

EA722
Laboratório de Princípios de Controle e
Servomecanismos

Universidade Estadual de Campinas - Faculdade de Engenharia Elétrica e de
Computação
Prof. Marconi Kolm Madrid

Experimento 6
Controle não Co-allocado

Equipamento utilizado: Sistema Torcional

Turma D

Participantes:

Nicolas Pereira da Silva (RA: 247298)

Pedro Nicolás Sampaio Gomes (RA: 247333)

Vinícius Esperança Mantovani (RA: 247395)

Motivação

Este experimento teve como motivação principal a busca por entender melhor o controle não co-allocado aplicado para o sistema físico torcional utilizado no laboratório. Desse modo, do mesmo modo que no experimento 5 buscou-se controlar o disco três do sistema torcional, aplicaremos as técnicas de controle não co-allocado, um sistema de controle mais “poderoso” para a situação, e analisaremos seus resultados para entender como a aplicação dessa técnica impacta o desempenho e funcionamento do controle da máquina. Além disso, visou-se a entender as diferenças entre os dois controles implementados e a diferença de resultados obtidos a partir de cada um deles.

Introdução

Pensando na configuração do sistema torcional considerando a aplicação da técnica de controle não co-allocado, foi desenvolvido um programa Matlab que permitisse encontrar (com auxílio de um operador) os valores que as constantes de ganho do sistema deveriam assumir para que o sistema fosse capaz de eliminar o efeito de perturbações externas no sistema torcional de modo relativamente rápido. Com isso, no decorrer dos procedimentos feitos, foram coletados dados para respostas do sistema ao degrau e à rampa, de maneira a possibilitar a análise dos resultados e, consequentemente, de se compreender o desempenho do controlador aqui implementado sobre o sistema proposto.

Itens Pré Procedimento Experimental

Projeto de realimentação do Disco 1 (disco de atuação):

Para esta etapa do laboratório 6, foi escrito um programa Matlab responsável por implementar as funções de transferência da planta, usar o lugar das raízes para determinar o valor de k_v para o qual o amortecimento do sistema é máximo e, finalmente, implementando o valor de k_v encontrado para encontrar os valores correspondentes dos coeficientes de ganho K_p e K_d e determinar os pólos da função de transferência $G^*(s)$. (Script enviado em conjunto com o relatório).

Como essa parte inicial do relatório propõe cálculos e procedimentos idênticos aos realizados no pré-relatório 6, está exposto abaixo o conteúdo que foi escrito para a resposta desses mesmos problemas no citado pré-relatório, com a adição de alguns gráficos e comentários específicos que foram solicitados pelo enunciado do experimento 6. Além disso, os scripts Matlab que foram escritos para a realização dos passos propostos serão enviados em conjunto com o relatório.

1.

Neste item, foi encontrada a função de transferência entre $\Theta_1(s)$ e $R^*(s)$ e a função usada para o lugar das raízes, sendo desenvolvidos os cálculos matemáticos de acordo com as informações presentes no enunciado e extraídas do diagrama do sistema presente na figura 10 do roteiro, conforme as imagens abaixo. Não foram utilizados os valores numéricos agora. Ao invés disso, esses valores foram determinados a partir de um programa Matlab, permitindo um desenvolvimento mais claro dos passos algébricos, permitindo com que o computador lidasse com a parte numérica. Por isso, também foi possível determinar a função de transferência interna $G^*(s)$, em função da variável k_v .

$$\Theta_1 = \frac{N_1}{D} k_{hw} (R^* - k_v \Theta_1 s)$$

$$\Theta_1 = \frac{N_1 k_{hw} R^*}{D + N_1 k_{hw} k_v s}$$

$$\Theta_2 = \frac{N_3}{N_1} \frac{N_1 k_{hw} R^*}{D + N_1 k_{hw} k_v s} = \frac{k_{eq} k_{hw} R^*}{D + k_{hw} k_v s N_1}$$

$$G_1^* = k_{hw} \frac{N_3}{D + k_{hw} k_v s N_1}$$

$$\Theta_1 = \frac{N_1}{D} k_{hw} (R^* - k_v s \Theta_1)$$

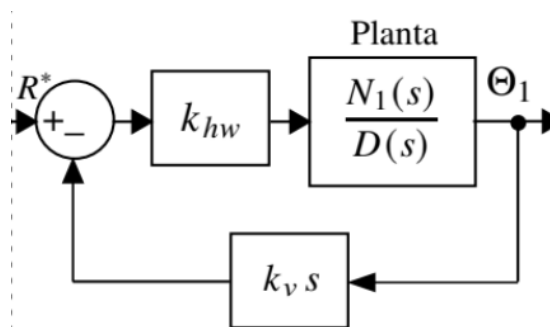
$$\rightarrow \frac{\Theta_1}{R^*} = \frac{\frac{N_1 k_{hw}}{D}}{1 + k_v s k_{hw} \frac{N_1}{D}} = \frac{N_1 k_{hw}}{D + k_v s k_{hw} N_1}$$

$$\rightarrow \frac{\Theta_1}{R^*} = \frac{(J_2 s^2 + c_2 s + k_{eq}) k_{hw}}{D + k_v k_{hw} (J_2 s^3 + c_2 s^2 + k_{eq} s)}$$

