



UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO TECNOLÓGICO
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE AUTOMAÇÃO E
SISTEMAS

Ana Cláudia Banderchuk

**Teoria de Controle Robusto Combinada com Ferramentas de Aprendizado de Máquina
para Sistemas Não-Lineares Incertos Sujeitos à Perturbações Persistentes**

Florianópolis
2023

Ana Cláudia Banderchuk

**Teoria de Controle Robusto Combinada com Ferramentas de Aprendizado de Máquina
para Sistemas Não-Lineares Incertos Sujeitos à Perturbações Persistentes**

Dissertação submetida ao Programa de Pós-Graduação em Engenharia de Automação e Sistemas da Universidade Federal de Santa Catarina para a obtenção do título de mestre em Engenharia de Automação e Sistemas.
Orientador: Prof. Daniel Ferreira Coutinho, Dr.
Coorientador: Prof. Eduardo Camponogara, Dr.

Florianópolis
2023

Ficha de identificação da obra elaborada pelo autor,
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

Banderchuk, Ana Cláudia

Teoria de Controle Robusto Combinada com Ferramentas de
Aprendizado de Máquina para Sistemas Não-Lineares Incertos
Sujeitos à Perturbações Persistentes / Ana Cláudia
Banderchuk ; orientador, Daniel Ferreira Coutinho,
coorientador, Eduardo Camponogara, 2024.

63 p.

Dissertação (mestrado) - Universidade Federal de Santa
Catarina, Centro Tecnológico, Programa de Pós-Graduação em
Engenharia de Automação e Sistemas, Florianópolis, 2024.

Inclui referências.

1. Engenharia de Automação e Sistemas. 2. Controle
robusto. 3. Aprendizado de Máquina. 4. Sistemas Não
Lineares. I. Coutinho, Daniel Ferreira. II. Camponogara,
Eduardo. III. Universidade Federal de Santa Catarina.
Programa de Pós-Graduação em Engenharia de Automação e
Sistemas. IV. Título.

Ana Cláudia Banderchuk

**Teoria de Controle Robusto Combinada com Ferramentas de Aprendizado de Máquina
para Sistemas Não-Lineares Incertos Sujeitos à Perturbações Persistentes**

O presente trabalho em nível de mestrado foi avaliado e aprovado por banca examinadora composta pelos seguintes membros:

Prof. Daniel Ferreira Coutinho, Dr.
Universidade Federal de Santa Catarina

Prof. Eric Aislan Antonelo, Dr.
Universidade Federal de Santa Catarina

Prof. Aurélio Tergolina Salton, Dr.
Universidade Federal do Rio Grande do Sul

Certificamos que esta é a **versão original e final** do trabalho de conclusão que foi julgado adequado para obtenção do título de mestre em Engenharia de Automação e Sistemas.

Coordenação do Programa de Pós-Graduação

Prof. Daniel Ferreira Coutinho, Dr.
Orientador

Florianópolis, 2023.

RESUMO

Este trabalho apresenta uma estratégia que combina um controlador robusto e uma lei de controle gerada por uma Rede Neural de Estado de Eco para estabilizar uma classe de sistemas incertos não-lineares de tempo discreto sujeitos a perturbações persistentes. Primeiro, o controlador robusto é projetado para assegurar a estabilidade entrada-estado do sistema em malha fechada, considerando uma região de estabilidade garantida, independente da ação de controle gerada pela rede neural e das perturbações exógenas. Em seguida, treina-se a rede neural para minimizar os efeitos das perturbações na saída do sistema. A eficácia dessa estratégia é numericamente demonstrada nos sistemas do Oscilador de Van der Pol e de uma bomba elétrica submersível.

Palavras-chave: Controle robusto. Rede Neural de Estado de Eco. Sistemas Não-Lineares.

ABSTRACT

This work proposes a control strategy consisting of a robust controller and an Echo State Network (ESN) based control law for stabilizing a class of uncertain nonlinear discrete-time systems subject to persistent disturbances. Firstly, the robust controller is designed to ensure that the closed-loop system is Input-to-State Stable (ISS) with a guaranteed stability region regardless of the ESN control action and exogenous disturbances. Then, the ESN-based controller is trained in order to mitigate the effects of disturbances on the system output. The potential of the proposed control design method is numerically demonstrated in the systems of the Van der Pol Oscillator and a submersible electric pump.

Keywords: Robust control. Echo State Network. Nonlinear Systems.

LISTA DE FIGURAS

Figura 1 – Agrupamentos de dados com entradas bidimensionais. Adaptado de Nelles (2013).	21
Figura 2 – Mínimos quadrados para ajuste de uma reta. Adaptado de Goodfellow, Bengio e Courville (2016).	22
Figura 3 – Estrutura básica de uma rede neural com três camadas, onde as setas representam cada uma das conexões entre os neurônios, que por sua vez, são representados pelos círculos.	24
Figura 4 – Estrutura de um neurônio i de uma camada l de uma rede neural, com $w_{ij}^{(\ell)}$ sendo os elementos de $W^{(\ell)}$	24
Figura 5 – Funções de Ativação.	25
Figura 6 – Ilustração de uma Rede Neural de Estado de Eco.	27
Figura 7 – Diagrama simplificado do Controlador Proposto.	30
Figura 8 – Configuração da ESN para treinamento do modelo inverso.	36
Figura 9 – Configuração da ESN para controle utilizando modelo inverso.	37
Figura 10 – Representação do processo de geração do sinal utilizado para o treinamento da rede neural.	40
Figura 11 – Resposta do sistema de Van der Pol em malha fechada (representada pelas linhas vermelhas na parte superior das Figuras) para $u(t) = u_1(t)$ (linha tracejada) e $u(t) = u_1(t) + u_2(t)$ (linha contínua). As condições iniciais são $x_0^T = [-0,001 \ 0,001 \ 0]$, e o parâmetro θ é fixo em 0,75; a perturbação no sistema é destacada em azul na parte inferior das Figuras.	41
Figura 12 – Sinal gerado pela configuração de controle proposta para o sistema de Van der Pol.	42
Figura 13 – Histograma da distribuição de densidade de probabilidade do ganho de desempenho para perturbações do tipo $d(t) = 0,25(\text{sen}(\omega_a t) + \text{sen}(\omega_b t))$. A linha tracejada em vermelho representa a função de densidade de probabilidade de uma distribuição Gaussiana com média de 77,36% e variância de 10,43% de $G(y, y_0)$	43
Figura 14 – Estimativa da região de atração e retrato de fase das trajetórias do estado para $x_0^T = [0,0912 \ -0,286 \ -0,0279]$, $\theta = 0,75$, e $d(t) = 0,25(1 + \sin(\frac{5}{4}t) \sin(\frac{1}{4}t))$, considerando as leis de controle $u = u_1$ (linha tracejada vermelha) e $u = u_1 + u_2$ (linha contínua preta).	43
Figura 15 – Representação do sistema de um poço de produção de petróleo com elevação artificial de fluidos controlada com uma ESP, adaptado de Binder, Pavlov e Johansen (2015).	44

Figura 16 – Resposta do sistema da bomba elétrica submersível em malha fechada (representada pelas linhas vermelhas na parte superior das Figuras) para $u(t) = u_1(t)$ (linha tracejada) e $u(t) = u_1(t) + u_2(t)$ (linha contínua). As condições iniciais são $x_0^T = [-0,005 \ 0,01 \ -0,015 \ 0,2]$; a perturbação no sistema é destacada em azul na parte inferior das Figuras.	48
Figura 17 – Sinal gerado pela configuração de controle proposta para o sistema da bomba elétrica submersível.	49
Figura 18 – Histograma da distribuição de densidade de probabilidade do ganho de desempenho para perturbações do tipo $d(t) = 0,05(\text{sen}(w_a t) + \text{sen}(w_b t))$. A linha tracejada em vermelho representa a função de densidade de probabilidade de uma distribuição Gaussiana com média de 38,67% e variância de 20,08% de $G(y, y_0)$	50
Figura 19 – Representação de (a) um conjunto convexo, (b) um conjunto não-convexo e (c) um envoltório convexo.	58
Figura 20 – Exemplo de código em Python utilizando framework Optuna para minimização da função $(x - 2)^2$	61
Figura 21 – Função objetivo para otimização dos hiperparâmetros do sistema de Van der Pol utilizando o framework Optuna em Python.	62

LISTA DE ABREVIATURAS E SIGLAS

ESN	<i>Echo State Network</i>
ESP	<i>Electric Submersible Pump</i>
ISS	<i>Input-to-State Stability</i>
LMI	<i>Linear Matrix Inequalities</i>
LPV	<i>Linear Parameter-Varying</i>
RNN	<i>Recurrent Neural Network</i>
SDP	<i>Semidefinite Programming</i>

LISTA DE SÍMBOLOS

\mathbb{R}	Conjunto de todos os números reais
\mathbb{R}^n	Conjunto de todos os vetores reais de dimensão n
$\mathbb{R}^{n \times m}$	Conjunto de todas as matrizes reais de dimensão $n \times m$
\mathbb{R}_{\geq}	Conjunto de todos os números reais não negativos
$\mathbb{R}_{>}$	Conjunto de todos os números reais positivos
\mathbb{R}_{\leq}	Conjunto de todos os números reais não positivos
$\mathbb{R}_{<}$	Conjunto de todos os números reais negativos
$P > 0$	P é uma matriz simétrica definida positiva
$P \geq 0$	P é uma matriz simétrica semi-definida positiva
$P < 0$	P é uma matriz simétrica definida negativa
$P \leq 0$	P é uma matriz simétrica semi-definida negativa
\mathbb{S}^n	Conjunto de todas as matrizes simétricas em $\mathbb{R}^{n \times n}$
I_n	Matriz identidade de ordem n
$0_{m \times n}$	Matriz de dimensão $m \times n$ onde todos os elementos são zero
$A \otimes B$	Produto de Kronecker entre as matrizes A e B
$\begin{bmatrix} a & \star & \star \\ b & c & \star \\ d & e & f \end{bmatrix}$	Matriz simétrica formada por elementos reais em que o símbolo \star no elemento ij denota o transposto do elemento ji
x_+	Representação da operação de deslocamento temporal de um passo à frente de $x(k)$
x_0	Valor do vetor de estados no tempo t (ou amostra k) igual a zero

SUMÁRIO

1	INTRODUÇÃO	11
1.1	OBJETIVOS	13
1.2	CONTRIBUIÇÕES	13
1.3	RESUMO DA DISSERTAÇÃO	14
2	RESULTADOS INSTRUMENTAIS	15
2.1	CONTROLE ROBUSTO	15
2.1.1	Estabilidade por Lyapunov	15
2.1.2	Estabilização por Lyapunov	18
2.2	APRENDIZADO DE MÁQUINA	20
2.2.1	Mínimos Quadrados	21
2.2.2	Redes Neurais	23
2.2.3	Redes Neurais Recorrentes	26
2.2.4	Redes Neurais de Estado de Eco	27
3	PROJETO DE CONTROLE	30
3.1	DEFINIÇÃO DO PROBLEMA DE CONTROLE	30
3.2	PROJETO DE CONTROLE ROBUSTO	32
3.3	PROJETO DE APRENDIZADO DE MÁQUINA	35
4	RESULTADOS NUMÉRICOS	38
4.1	VAN DER POL	38
4.2	BOMBA ELÉTRICA SUBMERSÍVEL	42
5	CONCLUSÕES	51
5.1	PERSPECTIVAS FUTURAS	51
	REFERÊNCIAS	53
	APÊNDICE A – DESIGUALDADES MATRICIAIS LINEARES	57
	APÊNDICE B – DEFINIÇÕES RELEVANTES PARA ANÁLISE CONVEXA	58
	APÊNDICE C – LEMAS ESSENCIAIS PARA CONTROLE ROBUSTO COM USO DE LMIS	60
	APÊNDICE D – OTIMIZAÇÃO DE HIPERPARÂMETROS COM O FRAMEWORK OPTUNA	61

1 INTRODUÇÃO

Sistemas dinâmicos estão presentes em toda a parte, abrangendo tanto sistemas naturais, como os biológicos e climáticos, quanto sistemas artificiais, ou seja, aqueles que são projetados pelo ser humano, como robôs e aeronaves. De forma geral, quando é possível manipular as variáveis de entrada de um sistema e medir suas respostas correspondentes, esses sistemas, denominados de plantas, podem ser controlados através do uso de um controlador.

Esses controladores podem ser projetados de diversas formas, utilizando desde técnicas clássicas, como controle PID, até abordagens de controle moderno, como controle ótimo e controle preditivo baseado em modelo. Em geral, a grande maioria dessas técnicas requerem a modelagem ou a identificação do modelo da planta, para que, em seguida, o controlador seja projetado com base no modelo da planta obtido, considerando que esse modelo representa o sistema real (HOU; WANG, Z., 2013).

Porém, a necessidade de atender a requisitos mais rigorosos, mesmo em ambientes incertos e desconhecidos, tem desafiado a eficácia dos métodos de controle dependentes de modelagem da planta (SARANGAPANI, 2006), já que muitas vezes o modelo não representa com exatidão a planta real, deixando de fora dinâmicas não-modeladas ou parâmetros incertos. Portanto, a utilização de algum tipo de controlador que tenha a capacidade de aprendizado se torna justificável para lidar com sistemas complexos, especialmente aqueles que são difíceis de modelar, como sistemas não lineares ou de natureza desconhecida.

Nesse contexto, ferramentas de Aprendizado de Máquina têm sido amplamente empregadas para o aprendizado e controle de fenômenos complexos. Esses controladores inteligentes utilizam redes neurais artificiais, lógica fuzzy, algoritmos genéticos ou uma combinação dessas abordagens (SARANGAPANI, 2006). Dentre elas, as redes neurais recorrentes (RNNs – *Recurrent Neural Networks*) têm se destacado, já que suas estruturas são construídas com base em princípios fundamentais de sistemas dinâmicos (BRUNTON; KUTZ, 2022). Um exemplo é apresentado no trabalho de Waegeman, Wyffels e Schrauwen (2012). Nessa estratégia, foi considerada uma RNN de tempo discreto treinada por meio da abordagem de modelo inverso, para aprender um controlador que pode ser aplicado a uma ampla variedade de plantas lineares e não-lineares. Outros exemplos de trabalhos utilizando RNNs são os apresentados por Li, Zhou e Luo (2018) e Pan e Jun Wang (2012). Li, Zhou e Luo (2018) utilizam um tipo de RNN para o controle de manipuladores redundantes em ambientes ruidosos. Já Pan e Jun Wang (2012) empregam duas RNNs para diferentes tarefas em uma abordagem de controle de modelo preditivo para sistemas dinâmicos não-lineares desconhecidos. No entanto, as RNNs geralmente são difíceis de serem treinadas devido a problemas relacionados à técnica de retro-propagação para intervalos de tempo longos, como pode ser visto no trabalho de Bengio, Simard e Frasconi (1994).

Dentre os diversos tipos de RNNs disponíveis em literatura especializada, as Redes Neurais de Estado de Eco (ESNs – *Echo State Networks*) têm se mostrado bastante adequadas

para implementações em tempo real (MAHMOUD, Tarek A.; ELSHENAWY, 2015). Isso se deve ao fato de que elas utilizam algoritmos rápidos de regressão linear para o treinamento (JAEGER, 2001). Em razão da capacidade desse rápido aprendizado, as ESNs têm chamado atenção de pesquisadores da área de controle, como evidenciado pelos resultados apresentados nos trabalhos de Waegeman, Wyffels e Schrauwen (2012) e Tarek A Mahmoud *et al.* (2021), que exploraram o uso de ESNs para controle de diferentes classes de plantas, lineares e não-lineares.

Uma das principais limitações do uso de RNNs para modelagem e controle de sistemas dinâmicos é a falta de ferramentas formais de verificação que possam oferecer garantias de estabilidade e desempenho (BARABANOV; PROKHOROV, 2002). No entanto, diversos estudos têm considerado ferramentas de controle robusto baseadas em programação semi-definida (SDP – *Semidefnite Programming*) para lidar com questões de estabilidade dessas estruturas.

Por exemplo, em *Artificial Neural Networks for Modelling and Control of Non-Linear Systems* de Suykens, Vandewalle e Moor (1995), é apresentado um framework para descrever a dinâmica das redes neurais recorrentes, abordando a análise de estabilidade usando ferramentas de controle robusto. Uma abordagem parecida é proposta por Barabanov e Prokhorov (2002). Nesse caso, o framework apresentado permite o uso de bias, ao contrário do proposto por Suykens, Vandewalle e Moor (1995), onde a análise de estabilidade é feita em casos onde não existe termo de bias. Já no trabalho de Nguyen *et al.* (2021), para garantir a estabilidade de sistemas controlados por redes neurais, os autores exploram a característica das funções de ativação mais comumente utilizadas, como a tangente hiperbólica, que são limitadas em setores. Desse modo, eles reformulam o sistema em malha fechada em termos de uma inclusão diferencial não-linear, proporcionando assim as condições necessárias para a estabilidade do sistema controlado.

No entanto, em ambos os casos, a estrutura das RNNs em cenários com dimensões elevadas podem levar a um esforço computacional grande para a busca por uma solução do problema de estabilidade, uma vez que esses métodos utilizam desigualdades matriciais lineares (LMIs), onde a dimensão das matrizes é proporcional ao tamanho das estruturas das RNNs analisadas. Além disso, a estabilidade assegurada por esses métodos se aplica a uma rede neural que já foi treinada. Em cenários em que a rede neural é utilizada para controlar um sistema de forma online, ou seja, onde seus pesos precisam ser ajustados ao longo do tempo, torna-se necessária uma nova análise de estabilidade.

Neste trabalho, é apresentada uma solução diferente para garantir a estabilidade de sistemas em malha fechada que utilizam leis de controle com aprendizado incorporadas, sem a necessidade de cálculos computacionais complexos para lidar com sistemas de realimentação de grande escala. Em particular, considera-se uma abordagem em duas malhas, que envolve a utilização de um controlador robusto na malha direta e um controlador baseado em ESN na malha externa. Essa abordagem visa atenuar os efeitos de perturbações persistentes em uma classe de sistemas não-lineares incertos, que podem ser instáveis em malha aberta.

Primeiramente, projeta-se um controlador robusto com o propósito de garantir a esta-

bilidade do sistema em malha fechada em relação ao conceito de entrada-estado estável (ISS – *Input-to-State Stable*). Essa garantia de estabilidade é independentemente da lei de controle gerada pela ESN e das influências das perturbações externas. Esse controlador robusto é desenvolvido utilizando técnicas SDP com o intuito de atenuar os efeitos das perturbações na planta. Posteriormente, é feito o treinamento offline da lei de controle baseada em ESN para aprimorar o desempenho do sistema em malha fechada, levando em consideração a propriedade ISS.

Dessa forma, por meio do controlador robusto, é possível garantir a estabilidade do sistema, enquanto a utilização de ESNs visa aprimorar o desempenho da saída do sistema, aproveitando sua capacidade de aprendizado e adaptação.

1.1 OBJETIVOS

Neste trabalho, o principal objetivo é desenvolver uma abordagem de controle robusto utilizando Redes Neurais de Estado de Eco para estabilizar sistemas não lineares incertos sujeitos a perturbações persistentes, garantindo a estabilidade entrada-estado.

Para que seja possível alcançar esse objetivo, as seguintes etapas são realizadas:

1. Projetar um controlador robusto que assegure a estabilidade de entrada-estado do sistema de malha fechada, independentemente da lei de controle gerada pela ESN e de perturbações externas.
2. Gerar uma lei de controle baseada em ESN para melhorar o desempenho do sistema em malha fechada. Essa lei de controle será treinada de forma *offline*, utilizando técnicas de aprendizado apropriadas para ajustar os parâmetros da ESN.
3. Avaliar por meio de experimentos numéricos a abordagem proposta em sistemas não-lineares incertos, analisando a mitigação dos efeitos das perturbações persistentes na saída do sistema.

1.2 CONTRIBUIÇÕES

O trabalho apresenta contribuições significativas no campo do controle de sistemas dinâmicos, incluindo:

1. Desenvolvimento de Metodologia de Controle Robusto com Aprendizado de Modelo Inverso: Foi desenvolvida uma abordagem que combina controle robusto e aprendizado de modelo inverso para sistemas dinâmicos incertos. Essa metodologia visa garantia de estabilidade e aprimoramento do desempenho de controle, mesmo diante de variações nos parâmetros do sistema.
2. Avaliação do Desempenho da Metodologia Proposta: A metodologia de controle proposta foi aplicada em sistemas dinâmicos representativos, incluindo o oscilador de Van der Pol e o sistema de uma bombas elétrica submersível. Análises dos resultados

por simulação mostram que a estratégia de controle apresenta ganhos significativos de desempenho em comparação com controladores robustos atuando sozinhos.

