ENEMP 2022
XL CONGRESSO BRASILEIRO DE SISTEMAS PARTICULADOS
23 a 26 de outubro de 2022
Uberlândia/MG

CONTROLE INTELIGENTE DE UMA COLUNA DE FLOTAÇÃO POR NEUROEVOLUÇÃO

RAFAEL BRUSIQUESI MARTINS[1], LUÍS C. OLIVEIRA-LOPES [1, *]

[1]Universidade Federal de Uberlândia, Faculdade de Engenharia Química
*e-mail: LCOL@UFU.BR

RESUMO – No meio industrial, a demanda por controle de processos mostra-se crescente, e cada vez mais tal paradigma é exigido para a operação de sucesso de processos desafiadores na indústria. Simultaneamente, nos cenários atuais há grande disponibilidade de informações sobre os processos, medidas de operações normais e anormais (históricas e em linha) e de elevado poderio computacional cada vez mais barato e resiliente. Nesse trabalho é proposto um paradigma de controle de processos utilizando neuroevolução, um conjunto de algoritmos que unem a capacidade regressora universal das redes neuronais com o poder de otimização global extremamente eficiente dos algoritmos genéticos. Aliados a uma nova formulação para controle de processos, chamado por simplicidade de controle inteligente por usar ferramentas de aprendizado e de inteligência artificial na tomada de decisão. O trabalho é ilustrado com a geração de informações por simulação e posterior obtenção de um controlador bem sucedido no controle de um modelo identificado de uma coluna de flotação de difícil controle, provando a eficácia do método proposto.

INTRODUÇÃO

Os avanços tecnológicos da indústria têm gerado estruturas cada vez mais complexas em termos de operações e suas agregações dentro do corpo da planta, ou na própria conformação estrutural de seus produtos gerados; e em um contexto onde a economia exige uma crescente adaptação da indústria para a minimização de custos e maximização de lucros, é de extrema importância a adoção de princípios de automação industrial, assim como a necessidade de controle de alta tecnologia para a própria viabilização de processos que sem este, não seriam possíveis, como por exemplo, a estabilização de reatores de fusão nuclear (Degrave et al., 2022). Dessa maneira, conclui-se que o avanço tecnológico é imprescindível para a oxigenação dos meios de produção e do próprio aumento da capacidade humana de transformar o mundo que a cerca.

Dentro do escopo de produção de valor material para o planeta, a indústria química ostenta fatia significativa da economia mundial, sendo os processos que a compõem, de grande interesse para viabilização e estudo. Sob tal ótica, pode-se inferir a necessidade de tais integrações tecnológicas citadas anteriormente dentro dos mais diversos processos industriais.

Assim, pode-se observar que a vasta maioria dos processos de interesse comercial possui natureza dinâmica complexa, afinal, uma planta de produção, geralmente configura-se como uma concatenação de diversos processos em série e em paralelo, os quais são comumente conhecidos como operações unitárias, não suficientemente, cada operação unitária compõe-se de diversos fenômenos físicos de natureza não linear, isso implica em sistemas de comportamento dinâmico de alta ordem e difícil controle, problema que deve ser enfrentado, pois, como já pontuado, vários desses processos necessitam imperativamente de controle para a sua existência e operação ótima e segura.

O presente trabalho introduz uma técnica alternativa de controle multivariável não linear de escopo geral utilizando redes neuronais e algoritmos genéticos aplicada em um estudo de caso para uma coluna de flotação para sistemas particulados, assim como a verificação de sua

eficácia por simulação dinâmica do modelo da coluna identificado por Persechini e Peres (2000), um processo de difícil controle, sendo assim, um bom indicador para a validade do método de controle proposto.

O CONTROLE INTELIGENTE

O algoritmo de controle baseia-se na utilização de uma estrutura de aprendizado conhecida como neuroevolução, em que são utilizados algoritmos genéticos para a otimização de uma rede neuronal, de modo a que ela performe determinadas ações de interesse do engenheiro.

Na formulação da heurística proposta, o objetivo da rede neuronal é aproximar uma política de ação adequada para controlar com sucesso o sistema no qual o agente se encontra. Dessa maneira, os parâmetros da rede devem ser otimizados por algum método de otimização; na literatura, grande parte das aplicações para redes neuronais são executadas como a solução de um problema de regressão diferencial, ou seja, dado um conjunto de dados de resposta mapeada, é calculado um desvio entre a resposta da rede e a resposta rotulada, gerando uma função objetivo diferenciável que pode ser otimizada por métodos baseados em gradientes, como gradiente descendente, gradiente descendente estocástico, otimização ADAM (*Adaptive Moment Estimation*) entre outros, caracterizando aprendizado supervisionado. (Kinsley, Kukiela, 2020).

Entretanto, para problemas de controle, não é tarefa simples obter dados de uma ação de controle ótima de modo a aplicar-se aprendizado supervisionado baseado em regressão, nota-se inclusive, que a obtenção de tais dados representaria a solução do próprio problema de controle. Dessa forma, é proposto uma metodologia de aprendizado não supervisionado para a otimização dos parâmetros da rede neural, no presente trabalho, sendo usado um algoritmo genético de escopo no conjunto dos números reais em formulação vetorial n-dimensional $\mathbb{R}^n$.

O Algoritmo Genético

Os algoritmos genéticos são uma classe de heurísticas que tentam capturar as propriedades de seleção apresentadas pelos sistemas naturais, otimizando um determinado objeto em respeito à sua aptidão à sobrevivência em determinado ambiente, dessa forma, obtém-se um mecanismo de escopo mais generalista, podendo otimizar parâmetros e estruturas diversas, não necessariamente dependendo das abordagens clássicas de otimização analítica e diferencial. Dada essa robustez intrínseca dos algoritmos genéticos, faz-se possível a otimização de sistemas das mais variadas sortes, variando desde funções analíticas bem definidas matematicamente, passando por sistemas reais não facilmente parametrizáveis e até mesmo objetos abstratos que interagem com algum ambiente de referência; com base nessas propriedades, pode-se encontrar um imenso espaço de aplicação prática destes métodos.

A definição clássica do algoritmo genético consiste na otimização dos parâmetros de uma função n-dimensional de modo a que se obtenha uma boa aproximação de seu ponto crítico de Hessiana negativa ou positiva definida a depender da natureza de maximização ou minimização do problema

O algoritmo estudado percorrerá o espaço de busca caracterizado como domínio da função objetivo, utilizando de operadores genéticos baseados na seleção natural de Darwin (1869), criando gerações de populações geradas com base em regras bioinspiradas e perturbações pseudoaleatórias nos parâmetros de cada solução, dessa maneira, espera-se que o algoritmo consiga percorrer tal domínio de maneira esparsa, porém orientada, alcançando em hipótese a vizinhança do ponto de ótimo global da função a ser otimizada.

