

MODELAGEM, CONTROLE E SIMULAÇÃO 3D DOS SISTEMAS BOLA-VIGA E BOLA-PRATO

DOI: 10.37702/2175-957X.COBENGE.2023.4236

Gustavo da Silva Viana - gustavo.viana@poli.ufrj.br
Universidade Federal do Rio de Janeiro

Diego Angelo Libanio - diego.alibanio@poli.ufrj.br
Universidade Federal do Rio de Janeiro

Resumo: Este projeto apresenta uma alternativa para as aulas remotas de laboratório de Sistemas de Controle realizadas na Universidade Federal do Rio de Janeiro. Para isso, é desenvolvida uma simulação tridimensional em um ambiente virtual de dois sistemas, um composto por uma bola e uma viga, e o outro por uma bola e um prato. Esses sistemas são modelados a partir da metodologia de espaço de estados, em que o controle da posição da bola sobre as superfícies é feito através de um controle por realimentação de estados, sendo os ganhos calculados por alocação de polos ou por meio de um regulador linear quadrático. A partir desse modelo, um projeto 3D dos sistemas é criado no software Matlab/Simulink, que representa a simulação em tempo real de forma dinâmica e intuitiva. Além disso, uma interface é desenvolvida para permitir que o usuário modifique as entradas do controlador dos sistemas Bola-Viga e Bola-Prato.

Palavras-chave: Simulação 3D, ensino remoto, realimentação de estados.

MODELAGEM, CONTROLE E SIMULAÇÃO 3D DOS SISTEMAS BOLA-VIGA E BOLA-PRATO

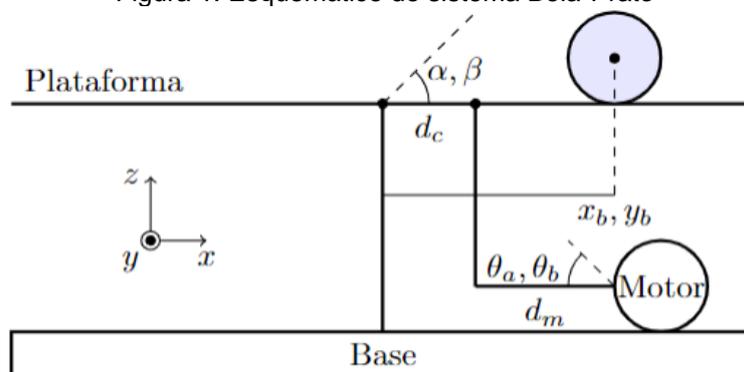
1 INTRODUÇÃO

O ano de 2020 foi marcado por uma das maiores pandemias da história da humanidade, limitando o contato social. Dessa forma, as aulas, antes administradas presencialmente, tiveram de se adaptar para transmitir o mesmo conhecimento através do ensino remoto. Nesse contexto, as matérias mais afetadas foram os laboratórios, que continham uma aplicação prática da teoria aprendida em matérias anteriores. De acordo com (LINDSAY et al., 2007) "o conhecimento deve ser construído pelo aprendiz, por meio da interação com o objeto, e não somente transmitido pelo professor".

Assim, neste trabalho é desenvolvida uma ferramenta didática para o auxílio do ensino da disciplina de "Sistema de Controle II" do curso de engenharia elétrica da UFRJ, de forma a aproximar o estudante a uma prática laboratorial de maneira interativa e visual, podendo, também ser apresentada em disciplinas teóricas como motivação aos alunos. Como o ensino remoto torna difícil a aplicação de sistemas de controle em ambientes residenciais, devido à limitação de equipamentos, a solução está presente no aparelho mais utilizado no ensino remoto, o computador. Portanto, o objetivo do trabalho está na criação de uma plataforma interativa de controle em que o estudante possa modificar parâmetros e visualizar, através de uma simulação, como um modelo 3D do projeto se comportaria.

O projeto será um modelo conhecido na literatura de controle caracterizado pelo controle de posição de uma bola sobre uma superfície. Serão abordados dois sistemas: Bola-Viga e Bola-Prato, em que o segundo, representado esquematicamente na Figura 1, pode ser entendido como uma extensão do primeiro em termos de grau de liberdade (Nise, 2012). A bola sobre uma viga unilateral pode se mover por um eixo enquanto sobre um prato pode transladar por dois eixos. Portanto, a construção do modelo Bola-Prato será desenvolvida a partir do modelo da Bola-Viga.

