



UNIVERSIDADE ESTADUAL DE CAMPINAS

Faculdade de Engenharia Mecânica

PEDRO FRANCIS LOPES

**Neural network modelling of automatic brake
systems for heavy haul railway vehicles**

**Modelamento por redes neurais de sistema de freio
automático para veículos ferroviários de carga**

Campinas

2020

PEDRO FRANCIS LOPES

Neural network modelling of automatic brake systems for heavy haul railway vehicles

Dissertação de Mestrado apresentada à Faculdade de Engenharia Mecânica da Universidade Estadual de Campinas como parte dos requisitos exigidos para a obtenção do título de Mestre em Engenharia Mecânica, na Área de Mecânica dos Sólidos e Projeto Mecânico

Orientador: Prof. Dr. Auteliano Antunes dos Santos Junior

ESTE TRABALHO CORRESPONDE À
VERSÃO FINAL DA DISSERTAÇÃO DE
MESTRADO DEFENDIDA PELO ALUNO
PEDRO FRANCIS LOPES, E ORIENTADA
PELO PROF. DR. AUTELIANO ANTUNES
DOS SANTOS JUNIOR.

Campinas

2020

Ficha catalográfica
Universidade Estadual de Campinas
Biblioteca da Área de Engenharia e Arquitetura
Rose Meire da Silva - CRB 8/5974

L881n Lopes, Pedro Francis, 1981-
Neural network modelling of automatic brake systems for heavy haul
railway vehicles / Pedro Francis Lopes. – Campinas, SP : [s.n.], 2021.

Orientador: Auteliano Antunes dos Santos Júnior.
Dissertação (mestrado) – Universidade Estadual de Campinas, Faculdade
de Engenharia Mecânica.

1. Ferrovias - Freios. 2. Locomotivas - Freios. 3. Engenharia ferroviária. 4.
Redes neurais (Computação). 5. Transporte ferroviário. I. Santos Júnior,
Auteliano Antunes dos, 1963-. II. Universidade Estadual de Campinas.
Faculdade de Engenharia Mecânica. III. Título.

Informações para Biblioteca Digital

Título em outro idioma: Modelamento por redes neurais de sistemas de freio automático
para veículos de carga pesada ferroviários

Palavras-chave em inglês:

Railway - Brakes

Locomotives - Brakes

Rail engineering

Neural Networks (Computing)

Rail Transport

Área de concentração: Mecânica dos Sólidos e Projeto Mecânico

Titulação: Mestre em Engenharia Mecânica

Banca examinadora:

Auteliano Antunes dos Santos Júnior [Orientador]

André Ricardo Fioravanti

Romis Ribeiro de Faissol Attux

Data de defesa: 14-01-2021

Programa de Pós-Graduação: Engenharia Mecânica

Identificação e informações acadêmicas do(a) aluno(a)

- ORCID do autor: <https://orcid.org/0000-0002-2697-1732>

- Currículo Lattes do autor: <http://lattes.cnpq.br/0922142894541160>

**UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE ENGENHARIA MECÂNICA**

DISSERTAÇÃO DE MESTRADO

**Neural network modelling of automatic brake
systems for heavy haul railway vehicles**

Autor: Pedro Francis Lopes

Orientador: Prof. Dr. Auteliano Antunes dos Santos Junior

Coorientador:

A Banca Examinadora composta pelos membros abaixo aprovou esta Dissertação de Mestrado:

Prof. Dr. Auteliano Antunes dos Santos Junior

Departamento de Sistemas Integrados / FEM / UNICAMP

Prof. Dr. André Ricardo Fioravanti

Departamento de Mecânica Computacional / FEM / UNICAMP

Prof. Dr. Romis Ribeiro de Faissol Attux

**Departamento de Engenharia de Computação e Automação Industrial / FEEC /
UNICAMP**

A Ata de Defesa com as respectivas assinaturas dos membros encontra-se no SIGA/Sistema de Fluxo de Dissertação/Tese e na Secretaria do Programa da Unidade.

Campinas, 15 de janeiro de 2021

ACKNOWLEDGEMENTS

I would like to thank Vale for the economical support on the early stages of this research.

I would also like to thank FEM, specially prof. Auteliano and the academic community for receiving me on its bosom after so many years estranged .

Icaro and Jony for the insights and help with the previous programs.

My family for the unflinching support though the many difficulties and hurdles.

Also my friends and colleagues for putting up with me.

"You all know how important you are"

RESUMO

Saber o comportamento do sistema de freio é essencial para simular as forças em uma composição ferroviária (locomotiva + vagões). Detalhar o seu funcionamento é particularmente útil para a análise de segurança, otimização de performance, investigação de acidentes, eficiência da malha, etc. A válvula de freio automática é um fundamento da infraestrutura ferroviária e o componente principal usado durante a frenagem. Em composições longas, sua ação é crucial para garantir que o tempo de frenagem dos veículos é curto para evitar choques excessivos nos aparelhos de choque-tração. Em simuladores ferroviários de tempo real o tempo de execução é importante e a complexidade da resposta da válvula é um desafio para os desenvolvedores de programa. A intenção deste trabalho é criar um modelo que possa representar o comportamento da válvula de freio automática de forma precisa e rápida. Para atingir este objetivo são investigados a linearidade do sistema e métodos para a sua adequada representação. Para a abordagem não linear, o trabalho explora redes neurais NARX como caixa preta e o impacto na alteração de seus parâmetros. Um programa que representa a válvula ABDX de freio é elaborado e sua precisão é avaliada em relação ao objetivo e o tempo de execução é comparado com os algoritmos atualmente disponíveis, mostrando uma redução de mais de três vezes no tempo de processamento.

Palavras-chave: Válvula de freio automática; simulação ferroviária; ARX; NARX; Redes Neurais

ABSTRACT

Knowing the behaviour of the brake system is essential to properly simulate the forces within a railway composition (locomotive + freight cars). Detailing how it works is particularly useful on the safety analysis, performance optimization, accident investigation, network efficiency, etc. The automatic brake valve is the mainstay of the rail brake infrastructure and the main component used during the braking. In long compositions, its action is crucial to guarantee that the time until the last car to brake is short enough to prevent shocks in the draft gears. In real-time train simulators, the processing time is important, and the complexity of the valve response is a challenge for the software developers. The intent of this work is to create a model that can represent the automatic valve behaviour in a fast and accurate way. To achieve this objective the linearity of the system and the adequate representation methods are investigated. For the nonlinear approach, the work explores a black box NARX Neural Network and the impact of changing its parameters. A program that represents the ABDX valve is concocted, the accuracy against the target is assessed and the running time is compared with a model currently used in simulators, showing a reduction of more than three in the processing times.

Keywords: Automatic Brake Valve, Rail Simulation, ARX, NARX, Neural Networks

LIST OF FIGURES

Figure 2.1 – Basic topology of heavy haul freight wagon brake system	31
Figure 2.2 – Basic topology of heavy haul freight wagon brake system	31
Figure 2.3 – Basic topology of locomotive automatic brake system	32
Figure 2.4 – 26-C valve that commands the 26-L brake system (NYAB, 1964)	33
Figure 2.5 – Typical location of the valve on a freight car.	35
Figure 2.6 – Cross section of the AB automatic brake valve (Corrêa, a). On the left is the emergency portion and on the right the service portion.	36
Figure 2.7 – Cross section of the ABD automatic brake valve (Corrêa, b). On the left is the emergency portion and on the right the service portion	36
Figure 2.8 – cross section of the ABDX automatic brake valve (Corrêa, c). On the left is the emergency portion and on the right the service portion	37
Figure 3.1 – problem topology and required models	41
Figure 3.2 – overall strategy to address the problem	42
Figure 3.3 – Modelling data : Brake Cylinder (top) Brake Pipe & Auxiliary Reservoir (bottom) horizontal values are time [ds] and vertical values are pressure [psi]	44
Figure 3.4 – Validation VAL01- Top, output: Brake Cylinder - bottom, Input: Brake Pipe & Auxiliary Reservoir. Horizontal values are time [ds] and vertical values are pressure [psi]	45
Figure 3.5 – Validation VAL02- Top, output: Brake Cylinder - bottom, Input: Brake Pipe & Auxiliary Reservoir. Horizontal values are time [ds] and vertical values are pressure [psi]	45
Figure 3.6 – Validation VAL03- Top, output: Brake Cylinder - bottom, Input: Brake Pipe & Auxiliary Reservoir. Horizontal values are time [ds] and vertical values are pressure [psi]	46
Figure 3.7 – Validation VAL04- Top, output: Brake Cylinder - bottom, Input: Brake Pipe & Auxiliary Reservoir. Horizontal values are time [ds] and vertical values are pressure [psi]	46
Figure 3.8 – Decision tree rationale for the search algorithm	52
Figure 3.9 – A NARX network (Jr.; Barreto, 2006)	53

Figure 3.10–Comparison between output of the model and legacy output.	55
Figure 3.11–Comparison between output of the model and legacy output - Validation data 1 & 2.	55
Figure 3.12–Comparison between output of the model and legacy output - Validation data 4 & 3.	56
Figure 3.13–comparison between output of the model and legacy data- Estimation data & validation data 1	57
Figure 3.14–Comparison between output of the model and legacy data - Validation data 2 and 4	57
Figure 3.15–Comparison between output of the model and legacy data - Validation data 3	58
Figure 3.16–NARX implementation and training on MATLAB	59
Figure 3.17–NARX results for the validation data (pressure [psi] x time [ds])	60
Figure 4.1 – New Brake Pipe and Aux Reservoir inputs for EST01 - detail of fluctuations.	62
Figure 4.2 – Training and validation data.	63
Figure 4.3 – Targets, outputs, error and response for estimation data & correlation (EST01).	74
Figure 4.4 – Targets, outputs, error and response & correlation for validation data (VAL01).	74
Figure 4.5 – Targets, outputs, error and response & correlation for validation data (VAL02).	75
Figure 4.6 – Targets, outputs, error and response for validation data (VAL03).	76
Figure 4.7 – Targets, outputs, error and response & correlation for validation data (VAL04).	76
Figure 4.8 – Targets, outputs, error and response & correlation for estimation data (EST01).	78
Figure 4.9 – Targets, outputs, error and response for estimation data (VAL01).	78
Figure 4.10–Targets, outputs, error and response & correlation for estimation data (VAL02).	79
Figure 4.11–Targets, outputs, error and response & correlation for estimation data (VAL03).	79
Figure 4.12–Targets, outputs, error and response & correlation for estimation data (VAL04).	80
Figure 4.13–Accuracy over the sweep.	81
Figure 4.14–Accuracy over the sweep.	82
Figure 5.1 – Problem topology and required models.	83
Figure 5.2 – Overall strategy to address the problem.	83
Figure 5.3 – Most accurate ANN for BP+AR->BC.	84
Figure 5.4 – Most accurate ANN for BP+AR->BC on the closed version.	84

Figure 5.5 – Targets, outputs, error and response for estimation data & Target x estimation correlation (EST01).	85
Figure 5.6 – Targets, outputs, error and response for estimation data & Target x estimation correlation (VAL01).	85
Figure 5.7 – Targets, outputs, error and response for estimation data & Target x estimation correlation (VAL02).	86
Figure 5.8 – Targets, outputs, error and response for estimation data & Target x estimation correlation (VAL03).	86
Figure 5.9 – Targets, outputs, error and response for estimation data & Target x estimation correlation (VAL04).	87
Figure 5.10–Targets, outputs, error and response for estimation data & Target x estimation correlation (EST01).	89
Figure 5.11–Targets, outputs, error and response for estimation data & Target x estimation correlation (VAL01).	89
Figure 5.12–Targets, outputs, error and response for estimation data & Target x estimation correlation (VAL02).	90
Figure 5.13–Targets, outputs, error and response for estimation data & Target x estimation correlation (VAL03).	90
Figure 5.14–Targets, outputs, error and response for estimation data & Target x estimation correlation (VAL04).	91
Figure 5.15–Time series for STEP function.	95
Figure 5.16–Model representation for STP -> BP (open loop).	97
Figure 5.17–Targets, outputs, error and response for estimation data & Target x estimation correlation (EST01).	97
Figure 5.18–Targets, outputs, error and response for estimation data & Target x estimation correlation (VAL01).	98
Figure 5.19–Targets, outputs, error and response for estimation data & Target x estimation correlation (VAL02).	98
Figure 5.20–targets, outputs, error and response for estimation data & Target x estimation correlation (VAL03).	99
Figure 5.21–Targets, outputs, error and response for estimation data & Target x estimation correlation (VAL04).	99

Figure 5.22–Model representation for STP -> BP (closed loop).	100
Figure 5.23–Targets, outputs, error and response for estimation data & Target x estimation correlation (EST01).	101
Figure 5.24–Targets, outputs, error and response for estimation data & Target x estimation correlation (VAL01).	101
Figure 5.25–Targets, outputs, error and response for estimation data & Target x estimation correlation (VAL02).	102
Figure 5.26–Targets, outputs, error and response for estimation data & Target x estimation correlation (VAL03).	102
Figure 5.27–Targets, outputs, error and response for estimation data & Target x estimation correlation (VAL04).	103
Figure 5.28–Targets, outputs, error and response for estimation data & Target x estimation correlation (EST01).	105
Figure 5.29–Targets, outputs, error and response for estimation data & Target x estimation correlation (VAL01).	105
Figure 5.30–Targets, outputs, error and response for estimation data & Target x estimation correlation (VAL02).	106
Figure 5.31–Targets, outputs, error and response for estimation data & Target x estimation correlation (VAL03).	106
Figure 5.32–Targets, outputs, error and response for estimation data & Target x estimation correlation (VAL04).	107
Figure 5.33–Model representation for BP -> AR (open loop).	109
Figure 5.34–Targets, outputs, error and response for estimation data & Target x estimation correlation (EST01).	109
Figure 5.35–Targets, outputs, error and response for estimation data & Target x estimation correlation (VAL01).	109
Figure 5.36–Targets, outputs, error and response for estimation data & Target x estimation correlation (VAL02).	110
Figure 5.37–Targets, outputs, error and response for estimation data & Target x estimation correlation (VAL03).	110
Figure 5.38–Targets, outputs, error and response for estimation data & Target x estimation correlation (VAL04).	111

Figure 5.39–Model representation for BP -> AR (closed loop).	112
Figure 5.40–Targets, outputs, error and response for estimation data & Target x estimation correlation (EST01).	113
Figure 5.41–Targets, outputs, error and response for estimation data & Target x estimation correlation (VAL01).	113
Figure 5.42–Targets, outputs, error and response for estimation data & Target x estimation correlation (VAL02).	114
Figure 5.43–Targets, outputs, error and response for estimation data & Target x estimation correlation (VAL03).	114
Figure 5.44–Targets, outputs, error and response for estimation data & Target x estimation correlation (VAL04).	115
Figure 5.45–Targets, outputs, error and response for estimation data & Target x estimation correlation (EST01).	116
Figure 5.46–Targets, outputs, error and response for estimation data & Target x estimation correlation (VAL01).	117
Figure 5.47–Targets, outputs, error and response for estimation data & Target x estimation correlation (VAL02).	117
Figure 5.48–Targets, outputs, error and response for estimation data & Target x estimation correlation (VAL03).	118
Figure 5.49–Targets, outputs, error and response for estimation data & Target x estimation correlation (VAL04).	118
Figure 5.50–Training dataset based on the portmanteau of VAL01 and VAL04. Blue line is the BP pressure and red line is the Aux. Reservoir.	119
Figure 5.51–Targets, outputs, error and response for estimation data & Target x estimation correlation (EST01).	120
Figure 5.52–Targets, outputs, error and response for estimation data & Target x estimation correlation (VAL01).	120
Figure 5.53–Targets, outputs, error and response for estimation data & Target x estimation correlation (VAL02).	121
Figure 5.54–Targets, outputs, error and response for estimation data & Target x estimation correlation (VAL03).	121

Figure 5.55–Targets, outputs, error and response for estimation data & Target x estimation correlation (VAL04).	122
Figure 5.56–Model representation for BP -> AR (closed loop).	124
Figure 5.57–Targets, outputs, error and response for estimation data & Target x estimation correlation (EST01).	124
Figure 5.58–Targets, outputs, error and response for estimation data & Target x estimation correlation (VAL01).	125
Figure 5.59–Targets, outputs, error and response for estimation data & Target x estimation correlation (VAL02).	125
Figure 5.60–Targets, outputs, error and response for estimation data & Target x estimation correlation (VAL03).	126
Figure 5.61–Targets, outputs, error and response for estimation data & Target x estimation correlation (VAL04).	126
Figure 6.1 – Problem topology and required models.	127
Figure 6.2 – Overall strategy to address the problem.	127
Figure 6.3 – Overall simulation outputs for EST01 - STP, BP, AR, BC	134
Figure 6.4 – Targets, outputs, error and response for estimation data & Target x estimation correlation (BP - EST01).	135
Figure 6.5 – Targets, outputs, error and response for estimation data & Target x estimation correlation (AR - EST01).	135
Figure 6.6 – Targets, outputs, error and response for estimation data & Target x estimation correlation (BC - EST01).	136
Figure 6.7 – Overall simulation outputs for VAL01 - STP, BP, AR, BC.	136
Figure 6.8 – Targets, outputs, error and response for estimation data & Target x estimation correlation (BP - VAL01).	137
Figure 6.9 – Targets, outputs, error and response for estimation data & Target x estimation correlation (AR - VAL01).	137
Figure 6.10–Targets, outputs, error and response for estimation data & Target x estimation correlation (BC - VAL01).	138
Figure 6.11–Overall simulation outputs for VAL01 - STP, BP, AR, BC.	138
Figure 6.12–Targets, outputs, error and response for estimation data & Target x estimation correlation (BP - VAL01).	139

Figure 6.13–Targets, outputs, error and response for estimation data & Target x estimation correlation (AR - VAL01).	139
Figure 6.14–Targets, outputs, error and response for estimation data & Target x estimation correlation (BC - VAL01).	140
Figure 6.15–Overall simulation outputs for VAL02 - STP, BP, AR, BC.	140
Figure 6.16–Targets, outputs, error and response for estimation data & Target x estimation correlation (BP - VAL02).	141
Figure 6.17–Targets, outputs, error and response for estimation data & Target x estimation correlation (AR - VAL02).	141
Figure 6.18–Targets, outputs, error and response for estimation data & Target x estimation correlation (BC - VAL02).	142
Figure 6.19–Overall simulation outputs for VAL03 - STP, BP, AR, BC	142
Figure 6.20–Targets, outputs, error and response for estimation data & Target x estimation correlation (BP - VAL03).	143
Figure 6.21–Targets, outputs, error and response for estimation data & Target x estimation correlation (AR - VAL03).	143
Figure 6.22–Targets, outputs, error and response for estimation data & Target x estimation correlation (BC - VAL03).	144
Figure 6.23–Overall simulation outputs for VAL04 - STP, BP, AR, BC.	144
Figure 6.24–Targets, outputs, error and response for estimation data & Target x estimation correlation (BP - VAL04).	145
Figure 6.25–Targets, outputs, error and response for estimation data & Target x estimation correlation (AR - VAL04).	145
Figure 6.26–Targets, outputs, error and response for estimation data & Target x estimation correlation (BC - VAL04).	146
Figure 6.27–Model representation for the new version of STP -> BP.	147
Figure 6.28–Targets, outputs, error and response for estimation data & Target x estimation correlation (EST01).	148
Figure 6.29–Targets, outputs, error and response for estimation data & Target x estimation correlation (VAL01).	148
Figure 6.30–Targets, outputs, error and response for estimation data & Target x estimation correlation (VAL02).	149

Figure 6.31–Targets, outputs, error and response for estimation data & Target x estimation correlation (VAL03).	149
Figure 6.32–Targets, outputs, error and response for estimation data & Target x estimation correlation (VAL04).	150
Figure 6.33–Overall simulation outputs for EST01 - STP, BP, AR, BC.	152
Figure 6.34–Targets, outputs, error and response for estimation data & Target x estimation correlation (BP - EST01).	152
Figure 6.35–Targets, outputs, error and response for estimation data & Target x estimation correlation (AR - EST01).	153
Figure 6.36–Targets, outputs, error and response for estimation data & Target x estimation correlation (BC - EST01).	153
Figure 6.37–Overall simulation outputs for VAL01 - STP, BP, AR, BC.	154
Figure 6.38–Targets, outputs, error and response for estimation data & Target x estimation correlation (BP - VAL01).	154
Figure 6.39–Targets, outputs, error and response for estimation data & Target x estimation correlation (AR - VAL01).	155
Figure 6.40–Targets, outputs, error and response for estimation data & Target x estimation correlation (BC - VAL01).	155
Figure 6.41–Overall simulation outputs for VAL01 - STP, BP, AR, BC.	156
Figure 6.42–Targets, outputs, error and response for estimation data & Target x estimation correlation (BP - VAL01).	156
Figure 6.43–Targets, outputs, error and response for estimation data & Target x estimation correlation (AR - VAL01).	157
Figure 6.44–Targets, outputs, error and response for estimation data & Target x estimation correlation (BC - VAL01).	157
Figure 6.45–Overall simulation outputs for VAL02 - STP, BP, AR, BC	158
Figure 6.46–Targets, outputs, error and response for estimation data & Target x estimation correlation (BP - VAL02).	158
Figure 6.47–Targets, outputs, error and response for estimation data & Target x estimation correlation (AR - VAL02).	159
Figure 6.48–Targets, outputs, error and response for estimation data & Target x estimation correlation (BC - VAL02).	159

Figure 6.49–Overall simulation outputs for VAL03 - STP, BP, AR, BC.	160
Figure 6.50–Targets, outputs, error and response for estimation data & Target x estimation correlation (BP - VAL03).	160
Figure 6.51–Targets, outputs, error and response for estimation data & Target x estimation correlation (AR - VAL03).	161
Figure 6.52–Targets, outputs, error and response for estimation data & Target x estimation correlation (BC - VAL03).	161
Figure 6.53–Overall simulation outputs for VAL04 - STP, BP, AR, BC.	162
Figure 6.54–Targets, outputs, error and response for estimation data & Target x estimation correlation (BP - VAL04).	162
Figure 6.55–Targets, outputs, error and response for estimation data & Target x estimation correlation (AR - VAL04).	163
Figure 6.56–Targets, outputs, error and response for estimation data & Target x estimation correlation (BC - VAL04).	163
Figure 7.1 – Proposed topology with emergency reservoir.	167
Figure 7.2 – Emergency brake validation data	168
Figure 7.3 – Estimation data (EST01 v2) with the added emergency brake and reservoir.	168
Figure 7.4 – Validation (VAL01 v2) with the added emergency brake and reservoir.	169
Figure 7.5 – Emergency reservoir pressure for VAL02, VAL03 and VAL04.	171
Figure 7.6 – Targets, outputs, error and response for estimation data & Target x estimation correlation (BP - EST01) - time in [ds] and values in [psi].	172
Figure 7.7 – Targets, outputs, error and response for estimation data & Target x estimation correlation (BP - VAL01) - time in [ds] and values in [psi].	173
Figure 7.8 – Targets, outputs, error and response for estimation data & Target x estimation correlation (BP - VAL02) - time in [ds] and values in [psi].	173
Figure 7.9 – Targets, outputs, error and response for estimation data & Target x estimation correlation (BP - VAL03) - time in [ds] and values in [psi].	174
Figure 7.10–Targets, outputs, error and response for estimation data & Target x estimation correlation (BP - VAL04) - time in [ds] and values in [psi].	174
Figure 7.11–Targets, outputs, error and response for estimation data & Target x estimation correlation (BP - VAL05) - time in [ds] and values in [psi].	175

Figure 7.12–Targets, outputs, error and response for estimation data & Target x estimation correlation (ER - EST01) - time in [ds] and values in [psi].	177
Figure 7.13–Targets, outputs, error and response for estimation data & Target x estimation correlation (ER - VAL01) - time in [ds] and values in [psi].	178
Figure 7.14–Targets, outputs, error and response for estimation data & Target x estimation correlation (ER - VAL02) - time in [ds] and values in [psi].	178
Figure 7.15–Targets, outputs, error and response for estimation data & Target x estimation correlation (ER - VAL03) - time in [ds] and values in [psi].	179
Figure 7.16–Targets, outputs, error and response for estimation data & Target x estimation correlation (ER - VAL04) - time in [ds] and values in [psi].	179
Figure 7.17–Targets, outputs, error and response for estimation data & Target x estimation correlation (ER - VAL05) - time in [ds] and values in [psi].	180
Figure 7.18–Targets, outputs, error and response for estimation data & Target x estimation correlation (AR - EST01) - time in [ds] and values in [psi].	182
Figure 7.19–Targets, outputs, error and response for estimation data & Target x estimation correlation (AR - VAL01) - time in [ds] and values in [psi].	182
Figure 7.20–Targets, outputs, error and response for estimation data & Target x estimation correlation (AR - VAL02) - time in [ds] and values in [psi].	183
Figure 7.21–Targets, outputs, error and response for estimation data & Target x estimation correlation (AR - VAL03) - time in [ds] and values in [psi].	183
Figure 7.22–Targets, outputs, error and response for estimation data & Target x estimation correlation (AR - VAL04) - time in [ds] and values in [psi].	184
Figure 7.23–Targets, outputs, error and response for estimation data & Target x estimation correlation (AR - VAL05) - time in [ds] and values in [psi].	184
Figure 7.24–Targets, outputs, error and response for estimation data & Target x estimation correlation (BC - EST01) - time in [ds] and values in [psi].	186
Figure 7.25–Targets, outputs, error and response for estimation data & Target x estimation correlation (BC - VAL01) - time in [ds] and values in [psi].	186
Figure 7.26–Targets, outputs, error and response for estimation data & Target x estimation correlation (BC - VAL02) - time in [ds] and values in [psi].	187
Figure 7.27–Targets, outputs, error and response for estimation data & Target x estimation correlation (BC - VAL03) - time in [ds] and values in [psi].	187

Figure 7.28–Targets, outputs, error and response for estimation data & Target x estimation correlation (BC - VAL04) - time in [ds] and values in [psi].	188
Figure 7.29–Targets, outputs, error and response for estimation data & Target x estimation correlation (BC - VAL05) - time in [ds] and values in [psi].	188
Figure 7.30–Assembly 1 (assy 1) model results for VAL01 and comparison between model output and target for BC.	191
Figure 7.31–Assembly 2 (assy 2) model results for EST01 and VAL01.	191
Figure 7.32–Assembly 3 (assy 3) model results for EST01 and comparison between model output and target for BC.	191
Figure 7.33–Assembly 3 (assy 3) model results for VAL01 and comparison between model output and target for BC.	192
Figure 7.34–Assembly 3 (assy 3) model results for VAL02 and comparison between model output and target for BC.	192
Figure 7.35–Assembly 3 (assy 3) model results for VAL03 and comparison between model output and target for BC.	192
Figure 7.36–Assembly 3 (assy 3) model results for VAL04 and comparison between model output and target for BC.	193
Figure 7.37–Assembly 3 (assy 3) model results for VAL05 and comparison between model output and target for BC.	193
Figure 7.38–Assembly 3 (assy 3) model results for EST01 and comparison between model output and target for BC.	194
Figure 7.39–Assembly 3 (assy 3) model results for VAL01 and comparison between model output and target for BC.	196
Figure 7.40–Assembly 3 (assy 3) model results for VAL02 and comparison between model output and target for BC.	196
Figure 7.41–Assembly 3 (assy 3) model results for VAL03 and comparison between model output and target for BC.	196
Figure 7.42–Assembly 3 (assy 3) model results for VAL04 and comparison between model output and target for BC.	197
Figure 7.43–Assembly 3 (assy 3) model results for VAL05 and comparison between model output and target for BC.	197

Figure A.1 – Encounter at Promotory, or the Golden Spike Cerimony - the competition of the transcontinental railway (US... ,)	207
Figure A.2 – The evolution of track quantity and British investment in Brazil (Abreu, 2000) (Histórico... ,)	211
Figure A.3 – Relation between iron ore prices and locomotives built in Brazil (ABIFER... ,) (Mdic... ,)	214
Figure C.1 – Program flow (simulink) for main original brake program	222

LIST OF TABLES

Table 4.1 – Accuracy (error) and time results for the lowest error neural network	73
Table 4.2 – Accuracy (error) and time results for the fastest running neural network.	77
Table 4.3 – Best mean accuracy combination.	80
Table 4.4 – Fastest mean running time combination.	82
Table 5.1 – Open and close loop comparison.	88
Table 5.2 – Accuracy (error) and time results for the most accurate neural network (STP->BP - Open).	97
Table 5.3 – Accuracy (error) and time results for the most accurate neural network (STP->BP - Closed).	100
Table 5.4 – Accuracy (error) and time results for the most accurate neural network (STP->BP - Closed & Retrained).	104
Table 5.5 – Accuracy (error) and run time results for the most accurate neural network (BP -> RA - Open Loop).	108
Table 5.6 – Accuracy (error) and time results for the most accurate neural network (BP -> AR - Closed).	112
Table 5.7 – Accuracy (error) and time results for the most accurate neural network (BP -> AR - Closed; retrained).	116
Table 5.8 – Accuracy (error) and time results for the most accurate neural network (BP -> AR - Closed; retrained 2).	120
Table 5.9 – Accuracy (error) for candidate artificial neural network (BP -> AR - Closed).	123
Table 5.10–Run time for candidate artificial neural network (BP -> AR - Closed).	124
Table 6.1 – Simulation results - performance & run time for simulation program.	134
Table 6.2 – Comparison between the new STP -> BP model and the old.	147
Table 6.3 – Results of the new simulation and previous simulation.	151
Table 6.4 – Run time for original model and proposed model.	165
Table 6.5 – Run time for original model and proposed model.	165
Table 7.1 – Results of the STP -> BP model.	172
Table 7.2 – Results of the BP -> ER model.	176
Table 7.3 – Results of the BP + ER -> RA model.	181

Table 7.4 – Results of the BP + AR + ER -> BC model. 185

Table 7.5 – Summary of results for the combined model with different sub-models. . . . 190

Table 7.6 – Summary of results for the combined model with different sub-models with
retraining. 195

Table 7.7 – Summary times for the various models presented on this work 198

ABBREVIATIONS

Simbols and abbreviations will be presented on demand in the body of the text

CONTENTS

1	Introduction	27
1.1	Objective	28
1.2	Structure of the Work	29
2	Basic Concepts	30
2.1	Typical brake commands	32
2.2	Types of automatic brake valves	34
2.2.1	AB	34
2.2.2	ABD	35
2.2.3	ABDX	35
2.3	Current Works on Brake Simulation	37
2.4	Trends on Machine Learning Applications on the Railway and Related Areas	38
3	Materials & Methods	41
3.1	Inputs for Linearity Assessment	43
3.2	Methods Used	46
3.2.1	ARX (Auto Regressive eXogenous)	47
3.2.2	Transfer Function Estimates	48
3.2.3	Model search algorithm for TFEST and ARX	51
3.2.4	NARX	52
3.3	Method Selection	54
3.3.1	ARX	54
3.3.2	Transfer Function	56
3.3.3	Discussion	57
3.3.4	NARX	58
4	Open Loop Model Exploration	61
4.1	Redefining the Inputs	61
4.2	Changing the Parameters	63
4.3	Training Function	64
4.4	Assessing Accuracy	67
4.5	Parameter Sweep	68

4.6	Parameter Sweep Results	72
4.6.1	Best Accuracy	73
4.6.1.1	EST01	73
4.6.1.2	Validation data	74
4.6.2	Assessing for Run Speed	77
4.6.2.1	EST01	77
4.6.2.2	Validation Datasets	78
4.6.3	Best Combination between speed and accuracy	80
4.6.4	Other results	80
4.6.4.1	Overall Performance	80
4.6.4.2	Overall running time	81
5	Model Deployment	83
5.1	Brake Pipe + Aux. Reservoir to Brake Cylinder: BP + AR -> CF	83
5.1.1	Closing the Neural Network	83
5.1.2	Training the Network Closed BP + AR -> BC	88
5.2	From Brake Handle to Brake Pipe: STP -> BP	92
5.2.1	Command Matrix	92
5.2.1.1	EST01	92
5.2.1.2	VAL01	92
5.2.1.3	VAL02	93
5.2.1.4	VAL03	93
5.2.1.5	VAL04	93
5.2.2	Algorithm to Build Step Time Series	96
5.2.3	STP -> BP Model Exploration - Open Loop	96
5.2.4	Closing the Loop	100
5.2.5	Retraining the Closed loop Model	104
5.3	Brake Pipe & Auxiliary Reservoir Interaction: BP -> AR	108
5.3.1	BP -> AR Model Exploration - Open Loop	108
5.3.2	Closing the loop	112
5.3.3	Retraining the Closed Loop Model	116
5.3.4	Closed Loop Retraining 2	119
5.3.5	BP -> AR Closed Loop Model Exploration	123

6	Simulation	127
6.1	Model Implementation	127
6.1.1	Functions	127
6.1.1.1	Matrix to Step	127
6.1.1.2	ANN execution	128
6.1.1.3	Performance Evaluation	129
6.1.2	Main Program	130
6.2	Simulation Results	133
6.2.1	Revisiting STP-BP	147
6.2.2	Simulation Results Redux	151
6.3	Comparing the Proposed with the Original Model	164
7	Emergency Reservoir and Brake	166
7.1	Inputs	167
7.1.1	VAL05 - Emergency Brake	167
7.1.2	EST01 - Estimation Data	167
7.1.3	VAL01 - Validation Data	169
7.1.4	Other Validation Data	170
7.2	Training the Models	171
7.2.1	STEP -> BP	171
7.2.2	BP -> ER	176
7.2.3	BP +ER -> AR	181
7.2.4	BP + AR + ER -> BC	185
7.3	Combined Brake Model	189
7.4	Alternative Combined Brake Model	194
7.5	Combined and Service Model Run Time Comparison	198
8	Conclusion	199
	Bibliography	201
	APPENDIX A Introduction to the history of railways	206
A.1	The Brazilian Case	208
A.1.1	The Empire	208
A.1.2	The economics of rail	210

A.1.3	Revolution	211
A.1.4	Nationalization	212
A.1.5	Privatisation & Ressurgence	213
A.1.6	Future perspectives	213
APPENDIX B	Computer Characterisation	215
APPENDIX C	Original Program Listing	216
C.1	Run_Freio_V01	216
C.2	Freio_V02	222

1 INTRODUCTION

Rail transport is characterised by two main factors: It is guided, i.e. the rails guide the vehicle and the consist tend to be long and interconnected. In the specific case of heavy haul transport, compositions tend to have more than 50 vehicles with one or more traction units, or locomotives. The cost of building a rail network is very high. However, once the tracks are laid the challenge is to improve efficiency by transporting the largest amount of cargo with less energy and time.

The travel of the vehicles over the track is quite tricky and complicated. Wagons -that are unpowered vehicles - have very peculiar damping and dynamics and, specially on grain applications, the centre of gravity can be quite high, particularly for metre gauge tracks. There are other concerns as well, such as considerations of maximum (and minimum) turn speeds due to superelevation, forces on the shock absorption apparatus, power required to increase or to reduce speed, braking performance, etc.

In order to improve the capacity of cargo transported on the same infrastructure, rail operators rely on the increase of the number of compositions on the network and the increase of their speed. Though studies can be done using the compositions themselves - what is popular among some professionals - they are safer if done using simulators.

A proper simulator for the longitudinal dynamics must contain several sub-modules such as the shock apparatus behaviour, inertia of the system, energy balance for acceleration, deceleration, etc. One of such modules is the brake behaviour, that can be divided into two parts, the brake valve and the pressure flow on the consist.

Current state of art in this area uses complex Navier-Stokes equations to describe the pressure and volume flows ([Teodoro, 2016](#)). This approach yields good results. Nonetheless the fact that the Navier-Stokes equations do not have analytical solutions, the model had to resort to numerical integration. In order to maintain the numerical stability and assure convergence, the time steps have to be quite short, what with current technologies, implies on a long running time.

The extra time required to run the brake performance algorithm increases the time required for optimisation running the simulator and thus reduce its usability. Therefore, there is an opportunity to reduce the running time by using a mathematical model to replace the Navier-

Stokes model. If the longitudinal dynamics models can run faster with comparable precision, there are more opportunities to try different optimization techniques or even try to improve the model further. A longitudinal dynamics simulation that runs faster can also be used to train conductors, among other applications.

This trend in machine learning and making the models more nimble and suited to specific applications is part of a broader revolution. At the time of this writing, there are two competitive models for autonomous train conduction. Neither applies the brakes without the engineer input. An addition such as the one presented may help in improving such commercial models.

Moving forward, models can be done to take into account the wear and tear of the systems and/or assessing each consist with a particular behaviour based on each individual consisting vehicle. Further, the models can learn and adapt to different conditions on the fly. Cases such as rain, difficult adhesion and even cold weather air leaks could potentially be identified and modelled by a system that not only is aware of its environment and components but also improves without external inputs.

Locomotives and circulation could also benefit from machine learning models. The vehicle flow depending on the different boundary conditions - such as neighbouring traffic, rain, the proximity of a school to the tracks, etc. - could be learned and the schedule could be nudged to improve safety and efficiency. On the locomotives, models for the wear of components and technologies such as digital twins or sensor fusion to improve performance could be using to run them cheaper and for longer, reducing maintenance and fuel consumption.

This work presents a small part of a set of tools that promise to change the railways and other applications in the near future.

A brief history of the railways in Brazil can be found on appendix 1. It tells the conditions that allowed its growth and points some possibilities to explain its reduction in size - but not importance.

1.1 Objective

This work aims to develop a fast and accurate mathematical approach to overcome the current Navier-Stokes method for applications in simulators, employing artificial intelligence tools.

1.2 Structure of the Work

This work starts with an introduction to the subject and an overall panorama of the the relevance and the objective.

The second chapter reviews the basic concepts what are going to be used, background and the physical implementation of the problem

Chapter three presents the inputs considered for the linearity evaluation as well as reviews the methods to be used on the problem modelling. It also presents the results of the model application and selects the proper method to be used on the rest of the work.

Chapter four redefines the inputs for the ADBX valve instead of the AB used on the previous chapter and explores the impact of the number of delays and neurons on the training results of an open neural network.

Chapter five closes the neural network and prepares for its deployment as well as addresses the issues that have risen with closing the artificial neural network.

Chapter six implements the service brake simulator and solves issues with the model when the individual parts are assembled. The chapter also contains the comparison between the original model and the proposed model.

Chapter seven updates the models with the emergency brakes and again compares the performance of the different models.

Chapter 8 contains the conclusion and next possible works.

2 BASIC CONCEPTS

Consists consist in two main types of vehicles, wagons and locomotives. The locomotives are the powered vehicles that provide the force to start the movement, keep it and also provide part of the braking effort. Trains have three main braking means, two on the locomotive and one on the wagons. For braking, the locomotive can use the Dynamic Brake - when it turns the traction motors into generators and exerts a torque contrary to movement. This is used to control and adjust the speed during travel and is the preferred way to do so because it not only reduces the wear on the brake shoes but also allows for a finer speed adjustment. The other brake system on the locomotive is the air brake on the wheels, that acts similarly to the ones on the wagons - that will be explained shortly - and is seldom used during normal travel, being reserved for when the locomotive is alone or for parking.

Wagons consist of a container - where goods are transported-, wheel base assembly - wheels, suspension, damping, etc. - shock apparatus, and the pneumatic brakes - our main interest on this work. The standard brake system on the heavy haul freight wagon consists on a brake pipe (EG) that acts as pressure connection between the vehicles on the consist. It acts as both communication - for an increase or decrease in pressure will activate the automatic brake valve - and as pressure source to feed the system. The brake pipe is connected at each end to the other vehicle on the composition via flexible hose and a cut out cock. In general, these pipes are made of schedule pipe of 1 1/2". The Automatic Brake Valve is connected to the brake pipe, the auxiliary reservoir (RA), the emergency reservoir (RE), the brake cylinder (CF), the relief retention valve and the load/unloaded valve. A basic topology is presented on figure 2.1

There are many types automatic brake valves, henceforth called valve. Its basic functionality is to compare the pressure of the brake pipe and the aux reservoir and allows pressure to flow from the former to the brake cylinder, actuating the brake pad and causing an extraction of power from the wheel in case the brake pipe has a lower pressure than the aux reservoir. This is called service brake application. If the pressure on the brake pipe raises more than 2 psi, the valve releases the pressure of the brake cylinder, removing the force, and starts to fill the aux reservoir. This is called brake release. If the valve sensed the rate of pressure drop on the brake pipe to be higher than a designated threshold, it then applies the maximum force to the cylinder in what is called "emergency application". The particularities of the types

of automatic brake valves are going to be discussed on a specific section.

The cylinder then exerts force in a set of levers that in turn force the brake shoe into the wheel. This system has some peculiarities, such as the resilience of the lever assembly that may reduce or increase the force applied or the course of the cylinder, altering its internal volume and thus pressure. A representation of this system can be seen on figure 2.2

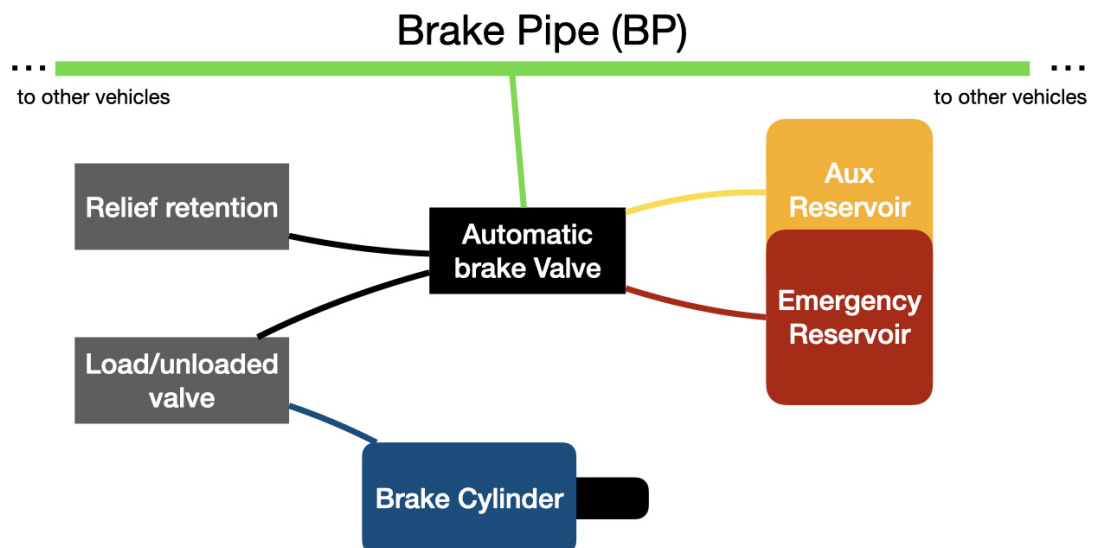


Figure 2.1 – Basic topology of heavy haul freight wagon brake system

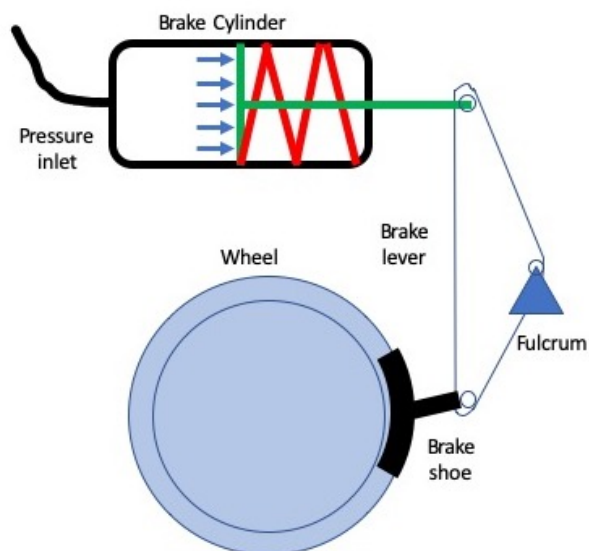


Figure 2.2 – Basic topology of heavy haul freight wagon brake system

The brake pipe responds to the command of the conductor. The contemporary brake systems are in most part computer aided and have a similar function as the 26L system from

NYAB - New York Air Brake Co. The automatic brakes lever is integrated on the control stand - that control the brake on the wagons via brake pipe. The independent brake lever is located on the lower position on the control stand and commands the air brakes for the locomotive.

The conductor positions the automatic brake lever at the desired service position and through a set of valves, actuators, control volumes and sensors, the computer sets the pressure of the brake pipe at the pressure commanded by the lever. So, the locomotive brake system translates a STEP input into a pressure decrease on the brake pipe. There are cases where locomotives are remotely controlled. In such, the leader locomotive sends signals to properly set other locomotives in the consist that apply the brakes accordingly. Figure 2.3 shows the basic topology of the locomotive automatic brake system.

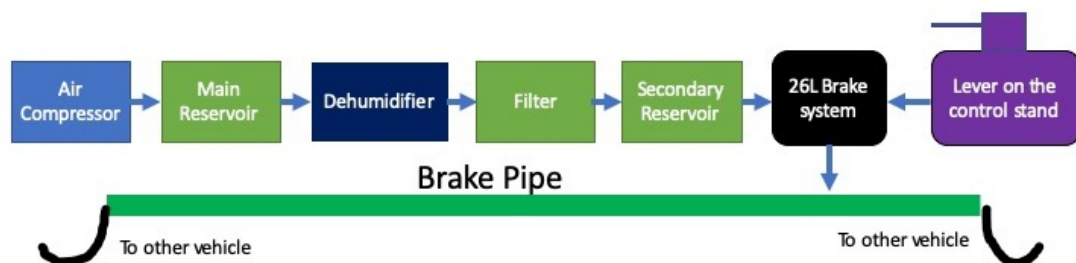


Figure 2.3 – Basic topology of locomotive automatic brake system

The pressure then travels through the pipe and the consist - what can be a quite long travel, from 50 to 100 wagons, what translates to about 1000 m and, at each vehicle or couple of vehicles, the automatic brake valve senses the pressure difference between the brake pipe and the brake cylinder. The valve then applies the pressure from the aux reservoir to the brake cylinder, depleting the former and increasing the pressure on the latter, as mentioned before.

Though very important, the effect of the delay caused by the travel time of the pressure wave and associated chocks will not be part of this study at this point in time. It is believed that such effects can be addressed at a later date with little change to the program and the method chosen.

2.1 Typical brake commands

This commands represent positions on the valve 26-C (or equivalent on the CCBII - Electronic brake system) that is the handle on the control stand (figure 2.4).

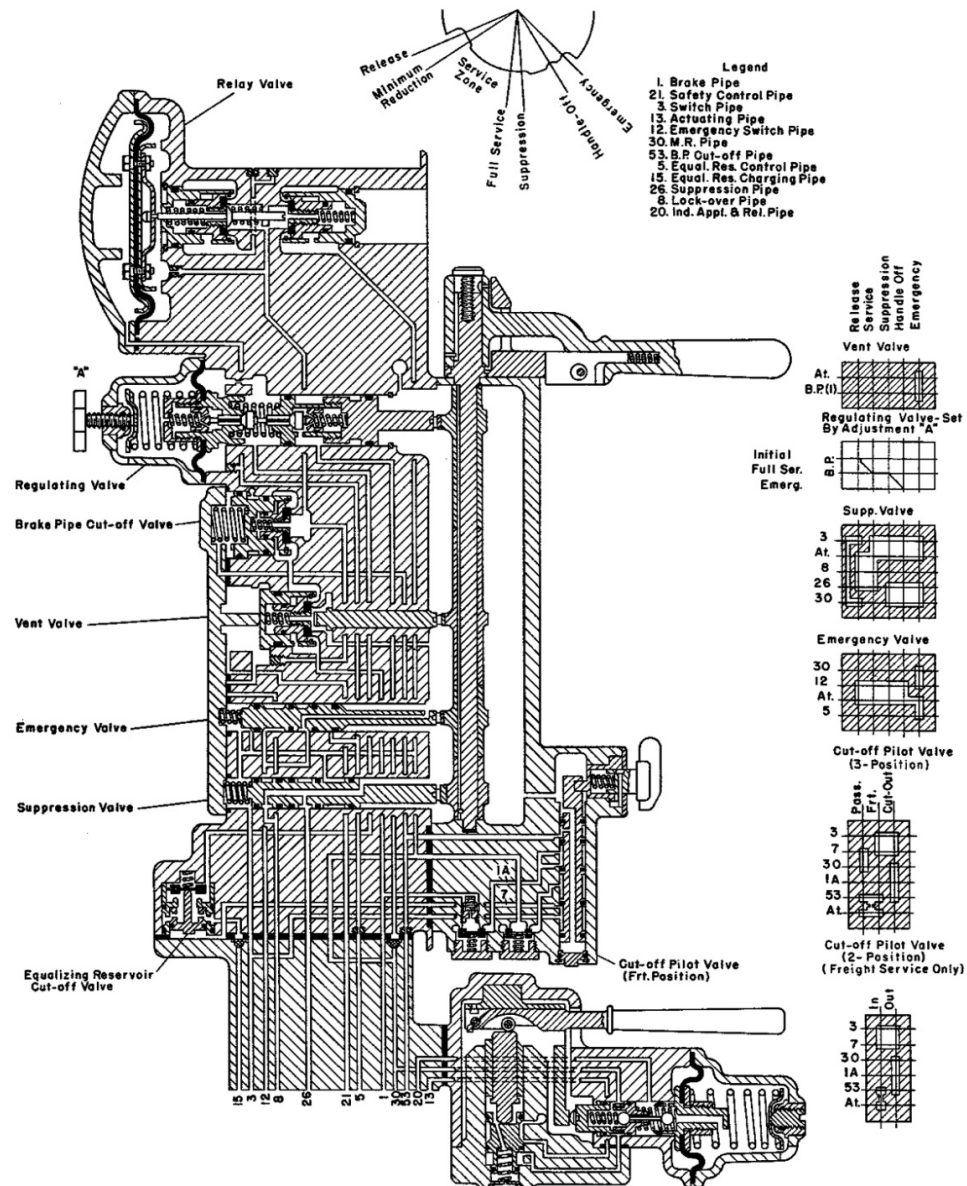


Figure 2.4 – 26-C valve that commands the 26-L brake system (NYAB, 1964)

- **Release / Running:** when the system starts and the locomotive(s) provide air to the various reservoirs and equalise all pressures at the designated running pressure - typically 90 psi. Once charged, the brakes will be ready for operation and will be on the release position. Note: When the brake system reservoirs are at ambient pressure the brake shoes are not applied and the consist is on a non brake position. On the absence of pressure on the brake system reservoirs, the consist must be on a plane and/or the parking brakes must be applied.
- **Minimum Reduction position:** minimum brake pressure is obtained on the brake pipe and

a minimum application of pressure on the cylinder, typically, 6 psi.

- Service position: Increase on the pressure reduction on the brake pipe and increase on the applied force on the brake shoe by the brake cylinder. Once the pressure decrease is achieved on the brake valve, lapping occurs and there is no more air flow between the brake pipe and the reservoirs and brake cylinder, keeping the pressure stable. This occurs when the brake pipe pressure and aux reservoir pressure are the same.
- Suppression position: It applies full service brake - typically 64 psi on the brake pipe - there is also a suppression of the overspeed control and safety application.
- Handle-off position: the handle can be removed on this position (or locked in place using a pin). It is the position to be used on trailing units on a multi unit consist or locomotives being towed "dead". This apply full service brakes. In Brazil, the brake handle shall be placed in this position when the conductor leaves the cabin. Brazil uses a single person operation regimen, i.e. there is no helper, just the conductor.
- Emergency position: this applies the emergency brake to the consist. Emergency application commands higher pressure to the cylinder. Care shall be taken to prevent the wheel from locking, in which case, the braking effort will be reduced due to wheel slip. There are other events that may cause the application of the emergency brakes, such as the conduction companion, when present, may activate the valve located in the cabin or if there is a break on the train that causes the pressure on the brake pipe to fall at a velocity sufficiently high to apply the emergency brakes.

2.2 Types of automatic brake valves

There are three main automatic brake valve models in use in Brazil.: AB, ABD and ABDX. The DB60 valve is equivalent to the ABDX. Figure 2.5 shows a typical location of an automatic brake valve on a freight car.

2.2.1 AB

The AB valve is the most simple of those that work with the pressure difference between aux reservoir and brake pipe (figure 2.6). It connects the brake pipe with the freight car components (reservoir and brake cylinder) and has three sections. The centre section that acts



Figure 2.5 – Typical location of the valve on a freight car.

as inlet and outlet and contains the emergency equalising reservoir. The emergency portion that monitors the pressure gradient comparing with the rate of pressure decrease of the emergency equalising reservoir to decide if the pressure change is an emergency application and also identify the relief by the increase in pressure. This is done by the movement of the internal parts of emergency control valve and its internal pressures. The service portion controls the service brake applications as well as the brake relief by the movement caused by the pressure differentials on the service control valve. The service portion also has the relief retention valve that assures the release of the brakes

2.2.2 ABD

The ABD valve is the evolution of the AB valve (figure 2.7). The control valves are now on the vertical position rather than on the horizontal. This has greatly reduced unwanted applications - especially of emergency brakes - due to longitudinal pressure shocks. The basic functions of the valve were kept the same with the exception of the relief assurer that was altered to connect the emergency reservoir to the brake pipe, increasing the pressurisation speed of the later and thus increasing the propagation speed of the relief signal on the consist

2.2.3 ABDX

Again the ABD valve has evolved into the ABDW and then the ABDX valve. The ABDW valve was not well received in Brazil and all were replaced or converted to ABD or ABDX. The main improvement of the ABDX valve (figure 2.8) is to enhance the pressure drop

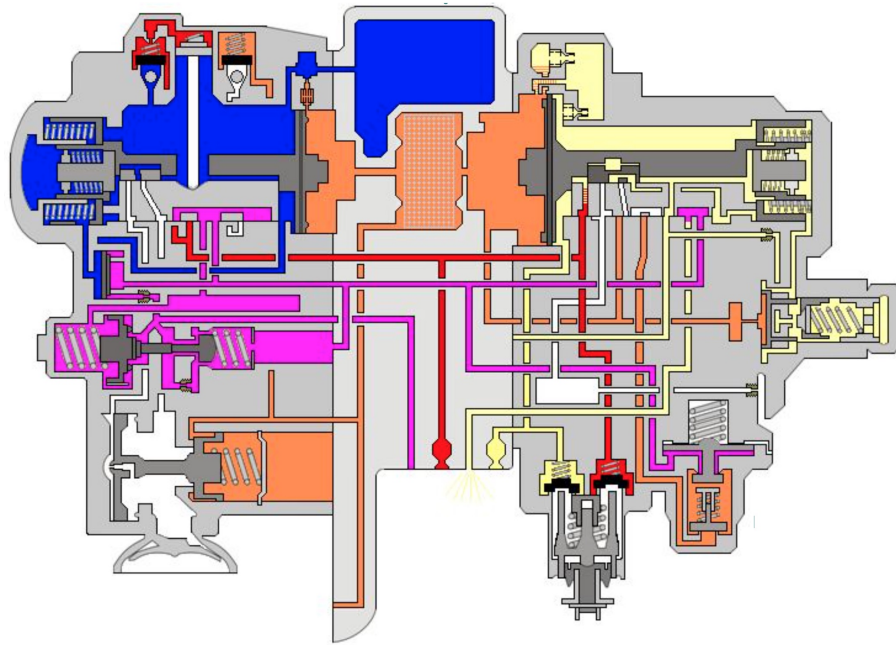


Figure 2.6 – Cross section of the AB automatic brake valve (Corrêa, a). On the left is the emergency portion and on the right the service portion.

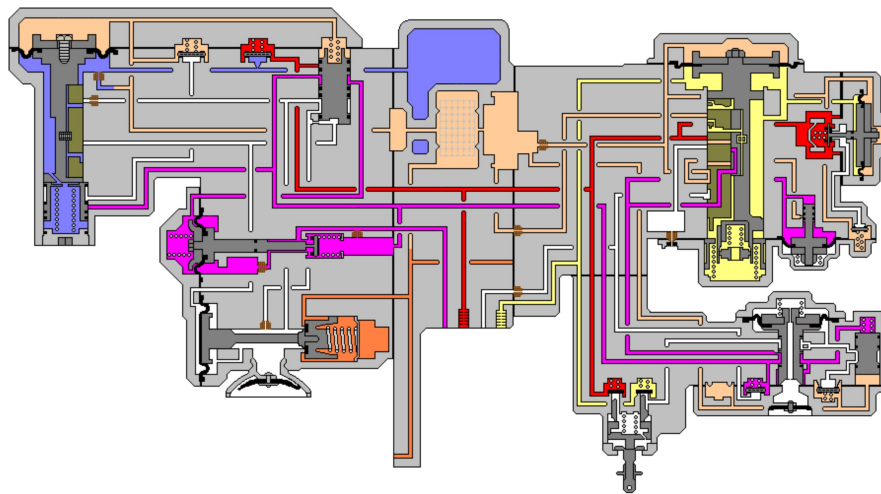


Figure 2.7 – Cross section of the ABD automatic brake valve (Corrêa, b). On the left is the emergency portion and on the right the service portion

on the brake pipe during the brake application. The emergency control valve was modified to allow the brake pipe to be connected to the atmosphere for a brief moment reducing the time to achieve the desired pressure and thus improving the pressure signal propagation speed through the train

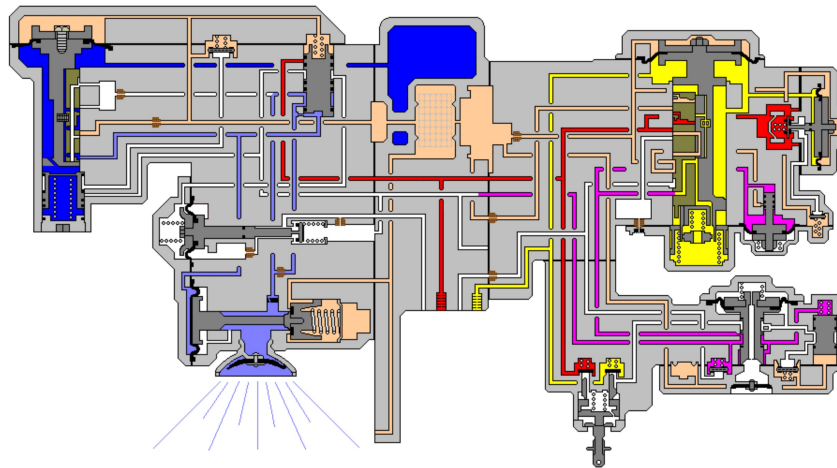


Figure 2.8 – cross section of the ABDX automatic brake valve (Corrêa, c). On the left is the emergency portion and on the right the service portion

2.3 Current Works on Brake Simulation

The understanding of the brake system improves the reliability and precision of the train calculations, allowing new conduction routines, accident investigation, reduction of transit time and other desirable investigations

The current state of the art of longitudinal train simulation models uses an automatic brake valve algorithm based on orifices. The model simulates the pressurisation of the system assuming atmospheric air to be a perfect gas and uses orifices to control and represent the mass movements on the system and adjusting the model to reflect reality.

In previous works, Ribeiro (Ribeiro, 2017) adopted an orifice model based on mass flow and pressure drop between reservoirs and valves and the system, initially proposed by Pugi (Pugi *et al.*, 2015). Teodoro (Teodoro, 2016) has proposed an improvement on the previous method using finite volumes to model the compressed air flow through the system on a more precise and detailed simulation with less simplifications.

Both methods use iterative numerical models and require time steps to be sufficiently small in order to achieve stability, what may lead to long execution times. Teodoro *et al* (Teodoro *et al.*, 2019), have been working on reducing the calculation time by implementing parallel computing both on the CPU and GPU.

The actual implementation of the algorithms for high performance simulation purposes is done by calculating tables offline and porting these into the overall code. This way, there is no substantial time impact. However, this workaround reduces the versatility of the sys-

tem and also creates a cumbersome maze of conditional statements that is not optimal and may also increase the computational time. Nonetheless, the current methods present a good agreement with the benchmark data provided by VALE, that sponsor the projects for the development of brake models at UNICAMP.

An alternative to this approach is to create a mathematical concoction, that describe the automatic brake valve behaviour. This way, it is possible to combine the speed of the simplified model from Ribeiro (Ribeiro, 2017) and the precision of the finite volume model from Teodoro (Teodoro, 2016) avoiding the tangle of conditional programming and the time step limitation of the iterative numerical solvers, improving stability and allowing the codes to run faster and precise.

For this, several seminal papers are listed on the reference section and serve as guidance for the task of building this model (YOUNG; JAKEMAN, 1980) for the transfer function estimates, (Ljung, 1987) and (Ljung; Soderstrom, 1983) for the overall knowledge of system identification, (Forssell; Ljung, 1999) for the closed loop identification used on the linear models. These, however these may not be adequate for the problem of interest for they approach the matter in a linear fashion.

Thus, in order to correctly represent the automatic air brake valve, one may have to resort to non linear strategies. Though there are many options, NARX models based on neural networks have been chosen for they seem to yield good results on these contexts. This decision is based on the material available and proximity with the ARX method, as well as the availability of the techniques

Also, the NARX applied to a non linear time series (Jr.; Barreto, 2006) and as a black/grey box modelling strategy (Barbosa *et al.*, 2011) and its parent method NARMAX are quite versatile and may serve as a stepping stone for the modelling of locomotive parameters, such as turbocharger speed, water pump pressure, etc. that shall be subject of future works

2.4 Trends on Machine Learning Applications on the Railway and Related Areas

In the last years, work has been done on the application of the latest techniques of machine learning to rail applications. The field is by no means consolidated and there is still a lot of development happening at universities and businesses. However, the machine learning approach as well as other system identification techniques are sufficiently mature to be applied to real life problems and their results to be evaluated in terms of pertinence and accuracy.

One such work has done by Li et al ([Li et al., 2014](#)) to improve travel speed and predictive maintenance.

In terms of rail operation, neural networks are being used to foresee delays and allow for the traffic controllers to reorganise schedules to reduce overall impacts ([Martin, 2016](#)). An interesting study by Prokhorchenko et al. ([Prokhorchenko et al., 2019](#)) uses artificial neural networks to predict the time of arrival of cargo dispatched by rail in a non schedule system - what is quite common in the Brazilian rail networks. Wen et al ([Wen et al., 2019b](#)) use Long Term Short Memory (LSTM) model - a Recurring Neural Network (RNN) with limited memory to predict delays on a Dutch railway line. On a subsequent paper, the same author investigates data driven models for train dispatch ([Wen et al., 2019a](#)) and concludes that machine learning shows potential in the field. There are also studies that use Feed Forward Back Propagation Neural Networks (FFBNN) associated with Artificial Bee Colony (ABC) algorithms to predict the position of moving vehicles ([BabyAnitha; Duraiswamy, 2014](#)) that could be used on intricate rail networks or could be simplified to be used on reduced complexity networks.

Carvalho et al. ([Carvalho et al., 2019](#)) provide an interesting and contemporary overview of the use of machine learning in predictive maintenance. Coraddu et al. ([Coraddu et al., 2015](#)) uses artificial intelligence and sensor fusion in the maintenance of complex machines and overall CBM (Condition Based Maintenance). This could be applied to the locomotives engines and motors in rail service. Sysyn et al. ([Sysyn et al., 2019](#)) uses inertial sensors on the bogies to assess the remaining life of the components. Black box modelling is being used to assess abnormal noises on permanent magnet motors used on electric vehicles ([Ma; Zuo, 2014](#)). This could be potentially used to identify premature bearing failures using the already installed wayside infrastructure. Also, if modified could be used to assess locomotive traction motor health using either on board sensors or the wayside instrumentation.

Another application of machine learning on the railways is the use of 2D convolution neural networks and vector support machines to identify wheel defects ([Krummenacher et al., 2018](#)).

There are also several applications of machine learning vision based strategies. On a paper the author proposes a visual identification setup to inspect the rail and crossties using a multitask learning framework and find defects on the various hardware with a limited training set ([Gibert et al., 2017](#)). Several papers followed, focusing on the identification of defective or missing fasteners - that may pose a serious danger to the operation of a railway - an inadequate

fastener may cause a derailment that, depending on the conditions or the transported goods may cause large scale accidents. One such paper uses machine vision to identify the fasteners and the missing ones ([Prasongpongchai et al., 2017](#)) . Taheri et al ([Taheri et al., 2019](#)) uses successfully an expert system to identify the fasteners, though with less efficiency than the machine vision approach. Liu et al. ([Liu et al., 2019](#)) propose an approach with minimum annotated fastener templates. They use a special a convoluted set of cameras to achieve excellent results.

Machine vision has also been used to increase user safety on platforms with the use of cameras and lasers to engage alerts or even brake trains when the tracks are violated ([Scholz, 2016](#)).

There are works focused on the rails, specifically the geometry, the Lasisi and Atttoh-Okine uses a neural network to evaluate the TQI - track quality index - and another neural network to identify defect predictors ([Lasisi; Atttoh-Okine, 2018](#)). On a more recent work, they propose the use of an unsupervised learning framework to connect the TQI with safety indexes ([Lasisi; Atttoh-Okine, 2019](#)).

Defect prediction on the tracks is one aspect of the machine learning application, another is the scheduling of the repair as proposed by Gerum et al. ([Gerum et al., 2019](#))

Consilvio et al. ([Consilvio et al., 2020](#)) propose a generic framework based on machine learning to manage the upkeep of assets of the railways - in their case study, earthworks and track circuits.

There is also an interesting trend on using machine learning to simulate multi body dynamics (MBD). Taheri and Ahmadian used a stochastic model to improve the dynamic simulation of a rail vehicle with interesting results ([Taheri; Ahmadian, 2016](#)). Tsunashima ([Tsunashima, 2019](#)) dynamic simulations, instrumentation and machine learning to detect track defects using car-body vibrations.

As it can be seen by the extract of papers, there are several interesting applications of machine learning techniques on rail problems and it is expected that, as the knowledge evolves, more applications will arise.

3 MATERIALS & METHODS

Current train air brake models use Navier-Stokes equations that have to be solved using small step numerical integrations. This hinders the speed of current simulators and increase the time to run, what reduces the usability of certain iterative optimisation methods such as genetic algorithms. Also, another important use of simulators is to guarantee that safety is achieved during the travel of new train models or on permanent ways with different characteristics. Speed is also desirable in this case, for several iterations have to be run in order to create "safety envelopes" for the various parameters.

The brake topology for this study is comprised of three main models. The one that converts the step pressure commanded on the brake lever and is translated to a pressure drop on the brake pipe by the 26-L system. A second model from the automatic brake valve that correlates the pressure on the brake pipe to the auxiliary reservoir and a third model that correlates the pressure drop on the brake pipe and on the aux reservoir to the increase in pressure on the brake cylinder. Figure 3.1 shows the proposed topology. At this point, the model does not take into account the position of the vehicle on the consist nor the pressure on the emergency reservoir. Once the application of the methods is stablished, there will be little to no trouble in adding these features to the model and training it again.

