Revista Brasileira de Computação Aplicada, Novembro, 2024

DOI: 10.5335/rbca.v16i3.15852
Vol. 16, Nº 3, pp. 74–88
Homepage: seer.upf.br/index.php/rbca/index

ARTIGO ORIGINAL

Análise comparativa do controle PID e MPC nos ambientes CartPole e LunarLander

Comparative analysis of PID control and MPC in CartPole and LunarLander environments

Gabriel Bueno Leandro [1], and Samir Angelo Milani Martins [1,2]

[1]Grupo de Controle e Modelagem - GCoM - Universidade Federal de São João del-Rei, [2]Departamento de Engenharia Elétrica - Universidade Federal de São João del-Rei

*gabrielbuenoleandro00@aluno.ufsj.edu.br
*martins@ufsj.edu.br

Recebido: 06/05/2024. **Revisado:** 12/11/2024. **Aceito:** 30/11/2024.

Resumo

Com o avanço tecnológico, sistemas cada vez mais complexos surgem, impulsionando a busca incessante por métodos de controle mais eficazes. Neste estudo, será explorado o controle de dois ambientes do Gymnasium: o LunarLander e o CartPole. O LunarLander representa uma simplificação da autonomia de veículos, enquanto o CartPole está intimamente ligado à robótica. O objetivo é aplicar o controle Preditivo Baseado em Modelo (MPC) e o controle Proporcional-Integral-Derivativo (PID) a esses ambientes, visando alcançar seus objetivos específicos. As técnicas empregadas demonstraram sucesso no controle dos ambientes, destacando a eficácia desses métodos em desafios complexos. Além disso, este estudo aborda diversos conceitos de engenharia, desde fundamentos físicos até identificação de sistemas, aproveitando-se de pacotes de Python para oferecer uma abordagem acessível tanto à academia quanto à indústria. Essa análise multidisciplinar promete contribuir significativamente para o avanço do conhecimento e aplicação prática no campo do controle de sistemas complexos.

Palavras-Chave: CartPole; Controle Preditivo Baseado em Modelo; Controle Proporcional-Integral-Derivativo; LunarLander.

Abstract

With technological advancement, increasingly complex systems emerge, driving the relentless pursuit of more effective control methods. In this study, the control of two Gymnasium environments will be explored: LunarLander and CartPole. LunarLander represents a simplification of vehicle autonomy, while CartPole is closely related to robotics. The goal is to apply Model Predictive Control (MPC) and Proportional-Integral-Derivative (PID) control to these environments, aiming to achieve their specific objectives. The techniques employed have shown success in controlling the environments, highlighting the effectiveness of these methods in complex challenges. Additionally, this study addresses various engineering concepts, from physical fundamentals to system identification, leveraging Python packages to offer an accessible approach for both academia and industry. This multidisciplinary analysis promises to significantly contribute to the advancement of knowledge and practical application in the field of complex system control.

Keywords: CartPole; Model Predictive Control; Proportional-Integral-Derivative Control; LunarLander.

1 Introdução

Na atualidade tecnológica, a pesquisa em controle de sistemas dinâmicos ganha proeminência devido à busca constante por aprimoramentos no desempenho de sistemas complexos. Este estudo investigou duas técnicas de controle amplamente reconhecidas, o Controlador-Proporcional-Integral-Derivativo (PID) e o Controle Preditivo Baseado em Modelo (MPC). Ao final da análise, são realizadas ponderações comparativas entre essas abordagens. Para contextualizar e aplicar essas técnicas, foram utilizadas as interfaces de programação de aplicativos (APIs) CartPole (pêndulo invertido) e LunarLander (Sonda Lunar). Esses ambientes ofereceram cenários práticos para avaliar e comparar as técnicas de controle PID e MPC.

O engenheiro russo Nikolai Fyodorovich Minorsky (1885-1970) é reconhecido como o pioneiro do controlador de três termos, que se originou do projeto de direção automática de navios para a marinha norte-americana. Esses três termos correspondem às componentes proporcional, integral e derivativa, formando o que é conhecido como controlador PID (Minorsky, 1922). Apesar de ter mais de um século de existência, o controlador PID é atualmente o mais utilizado nas malhas fechadas industriais devido à sua robustez e facilidade de implementação (Deulkar and Hanwate, 2020). Por esse motivo, continua sendo amplamente explorado no meio acadêmico.

O MPC é uma estratégia que se baseia na previsão do comportamento futuro por meio de um modelo do sistema. Neste contexto, foi adotada a abordagem do Controle Preditivo Generalizado (GPC), que é atualmente a técnica mais empregada no âmbito do MPC. Originando-se do desenvolvimento do controle de mínima variância (Camacho and Bordons, 2004), o GPC é uma técnica avançada que emprega modelos paramétricos sendo projetado a partir de modelos auto-regressivos, média móvel e com sinal exógeno do sistema (ARMAX, do inglês *Auto-Regressive Moving Average with Exogenous Inputs*), permitindo antecipar o comportamento futuro e otimizar as ações de controle. No presente trabalho, foram aplicadas técnicas de identificação de sistemas utilizando o pacote SysIdentPy (Lacerda et al., 2020) para estimar o modelo paramétrico. A ISA (*International Society of Automation*, em inglês) considera que o controle preditivo é uma ferramenta importante capaz de diferenciar entre um bom e um excelente Engenheiro de Controle.

O desafio apresentado pelo pêndulo invertido é um clássico na engenharia de controle, estando presente em alguns livros nessa área, dentre eles, Ogata and Katsuhiko (2010) e Castrucci et al. (2011). Este sistema é frequentemente utilizado como um exemplo didático devido à sua natureza instável e não linear, que pode ser linearizada por meio de aproximações. O pêndulo invertido encontra aplicação prática em diversas áreas, como por exemplo, na robótica (Boubaker, 2012). Para explorar e simular o comportamento do sistema de pêndulo invertido, foi utilizado o ambiente CartPole da Gymnasium.

O desafio do pouso lunar foi abordado por meio da simulação no ambiente LunarLander, onde uma sonda deve realizar um pouso preciso. Esse problema é notável por sua complexidade, uma vez que envolve um sistema MIMO (Múltiplas Entradas e Múltiplas Saídas).

2 Embasamento teórico

2.1 Gymnasium

O estudo utiliza os ambientes de simulação CartPole e LunarLander, fornecidos pelo Gymnasium, uma bifurcação da biblioteca Gym da OpenAI, voltada para aprendizado por reforço (RL). O Gymnasium facilita a interação entre algoritmos e ambientes por meio de uma API padronizada, promovendo o treinamento de agentes para maximizar recompensas acumuladas em ambientes dinâmicos.

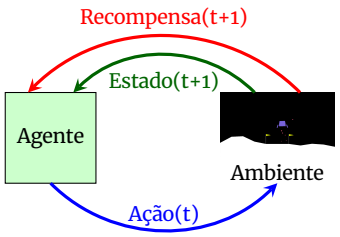


Figura 1: Gymnasium.

Conforme a Fig. 1, a interação entre a API do Gymnasium e o agente, seja este um controlador PID/MPC ou um algoritmo de aprendizado por reforço, engloba a observação do estado atual do ambiente, a tomada de decisão de uma ação com base nessa observação e a execução dessa ação no ambiente. O processo continua em loop até que um critério de término seja alcançado.

2.2 CartPole

Neste cenário, o pêndulo é posicionado verticalmente sobre o carro, tendo como meta equilibrar o pêndulo ao aplicar forças nas direções esquerda e direita no carro (Barto, 2023). O CartPole é um problema clássico que envolve um pêndulo invertido, como representado na Figura Fig. 2.

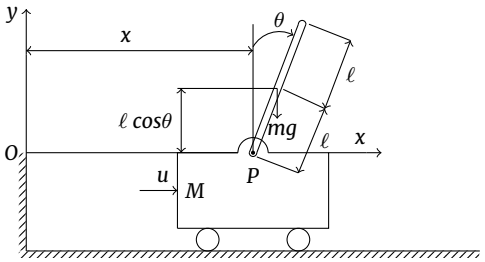


Figura 2: CartPole.

A interação do CartPole com algoritmo se dá por meio de sua entrada, conforme ações apresentadas na Tabela 1.

Tabela 1: Entrada do CartPole.

Número	Ação
0	Empurre o carrinho para à esquerda
1	Empurre o carrinho para à direita

O seu espaço de observação é mostrado na Tabela 2.

Tabela 2: Espaço de observação do *Cart Pole*.

Número	Observação	Mínimo	Máximo
0	Posição do carrinho	-4,8	4,8
1	Velocidade do carrinho	-Inf	Inf
2	Ângulo do polo	-24°	24°
3	Velocidade angular do polo	-Inf	Inf

O ambiente se encerra quando o CartPole atinge 500 etapas.

2.3 LunarLander

Neste ambiente, a tarefa consiste em realizar um pouso suave de uma sonda espacial na superfície lunar, lidando com desafios como a gravidade, obstáculos e utilizando informações cruciais, como a posição, ângulo e velocidade da sonda, conforme a Fig. 3.

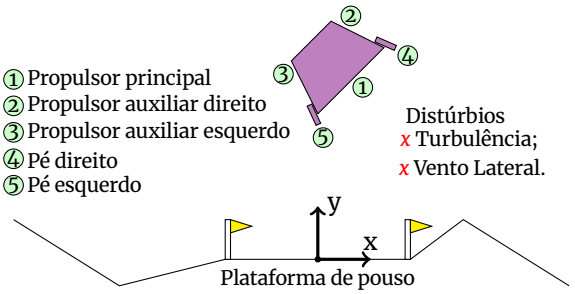


Figura 3: LunarLander.

A comunicação entre o LunaLander e o agente ocorre através de suas entradas, como detalhado na Tabela 3.

Tabela 3: Entradas do LunarLander.

Número	Ação
0	Não fazer nada
1	Motor de orientação à esquerda da sonda
2	Acionamento do motor principal
3	Motor de orientação à direita da sonda

O espaço de observação é mostrado na Tabela 4.