Figura 1: Esquemático do sistema Bola-Prato



O controle será feito por realimentação de estados, a partir do programa *Matlab*, devido à vasta biblioteca de controle e modelagem que este apresenta. Serão utilizadas as metodologias para obtenção de ganho por alocação de polos e pelo método linear quadrático (LQR), de forma a incentivar o estudante a explorar as vantagens e

desvantagens de cada método. Já o ambiente 3D será desenvolvido através de um recurso do *Matlab*, chamado *Simulink 3D Animation*. A motivação da escolha desta ferramenta está na compatibilidade entre o código desenvolvido e a simulação feita pelos dois programas. Além disso, será utilizado o construtor de diagramas de blocos do *Matlab*, o *Simulink*, que fará a conexão entre o código e a simulação. Por fim, para facilitar a interação entre a estudante e a simulação, serão desenvolvidas duas interfaces que permitirão a alteração de diversos parâmetros da simulação, uma que controlará o projeto Bola-Viga e outra, o projeto Bola-Prato.

Este projeto está organizado da seguinte forma. Na Seção 2, é apresentada a fundamentação teórica necessária para a realização do projeto. Na Seção 3, são desenvolvidos tanto a modelagem quanto o controle dos sistemas Bola-Viga e Bola-Prato. Na Seção 4, a modelagem e simulação 3D de ambos sistemas é realizada, além do desenvolvimento das duas interfaces que permitem a interação usuário-simulação. Por fim, na Seção 5, são apresentadas as conclusões do projeto.

2 FUNDAMENTOS TEÓRICOS

Esta seção introduz os conceitos necessários para o desenvolvimento do projeto. Uma vez que os projetos Bola-Viga e Bola-Prato são divididos em modelagem e controle, serão apresentados conceitos teóricos para cada etapa.

2.1 Modelagem pelo domínio do tempo

De acordo com (Nise, 2012), se o sistema for linear, as equações de estado podem ser escritas na forma vetorial-matricial através da Equação (1), em que \underline{x} é o vetor das variáveis de estado e, portanto, $\dot{\underline{x}}$ é a derivada em função do tempo dessas variáveis e \underline{u} é o vetor das entradas do sistema. De forma semelhante, a equação de saída pode ser representada pela Equação (2) em que \underline{y} é o vetor de saída.

$$\dot{\underline{x}} = A\underline{x} + B\underline{u} \quad (1)$$

$$\underline{y} = C\underline{x} + D\underline{u} \quad (2)$$

O vetor de saída \underline{y} , pode ser expresso pela solução no domínio do tempo de equações de estado, expressa na Equação (3), em função das matrizes A, B, C, D e do vetor de estado inicial \underline{x}_0 .

$$\underline{y}(t) = Ce^{At}\underline{x}_0 + C \int_0^t e^{A(t-\tau)} B\underline{u}(\tau) d\tau + D\underline{u}(t) \quad (3)$$

2.2 Equações de Euler-Lagrange

A equação de Euler-Lagrange (Arfken, 1985) pode ser definida genericamente para um sistema não conservativo conforme a Equação (4).

$$\begin{bmatrix} \dot{x} \\ \dot{x}_N \end{bmatrix} = \begin{bmatrix} (A - BK) & BK_e \\ -C & 0 \end{bmatrix} \begin{bmatrix} x \\ x_N \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} r \quad (8)$$

$$y = [C \quad 0] \begin{bmatrix} x \\ x_N \end{bmatrix} \quad (9)$$

Outro método de calcular o vetor de ganhos K é a partir do regulador quadrático linear (LQR), um controlador ótimo que minimiza uma função custo quadrática. Visando o uso de uma ferramenta computacional, é desenvolvido um método a partir do índice de desempenho (J), representado pela Equação (10), em que x é o vetor das variáveis de estado do sistema, I a matriz identidade e r o fator de ponderação de esforço, sendo positivo e definido. Note que este sistema considera uma única saída u .

$$J = \int_0^{\infty} (x^T I x + r u^2) dt \quad (10)$$

Este índice de desempenho é minimizado quando $\underline{K} = P B r^{-1}$. Por fim, o cálculo para a matriz P é enunciado pela Equação (11).

$$A^T P + P A - P B B^T P r^{-1} + I = 0 \quad (11)$$

2.4 Fórmulas de medidas de desempenho para resposta ao degrau

Considerando que uma função de transferência de segunda ordem pode ser escrita como $F(s) = k \omega_n / (s^2 + 2\zeta \omega_n s + \omega_n^2)$, em que k é o ganho do sistema ζ é o coeficiente de amortecimento e ω_n é a frequência natural, o percentual de ultrapassagem M_p (%) e tempo de acomodação t_s , são calculados nas Equações (12) e (13) (Ogata, 2010).

$$M_p(\%) = e^{-\frac{\zeta \pi}{\sqrt{1-\zeta^2}}} * 100 \quad (12)$$

$$t_s = \frac{4}{\zeta \omega_n} \quad (\text{critério de 2\%}) \quad (13)$$

3 MODELAGEM E CONTROLE

Nesta seção será realizada a modelagem dos sistemas Bola-Viga e Bola-Prato, apresentados na Seção 1, a partir das equações dinâmicas que descrevem seus respectivos comportamentos, obtidas a partir da equação de *Euler-Lagrange*. Com o comportamento dinâmico do sistema em função do tempo, um controle em malha fechada através de realimentação de estados será desenvolvido.

