

UNIVERSIDADE FEDERAL DE SÃO CARLOS
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA QUÍMICA

MARCUS VINICIUS SGOBI

SIMULAÇÃO DA SECAGEM DE BAGAÇO DE LARANJA EM LEITO DE JORRO
POR UM MODELO EM CASCATA USANDO SOFTWARES GRATUITOS

Trabalho de Graduação apresentado ao
Departamento de Engenharia Química da
Universidade Federal de São Carlos.

Orientador: Prof. Dr. Fábio Bentes Freire

SÃO CARLOS

2024

BANCA EXAMINADORA

Trabalho de Graduação apresentado no dia 22 de janeiro de 2024 perante a seguinte banca examinadora:

Orientador: Prof. Dr. Fábio Bentes Freire

Departamento de Engenharia Química - Universidade Federal de São Carlos

Convidado: Profa. Dra. Maria do Carmo Ferreira

Departamento de Engenharia Química - Universidade Federal de São Carlos

Professor da Disciplina: Prof. Dr. José Maria Corrêa Bueno

Departamento de Engenharia Química - Universidade Federal de São Carlos

AGRADECIMENTOS

Tolkien, em 1954, narrou um conhecimento passado do velho Bilbo ao seu sobrinho Frodo que percebo, hoje mais do que nunca, se aplicar não só para os pequenos hobbits, com suas vidas tranquilas em seu país longínquo, mas também a mim, e tenho certeza que a muitos outros que vivenciaram, vivenciam e vivenciarão este período de grande transição e transformação em suas vidas. Segundo Bilbo: “It's a dangerous business, Frodo, going out your door. You step onto the road, and if you don't keep your feet, there's no knowing where you'll be swept off to”.

É muito difícil que a intenção de Tolkien ao escrever esse trecho fosse sua interpretação literal, ainda mais vindo de Bilbo, um amante nato de poesias e de incutir sentidos e significados ocultos nas entrelinhas de suas falas; mais que isso, a estrada pode ser entendida como os caminhos que todos percorrem durante a vida; suas bifurcações e encruzilhadas como as diversas escolhas que nos são apresentadas; e o sair da porta como ter a liberdade de seguir os próprios caminhos, sem ser mais protegido dos perigos e das maravilhas que a estrada pode proporcionar. E é claro, não se pode esquecer dos muitos encontros que podem ocorrer, uns fortuitos, outros nem tanto.

O momento do meu ingresso na UFSCar foi o equivalente a dar o primeiro passo fora da porta de casa para seguir a grande estrada - que sempre continuará seus caminhos, mesmo que eu não mais os percorra - tendo diante de mim um mundo repleto de escolhas, que me mudaram, mudam e mudarão. Agora, ao final da graduação, me sinto como numa grande encruzilhada, com diversas opções bastante proveitosas de seguir em frente, mas ainda indeciso do caminho a trilhar.

Enquanto descanso numa sombra e penso sobre meus caminhos, gostaria de agradecer à possibilidade que tive de encontrar grandes figuras sem as quais certamente teria me perdido; quiçá voltado à porta de casa por engano: aos meus pais, pela presença constante e auxílio incondicional; a todos professores que já tive, não só no sentido acadêmico, pelo conhecimento a mim transmitido; ao meu orientador, que mesmo com inúmeras turbulências na sua vida, foi sempre muito compreensivo e esteve sempre disponível a qualquer momento; e, como não poderia deixar de ser, a todos meus amigos, que, mesmo com a distância, sempre são capazes de fazer com que meus pés não fiquem cansados da caminhada.

“For better or worse, whether he wishes it or not, no man on this earth rides alone. We are all of us bound together, heart to heart and soul to soul -- all of us bit players in stories bigger than we know - stories we have no hand in writing.”

Gudsforladt

RESUMO

Devido ao consumo e produção cada vez maior dos derivados da laranja, uma grande quantidade de biomassa, denominada genericamente como bagaço, é gerada. Essa biomassa tem potencial para diversas aplicações, tais como uso para ração animal ou combustão, mas todas elas necessitam de uma etapa de secagem. Dentre os vários métodos que podem ser usados para realizar essa secagem, um que vem acumulando considerável volume de estudos é o do leito de jorro, que, por mais que possua suas desvantagens, sempre se mostrou bastante viável na pequena escala e mesmo na média escala. Contudo, antes que qualquer processo químico seja consolidado, é requerida uma etapa de simulação, que visa descrever matematicamente a realidade física. Ter um modelo plenamente desenvolvido e com validade para uma ampla gama de condições torna possível avançar em certas etapas da consolidação do processo de secagem, tais como o desenvolvimento de um sistema de controle eficiente e o aumento de escala. O presente Trabalho de Graduação teve como seu objetivo desenvolver e simular um modelo adaptativo e de parâmetros concentrados para secagem do bagaço de laranja em leito de jorro usando de um software livre, e posteriormente realizar a validação de tal modelo, comparando os resultados simulado com dados experimentais disponíveis de antemão. Até onde foi possível chegar, com o curto tempo disponível para o desenvolvimento de um TG bastante desafiador, infelizmente, não foi concluída de forma satisfatória a validação experimental, pois foi identificada uma falha bastante pontual do código que previnha a conclusão dessa etapa.

LISTA DE FIGURAS

Figura 2.1 – Fluxograma simplificado de uma unidade de processamento de laranja típica. .	16
Figura 2.2 - Curva característica para a taxa de secagem em função do tempo.	22
Figura 2.3 - Representação de uma das possíveis modificações estruturais no leito de jorro, a agitação mecânica.	25
Figura 2.4 – Representação esquemática de um leito de jorro convencional.	26
Figura 2.5 – Reflexão de pontos no método de busca simplex em duas dimensões.	35
Figura 2.6 – Representação da modelagem em cascata.	36
Figura 3.1 – Fluxograma do aparato experimental utilizado.	40
Figura 3.2 – Gráficos das variáveis de interesse para o algoritmo, a partir dos dados experimentais sem tratamento, com destaque para o ruído que apresentam.	43
Figura 3.3 – Gráficos das variáveis de interesse para o algoritmo, utilizando um filtro FIR para tratar o ruído nelas presente.	43
Figura 4.1 – Gráfico da correspondência das umidades absolutas do gás na saída do leito de jorro medida e simulada.	50
Figura 4.2 – Gráfico da correspondência das temperaturas do gás na saída do leito de jorro medida e simulada.	51
Figura 4.3 – Gráfico da evolução da constante cinética de secagem em função do tempo.	52
Figura 4.4 – Gráfico da evolução do coeficiente global de transferência de calor em função do tempo.	52