Tais operadores genéticos podem ser desenvolvidos de diversas maneiras, e então serão aplicados a um genoma, definido neste artigo como um tensor de primeira ordem de dimensão n, que representa os parâmetros otimizáveis da função objetivo.

Neste trabalho, a implementação do algoritmo genético seguiu os princípios clássicos de Seleção, Mutação, Cruzamento e Elitismo (Omelianenko, 2019), sendo descritos a seguir.

Seleção: O método de seleção utilizado no algoritmo desenvolvido foi a seleção por torneio, que foi implementada retirando aleatoriamente amostras de k genomas e

selecionando o melhor indivíduo em termos de aptidão desse subgrupo; este processo é repetido pelo mesmo número de vezes que o tamanho da população definida nos hiperparâmetros do algoritmo.

Mutação: A operação de mutação consiste basicamente na substituição de um parâmetro do genoma por outro gerado aleatoriamente no mesmo intervalo.

Cruzamento: É realizado o cruzamento de Radcliff, que define cada parâmetro de um par de genomas selecionados para o *crossover* com base nas Equações 1 e 2.

$$p_{novo\,1} = \beta p_1 + (1-\beta)p_2 \quad (1)$$

$$p_{novo\,2} = \beta p_2 + (1-\beta)p_1 \quad (2)$$

Em que β representa um valor aleatório entre 0 e 1 e p representa cada parâmetro a ser modificado no cruzamento.

Elitismo: Consiste basicamente em perpetuar a existência de genoma bem cotado em termos da função objetivo para as próximas gerações, pode ser aplicado de diversas formas, no projeto realizado, o Elitismo consiste estritamente na ordenação dos genomas com base no seu valor de aptidão, posterior extração de um corte da lista resultante com base na quantidade de indivíduos elitistas desejada, definida por um hiperparâmetro, e na sua inserção no método de seleção para a próxima geração, retirando indivíduos sobressalentes para manter a população filha com o mesmo número de indivíduos que a população mãe.

Um compreensível e simples exemplo desse processo foi introduzido por Tomassini (1995), em que foi postulado um problema para encontrar um ponto crítico de uma função específica definida no conjunto dos números pertencentes a um conjunto $d' \subset \mathbb{R}$, descrita pela Equação 3.

$$f(x^*) = \min\left\{-\left|x \sin\left(\sqrt{|x|}\right)\right| \forall\, x \in d'\right\} \quad (3)$$

Em que d' é definido por todos os números igualmente espaçados dentro do intervalo [0, 512] que podem ser descritos por operações sobre um vetor binário descrito por 10 bits, dessa maneira, teremos uma granularidade numérica de $\frac{512}{2^{10}} = 0{,}5$. Objetiva-se minimizar a função utilizando o algoritmo.

O algoritmo genético básico é um bom candidato para satisfazer tal proposição. Aplicando uma transformação na função objetivo para a conversão do problema para outro de maximização, pode-se obter o resultado ilustrado na Figura 1.

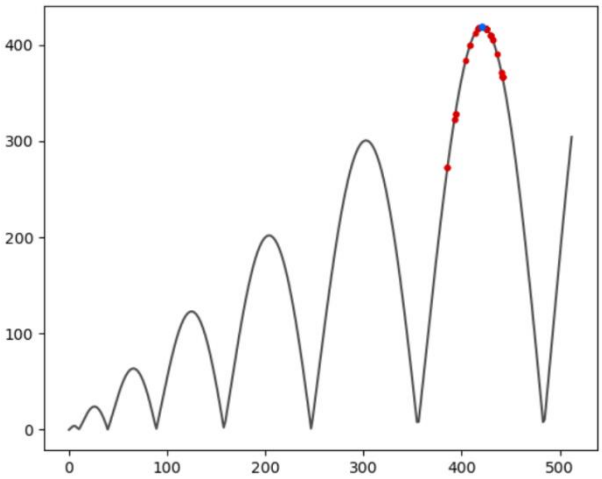

Figura 1: Otimização exemplo pelo AG-Ilustração.

É válido ressaltar que o a resolução de tal exemplo nas condições descritas pelo autor, exigiria a adaptação do algoritmo genético para sua formulação genômica discretizada em uma sequência de bits.

Neuroevolução

Uma promissora alternativa para a geração de agentes inteligentes para a operação em um sistema dinâmico de característica geral é o paradigma de neuroevolução, que consiste em otimizar os parâmetros de uma rede neuronal utilizando justamente a classe de algoritmos genéticos abordados na seção anterior. Dessa maneira, poder-se-á otimizar a rede neuronal com o uso de funções objetivo diferentes do que seu simples erro de output, possibilitando treinamentos mais específicas e, no presente caso, a utilização das redes neurais para gerar novas leis de controle para sistemas dinâmicos, em especial a coluna de flotação, que será abordada como exemplo de ilustração em seções subsequentes.

Uma rede neural consiste em um objeto matemático inspirado no funcionamento conhecido do cérebro dos seres biológicos inteligentes, consiste em uma rede

interconectada de neurônios, caracterizados matematicamente por números reais, submetidos a funções de ativação não lineares, de modo a que tal objeto funcione como um aproximador de função universal, podendo representar teoricamente qualquer função analítica e lógica, dado um número suficiente de neurônios (Aggarwal, 2018).

Assim, as redes neurais representam boas candidatas para realizar o mapeamento entre observações e ações dos agentes dentro de um sistema Markoviano (Sutton, 2018), reagindo bem na aplicação em ambientes de controle.

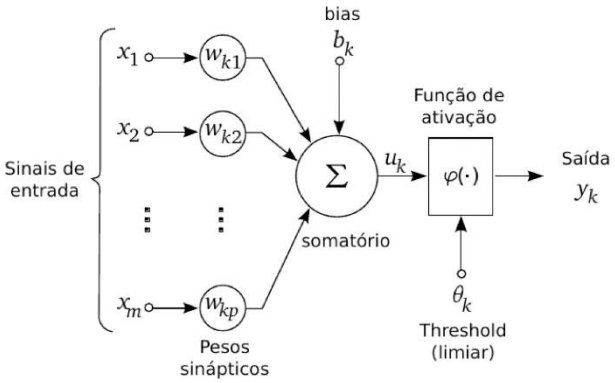

Figura 2: Modelo de rede neural, segundo McCulloch-Pitts (1943)

Para sua implementação, foi utilizada a definição clássica de rede neural, sendo essa composta por camadas de neurônios interligados que processa um sinal de entrada por cada uma até retornar um output na dimensão definida em sua topologia.

