

Universidade de Brasília  
Departamento de Engenharia Elétrica  
ENE0167 - Controle Digital



## **Controle PI de Posição para Servomecanismo Translacional no Espaço de Estados**

### **Grupo 8:**

Isabela Maria Pereira Cruzeiro, 18/0102362

Matheus Teixeira de Sousa, 18/0107101

Brasília, DF  
2023

## Resumo

Os sistemas de controle são um pilar da civilização moderna, por meio deles foram possíveis avanços em diversas áreas como a agronomia, medicina e até a exploração espacial. Com os avanços nos campos da computação digital e da teoria de controle moderno, a implementação de controladores digitais para problemas complexos se tornou uma realidade. É com bases nessas definições que este trabalho se propõe a projetar um controlador PI digital, no espaço de estados, para controle de posição do servomecanismo translacional IP02 da Quanser a ser implementado no *Simulink* do Matlab. O projeto do controlador envolve desde a caracterização do sistema até o projeto de um observador de ordem mínima. Dentre os resultados obtidos, alcançamos respostas adequadas em ambiente simulado para diferentes respostas, mas, no laboratório, não o conseguimos em função da falta de familiaridade com o equipamento.

**Palavras-chaves:** controle digital, servomecanismo, controlador PI, espaço de estados.

## Sumário

<b>1</b>	<b>Introdução</b>	<b>4</b>
<b>2</b>	<b>Fundamentação Teórica</b>	<b>5</b>
2.1	Teorema da amostragem . . . . .	5
2.2	Resposta transitória . . . . .	5
2.2.1	Resposta transitória no plano $z$ . . . . .	6
2.3	Espaço de estados . . . . .	6
2.3.1	Espaço de estados em tempo discreto . . . . .	7
2.3.2	Controlabilidade . . . . .	7
2.3.3	Observabilidade . . . . .	7
2.3.4	Forma canônica controlável . . . . .	7
2.4	Realimentação de estados . . . . .	8
2.5	Observador de estados . . . . .	9
2.5.1	Observador de ordem mínima . . . . .	10
2.6	Controle de servomecanismo no espaço de estados . . . . .	11
<b>3</b>	<b>Metodologia</b>	<b>14</b>
3.1	Modelagem do servomecanismo . . . . .	14
3.2	Caracterização do sistema . . . . .	14
3.3	Discretização no espaço de estados . . . . .	15
3.4	Controlador PI . . . . .	16
3.4.1	Transformação para FCC . . . . .	17
3.4.2	Polos no plano $z$ . . . . .	17
3.4.3	Ganhos $K_1$ e $K_2$ . . . . .	18
3.5	Observador de estados . . . . .	18
3.6	Implementação no <i>Simulink</i> . . . . .	19
3.6.1	Adaptação do arquivo do laboratório . . . . .	20
<b>4</b>	<b>Resultados</b>	<b>21</b>
4.1	Resultados simulados . . . . .	21
4.1.1	Resposta ao degrau . . . . .	21
4.1.2	Resposta a onda quadrada . . . . .	22
4.1.3	Resposta à rampa . . . . .	23
4.2	Resultados experimentais . . . . .	24

<b>5 Conclusão</b>	<b>26</b>
<b>Referências</b>	<b>27</b>
<b>APÊNDICE A - Modelagem do servomecanismo</b>	<b>28</b>
<b>APÊNDICE B - Códigos do projeto</b>	<b>29</b>
B.1 - Caracterização do modelo . . . . .	29
B.2 - Projeto do controlador PI e do observador . . . . .	30
B.3 - Exporta parâmetros calculados . . . . .	32
<b>APÊNDICE C - Resultados experimentais complementares</b>	<b>33</b>
<b>APÊNDICE D - Comentários pós-apresentação</b>	<b>34</b>

## 1. Introdução

Os sistemas de controle são um pilar da civilização moderna, por meio deles foram possíveis avanços em diversas áreas como a agronomia, medicina e até a exploração espacial. Para Yamamoto [2022], os sistemas de controle são onipresentes, uma vez que nos rodeiam diariamente, desde os sistemas que regulam o nosso corpo até os telefones celulares. Assim, como a tendência dos sistemas modernos é aumentar a sua complexidade, o entendimento das teorias relativas a esses mecanismos é fundamental para diversos campos da engenharia. É exatamente nesse contexto que, desde aproximadamente 1960, a teoria de controle moderno tem sido desenvolvida [Ogata, 2010], capaz de lidar com sistemas de múltiplas entradas e saídas (MIMO), que podem ser lineares ou não e variantes ou invariantes no tempo.

As etapas abordadas no projeto de controle buscam a confiabilidade, robustez, economia e a performance, de maneira a otimizar o comportamento do sistema levando em conta as restrições encontradas como incertezas, medidas de sensores. A teoria de controle de sistemas em tempo contínuo já está estabelecida nas diversas áreas comentadas, mas, com os avanços na computação digital, os controladores passaram a ser mais velozes e as restrições em relação à capacidade de processamento e custo ficaram menores. Estes controladores são de fácil implementação em diversos sistemas físicos, apresentam menor sensibilidade a ruídos e desgastes de componentes.

É com base nas definições apresentadas até então que este trabalho se propõe a projetar um controlador PI digital, no espaço de estados, para controle de posição do servomecanismo translacional IP02 da Quanser a ser implementado no *Simulink* do Matlab. Primeiro, a seção 2, apresenta os principais conceitos utilizados durante o desenvolvimento do projeto. Depois, a seção 3 detalha as etapas de desenvolvimento da solução proposta, desde a caracterização do sistema utilizado até o projeto do observador de estados. Então, a seção 4 discute os resultados obtidos em ambiente simulado e em laboratório. Por fim, a seção 5 é dedicada a observações finais sobre o projeto e os resultados.

## 2. Fundamentação Teórica

### 2.1. Teorema da amostragem

Se definirmos  $\omega_s = 2\pi/T$  como a frequência de amostragem, em que  $T$  é o período de amostragem, temos que o Teorema da amostragem define que se  $\omega_s$  for maior que  $2\omega_1$ , em que  $\omega_1$  é a componente de maior frequência no sinal original, então este pode ser completamente reconstruído a partir do sinal amostrado [Ogata, 1995]. Usualmente,  $\omega_s$  é escolhido para ser entre 10 e 20 vezes o valor de  $\omega_1$ .

### 2.2. Resposta transitória

A resposta temporal de um sistema de controle é composta de duas partes, resposta transitória e resposta estacionária. Esta se refere ao comportamento do sinal quando o tempo tende ao infinito, enquanto aquela descreve o comportamento do sistema ao passar do estado inicial ao estado final Ogata, 2010. Em sistemas de segunda ordem, a função de transferência em malha fechada do sistema é dada pela Equação 1, chamada de forma-padrão do sistema,

$$\frac{Y(s)}{R(s)} = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}. \quad (1)$$

O comportamento dinâmico da resposta depende dos dois parâmetros apresentados na Equação 1:  $\omega_n$ , frequência natural não amortecida, e  $\zeta$ , coeficiente de amortecimento do sistema. Para  $0 < \zeta < 1$ , o sistema é subamortecido e a resposta transitória ao degrau oscila com uma envoltória exponencial decrescente. Enquanto isso, para  $\zeta = 0$ , o sistema não tem amortecimento e a resposta ao degrau tem oscilação sustentada. Por fim, para  $\zeta = 1$ , sistema criticamente amortecido, e  $\zeta > 1$ , sistema superamortecido, a resposta ao degrau não tem oscilação [Ogata, 2010]. Se o sistema for subamortecido, a Equação 1 pode ser reescrita como

$$\frac{Y(s)}{R(s)} = \frac{\omega_n^2}{(s + \sigma + j\omega_d)(s + \sigma - j\omega_d)}, \quad (2)$$

em que  $\sigma = \zeta\omega_n$  e  $\omega_d = \omega_n\sqrt{1 - \zeta^2}$ , frequência natural amortecida, definem, respectivamente, as partes real e imaginária dos polos do sistema. Usualmente, as características de desempenho são especificadas em termos da resposta transitória ao degrau unitário. Nesse contexto, é comum especificar o seguinte:

- Tempo de atraso ( $t_d$ ): tempo necessário para que a resposta alcance metade de seu valor final pela primeira vez;
- Tempo de subida ( $t_r$ ): tempo necessário para que a resposta passe de 10% a 90% (ou de 0 a 100%) do seu valor final. Em função dos parâmetros da Equação 2, pode ser definido como

$$t_r = \frac{1}{\omega_d} \arctan\left(\frac{\omega_d}{-\sigma}\right); \quad (3)$$

- Tempo de pico ( $t_p$ ): tempo para que a resposta atinja o valor máximo de sobressinal. Em função dos parâmetros da Equação 2, pode ser definido como

$$t_p = \frac{\pi}{\omega_d}; \quad (4)$$

- Máximo sobressinal ( $M_p$ ): valor máximo de pico da curva de resposta. Em função dos parâmetros da Equação 2, pode ser definido como

$$M_p = e^{-\sigma\pi/\omega_d} = e^{-(\zeta\pi/\sqrt{1-\zeta^2})}. \quad (5)$$

É comum definir o sobressinal em termos percentuais. Para isso, basta multiplicar o resultado por 100%;

- Tempo de acomodação ( $t_s$ ): tempo necessário para que a resposta alcance valores em uma faixa de 2% (Equação 6) ou 5% (Equação 7) em torno do valor final.

$$t_s = \frac{4}{\sigma}, \quad (6)$$

$$t_s = \frac{3}{\sigma}. \quad (7)$$

### 2.2.1. Resposta transitória no plano $z$

Até então, as especificações da resposta transitória foram definidas em tempo contínuo considerando o plano de Laplace e os polos no formato  $s = -\sigma \pm j\omega_d$ . No entanto, para fins de projeto em tempo discreto, é possível mapear a localização dos polos do plano  $s$  para o plano  $z$  e, conseqüentemente, a resposta desejada por meio da expressão

$$z = e^{sT} \rightarrow z = e^{-\sigma T} e^{j\omega_d T}, \quad (8)$$

em que  $T$  é o período de amostragem, e  $e^{-\sigma T}$  e  $\omega_d T$  são o módulo e a fase do polo  $z$ . Dessa forma, partindo de especificações e polos em tempo contínuo, é possível mapear polos correspondentes em tempo discreto.

### 2.3. Espaço de estados

À medida que a complexidade de um sistema aumenta, múltiplas entradas e saídas inter-relacionadas de maneira complexa, a abordagem tradicional de relacionar entrada e saída via função de transferência pode não ser a mais adequada. Nesse sentido, a teoria de controle moderno é baseada na descrição de um sistema em termos de  $n$  equações diferenciais de primeira ordem combinadas em um formato matricial [Ogata, 2010]. Considerando, inicialmente, um sistema linear invariante no tempo em tempo contínuo, temos a representação no espaço de estados dada por

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t), \\ y(t) &= Cx(t) + Du(t), \end{aligned} \quad (9)$$

em que  $y(t)$  é o vetor de saída,  $u(t)$  é o vetor de controle e  $x(t)$  é vetor de estados. Além disso, as matrizes  $A$ ,  $B$ ,  $C$  e  $D$  representam, respectivamente, a matriz de estado, a matriz de entrada, a matriz de saída e a matriz de transmissão direta [Ogata, 2010].

