

## Simulação e implementação de um Controlador para o sistema *Ball and Beam* utilizando *Feedback Linearization*

Yan U. S. Correa\* Bruno A. Angélico\* Eduardo A. Tannuri\*\*

\* Departamento de Engenharia de Telecomunicação e Controle, Escola  
Politécnica, Universidade de São Paulo, SP (e-mail:  
yan.correa@usp.br; angelico@lac.usp.br)

\*\* Departamento de Engenharia Mecatrônica e de Sistemas Mecânicos,  
Escola Politécnica, Universidade de São Paulo, SP (e-mail:  
eduat@usp.br)

**Abstract:** This work presents the construction of a *Ball and Beam* system and the design of a nonlinear controller applied to it. This problem consists of controlling the position of a sphere that moves over a bar, keeping it at the desired set-point. Based on the system model, it was designed a *Feedback Linearization* controller capable of stabilizing the system and, thus, controlling the sphere in the desired position. The controller was firstly validated in simulation and then was embedded in a microcontroller to control the real system.

**Resumo:** Este trabalho apresenta a construção de um sistema *Ball and Beam* e o projeto de um controlador não linear aplicado a ele. Este problema consiste em controlar a posição de uma esfera que se move sobre uma barra, mantendo-a no set-point desejado. Com base no modelo do sistema, foi projetado um controlador por *Feedback Linearization* capaz de estabilizar o sistema e, assim, controlar a esfera na posição desejada. O controlador foi primeiramente validado em simulação e, em seguida, embarcado em um microcontrolador para controlar o sistema real.

**Keywords:** Feedback Linearization; Ball and Beam; Digital Control; Nonlinear control.

**Palavras-chaves:** Feedback Linearization; Ball and Beam; Controle Digital; Controle Não Linear.

### 1. INTRODUÇÃO

O *Ball and Beam*, mostrado na Figura 1, é um sistema não linear e tem sua dinâmica instável, o que o torna mais difícil de controlar. Consiste em uma esfera que se movimenta sobre uma barra, a qual tem uma extremidade presa em um eixo de rotação e, a outra, acoplada a um braço que está conectado ao servo motor. Este, por sua vez, é responsável por atuar sobre a barra e, assim, incliná-la causando o movimento da esfera. Este sistema tem dois graus de liberdade (Krishna et al. (2012)).

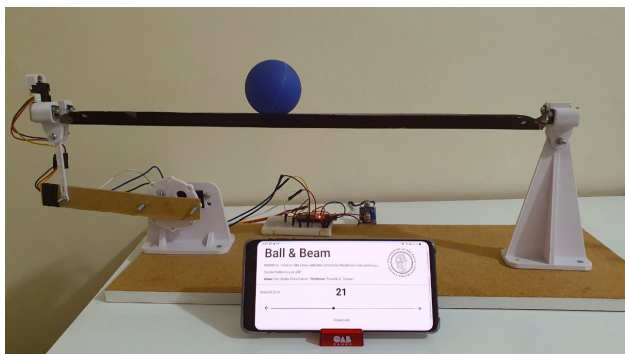


Figura 1. Planta didática do sistema *Ball and Beam*.

O objetivo é controlar a esfera na posição desejada, sendo que seu deslocamento atual é medido com um sensor de posição (Sathiyavathi and Krishnamurthy (2013)). Como a bola se movimenta continuamente e tal deslocamento não é limitado por um ângulo fixo da barra, o sistema se torna instável em malha aberta. Com isso, a função do controlador é ajustar o ângulo da barra de modo que a esfera mantenha-se na posição desejada e, caso haja distúrbios, seja capaz de reestabilizar o sistema, levando-o de volta ao ponto de interesse.

Para este modelo de sistema, muitos pesquisadores afirmam que um controle PID, apesar de sua simplicidade, apresenta uma performance satisfatória (Amjad et al. (2010)), (Rana et al. (2011)).

Porém, o *tuning* (ajuste de parâmetros) deste controlador é realizado em torno de um ponto de operação, ou seja, a planta é linearizada em torno de uma região de interesse e, assim, realizado o projeto do controlador. Dada a natureza não linear do sistema *Ball and Beam*, uma outra abordagem é projetar um controlador também não linear, como o *Feedback Linearization*.