Tabela 4: Espaço de observação do LunarLander.

Número	Observação	Mínimo	Máximo
0	Coordenada x	-1,5	1,5
1	Coordenada y	-1,5	1,5
2	Velocidade linear em x	-5	5
3	Velocidade linear em y	-5	5
4	Ângulo	−360°	360°
5	Velocidade angular	-5	5
6	Contato pé esquerdo	0	1
7	Contato pé direito	0	1

Após cada passo no ambiente LunarLander, uma recompensa é concedida ao controlador, e a recompensa total de um episódio é a soma dessas recompensas ao longo de todas as etapas do episódio. A recompensa por etapa é determinada por quanto aproxima da plataforma de pouso, pela velocidade do módulo de pouso, pelo ângulo de inclinação do módulo e pelo número de pernas em contato com o solo. Além disso, há penalidades para o uso dos motores laterais e do motor principal. O episódio também recebe uma recompensa adicional de -100 ou +100 pontos por bater ou pousar com segurança, respectivamente. Um episódio é considerado uma solução se marcar pelo menos 200 pontos (Klimov, 2023).

2.4 Controle PID

Um controlador PID é um sistema de controle composto por três componentes ajustáveis: proporcional, integral e derivativo (Minorsky, 1922), conforme a Fig. 4.

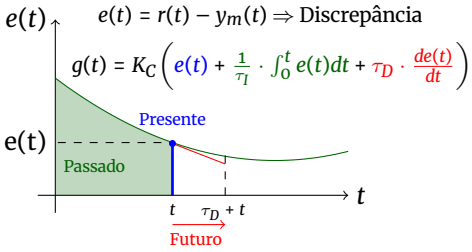


Figura 4: Componentes PID, uma ilustração.

Estes elementos são adaptados com base na discrepância entre a referência ($r(t)$) e uma variável de processo medida em malha fechada ($y_m(t)$), conforme a Fig. 5.

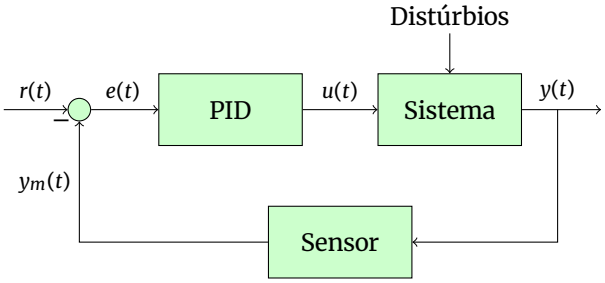


Figura 5: PID aplicado a uma malha de controle.

2.5 Controle Preditivo Baseado em Modelo

O MPC busca, por meio de um modelo, estimar a saída futura de um sistema. Essa saída pode ser fragmentada em dois termos, sendo $\hat{y} = f_p(u_{\text{Passado}}) + f_t(\Delta u_{\text{Futuro}}) = y_{\text{Liv}} + y_{\text{For}}$, onde y_{Liv} depende das condições iniciais no instante k. Logo, essa parcela não pode ser alterada. O que pode ser feito é alterar y_{For} por meio da entrada Δu_{Futuro}, de modo que $\hat{y}$ se aproxime da trajetória de referência, conforme mostrado na Fig. 6.

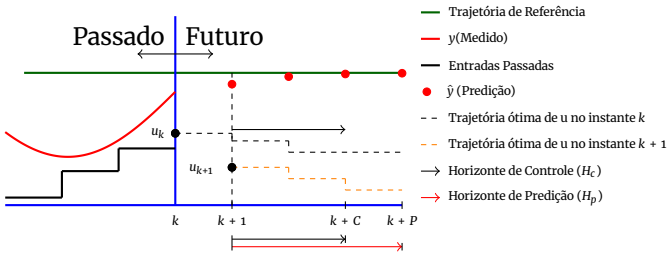


Figura 6: Controle Baseado em Modelo.

O MPC é uma estratégia de controle com as seguintes características:

- Utiliza um modelo explícito do processo para predizer sua saída em um horizonte finito;
- Calcula as ações de controle futuras minimizando uma função objetivo específica;
- Apresenta um horizonte de previsão deslizante, ou seja, a cada período de amostragem, o horizonte é deslocado um passo à frente. O sinal de controle no instante atual é aplicado ao processo, desconsiderando o restante do horizonte de controle.

O Controle Preditivo Generalizado (GPC) foi proposto inicialmente em 1987 (Clarke et al., 1987a,b) e tornou-se uma das estratégias mais amplamente adotadas tanto no meio acadêmico quanto na indústria no âmbito do MPC (Camacho and Bordons, 2004). O GPC utiliza modelos paramétricos, conforme o diagrama de blocos mostrado na Fig. 7.

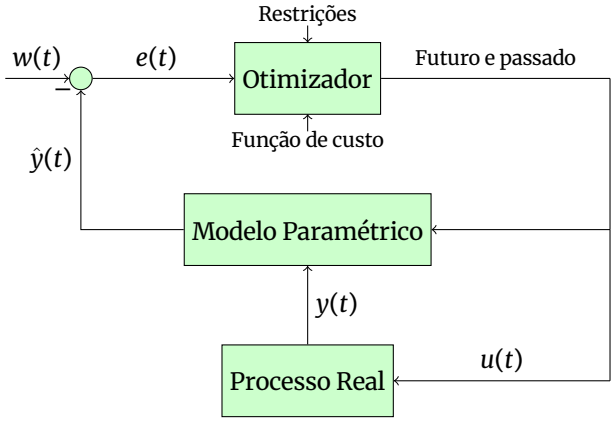


Figura 7: Diagrama de Blocos do GPC.

2.6 Método de Pesquisa em Grade

Como discutido, os ambientes têm entradas inteiras, o que sugere a minimização de uma função custo por meio de otimização inteira. Dada a quantidade limitada de entradas, o Método de Pesquisa em Grade é viável. Apesar de uma complexidade $O(n!)$, indicando crescimento fatorial do tempo de execução devido a n, sua aplicação é factível para horizontes de controle mais curtos. Este método examina todas as combinações possíveis de variáveis, evitando convergir erroneamente para mínimos/máximos locais ao buscar o valor mínimo/máximo global.

2.7 Algoritmo Subida de Encosta

O Método de Subida de Encosta é um algoritmo clássico para otimização (Carneiro, 2020). Esse algoritmo, inicia-se a partir de um ponto aleatório X e realiza-se sua avaliação. Em seguida, ocorre o deslocamento do ponto original X para um novo ponto vizinho, designado como X'. Se o novo ponto X' representar uma solução superior à do ponto anterior, permanece-se nele e o processo é repetido. Caso não encontre tal aprimoramento, a execução se encerra.

3 Metodologia

3.1 CartPole - PID

3.1.1 *Função de Transferência*

O CartPole é um pêndulo invertido montado sobre um carro motorizado, como mostrado na Fig. 2.

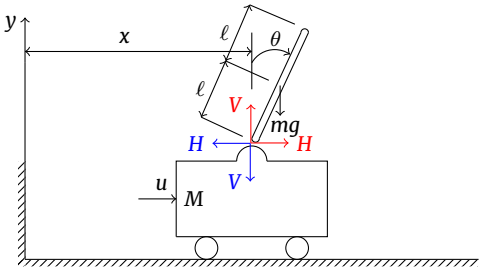


Figura 8: Diagrama de corpo livre do pêndulo invertido.

A Eq. (1) descreve o deslocamento horizontal do carro:

$$M\frac{d^2x}{dt} = u - H. \tag{1}$$

A Fig. 9 ilustra deslocamento do centro de massa (x_c).

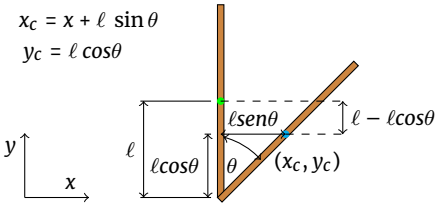


Figura 9: Deslocamento do centro de massa do pêndulo.

As Eq. (2) e Eq. (3) descrevem o equilíbrio das forças horizontais e verticais no centro de massa, respectivamente.

$$m\frac{d^2}{dt}(x + \ell\, sen\theta) = H. \tag{2}$$

$$m\frac{d^2}{dt}(\ell - \ell\, cos\theta) = mg - V. \tag{3}$$

Ao considerar que o CartPole encerra o episódio quando o ângulo excede $\pm 12°$, é razoável afirmar que, dentro desse intervalo, $sen\theta \approx \theta$ e $cos\theta \approx 1$, logo:

$$m\frac{d^2}{dt}(x + \ell\theta) = H. \tag{4}$$

$$\begin{aligned} m\frac{d^2}{dt}\overbrace{(\ell - \ell)}^{0} &= mg - V, \\ mg &= V. \end{aligned} \tag{5}$$

O próximo passo consiste em determinar o deslocamento rotacional, conforme mostrado na Fig. 10.

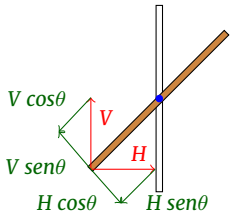


Figura 10: Deslocamento rotacional do pêndulo.