### 2.3.1. Espaço de estados em tempo discreto

Agora, considerando o tempo discreto, temos a representação no espaço de estados dada pelas Equações 10, em que as matrizes  $C$  e  $D$  são as mesmas já apresentadas em tempo contínuo e as matrizes  $G$  e  $H$  são as novas matrizes de estado e de entrada. A equação de estados é chamada de equação equivalente com segurador de ordem-zero [Ogata, 1995].

$$\begin{aligned} x[k+1] &= Gx[k] + Hu[k], \\ y[k] &= Cx[k] + Du[k]. \end{aligned} \quad (10)$$

A discretização das matrizes  $A$  e  $B$  nas matrizes  $G$  e  $H$  é dada pelas Equações 11 considerando um período de amostragem  $T$ . No cálculo das matrizes  $G$  e  $H$  as propriedades de exponencial de matriz,  $e^A = \mathcal{L}^{-1}\{(Is - A)^{-1}\}$ , integral de uma matriz e multiplicação de matrizes devem ser consideradas.

$$G = e^{AT} \quad e \quad H = \left( \int_0^T e^{A\tau} d\tau \right) B. \quad (11)$$

### 2.3.2. Controlabilidade

Considerando a equação de estados das Equações 10, o par  $(G, H)$  é dito controlável se, e somente se, o determinante da matriz de controlabilidade de tamanho  $n \times n$  tiver posto igual a  $n$ . Nesse caso, para quaisquer estados inicial  $x_0$  e final  $x_1$ , existe uma entrada  $u[k]$  que leva o sistema de  $x_0$  para  $x_1$  em um intervalo finito de tempo [Ferreira, 2023a]. A matriz de controlabilidade pode ser definida como

$$\mathcal{C} = [H \quad GH \quad \dots \quad G^{n-1}H]. \quad (12)$$

### 2.3.3. Observabilidade

Considerando as Equações 10, o par  $(G, C)$  é dito observável se, e somente se, o determinante da matriz de observabilidade de tamanho  $n \times n$  tiver posto igual a  $n$ . Nesse caso, para qualquer estado inicial  $x_0$  desconhecido e para entrada  $u[k]$  e saída  $y[k]$  conhecidas, é suficiente para determinar univocamente o estado inicial  $x_0$  [Ferreira, 2023a]. A matriz de observabilidade pode ser definida como

$$\mathcal{O} = [C^* \quad G^*C^* \quad \dots \quad (G^*)^{n-1}C^*], \quad (13)$$

em que a notação  $X^*$  define o conjugado transposto da matriz  $X$ .

### 2.3.4. Forma canônica controlável

Considerando a relação entrada e saída dada pela seguinte função de transferência no tempo discreto dada por



$$\frac{Y(z)}{U(z)} = \frac{b_0 z^n + b_1 z^{n-1} + \dots + b_n}{z^n + a_1 z^{n-1} + \dots + a_n}, \quad (14)$$

é possível escrever o sistema no espaço de estados como

$$\begin{bmatrix} x_1[k+1] \\ x_2[k+1] \\ \vdots \\ x_{n-1}[k+1] \\ x_n[k+1] \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \\ -a_n & -a_{n-1} & -a_{n-2} & \dots & -a_1 \end{bmatrix} \begin{bmatrix} x_1[k] \\ x_2[k] \\ \vdots \\ x_{n-1}[k] \\ x_n[k] \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} u[k], \quad (15)$$

$$y[k] = [b_n - a_n b_0 \quad b_{n-1} - a_{n-1} b_0 \quad \dots \quad b_1 - a_1 b_0] \begin{bmatrix} x_1[k] \\ x_2[k] \\ \vdots \\ x_n[k] \end{bmatrix} + b_0 u[k]. \quad (16)$$

As Equações 15 e 16 são as equações de estado e saída, e esta representação é comumente chamada de forma canônica controlável (FCC) [Ogata, 1995]. O sistema de Equações 10 com matrizes  $G$ ,  $H$ ,  $C$  e  $D$  genéricas pode ser levado para a forma canônica controlável por meio de uma matriz de transformação  $P$  da forma

$$P = \mathcal{C} \begin{bmatrix} a_{n-1} & a_{n-2} & \dots & a_1 & 1 \\ a_{n-2} & a_{n-3} & \dots & 1 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_1 & 1 & \dots & 0 & 0 \\ 1 & 0 & \dots & 0 & 0 \end{bmatrix} = \mathcal{C}W, \quad (17)$$

em que os elementos da segunda matriz (chamada de  $W$ ) são dados pela equação característica

$$|zI - G| = z^n + a_1 z^{n-1} + \dots + a_{n-1} z + a_n = 0. \quad (18)$$

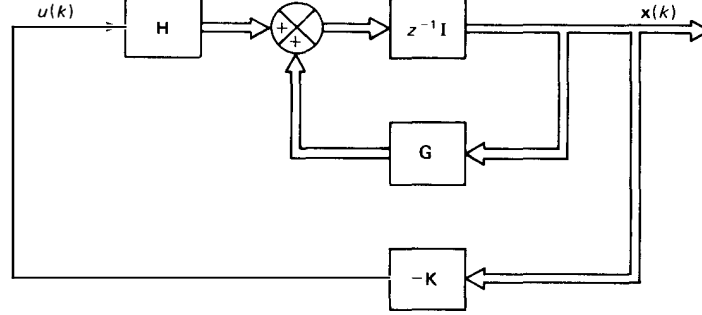
Assim, se definirmos  $x[k] = P\bar{x}[k]$ , temos que as matrizes  $\bar{G}$ ,  $\bar{H}$ ,  $\bar{C}$  e  $\bar{D}$  do sistema na forma canônica controlável são definidas como

$$\bar{G} = P^{-1}GP, \quad \bar{H} = P^{-1}H, \quad \bar{C} = CP, \quad e \quad \bar{D} = D. \quad (19)$$

#### 2.4. Realimentação de estados

No método de projeto de via alocação de polos, um ganho de realimentação é projetado para levar os polos do sistema em malha fechada para uma localização de interesse. Neste método, é assumido que todas as variáveis de estado são mensuráveis e estão disponíveis, e o sistema é completamente controlável. Considerando o sistema apresentado na Figura 1, temos a seguinte descrição no espaço de estados:

$$\begin{aligned} x[k+1] &= Gx[k] + Hu[k], \\ u[k] &= -Kx[k], \end{aligned} \quad (20)$$



**Figura 1. Diagrama de blocos - Realimentação de estados, [Ogata, 1995, p. 403]**

em que a matriz  $K$  é o ganho de realimentação e para um sistema com  $n$  estados é escrita como

$$K = [k_1 \ \cdots \ k_n]. \quad (21)$$

Então, considerando as equações apresentadas, temos que a equação característica do sistema é dada por

$$|zI - G + HK| = (z - \lambda_1) \cdots (z - \lambda_n) = z^n + \alpha_1 z^{n-1} + \cdots + \alpha_n = 0, \quad (22)$$

em que os valores de  $\lambda_1$  até  $\lambda_n$  são os polos dos sistema em malha fechada. Dessa forma, sabendo a equação característica dos polos desejados, é possível escolher os valores de  $k_1$  até  $k_n$  tal que as equações sejam iguais. Quando o sistema está na forma canônica controlável, o ganho de realimentação pode ser calculado como

$$K = [0 \ 0 \ \cdots \ 0 \ 1] \mathcal{C}^{-1} \phi(G), \quad (23)$$

em que  $\phi(G)$  é o polinômio característico com  $z = G$ , ou seja,

$$\phi(G) = G^n + \alpha_1 G^{n-1} + \cdots + \alpha_{n-1} G + \alpha_n I. \quad (24)$$

## 2.5. Observador de estados

No entanto, em alguns sistemas, nem todas as variáveis de estado podem ser diretamente observadas e, nesses casos, não é possível projetar uma realimentação como visto na seção anterior. Quando o sistema é completamente observável, é possível estimar os estados por meio de um observador de estados. Esse subsistema é capaz de estimar baseado na saída do sistema e nas variáveis de controle [Ogata, 1995]. Considerando um sistema com a matriz  $D$  igual a zero, o vetor de estados  $x[k]$  pode ser aproximado pelo vetor de estados  $\tilde{x}[k]$  pelo modelo

$$\begin{aligned}\tilde{x}[k+1] &= G\tilde{x}[k] + Hu[k], \\ \tilde{y}[k] &= C\tilde{x}[k].\end{aligned}\tag{25}$$

O erro da estimativa do observador pode ser definido como a diferença entre o estado real e o estado estimado (Equação 26). Se subtraímos as equações de estado do sistema observador e do estimado pelo observador, temos a Equação 27.

$$e[k] = x[k] - \tilde{x}[k] \rightarrow e[k+1] = x[k+1] - \tilde{x}[k+1],\tag{26}$$

$$x[k+1] - \tilde{x}[k+1] = G(x[k] - \tilde{x}[k]) \rightarrow e[k+1] = Ge[k].\tag{27}$$

Se a matriz  $G$  for estável, então  $e[k]$  vai se aproximar de zero e  $\tilde{x}[k]$  vai se aproximar de  $x[k]$ . No entanto, para suportar situações nas quais a matriz  $G$  não é estável, a equação de estado do observador deve ser modificada para

$$\tilde{x}[k+1] = G\tilde{x}[k] + Hu[k] + K_e(y[k] - C\tilde{x}[k]),\tag{28}$$

em que a matriz  $K_e$  é um ganho para levar os polos do observador para uma posição adequada e, agora, a equação passa a pesar o erro entre as saídas observada e medida. Além disso, vale destacar que os polos do observador devem ser mais rápidos que aqueles projetados para a planta para que este seja capaz de estimar os estados com velocidade adequada. Depois, uma vez que o vetor de estados estimado é calculado, este pode ser usado na realimentação de estados projetada na seção anterior.

### 2.5.1. Observador de ordem mínima

A ordem de um observador é definida pelo número de estados que ele estima. Quando o observador estima todos os estados de um sistema, este é dito de ordem completa. No entanto, em diversos sistemas, alguns dos estados podem ser medidos de forma acurada e, portanto, não precisam ser estimados. Se o observador estima menos que  $n$  estados, este é chamado de observador de ordem reduzida. Quando a ordem do observador é a menor possível, este é chamado de observador de ordem mínima.

O projeto de um observador de ordem mínima parte da divisão entre os estados medidos,  $x_a[k]$ , e os não medidos,  $x_b[k]$  do sistema original. Assim, temos a seguinte representação no espaço de estados

$$\begin{aligned}\begin{bmatrix} x_a[k+1] \\ x_b[k+1] \end{bmatrix} &= \begin{bmatrix} G_{aa} & G_{ab} \\ G_{ba} & G_{bb} \end{bmatrix} \begin{bmatrix} x_a[k] \\ x_b[k] \end{bmatrix} + \begin{bmatrix} H_a \\ H_b \end{bmatrix} u[k], \\ y[k] &= \begin{bmatrix} I & 0 \end{bmatrix} \begin{bmatrix} x_a[k] \\ x_b[k] \end{bmatrix},\end{aligned}\tag{29}$$

e a equação característica do observador de ordem mínima é dada por

$$|zI - G_{bb} + K_e G_{ab}| = 0.\tag{30}$$

A Figura 2 apresenta o diagrama de blocos do observador de ordem mínima com a implementação do ganho de realimentação nos estados estimados. Na figura, o sistema possui  $n$  estados, dos quais  $m$  não precisa ser estimados. Dessa forma, basta estimar  $n-m$  estados e depois juntar os estados estimados com aqueles medidos diretamente.

