

Escola Politécnica da Universidade de São Paulo

Pedro Simacek

6846130

**Controlador Neural como Método Integrativo de
Arquiteturas Clássicas de Controle
em Sistemas Flexíveis de Distribuição de Água em Canais**

São Paulo

2015

Escola Politécnica da Universidade de São Paulo

Pedro Simacek

6846130

Controlador Neural como Método Integrativo de Arquiteturas Clássicas de Controle em Sistemas Flexíveis de Distribuição de Água em Canais

**Monografia apresentada no
Departamento de Engenharia
Mecatrônica e Sistemas Mecânicos
da Escola Politécnica da
Universidade de São Paulo para
obtenção do título de Engenheiro.**

**Área de Concentração:
Engenharia Mecatrônica**

**Orientador: Professor Doutor
Newton Maruyama**

**São Paulo
2015**

Declaração de Originalidade

Este relatório é apresentado como requisito parcial para obtenção do título de Engenheiro na Escola Politécnica da Universidade de São Paulo. É o produto do nosso próprio trabalho, exceto onde indicado no texto. O relatório pode ser livremente copiado e distribuído desde que a fonte seja citada.

São Paulo, 19 de novembro de 2015

A handwritten signature in blue ink, appearing to read "Pedro Simacek".

Pedro Simacek

6846130

Agradecimento

Pelos meios e encantos,
No enlace e no enroscos,
Ao professor Newton pela oportunidade e assistência,
À minha esposa, a quem supriu me apoio, amor e dedicação,
Aos meus pais, pela vida e ensinamentos,
A vocês que foram vitais,
E sem os quais,
Nada seria possível.

Catalogação-na-publicação

Simacek, Pedro

Controlador Neural como Método Integrativo de Arquiteturas Clássicas
de Controle em Sistemas Flexíveis de Distribuição de Água em Canais / P.
Simacek -- São Paulo, 2015.

45 p.

Trabalho de Formatura - Escola Politécnica da Universidade de São
Paulo. Departamento de Engenharia Mecatrônica e de Sistemas Mecânicos.

1.Sistemas de Transporte de Água 2.Redes Neurais 3.Controle Clássico
4.Canal de Irrigação 5.Controle Proporcional-Integral I.Universidade de São
Paulo. Escola Politécnica. Departamento de Engenharia Mecatrônica e de
Sistemas Mecânicos II.t.

Resumo

Os sistemas de controle atualmente empregados ao redor do mundo vem passando por um processo de diversificação e evolução com o surgimento de novas tecnologias e teorias, e.g., o controle moderno. A teoria de redes neurais é uma revolução em diversas áreas de conhecimento. Desde a identificação de sistemas, previsão de funções temporais até análises de correlação, esta teoria está se inserindo no cotidiano acadêmico e empresarial cada dia mais. Muito se tem usado este pensamento em controle, principalmente em controle adaptativo, e o uso de controladores neurais é onde se situa este trabalho. Baseando-se em um canal de irrigação e em suas arquiteturas habituais de controladores de nível, estipulou-se um sistema de controle neural que, treinado em ambas as arquiteturas habituais, apresentou peculiaridades de ambas as arquiteturas, bem como melhorias comprovadas dado o benchmark utilizado para sua avaliação. Este estudo confirma a capacidade das redes neurais de assimilar correlações não-explicícitas de modo eficaz para o caso das arquiteturas usuais de controle em canais de irrigação.

Palavras-chave: Sistemas de Transporte de Água, Redes Neurais, Controle Clássico, Canal de Irrigação, Controle Proporcional-Integral

Abstract

Control systems currently in use around the world has been undergoing a process of diversification and evolution with the emergence of new technologies and theories, e.g., the modern control. The theory of neural networks is a revolution in various fields of knowledge. From systems' identification, forecasting temporal functions until correlation analysis, this theory is entering the academic and business world every day. This thought is largely use in control, especially in adaptive control, and the use of neural controllers is the main approach of this work. Based on an irrigation channel and in their usual architectures of level controllers, it has set up a neural control system, trained on both the usual architectures, which presented peculiarities of both architectures, as well as proven improvements since the benchmark used for its evaluation. This study confirms the ability of neural networks to assimilate non-explicit correlations effectively to the case of the usual architectures of control in irrigation canals.

Keywords: Water Transportation Systems, Neural Networks, Classic Controller, Irrigation Canal, Proportional-Integral Control

Sumário

1 Introdução.....	1
1.1 Motivação	2
1.2 Metodologia	2
1.3 Descrição sumária do trabalho	3
2 Fundamentação teórica.....	5
2.1 Redes neurais	5
2.2.1 Funções de ativação	7
2.2.2 Arquiteturas comuns.....	8
2.2.3 Aprendizagem e treinamento.....	9
2.2 Simulação e avaliação	10
3 Desenvolvimento do controlador	15
3.1 Requisitos	15
3.2 Definição de redes neurais.....	16
3.3 O processo de treinamento.....	19
3.4 Implementação no simulador	22
3.5 Simulação.....	23
4 Resultados.....	24
4.1 Os controladores	24
4.2 Análise da resposta ao tempo.....	25
4.3 Indicadores de desempenho.....	27
4.3.1 η_{ad}.....	27
4.3.2 η_e.....	28
4.3.3 N_m e N_r	29
4.4 Discussão	34
5 Conclusão	35
Referências	36

Sumário de Figuras

Figura 1: Modelo de neurônio (Haykin, 2001)	5
Figura 2: Rede Neural Multicamadas genérica (MLP) (Haykin, 2001).....	6
Figura 3: Função de Ativação Limiar (Haykin, 2001)	7
Figura 4: Função de Ativação Linear por Partes (Haykin, 2001)	7
Figura 5: Função de Ativação Sigmoide (Haykin, 2001).....	8
Figura 6: (a) Rede Feedfoward (b) Rede Feedback (Haykin, 2001).....	9
Figura 7: Arquitetura de Controle Local a Jusante (Simacek, 2015)	11
Figura 8: Arquitetura de Controle Distante a Montante (Simacek, 2015)	12
Figura 9: Cenários. a) SP, Abertura Sincronizada com Influência Pontual Máxima, b) SS, Abertura Sincronizada com Influência Sequencial, c) EP, Abertura Estática com Influência Pontual Máxima, d) ES, Abertura Estática com Influência Sequencial (Simacek, 2015)	14
Figura 10: Estrutura neural de um controlador PI (Adaptado Ferrari, 2010)	17
Figura 11: Arquitetura das redes neurais de controle das comportas internas do canal (Adaptado Ferrari, 2010).....	18
Figura 12: Novo controlador neural com anti-wind up	22
Figura 13: Resposta temporal ao ensaio SS em amplitude máxima de extração	25
Figura 14: Detalhe da ação CDM e CLJ nos controladores	26
Figura 15: η_{ad} para o controlador desenvolvido.....	27
Figura 16: Resultados do η_{ad} para as arquiteturas CDM e CLJ (Simacek, 2015).....	28
Figura 17: η_e para o controlador neural	28
Figura 18: Resultados do η_e para os controladores originais (Simacek, 2015)	29
Figura 19: η_m e η_r para a piscina 1 com controlador neural	30
Figura 20: η_m e η_r para a piscina 2 com controlador neural	30
Figura 21: η_m e η_r para a piscina 3 com controlador neural	31
Figura 22: η_m e η_r para a piscina 4 com controlador neural	31
Figura 23: η_m e η_r para a piscina 4 com CLJ (Simacek, 2015)	32
Figura 24: η_m e η_r para a piscina 1 com CLJ (Simacek, 2015)	32
Figura 25: η_m e η_r para a piscina 1 com CDM (Simacek, 2015)	33
Figura 26: η_m e η_r para a piscina 4 com CDM (Simacek, 2015)	33

Sumário de Tabelas

Tabela 1: Valores originais dos controladores (Simacek, 2015) e os obtidos com o treinamento da rede neural.....	19
Tabela 2: Dados dos controladores originais (Simacek, 2015) e dados estatísticos.	20
Tabela 3: Pesos dos controladores neurais encontrados	22
Tabela 4: Variação dos parâmetros entre os originais e os treinados	24

Tabela de Siglas

RNA	Rede Neural Artificial
NARX	<i>Nonlinear autorregressive with exogenous input</i>
CLJ	Controle Local a Jusante
CDM	Controle Distante a Montante
P	Proporcional
I	Integral
PI	Proporcional-Integral

1 Introdução

O processo de renovação tecnológica, com o avanço da ciência e da globalização, um dos principais processos ocorrendo no mundo. Apesar de, historicamente, a tecnologia se renovar continuamente, desde a revolução industrial a taxa dessa renovação anda em passos largos e crescentes.

Certas indústrias, entretanto, que outrora eram o berço das novas tecnologias, hoje estão emperradas no século passado. Um bom exemplo destes ramos são alguns sistemas de transportes de água.

Ainda é notória a tecnologia romana na construção de seus imponentes aquedutos, inclusive, sendo ainda mais majestoso o fato de muitos destes ainda estarem em funcionamento. Por vezes a humanidade se fez reinar a natureza com eclusas, canais, transposições e, talvez, por tantos trunfos, houve medo de se deixar a glória do passado em nome do progresso.

Em específico no caso de canais de irrigação, os quais foram vastamente implementados desde a Mesopotâmia, foram os responsáveis pela evolução humana, abrindo caminho para a conquista de zonas áridas e provendo maior capacidade da agricultura.

Contudo, hoje no mundo ainda existem muitos canais de irrigação operando com processos antiquados de controle. Enquanto outros sistemas semelhantes evoluíram rapidamente, como o controle de eclusas e barragens, pouco se fez na irrigação.

Apesar disso, graças ao pensamento em prol do meio ambiente, tais áreas da indústria começaram a investir na sua atualização tecnológica. O sensoriamento dos canais, bem como a capacidade de se controlar as comportas se fez necessidade e, ao notarem os ganhos que o sistema apresentava com tais implementações, a evolução começou a ser dispersada.

Com base nesta premissa, este trabalho buscou evidenciar que mesmo teorias relativamente recentes para a indústria podem, neste momento, ter uma aplicação prática nos canais de irrigação.

Utilizando-se de arquiteturas usualmente implementadas e de grande difusão, mesmo que definidas sobre preceitos básico de controle clássico, buscou-se encontrar um

controlador baseado em redes neurais que fosse capaz de representar melhoras substanciais a tais arquiteturas, utilizando apenas os dados gerados por elas.

O caso de estudo ambienta-se em um simulador do Canal de Irrigação de Vila Nova de Milfontes, situado no Alentejo de Portugal, o qual encontra-se inserido no Projeto Orchestra, financiado pela Fundação pela Ciência e Tecnologia, que visa a criação de tecnologias e desenvolvimentos para o setor de irrigação de Portugal.

1.1 Motivação

O uso de água pela população em geral é um dos grandes questionamentos da atualidade, dada a aceitação de que a propriedade renovável da água depende do modo como nós a usamos.

Neste fato, o uso consciente deste recurso é uma das prioridades mundiais. Inseridos neste contexto, os canais de irrigação são uma construção vital para a vida humana, bem como uma das zonas de maior utilização dos recursos hídricos.

Este estudo tem, em essência, a vontade de mostrar que as novas tecnologias podem ser aplicadas para ganhos reais nestes sistemas, apresentando métodos modernos para se conseguir um controle adequado dos recursos hídricos e, aos poucos, colaborar com a conservação de mananciais.

Além disso, apesar de estar inserido em um caso especificamente do espaço português, dado que a fonte de estudo é o Canal de Rega de Vila Nova de Milfontes, todo e qualquer sistema de transporte de água e fluidos pode-se valer dos resultados, inserindo-se no Brasil no caso da transposição do Rio São Francisco, no Nordeste, bem como no caso mais recente das barragens de detritos de mineradoras, em Minas Gerais.

1.2 Metodologia

Todo o desenvolvimento do controlador neural deu-se em ambientes virtuais de MatLab e Simulink, baseando-se no simulador do Canal de Rega de Vila Nova de Milfontes, realizado por Nabais (2011) sob a guarda do Projeto Orchestra.