O movimento rotacional da haste do pêndulo ao redor de seu centro de gravidade é descrito pela Eq. (6):

$$\begin{aligned} I\frac{d^2\theta}{dt} &= V\ell\overbrace{sen\theta}^{\theta} - H\ell\overbrace{cos\theta}^{1}, \\ I\frac{d^2\theta}{dt} &= V\ell\theta - H\ell. \end{aligned} \tag{6}$$

Note que H e V são desconhecidos, mas a *priori* não se trata de um problema, pois pela Eq. (4) é possível expressar H. Pela Eq. (5), tem-se que $V = mg$, levando ao próximo passo, substituir esses valores na Eq. (6), obtendo:

$$\begin{aligned} I\frac{d^2\theta}{dt} &= mg\theta\ell - m\frac{d^2}{dt}(x + \ell\theta)\ell, \\ (I + m\ell^2)\frac{d^2\theta}{dt^2} &+ m\ell\frac{d^2x}{dt^2} - mg\ell\theta = 0. \end{aligned} \tag{7}$$

Substituindo a Eq. (4) na Eq. (1), obtém-se:

$$\begin{aligned} M\frac{d^2x}{dt^2} &= u - m\frac{d^2}{dt^2}(x + \ell\theta). \\ M\frac{d^2x}{dt^2} &+ m\frac{d^2}{dt^2}(x + \ell\theta) = u. \end{aligned} \tag{8}$$

A fim de derivar a função de transferência do sistema, é necessário converter a Eq. (7) e Eq. (8) para o domínio de Laplace, considerando condições iniciais nulas:

$$\begin{aligned} (I + m\ell^2)s^2\Theta(s) + m\ell s^2X(s) - mg\ell\Theta(s) = 0, \\ ((I + m\ell^2)s^2 - mg\ell)\Theta(s) + m\ell s^2X(s) = 0. \end{aligned} \tag{9}$$

$$(M + m)s^2X(s) + m\ell s^2\Theta(s) = U(s). \tag{10}$$

Isolando $X(s)$, a Eq. (9) é reescrita como:

$$X(s) = -\frac{\{(I + m\ell^2)s^2 - mg\ell\}}{m\ell s^2}\Theta(s), \tag{11}$$

substituindo a Eq. (11) na Eq. (10), encontra-se:

$$\begin{aligned} -(M + m)\cancel{s^2}\frac{\{(I + m\ell^2)s^2 - mg\ell\}}{m\ell\cancel{s^2}}\Theta(s) + m\ell s^2\Theta(s) = U(s), \\ \frac{[m^2\ell^2s^2 - (M + m)\{(I + m\ell^2)s^2 - mg\ell\}]}{m\ell}\Theta(s) = U(s). \end{aligned} \tag{12}$$

Por fim, após modelar o CartPole (Ogata and Katsuhiko, 2010), obtém-se a FT:

$$\frac{\Theta(s)}{U(s)} = \frac{m\ell}{(m^2\ell^2 - (M + m)(I + m\ell^2))s^2 + (M + m)mg\ell}. \tag{13}$$

Ainda é necessário saber quais são os valores de massa, comprimento e momento de inércia. Na verdade, não é preciso conhecer o valor exato de cada uma dessas grandezas, pode-se aproximá-las com simulações e alguns conceitos fundamentais da física (Roberts, 2021). Portanto, será aplicada uma força à direita (aplicar 1 a entrada, conforme a Tabela 1) para observar como o sistema se comporta. Como o ambiente fornece os valores de velocidade e velocidade angular, é possível plotá-los para iniciar uma investigação, como mostrado na Fig. 11.

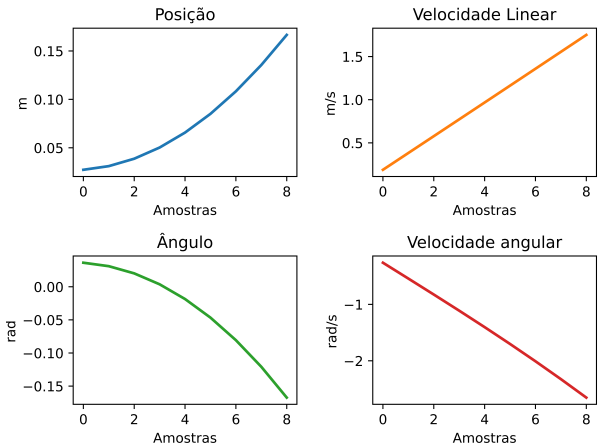


Figura 11: Força constante à direita.

As acelerações linear e angular é por definição a in-

clinação da velocidade. Para calcular essas inclinações, recorreu-se à função linregress do pacote Scipy. Os resultados obtidos foram:

$$\ddot{x} = 0,19524 \frac{m}{s^2} \quad \text{e} \quad \ddot{\theta} = -0,29775 \frac{rad}{s^2}. \tag{14}$$

Os valores das acelerações permanecem constantes, independentemente do número de simulações, indicando uma base física por trás do CartPole. Considere a Fig. 12.

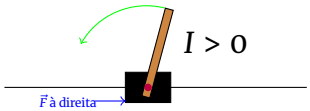


Figura 12: CartPole.

O momento de inércia deve ser sempre positivo, conforme a Fig. 12. Para tal análise, foi utilizada a Eq. (15):

$$(I + m\ell^2)\frac{d^2\theta}{dt^2} + m\ell\frac{d^2x}{dt^2} - mg\ell\theta = 0. \tag{15}$$

Dado o objetivo de manter o sistema em equilíbrio, onde $\theta \approx 0$, surge a oportunidade de reformular a Eq. (15):

$$(I + m\ell^2)\frac{d^2\theta}{dt^2} = -m\ell\frac{d^2x}{dt^2}, \tag{16}$$

onde $\frac{d^2\theta}{dt^2} = -0,29775 \frac{rad}{s^2}$ e $\frac{d^2x}{dt^2} = 0,19524 \frac{m}{s^2}$, o valor adotado para m será $0,5kg$. Logo:

$$I = \frac{0,0976\ell - 0,1488\ell^2}{0,29775}. \tag{17}$$

A Equação Eq. (17) expressa I em função de ℓ. Ao variar ℓ de 0 a $1m$, obtém-se o Fig. 13.

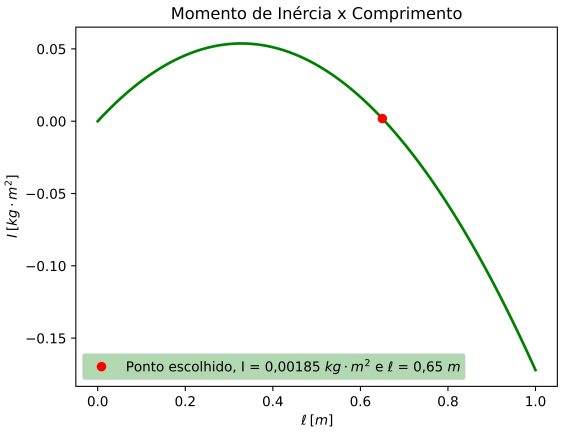


Figura 13: Relação entre I e o ℓ, para $m = 0,5kg$.

Note que o valor escolhido de ℓ apresenta I positivo.

O próximo passo envolve a estimativa da M (massa do carro), o isolando na Eq. (8):

$$M = \frac{u - m\ell\ddot{\theta} - m\ddot{x}}{\ddot{x}}. \tag{18}$$

Ao substituir os valores conhecidos, como $m = 0,5\,\text{kg}$, $u = 1\,\text{N}$ e $\ell = 0,65\,\text{m}$ na Eq. (18), foi encontrado $M = 5,11754\,\text{kg}$. Os resultados obtidos ou definidos estão apresentados na Tabela 5.

Tabela 5: Valores dos parâmetros.

$M[kg]$	$m[kg]$	$\ell[m]$	$I[kg \cdot m^2]$
5,11754	0,5	0,65	$1,8537 \times 10^{-3}$

Ao substituí-los na Função de transferência da Eq. (13), chega-se:

$$\frac{\Theta(s)}{U(s)} = \frac{0,325}{-1,0914s^2 + 17,9101}. \tag{19}$$

A fim de validar a função de transferência no ambiente CartPole do Gymnasium, foi realizado um experimento consistindo na aplicação da mesma entrada tanto no ambiente CartPole quanto na função de transferência, limitando-se a um único episódio (até que o sistema perda a estabilidade). Para adaptar a entrada, adotou-se a seguinte convenção: no ambiente CartPole, a entrada 1 indica movimento para a direita, enquanto 0 representa movimento para a esquerda. Por outro lado, na função de transferência, movimento para a direita também é representado por 1, mas para a esquerda, é representado por −1. Dessa forma, foi gerado o gráfico da Fig. 14.

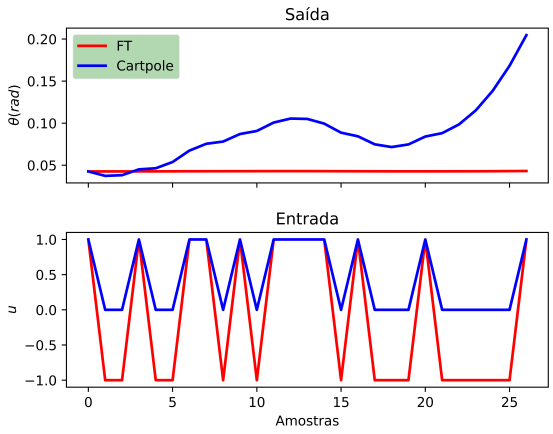


Figura 14: Entrada e saída.

A partir da Fig. 14, nota-se que a FT não conseguiu representar o sistema adequadamente. A FT responde de forma muito lenta aos estímulos de entrada. Portanto, é necessário ajustá-la para obter respostas mais abruptas.

Ao observar os passos anteriores, a aceleração considerada levou em conta 8 amostras (Ver Fig. 11), com um intervalo de 1*s* para um episódio, com o intuito de encontrar a inclinação. No entanto, devido à natureza computacional, o tempo é significativamente menor. Assim, as acelerações em linear e angular serão multiplicadas por um fator (k) até que a FT alcance um desempenho satisfatório. Dessa forma, será retornado a equação da inércia:

$$I = \frac{-m\ell \not{k}\ddot{x} - m\ell^2 \not{k}\ddot{\theta}}{\not{k}\ddot{\theta}}, \tag{20}$$

o momento de inércia independe do valor de k. Portanto, os valores de I e ℓ permanecem inalterados.

