



*Trabalho de Conclusão de Curso*

# Modelagem dinâmica de um pêndulo duplo utilizando Redes Neurais Lagrangianas

Luana Júlia Nunes Ferreira

orientado por

Prof. Dr. Ícaro Bezerra Queiroz de Araújo

Universidade Federal de Alagoas  
Instituto de Computação  
Maceió, Alagoas  
5 de maio de 2023

UNIVERSIDADE FEDERAL DE ALAGOAS  
Instituto de Computação

## MODELAGEM DINÂMICA DE UM PÊNDULO DUPLO UTILIZANDO REDES NEURAIAS LAGRANGIANAS

Trabalho de Conclusão de Curso apresentado  
ao Instituto de Computação da Universidade  
Federal de Alagoas como requisito parcial  
para a obtenção do grau de Bacharel em En-  
genharia de Computação.

Luana Júlia Nunes Ferreira

*Orientador: Prof. Dr. Ícaro Bezerra Queiroz de Araújo*

### **Banca Examinadora:**

Ícaro Bezerra Queiroz de Araújo	Prof. Dr., IC-UFAL
Tiago Alves de Almeida	Prof. Dr., IC-UFAL
Glauber Rodrigues Leite	MsC, IC-UFAL

Maceió, Alagoas  
5 de maio de 2023

**Catálogo na fonte**  
**Universidade Federal de Alagoas**  
**Biblioteca Central**  
**Divisão de Tratamento Técnico**

Bibliotecário: Jorge Raimundo da Silva – CRB-4 –1528

F383m Ferreira, Luana Júlia Nunes.

Modelagem dinâmica de um pêndulo duplo utilizando redes neurais lagrangianas / Luana Júlia Nunes Ferreira. – 2023.  
36 f.

Orientador: Icaro Bezerra Queiroz de Araújo.

Monografia (Trabalho de Conclusão de Curso em Engenharia da Computação) – Universidade Federal de Alagoas. Instituto de Computação. Maceió, 2023.

Bibliografia: f. 35-36.

1. Redes Neurais. 2. Lagrangiana. 3. Sistemas Dinâmicos. 4. Controle. 5. Pêndulo duplo. I. Título.

CDU: 004.7

# Dedicatória

Dedico este trabalho às memórias de minhas bisavós, Emozita Medeiros e Júlia Nunes.

# Agradecimentos

Gostaria de agradecer a todos os professores e técnicos administrativos do Instituto de Computação da UFAL que contribuíram para a realização deste trabalho. Em especial, agradeço ao professor Ícaro Araújo que me orientou durante o processo. Também sou grata aos técnicos administrativos, em especial, Ana Ferreira e Marcelo, que sempre estiveram disponíveis para ajudar e tornar o ambiente acadêmico mais agradável e acolhedor.

Agradeço meus amigos de turma por todo o companheirismo e diversão ao longo de 5 anos de curso. Para demonstrar o tamanho da minha gratidão, faço questão de citá-los nominalmente: João Pedro, Hiago, Jhonnye, Tiago Cabral, John Davi, Bruna, Mateus, Ruan e Hugo. Obrigada por nunca me deixarem desistir e por me fazerem ter as melhores risadas.

Agradeço aos membros do projeto KATIE, pois a nossa união tornou a graduação mais leve para muitas meninas, que assim como eu, eram as únicas de suas turmas.

Agradeço também meu amigo Miquéias, do Instituto de Física da UFAL, por todo o apoio nos pontos críticos da vida.

Agradeço meu namorado, Lucas Massa, por ser meu parceiro e melhor amigo durante a graduação, surtar junto comigo e estar sempre ao meu lado mesmo quando eu passo 3 dias sem dormir.

Agradeço meus tios, Marcielle e Jarbas, meu (pai)drasto, Luiz Gustavo, e minha avó, Márcia, por terem sido fundamentais na minha educação básica. Obrigada por tudo.

Agradeço ao meu irmão, Lucas Ferreira, pelas caronas até a UFAL.

Agradeço meu primeiro orientador, professor Dr. Carlos Argolo, do Instituto Federal de Alagoas, por todos os ensinamentos que carrego para a vida, por me mostrar o caminho da ciência e da docência.

Agradeço imensamente à minha mãe, Júlia Nunes, não só por todo o investimento de recursos em minha educação e pelo apoio incondicional nos momentos mais difíceis, mas principalmente pelo final de semana em 2009 no qual ela se dedicou a me ensinar operações com números inteiros.

Maceió, 5 de maio de 2023.

# Resumo

O presente trabalho tem como objetivo estudar a aplicação de redes neurais na modelagem de um pêndulo duplo, um sistema dinâmico complexo e desafiador. Uma nova classe de Redes Neurais Hamiltonianas (HNN) para modelar o pêndulo duplo foi proposta, porém apresenta falha em muitos sistemas físicos que não podem ser expressos como um sistema Hamiltoniano e, por isso, foi proposto o uso de redes neurais com formulação Lagrangiana (LNN). Logo, este trabalho busca avaliar a eficácia dessa abordagem, reproduzindo o trabalho de Cranmer et al. (2020). Os resultados dessa pesquisa confirmam que a abordagem LNN foi eficaz na obtenção de soluções numéricas próximas à solução analítica. Além disso, o sistema também se mostrou sensível às condições iniciais e a análise da função de *loss* indicou estabilidade nos resultados após o período de treinamento. A comparação entre as velocidades e acelerações angulares provenientes da solução analítica e previstas mostrou uma relação linear e mínima interferência de outliers. O mapa de calor gerado a partir da análise do espaço de coordenadas mostrou que o modelo se comportou de forma satisfatória.

***Palavras-chave:*** Redes Neurais, Lagrangiana, Sistemas Dinâmicos, Controle, Pêndulo duplo.

# Abstract

This work aims to study the application of neural networks in the modeling of the double pendulum, a complex and challenging dynamic system. A new class of Hamiltonian Neural Networks (HNN) was proposed to model the double pendulum, but it fails in many physical systems that cannot be expressed as a Hamiltonian system. Therefore, the use of neural networks with Lagrangian formulation (LNN) was proposed. This work seeks to evaluate the effectiveness of this approach, using the double pendulum as a case study and reproducing the work of Cranmer et al. (2020). The results of this research confirm that the LNN approach was effective in obtaining numerical solutions close to the ideal analytical solution. Furthermore, the system also proved to be sensitive to initial conditions, and the analysis of the loss function indicated stability in the results after the training period. The comparison between real and predicted angular velocities and accelerations showed a linear relationship and minimal interference from outliers. The heat map generated from the analysis of the coordinate space showed that the model behaved satisfactorily

*Keywords: Neural Networks, Lagrangian, Dynamical Systems, Control, Double Pendulum.*

# Lista de Figuras

2.1	Representação gráfica de uma rede neural do tipo <i>feedforward</i> . Modificado de Hemeida et al. (2020) . . . . .	10
3.1	Representação gráfica do Pêndulo Duplo no sistema de coordenadas cartesiano. Fonte: autora, 2023. . . . .	12
3.2	Diagrama de blocos para o fluxo do pré-processamento. . . . .	15
4.1	Comparação entre as soluções. (Unidades de medida - estado: graus, tempo: segundos) (a) Comparação entre a solução analítica sem perturbação e com uma perturbação $\epsilon = 3 \cdot 10^{-11}$ ; (b) Comparação entre a solução analítica sem perturbação e com uma perturbação $\epsilon = 2 \cdot 10^{-12}$ ; (c) Comparação entre a solução analítica e o autogradiente calculado pela ferramenta JAX. . . . .	16
4.2	Comparação da distribuição dos dados entre conjuntos de treinamento e teste. . . . .	17
4.3	Loss calculado através do MSE ao longo do treinamento . . . . .	18
4.4	(a) Velocidade angular real vs. prevista; (a) aceleração angular real vs. prevista; . . . . .	18
4.5	Desempenho dos modelos reais e previstos no espaço de coordenadas no quesito velocidade angular. . . . .	19
4.6	Desempenho dos modelos reais e previstos no espaço de coordenadas no quesito aceleração angular. . . . .	20
4.7	Comparação entre (a) solução analítica vs. perturbação; (b) solução analítica vs. LNN. . . . .	20