Este trabalho está organizado da seguinte forma: na Seção 2 está presente a modelagem da planta. Em seguida, na Seção 3, é apresentada a montagem do protótipo. Na Seção 4 está presente a simulação em malha aberta. Na Seção 5

é apresentada a proposta do controlador e os requisitos de controle. Na Seção 6 projeta-se o controlador proposto que é testado em simulação na seção seguinte. Na Seção 8 são realizados os testes na planta real e, por fim, as conclusões na Seção 9.

## 2. MODELAGEM DO SISTEMA

A modelagem do sistema é uma etapa muito importante pois, além de descrever sua dinâmica, também é base para se projetar os controladores. A definição de um modelo que descreve bem o sistema em estudo pode facilitar e tornar o desempenho do controle ainda melhor. Existem vários métodos de obtê-lo como, por exemplo, através de equações matemáticas, que será o mecanismo adotado neste trabalho.

Para calcular o modelo matemático que descreve a dinâmica do sistema *Ball and Beam*, pode-se utilizar o método de Lagrange (Maalini et al. (2016)). Considera-se que a esfera rola sem escorregar e o atrito entre a ela e a barra (beam) é desprezível. As variáveis utilizadas para descrever o modelo estão listadas a seguir.

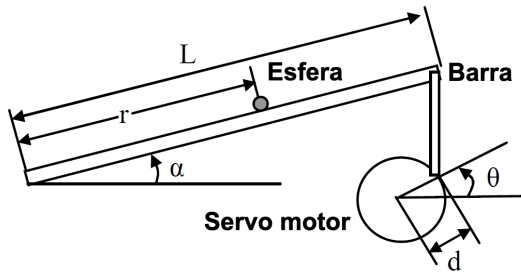


Figura 2. Esquema do sistema *Ball and Beam*.

- $m$ : massa da bola
- $R$ : raio da bola
- $J$ : momento de inércia da bola
- $L$ : tamanho da barra
- $g$ : aceleração da gravidade
- $r$ : posição da bola sobre a barra
- $\alpha$ : ângulo da barra
- $\theta$ : ângulo do servo motor
- $d$ : distância do centro da engrenagem ao ponto de fixação do braço de elevação da barra

Tendo como base o diagrama da Figura 2, a equação de Lagrange para o movimento da bola pode ser escrita como:

$$\left(\frac{J}{R^2} + m\right) \ddot{r} + mg \sin \alpha - mr \dot{\alpha}^2 = 0 \quad (1)$$

A equação que relaciona o ângulo da barra com o da engrenagem pode ser escrita, aproximadamente, como

$$\alpha = \frac{d}{L} \theta \quad (2)$$

Fazendo  $x_1 = r$  e  $x_2 = \dot{r}$  tem-se:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} \dot{r} \\ \ddot{r} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{mr \dot{\alpha}^2 - mg \sin(\alpha)}{\frac{J}{R^2} + m} \end{bmatrix} \quad (3)$$

Note que, neste caso,  $\theta$  é a variável de controle, ou seja, é o ângulo do braço do servo motor. Portanto a relação dada pela Equação (2) deve ser considerada durante a implementação do controle.

## 3. CONFECÇÃO DO PROTÓTIPO

Para este trabalho, foi confeccionado um protótipo onde pudesse ser programado o controle desejado e, assim, observar na prática a eficiência do controlador. Para isso, para a estrutura, foram utilizados materiais de baixo custo e de fácil acesso, a fim de facilitar o processo de montagem e reduzir o investimento financeiro.

Primeiramente, foi utilizada uma chapa de madeira *MDF*, com dimensões de 505 mm × 180 mm × 15 mm, que serviu como base da planta, onde todas as peças foram fixadas. Além disso, uma pequena haste de *MDF*, com dimensões de 175 mm × 25 mm × 3 mm, foi acoplada à engrenagem, a fim de aumentar a distância  $d$ , que se refere à distância entre o centro da engrenagem e o ponto de fixação do braço de elevação da barra.

Foram modeladas algumas peças em *3D*, com o intuito de facilitar a fabricação dos suportes e, ao mesmo tempo, aumentar a precisão dos recortes e furos de fixação do sensor de distância, do servo motor e da barra. A Figura 3 mostra as peças no software de impressão, prontas para serem produzidas.

