

Roberto de Freitas Cabral

Modelagem de um Sistema Hidráulico Através de Redes Neurais Artificiais

Brasil

2018

Roberto de Freitas Cabral

Modelagem de um Sistema Hidráulico Através de Redes Neurais Artificiais

Projeto de Graduação apresentado ao Curso de Engenharia Mecânica da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Engenheiro.

Universidade Federal do Rio de Janeiro – UFRJ

Curso de Engenharia Mecânica

Orientador: M. Sc. Marcelo Silva

Brasil

2018

Cabral, Roberto de Freitas

Modelagem de um Sistema Hidráulico Através de Redes Neurais Artificiais/ Roberto de Freitas Cabral. – Rio de Janeiro: UFRJ, 2018

137 p.: il. (algumas color.); 29,7 cm.

Orientador: M. Sc. Marcelo Silva

Projeto de Graduação – UFRJ/Curso de Engenharia Mecânica, 2018

Referências Bibliográficas: p. 119 – 121

1. Modelagem. 2. Sistema hidráulico. 3. Redes neurais artificiais. I. Silva, Marcelo. II. Universidade Federal do Rio de Janeiro, Curso de Engenharia Mecânica. III. Modelagem de um Sistema Hidráulico Através de Redes Neurais Artificiais.

Roberto de Freitas Cabral

Modelagem de um Sistema Hidráulico Através de Redes Neurais Artificiais

Projeto de Graduação apresentado ao Curso de Engenharia Mecânica da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Engenheiro.

Trabalho aprovado. Brasil, 7 de dezembro de 2018:

M. Sc. Marcelo Silva
Orientador

M. Sc. Matheus Ferreira de Barros
Convidado 1

M. Sc. Rudineli Demarque
Convidado 2

Brasil
2018

Agradecimentos

A Deus, que me proporcionou o dom da vida e a força necessária para a longa e árdua caminhada que percorri durante o meu curso de Engenharia Mecânica na UFRJ. Os desafios que enfrentei foram muitas vezes extenuantes, mas Ele, por meio de sua grande misericórdia, me sustentou até aqui.

Aos meus pais que com sacrifício próprio conseguiram me apoiar durante toda a minha formação.

Ao professor Marcelo Silva, que gentilmente aceitou ser o meu orientador, e por ter sido um excelente professor ao longo do meu curso, bem como um amigo sincero.

Aos professores Matheus Barros e Rudineli Demarque, que prontamente aceitaram fazer parte da banca avaliadora, além de fornecerem observações valiosas que foram incorporadas à versão final do presente trabalho.

Aos demais bons professores e funcionários da UFRJ que fazem esta instituição ter o seu bom nome e prestígio, e que se empenham diuturnamente em fazer dela um importante motor no progresso científico nacional.

A todos os amigos e colegas com os quais convivi durante a minha formação, em especial David Oliveira, Marcus Moreno e Vinícius Muniz.

À população brasileira que torna possível a existência de uma universidade pública de qualidade como a UFRJ.

*“Somos o que repetidamente fazemos.
A excelência, portanto, não é um feito,
mas um hábito.”*
Aristóteles

Resumo

A modelagem de sistemas hidráulicos constitui uma etapa imprescindível nos projetos de engenharia relacionados ao transporte de líquidos. A modelagem tradicional emprega equações, gráficos e tabelas, bem como modelos computacionais baseados em dinâmica dos fluidos computacional. Apesar do sucesso do modo tradicional, a modelagem através de redes neurais artificiais tem atraído uma atenção crescente de engenheiros e cientistas nas últimas décadas. Fornecida uma quantidade suficiente de dados durante a etapa de treinamento, as redes neurais artificiais conseguem modelar sistemas físicos com grande precisão, sem a necessidade de implementação das equações físicas do modo tradicional. Desta forma, a abordagem de redes neurais torna-se atraente na presente época, marcada pela aquisição cada vez maior de dados por empresas e organizações.

Este projeto visou criar uma rede neural artificial capaz de modelar um sistema hidráulico elevatório. Para isto, utilizou-se o software MATLAB tanto para a confecção e treinamento da rede neural, quanto para a implementação algorítmica do modelo físico-analítico tradicional. Comparações entre os dois modelos foram feitas, em que se pôde constatar que a rede neural conseguiu fornecer resultados precisos, mesmo tendo sido exposta a um conjunto de treinamento 100 vezes menor que o conjunto de avaliação, isto é, a rede foi capaz de generalizar. As respostas dadas pela rede neural artificial desviaram-se em, no máximo, 1,0% daquelas fornecidas pela modelagem tradicional para 98,39% das comparações realizadas. Além disso, 73,82% das respostas dadas pela rede situaram-se abaixo de 0,1% de desvio em relação ao modelo tradicional, e em nenhuma ocorrência o desvio superou os 4,0%.

Palavras-chave: modelagem. sistema hidráulico. redes neurais artificiais.

Abstract

The modeling of hydraulic systems is an essential step in engineering projects related to the transportation of liquids. Traditional modeling employs equations, graphs, and tables as well as computational models based on computational fluid dynamics. Despite the success of the traditional mode, modeling through artificial neural networks has attracted growing attention from engineers and scientists in recent decades. Given enough data during the training stage, artificial neural networks can model physical systems with great precision, without the need to implement the physical equations of the traditional mode. In this way, the neural network approach becomes attractive in the present era, marked by the increasing acquisition of data by companies and organizations.

This project aimed to create an artificial neural network capable of modeling a hydraulic lifting system. For this, MATLAB software was used both for the preparation and training of the neural network, and for the algorithmic implementation of the traditional physical-analytical model. Comparisons between the two models were made, in which it was possible to verify that the neural network was able to provide accurate results, even though it was exposed to a training set 100 times smaller than the evaluation set, that is, the network was able to generalize. The answers given by the artificial neural network deviated in a maximum of 1.0% from those provided by traditional modeling to 98.39% of the comparisons made. In addition, 73.82% of the responses provided by the network were below 0.1% deviation from the traditional model, and in no event did the deviation exceed 4.0%.

Keywords: modeling. hydraulic system. artificial neural networks.

Lista de ilustrações

Figura 1 – A expressão fenomenológica e a rede neural artificial são expostas às mesmas entradas e têm o mesmo objetivo. O termo “head” refere-se à carga manométrica.	26
Figura 2 – Diferentes formas de energia.	30
Figura 3 – Uma máquina de Carnot opera entre dois reservatórios a temperatura T_L e T_H . Nem toda energia é convertida em trabalho W , mas parte dela é transferida ao reservatório à temperatura T_L	32
Figura 4 – Fluidos newtonianos possuem valores constantes de viscosidade.	36
Figura 5 – A viscosidade do fluido é a causa da condição de não-escorregamento e do surgimento do perfil de velocidades.	37
Figura 6 – A velocidade média $V_{méd}$ é obtida por integração da velocidade ao longo da seção transversal ao escoamento e posterior divisão da integração pela área da seção.	41
Figura 7 – Volume de controle no qual o fluido escoar apenas através das seções 1 e 2.	42
Figura 8 – Sistema de forma arbitrária com área A e com uma pressão P incidindo sobre ele.	45
Figura 9 – A perda de pressão ΔP_L do ponto 1 para o ponto 2 de um fluido em escoamento numa tubulação é causada pelos efeitos viscosos.	52
Figura 10 – Quando um componente está presente num escoamento (como uma válvula) a queda de pressão entre 1 e 2 é maior do que na ausência deste.	53
Figura 11 – Medidas de afastamento dos pontos do perfil de rugosidade em relação à linha média do percurso de medição.	55
Figura 12 – O Diagrama de Moody.	58
Figura 13 – A curva do sistema.	60
Figura 14 – As curvas características do sistema e da máquina de fluxo e o ponto de funcionamento na interseção entre as curvas.	61
Figura 15 – Os cinco modelos computacionais que abordam o problema da computabilidade. O modelo de redes neurais (do inglês <i>neural networks</i>) é um deles.	63
Figura 16 – Um neurônio artificial usa a função f como primeira etapa de processamento interno das diferentes entradas x_i ($i = 1, \dots, n$).	64
Figura 17 – A função logística representada por várias curvas, cada uma com um parâmetro de temperatura T particular.	65
Figura 18 – A função tangente hiperbólica resulta em valores que vão de -1 à 1.	66

Figura 19 – Uma rede neural artificial genérica em que os dados de entrada (<i>inputs</i>) entram na primeira camada e são processados pela rede, resultando no dado de saída (<i>output</i>). Os pesos $w_{i,j}$ foram omitidos por questão de simplicidade.	67
Figura 20 – Um perceptron de camada única. Cada conexão dos neurônios i_n com o neurônio Ω tem um peso associado.	70
Figura 21 – Um perceptron multicamadas.	70
Figura 22 – De cima para baixo, funções aproximadas por um polinômio, por uma série de Fourier, e por uma rede neural artificial. As linhas tracejadas correspondem às funções, e as linhas contínuas, às aproximações. . . .	71
Figura 23 – O gradiente de uma função f no ponto (4,3).	72
Figura 24 – Dada uma função f usa-se o método do gradiente descendente para minimizá-la. O gradiente descendente parte de um valor e a cada iteração vai em direção a um menor valor de f	72
Figura 25 – Um perceptron multicamadas com as três camadas K, H e L em destaque.	76
Figura 26 – Diagrama mostrando os “caminhos” dos algoritmos de retropropagação do erro com um pequeno (à esq.) e um grande (à dir.) coeficiente de aprendizado.	77
Figura 27 – Diagrama do sistema hidráulico escolhido.	79
Figura 28 – Válvulas presentes no sistema hidráulico: a) Válvula gaveta e b) Válvula globo.	80
Figura 29 – Primeiras colunas da <i>matrizEntradas</i> usada na fase de treinamento da <i>RNA</i>	88
Figura 30 – Últimas colunas da <i>matrizEntradas</i> usada na fase de treinamento da <i>RNA</i>	88
Figura 31 – A <i>matrizEntradas</i> pós-embaralhamento.	89
Figura 32 – Colunas iniciais da <i>matrizAlvos</i> usada na fase de treinamento da <i>RNA</i>	90
Figura 33 – A <i>struct ts</i> armazena valores resultantes da normalização.	91
Figura 34 – Colunas iniciais da <i>matrizEntradasNorm</i> usada na fase de treinamento da <i>RNA</i>	92
Figura 35 – Colunas iniciais da <i>matrizAlvosNorm</i> usada na fase de treinamento da <i>RNA</i>	92
Figura 36 – Botão para a ferramenta <i>Neural Net Fitting (nftool)</i> no <i>MATLAB</i>	92
Figura 37 – Seleção dos padrões de entrada (<i>Inputs</i>) e dos alvos (<i>Targets</i>) na ferramenta <i>Neural Net Fitting</i>	93
Figura 38 – Divisão dos dados de amostra em três grupos.	94
Figura 39 – Escolha do número de neurônios na camada oculta.	94
Figura 40 – Escolha do algoritmo Levenberg-Marquardt para o treinamento da rede neural artificial.	94

Figura 41 – Monitoramento do treinamento da rede neural artificial.	95
Figura 42 – Pesos das conexões dos neurônios i da camada de entrada aos neurônios j da camada oculta.	96
Figura 43 – Vieses b dos neurônios da camada oculta.	96
Figura 44 – Pesos das conexões dos neurônios j da camada oculta ao neurônio único da camada de saída.	96
Figura 45 – Seção inicial da <i>matrizEntradas</i> usada na fase de avaliação da <i>RNA</i> . . .	98
Figura 46 – Seção inicial da <i>matrizEntradas</i> embaralhada usada na fase de avaliação da <i>RNA</i>	99
Figura 47 – Seção inicial da <i>matrizEntradasNorm</i> usada na fase de avaliação da <i>RNA</i> . .	99
Figura 48 – Colunas iniciais da <i>matrizHeads_fenom</i> usada na fase de avaliação da <i>RNA</i>	100
Figura 49 – Colunas iniciais da <i>matrizHeads_RNA</i>	101
Figura 50 – Colunas iniciais da <i>matrizHeads_RNA_desnorm</i>	101
Figura 51 – Diagrama resumindo o processo de obtenção das saídas pelos dois métodos.	102
Figura 52 – Curvas do sistema geradas pela expressão fenomenológica (linha verme- lha) e pela rede neural artificial (linha azul).	102
Figura 53 – Ilustração de como serão as plotagens para comparação entre os dois modelos. A linha vermelha corresponde à expressão fenomenológica, e a azul, à rede neural artificial.	105
Figura 54 – Colunas iniciais da <i>matrizHeads_fenom</i>	109
Figura 55 – Colunas iniciais da <i>matrizHeads_RNA_desnorm</i>	109
Figura 56 – Regressão linear entre as saídas dos dois modelos para todo o conjunto de amostras.	111
Figura 57 – Histograma de desvios relativos. Cada barra tem uma largura de 0,1%. .	112
Figura 58 – Curvas do sistema usando os dois modelos (conj = 1).	113
Figura 59 – Curvas do sistema usando os dois modelos (conj = 7).	114
Figura 60 – Curvas do sistema usando os dois modelos (conj = 7), com o eixo dos <i>heads</i> em maior escala.	114
Figura 61 – Curvas do sistema usando os dois modelos para diferentes valores de parâmetros.	115

Lista de tabelas

Tabela 1 – Valores de rugosidade absoluta de tubos de PVC encontrados em diferentes referências.	55
Tabela 2 – Comparação entre aproximações explícitas da equação de Colebrook. .	57
Tabela 3 – Variáveis e constantes físicas.	81
Tabela 4 – Valores das constantes.	81
Tabela 5 – Faixas de valores das variáveis.	82
Tabela 6 – Valores de K para as válvulas globo e gaveta em função do diâmetro. As válvulas são flangeadas.	82
Tabela 7 – Contabilização dos desvios relativos para cada faixa de desvios. . . .	110
Tabela 8 – Medidas descritivas dos desvios relativos.	112
Tabela 9 – Valores dos parâmetros usados nas plotagens das curvas do sistema. .	113

Lista de abreviaturas e siglas

MAPE	<i>Mean Absolute Percentage Error</i>
RNA	Rede Neural Artificial
SC	Superfície de Controle
TTR	Teorema de Transporte de Reynolds
VC	Volume de Controle

Sumário

1	INTRODUÇÃO	25
1.1	Motivação	25
1.2	Objetivo	26
1.3	Organização do trabalho	26
2	CONSIDERAÇÕES DE MECÂNICA DOS FLUIDOS	29
2.1	Pressão absoluta e manométrica	29
2.2	Energia	29
2.2.1	Energia total de um sistema	29
2.2.2	Energia mecânica	30
2.2.3	Energia interna	31
2.3	A Primeira Lei da Termodinâmica	31
2.4	A Segunda Lei da Termodinâmica	32
2.4.1	Atrito e perda de energia	32
2.5	Energia de um fluido em escoamento	33
2.6	Escoamentos laminar e turbulento	33
2.7	Escoamentos interno e externo	34
2.8	Regime de escoamento permanente	34
2.9	Escoamento incompressível	35
2.10	Viscosidade	35
2.10.1	Viscosidade dinâmica e cinemática	35
2.11	Condição de não-escorregamento e perfil de velocidades	36
2.12	O número de Reynolds	37
2.13	Conservação de massa	38
2.14	Vazões em massa e em volume	40
2.15	Conservação de energia	44
2.16	A equação de Bernoulli	50
2.17	Perda de pressão e perda de carga	51
2.18	Rugosidade	54
2.19	Fator de atrito de Darcy-Weisbach	55
2.20	O diagrama de Moody	57
2.21	Versão final da equação da energia	58
2.22	O head do sistema	59
2.23	A curva do sistema	60
3	CONCEITOS COMPUTACIONAIS	63

3.1	Os principais paradigmas computacionais	63
3.2	O neurônio artificial	63
3.3	Componentes do neurônio artificial	64
3.4	Redes neurais artificiais	67
3.5	Capacidade de generalização de uma RNA	67
3.6	Modos de aprendizado das RNAs	68
3.7	O perceptron	69
3.7.1	O perceptron de camada única - SLP	70
3.7.2	O perceptron multicamadas - MLP	70
3.8	Minimização de uma função através do gradiente descendente . . .	71
3.9	A lei de aprendizado de Hebb	73
3.10	A função erro	73
3.11	Algoritmos de treinamento de redes neurais	74
3.11.1	A Regra Delta	75
3.11.2	O algoritmo de Levenberg-Marquardt	77
4	PROCEDIMENTOS E METODOLOGIA	79
4.1	Definição do sistema hidráulico	79
4.2	Parâmetros, constantes e variáveis	80
4.3	Delimitações adicionais	83
4.4	Conceitos e definições	83
4.4.1	Conjunto de parâmetros	83
4.4.2	Resolução, número de amostras e quantidade de conjuntos de parâmetros .	84
4.5	Fase de treinamento da RNA	86
4.5.1	Obtenção dos padrões de entrada p	86
4.5.2	Obtenção dos alvos t	89
4.5.3	Normalização dos padrões de entrada e dos alvos	90
4.5.4	Treinamento da rede neural artificial	92
4.6	Fase de avaliação da RNA	96
4.6.1	Obtenção de novas amostras	96
4.6.2	Cálculos dos valores de saída da expressão fenomenológica	99
4.6.3	Cálculos dos valores de saída da rede neural artificial	100
4.6.4	Comparações entre a rede neural artificial e a expressão fenomenológica . .	101
5	RESULTADOS E DISCUSSÃO	109
5.1	Avaliação ao longo de todo o conjunto de amostras	109
5.2	Avaliação em específicos conjuntos de parâmetros	112
6	CONCLUSÕES	117

REFERÊNCIAS	119
-----------------------	-----

ANEXOS	123
--------	-----

ANEXO A – A FUNÇÃO <i>FN_HEAD_SISTEMA</i> E SUAS AUXILI-	
ARES	
	125
A.1	A função <i>fn_head_sistema</i>
	125
A.2	A função <i>fn_peso_gamma</i>
	127
A.3	A função <i>fn_K_cotov_90</i>
	127
A.4	A função <i>fn_K_gaveta</i>
	128
A.5	A função <i>fn_K_globo</i>
	128
A.6	A função <i>fn_area_escoam</i>
	128
A.7	A função <i>fn_vel_escoam_c</i>
	129
A.8	A função <i>fn_Re</i>
	129
A.9	A função <i>fn_f_Darcy</i>
	130
A.10	A função <i>fn_h_c</i>
	130
A.11	A função <i>fn_h_l</i>
	131
A.12	A função <i>fn_h_f</i>
	131
A.13	A função <i>fn_energ_mec</i>
	131
A.14	A função <i>fnMat_interpol_linear</i>
	132
A.15	A função <i>fnMat_interpol_faixas</i>
	133
Índice	
	135

1 Introdução

1.1 Motivação

De acordo com Çengel e Cimbala (2012, p. 278):

Embora a teoria do escoamento de fluidos seja razoavelmente bem compreendida, as soluções teóricas são obtidas apenas para alguns poucos casos simples, como o escoamento laminar totalmente desenvolvido em um tubo circular. Assim, devemos nos basear nos resultados experimentais e nas relações empíricas na maioria dos problemas de escoamento de fluidos em vez de em soluções analíticas fechadas.

Adicionalmente, White (2010, p. 353) afirma:

A verdade é que ainda não existe análise geral do escoamento de um fluido. Há várias dezenas de soluções particulares conhecidas, existem muitas soluções aproximadas obtidas em computador e uma quantidade enorme de dados experimentais. Há bastante teoria disponível se desprezarmos efeitos importantes como a viscosidade e a compressibilidade (Capítulo 8), mas não existe teoria geral, e talvez jamais venha a existir.

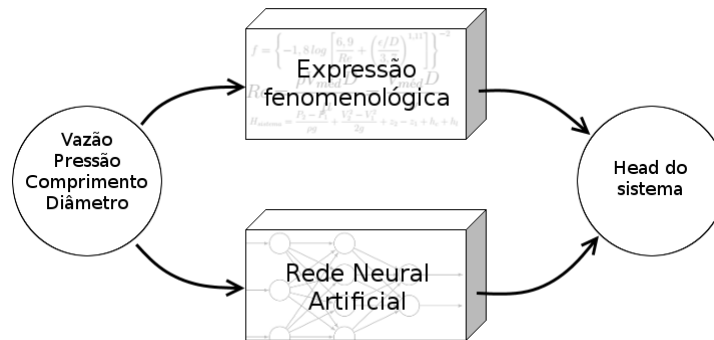
Nos últimos trechos transcritos, os autores parecem reconhecer as limitações dos modelos analíticos desenvolvidos no âmbito da mecânica dos fluidos. Dessa forma, o emprego das redes neurais artificiais se torna atraente, já que esse tipo abordagem é adequado aos casos em que não é possível modelar explícita e analiticamente determinados fenômenos físicos com total precisão. Enquanto que a modelagem físico-analítica tradicional, com fórmulas matemáticas, gráficos e tabelas requer um amplo conhecimento dos fenômenos envolvidos, a modelagem através de redes neurais artificiais se faz de maneira automática¹. Uma rede neural artificial suficientemente treinada pode equivaler aos modelos analíticos tradicionais no que diz respeito aos resultados fornecidos pela primeira. Essa ideia é ilustrada na Figura 1, em que se mostra um mesmo sistema hidráulico sendo modelado tanto pela tradicional expressão fenomenológica quanto através de uma rede neural artificial. Ambos os modelos recebem as mesmas entradas, em relação às quais fornecem as mesmas saídas.

Atualmente empresas e organizações estão coletando dados sobre os mais diversos processos, o que inclui os valores dos parâmetros físicos presentes no escoamento em tubulações. Desta forma, com uma quantidade de dados disponíveis cada vez maior, torna-se possível o uso da abordagem de redes neurais artificiais para modelar um sistema

¹ Se providos dados em quantidade suficiente.

hidráulico potencialmente mais condizente com a realidade do que os tradicionais modelos físicos.

Figura 1 – A expressão fenomenológica e a rede neural artificial são expostas às mesmas entradas e têm o mesmo objetivo. O termo “head” refere-se à carga manométrica.



Fonte: Elaborado pelo autor.

1.2 Objetivo

O objetivo geral deste trabalho é modelar um sistema hidráulico fazendo uso de redes neurais artificiais (abrev.: *RNA*) e comparar os resultados fornecidos por esta com aqueles dados pela modelagem tradicional.

Como objetivos específicos, este trabalho tratará de modelar um sistema hidráulico por meio de equações tradicionais de mecânica dos fluidos a fim de usar este modelo tradicional para a geração de dados e comparação com a modelagem feita através da rede neural artificial. Além disto, se buscará implementar um método de comparação entre os dois modelos, sendo este método comparativo constituído por algoritmos que resultarão em gráficos e tabelas que permitirão a avaliação.

1.3 Organização do trabalho

Primeiramente, o Capítulo 1 apresentará uma introdução ao presente trabalho. Neste capítulo, se discutirão a motivação e os objetivos deste texto, bem como a sua organização.

Em seguida, no Capítulo 2, serão abordados aspectos teóricos relacionados aos sistemas hidráulicos; esta teoria desempenhará importante papel nos capítulos seguintes. As equações de mecânica dos fluidos serão apresentadas de maneira sistemática e as deduções de algumas equações serão feitas ao longo do texto.

No Capítulo 3, serão discutidos os aspectos computacionais, isto é, as redes neurais artificiais, indo desde os seus elementos fundamentais até alguns algoritmos de treinamento.

No Capítulo 4 será definido o sistema hidráulico a ser modelado, com todos os seus parâmetros. Também neste capítulo se modelará o sistema hidráulico usando algoritmos escritos no *MATLAB*, sendo o conjunto destes algoritmos aquilo que fornecerá os dados necessários ao treinamento da rede neural. Obtidos estes valores, será iniciado o processo de treinamento da rede neural artificial, também descrito no mesmo capítulo (seção 4.5). Em seguida, será abordada a fase de avaliação da rede (seção 4.6), em que serão comparados os valores fornecidos pela rede àqueles resultantes da expressão fenomenológica.

O Capítulo 5 fará uma breve abordagem dos resultados obtidos pela rede neural artificial e discutirá como eles se aproximam daqueles fornecidos pela expressão fenomenológica.

Finalmente, o Capítulo 6 fará a conclusão do trabalho além de fornecer diversas recomendações para trabalhos futuros.

2 Considerações de Mecânica dos Fluidos

2.1 Pressão absoluta e manométrica

Como afirmado por Çengel e Cimbala (2012, p. 57):

A pressão real em determinada posição é chamada de *pressão absoluta*, e é medida com relação ao vácuo absoluto (ou seja, a pressão absoluta zero). A maioria dos dispositivos de medição da pressão, porém, é calibrada para ler o zero na atmosfera e, assim, ela indica a diferença entre a pressão absoluta e a pressão atmosférica local. Essa diferença é chamada de *pressão manométrica*.

Dessa afirmação de Çengel e Cimbala (2012), pode-se relacionar as diferentes formas de pressão citadas através da Equação 2.1:

$$P_{man} = P_{abs} - P_{atm} \quad (2.1)$$

em que P_{man} é a pressão manométrica, P_{abs} é a pressão absoluta, e P_{atm} é a pressão atmosférica. As unidades geralmente usadas para medição de pressão são o Pascal (Pa) e a atmosfera (atm).

Neste trabalho, se considerará o valor de pressão atmosférica igual a $101,325 \text{ kPa}$ ou 1 atm , que é a pressão absoluta exercida pela atmosfera ao nível do mar e a uma temperatura de $20^\circ C$.

2.2 Energia

Segundo Tipler e Mosca (2009, p. 171), a “energia de um sistema é uma medida de sua habilidade em realizar trabalho”. Esse ente físico existe sob diversas formas, tais como: energia mecânica, energia térmica, energia química, etc.

2.2.1 Energia total de um sistema

De acordo com Çengel e Cimbala (2012, p. 36), “a energia existe sob formas numerosas tais como térmica, mecânica, cinética, potencial, elétrica, magnética, química e nuclear, e a soma de todas elas constitui a energia total de um sistema”. Neste trabalho,

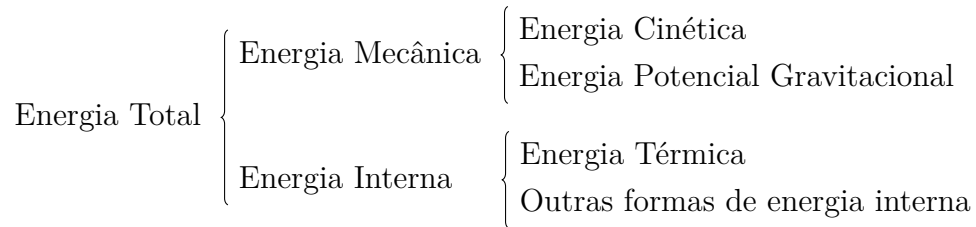
a energia total será representada pela letra E (ou e , se for medida numa base de massa unitária), sendo dada pela Equação 2.2:

$$E = E_{mec} + E_{int} \quad (2.2)$$

em que E é a energia total do sistema, E_{mec} é a energia mecânica, e E_{int} é a energia interna. Todas as energias estão expressas em *joules* (J).

Tanto a energia mecânica E_{mec} quanto a energia interna E_{int} possuem subdivisões, conforme mostrado na Figura 2.

Figura 2 – Diferentes formas de energia.



Fonte: Elaborado pelo autor.

2.2.2 Energia mecânica

Çengel e Cimbala (2012, p. 156) afirmam que:

A energia mecânica pode ser definida como a forma de energia que pode ser convertida direta e completamente em trabalho mecânico por um dispositivo mecânico ideal como, por exemplo, uma turbina ideal.

A energia mecânica de um sistema, E_{mec} , é a soma de suas energias cinética, E_{cin} , e energia potencial, E_{pot} , como definido na Equação 2.3:

$$E_{mec} = E_{cin} + E_{pot} \quad (2.3)$$

A energia cinética é atribuída a um sistema de massa m que se movimenta em relação a um dado referencial inercial, sendo expressa pela Equação 2.4:

$$E_{cin} = \frac{1}{2}mV^2 \quad (2.4)$$

onde E_{cin} é a energia cinética (em J), m é a massa do sistema (em kg), e V é a sua velocidade (em m/s).

A energia potencial gravitacional de um sistema surge devido à presença de um corpo num campo gravitacional, como o terrestre, por exemplo. Ao sistema corpo-Terra, então, se atribui uma energia potencial gravitacional dada pela Equação 2.5:

$$E_{pot} = mgz \quad (2.5)$$

onde E_{pot} é a energia potencial (em J), m é a massa do corpo (em kg), g é a constante gravitacional¹ (em m/s^2), e z é a altura em relação a um referencial especificado (em m).

2.2.3 Energia interna

Um sistema que possua uma certa quantidade de massa possui uma determinada quantidade de energia, E_{int} , seja devido às vibrações de suas moléculas ou devido à própria organização estrutural das mesmas. Nas palavras de Tipler e Mosca (2009, p. 600), “a energia interna de um corpo é sua energia total no referencial de seu centro de massa.” Apesar de existirem várias formas de energias internas, apenas a energia térmica será considerada no presente trabalho.

A energia térmica de um sistema é caracterizada pelo grau de agitação de suas moléculas constituintes, e pode ser medida pela temperatura do sistema. Quanto maior for a temperatura de um sistema, pode-se dizer que maior será a sua energia térmica².

2.3 A Primeira Lei da Termodinâmica

Como exposto por Tipler e Mosca (2009, p. 607), a Primeira Lei da Termodinâmica “é, simplesmente, um enunciado da lei de conservação da energia”. A primeira lei afirma que, para um dado sistema, a sua energia interna sofrerá variação se, e somente se, energia cruzar as fronteiras do sistema nas formas de calor e/ou trabalho. Expressando essa lei matematicamente, tem-se a Equação 2.6:

$$\Delta E_{int} = Q_{entra} + W_{sobre} \quad (2.6)$$

onde ΔE_{int} é a variação da energia interna do sistema (em J), Q_{entra} é a quantidade de energia que entra no sistema em forma de calor (em J), e W_{sobre} é a quantidade de energia que entra no sistema em forma de trabalho (em J).

¹ Neste trabalho o valor dessa constante será assumido como $9,81m/s^2$, que equivale ao valor médio da aceleração da gravidade na superfície da Terra.

² Excetuando-se os pontos em que ocorrem mudanças de fase. Nestes pontos, a temperatura do sistema não varia, mesmo com o aumento da energia interna.

2.4 A Segunda Lei da Termodinâmica

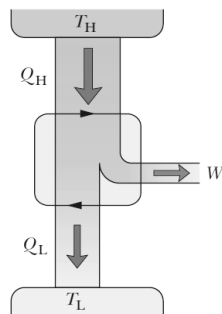
Tipler e Mosca (2009, p. 638) enunciam a Segunda Lei da Termodinâmica com o enunciado para máquinas térmicas:

É impossível para uma máquina térmica, operando em um ciclo, produzir como único efeito o de retirar calor de um único reservatório e realizar uma quantidade equivalente de trabalho.

Desse enunciado deduz-se que a transferência de calor para a realização de trabalho implica numa perda de energia, ou seja, que nem toda energia empregada para a realização de trabalho será usada, mas que uma parcela da mesma será dissipada para a vizinhança sem a possibilidade de posterior reaproveitamento, ou seja, “não existe uma série de processos cujo único resultado seja a conversão total em trabalho da energia contida em uma fonte de calor”. (HALLIDAY; RESNICK; WALKER, 2009, p. 257).

A Figura 3 mostra uma máquina de Carnot em que parte da energia recebida em forma de calor Q_H é convertida em trabalho W , enquanto que o restante da energia é transferida para um reservatório frio a uma temperatura T_L sob a forma de calor Q_L .

Figura 3 – Uma máquina de Carnot opera entre dois reservatórios a temperatura T_L e T_H . Nem toda energia é convertida em trabalho W , mas parte dela é transferida ao reservatório à temperatura T_L .



Fonte: Halliday, Resnick e Walker (2009, p. 255).

2.4.1 Atrito e perda de energia

O atrito mecânico em um sistema causa a elevação de temperatura do mesmo. Se esse sistema estiver inicialmente em equilíbrio térmico com a vizinhança, após a atrição do sistema é provável que surja uma diferença de temperaturas entre o sistema e a sua vizinhança. Essa diferença de temperaturas, por sua vez, resultará numa transferência líquida de energia térmica do sistema para a vizinhança, e a este processo de transferência de energia dá-se o nome de *calor*.

Pela segunda lei da termodinâmica, considera-se que essa energia transferida em forma de calor para a vizinhança não pode ser totalmente convertida em trabalho. Sendo assim, considera-se que parte desse calor é uma perda de energia, e, retroativamente, considera-se que o atrito está associado à perda energética, já que a causa inicial desse calor foi o próprio atrito mecânico.

2.5 Energia de um fluido em escoamento

Para os fins deste trabalho, serão consideradas apenas as formas mecânicas de energia de um sistema (energias *cinética* e *potencial gravitacional*) e uma forma adicional de energia denominada *energia de escoamento*. A energia térmica será referenciada apenas no âmbito da perda de carga.

As energias cinética e potencial já foram abordadas na seção 2.2. Além dessas energias, de acordo com Çengel e Cimbala (2012, p. 157), em mecânica dos fluidos ainda se considera uma terceira forma de energia associada à energia mecânica do fluido que é a energia de escoamento, dada pela Equação 2.7:

$$E_{esc} = PV \quad (2.7)$$

onde E_{esc} é a energia de escoamento (em J), P é a pressão (em Pa), e V é o volume (em m^3).