3.1 Modelagem dinâmica e representação em estados

A modelagem é feita inicialmente para o sistema Bola-Prato, representado na Figura 1. A energia cinética da esfera (K_b) consiste na soma da energia de translação e da energia de rotação conforme Equação (14), em que m_b e r_b são a massa e o raio da esfera, respectivamente, J_b é o momento de inércia dela, e \dot{x}_b e \dot{y}_b são as velocidades de translação da esfera. Já a energia potencial da esfera U_b é calculada de acordo com a Equação (15), em que g é a aceleração da gravidade.

$$K_b = \frac{1}{2} \left(m_b + \frac{J_b}{r_b^2} \right) (\dot{x}_b^2 + \dot{y}_b^2) \quad (14)$$

$$U_b = m_b g (x_b \left(\frac{d_m}{L_m} \theta_a \right) + y_b \left(\frac{d_m}{L_m} \theta_b \right)) \quad (15)$$

Resolvendo a Equação (4), na qual a função de Lagrange é definida pela soma das Equações (14) e (15), as forças externas Q_i^{ext} são as forças de atrito entre a esfera e a plataforma, caracterizadas por $-f_c(\dot{x}_b + \dot{y}_b)$, sendo f_c o coeficiente de atrito, e as coordenadas generalizadas são x_b , y_b , θ_a , e θ_b , obtemos as duas equações dinâmicas do sistema.

$$\left(m_b + \frac{J_b}{r_b^2} \right) \ddot{x}_b + f_c \dot{x}_b + m_b g \frac{d_m}{L_m} \theta_a = 0 \quad (16)$$

$$\left(m_b + \frac{J_b}{r_b^2} \right) \ddot{y}_b + f_c \dot{y}_b + m_b g \frac{d_m}{L_m} \theta_b = 0 \quad (17)$$

Com as Equações (16) e (17), podemos escrever as representações de estados para os quatro graus de liberdade. Serão consideradas como sendo estados do sistema as posições (x_b, y_b) e velocidades (\dot{x}_b, \dot{y}_b) da bola, enquanto as entradas serão os ângulos (θ_a, θ_b) do motor. Devido à simetria do problema, podemos escrever a equação de estados para o eixo generalizado x_b e, analogamente, escrever a equação de y_b . Portanto, a representação de estados pode ser escrita de acordo com as Equações (18) e (19).

$$\begin{bmatrix} \dot{x}_b \\ \dot{\dot{x}}_b \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & \frac{-f_c}{\left(m_b + \frac{J_b}{r_b^2} \right)} \end{bmatrix} \begin{bmatrix} x_b \\ \dot{x}_b \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{-m_b g d_m}{L_m \left(m_b + \frac{J_b}{r_b^2} \right)} \end{bmatrix} \theta_a \quad (18)$$

$$y = [1 \quad 0] \begin{bmatrix} x_b \\ \dot{x}_b \end{bmatrix} \quad (19)$$

Para o sistema Bola-Viga, temos apenas um eixo de liberdade, de forma que basta igualarmos $y_b = 0$, $\dot{y}_b = 0$, e $\theta_b = 0$ nas Equações (14) e (15). Note que a representação por espaço de estado deste sistema se resume às Equações (18) e (19).

Por fim, a Tabela 1 apresenta os parâmetros do sistema a ser controlado (Control Tutorials Umich, 2020). Note que o valor negativo da força da gravidade é devido ao sentido da coordenada Z, definido na Figura 1 (sentido contrário à força da gravidade).

Tabela 1: Parâmetros do sistema Bola-Prato.