# Lista de Símbolos

- $O$  Origem do sistema de coordenadas cartesiano
- $F_i^{\vec{a}}$  Força externa aplicada sobre o corpo  $i$
- $\vec{F}_i$  Força conservativa aplicada sobre o corpo  $i$
- $\dot{p}_i$  Força efetiva invertida
- $\delta_{\vec{r}_i}$  Variação infinitesimal arbitrária das coordenadas generalizadas
- $\vec{r}_i$  Vetor de Coordenadas Generalizadas da partícula  $i$
- $T$  Energia Cinética da Lagrangiana
- $V$  Energia Potencial da Lagrangiana
- $q_i$  Componente da Coordenada Generalizada da partícula  $i$
- $\dot{q}_i$  Velocidade angular da partícula
- $Q_i$  Componentes da força generalizada
- $\delta_{q_i}$  Deslocamentos virtuais independentes da partícula  $i$
- $L$  Lagrangiana
- $t$  Variável temporal
- $\eta(t)$  Parâmetro que mede a diferença entre duas curvas
- $S$  Valor escalar atribuído a cada um dos caminhos possíveis entre dois pontos
- $\alpha$  Parâmetro escalar de correção
- $\theta_1$  Ângulo formado entre a partícula 1 e a componente vertical
- $\theta_2$  Ângulo formado entre a partícula 2 e a componente vertical
- $\dot{\theta}_1$  Velocidade angular da partícula 1

$\dot{\theta}_2$	Velocidade angular da partícula 2
$g$	Aceleração Gravitacional da Terra
$x_1$	Abscissa da partícula 1
$x_2$	Abscissa da partícula 2
$y_1$	Ordenada da partícula 1
$y_2$	Ordenada da partícula 2
$m_1$	Massa da partícula 1
$m_2$	Massa da partícula 2
$l_1$	Comprimento da haste rígida da partícula 1
$l_2$	Comprimento da haste rígida da partícula 2
$\ddot{q}$	Aceleração angular da partícula
$v_1$	Velocidade da partícula 1
$v_2$	Velocidade da partícula 2
$\dot{x}_1$	Primeira derivada da abscissa da partícula 1
$\dot{x}_2$	Primeira derivada da abscissa da partícula 2
$\dot{y}_1$	Primeira derivada da ordenada da partícula 1
$\dot{y}_2$	Primeira derivada da ordenada da partícula 2
$\nabla$	Gradiente de uma função
$\nabla^T$	Gradiente transposto de uma função
$\epsilon$	Perturbação inserida no sistema
$e$	Número de Euler

# Lista de Abreviações

**HNN** *Hamiltonian Neural Network*

**ICRA** *International Conference on Robotics and Automation*

**JAX** *Just Another Transformation Library*

**LNN** *Lagrangian Neural Networks*

**MSE** *Mean Squared Error*

**NeurIPS** Conferência de Redes Neurais Avançadas

**ReLU** *Rectified Linear Unit*

# Sumário

<b>1</b>	<b>Introdução</b>	<b>1</b>
1.1	Objetivos . . . . .	3
1.1.1	Objetivos Gerais . . . . .	3
1.1.2	Objetivos Específicos . . . . .	3
1.2	Organização do trabalho . . . . .	3
<b>2</b>	<b>Fundamentação Teórica</b>	<b>5</b>
2.1	Mecânica Clássica . . . . .	5
2.1.1	Função Lagrangiana . . . . .	5
2.1.2	Princípio da Mínima Ação . . . . .	7
2.1.3	Equação de Euler-Lagrange . . . . .	8
2.2	Redes Neurais e Aprendizado Profundo . . . . .	10
<b>3</b>	<b>Metodologia</b>	<b>12</b>
3.1	Modelagem Lagrangiana do Pêndulo Duplo . . . . .	12
3.2	Aprendizado Profundo Lagrangiano . . . . .	14
3.2.1	Implementação da LNN . . . . .	15
<b>4</b>	<b>Resultados e Discussões</b>	<b>16</b>
<b>5</b>	<b>Conclusão</b>	<b>22</b>
5.1	Trabalhos Futuros . . . . .	23
	<b>Bibliografia</b>	<b>24</b>

# Capítulo 1

## Introdução

O pêndulo duplo é um sistema dinâmico complexo e desafiador de se modelar devido à sua natureza não linear (Taylor (2005)). O movimento desse sistema é influenciado não apenas pela posição inicial dos pêndulos, mas também por suas velocidades e acelerações. Além disso, as forças que agem no sistema, como a força da gravidade e a tensão nas hastes, estão continuamente se alterando em resposta ao movimento dos pêndulos. Como resultado, pequenas variações nas condições iniciais podem levar a grandes diferenças no comportamento do sistema ao longo do tempo (Greiner and Bromley (2003)). Por essa razão, o pêndulo duplo é frequentemente utilizado como um desafio de engenharia e controle no contexto dos sistemas dinâmicos.

Nesse cenário, redes neurais têm sido amplamente utilizadas para a modelagem e o controle de sistemas dinâmicos complexos, possuindo diversas aplicações nas áreas de Robótica (Li et al. (2020), Liu et al. (2020)) e Sistemas de Controle (Antsaklis et al. (1990), Selmic and Lewis (2000)).

O estudo de Lee and Carlberg (2020) propõe uma estratégia de redução de modelo para sistemas dinâmicos em variedades não lineares baseado em *Autoencoders* Convolucionais Profundos. O método apresentado utiliza redes neurais profundas para aprender a representação latente das trajetórias do sistema em uma variedade não linear, reduzindo assim a dimensionalidade do sistema original. A rede neural é treinada para reconstruir as trajetórias do sistema na variedade de baixa dimensionalidade, permitindo a modelagem do comportamento dinâmico do sistema em um espaço de menor dimensão. Os resultados obtidos mostraram que o modelo é capaz de reduzir significativamente o número de variáveis necessárias para modelar o comportamento dinâmico de sistemas não lineares complexos como o pêndulo duplo.

Em 2020, uma técnica de controle antibalço para um guindaste de pêndulo duplo usando uma rede neural pura foi proposto por Chen et al. (2021) na 22ª edição do *Asian Journal of Control*. A abordagem proposta baseia-se na Teoria de Estabilidade de Lyapunov e utiliza uma rede neural com duas camadas para aproximar a função de controle. A rede neural é treinada usando o algoritmo *backpropagation* para otimizar a função de

custo, que é baseada no erro de posição e velocidade. Os resultados experimentais mostraram que essa abordagem é capaz de controlar efetivamente o movimento do guindaste de pêndulo duplo, reduzindo significativamente a oscilação do guindaste em relação à posição desejada.

Na pesquisa de Zhang and Jing (2021), os autores apresentam um método de controle adaptativo baseado em redes neurais para sistemas de guindaste de torre de pêndulo duplo com entradas não ideais. A metodologia desenvolvida utiliza *Deep Learning* para modelar o comportamento dinâmico do sistema e adaptar os parâmetros do controlador de acordo com as mudanças nas condições operacionais. Os resultados práticos mostraram que essa metodologia é eficaz para rastrear trajetórias desejadas em condições adversas, como ventos fortes e oscilações não lineares, melhorando a segurança e a eficiência operacional.

A 33<sup>a</sup> Conferência de Redes Neurais Avançadas (NeurIPS 2019) trouxe uma nova classe de redes neurais que utilizam a formulação hamiltoniana para modelar o pêndulo duplo com o trabalho de Greydanus et al. (2019). A formulação hamiltoniana é uma técnica matemática que descreve o comportamento de sistemas físicos em termos de suas propriedades de energia e movimento. As Redes Neurais Hamiltonianas (HNN) propostas neste artigo são capazes de aprender as dinâmicas dos sistemas físicos sem a necessidade de um modelo explícito, permitindo a modelagem de sistemas complexos e não lineares. Os resultados empíricos mostraram que as HNNs são capazes de modelar com precisão sistemas físicos complexos, como pêndulos duplos, e que possuem propriedades importantes, como conservação de energia e invariância temporal.

Entretanto, de acordo com Cranmer et al. (2020) a formulação do pêndulo duplo com HNNs falha porque muitos sistemas físicos não possuem uma formulação Hamiltoniana simples ou não podem ser expressos como um sistema Hamiltoniano, pois requerem coordenadas canônicas.

Sendo assim, o artigo “*Lagrangian Neural Networks*” foi publicado em 2019 durante a *International Conference on Robotics and Automation (ICRA)*, ocasião em que os autores propõem a formulação Lagrangiana, a qual utiliza uma rede neural para aproximar a função Lagrangiana do sistema e, em seguida, derivar as equações de movimento usando a equação de Euler-Lagrange. Os autores demonstram que a modelagem baseada em Redes Neurais Lagrangianas (LNN) é mais flexível e pode ser aplicada a uma ampla gama de sistemas físicos, incluindo sistemas com atrito e sistemas não conservativos. Além disso, o artigo discute como a abordagem pode ser usada para problemas de controle, como a trajetória ótima de um robô, e como ela pode ser estendida para lidar com sistemas de várias partes. Os autores demonstraram que a modelagem Lagrangiana baseada em redes neurais supera os métodos tradicionais de modelagem em termos de precisão e flexibilidade.

Portanto, este Trabalho de Conclusão de Curso busca estudar a metodologia descrita por Cranmer et al. (2020) com o objetivo de modelar o sistema físico do pêndulo duplo

utilizando as Equações de Lagrange e a LNN, avaliando a capacidade da rede em generalizar para novos dados e comparando a precisão do modelo com a solução numérica. Além disso, este trabalho investiga a utilização das LNNs como uma alternativa para a solução numérica de sistemas físicos complexos que demandam conservação de energia.