Considerando um fluido em escoamento, pode-se dizer que cada unidade de massa desse fluido possuirá uma energia e dada pela Equação 2.8 como:

$$e = \frac{1}{m}(E_{esc} + E_{cin} + E_{pot}) = \frac{P}{\rho} + \frac{1}{2}V^2 + gz \quad (2.8)$$

onde m é a massa do elemento de fluido (em kg), E_{esc} é a energia de escoamento (em J), E_{cin} é a energia cinética (em J), E_{pot} é a energia potencial (em J), P é a pressão (em Pa), ρ é a massa específica do fluido (em kg/m^3), V é a velocidade do elemento de fluido (em m/s), g é a constante gravitacional (em m/s^2) e z é a altura em relação a algum referencial (em m).

2.6 Escoamentos laminar e turbulento

Dependendo de como as partículas de fluido se comportam num escoamento, pode-se dizer que alguns escoamentos são laminares e outros, turbulentos.

Os escoamentos laminares caracterizam-se por uma maior ordenação. Nesse tipo de escoamento, as camadas do fluido (ou lâminas) são espacialmente dispostas de tal forma

que uma camada não interpenetra na outra. Isto é, se uma partícula de fluido pertence a uma camada num ponto, aquela permanecerá na mesma camada num ponto mais adiante no sentido do escoamento. Nas palavras de Brunetti (2008, p. 69), “escoamento laminar é aquele em que as partículas se deslocam em lâminas individualizadas, sem trocas de massa entre elas”.

Já os escoamentos turbulentos possuem uma característica mais caótica e imprevisível. Essa imprevisibilidade decorre do fato de as camadas de fluido se “embaralharem” quando submetidas a determinadas condições de escoamento. Num regime turbulento, portanto, não é possível ter certeza da posição de uma mesma partícula de fluido em diferentes instantes de tempo. De acordo com Brunetti (2008, p. 69), “escoamento turbulento é aquele em que as partículas apresentam um movimento aleatório macroscópico, isto é, a velocidade apresenta componentes transversais ao movimento geral do conjunto do fluido”.

O parâmetro que se usa para diferenciar os dois tipos de escoamento citados é o número de Reynolds que será discutido na seção 2.12.

2.7 Escoamentos interno e externo

Na palavras de White (2010, p. 358), “tanto o escoamento laminar quanto o escoamento turbulento podem ser internos, isto é, ‘limitados’ por paredes, ou externos e não limitados”.

O *escoamento interno* é caracterizado por um fluido escoando num canal delimitado por superfícies sólidas. Já o *escoamento externo* é caracterizado pelo fluido escoando *sobre* uma superfície, não estando o mesmo delimitado por fronteiras.

Como exemplo de escoamento *interno* pode-se citar a água escoando nas tubulações de uma residência, e um exemplo típico de um escoamento *externo* é o de uma esfera imersa numa corrente de fluido.

2.8 Regime de escoamento permanente

O presente trabalho considera que o regime de escoamento no interior da tubulação é do tipo *permanente*. De acordo com Çengel e Cimbala (2012, p. 10), “o termo *em regime permanente* implica não haver mudança com o passar do tempo”. Dito em outras palavras, cada parâmetro (ex.: velocidade do escoamento) em cada ponto possui um valor constante, invariável. A única variação que pode ocorrer é em relação ao *espaço*, ou seja, o escoamento pode não ser *uniforme*. Assim, num determinado ponto o valor do parâmetro permanece constante, porém esse mesmo valor pode variar de um ponto para outro. Essa consideração

de invariabilidade temporal pode ser expressa pela Equação 2.9:

$$\frac{d\varphi(x, y, z)}{dt} = 0 \quad (2.9)$$

onde φ representa qualquer parâmetro cujo valor é medido num ponto específico do espaço, e $d\varphi/dt$ representa a taxa de variação deste parâmetro φ com o tempo.

2.9 Escoamento incompressível

Para a maioria dos problemas de engenharia que envolvem mecânica dos fluidos, pode-se considerar que a massa específica de um líquido é constante, independentemente de variações da pressão ou velocidade. Para pressões relativamente baixas, a massa específica do líquido é considerada constante. Segundo Çengel e Cimbala (2012, p. 9), “uma pressão de 210 atm atuando sobre água líquida causa mudança no valor da densidade da água líquida a 1 atm de somente 1%”. Como neste trabalho as pressões utilizadas são muito menores que o valor de 210 atm, a incompressibilidade é assumida tanto na parte teórica quanto nos cálculos.

2.10 Viscosidade

Nas palavras de White (2010, p. 37), “a viscosidade é uma medida quantitativa da resistência de um fluido ao escoamento. Mais especificamente, ela determina a taxa de deformação do fluido que é gerada pela aplicação de uma dada tensão de cisalhamento.” A viscosidade também pode ser entendida como uma medida da *fluidéz* de um determinado fluido, surgindo esta propriedade em decorrência das forças coesivas entre as moléculas do mesmo. Quanto mais significativas forem essas forças coesivas, mais viscoso o fluido será.

A viscosidade é responsável pelo atrito existente entre as diferentes camadas adjacentes de um fluido em escoamento. Para fluidos com maiores valores de viscosidade, a influência que uma camada terá sobre uma outra tenderá a ser maior do que em um fluido de menor viscosidade. Esse fato faz com que fluidos com maiores valores de viscosidade tendam a ter um escoamento mais ordenado (laminar) do que outros fluidos com pequenos valores de viscosidade, que podem passar de um regime laminar para um regime turbulento com maior facilidade.

2.10.1 Viscosidade dinâmica e cinemática

A *viscosidade dinâmica* (ou *absoluta*) μ (em $Pa.s$) é medida como a taxa de deformação em relação ao tempo que uma tensão cisalhante τ pode causar num fluido,

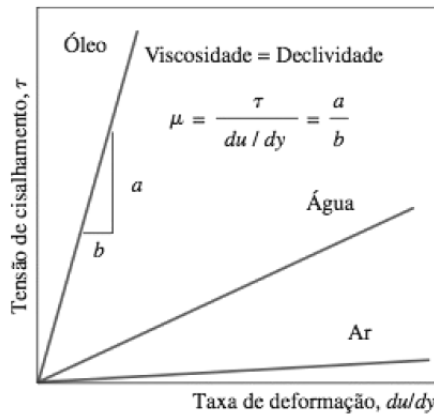
sendo dada por Çengel e Cimbala (2012, p. 42) como a Equação 2.10:

$$\mu = \frac{\tau}{du/dy} \quad (2.10)$$

onde τ é a tensão de cisalhamento (em Pa) e du/dy é uma medida da taxa de deformação do fluido (em s^{-1}) conhecida também como *gradiente de velocidades*.

No presente trabalho serão considerados apenas *fluidos newtonianos*, que são aqueles cujos valores de viscosidade não variam em função da tensão de cisalhamento τ . A Figura 4 mostra uma comparação de três fluidos newtonianos. Aqueles com maior viscosidade apresentam menores taxas de deformação du/dy quando submetidos a uma mesma tensão de cisalhamento τ .

Figura 4 – Fluidos newtonianos possuem valores constantes de viscosidade.



Fonte: Çengel e Cimbala (2012, p. 42).

Além da viscosidade dinâmica, μ , em mecânica dos fluidos também se faz conveniente, em muitos casos, o uso da *viscosidade cinemática*, ν (em m^2/s), que é, de acordo com White (2010, p. 40), dada pela Equação 2.11:

$$\nu = \frac{\mu}{\rho} \quad (2.11)$$

em que ρ refere-se à massa específica do fluido (em kg/m^3).

2.11 Condição de não-escorregamento e perfil de velocidades

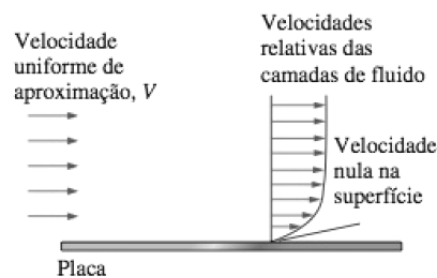
Nas palavras de White (2010, p. 49):

Quando um escoamento de fluido é limitado por uma superfície sólida, as interações moleculares fazem o fluido, em contato com a superfície, buscar o equilíbrio de quantidade de movimento e energia com tal superfície.

Todos os líquidos estão essencialmente em equilíbrio com as paredes de contato. Todos os gases também estão, exceto em condições muito rarefeitas. Então, excluindo os gases rarefeitos, todos os fluidos em um ponto de contato com um sólido assumem a velocidade e a temperatura dessa superfície.

Um fluido que escoa de maneira adjacente a uma superfície sólida impermeável (ex.: um líquido escoando no interior de um duto) apresenta a característica de que as partículas do fluido adjacentes à superfície sólida apresentam velocidade nula, isto é, não “escorregam”. Essa condição de não-escorregamento surge em decorrência da viscosidade do próprio fluido. A camada cuja velocidade é nula desacelera uma outra camada de fluido adjacente devido às forças viscosas entre ambas; esse mesmo processo se estende a todas as demais camadas em escoamento causando assim a desaceleração³ das camadas adjacentes ao longo do eixo transversal ao sentido do escoamento. Isso acarreta no surgimento de um *perfil de velocidades* cuja ilustração pode ser vista na Figura 5.

Figura 5 – A viscosidade do fluido é a causa da condição de não-escorregamento e do surgimento do perfil de velocidades.



Fonte: Çengel e Cimbala (2012, p. 6).

2.12 O número de Reynolds

Em 1883, Osborne Reynolds, professor de engenharia britânico, formulou o que mais tarde veio a ser denominado *número de Reynolds* (REYNOLDS, 1883). Este número é um parâmetro adimensional que indica a relação entre as forças inerciais e as forças viscosas do escoamento, sendo dado pela seguinte Equação 2.12:

$$Re = \frac{\rho V_{méd} D}{\mu} = \frac{V_{méd} D}{\nu} \quad (2.12)$$

onde Re é o número de Reynolds (adim.), ρ é a massa específica do fluido (em kg/m^3), $V_{méd}$ é a velocidade média do fluido em escoamento (em m/s), D é um comprimento

³ Essa variação de velocidade é espacial e não temporal. Como foi dito na seção 2.8, o valor de uma propriedade do escoamento num ponto do espaço não varia com o tempo.

característico da geometria (em m), μ é a viscosidade dinâmica do fluido (em $Pa.s$), e ν é a sua viscosidade cinemática (em m^2/s).

Quanto *maior* for o número de Reynolds, mais preponderantes serão as forças inerciais em relação às forças viscosas, o que vai resultar em flutuações aleatórias e repentinas no fluido em escoamento, caracterizando um regime caótico (ou *turbulento*). Por outro lado, quanto *menor* for Re , mais preponderantes serão as forças viscosas, o que resultará num escoamento mais ordenado (ou *laminar*).

O valor do número de Reynolds que caracteriza a passagem do regime laminar para o regime turbulento recebe o nome de *número de Reynolds crítico*, Re_{cr} . De acordo com os autores Çengel e Cimbala (2012, p. 280) e White (2010, p. 358), o valor atualmente aceito do número de Reynolds crítico é $Re_{cr} = 2300$ para o escoamento interno em um tubo circular⁴.

2.13 Conservação de massa

De acordo com Çengel e Cimbala (2012, p. 151):

O princípio de conservação da massa para um volume de controle pode ser expresso como: a transferência total de massa para dentro ou para fora de um volume de controle durante um intervalo de tempo Δt é igual à variação total (aumento ou diminuição) da massa total dentro do volume de controle durante Δt .

A afirmação acima pode ser representada em forma de taxa pela Equação 2.13:

$$\dot{m}_e - \dot{m}_s = \frac{dm_{VC}}{dt} \quad (2.13)$$

onde m_e e m_s são as vazões mássicas totais (em kg/s) entrando e saindo do volume de controle, respectivamente, m_{VC} é a massa (em kg) no interior do volume de controle, t é o tempo (em s), e dm_{VC}/dt é a taxa de variação da massa no volume de controle (em kg/s).

No presente trabalho, a massa no interior de um volume de controle será constante, isto é, será *conservada*. Se isto for levado em conta na Equação 2.13, obtém-se então a relação de conservação de massa dada pela Equação 2.14:

$$\dot{m}_e - \dot{m}_s = \frac{dm_{VC}}{dt} = 0 \quad (2.14)$$

⁴ Não existe um valor exato de Re_{cr} . Na prática, esse valor é aproximado, e determina a passagem do escoamento laminar para um *escoamento de transição* em que o escoamento muda de maneira aleatória entre laminar e turbulento. Segundo Çengel e Cimbala (2012, p. 280), numa faixa de Re entre 2300 e 4000, o escoamento é de transição, e somente depois de $Re = 4000$ é que ele se torna plenamente turbulento.

Na Equação 2.14 aparece o termo m_{VC} , que corresponde à massa total no interior do volume de controle, sendo esta dada pela Equação 2.15:

$$m_{VC} = \int_{VC} \rho dV \quad (2.15)$$

Derivando-se a Equação 2.15 em relação ao tempo, obtém-se a Equação 2.16:

$$\frac{dm_{VC}}{dt} = \frac{d}{dt} \int_{VC} \rho dV \quad (2.16)$$

onde dm_{VC}/dt é a taxa de variação da massa no volume de controle (em kg/s), ρ é a massa específica do fluido (em kg/m^3) e V é o volume (em m^3).

Considerando agora a totalidade da superfície que envolve o volume de controle, tem-se que a vazão total de massa através dessa superfície é dada pela Equação 2.17:

$$\dot{m}_{tot} = \int_{SC} \rho V_n dA = \int_{SC} \rho (\vec{V} \cdot \vec{n}) dA \quad (2.17)$$

onde \dot{m}_{tot} é a vazão total de massa através da superfície de controle (em kg/s), ρ é a massa específica do fluido (em kg/m^3), V_n é a componente da velocidade (em m/s) do escoamento normal à superfície de controle de área A (em m^2), \vec{V} é a velocidade do fluido em escoamento (em m/s) e \vec{n} é o vetor normal ao elemento de área dA (em m^2) num dado ponto da superfície de controle.

Reorganizando a Equação 2.14, se obtém a Equação 2.18:

$$\frac{dm_{VC}}{dt} - (\dot{m}_e - \dot{m}_s) = 0 \quad (2.18)$$

A fim de se obter uma expressão da conservação geral de massa, substitui-se na Equação 2.18 o termo dm_{VC}/dt pelo seu equivalente dado na Equação 2.16 e considera-se que $-(\dot{m}_e - \dot{m}_s)$ é equivalente à vazão total de massa dada pela Equação 2.17. Com isso, surge a expressão da conservação geral de massa como dada por Çengel e Cimbala (2012, p. 152) na Equação 2.19:

$$\frac{d}{dt} \int_{VC} \rho dV + \int_{SC} \rho (\vec{V} \cdot \vec{n}) dA = 0 \quad (2.19)$$

na qual a primeira parcela refere-se à taxa de variação com o tempo da massa no interior do volume de controle e a segunda parcela refere-se ao fluxo total de massa através da superfície de controle.

A Equação 2.19 é na verdade um caso específico de uma expressão mais genérica conhecida como *Teorema de Transporte de Reynolds* (TTR) (REYNOLDS, 1903), que tem como expressão matemática a Equação 2.20:

$$\frac{dB_{sist}}{dt} = \frac{d}{dt} \int_{VC} b\rho dV + \int_{SC} b\rho(\vec{V} \cdot \vec{n}) dA \quad (2.20)$$

onde dB_{sist}/dt é a taxa de transferência de uma propriedade *extensiva* B (ex.: energia, massa, etc.) para o interior de um sistema e $b = B/m$ é a propriedade *intensiva* correspondente. A primeira parcela do lado direito da Equação 2.20 refere-se à taxa de variação com o tempo da propriedade B no interior do volume de controle, e a segunda parcela refere-se ao fluxo total dessa mesma propriedade através da superfície de controle. No caso específico da Equação 2.19, esta mesma pode ser obtida através do TTR simplesmente substituindo-se B por m e b por 1 (pois $b = B/m = m/m = 1$), e considerando-se $dB_{sist}/dt = dm_{sist}/dt = 0$, já que se trata da expressão de conservação de massa.

2.14 Vazões em massa e em volume

Os conceitos de vazão em massa e em volume são centrais em mecânica dos fluidos. Porém, antes de tratar desses conceitos especificamente, uma breve discussão sobre a velocidade média de escoamento é necessária, introduzindo uma simplificação no tratamento de fluidos escoando no interior de dutos.

Velocidade média de escoamento

Um fluido escoando internamente em um duto apresentará um *perfil de velocidades* do mesmo tipo que aquele apresentado na Figura 6. Os diversos vetores desenhados correspondem às velocidades de diferentes partículas do fluido num determinado ponto da coordenada transversal ao escoamento. Segundo Fox, Pritchard e McDonald (2011, p. 346), o perfil de velocidades de um fluido escoando no interior de um duto reto é dado pela Equação 2.21:

$$u(r) = -\frac{R^2}{4\mu} \left(\frac{\partial p}{\partial x} \right) \left[1 - \left(\frac{r}{R} \right)^2 \right] \quad (2.21)$$

onde u é a velocidade do fluido em escoamento (em m/s), R é o raio interno do duto (em m), μ é a viscosidade dinâmica do fluido (em $Pa.s$), $\partial p/\partial x$ é o gradiente de pressão ao longo do comprimento do duto (em Pa/m), e r é a coordenada radial do duto partindo da linha central (em m).

De acordo com Çengel e Cimbala (2012, p. 284), a Equação 2.21 pode ser usada a fim de se obter uma equação mais simples para o perfil de velocidades, resultando na Equação 2.22:

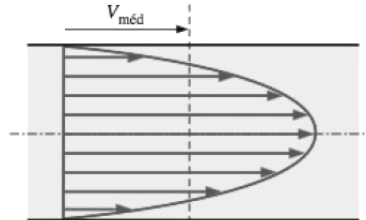
$$u(r) = 2V_{\text{méd}} \left[1 - \left(\frac{r}{R} \right)^2 \right] \quad (2.22)$$

A velocidade média desse escoamento é calculada como sendo a soma de todas as velocidades do perfil de velocidades dividida pela área da seção transversal. Empregando um cálculo semelhante ao usado por Stewart (2010, p. 418) no *Teorema do Valor Médio para Integrais*, a velocidade média de escoamento pode ser calculada pela Equação 2.23:

$$V_{\text{méd}} = \frac{1}{A_c} \int_{A_c} V_n dA \quad (2.23)$$

onde V_n é a componente da velocidade (em m/s) do escoamento normal à superfície de controle de área A_c (em m^2), sendo esta superfície orientada perpendicularmente ao sentido do escoamento.

Figura 6 – A velocidade média $V_{\text{méd}}$ é obtida por integração da velocidade ao longo da seção transversal ao escoamento e posterior divisão da integração pela área da seção.



Fonte: Çengel e Cimbala (2012, p. 151).

Vazão em massa

A relação de vazão em massa pode ser obtida pela manipulação da expressão de conservação geral de massa, Equação 2.19, mostrada novamente:

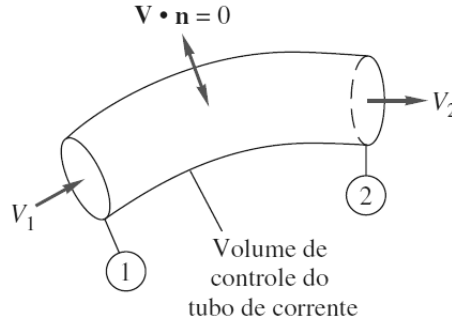
$$\frac{d}{dt} \int_{VC} \rho dV + \int_{SC} \rho(\vec{V} \cdot \vec{n}) dA = 0 \quad (2.24)$$

Considerando um volume de controle hipotético através do qual um fluido escoar apenas através de duas de suas seções 1 e 2 (v. Figura 7), e considerando o caso em que a

quantidade de massa permanece constante no interior do volume de controle (Equação 2.14), obtém-se da expressão de conservação geral de massa na Equação 2.25:

$$\int_{SC} \rho(\vec{V} \cdot \vec{n}) dA = 0 \quad (2.25)$$

Figura 7 – Volume de controle no qual o fluido escoar apenas através das seções 1 e 2.



Fonte: White (2010, p. 163).

Tendo em mente que o fluido escoar apenas através das seções 1 e 2, a Equação 2.25 pode ser convenientemente manipulada para⁵ a Equação 2.26:

$$-\int_{A_1} \rho V_1 dA_1 + \int_{A_2} \rho V_2 dA_2 = 0 \quad (2.26)$$

Integrando a Equação 2.26 e levando em conta a definição de velocidade média obtida na Equação 2.23, obtém-se a Equação 2.27:

$$-\rho V_{1,méd} A_1 + \rho V_{2,méd} A_2 = 0 \quad (2.27)$$

que equivale a dizer que a quantidade de massa que escoar através da área da seção 1 é igual àquela que escoar através da área da seção 2. Portanto, essa quantidade de massa em escoamento através de quaisquer seções numa tubulação é *constante*, e a essa constante pode-se atribuir o nome de *vazão em massa*, cuja expressão geral é dada pela Equação 2.28:

$$\dot{m} = \rho V_{méd} A \quad (2.28)$$

onde \dot{m} é a vazão em massa (em kg/s) de um fluido de massa específica ρ (em kg/m^3) escoando através de uma seção transversal de área A (em m^2) a uma velocidade média $V_{méd}$ (em m/s).

⁵ Na Equação 2.26, o sinal de negativo aparece para a primeira integral pois na seção 1 o vetor velocidade e o vetor normal à área possuem direções opostas. Nesse caso, o produto escalar $\vec{V} \cdot \vec{n}$ da Equação 2.25 é negativo.

Substituindo a Equação 2.28 na Equação 2.27, se obtém a expressão de conservação de massa dada pela Equação 2.29:

$$-\dot{m}_1 + \dot{m}_2 = 0 \quad (2.29)$$

que pode ser manipulada para se obter a Equação 2.30 dada por Çengel e Cimbala (2012, p. 154):

$$\dot{m}_1 = \dot{m}_2 \quad (2.30)$$

em que \dot{m} corresponde à vazão em massa (em kg/s) tanto na seção 1 quanto na 2.

Vazão em volume

A relação de vazão em volume de um fluido em escoamento pode ser obtida simplesmente dividindo a Equação 2.28 pela massa específica ρ do fluido, obtendo-se a Equação 2.31:

$$\frac{\dot{m}}{\rho} = \frac{\rho V_{méd} A}{\rho} \quad (2.31)$$

resultando na Equação 2.32:

$$\dot{V} = V_{méd} A \quad (2.32)$$

onde \dot{V} é a vazão em volume (em m^3/s), A é a área da seção transversal ao escoamento (em m^2), e $V_{méd}$ é a velocidade média de escoamento no interior do duto (em m/s).

A vazão em massa e a vazão em volume estão relacionadas pela Equação 2.33 dada por Çengel e Cimbala (2012, p. 151):

$$\dot{m} = \rho \dot{V} \quad (2.33)$$

onde \dot{m} é vazão em massa (em kg/s), ρ é a massa específica do fluido (em kg/m^3) e \dot{V} é a vazão em volume (em m^3/s).

Obtenção da velocidade média a partir da vazão

A velocidade média $V_{méd}$ é um parâmetro de grande importância em qualquer análise de mecânica dos fluidos, e por este motivo é conveniente que o seu valor seja obtido de uma maneira mais prática do que aquela demonstrada pela Equação 2.23. Ao se observar as equações Equação 2.28 e Equação 2.32, percebe-se que ambas relacionam

alguma vazão (mássica ou volumétrica) à velocidade média $V_{méd}$. Desta forma, num dado escoamento, basta conhecer a vazão para se obter a velocidade média como o faz White (2010, p. 106) na Equação 2.34:

$$V_{méd} = \frac{\dot{m}}{\rho A} = \frac{\dot{V}}{A} \quad (2.34)$$

2.15 Conservação de energia

De acordo com a Primeira Lei da Termodinâmica discutida na seção 2.3, a energia não pode ser criada nem destruída durante um processo, mas pode apenas mudar de forma. Ainda de acordo com a primeira lei, a quantidade de energia em um sistema pode apenas ser alterada por transferência de calor, Q , e por realização de trabalho, W . Assim, para um dado sistema, tem-se a Equação 2.35 dada por Çengel e Cimbala (2012, p. 176):

$$\frac{dE_{sis}}{dt} = \frac{dQ_{tot}}{dt} + \frac{dW_{tot}}{dt} \quad (2.35)$$

onde dE_{sis}/dt é a taxa de variação da quantidade de energia do sistema (em J/s), dQ_{tot}/dt é a taxa de transferência total de energia em forma calor para o sistema (em J/s), e dW_{tot}/dt é a taxa de energia transferida em forma de trabalho para o sistema (em J/s).

Çengel e Cimbala (2012, p. 176) reescrevem a Equação 2.35 como a Equação 2.36:

$$\frac{d}{dt} \int_{sis} \rho e dV = \frac{dQ_{tot}}{dt} + \frac{dW_{tot}}{dt} \quad (2.36)$$

onde a integral do lado esquerdo equivale ao dE_{sis}/dt da Equação 2.35, ρ é a massa específica do fluido (em kg/m^3), e e é a energia total específica (em J/kg). Segundo Çengel e Cimbala (2012, p. 176), essa energia específica e é expressa pela Equação 2.37:

$$e = u + e_{cin} + e_{pot} = u + \frac{V^2}{2} + gz \quad (2.37)$$

onde u é a energia interna específica (em J/kg), $e_{cin} = V^2/2$ é a energia cinética específica (em J/kg), $e_{pot} = gz$ é a energia potencial gravitacional específica (em J/kg), V é a velocidade (em m/s), g é a constante gravitacional (em m/s^2) e z é a altura em relação a um plano de referência (em m).

Transferência de energia por trabalho

No presente texto, serão consideradas duas formas de transferência de energia por trabalho: o *trabalho de eixo* W_{eixo} , que é aquele transmitido por um eixo giratório, e o *trabalho de pressão* $W_{pressão}$, que é a transferência de energia realizada pelas forças de

pressão. A soma desses dois tipos de trabalho constitui aquilo que se chama de *trabalho total* que é expresso pela Equação 2.38:

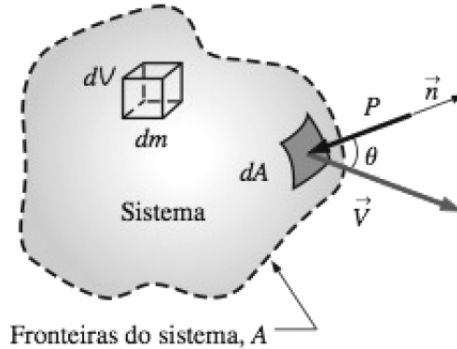
$$W_{tot} = W_{eixo} + W_{pressão} \quad (2.38)$$

onde W_{tot} é a quantidade total de energia em forma de trabalho transferida para o sistema (em J), W_{eixo} é a quantidade de energia transferida em forma de trabalho para o sistema devido ao eixo giratório (em J), e $W_{pressão}$ é a quantidade de energia transferida em forma de trabalho para o sistema devido às forças de pressão (em J).

Trabalho realizado por forças de pressão

Para a obtenção de uma relação para o trabalho realizado por forças de pressão sobre um sistema, $W_{pressão}$, será considerado um sistema de forma arbitrária como ilustrado na Figura 8.

Figura 8 – Sistema de forma arbitrária com área A e com uma pressão P incidindo sobre ele.



Fonte: Çengel e Cimbala (2012, p. 177).

O trabalho realizado pelas forças de pressão, $W_{pressão}$, sobre um sistema de formato genérico é dado pela Equação 2.39 como:

$$W_{pressão} = - \int_A P(\vec{s} \cdot \vec{n}) dA \quad (2.39)$$

onde $W_{pressão}$ é o trabalho realizado pelas forças de pressão (em J), A é a área da superfície de controle que envolve o sistema (em m^2), P é o módulo da pressão sendo exercida sobre o sistema (em Pa), \vec{s} é o vetor deslocamento das fronteiras do sistema (em m), e \vec{n} é o vetor normal relativo ao elemento de área dA . O sinal de negativo da Equação 2.39 e o produto escalar entre \vec{s} e \vec{n} faz com que a compressão do sistema resulte em trabalho realizado *sobre ele* (o sistema ganha energia), e a expansão resulte em trabalho realizado *pelo* sistema (o sistema perde energia).

Na Figura 8, ao invés de um deslocamento \vec{s} , está sendo mostrada uma velocidade \vec{V} . Essa velocidade corresponde a $\vec{V} = d\vec{s}/dt$, e quando a mesma é considerada no lugar de \vec{s} na Equação 2.39, obtém-se a Equação 2.40 dada por Çengel e Cimbala (2012, p. 178):

$$\frac{dW_{\text{pressão}}}{dt} = - \int_A P(\vec{V} \cdot \vec{n}) dA \quad (2.40)$$

em que $dW_{\text{pressão}}/dt$ é a taxa de transferência de trabalho para o sistema (ou *potência*) pelas forças de pressão (em W^6 ou J/s), e \vec{V} é a velocidade referente ao deslocamento da superfície de controle (em m/s). Os demais termos já foram explicitados na Equação 2.39.

Çengel e Cimbala (2012, p. 178) modificam a Equação 2.40 para uma versão em que a massa específica ρ (em kg/m^3) aparece no interior da integral da Equação 2.41:

$$\frac{dW_{\text{pressão}}}{dt} = - \int_A \frac{P}{\rho} \rho(\vec{V} \cdot \vec{n}) dA \quad (2.41)$$

Equação da energia para um volume de controle

Considerando novamente a Equação 2.35 dada por Çengel e Cimbala (2012, p. 176):

$$\frac{dE_{\text{sis}}}{dt} = \frac{dQ_{\text{tot}}}{dt} + \frac{dW_{\text{tot}}}{dt}$$

e considerando-se que o trabalho total, W_{tot} , é a soma dos trabalhos de eixo e de pressão (Equação 2.38), tem-se a Equação 2.42:

$$\frac{dE_{\text{sis}}}{dt} = \frac{dQ_{\text{tot}}}{dt} + \frac{dW_{\text{eixo}}}{dt} + \frac{dW_{\text{pressão}}}{dt} \quad (2.42)$$

onde dE_{sis}/dt é a taxa de variação da quantidade de energia do sistema (em J/s), dQ_{tot}/dt é a taxa de transferência total de energia em forma calor para o sistema (em J/s), dW_{eixo}/dt é a taxa de transferência de energia em forma de trabalho para o sistema devido ao eixo giratório (em J/s), e $dW_{\text{pressão}}/dt$ é a taxa de transferência de energia em forma de trabalho para o sistema devido às forças de pressão (em J/s).

A parte esquerda da Equação 2.42, dE_{sis}/dt , pode ser aplicada juntamente com o Teorema de Transporte de Reynolds (Equação 2.20) para a obtenção de uma relação da conservação de energia em um volume de controle. Dessa forma, substituindo B_{sis} por E_{sis} e b por e no *TTR*, obtém-se a Equação 2.43:

$$\frac{dE_{\text{sis}}}{dt} = \frac{d}{dt} \int_{VC} e \rho dV + \int_{SC} e \rho(\vec{V} \cdot \vec{n}) dA \quad (2.43)$$

⁶ *Watts*, unidade de potência equivalente a J/s .

onde dE_{sis}/dt é a taxa de transferência de energia para o interior do sistema (em J/s), com a primeira integral do lado direito referindo-se à taxa de variação do conteúdo de energia no interior do volume de controle (em J/s), e com a segunda integral referindo-se à taxa de transferência de energia para fora do volume de controle (em J/s).

Substituindo dE_{sis}/dt da Equação 2.43 pelo seu equivalente da Equação 2.42, tem-se a Equação 2.44:

$$\frac{dQ_{tot}}{dt} + \frac{dW_{eixo}}{dt} + \frac{dW_{pressão}}{dt} = \frac{d}{dt} \int_{VC} e \rho dV + \int_{SC} e \rho (\vec{V} \cdot \vec{n}) dA \quad (2.44)$$

A terceira parcela do lado esquerdo da Equação 2.44, $dW_{pressão}/dt$, é dada pela Equação 2.41. Fazendo a devida substituição e rearranjado⁷, obtém-se a Equação 2.45 dada por Çengel e Cimbala (2012, p. 178):

$$\frac{dQ_{tot}}{dt} + \frac{dW_{eixo}}{dt} = \frac{d}{dt} \int_{VC} e \rho dV + \int_{SC} \left(\frac{P}{\rho} + e \right) \rho (\vec{V} \cdot \vec{n}) dA \quad (2.45)$$

Uma simplificação adicional ainda pode ser realizada na Equação 2.45. Como o fluido só atravessa as partes da superfície de controle que não coincidem com superfícies sólidas (paredes do duto), tem-se então que a energia só será transferida através de seções da superfície de controle através das quais o fluido pode entrar e sair (Figura 7). Considerando-se também a vazão em massa, \dot{m} , que é dada pela Equação 2.46:

$$\dot{m} = \int_A \rho (\vec{V} \cdot \vec{n}) dA \quad (2.46)$$

conclui-se que a Equação 2.45 pode ser reescrita como a Equação 2.47:

$$\frac{dQ_{tot}}{dt} + \frac{dW_{eixo}}{dt} = \frac{d}{dt} \int_{VC} e \rho dV + \sum_s \dot{m} \left(\frac{P}{\rho} + e \right) - \sum_e \dot{m} \left(\frac{P}{\rho} + e \right) \quad (2.47)$$

na qual as duas últimas parcelas do lado direito (isto é, os somatórios) referem-se aos fluxos de energia tanto na seção de saída quanto na de entrada do volume de controle, respectivamente.