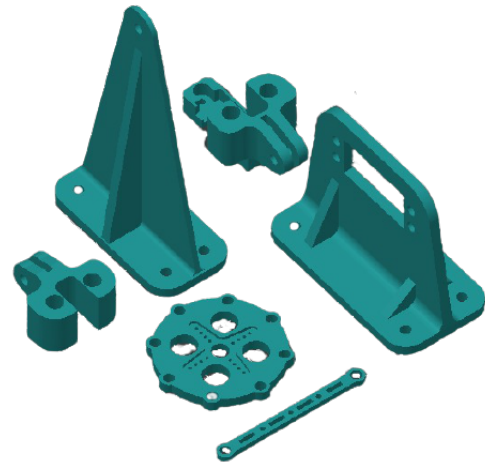


Figura 3. Peças 3D prontas para impressão.

Para a barra, foi utilizada uma canaleta de alumínio de 1mm de espessura, dobrada em *V*. Seu formato permite que a bola percorra toda sua extensão sem cair para os lados.

O servo motor escolhido foi o modelo *MG996R* de 180 graus (Figura 4). Este modelo é facilmente encontrado no mercado brasileiro e é muito utilizado em aeromodelismo de grande escala, devido à sua rápida resposta e seu torque, que é de cerca de 15kg-cm quando alimentado com 6V.

Para aferir a posição da bola, foi utilizado um sensor infravermelho da *SHARP*, modelo *GP2Y0A21YK0F*, capaz de medir a distância de objetos na faixa de 10 a 80cm (Figura 5).



Figura 4. Servo motor MG996R.



Figura 5. Sensor de distância SHARP GP2Y0A21K0F.

As partes em *MDF*, as peças *3D*, a canaleta de alumínio, o servo motor e o sensor de distância foram fixados com parafusos de tamanhos variados.

O microcontrolador escolhido foi o *ESP32* (Figura 6), uma evolução do *ESP8266*, porém com maior poder de processamento e com bluetooth BLE 4.2 integrado. Este microcontrolador é facilmente encontrado no mercado e seu preço é bem acessível. Além disso, conta com um clock de  $240\text{MHz}$  e possui diversas portas de propósito geral (*GPIOs*). A tensão de operação (nível lógico) de suas portas é de  $3,3\text{V}$ , porém a placa pode ser alimentada com até  $12\text{V}$  pois possui um regulador de tensão interno (*AMS1117*).

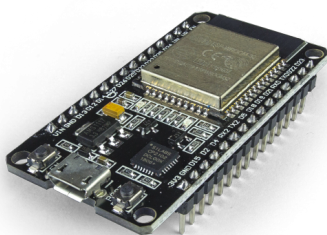


Figura 6. Microcontrolador ESP32.

Dito isso e, considerando a corrente consumida pelo servo motor que pode chegar em até  $1\text{A}$ , foi utilizada uma fonte chaveada de  $12\text{V}$  e  $2,5\text{A}$  para alimentar a planta. Além disso, o sensor de distância precisa ser ligado em  $5\text{V}$  e, como o *ESP32* não possui saída regulada nesta tensão, foi utilizado um regulador de tensão *DC-DC* ajustável, *LM2596 HW-411*. Todas as ligações foram feitas utilizando fios e *jumpers* variados. O esquemático do circuito eletrônico é mostrado na Figura 7.

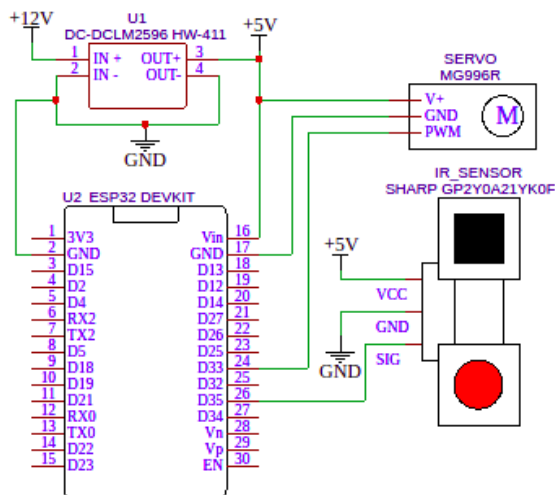


Figura 7. Esquemático do circuito eletrônico da planta.