LISTA DE TABELAS

Tabela 3.1 – Medidas iniciais e finais do banco de dados obtido no artigo de (FREIRE et al., 2017).....	42
Tabela 4.1 – Tempos de execução dependendo da tolerância para finalização do algoritmo de Nelder-Mead e número de parâmetros ajustados.	49

NOMENCLATURA

C _p	Calor específico	$L^2 t^{-2} T^{-1}$
h _A	Coefficiente global de transferência de calor	$M t^{-3} T^{-1}$
k	Constante cinética de secagem	t^{-1}
m	Massa	M
MW	Peso molecular	-
P	Pressão	$M L^{-3} t^{-2}$
Q̇	Vazão volumétrica de ar	$L^3 t^{-1}$
Q _{latent}	Calor latente de vaporização	$L^2 t^{-2}$
RH	Umidade relativa do ar	-
T	Temperatura	T
t	Tempo	t
W	Umidade em base úmida	-
x	Variável independente	-
X	Umidade em base seca	-
y	Variável dependente	-
Y	Umidade absoluta do ar	$M.L^{-3}$

Letras gregas

ρ	Densidade	$M.L^{-3}$
---	-----------	------------

Subscritos e sobrescritos

exp	Experimental
g	Gás
i	Entrada (“inlet”)
n	Enésimo valor de uma função
o	Saída (“outlet”)
s	Sólidos
v	Vapor
w	Água

Abreviações

CSB	Conventional Spouted Bed (Leito de Jorro Convencional)
FAO	Food and Agriculture Organization of the United Nations
GPU	Graphics Processing Unit
IDE	Integrated Development Enviroment
PID	Proporcional Integrativo Derivativo
POO	Programação Orientada a Objetos
PVI	Problema de Valor Inicial
QDMC	Quadratic Programming Solution of Dynamic Matrix Control
FIR	Finite Impulse Response
RAM	Random Access Memory
SSR	Solid State Relay
VS Code	Visual Studio Code

SUMÁRIO

BANCA EXAMINADORA	ii
AGRADECIMENTOS	iii
RESUMO	v
LISTA DE FIGURAS	vi
LISTA DE TABELAS	vii
NOMENCLATURA	viii
SUMÁRIO	x
1 - INTRODUÇÃO	12
2 – REVISÃO BIBLIOGRÁFICA	15
2.1. BAGAÇO DE LARANJA	15
2.2. FUNDAMENTOS DE SECAGEM	18
2.3. LEITOS DE JORRO	24
2.4. MODELAGEM MATEMÁTICA	27
2.4.1. CLASSIFICAÇÕES DE MODELOS MATEMÁTICOS	28
2.4.2. MODELO MATEMÁTICO DA SECAGEM DE BAGAÇO DE LARANJA EM LEITO DE JORRO	29
2.5. MÉTODOS NUMÉRICOS	32
2.5.1. SOLUÇÃO NUMÉRICA DE EQUAÇÕES DIFERENCIAIS	32
2.5.2. MÉTODOS DE RUNGE-KUTTA.....	33
2.5.3. OTIMIZAÇÃO	34
2.5.4. MÉTODO DE NELDER-MEAD.....	35
2.5.5. MODELAGEM EM CASCATA	36
2.6. LINGUAGENS DE PROGRAMAÇÃO	37
2.6.1. PYTHON.....	37
2.6.2. LINGUAGENS DE PROGRAMAÇÃO COMPILADAS E INTERPRETADAS	38
2.6.3. PROGRAMAÇÃO ORIENTADA AO OBJETO.....	38
3 – MATERIAIS E MÉTODOS	40
3.1. BANCO DE DADOS	40
3.1.1. OBTENÇÃO DOS DADOS	40

3.1.2. DISPOSIÇÃO DOS DADOS.....	41
3.1.3 TRATAMENTO DO BANCO DE DADOS.....	42
3.2. MATERIAIS.....	45
3.2.1 HARDWARE.....	45
3.2.2 SOFTWARES	45
3.3. DESENVOLVIMENTO DO ALGORITMO.....	47
4 – RESULTADOS E DISCUSSÃO	48
4.1. TEMPO DE EXECUÇÃO.....	48
4.2. VALIDAÇÃO DO MODELO	50
5 – CONCLUSÕES E SUGESTÕES PARA TRABALHOS FUTUROS.....	54
REFERÊNCIAS BIBLIOGRÁFICAS	56
APÊNDICE A – CÓDIGO NA ÍNTEGRA	60
APÊNDICE A.1 MÓDULO DATAPROCESSING	60
APÊNDICE A.2 MÓDULO MODEL	64
APÊNDICEA.3 MÓDULO SIMULATION.....	70
APÊNDICE A.4 MÓDULO CASCADEMODEL.....	72

1 - INTRODUÇÃO

O Brasil figura como o maior produtor mundial de suco de frutas cítricas (FAO, 2021), com destaque ao suco de laranja, que é uma das bebidas mais consumidas no mundo. Boa parte da produção nacional é exportada na forma de suco concentrado, extraído mecanicamente através de prensagem. A grande vantagem em comercializar o suco concentrado está na redução dos custos de armazenamento e, principalmente, de transporte. A produção de suco concentrado de laranja envolve a remoção da água em excesso através da secagem térmica com ar aquecido, seguida da recuperação de voláteis que dão o aroma característico ao produto. Desde a entrada da matéria-prima *in natura*, até a saída do produto pronto para ser comercializado, geram-se resíduos e subprodutos em grandes quantidades, com destaque ao bagaço de laranja pós-prensagem (TETRAPAK, 2017).

Atualmente, dentro do conceito de economia circular em que há um esforço do setor produtivo para aumentar a sustentabilidade e reduzir a pegada de carbono, as indústrias procuram formas de valorizar resíduos em vez de simplesmente tratá-los. Só para esclarecimento, uma pegada de carbono é o total de emissões de gases de efeito estufa causadas por um indivíduo, evento, organização, serviço, local ou produto, expresso em dióxido de carbono equivalente. Ao mesmo tempo em que isso acontece, a insegurança alimentar aumenta a taxas elevadas, com desdobramentos tanto no desenvolvimento quanto na inovação dentro da produção de alimentos. No caso específico do bagaço de laranja, gerado em larga escala, uma estratégia comum de valorização transforma o subproduto em alimentos secundários de alto valor para animais. Outra alternativa é o aproveitamento energético do bagaço de laranja através da combustão, pirólise ou gaseificação. Em qualquer um dos casos, deve-se secar o resíduo sólido antes de usá-lo.