Assim, cada neurônio é ligado aos demais, e então ponderados por um peso multiplicador (*weight*), deslocados por um fator de *offset* (*bias*) e submetidos a uma função de ativação, para cada camada. O sinal passado por essa sequência é, então, computado por uma operação de multiplicação matricial dos pesos, e somado posteriormente ao vetor dos *biases*, foram implementadas três funções de ativação diferentes, sendo essas a Unidade Linear Retificada (ReLU), Sigmoide e Tangente Hiperbólica (TanH).

Percebe-se que a formulação das redes neurais nada mais é que uma função paramétrica de alta quantidade de parâmetros, caracterizando-se como um problema facilmente adaptável para o algoritmo genético até então desenvolvido, bastando codificar o genoma do algoritmo de maneira a que contemple os pesos e offsets que caracterizam a rede.

Para adaptar tal genoma, basta realizar operações de alteração topológica do tensor resultante de suas iterações para satisfazer a topologia tensorial de segunda ordem dos pesos e de primeira ordem dos valores de offset (biases), realizada essa tradução, a própria característica de otimização do algoritmo se encarregará de encontrar o vetor que cria a rede neural de melhor aptidão.

Outra formulação promissora para a estrutura de neuroevolução seria o algoritmo introduzido por Stanley (2002), conhecido como *Neuroevolution of Augmenting Topologies* (NEAT), ou sua versão refinada para subdivisão geométrica de regiões, procurando reproduzir a regionalidade das estruturas do cérebro humano, conhecida como *Hypercubed based NEAT* (Omelianenko, 2019). Tais formulações não serão abordadas neste trabalho.

O Problema de Controle

O problema de controle pode ser definido como a automação da tomada de decisão ótima, assim, supõe-se um sistema dinâmico como um sistema de equações diferenciais da forma:

$$\frac{d\boldsymbol{x}}{dt} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{d}, t) \tag{4}$$

Sendo $\boldsymbol{x}$, o vetor de variáveis do sistema, $\boldsymbol{u}$, o vetor de variáveis de entrada, $\boldsymbol{d}$, as perturbações do sistema, $\boldsymbol{f}$, uma função de natureza linear ou não, e t, o tempo.

O objetivo do problema de controle é selecionar uma dinâmica das variáveis manipuladas do sistema ($\boldsymbol{u}$), de maneira a que tal sistema se comporte de maneira desejada pelo engenheiro.

Assim, foram propostas inúmeras metodologias na literatura para resolver tal problema, em grande parte, envolvendo técnicas de linearização e simplificação das características do sistema para encontrar uma função de Lyapunov (Lyapunov, 1892) que garante a estabilidade do controle de tal sistema, gerando um modelo de controlador capaz de gerar uma resposta de entradas que atenda ao objetivo de controle.

Entretanto, para sistemas altamente não lineares, de difícil controle, ou objetivos de controle que não repousam sobre um estado estacionário fixo, a teoria de controle linear pode falhar, gerando a necessidade de encontrar modelos de controladores sob diferentes formulações, dessa maneira, é proposta uma definição adaptada do problema de controle para sistemas dinâmicos complexos sob o escopo do paradigma de aprendizado por reforço (*Reinforcement Learning*).

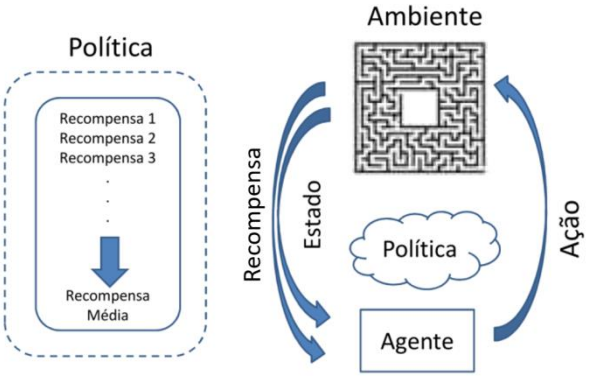

Figura 3: Formulação do problema de controle como Aprendizado por reforço (*Reinforcement Learning*). Adaptado de Sutton e Barto (2018).

Com base no esquema da Figura 3, pode-se reestruturar o problema de controle como o fornecimento do estado do sistema (ou uma observação do sistema) ao agente, que também recebe um sinal de recompensa, e posteriormente realiza uma ação de volta ao ambiente (sistema dinâmico), o que caracteriza uma malha com realimentação (*feedback*) de objetivo a obter uma política de ação de modo maximizar o valor esperado da soma das recompensas a serem recebidas pelo agente. (Sutton, 2018)

De posse dessa formulação, pode-se definir o tal agente como uma rede neuronal a ser otimizada pelo algoritmo de neuroevolução, a função de aptidão do algoritmo como o resíduo entre os valores de setpoint e as variáveis controladas em cada iteração, e haver-se-á um paradigma de aprendizado autônomo para sistemas dinâmicos de sorte generalista.

O Modelo da Coluna de Flotação

Para exemplificar a ideia apresentada, selecionou-se o controle multivariável de uma coluna de flotação, operação unitária geralmente utilizada para a separação de material particulado de uma corrente de líquido, caracterizando um processo de múltiplas fases.

Um modelo identificado de uma coluna de flotação está descrito no trabalho de Persechini et al. (2000), em que a partir de dados experimentais, é gerada uma matriz de funções de transferência no domínio de Laplace que assume a forma das Equações a seguir.

