

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL  
ESCOLA DE ENGENHARIA  
ENG. DE CONTROLE E AUTOMAÇÃO

**DANIEL VARGAS ZANETTE - 00275635**

**ANÁLISE DA ESTABILIDADE DE  
SISTEMAS LINEARES  
CONTROLADOS POR REDES  
NEURAIS RELU**

Porto Alegre  
2023

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL  
ESCOLA DE ENGENHARIA  
ENG. DE CONTROLE E AUTOMAÇÃO

**DANIEL VARGAS ZANETTE - 00275635**

**ANÁLISE DA ESTABILIDADE DE  
SISTEMAS LINEARES  
CONTROLADOS POR REDES  
NEURAIS RELU**

Trabalho de Conclusão de Curso (TCC-CCA)  
apresentado à COMGRAD-CCA da Universi-  
dade Federal do Rio Grande do Sul como parte  
dos requisitos para a obtenção do título de *Ba-  
charel em Eng. de Controle e Automação* .

**ORIENTADOR:**

Prof. Dr. João Manoel Gomes da Silva Jr.

Porto Alegre  
2023

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL  
ESCOLA DE ENGENHARIA  
ENG. DE CONTROLE E AUTOMAÇÃO

**DANIEL VARGAS ZANETTE - 00275635**

**ANÁLISE DA ESTABILIDADE DE  
SISTEMAS LINEARES  
CONTROLADOS POR REDES  
NEURAIS RELU**

Este Trabalho de Conclusão de Curso foi julgado adequado para a obtenção dos créditos da Disciplina de TCC do curso *Eng. de Controle e Automação* e aprovado em sua forma final pelo Orientador e pela Banca Examinadora.

Orientador: \_\_\_\_\_

Prof. Dr. João Manoel Gomes da Silva Jr., UFRGS  
Doutor pela Université Paul Sabatier – Toulouse, França

Banca Examinadora:

Prof. Dr. João Manoel Gomes da Silva Jr., UFRGS  
Doutor pela Université Paul Sabatier – Toulouse, França

Prof. Dr. Alexandre Sanfelici Bazanella, UFRGS  
Doutor pela Universidade Federal de Santa Catarina - Florianópolis, Brasil

Profa. Dra. Lucíola Campestrini, UFRGS  
Doutora pela Universidade Federal do Rio Grande do Sul – Porto Alegre, Brasil

\_\_\_\_\_  
Alceu Heinke Frigeri  
Coordenador de Curso  
Eng. de Controle e Automação

Porto Alegre, Setembro 2023

## RESUMO

Este trabalho estuda o problema de análise de estabilidade exponencial da origem de um sistema linear a tempo discreto realimentado por uma rede neural (RN) do tipo ReLU. Primeiramente é mostrado que o sistema em malha fechada (MF) é equivalente a um sistema afim por partes (PWA) que pode ser descrito por uma representação implícita baseada em funções rampa. A partir desta representação, condições LMI para a certificação de estabilidade da origem do sistema em MF operado com o controle por RN são propostas. Além disso, as restrições de amplitude sobre o sinal de controle são levadas em consideração na análise de estabilidade em forma de uma saturação (potencialmente assimétrica) da saída da RN. Por fim, um exemplo numérico demonstra a aplicação das condições propostas.

**Palavras-chave:** Redes Neurais, Controle Preditivo por Modelo, Métodos de Lyapunov.

## ABSTRACT

This work studies the problem of exponential stability analysis of the origin of a discrete-time linear system controlled by a ReLU-type neural network (NN). Firstly, it is shown that the closed-loop system is equivalent to a piecewise affine (PWA) system that can be described by an implicit representation based on ramp functions. Using this representation, conditions in the form of LMIs for the stability certification of the origin of the closed-loop system operated with NN control are derived. In addition, the amplitude constraints on the control signal are taken into account in the stability analysis in the form of a (potentially asymmetric) saturation in the NN output. Finally, a numerical example demonstrates the application of the proposed conditions.

**Keywords:** Neural Network; Model Predictive Control; Lyapunov methods.

## LISTA DE ILUSTRAÇÕES

1	Exemplo de neurônio artificial. . . . .	14
2	Exemplo de uma RN <i>feedforward</i> com 3 entradas $(x_1, x_2, x_3)$ , 2 saídas $(y_1, y_2)$ e 1 camada oculta com 4 neurônios. . . . .	14
3	Gráfico de funções de ativação típicas de RN. . . . .	15
4	Gráfico da função PWA (7). . . . .	16
5	Gráfico da função rampa $r$ . . . . .	17
6	Evolução dos estados do sistema do exemplo 1 em MF para uma condição inicial de $x_1 = x_2 = 20$ . O sistema com controlador MPC é representado pelos círculos azuis, enquanto o sistema com controlador de RN é representado pelas cruces vermelhas. . . . .	27
7	Sinal de controle ao longo da trajetória do sistema do exemplo 1 em MF para uma condição inicial de $x_1 = x_2 = 20$ . A ação de controle do MPC e da RN são representados, respectivamente, por círculos azuis e por cruces vermelhas. . . . .	27
8	Evolução dos estados do sistema do exemplo 2 em MF para uma condição inicial de $x_1 = x_2 = 20$ . O sistema com controlador MPC é representado pelos círculos azuis, enquanto o sistema com controlador de RN é representado pelas cruces vermelhas. . . . .	29
9	Sinal de controle ao longo da trajetória do sistema do exemplo 2 em MF para uma condição inicial de $x_1 = x_2 = 20$ . A ação de controle do MPC e da RN são representados, respectivamente, por círculos azuis e por cruces vermelhas. . . . .	30

## LISTA DE TABELAS

1	Hiperparâmetros utilizados para o treinamento da RN. ....	26
2	Cr�terios de parada do treinamento da RN. ....	26

## LISTA DE ABREVIATURAS

<b>LMI</b>	<i>Linear Matrix Inequality</i> (Desigualdade Matricial Linear)
<b>MF</b>	Malha Fechada
<b>MPC</b>	<i>Model Predictive Control</i> (Controle Preditivo por Modelo)
<b>PWA</b>	<i>Piecewise Affine</i> (Afim por Partes)
<b>PWQ</b>	<i>Piecewise Quadratic</i> (Quadrática por Partes)
<b>QP</b>	<i>Quadratic Programming</i> (Programação Quadrática)
<b>ReLU</b>	<i>Rectified Linear activation Unit</i> (Unidade de Retificação Linear)
<b>RN</b>	Rede Neural



## LISTA DE SÍMBOLOS

$\Sigma$	Somatório
$\in$	Pertence
$\forall$	Para todo
$:=$	Definição
$v_{(i)}$	Indica o $i$ -ésimo elemento do vetor $v$
$M_{(i,j)}$	Indica o elemento $(i, j)$ da matriz $M$
$I_n$	Matriz identidade de dimensão $n$
$0_n$	Matriz nula de dimensão $n \times n$
$0_{n \times m}$	Matriz nula de dimensão $n \times m$
$v^\top$	Vetor transposto de $v$
$M^\top$	Matriz transposta de $M$
$\mathbb{R}^n$	Espaço euclidiano de dimensão $n$
$\mathbb{R}^{n \times m}$	Conjunto das matrizes reais de dimensão $n \times m$
$\mathbb{S}^n$	Conjunto das matrizes simétricas de dimensão $n \times n$
$\mathbb{D}^n$	Conjunto das matrizes diagonais de dimensão $n \times n$
$\mathbb{P}^n$	Conjunto das matrizes positivas e simétricas de dimensão $n \times n$
$He\{M\}$	Soma com a matriz transposta, $He\{M\} := M + M^\top$
$\preceq$	Maior ou igual definido elemento a elemento
$diag(v)$	Matriz diagonal do vetor $v$
$\ x\ $	Norma euclidiana de $x$
$\ln(x)$	Logaritmo natural de $x$
$f'(x)$	Derivada da função $f(x)$

# SUMÁRIO

1	INTRODUÇÃO .....	10
2	REVISÃO DA LITERATURA.....	12
2.1	Controle preditivo baseado em modelo.....	12
2.2	Redes neurais artificiais .....	13
2.3	Sistemas PWA.....	16
2.3.1	Representação implícita de sistemas PWA .....	17
3	METODOLOGIA .....	19
3.1	Definição do problema .....	19
3.2	Representação do sistema PWA em MF .....	20
3.2.1	Representação implícita da RN.....	20
3.2.2	Representação implícita da saturação.....	21
3.2.3	Função composta PWA .....	21
3.2.4	Representação implícita do sistema em MF.....	22
3.3	Certificação de estabilidade .....	23
4	RESULTADOS E DISCUSSÕES.....	25
4.1	Exemplo 1 .....	25
4.1.1	Projeto do MPC .....	25
4.1.2	RN reduzida.....	25
4.1.3	Estabilidade Global da Origem.....	28
4.2	Exemplo 2 .....	28
4.2.1	Projeto do MPC .....	28
4.2.2	RN reduzida.....	29
4.2.3	Estabilidade Global da Origem.....	30
5	CONCLUSÃO .....	31
	REFERÊNCIAS .....	33
	APÊNDICES .....	36
	APÊNDICE A - ESTABILIDADE NO SENTIDO DE LYAPUNOV .....	37
A.1	<i>Linear Matrix Inequalities</i> .....	39

APÊNDICE B - TREINAMENTO DE RN .....	40
B.1 <i>Backpropagation</i> .....	40
B.2      Método do gradiente descendente .....	41
B.3      Método de Gauss-Newton .....	42
B.4      Método de Levenberg-Marquardt .....	42
APÊNDICE C - MATRIZES DA RN .....	43
C.1      Exemplo 1 .....	43
C.2      Exemplo 2 .....	43
APÊNDICE D - MATRIZ P .....	44
D.1      Exemplo 1 .....	44
D.2      Exemplo 2 .....	44

# 1 INTRODUÇÃO

Redes neurais e aplicações de inteligência computacional não são algo novo, a base teórica e conceitos sobre o assunto existem desde os anos 60 (SANGER; BALJEKAR, 1958). Porém, somente com os avanços recentes nas áreas de tecnologia da informação, em especial o aumento do poder computacional e a otimização dos processos para gerar/obter, analisar e armazenar dados, se tornou possível a aplicação de RNs para soluções de problemas reais (AGGARWAL, 2019). Devido a isto, a utilização de RNs para o controle de sistemas dinâmicos vem sendo amplamente explorada na área acadêmica, em especial para controle de sistemas não lineares ou sistemas lineares sujeitos a não linearidades como a saturação zona morta (LEE et al., 2020; ZHOU et al., 2019; BOSE, 2007).

Apesar dos avanços nas áreas de inteligência computacional e do aumento do interesse acadêmico na utilização de RNs para aplicações de controle, os controladores baseados em RN ainda sofrem com a falta de certificação de estabilidade e segurança devido à complexidade da estrutura das RNs. Uma possível forma de certificar a estabilidade da origem para sistemas controlados por RN é utilizar as condições apresentadas em (YIN; SEILER; ARCAK, 2022). Porém, tais condições são baseadas em condições de setor ao invés de uma representação exata do sistema, o que pode acrescentar conservadorismo na análise. Outro problema surge quando o sistema a ser controlado apresenta limitações na sua entrada de controle: como essas restrições de amplitude do sinal de controle não são explicitamente garantidos pela RN, mesmo que a estabilidade da origem do sistema realimentado pela RN possa ser certificada, tal conclusão não é válida para a RN sujeita aos limites na entrada de controle.

Com isso em vista, este trabalho propõe o uso da *representação implícita baseada em rampas* para sistemas afins por partes (PWA, do inglês *Piecewise Affine*) (GROFF; VALMORBIDA; GOMES DA SILVA JR., 2023) para analisar a estabilidade exponencial da origem do sistema em MF realimentado por uma RN ReLU sujeita à saturação (potencialmente assimétrica) na entrada de controle. Nesta abordagem os limites impostos na entrada de controle pelas possíveis restrições do atuador são levados em consideração na análise de estabilidade na forma de uma saturação de entrada, a qual é incorporada no modelo PWA. Assim, qualquer certificado de estabilidade exponencial da origem obtido é válido para a entrada de controle limitada, algo que não é diretamente obtido com técnicas de RN. Outra vantagem da análise de estabilidade proposta neste trabalho deriva do fato de que a representação implícita se mostrou menos conservadora do que abordagens que usam condições de setor para levar em conta o comportamento não linear induzido pelas funções rampa de ativação da RN na análise de estabilidade de sistemas sujeito à saturação, conforme visto em (CABRAL; VALMORBIDA; GOMES DA SILVA JR., 2022).

Para ilustrar a abordagem proposta, ao final deste trabalho são realizados exemplos numéricos onde uma RN é treinada para emular o comportamento de um MPC (do inglês *Model Predictive Control*) no controle de um sistema linear sujeito à saturação do sinal de controle. Este exemplo tem aplicações práticas, já que para sistemas onde o

período de amostragem é muito pequeno ou o cálculo de otimização do MPC é muito complexo computacionalmente, pode não ser possível computar a solução do MPC em tempo hábil. Neste caso, a aproximação do MPC por uma RN reduzida pode diminuir substancialmente o esforço computacional para o cálculo do sinal de controle sem grandes perdas de desempenho. Além disso, é possível garantir um limitante superior para o erro entre a lei de controle do MPC e a aproximação da RN (ROSS et al., 2022).

Com este escopo, a organização deste trabalho se dá seguinte forma: no Capítulo 2 é realizada uma revisão dos conceitos necessários para execução do trabalho, os quais envolvem MPC, RNs e técnicas para representação e certificação de estabilidade de sistemas PWA. No Capítulo 3 são descritos os passos realizados para formulação das condições de estabilidade do sistema, passando pela definição matemática do problema a ser explorado, pela representação do sistema linear controlado por RN sujeito à saturação no formato PWA e por último apresentando o teorema proposto para certificação da estabilidade do sistema. No Capítulo 4 são apresentados os exemplos numéricos realizados como forma de ilustrar a abordagem proposta. Por fim, no Capítulo 5 são apresentadas conclusões sobre os resultados e possíveis trabalhos futuros.

## 2 REVISÃO DA LITERATURA

Neste capítulo faz-se uma breve revisão de conceitos que serão utilizados ao longo do desenvolvimento deste trabalho. Primeiramente conceitos gerais sobre MPC são abordados, bem como alguns dos problemas que este controle apresenta e que espera-se mitigar com a utilização de RNs. Depois é apresentada uma revisão sobre RN, a sua estrutura e sua representação matemática. Por fim, são abordados conceitos referentes sistemas PWA e são apresentados os métodos e os lemas utilizados para desenvolver as condições para certificar a estabilidade assintótica destes sistemas.

