

Ambiente para simulação de sistemas físicos com aplicação ao problema do carro com pêndulo invertido e controle PID.

Environment for simulation of physical systems applied to inverted pendulum on a cart and PID control problem.

RESUMO

Dentro do universo de sistemas para controle de processos, métodos são frequentemente atualizados. Com a necessidade crescente de otimização e de adaptação dos controladores nos encontramos constantemente em busca de novas abordagens e tecnologias. No intuito de apresentar uma abordagem inovadora do controlador PID junto a uma plataforma até então ainda pouco explorada no mundo acadêmico, esse trabalho demonstra a simulação de um pêndulo invertido sobre um carro em um ambiente 2D utilizando a linguagem Python, criado a partir de um framework de simulação de partículas, Pymunk. Para avaliação do desempenho do sistema, são apresentados quatro critérios de erro: Integral do erro quadrático (ISE) de $4,14e-5$, Integral do erro absoluto (IAE) com 0,0776, Integral do erro quadrático multiplicado pelo tempo (ITSE) com $6,93e-4$ e por fim a Integral do erro absoluto multiplicado pelo tempo (ITAE) com 1,2915.

PALAVRAS-CHAVE: Pêndulo invertido. Controle PID. Simulação de sistemas dinâmicos.

ABSTRACT

Within the universe of systems for process control, methods are frequently updated. With the growing need for optimization and adaptation of controllers we are constantly looking for new approaches and Technologies. In order to present a novel approach to PID controllers together with a platform hitherto little explored in the academic world, this work demonstrates the simulation of an inverted pendulum on a cart in a 2D environment using Python language, created from a framework for simulation of particles, Pymunk. For evaluation of system performance, four criteria for errors are present: Integral Squared Error (ISE) with $4.16e-5$, Integral of Absolute Error (IAE) with 0.0776, Integral of Time Weighted Squared Error (ITSE) with $6.93e-4$ and finally Integral of Time Weighted Absolute Error (ITAE) with 1.29.

KEYWORDS: Inverted pendulum. PID Control. Dynamic systems simulation.

Lucas de Freitas
lucasfreitas.2017@alunos.utfpr.edu.br
Universidade Tecnológica Federal do Paraná, Toledo, Paraná, Brasil

Daniel Cavalcanti Jeronymo
danielc@utfpr.edu.br
Universidade Tecnológica Federal do Paraná, Toledo, Paraná, Brasil

Recebido: 19 ago. 2020.
Aprovado: 01 out. 2020.

Direito autorial: Este trabalho está licenciado sob os termos da Licença Creative Commons-Atribuição 4.0 Internacional.



INTRODUÇÃO

Em meio a inúmeras formas de sintonizar o controlador PID, contabilizadas em 1731 métodos de sintonia por O'Dwyer (2009), algumas mais novas outras nem tanto, acabamos por perceber que a maior parte do tempo gasto na implementação do controlador, está na sua sintonização. Não é de se estranhar, pois é a qualidade da sintonia que nomeará o controlador como eficiente ou como um gasto não tão bem aproveitado. Metodologias de testes de sintonias passam pelas fases analíticas, onde a sintonia é fundamentada matematicamente para então seguir aos testes através de simulação e aplicações *hardware-in-the-loop*, para finalmente então atingir aplicação real (COELHO, JERONYMO, ARAÚJO, 2019).

Todorov, Erez e Tassa (2012) e Erez, Tassa, Todorov (2015) descrevem a necessidade acadêmica e industrial por softwares que possam simular com precisão as dinâmicas complexas e não-lineares de sistemas robóticos e o produto resultante em busca de uma solução, MuJoCo. Outros softwares desta categoria de simulação são Webots, Simpark, RoboDK e Gazebo. Entretanto, estas soluções sofrem ao encontrar aplicação acadêmica ou pelo custo excessivo ou pela complexidade dificultando o acesso.