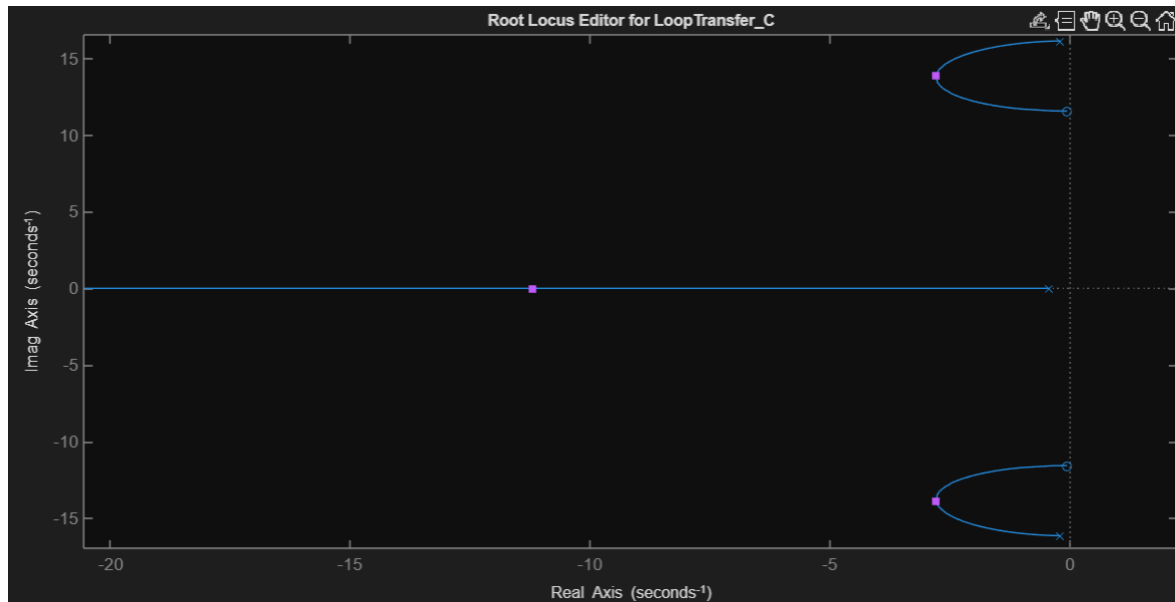
$$1 + k_v \frac{k_{hw} (J_2 s^3 + c_2 s^2 + k_{eq} s)}{D}$$

2.

A partir da utilização de ferramentas em ambiente Matlab, levando em consideração a função $\Theta_1(s)/R^*(s)$ do sistema que fora descrita no item anterior e com a utilização de uma chamada que nos permite analisar o lugar das raízes no nosso programa (rlocus), foi possível concluir que, para obter o maior amortecimento dos polos dessa função de transferência, devemos adotar o valor da variável de ganho $k_v = 0.00979$. A função $\Theta_1(s)/R^*(s)$ citada está relativa ao diagrama abaixo:



Além disso, vale detalhar que, para encontrar de modo mais preciso o valor do coeficiente k_v , utilizamos o artifício de controle mais acurado do gráfico de lugar das raízes a partir da chamada “controlSystemDesigner”, a qual recebeu como parâmetro a função de transferência de $\Theta_1(s)/R^*(s)$. A partir desse comando, foi possível movimentar de maneira mais livre os polos relativos à função de transferência em questão, permitindo posicioná-los, de modo mais rigoroso, nos pontos mais distantes do eixo imaginário, resultante em máximo amortecimento fornecido por esses polos. Veja abaixo o resultado da manipulação do gráfico de lugar das raízes que permitiu encontrar o valor de k_v adotado.



3.