3. Artigo Publicado no SBAI 2023: Como resultado dessa metodologia, foi elaborado e submetido um artigo científico que foi aceito para apresentação no Simpósio Brasileiro de Automação Inteligente (SBAI 2023). O artigo concentra-se na aplicação prática do controlador desenvolvido para o controle de bombas elétricas submersíveis em poços de petróleo, com os resultados detalhados nesta dissertação.
4. Participação no CDC 2023: Além disso, a pesquisa desenvolvida neste trabalho foi aceita para apresentação no *62nd IEEE Conference on Decision and Control*, abordando o uso do controlador proposto no controle de um Oscilador de Van der Pol, cujos resultados também são apresentados neste documento.

1.3 RESUMO DA DISSERTAÇÃO

A estrutura do documento segue a organização a seguir. No Capítulo 2, são abordados os conceitos necessários para estabelecer uma base sólida para o entendimento do projeto de controle proposto neste trabalho. Inicialmente, é realizada uma revisão abrangente de Controle Robusto, apresentando os conceitos de estabilidade e estabilização de sistemas dinâmicos. Além disso, nesse capítulo, é realizada uma revisão de aprendizado de máquina, na qual são apresentados conceitos importantes, tais como Mínimos Quadrados, Redes Neurais, Redes Neurais Recorrentes e Redes Neurais de Estado de Eco.

No Capítulo 3, explora-se o projeto de controle proposto, no qual é definido o problema de controle específico abordado neste trabalho. Em seguida, são explorados os projetos de controle robusto e de aprendizado de máquina, apresentando a abordagem que combina ambos para a solução do problema de controle.

No Capítulo 4, apresentam-se os resultados de simulação do controlador aplicado em dois sistemas distintos: o sistema de Van der Pol e uma Bomba Elétrica Submersível.

Por fim, no Capítulo 5, são discutidas as conclusões do trabalho, juntamente com as perspectivas futuras para o projeto.

2 RESULTADOS INSTRUMENTAIS

Neste capítulo, será apresentada uma revisão dos principais conceitos relacionados a controle robusto e aprendizado de máquina, de forma a fornecer as bases teóricas necessárias para o entendimento da estratégia de controle proposta.

2.1 CONTROLE ROBUSTO

Controladores robustos são projetados para garantir a estabilidade e desempenho de sistemas incertos. Modelos incertos sempre existem na realidade, uma vez que não é possível modelar um sistema físico real com precisão. Porém, modelos matemáticos são necessários para o projeto de controle. Uma forma de lidar com as incertezas é adotar um modelo nominal que será utilizado para o projeto de controle, juntamente com uma margem de incertezas. Assim, obtém-se um conjunto de modelos que abrange uma gama de possíveis sistemas onde o sistema real também estará contido (LIU; YAO, 2016). Nisso se baseia o princípio do controle robusto: garantia de estabilidade e desempenho tanto para o modelo nominal quanto para a família de modelos (LIU; YAO, 2016).

A análise de estabilidade desempenha um papel fundamental no controle de sistemas dinâmicos. Nas próximas seções, serão explorados os conceitos relacionados à estabilidade e estabilização de sistemas dinâmicos tendo como base a teoria de Lyapunov, que analisa a estabilidade de pontos de equilíbrio. Este método estabeleceu não apenas a base para o controle de sistemas não-lineares, mas também se tornou um dos princípios básicos do controle robusto (LIU; YAO, 2016).

2.1.1 Estabilidade por Lyapunov

Contrastando com a estabilidade de sistemas lineares que depende apenas da dinâmica do sistema, a estabilidade de sistemas não lineares está também associada ao ponto de equilíbrio sob análise. Geralmente, a análise de estabilidade do ponto de equilíbrio é caracterizada no sentido de Lyapunov que considera a origem do espaço \mathbb{R}^n como ponto de equilíbrio. Neste caso, seja o sistema não-linear autônomo de tempo discreto representado por

$$x_+ = f(x) \quad (1)$$

com $x \in \mathbb{R}^{n_x}$ sendo o vetor das variáveis de estado e x_+ representando a próxima amostra de x , isto é, $x_+ = x(k+1)$. Nesse caso, um ponto de equilíbrio \bar{x} está localizado na origem de \mathbb{R}^{n_x} se $f(\bar{x}) = \bar{x} = 0$.

É possível, sem perda de generalidade, fazer uma troca de variáveis para transladar o ponto de equilíbrio $\bar{x} \neq 0$ para a origem do sistema. Por exemplo, considere o sistema definido em (1) com $\bar{x} \neq 0$ e seja o novo estado do sistema definido por $\xi = x - \bar{x}$. Então, o sistema transladado pode ser reescrito na seguinte forma

$$\xi_+ = x_+ - \bar{x} = f_\xi(\xi), \quad f_\xi(\xi) = f(\xi + \bar{x}) - \bar{x}, \quad (2)$$

sendo $\bar{\xi} = 0$ o ponto de equilíbrio do sistema pois $f(\bar{x}) - \bar{x} = 0$.

Definição 1 (Estabilidade por Lyapunov) (BOF; CARLI; SCHENATO, 2018) *Seja $x = 0$ um ponto de equilíbrio para o sistema (1) e $\mathcal{X} \subset \mathbb{R}^{n_x}$ um domínio que contém $x = 0$. Seja $V : \mathcal{X} \rightarrow \mathbb{R}_{>}$ sendo uma função de Lyapunov contínua e que*

$$V(0) = 0 \text{ e } V(x) > 0, \quad \forall x \in \mathcal{X} - \{0\} \text{ e}$$

$$\Delta V(x) := V(x_+) - V(x) \leq 0, \quad \forall x \in \mathcal{X}.$$

Então, o ponto de equilíbrio $x = 0$ é estável. Além do mais, se

$$\Delta V(x) < 0, \quad \forall x \in \mathcal{X} - \{0\},$$

então o ponto de equilíbrio $x = 0$ é assintoticamente estável.

A análise de estabilidade entrada-estado (*Input-to-State Stability*, ISS) é utilizada para sistemas dinâmicos sujeitos a entradas.

Lema 1 (Caracterização de Lyapunov para ISS) *Seja o sistema de tempo discreto*

$$x_+ = f(x, w), \quad x(0) = x_0 \quad (3)$$

com $f(0, 0) = 0$, $x \in \mathcal{X} \subset \mathbb{R}^{n_x}$, com \mathcal{X} sendo um domínio que contém $x = 0$ e $w \in \mathcal{W}$ onde

$$\mathcal{W} := \{w \in \mathbb{R}^{n_w} : w^T w \leq 1\}. \quad (4)$$

A origem do sistema (3) é ISS se existe uma função $V : \mathcal{X} \rightarrow \mathbb{R}_{\geq}$ e um escalar $\mu \in (0, 1)$ de forma que

$$V(x) > 0, \quad \forall x \in \mathcal{X} \text{ e}$$

$$\Delta V(x) \leq \mu(w^T w - V(x)), \quad \forall (x, w) \in (\mathcal{X} \times \mathcal{W}). \quad (5)$$

Além disso, para todo $x_0 \in \mathcal{R}$ e $w \in \mathcal{W}$, a trajetória de estado $x(k) \in \mathcal{R}$ para todo $k \geq 0$, onde

$$\mathcal{R} := \{x \in \mathcal{X} : V(x) \leq 1\} \quad (6)$$

satisfazendo $\mathcal{R} \subset \mathcal{X}$.

Prova: Suponha que $x(k) \in \mathcal{R}$ e $w(k) \in \mathcal{W}$ para um dado $k \geq 0$. Então, a variação $\Delta V(x)$, pode ser expressa como:

$$\Delta V(x) = V(x(k+1)) - V(x(k)) \leq \mu(w(k)^T w(k)) - \mu V(x(k)),$$

implicando que

$$V(x(k+1)) \leq \mu(w(k)^T w(k)) + (1 - \mu)V(x(k)).$$

Dado que $V(x(k)) \leq 1$ e $w(k)^T w(k) \leq 1$, pode-se concluir que

$$V(x(k+1)) \leq \mu + (1 - \mu) = 1 \Rightarrow x(k+1) \in \mathcal{R}.$$

Portanto, \mathcal{R} é um conjunto invariante. ■

A estimativa da região de estabilidade \mathcal{R} definida no Lema 1 é um conjunto invariante contendo as trajetórias do sistema para todo $k \geq 0$ geradas a partir de uma condição inicial $x(0) \in \mathcal{R}$ e $w \in \mathcal{W}$.

A partir da Definição 1, pode-se verificar a estabilidade por Lyapunov para sistemas lineares de tempo discreto do tipo

$$x_+ = Ax, \quad x(0) = x_0 \quad (7)$$

com $x \in \mathbb{R}^{n_x}$ e $A \in \mathbb{R}^{n_x \times n_x}$. Nesse caso, avalia-se a existência de uma matriz simétrica $P \in \mathbb{S}^{n_x} = \{M : M = M^T \in \mathbb{R}^{n_x \times n_x}\}$ que satisfaça as seguintes condições:

$$\begin{cases} V(x) = x^T P x > 0 \\ \Delta V(x) = x^T (A P A^T - P) x < 0 \end{cases} .$$

Essa condição de estabilidade pode ser expressa de maneira equivalente como uma desigualdade linear matricial (LMI, *Linear Matrix Inequality*):

$$\exists P > 0 : A P A^T - P < 0. \quad (8)$$

É importante observar que o conjunto de soluções de (8) forma um conjunto convexo (DUAN; YU, 2013) e pode ser encontrado de forma numérica utilizando pacotes computacionais projetados para a resolução de SDP.

Na presença de incertezas ou quando apenas parte dos parâmetros do modelo é conhecida, pode-se trabalhar com modelos incertos. Nesse contexto, seja a seguinte família de sistemas incertos:

$$x_+ = A(\delta(t))x, \quad \delta(t) \in \Delta \subset \mathbb{R}^{n_l}, \quad (9)$$

com $x \in \mathbb{R}^{n_x}$ e $\delta(t)$ sendo um vetor de parâmetros incertos limitados ao conjunto Δ . Quando a dependência paramétrica for afim, isto é, pode ser expressa como

$$A(\delta(t)) = A_0 + \sum_{i=1}^{n_l} \delta_i(t) A_i, \quad (10)$$

com $A_0, A_1, \dots, A_{n_l} \in \mathbb{R}^{n_x \times n_x}$ e o conjunto Δ for convexo, a estabilidade do sistema (9) pode ser determinada por meio de um conjunto finito de LMIs. Além disso, quando Δ pode ser escrito em termos de um politopo, a estabilidade do sistema (9) pode ser verificada por meio da seguinte LMI:

$$\exists P > 0 : A(\delta(t)) P A^T(\delta(t)) - P < 0, \quad \forall \delta(t) \in \mathcal{V}(\Delta). \quad (11)$$

com $P \in \mathbb{S}^{n_x}$ e $\mathcal{V}(\Delta)$ representando os vértices de Δ .

A aplicação de controle robusto pode ser estendida também para o controle de sistemas não-lineares, onde as não-linearidades presentes no modelo do sistema a ser controlado podem ser matematicamente descritas por um parâmetro variante limitado em magnitude que aparece de forma linear no modelo. Esses sistemas são chamados de sistemas lineares com parâmetros variantes, também conhecidos como sistemas LPV (*linear parameter-varying*).

Considere o seguinte sistema:

$$x_+ = A(\delta(t), \theta(x(t)))x, \quad (12)$$

com $x \in \mathcal{X} \subset \mathbb{R}^{n_x}$, $\theta \in \Theta \subset \mathbb{R}^{n_s}$ sendo um parâmetro variante no tempo e $\delta \in \Delta \subset \mathbb{R}^{n_l}$ uma incerteza paramétrica. Quando a matriz $A(\delta(t), \theta(x(t)))$ pode ser representada como uma função afim de $\theta(x(t))$ e $\delta(t)$, ou seja,

$$A(\delta(t), \theta(x(t))) = A_0 + \sum_{i=1}^{n_l} \delta_i(t)A_i + \sum_{j=1}^{n_s} \theta_j(x(t))\check{A}_j. \quad (13)$$

com $A_0, A_1, \dots, A_{n_l}, \check{A}_1, \dots, \check{A}_{n_s} \in \mathbb{R}^{n_x \times n_x}$, e os conjuntos \mathcal{X} , Δ e Θ puderem ser escritos na forma de politopos, a estabilidade do sistema LPV (12) pode ser verificada considerando a seguinte LMI:

$$\begin{aligned} \exists P > 0 : A(\delta(t), \theta(x(t)))PA^T(\delta(t), \theta(x(t))) - P < 0, \\ \forall (x, \delta(t), \theta(x(t))) \in \mathcal{V}(\mathcal{X} \times \Delta \times \Theta), \end{aligned} \quad (14)$$

com $P \in \mathbb{S}^{n_x}$.

2.1.2 Estabilização por Lyapunov

Nessa seção serão utilizados os conceitos de estabilidade por Lyapunov, discutidos anteriormente, para apresentar o projeto de controladores que asseguram a estabilidade de um sistema em malha fechada.

Seja o sistema linear discreto representado por:

$$x_+ = Ax + Bu, \quad u = Kx \quad (15)$$

com $x \in \mathbb{R}^{n_x}$ sendo o vetor de estados, $u \in \mathbb{R}^{n_u}$ como um vetor de entrada de controle, e $A \in \mathbb{R}^{n_x \times n_x}$, $B \in \mathbb{R}^{n_x \times n_u}$ e $K \in \mathbb{R}^{n_u \times n_x}$. Para um dado ganho K , estabilidade deste sistema é garantida se existir uma matriz $P \in \mathbb{S}^{n_x}$ que satisfaça as seguintes condições:

$$\begin{cases} V(x) = x^T P x > 0 \\ \Delta V(x) = x^T ((A + BK)P(A + BK)^T - P)x < 0 \end{cases} .$$

Essa condição de estabilidade pode ser reescrita na forma de uma LMI da seguinte maneira:

$$\exists P > 0, W : \begin{bmatrix} -P & AP + BW \\ PA^T + WB^T & -P \end{bmatrix} < 0.$$

com $P \in \mathbb{S}^{n_x}$, $W \in \mathbb{R}^{n_u \times n_x}$. Sendo assim, pode-se projetar uma matriz de ganhos $K = WP^{-1}$ por meio da resolução de uma LMI.

De forma semelhante, para sistemas incertos do tipo

$$x_+ = A(\delta(t))x + B(\delta(t))u. \quad (16)$$

com $x \in \mathbb{R}^{n_x}$, $u \in \mathbb{R}^{n_u}$, $A(\delta(t))$ definido em (10) e

$$B(\delta(t)) = B_0 + \sum_{i=1}^{n_l} \delta_i(t) B_i,$$

onde $B_0, B_1, \dots, B_{n_l} \in \mathbb{R}^{n_x \times n_u}$ e $\delta(t) \in \Delta \subset \mathbb{R}^{n_l}$, é possível projetar uma lei de controle $K = WP^{-1}$ que garanta a estabilidade do sistema por meio da LMI:

$$\exists P > 0, W : \begin{bmatrix} -P & A(\delta(t))P + B(\delta(t))W \\ PA(\delta(t))^T + WB(\delta(t))^T & -P \end{bmatrix} < 0, \quad \forall \delta \in \mathcal{V}(\Delta),$$

onde $P \in \mathbb{S}^{n_x}$, $W \in \mathbb{R}^{n_u \times n_x}$ e $\delta(t)$ é um vetor de parâmetros incertos limitados a um conjunto politópico Δ .

Para o projeto de controladores que garantam a estabilidade em malha fechada de sistemas sujeitos a perturbação w do tipo

$$x_+ = A(\delta(t))x + B_u(\delta(t))u + B_w(\delta(t))w, \quad \delta(t) \in \Delta, \quad (17)$$

com $x \in \mathbb{R}^{n_x}$, $u \in \mathbb{R}^{n_u}$, $w \in \mathbb{R}^{n_w}$, $A(\delta(t))$ definida em (10) e

$$B_u(\delta(t)) = B_{u0} + \sum_{i=1}^{n_l} \delta_i(t) B_{ui} \quad \text{e} \quad B_w(\delta(t)) = B_{w0} + \sum_{i=1}^{n_l} \delta_i(t) B_{wi},$$

onde $B_{u0}, B_{u1}, \dots, B_{un_l} \in \mathbb{R}^{n_x \times n_u}$ e $B_{w0}, B_{w1}, \dots, B_{wn_l} \in \mathbb{R}^{n_x \times n_w}$, pode-se utilizar o conceito de estabilização entrada-estado, derivado do Lema 1.

Definição 2 (Estabilização Entrada-Estado) *Seja o sistema (17), com $w \in \mathcal{W} = \{w \in \mathbb{R}^{n_w} : w^T w \leq 1\}$. Para um dado um escalar $\mu \in (0, 1)$, se existem as matrizes $Q > 0 \in \mathbb{S}^{n_x}$, $M \in \mathbb{R}^{n_u \times n_x}$ e $G \in \mathbb{R}^{n_x \times n_x}$ de forma que*

$$\Phi(\delta(t)) \leq 0, \quad \forall \delta \in \mathcal{V}(\Delta),$$

com

$$\Phi(\delta(t)) = \begin{bmatrix} -(1 - \mu)(Q - G - G^T) & 0 & G^T A^T(\delta(t)) + M^T B_u^T(\delta(t)) \\ 0 & -\mu I_{n_w} & B_w^T(\delta(t)) \\ A(\delta(t))G + B_u(\delta(t))M & B_w(\delta(t)) & -Q \end{bmatrix},$$

então, o sistema (17) em malha fechada é entrada-estado estável com $u = Kx = (MG^{-1})x$.

Quando os parâmetros variantes no tempo de um sistema LPV podem ser medidos em tempo real, é possível ajustar o ganho do controlador projetado de acordo com esses parâmetros. Essa abordagem permite um controle mais eficaz do sistema em comparação com o uso de controladores de ganho fixo (LIU; YAO, 2016). Esse método é conhecido como *gain scheduling*.

Considere o seguinte sistema:

$$x_+ = A(\delta(t), \theta(x(t)))x + Bu, \quad (18)$$

com $x \in \mathcal{X} \subset \mathbb{R}^{n_x}$, $u \in \mathbb{R}^{n_u}$, $\theta \in \Theta \subset \mathbb{R}^{n_s}$ sendo um parâmetro variante no tempo que pode ser medido online e $\delta \in \Delta \subset \mathbb{R}^{n_t}$ uma incerteza paramétrica. Quando a matriz $A(\delta(t), \theta(x(t)))$ pode ser representada como em (13) e os conjuntos \mathcal{X} , Θ e Δ são politópicos, pode-se estabelecer uma lei de controle do tipo $u = K(\theta)x$ de forma que o sistema (18) em malha fechada seja estável. Para isso, busca-se encontrar uma matriz $P \in \mathbb{S}^{n_x}$, $P > 0$, e as matrizes $W_0, \dots, W_{n_s} \in \mathbb{R}^{n_u \times n_x}$ de forma que

$$\begin{bmatrix} -P & A(\delta, \theta)P + BW(\theta) \\ PA(\delta, \theta)^T + W(\theta)^T B^T & -P \end{bmatrix} < 0, \quad \forall (x, \delta(t), \theta(x(t))) \in \mathcal{V}(\mathcal{X} \times \Delta \times \Theta),$$

com

$$W(\theta) = W_0 + \sum_{i=1}^{n_s} \theta_i(t)W_i.$$

Dessa forma, o ganho que estabiliza o sistema (18) é

$$K(\theta) = W(\theta)P^{-1}$$

2.2 APRENDIZADO DE MÁQUINA

Um algoritmo de aprendizado de máquina é um algoritmo que é capaz de aprender a partir de dados (GOODFELLOW; BENGIO; COURVILLE, 2016). Esses algoritmos podem ser classificados como algoritmos de aprendizado supervisionado ou não-supervisionado.

No aprendizado não-supervisionado, a saída do modelo é desconhecida ou presume-se que seja desconhecida (NELLES, 2013). Um exemplo do uso do aprendizado não-supervisionado é no agrupamento de dados que possuem características semelhantes, como ilustra a Figura 1. Nesse caso, um algoritmo de aprendizado não-supervisionado seria capaz de descobrir os quatro grupos de dados distintos mesmo sem informações sobre as saídas associadas (NELLES, 2013).

Por outro lado, nos problemas de aprendizado supervisionado, o treinamento é realizado com um conjunto de dados que contém exemplos de entrada e as respectivas saídas esperadas (BISHOP, 2006). No aprendizado supervisionado, destaca-se dois tipos principais de tarefas das quais os algoritmos podem ser projetados para executar: regressão e classificação.