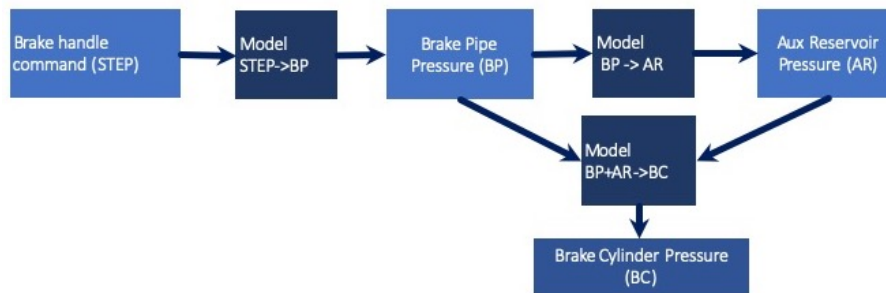


Figure 3.1 – problem topology and required models

The strategy to address the problem is as follows (figure 3.2): first the problem will be tested with two linear system identification approaches - ARX and Transfer Function estimate - and a non linear machine learning approach - NARX. The results will then be evaluated and the most appropriate method will be further investigated through permutations and evaluating what best parameter combination is the most suitable for the problem. Later, a program to simulate the overall problem will be crafted and tested against the current model for speed and

accuracy. Initially, in order to expedite the evaluations, the third model - between brake pipe (BP) and auxiliary reservoir (AR) and the brake cylinder (BC) - will be used as a benchmark in the assumption that it is the most complex - two inputs and one output (MISO). The other models have a single input and output (SISO). Once the appropriate models and parameters are chosen, the two remaining models will be made, deployed and compared with the current Navier-Stokes solution

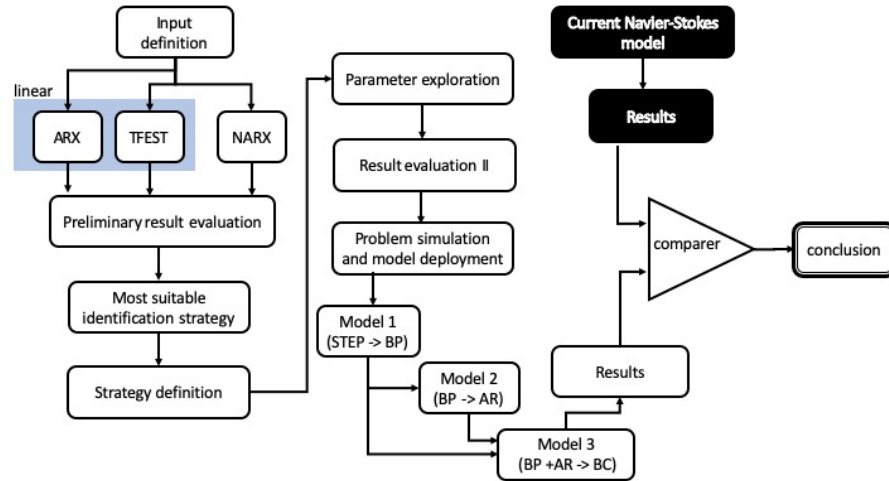


Figure 3.2 – overall strategy to address the problem

The first task at hand is to determine whether the system is linear or not - or if it can be linearized with sufficient complexity.

A linear system is one that the output is a superimposition of the inputs. In nature - and engineering by extension - no system is purely linear when seen close enough. Nevertheless, most systems may be represented as linear, what makes most calculations easier and the results acceptable.

A possible way to verify if a system is linear is to identify its governing equations and then run with a subset of the data and see if it responds accordingly - within a margin of error. If the the system equations are able to properly represent an envelope of data and values properly but not data outside of it, one can still use and assume the model to be linear for that envelope. However, if the boundaries of the area of interest cannot be properly represented by the same model, the system cannot be assumed linear.

The first data set used to describe the system was made with the AB valve representing the pressure of the brake cylinder (BC) with the brake pipe (BP) and auxiliary reservoir (AR) as inputs. This input and output combination was chosen for it is assumed to be more

complex than the models described earlier.

The first linearization attempt was made using Auto Regressive eXogenous (ARX) linear models. After this assessment, transfer function models based on Ljung's article ([Ljung, 2009](#)) were used to try to improve the system identification - and check if the linearization results were model related or problem related.

Since it is necessary to understand the space one is working on in regards to the identification methods, a parameter sweep was performed using these two techniques. The upper boundary is proposed as the limit where the computational effort of this approach is deemed too long.

With the proposed boundary conditions - linear, sufficiently complex and computationally nimble - the purpose is to check which model can more accurately represent the physical system at hand and identify if it is linear.

For the non linear attempts, the same rationale is used varying the number of neurons. A more detailed outline will be shown on the next section

Errors are presented in relation to the estimated pressures.

3.1 Inputs for Linearity Assessment

The initial dataset was provided VALE and is based on their resident legacy software called TOS, from Alion Science ([Ribeiro, 2017](#)) and is assumed to have been validated several years ago. The same dataset was used at the work of Teodoro ([Teodoro, 2016](#)). This data was obtained by tracing the graphs plotted by the program into paper and measuring the points with a ruler.

The AB automatic brake valve was chosen with the rationale that if the model is able to comply with the behaviour of this configuration, it will also be able to comply with the other - valves and vehicle numbers - provided properly instructed. The AB valve is still quite popular on the freight fleet. If the model behaves accordingly, more modern valves will be used on the latter part of this work

The normal operation of the brake valve consists into the translation of the pressure signals into braking force on the car wheel. The valve compares the pressure of the BP with the AR (Auxiliary Reservoir) and applies the brakes according to the pressure difference. If the pressure of the brake pipe is increased by at least 1.5 psi, the valve releases the pressure on the brake cylinder and fills the aux reservoir with the pressure from the brake pipe.

For the evaluation of the linear identification methods proposed at this stage, a "training - or modelling- data package" was concocted. This consists of several brake applications and releases of both maximum and minimum intensity. The original pressure plateaus were expanded in order to assess the model stability. Initially, the applications were very brief. This expansions were made respecting the system transient and dynamic behaviour, i.e. only the fully developed and stable plateaus were lengthened.

Figure 3.3 demonstrates the datasets used for the modelling. The top portion of the figure is the output and the bottom are the two inputs. Note the relation between both inputs. Though they have an intrinsic relation between themselves, it was decided that they will be kept independent for the time being.

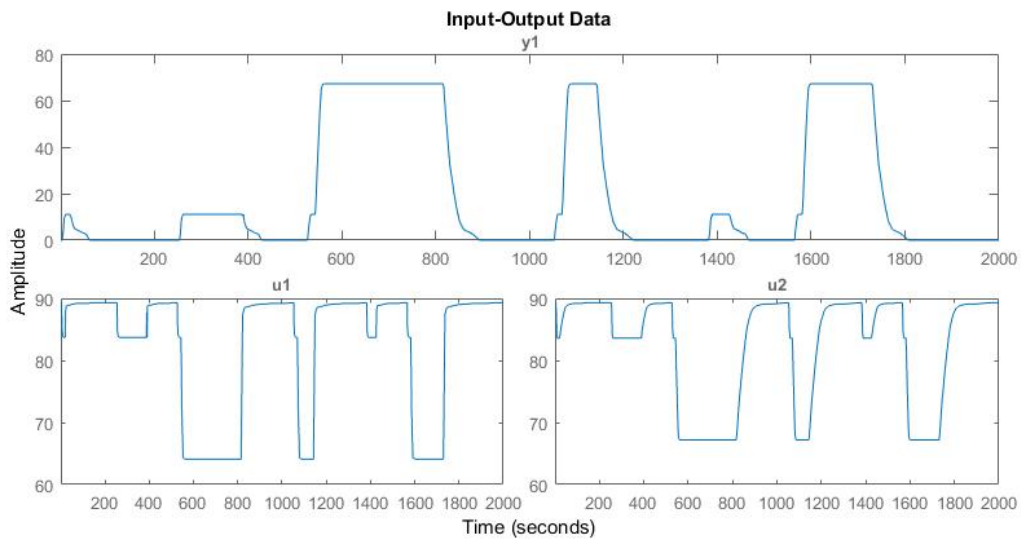


Figure 3.3 – Modelling data : Brake Cylinder (top) Brake Pipe & Auxiliary Reservoir (bottom) horizontal values are time [ds] and vertical values are pressure [psi]

In addition to this, several other datasets were concocted for model validation. VAL01 is analogous to the modelling data, on different order of applications and releases. Data VAL02 and VAL04 are minimum brake applications with different plateau lengths. The aim of this is to train the system for different applications and assess if the system stabilises adequately after applications. Finally, VAL03 is a single maximum brake application. The comparison between VAL02 and VAL03 is crucial in the evaluation of ability of the proposed methods to linearize the problem at hand. In other words, to assess if the identified model is sufficiently complex to linearly represent a possibly non linear system.

Initially, the data available did not contain equally spaced steps in time. i.e. on the transitions, the steps had an interval of seconds, whereas on the plateaus the steps were several

seconds apart. This is a telltale of the graphic tracing method used to obtain this data. Though the values are constant on the plateaus by definition, the methods require that the step time difference to be constant. Therefore, the time series was rearranged around this requirement by extending the plateaus with constant pressure values for the missing steps, assuming them constant. The nature of the available data is discrete and the time step chosen was 1s. Figures 3.4 to 3.7 are the graphical representation of the additional test cases. Note that the application varies in intensity - the scales change from picture to picture.

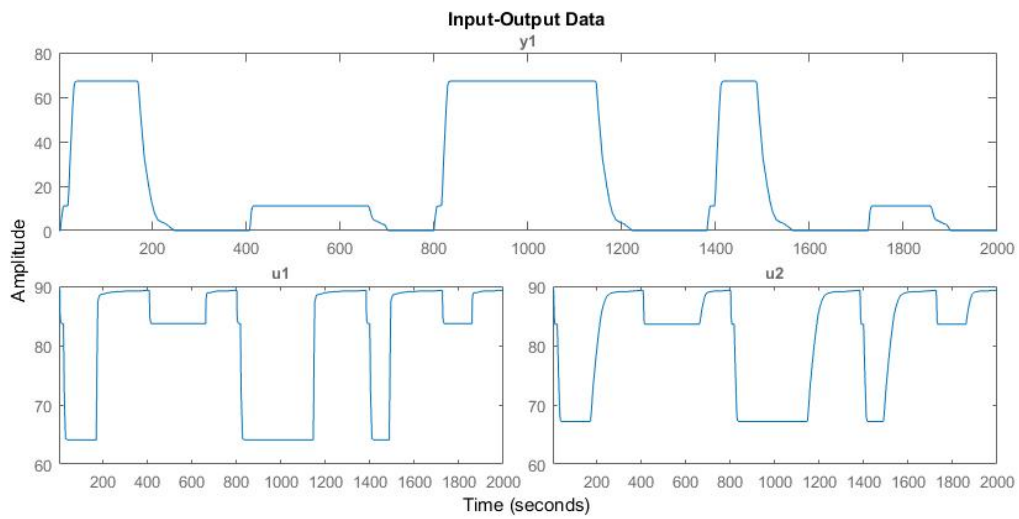


Figure 3.4 – Validation VAL01- Top, output: Brake Cylinder - bottom, Input: Brake Pipe & Auxiliary Reservoir. Horizontal values are time [ds] and vertical values are pressure [psi]

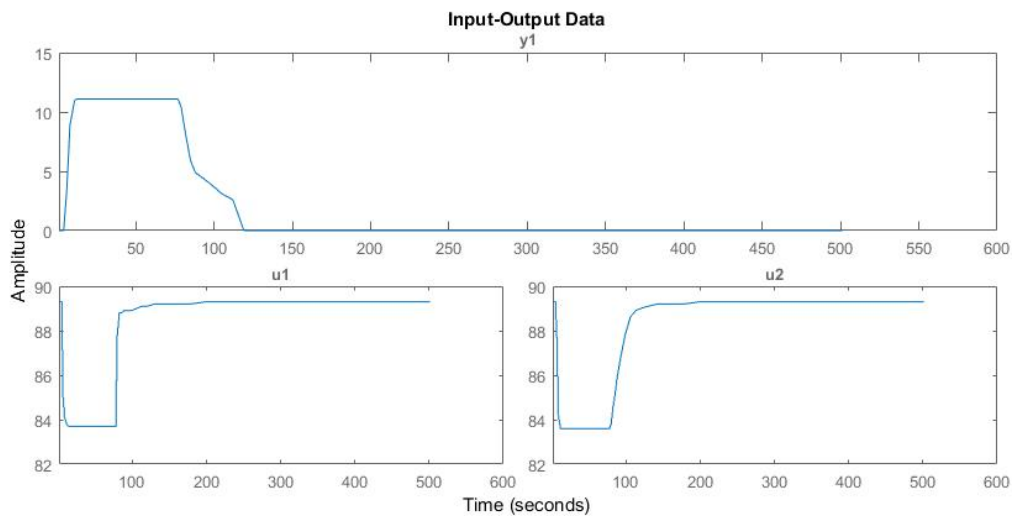


Figure 3.5 – Validation VAL02- Top, output: Brake Cylinder - bottom, Input: Brake Pipe & Auxiliary Reservoir. Horizontal values are time [ds] and vertical values are pressure [psi]

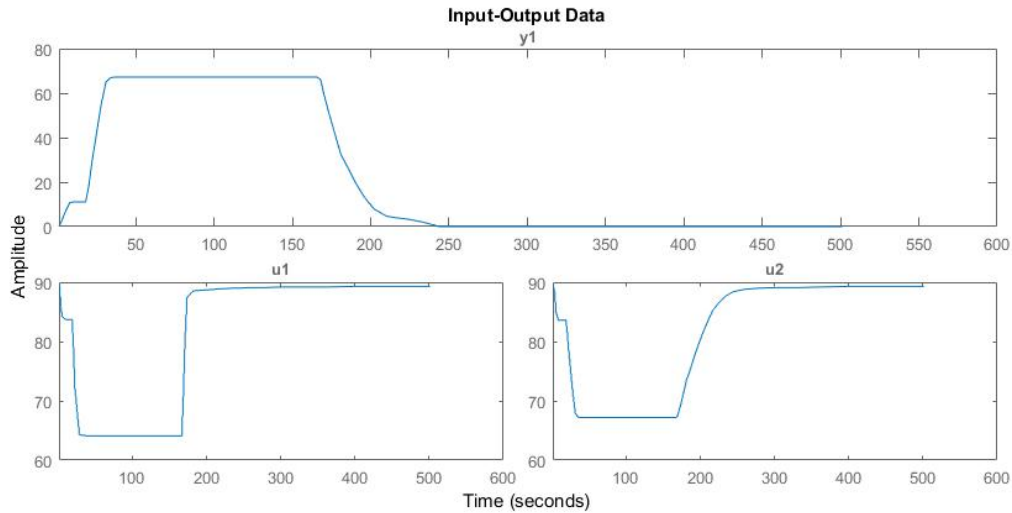


Figure 3.6 – Validation VAL03- Top, output: Brake Cylinder - bottom, Input: Brake Pipe & Auxiliary Reservoir. Horizontal values are time [ds] and vertical values are pressure [psi]

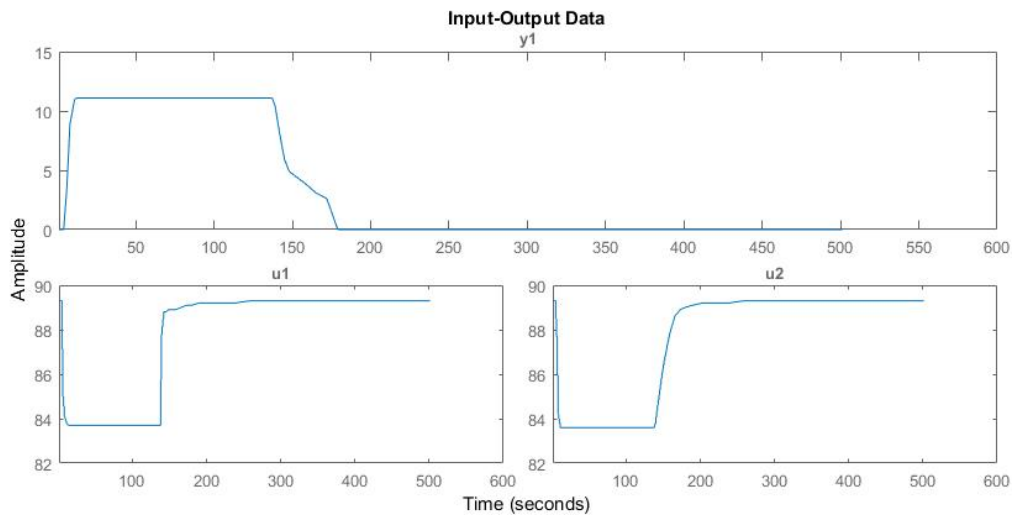


Figure 3.7 – Validation VAL04- Top, output: Brake Cylinder - bottom, Input: Brake Pipe & Auxiliary Reservoir. Horizontal values are time [ds] and vertical values are pressure [psi]

3.2 Methods Used

Since the aim is to build a mathematical model that represents the automatic brake valve and runs faster than the current physics based model, it is important to understand the nature of the problem at hand. So, an investigation on whether the model is linear or non linear is paramount.

The use of linear systems to represent non linear systems is commonplace in mechanical engineering. An example that comes to mind is the description of the movement of a

pendulum. For small angles, it is possible to replace the sine with the angle itself. However, for large displacements, the error between the angle and the sine of the angle increases and yield different results. The linear approach is not perfect and error free, nonetheless, the expectation is that the error are so minimum in terms of the outputs, that they are acceptable.

Initially, two methods were selected, polynomial regression with exogenous values (ARX) and a transfer function estimation model.

The ARX is one of the earlier and simpler models and is considered as a benchmark. The transfer function estimation is more complex and was chosen for there is a vast mathematical toolbox developed within control engineering that uses the transfer functions as a starting point. It is important to note, however, that the latter is frequency based. The NARX was chosen as a starting point on the neural network class of models. It is a well developed classical model that tend to be versatile and generally yields good results.

3.2.1 ARX (Auto Regressive eXogenous)

A straight forward way to find a relation between an input and output could be though modelling of the equation errors, on an equation similar to eq 3.1. In this equation one alters the variables a and b in order to minimize the error.

In this case, the AR is the Autoregressive part of the model or $A(q)y(t)$ and the eXogenous part is $B(q)u(t)$. The parameters for the ARX model may be estimates using the least squares linear regression.

Assuming the model,

$$y(t) + a_1y(t-1) + \dots + a_nay(t-na) = b_1u(t-nk) + \dots + b_nbu(t-nb-nk+1) + e(t), \quad (3.1)$$

where:

$y(t)$ - output at time t ;

n_a - number of poles;

n_b - number of zeros +1;

n_k - number samples before the outputs are affected by the input. Also known as dead time;

$y(t) \dots y(t - n_a)$ - outputs dependent of the current input;

$u(t - n_k) \dots y(t - n_k - n_{b+1})$ - delayed inputs that the current input depend upon;

$e(t)$ - Error ;

Or in compact form,

$$A(q)y(t) = B(q)u(t - n_k) + e(t), \quad (3.2)$$

where q is the delay operator.

Separating the equations,

$$A(q) = 1 + a_1q^{-1} + \dots + a_{na}q^{-n}, \quad (3.3)$$

$$B(q) = b_1 + b_2q^{-1} + \dots + b_{nb}q^{-n+1}, \quad (3.4)$$

In the case of a MISO (Multiple Inputs, Single Output) system, nk and nb are vectors. The Algorithm then solves a least square problem similar to,

$$(J^T J)\theta = J^T y, \quad (3.5)$$

In equation 3.5, θ is the parameter vector from the ARX method, J is the regressors matrix and y is the measured output,

$$\theta = (J^T J)^{-1} J^T y, \quad (3.6)$$

The ARX method was implemented using MATLAB native algorithm.

3.2.2 Transfer Function Estimates

Alternative to the ARX, the transfer function estimate method was also explored. The native MATLAB function "tfest" was used as standard implementation. The algorithm is as follows:

1. Perform Bilinear mapping to transform the domain into the transfer function domain;

2. Use the S-K iterations ([Drmac et al., 2015](#)) to solve the non linear least squares problem - assuming several inputs and a single output. The non linear least square loss function problem is as

$$\text{minimize}_{D_m N_i} \sum_{k=1}^{n_f} \left| W(\omega_k) \left(y(\omega_k) - \sum_{i=1}^{n_u} \frac{N_i(\omega_k) D(\omega_k)}{u} (\omega_k) \right) \right|^2, \quad (3.7)$$

where:

W = frequency independent weight;

D = Transfer Function denominator to be estimated;

N_i = i -th numerator referent to the i -th input;

y = system output;

u = system input;

n_f = number of frequencies;

n_u = number of inputs;

ω = frequency;

Making the S-K iterations, the algorithm minimizes using iterating methods,

$$\text{minimize}_{D_m N_{i,m}} \sum_{k=1}^{n_f} \left| \frac{W(\omega_k)}{D_{m-1}(\omega_k)} (D_m(\omega_k) y(\omega_k) - \sum_{i=1}^{n_u} N_i(\omega_k) u_i(\omega_k)) \right|^2, \quad (3.8)$$

where:

m is the current iteration;

$D_{m-1}(\omega)$ is the denominator identified on the previous iteration;

Each step is a linear problem of least squares. where the identified parameters are the solutions $D_m(\omega)$ and $N_{i,m}(\omega)$ for $i = 1, 2, \dots, n_u$.

The iteration is initialized with $D_0 = 1$. The actions of each iteration are described below.

- The first iteration identifies $D_1(\omega)$. $D_1(\omega)$ and $N_{i,m}(\omega)$ are polinomials expressed on the monomial base;
- The second and subsequent iterations express $D_m(\omega)$ and $N_{i,m}(\omega)$ in terms of rational orthogonal base functions on the unit circle and are of such form:

$$B_{j,m}(\omega) = \left(\frac{\sqrt{1 - |\lambda_{j,m-1}|^2}}{q - \lambda_{j,m-1}} \right) \prod_{r=0}^{j-1} \frac{1 - (\lambda_{j,m-1} * q(\omega))}{q(\omega) - \lambda_{r,m-1}}, \quad (3.9)$$

where:

$\lambda_{j,m-1}$ is the j -th pole identified at the previous step $m - 1$;

$\lambda_{j,m-1}^*$ is the complex conjugate of $\lambda_{j,m-1}$

q is the frequency domain variable on the unity disk;

- The algorithm run a maximum of 20 iterations that may be ended before that if there is no significant change on the loss function (>0.001 on the last 3 iterations);

3. Perform linear refinements - The S-K iterations, even when converge, not always translate on a local optimal solution. To find the critical point of the system, a second batch of iterations are performed. The critical points are solutions to the non linear equation groups. The algorithm searches for the critical point building linear approximations for the non-linear equations solving successively the resulting linear equations on the least squares sense. The equations are

- Equation for the j -th parameter of the denominator

$$0 = 2 \sum_{k=1}^{n_f} \text{Re} \left\{ \frac{|W(\omega_k)|^2 B_j^* \sum_{i=1}^{n_u} N_{i,m-1}^*(\omega_k) u_i^*(\omega_k)}{D_{m-1}^*(\omega_k) |D_{m-1}(\omega_k)|^2} \right. \\ \left. (D_m(\omega_k) y(\omega_k) - \sum_{i=1}^{n_u} N_{i,m}(\omega_k) u_i(\omega_k)) \right\}, \quad (3.10)$$

- Equation for the j -th parameter of the numerator

$$0 = -2 \sum_{k=1}^{n_f} \text{Re} \left\{ \frac{|W(\omega_k)|^2 B_j^* u_i^*(\omega_k)}{|D_{m-1}(\omega_k)|^2} (D_m(\omega_k) y(\omega_k) - \sum_{i=1}^{n_u} N_{i,m}(\omega_k) u_i(\omega_k)) \right\}, \quad (3.11)$$

The first iteration is initiated with the best solution found for the numerator N_i and denominator D_i , using the S-K iterations. Different from the S-K, the base function $B_j(\omega)$ do not change at each iteration and the functions that yielded the best results are used. As before, the algorithm runs for 20 iterations and the stop criteria is the same.

4. Return the transfer function parameters corresponding to the optimum solution within the S-K iterations and linear refinement;
5. Invert the bilinear mapping from step 1;

6. Perform an iterative refinement of step 5, in case it is not stable - we have chosen to enforce the model stability - using a non-linear least square search. In other words, with the flag activated, the algorithm will proceed with a search for a stable version of the system - if it is available;

This methodology was adapted from MATLAB help on TFEST function. (YOUNG; JAKEMAN, 1980) (Ljung, 2009)

3.2.3 Model search algorithm for TFEST and ARX

Given the nature of the problem, its complexity and process instability it has been chosen to perform a sweep on various pole and zero and delay values. The other option would be to run manually the algorithm for each step. The various identified parameters should provide a fair evaluation of the space we are working on and assess whether this is a suitable method or not for the problem at hand.

The program runs all cases according to figure 3.8, comparing the model with the estimation data and validation, from 1 to 4, averaging the adherences and calculating the standard deviation and record the 4 top models so far.

A total 30 poles and zeros is identified as well as delays from 0 to 4 seconds (the data is organised in discrete steps, 1 second apart) on each of the input variables. The results do not improve with the increased complexity and the computational times grow larger.

For the transfer function models, $np > nz$, i.e. the number of poles is larger than of zeros.

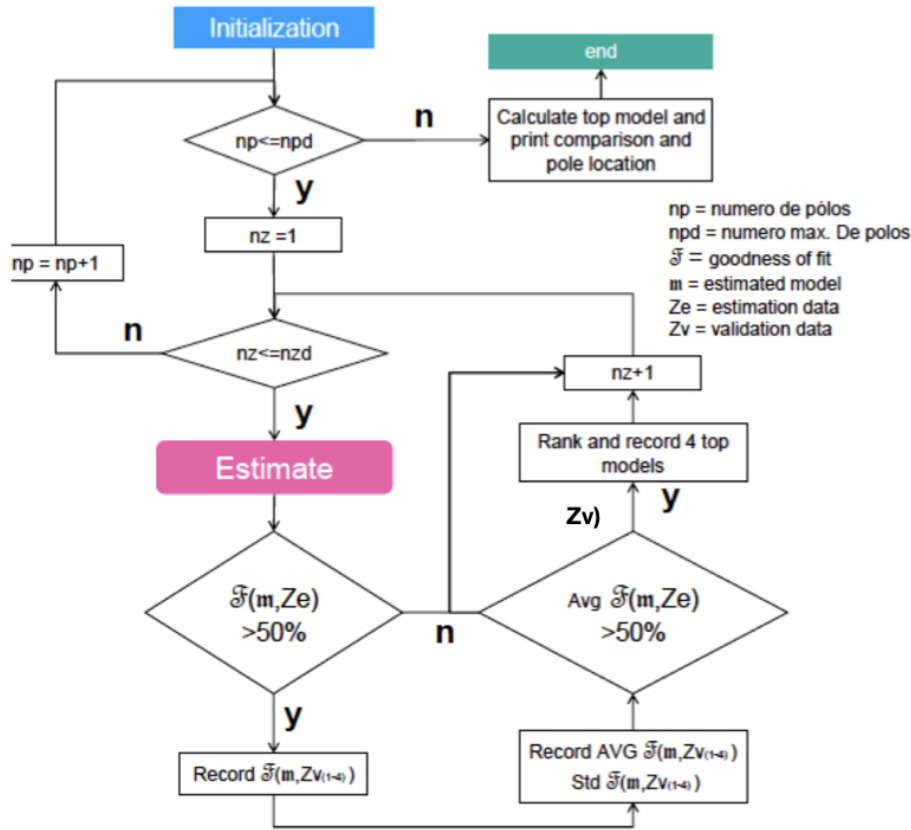


Figure 3.8 – Decision tree rationale for the search algorithm

3.2.4 NARX

The NARX models are a particular case of the NARMAX family. NARMAX stands for Nonlinear autoregressive with moving average and exogenous variables, whereas, NARX are Nonlinear autoregressive models with exogenous variables. Menezes (Jr.; Barreto, 2006) proposes an interesting implementation of the NARX based on artificial neural networks or ANN for short. This recurrent dynamic network has feedback connections on several of its neurons.

The defining equation is

$$y(n+1) = f[y(n), \dots, y(n-d_y); u(n), u(n-1), \dots, u(n-d_u)], \quad (3.12)$$

where:

$u(n)$ and $y(n)$ are real and denote the input and output respectively;

n is the time (in this case, discrete);

It can be written in the vector form as

$$y(n+1) = f[\mathbf{y}(n), \mathbf{u}(n)], \quad (3.13)$$

The function f is the one that needs to be approximated.

Figure 3.9 is a representation of the topology of the artificial neural network with the input layer with its delays, the middle layer and the output layer. Once set, the ANN have to be

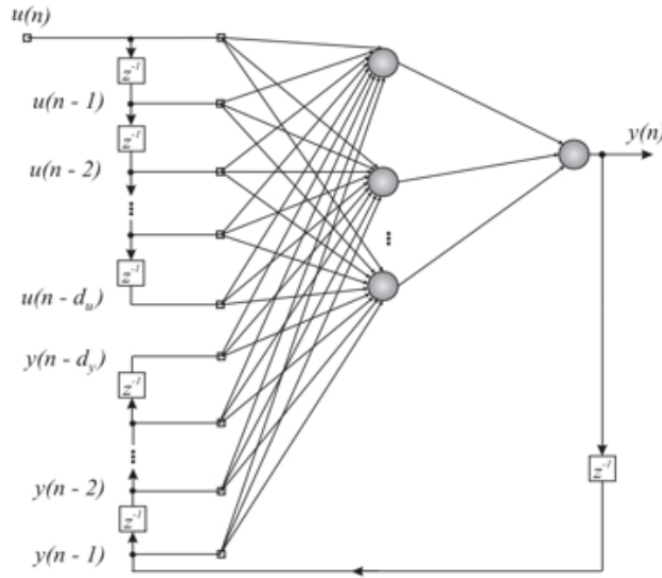


Figure 3.9 – A NARX network (Jr.; Barreto, 2006)

trained. The standard iterative back propagation algorithm can be used. There are, however, two options for this training, parallel mode and Series-Parallel (SP) mode. The first uses estimated regressors to train the output, making it less accurate. In the case at hand, we will feed the actual data to the training of the feedforward network. The equation for the SP training mode is as follows

$$\begin{aligned} \hat{y}(t) &= \hat{f}[y_{sp}(t-1), \dots, y(t-n_y); u(t), u(t-1), \dots, u(t-d_u)] \\ &= \hat{f}[\mathbf{y}_{sp}(\mathbf{t}-1); \mathbf{u}(\mathbf{t}-1)], \end{aligned} \quad (3.14)$$

For the training, initially, the Levenberg-Marquardt algorithm was chosen. Though it requires more memory compared to the methods currently available on MATLAB artificial neural network toolbox for time series, it is faster compared to the alternative methods. Alternatively, Bayesian verification will also be used to tackle more complex cases, such as the closed loop trainings later in this work. This technique uses the code implemented on MATLAB (Foresee; Hagan, 1997) (MacKay, 1992)

No result normalization was done for the ANN other than those on the already implemented algorithms.

3.3 Method Selection

The algorithms performed accordingly and yielded results that are presented below have been selected among the several created during the investigations. They present the best achievable adherence with the given methods and implementations.

There is indication that the problem is not linear and may not be represented by a linear model with reasonable complexity. Therefore, the non linear methods may be more adequate to describe the problem at hand.

3.3.1 ARX

More than 252301 iterations were searched for the best ARX based model. The best candidate has $na = 1$, $nb = \begin{bmatrix} 30 & 30 \end{bmatrix}$ and $nk = \begin{bmatrix} 1 & 1 \end{bmatrix}$, where na is the order of the output polynomial nb the input (Brake Pipe and Aux. Reservoir respectively) and nk , the delay. A discrete time system was used with z as operator - in lieu of the continuous time s variable - in other words, the study was done on using the discrete Z operator instead of the continuous Laplace operator. The figures show the adherence of the estimated model in comparison to the identification data. At first glance, one might consider the fit of 89.72% as adequate notwithstanding, at a more careful inspection, it is possible to see on the figure 3.10 that the results are not as expected, specially at the lower pressure levels of the minimum brake application.

The same undesirable effect can be seen on the validation 1 data (VAL01) - figure 3.11, where a good fit is found for the maximum applications - high pressures - and an unacceptable fit on the lower pressures. It is clear from the results that the ARX with the selected parameter set has not been able to properly represent the physical model or the legacy data behaviour accordingly.

The same effect is seen on the individual applications. Figures 3.11, and 3.12 show that the model can properly represent the maximum service pressure and fails at the minimum service applications, at times provoking an instability that render the model useless for its intended use.

For the ARX, a maximum order of 30 has used. It is assumed that above this, the complexity is high and there would be little benefit in using the proposed model rather than the

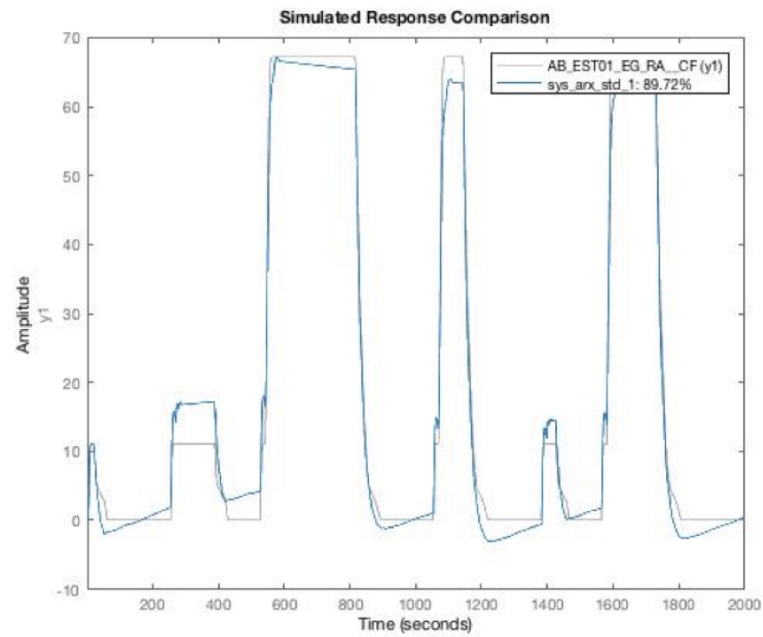


Figure 3.10 – Comparison between output of the model and legacy output.

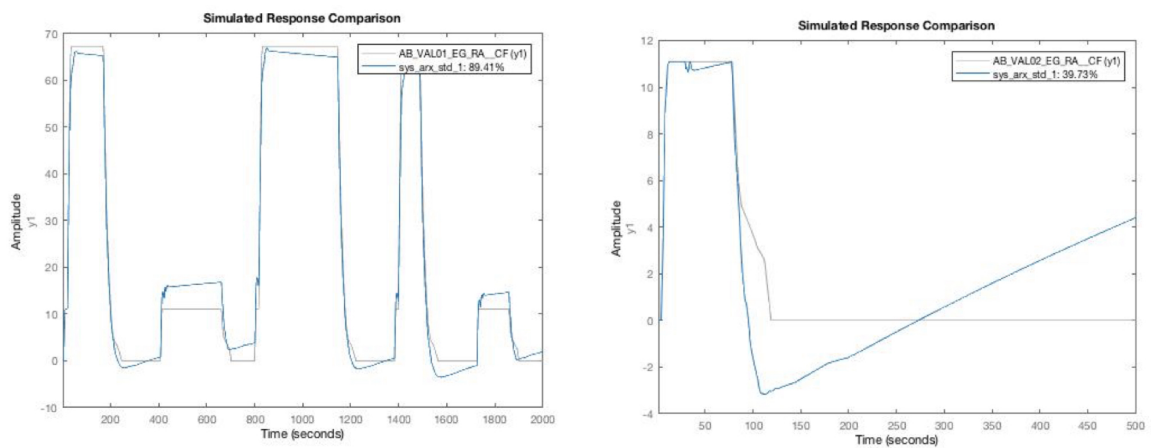


Figure 3.11 – Comparison between output of the model and legacy output - Validation data 1 & 2.

legacy approach.

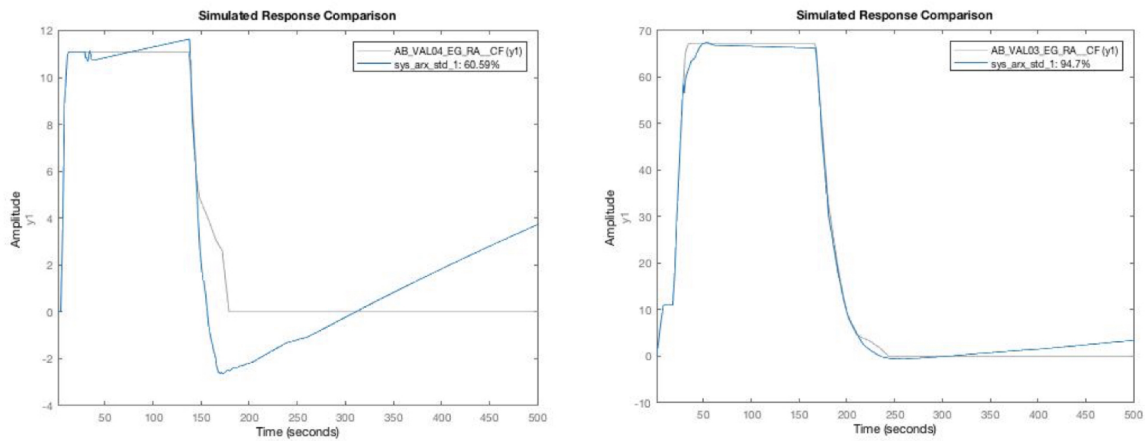


Figure 3.12 – Comparison between output of the model and legacy output - Validation data 4 & 3.

3.3.2 Transfer Function

For this method, the maximum number of poles was 20. No investigation of delays was done for this method - it is more complex and takes more time to calculate, what goes against the purpose of reducing the take taken to obtain the results. Same as previously, as the data is organised in discrete steps, it was chosen to consider the time as discrete and the z operator, rather than s was used (for the data is discrete and the computer, digital).

The best fit model has $\begin{bmatrix} 19 & 16 \end{bmatrix}$ and $\begin{bmatrix} 14 & 11 \end{bmatrix}$ poles for the brake pipe (BP) and the aux reservoir (AR) inputs respectively. Although up to 20 poles were allowed, the more complex model was not the one with better adherence. This seems to indicate that the physical model might be within the designated number of poles. Figure 3.13 shows the model response to the identification data and to validation data 1, that is purposely similar. The fit is 94.% for the estimation data and 93% for the validation. The more complex method is more accurate than the ARX. Nevertheless, there are still issues on the lower pressures or minimum brake applications, what might indicate that the system is indeed non-linear.

Again, the model has a good representation of the problem, but accuracy lacks for real use. On the validation data 02 to 04, figures 3.14 and 3.15 show the difficulty of the model in representing the minimum brake applications. The figures show the overshoots and swings the model goes through to represent the behaviour

For the maximum application, figure 3.15, the model does an excellent job with minimum oscillations and a fit of 98% whereas on figure 3.14, though the fit is very good at more than 80%, there are oscillations and the graph shows that there is not a good agreement

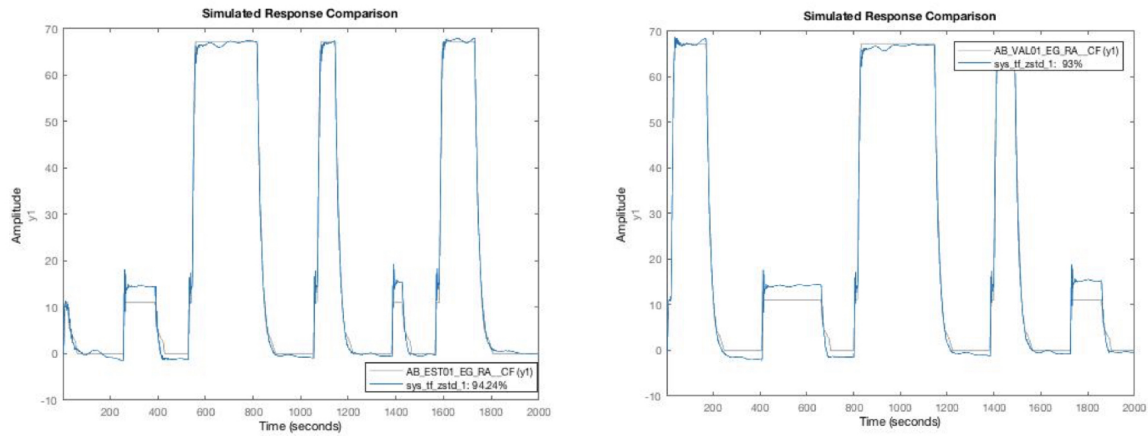


Figure 3.13 – comparison between output of the model and legacy data- Estimation data & validation data 1

between the model and the target.

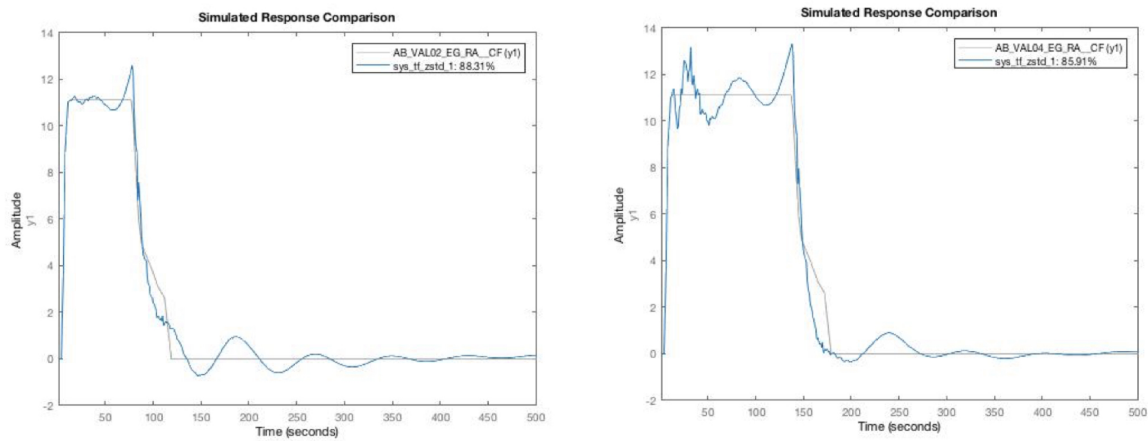


Figure 3.14 – Comparison between output of the model and legacy data - Validation data 2 and 4

3.3.3 Discussion

As it was observed on the results for the two first proposed methods of system identification, they were not adequate. The models were not able to correctly represent and simulate both the maximum application and minimum application on the automatic brake valve.

Therefore, from the obtained results, it is possible to infer that linear models may not be adequate for the problem at hand, with the proposed complexity, so the need to evolve to nonlinear models.

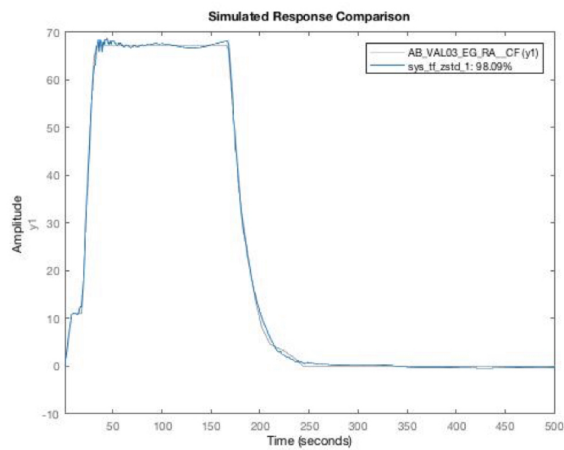


Figure 3.15 – Comparison between output of the model and legacy data - Validation data 3

3.3.4 NARX

Using some already implemented toolboxes on MATLAB, it was possible to assess the potential of using artificial neural networks

Figure 3.16 shows the overall interface to the neural network toolbox on MATLAB. Figure 3.17 shows the adherence of the estimation by a 10 neurons open network of the given data with 2 delays on the input and output

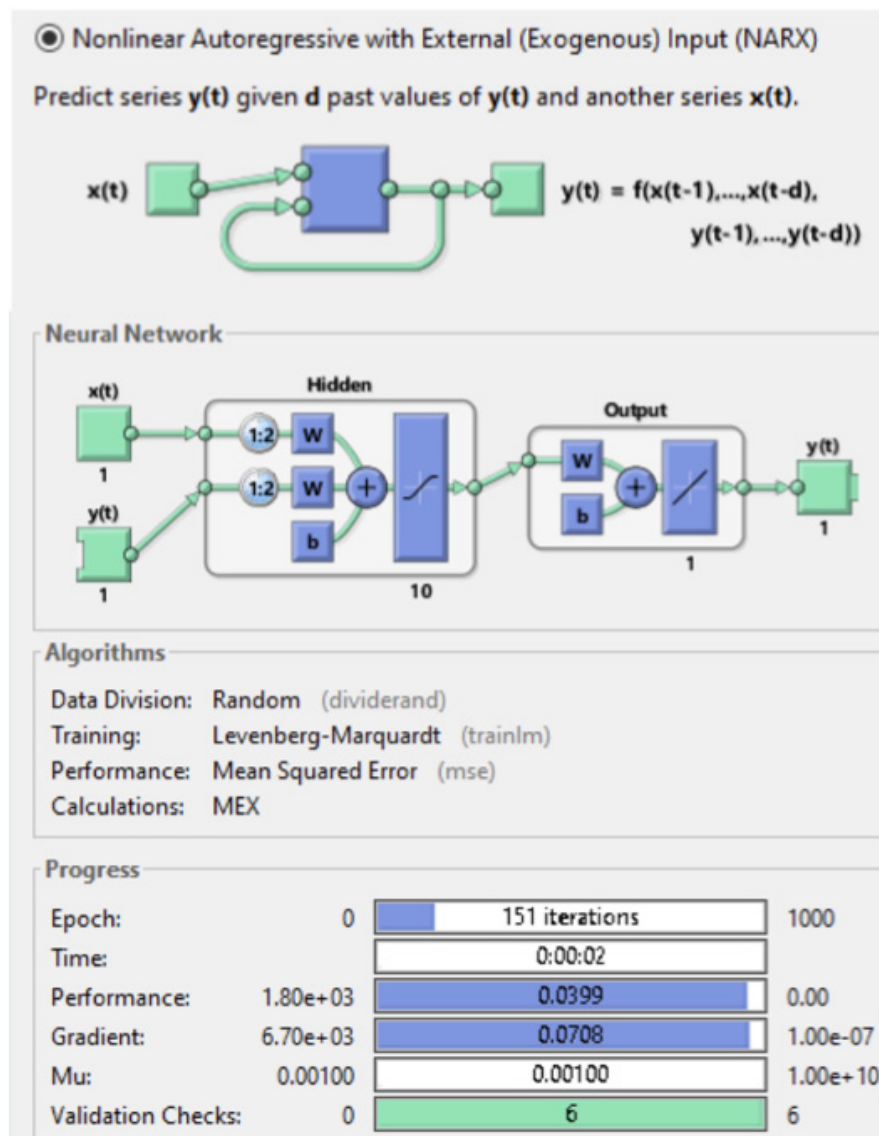


Figure 3.16 – NARX implementation and training on MATLAB

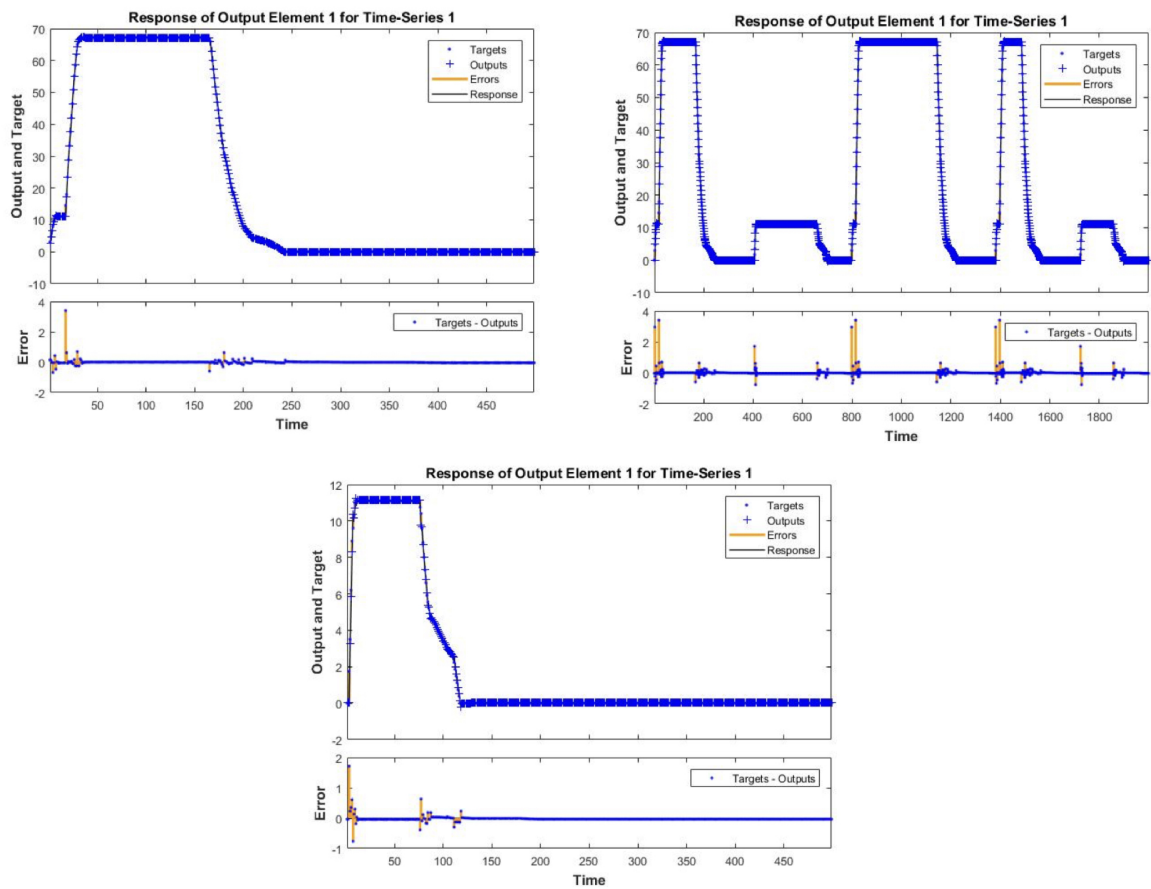


Figure 3.17 – NARX results for the validation data (pressure [psi] x time [ds])

4 OPEN LOOP MODEL EXPLORATION

On the previous chapter, NARX artificial neural network appears to be the most adequate method to identify the the problem at hand, among those explored. In this chapter we will investigate the method further, create the model and train it as well as exploring different parameters to achieve better results.

Traditionally, the neural network is trained in open loop - i.e. the input $y(n)$ is taken from the target rather than the calculated data. This is a fast and efficient way to train the ANN. Once the model has been trained, the loop is then closed and the $y(n)$ is calculated from $y(n - 1)$. Open loop ANN cannot be deployed for they require target information that is not present, closed loop ANN can.

4.1 Redefining the Inputs

Having chosen the NARX and artificial neural network approach, the work proceeds to the exploration of this strategy. At first, the perceived more complex model will be addressed, the brake cylinder pressure output from the brake pipe and auxiliary reservoir input. A new set of inputs was created based on the previous work by Teodoro ([Teodoro, 2016](#)). In a similar manner, five datasets were created for the first freight car in the consist and instead of using the AB valve, the simulations were done using the contemporary ABDX. The data for the Brake Pipe and the Auxiliar Reservoir were used as inputs and the pressure at the brake cylinder was the output.

The data represents the second vehicle (first wagon after the locomotive) on a 101 vehicle train comprised of 50 wagon pairs and 1 locomotive.

The data was constructed using the numerical solution of the Navier-Stokes equations and the time interval was 0.001s. There are numerical variations on the signal that are not physically real, i.e the hight frequency sharp edged fluctuations in the results. The fluid inertia would not allow for this fluctuations to persist and would promote the decay and smoothing of the curve quite fast. Figure 4.1 shows the general form of the brake pipe and the aux reservoir pressures. It also shows a zoom of the aforementioned perturbations

Two additional modifications are necessary to the input before proceeding, the conversion from Pascal to pounds per square inch (psi) and the reduction of the number of time

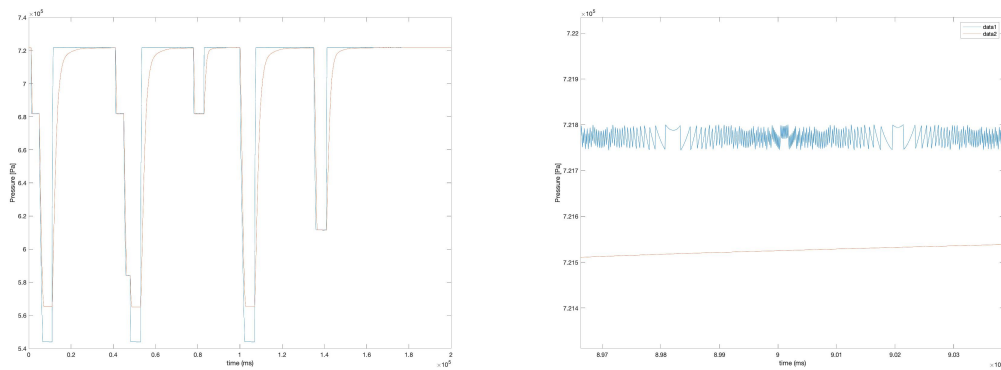


Figure 4.1 – New Brake Pipe and Aux Reservoir inputs for EST01 - detail of fluctuations.

steps. A simple algorithm was wrote to perform this changes. For the decimation strategy, it was chosen to average the ten data points into a single value and write on the new vector. However, the averaging did not stabilise the signal sufficiently, so the high frequency was dealt by rounding off the signal - what was only done on the brake pipe and brake cylinder. The aux reservoir did not require the rounding. Attempts were made to use filters however, they did not preserve the signal adequately, creating several artifacts, mainly due to the abrupt changes the signal undergoes at some regions.

In the same manner as previously, there is an input for model estimation - EST01, that consists of several full, minimum and intermediary service applications and four validation datasets - VAL01, that mimics the same as EST01 with sequential applications and releases, VAL02 that has a full service application, VAL03 has an intermediate service application and VAL04 has a minimum service application. The plots for these are on figure 4.2.

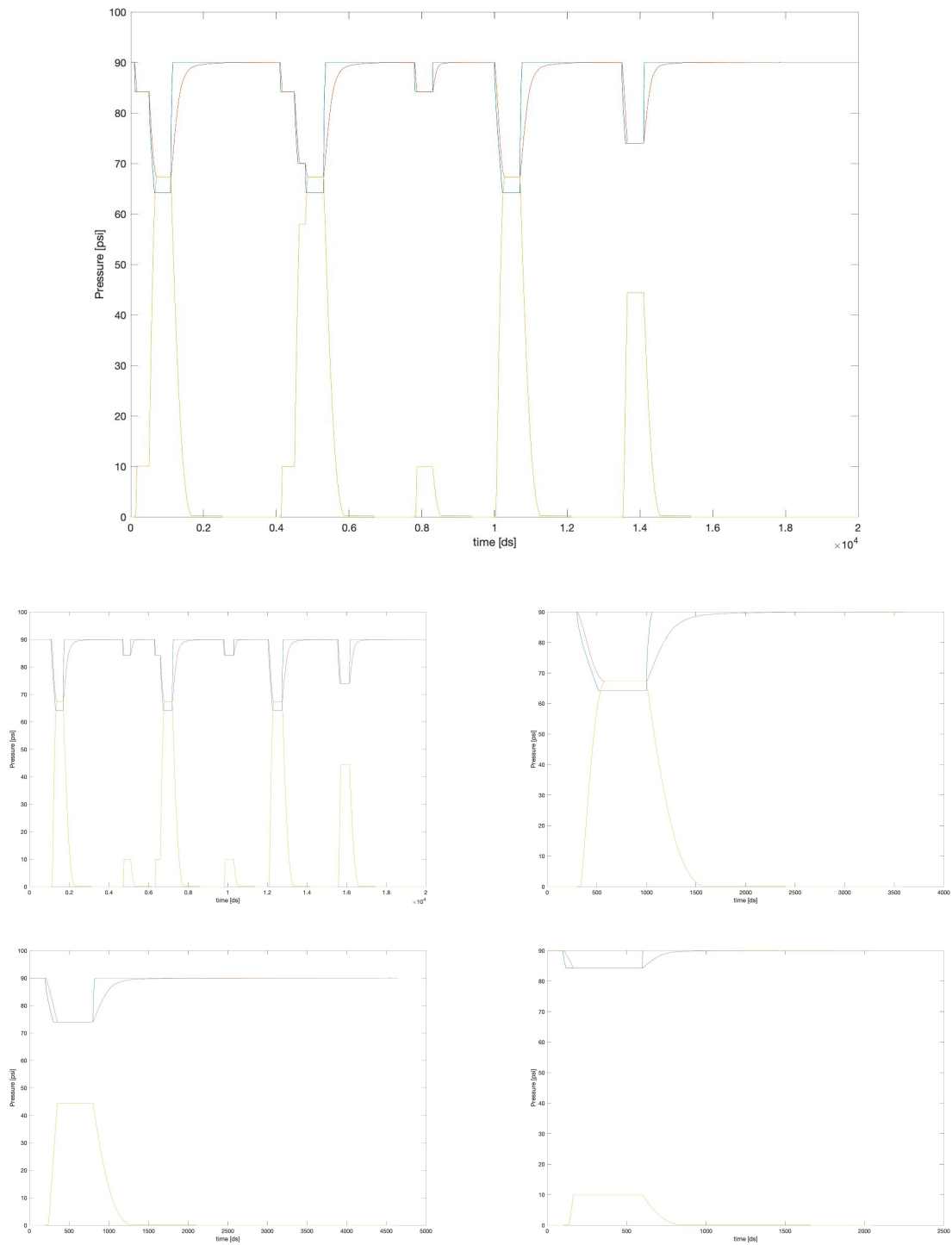


Figure 4.2 – Training and validation data.

4.2 Changing the Parameters

Having defined that the non linear ARX neural network is the most adequate strategy within those presented, an exploration shall be done on how the neural network (NN) be-

haves when the parameters are altered for this specific problem

To achieve this, a variable sweep was created. The parameters to be altered are:

- *id*: input delay - or the number of steps previous to current are used on the estimation. The explored range varies from 1 to 5, i.e. the estimation uses from $x(n)$ to $x(n - 5)$;
- *od*: output delay - same as previous, instead of considering different input steps, the network will consider different output steps and this shall vary from 1 to 5, i.e. from $y(n)$ to $y(n - 5)$ considering the estimation output as $y(n + 1)$;
- *nl*: number of neurons - this determines the number of hidden neurons on the neural network. The explored values range from 1 to 20.

In total there will be 500 different possibilities explored. The parameters recorded for the evaluation will be

- *mperf*: the average accuracy - measured as the mean square error (mse) for each of the validation cases and the estimation case. The smaller the better;
- *mtempo*: the average time for the execution of the estimation for each of the validation cases and the estimation case;
- *perf*: the accuracy as mse (mean square error) for each of the individual validation cases and the estimation case. This is a 1x5 vector that when averaged yields *mperf*;
- *tempo*: is the time required to run the estimation of each validation case and the estimation case. This is also a 1x5 vector. Times shall not be considered absolute and shall be treated as a comparison. The computer load was kept constant throughout the execution;
- *tTrain*: this is the time required for the training process to occur. This metric will be used to understand the training process only and shall not bear indication on the decision.

4.3 Training Function

Based on MATLAB functions an algorithm was devised to perform the set up and training of the NARX network as follows. This function has as inputs the delay on the input data, the output data delay, the number of hidden neurons, the training function - for this case, all iterations use the Levenberg-Marquardt function for faster results, the traindata - a vector that

contains the training, validation and testing ratios, the input vector- in this case a $n \times 2$ matrix indicating the two inputs - brake pipe and auxiliar reservoir, the target output - an $n \times 1$ vector with the brake cylinder pressures and a counter for the figure generation

The function output is the trained neural network, the counter for figure generation and the training time.

Inside the function, it unpacks the information and translate the input variable to the internal variables. This is not required however, it was done so for clarity (lines 1..10).

The function then creates the artificial neural network (ANN) structure according to the inputs. Note that it was chosen to use an open network, i.e. it used the $y(n), y(n - 1) \dots$ from the targets and the final calculation, rather than the internal values on the last layer, before the final neurons (line 11).

The inputs are then prepared and organised in the way MATLAB expects (line 12). The train, validation and test dataset is separated in a random manner with the prescribed parameters. We shall use 70% of the data for training, 15% of the data for validation and 15% of the data for testing (lines 13..17). These seem to be common practice. As the network training process occurred as planned, no different ratios were tested.

The function then instructs that the accuracy metric is the mean square error (mse) (line 18). Some function flags are activated to allow graphics (line 19) and finally the training happens (line 21).

```
function [out1,fout,tTrain]=
=funNet(inDly,fdbkDly,lSize,trainFunc,traindata,Xin,Tin,fin)

1   trainFcn=trainFunc;
2   inputDelays=inDly;
3   feedbackDelays=fdbkDly;
4   hiddenLayerSize=lSize;
5   trainratio=traindata(1);
6   valratio=traindata(2);
7   testratio=traindata(3);

8   X=Xin;
9   T=Tin;
```

```

10     f=fin;

% creates the neural network structure according to parameters -
% has not trained it just yet
11     net=narxnet(inputDelays,feedbackDelays, ...
...,hiddenLayerSize,'open',trainFcn);

%preparing the inputs
12     [x,xi,ai,t]=preparets(net,X,{},T);

%dividing the data for training, validation and testing
13     net.divideFcn='dividerand'; %random divide the data
14     net.divideMode='time';
15     net.divideParam.trainRatio=trainratio;
16     net.divideParam.valRatio=valratio;
17     net.divideParam.testRatio=testratio;

18     net.performFcn='mse'; %accuracy metric- mean square error

19     net.plotFcns = {'plotperform','plottrainstate',...
'ploterrhist', 'plotregression', 'plotresponse', ...
'ploterrcorr', 'plotinerrcorr','plotwb'};

%train
20     tic;
21     [net,tr]=train(net,x,t,xi,ai);
22     tTrain=toc;

23     out1=net;
24     fout=f;

end

```

Note that the *tic* and *toc* commands at the train command line give the training time.

4.4 Assessing Accuracy

Once the artificial neural network (ANN) has been trained, it will be run for the estimation data and validation data. This validation shall not be confused with the internal validation used to train the ANN. Assessing the model with a different data set helps to spot overtraining

The inputs are the neural network obtained on the previous function (*in0*) the input data (*in1*), the target data (*in2*) and the figure counter (*in3*). The outputs are the accuracy, the model estimated output for the input data (*yout*) and the figure counter (*fout*).

Similarly to the previous, the first part is the transposing of the inputs (lines 1..4). Then the data is conditioned in the way MATLAB expects to receive it (line 5). The model run command is quite simple and is on line 7. Note that there are no targets as inputs. Note that the command uses as input the simulated output (*y*) and the target data (*t*) as well as the NN.

The simple error is computed by subtracting the simulated output *y* with the target output *t* (line 9). The accuracy is calculated based on the accuracy metric that was used previously, the mean square error (line 10).

```
function [performance1,yout,tn,fout]=func01(in0,in1,in2,in3)
```

```
1  f=in3;
2  net=in0;

3  X=tonndata(in1,false,false);
4  T=tonndata(in2,false,false);

5  [x,xi,ai,t]=preparets(net,X,{},T);

6  tic;
7  y = net(x,xi,ai);
8  tn=toc;

9  e = gsubtract(t,y);
```

```

10 performance = perform(net,t,y);
11 performancel=performance;
12 yout=y;

13 fout=f;
end

```

4.5 Parameter Sweep

Using the previous functions, a series of iterations was constructed to try and assess several neural networks. The program is organised in sections. First basic clean up and data conditioning (lines 1..19). This section loads the input data described at the beginning of this section. It then initiate the variables that will be used on the program. Then it defines the validation/train/test ratios and conditions the input data for the estimation. The sweep iterations are from line 20 onwards. The training function is run on line 27. A new network is trained for every combination of parameters. Lines 28..31 indicate a general accuracy assessment. The accuracy and time to run are organised into vectors. This step is repeated for all the input cases - estimation and validation - until line 47. The vector with the execution time and the accuracy are then averaged over the estimation and validation cases and the results structure is constructed. This structure contains the input and output delays and the number of neurons used to define the neural network as well as the number of the run (c), the average accuracy, average running time, the accuracy and time vectors and the time it took to train the NN. The hardware used is on appendix 2 After the result structure is filled, the program searches for the best accuracy, fastest execution and best combination of both (BB). Lines 53 to 96 outline this search. When a better network than the currently in store is found, a vector is constructed with its information and the trained NN is saved. Finally, line 97 saves the results for later assessment.

```

01 clear all
02 close all
03 clc
04 load ('input data')

```



```

%% Data conditioning

%Estimation data
05  f=1;
06  n=1;
07  bestTempo=1;
08  bestPerf=1;
09  bestTP=1;
10  c=1;

11  trainratio=70/100;
12  valratio=15/100;
13  testratio=15/100;

14  traindata(1)=trainratio;
15  traindata(2)=valratio;
16  traindata(3)=testratio;

17  X=tonndata(ABDX_EST01_EG_RA_1,false,false);
18  T=tonndata(ABDX_EST01_CF_1,false,false);
19  trainFunc='trainlm';
%using the Levenberg-Marquardt training,
% others are 'trainbr' and 'trainscg'

%begin the sweep:

20  for id=1:5
21      for od=1:5
22          for nl=1:20
23              inDly=1:id;
24              fdbkDly=1:od;
25              lSize=nl;

```

```

26             n=1;

27             [net,f,tTrain]=funNet (inDly,fdbkDly,lSize,trainFunc,...
...traindata,X,T,f);

28             [perfEST01,yEST01,tnEST01,f]=func01 (net,...
...ABDX_EST01_EG_RA_1,ABDX_EST01_CF_1,f);
29             perf (n)=perfEST01;
30             tempo (n)=tnEST01;
31             n=n+1;

32             [perfVal01,yVal01,tnVal01,f]=func01 (net,...
...ABDX_VAL01_EG_RA_1,ABDX_VAL01_CF_1,f);
33             perf (n)=perfVal01;
34             tempo (n)=tnVal01;
35             n=n+1;

36             [perfVal02,yVal02,tnVal02,f]=func01 (net,...
...ABDX_VAL02_EG_RA_1,ABDX_VAL02_CF_1,f);
37             perf (n)=perfVal02;
38             tempo (n)=tnVal02;
39             n=n+1;

40             [perfVal03,yVal03,tnVal03,f]=func01 (net,...
...ABDX_VAL03_EG_RA_1,ABDX_VAL03_CF_1,f);
41             perf (n)=perfVal03;
42             tempo (n)=tnVal03;
43             n=n+1;

44             [perfVal04,yVal04,tnVal04,f]=func01 (net,...
...ABDX_VAL04_EG_RA_1,ABDX_VAL04_CF_1,f);
45             perf (n)=perfVal04;

```

```

46         tempo(n)=tnVal04;
47         n=n+1;

48         mtempo=mean(tempo);
49         mperf=mean(perf);

50         Resul(c,:)={id,od,nl,c,mperf,mtempo,...
...perf,tempo,tTrain};

51         Resul(c,1:6)

52         c=c+1

53         if bestPerf>mperf
54             best(1)=id;
55             best(2)=od;
56             best(3)=nl;
57             best(4)=c;
58             best(5)=mperf;
60             best(6)=mtempo;
61             best(7)=tTrain;
62             bestPerf=mperf;
63             best
64             netBest=net;
65             save netBest;
66         end
67         if bestTempo>mtempo
68             fast(1)=id;
69             fast(2)=od;
70             fast(3)=nl;
71             fast(4)=c;
72             fast(5)=mperf;

```

```

73             fast(6)=mtempo;
74             fast(7)=tTrain
75             bestTempo=mtempo;
76             fast
77             netFast=net;
78             save netFast;
79         end
80         bestTP1=mtempo*mperf;
81         if bestTP>bestTP1
82             BB(1)=id;
83             BB(2)=od;
84             BB(3)=nl;
85             BB(4)=c;
86             BB(5)=mperf;
87             BB(6)=mtempo;
88             BB(7)=tTrain
89             bestTP=bestTP1;
90             BB
91             netBB=net;
92             save netBB;
93         end
94     end
95 end
96 end
97 save _Results_

```

4.6 Parameter Sweep Results

Here we present the results of the parameter sweep. They are divided into 3 sections, accuracy, execution time and the best combination of both.