A princípio, o estudo baseia-se no trabalho de Simacek (2015), onde foram exemplificadas arquiteturas de controle usuais e proposta uma metodologia de avaliação destes sistemas de controle, também baseados no simulador de Nabais (2011) do Canal de Rega de Vila Nova de Milfontes.

A partir deste estudo, definiu-se as variáveis e dados a serem utilizados no decorrer de todo o estudo. Para a definição do modelo de redes neurais a serem utilizados, estudou-se os trabalhos de Hayjin (2001), Ferrari (2010), Hassan e Kothapalli (2010), Shu e Pi (2000) e o livro *Neural Network Applications in Control*, de Irwin, Warwick e Hunt (1995), estudos os quais exemplificam os diversos modelos de redes em funcionamento, principalmente em comparação aos controles proporcionais-integrrais, foco do estudo.

Também foram realizados estudos nos trabalhos de Bohn e Atherton (1998), Ghoshal e John (2010) e Li, Park e Shin (2007), para uma melhor análise sobre a implementação de um sistema de *Anti Wind Up* no controlador neural.

O estudo de Nabais, Mendonça e Botto (2013) também foi utilizado para entendimento do sistema em caso de falhas.

Por fim, houve uso constante da internet, em especial o site MathWorks, para avaliação dos métodos computacionais utilizados pelo MatLab e Simulink, que envolvem o treinamento e adaptação das redes neurais artificiais estabelecidas.

1.3 Descrição sumária do trabalho

Este estudo teve por objetivos a comprovação da eficiência da metodologia proposta de integração de arquiteturas de controle usualmente aplicadas no controle dos níveis de um canal de irrigação por meio de redes neurais artificiais.

Com base em no canal de irrigação de Vila Nova de Milfontes, situado no Alentejo português, simulado em ambiente virtual do Simulink criado por João Nabais, e no estudo das arquiteturas de Controle Local a Jusante (CLJ) e Controle Distante a Montante (CDM), bem como no *benchmark* de testes estabelecido por Simacek (2015), definiu-se os dados a serem usados para o treinamento da rede neural.

Segundo os estudos realizados por Haykin (2001), Ferrari (2010) e Hassan e Kothapalli (2010), definiu-se o modelo de rede neural a ser utilizado para o modelamento dos controles proporcional-integral a serem retratados.

Após as primeiras iterações de treinamento, efetivou-se um estudo minucioso dos valores de *bias* e pesos nas redes criadas afim de entender como o sistema estava a ser modelado e, com isso, definiu-se a taxa de aprendizagem do processo de adaptação das redes.

Por fim, com todos os parâmetros definidos, efetuaram-se os treinamentos e adaptações necessárias para o correto funcionamento do controlador neural final, o qual foi avaliado segundo o *benchmark* proposto por Simacek (2015) e comparado com os controladores por esse já testados na dissertação.

Este documento encontra-se dividido em 5 capítulos, a saber:

- Capítulo 1: Apresenta-se o trabalho, sua proposta e definição geral da metodologia aplicada.
- Capítulo 2: Revisa a teoria das redes neurais, bem como apresenta as ferramentas e teorias utilizadas para a obtenção dos resultados.
- Capítulo 3: Expõe o processo de criação do controlador neural, bem como todas as nuances que envolveram seu projeto.
- Capítulo 4: Os resultados do desempenho do controlador, bem como sua comparação com as arquiteturas as quais foi baseado, são apresentados e discutidos nesta parte.
- Capítulo 5: Apresentam-se a conclusão e as considerações finais acerca do trabalho realizado.

2 Fundamentação teórica

Neste capítulo, serão discutidas as duas premissas básicas para este estudo: as teorias acerca de redes neurais, desde sua base até sua aplicação em sistemas de controle, bem como suas variáveis de implementação, i.e., funções de treinamento, seleção de dados, parâmetros internos, etc.; o simulador ao qual será implementado o controlador desenvolvido, bem como a sua avaliação e comparação com as arquiteturas inspiradoras do controlador.

2.1 Redes neurais

O conceito por de trás da teoria de redes neurais artificiais (RNA) é inspirado na célula base do cérebro humano, o neurônio. O poder de processamento do cérebro ainda não é equiparável por nenhuma tecnologia ou teoria, contudo, o entendimento de seu funcionamento vem sendo re-aplicado nesta área.

A rede neural, assim sendo, pode ser entendida como um grupo de neurônios interligados que, a partir de um dado estímulo, executam uma tarefa previamente programada.

Tais redes assimilaram duas características cerebrais: o conhecimento adquirido vem do ambiente ao qual ela se insere, através de um processo de aprendizagem; o método de armazenagem de aprendizado, chamados de pesos sinápticos (Haykin, 2001).

A célula base de uma RNA, o neurônio, pode ser parametrizado por três variáveis: o peso de cada entrada, o *Bias*, e a função de ativação.

Segundo o modelo:

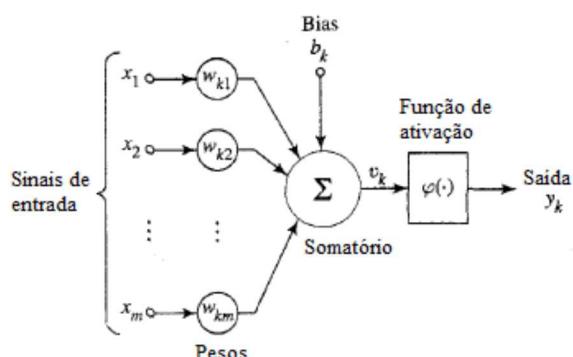


Figura 1: Modelo de neurônio (Haykin, 2001)

Ou, expresso em modelamento matemático:

$$u_k = \sum_{j=1}^m w_{kj}x_j \quad (1)$$

$$v_k = u_k + b_k \quad (2)$$

$$y_k = \varphi(v_k) \quad (3)$$

Onde x_j representa os sinais de entrada, w_{kj} o peso de cada entrada para o neurônio k , u_k é a somatória das entradas ponderadas, b_k é o *bias*, parâmetro intrínseco do neurônio k , v_k é o potencial de ativação do neurônio k , φ é a função de ativação e y_k é a saída do neurônio k (Haykin, 2001).

Assim sendo, uma RNA genérica pode ser representada pelo seguinte diagrama:

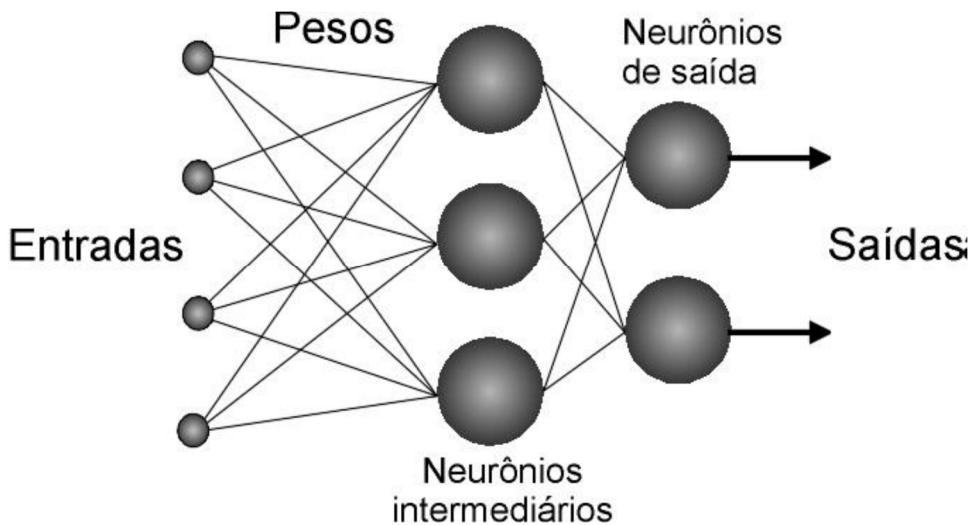


Figura 2: Rede Neural Multicamadas genérica (MLP) (Haykin, 2001)

Um sistema gerido por uma rede neural consegue, assim, generalizar o conhecimento, seja ele proveniente de um ambiente físico linear ou não, tendo seu aprendizado baseado em um conjunto de dados oriundos do sistema a ser representado, o que também provê a capacidade adaptativa de tais sistemas.

Portanto, para o desenvolvimento de uma RNA, deve-se determinar sua forma, ou seja, o número de entradas, o número de saídas (onde o número de neurônios na camada de

saída da rede será de mesmo número), a quantidade de camadas intermediárias (bem como o número de neurônios em cada uma) e a função de ativação.

2.2.1 Funções de ativação

A função responsável pela saída de um dado neurônio pode assumir muitas formas, mas, em geral, apresenta três formas básicas (Haykin, 2001) que, apesar de contemplarem grande parte da gama de aplicações, podem ser modificadas às necessidades do sistema.

A primeira forma, a *Função Limiar*, é dada por:

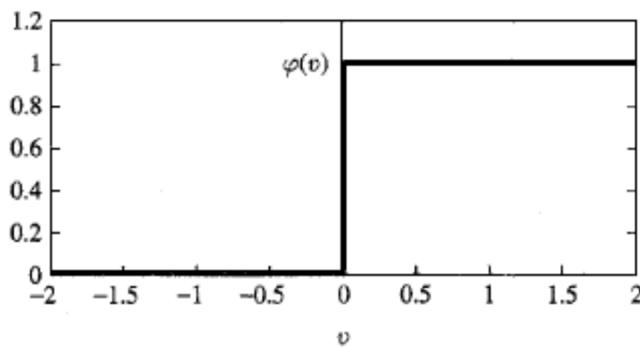


Figura 3: Função de Ativação Limiar (Haykin, 2001)

A segunda forma, a *Função Linear por Partes*, que é dada por:

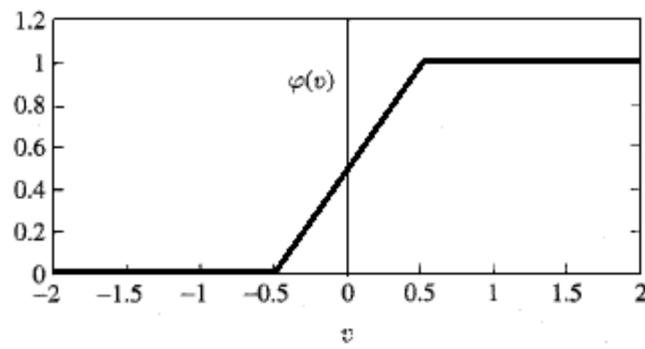


Figura 4: Função de Ativação Linear por Partes (Haykin, 2001)

Essa função, em exemplo, será usada de forma modificada na concepção dos controladores neurais a serem descritos no próximo capítulo, assumindo a forma dada pela equação:

$$\varphi(v) = v \quad (4)$$

E, por fim, a *Função Sigmoide*, a função mais aplicada em RNA, por apresentar uma forma balanceada entre o comportamento linear e não-linear, sendo ela dada por funções hiperbólicas e logarítmicas com uma variável a , dada por, em exemplo:

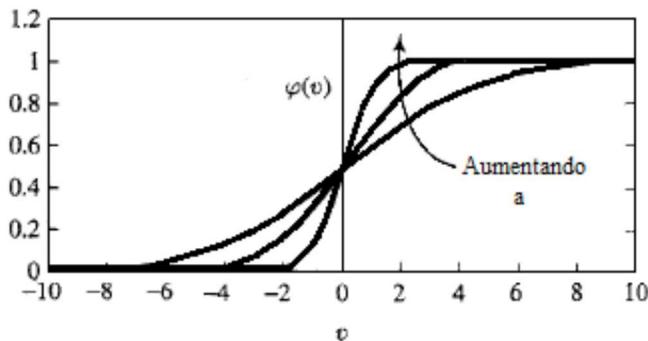


Figura 5: Função de Ativação Sigmoide (Haykin, 2001)

2.2.2 Arquiteturas comuns

No que diz respeito a arquitetura de uma RNA, ou seja, a forma que levam suas camadas internas e de saída, muitas teorizações e modelos são aplicados. Apesar da liberdade do usuário de utilizar uma arquitetura de qualquer proveniência e concepção, alguns casos generalistas valem ser exemplificados.