Na resolução de problemas de classificação, o algoritmo procura definir uma função $f : \mathbb{R}^n \rightarrow \{1, \dots, K\}$ que mapeia os vetores de entrada v para as classes discretas correspondentes C_k , com $k = 1, \dots, K$ (BISHOP, 2006). Já em problemas de regressão, o objetivo é gerar uma saída numérica contínua para uma determinada entrada vetorial v , estabelecendo uma função $f : \mathbb{R}^n \rightarrow \mathbb{R}$.

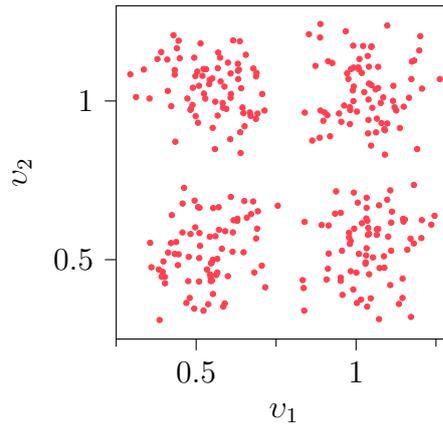


Figura 1 – Agrupamentos de dados com entradas bidimensionais. Adaptado de Nelles (2013).

2.2.1 Mínimos Quadrados

Mínimos quadrados é um algoritmo simples para a resolução de problemas de regressão linear. Ele pode ser empregado em técnicas de controle ótimo e em diferentes métodos de estimação de parâmetros e ajuste de dados (BOYD, Stephen; VANDENBERGHE, 2004).

O objetivo é construir um modelo que utilize μ amostras do sinal de entrada $v \in \mathbb{R}^{n_v}$ organizados em uma matriz $\Upsilon \in \mathbb{R}^{\mu \times n_v}$ para encontrar uma saída $\sigma = \Upsilon w$, com $\sigma \in \mathbb{R}^\mu$, que melhor se aproxime da saída real $\sigma_0 \in \mathbb{R}^\mu$. Para isso, busca-se determinar um vetor $w \in \mathbb{R}^{n_v}$ que minimize uma função de custo $E(w)$ que mede o erro quadrático.

Considerando que o erro entre a saída real σ_0 e saída predita pelo modelo σ pode ser escrito como

$$\begin{aligned} e(w) &= \sigma_0 - \sigma \\ &= \sigma_0 - \Upsilon w, \end{aligned} \quad (19)$$

o problema dos mínimos quadrados se torna

$$\min_w E(w) = \frac{1}{2} e(w)^T e(w). \quad (20)$$

Um exemplo do uso do problema de mínimos quadrados está no ajuste de uma curva à dados aleatórios, como apresentado na Figura 2. Nesse caso, são utilizadas $\mu = 20$ amostras de um conjunto $\{v^{(i)}, \sigma_0^{(i)}\}$, com $i = 1, \dots, \mu$, representadas pelos pontos na Figura 2a de forma a encontrar um modelo que minimize o erro quadrático $E(w)$ entre os pontos reais e os pontos previstos pela reta. Na Figura 2b, é destacado o valor de $w \in \mathbb{R}$ que minimiza a função de erro $E(w)$.

Será apresentada agora a solução do problema de mínimos quadrados, que se baseia no desenvolvimento apresentado por Nelles (2013). Utilizando a definição do erro de (19), a função de custo de (20) pode ser escrita como

$$\begin{aligned} E(w) &= \frac{1}{2} (\sigma_0 - \Upsilon w)^T (\sigma_0 - \Upsilon w) \\ &= \frac{1}{2} w^T H w + h^T w + h_0, \end{aligned} \quad (21)$$

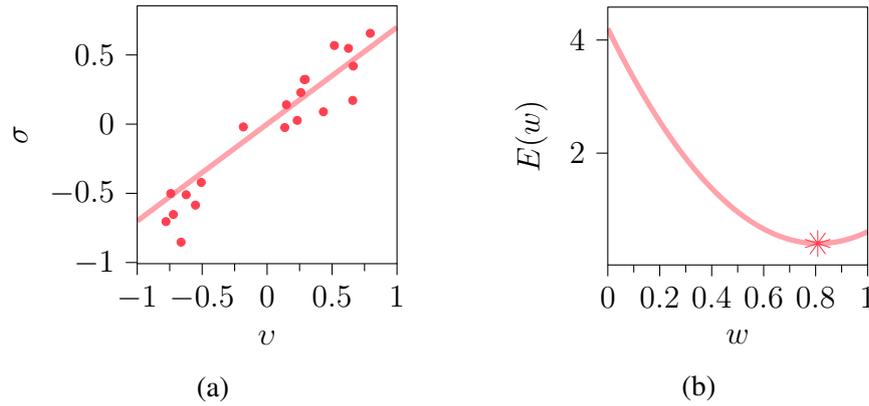


Figura 2 – Mínimos quadrados para ajuste de uma reta. Adaptado de Goodfellow, Bengio e Courville (2016).

com

$$\begin{aligned} H &= \Upsilon^T \Upsilon, \\ h &= -\Upsilon^T \sigma_0 \quad \text{e} \\ h_0 &= \sigma_0^T \sigma_0. \end{aligned}$$

Essa é uma função hiperbólica em w , cujo mínimo global pode ser obtido a partir da comparação do cálculo do gradiente de $E(w)$ em relação à w com zero, ou seja,

$$\frac{\partial E(w)}{\partial w} = -\Upsilon^T (\sigma_0 - \Upsilon w) = 0. \quad (22)$$

A partir dessa equação, a estimativa de w para a solução do problema dos mínimos quadrados é

$$\hat{w} = (\Upsilon^T \Upsilon)^{-1} \Upsilon^T \sigma_0. \quad (23)$$

A matriz $H = \Upsilon^T \Upsilon$ de (21) corresponde à matriz hessiana da função de erro $E(w)$. É importante notar que matrizes hessianas bem condicionadas estão relacionadas com a precisão das estimativas dos parâmetros w (NELLES, 2013). Por outro lado, matrizes hessianas mal condicionadas tendem a surgir em cenários como quando as dimensões das matrizes são excessivamente grandes ou quando o sistema não foi adequadamente excitado (NELLES, 2013). Quando isso acontece, é uma prática comum recorrer à técnica de regularização, a fim de obter estimativas mais precisas para os parâmetros.

Um método comum para regularização em problemas de otimização é alterar a função a ser minimizada considerando a adição de um termo dependente de w . Isso transforma o problema dos mínimos quadrados em

$$\min_w E(w) = \frac{1}{2} (e^T e + \alpha w^T w). \quad (24)$$

Esse método é chamado de *ridge regression*, que é um caso especial da regularização de Tikhonov (NELLES, 2013). A solução para o problema de mínimos quadrados com regularização é dada por

$$\hat{w} = (\Upsilon^T \Upsilon + \alpha I)^{-1} \Upsilon^T \sigma_0, \quad (25)$$

onde $I \in \mathbb{R}^{n_v \times n_v}$ representa a matriz identidade e α é um parâmetro que controla o nível de regularização aplicado ao modelo.

Na regularização de Tikhonov, ao invés da matriz identidade I em (25), utiliza-se uma matriz L . Muitas vezes essa matriz é escolhida como uma aproximação para a primeira ou segunda derivada do modelo, de forma que a regularização tende para parâmetros que representam um modelo constante (para a primeira derivada) ou um modelo linear (para a segunda derivada) (NELLES, 2013).

2.2.2 Redes Neurais

Em aprendizado de máquina, as redes neurais desempenham um papel importante no que diz respeito ao aprendizado de dados complexos. Elas são utilizadas para tarefas como processamento, aprendizado e adaptação de informações (NELLES, 2013). Suas formas mais complexas têm sido amplamente utilizadas em tarefas de visão computacional, mostrando grande desempenho em tarefas cruciais de classificação e identificação dentro desse domínio (BRUNTON; KUTZ, 2022).

De acordo com Nelles (2013), algumas das características mais significativas das redes neurais incluem a presença de um grande número de nós, que estão fortemente interconectados e operam em paralelo, além de serem altamente robustas em relação a falhas individuais em seus nós. Sua estrutura foi inicialmente inspirada na organização biológica do cérebro (NELLES, 2013), onde os nós são conhecidos como neurônios. De maneira geral, uma rede neural possui κ camadas. Cada camada ℓ possui n_ℓ neurônios, agrupados em um vetor $\xi^{(\ell)} \in \mathbb{R}^{n_\ell}$. A primeira primeira camada é conhecida como camada de entrada, $\xi^{(0)} = v$, e a última camada é conhecida como camada de saída, $\xi^{(\kappa)} = \sigma$. As demais camadas são chamadas de camadas ocultas. Conexões que interligam os neurônios de uma camada a outra são responsáveis por transmitir informações através da rede. Essas conexões são ponderadas através de uma matriz de pesos $W^{(\ell)} \in \mathbb{R}^{n_\ell \times n_{\ell-1}}$ que mapeia cada variável de uma camada $\ell - 1$ para a camada seguinte ℓ (BRUNTON; KUTZ, 2022). A Figura 3 ilustra a configuração básica de uma rede neural de três camadas.

A saída de um neurônio é determinada pela aplicação de uma função de ativação $f^{(\ell)}(\cdot)$ a uma combinação linear dos valores dos neurônios da camada anterior, juntamente com um termo constante adicional $b^{(\ell)}$, conhecido como bias. A Figura 4 apresenta um diagrama de um neurônio i da camada ℓ de uma rede neural.

O valor dos neurônios da camada ℓ de uma rede neural pode ser obtido de acordo com

$$\xi^{(\ell)} = f^{(\ell)}(z^{(\ell)}), \quad (26)$$

com

$$z^{(\ell)} = W^{(\ell)}\xi^{(\ell-1)} + b^{(\ell)}.$$

Dessa forma, a saída predita pela rede neural com κ camadas pode ser escrita em função da

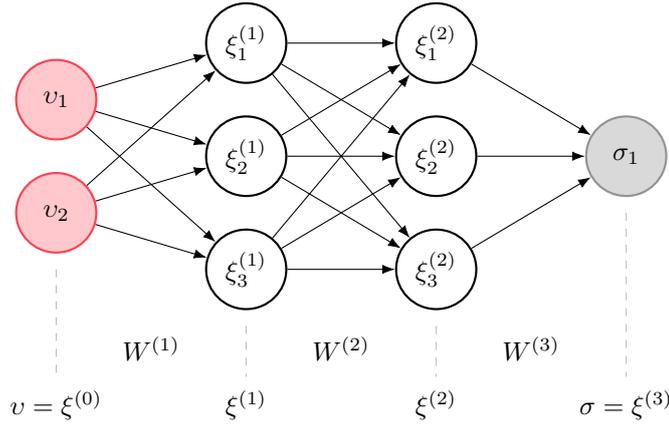


Figura 3 – Estrutura básica de uma rede neural com três camadas, onde as setas representam cada uma das conexões entre os neurônios, que por sua vez, são representados pelos círculos.

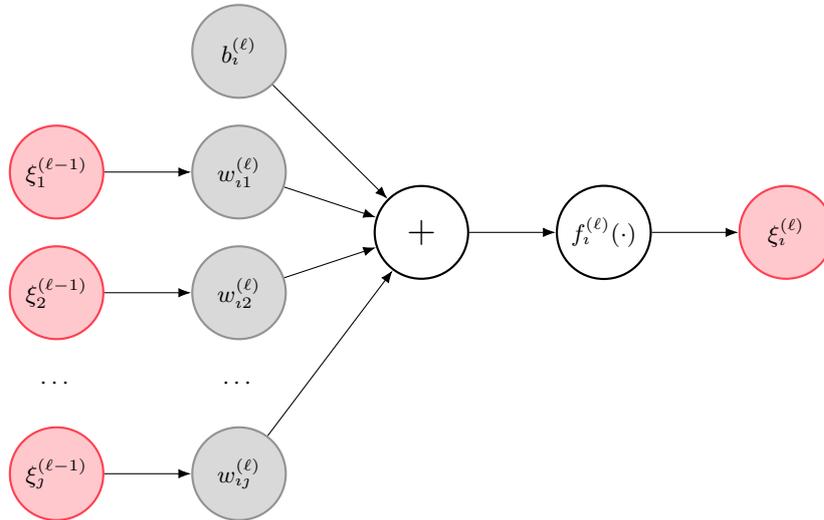


Figura 4 – Estrutura de um neurônio i de uma camada l de uma rede neural, com $w_{ij}^{(\ell)}$ sendo os elementos de $W^{(\ell)}$.

entrada v como

$$\sigma = \xi^{(\kappa)} = f^{(\kappa)}(W^{(\kappa)} f^{(\kappa-1)}(W^{(\kappa-1)} \dots f^{(1)}(W^{(1)}v + b^{(1)}) \dots + b^{(\kappa-1)}) + b^{(\kappa)}). \quad (27)$$

As funções de ativação $f^{(\ell)}(\cdot)$ são geralmente funções não-lineares. Isso representa uma vantagem significativa em relação ao uso de funções lineares, pois permite um conjunto mais diversificado de respostas (BRUNTON; KUTZ, 2022). A maioria das redes neurais modernas adota a função de ativação ReLU (*Rectified Linear Unit*) (GOODFELLOW; BENGIO; COURVILLE, 2016), apresentada na Figura 5a. No entanto, existem funções como a tangente hiperbólica (Figura 5b) e a sigmoide (Figura 5c), que também são amplamente utilizadas em diferentes contextos. As definições dessas funções estão apresentadas nas Equações (28)a, (28)b e (28)c, respectivamente.

$$f(z) = \text{ReLU}(z) = \begin{cases} 0 & \text{para } z \leq 0, \\ z & \text{para } z > 0. \end{cases}, \quad (28)\text{a}$$

$$f(z) = \tanh(z), \quad (28)\text{b}$$

$$f(z) = \text{sig}(z) = \frac{1}{1 + \exp(-z)}. \quad (28)\text{c}$$

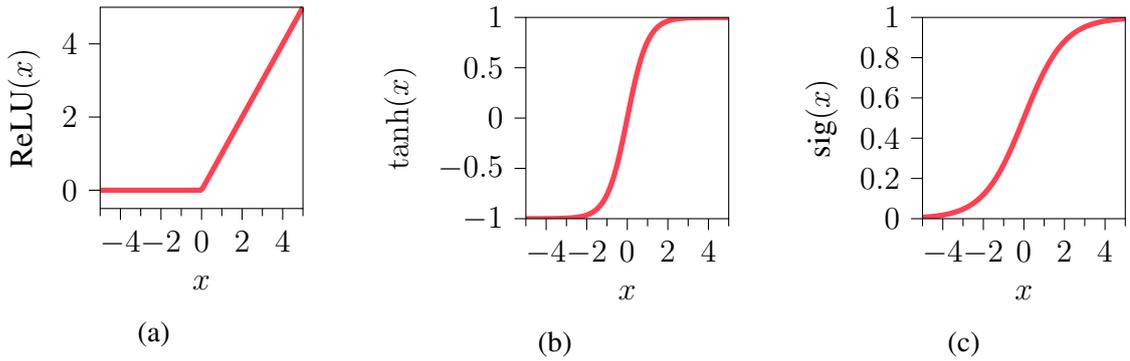


Figura 5 – Funções de Ativação.

A seguir, será apresentado o processo de treinamento de uma rede neural multicamadas para a obtenção da matriz de pesos W e do vetor de bias b . Para isso, utiliza-se um conjunto de exemplos de treinamento $\{(v^{(i)}, \sigma_0^{(i)})\}$, com $i = 1, \dots, \mu$, que contém μ amostras dos sinais de entrada e saída.

O objetivo do treinamento é ajustar W e b de todas as camadas da rede neural de forma a minimizar uma função de erro $E(W, b)$ que também é conhecida como função de perda $\mathcal{L}(\sigma_0, \sigma)$.

A maioria das técnicas de otimização para funções contínuas não-lineares envolve a escolha de um valor inicial para os pesos $W^{(\ell)}[0]$ e $b^{(\ell)}[0]$. Porém, devido à natureza altamente não-linear da função minimizada, é comum que haja mais de um ponto onde o gradiente se torna praticamente nulo (BISHOP, 2006), de forma que a escolha de $W^{(\ell)}[0]$ e $b^{(\ell)}[0]$ influencia diretamente no resultado de otimização. Após a inicialização, os valores de $W^{(\ell)}[\tau]$ e $b^{(\ell)}[\tau]$ para $\ell = 1, \dots, \kappa$ são atualizados iterativamente conforme

$$W^{(\ell)}[\tau + 1] = W^{(\ell)}[\tau] + \Delta W^{(\ell)}[\tau] \quad \text{e} \quad (29)\text{a}$$

$$b^{(\ell)}[\tau + 1] = b^{(\ell)}[\tau] + \Delta b^{(\ell)}[\tau], \quad (29)\text{b}$$

onde τ representa a etapa da iteração. Um algoritmo muito utilizado é da retropropagação (também conhecido como *backpropagation*). Este método utiliza a regra da cadeia para calcular o gradiente do erro, tirando proveito da estrutura multicamadas da rede (BRUNTON; KUTZ, 2022), avaliando a cada iteração τ o valor de $E(\sigma)$ para os novos pesos $W^{(\ell)}[\tau + 1]$ e $b^{(\ell)}[\tau + 1]$ (BISHOP, 2006). Considerando isso, a atualização dos pesos em (29) no algoritmo

de retropropagação ocorre de acordo com

$$\Delta W^{(\ell)} = -\alpha \nabla_{W^{(\ell)}} E(W, b), \quad (30a)$$

$$\Delta b^{(\ell)} = -\alpha \nabla_{b^{(\ell)}} E(W, b), \quad (30b)$$

onde α representa uma taxa de aprendizagem. Em cada etapa (iteração τ), o vetor de pesos é movido na direção da maior taxa de redução da função de erro $E(W, b)$, sendo assim, essa abordagem é conhecida como gradiente descendente (*gradient descent*) (BISHOP, 2006).

Os valores dos gradientes da Equação (30) são calculados com base na função de erro empregada e na função de ativação de cada neurônio. O método da gradiente descendente para cada iteração τ inicia-se do fim da rede neural ($\ell = \kappa$) até o início da rede neural ($\ell = 1$). De forma geral, para uma camada ℓ , considerando um exemplo de treinamento, os gradientes de (30) são obtidos por meio de

$$\nabla_{W^{(\ell)}} E(W, b) = \delta^{(\ell)} \xi^{(\ell-1)T} \quad (31a)$$

$$\nabla_{b^{(\ell)}} E(W, b) = \delta^{(\ell)} \quad (31b)$$

com

$$\delta^{(\ell)} = \begin{cases} \nabla_{\sigma} \mathcal{L}(\sigma_0, \sigma) \circ (f^{(\ell)})'(z^{(\ell)}) & \ell = \kappa, \\ (W^{(\ell+1)T} \delta^{(\ell+1)}) \circ f'^{(\ell)}(z^{(\ell)}) & 1 \leq \ell < \kappa, \end{cases} \quad (32)$$

onde \circ representa o produto elemento a elemento e $(f^{(\ell)})'(z^{(\ell)})$ denota a derivada elemento a elemento em respeito a $(z^{(\ell)})$ (STANFORD UNIVERSITY, 2019). A função de perda $\mathcal{L}(\sigma_0, \sigma) = E(W, b)$ representa a medida de erro. Em problemas de regressão, é comum utilizar o erro quadrático, definido como

$$\mathcal{L}(\sigma_0, \sigma) = \frac{1}{2}(\sigma - \sigma_0)^2.$$

2.2.3 Redes Neurais Recorrentes

Redes Neurais Recorrentes (RNN) são uma classe importante de arquitetura de redes neurais, onde suas estruturas são projetadas com base em princípios fundamentais de sistemas dinâmicos (BRUNTON; KUTZ, 2022). Normalmente, as RNNs utilizam uma arquitetura multicamadas. No entanto, a característica difere esse tipo de rede neural de outras é que o estado atual dos neurônios em uma RNN é influenciado pelo estado anterior desses neurônios. Por isso, elas são ideais para lidar, por exemplo, com fluxos de dados sequenciais, como reconhecimento de fala (BRUNTON; KUTZ, 2022) e com a identificação de sistemas dinâmicos (NELLES, 2013).

