



Redes Neurais Autoassociativas com
Restrições de Balanço para
Reconciliação de Dados em Plantas
Químicas

Andre Luiz Pessoa Galliez Pinto

Monografia em Engenharia Química

Orientadores

Prof. Argimiro Resende Secchi, D.Sc.

Prof. Mauricio Bezerra de Souza Jr., D.Sc.

Julho de 2020

**Redes Neurais Autoassociativas com Restrições de
Balanço para Reconciliação de Dados em Plantas
Químicas**

Andre Luiz Pessoa Galliez Pinto

Monografia em engenharia química submetida ao Corpo Docente da Escola de Química, como parte dos requisitos necessários à obtenção do grau de Engenheiro Químico.

Aprovado por:

Bruno Didier Olivier Capron, D.Sc.

Flávio Leite Loução Jr., M.Sc.

Orientado por:

Argimiro Resende Secchi, D.Sc.

Mauricio Bezerra de Souza Jr., D.Sc.

Rio de Janeiro, RJ - Brasil

Julho de 2020

Galliez, Andre Pessoa

(Inserir Título)/Andre Luiz Pessoa Galliez Pinto - Rio de Janeiro: UFRJ/EQ,
2020

XI, 86 p.; il.

(Monografia) - Universidade Federal do Rio de Janeiro, Escola de Química, 2020.

Orientadores: Argimiro Resende Secchi e Mauricio Bezerra de Souza Jr.

1. Reconciliação de Dados. 2. Redes Neurais. 3. Ciência de Dados.
4. Monografia. (Graduação - UFRJ/EQ). 5 .Argimiro Resende Secchi. e
Mauricio Bezerra de Souza Jr. I. Redes Neurais Autoassociativas com
Restrições de Balanço para Reconciliação de Dados em Plantas Químicas

*Aos meus pais, Daniella e Maneco, e à minha falecida avó,
Bebete. Os alicerces sobre os quais construí tudo o que me tornei
em minha vida.*

“A jornada diligente faz valer o sacrifício”

-Black Alien, Gustavo

Agradecimentos

Inicialmente, agradeço aos meus pais, Daniella e Maneco, por sempre terem feito questão de me oferecer a melhor educação possível, além de garantir que eu tivesse sempre condições excelentes para desfrutar dela. Aqui, vale um agradecimento muito especial à minha mãe por ter revisado o texto, tornando a leitura mais agradável para todos nós.

Em seguida, agradeço aos meus orientadores, Argimiro e Maurício, por todo o conhecimento compartilhado e pela orientação extremamente precisa. A disponibilidade e paciência de vocês para lidar com a minha gestão de tempo, que nem sempre foi a melhor de todas, foi essencial para garantir que este trabalho fosse entregue no prazo e com qualidade.

Agradeço também a todos os meus amigos, em especial àqueles que de alguma forma me estimulam a ser sempre uma versão melhor de mim mesmo. Cabe aqui um agradecimento especial à minha namorada, Lara, por todo o carinho e apoio na reta final do desenvolvimento deste trabalho.

Agradeço aos colegas da Escola de Química, que me ajudaram não só a enfrentar os "10" períodos da Engenharia Química, mas também a entender o impacto que nós podemos causar no mundo, e para quê de fato serve o diploma de Engenheiro. Um salve especial pro Fiuza, que além de tudo isso ainda me livrou de encarar o temível 485 logo no 2^o período da faculdade.

Agradeço ao Felipe, aos ex-colegas do LabTER e à professora Verônica Calado por terem me dado minha primeira oportunidade de participar do processo de produção científica.

Agradeço também ao professor Thiago Cardozo e aos ex-colegas do La-

boratório de Química Teórica com quem eu dei os meus primeiros passos no mundo da programação e aprendi demais sobre o processo dialético de produção científica.

Agradeço aos meus colegas da Radix por todo o apoio neste período. Agradeço ao Gustavo por ter sido meu primeiro mentor e ter me incentivado a buscar escrever meu projeto final na área de *Data Science*. Agradeço ao Barg pelas ideias trocadas e por aquela primeira implementação de rede neuronal que serviu de base para o código desenvolvido neste trabalho. Agradeço ao Flávio por ter sido o melhor chefe de todos os tempos e sempre ter me incentivado a me aprofundar na pesquisa científica desenvolvida aqui, realizada paralelamente ao meu trabalho na Radix.

Por último, agradeço acima de tudo à minha falecida avó, Bebete, por ter feito questão que eu aprendesse desde muito cedo a importância do estudo. "Faça seus deveres de casa meu filho, um dia você vai me agradecer". Hoje, talvez mais do que nunca, é hora de dizer: "Obrigado vó".

Resumo da Monografia apresentada à Escola de Química como parte dos requisitos necessários para obtenção do grau de Engenheiro Químico.

Redes Neurais Autoassociativas com Restrições de Balanço para Reconciliação de Dados em Plantas Químicas

Andre Luiz Pessoa Galliez Pinto

Julho, 2020

Orientadores: Prof. Argimiro Resende Secchi, D.Sc.

Prof. Mauricio Bezerra de Souza Jr., D.Sc.

Resumo

No contexto da nova onda de transformações digitais conhecida como 4^a revolução industrial, técnicas baseadas em Inteligência Artificial tem se popularizado para resolver problemas típicos da indústria como a reconciliação de dados. Para este fim, algoritmos de *machine learning* se apresentam como uma alternativa promissora frente às abordagens clássicas de programação não-linear, devido ao seu menor custo computacional. Neste trabalho investiga-se o uso de redes neurais autoassociativas para reconciliar variáveis de uma planta simulada de produção de fenilbenzeno em estado estacionário. O desempenho da rede é avaliado assim como as implicações de combiná-la com equações de balanço de massa. As técnicas utilizadas apresentaram bons resultados, sendo capaz de produzir estimativas melhores do que as medições originais e sugerindo que o desempenho do modelo pode ser melhorado pela inserção de equações fenomenológicas na função objetivo.

Conteúdo

Lista de Tabelas	xii
Lista de Figuras	xiv
Siglas	xx
1 Introdução	1
1.1 Motivação	3
1.2 Objetivos	5
1.3 Organização	6
2 Revisão Bibliográfica	7
2.1 A técnica de reconciliação de dados	7
2.2 Erros de Medição	10
2.3 Classificação de Variáveis	12
2.4 Detecção de erros grosseiros	14

2.5	Estratégias de formulação	16
2.6	Formulação clássica	18
2.6.1	O problema de otimização	18
2.6.2	Reconciliação de dados clássica: revisão de trabalhos relevantes	21
2.7	Redes Neurais Artificiais	24
2.7.1	Treinando a rede	28
2.7.2	O problema de reconciliação usando redes neuronais . .	32
2.7.3	Redes neuronais aplicadas à reconciliação: revisão de trabalhos relevantes	34
3	Sistema Estudado	41
3.1	Descrição da planta e modelagem matemática	43
3.1.1	Reator	44
3.1.2	Tanque de Flash	46
3.1.3	Conectores	49
3.1.4	Aquecedor e resfriador	49
3.2	Faixa de operação	50
3.2.1	Pressão no reator	50
3.2.2	Temperatura no reator	51
3.2.3	Vazão de alimentação	54

3.2.4	Pressão e Temperatura no Flash	54
3.2.5	Razão de reciclo	57
3.3	Validação da Simulação	59
4	Formulação do Problema	61
4.1	Distribuição dos Dados	62
4.2	Pré-processamento	64
4.3	Função objetivo	65
4.4	Topologia	67
4.5	Treinamento	68
4.6	Métricas	69
5	Resultados e Discussão	71
5.1	Reconciliação sem penalização	72
5.1.1	Flash	72
5.1.2	Reator	74
5.2	Reconciliação com as violações de balanço penalizadas	77
5.2.1	Flash	78
5.2.2	Reator	81
5.3	Reconciliação com penalização de <i>Tikhonov</i>	84
5.3.1	Flash	84

5.3.2	Reator	87
6	Conclusão e sugestões para trabalhos futuros	90
	Bibliografia	93
	Bibliografia	93
A	Resultados e distribuição dos dados	101
A.1	Distribuição das variáveis	102
A.2	Distribuição dos desbalanceamentos	108
A.3	Flash com $\lambda = 0$ e $l_2 = 0$	112
A.4	Reator com $\lambda = 0$ e $l_2 = 0$	116
A.5	Flash com $\lambda = 0,25$ e $l_2 = 0$	119
A.6	Flash com $\lambda = 0,25$, 4 neurônios na camada de gargalo e $l_2 = 0$.	123
A.7	Reator com $\lambda = 0,1$ e $l_2 = 0$	127
A.8	Reator com $\lambda = 0,1$, 4 neurônios na camada de gargalo e $l_2 = 0$	130
A.9	Flash com $\lambda = 0,25$ e $l_2 = 0,0005$	133
A.10	Reator com $\lambda = 0,1$ e $l_2 = 0,0004166$	137

Lista de Tabelas

3.1	Representação dos compostos	44
3.2	Constantes de velocidade das reações de formação de fenilbenzeno ($n = 1$) e meta-terfenilo ($n = 2$).	46
3.3	Parâmetros usados na equação de Antoine [20]	48
3.4	Parâmetros usados na equação de Wagner.	48
3.5	Inputs usados na simulação de validação.	59
3.6	Comparação entre as simulações.	60
4.1	Valores utilizados para as variáveis de entrada.	62
4.2	Valores dos ruídos inseridos	63
5.1	Redução do EQM e razão entre os desvios padrão real e das medições	74
5.2	Violações de Balanço no flash com $\lambda = 0$	76
5.3	Redução do EQM e razão entre os desvios padrão real e das medições	77

5.4	Violações de Balanço no reator com $\lambda = 0$	77
5.5	Violações de Balanço no flash	85
5.6	Violações de Balanço no reator	88

Lista de Figuras

2.1	Etapas do processo de RD.	10
2.2	Esquema de uma rede neuronal <i>feedforward</i>	25
2.3	Esquema de uma rede neuronal com recorrências.	27
2.4	Exemplo de <i>overfitting</i>	32
2.5	Topologia de uma rede neuronal autoassociativa.	37
3.1	Fluxograma do processo.	43
3.2	Frações dos componentes B e D na saída do reator a 973K. . .	51
3.3	Conversão do Benzeno no reator versus P_r	52
3.4	P_d^{sat} versus T . Linha pontilhada em 12 12bar.	53
3.5	Frações dos componentes B e D na saída do reator a 12bar . .	53
3.6	Frações dos componentes B e D versus F_o	54
3.7	Variação do tempo de residência com F_o	55
3.8	Influência de T_F e P_F no fator β	56

3.9	Influência de T_F e P_F na separação dos componentes A e B	57
3.10	Influência de C_s nos produtos obtidos.	58
3.11	Influência de C_s na vazão de entrada do reator.	58
5.1	Evolução de $\hat{\sigma}_{rel}$ e EQM ao longo do treinamento. Treina- mento interrompido na linha pontilhada.	73
5.2	Relação entre a diminuição do Erro Quadrático Médio e σ_{rel} no flash com $\lambda = 0$	75
5.3	Evolução de $\hat{\sigma}_{rel}$ e EQM ao longo do treinamento. Treina- mento interrompido na linha pontilhada.	76
5.4	Relação entre a diminuição do Erro Quadrático Médio e $\hat{\sigma}_{rel}$ no reator com $\lambda = 0$	78
5.5	EQM em função de λ para o flash. Resultado para $\lambda = 0$ representado pela linha pontilhada. Melhor valor de $\lambda = 0,25$ representado por círculo maior em vermelho. Valores repre- sentados por círculos azuis representam cálculos que não atin- giram $\sigma_{rel} = 0,96$	79
5.6	Desbalanceamentos em função de λ para o flash. Melhor valor de λ marcado com um x.	80
5.7	EQM e desbalanceamentos do flash em função do número de neurônios da camada de gargalo.	81

5.8	EQM em função de λ para o reator. Resultado para $\lambda = 0$ representado pela linha pontilhada. Melhor valor de $\lambda = 0,1$ representado por círculo maior em vermelho. Valores representados por círculos azuis representam cálculos que não atingiram $\sigma_{rel} = 0,96$	82
5.9	Desbalanceamentos em função de λ para o reator. Melhor valor de λ marcado com um x.	82
5.10	EQM e desbalanceamentos no reator em função do número de neurônios da camada de gargalo.	83
5.11	EQM em função de l_2 para o flash. Resultado para $l_2 = 0$ representado pela linha pontilhada. Melhor valor de $l_2 = 0,0005$ indicado pelo círculo maior em vermelho.	85
5.12	EQM em função de l_2 para o flash. Resultado para $l_2 = 0$ representado pela linha pontilhada. Melhor valor de $l_2 = 0,0005$ indicado pelo círculo maior em vermelho.	86
5.13	Desbalanceamentos em função de l_2 para o flash. Melhor valor de l_2 marcado com um x.	87
5.14	EQM em função de l_2 para o reator. Resultado para $l_2 = 0$ representado pela linha pontilhada. Melhor valor de $l_2 = 0,0004166$ indicado pelo círculo maior em vermelho.	88
5.15	Desbalanceamentos em função de l_2 para o reator. Melhor valor de l_2 marcado com um x.	89
A.1	Distribuição das vazões molares.	102

A.2	Distribuição das composições na entrada do reator.	103
A.3	Distribuição das composições na saída do reator/entrada do flash.	104
A.4	Distribuição das composições na corrente líquida de saída do flash.	105
A.5	Distribuição das composições na corrente gasosa de saída do flash.	106
A.6	Distribuição de pressão e temperatura no reator e flash.	107
A.7	Distribuição das violações no balanço de massa global no reator.	108
A.8	Distribuição das violações no balanço de massa por componente no reator.	109
A.9	Distribuição das violações no balanço de massa global no flash.	110
A.10	Distribuição das violações no balanço de massa por componente no flash.	111
A.11	Resultado da reconciliação para o flash com $\lambda = 0$ e $l_2 = 0$. . .	112
A.12	Resultado da reconciliação para o flash com $\lambda = 0$ e $l_2 = 0$. . .	113
A.13	Resultado da reconciliação para o flash com $\lambda = 0$ e $l_2 = 0$. . .	114
A.14	Resultado da reconciliação para o flash com $\lambda = 0$ e $l_2 = 0$. . .	115
A.15	Resultado da reconciliação para o reator com $\lambda = 0$ e $l_2 = 0$. .	116
A.16	Resultado da reconciliação para o reator com $\lambda = 0$ e $l_2 = 0$. .	117
A.17	Resultado da reconciliação para o reator com $\lambda = 0$ e $l_2 = 0$. .	118

A.18	Resultado da reconciliação para o flash com $\lambda = 0,25$ e $l_2 = 0$.	119
A.19	Resultado da reconciliação para o flash com $\lambda = 0,25$ e $l_2 = 0$.	120
A.20	Resultado da reconciliação para o flash com $\lambda = 0,25$ e $l_2 = 0$.	121
A.21	Resultado da reconciliação para o flash com $\lambda = 0,25$ e $l_2 = 0$.	122
A.22	Resultado da reconciliação para o flash com 4 neurônios na camada de gargalo, $\lambda = 0,25$ e $l_2 = 0$.	123
A.23	Resultado da reconciliação para o flash com 4 neurônios na camada de gargalo, $\lambda = 0,25$ e $l_2 = 0$.	124
A.24	Resultado da reconciliação para o flash com 4 neurônios na camada de gargalo, $\lambda = 0,25$ e $l_2 = 0$.	125
A.25	Resultado da reconciliação para o flash com 4 neurônios na camada de gargalo, $\lambda = 0,25$ e $l_2 = 0$.	126
A.26	Resultado da reconciliação para o reator com $\lambda = 0,1$ e $l_2 = 0$	127
A.27	Resultado da reconciliação para o reator com $\lambda = 0,1$ e $l_2 = 0$	128
A.28	Resultado da reconciliação para o reator com $\lambda = 0,1$ e $l_2 = 0$	129
A.29	Resultado da reconciliação para o reator com 4 neurônios na camada de gargalo, $\lambda = 0,1$ e $l_2 = 0$.	130
A.30	Resultado da reconciliação para o reator com 4 neurônios na camada de gargalo, $\lambda = 0,1$ e $l_2 = 0$.	131
A.31	Resultado da reconciliação para o reator com 4 neurônios na camada de gargalo, $\lambda = 0,1$ e $l_2 = 0$.	132

A.32 Resultado da reconciliação para o flash com $\lambda = 0,25$ e $l_2 =$ 0,0005.	133
A.33 Resultado da reconciliação para o flash com $\lambda = 0,25$ e $l_2 =$ 0,0005.	134
A.34 Resultado da reconciliação para o flash com $\lambda = 0,25$ e $l_2 =$ 0,0005.	135
A.35 Resultado da reconciliação para o flash com $\lambda = 0,25$ e $l_2 =$ 0,0005.	136
A.36 Resultado da reconciliação para o reator com $\lambda = 0,1$ e $l_2 =$ 0,0004166.	137
A.37 Resultado da reconciliação para o reator com $\lambda = 0,1$ e $l_2 =$ 0,0004166.	138
A.38 Resultado da reconciliação para o reator com $\lambda = 0,1$ e $l_2 =$ 0,0004166.	139

Siglas

AANN Redes Neurais Autoassociativas. 4, 38, 39, 67, 71

EKF Filtros de Kalman Extendidos. 3, 17, 35

EQM Erro Quadrático Médio. xii, xv, xvi, 72–83, 85–88

MQP Mínimos-Quadrados Ponderados. 19, 21, 23, 65

NLP Programação Não-linear. 3, 4, 17

PCA Análise de Componentes Principais. 36, 91

RD Reconciliação de Dados. xiv, 3, 5, 6, 9, 10, 12, 14, 16–18, 20–22, 24, 32, 38, 39, 41, 42, 65, 92

RN Rede Neuronal. 25, 27, 28, 36

RNs Redes Neurais. 4–6, 17, 27, 34, 38, 41, 42, 61, 64, 90

SBANN Rede Neuronal Autoassociativa Baseada em Sistema. 39

Capítulo 1

Introdução

O paradigma da coleta, armazenamento e processamento de dados vem ganhando muita visibilidade nas últimas duas décadas. O recente crescimento do uso de novas tecnologias capazes de utilizar grandes quantidades de dados para gerar transformações em procedimentos cotidianos tem causado significativo impacto no dia-a-dia do cidadão pós-moderno. O uso de técnicas como *Big Data*, *Machine Learning* (IA), Internet das coisas (IoT) e Computação em Nuvem (CC), entre outros, permite que grandes quantidades de informação sejam rapidamente extraídas, processadas e armazenadas, agregando valor e facilitando processos de tomada de decisão em diversas esferas. O desenvolvimento e a aplicação dessas tecnologias têm ficado popularmente conhecidos como a "4ª Revolução Industrial" graças às mudanças disruptivas que as mesmas têm promovido no âmbito da automação e integração das atividades industriais. As recentes transformações nessa esfera sugerem que a humanidade pode estar novamente frente a um momento histórico, onde a mudança na lógica de produção dos bens de consumo pode transformar completamente as relações de trabalho e a sociedade, como ocorreu na virada

para o século XIX e no início do século XX.

A expressão "*Data is the new oil*", cunhada pelo matemático britânico Clive Humby em 2006 [5], ficou famosa ao sugerir que os dados serão a grande *commodity* a ser extraída e processada no século XXI, assim como o petróleo o foi no século XX. Da mesma maneira que o petróleo, dados também são um ativo que necessita ser extraído e processado de forma a se tornar produto de valor capaz de proporcionar vantagens e ganhos para empresas e indivíduos. Na atual realidade de processos químicos e bioquímicos, extrair dados significa instrumentar e historiar suas variáveis, e refiná-los significa utilizá-los para extrair da planta a maior quantidade possível de informações úteis e precisas. Estas informações podem então ser utilizadas para reduzir custos e promover ganhos de eficiência. No contexto operacional, isso pode significar desde quantificar a saúde de equipamentos para se minimizar os custos associados a paradas e manutenções na planta, até cruzar informações para se estimar parâmetros que sejam estratégicos na tomada de decisão com alta confiabilidade. Na prática, o acesso facilitado a informações mais confiáveis, em tempo real, tem se transformado em um ativo essencial para que as empresas possam competir em um mercado globalizado e atender a legislações ambientais cada vez mais exigentes.

Nesta conjuntura, as técnicas de retificação de dados surgem como uma estratégia para desempenhar a tarefa de trazer mais confiabilidade às informações obtidas. Valendo-se das redundâncias que surgem ao se combinar modelos fenomenológicos com medições oriundas dos sensores em uma planta, esta técnica é capaz de avaliar e corrigir erros provenientes de problemas inerentes às medições. Assim, ao garantir que as informações obtidas estejam o mais próximo possível da realidade, a retificação de dados torna-se uma

ferramenta fundamental para garantir qualidade e segurança bem como para aumentar a eficiência dos processos.

1.1 Motivação

O estudo de técnicas de reconciliação de dados (RD) no contexto da engenharia química tem início na década de 60 com a solução apresentada por Kuhen & Davidson [33] para um estado estacionário com restrições lineares. Ao longo das décadas seguintes, diversas abordagens foram propostas na literatura para solucionar casos mais complexos e próximos da realidade. Estas abordagens consideravam a dinâmica e não-linearidade dos processos, além de diferentes estratégias para a identificação de erros seguindo distribuições estatísticas desconhecidas.

Contudo, o problema de RD apresenta características que fazem com que sua implementação em situações reais se torne tarefa difícil. Casos reais desse problema frequentemente impõem que sistemas complexos de equações algébricas diferenciais não-lineares sejam avaliados, exigindo considerável esforço computacional. Por outro lado, para agregar valor à operação de uma planta química é necessário que o problema de reconciliação seja resolvido em tempo real. Dessa forma, muitas estratégias foram sugeridas nos últimos 50 anos a fim de tentar produzir bons resultados e manter o tempo de cálculo baixo o suficiente para que seja possível uma implementação *on-line*. A grande maioria dessas estratégias publicadas consiste em algoritmos que podem ser classificados em três grupos: Filtros de Kalman Estendidos (EKF), Programação Não-linear (NLP) e Redes Neurais Artificiais.

Técnicas de EKF, embora amplamente empregadas, apresentam desem-

penho limitado devido ao fato de considerarem que as restrições de processo são localmente lineares e que a matriz covariância é conhecida, o que frequentemente não acontece [8, 26]. As técnicas de NLP foram amplamente estudadas e apresentaram resultados bastante satisfatórios utilizando estimadores robustos associados a técnicas de otimização meta-heurísticas e janelas de tempo de amostragem incrementais [27, 43, 49]. Contudo, esta abordagem apresenta a desvantagem de ser computacionalmente intensiva o que pode inviabilizar sua implementação em tempo real.

Por outro lado, o uso de RNs possui a vantagem de fornecer resultados robustos com baixíssimo custo computacional uma vez que a rede tenha sido treinada. Resultados promissores foram publicados por Bai et al. [8] usando redes neuronais autoassociativas (AANN). A grande desvantagem desse algoritmo consiste no fato de ele necessitar de grandes quantidades de dados para ser treinado, o que muitas vezes não está disponível em situações reais.

No entanto, novos e recentes fatores provenientes do advento da 4^a Revolução Industrial têm criado condições cada vez mais favoráveis para a aplicação de Redes Neuronais. Um deles consiste no fato de que o aumento no uso de técnicas de *machine learning* tem feito com que muito conhecimento a respeito desse tipo de algoritmo seja amplamente difundido, o que faz com que seja cada vez mais fácil se criar implementações eficientes dele. Outro fator importante diz respeito à principal desvantagem, já referida, das Redes Neuronais: a necessidade de grandes amostras de dados para treiná-las. A maturidade de empresas e instituições a respeito da importância de se possuir uma arquitetura consistente de armazenamento de dados tem crescido muito nas últimas décadas, ajudando a garantir a disponibilidade de

amostras grandes o suficiente para realizar o treinamento.

Aliado a isso, as tecnologias recentes voltadas para obtenção, armazenamento e processamento de dados têm tornado cada vez mais fácil o desenvolvimento de arquiteturas desse tipo. Essas condições fazem com que muito mais dados estejam disponíveis para o treinamento de RNs, viabilizando o seu uso em maior escala.

1.2 Objetivos

Devido aos fatores apresentados na seção anterior e aos bons resultados já publicados na literatura [8, 47], o uso de RNs aplicadas ao problema de RD se apresenta como uma alternativa promissora. Por originalmente consistirem em uma técnica totalmente baseada em dados, as redes inferem os padrões expressados a partir dos modelos físico-químicos puramente por meio das correlações presentes no conjunto de dados utilizado [26, 31, 47]. Neste ponto, esta abordagem se diferencia significativamente das técnicas de reconciliação clássicas, as quais utilizam estes modelos como fonte de informação a respeito do comportamento das variáveis [13, 33, 40].

Dentre as diferentes implementações de redes neuronais já realizadas para reconciliação de dados, destaca-se o uso de redes autoassociativas [9, 22, 31]. Baseando-se em um princípio de compressão e regeneração de informações[32], esta técnica demonstrou ser capaz de filtrar ruídos presentes nas medições e promover uma melhoria na qualidade dos dados mesmo na ausência das equações fenomenológicas [22, 26, 47]. Contudo, trabalhos recentes sugerem que o desempenho desta abordagem pode ser melhorado combinando-a com estas equações [8, 9].

A partir do panorama traçado, este trabalho se propõe a avaliar o desempenho das redes autoassociativas para realizar a reconciliação em uma planta estacionária simulada de fenilbenzeno. São considerados cenários com e sem a presença das equações de balanço, de forma a avaliar qual é o impacto do uso destas no desempenho das RNs para melhorar a qualidade das medições e reduzir as violações nos modelos físicos.

1.3 Organização

Este trabalho está estruturado em 6 capítulos, incluindo esta breve Introdução. No Capítulo 2 é realizada uma revisão bibliográfica do problema de RD e do uso de RNs no contexto da engenharia química. No Capítulo 3, o processo de produção de fenilbenzeno usado como base para a exploração do problema de RD é descrito em detalhes. No Capítulo 4 são introduzidas as diferentes condições e formulações matemáticas utilizadas para se treinar as RNs e avaliar seu desempenho. No Capítulo 5 são apresentados os resultados obtidos e no Capítulo 6 são descritas as principais conclusões e sugestões para trabalhos futuros.

Capítulo 2

Revisão Bibliográfica

2.1 A técnica de reconciliação de dados

De forma geral, a operação correta de uma planta química depende de medidas de processo confiáveis. Procedimentos rotineiros como o controle da qualidade do produto gerado, o monitoramento e ajuste das faixas de operação e a garantia da segurança dos operadores são altamente influenciados pela acurácia das medições disponíveis, sejam elas oriundas de sensores ou análises laboratoriais. Assim, para se garantir a eficiência e a viabilidade econômica da operação, é essencial ser capaz de estimar com alguma confiança seus parâmetros relevantes.

No entanto, a grande maioria das informações obtidas diretamente dos instrumentos da planta e equipamentos de laboratório não tem um grau de confiabilidade tão elevado quanto o desejado. Tipicamente, os valores obtidos por sensores e rotinas de medição estão corrompidos por diferentes tipos de erro. A operação correta desses sensores está sujeita a vários problemas

como má instalação, má calibração, falhas mecânicas, mau funcionamento de componentes físicos, flutuações de energia elétrica, deterioração e a imprecisão inerentes aos próprios equipamentos. Essa deterioração dos dados pode se manifestar na forma de erros pequenos e aleatórios ou de erros ditos grosseiros, que representam erros sistemáticos ou valores espúrios, muito distantes do valor real da variável sendo medida [36].

Apesar da dificuldade de se obter valores confiáveis das medições diretas, pode-se utilizar métodos matemáticos para ajustar as medições cruas obtidas. Isso porque informações a respeito do estado da planta podem ser extraídas a partir de modelos teóricos ou empíricos, como balanços de massa e energia ou equações com parâmetros estimados. Além disso, a maioria das plantas monitora mais variáveis do que seria necessário para se determinar completamente o seu estado usando as equações fenomenológicas. Assim, é possível combinar as medições disponíveis com modelos e balanços para se chegar a estimativas das variáveis mais próximas da realidade. Vale ressaltar que o rigor considerado ao avaliar a acurácia de um modelo deve ser inversamente proporcional à magnitude dos ruídos de medição. Se a variância dos erros de medição for grande, modelos não tão acurados podem gerar ganhos valiosos para a retificação dos dados. No entanto, se os ruídos forem pequenos, será necessário modelos muito acurados para garantir que os resultados obtidos pela retificação sejam de fato mais próximos da realidade do que as informações advindas diretamente dos sensores [35, 40].

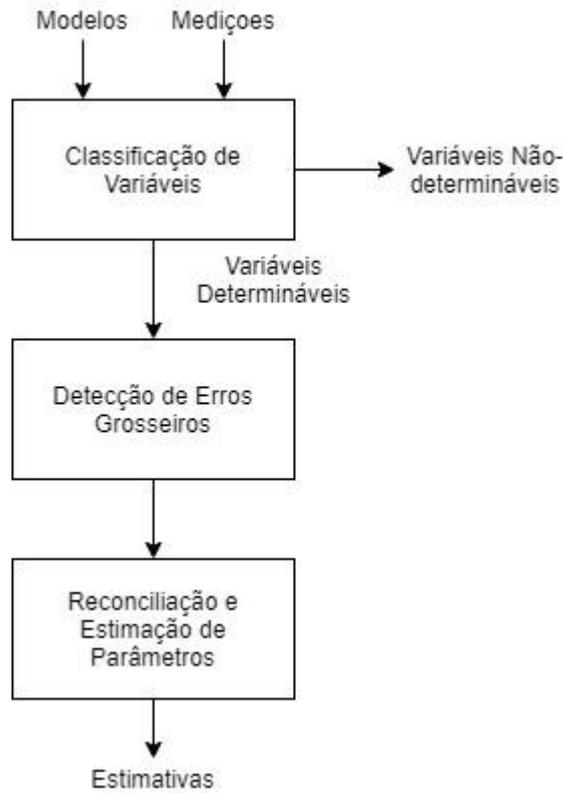
Nesse contexto, a reconciliação de dados surge como uma técnica que utiliza as redundâncias espaciais e temporais das medidas em relação aos modelos fenomenológicos para gerar melhores estimativas para as variáveis da planta. Medições são ditas espacialmente redundantes quando existem

mais dados do que o suficiente para se definir completamente o estado de um modelo em um dado instante de tempo, e temporalmente redundantes quando medições de instantes de tempo passados estão disponíveis e podem ser utilizadas para estimação de outras variáveis no instante de tempo atual [34]. Redundâncias temporais são restritas ao problema de RD dinâmico.

O processo de RD é normalmente dividido em três etapas: a classificação das variáveis relevantes, a detecção de erros grosseiros e a reconciliação propriamente dita, como mostra a figura 2.1. Na primeira etapa, as variáveis são separadas entre aquelas que podem ser obtidas ou estimadas a partir das medições, e aquelas que não são acessíveis com as informações disponíveis, ditas não-observáveis. Na segunda etapa, as medições contendo valores espúrios são identificadas através de métodos estatísticos para que possam ser tratadas de forma diferente dos pequenos erros aleatórios durante a reconciliação de dados. Na terceira etapa, novos valores para as variáveis medidas e parâmetros observáveis são estimados a partir da resolução de um problema de otimização com restrições [7, 34]. Outras técnicas também empregadas ao problema de RD procuram reduzir as distorções causadas por erros grosseiros usando estimadores robustos, lidando com a sua estimação e com a reconciliação de dados de forma simultânea [4, 40, 50].

Assim, a reconciliação de dados visa combinar de maneira ótima o conhecimento disponível dos sensores e dos modelos matemáticos, respeitando as particularidades do sistema, para conferir o máximo de confiabilidade aos parâmetros relevantes do processo. Contudo, fatores como a complexidade dos modelos de engenharia química, a dificuldade em se prever a distribuição estatística dos erros, e a necessidade de se resolver o problema *on-line* tornam difícil a implementação da RD em uma situação real. Dessa forma, é de suma

Figura 2.1: Etapas do processo de RD.



Fonte: Adaptado de Liebman et al. [34].

importância conhecer bem as estratégias disponíveis e as particularidades do sistema em questão para que se possa adotar um algoritmo apropriado.

2.2 Erros de Medição

Conforme exposto, os erros presentes nos dados provenientes de uma planta podem ser discriminados de acordo com suas características e natureza estatística. Liebman et al. [34] classificaram os erros típicos de processos reais nas seguintes categorias:

- Pequenos erros aleatórios, que resultam de flutuações nas medições devido à imprecisão dos sensores. Estes erros seguem uma distribuição normal com média zero. Por serem oriundos de imprecisões aleatórias nas medições dos sensores, erros desse tipo são considerados não-correlacionados, ou seja, não têm relação com erros provenientes de outros sensores ou do mesmo sensor em instantes de tempo diferentes.
- Erros sistemáticos, também chamados *bias*, que resultam de problemas reais na planta, como sensores mal instalados ou mal calibrados inserindo algum tipo de viés na medição. Este tipo de erro tem a média diferente de zero, provocando medições com valores inferiores ou superiores ao valor verdadeiro, e costuma apresentar correlação temporal. Dependendo de sua magnitude, este tipo de erro pode ou não representar um valor espúrio.
- Erros grosseiros, que podem ser provocados por falhas graves nos sensores, vazamentos na planta ou distúrbios no processo. Erros deste tipo apresentam valores espúrios, ou seja, discrepâncias muito significativas com relação aos valores verdadeiros. Normalmente, eles podem ser identificados com facilidade ao se analisar o sinal. Em muitos casos, erros grosseiros não são correlacionados com outros erros. Porém, quando decorrentes de problemas na planta, pode haver correlação temporal ou espacial entre eles.

Apesar da distinção feita por Liebman entre erros grosseiros e sistemáticos, outros autores como Kong et al. [30] e Ramamurthi et al. [43] discriminam os tipos de erro em apenas duas categorias: aleatórios e grosseiros, sendo os erros sistemáticos incluídos na categoria dos grosseiros. Embora haja diferenças nas causas e naturezas estatísticas dos tipos de erro, a classificação em apenas

duas categorias é útil, uma vez que é necessário identificar a presença tanto de erros grosseiros quanto sistemáticos antes de se determinar a técnica utilizada na formulação do problema de RD. Assim, a partir deste ponto, este trabalho utilizará o termo erros grosseiros para a identificação destes dois tipos de erro.

Em casos onde o funcionamento dos sensores se dá de maneira correta e as equações fenomenológicas adotadas representam bem o processo, espera-se que apenas erros pequenos e aleatórios ocorram. Neste caso, o problema de reconciliação de dados torna-se mais simples e não há a necessidade de muita preocupação com a etapa de identificação de erros grosseiros. No entanto, é sempre importante realizar análises estatísticas para identificar se há ou não a presença de erros grosseiros, pois a presença dos mesmos introduz um novo nível de complexidade no problema, o que pode tornar necessário o uso de estimadores e técnicas matemáticas mais robustas [36, 40].

2.3 Classificação de Variáveis

Discriminar as variáveis pertencentes ao sistema estudado de acordo com sua disponibilidade e observabilidade é crucial para formular estrategicamente o problema de RD. Segundo Crowe [12], variáveis podem ser classificadas em redundantes ou não-redundantes e observáveis ou não-observáveis. Variáveis observáveis são aquelas que podem ser obtidas explicitamente através de medições ou implicitamente a partir dos modelos do processo fazendo uso de outras medições disponíveis. Analogamente, uma variável é dita não-observável caso a mesma não seja medida ou não possa ser obtida através dos modelos e medições. Do ponto de vista da redundância, uma variável será

considerada redundante se for possível estimá-la mesmo após sua medição ser removida do sistema. Ou seja, uma variável redundante pode ser obtida tanto a partir de medições explícitas quanto implícitas. Dessa forma, uma variável é dita não-redundante se a mesma se torna não-observável ao ter sua medição removida.

Se empregada juntamente às etapas de projeto de uma planta química, a classificação de variáveis pode auxiliar na garantia do bom funcionamento da supervisão do processo. Mapear fatores como a sensibilidade de uma variável para o processo, o número de vezes que ela aparece nos balanços e qual a incerteza inerente ao sensor utilizado para medi-la, ajuda a assegurar a observabilidade dos parâmetros relevantes para a operação. Realizar esse estudo durante o projeto da planta pode significar instalar mais instrumentos de medição em unidades cujos sensores possuam incertezas maiores, ou utilizar estratégias de medição mais robustas em partes do processo mais sensíveis. Essas estratégias são importantes para prevenir que parâmetros importantes se tornem não-observáveis mediante falhas de sensores, o que poderia levar a condições perigosas de operação ou até possíveis paradas da planta [36].

Do ponto de vista da reconciliação de dados, a classificação de variáveis é relevante para definir inicialmente quais variáveis farão parte do problema, quais serão reconciliadas e quais serão estimadas. Esse conhecimento é determinante para a formulação da função objetivo. Além disso, no contexto do problema, também é importante ter em mente quais variáveis podem se tornar não-observáveis na ausência de algum dos dados, e quais variáveis limitarão mais o problema. Variáveis que aparecerem mais vezes nas equações de balanço serão mais participativas, influenciando mais no processo de reconciliação. Variáveis não-redundantes podem tornar inviável a formulação

do problema se, por algum motivo, sua medição for interrompida [11, 13, 36].

Algumas metodologias para classificação de variáveis foram publicadas na literatura. Mah [36] desenvolveu um método baseado em grafos onde as unidades do processo são representadas por nós e as correntes por arcos. Outra abordagem baseia-se no ordenamento de matrizes e foi desenvolvida por Crowe et al. [13] para o caso estritamente linear, e estendida por Crowe [11] para o caso bilinear. Esta técnica se vale das redundâncias para separar o problema em duas etapas, sendo a primeira a reconciliação das variáveis medidas e a segunda a obtenção das variáveis não-medidas.

2.4 Detecção de erros grosseiros

Como já pontuado, o objetivo principal da RD é se valer das informações disponíveis para produzir as melhores estimativas possíveis para as variáveis do processo. Dessa maneira, para que essa técnica seja bem sucedida é importante distinguir, dentre os dados utilizados, quais são de fato representativos dos que não representam a realidade do comportamento do sistema. Estes são ditos erros grosseiros e são normalmente decorrentes de falhas nos instrumentos, no processo ou no modelo matemático. Embora erros grosseiros possam ser indicativos importantes de que algo deve ser revisto no processo ou no modelo, eles não são detectáveis pelas técnicas de retificação tradicionais, e podem prejudicar os resultados obtidos caso sejam incluídos na reconciliação, como será elucidado adiante.

A técnica mais comum utilizada para a detecção de erros grosseiros é o teste estatístico de hipótese. Nele, assume-se que os erros ou resíduos da reconciliação seguem alguma distribuição probabilística conhecida, e testa-se

essa hipótese calculando-se algum parâmetro relevante a partir dos dados disponíveis. Dentre os testes mais tradicionais estão o Teste Global [44], o Teste Nodal [35] e o Teste de Medida [13]. O teste global apenas sinaliza se há a presença de erros grosseiros, enquanto os outros dois também apontam em qual das medições o erro se localiza. Baseando-se nos testes citados, Narasimhan & Mah [39] desenvolveram o teste da Razão Probabilística Generalizada que é vantajoso por ser capaz de diferenciar tipos de erro.

Os testes estatísticos são empregados antes da reconciliação propriamente dita e apontam quais medições devem passar por algum tipo de pré-tratamento antes da próxima etapa. Uma vez que tenha sido detectado algum erro grosseiro, duas estratégias são possíveis: a remoção ou a compensação da medição corrompida. A primeira consiste basicamente em remover a medição do problema, realizar a reconciliação e estimar o seu valor utilizando-se os novos valores das variáveis e os modelos. A segunda consiste em se utilizar de alguma técnica para se estimar a magnitude do erro e assim compensar o valor da variável antes de prosseguir para a reconciliação [7].

Caso haja a presença de um único valor espúrio por medição, os procedimentos descritos acima podem ser realizados sem maiores complicações. Entretanto, se existe a possibilidade de mais de um erro grosseiro, estratégias mais complexas são necessárias para tratá-los. Narasimhan & Jordache [38] destacam três estratégias principais para a detecção múltipla de erros grosseiros: eliminação serial, compensação serial e a compensação simultânea ou coletiva.

A eliminação serial consiste na realização iterativa de testes de hipótese até que todos os erros grosseiros tenham sido detectados. Contudo, essa abordagem pode criar situações onde as medidas removidas se tornam

não-observáveis. A compensação serial é similar à eliminação serial, mas as medições corrompidas são compensadas em vez de eliminadas. Isso garante a redundância do problema, mas torna os resultados da reconciliação dependentes da acurácia do estimador utilizado. Por fim, a compensação simultânea consiste na detecção de todos os erros grosseiros simultaneamente, em uma única iteração.

Além das técnicas sequenciais apresentadas até aqui, outros autores propuseram o uso de estimadores robustos para realizar a detecção de erros grosseiros simultaneamente à reconciliação. Essa abordagem foi proposta inicialmente por Tjoa & Biegler [50] e consiste em se utilizar estimadores robustos para a resolução do problema de reconciliação. Esses estimadores são funções matemáticas capazes de conferir menos peso a observações consideradas enviesadas, e será mais aprofundada adiante. Prata [40] fez uma extensa revisão a respeito do uso de estimadores robustos em problemas de reconciliação. No caso onde há a presença de múltiplos erros grosseiros, essa técnica apresenta vantagens com relação aos métodos sequenciais. Como exposto por Bagajewicz [7], métodos sequenciais podem se tornar computacionalmente custosos na presença de múltiplos erros grosseiros devido ao seu caráter iterativo. Além disso, também existe uma dificuldade inerente a estes métodos em identificar a real localização da medição corrompida em estados estacionários devido à possibilidade de casos degenerados.

2.5 Estratégias de formulação

O problema de RD aplicado à engenharia química foi inicialmente formulado por Kuhen & Davidson [33] como um problema de otimização. Neste

problema a solução ótima corresponde aos valores das variáveis que minimizam a diferença entre os valores medidos e os valores estimados e, simultaneamente, respeitam as equações de balanço. Desde então, diversos métodos diferentes foram propostos para lidar com a RD. Como discutido no Capítulo 1, estes métodos podem ser divididos majoritariamente em três tipos de abordagem: Filtros de Kalman Extendidos, Redes Neurais e Programação Não-Linear [30], cada uma com uma formulação diferente.

O EKF consiste em um algoritmo que combina as medições das variáveis \tilde{x} com estimativas produzidas pelos modelos e a matriz covariância dos erros para gerar uma nova estimativa \hat{x} [54]. Nesta abordagem o modelo dinâmico é linearizado a cada tempo de amostragem e o problema de RD é formulado de forma sequencial e iterativa. A principal vantagem do EKF é ser capaz de fornecer as estimativas com o mínimo de variância possível. No entanto, essa característica só é alcançada em situações onde o modelo se comporta de forma localmente linear e as variáveis e medições seguem a distribuição normal [8]. Na maioria dos processos químicos essas condições normalmente não são atendidas, o que deteriora a performance do algoritmo. Além disso, esse algoritmo também necessita que a matriz correlação dos erros do processo sejam estimadas, o que pode ser uma tarefa difícil.