Num sistema em que só existe uma entrada e uma saída⁸, e em que o regime de escoamento seja do tipo *permanente*⁹, a Equação 2.47 é reescrita como a Equação 2.48:

$$\frac{dQ_{tot}}{dt} + \frac{dW_{eixo}}{dt} = \dot{m} \left(\frac{P_2 - P_1}{\rho} + e_2 - e_1 \right) \quad (2.48)$$

⁷ O rearranjo citado se dá passando $dW_{pressão}/dt$ para o lado direito da equação, e colocando a sua expressão dentro da integral de superfície.

⁸ Como em dispositivos de corrente simples.

⁹ Quando o regime de escoamento é permanente, a primeira parcela do lado direito da Equação 2.47 é zero, pois não há variação com o tempo do conteúdo interno de energia no volume de controle.

onde os subscritos 1 e 2 (como em P_1 e P_2) servem para indicar que os parâmetros referem-se à entrada e à saída do sistema, respectivamente.

Como e é a energia total específica dada pela Equação 2.37:

$$e = u + \frac{V^2}{2} + gz$$

fazendo-se a devida substituição na Equação 2.48 e rearranjando, se obtém a Equação 2.49:

$$\frac{dQ_{tot}}{dt} + \frac{dW_{eixo}}{dt} = \dot{m} \left[\frac{P_2 - P_1}{\rho} + u_2 - u_1 + \frac{V_2^2 - V_1^2}{2} + g(z_2 - z_1) \right] \quad (2.49)$$

Dividindo a Equação 2.49 pela vazão mássica, \dot{m} , a mesma se transforma na Equação 2.50:

$$q_{tot} + w_{eixo} = \frac{P_2 - P_1}{\rho} + u_2 - u_1 + \frac{V_2^2 - V_1^2}{2} + g(z_2 - z_1) \quad (2.50)$$

Rearranjando a Equação 2.50, a mesma pode ser escrita como a Equação 2.51:

$$w_{eixo} + \frac{P_1}{\rho} + \frac{V_1^2}{2} + gz_1 = \frac{P_2}{\rho} + \frac{V_2^2}{2} + gz_2 + (u_2 - u_1 - q_{tot}) \quad (2.51)$$

onde w_{eixo} é a energia em forma de trabalho de eixo transferida para o sistema (em J/kg), P é a pressão (em Pa), ρ é a massa específica (em kg/m^3), V é a velocidade (em m/s), g é a constante gravitacional (em m/s^2), z é a altura em relação a um referencial arbitrário (em m), u é a energia interna específica (em J/kg), e q_{tot} é a quantidade de energia específica em forma de calor transferida para o sistema (em J/kg).

A Equação 2.51 tem como a sua última parcela a expressão $u_2 - u_1 - q_{tot}$ podendo a mesma ser substituída por $e_{mec,perda}$, resultando na Equação 2.52:

$$e_{mec,perda} = u_2 - u_1 - q_{tot} \quad (2.52)$$

onde $e_{mec,perda}$ é a energia mecânica específica dissipada (em J/kg), u é a energia interna específica (em J/kg), e q_{tot} é a quantidade de energia específica em forma de calor transferida para o sistema (em J/kg). A Equação 2.52 foi definida desta forma pois qualquer aumento de $u_2 - u_1$ acima de q_{tot} significa a conversão irreversível de energia mecânica em energia térmica devido aos efeitos de atrito¹⁰.

¹⁰ $u_2 - u_1 > q_{tot}$ significa que a energia interna do sistema entre os pontos 1 e 2 aumentou além do que aumentaria se apenas energia em forma de calor, q_{tot} , tivesse sido transferida para o sistema. Disto deduz-se que houve aumento da energia interna do sistema devido ao atrito mecânico.

Finalmente, a Equação 2.51 pode ser reescrita, obtendo-se assim a *equação da energia* como dada na Equação 2.53 por Çengel e Cimbala (2012, p. 180):

$$w_{eixo} + \frac{P_1}{\rho} + \frac{V_1^2}{2} + gz_1 = \frac{P_2}{\rho} + \frac{V_2^2}{2} + gz_2 + e_{mec,perda} \quad (2.53)$$

onde w_{eixo} é a energia específica em forma de trabalho de eixo transferida para o sistema (em J/kg), P é a pressão (em Pa), ρ é a massa específica (em kg/m^3), V é a velocidade (em m/s), g é a constante gravitacional (em m/s^2), z é a altura em relação a um referencial arbitrário (em m), e $e_{mec,perda}$ é a dissipação (ou *perda*) da energia mecânica específica (em J/kg).

Equação da energia em termos de cargas

De acordo com Henn (2006, p. 40), para os casos de máquinas que trabalham com fluido incompressível, é comum associar a energia recebida (caso das bombas) pelo fluido, ao passar pela máquina, a uma altura de coluna de fluido, H (*head* ou *carga manométrica*). Quando se divide a energia específica do fluido pela constante gravitacional, obtém-se o valor do *head* em metros, como em Equação 2.54:

$$H = Y/g \quad (2.54)$$

onde H é o valor de *head* (em m), Y é a energia específica (em J/kg) e g é a constante gravitacional (em m/s^2).

Usando-se, então, a mesma ideia expressa no parágrafo anterior e na Equação 2.54, a equação da energia, Equação 2.53, pode ser reescrita em termos de *cargas manométricas* ao invés de *energias específicas*. Quando se divide cada um dos termos da Equação 2.53 por g (constante gravitacional), obtém-se a Equação 2.55:

$$h_{eixo} + \frac{P_1}{\rho g} + \frac{V_1^2}{2g} + z_1 = \frac{P_2}{\rho g} + \frac{V_2^2}{2g} + z_2 + h_{mec,perda} \quad (2.55)$$

em que cada uma das parcelas está em *metros* (m). Os termos h_{eixo} e $h_{mec,perda}$ são, respectivamente, o trabalho de eixo e a energia dissipada (perda de carga) (em m).

Na Equação 2.55 a parcela h_{eixo} refere-se à carga transferida para o sistema (energia por unidade de peso de fluido) através de um eixo giratório. Se esse eixo for uma bomba, carga será transferida *para* o sistema, aumentando a sua energia; se, porém, for uma turbina, carga será transferida *do* sistema, diminuindo a sua energia. Dessa forma, a Equação 2.56 é escrita como:

$$h_{eixo} = h_{bomba} - h_{turbina} \quad (2.56)$$

onde h_{eixo} é o trabalho de eixo *total* (em m) transferido entre as máquinas e o fluido.

Substituindo h_{eixo} da Equação 2.55 pelo seu equivalente da Equação 2.56 e rearranjando as parcelas de forma conveniente, obtém-se a Equação 2.57 dada por Çengel e Cimbala (2012, p. 181):

$$\frac{P_1}{\rho g} + \frac{V_1^2}{2g} + z_1 + h_{bomba} = \frac{P_2}{\rho g} + \frac{V_2^2}{2g} + z_2 + h_{turbina} + h_{mec,perda} \quad (2.57)$$

onde P é a pressão (em Pa), ρ é a massa específica (em kg/m^3), g é a constante gravitacional (em m/s^2), V é a velocidade do fluido em escoamento (em m/s), z é a altura em relação a um referencial (em m), h_{bomba} é a carga fornecida ao sistema por uma bomba (em m), $h_{turbina}$ é a carga transferida do sistema a uma turbina (em m), e $h_{mec,perda}$ é a carga dissipada pelos efeitos de atrito (em m).

2.16 A equação de Bernoulli

Segundo Çengel e Cimbala (2012, p. 161), “a equação de Bernoulli é uma relação aproximada entre pressão, velocidade e elevação e é válida em regiões de escoamento incompressível e em regime permanente, onde as forças de atrito resultantes são desprezíveis.”

Sendo a equação de Bernoulli uma expressão do princípio da conservação de energia, ela afirma que a energia total do escoamento numa linha de corrente é *constante*, o que é expresso pela Equação 2.58:

$$\frac{P}{\rho} + \frac{V^2}{2} + gz = const. \quad (2.58)$$

onde P é a pressão (em Pa), ρ é a massa específica do fluido (em kg/m^3), V é a velocidade do escoamento (em m/s), g é a constante gravitacional (em m/s^2), e z é a altura relativa a um referencial (em m).

A equação de Bernoulli pode ser obtida através da Equação 2.53, porém, considerando que tanto o trabalho de eixo w_{eixo} quanto as perdas mecânicas $e_{mec,perda}$ são iguais a zero.

Pressões estática, dinâmica e hidrodinâmica

Uma das utilidades da equação de Bernoulli é a possibilidade de se analisar como as diferenças pressões em um fluido em escoamento se relacionam. De acordo com Çengel e Cimbala (2012, p. 164), a multiplicação de toda a equação de Bernoulli (Equação 2.58)

pela massa específica ρ faz com que todos os termos tenham unidade de pressão, o que resulta na Equação 2.59:

$$P + \frac{\rho V^2}{2} + \rho g z = \text{const.} \quad (2.59)$$

Cada uma das parcelas da Equação 2.59 é listada a seguir:

- P : pressão estática (em Pa);
- $\rho V^2/2$: pressão dinâmica (em Pa);
- $\rho g z$: pressão hidrostática (em Pa).

A *pressão estática* P representa a pressão termodinâmica real do fluido e não incorpora nenhum efeito dinâmico, e a *pressão dinâmica* $\rho V^2/2$ refere-se ao aumento da pressão quando o fluido em movimento é parado de forma isoentrópica. Já a *pressão hidrostática* representa os efeitos da altura na pressão. Por exemplo: se um elemento de fluido estiver num ponto 1 e migrar para um ponto 2 mais abaixo, a sua pressão estática aumentará numa mesma quantidade em módulo da diminuição da pressão hidrostática (se a pressão dinâmica se mantiver constante). Todas essas pressões são medidas em Pa .

2.17 Perda de pressão e perda de carga

Conforme o escoamento avança ao longo da tubulação, o mesmo perde pressão (ou *carga*) devido ao atrito com as paredes da própria tubulação e com outros obstáculos. Nas palavras de Brunetti (2008, p. 104), “a perda de carga deverá ser interpretada pelo aumento da energia térmica ou por uma perda de energia de pressão, reduzindo-se, portanto, o conteúdo de energia mecânica do fluido.”

Existem dois tipos principais de perdas de carga: perdas de carga contínuas (ou *grandes perdas*), que são aquelas relacionadas ao atrito do fluido em escoamento com a superfície da parede da tubulação, e as perdas de carga locais (ou *perdas menores*) que são relacionadas ao atrito do fluido em escoamento com obstáculos como válvulas, curvas, conexões, etc. Esses dois tipos de perdas serão discutidos com mais detalhes a seguir.

Perda de carga contínua

As perdas de carga contínuas (ou *grandes perdas*) surgem devido à tensão de cisalhamento entre as camadas do fluido em escoamento e também entre o fluido e as paredes da tubulação. De acordo com Brunetti (2008, p. 168), esta perda “é a que acontece ao longo de tubos retos, de seção constante, devido ao atrito das próprias partículas do fluido entre si.” Assim, o atrito causado pelos efeitos viscosos faz com que o fluido em escoamento perca pressão (ou *carga*).

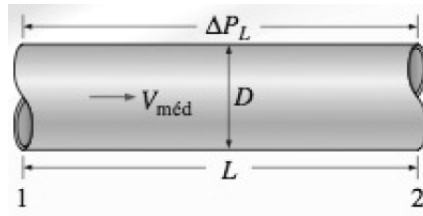
Como expressão geral para a perda de pressão, tem-se a Equação 2.60:

$$\Delta P = P_1 - P_2 = f \frac{L}{D} \frac{\rho V_{méd}^2}{2} \quad (2.60)$$

onde ΔP é a diferença de pressão (em Pa) entre os pontos 1 e 2 separados por uma distância L (em m), f é o fator de atrito de Darcy-Weisbach (adim.), D é o diâmetro da tubulação (em m), ρ é a massa específica do fluido (em kg/m^3), e $V_{méd}$ é a velocidade média do escoamento (em m/s).

A Figura 9 mostra esquematicamente uma tubulação de diâmetro D em que o fluido apresenta uma perda de pressão ΔP entre os pontos 1 e 2.

Figura 9 – A perda de pressão ΔP_L do ponto 1 para o ponto 2 de um fluido em escoamento numa tubulação é causada pelos efeitos viscosos.



Fonte: Adaptado de Çengel e Cimbala (2012, p. 284).

A perda de carga contínua pode ser facilmente obtida a partir da Equação 2.60, bastando dividir a mesma pelo produto ρg ¹¹ obtendo-se a Equação 2.61:

$$h_c = \frac{\Delta P}{\rho g} = f \frac{L}{D} \frac{V_{méd}^2}{2g} \quad (2.61)$$

onde h_c é a perda de carga contínua (em m), ΔP é a diferença de pressão (em Pa), ρ é a massa específica do fluido (em kg/m^3), g é a constante gravitacional (em m/s^2), f é o fator de atrito de Darcy-Weisbach (adim.), L é o comprimento da tubulação entre os pontos 1 e 2 (em m), D é o diâmetro da tubulação (em m), e $V_{méd}$ é a velocidade média do escoamento (em m/s).

Tanto a Equação 2.60 quanto a Equação 2.61 são aplicáveis para os escoamentos de tipos laminar e turbulento.

Perda de carga local

As perdas de carga locais (ou *perdas menores*), de acordo com Brunetti (2008, p. 168), “acontecem em locais das instalações em que o fluido sofre perturbações bruscas no seu escoamento.” As mesmas são causadas quando o fluido em escoamento passa

¹¹ O produto ρg é também conhecido como *peso específico*, γ .

por válvulas, conexões, curvas, cotovelos, tês, entradas, saídas, reduções e extensões. A passagem do fluido em escoamento por esses componentes causa perdas adicionais por causa da separação do escoamento e da mistura que os mesmos causam no fluido.

Segundo Çengel e Cimbala (2012, p. 302), a perda de carga causada por um componente é dada pela Equação 2.62:

$$h_l = K \frac{V^2}{2g} \quad (2.62)$$

onde h_l é a perda de carga local (em m), K é o coeficiente de perda (ou coeficiente de resistência) característico do componente (adimensional), V é a velocidade do fluido em escoamento (em m/s), e g é a constante gravitacional (em m/s^2).

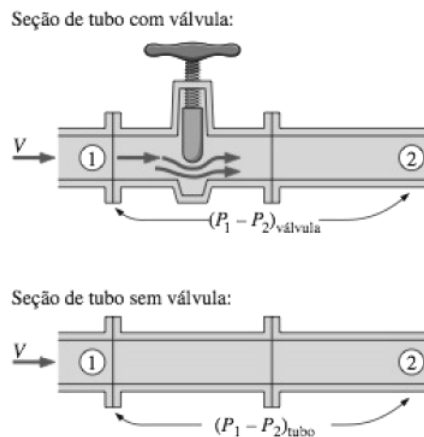
O coeficiente de perda K da Equação 2.62 é obtido empiricamente através da Equação 2.63:

$$K = \frac{\Delta P}{\frac{1}{2}\rho V^2} \quad (2.63)$$

onde ρ é a massa específica do fluido (em kg/m^3), V é a velocidade do escoamento (em m/s), e ΔP é a diferença entre a queda de pressão entre os pontos 1 e 2 *com* o componente e a queda de pressão entre 1 e 2 *sem* o componente (v. Figura 10 para um exemplo com válvula). Ambos os ΔP estão em Pa , com o cálculo da queda de pressão ΔP entre as duas situações sendo dado pela Equação 2.64:

$$\Delta P = (P_1 - P_2)_{com} - (P_1 - P_2)_{sem} \quad (2.64)$$

Figura 10 – Quando um componente está presente num escoamento (como uma válvula) a queda de pressão entre 1 e 2 é maior do que na ausência deste.



Fonte: Adaptado de Çengel e Cimbala (2012, p. 301).

Diferentes componentes causam diferentes perdas de carga, o que se traduz em diferentes coeficientes de perda K . Esses coeficientes são obtidos empiricamente através do correto uso da Equação 2.63 e disponibilizados em tabelas nos livros e manuais de engenharia.

Perda de carga total

A perda de carga total, $h_{mec,perda}$, é simplesmente a soma das perdas de carga local (h_l) e contínua (h_c), de modo que se tem a Equação 2.65:

$$h_{mec,perda} = h_c + h_l \quad (2.65)$$

em que todos os termos são expressos em *metros* (m).

2.18 Rugosidade

A rugosidade é um parâmetro importante na análise de escoamento de fluidos no interior de dutos pois a mesma possui influência direta e importante no cálculo da perda de carga e, portanto, na modelagem do sistema hidráulico. Nas palavras de Brunetti (2008, p. 168), “os condutos apresentam asperezas nas paredes internas que influem na perda de carga dos fluidos em escoamento. Em geral, tais asperezas não são uniformes, mas apresentam uma distribuição aleatória tanto em altura quanto em disposição.”

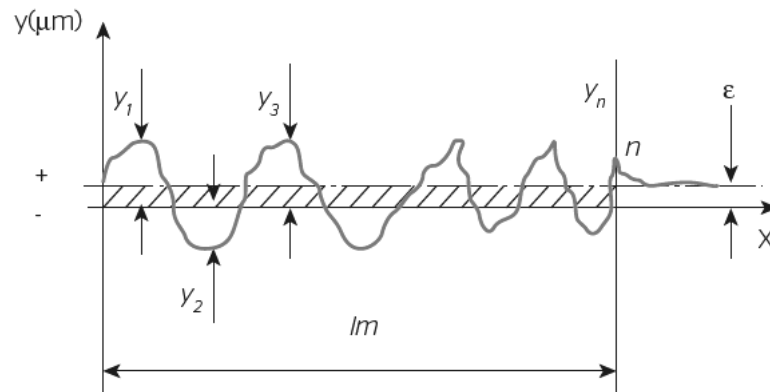
Há basicamente duas classificações de rugosidade: absoluta e relativa. Sobre a *rugosidade absoluta*, Macintyre (1997, p. 643) diz que esta “é a altura média das saliências da rugosidade de uma superfície. É geralmente medida em milímetros e se representa pela letra ϵ .” A Figura 11 mostra um diagrama em que se podem ver várias alturas das n saliências denotadas por y_i ($i = 1, \dots, n$), e, por fim, um ϵ que indica o tamanho da rugosidade absoluta, calculado pela média aritmética de todas as medidas y_i , como exposto na Equação 2.66:

$$\epsilon = \frac{y_1 + y_2 + \dots + y_n}{n} \quad (2.66)$$

Sobre a *rugosidade relativa*, o mesmo autor afirma que esta “é o quociente da rugosidade absoluta pelo diâmetro interno do encanamento” (MACINTYRE, 1997, p. 643). A rugosidade relativa é, portanto, dada por ϵ/D .

Faixas de valores de rugosidade absoluta de tubos de PVC encontrados em diversas fontes foram agregados por Kellner, Akutsu e Reis (2016) na Tabela 1.

Figura 11 – Medidas de afastamento dos pontos do perfil de rugosidade em relação à linha média do percurso de medição.



Fonte: Adaptado de Kellner, Akutsu e Reis (2016, p. 3).

Tabela 1 – Valores de rugosidade absoluta de tubos de PVC encontrados em diferentes referências.

Rugosidade absoluta ϵ (mm)	Fonte
0,0015 – 0,010	(PORTO, 2000)
0,01 – 0,06	(MARQUES; SOUSA, 2011)
0,08 – 0,12	(ABNT, 1977)
0,1	(NETTO; ALVAREZ, 1982)
0,0015	(ROSSMAN, 2000)
0,0015 – 0,12	Faixa resultante

Fonte: Kellner, Akutsu e Reis (2016, p. 2).

2.19 Fator de atrito de Darcy-Weisbach

O fator de atrito de Darcy-Weisbach f é um número adimensional que é responsável por relacionar a tensão de cisalhamento na parede do tubo τ_w (em Pa) com a velocidade média do escoamento $V_{méd}$ (em m/s) e com a massa específica ρ (em kg/m^3). De acordo com Çengel e Cimbala (2012, p. 285) tem-se a Equação 2.67:

$$f = \frac{8\tau_w}{\rho V_{méd}^2} \quad (2.67)$$

De acordo com a Equação 2.67, a tensão de cisalhamento entre o fluido e a superfície interna do tubo por onde o fluido escoar é diretamente proporcional ao fator de atrito f , à massa específica ρ e à velocidade média do escoamento $V_{méd}$. O fato de a tensão de cisalhamento τ_w ser proporcional ao fator de atrito, faz deste último um importante parâmetro para o escoamento de fluido no interior de tubulações, já que vai ajudar a

determinar, por exemplo, a potência que uma bomba deve adicionar a um escoamento para que este supere as perdas de carga.

O fator de atrito de Darcy-Weisbach apresenta valores e padrões diferentes dependendo se o escoamento for laminar ou turbulento, com ambos os tipos sendo apresentados a seguir.

Fator de atrito no escoamento laminar

Para o escoamento laminar, o fator de atrito é dependente somente do número de Reynolds de maneira inversamente proporcional a este, podendo ser expresso por uma simples relação, dada por White (2010, p. 360) como a Equação 2.68:

$$f = \frac{64}{Re} \quad (2.68)$$

Fator de atrito no escoamento turbulento

Diferentemente do caso para escoamento laminar, o fator de atrito para o caso turbulento não possui uma fórmula que tenha sido deduzida analiticamente, mas sim empiricamente. Além disto, para o caso turbulento, o fator de atrito depende não somente do número de Reynolds, mas também da *rugosidade relativa*, definida por ϵ/D , onde ϵ é a rugosidade (geralmente medida em *mm*), e D é o diâmetro da tubulação (em *m*)¹².

Munido de dados experimentais de tubos comerciais, Cyril F. Colebrook obteve em 1939 a *equação de Colebrook* (COLEBROOK, 1939), que abrange toda região de escoamento turbulento sendo dada pela Equação 2.69:

$$\frac{1}{\sqrt{f}} = -2,0 \log \left(\frac{\epsilon/D}{3,7} + \frac{2,51}{Re\sqrt{f}} \right) \quad (2.69)$$

onde f é o fator de atrito de Darcy-Weisbach (adimensional), ϵ/D é a rugosidade relativa (adimensional), e Re é o número de Reynolds (adimensional).

A equação de Colebrook é implícita em f , o que pode fazer com que ela não seja muito prática. Em 1983, S. E. Haaland obteve uma expressão explícita aproximada para f (HAALAND, 1983), e que pode ser mais facilmente usada, sendo dada pela Equação 2.70:

$$\frac{1}{\sqrt{f}} \approx -1,8 \log \left[\frac{6,9}{Re} + \left(\frac{\epsilon/D}{3,7} \right)^{1,11} \right] \quad (2.70)$$

¹² Como a rugosidade ϵ é geralmente dada em *mm* e o diâmetro em *m*, uma conversão de um dos dois (de *mm* para *m* ou vice-versa) é necessária para o cálculo da rugosidade relativa ϵ/D .

onde f é o fator de atrito de Darcy-Weisbach (adimensional), ϵ/D é a rugosidade relativa (adimensional), e Re é o número de Reynolds (adimensional). De acordo com White (2010, p. 361), a Equação 2.70 varia menos que 2% em relação à equação de Colebrook (Equação 2.69) sendo, portanto, uma boa aproximação.

Apesar de a equação de Haaland (Equação 2.70) apresentar uma boa aproximação em relação à equação original de Colebrook (Equação 2.69), Zigrang e Sylvester (1982) desenvolveram uma equação com resultados ainda mais acurados, sendo a mesma expressa pela Equação 2.71:

$$\frac{1}{\sqrt{f}} = -2 \log \left\{ \frac{\epsilon/D}{3,7} - \frac{5,02}{Re} \log \left[\frac{\epsilon/D}{3,7} - \frac{5,02}{Re} \log \left(\frac{\epsilon/D}{3,7} + \frac{13}{Re} \right) \right] \right\} \quad (2.71)$$

onde f é o fator de atrito de Darcy-Weisbach (adimensional), ϵ/D é a rugosidade relativa (adimensional), e Re é o número de Reynolds (adimensional).

Em um estudo conduzido por Genić et al. (2011), verificou-se que a Equação 2.71 de Zigrang e Sylvester apresenta um erro em relação à equação de Colebrook sensivelmente menor do que o apresentado pela equação proposta por Haaland. A comparação entre as duas aproximações da equação de Colebrook pode ser vista na Tabela 2. Enquanto que a expressão de Haaland apresenta um erro relativo médio de 0,207% e um desvio padrão de 0,222%, a de Zigrang e Sylvester apresenta as mesmas medidas com os valores bem menores: erro relativo médio de 0,000612% e desvio padrão de 0,00615%.

Tabela 2 – Comparação entre aproximações explícitas da equação de Colebrook.

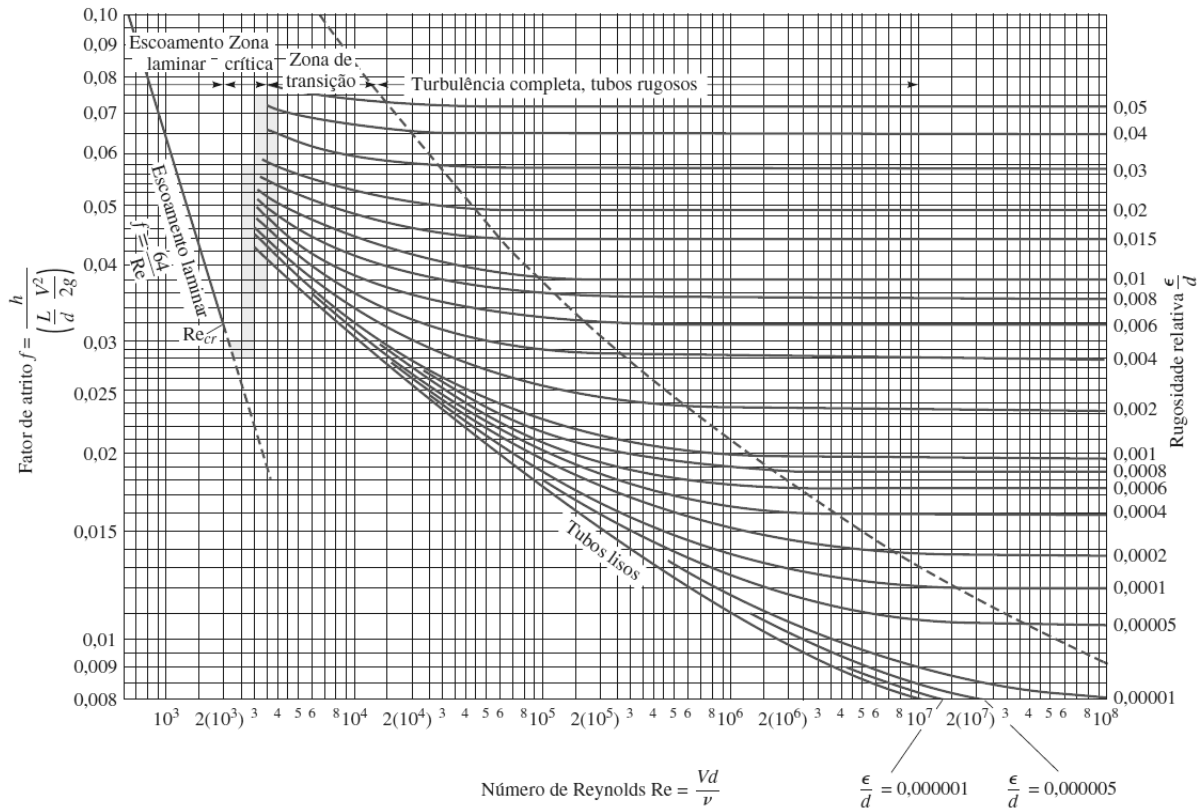
Autor	ER_{méd} (%)	ER_{máx+} (%)	ER_{máx-} (%)	dp (%)
Haaland (1983)	0,207	1,42	-1,314	0,222
Zigrang & Sylvester (1982)	0,000612	0,114	-0,0496	0,00615

Fonte: Adaptado de Genić et al. (2011, p. 69).

2.20 O diagrama de Moody

O diagrama de Moody é uma das ferramentas mais importantes em mecânica dos fluidos pois relaciona o fator de atrito de Darcy-Weisbach, f , ao número de Reynolds, Re , para diferentes valores de rugosidade relativa ϵ/D , e para os dois regimes de escoamento (laminar e turbulento). Esse diagrama foi plotado por Moody (1944), sendo apresentado na Figura 12.

Figura 12 – O Diagrama de Moody.



Fonte: White (2010, p. 376).

2.21 Versão final da equação da energia

Na seção 2.15 foi discutido o procedimento de obtenção da Equação 2.57, que também pode ser chamada de *equação da energia*, mostrada novamente a seguir:

$$\frac{P_1}{\rho g} + \frac{V_1^2}{2g} + z_1 + h_{bomba} = \frac{P_2}{\rho g} + \frac{V_2^2}{2g} + z_2 + h_{turbina} + h_{mec,perda} \quad (2.72)$$

Na seção 2.17, na parte que trata da *perda de carga total*, o termo $h_{mec,perda}$ é obtido na Equação 2.65, novamente reproduzida a seguir:

$$h_{mec,perda} = h_c + h_l \quad (2.73)$$

Substituindo o termo $h_{mec,perda}$ da Equação 2.65 na Equação 2.57, obtém-se a seguinte versão final da equação da energia que será usada neste trabalho (Equação 2.74):

$$\frac{P_1}{\rho g} + \frac{V_1^2}{2g} + z_1 + h_{bomba} = \frac{P_2}{\rho g} + \frac{V_2^2}{2g} + z_2 + h_{turbina} + h_c + h_l \quad (2.74)$$

A Equação 2.74 ainda pode ser manipulada, resultando em na Equação 2.75:

$$h_{bomba} - h_{turbina} = \frac{P_2 - P_1}{\rho g} + \frac{V_2^2 - V_1^2}{2g} + z_2 - z_1 + h_c + h_l \quad (2.75)$$

onde h_{bomba} é a carga fornecida ao sistema por uma bomba (em m), $h_{turbina}$ é a carga transferida do sistema a uma turbina (em m), P é a pressão (em Pa), ρ é a massa específica (em kg/m^3), g é a constante gravitacional (em m/s^2), V é a velocidade do fluido em escoamento (em m/s), z é a altura em relação a um referencial (em m), h_c é a perda de carga contínua (em m), e h_l é a perda de carga local (em m).

2.22 O head do sistema

O *head* (ou *carga manométrica*) de um sistema hidráulico, H_{sis} , é o valor de energia que deve ser adicionado ao fluido para que este vá de um ponto com pressão P_1 , velocidade V_1 e altura z_1 , a um outro ponto com pressão P_2 , velocidade V_2 e altura z_2 . De acordo com Henn (2006, p. 40), “é comum associar a energia recebida (caso das bombas) ou fornecida (caso das turbinas) pelo fluido, ao passar pela máquina, a uma altura de coluna de fluido, H (head).”

Considerando a Equação 2.75, verifica-se que a expressão do lado esquerdo da igualdade refere-se à uma diferença entre cargas fornecidas e extraídas do sistema respectivamente por uma bomba e por uma turbina hipotéticas. Como este trabalho lida apenas com o fornecimento de energia para um fluido ser elevado a uma altura específica, o termo $h_{turbina}$ da Equação 2.75 pode ser desprezado, e o termo h_{bomba} pode ser igualado a H_{sis} . Estas modificações resultam na Equação 2.76:

$$H_{sis} = \frac{P_2 - P_1}{\rho g} + \frac{V_2^2 - V_1^2}{2g} + z_2 - z_1 + h_c + h_l \quad (2.76)$$

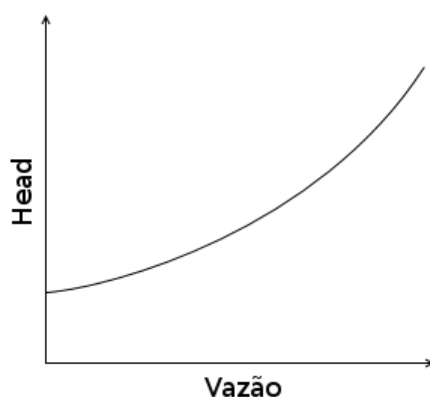
onde H_{sis} é o head do sistema (em m), P é a pressão (em Pa), ρ é a massa específica (em kg/m^3), g é a constante gravitacional (em m/s^2), V é a velocidade do fluido em escoamento (em m/s), z é a altura em relação a um referencial (em m), h_c é a perda de carga contínua (em m), e h_l é a perda de carga local (em m).

Aqui é importante frisar que apesar a dedução da Equação 2.76 ter usado termos como *bomba* e *turbina*, esta equação é característica *do sistema* e não de uma máquina de fluxo. Assim, H_{sis} refere-se à energia que um sistema hidráulico requer (no caso de uma bomba) ou fornece (no caso de uma turbina).

2.23 A curva do sistema

Devido às suas características próprias, os sistemas hidráulicos possuem um valor de carga manométrica para cada valor de vazão. A reunião desses valores em pares ordenados num plano cartesiano constitui aquilo que se convencionou chamar de *curva do sistema* (ou *curva do encanamento*, como em Macintyre (1997, p. 167)), sendo um caso típico ilustrado na Figura 13. O eixo das abscissas reúne os valores de vazão geralmente dados em metros cúbicos por hora, m^3/h , e o eixo das ordenadas reúne os valores de *head* expressos em metros, m .