#### 4. SIMULAÇÃO EM MALHA ABERTA

Primeiramente, conhecendo as dimensões da planta, massa e raio da bola e sabendo que o momento de inércia de uma esfera maciça é dado pela Equação (4), substituiu-se os valores em (3), resultando em (5).

$$J = \frac{2 \cdot m \cdot 0,025^2}{5} = 0,4 \cdot 0,102 \cdot 0,025^2 = 2,55 \cdot 10^{-5} \quad (4)$$

Logo, os parâmetros do modelo, considerando o protótipo desenvolvido, são dados por:

- $m = 0,102$
- $R = 0,025$
- $J = 2,55 \cdot 10^{-5}$
- $L = 0,48$
- $d = 0,115$
- $g = -9,8$
- $\alpha = \frac{d}{L}\theta = 0,2396 \cdot \theta$

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} \dot{r} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{0,102x_1\dot{\alpha}^2 - 0,102(-9,8)\text{sen}(\alpha)}{\frac{2,55 \cdot 10^{-5}}{0,025^2} + 0,102} \end{bmatrix} \quad (5)$$

Realizou-se a simulação do sistema em malha aberta, isto é, sem ação de controle, considerando  $x_1 = x_2 = 0$  como condições iniciais e impondo  $\theta = 1$ , pela Equação (2) tem-se que  $\alpha = \frac{d}{L}\theta = \frac{0,115}{0,48} = 0,2396$ . A Figura 8 apresenta as variáveis  $x_1 = r$  e  $x_2 = \dot{r}$  que são a posição e a velocidade da bola, respectivamente.

Nota-se que o sistema é instável em malha aberta, fazendo necessário o uso de um controlador para estabilizá-lo.

#### 5. PROPOSTA DE CONTROLADOR E REQUISITOS DE CONTROLE

Com a finalidade de estabilizar a planta e impor certa dinâmica à resposta, propõe-se a utilização de um controle pela técnica *Feedback Linearization*.

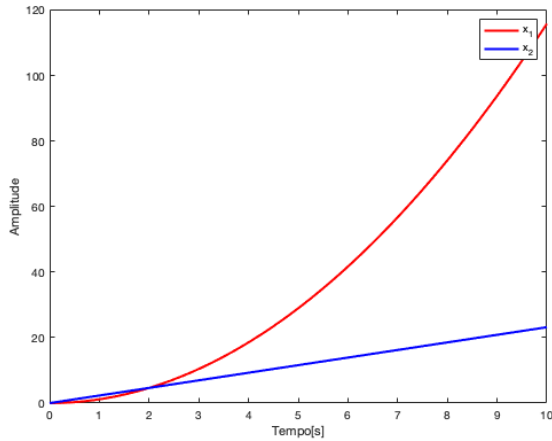


Figura 8. Simulação da planta em malha aberta.

O *Feedback Linearization* pode ser entendido como um método de projeto de controle não linear. A ideia é transformar um sistema não linear em um totalmente ou parcialmente linear e, então, projetar o controlador utilizando métodos convencionais de projeto de controladores lineares (Slotine et al. (1991)), como alocação de polos. Tal técnica equivale a cancelar as não linearidades de um sistema não linear, de modo que a dinâmica de malha fechada apresente uma forma linear.

Logo, tendo um sistema na forma canônica de controlabilidade, ou seja

$$\ddot{x}^{(n)} = f(\tilde{x}) + b(\tilde{x})u \quad (6)$$

onde  $\tilde{x} = [x, \dot{x}, \ddot{x}, \dots, x^{(n-1)}]$ , tem-se o controlador na forma

$$u = \frac{1}{b(\tilde{x})}[v(t) - f(\tilde{x})] \quad (7)$$

que cancela a parcela não linear do sistema, deixando apenas a dinâmica de  $v(t) = -K_0\tilde{y} - K_1\dot{\tilde{y}} - \dots - K_{n-1}\tilde{y}^{(n-1)} + \tilde{y}_d^{(n)}$ , onde  $\tilde{y} = y - y_d$  e  $y_d$  é a referência.

Por fim, deseja-se projetar um controlador por *Feedback Linearization* que apresente máximo sobressinal de 15% e tempo de acomodação (2%) inferior a 7 segundos.

## 6. PROJETO DO CONTROLADOR

Primeiramente, deve-se notar que o objetivo é controlar a posição da bola sobre a barra, portanto, a variável  $r$ . Logo, tem-se que  $y = x_1 = r$ . É possível reescrever o sistema fazendo  $x_3 = \alpha$  e  $x_4 = \dot{\alpha}$ , resultando em:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} \dot{r} \\ \ddot{r} \\ \dot{\alpha} \\ \ddot{\alpha} \end{bmatrix} = \begin{bmatrix} B(x_1x_4^2 - g\sin(x_3)) \\ x_2 \\ x_4 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} u \quad (8)$$

onde  $B$  é uma constante que depende de  $m$ ,  $J$  e  $R$ , conforme apresentado em (9).