As RNs consistem em uma função composta por camadas de neurônios onde cada um representa uma função simples com parâmetros próprios que devem ser estimados. Elas vem sendo usadas como alternativa para resolver o problema de retificação desde a década de 1990 [47]. Nesta abordagem, o conhecimento do processo não é utilizado para realizar a reconciliação como no caso da NLP. A resolução do problema de RD usando RNs consiste em se ajustar os parâmetros da rede através de um problema de otimização usando

os dados do processo para obter o estimador que minimize o erro entre os valores reais e os valores estimados. Assim, as medições reconciliadas são obtidas de forma indireta, através de uma função treinada com os dados disponíveis do processo. A grande vantagem das Redes Neurais é que, uma vez treinada, a estimativa é feita com baixíssimo esforço computacional. Contudo, podem ser necessárias grandes quantidades de dados de boa qualidade para treinar a rede[21, 47].

Por fim, a programação não-linear consiste em resolver o problema de otimização proposto por Kuhen & Davidson [33] usando diferentes estratégias para acomodar a dinâmica e restrições de processo. Essa foi a estratégia mais estudada até hoje para resolver o problema de RD e foram atingidos bons resultados para diversos casos. Sua principal desvantagem ainda é o elevado tempo de cálculo que dificulta sua implementação *on-line* [10, 40].

Antes de apresentar uma abordagem para o problema de RD usando Redes Neurais é fundamental, avaliar mais de perto como funciona esta técnica e qual o seu panorama atual. Contudo, também é interessante elucidar a formulação clássica do problema, baseada em otimização e programação não-linear, pois os trabalhos publicados nessa linha apresentam resultados interessantes e trazem valiosas contribuições para a implementação proposta.

2.6 Formulação clássica

2.6.1 O problema de otimização

O problema clássico de reconciliação de dados consiste em um problema de otimização onde o objetivo é encontrar o conjunto de valores reconciliados $\hat{\mathbf{x}}$,

parâmetros estimados \underline{p} , e variáveis não medidas \underline{u} que minimizam a função objetivo escolhida considerando o conjunto de medições $\tilde{\underline{x}}$: [10, 16]

$$\min_{\hat{\underline{x}}, \underline{u}, \underline{p}} F(\tilde{\underline{x}}, \hat{\underline{x}}) \quad (2.1)$$

sujeito às restrições:

$$\underline{h}(\hat{\underline{x}}, \underline{u}, \underline{p}) = 0 \quad (2.2)$$

$$\underline{g}(\hat{\underline{x}}, \underline{u}, \underline{p}) \leq 0 \quad (2.3)$$

$$\hat{\underline{x}}^I \leq \underline{x} \leq \hat{\underline{x}}^S \quad (2.4)$$

$$\underline{p}^I \leq \underline{p} \leq \underline{p}^S \quad (2.5)$$

$$\underline{u}^I \leq \underline{u} \leq \underline{u}^S \quad (2.6)$$

Em que \underline{h} e \underline{g} são os vetores que correspondem respectivamente às restrições de igualdade e desigualdade do problema. Os superescritos I e S correspondem, respectivamente, aos limites inferior e superior das variáveis expressos pelas equações 2.4-2.6. Aqui, a escolha da função objetivo F deve considerar a distribuição estatística dos erros [50], como melhor detalhado a seguir.

A função objetivo, inicialmente proposta por Kuhlen & Davidson [33] e majoritariamente utilizada na literatura até a década de 1990, é o estimador de mínimos-quadrados ponderados (MQP):

$$\sum_k \frac{(\hat{x}_k - \tilde{x}_k)^2}{\sigma_k^2} \quad (2.7)$$

Na qual a diferença para cada uma das k variáveis medidas \tilde{x}_k e reconci-

liadas \hat{x}_k corresponde aos erros daquela variável em um dado instante de tempo, e σ_k^2 é variância da k -ésima variável. Em casos onde os erros sejam considerados espacialmente correlacionados, pode-se utilizar a forma:

$$(\hat{\underline{x}} - \tilde{\underline{x}})\underline{\underline{V}}^{-1}(\hat{\underline{x}} - \tilde{\underline{x}}) \quad (2.8)$$

Em que $\hat{\underline{x}}$ e $\tilde{\underline{x}}$ são respectivamente os vetores com as variáveis medidas e reconciliadas e $\underline{\underline{V}}$ é a matriz covariância das medições. Nota-se que a Equação (2.8) reduz-se à Equação (2.7) para o caso em que as variáveis são não correlacionadas.

Esse estimador corresponde ao estimador de máxima verossimilhança para o caso no qual o erro é normalmente distribuído [24]. Contudo, conforme abordado por Tjoa & Biegler [50], para casos em que o erro não siga uma distribuição normal ou não possua média zero, como na presença de *outliers* ou desvios sistemáticos, esse estimador torna-se enviesado. Neste caso, devem-se utilizar estimadores ditos robustos, capazes de associar menos peso a medições enviesadas.

Considerando a formulação matemática acima, a melhor estratégia para resolver o problema de RD depende das restrições impostas que refletem a natureza do sistema - linear, não-linear, dinâmico ou estacionário. A seguir, foi feita uma breve revisão a respeito de como estas restrições foram abordadas até aqui no contexto da engenharia química.

2.6.2 Reconciliação de dados clássica: revisão de trabalhos relevantes

Como já exposto, o problema de RD na engenharia química foi inicialmente resolvido por Kuhen & Davidson [33]. Eles utilizaram a função MQP para reconciliar apenas variáveis medidas em um processo estacionário com restrições lineares. Essa formulação matemática permite que o problema seja resolvido analiticamente usando multiplicadores de Lagrange. Mah et al. [35] usaram a solução analítica de Kuhen e Davidson [33] em um processo onde existiam variáveis e parâmetros observáveis não medidos, estendendo o procedimento para um conjunto incompleto de medidas. Posteriormente, Knepper & Gorman [29] usaram técnicas de linearizações sucessivas para aprimorar a técnica proposta, aplicando-a ao caso com restrições não-lineares.

Uma abordagem diferente foi proposta por Crowe et al. [13] para separar o problema de obtenção das variáveis reconciliadas do problema de inferência das variáveis observáveis. Foi utilizada uma matriz de projeção para separar os balanços de massa de unidades do processo com composições não medidas dos demais balanços. Assim, foi possível dividir o problema linear em um problema de reconciliação seguido por um de estimação de parâmetros. Esta abordagem foi estendida por Crowe [11] para o caso onde existem concentrações medidas em correntes cuja vazão total é desconhecida. Uma vez que a separação dos balanços de massa das unidades com variáveis desconhecidas tenha sido efetuada, aplica-se uma nova matriz de projeção para separar as vazões totais desconhecidas das concentrações conhecidas. As concentrações são então ajustadas iterativamente a partir de estimativas iniciais fornecidos para as vazões desconhecidas. Esta abordagem contempla a solução de problemas de reconciliação bilineares.

A partir da década de 1990, as técnicas de programação não-linear passaram a ser amplamente utilizadas para a resolução do problema de RD, sendo empregadas inclusive em problemas dinâmicos. Tjoa & Biegler [49] utilizaram a colocação ortogonal de elementos finitos e propuseram um método de programação quadrática sequencial para resolver o problema de otimização, atingindo bons resultados com um tempo de cálculo consideravelmente menor. Posteriormente, Liebman et al. [34] e Kim et al. [27] utilizaram técnicas de programação não-linear, combinando colocação não-ortogonal de elementos finitos com uma abordagem de janela móvel de tempo. A estratégia de janela consiste em utilizar uma quantidade fixa de medições em uma janela que é atualizada com os novos dados sempre que novos dados amostrais são disponibilizados. Assim, novos dados são inseridos e dados mais antigos são descartados, reduzindo a extensão do acoplamento temporal e, por conseguinte, o tempo de cálculo. Esta estratégia tem sido utilizada inclusive em trabalhos mais recentes para resolver o problema dinâmico [40, 43].

Contudo, conforme ressaltado por Liebman et al. [34], é importante avaliar o tamanho da janela e a quantidade de instantes de tempo utilizados. Intervalos pequenos podem levar a resultados muito ruidosos e intervalos demasiado grandes aumentam consideravelmente o tempo necessário para a realização dos cálculos. Ramamurthi et al. [43] demonstraram como é possível se promover melhorias na performance de controladores utilizando técnicas de RD. Os autores utilizaram uma técnica de linearização das restrições dentro do horizonte de estimação para acelerar os cálculos e tornar possível a implementação do algoritmo *on-line*.

Além de metodologias aplicadas ao problema de RD dinâmico, também cresceu, no fim do século XX, a quantidade de trabalhos abordando a esti-

mação simultânea de erros grosseiros. Como pontuado na seção de detecção de erros grosseiros, uma das formas de fazer isso é utilizando estimadores robustos. Essa abordagem consiste em utilizar na função objetivo modelos que atribuam menos peso a medições que não sigam uma distribuição estatística proposta.

Tjoa & Biegler [50] utilizaram uma distribuição normal contaminada para contabilizar possíveis *outliers*. Essa técnica apresentou bons resultados, mas presume que a frequência e distribuição dos valores espúrios sejam aproximadamente conhecidas. Albuquerque & Biegler [4] comparam um estimador *Fair* com a normal contaminada, indicando superioridade do primeiro em casos onde há a presença de variáveis cujo erro médio não é zero. Segundo os autores, essa distribuição é capaz de contabilizar os *outliers*, mas assume, assim como o estimador MQP, que a média dos erros é zero, o que frequentemente não é verdade em situações reais. Os autores também ressaltam que os ganhos em se utilizar essa técnica para reconciliar variáveis não-redundantes podem não ser tão significativos.

Como estimadores robustos atribuem pouca importância a informações advindas de observações suspeitas, o valor da variável nesses casos é obtido principalmente das restrições. Assim, em casos onde a variável não pode ser obtida a partir das outras, essa técnica pode não funcionar bem, levando inclusive a problemas de convergência. Liebman et al. [34] propuseram uma abordagem para a detecção simultânea de erros sistemáticos em sensores. Os autores sugeriram modificar a função objetivo incluindo parâmetros a serem estimados para contabilizar os erros. Assim, seria possível resolver o problema de reconciliação, identificar quais sensores possuem desvios sistemáticos e estimar o tamanho desses desvios. No entanto, a presença de

muitos sensores com potenciais desvios sistemáticos nesse panorama significa uma quantidade muito grande de parâmetros a serem estimados, aumentando muito o esforço computacional e levando a potenciais problemas de convergência.

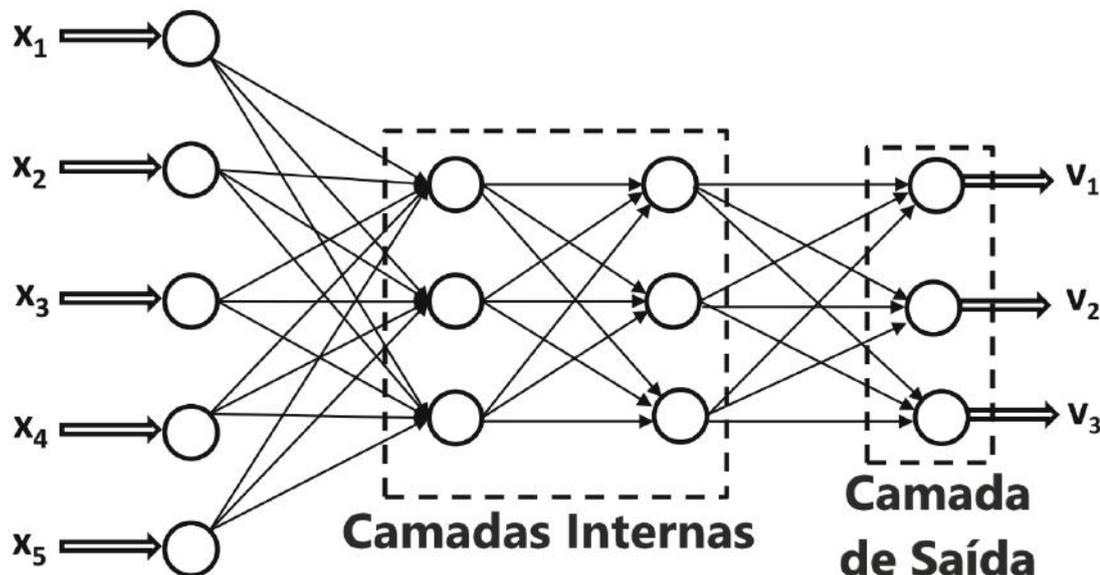
O uso de métodos não-determinísticos para o problema não-linear restrito de RD foi inicialmente abordado por Wongrat et al. [55]. Os autores utilizaram o algoritmo genético para analisar os problemas de descontinuidade e não convexidade das funções robustas.

2.7 Redes Neurais Artificiais

Redes neuronais artificiais são modelos, inspirados no sistema nervoso de animais, que podem ser treinados para estimar estados e reconhecer padrões. O sistema nervoso é composto por células nervosas, ditas neurônios, as quais se conectam entre si por meio de dendritos e axônios. Os dendritos são prolongamentos presentes na célula, por onde os estímulos nervosos são recebidos. O axônio é o prolongamento celular por onde os estímulos são enviados. Um neurônio normalmente possui alguns dendritos, mas somente um axônio. Dessa forma, o axônio do neurônio transmissor se conecta aos dendritos dos neurônios receptores e assim por diante, formando as redes neuronais que compõem o sistema nervoso [3]

Analogamente, redes neuronais artificiais também são compostas por células que recebem entradas de dados, realizam cálculos e retornam valores que podem ser enviados aos próximos neurônios para que sejam realizadas novas operações.

Figura 2.2: Esquema de uma rede neuronal *feedforward*.



Fonte: Adaptado de Aggarwal [3].

A figura 2.2 representa uma RN do tipo *feedforward*. Observa-se que a rede possui uma camada de entrada por onde entram os dados, camadas internas onde esses dados são processados e, por fim, uma camada de saída que processa os dados recebidos da última camada interna, para que sejam obtidos resultados no formato desejado. Uma rede pode possuir tantas camadas internas quanto for desejado, e cada uma delas pode possuir tantos neurônios quanto necessário. Assim, apenas as camadas de entrada e saída têm o formato restrito à quantidade de variáveis que entram e saem da rede.

Cada informação que trafega entre dois neurônios possui um peso associado w_{mn} , e cada neurônio possui um *bias* b_n , em que n identifica o neurônio em questão e m o neurônio da camada anterior que envia a informação associada ao peso w_{mn} . Além disso, os neurônios também possuem funções de ativação que transformam as informações recebidas nos valores enviados para

a próxima camada. A representação matemática da resposta de um neurônio em função das entradas é dada por:

$$S_n(\underline{x}) = f\left(\sum_m w_{mn}x_m + b_n\right) \quad (2.9)$$

Em que S_n é a saída do n ésimo neurônio e \underline{x} compreende os m sinais recebidos por ele, provenientes da camada anterior da rede.

Existem alguns exemplos de funções que podem ser empregadas como função de ativação. Dentre elas, a mais tradicional é a sigmoide [14]. Essa função é caracterizada por possuir o formato de uma curva em S e por mapear o domínio de todos os números reais para o intervalo $[0,1]$.

$$f(x) = \frac{1}{1 + e^{-x}} \quad (2.10)$$

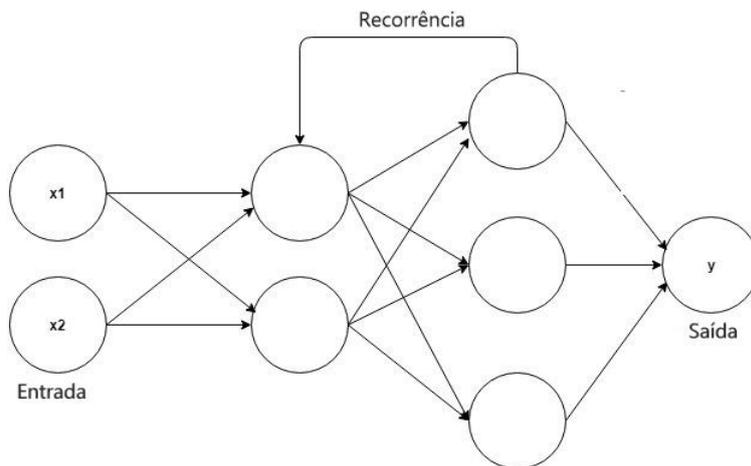
Para uma rede com múltiplas camadas, as variáveis de saída são obtidas através da propagação da informação ao longo das camadas, segundo a Equação 2.11.

$$\hat{y}(\underline{x}) = f^L(\underline{W}^L f^{L-1}(\underline{W}^{L-1} f^{L-2}(\dots f^1(\underline{W}^1 \underline{x} + \underline{b}^1)\dots) + \underline{b}^{L-1}) + \underline{b}^L) \quad (2.11)$$

Em que L é o número de cada camada e f^L é a função de ativação dos neurônios da camada L , \underline{W}^L é a matriz com os pesos dos neurônios da camada L , e \underline{b}^L o vetor com os *bias* dos neurônios da camada L .

É bastante comum que uma RN com múltiplas camadas use funções de ativação diferentes para cada camada visando capturar eficientemente todas as facetas de fenômenos complexos. Além das redes do tipo *feedforward*, onde a informação é propagada sempre no sentido entrada-saída da rede, vale a pena destacar uma outra topologia utilizada para RNs: redes ditas recorrentes [25, 26], mostrada na figura 2.3. Nesta topologia, a informação proveniente da camada de saída ou de alguma das camadas internas pode ser reinserida na rede e o *output* é obtido de forma iterativa.

Figura 2.3: Esquema de uma rede neuronal com recorrências.



Fonte: produzido pelo autor.

Dentro do panorama da engenharia química, Himmelblau [21] publicou uma revisão a respeito das possibilidades de aplicação para este modelo. Além da reconciliação de dados, destacam-se o uso desse modelo para controle de processos [41], predição de falhas em equipamentos, e modelagem de processos com parâmetros não-observáveis. Um exemplo deste último caso é a previsão do índice de fluidez de polímeros, ou modelagem de parâmetros reacionais [42].

2.7.1 Treinando a rede

Como exposto acima, uma rede é capaz de aprender padrões, sendo capaz de modelar fenômenos complexos e fortemente não lineares, característica que torna sua aplicação interessante no campo da engenharia química. No entanto, para que uma rede seja um modelo fidedigno do sistema de interesse, ela deve ser treinada a partir de dados representativos do sistema para que se obtenham valores ótimos dos seus parâmetros w_{mn} e b_n .

No processo de treinamento, inicialmente uma topologia para a RN deve ser definida: número de entradas, número de saídas, número de camadas internas, quantidade de neurônios das camadas internas, função de ativação de cada camada, presença ou não de recorrência, entre outros fatores. Uma vez que a topologia tenha sido definida, é resolvido um problema de otimização onde o vetor solução são os parâmetros da rede que minimizam uma função erro definida. A função erro tipicamente calcula a diferença entre as previsões realizadas pela rede e os valores reais das variáveis, frequentemente chamados de *targets*, mas sua morfologia pode variar com o tipo de padrão que se deseja ensinar para a rede. Em problemas de regressão, onde as variáveis estimadas pela rede são os valores numéricos das variáveis, a função erro de mínimos quadrados é amplamente utilizada (Equação 2.12). Em problemas de classificação, onde a resposta da rede é a probabilidade de se obter um determinado estado, destaca-se o uso da função perda logarítmica (Equação 2.13) [19].

$$J(\underline{\hat{y}}) = \sum_k (y_k - \hat{y}_k)^2 \quad (2.12)$$

$$J(\underline{\hat{y}}) = - \sum_k y_k \log(\hat{y}_k) \quad (2.13)$$

Exemplificando, em um problema de regressão onde deseja-se estimar as variáveis \hat{y} a partir das variáveis \hat{x} usando-se uma rede neuronal, o treinamento da rede é realizado a partir da minimização da seguinte função objetivo J :

$$\min J(\underline{w}, \underline{x}, \underline{b}) = \sum_k (y_k - \hat{y}_k(\underline{w}, \underline{x}, \underline{b}))^2 \quad (2.14)$$

Em que \underline{x} é o vetor com as variáveis de entrada da rede, $\underline{\hat{y}}$ o vetor variáveis de saída, \underline{y} o vetor de valores reais das variáveis que se deseja estimar, e \underline{w} e \underline{b} os vetores com os parâmetros ótimos que se deseja obter. O conjunto de dados em questão é então utilizado para realizar a avaliação da função objetivo em diferentes pontos. Apesar deste procedimento necessitar que a rede possua uma topologia definida, é importante realizá-lo para várias arquiteturas diferentes a fim de descobrir quais hiperparâmetros são os mais adequados. Estes podem ser definidos como os parâmetros do problema que controlam o processo de aprendizado e, portanto, precisam ser definidos previamente ao treinamento da rede. Alguns exemplos de hiperparâmetros são o número de camadas da rede e o número de neurônios de cada camada [3].

É importante ressaltar que esse algoritmo só é capaz de aprender padrões que estejam bem representados nos dados fornecidos. Na maioria dos casos, a rede não é capaz de extrapolar com confiabilidade fora da faixa das variáveis que foram usadas no treinamento. Além disso, a eficiência do algoritmo também está intimamente ligada à qualidade dos dados de treinamento. Uma rede treinada com dados pouco confiáveis não será capaz de fornecer estima-

tivas confiáveis.

Do ponto de vista da obtenção dos parâmetros, a técnica de otimização mais comumente utilizada é o *backpropagation* [3]. Esta técnica consiste em se calcular o gradiente da função objetivo utilizada com relação aos parâmetros da rede, e usar o seu valor multiplicado por uma taxa de aprendizado η para atualizar os valores. O gradiente pode ser calculado em um ponto específico do conjunto de dados ou em uma amostra de alguns pontos (Equações 2.15 e 2.16).

$$w_{mn}^{t+1} = w_{mn}^t - \eta \left(\frac{\partial J}{\partial w_{mn}} \right) \quad (2.15)$$

$$b_n^{t+1} = b_n^t - \eta \left(\frac{\partial J}{\partial b_n} \right) \quad (2.16)$$

A derivada em questão é calculada aplicando-se a regra da cadeia na Equação 2.11, com o erro sendo propagado ao longo das camadas de neurônios no sentido contrário da informação – daí o nome da técnica [3].

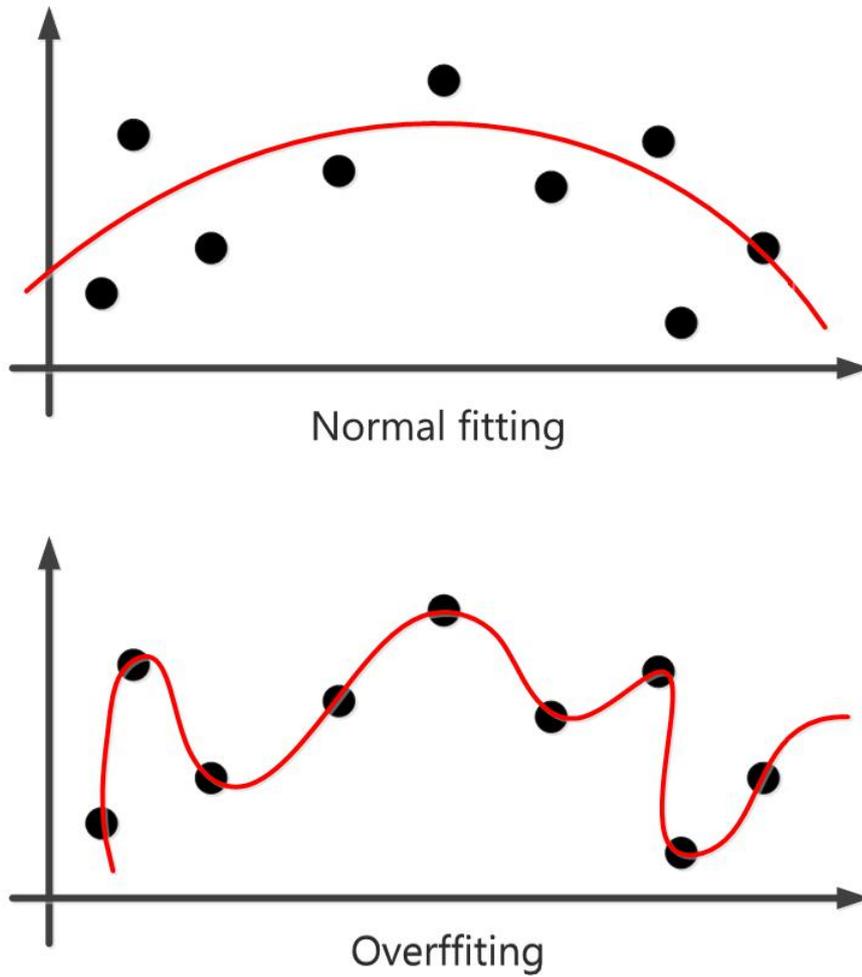
Existem algumas técnicas diferentes para realizar o cálculo do gradiente em questão. A mais simples é o gradiente descendente estocástico, onde um ponto do conjunto de dados é aleatoriamente escolhido para a realização do cálculo. Recentemente, foram desenvolvidas novas e mais complexas técnicas para lidar com dificuldades no problema de otimização como gradientes ruidosos e esparsos [56]. Um exemplo popular na literatura é o método conhecido como ADAM [28], que além do cálculo do gradiente também adapta as taxas de aprendizado de acordo com critérios específicos para tornar a otimização mais eficaz. Vale destacar que, independentemente da técnica utilizada, pode ser necessária uma boa quantidade de avaliação de gradientes

para que o algoritmo chegue a um resultado próximo a um mínimo local, devido à grande quantidade de parâmetros que uma rede pode ter. Devido a este fato, é crucial que se tenha um conjunto de dados com informações suficientes para capturar toda a complexidade do problema em questão, obtendo um bom conjunto de parâmetros para a rede treinada.

Um outro aspecto pertinente a modelos de *machine learning* deve ser evidenciado antes de prosseguirmos: o conceito de *overfitting*. Em estatística, esse conceito pode ser definido como “o desenvolvimento de uma análise que é demasiadamente representativa das particularidades de um determinado conjunto de dados e pode, portanto, falhar ao realizar estimativas fora desse conjunto de dados ou observações futuras” [1]. A figura 2.4 demonstra graficamente no que consiste este fenômeno.

Isso muitas vezes ocorre quando se utiliza um modelo com mais parâmetros do que o necessário para se modelar um dado sistema. Assim, para evitar que isso aconteça, é importante separar uma amostra de dados para testar o algoritmo após o fim do treinamento. Avaliar um algoritmo com um conjunto de dados não utilizado para treiná-lo é uma maneira eficiente de garantir que o mesmo não sofra de *overfitting*.

Figura 2.4: Exemplo de *overfitting*.



Fonte: produzido pelo autor.

2.7.2 O problema de reconciliação usando redes neurais

Assim como o problema clássico apresentado anteriormente, a RD usando redes neurais também consiste em um problema de otimização. No en-

tanto, a sua solução não é um vetor com as variáveis da planta reconciliadas mas sim o vetor de parâmetros ótimos da rede para o conjunto de dados estudado. Ou seja, em vez de se buscar diretamente os valores reconciliados, busca-se a melhor rede capaz de reconciliar os dados naquelas condições.

Do ponto de vista matemático, o problema deixa de ser uma otimização restrita para se tornar uma otimização irrestrita, onde a função objetivo reflete a diferença entre os valores reais das variáveis e os valores estimados pela rede.

$$\min J(\underline{w}, \underline{\tilde{x}}, \underline{b}) = \sum_k (x_k - \hat{x}_k(\underline{w}, \underline{\tilde{x}}, \underline{b}))^2 \quad (2.17)$$

Dessa maneira, buscam-se os parâmetros da rede que minimizam a diferença entre as suas estimativas e os valores reais das variáveis. Embora modelos físico-químicos e equações de balanço não sejam incluídos de forma explícita nessa formulação, uma rede com parâmetros e hiperparâmetros otimizados é capaz de inferir estas relações a partir dos dados fornecidos. Todavia, existem estratégias, como a penalização da função objetivo, que foram utilizadas na literatura para incluir os balanços de forma explícita no problema [15].

$$\min J(\underline{w}, \underline{\tilde{x}}, \underline{b}) = \sum_k (x_k - \hat{x}_k(\underline{w}, \underline{\tilde{x}}, \underline{b}))^2 + \underline{\underline{A}} \hat{\underline{x}}(\underline{w}, \underline{\tilde{x}}, \underline{b}) \quad (2.18)$$

Em que $\underline{\underline{A}}$ é a matriz com os coeficientes referentes aos balanços, penalizando a função objetivo a partir dos desbalanceamentos gerados pelas estimativas da rede.

Conforme já apontado, essa formulação do problema tem como vantagem

um baixíssimo esforço computacional para a obtenção dos valores reconciliados, uma vez que a rede tenha sido treinada. Além disso, alguma robustez foi reportada ao usar redes neurais para reconciliar dados com presença de valores espúrios e desvios sistemáticos [8, 47]. No entanto, uma vez que os modelos são inferidos pela rede a partir dos dados, caso haja a suspeita de que as condições de operação da planta possam ter sido alteradas, é necessário retreinar a rede com dados representativos dessa nova realidade. Por fim, um dos principais desafios essa formulação consiste na necessidade de se adaptar a função objetivo originalmente utilizada no treinamento de redes neurais devido à ausência de dados que representem o valor verdadeiro das variáveis estimadas. Como veremos a seguir, algumas estratégias diferentes foram apresentadas na literatura para lidar com a ausência destes valores. Bons resultados foram obtidos por Bai et al. [8] em uma coluna de destilação simulada. Por ter sido utilizado um sistema simulado, os autores foram capazes de treinar a rede utilizando dados perturbados e em seguida avaliar os resultados usando os valores “reais” da simulação.

2.7.3 Redes neurais aplicadas à reconciliação: revisão de trabalhos relevantes

Os primeiros trabalhos encontrados na literatura onde RNs foram utilizadas para resolver o problema de reconciliação de dados datam do início da década de 1990.

Karjala et al. [25] utilizaram uma rede recorrente para retificar dados com ruídos gaussianos em um tanque de armazenamento. Para contornar a ausência de dados verdadeiros das variáveis os autores usaram uma estratégia

onde a rede era usada para prever o estado do sistema um instante de tempo depois. Os dados perturbados do próximo instante de tempo foram dados como *target* para a rede. Mostrou-se que a rede foi capaz de reconciliar a altura do tanque com razoável eficiência, mas não a vazão de entrada, cujo sinal se tornou ainda mais ruidoso. Esse problema foi posteriormente revisitado pelos autores Himmelblau & Karjala [22] com a inclusão de erros grosseiros que consistiam de *outliers*. Novamente, o algoritmo apresentou bom desempenho bem para reconciliar a altura do tanque, mas tornou a vazão de entrada mais ruidosa.

Karjala & Himmelblau [26] também avaliaram o uso da mesma técnica em um reator CSTR não-isotérmico dinâmico. Os resultados da rede recorrente foram comparados com um EKF e com os resultados obtidos por Liebman et al. [34] para o mesmo sistema. Os resultados sugerem que a rede apresentou um desempenho ligeiramente melhor do que os outros dois algoritmos. Os autores também ressaltam que em casos onde a acurácia dos modelos não é tão alta, a rede tem um desempenho consideravelmente melhor do que os outros algoritmos, por inferir as estimativas diretamente dos dados.

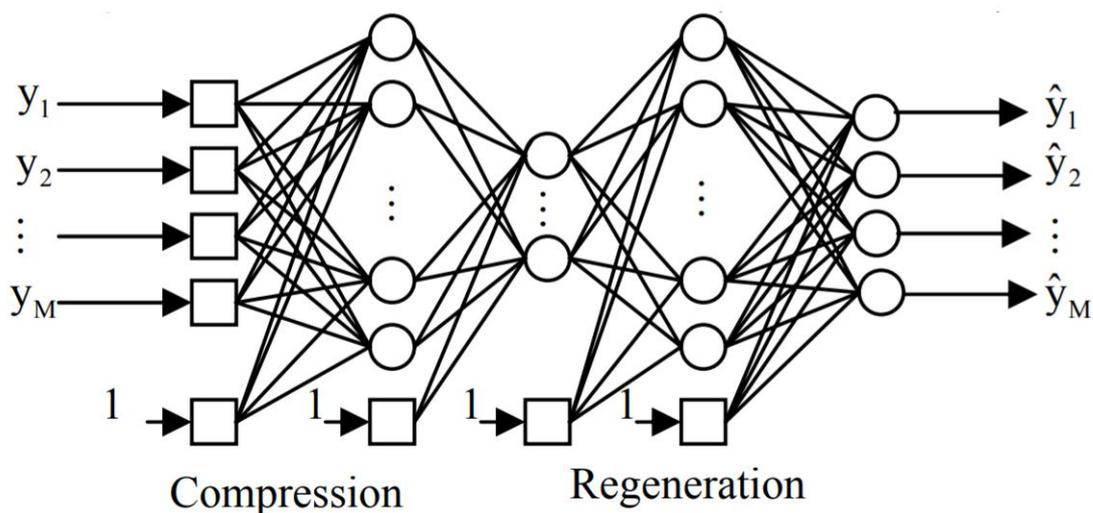
Terry & Himmelblau [47] utilizaram uma rede *feedforward* para reconciliar dados em um trocador de calor simulado em estado estacionário. Para lidar com a falta de *targets*, a rede foi treinada iterativamente com os resultados obtidos em uma iteração i sendo utilizados como *target* na iteração $i + 1$. Na primeira iteração foi utilizada a média das variáveis de entrada. A rede foi treinada até que o valor da função atingisse um critério de convergência. Foram realizadas simulações considerando a presença apenas de erros aleatórios e de erros aleatórios junto com erros grosseiros. O desempenho do algoritmo foi comparado com uma técnica de reconciliação clássica

via programação não-linear. Os resultados sugerem que o desempenho da programação não-linear é levemente superior na presença de apenas erros aleatórios, mas que a rede apresenta melhor desempenho na presença de dados com valores espúrios e erros grosseiros. Caso seja utilizada uma técnica de remoção de *outliers* previamente à reconciliação, o desempenho dos dois algoritmos é equiparável.

O artigo levanta a suspeita de que esse resultado é proveniente da atuação das camadas internas da rede como um filtro que atua de forma similar a uma análise de componentes principais (PCA) não-linear. Essa característica foi inicialmente investigada por Kramer [32]. Ele sugere que redes neuronais podem funcionar decompondo os sinais das variáveis de entrada em componentes principais não-lineares na camada interna para depois reconstruí-los na próxima camada, prevenindo assim que os resultados sejam enviesados por valores espúrios.

Kramer [31] propôs o uso de redes neuronais autoassociativas para a resolução do problema de RD como uma espécie de reconciliação via PCA não-linear [32]. Este algoritmo consiste em uma RN *feedforward* com uma topologia específica capaz de mapear os inputs para o espaço dos componentes principais não-lineares e em seguida reconstruí-los, gerando valores a partir de combinações não-lineares dos componentes principais e, portanto, mais próximos dos valores verdadeiros. A rede em questão é composta tipicamente por uma camada de entrada, três camadas internas, e uma camada de saída com o mesmo número de neurônios da camada de entrada. A primeira camada é alimentada com as medições do processo, e portanto possui dimensão igual a do vetor de medidas. A camada de saída produz uma versão filtrada das medições e, normalmente, de mesma dimensão. As três

Figura 2.5: Topologia de uma rede neuronal autoassociativa.



Fonte: Bai et al. [9].

camadas internas são denominadas respectivamente camada de mapeamento (*mapping*), gargalo (*bottleneck*) e desmapeamento (*demapping*). A camada gargalo deve possuir uma quantidade significativamente menor de neurônios do que as outras. Segundo o autor, a redução do número de neurônios da camada de mapeamento para a camada gargalo faz as informações provenientes dos dados serem comprimidas, decompondo os dados em componentes principais não-lineares. Este processo permite que a rede filtre ruídos e outras informações que não contribuem para a modelagem do fenômeno que se deseja descrever. Destaca-se que o número de neurônios das camadas de mapeamento e desmapeamento deve ser grande o suficiente para garantir uma boa performance sem causar *overfitting*. Por outro lado, a camada gargalo deve possuir o menor número de neurônios possível para que possa comprimir os dados sem provocar perda de informação. A topologia da rede pode ser melhor visualizada na figura 2.5

Além disso, Kramer sugere que as AANNs podem ser vistas como a combinação de duas redes em série com uma camada interna cada [31]. A primeira delas, composta pelas camadas de entrada, mapeamento e gargalo, projeta os dados de entrada para um espaço de menor dimensão por meio de uma transformação não-linear. Na sequência, a segunda rede, composta pelas camadas de desmapeamento e saída, reconstrói o sinal a partir desse espaço de menor dimensão, filtrando os ruídos. Como apontado acima, este procedimento é análogo ao que ocorre na análise de componentes principais, onde as variáveis são projetadas por meio de uma transformação linear em um espaço de menor dimensão, e então reconstruídas a partir dele [32].

Este algoritmo foi testado pelo autor para reconciliar os valores de temperatura em 5 pratos distintos de uma coluna de destilação com 50 pratos [31]. O autor descreve dois procedimentos diferentes para se treinar a rede: um a ser usado na presença de erros grosseiros, e outro na ausência deles. Portanto, esta abordagem apresenta a desvantagem de tornar necessária a identificação prévia de erros grosseiros, o que implica na utilização de alguns métodos descritos. Contudo, o autor demonstra que, uma vez que a rede tenha sido treinada, bons resultados para a reconciliação podem ser obtidos mesmo na presença destes erros. Também é descrito um procedimento para a estimação de parâmetros e dados faltantes.

Du et al. [15] foram os primeiros autores a considerar o uso das equações de balanço explicitamente ao usar RNs para resolver o problema de RD. Os autores destacam que, embora os trabalhos anteriores mostrem que RNs são capazes de diminuir a variância das variáveis de processo, não há garantia de que as variáveis reconciliadas atendam melhor às equações de balanço. No intuito de melhorar este aspecto do algoritmo, o trabalho propõe uma abor-

dagem onde as equações de balanço são inseridas como penalização na função objetivo, de forma a forçar que os valores reconciliados atendam aos modelos. Os autores adotaram uma AANN *feedforward* para reconciliar dados de um processo de processamento de minerais. A rede foi treinada considerando-se os balanços de massa como penalização na função objetivo. A nova técnica foi denominada rede neuronal autoassociativa baseada em sistema (SBANN). O algoritmo foi comparado com uma AANN tradicional.

Os resultados mostram que os valores obtidos pela nova abordagem são melhores do que os obtidos por uma rede autoassociativa treinada sem considerar os balanços explicitamente. Na presença de apenas erros aleatórios, a nova técnica apresentou desempenho ligeiramente melhor mas quando foram adicionados erros grosseiros, resultados bem mais robustos foram obtidos pela abordagem desenvolvida. A SBANN também se mostrou capaz de estimar variáveis observáveis não-medidas com razoável assertividade.

Bai et al. [9] usaram a mesma estratégia de Du et al. [15] para o problema de RD dinâmico. Eles incluíram as equações de balanço na função objetivo usando a discretização das derivadas temporais. O algoritmo utilizado foi a AANN, mas recorrências foram adicionadas na variável que correspondia à altura do tanque para ajudar a contabilizar a dinâmica do processo. Os autores também utilizaram esta estratégia para reconciliar dados em uma coluna de destilação [8].

Neste caso, devido à dificuldade de se obter modelos fenomenológicos que representassem variáveis como as temperaturas dos pratos em colunas de destilação, os autores utilizaram um modelo *black-box* para penalizar a função objetivo. Os modelos consistiam em redes neurais *feedforward* capazes de estimar valores de variáveis desconhecidas em um instante futuro a partir

das variáveis reconciliadas em um dado instante. Foi utilizada uma rede para reconciliar cada variável. Em ambos os casos os resultados foram promissores, incluindo a reconciliação eficiente de dados em casos onde foram induzidas falhas em sensores.

Capítulo 3

Sistema Estudado

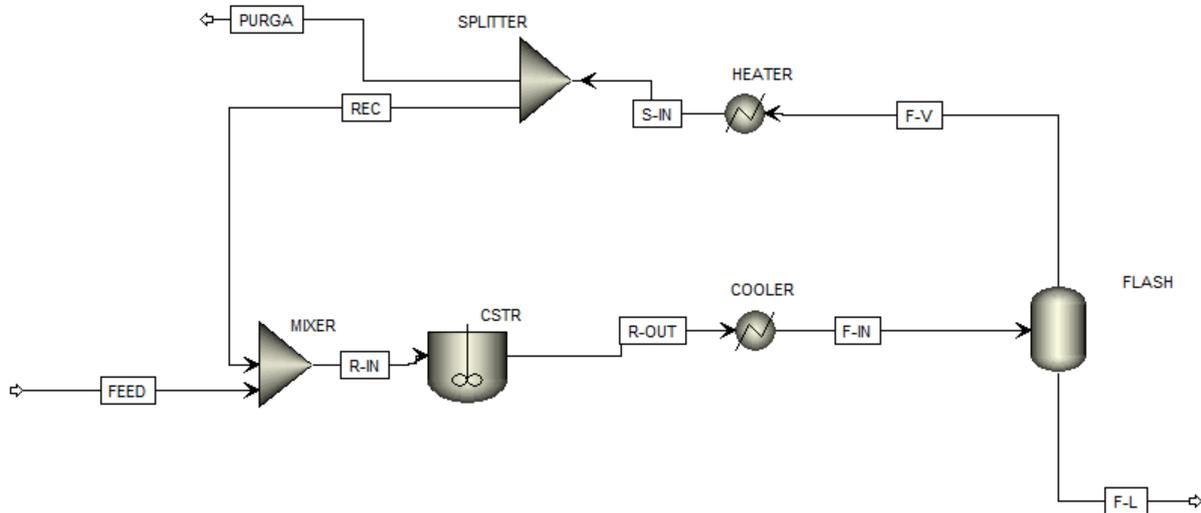
Com o intuito de se investigar desempenho e limitações da aplicação de RNs ao problema de RD, foi realizada a simulação de uma planta química de produção de fenilbenzeno em estado estacionário. A modelagem foi desenvolvida através da implementação das equações de balanço e modelos físico-químicos por meio da linguagem de programação *python* com o auxílio das bibliotecas *scipy* [52] e *numpy* [51].