Após o desenvolvimento da função de transferência $G^*(s)$, já aplicando o valor k_v encontrado no item anterior, foram obtidos os seguintes valores para os polos da função, obtidos a partir das raízes do denominador $D^*(s)$:

- $p_0 = 0.0$
- $p_1 = -2.79315449397212 + 13.8733630482637i$
- $p_2 = -2.79315449397212 - 13.8733630482637i$
- $p_3 = -11.2080142384252$

Observando os polos obtidos, vemos, claramente, que os polos complexos conjugados da função de transferência interna foram os denominados p_1 e p_2 . A partir de agora, sabemos quais polos considerar na hora de projetar o filtro Notch para ser aplicado ao sistema, conforme mostra o diagrama da figura 10 do roteiro, anteriormente citado.

Projeto do filtro notch:

Neste parte do procedimento, da mesma forma como na seção seguinte, não foram feitas alterações com relação ao que foi feito durante o pré-relatório deste experimento. Logo, falado anteriormente, seguem abaixo os itens respondidos para as mesmas questões dessa seção no PR6, cujos achados permaneceram inalterados até o desenvolvimento deste documento.

1.

Para construir um filtro Notch capaz de cancelar os dois polos de $G^*(s)$ que proporcionam menos amortecimento ao sistema a partir de dois zeros, tendo, além disso, dois polos complexos conjugados de frequências naturais 5 Hz e 11 Hz, com coeficiente de amortecimento $\sqrt{2}/2$, conforme orienta o enunciado, a estratégia foi separar o projeto do filtro em duas partes: o cálculo de seu numerador, que determinará os zeros, e o cálculo do denominador, que vai determinar seus polos.

Visando o projeto do seu denominador, que denotamos com $D_n(s)$, precisamos, conforme dito encontrar dois pares de polos conjugados, considerando o par de frequências $f_{n1} = 5$ Hz e $f_{n2} = 11$ Hz, com $\xi = \sqrt{2}/2$. Primeiramente, vamos transformar as frequências, mensuradas inicialmente em Hertz, para sua forma de frequência angular. Isto é, aplicamos as transformações $\omega_{n1} = f_{n1} * 2\pi = 10\pi$ rad/s e $\omega_{n2} = f_{n2} * 2\pi = 22\pi$ rad/s. Para gerar o denominador $D_n(s)$ desejado em si, vamos considerar que para gerar apenas um par de polos complexos conjugados, considerando frequência natural ω_n e coeficiente de amortecimento ξ , o denominador do filtro Notch será dado por $D(s) = s^2 + 2\xi\omega_n s + \omega_n^2$. Considerando isto e sabendo que precisamos de dois pares de polos, com frequências naturais de acordo com as encontradas anteriormente, então teremos que $D_{n1}(s) = s^2 + 2*(\sqrt{2}/2)*10\pi + (10\pi)^2$ e $D_{n2}(s) = s^2 + 2*(\sqrt{2}/2)*22\pi + (22\pi)^2$. A partir daí, basta fazermos $D_n(s) = D_{n1}(s) * D_{n2}(s)$. Isso nos dá, como resultado, $D_n(s) = s^4 + 32\pi s^3\sqrt{2} + 1024\pi^2 s^2 + 7040\pi^3 s\sqrt{2} + 48400\pi^4$. Com o coeficiente de maior grau de $D_n(s)$ sendo 1, basta adicionar o termo $48400\pi^4$ em $N_n(s)$ para que o ganho estático da função de transferência seja 1, ou seja $N_n(0)/D_n(0) = 1$.

Agora, para encontrar o numerador, denotado por $N_n(s)$, a lógica, conforme proposta pelo enunciado, foi construí-lo de forma a anular os dois polos de $G^*(s)$ com pequena função de amortecimento. Já descobrimos estes polos anteriormente, sendo estes o par conjugado p_1 e p_2 mostrados anteriormente, obtidos a partir das raízes do denominador $D^*(s)$ referente à função de transferência $G^*(s)$. Assim, para eliminá-los a partir da geração de zeros de mesmo valor a partir do numerador $N_n(s)$, temos que encarar que esses zeros precisam ter os mesmos componentes reais e imaginários que os polos complexos conjugados que temos originalmente. Para isso, vamos utilizar a propriedade do filtro Notch para a anulação desses polos, determinando que, para anular o efeito de dois polos, que podem ser expressos por $p_1 = r + j\omega$ e $p_2 = r - j\omega$, onde r é a parte real e ω é a frequência angular, que representa a parte imaginária. Para encontrar esse par por meio de raízes de um polinômio, este polinômio seria $(s - p_1)(s - p_2) = (s - (r + j\omega))(s - (r - j\omega)) = s^2 - 2rs + (r^2 + \omega^2)$. Desse modo, podemos dizer que $r = \text{Re}(p_1)$ e $\omega = \text{Im}(p_1)$. Logo, já que essa estrutura irá gerar os zeros que possuem mesmo valor dos polos, anulando-os, podemos então escrever que o nosso numerador do filtro Notch será $N_n(s) = s^2 - 2 * \text{Re}(p_1) * s + (\text{Re}(p_1)^2 + \text{Im}(p_1)^2)$.