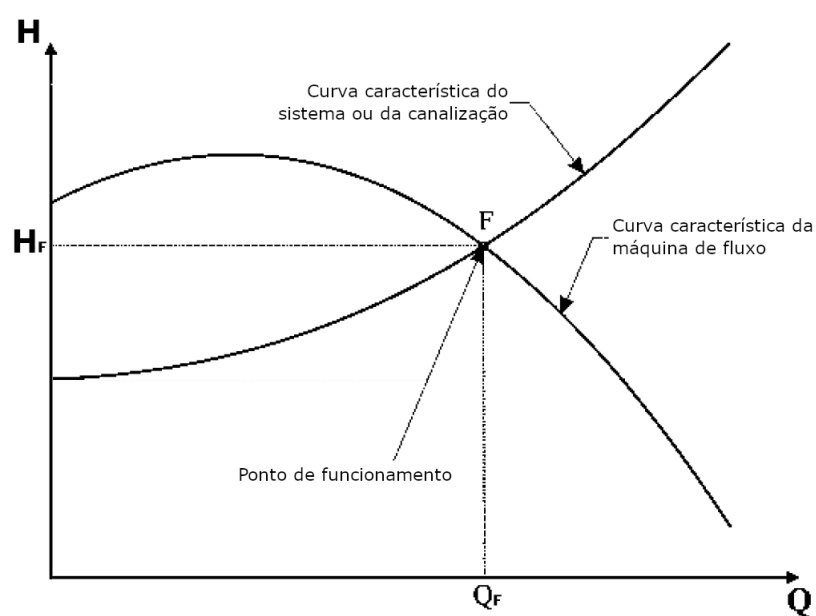
Figura 13 – A curva do sistema.



Fonte: Elaborado pelo autor.

A Figura 14 mostra as curvas características do sistema (ou da canalização) e da máquina de fluxo. O ponto de funcionamento dado pelo par ordenado (Q_F, H_F) , é o ponto comum às duas curvas. É com esses valores de vazão e carga manométrica (ou *head*) que a máquina de fluxo funcionará quando acoplada ao sistema hidráulico. Por isto, o correto dimensionamento e modelagem são imprescindíveis. A escolha de uma máquina de fluxo para um sistema hidráulico é condicionada às características deste mesmo sistema, ou seja, é necessário que se conheça bem a curva do sistema *a priori*, sendo, portanto, o processo de modelagem do sistema hidráulico uma etapa crítica na engenharia.

Figura 14 – As curvas características do sistema e da máquina de fluxo e o ponto de funcionamento na interseção entre as curvas.



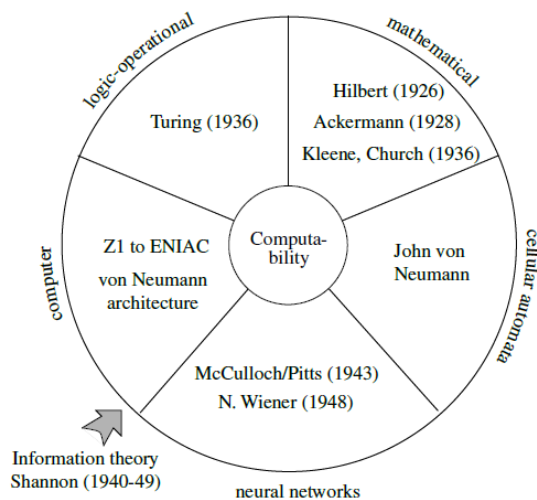
Fonte: Adaptado de Henn (2006, p. 230).

3 Conceitos computacionais

3.1 Os principais paradigmas computacionais

Durante o século XX, diferentes formas de abordar o problema da computabilidade foram desenvolvidas. Após décadas de pesquisas e desenvolvimento, emergiu dominante o paradigma de computação denominado *arquitetura de von Neumann*, sobre o qual se baseiam boa parte dos computadores modernos. Dentre os cinco modelos de computação existentes e elencados por Rojas (1996, p. 7) (v. Figura 15), o presente trabalho se baseará no modelo das redes neurais artificiais¹ (abrev.: *RNA*).

Figura 15 – Os cinco modelos computacionais que abordam o problema da computabilidade. O modelo de redes neurais (do inglês *neural networks*) é um deles.



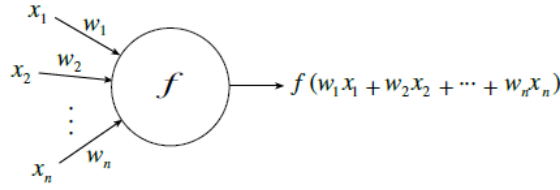
Fonte: Rojas (1996, p. 7).

3.2 O neurônio artificial

As redes neurais artificiais, introduzidas pela primeira vez em 1943 no trabalho de Warren McCulloch e Walter Pitts (MCCULLOCH; PITTS, 1943), têm como elemento fundamental o *neurônio artificial*. Este neurônio é um ente capaz de receber dados, processá-los e então fornecer um valor de saída. Uma representação genérica de um neurônio artificial pode ser vista na Figura 16.

¹ O termo *redes neurais artificiais* em inglês é traduzido para *artificial neural networks* (abrev.: *ANN*).

Figura 16 – Um neurônio artificial usa a função f como primeira etapa de processamento interno das diferentes entradas x_i ($i = 1, \dots, n$).



Fonte: Rojas (1996, p. 23).

3.3 Componentes do neurônio artificial

De acordo com Kriesel (2007), um neurônio artificial é associado a três funções: a função de propagação f_{prop} , a função de ativação f_{act} e a função de saída f_{out} . Além destas funções, é também associado a todo neurônio artificial um *viés* (ou *bias*, do inglês), que é basicamente um valor que contribuirá com o estado de ativação do neurônio.

A função de propagação

A função de propagação de um dado neurônio j , f_{prop} , é responsável por converter os valores de entrada de um neurônio (geralmente em forma vetorial) em um valor escalar. Esse escalar é então atribuído à variável denominada *entrada de rede*, net_j , como mostrado na Equação 3.1:

$$net_j = f_{prop}(o_{i_1}, \dots, o_{i_n}, w_{i_1,j}, \dots, w_{i_n,j}, b_j) \quad (3.1)$$

De acordo com Rumelhart, Widrow e Lehr (1994, p. 87), frequentemente se usa uma *soma ponderada*² como função de propagação. Assim, pode-se calcular a entrada de rede através da Equação 3.2:

$$net_j = \sum_{i \in I} (o_i \cdot w_{i,j}) + b_j \quad (3.2)$$

onde o_i é o valor fornecido por um neurônio i pertencente a uma camada de neurônios I anterior à camada do neurônio j , $w_{i,j}$ é o *peso* da conexão entre os neurônios i e j , e b_j é o valor do *viés* associado ao neurônio j . Como já foi mencionado, net_j é um escalar resultante da função de propagação.

² Do inglês *weighted sum*.

A função de ativação

A função de ativação de um dado neurônio j , f_{act} , é responsável por processar o valor fornecido pela função de propagação; em outras palavras, f_{act} recebe como entrada o valor net_j , que é a entrada de rede, e calcula a *ativação* do neurônio, a_j . A ativação de um neurônio (ou *estado de ativação*) é uma medida do quão ativo esse neurônio está. Assim, pode-se expressar a ativação de um neurônio j pela Equação 3.3:

$$a_j = f_{act}(net_j) \quad (3.3)$$

onde net_j é a entrada de rede fornecida pela função de propagação.

Segundo Rojas (1996, p. 151), o algoritmo de retropropagação do erro³ requer que se use uma função de ativação que seja contínua e diferenciável, e por este motivo geralmente usa-se uma *função sigmóide*. As formas mais comuns de funções sigmóides utilizadas em redes neurais são citadas por Kriesel (2007, p. 37), dentre as quais estão a *função logística* (ou *função de Fermi*) dada pela Equação 3.4:

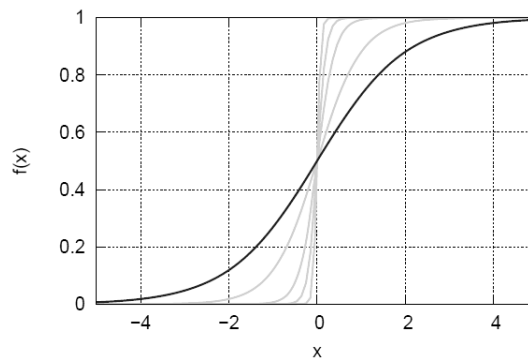
$$f(x) = \frac{1}{1 + e^{-x/T}} \quad (3.4)$$

e a *função tangente hiperbólica* expressa pela Equação 3.5:

$$f(x) = \tanh(x) = \frac{1 - e^{-2x}}{1 + e^{-2x}} \quad (3.5)$$

Na função logística (Equação 3.4) há um parâmetro T denominado *parâmetro de temperatura* cuja variação gera uma família de funções logísticas, como pode ser visto na Figura 17.

Figura 17 – A função logística representada por várias curvas, cada uma com um parâmetro de temperatura T particular.

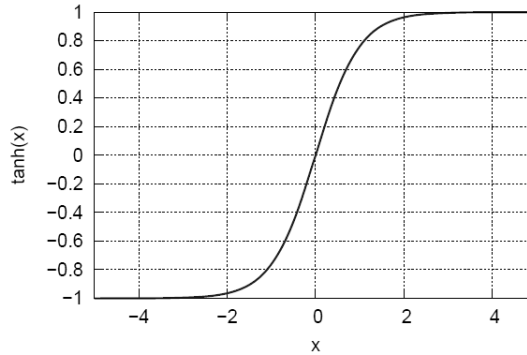


Fonte: Kriesel (2007, p. 38).

³ O algoritmo de retropropagação do erro será abordado mais adiante na seção 3.11.

A função tangente hiperbólica (Equação 3.5) apresenta um aspecto muito semelhante ao da função logística já que ambas são sigmóides. Uma diferença entre elas diz respeito às suas saídas: enquanto que a tangente hiperbólica resulta em valores que vão de -1 à 1 , a função logística varia de 0 à 1 . Uma representação gráfica da função tangente hiperbólica é dada na Figura 18.

Figura 18 – A função tangente hiperbólica resulta em valores que vão de -1 à 1 .



Fonte: Kriesel (2007, p. 38).

Como foi mencionado anteriormente, o fato de as funções de tipo sigmóide serem contínuas e diferenciáveis em todo o seu domínio é o que torna possível o treinamento das redes neurais artificiais usando algoritmos de aprendizado como a *Regra Delta* (subseção 3.11.1) que dá origem ao algoritmo de retropropagação do erro. Rojas (1996, p. 152) realiza o processo de diferenciação em relação a x da função logística (Equação 3.4), obtendo, de fato, uma expressão simples dada pela Equação 3.6:

$$\frac{d}{dx}f(x) = \frac{e^{-x}}{(1 + e^{-x})^2} = f(x)(1 - f(x)) \quad (3.6)$$

A função de saída

A função de saída, f_{out} , é responsável por fazer um processamento adicional dentro do neurônio. No entanto, segundo Kriesel (2007, p. 38), na maioria das aplicações a função de saída é a *identidade*, isto é, a mesma fornece como saída o próprio valor de ativação recebido por ela. Rojas (1996, p. 125) chega a afirmar que a função de ativação é também chamada de função de saída, portanto, não fazendo distinção entre ambas. Expressando este fato em termos matemáticos, tem-se a Equação 3.7:

$$o_j = f_{out}(a_j) = a_j \quad (3.7)$$

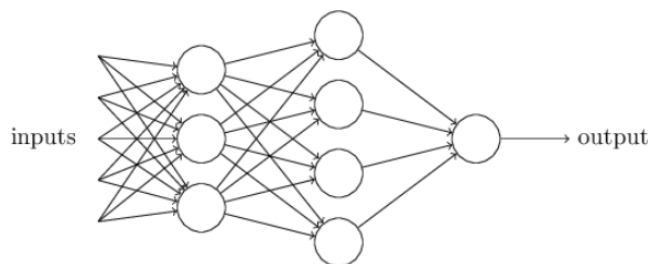
onde o_j é o valor de saída do neurônio calculado pela função de saída f_{out} , e a_j é a ativação do neurônio j .

3.4 Redes neurais artificiais

As redes neurais artificiais (abrev.: *RNA*)⁴ são constituídas por diferentes neurônios artificiais conectados entre si formando, nas palavras de Zurada (1992, p. 2), “uma malha densa de nós de computação e conexões” que “operam coletiva e simultaneamente na maioria ou em todos os dados e entradas”. As conexões interneuronais são caracterizadas por um fator atribuído a elas chamado *peso*, geralmente simbolizado por $w_{i,j}$ ou $\omega_{i,j}$, e que reflete o grau de importância daquela conexão entre os dois neurônios participantes i e j .

As redes neurais artificiais são divididas em camadas, em que a primeira é a *camada de entrada* por onde os dados de entrada (ou *inputs*) são primeiramente apresentados à rede, e a última, a *camada de saída*, que fornece o resultado da rede como dados de saída (ou *outputs*). As camadas intermediárias são denominadas *camadas ocultas*, sendo estas compostas por neurônios que recebem dados das camadas anteriores, os processam, e então repassam o produto do processamento para os neurônios das camadas posteriores.

Figura 19 – Uma rede neural artificial genérica em que os dados de entrada (*inputs*) entram na primeira camada e são processados pela rede, resultando no dado de saída (*output*). Os pesos $w_{i,j}$ foram omitidos por questão de simplicidade.



Fonte: Nielsen (2015).

Os neurônios da camada de saída receberão uma notação especial nesse trabalho, que é a letra grega ômega maiúscula, Ω .

É importante observar que apesar de a Figura 19 apresentar 3 neurônios na camada de entrada, 4 na camada oculta, e 1 na camada de saída, uma rede neural pode ter qualquer quantidade de neurônios em cada uma dessas camadas⁵.

3.5 Capacidade de generalização de uma RNA

De acordo com Rojas (1996, p. 229):

⁴ Zurada (1992) apresenta vários termos sinônimos às redes neurais: neurocomputação, computação em rede, connexionismo, processamento distribuído paralelo, sistemas adaptativos em camadas, redes auto-organizáveis, e sistemas (ou redes) neuromórficos.

⁵ Essa escolha, no entanto, está subordinada à natureza do problema que se quer resolver, não fazendo sentido, por exemplo, o uso de mais neurônios na camada de entrada do que a quantidade de parâmetros de entrada presentes num problema.

Por um lado, nós esperamos que a rede mapeie as entradas conhecidas em relação às saídas conhecidas o mais exatamente possível. Mas, por outro lado, a rede precisa ser capaz de *generalizar*, isto é, entradas desconhecidas devem ser comparadas às conhecidas e a saída produzida ser um tipo de interpolação de valores aprendidos.

Adicionalmente, Kriesel (2007, p. 51) diz:

[...] a característica mais interessante de redes neurais é a sua capacidade de familiarizar com problemas por meio de treinamento e, após treino suficiente, ser capaz de resolver problemas desconhecidos da mesma classe.

Uma rede neural artificial adequadamente treinada é capaz de *generalizar*. O processo de generalização, no âmbito das redes neurais artificiais, consiste em inferir resultados corretos para padrões de entrada não vistos anteriormente durante a fase de treinamento. A capacidade de generalização é, na verdade, o próprio motivo de uso das redes neurais artificiais, de modo que ao falhar na obtenção desta capacidade, a rede será apenas um mero dispositivo de armazenamento alternativo, pois terá apenas memorizado os resultados relacionados aos padrões que foi submetida durante o seu treinamento, não sendo, portanto, capaz de extrapolar os limites do conjunto de treinamento.

Fazendo uma breve analogia ao cérebro animal, a capacidade de generalização torna este capaz de prever a queda de uma maçã do alto de uma árvore depois de ter presenciado casos específicos em que *outros objetos* caíam quando expostos à ação gravitacional. O que é o cérebro animal fez ao presenciar aquelas outras quedas foi *generalizar*, isto é, o cérebro formulou uma ideia, um princípio, segundo o qual todos os objetos, ao serem submetidos à força da gravidade, caem. Com isto, mesmo sem nunca ter visto uma maçã despencar do alto de uma árvore, o animal é capaz de conjecturar, com razoável certeza, que isto ocorrerá, já que presenciou anteriormente a queda de outros objetos. O mesmo atributo pode ser conferido às redes neurais artificiais quando as mesmas são submetidas a um número suficiente de exemplos.

3.6 Modos de aprendizado das RNAs

Nas palavras de Rumelhart, Widrow e Lehr (1994, p. 89):

O problema do aprendizado em redes neurais artificiais é simplesmente o problema de se encontrar um conjunto de pesos das conexões que permitam à rede fazer a computação desejada.

Uma rede neural artificial *aprende* através de ajustagens nas conexões entre os seus diferentes neurônios até que o resultado fornecido *efetivamente* pela rede se aproxime

suficientemente do resultado *desejado* (ou *alvo*⁶). Dito de outro modo: para cada neurônio i , o peso da sua conexão ao neurônio j , isto é, $w_{i,j}$, vai sofrendo sucessivas alterações na etapa de aprendizado até que a saída efetiva da rede neural esteja a uma distância satisfatória do resultado desejado (alvo). A esta distância entre saída efetiva e saída desejada dá-se o nome de *erro*, que será abordado com mais detalhes na seção 3.10.

De acordo com Kriesel (2007, p. 52-53), existem basicamente três modos pelos quais uma rede neural pode aprender:

- Aprendizado não-supervisionado;
- Aprendizado por reforço;
- Aprendizado supervisionado.

No presente trabalho se usará somente o aprendizado supervisionado. Neste tipo de aprendizado, uma dada rede neural artificial é exposta a um conjunto de treinamento P que contém em si $|P|$ duplas de padrões de entrada p e alvos t ⁷, com cada um dos dois elementos podendo ter qualquer dimensão n . Ou seja, P é um conjunto composto por $|P|$ duplas (p, t) , em que p e t podem ser vetores de qualquer dimensão n (Equação 3.8):

$$P = \{(p, t)_1, \dots, (p, t)_{|P|} \mid p = (p_1, \dots, p_n), t = (t_1, \dots, t_n)\} \quad (3.8)$$

3.7 O perceptron

Um perceptron é uma espécie de rede neural do tipo *feedforward*⁸ tendo sido proposto pela primeira vez por Rosenblatt (1958) como um modelo computacional mais genérico que as unidades de McCulloch e Pitts (1943). De acordo com Rojas (1996, p. 56), “de um ponto de vista formal, a única diferença entre os elementos de McCulloch-Pitts e os perceptrons é a presença de pesos nas redes.”

Segundo Kriesel (2007, p. 73):

O perceptron é uma rede neural *feedforward* contendo uma retina que é usada somente para aquisição de dados e que tem conexões de pesos fixos com a primeira camada de neurônios (camada de entrada). A camada de pesos fixos é seguida por pelo menos uma camada de pesos ajustáveis. Uma camada neuronal é completamente conectada à próxima camada. A primeira camada do perceptron consiste nos neurônios de entrada definidos acima.

⁶ *Target*, em inglês.

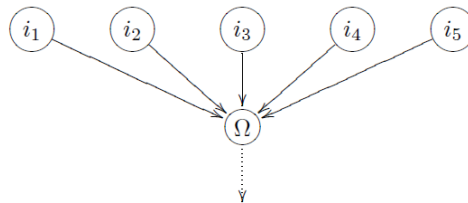
⁷ Os alvos t recebem o nome de *target* ou *teaching input* na literatura em língua inglesa.

⁸ Uma rede neural *feedforward* é aquela em que os dados vão somente na direção da camada de saída a partir da camada de entrada. Não existe interferência de um neurônio com ele mesmo, seja direta ou indiretamente, como ocorre em redes do tipo *recorrente*. Segundo Rumelhart, Widrow e Lehr (1994), em redes *feedforward* a atividade de uma dada unidade não pode influenciar, mesmo indiretamente, as suas próprias entradas.

3.7.1 O perceptron de camada única - SLP

Conforme afirmado por Kriesel (2007, p. 74), “um perceptron de camada única⁹ é um perceptron tendo apenas uma camada de pesos variáveis e uma camada de neurônios de saída Ω .” Esse tipo básico de perceptron (v. Figura 20) é capaz de resolver apenas problemas do tipo *linearmente separáveis*¹⁰.

Figura 20 – Um perceptron de camada única. Cada conexão dos neurônios i_n com o neurônio Ω tem um peso associado.



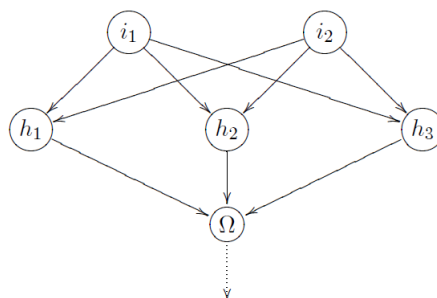
Fonte: Kriesel (2007, p. 72).

3.7.2 O perceptron multicamadas - MLP

O *perceptron multicamadas* (do inglês *multilayer perceptron*, MLP) é um desenvolvimento do perceptron de camada única (SLP). Como o próprio nome aponta, o perceptron multicamadas possui mais de uma camada, com uma representação do mesmo sendo mostrada na Figura 21.

De acordo com Kriesel (2007) os perceptrons multicamadas são mais poderosos do que os de camada única, já que não estão restritos apenas a problemas linearmente separáveis, como os SLPs.

Figura 21 – Um perceptron multicamadas.



Fonte: Kriesel (2007, p. 85).

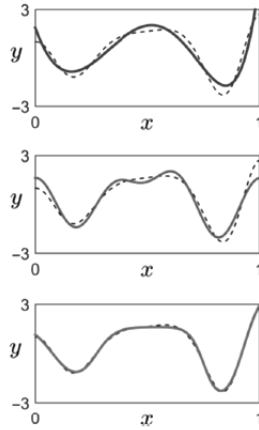
De acordo com o Teorema de Cybenko (CYBENKO, 1989), um perceptron multicamadas é um tipo de *aproximador universal de funções*. Como se pode ver na Figura 22,

⁹ Do inglês *single-layer perceptron*, SLP

¹⁰ Problemas linearmente separáveis são aqueles em que retas, planos e hiperplanos são suficientemente capazes de dividir conjuntos de dados em grupos separados.

uma rede neural artificial constitui um excelente aproximador, já que a linha que representa esta aproximação (v. o gráfico mais abaixo na Figura 22) se aproxima da função tão bem quanto as outras duas (polinômio e série de Fourier).

Figura 22 – De cima para baixo, funções aproximadas por um polinômio, por uma série de Fourier, e por uma rede neural artificial. As linhas tracejadas correspondem às funções, e as linhas contínuas, às aproximações.



Fonte: Adaptado de Watt, Borhani e Katsaggelos (2016, p. 138).

3.8 Minimização de uma função através do gradiente descendente

O gradiente descendente é um método matemático cujo objetivo é encontrar o valor mínimo de uma dada função através de sucessivas modificações dos valores das variáveis independentes. O mínimo encontrado poderá ser *local* ou *global*.

Gradiente de uma função

De acordo com Leithold (1994, p. 975-976), dada uma função $f = f(x_1, x_2, \dots, x_n)$, o seu gradiente g é um vetor de dimensão n que aponta na direção de maior crescimento de f num dado ponto. Em outras palavras, tem-se a Equação 3.9 expressando este fato como:

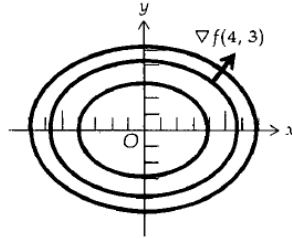
$$g(x_1, x_2, \dots, x_n) = \nabla f(x_1, x_2, \dots, x_n) \quad (3.9)$$

em que o operador ∇ (*nabla*) deriva a função f em relação a cada uma das variáveis x_i (i de 1 à n) e soma essas derivadas parciais resultando num vetor de dimensão n como pode ser visto na equação Equação 3.10:

$$\nabla f(x_1, x_2, \dots, x_n) = \frac{\partial f}{\partial x_1} \hat{x}_1 + \frac{\partial f}{\partial x_2} \hat{x}_2 + \dots + \frac{\partial f}{\partial x_n} \hat{x}_n \quad (3.10)$$

onde \hat{x}_i é o vetor unitário na direção i , e $\partial f/\partial x_i$ é a derivada parcial da função f em relação a x_i . Pode-se dizer que o gradiente de uma função é a generalização da ideia de derivada para n dimensões.

Figura 23 – O gradiente de uma função f no ponto $(4,3)$.



Fonte: Leithold (1994, p. 976).

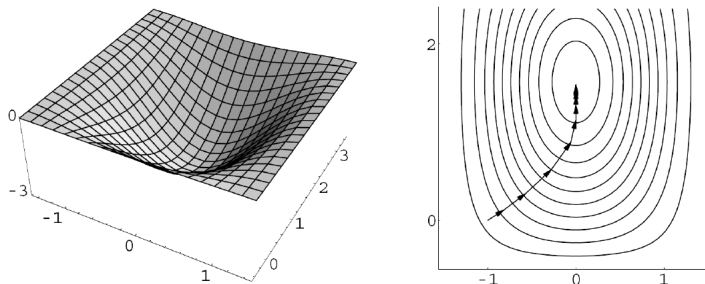
Gradiente descendente

O método de minimização de uma função por gradiente descendente é definido nas palavras de Press et al. (1988, p. 515):

Gradiente descendente: inicie em um ponto P_0 . Quantas vezes forem necessárias, mova-se do ponto P_i ao ponto P_{i+1} minimizando ao longo da linha de P_i na direção do gradiente decrescente $-\nabla f(P_i)$.

O método do gradiente descendente busca minimizar uma função f iterativamente. Seja f uma função n -dimensional e $s = (s_1, s_2, \dots, s_n)$ um ponto qualquer pertencente ao domínio desta função. O procedimento do gradiente descendente consiste em partir de $f(s)$ numa direção contrária ao gradiente $g = \nabla f$, isto é, na direção de $-g = -\nabla f$ com incrementos de tamanho $|g|$, o que resultará naturalmente em valores cada vez menores de f . Este método é ilustrado na Figura 24.

Figura 24 – Dada uma função f usa-se o método do gradiente descendente para minimizá-la. O gradiente descendente parte de um valor e a cada iteração vai em direção a um menor valor de f .



Fonte: Kriesel (2007, p. 62).

3.9 A lei de aprendizado de Hebb

Em 1949 o biólogo Donald O. Hebb propôs no seu célebre artigo *The organization of behavior: A neuropsychological theory* (HEBB, 1949) uma lei genérica que define o modo de treinamento de uma rede neural artificial. Segundo a *lei de aprendizado de Hebb*, se dois neurônios artificiais estão conectados e se ambos estão fortemente ativos num determinado instante, então o peso da conexão entre ambos deve ser incrementado, resultando numa conexão ainda mais significativa, isto é, com um maior valor de peso.

Mais tarde, Rumelhart e McClelland (1986) propuseram uma forma ainda mais genérica para a lei de aprendizado de Hebb. Segundo esta forma, a variação do peso da conexão entre dois neurônios, $\Delta w_{i,j}$, deveria se dar de acordo com a Equação 3.11:

$$\Delta w_{i,j} = \eta \cdot h(o_i, w_{i,j}) \cdot g(a_j, t_j) \quad (3.11)$$

onde $\Delta w_{i,j}$ é a alteração do peso da conexão entre os neurônios i e j , η é uma constante chamada de *coeficiente de aprendizado*, o_i é o valor de saída do neurônio i , $w_{i,j}$ é o peso da conexão entre os neurônios i e j , a_j é a ativação do neurônio j , e t_j é o valor de saída desejado (ou *alvo*) para o neurônio j . Já as funções h e g não são definidas, mas a explicitação das mesmas surgirá naturalmente de acordo com cada algoritmo de treinamento da rede neural como a *Regra Delta*, por exemplo, que será abordada na subseção 3.11.1.

3.10 A função erro

Quando se propõe uma rede neural com a finalidade de modelar algum fenômeno, deve-se ter algum parâmetro que seja capaz de medir o quão bem esta rede está realizando a tarefa proposta. Uma forma de medir a qualidade da rede é através da avaliação da diferença entre os resultados efetivos fornecidos por ela e os resultados desejados (ou *alvos*), diferença esta que será chamada de *erro*.

Como os resultados fornecidos por uma rede neural artificial estão diretamente relacionados aos pesos de suas conexões, pode-se dizer que uma dada *função erro*, Err , recebe como entrada o vetor de pesos, W , e fornece como saída um valor real. Em termos matemáticos, Kriesel (2007, p. 77) define (Equação 3.12):

$$Err : W \rightarrow \mathbb{R} \quad (3.12)$$

A Equação 3.12 mostra de maneira genérica que a função erro relaciona os pesos da rede com os seus resultados reais. A forma específica da função erro que será considerada

neste texto é a mesma escolhida por Rojas (1996, p. 156) e Kriesel (2007, p. 78), sendo dada pela Equação 3.13:

$$Err_p(W) = \frac{1}{2} \sum_{\Omega \in O} (t_{p,\Omega} - y_{p,\Omega})^2 \quad (3.13)$$

em que $Err_p(W)$ é o erro específico para o padrão de treinamento p , Ω é o neurônio da camada de saída O , $t_{p,\Omega}$ é o valor de saída desejado para o neurônio Ω dado um padrão de entrada p , e $y_{p,\Omega}$ é o valor de saída atual do neurônio Ω ¹¹ dado um padrão de entrada p .

Como mencionado, Err_p é o valor da função erro para um único padrão de entrada $p \in P$, isto é, Err_p um *erro específico*. Já o *erro total* Err é obtido pela soma dos erros específicos referentes a todos os padrões de entrada p . Segundo Kriesel (2007, p. 79), o erro total é dado pela Equação 3.14:

$$Err(W) = \sum_{p \in P} Err_p(W) \quad (3.14)$$

Se o termo Err_p da Equação 3.14 for substituído pela expressão dada na Equação 3.13, obtém-se a Equação 3.15:

$$Err(W) = \frac{1}{2} \sum_{p \in P} \sum_{\Omega \in O} (t_{p,\Omega} - y_{p,\Omega})^2 \quad (3.15)$$

em que $Err(W)$ é o erro total, $t_{p,\Omega}$ é o valor de saída desejado para o neurônio Ω dado um padrão de entrada p , e $y_{p,\Omega}$ é o valor de saída atual do neurônio Ω dado um padrão de entrada p .

3.11 Algoritmos de treinamento de redes neurais

O aprendizado (ou *treinamento*) de uma rede neural artificial ocorre através da modificação de seus elementos. No presente trabalho, será considerado o aprendizado que ocorre através dos ajustes dos pesos das conexões. Esses ajustes são feitos através de algoritmos de aprendizado¹² que buscam modificar os pesos da rede neural de modo que a função erro seja minimizada, buscando-se assim mínimos (locais ou globais) desta função. Os mínimos da função erro implicam, naturalmente, em resultados mais precisos, já que são obtidas saídas da rede mais próximas dos valores desejados, isto é, ao longo do treinamento da rede neural, *y tende a t*, que é o objetivo final da rede neural.

¹¹ Como o neurônio da camada de saída é, também, a saída da rede neural, tem-se que a equivalência $y_{p,\Omega} = o_{p,\Omega}$ é válida.

¹² *Learning algorithms*, em inglês.

3.11.1 A Regra Delta

A *Regra Delta* (também chamada de *Regra Widrow-Hoff*, em homenagem aos pesquisadores que a formularam (WIDROW, 1962)) é um algoritmo de aprendizado deduzido de um método de gradiente descendente para a função erro e resulta num caso específico da lei de aprendizado de Hebb.

A dedução da Regra Delta parte da premissa de que a mudança dos pesos das conexões de uma rede neural deve ser proporcional ao negativo do gradiente da função erro. De acordo com Kriesel (2007, p. 78), essa ideia é expressa pela Equação 3.16:

$$\Delta W = -\eta \nabla \text{Err}(W) \quad (3.16)$$

onde W é o vetor contendo os valores dos pesos das conexões da rede neural, η é uma constante de proporcionalidade chamada *coeficiente de aprendizado*, e $\text{Err}(W)$ é a função erro.

Após uma extensa dedução matemática, chega-se à Regra Delta, que é aplicável a perceptrons de uma camada (SLPs), e à *retropropagação do erro*, usada em perceptrons multicamadas (MLPs). As duas formas serão tratadas a seguir.

A Regra Delta para um SLP

Para um perceptron de camada única (SLP), Kriesel (2007, p. 81) afirma que a Regra Delta adquire a forma da Equação 3.17:

$$\Delta w_{i,\Omega} = \eta \cdot o_i \cdot (t_\Omega - o_\Omega) = \eta o_i \delta_\Omega \quad (3.17)$$

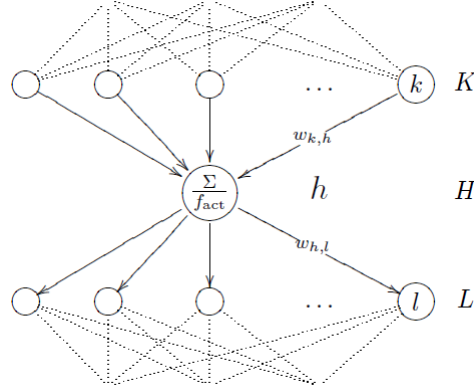
onde $\Delta w_{i,j}$ é a alteração do peso de uma conexão entre os neurônios i e Ω , η é o coeficiente de aprendizado, o_i é o valor fornecido como saída pelo neurônio i , t_Ω é o valor de saída desejado para o neurônio Ω , o_Ω é a saída efetiva do neurônio Ω , e δ_Ω é a diferença $t_\Omega - o_\Omega$.

Retropropagação do erro

A *retropropagação do erro* consiste numa generalização da Regra Delta para um perceptron multicamadas (MLP), sendo esta técnica publicada por Werbos (1974). Quando se usa a retropropagação do erro, o que a rede neural faz é enviar para os neurônios das camadas anteriores a informação do erro cometido. Com isto, a informação do erro propaga-se para trás, isto é, se *retropropaga*.