Existem diversas técnicas para secar grandes quantidades de sólidos, desde a secagem convencional direta ao sol em áreas abertas até processos mais controláveis em tambores rotativos, esteiras transportadoras e secadores de leito fluidizado. O leito de jorro pode ser uma alternativa aos leitos fluidizados, especialmente para o manuseio de partículas grosseiras, segundo o jargão da área, e irregulares. Em sua configuração convencional, o leito de jorro permite um contacto eficaz entre as diferentes fases do processo através de um movimento circular de sólidos, possuindo três zonas internas distintas, nomeadamente o canal de jorro, a fonte e o ânulo. O bom contato entre as partículas e a corrente de ar ascendente juntamente com

a circulação vigorosa de sólidos levam a altas taxas de transferência de calor e massa (MATHUS; EPSTEIN, 1974).

Por ser versátil, o leito de jorro pode ser encontrado em diversas aplicações, como sistema de alimentação de sólidos (MASSARO SOUSA; FERREIRA, 2020), combustão e pirólise (AMUTIO et al., 2012; ARTETXE et al., 2015; LOPEZ et al., 2015; LOPEZ et al. ., 2010; OLAZAR et al., 2006), secagem de polpa (BRITO et al., 2018; SOUSA et al., 2019) e secagem de biomassa e alimentos (FREIRE et al., 2017; OLAZAR et al., 2012; SAHIN et al., 2013; SPREUTELS et al., 2014). Uma das principais vantagens do leito de jorro sobre o leito fluidizado é o alto grau de mistura entre a fase gasosa e a fase sólida dentro do equipamento devido ao notável movimento cíclico e uniforme das partículas. Embora seja de fato um contator sólido-fluido versátil, o leito de jorro convencional (CSB) apresenta alto consumo de energia e problemas dimensionais de altura e diâmetro, o que limita o aumento de escala (JAY et al., 1994; MATHUR; EPSTEIN, 1974; YUE et al., 2021). Existem formas de melhorar o desempenho geral do leito de jorro, como operar de forma mais estável com sólidos apenas em base cônica (LOPEZ et al., 2017), além do uso de partes internas adicionais como agitação mecânica, tubos draft e confinadores de fontes que reduzem a velocidade de mínimo jorro e a queda de pressão (ALTZIBAR et al., 2013, 2017).

PINTO et. al (2023) mostraram os avanços fluidodinâmicos de um leito de jorro agitado mecanicamente com lâminas inclinadas e base cônica. O sistema de agitação mecânica permitiu uma diminuição significativa na vazão de ar aquecido, fazendo com que o secador trabalhasse mais próximo da saturação, aproveitando melhor a energia térmica. Estima-se que a secagem demanda algo em torno de 15% da energia total utilizada pelas indústrias. Além disso, sua eficiência é bastante baixa, variando de 25 a 50%, e até 10% em alguns casos. Consequentemente, os custos da secagem industrial estão intimamente relacionados com a gestão de energia. Tal como acontece com outras operações energeticamente intensivas, uma gestão energética de boa qualidade pode ser alcançada através da utilização de estratégias adequadas de monitoramento e de controle, uma vez que o equipamento funciona mais próximo das condições ideais. CORREA et. al. (2003) chegaram a economizar 50% de energia elétrica usando uma estratégia de controle QDMC durante a secagem de pasta em leito de jorro semi-industrial, fazendo o secador operar de forma estável com a mínima vazão possível do ar de secagem. Embora exista motivação suficiente para desenvolvimentos na área de controle de secagem, alguns gargalos ainda impedem que a automação seja amplamente encontrada em

aplicações industriais. Alguns desses gargalos são: a) a complexidade das interações entre as variáveis do processo, b) as dificuldades para obter modelos físico-matemáticos razoáveis, e c) a falta de sensores on-line disponíveis das principais variáveis de controle, como, por exemplo, o teor de umidade dos sólidos.

Nesse sentido, FREIRE et. al. (2017) desenvolveram um modelo dinâmico de secagem de resíduos sólidos de suco de laranja em leito de jorro, adequado para controle e automação. Para isso, foram feitos balanços gerais de calor e de massa das fases sólida e gasosa e as equações diferenciais de parâmetros concentrados resultantes foram usadas separadamente em uma estrutura de cálculo adaptativa para atualizar continuamente a transferência de massa variável no tempo e os coeficientes globais de transferência de calor, além da taxa de perdas de calor para o ambiente. Uma abordagem de modelagem semelhante foi anteriormente usada por SANTOS et. al. (2001) e FREIRE et. al. (2004) na terpolimerização de estireno, acrilato de butila e metacrilato de metila, levando a resultados encorajadores. Este trabalho fez a adequação de um script originalmente desenvolvido em MATLAB para a secagem de bagaço de laranja em leito de jorro convencional. O código resultante foi escrito em Python, uma linguagem de programação gratuita que permite integrar sistemas de forma mais simples e eficaz. A estrutura de simulação em cascata foi usada para desacoplar o balanço de massa do balanço de energia, permitindo que as equações diferenciais fossem resolvidas independentemente e mais rápido. A partir de medidas de temperatura e de umidade do ar na exaustão do secador, foram atualizados os parâmetros do modelo que variam no tempo. Ressalta-se, por fim, que esse programa que simula a secagem em leito de jorro desempenha um papel chave em qualquer problema de otimização e controle da operação do leito de jorro equipamento, podendo ser facilmente estendido, inclusive, para outros equipamentos que sejam bons misturadores e atendam às hipóteses assumidas pela modelagem de FREIRE et al. (2017).

Ressalta-se que, indo além de simplesmente traduzir o código de MATLAB para Python, buscou-se deixá-lo tão genérico quanto possível, com a finalidade de, tendo parâmetros equivalentes aos que foram utilizados para seu desenvolvimento, este possa ser aplicado a outros módulos de secagem. Esse feito foi facilitado pela utilização de métodos da programação orientada a objetos (POO), sobretudo a encapsulação dos métodos e abstração de diversas funções necessárias à simulação.