### 2.1 CONTROLE PREDITIVO BASEADO EM MODELO

Técnicas MPC consistem em utilizar o modelo dinâmico do processo para controlá-lo satisfazendo restrições de estado e/ou entrada de controle. Dado um sistema linear a tempo discreto representado por

$$x_{k+1} = Ax_k + Bu_k, \quad (1)$$

onde  $x_k$  e  $x_{k+1} \in \mathbb{R}^n$  são, respectivamente, o estado atual e sucessor em uma dada amostra  $k \geq 0 \in \mathbb{N}$ ,  $u_k \in \mathbb{R}^{n_u}$  é a entrada de controle nesta mesma amostra  $k$  e as matrizes  $A \in \mathbb{R}^{n \times n}$  e  $B \in \mathbb{R}^{n \times n_u}$  definem o sistema. O MPC pode ser definido a partir de um problema de Programação Quadrática (QP, do inglês *Quadratic Programming*) como, por exemplo (HERCEG et al., 2013),

$$\begin{aligned} \min_U \quad & \sum_{k=0}^{N_h-1} x_k^\top Q_x x_k + u_k^\top Q_u u_k, \\ \text{sujeito à (1), } & GU \leq R + Ex_0, \end{aligned} \quad (2)$$

onde  $Q_x \geq 0 \in \mathbb{S}^n$  e  $Q_u > 0 \in \mathbb{S}^{n_u}$  são matrizes de ponderação do problema e  $N_h$  é o tamanho do horizonte de predição. O vetor  $U^\top = [u_0^\top \ \dots \ u_{N_h-1}^\top] \in \mathbb{R}^{N_h n_u}$  representa a sequência de entradas de controle e as matrizes  $G \in \mathbb{R}^{n_c \times N_h n_u}$ ,  $E \in \mathbb{R}^{n_c \times n}$  e  $R \in \mathbb{R}^{n_c}$  representam as  $n_c$  restrições de (2), que podem representar limites nas entradas e/ou estados. Uma vez que a sequência de controle ótima  $U^*$  é computada, seu primeiro elemento, ou seja,  $u_0^*$ , é aplicado no processo e uma política de horizonte deslizante é empregada (CAMACHO; BORDONS, 2007; MACIEJOWSKI, 2000).

A política de horizonte deslizante do MPC exige que o problema QP (2) seja resolvido a cada período de amostragem, o que pode ser proibitivo em casos onde o mesmo é insuficiente para que a solução de (2) seja computada. Para contornar esse problema pode-se usar o *MPC explícito* (BEMPORAD et al., 2002), onde a lei de controle ótima resultante de (2), que é uma função continuamente Afim por Partes (PWA, do inglês *Piecewise Affine*), é calculada *offline*. Desta forma, durante o controle em tempo real do

sistema (1), o controlador deverá encontrar em uma tabela de consulta qual expressão da lei de controle PWA é válida para o estado atual. Entretanto, por vezes, mesmo a tarefa de busca da equação afim adequada pode não ser temporalmente possível dentro do período de amostragem definido, o que limita o MPC explícito para sistemas de tamanho moderado (VALMORBIDA; HOVD, 2023).

Outra abordagem é utilizar RNs para computar a ação de controle proveniente de um MPC, visto que funções continuamente PWA podem ser representadas de forma exata por uma RN com função de ativação ReLU (do inglês, *Rectifier Linear Unit*) (HE et al., 2020). Entretanto, tal representação exata pode implicar em uma RN demasiadamente grande e, com isso, ainda pode não ser factível computar sua saída no período de amostragem. Para contornar esse problema pode-se utilizar RNs reduzidas para aproximar a lei de controle proveniente de um MPC. Neste caso, é possível garantir um limitante superior para o erro entre a lei de controle do MPC e a aproximação da RN (ROSS et al., 2022), mas isso não implica imediatamente em uma garantia de estabilidade da origem do sistema em malha fechada (MF) operando com a RN. Para certificar a estabilidade da origem, pode-se utilizar os resultados apresentados em (YIN; SEILER; ARCAK, 2022). Porém, tais resultados fazem uso de condições de setor ao invés de uma representação exata da função de ativação da RN, o que pode acrescentar conservadorismo na análise, já que, ao usar condições de setor, acaba-se certificando a estabilidade para todas as possíveis não linearidades no setor. Outro problema surge da possível violação dos limites da entrada de controle por uma RN treinada para aproximar a lei de controle emergente de um MPC. Dessa forma, mesmo que a estabilidade da origem do sistema realimentado pela RN possa ser certificada, tal conclusão não é válida para a RN sujeita aos limites na entrada de controle.

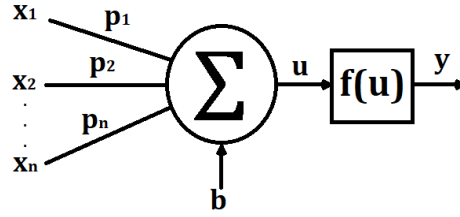
## 2.2 REDES NEURAIAS ARTIFICIAIS

As redes neurais artificiais (neste trabalho chamadas apenas de RNs por simplicidade) são modelos inspirados no comportamento das redes neurais biológicas do cérebro humano e buscam modelar as capacidades de processamento de informações dos sistemas nervosos (ROJAS, 1996). Segundo (HAYKIN, 2001), uma RN é um processador distribuído paralelamente constituído de unidades de processamento simples, com propensão natural para armazenar conhecimento experimental e torná-lo disponível para o uso. Uma RN é composta de diversos neurônios baseados no modelo *Perceptron* de um neurônio artificial (SANGER; BALJEKAR, 1958). Esse neurônio, apresentado na Figura 1 é composto por um conjunto  $n$  de entradas  $x_i$ ,  $i = 1, \dots, n$ , seus respectivos pesos  $p_i$ ,  $i = 1, \dots, n$ , um valor de *bias*  $b$  e uma função de ativação  $f$ , a qual pode ser uma função ReLU, sigmoide, tangente hiperbólica, entre outras.

A saída desse neurônio é o resultado da função de ativação do mesmo aplicada a uma variável  $u$  a qual corresponde ao somatório das  $n$  entradas multiplicadas pelos seus pesos e somadas ao *bias*. Em uma RN, esses neurônios são conectados de forma que as entradas de um neurônio são compostas das saídas de outros. A alta complexabilidade desses sistemas, gerada pela interconexão dos diferentes neurônios, torna as RNs uma ótima ferramenta para aproximação e/ou controle de sistemas dinâmicos não lineares como visto em (LEE et al., 2020; ZHOU et al., 2019; BOSE, 2007; LI et al., 2015).

Como visto em (JAIN; MAO; MOHIUDDIN, 1996), existem diversas formas de realizar a conexão dos neurônios em uma RN e dentre estas formas de conexão pode-se

**Figura 1:** Exemplo de neurônio artificial.



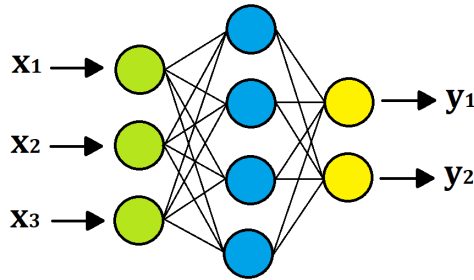
Fonte: Autor

separar as RNs em dois tipos principais:

- RNs *feedforward*: têm seus neurônios agrupados em camadas e a conexão entre eles dá-se de forma que as entradas dos neurônios de uma camada  $i$  são as saídas dos neurônios da camada anterior  $i - 1$ . Nesta configuração a camada 0 é chamada de camada de entrada e é composta das entradas propriamente ditas da RN, a última camada é chamada de camada de saída e sua saída é a saída propriamente dita da RN e as camadas intermediárias são chamadas de camadas ocultas.
- RNs recorrentes: possuem certo grau de realimentação entre seus neurônios de forma a possuir dinâmica interna tornando-as mais “poderosas” mas também tornando-as mais complexas e de difícil análise.

No escopo deste trabalho serão utilizadas RNs *feedforward*. Um exemplo deste tipo de RN pode ser visto na Figura 2.

**Figura 2:** Exemplo de uma RN *feedforward* com 3 entradas ( $x_1, x_2, x_3$ ), 2 saídas ( $y_1, y_2$ ) e 1 camada oculta com 4 neurônios.



Fonte: Autor

Além disso, será utilizada a seguinte representação matemática para uma RN *feedforward* de  $\ell$  camadas ocultas ( $\ell \geq 1 \in \mathbb{N}$ ) (YIN; SEILER; ARCAK, 2022):

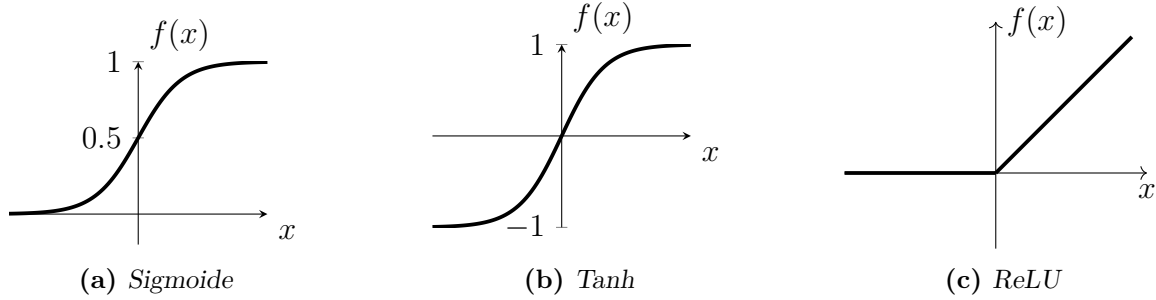
$$\begin{aligned} w_0 &= x, \\ w_i &= \phi(W_i w_{i-1} + b_i), \\ v &= \psi(W_{\ell+1} w_\ell + b_{\ell+1}), \quad i = 1, \dots, \ell, \end{aligned} \quad (3)$$

onde  $x \in \mathbb{R}^n$  é o vetor de entradas da RN,  $v \in \mathbb{R}^{n_v}$  é o vetor de saída da RN, e para a  $i$ -ésima camada  $w_i \in \mathbb{R}^{n_{w_i}}$  é o vetor de saída (ou seja, a saída dos  $n_{w_i}$  neurônios da  $i$ -ésima camada),  $b_i \in \mathbb{R}^{n_{w_i}}$  é o vetor dos *bias* e  $W_i \in \mathbb{R}^{n_{w_i} \times n_{w_{i-1}}}$  é a matriz dos pesos. Para definir o valor dos pesos e *bias* de uma RN, esta passa por um processo de treinamento o qual é



descrito no Apêndice B. Além disso,  $\phi : \mathbb{R}^{n_y} \rightarrow \mathbb{R}^{n_y}$  representa a função de ativação da camada oculta para uma entrada  $y \in \mathbb{R}^{n_y}$  e  $\psi : \mathbb{R}^{n_y} \rightarrow \mathbb{R}^{n_v}$  representa a função de ativação da camada de saída da RN. Tanto a função de ativação das camadas ocultas quanto a função de ativação da camada de saída da RN pode variar dependendo da aplicação, podendo ser uma função Sigmoide, tangente hiperbólica (Tanh), ReLU, entre outras. Na Figura 3 são apresentados os gráficos de algumas dessas funções de ativação.

**Figura 3:** Gráfico de funções de ativação típicas de RN.



Fonte: Autor

No caso da função de ativação da camada de saída, para problemas de regressão normalmente são utilizadas funções de ativação Softmax,

$$\psi_{(j)}(y) = \frac{e^{y_{(j)}}}{\sum_{i=0}^{n_y} e^{y_{(i)}}}, \quad j = 1, \dots, n_v, \quad (4)$$

onde  $n_v = n_y$ , de forma que cada uma das saídas da camada final da RN apresentará um resultado entre 0 e 1 que indicará a probabilidade dos dados de entrada serem de uma determinada classe e então a classe com maior probabilidade é escolhida como resultado da classificação. Já, para problemas de regressão/emulação, são normalmente utilizadas funções de ativação lineares, ou seja

$$\psi(y) = y, \quad (5)$$

pois se deseja que a saída da RN não seja restringida (como é o caso quando utilizada as funções de ativação como sigmoide e Softmax) e possa assumir qualquer valor.

Um ponto importante a ser ressaltado sobre as funções de ativação das RN é que algumas delas são não linearidades de setor, como é o caso da função ReLU e função Tanh e, portanto, podem ser modeladas como tal e utilizadas técnicas desenvolvidas para essa classe de não linearidades para derivar condições de estabilidade para sistemas que contenham tais redes neurais (YIN; SEILER; ARCAK, 2022). Porém, como já dito anteriormente na Seção 2.1, esta abordagem envolve certo conservadorismo na análise, pois acaba certificando a estabilidade para todas as possíveis não linearidades no setor.

Por fim, é importante destacar que existem vários atributos que influenciam no treinamento e performance do modelo da RN. Todos esses atributos selecionados antes do treinamento da RN são considerados hiperparâmetros, os quais consistem no número de neurônios e camadas ocultas da RN, função de ativação de cada camada, algoritmo de treinamento escolhido, parâmetros do algoritmo de treinamento escolhido, critérios de parada para o treinamento da RN, entre outros. A escolha dos hiperparâmetros utilizados para a RN é uma etapa muito importante para o desenvolvimento da RN e tem grande influência na performance da mesma, existindo vários estudos e métodos de otimização para

estes, como busca por grade, busca aleatória, algoritmo de enxame e algoritmo genético (MAN; TANG; KWONG, 1996). Porém, como o foco deste trabalho é na obtenção de critérios para análise da estabilidade de sistemas controlados por RN e não em obter a melhor RN possível para um certo problema, a maioria dos hiperparâmetros utilizados foram selecionados de forma empírica.

## 2.3 SISTEMAS PWA

Sistemas PWA em tempo discreto pertencem a uma classe de sistemas que podem ser representados por

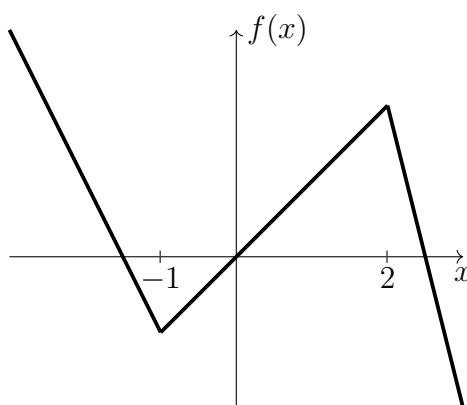
$$x^+ = f(x) \quad (6)$$

onde  $f(x)$  é uma função PWA,  $x$  é o vetor de estados do sistema no instante atual e  $x^+$  é o vetor de estados do sistema no próximo instante. Segundo (ARFKEN; WEBER, 2005), uma função PWA é uma função definida em  $L$  regiões  $\Gamma_i$ ,  $i = 1, \dots, L$ , onde em cada uma dessas regiões a função PWA é representada por uma função afim. Funções afins vetoriais são função do tipo  $f(x) = Ax + b$ , onde  $x$  é o vetor de entrada,  $A$  é uma matriz e  $b$  é um vetor. Um exemplo de função PWA de uma dimensão pode ser dado por

$$f(x) = \begin{cases} -2x - 3 & \text{se } x \leq -1 \\ x & \text{se } -1 < x \leq 2 \\ -4x + 10 & \text{se } x > 2, \end{cases} \quad (7)$$

e seu correspondente gráfico é apresentado na Figura 4.