As RNNs podem ser configuradas de diversas maneiras ao incorporar diferentes tipos de funções de recorrência no modelo (GOODFELLOW; BENGIO; COURVILLE, 2016). Uma forma de descrever as redes neurais recorrentes, conforme Goodfellow, Bengio e Courville (2016), é considerando a equação no espaço de estados de tempo discreto

$$\begin{cases} \xi_+ = f_{\xi}(W_h^h \xi + W_v^h v + b) \\ \sigma = f_{\sigma}(W_h^{\sigma} \xi_+ + c) \end{cases}, \quad (33)$$

onde os pesos $W_h^h \in \mathbb{R}^{n_\xi \times n_\xi}$, $W_v^h \in \mathbb{R}^{n_\xi \times n_v}$ e $W_h^\sigma \in \mathbb{R}^{n_\sigma \times n_\xi}$ representam, respectivamente, as ligações entre os neurônios ocultos, as ligações entre as entradas e os neurônios ocultos e as ligações entre os neurônios ocultos e a saída. Além disso, nesse caso, b e c são vetores de bias e as funções de ativação $f_\xi(\cdot)$ e $f_\sigma(\cdot)$ são as funções para a camada oculta e camada de saída, respectivamente.

O treinamento das redes neurais recorrentes ocorrem de forma similar ao explicado na seção anterior. Nesse caso, utiliza-se o algoritmo da retropropagação no tempo, que é uma extensão do algoritmo de retropropagação padrão (NELLES, 2013).

2.2.4 Redes Neurais de Estado de Eco

Redes Neurais de Estado de Eco (*Echo State Networks*) são um tipo de RNNs de rápida aprendizagem. Em certas condições, o estado ξ_+ de uma RNN é uma função das amostras passadas de entrada, ou seja, $\xi_+ = \xi(k+1) = g(v(k), v(k-1), v(k-2), \dots)$. Essa característica pode ser comparada a um 'eco' do histórico de entrada (JAEGER, 2001), o que deu origem ao nome desse tipo de rede neural.

As ESNs são formadas por uma camada de entrada, uma camada interna e uma camada de saída (JAEGER, 2002). A camada interna possui um grande número de neurônios que são esparsamente conectados, com tipicamente 1% de conectividade (BRUNTON; KUTZ, 2022), e é chamada de reservatório. A Figura 6 apresenta uma ilustração de uma ESN.

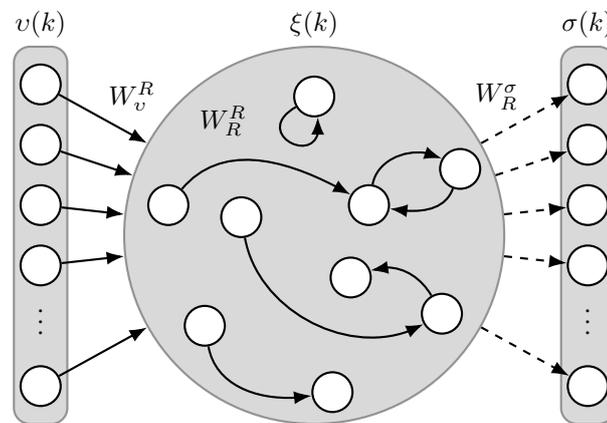


Figura 6 – Ilustração de uma Rede Neural de Estado de Eco.

O treinamento de RNNs tradicionalmente envolve o uso do algoritmo do gradiente descendente em relação a uma função de erro. No entanto, esse método apresenta limitações como ótimos locais e convergência lenta, como mencionado por Jaeger (2001). Por outro lado, as ESNs trabalham com uma abordagem diferente, onde apenas os pesos de saída são treinados utilizando métodos simples de resolução de problemas de regressão linear (SUN *et al.*, 2022), como mínimos quadrados. Para isso, os pesos das conexões na camada de entrada e no reservatório são mantidos fixos após a inicialização.

A dinâmica de uma ESN pode ser descrita na seguinte representação no espaço de estados de tempo discreto:

$$\begin{cases} \xi_+ = (1 - \gamma)\xi + \gamma f_\xi(W_R^R \xi + W_v^R v + b) \\ \sigma = W_R^\sigma \xi \end{cases} \quad (34)$$

onde $\xi \in \mathbb{R}^{n_\xi}$ é o vetor de estados da ESN que corresponde aos neurônios do reservatório, $v \in \mathbb{R}^{n_v}$ é a entrada da ESN, $\sigma \in \mathbb{R}^{n_\sigma}$ é a saída da ESN, f_ξ é a função de ativação, tipicamente $\tanh(\cdot)$ elemento a elemento, $\gamma \in (0, 1)$ é a taxa de escoamento, e W_R^R , W_v^R e W_R^σ são as matrizes de pesos reservatório-reservatório, entrada-reservatório e reservatório-saída, respectivamente, e b representa o termo de bias.

Os pesos das conexões que vão para o reservatório são inicializadas de forma aleatória e permanecem fixas, em geral seguindo os seguintes passos:

1. As matrizes $\overline{W}_R^R \in \mathbb{R}^{n_\xi \times n_\xi}$, $W_v^R \in \mathbb{R}^{n_\xi \times n_v}$ e o vetor de bias b são gerados de forma aleatória seguindo uma distribuição normal $\mathcal{N}(0, 1)$.
2. A matriz W_R^R é obtida ao reescalar \overline{W}_R^R de tal forma que seu raio espectral seja menor que 1, de forma a garantir a estabilidade interna. Isso é:

$$W_R^R = \frac{\rho_R}{\lambda_{\max}} \overline{W}_R^R, \quad (35)$$

onde $\rho_R \in (0, 1)$ é o raio espectral desejado e λ_{\max} é o maior valor singular de \overline{W}_R^R .

3. W_v^R e b são multiplicadas pelos escalares ρ_v e ρ_{bias} , respectivamente.

Os parâmetros ρ_R , ρ_v e ρ_{bias} são essenciais para a performance de aprendizagem da rede neural e influenciam diretamente a capacidade de memória e representação não-linear do reservatório. Além disso, valores de taxa de escoamento γ menores tendem a aumentar a capacidade de memória do reservatório, enquanto valores mais altos favorecem entradas e saídas que variam rapidamente no tempo.

Considerando uma sequência grande o suficiente de entradas e saídas coletadas do sistema, a matriz de pesos W_R^σ do reservatório para a saída é treinada por meio da solução de um problema de mínimos quadrados. Para treinar a ESN, os dados de entrada $v(k)$ são organizados em uma matriz Υ e a saída desejada $\sigma(k)$ em um vetor Σ . Cada linha v^T da matriz Υ corresponde a uma amostra k e as colunas estão relacionadas com cada entrada da ESN. Para simplificar, considera-se que há várias entradas e apenas uma saída. As linhas de Υ representam as entradas do reservatório para cada instante de tempo, gerando uma matriz de estados Ξ onde as linhas correspondem à sequência de estados resultantes.

Considerando a Regressão de Ridge, o vetor do peso reservatório-saída W_R^σ é obtido resolvendo o seguinte sistema linear:

$$(\Xi^T \Xi - \lambda I) W_R^\sigma = \Xi^T \Sigma \quad (36)$$

onde λ é o parâmetro de regularização de Tikhonov que é utilizado para penalizar a magnitude do peso de forma a evitar o sobre-ajuste. No caso de saídas múltiplas, o vetor de pesos $W_R^{\sigma_i}$ para

cada saída σ_i é calculado utilizando a mesma equação de (36), porém utilizando o vetor Σ_i com as saídas desejadas.

3 PROJETO DE CONTROLE

Os principais resultados deste trabalho são apresentados nesse capítulo. O objetivo do trabalho é regular a saída controlada y em relação a uma referência desejada r , considerando a minimização dos efeitos de uma perturbação d com magnitude limitada. Para isso, a entrada de controle u será determinada através da combinação de uma lei de controle robusta u_1 com um termo de correção u_2 , gerado por uma rede neural. Essa abordagem de controle é representada na Figura 7. Nela, é apresentado um diagrama simplificado do controlador proposto, cuja estrutura detalhada será abordada posteriormente.

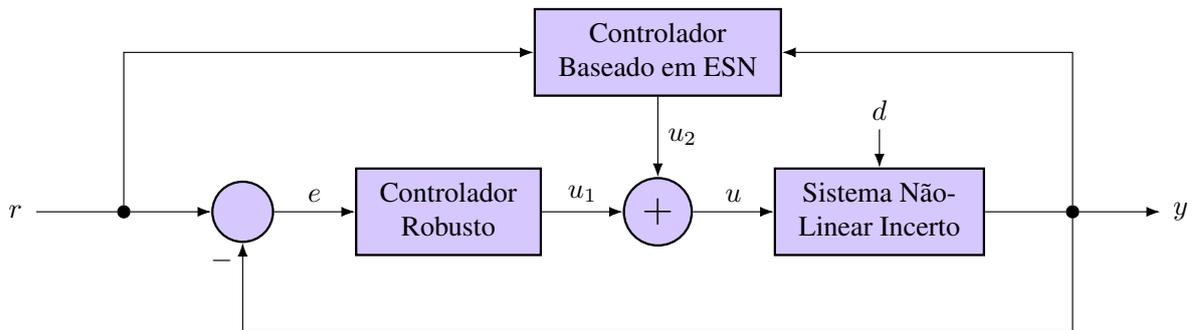


Figura 7 – Diagrama simplificado do Controlador Proposto.

Primeiramente será apresentada a definição do problema de controle e a classe de sistemas no qual ele pode ser empregado. Posteriormente, será abordado o projeto de um controlador robusto considerando uma realimentação de estados que estabilize o sistema. Por fim, será explorado o projeto de um controlador baseado em uma Rede Neural de Estado de Eco, cuja estratégia de controle baseia-se na obtenção da dinâmica inversa da planta.

3.1 DEFINIÇÃO DO PROBLEMA DE CONTROLE

O controlador proposto é projetado para o controle de uma classe \mathcal{G} de sistemas não-lineares de tempo discreto. Essa classe pode ser definida como

$$\mathcal{G} : \begin{cases} x_+ = f(x, \delta, \theta, u, d) \\ y = Cx \end{cases}, \quad (37)$$

onde $x \in \mathcal{X} \subset \mathbb{R}^{n_x}$ é o vetor de estados, $\delta \in \Delta \subset \mathbb{R}^{n_\delta}$ é um vetor de parâmetros variantes no tempo, $\theta \in \Theta \subset \mathbb{R}^{n_\theta}$ é um vetor de parâmetros variantes no tempo que podem ser medidos de forma online, $u \in \mathbb{R}^{n_u}$ é a entrada de controle, $d \in \mathcal{D} \subset \mathbb{R}^{n_d}$ é a entrada de perturbação, $y \in \mathbb{R}^{n_y}$ é a saída controlada, $f(\cdot)$ é uma função vetorial com dependência polinomial em relação a x e linear em respeito a (δ, θ, u, d) , e \mathcal{X} , Δ , Θ e \mathcal{D} são conjuntos compactos.

Em relação ao sistema (37), assume-se que:

A1) $f(0, \delta, \theta, 0, 0) = 0$ para todo $(\delta, \theta) \in \Delta \times \Theta$.

A2) \mathcal{X} é um conjunto convexo, com $0 \in \mathcal{X}$, que pode ser representado em termos do envoltório convexo de seus n_v vértices, *i.e.*,

$$\mathcal{X} := \text{Co}\{v_1, \dots, v_{n_v}\},$$

ou, alternativamente, como a interseção de n_h semi-planos

$$\mathcal{X} := \{x \in \mathbb{R}^{n_x} : h_i^T x \leq 1, i = 1, \dots, n_h\}$$

com $v_i \in \mathbb{R}^{n_x}$, $i = 1, \dots, n_v$, e $h_j \in \mathbb{R}^{n_x}$, $j = 1, \dots, n_h$, definindo respectivamente os vértices e faces de \mathcal{X} .

A3) Δ e Θ são conjuntos politópicos com vértices conhecidos.

A4) \mathcal{D} é um domínio limitado em magnitude dado por

$$\mathcal{D} := \{d \in \mathbb{R}^{n_d} : d^T d \leq \eta_d^2/2\} \quad (38)$$

com $\eta_d > 0$ definindo o tamanho de \mathcal{D} .

Neste trabalho, será abordada a regulação da saída controlada y em torno de uma referência desejada r , considerando a minimização dos efeitos da perturbação d . Com esse objetivo, a entrada de controle u é obtida através da combinação de uma lei de controle robusta u_1 e um termo de correção u_2 gerado pela ESN.

O sinal u_1 do controlador robusto é projetado para garantir que o sistema de malha fechada seja entrada-estado estável, independentemente da perturbação persistente d e do sinal de controle u_2 , supondo que ambos sinais possuem magnitude limitada com limites conhecidos. O sinal de correção $u_2 \in \mathcal{U}$ é obtido por meio da ESN de forma a minimizar os efeitos de d na saída y . Além disso, essa configuração de controle assegura a propriedade de estabilidade ISS para o sistema em malha fechada.

Por simplicidade, será considerado nesse trabalho que o problema de regulação do sistema em torno de uma referência nula, ou seja, $r \equiv 0$. Além disso, assume-se também que

A5) u_2 é restrito ao conjunto

$$\mathcal{U} := \{u_2 \in \mathbb{R}^{n_u} : u_2^T u_2 \leq \eta_u^2/2\}, \quad (39)$$

com $\eta_u > 0$ definindo o tamanho de \mathcal{U} .

Será considerado que o sinal de controle robusto é uma lei de controle de realimentação de estados polinomial conforme

$$u_1 = K(x)x, \quad (40)$$

com $K : \mathbb{R}^{n_x} \rightarrow \mathbb{R}^{n_u \times n_x}$ sendo uma função polinomial matricial a ser determinada.

Dessa forma, os problemas a serem abordados neste trabalho são:

- Problema de Controle Robusto: projetar $K(x)$ de forma que o sistema em malha fechada representado em (37) com $u = u_1 + u_2$ seja ISS para todo $x(0) \in \mathcal{R} \subset \mathcal{X}$, $u_2 \in \mathcal{U}$ e $d \in \mathcal{D}$ enquanto minimiza $\|x\|$, com \mathcal{R} representando uma estimativa superior do conjunto de atração.

- Problema de Controle Baseado em ESN: para uma dada lei de controle robusta u_1 , treinar a ESN de forma que a saída $\|y\|$ seja minimizada.

Nas seções seguintes, será apresentado o projeto proposto por esse trabalho para solucionar cada um desses problemas.

3.2 PROJETO DE CONTROLE ROBUSTO

O sistema \mathcal{G} em (37) pode ser reescrito, sem perda de generalidade, da seguinte forma

$$\mathcal{G} : \begin{cases} x_+ = A(x, \delta, \theta)x + B_u(\delta)(u_1 + u_2) + B_d(\delta)d \\ y = Cx \end{cases} \quad (41)$$

partindo do fato de que $f(x, \delta, \theta, u, d)$ é uma função polinomial em respeito a x e linear em respeito a (δ, θ, u, d) , onde $A(\cdot) \in \mathbb{R}^{n_x \times n_x}$ é uma função polinomial matricial de x e afim em respeito a δ e θ , e $B_u(\cdot) \in \mathbb{R}^{n_x \times n_u}$ e $B_d(\cdot) \in \mathbb{R}^{n_x \times n_d}$ são funções matriciais afins de δ .

A fim de obter uma solução convexa para o problema de controle robusto, seja q o maior grau dos polinômios de $A(x, \theta)$ e as seguintes definições:

$$B_w(\delta) = \begin{bmatrix} \eta_u B_u(\delta) & \eta_d B_d(\delta) \end{bmatrix}, \quad w = \begin{bmatrix} u_2 \\ d \end{bmatrix}, \quad n_w = n_u + n_d \quad (42)$$

$$A(x, \delta, \theta) = A_0(\delta) + \Pi(x, \theta)^T A_1(\delta), \quad (43)$$

onde $A_0(\delta) \in \mathbb{R}^{n_x \times n_x}$ e $A_1(\delta) \in \mathbb{R}^{q n_x \times n_x}$ são funções matriciais afins em δ e θ , e δ , respectivamente, e

$$\Pi(x, \theta) = \begin{bmatrix} m^{(1)}(x, \theta) \otimes I_{n_x} \\ \vdots \\ m^{(q)}(x, \theta) \otimes I_{n_x} \end{bmatrix} \quad (44)$$

com $m^{(l)}(\cdot) \in \mathbb{R}^{n_l}$, $l = 1, \dots, q$, representando o vetor formado por todos os monômios de grau l . Observa-se que, considerando (43) e (44), existem funções matriciais afins em x , i.e., $\Omega_0(x, \theta) \in \mathbb{R}^{n_m n_x \times n_x}$ e $\Omega_1(x, \theta) \in \mathbb{R}^{n_m n_x \times n_m n_x}$ de tal forma que o seguinte é válido:

$$\Omega_0(x, \theta) + \Omega_1(x, \theta)\Pi(x, \theta) = 0_{n_m n_x \times n_x} \quad (45)$$

$$\det\{\Omega_1(x, \theta)\} = c, \quad c \neq 0, \quad \forall x \in \mathbb{R}^{n_x} \quad (46)$$

com c sendo uma dada constante real e

$$n_m = n_1 + \dots + n_q. \quad (47)$$

Além disso, o ganho de controle $K(x)$ de (40) é definido como

$$K(x, \theta) = K_0 + K_1 \Pi(x, \theta) \quad (48)$$

onde $\Pi(x, \theta)$ é como apresentado em (44), e $K_0 \in \mathbb{R}^{n_U \times n_x}$ e $K_1 \in \mathbb{R}^{n_U \times n_m n_x}$ são matrizes a serem determinadas.

Dessa forma, considerando (41), (42) e (43), a dinâmica do sistema em malha fechada é

$$x_+ = \left(A_0(\delta) + \Pi(x, \theta)^T A_1(\delta) + B_u(\delta)(K_0 + K_1 \Pi(x, \theta)) \right) x + B_w(\delta) w \quad (49)$$

O seguinte resultado apresenta condições suficientes para o projeto dos ganhos K_0 e K_1 de forma que o sistema definido acima é ISS para todo $w \in \mathcal{W}$ em termos de um conjunto finito de restrições de LMIs.

Teorema 1 *Considere o sistema definido em (37) satisfazendo (A1)-(A4), com sua dinâmica de malha fechada em (49), com $\Pi(x, \theta)$ satisfazendo (45) e (46). Seja $\mu \in (0, 1)$ um dado escalar real. Suponha que existam $Q = Q^T$, G , M_0 , M_1 e L sendo matrizes reais, com dimensões apropriadas, que satisfazem as seguintes LMIs:*

$$\begin{bmatrix} 1 & h_i^T Q \\ Q h_i & Q \end{bmatrix} > 0, \quad i = 1, \dots, n_h \quad (50)$$

$$\Lambda(\delta, \theta) + L\Omega(x, \theta) + \Omega(x, \theta)^T L^T < 0, \quad \forall (x, \delta, \theta) \in \mathcal{V}\{\mathcal{X} \times \Delta \times \Theta\} \quad (51)$$

com

$$\Lambda(\delta, \theta) = \begin{bmatrix} (1-\mu)(Q-G-G^T) & \star & \star & \star & \star \\ 0 & 0 & \star & \star & \star \\ 0 & 0 & -\mu I_{n_w} & \star & \star \\ A_0(\delta)G+B_u(\delta)M_0 & B_u(\delta)M_1 & B_w(\delta) & -Q & \star \\ A_1(\delta)G & 0 & 0 & 0 & 0 \end{bmatrix}$$

onde $\mathcal{V}\{\mathcal{X} \times \Delta \times \Theta\}$ representa o conjunto de todos os vértices de $\mathcal{X} \times \Delta \times \Theta$ e

$$\Omega(x, \theta) = \begin{bmatrix} \Omega_0(x, \theta) & \Omega_1(x, \theta) & 0 & 0 & 0 \\ 0 & 0 & 0 & \Omega_0(x, \theta) & \Omega_1(x, \theta) \end{bmatrix}.$$

Dessa forma, o sistema em malha fechada de (49) com

$$K_0 = M_0 G^{-1}, \quad K_1 = M_1 G_a^{-1}, \quad G_a = \text{diag}\{G, \dots, G\}, \quad (52)$$

é localmente ISS estável. Além disso, para todo $x(0) \in \mathcal{R}$ e $w(k) \in \mathcal{W}$, a trajetória dos estados $x(k)$ permanece em \mathcal{R} , $\forall k > 0$, com

$$\mathcal{R} = \{x \in \mathbb{R}^{n_x} : x^T Q^{-1} x \leq 1\}. \quad (53)$$

Prova: em primeiro lugar, pode-se verificar que a partir da Equação (50), $Q > 0$. Além disso, ao aplicar o complemento de Schur (Lema 2), pode-se concluir que $1 - h_i^T Q h_i > 0$ para $i = 1, \dots, n_h$. Isso implica que $V(x) = x^T Q^{-1} x$, é maior que zero para todos os valores de x diferentes de zero, e que a região de atração \mathcal{R} definida em (53), está contida em \mathcal{X} .

Levando em consideração o conjunto de LMIs apresentado no Teorema 1, segue que (51) vale para para $(x, \delta, \theta) \in \mathcal{X} \times \Delta \times \Theta$ por meio de argumentos de convexidade.