4.6.1 Best Accuracy

As outlined on the program presented on the previous section, the sweep has recorded the best performance network on a vector. For the problem and the run at hand the most accurate arrangement is 4 inputs deep, i.e. the model considers $x(x), x(n-1), \dots, x(n-3)$ as well as 4 output delays or $y(n), y(n-1), \dots, y(n-3)$ to calculate $y(n+1)$ on a 20 layer neural network. This arrangement took 40s to train and an average of 0.3351s over all validation and estimation cases. Run times should not be taken in absolute terms for they depend on the hardware the program is run. The average mse (mean square error) between the output and the target is 2.517E-5. The values of accuracy and time for each of the cases are outlined on 4.1

	Error (mse) 10E-4	run time
Est01	0.2239	0.6859
Val01	0.2729	0.6501
Val02	0.0974	0.1195
Val03	0.2410	0.1511
Val04	0.4235	0.069

Table 4.1 – Accuracy (error) and time results for the lowest error neural network

4.6.1.1 EST01

The figure (4.3) shows the processed output of the model, note that the response match the target with very good accuracy and as mentioned earlier, the main errors are located at the transitions. However, the error is not larger than 0.2% at any condition. It can be noted that though the model is able to estimate the target accurately, there are small perturbations when the target has abrupt changes - specially at the plateaus. Also, it can be noted some minor difficulties following some of the target data. The correlation plot shows the estimation at one axis and the model at the other axis. As expected, it is a diagonal line showing the correlation mentioned earlier the regression is 1 (or higher than 0.999, MATLAB's precision for this parameter).

Since this is the best model and the overall mse (mean square error) is quite low, all these results are expected, in spite of the minor bumps and inaccuracies.

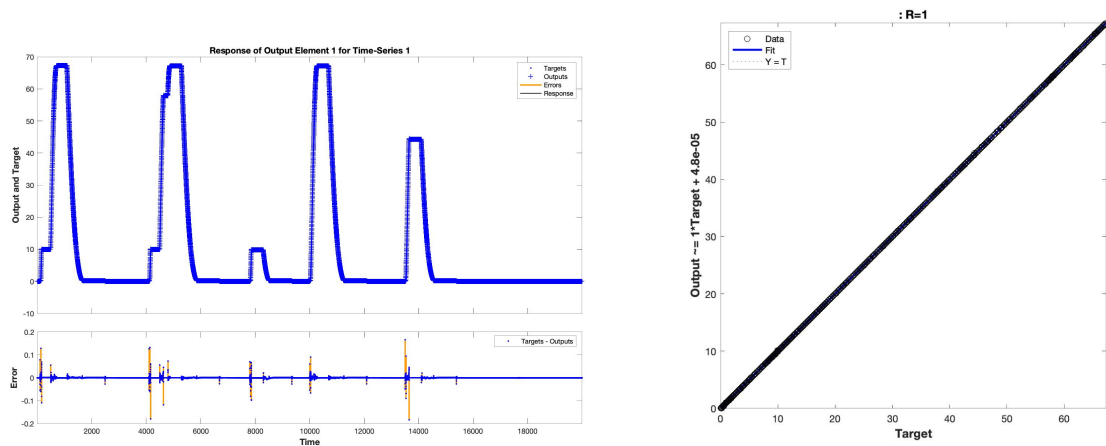


Figure 4.3 – Targets, outputs, error and response for estimation data & correlation (EST01).

4.6.1.2 Validation data

The same plots are presented for the estimation data are presented for the validation data. Validation 1 (VAL01) is similar to the estimation data and yield equally similar results. Was the model overfitted, it would not have been able to simulate VAL01 properly. The results are presented in the same order as previously, a comparison between the target (dotted line) and the model (solid line) and the simple error (magenta). On figure 4.4, the overall results shows a good agreement from the model to the target data. The correlation shows the model and target plotted on each axis, forming the desired diagonal line. The R for the model is higher than 0.999.

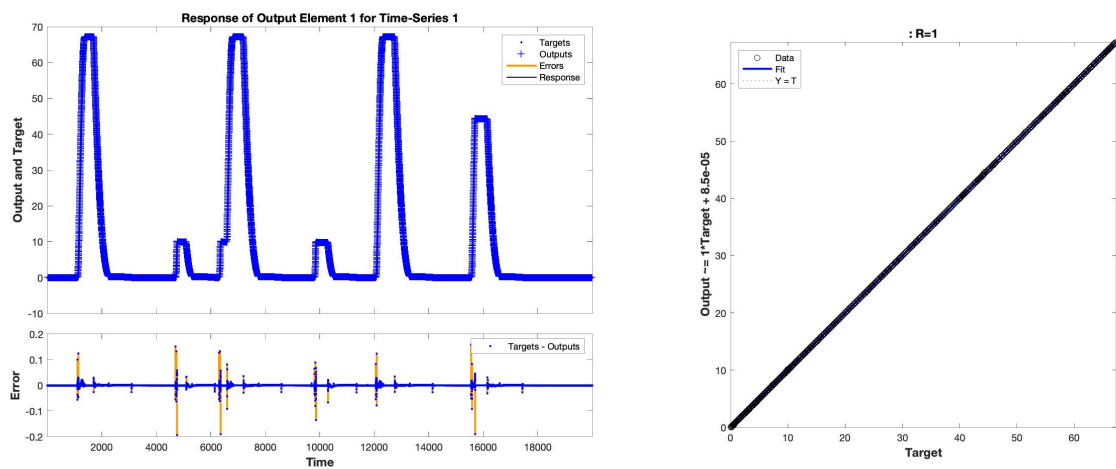


Figure 4.4 – Targets, outputs, error and response & correlation for validation data (VAL01).

Validation 2, 3 and 4 (VAL02, VAL03 and VAL04) are individual brake applica-

tions and show some interesting data. Similar to what happened to the other models on the second chapter of this work, the higher pressures yielded better results than the lower cylinder pressures. The results again will be presented in the same order as previously.

On the results for validation 2 model (VAL02) versus target, one can see the excellent agreement between the model and the target outputs. This good agreement between target and model explains the very low mean square error of $9.74\text{E-}6$, the smallest among the validation examples. Figure 4.5 shows the output response. Note that the error is no larger than 0.7%. The correlation shows the model vs target plot with very good agreement and the $R=1$.

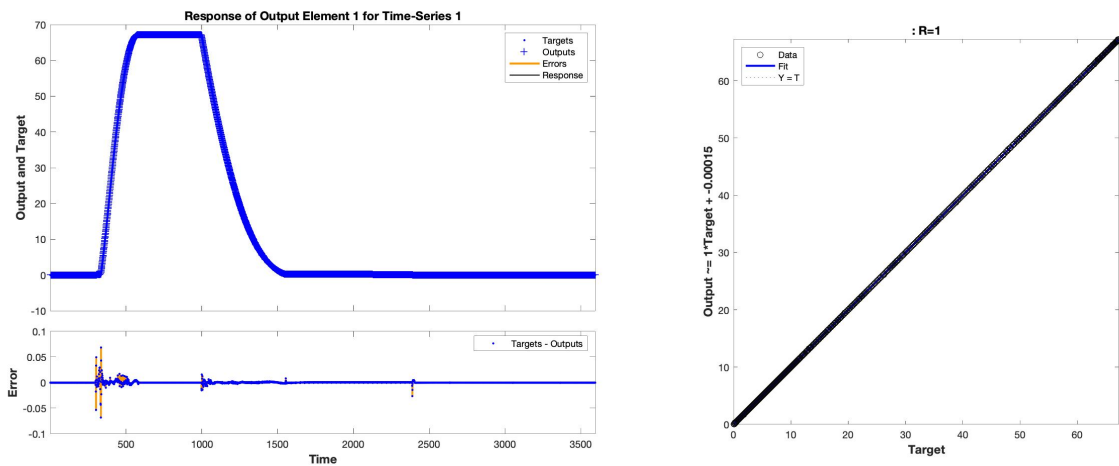


Figure 4.5 – Targets, outputs, error and response & correlation for validation data (VAL02).

Validation 3 is the intermediary application, as such, and as expected, it shows result performance in between validation 2 and 4. The same presentation scheme is used. The transition to the top cylinder pressure is not as smooth as previously, what creates an overshoot of 0.3%. The beginning also have higher instability compared the VAL02 (figure 4.6). The model x target plot show the expected diagonal behaviour.

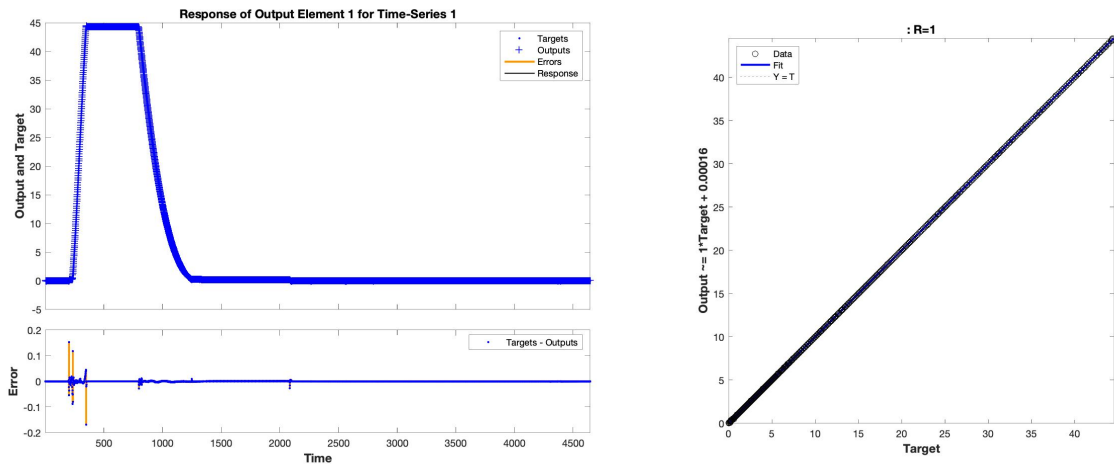


Figure 4.6 – Targets, outputs, error and response for validation data (VAL03).

Validation 4 has yielded the worst mean square error of all datasets: $4.235E-5$. Possibly, this result is related to the higher weight of the higher pressures at the estimation data. Also, the model seems to tend toward the higher pressures at the brake cylinder (VAL02 vs VAL04), what seems to also have been a trend on the linear models presented on chapter 2. Figure 4.7 shows the model response. The variations in the direction of the target data, i.e. derivative changes that are abrupt cause any continuous model would have difficulties adhering to the targets. This model does a good job in reducing the error at the expense of having some high frequency components, what is indicated by the fluctuations at the beginning of the brake cylinder pressure ascent. The correlation between target and model is, as expected, a diagonal line.

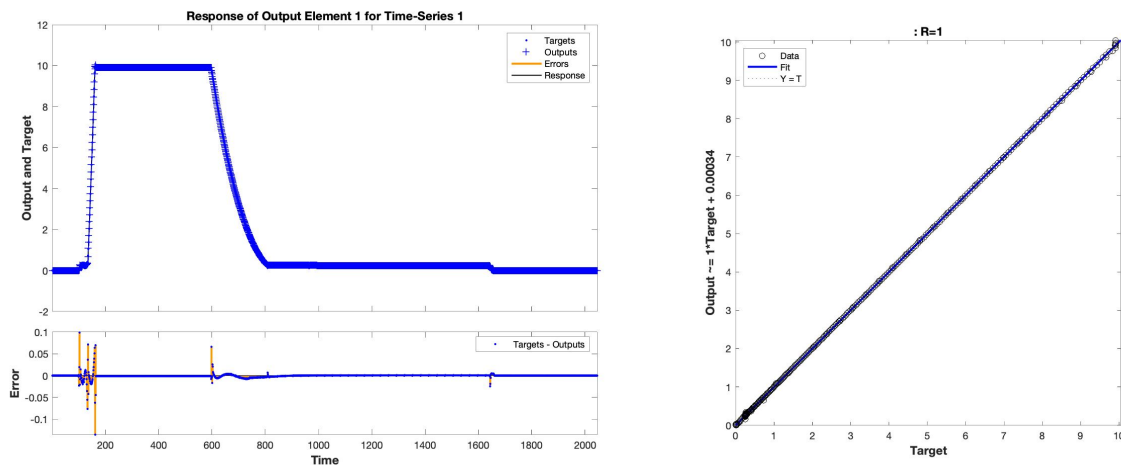


Figure 4.7 – Targets, outputs, error and response & correlation for validation data (VAL04).

All in all, the model that came out as best performer does a good job at predicting the response of the targets, specially when compared with the non-linear approach and also the other neural networks trained.

4.6.2 Assessing for Run Speed

The main proposed objective of this work is to investigate if a model would improve the run time of the current algorithm for brake cylinder pressure estimation. For this, the search sweep also recorded the fastest trained neural network. The table bellow 4.2 shows the summary of the accuracy and run time for the fastest trained neural network. Note that the error is an order of magnitude higher than the previous model with best accuracy - $10\text{E-}3$ against $10\text{E-}4$. The overall average error is $5.2177\text{E-}4$ (mean square error) and the average time it takes to run is 0.1917s , or about 42% less than the ANN that exhibited the best accuracy trained network. The training time was 11.34s , lower than the previous model by about 3.5 times. It is important to be noted that the previous neural network finished training by having hit the max iteration count not by having achieved the desired accuracy. The results for each case will be presented and discussed in a similar manner to the other cases. The fastest network uses no delay on the input nor output, i.e. it uses $x(n)$ and $y(n)$ and 5 hidden neurons.

	Error (mse) $10\text{E-}3$	time
Est01	0.5884	0.3760
Val01	0.5025	0.3715
Val02	0.1980	0.0753
Val03	0.5454	0.0924
Val04	0.7746	0.0435

Table 4.2 – Accuracy (error) and time results for the fastest running neural network.

4.6.2.1 EST01

The accuracy of the fastest run time is not as good as the neural network that had the best accuracy and this difference will be clear on the result figures. The overall plot of the target over the model output seems to indicate a good agreement between the two parameters. At closer inspection, it is visible that the model is off at the minimum pressure application on the brake cylinder. There is also some error at the pressure rise (figure 4.8). The error is also double of the previous trained network, specially at the pressure changes. The target x estimation plot show a certain distribution around the correlation between the model and the target.

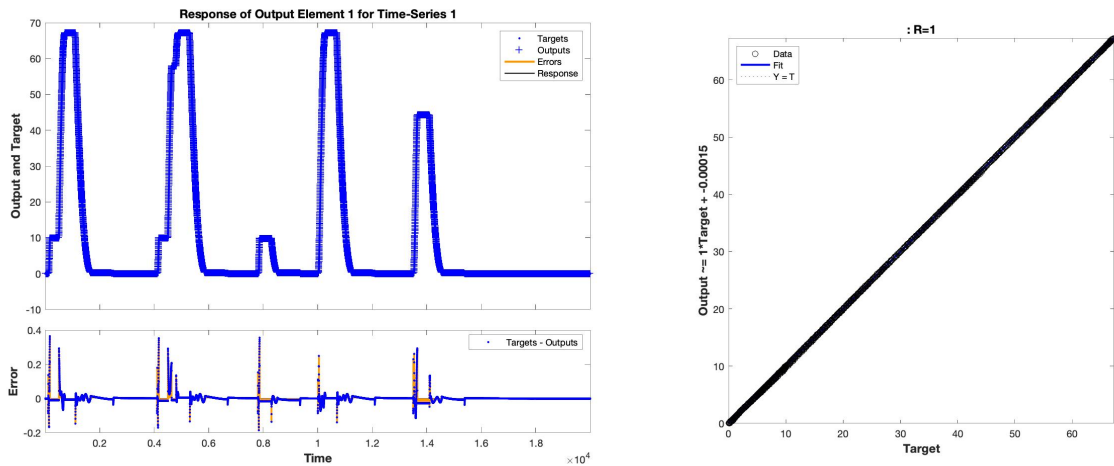


Figure 4.8 – Targets, outputs, error and response & correlation for estimation data (EST01).

4.6.2.2 Validation Datasets

The validation results tell a story similar to the estimation dataset. Again, the best performing is validation 2 (VAL02), that is the highest pressure at the brake cylinder - the same bias demonstrated previously on the linear models. This effect may be due to the makeup of the estimation data - with more full brake application than minimum brake application.

The Neural Network topology and training has the peculiarity of presenting an overshoot at the end of the application, as seen on figure 4.9 and there is some fluctuation at the beginning of the pressure change slope. Errors for the VAL01 data are in line with EST01. Correlation pictures show the expected results.

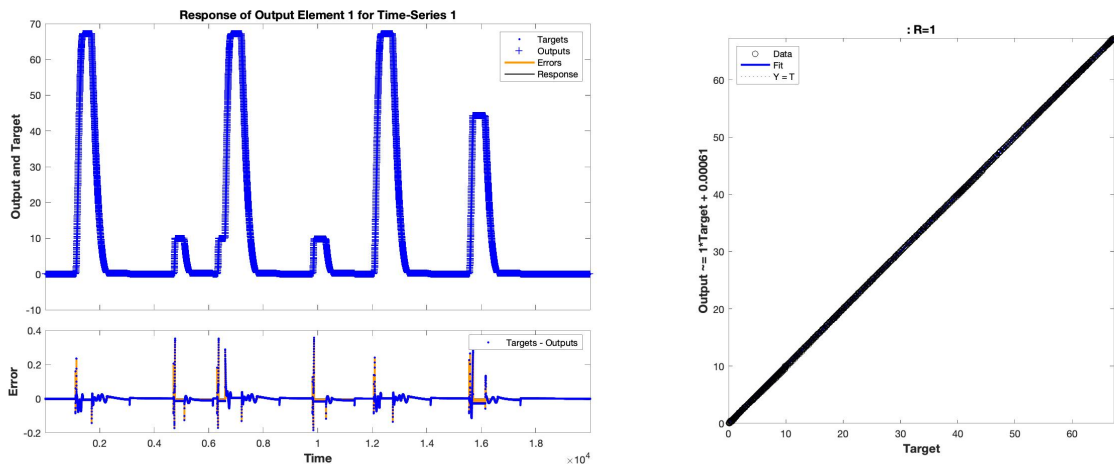


Figure 4.9 – Targets, outputs, error and response for estimation data (VAL01).

The other validation datasets show similar results telling the same story, speed over

accuracy. Yet, the reduction in accuracy is offsetted by the decrease in running time. Figures 4.10 to 4.12 show the target vs model results and the errors on the response. The correlation figure shows a good agreement between the model and target.

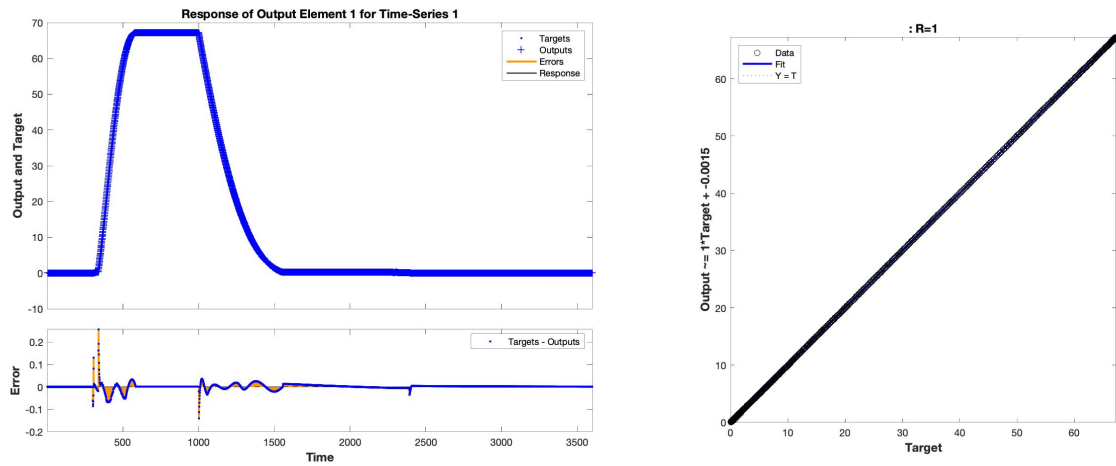


Figure 4.10 – Targets, outputs, error and response & correlation for estimation data (VAL02).

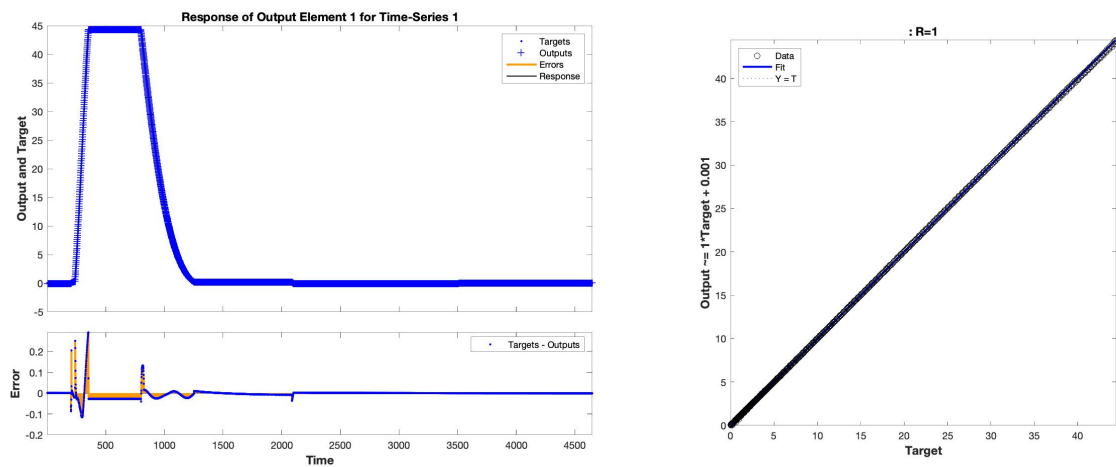


Figure 4.11 – Targets, outputs, error and response & correlation for estimation data (VAL03).

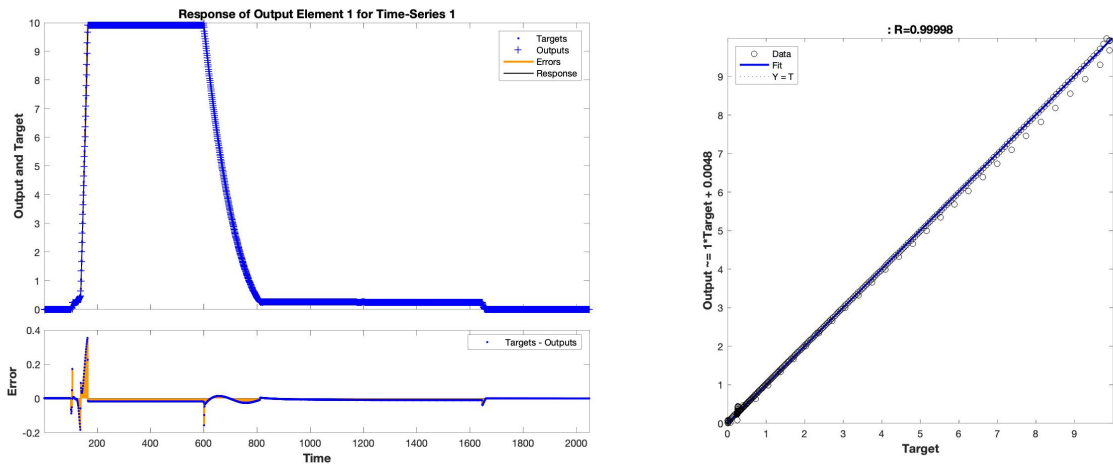


Figure 4.12 – Targets, outputs, error and response & correlation for estimation data (VAL04).

4.6.3 Best Combination between speed and accuracy

The sweep algorithm also had provisions to record a third topology/training combination. It multiplied the mean accuracy parameter with the mean running time and recorded the lowest value. In this sweep, the best accuracy combination of topology and training had also the lowest combination of both parameters.

4.6.4 Other results

4.6.4.1 Overall Performance

As mentioned, the sweep resulted in 500 iterations. These are organised on groups of delays on the input and output and on layer depth. Nonetheless, one can plot the accuracy over the whole exercise (figure 4.13). The lowest error, and therefore highest accuracy is iteration 380 (input delay: 4; output delay: 4, layer depth: 20) that was discussed earlier. There are other combinations that provided good results are on the table 4.3

	input delay	output delay	hidden layer	Average mean square error (mse)	Mean Running time [s]	iteration
1	4	4	20	2.5176E-5	0.3351	380
2	3	4	20	2.6821E-5	0.3350	280
3	3	3	17	2.7214E-5	0.3181	257
4	2	4	17	2.7311E-5	0.3702	177
5	2	3	18	2.7931E-5	0.3520	158

Table 4.3 – Best mean accuracy combination.

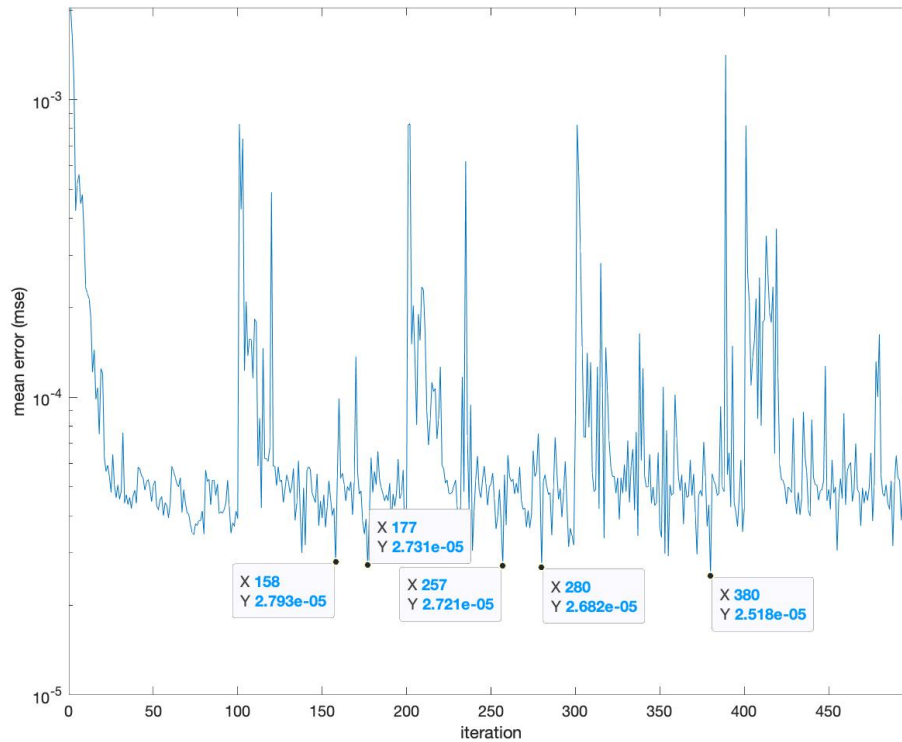


Figure 4.13 – Accuracy over the sweep.

4.6.4.2 Overall running time

The running timer over the iterations is plotted on figure 4.14. There is a clear discontinuity on the mean running time when the delay is larger than 1. This is due to the fact that the model has to wait for the other results. This phenomenon appears on the other sweep sets as well, every centennial iteration, that marks the reset of the layer quantity. At the beginning of the hundred iteration package, the way the sweep is constructed, there is an increase on the delay of the output, i.e. the more the delay on the output, the longer the running time - what is understandable. Also, there is another trend on the time the model takes to run and it is related to the number of hidden neurons. The lowest running times are located at the lowest delays and these are presented on table 5.2. The hardware used is on appendix 2

	input delay	output delay	hidden layer	Average mean square error (mse)	Mean Running time [s]	iteration
1	1	1	5	5.2177E-4	0.1917	5
2	1	1	9	3.544E-4	0.1935	9
3	1	1	13	1.857E-4	0.1940	13
4	1	1	8	4.8008E-4	0.1945	8
5	1	1	12	2.1426E-4	0.1950	12

Table 4.4 – Fastest mean running time combination.

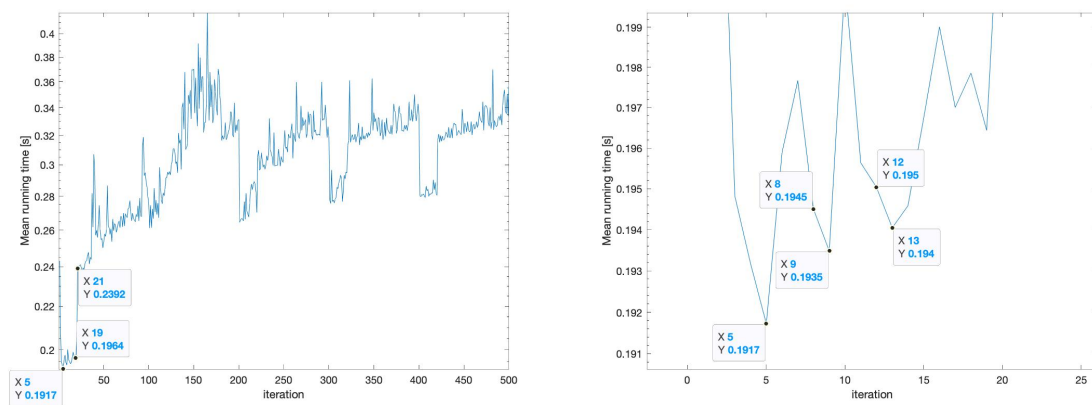


Figure 4.14 – Accuracy over the sweep.

5 MODEL DEPLOYMENT

Having chosen the most adequate method and determined the process to find the best parameter combination, it is time to build the other models outlined on the brake topology (figure 5.1) and continue on the strategy outlined previously (figure 5.2).

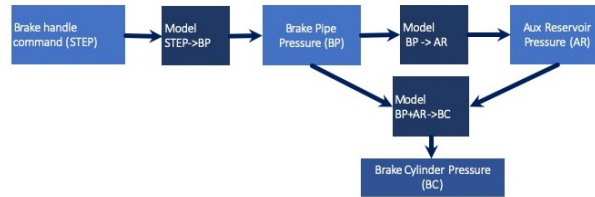


Figure 5.1 – Problem topology and required models.

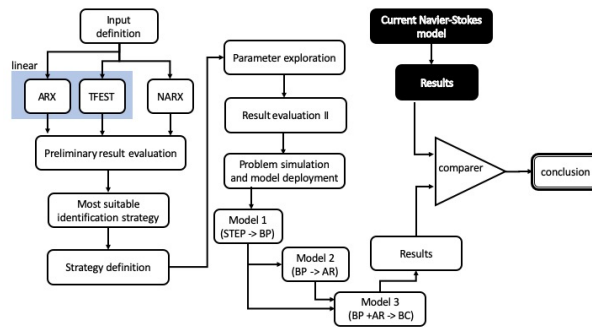


Figure 5.2 – Overall strategy to address the problem.

5.1 Brake Pipe + Aux. Reservoir to Brake Cylinder: BP + AR -> CF

On this section, the model between the brake pipe and the auxiliary reservoir pressure and the brake cylinder pressure will be explored.

5.1.1 Closing the Neural Network

As standard procedure, the artificial neural network is to be trained in the open state. This training uses the target output as the input for the trained model. Training the network on the open state allows for faster training times. However, for the deployment of the ANN, it must be closed - it would not be interesting to require the target output for every new model - what defeats the purpose.

Assuming the most accurate network trained on the previous chapter (figure 5.3).

The command

```
net=closeloop(net)
```

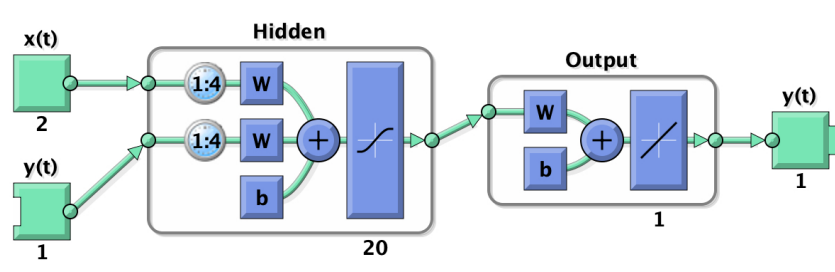


Figure 5.3 – Most accurate ANN for BP+AR->BC.

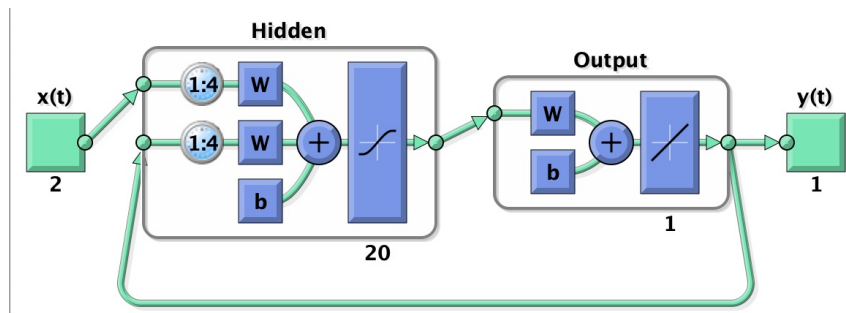


Figure 5.4 – Most accurate ANN for BP+AR->BC on the closed version.

The results obtained with the closed network are not usable. After the loop has been closed and the model is forced to use the calculated outputs $y(n)$ to predict $y(n+1)$, it loses the ability to predict the phenomenon. Therefore, for the problem at hand, a different approach has to be employed in order to obtain proper results. Figure 5.5 show the graphs for the estimation data. Figures 5.6 to 5.9 show the results for the rest of the validation cases the closer case is VAL04, but still none of the results are usable.

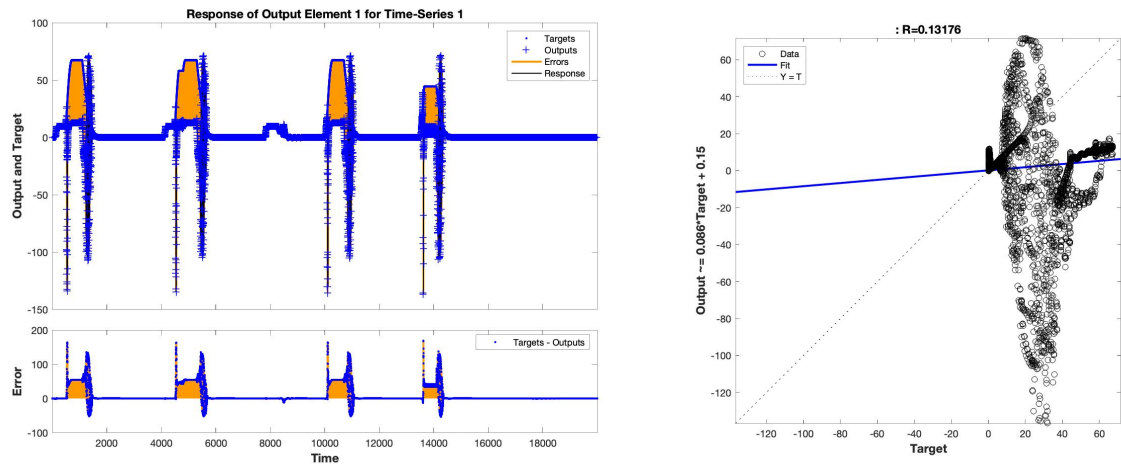


Figure 5.5 – Targets, outputs, error and response for estimation data & Target x estimation correlation (EST01).

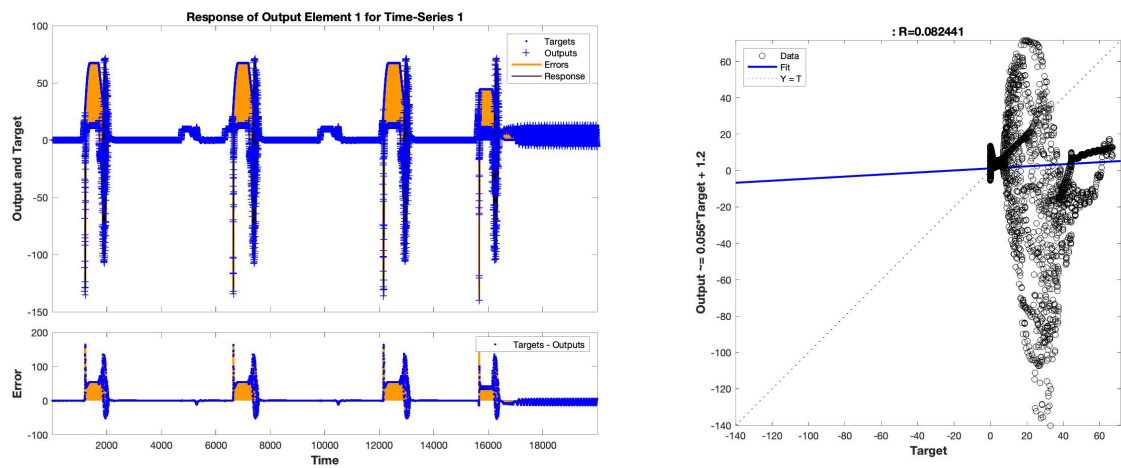


Figure 5.6 – Targets, outputs, error and response for estimation data & Target x estimation correlation (VAL01).

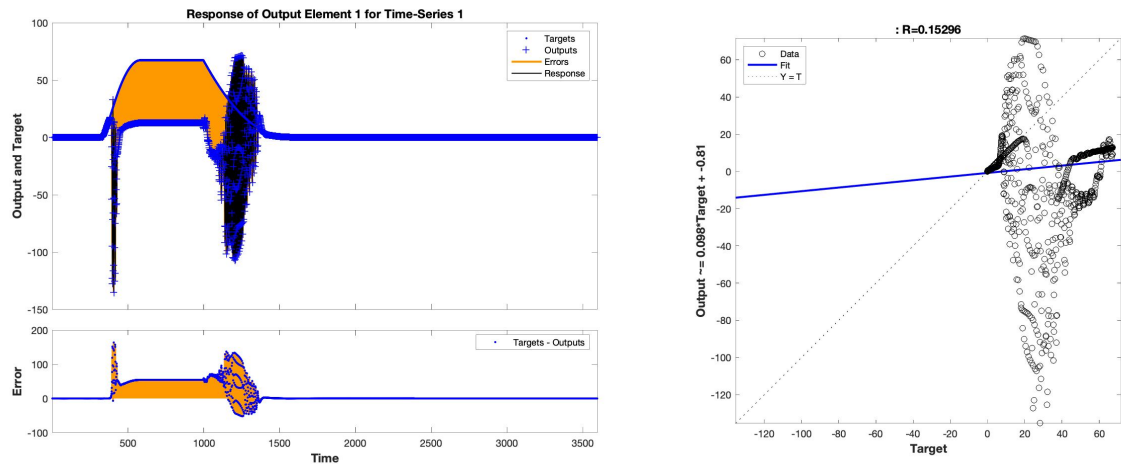


Figure 5.7 – Targets, outputs, error and response for estimation data & Target x estimation correlation (VAL02).

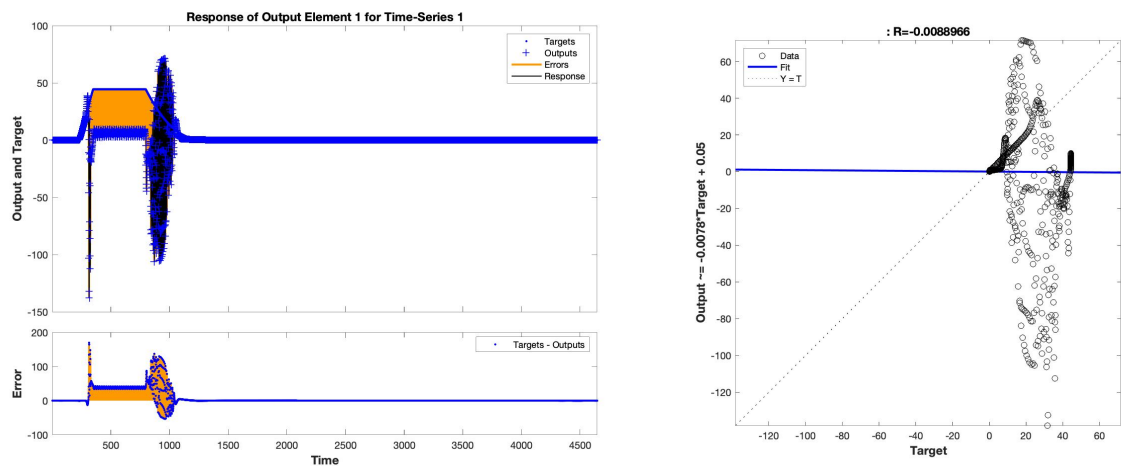


Figure 5.8 – Targets, outputs, error and response for estimation data & Target x estimation correlation (VAL03).

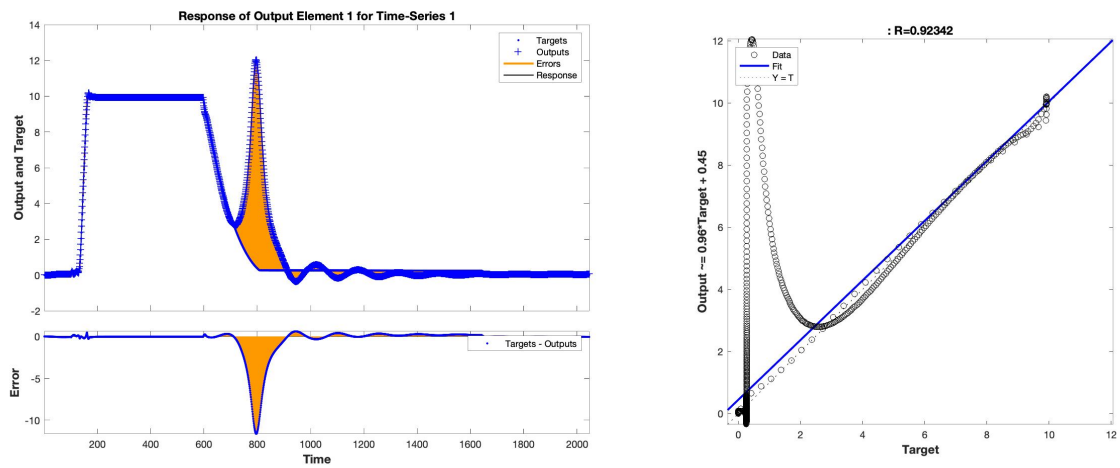


Figure 5.9 – Targets, outputs, error and response for estimation data & Target x estimation correlation (VAL04).

5.1.2 Training the Network Closed BP + AR -> BC

Since the model that is being worked on is the one that predicts the brake cylinder pressure from the brake pipe and auxiliary reservoir pressure this will be the first to be trained in the closed loop form. The same combination of delays (4 for input, 4 for output) and number of neurons (20) was used. The only difference on the training scheme is that the network is closed before the training takes place and that the training method is different.

The training in the close loop mode takes longer - the same network that took 40s to train in the open mode now took 4500s to train. The Levenberg-Marquardt backpropagation training scheme has not yielded good results, so alternatively, the Bayesian regularization backpropagation was used (*trainbr* on MATLAB instead of *trainlm*). The closed loop also has higher error in relation to the open loop model that uses the target to find $y(n + 1)$, but that was expected. A good side effect was that the running time was reduced compared with the open loop model. Table 5.3 shows the value comparison for error and run time between the models. Figures 5.10 to 5.14 shows the various graphs for the cases generated by the model output. Though not as good as the previous open model, the results are usable.

Unfortunately, as the training times are very long on the current hardware (single processor) it is not feasible to do a parameter sweep with the closed loop model. It might be possible to employ such technique with gpu multithreading or server grade installations, however, those are not available at this point in time.

case	Error - mse (10E-3)		run time [s]	
	close loop (retrained)	open loop	close loop (retrained)	open loop
EST01	4.149	0.0223	0.203	0.686
VAL01	5.656	0.0272	0.165	0.650
VAL02	2.742	0.0097	0.036	0.119
VAL03	3.012	0.0241	0.059	0.151
VAL04	15.865	0.0423	0.024	0.069

Table 5.1 – Open and close loop comparison.

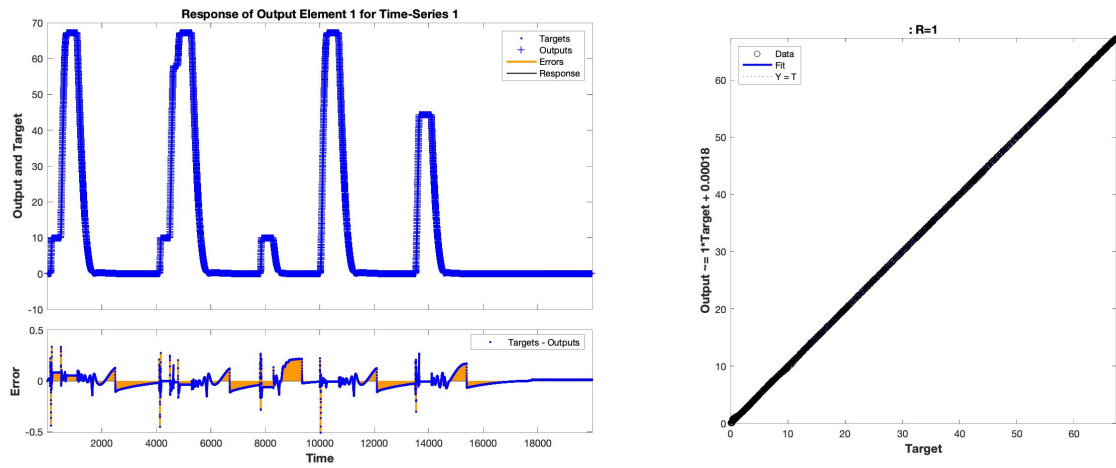


Figure 5.10 – Targets, outputs, error and response for estimation data & Target x estimation correlation (EST01).

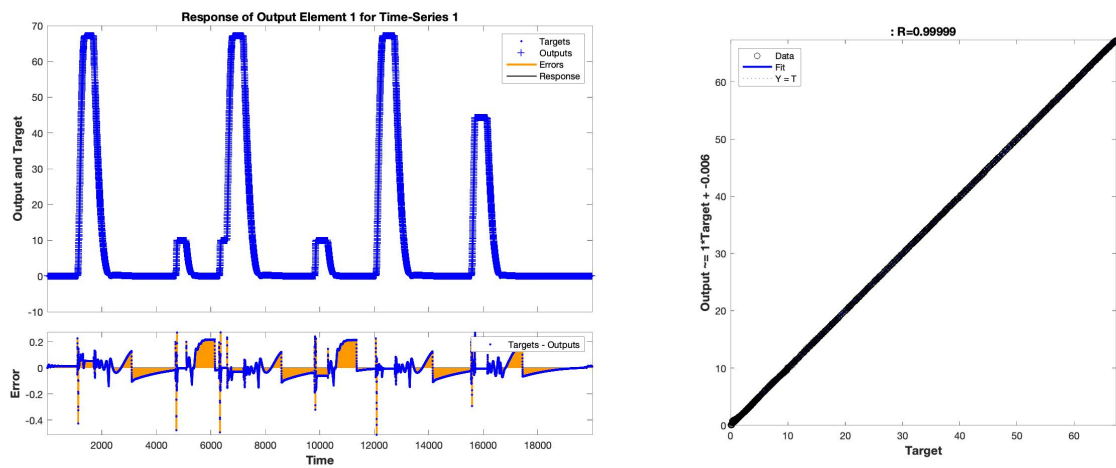


Figure 5.11 – Targets, outputs, error and response for estimation data & Target x estimation correlation (VAL01).

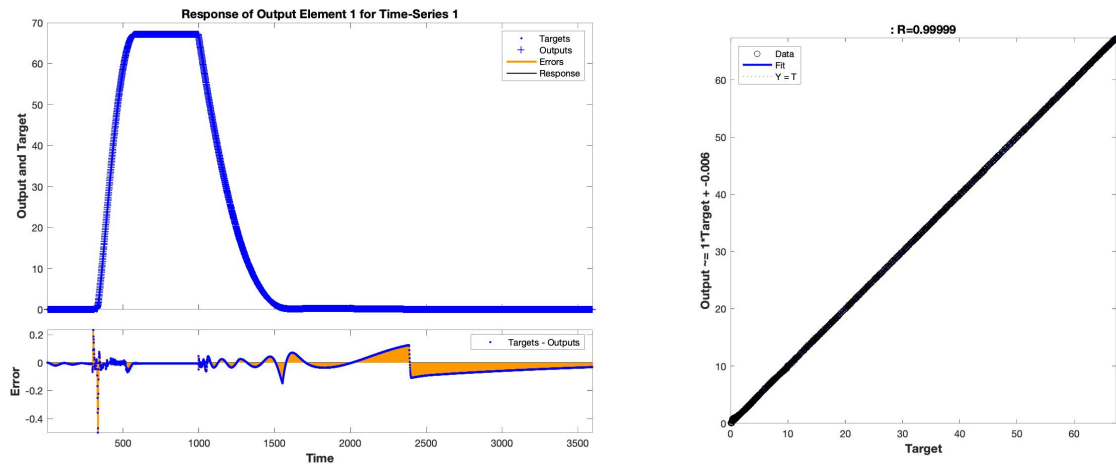


Figure 5.12 – Targets, outputs, error and response for estimation data & Target x estimation correlation (VAL02).

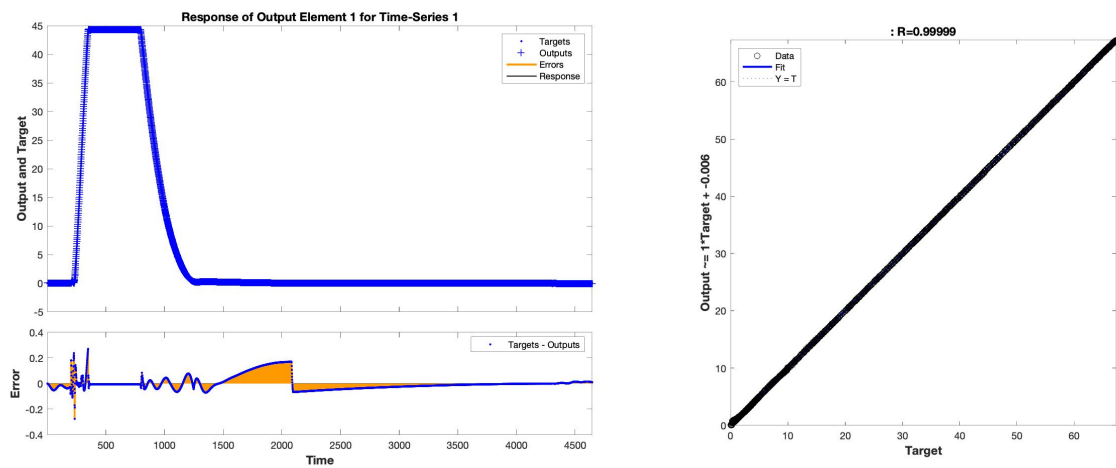


Figure 5.13 – Targets, outputs, error and response for estimation data & Target x estimation correlation (VAL03).

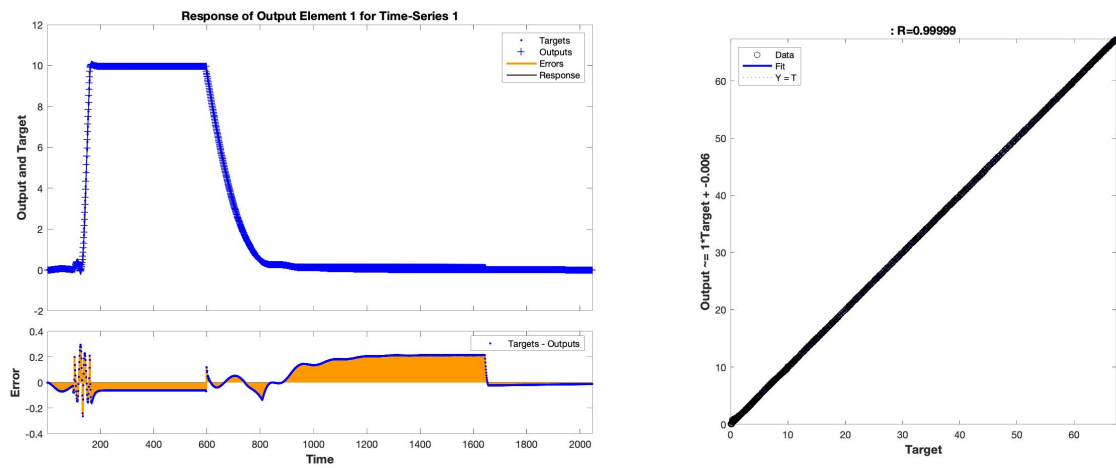


Figure 5.14 – Targets, outputs, error and response for estimation data & Target x estimation correlation (VAL04).

5.2 From Brake Handle to Brake Pipe: STP -> BP

On this section the model between the brake handle of the 26L style brake system to the pressure on the brake pipe will be explored. The first part of this endeavour is to build the step signal. It starts with the matrix that originated the inputs that are being used since chapter 3 and a simple algorithm to transform the discrete points into a time series. Figure 5.15 shows the graphical representation of the matrixes.

5.2.1 Command Matrix

5.2.1.1 EST01

```
vector_EST01=[ 0 90
               100 84.2
               500 64.2
               1100 90
               4100 84.2
               4500 70
               4800 64.2
               5300 90
               7800 84.2
               8300 90
               10000 64.2
               10700 90
               13500 74
               14100 90
               20000 90];
```

5.2.1.2 VAL01

```
vector_VAL01=[ 0 90
               1100 64.2
               1700 90
               4700 84.2
```



```
5100 90
6300 84.2
6600 64.2
7200 90
9800 84.2
10300 90
12050 64.2
12750 90
15550 74
16150 90
20000 90];
```

5.2.1.3 VAL02

```
vector_VAL02=[ 0 90
               300 64.2
               1000 90
               3600 90];
```

5.2.1.4 VAL03

```
vector_VAL03=[ 0 90
               200 74
               800 90
               4650 90];
```

5.2.1.5 VAL04

```
vector_VAL04=[ 0 90
```

```
100 84.2  
600 90  
2050 90];
```

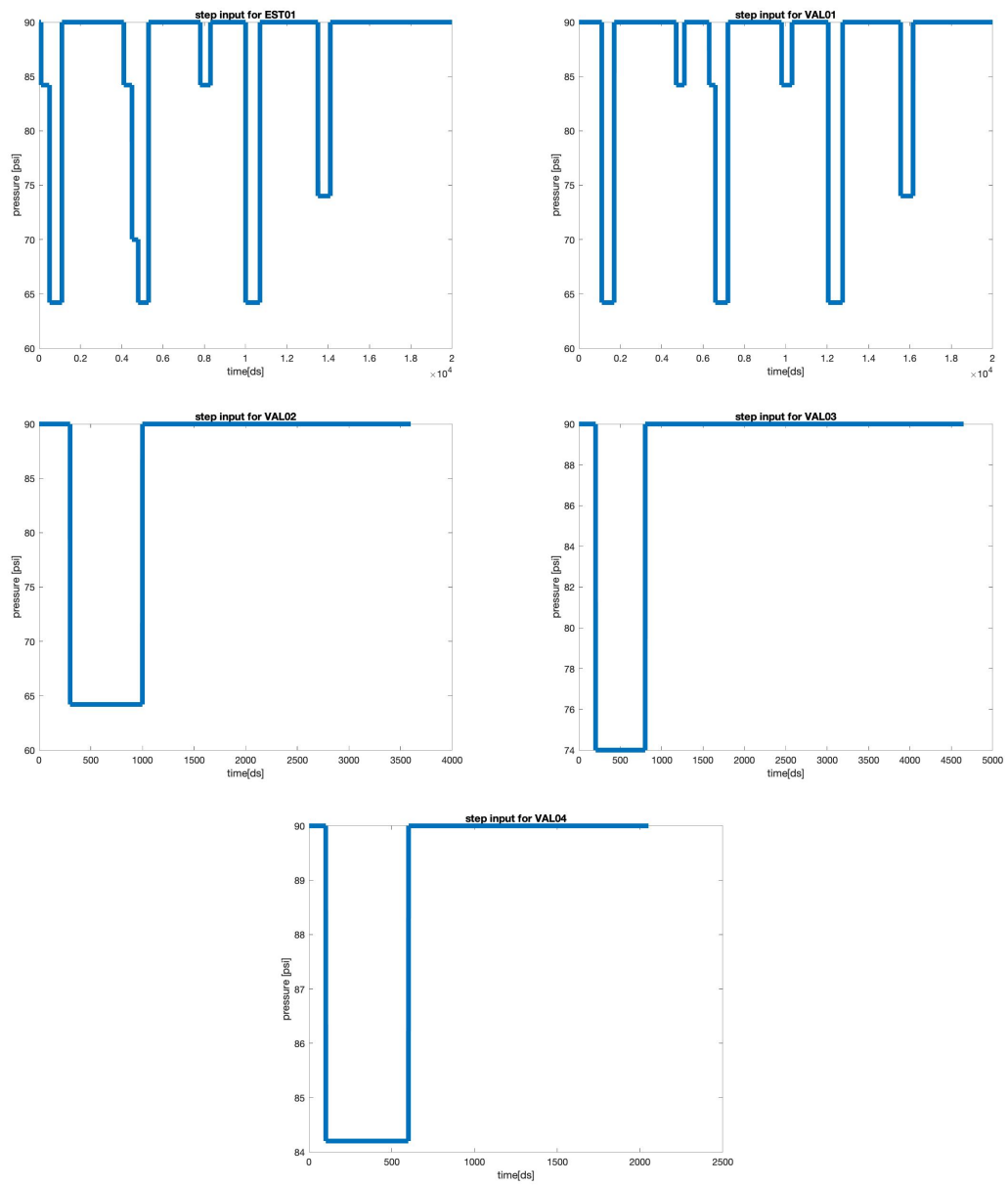


Figure 5.15 – Time series for STEP function.

5.2.2 Algorithm to Build Step Time Series

The following algorithm was used to transform the isolated matrix points into a time series

```

01 function output=step(input)
02 vector=input
03 [b c]=size(vector);
04 for id=1:(b-1)
05     a1=vector(id,1)+1;
06     a2=vector(id+1,1);
07     b1=vector(id,2);
08     for ik=a1:a2
09         output(ik,1)=b1;
10     end
11 end
12 end

```

5.2.3 STP -> BP Model Exploration - Open Loop

The same was that was done with the previous model, the training and permutations were done for the model that correlates the STEP commands and the brake pressure pipes. The most accurate model to arise from that training is the one with 2 delays on the input, 5 delays on the output and 15 neurons. It took 3.562s to train the model in the open loop configuration using the Levenberg-Marquardt strategy (*trainlm*). The model representation is shown on figure 5.16. Table 5.2 shows the values for error (mean square error) and run time for each case. Figure 5.17 to 5.21 show the results of the model over the target. The model does well for all cases except VAL04, where it overshoots on the recovery going above 90 psi. Should this persist, actions can be taken to prevent this from happening.

	Error (mse) 10E-3	run time [s]
Est01	4.632	0.575
Val01	6.595	0.577
Val02	4.206	0.108
Val03	2.969	0.135
Val04	6.015	0.066

Table 5.2 – Accuracy (error) and time results for the most accurate neural network (STP->BP - Open).

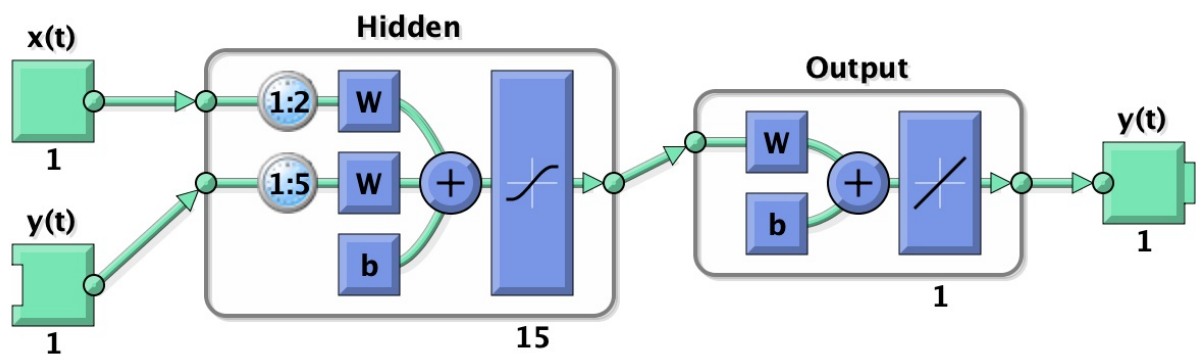


Figure 5.16 – Model representation for STP -> BP (open loop).

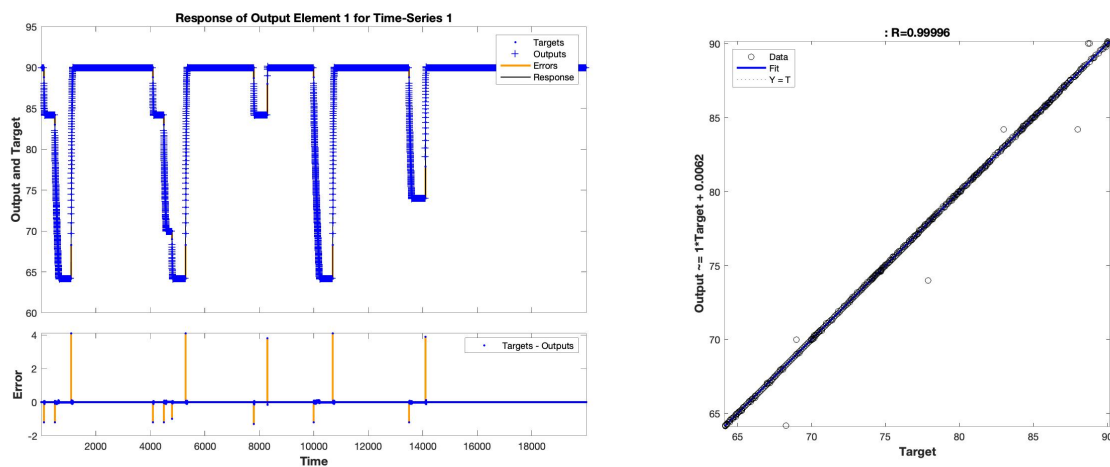


Figure 5.17 – Targets, outputs, error and response for estimation data & Target x estimation correlation (EST01).

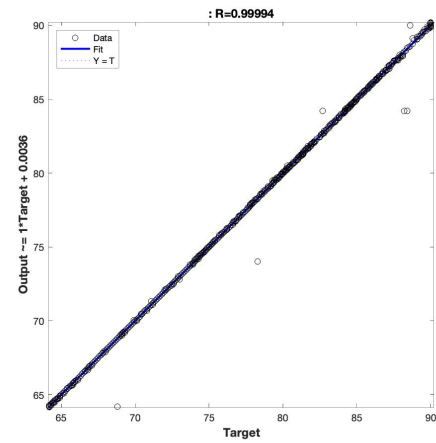
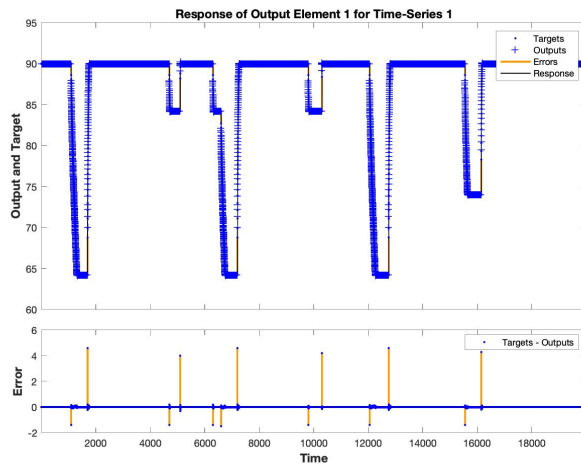


Figure 5.18 – Targets, outputs, error and response for estimation data & Target x estimation correlation (VAL01).

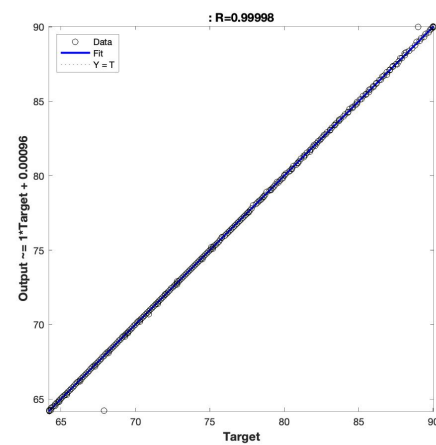
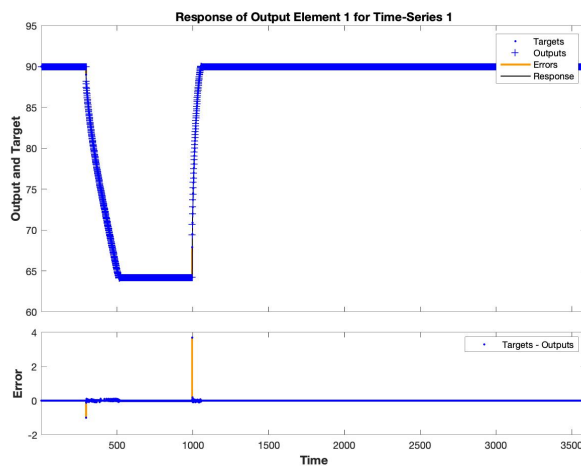


Figure 5.19 – Targets, outputs, error and response for estimation data & Target x estimation correlation (VAL02).

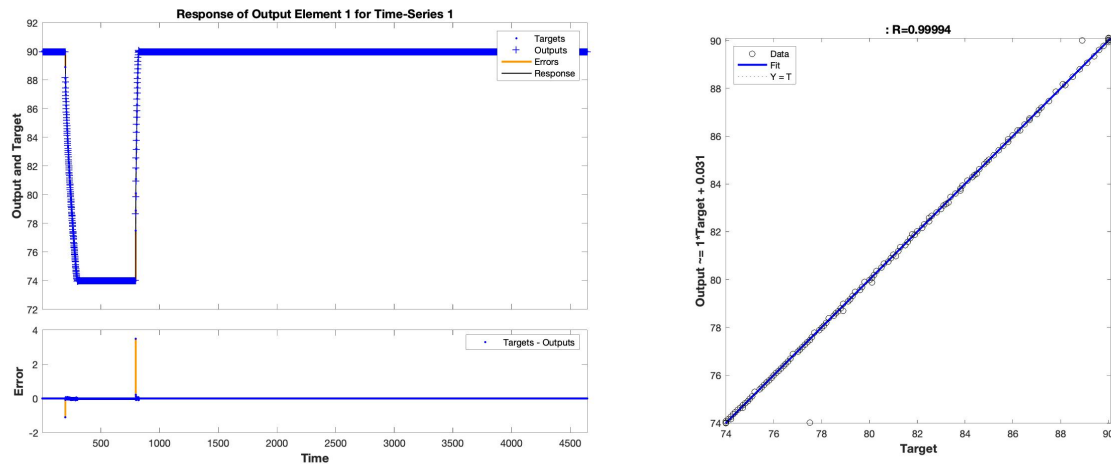


Figure 5.20 – targets, outputs, error and response for estimation data & Target x estimation correlation (VAL03).

