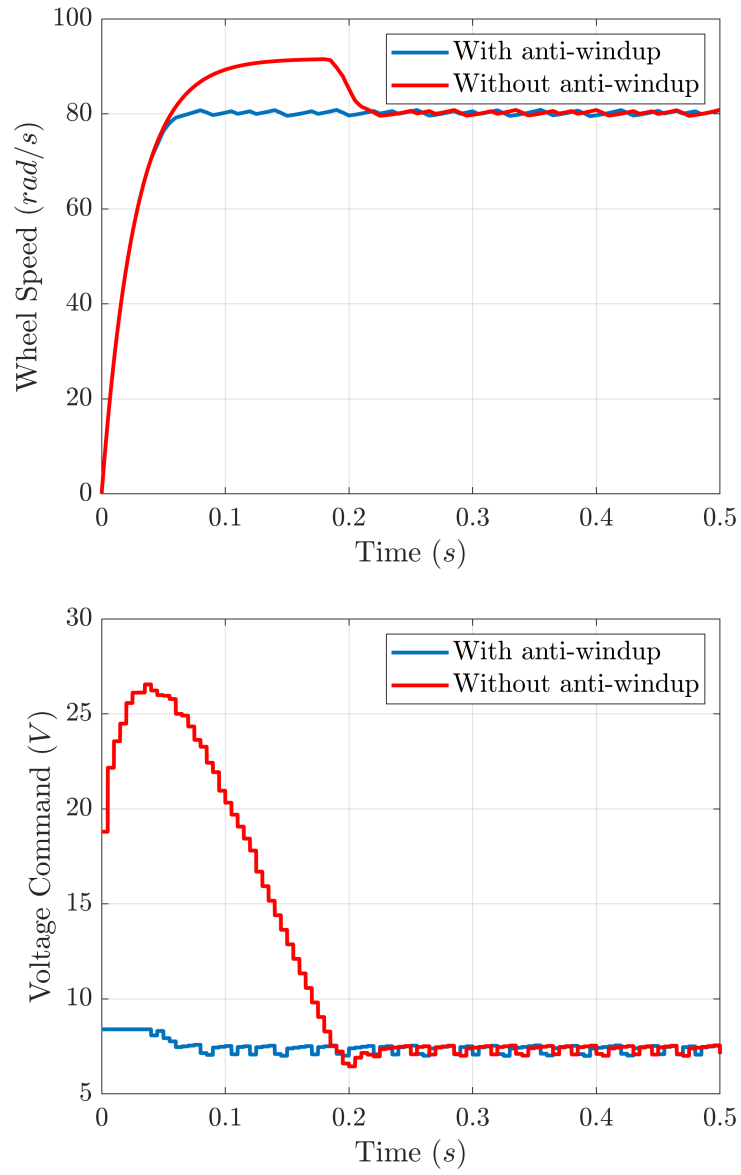


Figura 11: Avaliação do anti-windup

Veja que o *anti-windup* não deixa o controlador mandar uma tensão inatingível, o que reduz drasticamente o *overshoot*, pois não deixa o integrador acumular mais erro do que deveria.