Parâmetro	Valor
m_b	0.111 kg
r_b	0.05 m
J_b	$9.99e^{-6} \text{ kg} \cdot \text{m}^2$
d_m	0.03 m
L_m	1.3 m
f_c	0.02 Ns/m
g	-9.8 m/s^2

Substituindo os valores definidos na Equação (18) e, de forma análoga, representado a equação do eixo Y do sistema, é possível obter o sistema Bola-Prato representado pelas Equações (20) e (21).

$$\begin{aligned} \begin{bmatrix} \dot{x}_b \\ \dot{x}_b \end{bmatrix} &= \begin{bmatrix} 0 & 1 \\ 0 & -0.1739 \end{bmatrix} \begin{bmatrix} x_b \\ \dot{x}_b \end{bmatrix} + \begin{bmatrix} 0 \\ 0.2183 \end{bmatrix} \theta_a \\ y &= [1 \quad 0] \begin{bmatrix} x_b \\ \dot{x}_b \end{bmatrix} \end{aligned} \quad (20)$$

$$\begin{aligned} \begin{bmatrix} \dot{y}_b \\ \dot{y}_b \end{bmatrix} &= \begin{bmatrix} 0 & 1 \\ 0 & -0.1739 \end{bmatrix} \begin{bmatrix} y_b \\ \dot{y}_b \end{bmatrix} + \begin{bmatrix} 0 \\ 0.2183 \end{bmatrix} \theta_b \\ y &= [1 \quad 0] \begin{bmatrix} y_b \\ \dot{y}_b \end{bmatrix} \end{aligned} \quad (21)$$

3.2 Controle do sistema

O sistema será projetado $t_s \leq 3s$ e $M_p \leq 5\%$. A partir das Equações (12) e (13), encontram-se os polos dominantes do sistema. Já o terceiro polo é afastado à esquerda no plano complexo dos dominantes para que esse não influencie na resposta transitória. A Tabela 2 apresenta os polos definidos para o sistema Bola-Prato.

Tabela 2: Polos do sistema realimentado.

Polos	Valores
p_1	$-1.5 + 1.5i$
p_2	$-1.5 - 1.5i$
p_3	-7.5

De posse dos polos do sistema, utiliza-se a função "*place*(\hat{A} , \hat{B})" do *Matlab* para a definição dos ganhos, em que \hat{A} e \hat{B} são definidos de acordo com a Equação (22).

$$\begin{bmatrix} (A - BK) & BK_e \\ -C & 0 \end{bmatrix} = \begin{bmatrix} A & 0 \\ -C & 0 \end{bmatrix} - \begin{bmatrix} B \\ 0 \end{bmatrix} * [K^t \quad -K_e] = \hat{A} - \hat{B} * \hat{K} \quad (22)$$

O vetor de ganhos encontrado é $\hat{K} = [123.7 \quad 47.3 \quad -154.6]$. Portanto, obtemos os ganhos da realimentação $K = [123.7 \quad 47.3]$ e o ganho do integrador é $K_e = 154.6$.

Já para o controlador obtido a partir de um regulador linear quadrático (LQR), utilizamos o comando "*lqr*" da ferramenta *Matlab*, que possui como entrada, as matrizes A e B , conhecidas do sistema, e as matrizes Q e R , que representam, respectivamente, a ponderação ao desempenho e esforço. De forma a obter parâmetros de desempenhos similares ao controlador obtido por alocação de polos, foram definidas as matrizes Q e R de acordo com a Equação (23).

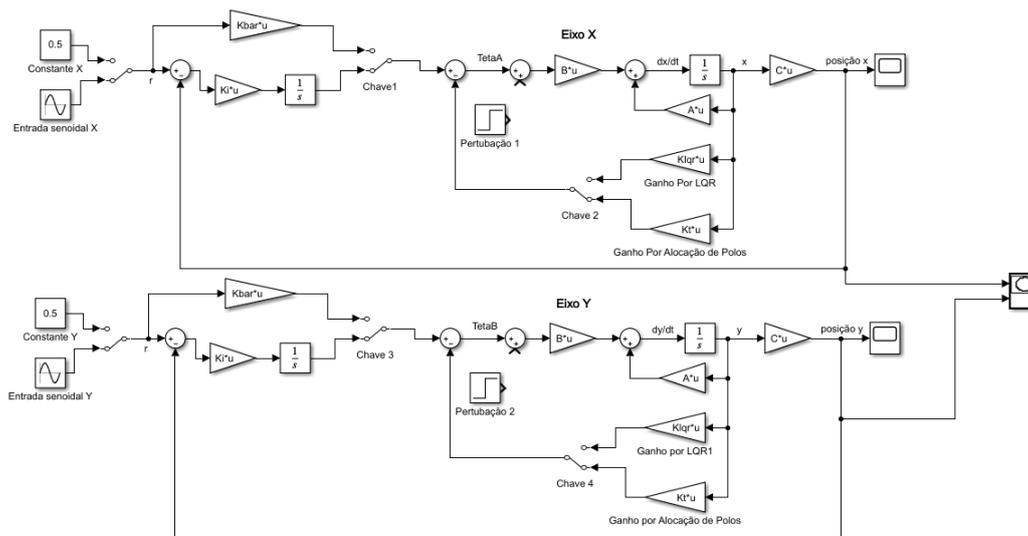
$$Q = \begin{bmatrix} 4 & 0 \\ 0 & 0.1 \end{bmatrix} \quad R = [0.01]. \quad (23)$$