**Figura 4:** Gráfico da função PWA (7).



Fonte: Autor

A classe de sistemas PWA tem sido usada para representar uma grande variedade de sistemas não lineares, desde circuitos (TESU; DARTU, 1992; JULIAN; DESAGES; AGAMENNONI, 1999) sistemas pneumáticos (ANDRIKOPOULOS et al., 2014) e biosistemas (AZUMA; YANAGISAWA; IMURA, 2008) até sistemas com não linearidades oriundas de sensores e atuadores, tais como saturações e zonas mortas (GOMES DA SILVA JR.; TARBOURIECH, 1999), já que essas não linearidades são funções PWA.

### 2.3.1 Representação implícita de sistemas PWA

Diferentes representações para funções PWA existem na literatura, cada uma tem suas vantagens e desvantagens e são aplicadas em diferentes áreas de interesse. Algumas das representações mais importantes são a representação PWA padrão (SONTAG, 1981) e a representação PWA canônica (CHUA; KANG, 1977). Neste trabalho será usada a representação PWA implícita apresentada em (GROFF; VALMORBIDA; GOMES DA SILVA JR., 2019), na qual uma função PWA  $f(x)$  pode ser descrita por

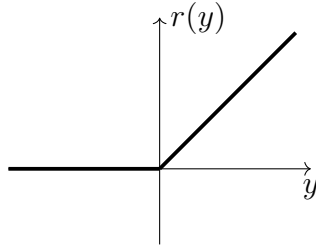
$$\begin{cases} f(x) = F_1x + F_2\phi(y) \\ y = F_3x + F_4\phi(y) + F_5, \end{cases} \quad (8)$$

onde  $x \in \mathbb{R}^n$ ,  $y \in \mathbb{R}^{n_y}$ ,  $F_1 \in \mathbb{R}^{n \times n}$ ,  $F_2 \in \mathbb{R}^{n \times n_y}$ ,  $F_3 \in \mathbb{R}^{n_y \times n}$ ,  $F_4 \in \mathbb{R}^{n_y \times n_y}$  e  $F_5 \in \mathbb{R}^{n_y}$ . Além disso,  $\phi : \mathbb{R}^{n_y} \rightarrow \mathbb{R}^{n_y}$  representa a função de ativação rampa vetorial, também chamada de ReLU, para uma entrada  $y \in \mathbb{R}^{n_y}$ , a qual é definida elemento a elemento pela função rampa escalar  $r : \mathbb{R} \rightarrow \mathbb{R}$ , ou seja,

$$\phi_{(j)}(y) = r(y_{(j)}) := \begin{cases} 0 & \text{se } y_{(j)} < 0, \\ y_{(j)} & \text{se } y_{(j)} \geq 0, \end{cases} \quad j = 1, \dots, n_y. \quad (9)$$

Na Figura 5 é apresentado o gráfico da função rampa  $r$ .

**Figura 5:** Gráfico da função rampa  $r$ .



Fonte: Autor

A principal vantagem de utilizar (8) é a caracterização não conservadora da função de rampa de valor vetorial  $\phi$  em termos de identidades e desigualdades, as quais serão expostas a seguir (GROFF; VALMORBIDA; GOMES DA SILVA JR., 2019).

**Lema 1.** A função rampa  $r$  satisfaz:

$$r(-y)r(y) = 0$$

**Prova.** caso  $y < 0$ ,  $r(y) = 0$ , já caso  $y > 0$ ,  $r(-y) = 0$ , por fim caso  $y = 0$ ,  $r(y) = r(-y) = 0$ .  $\square$

**Lema 2.** A função rampa  $r$  satisfaz:

$$r(-y) = r(y) - y$$

**Prova.** caso  $y < 0$ ,  $r(-y) = -y$  e  $r(y) = 0$  resultando na igualdade  $-y = -y$ , já caso  $y > 0$ ,  $r(-y) = 0$  e  $r(y) = y$  resultando na igualdade  $0 = 0$ , por fim caso  $y = 0$ ,  $r(-y) = 0$ ,  $r(y) = 0$  e  $y = 0$ .  $\square$

**Lema 3.** Para qualquer matriz  $T \in \mathbb{D}^{n_y}$  a função  $\phi$  definida em (9) satisfaz:

$$s_1(T, y) := \phi^\top(y)T\phi(-y) = 0 \quad \forall y \in \mathbb{R}^{n_y}.$$

**Prova.** Usando o Lema 1 e a definição da função rampa vetorial (9) deriva-se o Lema 3.  $\square$

**Lema 4.** Para qualquer vetor  $\xi \in \mathbb{R}^{n_\xi}$  e matriz  $R \in \mathbb{R}^{n_\xi \times n_y}$  a função  $\phi$  definida em (9) satisfaz:

$$s_2(\xi, R, y) := \xi R(y - (\phi(y) - \phi(-y))) = 0 \quad \forall y \in \mathbb{R}^{n_y}.$$

**Prova.** Usando o Lema 2 e a definição da função rampa vetorial (9) deriva-se o Lema 4.  $\square$

**Lema 5.** Para qualquer matriz  $M \in \mathbb{P}^{1+2n_y}$  a função  $\phi$  dada em (9) satisfaz:

$$s_3(M, y) := \begin{bmatrix} 1 \\ \phi(y) \\ \phi(-y) \end{bmatrix}^\top M \begin{bmatrix} 1 \\ \phi(y) \\ \phi(-y) \end{bmatrix} \geq 0 \quad \forall y \in \mathbb{R}^{n_y}.$$

**Prova.** Observando que  $r(\theta) \geq 0$  e usando a definição da função rampa vetorial (9) deriva-se o Lema 5.  $\square$

As propriedades enunciadas nos lemas acima provém diretamente da definição da função rampa vetorial e aplicam-se única e exclusivamente a este tipo de função. Além disso, a partir destas propriedades é enunciada a seguinte proposição para certificar a positividade de uma função quadrática por partes (PWQ, do inglês *Piecewise Quadratic*, apresentada no Apêndice A) descrita genericamente usando funções rampa.

**Proposição 1.** Dada uma função PWQ  $h : \mathbb{R}^n \rightarrow \mathbb{R}^n$  representada por

$$h(x) = \chi(x)^\top H \chi(x)$$

com

$$\chi^\top(x) = [1 \quad x^\top \quad \phi^\top(y(x)) \quad \phi^\top(-y(x))],$$

se existirem matrizes  $T \in \mathbb{D}^{n_y}$ ,  $R \in \mathbb{R}^{(1+n+2n_y) \times n_y}$  e  $M \in \mathbb{P}^{1+2n_y}$  que satisfaçam

$$h(x) + s_1(T, y(x)) + s_2(R, \chi(x), y(x)) - s_3(M, y(x)) \geq 0, \quad (10)$$

então  $h(x) \geq 0 \quad \forall x \in \mathbb{R}^n$ .

A prova da Proposição 1, apresentada em (GROFF; VALMORBIDA; GOMES DA SILVA JR., 2023), baseia-se na aplicação dos Lemas 3, 4 e 5 em (10). Com isto, dada uma função PWQ  $V(x)$ , é possível usar a Proposição 1 para derivar condições LMIs (a definição de LMIs é apresentada no Apêndice A.1) para provar que  $V(x) > 0$  e que  $\Delta V(x) = V(x^+) - V(x) < 0$  e consequentemente certificar estabilidade exponencial global da origem de um sistema PWA.

### 3 METODOLOGIA

Neste capítulo, inicialmente é realizada a definição matemática do problema a ser abordado, definindo o sistema a ser controlado, as características da RN a ser utilizada e as restrições de controle envolvidas. Posteriormente, é desenvolvida a representação implícita baseada em rampa para sistemas PWA em malha fechada a partir da representação implícita da RN, da representação implícita da saturação e da formulação da interconexão de dois sistemas PWA. Por fim, a partir da representação implícita do sistema em MF são desenvolvidas condições LMIs para verificar a estabilidade global exponencial da origem do sistema.

#### 3.1 DEFINIÇÃO DO PROBLEMA

Considerando um sistema linear a tempo discreto dado por

$$x^+ = Ax + Bu, \quad (11)$$

onde  $x$  e  $x^+ \in \mathbb{R}^n$  são, respectivamente, o estado atual e sucessor do sistema,  $u \in \mathbb{R}^{n_u}$  é a entrada de controle e as matrizes  $A \in \mathbb{R}^{n \times n}$  e  $B \in \mathbb{R}^{n \times n_u}$  definem o sistema.

Este sistema é controlado por uma RN, descrita por

$$\begin{aligned} w_0 &= x, \\ w_i &= \phi(W_i w_{i-1} + b_i), \\ v &= \psi(W_{\ell+1} w_\ell + b_{\ell+1}), \quad i = 1, \dots, \ell, \end{aligned} \quad (12)$$

onde  $x \in \mathbb{R}^n$  é o estado atual da planta,  $v \in \mathbb{R}^{n_v}$  é o sinal de controle da planta não saturado, e para a  $i$ -ésima camada  $w_i \in \mathbb{R}^{n_{w_i}}$  é o vetor de saída (ou seja, a saída dos  $n_{w_i}$  neurônios da  $i$ -ésima camada),  $b_i \in \mathbb{R}^{n_{w_i}}$  é o vetor dos *bias* e  $W_i \in \mathbb{R}^{n_{w_i} \times n_{w_{i-1}}}$  é a matriz dos pesos. Além disso,  $\phi$  é uma função ReLU definida em (9) e como a RN está sendo empregada para um problema de regressão/emulação, considera-se que a função de ativação da saída da RN  $\psi$  é uma função puramente linear (5).

Devido a restrições na ação de controle impostas pelo atuador, considera-se que o sinal aplicado à planta é uma versão saturada da saída  $v$  calculada pela RN. Seja  $\text{sat} : \mathbb{R} \rightarrow \mathbb{R}$  a função saturação escalar, a saída limitada  $u \in \mathbb{R}^{n_u}$  da RN é dada por

$$u_{(j)} = \text{sat}(v_{(j)}) := \begin{cases} \underline{u}_{(j)} & \text{se } v_{(j)} < \underline{u}_{(j)} \\ v_{(j)} & \text{se } \underline{u}_{(j)} \leq v_{(j)} \leq \bar{u}_{(j)} \\ \bar{u}_{(j)} & \text{se } v_{(j)} > \bar{u}_{(j)} \end{cases} \quad (13)$$

para  $j = 1, \dots, n_u$ , onde  $\underline{u} \preceq \bar{u} \in \mathbb{R}^{n_u}$  são, respectivamente, os vetores que concatenam os limites inferior e superior da saturação. Desta forma, sendo a origem do sistema em MF, dado por (11), (12) e (13), um ponto de equilíbrio, o seguinte problema é considerado.

**Problema 1.** *Certificar a estabilidade exponencial global da origem do sistema em MF dado por (11), (12) e (13).*

### 3.2 REPRESENTAÇÃO DO SISTEMA PWA EM MF

Observa-se que o sistema em MF dado por (11), (12) e (13) é um sistema PWA. Assim, para verificar a estabilidade do mesmo, será considerada a representação implícita baseada em rampas para sistemas PWA proposta em (GROFF; VALMORBIDA; GOMES DA SILVA JR., 2023). Neste caso o sistema em MF será reescrito na forma

$$\begin{cases} x^+ = F_1x + F_2\phi(y) \\ y = F_3x + F_4\phi(y) + F_5 \end{cases} \quad (14)$$

onde  $y \in \mathbb{R}^{n_y}$ ,  $F_1 \in \mathbb{R}^{n \times n}$ ,  $F_2 \in \mathbb{R}^{n \times n_y}$ ,  $F_3 \in \mathbb{R}^{n_y \times n}$ ,  $F_4 \in \mathbb{R}^{n_y \times n_y}$  e  $F_5 \in \mathbb{R}^{n_y}$ . Uma vez obtida a representação (14), pode-se então utilizar os resultados apresentados em (GROFF; VALMORBIDA; GOMES DA SILVA JR., 2023) para certificar a estabilidade exponencial global da origem do sistema em MF. Para reescrever o sistema no formato dado por (14), primeiramente são desenvolvidas as representações implícitas da RN (12) e da saturação (13) e, em sequência, é desenvolvida a formulação de funções compostas PWA para representar a interconexão em série da RN e da saturação.