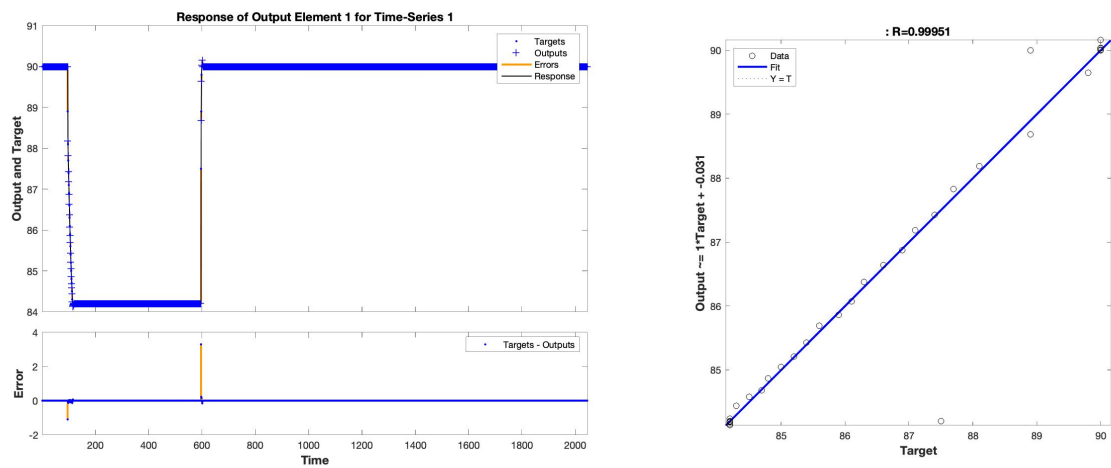


Figure 5.21 – Targets, outputs, error and response for estimation data & Target x estimation correlation (VAL04).

5.2.4 Closing the Loop

On this step, the operation $(net) = closeloop(net)$ is performed on the trained open network model for STP -> BP. The result summary is on table 5.3. The overall format of the model output has been preserved, however, the best accuracy now is VAL04. The model struggles at the slopes and the error has deteriorated greatly. The correlation shows some scattering on the results, though the R squared seem to be in good shape. Similarly to what was done on the previous model a new training on this new topology will be done and the results assessed. Figure 5.22 to 5.27 show the model output over target as well as the correlation. The running times have improved from due to the topology change.

case	Error - mse (10E-3)		run time [s]	
	close loop	open loop	close loop	open loop
EST01	124.287	4.632	0.2454	0.575
VAL01	128.228	6.595	0.1487	0.577
VAL02	76.191	4.206	0.0331	0.108
VAL03	194.73	2.969	0.0441	0.135
VAL04	41.001	6.015	0.0264	0.066

Table 5.3 – Accuracy (error) and time results for the most accurate neural network (STP->BP - Closed).

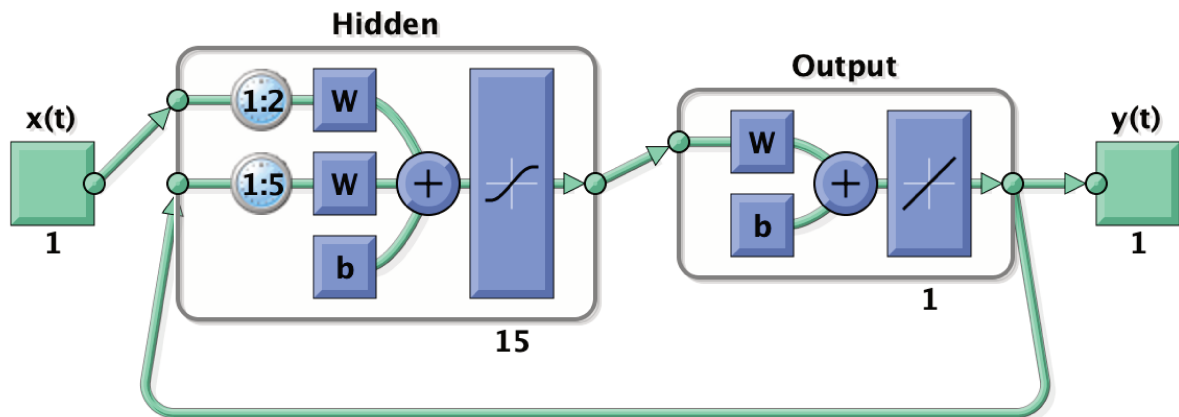


Figure 5.22 – Model representation for STP -> BP (closed loop).

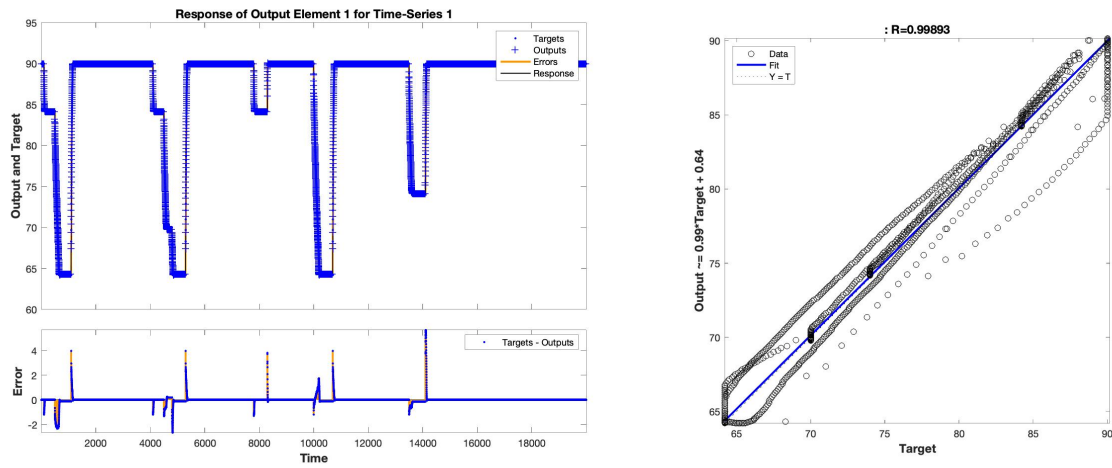


Figure 5.23 – Targets, outputs, error and response for estimation data & Target x estimation correlation (EST01).

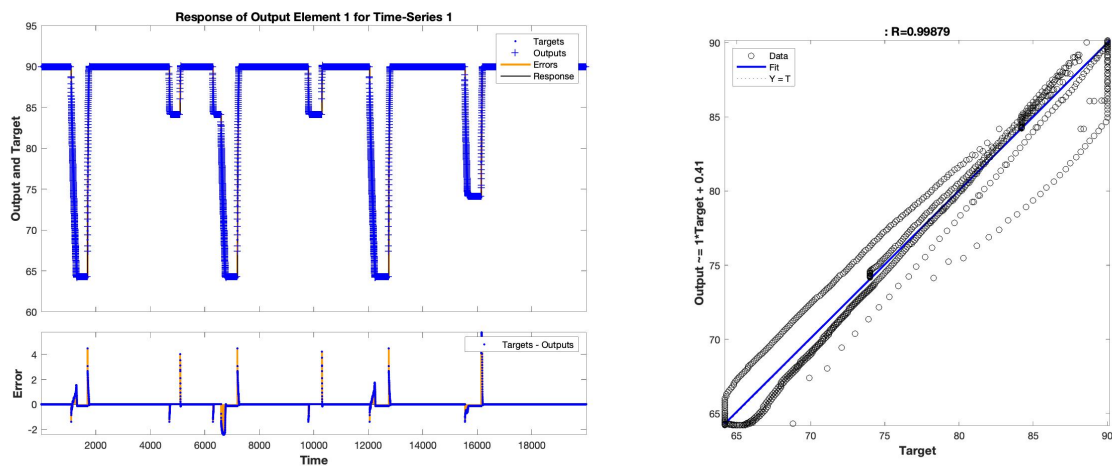


Figure 5.24 – Targets, outputs, error and response for estimation data & Target x estimation correlation (VAL01).

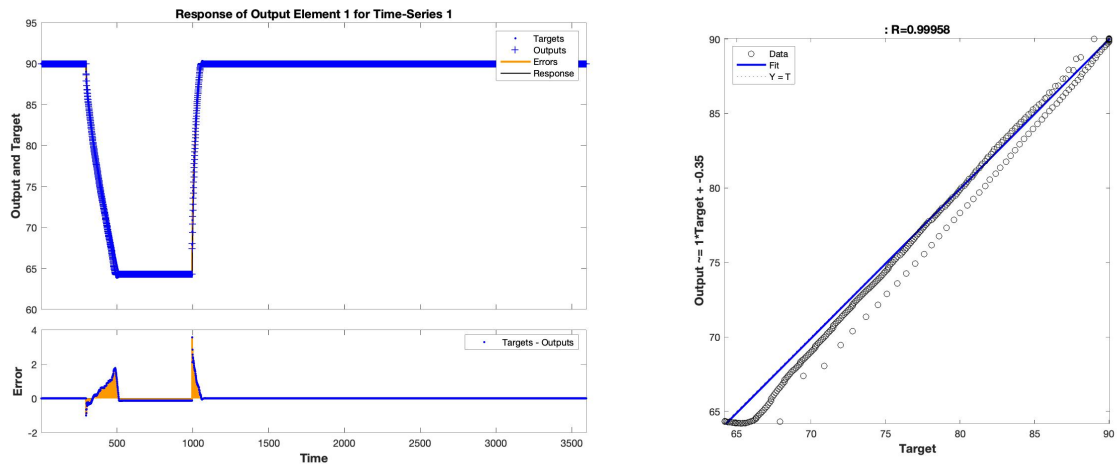


Figure 5.25 – Targets, outputs, error and response for estimation data & Target x estimation correlation (VAL02).

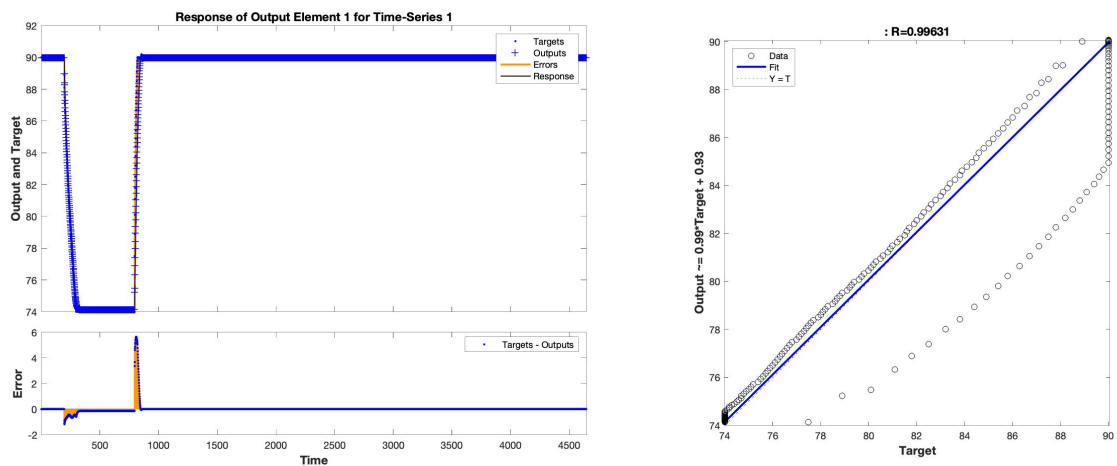


Figure 5.26 – Targets, outputs, error and response for estimation data & Target x estimation correlation (VAL03).

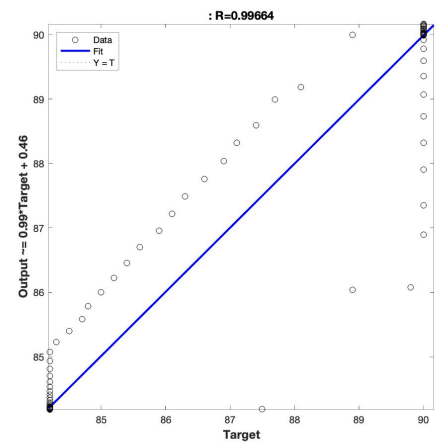
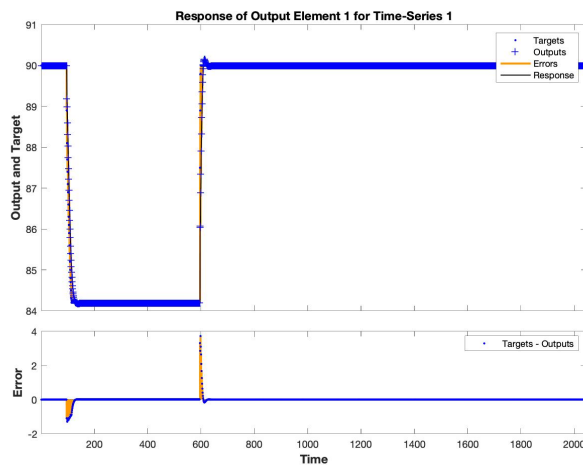


Figure 5.27 – Targets, outputs, error and response for estimation data & Target x estimation correlation (VAL04).

5.2.5 Retraining the Closed loop Model

For the retraining of the artificial neural network, the Levenberg-Marquardt with Bayesian regularization backpropagation (*trainbr*) was used instead of the standard Levenberg-Marquardt (*trainlm*). The former was not yielding good results in terms of training. Table 5.4 shows the improvement on the accuracy with the retraining in comparison to the closed loop keeping the open training. The closed loop retraining took 954.47 s compared with the 3.52 s of the open loop training. It would not be feasible with the current hardware/strategy to run a parameter sweep for the closed loop topology. The results shown on figures 5.28 to 5.32 shows the model response compared with the target. The issue on the VAL04 overshoot persists. One way to deal with this on the model implementation - if it causes adverse effects - would be to add an *if* line preventing the pressure to be above 90 psi. The 26L system could send the pressure of the brake pipe above 90 psi for the minimum normal pressure at the locomotive main compressed air reservoir is 120psi. So far, the results are acceptable for this model and we shall proceed to the last one - BP -> AR.

case	Error - mse (10E-3)			run time [s]		
	close loop retrained	close loop	open loop	close loop retrained	close loop	open loop
EST01	6.500	124.287	4.632	0.156	0.245	0.575
VAL01	8.571	128.228	6.595	0.150	0.149	0.577
VAL02	4.996	76.191	4.206	0.033	0.033	0.108
VAL03	6.412	194.730	2.969	0.041	0.044	0.135
VAL04	8.423	41.001	6.015	0.021	0.026	0.066

Table 5.4 – Accuracy (error) and time results for the most accurate neural network (STP->BP - Closed & Retrained).

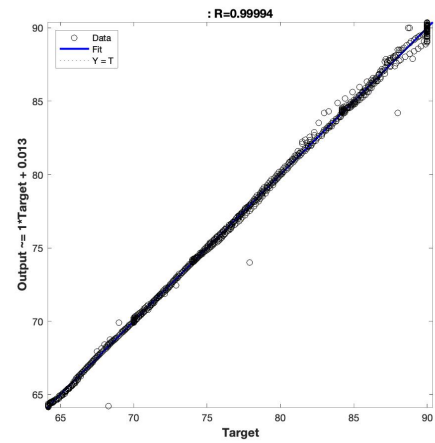
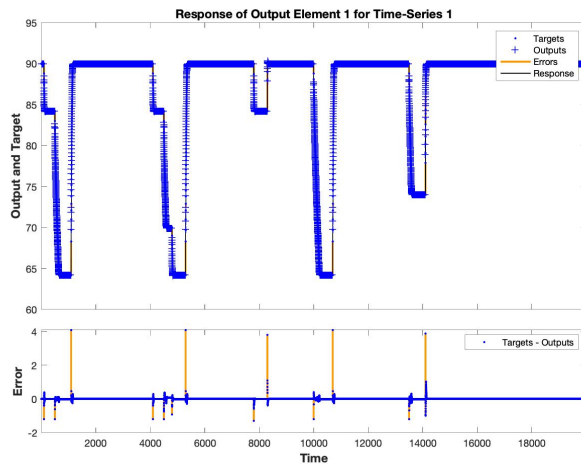


Figure 5.28 – Targets, outputs, error and response for estimation data & Target x estimation correlation (EST01).

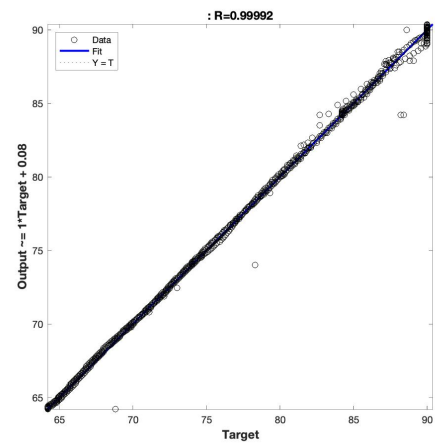
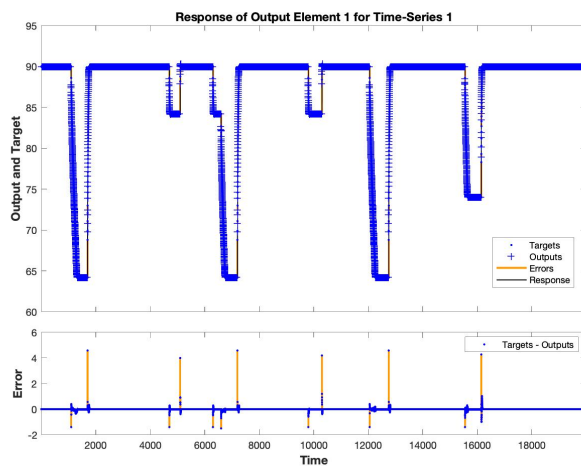


Figure 5.29 – Targets, outputs, error and response for estimation data & Target x estimation correlation (VAL01).

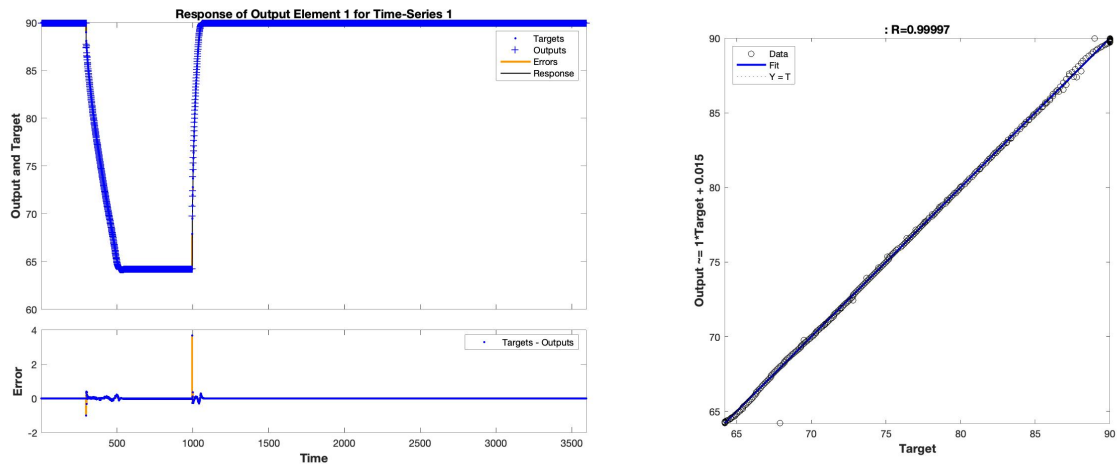


Figure 5.30 – Targets, outputs, error and response for estimation data & Target x estimation correlation (VAL02).

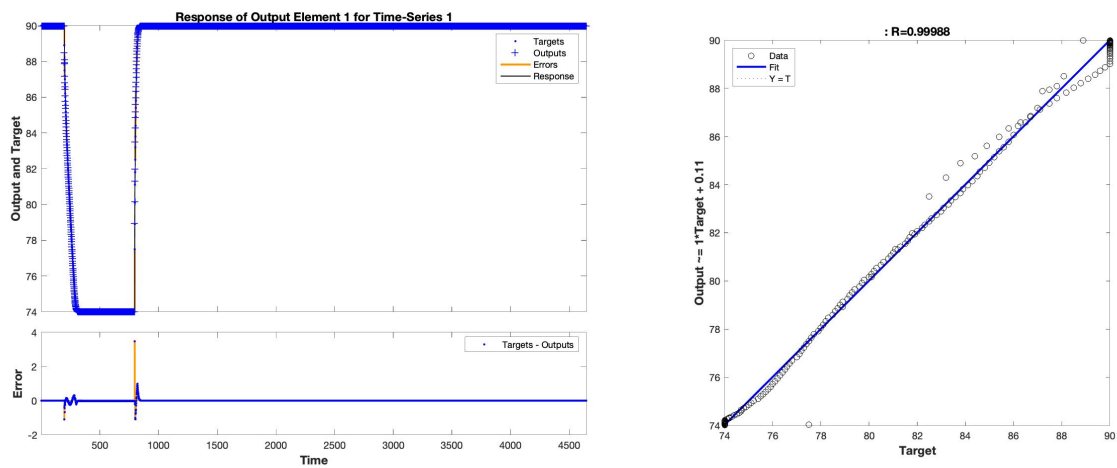


Figure 5.31 – Targets, outputs, error and response for estimation data & Target x estimation correlation (VAL03).

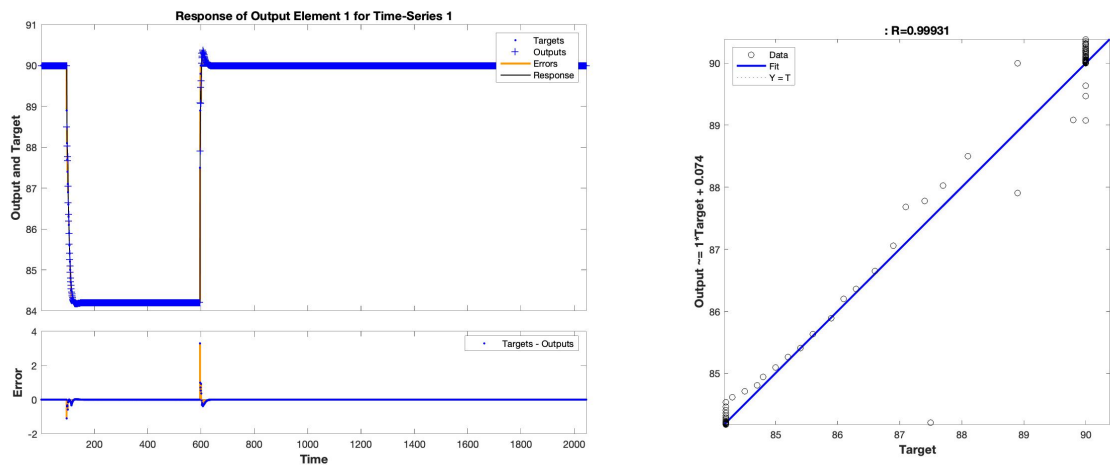


Figure 5.32 – Targets, outputs, error and response for estimation data & Target x estimation correlation (VAL04).

5.3 Brake Pipe & Auxiliary Reservoir Interaction: BP -> AR

Finally, the last model will describe the interaction between the brake pipe and the auxiliary reservoir pressure. As described previously, the ABDX valve compares the pressure at the auxiliary reservoir and the brake pipe and injects the difference in the brake cylinder. When releasing the brakes, the valve also uses the emergency reservoir to reestablish the pressure on the auxiliary reservoir and the brake pipe. In this study, the emergency reservoir pressure level is not considered directly. For the service brake model, the emergency reservoir is assumed to be full at all times. On the model identified in this study, this will be implicit on the behaviour of the auxiliary reservoir and the brake pipe. There would be a difference in the behaviour if the emergency reservoir is not full when the application starts. On the model with the emergency brake application added to the service brake, the emergency reservoir pressure will be addressed.

5.3.1 BP -> AR Model Exploration - Open Loop

The same method of parameter searching was used for this model. The most accurate model has 5 delays on the input, 4 delays on the output and 20 neurons. It sits uncomfortably close to the maximum prescribed values of 5 delays on the input and output and 20 neurons, however, the fit of the model is excellent as can be seen on table 5.5. Figure 5.33 shows the representation of the model. Figures 5.34 to 5.38 show the impressive agreement between model and target.

	Error (mse) 10E-3	run time [s]
Est01	0.00153	0.709
Val01	0.00183	0.602
Val02	0.00082	0.111
Val03	0.00129	0.142
Val04	0.00337	0.078

Table 5.5 – Accuracy (error) and run time results for the most accurate neural network (BP -> RA - Open Loop).

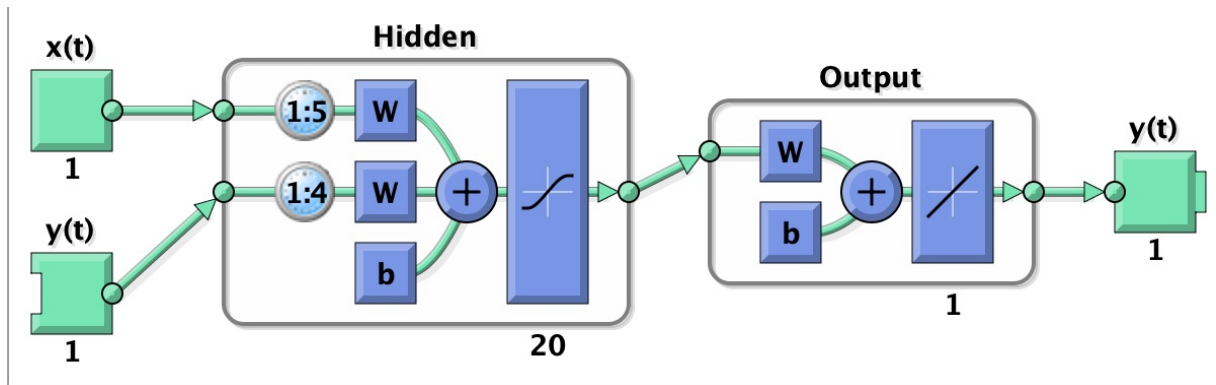


Figure 5.33 – Model representation for BP -> AR (open loop).

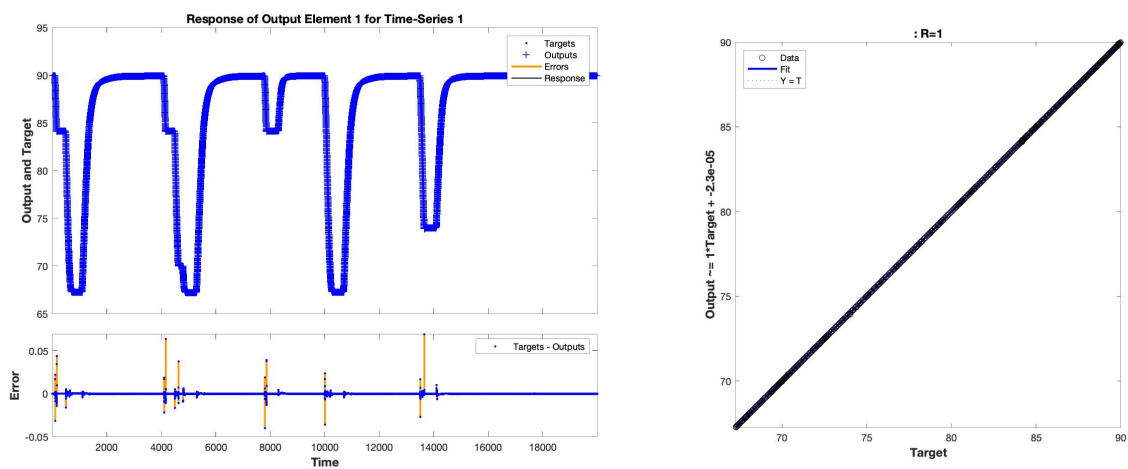


Figure 5.34 – Targets, outputs, error and response for estimation data & Target x estimation correlation (EST01).

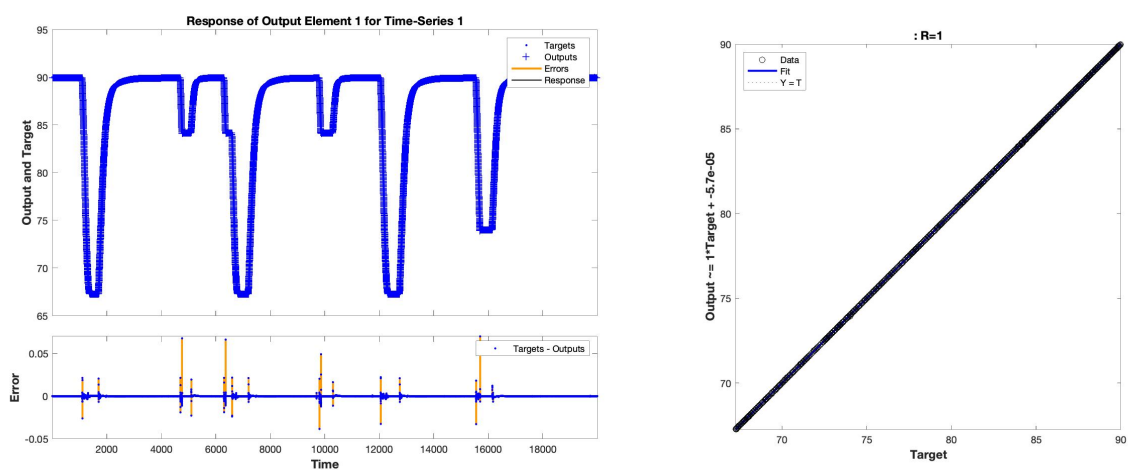


Figure 5.35 – Targets, outputs, error and response for estimation data & Target x estimation correlation (VAL01).

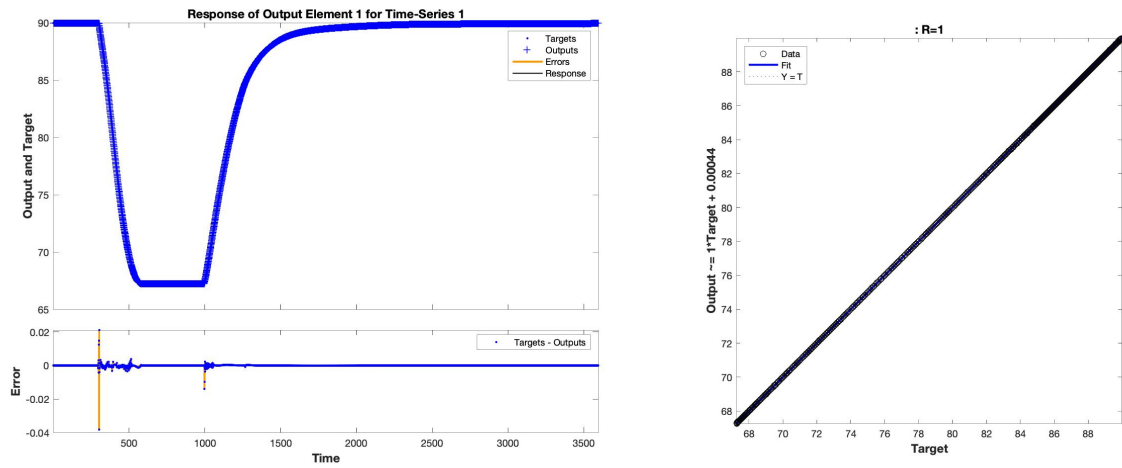


Figure 5.36 – Targets, outputs, error and response for estimation data & Target x estimation correlation (VAL02).

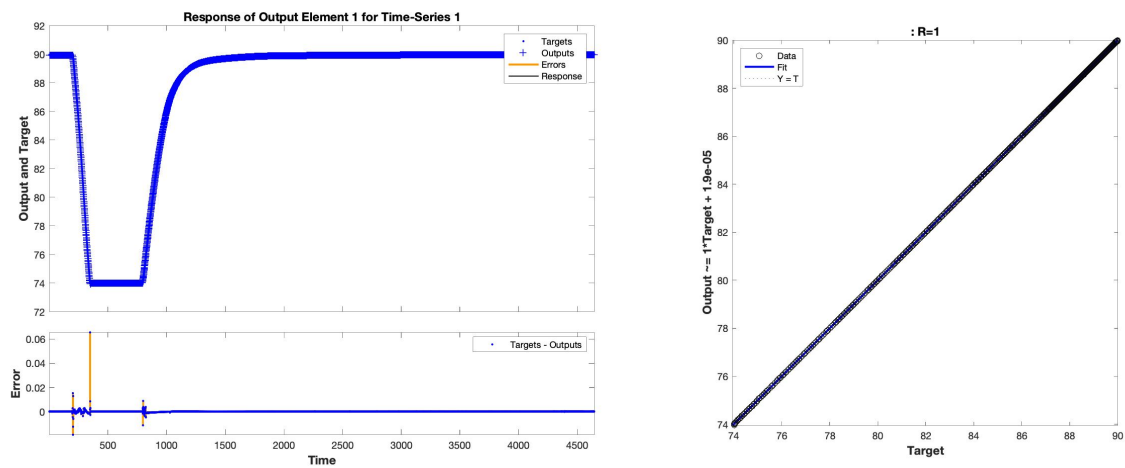


Figure 5.37 – Targets, outputs, error and response for estimation data & Target x estimation correlation (VAL03).

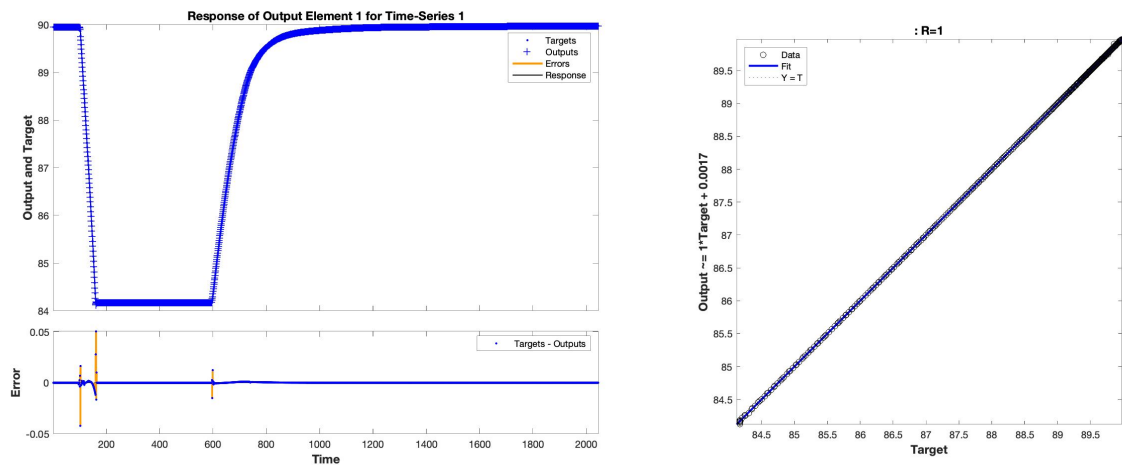


Figure 5.38 – Targets, outputs, error and response for estimation data & Target x estimation correlation (VAL04).

5.3.2 Closing the loop

Again, the same command was used to close the loop and obtain the results presented on table 5.6. The error increased by several orders of magnitude - the open loop had error in the order of $10E-6$ and closing the loop, the errors increase to the order of 10 - except on the minimum service application, where errors were compatible. Figure 5.39 shows the representation of the model with the closed loop. Figures 5.40 to 5.44 show that closing the model has effected its results severely. The only validation data set that retained a resemblance of matching the target was VAL04 - the minimum service application. Therefore, the network will be retrained on the closed topology.

case	Error - mse (10E-3)		run time [s]	
	close loop	open loop	close loop	open loop
EST01	1.449E4	0.00153	0.202	0.709
VAL01	1.3392E4	0.00183	0.139	0.602
VAL02	1.912E4	0.00082	0.031	0.111
VAL03	1.959E4	0.00129	0.037	0.142
VAL04	0.0036	0.00337	0.022	0.078

Table 5.6 – Accuracy (error) and time results for the most accurate neural network (BP -> AR - Closed).

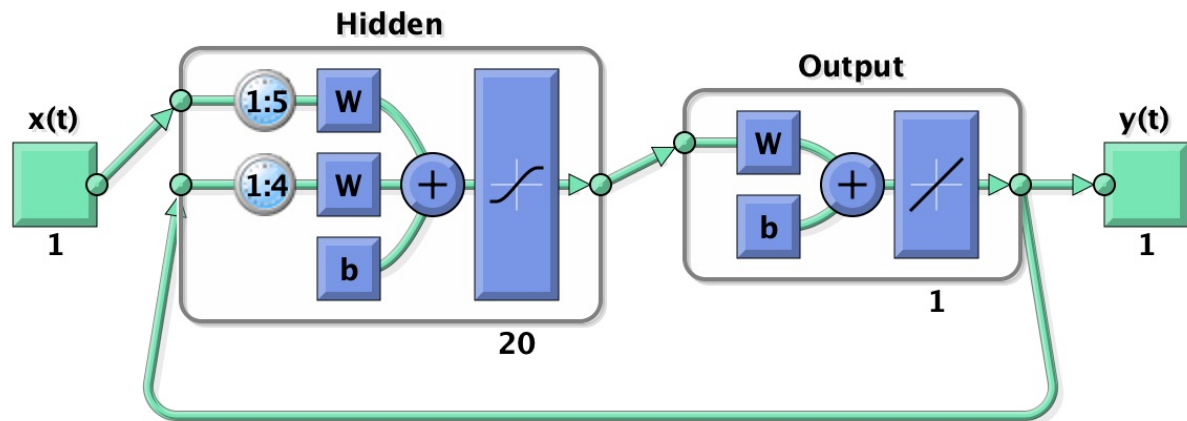


Figure 5.39 – Model representation for BP -> AR (closed loop).

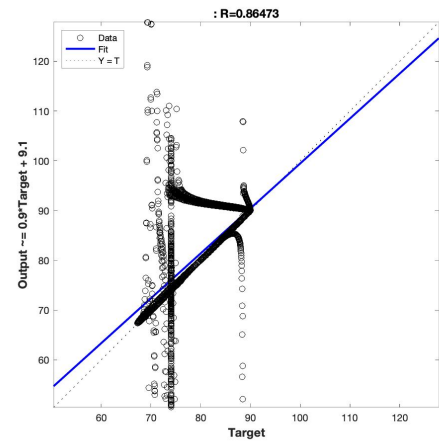
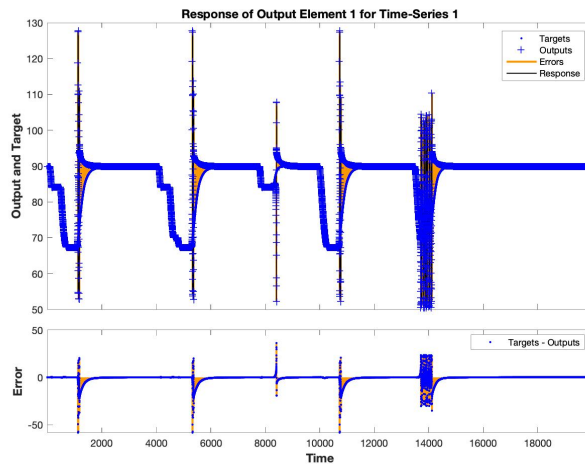


Figure 5.40 – Targets, outputs, error and response for estimation data & Target x estimation correlation (EST01).

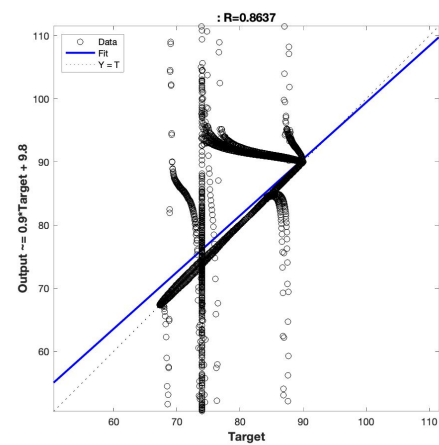
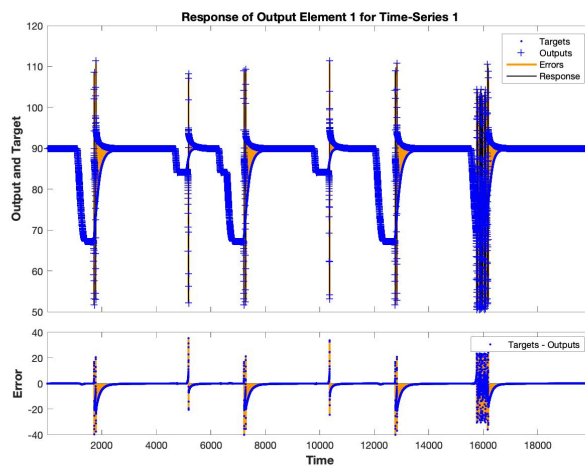


Figure 5.41 – Targets, outputs, error and response for estimation data & Target x estimation correlation (VAL01).

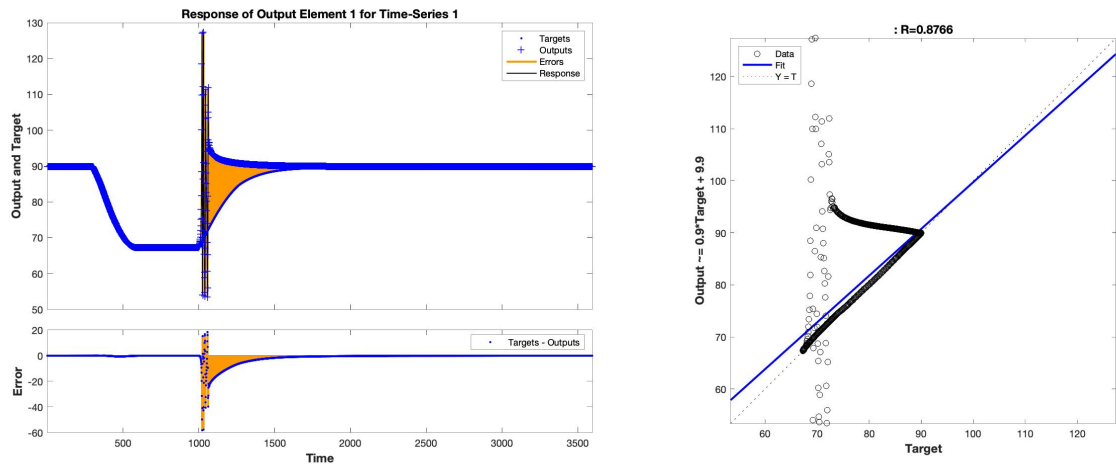


Figure 5.42 – Targets, outputs, error and response for estimation data & Target x estimation correlation (VAL02).

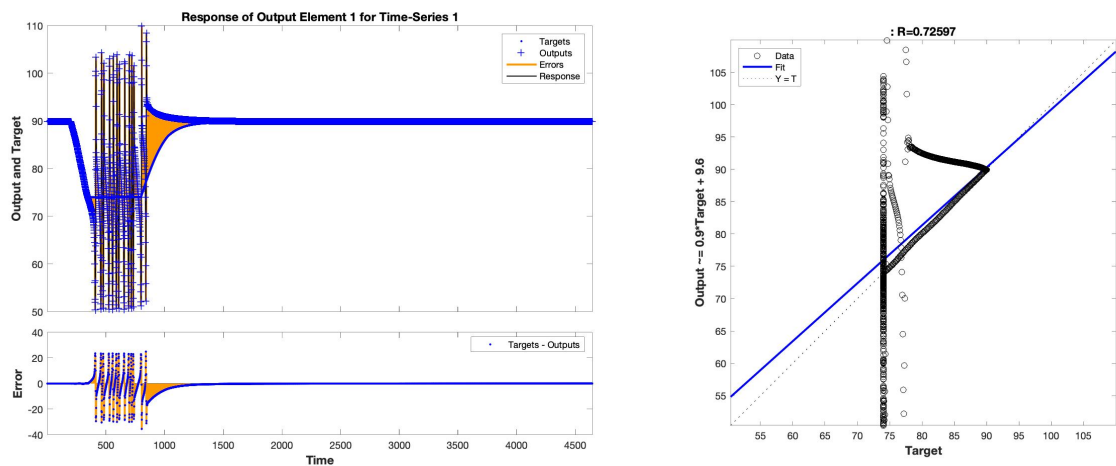


Figure 5.43 – Targets, outputs, error and response for estimation data & Target x estimation correlation (VAL03).

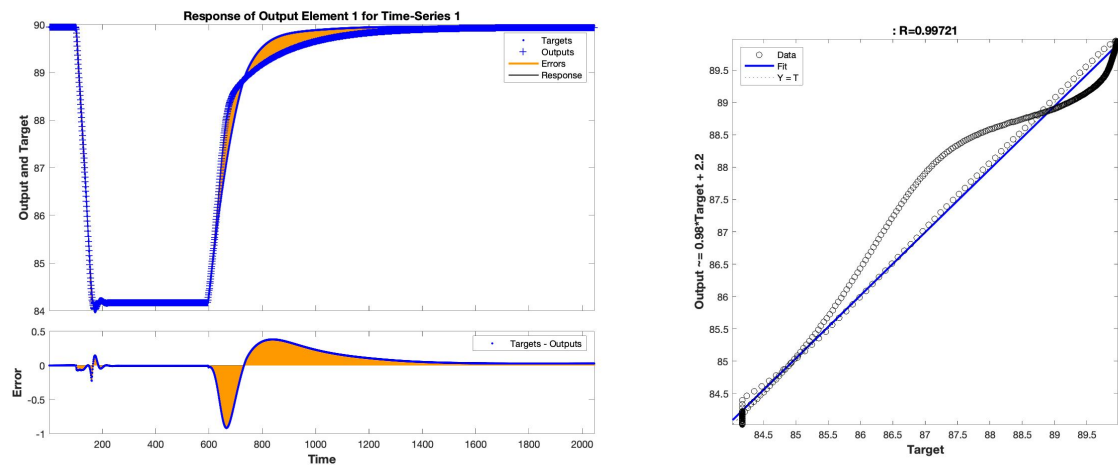


Figure 5.44 – Targets, outputs, error and response for estimation data & Target x estimation correlation (VAL04).

5.3.3 Retraining the Closed Loop Model

The retraining helped reduce the error greatly and presented a slight decrease in running time. Table 5.7 shows the comparison between the open loop, the closed loop and retrained closed loop. The model topology was not changed. Figures 5.45 to 5.49 show the model fitness to the target. The minimum service application has a substantial detachment on the brake release curve between the model and the target.

case	Error - mse (10E-3)			run time [s]		
	close loop (retrained)	close loop	open loop	close loop (retrained)	close loop	open loop
EST01	16.352	1.449E4	0.00153	0.172	0.202	0.709
VAL01	24.900	1.3392E4	0.00183	0.147	0.139	0.602
VAL02	8.008	1.912E4	0.00082	0.030	0.031	0.111
VAL03	15.716	1.959E4	0.00129	0.038	0.037	0.142
VAL04	80.478	0.0036	0.00337	0.025	0.022	0.078

Table 5.7 – Accuracy (error) and time results for the most accurate neural network (BP -> AR - Closed; retrained).

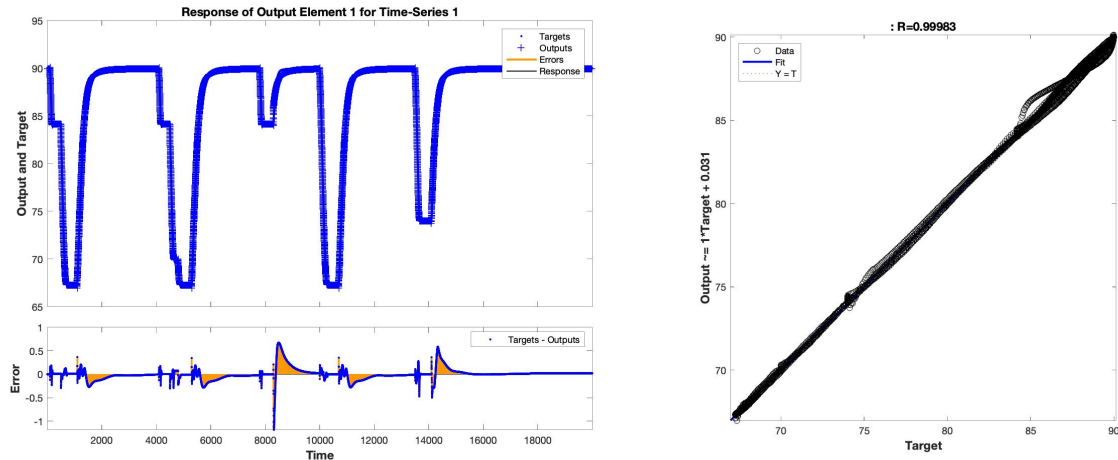


Figure 5.45 – Targets, outputs, error and response for estimation data & Target x estimation correlation (EST01).

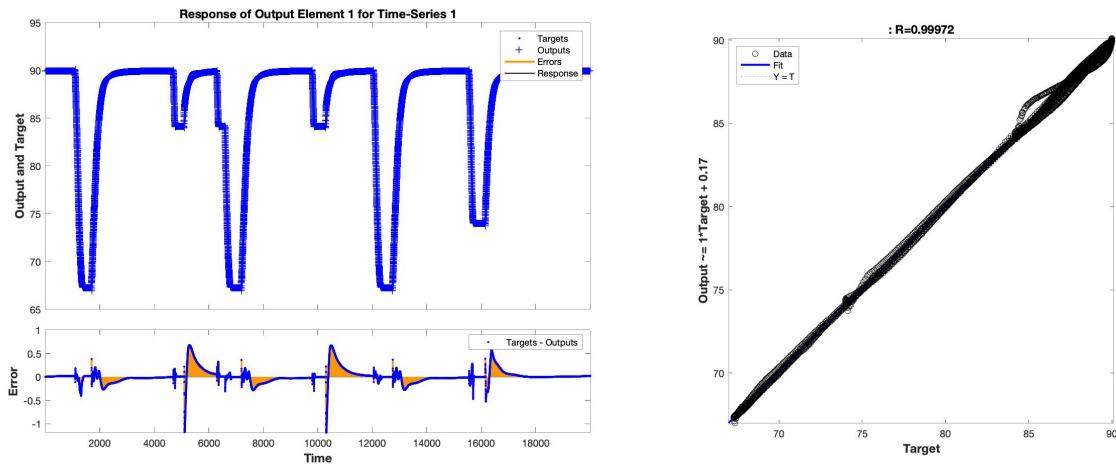


Figure 5.46 – Targets, outputs, error and response for estimation data & Target x estimation correlation (VAL01).

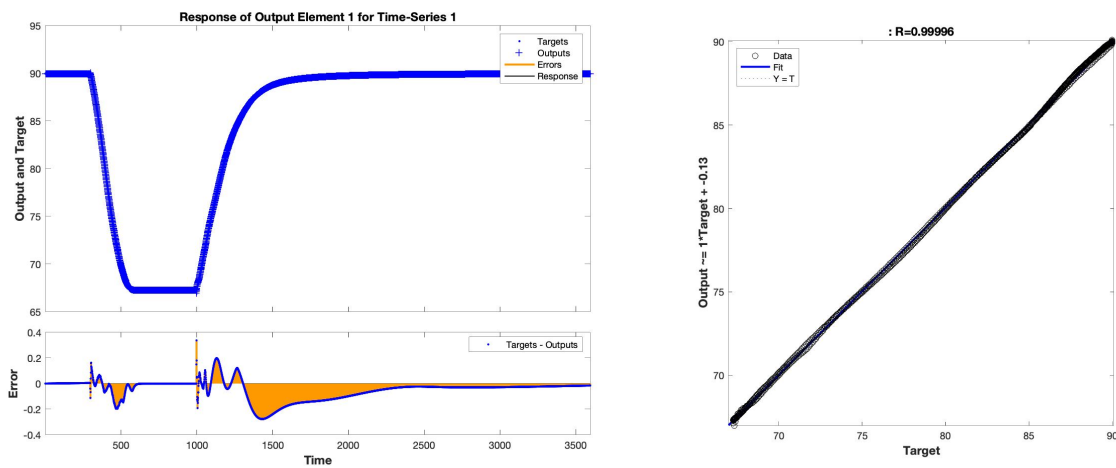


Figure 5.47 – Targets, outputs, error and response for estimation data & Target x estimation correlation (VAL02).

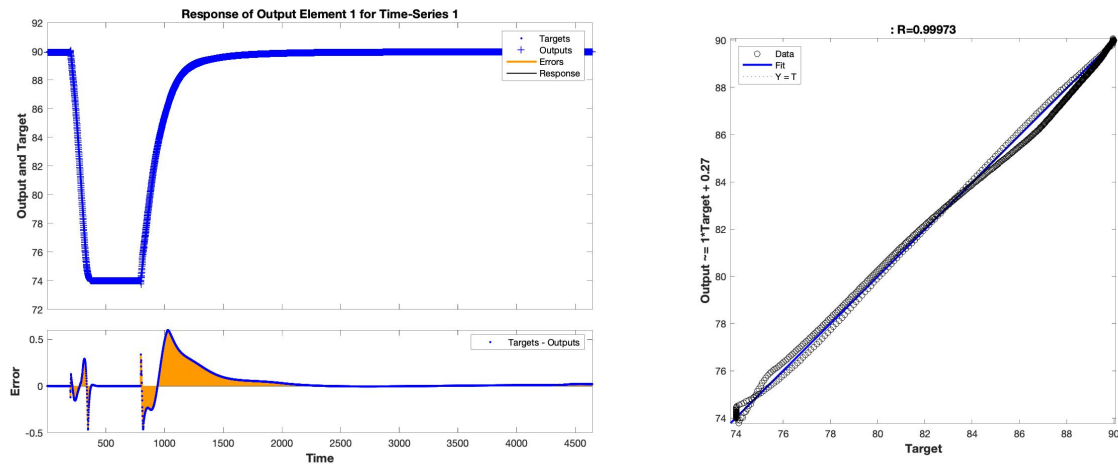


Figure 5.48 – Targets, outputs, error and response for estimation data & Target x estimation correlation (VAL03).

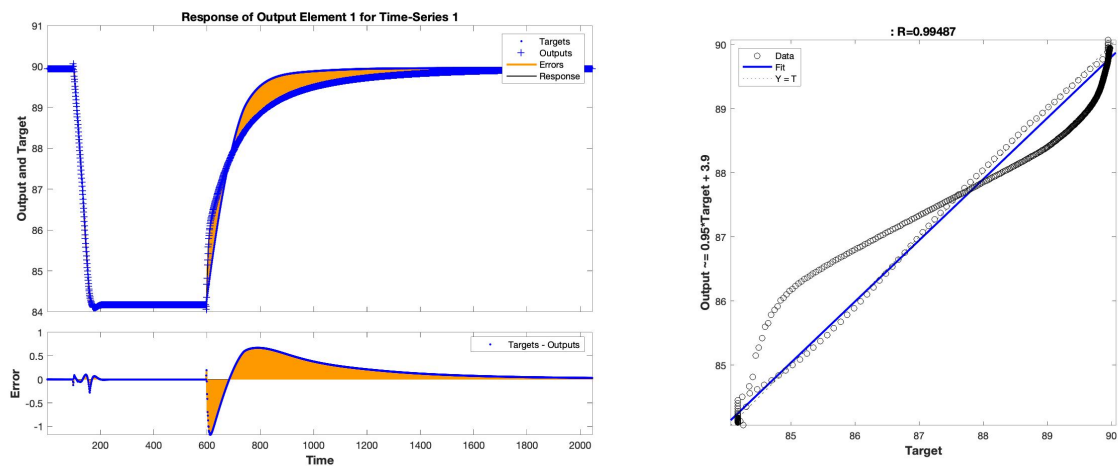


Figure 5.49 – Targets, outputs, error and response for estimation data & Target x estimation correlation (VAL04).

5.3.4 Closed Loop Retraining 2

As observed on the previous section, the VAL04 dataset had a non ideal fit. Therefore, instead of training the ANN with the EST01 dataset - that has a single minimum service brake application, an alternative training dataset was concocted by joining VAL01 and VAL04, creating a training dataset with more minimum applications. Figure 5.50 shows the new training dataset. Table 5.8 shows that though improvements were made on the error of VAL04, figure 5.55 shows a slight improvement on the VAL04 fitness.

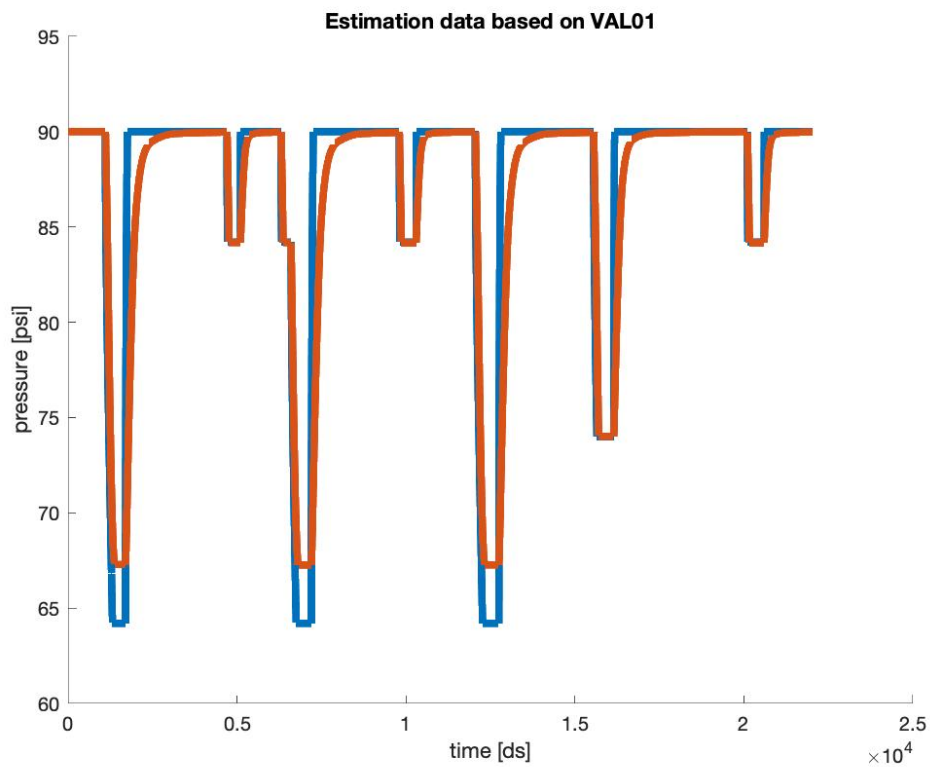


Figure 5.50 – Training dataset based on the portmanteau of VAL01 and VAL04. Blue line is the BP pressure and red line is the Aux. Reservoir.

case	Error - mse (10E-3)			run time [s]		
	close loop (retrained 2)	close loop (retrained)	open loop	close loop (retrained 2)	close loop (retrained)	open loop
EST01	23.008	16.352	0.00153	0.203	0.172	0.709
VAL01	23.310	24.900	0.00183	0.149	0.147	0.602
VAL02	14.832	8.008	0.00082	0.031	0.030	0.111
VAL03	14.357	15.716	0.00129	0.039	0.038	0.142
VAL04	57.387	80.478	0.00337	0.022	0.025	0.078

Table 5.8 – Accuracy (error) and time results for the most accurate neural network (BP -> AR - Closed; retrained 2).

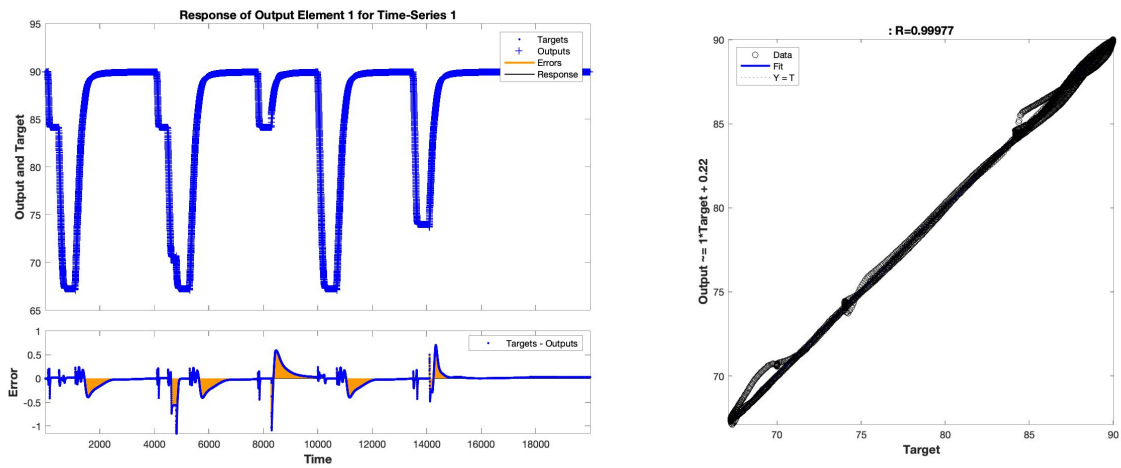


Figure 5.51 – Targets, outputs, error and response for estimation data & Target x estimation correlation (EST01).

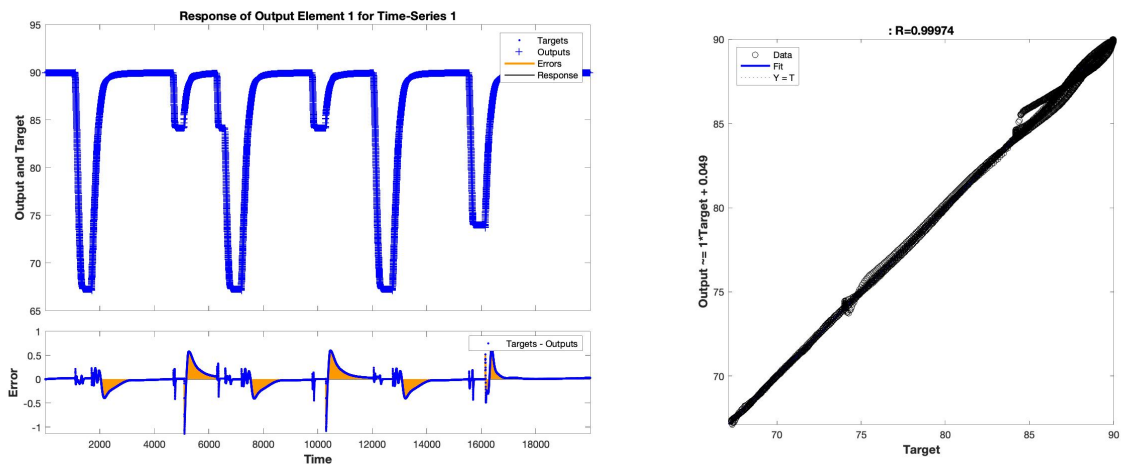


Figure 5.52 – Targets, outputs, error and response for estimation data & Target x estimation correlation (VAL01).

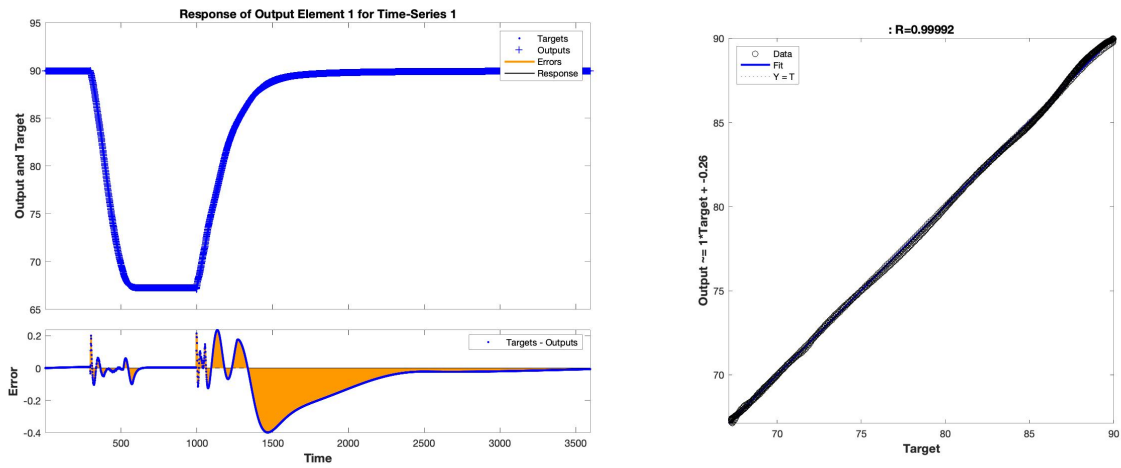


Figure 5.53 – Targets, outputs, error and response for estimation data & Target x estimation correlation (VAL02).

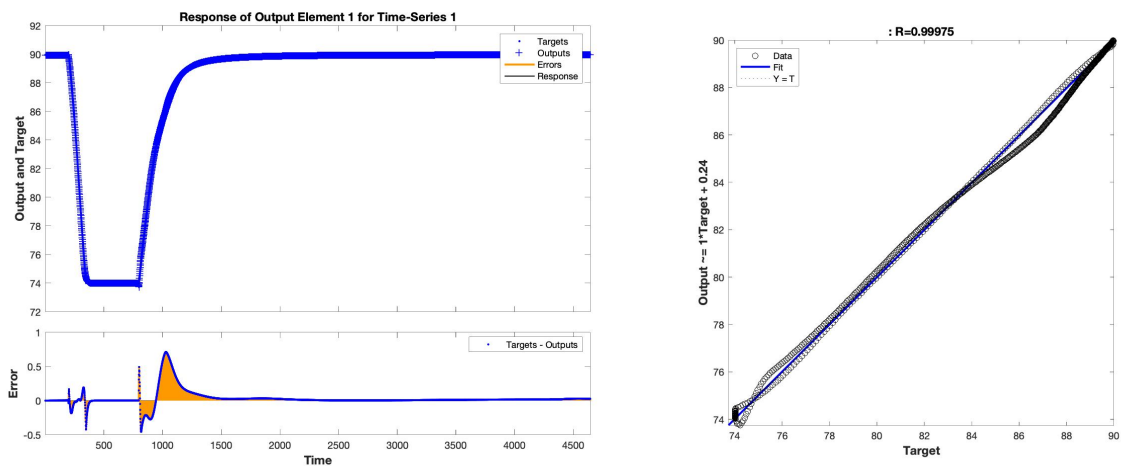


Figure 5.54 – Targets, outputs, error and response for estimation data & Target x estimation correlation (VAL03).

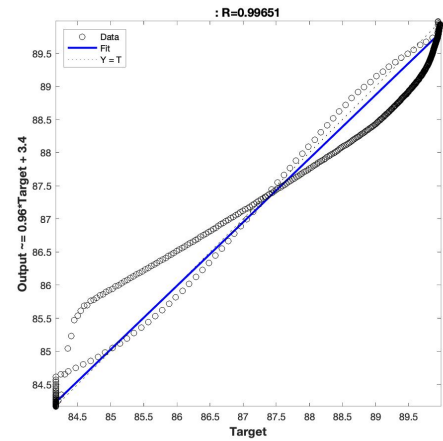
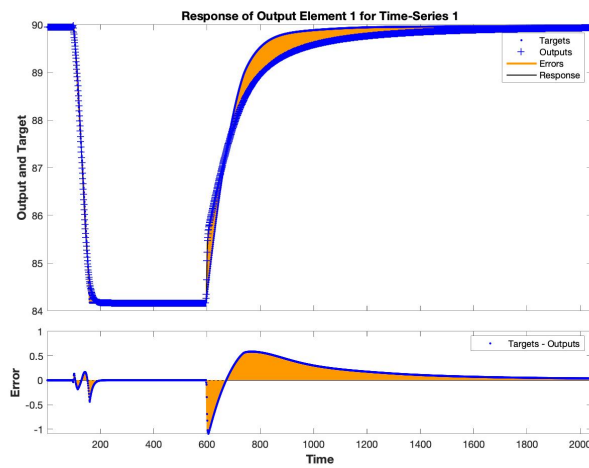


Figure 5.55 – Targets, outputs, error and response for estimation data & Target x estimation correlation (VAL04).