O processo simulado consiste em um reator CSTR, um tanque de flash, um aquecedor, um resfriador e uma corrente de reciclo. No reator, ocorre a conversão de benzeno em fenilbenzeno além da produção de subprodutos, como será melhor elucidado a seguir. O tanque de flash separa os diferentes componentes com base nos pontos de ebulição. Parte das frações mais leves retornam ao reator por meio de um reciclo. O resfriador e o aquecedor são utilizados para garantir que as correntes de reciclo e saída do reator atinjam as temperaturas de operação desejadas no reator e no tanque de flash, conforme ilustrado na figura 3.1. Considera-se que a temperatura da

corrente de alimentação é a temperatura utilizada no reator.

O processo em questão foi inspirado no trabalho realizado por Galantine et al. [18]. Foram mantidos o mesmo sistema reacional e a mesma sequência reator-separador-reciclo utilizada pelos autores, mas foram variadas as condições operacionais de modo a gerar mais dados para o processo de reconciliação. Esta configuração de processo possui um sistema reacional conhecido, com parâmetros reacionais e termodinâmicos disponíveis na literatura [17, 20]. Contudo, este sistema é governado por equações fenomenológicas não-lineares, representando um certo nível de complexidade matemática. Além disso, a planta escolhida representa um típico processo da indústria química com reação, separação, reciclo e purga, envolvendo os equipamentos típicos: reator, separador (originalmente uma destilação, mas simplificado para um flash), trocadores de calor, misturadores e divisores de corrente. Estes fatores motivaram a escolha deste processo para a avaliação do desempenho de RNs aplicadas à RD.

Figura 3.1: Fluxograma do processo.



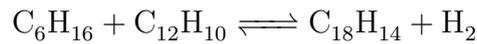
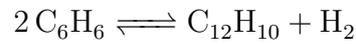
Fonte: produzido pelo autor.

3.1 Descrição da planta e modelagem matemática

Como já exposto, o sistema acima foi estudado em estado estacionário utilizando-se os modelos fenomenológicos disponíveis. Para cada simulação realizada foram definidas a vazão (F_o) e composição ($X_{i,o}$) de entrada da planta, as temperaturas e pressões no reator (P_r , T_r) e tanque de flash (P_F , T_F), e a razão de reciclo (C_s). A partir desses dados, todas as vazões e composições das outras correntes são calculadas. Em todas as simulações realizadas, a composição da corrente de entrada da planta ($X_{i,o}$) foi considerada como sendo benzeno puro ($X_{A,o} = 1.0$).

3.1.1 Reator

A principal reação presente neste sistema é a pirólise do benzeno em fenilbenzeno e gás hidrogênio (reação 1). Uma outra reação não desejada também ocorre em paralelo, gerando meta-terfenilo (reação 2). Ambas as reações são reversíveis [17].



Para a facilitar a representação dos diferentes componentes químicos em fórmulas matemáticas, os mesmos serão identificados como compostos A, B, C e D de acordo com a Tabela 3.1.

Tabela 3.1: Representação dos compostos

Fórmula	Representação
C_6H_6	A
$\text{C}_{12}\text{H}_{10}$	B
H_2	C
$\text{C}_{18}\text{H}_{14}$	D

Fonte: produzido pelo autor.

Para as reações deste processo foi utilizado o modelo de lei de potências com as constantes reacionais.

Realizando-se o balanço de massa por componente, obtemos a Equação 3.1, onde N_i é o número de mols do componente i dentro do reator, $F_{i,Rin}$ e $F_{i,Rout}$ são respectivamente as vazões molares de entrada e saída do compo-

nente i , V_r é o volume do reator, e (r_i) é a taxa volumétrica de produção do componente i no meio reacional.

$$\frac{dN_i}{dt} = F_{i,Rin} - F_{i,Rout} + V_r(r_i) \quad (3.1)$$

Considerando mistura perfeita e estado estacionário dentro do reator, e introduzindo a variável X_i que corresponde à fração molar do componente i , é obtida a Equação 3.2 a partir da Equação 3.1.

$$F_{Rin}X_{i,Rin} - F_{Rout}X_{i,Rout} + V_r(r_i) = 0 \quad (3.2)$$

As reações em questão ocorrem em fase gasosa. Pressão e temperatura são consideradas constantes durante a reação e, devido ao fato de ambas as reações serem equimolares, o volume reacional também pode ser considerado constante. Assim, usando o modelo de lei de potências com as pressões parciais de cada componente ($P_{i,R}$), as taxas (r_i) podem ser escritas de acordo com as Equações 3.4, 3.5, 3.6 e 3.7, em que k_{nd} , k_{nr} são respectivamente as constante de velocidade direta e reversa da reação n .

$$P_{i,R} = P_r X_{i,Rout} \quad (3.3)$$

$$(r_a) = -k_{1d}P_{a,R}^2 + k_{1r}P_{b,R}P_{c,R} - k_{2d}P_{a,R}P_{b,R} + k_{2r}P_{c,R}P_{d,R} \quad (3.4)$$

$$(r_b) = \frac{1}{2}[k_{1d}P_{a,R}^2 - k_{1r}P_{b,R}P_{c,R}] - k_{2d}P_{a,R}P_{b,R} + k_{2r}P_{c,R}P_{d,R} \quad (3.5)$$

$$(r_c) = \frac{1}{2}[k_{1d}P_{a,R}^2 - k_{1r}P_{b,R}P_{c,R}] + k_{2d}P_{a,R}P_{b,R} - k_{2r}P_{c,R}P_{d,R} \quad (3.6)$$

$$(r_d) = k_{2d}P_{a,R}P_{b,R} - k_{2r}P_{c,R}P_{d,R} \quad (3.7)$$

As constantes de velocidade são funções da temperatura T_r e podem ser calculadas usando a lei de Arrhenius (Equação 3.8). Esta depende da energia de ativação da reação E_a e da constante k_o , obtidas de [17] (tabela 3.2).

$$k_{nk} = k_{o(nk)} e^{\frac{-E_a}{RT}} \quad (3.8)$$

Tabela 3.2: Constantes de velocidade das reações de formação de fenilbenzeno ($n = 1$) e meta-terfenilo ($n = 2$).

Reação	$k_o(s^{-1})$	$E_a(cal/mol)$
Direta, $n = 1$	$3,235 \times 10^6$	30190
Reversa, $n = 1$	$1,0205 \times 10^5$	30190
Direta, $n = 2$	$3,7545 \times 10^6$	30190
Reversa, $n = 2$	$7,9544 \times 10^6$	30190

Fonte: produzido pelo autor.

Dessa forma, definindo-se V_r , T_r , P_r , a vazão F_{Rin} e as composições de entrada $X_{i,Rin}$, o sistema estará totalmente especificado. T_r juntamente com as informações da tabela 3.2 nos permite calcular as constantes reacionais. Inserindo as Equações 3.3, 3.4, 3.5, 3.6 e 3.7 nos balanços de massa por componente este conjunto de equações torna-se um sistema não-linear totalmente especificado, onde as incógnitas são $X_{i,Rout}$ e F_{Rout} , e pode ser resolvido por métodos numéricos. No presente trabalho, foi utilizado o método Powell híbrido modificado como implementado no pacote MINPACK [37]. Foi utilizado $V_r = 1m^3$.

3.1.2 Tanque de Flash

No tanque de flash, os componentes são separados a partir do equilíbrio líquido-vapor. A mistura forma duas fases em equilíbrio termodinâmico com

composições diferentes, sendo a fase líquida rica nos componentes menos voláteis e a fase vapor rica nos mais voláteis.

O balanço de massa por componente no tanque de flash pode ser escrito com o auxílio do parâmetro β que representa a fração de vapor no flash com relação à carga abastecida (Equações 3.9 e 3.10). As vazões e composições de entrada, fração de vapor e fração líquida são escritas respectivamente $(F_{Fin}, X_{i,Fin})$, $(F_{FV}, X_{i,FV})$ e $(F_{FL}, X_{i,FL})$.

$$\beta = \frac{F_{FV}}{F_{Fin}} \quad (3.9)$$

$$X_{i,Fin} = \beta X_{i,FV} + (1 - \beta) X_{i,FL} \quad (3.10)$$

A relação entre os componentes em cada fase pode ser obtida a partir de um modelo de equilíbrio líquido-vapor. No presente trabalho foi utilizada a lei de Raoult [45]. Este modelo assume que a fase gasosa tem comportamento ideal, e que a fase líquida forma uma mistura também ideal. O equilíbrio entre as fases pode ser expresso pela Equação 3.11 para cada componente i , em que P_F é a pressão total dentro do tanque, e $P_{i,F}^{sat}$ é a pressão de saturação do componente i na temperatura T_F .

$$P_F X_{i,FV} = P_{i,F}^{sat} X_{i,FL} \quad (3.11)$$

$P_{i,F}^{sat}$ é frequentemente calculado através de correlações semi-empíricas que estabelecem $P^{sat} = f(T)$. No presente trabalho foram utilizados os modelos de Antoine [48] para os compostos A, B e C, e a equação de Wagner [53] para o composto D (Equações 3.12 e 3.13).

$$\ln \frac{P^{sat}}{P_c} = a_1 - \frac{a_2}{T + a_3} \quad (3.12)$$

$$\ln \frac{P^{sat}}{1000} = a \ln(T) + \frac{b}{T} + c + dT^2 \quad (3.13)$$

Os parâmetros para a Equação 3.12 foram obtidos do livro de Henley e Seader [20] e estão dispostos na Tabela 3.3. Os parâmetros para a equação 3.13 foram obtidos da base de dados do software EMSO [46] e estão dispostos na Tabela 3.4.

Tabela 3.3: Parâmetros usados na equação de Antoine [20]

Componente	a_1	a_2	a_3	P_c
A	5.658375	5307.813	379.456	714.2
B	6.194778	7947.647	317.1246	557.0
C	5.602657	418.1773	474.214	190.8

Fonte: Henley & Seader [20].

Tabela 3.4: Parâmetros usados na equação de Wagner.

Componente	a	b	c	d
D	-14.7697	-15484.2	122.524	3.7852x10 ⁻⁶

Fonte: Dados do software EMSO [46].

Combinando as Equações 3.11 e 3.10 com os somatórios das frações das fases (Equações 3.16 e 3.17), obtemos a equação de Rachford-Rice (Equação 3.14), em que K_i é a proporção entre as frações do elemento i na fase vapor e líquida (Equação 3.15).

$$\sum_i \frac{X_{i,Fin} (K_i - 1)}{1 + \beta (K_i - 1)} = 0 \quad (3.14)$$

$$K_i = \frac{X_{i,FV}}{X_{i,FL}} = \frac{P_{i,F}^{sat}}{P_F} \quad (3.15)$$

$$\sum_i X_{i,FV} = 1 \quad (3.16)$$

$$\sum_i X_{i,FL} = 1 \quad (3.17)$$

Quando P_F , T_F , F_{Fin} e X_{Fin} estão definidos, o flash está totalmente especificado. Este problema é conhecido como flash P,T [45] e é solucionado usando-se um método numérico para resolver a Equação 3.14, obter β e, em seguida, as vazões e frações molares. No presente trabalho foi utilizado o método da secante [6].

3.1.3 Conectores

O processo possui dois conectores, um *mixer* e um *splitter*. O *mixer* mistura as corrente de entrada e reciclo, caso algo seja recirculado no flash antes da entrada no reator. Mistura perfeita é assumida no equipamento. O *splitter* determina a quantidade da corrente leve do flash que será purgada. O parâmetro C_s utilizado representa exatamente a fração da corrente que chega ao *splitter* que é recirculada - Equação 3.18.

$$C_s = \frac{F_{rec}}{F_{FV}} \quad (3.18)$$

3.1.4 Aquecedor e resfriador

O aquecedor e o resfriador são utilizados para garantir que as temperaturas de operação no reator e no flash sejam as desejadas. Os balanços de energia

e cargas térmicas necessárias não foram realizados, pois considera-se que os equipamentos são capazes de levar a corrente FV até $T = T_r$ e a corrente $Rout$ até $T = T_F$ - figura 3.1.

3.2 Faixa de operação

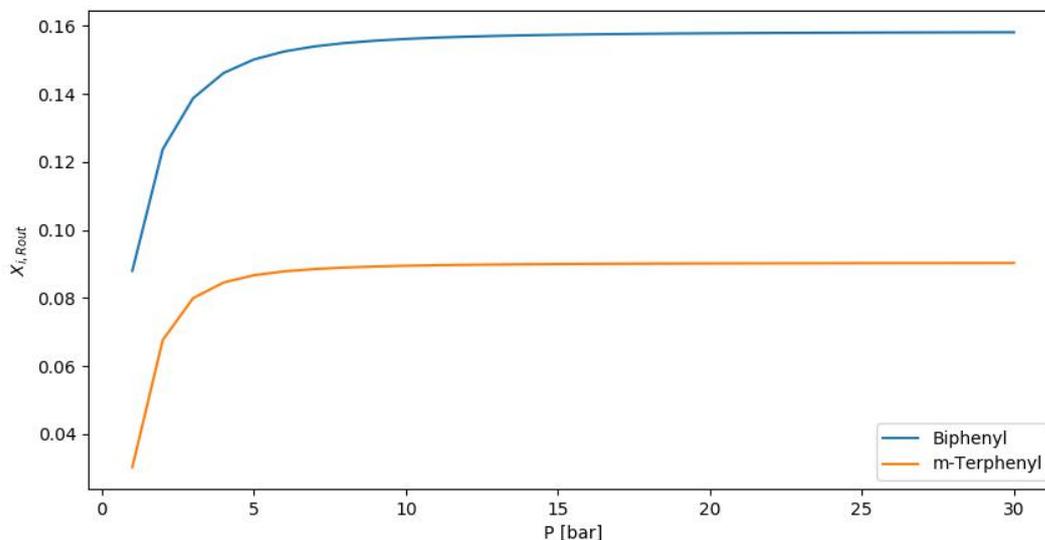
Antes de realizar as simulações necessárias para produzir os dados a serem estudados, é importante avaliar o sistema para garantir que a faixa de operação adotada é representativa das premissas adotadas na escolha dos modelos. Para tal, foi realizada uma análise de sensibilidade nas variáveis de entrada da simulação - F_o , P_r , T_r , P_F e T_F - para definir qual seria a faixa de operação mais apropriada e garantir que esta não viole as premissas básicas adotadas na modelagem do processo.

Antes de prosseguir, vale esclarecer que o objetivo desta sessão é garantir que a faixa de operação escolhida é verossímil. A intenção não é determinar a faixa ótima do processo. Assim, a análise aqui é meramente exploratória e busca valores apenas razoáveis para as variáveis do processo.

3.2.1 Pressão no reator

Inicialmente foi observado o efeito da pressão reacional na conversão dos reagentes. Galantine et al. [18] realizaram este estudo partindo de uma temperatura de $700^\circ C$, portanto foi utilizada esta mesma temperatura como ponto de partida. A Figura 3.2 mostra que a conversão de benzeno em m-terphenilo se estabiliza um pouco antes da conversão em biphenil. Contudo, em torno de $12bar$ ambas as frações molares parecem ter se estabilizado.

Figura 3.2: Frações dos componentes B e D na saída do reator a $973K$.



Fonte: produzido pelo autor.

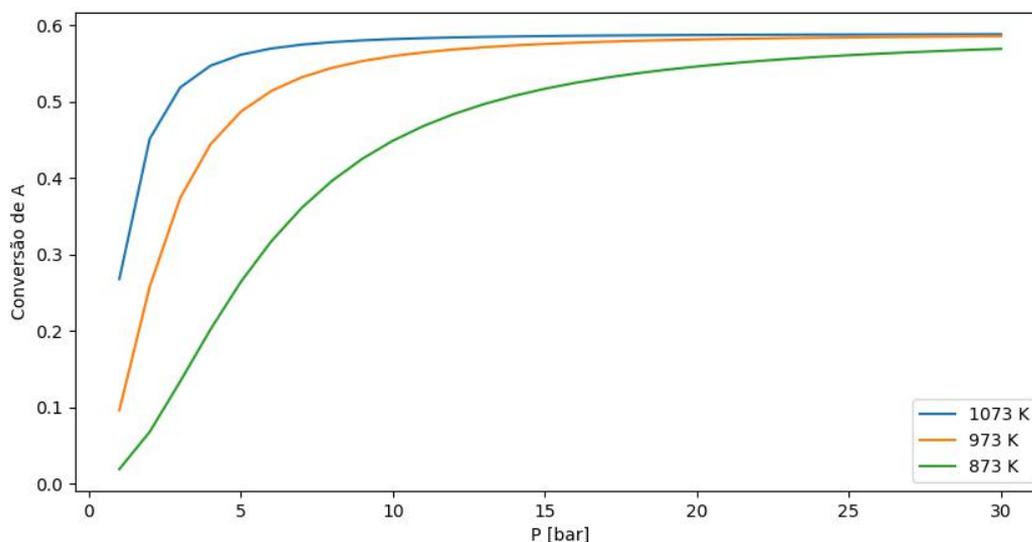
Considerando temperaturas em torno de $700^{\circ}C$, percebemos que temperaturas mais altas fazem com que a conversão de benzeno se estabilize a pressões um pouco mais baixas, e temperaturas mais altas provocam o efeito contrário - Figura 3.3.

É utilizado então $P_r = 12bar$, devido ao fato deste valor apresentar conversões razoáveis para temperaturas em torno de $700^{\circ}C$.

3.2.2 Temperatura no reator

Considerando que a reação é conduzida em fase gasosa, é importante certificar-se de que as condições operacionais dentro do reator não provoquem o surgimento de uma fase líquida. Para tal, foram calculadas as pressões de saturação dos compostos. Pode-se observar que a pressão de saturação do m-

Figura 3.3: Conversão do Benzeno no reator versus P_r .



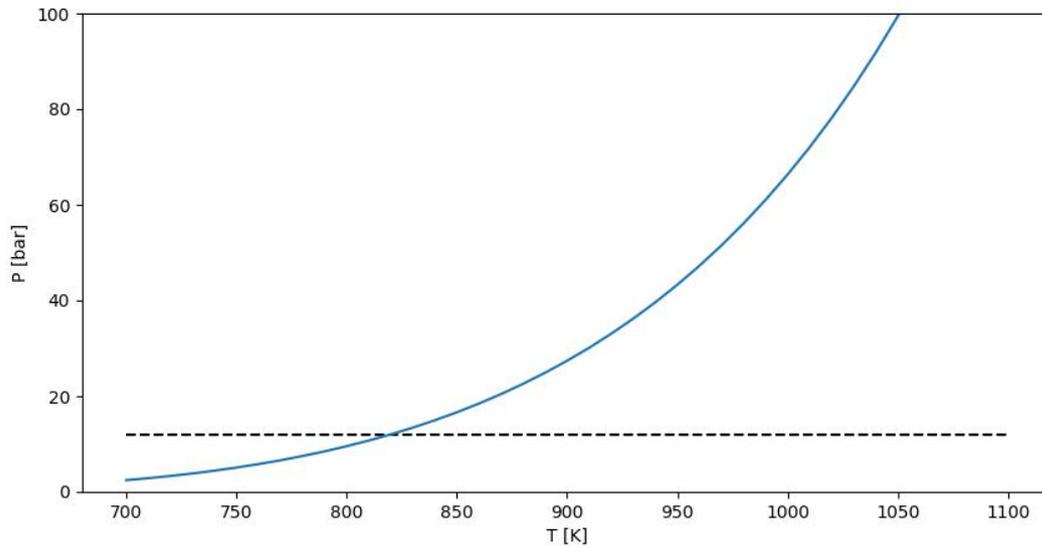
Fonte: produzido pelo autor.

terfenilo, o composto menos volátil presente no meio reacional, atinge 12bar em aproximadamente 820K (Figura 3.4), o que limita o reator a operar acima desta temperatura.

O efeito da temperatura reacional na conversão dos reagentes possui comportamento similar ao da pressão. A Figura 3.5 mostra que a conversão de benzeno em m-terphenilo estabiliza um pouco antes da em biphenil. Contudo, em torno de 1023K (750°C), ambas as frações molares parecem ter estabilizado.

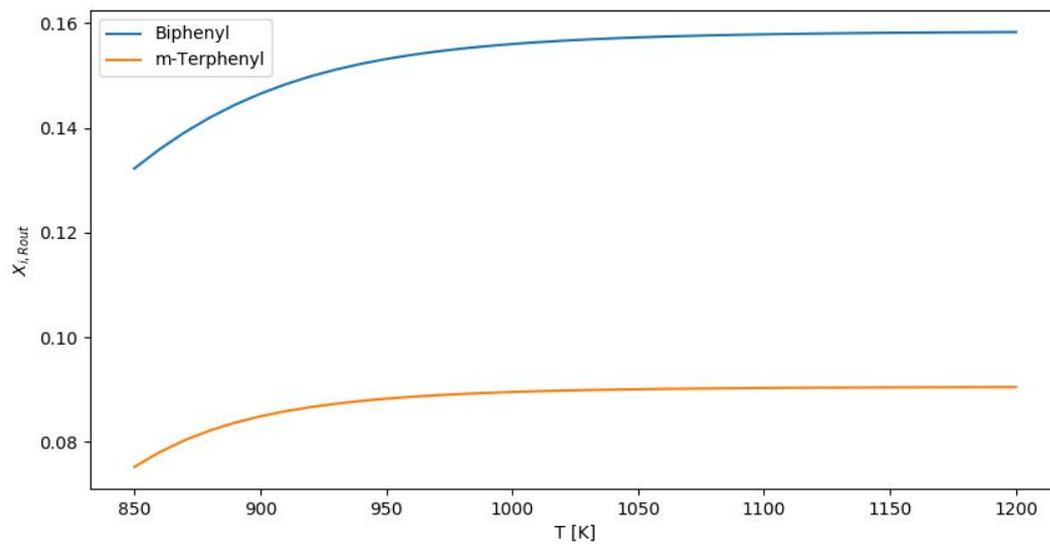
É utilizado então $T_r = 1023\text{K}$, devido ao fato deste valor representar conversões razoáveis nas redondezas de $P = 12\text{bar}$.

Figura 3.4: P_d^{sat} versus T . Linha pontilhada em 12 bar.



Fonte: produzido pelo autor.

Figura 3.5: Frações dos componentes B e D na saída do reator a 12 bar

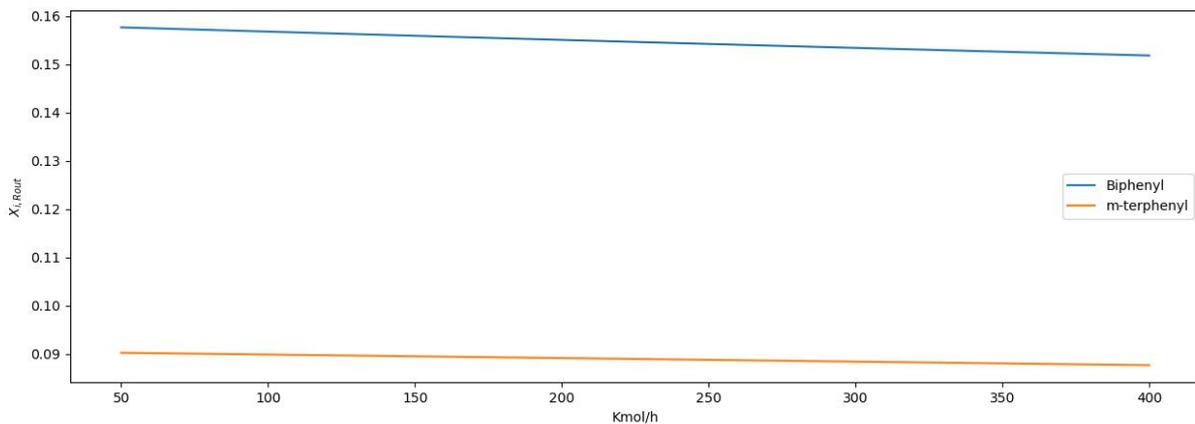


Fonte: produzido pelo autor.

3.2.3 Vazão de alimentação

A vazão de entrada afeta a conversão no reator à medida que afeta o tempo de residência do mesmo. Vazões maiores proporcionam um menor tempo de residência, o que diminui a conversão (Figuras 3.6 e 3.7). No entanto, vazões maiores de reagente também acabam por acarretar em uma maior quantidade de produtos. A Figura 3.6 mostra que a conversão decai lentamente com o aumento de F_o , sendo a formação de biphenil um pouco mais sensível à vazão do que a formação de m-terphenilo.

Figura 3.6: Frações dos componentes B e D versus F_o .



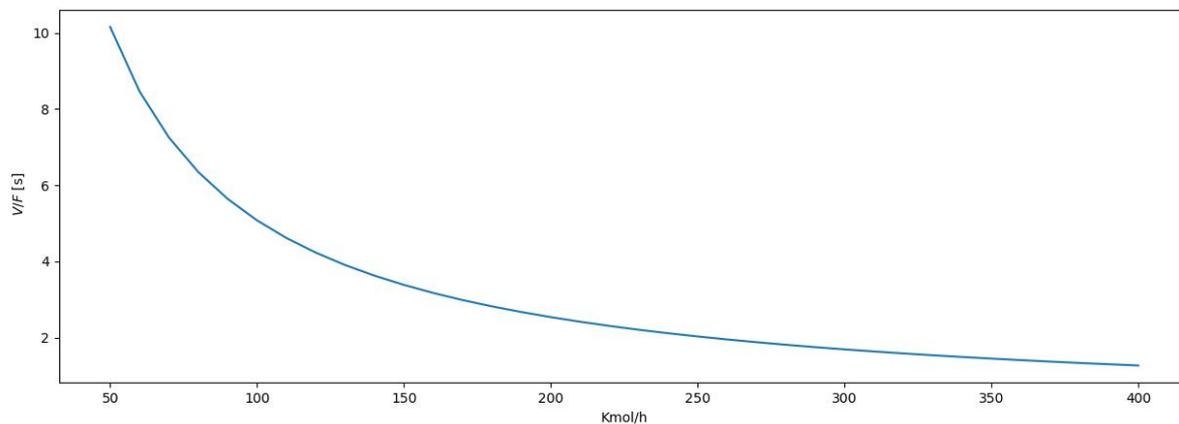
Fonte: produzido pelo autor.

Visando manter maiores conversões, foi escolhido manter a vazão de $100\text{kmol}/h$ usada por Galantine et al. [18] para a alimentação.

3.2.4 Pressão e Temperatura no Flash

O objetivo principal deste equipamento é separar o benzeno dos produtos formados no reator, para que o primeiro possa ser recirculado. Dessa forma,

Figura 3.7: Variação do tempo de residência com F_o .



Fonte: produzido pelo autor.

T_F e P_F são avaliados de acordo com sua capacidade de separar o Benzeno do bifenil.

Para realizar esta análise, foi considerada uma carga equivalente à produzida por um reator operando com $F_{Rin} = 100 \text{ kmol/h}$, $P_r = 12\text{bar}$, $T_r = 1023\text{K}$ e $V_r = 1\text{m}^3$:

$$F_{Rout} = 100,0\text{kmol/h}$$

$$X_{A,Rout} = 0,4234$$

$$X_{B,Rout} = 0,1548$$

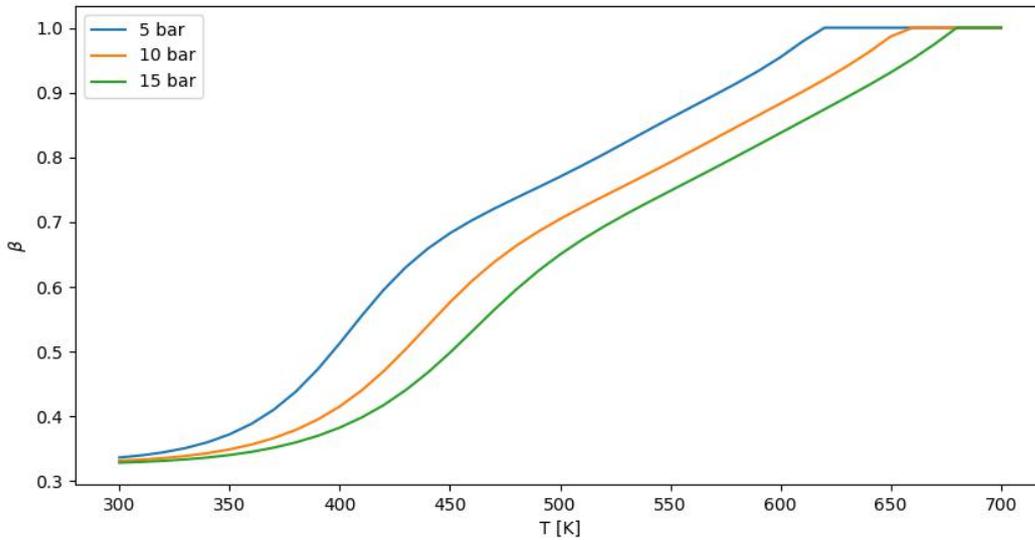
$$X_{C,Rout} = 0,3328$$

$$X_{D,Rout} = 0,0890$$

A Figura 3.8 permite avaliar que, a 300K , a fração de vapor no flash

é cerca 0.33, aproximadamente a fração total de H_2 no equipamento. Em torno de $700K$, praticamente toda a carga do equipamento sai pela corrente gasosa.

Figura 3.8: Influência de T_F e P_F no fator β .



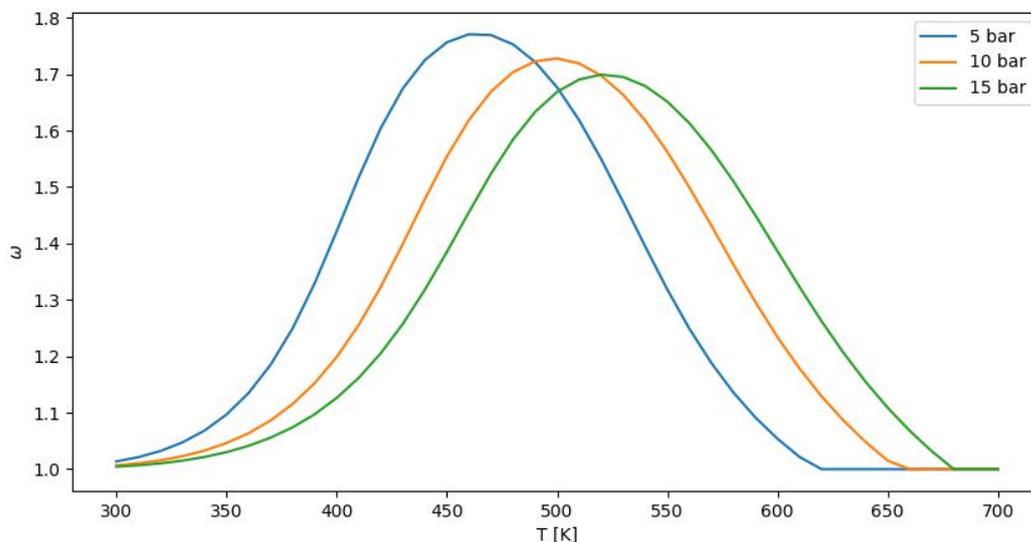
Fonte: produzido pelo autor.

Define-se então o parâmetro ω que corresponde à soma das porcentagens de benzeno na corrente gasosa e biphenil na corrente líquida - Equação 3.19. Este parâmetro é então avaliado com o objetivo de encontrar seu valor máximo no intervalo $300 - 700K$.

$$\omega = \frac{F_{FV} X_{A,FV}}{F_{Fin} X_{A,Fin}} + \frac{F_{FL} X_{B,FL}}{F_{Fin} X_{B,Fin}} \quad (3.19)$$

A Figura 3.9 permite concluir que pressões menores favorecem a separação destes dois componentes e que, a 5 bar, a temperatura ideal é cerca de $T_F = 460K$.

Figura 3.9: Influência de T_F e P_F na separação dos componentes A e B .



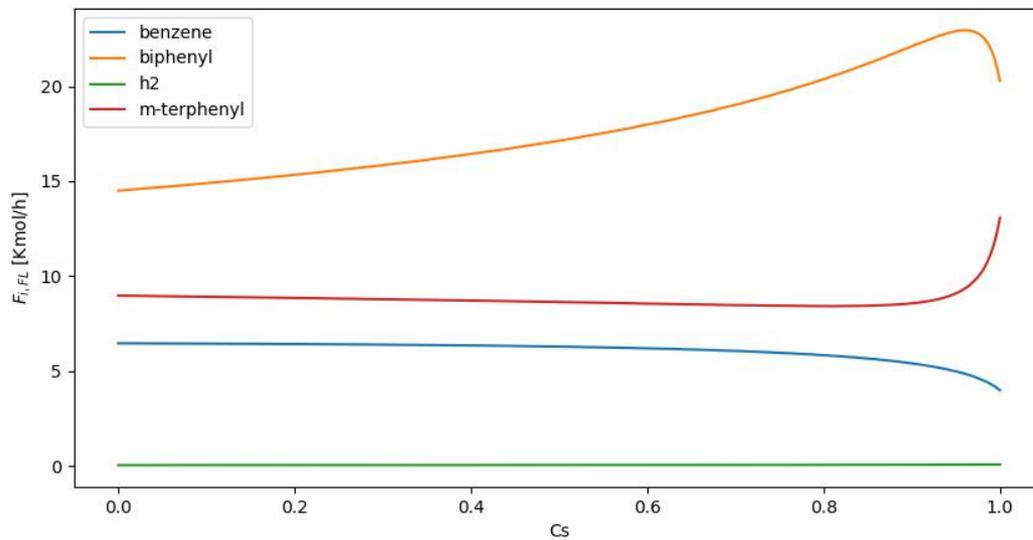
Fonte: produzido pelo autor.

3.2.5 Razão de reciclo

A utilização de uma corrente de reciclo possibilita que parte do benzeno não-reagido seja realimentado ao reator, o que aumenta a quantidade de biphenil produzido no reator e extraído na corrente líquida de flash. A Figura 3.10 mostra que a produção de biphenil aumenta com a razão de reciclo até um determinado ponto, onde esta então decresce acompanhada por um aumento na quantidade de m-terphenilo produzido.

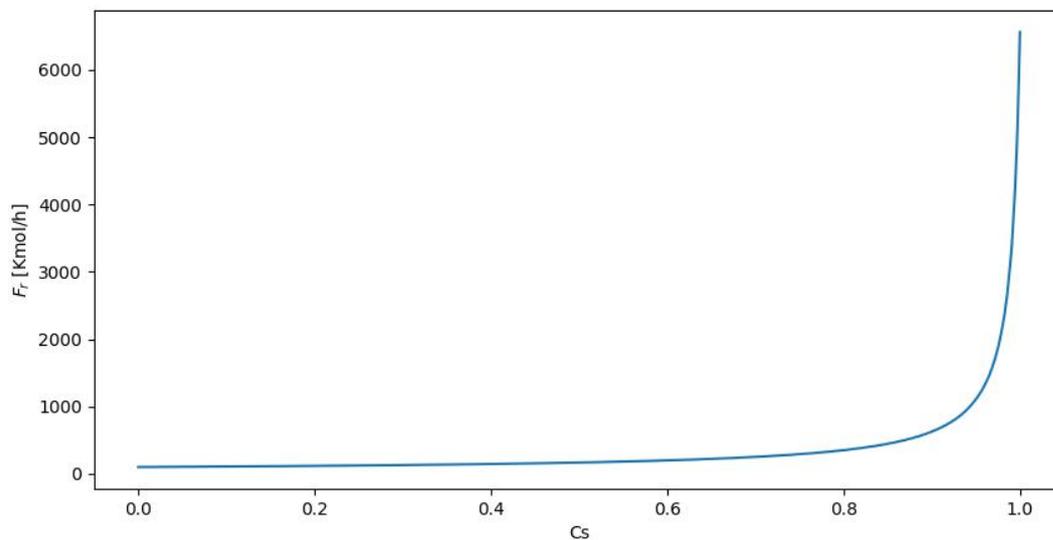
Contudo, a utilização de razões de reciclo muito grandes acarreta em altíssimas vazões de entrada no reator devido à grande recirculação de H_2 , que é totalmente recirculado com o benzeno (Figura 3.11). Em situações práticas, vazões desta magnitude podem acarretar em problemas operacionais. Portanto, a razão de reciclo é mantida em torno de 0,6.

Figura 3.10: Influência de C_s nos produtos obtidos.



Fonte: produzido pelo autor.

Figura 3.11: Influência de C_s na vazão de entrada do reator.



Fonte: produzido pelo autor.

3.3 Validação da Simulação

Com intuito de validar se os resultados obtidos foram fidedignos, foi realizada uma simulação utilizando o software ASPEN HYSYS versão 8.8. A mesma simulação foi então realizada utilizando-se o código desenvolvido. As variáveis de entrada da simulação estão dispostas na Tabela 3.5 e os resultados na Tabela 3.6.

Tabela 3.5: Inputs usados na simulação de validação.

Variáveis de Entrada	
F_o	100,0 <i>kmol/h</i>
$X_{A,o}$	1,0
$X_{B,o}$	0,0
$X_{C,o}$	0,0
$X_{D,o}$	0,0
P_r	10,0 <i>bar</i>
T_r	973 <i>K</i>
P_F	10,0 <i>bar</i>
T_F	473 <i>bar</i>
C_s	0,0

Fonte: produzido pelo autor.

A Tabela 3.6 mostra que a diferença entre os resultados é bastante pequena. A diferença relativa foi calculada com base nos valores obtidos pela simulação do HYSIS e teve valores significativos apenas em casos onde a variável comparada era de magnitude consideravelmente baixa. Com base nestes resultados, o código desenvolvido foi considerado válido.

Tabela 3.6: Comparação entre as simulações.

Variável [kmol/h]	python	HYSYS	Diferença Relativa (%)
$F_{Rout/Fin}$	100,0000	100,0000	0,000
$F_{A,Rout/Fin}$	42,9203	42,8562	0,149
$F_{B,Rout/Fin}$	15,2931	15,3239	-0,201
$F_{C,Rout/Fin}$	32,9555	32,9879	-0,098
$F_{D,Rout/Fin}$	8,8312	8,8320	-0,009
F_{FL}	34,9036	35,2571	-1,013
$F_{A,FL}$	11,6122	11,7152	-0,887
$F_{B,FL}$	14,4048	14,6401	-1,633
$F_{C,FL}$	0,0656	0,0814	-24,094
$F_{D,FL}$	8,8209	8,8204	0,006
F_{FV}	65,0964	64,7429	0,543
$F_{A,FV}$	31,3081	31,1410	0,534
$F_{B,FV}$	0,8883	0,6838	23,020
$F_{C,FV}$	32,8898	32,9065	-0,051
$F_{D,FV}$	0,0102	0,0116	-13,332

Fonte: produzido pelo autor.

Capítulo 4

Formulação do Problema

Como abordado na seção 2.7.2, o uso de RNs para reconciliação de dados consiste em estimar os melhores parâmetros possíveis para a rede resolvendo um problema de otimização. Neste trabalho, este procedimento foi realizado separadamente para o reator e para o flash apresentados no capítulo anterior. Foi utilizada a topologia das redes neuronais auto-associativas [31] devido à sua comprovada capacidade de reduzir ruídos a partir da compressão e descompressão de informação [8, 32]. As redes foram alimentadas com todas as medições representativas do equipamento em questão, para obtenção das mesmas variáveis com um grau menor de ruído. Em particular, este trabalho se dispõe a analisar se equações de balanço podem ser utilizadas para melhorar o desempenho de redes neuronais no problema de reconciliação em estado estacionário.

A seguir, são apresentados mais detalhes a respeito do pré-processamento utilizado para os dados, da escolha da função objetivo, dos hiper-parâmetros do modelo, do método de otimização e das técnicas de regularização utiliza-

das.

4.1 Distribuição dos Dados

Após uma breve análise das variáveis como descrito na seção 3.2, o sistema foi simulado em diferentes condições operacionais. Os diferentes valores utilizados para as variáveis entrada - F_o , P_r , T_r , P_F , T_F e C_s - podem ser observados na Tabela 4.1. Foram geradas 3375 entradas, utilizando todas as combinações possíveis dentre os valores presentes na tabela.

Tabela 4.1: Valores utilizados para as variáveis de entrada.

Variável	Valores
F_o	80,0; 90,0; 100,0; 110,0; 120; [kmol/h]
P_r	10,0; 11,0; 12,0; 13,0; 14,0 [bar]
T_r	1003,0; 1013,0; 1023,0; 1033,0; 1043,0 [K]
P_F	4,0; 5,0; 6,0 [bar]
T_F	450,0; 460,0; 470,0 [K]
C_s	0,7; 0,6; 0,5

Fonte: produzido pelo autor.

Em seguida, as variáveis foram perturbadas de modo tal que os valores gerados pelas simulações se tornaram mais próximos do que seria obtido em uma planta real. As variáveis foram perturbadas usando a biblioteca *numpy* [51] para gerar dados aleatórios, seguindo uma distribuição normal de média 0.0 e desvio padrão especificado. O desvio padrão dos ruídos aplicados a uma dada variável foi definido a partir de uma porcentagem do valor médio dela. Devido às diferenças existentes entre os tipos de sensores, foram utilizadas porcentagens diferentes para tipos de variáveis diferentes como mostrado na Tabela 4.2. Foram escolhidos valores entre 1-5% para as porcentagens devido

às perturbações comumente encontradas na literatura [8, 26, 34, 47, 50]. Além disso, foram usados valores ligeiramente maiores para as vazões molares e pressões para refletir o fato de que estas medições comumente apresentam imprecisões um pouco maiores.

Tabela 4.2: Valores dos ruídos inseridos

Variável	Porcentagem	Distribuição
Vazões	5,0%	$\varepsilon(F_j) \sim N(0; (0,05x\mu(F_j)))$
Composições	1,5%	$\varepsilon(X_{ij}) \sim N(0; (0,015x\mu(W_{ij})))$
Pressões	3,0%	$\varepsilon(P_j) \sim N(0; (0,03x\mu(P_j)))$
Temperaturas	1,0%	$\varepsilon(T_j) \sim N(0; (0,01x\mu(T_j)))$

Fonte: produzido pelo autor.

A distribuição dos dados simulados antes e após a inserção de ruídos pode ser observada no Apêndice A.1 - Figuras A.1-A.6. É possível observar que o uso de 3 valores diferentes para a razão de reciclo provoca uma distribuição com 3 picos para as frações dos componentes mais leves na entrada do reator - $F_{A,Rin}$ e $F_{C,Rin}$. Essa distribuição dos dados do principal reagente na entrada do reator - Figura A.2 - faz com que ela se propague também para os produtos gerados na saída - A.3. Essa distribuição só parece se alterar quando os componentes leves e pesados são parcialmente separados no tanque de flash - Figuras A.4 e A.5. A inserção de ruídos nos dados faz com que as distribuições se tornem mais suaves, principalmente onde há picos. Também é importante ressaltar que a maioria dos valores das variáveis, tanto os medidos quanto os reais, seguem distribuições diferentes entre si e consideravelmente distintas da distribuição normal.

Também foi analisado o quanto os balanços de massa foram violados pelas perturbações nos dados. É interessante notar que apesar das distribuições incomuns produzidas pelas simulações, as violações seguem a forma de uma

distribuição normal - Apêndice A.2, Figuras A.7-A.10.

