



SEI-SICITE 2021

Pesquisa e Extensão para o mundo em transformação

## **Ambiente para simulação de sistemas físicos com aplicação ao problema do sistema barra bola, computação visual e controle PID.**

*Environment for simulation of physical systems with application to the problem of the ball and beam system, visual computing and PID control.*

**Lucas de Freitas\***, **Daniel Cavalcanti Jeronymo†**

### **RESUMO**

Quando falamos em simulações computacionais de ambientes físicos e controle de processos, temos em mente a necessidade de integrar diferentes conhecimentos com o menor *delay* e a maior precisão possível. Ambientes virtuais também nos permitem abusar da criatividade e do *hardware* empregado, na tentativa de mostrar novas abordagens para problemas já estudados. Dentro dessas perspectivas, esse trabalho toma como áreas de interesse, a computação visual e o controle de sistemas, apresentando uma abordagem diferenciada para o controlador PID. Aplicado ao sistema barra bola e intermediado por processos de computação visual, o controlador passa por um processo de discretização por diferenças finitas *backward* e *tuning* off-line utilizando a regra da integral dupla do método SIMC. Foram escolhidos para esse trabalho, o framework de simulação robótica Pybullet e o biblioteca de computação visual, OpenCV. Para avaliação do desempenho do controlador, são apresentados quatro critérios de erro: Integral do erro quadrático (ISE) de 1,567, Integral do erro absoluto (IAE) com 0,0932, Integral do erro quadrático multiplicado pelo tempo (ITSE) com 5,53e-5 e por fim a Integral do erro absoluto multiplicado pelo tempo (ITAE) com 7,13e-4.

**Palavras-chave:** Barra-bola, Controle PID, Computação Visual, Simulação de sistemas dinâmicos.

### **ABSTRACT**

When we talk about computer simulations of physical environments and process control, we have in mind the need to integrate different knowledge with the least delay and the greatest possible precision. Virtual environments also allow us to abuse creativity and the hardware used, in an attempt to show new approaches to problems already studied. Within these perspectives, this work takes as areas of interest, visual computing and systems control, presenting a different approach to the PID controller. Applied to the “ball and beam” system and intermediated by visual computing processes, the controller goes through a backward finite difference discretization process and offline tuning using the rule of the double integral of the SIMC method, so that it is applied in the model more efficiently. The robotic simulation framework Pybullet and the visual computing library, OpenCV, were chosen for this work. To evaluate the controller's performance, four error criteria are presented: Squared error integral (ISE) of 1,567, Absolute error integral (IAE) with 0,0932, quadratic error integral multiplied by time (ITSE) with 5,53e-5 and finally the Integral of the absolute error multiplied by time (ITAE) with 7,13e-4.

**Keywords:** Ball and Beam, PID Control, Visual Computing, Dynamic systems simulation.

\* Engenharia da Computação, Universidade Tecnológica Federal do Paraná, Toledo, Paraná, Brasil; [lucasfreitas.2017@alunos.utfpr.edu.br](mailto:lucasfreitas.2017@alunos.utfpr.edu.br)

† Universidade Tecnológica Federal do Paraná, Campus Toledo; [danielc@utfpr.edu.br](mailto:danielc@utfpr.edu.br)



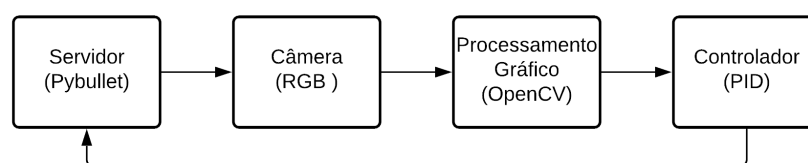
## 1 INTRODUÇÃO

Sendo a computação uma área do conhecimento relativamente dispersa, a intersecção de seus setores se torna cada vez mais necessária, uma vez que, os problemas e objetivos da engenharia se tornam mais ambiciosos. Atualmente podemos visualizar um enorme aumento de produtos, pesquisas e técnicas relacionadas à computação visual e ao processamento de imagens 3D, devido ao aumento da capacidade em processar dados visuais com uma velocidade adequada, segundo Han, Reily, Hoff, Zhang (2017). Para Azuma (1997), isso beneficia diretamente na resolução de problemas, antes agravados pela dificuldade de coletar os dados necessários para o sistema, via sensores comuns, como diagnóstico de pacientes, sistemas de controle de voo, reparo e manutenção de equipamentos, entre outros.

De forma concomitante, a aplicação da teoria de controle de sistemas, encontra em ambientes virtuais de simulação física, uma saída mais flexível e barata para a demonstração e aplicação de suas metodologias. Todorov, Erez e Tassa (2012) elencam a dificuldade de se encontrar ambientes que possam retornar a velocidade e a precisão necessária para o desenvolvimento de sistemas de controle.