5.3.5 BP -> AR Closed Loop Model Exploration

Possibly, this lack of fitness is due to the absence of the emergency reservoir information. A different dataset was tried for the training but this has not yielded good results. It seems that in this model, the parameter sweep done with the open loop was not effective. It is important to note that the most accurate model was near the edges of the search space. A closed loop parameter sweep with current hardware is not practical, so an empirical search will be conducted for alternative models, changing the input and output delays. Tables 5.9 and 5.10 show some of the best networks. The run time did not change substantially, however, model D presented the most accurate responses, followed by model F. The best combination of parameters was 6 delays on the input, 10 delays on the output and 20 neurons. These fall outside the open parameter sweep. However, the closed model behaves quite differently than the open one, especially on the brake release, were the emergency reservoir would have a high influence. It is uncertain if there would be any benefit in broadening the parameter search scope. Figure 5.56 shows the model representation. Figures 5.57 to 5.61 show the model output and the target. On the minimum service application, there are some disturbance on the pressure drop. Also, on the release of the VAL03 data there is some disagreement between the model and the targets. However, it is not believed that such fitness issues will be relevant enough to cause problems to the train simulation overall.

	Error - mse (10E-3)					
model	A	B	C	D	E	F
input delays	2	4	2	6	8	5
output delays	2	9	12	10	10	12
neurons	20	20	20	20	25	20
EST01	49.86	13.30	32.21	8.20	24.50	10.92
VAL01	61.60	8.30	15.20	6.56	9.67	7.81
VAL02	35.42	4.35	7.14	2.91	4.90	4.34
VAL03	35.02	21.15	21.43	19.20	18.87	20.96
VAL04	154.09	2.53	26.53	1.71	8.47	2.50
fitness	- - -	-	- -	+	o	+

Table 5.9 – Accuracy (error) for candidate artificial neural network (BP -> AR - Closed).

model	run time [s]					
	A	B	C	D	E	F
input delays	2	4	2	6	8	5
output delays	2	9	12	10	10	12
neurons	20	20	20	20	25	20
EST01	0.158	0.155	0.163	0.157	0.182	0.157
VAL01	0.152	0.158	0.162	0.152	0.165	0.157
VAL02	0.033	0.031	0.035	0.034	0.035	0.32
VAL03	0.034	0.037	0.045	0.040	0.040	0.41
VAL04	0.024	0.021	0.023	0.022	0.024	0.23
fitness	- - -	-	- -	+	0	+

Table 5.10 – Run time for candidate artificial neural network (BP -> AR - Closed).

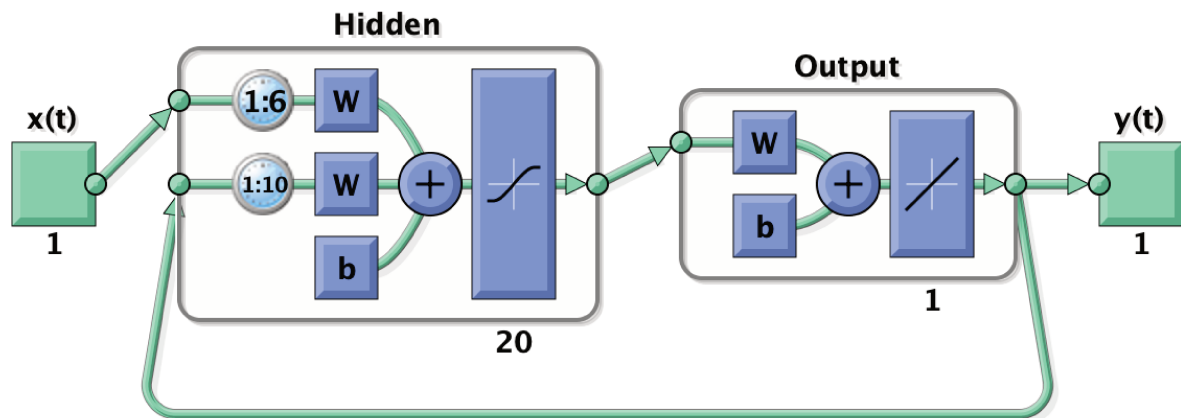


Figure 5.56 – Model representation for BP -> AR (closed loop).

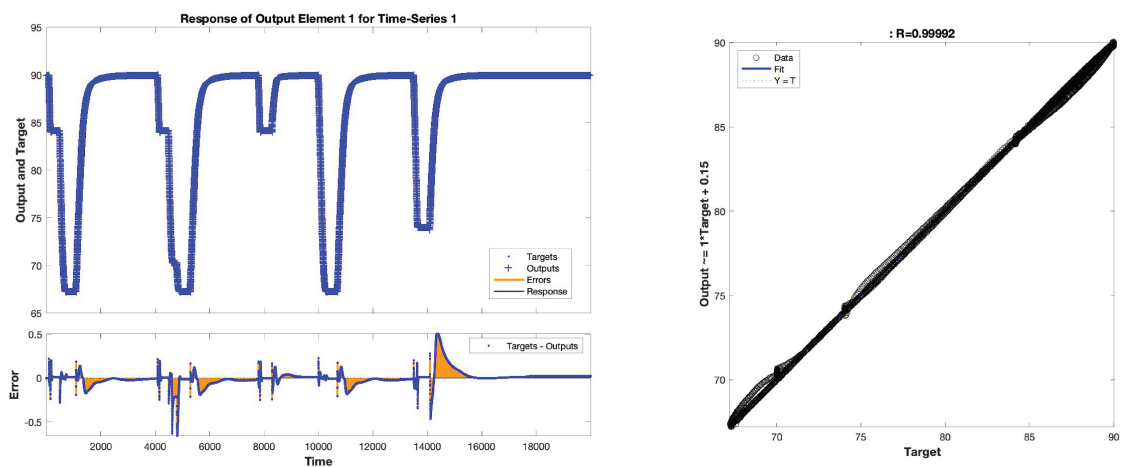


Figure 5.57 – Targets, outputs, error and response for estimation data & Target x estimation correlation (EST01).

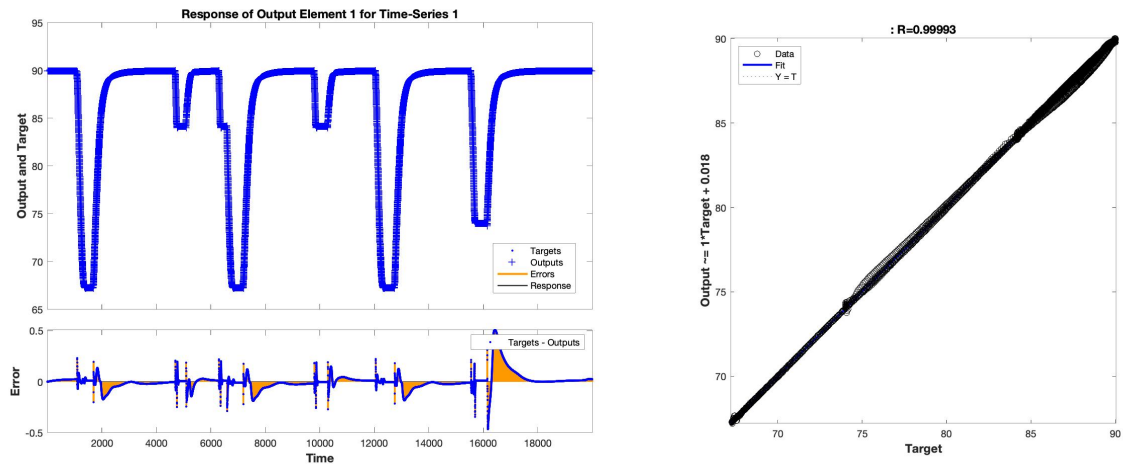


Figure 5.58 – Targets, outputs, error and response for estimation data & Target x estimation correlation (VAL01).

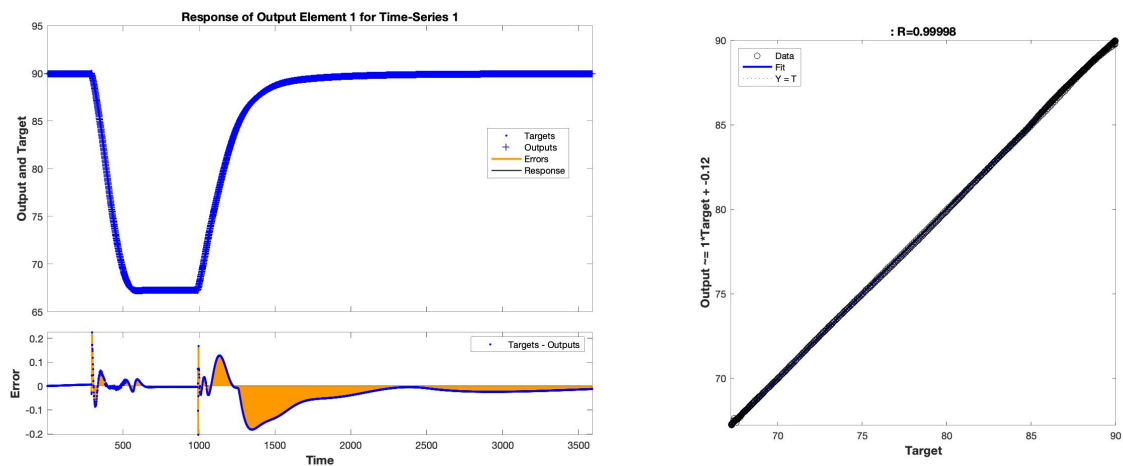


Figure 5.59 – Targets, outputs, error and response for estimation data & Target x estimation correlation (VAL02).

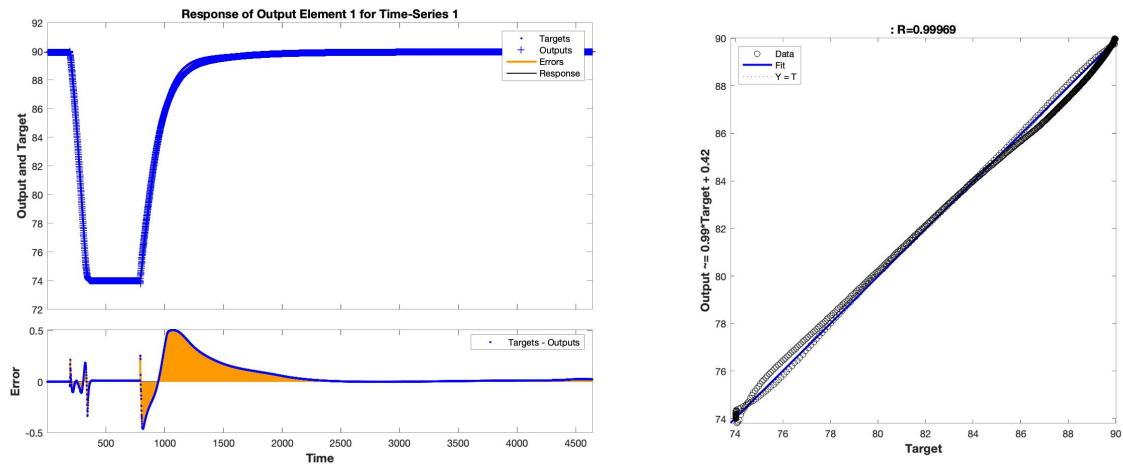


Figure 5.60 – Targets, outputs, error and response for estimation data & Target x estimation correlation (VAL03).

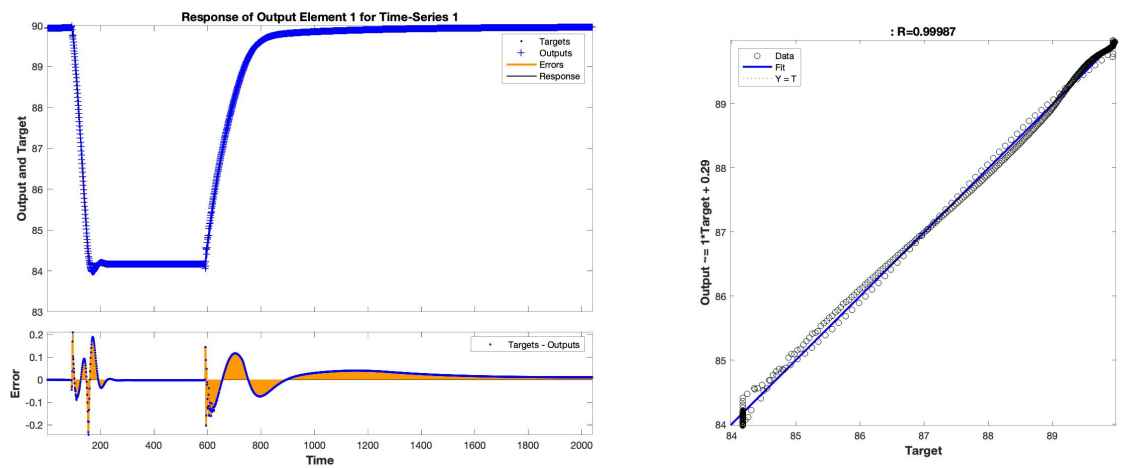


Figure 5.61 – Targets, outputs, error and response for estimation data & Target x estimation correlation (VAL04).

6 SIMULATION

Now that all the ANN models are trained and refined, one can continue on the initial path and create the model fusion to simulate the problem at hand. The topology of the problem (figure 6.1) and the strategy outlined previously (figure 6.2) are the same.

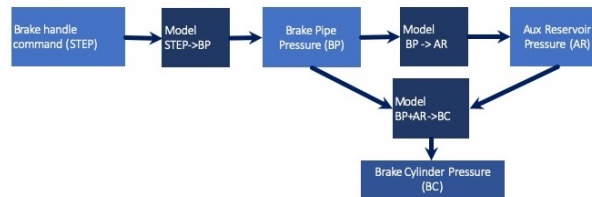


Figure 6.1 – Problem topology and required models.

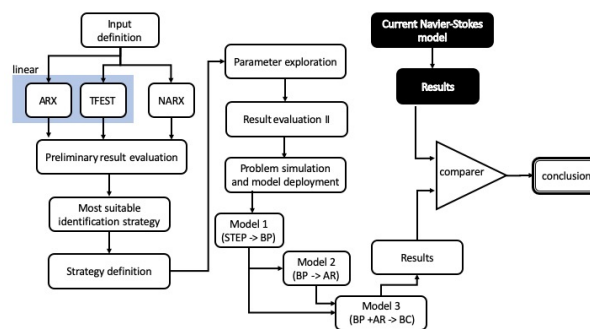


Figure 6.2 – Overall strategy to address the problem.

6.1 Model Implementation

To joint all models and take care of the required services, a MATLAB program was created. It uses the following functions and is organised per outlined on the next sections.

6.1.1 Functions

6.1.1.1 Matrix to Step

The function to transform the command matrix into a step time series is the same as used previously.

```
function output=step(input)
```

```

01  vector=input;
02  [b c]=size(vector);
03  for id=1:(b-1)
04      a1=vector(id,1)+1;
05      a2=vector(id+1,1);
06      b1=vector(id,2);
07      for ik=a1:a2
08          output(ik,1)=b1;
09      end
10  end
end

```

6.1.1.2 ANN execution

The function to run the artificial neural network is very similar to the one used previously. There is a difference that there is no target function, since the network is being run to simulate the problem. However, the target values are still necessary to construct the delays $y(0); y(-1); \dots; y(n)$, so, the input is repeated in order to obtain these values. The target values at the start are similar to the input values, so this strategy has no problem at this moment. Line 06 is where the ANN is run. On line 08 the output is converted from a cell format to a more usable vector format. The time counting parameters were removed for the time will be assessed on the program as a whole.

```

function [yout,tn]=RunNN(in0,in1)
01  net=in0;
02  X=tonndata(in1,false,false);
03  T=tonndata(in1,false,false);

04  [x,xi,ai,t]=preparets(net,X,{},T);
06  y = net(x,xi,ai);

07  yout=y;
08  yout=cell2mat(yout);

```

```
09  tn=NaN
end
```

The function used to run the CF model is very similar with a crucial difference - the addition of line 03. In this case, the expected output is the brake cylinder, and its pressure is supposed to be 0 at the start of the simulation. If the initial pressure were to be considered the same as the input (as was done on the previous function) there would be an error.

```
function [yout,tn]=RunNN2(in0,in1)
01  net=in0;
02  [a b]=size(in1);
03  in1_1=zeros(a,1);
04  X=tonndata(in1,false,false);
05  T=tonndata(in1_1,false,false);

06  [x,xi,ai,t]=preparets(net,X,{},T);
07  y = net(x,xi,ai);

08  yout=y;
09  yout=cell2mat(yout);
10  tn=NaN;
end
```

6.1.1.3 Performance Evaluation

To maintain consistency with the performance evaluations performed on the other stages of this work, the same method was used. Previously, the performance evaluation was built into the ANN execution function, however, for the simulation implementation, there is no reason for this to be so. Moreover, the performance calculation takes a bit of time and on the simulation, there is no target to check the performance against, at least in theory. The final deployment of the simulation will suppress this specific function.

```
function [performance,e,f]=Perf(net,target,output,f)
01  XR=tonndata(output,false,false);
02  TR=tonndata(target,false,false);
```

```

03  [xr,xir,air,tr]=preparets(net,XR,{},TR);
04  figure(f), plotregression(tr,xr),f=f+1;
05  figure(f), plotresponse(tr,xr),f=f+1;

06  performance = perform(net,tr,xr);

07  e=gsubtract(target,output);
08  figure(f);plot(e,'LineWidth',3);
09  title('error between target and output')
10  xlabel('time [ds]')
11  ylabel('pressure [psi]')
12  f=f+1;
end

```

6.1.2 Main Program

The main program uses the previously trained ANNs and the functions presented. It consists of an input in form of matrix commands, similar to the current program by Teodoro ([Teodoro, 2016](#)), in order to allow a seamless integration with the current longitudinal dynamics simulator. The output is a vector with the brake cylinder pressures, that can be used on the simulator as well. When the ANN is run, the delays on the inputs "consume" part of the time series. In order to align with the other inputs, the input time series has to be reconstituted (for example, lines 13 .. 28).

```

01  Clear all
02  load('trained_models') %file with the trained neural networks
03  load('Targets')%target file for performance evaluation

04  target_EG=ABDX_VAL01_EG_01;
05  target_RA=ABDX_VAL01_RA_01;
06  target_CF=ABDX_VAL01_CF_1;

07  input=VALXX; %command matrix

```

```

        %the neural networks
08  net1=net_STP_EG;
09  net2=net_EG_RA;
10  net3=net_EG_RA_CF;

11  tic %starting timer
        %STP
12  ABDX_STP=step(input);
        %STP -> BP
13  [ABDX_EG,t_STP_EG]=RunNN(net1,ABDX_STP);
14  ABDX_EG=ABDX_EG.';
        %the model "consumes" the delays - hence we have
        %to reconstitute the time series
15  clear a b g h a1 b1 u
16  a1=size(ABDX_EG);
17  b1=size(ABDX_STP);
18  a=a1(1,1); b=b1(1,1);
19  g=b-a;
20  ABDX_EG2=ABDX_EG;
21  ABDX_EG = lagmatrix(ABDX_EG, [g]);
22  for h=1:g
23      ABDX_EG(h)=ABDX_EG2(h);
24  end
25  u=a-g;
26  for h=1:g
27      ABDX_EG(a+h)=ABDX_EG2(u+h);
28  end
29  output_EG=ABDX_EG;
        %BP -> AR
30  [ABDX_RA,t_EG_RA]=RunNN(net2,ABDX_EG);

```

```

31 ABDX_RA=ABDX_RA.';
32 clear a b g h a1 b1 u
33 a1=size(ABDX_RA);
34 b1=size(ABDX_EG);
35 a=a1(1,1); b=b1(1,1);
36 g=b-a;
37 ABDX_RA2=ABDX_RA;
38 ABDX_RA = lagmatrix(ABDX_RA, [g]);
39 for h=1:g
40     ABDX_RA(h)=ABDX_RA2(h);
41 end
42 u=a-g;
43 for h=1:g
44     ABDX_RA(a+h)=ABDX_RA2(u+h);
45 end
46 output_RA=ABDX_RA;
    % BP + AR -> BC
47 ABDX_EG_RA(:,1)=ABDX_EG(:);
48 ABDX_EG_RA(:,2)=ABDX_RA(:);

49 [ABDX_CF,t_CF]=RunNN2(net3,ABDX_EG_RA);

50 ABDX_CF=ABDX_CF.';
51 clear a b g h a1 b1 u
52 a1=size(ABDX_CF);
53 b1=size(target_CF);
54 a=a1(1,1); b=b1(1,1);
55 g=b-a;
56 ABDX_CF2=ABDX_CF;
57 ABDX_CF = lagmatrix(ABDX_CF, [g]);
58 for h=1:g
59     ABDX_CF(h)=ABDX_CF2(h);

```



```

60 end
61 u=a-g;
62 for h=1:g
63     ABDX_CF(a+h)=ABDX_CF2(u+h);
64 end
65 output_CF=ABDX_CF;

66 TotalTime=toc
    %From now on images and performance calculations
67 figure(f)
68 hold on
69 plot(ABDX_STP,'b','LineWidth',2);
70 plot(ABDX_EG,'g','LineWidth',2);
71 plot(ABDX_RA,'r','LineWidth',2);
72 plot(ABDX_CF,'m','LineWidth',2);
73 title('ANN model output')
74 xlabel('time [ds]')
75 ylabel('pressure [psi]')
76 legend('STEP', 'BP','AR', 'BC')
77 hold off; f=f+1;
78 figure(f); plot(ABDX_CF,'k','LineWidth',2); f=f+1;

79 [performance_EG,e,f]=Perf(net1,target_EG,output_EG,f);
81 [performance_RA,e,f]=Perf(net2,target_RA,output_RA,f);
82 [performance_CF,e,f]=Perf(net3,target_CF,output_CF,f);

```

6.2 Simulation Results

Using the same datasets as previous - EST01, VAL01 to VAL04 - the performance evaluation was run for the simulation program. The results were not as expected as it can be seen on table 6.1. The values for the EST01, VAL01 and VAL04 are not as intended with errors as high as 14.6 (mse). Errors on the order of 10E-3 were desired. Apparently, there seems to be a problem with model BP-> AR. Figures 6.3 to 6.26 show the various graphs generated by

the program. From the graphs, it is possible to assess that the model for BP -> AR, specially on the minimum service application is not working as it should. Possibly, the model is not stable enough and, although having presented good results on its own, the noise generated by the STP -> BP model was sufficient to make it not behave adequately.

		Error - mse (10E-3)	Run time [s]
EST01	BP	6.500	1.354
	AR	1316.5	
	BC	2557.6	
VAL01	BP	8.571	0.825
	AR	2200.1	
	BC	3877.7	
VAL02	BP	4.996	1.050
	AR	5.456	
	BC	19.252	
VAL03	BP	6.412	0.739
	AR	20.655	
	BC	125.609	
VAL04	BP	8.424	0.612
	AR	10079.5	
	BC	14655.7	

Table 6.1 – Simulation results - performance & run time for simulation program.

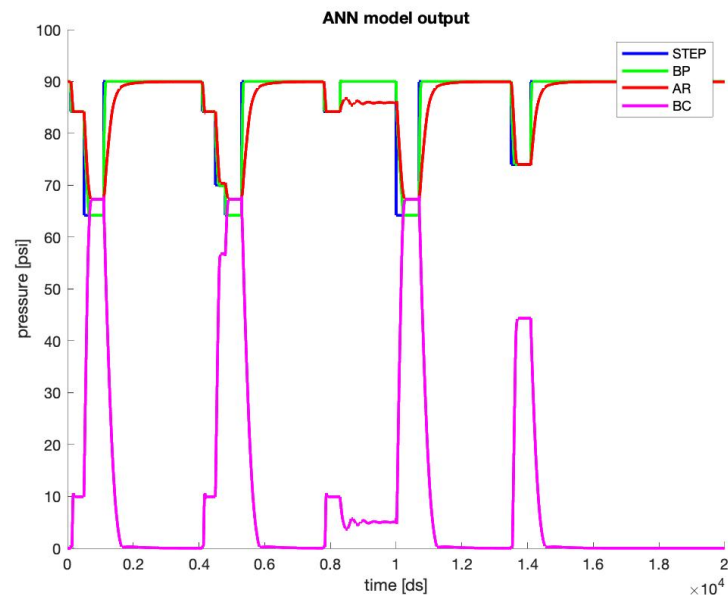


Figure 6.3 – Overall simulation outputs for EST01 - STP, BP, AR, BC

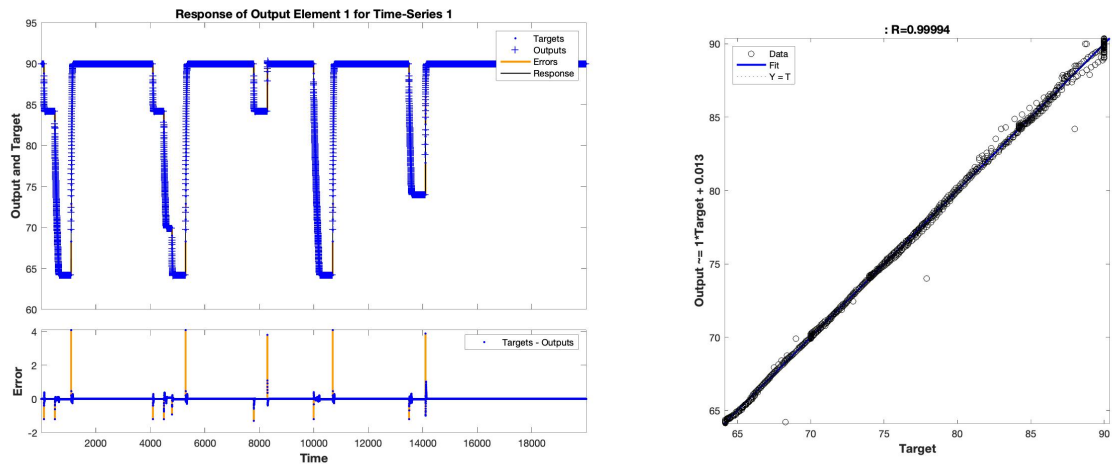


Figure 6.4 – Targets, outputs, error and response for estimation data & Target x estimation correlation (BP - EST01).

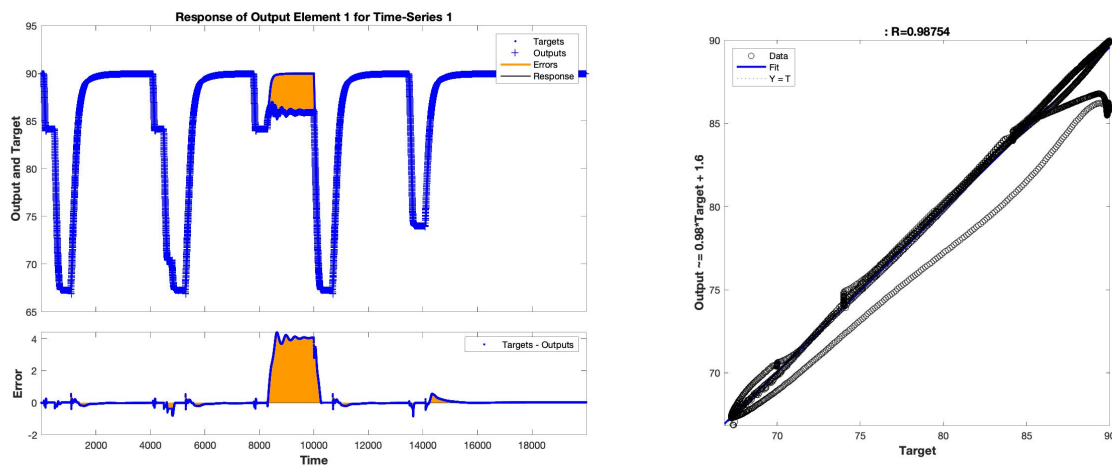


Figure 6.5 – Targets, outputs, error and response for estimation data & Target x estimation correlation (AR - EST01).

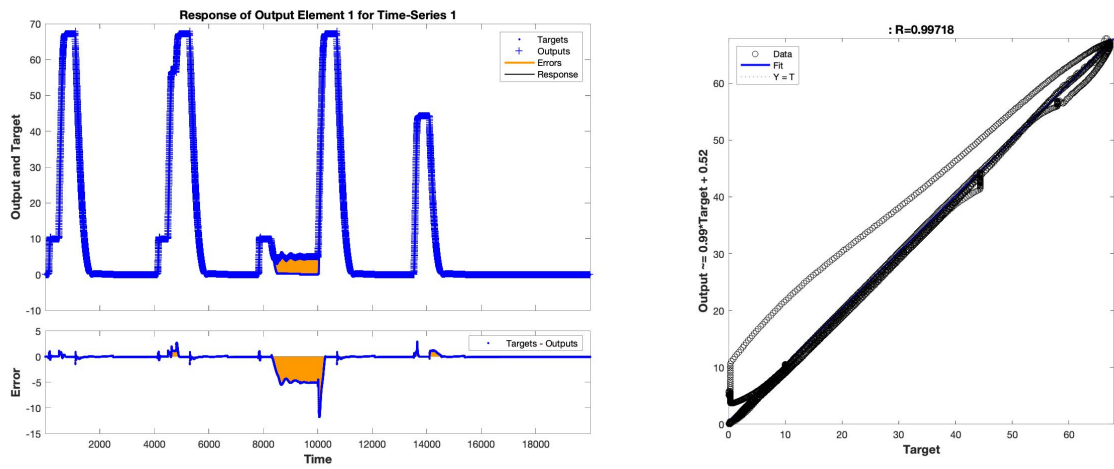


Figure 6.6 – Targets, outputs, error and response for estimation data & Target x estimation correlation (BC - EST01).

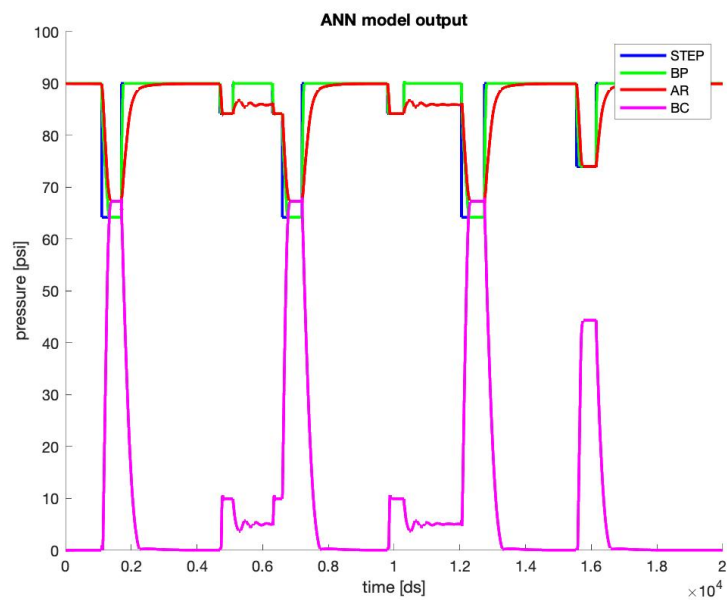


Figure 6.7 – Overall simulation outputs for VAL01 - STP, BP, AR, BC.

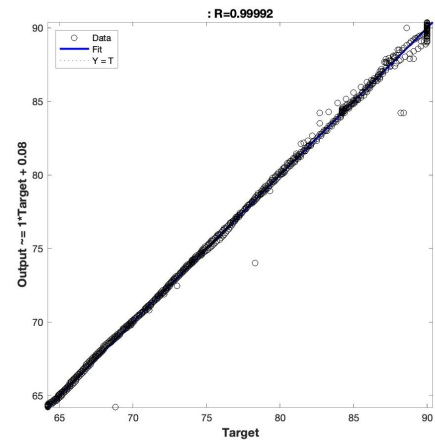
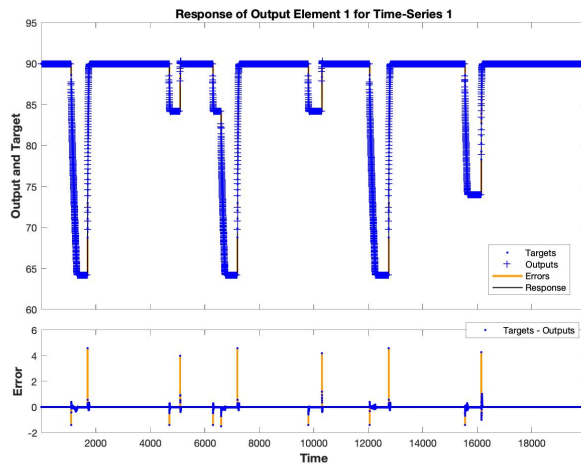


Figure 6.8 – Targets, outputs, error and response for estimation data & Target x estimation correlation (BP - VAL01).

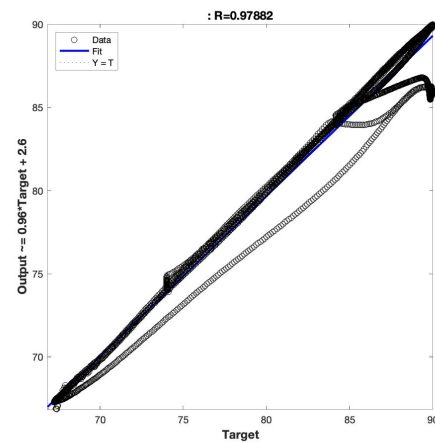
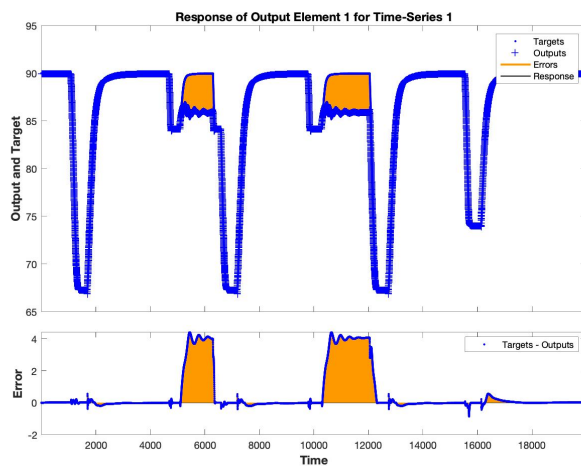


Figure 6.9 – Targets, outputs, error and response for estimation data & Target x estimation correlation (AR - VAL01).

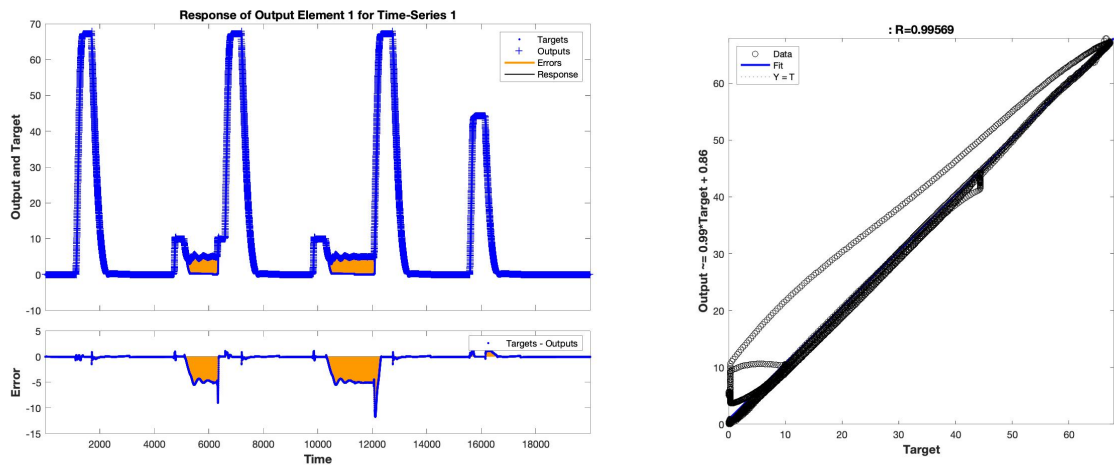


Figure 6.10 – Targets, outputs, error and response for estimation data & Target x estimation correlation (BC - VAL01).

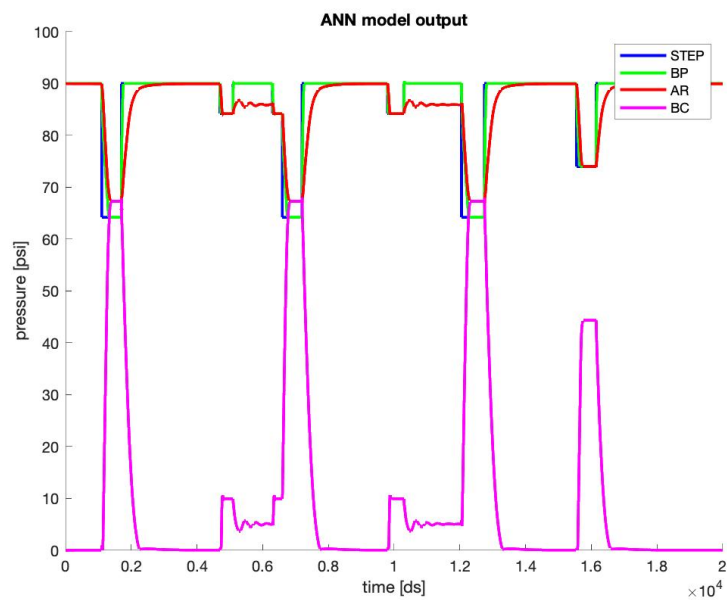


Figure 6.11 – Overall simulation outputs for VAL01 - STP, BP, AR, BC.

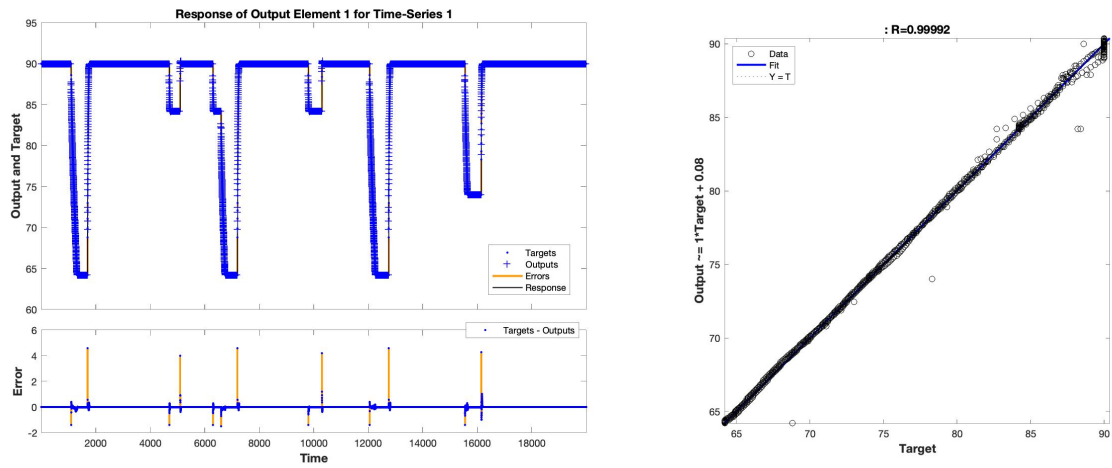


Figure 6.12 – Targets, outputs, error and response for estimation data & Target x estimation correlation (BP - VAL01).

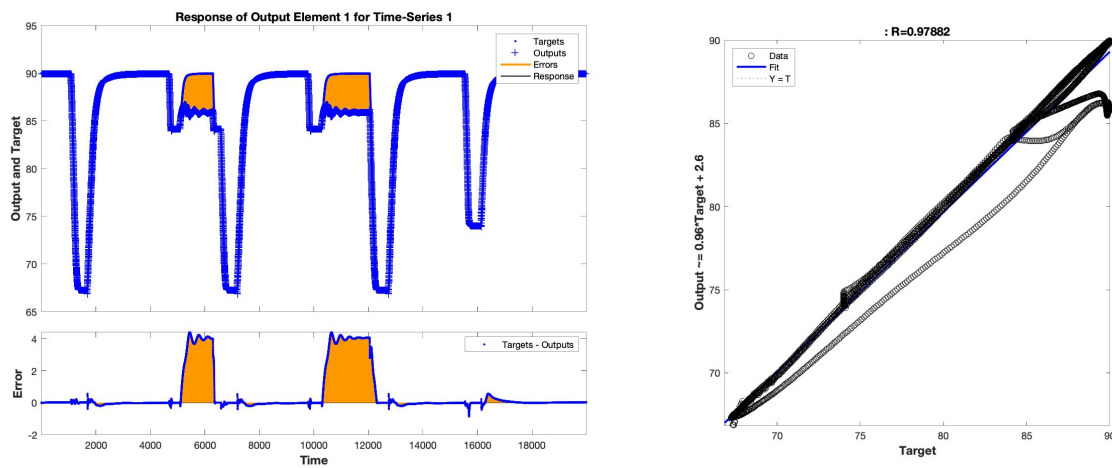


Figure 6.13 – Targets, outputs, error and response for estimation data & Target x estimation correlation (AR - VAL01).

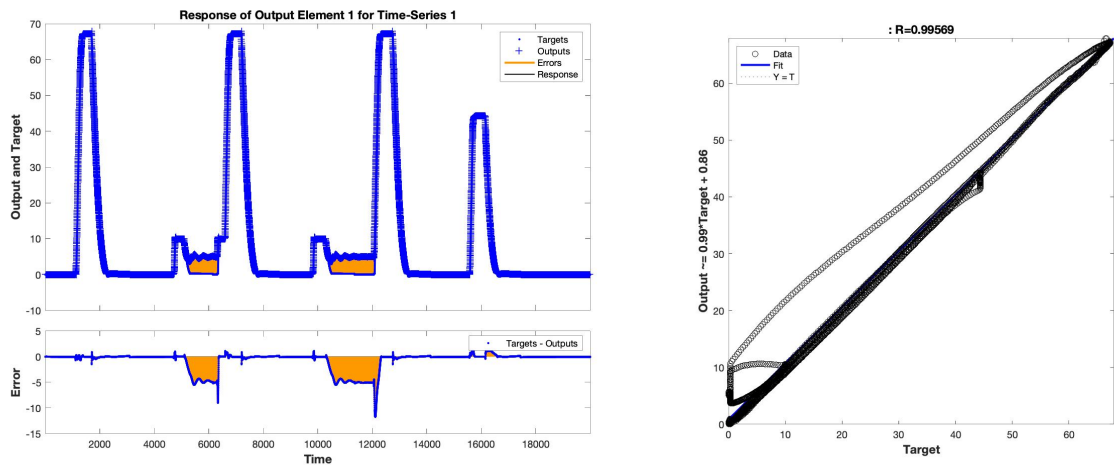


Figure 6.14 – Targets, outputs, error and response for estimation data & Target x estimation correlation (BC - VAL01).

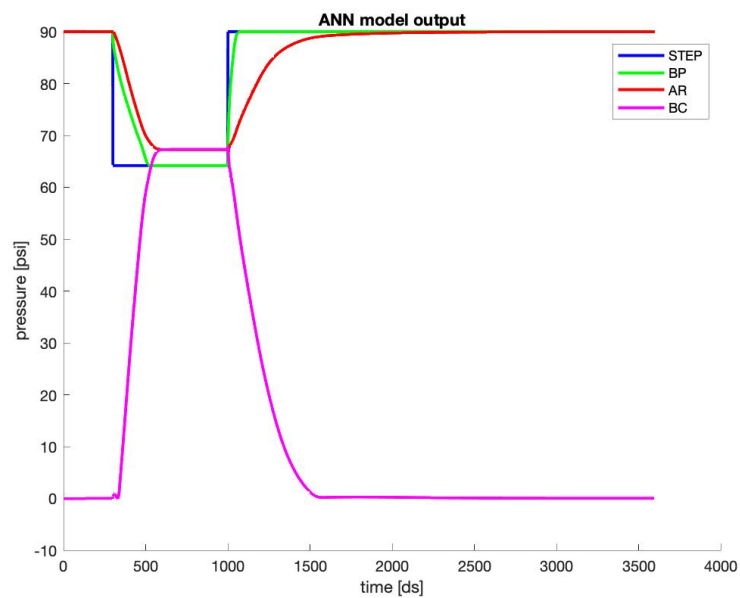


Figure 6.15 – Overall simulation outputs for VAL02 - STP, BP, AR, BC.

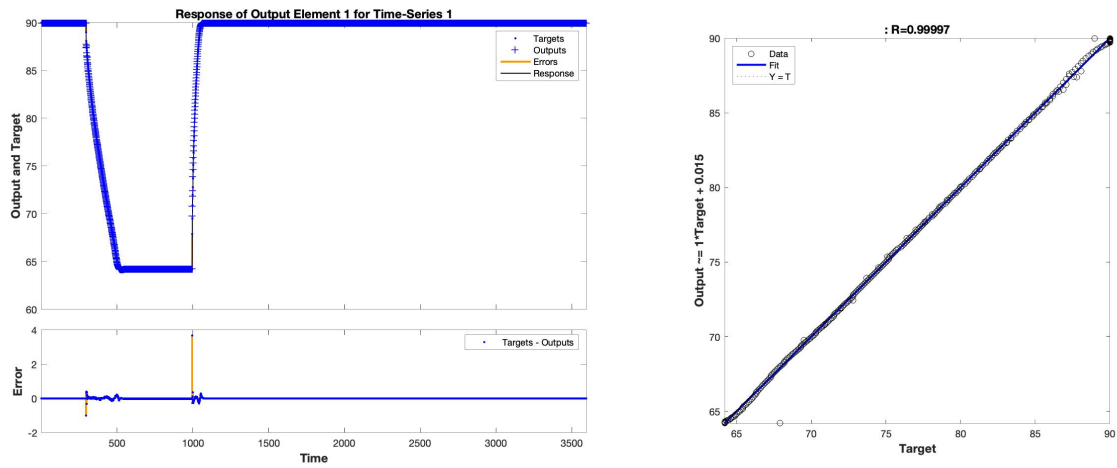


Figure 6.16 – Targets, outputs, error and response for estimation data & Target x estimation correlation (BP - VAL02).

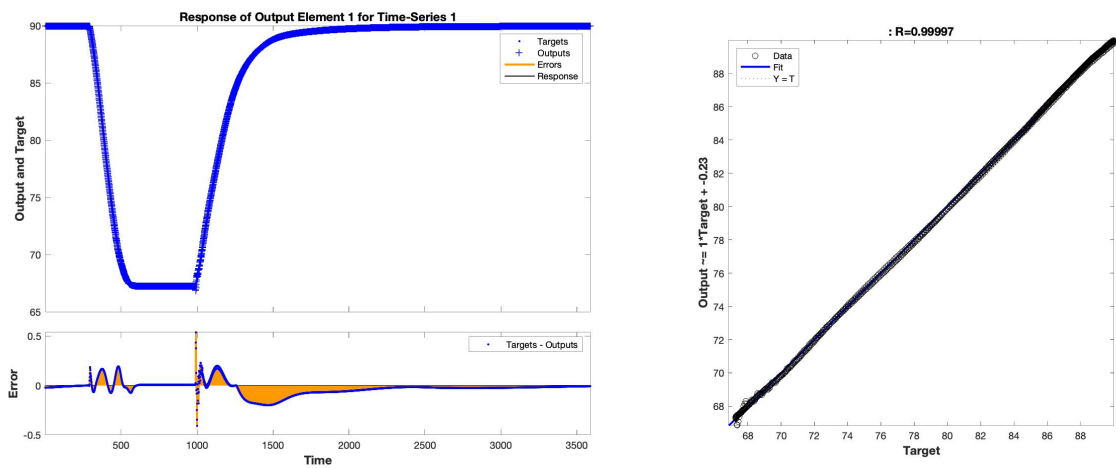


Figure 6.17 – Targets, outputs, error and response for estimation data & Target x estimation correlation (AR - VAL02).

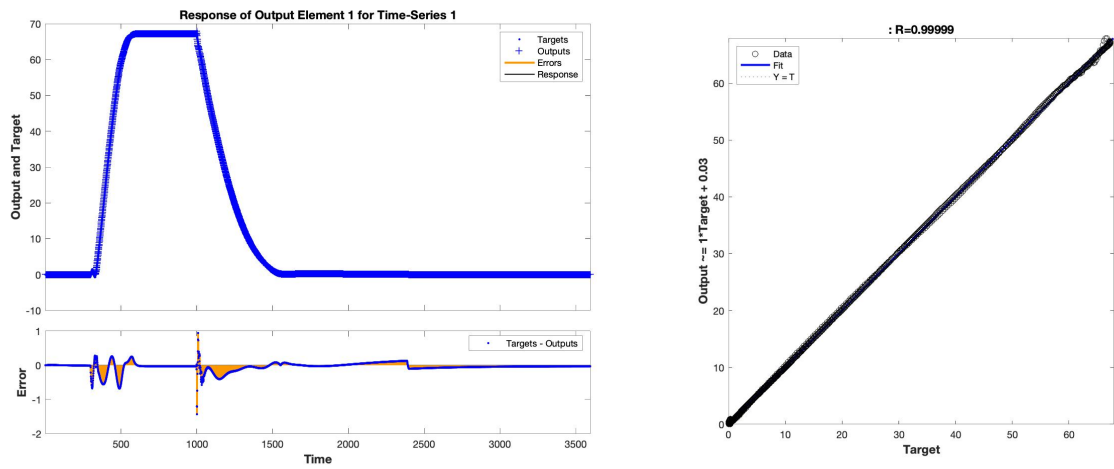


Figure 6.18 – Targets, outputs, error and response for estimation data & Target x estimation correlation (BC - VAL02).

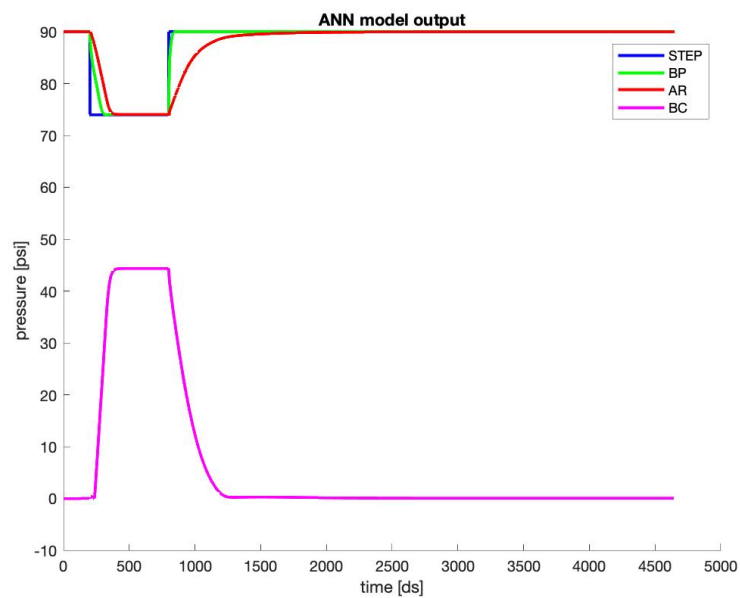


Figure 6.19 – Overall simulation outputs for VAL03 - STP, BP, AR, BC

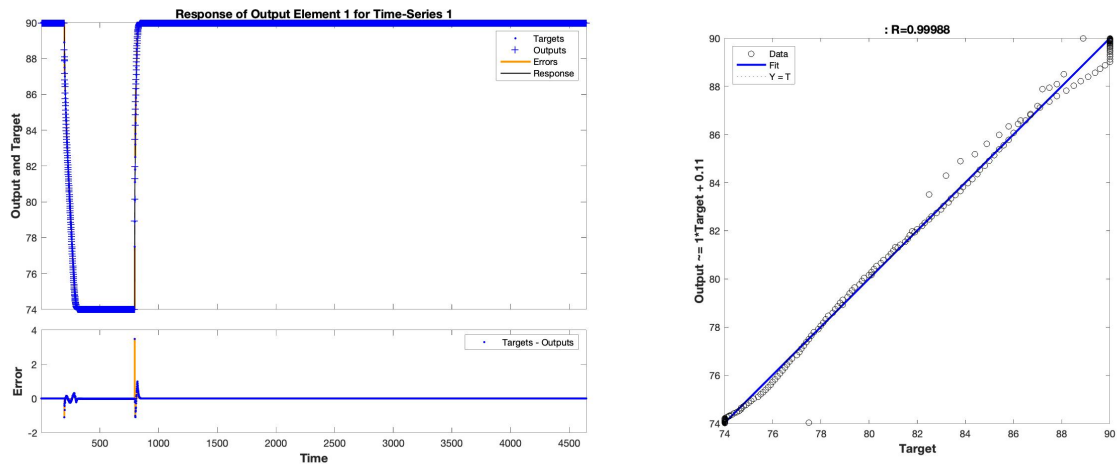


Figure 6.20 – Targets, outputs, error and response for estimation data & Target x estimation correlation (BP - VAL03).

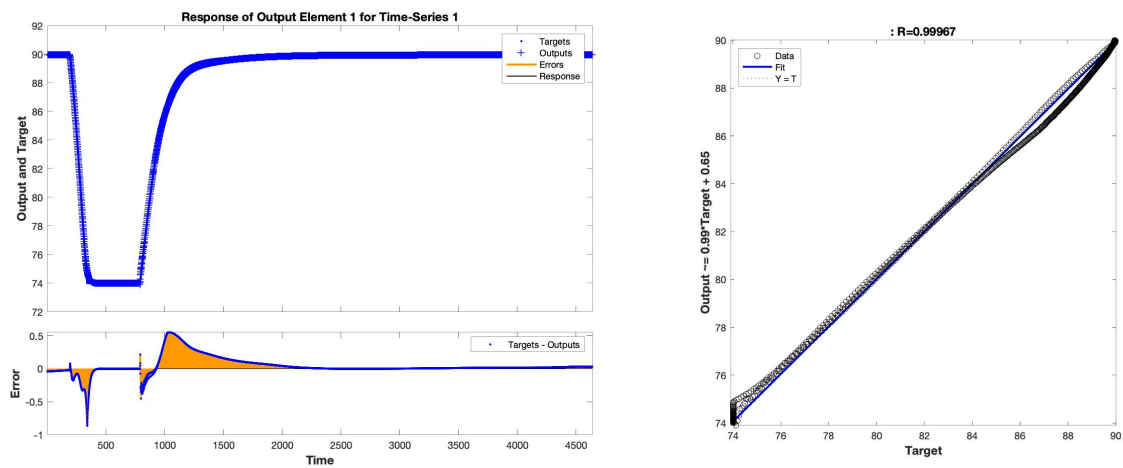


Figure 6.21 – Targets, outputs, error and response for estimation data & Target x estimation correlation (AR - VAL03).

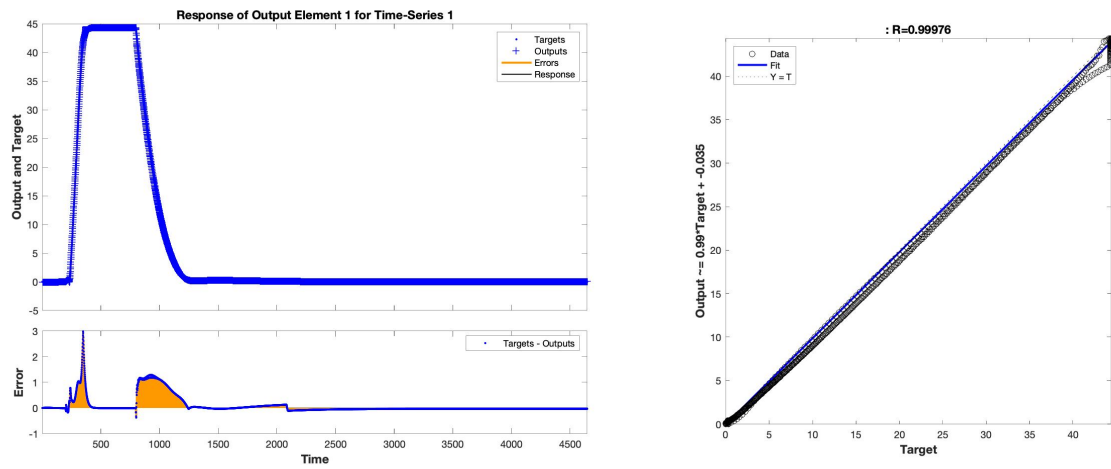


Figure 6.22 – Targets, outputs, error and response for estimation data & Target x estimation correlation (BC - VAL03).

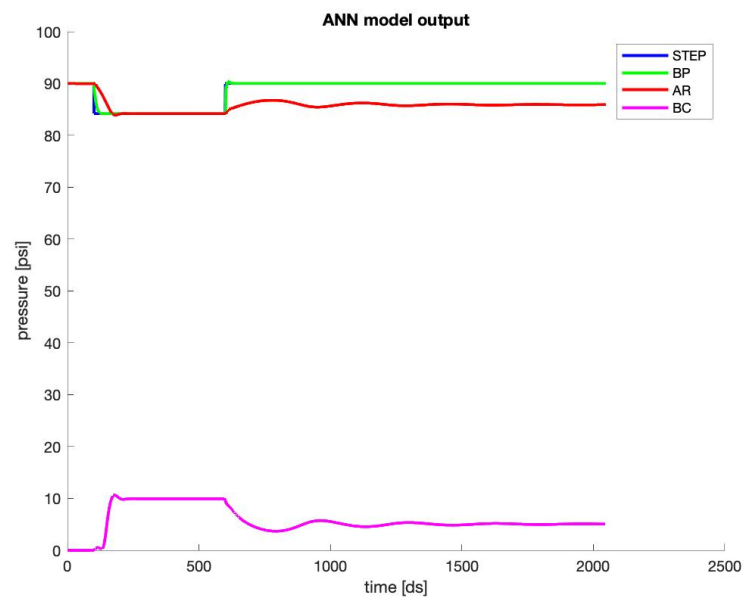


Figure 6.23 – Overall simulation outputs for VAL04 - STP, BP, AR, BC.

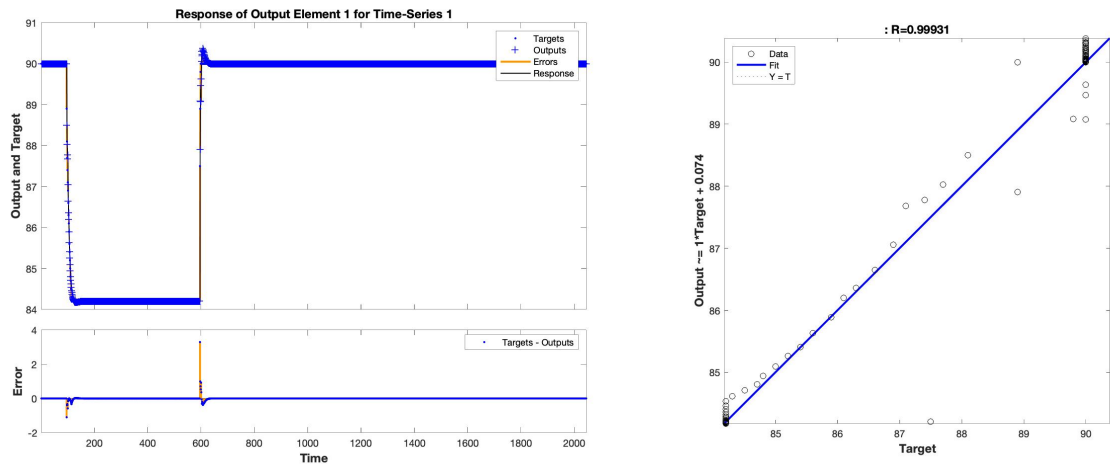


Figure 6.24 – Targets, outputs, error and response for estimation data & Target x estimation correlation (BP - VAL04).

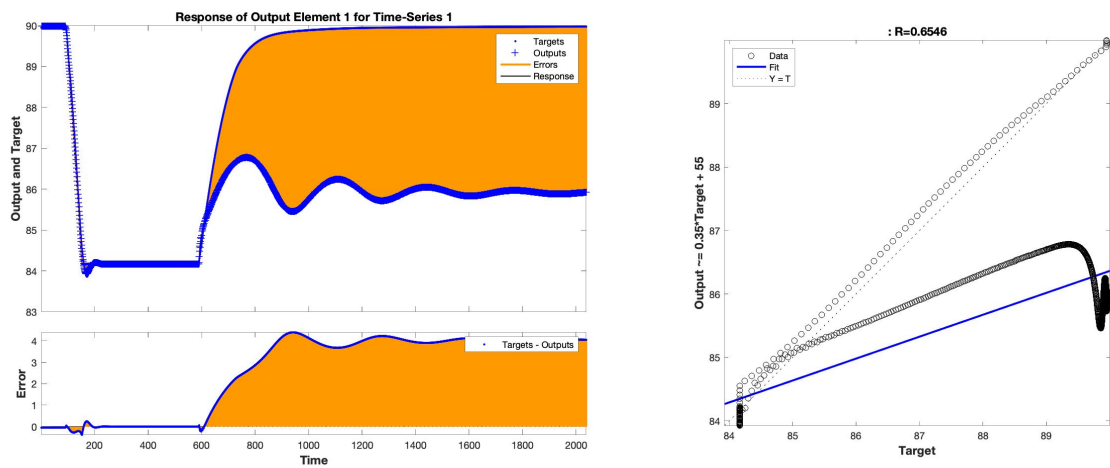


Figure 6.25 – Targets, outputs, error and response for estimation data & Target x estimation correlation (AR - VAL04).

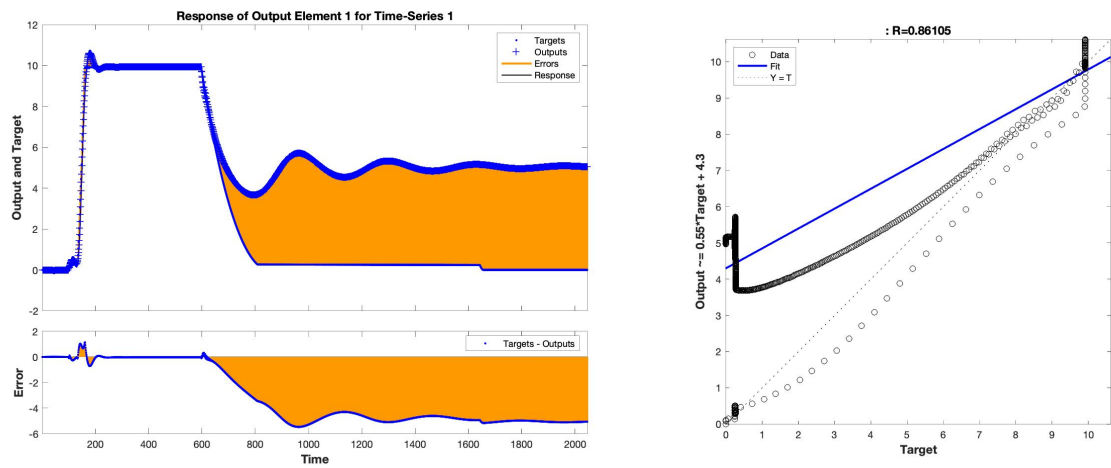


Figure 6.26 – Targets, outputs, error and response for estimation data & Target x estimation correlation (BC - VAL04).

6.2.1 Revisiting STP-BP

Given the inadequate performance with the proposed model, instead of fixing the BP $\rightarrow$ AR model, a new, more stable STP $\rightarrow$ BP model will be verified. During the training process, another model was promising it has 4 delays on the input, 4 delays on the output and 25 neurons - again, more than the initial search scope. Figure 6.27 shows the representation of this new model. The performance is only slight better on most validations (it is actually worse on VAL04). However, it is hoped that the more stable output will prevent undesired behaviours as observed previously. Figures 6.28 to 6.32 show the output and correlation for the new model. The autocorrelation values for the new model are narrower than the old model (now shown)

Case	Error - mse (10E-3)		run time [s]	
	new model	old model	new model	old model
EST01	5.213	6.500	0.1745	0.1560
VAL01	7.734	8.571	0.2983	0.1500
VAL02	4.336	4.996	0.0386	0.0326
VAL03	3.195	6.412	0.0338	0.0410
VAL04	10.046	8.423	0.0198	0.0209

Table 6.2 – Comparison between the new STP $\rightarrow$ BP model and the old.

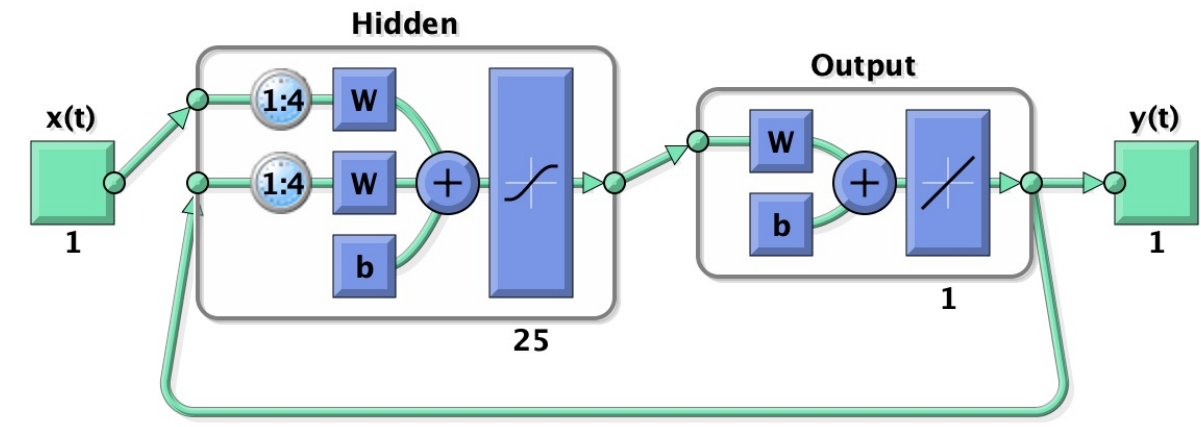


Figure 6.27 – Model representation for the new version of STP $\rightarrow$ BP.

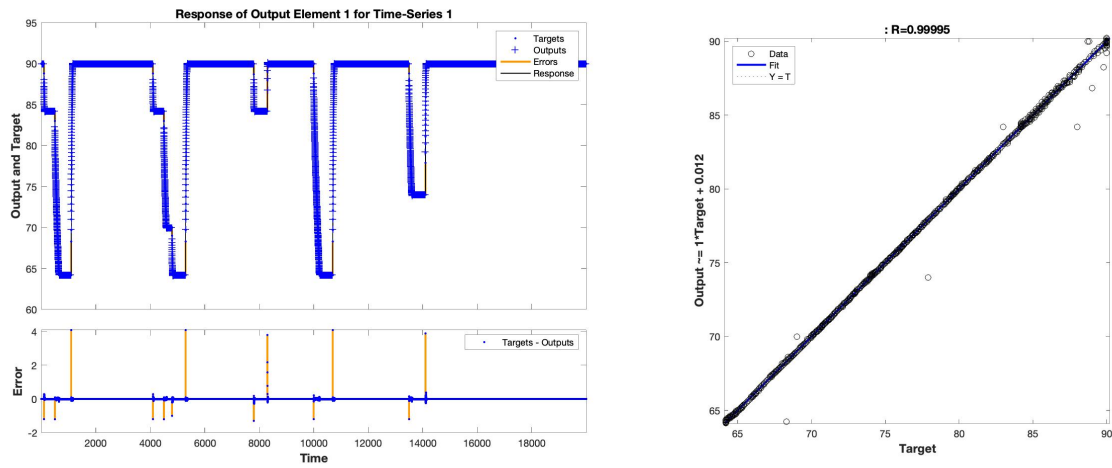


Figure 6.28 – Targets, outputs, error and response for estimation data & Target x estimation correlation (EST01).