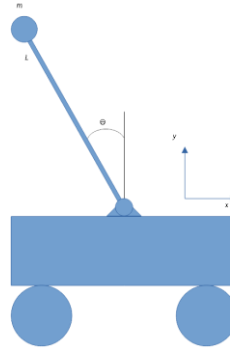
Neste trabalho é apresentado um ambiente para simulação de sistemas físicos em uma perspectiva em duas dimensões. O ambiente foi desenvolvido para uso didático em aulas, pesquisas e TCCs. Para demonstrar a efetividade do ambiente foi montado o modelo de um carro com um pêndulo invertido. A estrutura se baseia em uma plataforma móvel com rodas e uma haste que com a força da gravidade aplicada na simulação, tende a sair do seu ponto de equilíbrio realizando um movimento angular no plano. Logo, o objetivo se torna através do controle das rodas, conseqüentemente da velocidade da plataforma, manter o pêndulo em seu ponto de equilíbrio. Para isso foi elaborado um controlador PID objetivando aplicar um deslocamento na mesma direção do movimento do pêndulo para rebater o momento agindo sobre a haste. A fim de julgar a eficácia da sintonia do controlador PID foi realizada a análise dos critérios de erro ISE, IAE, ITSE e ITAE.

Este artigo está organizado da seguinte forma: é apresentada a descrição do framework utilizado na construção do ambiente e o problema do pêndulo invertido, assim como sua representação matemática e uma exposição do controlador PID, sua aplicação neste trabalho e a forma com que foi discretizado, por fim iremos exibir os resultados obtidos através dos quatro índices ISE, IAE, ITSE e ITAE.

AMBIENTE DE SIMULAÇÃO E PÊNDULO INVERTIDO

O pêndulo invertido, Figura 1, é um sistema clássico de estudo na Teoria de Controle, é um sistema com dois pontos de equilíbrio sendo um deles, o superior, instável, isto é, a partir do momento que é solto tende a cair a menos que sua instabilidade seja compensada por um controlador.

Figura 1 – Modelo matemático do pêndulo invertido



Fonte: Autoria própria (2020).

Para que o pêndulo se mantenha estável, devem ser equilibradas as somatórias de força sobre a haste que possui peso m e comprimento l , entre elas estão a somatória de forças nos eixos X e Y assim como o momento que atua na base da haste. Neste trabalho tomamos por objeto de controle a diferença do ângulo que existe entre a haste e a referência de estabilidade, 90 graus em relação a plataforma, θ .

Para a implementação do ambiente foi utilizada a linguagem Python com a biblioteca de física Pymunk, por sua vez construída em cima da biblioteca Chipmunk, baseada em C, com o objetivo de simular partículas físicas em um ambiente 2D. O modelo da plataforma, exibido na Figura 2, assume elos rígidos com as rodas e juntas do tipo pino.

Figura 2 – Ambiente de simulação exibindo o modelo do carro com pêndulo invertido



Fonte: Autoria própria (2020).

A simulação com o Pymunk **trabalha o conceito** de corpos rígidos e juntas de diversos tipos para servir como conexão entre os diferentes corpos do modelo. **O projeto de controlador objetiva fornecer torque às rodas da plataforma movimentando-a para** compensar a instabilidade do pêndulo.

CONTROLADOR PID

O controlador PID, apresentado completo pela primeira vez em 1911 pelo cientista inventor nova-iorquino Elmer Sperry (Sperry, 1922) com o objetivo de automatizar a direção dos navios da marinha americana, é composto por três ganhos. Primeiramente o ganho K_p (Proporcional), tem como objetivo reduzir o erro estacionário e o tempo de resposta do sistema, K_i (Integrativo) sendo responsável pela retirada desse erro estacionário remanescente tendo como custo retardar a resposta do sistema e aumentar o *overshoot* em uma resposta transiente

e por fim o kd (Derivativo) que de modo geral vem para otimizar o sistema aumentando a agilidade e estabilidade, porém introduz sensibilidade a ruídos.