$$B = \frac{m}{\frac{J}{R^2} + m} \quad (9)$$

Porém, derivando  $y = x_1$  até expor-se  $u$ , tem-se que:

$$\ddot{y} = B(x_2x_4^2 - gx_4\cos(x_3)) + 2Bx_1x_4 \cdot u \quad (10)$$

Aplicando o controlador apresentado na Equação (7), tem-se que:

$$u = \frac{1}{2Bx_1x_4}[v(t) - B(x_2x_4^2 - gx_4\cos(x_3))] \quad (11)$$

Um problema é evidente neste controlador. Note que o grau relativo do sistema é 3, uma vez que chegou-se à terceira derivada para aparecer a variável  $u$ . Por outro lado, existe uma singularidade em  $x_1x_4 = 0$  pois, quando tal fato ocorre, a parcela que multiplica  $u$  desaparece e o grau relativo do sistema não é bem definido (Zhang and Fernandez-Rodriguez (2006)).

Para tal, pode-se utilizar determinadas técnicas a fim de se obter uma aproximação do sistema como, por exemplo, eliminar a parcela da aceleração centrífuga (Hauser et al. (1992)). Pode-se também derivar  $y = x_1$  mais vezes até que apareça  $\dot{u}$ , gerando um compensador com dinâmica que implica em uma equação diferencial de  $u$  (Zhang and Fernandez-Rodriguez (2006)).

Desse modo, remove-se a parcela da aceleração centrífuga ( $mr\dot{\theta}^2$ ) presente no sistema, fazendo com que o termo  $x_1x_4^2$  desapareça da derivada segunda de  $r$ . Logo, pode-se escrever o sistema como:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} \dot{r} \\ \ddot{r} \\ \dot{\alpha} \\ \ddot{\alpha} \end{bmatrix} = \begin{bmatrix} x_2 \\ -Bg\sin(x_3) \\ x_4 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} u \quad (12)$$

Portanto, aplicando *Feedback Linearization* neste sistema, tem-se que:

$$\begin{aligned} y &= x_1 \\ \dot{y} &= \dot{x}_1 = x_2 \\ \ddot{y} &= \ddot{x}_2 = -Bg\sin(x_3) \\ \dot{\ddot{y}} &= -Bgx_4\cos(x_3) \\ \dot{\dot{\ddot{y}}} &= Bgx_4^2\sin(x_3) - Bg\cos(x_3) \cdot u \end{aligned} \quad (13)$$

Logo, pode-se escrever o controlador como:

$$u = \frac{1}{-Bg\cos(x_3)}[v(t) - Bgx_4^2\sin(x_3)] \quad (14)$$

onde  $b(\tilde{x}) = -Bg\cos(x_3)$  e  $f(\tilde{x}) = Bgx_4^2\sin(x_3)$ .

Por fim, para atender aos requisitos de controle, escolhe-se os polos de malha fechada em  $-0,5714 \pm j0,7796$ . Com isso, calcula-se  $v(t)$  que é dado por:

$$v(t) = -0,9344\tilde{y} - 1,143\dot{\tilde{y}} \quad (15)$$

## 7. SIMULAÇÃO DA PLANTA

Para verificar a eficiência do controlador proposto, foi realizada a simulação do sistema. Para tal, aplicou-se uma

referência em degrau em  $t = 0s$ , de amplitude  $30cm = 0,3m$ . Foram consideradas condições iniciais nulas e a simulação perdurou por 30 segundos. Ademais, adiciona-se uma saturação em  $\theta$  de  $\pm 60^\circ$  devido às limitações mecânicas do protótipo. A Figura 9 apresenta o gráfico dos estados  $x_1$  e  $x_2$ , respectivamente posição e velocidade da bola, e o ângulo do braço do servo motor ( $\theta$ ), em graus.

Pode-se notar que o controlador foi capaz de estabilizar o sistema e, bem como respeitar os requisitos de controle.