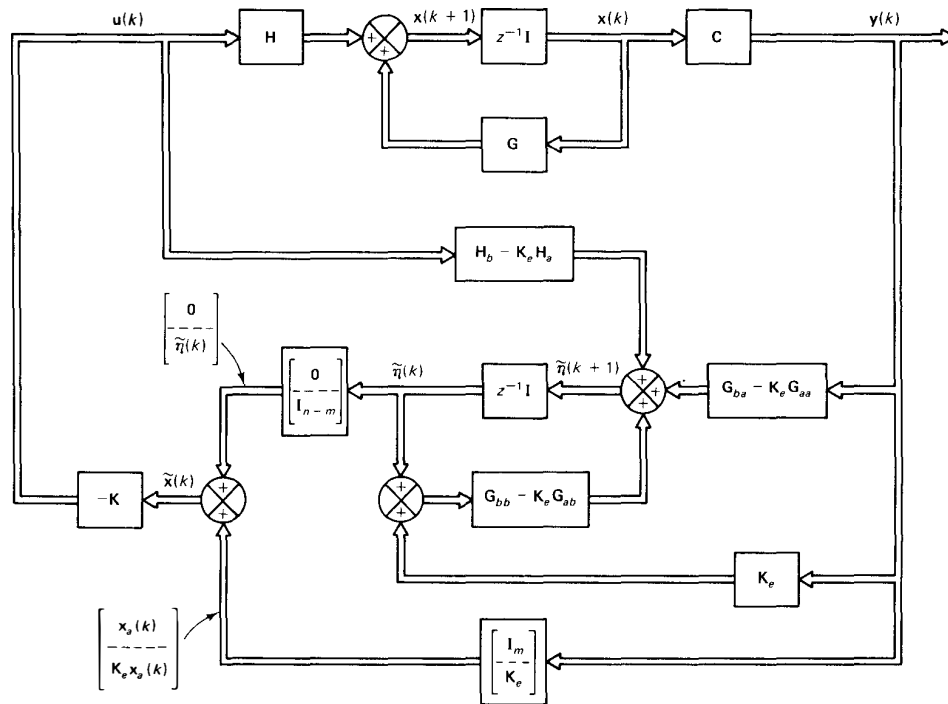


Figura 2. Diagrama de blocos - Observador de ordem mínima, [Ogata, 1995, p. 451]

## 2.6. Controle de servomecanismo no espaço de estados

O projeto de controle para servomecanismos usualmente envolve a adição de um ou mais integradores para eliminar o erro estacionário para entrada degrau. No espaço de estados, uma forma de introduzir um integrador é adicionar um estado que integra a diferença entre a saída  $y[k]$  e uma referência  $r[k]$ . A Figura 3 apresenta o diagrama de blocos do sistema proposto no capítulo 6 do livro do Ogata [1995].

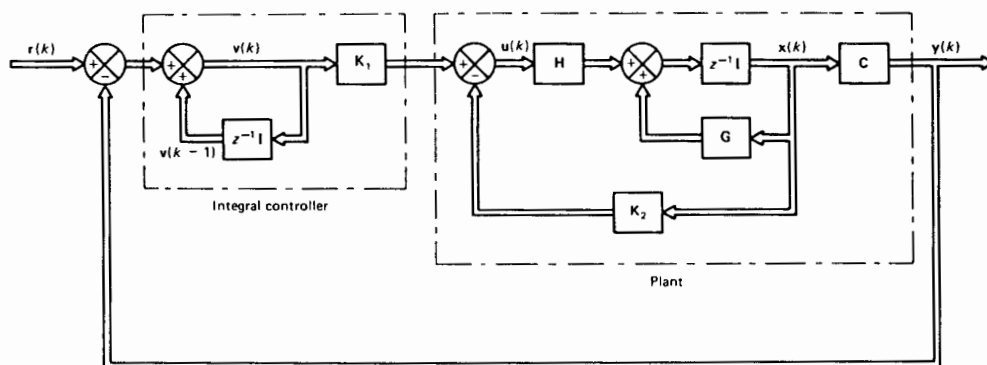


Figura 3. Diagrama de blocos - Servomecanismo, [Ogata, 1995, p. 461]

A equações de estado e de saída do sistema e de estado do integrador são definidas, a partir da figura, como

$$\begin{aligned} x[k+1] &= Gx[k] + Hu[k], \\ y[k] &= Cx[k], \end{aligned} \quad (31)$$

$$v[k] = v[k-1] + r[k] - y[k]. \quad (32)$$

Substituindo a saída na Equação 32 e escrevendo a ação de controle  $u[k]$ , temos

$$v[k+1] = -CGx[k] + v[k] - CHu[k] + r[k+1], \quad (33)$$

$$u[k] = -K_2x[k] + K_1v[k] \rightarrow u[k+1] = -K_2x[k+1] + K_1v[k+1]. \quad (34)$$

Então, substituindo as expressões de  $x[k+1]$  e  $v[k+1]$  na Equação 34, obtemos uma combinação linear de  $x[k]$  e  $u[k]$  e um novo vetor de estados formado por  $x[k]$  e  $u[k]$ . Com o novo vetor de estados, podemos escrever o sistema como

$$\begin{bmatrix} x[k+1] \\ u[k+1] \end{bmatrix} = \begin{bmatrix} G & H \\ K_2 - K_2G - K_1CG & I - K_2H - K_1CH \end{bmatrix} \begin{bmatrix} x[k] \\ u[k] \end{bmatrix} + \begin{bmatrix} 0 \\ K_1 \end{bmatrix} r[k+1], \quad (35)$$

$$y[k] = \begin{bmatrix} C & 0 \end{bmatrix} \begin{bmatrix} x[k] \\ u[k] \end{bmatrix}, \quad (36)$$

em que os polos de malha fechada do sistema são definidos pelos autovalores da matriz de estado da Equação 35. Para uma entrada  $r[k]$  degrau e definindo o erro dos estados como

$$\begin{aligned} x_e[k] &= x[k] - x[\infty] \quad \text{e} \\ u_e[k] &= u[k] - u[\infty], \end{aligned} \quad (37)$$

temos que a dinâmica do sistema pode ser definida como

$$\begin{bmatrix} x_e[k+1] \\ u_e[k+1] \end{bmatrix} = \begin{bmatrix} G & H \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_e[k] \\ u_e[k] \end{bmatrix} + \begin{bmatrix} 0 \\ I \end{bmatrix} w[k], \quad (38)$$

em que  $w[k]$  é dado por

$$w[k] = \begin{bmatrix} K_2 - K_2G - K_1CG & I - K_2H - K_1CH \end{bmatrix} \begin{bmatrix} x_e[k] \\ u_e[k] \end{bmatrix}. \quad (39)$$

Então, definindo  $\xi[k]$ ,  $\hat{G}$ ,  $\hat{H}$  e  $\hat{K}$  como

$$\xi[k] = \begin{bmatrix} x_e[k] \\ u_e[k] \end{bmatrix}, \quad \hat{G} = \begin{bmatrix} G & H \\ 0 & 0 \end{bmatrix}, \quad \hat{H} = \begin{bmatrix} 0 \\ I \end{bmatrix} \quad \text{e} \quad (40)$$

$$\hat{K} = - \begin{bmatrix} K_2 - K_2G - K_1CG & I - K_2H - K_1CH \end{bmatrix}. \quad (41)$$

temos o sistema no espaço de estados dado por

$$\begin{aligned} \xi[k+1] &= \hat{G}\xi[k] + \hat{H}w[k], \\ w[k] &= -\hat{K}\xi[k]. \end{aligned} \quad (42)$$

Uma vez que os polos em malha fechada são especificados, é possível calcular a matriz  $\hat{K}$  e, por fim, encontrar os ganhos  $K_1$  e  $K_2$  pela expressão

$$\begin{bmatrix} K_2 & K_1 \end{bmatrix} \begin{bmatrix} G - I & H \\ CG & CH \end{bmatrix} = \hat{K} + \begin{bmatrix} 0 & I \end{bmatrix}. \quad (43)$$

Depois que o ganho de realimentação for projetado, pode ser necessário implementar um observador de estados em função de estados não mensuráveis diretamente, como a aceleração. Nesse caso, é possível implementar um observador de estados de ordem mínima para os  $m$  estados não mensuráveis.

### 3. Metodologia

#### 3.1. Modelagem do servomecanismo

A função de transferência do servomecanismo translacional disponibilizado no laboratório é definida como

$$G(s) = \frac{b}{s(s+a)}, \quad (44)$$

em que os parâmetros  $a$  e  $b$  dependem de parâmetros do servomecanismo, e considera a tensão  $v_m(t)$  aplicada no motor e a posição do carro ao longo do trilho  $x(t)$  como entrada e saída, respectivamente [Ferreira, 2023b]. Além disso, o servomecanismo ainda pode ser descrito o espaço de estados como

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) \end{aligned} \quad (45)$$

em que as matrizes são definidas como

$$A = \begin{bmatrix} 0 & 1 \\ 0 & -a \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ b \end{bmatrix}, \quad C = [1 \quad 0] \quad (46)$$

e o vetor de estados é dado por  $x(t) = [x(t) \quad \dot{x}(t)]^T$ .

#### 3.2. Caracterização do sistema

Para a estimação dos parâmetros  $a$  e  $b$ , uma entrada degrau com amplitude 15V foi aplicada no sistema em malha fechada contendo um ganho de 420 em série com o servo mecanismo e realimentação unitária negativa (Figura 6). Na Figura 4, temos a resposta observada no servomecanismo, o tempo de pico e máximo sobre sinal. Das Equações 4 e 5, temos os parâmetros  $\omega_d$  e  $\sigma$  como

$$\omega_d = 23,982 \quad \text{e} \quad \sigma = 6,1903. \quad (47)$$

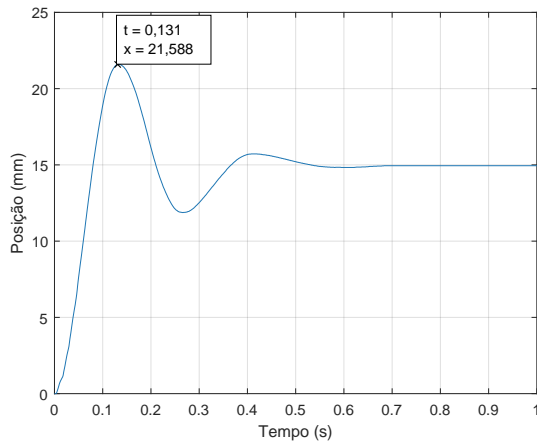


Figura 4. Resposta no modelo real

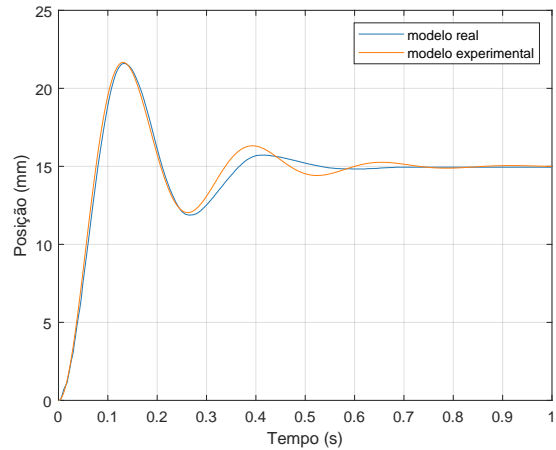
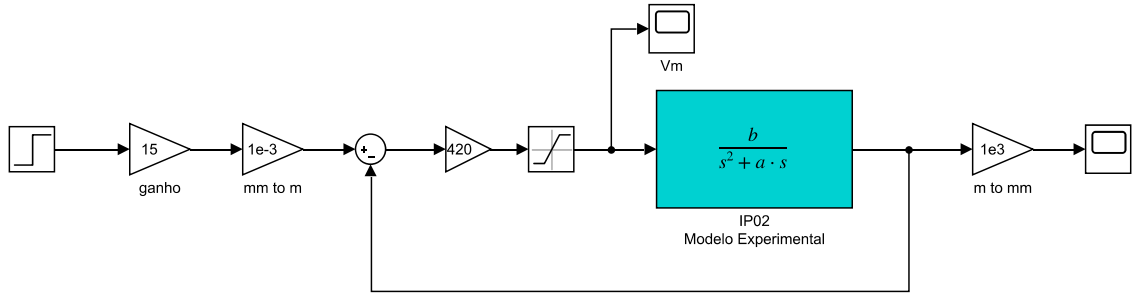