O vetor de ganhos ótimos obtidos ao solucionar a Equação (11) foram $K = [20 \quad 13.1271]$. Para o sistema rastrear a saída é necessário o ganho K_e , de acordo com a Figura 3, em que $K_e = 20$. Porém, nesse projeto, o controlador obtido pela metodologia LQR, não é robusto, ou seja, não adicionamos um integrador em série ao ganho K_e , de forma que este não rejeitará possíveis perturbações atribuídas ao sistema. Esta decisão é

proposital, com objetivo de demonstrar ao estudante os problemas que um controlador não robusto pode apresentar.

Por fim, através da ferramenta *Simulink*, é construído o diagrama de blocos para o sistema apresentado, representado na Figura 4, utilizando as duas abordagens de modelagem de controle realizadas nesta seção. Note que foi atribuída ao *Simulink* uma chave mecânica, de forma que o usuário possa escolher qual método para obtenção de ganho deseja utilizar.

Figura 4: Modelo do projeto de controle no software Simulink



4 SIMULAÇÃO E INTERATIVIDADE COM USUÁRIO

Esta seção aborda o desenvolvimento da simulação 3D dos modelos de controle da Bola-Viga e Bola-Prato, a partir da ferramenta *Simulink 3D Animations* (S3DA). O S3DA é uma extensão do programa *Matlab*, que possibilita ao usuário a criação de ambientes virtuais que possam ser conectados a sistemas do *Simulink*. Por exemplo, se o usuário possui a trajetória de um objeto em função dos 3 eixos cartesianos (x, y, z), é possível atribuir tal movimento a um objeto criado no S3DA e acompanhar seu movimento enquanto o projeto no *Simulink* estiver em execução.

Por padrão, o eixo x corresponde ao eixo horizontal, o eixo y corresponde ao eixo vertical e o eixo z corresponde à profundidade. Em termos de sinal, no caso do eixo x , o valor positivo corresponde à direita, do eixo y o sentido é para cima e do eixo z é saindo do computador. Já eixos de rotação são definidos pela regra da mão direita, definidos por um vetor $[x \ y \ z \ angle]$. Ou seja, as três primeiras informações correspondem ao sentido de rotação e a quarta à amplitude do sinal em radianos.

4.1 Construção dos modelos 3D

Para a construção dos modelos 3D que irão compor os sistemas Bola-Viga e Bola-Prato, utilizamos a biblioteca "*Components*" do *Simulink 3D Animations*. Esta biblioteca possui diversos objetos pré-definidos, que podem ser editados, em tamanho e rotação, para representar a componente desejada. Além disso, a ferramenta permite adicionar texturas aos objetos criados, de forma a deixar a simulação mais vívida e familiar ao usuário.

Primeiro, é modelado o sistema Bola-Viga. A viga é criada através de 5 "boxes" que corresponderão à base, à lateral direita, à lateral esquerda, à parte frontal e à parte traseira

desta. Respeitando o projeto, a base viga possui um comprimento de 1.3m e 0.15m de profundidade (para ser maior que a bola), enquanto a altura das bordas da viga mede 0.06m, dessa forma, não bloqueando a visualização da bola. Cada parte da viga possui uma espessura de 0.01m. Já para a localização de cada componente, basta alocar de forma lógica, considerando que a base da viga está presente na origem (0,0,0). Ou seja, a lateral direita estará localizada em (0.65, 0.03, 0), a esquerda em (-0.65, 0.03, 0), a frontal em (0, 0.03, 0.075) e a traseira em (0, 0.03, -0.075).

Já para a bola, selecionamos a componente "Sphere" e definimos seu raio como 0.05m, de acordo com o projeto. Sua posição é (0, 0.06, 0), pois a base da viga tem espessura 0.01m. A partir disso, é modelado o conector que liga o motor girante à base da viga e permitirá a rotação desta. Portanto, escolhemos um componente cilíndrico que possui altura de 0.3m e raio de 0.03m, de forma a ficar posicionado verticalmente. Para permitir a rotação da viga, é necessário posicioná-lo fora da origem do eixo x , portanto, transladando o conector para (0.5, 0.04, 0), em que a altura correspondente serve para encostar o objeto na base da viga.