## 1.1 Objetivos

### 1.1.1 Objetivos Gerais

1. Realizar a modelagem do sistema físico pêndulo duplo utilizando as Equações de Lagrange e a LNN;
2. Analisar a capacidade da rede neural lagrangiana em generalizar para novos dados e avaliar a precisão do modelo por meio da comparação com a solução analítica;
3. Investigar a utilização de LNNs como uma alternativa para a solução numérica de sistemas físicos complexos que demandam conservação de energia.

### 1.1.2 Objetivos Específicos

1. Analisar as pequenas perturbações no sistema e verificar se elas causam interferência significativa em seu comportamento;
2. Implementar a LNN para a modelagem do sistema físico pêndulo duplo e analisar os resultados obtidos a partir da solução analítica;
3. Dividir as amostras em conjuntos de treino e teste e avaliar a capacidade do modelo em aprender a reconhecer padrões de forma geral e generalizar para novas amostras;
4. Analisar a precisão e a estabilidade do modelo implementado a partir dos erros do sistema.

## 1.2 Organização do trabalho

Este trabalho foi organizado em capítulos que detalham os passos seguidos durante a modelagem do pêndulo duplo com LNN aprofundando-se em conceitos teóricos físicos e definições matemáticas que servem como base para este modelo.

O Capítulo 2 aprofunda-se em conceitos da mecânica clássica e definições matemáticas que servem como base para este modelo, tais quais a Função Lagrangiana, o Princípio da Mínima Ação e a Equação de Euler-Lagrange. Além disso, este capítulo traz um resumo sobre o universo das Redes Neurais e do Aprendizado Profundo.

O Capítulo 3 detalha a modelagem do pêndulo duplo utilizando a mecânica clássica e fornece detalhes da implementação da LNN utilizada.

O Capítulo 4 traz a análise e a discussão dos principais resultados da simulação, comparando sempre a solução analítica com a solução da LNN, analisando o pré-processamento dos dados, o processo de treinamento e a avaliação da precisão do modelo.



# Capítulo 2

## Fundamentação Teórica

### 2.1 Mecânica Clássica

#### 2.1.1 Função Lagrangiana

A forma Lagrangiana consiste numa formulação da mecânica clássica que difere desta, sobretudo, pelo formalismo matemático empregado. Em linhas gerais, a notação newtoniana usa um formalismo vetorial, enquanto que a lagrangiana faz uso de grandezas escalares e sua equação fundamental possui a mesma forma em todos os sistemas de coordenadas (Lemos (2007)).

#### O Princípio D'Alembert

Jean Le Rond d'Alembert (1717-1783) foi um matemático, físico e filósofo francês que se destacou por suas contribuições para a mecânica clássica, em particular na teoria das equações diferenciais parciais. Ele desenvolveu a teoria da dinâmica de sistemas de partículas, que permitiu a resolução de problemas de movimento em sistemas complexos. Também foi um dos primeiros a aplicar a análise matemática à física, e desenvolveu o Princípio D'Alembert, que afirma que a força resultante em um sistema em equilíbrio é nula (Nascimento (2020)).

Este princípio afirma que o movimento de um sistema mecânico é governado pela equação de movimento, a 2ª Lei de Newton, acrescida de um termo conhecido como força fictícia ou inercial. Essa força é uma função do deslocamento, da velocidade e da aceleração do sistema, e possui a propriedade de anular-se na solução final da equação de movimento. Daí, infere-se que:

$$\sum \left( F_i^{(a)} - \dot{p}_i \right) \cdot \delta \vec{r}_i = 0 \quad (2.1)$$

Onde  $F_i^{(a)}$  é a força externa aplicada sobre o corpo  $i$ ,  $\delta \vec{r}_i$  é a mudança infinitesimal arbitrária das coordenadas  $\vec{r}_i$  da partícula  $i$ ,  $\dot{p}_i$  é uma força efetiva invertida introduzida

para anular a equação.

## Deslocamento Virtual

Um deslocamento virtual é uma variação infinitesimal da trajetória de um sistema físico que não necessariamente representa um movimento real do sistema, mas que é usado para analisar o comportamento do sistema sob perturbações. No contexto lagrangiano, o deslocamento virtual é uma mudança hipotética na posição de um sistema físico, que não necessariamente ocorre na prática, mas é usada para analisar como as forças atuantes no sistema são afetadas por essa perturbação. Essa mudança é usualmente representada por uma pequena variação nos parâmetros do sistema, como posições, velocidades, acelerações ou ângulos.

## Demonstração da Lagrangiana

Com isso, é possível transformar o Princípio de D'Alembert em uma expressão que envolva deslocamentos virtuais das coordenadas generalizadas, de modo que sejam independentes entre si e que os coeficientes infinitesimais possam ser iguais a zero separadamente. Sendo assim, é feita uma translação do sistema de coordenadas  $\vec{r}_i$  para o sistema  $q_i$ , tal que:

$$\vec{r}_i = \vec{r}_i(q_1, \dots, q_n, t) \quad (2.2)$$

no qual a variável temporal  $t$  é introduzida como parte do sistema de coordenadas. Além disso, é fundamental para a abordagem lagrangiana que seja incorporado o conceito de energia cinética, representado pela variável  $T$ . Assim, obtém-se:

$$\sum \left\{ \left[ \frac{d}{dt} \left( \frac{\partial T}{\partial \dot{q}_i} \right) - \frac{\partial T}{\partial q_i} \right] - Q_i \right\} \delta_{q_i} = 0 \quad (2.3)$$

A equação 2.3 não possui dependência de um sistema de coordenadas cartesiano nos termos da energia cinética, simplificando a equação. Nesse cenário, as variáveis  $q_i$  são independentes do sistema de coordenadas,  $Q_i$  são as componentes da força generalizada que atua no sistema e existem deslocamentos virtuais  $\delta_{q_i}$  independentes tais que os coeficientes individuais anulam-se. Daí:

$$\frac{d}{dt} \left( \frac{\partial T}{\partial \dot{q}_i} \right) - \frac{\partial T}{\partial q_i} = Q_i \quad (2.4)$$

Em sequência, como a força sobre uma partícula depende apenas da posição e é independente do caminho percorrido pela partícula, então ela é conservativa e pode ser expressa como o gradiente da função potencial  $V$ :

$$\vec{F}_i = -\nabla_i V \Rightarrow Q_i = \sum_i \vec{F}_i \cdot \frac{\partial \vec{r}_i}{\partial q_i} = -\sum_i \nabla_i V \cdot \frac{\partial \vec{r}_i}{\partial q_i} \quad (2.5)$$

Observa-se que a equação 2.5 é equivalente a expressão obtida pela derivada parcial de uma função  $-V(\vec{r}_1, \dots, \vec{r}_n, t)$  em relação a  $q_i$ :

$$Q_i = -\frac{\partial V}{\partial q_i} \quad (2.6)$$

Assim, obtém-se:

$$\frac{d}{dt} \left( \frac{\partial T}{\partial \dot{q}_i} \right) - \frac{\partial(T - V)}{\partial q_i} = 0 \quad (2.7)$$

Esse sistema será conservativo somente quando a função potencial for independente do tempo. Como a função potencial independe das velocidades generalizadas, inclui-se um termo referente a derivada parcial em relação a  $\dot{q}_i$ , tal que:

$$\frac{d}{dt} \left( \frac{\partial(T - V)}{\partial \dot{q}_i} \right) - \frac{\partial(T - V)}{\partial q_i} = 0 \quad (2.8)$$

Da equação 2.8, obtém-se a definição de Lagrangiana  $L$  expressa por:

$$L = T - V \quad (2.9)$$

Substituindo a equação 2.9 na equação 2.8, obtém-se a expressão:

$$\frac{d}{dt} \left( \frac{\partial L}{\partial \dot{q}_i} \right) - \frac{\partial L}{\partial q_i} = 0 \quad (2.10)$$

A equação 2.10 é conhecida como Equação de Lagrange.

Outra notação Lagrangiana amplamente utilizada é representada na equação 2.11. Seja  $L(q, \dot{q}, t)$  a Lagrangiana de um sistema e  $F(q, t)$  uma função derivável qualquer das coordenadas generalizadas desse sistema, tal qual uma força atuante no sistema. Sabendo que as variáveis de uma Lagrangiana são as coordenadas generalizadas do sistema  $q_i$  e suas velocidades generalizadas  $\dot{q}_i$ , pode-se dizer que:

$$L'(q, \dot{q}, t) = L(q, \dot{q}, t) + \frac{dF}{dt} \quad (2.11)$$

## 2.1.2 Princípio da Mínima Ação

Até aqui, foi discutida a abordagem diferencial e independente do sistema de coordenadas por meio da Equação de Lagrange na descrição de um sistema. Nesta seção discute-se uma segunda abordagem, via princípio integral.