Figura 5. Respostas nos modelos real (azul) e experimental (laranja)



**Figura 6. Diagrama de blocos do modelo experimental**

Depois, comparando o denominador da Equação 2 com a expressão  $s^2 + as + 420b$ , oriunda da malha fechada do sistema experimental, temos

$$a = 12,3086 \quad \text{e} \quad b = 1,4606. \quad (48)$$

Então, o experimento foi repetido em simulação considerando a planta  $G(s)$  como o servomecanismo, cujo resultado pode ser visualizado na Figura 5. De fato, é possível observar que as duas curvas apresentam comportamento bastante semelhante, mas a saída real sofre efeitos de fenômenos não modelados, como o desgaste das engrenagens e o atrito do carrinho com o trilho. Dessa forma, a função de transferência do servomecanismo pode ser escrita como

$$G(s) = \frac{1,4606}{s(s + 12,3806)}, \quad (49)$$

e a representação no espaço de estados é dada por

$$\begin{aligned} \dot{x}(t) &= \begin{bmatrix} 0 & 1 \\ 0 & -12,3086 \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ 1,4606 \end{bmatrix} u(t) \\ y(t) &= [1 \quad 0] x(t) \end{aligned} \quad (50)$$

### 3.3. Discretização no espaço de estados

Considerando o Teorema da amostragem, o período de amostragem  $T$  deve ser tal que

$$\frac{\pi}{\omega_1} > T. \quad (51)$$

Em função do amortecimento do sinal, a componente de frequência observada na resposta ao degrau do sistema experimental é a frequência natural amortecida,  $\omega_d$ . Então, segue que

$$\frac{\pi}{\omega_d} = 0,1310 > T. \quad (52)$$



Tomando  $\omega_s$  como 20 vezes  $\omega_d$ , temos que o período de amostragem pode ser escolhido igual a 0,0131s. Para simplificar as contas, vamos adotar  $T = 0,01$ s, que corresponde a um  $\omega_s$  26 vezes maior que o  $\omega_d$ . Com o período de amostragem definido, podemos aplicar as Equações 11 para calcular as matrizes  $G$  e  $H$  como

$$G = e^{AT} = \begin{bmatrix} 1 & \frac{1-e^{-aT}}{a} \\ 0 & e^{-aT} \end{bmatrix} = \begin{bmatrix} 1 & 0,0094 \\ 0 & 0,8836 \end{bmatrix}, \quad (53)$$

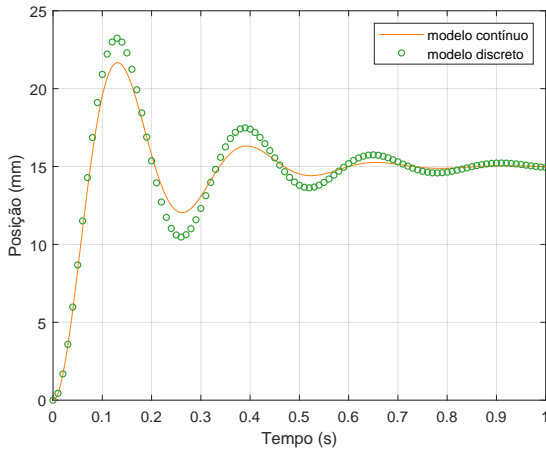
$$H = \left( \int_0^T e^{A\tau} d\tau \right) B = \begin{bmatrix} 0,0001 \\ 0,0137 \end{bmatrix}. \quad (54)$$

A representação completa de espaço de estados do servomecanismo translacional é dada por

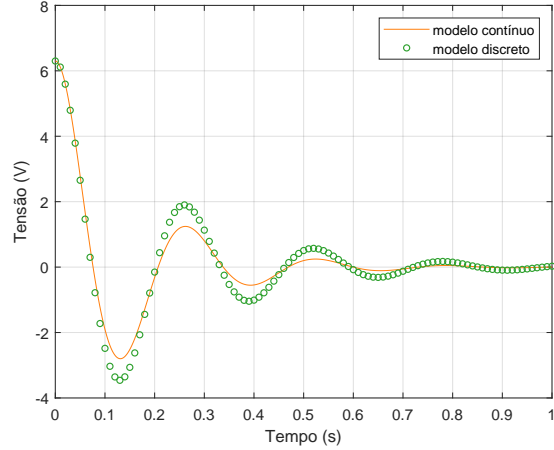
$$x[k+1] = \begin{bmatrix} 1 & 0,0094 \\ 0 & 0,8836 \end{bmatrix} x[k] + \begin{bmatrix} 0,0001 \\ 0,0137 \end{bmatrix} u[k]$$

$$y[k] = \begin{bmatrix} 1 & 0 \end{bmatrix} x[k].$$

As Figuras 7 e 8 apresentam comparações entre os modelos contínuo e discreto com os mesmos parâmetros utilizados na caracterização do sistema. Quanto menor é o período de amostragem, mais próximas são as duas curvas.



**Figura 7. Resposta ao degrau nos modelos contínuo (laranja) e discreto (verde)**



**Figura 8. Ação de controle nos modelos contínuo (laranja) e discreto (verde)**

### 3.4. Controlador PI

Em sistemas subamortecidos, é desejável que a resposta transitória seja suficientemente rápida e amortecida. Assim, para uma resposta transitória adequada, o coeficiente de amortecimento deve se situar entre 0,4 e 0,8. Para valores menores, há excessivo sobressinal e, para valores maiores, o sistema responde muito lentamente [Ogata, 2010]. O projeto da realimentação de estados visa reduzir o sobressinal de 44,33% para 10% e o tempo de acomodação de 0,57s para 0,3s para diminuir a oscilação e acelerar a acomodação da resposta.

Para esses requisitos, temos  $\zeta = 0,59$ , dentro do intervalo de valores adequados. Além disso, é preciso que o erro da resposta ao degrau em regime estacionário seja zero. Para atender os requisitos propostos, o projeto do controlador envolve um controlador PI e uma realimentação de estados, como desenvolvido na seção 2.6. Então, primeiro, reescrevemos a matriz  $H$  do sistema de modo a considerar o ganho de 420 utilizado na estimação dos parâmetros. Assim, a matriz  $H$  auxiliar é definida como

$$H_{aux} = 420 \begin{bmatrix} 0,0001 \\ 0,0137 \end{bmatrix} = \begin{bmatrix} 0,0294 \\ 5,7700 \end{bmatrix}. \quad (56)$$

Depois, verificamos a controlabilidade do sistema pela matriz

$$\mathcal{C} = [H_{aux} \quad GH_{aux}] = \begin{bmatrix} 0,0294 & 0,0837 \\ 5,7700 & 5,0981 \end{bmatrix}, \quad (57)$$

que possui posto igual a 2, e, portanto, o sistema é controlável.

#### 3.4.1. Transformação para FCC

Então, transformamos o sistema para a forma canônica controlável para facilitar o cálculo dos ganhos  $K_1$  e  $K_2$ . Para isso, extraímos a função de transferência da planta a partir da representação no espaço de estados como

$$\frac{0,0294z + 0,0283}{z^2 - 1,8836z + 0,8836}. \quad (58)$$

Dessa forma, temos que a transformação  $P$  pode ser escrita como

$$P = \mathcal{C} \begin{bmatrix} -1,8836 & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 0,0283 & 0,0294 \\ -5,7700 & 5,7700 \end{bmatrix}, \quad (59)$$

e o sistema na forma canônica controlável é dado por

$$\begin{aligned} \dot{\bar{x}}[k] &= \begin{bmatrix} 0 & 1 \\ -0,8836 & 1,8836 \end{bmatrix} \bar{x}[k] + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u[k], \\ y(t) &= [0,0283 \quad 0,0294] \bar{x}[k]. \end{aligned} \quad (60)$$

#### 3.4.2. Polos no plano $z$

Depois, considerando as especificações de tempo de acomodação e máximo sobressinal, temos, pelas Equações 6 e 5,  $\sigma = 13,33$  e  $\omega_d = 18,19$ . Dessa forma, os polos desejados no plano  $z$  são dados por

$$z = e^{-13,33T} / 18,19T = 0,8752 / 0,1819 = 0,8607 \pm 0,1583j, \quad (61)$$

em que, pelo módulo, é possível observar que os polos estão dentro do CRU (círculo de raio unitário) e são estáveis. Considerando que o polo extra, adicionado pelo novo estado, é igual a 0, temos que a equação característica desejada é dada por

$$z^3 - 1,721z^2 + 0,7659z = z^3 + \alpha_1 z^2 + \alpha_2 z + \alpha_3. \quad (62)$$

### 3.4.3. Ganhos K1 e K2

Para o cálculo dos ganhos, consideramos o desenvolvimento realizado supondo novas matrizes e vetor de estado. Então, considerando o sistema descrito pelas Equações 42, temos que as matrizes  $\hat{\bar{G}}$ ,  $\hat{\bar{H}}$  e  $\hat{\bar{K}}$  são dadas por

$$\hat{\bar{G}} = \begin{bmatrix} \bar{G} & \bar{H} \\ 0 & 0 \end{bmatrix}, \quad \hat{\bar{H}} = \begin{bmatrix} 0 \\ I \end{bmatrix}, \quad (63)$$

$$\hat{\bar{K}} = - \begin{bmatrix} K_2 - K_2 \bar{G} - K_1 \bar{C} \bar{G} & I - K_2 \bar{H} - K_1 \bar{C} \bar{H} \end{bmatrix}.$$

Como o sistema está na forma canônica controlável, temos, pela Equação 23, que

$$\hat{\bar{K}} = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \mathcal{C}(\hat{\bar{G}})^{-1} \phi(\hat{\bar{G}}) = \begin{bmatrix} -0,1432 & 0,1877 & 0,1621 \end{bmatrix}, \quad (64)$$

em que as matrizes  $\mathcal{C}(\hat{\bar{G}})$  e  $\phi(\hat{\bar{G}})$  são dadas por

$$\mathcal{C}(\hat{\bar{G}}) = \begin{bmatrix} \hat{\bar{H}} & \hat{\bar{G}} \hat{\bar{H}} & \hat{\bar{G}}^2 \hat{\bar{H}} \end{bmatrix} \quad \text{e} \quad \phi(\hat{\bar{G}}) = \hat{\bar{G}}^3 + \alpha_1 \hat{\bar{G}}^2 + \alpha_2 \hat{\bar{G}} + \alpha_3 I. \quad (65)$$