2.

Tendo descoberto tanto $N_n(s)$, quanto $D_n(s)$, ou seja, o numerador e denominador correspondentes ao filtro Notch que precisávamos, associamos a ele a nossa função de transferência $G^*(s)$ previamente projetada. Com isso, anula-se os dois polos p_1 e p_2 decorrentes do denominador $D^*(s)$ e são adicionados ao sistemas os dois pares de polos conjugados devido a $D_n(s)$ ao sistema, lembrando de normalizar o ganho, de forma que o ganho DC estático do sistema seja unitário.

Projeto do controlador P&D:

Mais uma vez, ressalta-se a aparição das respostas dadas no PR6 sem muitas alterações, com a exceção da adição de alguns gráficos e comentários relativos a eles, solicitados no roteiro do experimento 6. Desse modo, o detalhamento dessa parte do projeto segue representada abaixo conforme tal documento:

1.

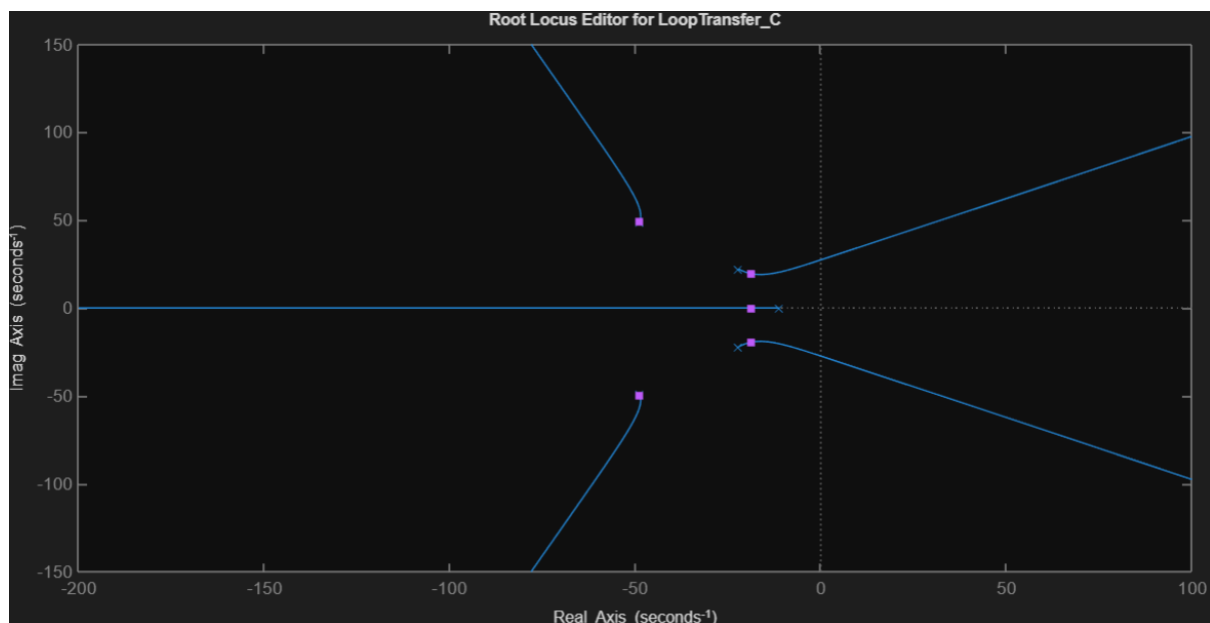
Para encontrar o lugar das raízes com relação a K_d , considerando a obtenção do máximo amortecimento para os pólos dominantes da função de transferência da saída $\Theta_3(t)$, foi feito o seguinte procedimento para encontrar o ganho K_d . Essa expressão foi projetada e aplicada no Matlab, utilizando os valores encontrados até agora:

$$\Theta_3(s) = G_r^* \frac{N_n}{D_n} (R' - \Theta_3 K_d s)$$

$$\frac{\Theta_3(s)}{R'(s)} = \frac{G_r^* \frac{N_n}{D_n}}{1 + G_r^* \frac{N_n}{D_n} K_d s} = G_r'(s)$$

$$\text{LR: } 1 + K_d \frac{G_r^* N_n s}{D_n} = 0 \rightarrow 1 + K_d \frac{K_{nw} N_3 s}{D + K_{nw} K_v S N_1} \frac{N_n}{D_n} = 0$$