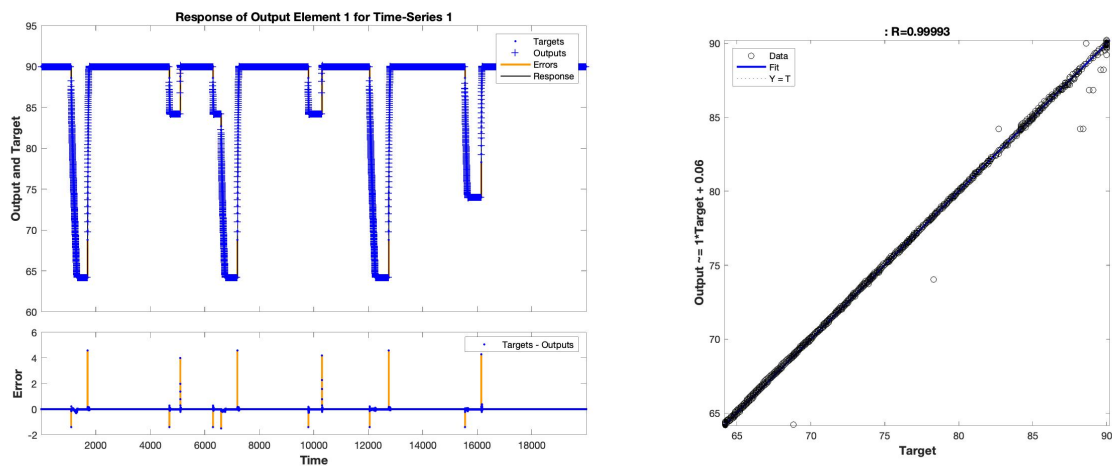


Figure 6.29 – Targets, outputs, error and response for estimation data & Target x estimation correlation (VAL01).

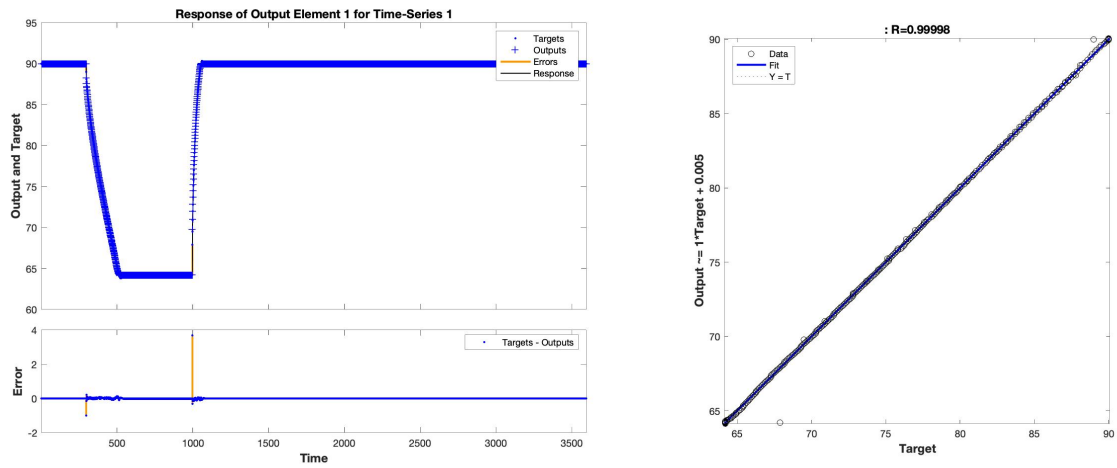


Figure 6.30 – Targets, outputs, error and response for estimation data & Target x estimation correlation (VAL02).

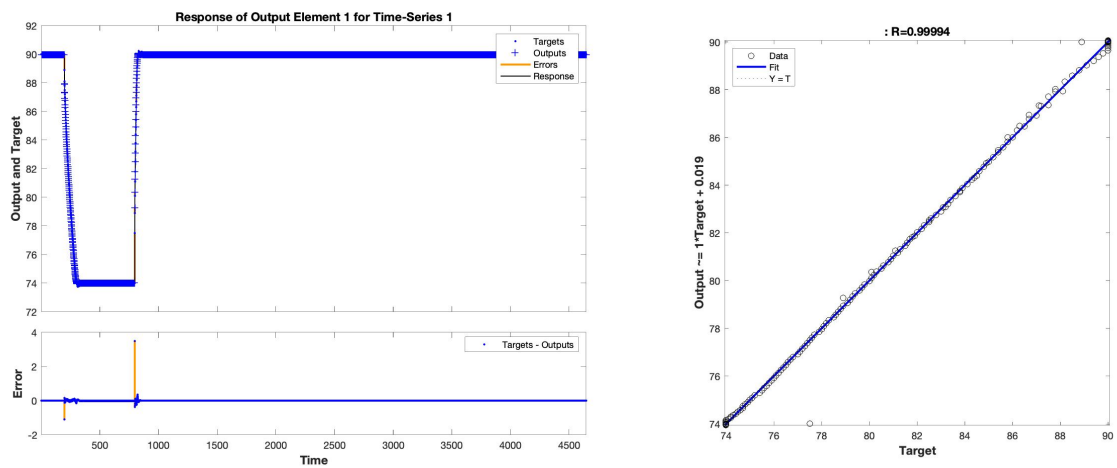


Figure 6.31 – Targets, outputs, error and response for estimation data & Target x estimation correlation (VAL03).

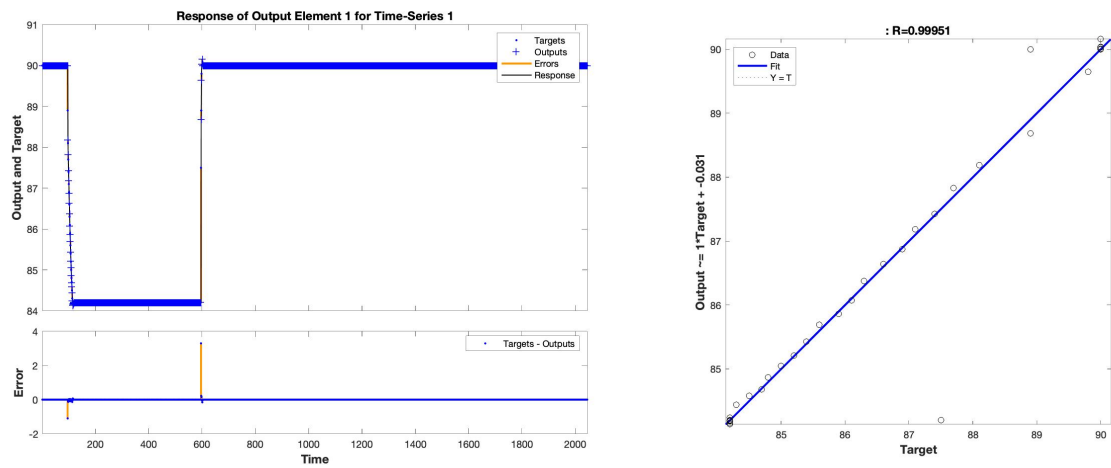


Figure 6.32 – Targets, outputs, error and response for estimation data & Target x estimation correlation (VAL04).

6.2.2 Simulation Results Redux

Having changed the old trained ANN for the the STP -> BP with the new, the model is run again. Table 6.3 shows the improvements on most or the errors. There are some errors that increased, such as VAL04 BP, but this was compensated by the improved stability on the interaction between models STP -> BP and BP -> AR. The values are still close to 10E-2 mse, however, the real effects of this errors can only be seen on the overall longitudinal simulation - that is outside the scope of this work. Figures 6.33 to 6.56 show the model outputs. The change in the STP -> BP has been successful in suppressing the instability seen on the original simulation model.

		Error - mse (10E-3)		Run time [s]	
		New Model	Previous Model	New Model	Previous Model
EST01	BP	5.213	6.500	1.349	1.354
	AR	12.392	1316.5		
	BC	67.390	2557.6		
VAL01	BP	7.734	8.571	1.436	0.825
	AR	9.845	2200.1		
	BC	32.815	3877.7		
VAL02	BP	4.336	4.996	0.762	1.050
	AR	5.917	5.456		
	BC	15.242	19.252		
VAL03	BP	3.195	6.412	0.732	0.739
	AR	18.581	20.655		
	BC	73.963	125.609		
VAL04	BP	10.046	8.424	0.624	0.612
	AR	11.513	10079.5		
	BC	33.471	14655.7		

Table 6.3 – Results of the new simulation and previous simulation.

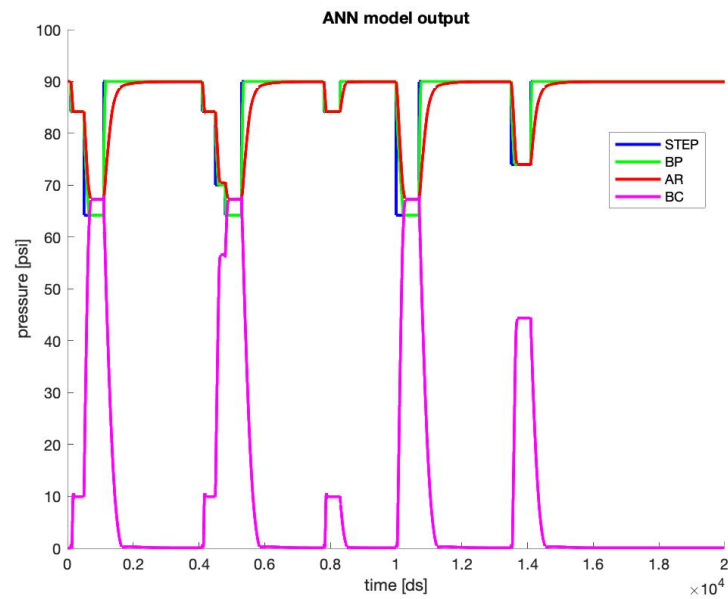


Figure 6.33 – Overall simulation outputs for EST01 - STP, BP, AR, BC.

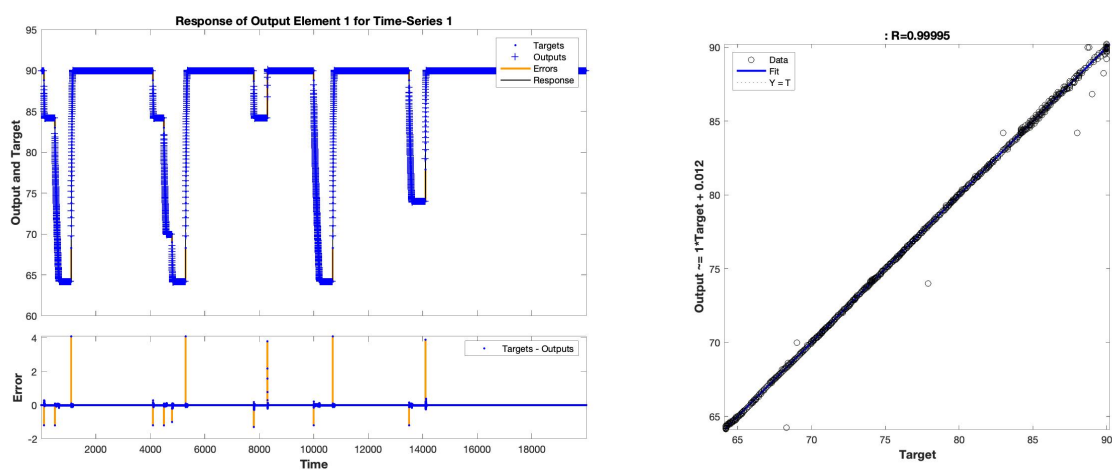


Figure 6.34 – Targets, outputs, error and response for estimation data & Target x estimation correlation (BP - EST01).

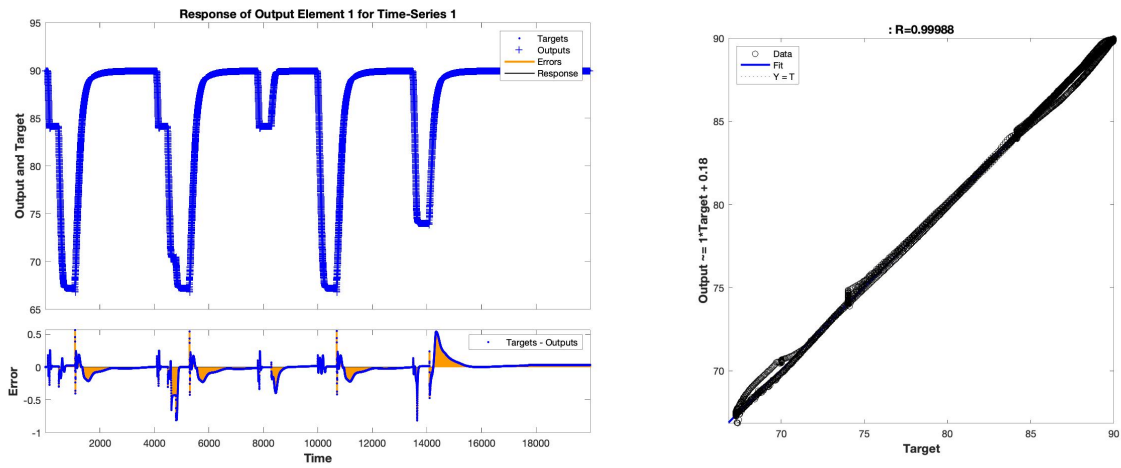


Figure 6.35 – Targets, outputs, error and response for estimation data & Target x estimation correlation (AR - EST01).

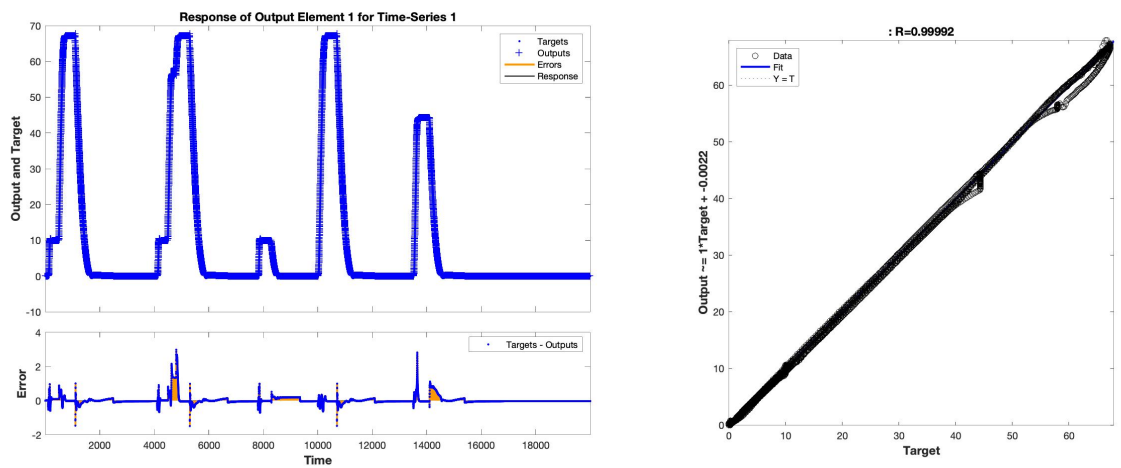


Figure 6.36 – Targets, outputs, error and response for estimation data & Target x estimation correlation (BC - EST01).

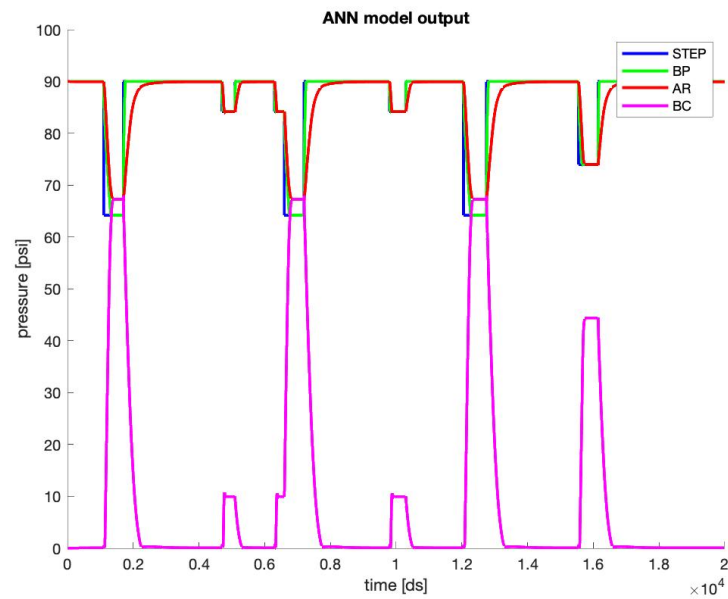


Figure 6.37 – Overall simulation outputs for VAL01 - STP, BP, AR, BC.

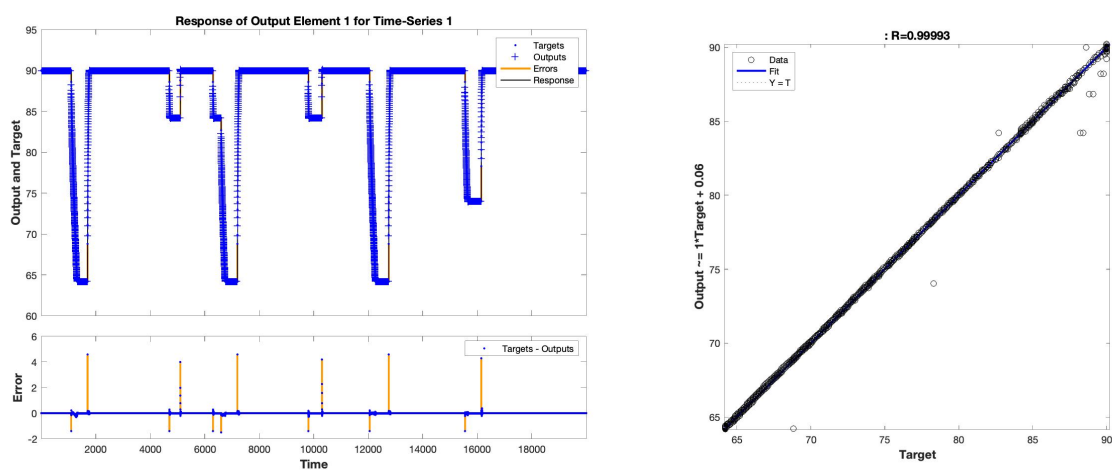


Figure 6.38 – Targets, outputs, error and response for estimation data & Target x estimation correlation (BP - VAL01).

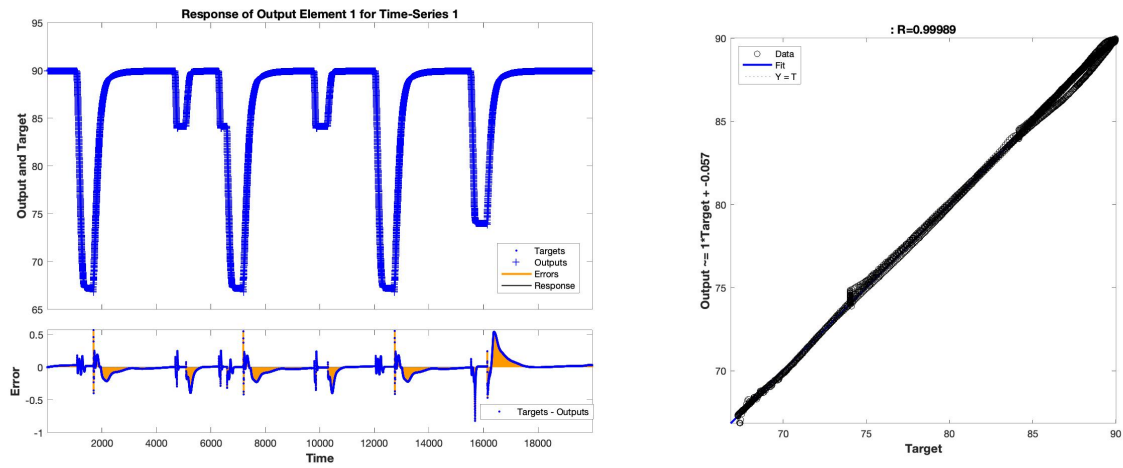


Figure 6.39 – Targets, outputs, error and response for estimation data & Target x estimation correlation (AR - VAL01).

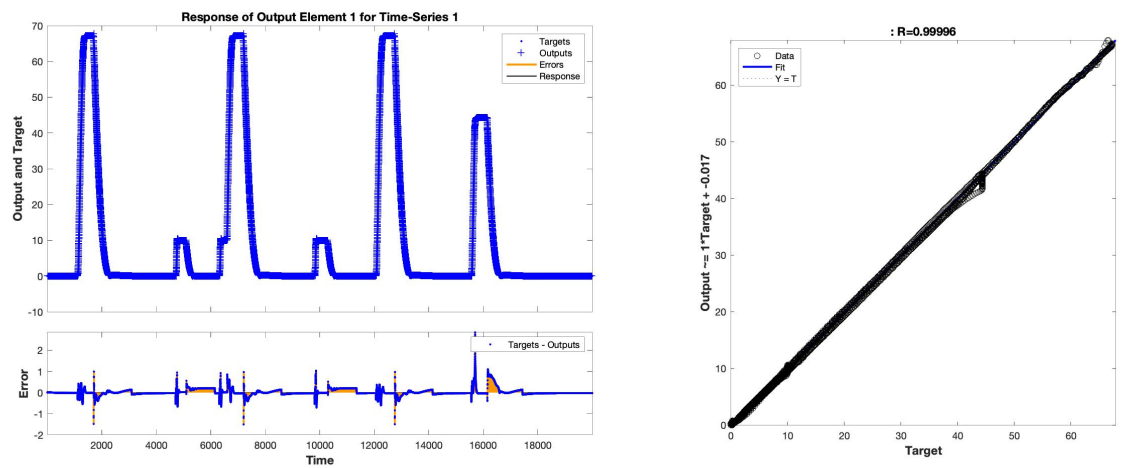


Figure 6.40 – Targets, outputs, error and response for estimation data & Target x estimation correlation (BC - VAL01).

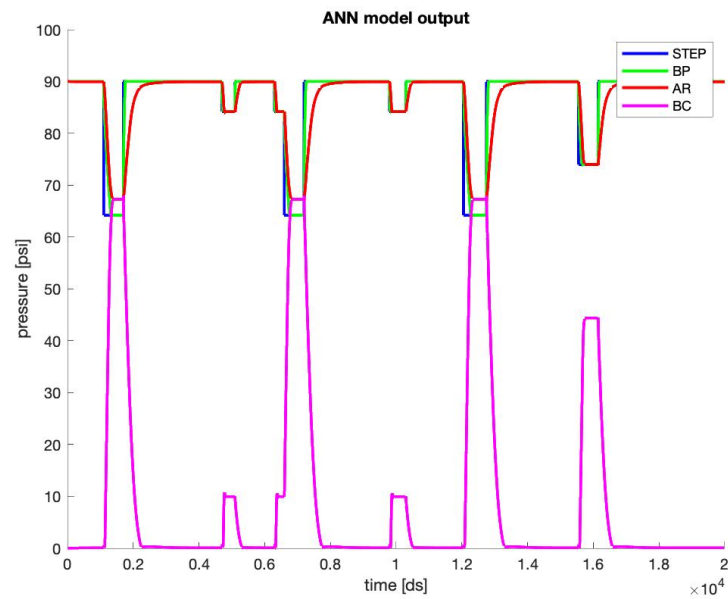


Figure 6.41 – Overall simulation outputs for VAL01 - STP, BP, AR, BC.

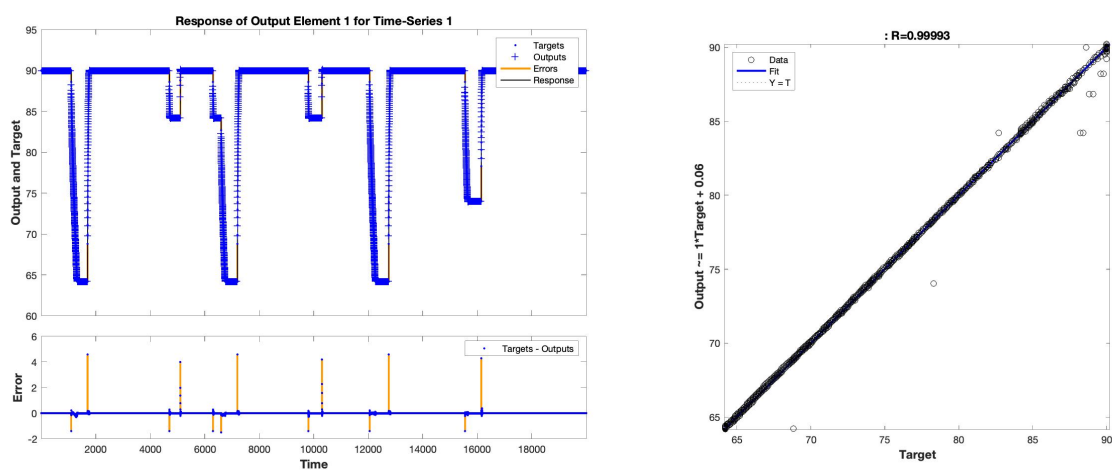


Figure 6.42 – Targets, outputs, error and response for estimation data & Target x estimation correlation (BP - VAL01).

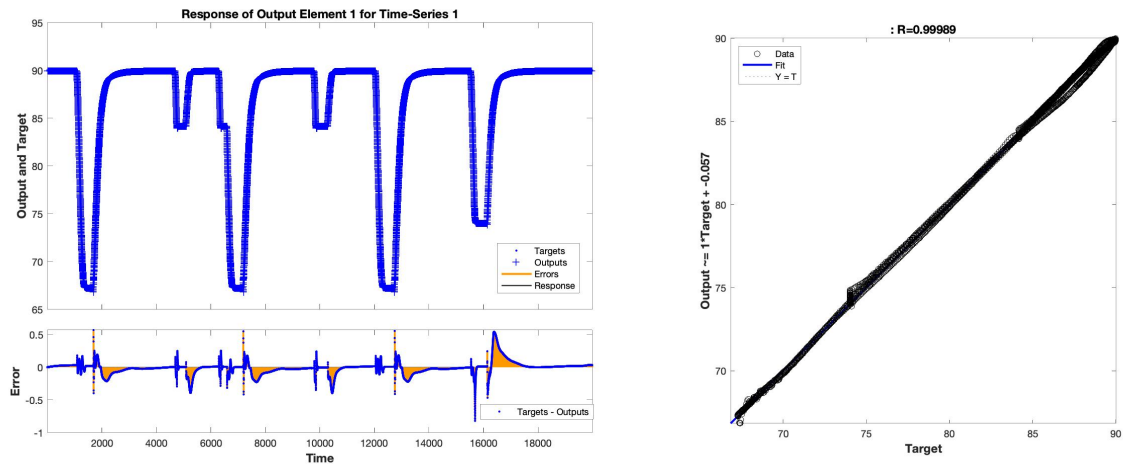


Figure 6.43 – Targets, outputs, error and response for estimation data & Target x estimation correlation (AR - VAL01).

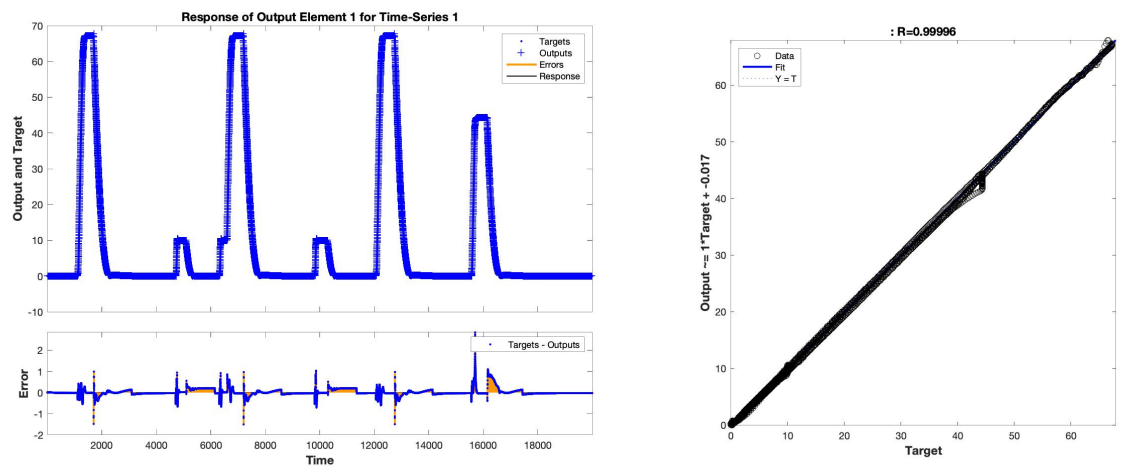


Figure 6.44 – Targets, outputs, error and response for estimation data & Target x estimation correlation (BC - VAL01).

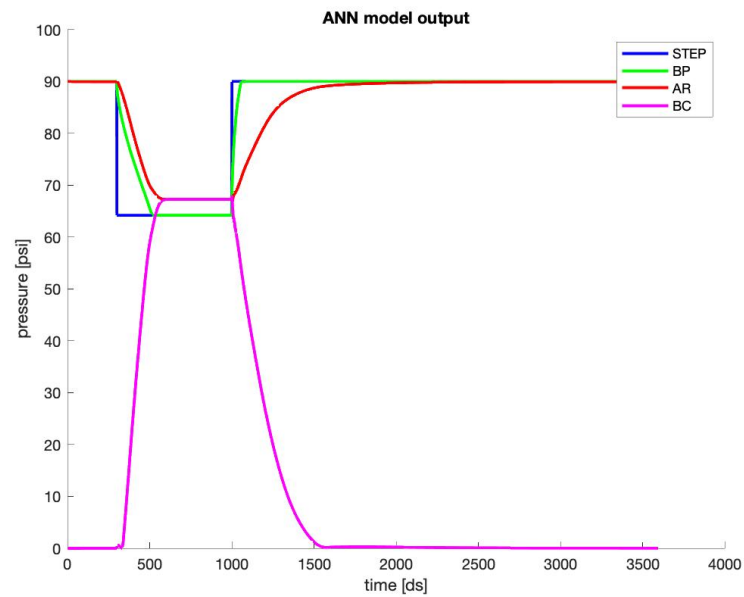


Figure 6.45 – Overall simulation outputs for VAL02 - STP, BP, AR, BC

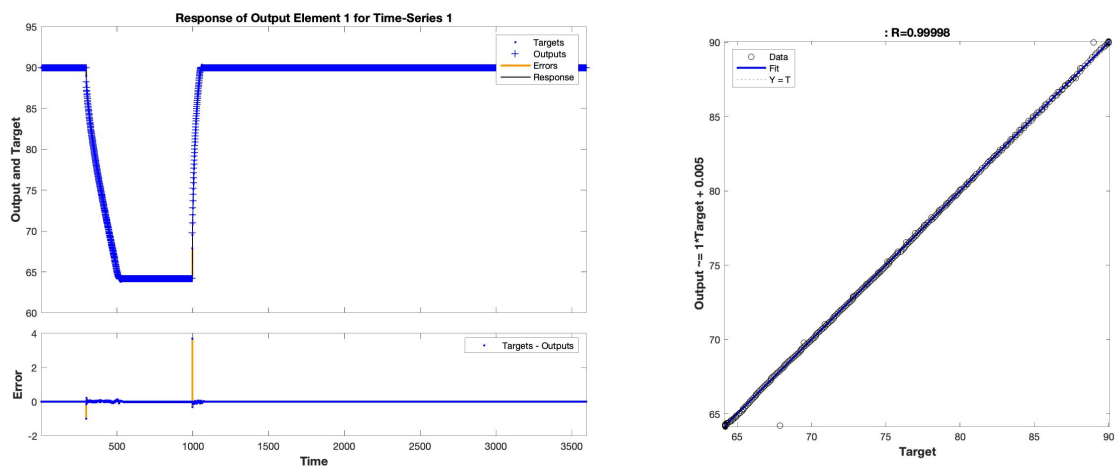


Figure 6.46 – Targets, outputs, error and response for estimation data & Target x estimation correlation (BP - VAL02).

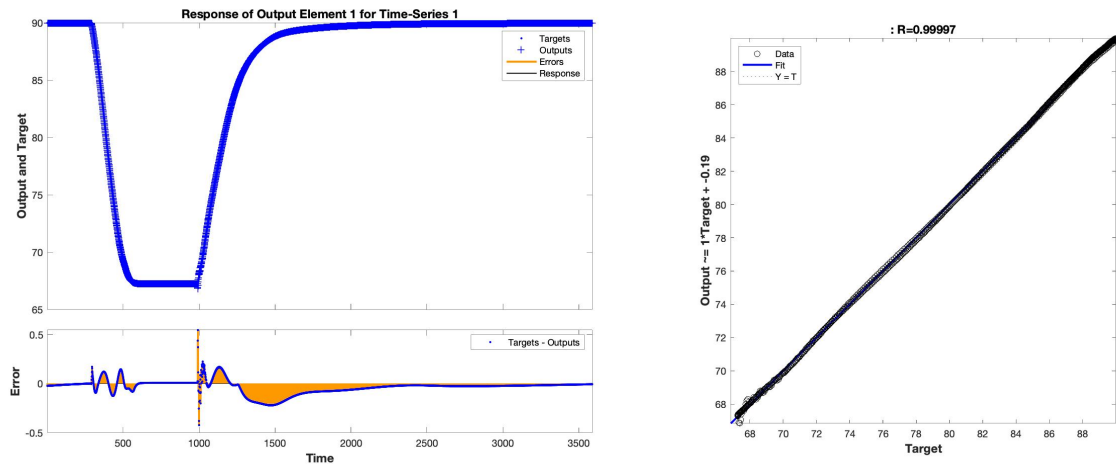


Figure 6.47 – Targets, outputs, error and response for estimation data & Target x estimation correlation (AR - VAL02).

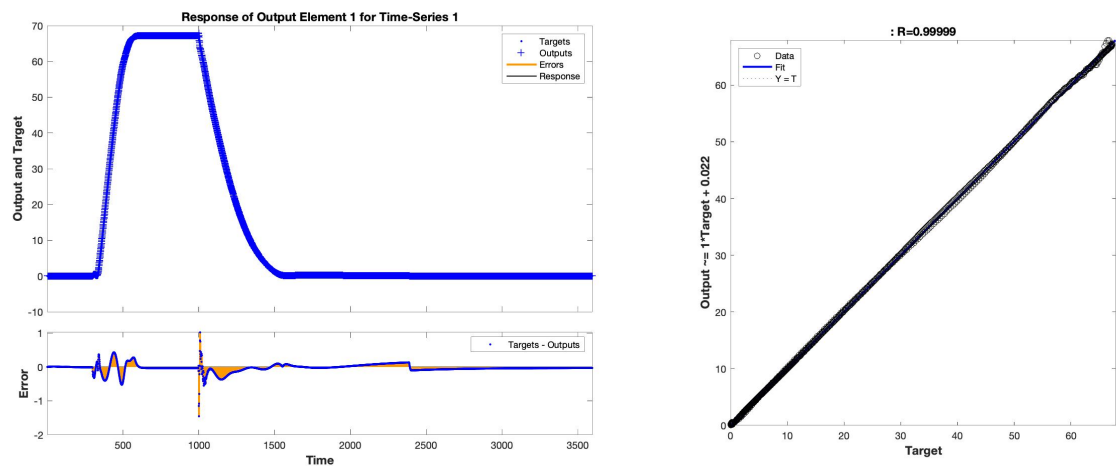


Figure 6.48 – Targets, outputs, error and response for estimation data & Target x estimation correlation (BC - VAL02).

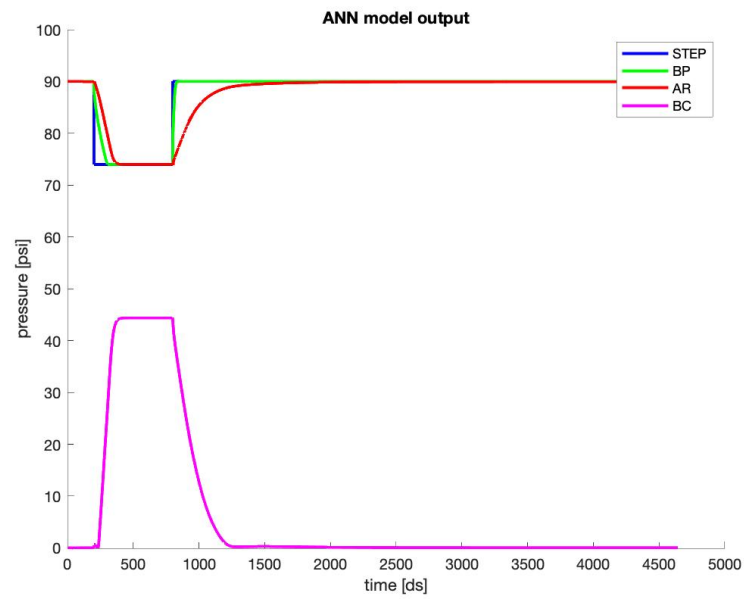


Figure 6.49 – Overall simulation outputs for VAL03 - STP, BP, AR, BC.

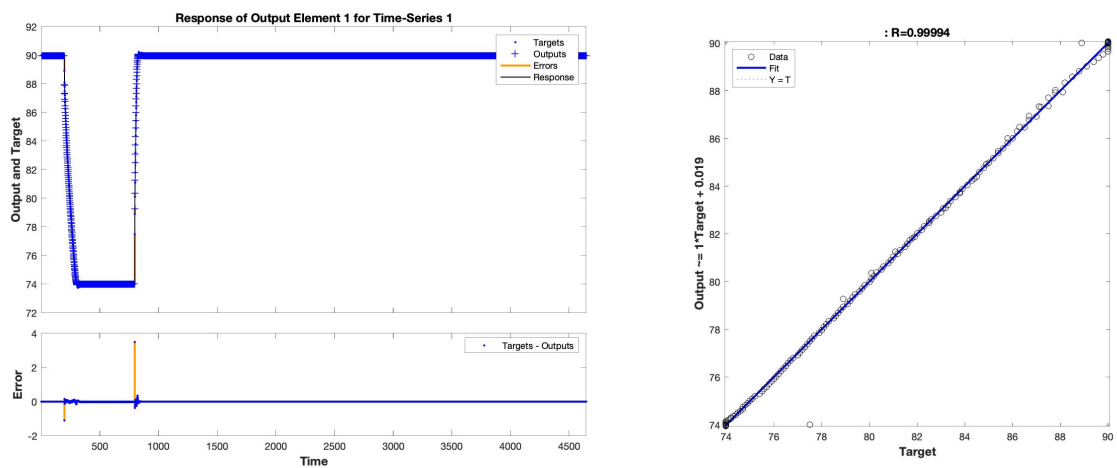


Figure 6.50 – Targets, outputs, error and response for estimation data & Target x estimation correlation (BP - VAL03).

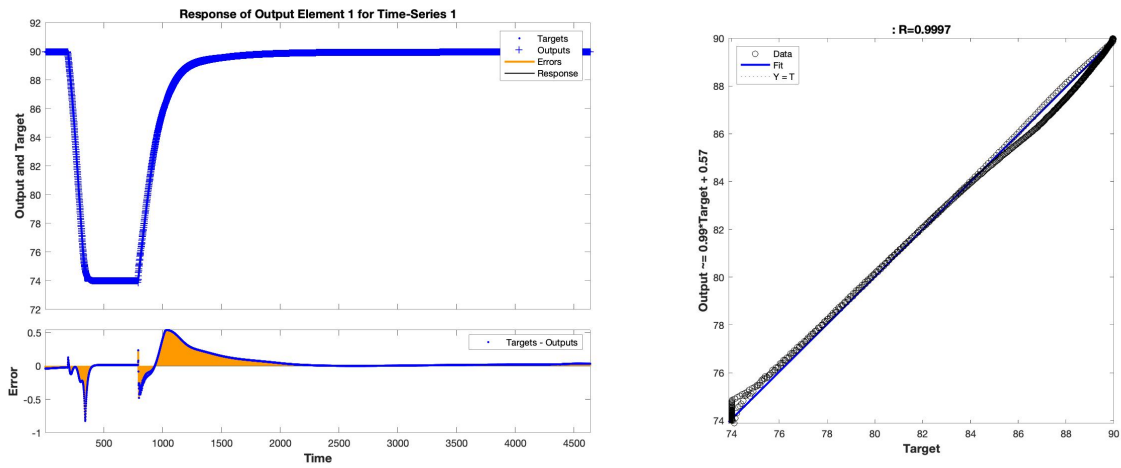


Figure 6.51 – Targets, outputs, error and response for estimation data & Target x estimation correlation (AR - VAL03).

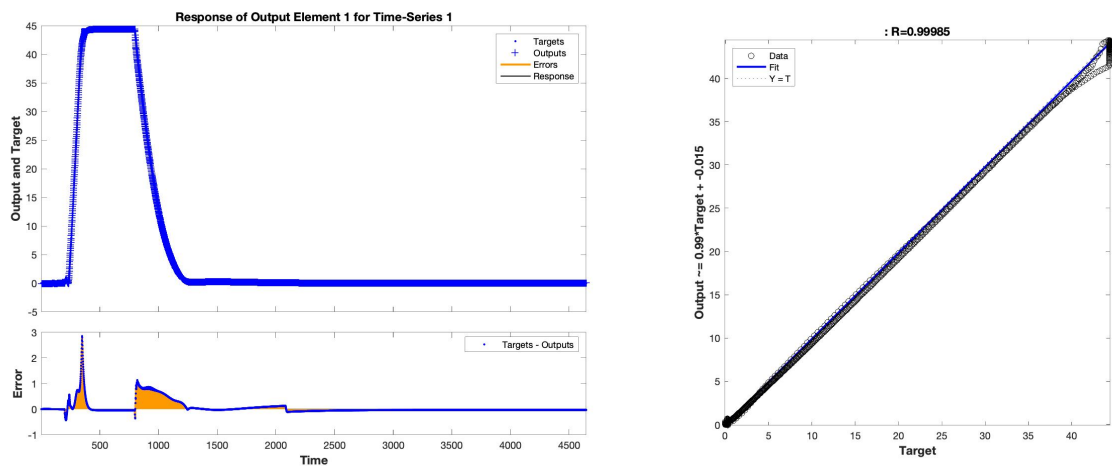


Figure 6.52 – Targets, outputs, error and response for estimation data & Target x estimation correlation (BC - VAL03).

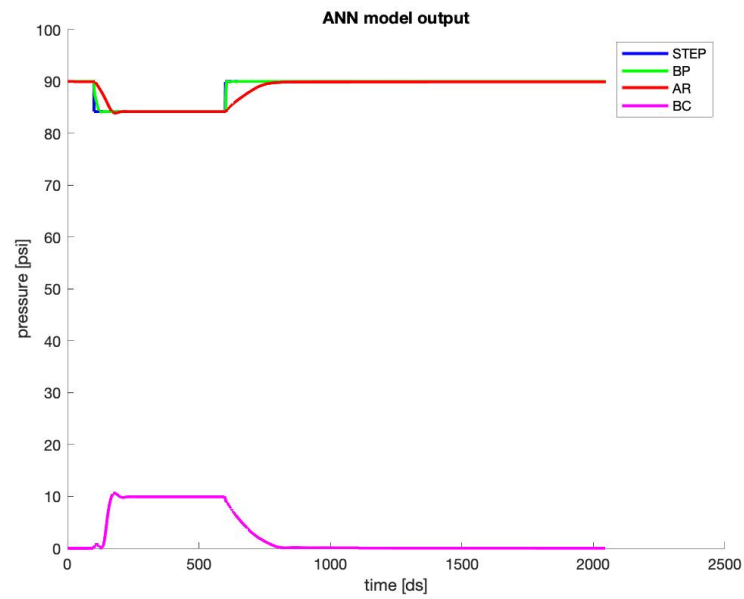


Figure 6.53 – Overall simulation outputs for VAL04 - STP, BP, AR, BC.

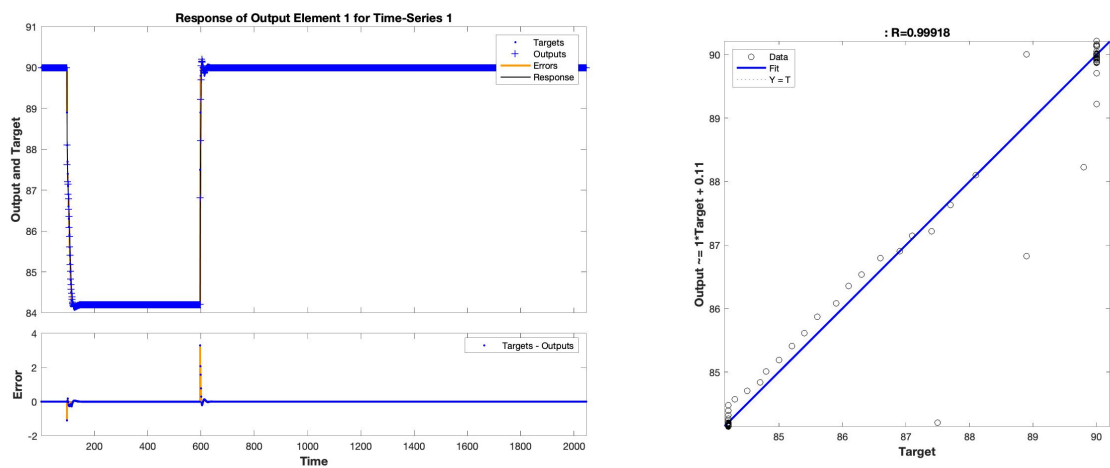


Figure 6.54 – Targets, outputs, error and response for estimation data & Target x estimation correlation (BP - VAL04).

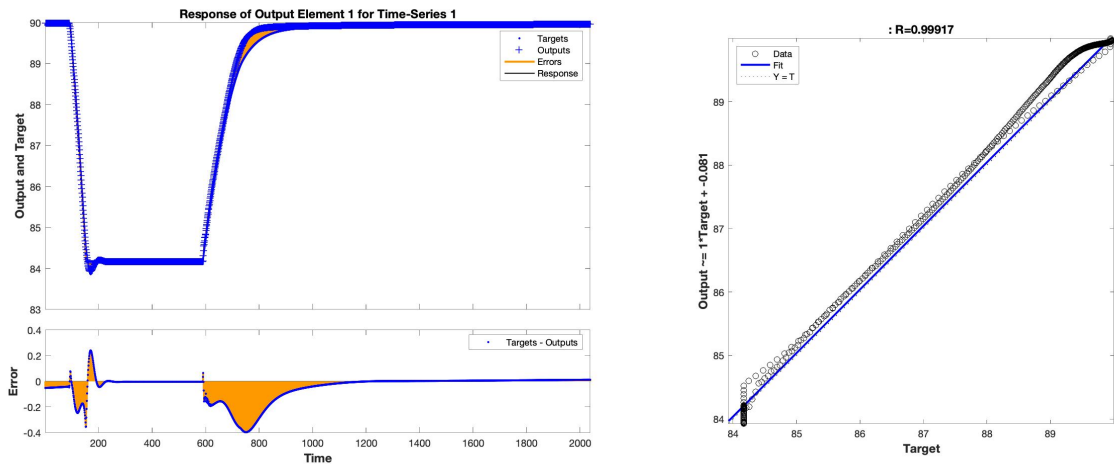


Figure 6.55 – Targets, outputs, error and response for estimation data & Target x estimation correlation (AR - VAL04).

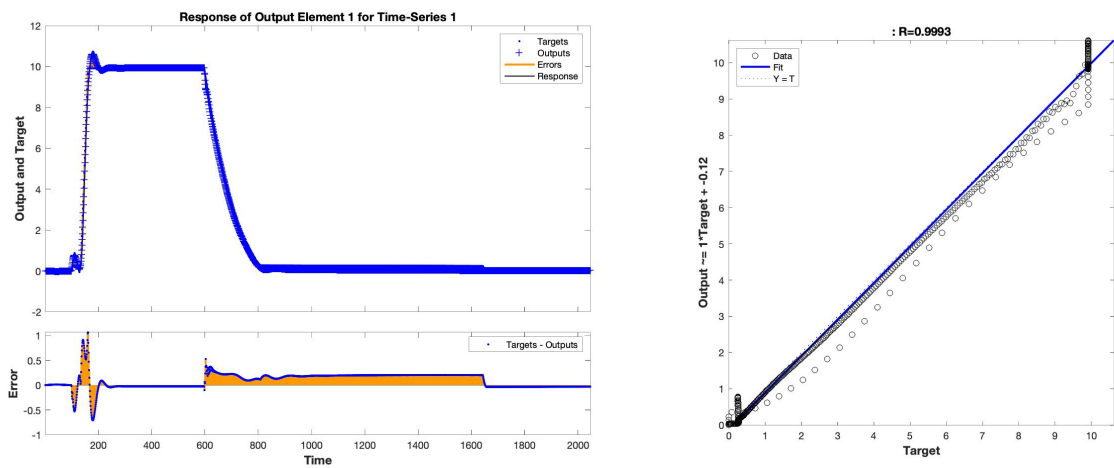


Figure 6.56 – Targets, outputs, error and response for estimation data & Target x estimation correlation (BC - VAL04).

6.3 Comparing the Proposed with the Original Model

A model that agrees with the target data was developed with reasonable accuracy. Now, the question to whether it is faster than the current model remains.

It was possible to get hold of a version of the previous program to solve the pressure of the brake system. The program is capable of handling a multi vehicle composition and service & emergency applications. At the present chapter, only the service brake feature was explored. The next chapter explores the combined emergency and service brake model. Also, the number of vehicles on the composition was set to the minimum that the program is capable of - 1 locomotive and a pair of ore freight cars.

A transcript of the program is on appendix 3. The program is comprised of three elements: a data conditioner called "Run_Freio", a processing part made on simulink (Freio) and a plot program for the output (Plot_Freio). The program is time based, so one has to manually inform the program how long each case is to be run. The middle processing part is what is going to be timed. To assess the timing, a simple line on matlab was used when running the program. The command is called as *tic; sim('Simulink_FREIO_V02'); toc* as only the processing is done on this bit, it is believed that the time assessment is accurate. The commands *tic* and *toc* record the time it takes to run the function in between. The original program, as commented earlier, outputs the pressure values in Pa rather than the usual psi unit. Also, the output is in ms and the proposed model output is in ds.

Five runs for each program were made. The pressures and run times were recorded to allow comparisons to be made. The run time distribution is assumed to be normal. The value interval assumes a confidence level of 2 sigma (96%). Table 6.4 shows the time differences between the original and proposed model. The proposed model has an advantage over the original in terms of run time. For the cases studied, there was a reduction in run time from 3.5x to 15x, depending on conditions. All runs were made on the same hardware and with the same programs open. Hardware specifications can be found on appendix 2.

	Run time interval [s]				simulation length [s]
Case	original model		proposed model		
EST01	19.97	22.30	1.37	1.48	2000
VAL01	19.41	22.46	1.40	1.49	2000
VAL02	3.21	5.36	0.76	0.86	360
VAL03	4.67	5.60	0.77	0.88	465
VAL04	2.23	2.62	0.66	0.79	205

Table 6.4 – Run time for original model and proposed model.

Additionally, one can also evaluate the run time per simulation length (table 6.5), where the simulation length is divided by the time it takes to run. The run time of each simulation second is quite consistent on the original model, whereas, on the proposed model, the time per simulation second increases with the decrease in simulation time. On the proposed model, for the small simulations, the time per simulated second increases substantially, indicating that other service functions are taking more time than the simulation itself. On a long simulation with several vehicles, the time taken on the service functions will be diluted on the overall time.

	time per simulation second [ms] interval				simulation length [s]
Case	original model		proposed model		
EST01	9.98	11.15	0.68	0.74	2000
VAL01	9.70	11.23	0.70	0.74	2000
VAL02	8.90	14.91	2.12	2.41	360
VAL03	10.05	12.05	1.66	1.89	465
VAL04	10.85	12.68	3.22	3.87	205

Table 6.5 – Run time for original model and proposed model.

7 EMERGENCY RESERVOIR AND BRAKE

The current models also allow for the application of the emergency brakes. These are commanded by the emergency reservoir and the rate of decrease of the brake pipe pressure. When an emergency is triggered, the brake pipe goes to zero psi. The fact that the pressure drops to the same as ambient pressure is not the triggering of the emergency brakes but rather the depressurisation rate. To allow for this, the implementation of the system has valves distributed on the freight cars that helps the depressurisation. The brake valves open the brake pipe to ambient to further allow the pressure to drop.

The emergency brake application can be commanded by the conductor or by the train itself, in the case of a "break in two" or if the consist is severed, separating the brake pipe in two parts and leading its pressure to zero. .

As a side note, the brake can be applied only if there is pressure on the reservoirs. A consist that is left without pressure on the air brake systems is free to move. This feature is crucial when assembling a train and moving the carts and locomotives, but can also be dangerous when vehicles are left on a grade unattended and the parking brakes are not properly engaged.

On this section the features of emergency brake and reservoir will be added to the model. The new proposed topology is presented on figure 7.1

First, the new inputs will be presented and discussed. A new version of the estimation data will be concocted as well as a new version for validation 1 and a new validation case, 5. The emergency reservoir has to be added to the validation and estimation cases. New models for the step input and the brake pipe outputs (STP -> BP) have to be trained (the pressure now goes to zero with the emergency application), a model for emergency reservoir pressure from brake pipe inputs (BP -> ER). The auxiliary reservoir pressure now is a function of the brake pipe and the emergency reservoir (BP + ER -> AR). This is assumed this way for on the ABDX valves, the emergency reservoir helps equalize the auxiliary reservoir on the recharging phase. The final model is the brake cylinder output - that takes into account the other pressure inputs (except the step that has already been converted to brake pipe pressure) - (BP + AR + ER -> BC). The new models have more inputs than those used on the service brake model.

As the training and the models were addressed in detail on the previous chapters, the training outputs will be presented on tables for the various options and graphs are presented

for the option chosen to move forward.

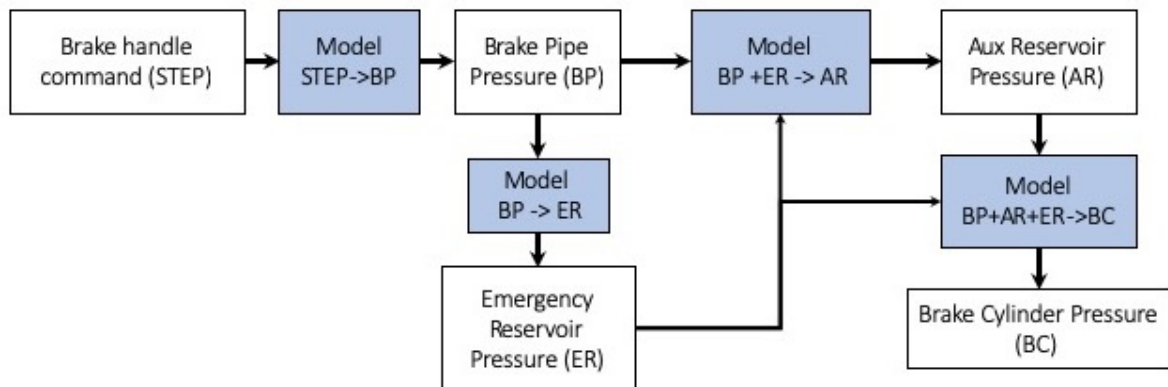


Figure 7.1 – Proposed topology with emergency reservoir.

7.1 Inputs

On this part, the new inputs will be explored and explained.

7.1.1 VAL05 - Emergency Brake

A new set of training data was received from the lumped parameter model. These data will be henceforth be called Validation 5 or VAL05 for short. The data set contains a pressure drop from 90 psi to 0 psi (STEP) and the pressures for the brake pipe (BP), auxiliary reservoir (AR), emergency reservoir (ER) and brake cylinder (BC). Figure 7.2 shows the time series plot.

The STEP vector for VAL05 is

```
vector_VAL05=[ 0 90
               151 0
               601 90
               3000 90];
```

7.1.2 EST01 - Estimation Data

The estimation/training data has also been updated by adding the emergency brake at the end of the previous estimation data. This new time series will be called EST01_v2. Also, the emergency reservoir was added to the estimation data. Figure 7.3 shows the new estimation/training data.

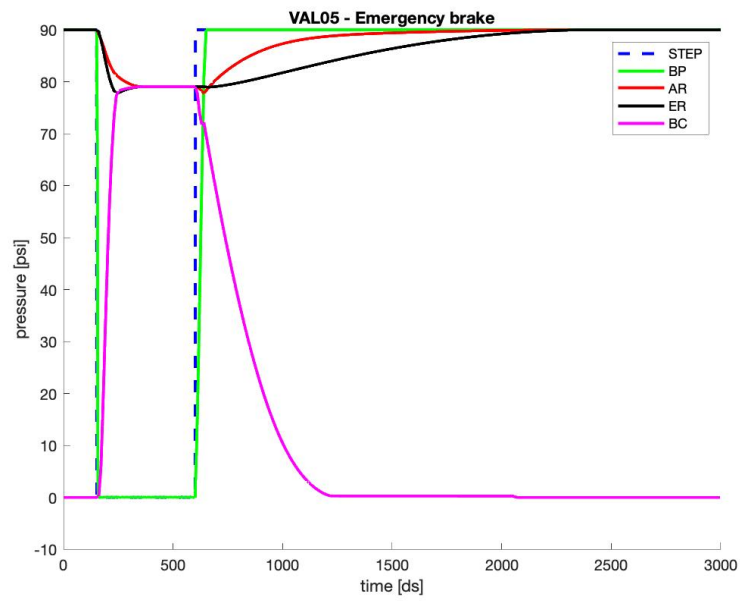


Figure 7.2 – Emergency brake validation data

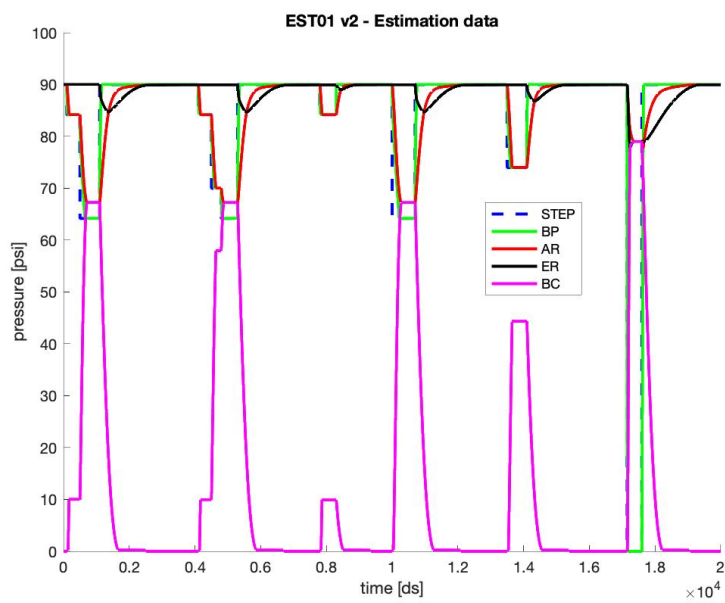


Figure 7.3 – Estimation data (EST01 v2) with the added emergency brake and reservoir.

The new STEP vector is

```
EST01 =[ 0 90
         100 84.2
         500 64.2
         1100 90
         4100 84.2
```

```

4500 70
4800 64.2
5300 90
7800 84.2
8300 90
10000 64.2
10700 90
13500 74
14100 90
17150 0
17600 90
20000 90];

```

7.1.3 VAL01 - Validation Data

On the same way, the contents of VAL05 were added to the VAL01 dataset, creating VAL01 v2. The emergency brakes were added before the other validations and after the composed brake applications. Also, the emergency reservoir pressure was added to the dataset. Figure 7.4 shows the time series plots.

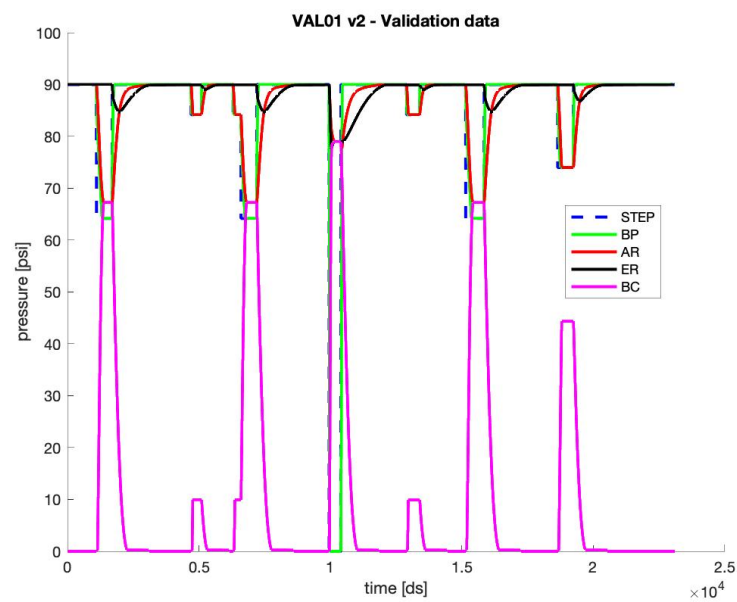


Figure 7.4 – Validation (VAL01 v2) with the added emergency brake and reservoir.

```

VAL01 =[ 0 90
         1100 64.2
         1700 90
         4700 84.2
         5100 90
         6300 84.2
         6600 64.2
         7200 90
         9950 0
         10401 90
         12900 84.2
         13400 90
         15150 64.2
         15850 90
         18650 74
         19250 90
         23096 90];

```

7.1.4 Other Validation Data

As for the other validation datasets, the emergency reservoir had to be added to allow for the proper estimation and description of the phenomena. The STEP vectors were not changed. Figure 7.5 shows the pressures for the other validation datasets.

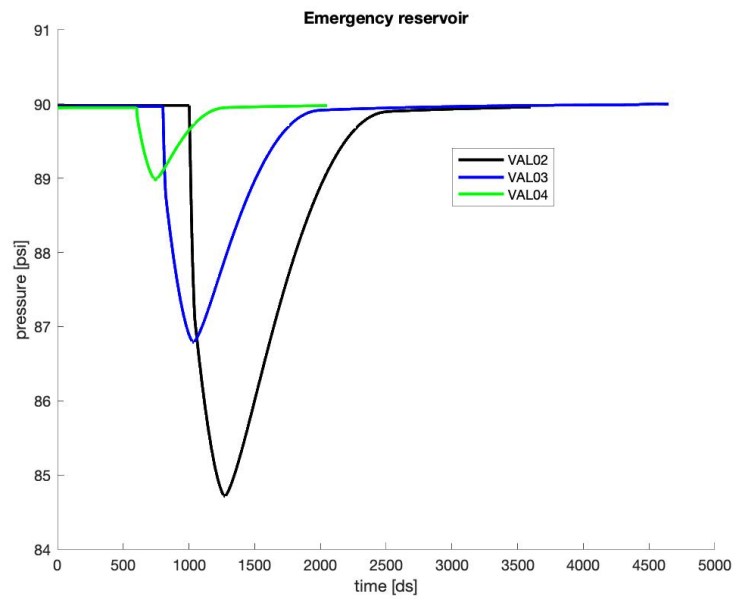


Figure 7.5 – Emergency reservoir pressure for VAL02, VAL03 and VAL04.

7.2 Training the Models

As outlined on the beginning of this chapter, a new set of models has to be trained in order to represent the new features of emergency brakes and reservoir. The next sections will show the trained models, their performance and the graphs of the best candidates to build the overall simulator model at the end of this chapter. The training of the parts were somewhat more complicated than previously. The approach was to train all the ANNs as closed loop, since previously, the open loop training did not work as intended when closed. The Levenberg-Marquardt training scheme with bayesian validation was used again. The trainings took longer than the service application models

7.2.1 STEP -> BP

Several combinations were run for the model that translates the discrete input to the pressure of the brake pipe. Table 7.1 shows the combinations. The main parameter is the performance and the secondary parameter is the run time. All run times were quite similar, so the best performance model was selected to compose the system simulation contraption. Figures 7.6 to 7.11 shows the model response against the targets as well as the regression plots. The model struggles with some of the transitions - especially on the emergency brake command. Each model took approximately 60 to 80 min to run.

STP ->BP	Case	A	B	C	D	E	F	G
input delays		4	4	4	4	4	6	4
output delays		4	4	4	4	4	6	4
hidden neurons		10	15	20	25	30	20	40
Error (mse)	Average	0.068	0.026	0.019	0.048	0.109	0.124	0.152
Error (mse) 1.00E-03	EST01	63.443	25.312	17.604	38.072	107.036	109.222	143.746
	VAL01	58.368	23.799	18.620	37.454	99.294	102.170	127.693
	VAL02	51.734	21.245	20.010	20.598	118.541	70.531	108.698
	VAL03	34.839	9.634	6.493	14.811	40.782	58.978	99.524
	VAL04	44.997	40.622	25.505	40.649	91.850	96.693	118.444
	VAL05	156.396	35.332	22.953	134.189	197.494	307.434	315.162
Run time [s]	Average	0.163	0.149	0.182	0.150	0.154	0.149	0.152
	EST01	0.362	0.325	0.300	0.306	0.316	0.323	0.327
	VAL01	0.209	0.196	0.239	0.206	0.195	0.201	0.200
	VAL02	0.121	0.112	0.181	0.118	0.131	0.114	0.112
	VAL03	0.065	0.050	0.073	0.055	0.057	0.545	0.055
	VAL04	0.186	0.177	0.246	0.177	0.189	0.169	0.185
	VAL05	0.037	0.032	0.054	0.036	0.035	0.031	0.035
		0	++	+++	+	-	--	---

Table 7.1 – Results of the STP -> BP model.

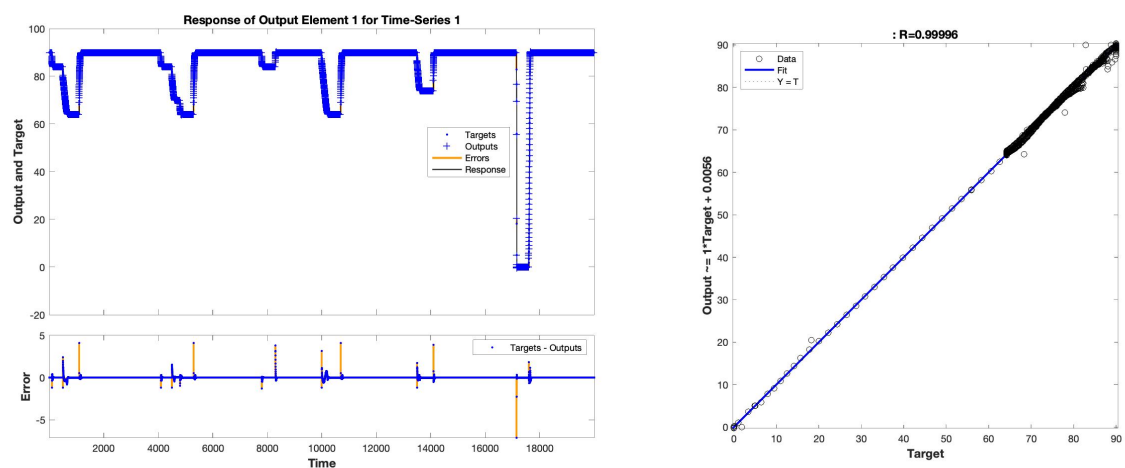


Figure 7.6 – Targets, outputs, error and response for estimation data & Target x estimation correlation (BP - EST01) - time in [ds] and values in [psi].