2 – REVISÃO BIBLIOGRÁFICA

2.1. BAGAÇO DE LARANJA

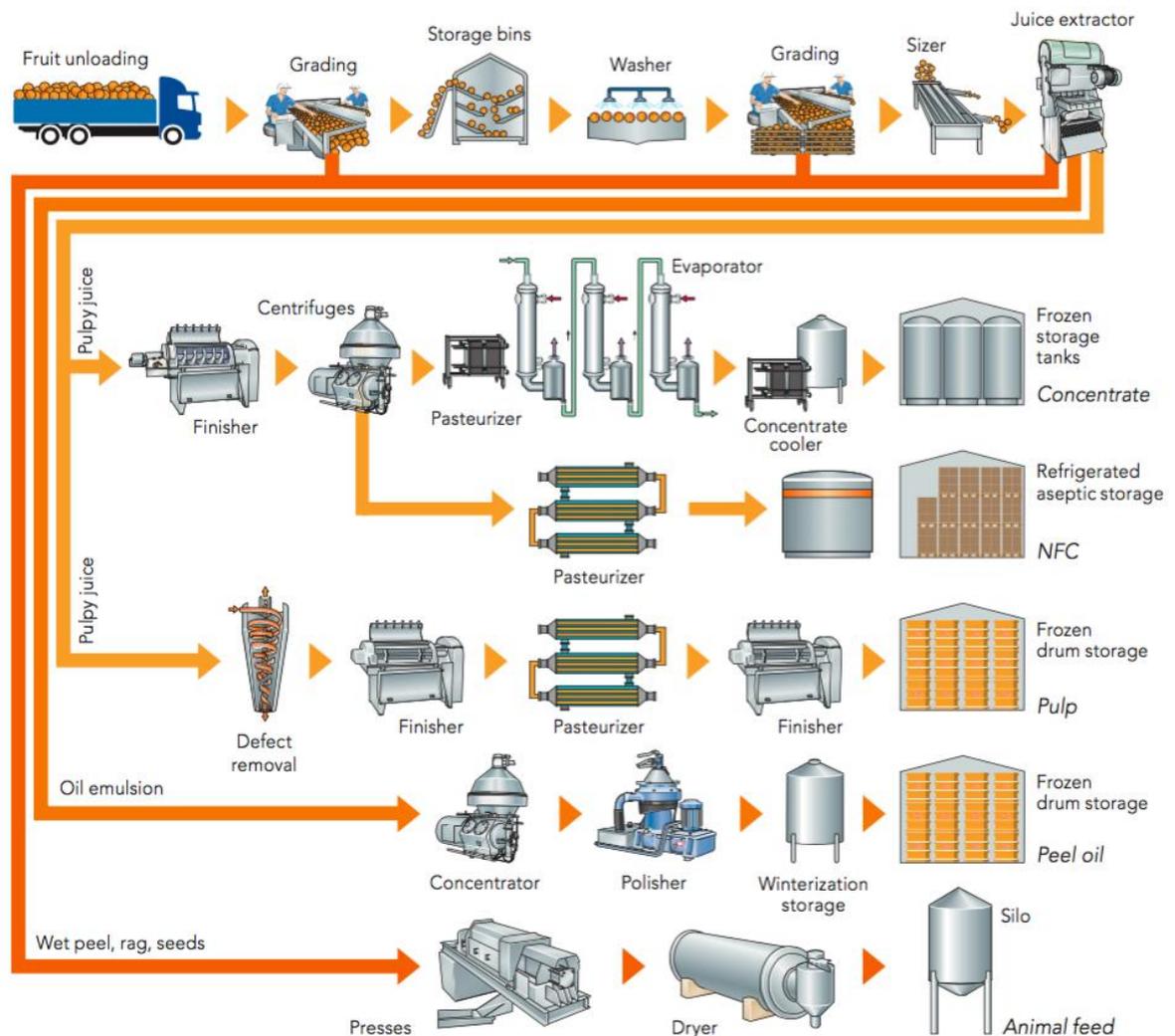
A laranja é um dos produtos agrícolas mais populares do mundo, tanto em termos de plantio quanto em processamento industrial, sendo que seus principais derivados são os sucos, tanto concentrados como integrais, que geram, em sua produção, o bagaço como resíduo. Pelas condições climáticas e de solo do país, o Brasil tem uma posição privilegiada na produção global de laranja; de fato, segundo os dados mais recentes da FAO, em 2019 a produção global total foi de 76.292,6 mil toneladas, sendo o Brasil responsável por 22,4% desse total (FAO, 2021), figurando, por isso, como líder global nesse setor.

Segundo o fluxograma da Figura 2.1 (TETRAPAK, 2017), o processamento industrial da laranja pode ser simplificado pelas seguintes etapas: (1) colheita das frutas na lavoura e transporte delas à unidade industrial, sendo pré-classificadas manualmente, armazenadas e lavadas; (2) classificação mecanizada das frutas por seu tamanho; (3) extração do suco, na qual há compressão mecânica das frutas e separação em diversos possíveis processamentos posteriores. Das etapas 1 e 3, há produção de bagaço, que é constituído, de maneira simplificada, por casca úmida, sementes e fiapos. Tradicionalmente, esse bagaço será submetido a um processo de compressão, secagem e armazenamento para posterior utilização como ração animal. A etapa de extração também gera uma emulsão de óleos essenciais das frutas, que passam por etapas de concentração e tem a eles adicionados compostos voláteis recuperados da evaporação do suco concentrados, sendo purificados por clarificação e centrifugação. Por fim, o suco com polpa, também vindo da etapa 3, é destinado a um processo de clarificação, no qual a polpa é separada e submetida a uma etapa de pasteurização e o clarificado possui duas rotas distintas, podendo ser submetido a um processo de evaporação caso o suco de laranja concentrado seja o produto de interesse ou resfriado diretamente para o suco integral, que possui características mais próximas do suco fresco.

Por mais que a alimentação de animais seja o uso mais tradicional do bagaço, existem diversas outras aplicações sendo estudadas para esse subproduto (ERDOGAN *et al.*, 2015; GUZMAN *et al.*, 2021; HUANG; MA, 2016; MADEIRA *et al.*, 2012; SHAHRAM; TAGHIAN DINANI, 2019; UCUNCU *et al.*, 2013) que necessita de uma disposição final, uma vez que corresponde a aproximadamente 50% da massa da fruta fresca e tem um pH na região de 3, com alto conteúdo de umidade e alta porcentagem de compostos orgânicos (aproximadamente

95% dos sólidos totais); características que o fazem ter um alto potencial contaminante se não for disposto adequadamente (BARBOSA et al., 2020). Para todas as aplicações, entretanto, a etapa de secagem faz-se altamente necessária, e por isso é de interesse cada vez maior da pesquisa na engenharia química identificar gargalos e otimizar processos que possam melhorar a rentabilidade do bagaço e harmonizar os interesses ambientais e econômicos que existem acerca dele.