$$\begin{bmatrix} h \\ \varepsilon_{gcz} \\ Q_B \end{bmatrix} = \begin{bmatrix} G_{11} & G_{12} & G_{13} \\ G_{21} & G_{22} & G_{23} \\ G_{31} & G_{32} & G_{33} \end{bmatrix} \begin{bmatrix} Q_W \\ Q_g \\ Q_T - Q_F \end{bmatrix} \quad (4)$$

$$G_{11} = \frac{-(1{,}029\cdot10^{-3}s+2{,}3\cdot10^{-5})}{(s+4{,}02\cdot10^{-4})(s+1{,}92\cdot10^{-2})} \quad (5)$$

$$G_{12} = \frac{-(1{,}59\cdot10^{-4}s+4{,}33\cdot10^{-7})}{(s+4{,}02\cdot10^{-4})(s+7{,}981\cdot10^{-3})} \quad (6)$$

$$G_{11} = \frac{(1{,}029\cdot10^{-3}s+2{,}3\cdot10^{-5})}{(s+4{,}02\cdot10^{-4})(s+1{,}92\cdot10^{-2})} \quad (7)$$

$$G_{21} = \frac{7{,}6\cdot10^{-5}}{s+1{,}92\cdot10^{-2}} \quad (8)$$

$$G_{22} = \frac{7{,}78\cdot10^{-5}}{s+7{,}81\cdot10^{-3}} \quad (9)$$

$$G_{23} = \frac{-7{,}6\cdot10^{-5}}{s+1{,}92\cdot10^{-2}} \quad (10)$$

$$G_{31} = s(K_1G_{11} + K_2G_{21}) \quad (11)$$

$$G_{32} = s(K_1G_{12} + K_2G_{22}) \quad (12)$$

$$G_{33} = 1 + s(K_1G_{13} + K_2G_{23}) \quad (13)$$

Pode-se representar o sistema no espaço de estados na formulação dinâmica usual para sistemas lineares, conforme as Equações (14)-(15).

$$\frac{dx'}{dt} = \boldsymbol{Ax'} + \boldsymbol{Bu'} \quad (14)$$

$$\boldsymbol{y'} = \boldsymbol{Cx'} + \boldsymbol{Du'} \quad (15)$$

A avaliação desse problema de controle permite observar com a matriz de ganhos, que o índice de condicionamento do sistema, assume o valor de 19831076. Pela alta magnitude do índice, pode-se concluir que o sistema é de controle extremamente difícil,

mostrando-se um excelente *benchmarking* para testar o algoritmo desenvolvido.

Para situações similares, em geral, o engenheiro de controle reduz a dimensionalidade desse sistema de controle para um de dimensão inferior que seja melhor condicionado.

Aplicação do Controle Inteligente ao Processo da Coluna de Flotação

Para testar o algoritmo de controle, foi desenvolvido um software em Python, encapsulando todo o algoritmo descrito nas seções acima, além de um simulador numérico de sistemas dinâmicos.

O software de simulação supõe cada passo da solução das equações diferenciais como uma iteração do problema formulado na Figura 3, supondo que o sistema em questão seja de característica Markoviana, ou seja, que toda a informação para a tomada de decisão seja encapsulada pela observação fornecida ao agente pela função de passo.

Para garantir que tal hipótese seja satisfeita, foi desenvolvida uma funcionalidade que armazena as M últimas observações base do sistema e as retorna para o agente como uma observação concatenada, fornecendo ao agente informação de memória.

Cada observação base é definida como uma tupla que contempla os valores de todas as variáveis controladas, descritas pelo vetor $\boldsymbol{y}'$, manipuladas, descritas pelo vetor $\boldsymbol{u}'$ e valores de setpoint no tempo da iteração. O valor inicial das variáveis de entrada e do *setpoint* ao longo da simulação escolhidos para o teste de *benchmarking* do algoritmo foram de $(1,1,1)$ e $(0,0,0)$, respectivamente.

RESULTADOS E DISCUSSÕES

Após o desenvolvimento de todo o código do algoritmo neuroevolutivo, do simulador e dos modelos para teste, procedeu-se a realizar uma série de testes para avaliar a eficácia do controle proposto.

Para se avaliar a importância do tamanho da população no desempenho do aprendizado, vários estudos foram feitos e não colocados aqui para concisão dos resultados.

Pode-se perceber pela Figura 4, que a dinâmica do sistema, para um vetor U em base unitária, possui característica bem diversa entre as variáveis de saída. Também é notável a diferença da magnitude dos valores de cada variável, mostrando a necessidade de aplicar uma normalização às equações que descrevem o sistema, visto que redes neuronais, assim como a vasta maioria dos métodos de aproximação funcional, operam melhor nas proximidades da origem cartesiana.

Assim, foi realizada a normalização do modelo, com vetor de normalização composto pelos valores 50, 100 e 0,025, respectivamente, gerando uma dinâmica mais próxima entre suas variáveis controladas, propiciando um aprendizado facilitado pela rede, como exibe a Figura 5.

Após o preparo do ambiente de aprendizado em consonância com o modelo, foi definida uma topologia de duas camadas ocultas de 8 neurônios e entrada e saída da rede com o número de nodos equivalentes a uma memória de duas observações e a entrada de atuação do agente no sistema, respectivamente.

De posse dos parâmetros operacionais adequados, pôde-se realizar o teste de desempenho final para o controle MIMO 3x3 da coluna de flotação, manipulando todas as variáveis do vetor $\boldsymbol{u}'$ de modo a controlar todas as variáveis do vetor $\boldsymbol{y}'$ normalizado. Assim, o controlador final obtido ao longo de 200 gerações para um tamanho populacional de 1000 indivíduos, é verificado nas Figuras 6 e 7.