#### 3.2.1 Representação implícita da RN

Definindo  $q_i := W_i w_{i-1} + b_i$  como a entrada da função de ativação da  $i$ -ésima camada e os vetores  $q \in \mathbb{R}^{n_w}$  e  $w \in \mathbb{R}^{n_w}$  (onde  $n_w = n_{w_1} + \dots + n_{w_\ell}$ ) como, respectivamente, a concatenação das entradas e saídas da função de ativação de cada camada da RN, ou seja,

$$q := \begin{bmatrix} q_1 \\ \vdots \\ q_\ell \end{bmatrix} \quad \text{e} \quad w := \begin{bmatrix} w_1 \\ \vdots \\ w_\ell \end{bmatrix}, \quad (15)$$

temos que  $w = \phi(q)$ . Dessa forma, a RN (12) pode ser expressa como

$$\begin{bmatrix} v \\ q \end{bmatrix} = N \begin{bmatrix} x \\ w \\ 1 \end{bmatrix}, \quad (16)$$

onde

$$\begin{aligned} N &:= \left[ \begin{array}{c|ccc|cc} 0_{n_u, n} & 0_{n_u, n_{w_1}} & \cdots & 0_{n_u, n_{w_{\ell-1}}} & W_{\ell+1} & b_{\ell+1} \\ W_1 & 0_{n_{w_1}, n_{w_1}} & \cdots & 0_{n_{w_1}, n_{w_{\ell-1}}} & 0_{n_{w_1}, n_{w_\ell}} & b_1 \\ 0_{n_{w_2}, n} & W_2 & \cdots & 0_{n_{w_2}, n_{w_{\ell-1}}} & 0_{n_{w_2}, n_{w_\ell}} & b_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0_{n_{w_\ell}, n} & 0_{n_{w_\ell}, n_{w_1}} & \cdots & W_\ell & 0_{n_{w_\ell}, n_{w_\ell}} & b_\ell \end{array} \right] \\ &= \begin{bmatrix} N_{vx} & N_{vw} & N_{vb} \\ N_{qx} & N_{qw} & N_{qb} \end{bmatrix} \end{aligned} \quad (17)$$

com  $N_{vx} \in \mathbb{R}^{n_u \times n}$ ,  $N_{vw} \in \mathbb{R}^{n_u \times n_w}$ ,  $N_{vb} \in \mathbb{R}^{n_u}$ ,  $N_{qx} \in \mathbb{R}^{n_w \times n}$ ,  $N_{qw} \in \mathbb{R}^{n_w \times n_w}$  e  $N_{qb} \in \mathbb{R}^{n_w}$ . Assim, a representação implícita da RN (12) com base na função rampa vetorial  $\phi$  é dada por

$$\begin{cases} v = N_{vx}x + \begin{bmatrix} N_{vw} & N_{vb} \end{bmatrix} \phi \left( \begin{bmatrix} q \\ 1 \end{bmatrix} \right) \\ \begin{bmatrix} q \\ 1 \end{bmatrix} = \begin{bmatrix} N_{qx} \\ 0_{1,n} \end{bmatrix} x + \begin{bmatrix} N_{qw} & 0_{n_w,1} \\ 0_{1,n_w} & 0 \end{bmatrix} \phi \left( \begin{bmatrix} q \\ 1 \end{bmatrix} \right) + \begin{bmatrix} N_{qb} \\ 1 \end{bmatrix}. \end{cases} \quad (18)$$

### 3.2.2 Representação implícita da saturação

Com o intuito de representar os limites de saída da RN como uma saturação, deve ser notado que a função saturação é também uma função PWA contínua e, portanto, pode ser escrita na forma implícita baseada na função rampa vetorial  $\phi$  como (GROFF, 2020)

$$\begin{cases} u = v + \begin{bmatrix} -I_{n_u} & I_{n_u} \end{bmatrix} \phi(y_s) \\ y_s = \begin{bmatrix} I_{n_u} \\ -I_{n_u} \end{bmatrix} v + \begin{bmatrix} -\bar{u} \\ \underline{u} \end{bmatrix} \end{cases} \quad (19)$$

Para demonstrar a equivalência entre (13) e (19), primeiramente nota-se que se a entrada  $v_{(j)}$  da saturação estiver na região linear, ou seja,  $\underline{u}_{(j)} < v_{(j)} < \bar{u}_{(j)}$ , temos que

$$\begin{aligned} \phi(v_{(j)} - \bar{u}_{(j)}) &= 0 \\ \phi(-v_{(j)} + \underline{u}_{(j)}) &= 0 \end{aligned}$$

e, conseqüentemente,  $u_{(j)} = v_{(j)}$ .

Por outro lado,  $v_{(j)} > \bar{u}_{(j)}$  implica em

$$\begin{aligned} \phi(v_{(j)} - \bar{u}_{(j)}) &= v_{(j)} - \bar{u}_{(j)} \\ \phi(-v_{(j)} + \underline{u}_{(j)}) &= 0 \end{aligned}$$

e, portanto,  $u_{(j)} = v_{(j)} - v_{(j)} + \bar{u}_{(j)} = \bar{u}_{(j)}$ .

Por fim, se  $v_{(j)} < \underline{u}_{(j)}$ , temos que

$$\begin{aligned} \phi(v_{(j)} - \bar{u}_{(j)}) &= 0 \\ \phi(-v_{(j)} + \underline{u}_{(j)}) &= -v_{(j)} + \underline{u}_{(j)} \end{aligned}$$

o que implica em  $u_{(j)} = v_{(j)} - v_{(j)} + \underline{u}_{(j)} = \underline{u}_{(j)}$ .

Dessa forma, a função implícita da saturação dada por (19) equivale à saturação aplicada elemento a elemento de  $v_{(j)}$  dada por (13).

### 3.2.3 Função composta PWA

Considerando duas funções PWA  $f_a : \mathbb{R}^{n_{x_a}} \rightarrow \mathbb{R}^{n_{f_a}}$  e  $f_b : \mathbb{R}^{n_{x_b}} \rightarrow \mathbb{R}^{n_{f_b}}$  na representação implícita baseada em rampas dadas por

$$\begin{cases} f_a(x_a) = F_{1a}x_a + F_{2a}\phi(y_a) \\ y_a = F_{3a}x_a + F_{4a}\phi(y_a) + F_{5a}, \end{cases} \quad (20)$$

$$\begin{cases} f_b(x_b) = F_{1b}x_b + F_{2b}\phi(y_b) \\ y_b = F_{3b}x_b + F_{4b}\phi(y_b) + F_{5b}. \end{cases} \quad (21)$$

A função composta  $f_c := f_b(f_a(x_a))$  pode então ser escrita como uma única função PWA com a seguinte representação implícita baseada em rampas

$$\begin{cases} f_c(x_a) = (F_{1b}F_{1a})x_a + \begin{bmatrix} F_{1b}F_{2a} & F_{2b} \end{bmatrix} \phi \left( \begin{bmatrix} y_a \\ y_b \end{bmatrix} \right) \\ \begin{bmatrix} y_a \\ y_b \end{bmatrix} = \begin{bmatrix} F_{3a} \\ F_{3b}F_{1a} \end{bmatrix} x_a + \begin{bmatrix} F_{4a} & 0 \\ F_{3b}F_{2a} & F_{4b} \end{bmatrix} \phi \left( \begin{bmatrix} y_a \\ y_b \end{bmatrix} \right) + \begin{bmatrix} F_{5a} \\ F_{5b} \end{bmatrix}. \end{cases} \quad (22)$$

### 3.2.4 Representação implícita do sistema em MF

Usando (22) e considerando que as funções  $f_a$  e  $f_b$  são, respectivamente, dadas pela RN (18) e pela saturação (19), a seguinte representação implícita é obtida para a RN sujeita à saturação em sua saída:

$$\begin{cases} u = \bar{F}_1 x + \bar{F}_2 \phi \left( \begin{bmatrix} q \\ 1 \\ y_s \end{bmatrix} \right) \\ \begin{bmatrix} q \\ 1 \\ y_s \end{bmatrix} = \bar{F}_3 x + \bar{F}_4 \phi \left( \begin{bmatrix} q \\ 1 \\ y_s \end{bmatrix} \right) + \bar{F}_5 \end{cases} \quad (23)$$

onde

$$\begin{aligned} \bar{F}_1 &= N_{vx}, & \bar{F}_2 &= \begin{bmatrix} N_{vw} & N_{vb} & -I_{n_u} & I_{n_u} \end{bmatrix}, \\ \bar{F}_3 &= \begin{bmatrix} N_{qx} \\ 0_{1,n} \\ N_{vx} \\ -N_{vx} \end{bmatrix}, & \bar{F}_4 &= \begin{bmatrix} N_{qw} & 0_{n_w,1} & 0_{n_w,2n_u} \\ 0_{1,n_w} & 0 & 0_{1,2n_u} \\ N_{vw} & N_{vb} & 0_{n_u,2n_u} \\ -N_{vw} & -N_{vb} & 0_{n_u,2n_u} \end{bmatrix}, \\ \bar{F}_5^\top &= \begin{bmatrix} N_{qb}^\top & 1 & -\bar{u}^\top & \underline{u}^\top \end{bmatrix}. \end{aligned}$$

Com isso, a representação implícita do sistema em MF dado por (11), (12) e (13) é dado por (14) com

$$\begin{aligned} F_1 &= A + BN_{vx}, & F_2 &= B \begin{bmatrix} N_{vw} & N_{vb} & -I_{n_u} & I_{n_u} \end{bmatrix}, \\ F_3 &= \begin{bmatrix} N_{qx} \\ 0_{1,n} \\ N_{vx} \\ -N_{vx} \end{bmatrix}, & F_4 &= \begin{bmatrix} N_{qw} & 0_{n_w,1} & 0_{n_w,2n_u} \\ 0_{1,n_w} & 0 & 0_{1,2n_u} \\ N_{vw} & N_{vb} & 0_{n_u,2n_u} \\ -N_{vw} & -N_{vb} & 0_{n_u,2n_u} \end{bmatrix}, \\ F_5^\top &= \begin{bmatrix} N_{qb}^\top & 1 & -\bar{u}^\top & \underline{u}^\top \end{bmatrix}. \end{aligned} \quad (24)$$

Na próxima seção, condições LMIs para a análise de estabilidade exponencial da origem do sistema em MF dado por (14) e (24) são apresentadas.



### 3.3 CERTIFICAÇÃO DE ESTABILIDADE

Nesta seção, são apresentadas condições LMIs que certificam a estabilidade exponencial da origem do sistema em MF dado por (14) e (24). Para derivar essas condições, são utilizados os lemas e proposições expostos em 2.3.1.

**Teorema 1.** *Considerando um sistema em MF representado por (14) com as matrizes  $F_1, F_2, F_3, F_4$  e  $F_5$  dadas em (24) e supondo que a origem  $x = 0$  é um ponto de equilíbrio, se existirem matrizes  $P_1 \in \mathbb{S}^n, P_2 \in \mathbb{R}^{n \times n_y}, P_3 \in \mathbb{S}^{n_y}, T_1 \in \mathbb{D}^{n_y}, T_2 \in \mathbb{D}^{n_y}, \tilde{T} \in \mathbb{D}^{2n_y}, M_1 \in \mathbb{P}^{1+2n_y}, M_2 \in \mathbb{P}^{1+2n_y}, \tilde{M} \in \mathbb{P}^{1+4n_y}, R_1 \in \mathbb{R}^{(1+n+2n_y) \times n_y}, R_2 \in \mathbb{R}^{(1+n+2n_y) \times n_y}$  e  $\tilde{R} \in \mathbb{R}^{(1+n+4n_y) \times 2n_y}$  e escalares positivos  $\epsilon_1, \epsilon_2$  e  $\eta < 1$  de forma que as seguintes inequações matriciais sejam satisfeitas*

$$H(\epsilon_1) + He\{\Psi(T_1) + R_1\Xi\} - \Upsilon M_1 \Upsilon^\top \geq 0, \quad (25a)$$

$$-H(\epsilon_2) + He\{\Psi(T_2) + R_2\Xi\} - \Upsilon M_2 \Upsilon^\top \geq 0, \quad (25b)$$

$$-\tilde{H}(\eta) + He\{\tilde{\Psi}(\tilde{T}) + \tilde{R}\tilde{\Xi}\} - \tilde{\Upsilon}\tilde{M}\tilde{\Upsilon}^\top \geq 0, \quad (25c)$$

onde  $n_y = n_w + 2n_u + 1$ ,

$$H(\epsilon_j) := \begin{bmatrix} 0 & 0_{1,n} & 0_{1,n_y} & 0_{1,n_y} \\ 0_{n,1} & P_1 - \epsilon_j I_n & P_2 & 0_{n,n_y} \\ 0_{n_y,1} & P_2^\top & P_3 & 0_{n_y} \\ 0_{n_y,1} & 0_{n_y,n} & 0_{n_y} & 0_{n_y} \end{bmatrix}, \quad \tilde{H}(\eta) := \begin{bmatrix} 0 & 0_{1,n} & 0_{1,2n_y} & 0_{1,2n_y} \\ 0_{n,1} & \tilde{P}_1(\eta) & \tilde{P}_2(\eta) & 0_{n,2n_y} \\ 0_{2n_y,1} & \tilde{P}_2^\top(\eta) & \tilde{P}_3(\eta) & 0_{2n_y} \\ 0_{2n_y,1} & 0_{2n_y,n} & 0_{2n_y} & 0_{2n_y} \end{bmatrix},$$

$$\Psi(T_j) := \begin{bmatrix} 0 & 0_{1,n} & 0_{1,n_y} & 0_{1,n_y} \\ 0_{n,1} & 0_n & 0_{n,n_y} & 0_{n,n_y} \\ 0_{n_y,1} & 0_{n_y,n} & 0_{n_y} & T_j \\ 0_{n_y,1} & 0_{n_y,n} & 0_{n_y} & 0_{n_y} \end{bmatrix}, \quad \tilde{\Psi}(\tilde{T}) := \begin{bmatrix} 0 & 0_{1,n} & 0_{1,2n_y} & 0_{1,2n_y} \\ 0_{n,1} & 0_n & 0_{n,2n_y} & 0_{n,2n_y} \\ 0_{2n_y,1} & 0_{2n_y,n} & 0_{2n_y} & \tilde{T} \\ 0_{2n_y,1} & 0_{2n_y,n} & 0_{2n_y} & 0_{2n_y} \end{bmatrix},$$

$$\Upsilon := \begin{bmatrix} 1 & 0_{1,2n_y} \\ 0_{n,1} & 0_{n,2n_y} \\ 0_{2n_y,1} & I_{2n_y} \end{bmatrix}, \quad \tilde{\Upsilon} := \begin{bmatrix} 1 & 0_{1,4n_y} \\ 0_{n,1} & 0_{n,4n_y} \\ 0_{4n_y,1} & I_{4n_y} \end{bmatrix}, \quad \Xi := \begin{bmatrix} F_5 & F_3 & F_4 - I_{n_y} & I_{n_y} \end{bmatrix},$$

$$\tilde{\Xi} := \begin{bmatrix} \tilde{F}_5 & \tilde{F}_3 & \tilde{F}_4 - I_{2n_y} & I_{2n_y} \end{bmatrix}, \quad \tilde{F}_3 := \begin{bmatrix} F_3 \\ F_3 F_1 \end{bmatrix}, \quad \tilde{F}_4 := \begin{bmatrix} F_4 & 0_{n_y} \\ F_3 F_2 & F_4 \end{bmatrix}, \quad \tilde{F}_5 := \begin{bmatrix} F_5 \\ F_5 \end{bmatrix},$$

$$\tilde{P}_1(\eta) := F_1^\top P_1 F_1 - (1 - \eta)P_1, \quad \tilde{P}_2(\eta) := \begin{bmatrix} F_1^\top P_1 F_2 - (1 - \eta)P_2 & F_1^\top P_2 \\ F_2^\top P_1 F_2 - (1 - \eta)P_3 & F_2^\top P_2 \end{bmatrix},$$

$$\tilde{P}_3(\eta) := \begin{bmatrix} F_2^\top P_1 F_2 - (1 - \eta)P_3 & F_2^\top P_2 \\ P_2^\top F_2 & P_3 \end{bmatrix},$$

para  $j \in \{1, 2\}$ , então a origem do sistema em MF é globalmente exponencialmente estável.