De acordo com a segunda Lei de Newton, à medida que a aceleração aumenta enquanto a força permanece constante, espera-se que a massa total ($M + m$) diminua. Em termos matemáticos, isso é evidenciado pela Eq. (21), onde m é mantido constante em $0,5kg$ (Massa do pêndulo), resultando em uma diminuição de M (Massa do carro), conforme descrito a seguir:

$$M = \frac{u - k\, m\ell\ddot{\theta} - k\, m\ddot{x}}{k\ddot{x}}. \tag{21}$$

Um aspecto crucial a ser considerado é que a massa do carro deve ser um valor positivo ($M > 0$). Portanto:

$$\begin{aligned} 0 &< \frac{u - k\, m\ell\ddot{\theta} - k\, m\ddot{x}}{k\ddot{x}}, \\ 0 &< 1 + k \cdot 5,5125 \times 10^{-4}. \end{aligned} \tag{22}$$

Ao resolver a Eq. (22), chega-se a $k < 1174,74$, logo, para determinar o valor de k, ele será variado de 1 a 1174, considerando 1000 etapas (amostras). Um conjunto de etapas compõe o episódio, sendo que cada episódio é concluído quando atinge 500 etapas ou quando o ângulo do pêndulo não está mais na faixa de $\pm 12^\circ$ ($\pm 0,2095rad$). Conforme demonstrado na Fig. 15.

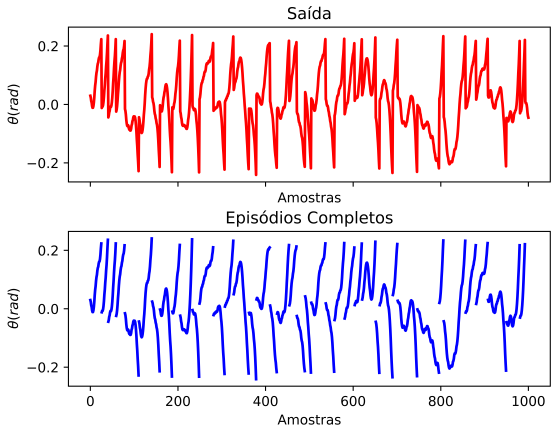


Figura 15: Episódios inteiros.

Ao total, foram realizados 47 episódios completos. Portanto, para cada valor de k, a simulação é repetida 47 vezes, e o somatório do Erro Quadrático Médio (*RMSE*)(Aguirre, 2015) é calculado. Em seguida, é determinada a média para obter o *RMSE* médio associado a k, da seguinte forma:

$$\overline{RMSE}_k = \frac{1}{47} \sum_{ep=1}^{47} \frac{\sqrt{\sum_{i=1}^{N} (y_{ep}(i) - \hat{y}_{ep}(i))^2}}{\sqrt{\sum_{i=1}^{N} (y_{ep}(i) - \bar{y}_{ep})^2}}, \tag{23}$$

onde $\hat{y}_{ep}(i)$ representa a simulação livre para cada episódio (FT) e y_k é o sinal medido para cada episódio, com a média ($\bar{y}_{ep}$) sendo calculada na janela de identificação. O resuldado da Eq. (23) será plotado na Fig. 16.

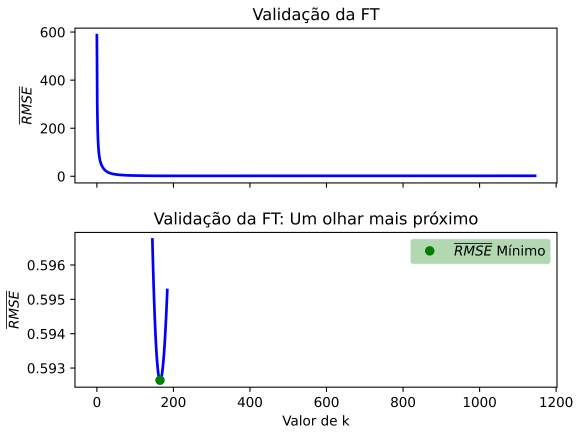


Figura 16: RMSE médio devido ao fator k.

Ao empregar o comando *min* do Python, obtém-se um valor de $\overline{RMSE}$ igual a 0,66 para $k = 165$. Ao tomar $k = 165$, obtém-se uma FT que responde melhor aos estímulos da entrada. De acordo com a Eq. (18), a massa do carro(M) é determinada como 0,02668 kg, resultando na FT.

$$\frac{\Theta(s)}{U(s)} = \frac{0,325}{-0,00661s^2 + 1,67919}. \tag{24}$$

Logo, pode-se representar o diagrama de blocos em malha aberta do sistema, conforme a Fig. 17.

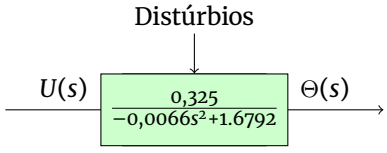


Figura 17: Diagrama do sistema em malha aberta.

Onde $U(s)$ representa a direção da força com módulo de $1N$ aplicada (com 1 indicando para a direita e -1 para a esquerda), enquanto Θ refere-se ao ângulo do pêndulo. Os

polos da FT são ±15, 9351, como mostrado na Fig. 18.

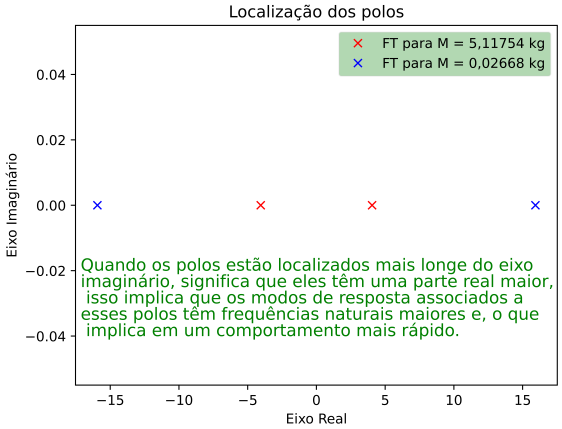


Figura 18: Polos.

É crucial notar que uma destas raízes está localizada no semiplano direito do eixo real, o que indica um sistema instável. Como exemplo do desempenho da FT, na Fig. 19, a mesma entrada será aplicada na FT e no CartPole.

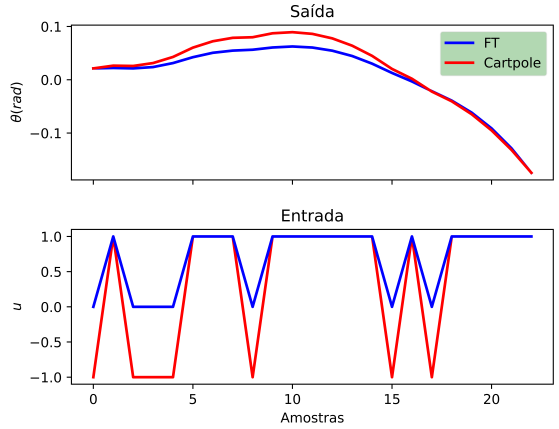


Figura 19: Entrada e saída - Validação da FT.

3.1.2 *Sintonia PID*

Para estabilizar o sistema, foi empregado um controlador PID como ilustrado na Fig. 20.

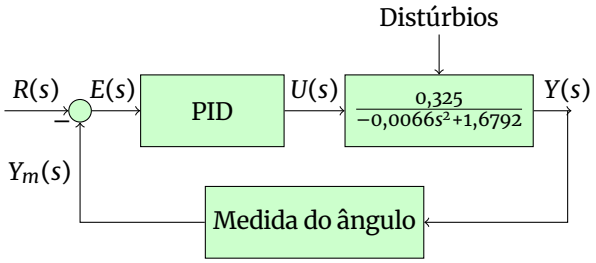


Figura 20: Malha de controle para o CartPole.

No âmbito temporal, a formulação da expressão para um controlador PID é delineada conforme a Eq. (25):

$$g(t) = K_C\left(e(t) + \frac{1}{\tau_I}\cdot\int_0^t e(t)dt + \tau_D\cdot\frac{de(t)}{dt}\right). \quad (25)$$

Para a determinar dos valores de K_P, K_I e K_D, foi utilizado o pacote GEKKO (Beal et al., 2018). Este pacote é uma ferramenta em Python dedicada ao aprendizado de máquina e otimização de inteiros mistos, bem como a equações algébricas diferenciais usada para resolver a Eq. (25). Nessa equação, os valores de τ_i e τ_d foram ajustados para 2 e 0,25, respectivamente, por meio de tentativa e erro. Assim, o pacote calcula o valor de K_C. A partir desse ponto, os parâmetros podem ser facilmente determinados.A referência consiste em um impulso, representando um distúrbio ideal:

$$e(t) = r(t) - y_m(t). \quad (26)$$

O GEKKO, ao buscar a minimização do erro, simplifica o processo, necessitando apenas da introdução da FT e da aplicação do impulso à entrada. Os valores estimados estão vinculados ao K_C determinado, o qual o GEKKO retorna o valor de 0, 5. Os valores dos ganhos proporcional, derivativo e integral do PID estão apresentados na Tabela 6.

Tabela 6: Parâmetros do PID.

Ganhos	Relação com K_C	Valor
Proporcional (K_P)	K_C	0,5
Integral (K_I)	$\frac{K_C}{\tau_I}$	0,25
Derivativo (K_D)	$K_C \cdot \tau_D$	0,125

Os valores dos parâmetros K_P, K_I e K_D foram incorporados à Eq. (25), resultando em:

$$g(t) = 0,5\cdot e(t) + 0,25\cdot\int_0^t e(t)dt + 0,125\cdot\frac{de(t)}{dt}. \quad (27)$$

Ainda é necessário determinar o valor do erro, sua integral e derivada no contexto do controle do CartPole. Dado que o *setpoint* $r(t)$ almejado é manter o pêndulo em pé, idealmente, esse *setpoint* é zero. Assim, o erro torna-se o próprio ângulo do CartPole, enquanto a derivada é a velocidade angular, ambas informações fornecidas pelo pacote Gymnasium.

Além disso, é crucial estimar a integral do erro. Fisicamente, a integral pode ser interpretada como a soma acumulativa do ângulo ao longo do tempo, já que o *setpoint* é zero. Este componente integral no controle é valioso para corrigir o erro acumulado ao longo do tempo, contribuindo para a estabilidade do sistema e a redução de desvios significativos. Matematicamente, a integral é dada por:

$$\int_0^t e(t)dt = \sum_{i=0}^{n} \theta_i, \tag{28}$$

aqui, θ_i denota o ângulo do pêndulo invertido, enquanto n representa o número atual da etapa.