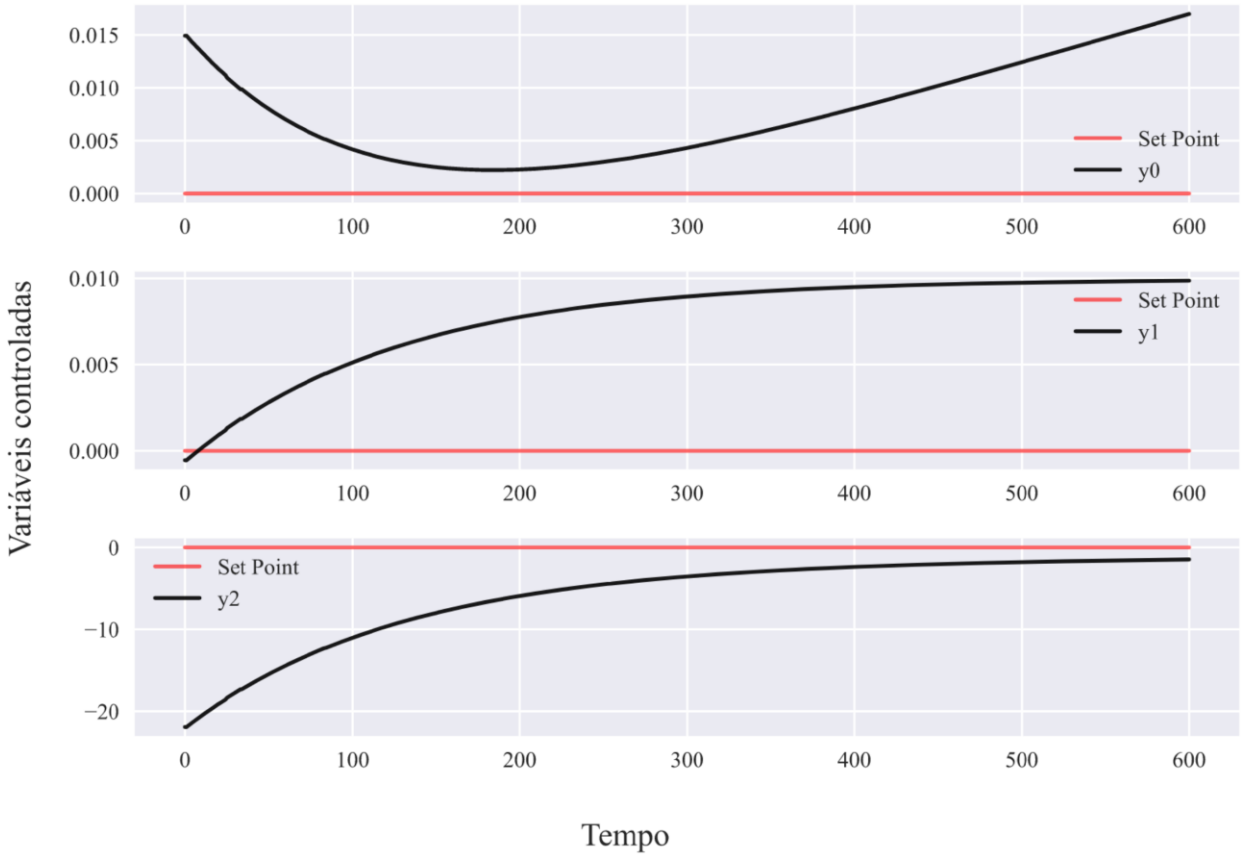

Figura 4 - Comportamento dinâmico das saídas da planta em malha aberta.