Dentre as arquiteturas mais comuns, as redes *Feedforward* são tidas como a arquitetura mais simples e, apesar disso, extremamente difundidas. Nestas redes, a saída de um neurônio sempre segue em sentido positivo na rede, i.e., um dado sinal não realimenta, em nenhum ponto, a rede neural, sempre sendo a saída dos neurônios de uma camada a entrada dos neurônios da próxima camada interna.

Por fim, outra arquitetura usualmente usada, no qual se baseará o controlador a ser definido, são as redes *Feedback*. Em tais redes, os neurônios são realimentados com as saídas da camada na qual se encontram.

Um exemplo desta arquitetura são as redes *NARX* (*Nonlinear autoregressive with exogenous input*), onde as camadas internas são dadas como uma rede *Feedforward*, mas as entradas da rede contam com a realimentação da saída.

As redes *Feedforward* são normalmente usadas para sistemas com funções não-diferenciais, enquanto as de *Feedback* são comumente usadas para séries e sistemas diferenciais.

A figura a seguir exemplifica tais arquiteturas.

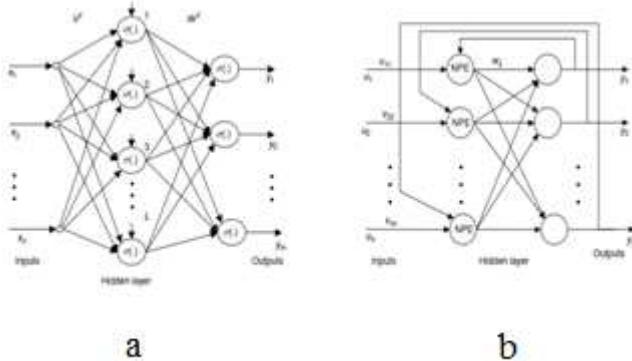


Figura 6: (a) Rede Feedforward (b) Rede Feedback (Haykin, 2001)

2.2.3 Aprendizagem e treinamento

No que diz respeito ao processo de aprendizagem, este pode ser dividido em duas classes (Haykin, 2001), se o aprendizado é supervisionado ou não.

Quando supervisionado, os pesos sinápticos são ajustados conforme um padrão ou um conjunto-saída padronizado, onde o ajuste decorre da tentativa de se anular o erro entre o resultado obtido a partir das entradas do exemplo e as saídas padronizadas.

Já quando a aprendizagem não é supervisionada, o ajuste se dá por regras e propriedades determinadas no conjunto de dados.

A aprendizagem de uma RNA dá-se pelo seu processo de treinamento, processo o qual se dá efetivamente o ajuste das variáveis livres da rede.

Em suma, pode-se dividir o treinamento em dois processos distintos.

O treinamento em *batch*, ou por lote, ajusta os pesos após o processamento de todos os dados de treinamento, ou seja, para cada espécime do treinamento, calcula-se o erro perante a saída e, após todo o processamento, faz-se o ajuste dos ganhos, definindo-se assim uma iteração.

Já o treinamento *online*, ou sequencial, ajusta os pesos de forma contínua após cada exemplo processado, até que o último exemplo seja processado, o que define uma

iteração. Ao tempo que este processo foi estipulado, os cientistas da área acreditavam que obteriam resultados menos precisos com este treinamento, dada a possibilidade de que, dentre um dado subconjunto amostral, o ajuste dos pesos pudesse reduzir o erro em uma forma não natural e que, ao fim da iteração, o erro total fosse maior do que se processado em *batch*, porém, apesar da falta de confirmações analíticas, este método mostra-se o mais eficaz no treinamento de redes e, assim sendo, será o foco neste estudo.

O principal algoritmo de treinamento sequencial é o algoritmo de Retropropagação, onde o ajuste dos pesos dá-se em dois processos sobre as camadas da rede neural.

A primeira etapa, chamada de *forward*, ocorre no sentido positivo da rede, i.e., a partir das entradas padronizadas define-se as saídas da rede, onde os pesos são mantidos fixos. A segunda etapa, chamada de *backward*, os pesos são ajustados no sentido negativo da rede, ou seja, da camada de saída até a primeira camada, segundo uma regra de aprendizagem de correção dos erros.

Salienta-se que este será o processo de aprendizagem e treinamento do foco de estudo.

2.2 Simulação e avaliação

A proposta de estudo é baseada no simulador desenvolvido por João Nabais do Canal de Irrigação de Vila Nova de Milfontes, no âmbito do Projeto Orchestra, em Simulink, sendo este formulado sobre os princípios definidos em Nabais et. al. (2011) e Lirico e Fromion (2004).

O canal simulado consta com quatro segmentações interligadas sequencialmente, que serão tratados por piscinas, sendo controladas as comportas a montante e a jusante de cada uma. Cada piscina conta, ao seu longo, de alguns pontos de extração de água equidistantes entre si, sendo que cada piscina possui um número determinado de pontos de tomada de água, com valores nominais e máximos próprios de cada ponto.

Com isso, no total, são 29 pontos de extração no canal, 5 comportas controladas, sendo uma de entrada do canal, uma de saída do canal, e 3 intermediárias, que fazem a ligação entre as piscinas.

As comportas simuladas são do tipo guilhotinas com secção transversal trapezoidal, igual à do canal, sendo sua componente de controle a altura da comporta, sendo seus limites de ação dados por 0, quando está fechada, e pelo nível a montante da comporta como limite superior.

O simulador divide-se em duas estruturas principais: a estrutura e modelamento do canal em si, parte majoritariamente provida do trabalho de João Nabais: o controlador, onde será implementado o trabalho por este realizado e parte majoritariamente provida por Simacek (2015), onde se buscou os dados a serem tratados neste trabalho.

Em Simacek (2015), propôs-se dois sistemas de controladores proporcional-integral (PI) em arquiteturas diferentes para se controlar o canal. Em ambas, o controlador recebe o erro entre a referência e o nível a ser controlado e a saída do controlador é a variação da altura da comporta em relação a altura em operação nominal.

O Controle Local a Jusante (CLJ), onde, para uma dada piscina do canal simulado, o controle do nível a jusante foi realizado localmente, pela comporta a jusante do canal.

Tal arquitetura é dada pelo diagrama a seguir.

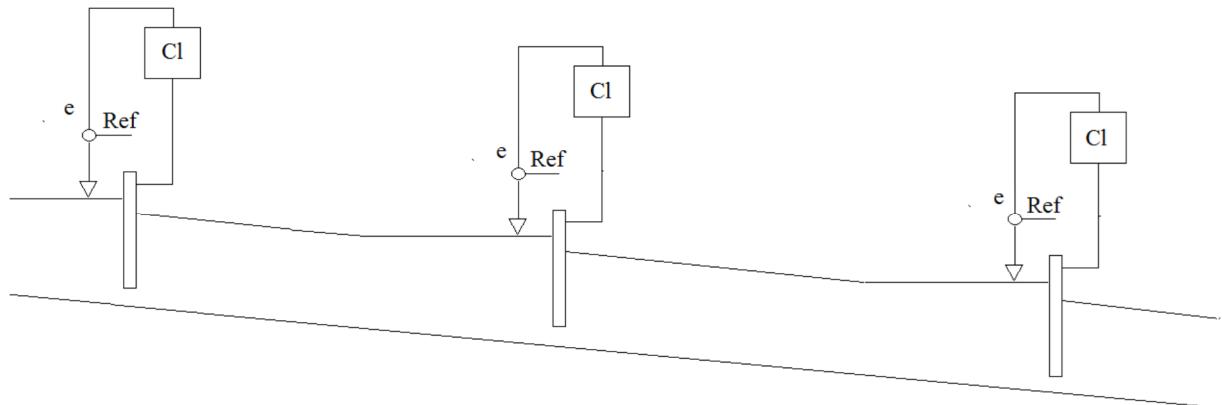


Figura 7: Arquitetura de Controle Local a Jusante (Simacek, 2015)

A segunda arquitetura, o Controle Distante a Montante (CDM), onde o nível a jusante de uma dada piscina é controlado pela comporta a montante da mesma, distante do nível, o que acarreta em atrasos no sistema. Tal arquitetura é dada pela figura 8.

Tais estruturas, bem como os controladores desenvolvidos para o simulador do canal em Simacek (2015) serão utilizados como a base de dados para este trabalho.

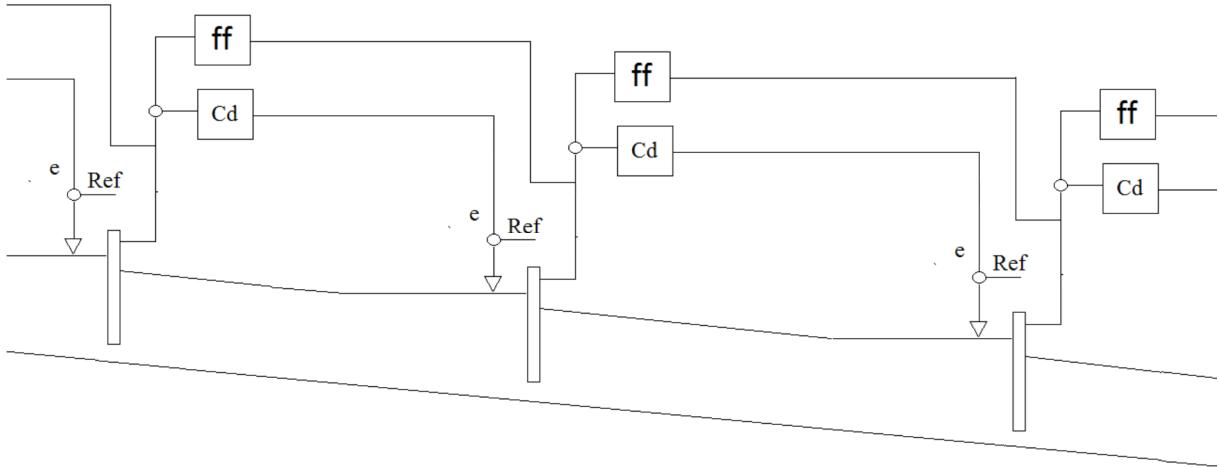


Figura 8: Arquitetura de Controle Distante a Montante (Simacek, 2015)

Para o processo de avaliação do controlador desenvolvido, usar-se-á o *benchmark* descrito a seguir, onde os testes baseiam-se no sequenciamento de variação das tomadas de água, sendo avaliados por três indicadores, um indicador composto por dois parâmetros para a eficiência técnica do sistema, um para a eficiência energética e outro para a eficiência de consumo de água (Simacek, 2015).

O indicador de Desvio do Nível de Referência, para um sistema discreto, tem dois índices que refletem o erro médio e o desvio padrão da média para cada piscina do canal, dados a seguir.

$$\eta_{m_i} = \frac{\sum(Y_i(k))}{n} - Ref_i \quad (5)$$

$$\eta_{r_i} = \sqrt{\frac{\sum((Y_i(k) - \eta_{m_i} - Ref_i)^2)}{n-1}} \quad (6)$$

Onde η_{m_i} representa o erro médio da piscina i , η_{r_i} representa o desvio padrão da média da piscina i , $Y_i(k)$ é o valor do nível da piscina i no momento k , n é a quantidade de intervalos avaliados e Ref_i é o nível de referência da piscina i .

O indicador de Desvio de Consumo de Água, avalia a relação entre o total de água que entra no sistema e o total de água extraído com sistema nas condições de teste e nas condições nominais, segundo a fórmula a seguir.

$$\eta_{da} = \frac{\int Q_{in} dt / \int Q_{outs} dt}{\int Q_{in_{nominal}} dt / \int Q_{outs_{nominal}} dt} \quad (7)$$

Onde η_{da} representa o índice de consumo, Q_{in} representa a vazão de entrada de água no canal e Q_{outs} é o fluxo total de água extraído do canal.