Para realizar a implementação digital o controlador foi discretizado utilizando o método de diferenças finitas *backward*. Considera-se a equação do controlador no domínio contínuo:

$$u(t) = Kp \cdot e(t) + Ki \cdot \int_0^t e(t) \cdot dt + Kd \cdot \frac{de(t)}{dt} \quad (1)$$

Assume-se que a variável t é quantizada em períodos de tempo T_s , também chamado de tempo de amostragem, pela troca de variável $t=Kt_s$, onde $K \in \mathbb{Z}^-$. Para representar o sinal no período discreto, é estabelecida a seguinte notação para representar e diferenciar as variáveis discretas das contínuas, tal que f representa um sinal no tempo.

$$f[k] = f(KT_s) \quad (2)$$

A transposição da integral para o domínio discreto é a somatória, sendo assim:

$$\sum_{i=0}^k e[i] \cdot T_s \quad (3)$$

A derivada no espaço discreto segue a própria definição de derivada

$$\frac{de(t)}{dt} = \lim_{h \rightarrow 0} \inf \left(\frac{e(t+h) - e(t)}{h} \right) = \lim_{h \rightarrow 0} \inf \frac{e(h) - e(t-h)}{h} \quad (4)$$

No tempo discreto tem-se a derivada como:

$$\frac{e[k] - e[k-1]}{T_s} \quad (5)$$

Combinando as definições (4), (5) e (6) na equação do controlador PID (1) atinge-se a forma posicional de controle:

$$u[k] = Kp \cdot e[k] + Ki \cdot \sum_{i=0}^k e[i] \cdot T_s + Kd \cdot \frac{e[k] - e[k-1]}{T_s} \quad (6)$$

Considerando a diferença entre os sinais $u[k] - u[k-1]$, chega-se na forma de velocidade do controlador, onde o termo integral é englobado pela diferença:

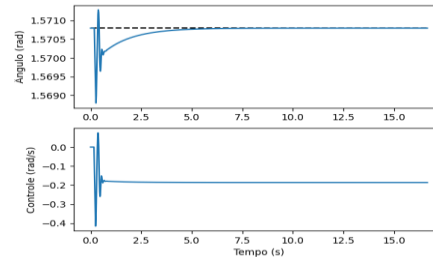
$$\Delta u[k] = \left(Kp + Ki \cdot T_s + \frac{Kd}{T_s} \right) \cdot e[k] - \left(Kp + 2 \cdot \frac{Kd}{T_s} \right) \cdot e[k-1] + \frac{Kd}{T_s} \cdot e[k-2] \quad (7)$$

O termo de erro consiste na diferença entre uma referência a ser mantida, o ângulo de 90 graus, e a saída do processo, consistindo no ângulo do pêndulo. A saída do controlador atua como sinal exercendo torque sobre os motores criados para controlar as rodas da plataforma

RESULTADOS

O ambiente foi simulado por quinze segundos com a referência de 90 graus para o controlador buscar manter o pêndulo na posição de equilíbrio. Durante a simulação é aplicada uma perturbação ao sistema no instante $T=0.16s$ e conforme exposto na Figura 3, o controlador reage para corrigir a oscilação e retornar o pêndulo para o valor referencial 90 graus ou $1.5708(\pi/2)$ radianos.

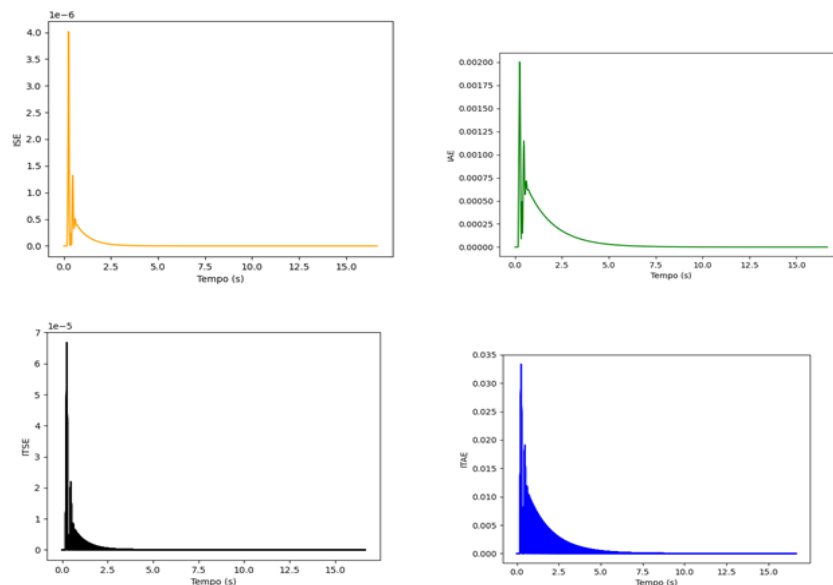
Figura 3 – Ângulo do pêndulo e ação de controle.