Então, foi usado o Matlab para se encontrar o valor de K_d desejado, para maior amortecimento possível, sendo ele $K_d = 0.00242$. A técnica utilizada se baseou no uso da função que analisa o lugar das raízes para a função de transferência do item, que agora pode ser expressa numericamente, já que descobrimos o valor de k_v anteriormente. Manipulou-se as raízes de modo que todas as raízes estivessem o mais afastadas possível do eixo imaginário simultaneamente, garantindo maior amortecimento associado ao sistema. A ferramenta que auxiliou esse processo foi, mais uma vez, a chamada “controlSystemDesigner”, agora aplicada para a função de transferência em questão e lembrando, novamente, de realizar o processo de normalização da mesma em relação ao seu ganho DC para permitir encontrarmos o coeficiente K_d de modo correto. Veja a seguir o gráfico de lugar das raízes relativo ao item em questão, bem como a localização dos polos após terem sido “movimentados” para que representassem o máximo amortecimento.

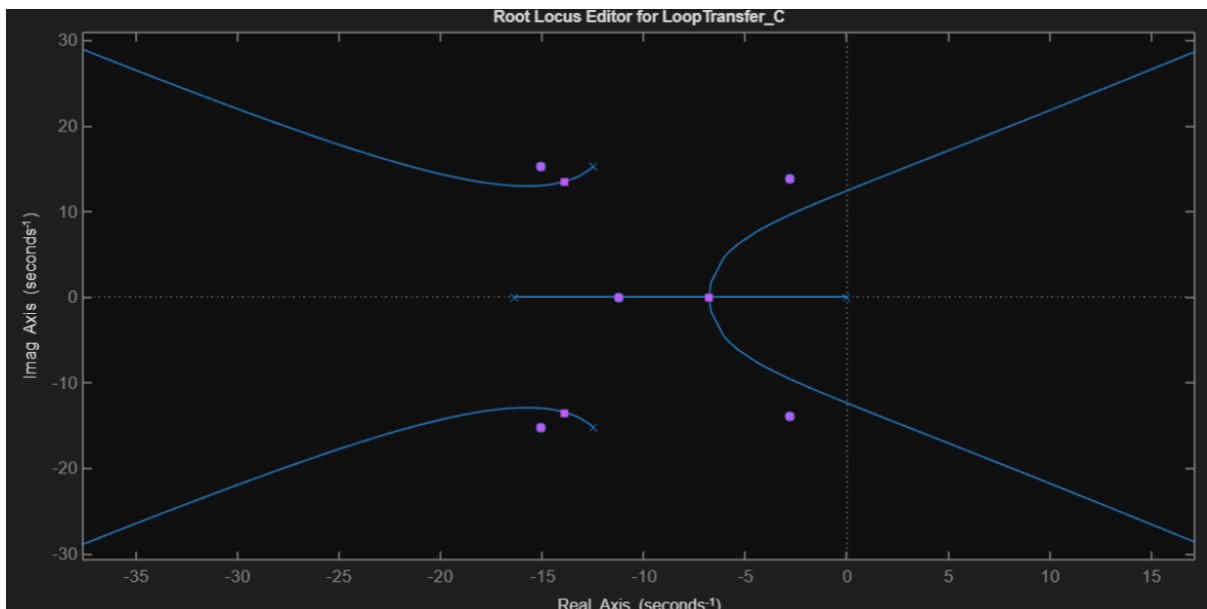


2.

Para encontrar o valor de K_p , em processo semelhante ao usado para k_v e K_d , foi feito o seguinte procedimento para chegar a $G^*(s)$ e ao lugar das raízes:

$$\begin{aligned}\Theta_3(s) &= G_1'(s) K_p (R - \Theta_3) \\ \frac{\Theta_3(s)}{R(s)} &= G_1(s) = \frac{G_1'(s) K_p}{1 + G_1' K_p} \\ L.R.: & \quad 1 + G_1' K_p = 0\end{aligned}$$

Pelos mesmos procedimentos e através da geração de programas em ambiente Matlab, foi encontrado o valor $K_p = 0.0345$, também considerando a manipulação partindo da função de representação gráfica de lugar das raízes a fim de obter o máximo amortecimento associado aos polos, seguindo, assim como em itens anteriores, a utilização da chamada “controlSystemDesigner” para a função de transferência, permitindo a manipulação dos seus polos para o máximo amortecimento do sistema, algo que nos permite encontrar o coeficiente K_p . Veja abaixo o gráfico de lugar das raízes resultante após essa referida manipulação:



Implementação no software ECP:

Ainda utilizando a estrutura construída para o PR6, utilizamos os resultados para os coeficientes polinomiais t_0 , t_1 , t_2 , s_0 , s_1 , s_2 , s_3 , r_0 , r_1 , r_2 , r_3 e r_4 que foram utilizados para as atividades experimentais seguintes deste laboratório. Os resultados obtidos para o PR6 que foram também considerados corretos e utilizados para as atividades deste experimento estão expostos abaixo, na ordem de valor de termo de menor coeficiente estando à esquerda (primeiro aparece o valor relativo a t_0 , depois a t_1 , e assim sucessivamente). Relembrando que o valor adotado para r_4 foi 1, uma vez que a aproximação decorrente da execução do programa não nos permite visualizar o real valor para esse coeficiente, após listando-o como $10^6 * 0.0000$.

```
t =  
1.0e+05 *  
1.6265    0.0481    0.0087  
  
s =  
1.0e+05 *  
1.6265    0.1622    0.0121    0.0006  
  
r =  
1.0e+06 *  
4.7146    0.3087    0.0101    0.0001    0.0000
```

Procedimento Experimental

Durante o procedimento experimental, após a implementação do sistema no software ECP e a configuração do sistema como se desejava, foram gravadas algumas respostas ao degrau e à rampa geradas por tal sistema. Assim, após a apresentação dos gráficos gerados que se seguem, dispõem-se também, comentários a respeito dos resultados observados. Ademais, tem-se também a apresentação de alguns resultados obtidos com a aplicação de distúrbios externos ao sistema controlado (aplicada pelos próprios integrantes do grupo).

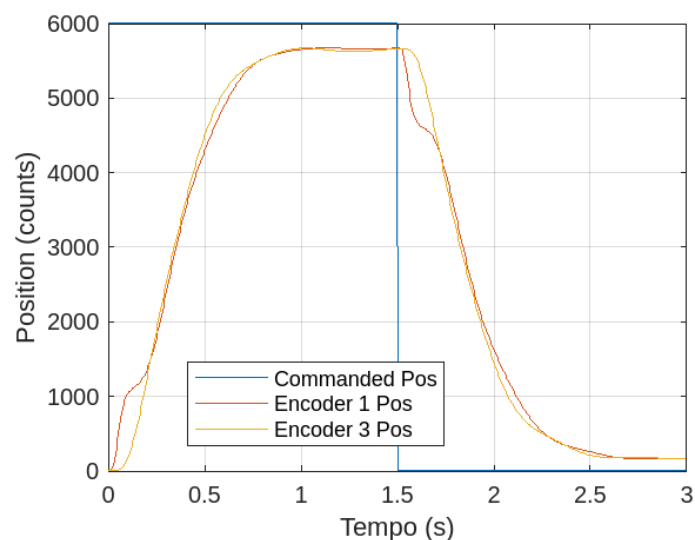


Figura 1: Resposta ao degrau do sistema antes da alteração iterativa das constantes de controle

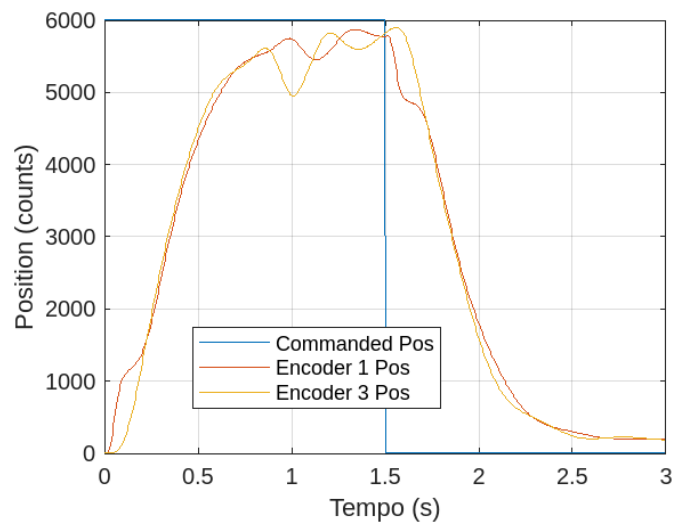


Figura 2: Resposta ao degrau do sistema antes da alteração iterativa das constantes de controle, com distúrbio

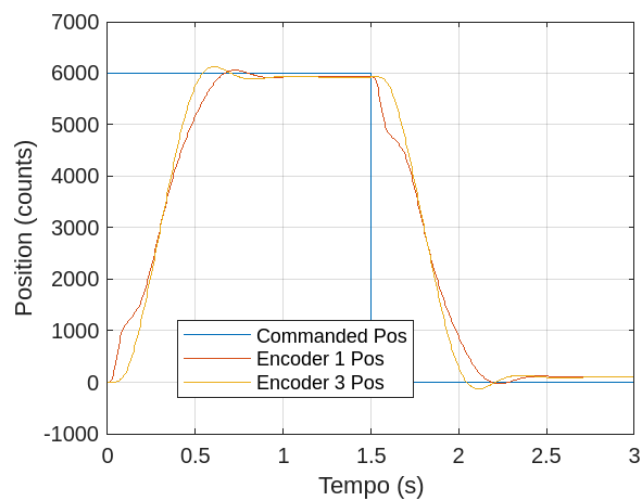


Figura 3: Resposta ao degrau do sistema após alteração iterativa das constantes de controle