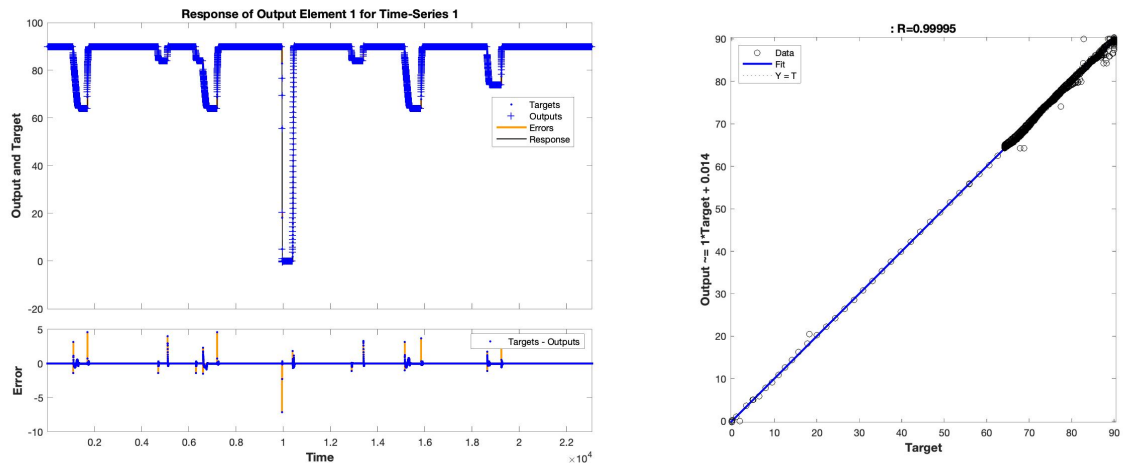


Figure 7.7 – Targets, outputs, error and response for estimation data & Target x estimation correlation (BP - VAL01) - time in [ds] and values in [psi].

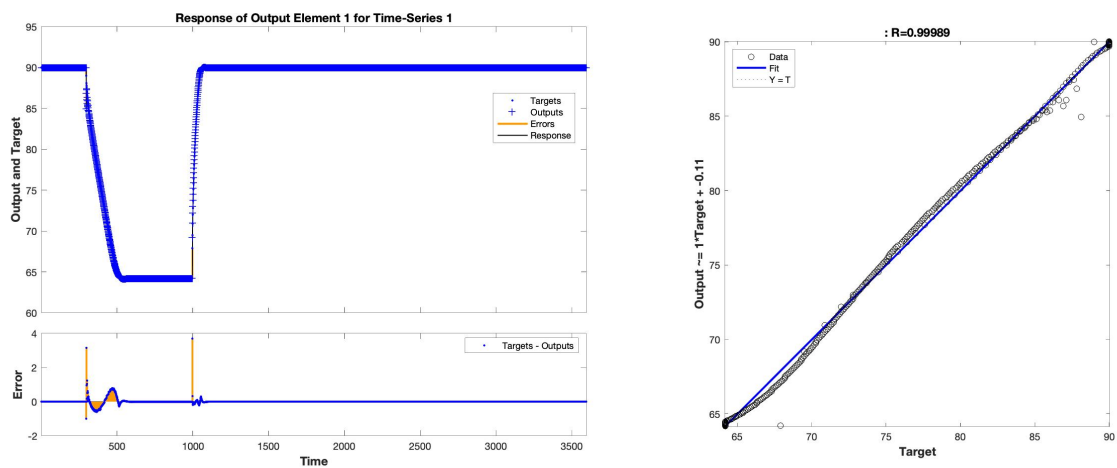


Figure 7.8 – Targets, outputs, error and response for estimation data & Target x estimation correlation (BP - VAL02) - time in [ds] and values in [psi].

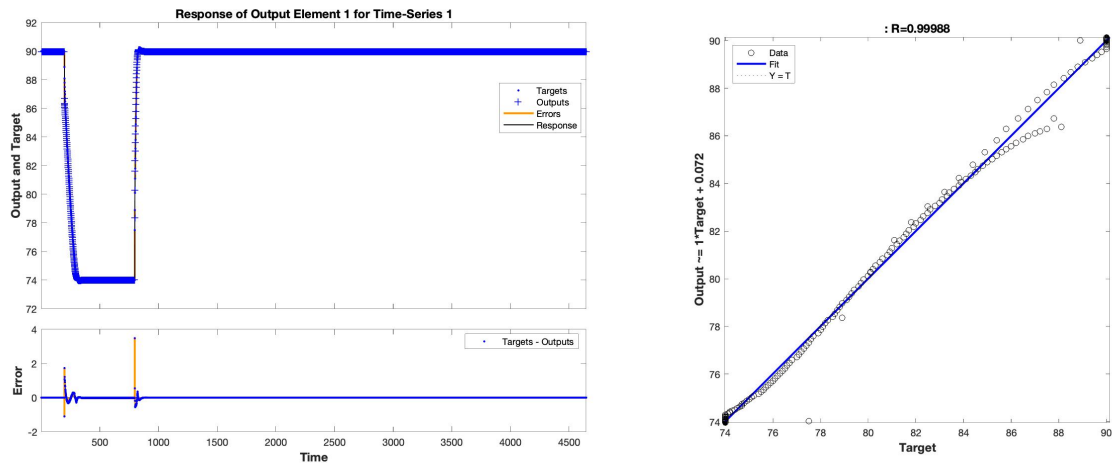


Figure 7.9 – Targets, outputs, error and response for estimation data & Target x estimation correlation (BP - VAL03) - time in [ds] and values in [psi].

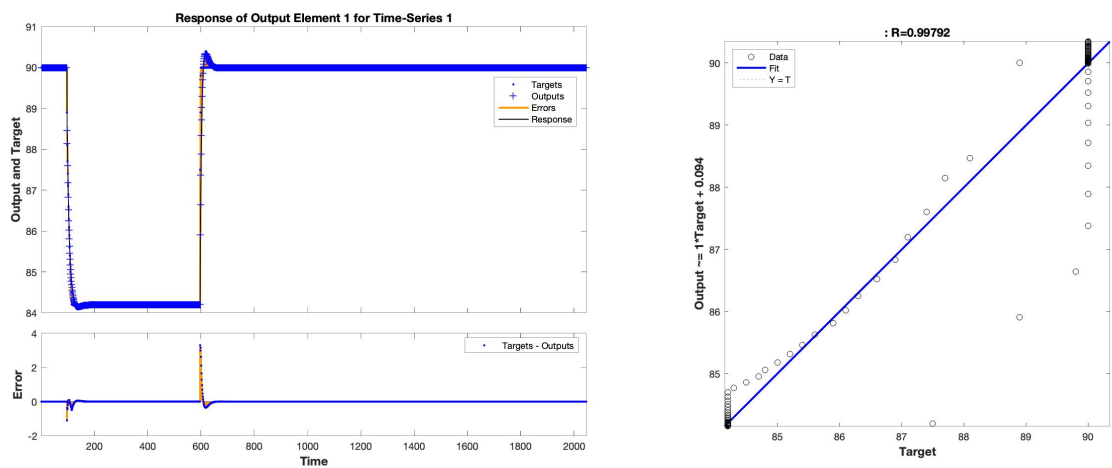


Figure 7.10 – Targets, outputs, error and response for estimation data & Target x estimation correlation (BP - VAL04) - time in [ds] and values in [psi].

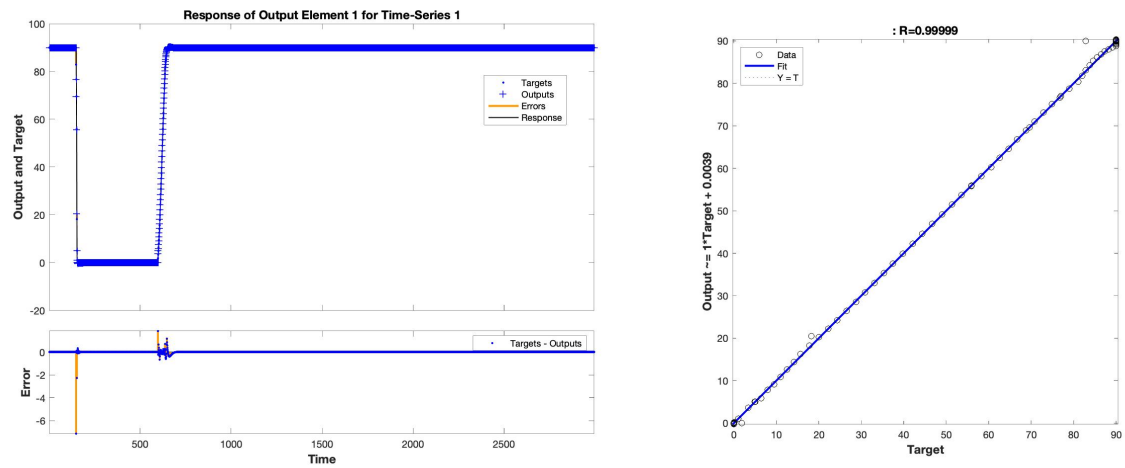


Figure 7.11 – Targets, outputs, error and response for estimation data & Target x estimation correlation (BP - VAL05) - time in [ds] and values in [psi].

7.2.2 BP -> ER

This was perhaps the most challenging model to train. In the physical world the pressures of the aux. and emergency reservoir are linked with the brake pipe and the brake cylinder. The previous model addresses this by calculating all the parameters on a time step, allowing for their interactions. However, the approach taken on this work was to calculate the time series of the pressures of each of the elements separately. An additional feature of the model is that it acts on the pressure raise for the service brake application and on the pressure fall for the emergency application. There is no way to inform this peculiarity to the artificial neural network that is being used, other than separate the model in two - one for the service and other for the emergency applications - what is not a desirable approach in terms of versatility. The model training times took between 360 and 480 min - among the longest experienced on this work. Table 7.2 shows the best cases and combinations. Runs that yielded error higher than 1 mse are not reported. The best model combination is model D - 15 delays on the input and output and 20 hidden neurons. An alternative model is C with 18 delays and 20 neurons. Model E shows that increasing the number of neurons does not increase the performance for this case.

BP ->ER	Case	A	B	C	D	E
input delays		5	2	18	15	18
output delays		5	2	18	15	18
hidden neurons		38	20	20	20	40
Error (mse)	Average	0.111	0.427	0.008	0.002	0.025
Error (mse) 1.00E-03	EST01	93.326	321.890	5.781	2.547	29.339
	VAL01	100.335	937.613	6.749	2.422	26.824
	VAL02	111.298	202.472	4.438	2.932	32.265
	VAL03	131.723	678.417	3.460	2.405	26.604
	VAL04	43.596	81.626	19.096	1.310	17.620
	VAL05	187.147	336.998	6.320	1.689	19.679
Run time [s]	Average	0.069	0.069	0.105	0.070	0.079
	EST01	0.135	0.136	0.179	0.141	0.161
	VAL01	0.161	0.156	0.247	0.158	0.169
	VAL02	0.030	0.032	0.051	0.031	0.036
	VAL03	0.041	0.038	0.064	0.038	0.044
	VAL04	0.022	0.021	0.039	0.022	0.028
	VAL05	0.027	0.031	0.053	0.028	0.035
		--	---	+	+++	o

Table 7.2 – Results of the BP -> ER model.

Figures 7.12 to 7.17 show the response and the regression for the model over the validation data sets. At first glance, figure 7.16 may cause alarm. The model regression is

among the lowest among accepted and presented models at an R squared of 0.899 (other models with worst performances were trained and are not presented). However, the absolute values are quite low - the emergency reservoir does not play an important part on the minimum brake application and the absolute errors are lower than 1psi, in line with the other data sets. Effort was made to try to align the model with the VAL04 data but, in general, models that presented a good result for this case were not acceptable for the other validation cases. This is an important point to be noted when deploying the model on the simulator. If it plays an important part on the longitudinal dynamics, it will have to be reassessed. Nonetheless, it is not believed that this parameter plays an important role on the train simulator for the main input - brake pipe pressure - and the main output - brake cylinder pressure - are well aligned with the target values and the emergency reservoir plays an important role on the emergency brake application and on the pressure recovery after maximum service brake applications.

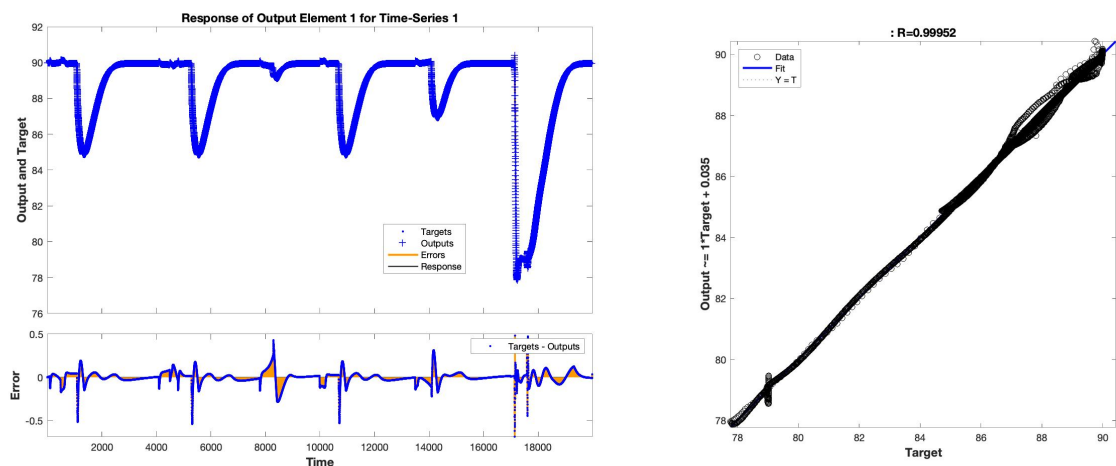


Figure 7.12 – Targets, outputs, error and response for estimation data & Target x estimation correlation (ER - EST01) - time in [ds] and values in [psi].

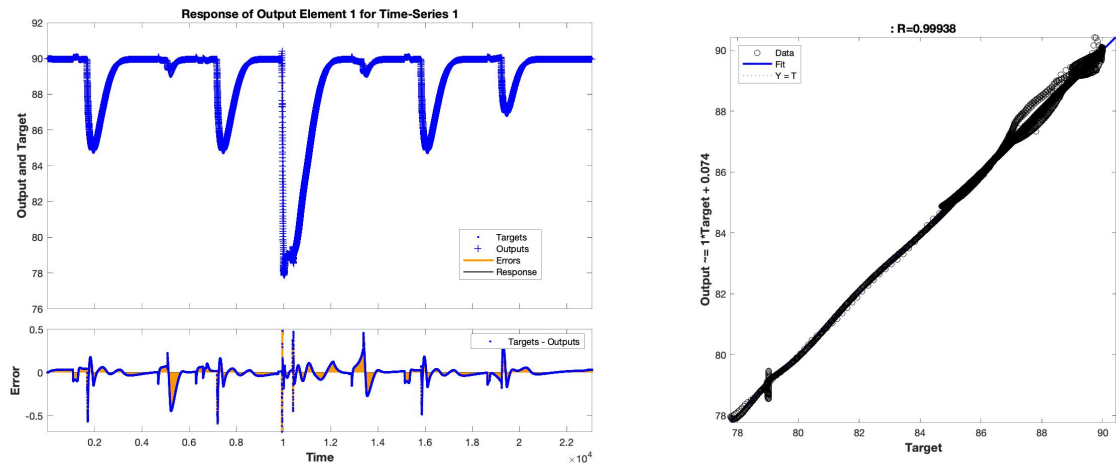


Figure 7.13 – Targets, outputs, error and response for estimation data & Target x estimation correlation (ER - VAL01) - time in [ds] and values in [psi].

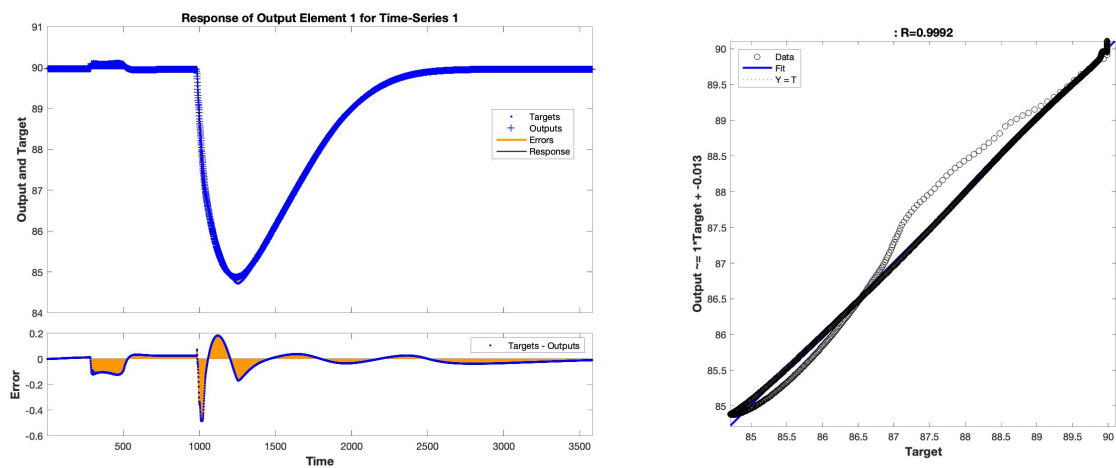


Figure 7.14 – Targets, outputs, error and response for estimation data & Target x estimation correlation (ER - VAL02) - time in [ds] and values in [psi].

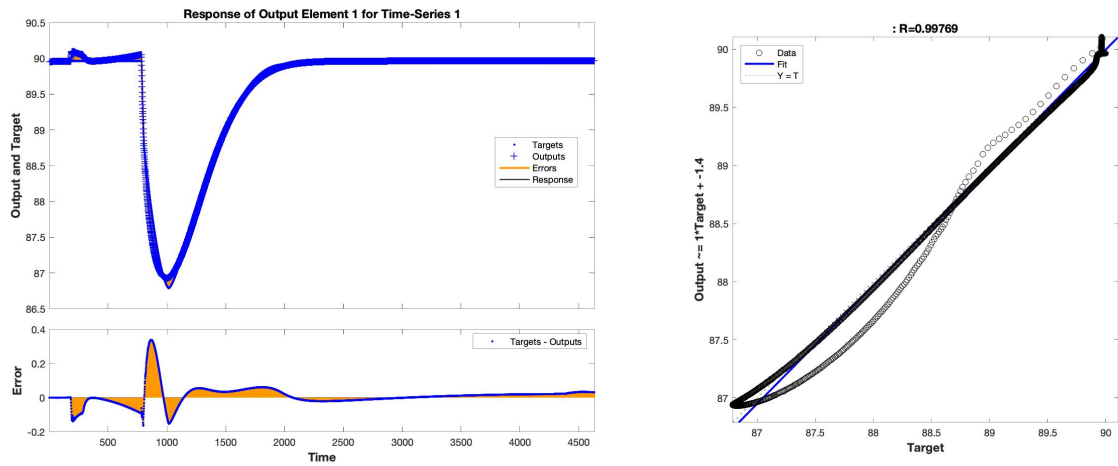


Figure 7.15 – Targets, outputs, error and response for estimation data & Target x estimation correlation (ER - VAL03) - time in [ds] and values in [psi].

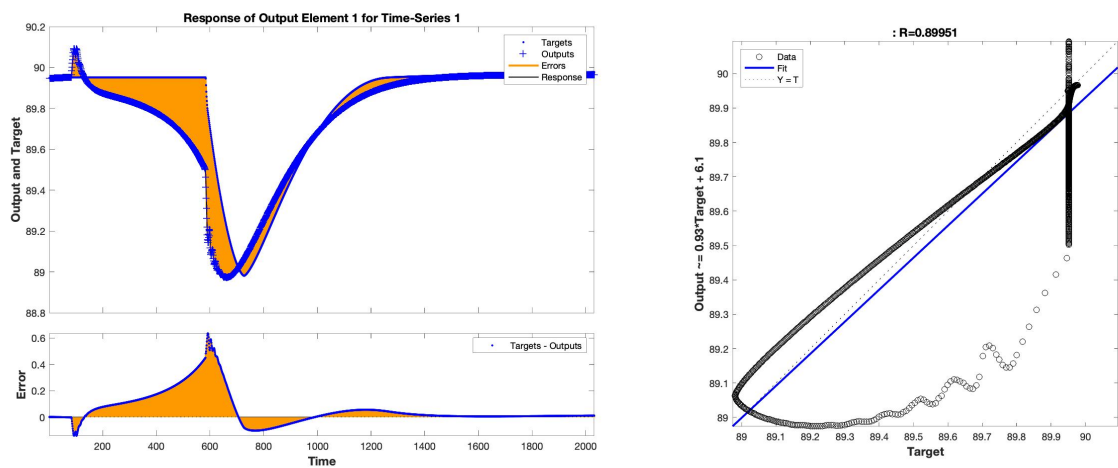


Figure 7.16 – Targets, outputs, error and response for estimation data & Target x estimation correlation (ER - VAL04) - time in [ds] and values in [psi].

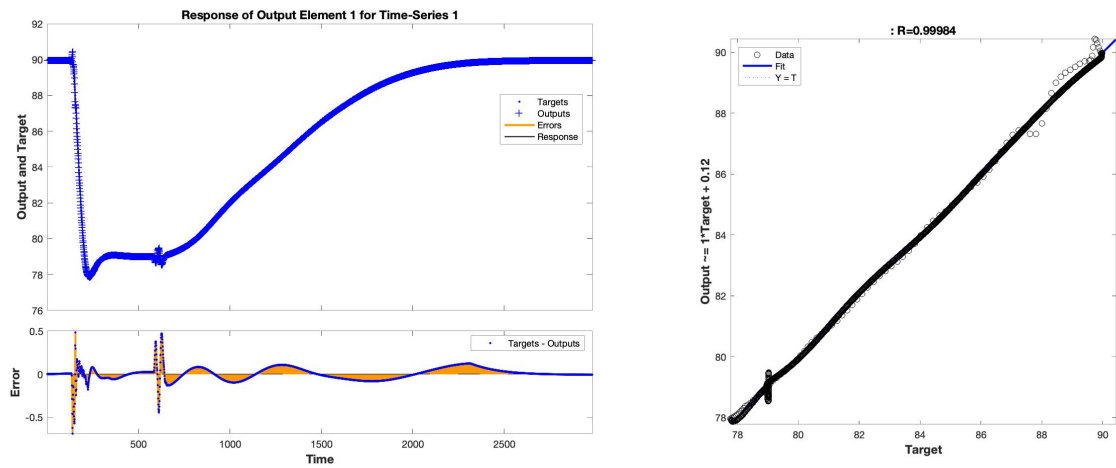


Figure 7.17 – Targets, outputs, error and response for estimation data & Target x estimation correlation (ER - VAL05) - time in [ds] and values in [psi].

7.2.3 BP +ER -> AR

Differently from the service application model, for the ANN with the emergency brake, the auxiliar reservoir will be a function of the brake pipe and the emergency reservoir. The ABDX valve uses the pressure stored on the latter to fill the brake pipe and the aux. reservoir and to allow for a faster reestablishment of the proper system pressures. Table 7.3 shows the models that had an error lower than 1 mse. The best model is B, with six delays on the input, 10 delays on the output and 20 hidden neurons. The results for this model are presented on figures 7.18 to 7.23

BP +ER ->AR	Case	A	B	C	D	E
input delays		4	6	8	8	8
output delays		4	10	10	6	8
hidden neurons		20	20	40	30	80
Error (mse)	Average	0.010	0.005	0.006	0.019	0.013
Error (mse) 1.00E-03	EST01	27.452	6.134	5.865	26.419	14.119
	VAL01	7.444	5.160	7.103	19.635	14.360
	VAL02	6.958	5.085	6.184	13.365	15.974
	VAL03	11.616	4.824	4.106	20.804	11.444
	VAL04	4.696	2.590	7.319	27.164	12.625
	VAL05	3.715	3.869	4.761	5.853	6.594
Run time [s]	Average	0.080	0.163	0.174	0.163	0.168
	EST01	0.153	0.339	0.392	0.350	0.368
	VAL01	0.185	0.232	0.253	0.226	0.243
	VAL02	0.038	0.140	0.121	0.125	0.119
	VAL03	0.045	0.058	0.058	0.059	0.061
	VAL04	0.025	0.172	0.184	0.179	0.179
	VAL05	0.033	0.039	0.038	0.038	0.039
		+	+++	++	-	o

Table 7.3 – Results of the BP + ER -> RA model.

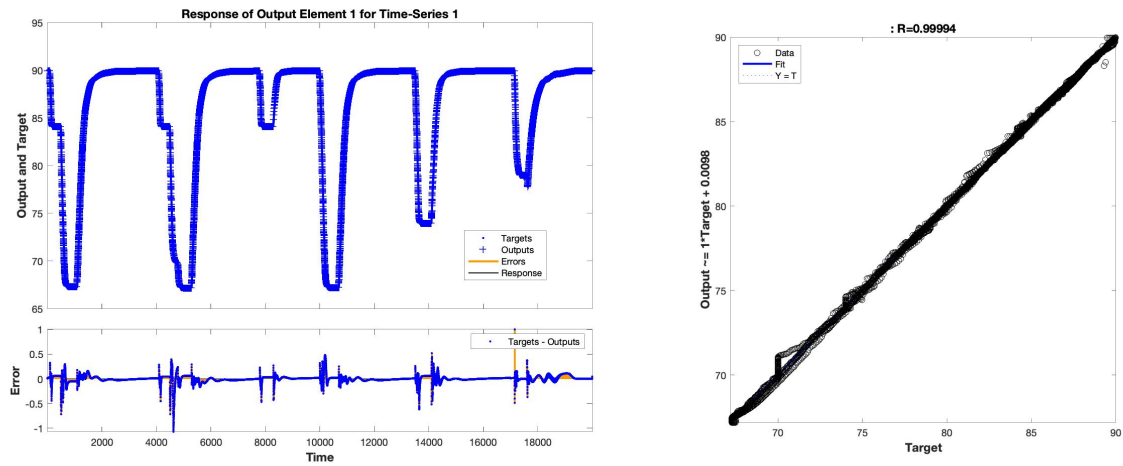


Figure 7.18 – Targets, outputs, error and response for estimation data & Target x estimation correlation (AR - EST01) - time in [ds] and values in [psi].

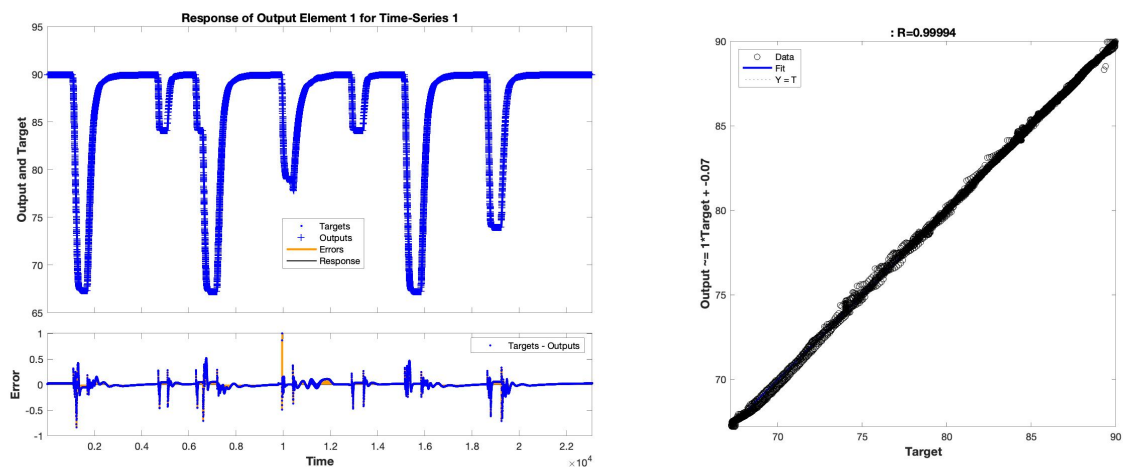


Figure 7.19 – Targets, outputs, error and response for estimation data & Target x estimation correlation (AR - VAL01) - time in [ds] and values in [psi].

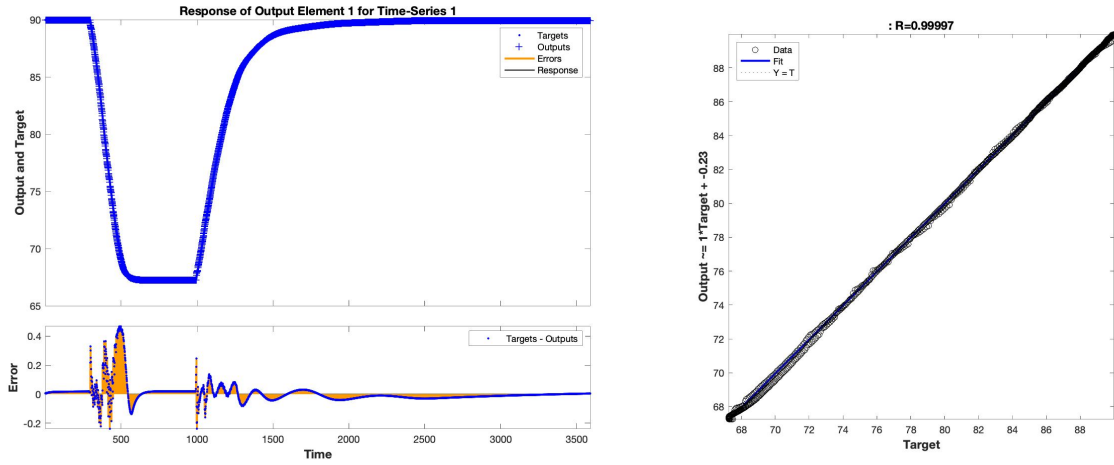


Figure 7.20 – Targets, outputs, error and response for estimation data & Target x estimation correlation (AR - VAL02) - time in [ds] and values in [psi].

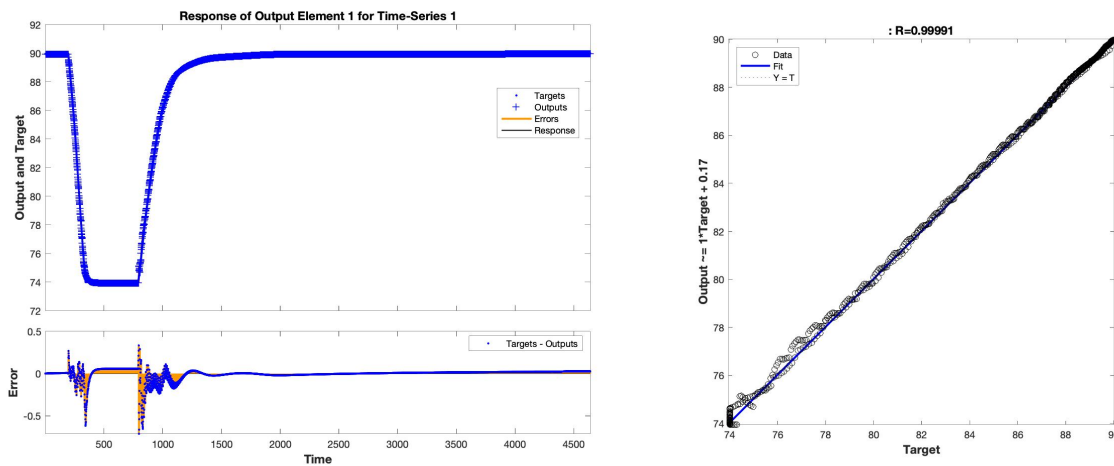


Figure 7.21 – Targets, outputs, error and response for estimation data & Target x estimation correlation (AR - VAL03) - time in [ds] and values in [psi].

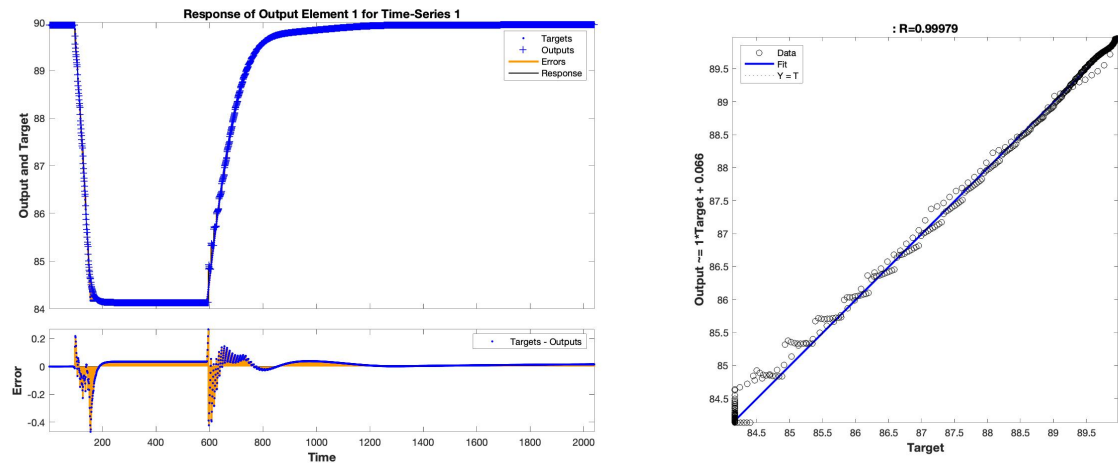


Figure 7.22 – Targets, outputs, error and response for estimation data & Target x estimation correlation (AR - VAL04) - time in [ds] and values in [psi].

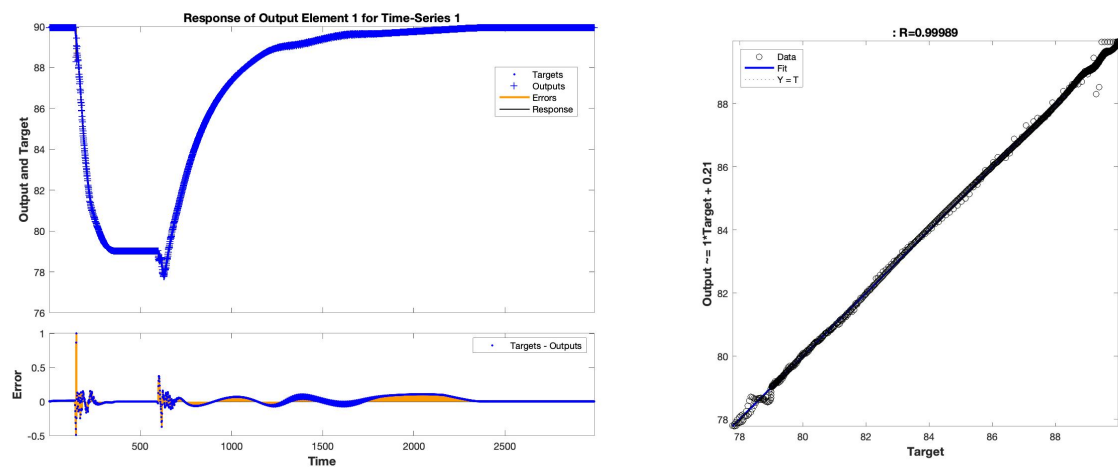


Figure 7.23 – Targets, outputs, error and response for estimation data & Target x estimation correlation (AR - VAL05) - time in [ds] and values in [psi].

7.2.4 BP + AR + ER -> BC

As for the last model, it is the one with most inputs on this whole work. The training times were comparable with the model for service application only. Model A presents the best adherence to the target, followed by model C. Both have four delays on the inputs and outputs. The running time is very similar to all models, E and G being the highest. Moving forward, model A will be used for the combined (service & emergency) brake simulator. Table 7.4 summarises the results.

BP+AR+ER ->BC	Case	A	B	C	D	E	F	G
input delays		4	4	4	4	5	6	6
output delays		4	4	4	4	5	6	6
hidden neurons		20	40	10	30	30	45	50
Error (mse)	Average	0.007	0.041	0.008	0.022	0.014	0.010	0.044
Error (mse) 1.00E-03	EST01	5.326	3.032	6.331	4.078	12.823	10.779	42.984
	VAL01	6.160	3.587	6.189	16.366	12.319	7.893	38.313
	VAL02	2.774	1.792	2.476	1.787	7.956	3.734	31.940
	VAL03	2.839	1.785	5.401	2.312	8.896	7.588	32.745
	VAL04	18.206	9.101	15.856	100.174	24.758	17.130	30.606
	VAL05	8.022	5.055	8.635	6.557	14.040	10.067	86.038
Run time [s]	Average	0.780	0.081	0.076	0.077	0.102	0.083	0.097
	EST01	0.183	0.174	0.162	0.163	0.218	0.169	0.188
	VAL01	0.160	0.179	0.165	0.174	0.229	0.185	0.226
	VAL02	0.033	0.034	0.034	0.036	0.047	0.038	0.047
	VAL03	0.040	0.046	0.039	0.039	0.055	0.050	0.056
	VAL04	0.023	0.024	0.023	0.023	0.027	0.024	0.029
	VAL05	0.030	0.031	0.030	0.028	0.033	0.034	0.039
		+++	--	++	-	0	+	--

Table 7.4 – Results of the BP + AR + ER -> BC model.

Figures 7.24 to 7.29 shows the model outputs and targets for model A. It has a very good performance and regression towards the targets.

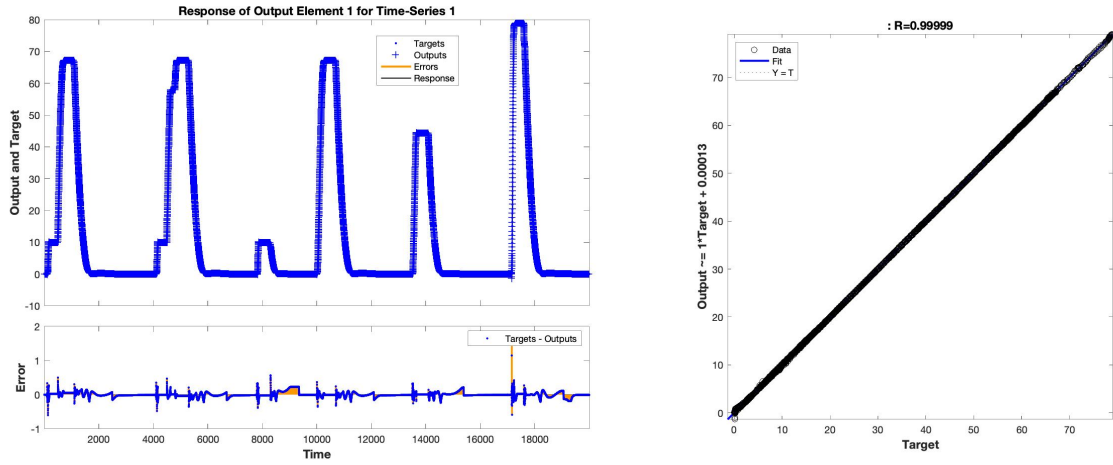


Figure 7.24 – Targets, outputs, error and response for estimation data & Target x estimation correlation (BC - EST01) - time in [ds] and values in [psi].

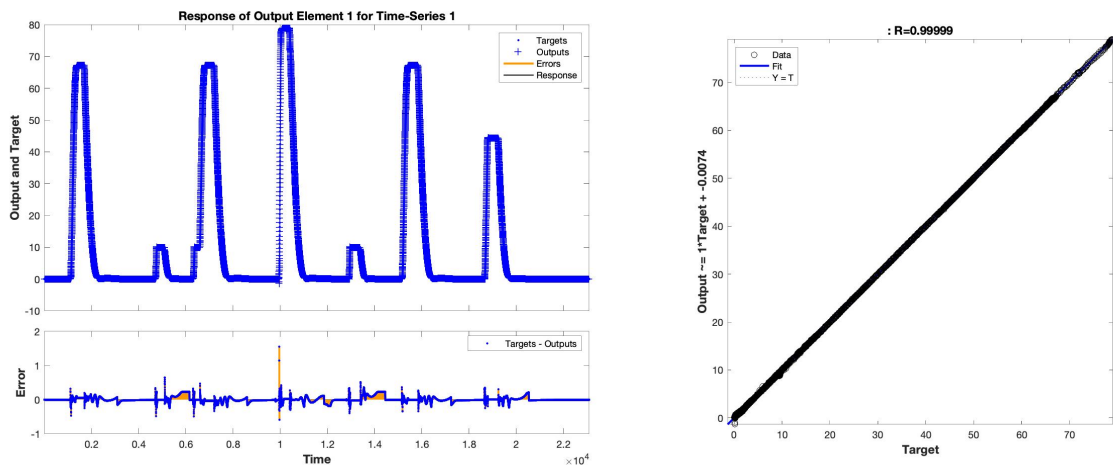


Figure 7.25 – Targets, outputs, error and response for estimation data & Target x estimation correlation (BC - VAL01) - time in [ds] and values in [psi].

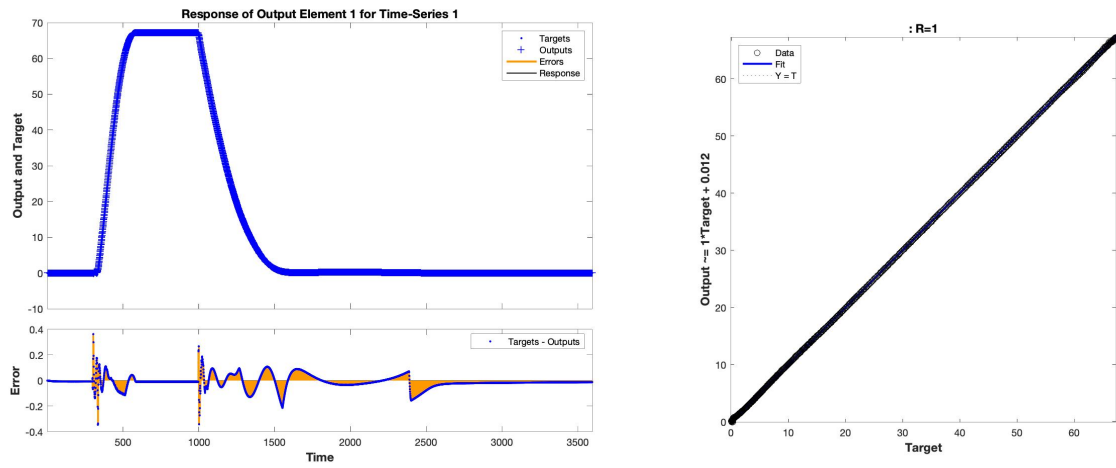


Figure 7.26 – Targets, outputs, error and response for estimation data & Target x estimation correlation (BC - VAL02) - time in [ds] and values in [psi].

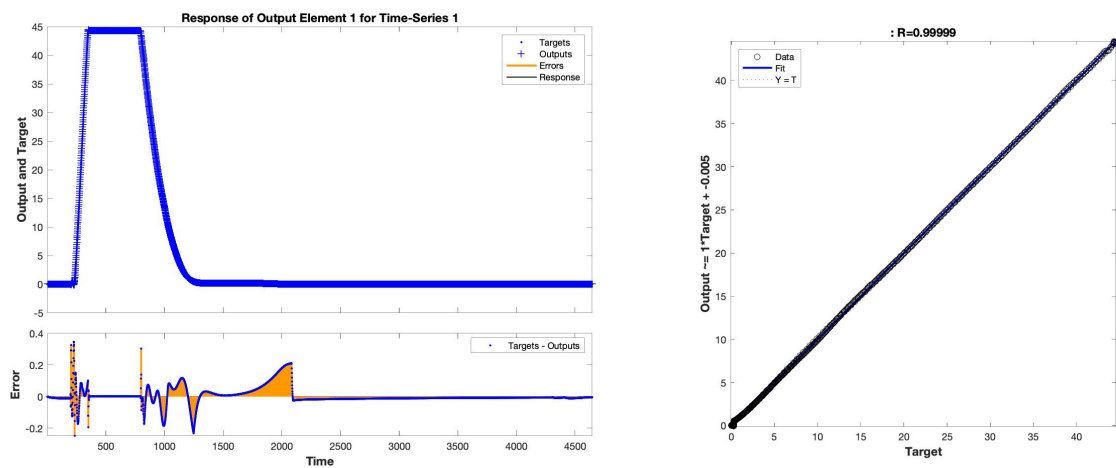


Figure 7.27 – Targets, outputs, error and response for estimation data & Target x estimation correlation (BC - VAL03) - time in [ds] and values in [psi].

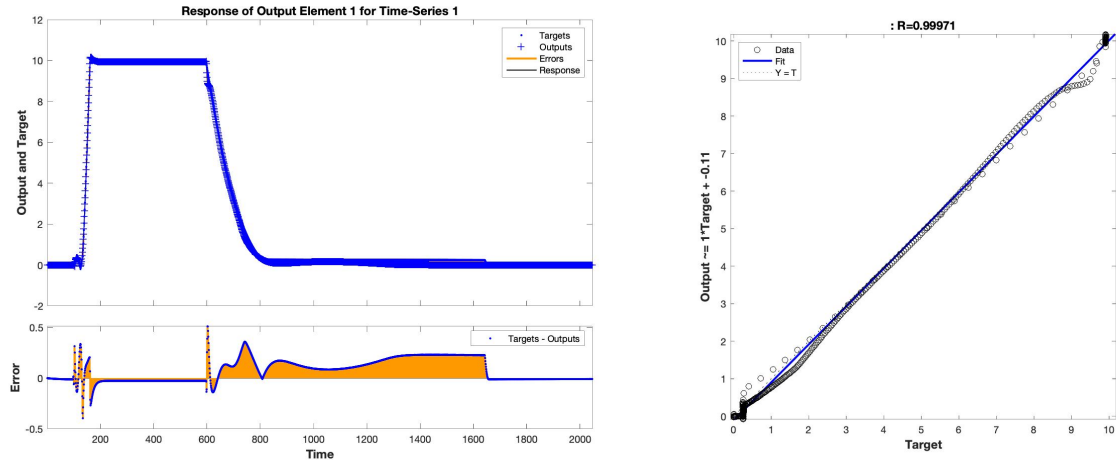


Figure 7.28 – Targets, outputs, error and response for estimation data & Target x estimation correlation (BC - VAL04) - time in [ds] and values in [psi].

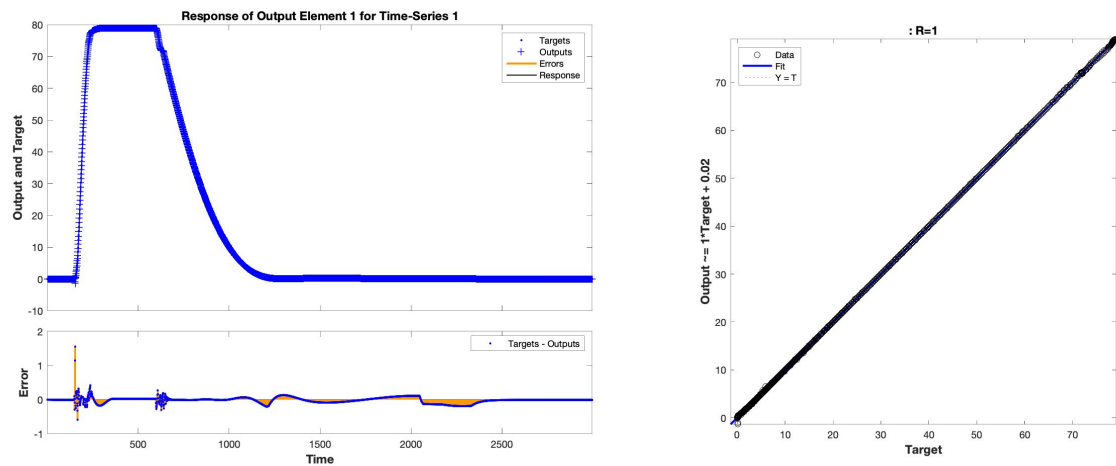


Figure 7.29 – Targets, outputs, error and response for estimation data & Target x estimation correlation (BC - VAL05) - time in [ds] and values in [psi].

7.3 Combined Brake Model

By combining the models presented previously on this chapter, one can concoct a service and emergency brake simulator that will predict the brake pipe pressure as well as the reservoir - aux. and emergency - and the brake cylinder. The first model is comprised of the most adherent sub-models using a structure that is similar to the one presented for the service brake simulator. Table 7.5 shows the results of the various cases. The combination of the best models will be called assembly 1 - or assy 1 for short. This combination presents instability on the ER model, with values of error quite high, in the order of 1 mse. Figure 4.10 shows the model output for VAL01 and the model and target comparison for the BC pressure. Note that both the ER and the BC pressures behave differently than expected.

The models for STP -> BP and BP -> ER work well and will not need to be changed. Assembly 2 (assy 2) changes the model for BP +ER -> AR from B to C and there is an improvement on both the adhesion of the aux. reservoir and the brake cylinder from model to target, with only the maximum service application being the main culprit of the error. On figure 7.31 the error on the maximum application is visible on the magenta brake cylinder response - that obscures the black aux. reservoir pressure signal.

Assembly 3 (assy 3) changes again the model BP + ER -> AR from C to D and the values become acceptable. The other cases change the brake cylinder model to slight improvements on the overall performance results. Figures 7.32 to 7.37 show the model results and the comparison to the brake cylinder. Although model D for BP +ER -> AR is not as precise as the others, it has shown more resilience. The run times between the models are within the expected statistical distribution without any outliers.

	Simulator	assy_1	assy_2	assy_3	assy_4	assy_5
Model	STP ->BP	C	C	C	C	C
	BP ->ER	D	D	D	D	D
	BP+ER->AR	B	C	D	D	D
	BP+AR+ER->BC	A	A	A	C	G
EST01 Error mse 1E-3	BP	17.605	17.605	17.605	17.605	17.605
	ER	14.924	14.924	14.924	14.924	14.924
	AR	4641.600	1545.000	33.597	33.597	33.597
	BC	72026.100	31067.000	395.692	369.633	388.704
	Run time [s]	1.779	1.852	1.797	1.828	1.722
VAL01 Error mse 1E-3	BP	51.493	51.493	51.493	51.493	51.493
	ER	13.878	13.878	13.878	13.878	13.878
	AR	4030.400	2418.000	24.463	24.463	24.463
	BC	84637.000	48862.000	273.564	261.283	264.024
	Run time [s]	1.556	1.507	1.602	1.567	1.483
VAL02 Error mse 1E-3	BP	20.010	20.010	20.010	20.010	20.010
	ER	20.097	20.097	20.097	20.097	20.097
	AR	85509.000	8558.700	27.536	27.536	27.536
	BC	106520.000	172050.000	247.663	252.807	284.909
	Run time [s]	0.332	0.333	0.318	0.324	0.323
VAL03 Error mse 1E-3	BP	6.493	6.493	6.493	6.493	6.493
	ER	9.201	9.201	9.201	9.201	9.201
	AR	47.733	6.326	25.302	25.302	25.302
	BC	396.809	77.361	297.839	278.346	254.543
	Run time [s]	0.421	0.389	0.381	0.383	0.367
VAL04 Error mse 1E-3	BP	25.506	25.506	25.506	25.506	25.506
	ER	5.293	5.293	5.293	5.293	5.293
	AR	8.887	7.413	29.622	29.622	29.622
	BC	74512.000	109.670	329.526	357.223	255.180
	Run time [s]	0.237	0.245	0.250	0.236	0.236
VAL05 Error mse 1E-3	BP	22.954	22.954	22.954	22.954	22.954
	ER	6.563	6.563	6.563	6.563	6.563
	AR	106.718	7.469	14.191	14.191	14.191
	BC	1561.500	158.001	416.107	323.211	385.033
	Run time [s]	0.290	0.301	0.287	0.330	0.285
		0	0	++	++	+++

Table 7.5 – Summary of results for the combined model with different sub-models.

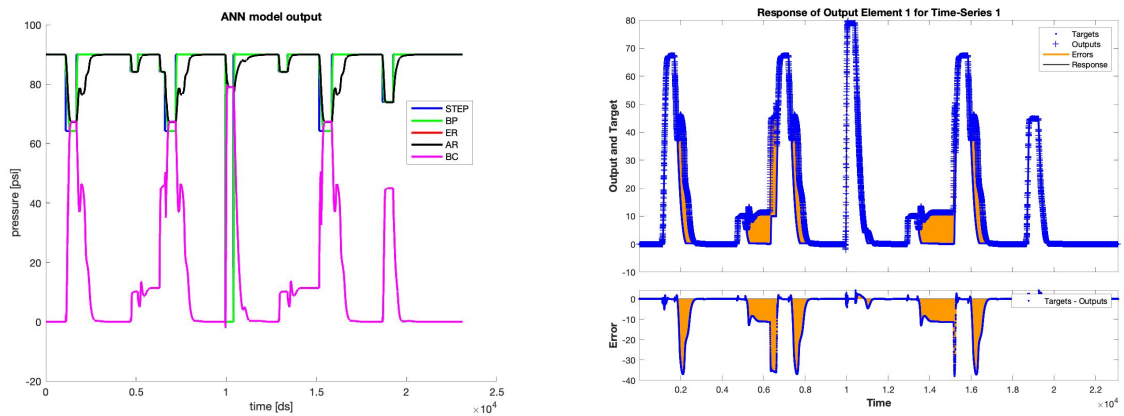


Figure 7.30 – Assembly 1 (assy 1) model results for VAL01 and comparison between model output and target for BC.

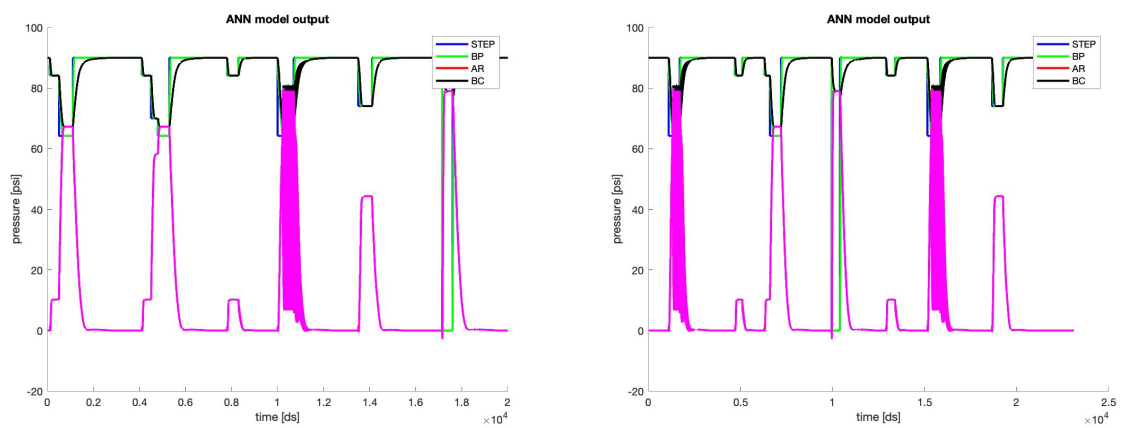


Figure 7.31 – Assembly 2 (assy 2) model results for EST01 and VAL01.

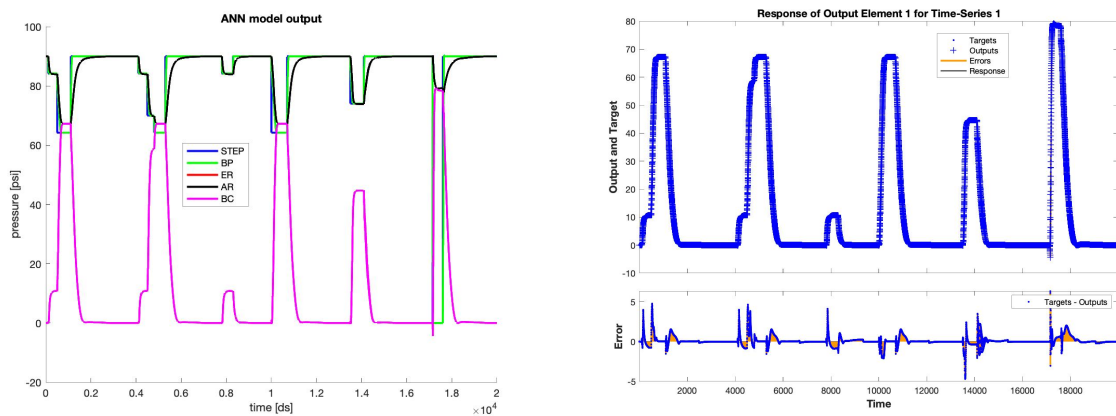


Figure 7.32 – Assembly 3 (assy 3) model results for EST01 and comparison between model output and target for BC.

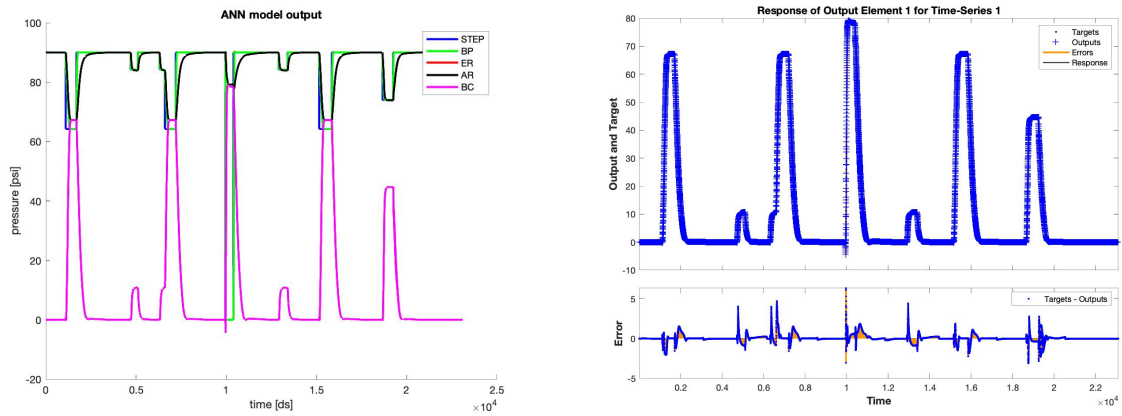


Figure 7.33 – Assembly 3 (assy 3) model results for VAL01 and comparison between model output and target for BC.

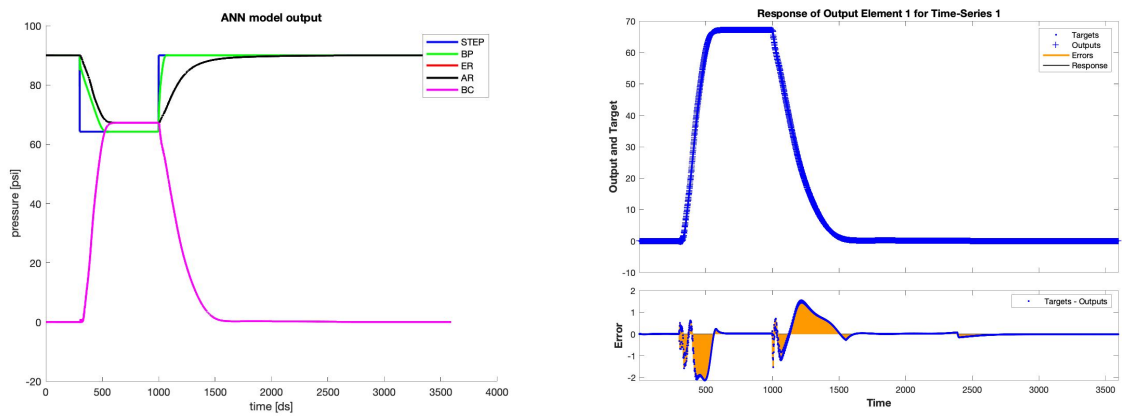


Figure 7.34 – Assembly 3 (assy 3) model results for VAL02 and comparison between model output and target for BC.

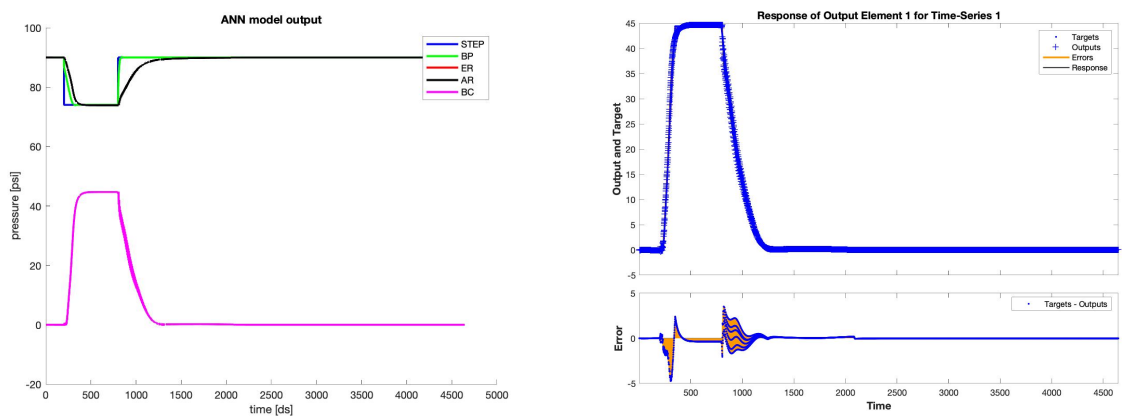


Figure 7.35 – Assembly 3 (assy 3) model results for VAL03 and comparison between model output and target for BC.

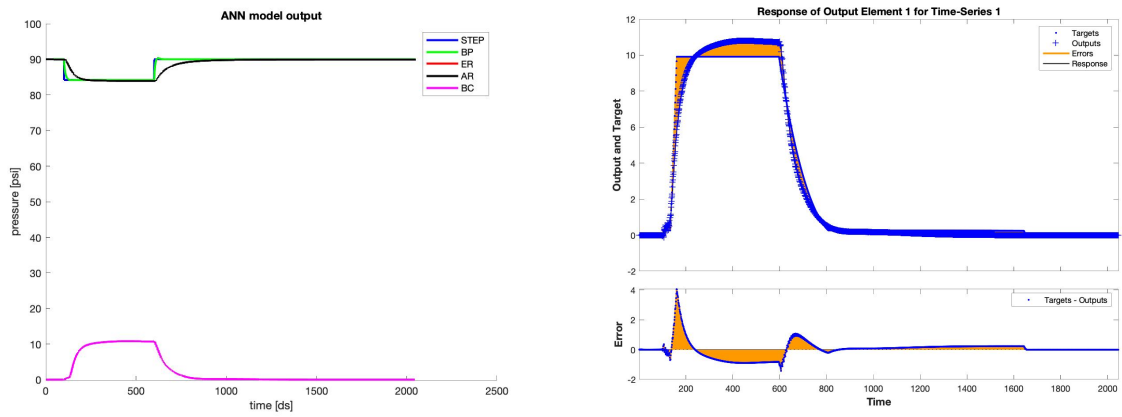


Figure 7.36 – Assembly 3 (assy 3) model results for VAL04 and comparison between model output and target for BC.

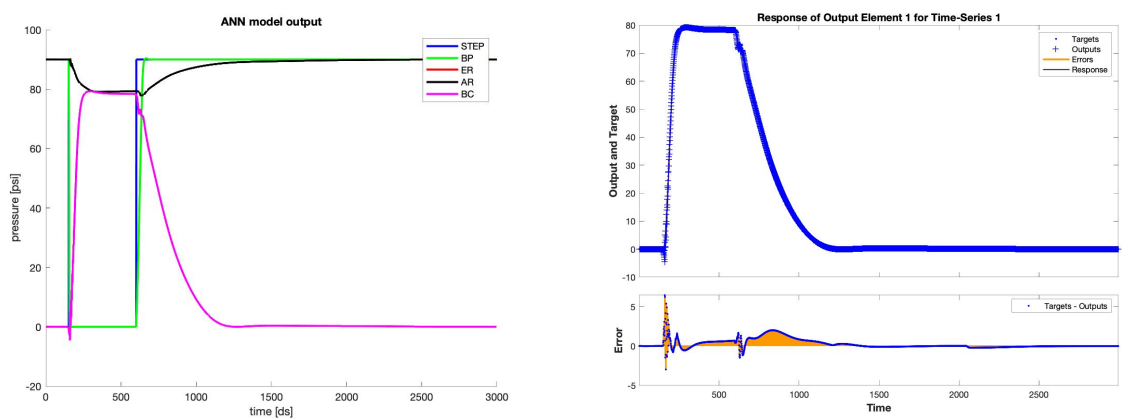


Figure 7.37 – Assembly 3 (assy 3) model results for VAL05 and comparison between model output and target for BC.

7.4 Alternative Combined Brake Model

The acceptance criteria for the model is twofold as mentioned earlier. It has to run faster than the current one and it has to keep an acceptable adherence to the targets

The best average mean square error for the combined model representing the brake cylinder pressure shown on the previous section was of the order of 0.250 psi. To assess what the effect of this error would be, one has to implement this model on the longitudinal dynamics simulator and compare the train behaviour, what falls outside the scope of this work.

Nonetheless, should the model not be sufficiently adherent, there is an alternative to assure that the error is minimised, train the artificial neural network input with the output of the previous models, rather than the baseline target data sets as inputs and keep the output target as the objective external dataset.

This experiment was done with the assembly 3 that will be called the baseline hitherto. The new BP + AR + ER → BC was trained using the EST01 data generated from the trained ANN and, as can be seen on table 7.6, there is an improvement on all cases, most of them improving an order (except VAL03 and VAL05 that show only minor improvements). Figures 7.38 to 7.43 show the model response and comparison to the brake cylinder targets.

There is no major run time difference between the baseline model assembly and the retrained.

If need be, further retraining using the previous model outputs may be used to increase model adherence to targets.

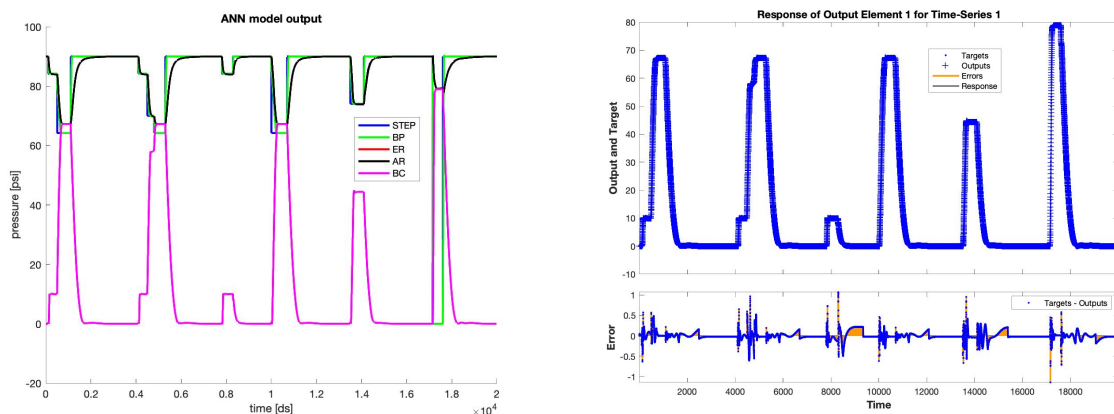


Figure 7.38 – Assembly 3 (assy 3) model results for EST01 and comparison between model output and target for BC.