Seja

$$\Pi_a = \begin{bmatrix} I_{n_x} & 0 & 0 \\ \Pi(x, \theta) & 0 & 0 \\ 0 & I_{n_w} & 0 \\ 0 & 0 & I_{n_x} \\ 0 & 0 & \Pi(x, \theta) \end{bmatrix},$$

têm-se que $\Omega(x, \theta)\Pi_a = 0$. Então, pré-multiplicando e pós-multiplicando a desigualdade matricial de (51) por Π_a^T e Π_a , respectivamente, obtêm-se

$$\Pi_a^T \Lambda(\delta, \theta) \Pi_a = \Phi(x, \theta) < 0, \quad \forall (x, \theta) \in \mathcal{X} \times \Theta, \quad (54)$$

onde

$$\Phi(x, \delta, \theta) = \begin{bmatrix} (1 - \mu)(Q - G - G^T) & \star & \star \\ 0 & -\mu I_{n_w} & \star \\ \left(\begin{array}{c} (A_0(\delta) + \Pi^T A_1(\delta))G + \\ B_u(\delta)(M_0 + M_1 \Pi(x, \theta)) \end{array} \right) & B_w(\delta) & -Q \end{bmatrix}.$$

Do bloco (1, 1) da matriz $\Phi(x, \delta, \theta)$, têm-se que $Q < G + G^T$, o que implica que G é uma matriz de posto completo desde que $Q > 0$. Sejam as parametrizações $M_0 = K_0 G$ e $M_1 = K_1 G_a$. Portanto, considerando o fato de que $G_a \Pi(x, \theta) = \Pi(x, \theta) G$ e que

$$-G^T Q^{-1} G \leq Q - G - G^T,$$

pode-se, a partir de (54), escrever que

$$G_d^T \Psi_a(x, \delta, \theta) G_d < 0, \quad \forall (x, \delta, \theta) \in \mathcal{X} \times \Delta \times \Theta \quad (55)$$

com $G_d = \text{diag}\{G, I_{n_w}, I_{n_x}\}$ e

$$\Psi_a(x, \delta, \theta) = \begin{bmatrix} \Gamma & \Psi^T(x, \delta, \theta) \\ \Psi(x, \delta, \theta) & -Q \end{bmatrix},$$

onde

$$\Gamma = \begin{bmatrix} (1 - \mu)Q^{-1} & 0 \\ 0 & -\mu I_{n_w} \end{bmatrix}$$

e

$$\Psi(x, \delta, \theta) = \begin{bmatrix} \left(\begin{array}{c} A_0(\delta) + \Pi^T A_1(\delta) + \\ B_u(\delta)(K_0 + K_1 \Pi(x, \theta)) \end{array} \right) & B_w(\delta) \end{bmatrix}.$$

De (55), têm-se que $\Psi_a(x, \delta, \theta) < 0$. Aplicando o complemento de Schur nesta desigualdade, obtêm-se que

$$\Gamma + \Psi(x, \delta, \theta) Q^{-1} \Psi^T(x, \delta, \theta) < 0.$$

Então, pré- e pós-multiplicando a desigualdade matricial resultante por $\begin{bmatrix} x^T & w^T \end{bmatrix}$ e sua transposta, respectivamente, chega-se em

$$\begin{aligned} x_+^T Q^{-1} x_+ - (1 - \mu)x^T Q^{-1} x - \mu w^T w < 0 &\Rightarrow \\ \Delta V \leq \mu(w^T w - V) < 0, \forall (x, \theta) \in \mathcal{X} \times \Theta, &\quad (56) \end{aligned}$$

onde $V = x^T Q^{-1} x$. O restante da prova é derivado do Lema 1. ■

A lei de controle robusta $u_1 = K(x)x$ pode ser projetada enquanto minimiza-se $\|x(k)\|$ ao considerar o seguinte problema de otimização

$$\min_{\lambda, Q, \dots, L} \lambda : \begin{cases} \lambda I_{n_x} - Q \geq 0, \\ (50) \text{ e } (51). \end{cases} \quad (57)$$

Prova: a matriz simétrica $Q > 0$ pode ser decomposta da seguinte maneira:

$$Q = M \Lambda M^T$$

sendo M uma matriz composta por autovetores de Q e Λ uma matriz diagonal com os autovalores correspondentes. Se λ representa o maior autovalor de Q , então

$$Q = M \Lambda M^T \leq M(\lambda I)M^T = \lambda(MM^T) = \lambda I,$$

o que implica que $Q \leq \lambda I$. Considerando $V(x) = x^T Q^{-1} x \leq 1$, pode-se então escrever

$$\lambda^{-1} x^T x \leq 1,$$

pois $\lambda^{-1} I \leq Q^{-1}$. Dessa forma,

$$\|x\|^2 \leq \lambda \Rightarrow \|x\| \leq \sqrt{\lambda}.$$

Assim, ao minimizar o maior autovalor λ da matriz Q , minimiza-se também $\|x\|$. ■

3.3 PROJETO DE APRENDIZADO DE MÁQUINA

Diversos algoritmos de controle clássicos utilizam conhecimento prévio de um sistema para alcançar o comportamento desejado, assim como acontece com o controlador robusto apresentado anteriormente. No entanto, sistemas complexos não-lineares podem não ser totalmente conhecidos ou modelados corretamente, o que justifica o uso de abordagens de aprendizado, como as redes neurais.

Neste trabalho, o uso de redes neurais é considerado para a minimização dos efeitos da perturbação na saída do sistema a ser controlado. Para isso, considera-se o aprendizado de um controlador baseado no modelo inverso de uma planta, que utiliza a saída desejada do sistema para calcular a entrada que leve a essa saída. Em outras palavras, esse controlador “inverte” a relação entre saídas e entradas do sistema.

Para criar o modelo inverso, será utilizada uma Rede Neural de Estado de Eco, de forma similar à explorada no trabalho de Jaeger (2008). Essa rede neural possui características desejáveis (WAEGERMAN; WYFFELS; SCHRAUWEN, 2012; JORDANOU; ANTONELLO; CAMPONOGARA, 2019), como treinamento fácil e rápido. Além disso, por ser uma rede neural recorrente, as ESNs possuem uma memória interna, o que permite que ela mantenha um contexto ao longo do tempo, tornando-a mais adequada para lidar com dependências temporais.

Para aprender o modelo inverso da planta, este trabalho considera a configuração mostrada na Figura 8. A entrada $v = (\tilde{y}, \tilde{u}_1, \tilde{u}_2)$ da ESN coleta as saídas passadas da planta $y[\cdot]$ além dos sinais de controle $u_1[\cdot]$ e $u_2[\cdot]$, espaçados no tempo de acordo com um parâmetro de atraso $\delta \in \mathbb{N}$. Ou seja, $\tilde{y} = (y[k], y[k - \delta], \dots, y[k - m\delta])$ consiste na amostra da saída atual do sistema e m amostras de saída passada, além de m amostras passadas das entradas de controle $\tilde{u}_1 = (u_1[k - \delta], \dots, u_1[k - m\delta])$ e $\tilde{u}_2 = (u_2[k - 2\delta], \dots, u_2[k - (m + 1)\delta])$. Dado o vetor de entrada da rede $v[k]$ no tempo k , o objetivo do modelo inverso é aprender a entrada de controle da planta $u_2[k - \delta]$, no tempo $k - \delta$, que levou o sistema à saída atual $y[k]$. O número de saídas passadas m e o atraso δ são hiperparâmetros a serem ajustados adequadamente.

Ao simular a planta com u_1 sendo o sinal do controlador robusto e u_2 sendo um sinal aleatório dentro dos limites de estabilidade definido em (39), obtêm-se uma matriz Ξ com a trajetória dos estados do reservatório e um vetor Σ com as saídas desejadas. Em seguida, a regressão ridge é aplicada para treinar o modelo inverso usando a Eq. (36), resultando na matriz de reservatório para saída W_R^σ .

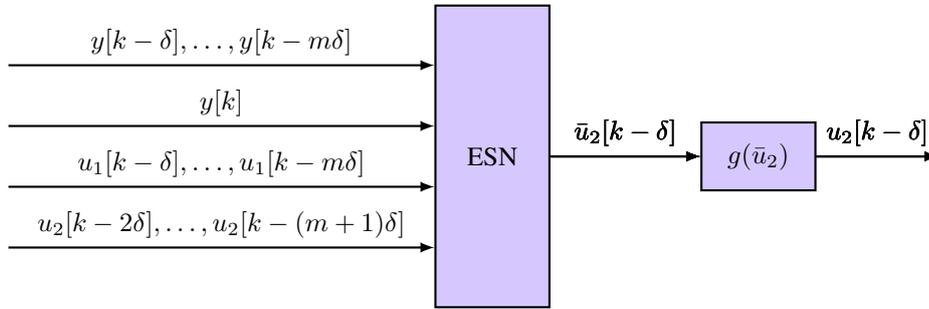


Figura 8 – Configuração da ESN para treinamento do modelo inverso.

Uma vez que a matriz W_R^σ tenha sido aprendida para o modelo inverso, a ESN resultante pode ser utilizada para gerar o sinal de controle u_2 que melhora o desempenho, mitigando o efeito da perturbação na saída do sistema. Isso é alcançado pelo deslocamento de tempo de δ passos em todos os sinais da configuração utilizada para o treinamento (Figura 8), o que leva à configuração para o controle com o modelo inverso, conforme mostrado na Figura 9. No tempo k , ao definir $y[k + \delta]$ igual à referência desejada no tempo $k + \delta$, a ESN fornecerá o sinal de controle $\bar{u}_2[k]$ que deve ser aplicado no momento atual para levar a planta à referência desejada.

Antes de utilizar o sinal obtido pela ESN no sistema, ele é ativado por uma função

$$g(\bar{u}_2) = \sqrt{1/2}\eta_u \tanh(\bar{u}_2)$$

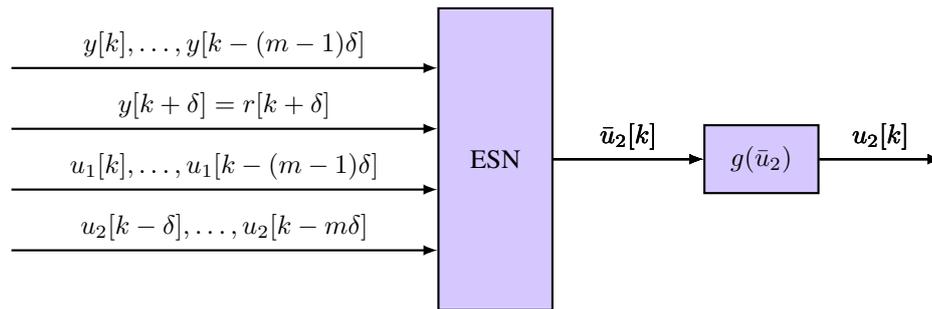


Figura 9 – Configuração da ESN para controle utilizando modelo inverso.

de forma a garantir que o sinal gerado esteja dentro dos limites estabelecidos no projeto de controle robusto, ou seja, $u_2 \in \mathcal{U}$.

4 RESULTADOS NUMÉRICOS

Neste capítulo, será apresentada a aplicação do controlador proposto em dois sistemas com diferentes características: o oscilador de Van der Pol, que em malha aberta consiste em um sistema instável, onde é considerado um parâmetro δ incerto e não mensurável; e o sistema de uma bomba elétrica submersível utilizado para elevação de fluidos em um poço de petróleo, que conta com parâmetros θ incertos mas mensuráveis.

4.1 VAN DER POL

Considere o seguinte sistema, que consiste na equação de Van der Pol com a adição de um integrador:

$$\begin{cases} \dot{x}_1(t) = x_2(t) \\ \dot{x}_2(t) = -x_1(t) + \delta(1 - x_1(t)^2)x_2(t) + u(t) + d(t) \\ \dot{x}_3(t) = x_1(t) \\ y(t) = x_1(t) \end{cases} \quad (58)$$

onde $x := [x_1 \ x_2 \ x_3]^T \in \mathcal{X} \subset \mathbb{R}^3$ é o vetor de estados, $\delta \in \Delta \subset \mathbb{R}$ é um parâmetro incerto, $u \in \mathbb{R}$ é a entrada de controle, $d \in \mathcal{D} \subset \mathbb{R}$ é uma perturbação exógena e $y \in \mathbb{R}$ é a saída controlada.

O objetivo de controle é regular $y(t)$ em torno de zero, considerando a seguinte lei de controle amostrada

$$u(t) = u[kT_s], \quad \forall t \in [kT_s, (k+1)T_s), \quad k \geq 0, \quad (59)$$

onde T_s é um período de amostragem suficientemente pequeno. Além disso, o sinal de controle discreto $u[kT_s] = u[k] = u$ será determinado utilizando a estratégia de controle descrita no Capítulo 3. Para isso, aplicando a aproximação para frente de Euler, definida como

$$x_+ = x + T_s \dot{x},$$

a representação discreta quase linear de (58), conforme definida em (41), pode ser descrita pelas matrizes

$$A(x, \delta, \theta) = A(x, \delta) = \begin{bmatrix} 1 & T_s & 0 \\ -T_s & 1 - T_s \delta (x_1^2 - 1) & 0 \\ T_s & 0 & 1 \end{bmatrix},$$

$$B_u(\delta) = B_u = \begin{bmatrix} 0 \\ T_s \\ 0 \end{bmatrix} \mathbf{e}$$

$$B_w(\delta) = B_w = \begin{bmatrix} 0 & 0 \\ T_s & T_s \\ 0 & 0 \end{bmatrix}.$$

Neste exemplo, será assumido que:

- $\mathcal{X} = \{x \in \mathbb{R}^3 : |x_1| \leq 1\}$;
- $\Delta = \{\delta \in \mathbb{R} : 0,5 \leq \delta \leq 0,9\}$;
- $\mathcal{D} = \{d \in \mathbb{R} : |d| \leq 0,5\}$; e
- $\mathcal{U} = \{u_2 \in \mathbb{R} : |u_2| \leq 0,5\}$.

Aqui, \mathcal{X} é limitado apenas na direção de x_1 , uma vez que $A(x, \delta)$ é apenas uma função de x_1 e δ .

Para projetar o controlador robusto, a matriz do sistema $A(x, \delta)$ é expressa da seguinte forma:

$$\begin{cases} A(x, \delta) = A_0(\delta) + \Pi(x)^T A_1(\delta) \\ 0_{6 \times 3} = \Omega_0(x) + \Omega_1(x)\Pi(x) \end{cases}$$

onde

$$A_0(\delta) = \begin{bmatrix} 1 & T_s & 0 \\ -T_s & T_s\delta + 1 & 0 \\ T_s & 0 & 1 \end{bmatrix}, A_1(\delta) = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & -T_s\delta & 0 \\ 0 & 0 & 0 \end{bmatrix},$$

$$\Pi(x) = \begin{bmatrix} x_1 \otimes I_3 \\ x_1^2 \otimes I_3 \end{bmatrix}, \Omega_0(x) = \begin{bmatrix} x_1 I_3 \\ 0_3 \end{bmatrix} \text{ e } \Omega_1(x) = \begin{bmatrix} -I_3 & 0_3 \\ x_1 I_3 & -I_3 \end{bmatrix}.$$

Então, ao considerar $T_s = 0,1$ s e $\delta = 0,75$, o problema de otimização (57) é resolvido aplicando uma busca em linha sobre $\mu \in (0, 1)$, o que leva a lei de controle robusto

$$u = (K_0 + K_1\Pi(x))x = K(x)x$$

para um $\mu = 0,315$ ótimo, onde

$$K(x)^T = \begin{bmatrix} -0,0643x_1^2 + 2,18 \cdot 10^{-5}x_1 - 72,5 \\ 0,693x_1^2 - 0,00163x_1 - 17,0 \\ -0,146x_1^2 + 0,000386x_1 - 101,0 \end{bmatrix},$$

e a seguinte estimativa da região de atração

$$\mathcal{R} = \{x \in \mathbb{R}^3 : 0,606x_1^2 + 0,0998x_1x_2 + 2,09x_1x_3 + 0,00786x_2^2 + 0,158x_2x_3 + 2,53x_3^2 \leq 10^{-3}\}.$$

Para o treinamento do controlador baseado em ESN, utilizou-se um conjunto de dados composto por 4994 amostras. Esse conjunto foi criado por meio de simulações do sistema em malha fechada, assumindo que $u[k] = u_1[k]$ e u_2 é um ruído branco filtrado. O ruído foi gerado com base em uma distribuição normal com média nula e desvio padrão unitário, aplicando um filtro passa-baixas com frequência de corte de $f_c = 2$ Hz para selecionar apenas as componentes

de baixa frequência, levando em consideração a dinâmica do sistema. Além disso, foi realizada uma operação para garantir que o sinal gerado permanecesse dentro dos limites estabelecidos no projeto de controle. Esse processo de geração do sinal para o treinamento está ilustrado na Figura 10. A perturbação $d(t)$ utilizada no sistema para o treinamento da rede neural também foi gerada considerando esse mesmo processo.

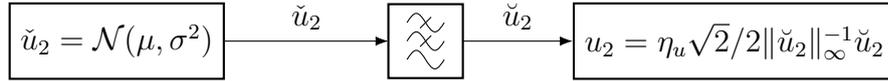


Figura 10 – Representação do processo de geração do sinal utilizado para o treinamento da rede neural.

Os hiperparâmetros da ESN foram otimizados utilizando o framework Optuna. Para este propósito, buscou-se maximizar uma função que avalia o ganho de desempenho do controlador proposto em relação ao controlador robusto. A expressão desta métrica pode ser definida como:

$$G(y, y_0) = 1 - \frac{\|y\|_2}{\|y_0\|_2}, \quad (60)$$

onde y é a saída do sistema quando o controlador proposto está atuando ($u = u_1 + u_2$) enquanto y_0 representa a saída do sistema apenas com o controlador robusto ($u = u_1$). Esse processo de otimização é explorado com maior detalhe no Apêndice D. Os hiperparâmetros obtidos são:

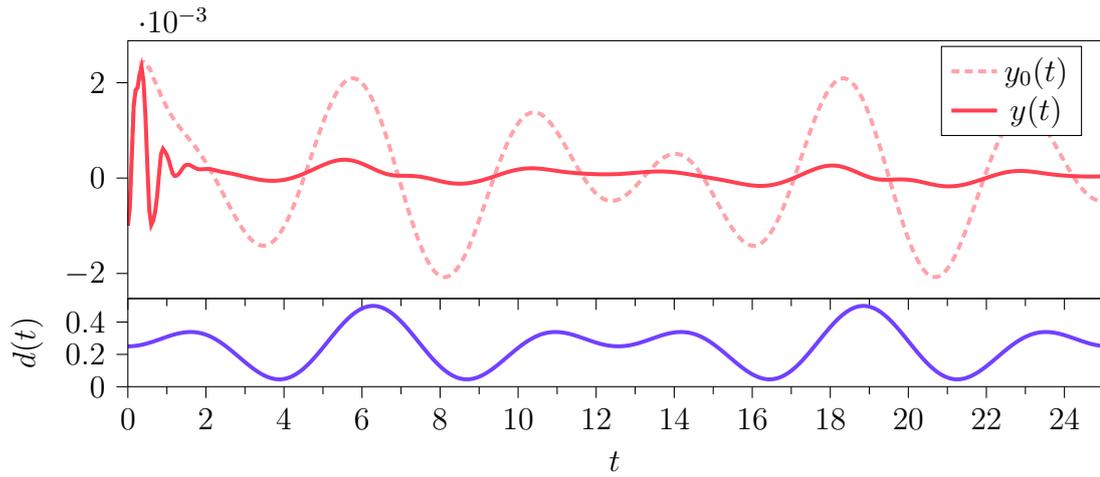
- (i) raio espectral $\rho = 0,414$;
- (ii) tamanho do reservatório $N = 161$;
- (iii) taxa de escoamento $\gamma = 0,365$;
- (iv) densidade do reservatório $0,638$;
- (v) número de amostras passadas $m = 1$; e
- (vi) parâmetro de atraso $\delta = 3$.

A performance da estratégia de controle proposta é avaliada por meio da simulação numérica considerando o modelo de tempo contínuo da planta e a lei de controle amostrada definida em (58) e (59), respectivamente. A Figura 11 apresenta a saída do sistema considerando a atuação do controlador proposto em comparação com o controlador robusto e a Figura 12 apresenta o sinal de controle gerado pela configuração de controle proposta para duas perturbações diferentes.