Assim, ao levar em conta o ângulo, a velocidade angular e a integral do erro, é viável construir um conjunto abrangente de informações para a implementação do PID no ambiente do CartPole da seguinte forma:

$$u_n = 0,5 \cdot \theta_n + 0,25 \cdot \sum_{i=0}^{n} \theta_i + 0,125 \cdot \omega_n. \tag{29}$$

Antes de prosseguir com as tarefas, é fundamental realizar uma análise gráfica da saída do GEKKO na Fig. 21. Essa visualização desempenha um papel crucial como base para as etapas subsequentes deste trabalho.

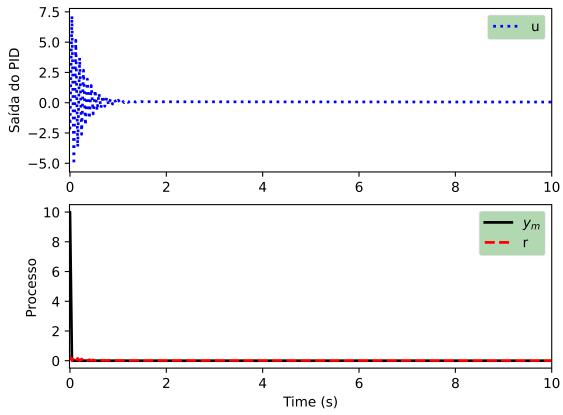


Figura 21: Resposta ao impulso para obtenção dos parâmetros do PID.

Pela Fig. 21, observe que a saída u assume valores contínuos, mas o ambiente CartPole utilizado é discreto (1 para força à direita e 0 para força à esquerda). Para resolver esse problema, foi implementada a seguinte estratégia: se $u > 0$, é aplicada uma força à direita; caso contrário, é aplicada uma força à esquerda.

3.2 GPC - CartPole

3.2.1 *Aplicação do GPC*

A trajetória de referência/*setpoint* (w) representa o comportamento do sinal desejado para a saída futura, sendo o primeiro passo para a aplicação do GPC. No caso do pêndulo invertido, ela será 0, pois a ideia é manter o pêndulo estável, o que implica em minimizar o θ, de maneira:

$$w = [0, 0, 0, \cdots, n_{etapas}], \tag{30}$$

onde n_{etapas} representa o número total de etapas.

O próximo passo consiste em estabelecer uma função de custo, considerando as recompensas do ambiente CartPole. Nesse contexto, uma recompensa é atribuída por cada passo dado, abrangendo inclusive a etapa de encerramento, uma vez que o objetivo é manter o poste ereto pelo maior tempo possível. O limite estabelecido para as recompensas é de 500 (Barto, 2023). Logo, o esforço de controle não será penalizado, apenas o erro em relação à predição da saída e à referência:

$$J(k) = \sqrt{(\sum_{j=d}^{h_p} [\hat{y}(j+k|k) - w(j+k)])^2}. \tag{31}$$

A função de custo utilizada enfatiza a capacidade do parâmetro θ de mudar de sinal, possibilitando sua proximidade ao limiar de 0 radianos.

No entanto, ainda é necessário identificar um modelo capaz de prever a saída ($\hat{y}$) em função de Δu, seguindo a metodologia do GPC. Nesse contexto, a saída é representada por 1 para movimento à direita e -1 para movimento à esquerda, a fim de manter consistência com a codificação utilizada na FT. Isso resulta em Δu assumindo três valores inteiros, conforme a Tabela 7.

Tabela 7: Δ_u possíveis.

u_{k-1}	u_k	Δu
1	-1	-2
1	1	0
-1	-1	0
-1	1	2

A definição do problema de otimização para o ambiente CartPole foi formulada de acordo com a Eq. (32).

$$\begin{aligned} \textbf{Minimizar} \quad & J(k) = \sqrt{(\textstyle\sum_{j=d}^{h_p} [\hat{y}(j+k|k) - w(j+k)])^2}. \\ \text{sujeito a:} \quad & \Delta u = \{-2, 0, 2\} \end{aligned} \tag{32}$$

Para minimizar essa função custo, emprega-se o Método de Pesquisa em Grade.

3.2.2 *Estimação do Modelo ARX*

Nesta seção há o processo de identificação do modelo ARX do sistema, utilizando um sinal PRBS (do inglês, *Pseudo-Random Binary Sequence*) para excitação. O PRBS (Aguirre, 2015) assume apenas dois valores possíveis, $+V$ e $-V$.

O menor intervalo no qual o nível do sinal é mantido é denominado T_b. Seu período pode ser determinado por $T = NT_b$, sendo N um número ímpar, de modo que $T_b = T_s$. Um resultado heurístico para a escolha de T_b é: $\frac{\tau_{\min}}{10} \leq T_b \leq \frac{\tau_{\min}}{3}$, onde $\tau_{\min}$ representa a menor constante de tempo de interesse, sugerida por Nelles (2001), que, para sistemas lineares, T_b seja escolhido próximo ao valor do intervalo de amostragem. Para sistemas não lineares, por outro lado, recomenda-se que T_b seja aproximadamente igual a τ_{max},

onde τ_{max} é a constante de tempo do sistema.

Ao empregar um modelo ARX, considera-se o tempo de amostragem T_b. Assim, obtém-se a saída devido ao PRBS conforme mostrado na Fig. 22.

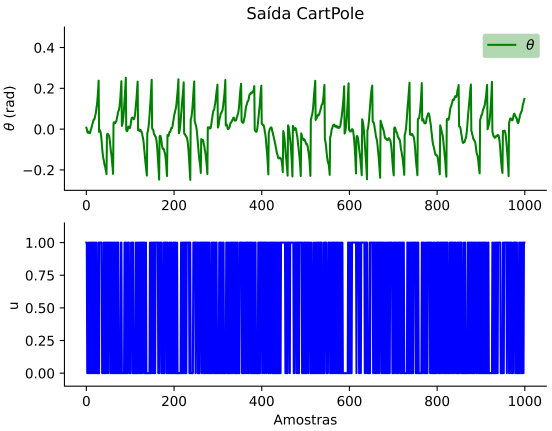


Figura 22: Aplicando o PRBS.

É importante observar que o ambiente CartPole reinicia automaticamente sempre que atinge um desvio de aproximadamente $\pm 12^{\circ}$ (equivalente a $\pm 0,209$ radianos). Essa característica representa um desafio, pois resulta em episódios muito curtos, dificultando a estimativa do modelo ARX. Inicialmente, uma abordagem para solucionar esse problema poderia envolver a redução do intervalo de tempo T_b. Isso implicaria em uma diminuição no tempo de amostragem do sistema, permitindo que as ações sejam executadas em períodos mais curtos e, assim, estendendo a duração dos episódios.

No entanto, é crucial observar que essa abordagem não é viável no contexto do ambiente CartPole no Gymnasium. Isso se deve ao fato de que a taxa de amostragem no ambiente CartPole não é diretamente ajustável. O ambiente CartPole foi projetado para simular a física real associada ao problema de um pêndulo invertido sobre um carro, e a taxa de amostragem é intrínseca a essa simulação. Portanto, modificar diretamente a taxa de amostragem do ambiente não é uma opção disponível. Esse aspecto limita a capacidade de ajuste do tempo de amostragem para atender às necessidades específicas do modelo ARX no contexto do problema CartPole.

Para enfrentar esse problema, foram realizadas 2 milhões de iterações no ambiente de simulação, sendo a entrada o sinal PRBS. Dentre essas iterações, o episódio mais longo, com o maior número de passos ou amostras, foi cuidadosamente selecionado para a estimativa do modelo ARX. Este episódio em particular se estende por 183 amostras, fornecendo assim uma base robusta e abrangente para a análise e modelagem do sistema. A escolha deste episódio mais extenso mostrado na Fig. 23 assegura que tenhamos dados significativos o suficiente para capturar as nuances e complexidades do sistema, permitindo a identificação do modelo ARX.

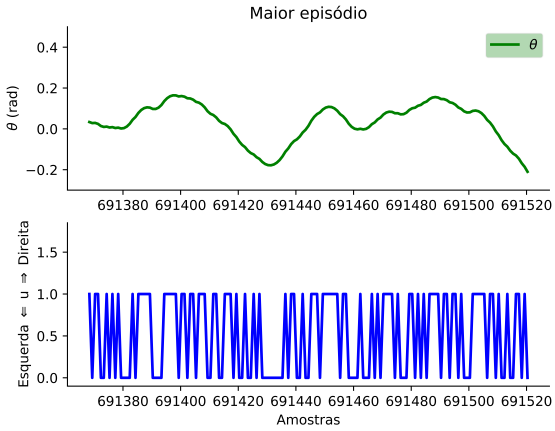


Figura 23: Maior episódio.

Foram utilizados 153 amostras para treinamento e 30 para validação. O SysIdentPy identificou o modelo ARX:

$$y(k) = 2 \cdot y(k-1) - 0,9936 \cdot y(k-2) - 0,0058 \cdot u(k-1), \quad (33)$$

sendo o $RMSE = 0,0057$.

Para lidar com a instabilidade do CartPole, adotou-se um horizonte de controle igual de predição ($H_p = H_c$), ajustando H_c com foco na eficiência do controle e tempo de execução, considere o tempo de execução (t) segundos para $H_c = 2$, conforme apresentado na Tabela 8.

Tabela 8: Horizonte de Controle em relação ao Tempo.

Horizonte de Conrole (H_c)	2	3	4	5	6
Tempo de Execução [s]	t	$1,07$	$1,21t$	$1,81t$	$3,58t$

A lógica é buscar o maior horizonte de controle viável, minimizando simultaneamente o tempo de execução, a escolha do horizonte de controle e predição foi de 4 passos à frente.