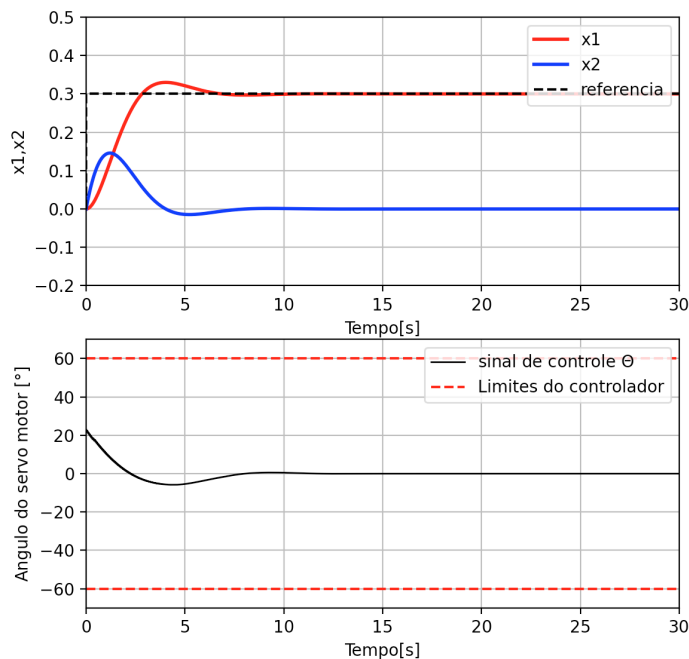


Figura 9. Simulação da planta com controlador considerando os limites mecânicos do protótipo.

## 8. TESTES NA PLANTA REAL

Através do *MATLAB* e *Simulink*, é possível gerar o código fonte do microcontrolador através de um bloco com as entradas e saídas do controlador, o que facilita o processo de testes na planta real e, além disso, é possível plotar em tempo real os gráficos da planta como referência, saída do sistema e sinal de controle.

Todavia, esta função está disponível apenas para alguns dispositivos, como o *Arduino*. O *Simulink* não possui, no momento da criação deste trabalho, suporte para o *ESP32* utilizado na confecção do protótipo. Logo, o código é implementado manualmente e a coleta de dados é feita utilizando a comunicação serial, onde os gráficos são, posteriormente, plotados com o auxílio do *MATLAB*.

### 8.1 Medição da Distância

O sensor de distância adotado possui uma curva de resposta não linear. Após a montagem da planta real foi efetuada uma calibração no sensor, realizando-se diversas medições e, através de uma interpolação, encontrou-se um polinômio de ordem 4 adequado à curva. A resolução do conversor A/D do *ESP32* é de 12 bits e, por isso, as medições foram normalizadas por um fator de  $1/100$  a fim

de não causar problema de *overflow* para números grandes elevados à terceira ou quarta potência.

A equação que relaciona o valor de  $x = ADC/100$  com a posição da bola sobre a barra, em centímetros, é dada por

$$r_{cm}(x) = 66,7 - 6,4x + 0,282x^2 - 0,00587x^3 + 0,0000461x^4 \quad (16)$$

### 8.2 Interface para atuar no setpoint

Visando facilitar a operação, foi desenvolvido um aplicativo para *smartphone* que possibilita alterar o *setpoint* da planta. A conexão do aplicativo com o *ESP32* é feita através de *bluetooth*, já embutido na placa. A Figura 10 mostra a tela do aplicativo.

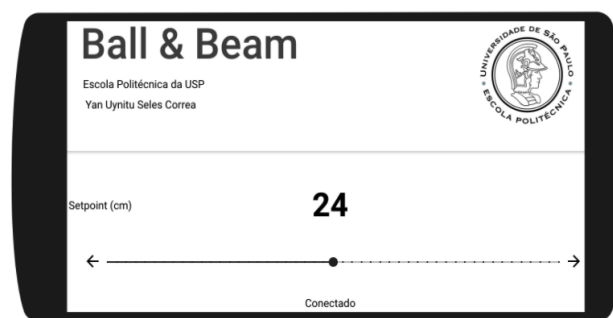


Figura 10. Aplicativo para *smartphone* para escolha de *setpoint*.

### 8.3 Resultados Práticos

Para aplicar o controle por *Feedback Linearization*, o seguinte algoritmo foi implementado:

- Medir  $r$  (distância da bola)
- Calcular  $\ddot{y}$
- Calcular  $\dot{y}$
- Calcular  $v$
- Atuar no servo motor ( $\theta$ )

Considerou-se uma frequência de 1 kHz, suficientemente inferior à menor constante de tempo do sistema. Por conseguinte, configurou-se uma interrupção no *timer 1* do *ESP32* que, a cada  $T_s = 0,001$  segundos, realiza a rotina do algoritmo descrito. A Figura 11 mostra o resultado do controle na planta real, com uma condição inicial aproximada  $(x_1, x_2) \approx (0,0)$  e tendo como referência  $y = x_1 = 0,3m = 30cm$ .