Figura 2.1 – Fluxograma simplificado de uma unidade de processamento de laranja típica.



Fonte: <https://orangebook.tetrapak.com/chapter/fruit-processing>.

Em termos de composição química, o bagaço é uma biomassa predominantemente lignocelulósica, mas que contém também alguns outros componentes de alto valor agregado cuja extração e comercialização é possível, dos quais se destacam a vitamina C, uma vez que provém de frutas cítricas, e óleos essenciais, também presentes em outras etapas do processo produtivo da laranja. Em termos percentuais, (MORAES; CRESTANI, 2018) estimaram que

esse subproduto contém $2,9\pm 0,2\%$ de cinzas; $38,5\pm 1,5\%$ de celulose; $33,1\pm 2,0\%$ de holocelulose; $24,0\pm 2,0\%$ de hemicelulose e $2,9\pm 0,5\%$ de lignina em base seca, o que ressalta o alto teor de compostos orgânicos frente às cinzas, e é uma possível justificativa para suas aplicações mais tradicionais na combustão (uma vez que possuir menos cinzas implica em uma maior disponibilidade de matéria orgânica a ser oxidada, o que se traduz imediatamente em maior potencial de energia a ser liberada) e produção de ração animal (aplicação na qual é imediato que um maior teor de matéria orgânica implica em mais nutrientes para os animais que conseguem digerir alimentos desse tipo).

2.2. FUNDAMENTOS DE SECAGEM

A secagem talvez seja uma das operações unitárias da engenharia química de associação mais imediata a aplicações práticas, já que é utilizada diariamente em tarefas caseiras como a secagem de roupas ou louças e sua definição é tão simples que mesmo alguém que nunca teve contato com um secador industrial ou com o projeto de um desses equipamentos a entenderia de imediato, não sendo nada mais que o procedimento empregado para reduzir os níveis de umidade de um material até um patamar com o intuito, entre outros, de retardar a deterioração do produto final e preservar suas características por períodos extensos de tempo (BRITO et al., 2019). Para atestar sua simplicidade, a humanidade começou a se utilizar da secagem muito antes de ter um registro escrito bem estabelecido, já que, segundo relatos históricos, os primeiros vestígios do uso da secagem para a preservação de carne remontam a aproximadamente 20.000 a.C. Desde então, as técnicas empregadas evoluíram consideravelmente, alcançando um estágio industrial por volta de 1856, quando foi patenteado o primeiro secador mecânico, construído com tijolos e utilizado para concentrar leite (OTAKE et al., 2020). Além de promover a conservação duradoura dos materiais, a secagem também é atraente devido à significativa redução de sua massa, resultando em uma diminuição dos custos relacionados ao manuseio e armazenamento (HAMAWAND et al., 2015).

Em termos mais comuns a um engenheiro químico, a secagem pode ser definida como a remoção de materiais voláteis, sobretudo a água para a maior parte das aplicações, de sólidos, pastas ou emulsões através da transferência simultânea de calor e de massa. Neste processo, a transferência de calor se dá a partir das vizinhanças, que mais usualmente são ar de secagem, vapor superaquecido ou as paredes do secador, para o material que se deseja secar, fornecendo energia o suficiente para evaporar as moléculas de material volátil que se encontram na superfície do material a ser seco. A transferência de massa acontece principalmente por duas etapas distintas, sendo a primeira a migração das moléculas de volátil do interior das partículas para o seu exterior e a segunda a evaporação de material volátil com a posterior migração deste para a corrente gasosa; entretanto, alguns secadores apresentam mecanismos diferentes para a transferência de massa, tais como os secadores de micro-ondas, que evaporam as moléculas de água já no interior das partículas. Ambos os mecanismos de transferência de massa, a difusão no interior do material (ou mecanismo interno) e a convecção para a corrente gasosa (ou mecanismo externo), influenciam diretamente a taxa de secagem, como diz (MUJUMDAR, 1988).

Quando a secagem é controlada pelos mecanismos externos de transferência de massa, as limitações do processo são causadas pela difusão de vapor na superfície do material. Deste modo, embora as propriedades superficiais do material a ser seco tais como suas características superficiais apresentem alguma influência na resistência ao transporte de massa, as propriedades da fase gasosa (velocidade, temperatura e grau de saturação principalmente) são dominantes. Estes mecanismos são predominantes principalmente em materiais com elevada saturação de água ou nos instantes iniciais da secagem, uma vez que, nesse momento, os gradientes de umidade no interior das partículas ainda não se estabeleceram. A pressão de vapor (P_v) controla a evaporação de água na superfície do sólido e está diretamente relacionada ao equilíbrio termodinâmico entre as fases líquido e vapor. Para que a transferência de água da fase líquida para a fase gasosa aconteça, é necessário que a pressão de vapor na corrente gasosa seja inferior à pressão de vapor máxima (P_{v0}) nessa mesma corrente para uma temperatura determinada, o que é conhecido como condição de saturação para a fase gasosa.

Dito isso, e sabendo que o ar é a principal fase gasosa utilizada em processos de secagem, é bastante útil definir uma variável que quantifique a quantidade absoluta de água presente nesse ar, assim como uma outra que permita uma quantificar percentualmente o quão próximo da saturação em água está esse ar. Assim, definem-se primeiro a umidade absoluta do ar, normalmente representada por Y , e que é definida como a razão entre a massa de vapor de água no ar e a massa total de ar, sendo de grande importância na descrição dos processos de secagem. Já a umidade relativa do ar, doravante chamada de RH, é definida pela razão entre a pressão de vapor P_v e a pressão de saturação na temperatura correspondente P_{v0} , descreve a fração da umidade de saturação presente no ar (MUJUMDAR, 1988). Uma das formas de calcular a umidade absoluta, que será utilizada mais adiante na modelagem do processo aqui estudado, está dada na Equação 1:

$$Y = 2,17 \frac{P_w}{T_g} \quad (1)$$

Por sua vez, a secagem controlada pelos mecanismos internos de transferência de massa requer uma maior atenção, uma vez que estes mecanismos são predominantes por um maior período de tempo num processo de secagem típico e o transporte de massa pode ocorrer de diversas maneiras, tais como a difusão de líquidos, difusão de vapor no interior do material, difusão de Knudsen ou por gradientes de pressão hidrostática; maneiras essas que precisam ser plenamente compreendidas caso a caso, se há o desejo aumentar a eficiência de certo processo de secagem. A partir da umidade crítica, que acontece quando é removida a umidade em excesso

da superfície dos materiais, os mecanismos internos são predominantes até ser atingida a umidade de equilíbrio dinâmico, que sinaliza o término da secagem. A difusão da umidade ocorre pela formação de gradientes no interior do material a ser seco, sendo estes causados pela evaporação de água na superfície desse material, uma vez que os mecanismos convectivos num processo típico de secagem são muito mais rápidos que os de difusão, e a umidade na superfície desse material é menor que em qualquer ponto de seu interior, o que gera uma diferença de concentrações de água e, por conseguinte, força motriz para que aconteça mais transferência de massa.