	Simulator	Baseline	assy_3x
Model	STP ->BP	C	C
	BP ->ER	D	D
	BP+ER->AR	D	D
	BP+AR+ER->BC	A	X1
EST01 Error mse 1E-3	BP	17.605	17.605
	ER	14.924	14.924
	AR	33.597	33.597
	BC	395.692	11.386
	Run time [s]	1.797	1.783
VAL01 Error mse 1E-3	BP	51.493	51.493
	ER	13.878	13.878
	AR	24.463	24.463
	BC	273.564	38.889
	Run time [s]	1.602	1.495
VAL02 Error mse 1E-3	BP	20.010	20.010
	ER	20.097	20.097
	AR	27.536	27.536
	BC	247.663	36.935
	Run time [s]	0.318	0.351
VAL03 Error mse 1E-3	BP	6.493	6.493
	ER	9.201	9.201
	AR	25.302	25.302
	BC	297.839	243.112
	Run time [s]	0.381	0.232
VAL04 Error mse 1E-3	BP	25.506	25.506
	ER	5.293	5.293
	AR	29.622	29.622
	BC	329.526	31.022
	Run time [s]	0.250	0.279
VAL05 Error mse 1E-3	BP	22.954	22.954
	ER	6.563	6.563
	AR	14.191	14.191
	BC	416.107	194.149
	Run time [s]	0.287	0.293

Table 7.6 – Summary of results for the combined model with different sub-models with retraining.

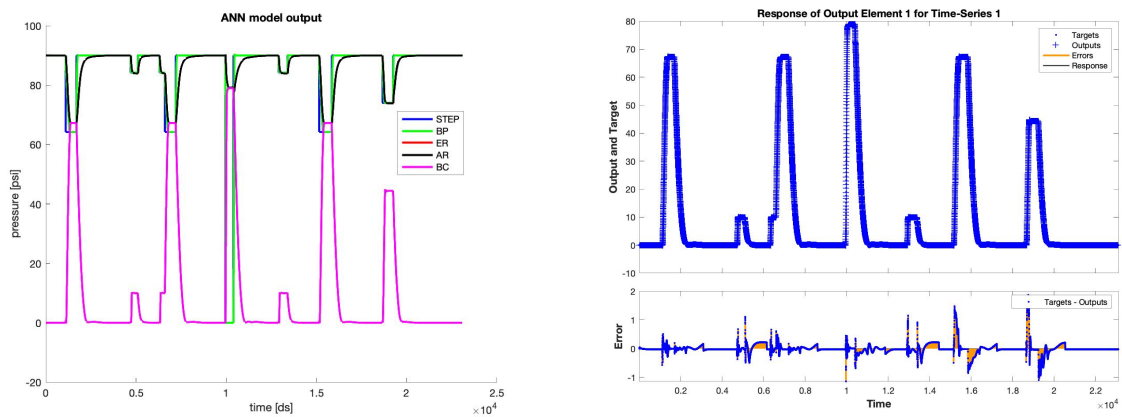


Figure 7.39 – Assembly 3 (assy 3) model results for VAL01 and comparison between model output and target for BC.

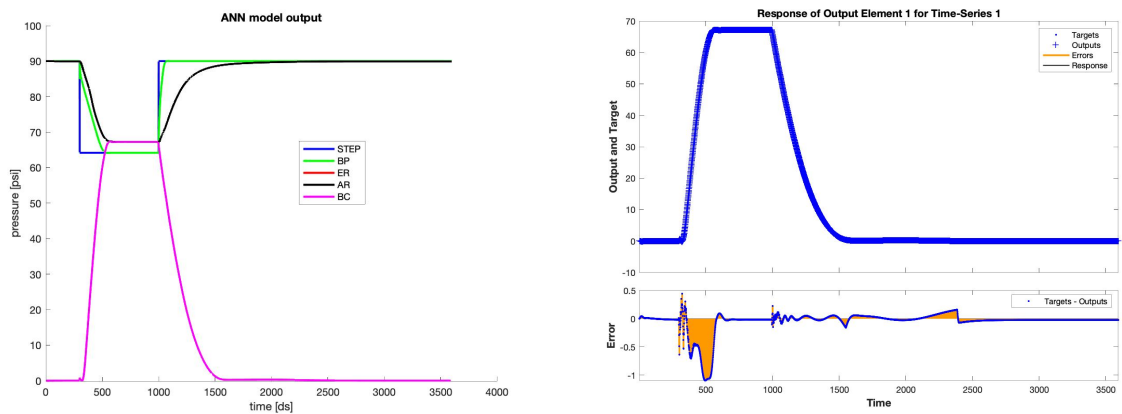


Figure 7.40 – Assembly 3 (assy 3) model results for VAL02 and comparison between model output and target for BC.

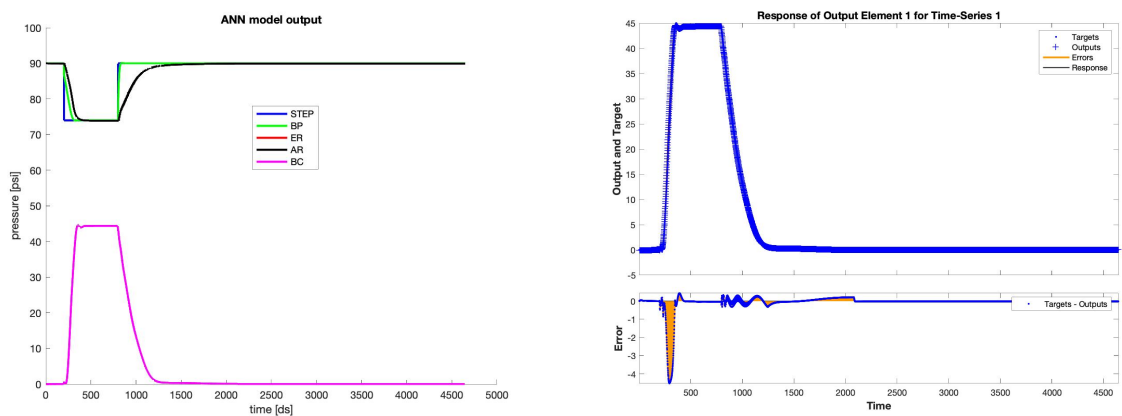


Figure 7.41 – Assembly 3 (assy 3) model results for VAL03 and comparison between model output and target for BC.

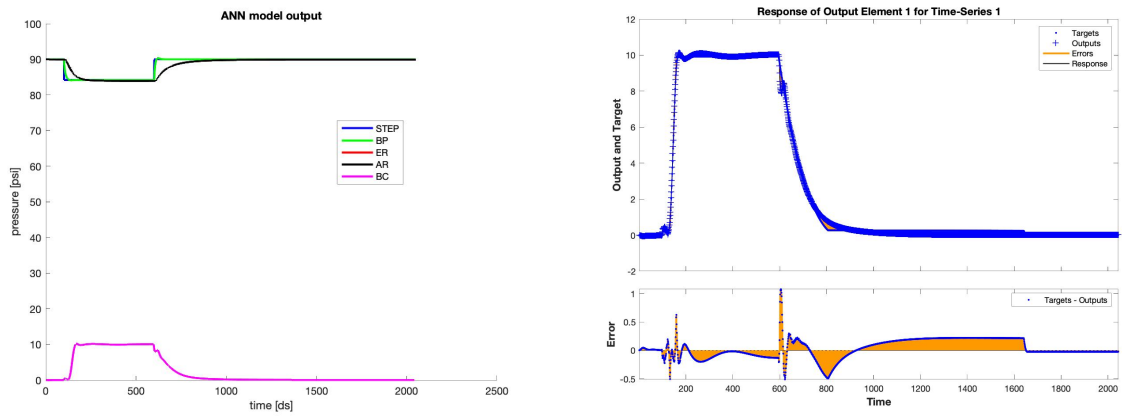


Figure 7.42 – Assembly 3 (assy 3) model results for VAL04 and comparison between model output and target for BC.

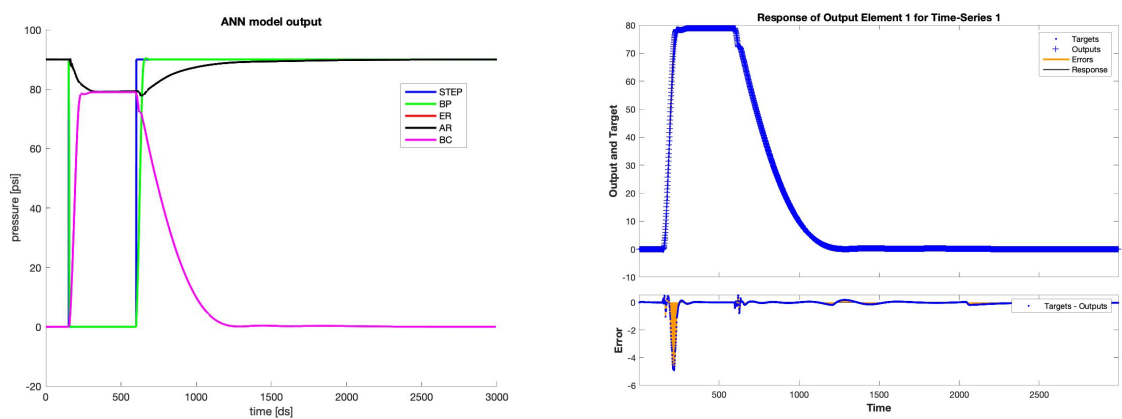


Figure 7.43 – Assembly 3 (assy 3) model results for VAL05 and comparison between model output and target for BC.

7.5 Combined and Service Model Run Time Comparison

The same methods to assess the run time were used. The combined model (service and emergency brake) results are presented on 7.7. For the estimation and VAL01 cases, the combined model takes longer for it has to calculate an additional model. On the individual applications, the combined models were more efficient and had a reduced run time when compared with the previous models. Regardless, both models run faster than the original model currently implemented on the simulator and transcribed on appendix 3.

Case	Run time interval [s]					
	Original model		Service Brake model		Combined model	
EST01	19.97	22.30	1.37	1.48	1.70	1.88
VAL01	19.41	22.46	1.40	1.49	1.44	1.63
VAL02	3.21	5.36	0.76	0.86	0.31	0.35
VAL03	4.67	5.60	0.77	0.88	0.23	0.49
VAL04	2.23	2.62	0.66	0.79	0.21	0.28
VAL05					0.26	0.33

Table 7.7 – Summary times for the various models presented on this work

8 CONCLUSION

We start the conclusion by going back to the stated objective: "Explore the possibility of creating a mathematical model to replace the current model by Ribeiro (Ribeiro, 2017) and Teodoro (Teodoro, 2016) evaluating the accuracy and running time. To this, the work can be considered successful. The mean square error between the the target and the service brake model was within $10E-2$ and for the combined emergency and brake models, the error was within 0.1 psi. The effect of this error has to be evaluated on the longitudinal dynamics simulator, a possible future work. It is believed that the effect will be minimum, if any. In the unlikely case that the error cause an adverse effect that hinder the usability of the longitudinal dynamics simulator further tweaking can be done on the ANN model to improve the accuracy, as demonstrated on the end of the last chapter.

Another difference between the original model and the proposed model is that the code is much smaller and leaner than the original which is transcribed on appendix 3.

Alternatively, at the time of the writing of this work, there are efforts to test actual automatic brake valves and acquire their practical responses. Once these are available, the artificial neural networks can be trained with the new data and should perform accordingly. The training process is quite straightforward and efficient.

For the automatic brake simulation, linear models have not been successful. Also, for this specific problem, the neural networks that were trained open and closed later on have not yielded good results also. Only the ANNs that were trained closed were able to solve the problem accordingly.

The final networks presented were handcrafted and took time and experience to find a good combination of parameters.

This work demonstrated that the problem of the automatic brake system pressure simulation can be solved with a reasonably small training dataset with a manageable size model that does not require large server infrastructure. On the future, practical data should be used to build a stochastic model that takes into consideration the actual variations of a physical valve.

The emergency brake and reservoir were added to the model, showing its flexibility and the worthiness of this approach.

Other features should be added to the model in future works, such as the effects of

multiple vehicles on the composition. Also, the simulation model can be made modular to show the effects on the consist separately, increasing flexibility.

In the future, other parts of the longitudinal dynamics simulator, as well as a coupled lateral dynamics simulator can be done using machine learning techniques in general and artificial neural networks, in particular to improve speed of the simulations, keeping a reasonable accuracy. Also, as has been indicated on the bibliography research, machine learning has been applied with success to several railway applications such as infrastructure monitoring, circulation control, etc.

Having addressed the objective and demonstrated that it is possible to create a model to replace the current automatic brake valve simulator that runs faster with reasonable accuracy and also demonstrated the flexibility of the techniques employed, this work is concluded.

BIBLIOGRAPHY

ABIFER - Associação Brasileira da Indústria Ferroviária.

Paiva Abreu, M. de. British business in brazil: maturity and demise (1850-1950). **Revista Brasileira de Economia**, v. 54, n. 4, dec 2000.

ANTF - Associação nacional dos Transportadores Ferroviários consultada em 30 de junho de 2020.

ANTT - Agencia Nacional de Transportes Terrestres consultada em 30 de junho de 2020.

BabyAnitha, E.; Duraiswamy, K. An efficient optimization based vehicle movement prediction with aid of feed forward back propagation neural network. **International Journal of Computer Applications**, v. 91, n. 12, p. 24–31, apr 2014.

Barbosa, B. H. G.; Aguirre, L. A.; Martinez, C. B.; Braga, A. P. Black and gray-box identification of a hydraulic pumping system. **IEEE Transactions on Control Systems Technology**, v. 19, n. 2, p. 398–406, mar 2011.

Carvalho, T. P.; Soares, F. A. A. M. N.; Vita, R.; P. Francisco, R. da; Basto, J. P.; Alcalá, S. G. S. A systematic literature review of machine learning methods applied to predictive maintenance. **Computers & Industrial Engineering**, v. 137, p. 106024, nov 2019.

City and Town Population Totals: 2010-2019. United States Census Bureau, Population Division. Retrieved May 21, 2020.

Consilvio, A.; Solís-Hernández, J.; Jiménez-Redondo, N.; Sanetti, P.; Papa, F.; Mingolarra-Garaizar, I. On applying machine learning and simulative approaches to railway asset management: The earthworks and track circuits case studies. **Sustainability**, v. 12, n. 6, p. 2544, mar 2020.

Coraddu, A.; Oneto, L.; Ghio, A.; Savio, S.; Figari, M.; Anguita, D. Machine learning for wear forecasting of naval assets for condition-based maintenance applications. In: **2015 International Conference on Electrical Systems for Aircraft, Railway, Ship Propulsion and Road Vehicles (ESARS)**. [S.l.]: IEEE, 2015.

Corrêa, L. M. **Válvula de controle - Modelo AB**. [S.l.]: VALER - Educação Vale.

Corrêa, L. M. **Válvula de controle - Modelo ABD**. [S.l.]: VALER - Educação Vale.

Corrêa, L. M. **Válvula de controle - Modelo ABDX**. [S.l.]: VALER - Educação Vale.

Crozet, A. B. Y. Efficiency indicators of railways in france. In: **International Transport Forum**. [S.l.: s.n.], 2014.

Die Finanzierung der Eisenbahn des Bundes, August 2015 - Deutsche Bahn AG.

Drmac, Z.; Gugercin, S.; Beattie, C. Quadrature-based vector fitting for discretized approximation. **Journal on Scientific Computing**, v. 37, n. 2, p. A625–A652, 2015.

- Foresee, F. D.; Hagan, M. T. Gauss-newton approximation to bayesian learning. In: **Proceedings of International Conference on Neural Networks ICNN 97**. [S.l.: s.n.], 1997. v. 3, p. 1930–1935 vol.3.
- Forssell, U.; Ljung, L. Closed-loop identification revisited. **Automatica**, v. 35, n. 7, p. 1215–1241, jul 1999.
- Georgiou, I. Mapping railway development prospects in brazil. **Transport Reviews**, v. 29, n. 6, p. 685–714, nov 2009.
- Gerum, P. C. L.; Altay, A.; Baykal-Gürsoy, M. Data-driven predictive maintenance scheduling policies for railways. **Transportation Research Part C: Emerging Technologies**, v. 107, p. 137–154, oct 2019.
- Gibert, X.; Patel, V. M.; Chellappa, R. Deep multitask learning for railway track inspection. **IEEE Transactions on Intelligent Transportation Systems**, v. 18, n. 1, p. 153–164, jan 2017.
- Histórico das ferrovias no Brasil - DNIT- Departamento Nacional de Infraestrutura Terrestre - Governo Federal do Brasil.
- Hobsbawm, E. J. **The age of revolution 1789-1848**. [S.l.]: New York: Vintage Books., 1996.
- Jr., J. P. M.; Barreto, G. A new look at nonlinear time series prediction with NARX recurrent neural network. In: **2006 Ninth Brazilian Symposium on Neural Networks (SBRN)**. [S.l.]: IEEE, 2006.
- Krummenacher, G.; Ong, C. S.; Koller, S.; Kobayashi, S.; Buhmann, J. M. Wheel defect detection with machine learning. **IEEE Transactions on Intelligent Transportation Systems**, v. 19, n. 4, p. 1176–1187, apr 2018.
- Lasisi, A.; Attoh-Okine, N. Principal components analysis and track quality index: A machine learning approach. **Transportation Research Part C: Emerging Technologies**, v. 91, p. 230–248, jun 2018.
- Lasisi, A.; Attoh-Okine, N. An unsupervised learning framework for track quality index and safety. **Transportation Infrastructure Geotechnology**, v. 7, n. 1, p. 1–12, jul 2019.
- Li, H.; Parikh, D.; He, Q.; Qian, B.; Li, Z.; Fang, D.; Hampapur, A. Improving rail network velocity: A machine learning approach to predictive maintenance. **Transportation Research Part C: Emerging Technologies**, v. 45, p. 17–26, aug 2014.
- Liu, J.; Huang, Y.; Zou, Q.; Tian, M.; Wang, S.; Zhao, X.; Dai, P.; Ren, S. Learning visual similarity for inspecting defective railway fasteners. **IEEE Sensors Journal**, v. 19, n. 16, p. 6844–6857, aug 2019.
- Ljung, L. **System Identification Theory for the User**. [S.l.]: PTR Prentice Hall, 1987.
- Ljung, L. Experiments with identification of continuous time models. **IFAC Proceedings Volumes**, v. 42, n. 10, p. 1175–1180, 2009.
- Ljung, L.; Soderstrom, T. **Theory and Practice of Recursive Identification (Signal Processing, Optimization, and Control)**. [S.l.]: The MIT Press, 1983. ISBN 0-262-12095-X.

- Ma, C.; Zuo, S. Black-box method of identification and diagnosis of abnormal noise sources of permanent magnet synchronous machines for electric vehicles. **IEEE Transactions on Industrial Electronics**, v. 61, n. 10, p. 5538–5549, oct 2014.
- MacKay, D. J. C. Bayesian interpolation. **Neural Computation**, v. 4, n. 3, p. 415–447, may 1992.
- Martin, L. J. W. Predictive reasoning and machine learning for the enhancement of reliability in railway systems. In: **Reliability, Safety, and Security of Railway Systems. Modelling, Analysis, Verification, and Certification**. [S.l.]: Springer International Publishing, 2016. p. 178–188.
- Mdic - Ministério do Desenvolvimento, Indústria e Comércio Exterior - Governo Federal - Brasil.
- NYAB (Ed.). **The 26-L BRake Equipment - Instruction Pamphlet No. 74**. [S.l.]: New York Air Brake Corporation, 1964.
- Prasongpongchai, T.; Chalidabhongse, T. H.; Leelhapanu, S. A vision-based method for the detection of missing rail fasteners. In: **2017 IEEE International Conference on Signal and Image Processing Applications (ICSIPA)**. [S.l.]: IEEE, 2017.
- Prokhorchenko, A.; Panchenko, A.; Parkhomenko, L.; Nesterenko, H.; Muzykin, M.; Prokhorchenko, H.; Kolisnyk, A. Forecasting the estimated time of arrival for a cargo dispatch delivered by a freight train along a railway section. **Eastern-European Journal of Enterprise Technologies**, v. 3, n. 3 (99), p. 30–38, jun 2019.
- Pugi, L.; Malvezzi, M.; Papini, S.; Tesi, S. Simulation of braking performance: The ansaldobreda emu v250 application. **Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit**, v. 229, n. 2, p. 160–172, 2015.
- Ribeiro, D. F. **Desenvolvimento de Modelos para Simulação em Tempo Real da Frenagem de Composições Ferroviárias de Carga**. Dissertação (Mestrado) — FEM UNICAMP, 2017.
- Sanchiz, J. M. C. Energy and railway workshops: An archaeology of the fepasa complex in jundiai, brazil. **The Journal of Transport History**, v. 39, n. 2, p. 213–235, may 2018.
- Scholz, S. An innovative system to detect obstacles on rail tracks. In: **Automated People Movers and Automated Transit Systems 2016**. [S.l.]: American Society of Civil Engineers, 2016.
- Sysyn, M.; Gerber, U.; Kluge, F.; Nabochenko, O.; Kovalchuk, V. Turnout remaining useful life prognosis by means of on-board inertial measurements on operational trains. **International Journal of Rail Transportation**, v. 8, n. 4, p. 347–369, nov 2019.
- Taheri, M.; Ahmadian, M. Machine learning from computer simulations with applications in rail vehicle dynamics. **Vehicle System Dynamics**, v. 54, n. 5, p. 653–666, mar 2016.
- Taheri, N.; Nejad, F. M.; Zakeri, H. A brief overview and new knowledge based system for rail direct fastening evaluation using digital image processing. **Archives of Computational Methods in Engineering**, v. 27, n. 3, p. 691–709, fev. 2019.
- Teodoro, I. P. **Estudo do comportamento do sistema de freio pneumático de trens de carga utilizando o método de volumes finitos**. Dissertação (Mestrado) — FEM UNICAMP, 2016.

Teodoro, Í. P.; Eckert, J. J.; Lopes, P. F.; Martins, T. S.; Santos, A. A. Parallel simulation of railway pneumatic brake using openMP. **International Journal of Rail Transportation**, v. 8, n. 2, p. 180–194, aug 2019.

Tsunashima, H. Condition monitoring of railway tracks from car-body vibration using a machine learning technique. **Applied Sciences**, v. 9, n. 13, p. 2734, jul 2019.

US National Archives catalog item 594940; Department of Agriculture. Office of the Secretary. Office of Information. 1925-ca. 1981 (Most Recent).

Wen, C.; Huang, P.; Li, Z.; Lessan, J.; Fu, L.; Jiang, C.; Xu, X. Train dispatching management with data- driven approaches: A comprehensive review and appraisal. **IEEE Access**, v. 7, p. 114547–114571, 2019.

Wen, C.; Mou, W.; Huang, P.; Li, Z. A predictive model of train delays on a railway line. **Journal of Forecasting**, v. 39, n. 3, p. 470–488, dec 2019.

YOUNG, P.; JAKEMAN, A. Refined instrumental variable methods of recursive time-series analysis part III. extensions. **International Journal of Control**, v. 31, n. 4, p. 741–764, apr 1980.

Appendix

APPENDIX A – INTRODUCTION TO THE HISTORY OF RAILWAYS

Railways play an important role in a country's development. The combination of the type of cargo, region topology and local solutions are a very powerful indicator of history, degree of development and society interests and decisions. When railways are mentioned, some examples come to mind, such as the USA and Europe, as well as Japan and China. These countries have taken very different decisions regarding to their choices and their current rail infrastructure - both rolling stock and permanent way. This have lead heavier and slower networks and to lighter and faster. For example, the European network system favours passenger traffic, i.e. light and fast, whereas the North American system favours heavy haul with slower and heavier trains

The modern rail history is credited to Stephenson and Son in England, 1825. Steam engines and rail guided vehicles vehicles already existed, their contribution and reason for fame is the creation of a system for intercity people transport system. This has shrink the world and was one of the pillars of the industrial revolution. The rail system has allowed factories to access raw materials over land in high quantities and little time, compared with the previous transportation method - horse drawn carriages. The system also allowed the spread of finished goods and most importantly, people and information. Before, most people spent their life in their county and most journeys were as far as walking distance. For the wealthy and well to do, that could afford travel, it was a dangerous and complicated matter, most likely by horse or carriage and prone to assaults and bandits. The railway changed this reality, for it allowed for long distance travel and communication to happen within a day.

The railway was and still is an important tool in war, revolution and country integration. The continental U.S.A could only be controlled and pacified by a central government with the completion of the transcontinental railway (figure A.1) in 1869 - after que civil war. The Franco-Prussian war of 1870, that ended the french second empire, exiled Napoleon III, consolidated the German Reich and laid the foundations of the contemporary Europe was won by the Prussian superior capacity of mobilisation allowed by the clever use of railways. On the other hand, the Great War of 1914 entered a stalemate that costed millions of lives by the incapacity of the Schliffen Plan to cope with the train delays and schedule changes due to the

failure in commandeering the Belgian rail infrastructure in due time and thus deploy the German troops ahead of France's and England's. Such importance the railway has that the armistice of 1918 was signed on a passenger car in Compiègne and the same passenger car was used to sign France's surrender in 1941



Figure A.1 – Encounter at Promotory, or the Golden Spike Ceremony - the completion of the transcontinental railway (US...,)

With all of this, it is not surprising that railways are darlings of governments and populations as harbingers of progress and development and are desired by the public since the XIX century.

The European railways were built on the XIX century tracks and on the middle of the XX century were slowly converted to high speed rail. The topography of Europe west of the Alps is mostly flat, what helps greatly on the rail transportation. Also, the excess resources from reconstruction and from the European economies have allowed for great improvements on the rail networks. The largest companies are state owned (SNCF - France; Deutsch Bahn - Germany; Trenitalia - Italy, etc.) and the network is highly subsidised. For example, the German railway received 17 billion euros in 2014 (Die...,), the French, 13.2 billion euros (Crozet, 2014), quite a substantial sum and the tariffs are most time on par with air tickets, with the advantage of departing from city center and arriving at city center. Another peculiarity of the European set up is the distance between the major capitals and population centres, that tend to be within 1000 km. The European railways transport passengers in their majority with electrical locomotives and passenger cars

The United States experience tend to be different. There are only 10 cities in the USA with more than 1 million people (City...,) and the topography does not favor high speed

rail. Also, on the United States, the major expansions happened after the invention of the automobile and the second large expansion after airplanes were developed and viable. So, passenger use of the rail declined with the advent of the aforementioned technologies. The USA, on the other hand, excels at heavy haul for the transportation of goods - finished and raw. The locomotives tend to run on Diesel instead of electricity and the current ownership of the tracks by the heavy haul companies and the lack of subsidies to the passenger transport seem to be the major forces preventing the expansion of the passenger service.

The Chinese case is quite an interesting one. The expansion of the passenger rail networks are quite recent fuelled by the financial surplus generated by the transferring of manufacturing to that country. Though coal, grain and ore are the majority of the cargo transported, the high speed system is focused on passengers. China has the world largest population, concentrated on the eastern most part of the country. In China, the railways are also political tools for country integration and control, specially in Xinjiang and Tibet, where there is a push for population relocation and military material transportation.

There are other cases of railway use - transport of large quantities of raw materials for export - especially in Brazil, Australia, Chile, Mozambique, etc. In these cases, passenger transport is an afterthought. The network connects not population centres but the mine - or the farm - to the port in a clear raw material export driven economy

A.1 The Brazilian Case

The next sections outline the railway history in Brazil in chronological order and will analyse the motivations and desired objectives as well as the background for the events and their consequences.

A.1.1 The Empire

The incentives to build a railway in Brazil started during the short first empire in 1828, to no avail. An Anglo-Brazilian company was created in 1832 with the intent of connecting the city of Porto Feliz to Santos, in order to reduce the logistical costs of export products. When the plans were presented, they were considered ahead of their time for the technology was not yet ready to transpose Serra do Mar - a 700 m steep hill between Brazilian coast and its highlands. In the meantime, there was a government change and the political backing was no longer present. In 1835, the regent Diogo Feijó issue imperial law 101, that would allow

whoever was capable to connect the Brazilian capital of Rio de Janeiro to the provincial capital of Minas Gerais, Rio Grande do Sul and Bahia to explore the service for 40 years. However, the cost of such endeavour was high compared to the projected gains and the project again was not implemented.

A new law goes into effect in 1852, bill 641, that increased the exploration right to 90 years, improved financing and it prevented railways to be installed within 33 km of each other (what would hamper the network later that century on the path to Santos). The bill limited the profits to 8%, still not attractive enough and no railway was build.

Finally, in 1854, Barão de Mauá builds EFM- Estrada de Ferro Mauá - on Rio de Janeiro, between his port (Praça XV) and Serra do Mar. The track and pinion infrastructure to overcome the hill was never constructed. It had 3 locomotives with Indian Gauge tracks (5'6"/1.676 m) and lasted until 1888, when it was absorbed by other systems.

In 1859, construction began on Recife & São Francisco Railway Company between Recife and Vila do Cabo - a 31.5 km distance with broad gauge (1.6 m). In 1901, the gauge was reduced to 1m.

On late 1859, work began on perhaps the most important railway of Brasil, Estrada de Ferro D. Pedro II - later called Estrada de Ferro Central do Brasil (EFCB). In the hight of its importance, it connected the provincial capitals of São Paulo, Rio de Janeiro and Minas Gerais. In 1877 EFCB connected with EFSP - Estrada de Ferro São Paulo - effectively linking the imperial capital to the financial capital of the Empire. This connection was particularly important on the machinations of São Paulo oligarchy to rule over the country

The city of Belo Horizonte, the new capital of the province of Minas Gerais, a planed city, was built around the EFCB station, where to this day lies the zero milestone for the city.

A law from 1873 created an incentive for the track construction. It would pay 30 contos de reis to each kilometre built. As good deeds don't go unpunished in Brazil, this led to tracks to be built with extra curves, a lack of structure standardisation, multiple gages and sometimes, the longest route was preferred rather than the direct one. Some effects of this law persist today, as most of the tracks use the paths opened in the XIX century.

At the time of the Republic proclamation, Brazil had 9583 km of tracks. During the first years of the regime 3383 km were added. However, the economic crisis that would eventually lead to the great depression has halted the expansion.

A.1.2 The economics of rail

Building railways is expensive. Not only the tracks, that are made of steel, had to be imported (for most of the XIX century and some of the XX century), but also, geography is not smooth and Serra do Mar, the hill that divides the coastline to the rest of the country is quite impressive, standing at 700m. Not to mention the locomotives, signaling instruments, coal, cars, wagons, etc. Most of railway growth in Brazil was financed by British money. When we take a look at the other side of the proverbial pond, there was an “economic mechanism” called Railway Mania at play. In short, people perceived tracks and locomotives as a “secure way” of investing money and the initial returns of some operations promised profits, however, as dotcom mania of the last decade of the XX century, the Rail Mania proved to be a fad and the bubble burst in 1846. Nevertheless, the bubble by mid XIX century had acquired a chronic behaviour and came back from time to time. With the maturing of the rail system in the UK (there is a limited space for railway implementation), the investment turned elsewhere, i.e. overseas. British money was not limited to railways, nonetheless, the rail mania seemed to be very much alive when examining the proportion of the money directed to railways in Brazil.

This investment has a close relation to the track expansion - with a few years shift - railways take some time to build. Figure A.2 shows the investment and track length. On the same figure, it is possible to see that 1960 was a turning point for the Brazilian rail network, something that will be explored later on this introduction. Also, it is possible to see on the investment, the effect of the Great War, the depression of 1929 and finally the Second World War.

The American rail expansion example can also teach us some lessons. After the American Civil War of 1861-1865 the Reconstruction era came. Later, on what was known as the Gilded Age, Cornelius Vanderbilt started the consolidation of the American railways in 1863. Most of the money came from the financial markets. As in England, the business of transporting people was not profitable. The great breakthrough (or break even) for the vast rail empire came with the contract to transport kerosene (for lighting) from Standard Oil (John D. Rockefeller). This steady flow of income allowed the railway to plan and expand. Today, American Class I railways are among the most important in the world, transporting mostly cargo. There are little or no subsidy to passenger transport, hence the lack of viable passenger trains in the USA to this day - except between Boston, NYC and Washington, DC (Amtrak receives US\$1.4 Billion yearly). In Europe, on the other hand, there is a large passenger high

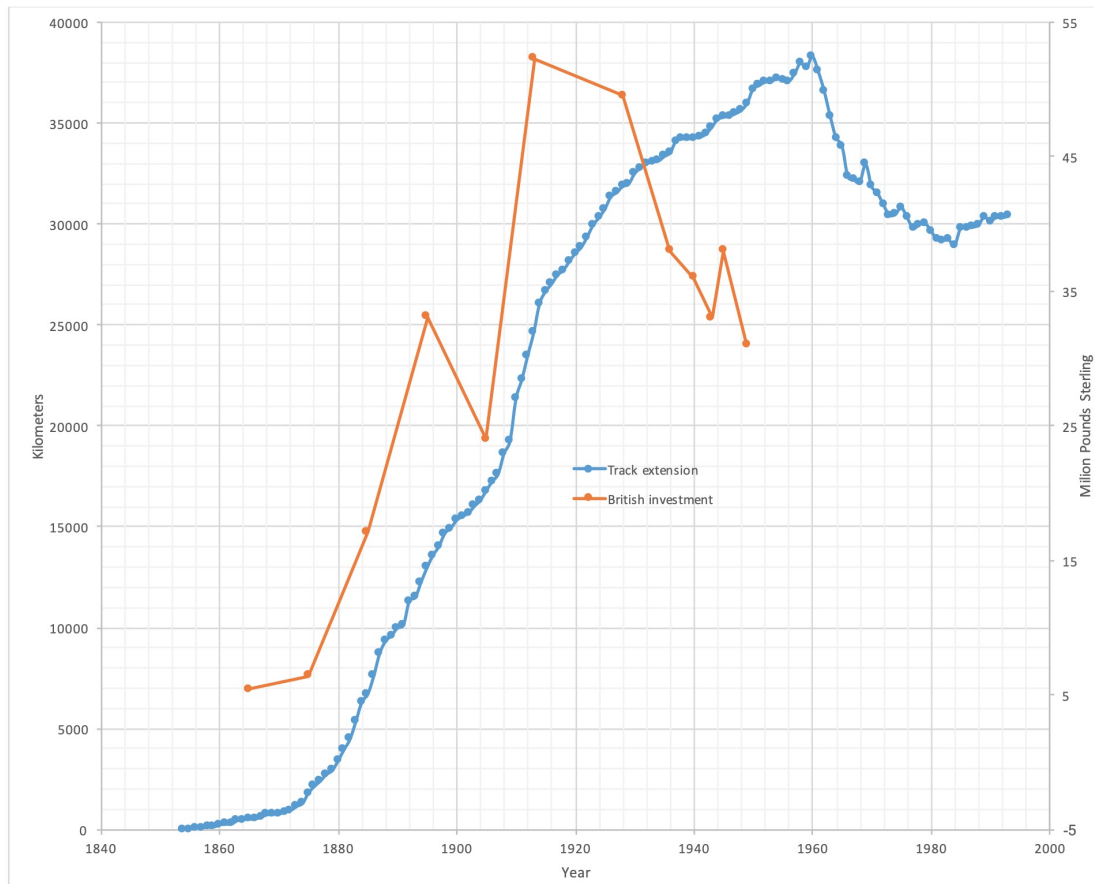


Figure A.2 – The evolution of track quantity and British investment in Brazil (Abreu, 2000) (Histórico...,)

speed network, that comes at approximately 73 billion Euro of early subsidies. In short, the economics that existed on rail transport at the time of Commodore Cornelius Vanderbilt are still at play, and passengers on rails require subsidy & state run companies (like SNCF, Deutsch Bahn, Thalys, etc) to be viable.

A.1.3 Revolution

In 1907, President Campos Sales advocated the privatisation of the railways. At the time, Brazil was fuelled primarily by coffee exports, what made the rail companies profitable. This explains how the rail network has expanded despite the lack of external (British) funds after the 1914-1918 conflict, though at a more modest pace per chart on figure A.2. Electric locomotives started replacing steam as early as 1930 (it was planned to start in 1922) and in 1936, the Diesel Electric locomotives appeared, however, the alternative to steam would only gain traction after the World War of 1939-1945. The revolution of 1930 - that marked the collapse of the so called Old Republic and was mainly caused by the fall of commodity prices

due to the crash of 1929 in Wall Street did not change much of the overall expansion panorama. It is to be noted that rail was still the best transportation method for people and cargo in general for there were very few cars, busses and trucks and there were even fewer good roads and highways. Nonetheless, it was loosing steam. The revolution and subsequent counter revolution also proved the importance of the rail as means of policy and war. In order to take power, the liberator Getulio Vargas took a train in his home state after having lost the popular vote for the government and his allies used the rail network to mobilise troops and take the capital. Later, in 1932, when S. Paulo became insurgent against the now dictator Vargas, again he used the rail network to mobilise the loyal troops from the rest of the country, crush the revolt and guarantee the unity of the country. Given that the economics have not changed and the rise of the automobile and the lower investment required by the tire transport, by 1940 most railways were bankrupt and, in a common movement by governments, they were nationalised

A.1.4 Nationalization

By 1956, the deficit in Brazil of the nationalised railway companies was about 14% of the total tax revenue of the country. To try to contain the deficit a state run company, RFFSA - Rede Ferroviária Federal Sociedade Anônima, was created - with the task of rationalising the system i.e. cutting down costs and network. From 1964 onward, synchronous with yet another military rule, RFFSA started closing lines that were deemed uneconomical. By this time, the automotive industry was firmly instituted and large highway projects had been done or were underway. The railway was not longer the passenger transport of choice, cars and busses had taken that role. Perhaps this was backed by the lack of participation of the railway on the “Redeeming revolution”, marking its decline as a political and revolutionary tool. The State of S. Paulo organised FEPASA in 1971 with 5000 km of tracks within its borders and one of the most electrified networks in the nation. As time went by, and as government goes, the electrification has proven to be much more expensive than initially anticipated and the theft of equipment plagued the system until its demise. The situation on the RFFSA was so dire in the end of the 1980s that its budget was only 19% that of the end of 1970s. By the end of the military regime most of the railway network and its parts were left for scrap and inoperable. However, during the period known as “Economic Miracle” spanning from 1969 to 1973 (and the oil crisis of 1974) the government focus was on other type of rails, the underground one. The Metro systems of São Paulo dates from 1974 and Rio’s from 1979 other capitals followed

in later years and though the network is still small compared to international benchmarks, it is difficult to fathom a city the size of São Paulo without a mass transport system such as the underground.

A.1.5 Privatisation & Ressurgence

FEPASA was merged with RFFSA in 1991 and in 1995 RFFSA was divested by the state and split into several parts. The major players today are MRS, VLI (FCA), RUMO (former ALL) and VALE, who was awarded Estrada de Ferro Vitória Minas (EFCM) and Estrada de Ferro Carajás (EFC) that are a vital part of their iron ore export system.

While some of these companies thrived, such as Vale (former CVRD), some faced dire economic situations. ALL was encamped by RUMO, its major client. The main cargo transported in Brazil is iron ore (VALE and MRS) and general cargo (VLI and Rumo) - mostly soy, sugar and paper pulp. VALE transports passengers on EFC and EFVM as part of their concession agreements, not making much profit on these operations.

By the middle of the first decade of this century, business started to improve, mainly due to the improvement of the ore business. Also, there was an increase on the iron ore prices and exports, that can be linked with the increase of sector activity, with the construction of wagons and locomotives - most of them financed by the Brazilian investment bank. Indeed, figure A.3 shows the relation between the iron ore prices and the locomotives built in Brazil, specially after privatisation. The locomotives referenced on the figure were delivered for MRS, RUMO/ALL and VLI.

A.1.6 Future perspectives

The Brazilian heavy haul rail market is mostly geared to exports, mostly ore and agricultural products, that seem to provide fair compensation, depending on the international commodity prices. The rail companies are battling to renew the concessions and the government is eager to use the money to invest on infrastructure. Rumo has recently completed the first contract for the São Paulo tracks and has recently won the concession for Ferrovia Norte Sul - centre and south. The events on a VALE mine in 2019 however have decreased the transported volume for MRS, that relies mostly on the ore mines of Minas Gerais. As the second period of the rail revolution in Brazil starts, the companies are investing in rationalisation and cost

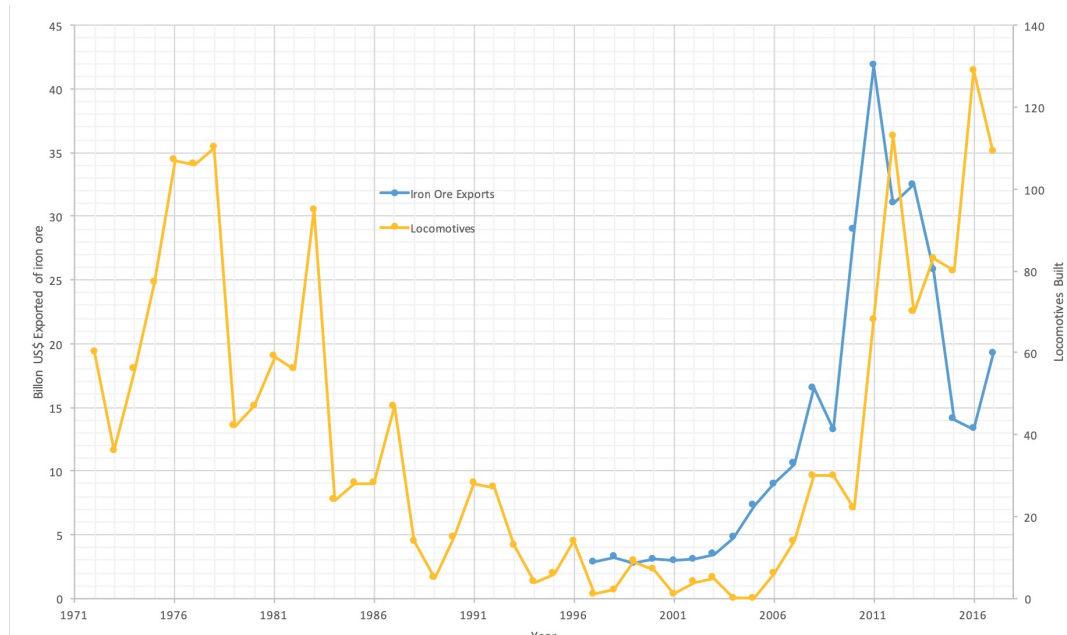


Figure A.3 – Relation between iron ore prices and locomotives built in Brazil ([ABIFER...](#),)
([Mdic...](#),)

cutting measures do ensure profitability, as well as new technologies and techniques, as will be presented in this work.

For more information please refer to ([Georgiou, 2009](#)); ([Ribeiro, 2017](#)); ([Teodoro, 2016](#)); ([Abreu, 2000](#)); ([Hobsbawm, 1996](#)); ([ANTT...](#),); ([ANTF...](#),); ([Sanchiz, 2018](#)); ([Histórico...](#),)

APPENDIX B – COMPUTER CHARACTERISATION

Computer Characterisation

The calculations were run using MATLAB 2019b (9.7.1190202) of 64 bit (maci64). The computer is a MacBook Pro mid 2015 with a i7 Quad-Core Intel processor running @ 2.5GHz. It has 16GB of DDR3 RAM memory running at 1.6GHz. It also has a discrete digital video card (AMD Radeon R9 M370X 2 GB) and an integrated video card (Intel Iris Pro 1536 MB). It is an of the shelf model MacBookPro11,5

APPENDIX C – ORIGINAL PROGRAM LISTING

The original model is comprised of 3 parts, the data pre-conditioner, the simulink file and the plot program.

The program listing be reproduced here. This program was not made by the author and was partially done by Ribeiro (([Ribeiro, 2017](#))) and Teodoro (([Teodoro, 2016](#))). For clarity, the commented command lines were suppressed from the transcript.

C.1 Run_Freio_V01

```

VALV = 0    % ABDX

% Comprimento da locomotiva [m]  VERIFICAR
Lloc=22.30;

% Comprimento do vagao [m]          VERIFICAR
% Lvag=19.40;
Lvag=9.75;
Legvag = 10.5;
% Posicao do ACT [mm]
% Lcti = 0
Lcti = 80E-03
% Lcti = 75E-03
% Lcti = 70E-03

% Numero de pares de vagoes
Nvag = 1;
% Nvag = 60;

%Nloco = [1 2
%      1 2 ];
Nloco = [1
        1 ];

```

```

NLOC = Nloco(1,end);
Nco = Nvag*2+NLOC;
COMPP = zeros(Nco,1);
COMPP(Nloco(2,:),1) = 1
FZ = zeros(Nco,1);
FACTi = zeros(Nco+1,1);
DXi = zeros(Nco+1,1);
Xi = zeros(Nco,1);
XPLOT = [1:1:Nco+1]

PP = size(COMPP);
P = PP(1,1);
y=1.4;
R=287;
T=298;
convPF = 0.03; %ICARO

conv = 6894.76;
Patm=1.01325e5;

PBi = (90*conv+Patm)*ones(PP(1,1),PP(1,2))
PRAi = (90*conv+Patm)*ones(PP(1,1),PP(1,2))
PREi = (90*conv+Patm)*ones(PP(1,1),PP(1,2))
PSRi = (90*conv+Patm)*ones(PP(1,1),PP(1,2))
PLOCi = (90*conv+Patm)*ones(PP(1,1),PP(1,2))

PFi = Patm*ones(PP(1,1),PP(1,2));
PCFi = Patm*ones(PP(1,1),PP(1,2))
ACFi = zeros(PP(1,1),PP(1,2))
VCFi = zeros(PP(1,1),PP(1,2))
AlCFi = zeros(PP(1,1),PP(1,2))

```

```

Xmin = 0.091;
Xst = 0.2122;

XCFi = Xmin* ones (PP (1,1),PP (1,2))

mc =0.25 ;
kc = 1;
cc = 1;
Fc = 100 ;
ACF =0.06477;

VOLCF = ACF*Xst*ones (PP (1,1),PP (1,2));

cont = 1;
for k = 1: Nvag+NLOC
    if COMPP(k,1) == 1
        PRAi(k,1) = 0;
        PREi(k,1) = 0;
        PCFi(k,1) = 0;
        PSRi(k,1) = 0;
        VCFi(k,1) =0;
        PFi(k,1) =0;
        cont = cont+1;
    else
        PRAi(cont,1) = 0;
        PREi(cont,1) = 0;
        PCFi(cont,1) = 0;
        PSRi(cont,1) = 0;
        VCFi(cont,1) =0;
        PFi(k,1) =0;
        cont = cont+2;
    end
end

```

```

        end
    end

    PFmin = 10*conv+Patm;
    Ploco = 90*conv+Patm;
    Peq = 90*conv+Patm;
    AlocoI = 9.5e-05;
    AlocoO = 8E-5;

    AEG = 8E-04;
    Vaux = 0.041;
    Vemg = 0.057;
    Vpsr = 0.015;
    AegAX = 2E-06;
    AegEM = 2E-06;
    AEGCF = 3.5e-06;
    Ascfc = 3E-06;
    Aesr = 8.5E-06;

    AegPSR = 1E-06;
    APSRatm = 9E-07;
    AEGatmPSR = 2E-07;
    AegaATMEM = 8E-04;    % EG atm emergencia
    ACFEG = 8.5E-6;

    DAC = 7000;
    DAL = 10000;
    DPE = 180000;
    DSRatm = 1000;
    DPcEG = 20000;

    S = size(COMPP);

```

```

SS = S(1,1);

if COMPP(end,1) ==1
    X = Lloc/2;
else
    X = Lvag/2;
end

X =18E3;
Xo(SS,1) = X;
for k = 1:SS-1
    VAG = COMPP(end+1-k,1) + COMPP(end-k,1);
    if VAG ==0
        X = X + Lvag+Lcti;
    elseif VAG == 2
        X = X + Lloc+Lcti;
    else
        X = X + (Lvag+Lloc)/2+Lcti;
    end

    Xo(SS-k,1) = X
end

MXO = max(Xo);
Xo;

% return
VO = 0;
IO = [VO*ones(Nco,1);Xo];

rEG = 0.015875;
Veg = pi*(2*rEG)^2/4*Legvag;

```



```

AM = 84;
AMe = 15;
SV = 78
SVe = 33;

VCloco = [ 0 90
           20 74
           80 90
           465 90];

SS = size(VCloco);
S =SS(1,1);
cont = 1;
difKM = 1E-10;
for k=1:S-1
    Vcontrol(cont,:) = VCloco(k,:);
    Vcontrol(cont+1,:) = [VCloco(k+1,1)-difKM    VCloco(k,2)];
    cont =cont+2;
end
Vcontrol(cont,:) = VCloco(end,:);

Tcontrol = Vcontrol(:,1);
FREIO = Vcontrol(:,2);

save Tcontrol.mat Tcontrol
save FREIO.mat FREIO

load('CMCQ.mat')
load('DPv.mat')

```

C.2 Freio_V02

This is the main program. It runs on simulink. The best attempts were made to translate the simulink program to here. Figure C.1 shows the general organisation of the program. The functions are described after the main program flow.

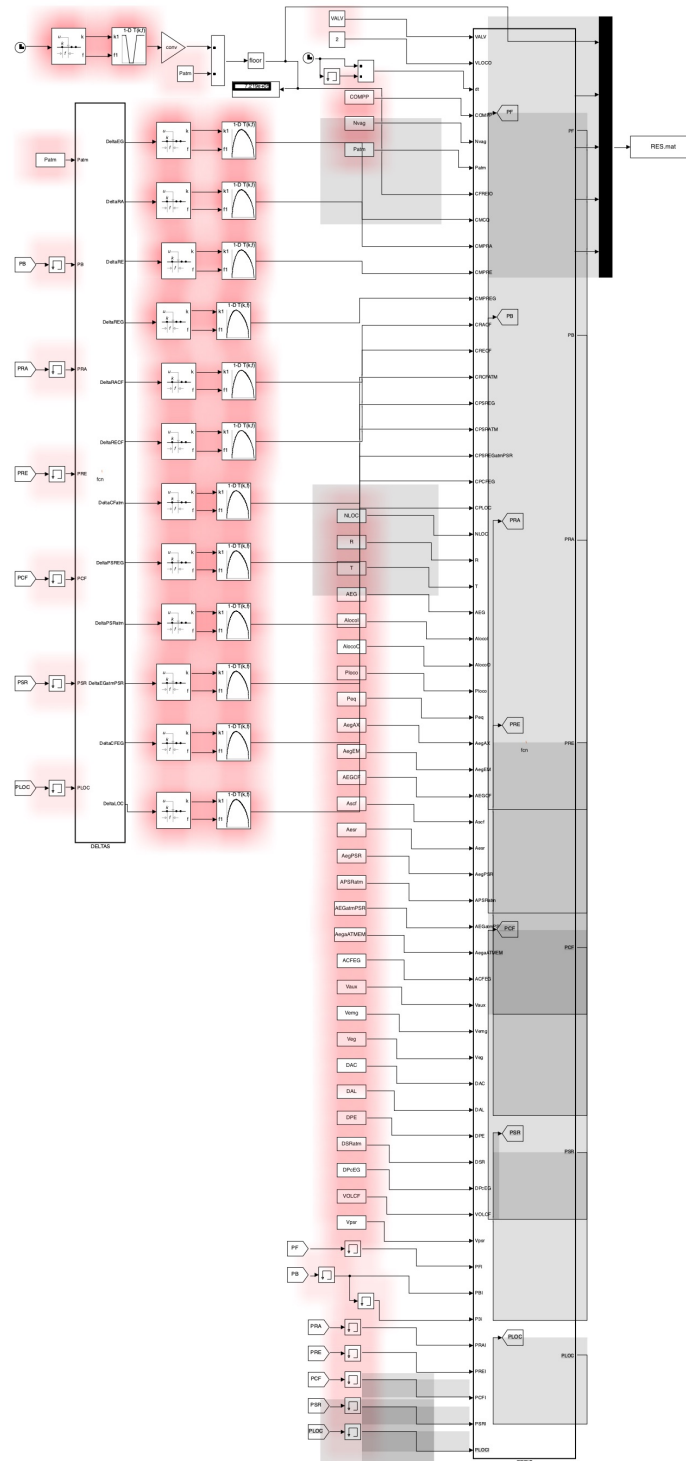


Figure C.1 – Program flow (simulink) for main original brake program

```

function [DeltaEG,DeltaRA,DeltaRE,DeltaREG,DeltaRACF,...
...DeltaRECF,DeltaCFatm,DeltaPSREG,DeltaPSRatm,...
...DeltaEGatmPSR,DeltaCFEG,DeltaLOC]
SPP = size(PB);
SP = SPP(1,1);
DeltaEG = zeros(SPP(1,1),SPP(1,2));
DeltaRA = zeros(SPP(1,1),SPP(1,2));
DeltaRE = zeros(SPP(1,1),SPP(1,2));
DeltaREG = zeros(SPP(1,1),SPP(1,2));
DeltaRACF = zeros(SPP(1,1),SPP(1,2));
DeltaRECF = zeros(SPP(1,1),SPP(1,2));
DeltaCFatm = zeros(SPP(1,1),SPP(1,2));
DeltaPSREG = zeros(SPP(1,1),SPP(1,2));
DeltaPSRatm = zeros(SPP(1,1),SPP(1,2));
DeltaEGatmPSR = zeros(SPP(1,1),SPP(1,2));
DeltaCFEG = zeros(SPP(1,1),SPP(1,2));
DeltaLOC = zeros(SPP(1,1),SPP(1,2));
for k = 1:SP(1,1)-1
    P1 = PB(k,1);
    P2 = PB(k+1,1); Pra = PRA(k+1,1);
    Pre = PRE(k+1,1); Pcf = PCF(k+1,1);...
        ... Psr = PSR(k+1,1); Ploc = PLOC(k,1);
    if P1>P2
        DeltaEG(k,1) = P2/P1;
    else
        DeltaEG(k,1) = P1/P2;
    end
    if P2>Pra
        DeltaRA(k+1,1) = Pra/P2;
    else
        DeltaRA(k+1,1) = P2/Pra;
    end
end

```

```

if Pra>Pre
    DeltaRE(k+1,1) = Pre/Pra;
else
    DeltaRE(k+1,1) = Pra/Pre;
end
if P2>Pre
    DeltaREG(k+1,1) = Pre/P2;
else
    DeltaREG(k+1,1) = P2/Pre;
end
if Pra>Pcf
    DeltaRACF(k+1,1) = Pcf/Pra;
else
    DeltaRACF(k+1,1) = Pra/Pcf;
end
if Pre>Pcf
    DeltaRECF(k+1,1) = Pcf/Pre;
else
    DeltaRECF(k+1,1) = Pre/Pcf;
end
    if Patm>Pcf
        DeltaCFatm(k+1,1) = Pcf/Patm;
    else
        DeltaCFatm(k+1,1) = Patm/Pcf;
    end
if Patm>Psr
    DeltaPSRatm(k+1,1) = Psr/Patm;
else
    DeltaPSRatm(k+1,1) = Patm/Psr;
end
if P2>Psr
    DeltaPSREG(k+1,1) = Psr/P2;

```

```

else
    DeltaPSREG(k+1,1) = P2/Psr;
end
if P2>=Patm
    DeltaEGatmPSR(k+1,1) = Patm/P2;
else
    DeltaEGatmPSR(k+1,1) = P2/Patm;
end
if P2>=Pcf
    DeltaCFEG(k+1,1) = Pcf/P2;
else
    DeltaCFEG(k+1,1) = P2/Pcf;
end
if P1>=Ploc
    DeltaLOC(k,1) = Ploc/P1;
else
    DeltaLOC(k,1) = P1/Ploc;
end
end
end

function [PF,PB,PRA,PRE,PCF,PSR,PLOC] = fcn (VALV,VLOCO,...
...dt,COMPP,Nvag,Patm,CFREIO,CMCQ,CMPRA,CMPRE,...
...CMPREG,CRACF,CRECF,CRCFATM,CPSREG,CPSRATM,...
...CPSREGatmPSR,CPCFEG,CPLOC,NLOC,R,T,AEG,AlocoI,...
...AlocoO,Ploco,Peq,AegAX,AegEM,AEGCF,Ascf,Aesr,AegPSR,...
...APSRatm,AEGatmPSR,AegaATMEM,ACFEG, Vaux,Vemg,...
...Veg,DAC,DAL,DPE,DSR,DPcEG,VOLCF,Vpsr, PFI,PBI,P3i,...
...PRAI,PREI,PCFI,PSRI,PLOCI)

```

```

PB = PBI;
PF = PFI;
PRA = PRAI;
PRE = PREI;
PCF =PCFI;
PSR = PSRI;
PLOC = PLOCI;

if VALV ==0
    cont  = 1;
    for k = 1:Nvag+NLOC
        if cont ==1
            if CFREIO>Peq
                if PB(cont,1)<=Peq
                    PB(cont,1) = Ploco;
                else
                    PB(cont,1) = Peq;
                end
            else
                if VLOCO<=0.01
                    PB(cont,1) = Ploco;
                else
                    if PB(cont+1,1)>CFREIO
                        PB(cont,1) = Patm;
                    else
                        PB(cont,1) = CFREIO;
                    end
                end
            end
        end
        cont = cont+1;
        elseif COMPP(cont,1)==1 && cont >=2
            P1 = PBI(cont-1,1);

```

```

P2 = PBI(cont,1);
P3 = PBI(cont+1,1);
if cont == 2
    if P1>P2
        A = AलोcoI;
    else
        if CFREIO == Patm
            A = 2*AलोcoO;
        else
            A = AलोcoO;
        end
    end
else
    A = AEG;
end

if P1>P2
    VM1 = A*CMCQ(cont-1,1)*P1/sqrt(T);
else
    VM1 = -A*CMCQ(cont-1,1)*P2/sqrt(T);
end

if P2>P3
    VM2 = AEG*CMCQ(cont,1)*P2/sqrt(T);
else
    VM2 = -AEG*CMCQ(cont,1)*P3/sqrt(T);
end

if CFREIO>Peq
    if PB(cont,1)<=Peq
        PLOC(cont,1) = Pलोco;
    else

```

```

        PLOC(cont,1) = Peq;
    end
else
    if VLOCO<=0.01
        PLOC(cont,1) = Ploco;
    else
        if PB(cont+1,1)>CFREIO
            PLOC(cont,1) = Patm;
        else
            PLOC(cont,1) = CFREIO;
        end
    end
end

if PLOC(cont,1)>P2
    ALOC = Alocol;
else
    if CFREIO == Patm
        ALOC = 2*Aloco0;
    else
        ALOC = Aloco0;
    end
end

if P2>PLOC(cont,1)
    VMlocT = -ALOC*CPLOC(cont,1)*P2/sqrt(T);
else
    VMlocT = ALOC*CPLOC(cont,1)*PLOC(cont,1)/sqrt(T);
end

PB(cont,1) = P2 + (R*T)/Veg*(VM1-VM2+VMlocT)*dt;
cont = cont+1;

```



```

else
    P1 = PBI(cont-1,1);
    P2 = PBI(cont,1);
    P3 = PBI(cont+1,1);
    if k == Nvag+NLOC
        P4 = PBI(cont+1,1);
    else
        P4 = PBI(cont+2,1);
    end
    PAX = PRAI(cont+1,1);
    PEX = PREI(cont+1,1);
    Pcf = PCFI(cont+1,1);
    Psr = PSRI(cont+1,1);

    if cont == 2
        if COMPP(cont-1,1) == 1
            if P1>P2
                A = AlocoI;
            else
                if CFREIO == Patm
                    A = 2*AlocoO;
                else
                    A = AlocoO;
                end
            end
        end
    else
        A = AEG;
    end
else
    A = AEG;
end

```

```

if P1>P2
    VM1 = A*CMCQ(cont-1,1)*P1/sqrt(T);
else
    VM1 = -A*CMCQ(cont-1,1)*P2/sqrt(T);
end

if P2>P3
    VM2 = AEG*CMCQ(cont,1)*P2/sqrt(T);
else
    VM2 = -AEG*CMCQ(cont,1)*P3/sqrt(T);
end
PB(cont,1) = P2 + (R*T)/Veg*(VM1-VM2)*dt;

if P3>PAX
    VMPAX = AegAX*CMPRA(cont+1,1)*PAX/sqrt(T);
else
    VMPAX = 0;
end

if PAX>PEX
    VMPEX = AegEM*CMPRE(cont+1,1)*PEX/sqrt(T);
else
    VMPEX = 0;
end

if P3>=Psr
    VMPSRin = AegPSR*CPSREG(cont+1,1)*Psr/sqrt(T);
else
    VMPSRin = 0;
end

if Psr-P3>=DSR && Psr-P3<=DPE

```

```

if Psr<=Patm
    VMPSRATM = -APSRatm*
        *CPSRATM(cont+1,1)*Patm/sqrt(T);
else
    VMPSRATM = APSRatm*
        *CPSRATM(cont+1,1)*Psr/sqrt(T);
end
if Psr-P3>=DSR*2
    if P3<=Patm
        VMEGATM = -AEGatmPSR*
            *CPSREGatmPSR(cont+1,1)*Patm/sqrt(T);
    else
        VMEGATM = AEGatmPSR*
            *CPSREGatmPSR(cont+1,1)*P3/sqrt(T);
    end
else
    VMEGATM =0;
end
else
    VMPSRATM = 0;
    VMEGATM =0;
end

if P3>P4
    VM3 = AEG*CMCQ(cont+1,1)*P3/sqrt(T);
else
    VM3 = -AEG*CMCQ(cont+1,1)*P4/sqrt(T);
end
if k == Nvag+NLOC
    VM3 =0;
end

```

```

if PAX-P3>DAC
    if PAX>=Pcf
        VMAXCF = AEGCF*CRACF(cont+1,1)*PAX/sqrt(T);
    else
        VMAXCF = -AEGCF*CRACF(cont+1,1)*Pcf/sqrt(T);
    end
else
    VMAXCF = 0;
end

if (P3 - CFREIO) > 7000
    if VLOCO<=0.01
        VMEGATM =0;
    else
        VMEGATM =  AegPSR*
            *CPSREGatmPSR(cont+1,1)*P3/sqrt(T);
    end
else
    VMEGATM =0;
end

if Psr-P3>DPE
    if VLOCO<=0.01
        VMEGATM =0;
    else
        if P3<=Patm
            VMEGATM =  -AegaATMEM*
                *CPSREGatmPSR(cont+1,1)*Patm/sqrt(T);
        else
            VMEGATM =  AegaATMEM*
                *CPSREGatmPSR(cont+1,1)*P3/sqrt(T);
        end
    end
end

```

```

        if Psr<=Patm
            VMPSRATM = -APSRatm*
            *CPSRATM(cont+1,1)*Patm/sqrt(T);
        else
            VMPSRATM = APSRatm*
            *CPSRATM(cont+1,1)*Psr/sqrt(T);
        end
    end
    if PEX>=Pcf
        VMEXCF = AEGCF*
        *CRECF(cont+1,1)*PEX/sqrt(T);
    else
        VMEXCF = -AEGCF*
        *CRECF(cont+1,1)*Pcf/sqrt(T);
    end
else
    VMEXCF = 0;
    if P3<=Patm && CFREIO<Peq && VLOCO>0.01
        if P3<=Patm
            VMEGATM = -AegaATMEM*
            *CPSREGatmPSR(cont+1,1)*Patm/sqrt(T);
        else
            VMEGATM = AegaATMEM*
            *CPSREGatmPSR(cont+1,1)*P3/sqrt(T);
        end
    end
end

if P3-DAL>=PAX
    VSATM = Ascf*CRCFATM(cont+1,1)*Pcf/sqrt(T);
    if PEX>=P3
        VRERA = Aesr*CMPREG(cont+1,1)*PEX/sqrt(T);
    end
end

```

```

else
    VRERA = 0;
end
if P3<PAX
    VMPAX = -AegAX*CMPRA(cont+1,1)*PAX/sqrt(T);
end
else
    if VLOCO<=0.01 || P3>=P3i(cont+1,1)
        VSATM = Ascfc*CRCFATM(cont+1,1)*Pcf/sqrt(T);
    else
        VSATM =0;
    end
    if P3>=CFREIO-DPcEG && P3<=CFREIO+2*
        *DPcEG || P3<=11E4
        VSATM =0;
    end
    VRERA =0;
end

if Pcf> P3 && P3> Psr
    VCFEG = ACFEG*CPCFEG(cont+1,1)*Pcf/sqrt(T);
else
    VCFEG = 0;
end

PB(cont+1,1) = P3 + (R*T)/Veg*
    * (VM2-VM3-VMPAX-VMPEX-VMPSRin-VMGATM+...
    ...+VRERA+VCFEG) *dt;

PCF(cont+1,1) = Pcf + (R*T)/VOLCF(cont+1,1)*
    * (VMAXCF+VMEXCF-VSATM-VCFEG) *dt;

```

```

PRA(cont+1,1) = PAX + (R*T) / Vaux * (VMPAX-VMAXCF) * dt;

PRE(cont+1,1) = PEX + (R*T) / Vemg * (VMPEX-VMEXCF-VRERA) * dt;

PSR(cont+1,1) = Psr + (R*T) / Vpsr * (VMPSRin-VMPSRATM) * dt;

PF(cont,1) = PCF(cont+1,1);
PF(cont+1,1) = PCF(cont+1,1);
cont = cont+2;
end
end

end

```

Plot_Freio_V01

```

% clear all
close all
clc

load('RES.mat')

Nco = 3

VFI = 4;
VFF = VFI + Nco - 2;

figure1 = figure('Color',[1 1 1], 'position',[40 550 600 300]);
axes1 = axes('Parent',figure1,'Layer','top','XAxisLocation',...
... 'bottom','FontSize',12,'FontName','Times New Roman');
box(axes1,'on');
hold(axes1,'all');
hold on

```

```

box on
grid on
set(plot (RES(1,:),RES(2,:),'k'),{'linewidth'},{3});
set(plot (RES(1,:),RES(VFI:VFF,:)),{'linewidth'},{1});
% set(plot (RES(1,:),RES(VFI:VFI+2,:)),{'linewidth'},{1});
xlabel('Tempo [s]','FontSize',12,'FontName','Times New Roman');
ylabel('Pressao EG [Pa]','FontSize',12,'FontName','Times New Roman');

% return
VI = VFF+1;
VF =VI+Nco-1;

figure1 = figure('Color',[1 1 1], 'position',[650 550 600 300]);
axes1 = axes('Parent',figure1,'Layer','top','XAxisLocation',...
... 'bottom','FontSize',12,'FontName','Times New Roman');
box(axes1,'on');
hold(axes1,'all');
hold on
box on
grid on
set(plot (RES(1,:),RES(VI:VF,:)),{'linewidth'},{1});
set(plot (RES(1,:),RES(VI,:),'k'),{'linewidth'},{2});
xlabel('Tempo [s]','FontSize',12,'FontName','Times New Roman');
ylabel('Pressao Reservatorio auxiliar [Pa]','FontSize',12,'...
...FontName','Times New Roman');

XI = VF+1;
XF =XI+Nco-1;

figure1 = figure('Color',[1 1 1], 'position',[1260 550 600 300]);
axes1 = axes('Parent',figure1,'Layer','top','XAxisLocation',...
... 'bottom','FontSize',12,'FontName','Times New Roman');

```



```

box(axes1,'on');
hold(axes1,'all');
hold on
box on
grid on
set(plot (RES(1,:),RES(XI:XF,:)),{'linewidth'},{1});
set(plot (RES(1,:),RES(XI,:),'k'),{'linewidth'},{1});
xlabel('Tempo [s]','FontSize',12,'FontName','Times New Roman');
ylabel('Pressao Reservatorio emergencia [Pa]','FontSize',12,...
... 'FontName','Times New Roman');

DI = XF+1;
DF =DI+Nco-1;

figure1 = figure('Color',[1 1 1], 'position',[40 150 600 300]);
axes1 = axes('Parent',figure1,'Layer','top','XAxisLocation',...
... 'bottom','FontSize',12,'FontName','Times New Roman');
box(axes1,'on');
hold(axes1,'all');
hold on
box on
grid on
set(plot (RES(1,:),RES(DI:DF,:)),{'linewidth'},{1});
xlabel('Tempo [s]','FontSize',12,'FontName','Times New Roman');
ylabel('Pressao Cilindro freio [Pa]','FontSize',12,...
... 'FontName','Times New Roman');

```