O indicador de Consumo de Energia relaciona a movimentação média das comportas segundo a fórmula a seguir.

$$\eta_e = \frac{\sum \sum |\Delta s_{i,j} \cdot (a_{i,j} + g)|}{n_g} \quad (8)$$

Onde η_e é o índice de consumo energético, Δs_{ij} é a variação da posição da comporta i entre os instantes j e j+1, a_{ij} representa a aceleração da comporta i no momento j, g a aceleração gravitacional e n_g o número de comportas avaliadas (Simacek, 2015).

O *benchmark* a ser utilizado prevê teste de tempo simulado de 24 horas, sendo a primeira metade dedicada a perturbação do sistema e a outra metade a retomada dos valores nominais. Estipula-se, também, quatro cenários para o sequenciamento de extração.

Cada cenário salienta duas variáveis quanto a sincronização, uma quanto a sincronização das tomadas de água ao longo de uma piscina, quanto na sincronização entre as piscinas.

No primeiro quesito, pondera-se entre as variações nas extrações afim de se somar as influências de cada uma, abrindo-as sequencialmente, chamada de Abertura Sequencial, ou para se explorar o máximo de variações em um ponto, variando-as no mesmo instante, chamado de Abertura Estática.

Quanto a sincronização das piscinas, vê-se duas possibilidades, igualmente acima mostrado. A primeira, chamada de Influência Pontual Máxima, onde todas as piscinas sofrem

suas perturbações em um dado instante, e a segunda, chamada de Influência Sequencial, onde a perturbação de uma piscina soma-se as perturbações das anteriores (Simacek, 2015).

A figura a seguir evidencia esses cenários.

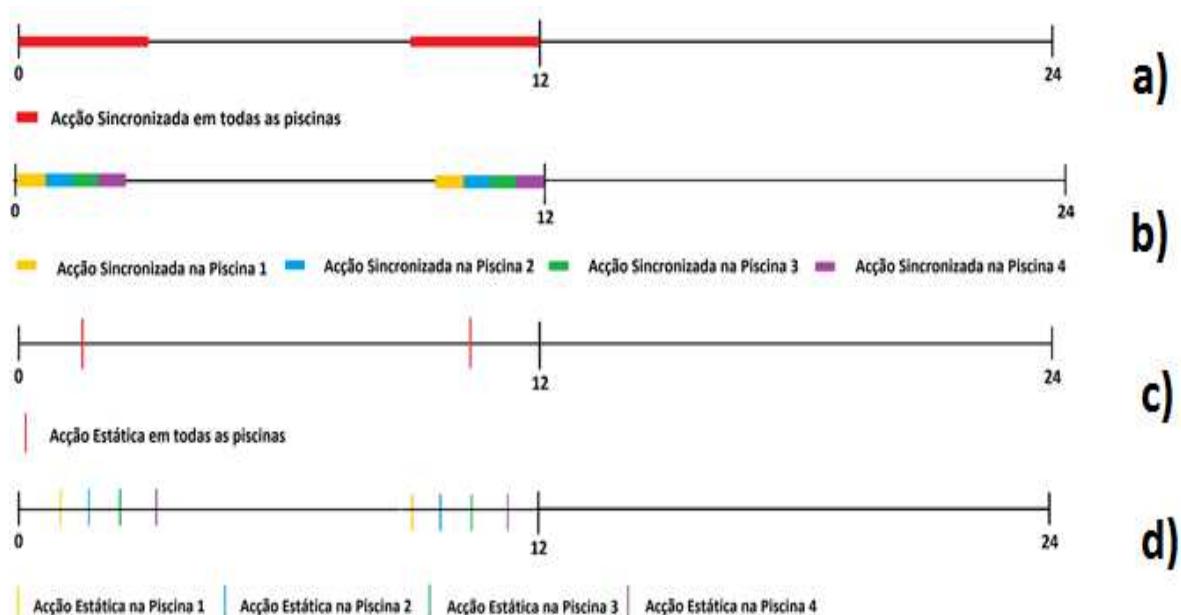


Figura 9: Cenários. a) SP, Abertura Sincronizada com Influência Pontual Máxima, b) SS, Abertura Sincronizada com Influência Sequencial, c) EP, Abertura Estática com Influência Pontual Máxima, d) ES, Abertura Estática com Influência Sequencial (Simacek, 2015)

3 Desenvolvimento do controlador

Neste capítulo, será abordado o desenvolvimento do controlador neural baseado nas arquiteturas CLJ e CDM, desde seus requisitos, definições e processos até sua versão final para este projeto.

3.1 Requisitos

Apesar de pouco restritivo, alguns conceitos e requisitos serão impostos ao controlador. Primeiramente, que o controle seja consistente para diversas perturbações, mantendo a resposta ao tempo em uma simulação congruente com a teoria de controle envolvida no processo.

Em segundo lugar, o controlador deve ser capaz de ter o tempo de assentamento o menor possível, sendo o limite superior de 10 horas no caso de maior extração de água possível, para que este estabilize em cada etapa do processo de avaliação do *benchmark*.

Espera-se também minimizar ao máximo o sobressinal da planta para os casos de maior variação das extrações, contudo, será considerado aceitável sobressinais (quando o simulador operar normalmente, não em casos de saturação) de 10%.

Sobretudo, dado que o sistema é considerado uma ponderação entre as duas arquiteturas anteriormente citadas, espera-se que seu desempenho seja melhor do que o limite inferior de desempenho das arquiteturas, i.e., a que apresente pior desempenho.

Assim sendo, espera-se valores menores no indicador de Desvio do Nível de Referência em comparação aos piores resultados obtidos pelas arquiteturas originais. Contudo, os outros índices não serão critérios para rejeição do controlador dado que, sem tratamento posterior do controlador desenvolvido, há a possibilidade de o controlador estabilizar-se em um novo ponto de funcionamento, dado o fato de, no sistema matemático, termos menos variáveis controláveis do que controladores, o que pode resultar em mais do que uma solução para o mesmo ponto de operação.

Por estes conceitos, define-se os requisitos para o controlador.

3.2 Definição de redes neurais

Para o desenvolvimento dos controladores neurais, inicialmente, buscou-se o melhor modelamento de um controlador PI genérico.

A função que define um controlador PI, no domínio do tempo, é dada a seguir.

$$u(t) = K_p e(t) + K_i \int_0^t e(t) dt \quad (9)$$

Onde u é a saída do controlador, e é o erro entre a referência e a saída da planta, K_p é o ganho proporcional e K_i é o ganho integral do controlador.

Reformulando a equação para o caso de tempo discreto, temos a seguinte expressão.

$$u(K) = K_p e(K) + K_i \sum_{n=0}^K T_s e(n) \quad (9)$$

$$u(K) = K_p e(K) + K_i T_s e(K) + \sum_{n=0}^{K-1} T_s e(n) \quad (10)$$

Estudando a composição da ação de controle, nota-se, por óbvio, uma componente proveniente da componente proporcional (definida por P) e outra da componente integral (definida por I). Assim sendo, temos.

$$u(K) = P(K) + I(K) \quad (11)$$

$$P(K) = K_p e(K) \quad (12)$$

$$I(K) = K_i \sum_{n=0}^K T_s e(n) = K_i T_s e(K) + I(K-1) \quad (13)$$

Assim sendo, podemos reescrever a função do controlador como dado a seguir.

$$u(K) = P(K) + K_i T_s e(K) + z^{-1} I(K) \quad (14)$$

Tal expressão pode ser traduzida em uma rede neural representada no diagrama a seguir.

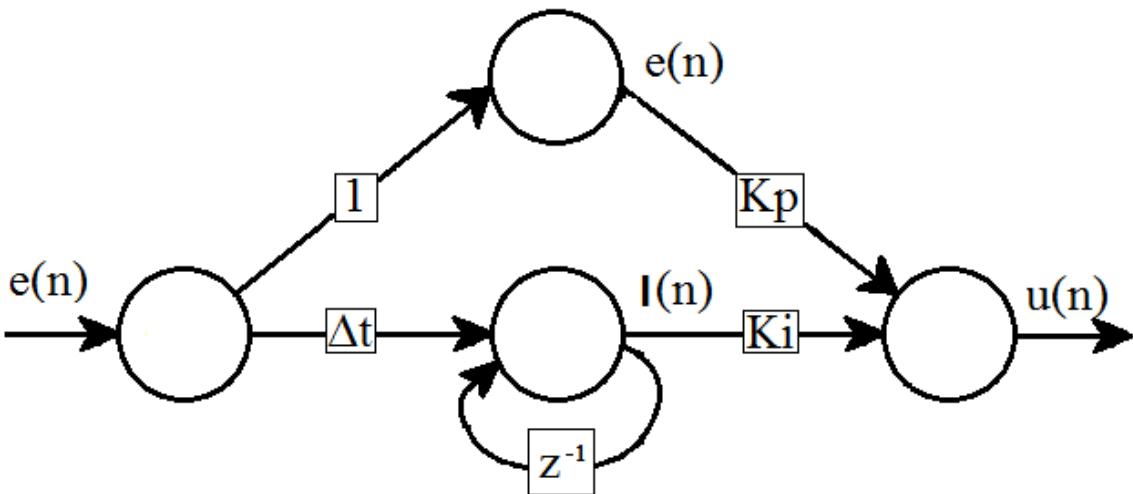


Figura 10: Estrutura neural de um controlador PI (Adaptado Ferrari, 2010)

Esta rede neural é do tipo recursiva, com *bias* nulos e função de ativação linear tal qual a Equação 4.

Assim sendo, a concepção do controlador neural para o simulador do canal de irrigação deve-se, antes, avaliar como a estrutura básica de controlador neural PI descrita acima irá agir para cada comporta.

Para o caso, em cada arquitetura, existem 4 comportas controladas e uma passiva, ou seja, a comporta de entrada do canal só é modelada para controle no caso da arquitetura CDM e a comporta de saída para o caso CLJ.

Tais comportas serão modeladas, então, apenas com uma unidade básica neural PI, exatamente como na figura 10, já que, para ambas, existe apenas um controlador para base, com apenas uma entrada de erro.

Já para as comportas intermediárias, que estão presentes em ambas as arquiteturas, analisou-se a relação entre os ganhos dos controladores em CDM e CLJ e notou-se a incidência de sinais opostos para as constantes de ganho.

Tal diferença vem do fato que, e.g., no caso de a referência ser maior que o nível em um dado instante, ou seja, o erro, que é dado pela diferença entre o nível e a referência, é negativo, mas a variação da comporta em CDM deve ser positiva a fim de abrir mais a comporta e, assim, aumentar o fluxo de água na entrada da piscina, enquanto no caso CLJ a ação de controle deve ser negativa, assim como o erro, para diminuir a abertura da comporta e diminuir o fluxo de saída do canal.

Usando-se deste fato, em um caso de ação de controle final for resultado da soma das ações em CLJ e CDM, em algum dado momento, as componentes integrais de cada parte da rede entrariam em sintonia, arquitetura a qual é representada no diagrama a seguir.