A difusão de vapor e líquido ocorrem simultaneamente no interior do material. Na secagem controlada pelos mecanismos internos, alguns parâmetros operacionais, como a velocidade do ar no secador não possuem uma influência significativa sobre as taxas de secagem, já que não contribuem para a transferência de massa interna ao material. Outros parâmetros, no entanto, são de grande importância para que o processo alcance taxas de secagem satisfatórias, especialmente a temperatura do ar, que influencia na sua saturação e, por conseguinte, capacidade de transportar água, e o tempo de residência deste no secador, já que um tempo de residência maior implica em maior contato entre as fases e possibilidade maior do ar carregar consigo moléculas de água.

Durante a secagem, a água é removida de um material (que doravante pode ser chamado equivalentemente de sólido, uma vez que estes são os tipos de material de interesse do presente trabalho) poroso até que a pressão de vapor na superfície desse material se iguale à pressão de vapor da fase gasosa. Nesta condição, a umidade do sólido é denominada umidade de equilíbrio, e nela não é possível reduzir mais a umidade do material pela exposição deste ao gás, sendo necessária a diminuição de sua umidade relativa, o que é mais facilmente feito pelo aumento de temperatura, que aumenta a pressão de saturação da água, de forma que a pressão de vapor não seja mais igual a ela.

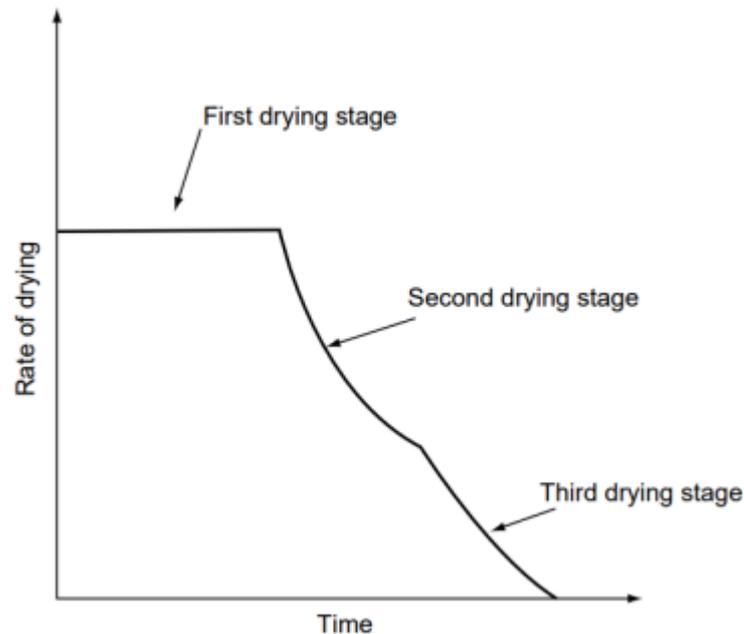
A umidade de um sólido ou pasta pode ser expressa de duas maneiras distintas, sendo elas a umidade em base seca, usualmente chamada de X , e definida como a razão entre a massa total de água e a massa do material totalmente seco, e a umidade em base úmida, ou W , que é definida como a razão entre a massa total de água e a massa total do material, ou seja, a massa de água somada à sua massa seca. As duas formas de expressar a umidade se relacionam por meio da Equação 2:

$$X = \frac{W}{1-W} \quad (2)$$

Os materiais podem ser classificados de acordo com sua estrutura física e interação superficial com a água em três categorias gerais, sendo elas: (1) meios porosos não-higroscópicos, que são materiais que apresentam uma estrutura porosa preenchida por líquido; entretanto, toda a umidade se encontra livre, sem qualquer ligação física à superfície do material. Também, não encolhem durante a secagem, preservando sua geometria, sendo exemplos de materiais desse tipo areia, alumina e certas cerâmicas; (2) meios porosos higroscópicos, correspondendo a materiais que, assim como os meios não-higroscópicos, apresentam poros que podem ser preenchidos por líquido; entretanto, são encolhidos pela secagem. Nesses materiais, a água se encontra ligada à superfície pela retenção nos capilares, pela formação de solução no interior do material ou por adsorção. Existe neles a possibilidade da água exceder a capacidade de ligação, sendo alguns exemplos seus argila e biomassa (é a classe que se enquadra; portanto, o bagaço da laranja, objeto desse estudo); por fim, os (3) meios coloidais não-porosos, como o nome sugere, não apresentam uma estrutura porosa, possuindo água ligada apenas à sua superfície e não havendo gradientes de umidade em seu interior. Por conseguinte, toda transferência de massa ocorre em sua superfície, e seus exemplos mais típicos são polímeros (MUJUMDAR, 1988).

Como já foi dito, é importante compreender qual a etapa limitante num processo de secagem, caso seja de interesse otimizá-lo; nesse âmbito, são de importância fundamental as curvas características de secagem, que permitem identificar a predominância dos mecanismos internos ou externos de transferência de massa. Graficamente, uma curva de secagem nada mais é que a plotagem da taxa de secagem em função do tempo. O comportamento característico da curva de secagem para um sólido poroso higroscópico, assim como o é o bagaço de laranja, está representado na Figura 2.2, sendo possível identificar nela três estágios distintos: o primeiro possui taxa de secagem constante, enquanto no segundo terceiro há uma diminuição da taxa de secagem conforme passa o tempo, mas em acelerações diferentes.

Figura 2.2 - Curva característica para a taxa de secagem em função do tempo.