A parte fixa do motor será representada por um cubo e a parte girante, por uma esfera. A parte fixa possui aresta de valor 0.26m e seu centro se localiza em (0.29, -0.11, 0). A parte girante possui raio de 0.115m para cobrir o conector durante a movimentação deste, e sua localização está em (0.47, -0.11, 0). Note que a diferença de posição x dos centros do conector e da parte girante do motor equivalem a 0.03m, ou seja, o valor do braço de alavanca (d_m) descrito no projeto.

Por fim, adiciona-se uma base para representar o sistema em uma superfície plana. Esta será um bloco de espessura 0.01m e com comprimento e profundidade equivalente a 3m, sendo localizada abaixo da parte fixa do motor, ou seja, (0, -0.24, 0). A Figura 5 representa o esboço do sistema bola-viga. Além disso, texturas são adicionadas a cada elemento do sistema para tornar a simulação mais real. Para a base da viga, foi criada uma régua de cor marrom, tal que o usuário pode visualizar melhor a posição que a bola está. Dessa forma, a Figura 6 representa a modelagem 3D final da Bola-Viga.

Figura 5: Esboço do sistema Bola-Viga.

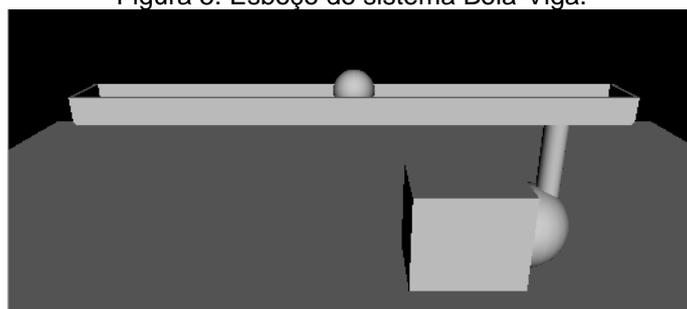


Figura 6: Modelagem 3D Bola-Viga final.



Construído o modelo 3D, a ferramenta ainda permite definir e salvar pontos de vista ("Viewpoints") específicos para serem acessados pelo usuário. Portanto, são criados 5 pontos de vista diferentes de forma a oferecer ao usuário uma liberdade e imersão na simulação, permitindo observar a movimentação do sistema de diferentes ângulos.

O modelo Bola-Prato é construído de maneira similar. A base plana do novo sistema pode ser construída através da base da viga, modificando seu tamanho (*geometry*) de (1.3, 0.01, 0.15) para (1.3, 0.01, 1.3). Para melhor visualização do usuário da posição da bola durante sua trajetória, foi desenhada uma extensão da régua utilizada como textura na base da viga. Nesse desenho são enumeradas as posições dos eixos x e y e também sinalizadas, com traços mais leves, as posições intermediárias. Para maior precisão, foi calculado o tamanho real do prato (1.32m considerando a espessura de ambos os lados) e relacionado com o tamanho em pixels da textura. Dessa forma, foi computado que 1 metro equivale a 44 pixels da textura a ser aplicada.

Para completar o modelo, basta adicionar o novo conjugado motor-conector que irá proporcionar a rotação no ângulo β do prato. Para tornar o desenho mais simétrico, é projetado este conjunto perpendicularmente ao primeiro. Dessa forma, a posição da parte fixa do motor é (0, -0.11, -0.29), da parte girante é (0, -0.11, -0.47) e do conector (0, 0.04, -0.5). Note que essa rotação foi feita invertendo os valores de posição x com z , em que o sinal negativo agora representa que este conjugado estará na parte traseira do prato. Os tamanhos e texturas desses componentes permanecem inalterados. Por fim, são criadas, 5 diferentes pontos de vista, para permitir ao usuário uma melhor visualização do projeto. A Figura 7 ilustra dois pontos de vistas criados, no qual é possível observar a textura quadriculada criada para representar a posição da bola no plano.

Figura 7: Pontos de vista Bola-Prato.



4.2 Animação dos modelos 3D

Após finalizar a modelagem 3D do sistema Bola-Viga e Bola-Prato, podemos atribuir as entradas que realizarão a animação. A partir do diagrama de blocos da Figura 5, é possível notar que seis variáveis são obtidas diretamente do sistema: as posições da bola x_b e y_b , as velocidades da bola \dot{x}_b e \dot{y}_b , e os ângulos de rotação do motor θ_a e θ_b .