A forma da Regra Delta para perceptrons multicamadas considerará a topologia genérica ilustrada na Figura 25.

Figura 25 – Um perceptron multicamadas com as três camadas K, H e L em destaque.



Fonte: Kriesel (2007, p. 87).

A forma matemática da retropropagação do erro é dada pela Equação 3.18:

$$\Delta w_{k,h} = \eta o_k \delta_h \quad (3.18)$$

onde $\Delta w_{k,h}$ é a alteração do peso de uma conexão entre os neurônios k e h , η é o coeficiente de aprendizado, o_k é o valor de saída do neurônio k , e δ_h é uma forma generalizada do δ que estará condicionado à posição da camada H na rede neural.

De acordo com Kriesel (2007, p. 90), se a camada H for a camada de saída da rede, δ_h será calculado com a Equação 3.19:

$$\delta_h = f'_{act}(net_h) \cdot (t_h - y_h) \quad (3.19)$$

onde $f'_{act}(net_h)^{13}$ é a derivada da função de ativação f_{act} em relação à entrada de rede net_h , t_h é a saída esperada do neurônio h e y_h é a saída efetiva do mesmo¹⁴.

Se, no entanto, a camada H não for a camada de saída da rede neural, Kriesel (2007, p. 90) afirma que o cálculo de δ_h será feito através da Equação 3.20:

$$\delta_h = f'_{act}(net_h) \cdot \sum_{l \in L} (\delta_l w_{h,l}) \quad (3.20)$$

onde $f'_{act}(net_h)$ é a derivada da função de ativação f_{act} em relação à entrada de rede net_h , δ_l é o δ pertencente ao neurônio da camada L que já terá sido calculado através da Equação 3.19, e $w_{h,l}$ é o peso da conexão que conecta um neurônio h ao neurônio l da camada seguinte.

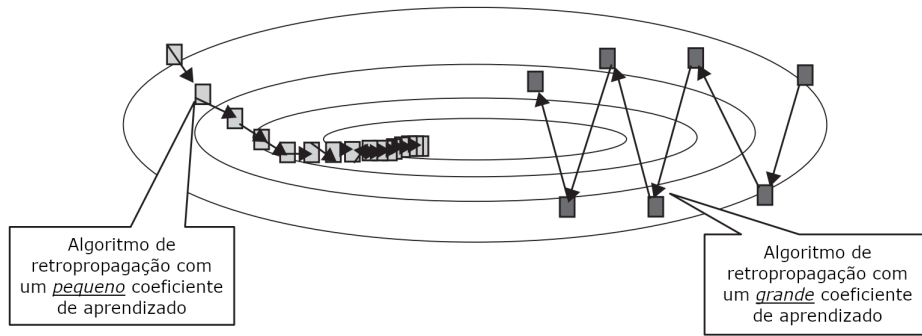
¹³ Na dedução segundo Kriesel (2007), $f'_{act}(net_h)$ equivale à $\partial f_{act}(net_h) / \partial net_h$.

¹⁴ Que equivale à saída da rede, já que a camada H, neste caso, é a última camada.

3.11.2 O algoritmo de Levenberg-Marquardt

De acordo com Wilamowski e Irwin (2016), a Regra Delta baseada no método de gradiente descendente (discutida na subseção 3.11.1) possui como principal desvantagem uma convergência lenta (v. Figura 26). Essa desvantagem pode ser mitigada usando-se o algoritmo de Gauss-Newton, com a restrição de que a superfície da função erro possa ser aproximada suficientemente por uma função quadrática, o que nem sempre é possível.

Figura 26 – Diagrama mostrando os “caminhos” dos algoritmos de retropropagação do erro com um pequeno (à esq.) e um grande (à dir.) coeficiente de aprendizado.



Fonte: Adaptado de Wilamowski e Irwin (2016, p. 2).

O algoritmo de Levenberg-Marquardt desenvolvido nos trabalhos de Levenberg (1944) e Marquardt (1963), fornece uma solução numérica ao problema de minimização de uma função não-linear, além de ter uma convergência rápida e estável quando comparado a outros algoritmos. O algoritmo de Levenberg-Marquardt mistura o método de gradiente descendente com o algoritmo de Gauss-Newton, herdando deste último a rapidez de convergência, bem como a estabilidade do primeiro método.

Segundo Wilamowski e Irwin (2016), a ideia básica do algoritmo de Levenberg-Marquardt é realizar um processo de treinamento combinado: numa área em que a superfície da função erro assume uma curvatura complexa, o algoritmo muda para o método de gradiente descendente até que a curvatura local se torne apropriada para que se realize uma aproximação quadrática; nesse ponto, então, se passa a usar o método de Gauss-Newton, o que aumenta a velocidade de convergência significativamente.

No algoritmo de Levenberg-Marquardt a atualização dos pesos ocorre como na Equação 3.21:

$$\mathbf{w}_{k+1} = \mathbf{w}_k - (\mathbf{J}_k^T \mathbf{J}_k + \mu \mathbf{I})^{-1} \mathbf{J}_k \mathbf{e}_k \quad (3.21)$$

onde \mathbf{w}_{k+1} e \mathbf{w}_k são os valores de pesos nas iterações $k + 1$ e k , respectivamente, \mathbf{J}_k é o jacobiano da função erro em relação aos pesos w , μ é um coeficiente sempre positivo

introduzido na dedução do algoritmo de Levenberg-Marquardt, \mathbf{I} é a matriz identidade, e \mathbf{e}_k é o vetor erro.

De acordo com Wilamowski e Irwin (2016), o uso da Equação 3.21 faz com que o algoritmo de Levenberg-Marquardt varie entre os algoritmos de Gauss-Newton e de gradiente descendente durante o treinamento da rede neural. Quando o coeficiente μ é muito pequeno (próximo de zero), a Equação 3.21 aproxima-se da equação usada no método de Gauss-Newton; por outro lado, quando o valor de μ é muito grande, a equação Equação 3.21 se aproxima daquela utilizada no método de gradiente descendente. Dessa forma o processo de treinamento da rede neural através do algoritmo de Levenberg-Marquardt se beneficia dos dois métodos mencionados.

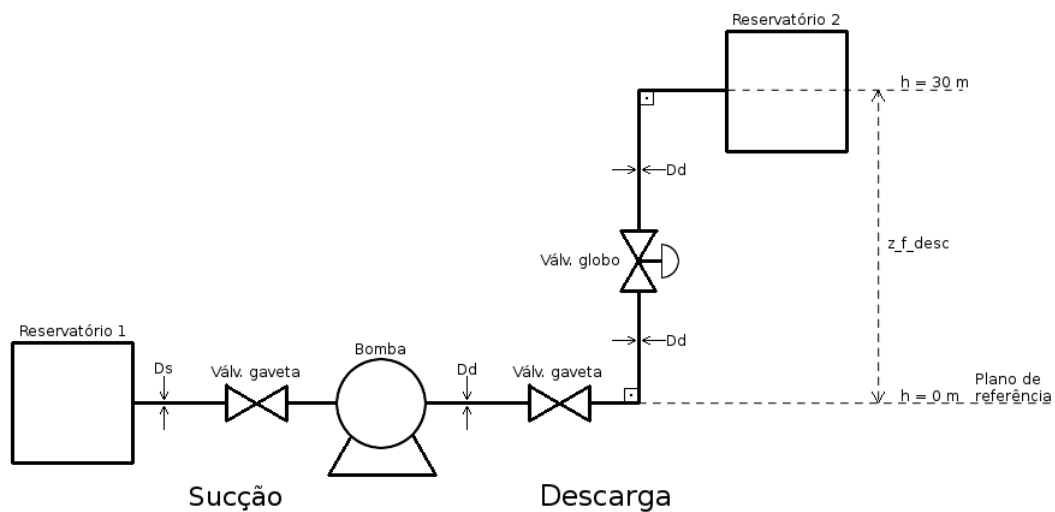
4 Procedimentos e metodologia

Neste capítulo se definirão o sistema hidráulico, os parâmetros (constantes e variáveis), e se discutirá o algoritmo da expressão fenomenológica bem como o processo de treinamento de uma rede neural artificial capaz de simular esta expressão. Por fim, se falará como será feita a comparação entre os resultados fornecidos pela rede neural e a expressão fenomenológica, para fins de avaliação da primeira.

4.1 Definição do sistema hidráulico

O sistema hidráulico considerado neste trabalho consiste de dutos de PVC¹ conectados em série, cujo objetivo é transportar água líquida de um reservatório a outro, estando este último a uma altura pré-definida de 30 metros. A Figura 27 mostra um diagrama do sistema citado.

Figura 27 – Diagrama do sistema hidráulico escolhido.



Fonte: Elaborado pelo autor.

O sistema hidráulico mostrado na Figura 27 divide-se em duas regiões principais: à esquerda da bomba situa-se a seção denominada *sucção*, e à direita, a região denominada *descarga*.

Além dos dutos, a sucção contém uma válvula gaveta. Já a descarga conta com os seus dutos, com uma válvula gaveta, com uma válvula globo, além de dois cotovelos de

¹ PVC é um acrônimo para *policloreto de vinila* (em inglês: *polyvinyl chloride*), material amplamente empregado na fabricação de tubos comerciais.

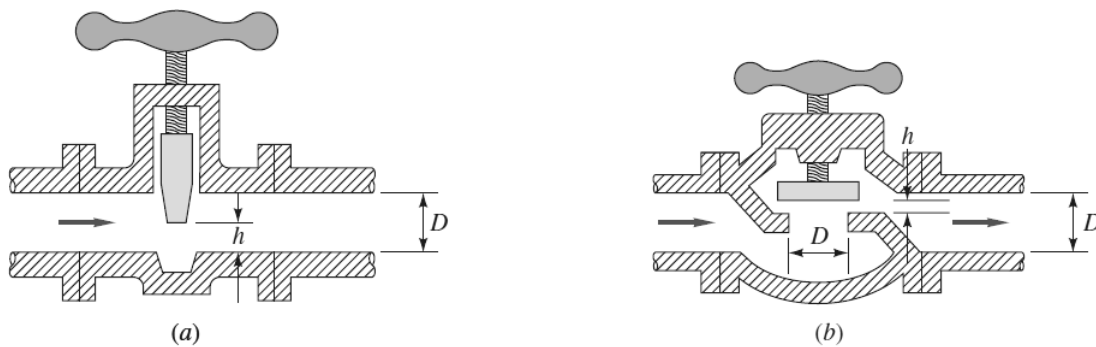
noventa graus de inclinação, que são dutos mudando de direção rumo ao reservatório mais elevado.

Apesar de uma bomba aparecer na Figura 27, o presente trabalho não se concentrará nesta máquina de fluxo, mas sim no sistema hidráulico constituído pelos dutos e acessórios. A bomba na Figura 27 aparece como forma de dividir o sistema nas regiões de sucção e descarga, e também como um meio de mostrar que o sistema é destinado à elevação, já que bombas são máquinas de fluxo geralmente empregadas para este fim.

O tipo de duto adotado é o PVC (policloreto de vinila), sendo este tipo amplamente utilizado nos mais diversos projetos de engenharia e também com uma boa quantidade de literatura disponível.

Finalmente, é importante mencionar que as válvulas têm abertura total, sendo os tipos de válvulas usados neste trabalho (gaveta e globo) mostradas na Figura 28.

Figura 28 – Válvulas presentes no sistema hidráulico: a) Válvula gaveta e b) Válvula globo.



Fonte: Adaptado de White (2010, p. 396).

4.2 Parâmetros, constantes e variáveis

No total, 14 parâmetros são considerados, sendo que 7 deles são variáveis, e os outros 7, constantes. Todos estes parâmetros (variáveis e constantes) estão descritos na Tabela 3².

Os valores das constantes são listados na Tabela 4, e as faixas de valores para as variáveis são definidas na Tabela 5.

O valor de rugosidade absoluta utilizado neste trabalho (0,1 mm) encontra-se na faixa resultante da Tabela 1 obtida por Kellner, Akutsu e Reis (2016).

² Os nomes de alguns parâmetros na Tabela 3, como *rho* e *mu*, estão em inglês por deliberação do próprio autor.

Tabela 3 – Variáveis e constantes físicas.

Parâmetro	Tipo	Nome	Unidade
Vazão volumétrica	Variável	Q_vazao	m^3/h
Pressão no final da descarga	Variável	pfd	kPa
Diâmetro da tubulação de descarga	Variável	Dd	m
Comprimento da tubulação de descarga	Variável	Ld	m
Pressão no início da sucção	Variável	p0s	kPa
Diâmetro da tubulação de sucção	Variável	Ds	m
Comprimento da tubulação de sucção	Variável	Ls	m
Constante gravitacional	Constante	g	m/s^2
Massa específica da água	Constante	rho	kg/m^3
Viscosidade dinâmica da água a 20 °C	Constante	mu	$Pa.s$
Altura final da descarga	Constante	z_f_desc	m
Rugosidade da tubulação de descarga	Constante	e_desc	mm
Altura inicial da sucção	Constante	z_0_suc	m
Rugosidade da tubulação de sucção	Constante	e_suc	mm

Fonte: Elaborado pelo autor.

Os valores de pressão referem-se à pressão manométrica e não à absoluta. Assim, a pressão de 0 kPa corresponde a uma pressão absoluta de 101,325 kPa (ou 1 atm) (v. seção 2.1).

Tabela 4 – Valores das constantes.

Constante	Nome	Valor	Unidade
Constante gravitacional	g	9,81	m/s^2
Massa específica da água	rho	1000	kg/m^3
Viscosidade dinâmica da água a 20 °C	mu	0,001	$Pa.s$
Altura final da descarga	z_f_desc	30	m
Rugosidade da tubulação de descarga	e_desc	0,1	mm
Altura inicial da sucção	z_0_suc	0	m
Rugosidade da tubulação de sucção	e_suc	0,1	mm

Fonte: Elaborado pelo autor.

Existem ainda os coeficientes de perda local K das válvulas, cujos valores obtidos em White (2010, p. 397) são reproduzidos na Tabela 6.

Tabela 5 – Faixas de valores das variáveis.

Variável	Nome	Faixa	Unidade
Vazão volumétrica	Q_vazao	0 - 100	m^3/h
Pressão no final da descarga	pfd	0 - 101,325	kPa
Diâmetro da tubulação de descarga	Dd	0,1 - 0,3	m
Comprimento da tubulação de descarga	Ld	30 - 60	m
Pressão no início da sucção	p0s	0 - 101,325	kPa
Diâmetro da tubulação de sucção	Ds	0,1 - 0,3	m
Comprimento da tubulação de sucção	Ls	0 - 60	m

Fonte: Elaborado pelo autor.

Tabela 6 – Valores de K para as válvulas globo e gaveta em função do diâmetro. As válvulas são flangeadas.

Diâmetro (mm)	25	50	100	200	500
Globo	13,00	8,50	6,00	5,80	5,50
Gaveta	0,80	0,35	0,16	0,07	0,03

Fonte: Adaptado de White (2010, p. 397).

Por fim, se define o *número de Reynolds crítico* como $Re_{cr} = 2300$.

4.3 Delimitações adicionais

O presente trabalho considerará o escoamento interno, totalmente desenvolvido e em regime estacionário.

Serão considerados apenas valores de comprimento das tubulações (tanto de descarga quanto de sucção) que sejam maiores ou iguais a *dez vezes* os seus respectivos diâmetros. Essa delimitação é necessária já que, segundo Çengel e Cimbala (2012, p. 282), os *efeitos da entrada* tornam-se insignificantes somente para valores de comprimento de tubo maiores do que dez diâmetros (Equação 4.1):

$$L \geq 10D \quad (4.1)$$

O fator de correção da energia cinética α será desconsiderado no presente trabalho pois segundo Çengel e Cimbala (2012, p. 182), “(1) a maioria dos escoamentos encontrados na prática são turbulentos, para os quais o fator de correção é próximo da unidade e (2) os termos da energia cinética quase sempre são pequenos com relação aos outros termos da equação da energia, e a sua multiplicação por um fator menor que 2,0³ não faz muita diferença.”

4.4 Conceitos e definições

4.4.1 Conjunto de parâmetros

Um *conjunto de parâmetros* é um vetor de valores das variáveis da Tabela 5 (v. p. 82), excluía a vazão volumétrica⁴, Q_vazao . Desta forma, tem-se que um conjunto de parâmetros C_p é definido pela Equação 4.2:

$$C_p = \{pfd, Dd, Ld, p0s, Ds, Ls\} \quad (4.2)$$

em que cada elemento do conjunto é descrito na Tabela 5.

Como exemplo de um conjunto de parâmetros específico, tem-se:

$$C_p = \{20,265; 0,14; 51; 81,060; 0,12; 36\}$$

³ Para escoamentos laminares o fator de correção vale 2,0; já para os turbulentos, varia entre 1,04 e 1,11.

⁴ A vazão volumétrica não é considerada no conjunto de parâmetros pois ela é a variável que será empregada na comparação entre o modelo fenomenológico e a RNA. Ou seja, para um conjunto de parâmetros específico, serão calculados valores de *head*, um para cada valor de vazão.

4.4.2 Resolução, número de amostras e quantidade de conjuntos de parâmetros

Na fase de treinamento da rede neural artificial, a mesma precisa ser exposta a vários padrões de entrada p^5 , cada um destes consistindo de valores do conjunto de parâmetros C_p mais o valor da vazão volumétrica Q_vazao como explicitado na Equação 4.4:

$$p = Q_vazao \cup C_p \quad (4.3)$$

$$p = \{Q_vazao, pfd, Dd, Ld, p0s, Ds, Ls\} \quad (4.4)$$

Como cada elemento do padrão de entrada variará de acordo com as faixas de valores definidas na Tabela 5, é necessário que se estabeleça o *incremento* dos valores de cada um deles. Dessa forma, por exemplo, se os possíveis valores de vazões volumétricas forem $\{0, 20, 40, \dots, 100\}$ com variações de $20 \text{ m}^3/h$, pode-se ter também os possíveis valores de diâmetros como $\{0,100; 0,140; 0,180; \dots; 0,300\}$ com variações de $0,040 \text{ m}$, e assim por diante para as demais variáveis. O que todos os conjuntos de valores têm em comum é o que aqui será chamado de *resolução*, um conceito que será explicado nas próximas linhas.

Resolução

A resolução consiste na quantidade de divisões a que uma faixa de valores de uma dada variável é submetida. Estas divisões resultam em um número de possibilidades correspondente à quantidade de divisões *mais* um. P. ex.: a vazão volumétrica tem uma faixa de valores que vai de 0 à $100 \text{ m}^3/h$. Se for escolhida uma resolução de 5 para esta variável, a mesma apresentará 6 valores possíveis, como constatado a seguir:

$$Q_vazao = \{0; 20; 40; 60; 80; 100\}$$

O que se fez para obter o conjunto Q_vazao foi dividir a amplitude da faixa de valores pelo valor da resolução, e então usar este resultado como um incremento para obter os demais valores do conjunto Q_vazao , partindo-se do menor valor da faixa para esta variável, que é 0 (zero), nesse caso específico.

Pode-se entender a resolução como um valor auxiliar na obtenção da razão r de uma progressão aritmética. Deste modo, o cálculo do valor de r é necessário para se obter os demais valores do conjunto Q_vazao , como é dado na Equação 4.5:

$$r = \frac{Q_vazao_{m\acute{a}x} - Q_vazao_{m\acute{i}n}}{resolucao} \quad (4.5)$$

onde $Q_vazao_{m\acute{a}x}$ e $Q_vazao_{m\acute{i}n}$ são os extremos da faixa de valores para a vazão volumétrica dados na coluna *Faixa* da Tabela 5.

⁵ Os padrões de entrada p foram apresentados na seção 3.6

Obtido o valor da razão aritmética, r , pode-se calcular então o valor cada elemento do conjunto Q_vazao de acordo com a Equação 4.6:

$$Q_vazao_n = Q_vazao_{min} + (n - 1) \cdot r \quad (4.6)$$

onde n corresponde à posição do elemento dentro da progressão aritmética.

Cabe observar que esse método aritmético aqui exposto serve apenas para ajudar na compreensão do leitor. O importante é o entendimento de que cada variável apresentará valores igualmente espaçados, indo do mínimo ao máximo nas suas respectivas faixas definidas na Tabela 5. Mais adiante será definido o algoritmo de obtenção dos padrões de entrada (v. Algoritmo 4.1 na p. 87) em que usam-se *loops* aninhados, o que dispensa o cálculo dos valores de variáveis usando a Equação 4.6. Os *passos* desses *loops* serão calculados de maneira análoga à exposta na Equação 4.5.

Número de amostras

O número de amostras⁶ será dado pelo produto do número de possibilidades de cada uma das variáveis, correspondendo, portanto, ao número de combinações possíveis dos sete parâmetros. Como cada parâmetro pode apresentar, potencialmente, um número de valores igual ao valor da variável *resolucao mais um*, então tem-se que o tamanho da amostra será dado pela Equação 4.7 como:

$$tamAmostra = (resolucao + 1)^7 \quad (4.7)$$

Portanto, para uma resolução de 5, calcula-se um número de amostras como na Equação 4.8:

$$tamAmostra = 6^7 = 279.936 \quad (4.8)$$

onde $tamAmostra$ corresponde à quantidade de possíveis combinações das 7 variáveis, e também à quantidade de padrões de entrada p . Recorrendo à notação estabelecida na Equação 3.8, tem-se o resultado fornecido pela Equação 4.9:

$$|P| = tamAmostra = 6^7 = 279.936 \quad (4.9)$$

A quantidade de conjuntos de parâmetros ($qtdConjs$), o tamanho da amostra ($tamAmostra$), e a resolução ($resolucao$) estão relacionadas pela Equação 4.10:

$$qtdConjs = \frac{tamAmostra}{resolucao + 1} \quad (4.10)$$

⁶ Isto é, a quantidade de conjuntos p de padrões de entrada.

Para uma resolução de 5, e um tamanho de amostra de 279.936, haverão então:

$$qtdConjs = \frac{279.936}{5 + 1} = 46.656 \quad (4.11)$$

conjuntos de parâmetros.

Da combinação da Equação 4.7 com a Equação 4.10, deduz-se uma relação mais direta para o cálculo da quantidade de conjuntos, *qtdConjs*, que fica em função apenas da resolução, como demonstrado na Equação 4.12:

$$qtdConjs = \frac{tamAmostra}{resolucao + 1} = \frac{(resolucao + 1)^7}{resolucao + 1} = (resolucao + 1)^6 \quad (4.12)$$

4.5 Fase de treinamento da RNA

A presente seção se encarregará de mostrar os algoritmos para a obtenção dos padrões de entrada e dos alvos que serão usados em seguida no treinamento da rede neural e em sua avaliação. O ambiente usado para a implementação desses algoritmos é o programa *MATLAB*.

4.5.1 Obtenção dos padrões de entrada *p*

Para o armazenamento dos padrões de entrada, é criada no *MATLAB* uma matriz que aqui será chamada de *matrizEntradas*. Essa matriz consiste em 7 linhas, cada uma referente a uma das variáveis citadas na Tabela 5 (v. p. 82), e em um número de colunas igual ao tamanho da amostra. É importante observar que devido à restrição do comprimento das tubulações, que deve ser maior ou igual a dez vezes o seu diâmetro, a quantidade de amostras será um pouco *menor* do que aquela que seria obtida diretamente pela Equação 4.7. A criação e o preenchimento da *matrizEntradas* se dará conforme o exposto no Algoritmo 4.1.

O resultado do código mostrado na Algoritmo 4.1 é a *matrizEntradas* preenchida completamente, contando com 7 linhas e 233.280 colunas. O número de colunas resultante é menor do que aquele obtido pela Equação 4.8, já que não foram armazenadas na *matrizEntradas* amostras que contivessem comprimentos de tubulação (de descarga e/ou sucção) menores que dez vezes o diâmetro⁷. Esta exclusão foi causada pela condicional *if* que inicia na linha 17 do Algoritmo 4.1. O índice da coluna da *matrizEntradas*, dado por *j*, sofre um incremento de 1 se, e somente se, a condição de restrição do comprimento for atendida. Caso essa condição *não* seja satisfeita, *j* permanece com o mesmo valor *até* que a condição seja válida. Quando, então, a condição de restrição dada pelo *if* é válida,

⁷ A restrição citada encontra-se definida na seção 4.3

se dão as atribuições de valores à *matrizEntradas* mostradas nas linhas 18 à 24, e j sofre um incremento de 1. Os valores da *matrizEntradas* na presente fase de treinamento são gerados usando-se uma resolução igual a 5 (v. linha 3 do Algoritmo 4.1).

Algoritmo 4.1 – *Loops* aninhados responsáveis por preencher a *matrizEntradas* da fase de treinamento da RNA.

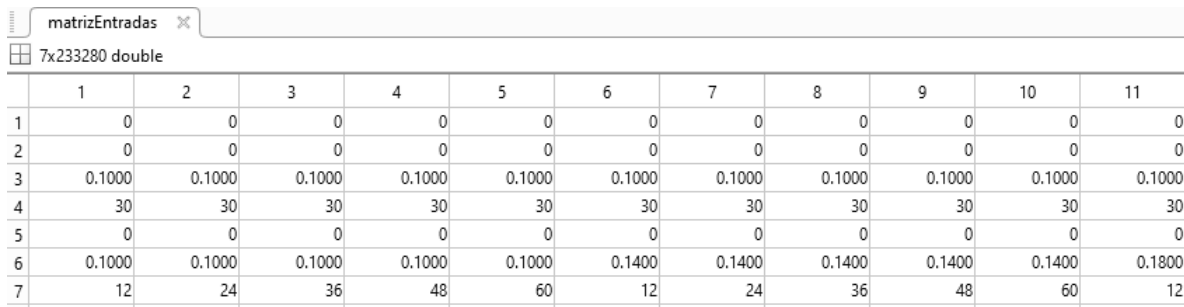
```

1 % resolucao é a resolução, isto é, o número de divisões a que o
2 % intervalo/range de uma determinada variável será submetido.
3 resolucao = 5;
4
5 % Criação da matrizEntradas, contando com 7 linhas e um número
6 % indefinido de colunas.
7 matrizEntradas = double.empty(7,0);
8
9 j = 1; % "j" é o índice da coluna da matrizEntradas.
10 for Q_vazao = 0 : 100/resolucao : 100
11     for pfd = 0 : 101.325/resolucao : 101.325
12         for Dd = 0.100 : 0.200/resolucao : 0.300
13             for Ld = 30 : 30/resolucao : 60
14                 for p0s = 0 : 101.325/resolucao : 101.325
15                     for Ds = 0.100 : 0.200/resolucao : 0.300
16                         for Ls = 0 : 60/resolucao : 60
17                             if (Ld >= Dd*10) == true && (Ls >= Ds*10) == true
18                                 matrizEntradas(1,j) = Q_vazao;
19                                 matrizEntradas(2,j) = pfd;
20                                 matrizEntradas(3,j) = Dd;
21                                 matrizEntradas(4,j) = Ld;
22                                 matrizEntradas(5,j) = p0s;
23                                 matrizEntradas(6,j) = Ds;
24                                 matrizEntradas(7,j) = Ls;
25                                 j = j + 1;
26                             end
27                         end
28                     end
29                 end
30             end
31         end
32     end
33 end
34
35 % tamAmostra guarda o número de colunas da matrizEntradas.
36 tamAmostra = size(matrizEntradas,2);
37
38 % Embaralhamento da matrizEntradas ao longo das colunas.
39 matrizEntradas = matrizEntradas(:,randperm(end));

```

Um aspecto da *matrizEntradas* refere-se à sua ordem de preenchimento: na Figura 29), os valores da última linha (sétima linha, referente ao comprimento da tubulação de sucção, L_s) sofrem incrementos enquanto os valores nas demais linhas permanecem constantes até a quinta coluna. Somente após o valor da sétima linha atingir o seu máximo (60), é que a linha anterior (a sexta, referente ao diâmetro da tubulação de sucção, D_s) sofre um incremento, passando de 0.1000 para 0.1400, dando-se início a um novo ciclo de preenchimento a partir de sucessivos incrementos dos valores da última linha. Esse processo ocorre da mesma forma para todas as variáveis, até o ponto em que todas apresentam os seus respectivos valores máximos, expressos na última coluna da *matrizEntradas* (v. Figura 30). Esse processo de preenchimento da *maztrizEntradas* garante que haja a maior quantidade possível de padrões de entrada, já que todas as possíveis combinações das sete variáveis estarão presentes, excetuando-se aquelas combinações em que os comprimentos das tubulações (de sucção e/ou de descarga) sejam menores que dez vezes os seus respectivos diâmetros.

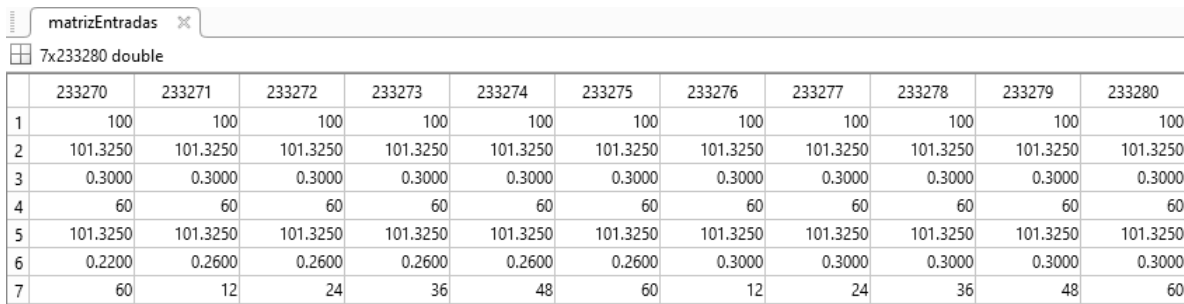
Figura 29 – Primeiras colunas da *matrizEntradas* usada na fase de treinamento da *RNA*.



	1	2	3	4	5	6	7	8	9	10	11
1	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0
3	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000
4	30	30	30	30	30	30	30	30	30	30	30
5	0	0	0	0	0	0	0	0	0	0	0
6	0.1000	0.1000	0.1000	0.1000	0.1000	0.1400	0.1400	0.1400	0.1400	0.1400	0.1800
7	12	24	36	48	60	12	24	36	48	60	12

Fonte: Elaborado pelo autor.

Figura 30 – Últimas colunas da *matrizEntradas* usada na fase de treinamento da *RNA*.



	233270	233271	233272	233273	233274	233275	233276	233277	233278	233279	233280
1	100	100	100	100	100	100	100	100	100	100	100
2	101.3250	101.3250	101.3250	101.3250	101.3250	101.3250	101.3250	101.3250	101.3250	101.3250	101.3250
3	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000	0.3000
4	60	60	60	60	60	60	60	60	60	60	60
5	101.3250	101.3250	101.3250	101.3250	101.3250	101.3250	101.3250	101.3250	101.3250	101.3250	101.3250
6	0.2200	0.2600	0.2600	0.2600	0.2600	0.2600	0.3000	0.3000	0.3000	0.3000	0.3000
7	60	12	24	36	48	60	12	24	36	48	60

Fonte: Elaborado pelo autor.

Na linha 36 do Algoritmo 4.1, é atribuído a uma variável chamada *tamAmostra* o número de colunas resultante da *matrizEntradas* através da função *size*, que é nativa do *MATLAB*. Sendo assim, *tamAmostra* passa a ter como valor armazenado o número de amostras a que a rede neural artificial será submetida mais adiante.

Após o preenchimento da *matrizEntradas*, a mesma é embaralhada ao longo de suas colunas, de modo que apenas as colunas troquem de posição entre si dentro da matriz, de maneira aleatória. Esse embaralhamento é conseguido através da linha de código 39 do Algoritmo 4.1 por meio da função *randperm*, que é nativa do *MATLAB*. Uma parte do resultado do embaralhamento da *matrizEntradas* pode ser vista na Figura 31.

Figura 31 – A *matrizEntradas* pós-embaralhamento.

matrizEntradas											
7x233280 double											
	1	2	3	4	5	6	7	8	9	10	11
1	100	60	20	100	80	60	20	20	80	0	0
2	101.3250	101.3250	20.2650	60.7950	20.2650	60.7950	81.0600	81.0600	81.0600	40.5300	101.3250
3	0.2200	0.1800	0.3000	0.3000	0.2200	0.1800	0.1400	0.3000	0.2200	0.2200	0.2200
4	42	60	36	60	36	36	54	42	54	36	30
5	60.7950	20.2650	101.3250	101.3250	60.7950	60.7950	0	0	60.7950	81.0600	101.3250
6	0.1000	0.2200	0.2200	0.2200	0.2200	0.2600	0.1800	0.1400	0.1800	0.1000	0.1800
7	12	36	60	12	48	36	12	12	24	36	60

Fonte: Elaborado pelo autor.

4.5.2 Obtenção dos alvos t

O armazenamento dos valores alvo será feito numa nova matriz, a *matrizAlvos* (v. Algoritmo 4.2). O valor de uma dada célula da *matrizAlvos* corresponde ao *head* do sistema calculado em função dos valores presentes em uma dada coluna da *matrizEntradas*. Esse cálculo é realizado pela função *fn_head_sistema* que recebe como parâmetros de entrada os valores presentes em uma coluna j da *matrizEntradas*. A *matrizAlvos* possuirá apenas uma linha e o seu número de colunas coincidirá, ao final do processo, com o número de colunas da *matrizEntradas*. Alguns valores da *matrizAlvos* podem ser vistos na Figura 32.