**Prova.** Considerando uma função de Lyapunov quadrática por partes  $V : \mathbb{R}^n \rightarrow \mathbb{R}$  dada por

$$V(x) = \begin{bmatrix} x \\ \phi(y(x)) \end{bmatrix}^\top \begin{bmatrix} P_1 & P_2 \\ P_2^\top & P_3 \end{bmatrix} \begin{bmatrix} x \\ \phi(y(x)) \end{bmatrix}. \quad (26)$$

A partir da Proposição 1 considerando  $h(x)$  como

$$h_1(x) := V(x) - \epsilon_1 x^\top x = \chi^\top(x) H(\epsilon_1) \chi(x)$$

e

$$h_2(x) := -V(x) + \epsilon_2 x^\top x = -\chi^\top(x) H(\epsilon_2) \chi(x)$$

as inequações (25a) e (25b) garantem limitantes quadráticos inferior e superior, respectivamente, tal que

$$\epsilon_1 \|x\|^2 \leq V(x) \leq \epsilon_2 \|x\|^2. \quad (27)$$

Portanto,  $V(x)$  é definida positiva e radialmente ilimitada.

Note agora que a partir de (41),  $V(x^+)$ , isto é, o valor da função de Lyapunov na amostra seguinte, é dada por

$$V(x^+) = \begin{bmatrix} x^+ \\ \phi(y(x^+)) \end{bmatrix}^\top \begin{bmatrix} P_1 & P_2 \\ P_2^\top & P_3 \end{bmatrix} \begin{bmatrix} x^+ \\ \phi(y(x^+)) \end{bmatrix},$$

onde  $x^+$  é o estado sucessor da amostra atual. Observando que a partir de (14) temos que

$$\begin{aligned} y(x^+) &= y(F_1 x + F_2 \phi(y(x))) \\ &= F_3 F_1 x + F_3 F_2 \phi(y) + F_4 \phi(y^+) + F_5, \end{aligned}$$

a expressão para  $V(x^+)$  pode ser reescrita como

$$V(x^+) = \begin{bmatrix} x \\ \phi(y) \\ \phi(y^+) \end{bmatrix}^\top \begin{bmatrix} F_1^\top P_1 F_1 & F_1^\top P_1 F_2 & F_1^\top P_2 \\ F_2^\top P_1 F_1 & F_2^\top P_1 F_2 & F_2^\top P_2 \\ P_2^\top F_1 & P_2^\top F_2 & P_3 \end{bmatrix} \begin{bmatrix} x \\ \phi(y) \\ \phi(y^+) \end{bmatrix},$$

onde, por simplicidade,  $y^+ := y(x^+)$ . Portanto, definindo

$$\tilde{y}(x) := \begin{bmatrix} y(x) \\ y(x^+) \end{bmatrix} = \tilde{F}_3 x + \tilde{F}_4 \tilde{y}(x) + \tilde{F}_5$$

e  $\tilde{\chi}^\top(x) := \begin{bmatrix} 1 & x^\top & \phi(\tilde{y}(x))^\top & \phi(-\tilde{y}(x))^\top \end{bmatrix}$  a condição

$$V(x^+) \leq (1 - \eta)V(x) \quad (28)$$

é equivalente a

$$h_3(x) = \tilde{\chi}^\top(x) \tilde{H}(\eta) \tilde{\chi}(x) \leq 0,$$

de forma que a partir da Proposição 1 com  $h(x) = h_3(x)$ , a inequação (25c) garante (28) ao longo das trajetórias do sistema dado por (14) e (24). Assim, (27) e (28) permitem concluir que  $\|x\| \leq c e^{\delta k} \|x_0\|$  com  $c = (\frac{\epsilon_2}{\epsilon_1})^{\frac{1}{2}}$  e  $\delta = \ln(\sqrt{1 - \eta})$ ,  $\forall x_0 \in \mathbb{R}^n$ , o que implica na estabilidade exponencial global da origem.  $\square$

Observa-se que, uma vez fixado um valor para  $\eta$  (ou seja, o decaimento exponencial), as inequações matriciais (25a), (25b) e (25c) se tornam LMIs.

## 4 RESULTADOS E DISCUSSÕES

Com o objetivo de ilustrar a aplicação das condições propostas, foram realizados dois exemplos numéricos nos quais é treinada uma RN com o objetivo de emular o comportamento de um MPC para um sistema linear a tempo discreto sujeito à saturação na saída de controle. Posteriormente, é verificada a estabilidade exponencial global do sistema em MF fazendo uso do Teorema 1.

### 4.1 EXEMPLO 1

Considerando um sistema linear a tempo discreto dado por (11) com

$$A = \begin{bmatrix} 0.7326 & -0.0861 \\ 0.1722 & 0.9909 \end{bmatrix}, \quad B = \begin{bmatrix} 0.6090 \\ 0.0640 \end{bmatrix} \quad (29)$$

e com uma saturação no sinal de controle dada por (13) com  $\bar{u} = 1$  e  $\underline{u} = -2$ . Para controlar esse sistema, deseja-se implementar um MPC levando em conta as restrições de amplitude (*i.e.* saturação) sobre a entrada de controle. Porém, devido ao potencial alto custo computacional para o cálculo da ação de controle do MPC, será treinada uma RN reduzida para aproximar o comportamento do mesmo. Por fim, para certificar a estabilidade do sistema em MF controlado pela RN reduzida, as condições do Teorema 1 são empregadas.

#### 4.1.1 Projeto do MPC

Foi projetado um MPC para o sistema dado por (11) e (29) considerando a saturação assimétrica (13) com  $\bar{u} = 1$  e  $\underline{u} = -2$ . O MPC resolve o problema QP (2) para calcular o valor da entrada de controle  $u$ , considerando  $Q_x = I_2$ ,  $Q_u = 0.5$ ,  $N_h = 5$ ,  $E = 0_{10,2}$ ,  $R^\top = [1 \ 1 \ 1 \ 1 \ 1 \ 2 \ 2 \ 2 \ 2 \ 2]$  e  $G^\top = [I_5 \ -I_5]$ .

#### 4.1.2 RN reduzida

Com o objetivo de reduzir o custo computacional para utilização do MPC projetado, foi treinada uma RN a partir do MPC com objetivo de aproximar o comportamento do mesmo. Foi utilizada uma RN ReLU com 2 entradas, 1 saída e 2 camadas ocultas com o seguinte número de neurônios {2 2}. Para treinar essa rede, foram calculadas as saídas do MPC para diferentes estados do sistema. Foi feita uma varredura no espaço de estados

do sistema usando um quadrado onde os valores de cada um dos estados varia de -5 até 5 com espaço de 0.1 entre eles. Esses valores foram então utilizados para o treinamento da RN com o uso do algoritmo *Levenberg-Marquardt backpropagation* (apresentado no Apêndice B.4), a Tabela 1 descreve os hiperparâmetros utilizados para o treinamento da RN e a Tabela 2 descreve os critérios de parada utilizados para tal treinamento, onde  $\alpha$  é a velocidade de aprendizado (descrita no Apêndice B.1) e  $\sigma$  é o parâmetro de amortecimento (descrito no Apêndice B.4).

**Tabela 1:** *Hiperparâmetros utilizados para o treinamento da RN.*

Hiperparâmetro	Valor
$\alpha$	1
$\sigma$ inicial	$10^{-3}$
Decréscimo de $\sigma$	0.1
Incremento de $\sigma$	10

Fonte: Autor

**Tabela 2:** *Critérios de parada do treinamento da RN.*

Critério de parada	Valor
Número máximo de épocas	$10^3$
Erro mínimo	0
Gradiente mínimo	$10^{-7}$
$\sigma$ máximo	$10^{10}$

Fonte: Autor

Após o treinamento da RN, o valor do *bias* da camada de saída da mesma ( $b_{\ell+1}$ ) foi ajustado para a garantir que saída da RN tenha valor zero para uma entrada nula, assim garantindo que a origem do sistema em MF é um ponto de equilíbrio. Para isso o valor de  $b_{\ell+1}$  foi atualizado da seguinte forma:

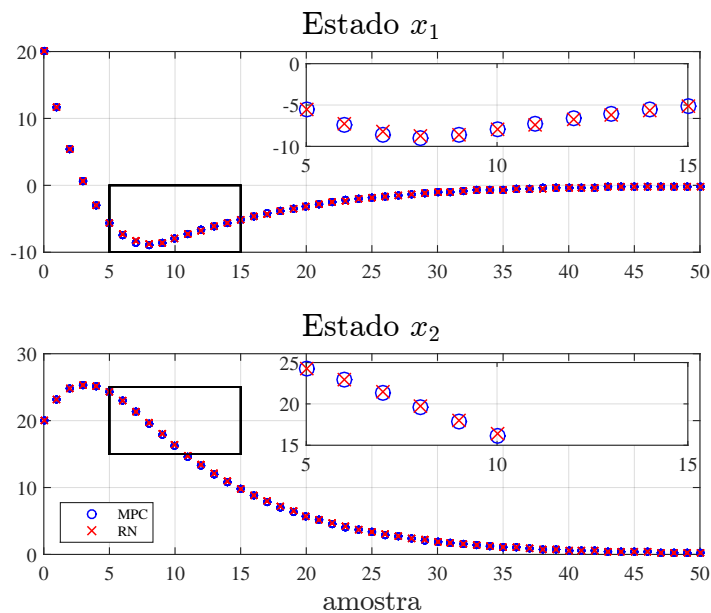
$$b_{\ell+1} = \bar{b}_{\ell+1} - v_0 \quad (30)$$

onde  $\bar{b}_{\ell+1}$  é o valor do *bias* da última camada obtido a partir do processo de treinamento e  $v_0$  é o valor da saída da RN treinada para uma entrada nula.

A RN obtida é descrita por (12) com as matrizes apresentadas no Apêndice C.1. A Figura 6 apresenta a comparação de uma trajetória do sistema controlado pelo MPC (2) parametrizado de acordo com a Seção 4.1.1 e pela RN que emula seu comportamento. Na Figura 7 pode-se observar a comparação do sinal de controle gerado pelo MPC com o sinal de controle da RN.

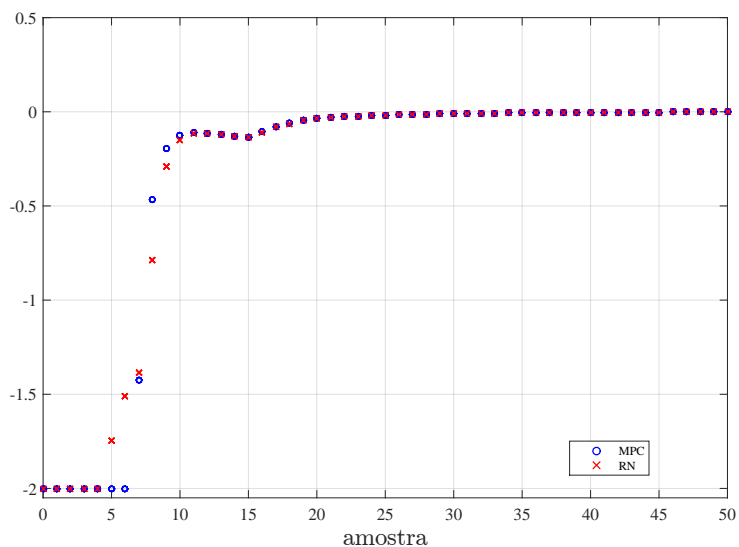
Nas duas figuras pode-se observar que a RN treinada tem o desempenho desejado tendo uma performance próxima do MPC projetado mesmo para condições iniciais fora do quadrado utilizado para o treinamento da mesma. Nos pontos das amostras de 5 a 10, onde ocorre a saturação do sinal de controle em  $-2$ , ocorreu a maior diferença entre o sinal de controle da RN e sinal de controle do MPC, indicando que a RN treinada não conseguiu emular tão bem o comportamento do MPC para estes valores, o que ocorre devido ao baixo número de neurônios da RN. Também destaca-se que o tempo necessário para calcular a ação de controle do MPC, calculada usando pacote MPT3 (HERCEG et al.,

**Figura 6:** Evolução dos estados do sistema do exemplo 1 em MF para uma condição inicial de  $x_1 = x_2 = 20$ . O sistema com controlador MPC é representado pelos círculos azuis, enquanto o sistema com controlador de RN é representado pelas cruces vermelhas.



Fonte: Autor

**Figura 7:** Sinal de controle ao longo da trajetória do sistema do exemplo 1 em MF para uma condição inicial de  $x_1 = x_2 = 20$ . A ação de controle do MPC e da RN são representados, respectivamente, por círculos azuis e por cruces vermelhas.



Fonte: Autor

2013), foi da ordem de grandeza de milissegundos, já o tempo necessário para calcular a ação de controle da RN, calculada através de (12) e (13), foi da ordem de grandeza de microssegundos, de forma a cumprir o objetivo de reduzir o custo computacional para o cálculo da ação de controle.

### 4.1.3 Estabilidade Global da Origem

Por fim, utilizando as matrizes do Apêndice C foi certificada a estabilidade exponencial global da origem do sistema em MF utilizando as LMIs (25a), (25b) e (25c) obtidas para  $\eta = 0.0001$  e uma função de Lyapunov quadrática por partes (41). As LMIs foram resolvidas utilizando YALMIP (LÖFBERG, 2004), MOSEK (APS, 2021) e MATLAB. A Matriz  $P = \begin{bmatrix} P_1 & P_2 \\ P_2^\top & P_3 \end{bmatrix}$  obtida é apresentada no Apêndice D.1.

## 4.2 EXEMPLO 2

Considerando um sistema linear a tempo discreto dado por (11) com

$$A = \begin{bmatrix} 0.9210 & 0.000 & 0.0410 \\ 0.0000 & 0.918 & 0.0000 \\ 0.0000 & 0.000 & 0.9240 \end{bmatrix}, \quad B = \begin{bmatrix} 0.0170 & 0.0010 \\ 0.0010 & 0.0230 \\ 0.0000 & 0.0610 \end{bmatrix} \quad (31)$$

e com uma saturação no sinal de controle dada por (13) com  $\bar{u} = [1 \ 2]^\top$  e  $\underline{u} = [-2 \ -4]^\top$ . Para controlar esse sistema, deseja-se implementar um MPC levando em conta as restrições de amplitude (*i.e.* saturação) sobre a entrada de controle. Porém, devido ao potencial alto custo computacional para o cálculo da ação de controle do MPC, será treinada uma RN reduzida para aproximar o comportamento do mesmo. Por fim, para certificar a estabilidade do sistema em MF controlado pela RN reduzida, as condições do Teorema 1 são empregadas.