Neste trabalho é apresentado um ambiente de simulação virtual tridimensional que possui aplicações de computação visual. Para sua demonstração foi montado um modelo de barra com uma esfera que se locomove no sentido longitudinal e com um eixo no meio da estrutura permitindo sua inclinação. A rotação do eixo é controlada pela ação de um motor acoplado, determinada por meio de um controlador PID parametrizado em um *tuning off-line* e discretizado por meio de diferenças finitas *backward*, com o objetivo de controlar a esfera em uma posição específica. O controlador utiliza o ângulo de inclinação da plataforma e a posição da esfera, retirados do ambiente por meio de uma câmera inserida dentro da simulação, que através de processamento gráfico, alimenta o controlador. Para a avaliação do seu desempenho é realizada uma análise dos critérios de erro ISE, IAE, ITSE, ITAE. O fluxo da simulação pode ser visto na Figura 1.

**Figura 1 – Diagrama da simulação**



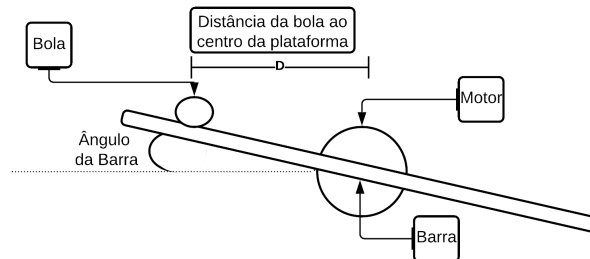
**Fonte: Autoria própria (2021)**

Este artigo está organizado na seguinte sequência: é apresentado a estrutura do modelo e a descrição do framework utilizado, os processamentos gráficos referente aos dados enviados ao controlador, descrição do controlador e do processo de *tuning*, apresentação dos resultados obtidos e considerações finais.

## 2 MODELO BARRA-BOLA E AMBIENTE DE SIMULAÇÃO

O sistema bola barra, representado na Figura 2, já conhecido de estudos na Teoria de Controle, se baseia em uma barra, onde no seu ponto central é adicionado um eixo e juntamente a ele é acoplado um motor, responsável por controlar o ângulo da barra, que por sua vez, provoca o movimento da bola.

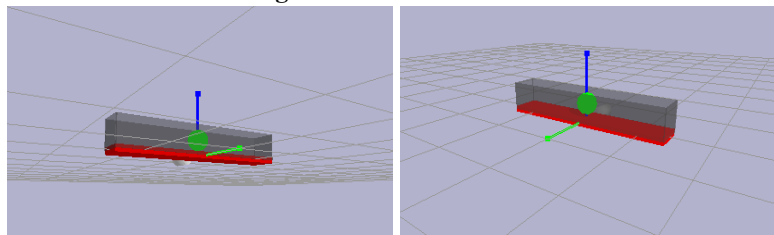
**Figura 2 – Modelo matemático do sistema Bola-Barra**



**Fonte: Autoria própria (2021)**

Este trabalho utiliza o modelo tridimensional representado na Figura 3. Criado utilizando a formatação URDF(Unified Robotic Description Format) para arquivos xml, o controle é aplicado a uma junta, do tipo “Revolução”, que realiza o papel do motor. As cores foram especialmente escolhidas para facilitar a ação do OpenCV mais a frente.