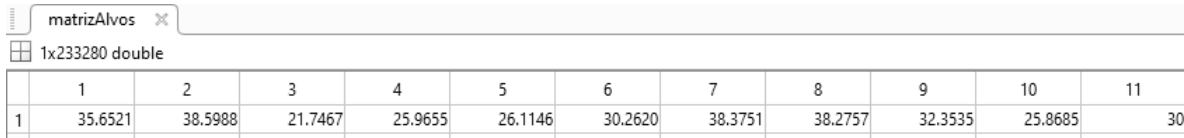
Algoritmo 4.2 – Código de preenchimento da *matrizAlvos*.

```

1 % Criação da matrizAlvos, com 1 linha e um número indefinido de colunas.
2 matrizAlvos = double.empty(1,0);
3 % O loop abaixo é usado para calcular o head usando-se cada uma das colunas
4 % da matrizEntradas como parâmetro de entrada da função fn_head_sistema.
5 for j = 1:tamAmostra
6     matrizAlvos(1,j) = fn_head_sistema(matrizEntradas(:,j));
7 end

```

A fim de manter a integridade visual deste capítulo, a implementação da função *fn_head_sistema* e de todas as suas funções auxiliares serão demonstradas no Anexo A.

Figura 32 – Colunas iniciais da *matrizAlvos* usada na fase de treinamento da *RNA*.


	1	2	3	4	5	6	7	8	9	10	11
1	35.6521	38.5988	21.7467	25.9655	26.1146	30.2620	38.3751	38.2757	32.3535	25.8685	30

Fonte: Elaborado pelo autor.

4.5.3 Normalização dos padrões de entrada e dos alvos

Para que se treine a rede neural artificial, é necessário que antes se *normalizem* os valores de entrada (presentes na *matrizEntradas*) e os alvos (os *heads* armazenados na *matrizAlvos*). Para este fim, recorre-se à função *mapminmax*, que é nativa do *MATLAB*, como visto no Algoritmo 4.3.

O que a função *mapminmax* faz é *associar* (ou *mapear*) os valores mínimos e máximos de uma linha da matriz a -1 e 1 , respectivamente. Ou seja, *mapminmax* converte o menor valor de uma linha da matriz em -1 , e o maior em 1 . O mapeamento também ocorrerá com os valores intermediários, mas estes serão associados a valores entre, mas não iguais a, -1 e 1 . Os resultados dos dois processos de normalização ficarão armazenados em duas novas matrizes: a *matrizEntradasNorm* armazenará os resultados da normalização da *matrizEntradas*, e a *matrizAlvosNorm* receberá os valores da *matrizAlvos* normalizados.

Algoritmo 4.3 – Código de normalização da *matrizEntradas* e da *matrizAlvos*.

```

1 % matrizEntradasNorm consistirá numa matriz que possuirá os mesmos valores
2 % que a matrizEntradas, porém, mapeados de -1 à 1, de acordo com a função
3 % mapminmax.
4 matrizEntradasNorm = mapminmax(matrizEntradas);
5 % matrizAlvosNorm é um vetor que armazenará os valores da matrizAlvos
6 % mapeados de -1 à 1, conforme a função mapminmax. ts refere-se a
7 % parâmetros usados durante o processo de mapeamento/normalização.
8 [matrizAlvosNorm, ts] = mapminmax(matrizAlvos);

```

Na linha 8 do Algoritmo 4.3 há uma variável chamada *ts*. Essa variável é um tipo especial de estrutura de dados do *MATLAB* chamada *struct*, sendo responsável por armazenar valores resultantes do mapeamento realizado pela função *mapminmax*. Os dados contidos na *struct ts* serão usados mais adiante para o processo inverso do mapeamento, isto é, para obter os valores de *head* a partir de valores fornecidos pela rede neural, estes últimos variando de -1 a 1 . Em outras palavras, *ts* terá papel no processo de *desnormalização* dos resultados fornecidos pela rede neural artificial.

Dos campos e valores presentes na *struct ts*, pode-se observar que o menor e o maior

valor da *matrizAlvos* são, respectivamente, 19,6713 e 60,5899⁸ (v. Figura 33). Esses dois valores passaram a corresponder, após o processo de normalização, a -1 e 1 , e portanto, espera-se que os valores intermediários da *matrizAlvos* sejam mapeados entre -1 e 1 . Por exemplo: se na *matrizAlvos* houver um valor igual a 40,1306 espera-se que o mesmo seja mapeado para 0 (zero) na *matrizAlvosNorm*, já que tanto 40,1306 quanto 0 correspondem exatamente às *medianas*⁹ de suas respectivas faixas. O que foi exposto no nas últimas linhas pode ser expresso da seguinte forma:

$$\begin{aligned} x_{min} + x_{range} \cdot \frac{1}{2} &= 19,6713 + 40,9187 \cdot \frac{1}{2} = 40,1306 \\ x_{max} - x_{range} \cdot \frac{1}{2} &= 60,5899 - 40,9187 \cdot \frac{1}{2} = 40,1306 \end{aligned}$$

Parte da *matrizEntradasNorm* é apresentada na Figura 34 e parte da *matrizAlvosNorm*, na Figura 35.

A comparação entre a *matrizEntradas* e a *matrizEntradasNorm* fornece sólido indício de que o processo de normalização ocorreu da maneira correta. Por exemplo: a célula (1,1) da *matrizEntradas* possui um valor de 100 (v. Figura 31), que é o maior valor possível para a linha 1 desta matriz¹⁰. Desta forma, espera-se que a mesma célula na *matrizEntradasNorm* apresente um valor de 1, já que após a normalização este é o maior valor de uma dada linha. De fato, quando se analisa a *matrizEntradasNorm*, o valor da célula (1,1) é exatamente 1 (v. Figura 34).

Figura 33 – A *struct ts* armazena valores resultantes da normalização.

Field	Value
name	'mapminmax'
xrows	1
xmax	60.5899
xmin	19.6713
xrange	40.9187
yrows	1
ymax	1
ymin	-1
yrange	2
gain	0.0489
xoffset	19.6713
no_change	0

Fonte: Elaborado pelo autor.

⁸ Esses valores estão contidos nas variáveis *xmin* e *xmax* da *struct ts*.

⁹ Em Estatística, a mediana é o valor que, grosso modo, encontra-se *no meio* de um intervalo.

¹⁰ A linha 1 corresponde aos valores de vazão volumétrica.

Figura 34 – Colunas iniciais da *matrizEntradasNorm* usada na fase de treinamento da *RNA*.

matrizEntradasNorm											
7x233280 double											
	1	2	3	4	5	6	7	8	9	10	11
1	1	0.2000	-0.6000	1	0.6000	0.2000	-0.6000	-0.6000	0.6000	-1	-1
2	1	1	-0.6000	0.2000	-0.6000	0.2000	0.6000	0.6000	0.6000	-0.2000	1
3	0.2000	-0.2000	1	1	0.2000	-0.2000	-0.6000	1	0.2000	0.2000	0.2000
4	-0.2000	1	-0.6000	1	-0.6000	-0.6000	0.6000	-0.2000	0.6000	-0.6000	-1
5	0.2000	-0.6000	1	1	0.2000	0.2000	-1	-1	0.2000	0.6000	1
6	-1	0.2000	0.2000	0.2000	0.2000	0.6000	-0.2000	-0.6000	-0.2000	-1	-0.2000
7	-1	0	1	-1	0.5000	0	-1	-1	-0.5000	0	1

Fonte: Elaborado pelo autor.

Figura 35 – Colunas iniciais da *matrizAlvosNorm* usada na fase de treinamento da *RNA*.

matrizAlvosNorm											
1x233280 double											
	1	2	3	4	5	6	7	8	9	10	11
1	-0.2189	-0.0749	-0.8986	-0.6924	-0.6851	-0.4823	-0.0858	-0.0907	-0.3801	-0.6971	-0.4952

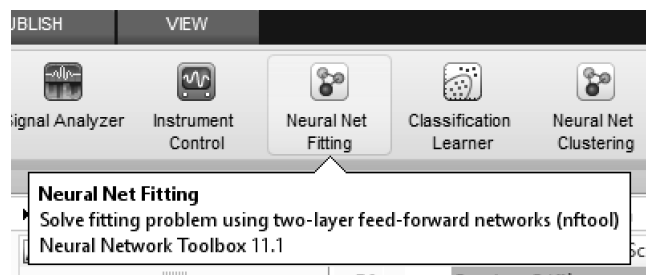
Fonte: Elaborado pelo autor.

4.5.4 Treinamento da rede neural artificial

De posse dos valores de entrada e alvos normalizados nas formas da *matrizEntradasNorm* e *matrizAlvosNorm* (v. subseção 4.5.3), pode-se dar início ao treinamento da rede neural artificial.

O *MATLAB* vem com um módulo de redes neurais que possui uma ferramenta chamada *Neural Net Fitting*. O botão que dá acesso a esta ferramenta é mostrado na Figura 36.

Figura 36 – Botão para a ferramenta *Neural Net Fitting* (*nftool*) no *MATLAB*.



Fonte: Elaborado pelo autor.

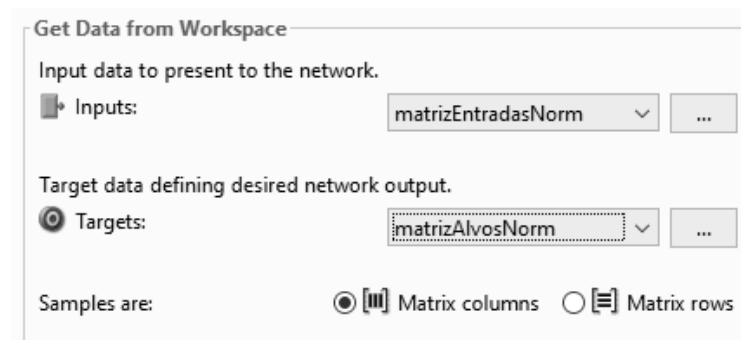
Ao abrir a ferramenta, uma janela de boas-vindas aparece, onde uma breve explicação é fornecida. Essa explicação afirma que uma rede do tipo *feedforward*¹¹ de duas

¹¹ Numa rede de tipo feedforward o fluxo de dados se dá numa única direção: da camada de entrada para a saída, sem recorrência.

camadas será criada, com a função de ativação da camada oculta sendo do tipo sigmóide e a função da camada de saída sendo linear. Além disto, o programa também informa que a rede neural será treinada com o algoritmo de retropropagação Levenberg-Marquardt. Esse algoritmo de treinamento é usado por causa de sua flexibilidade e performance (v. subseção 3.11.2) em comparação aos demais.

Na próxima janela pede-se que se selecionem os dados de entrada no campo *Inputs* e os alvos no campo *Targets* (v. Figura 37). Nessa etapa se seleciona a *matrizEntradasNorm* para o campo *Inputs* e a *matrizAlvosNorm* para o campo *Targets*. A opção *Samples are* deve ser marcada como *Matrix columns*, já que as amostras¹² estão dispostas em colunas, isto é, cada coluna corresponde a uma amostra.

Figura 37 – Seleção dos padrões de entrada (*Inputs*) e dos alvos (*Targets*) na ferramenta *Neural Net Fitting*.



Fonte: Elaborado pelo autor.

Em seguida é solicitado que se escolha o tamanho relativo (em porcentagem) de cada um dos três conjuntos menores que dividirão o conjunto total de amostras: *training*, *validation* e *testing*¹³ (v. Figura 38). O conjunto *training* consiste nos dados de entrada que serão apresentados à rede durante a etapa de treinamento; o conjunto *validation* serve para medir a *generalização* da rede, encerrando o treinamento assim que a capacidade de generalização para de avançar; já o conjunto *testing* não tem efeito sobre o treinamento da rede neural, servindo somente para prover uma medida independente da performance durante o treinamento. A divisão do conjunto total de amostras é então escolhida da seguinte forma¹⁴: 70% para o conjunto *training*, 15% para o conjunto *validation* e 15% para o conjunto *testing*. Dessa forma, a rede neural em si será apresentada à 163.296 amostras, número que corresponde à 70% de 233.280.

A próxima janela (v. Figura 39) solicita ao usuário que seja definido o número de neurônios artificiais da camada oculta. Segundo Zurada (1992, p. 216), a determinação do

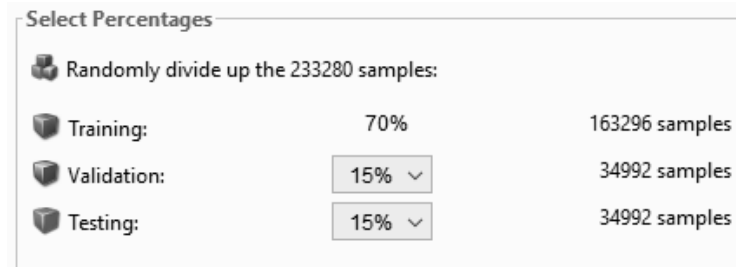
¹² Em inglês, *amostras* corresponde a *samples*.

¹³ *Training*, *validation* e *testing* correspondem, respectivamente, a *treinamento*, *validação* e *teste*, em português.

¹⁴ Essa forma de dividir é a padrão do *MATLAB*.

número de neurônios na camada oculta constitui um estudo complexo. Como afirmado por Swingler (1996), nunca será necessário mais do que o dobro de unidades ocultas do que as unidades de entrada. Por padrão, o *MATLAB* define o número de dez neurônios, o que será mantido desta forma já que este valor não excede o dobro de entradas¹⁵.

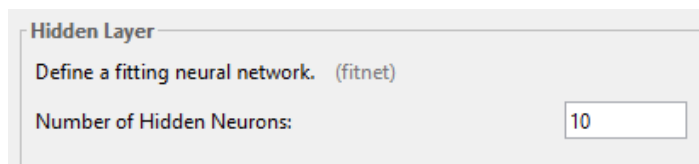
Figura 38 – Divisão dos dados de amostra em três grupos.



Select Percentages		
Randomly divide up the 233280 samples:		
Training:	70%	163296 samples
Validation:	15%	34992 samples
Testing:	15%	34992 samples

Fonte: Elaborado pelo autor.

Figura 39 – Escolha do número de neurônios na camada oculta.



Hidden Layer

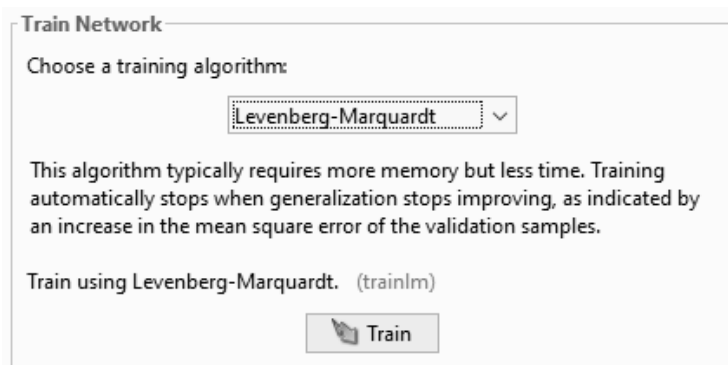
Define a fitting neural network. (fitnet)

Number of Hidden Neurons: 10

Fonte: Elaborado pelo autor.

A seguir, o algoritmo de treinamento (v. Figura 40) é escolhido como o *Levenberg-Marquardt* (por padrão). Ao clicar no botão *Train*¹⁶ o treinamento da rede neural artificial tem início.

Figura 40 – Escolha do algoritmo Levenberg-Marquardt para o treinamento da rede neural artificial.



Train Network

Choose a training algorithm:

Levenberg-Marquardt

This algorithm typically requires more memory but less time. Training automatically stops when generalization stops improving, as indicated by an increase in the mean square error of the validation samples.

Train using Levenberg-Marquardt. (trainlm)

Train

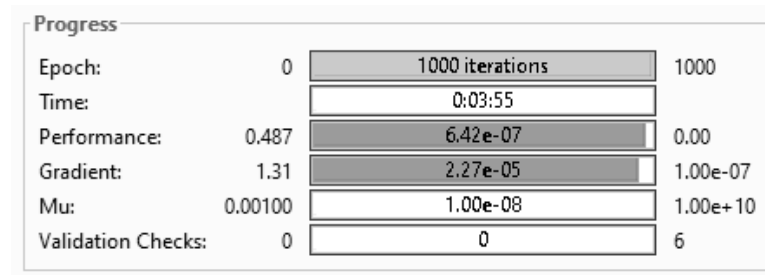
Fonte: Elaborado pelo autor.

¹⁵ Este trabalho considera sete parâmetros de entradas.

¹⁶ *Train* em português significa *treinar*.

Quanto maior for a quantidade de amostras, maior será o tempo necessário para o treinamento da rede neural. Na Figura 41 pode-se ver que o tempo de treino para mil iterações (ou *épocas*) através das 163.296 amostras foi relativamente curto, durando apenas 3 minutos e 55 segundos.

Figura 41 – Monitoramento do treinamento da rede neural artificial.



Fonte: Elaborado pelo autor.

Findado o treinamento da rede neural, o *MATLAB* gera a função *RNA_head_sistema*, sendo esta a responsável por implementar a rede em forma de uma função capaz de receber como entrada um vetor de sete elementos e de fornecer como saída um resultado. A função *RNA_head_sistema* possui em si variáveis que armazenam os vieses¹⁷ de cada um dos neurônios das duas camadas (camada oculta e camada de saída), bem como os pesos de todas as conexões da rede. Ao todo, a rede neural conta com 11 vieses (10 para a camada oculta e 1 pra camada de saída) e 80 conexões (70 conexões entre os 7 neurônios de entrada e os 10 neurônios da camada oculta, e 10 conexões entre os 10 neurônios da camada oculta e o único neurônio da camada de saída).

A Figura 42 mostra os pesos das conexões entre a camada de entrada e a camada oculta, da seguinte forma: o valor da célula (i, j) ¹⁸ corresponde ao peso da conexão que vai do neurônio i da camada de entrada ao neurônio j da camada oculta. A Figura 43 mostra os vieses da camada oculta: cada célula corresponde ao viés de cada um dos dez neurônios desta camada. A Figura 44 mostra os pesos das conexões que vão dos neurônios da camada oculta (representados pelas colunas) ao único neurônio da camada de saída (representado pela linha única). O viés do único neurônio da camada de saída é igual a 2,4699.

¹⁷ *Biases*, em inglês.

¹⁸ Sendo i a linha e j a coluna.

Figura 42 – Pesos das conexões dos neurônios i da camada de entrada aos neurônios j da camada oculta.

IW1_1										
7x10 double										
	1	2	3	4	5	6	7	8	9	10
1	-1.0079	-0.5553	-0.6548	-1.7370	-0.2081	-0.0273	-0.9396	-0.4765	-0.5893	0.8072
2	-1.6075e-04	-1.2118e-05	-5.8891e-04	-0.0063	0.0020	0.0153	0.0058	2.9554e-05	-7.1263e-04	1.0699e-04
3	-8.5609e-04	2.1905	-0.0023	0.0069	-0.1239	-0.0166	-0.7318	2.2745	-0.0029	-2.3679
4	-0.0017	-0.0293	-0.0031	-0.0302	-0.0040	-5.0465e-04	-0.0151	-0.0356	-0.0038	-0.0113
5	2.0707e-04	1.5521e-04	7.1074e-04	0.0066	-0.0021	-0.0154	-0.0061	1.2922e-04	8.6533e-04	-3.3006e-04
6	3.0866	4.5568e-04	3.3740	2.1707	0.0020	2.1751e-04	-0.0038	-8.6042e-05	3.4774	-0.0017
7	0.2168	-2.8639e-04	-0.2274	-0.6727	-0.0020	-2.2624e-04	0.0012	2.2173e-04	-0.2441	0.0013

Fonte: Elaborado pelo autor.

Figura 43 – Vieses b dos neurônios da camada oculta.

b1										
1x10 double										
	1	2	3	4	5	6	7	8	9	10
1	4.6021	3.2736	4.5248	2.2350	0.1945	0.0091	-1.1734	2.9726	4.5657	-3.7312

Fonte: Elaborado pelo autor.

Figura 44 – Pesos das conexões dos neurônios j da camada oculta ao neurônio único da camada de saída.

LW2_1										
1x10 double										
	1	2	3	4	5	6	7	8	9	10
1	0.3364	-7.6986	-2.8662	-0.0077	-2.2031	16.7454	-0.0709	2.8932	2.1817	-2.4106

Fonte: Elaborado pelo autor.

4.6 Fase de avaliação da RNA

Durante a fase de avaliação da RNA se testará a capacidade de generalização da mesma. Para tanto, é necessário expor a rede neural a amostras que não foram usadas durante a fase de treinamento. Os resultados fornecidos pela RNA em função dessas novas amostras serão então comparados àqueles fornecidos pela expressão fenomenológica¹⁹ para as mesmas amostras.

4.6.1 Obtenção de novas amostras

Como a capacidade de generalização da RNA deve ser avaliada, é necessário que se obtenham valores amostrais em relação aos quais a rede não foi submetida durante a fase de treinamento. Isto será feito de forma muito semelhante à da fase de treinamento,

¹⁹ Dada pela função $fn_head_sistema$.

porém, com uma diferença fundamental: a resolução aqui usada terá o valor de 10 (v. Algoritmo 4.4, linha 3), resultando em um número amostras *substancialmente maior* do que o da fase de treinamento, na qual a resolução utilizada foi de 5.

Algoritmo 4.4 – *Loops* aninhados responsáveis por preencher a *matrizEntradas* da fase de avaliação da *RNA*.

```

1 % resolucao é a resolução, isto é, o número de divisões a que o
2 % intervalo/range de uma determinada variável será submetido.
3 resolucao = 10;
4
5 % Criação da matrizEntradas com 7 linhas e um número indefinido de colunas.
6 matrizEntradas = double.empty(7,0);
7
8 j = 1; % "j" é o índice da coluna da matrizEntradas.
9 for Q_vazao = 0 : 100/resolucao : 100
10     for pfd = 0 : 101.325/resolucao : 101.325
11         for Dd = 0.100 : 0.200/resolucao : 0.300
12             for Ld = 30 : 30/resolucao : 60
13                 for p0s = 0 : 101.325/resolucao : 101.325
14                     for Ds = 0.100 : 0.200/resolucao : 0.300
15                         for Ls = 0 : 60/resolucao : 60
16                             if (Ld >= Dd*10) == true && (Ls >= Ds*10) == true
17                                 matrizEntradas(1,j) = Q_vazao;
18                                 matrizEntradas(2,j) = pfd;
19                                 matrizEntradas(3,j) = Dd;
20                                 matrizEntradas(4,j) = Ld;
21                                 matrizEntradas(5,j) = p0s;
22                                 matrizEntradas(6,j) = Ds;
23                                 matrizEntradas(7,j) = Ls;
24                                 j = j + 1;
25                             end
26                         end
27                     end
28                 end
29             end
30         end
31     end
32 end
33
34 % tamAmostra guarda o número de colunas da matrizEntradas.
35 tamAmostra = size(matrizEntradas,2);

```

De acordo com a Equação 4.7, o número de amostras *esperado* para um valor de

resolução igual a 10 é de 19.487.171, como exposto na Equação 4.13:

$$tamAmostra = (resolucao + 1)^7 = 11^7 = 19.487.171 \quad (4.13)$$

O preenchimento da nova matriz *matrizEntradas* através do Algoritmo 4.4 resulta em um número de amostras um pouco menor do que o esperado: 17.715.610²⁰. A comparação desse valor com aquele usado durante a fase de treinamento (163.296) mostra que o tamanho da amostra da fase de avaliação é cerca de 108,49 vezes maior; em outras palavras, o número de amostras da fase de treinamento corresponde a apenas 0,92% ao da fase de avaliação.

A seção inicial da *matrizEntradas* gerada pelo Algoritmo 4.4 pode ser vista na Figura 45, na qual se constata que os valores da última linha são incrementados de 6 em 6, e não mais de 12 em 12 como ocorria na fase de treinamento da *RNA*. Essa diferença é causada pelo uso de uma resolução *maior* na fase de avaliação (10) em relação à fase de treinamento (5). O mesmo ocorre para as demais variáveis presentes nas outras linhas.

Figura 45 – Seção inicial da *matrizEntradas* usada na fase de avaliação da *RNA*.

	1	2	3	4	5	6	7	8	9	10	11
1	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0
3	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000
4	30	30	30	30	30	30	30	30	30	30	30
5	0	0	0	0	0	0	0	0	0	0	0
6	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000	0.1000	0.1200
7	6	12	18	24	30	36	42	48	54	60	6

Fonte: Elaborado pelo autor.

Logo após a obtenção dos valores da matriz *matrizEntradas*, ocorre o embaralhamento de suas colunas (v. Figura 46) e a sua posterior normalização, como visto no Algoritmo 4.5. O resultado é a *matrizEntradasNorm*, que consiste de valores resultantes da normalização da *matrizEntradas*. A seção inicial da *matrizEntradasNorm* pode ser vista na Figura 47.

²⁰ O resultado final é menor que o esperado devido à restrição de que os comprimentos das tubulações devem ser maiores ou iguais que dez vezes os seus respectivos diâmetros.

Algoritmo 4.5 – Código de embaralhamento e normalização da *matrizEntradas* na fase de avaliação da *RNA*.

```

1 % Embaralhamento da matrizEntradas ao longo das colunas.
2 matrizEntradas = matrizEntradas(:,randperm(end));
3
4 % matrizEntradasNorm consistirá numa matriz que possuirá os mesmos valores
5 % que a matrizEntradas, porém, mapeados de -1 à 1, de acordo com a função
6 % mapminmax.
7 matrizEntradasNorm = mapminmax(matrizEntradas);

```

Figura 46 – Seção inicial da *matrizEntradas* embaralhada usada na fase de avaliação da *RNA*.

matrizEntradas											
7x17715610 double											
	1	2	3	4	5	6	7	8	9	10	11
1	100	60	100	10	60	30	0	0	0	80	60
2	10.1325	101.3250	10.1325	101.3250	101.3250	30.3975	60.7950	60.7950	0	10.1325	91.1925
3	0.2600	0.1400	0.2200	0.1400	0.2600	0.1400	0.2800	0.1400	0.1400	0.1800	0.2000
4	48	33	54	45	48	51	48	60	54	54	51
5	81.0600	101.3250	10.1325	0	101.3250	10.1325	10.1325	0	10.1325	40.5300	30.3975
6	0.2200	0.2200	0.1800	0.1200	0.3000	0.2400	0.3000	0.2000	0.1200	0.1000	0.2000
7	42	42	42	6	6	54	30	24	6	60	48

Fonte: Elaborado pelo autor.

Figura 47 – Seção inicial da *matrizEntradasNorm* usada na fase de avaliação da *RNA*.

matrizEntradasNorm											
7x17715610 double											
	1	2	3	4	5	6	7	8	9	10	11
1	1	0.2000	1	-0.8000	0.2000	-0.4000	-1	-1	-1	0.6000	0.2000
2	-0.8000	1	-0.8000	1	1	-0.4000	0.2000	0.2000	-1	-0.8000	0.8000
3	0.6000	-0.6000	0.2000	-0.6000	0.6000	-0.6000	0.8000	-0.6000	-0.6000	-0.2000	2.2204e-16
4	0.2000	-0.8000	0.6000	0	0.2000	0.4000	0.2000	1	0.6000	0.6000	0.4000
5	0.6000	1	-0.8000	-1	1	-0.8000	-0.8000	-1	-0.8000	-0.2000	-0.4000
6	0.2000	0.2000	-0.2000	-0.8000	1	0.4000	1	2.2204e-16	-0.8000	-1	2.2204e-16
7	0.3333	0.3333	0.3333	-1	-1	0.7778	-0.1111	-0.3333	-1	1	0.5556

Fonte: Elaborado pelo autor.

4.6.2 Cálculos dos valores de saída da expressão fenomenológica

A obtenção dos valores fornecidos pela expressão fenomenológica é necessária para fins de comparação com aqueles resultantes da rede neural artificial. Dessa forma, para os padrões de entradas contidos na *matrizEntradas*, se calcularão os valores de *head* através da função *fn_head_sistema*, processo este demonstrado no Algoritmo 4.6. Os resultados da expressão fenomenológica são armazenados na matriz *matrizHeads_fenom* para posterior utilização na fase de avaliação, sendo as suas colunas iniciais mostradas na Figura 48.

Algoritmo 4.6 – Código de cálculo das saídas da expressão fenomenológica.

```

1 % Criação da matriz que armazenará os valores de saída da expressão
2 % fenomenológica.
3 matrizHeads_fenom = double.empty(num_param,0);
4
5 % O loop abaixo é usado para calcular o head usando-se cada um dos
6 % parâmetros da matrizEntradas.
7 for j = 1:tamAmostra
8     matrizHeads_fenom(1,j) = fn_head_sistema(matrizEntradas(:,j));
9 end

```

Figura 48 – Colunas iniciais da *matrizHeads_fenom* usada na fase de avaliação da *RNA*.

	1	2	3	4	5	6	7	8	9	10
1	23.0011	30.7718	30.5592	40.3575	30.0548	32.3062	35.1644	36.1972	28.9671	32.2561

Fonte: Elaborado pelo autor.

4.6.3 Cálculos dos valores de saída da rede neural artificial

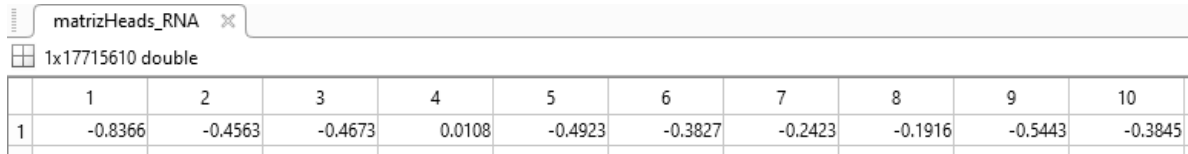
Com a função *RNA_head_sistema* pode-se obter os valores de *heads* do sistema fornecidos pela rede neural artificial. Uma matriz chamada *matrizHeads_RNA* (v. Algoritmo 4.7) será criada para armazenar estes valores de saída da rede neural, que se situarão na faixa de -1 a 1 . Para cada coluna da matriz *matrizEntradasNorm* obtida no Algoritmo 4.5, será calculada uma saída da função *RNA_head_sistema* cujo valor será armazenado na coluna correspondente da matriz *matrizHeads_RNA*. Os valores das primeiras colunas da *matrizHeads_RNA* podem ser vistos na Figura 49.

Algoritmo 4.7 – Código de cálculo das saídas da rede neural.

```

1 % Criação da matriz que vai armazenar os valores de saída da rede neural.
2 matrizHeads_RNA = double.empty(1,0);
3 % Preenchimento da matrizHeads_RNA.
4 for j = 1:tamAmostra
5     matrizHeads_RNA(1,j) = RNA_head_sistema(matrizEntradasNorm(:,j));
6 end
7
8 % A função mapminmax será aplicada de maneira reversa para transformar os
9 % valores de saída da rede neural (que vão de -1 a 1) em valores que se
10 % encontram entre os extremos dados pela struct ts.
11 matrizHeads_RNA_desnorm = mapminmax('reverse',matrizHeads_RNA,ts);

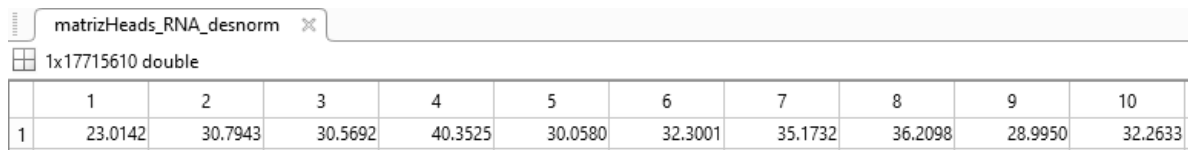
```

Figura 49 – Colunas iniciais da *matrizHeads_RNA*.


	1	2	3	4	5	6	7	8	9	10
1	-0.8366	-0.4563	-0.4673	0.0108	-0.4923	-0.3827	-0.2423	-0.1916	-0.5443	-0.3845

Fonte: Elaborado pelo autor.

Após a obtenção dos valores de *matrizHeads_RNA*, um processo inverso ao da normalização (aqui chamado de *desnormalização*) será conduzido a fim de relacionar os valores de saída da rede (de -1 a 1) aos seus valores correspondentes de *head*, o que irá resultar na *matrizHeads_RNA_desnorm* (v. linha 11 do Algoritmo 4.7). Essa correlação inversa será possibilitada pelo uso da *struct ts* que fora obtida no processo de normalização da *matrizAlvos* durante a fase de treinamento da *RNA* (v. Algoritmo 4.3 e Figura 33). Os valores das primeiras colunas da *matrizHeads_RNA_desnorm* podem ser vistos na Figura 50.

Figura 50 – Colunas iniciais da *matrizHeads_RNA_desnorm*.


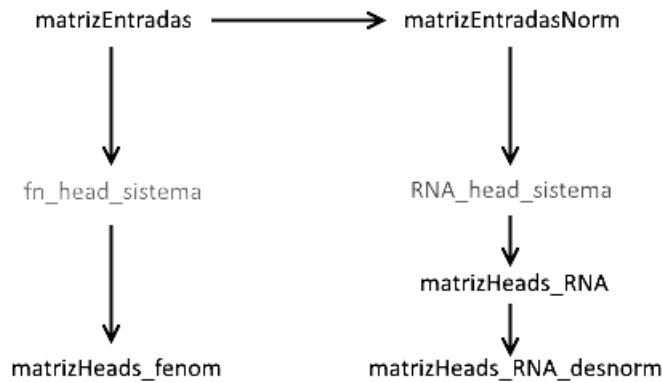
	1	2	3	4	5	6	7	8	9	10
1	23.0142	30.7943	30.5692	40.3525	30.0580	32.3001	35.1732	36.2098	28.9950	32.2633

Fonte: Elaborado pelo autor.