### 4.2.1 Projeto do MPC

Foi projetado um MPC para o sistema dado por (11) e (31) considerando a saturação assimétrica (13) com  $\bar{u} = [1 \ 2]^\top$  e  $\underline{u} = [-2 \ -4]^\top$ . O MPC resolve o problema QP (2) para calcular o valor da entrada de controle  $u$ , considerando  $Q_x = I_3$ ,  $Q_u = 0.2I_2$ ,  $N_h = 5$ ,  $E = 0_{20,3}$ ,

$$R^\top = \begin{bmatrix} R_1^\top & R_2^\top \end{bmatrix}$$

com

$$R_1^\top = [1 \ 1 \ 1 \ 1 \ 1 \ 2 \ 2 \ 2 \ 2 \ 2] \quad \text{e} \quad R_2^\top = [2 \ 2 \ 2 \ 2 \ 2 \ 4 \ 4 \ 4 \ 4 \ 4]$$

e

$$G^\top = \begin{bmatrix} I_5 & -I_5 & 0_5 & 0_5 \\ 0_5 & 0_5 & I_5 & -I_5 \end{bmatrix}.$$

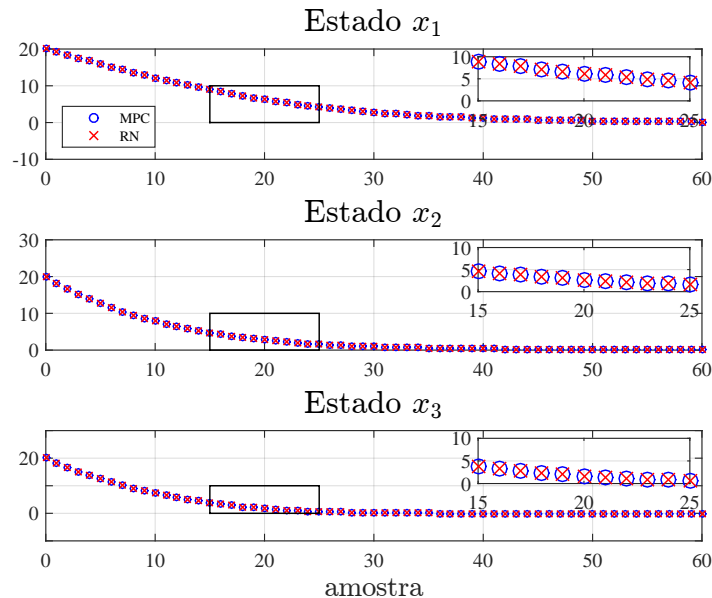
### 4.2.2 RN reduzida

Com o objetivo de reduzir o custo computacional para utilização do MPC projetado, foi treinada uma RN a partir do MPC com objetivo de aproximar o comportamento do mesmo. Foi utilizada uma RN ReLU com 3 entradas, 2 saída e 2 camadas ocultas com o seguinte número de neurônios  $\{5\ 6\}$ . Para treinar essa rede, foram calculadas as saídas do MPC para diferentes estados do sistema. Foi feita uma varredura no espaço de estados do sistema usando um cubo onde os valores de cada um dos estados varia de -5 até 5 com espaço de 0.1 entre eles. Esses valores foram então utilizados para o treinamento da RN com o uso do algoritmo *Levenberg-Marquardt backpropagation* (apresentado no Apêndice B.4) e os mesmos hiperparâmetros e critérios de parada para o treinamento utilizados no exemplo anterior apresentados respectivamente nas tabelas 1 e 2.

Após o treinamento da RN, o valor do *bias* da camada de saída da mesma ( $b_{\ell+1}$ ) foi ajustado para a garantir que saída da RN tenha valor zero para uma entrada nula, conforme (30), assim garantindo que a origem do sistema em MF é um ponto de equilíbrio.

A RN obtida é descrita por (12) com as matrizes apresentadas no Apêndice C.2. A Figura 8 apresenta a comparação de uma trajetória do sistema controlado pelo MPC (2) parametrizado de acordo com a Seção 4.1.1 e pela RN que emula seu comportamento. Na Figura 7 pode-se observar a comparação do sinal de controle gerado pelo MPC com o sinal de controle da RN.

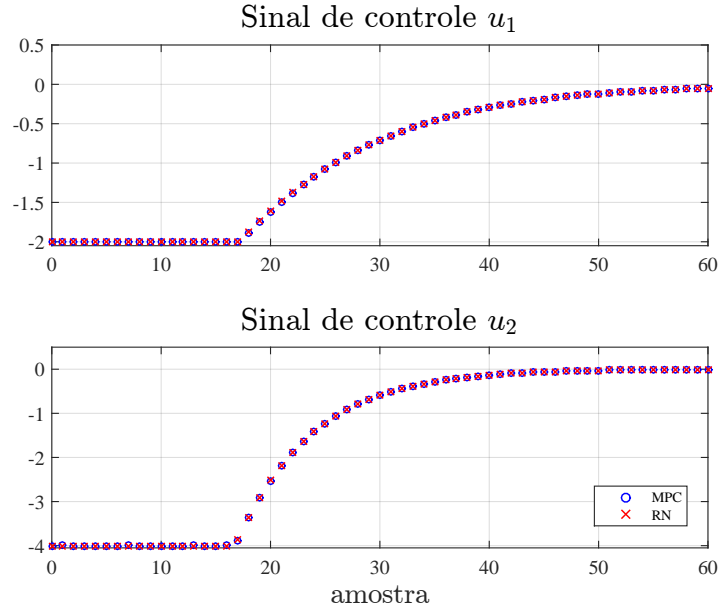
**Figura 8:** Evolução dos estados do sistema do exemplo 2 em MF para uma condição inicial de  $x_1 = x_2 = 20$ . O sistema com controlador MPC é representado pelos círculos azuis, enquanto o sistema com controlador de RN é representado pelas cruces vermelhas.



Fonte: Autor

Pode-se observar que a RN treinada tem o desempenho desejado tendo um desempenho próxima do MPC projetado mesmo para condições iniciais fora do cubo utilizado para o treinamento da mesma. Aqui, diferentemente do exemplo 1, a performance da RN continua muito próxima da performance do MPC inclusive quando o sinal de controle

**Figura 9:** Sinal de controle ao longo da trajetória do sistema do exemplo 2 em MF para uma condição inicial de  $x_1 = x_2 = 20$ . A ação de controle do MPC e da RN são representados, respectivamente, por círculos azuis e por cruces vermelhas.



Fonte: Autor

satura, o que provavelmente ocorre devido à utilização de uma RN com um maior número de neurônios. Também destaca-se que o tempo necessário para calcular a ação de controle do MPC, calculada usando pacote MPT3 (HERCEG et al., 2013), foi da ordem de grandeza de milissegundos, já o tempo necessário para calcular a ação de controle da RN, calculada através de (12) e (13), foi da ordem de grandeza de microssegundos, de forma a cumprir o objetivo de reduzir o custo computacional para o cálculo da ação de controle.

### 4.2.3 Estabilidade Global da Origem

Por fim, utilizando as matrizes do Apêndice C foi certificada a estabilidade exponencial global da origem do sistema em MF utilizando as LMIs (25a), (25b) e (25c) obtidas para  $\eta = 0.0001$  e uma função de Lyapunov quadrática por partes (41). As LMIs foram resolvidas utilizando YALMIP (LÖFBERG, 2004), MOSEK (APS, 2021) e MATLAB. A Matriz  $P = \begin{bmatrix} P_1 & P_2 \\ P_2^\top & P_3 \end{bmatrix}$  obtida é apresentada no Apêndice D.2.



## 5 CONCLUSÃO

Neste trabalho foi abordado o problema da certificação de estabilidade global de um sistema linear a tempo discreto controlado por uma RN ReLU. Para isso, foi utilizada representação implícita baseada em rampas para sistemas afins por partes, a qual foi apresentada no Capítulo 2.

No Capítulo 3, foi desenvolvida a representação implícita baseada em rampas para um sistema linear controlado por RN. Para isto foram desenvolvidas a representação implícita da RN, a representação implícita da saturação e a composição de dois sistemas PWA interligados. Então, com a representação implícita de um sistema linear controlado por RN em mãos, foram derivadas condições LMIs que certificam a estabilidade global da origem do sistema em malha fechada.

No Capítulo 4, como forma de ilustrar a aplicação das condições propostas, foram realizados dois exemplos numéricos onde se utiliza o teorema apresentado para verificar a estabilidade exponencial global da origem de um sistema em malha fechada onde o controlador é uma RN treinada para aproximar a ação de controle de um MPC.

As principais vantagens da abordagem proposta são a redução do custo computacional para o cálculo da ação de controle comparado a métodos como o MPC, a certificação da estabilidade do sistema em malha fechada e a consideração explícita dos limites da ação de controle como uma saturação na saída da RN.

Destaca-se que apesar do trabalho utilizar uma RN treinada para aproximar um MPC para ilustrar o problema, os resultados obtidos neste trabalho são válidos para qualquer controlador que faça uso de RNs ReLU.

Os resultados desse trabalho foram aceitos para apresentação no SBAI 2023 (ZANETTE et al., 2023). Possíveis trabalhos futuros incluem:

- A análise da estabilidade local de sistemas lineares a tempo discreto controlados por RNs ReLU, para isso pode-se utilizar os resultados desenvolvidos em (CABRAL; VALMORBIDA; GOMES DA SILVA JR., a ser submetido);
- A generalização do trabalho para sistemas não-lineares controlados por RNs. possíveis abordagens para esse problema são a utilização de uma modelagem LPV para o sistema não linear, ou considerar sistemas com não linearidades de setor e utilizar as técnicas desenvolvidas para esse tipo de não linearidades;
- A expansão do trabalho para um sistema com seguimento de referência e realimentação da saída, para isso pode-se treinar uma RN para emular um MPC que tenha como entrada as saídas do sistema ao longo do tempo como o descrito em (NGUYEN et al., 2021);
- A expansão do trabalho para outros tipos de RN, como, por exemplo, RNs recorrentes, as quais podem ser utilizadas para aproximar um MPC com um observador de

estados. A expansão do trabalho para RN recorrentes implicaria em uma matriz de representação da RN (17) com diferentes matrizes de pesos  $W_{ij}$  no lugar dos zeros, as quais representariam ligações da camada oculta  $i$  para a camada oculta  $j$  e não apenas ligações de uma camada oculta para a sua sucessora.

## REFERÊNCIAS

- AGGARWAL, C. C. *Neural Networks and Deep Learning: A Textbook*. Berlim: Springer, 2019. P. 520. ISBN 3-03-006856-0.
- ANDRIKOPOULOS, G. et al. Piecewise Affine Modeling and Constrained Optimal Control for a Pneumatic Artificial Muscle. *IEEE Transactions on Industrial Electronics*, v. 61, n. 2, p. 904–916, 2014. DOI: 10.1109/TIE.2013.2254094.
- APS, M. *MOSEK Optimization Toolbox for MATLAB 9.2.49*. [S.l.], 2021. Disponível em: <<https://docs.mosek.com/9.2/toolbox/index.html>>.
- ARFKEN, G. B.; WEBER, H. J. *Mathematical Methods for Physicists, 6th Edition*. New York: Academic Press, 2005. P. 1200. ISBN 0-12-059876-0.
- ASTROM, K. A.; WITTENMARK, B. *Computer-Controlled Systems: Theory and Design*. New York: Dover Publications, 2011. P. 576. ISBN 0-48-648613-0.
- AZUMA, S.-I.; YANAGISAWA, E.; IMURA, J.-I. Controllability Analysis of Biosystems Based on Piecewise-Affine Systems Approach. *IEEE Transactions on Automatic Control*, v. 53, Special Issue, p. 139–152, 2008. DOI: 10.1109/TAC.2007.911316.
- BEMPORAD, A. et al. The explicit linear quadratic regulator for constrained systems. *Automatica*, v. 38, n. 1, p. 3–20, 2002.
- BOSE, B. K. Neural Network Applications in Power Electronics and Motor Drives-An Introduction and Perspective. *IEEE Transactions on Industrial Electronics*, IEEE, v. 54, n. 1, p. 14–33, 2007.
- BOYD, S. et al. *Linear Matrix Inequalities in System and Control Theory*. Philadelphia: SIAM studies in applied mathematics: 15, 1994.
- CABRAL, L.; VALMORBIDA, G.; GOMES DA SILVA JR., J. M. Estabilidade Regional de Sistemas Lineares de Tempo Discreto sob Saturação de Controle Assimétrica. In: XXIV Congresso Brasileiro de Automatica (CBA 2022). Fortaleza: [s.n.], 2022. P. 1–6.
- CABRAL, L.; VALMORBIDA, G.; GOMES DA SILVA JR., J. M. Regional Stability Analysis of Discrete-Time Piecewise Affine Systems. *IEEE Transactions on Automatic Control*, p. 1–12, a ser submetido.
- CAMACHO, E.; BORDONS, C. *Model Predictive Control*. Berlim: Springer, 2007.
- CHUA, L.; KANG, S. M. Section-wise piecewise-linear functions: Canonical representation, properties, and applications. *Proceedings of the IEEE*, v. 65, n. 6, p. 915–929, 1977. DOI: 10.1109/PROC.1977.10589.
- GAVIN, H. P. The Levenberg-Marquardt method for nonlinear least squares curve-fitting problems c ©. In. Disponível em: <<https://api.semanticscholar.org/CorpusID:5708656>>.