3.3 LunarLander - PD

3.3.1 *Implementação do Controlador PD*

O controlador PD foi retirado do repositório Lunar Lander OpenAIGym (Makhija, 2017) com algumas modificações. A ideia de modelar o sistema por meio de um modelo caixa branca não foi bem-sucedida, uma vez que se trata de um sistema MIMO, o que eleva o grau de complexidade. Seria necessária uma função de transferência para cada entrada e saída, tornando complicada a determinação dos parâmetros intrínsecos ao sistema. Assim, este repositório apresenta uma abordagem mais empírica, semelhante ao que ocorre no meio industrial, onde o PID é amplamente utilizado.

O primeiro passo é determinar as variáveis de processo a serem controladas, sendo os *boosters* do motor principal e secundário, usados respectivamente para controlar a altitude e o ângulo e/ou posição horizontal da nave.

A sonda ajusta-se com base nos sensores para mi-

nimizar erros ao longo do tempo, utilizando controle proporcional-derivativo (PD). Altitude, ângulo e velocidades são conhecidos em cada etapa. O erro é calculado como a diferença entre os *setpoints* e as medições atuais, permitindo controle proporcional. As velocidades são usadas para o controle derivativo. Para a implementação do PD, os pontos de ajuste (*setpoints*) serão definidos, conforme apresentado nas Fig. 24 e Fig. 25.

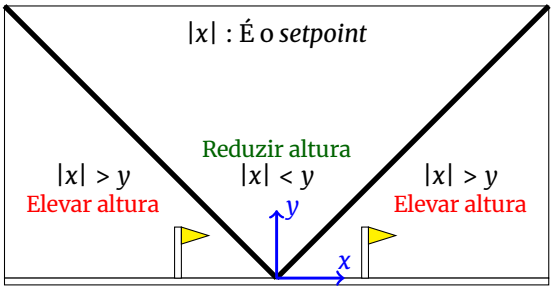


Figura 24: *Setpoint:* Altura.

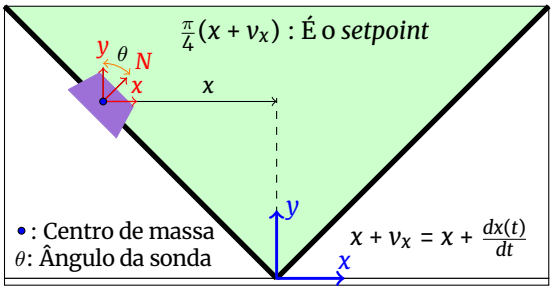


Figura 25: *Setpoint:* Ângulo.

O primeiro *setpoint* considerado é a altura, sendo o módulo da posição x, conforme a Fig. 24. Se a sonda estiver dentro do triângulo, deve descer, caso contrário, deve subir. Isso define o termo proporcional. Para o termo derivativo, utiliza-se a velocidade linear em y:

$$y_{PD} = k_{p2} \cdot (|x| - y) + k_{d2} \cdot v_y. \tag{34}$$

É crucial manter uma inclinação constante da nave em direção ao seu objetivo, pois isso determina a direção do impulso do propulsor principal. Para implementar essa abordagem, utiliza-se o *setpoint* $x + v_x$, onde v_x é a taxa de variação em x, entendendo essa expressão como x_{t+1} para minimizar θ. Quando a sonda está nas bordas extremas do triângulo, ela deve se inclinar 45° em direção a plataforma, com essa inclinação diminuindo à medida que a sonda se aproxima do alvo (0,0), de acordo com o *setpoint* x_{t+1}. O controle proporcional é aplicado, onde a posição x diminui à medida que a sonda se aproxima do alvo, enquanto o termo derivativo utiliza a velocidade angular:

$$\theta_{PD} = k_{p2} \cdot \left[\frac{\pi}{4} \cdot (x + v_x) - \theta\right] + k_{d2} \cdot v_\theta. \tag{35}$$

Todos os elementos essenciais foram identificados, exceto pelos valores apropriados dos quatro parâmetros k_{p1}, k_{d1}, k_{p2} e k_{d2}. Para determinar esses pesos, foi adotada a técnica de Otimização por Subida de Encosta. Ao que tudo indica, diferentemente do que é feito no repositório LunarLander OpenAIGym (Makhija, 2017), onde k_p e k_d são iguais no PD referente ao ângulo e à altura, o que leva a crer que foi sintonizado manualmente. Essa metodologia inicia com a premissa de que todos os parâmetros são inicialmente nulos (sem controle). Após cada tentativa de aterrissagem da sonda e avaliação da pontuação, os parâmetros são ajustados com pequenas variações aleatórias. Se a pontuação da sonda melhorar, os novos valores são mantidos e o processo é repetido. Caso contrário, os novos valores são descartados e tenta-se adicionar ruído aleatório novamente. Os valores encontrados foram: $k_{p1} = 14,028$, $k_{d1} = -17,842$, $k_{p2} = 11,957$ e $k_{d2} = -6,91$.

3.4 GPC - LunarLander

3.4.1 *Formulação e Aplicação*

As trajetórias de referência (w) do LunarLander são um pouco mais complexa, o que levou a adoção de algumas estratégias.

Sabe-se que o ambiente no qual a sonda interage é traduzido por meio de um plano cartesiano. Essa abordagem permite conhecer valores como posição e direção, aspectos de suma importância na navegação da sonda.

Com intuito de implementar o GPC, foi empregado uma matriz rotação para rotacionar o plano cartesiano em 45° (o motivo será explicado adiante), por ser bidimensional, tem-se:

$$\begin{bmatrix} p_x \\ p_y \end{bmatrix} = \begin{bmatrix} cos\phi & sen\phi \\ -sen\phi & cos\phi \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}, \tag{36}$$

a ideia é rotacionar o plano cartesiano em 45°, conforme ilustrado na Fig. 26, logo $\phi = 45°$.

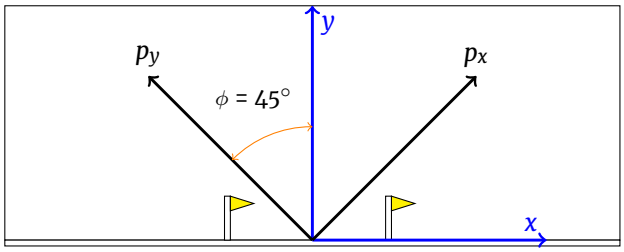


Figura 26: Plano cartesiano do LunarLander.

Com o plano cartesiano rotacionado, é possível estimar três trajetórias de referência. A primeira, em ciano, é definida por $p_x = p_y$. Similarmente ao CartPole, o ângulo θ do LunarLander deve estar próximo de zero, representando a segunda trajetória de referência. O próximo passo envolve determinar a terceira trajetória de referência para a velocidade linear em p_y, expressa por $v_y = -0,725 \cdot p_y - 0,125$. Esta arranjo será mostrado na Fig. 27.

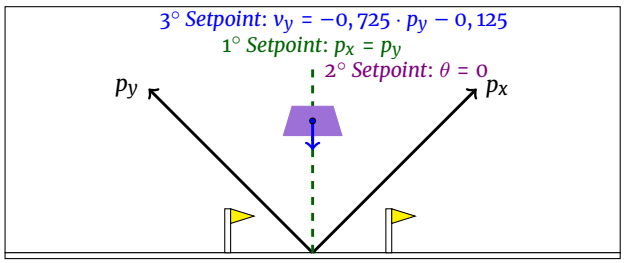
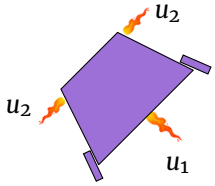


Figura 27: *Setpoints* para o GPC.

Para facilitar a identificação do modelo ARX, os propulsores auxiliares de orientação direita e esquerda foram considerados como um só (u_2), assumindo 1 para direita e -1 para esquerda, com 0 representando a condição desligada. Para o propulsor principal (u_1), 1 indica o acionamento e 0 representa que está desligado, como ilustrado na Fig. 28.



Ambiente	u_1	u_2
0	0	0
1	0	1
2	1	0
3	0	-1

Figura 28: Relação de u_2 e u_1 com o ambiente.

Em relação ao incremento de controle (Δu), uma metodologia alternativa foi considerada para o ambiente LunarLander. Enquanto o MPC tradicionalmente penaliza o Δu com base em sua magnitude, neste contexto específico, a penalização varia com os propulsores: auxiliares (u_2) têm penalização de 0,03, o principal (u_1) de 0,3 e nenhum acionamento não é penalizado. Dessa forma, a proposta é focar no controle absoluto u em vez do incremento Δu. Quando o horizonte de controle (H_c) é menor que o de previsão (H_p), u é mantido em 0, permitindo respostas que maximizam a recompensa do ambiente.

Conforme mencionado anteriormente, a metodologia de aplicação do modelo define a função de custo conforme a Eq. (37):

$$J(k) = \sum_{j=d}^{h_p} [\{\alpha \cdot (\hat{p}_x(j+k|k) - \hat{p}_y(j+k|k)) - \beta \cdot \hat{\theta}(j+k|k) + \delta \cdot (\hat{v}_y(j+k|k) - 0,725 \cdot \hat{p}_y(j+k|k) - 0,125)\}^2]^{\frac{1}{2}}, \tag{37}$$

sujeito a:

$$Propulsor = \begin{Bmatrix} 0 & 0 & 1 & 0 \\ -1 & 0 & 0 & 1 \end{Bmatrix} \begin{matrix} u_1 \Rightarrow \text{Propulsor principal.} \\ u_2 \Rightarrow \text{Propulsor auxiliar} \end{matrix}$$

As entradas devem ser intercaladas ou completamente desligadas. Em outras palavras, se uma estiver ativa, a outra deve permanecer desligada, ou ambas devem estar desligadas simultaneamente. O horizonte de controle foi definido como igual a 2 ($H_c = 2$) e o horizonte de predição como 4 ($H_p = 4$). Esses valores foram definidos experimentalmente e, apesar de serem pequenos, apresentaram uma resposta sólida e um tempo de execução eficaz.