4.6.4 Comparações entre a rede neural artificial e a expressão fenomenológica

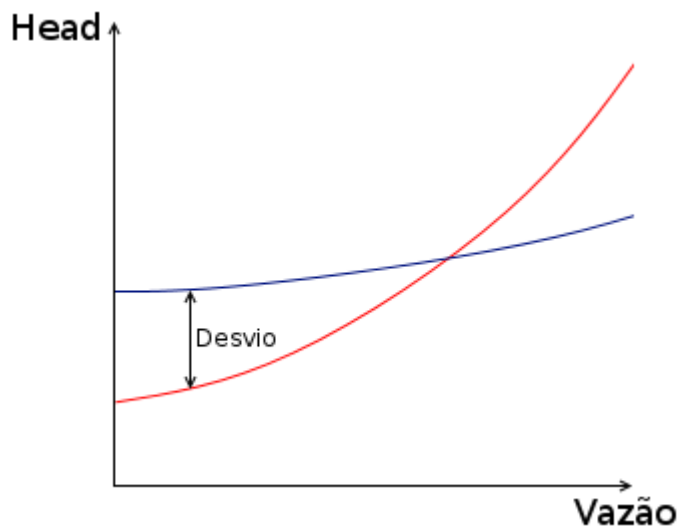
O objetivo da rede neural artificial desenvolvida neste trabalho é modelar um sistema hidráulico de tal forma que a *RNA* represente um modelo equivalente à expressão fenomenológica, isto é, que para os mesmos valores de entrada a rede seja capaz de fornecer os mesmos resultados. Assim, a fim de se compararem os dois métodos, o procedimento mais direto a se fazer é aplicá-los com as mesmas entradas e então verificar o quão semelhantes são as respostas de ambos: se as saídas fornecidas pela rede neural se aproximarem suficientemente daquelas dadas pela expressão fenomenológica, então o treinamento da *RNA* terá sido bem-sucedido. O diagrama mostrado na Figura 51 resume o processo de obtenção dos resultados dos dois métodos usando-se um mesmo conjunto de entradas. Adicionalmente, a Figura 52 mostra a ideia por detrás da comparação entre os dois modelos: se os desvios forem suficientemente pequenos, se poderá dizer que os dois modelos são equivalentes.

Figura 51 – Diagrama resumindo o processo de obtenção das saídas pelos dois métodos.



Fonte: Elaborado pelo autor.

Figura 52 – Curvas do sistema geradas pela expressão fenomenológica (linha vermelha) e pela rede neural artificial (linha azul).



Fonte: Elaborado pelo autor.

Avaliação ao longo de todo o conjunto de amostras

Na avaliação ao longo de todo o conjunto de amostras se compararão os valores armazenados na *matrizHeads_fenom* com aqueles presentes na *matrizHeads_RNA_desnorm*, sendo ambas as matrizes oriundas de uma mesma fonte: a *matrizEntradas* (v. Figura 51). Os resultados da *matrizHeads_fenom* serão obtidos através da aplicação da função fenomenológica *fn_head_sistema*, enquanto que os valores da *matrizHeads_RNA_desnorm* serão calculados pela função da rede neural *RNA_head_sistema*.

A comparação será feita através do cálculo dos *desvios relativos* (em módulo²¹)

²¹ Aqui se escolheu o cálculo do desvio relativo *em módulo* porque se quer saber apenas o *grau* do desvio,

entre os valores presentes na *matrizHeads_fenom* e aqueles presentes na *matrizHeads_RNA_desnorm*. O cálculo do desvio relativo será feito como na Equação 4.14:

$$desvRel = \frac{|valor - correto|}{correto} \cdot 100\% \quad (4.14)$$

em que se obtém o desvio relativo *desvRel* (em porcentagem) de um dado *valor* em relação ao valor *correto*. A comparação é ilustrada pela Figura 52.

Os cálculos dos desvios relativos definidos pela Equação 4.14 são implementados pela função *fnMat_desvRel* (Algoritmo 4.8) e os resultados serão armazenados em um vetor chamado *desvRels* (Algoritmo 4.9).

Algoritmo 4.8 – Função *fnMat_desvRel* .

```

1 function desvRel = fnMat_desvRel(val,correto)
2   % val:      valor que quer se comparar ao "correto".
3   % correto: valor correto em relação ao qual será calculado o desvio
4   % relativo de val.
5   % desvRel: desvio relativo em % (saída da função).
6
7   desvRel = (abs(val - correto)/correto)*100;
8 end

```

Algoritmo 4.9 – Código de contabilização dos desvios relativos.

```

1 % O vetor desvRels armazenará os desvios relativos entre os resultados
2 % fornecidos pela rede neural (armazenados em matrizHeads_RNA_desnorm)
3 % e pela expressão fenomenológica (armazenados em matrizHeads_fenom).
4 desvRels = double.empty(1,0); % Vetor para desvios relativos.
5
6 % Cálculo dos erros/desvios relativos dos resultados fornecidos pela
7 % rede neural em relação àqueles resultados dados pela expressão
8 % fenomenológica.
9 for j = 1:tamAmostra
10  desvRels(1,j) = fnMat_desvRel(matrizHeads_RNA_desnorm(1,j),
    matrizHeads_fenom(1,j));
11 end

```

Na prática o que a linha 10 do Algoritmo 4.9 faz é implementar a Equação 4.14 da

e não se o mesmo foi positivo ou negativo em relação à expressão fenomenológica.

seguinte forma:

$$desvRels(1, j) = \frac{|matrizHeads_RNA_desnorm(1, j) - matrizHeads_fenom(1, j)|}{matrizHeads_fenom(1, j)} \quad (4.15)$$

ou seja, se está admitindo implicitamente que o valor contido em *matrizHeads_fenom* é o valor *correto* em relação ao qual se irá comparar o valor contido em *matrizHeads_RNA_desnorm(1, j)*.

Avaliação segundo específicos conjuntos de parâmetros

Ao comparar os dois modelos em relação a apenas certos conjuntos de parâmetros, o objetivo é verificar como ambos se comportam em função *apenas* da variável *vazão volumétrica*. Em outras palavras, o que se deseja é verificar as curvas do sistema geradas por ambos os modelos e compará-las levando-se em conta alguns conjuntos de parâmetros. O conceito de conjunto de parâmetros foi abordado na subseção 4.4.1, sendo este o conjunto de valores das variáveis de entrada excetuando-se a vazão volumétrica. A comparação é possibilitada pela execução do Algoritmo 4.10 (v. p. 106).

No Algoritmo 4.10 se vê o uso de uma função chamada *fnMat_desvRelPosNeg* na linha 47 para o cálculo dos desvios relativos. Esta se difere da *fnMat_desvRel* (Algoritmo 4.8) apenas por não empregar o módulo no numerador da fração. Assim, enquanto os resultados *fnMat_desvRel* são apenas desvios relativos em *módulo*, sem sinal, a função *fnMat_desvRelPosNeg* fornece valores *positivos e negativos*. A função *fnMat_desvRelPosNeg* baseia-se na equação Equação 4.16, sendo esta implementada através do Algoritmo 4.11 (v. p. 107). A Equação 4.16 é dada por:

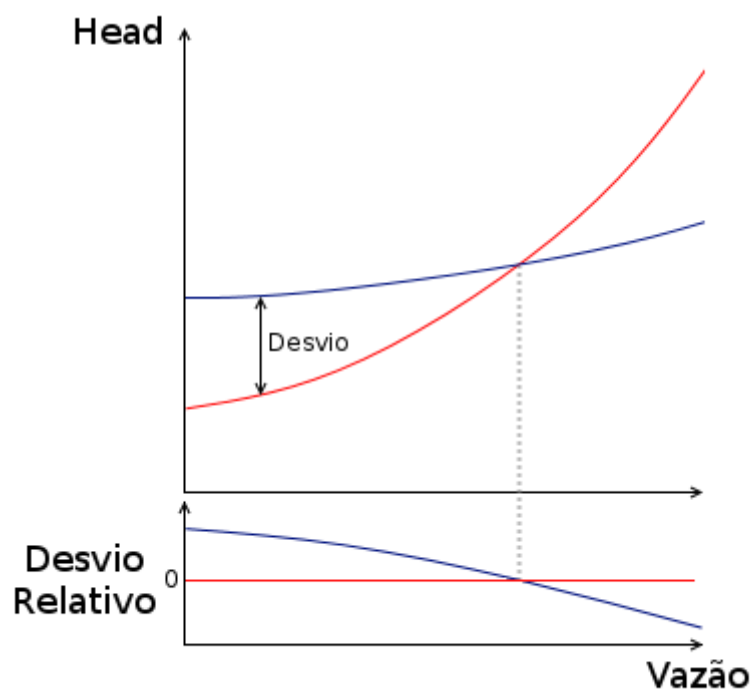
$$desvRelPosNeg = \frac{valor - correto}{correto} \cdot 100\% \quad (4.16)$$

Resumidamente, o que o Algoritmo 4.10 faz é calcular os *heads* em função dos valores de vazão volumétrica dados no vetor *vazao*. Os cálculos dos valores de *head* são feitos tanto pela expressão fenomenológica, *fn_head_sistema*, quanto pela função da rede neural, *RNA_head_sistema*. Os resultados da expressão fenomenológica são armazenados no vetor *head_fenom*, e as saídas já desnormalizadas da função da rede neural são armazenadas no vetor *head_RNA_desnorm*.

As comparações entre a expressão fenomenológica e a rede neural artificial nesta subseção serão feitas através de gráficos, sendo estes plotados no *MATLAB* por meio das linhas de código expostas no Algoritmo 4.12 (v. p. 107). Adicionalmente, abaixo da plotagem das curvas do sistema, serão plotados pequenos gráficos mostrando os desvios relativos referentes ao mesmo conjunto de parâmetros e valores de vazão usados para a

plotagem das curvas do sistema. Esses gráficos servem para demonstrar os desvios relativos entre as duas curvas de $fn_head_sistema$ e $RNA_head_sistema$, com uma ilustração da ideia podendo ser vista na Figura 53.

Figura 53 – Ilustração de como serão as plotagens para comparação entre os dois modelos. A linha vermelha corresponde à expressão fenomenológica, e a azul, à rede neural artificial.



Fonte: Elaborado pelo autor.

Algoritmo 4.10 – Avaliação da rede neural artificial para um conjunto de parâmetros.

```

1 % O vetor vazao conterá os valores de vazão volumétrica que deverão ser
2 % usados na avaliação do conjunto de parâmetros específico.
3 vazao = 0 : 1 : 100;
4
5 % Normalização dos valores de vazão presentes no vetor vazao por meio
6 % da função nativa mapminmax.
7 vazao_norm = mapminmax(vazao);
8
9 % Se estabelece o número do conjunto de parâmetros em relação ao qual
10 % se quer obter as curvas do sistema para ambos os modelos. Esse
11 % número corresponde ao da coluna da matrizEntradasNorm.
12 conj = 1;
13
14 % O vetor head_RNA_norm irá armazenar os resultados fornecidos pela
15 % função da rede neural, RNA_head_sistema.
16 head_RNA_norm = double.empty(1,0);
17 for j = 1:length(vazao)
18     % RNA_head_sistema recebe como entrada um valor de vazão, dado
19     % por vazao_norm(1,j) e seis valores dos 6 parâmetros pertencentes
20     % ao conjunto de parâmetros definido pela variável conj. Após o
21     % cálculo, o resultado é armazenado na coluna j do vetor
22     % head_RNA_norm e então o loop faz o mesmo cálculo, porém, com o
23     % próximo valor presente no vetor vazao.
24     head_RNA_norm(1,j) = RNA_head_sistema([vazao_norm(1,j);matrizEntradasNorm
        (2:7,conj)]);
25 end
26
27 % Os valores presentes em head_RNA_norm estão normalizados de -1 a 1 e,
28 % portanto, serão desnormalizados e armazenados na head_RNA_desnorm. A
29 % struct ts é a mesma obtida durante a normalização da matrizAlvos, e
30 % aqui servirá como elemento de desnormalização.
31 head_RNA_desnorm = mapminmax('reverse',head_RNA_norm,ts);
32
33 % Para um dado conjunto de parâmetros definido pela variável conj, se
34 % calculam os heads para os vários valores de vazão dados no vetor
35 % vazao usando a função fenomenológica fn_head_sistema.
36 head_fenom = double.empty(1,0);
37 for j = 1:length(vazao)
38     head_fenom(1,j) = fn_head_sistema([vazao(j);matrizEntradas(2:7,conj)]);
39 end
40
41 % Cálculo dos desvios relativos entre os valores de head desnormalizados
42 % (head_RNA_desnorm) e os valores de head fornecidos pela função
43 % fenomenológica (armazenados em head_fenom). Os valores dos desvios
44 % relativos serão armazenados no vetor desviosRelativos.
45 desviosRelativos = double.empty(1,0);
46 for j = 1:length(vazao)
47     desviosRelativos(1,j) = fnMat_desvRelPosNeg(head_RNA_desnorm(1,j),
        head_fenom(1,j));
48 end

```

Algoritmo 4.11 – Função *fnMat_desvRelPosNeg* .

```
1 function y = fnMat_desvRelPosNeg(val,correto)
2 % val:      valor que quer se comparar ao "correto".
3 % correcto: valor correcto em relação ao qual será calculado o desvio
4 %           relativo.
5 % y:        desvio relativo em % (saída da função).
6
7 y = ((val - correcto)/correcto)*100;
8 end
```

Algoritmo 4.12 – Plotagens para a avaliação referente a um conjunto de parâmetros.

```
1 % O subplot de cima (das curvas do sistema) vai ocupar os espaços 1, 2 e 3.
2 subplot(4,1,[1 2 3]);
3 plot(vazao,head_RNA_desnorm,vazao,head_fenom);
4
5 % O subplot de baixo (dos desvios relativos versus a vazão) vai ocupar
6 % apenas o espaço 4.
7 subplot(4,1,4);
8 plot(vazao,desviosRelativos);
```

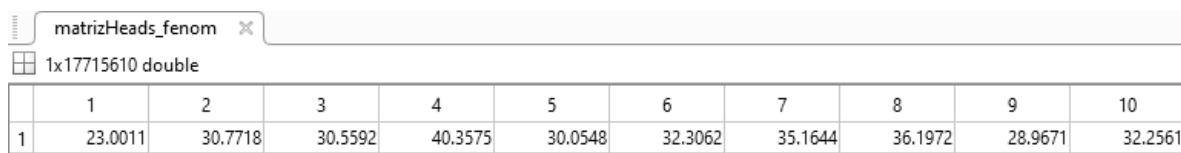

5 Resultados e discussão

Neste capítulo serão mostrados os resultados da aplicação do que fora exposto no Capítulo 4 (*Procedimentos e metodologia*). Aqui serão mostrados os resultados decorrentes da análise para todo o conjunto das amostras bem como aqueles referentes a conjuntos de parâmetros específicos.

5.1 Avaliação ao longo de todo o conjunto de amostras

Como apontado na seção 4.6.4, a avaliação da rede neural artificial ao longo de todo o conjunto de amostras ocorrerá através da comparação entre os valores armazenados na *matrizHeads_fenom* (v. Figura 54), oriundos da função fenomenológica *fn_head_sistema*, e os da *matrizHeads_RNA_desnorm* (v. Figura 55), obtidos da função da rede neural artificial, *RNA_head_sistema*. Assim, será possível comparar os dois modelos por meio de seus resultados.

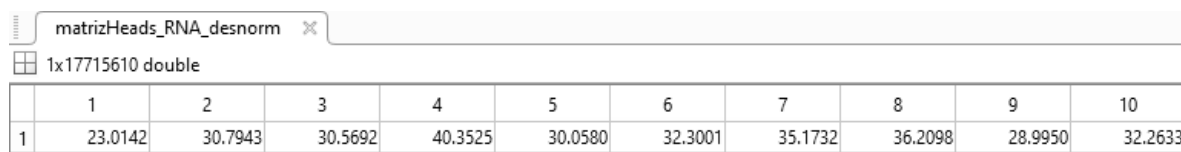
Figura 54 – Colunas iniciais da *matrizHeads_fenom*.



	1	2	3	4	5	6	7	8	9	10
1	23.0011	30.7718	30.5592	40.3575	30.0548	32.3062	35.1644	36.1972	28.9671	32.2561

Fonte: Elaborado pelo autor.

Figura 55 – Colunas iniciais da *matrizHeads_RNA_desnorm*.



	1	2	3	4	5	6	7	8	9	10
1	23.0142	30.7943	30.5692	40.3525	30.0580	32.3001	35.1732	36.2098	28.9950	32.2633

Fonte: Elaborado pelo autor.

Após a contabilização dos desvios relativos feita pelo Algoritmo 4.9, os resultados foram reunidos na Tabela 7, responsável por mostrar o número de ocorrências (segunda coluna) de desvios relativos presentes em determinadas faixas (primeira coluna). Por exemplo: 2.300.386 valores de desvios relativos encontram-se na faixa de 0,1% à 0,2%. Além disto, essas 2.300.386 ocorrências de desvios relativos entre 0,1% e 0,2% correspondem à 12,99% de todas as 17.715.610 ocorrências de desvios relativos. Finalmente, 86,81% dos desvios relativos encontram-se nas faixas [0; 0,1)% e [0,1; 0,2)% (v. a última coluna

chamada “*Acúmulo (%)*”), ou seja, no intervalo $[0; 0,2)\%$ encontram-se 86,81% de todos os desvios relativos.

A Tabela 7 mostra que 98,39% dos resultados fornecidos pela rede neural artificial se encontram a uma distância relativa máxima de até 1,0% (exclusive¹) da resposta fornecida pela expressão fenomenológica, e que todos os resultados fornecidos pela rede neural encontram-se, nos piores dos casos, a uma distância relativa de, no máximo, 4,0%². Os desvios relativos na faixa de 1,0% a 4,0% respondem por 1,61%³ do total de ocorrências, estando todo o restante (98,39%) concentrado abaixo de 1,0% de desvio relativo.

Tabela 7 – Contabilização dos desvios relativos para cada faixa de desvios.

Faixa (%)	Ocorrências	Frequência (%)	Acúmulo (%)
[0 ; 0,1)	13.077.927	73,82	73,82
[0,1 ; 0,2)	2.300.386	12,99	86,81
[0,2 ; 0,3)	863.463	4,87	91,68
[0,3 ; 0,4)	375.649	2,12	93,80
[0,4 ; 0,5)	217.463	1,23	95,03
[0,5 ; 0,6)	171.304	0,97	96,00
[0,6 ; 0,7)	154.379	0,87	96,87
[0,7 ; 0,8)	117.276	0,66	97,53
[0,8 ; 0,9)	85.691	0,48	98,01
[0,9 ; 1,0)	66.312	0,37	98,39
[1,0 ; 1,5)	179.400	1,01	99,40
[1,5 ; 2,0)	67.788	0,38	99,78
[2,0; 4,0]	38.572	0,22	100,00
[0 ; 4,0]	17.715.610	100,00	100,00

Fonte: Elaborado pelo autor.

O desempenho da rede neural artificial dado pela Tabela 7 é demonstrado de outro modo pela regressão plotada na Figura 56. O eixo horizontal corresponde aos valores fornecidos pela rede neural artificial armazenados na matriz *matrizHeads_RNA_desnorm*, e o eixo vertical refere-se aos resultados calculados pela expressão fenomenológica armazenados na matriz *matrizHeads_fenom*. Um ponto no gráfico corresponde às respostas da rede neural e da expressão fenomenológica, de modo que se os resultados forem muito

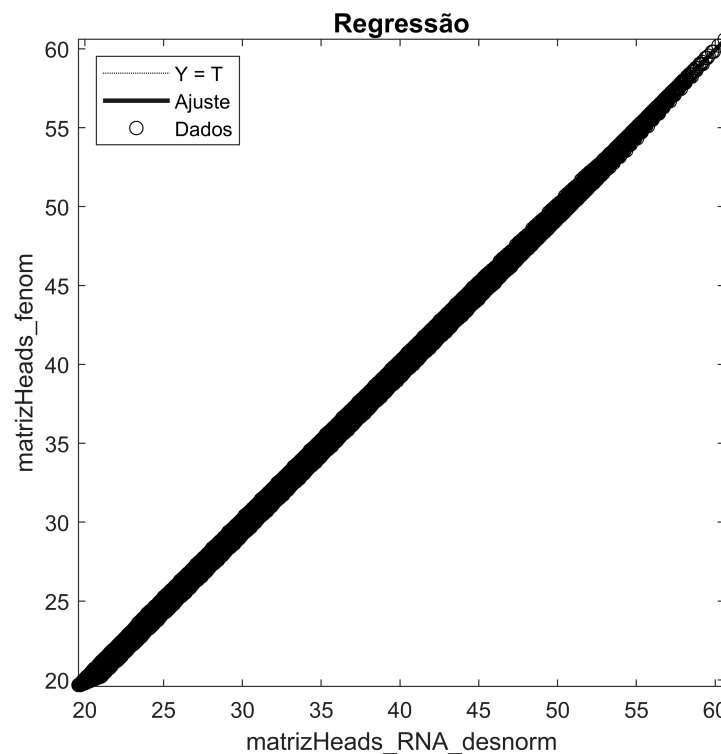
¹ A faixa de valores conta com todos os valores de 0% à 1%, excetuando-se o próprio 1%. Por isto o uso do termo *exclusive*, que é o antônimo do advérbio *inclusive*.

² Já que o somatório dos números de ocorrências em todas as faixas até 4,0% corresponde exatamente ao número de amostras.

³ Este valor de 1,61% é obtido pela soma dos percentuais das faixas que iniciam em 1,0% e vão até 4,0%.

próximos, eles tenderão a se encontrar próximos à reta diagonal $Y = T$. A regressão da Figura 56 corrobora os resultados demonstrados na Tabela 7, pois observa-se que há uma alta concentração de pontos em torno da reta $Y = T$, isto é, os resultados fornecidos pelos dois modelos são muito próximos. Além disto, o coeficiente de correlação entre os valores fornecidos pela RNA e pela expressão fenomenológica é igual a 0,9998.

Figura 56 – Regressão linear entre as saídas dos dois modelos para todo o conjunto de amostras.



Fonte: Elaborado pelo autor.

As medidas descritivas do conjunto de desvios relativos são reunidas na Tabela 8. O erro percentual absoluto médio⁴ é de 0,1161% com um desvio padrão de 0,2336%. O menor desvio relativo é muito próximo de zero, enquanto que o maior é de 4,00%.

A Figura 57 mostra a distribuição dos desvios relativos em forma de histograma. Cada barra do histograma tem uma largura de 0,1%. Nesse histograma pode-se ver como os desvios relativos estão bem concentrados, o que é um excelente indicativo de performance da rede neural artificial.

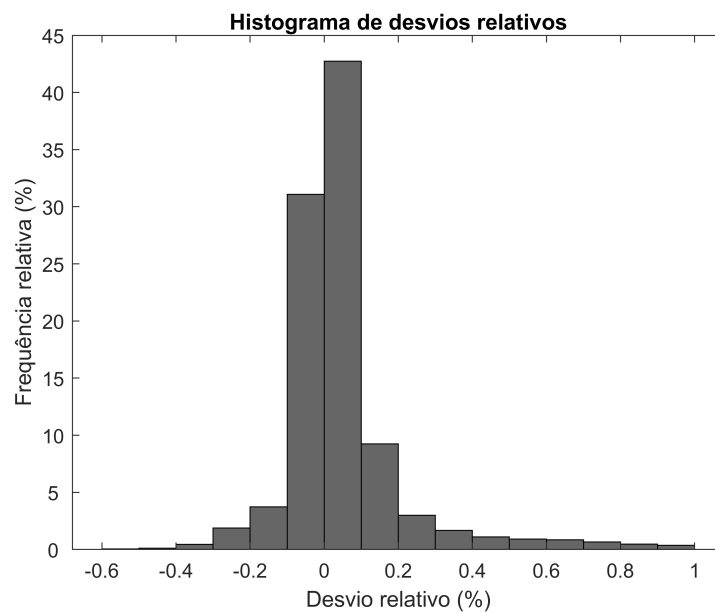
⁴ Essa medida é comumente descrita pela sigla MAPE, do inglês *mean absolute percentage error*.

Tabela 8 – Medidas descritivas dos desvios relativos.

	(%)
MAPE	0,1161
Desvio padrão	0,2336
Mínimo	$1,08 \times 10^{-9}$
Máximo	4,00

Fonte: Elaborado pelo autor.

Figura 57 – Histograma de desvios relativos. Cada barra tem uma largura de 0,1%.



Fonte: Elaborado pelo autor.

5.2 Avaliação em específicos conjuntos de parâmetros

Os valores de alguns conjuntos de parâmetros usados nesta seção são mostrados na Tabela 9. Os valores expressos nesta tabela são exatamente os mesmos que da *matrizEntradas* gerada na fase de avaliação (v. Figura 46 na p. 99), excetuando a primeira linha, da vazão, já que as vazões não fazem parte do conceito de conjuntos de parâmetros definidos neste trabalho (v. subseção 4.4.1).

A plotagem das curvas do sistema para o conjunto de parâmetros 1 aparece na Figura 58, e a do conjunto 7, na Figura 59.

Em relação ao conjunto 1 (v. Figura 58), a plotagem das curvas parece indicar que o treino da *RNA* foi bem-sucedido. No gráfico de desvios relativos *versus* vazão que se situa abaixo das curvas do sistema, constata-se que o maior desvio relativo foi de 0,058%⁵.

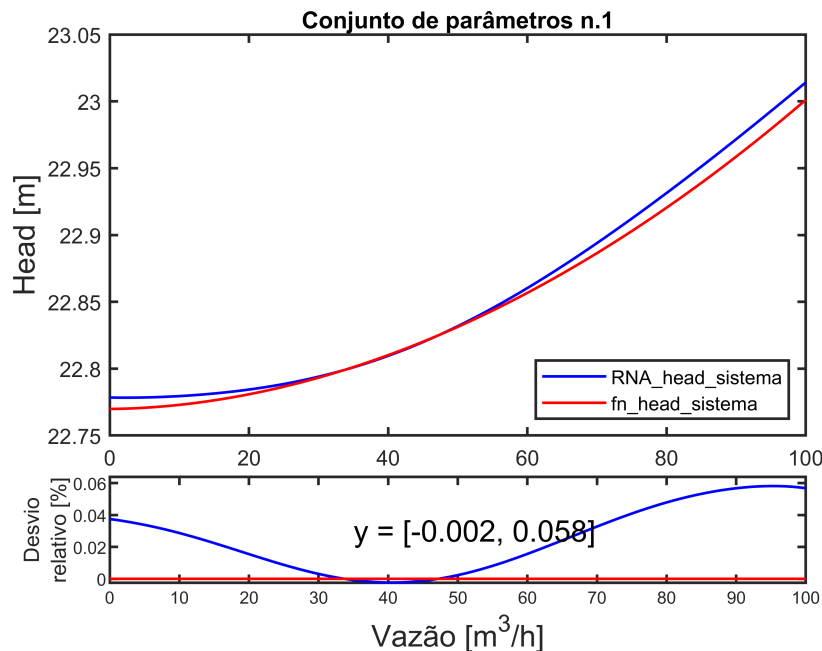
⁵ A amplitude de desvios relativos é mostrada dentro do gráfico e delimitada por dois colchetes, como

Tabela 9 – Valores dos parâmetros usados nas plotagens das curvas do sistema.

Conj.	1	7	11	22	31	49	60	95
pfd (kPa)	10,1325	60,795	91,1925	70,9275	91,1925	50,6625	30,3975	10,1325
Dd (m)	0,26	0,28	0,2	0,26	0,1	0,1	0,18	0,12
Ld (m)	48	48	51	57	48	30	51	48
p0s (kPa)	81,06	10,1325	30,3975	101,325	60,795	60,795	20,265	70,9275
Ds (m)	0,22	0,3	0,2	0,2	0,2	0,14	0,12	0,12
Ls (m)	42	30	48	48	60	6	24	48

Fonte: Elaborado pelo autor.

Figura 58 – Curvas do sistema usando os dois modelos (conj = 1).



Fonte: Elaborado pelo autor.

Quanto ao conjunto 7 (v. Figura 59), apesar de *visualmente* a curva da rede neural (linha azul) apresentar um maior distanciamento em relação à curva da expressão fenomenológica (linha vermelha), o maior desvio relativo foi de 0,058%. Deduz-se daí que se a escala do eixo dos *heads* fosse maior, esse distanciamento visual seria menor, tornado-se quase imperceptível dependendo do tamanho desta escala vertical. A mesma plotagem para o conjunto 7 pode ser vista na Figura 60, em que se percebe que, de fato, para uma escala maior, o distanciamento visual é menor, apesar de os desvios permanecerem essencialmente os mesmos.

Figura 59 – Curvas do sistema usando os dois modelos (conj = 7).

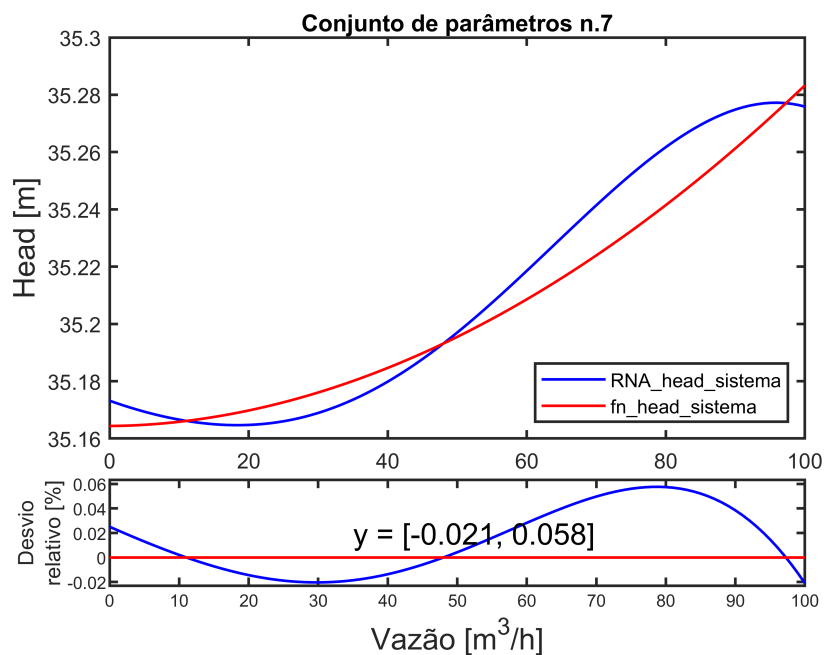
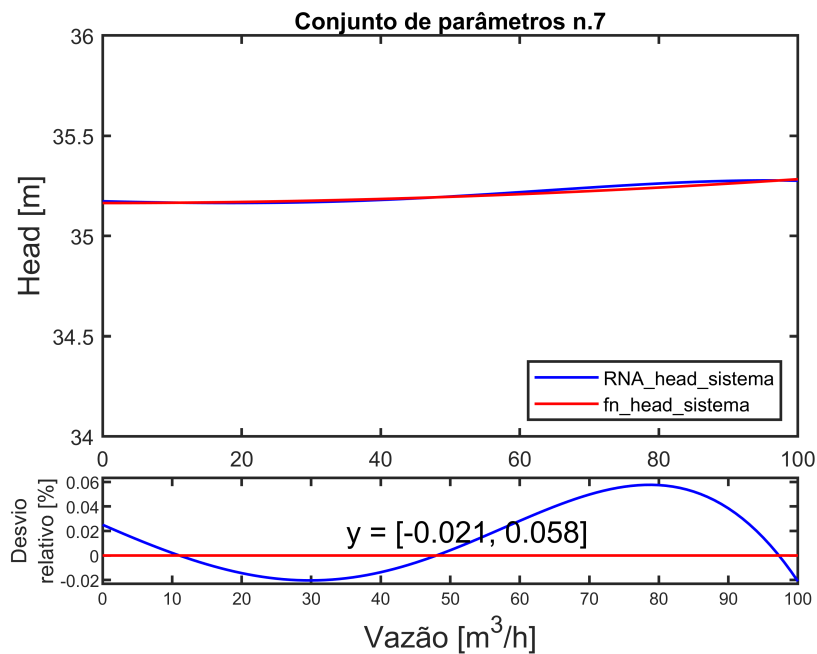
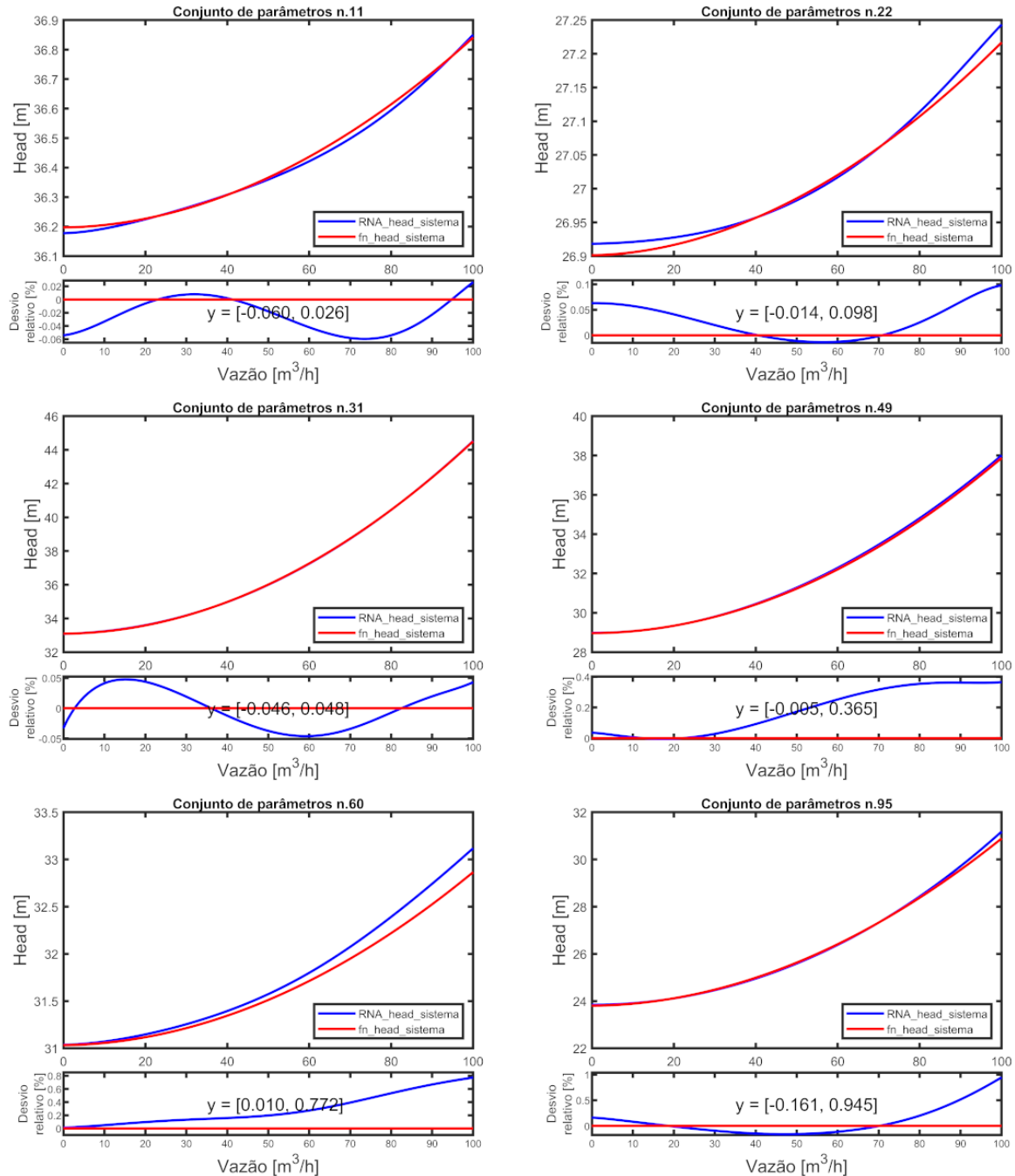


Figura 60 – Curvas do sistema usando os dois modelos (conj = 7), com o eixo dos *heads* em maior escala.