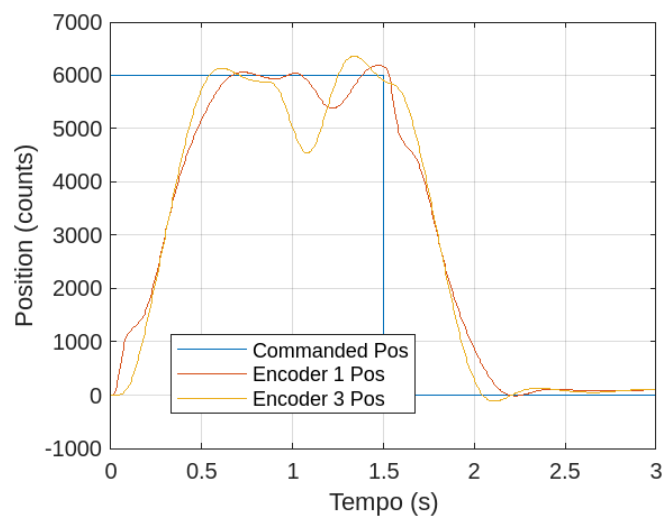


Figura 4: Resposta ao degrau do sistema após alteração iterativa das constantes de controle com distúrbio

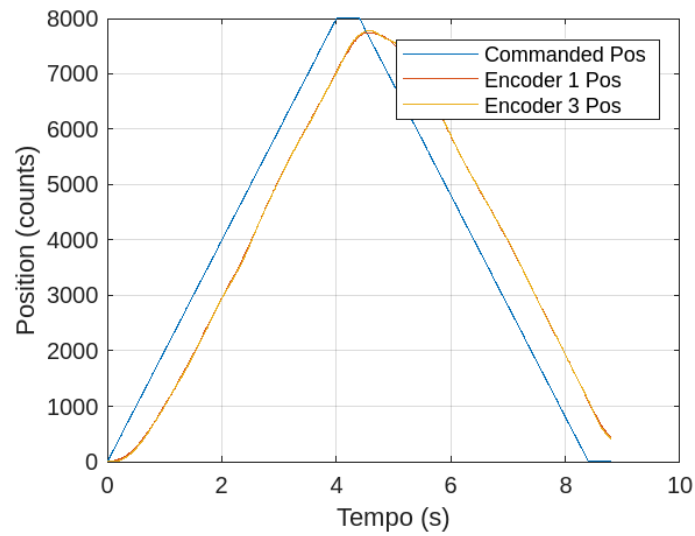


Figura 5: Resposta à rampa do sistema antes da alteração iterativa das constantes de controle

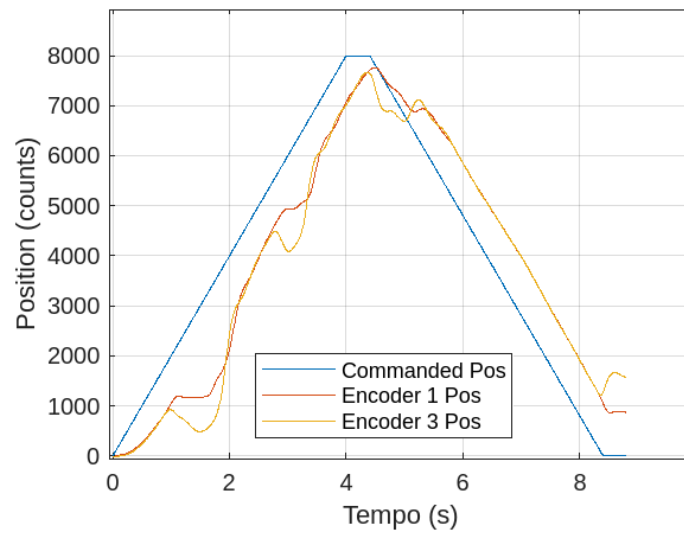


Figura 6: Resposta à rampa do sistema antes da alteração iterativa das constantes de controle com distúrbio