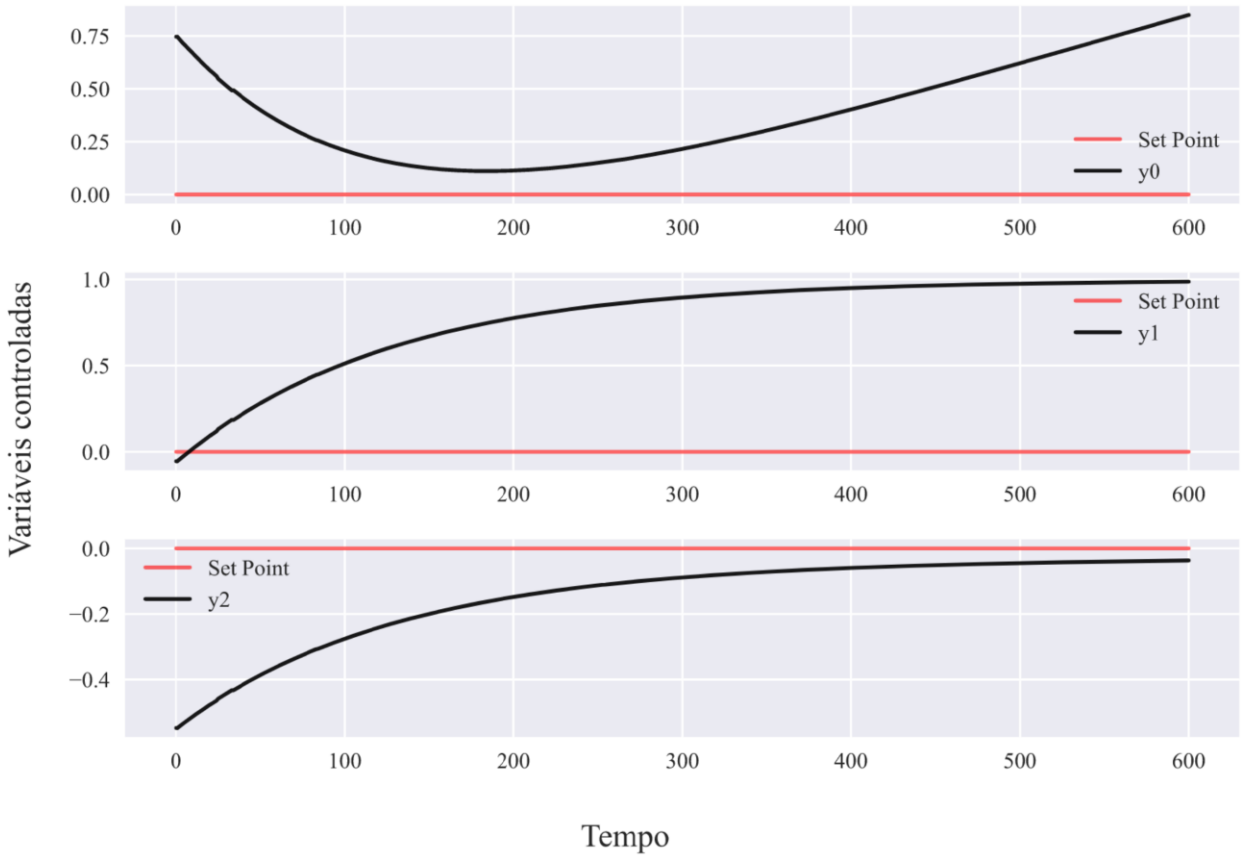

Figura 5: Simulação com valores normalizados para a coluna em malha aberta

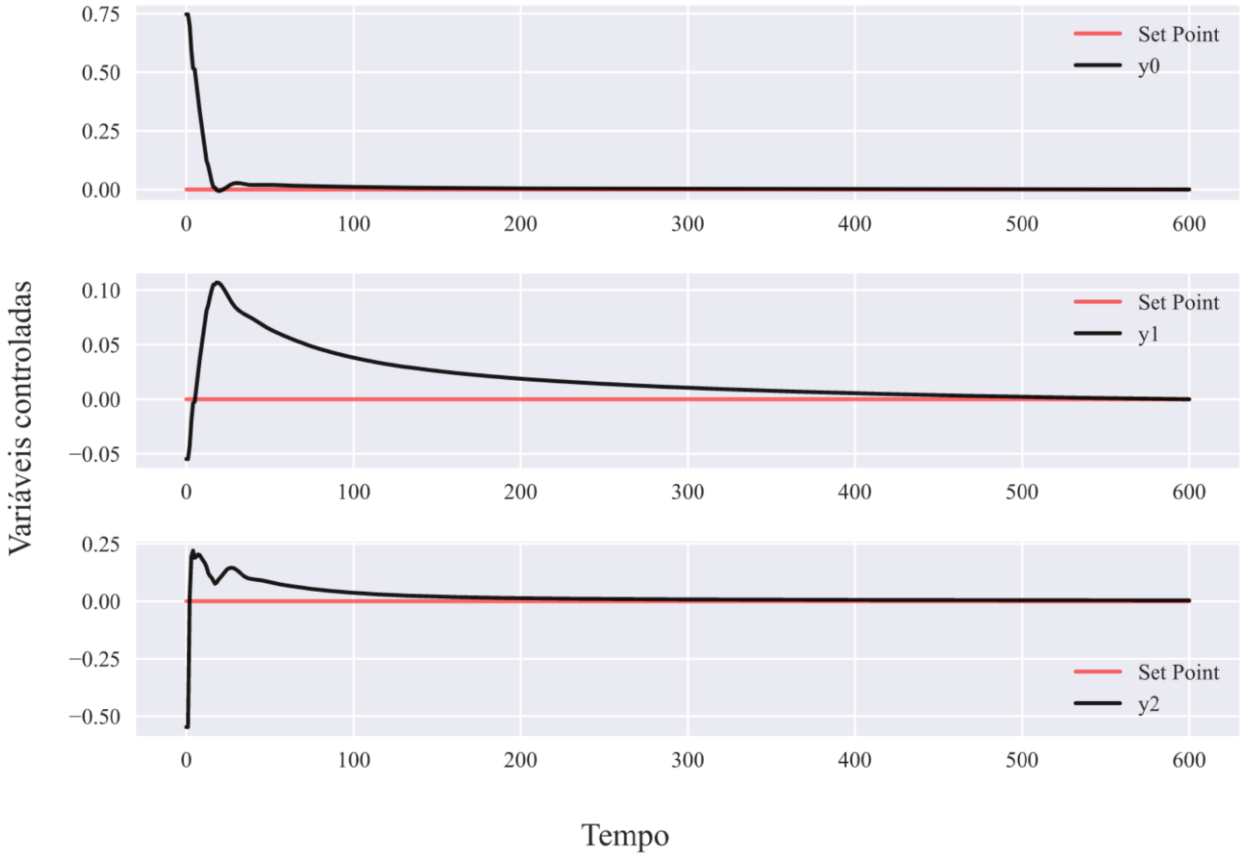

Figura 6: Simulação do sistema controlado resultante da coluna de flotação.

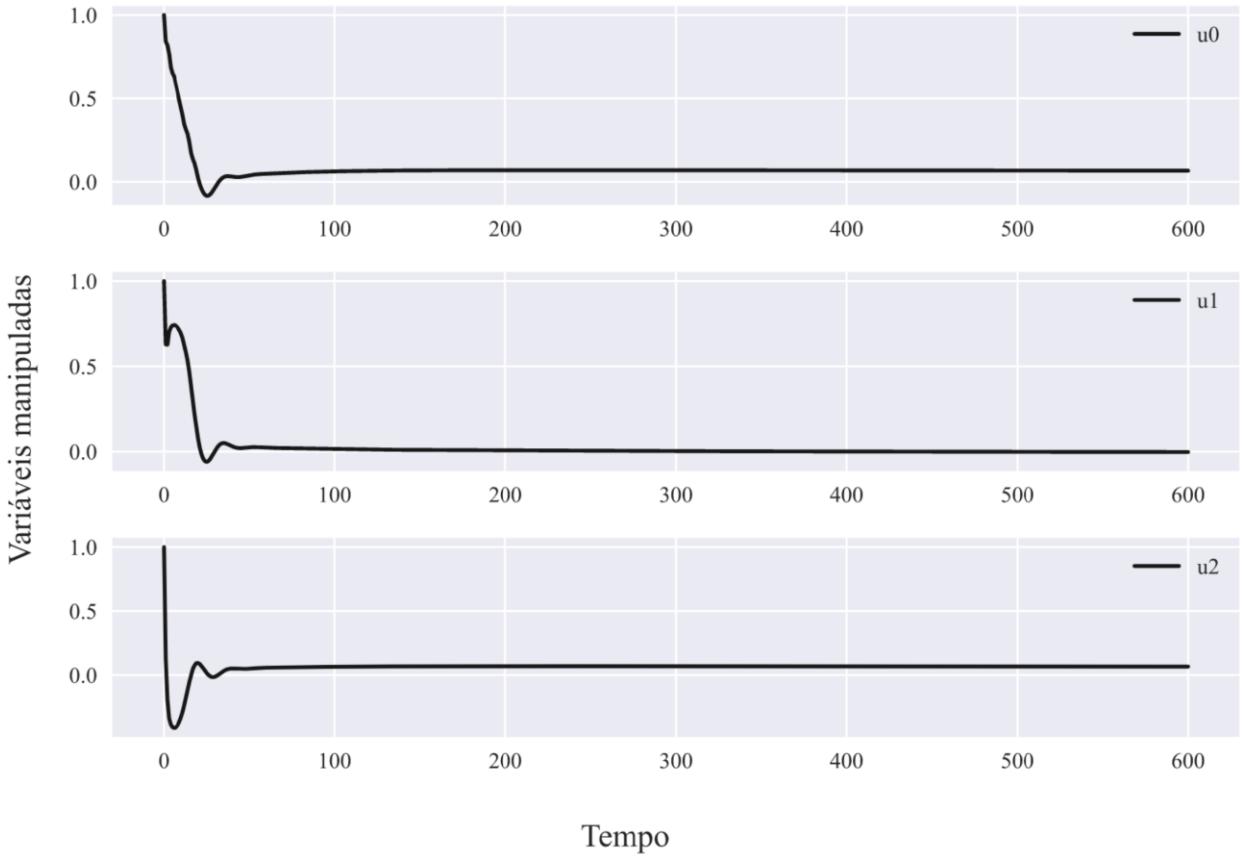

Figura 7: Simulação das entradas manipuladas para a coluna de flotação.

Pode-se perceber que o agente obtido desenvolveu uma lei de controle que manipula as três variáveis selecionadas de modo a minimizar a soma do desvio quadrático das três variáveis de controle em relação a seus respectivos valores de *setpoint*.

Como teste final da robustez do algoritmo proposto, foi realizada uma simulação com perturbações degrau aleatória nos valores das variáveis de sistema para o modelo da coluna, de modo a avaliar a generalização da rede atuante para também o problema regulatório, obtendo assim, as Figuras 8 e 9.