Também conhecido como Princípio de Hamilton, o Princípio da Mínima Ação afirma que o comportamento de um sistema físico entre dois pontos no tempo é tal que a ação

total é mínima (Goldstein et al. (2002)). A ação é definida como a integral ao longo do tempo do Lagrangiano do sistema, tal que:

$$\delta \int_{t_1}^{t_2} L dt = \delta \int_{t_1}^{t_2} (T - V) dt = 0 \quad (2.12)$$

Em outras palavras, diz-se que dentre as diversas variações de caminhos que um sistema dinâmico pode trilhar, a natureza opta, espontaneamente, pelo caminho que torna menor as diferenças entre as energias cinética e potencial. Esse caminho pode ser obtido a partir da equação 2.12 (Greiner and Bromley (2003)).

### 2.1.3 Equação de Euler-Lagrange

A evolução do sistema entre dois pontos no tempo é determinada pelas equações de Euler-Lagrange, as quais são obtidas a partir do Princípio da Mínima Ação (2.1.2). Seja  $S$  um valor escalar atribuído a cada um dos caminhos possíveis entre dois pontos, obtém-se que:

$$S = \int_{t_1}^{t_2} L[q(t), \dot{q}(t), t] dt \quad (2.13)$$

onde  $q(t_1) = q_1$  e  $q(t_2) = q_2$ .

De acordo com Taylor (2005), essas equações descrevem as trajetórias do sistema que extremizam a ação total, ou seja, o objetivo é determinar todas as curvas possíveis que tornem o valor de  $S$  mínimo, máximo ou estacionário, isto é, constante ao longo do tempo.

Sejam  $P_1$  e  $P_2$  dois pontos sobre a curva obtida na minimização de  $S$ . Como  $L = L(q(t), \dot{q}(t), t)$  e  $q = q(t)$ , é fácil ver que o integrando é uma função apenas de  $t$ , pois  $q$  só depende de  $t$ . Além disso, seja:

- $q = q(t)$  a curva que representa a solução correta;
- $Q = Q(t)$  a curva que representa a solução errada;

Podemos definir a relação entre essas curvas por um parâmetro  $\eta$  que mede a diferença entre essas curvas, de modo que:

$$Q(t) = q(t) + \eta(t) \quad (2.14)$$

Como  $Q(t)$  contém os pontos  $P_1$  e  $P_2$ , então:

$$\eta(q_1) = \eta(q_2) = 0 \quad (2.15)$$

A integral  $S$  obtida de  $Q(t)$  tem que ser maior do que o valor obtido da integração sobre a curva  $q(t)$ . Assim, é introduzido um parâmetro  $\alpha$  com o fito de efetuar essa correção. Daí:

$$Q(t) = q(t) + \alpha \cdot \eta(t) \quad (2.16)$$

Agora a integral  $S$  obtida de  $Q(t)$  depende de  $\alpha$ . Logo,  $S = S(\alpha)$ . Ademais, a curva correta  $q(t)$  é obtida quando  $\alpha = 0$ . Como  $S$  é mínimo para  $q(t)$ , então  $S(\alpha)$  é mínimo quando  $\alpha = 0$ , portanto:

$$\alpha = 0 \Rightarrow \frac{dS}{d\alpha} = 0. \quad (2.17)$$

Daí, obtém-se:

$$S(\alpha) = \int_{q_1}^{q_2} L(q, \dot{q}, t) dt = \int_{q_1}^{q_2} L(q + \alpha\eta, \dot{q} + \alpha\dot{\eta}, t) dt \quad (2.18)$$

Como  $\alpha$  aparece no integrando  $L$ , é preciso avaliar  $\frac{\partial L}{\partial \alpha}$ . Aplicando a regra da cadeia, obtém-se:

$$\frac{\partial L(q + \alpha\eta, \dot{q} + \alpha\dot{\eta}, t)}{\partial \alpha} = \eta \frac{\partial L}{\partial q} + \dot{\eta} \frac{\partial L}{\partial \dot{q}} \quad (2.19)$$

De acordo com a condição estabelecida em 2.17, sabe-se que:

$$\frac{dS}{d\alpha} = \int_{q_1}^{q_2} \frac{\partial L}{\partial \alpha} dt = \int_{q_1}^{q_2} \left( \eta \frac{\partial L}{\partial q} + \dot{\eta} \frac{\partial L}{\partial \dot{q}} \right) dt = 0 \quad (2.20)$$

Esta condição deve satisfazer a equação 2.15  $\forall n(t)$ . Dessa maneira, integrando por partes o segundo termo do lado direito da equação 2.20, obtém-se:

$$\int_{q_1}^{q_2} \dot{\eta}(t) \frac{\partial L}{\partial \dot{q}} dt = \underbrace{\left[ \eta(t) \frac{\partial L}{\partial \dot{q}} \right]_{q_1}^{q_2}}_0 - \int_{q_1}^{q_2} \eta(t) \frac{d}{dt} \left( \frac{\partial L}{\partial \dot{q}} \right) dt \quad (2.21)$$

Nesse ponto, pode-se constatar que o primeiro termo do lado direito da equação 2.21 é igual a zero devido a condição 2.15, pois deve ser válida para qualquer função  $\eta(t)$ . Sendo assim:

$$\int_{q_1}^{q_2} \dot{\eta}(t) \frac{\partial L}{\partial \dot{q}} dt = - \int_{q_1}^{q_2} \eta(t) \frac{d}{dt} \left( \frac{\partial L}{\partial \dot{q}} \right) dt \quad (2.22)$$

Substituindo a equação 2.22 em 2.20, obtém-se:

$$\int_{q_1}^{q_2} \eta(t) \underbrace{\left( \frac{\partial L}{\partial q} - \frac{d}{dt} \frac{\partial L}{\partial \dot{q}} \right)}_0 dt = 0 \quad (2.23)$$

Diante da condição estipulada em 2.15, mais uma vez, o termo entre parênteses na equação 2.23, deve ser anulado. Por fim, obtém-se:

$$\frac{\partial L}{\partial q} - \frac{d}{dt} \frac{\partial L}{\partial \dot{q}} = 0 \quad (2.24)$$

A equação 2.24 é conhecida como Equação de Euler-Lagrange.

## 2.2 Redes Neurais e Aprendizado Profundo

As redes neurais são sistemas projetados para imitar o cérebro humano (Xiang et al. (2018)). Em termos matemáticos, uma rede neural é uma coleção de unidades de processamento interconectadas, capazes de realizar cálculos em paralelo e aprender a partir de exemplos. O aprendizado profundo, por sua vez, é uma técnica de treinamento de redes neurais profundas, com muitas camadas, que permite a obtenção de modelos altamente precisos em diversas tarefas.

Essas unidades de processamento interconectadas formam as camadas da rede. Cada camada possui nós que conectam-se aos nós da camada anterior e aos nós da próxima camada. Cada nó em uma rede neural é capaz de efetuar cálculos simples. Quando os nós são conectados em uma rede, o sistema é capaz de calcular modelos lineares e não lineares de maneira distribuída e oportuna. A figura 2.1 traz uma representação de uma rede neural do tipo *feedforward*:

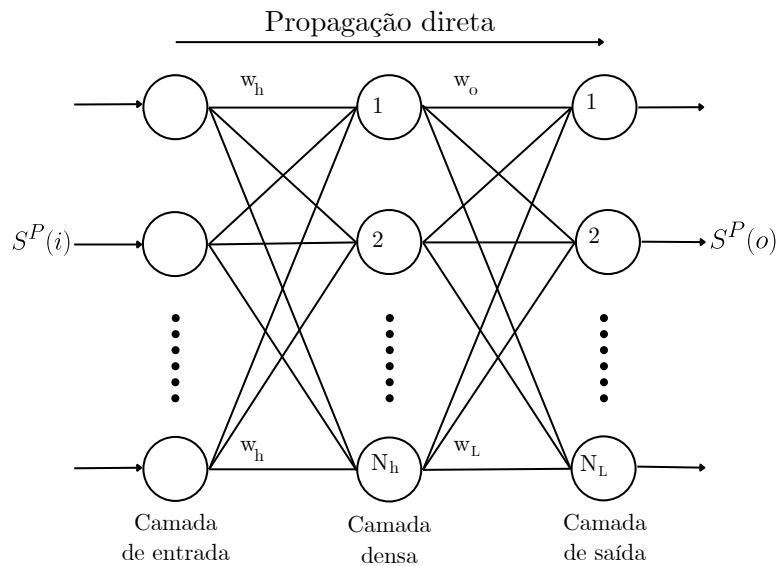


Figura 2.1: Representação gráfica de uma rede neural do tipo *feedforward*. Modificado de Hemeida et al. (2020)

As redes neurais são treinadas com uma quantidade finita de dados de entrada e saída, e é esperado que estas generalizem esses dados e produzam saídas desejáveis para as entradas fornecidas.