- GOMES DA SILVA JR., J. M.; TARBOURIECH, S. Polyhedral regions of local stability for linear discrete-time systems with saturating controls. *IEEE Transactions on Automatic Control*, v. 44, n. 11, p. 2081–2085, 1999. DOI: 10.1109/9.802920.
- GROFF, L. B.; VALMORBIDA, G.; GOMES DA SILVA JR., J. M. An implicit representation for the analysis of piecewise affine discrete-time systems. *Automatica*, v. 147, p. 110730, 2023.
- GROFF, L. B.; VALMORBIDA, G.; GOMES DA SILVA JR., J. M. Stability analysis of piecewise affine discrete-time systems. In: 2019 IEEE 58th Conference on Decision and Control (CDC). Nice: [s.n.], 2019. P. 8172–8177.
- GROFF, L. B. *Event-triggered control for piecewise affine discrete-time systems*. 2020. F. 130. Tese (Doutorado em engenharia) – o Programa de Pós-Graduação em Engenharia Elétrica of Universidade Federal do Rio Grande do Sul e to Ecole Doctorale Sciences et Technologies de l’Information et de la Communication of Université Paris-Saclay, Porto Alegre.
- HAYKIN, S. *Redes Neurais: Princípios e Prática*. São Paulo: Bookman, 2001. P. 898. ISBN 8-57-307718-2.
- HE, J. et al. Relu Deep Neural Networks and Linear Finite Elements. *Journal of Computational Mathematics*, Global Science Press, v. 38, n. 3, p. 502–527, jun. 2020.
- HERCEG, M. et al. Multi-Parametric Toolbox 3.0. In: PROC. OF the European Control Conference. Zürich, Switzerland: [s.n.], jul. 2013. P. 502–510. <http://control.ee.ethz.ch/~mpt>.
- J. -J E. SLOTINE, J.-J. S.; LI, W. *Applied Nonlinear Control*. New Jersey: Prentice Hall, 1990. P. 352. ISBN 0-13-040890-7.
- JAIN, A.; MAO, J.; MOHIUDDIN, K. Artificial neural networks: a tutorial. *Computer*, v. 29, n. 3, p. 31–44, 1996. DOI: 10.1109/2.485891.
- JULIAN, P.; DESAGES, A.; AGAMENNONI, O. High-level canonical piecewise linear representation using a simplicial partition. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, v. 46, n. 4, p. 463–480, 1999. DOI: 10.1109/81.754847.
- KHALIL, H. K. *Nonlinear Systems*. New Jersey: Prentice Hall, 2002. P. 750. ISBN 0-13-067389-7.
- LEE, J. et al. Learning quadrupedal locomotion over challenging terrain. *Science Robotics*, American Association for the Advancement of Science (AAAS), v. 5, n. 47, 2020.
- LI, Z. et al. Missile Guidance Law Based on Robust Model Predictive Control Using Neural-Network Optimization. *IEEE Transactions on Neural Networks and Learning Systems*, v. 26, n. 8, p. 1803–1809, 2015. DOI: 10.1109/TNNLS.2014.2345734.
- LÖFBERG, J. YALMIP : A Toolbox for Modeling and Optimization in MATLAB. In: IN Proceedings of the CACSD Conference. Taipei, Taiwan: [s.n.], 2004.
- MACIEJOWSKI, J. *Predictive Control With Constraints*. London: Pearson, 2000.
- MAN, K.; TANG, K.; KWONG, S. Genetic algorithms: concepts and applications [in engineering design]. *IEEE Transactions on Industrial Electronics*, v. 43, n. 5, p. 519–534, 1996. DOI: 10.1109/41.538609.

- MARQUARDT, D. W. An Algorithm for Least-Squares Estimation of Nonlinear Parameters. *Journal of the Society for Industrial and Applied Mathematics*, Society for Industrial e Applied Mathematics, v. 11, n. 2, p. 431–441, 1963.
- NGUYEN, T. W. et al. Predictive Cost Adaptive Control: A Numerical Investigation of Persistency, Consistency, and Exigency. *IEEE Control Systems Magazine*, v. 41, n. 6, p. 64–96, 2021. DOI: 10.1109/MCS.2021.3107647.
- ROJAS, R. *Neural Networks - A Systematic Introduction*. Berlin: Springer-Verlag, 1996. Disponível em: <[http://www.inf.fu-berlin.de/inst/ag-ki/rojas\\_home/pmwiki/pmwiki.php?n=Books.NeuralNetworksBook](http://www.inf.fu-berlin.de/inst/ag-ki/rojas_home/pmwiki/pmwiki.php?n=Books.NeuralNetworksBook)>.
- ROSS, R. et al. Bounding the Difference Between Model Predictive Control and Neural Networks. In: PROCEEDINGS of The 4th Annual Learning for Dynamics and Control Conference. Palo Alto: [s.n.], 2022. v. 168. (Proceedings of Machine Learning Research), p. 817–829.
- RUMELHART, D. E.; HINTON, G. E.; WILLIAMS, R. J. Learning representations by back-propagating errors. *Nature*, v. 323, p. 533–536, 1986. Disponível em: <<https://api.semanticscholar.org/CorpusID:205001834>>.
- SANGER, T.; BALJEKAR, P. N. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, v. 65 6, p. 386–408, 1958. Disponível em: <<https://api.semanticscholar.org/CorpusID:12781225>>.
- SONTAG, E. Nonlinear regulation: The piecewise linear approach. *IEEE Transactions on Automatic Control*, v. 26, n. 2, p. 346–358, 1981. DOI: 10.1109/TAC.1981.1102596.
- TESU, I.; DARTU, F. A comment on 'A generalized canonical PWL representation'. *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, v. 39, n. 12, p. 885–887, 1992. DOI: 10.1109/82.208588.
- VALMORBIDA, G.; HOVD, M. Quadratic programming with ramp functions and fast online QP-MPC solutions. *Automatica*, v. 153, p. 111011, 2023.
- YIN, H.; SEILER, P.; ARCAK, M. Stability Analysis Using Quadratic Constraints for Systems With Neural Network Controllers. *IEEE Transactions on Automatic Control*, v. 67, n. 4, p. 1980–1987, 2022.
- ZANETTE, D. V. et al. Análise de estabilidade de sistemas lineares controlados por redes neurais emulando MPC. In: SIMPÓSIO Brasileiro de Automação Inteligente (SBAI 2023). Amazonas: [s.n.], 2023.
- ZHOU, Q. et al. Adaptive Neural Network Tracking Control for Robotic Manipulators With Dead Zone. *IEEE Transactions on Neural Networks and Learning Systems*, IEEE, v. 30, n. 12, p. 3611–3620, 2019.

# Apêndices

## Apêndice A - ESTABILIDADE NO SENTIDO DE LYAPUNOV

Uma das características mais importantes para qualquer sistema é a sua estabilidade, tendo em vista que sistemas instáveis, em geral, tem pouca utilidade e muitas vezes podem ser perigosos (J. -J E. SLOITINE; LI, 1990). A principal ferramenta para a análise de estabilidade de sistemas não lineares é a utilização dos métodos de Lyapunov (KHALIL, 2002). Desta forma, neste apêndice, é revisado o conceito de estabilidade no sentido de Lyapunov para sistemas a tempo discreto com base no exposto em (ASTROM; WITTENMARK, 2011).

A estabilidade de um sistema baseado na teoria de Lyapunov está relacionada com os pontos de equilíbrio do mesmo. Assim, considerando a seguinte sistema de tempo discreto:

$$x^+ = g(x). \quad (32)$$

Um ponto de equilíbrio pode ser definido como:

**Definição 1.** Um estado  $x^*$  é considerado um ponto de equilíbrio do sistema se, uma vez que  $x = x^*$  em um instante  $k = k_1$ ,  $x$  permanece igual a  $x^*$  para qualquer instante seguinte, ou seja, temos que  $g(x^*) = x^* \quad \forall k \geq k_1$ .

Normalmente, por conveniência, é considerado que o ponto de equilíbrio do sistema é a origem do mesmo, ou seja  $x^* = 0$ . Quando isso não ocorre, o ponto pode ser transferido para a origem através de uma mudança de variáveis.

**Definição 2.** Seja  $x = 0$  um ponto de equilíbrio do sistema (32). Ele é

- considerado estável se para todo  $\epsilon > 0$  existe  $\gamma > 0$  tal que

$$\|x_0\| < \gamma \Rightarrow \|x\| < \epsilon, \quad \forall k > 0; \quad (33)$$

- considerado instável se não for estável;
- considerado assintoticamente estável se for estável e  $\gamma$  pode ser escolhido de forma que

$$\|x_0\| < \gamma \Rightarrow \lim_{k \rightarrow \infty} \|x\| = 0 \quad (34)$$

Esta definição de estabilidade, chamada de estabilidade no sentido de Lyapunov, significa que, se um ponto de equilíbrio é estável, as trajetórias do sistema podem ser mantidas arbitrariamente próximas ao ponto de equilíbrio caso comecem suficientemente próximas dele. Uma forma de verificar a estabilidade no sentido de Lyapunov de um sistema, consiste em utilizar uma função escalar dos estados do sistema  $V$ . Se essa função for positiva e decrescente no tempo em uma região em torno do ponto de equilíbrio, as trajetórias irão se aproximar deste ponto e ele será assintoticamente estável. Este é o teorema da estabilidade Lyapunov, o qual é formalizado a seguir.

**Teorema 2.** *Seja  $x = 0$  um ponto de equilíbrio do sistema (32) e  $\mathcal{D} \in \mathbb{R}^n$  é um domínio contendo  $x = 0$ , o ponto de equilíbrio é considerado estável se existe uma função escalar  $V : \mathcal{D} \rightarrow \mathbb{R}$  contínua em  $x$ , tal que*

$$\begin{aligned} V(0) &= 0 \\ V(x) &> 0, \quad \forall x \neq 0 \in \mathcal{D} \\ \Delta V(x) &= V(x^+) - V(x) \leq 0, \quad \forall x \in \mathcal{D} \end{aligned} \quad (35)$$

*Além disso,  $x = 0$  é considerado assintoticamente estável se*

$$\Delta V(x) < 0, \quad \forall x \neq 0 \in \mathcal{D}. \quad (36)$$

Tanto a estabilidade quanto a estabilidade assintótica podem ser consideradas globais se  $\mathcal{D} = \mathbb{R}^n$  e  $V(x)$  for radialmente ilimitado, ou seja

$$\lim_{\|x\| \rightarrow \infty} V(x) = \infty. \quad (37)$$

Uma função  $V$  que satisfaz as condições (35) é chamada de função de Lyapunov e de maneira geral podem existir diferentes funções de Lyapunov para um mesmo sistema. É importante destacar que as condições do Teorema 2 são apenas suficientes e não necessárias para a estabilidade da origem do sistema. Caso uma função candidata verifique as condições do teorema o ponto de equilíbrio é considerado estável, mas caso a função candidata não verifique as condições do teorema nada pode ser afirmado.

Uma classe de funções escalares normalmente utilizadas como candidatas de Lyapunov são as funções quadráticas, dadas por

$$V(x) = x^\top P x, \quad (38)$$

onde  $P$  é uma matriz simétrica positiva definida de forma a satisfazer  $V(x) > 0$ .

Segundo, (GROFF; VALMORBIDA; GOMES DA SILVA JR., 2023) uma extensão das funções quadráticas, que deriva condições de estabilidade menos conservadoras que essas para sistemas PWA devido a estas incorporarem informações sobre as regiões do sistema, são função quadráticas por partes (PWQ, do inglês *Piecewise Quadratic*).

Considerando um sistema PWA na sua representação padrão dado por

$$f(x) = A_i x + b_i \quad \forall x \in \Gamma_i \subset \mathbb{R}^n, i = 1, \dots, L, \quad (39)$$

onde  $L$  é o número de regiões do sistema e  $\Gamma_i$  é um poliedro convexo que representa uma das  $L$  regiões do sistema, pode-se usar uma função PWQ candidata de Lyapunov como

$$V(x) = V_i(x) = x^\top P_i x \quad \forall x \in \Gamma_i, i = 1, \dots, L, \quad (40)$$

onde  $P_i$  são matrizes simétricas positiva definidas de forma a satisfazer  $V(x) > 0$ .

Nota-se que as função PWQ também podem ser escritas utilizando uma representação implícita baseada em rampas de forma que

$$V(x) = \begin{bmatrix} x \\ \phi(y(x)) \end{bmatrix}^\top \begin{bmatrix} P_1 & P_2 \\ P_2^\top & P_3 \end{bmatrix} \begin{bmatrix} x \\ \phi(y(x)) \end{bmatrix}. \quad (41)$$

onde  $P_1 = P_1^\top \in \mathbb{R}^{n \times n}$ ,  $P_2 \in \mathbb{R}^{n \times n_y}$  e  $P_3 = P_3^\top \in \mathbb{R}^{n_y \times n_y}$ .



## A.1 LINEAR MATRIX INEQUALITIES

Segundo (BOYD et al., 1994), referência usada para o descrito nessa sessão, pode-se reduzir uma grande variedade de problemas na teoria de área de controle de sistemas (como por exemplo a verificação do Teorema de Lyapunov) a alguns problemas de otimização convexos ou quase convexos envolvendo desigualdades matriciais lineares (do inglês, LMI).

Uma LMI, tem a seguinte forma

$$F(x) = F_0 + \sum_{i=1}^m x_i F_i > 0, \quad (42)$$

onde  $x \in \mathbb{R}$  é um vetor de variáveis a serem computadas e as matrizes  $F_i = F_i^\top \in \mathbb{R}^{n \times n}$ ,  $i = 0, \dots, m$  são conhecidas. Uma LMI é equivalente a um conjunto de desigualdades polinomiais e pode ter a forma de uma desigualdade estrita como em (42) ou não estrita  $F(x) \geq 0$ . É importante observar que (42) é uma restrição convexa em  $x$ , ou seja, o conjunto  $\{x | F(x) > 0\}$  é convexo de forma que uma LMI pode ser formulada como um problema de otimização convexo passível de solução computacional e conseqüentemente pode-se resolver de forma confiável muitas LMIs para as quais nenhuma “solução analítica” foi encontrada (ou provavelmente será encontrada) (BOYD et al., 1994).

## Apêndice B - TREINAMENTO DE RN

Para utilizar uma RN de forma proveitosa é preciso ajustar os valores de seus pesos e *bias* para que a mesma gere os valores desejados, este processo de ajuste de pesos e *bias* é chamado de treinamento da RN.

Existem duas principais classificações de métodos de treinamento de RNs, aprendizado supervisionado e aprendizado não supervisionado, sendo o aprendizado supervisionado ainda dividido em aprendizado por reforço e aprendizado corretivo. Nesta seção focaremos em métodos de aprendizado corretivo já que estes são os mais utilizados para o treinamento de RN *feedforward*, as quais são o escopo deste trabalho.

Em aprendizado supervisionado são utilizados dados rotulados, conjuntos de entradas e saídas com os valores esperados da RN. No aprendizado corretivo, a RN é alimentada com esses dados e, a partir de uma análise do valor do erro  $e$  da saída a RN

$$e_{(i)} = v_{(i)} - v_{(i)}^* \quad i = 1, \dots, n_v \quad (43)$$

onde  $v$  é a saída de RN e  $v^*$  é o valor esperado, o algoritmo realiza correções nos valores dos pesos e *bias* da RN em cada iteração (chamada de época) do mesmo até que ocorra algum critério de parada do treinamento (número máximo de épocas, erro mínimo, mínimo ganho de performance entre as épocas, entre outros).

Por fim, com o objetivo de verificar se a RN treinada foi capaz de generalizar o aprendido e gerar os valores esperados para novos dados não utilizados no treinamento da mesma, costuma-se dividir os dados disponíveis para o treinamento da RN em dois grupos: dados de treinamento e dados de teste (normalmente 70% dos dados são utilizados para treinamento e 30% para teste, mas essa divisão é um hiperparâmetro do treinamento da RN). Deste forma os dados de treinamento são utilizados para treinar a RN e os dados de teste são utilizado para analisar o “real” desempenho da mesma, verificando se ela realmente aprendeu o que gostaria-se ou se ela apenas “decorou” os dados de treinamento (fenômeno chamado de *overfitting*).