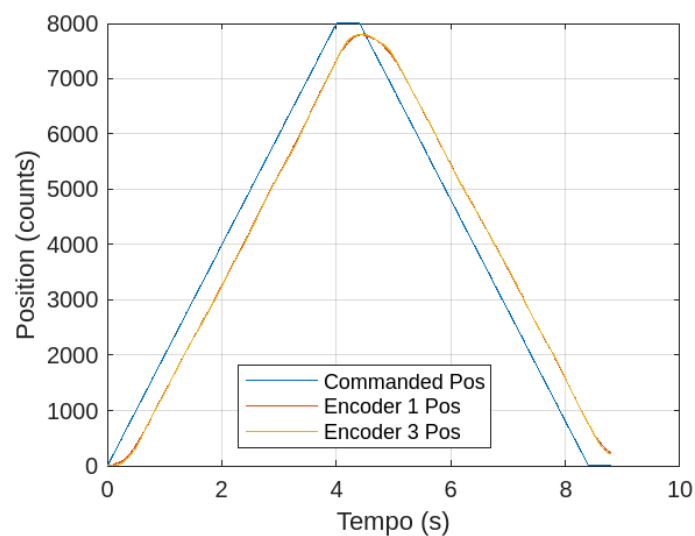


Figura 7: Resposta à rampa do sistema após alteração iterativa das constantes de controle

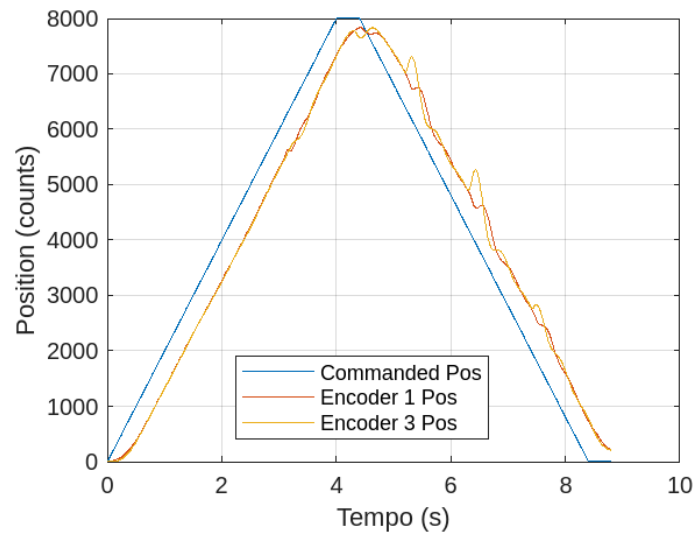


Figura 8: Resposta à rampa do sistema após alteração iterativa das constantes de controle com distúrbio

Uma coisa que podemos perceber, logo ao observar os gráficos que exibem as respostas dos sistemas, é a presença de um pequeno erro persistente durante os períodos de estabilização, em ambas as situações de entradas em degrau e rampa. Já podemos atribuir essa particularidade devido à característica não integrativa em relação ao modelo de controlador implementado, resultando numa persistência do erro de regime das situações vistas. A alteração dos coeficientes de ganho após o processo iterativo reduz o erro médio por adequar melhor a posição dos discos real com a posição comandada, mas não é capaz de realizar a mesma adequação feita por sistemas com a implementação de ganhos integrativos.

Além disso, podemos perceber como a implementação do controle não-allocado impacta positivamente no desempenho do sistema: após o ajuste paramétrico iterativo, vemos que o disco 3, que, antes, conforme observado no laboratório 5, era extremamente difícil de se controlar e estabilizar, passa a conseguir acompanhar o movimento do disco 1, aproximando-se, conseqüentemente, da posição comandada ideal.

Podemos ainda citar a capacidade do sistema de ajustar-se frente à introdução de forças externas ao sistema: durante algumas instâncias de funcionamento do sistema para as entradas de degrau e de rampa, os discos do sistema torcional foram manualmente rotacionados, de modo leve, para não atrapalhar o funcionamento da máquina ou alterar muito o curso da resposta. Isso, obviamente, altera a posição mensurada dos discos. Porém, como podemos ver, o sistema consegue se ajustar, com velocidade considerável, após pararmos de introduzir esses distúrbios, tanto no disco 1, quanto no disco 3.

Conclusão

Para concluir, devemos destacar que foi possível verificar a capacidade de um sistema baseado em um controlador não co-allocado, destacando-se, especialmente, a sua capacidade de resistir e se ajustar após a introdução de distúrbios externos ao sistema. Além disso, foi possível observar o efeito da variação dos parâmetros de modo grande, mostrando que pequenas alterações possuem mudanças significativas no modo de operação do sistema, como pudemos perceber ao realizar todo o processo de cálculos partindo dos valores encontrados para k_v , e em seguida, para K_p e K_d .