Considere como exemplo a translação da bola no sistema Bola-Viga. Os movimentos de translação possuem uma matriz entrada $[x \ y \ z]$, em que x é o resultado da simulação feita no *Simulink*. Já a posição y pode ser calculada através do ângulo α da viga e da posição x da bola. Como o diagrama de blocos do *Simulink* fornece o ângulo θ_a , em que, $\alpha = d_m \theta_a / L_m$, podemos expressar a altura da bola pela Equação (24).

$$y = x * \operatorname{tg} \left(-\frac{d_m}{L_m} \theta_a \right) + 0.25 \quad (24)$$

O sinal negativo, presente na Equação (24), corresponde ao sentido de rotação do modelo 3D criado, enquanto a constante "0.25" compensa o deslocamento criado no momento de posicionar a bola no ambiente virtual. Vale ressaltar que é necessário sempre lembrar o posicionamento tridimensional dos objetos, pois o S3DA sempre considera como centro de translação a origem (0,0,0) do ambiente virtual.

Agora considerando a movimentação da bola no sistema Bola-Prato teremos três graus de liberdade: a translação pelo eixo x , a translação do eixo y (coordenada z no S3DA) e a altura referente ao deslocamento do prato. Os dois primeiros podem ser obtidos diretamente do diagrama de blocos do *Simulink*. Já a altura da bola pode ser calculada pela soma das alturas resultantes de cada eixo, que, adaptando da Equação (24), pode ser expressa pela Equação (25).

$$y = x * \operatorname{tg} \left(-\frac{d_m}{L_m} \theta_a \right) + z * \operatorname{tg} \left(-\frac{d_m}{L_m} \theta_b \right) + 0.25 \quad (25)$$

Portanto, é possível computar a movimentação das componentes do sistema a partir das variáveis fornecidas do *Simulink* e da Equação (25). Assim, obtemos os movimentos de translação para a bola e conectores, que ligam os motores girantes à base da viga, além dos movimentos de rotação do prato, viga e motores.

4.3 Interface simulação-usuário.

A interação entre usuário e simulação é uma etapa fundamental, pois está diretamente conectada à parte didática do projeto. O software *Matlab* possui uma ferramenta chamada *AppDesigner*, que possibilita o desenvolvimento de interfaces compatíveis com outros recursos do programa.

A interface de controle do sistema Bola-Viga será dividida em três partes. Na primeira seção são selecionados dois "Knobs" que permitirão o ajuste numérico dos ganhos, além de um "Switch" que seleciona qual metodologia de obtenção de ganhos será usada pela simulação. Na segunda seção, adicionam-se três "Sliders", que controlarão o valor da entrada constante, assim como a amplitude e frequência da entrada senoidal. Inclui-se também um "Rocker Switch", que permite a transição entre os tipos de entradas, além de dois "Edit Fields" que indicarão ao usuário o valor numérico da entrada selecionada nos Sliders. A terceira e última seção será composta de apenas um "Slider", que incluirá uma perturbação constante ao sistema. A Figura 8 mostra a interface gráfica da Bola-Viga.

A interface que controlará a posição da bola sobre um prato será desenvolvida a partir da interface Bola-Viga. A ideia consiste em adicionar um eixo (no caso, o eixo y) ao controle de posições e perturbações. A Figura 9 apresenta a interface Bola-Prato. O botão *Reset* leva os parâmetros do sistema aos seus valores iniciais quando pressionado. Já no novo painel, chamado de "Formas Especiais", é possível selecionar cinco trajetórias diferentes para a bola percorrer no prato, estas são: círculo, elipse, infinito, quadrado, triângulo. Além disso, a interface permite ao usuário selecionar os parâmetros destas formas, como o raio da circunferência ou o tamanho da aresta do quadrado, por exemplo.

Figura 8: Interface gráfica Bola-Viga.