Fonte: (MUJUMDAR, 1988)

Na superfície do sólido existem dois fenômenos distintos que promovem a transferência da água para a corrente gasosa: (1) a vaporização, sendo essa a convecção na interface sólido-gás e o posterior arraste pelo movimento da corrente gasosa, e (2) a evaporação, que é um fenômeno lento ocorrido quando a pressão de vapor na interface sólido-gás é igual à pressão da fase gasosa. No primeiro período de secagem, ou seja, naquele em que a taxa de secagem permanece constante, a superfície do material se encontra completamente recoberta por água em excesso, que é removida do meio poroso pelo processo de vaporização; assim, os mecanismos externos são considerados predominantes nesta etapa. À medida que a secagem continua e uma maior quantidade de umidade vai sendo removida do material, parte de sua superfície é exposta diretamente à corrente gasosa, sendo que essa parte terá a cinética do processo controlada por mecanismos internos de gradiente, enquanto que a fração de superfície que ainda possui umidade em excesso terá a cinética controlada por mecanismos convectivos externos. Como os mecanismos internos de transferência de massa são consideravelmente mais lentos que os externos, a taxa de secagem total começa a diminuir nessa fase. Evidentemente, em algum momento toda ou quase toda a superfície do material poroso estará isenta de umidade em excesso, e nesse ponto do tempo é que se dá como iniciada a terceira etapa de secagem, com uma redução mais acentuada (ou seja, uma aceleração negativa maior) da taxa de secagem. Moléculas de água ligadas ao meio poroso podem ser removidas nos estágios de secagem à taxa

decrecente, enquanto que no estágio de secagem à taxa constante ocorre somente a remoção de água não ligada (MUJUMDAR, 1988).

Em engenharia e principalmente em ciência de materiais, quase nunca é razoável generalizar toda uma classe de materiais por um comportamento único; ou seja, a Figura 2.2 não é representativa de todos os sólidos não-porosos higroscópicos, já que alguns destes podem, por exemplo, não ter um período observável de secagem à taxa constante. Mesmo para materiais classificados sob um mesmo rótulo é possível que defeitos de superfície significativos gerem propriedades diferentes.

Desse modo, caso seja de interesse, pode ser definida a curva característica de cada material a partir de uma variável denominada umidade adimensional do sólido. Para os objetivos desse trabalho, não se julgou necessário tal aprofundamento.

2.3. LEITOS DE JORRO

Usualmente, leitos de jorro são uma operação unitária pouco vista em um curso tradicional de engenharia química (seja pela falta de tempo para abranger esse tipo de secador ou pelo foco maior que se tem usualmente em equipamentos de maior escala que possuem princípios de funcionamento semelhantes), uma vez que suas aplicações são limitadas pelas atuais barreiras na escalabilidade desse equipamento, sendo ele viável economicamente apenas para escalas menores, devido à ampla necessidade energética para que se atinja um estado de jorro. Ainda assim, estudar esse equipamento é bastante adequado se for levada em consideração a secagem de sólidos (ou mesmo a secagem de suspensões, como é a do leite para produção de leite em pó) por pequenas e médias indústrias e cooperativas, uma vez que estas constituem a maioria numérica das indústrias em todo e qualquer país.

Em síntese, o funcionamento de um leito de jorro convencional pode ser descrito como se faz a seguir: pequenas esferas de um material adequado ao que se deseja secar (polímeros e materiais vítreos são os mais comuns por sua baixa reatividade e baixo potencial de contaminação de produtos alimentícios, que são usos comuns desse leito) são colocadas na base cônica de um leito, que pode funcionar à batelada ou de modo contínuo em relação ao sólido a ser seco. Em ambos os casos, um fluxo de fluido aquecido à altas velocidades é alimentado à base, de forma que provoque um movimento circular vigoroso dessas esferas, fazendo com que a área de contato para a transferência de massa e energia seja aumentada e a secagem possa progredir. Essa corrente de fluido, que na grande maioria dos casos se trata de ar atmosférico, é responsável por retirar a umidade do sólido sendo seco.

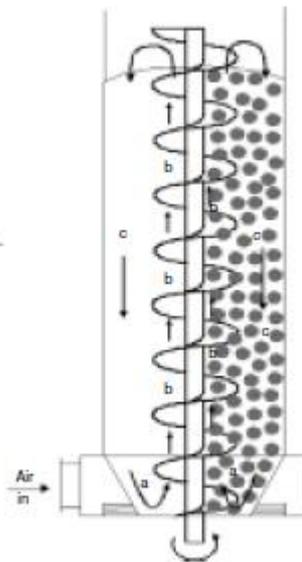
Historicamente, o primeiro leito de jorro foi desenvolvido em 1954 como um método alternativo para um leito fluidizado na secagem de grãos de trigo. Pela circulação vigorosa de partículas, foi permitido que se utilizasse uma temperatura do ar muito maior que a comum sem que houvesse dano às partículas de grãos. Em 1974, já haviam 250 publicações, patentes inclusas, sobre esse tipo de leito, enquanto que em 2010 esse número já era maior que 1300, o que indica que houve um acentuado interesse na nova tecnologia quando de sua proposição ao meio científico, e que ainda há pesquisa nesse tema, ainda que não tenha acelerado de forma exponencial (MATHUR; EPSTEIN, 1974) (EPSTEIN; GRACE, 2010).

Tanto é um equipamento efetivo que logo em 1962, apenas 8 anos após a primeira publicação, a primeira unidade comercial foi instalada no Canadá, com o objetivo de secar ervilhas, lentilhas e linho. Desde essa data, muitas outras unidades foram construídas no mundo

todo com funções variadas, desde a secagem evaporativa de soluções, suspensões e pastas num leito de jorro com partículas inertes, assim como para revestimento, granulação. Os processos a nível comercial bem-sucedidos são sobretudo físicos, mas já existe a pesquisa de alguns processos químicos em leito de jorro a nível de bancada (EPSTEIN; GRACE, 2010).

Atualmente, existem várias pesquisas acerca de qual é a melhor estrutura para um leito de jorro levando em conta as características fluidodinâmicas de cada aplicação e como se pode otimizar as transferências de massa e energia nele, de forma a diminuir custos de operação pelo uso mais efetivo da energia (uma vez que o ar precisa ser aquecido a temperaturas e velocidades relativamente altas para que passe mais efetivamente pelo leito) sem danificar o material sendo seco. Um exemplo disso pode ser visto nos leitos com agitação mecânica, assim como mostra a Figura 2.3.