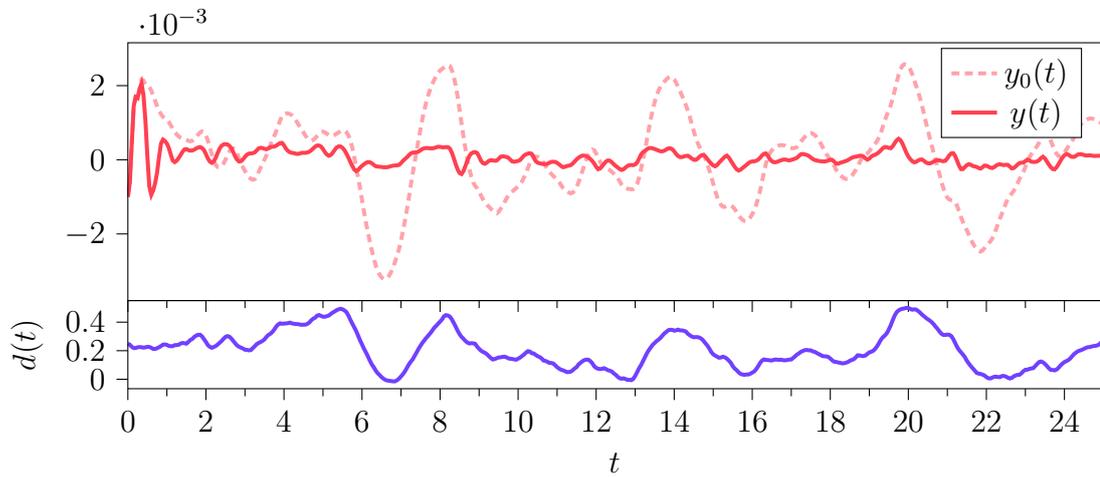
Além disso, realizou-se um experimento considerando 100 diferentes sinais de perturbação do tipo

$$d(t) = \eta_d \frac{\sqrt{2}}{2} \left(\frac{1}{2} \text{sen}(\omega_a t) + \frac{1}{2} \text{sen}(\omega_b t) \right), \quad (61)$$

com a duração de 25 s, onde as frequências ω_a e ω_b foram geradas de maneira aleatória, respeitando uma distribuição normal com frequência máxima de 2 Hz. Os resultados desse experimento são apresentados na forma de um histograma na Figura 13. Esse histograma mostra a



(a) $d(t) = 0,25(1 + \sin(\frac{5}{4}t) \sin(\frac{1}{4}t))$.



(b) $d(t)$ como ruído branco filtrado.

Figura 11 – Resposta do sistema de Van der Pol em malha fechada (representada pelas linhas vermelhas na parte superior das Figuras) para $u(t) = u_1(t)$ (linha tracejada) e $u(t) = u_1(t) + u_2(t)$ (linha contínua). As condições iniciais são $x_0^T = [-0,001 \ 0,001 \ 0]$, e o parâmetro θ é fixo em $0,75$; a perturbação no sistema é destacada em azul na parte inferior das Figuras.

distribuição de densidade de probabilidade da métrica do ganho de desempenho $G(y, y_0)$, que foi a mesma utilizada no processo de otimização dos hiperparâmetros da rede neural.

Os resultados mostram que a lei de controle resultante da combinação dos controladores robusto e por ESN trouxeram um ganho de performance em relação ao controlador robusto atuando sozinho. Para a perturbação apresentada na Figura 11a, esse ganho foi de $G(y, y_0) = 75,98\%$ e, para a Figura 11b, $G(y, y_0) = 76,23\%$. Já para o conjunto de resultados apresentados no histograma da Figura 13, obteve-se uma média de ganho de $G(y, y_0) = 77,36\%$, com um desvio padrão de $10,43\%$.

Além disso, a Figura 14 mostra o retrato de fase das trajetórias do estado (quando o

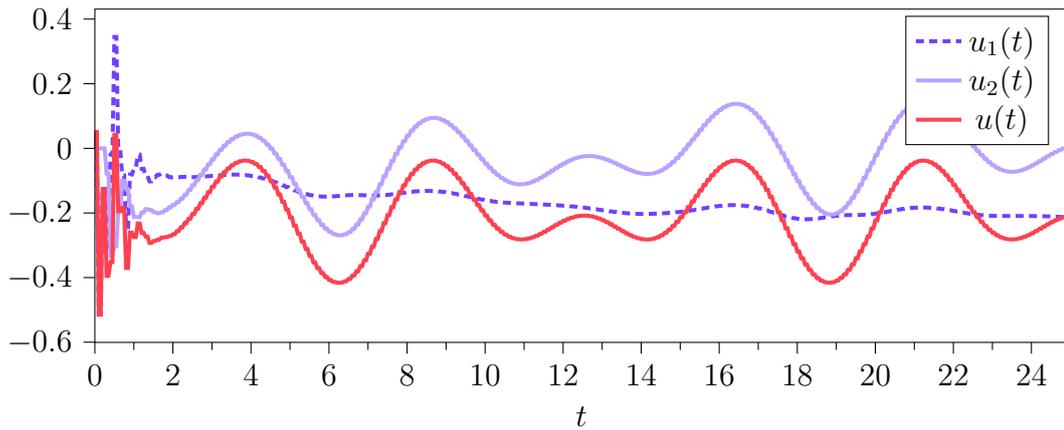
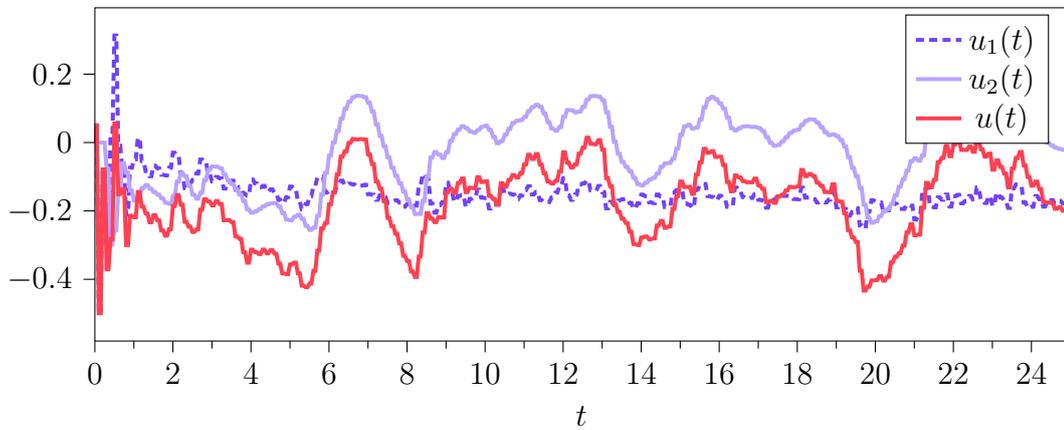
(a) $d(t) = 0,25(1 + \sin(\frac{5}{4}t) \sin(\frac{1}{4}t))$.(b) $d(t)$ como ruído branco filtrado.

Figura 12 – Sinal gerado pela configuração de controle proposta para o sistema de Van der Pol.

sistema é controlado apenas pelo controlador robusto em comparação quando controlado pela configuração de controle proposta) e a estimativa da região de atração \mathcal{R} do sistema em malha fechada, onde pode-se notar que as trajetórias do estado permanecem contidas no conjunto \mathcal{R} para todo $t \geq 0$, conforme esperado.

4.2 BOMBA ELÉTRICA SUBMERSÍVEL

O segundo resultado numérico apresentado aqui consiste na aplicação do controlador proposto em uma bomba elétrica submersível (*Electric Submersible Pump*, ESP) utilizada para elevação artificial de fluidos em um poço de produção de petróleo, ilustrado na Figura 15. O seguinte conjunto de equações diferenciais descreve o modelo simplificado desse sistema

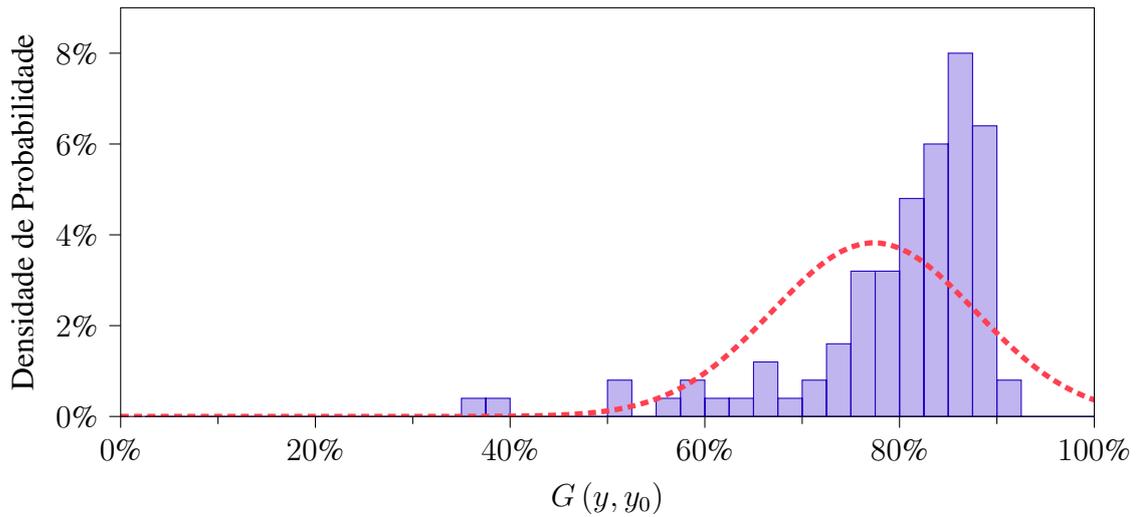


Figura 13 – Histograma da distribuição de densidade de probabilidade do ganho de desempenho para perturbações do tipo $d(t) = 0,25(\text{sen}(\omega_a t) + \text{sen}(\omega_b t))$. A linha tracejada em vermelho representa a função de densidade de probabilidade de uma distribuição Gaussiana com média de 77,36% e variância de 10,43% de $G(y, y_0)$.

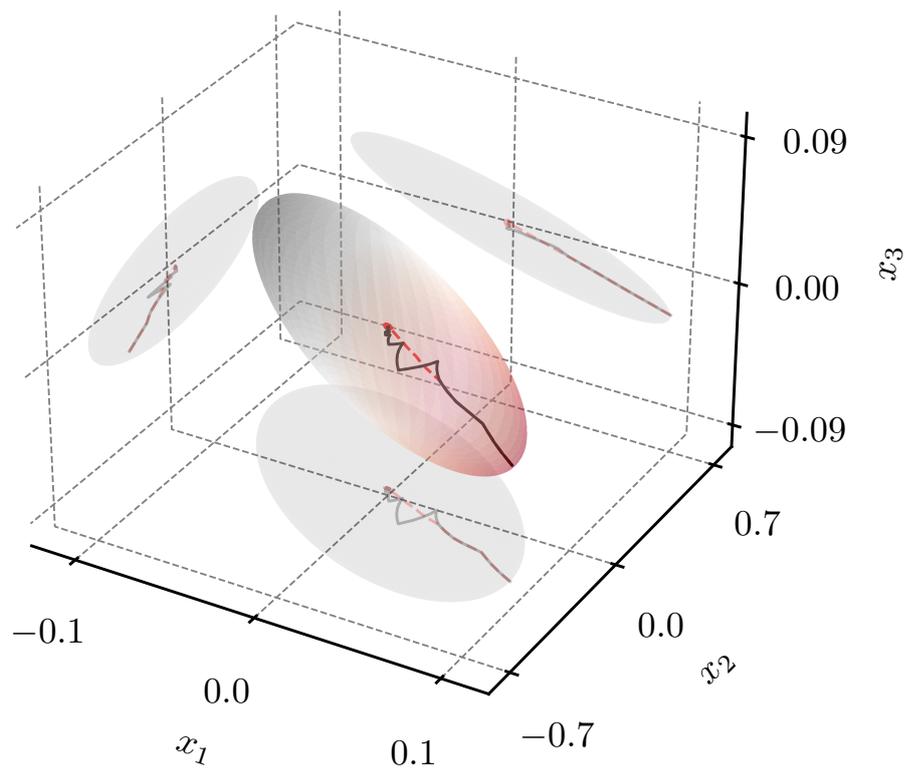


Figura 14 – Estimativa da região de atração e retrato de fase das trajetórias do estado para $x_0^T = [0,0912 \quad -0,286 \quad -0,0279]$, $\theta = 0,75$, e $d(t) = 0,25(1 + \sin(\frac{5}{4}t) \sin(\frac{1}{4}t))$, considerando as leis de controle $u = u_1$ (linha tracejada vermelha) e $u = u_1 + u_2$ (linha contínua preta).

(PAVLOV *et al.*, 2014; BINDER; PAVLOV; JOHANSEN, 2015):

$$\begin{cases} \dot{p}_{bh} = \frac{\beta_1}{V_1} (q_r - q) \\ \dot{p}_{wh} = \frac{\beta_2}{V_2} (q - q_c) \\ \dot{q} = \frac{1}{M} (p_{bh} - p_{wh} - \rho g h_w - \Delta p_f + \Delta p_p) \end{cases} \quad (62)$$

com

$$q_r = PI(p_r - p_{bh})$$

$$q_c = C_c \sqrt{p_{wh} - p_m z}$$

e variáveis e parâmetros descritos na Tabela 1. As equações que medem a perda de pressão por atrito no poço Δp_f e diferença de pressão na ESP Δp_p foram obtidas a partir do trabalho de Pavlov *et al.* (2014) e Binder, Pavlov e Johansen (2015) ao fixar o valor da viscosidade do fluido produzido μ_e em 0,175 Pa·s, resultando em

$$\Delta p_f = 2,4453 \cdot 10^7 q^2 \quad e$$

$$\Delta p_p = -2,4228 \cdot 10^{10} q^2 + 1,6778 \cdot 10^6 q f + 2,4714 \cdot 10^3 f^2.$$

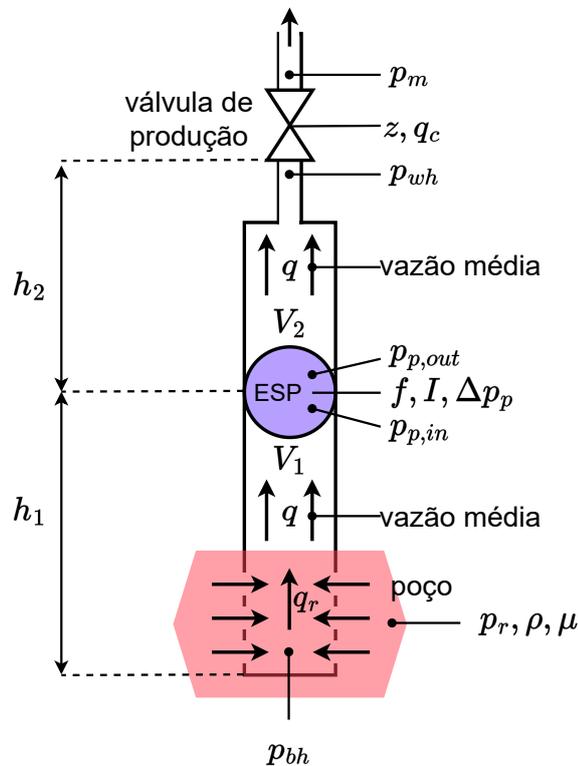


Figura 15 – Representação do sistema de um poço de produção de petróleo com elevação artificial de fluidos controlada com uma ESP, adaptado de Binder, Pavlov e Johansen (2015).

Tabela 1 – Parâmetros e constantes do modelo da ESP.

g	Aceleração gravitacional	9,81	m/s ²
C_c	Constante da válvula de <i>choke</i>	$2 \cdot 10^{-5}$	-
h_w	Distância vertical total no poço	$1 \cdot 10^3$	m
V_1	Volume do tubo abaixo da ESP	4,054	m ³
V_2	Volume do tubo acima da ESP	9,729	m ³
β_1	Módulo volumétrico abaixo da ESP	$1,5 \cdot 10^9$	Pa
β_2	Módulo volumétrico acima da ESP	$1,5 \cdot 10^9$	Pa
M	Parâmetro de inércia do fluido produzido	$1,992 \cdot 10^8$	kg/m ⁴
ρ	Densidade do fluido produzido	$9,5 \cdot 10^2$	kg/m ³
μ_e	Viscosidade do fluido produzido	0,175	Pa·s
p_r	Pressão do reservatório	$1,2 \cdot 10^7$	Pa
p_m	Pressão do coletor	$2 \cdot 10^6$	Pa
PI	Índice de produtividade do poço	$2,32 \cdot 10^{-9}$	m ³ /s/Pa
f	Frequência da ESP	55	Hz
f_0	Frequência característica de referência da ESP	60	Hz

Neste exemplo, o objetivo do controle é regular a saída do sistema, $\tilde{y} = p_{bh}$, em torno de um ponto de equilíbrio, por meio da determinação de um sinal de controle em tempo discreto, utilizando a estratégia descrita no Capítulo 3.

Para atingir esse objetivo, consideram-se as variáveis de estado $\tilde{x} = [p_{bh} \ p_{wh} \ q]^T$, a entrada de controle $\tilde{u} = z$, definida como a abertura da válvula *choke*, e a perturbação exógena $\tilde{d} = q_r$, que representa o influxo do reservatório para o poço. Então, o sistema (62) é transladado para a origem considerando

$$x = \tilde{x} - \bar{x},$$

$$u = \tilde{u} - \bar{u},$$

$$d = \tilde{d} - \bar{d},$$

onde \bar{x} é o ponto de equilíbrio das variáveis de estado, \bar{u} é o ponto de operação da entrada de controle, e \bar{d} é o valor nominal da perturbação.

Neste exemplo, considera-se a seguinte condição operacional:

- $\bar{x}_1 = 71,4686$ [bar];
- $\bar{x}_2 = 32,6771$ [bar];
- $\bar{x}_3 = 40,5334$ [m³/h];
- $\bar{u} = 0,5$; e
- $\bar{d} = 120$ [bar].

Ao adicionar um integrador ao sistema, reescrever as não-linearidades em função de parâmetros mensuráveis θ , de forma que ele possa ser representado na forma quase-linear de (41), e em seguida aplicar a aproximação de Euler $x_+ = x + T_s \dot{x}(t)$ para um período de amostragem $T_s = 0,2$ s, obtêm-se o sistema em tempo discreto definido pelas matrizes apresentadas em (63),

considerando pressões em bar e vazões em m³/h e os parâmetros definidos na Tabela 2.

$$A(x, \theta) = \begin{bmatrix} a_1 & 0 & a_2 & 0 \\ 0 & a_3 + a_4\theta_1 & a_5 & 0 \\ a_6 & a_7 & a_8 + a_9x_3 & 0 \\ a_{10} & 0 & 0 & 1 \end{bmatrix}$$

$$B_u(\theta) = \begin{bmatrix} 0 \\ b_{u1} + b_{u2}\theta_2 \\ 0 \\ 0 \end{bmatrix}, \quad B_d(\theta) = B_d = \begin{bmatrix} b_{d1} \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (63)$$

$$B_w(\theta) = \begin{bmatrix} \eta_u B_u(\theta) & \eta_d B_d \end{bmatrix}, \quad \theta = \begin{bmatrix} \theta_1 & \theta_2 \end{bmatrix}^T$$

$$\theta_1 = 25,2 \left((0,07888x_2 + 1)^{1/2} - 1 \right) x_2^{-1} - 1$$

$$\theta_2 = (0,07888x_2 + 1)^{1/2} - 0,998$$

Tabela 2 – Parâmetros para as matrizes do sistema da ESP.

a_1	0,82832	a_2	-0,20556	a_3	0,86225
a_4	-0,13775	a_5	0,08565	a_6	0,36144
a_7	-0,36144	a_8	0,54432	a_9	-0,00676
a_{10}	-0,2	b_{u1}	-6,92991	b_{u2}	-6,94374
b_{d1}	0,17168	η_u	0,14142	η_d	0,14142

Neste exemplo, consideram-se os seguintes limites operacionais:

- $\mathcal{X} = \{x \in \mathbb{R}^4 : |x_1| \leq 4,8531, |x_2| \leq 2,7323 \text{ e } |x_3| \leq 1,9467\}$;
- $\mathcal{D} = \{d \in \mathbb{R} : |d| \leq 0,1\}$; e
- $\mathcal{U} = \{u_2 \in \mathbb{R} : |u_2| \leq 0,1\}$.