Observa-se que, apesar do controlador ser capaz de estabilizar o sistema, o sinal de controle oscila excessivamente. Tal fato é pronunciado, em grande parte pela presença de ruído de medição (sensor de distância), o que prejudica o cálculo do erro  $\tilde{y}$  e, conseqüentemente, de  $\dot{\tilde{y}}$  e do sinal de controle.

Na tentativa de suavizar o sinal de controle adiciona-se um filtro passa baixa de primeira ordem na medição da distância, em outras palavras, no sensor *IR*, conforme mostrado na Equação (17), onde  $\alpha = 0,8$ .



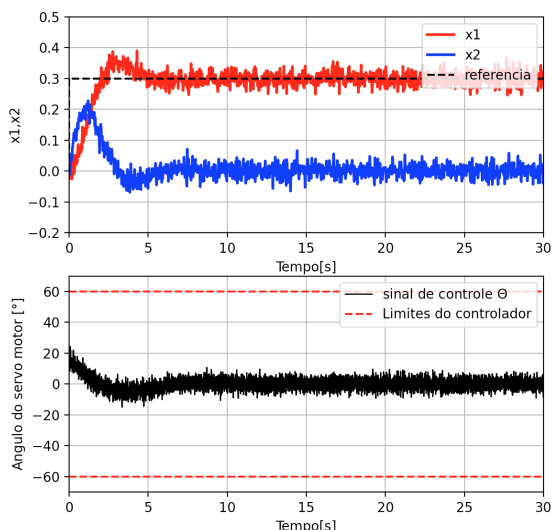


Figura 11. Teste do controlador na planta real.

$$y[n] = \alpha x[n] + (1 - \alpha)y[n - 1] \quad (17)$$

A Figura 12 apresenta a leitura da posição da bola, fixada a 10cm do sensor, antes e depois da utilização do filtro.

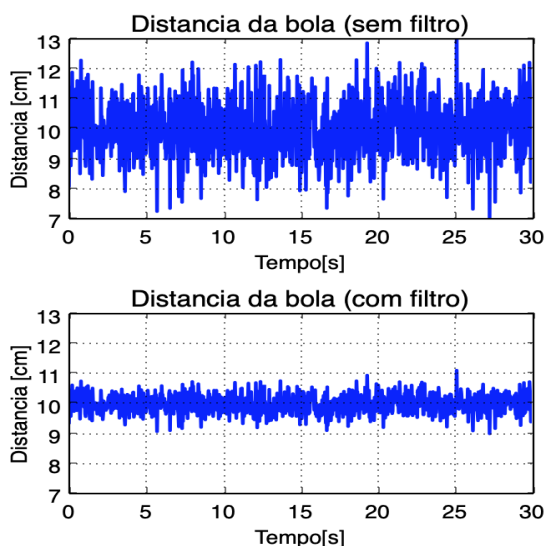


Figura 12. Distância da bola (com e sem filtro).

Pode-se notar que, mesmo com a bola parada a uma distância fixa do sensor, a medição da distância variava em  $\approx \pm 2\text{cm}$ . Já com a utilização do filtro obteve-se uma melhora significativa no sensoreamento da variável de interesse. A Figura 13 mostra o teste no sistema real após o acoplamento do filtro mencionado.

É evidente que o sinal de controle apresentou ruído quando comparado ao controlador anterior. Um fato a se notar é que o sistema real apresentou uma dinâmica levemente distinta do sistema simulado. Isso se dá pela utilização do filtro passa-baixas e por fenômenos não considerados no projeto do controlador como a malha de controle interna do servomotor e folgas mecânicas. Além disso, o método utilizado propõe que se conheça perfeitamente as variáveis da planta com precisão (massa da bola, raio, tamanho da barra etc), o que não é verdade uma vez que tais

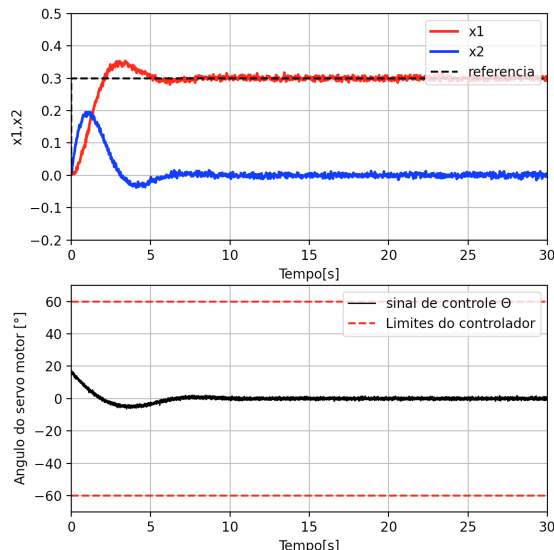


Figura 13. Planta real com uso de filtro passa baixa na medição da distância.

medições foram realizadas utilizando régua e balanças onde a medição não pode ser considerada exata.