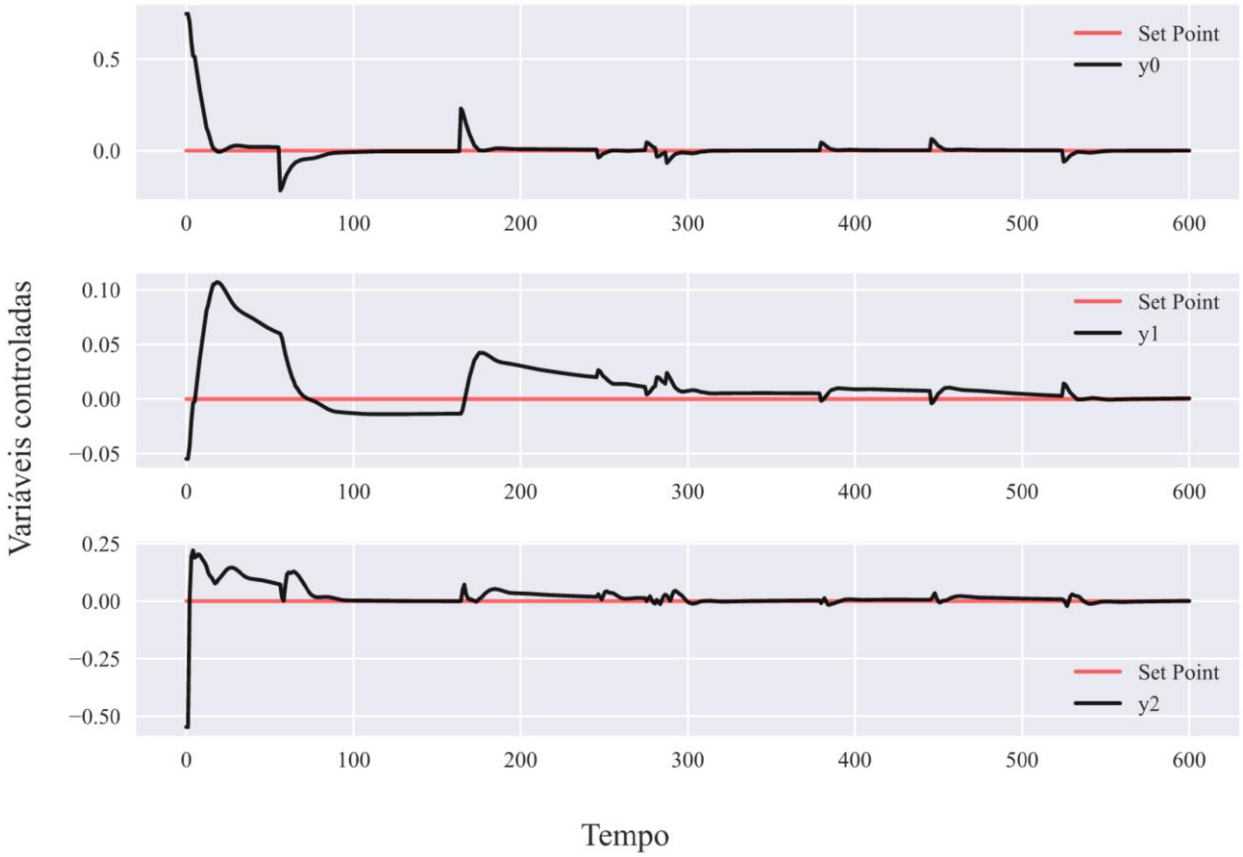

Figura 8: Simulação do sistema controlado com perturbações.

A partir do teste desenvolvido, é notável o desempenho do controlador na rejeição de perturbações, para cada simulação, o desenvolvimento dinâmico do sistema é diferente, dada a natureza aleatória de tais perturbações, ainda assim, o agente obteve sucesso na rejeição destas, levando o sistema para seus valores de *setpoint* com eficiência e eficácia, o que mostra robustez no controle desenvolvido, descartando a possibilidade de a rede neuronal apenas ter decorado todos os estados possíveis do problema, efeito conhecido por *overfitting*.

Assim, com base nos resultados obtidos, pode-se inferir que mesmo para um sistema multivariável de controle desafiador, o algoritmo conseguiu efetuar um controle bastante satisfatório.

CONCLUSÕES E SUGESTÕES

Os resultados obtidos são promissores e motivam investigações adicionais sobre potencial dos métodos neuroevolutivos no projeto de sistemas de controle, de maneira geral, o que pode caracterizar uma alternativa eficaz para o controle de sistemas complexos além de apresentar o aspecto de geração automática do modelo do controlador, isentando o engenheiro, o encargo de projetar o controlador de maneira específica, ou de recair no controle sub-ótimo apresentado pelos controladores lineares Proporcionais, Integrais e Derivativos, nem sempre eficazes.

A modelagem e simulação da coluna de flotação representou um bom teste de

benchmarking para a técnica e viabilizou a sua avaliação de maneira satisfatória.

É proposto para próximos estudos avaliar o comportamento do controlador neuroevolutivo para sistemas não lineares, simulações em robótica e outros processos industriais, em comparação com outros métodos de controle, clássicos ou não, bem como a implantação da rede neuronal resultante em um processo real, para verificar experimentalmente sua capacidade de controle.

É importante observar que um sistema de controle inteligente precisará ter refinamento contínuo para encapsular novos cenários e perturbações não treinadas, ao tempo que amplia os conhecimentos com o aprendizado em linha e monitora ataques adversários, que ocorrem quando um atacante altera a entrada do sistema inteligente para induzir um viés na tomada de decisão.