4.2 Pré-processamento

É bem estabelecido na literatura [3, 19] que o desempenho de algoritmos de *machine learning* pode ser consideravelmente melhorado caso sejam aplicados tratamentos adequados antes de alimentá-los ao modelo. Dentre as diversas estratégias de pré-tratamento, destacam-se a normalização dos dados, remoção de *outliers*, aplicação de filtros, redução de dimensionalidade, *feature engineering*, separação dos dados em teste e treino, entre outros [3, 23]. No caso específico de RNs, é importante garantir que os dados inseridos no modelo encontram-se dentro da mesma escala para que grandezas de maior magnitude não influenciem de forma desigual o modelo.

Neste trabalho, devido ao interesse em se explorar as violações de balanço de massa, foram aplicados somente normalização e separação do conjunto de dados em teste e treino. Foi utilizada como técnica de normalização a normalização baseada em mínimos e máximos - Equação 4.1. Esta estratégia garante que todas as variáveis inseridas na rede encontrem-se entre 0 e 1, característica que também é desejável quando se utilizam funções de ativação do tipo sigmoide.

$$x' = \frac{x - x_{min}}{x_{max} - x_{min}} \quad (4.1)$$

4.3 Função objetivo

Como abordado no Capítulo 2, a função objetivo mais comumente utilizada no problema de RD é a de MQP. Aqui foi adotada uma variação da mesma para contabilizar as violações nos balanços de massa. Além disso, valores reais das variáveis não foram utilizados no cálculo da função objetivo devido à ausência característica destes no problema de RD. Aplicando estas modificações, a Equação 2.18 se converte na Equação 4.2.

$$\min J(\underline{w}, \underline{\tilde{x}}, \underline{b}) = \sum_k \left(\frac{\tilde{x}_k - \hat{x}_k(\underline{w}, \underline{\tilde{x}}, \underline{b})}{\Delta x_k} \right)^2 + \lambda \sum_y \frac{f_y(\hat{x}_k(\underline{w}, \underline{\tilde{x}}, \underline{b}))^2}{\mu(f_y(\underline{\tilde{x}}))^2} \quad (4.2)$$

Esta função objetivo representa um compromisso entre dois fatores distintos. O primeiro termo, dito mínimos quadrados ponderados, contabiliza a diferença entre as medições $\tilde{\mathbf{x}}$ e as variáveis estimadas pela rede $\hat{\mathbf{x}}$. O fator Δx_k equivale a $\tilde{x}_{max} - \tilde{x}_{min}$ e é utilizado para normalizar os valores de cada uma das k variáveis. O segundo contabiliza o quanto as estimativas da rede violam os balanços de massa, onde a função f_y equivale ao cálculo de cada um dos y balanços de massa associados ao equipamento. f_y tem a forma das equações 4.3, 4.4, 4.5, 4.6, 4.7 para o reator e das equações 4.8, 4.9, 4.10, 4.11, 4.12 para o flash. O fator $\mu(f_y(\tilde{\mathbf{x}}))$ é igual à média do valor absoluto das violações dos balanços de massa calculados usando as variáveis medidas. Por fim, a constante λ é o parâmetro do modelo que determina o quanto as penalizações devem ser consideradas frente aos mínimos quadrados. Este deve ser ajustado para cada um dos sistemas.

$$f_g = F_{Rin} - F_{Rout} \quad (4.3)$$

$$f_A = F_{Rin}X_{A,Rin} - F_{Rout}X_{A,Rout} + V_r(r_A) \quad (4.4)$$

$$f_B = F_{Rin}X_{B,Rin} - F_{Rout}X_{B,Rout} + V_r(r_B) \quad (4.5)$$

$$f_C = F_{Rin}X_{C,Rin} - F_{Rout}X_{C,Rout} + V_r(r_C) \quad (4.6)$$

$$f_D = F_{Rin}X_{D,Rin} - F_{Rout}X_{D,Rout} + V_r(r_D) \quad (4.7)$$

$$f_g = F_{Fin} - F_{FV} - F_{FL} \quad (4.8)$$

$$f_A = F_{Fin}X_{A,Fin} - F_{FV}X_{A,FV} - F_{FL}X_{A,FL} \quad (4.9)$$

$$f_B = F_{Fin}X_{B,Fin} - F_{FV}X_{B,FV} - F_{FL}X_{B,FL} \quad (4.10)$$

$$f_C = F_{Fin}X_{C,Fin} - F_{FV}X_{C,FV} - F_{FL}X_{C,FL} \quad (4.11)$$

$$f_D = F_{Fin}X_{D,Fin} - F_{FV}X_{D,FV} - F_{FL}X_{D,FL} \quad (4.12)$$

Além da penalização por violação dos balanços, também foi testado um método de regularização que considera penalização por contração nos pesos da rede. Esta técnica é amplamente aplicada na literatura a problemas de regressão para evitar *overfitting*, e é conhecida pelo nome de regressão *Ridge* ou penalização de *Tikhonov*. Ela consiste em se adicionar à função objetivo um termo que contabilize a magnitude dos parâmetros, desfavorecendo que estes assumam valores maiores [23]. Adicionando o termo à Equação 4.2 obtemos a Equação 4.13, onde o fator l_2 determina o quanto o crescimento dos pesos deve ser penalizado. Este fator precisa ser ajustado de forma apropriada para cada problema.

$$J(\underline{w}, \tilde{\underline{x}}, \underline{b}) = \sum_k \left(\frac{\tilde{x}_k - \hat{x}_k(\underline{w}, \tilde{\underline{x}}, \underline{b})}{\Delta x_k} \right)^2 + \lambda \sum_y \frac{f_y(\hat{x}_k(\underline{w}, \tilde{\underline{x}}, \underline{b}))^2}{\mu(f_y(\tilde{\underline{x}}))^2} l_2 \sum_{m,n} w_{mn}^2 \quad (4.13)$$

4.4 Topologia

Conforme abordado na seção 2.7.3, a arquitetura das AANNs consiste em camadas de entrada e saída com a mesma dimensão dos vetores de medida, além de 3 camadas internas ao longo das quais as informações são comprimidas e reconstruídas.

Esta topologia foi aplicada para reconciliar as variáveis do reator e do flash de forma separada. No flash as variáveis reconciliadas foram F_{Fin} , $X_{i,Fin}$, F_{FV} , $X_{i,FV}$, F_{FL} , $X_{i,FL}$, T_f e P_f . No reator, foram F_{Rin} , $X_{i,Rin}$, F_{Rout} , $X_{i,Rout}$, T_r e P_r . Com relação às penalizações apresentadas na Equação 4.2, vale ressaltar uma diferença ao aplicá-las a cada um dos equipamentos. No flash, temperatura e pressão não aparecem nas equações de balanço, o que faz com que estas variáveis sejam afetadas apenas pelo termo referente aos mínimos quadrados. Já no reator, T_r e P_r estão acoplados aos balanços através das expressões das taxas - Equações 3.4-3.7. Isto faz com que estas variáveis sejam afetadas por ambos os termos da Equação 4.2.

Foram utilizadas funções de ativação do tipo sigmoide nas camadas de mapeamento e desmapeamento visando capturar a forte não-linearidade do processo. Nas camadas de gargalo e saída foram utilizadas funções lineares [31]. Também foi realizada uma busca para encontrar o número ideal de neurônios das camadas internas, como será esclarecido mais à frente.

4.5 Treinamento

Antes de iniciar o treinamento, os dados foram normalizados e divididos de forma aleatória em um conjunto de treino e outro de teste. Metade das 3375 amostras foram utilizadas no treinamento. Foi utilizada a técnica de otimização ADAM [28], baseada em gradiente e estimativas adaptativas individuais das taxas de aprendizado para cada parâmetro. A cada iteração, o gradiente da função foi calculado utilizando todos os 1688 pontos do conjunto de dados de treino. Para se determinar a interrupção da otimização, foi utilizado um critério de convergência baseado no desvio padrão relativo dos dados reconciliados $\hat{\sigma}_{rel}$. Este critério será melhor elucidado na Seção 4.6. Após uma análise preliminar, também foi estabelecida a quantidade limite de 35000 épocas para interromper cálculos incapazes de atingir o critério $\hat{\sigma}_{rel}$ estabelecido.

A cada iteração são calculadas médias móveis ponderadas exponencialmente do gradiente e do gradiente ao quadrado. Estes valores são então utilizados para determinar individualmente a taxa de aprendizado a ser utilizada para cada um dos parâmetros treinados. As Equações 4.14-4.16 demonstram como é realizado o cálculo dos parâmetros θ do modelo a cada t iteração:

$$\underline{\theta}_t = \underline{\theta}_{t-1} - \alpha \frac{\underline{\hat{m}}_t}{\sqrt{\underline{\hat{v}}_t} + \epsilon} \quad (4.14)$$

$$\underline{\hat{m}}_t = \frac{(1 - \beta_1) \sum_{t=0}^t \frac{\partial J}{\partial \underline{\theta}_t}}{1 - \beta_1^t} \quad (4.15)$$

$$\underline{\hat{v}}_t = \frac{(1 - \beta_2) \sum_{t=0}^t \left(\frac{\partial J}{\partial \underline{\theta}_t}\right)^2}{1 - \beta_2^t} \quad (4.16)$$

Em que $\underline{\theta}$ é o vetor de parâmetros da rede que compreende os pesos w_{mn} e b_n da rede, e α , β_1 , β_2 e ϵ são parâmetros do método de otimização. O primeiro deles, α , influencia o tamanho médio dos passos estimados. β_1 e β_2 determinam o quão rápido é o decaimento exponencial das médias móveis do gradiente e gradiente ao quadrado. Por fim, ϵ é uma constante usada para impedir instabilidade numérica em casos onde \hat{v} tende a zero.

Para β_1 e β_2 foram mantidos os valores de 0.9 e 0.99 inicialmente sugeridos por Kingma & Ba [28]. Para o parâmetro α , o valor originalmente proposto de 10^{-3} foi reduzido para 10^{-4} de modo a tentar assegurar uma convergência mais suave. Para ϵ , foi usado o valor de 10^{-7} , o padrão na implementação da técnica no *Tensorflow*, um *framework* para a implementação de redes neurais em *python* [2].

Devido às modificações realizadas na função objetivo, o cálculo dos gradientes via *backpropagation* também precisou ser modificado. A inserção do segundo termo na Equação 4.2 introduz o cálculo de novas contribuições ao gradiente da função objetivo através de $(\frac{\partial f_y}{\partial w_{mn}})$ e $(\frac{\partial f_y}{\partial b_m})$. Para computar os gradientes, foi utilizado o algoritmo de autodiferenciação da versão 2.2 do *Tensorflow* [2].

4.6 Métricas

Para garantir que o modelo proposto atenda aos requisitos de redução de ruídos e satisfação dos balanços, três métricas foram propostas e monitoradas durante o treinamento. A primeira delas é expressa pela Equação 4.17 e calcula o erro quadrático médio dos dados reconciliados para cada uma das K variáveis e Z amostras. Essa métrica utiliza os valores normalizados

dos dados para contabilizar de forma equivalente as diferentes escalas das variáveis. Contudo, seu valor só pode ser calculado caso os valores reais das variáveis sejam conhecidos, o que não ocorre em uma situação real.

$$EQM(\hat{x}') = \frac{1}{Z} \sum_z \frac{1}{K} \sum_k (\hat{x}'_k - x'_k)^2 \quad (4.17)$$

A segunda métrica utilizada é o desvio padrão relativo - Equação 4.18, que foi usada para determinar a interrupção da otimização. O seu valor reflete a razão entre o desvio padrão dos dados reconciliados e das medições. Seu cálculo também foi realizado utilizando-se os valores normalizados das variáveis para compensar as diferenças entre as escalas. Quando este indicador se aproxima de 1, o valor dos dados reconciliados tende ao das medições, o que indica que o estimador falhou em sua função de reduzir ruídos. Para interromper o treinamento, utilizou-se $\hat{\sigma}_{rel} = 0,96$. Este valor foi escolhido calculando-se $\hat{\sigma}_{rel}$ com o desvio padrão dos dados reais no numerador da Equação 4.18, de tal modo que o treinamento é interrompido quando o desvio padrão dos dados reconciliados é aproximadamente o mesmo dos dados reais. Este valor de 0,96 também não estaria disponível em uma situação real, embora a métrica pudesse ser calculada.

$$\hat{\sigma}_{rel}(\hat{x}') = \frac{\sum_k \sigma(\hat{x}'_k)}{\sum_k \sigma(\tilde{x}'_k)} \quad (4.18)$$

Por fim, a terceira métrica utilizada consiste nas próprias violações de balanço caracterizadas pelas Equações 4.5 a 4.6, 4.7 para o reator e 4.8 a 4.12 para o flash. Uma vez que a função objetivo escolhida reflete o compromisso entre a tendência dos dados e a violação dos balanços, esta métrica também é relevante para avaliar o desempenho do algoritmo.

Capítulo 5

Resultados e Discussão

Neste capítulo são abordados os resultados obtidos utilizando os modelos descritos no Capítulo 4 e considerando diferentes combinações de parâmetros. Inicialmente, o desempenho da AANN é avaliado sem considerar as penalizações na função objetivo, com a redução de ruídos sendo realizada exclusivamente pela compressão dos dados na camada gargalo. Posteriormente, são inseridas as penalizações através dos balanços de massa e da regularização de *Ridge*, com o intuito de observar como eles afetam a variância e a satisfação dos balanços pelos dados reconciliados. A influência do número de neurônios da camada de gargalo também é estudada com o objetivo de se determinar qual seria o número ideal de componentes principais não-lineares a ser utilizado para reconciliar as variáveis em cada caso.

O número de neurônios das camadas internas de cada rede foi inicialmente definido tomando-se como referência a dimensão do vetor de variáveis de entrada. Para as camadas de mapeamento e desmapeamento foi utilizada uma quantidade de neurônios próxima ao dobro do número de variáveis de

entrada. Para a camada de gargalo, foi utilizada uma quantidade próxima à metade da dimensão das variáveis de entrada. Esta referência para o tamanho das camadas foi utilizada com o intuito de se garantir que a rede seja capaz de captar a complexidade do modelo sem comprometer a compressão das informações necessária para a reconciliação dos dados [32].

As figuras onde são apresentadas as distribuições das variáveis antes e depois da reconciliação foram incluídas nos Apêndices A.3-A.10 para facilitar a leitura do texto.

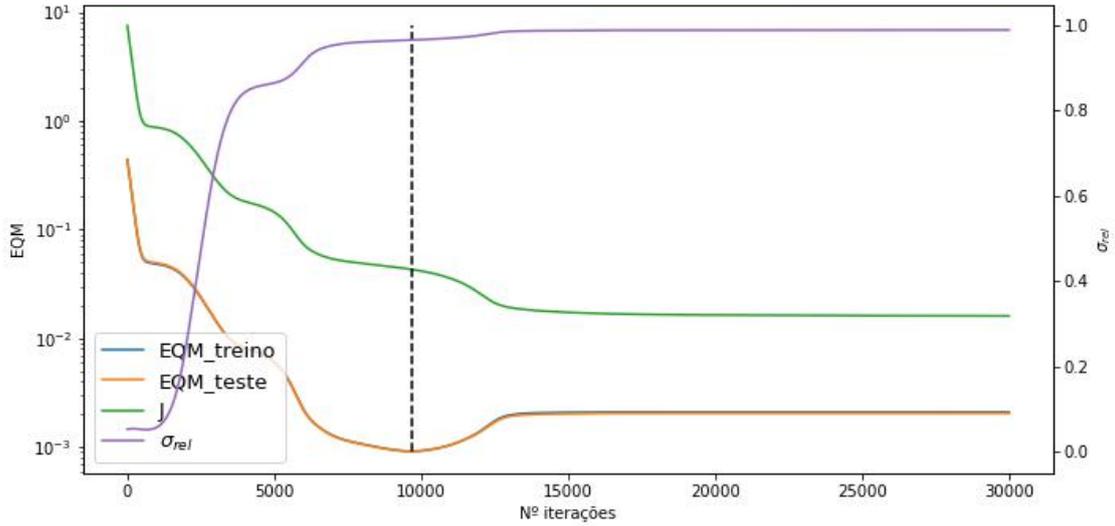
5.1 Reconciliação sem penalização

5.1.1 Flash

Inicialmente foram utilizados 32 neurônios para as camadas de mapeamento e desmapeamento, e 8 neurônios na camada de gargalo para reconciliar os dados do flash. A Figura 5.1 mostra que o critério de $\hat{\sigma}_{rel} = 0.96$ para parada da otimização é satisfatório, uma vez que em torno dessa região o EQM começa a aumentar.

As figuras no Apêndice A.3 mostram que essa configuração consegue aprender o comportamento das variáveis mesmo sem a presença dos modelos fenomenológicos na função objetivo. Contudo, nem todas as variáveis reconciliadas apresentam valores mais acurados e menos ruidosos do que as medições iniciais. A Tabela 5.1 apresenta a razão $EQM(\hat{x}'_k)/EQM(\tilde{x}'_k)$ e mostra que o resultado para algumas variáveis como $X_{C,FL}$, $X_{B,FV}$ e $X_{D,FV}$ piora após a reconciliação. Essas variáveis têm em comum o fato de representarem frações molares muito pequenas e possuem um grau de incerteza

Figura 5.1: Evolução de $\hat{\sigma}_{rel}$ e EQM ao longo do treinamento. Treinamento interrompido na linha pontilhada.



Fonte: produzido pelo autor.

maior com relação aos seus valores reais - Tabela 3.6. Isto possivelmente torna mais difícil para a rede inferir seus valores a partir das medições, dado que elas provavelmente apresentarem uma correlação mais fraca com os outros dados.

A Tabela 5.1 também evidencia que há uma correlação entre a diminuição do erro quadrático médio e o desvio padrão de cada variável. A segunda coluna apresenta a razão entre o desvio padrão dos dados reais e das medições para cada uma das k variáveis $\sigma(x'_k)/\sigma(\tilde{x}'_k)$. A Figura 5.2 reforça que o resultado piora em casos onde $\sigma(x'_k)/\sigma(\tilde{x}'_k)$ se aproxima de 1.

Os resultados também apontam que os dados reconciliados atendem melhor aos balanços de massa mesmo sem que estes sejam incluídos na função objetivo. A Tabela 5.2 mostra a redução relativa das violações e o quanto os

Tabela 5.1: Redução do EQM e razão entre os desvios padrão real e das medições

	$\frac{EQM(\hat{x}'_k)}{EQM(\tilde{x}'_k)}$	$\frac{\sigma(x'_k)}{\sigma(\tilde{x}'_k)}$
F_{Fin}	0,732904	0,969833
F_{FL}	0,771206	0,951756
F_{FV}	0,808902	0,978581
$X_{A,Fin}$	0,339460	0,780620
$X_{B,Fin}$	0,875681	0,989052
$X_{C,Fin}$	0,442883	0,966825
$X_{D,Fin}$	1,551442	0,995613
$X_{A,FL}$	2,325514	0,996099
$X_{B,FL}$	0,665178	0,938481
$X_{C,FL}$	1,360526	0,995419
$X_{D,FL}$	0,625630	0,995210
$X_{A,FV}$	0,412966	0,935510
$X_{B,FV}$	2,434987	0,997476
$X_{C,FV}$	0,361311	0,901278
$X_{D,FV}$	5,607862	0,999321
T_F	0,388323	0,875158
P_F	0,775818	0,980993

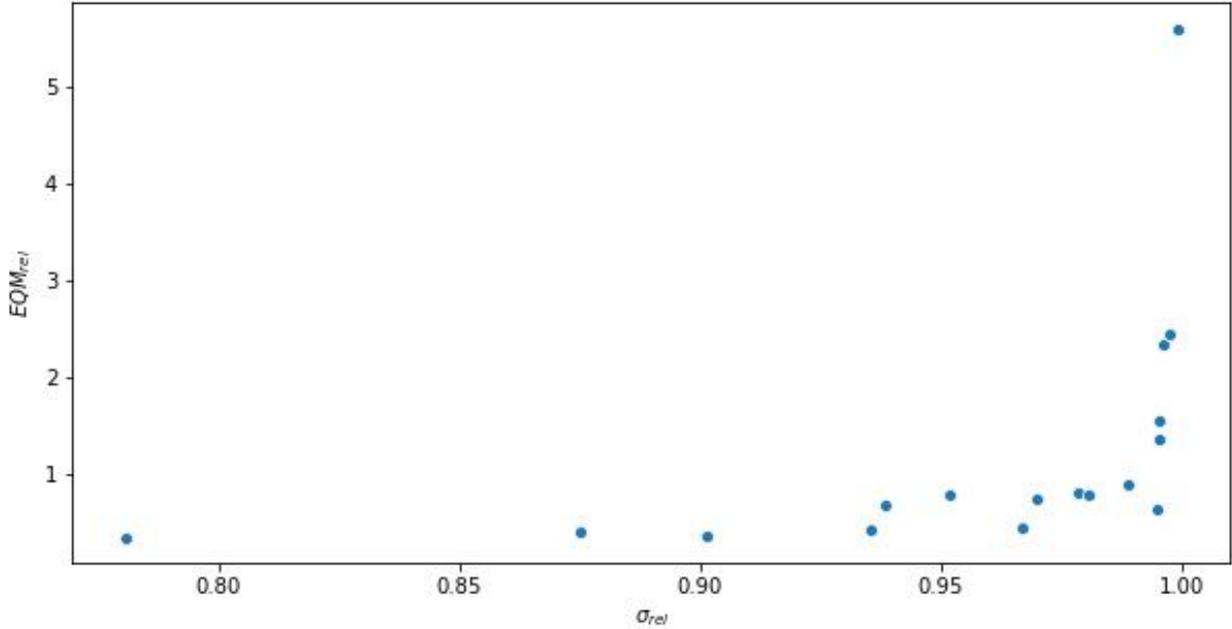
Fonte: produzido pelo autor.

balanços são violados pelas medições e pelos dados reconciliados.

5.1.2 Reator

Para o reator, foram utilizados 24 neurônios para as camadas de mapeamento e desmapeamento, e 5 neurônios na camada de gargalo para reconciliação preliminar. A Figura 5.3 mostra que o critério de $\hat{\sigma}_{rel} = 0,96$ para parada da otimização não é tão adequado quanto no caso do flash, uma vez o EQM continua a diminuir após este ponto. Contudo, este critério apresenta a vantagem de não ser necessário se conhecer precisamente o valor real das

Figura 5.2: Relação entre a diminuição do Erro Quadrático Médio e σ_{rel} no flash com $\lambda = 0$.



Fonte: produzido pelo autor.

variáveis, e portanto foi mantido.

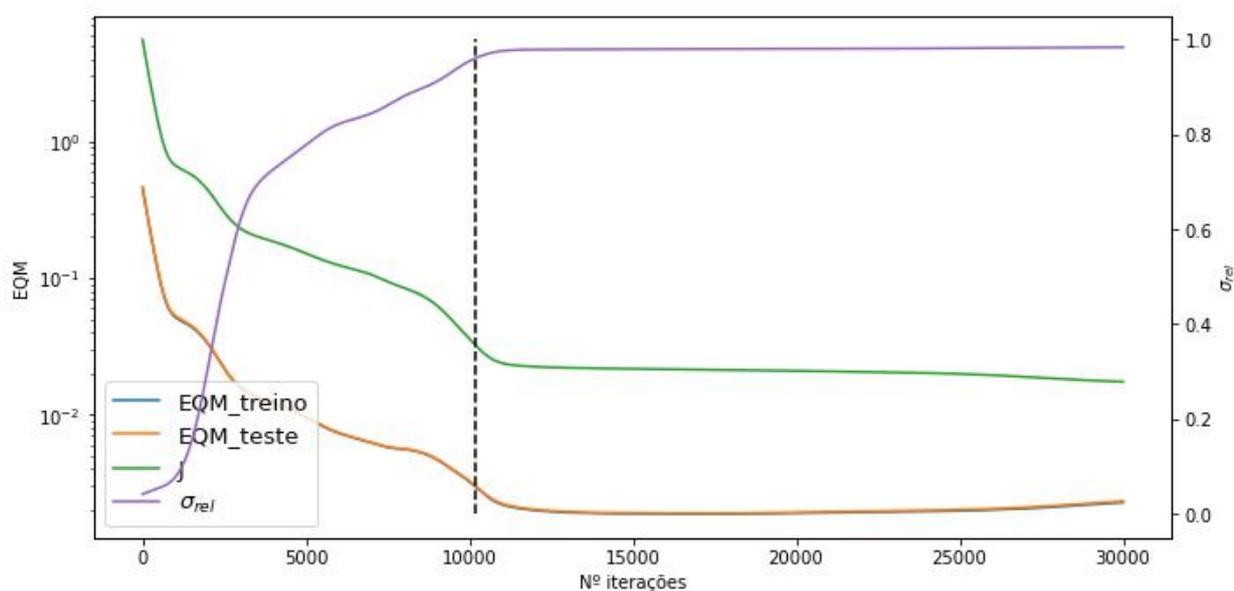
O Apêndice A.4 mostra novamente que a rede consegue aprender o comportamento das variáveis mesmo sem o uso explícito dos balanços. É interessante observar aqui que as variáveis $X_{B,Rin}$ e $X_{D,Rin}$ apresentaram comportamento similar ao de $X_{C,FL}$, $X_{B,FV}$ e $X_{D,FV}$ para o flash. Este resultado era esperado, uma vez que as frações molares de B e D na entrada do reator são diretamente influenciadas pelas frações destes componentes na saída do flash, e portanto sofrem dos mesmos problemas de baixa confiabilidade e baixa correlação com os outros dados. A Tabela 5.3 é equivalente à Tabela 5.3 para o reator e apresenta o quanto cada variável melhora ou piora com a reconciliação. Novamente, a Figura 5.4 reforça que o resultado da reconciliação piora as medições exclusivamente nos casos onde $\sigma(x'_k)/\sigma(\tilde{x}'_k)$ está muito

Tabela 5.2: Violações de Balanço no flash com $\lambda = 0$.

	Global	A	B	C	D
Reconciliados	2,2441	0,9451	0,4552	1,0435	0,258790
Medições	10,6478	4,6085	1,1785	5,4861	0,520144
Redução (%)	78,92	79,49	61,38	80,98	50,2466

Fonte: produzido pelo autor.

Figura 5.3: Evolução de $\hat{\sigma}_{rel}$ e EQM ao longo do treinamento. Treinamento interrompido na linha pontilhada.



Fonte: produzido pelo autor.

próximo de 1.

Novamente, os balanços passam a ser melhor atendidos após a reconciliação mesmo sem a sua presença na função objetivo como mostra a Tabela 5.4.

Tabela 5.3: Redução do EQM e razão entre os desvios padrão real e das medições

	$\frac{EQM(\hat{x}'_k)}{EQM(\tilde{x}'_k)}$	$\frac{\sigma(x'_k)}{\sigma(\tilde{x}'_k)}$
F_{Fin}	0,851685	0,965061
F_{FL}	0,791451	0,967917
$X_{A,Rin}$	0,458195	0,971446
$X_{B,Rin}$	3,114286	0,999583
$X_{C,Rin}$	1,099059	0,996963
$X_{D,Rin}$	3,361341	0,999196
$X_{A,Rout}$	0,310176	0,813936
$X_{B,Rout}$	0,942461	0,987688
$X_{C,Rout}$	0,362426	0,953288
$X_{D,Rout}$	0,932081	0,994217
T_F	0,913300	0,807865
P_F	0,908408	0,882579

Fonte: produzido pelo autor.

Tabela 5.4: Violações de Balanço no reator com $\lambda = 0$.

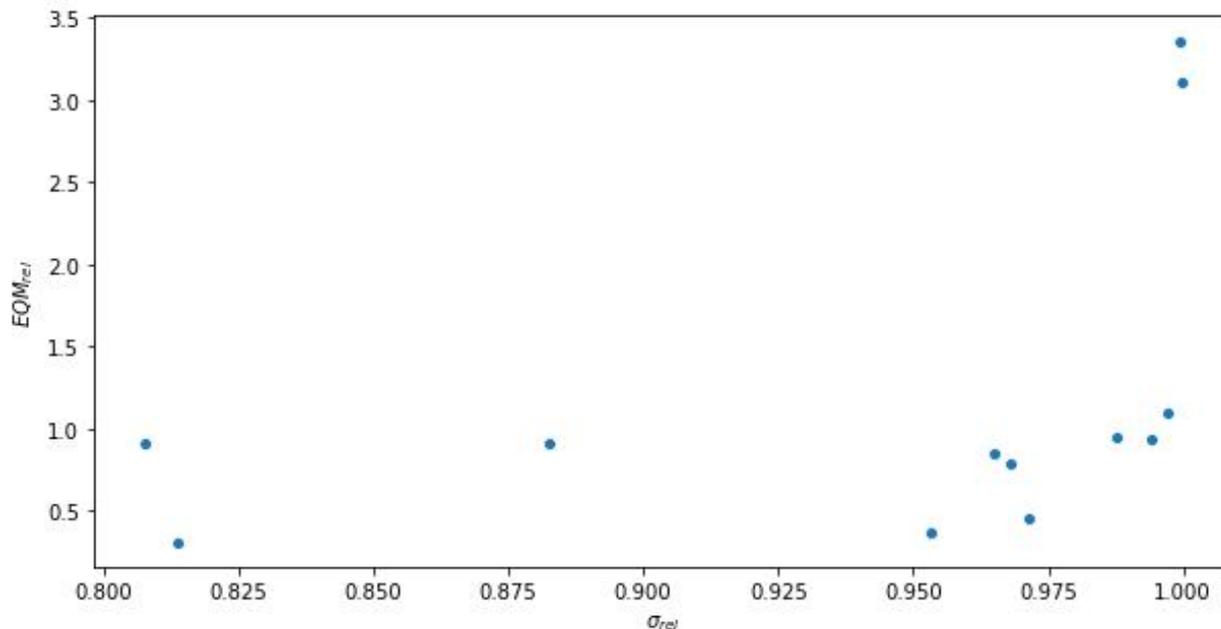
	Global	A	B	C	D
Reconciliados	4,158331	7,376192	6,565421	4,758213	3,420961
Medições	11,773091	33,568130	13,290157	19,340944	7,174345
Redução (%)	64,68	78,03	50,60	75,40	52,32

Fonte: produzido pelo autor.

5.2 Reconciliação com as violações de balanço penalizadas

Nesta etapa os modelos são avaliados considerando $\lambda > 0$. Inicialmente, foi feito para cada caso uma busca pelo melhor valor deste parâmetro, considerando a mesma quantidade de neurônios nas camadas internas utilizados na seção anterior. Na sequência, o efeito da quantidade de neurônios da camada de gargalo foi avaliado. Por fim, os resultados são comparados com os

Figura 5.4: Relação entre a diminuição do Erro Quadrático Médio e $\hat{\sigma}_{rel}$ no reator com $\lambda = 0$.



Fonte: produzido pelo autor.

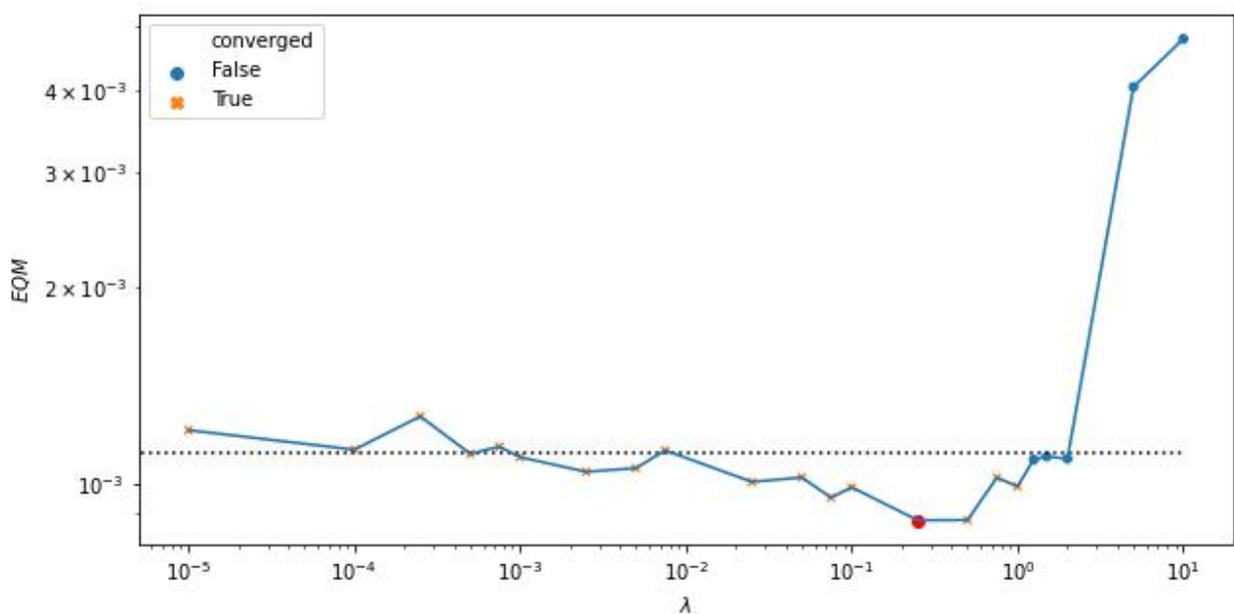
obtidos para $\lambda = 0$.

5.2.1 Flash

O melhor valor encontrado no flash para λ está em torno de 0,25 como mostra a Figura 5.5. A linha pontilhada corresponde ao EQM obtido para $\lambda = 0$ e mostra que existe uma região onde os resultados melhoram com a adição das penalizações. Quando são utilizados valores muito altos para λ , o algoritmo tem dificuldade de convergir, e não consegue aprender satisfatoriamente a tendência dos dados. A razão $EQM(\lambda = 0,25)/EQM(\lambda = 0)$ equivale a 0,7907, resultando em uma redução de 21% no EQM . Os resultados no Apêndice A.5 parecem indicar que um pouco de *bias* é inserido

nos dados reconciliados para o caso onde $\lambda = 0,25$, principalmente para o componente B. Este resultado não é surpreendente, uma vez que um dos objetivos da reconciliação é diminuir a variância dos dados. Isto implica que muito provavelmente o modelo deve ser influenciado pelo *tradeoff* entre erros de *bias* e variância já reportado na literatura [23].

Figura 5.5: EQM em função de λ para o flash. Resultado para $\lambda = 0$ representado pela linha pontilhada. Melhor valor de $\lambda = 0,25$ representado por círculo maior em vermelho. Valores representados por círculos azuis representam cálculos que não atingiram $\sigma_{rel} = 0,96$.

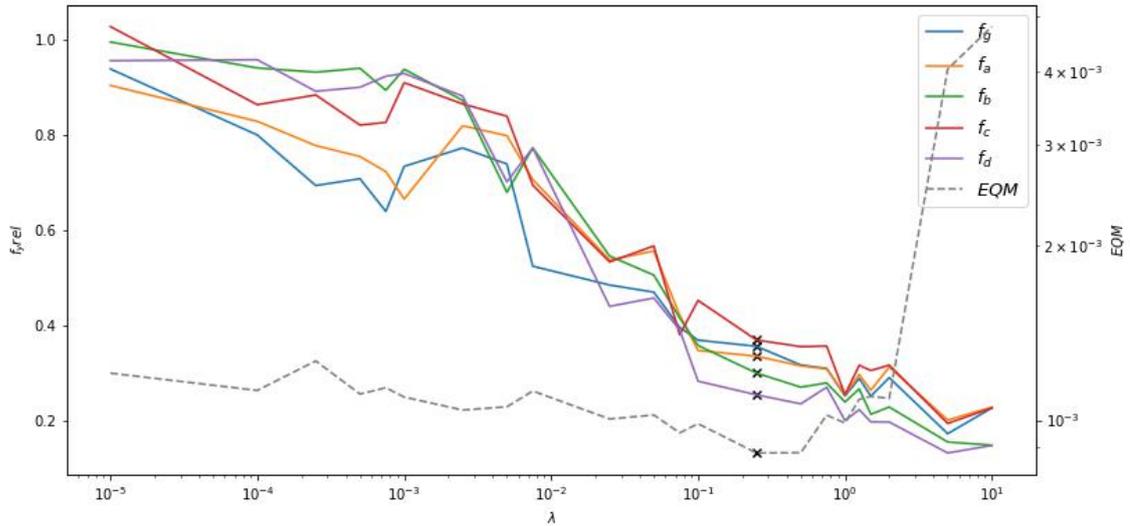


Fonte: produzido pelo autor.

A Figura 5.6 mostra como evolui a razão $f_y(\lambda)/f_y(\lambda = 0)$. Conforme λ tende a zero, esta razão tende a 1, e vai diminuindo conforme a penalização dos balanços aumenta. Nesta Figura é possível perceber o compromisso entre os desbalanceamentos e o EQM. Por outro lado, a partir de um certo ponto, os dados reconciliados passam a representar uma solução que atende aos balanços mas está distante da solução desejada, o que pode ser observado

pelo aumento do EQM.

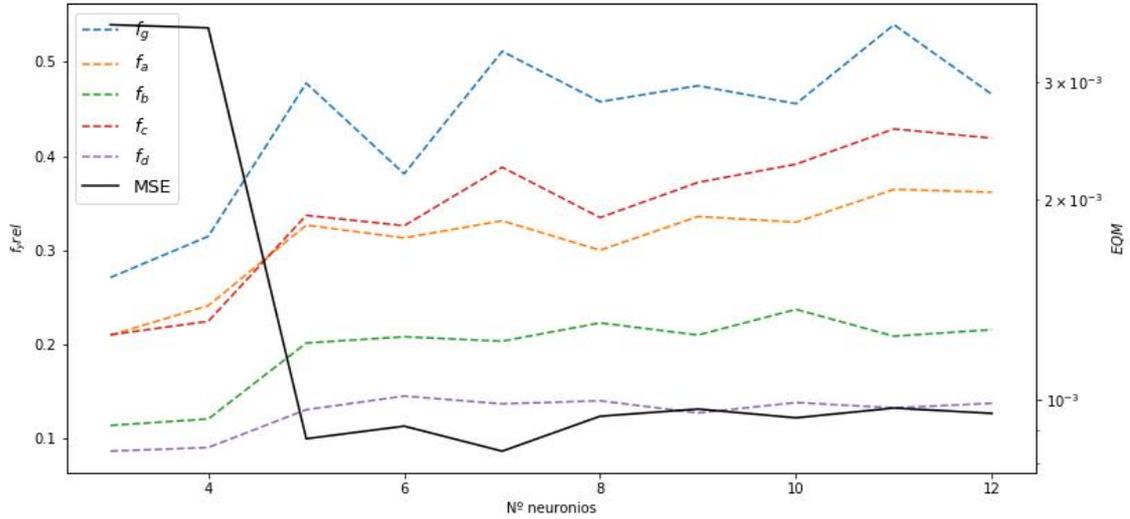
Figura 5.6: Desbalanceamentos em função de λ para o flash. Melhor valor de λ marcado com um x.



Fonte: produzido pelo autor.

Como o esperado, a rede é sensível ao número de neurônios na camada gargalo. A Figura 5.7 mostra que o resultado melhora consideravelmente a partir de uma determinada quantidade de neurônios e depois não varia tanto. Os resultados no Apêndice A.6 mostram que, com 4 neurônios ou menos, a rede tem dificuldade em estimar corretamente os valores das correntes. Isso indica que 5 componentes principais não-lineares é o mínimo para que a rede consiga prever a tendência de todas as variáveis. As violações de balanço não parecem ser tão sensíveis a este parâmetro, mas é possível observar uma tendência de crescimento, provavelmente devido ao aumento de ruídos provocados por permitir que mais informação se propague ao longo da camada de gargalo. Com base nisso, provavelmente um número menor de neurônios, entre 5 e 7, teria gerado resultados melhores do que a escolha inicial de 8.

Figura 5.7: EQM e desbalanceamentos do flash em função do número de neurônios da camada de gargalo.

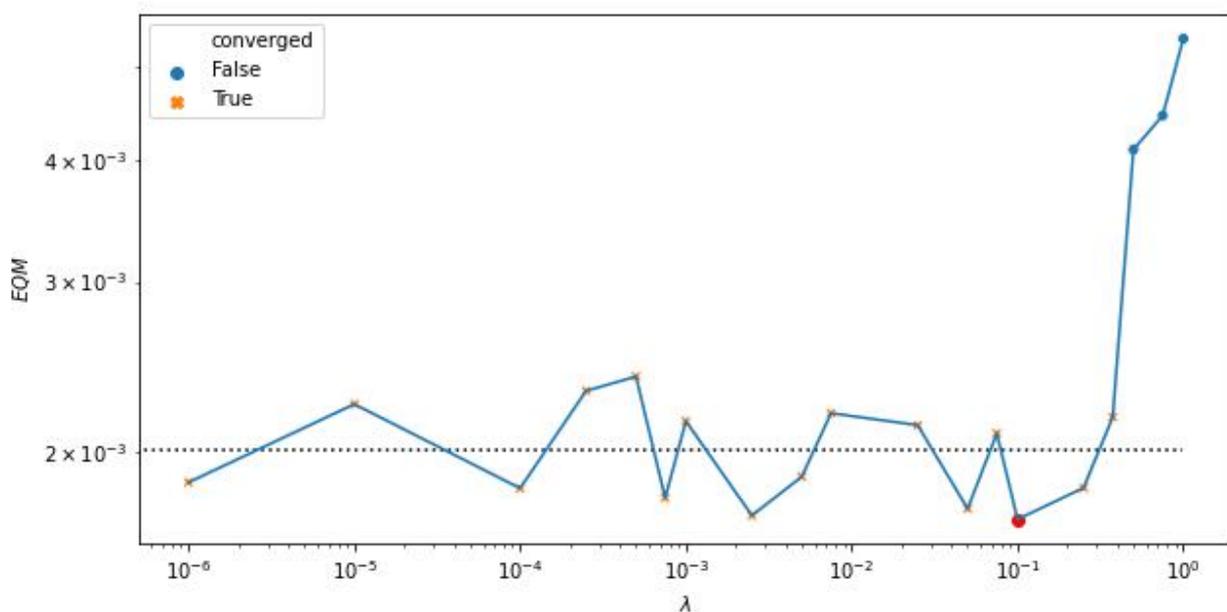


Fonte: produzido pelo autor.