Quanto aos requisitos de controle, observa-se que o overshoot ficou abaixo dos 15% e que o tempo de acomodação ( $t_r$ ), apesar da presença de ruído, está próximo aos 7 segundos propostos. Portanto, o resultado é satisfatório.

Um vídeo do protótipo em funcionamento pode ser visto no seguinte endereço:

<https://www.youtube.com/watch?v=tavYHRIU5PA>.

## 9. CONCLUSÃO

Durante o projeto, encarou-se os problemas encontrados em sistemas reais como, por exemplo, a presença de ruído de medição. Além disso, problemas com a confecção da estrutura da planta real foram desafiadores, como a avaria de diversas engrenagens do servo motor por não suportar o peso da estrutura, principalmente quando realizavam-se giros em sentidos opostos e com grande amplitude. Tal problema foi resolvido substituindo todas as engrenagens por suas equivalentes em metal. A barra que é percorrida pela bola contém algumas irregularidades em sua superfície que causou, em alguns pontos, certa dificuldade no movimento da esfera sobre ela, dificultando o controle.

Apesar das adversidades, o controle por *Feedback Linearization* apresentou bons resultados para o problema de estabilidade do *Ball and Beam*, validando o modelo matemático encontrado e também o projeto do controlador realizado.

A implementação do controlador no microcontrolador foi realizada utilizando as interrupções da CPU para acionar o algoritmo descrito em 8.3, a fim de impor que as iterações fossem processadas a cada  $T_s$  segundos.

Para trabalhos futuros, sugere-se a substituição do sensor de distância por um de melhor qualidade, visando a diminuição do ruído de medição. Também existe o fato de que o projeto do controlador foi realizado descartando

a aceleração centrífuga do modelo, o que não é ideal uma vez que ela existe e interfere na dinâmica do modelo. Além disso, uma abordagem para contornar o problema das incertezas do sistema é a substituição do controle atual por um controle robusto, como o *Sliding Mode*, a fim de tornar o controle tolerante aos erros provocados pelas incertezas do modelo como massa da esfera, o raio e também outros fatores advindos da modelagem.

#### REFERÊNCIAS

- Amjad, M., Ishaque, K., Abdullah, S., and Shareef, Z. (2010). Fuzzy logic control of ball and beam system. V3-489. doi:10.1109/ICETC.2010.5529494.
- Hauser, J., Sastry, S., and Kokotović, P. (1992). Nonlinear control via approximate input-output linearization: The ball and beam example. *Automatic Control, IEEE Transactions on*, 37, 392 – 398. doi:10.1109/9.119645.
- Krishna, B., Gangopadhyay, S., and George, J. (2012). Design and simulation of gain scheduling pid controller for ball and beam system.
- Maalini, P.V.M., Prabhakar, G., and Selvaperumal, S. (2016). Modelling and control of ball and beam system using pid controller. In *2016 International Conference on Advanced Communication Control and Computing Technologies (ICACCCT)*, 322–326. doi:10.1109/ICACCCT.2016.7831655.
- Rana, M.A., Usman, Z., and Shareef, Z. (2011). Automatic control of ball and beam system using particle swarm optimization. In *2011 IEEE 12th International Symposium on Computational Intelligence and Informatics (CINTI)*, 529–534. doi:10.1109/CINTI.2011.6108563.
- Sathiyavathi, S. and Krishnamurthy, K. (2013). Pid control of ball and beam system - a real time experimentation. *Journal of Scientific and Industrial Research*, 72, 481–484.
- Slotine, J.J.E., Li, W., et al. (1991). *Applied nonlinear control*, volume 199. Prentice hall Englewood Cliffs, NJ.
- Zhang, F. and Fernandez-Rodriguez, B. (2006). Feedback linearization control of systems with singularities: a ball-beam revisit. In *Proc. of the Int. Conf. on Complex Systems*, 20.