Nos últimos anos, muitos trabalhos têm sido realizados na área de redes neurais e aprendizado profundo, com destaque para aplicações em reconhecimento de voz, processamento de imagens, reconhecimento de texto e jogos (Dong et al. (2021)).

Existem diversos tipos de redes neurais, como as Redes Neurais *Feedforward*, em que as informações se propagam em uma única direção (Hemeida et al. (2020)), e as Redes Neurais Recorrentes, em que há conexões que permitem a retroalimentação de informações (Yu et al. (2019)).

Além disso, as redes neurais podem ter diferentes tipos de funções de ativação, que determinam como o sinal de entrada é transformado em um sinal de saída. Entre as funções de ativação mais comuns estão a função sigmoide, a função ReLU e a função tangente hiperbólica (Sharma et al. (2017)). Uma outra função de ativação amplamente utilizada é a função softplus.

A função de ativação softplus é uma função não-linear que é comumente usada em redes neurais artificiais. Essa função é uma versão suavizada da função de ativação ReLU (*Rectified Linear Unit*), que permite que a saída seja uma curva suave e contínua (Zheng et al. (2015)). A função softplus é definida como:

$$\text{softplus}(x) = \log(1 + e^x) \quad (2.25)$$

Essa função é positiva para todos os valores de entrada, o que significa que ela pode ser usada para ativar todas as unidades de uma rede neural. Além disso, a função de ativação softplus é derivável em todos os pontos, o que a torna útil para cálculo de gradientes durante o treinamento da rede.

# Capítulo 3

## Metodologia

### 3.1 Modelagem Lagrangiana do Pêndulo Duplo

Considere o sistema dinâmico constituído por um pêndulo duplo formado a partir da junção simples de um pêndulo a outro. Cada pêndulo consiste de uma partícula cujas massas são  $m_1$  e  $m_2$  e uma haste rígida de comprimentos  $l_1$  e  $l_2$ , respectivamente, e de massas desprezíveis. O pivô de rotação do sistema encontra-se na origem  $O$  e cada partícula possui suas coordenadas cartesianas  $(x_1, y_1)$  e  $(x_2, y_2)$  tal qual caracterizadas na figura 3.1:

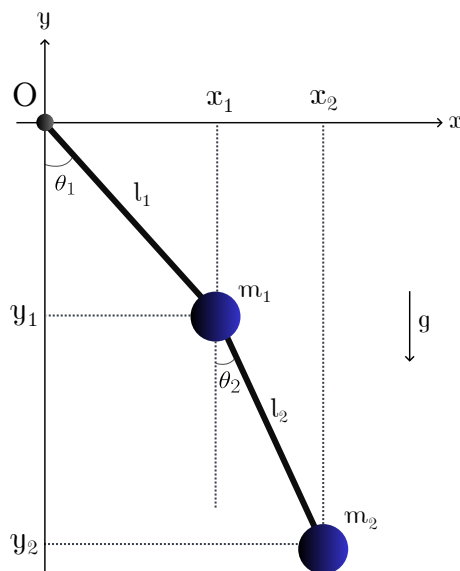


Figura 3.1: Representação gráfica do Pêndulo Duplo no sistema de coordenadas cartesiano. Fonte: autora, 2023.

Adicionalmente, sejam  $\theta_1$  e  $\theta_2$  os ângulos formados entre a primeira e a segunda haste com a direção vertical, respectivamente. Sendo assim, as coordenadas cartesianas das partículas podem ser representadas pelo seguinte sistema de equações:



$$\begin{aligned} x_1 &= l_1 \sin \theta_1 & y_1 &= -l_1 \cos \theta_1 \\ x_2 &= l_1 \sin \theta_1 + l_2 \sin \theta_2 & y_2 &= -l_1 \cos \theta_1 - l_2 \cos \theta_2 \end{aligned} \quad (3.1)$$

Derivando as equações do sistema 3.1 em relação ao tempo, obtém-se:

$$\begin{aligned} \dot{x}_1 &= l_1 \dot{\theta}_1 \cos \theta_1 & \dot{y}_1 &= -l_1 \dot{\theta}_1 \sin \theta_1 \\ \dot{x}_2 &= l_1 \dot{\theta}_1 \cos \theta_1 + l_2 \dot{\theta}_2 \cos \theta_2 & \dot{y}_2 &= -l_1 \dot{\theta}_1 \sin \theta_1 - l_2 \dot{\theta}_2 \sin \theta_2 \end{aligned} \quad (3.2)$$

Nesse ponto, percebe-se que ao cometer pequenas alterações nas condições iniciais desse sistema, a modelagem resultará em trajetórias totalmente diferentes e isso ocorre em um sistema dinâmico devido a sua natureza caótica. A partir daí, obtém-se um modelo de um sistema físico que depende apenas dos valores dos ângulos  $(\theta_1, \theta_2)$  e de suas velocidades angulares associadas  $(\dot{\theta}_1, \dot{\theta}_2)$ , sem depender das coordenadas cartesianas. Logo, pode-se calcular o termo cinético por:

$$T = \frac{1}{2} m_1 v_1^2 + \frac{1}{2} m_2 v_2^2 \quad (3.3)$$

Decompondo as velocidades no eixo cartesiano, obtém-se que:  $v_1^2 = \dot{x}_1^2 + \dot{y}_1^2$  e  $v_2^2 = \dot{x}_2^2 + \dot{y}_2^2$ . Substituindo essas expressões em 3.3:

$$T = \frac{1}{2} m_1 (\dot{x}_1^2 + \dot{y}_1^2) + \frac{1}{2} m_2 (\dot{x}_2^2 + \dot{y}_2^2) \quad (3.4)$$

Utilizando os valores obtidos no sistema de equações 3.2 na expressão 3.4, obtém-se a equação da energia cinética para o pêndulo duplo:

$$T = \frac{1}{2} m_1 l_1^2 \dot{\theta}_1^2 + \frac{1}{2} m_2 [l_1^2 \dot{\theta}_1^2 + l_2^2 \dot{\theta}_2^2 + 2l_1 l_2 \dot{\theta}_1 \dot{\theta}_2 \cos(\theta_1 - \theta_2)] \quad (3.5)$$

Em seguida, calcula-se o termo potencial da Lagrangiana, a qual é expressa por:

$$V = m_1 g y_1 + m_2 g y_2 \quad (3.6)$$

Substituindo 3.1 em 3.6, obtém-se a equação da energia potencial para o pêndulo duplo em coordenadas não-cartesianas, tal qual a equação da energia cinética:

$$V = -(m_1 + m_2) g l_1 \cos \theta_1 - m_2 g l_2 \cos \theta_2 \quad (3.7)$$

Portanto, o valor da Lagrangiana equivalente ao sistema dinâmico pêndulo duplo é dado por:

$$\begin{aligned} L &= \frac{1}{2} (m_1 + m_2) l_1^2 \dot{\theta}_1^2 + \frac{1}{2} m_2 l_2^2 \dot{\theta}_2^2 + m_2 l_1 l_2 \dot{\theta}_1 \dot{\theta}_2 \cos(\theta_1 - \theta_2) \\ &\quad + (m_1 + m_2) g l_1 \cos \theta_1 + m_2 g l_2 \cos \theta_2 \end{aligned} \quad (3.8)$$

A partir daí, o momento canônico associado aos valores dos ângulos  $(\theta_1, \theta_2)$  pode ser obtido a partir da derivada parcial da Lagrangiana em relação às suas velocidades angulares  $(\dot{\theta}_1, \dot{\theta}_2)$ , de modo que:

$$\begin{aligned} p_{\theta_1} &= \frac{\partial L}{\partial \dot{\theta}_1} = (m_1 + m_2)l_1^2\dot{\theta}_1 + m_2l_1l_2\dot{\theta}_2 \cos(\theta_1 - \theta_2) \\ p_{\theta_2} &= \frac{\partial L}{\partial \dot{\theta}_2} = m_2l_2^2\dot{\theta}_2 + m_2l_1l_2\dot{\theta}_1 \cos(\theta_1 - \theta_2) \end{aligned} \quad (3.9)$$

Da equação 2.24, segue que as equações de movimento para o sistema pêndulo duplo são obtidas através da expressão:

$$\frac{\partial L}{\partial q_i} - \frac{d}{dt} \frac{\partial L}{\partial \dot{q}_i} = 0 \Rightarrow \frac{dp_{\theta_i}}{dt} - \frac{\partial L}{\partial \theta_i} = 0 \quad (3.10)$$

onde  $i = 1, 2$ . Em posse das equações supracitadas, obtém-se a solução analítica para as equações físicas que descrevem o movimento duplo-pendular, ocasião em que serão extraídos os dados de simulação para fins de comparação com a Rede Neural Lagrangiana implementada.