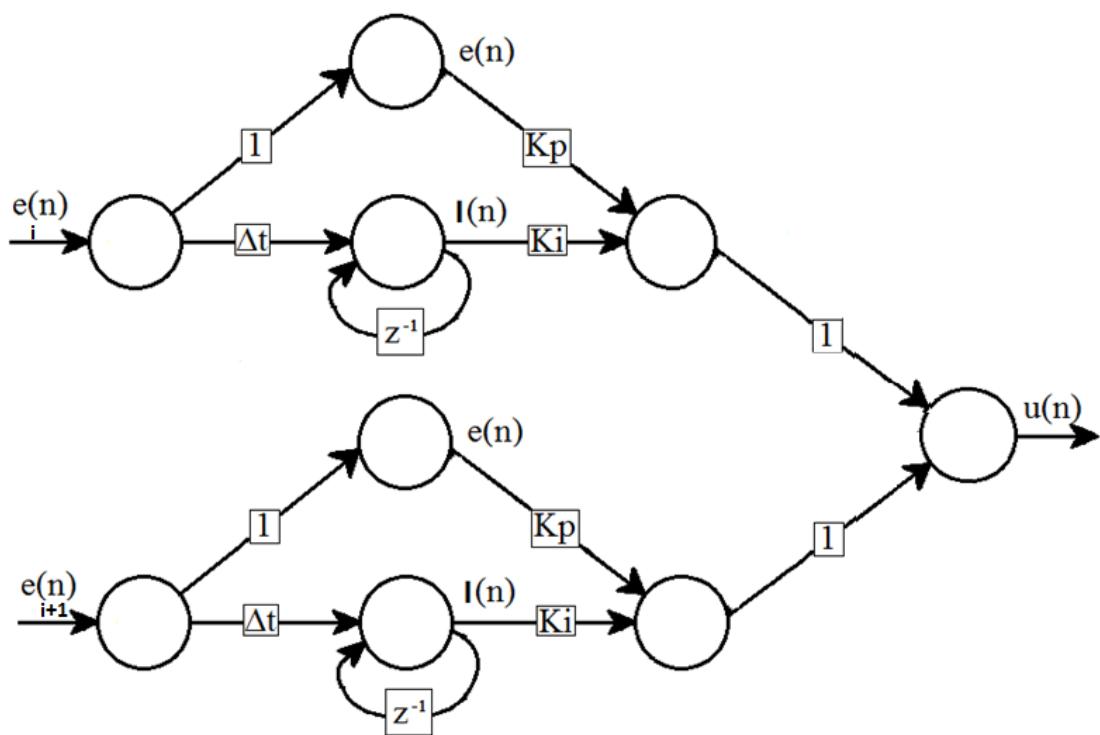


Figura 11: Arquitetura das redes neurais de controle das comportas internas do canal (Adaptado Ferrari, 2010)

Em um caso onde ambas as ações tivessem o mesmo sinal, o módulo da componente final seria amplificado e, tomando-se precauções para que não ultrapasse os limites de estabilidade, teria um desempenho melhor. No caso onde as ações de controle primárias tivessem sinais o contrário, a sintonia faria que, em algum dado instante, as componentes integrais anulassem a derivada temporal da saída do controlador, ou seja, os erros e ganhos estariam em um estado onde a ação de controle manter-se-ia a mesma.

Tal fato acarretaria em certo retardo na execução das arquiteturas caso comparadas separadamente com suas bases originais, mas, como os níveis envolvidos também sofreriam com as influências da comporta anterior ou posterior, na mesma tentativa de minimizar o erro, tal retardo não deve comprometer o funcionamento dos controladores, apesar do cuidado na implementação a ser considerado.

3.3 O processo de treinamento

O processo principal para a criação da rede neural é o seu treinamento, onde, enfim, suas variáveis de processo serão definidas e, assim, sendo capaz de funcionar como foi designada.

Contudo, envolvendo duas arquiteturas não correlacionadas por modelos físicos, o treinamento seria, em suma, impossível. Por impossível entende-se que as relações estabelecidas não seriam reais e, assim sendo, o processo do controlador seria errôneo.

Para o correto processo de controle, então, definiu-se um método de treinamento onde os pesos fossem atualizados conforme séries de dados, sendo que cada série era correlata a uma estrutura de controle PI, e sucessivas adaptações destes pesos afim de se encontrar a melhor resposta.

Para os casos da comporta de entrada e para a comporta de saída, elas foram treinadas conforme as arquiteturas as quais correspondem, como explicados anteriormente. Para tal, utilizou-se séries de dados obtidas no controlador para casos onde os controladores envolvidos não entrassem em saturação, para evitar a perda de informação correlacionada ao controlador.

Tal processo foi realizado como um simples treinamento e os resultados comparados aos valores dos controladores originais.

Tabela 1: Valores originais dos controladores (Simacek, 2015) e os obtidos com o treinamento da rede neural

Comporta	Original		Neural	
	Kp	Ki	Kp	Ki
1	-8,03E+00	-5,32E-04	-7,98E+00	-5,38E-04
5	1,72E+02	2,23E+00	1,65E+02	2,35E+00

Os valores amostrados mostram grande congruência com os valores teóricos, sob os quais eles foram baseados, tanto para a comporta de entrada (comporta 1) e para a de saída (comporta 5).

Para as comportas de 2 a 4, o processo de treinamento levou em conta duas variáveis: o número de adaptações a serem feitas; a taxa de aprendizado para cada adaptação.

Adaptação é o modo pela qual a rede neural atualiza os valores dos seus ganhos mesmo depois de treinada. Durante a adaptação, ocorre um novo treinamento sob as mesmas premissas do treinamento original da rede, com uma nova fonte de dados, e os pesos são atualizados segundo a taxa de aprendizado, sob a fórmula a seguir.

$$w_{kj\ Novo} = (1 - t_a)w_{kj\ original} + t_a w_{kj\ Adaptação} \quad (15)$$

Onde w são os valores do peso j novo, original e oriundo da adaptação do neurônio k , e t_a é a taxa de aprendizado.

Para um correto funcionamento do controlador, i.e., não haja sobrepujança de uma das arquiteturas perante a outra, prevê-se que a magnitude dos ganhos deve ser próxima, ou seja, na mesma ordem de grandeza, tanto para o ganho integral quanto o proporcional.

Logicamente, este fato deve ser verdadeiro, já que, para perturbações semelhantes, as partes do controlador neural devem apresentar respostas de mesma magnitude, para que a ação de uma delas seja ofuscada pela do outro.

Assim sendo, estudou-se os ganhos dos controladores originais.

Tabela 2: Dados dos controladores originais (Simacek, 2015) e dados estatísticos.

Comporta	CLJ		CDM		Kp (CDM/CLJ)	Ki (CDM/CLJ)
	Kp	Ki	Kp	Ki		
2	1,72E+02	2,25E+00	-1,24E+00	-1,35E-04	0,72%	0,01%
3	1,85E+02	2,40E+00	-2,09E+00	-1,25E-03	1,13%	0,05%
4	1,61E+02	2,10E+00	-2,38E+00	-2,27E-03	1,47%	0,11%
				Médias	1,11%	0,06%

Nota-se, assim, que os valores do ganho integral são da ordem de 2000 vezes menor no CDM do que no CLJ, e de 100 vezes para o ganho proporcional. Por tanto, como a ação de controle de uma parte depende apenas do fator integral quando o tempo tende ao infinito e contando erro estacionário nulo, escolhemos, arbitrariamente, que a proporção da força de ação CLJ deve ser 1000 vezes menor que a força de ação CDM.

Tal definição vem diferença de grandeza entre os ganhos das arquiteturas e servirá para definir os parâmetros de treinamento.

Para o treinamento, foram considerados 66 testes nos controladores originais, que foram realizados no estudo do *benchmark* proposto para avaliação (Simacek, 2015), sendo este 33 em cada arquitetura, com padrões de extração diferentes.

Definiu-se que os valores iniciais dos pesos seriam os encontrados nos controladores originais, como na tabela 2. Os testes aos quais a adaptação será submetida foram organizados em uma sequência de iterações, intercalando um teste CDM com um teste CLJ (em situações semelhantes de simulação), com taxas de aprendizado iguais a $1-t_a$ para o caso CDM e t_a para o caso CLJ. Os pesos a serem dados como estimativa inicial no processo de adaptação serão os obtidos como resultado na iteração anterior.

Ou seja, resume-se as iterações ao algoritmo abaixo.

```
if Teste(i) == CLJ
    Net(i) = adapt(Net(i-1), Pesos(i-1), Outputs(i), Inputs(i), ta)
elseif Teste(i) == CDM
    Net(i) = adapt(Net(i-1), Pesos(i-1), Outputs(i), Inputs(i), 1-ta)
end
```

Segundo estas iterações e, considerando a hipótese de que o processo de adaptação seja perfeito, ou seja, para um teste CLJ os valores da parte do controlador responsável por esta estrutura serão idênticos aos valores originais e a outra parte terá ganhos nulos, e vice-versa, temos que, para se reduzir os ganhos da parte CLJ em mil vezes e praticamente manter os ganhos da parte CDM normais, deve se usar uma taxa de aprendizado (t_a) igual a 0,001, como era esperado.

Para o processo de treinamento especificado, então, criou-se a base neural para os controladores a serem implementados no controlador que, segundo o esquema apresentado na figura 11, apresenta os seguintes resultados:

Tabela 3: Pesos dos controladores neurais encontrados

Comporta	CLJ		CDM		Kp (CDM/CLJ)	Ki (CDM/CLJ)
	Kp	Ki	Kp	Ki		
2	3,45E+00	4,59E-03	-2,58E+00	-4,40E-03	74,88%	95,94%
3	1,64E+01	2,25E-02	-4,22E+00	-3,26E-02	25,68%	144,70%
4	1,51E+01	2,05E-02	-4,86E+00	-6,95E-02	32,06%	339,85%
				Médias	44,21%	193,50%

Como previsto, os ganhos apresentam grandezas próximas, estando prontos, então, para implementação no simulador.

3.4 Implementação no simulador

Para a implementação dos controladores neurais no simulador, notou-se a necessidade da implementação de um sistema de *Anti-Wind Up*, para melhor resposta do tempo.

Para tal, adicionou-se uma entrada no controlador neural, cuja função é:

$$Sat(K) = \begin{cases} 0, & \text{sem saturação} \\ e(K), & \text{com saturação} \end{cases} \quad (16)$$

E o diagrama do novo segmento de controlador neural é dado pela seguinte figura.

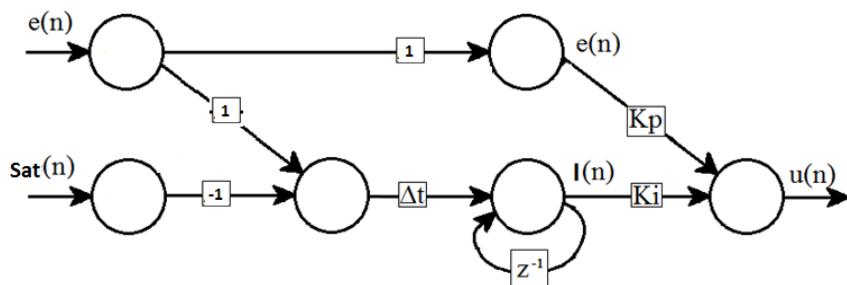


Figura 12: Novo controlador neural com anti-wind up

Desta forma, caso haja saturação, a entrada na parte integral será nula e, então, o efeito de *Wind Up* será cortado.

3.5 Simulação

Como descrito no capítulo 2, o sistema foi submetido ao *benchmark* proposto (Simacek, 2015) a fim de comprovar sua eficiência perante aos controladores clássicos utilizados anteriormente.

Serão realizados 33 testes, sendo um em condições nominais, 16 em cenários de variação positivas na extração e outros 16 em situação inversa. Cada teste, ainda, está enquadrado em um dos 4 cenários de extração anteriormente descritos.

4 Resultados

Nesta seção serão apresentados e discutidos os resultados dos controladores, sejam eles a própria definição dos controladores, quanto aos resultados obtidos a partir da simulação do controlador neural em funcionamento.

4.1 Os controladores

Como evidenciado no capítulo anterior, os ganhos dos controladores neurais apresentavam a mesma magnitude, como desejado. Contudo, os valores divergiam em relação à proporção esperada, no caso das adaptações fossem perfeitas, como descrito anteriormente.

Segundo a proposta, esperava-se, apesar da ciência de que não seria o ocorrido, que os ganhos da parte CLJ do controlador fossem decair em uma margem de 1000 vezes, enquanto a parte do CDM fosse se manter basicamente igual.

Segundo a tabela a seguir, podemos ver como foi dada a variação nos controladores.

Tabela 4: Variação dos parâmetros entre os originais e os treinados

Comporta	CLJ		CDM	
	Kp	Ki	Kp	Ki
2	2,01%	0,20%	208,56%	3249,15%
3	8,88%	0,94%	201,96%	2598,64%
4	9,40%	0,98%	204,34%	3066,27%

Podemos notar que a queda nos parâmetros da parte CLJ do controlador neural não foram tão grandes como esperado, contudo, foi compensado pela variação positiva do módulo dos ganhos da parte CDM.