3.4.2 *Estimação do Modelo ARX*

Nesse processo, optou-se por empregar uma entrada aleatória dentro das possíveis entradas, com o objetivo de obter um modelo para as coordenadas p_x e p_y, θ e v_y (velocidade linear em y). Durante a obtenção do modelo, observou-se uma correlação de cada uma das saídas com uma entrada específica, conforme evidenciado pela Tabela 9.

Tabela 9: Relação entre entradas e saídas.

Saída	p_x	θ	p_y	v_y
Entrada	u_2	u_2	u_1	u_1

Dessa forma, por meio da metodologia começou-se a investigar várias alternativas para cada entrada. A cada iteração, um valor era selecionado aleatoriamente dentre as opções disponíveis. Para u_1, o intervalo considerado foi [0, 2] do ambiente e [0, 1] para a identificação, enquanto u_2 apresentava três possíveis respostas [0, 1, 3] e [0, 1, -1] para a identificação.

Novamente, foi usado o SysIdentPy para estimar os modelos ARX para os itens já mencionados:

$$p_x(k) = 1,998 \cdot p_x(k-1) - 0,998 \cdot p_x(k-2) - 7,364 \cdot 10^{-5} \cdot u_2(k-1) - 4,797 \cdot 10^{-4}, \tag{38}$$

$$p_y(k) = 1,992 \cdot p_y(k-1) - 0,992 \cdot p_y(k-2) + 8,174 \cdot 10^{-5} \cdot u_1(k-1) - 3,945 \cdot 10^{-4}, \tag{39}$$

$$\theta(k) = -1,989 \cdot \theta(k-1) + 0,989 \cdot \theta(k-2) + 1,72 \cdot 10^{-4} \cdot u_2(k-1), \tag{40}$$

$$v_y(k) = 1,25 \cdot v_y(k-1) - 0,251 \cdot v_y(k-2) - 1,379 \cdot 10^{-2} \cdot u_1(k-1) + 2,364 \cdot 10^{-3}. \tag{41}$$

Foram utilizadas 70 amostras para treinamento e 15 para validação. Os resultados da validação estão detalhados na Tabela 10.

Tabela 10: Validação.

Modelo ARX	p_x	p_y	θ	v_y
RMSE	0,009	0,0355	0,007	0,0549

Os modelos foram capazes de simular o LunarLander com precisão.

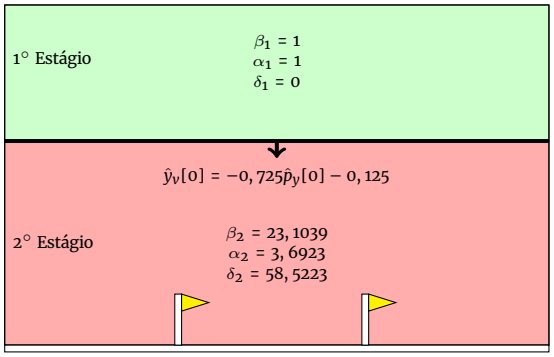
3.4.3 *Determinação dos Pesos da Função Custo*

Inicialmente, a estimação dos pesos foi feita por tentativa e erro, onde foram identificados alguns pontos importantes posteriormente utilizados. O principal foi a separação do ambiente lunar em dois estágios. O primeiro estágio consiste em verificar se a velocidade da sonda é inferior à

trajetória de referência. Nesse caso, o peso δ na Eq. (37), sendo zero, ajuda o modelo a não tomar decisões equivocadas, como inclinar em um ângulo θ grande e utilizar os motores auxiliares para acelerar e alcançar a referência. Dessa maneira, a sintonização manual se tornou mais tranquila, onde no primeiro estágio o algoritmo monitora apenas com os *setpoints* de posição e inclinação angular. No segundo estágio, a velocidade linear em y passa a fazer parte dos *setpoints*.

Depois de muitas tentativas, o GPC apresentou resultados semelhantes ao PD do repositório LunarLander OpenAIGym (Makhija, 2017). No entanto, ao adotar um algoritmo de Subida de Encosta para estimar os parâmetros proporcionais e derivativos do PD deste trabalho, houve uma melhoria significativa no desempenho do PD. Assim, um ajuste manual do GPC para se equiparar ao PD tornou-se complicado. A proposta consistiu em aplicar o algoritmo de Subida de Encosta para estimar os pesos do segundo estágio (α_2, β_2, e δ_2), enquanto os pesos do primeiro estágio foram previamente definidos manualmente como $\alpha = \beta = 1$ e $\delta = 0$, mantidos devido ao seu desempenho satisfatório. Os pesos empregados se encontram na Fig. 29.

Figura 29: Pesos da Função de Custo para o LunarLander.



Os pesos (α_2, β_2 e δ_2) derivados pelo algoritmo de subida de Encosta resultaram em uma notável melhoria no desempenho do GPC, em comparação com os ajustes manuais realizados por tentativa e erro. Essa otimização automática permitiu uma sintonia mais precisa e eficiente dos parâmetros do controlador, levando a um funcionamento mais eficaz do sistema de controle.

4 Resultados e Discussões

4.1 CartPole

Ambas as técnicas demonstraram capacidade de estabilizar o CartPole, como evidenciado pela Fig. 30. No entanto, a capacidade antecipativa do GPC fundamenta-se na consideração não apenas do estado presente do sistema, mas também na previsão de seu comportamento futuro. Apesar disso, neste caso específico, o horizonte de previsão e controle relativamente curto limita a visão do futuro pelo GPC. Como consequência, ocorre a persistência de uma ação de controle específica ($\Delta u = 0$), o que exacerba a aceleração angular do CartPole.

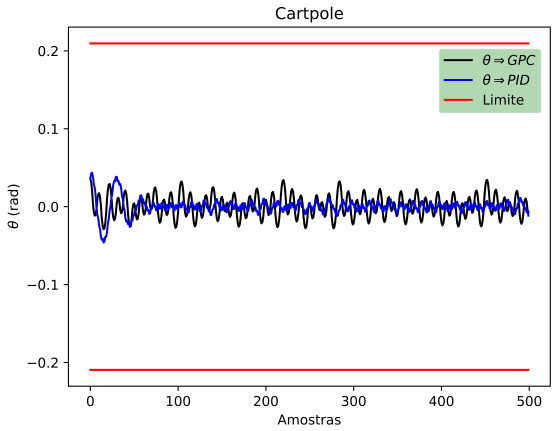


Figura 30: Comparação - Condições iniciais idênticas.

O PID manteve o CartPole mais próximo da referência ($\theta = 0$):

Tabela 11: PID x CPC.

Controle	$\bar{\theta}$	θ_{max}	Etapas	Tempo [s]
PID	$1,105 \cdot 10^{-5}$	$4,639 \cdot 10^{-2}$	5000	9,042
GPC	$-9,754 \cdot 10^{-5}$	$7,333 \cdot 10^{-2}$	5000	15,142

A consequente ampliação da velocidade angular do pêndulo implica uma força contrária prolongada para corrigir (θ) quando há inversão de sinal. Esse processo se repete em um ciclo contínuo.

Esse horizonte de controle e previsão ($H_c = H_p$) curto se deve ao problema de otimização inteira. Para sua resolução, foi utilizado o método de pesquisa em grade, que é custoso computacionalmente. Mesmo após um estudo indicar que $H_c = H_p = 4$ seria a melhor opção, o GPC ainda demonstra ser mais lento que o PID. Se houver um aumento de uma unidade nos horizontes, o ganho não é tão significativo em termos de desempenho, no entanto, o tempo de resposta aumenta consideravelmente.

4.2 LunarLander

Tanto o GPC quanto o PD conseguiram realizar a aterrissagem da sonda lunar, como ilustrado na Fig. 31. Observa-se que o GPC adota uma abordagem mais cautelosa em relação ao PD, realizando mais etapas para completar o pouso. Isso se deve ao fato de que o GPC inicia a desaceleração da sonda mais cedo, resultando em um pouso mais suave, embora demandando um número maior de etapas.

Para a validação, foram realizadas 1000 tentativas de pouso, conforme detalhado na Tabela 12, nas quais ambos os controladores demonstraram excelente desempenho no controle do pouso lunar. As tentativas ocorreram em ambientes totalmente aleatórios (sem a utilização de sementes). Em termos de pontuação e sucesso dos pousos, tanto na plataforma quanto os "perfeitos"(+200 pontos),

o GPC apresentou desempenho superior, alcançando taxas de sucesso de 93, 6% e 88, 6%, respectivamente. Esse resultado também se traduziu em uma pontuação média mais alta.

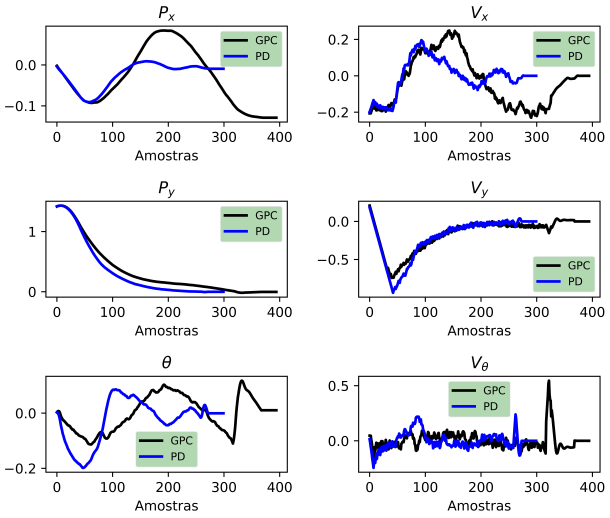


Figura 31: Comparação - Condições iniciais idênticas.

Tabela 12: PID x GPC - *LunarLander*.