Pela definição dos parâmetros variantes em (63) e os limites operacionais acima definidos, obtém-se

$$\Theta = \{\theta \in \mathbb{R}^2 : |\theta_1| \leq 0,05436, |\theta_2| \leq 0,10450\}.$$

Para projetar o controlador robusto, considerando a lei de controle amostrada de (59), as matrizes do sistema são reescritas na forma

$$\begin{cases} A(x, \delta, \theta) = A_0(x, \delta, \theta) + \Pi(x, \theta)^T A_1(\delta) \\ 0_{4 \times 4} = \Omega_0(x, \theta) + \Omega_1(x, \theta) \Pi(x, \theta) \end{cases}$$

onde $A_0(x, \delta, \theta) = A_0$, $A_1(\delta) = A_1 \mathbf{e}$

$$A_0 = \begin{bmatrix} a_1 & 0 & a_2 & 0 \\ 0 & a_3 & a_5 & 0 \\ a_6 & a_7 & a_8 & 0 \\ a_{10} & 0 & 0 & 1 \end{bmatrix}, A_1 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & a_4 & 0 & 0 \\ 0 & 0 & a_9 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T$$

$$\Pi(x, \theta) = \begin{bmatrix} x_3 \otimes I_4 & \theta_1 \otimes I_4 \end{bmatrix}^T,$$

$$\Omega_0(x, \theta) = \Pi(x, \theta) \mathbf{e} \quad \Omega_1(x, \theta) = -I_4.$$

Logo, o problema de otimização (57) é resolvido com uma busca linear sobre $\mu \in (0, 1)$, resultando na lei de controle robusta

$$u_1 = (K_0 + K_1 \Pi(x, \theta))x = K(x, \theta)x$$

para $\mu = 0,456$ ótimo, sendo

$$K(x, \theta) = \begin{bmatrix} 0,000363\theta_1 - 5,57 \cdot 10^{-6}x_3 + 0,58 \\ -0,0184\theta_1 - 5,69 \cdot 10^{-6}x_3 + 0,209 \\ 7,62 \cdot 10^{-5}\theta_1 + 0,00169x_3 - 0,248 \\ -0,000532\theta_1 + 8,1 \cdot 10^{-6}x_3 - 0,873 \end{bmatrix}^T$$

e a estimativa da região de atração dada por

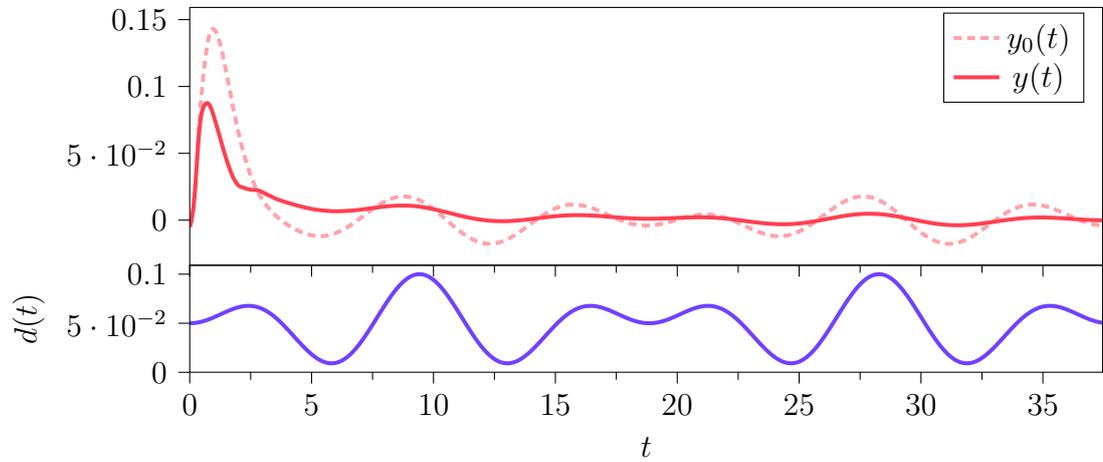
$$\mathcal{R} = \{x \in \mathbb{R}^4 : 13,6x_1^2 + 3,165x_1x_2 - 9,781x_1x_3 - 38,78x_1x_4 + \\ + 0,3667x_2^2 - 1,329x_2x_3 - 4,615x_2x_4 + 2,073x_3^2 + 13,94x_3x_4 + 29,25x_4^2 \leq 1\}.$$

Assim, para treinar o controlador baseado em ESN, um conjunto de dados com 2496 amostras foi construído com base em simulações do sistema em malha fechada (ou seja, com $u[k] = u_1[k]$). Da mesma forma que no exemplo anterior, assumiu-se que tanto $d(t)$ quanto u_2 são ruídos brancos filtrados, cujo processo de geração é ilustrado na Figura 10. Aqui, a frequência de corte utilizada no filtro passa-baixas foi de $f_c = 2$ Hz. Os hiperparâmetros para a ESN foram obtidos utilizando o framework Optuna, cujo processo de otimização resultou em:

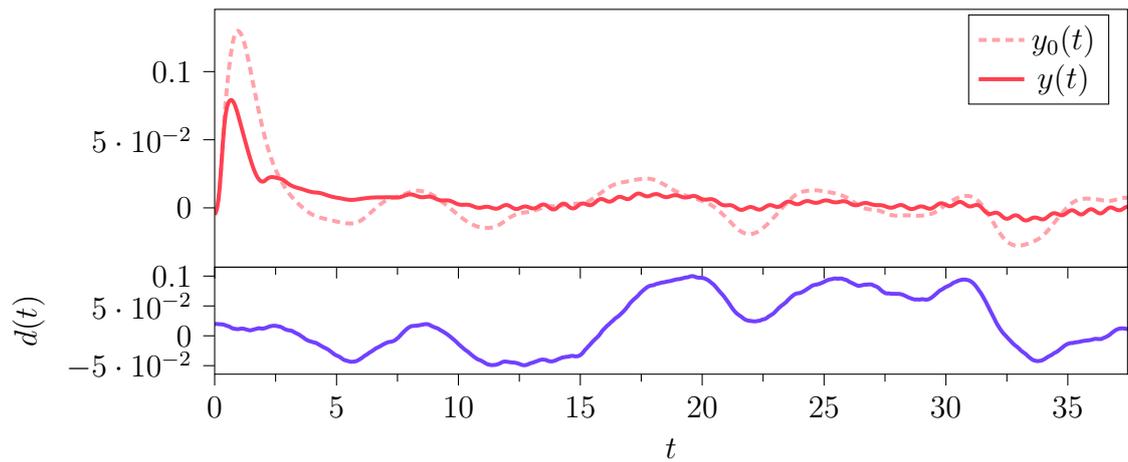
- (i) raio espectral $\rho = 0,570$;
- (ii) tamanho do reservatório $N = 90$;
- (iii) taxa de escoamento $\gamma = 0,967$;
- (iv) densidade do reservatório $0,785$;
- (v) número de amostras passadas $m = 3$; e
- (vi) parâmetro de atraso $\delta = 1$.

O desempenho do controlador proposto é avaliado considerando simulações numéricas do modelo contínuo da planta e a lei de controle amostrada definida em (62) e (59), respectivamente. A Figura 16 apresenta a saída do sistema considerando a atuação do controlador proposto

($u = u_1 + u_2$) em comparação com o controlador robusto ($u = u_1$) e a Figura 17 apresenta o sinal de controle gerado pela configuração de controle proposta para duas perturbações diferentes.



(a) $d(t) = 0,05(1 + \sin(\frac{5}{6}t) \sin(\frac{1}{6}t))$.

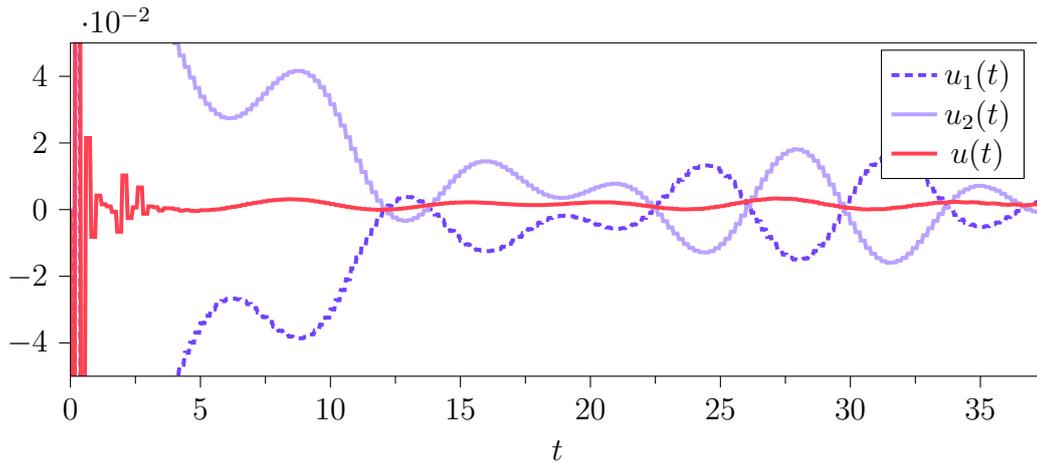


(b) $d(t)$ como ruído branco filtrado.

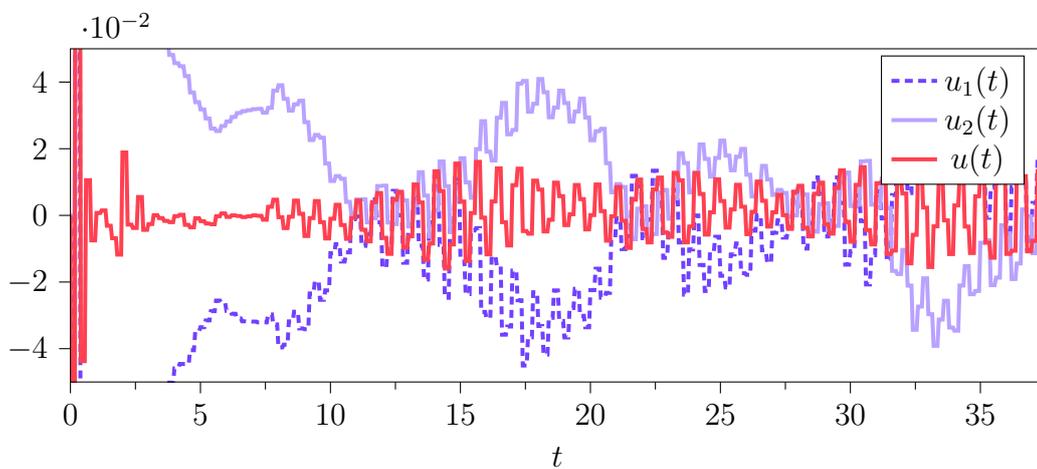
Figura 16 – Resposta do sistema da bomba elétrica submersível em malha fechada (representada pelas linhas vermelhas na parte superior das Figuras) para $u(t) = u_1(t)$ (linha tracejada) e $u(t) = u_1(t) + u_2(t)$ (linha contínua). As condições iniciais são $x_0^T = [-0,005 \ 0,01 \ -0,015 \ 0,2]$; a perturbação no sistema é destacada em azul na parte inferior das Figuras.

Também realizou-se para esse sistema um experimento considerando 100 diferentes sinais de perturbação apresentada em (61), para uma duração de 35 s e as frequências do sinal geradas de maneira aleatória, considerando uma frequência máxima de 2 Hz. Os resultados desse experimento estão apresentados no histograma da Figura 18.

Da mesma forma que para o exemplo anterior, os resultados do sistema da ESP em malha fechada mostram que a lei de controle resultante da combinação dos controladores robusto e por ESN (apresentadas na Figura 16) trouxeram um ganho de performance em relação ao controlador robusto atuando sozinho, onde, para o resultado apresentado na Figura 16a, obteve-se $G(y, y_0) =$



(a) $d(t) = 0,05(1 + \sin(\frac{5}{6}t) \sin(\frac{1}{6}t))$.



(b) $d(t)$ como ruído branco filtrado.

Figura 17 – Sinal gerado pela configuração de controle proposta para o sistema da bomba elétrica submersível.

43.29% e, para a Figura 16b, $G(y, y_0) = 45.89\%$. Para o conjunto de simulações cujos resultados são apresentados no histograma da Figura 18, obteve-se uma média de $G(y, y_0) = 38,67\%$ com um desvio padrão de 20,08%.

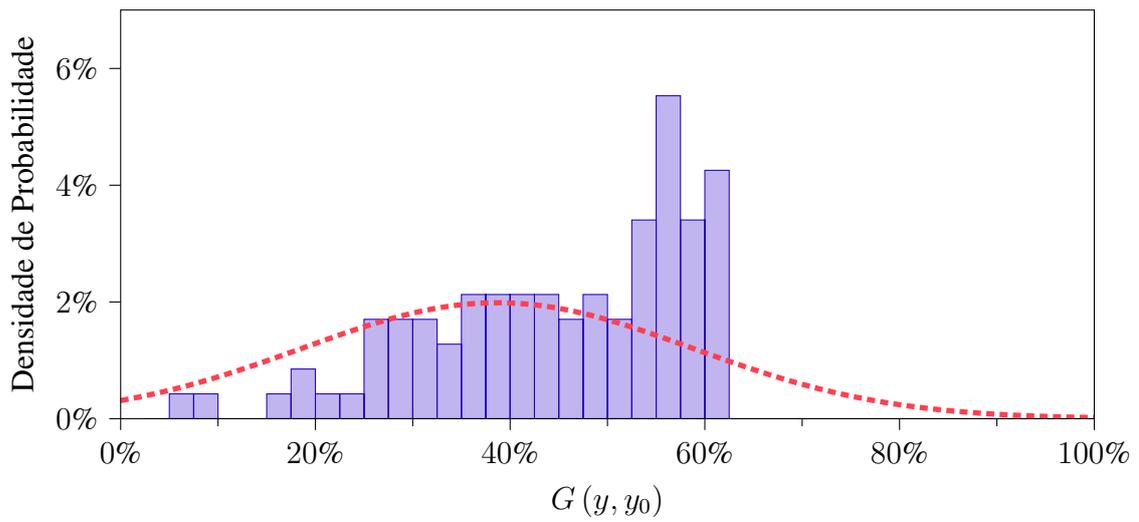


Figura 18 – Histograma da distribuição de densidade de probabilidade do ganho de desempenho para perturbações do tipo $d(t) = 0,05(\text{sen}(w_a t) + \text{sen}(w_b t))$. A linha tracejada em vermelho representa a função de densidade de probabilidade de uma distribuição Gaussiana com média de 38,67% e variância de 20,08% de $G(y, y_0)$.

5 CONCLUSÕES

Este trabalho apresentou uma estratégia de controle robusto combinando capacidades de aprendizado utilizando uma Rede Neural de Estado de Eco (ESN). Essa abordagem uniu as vantagens de um controlador robusto com as capacidades de aprendizado de uma ESN para melhorar o desempenho de sistemas em malha fechada.

A primeira etapa dessa estratégia envolveu o projeto de uma realimentação de estados não-linear, com o objetivo de garantir que a trajetória dos estados permanecesse limitada a um conjunto positivamente invariante, independentemente da lei de controle da ESN, assumindo uma ação limitada. Essa realimentação desempenhou um papel fundamental na estabilização do sistema, mesmo na presença de perturbações persistentes e condições iniciais não nulas.

A etapa seguinte consistiu no treinamento do controlador baseado em ESN. Através do aprendizado offline, a ESN foi capaz de minimizar os efeitos das perturbações na saída do sistema, melhorando a performance do sistema em malha fechada.

A estratégia de controle proposta foi validada experimentalmente em sistemas não-lineares incertos, permitindo demonstrar de maneira eficaz sua capacidade de atenuar os efeitos das perturbações persistentes nas saídas de sistemas complexos, como o Oscilador de Van der Pol e a Bomba Elétrica Submersível.

O trabalho desenvolvido oferece uma solução eficiente para a estabilização de sistemas em malha fechada que utilizam controladores baseados em redes neurais.

5.1 PERSPECTIVAS FUTURAS

Considerando a facilidade de treinamento das ESNs empregadas nesse trabalho, o controlador proposto oferece a vantagem de ser aplicável ao controle online de sistemas não-lineares e incertos. Isso permite que o controlador se adapte dinamicamente, permitindo o controle de plantas cujo comportamento varia ao longo do tempo. Além disso, mesmo que os pesos da rede neural mudem com o tempo, não há necessidade do reprojeto do controlador robusto, uma vez que a lei de controle gerada pela ESN sempre permanece dentro de limites conhecidos, garantindo a estabilidade do sistema independentemente das variações nos valores dos pesos dos neurônios, ao contrário das propostas de garantia de estabilidade de Suykens, Vandewalle e Moor (1995) e Barabanov e Prokhorov (2002), por exemplo.

Assim, é possível empregar algoritmos de mínimos quadrados recursivos para treinar a ESN, seguindo a abordagem utilizada no estudo de Waegeman, Wyffels e Schrauwen (2012), possibilitando a aplicação online do controlador proposto. Nesse contexto, já foram obtidos resultados preliminares para o controlador atuando de forma online, porém há a necessidade de amadurecimento e refinamento dos resultados.

Adicionalmente, é possível empregar uma estratégia para limitar o sinal de controle gerado pela ESN, introduzindo restrições durante o processo de otimização dos pesos de saída. Essa abordagem elimina a necessidade de utilizar uma função de ativação na saída enquanto

mantém o sinal gerado dentro dos limites estabelecidos para garantia de estabilidade do sistema em malha fechada, evitando possíveis limitações de desempenho no sinal de controle produzido pelo treinamento da rede neural.

Também, pode-se considerar o projeto do controlador robusto para sistemas de tempo contínuo, com o propósito de assegurar a estabilidade em um ambiente real (contínuo). Esta abordagem difere do visto neste trabalho, onde a estabilidade é assegurada apenas para sistemas discretos (ou discretizados), presumindo que sua representação seja uma aproximação suficientemente precisa do sistema real (contínuo).

Além dessas perspectivas futuras apresentadas, pode-se explorar o uso do controlador proposto atuando em sistemas cujo objetivo é seguimento de referência (constantes ou não) e realizar testes para diferentes tipos de redes neurais, procurando comparar a melhora de desempenho de treinamento offline frente ao uso de ESNs.

REFERÊNCIAS

AKIBA, Takuya; SANO, Shotaro; YANASE, Toshihiko; OHTA, Takeru; KOYAMA, Masanori. Optuna: A Next-Generation Hyperparameter Optimization Framework. *In: PROCEEDINGS of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. Anchorage, AK, USA: Association for Computing Machinery, 2019. (KDD '19), p. 2623–2631.

BARABANOV, Nikita E.; PROKHOROV, Danil V. Stability Analysis of Discrete-Time Recurrent Neural Networks. **IEEE Transactions on Neural Networks**, IEEE, v. 13, n. 2, p. 292–303, 2002.

BENGIO, Yoshua; SIMARD, Patrice; FRASCONI, Paolo. Learning Long-Term Dependencies with Gradient Descent is Difficult. **IEEE Transactions on Neural Networks and Learning Systems**, IEEE, v. 5, n. 2, p. 157–166, 1994.

BINDER, Benjamin J.T.; PAVLOV, Alexey; JOHANSEN, Tor A. Estimation of Flow Rate and Viscosity in a Well with an Electric Submersible Pump using Moving Horizon Estimation. *In: 6. v. 48*, p. 140–146. 2nd IFAC Workshop on Automatic Control in Offshore Oil and Gas Production.

BISHOP, C.M. **Pattern Recognition and Machine Learning**. [S.l.]: Springer, 2006. (Information Science and Statistics). ISBN 9780387310732.

BOF, Nicoletta; CARLI, Ruggero; SCHENATO, Luca. **Lyapunov Theory for Discrete Time Systems**. [S.l.: s.n.], 2018. arXiv: 1809.05289 [math.OA].

BOYD, S.; EL GHAOU, L.; FERON, E.; BALAKRISHNAN, V. **Linear Matrix Inequalities in System and Control Theory**. Philadelphia, PA: SIAM, jun. 1994. v. 15. (Studies in Applied Mathematics).

BOYD, Stephen; VANDENBERGHE, Lieven. **Convex Optimization**. [S.l.]: Cambridge University Press, mar. 2004. ISBN 0521833787.

BRUNTON, S.L.; KUTZ, J.N. **Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control**. [S.l.]: Cambridge University Press, 2022. ISBN 9781009098489.

DUAN, G.R.; YU, H.H. **LMIs in Control Systems: Analysis, Design and Applications**. [S.l.]: CRC Press, 2013. ISBN 9781466583009.

ELSHAWI, Radwa; MAHER, Mohamed; SAKR, Sherif. Automated Machine Learning: State-of-The-Art and Open Challenges. **arXiv - cs.LG**, 2019.

GOODFELLOW, Ian; BENGIO, Yoshua; COURVILLE, Aaron. **Deep Learning**. [S.l.]: MIT Press, 2016. <http://www.deeplearningbook.org>.

HOU, Zhong-Sheng; WANG, Zhuo. From model-based control to data-driven control: Survey, classification and perspective. **Information Sciences**, v. 235, p. 3–35, 2013.