Com a matriz  $\hat{\bar{K}}$  definida, aplicamos a Equação 43 para calcular os ganhos como

$$\begin{bmatrix} \bar{K}_2 & \bar{K}_1 \end{bmatrix} = \left( \hat{\bar{K}} + \begin{bmatrix} 0 & I \end{bmatrix} \right) \begin{bmatrix} \bar{G} - I & \bar{H} \\ \bar{C} \bar{G} & \bar{C} \bar{H} \end{bmatrix}^{-1} = \begin{bmatrix} -0,8836 & 1,1394 & 0,7706 \end{bmatrix}, \quad (66)$$

$$\bar{K}_1 = 0,7706 \quad \text{e} \quad \bar{K}_2 = \begin{bmatrix} -0,8836 & 1,1394 \end{bmatrix}.$$

Por fim, em função da transformação aplicada no sistema, anti-transformamos os ganhos  $\bar{K}_1$  e  $\bar{K}_2$  por meio das relações

$$K_1 = \bar{K}_1 = 0,7706 \quad \text{e} \quad K_2 = \bar{K}_2 P^{-1} = \begin{bmatrix} 4,4341 & 0,1748 \end{bmatrix}. \quad (67)$$

### 3.5. Observador de estados

Como definido na descrição do servomecanismo, a saída é a posição do carrinho. Apesar disso, um dos estados é a velocidade e não pode ser mensurada diretamente para a realimentação de estados. Primeiro, considerando a representação com a matriz  $H$  auxiliar, verificamos a observabilidade pela matriz

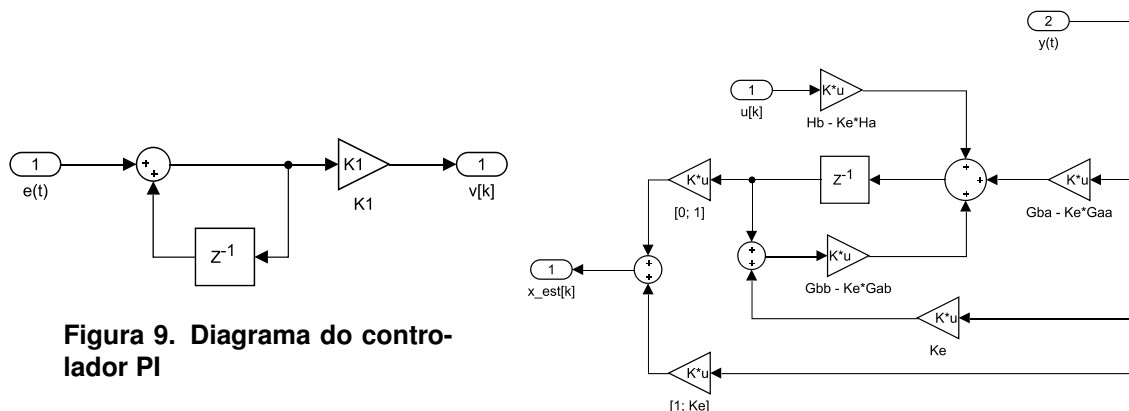
$$\mathcal{O} = [C^* \quad G^* C^*] = \begin{bmatrix} 1 & 0 \\ 1 & 0.0094 \end{bmatrix}, \quad (68)$$

que possui posto igual 2, e, portanto, o sistema é observável. A representação com a matriz  $H$  auxiliar já está na forma descrita pelas Equações 29 e, dessa forma, escolhendo o polo do observador como 0,1 (bem mais rápido que os polos da realimentação), temos que o ganho  $K_e$  é calculado como

$$K_e = \frac{G_{bb} - 0,1}{G_{ab}} = \frac{0,8836 - 0,1}{0,0094} = 83,3056. \quad (69)$$

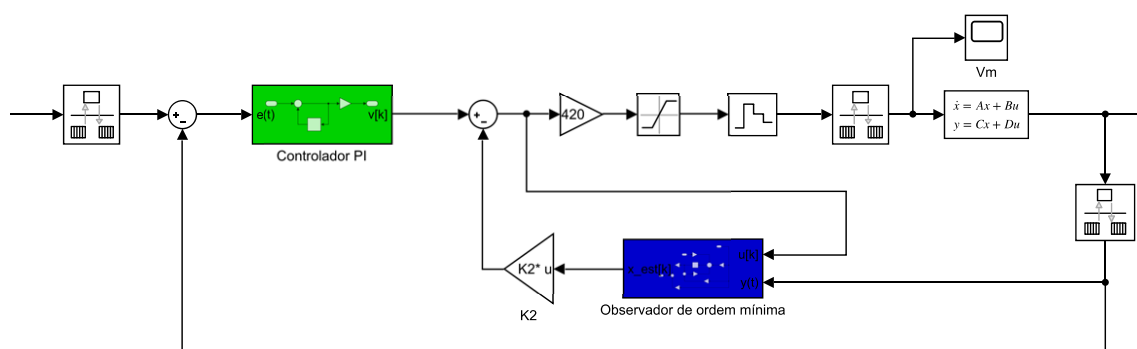
### 3.6. Implementação no *Simulink*

O sistema projetado foi implementado no *Simulink* na versão r2018a do Matlab com as mesmas configurações (passo de tempo, solucionador) do arquivo disponibilizado no laboratório. O diagrama do controlador PI, sem o ganho da realimentação de estados, foi inserido em um bloco de cor verde com uma entrada  $e(t)$  e uma saída  $v[k]$  (Figura 9). Depois, o diagrama do observador de ordem mínima foi inserido em um bloco de cor azul com entradas  $u[k]$  e  $y(t)$ , e saída  $x_{est}[k]$  (Figura 10).



**Figura 9. Diagrama do controlador PI**

**Figura 10. Diagrama do observador de ordem mínima**



**Figura 11. Diagrama de blocos do modelo teórico**

Com estes blocos implementados, montamos o diagrama de blocos do modelo

teórico apresentado na Figura 11, na qual entrada e saída estão ocultas, porque são iguais às apresentadas na Figura 6. Além da planta no espaço de estados, do segurador de ordem zero e dos elementos já utilizados na etapa de estimação dos parâmetros, também inserimos três blocos chamados *Rate Transition* para lidar com a conversão entre o passo de tempo definido para a simulação e o período de amostragem utilizado no projeto.

### 3.6.1. Adaptação do arquivo do laboratório

No arquivo disponibilizado para o laboratório, implementamos o diagrama apresentado na Figura 11 dentro de um bloco chamado **IP02\_experimental** para diminuir a quantidade de elementos visíveis no arquivo (Figura 12). Depois, adicionamos os blocos do controlador PI, do observador de ordem mínima e o ganho de realimentação conforme apresentado na Figura 13. Além disso, adicionamos um segurador de ordem zero entre a saturação e o bloco do servomecanismo para segurar o sinal e realizar a discretização do sistema.

Assim como no modelo teórico, adicionamos os blocos *Rate Transition* para tratar das mudanças no período utilizado entre os blocos. Por fim, adicionamos blocos para capturar os valores do sinal de controle, dos estados e da saída para análise dos resultados obtidos. Fora as modificações já citadas, não foram realizadas alterações no arquivo original ou na configurações da simulação.

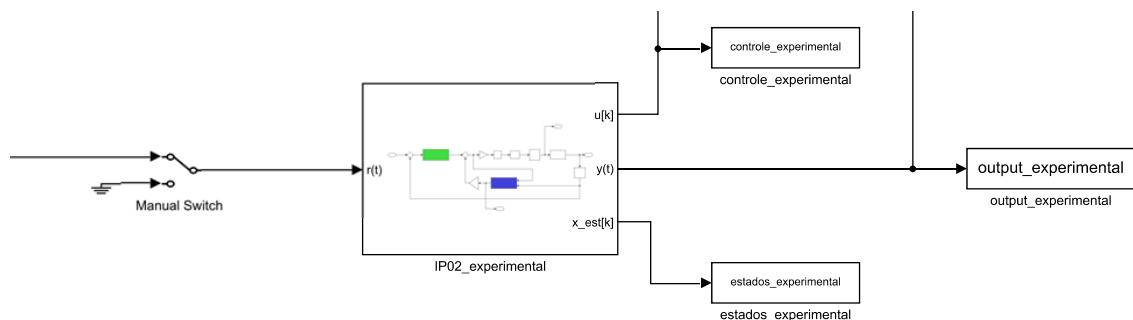


Figura 12. Diagrama com o bloco IP02\_experimental

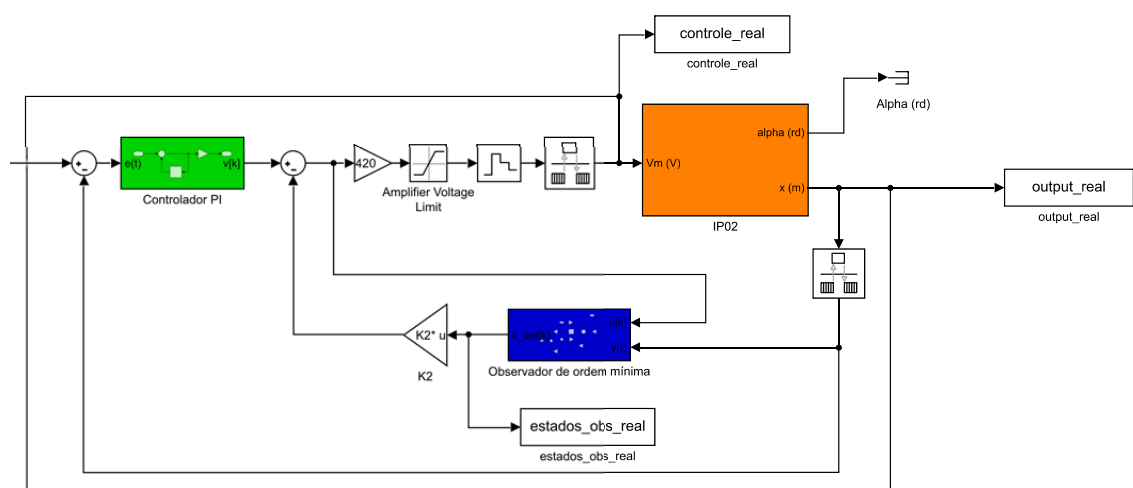


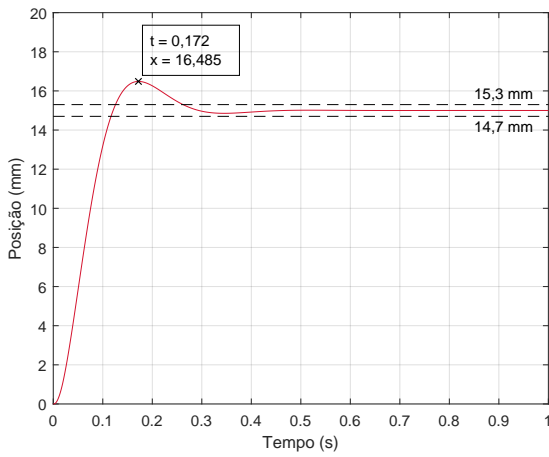
Figura 13. Diagrama de blocos do servomecanismo