Como o método de treinamento visa diminuir os erros entre a saída calculada e as saídas reais impostas pela série de dados, não há como prever como as iterações se dão, contudo, estas são, de fato, soluções sub-ótimas para minimizar o erro entre a ação do controlador neural com as arquiteturas base segundo a proporção dada.

Nota-se que, como a piscinas 3 e 4 são muito parecidas, os controladores neurais de ambas, apesar de terem sido treinados separadamente, apresentam resultados extremamente similares, tanto nos resultados absolutos expressos no capítulo anterior quanto na variação dos ganhos dos controladores base.

A comporta dois apresenta resultados mais dispares, pois, a piscina 1 e 2 são muito diferentes entre si, sendo ambas de extensões maiores do que as demais piscinas, sendo então creditado a isso o fato do treinamento ter ponderado os valores dos ganhos mais baixos para este controlador.

4.2 Análise da resposta ao tempo

Nesta etapa do trabalho, evidenciar-se-á o teste no cenário SS sob o efeito de máxima extração possível no canal, para analisar os efeitos dos controladores nos níveis controlados bem como a presença da ação CLJ e CDM juntas no mesmo cenário.

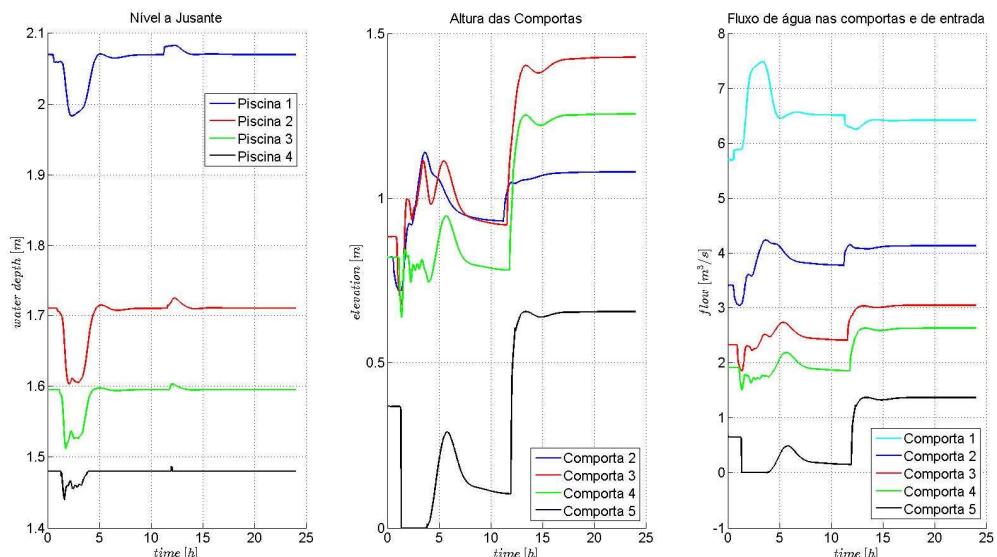


Figura 13: Resposta temporal ao ensaio SS em amplitude máxima de extração

A resposta temporal no pior cenário possível para o controlador apresenta resultados adequados ao esperado.

Primeiramente, nota-se que a correção dos níveis foi escalonada, sendo o nível da piscina 4 o primeiro a ter seu nível normalizado, como era de se esperar. Nota-se, também, o comportamento CDM e CLJ sendo executados, segundo a ampliação a seguir.

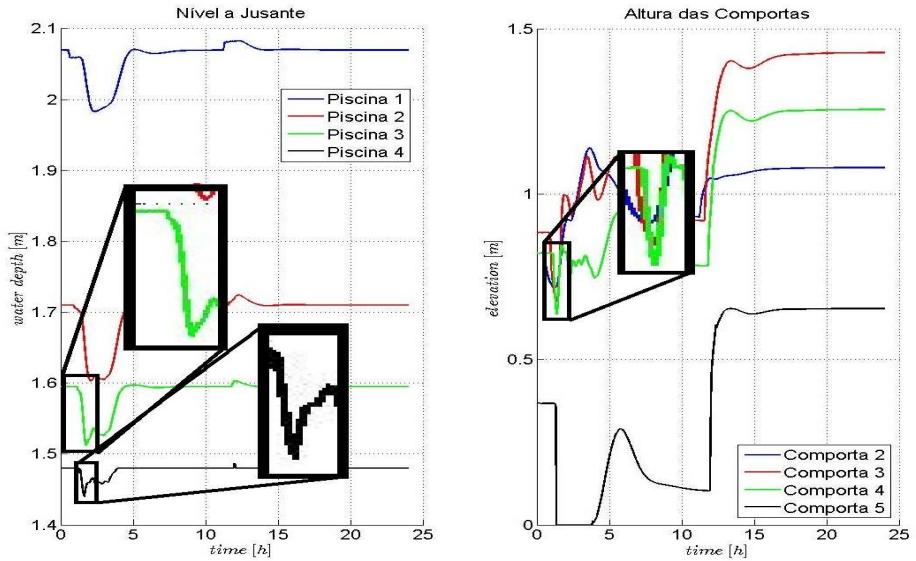


Figura 14: Detalhe da ação CDM e CLJ nos controladores

No detalhe do gráfico a direita, podemos notar que o controlador, em um primeiro momento, tende a fechar a comporta, o que corresponde ao momento em que o nível da comporta a montante da comporta perde nível, sendo, então, a parte CLJ do controlador neural entrando em ação.

Em um segundo instante, a comporta abre-se novamente, em resposta ao estímulo vindo do nível a jusante desta, que também tem seu nível diminuído, demonstrando o efeito CDM do controlador neural.

Ainda mais, dá-se a noção da ressonância explicitada anteriormente, pois a comporta acaba por encontrar um ponto de equilíbrio onde os dois níveis mantêm-se constantes, em uma ponderação com o controlador, até que o fluxo oriundo da entrada do canal faça-se presente no nível e, assim, o corrija.

Outro fato interessante, como visto na figura 13, o nível das comportas ao fim da simulação é maior do que ao início, apesar de estar em mesmas condições de extração. Isso se dá graças ao controlador da comporta de entrada do canal que, como o erro do nível que controla foi ajustado pela face CLJ do controlador neural da primeira comporta, acaba por manter-se com o integrador carregado.

Isso também era esperado, dado o fato do sistema estar subdefinido, ou seja, encontra-se com mais variáveis de controle do que variáveis a serem controladas.

Tal fato deve degradar o resultado do indicador de consumo de água, contudo, por não ser o foco deste estudo, não se tentou instalar métodos de descarga do integrador do controlador neural do fluxo de entrada.

4.3 Indicadores de desempenho

Nesta seção serão apresentados os indicadores de desempenho para o sistema aqui desenvolvido, bem como a comparação para com os sistemas de controle base, avaliados em outros trabalhos (Simacek, 2015).

4.3.1 η_{ad}

O indicador de desvio no consumo de água apresentou o seguinte resultado, para cada teste.

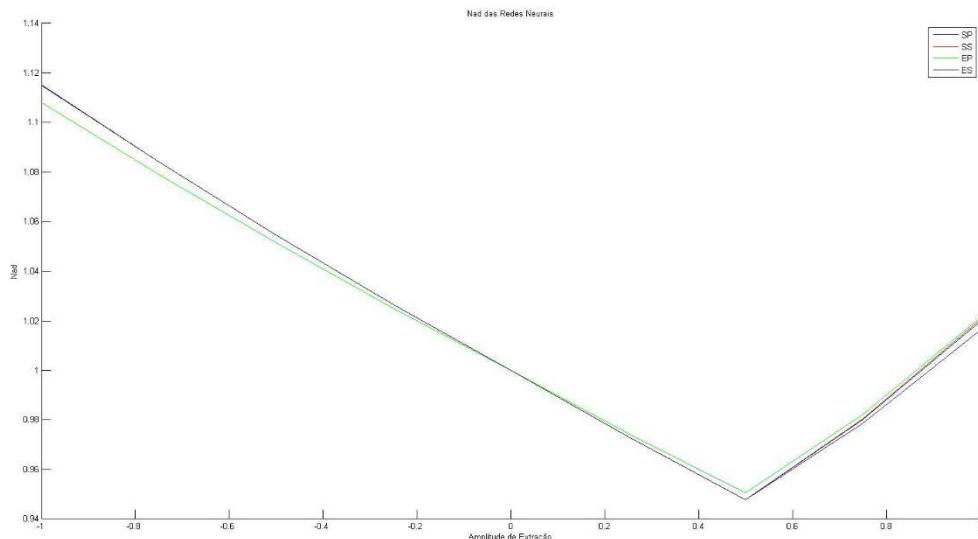


Figura 15: η_{ad} para o controlador desenvolvido

Podemos notar que não há grande variância entre os diversos cenários, mas, sim, com a amplitude de extração, com desvio de consumo máximo de, aproximadamente, 12% e mínimo de 4%.

Como era esperado, dado o ponto de equilíbrio anômalo apresentado nas repostas ao tempo, o desvio neste indicador é maior do que o encontrado nos sistemas base deste.

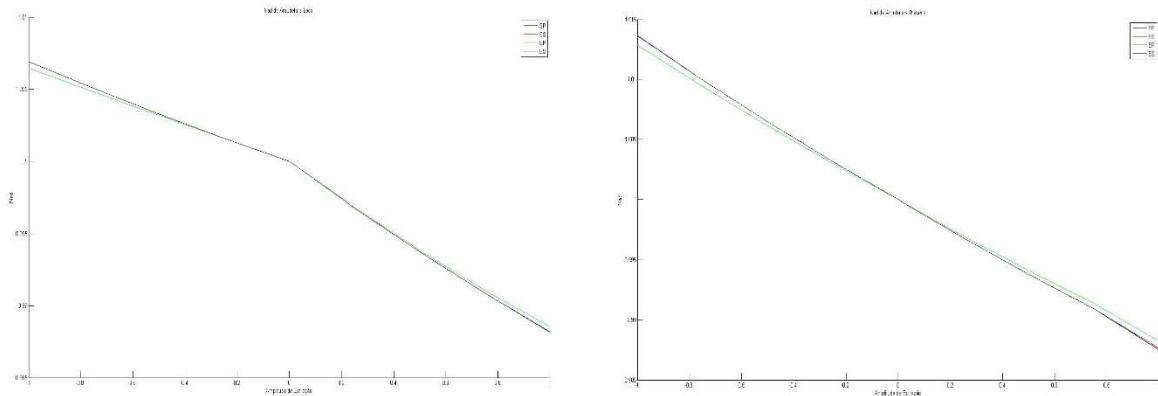


Figura 16: Resultados do η_{ad} para as arquiteturas CDM e CLJ (Simacek, 2015)

Os sistemas anteriores tiveram resultado muito favorável, com desvio de consumos de módulo menor a 1%.

Isto evidencia a necessidade de tratar o ponto de equilíbrio do canal para o caso do controlador neural.

4.3.2 η_e

Quando ao índice de consumo energético, esperava-se que o resultado fosse dado entre os valores dos controladores originais, e isso se comprovou, conforme o gráfico.

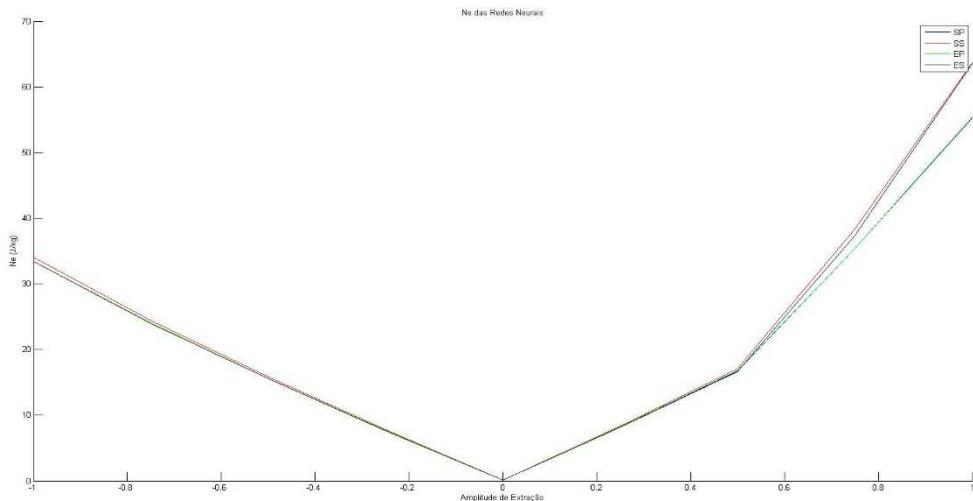


Figura 17: η_e para o controlador neural

Para os casos onde há excesso de água no sistema (amplitude de extração negativa), o consumo é menor do que o em caso de excesso, bem como nos controladores originais. Nota-se, também, valores mais próximos do CLJ, contudo, ainda mais especificamente no caso de excesso de água, o consumo energético foi menor.