Controle	PD	GPC
Pouso	852	936
Pouso com +200 pontos	851	886
Etapas	215812	380073
Tempo de execução [s]	129, 131	621, 169
Pontuação média	218, 023	231, 035

O GPC apresentou um tempo de tomada de decisão superior. Aqui, dois pontos são relevantes: primeiro, devido a algumas limitações na imposição dos *setpoints*, o PD foi ajustado de maneira mais audaciosa, sempre procurando obter pousos com pontuação superior a 200 pontos. Tanto é que, dos 852 pousos, 851 atingiram mais de 200 pontos, o ajuste do PD por ser mais arrojado, com menos dependência dos propulsores e, na simulação, requer um número menor de etapas em relação ao GPC. Em segundo lugar, nas ocasiões em que não conseguiu obter a pontuação para resolver o episódio, o PD perdeu o controle da sonda precocemente (em poucas etapas), ou seja, muitas das vezes caiu sem ao menos conseguir pousar de forma básica. Isso levou o PD a realizar 1000 tentativas de pouso em 215812 etapas, enquanto o GPC utilizou 380073 etapas. O tempo de tomada de decisão é calculado como $\frac{129,131}{215812} = 0,598ms$ para o PD e $\frac{621,169}{380073} = 1,634ms$ para o GPC (Esses tempos também incluem o tempo em que o ambiente LunarLander recebe a entrada, a processa e retorna o espaço de observação.). Apesar de o tempo de reação do PD ser quase três vezes inferior ao do GPC, ambos são rápidos, especialmente considerando que o GPC resolve um problema de otimização em cada etapa utilizando o Método de Pesquisa em Grade.

Para uma avaliação mais abrangente, foram incorporados distúrbios ao ambiente do *LunarLander*. Esses distúrbios incluem uma turbulência ajustada para 0, 75, um vento definido em 2, 5, e a gravidade foi ajustada para $10,5m/s^2$. Este experimento será conduzido através de 1000 tentativas de pouso novamente.

Tabela 13: PID x GPC - *LunarLander* com distúrbios.

Controle	PD	GPC
Pouso	655	847
Pouso com +200 pontos	654	712
Etapas	243201	370242
Tempo de execução [s]	122, 113	746, 245
Pontuação média	163, 532	177, 851

Mesmo o PD e o GPC sendo implementados no ambiente LunarLander sem quaisquer distúrbios e com uma gravidade padrão de $9,8\frac{m}{s^2}$, conseguiram um desempenho satisfatório quando expostos a distúrbios. De maneira geral, o GPC se mostrou um pouco mais robusto em relação ao PD, conforme apresentado na Tabela 13.

5 Conclusão

O objetivo do presente trabalho era integrar as técnicas de controle, unindo a teoria de controle clássica e avançada para serem empregadas em dois ambientes do Gymnasium, uma API para aprendizado por reforço. Ao longo do processo, mesmo que toda a teoria empregada sendo consolidada na literatura, as formulações precisaram passar por algumas adaptações, exigindo criatividade. Por exemplo, para obter a função de transferência do CartPole, foi necessário que ela representasse o sistema da melhor maneira possível, então foram utilizados fundamentos físicos. Além da identificação de sistemas, outro ponto foi o uso da ferramenta de otimização GEKKO para ajustar os parâmetros do PID, que se mostrou uma ferramenta interessante para essa tarefa no *Open Source*.

Para implementar o PD no LunarLander, inicialmente planejou-se seguir os passos adotados no CartPole. No entanto, a obtenção da Função de Transferência revelou-se custosa devido ao sistema ser MIMO (Múltiplas Entradas e Múltiplas Saídas). Além disso, a determinação dos parâmetros intrínsecos do sistema por experimentação mostrou-se inviável. Assim, optou-se por uma abordagem mais empírica, semelhante à aplicação industrial de controladores PID, onde muitos deles vêm com funcionalidades de ajuste automático incorporadas (*Auto-tuning*). Neste trabalho, o algoritmo de Subida de Encosta foi utilizado para ajustar os parâmetros conforme a recompensa, que se mostrou uma estratégia interessante para a sintonizar PID.

A implementação do controle preditivo generalizado foi auxiliada pelo pacote SysIdentPy, uma ferramenta eficiente para identificação de sistemas. Um dos maiores desafios na identificação foi em relação ao CartPole, onde mesmo com o auxílio do PRBS, foi necessário fixar o número de passos em dois milhões para selecionar o maior

episódio, com o intuito de utilizá-lo na identificação. Este problema foi devido à natureza instável do sistema; no entanto, o controle do CartPole foi efetuado com sucesso. No LunarLander, o SysIdentPy obteve ótimos modelos com poucas amostras, e os ajustes dos pesos na função custo foram realizados por meio de tentativa e erro e também com auxílio do algoritmo de Subida de Encosta. O resultado atendeu aos objetivos estabelecidos.

Um dos principais obstáculos na implementação do GPC foi lidar com um problema de otimização inteira, que foi resolvido pelo Método de Pesquisa em Grade, um algoritmo com complexidade $O(n!)$, ou seja, o tempo de execução aumenta fatorialmente em relação à entrada. Isso levou o PID a ser mais ágil no tempo de resposta, em ambos os casos.

Em resumo, tanto o método PID quanto o MPC demonstraram capacidade de controlar sistemas complexos. Porém, o PID apresentou melhor desempenho no controle do CartPole, enquanto o GPC se destacou no controle do LunarLander. Ambos os métodos foram bem-sucedidos nos dois ambientes com espaço de ação discreto, alcançando pontuações significativas em comparação com os algoritmos descritos na literatura (Hafiz et al., 2023).

Este estudo contribui para a compreensão e aplicação prática dessas técnicas em ambientes virtuais, utilizando ferramentas acessíveis a toda a comunidade. Ele pode ser perfeitamente replicado, por exemplo, no ensino de controle de maneira prática, pois vai além da teoria, permitindo aos alunos verem seus controles em ação. O Gymnasium possui uma renderização que permite visualizar o ambiente interagindo com o controlador, tornando-se uma excelente ferramenta não apenas para o aprendizado de máquina, mas também para o compreendimento de técnicas de controle.

Os trabalhos futuros incluiriam a implementação de pacotes de otimização inteira do Python, como o Python-MIP. Isso permitiria estender o horizonte de controle, uma vez que o pacote oferece algoritmos muito mais eficientes do que o Método de Pesquisa em Grade para resolver o problema de otimização inteira. Outro ponto interessante seria a utilização de algoritmos genéticos para sintonizar os controladores. Isso aumentaria a ênfase na otimização, abrangendo um aspecto crucial no campo da engenharia atualmente. Com isso, espera-se uma melhora significativa tanto no tempo de execução quanto na capacidade de resolver o ambiente de maneira mais eficiente possível.

Referências

Aguirre, L. (2015). *Introdução à Identificação de Sistemas*, UFMG, Belo Horizonte, BR.

Barto, S. e. A. (2023). Cartpole. Acesso em dezembro 2023. Disponível em https://gymnasium.farama.org/environments/classic_control/cart_pole/.

Beal, L., Hill, D., Martin, R. and Hedengren, J. (2018). Gekko optimization suite, *Processes* **6**(8): 106. https://doi.org/10.1016/0005-1098(87)90088-4.

Boubaker, O. (2012). The inverted pendulum: A fundamental benchmark in control theory and robotics, *International conference on education and e-learning innovations*, IEEE, Sousse, TN, pp. 1–6. https://doi.org/10.1109/ICEELI.2012.6360606.

Camacho, E. F. and Bordons, C. (2004). *Model Predictive Control*, Springer, Berlin, DE.

Carneiro, A. L. C. (2020). Algoritmos de otimização: Hill climbing e simulated annealing. Acesso em dezembro 2023. Disponível em https://medium.com/data-hackers/algoritmos-de-otimiza%C3%A7%C3%A3o-hill-climbing-e-simulated-annealing-3803061f66f0.

Castrucci, P. B. d. L., Bittar, A. and Sales, R. M. (2011). *Controle Automático*, 1 edn, LTC, Rio de Janeiro, BR.

Clarke, D. W., Mohtadi, C. and Tuffs, P. S. (1987a). Generalized predictive control—part i. the basic algorithm, *Automatica* **23**(2): 137–148. https://doi.org/10.1016/0005-1098(87)90087-2.

Clarke, D. W., Mohtadi, C. and Tuffs, P. S. (1987b). Generalized predictive control—part ii extensions and interpretations, *Automatica* **23**(2): 149–160. https://doi.org/10.1016/0005-1098(87)90088-4.

Deulkar, P. and Hanwate, S. (2020). Analysis of pso-pid controller for cstr temperature control, *2020 IEEE First International Conference on Smart Technologies for Power, Energy and Control (STPEC)*, IEEE, Nagpur, IN, pp. 1–6. https://doi.org/10.1109/STPEC49749.2020.9297750.

Hafiz, A. M., Hassaballah, M., Alqahtani, A., Alsubai, S. and Hameed, M. A. (2023). Reinforcement learning with an ensemble of binary action deep q-networks., *Computer Systems Science & Engineering* **46**(3). http://dx.doi.org/10.32604/csse.2023.031720.

Klimov, O. (2023). Lunarlander. Acesso em dezembro 2023. Disponível em https://gymnasium.farama.org/environments/box2d/lunar_lander/.

Lacerda, W. R., da Andrade, L. P. C., Oliveira, S. C. P. and Martins, S. A. M. (2020). Sysidentpy: A python package for system identification using narmax models, *Journal of Open Source Software* **5**(54): 2384. https://doi.org/10.21105/joss.02384.

Makhija, P. (2017). Lunarlander_openaigym. Acesso em novembro 2023. Disponível em https://github.com/piyushmakhija5/LunarLander_OpenAIGym.

Minorsky, N. (1922). Directional stability of automatically steered bodies, *Journal of the American Society for Naval Engineers* **34**(2): 280–309. https://doi.org/10.1111/j.1559-3584.1922.tb04958.x.

Nelles, O. (2001). *Nonlinear system identification. From classical approaches to neural networks and fuzzy models*, Springer Berlin. https://doi.org/10.1007/978-3-662-04323-3.

Ogata and Katsuhiko (2010). *Engenharia de controle moderno*, 5 edn, Prentice Hall do Brasil, São Paulo, BR.

Roberts, E. (2021). sing pid to cheat an openai challenge. Acesso em setembro 2023. Disponível em https://ethanr2000.medium.com/using-pid-to-cheat-an-openai-challenge-f17745226449.