5.2.2 Reator

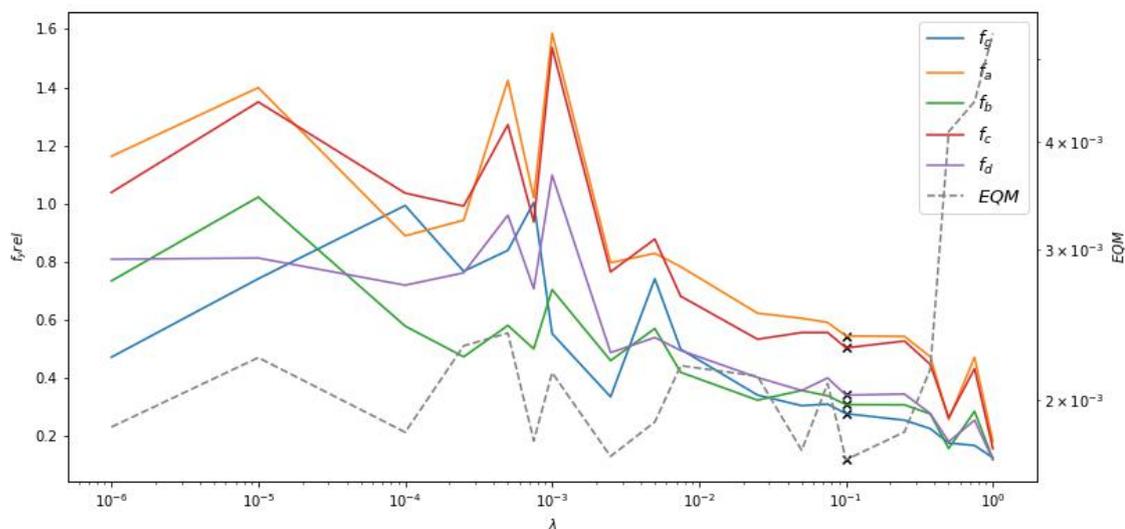
Para o reator, o λ ideal foi de 0,1 como pode ser observado na Figura 5.8. Novamente, os resultados parecem melhorar com as penalizações, principalmente na região de λ entre 0,5 e 0,1. A razão $EQM(\lambda = 0, 1)/EQM(\lambda = 0)$ equivale a 0,8472, resultando em uma redução de 15% no EQM . Além disso, as violações parecem também decrescer linearmente nessa região, sugerindo que este parâmetro ajuda a encontrar soluções que atendem melhor aos balanços de massa - Figura 5.9. Neste caso, as estimativas obtidas para as vazões molares foram levemente enviesadas - Apêndice A.7.

Figura 5.8: EQM em função de λ para o reator. Resultado para $\lambda = 0$ representado pela linha pontilhada. Melhor valor de $\lambda = 0,1$ representado por círculo maior em vermelho. Valores representados por círculos azuis representam cálculos que não atingiram $\sigma_{rel} = 0,96$.



Fonte: produzido pelo autor.

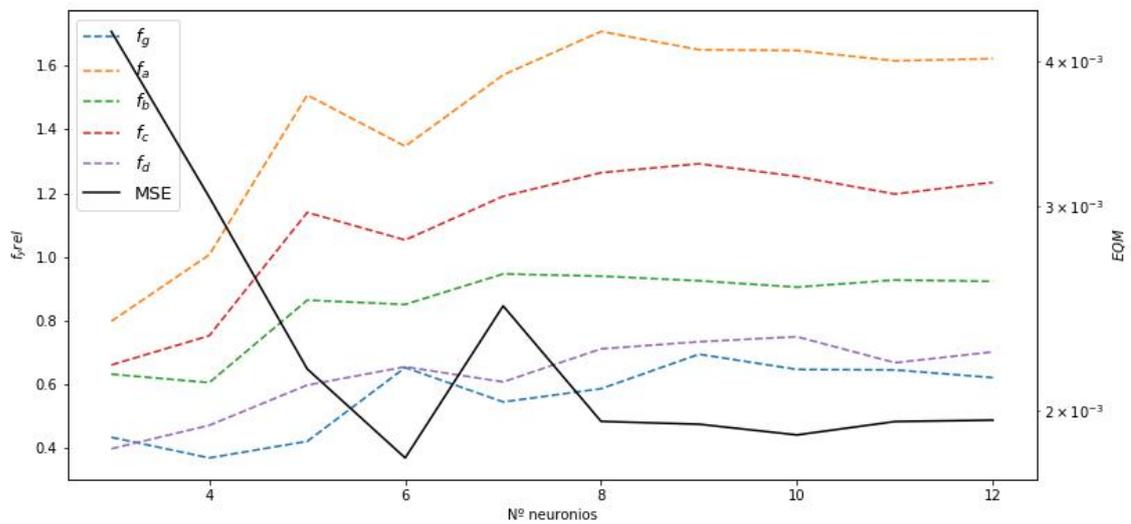
Figura 5.9: Desbalanceamentos em função de λ para o reator. Melhor valor de λ marcado com um x.



82
Fonte: produzido pelo autor.

O reator também apresentou sensibilidade ao número de neurônios na camada de gargalo. A Figura 5.10 sugere que 6 é o número ideal de neurônios a ser usado e que o resultado melhora muito a partir de 5. Novamente, as vazões molares foram as variáveis mais afetadas ao se utilizar um número insuficiente de neurônios nesta camada - Apêndice A.8. Assim como no flash, as violações de balanço apresentam uma tendência de crescimento com o número de neurônios, reforçando a hipótese de que isso pode se dar por uma maior tolerância a ruídos.

Figura 5.10: EQM e desbalanceamentos no reator em função do número de neurônios da camada de gargalo.



Fonte: produzido pelo autor.

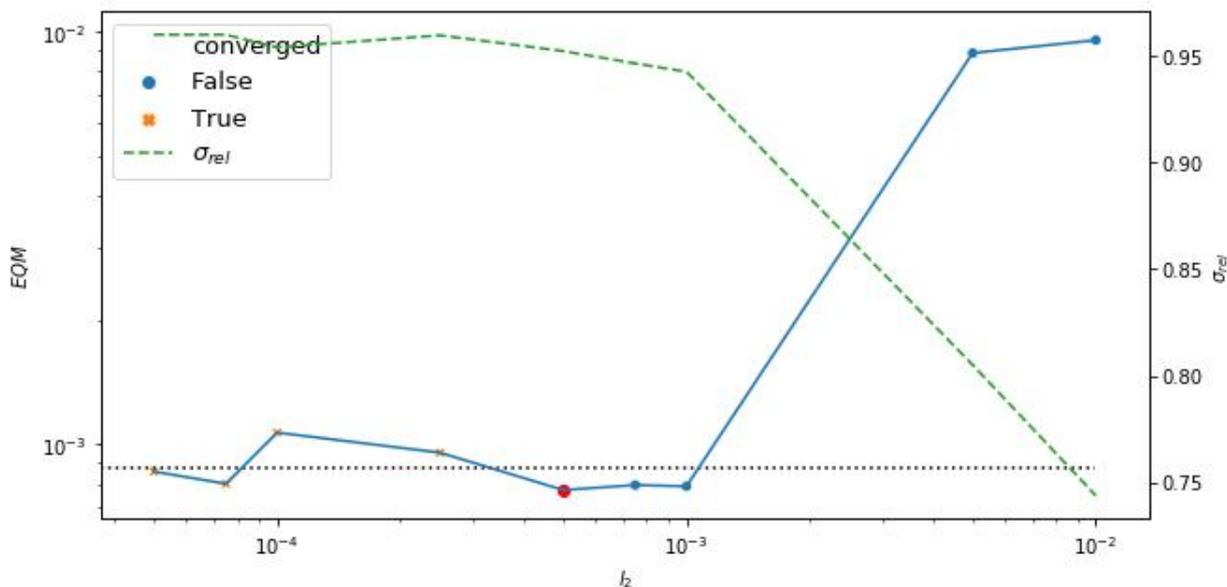
5.3 Reconciliação com penalização de *Tikhonov*

Nesta etapa, os modelos são avaliados considerando o melhor valor de λ encontrado na seção anterior e valores de $l_2 > 0$. Para facilitar a comparação, a quantidade de neurônios de cada modelo foi a mesma utilizada na busca pelo λ ótimo e nos resultados preliminares.

5.3.1 Flash

Assim como ocorreu quando penalizamos usando apenas os balanços de massa, existe uma região onde a regularização l_2 produz melhorias no resultado do modelo. A razão $EQM(\lambda = 0,25; l_2 = 0,0005)/EQM(\lambda = 0; l_2 = 0)$ equivale a 0,6968, resultando em uma redução de 30% no EQM , 10% melhor do que para o caso com $l_2 = 0$. Aqui, é interessante notar que bons resultados foram obtidos em cálculos onde o modelo não conseguiu atingir $\sigma_{rel} = 0,96$ mas ficou bastante próximo - curva verde pontilhada da Figura 5.12 - o que sugere uma redução na variância dos dados.

Figura 5.11: EQM em função de l_2 para o flash. Resultado para $l_2 = 0$ representado pela linha pontilhada. Melhor valor de $l_2 = 0,0005$ indicado pelo círculo maior em vermelho.



Fonte: produzido pelo autor.

A tabela 5.5 mostra a evolução na redução dos desbalanceamentos para os três casos estudados. A partir dela, é possível concluir que soluções que atendem melhor aos balanços são favorecidas principalmente pela inclusão dos balanços na função objetivo, enquanto as penalizações de *Tikhonov* parecem não surtir um efeito tão significativo.

Tabela 5.5: Violações de Balanço no flash

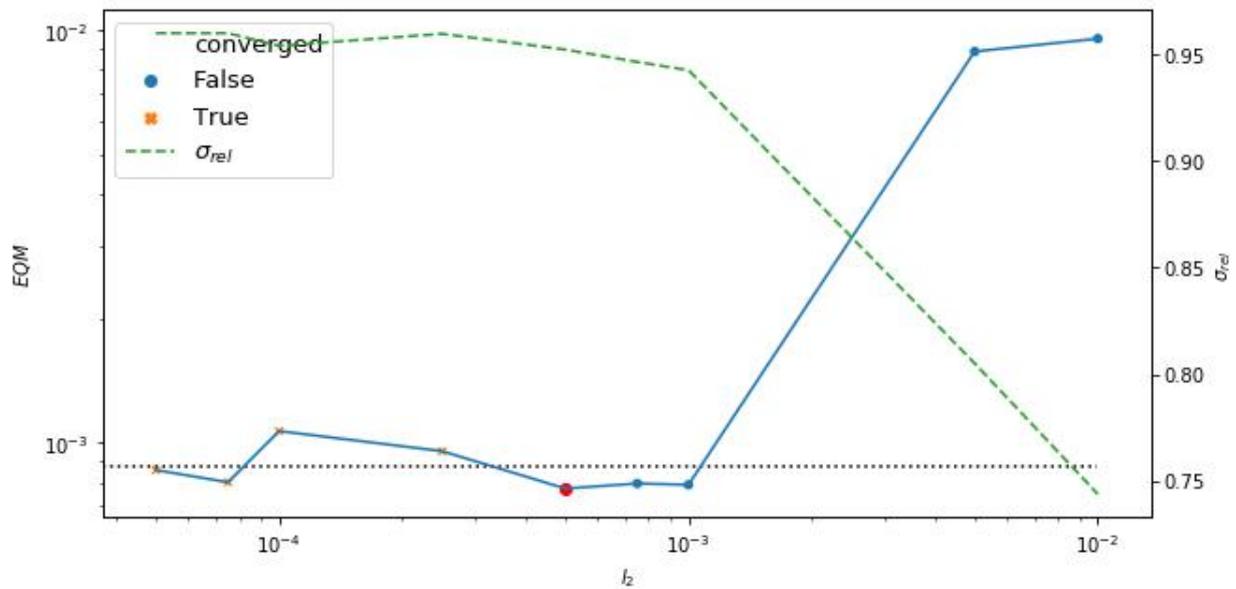
Redução (%)	Global	A	B	C	D
$\lambda = 0,0$ e $l_2 = 0,0$	78,92	79,49	61,37	80,98	50,24
$\lambda = 0,25$ e $l_2 = 0,0$	97,33	97,69	96,53	97,41	96,79
$\lambda = 0,25$ e $l_2 = 0,0005$	98,78	98,73	97,11	98,43	97,04

Fonte: produzido pelo autor.

A figura 5.13 ilustram como as violações de balanço se comportam com a

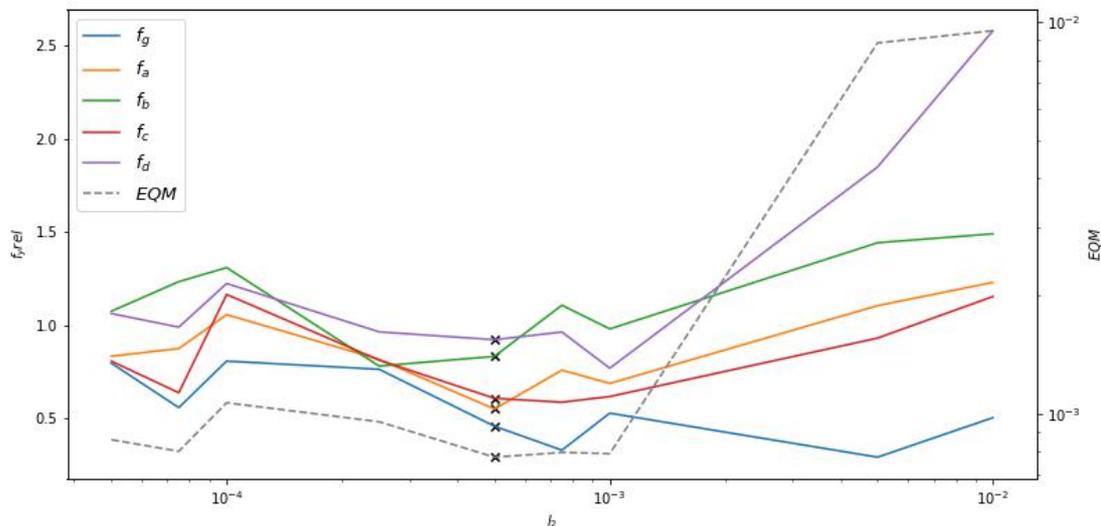
variação de l_2 . Estas apresentaram melhora na região próxima ao l_2 ótimo. Os resultados obtidos com $\lambda = 0,25$ e $l_2 = 0,0005$ podem ser encontrados no Apêndice A.9.

Figura 5.12: EQM em função de l_2 para o flash. Resultado para $l_2 = 0$ representado pela linha pontilhada. Melhor valor de $l_2 = 0,0005$ indicado pelo círculo maior em vermelho.



Fonte: produzido pelo autor.

Figura 5.13: Desbalanceamentos em função de l_2 para o flash. Melhor valor de l_2 marcado com um x .



Fonte: produzido pelo autor.

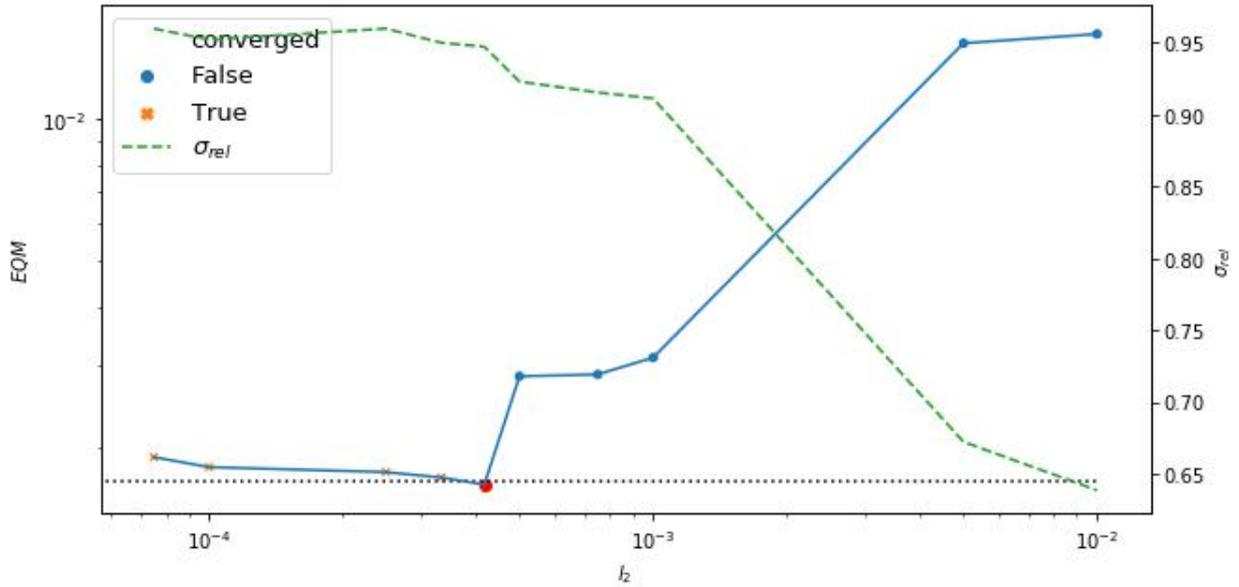
5.3.2 Reator

A reconciliação no reator parece ser menos beneficiada com a inclusão da penalização de *Tikhonov* em comparação com o flash. O melhor resultado obtido, usando $l_2 = 0,0004166$, não apresentou um valor de EQM significativamente mais baixo do que o caso utilizando apenas os balanços como penalização como mostra a Figura 5.14. A razão $EQM(\lambda = 0, 1; l_2 = 0,0004166)/EQM(\lambda = 0; l_2 = 0)$ equivale a 0,8333, resultando em uma redução de 16% no EQM, o que não representa um ganho expressivo para o caso em que $l_2 = 0$.

As violações de balanço também não apresentaram uma tendência conclusiva de melhora como mostram a Figura 5.15 e a tabela 5.6. Dados recon-

ciliados no Apêndice A.10.

Figura 5.14: EQM em função de l_2 para o reator. Resultado para $l_2 = 0$ representado pela linha pontilhada. Melhor valor de $l_2 = 0,0004166$ indicado pelo círculo maior em vermelho.



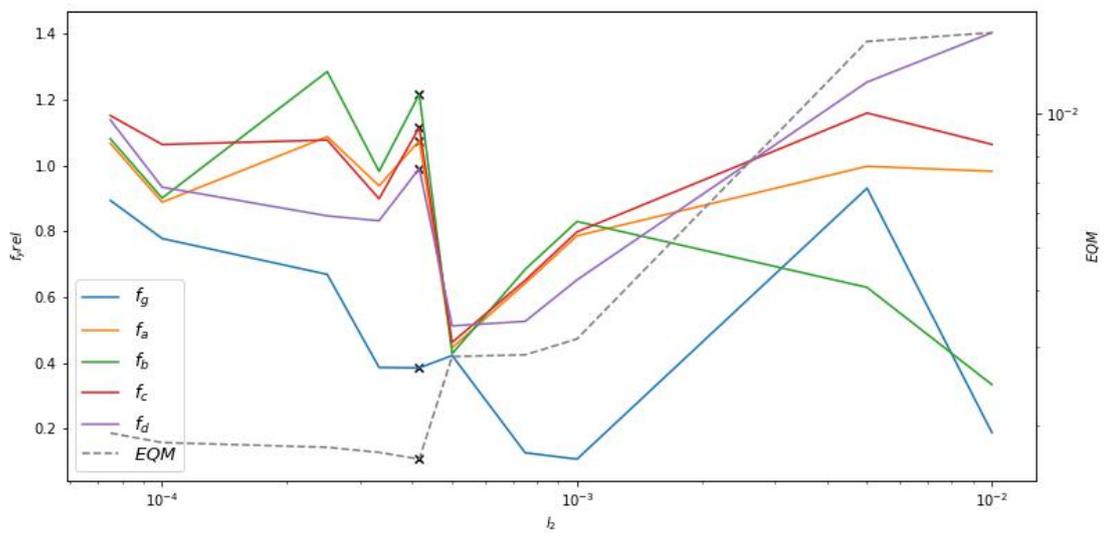
Fonte: produzido pelo autor.

Tabela 5.6: Violações de Balanço no reator

Redução (%)	Global	A	B	C	D
$\lambda = 0,0$ e $l_2 = 0,0$	64.68	78.03	50.60	75.40	52.32
$\lambda = 0,25$ e $l_2 = 0,0$	97.27	93.48	95.29	93.75	94.44
$\lambda = 0,25$ e $l_2 = 0,0005$	98.95	92.99	94.27	93.03	94.49

Fonte: produzido pelo autor.

Figura 5.15: Desbalanceamentos em função de l_2 para o reator. Melhor valor de l_2 marcado com um x.



Fonte: produzido pelo autor.

Capítulo 6

Conclusão e sugestões para trabalhos futuros

Com base nos resultados apresentados, é possível concluir que as redes autoassociativas foram bem-sucedidas na retificação dos dados, principalmente para variáveis com valores de $\sigma_k/\tilde{\sigma}_k$ mais baixos. As RNs se mostraram capazes de aprender as correlações entre os dados e reduzir a magnitude dos ruídos de medição mesmo na ausência dos modelos fenomenológicos. Além disso, os resultados incluindo penalizações sugerem que o desempenho de redes neuronais pode ser melhorado quando estas são combinadas com as equações de balanço. Contudo, é importante destacar que a técnica utilizada ainda apresenta deficiências que precisam ser resolvidas, para que seja possível implementá-la em uma situação real.

Dentre todas as limitações inerentes à abordagem escolhida, a dificuldade em se avaliar o algoritmo na ausência de dados reais é sem dúvida a mais proeminente. Embora o treinamento da rede seja realizado de forma não-

supervisionada, os valores verdadeiros das variáveis foram imprescindíveis para definir o critério de parada e impedir que o algoritmo fosse treinado a ponto de reproduzir as próprias medições. Além disso, os dados reais também foram utilizados para avaliar os hiperparâmetros do modelo, como o número de neurônios das camadas internas, λ e l_2 .

Em uma situação real onde valores verdadeiros não são conhecidos, seria necessário utilizar alguma estratégia alternativa para avaliar os hiperparâmetros e o critério de parada. Uma opção viável para este último seria estimar os desvios padrão desejados das variáveis e calcular σ_{rel} de parada. Isso possivelmente viabilizaria a reconciliação às custas da inserção de um certo grau de viés nos valores estimados.

Também é válido destacar que os resultados obtidos para as curvas de desempenho dos hiperparâmetros foram bastante ruidosos. Para se ter uma visão mais acurada da sensibilidade do modelo com relação a eles, seria válido refinar as curvas com mais pontos e utilizar validação cruzada no cálculo de cada um deles.

Com base nos resultados observados, sugere-se para trabalhos futuros avaliar a utilização de múltiplos modelos para realizar a reconciliação dos dados em sistemas complexos. Ao se abordar sistemas com uma grande quantidade de variáveis, é possível que modificações nos parâmetros do modelo contribuam positivamente para estimar um grupo de variáveis e negativamente para estimar outro. Neste caso, é interessante estudar a sensibilidade das variáveis a estes parâmetros de forma separada, e utilizar as configurações mais favoráveis para reconciliar cada grupo. Além disso, devido à característica das redes autoassociativas de atuar como um PCA não-linear, é possível que componentes principais diferentes influenciem variáveis diferentes de forma

distinta, reforçando a hipótese de que pode ser vantajoso separar as variáveis em grupos para realizar a reconciliação.

De forma geral, parece haver ainda um longo caminho a ser percorrido para que algoritmos baseados em Inteligência Artificial estejam à altura das técnicas clássicas na resolução do problema de RD. Todavia, sua versatilidade aliada à crescente disponibilidade de dados industriais tem feito com que eles se tornem cada vez mais uma opção a ser estudada. O presente trabalho ajuda a reforçar a ideia de que combinar abordagens baseadas em IA com modelos fenomenológicos e técnicas de regularização pode ser uma alternativa promissora para melhorar o desempenho dessas técnicas e torná-las vantajosas frente aos métodos clássicos.

Bibliografia

- [1] Definição de "overfitting" (sobreajuste) em oxforddictionaries.com. OBS: Essa definição é exclusiva para o campo da estatística.
- [2] ABADI, M., AGARWAL, A., BARHAM, P., BREVDO, E., CHEN, Z., CITRO, C., CORRADO, G., DAVIS, A., DEAN, J., DEVIN, M., GHEMAWAT, S., GOODFELLOW, I., HARP, A., IRVING, G., ISARD, M., JIA, Y., JOZEFOWICZ, R., KAISER, L., KUDLUR, M., LEVENBERG, J., MANÉ, D., MONGA, R., MOORE, S., MURRAY, D., OLAH, C., SCHUSTER, M., SHLENS, J., STEINER, B., SUTSKEVER, I., TALWAR, K., TUCKER, P., VANHOUCKE, V., VASUDEVAN, V., VIÉGAS, F., VINYALS, O., WARDEN, P., WATTENBERG, M., WICKE, M., YU, Y., E ZHENG, X. *Tensorflow: Large-scale machine learning on heterogeneous distributed systems*, 2015.
- [3] AGGARWA, C. *Neural Networks and Deep Learning*. Springer, 2018.
- [4] ALBUQUERQUE, J. S., E BIEGLER, L. T. Data reconciliation and gross-error detection for dynamic systems. *AIChE Journal* 42, 10 (1996), 2841–2856.
- [5] ARTHUR, C. Tech giants may be huge, but nothing matches big data. *The Guardian* (2013).

- [6] AVRIEL, M. *Nonlinear programming : analysis and methods*. Prentice Hall, 1976.
- [7] BAGAJEWICZ, M. A brief review of recent developments in data reconciliation and gross error detection/estimation. *Latin American Applied Research* 30 (10 2000), 335–342.
- [8] BAI, S., MCLEAN, D., E THIBAUT, J. Autoassociative neural networks for robust dynamic data reconciliation. *AIChE Journal* 53 (2007), 438 – 448.
- [9] BAI, S., THIBAUT, J., E MCLEAN, D. D. Use of an autoassociative neural network for dynamic data reconciliation. *IFAC Proceedings Volumes* 38, 1 (2005), 65 – 70. 16th IFAC World Congress.
- [10] COELHO, J. *Reconciliação de Dados Aplicada a um Processo Descontínuo de Polimerização em Suspensão*. Dissertação de mestrado, COPPE/UFRJ, 2015.
- [11] CROWE, C. M. Reconciliation of process flow rates by matrix projection. part ii: The nonlinear case. *AIChE Journal* 32 (1986), 616–623.
- [12] CROWE, C. M. Observability and redundancy of process data for steady state reconciliation. *Chemical Engineering Science* 44, 12 (1989), 2909 – 2917.
- [13] CROWE, C. M., GARCIA CAMPOS, Y., E HRYMAK, A. Reconciliation of process flow rates by matrix projection. part i: Linear case. *AIChE Journal* 29 (1983), 881–888.

- [14] DE SOUZA JR, M. *Redes Neurais Multicamadas Aplicadas a Modelagem e Controle de Processos Químicos*. Tese de doutorado, COPPE/UFRJ, 1993.
- [15] DU, Y.-G., HODOUIN, D., E THIBAUT, J. Use of a novel autoassociative neural network for nonlinear steady-state data reconciliation. *AIChE Journal* 43, 7 (1997), 1785–1796.
- [16] FELDMAN, R. *Reconciliação De Dados Em Tempo Real Para Monitoração e Detecção de Falhas em Terminal de Transporte e Armazenamento de Derivados de Petróleo*. Dissertação de mestrado, COPPE/UFRJ, 2007.
- [17] FOGLER, F. S. *Elements of Chemical Reaction Engineering*. Prentice Hall PTR, 1999.
- [18] GALANTINE, L., FONSECA, P., AMORIM, V., E SECCHI, A. Simulação da produção de fenilbenzeno, parte 1. Relatório técnico, Escola de Química/UFRJ, 2017.
- [19] GOODFELLOW, I., BENGIO, Y., E COURVILLE, A. *Deep Learning*. MIT Press, 2016.
- [20] HENLEY, E., E SEADER, J. D. *Equilibrium-Stage Operations in Chemical Engineering*. John Wiley & Sons, 1981.
- [21] HIMMELBLAU, D. Applications of artificial neural networks in chemical engineering. *Korean Journal of Chemical Engineering* 17 (07 2000), 373–392.
- [22] HIMMELBLAU, D. M., E KARJALA, T. W. Rectification of data in a dynamic process using artificial neural networks. *Computers & Chemical*

- Engineering* 20, 6 (1996), 805 – 811. Fifth International Symposium on Process Systems Engineering.
- [23] JAMES, G., WITTEN, D., HASTIE, T., E TIBSHIRANI, R. *An Introduction to Statistical Learning*. Springer Science+Business Media, 2013.
- [24] JOHNSTON, L. P. M., E KRAMER, M. A. Maximum likelihood data rectification: Steady-state systems. *AIChE Journal* 41, 11 (1995), 2415–2426.
- [25] KARJALA, T., HIMMELBLAU, D., E MIIKKULAINEN, R. Data rectification using recurrent (elman) neural networks. *IJCNN International Joint Conference on Neural Networks 2* (1992), 901 – 906 vol.2.
- [26] KARJALA, T. W., E HIMMELBLAU, D. M. Dynamic data rectification by recurrent neural networks vs. traditional methods. *AIChE Journal* 40, 11 (1994), 1865–1875.
- [27] KIM, I.-W., EDGAR, T., E BELL, N. Parameter estimation for a laboratory water—gas-shift reactor using a nonlinear error-in-variables method. *Computers & Chemical Engineering* 15, 5 (1991), 361 – 367.
- [28] KINGMA, D., E BA, J. Adam: A method for stochastic optimization. *International Conference on Learning Representations* (12 2014).
- [29] KNEPPER, J. C., E GORMAN, J. W. Statistical analysis of constrained data sets. *AIChE Journal* 26, 2 (1980), 260–264.
- [30] KONG, M., CHEN, B., E LI, B. An integral to dynamic data rectification. *Computers and Chemical Engineering* 24 (2000), 749 – 753.

- [31] KRAMER, M. Autoassociative neural networks. *Computers & Chemical Engineering* 16, 4 (1992), 313 – 328. Neural network applications in chemical engineering.
- [32] KRAMER, M. A. Nonlinear principal component analysis using autoassociative neural networks. *AIChE Journal* 37, 2 (1991), 233–243.
- [33] KUHEN, D. R., E DAVIDSON, H. Computer control. ii. mathematics of control. *Computers & Chemical Engineering* 57 (1961), 44–47.
- [34] LIEBMAN, M. J., EDGAR, T. F., E LASDON, L. S. Efficient reconciliation and estimation for dynamic processes using nonlinear programming techniques. *Computers & Chemical Engineering* 16 (1992), 963–986.
- [35] MAH, R. S., STANLEY, G. M., E DOWNING, D. M. Reconciliation and rectification of process flow and inventory data. *Industrial & Engineering Chemistry Process Design and Development* 15, 1 (1976), 175–183.
- [36] MAH, R. S. H. *Chemical Process Structures and Information Flows*. Butterworth, 1990.
- [37] MORE, J. J., GARBOW, B. S., E HILLSTROM, K. E. *User Guide for MINPACK-1*, 1980.
- [38] NARASIMHAN, S., E JORDACHE, C. *Data Reconciliation and Gross Error Detection: An Intelligent Use of Process Data*. Gulf Professional Publishing, 2000.
- [39] NARASIMHAN, S., E MAH, R. S. H. Generalized likelihood ratios for gross error identification in dynamic processes. *AIChE Journal* 34, 8 (1988), 1321–1331.

- [40] PRATA, D. *Reconciliação Robusta de Dados para Monitoramento em Tempo Real*. Tese de doutorado, COPPE/UFRJ, 2009.
- [41] PSICHOGIOS, D. C., E UNGAR, L. H. Direct and indirect model based control using artificial neural networks. *Industrial & Engineering Chemistry Research* 30, 12 (1991), 2564–2573.
- [42] PSICHOGIOS, D. C., E UNGAR, L. H. Process modeling using structured neural networks. In *1992 American Control Conference (1992)*, pp. 1917–1921.
- [43] RAMAMURTHI, Y., SISTU, P., E BEQUETTE, B. Control-relevant dynamic data reconciliation and parameter estimation. *Computers & Chemical Engineering* 17, 1 (1993), 41 – 59.
- [44] RIPPS, D. L. Adjustment of experimental data. *Chemical Engineering Progress Symposium Series* 61 (1965), 8 – 13.
- [45] SMITH, J. M., VAN NESS, H. C., E ABBOTT, M. M. *Introdução à Termodinâmica da Engenharia Química*. LTC, 2007.
- [46] SOARES, R. P., E SECCHI, A. R. Ems: A new environment for modelling, simulation and optimisation. In *European Symposium on Computer Aided Process Engineering-13*, A. Kraslawski e I. Turunen, Eds., vol. 14 of *Computer Aided Chemical Engineering*. Elsevier, 2003, pp. 947 – 952.
- [47] TERRY, P. A., E HIMMELBLAU, D. M. Data rectification and gross error detection in a steady state process via artificial neural networks. *Industrial & Engineering Chemistry Research* 32, 12 (1993), 3020–3028.

- [48] THOMSON, G. W. The antoine equation for vapor-pressure data. *Chemical Reviews* 38, 1 (1946), 1–39.
- [49] TJOA, I., E BIEGLER, L. Simultaneous solution and optimization strategies for parameter estimation of differential-algebraic equation systems. *Industrial & Engineering Chemistry Research* 30, 2 (1991), 376 – 385.
- [50] TJOA, I., E BIEGLER, L. Simultaneous strategies for data reconciliation and gross error detection of nonlinear systems. *Computers & Chemical Engineering* 15, 10 (1991), 679 – 690.
- [51] VAN DER WALT, S., COLBERT, S. C., E VAROQUAUX, G. The numpy array: A structure for efficient numerical computation. *Computing in Science Engineering* 13, 2 (2011), 22–30.
- [52] VIRTANEN, P., GOMMERS, R., OLIPHANT, T. E., HABERLAND, M., REDDY, T., COURNAPEAU, D., BUROVSKI, E., PETERSON, P., WECKESSER, W., BRIGHT, J., VAN DER WALT, S. J., BRETT, M., WILSON, J., JARROD MILLMAN, K., MAYOROV, N., NELSON, A. R. J., JONES, E., KERN, R., LARSON, E., CAREY, C., POLAT, İ., FENG, Y., MOORE, E. W., VAND ERPLAS, J., LAXALDE, D., PERKTOLD, J., CIMRMAN, R., HENRIKSEN, I., QUINTERO, E. A., HARRIS, C. R., ARCHIBALD, A. M., RIBEIRO, A. H., PEDREGOSA, F., VAN MULBREGT, P., E CONTRIBUTORS, S. . . SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods* 17 (2020), 261–272.

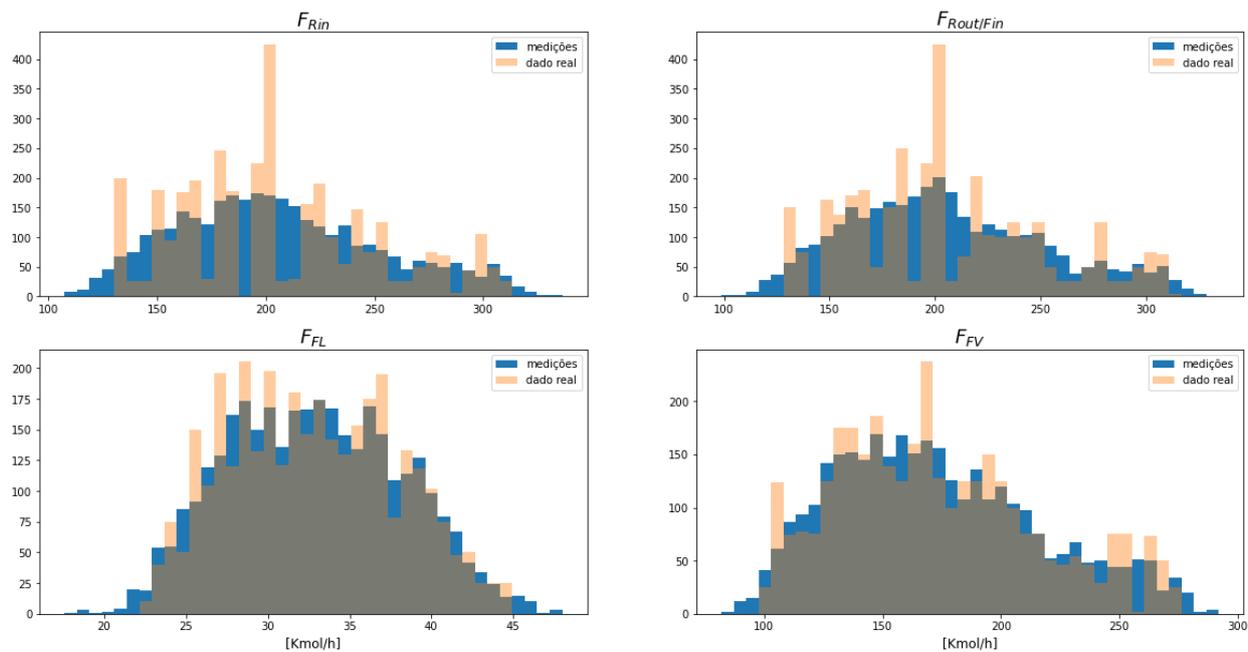
- [53] WAGNER, W. New vapour pressure measurements for argon and nitrogen and a new method for establishing rational vapour pressure equations. *Cryogenics* 13, 8 (1973), 470 – 482.
- [54] WILSON, D., AGARWAL, M., E RIPPIN, D. Experiences implementing the extended kalman filter on an industrial batch reactor. *Computers & Chemical Engineering* 22, 11 (1998), 1653 – 1672.
- [55] WONGRAT, W., SRINOPHAKUN, T., E SRINOPHAKUN, P. Modified genetic algorithm for nonlinear data reconciliation. *Computers & Chemical Engineering* 29, 5 (2005), 1059 – 1067.
- [56] ZEILER, M. D. ADADELTA: an adaptive learning rate method. *CoRR abs/1212.5701* (2012).

Apêndice A

Resultados e distribuição dos dados

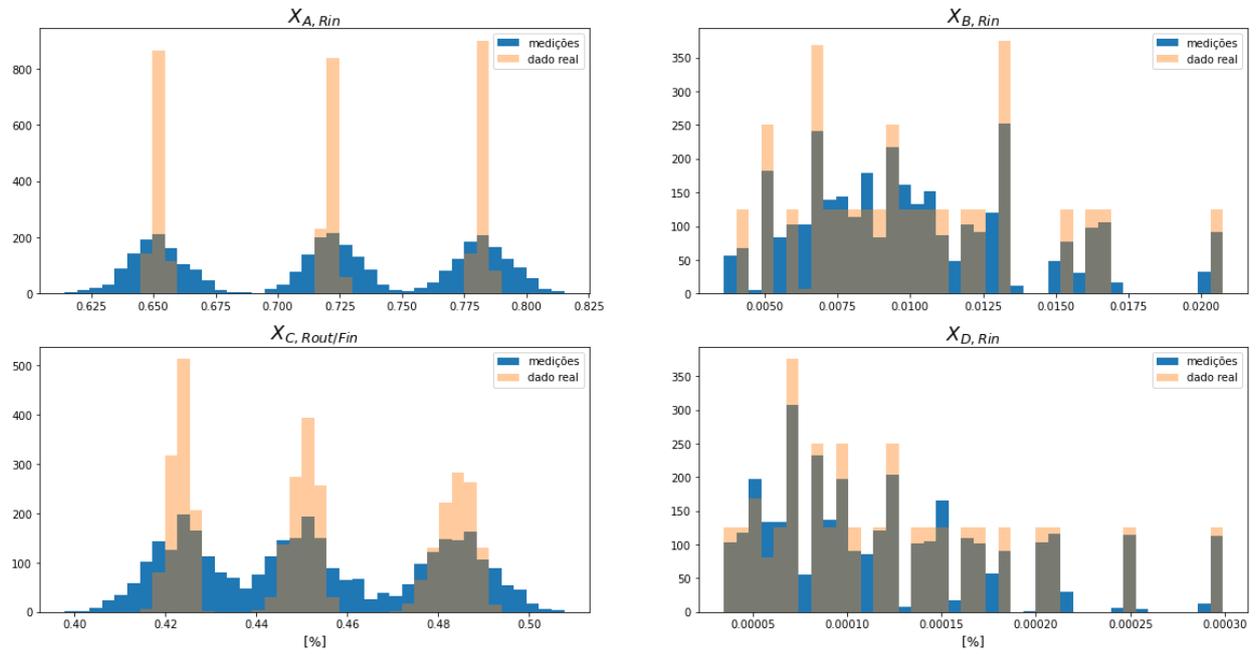
A.1 Distribuição das variáveis

Figura A.1: Distribuição das vazões molares.



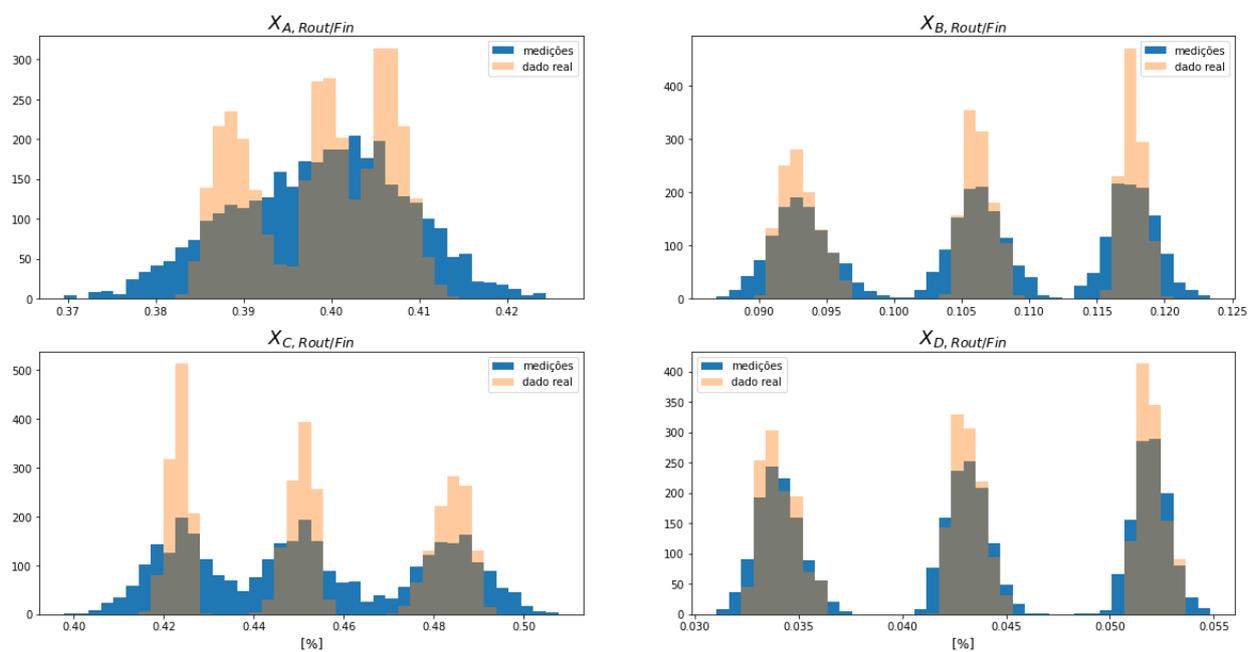
Fonte: produzido pelo autor.

Figura A.2: Distribuição das composições na entrada do reator.