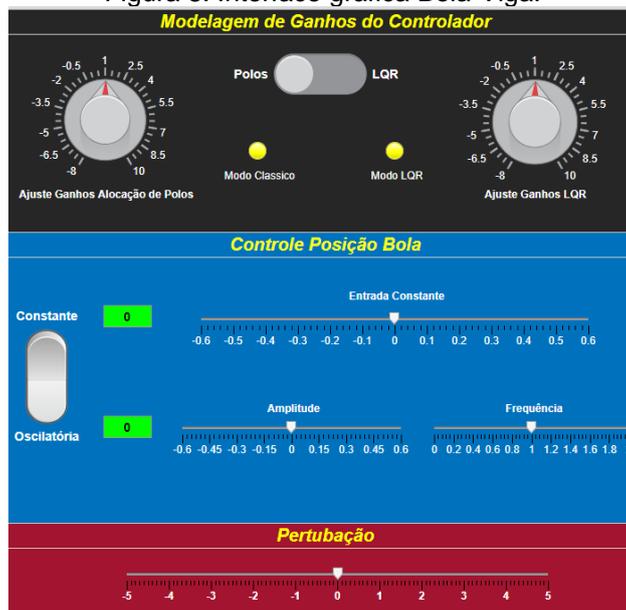
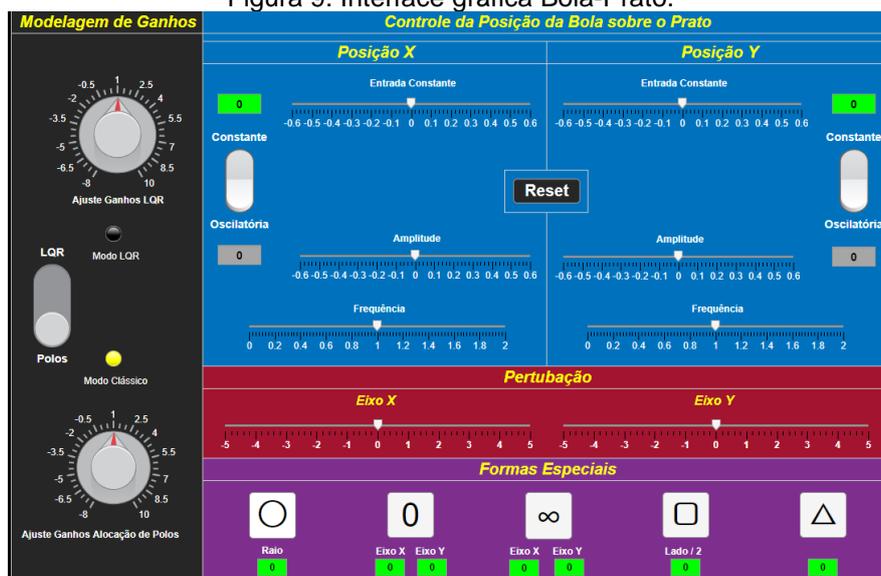


Figura 9: Interface gráfica Bola-Prato.



5 CONCLUSÃO

Este trabalho teve como objetivo principal desenvolver uma simulação 3D de um sistema Bola-Viga e Bola-Prato para ensino da disciplina de Sistemas de Controle II da UFRJ. Esta ferramenta busca contornar a dificuldade de implementação física de sistemas de controle, principalmente para disciplinas de ensino remoto. Sendo assim, os alunos puderam aprender conceitos matemáticos de controle e aplicá-los remotamente em uma planta virtual. As simulações desenvolvidas neste trabalho podem ser encontradas em https://drive.google.com/drive/folders/1PVnpbDwcQ4X1UTVUfuAQbOYnqlthDSQO?usp=s_haring.

AGRADECIMENTOS

Este trabalho foi parcialmente apoiado pela FAPERJ sob o auxílio E-26/010/002620/2019, e em parte pelo apoio financeiro do Programa de Recursos Humanos da Agência Nacional do Petróleo, Gás Natural e Biocombustíveis – PRH-ANP.

REFERÊNCIAS

Arfken, G. Mathematical Methods for Physicists. 3. Ed, Orlando, FL: Academic Press, 1985.

Lindsay, G. (2007). Educational Psychology and the Effectiveness of Inclusive Education/Mainstreaming. British Journal of Educational Psychology, n. 77, p. 1-24.

MathWorks (2015), Simulink 3D Animation User Guide. Disponível em:
<https://www.mathworks.com/help/sl3d/>. Acesso em: 30 jan. 2020.

Nise N. S., Engenharia de Sistemas de Controle. 2. Ed, São Paulo: Editora LTC 2012.

Ogata K., Modern Control Engineering. 5. ed, São Paulo: Pearson Prentice Hall, 2010.

University of Michigan, Control Tutorials. Disponível em:
<https://ctms.engin.umich.edu/CTMS/index.php?aux=Home>. Acesso em: 20 jan. 2020.

3D MODELING, CONTROL, AND SIMULATION OF BALL-BEAM AND BALL-PLATE SYSTEMS

Abstract: *This project presents a remote alternative to laboratory classes in Control Systems at the Federal University of Rio de Janeiro. This can be done by developing a three-dimensional simulation in a virtual environment of two systems, one composed by a sphere and a beam, and the other by a sphere and a plate. These systems are modeled in state space, where the control of the position of the sphere on the surfaces is done by a state feedback controller, where the gains are computed by pole allocation or using a quadratic linear regulator. From this model, a 3D design of the systems is created in Matlab/Simulink software, which represents the real-time simulation in a dynamic and intuitive way. In addition, an interface is developed to allow the user to modify the controller inputs of the Ball-Beam and Ball-Plate systems.*

Keywords: *3D simulation, feedback state controller, Remote teaching*