## 3.2 Aprendizado Profundo Lagrangiano

A partir desse ponto a função de Lagrange passa a ser uma caixa preta para a rede neural, isto é, uma função arbitrária, complexa e desconhecida para a rede. Como os dados do sistema físico foram obtidos na seção 3.1, o próximo passo deve ser a parametrização do Lagrangiano com uma rede neural tal que  $L = L_\theta$ . Então, ajustando a equação 3.10 para a notação vetorial, obtém-se:

$$\frac{d}{dt} \nabla_{\dot{q}} L = \nabla_q L \quad (3.11)$$

Expandindo a equação anterior com a derivada temporal, obtém-se:

$$(\nabla_{\dot{q}} \nabla_{\dot{q}}^\top L) \ddot{q} + (\nabla_q \nabla_{\dot{q}}^\top L) \dot{q} = \nabla_q L \quad (3.12)$$

Por fim, a dinâmica do movimento é obtida com a matriz inversa resolvida em  $\ddot{q}$ :

$$\ddot{q} = (\nabla_{\dot{q}} \nabla_{\dot{q}}^\top L)^{-1} [\nabla_q L - (\nabla_q \nabla_{\dot{q}}^\top L) \dot{q}] \quad (3.13)$$

Sendo assim, para cada conjunto de coordenadas  $x_t = (q_t, \dot{q}_t)$ , existe um método para calcular  $\dot{x}_t = (\dot{q}_t, \ddot{q}_t)$  já com a restrição imposta pela Equação de Euler-Lagrange. Para avançar, é necessário retropropagar os dados através dessa restrição e treinar um modelo paramétrico que se aproxime do Lagrangiano real.

### 3.2.1 Implementação da LNN

#### Comparação entre os modelos Analítico e Autograd

A solução da equação 3.13 torna-se elementar com a utilização da biblioteca JAX (Bradbury et al. (2018)), a qual possui a finalidade de realizar cálculos numéricos eficientes em GPUs e TPUs, especialmente para aplicações de aprendizado de máquina.

Antes do próximo passo, é preciso conferir se os modelos obtidos estão se aproximando do caso real para o pêndulo duplo via solução analítica. Sendo assim, foi escolhido um estado inicial aleatório e obteve-se a sua dinâmica através do método de cálculo de auto-gradiente do JAX e sua solução analítica. Em seguida, a solução analítica foi minimamente perturbada para medir sua sensibilidade às condições iniciais do sistema.

#### Pré-processamento e Treinamento da LNN

As coordenadas generalizadas do sistema foram normalizadas no intervalo  $[-\pi, \pi]$ . Os parâmetros foram estabelecidos da seguinte forma:  $m_1 = m_2 = 1kg$ ,  $l_1 = l_2 = 1m$  e  $g = 9,8m/s^2$ . O estado inicial do sistema foi definido com os módulos:  $q(t_0) = \dot{q}(t_0) = 10^{-10}$ .

Em seguida, utilizando o método numérico Runge-Kutta de 4ª Ordem associado as funções da biblioteca JAX, foram geradas 1500 amostras de treinamento e 1500 amostras de teste para as coordenadas generalizadas de ambas partículas constituintes do pêndulo duplo.

A arquitetura da LNN é composta por 3 camadas densas e a função de ativação utilizada foi a *Softplus*, visto que esta possui uma performance mais acentuada em uma simulação que lida com equações diferenciais parciais de 2ª ordem (Cranmer et al. (2020)).

A função de *loss* foi implementada utilizando o Erro Quadrático Médio (MSE) com base em funções *built-in* da biblioteca JAX. O treinamento foi realizado com 1500 *batches* de tamanho igual a 100. Ademais, foi utilizado o otimizador Adam e uma taxa de aprendizado igual a  $10^{-3}$ .

A figura 3.2 ilustra o diagrama que representa o fluxo do processo:

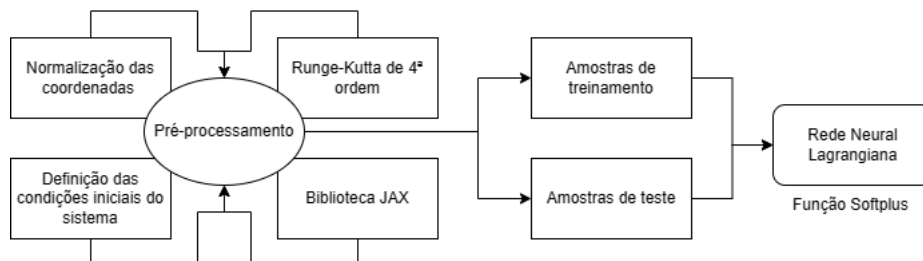


Figura 3.2: Diagrama de blocos para o fluxo do pré-processamento.

# Capítulo 4

## Resultados e Discussões

Em posse das equações que descrevem a dinâmica do sistema físico pêndulo duplo e dos dados obtidos por este modelo, foi preciso constatar se as soluções analítica e a obtida pela função Autograd da biblioteca JAX seriam próximas, para confirmar a hipótese de que a solução via Autograd é uma abordagem viável. Essa comparação sucedeu-se de modo que foram introduzidas perturbações no sistema da ordem de  $10^{-11}$  (figura 4.1 (a)) e, posteriormente,  $10^{-12}$  (figura 4.1 (b)), a fim de avaliar o comportamento do sistema perturbado em relação ao sistema ideal. Dessa forma, é estabelecido um padrão que permite distinguir uma leve perturbação no sistema de uma solução errônea. Em seguida, comparou-se a solução analítica à solução proveniente do Autograd (figura 4.1 (c)).

Dessas observações, constata-se que o sistema dinâmico em questão é sensível às condições iniciais e que a solução analítica ideal do sistema aproxima-se da solução via Autograd de maneira análoga a forma como a solução ideal aproxima-se de uma perturbação de ordem  $10^{-11}$ . Além disso, os resultados apontam que pequenas perturbações não causam interferência significativa no comportamento do pêndulo.

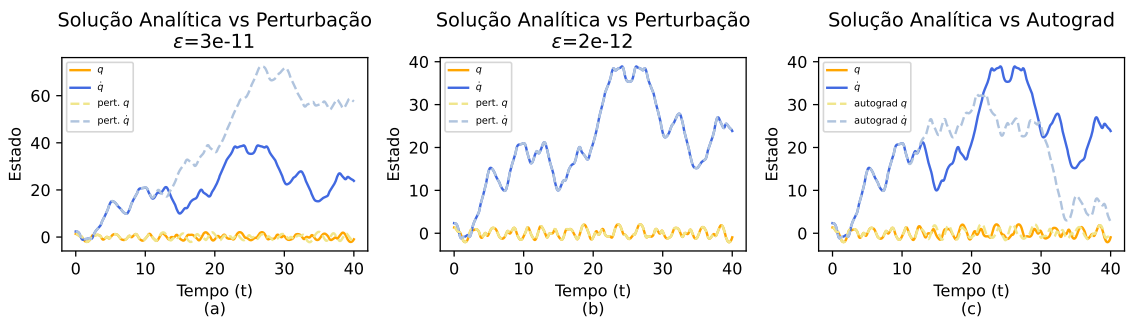


Figura 4.1: Comparação entre as soluções. (Unidades de medida - estado: graus, tempo: segundos) (a) Comparação entre a solução analítica sem perturbação e com uma perturbação  $\epsilon = 3 \cdot 10^{-11}$ ; (b) Comparação entre a solução analítica sem perturbação e com uma perturbação  $\epsilon = 2 \cdot 10^{-12}$ ; (c) Comparação entre a solução analítica e o autogradiente calculado pela ferramenta JAX.

Posteriormente, foi feita a divisão das amostras geradas em um conjunto de dados de teste e um conjunto de dados de treino. Com o fito de visualizar a distribuição bidimensional desses dados, esses valores foram plotados em formatos de vetores  $(\theta_1, \theta_2)$  e observou-se que os dados estavam equivalentemente distribuídos entre treinamento (figura 4.2 (a)) e teste (figura 4.2 (b)), e o modelo é capaz de aprender a reconhecer padrões de forma mais geral e pode generalizar melhor para novas amostras. Isso significa que o modelo tem uma maior capacidade de classificar corretamente amostras que nunca foram vistas antes. Vale ressaltar que se os conjuntos de treino e teste não estivessem distribuídos de forma equivalente, o modelo poderia ser enviesado ou ter uma precisão falsamente elevada, ou seja, um *overfitting*.