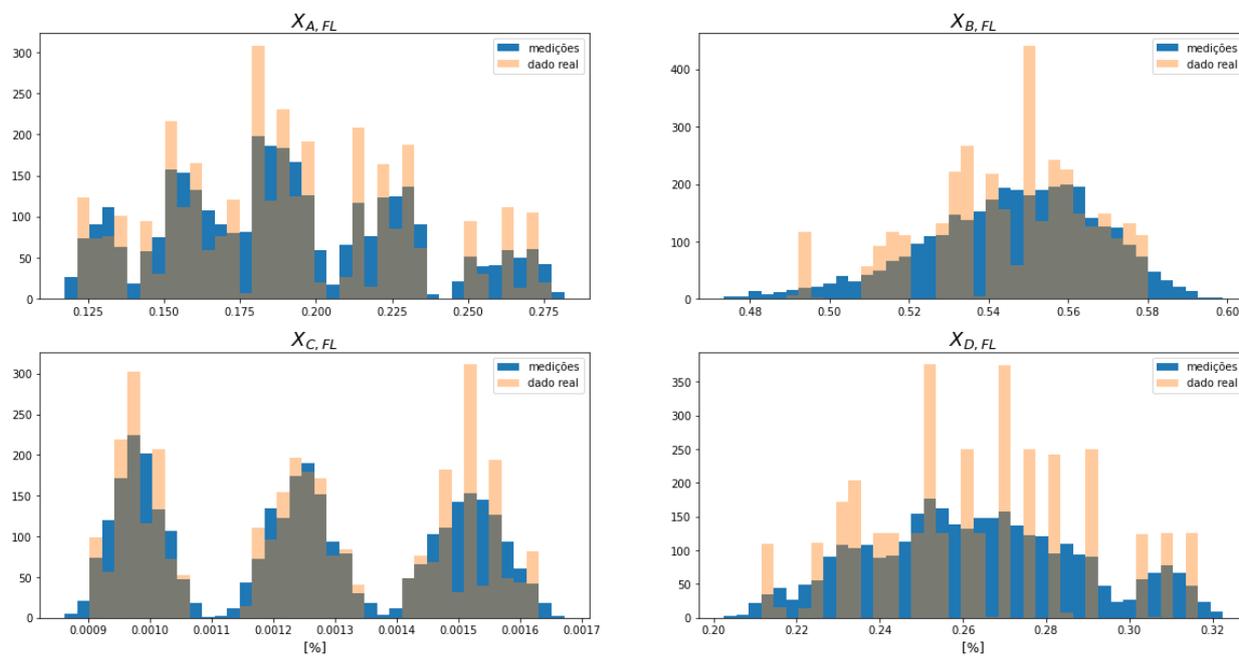
Fonte: produzido pelo autor.

Figura A.3: Distribuição das composições na saída do reator/entrada do flash.



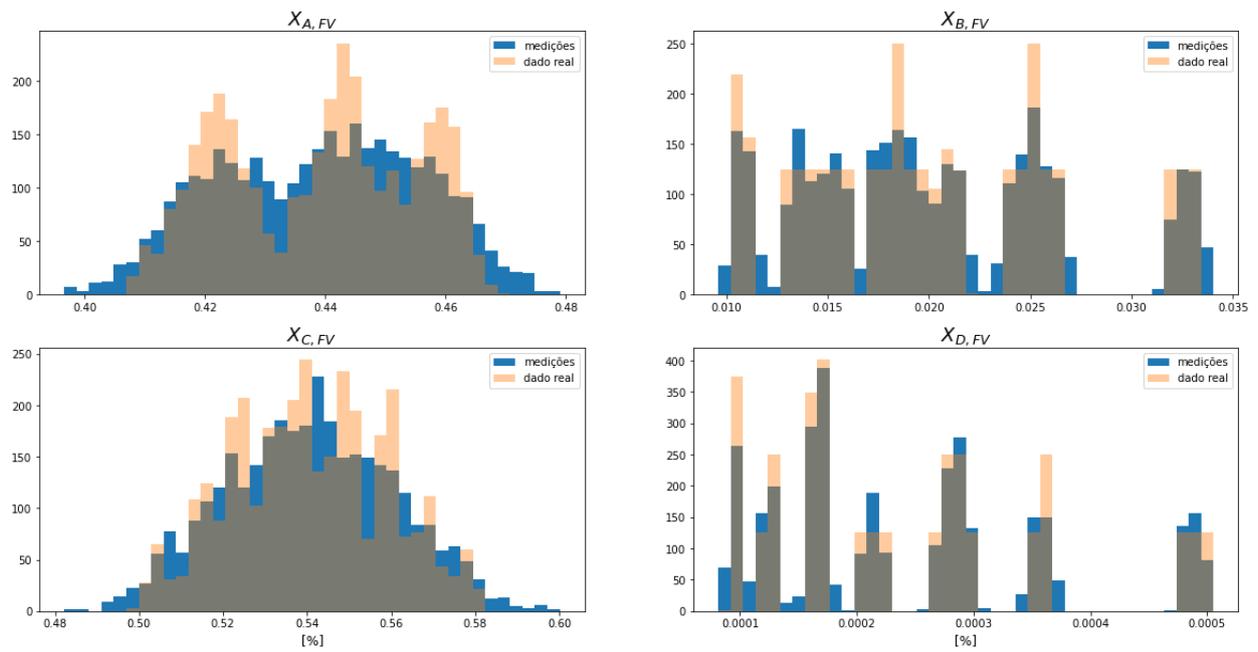
Fonte: produzido pelo autor.

Figura A.4: Distribuição das composições na corrente líquida de saída do flash.



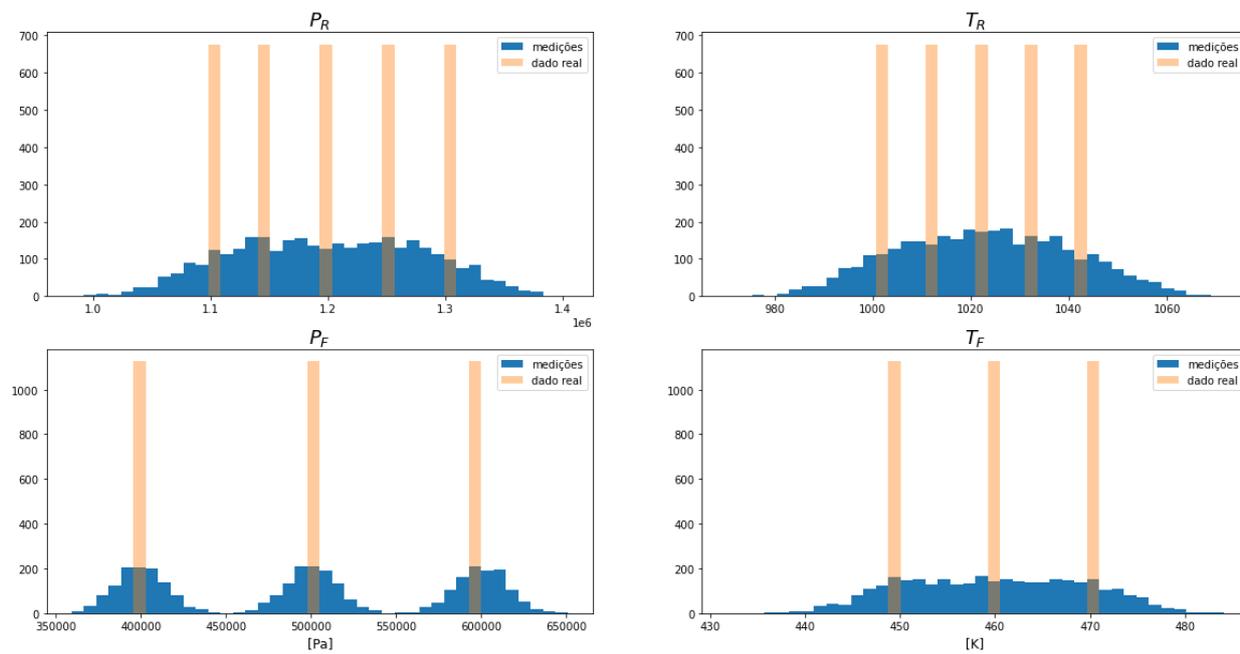
Fonte: produzido pelo autor.

Figura A.5: Distribuição das composições na corrente gasosa de saída do flash.



Fonte: produzido pelo autor.

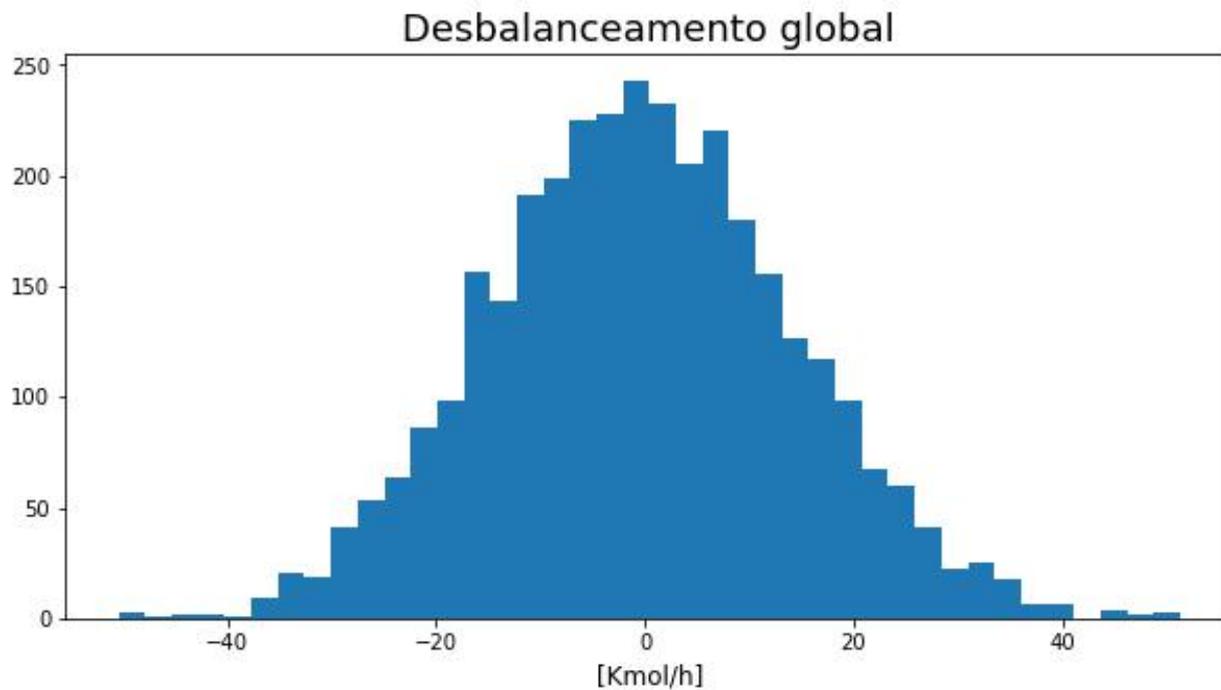
Figura A.6: Distribuição de pressão e temperatura no reator e flash.



Fonte: produzido pelo autor.

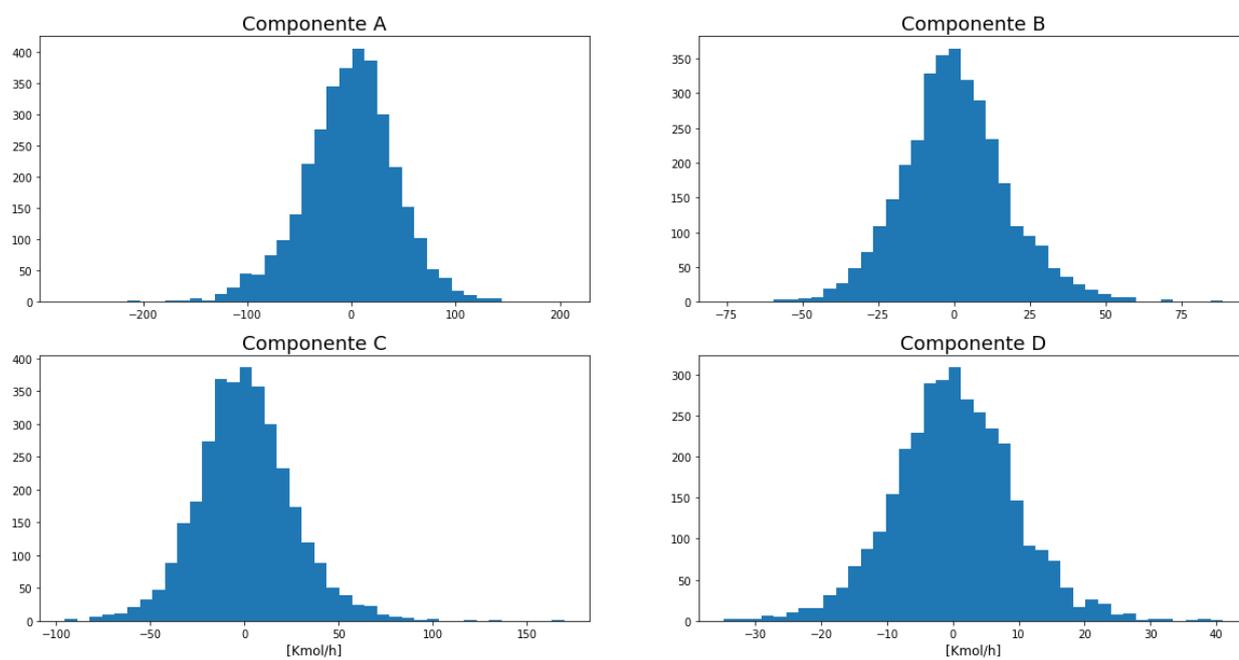
A.2 Distribuição dos desbalanceamentos

Figura A.7: Distribuição das violações no balanço de massa global no reator.



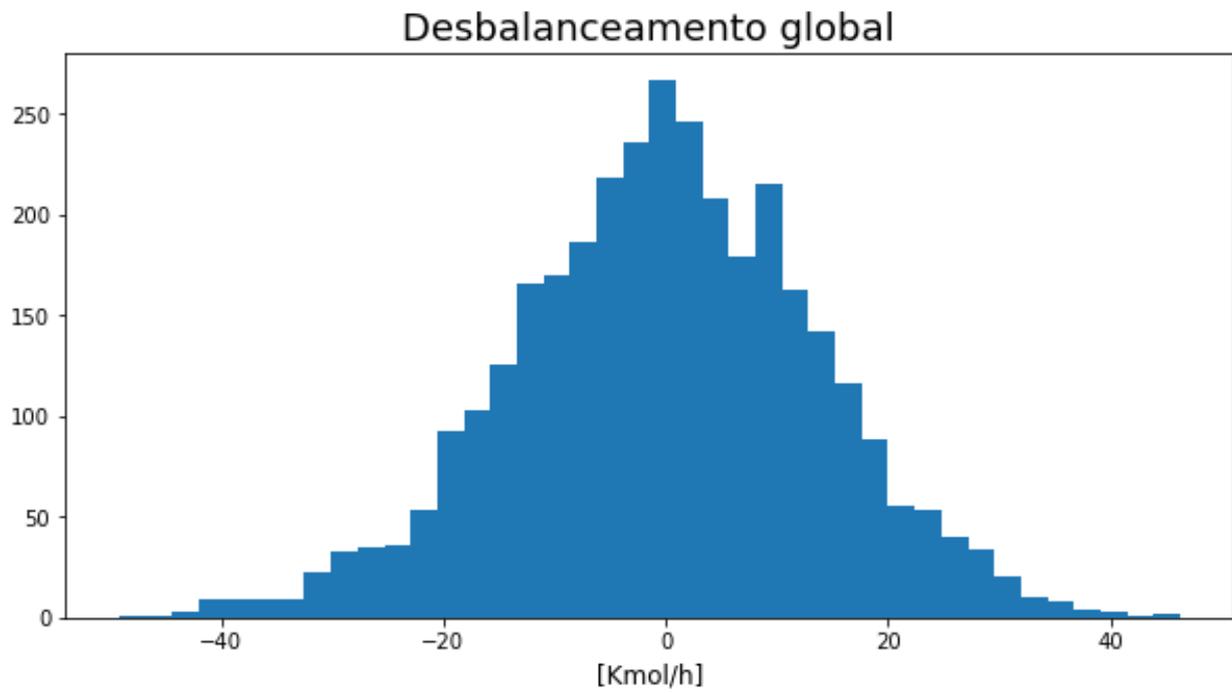
Fonte: produzido pelo autor.

Figura A.8: Distribuição das violações no balanço de massa por componente no reator.



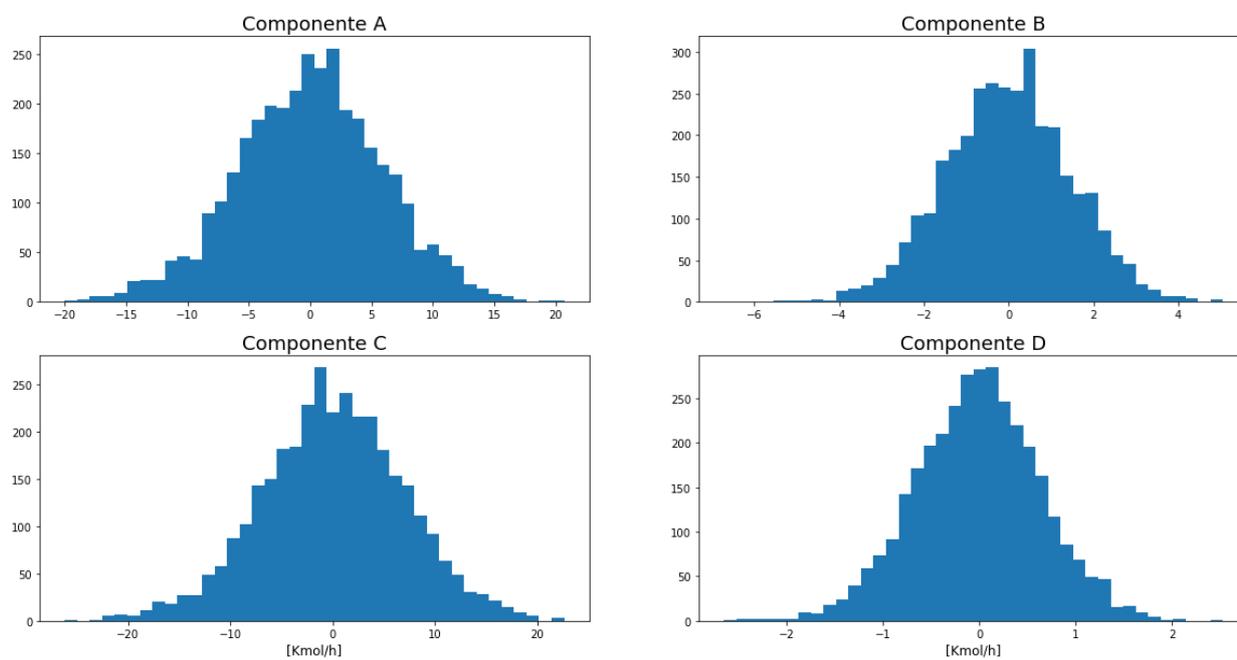
Fonte: produzido pelo autor.

Figura A.9: Distribuição das violações no balanço de massa global no flash.



Fonte: produzido pelo autor.

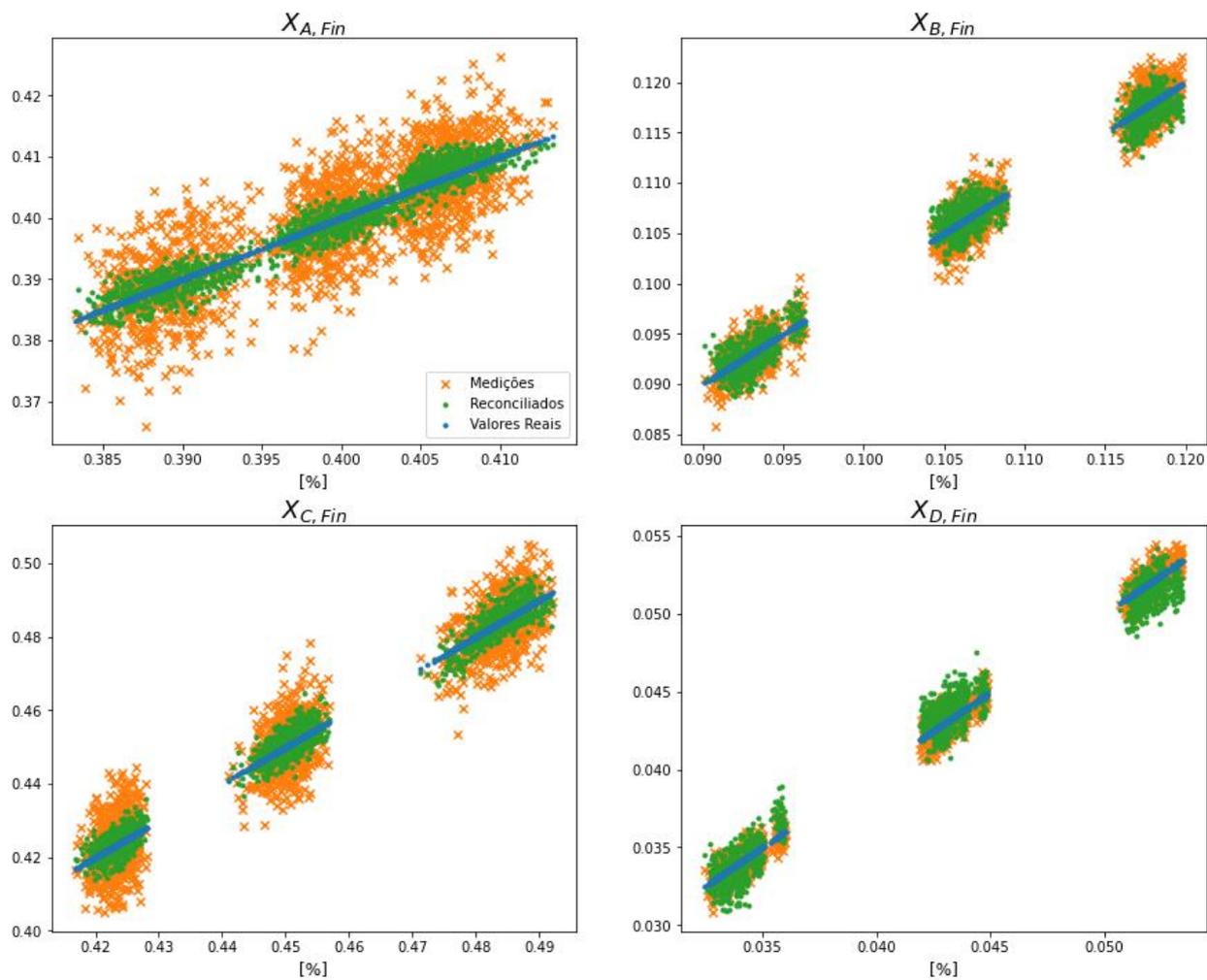
Figura A.10: Distribuição das violações no balanço de massa por componente no flash.



Fonte: produzido pelo autor.

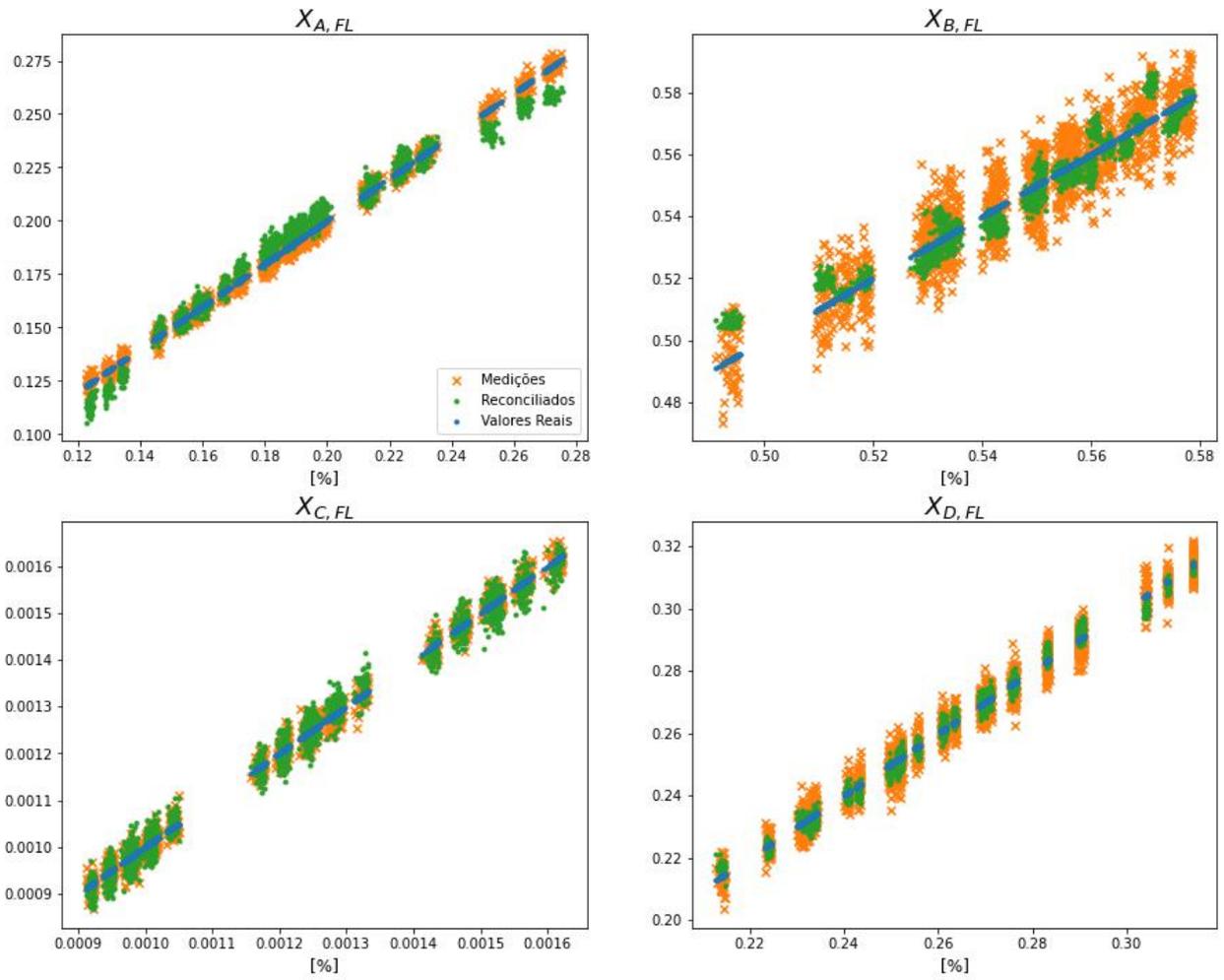
A.3 Flash com $\lambda = 0$ e $l_2 = 0$

Figura A.11: Resultado da reconciliação para o flash com $\lambda = 0$ e $l_2 = 0$.



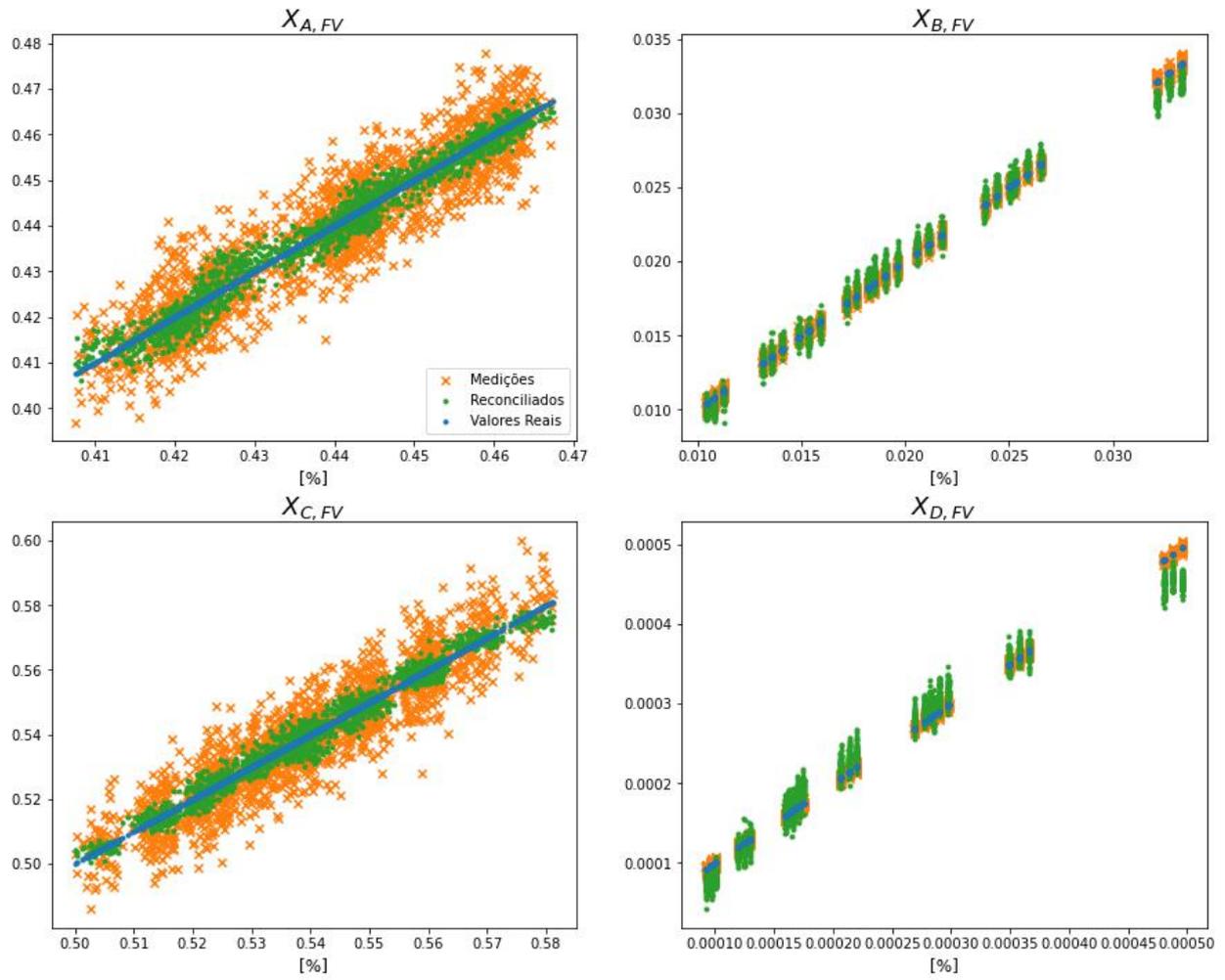
Fonte: produzido pelo autor.

Figura A.12: Resultado da reconciliação para o flash com $\lambda = 0$ e $l_2 = 0$.



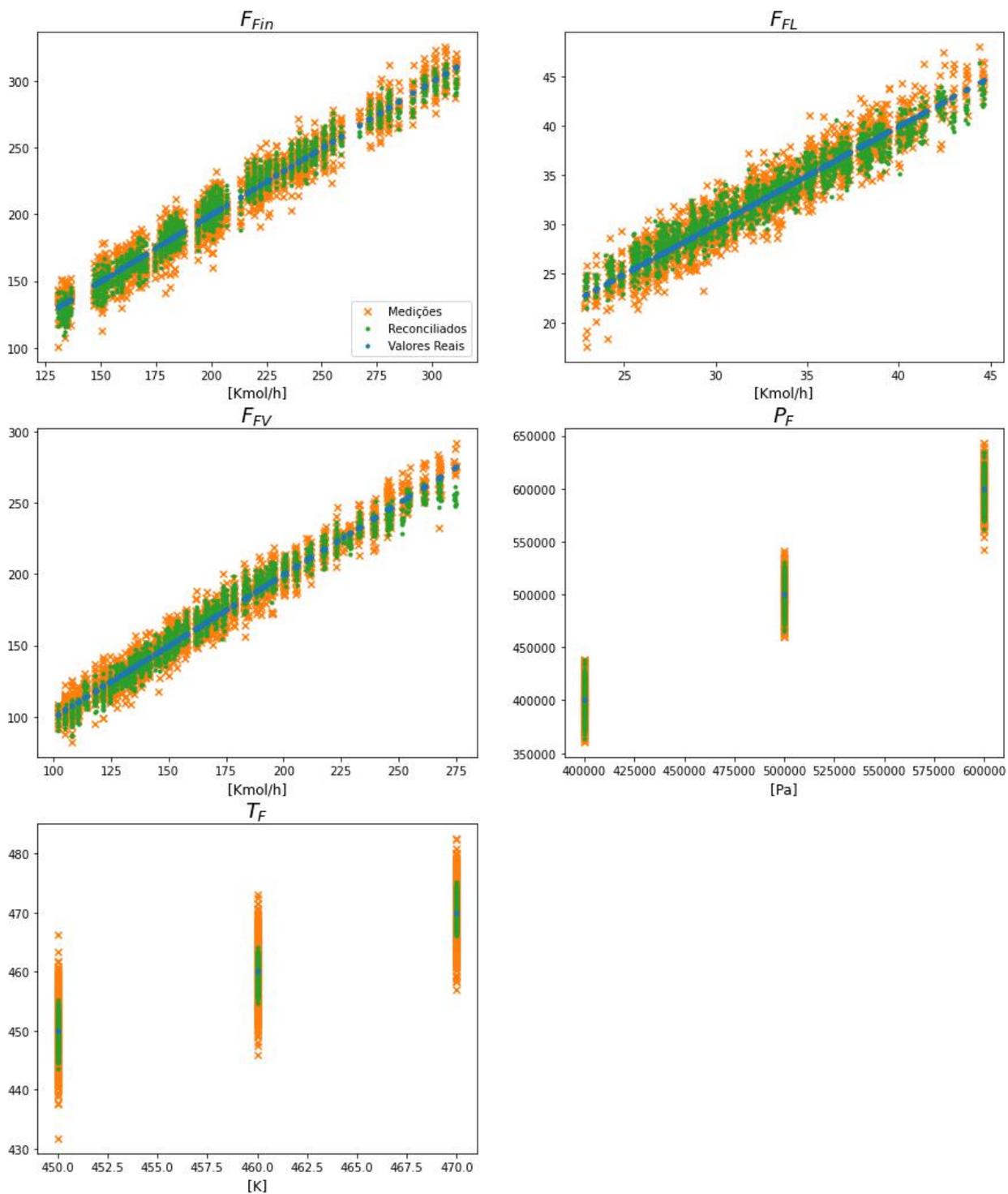
Fonte: produzido pelo autor.

Figura A.13: Resultado da reconciliação para o flash com $\lambda = 0$ e $l_2 = 0$.



Fonte: produzido pelo autor.

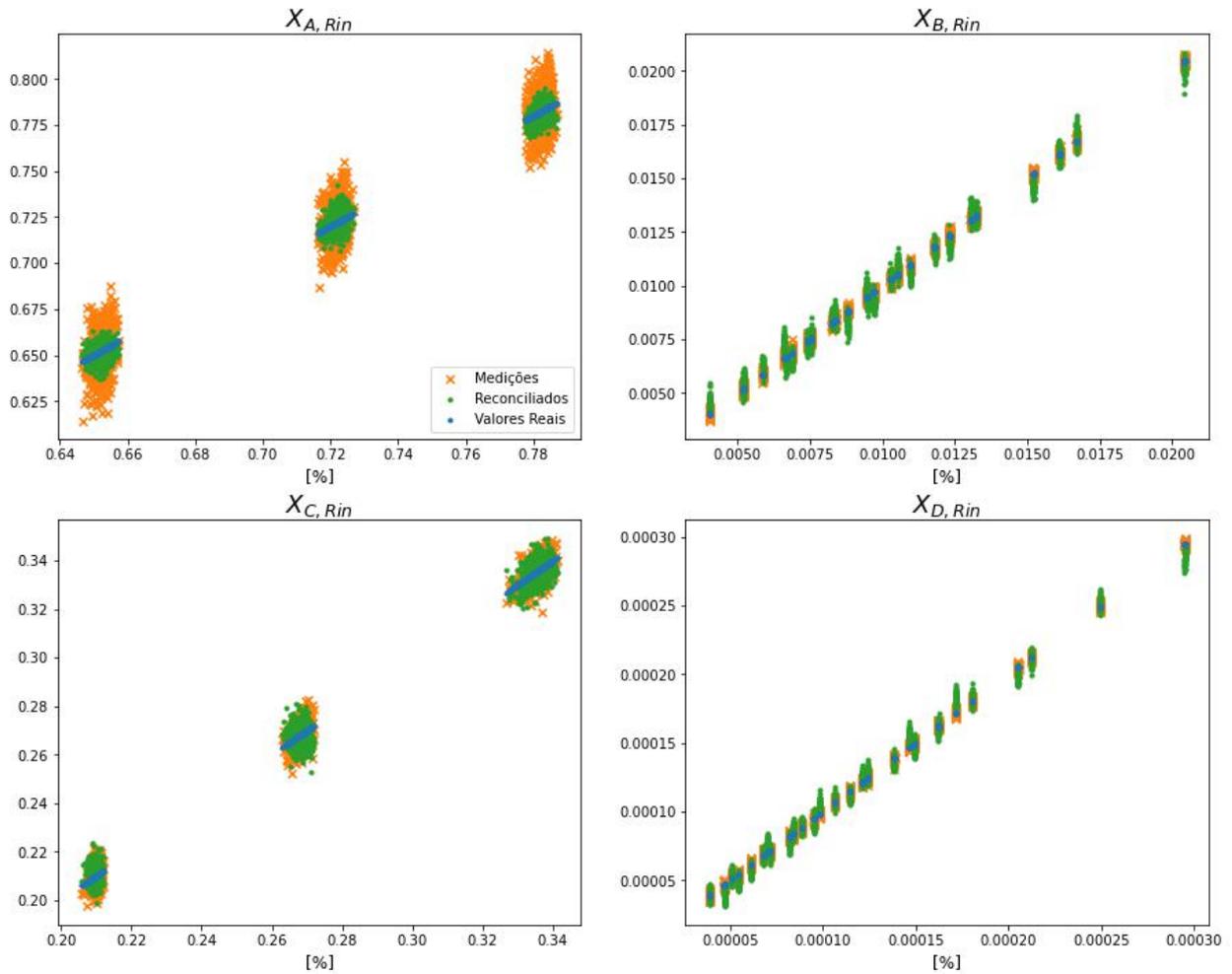
Figura A.14: Resultado da reconciliação para o flash com $\lambda = 0$ e $l_2 = 0$.



Fonte: produzido pelo autor.

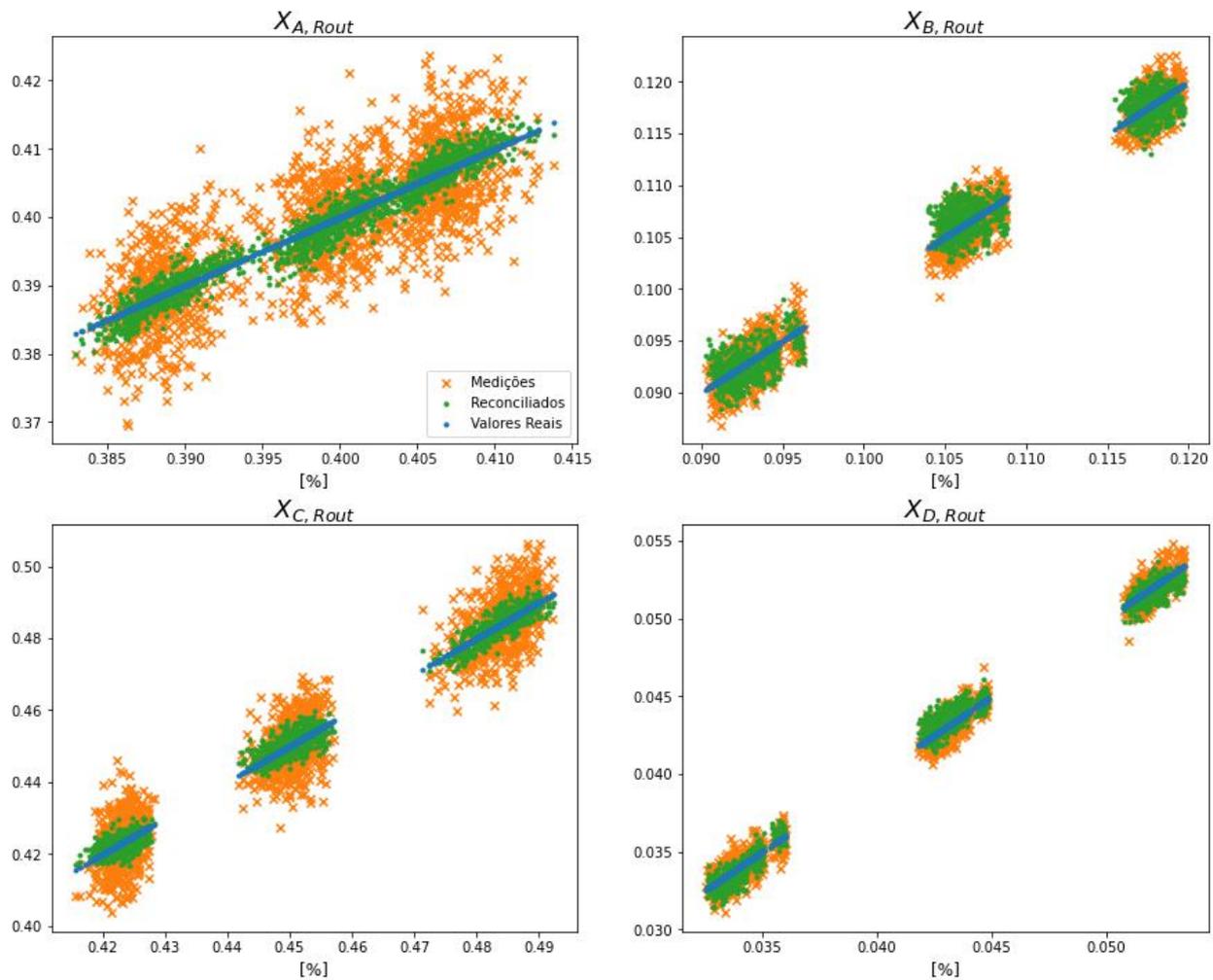
A.4 Reator com $\lambda = 0$ e $l_2 = 0$

Figura A.15: Resultado da reconciliação para o reator com $\lambda = 0$ e $l_2 = 0$.