JAEGER, Herbert. Adaptive nonlinear system identification with echo state networks. **Advances in Neural Information Processing Systems**, v. 15, 2002.

JAEGER, Herbert. **Method for Supervised Teaching of a Recurrent Artificial Neural Network**. 2008. US7321882B2. Patent. Assigned to Fraunhofer Gesellschaft zur Forderung der Angewandten Forschung eV.

JAEGER, Herbert. The “echo state” approach to analysing and training recurrent neural networks-with an erratum note. **Bonn, Germany: German National Research Center for Information Technology GMD Technical Report**, Bonn, v. 148, n. 34, p. 13, 2001.

JORDANOU, Jean; ANTONELLO, Eric; CAMPONOGARA, Eduardo. Online learning control with Echo State Networks of an oil production platform. **Engineering Applications of Artificial Intelligence**, v. 85, p. 214–228, 2019.

LI, Shuai; ZHOU, MengChu; LUO, Xin. Modified Primal-Dual Neural Networks for Motion Control of Redundant Manipulators With Dynamic Rejection of Harmonic Noises. **IEEE Transactions on Neural Networks and Learning Systems**, v. 29, n. 10, p. 4791–4801, 2018.

LIU, Kang-Zhi; YAO, Yu. **Robust Control: Theory and Applications**. [S.l.]: John Wiley & Sons (Asia) Pte Ltd, 2016. ISBN 9781118754375.

MAHMOUD, Tarek A; ABDO, Mohamed I; ELSHEIKH, Emad A; ELSHENAWY, Lamiaa M. Direct adaptive control for nonlinear systems using a TSK fuzzy echo state network based on fractional-order learning algorithm. **Journal of the Franklin Institute**, Elsevier, v. 358, n. 17, p. 9034–9060, 2021.

MAHMOUD, Tarek A.; ELSHENAWY, Lamiaa M. Echo State Neural Network Based State Feedback Control for SISO Afine Nonlinear Systems. **IFAC-PapersOnLine**, v. 48, n. 11,

p. 354–359, 2015. 1st IFAC Conference on Modelling, Identification and Control of Nonlinear Systems.

NELLES, O. **Nonlinear System Identification: From Classical Approaches to Neural Networks and Fuzzy Models**. [S.l.]: Springer Berlin Heidelberg, 2013. ISBN 9783662043233.

NGUYEN, Hoang Hai; ZIEGER, Tim; WELLS, Sandra C; NIKOLAKOPOULOU, Anastasia; BRAATZ, Richard D; FINDEISEN, Rolf. Stability certificates for neural network learning-based controllers using robust control theory. *In: IEEE. 2021 American Control Conference (ACC)*. [S.l.: s.n.], 2021. P. 3564–3569.

OLIVEIRA, Maurício C. de; SKELTON, Robert E. Stability tests for constrained linear systems. *In: MOHEIMANI, S.O. Reza (Ed.). Perspectives in robust control*. London: Springer London, 2001. P. 241–257.

PAN, Yunpeng; WANG, Jun. Model Predictive Control of Unknown Nonlinear Dynamical Systems Based on Recurrent Neural Networks. **IEEE Transactions on Industrial Electronics**, v. 59, n. 8, p. 3089–3101, 2012.

PAVLOV, Alexey; KRISHNAMOORTHY, Dinesh; FJALESTAD, Kjetil; ASKE, Elvira; FREDRIKSEN, Morten. Modelling and model predictive control of oil wells with electric submersible pumps. *In: IEEE. 2014 IEEE Conference on Control Applications (CCA)*. [S.l.: s.n.], 2014. P. 586–592.

SARANGAPANI, Jagannathan. **Neural Network Control of Nonlinear Discrete-Time Systems**. [S.l.]: CRC Taylor Francis, 2006. (Control Engineering Series). ISBN 0824726774, 9780824726775, 9781420015454, 1420015451.

STANFORD UNIVERSITY. **CS229: Additional Notes on Backpropagation**. Lecture notes for the CS229 course at Stanford University. 2019. Disponível em: <http://cs229.stanford.edu/notes-spring2019/backprop.pdf>.

SUN, Chenxi; SONG, Moxian; CAI, Derun; ZHANG, Baofeng; HONG, Shenda; LI, Hongyan. A Systematic Review of Echo State Networks from Design to Application. **IEEE Transactions on Artificial Intelligence**, IEEE, 2022.

SUYKENS, J.A.K.; VANDEWALLE, J.P.L.; MOOR, B.L. de. **Artificial Neural Networks for Modelling and Control of Non-Linear Systems**. [S.l.]: Springer US, 1995. ISBN 9780792396789.

WAEGEMAN, Tim; WYFFELS, Francis; SCHRAUWEN, Benjamin. Feedback Control by Online Learning an Inverse Model. **IEEE Transactions on Neural Networks and Learning Systems**, v. 23, n. 10, p. 1637–1648, 2012.

YANG, Li; SHAMI, Abdallah. On hyperparameter optimization of machine learning algorithms: Theory and practice. **Neurocomputing**, v. 415, p. 295–316, 2020. ISSN 0925-2312.

APÊNDICE A – DESIGUALDADES MATRICIAIS LINEARES

As desigualdades matriciais lineares, conhecidas como LMIs (*Linear Matrix Inequalities*), são uma ferramenta que desempenham um papel importante no controle robusto. Elas são utilizadas tanto para análise de estabilidade quanto para síntese de controladores. Além disso, abordagens utilizando LMIs permitem a resolução de problemas de projeto de controle que tenham diferentes especificações de projeto (DUAN; YU, 2013). Quando todas essas especificações são formuladas em termos de uma função de Lyapunov comum, o projeto do controlador se resume a resolver um sistema de LMIs (DUAN; YU, 2013).

A desigualdade matricial linear da função de Lyapunov foi a primeira LMI utilizada para análise da estabilidade de sistemas dinâmicos, publicada por Lyapunov em 1890 (DUAN; YU, 2013),

$$F(P) = A^T P + P A < 0, \quad (64)$$

sendo $A \in \mathbb{R}^{n \times n}$ uma matriz conhecida e

$$P \in \mathbb{S}^n = \{M \mid M = M^T \in \mathbb{R}^{n \times n}\}$$

uma variável de decisão. Nesse caso, um sistema $\dot{x}(t) = Ax(t)$, $x \in \mathbb{R}^n$ é assintoticamente estável se e somente se existir uma matriz $P > 0$ sujeita à (64). Aqui, $P > 0$ é uma matriz simétrica chamada de matriz definida positiva. De forma geral, para uma matriz simétrica $M \in \mathbb{S}^m$,

- todos os autovalores de M são positivos $\Leftrightarrow M > 0$;
- todos os autovalores de M são não negativos $\Leftrightarrow M \geq 0$;
- todos os autovalores de M são negativos $\Leftrightarrow M < 0$;
- todos os autovalores de M são não positivos $\Leftrightarrow M \leq 0$.

Essa definição também é válida para a LMI (64), onde a desigualdade ' $<$ ' indica que deseja-se encontrar uma matriz P tal que a matriz resultante $F(P)$ seja uma matriz definida negativa.

APÊNDICE B – DEFINIÇÕES RELEVANTES PARA ANÁLISE CONVEXA

Considerando o contexto deste trabalho, serão apresentadas aqui algumas definições relevantes para a análise convexa, que podem ser relacionadas ao uso de SDP para problemas de controle robusto.

Definição 3 (Conjuntos Convexos) (DUAN; YU, 2013) Um conjunto $\mathbb{F} \subset \mathbb{R}^n$ é considerado convexo se para quaisquer $x_1, x_2 \in \mathbb{F}$ e $\theta \in [0, 1]$,

$$\theta x_1 + (1 - \theta)x_2 \in \mathbb{F}.$$

Definição 4 (Envoltório Convexo) (BOYD, Stephen; VANDENBERGHE, 2004) Seja $\mathbb{F} \subset \mathbb{R}^n$. Então, o conjunto de todas as combinações convexas dos pontos $x_i \in \mathbb{F}$, $i = 1, \dots, k$, é chamado de envoltório convexo de \mathbb{F} e é denotada por $Co(\mathbb{F})$:

$$Co(\mathbb{F}) = \{\theta_1 x_1 + \dots + \theta_k x_k \mid \theta_i \geq 0, \theta_1 + \dots + \theta_k = 1\}$$

A Figura 19 ilustra as Definições 3 e 4. Em 19a e 19b estão representados exemplos de conjuntos convexo e não-convexo, respectivamente, e em 19c está ilustrado o envoltório convexo do conjunto não-convexo de 19b.

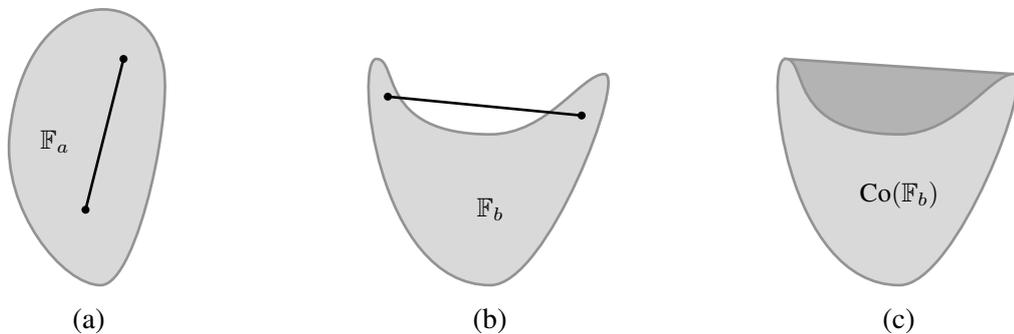


Figura 19 – Representação de (a) um conjunto convexo, (b) um conjunto não-convexo e (c) um envoltório convexo.

Definição 5 (Hiperplanos e Semi-Planos) (DUAN; YU, 2013) Suponha que $0 \neq h \in \mathbb{R}^n$. Então,

- Um conjunto na forma $\mathbb{H}_p = \{x \in \mathbb{R}^n \mid h^T x = b\}$ é chamado de hiperplano em \mathbb{R}^n .
- Um conjunto na forma $\mathbb{H}_s = \{x \in \mathbb{R}^n \mid h^T x \leq b\}$ é chamado de semi-plano em \mathbb{R}^n .

Definição 6 (Poliedros) (BOYD, Stephen; VANDENBERGHE, 2004) Um poliedro $\mathbb{P} \subset \mathbb{R}^n$ é um conjunto convexo que é definido pela interseção de um número finito de semi-planos e hiperplanos.

Definição 7 (Politopos) (BOYD, Stephen; VANDENBERGHE, 2004) Um politopo é o envoltório convexo de um poliedro compacto $\mathbb{P} \subset \mathbb{R}^n$.

Definição 8 (Elipsoides) (BOYD, Stephen; VANDENBERGHE, 2004) Uma elipsoide $\mathcal{E} \subset \mathbb{R}^n$ centrada na origem é um conjunto convexo que tem a forma

$$\mathcal{E} = \{x \in \mathbb{R}^n \mid x^T P^{-1} x \leq 1\}.$$

com $P \in \mathbb{S}^n$ sendo uma matriz simétrica positiva definida.

Definição 9 (Politopo contendo uma elipsoide) Considere um politopo descrito em termos da interseção de seus n_h semi-planos

$$\mathcal{X} = \{x \in \mathbb{R}^n \mid h_i^T x \leq 1, i = 1, \dots, n_h\}.$$

com $h_i \in \mathbb{R}^{n_x}$ definindo as faces de \mathcal{X} . Uma elipsoide $\mathcal{E} = \{x \in \mathbb{R}^n \mid x^T P^{-1} x \leq 1\}$ com $P \in \mathbb{S}^n$, estará dentro dos limites deste politopo se

$$\begin{bmatrix} 1 & h_i^T P^{-1} \\ P^{-1} h_i & P^{-1} \end{bmatrix} > 0, i = 1, \dots, n_h.$$

APÊNDICE C – LEMAS ESSENCIAIS PARA CONTROLE ROBUSTO COM USO DE LMIS

Aqui serão introduzidos alguns lemas essenciais que servem como suporte no desenvolvimento de projetos que envolvem LMIs, os quais são utilizados no projeto de controle apresentado neste trabalho.

Lema 2 (Lema do Complemento de Schur) (LIU; YAO, 2016) *Considere a matriz particionada*

$$A = \begin{bmatrix} A_1 & A_2 \\ A_2^T & A_3 \end{bmatrix}$$

onde $A_1 \in \mathbb{S}^n$, $A_2 \in \mathbb{R}^{n \times m}$ e $A_3 \in \mathbb{S}^m$. Então, é correto dizer que $A > 0$ se e somente se uma das seguintes condições é satisfeita:

1. $A_3 > 0$, $A_1 - A_2 A_3^{-1} A_2^T > 0$;
2. $A_1 > 0$, $A_3 - A_2^T A_1^{-1} A_2 > 0$.

Lema 3 (Lema de Finsler) (OLIVEIRA; SKELTON, 2001) *Seja $x \in \mathbb{R}^n$, $Q \in \mathbb{S}^n$ e $B \in \mathbb{R}^{m \times n}$, sendo $\text{rank}(B) < n$. As seguintes afirmações são equivalentes:*

1. $x^T Q x < 0$ para todo $x \neq 0$ de forma que $Bx = 0$;
2. $N_b^T Q N_b < 0$ com N_b sendo a base do espaço nulo de B ;
3. Existe um escalar $\mu \in \mathbb{R}$ de forma que $Q - \mu B^T B < 0$;
4. Existe uma matriz $L \in \mathbb{R}^{n \times m}$ de forma que $Q + LB + B^T L^T < 0$.

Lema 4 (S-Procedure) (BOYD, S. et al., 1994) *Sejam $T_0, \dots, T_p \in \mathbb{R}^{n \times n}$ matrizes simétricas conhecidas. Então a seguinte condição para T_0, \dots, T_p é válida:*

$$\zeta^T T_0 \zeta > 0, \forall \zeta \neq 0 \quad \text{tal que} \quad \zeta^T T_i \zeta \geq 0, \quad i = 1, \dots, p \quad (65)$$

se existirem $\tau_1 \geq 0, \dots, \tau_p \geq 0$ de modo que

$$T_0 - \sum_{i=1}^p \tau_i T_i > 0.$$

APÊNDICE D – OTIMIZAÇÃO DE HIPERPARÂMETROS COM O FRAMEWORK OPTUNA

A criação de um modelo de aprendizado de máquina de alta qualidade é um processo iterativo, complexo e demorado, demandando experiência do projetista para escolha correta dos hiperparâmetros (ELSHAWI; MAHER; SAKR, 2019). A escolha de hiperparâmetros que leva a uma configuração ótima é um processo fundamental para a construção de um modelo eficaz (YANG; SHAMI, 2020).

Neste trabalho, o processo de otimização dos hiperparâmetros foi conduzido utilizando o framework open-source Optuna para Python, proposto por Akiba *et al.* (2019). O Optuna apresenta uma arquitetura fácil de configurar, além de ser versátil, podendo ser adaptado a uma ampla variedade de tarefas (AKIBA *et al.*, 2019). A principal característica destacada por Akiba *et al.* (2019) é sua funcionalidade *define-by-run*, que permite que os usuários construam o espaço de busca de forma dinâmica. O algoritmo de busca de hiperparâmetros combina dois métodos de amostragem: o relacional, que explora as correlações entre os parâmetros, e o independente, que amostra cada parâmetro de forma independente (AKIBA *et al.*, 2019). Nesse caso ele primeiro amostra dados do espaço de busca de forma independente, identificando as relações entre esses dados e, em seguida, utiliza essas relações para realizar amostragens mais eficazes.

O processo de otimização consiste em minimizar ou maximizar uma função objetivo que utiliza como entrada os hiperparâmetros e retorna a pontuação obtida. A Figura 20 apresenta um exemplo simples usado para encontrar o valor de x , que varia no intervalo de -10 a 10 , minimizando a função $(x - 2)^2$.

```

1 import optuna
2
3 def objective(trial):
4     x = trial.suggest_float('x', -10, 10)
5     return (x - 2) ** 2
6
7 study = optuna.create_study()
8 study.optimize(objective, n_trials=100)
9
10 study.best_params # E.g. {'x': 2.002108042}

```

Figura 20 – Exemplo de código em Python utilizando framework Optuna para minimização da função $(x - 2)^2$.

Para encontrar os melhores hiperparâmetros para ESN utilizada no controlador responsável pela geração da entrada de controle u_2 nos sistemas de Van der Pol e da bomba elétrica submersível (consulte o Capítulo 5), foram realizadas 2000 iterações para otimização. A função objetivo associada ao problema de Van der Pol (Seção 4.1) pode ser encontrada no código apresentado na Figura 21. Da mesma forma, adaptou-se o problema para a busca pelos melhores hiperparâmetros da ESN para o sistema da bomba elétrica submersível, considerando o mesmo espaço de busca dos hiperparâmetros.

As linhas de código de 2 a 9 apresentam as informações sobre os hiperparâmetros otimizados e os intervalos de busca correspondentes. Os parâmetros δ e m desempenham um papel fundamental na geração do conjunto de dados para treinamento da ESN. Portanto, na linha 13, esse conjunto é construído a partir de dados de simulação do sistema para cada iteração. Estes dados de simulação consideram $u = u_1$ (sistema em malha fechada utilizando apenas o controlador robusto) e d e u_2 com ruído branco filtrado, levando em consideração apenas as componentes de baixa frequência. Esse conjunto de dados é posteriormente utilizado no treinamento da ESN, como indicado na linha 26.

```

1 def objective(trial):
2     nn_sr = trial.suggest_float("nn_sr", 0.3, 0.99)
3     nn_nr = trial.suggest_int("nn_nr", 50, 500)
4     nn_lr = trial.suggest_float("nn_lr", 0.3, 0.99)
5     nn_rd = trial.suggest_float("nn_rd", 0.3, 0.99)
6     nn_rp = trial.suggest_categorical("nn_rp", [1e-12, 1e-10, 1e-8,
7     1e-6, 1e-4])
8     nn_bias = trial.suggest_categorical("nn_bias", [0, 1])
9     delta = trial.suggest_int("delta", 1, 5)
10    m = trial.suggest_int("n", 1, 5)
11
12    sim_data = open('sim_data.txt')
13
14    X, Y = dataset_gen(delta, m, sim_data)
15
16    esn = PredictionESN(n_input=X.shape[1],
17                       n_output=Y.shape[1],
18                       spectralRadius=nn_sr,
19                       n_reservoir=nn_nr,
20                       leakingRate=nn_lr,
21                       reservoirDensity=nn_rd,
22                       outputBias=nn_bias,
23                       bias=nn_bias,
24                       regressionParameters=nn_rp,
25                       solver="lsqr")
26
27    esn.fit(X, Y)
28
29    x0 = np.array([[0, 0, 0]]).T
30    tmax = 25
31    t = np.arange(0, tmax, t_int)
32
33    w_l = [ [filtered_wn(len(t), f = [2/np.pi], f_s = 1/t_int, h = float
34    (d_d))],
35            [0.25*d_d*(1+np.sin(0.25*t)+np.sin(0.50*t)+np.sin(0.75*t))] ]
36
37    err = []
38
39    for w in w_l:
40        y = sim_vanderpol(esn, w, x0, t, ts, theta = 0.75)
41        hat_y = 0*t
42        err_w = ((RMSE(hat_y, y['u=u_1'])-RMSE(hat_y, y['u=u_1+u_2']))/
43        RMSE(hat_y, y['u=u_1']))*100
44        if err_w <= 0:
45            err_w = 0
46        err.append(err_w)
47
48    return float((err[0] + 2*err[1])/3)

```

Figura 21 – Função objetivo para otimização dos hiperparâmetros do sistema de Van der Pol utilizando o framework Optuna em Python.

A partir da rede neural treinada, realiza-se uma nova simulação do sistema em malha fechada (linhas 28 a 38). Nesta simulação, considera-se agora o uso do sinal u_2 gerado pela ESN treinada naquela iteração e a utilização de dois diferentes tipos de perturbações. Um deles é um ruído branco filtrado, como definido na linha 32, enquanto o outro é uma soma de senoides de frequências diferentes, conforme especificado na linha 33. Esses ruídos foram escolhidos como uma tentativa de tornar o controlador mais robusto e capaz de lidar com diferentes tipos de perturbações que podem ocorrer em aplicações reais. Dessa forma, o controlador pode adaptar-se a uma variedade de cenários quando for utilizado posteriormente.

O resultado da simulação é então utilizado para calcular o valor da função objetivo a ser maximizada (linha 45). Essa função representa uma média ponderada considerando as duas perturbações diferentes de uma métrica que avalia o ganho de desempenho do controlador quando $u = u_1 + u_2$ em comparação ao cenário em que apenas o controlador robusto está ativo, ou seja, $u = u_1$.