**Figura 3 – Modelo criado**



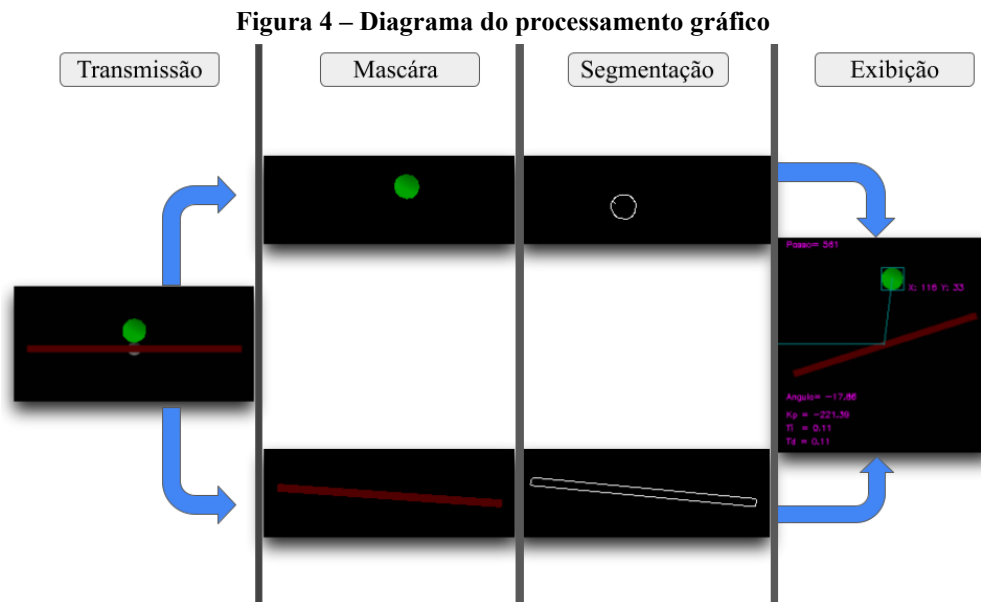
**Fonte: Autoria própria (2021)**

### 3 PROCESSAMENTO GRÁFICO

Entre a ação de controle e o servidor com o modelo, existe uma comunicação montada por meio de computação visual, utilizando o framework OpenCV.

Primeiramente foi posicionada uma câmera dentro da simulação transmitindo em tempo real o modelo, podemos ver sua imagem na etapa da **transmissão** na Figura 4. Com esse intuito, precisamos realizar a segmentação do contorno dos objetos, identificando pontos cruciais como o centro da esfera e as dimensões da plataforma. Como a câmera posicionada nos retorna uma matriz RGB, vamos realizar uma conversão para o modelo de cor HSV (Hue, Saturation, Value), transformando o processo de mapear uma cor dentro da matriz imagem, mais simples, pois podemos fixar a componente “hue” e assim controlar separadamente os níveis de saturação e luminosidade. Necessitamos retirar duas informações da simulação para a ação do controlador, o ângulo de inclinação da plataforma e a posição da bola em relação à mesma. Com esse intuito, precisamos realizar a segmentação do contorno dos objetos, identificando pontos cruciais como o centro da esfera e as dimensões da plataforma. Como a câmera posicionada nos retorna uma matriz RGB, vamos realizar uma conversão para o modelo de cor HSV (Hue, Saturation, Value), transformando o

processo de mapear uma cor dentro da matriz imagem, mais simples, pois podemos fixar a componente “hue” e assim controlar separadamente os níveis de saturação e luminosidade.



Fonte: Autoria própria (2021)

Após esse processo, são criadas duas matrizes que serão usadas como filtros, uma verde e a outra vermelha, para criação de máscaras, verificamos cada pixel da imagem e caso a cor não corresponda ao intervalo que determinamos, o pixel em questão receberá o valor zero. Para facilitar a segmentação, aplicamos o operador “bitwise and”, que compara a imagem transmitida com a máscara criada, e caso o valor do pixel correspondente em cada matriz for diferente um do outro, esse pixel é anulado. Com isso conseguimos obter duas matrizes de imagem, uma para a barra e outra para a bola, como visto da etapa de **máscara**.

Necessitamos então detectar as bordas desses objetos, para isso utilizaremos o método Canny, desenvolvido por John F. Canny em 1986 e que consta como função dentro da biblioteca OpenCV. O método se inicia com a aplicação de um filtro gaussiano na imagem no intuito de diminuir o ruído, em seguida aplicamos o Operador de Sobel para encontrar o vetor gradiente e a direção de cada pixel. Por fim, fazemos uma busca na matriz por pixels que não constituem as bordas do objeto, verificando os valores de máximos locais, caso o pixel não seja um máximo local, ele será suprimido, obtendo o resultados da etapa **segmentação**.

No caso da bola, podemos retirar da imagem com o método Canny, as coordenadas x e y de cada pixel que constitui a borda, dessa forma, se elegermos os maiores e menores valores dessas componentes, e permutarmos para obter pontos na imagem, conseguimos desenhar uma delimitação quadrada ao redor da bola. Com relação a plataforma precisamos encontrar sua orientação, para isso utilizamos PCA (**Análise de Componentes Principais**), criado em 1901 por Karl Pearson. Nesse método, conseguimos perceber o sentido da maior variação do grupo de dados presentes na imagem da barra que obtidos com o método Canny e atribuímos um eixo no sentido da maior variação, chamamos ele de “primeira componente”,



consequentemente ela representa o sentido longitudinal da plataforma, e teremos a sentido latitudinal como a segunda componente. Com esses autovetores reconhecidos, podemos aplicar a função atan2 (Arco tangente com 2 variáveis) que nos retorna o ângulo em radianos no plano Euclidiano, do raio criado pelas componentes da Primeira Componente e o eixo X no sentido positivo. Assim conseguimos reconhecer o ângulo da plataforma e traçar a distância da bola até ele. O resultado pode ser visto na etapa de **exibição**.