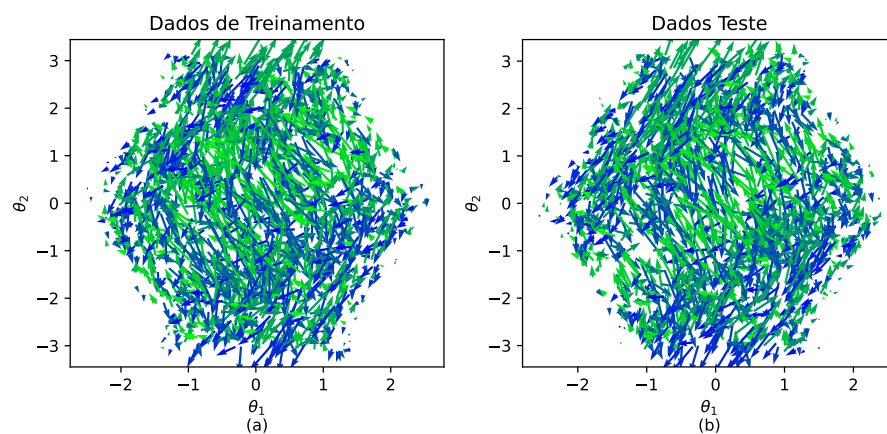


Figura 4.2: Comparação da distribuição dos dados entre conjuntos de treinamento e teste.

Dando sequência a modelagem, após o treinamento foi obtido o gráfico referente a função de *loss* do modelo. O MSE é uma medida de quão bem o modelo está conseguindo prever os valores esperados e, portanto, um baixo MSE indica um bom desempenho do modelo. Na figura 4.3, é possível inferir que entre os passos 0 e 800, o modelo está no seu período de treinamento, pois o MSE está diminuindo. Isso indica que o modelo está aprendendo a relação entre as variáveis e melhorando seu desempenho.

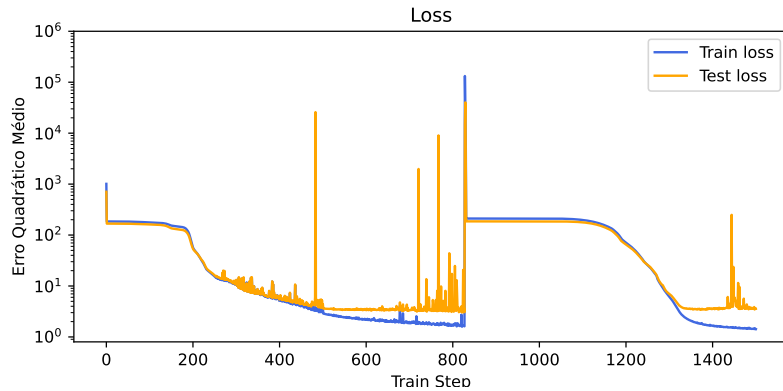


Figura 4.3: Loss calculado através do MSE ao longo do treinamento

Entretanto, observa-se que entre o passo 800 e 900 há uma grande variação positiva do valor do MSE seguida por uma diminuição deste. É importante mencionar que se o modelo de *deep learning* fizer uma previsão muito ruim em relação ao valor real de uma observação, a elevação ao quadrado presente na função do MSE irá amplificar esse erro.

Isso ocorre porque o termo quadrático torna o erro positivo, independentemente da direção em que o erro ocorre. Dessa forma, um erro maior será mais significativo e terá um impacto maior no valor total do MSE. Contudo, apesar dos *outliers* presentes neste intervalo, o modelo não apresenta grandes flutuações e encontra estabilidade nos passos seguintes.

Isso fica evidente na figura 4.4, onde os gráficos que comparam as velocidades angulares *vs.* previstas (4.4 (a)) e as acelerações angulares *vs.* previstas (4.4 (b)) sugerem uma relação linear entre estas variáveis, destacando a mínima interferência de *outliers*.

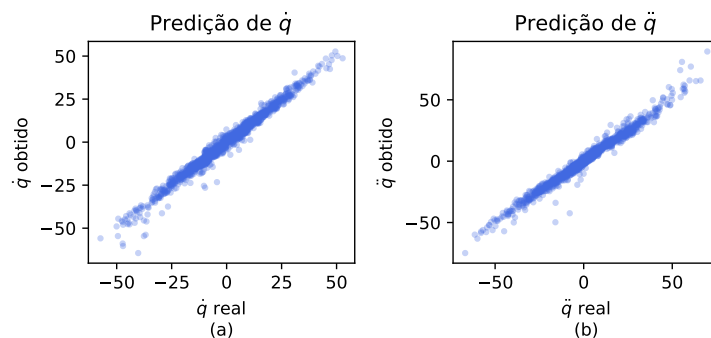


Figura 4.4: (a) Velocidade angular real *vs.* prevista; (a) aceleração angular real *vs.* prevista;

Um método de avaliar a eficiência do modelo é examinar como o modelo está se comportando no espaço de coordenadas bidimensional. Para fazer essa análise, é preciso testar o modelo em diferentes pontos do espaço de coordenadas e verificar como as previsões estão se comportando em cada um desses pontos, como ilustrado na figura 4.5:

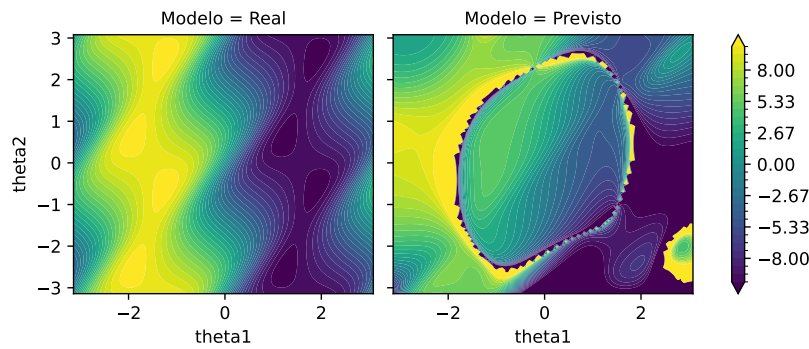


Figura 4.5: Desempenho dos modelos reais e previstos no espaço de coordenadas no quesito velocidade angular.

O mapa de calor acima revela as curvas de nível para as velocidades angulares atingidas pelo sistema pêndulo duplo real no quadro da esquerda e pelo modelo previsto no quadro à direita. Daí, observa-se que as regiões amarelas e as regiões roxas estão predominantemente ocupando o mesmo espaço de coordenadas em ambos os gráficos, mostrando que o sistema previsto comporta-se bem em amplitudes maiores.

Adicionalmente, identifica-se regiões de transição entre as curvas de nível nas cores azuladas, representando os *outliers* do modelo, além de indicar que o sistema é mais impreciso perto da posição de equilíbrio, ou seja, quando os movimentos possuem amplitudes menores. Isso ocorre em função dos valores muito pequenos para os ângulos próximo a posição de equilíbrio, os quais são amplificados no cálculo do MSE, pois são elevados ao quadrado. Os *outliers* nos resultados da simulação representam previsões errôneas da rede neural sobre as velocidades e acelerações angulares das partículas do sistema. Nesse caso, há poucos *outliers*, indicando que o modelo é satisfatoriamente acurado.

Analogamente, o mapa de calor da figura 4.6 faz uma avaliação em relação à aceleração angular das partículas constituintes do pêndulo. Nesse gráfico, identifica-se uma maior similaridade do modelo real com o modelo previsto, porém, contendo as particularidades à precisão do sistema, o que pode ser entendido como pequenas perturbações na conservação da energia.

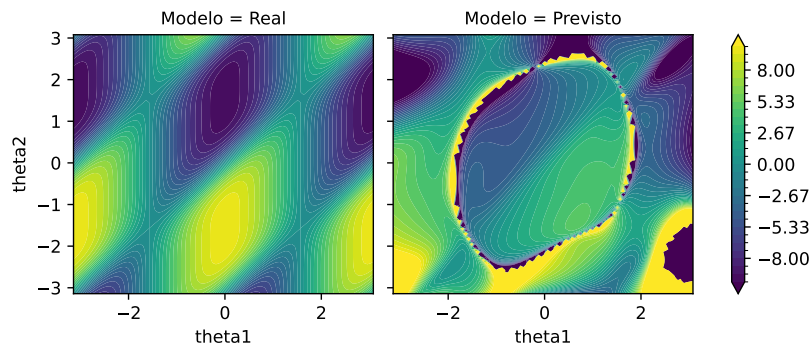


Figura 4.6: Desempenho dos modelos reais e previstos no espaço de coordenadas no quesito aceleração angular.

Por fim, foi feita uma comparação entre os dados obtidos pela solução analítica e os dados obtidos após a inserção de uma perturbação  $\epsilon = 0,001$  no sistema (figura 4.7 (a)) em função do tempo com o objetivo de estabelecer uma expectativa para os resultados da LNN. Na figura 4.7 (b), tem-se a comparação entre a solução analítica e a solução proveniente da LNN.

Essa comparação evidencia a similaridade entre o comportamento das partículas previsto pela rede neural e obtido pela solução analítica, com e sem perturbação inserida, por meio das curvas de velocidade angular para as duas partículas.