Todas essas evidências se comprovam segundo o gráfico para comparação a seguir.

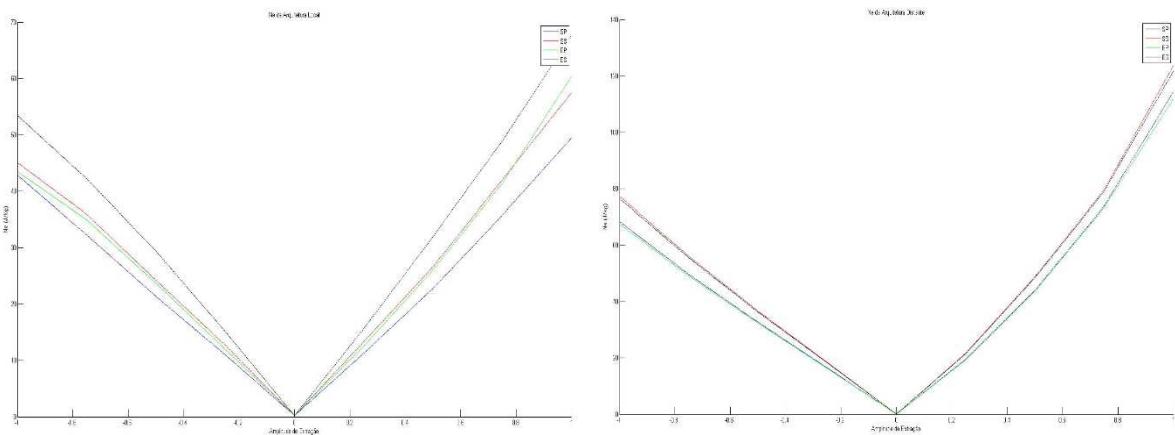


Figura 18: Resultados do η_e para os controladores originais (Simacek, 2015)

O interessante deste resultado é a semelhança em forma do gráfico do sistema neural com o sistema CDM, mas com os valores mais próximos ao CLJ, o que evidencia como ambas arquiteturas tiveram sua importância no desenvolvimento do controlador neural.

4.3.3 N_m e N_r

Para os indicadores técnicos do sistema, a análise se baseia em dois aspectos: a média do erro e o desvio padrão. A parâmetro η_m , representante do erro médio, não representa o erro propriamente dito, mas sim se o controlador teve alguma tendência em seu desenvolvimento, ou se o período de recuperação à condição normal deu-se de modo semelhante ao início da perturbação.

O parâmetro η_r , que representa o desvio padrão da média, este sim nos dá valores mais próprios sobre o erro, fornecendo a média do módulo do erro, informação mais palpável para se entender a capacidade do controlador em controlar o nível.

Em todos os testes, a piscina 1 e 4 são as que apresentam os piores resultados, sendo a 1 pela sua grande extensão e, assim, maior necessidade de ação do controlador, e a piscina 4 por ser o fim da cadeia, o que acarreta em ser a primeira a sofrer com saturação.

Assim sendo, os gráficos a seguir mostram os resultados para o controlador neural para todas as piscinas.

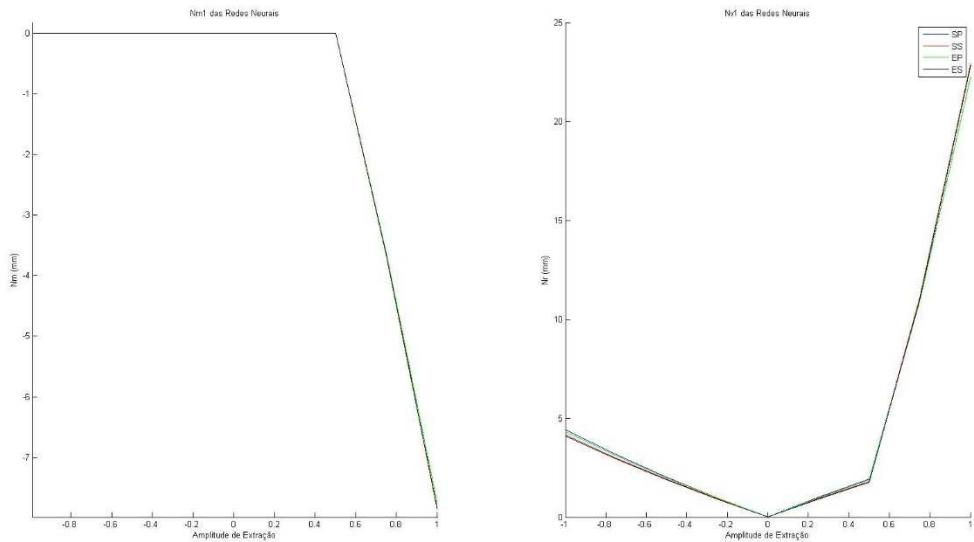


Figura 19: η_m e η_r para a piscina 1 com controlador neural

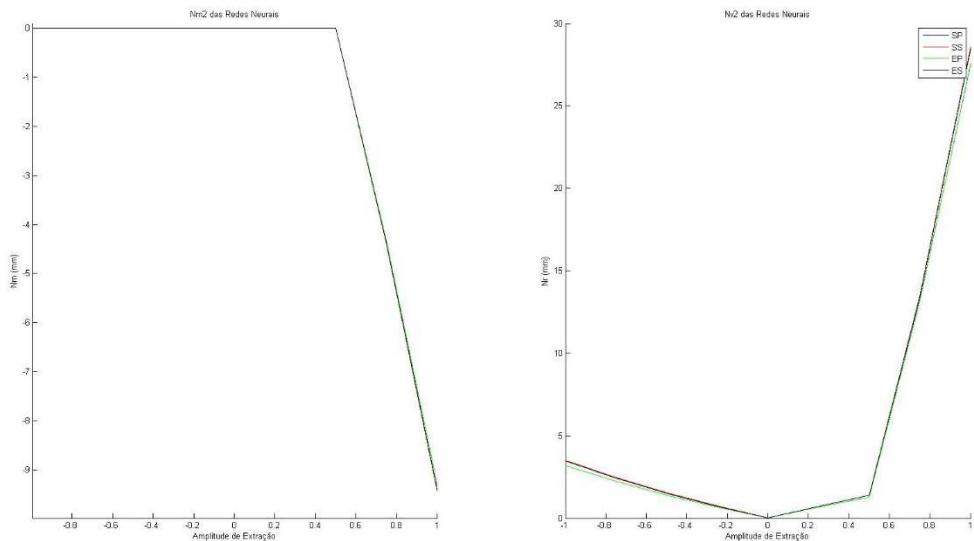


Figura 20: η_m e η_r para a piscina 2 com controlador neural

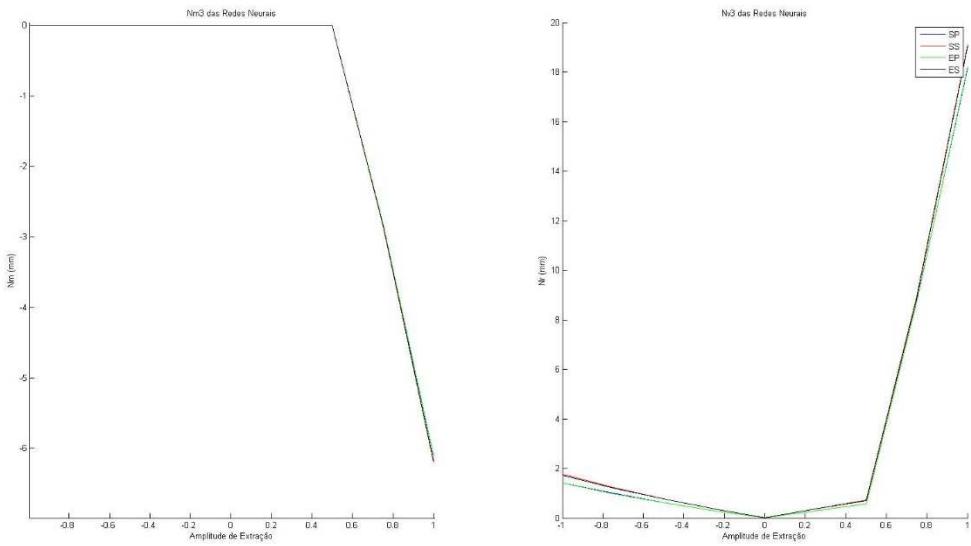


Figura 21: η_m e η_r para a piscina 3 com controlador neural

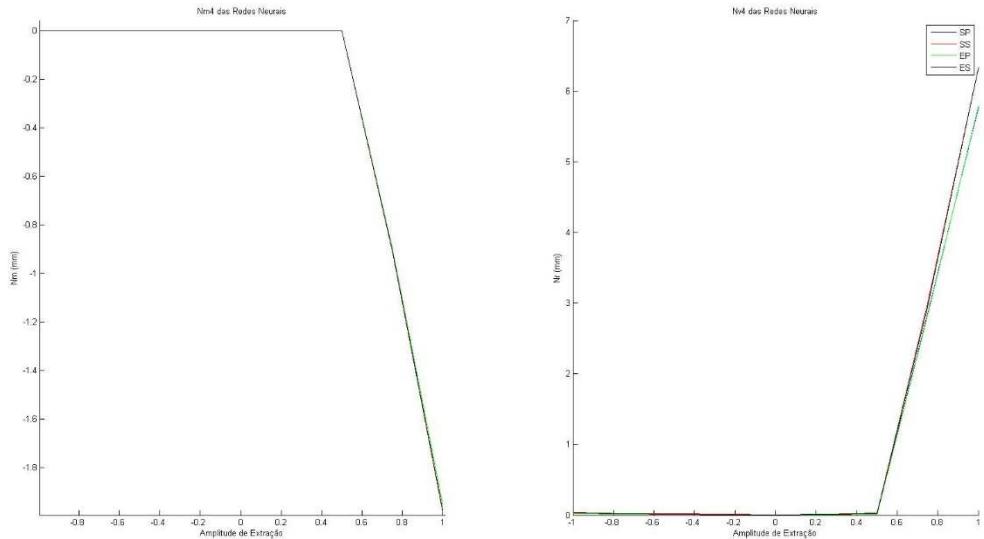


Figura 22: η_m e η_r para a piscina 4 com controlador neural

Como pode se evidenciar, todas as piscinas apresentam resultados variando apenas com a amplitude de extração, e pouco se variando com o cenário de extração. Além disso, nota-se que, apenas nos casos extremos de extração com amplitude positiva há η_m muito diferente de zero, o que é causado pela saturação inferior do controlador ou pela ação de ressonância, como dito anteriormente, entre a parte CDM e CLJ do controlador neural.

A título de comparação, serão apresentados, a seguir, os gráficos das piscinas 1 e 4 para os casos testados com os controladores originais.