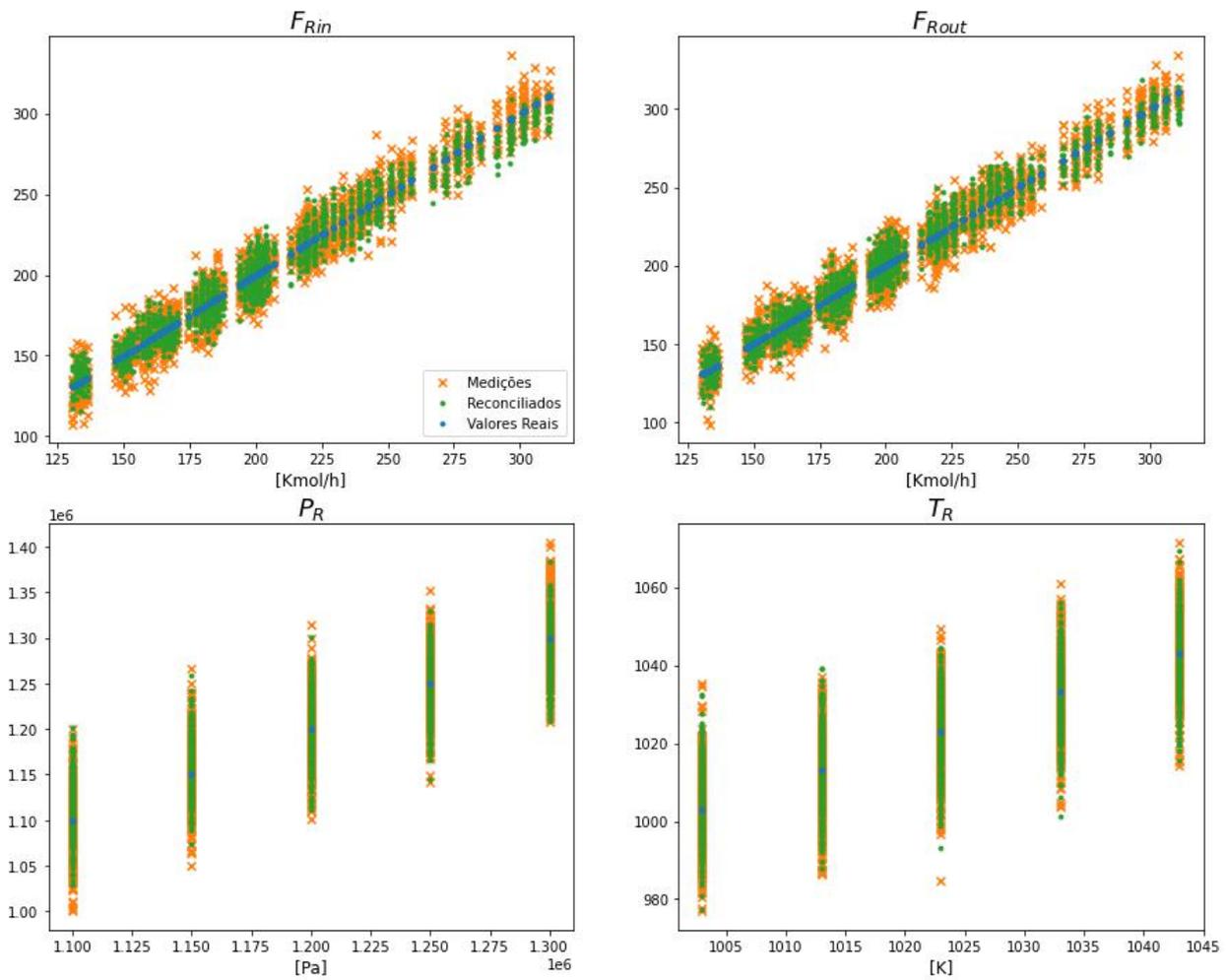
Fonte: produzido pelo autor.

Figura A.16: Resultado da reconciliação para o reator com $\lambda = 0$ e $l_2 = 0$.



Fonte: produzido pelo autor.

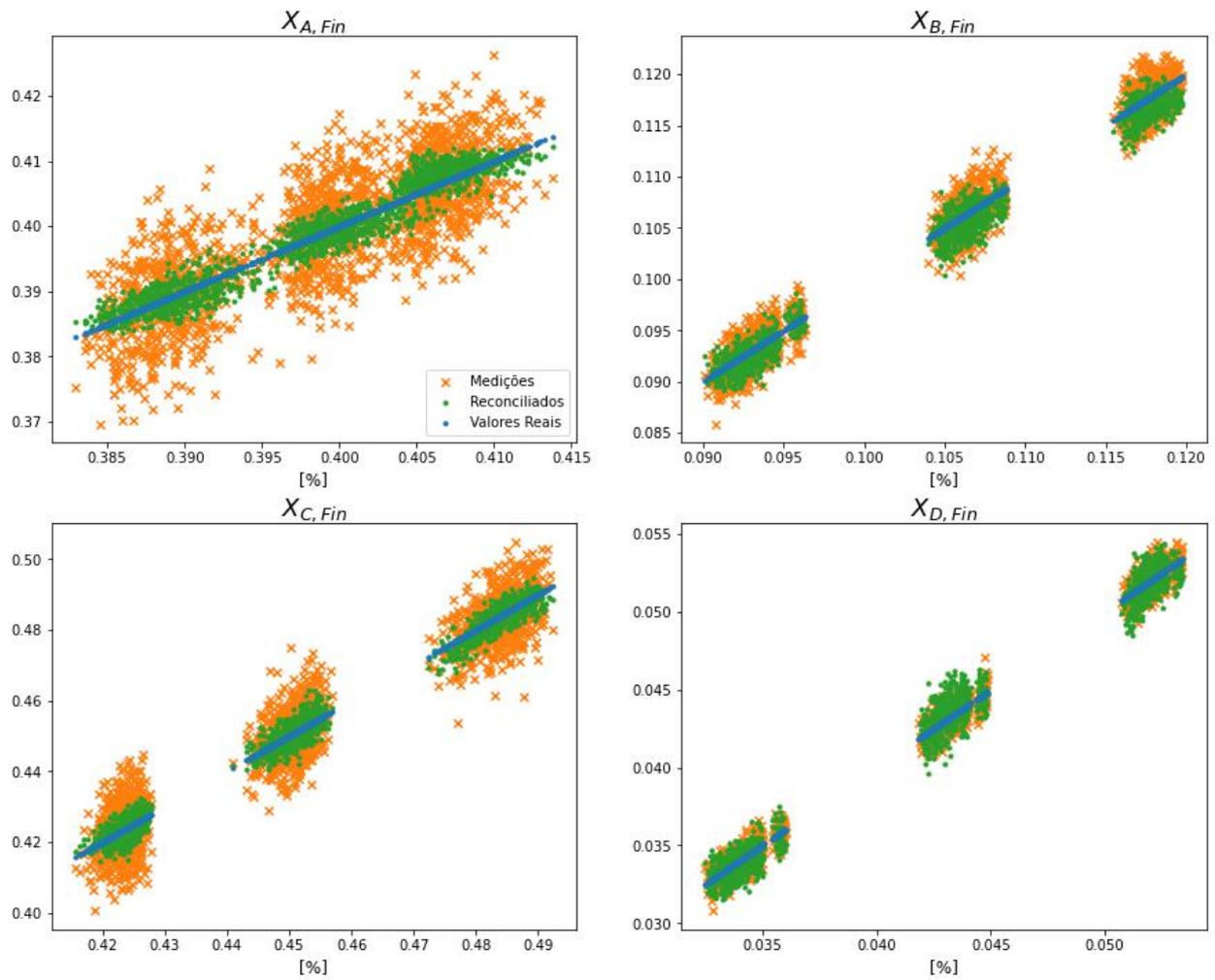
Figura A.17: Resultado da reconciliação para o reator com $\lambda = 0$ e $l_2 = 0$.



Fonte: produzido pelo autor.

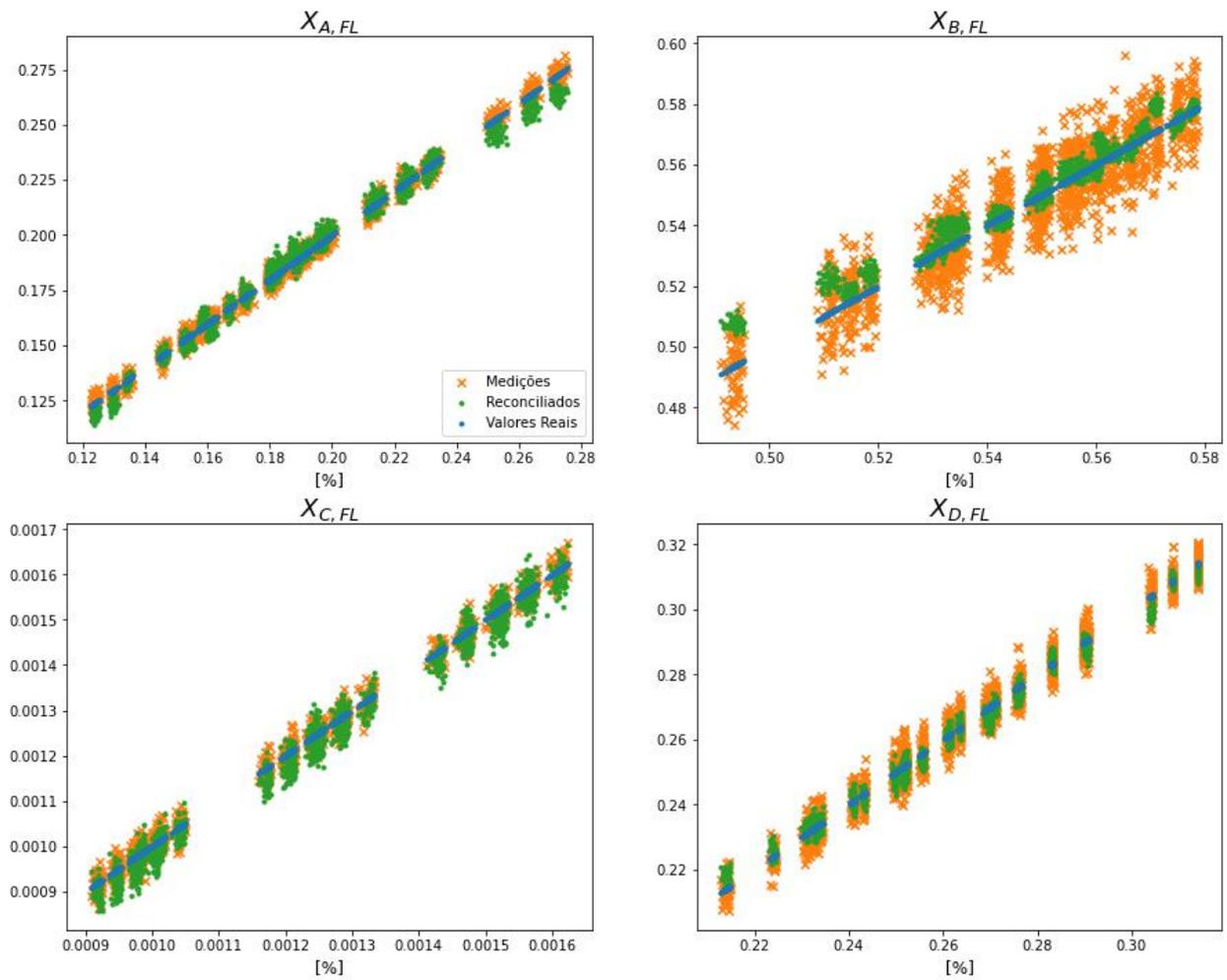
A.5 Flash com $\lambda = 0,25$ e $l_2 = 0$

Figura A.18: Resultado da reconciliação para o flash com $\lambda = 0,25$ e $l_2 = 0$.



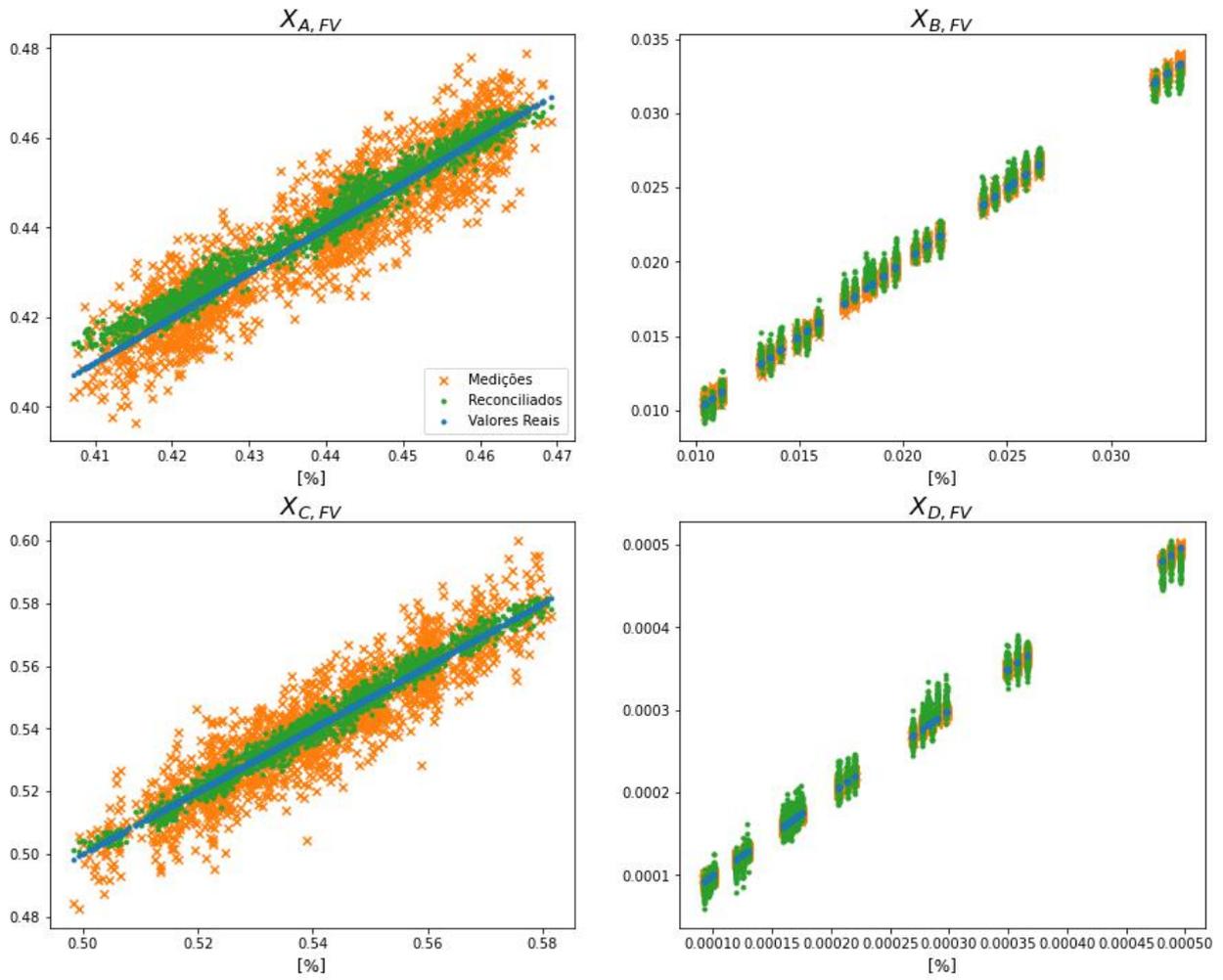
Fonte: produzido pelo autor.

Figura A.19: Resultado da reconciliação para o flash com $\lambda = 0,25$ e $l_2 = 0$.



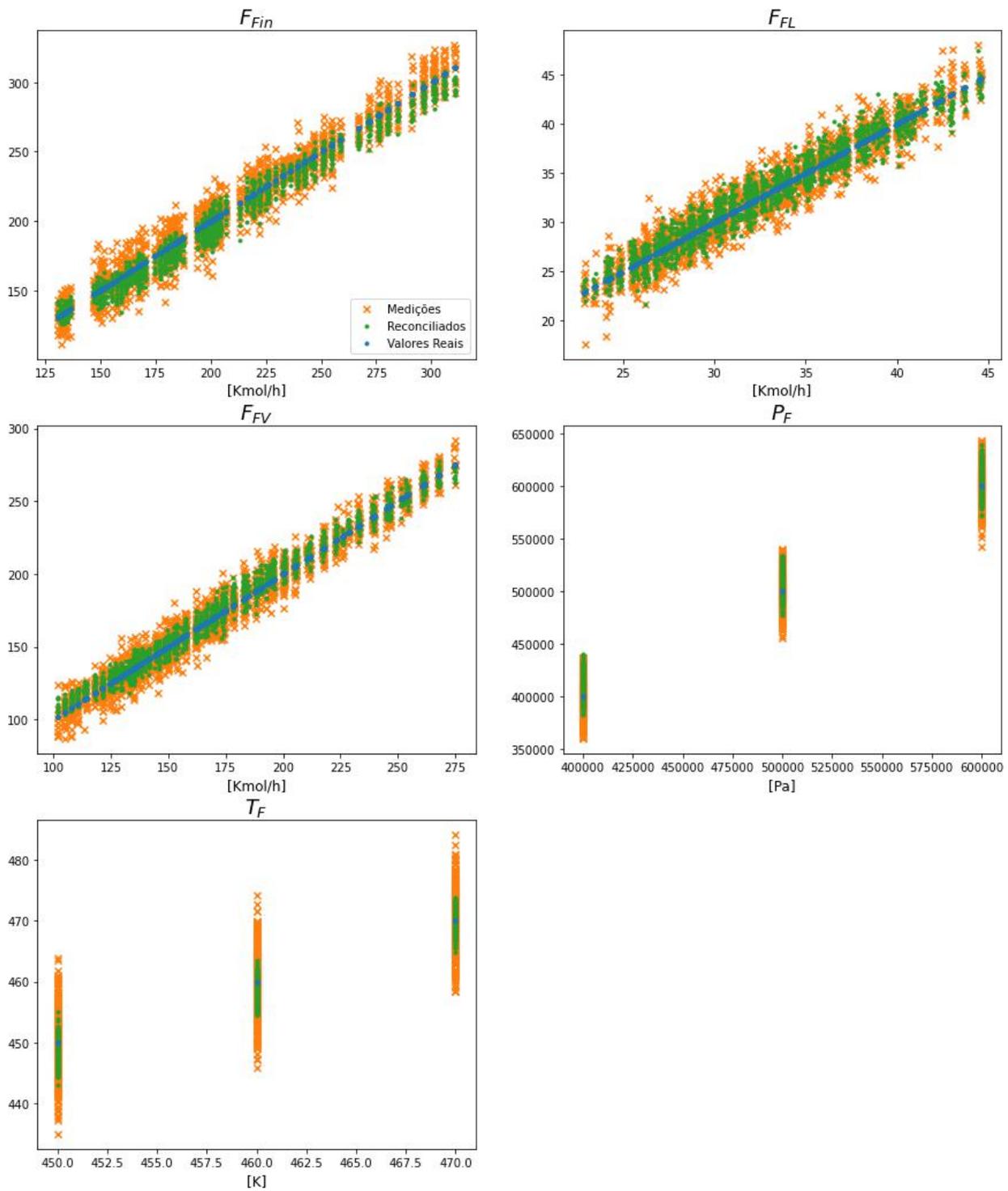
Fonte: produzido pelo autor.

Figura A.20: Resultado da reconciliação para o flash com $\lambda = 0,25$ e $l_2 = 0$.



Fonte: produzido pelo autor.

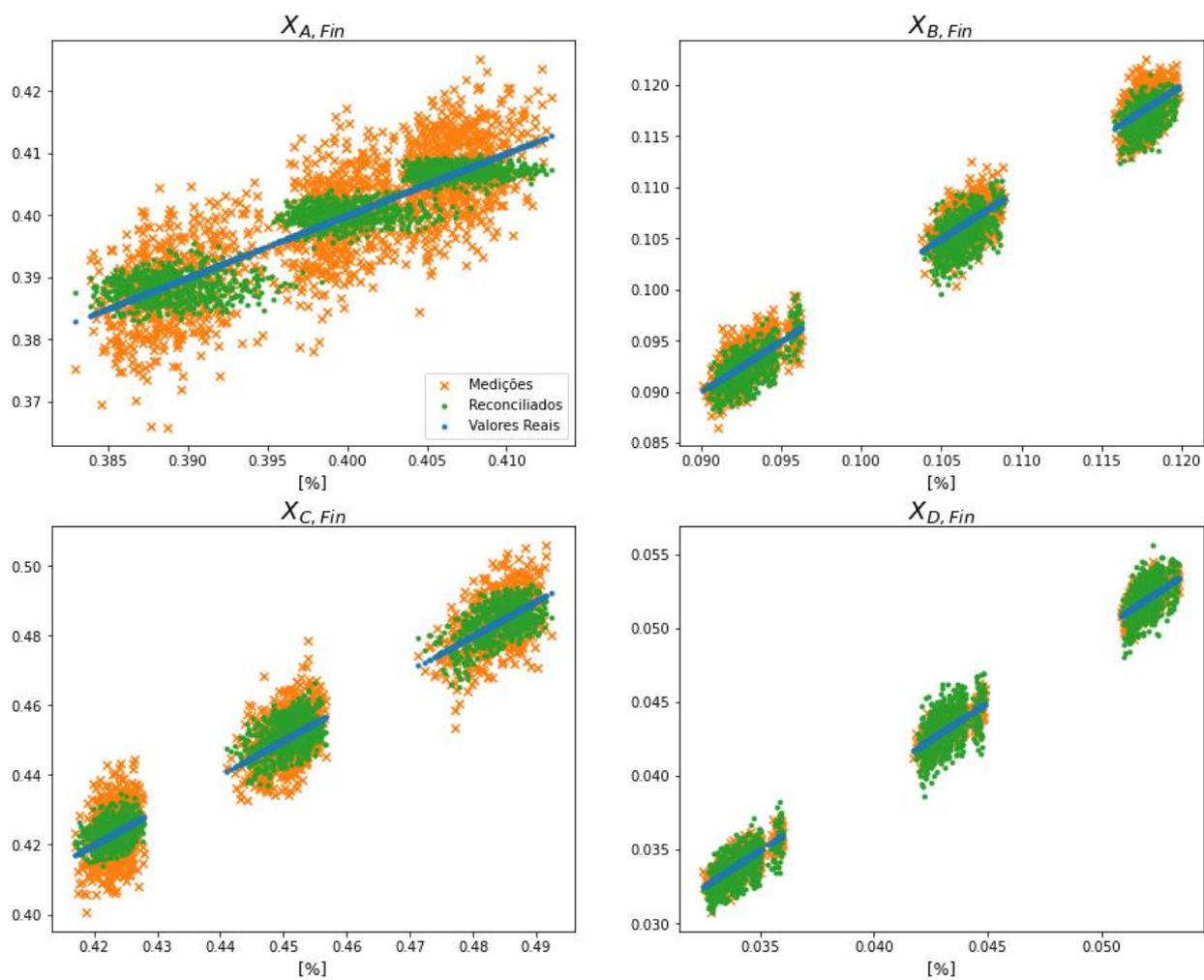
Figura A.21: Resultado da reconciliação para o flash com $\lambda = 0,25$ e $l_2 = 0$.



Fonte: produzido pelo autor.

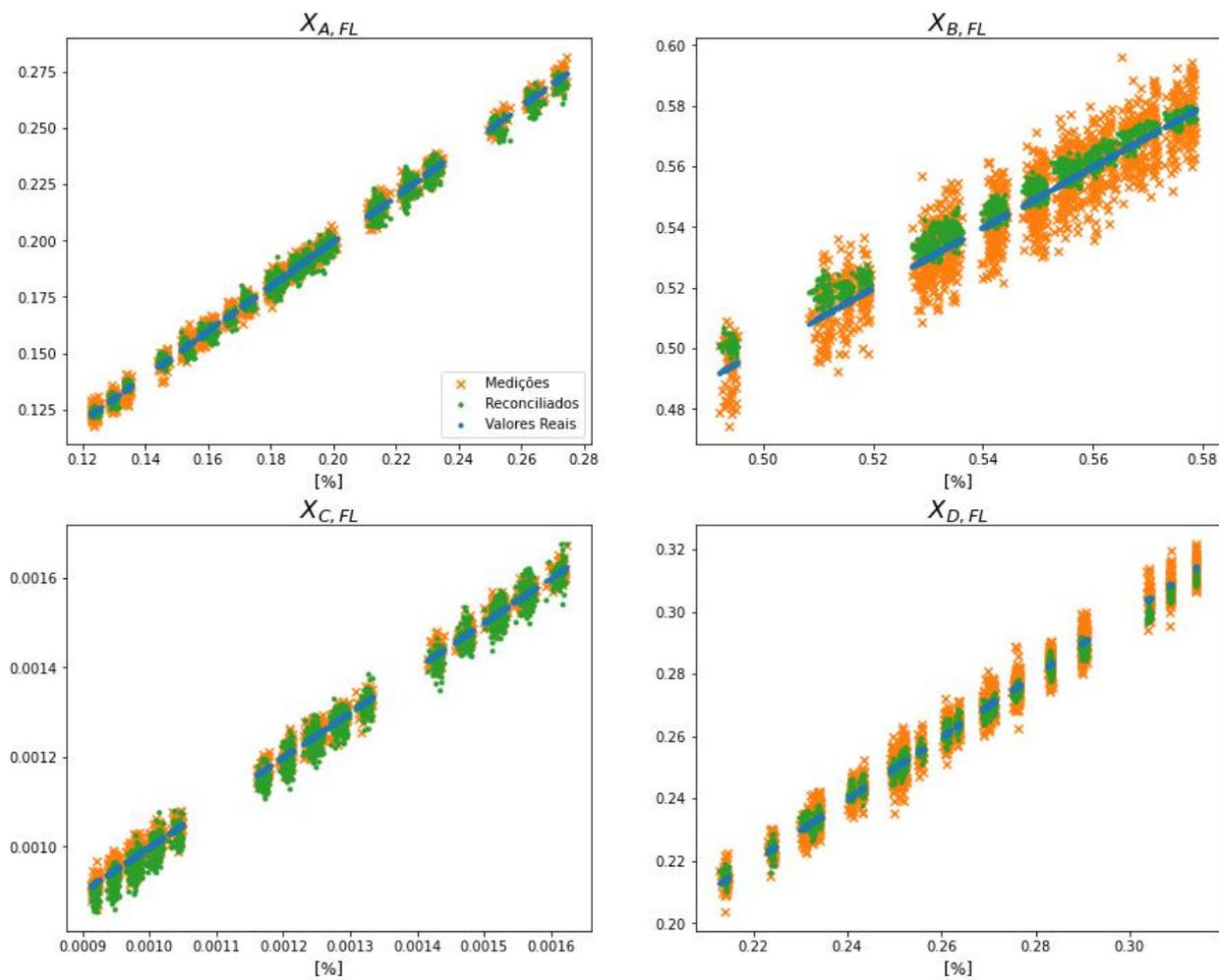
A.6 Flash com $\lambda = 0,25$, 4 neurônios na camada de gargalo e $l_2 = 0$.

Figura A.22: Resultado da reconciliação para o flash com 4 neurônios na camada de gargalo, $\lambda = 0,25$ e $l_2 = 0$.



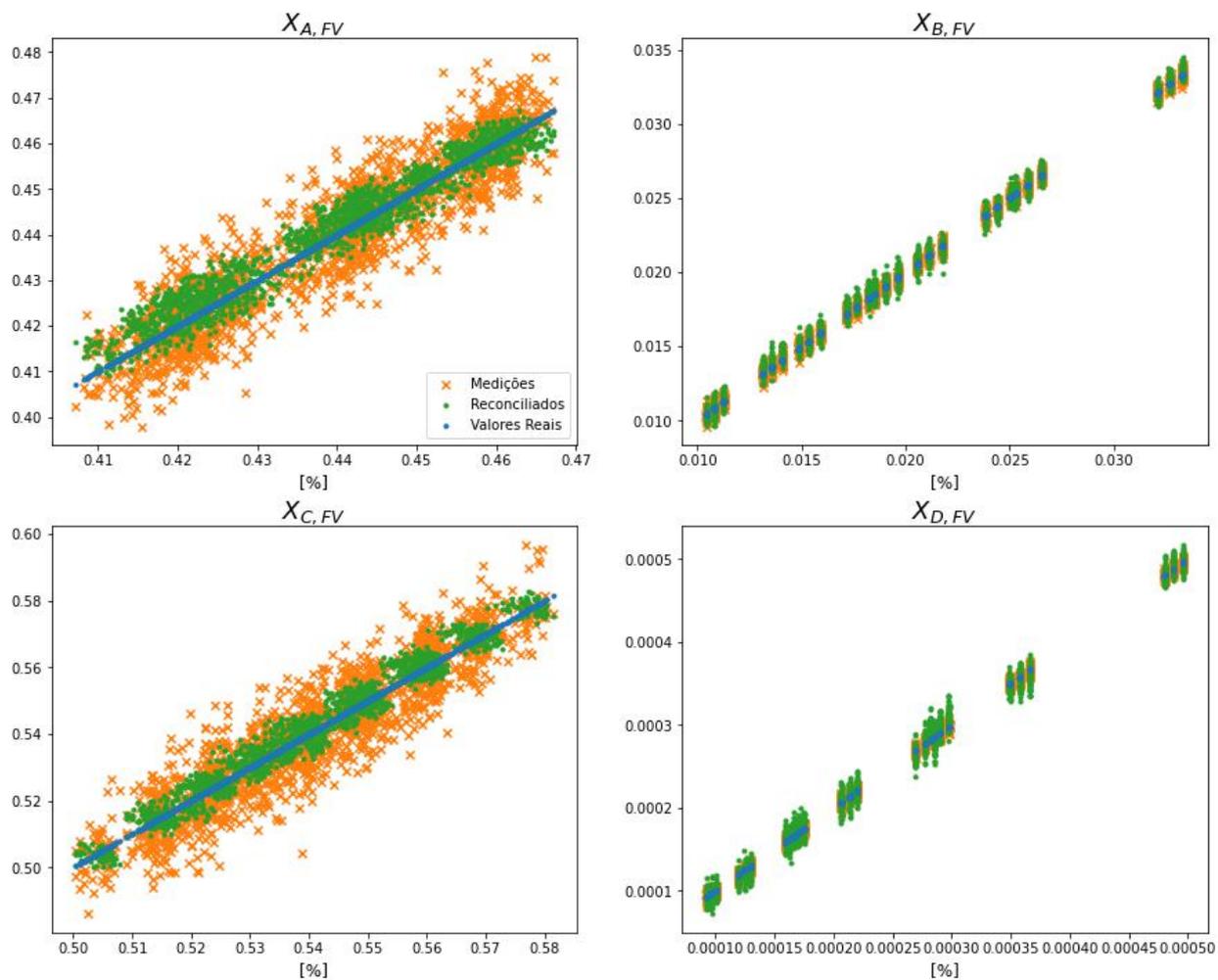
Fonte: produzido pelo autor.

Figura A.23: Resultado da reconciliação para o flash com 4 neurônios na camada de gargalo, $\lambda = 0,25$ e $l_2 = 0$.



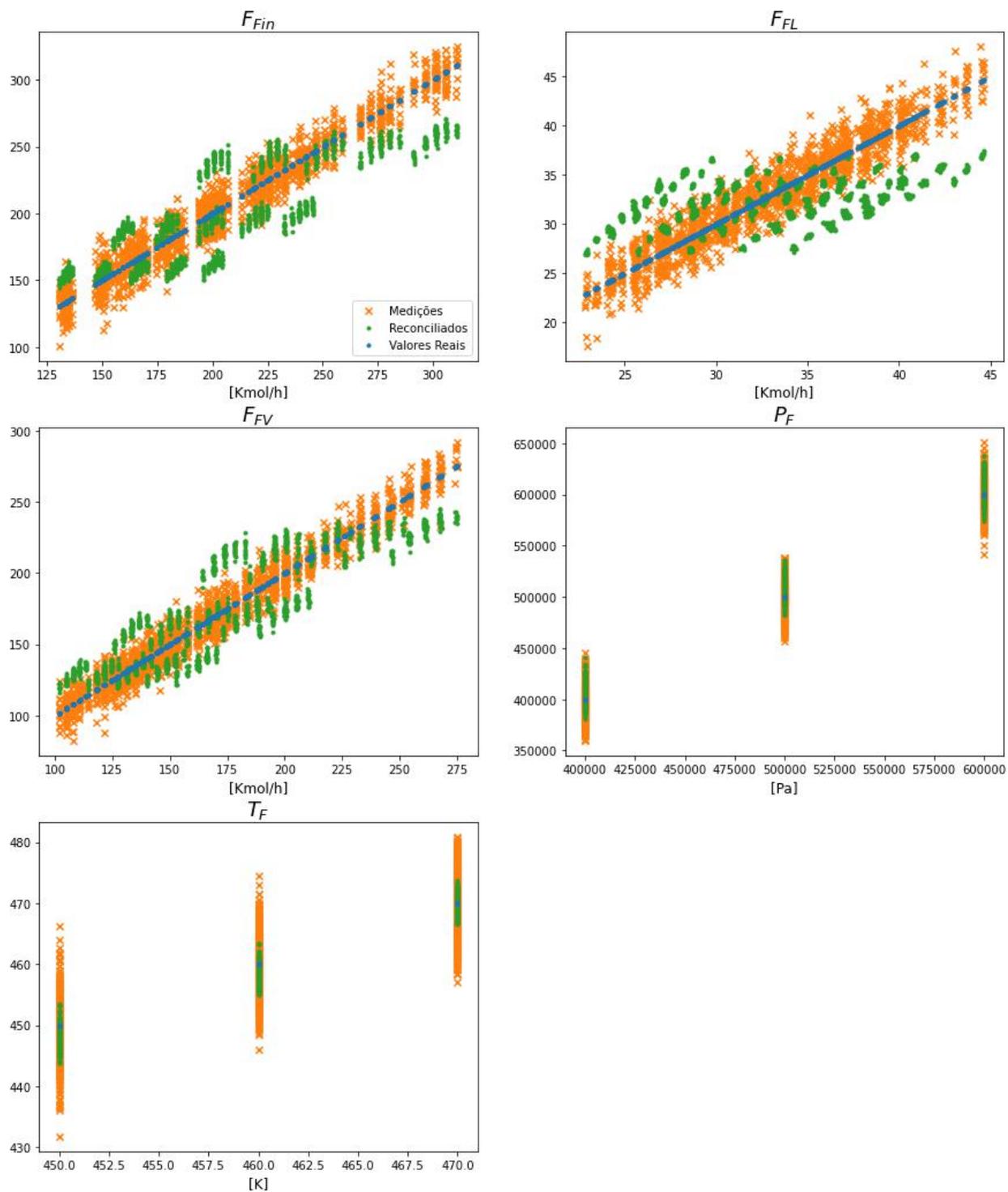
Fonte: produzido pelo autor.

Figura A.24: Resultado da reconciliação para o flash com 4 neurônios na camada de gargalo, $\lambda = 0,25$ e $l_2 = 0$.



Fonte: produzido pelo autor.

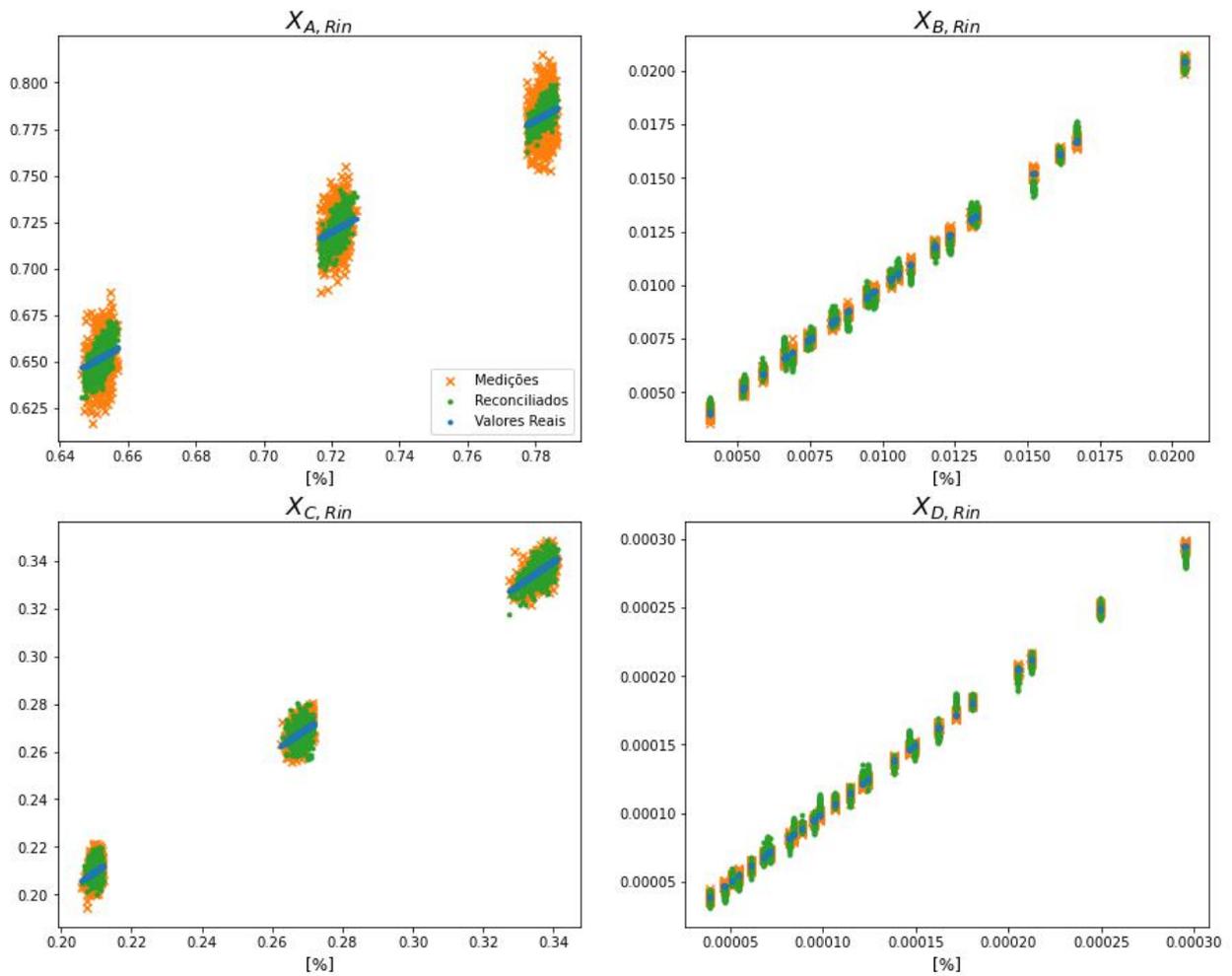
Figura A.25: Resultado da reconciliação para o flash com 4 neurônios na camada de gargalo, $\lambda = 0,25$ e $l_2 = 0$.



Fonte: produzido pelo autor.

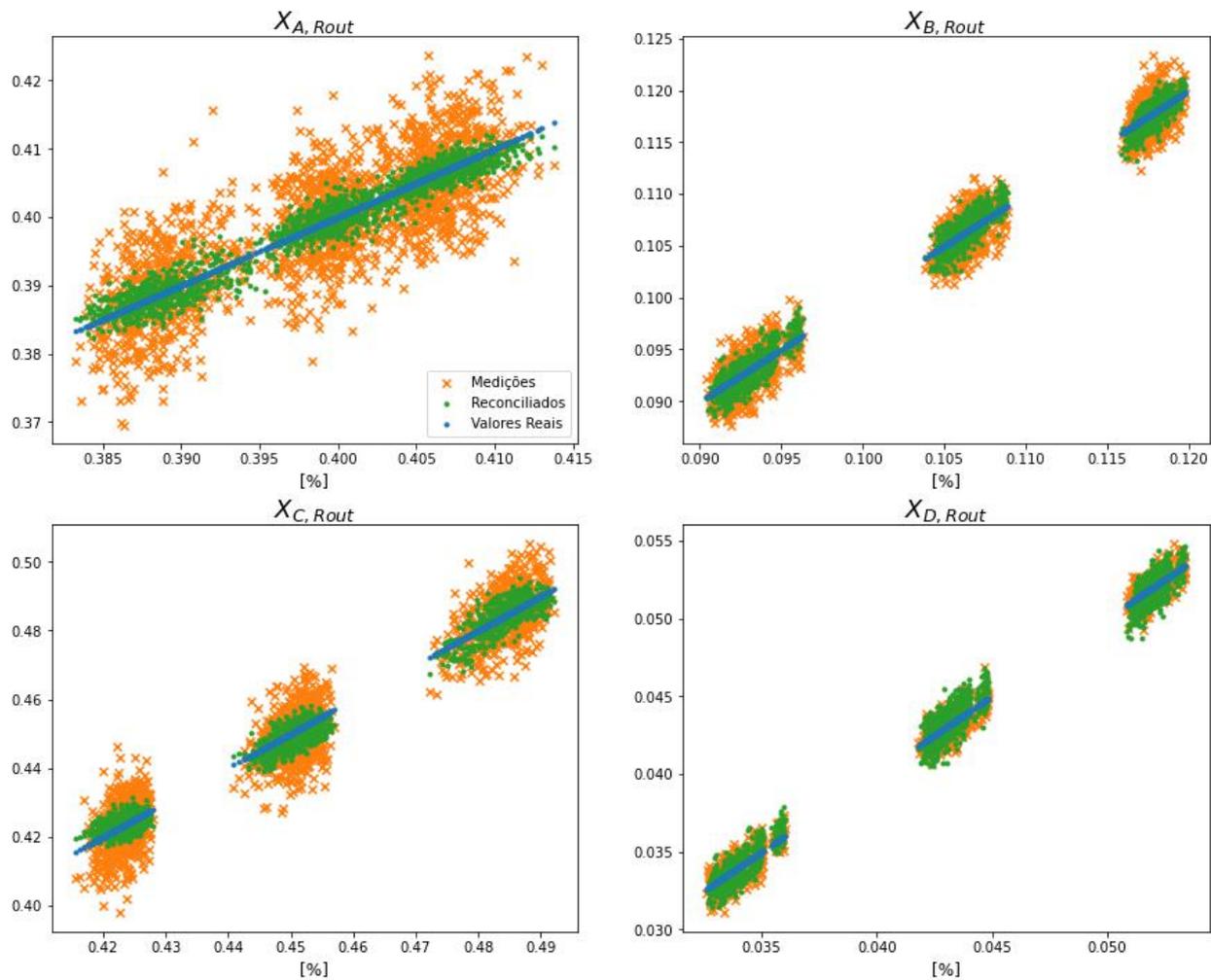
A.7 Reator com $\lambda = 0, 1$ e $l_2 = 0$

Figura A.26: Resultado da reconciliação para o reator com $\lambda = 0, 1$ e $l_2 = 0$



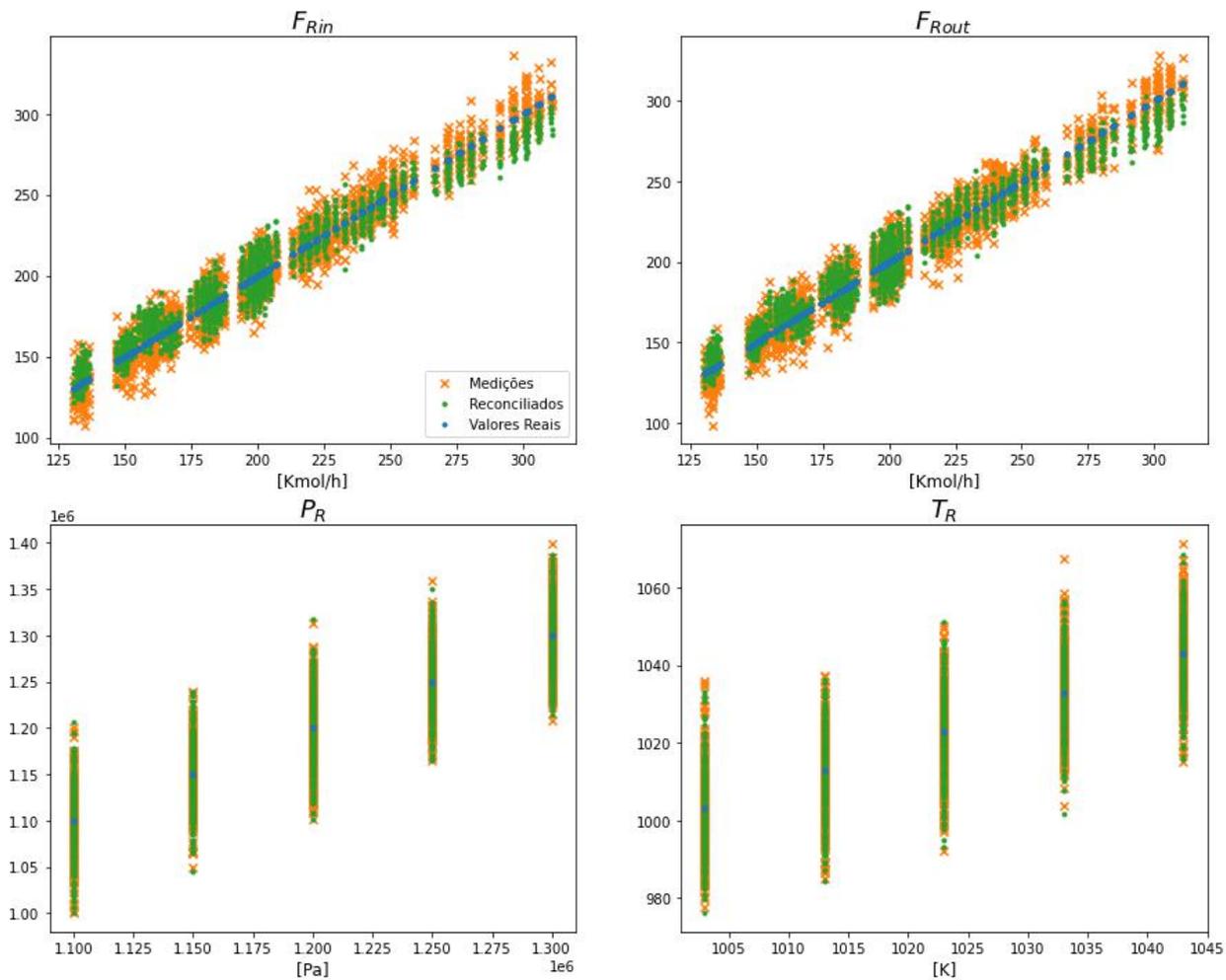
Fonte: produzido pelo autor.