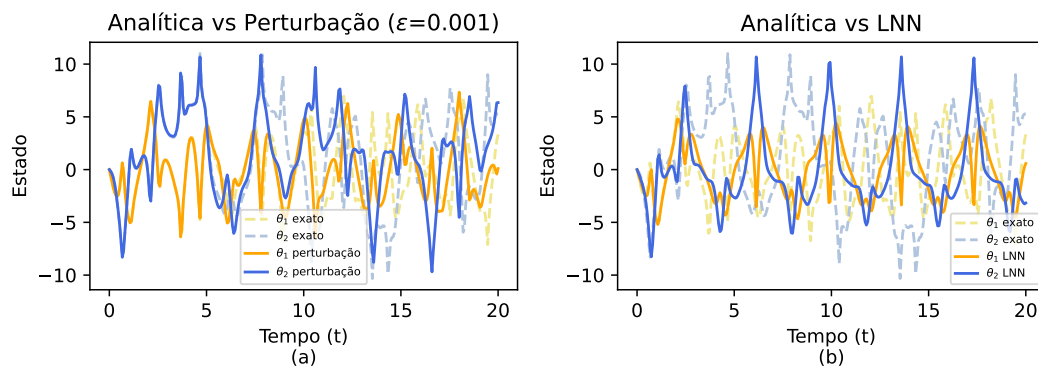


Figura 4.7: Comparação entre (a) solução analítica vs. perturbação; (b) solução analítica vs. LNN.

Diante disso, observa-se que a LNN aproxima-se da solução de forma muito semelhante a que a solução aproxima-se da solução com perturbação e que, apesar das flutuações no valor das amplitudes, a LNN é capaz de simular a conservação de energia no sistema e prever o movimento do pêndulo duplo ideal, pois não há redução significativa da amplitude e o pêndulo permanece oscilando.

Por fim, foi elaborada uma representação visual da dinâmica do pêndulo duplo lagrangiano e esta encontra-se disponível em: <https://drive.google.com/file/d/>



[14PdIuYCe1T4-b5PvTNVnb0qsJ3cf76Gx/view?usp=share\\_link](https://www.google.com/search?q=14PdIuYCe1T4-b5PvTNVnb0qsJ3cf76Gx/view?usp=share_link)

# Capítulo 5

## Conclusão

Após analisar os resultados obtidos a partir da modelagem do sistema físico pêndulo duplo, é possível concluir que a solução via LNN é uma abordagem viável para se obter soluções numéricas que se aproximam da solução analítica ideal do sistema. Além disso, verificou-se que pequenas perturbações no sistema não causam interferência significativa em seu comportamento.

Foi possível também observar que o sistema dinâmico é sensível às condições iniciais, o que indica a importância de se realizar uma análise cuidadosa e minuciosa na escolha dessas condições para se obter resultados precisos e confiáveis.

A divisão das amostras em um conjunto de treino e um conjunto de teste possibilitou visualizar a distribuição bidimensional desses dados e avaliar a capacidade do modelo de aprender a reconhecer padrões de forma mais geral e generalizar melhor para novas amostras. Verificou-se que os conjuntos de treino e teste estavam distribuídos de forma equivalente, o que indica que o modelo não apresenta enviesamento e terá uma maior precisão.

A análise da função de *loss* permitiu inferir que o modelo possui um período de treinamento e que, a partir de um certo momento, o modelo apresenta estabilidade em seus resultados a partir dos valores do MSE encontrados.

A comparação entre as velocidades e acelerações angulares reais e previstas sugeriu uma relação linear entre essas variáveis e evidenciou a mínima interferência de outliers. O mapa de calor gerado a partir da análise das curvas de nível para as velocidades angulares atingidas pelo sistema pêndulo duplo e pelo modelo previsto revelou que as regiões de maior amplitude estão ocupando, majoritariamente, o mesmo espaço de coordenadas em ambos os casos, o que indica que o modelo está se comportando de maneira satisfatória.

Logo, a modelagem do sistema físico pêndulo duplo com a LNN foi bem sucedida e os resultados obtidos a partir da solução via Autograd se aproximaram da solução analítica do sistema. Portanto, a metodologia estudada propõe uma abordagem promissora para a modelagem de sistemas dinâmicos que pode ser útil para modelar uma ampla variedade de sistemas dinâmicos, incluindo sistemas mecânicos, elétricos e biológicos com a introdução

de uma nova classe de modelos neurais que incorpora princípios físicos, como a equação de Lagrange, para descrever a conservação da energia do sistema.

## **5.1 Trabalhos Futuros**

Com base nos resultados obtidos a partir da modelagem do sistema físico pêndulo duplo utilizando a abordagem da LNN, surgem diversas oportunidades para futuras pesquisas e desenvolvimentos nessa área promissora, tais quais a investigação mais aprofundada sobre o requisito computacional da abordagem LNN, a qual é necessária para entender melhor a eficiência e escalabilidade da técnica; e a aplicação da LNN em outros sistemas dinâmicos. Portanto, um próximo passo natural seria aplicar essa técnica a outros sistemas dinâmicos, como sistemas mecânicos, elétricos e biológicos.

# Referências Bibliográficas

- Antsaklis, P. J. et al. (1990). Neural networks for control systems. *IEEE Transactions on Neural Networks*, 1(2):242–244.
- Bradbury, J., Frostig, R., Hawkins, P., Johnson, M. J., Leary, C., Maclaurin, D., Necula, G., Paszke, A., VanderPlas, J., Wanderman-Milne, S., and Zhang, Q. (2018). JAX: composable transformations of Python+NumPy programs.
- Chen, Q., Cheng, W., Gao, L., and Fottner, J. (2021). A pure neural network controller for double-pendulum crane anti-sway control: Based on lyapunov stability theory. *Asian Journal of Control*, 23(1):387–398.
- Cranmer, M., Greydanus, S., Hoyer, S., Battaglia, P., Spergel, D., and Ho, S. (2020). Lagrangian neural networks. *arXiv preprint arXiv:2003.04630*.
- Dong, S., Wang, P., and Abbas, K. (2021). A survey on deep learning and its applications. *Computer Science Review*, 40:100379.
- Goldstein, H., Poole, C., and Safko, J. (2002). *Classical mechanics*. American Association of Physics Teachers.
- Greiner, W. and Bromley, D. A. (2003). *Classical mechanics: systems of particles and Hamiltonian dynamics*. Springer.
- Greydanus, S., Dzamba, M., and Yosinski, J. (2019). Hamiltonian neural networks. *Advances in neural information processing systems*, 32.
- Hemeida, A. M., Hassan, S. A., Mohamed, A.-A. A., Alkhalaf, S., Mahmoud, M. M., Senjyu, T., and El-Din, A. B. (2020). Nature-inspired algorithms for feed-forward neural network classifiers: A survey of one decade of research. *Ain Shams Engineering Journal*, 11(3):659–675.
- Lee, K. and Carlberg, K. T. (2020). Model reduction of dynamical systems on nonlinear manifolds using deep convolutional autoencoders. *Journal of Computational Physics*, 404:108973.
- Lemos, N. A. (2007). *Mecânica analítica*. Editora Livraria da Física.

- Li, Q., Gama, F., Ribeiro, A., and Prorok, A. (2020). Graph neural networks for decentralized multi-robot path planning. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 11785–11792. IEEE.
- Liu, C., Wen, G., Zhao, Z., and Sedaghati, R. (2020). Neural-network-based sliding-mode control of an uncertain robot using dynamic model approximated switching gain. *IEEE transactions on cybernetics*, 51(5):2339–2346.
- Nascimento, M. (2020). Histedbr UNICAMP - Navegando e Aprendendo: Glossário - Jean Le Rond d’Alembert. [http://www.histedbr.fe.unicamp.br/navegando/glossario/verb\\_b\\_jean\\_le\\_rond\\_d.htm](http://www.histedbr.fe.unicamp.br/navegando/glossario/verb_b_jean_le_rond_d.htm). Acesso em: 30 de abril de 2023.
- Selmic, R. R. and Lewis, F. L. (2000). Deadzone compensation in motion control systems using neural networks. *IEEE Transactions on Automatic Control*, 45(4):602–613.
- Sharma, S., Sharma, S., and Athaiya, A. (2017). Activation functions in neural networks. *Towards Data Sci*, 6(12):310–316.
- Taylor, J. R. (2005). *Classical mechanics*. Number 531 TAY.
- Xiang, W., Musau, P., Wild, A. A., Lopez, D. M., Hamilton, N., Yang, X., Rosenfeld, J., and Johnson, T. T. (2018). Verification for machine learning, autonomy, and neural networks survey. *arXiv preprint arXiv:1810.01989*.
- Yu, Y., Si, X., Hu, C., and Zhang, J. (2019). A review of recurrent neural networks: Lstm cells and network architectures. *Neural computation*, 31(7):1235–1270.
- Zhang, M. and Jing, X. (2021). Adaptive neural network tracking control for double-pendulum tower crane systems with nonideal inputs. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 52(4):2514–2530.
- Zheng, H., Yang, Z., Liu, W., Liang, J., and Li, Y. (2015). Improving deep neural networks using softplus units. In *2015 International joint conference on neural networks (IJCNN)*, pages 1–4. IEEE.