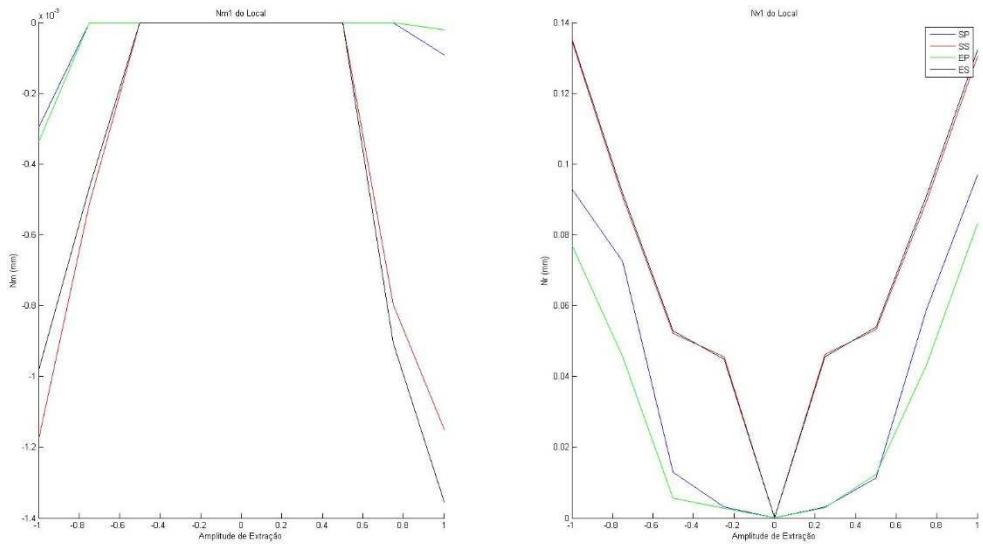


Figura 24: η_m e η_r para a piscina 1 com CLJ (Simacek, 2015)

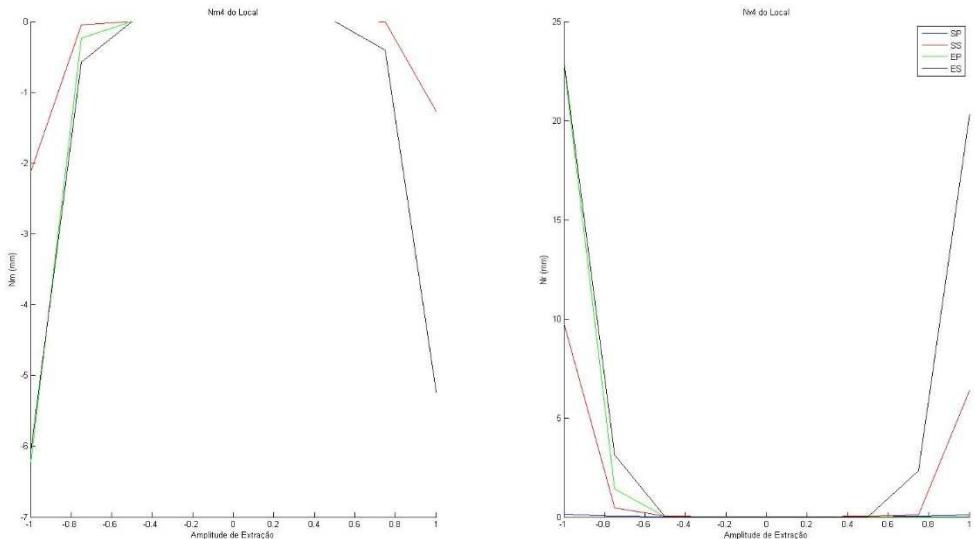


Figura 23: η_m e η_r para a piscina 4 com CLJ (Simacek, 2015)

O resultado da primeira piscina em CLJ mostra valores extremamente pequenos, mesmo de η_m e η_r , o que é o padrão para todas as piscinas, exceto a 4 que, como evidenciado, encontra-se com valores muito dispares das demais, graças a saturação que acontece no controlador.

Ainda se nota a dependência do sistema em relação ao cenário de extração, diferente do caso neural antes exemplificado. Por fim, os valores cenários onde o controlador neural tem η_m próximos a zero, apresentam-se valores próximos aos dessa arquitetura.

A seguir, os resultados para o caso CDM para as piscinas 1 e 4.

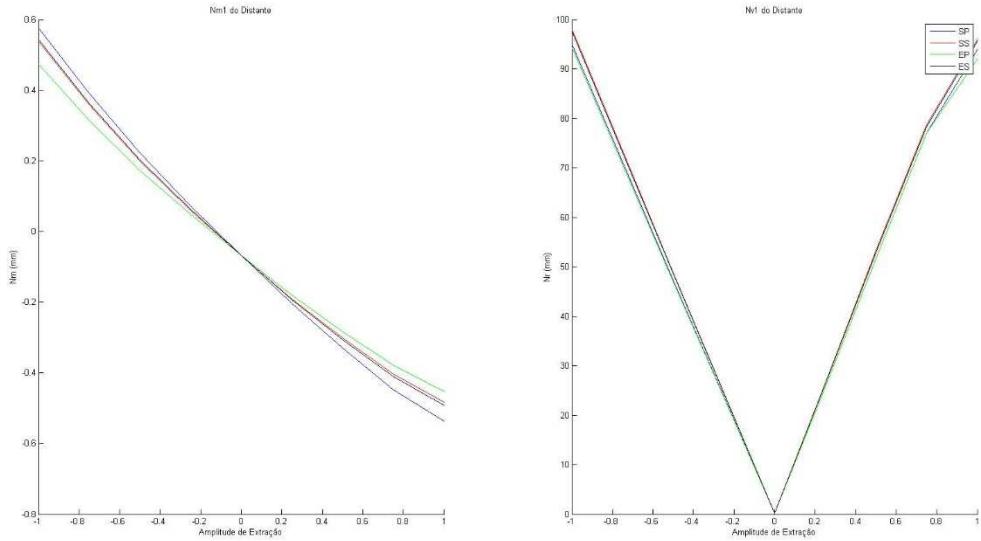


Figura 25: η_m e η_r para a piscina 1 com CDM (Simacek, 2015)

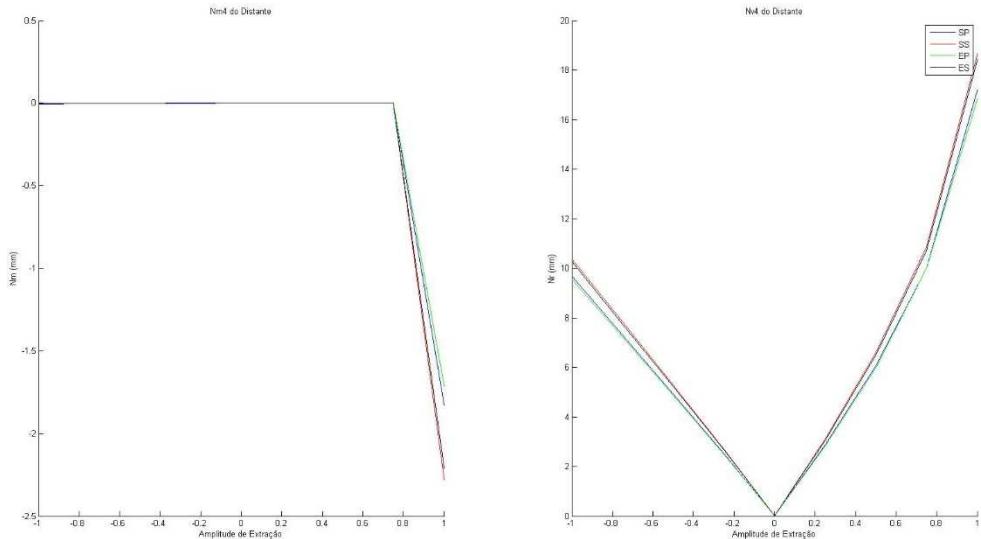


Figura 26: η_m e η_r para a piscina 4 com CDM (Simacek, 2015)

Pode-se notar maior semelhança comportamental entre o CDM e o sistema com o controlador neural, o que era esperado. Contudo, tal comportamento dá-se mais semelhante referente ao η_m , já quanto ao η_r , o desempenho do controlador neural é maior, com desvios menores, e sofre menos influência em casos de amplitude de extração negativa.

Por fim, nota-se o que era esperado, que o controlador neural, apesar de uma semelhança maior com controlador CDM, seu desempenho dá-se muito melhor, graças a ação em conjunta das arquiteturas em sua aplicação.

4.4 Discussão

Com os resultados aqui apresentados, mostrou-se o que era o objetivo dessa monografia. O sistema com controladores neurais desenvolvidos, baseados nas duas arquiteturas usuais concebidas com controladores PI clássicos, mostrou vantagens em cima das duas arquiteturas.

Mostrou-se capaz de operar autonomamente, o que não era esperado pela arquitetura CLJ, onde a entrada de fluxo de água no canal é tida como manual (Simacek, 2015). Conseguiu resultados técnicos substancialmente melhores do que a arquitetura CDM, com desvios padrão da média menores.

Contudo, o controlador neural falha na fase a qual o sistema deve voltar ao estado nominal de funcionamento, dado o fato do sistema ser superdefinido, sendo sua principal falha em relação aos outros controladores.

Por fim, vale notar que o controlador neural tem uma semelhança maior em relação ao controlador CDM do que com o CLJ, como era esperado desde sua concepção, dado o processo de treinamento desenvolvido.

5 Conclusão

Com todas as evidências dadas, o objetivo desse trabalho dá-se por conquistado, tendo em vista que foi criado um controlador baseado em redes neurais, cujo treinamento foi objetificado nos dados de duas arquiteturas de controle dispares, que conseguiu obter desempenho superior às suas arquiteturas-mãe.

Sendo pautada em noções de controle básico, cálculo e na teoria de redes neurais, o principal foco do trabalho foi o desenvolvimento do método pelo qual elas seriam treinadas.

O treinamento diferenciado, baseado na adaptação sucessiva da rede com taxas de aprendizagem diferentes de iteração para iteração, foi a base do sucesso da criação desse sistema de controle.

Apesar do controlador neural ser superdefinido, o que acarreta em múltiplas soluções para o caso nominal do problema, ele se mostrou extremamente eficiente em relação às duas outras arquiteturas.

O consumo de água, apesar de maior do que nas demais arquiteturas, pode ser contornado com alguns adendos ao controlador da primeira comporta, o que não foi o foco deste trabalho.

Por fim, o trabalho realizado dá-se por finalizado em vista de ter atingido seus objetivos.

Referências

- [1] Controlador PID sintonizado por Redes Neurais Artificiais - Ferrari, Allan - Universidade Federal do Paraná (2010)
- [2] Comparison between neural network based PI and PID controllers - Mohammed Hassan, Ganesh Kothapalli - Edith Cowan University (2010)
- [3] PID Neural Networks for Time-Delay Systems — H.L. Shu, Y. Pi - Guangzhou University (2000)
- [4] Adaptive System Control with PID Neural Networks — F. Shahrokia, M.A. Fanaeib, A.R. Arjomandzadeha - University of Sistan and Baluchestan (2009)
- [5] Control System Design — Karl Johan Åström - University of California (2002)
- [6] Neural Network Applications in Control - George William Irwin, K. Warwick, Kenneth J. Hunt - Institution of Electrical Engineers (1995)
- [7] Nabais, J. L., et al. - A multi-agent architecture for diagnosing simultaneous faults along water canals - Control Engineering Practice (2013)
- [8] Ooi, S. K. and Weyer, E. - Control Design for an irrigation channel from physical data - Control Engineering Practice 16 (2008)
- [9] Litrico, X., et al. - Modelling and PI control of an irrigation canal - *European Control Conference* (2003)
- [10] Nabais, J. L., et al. - Flexible Framework for Modeling Water Conveyance Networks - SIMULTECH (2011)
- [11] Bohn, C. and Atherton, D. P. - An analysis package comparing PID antiwindup strategies - IEEE Systems Magazine (1995)
- [12] Simacek, Pedro - Controlo Tolerante a Falhas baseado em Estratégias de Modelo Preditivo como Ferramenta para Melhoramento de Sistemas Flexíveis de Distribuição de Água em Canais de Distribuição de Água – Instituto Superior Técnico (2015)
- [13] Ogata, K - Engenharia de Controle Moderno - Person Education (2011)
- [14] Haykin, S - Redes Neurais - Princípios e Prática – Bookman (2001)