Figura A.27: Resultado da reconciliação para o reator com $\lambda = 0,1$ e $l_2 = 0$



Fonte: produzido pelo autor.

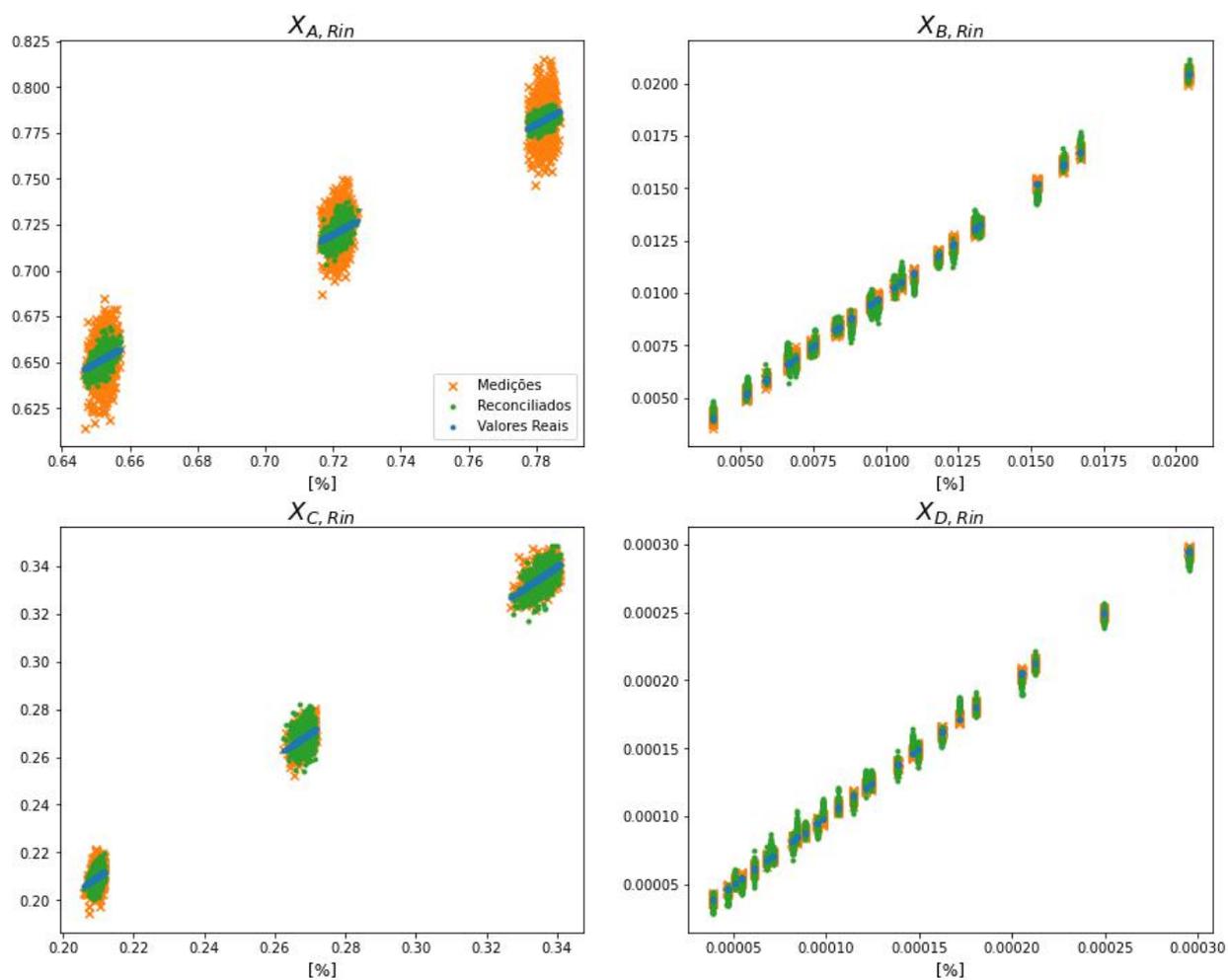
Figura A.28: Resultado da reconciliação para o reator com $\lambda = 0, 1$ e $l_2 = 0$



Fonte: produzido pelo autor.

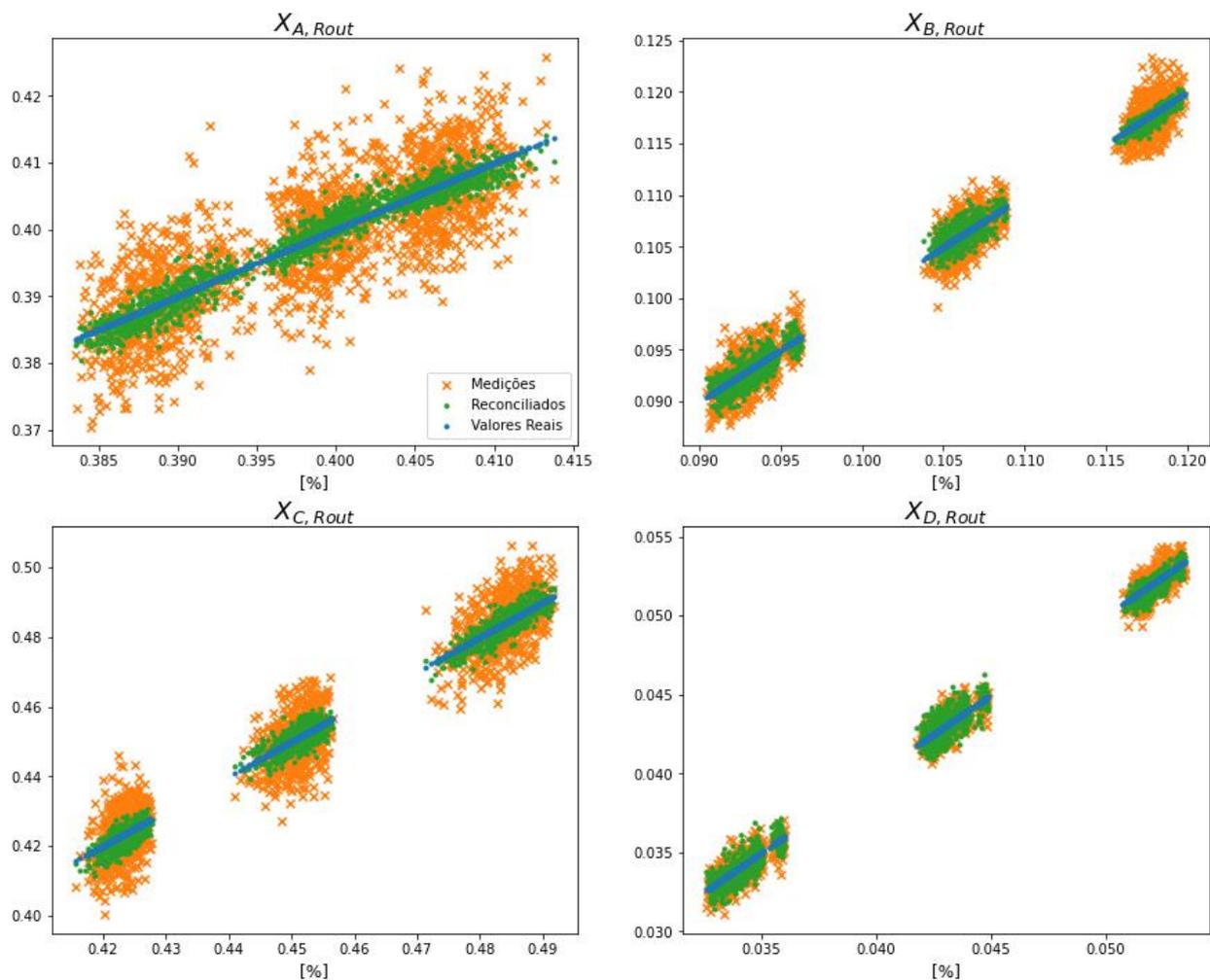
A.8 Reator com $\lambda = 0, 1, 4$ neurônios na camada de gargalo e $l_2 = 0$

Figura A.29: Resultado da reconciliação para o reator com 4 neurônios na camada de gargalo, $\lambda = 0, 1$ e $l_2 = 0$.



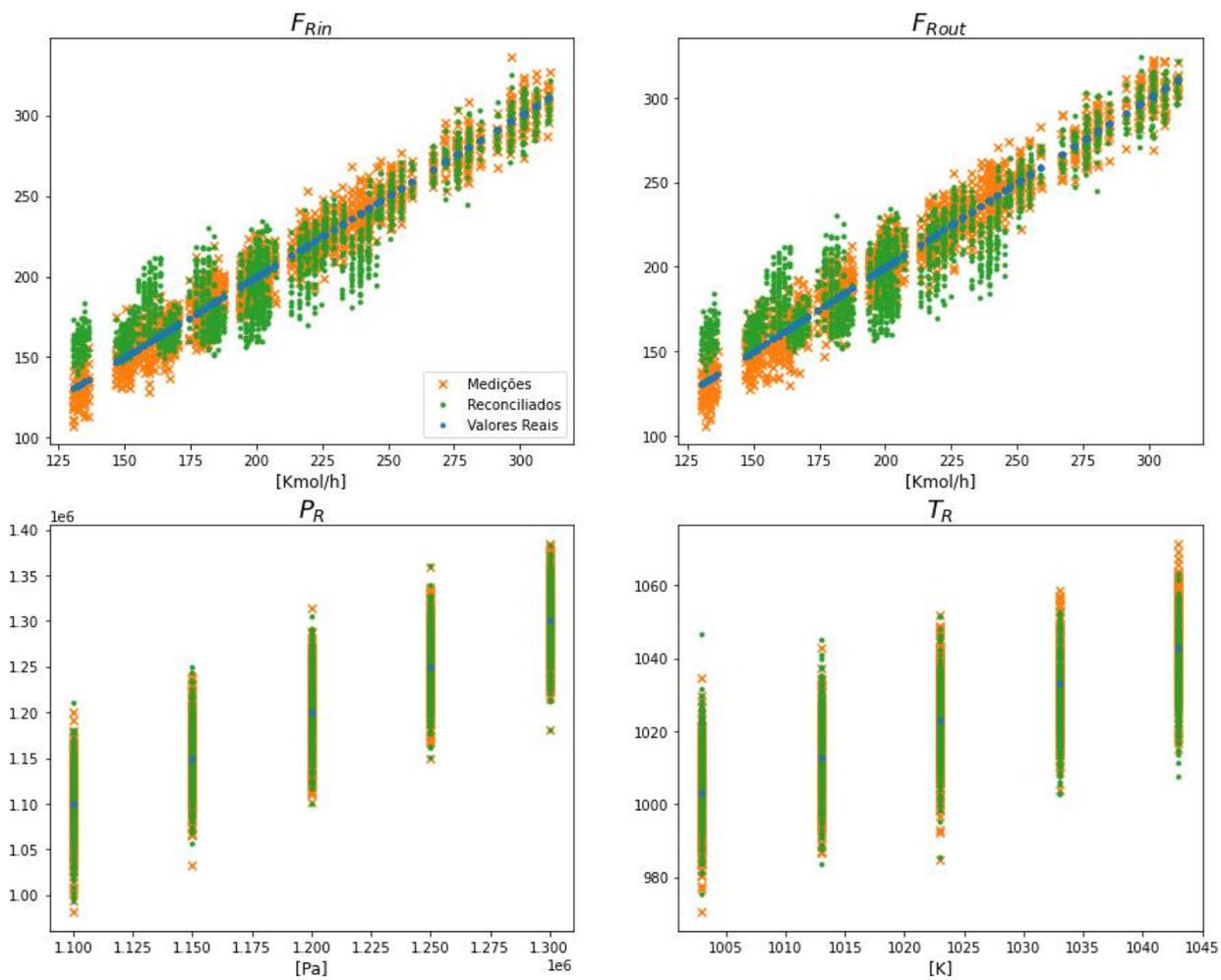
Fonte: produzido pelo autor.

Figura A.30: Resultado da reconciliação para o reator com 4 neurônios na camada de gargalo, $\lambda = 0,1$ e $l_2 = 0$.



Fonte: produzido pelo autor.

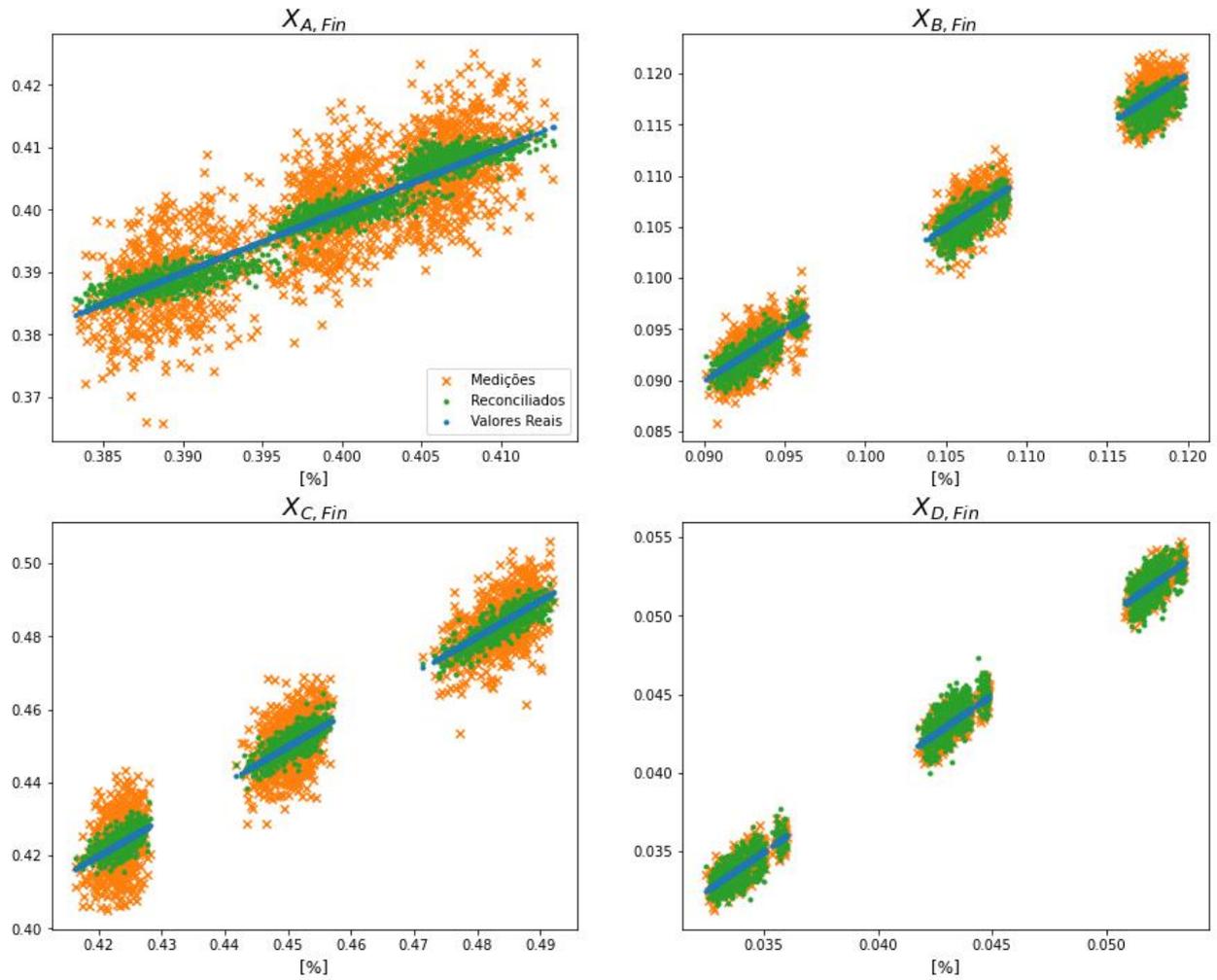
Figura A.31: Resultado da reconciliação para o reator com 4 neurônios na camada de gargalo, $\lambda = 0,1$ e $l_2 = 0$.



Fonte: produzido pelo autor.

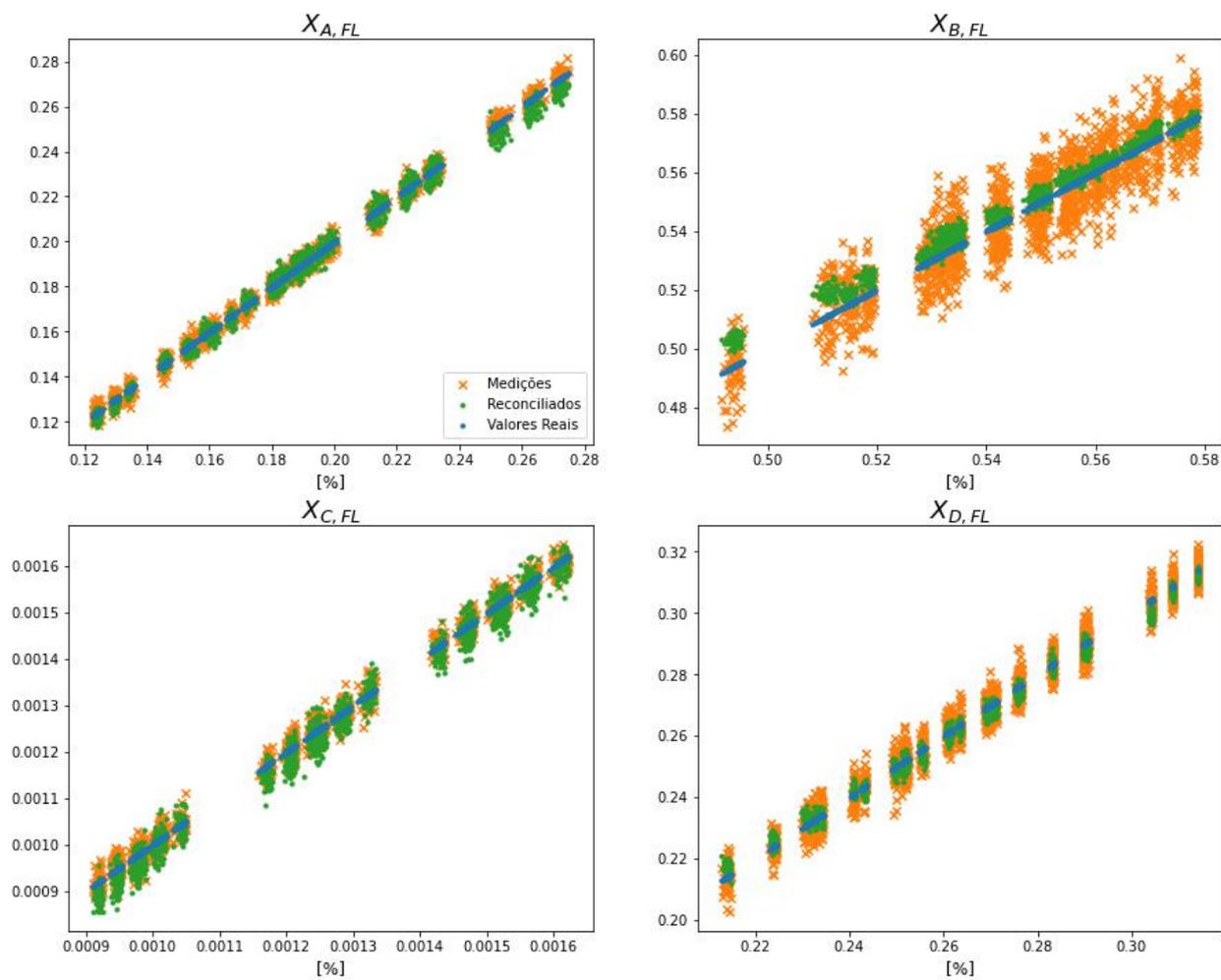
A.9 Flash com $\lambda = 0,25$ e $l_2 = 0,0005$

Figura A.32: Resultado da reconciliação para o flash com $\lambda = 0,25$ e $l_2 = 0,0005$.



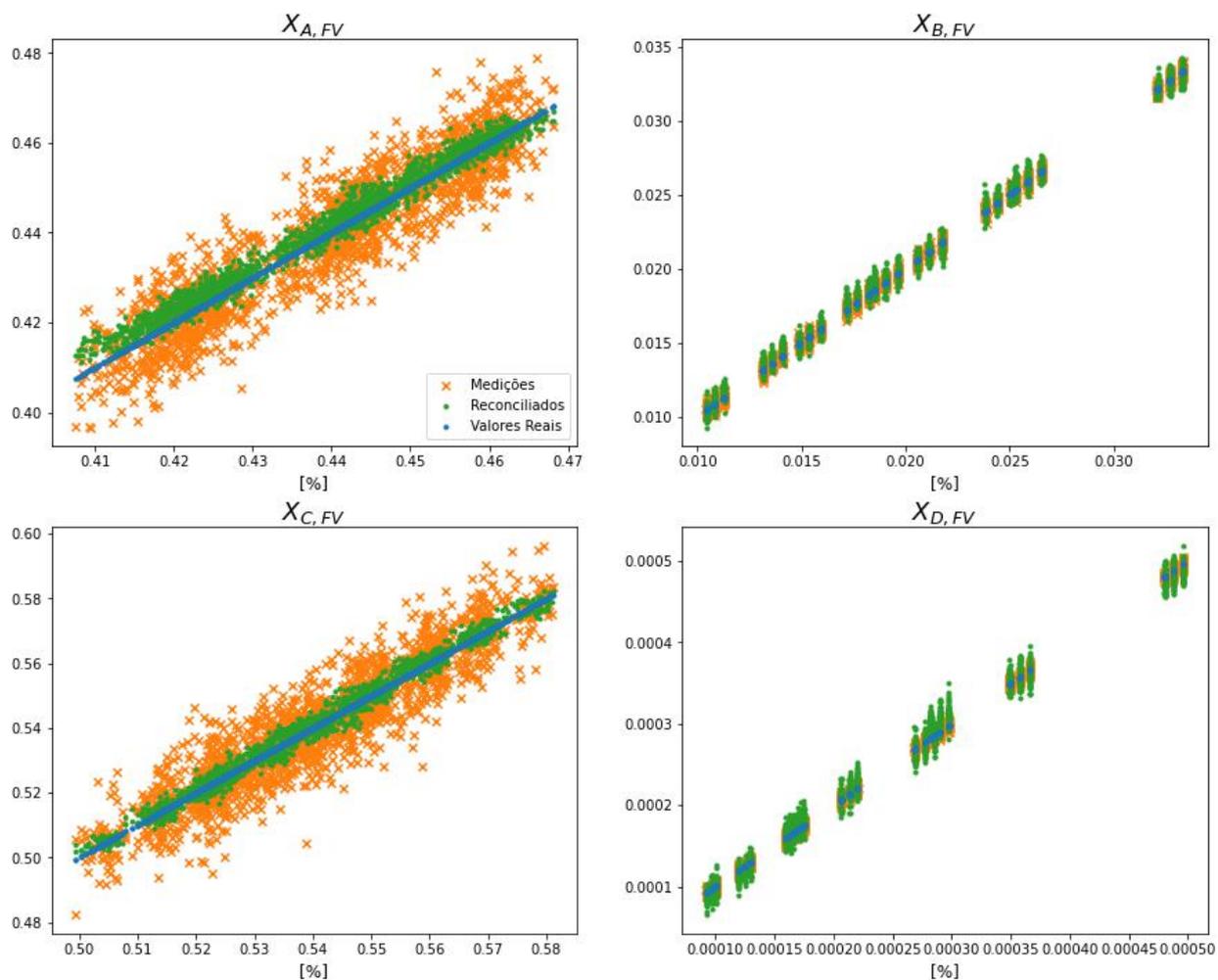
Fonte: produzido pelo autor.

Figura A.33: Resultado da reconciliação para o flash com $\lambda = 0,25$ e $l_2 = 0,0005$.



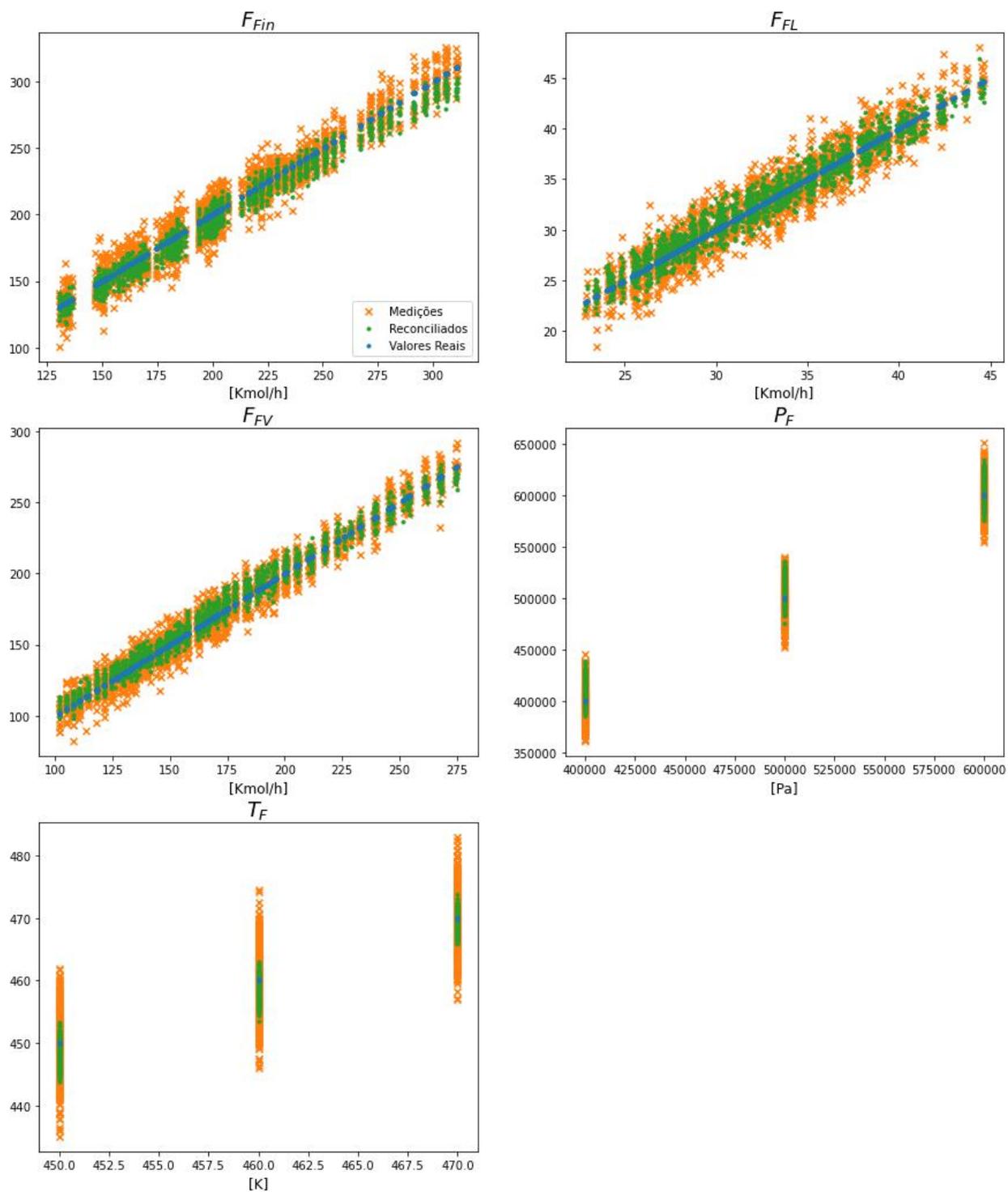
Fonte: produzido pelo autor.

Figura A.34: Resultado da reconciliação para o flash com $\lambda = 0,25$ e $l_2 = 0,0005$.



Fonte: produzido pelo autor.

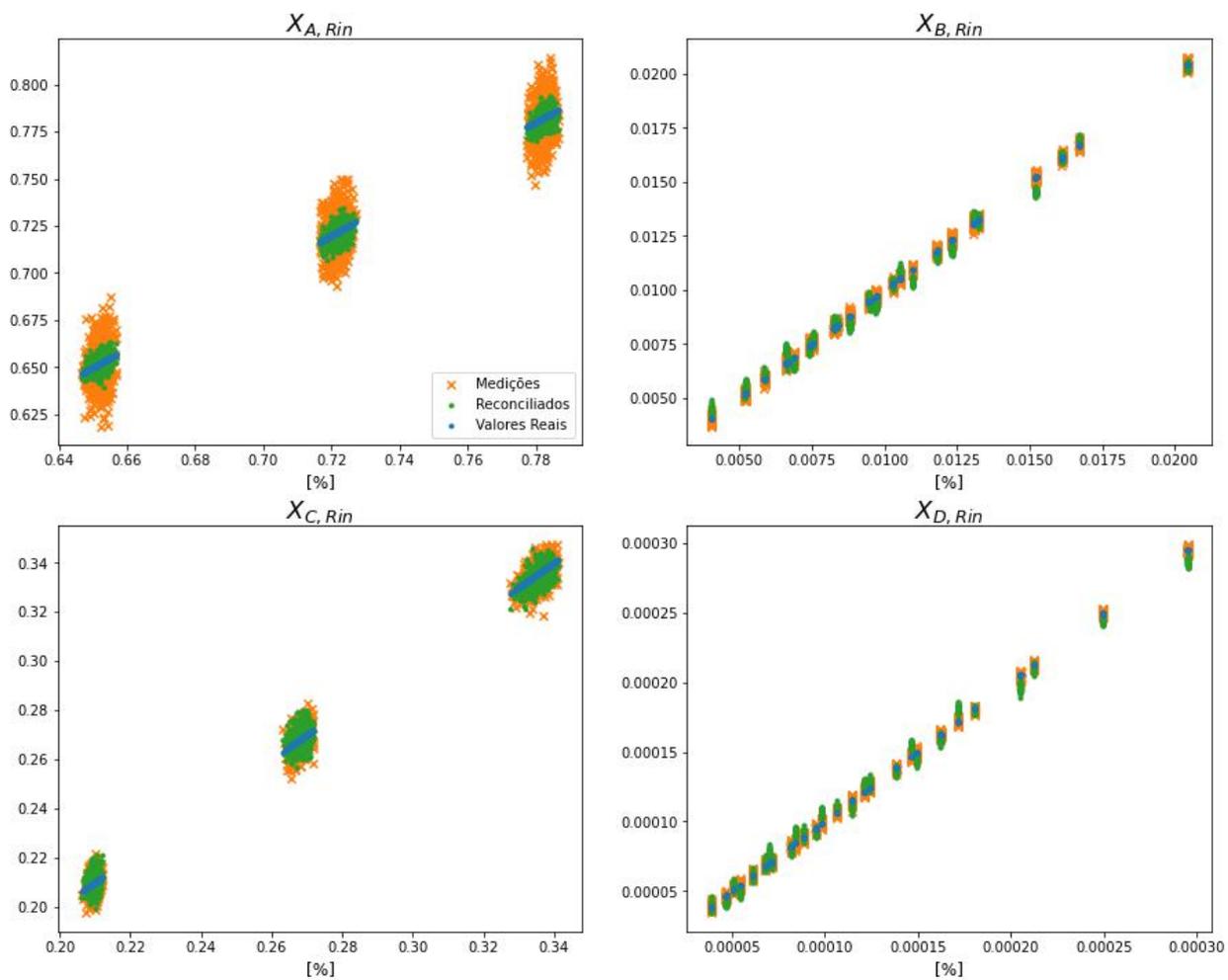
Figura A.35: Resultado da reconciliação para o flash com $\lambda = 0,25$ e $l_2 = 0,0005$.



Fonte: produzido pelo autor.

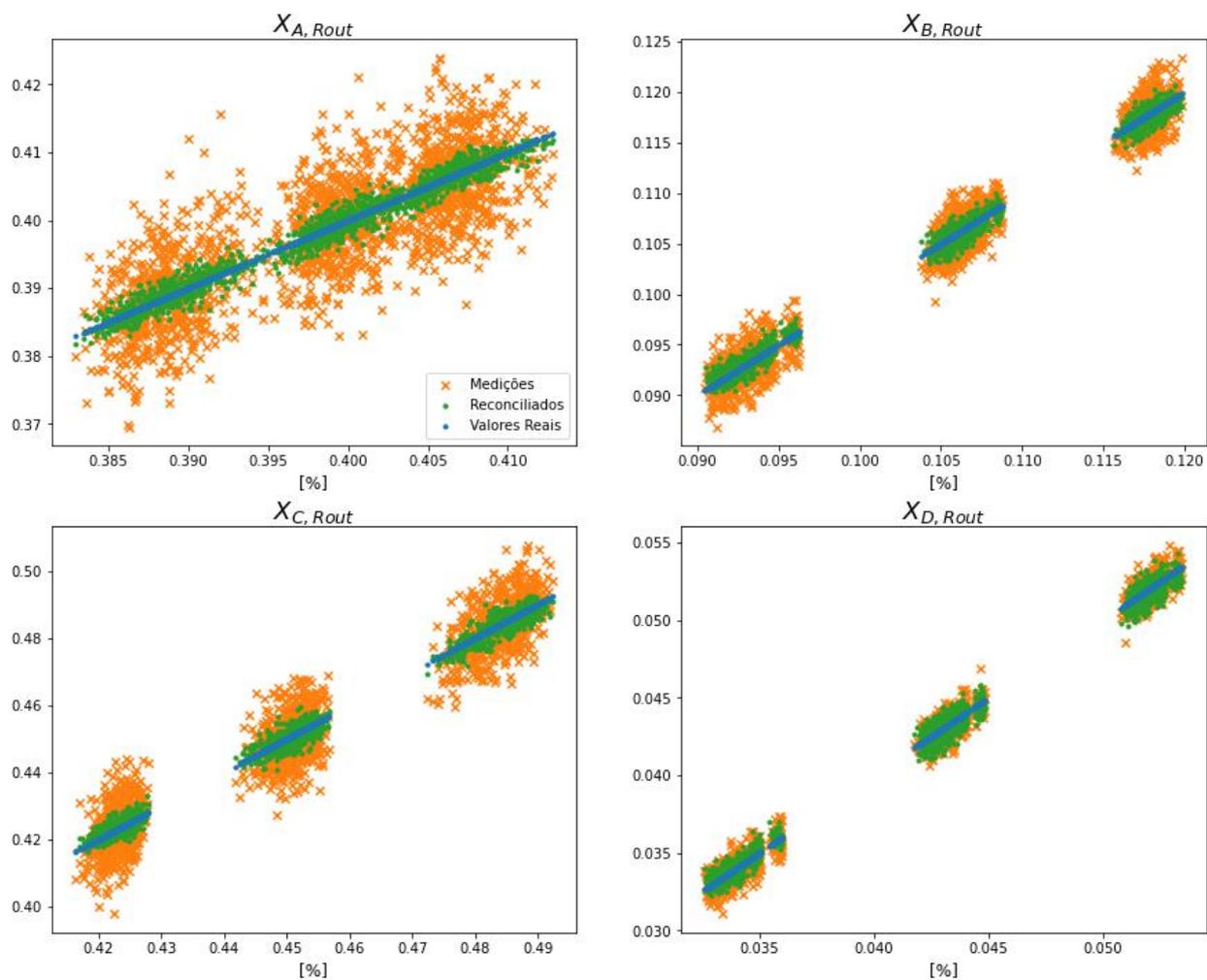
A.10 Reator com $\lambda = 0,1$ e $l_2 = 0,0004166$

Figura A.36: Resultado da reconciliação para o reator com $\lambda = 0,1$ e $l_2 = 0,0004166$.



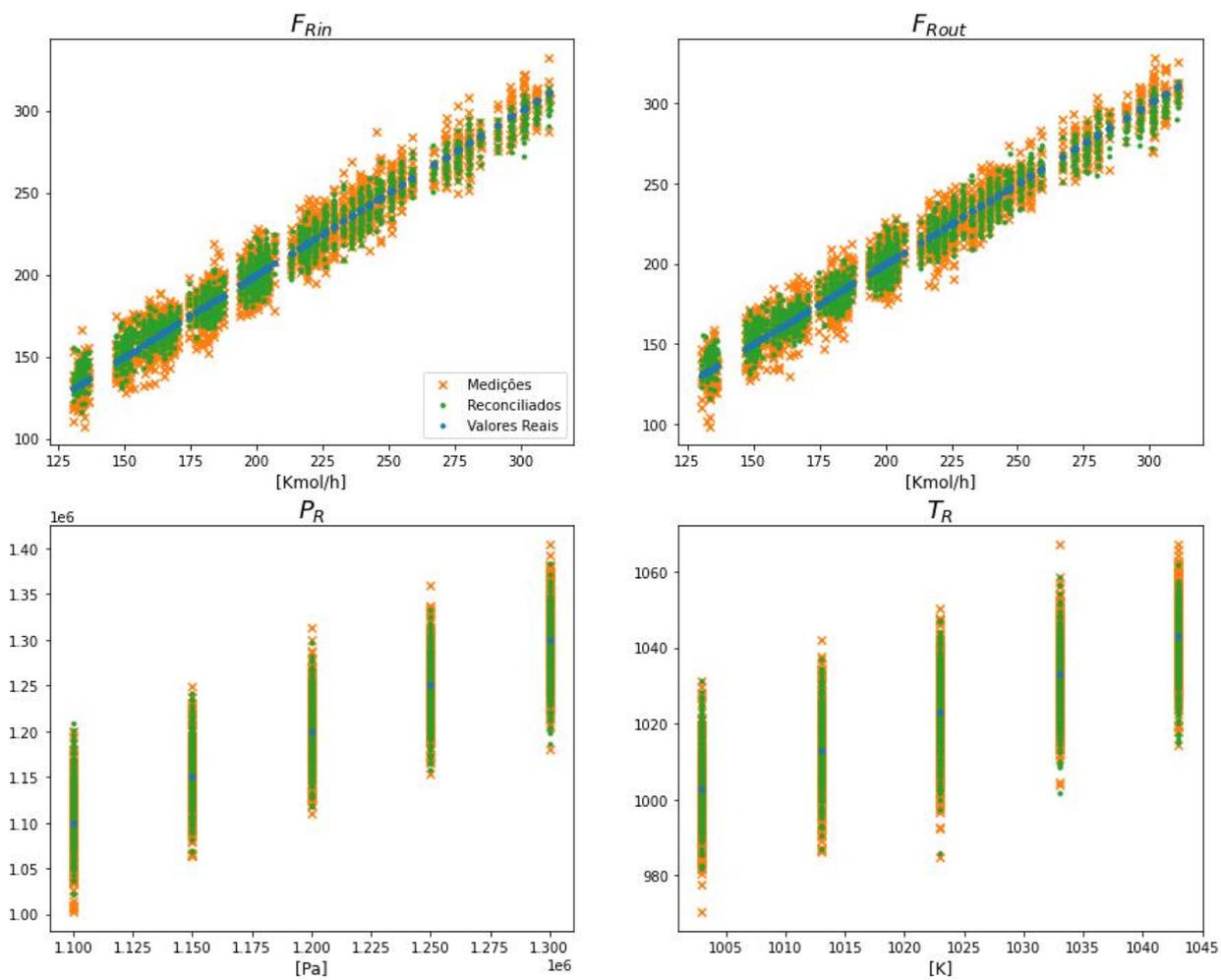
Fonte: produzido pelo autor.

Figura A.37: Resultado da reconciliação para o reator com $\lambda = 0,1$ e $l_2 = 0,0004166$.



Fonte: produzido pelo autor.

Figura A.38: Resultado da reconciliação para o reator com $\lambda = 0,1$ e $l_2 = 0,0004166$.



Fonte: produzido pelo autor.