## 4. Resultados

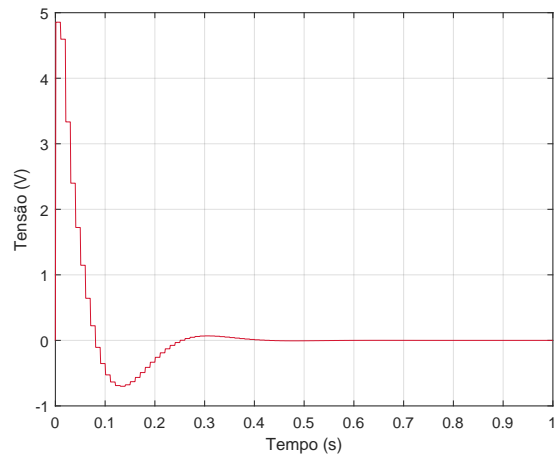
### 4.1. Resultados simulados

#### 4.1.1. Resposta ao degrau

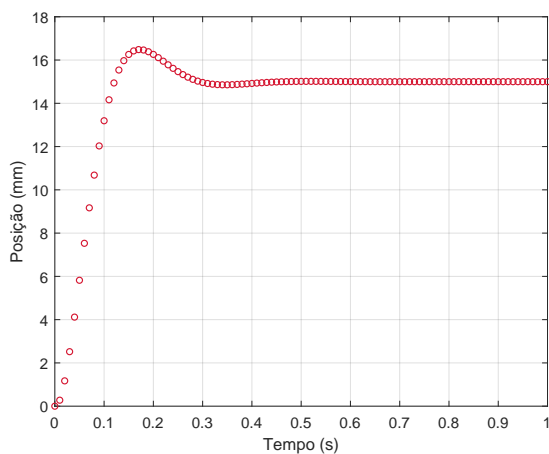
Para verificar o comportamento do sistema projetado, simulamos, novamente, a resposta ao degrau de amplitude 15V utilizado nas etapas anteriores. As Figuras 14 e 15 apresentam a resposta e o sinal de controle observados. Comparando a resposta obtida com a resposta original (Figura 7), é possível observar que tanto o sobressinal quanto o tempo de acomodação reduziram. A partir do gráfico, é possível extrair o máximo sobressinal e o tempo de acomodação como 9,9% e 0,26s aproximadamente. A diferença em relação aos valores de projeto pode ser atribuída ao período de amostragem.



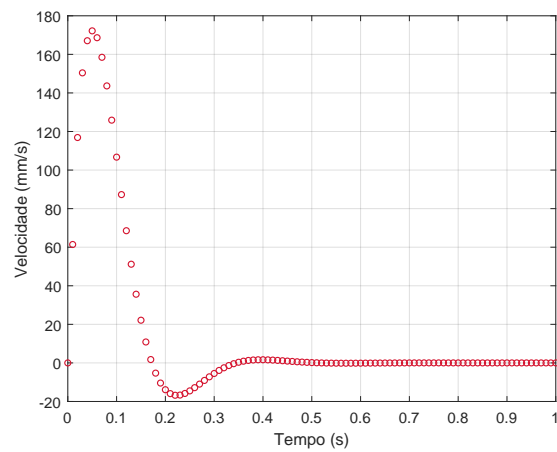
**Figura 14. Resposta ao degrau no modelo teórico**



**Figura 15. Ação de controle no modelo teórico**



**Figura 16. Estado  $x_1$  (posição) no modelo teórico**



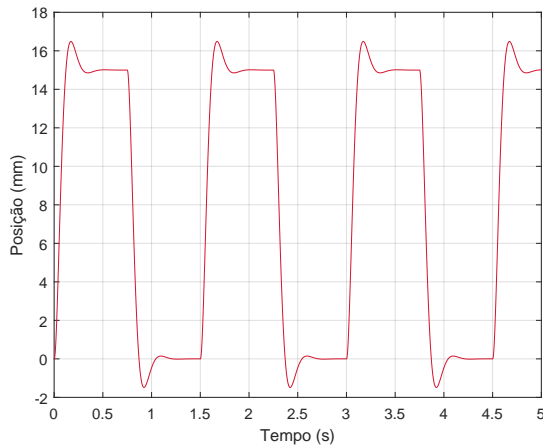
**Figura 17. Estado  $x_2$  (velocidade) no modelo teórico**

Além disso, comparando a ação de controle simulada com a ação de controle observada para o modelo experimental utilizado na caracterização do sistema (Figura 8), temos que a ação desejada possui menor magnitude e respeita as margens definidas no

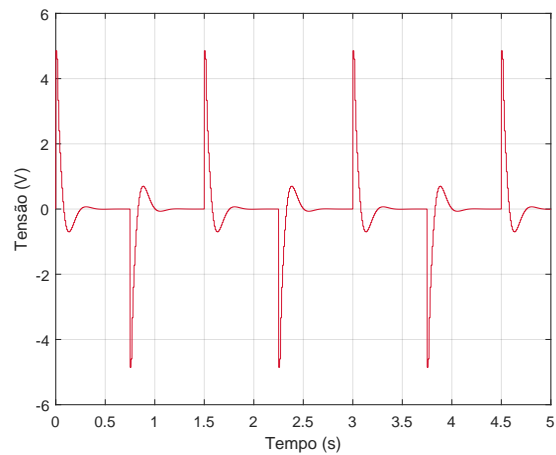
saturador (12V e -12V) colocado antes da planta. Por fim, nas Figuras 16 e 17 temos os estados  $x_1$  e  $x_2$ . É possível observar que, como a saída e o estado  $x_1$  são a posição do carrinho, as duas curvas são idênticas. Depois, para o estado  $x_2$  (estimado pelo observador), temos a curva da velocidade.

#### 4.1.2. Resposta a onda quadrada

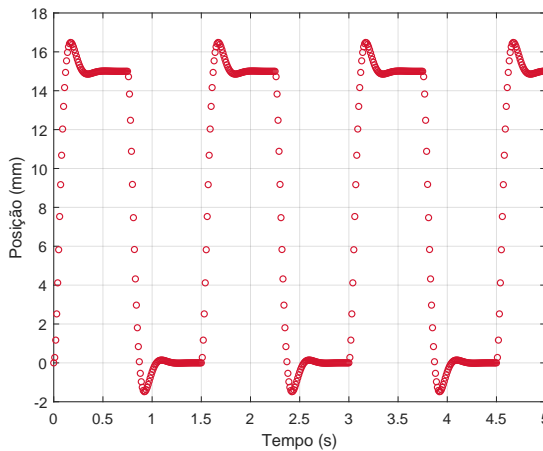
Depois, verificamos o comportamento do sistema projeto para uma entrada de onda quadrada com 15V de amplitude, período de 1,5 segundos e ciclo de trabalho 50% ao longo de 5 segundos. É possível notar, pelas Figuras 18 e 19, que o comportamento é idêntico ao observado na entrada degrau. De fato, como o tempo de ciclo é maior que o tempo de acomodação, o formato da saída é igual ao projetado para o degrau para ciclos ativos e espelhado sobre o eixo  $x$  para ciclos inativos.



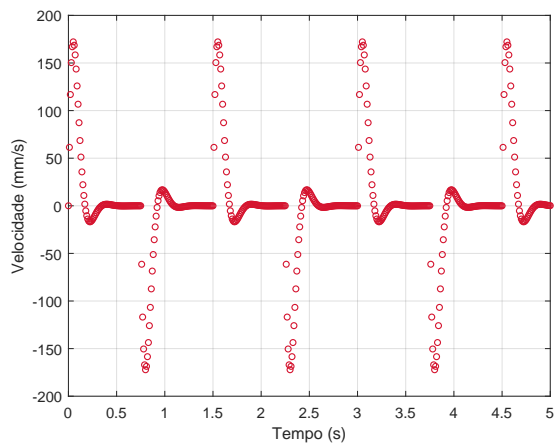
**Figura 18. Resposta a onda quadrada no modelo teórico**



**Figura 19. Ação de controle no modelo teórico**



**Figura 20. Estado  $x_1$  (posição) no modelo teórico**



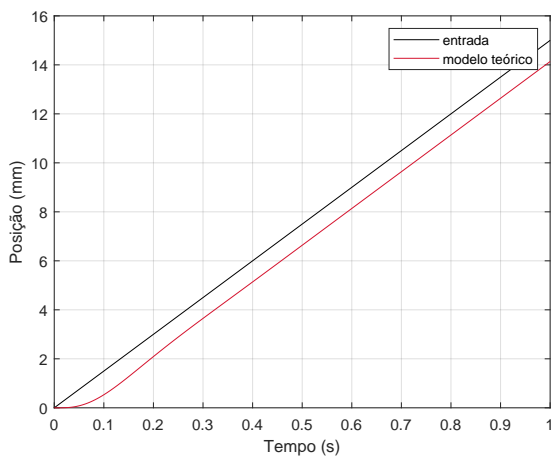
**Figura 21. Estado  $x_2$  (velocidade) no modelo teórico**

Se alterarmos o valores do tempo de ciclo, modificando o ciclo de trabalho ou

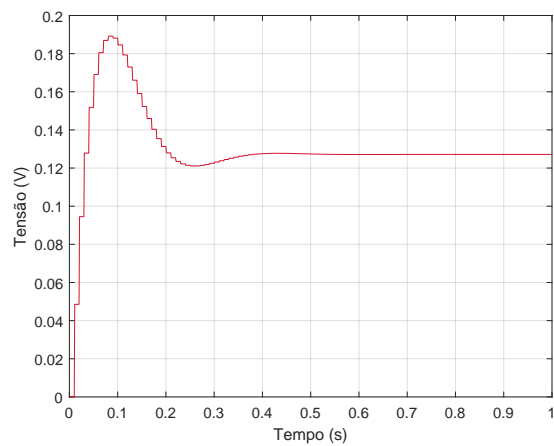
o período, é possível modificar a resposta obtida. Par ciclos de trabalho menor que o tempo de acomodação, a resposta obtida não ficará estável na referência antes da próxima troca. Assim como já observado para saída e ação de controle, os estados  $x_1$  e  $x_2$  também apresentam comportamento idêntico ao observado na resposta ao degrau dentro de um determinado ciclo.

#### 4.1.3. Resposta à rampa

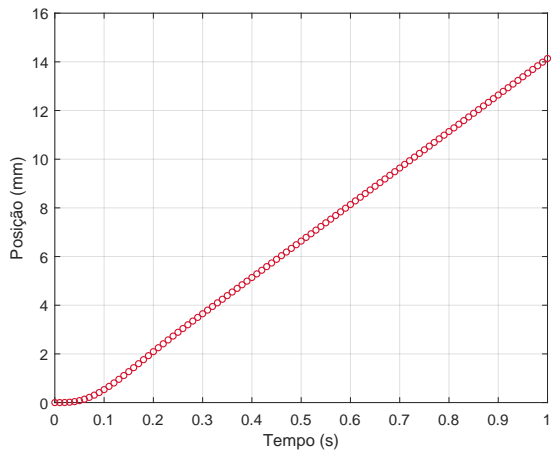
Por fim, verificamos o comportamento do modelo para uma entrada rampa com um ganho de 15V. Pela Figura 22, é possível observar que, apesar de apresentar erro estacionário nulo para entrada degrau, o modelo apresenta erro de velocidade diferente de zero. Em 1 segundo a saída está em aproximadamente 14 mm, então o erro apresentado é aproximadamente 1 mm. Por fim, vale destacar o comportamento da velocidade estimada pelo observador, apresentado na Figura 25. Inicialmente, esta sobe rapidamente, mas, depois, se estabiliza em 15 mm/s para seguir a entrada com erro constante.