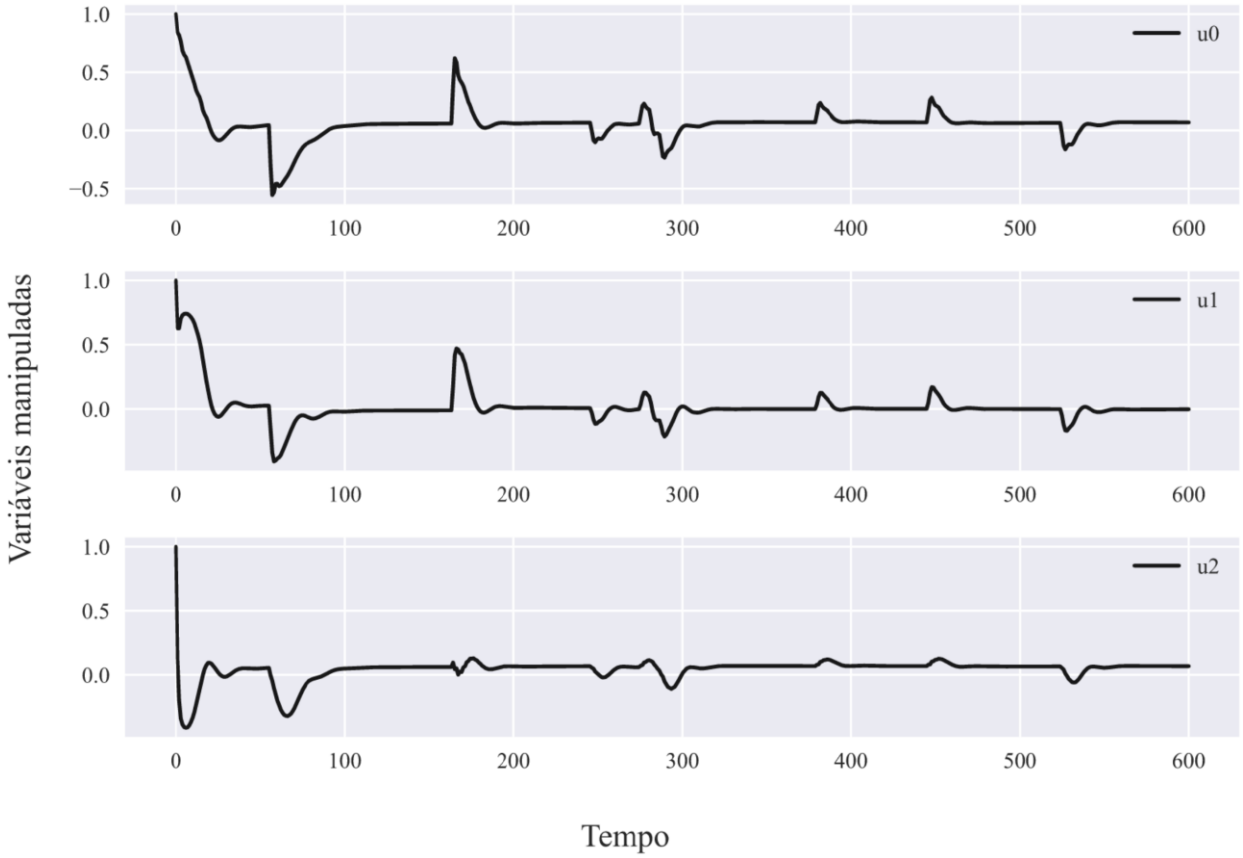

Figura 9: Simulação do sistema manipulado com perturbações.

NOMENCLATURA

$n \rightarrow$ Dimensão do vetor genoma.

$k \rightarrow$ Número de indivíduos selecionados por subgrupo do torneio.

$\beta \rightarrow$ Fator de paternidade definido para o cruzamento.

$p \rightarrow$ Parâmetros modificados pelo cruzamento.

$d' \rightarrow$ Número total de discretizações do espaço de busca do exemplo por Tomassini (1995).

$\boldsymbol{x} \rightarrow$ Variáveis de estado de um sistema dinâmico.

$\boldsymbol{u} \rightarrow$ Variáveis de entrada de um sistema dinâmico.

$\boldsymbol{d} \rightarrow$ Variáveis de perturbação de um sistema dinâmico.

$t \rightarrow$ Tempo.

$G_{11} \rightarrow$ Função de transferência para o sistema da coluna de flotação.

$G_{12} \rightarrow$ Função de transferência para o sistema da coluna de flotação.

$G_{13} \rightarrow$ Função de transferência para o sistema da coluna de flotação.

$G_{21} \rightarrow$ Função de transferência para o sistema da coluna de flotação.

$G_{22} \rightarrow$ Função de transferência para o sistema da coluna de flotação.

G_{23} → Função de transferência para o sistema da coluna de flotação.

G_{31} → Função de transferência para o sistema da coluna de flotação.

G_{32} → Função de transferência para o sistema da coluna de flotação.

G_{33} → Função de transferência para o sistema da coluna de flotação.

K_1 → Constante de ganho do processo com o valor igual a $-4{,}8 \cdot 10^4$.

K_2 → Constante de ganho do processo com o valor igual a $-3{,}84 \cdot 10^5$.

s → Variável dinâmica do processo no domínio de Laplace.

$\boldsymbol{x'}$ → Variáveis de estado de um sistema dinâmico em variável desvio.

$\boldsymbol{u'}$ → Variáveis de entrada de um sistema dinâmico em variável desvio.

$\boldsymbol{y'}$ → Variáveis monitoradas de um sistema dinâmico em variável desvio.

$\boldsymbol{A}$ → Matriz característica de estados de um sistema dinâmico.

$\boldsymbol{B}$ → Matriz característica de entradas de um sistema dinâmico.

$\boldsymbol{C}$ → Matriz característica de estados de um sistema dinâmico para a grandeza monitorada.

$\boldsymbol{D}$ → Matriz característica de entradas de um sistema dinâmico para a grandeza monitorada.

M → Quantidade de últimas observações fornecidas como estado para o agente.

REFERÊNCIAS

AGGARWAL, C. C (2018). Neural Networks and Deep Learning: A Textbook Springer, 497 pág.

DARWIN, C. (1869), On the Origin of Species by Means of Natural Selection, or the Preservation of Favoured Races in the Struggle of Life. London.

DEGRAVE *et al.* (2022), Magnetic Control of Tokamak Plasmas Through Deep Reinforcement Learning, Nature 602, p. 414-419.

LYAPUNOV, A. M. (1892), The General Problem of the Stability of Motion, Univ. Kharkov.

KINSLEY, H., KUKIELA, D. (2020) Neural Networks from Scratch in Python., p. 249-309.

McCULLOCH, W.S.; PITTS, W., (1943). Bull. Math. Biophys, 5, 115.

OMELIANENKO, I. (2019), Hands-On Neuroevolution with Python, Packt Publishing, Birmingham, UK.

PERSECHINI *et al.* (2000), Dinamic Model of a Flotation Column. Minerals Engineering, vol. 13, No. 14-15.

SUTTON, R. S., BARTO, A. G. (2018). Reinforcement Learning: An Introduction. MIT press.

STANLEY, K. O., MIIKKULAINEN, R. (2002). Evolving Neural Networks Through Augmenting Topologies. Evolutionary Computation, vol. 10, n. 2, p. 99-127.

TOMASSINI, M. (1995), A Survey of Genetic Algorithms. Laboratoire de Systèmes Logiques. Ecole Polytechnique Fèderale de Lausanne.