Fonte: Autoria própria (2020).

A eficácia do controlador é observada graficamente na Figura 4, apresentando os quatro critérios: ISE, IAE, ITSE e ITAE respectivamente, projetados para instante de tempo.

Figura 4 – Gráficos dos critérios de erro.



Fonte: Autoria própria (2020).

O critério ISE (Integral do erro quadrático) exibe uma penalidade maior no início do tempo pois essa é uma característica do critério, quantificar em maior escala erros maiores em relação aos erros menores. Caso este critério fosse escolhido para ser minimizado, os erros grandes do início seriam reduzidos rapidamente e os erros menores durariam por maior tempo, podendo levar a respostas rápidas com oscilações de baixa amplitude.

O critério IAE (Integral do erro absoluto), apresenta resposta mais suavizada quando comparado ao ISE por não penalizar o sinal do erro, a priorização deste critério levaria a uma resposta mais lenta e com menores oscilações.

Os critérios ITSE (Integral do erro quadrático multiplicado pelo tempo) e ITAE (Integral do erro absoluto multiplicado pelo erro), análogos ao ISE e IAE, realizam adicionalmente um compromisso entre a magnitude de erro e o tempo de acomodação, ao penalizar mais fortemente erros propagados no tempo.

Na Tabela 1 são apresentados os valores de cada critério ao final da simulação.

Tabela 1 – Critérios de erro ao final da simulação

Critérios	Valores
IAE	0,0776
ITAE	1,2915
ISE	4,16e-5
ITSE	6,93e-4

Fonte: A autoria própria (2020).

CONCLUSÃO

Neste trabalho foi apresentado um ambiente desenvolvido para aplicações didáticas e de pesquisa para problemas que possam ser expressos e estudados em duas dimensões. Como demonstração de uso do ambiente foi criado o modelo de um carro com pêndulo invertido, um sistema não linear, com aplicação de um controlador PID com o objetivo de manter o pêndulo em sua posição instável. Os resultados apresentados apontam a aplicabilidade do sistema na simulação e estudo de problemas bidimensionais.

Trabalhos futuros irão focar no aprimoramento da construção do modelo. O modelo físico nesta versão do ambiente é construído manualmente através de código. Almeja-se a construção de modelos graficamente, de tal maneira que o usuário tenha opções de arrastar e desenhar formas, motores, juntas e sensores para compor a estrutura final e armazená-la em arquivos XML ou JSON.

REFERÊNCIAS

- COELHO, A. A. R. ; JERONYMO, D. C. ; ARAUJO, R. B. . Sistemas dinâmicos: Controle clássico e preditivo discreto. 1. Ed. Florianópolis: Editora UFSC, 2019. V. 1.374p.
- EREZ, T., TASSA, Y. and TODOROV, E., "Simulation tools for model-based robotics: Comparison of Bullet, Havok, MuJoCo, ODE and PhysX," 2015 IEEE International Conference on Robotic and Automation (ICRA), Seattle, WA, 2015, pp. 4397-4404, doi: 10.1109/ICRA.2015.7139807
- O'DWYER, A. Handbook of PI and PID Controller Tuning Rules. Imperial College Press, London, UK, 2009.
- SPERRY, E. Automatic steering. Transactions of the Society of Naval Architects and Marine Engineers, v. 30, p. 53-57, 1922.
- TODOROV, E., EREZ, T. and TASSA, Y., "MuJoCo: A Physics engine for model-based control," 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, Vilamoura, 2012, pp. 5026-5033, DOI: 10.1109/IROS.2012.6386109.