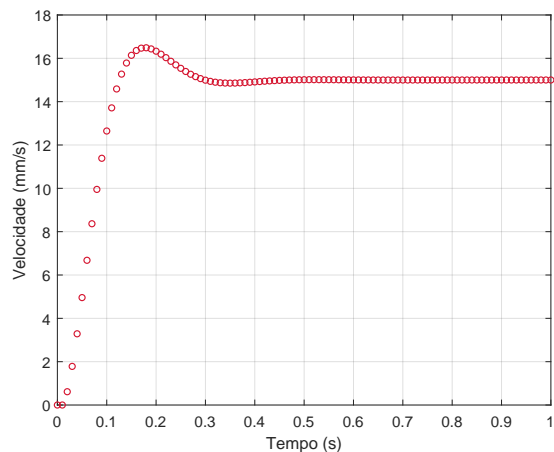
**Figura 22. Resposta à rampa no modelo teórico**



**Figura 23. Ação de controle no modelo teórico**



**Figura 24. Estado  $x_1$  (posição) no modelo teórico**

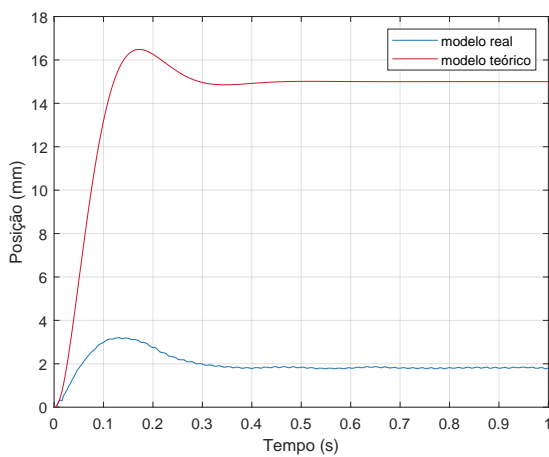


**Figura 25. Estado  $x_2$  (velocidade) no modelo teórico**

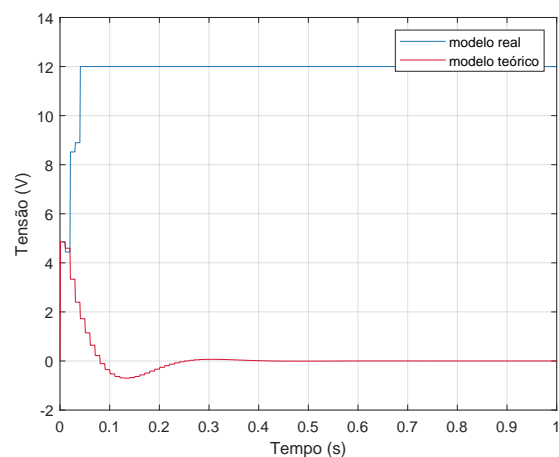


## 4.2. Resultados experimentais

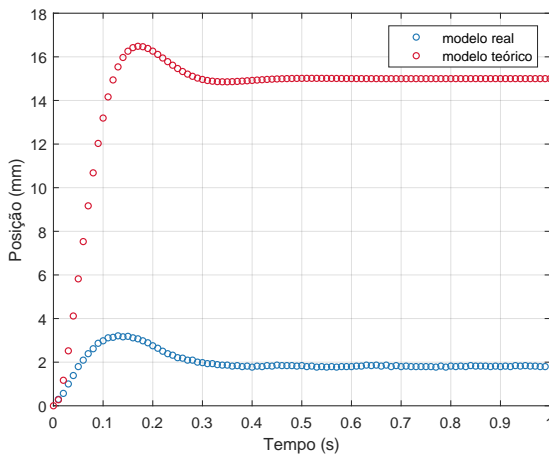
Em laboratório, utilizando o servomecanismo, repetimos a entrada ao degrau com amplitude 15V. As Figuras 26 e 27 apresentam a resposta e o sinal de controle observados para os modelos real e teórico (projetado). Pela curva da primeira figura, é possível observar que a resposta obtida pelo modelo real durante o teste, apesar de apresentar um formato semelhante, foi diferente da resposta esperada. Enquanto isso, pela curva da ação de controle, temos que o sinal satura em 12V logo no começo da simulação.



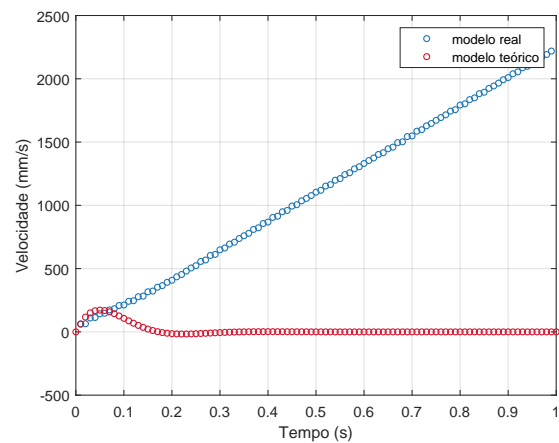
**Figura 26. Resposta ao degrau nos modelos real (azul) e teórico (vermelho)**



**Figura 27. Ação de controle nos modelos real (azul) e teórico (vermelho)**



**Figura 28. Estado  $x_1$  (posição) nos modelos real (azul) e teórico (vermelho)**



**Figura 29. Estado  $x_2$  (velocidade) nos modelos real (azul) e teórico (vermelho)**

Além disso, mesmo com o sinal de controle saturado em valores positivos, a curva de resposta apresentou uma oscilação. Este resultado parece sugerir que, durante o teste, houve um problema com a realimentação da leitura da posição do carrinho. O problema de realimentação também pode explicar o resultado apresentado na Figura 29 para o estado  $x_2$ , no qual, mesmo com a posição já estabilizada, a velocidade estimada continua a

aumentar. Em função deste resultado, não realizamos testes no sistema real com outras entradas.

Outros três testes foram realizados com o servomecanismo e a entrada degrau, mas o resultados obtidos foram igualmente díspares ao apresentado neste relatório. Em todos os casos, as características apontadas no resultado discutido também estavam presentes, como saturação da ação de controle e aumento da velocidade estimada mesmo com a posição estável. Como o sistema simulado apresenta comportamento adequado, o resultado obtido não parece ser um problema do projeto de controle, mas implementação no sistema físico em função, talvez, da falta de familiaridade com o equipamento.

## 5. Conclusão

Ao longo deste trabalho, projetamos um controlador PI digital por meio da técnica de espaço de estados para o controle de posição de ser o servomecanismo translacional IP02 da Quanser. Ao longo da seção 3, detalhamos todas as etapas da proposta de solução desenvolvida, desde caracterização do sistema, passando pela discretização e definição dos requisitos de projeto, até chegar no projeto da realimentação e do observador de estados de ordem mínima. Em relação à resposta do controlador projetado, obtivemos bons resultados em ambiente simulado para entradas degrau, onda quadrada e rampa.

Apesar disso, em laboratório, usando o sistema físico do servomecanismo IP02, não conseguimos obter resultados adequados. O resultado obtido e discutido neste trabalho sugere um problema na realimentação da posição para o controlador. Com mais tempo disponível, a estratégia para melhoria dos resultados começa com uma investigação da resposta obtida para identificar a causa do erro, seja a falta de familiaridade com o equipamento ou então um problema na montagem do sistema físico.

## Referências

- Ferreira, H. C. Notas de aula de controle digital, 2023a. Disciplina: Controle Digital, UnB.
- Ferreira, H. C. Laboratório de Controle Digital - Controle de posição de servomecanismo, 2023b. Disciplina: Controle Digital, UnB.
- Ogata, K. *Discrete-Time Control Systems*. Prentice-Hall International, 2ª edição, 1995. ISBN 0133286428.
- Ogata, K. *Engenharia de Controle Moderno*. Pearson Education, São Paulo, 5ª edição, 2010. ISBN 9788543013756.
- Yamamoto, Y. Control systems are ubiquitous, 2022. Disponível em: <http://ieeecss.org/control-systems-are-ubiquitous> – acesso em 04 de fev. de 2023.

## APÊNDICE A - Modelagem do servomecanismo

A Figura 30 apresenta um servomecanismo composto por um controlador proporcional e elementos de carga (elementos de inércia e atrito viscoso) [Ogata, 2010], em que se deseja controlar a posição da saída  $c$  a partir da posição de entrada  $r$ . A equação dos elementos de carga é dada por

$$J\ddot{c} + B\dot{c} = T, \quad (70)$$

em que  $T$  é o torque produzido pelo controlador proporcional. Então, pela transformada de Laplace e considerando condições iniciais nulas, temos

$$Js^2C(s) + BsC(s) = T(s). \quad (71)$$

Então, a função de transferência entre  $C(s)$  e  $T(s)$  é dada por

$$\frac{C(s)}{T(s)} = \frac{1}{s(Js + B)}. \quad (72)$$

Se compararmos a função obtida com a apresentada na seção 3.1, é possível relacionar os parâmetros  $a$  e  $b$  com os parâmetros  $J$  e  $B$  como

$$a = \frac{B}{J} \quad \text{e} \quad b = \frac{1}{J}. \quad (73)$$

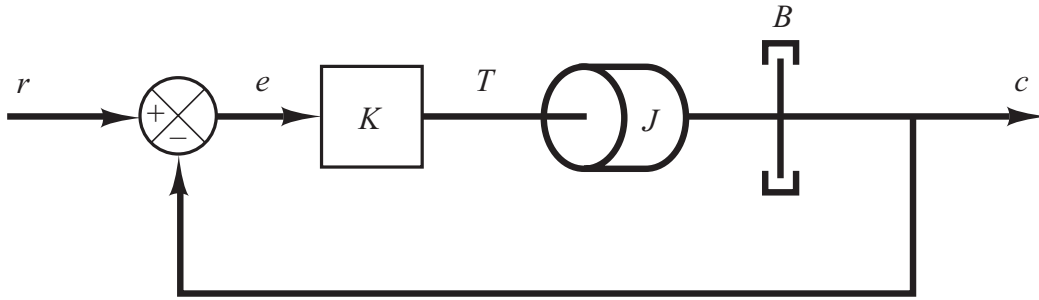


Figura 30. Servomecanismo - [Ogata, 2010, p. 150]

## APÊNDICE B - Códigos do projeto

### B.1 - Caracterização do modelo

Este código implementa uma função chamada **fit\_transfer\_function**, que estima os parâmetros  $a$  e  $b$  do servomecanismo modelado a partir dos dados de saída de uma resposta ao degrau. O arquivo **real\_data.mat**, disponível no repositório do projeto<sup>1</sup>, apresenta a estrutura esperada para o arquivo com os dados. Os parâmetros são estimados por meio do instante de tipo (Equação 4) e máximo sobressinal (Equação 5).

Código 1. Arquivo fit\_transfer\_function.m

```
1 % Autores: Isabela Maria Pereira Cruzeiro
2 %           Matheus Teixeira de Sousa
3 %
4 % Este código calcula os parâmetros a e b a partir dos
   dados coletados
5
6 function [a, b] = fit_transfer_function()
7     % Lê os valores salvos
8     load('real_data.mat');
9     out_real = mean_output;
10
11     % Pega o valor máximo do output
12     [out_max, idx_max] = max(out_real);
13     t_max = time(idx_max);
14
15     % Calcula os polos do sistema
16     omega_d = pi/t_max;
17     sigma = -log((out_max - out_real(1000))/out_real(1000))
        /t_max;
18
19     % Calcula o formato do denominador
20     z = tf('z');
21     den = (z + sigma)^2 + (omega_d)^2;
22     aux = get(den);
23     coefs_cell = aux.Numerator();
24     coefs = coefs_cell{1};
25
26     % Calcula o valor de a e b
27     a = coefs(2);
28     b = coefs(3)/420;
29
30 end
```

<sup>1</sup>O repositório do projeto está disponível no GitHub neste link.