#### 4 CONTROLADOR PID E SINTONIZAÇÃO

Com sua vasta aplicação e desenvolvimentos a partir do modelo PID robusto e confiável para a atividade industrial após 1940 (Bennet, 2000), sua composição em série possui três parâmetros:  $K_c$  (ganho do controlador),  $\tau_i$  (tempo de integração),  $\tau_d$  (tempo de derivação). Para a aplicação de forma digital, necessitamos realizar uma discretização da equação do controlador, para que possamos utilizá-la em função dos passos da simulação, neste trabalho usaremos o método de discretização por diferenças finitas *backward*. Vamos partir da equação do controlador em série, no domínio de Laplace, representada pela Eq. (1).

$$u(s) = k \cdot \left(1 + \frac{1}{s \cdot \tau_i}\right) \cdot (1 + s \cdot \tau_d) = \frac{K_c}{\tau_i \cdot s} \cdot (\tau_i \tau_d s^2 + (\tau_i + \tau_d) \cdot s + 1) \quad (1)$$

Após a aplicação da discretização por meio de diferenças finitas *backward*, temos forma posicional do controlador:

$$u[k] = Kp \cdot \left(1 + \frac{\tau_d}{\tau_i} + \frac{\tau_d}{T_s} + \frac{T_s}{\tau_i}\right) \cdot e[k] - Kp \cdot \left(1 + \frac{\tau_d}{\tau_i} + \frac{2}{T_s}\right) \cdot e[k - 1] + Kp \cdot \frac{\tau_d}{T_s} \cdot e[k - 2] \quad (2)$$

Para o processo de sintonização, primeiro necessitamos obter a função de transferência do sistema, para isso optamos por utilizar o estimador dos mínimos quadrados não recursivo, que nos possibilita encontrar o **vetor de medidas** ( $\varphi(t)$ ) e o **vetor de parâmetros** ( $\theta(t)$ ) a partir de um conjunto de dados que mostre a relação entre a variação do ângulo da plataforma e da posição da bola.

Tendo ambos vetores definidos, tal que  $\theta(t)=[a_1, a_2 \dots a_n, b_0, b_1 \dots b_n]$ , retiramos o valor do ganho ( $k$ ), conforme expresso a seguir:

$$k = \frac{b_0}{(1+a_1)} \quad (3)$$

Por fim podemos encontrar,  $\tau_c$ ,  $\tau_i$  e  $Kp$  utilizando o método SIMC (Skogestad, 2012) pela regra da

**Integral Dupla**, como pode ser observado nas equações a seguir:

$$\tau_c = 10 \cdot T_s \quad (4)$$

$$\tau_i = \tau_d = 4 \cdot (\tau_c + T_s) \quad (5)$$

$$Kp = \frac{1}{k} \cdot \frac{1}{4 \cdot (\tau_c + T_s)^2} \quad (6)$$



## 5 RESULTADOS

Os dados apresentados são referentes a 2500 passos da simulação com o ponto de referência no centro da plataforma. A bola inicia deslocada para a direita. A Tabela 1 aponta os valores acumulados dos critérios de erro.

**Tabela 1 – Critérios de erro ao final da simulação**

<b>Critérios</b>	<b>Valores</b>
ISE	0,0932
IAE	1,567
ITSE	5,53e-5
ITAE	7,13e-4

**Fonte: Autoria própria (2021).**

## 6 CONCLUSÃO

Neste trabalho foi apresentado um ambiente para simulações didáticas, tridimensionais, com aplicação de computação visual e controle de sistemas. A nível de demonstração, foi montado um modelo de um sistema barra-bola, tridimensional, onde a posição da bola pode ser obtida por meio de um controlador PID, sintonizado por meio de SIMC.

Trabalho futuros irão se desenvolver com o intuito de melhorar a modelagem e a computação gráfica, com o objetivo de otimizar o trabalho do controlador, diminuindo a quantidade de ruído transmitida no sistema.

## REFERÊNCIAS

Azuma, Ronald T. **A Survey of Augmented Reality**, Applications, Hughes Research Laboratories 3011 Malibu Canyon Road, Malibu, CA 90265, USA, 1997.

Bennett, Stuart. **The Past of PID Controllers**, Introduction, IFAC Proceedings Volumes, Volume 33, Issue 4, Terrassa, Spain, 2000, pages 1-11, ISSN: 1474-6670.

HAN, Fei; REILY, Brian; HOFF, William; ZHANG, Hao. **Space-Time Representation of People Based on 3D Skeletal Data: A Review**, Introduction, Colorado School of Mines, Golden, CO 80401, USA, 2016.

Skogestad, Sigurd; Grimholt, Chriss. **The SIMC Method do Smooth PID Controller Tuning**, Chapter 5 in : R. Vilanova, A. Visioli (eds.), PID Control in the Third Millennium, Advances in Industrial Control, DOI 10.1007/978-1-4471-2425-2\_5, Springer-Verlag London Limited 2012.