Plotagens dos demais conjuntos de parâmetros presentes na Tabela 9 são mostradas na Figura 61.

Figura 61 – Curvas do sistema usando os dois modelos para diferentes valores de parâmetros.



Fonte: Elaborado pelo autor.

Nas plotagens da Figura 61, percebe-se que as curvas do sistema fornecidas pela rede neural artificial aproximam-se consideravelmente daquelas fornecidas pela expressão fenomenológica, com este mesmo padrão se repetindo para todos os demais conjuntos

de parâmetros pertencentes a todo conjunto de amostras, sendo alguns poucos casos semelhantes ao do conjunto 7.

6 Conclusões

O objetivo de treinar uma rede neural artificial para a modelagem de um sistema hidráulico elevatório foi alcançado. Tendo sido treinada com 163.296 amostras, a rede neural foi capaz de fornecer bons resultados para um número de entradas cerca de 108 vezes maior, isto é, 17.715.610, o que comprova que a mesma alcançou excelente capacidade de generalização.

A boa performance mencionada no último parágrafo é constatada pelos números: 98,39% dos resultados fornecidos pela rede apresentaram um desvio relativo de, no máximo, 1,0%. Além disto, 73,82% das respostas dadas pela rede situaram-se abaixo de 0,1% de desvio em relação ao modelo tradicional, e em nenhuma ocorrência o desvio superou os 4,0%. A plotagem da regressão na Figura 56 fornece uma constatação visual dos resultados mencionados, uma vez que a reunião dos dados resulta numa linha bem concentrada em torno da reta $Y = T$, com nenhum ponto encontrando-se demasiadamente afastado. Além disto, as plotagens das curvas do sistema fornecidas pela rede neural artificial comprovam que esta foi capaz de modelar bem o sistema hidráulico, uma vez que as curvas plotadas pelos dois modelos assemelham-se numericamente e em forma. Quando há divergência de forma, como nos casos em que a curva do sistema plotada pela rede neural oscila, o desvio relativo é pequeno, podendo ser considerado desprezível a depender dos parâmetros do projeto.

Como possibilidade de trabalhos futuros, o autor sugere a modelagem de um sistema hidráulico semelhante, porém com tubulações em paralelo. Há também a possibilidade de modelagem de sistemas hidráulicos em queda, diferentemente deste trabalho, que modelou apenas casos de sistemas elevatórios. Essa mudança poderia ser conseguida, basicamente, transformando em variável a altura final de descarga, que aqui é constante. A utilização de mais variáveis do que as sete aqui empregadas também representa uma potencial fonte de novos trabalhos, devendo-se atentar para o fato de que o emprego de um maior número de variáveis pode não constituir, em essência, um novo trabalho. Além disto, é necessário maior poder computacional para um maior número de variáveis. Outras sugestões incluem: uso de válvulas com diferentes aberturas; inclusão de tubulações com comprimentos menores do que dez vezes os seus respectivos diâmetros, o que introduz a necessidade de modelagem cuidadosa e localizada referente às regiões de entrada; uso de outros tipos de redes neurais artificiais; consideração de outros líquidos que não somente a água (como óleos, por exemplo). O autor acredita que uma sugestão em especial seria a melhor para a continuação deste trabalho: o uso de dados do mundo real para o treinamento da rede neural artificial.

Referências

- ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. *ABNT 1977*: Elaboração de Projetos de Sistemas de Adução de água para Abastecimento Público. Rio de Janeiro, 1977. 55
- BRUNETTI, F. *Mecânica dos fluidos*. 2. ed. São Paulo: Pearson Prentice Hall, 2008. 34, 51, 52, 54
- COLEBROOK, C. F. Turbulent flow in pipes, with particular reference to the transition region between the smooth and rough pipe laws. *Journal of the Institution of Civil Engineers*, v. 11, n. 4, p. 133–156, 1939. Disponível em: <<https://doi.org/10.1680/ijoti.1939.13150>>. Acesso em: 23 out. 2018. 56
- CYBENKO, G. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals, and Systems (MCSS)*, Springer London, v. 2, n. 4, p. 303–314, dec 1989. Disponível em: <<http://dx.doi.org/10.1007/BF02551274>>. Acesso em: 23 out. 2018. 70
- ÇENGEL, Y. A.; CIMBALA, J. M. *Mecânica dos fluidos: fundamentos e aplicações*. 3. ed. Porto Alegre: AMGH, 2012. 25, 29, 30, 33, 34, 35, 36, 37, 38, 39, 41, 43, 44, 45, 46, 47, 49, 50, 52, 53, 55, 83
- FOX, R.; PRITCHARD, P.; MCDONALD, A. *Fox and McDonald's Introduction to Fluid Mechanics*. 8. ed. Hoboken, NJ, US: John Wiley & Sons, Inc., 2011. 40
- GENIĆ, S. et al. A review of explicit approximations of Colebrook's equation. *FME Transactions*, v. 39, n. 2, p. 67–71, 2011. Disponível em: <https://www.mas.bg.ac.rs/_media/istrazivanje/fme/vol39/2/04_mjaric.pdf>. Acesso em: 06 nov. 2018. 57
- HAALAND, S. E. Simple and explicit formulas for the friction factor in turbulent pipe flow. *Journal of Fluids Engineering*, ASME, v. 105, n. 1, p. 89–90, 1983. Disponível em: <<http://dx.doi.org/10.1115/1.3240948>>. Acesso em: 23 out. 2018. 56
- HALLIDAY, D.; RESNICK, R.; WALKER, J. *Fundamentos de física, volume 2: gravitação, ondas e termodinâmica*. 8. ed. Rio de Janeiro: LTC, 2009. v. 2. 32
- HEBB, D. O. *The organization of behavior: A neuropsychological theory*. New York: Wiley, 1949. 73
- HENN, E. *Máquinas de Fluido*. 2. ed. Santa Maria: Editora da UFSM, 2006. 49, 59, 61
- KELLNER, E.; AKUTSU, J.; REIS, L. Avaliação da rugosidade relativa dos tubos de PVC com vistas ao dimensionamento das redes de distribuição de água. v. 21, 04 2016. Disponível em: <http://www.scielo.br/pdf/esa/v21n2/1809-4457-esa-S1413_41522016141081.pdf>. Acesso em: 06 nov. 2018. 54, 55, 80
- KRIESEL, D. *A Brief Introduction to Neural Networks*. [s.n.], 2007. Disponível em: <<http://www.dkriesel.com>>. Acesso em: 06 nov. 2018. 64, 65, 66, 68, 69, 70, 72, 73, 74, 75, 76
- LEITHOLD, L. *O Cálculo com Geometria Analítica*. São Paulo: Harbra, 1994. 71, 72

- LEVENBERG, K. A method for the solution of certain non-linear problems in least squares. *Quarterly Journal on Applied Mathematics*, n. 2, p. 164–168, 1944. Disponível em: <<https://cs.uwaterloo.ca/~y328yu/classics/levenberg.pdf>>. Acesso em: 06 nov. 2018. 77
- MACINTYRE, A. *Bombas e instalações de bombeamento*. Rio de Janeiro: Livros Técnicos e Científicos, 1997. 54, 60
- MARQUARDT, D. W. An algorithm for least-squares estimation of nonlinear parameters. *SIAM Journal on Applied Mathematics*, SIAM, v. 11, n. 2, p. 431–441, 1963. Disponível em: <<http://dx.doi.org/10.1137/0111030>>. Acesso em: 06 nov. 2018. 77
- MARQUES, J.; SOUSA, J. de O. *Hidráulica Urbana. Sistemas de Abastecimento de Água e de Drenagem de Águas Residuais*. 3. ed. Coimbra: Universidade de Coimbra, 2011. 55
- MCCULLOCH, W. S.; PITTS, W. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, v. 5, n. 4, p. 115–133, Dec 1943. ISSN 1522-9602. Disponível em: <<https://doi.org/10.1007/BF02478259>>. Acesso em: 27 out. 2018. 63, 69
- MOODY, L. Friction factors for pipe flow. *Trans. ASME*, v. 66, n. 8, p. 671–677, 1944. 57
- NETTO, J.; ALVAREZ, G. *Manual de hidráulica*. 7. ed. São Paulo: E. Blücher, 1982. 55
- NIELSEN, M. A. *Neural Networks and Deep Learning*. [S.l.]: Determination Press, 2015. 67
- PORTO, R. *Hidráulica Básica*. 2. ed. São Carlos, SP: EESC, 2000. 55
- PRESS, W. H. et al. *Numerical Recipes in C: The Art of Scientific Computing*. New York, NY, USA: Cambridge University Press, 1988. 72
- REYNOLDS, O. An experimental investigation of the circumstances which determine whether the motion of water shall be direct or sinuous, and of the law of resistance in parallel channels. *Philosophical Transactions of the Royal Society of London*, The Royal Society, v. 174, p. 935–982, 1883. Disponível em: <<http://www.jstor.org/stable/109431>>. Acesso em: 25 out. 2018. 37
- REYNOLDS, O. *Papers on Mechanical and Physical Subjects: The sub-mechanics of the universe*. [S.l.]: Cambridge University Press, 1903. 40
- ROJAS, R. *Neural Networks - A Systematic Introduction*. 1. ed. Berlin: Springer-Verlag, 1996. 63, 64, 65, 66, 67, 69, 74
- ROSENBLATT, F. The Perceptron: A Probabilistic Model for Information Storage and Organization in The Brain. *Psychological Review*, p. 65–386, 1958. Disponível em: <<http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.588.3775>>. Acesso em: 27 out. 2018. 69
- ROSSMAN, L. *EPANET 2 - Users Manual*. Washington, D.C., 2000. 55
- RUMELHART, D. E.; MCCLELLAND, J. L. (Ed.). *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 1: Foundations*. Cambridge, MA, USA: MIT Press, 1986. 73

- RUMELHART, D. E.; WIDROW, B.; LEHR, M. The Basic Ideas in Neural Networks. *Commun. ACM*, v. 37, p. 87–92, 03 1994. Disponível em: <<http://doi.acm.org/10.1145/175247.175256>>. Acesso em: 06 nov. 2018. 64, 68, 69
- STEWART, J. *Cálculo, volume I - Tradução da 6a edição norte-americana*. São Paulo: Cengage Learning Edições Ltda., 2010. v. 1. 41
- SWINGLER, K. *Applying Neural Networks: A Practical Guide*. San Francisco: Academic Press, 1996. 94
- TIPLER, P. A.; MOSCA, G. *Física para cientistas e engenheiros, volume 1: mecânica, oscilações e ondas, termodinâmica*. 6. ed. Rio de Janeiro: LTC, 2009. v. 1. 29, 31, 32
- WATT, J.; BORHANI, R.; KATSAGGELOS, A. K. *Machine Learning Refined: Foundations, Algorithms, and Applications*. Cambridge: Cambridge University Press, 2016. 71
- WERBOS, P. J. *Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences*. Tese (Doutorado) — Harvard University, 1974. 75
- WHITE, F. *Mecânica dos Fluidos*. 6. ed. Porto Alegre: McGraw Hill Brasil, 2010. 25, 34, 35, 36, 38, 42, 44, 56, 57, 58, 80, 81, 82
- WIDROW, B. Generalization and information storage in networks of adaline neurons. *Self-Organizing Systems*, Spartan, p. 435–461, 1962. 75
- WILAMOWSKI, B.; IRWIN, J. *The Industrial Electronics Handbook - Intelligent Systems*. Boca Raton, FL, US: CRC Press, 2016. (ENGnetBASE 2015). 77, 78
- ZIGRANG, D. J.; SYLVESTER, N. D. Explicit approximations to the solution of Colebrook's friction factor equation. *AIChE Journal*, v. 28, n. 3, p. 514–515, 1982. Disponível em: <<https://onlinelibrary.wiley.com/doi/abs/10.1002/aic.690280323>>. Acesso em: 25 out. 2018. 57
- ZURADA, J. *Introduction to Artificial Neural Systems*. St. Paul, MN, USA: West Publishing Co., 1992. 67, 93

Anexos

ANEXO A – A função *fn_head_sistema* e suas auxiliares

A função *fn_head_sistema* é primeiro apresentada integralmente. As funções cujas chamadas aparecem dentro de *fn_head_sistema* serão apresentadas nas seções seguintes, na mesma ordem em que aparecerem na *fn_head_sistema*. As funções *fn_K_cotov_90*, *fn_K_gaveta*, e *fn_K_globo* fazem uso de funções de interpolação que serão mostradas na seção A.14 e na seção A.15.

A.1 A função *fn_head_sistema*

```

1 function head_sist = fn_head_sistema(matParamsEntr)
2
3 % Abaixo se atribui a cada variável o valor correspondente contido na
4 % matParamsEntr (parâmetro de entrada da atual função):
5 Q_vazao = matParamsEntr(1); % Vazão volumétrica [ $m^3/h$ ]
6 pfd = matParamsEntr(2); % Pressão no final da descarga [ $kPa$ ]
7 Dd = matParamsEntr(3); % Diâmetro da tubulação de descarga [ $m$ ]
8 Ld = matParamsEntr(4); % Comprimento da tubulação de descarga [ $m$ ]
9 p0s = matParamsEntr(5); % Pressão no início da sucção [ $kPa$ ]
10 Ds = matParamsEntr(6); % Diâmetro da tubulação de sucção [ $m$ ]
11 Ls = matParamsEntr(7); % Comprimento da tubulação de sucção [ $m$ ]
12
13 % Inicialização das constantes referentes ao fluido e ao ambiente:
14 g = 9.81; % [ $m/s^2$ ] – Constante gravitacional na superfície da Terra
15 rho = 1000; % [ $kg/m^3$ ] – Massa específica da água
16 mu = 0.001; % [ $Pa.s$ ] – Viscosidade dinâmica da água a 20 graus Celsius
17
18 % Inicialização das variáveis referentes ao fluido:
19 Q_seg = Q_vazao/(3600); % [ $m^3/s$ ] – Vazão convertida de  $m^3/h$  para  $m^3/s$ 
20 gamma = fn_peso_gamma(rho,g); % [ $N/m^3$ ] – Peso específico da água
21
22 % -----
23 % DESCARGA
24 % -----
25 % Inicialização da pressão no final da descarga:
26 % Como a função fn_head_sistema recebe essa pressão em  $kPa$ , uma
27 % conversão para  $Pa$  é necessária multiplicando-se por 1000.
28 p_f_desc = pfd*1000; % [ $Pa$ ] – Pressão no final da descarga
29

```

```

30 % Inicialização das variáveis referentes à estrutura da tubulação:
31 D_desc = Dd; % [m] – Diâmetro da tubulação de descarga
32 L_desc = Ld; % [m] – Comprimento da tubulação de descarga
33 z_f_desc = 30; % [m] – Altura final da descarga
34 e_desc_met = 0.1/1000; % [m] – Rugosidade absoluta da descarga (0,1 mm)
35
36 % Cálculos das grandezas derivadas:
37 K_desc = 2*fn_K_cotov_90(D_desc) + 1*fn_K_gaveta(D_desc) + 1*fn_K_globo(
    D_desc); % Coef. de perda local.
38 A_desc = fn_area_escoam(D_desc); % Área da seção transversal
39 c_desc = fn_vel_escoam_c(Q_seg,A_desc); % Velocidade de escoamento
40 Re_desc = fn_Re(rho,c_desc,D_desc,mu); % Núm. de Reynolds
41 f_desc = fn_f_Darcy(e_desc_met,D_desc,Re_desc); % Fator de atrito
42
43 % Cálculos das perdas de carga (descarga):
44 h_c_desc = fn_h_c(f_desc,L_desc,D_desc,c_desc,g); % Perda de carga contí-
    nua
45 h_l_desc = fn_h_l(K_desc,c_desc,g); % Perda de carga local
46 h_f_desc = fn_h_f(h_c_desc,h_l_desc); % Perda de carga (total)
47
48 % -----
49 % SUÇÃO
50 % -----
51 % Inicialização da pressão no início da sucção:
52 % Como a função fn_head_sistema recebe essa pressão em kPa, uma
53 % conversão para Pa é necessária multiplicando-se por 1000.
54 p_0_suc = p0s*1000; % [Pa] – Pressão no início da sucção
55
56 % Inicialização das variáveis referentes à estrutura da tubulação:
57 D_suc = Ds; % [m] – Diâmetro da tubulação de sucção
58 L_suc = Ls; % [m] – Comprimento da tubulação de sucção
59 z_0_suc = 0; % [m] – Altura inicial da sucção
60 e_suc_met = 0.1/1000; % [m] – Rugosidade absoluta da sucção (0,1 mm)
61
62 % Cálculos das grandezas derivadas:
63 K_suc = 1*fn_K_cotov_90(D_suc) + 1*fn_K_gaveta(D_suc); % Coef. de perda
    local.
64 A_suc = fn_area_escoam(D_suc); % Área da seção transversal
65 c_suc = fn_vel_escoam_c(Q_seg,A_suc); % Velocidade de escoamento
66 Re_suc = fn_Re(rho,c_suc,D_suc,mu); % Núm. de Reynolds
67 f_suc = fn_f_Darcy(e_suc_met,D_suc,Re_suc); % Fator de atrito
68
69 % Cálculos das perdas de carga (sucção):
70 h_c_suc = fn_h_c(f_suc,L_suc,D_suc,c_suc,g); % Perda de carga contínua
71 h_l_suc = fn_h_l(K_suc,c_suc,g); % Perda de carga local
72 h_f_suc = fn_h_f(h_c_suc,h_l_suc); % Perda de carga (total)
73

```



```

74 % -----
75
76 % Cálculo da diferença energética entre a descarga e a sucção.
77 delta_en_mec = fn_energ_mec(z_f_desc,p_f_desc,c_desc,gamma,g) -
    fn_energ_mec(z_0_suc,p_0_suc,c_suc,gamma,g);
78
79 % Calcula-se o head do sistema (head_sist) como sendo a diferença
80 % de energias mecânicas entre a descarga e a sucção (delta_en_mec)
81 % mais as perdas de carga na sucção e na descarga.
82 head_sist = delta_en_mec + h_f_desc + h_f_suc;
83
84 end % Fim da função fn_head_sistema

```

A.2 A função *fn_peso_gamma*

```

1 function p_gamma = fn_peso_gamma(rho,grav)
2     % rho: massa específica [kg/m³]
3     % grav: acel. da gravidade [m/s²]
4     % p_gamma: peso específico [N/m³] (saída da função)
5
6     p_gamma = rho*grav;
7 end

```

A.3 A função *fn_K_cotov_90*

```

1 function K_cotov_90_normal = fn_K_cotov_90(D)
2     % D: diâmetro [m]
3     % K_cotov_90_normal: coef. de perda local [adim.] (saída da função)
4
5     % Os dados usados para interpolação foram obtidos na Tabela 6.5 (p.
6     % 397) do livro Mecânica dos Fluidos – Frank M. White.
7     % Flangeada. 90 graus normal.
8     diams = [0.025 0.050 0.100 0.200 0.500];
9     val_K = [0.500 0.390 0.300 0.260 0.210];
10
11     K_cotov_90_normal = fnMat_interpol_faixas(D,diams,val_K);
12 end

```

A.4 A função `fn_K_gaveta`

```
1 function K_valv_gaveta = fn_K_gaveta(D)
2     % D: diâmetro [m]
3     % K_valv_gaveta: coef. de perda local [adim.] (saída da função)
4
5     % Os dados usados para interpolação foram obtidos na Tabela 6.5 (p.
6     % 397) do livro Mecânica dos Fluidos – Frank M. White.
7     % Flangeada.
8     diams = [0.025 0.050 0.100 0.200 0.500];
9     val_K = [0.800 0.350 0.160 0.070 0.030];
10
11     K_valv_gaveta = fnMat_interpol_faixas(D, diams, val_K);
12 end
```

A.5 A função `fn_K_globo`

```
1 function K_valv_globo = fn_K_globo(D)
2     % D: diâmetro [m]
3     % K_valv_globo: coef. de perda local [adim.] (saída da função)
4
5     % Os dados usados para interpolação foram obtidos na Tabela 6.5 (p.
6     % 397) do livro Mecânica dos Fluidos – Frank M. White.
7     % Flangeada.
8     diams = [0.025 0.050 0.100 0.200 0.500];
9     val_K = [13.0 8.500 6.000 5.800 5.500];
10
11     K_valv_globo = fnMat_interpol_faixas(D, diams, val_K);
12 end
```

A.6 A função `fn_area_escoam`

```
1 function A = fn_area_escoam(D)
2     % D: diâmetro da seção transversal [m]
3     % A: área da seção transversal [m2] (saída da função)
4
5     A = pi*((D/2)^2);
6 end
```

A.7 A função `fn_vel_escoam_c`

```
1 function c = fn_vel_escoam_c(Q,A)
2     % Q: vazão [ $m^3/s$ ]
3     % A: área da seção transversal [ $m^2$ ]
4     % c: velocidade do escoamento [ $m/s$ ] (saída da função)
5
6     c = Q/A;
7 end
```

A.8 A função `fn_Re`

```
1 function Reynolds = fn_Re(rho,c,D,mu)
2     % rho: massa específica [ $kg/m^3$ ]
3     % c: velocidade do escoamento [ $m/s$ ]
4     % D: diâmetro da seção transversal [ $m$ ]
5     % mu: viscosidade dinâmica [ $Pa.s$ ]
6     % Re: número de Reynolds [adim.] (saída da função)
7
8     Reynolds = (rho*c*D)/mu;
9 end
```

A.9 A função `fn_f_Darcy`

```

1 function f = fn_f_Darcy(e,D,Re)
2   % e: rugosidade absoluta [m]
3   % D: diâmetro da seção transversal [m]
4   % Re: número de Reynolds [adim.]
5   % f: fator de atrito de Darcy [adim.] (saída da função)
6
7   if Re <= 2300 % Se o regime for laminar
8       f = 64/Re;
9   else % Se o regime for turbulento (Re > 2300)
10      % As variáveis a, b, c, H, e J são auxiliares.
11      % f é dado por Zigrang e Sylvester (1982).
12      a = (e/D)/3.7;
13      b = 5.02/Re;
14      c = 13/Re;
15      H = log10(a+c);
16      J = log10(a-(b*H));
17      f = (-2*log10(a-(b*J)))^(-2);
18  end

```

A.10 A função `fn_h_c`

```

1 function h_cont = fn_h_c(f,L,D,c,grav)
2   % f: fator de atrito de Darcy [adim.]
3   % L: comprimento da tubulação [m]
4   % D: diâmetro da seção transversal [m]
5   % c: velocidade do escoamento [m/s]
6   % grav: acel. da gravidade [m/s²]
7   % h_cont: perda de carga contínua [m] (saída da função)
8
9   % O if abaixo é necessário, pois para o cálculo da perda contínua para
10  % uma vazão igual à zero (que implica numa vel. de escoamento igual à
11  % zero), o fator de atrito f resulta num NaN ("not a number"). Esse
12  % cuidado é necessário somente aqui no cálculo da perda de carga
13  % contínua, pois essa é a única perda de carga que considera o fator de
14  % atrito de Darcy.
15  if c~= 0 % Se a velocidade do escoamento for diferente de 0.
16      h_cont = f*(L/D)*((c^2)/(2*grav));
17  else % Se a velocidade for igual a 0.
18      h_cont = 0;
19  end
20 end

```

A.11 A função *fn_h_l*

```
1 function h_local = fn_h_l(K,c,grav)
2     % K: constante relacionada ao(s) obstáculo(s) [adim.]
3     % c: velocidade do escoamento [m/s]
4     % grav: acel. da gravidade [m/s2]
5     % h_local: perda de carga local [m] (saída da função)
6
7     h_local = K*((c^2)/(2*grav));
8 end
```

A.12 A função *fn_h_f*

```
1 function h_f_total = fn_h_f(h_c,h_l)
2     % h_c: perda de carga contínua [m]
3     % h_l: perda de carga local [m]
4     % h_f_total: perda de carga total [m] (saída da função)
5
6     h_f_total = h_c + h_l;
7 end
```

A.13 A função *fn_energ_mec*

```
1 function energMec = fn_energ_mec(z,p,c,gamma,grav)
2     % z: altura [m]
3     % p: pressão [Pa]
4     % c: velocidade do escoamento [m/s]
5     % gamma: peso específico [N/m3]
6     % grav: acel. da gravidade [m/s2]
7     % energMec: energia mecânica do fluido [m] (saída da função)
8
9     energMec = (p/gamma) + (c^2)/(2*grav) + z;
10 end
```

A.14 A função *fnMat_interpol_linear*

```
1 function y = fnMat_interpol_linear(x,x1,x2,y1,y2)
2     % x: valor da variável em relação a qual se quer saber o valor de y.
3     % x1: extremo mínimo do intervalo da variável.
4     % x2: extremo máximo do intervalo da variável.
5     % y1: extremo mínimo do intervalo do valor da função.
6     % y2: extremo máximo do intervalo do valor da função.
7     % y: valor interpolado para x (saída da função).
8
9     % m é a inclinação da reta entre os pontos (x1,y1) e (x2,y2).
10    m = (y2 - y1)/(x2 - x1);
11    y = m*(x - x1) + y1;
12 end
```

A.15 A função `fnMat_interpol_faixas`

```
1 function y = fnMat_interpol_faixas(x,vetorX,vetorY)
2     % Descrição:
3     % Essa função faz a interpolação de um valor y de acordo com faixas de
4     % valores fornecidas pelos vetores vetorX e vetorY. Dado um certo valor
5     % x de entrada, se achará o y interpolado.
6
7     % x: variável genérica em relação à qual se quer saber o y
8     % correspondente obtido por interpolação.
9     % vetorX: vetor de valores da variável independente.
10    % vetorY: vetor de valores da variável dependente.
11    % y: valor interpolado (saída da função).
12
13    if vetorX(1) <= x && x < vetorX(2)
14        minX = vetorX(1);
15        maxX = vetorX(2);
16        minY = vetorY(1);
17        maxY = vetorY(2);
18        y = fnMat_interpol_linear(x,minX,maxX,minY,maxY);
19    elseif vetorX(2) <= x && x < vetorX(3)
20        minX = vetorX(2);
21        maxX = vetorX(3);
22        minY = vetorY(2);
23        maxY = vetorY(3);
24        y = fnMat_interpol_linear(x,minX,maxX,minY,maxY);
25    elseif vetorX(3) <= x && x < vetorX(4)
26        minX = vetorX(3);
27        maxX = vetorX(4);
28        minY = vetorY(3);
29        maxY = vetorY(4);
30        y = fnMat_interpol_linear(x,minX,maxX,minY,maxY);
31    elseif vetorX(4) <= x && x <= vetorX(5)
32        minX = vetorX(4);
33        maxX = vetorX(5);
34        minY = vetorY(4);
35        maxY = vetorY(5);
36        y = fnMat_interpol_linear(x,minX,maxX,minY,maxY);
37    end
38
39 end % Fim da função fnMat_interpol_faixas.
```


Índice

- Algoritmo
 - de aprendizado, 74
 - de Gauss-Newton, 77
 - de Levenberg-Marquardt, 77, 93
- Alvo, 89, 93
- Aproximador universal de funções, 70
- Atrito mecânico, 32
- Calor, 31, 32, 44
 - transferência de, 32, 48
- Camada
 - de entrada, 67, 95
 - de saída, 67, 95
 - oculta, 67, 93
- Carga manométrica, 49
- Coefficiente
 - de aprendizado, 73, 75
 - de perda local, K , 53, 81
- Condição de não-escorregamento, 37
- Conjunto de parâmetros, 83, 104
 - quantidade de, 85
- Constante gravitacional, 49
- Constantes, 80
- Curva do sistema, 104
- Desnormalização, 101
- Desvio, 101
 - relativo, 102
- Diagrama de Moody, 57
- Efeitos da entrada, 83
- Energia, 29
 - cinética, 30
 - específica, 44
 - fator de correção, 83
 - de escoamento, 33
 - equação da, 58
 - interna, 31
 - específica, 44, 48
 - mecânica, 29, 30
 - dissipada, 48
 - potencial gravitacional, 30, 31
 - específica, 44
 - química, 29
 - térmica, 29, 31
 - total do sistema
 - específica, 44, 48
- Entrada de rede, 64, 93
- Equação
 - de Bernoulli, 50
 - de Colebrook, 56
- Erro (RNA), 69, 73
 - específico, 74
 - retropropagação do, 75
 - total, 74
- Escoamento
 - em regime estacionário, 50, 83
 - em regime permanente, 34
 - externo, 34
 - incompressível, 35, 50
 - interno, 34, 83
 - invíscido, 50
 - laminar, 33, 35, 38
 - turbulento, 34, 35, 38
- Expressão fenomenológica, 79, 96, 101
- Fator de atrito, 52, 55
 - Escoamento laminar, 56
 - Escoamento turbulento, 56
- Função
 - de ativação, 65
 - de propagação, 64
 - de saída, 66

- erro, 73
- logística, 65
- sigmóide, 65
- tangente hiperbólica, 65
- Generalização (RNA), 68, 93, 96
- Gradiente
 - de uma função, 71
 - descendente, 71, 72
- Head
 - do sistema, 59
- Lei de aprendizado de Hebb, 73
- Máquina de Carnot, 32
- Máquina de fluxo, 80
- Massa
 - conservação de, 38, 41
 - específica, 35, 43, 51
- MATLAB, 86, 92
- Número de amostras, 85, 98
- Número de Reynolds, 34, 37, 56
 - crítico, 38, 82
- Neurônio artificial, 63
- Normalização, 90
- Padrão de entrada, 86
- Perceptron, 69
 - de camada única, 70
 - multicamadas, 70
- Perda de carga, 33, 51
 - contínua, 51
 - local, 52
 - total, 54
- Perfil de velocidades, 37
- Peso (RNA), 67, 73, 95
- Pressão, 45
 - dinâmica, 51
 - estática, 51
 - forças de, 46
 - hidrostática, 51
 - perda de, 51
- Redes Neurais Artificiais, 26, 63, 67, 79
 - feedforward*, 69, 92
- Região
 - de descarga, 79
 - de sucção, 79
- Regra Delta, 75
- Resolução, 84, 87, 97
- Rugosidade, 54
 - absoluta, 54, 80
 - relativa, 54, 56
- Sistema, 45
 - hidráulico, 79
- Superfície de controle, 40
- Tensão de cisalhamento, 51, 55
- Teorema de Cybenko, 70
- Teorema de Transporte de Reynolds, 40, 46
- Teorema do valor médio para integrais, 41
- Termodinâmica
 - Primeira Lei da, 31, 44
 - Segunda Lei da, 32
- Trabalho, 29, 30, 44
 - de eixo, 44, 50
 - de pressão, 44, 45
 - total, 45
 - transferência de energia por, 44
- Treinamento da RNA, 92
- Válvula
 - gaveta, 80
 - globo, 80
- Variáveis, 80
- Vazão
 - em massa, 40–42
 - em volume, 40, 43, 104
- Velocidade

-
- média de escoamento, 40–43
 - perfil de, 40
 - Viés (RNA), 64, 95
 - Viscosidade, 35
 - cinemática, 36
 - dinâmica, 35
 - Volume de controle, 40, 41