## B.2 - Projeto do controlador PI e do observador

Este código implementa uma função chamada **setup\_pi\_control**, que calcula as matrizes  $G$  e  $H$  do modelo discretizado, e os ganhos  $K_1$ ,  $K_2$  e  $K_e$  a partir dos valores dos parâmetros  $a$  e  $b$ , do período de amostragem  $T$ , do tempo de acomodação  $t_s$  e do máximo sobressinal  $M_p$ .

**Código 2. Arquivo setup\_pi\_control.m**

```
1 % Autores: Isabela Maria Pereira Cruzeiro
2 %           Matheus Teixeira de Sousa
3 %
4 % Este código recebe os parâmetros a e b, o período de
   amostragem e os requisitos de projeto, e retorna as
   matrizes calculadas
5
6 function [A, B, C, D, K1, K2, Ke, Gaa, Gab, Gba, Gbb, Ha,
   Hb] = setup_pi_control(a, b, T, ts, Mp)
7     % Malha aberta
8     K_ma = 420;
9     A = [0 1; 0 -a];
10    B = [0; b];
11    C = [1 0];
12    D = 0;
13
14    % Espaço de estados discretizado
15    [G, H] = c2d(A, B, T);
16    H = K_ma*H;
17
18    % Calcula função de transferência
19    [num, den] = ss2tf(G, H, C, D);
20
21    % Transforma para canônica controlável
22    M = [H G*H];
23    W = [den(2) 1; 1 0];
24    P = M*W;
25
26    G_bar = (P\G)*P;
27    H_bar = P\H;
28    C_bar = C*P;
29    D_bar = D;
30
31    % Calcula os polos desejados em S
32    sigma = 4/ts;
33    omega_d = -sigma*pi/log(Mp);
34
35    modulo_z = exp(-sigma*T);
36    fase_z = omega_d*T;
37    polos = [modulo_z*cos(fase_z)+modulo_z*sin(fase_z)*1j;
```

```

38     modulo_z*cos(fase_z)-modulo_z*sin(fase_z)*1j];
39
40     z = tf('z');
41     den_z = (z)*(z-polos(1))*(z-polos(2));
42     aux = get(den_z);
43     coefs_cell = aux.Numerator();
44     coefs = coefs_cell{1};
45
46     alpha1 = coefs(2);
47     alpha2 = coefs(3);
48     alpha3 = coefs(4);
49
50     % Calcula o ganho K_chapeu
51     G_chapeu = [G_bar H_bar; 0 0 0];
52     H_chapeu = [0; 0; 1];
53
54     phi_G = G_chapeu^3 + alpha1*G_chapeu^2 + alpha2*
55             G_chapeu + alpha3*eye(3);
56     K_chapeu = [0 0 1]*inv([H_chapeu G_chapeu*H_chapeu (
57             G_chapeu^2)*H_chapeu])*phi_G;
58
59     % Calcula os ganhos K2 e K1
60     N = [G_bar - eye(2) H_bar; C_bar*G_bar C_bar*H_bar];
61     K2K1 = (K_chapeu + [0 0 1])/(N);
62     K2_bar = [K2K1(1) K2K1(2)];
63     K1_bar = K2K1(3);
64
65     K2 = K2_bar/P;
66     K1 = K1_bar;
67
68     % Projeto do observador ordem mínima
69     polo_min = 0.1;
70     Gaa = G(1, 1);
71     Gab = G(1, 2);
72     Gba = G(2, 1);
73     Gbb = G(2, 2);
74     Ha = H(1, 1);
75     Hb = H(2, 1);
76     Ke = (Gbb - polo_min)/Gab;
77
78 end

```



### B.3 - Exporta parâmetros calculados

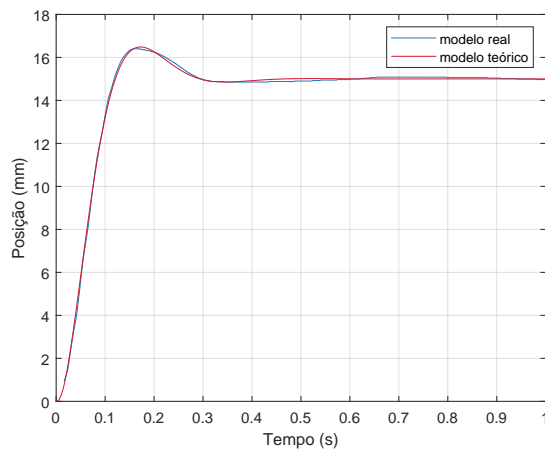
Este código implementa o código que chama as funções anteriores e salva as matrizes e parâmetros em um arquivo chamado **params\_ts03\_mp10.mat** para ser importado no laboratório.

**Código 3. Arquivo export\_params.m**

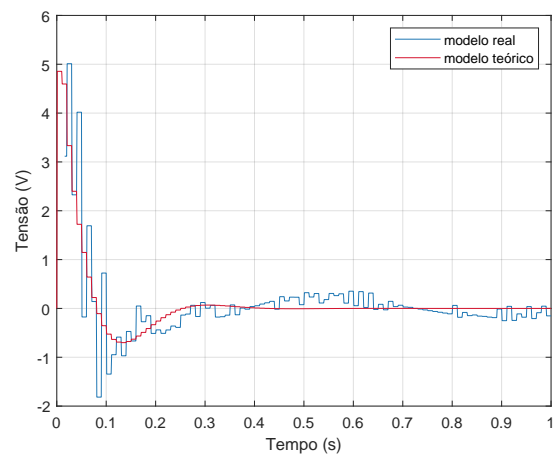
```
1 % Autores: Isabela Maria Pereira Cruzeiro
2 %           Matheus Teixeira de Sousa
3 %
4 % Este código salva os parâmetros de projeto em um arquivo
   .mat
5
6 % Estima os valores de a e b
7 [a, b] = fit_transfer_function();
8
9 % Define os requisitos de projeto
10 T = 0.01;
11 ts = 0.3;
12 Mp = 0.1;
13
14 % Calcula os ganhos do sistema
15 [A, B, C, D, K1, K2, Ke, Gaa, Gab, Gba, Gbb, Ha, Hb] =
   setup_pi_control(a, b, T, ts, Mp);
16
17 % Salva o workspace com as matrizes projetadas
18 save('params_ts03_mp10.mat')
```

## APÊNDICE C - Resultados experimentais complementares

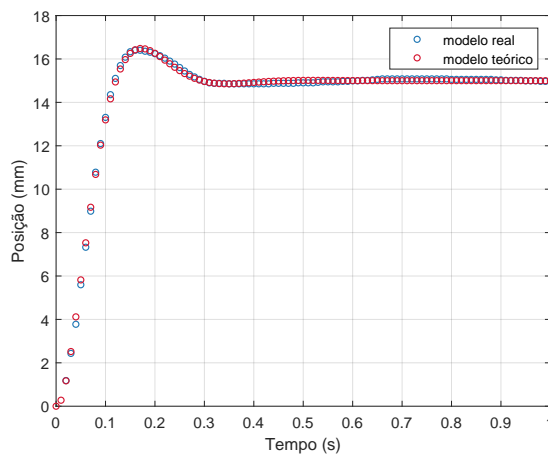
Durante a apresentação do sistema projetado, repetimos a análise para entrada de grau, cujo resultado é apresentado a seguir. Dessa vez, é possível observar que a resposta obtida foi bem mais próxima da desejada (Figura 31). Esse resultado sugere que, de fato, havia um problema no servomecanismo no dia da coleta do primeiro resultado experimental. Apesar disso, pela Figura 32, é possível observar que o sistema oscila ao redor do valor de referência. Esse resultado pode ser explicado pelo atrito entre o carrinho e o trilho, que interfere na amplitude da ação de controle na tentativa de corrigir a posição.



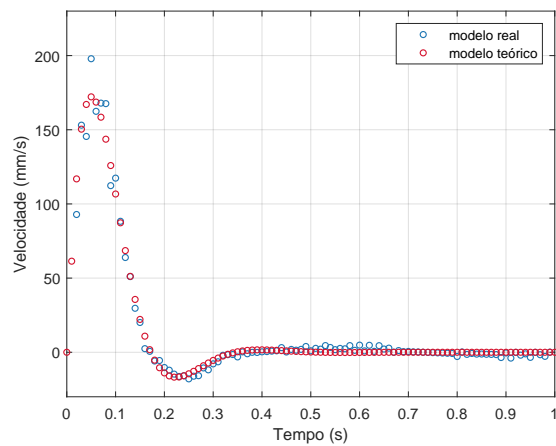
**Figura 31. Resposta ao degrau nos modelos real (azul) e teórico (vermelho)**



**Figura 32. Ação de controle nos modelos real (azul) e teórico (vermelho)**



**Figura 33. Estado  $x_1$  (posição) nos modelos real (azul) e teórico (vermelho)**



**Figura 34. Estado  $x_2$  (velocidade) nos modelos real (azul) e teórico (vermelho)**

## APÊNDICE D - Comentários pós-apresentação

O projeto do laboratório de controle digital pode ser uma das primeiras aplicações em sistemas reais das técnicas estudadas nas disciplinas de controle. Por isso, é natural que surjam dúvidas e problemas durante as etapas de planejamento, execução e análise dos resultados. Também é natural que após os comentários do professor, durante a apresentação, alguns equívocos sejam percebidos. Dessa forma, essa seção se dedica a alguns comentários e orientações sobre o projeto compilados após a apresentação.

Primeiro, em todos os laboratórios, é importante levar um *pen drive* para salvar os arquivos utilizados (**especialmente no primeiro dia**). A versão do Matlab instalada nos computadores das bancadas pode ser diferente da instalada no seu computador pessoal, no entanto, é possível exportar os arquivos do *Simulink* para a versão do laboratório pela função **Simulink.exportToVersion**<sup>2</sup>. Então, é recomendável realizar todo o projeto antes do laboratório e exportar os arquivos necessários para a versão adequada. Também é importante planejar cada aula (o que vai ser feito e em qual ordem) para aproveitar melhor cada aula.

Depois, em relação ao desenvolvimento do controlador, vale destacar duas coisas: confiança na teoria aplicada e justificativa das escolhas de projeto. Independente da bibliografia adotada, é preciso se apropriar da teoria utilizada para garantir o entendimento das etapas realizadas e a confiabilidade do método. Além disso, para cada escolha realizada, é preciso definir as justificativas correspondentes. Uma forma interessante para definir as escolhas é consultar outros projetos e bibliografias em fóruns e materiais de outras instituições.

Por fim, sobre questões práticas de projeto, o tempo de acomodação não é adequado para ser escolhido como requisito de resposta, porque o carrinho é freado pelo atrito com o trilho. Dessa forma, a resposta do sistema real é achatada, como visto na Figura 5. Alternativamente, o tempo de pico e o máximo sobressinal devem ser adotados, porque são coincidentes nos dois modelos. Além disso, os resultados reais podem ser diferentes dos esperados por diversos motivos, como problemas na montagem e não linearidades não modeladas, e cada caso deve ser investigado.

---

<sup>2</sup>A documentação dessa função está disponível neste link.