### B.1 BACKPROPAGATION

O principal método de treinamento corretivo de RN é conhecido como *backpropagation* (RUMELHART; HINTON; WILLIAMS, 1986). Este algoritmo busca reduzir o erro da saída da RN através da retro-propagação do mesmo para suas camadas internas. Assim o algoritmo de *Backpropagation* atualiza os valores dos pesos e *bias* da RN a cada iteração sua de acordo com a parcela de erro de cada um dos neurônios, propagando o erro medido na saída da RN para os pesos e *bias* de cada um dos neurônios da mesma. Desta forma,

considerando a representação matemática da RN dada em (3), a atualização das matrizes de pesos ( $W_i$ ) e *bias* ( $b_i$ ) de uma camada  $i$  pode ser dada por:

$$W_i = W_i - \alpha \delta_i w_{i-1}^\top \quad (44)$$

$$b_i = b_i - \alpha \delta_i \quad (45)$$

onde  $\alpha \in \mathbb{R}$  é o parâmetro de velocidade de aprendizado e  $\delta_i \in \mathbb{R}^{n_{w_i}}$  é o vetor do erro retro-propagado até a camada  $i$ . Nas próximas seções serão explicados alguns dos métodos mais comuns para o cálculo desse erro retro-propagado.

De maneira geral o algoritmo de *Backpropagation*, tem as seguintes fases:

1. Inicialização dos pesos da RN (normalmente de forma aleatória);
2. Cálculo da saída da RN a partir das entradas do treinamento;
3. Cálculo do erro da RN a partir da comparação dos valores da saída da RN com os valores esperados pelos dados de treinamento;
4. Propagação do erro para as camadas internas da RN;
5. Atualização dos pesos e *bias* da RN de acordo com o erro retro-propagado de cada camada buscando diminuir o mesmo;
6. Repetição dos passos de 2 a 5 até que algum dos critérios de parada de treinamento seja alcançado.

## B.2 MÉTODO DO GRADIENTE DESCENDENTE

Um dos principais métodos para a atualização dos parâmetros da RN por *Backpropagation*, o qual é amplamente utilizado e serve de base para muitos dos métodos mais sofisticados é o método do gradiente descendente. Nele o erro retro-propagado até a  $i$ -ésima camada  $\delta_i$  é calculado pelo gradiente do mesmo (ROJAS, 1996)

$$\delta_i = J_i^\top e, \quad (46)$$

onde  $J_i$  é a matriz jacobiana da  $i$ -ésima camada, dada por

$$J_i = \frac{\partial q_{i+1}}{\partial q_i} J_{i+1} = \frac{\partial q_{i+1}}{\partial w_i} \frac{\partial w_i}{\partial q_i} J_{i+1} = W_{i+1} \text{diag}(\phi'(q_i)) J_{i+1} \quad \text{e} \quad (47)$$

$$J_{\ell+1} = \frac{\partial v}{\partial q_{\ell+1}} = \text{diag}(\psi'(q_{\ell+1})),$$

onde

$$q_i = W_i w_{i-1} + b_i. \quad (48)$$

### B.3 MÉTODO DE GAUSS-NEWTON

Uma evolução do método do gradiente descendente é o método de Newton, o qual usa uma expansão de Taylor de segunda ordem da função de ativação para calcular  $\delta_i$

$$\delta_i = H_i^{-1} J_i^\top e. \quad (49)$$

Realizar o cálculo da matriz Hessiana para RNs *feedforward* é complexo e custoso computacionalmente. Como forma de contornar esse problema, o método de Gauss-Newton aproxima a matriz Hessiana por (ROJAS, 1996)

$$H_i = J_i^\top J_i, \quad (50)$$

desta forma o erro retro-propagado pode ser dado de forma recorrente por

$$\delta_i = (J_i^\top J_i)^{-1} J_i^\top e. \quad (51)$$

### B.4 MÉTODO DE LEVENBERG-MARQUARDT

Um dos métodos de atualização dos parâmetros de RNs considerado um dos métodos mais rápidos para o treinamento de RN *feedforward* de tamanho moderado é o algoritmo de Levenberg-Marquardt (MARQUARDT, 1963). Este método varia adaptativamente as atualizações dos parâmetros entre a atualização por gradiente descendente e a atualização de Gauss-Newton, através de um parâmetro de amortecimento  $\sigma$  (GAVIN, 2013). Assim, o valor do erro propagado para a camada anterior é dado de forma recorrente por

$$\delta_i = (J_i^\top J_i + \sigma I)^{-1} J_i^\top e, \quad (52)$$

de forma que quando  $\sigma = 0$  o método de Levenberg-Marquardt é equivalente ao método de Gauss-Newton e quando  $\sigma$  é muito grande o método de Levenberg-Marquardt é equivalente ao método do gradiente descendente. A ideia é utilizar o método do gradiente descendente quando o erro da RN estiver longe de um ponto mínimo, onde o erro de aproximação da matriz Hessiana pode ter mais efeito, e, conforme o erro se aproxima de um ponto mínimo, ir transicionado para o método de Gauss-Newton já que este é considerado mais rápido e preciso que o método do gradiente descendente (GAVIN, 2013). Desta forma, o algoritmo de Levenberg-Marquardt funciona da seguinte forma:

- O valor de  $\sigma$  é inicialmente alto;
- Quando uma iteração do *Backpropagation* resulta em um menor valor para função de desempenho, o valor de  $\sigma$  é reduzido e os pesos são atualizados;
- Quando uma iteração do *Backpropagation* resulta em um maior valor para função de desempenho, o valor de  $\sigma$  é aumentado e os pesos não são atualizados.

Desta forma, o método de Levenberg-Marquardt inicia próximo do método do gradiente descendente e se aproxima do método de Gauss-Newton ao se aproximar do ponto mínimo, assim acelerando até o mínimo local.

## Apêndice C - MATRIZES DA RN

### C.1 EXEMPLO 1

$$W_1 = \begin{bmatrix} 1.0445 & 0.5765 \\ -0.3736 & 0.1259 \end{bmatrix}, \quad W_2 = \begin{bmatrix} 0.3493 & 0.0131 \\ 1.4562 & -0.5387 \end{bmatrix}, \quad W_3 = [1.8936 \quad -0.4543],$$

$$b_1 = \begin{bmatrix} 2.1176 \\ -2.7835 \end{bmatrix}, \quad b_2 = \begin{bmatrix} -1.7966 \\ -0.8850 \end{bmatrix}, \quad b_3 = 0.9987.$$

### C.2 EXEMPLO 2

$$W_1 = \begin{bmatrix} 0.3141 & 0.3595 & 0.9386 \\ 0.2753 & 0.2846 & -0.7928 \\ 0.0261 & -0.0233 & -0.0494 \\ 0.3716 & 0.0195 & 0.0342 \\ 0.0247 & 0.1236 & 0.3390 \end{bmatrix},$$

$$W_2 = \begin{bmatrix} -0.3501 & -0.8820 & -0.2053 & 0.4176 & -0.2186 \\ 0.5329 & 0.6324 & 0.6689 & 1.5067 & -1.0365 \\ -0.1514 & -0.4387 & 1.2276 & 0.1016 & -0.6413 \\ 0.2337 & -0.0236 & -1.0584 & 1.3184 & -0.4586 \\ -0.4352 & 0.7806 & 0.7412 & 0.1597 & 1.2993 \\ -0.2614 & 0.7720 & 0.0271 & 0.1683 & -0.2683 \end{bmatrix},$$

$$W_3 = \begin{bmatrix} 0.0028 & 0.0002 & -0.2239 & -0.4732 & -0.2551 & 0.1171 \\ 0.0004 & 0.0007 & 0.6239 & -0.1034 & -1.2361 & 1.6013 \end{bmatrix},$$

$$b_1 = \begin{bmatrix} 1.2488 \\ -0.2518 \\ 0.1790 \\ 1.5078 \\ 0.8362 \end{bmatrix}, \quad b_2 = \begin{bmatrix} 0.7744 \\ -1.8967 \\ 2.3672 \\ 0.4077 \\ 1.4410 \\ 1.8960 \end{bmatrix}, \quad b_3 = \begin{bmatrix} 1.8631 \\ -0.6930 \end{bmatrix}$$

## Apêndice D - MATRIZ P

### D.1 EXEMPLO 1

$$P_1 = \begin{bmatrix} 13.7229 & 6.7768 \\ 6.7768 & 5.3596 \end{bmatrix}, \quad P_2^\top = \begin{bmatrix} 0.9840 & 0.5517 \\ 0.0767 & 0.0099 \\ 6.1991 & 3.2368 \\ -1.2966 & -0.6940 \\ 2.3240 & 1.2170 \\ -2.5178 & -1.2305 \\ 3.2880 & 1.8005 \end{bmatrix},$$

$$P_3 = \begin{bmatrix} -0.6861 & 0.1372 & -2.4933 & 0.9644 & -0.0649 & 0.8271 & -1.9503 \\ 0.1372 & 1.5509 & 0.3641 & -0.5836 & 0.0694 & -0.6583 & -0.6766 \\ -2.4933 & 0.3641 & 0.1354 & 0.6501 & 3.5684 & 0.1569 & -0.4453 \\ 0.9644 & -0.5836 & 0.6501 & -0.9828 & -1.3189 & -0.8260 & 0.4245 \\ -0.0649 & 0.0694 & 3.5684 & -1.3189 & 4.9215 & -0.9800 & 0.1006 \\ 0.8271 & -0.6583 & 0.1569 & -0.8260 & -0.9800 & -0.5372 & -0.5376 \\ -1.9503 & -0.6766 & -0.4453 & 0.4245 & 0.1006 & -0.5376 & -2.0095 \end{bmatrix}.$$

### D.2 EXEMPLO 2

$$P_1 = \begin{bmatrix} 83.5509 & 9.9806 & -6.5458 \\ 9.9806 & 164.9132 & -0.0820 \\ -6.5458 & -0.0820 & 177.4515 \end{bmatrix},$$

$$P_2^\top = \begin{bmatrix} 0.0368 & -0.0634 & -0.3695 \\ -6.6937 & -6.3379 & 18.4951 \\ -0.1122 & 1.5784 & -1.9121 \\ 2.5984 & -0.3941 & -1.0942 \\ -1.2382 & 1.5603 & 0.5778 \\ -0.2576 & -0.0096 & 0.6076 \\ -0.5725 & -0.1656 & 1.3526 \\ 0.0863 & 0.7508 & 2.6212 \\ -0.6225 & 0.3271 & 0.1755 \\ -0.4769 & -1.8599 & -4.0551 \\ 0.6345 & 1.1073 & 4.9735 \\ 0.4451 & -0.2225 & -2.3551 \\ -0.7991 & 0.4915 & 0.0949 \\ -0.3397 & -0.2084 & -3.6179 \\ 0.6242 & 0.3463 & -0.6208 \\ 1.0152 & 1.1700 & 2.5735 \end{bmatrix}, \quad P_3 = \begin{bmatrix} P_{3a} & P_{3b} \\ P_{3b}^\top & P_{3c} \end{bmatrix},$$

$$P_{3a} = \begin{bmatrix} 0.9415 & 2.5051 & -0.1143 & 0.3682 & 1.9414 & 1.1892 & 0.1527 & -0.2435 \\ 2.5051 & 42.5335 & -2.8889 & -1.3879 & 1.4313 & 3.2236 & 1.0373 & 1.7005 \\ -0.1143 & -2.8889 & -19.5318 & 0.2663 & 0.3383 & 0.1562 & -0.2538 & -2.1241 \\ 0.3682 & -1.3879 & 0.2663 & -4.0502 & 2.0917 & -0.2759 & 1.6811 & 2.2900 \\ 1.9414 & 1.4313 & 0.3383 & 2.0917 & 0.3261 & -0.4346 & -2.3686 & 0.7756 \\ 1.1892 & 3.2236 & 0.1562 & -0.2759 & -0.4346 & 4.5493 & -0.4847 & -0.4106 \\ 0.1527 & 1.0373 & -0.2538 & 1.6811 & -2.3686 & -0.4847 & -1.8221 & -0.6131 \\ -0.2435 & 1.7005 & -2.1241 & 2.2900 & 0.7756 & -0.4106 & -0.6131 & 4.6442 \end{bmatrix},$$

$$P_{3b} = \begin{bmatrix} -2.9151 & -0.0984 & 0.0755 & -0.2932 & -0.9229 & 0.4295 & 0.1315 & -1.1382 \\ -0.1991 & 0.1307 & 1.5765 & -0.0387 & -0.6534 & -1.1549 & -0.5074 & -0.8544 \\ 8.7130 & -1.7487 & -1.4003 & 1.3990 & -0.7357 & -0.1565 & 0.4052 & 0.3402 \\ -10.5547 & 0.6394 & -1.5703 & 1.7467 & 0.7316 & 0.9662 & 1.3601 & -0.5687 \\ 2.0771 & -2.2706 & 3.5235 & -1.9218 & -0.9326 & -0.6267 & -0.1284 & 1.1483 \\ -0.7300 & -0.3045 & 0.3876 & -0.5343 & -1.0960 & -0.4639 & -0.2031 & -0.0407 \\ -0.0676 & 0.2887 & 1.5742 & -2.1478 & -0.0940 & -0.8197 & -1.1758 & 0.3761 \\ -1.4279 & -1.6869 & 0.0320 & -1.0490 & -0.7873 & -1.0537 & -0.0620 & 1.0975 \end{bmatrix},$$

$$P_{3c} = \begin{bmatrix} 15.8576 & -0.0999 & -0.5604 & -0.0518 & -1.5949 & 0.6929 & -0.2854 & 0.4541 \\ -0.0999 & 2.9069 & -1.9643 & -0.2449 & -1.2982 & 2.3589 & -0.9912 & -0.6575 \\ -0.5604 & -1.9643 & 2.7131 & -0.1458 & -0.2999 & -3.1377 & 0.4359 & -0.9492 \\ -0.0518 & -0.2449 & -0.1458 & -2.6895 & 3.2045 & 1.1189 & 1.4578 & 1.2685 \\ -1.5949 & -1.2982 & -0.2999 & 3.2045 & -4.4482 & -0.4154 & -0.4174 & 0.0131 \\ 0.6929 & 2.3589 & -3.1377 & 1.1189 & -0.4154 & 2.8902 & -0.4273 & 0.1006 \\ -0.2854 & -0.9912 & 0.4359 & 1.4578 & -0.4174 & -0.4273 & 1.4467 & 0.5503 \\ 0.4541 & -0.6575 & -0.9492 & 1.2685 & 0.0131 & 0.1006 & 0.5503 & -0.8160 \end{bmatrix}.$$