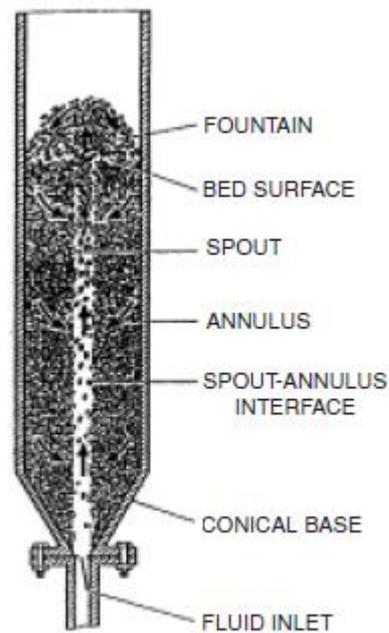
Figura 2.3 - Representação de uma das possíveis modificações estruturais no leito de jorro, a agitação mecânica.



Fonte: (EPSTEIN; GRACE, 2010)

Entretanto, como o artigo de (FREIRE et al., 2017) se utiliza de um leito de jorro convencional, e esse artigo é a base para a modelagem aqui utilizada, sua estrutura está dada em um pouco mais de detalhe na Figura 2.4. O canal central do leito é chamado de canal de jorro, sendo a região que o circunda denominada de ânulo. Os sólidos inertes circulando no leito são chamados de fonte, sendo que a entrada de fluido é feita pela parte inferior da base cônica.

Figura 2.4 – Representação esquemática de um leito de jorro convencional.



Fonte: (EPSTEIN; GRACE, 2010)

Evidentemente, existem inúmeros critérios e regras para determinar cada mínimo detalhe de construção de um equipamento desse, pois uma simples mudança de ângulo na base cônica pode causar uma diferença significativa no desempenho de uma secagem, mas para um trabalho de simulação dos processos de transferência não são de tanta relevância, uma vez que os dados vêm de um equipamento já construído.

2.4. MODELAGEM MATEMÁTICA

Por sua curiosidade e engenhosidade natural, o ser humano sempre foi propenso a construir modelos para melhor compreender o ambiente que o cerca. Ora, pode-se argumentar que as mais antigas pinturas rupestres são modelos, ou seja, abstrações da realidade que serviram à alguma finalidade para seus autores. Contudo, é óbvio que o grau de adequabilidade de um modelo vai sempre depender da finalidade para que é usado, motivo pelo qual pode-se ou não valer de simplificações. Usando como exemplo uma vaca, observa-se como podem ser diferentes as possibilidades de sua representação: para o homem primitivo, autor das pinturas rupestres supracitadas, um risco vertical talhado na parede da caverna em que sua tribo habita pode representar essa vaca de maneira satisfatória, mas o mesmo não se aplica para um estudante de fluidodinâmica que busca fazer simulações computacionais da aerodinâmica desse animal; para ele, um modelo constituído de vários polígonos sendo o mais fiel possível à imagem de um animal real será muito mais adequado, modelo esse que também serve muito bem para um desenvolvedor de jogos, por exemplo, mas não para um estudante que acabou de ingressar na graduação, para o qual uma esfera pode ser uma forma de representação mais adequada.

Como ilustrado, um modelo tem diversas ferramentas para ser expressado, sejam elas artísticas, matemáticas e mesmo textuais; contudo, dando enfoque ao que interessa para o presente trabalho, que são modelos matemáticos, (EYKHOFF, 1974) os definiu como sendo uma representação dos aspectos essenciais de um sistema existente (ou a ser construído) que permitam transmitir conhecimento desse sistema de uma forma utilizável.

A engenharia química não é nada avessa a esse tipo de modelo, uma vez que os balanços de massa e energia, a termodinâmica e os fenômenos de transporte (ou seja, os três principais pilares teóricos) usam como ferramentas diversos modelos matemáticos que, com as corretas hipóteses e considerações, podem ser aplicados em equipamentos já existentes ou a ser projetados e possibilitarão descrever numericamente a realidade de um processo, ou mesmo de uma indústria como um todo.

Além disso, um bom modelo é ideal para que possa ser exercido um bom controle do processo, uma vez que, na prática, todo processo apresenta desvios nas variáveis de interesse em relação ao ponto ótimo para elas definido quando de seu projeto.

Por mais que sejam uma ferramenta muito útil, que poupa muito tempo ao permitir uma abstração sem medições e experimentação repetitiva, nenhum modelo, pelo menos no âmbito

de processos químicos, consegue hoje incorporar todas as nuances presentes num processo real; por isso, é sempre muito importante levar em conta hipóteses coerentes e razoáveis, que, por sua vez, exigem alguém com profundo conhecimento no processo a ser modelado para serem propostas.

Desse modo, há diversas categorizações que podem ser usadas quando se modela um processo; a seguir, serão exploradas aquelas mais pertinentes à simulação feita e, em sequência, exposto o modelo desenvolvido por (FREIRE et al., 2017).

2.4.1. CLASSIFICAÇÕES DE MODELOS MATEMÁTICOS

De forma mais generalizada, há dois possíveis caminhos para o desenvolvimento de um modelo matemático, sendo o primeiro baseado em leis físico-químicas (como balanço de massa e energia, cinética e termodinâmica), que pode ser chamado também de determinístico e é mais atraente por abrir a possibilidade de projeto, e o segundo em descrições empíricas, ou seja, observações e experiência em processos químicos que permitem um ajuste matemático descrevendo-o sob condições limitadas e definidas (EDGAR; HIMMELBLAU; LASDON, 2001).

Ainda, existem outras possibilidades de classificação, sendo a primeira delas quanto à linearidade do modelo: modelos lineares são aqueles em que a variável dependente ou suas derivadas aparecem somente como uma variável de primeira ordem, enquanto os não-lineares englobam todo o restante das situações possíveis. Essa classificação é útil pois modelos lineares são em geral os que apresentam soluções e manipulações mais simples que os não-lineares (EDGAR; HIMMELBLAU; LASDON, 2001).

Em função das variáveis dependentes permanecerem ou não constantes com respeito ao tempo, pode-se ainda classificar modelos em estado estacionário ou estado transiente. Na primeira dessas categorias, a derivada parcial de cada variável dependente com relação ao tempo é 0, ou seja, não há mudança delas com o tempo, enquanto que, para o estado transiente, essa derivada assume valores estritamente diferentes de 0. Novamente, modelos em estado transiente apresentam um grau de complexidade consideravelmente maior que aqueles em estado estacionário (EDGAR; HIMMELBLAU; LASDON, 2001).

Em mais uma forma de classificação, pode-se determinar um modelo a parâmetros concentrados ou distribuídos. Um modelo com parâmetros concentrados implica que as variações espaciais de diversas propriedades de estado do sistema podem ser consideradas

constantes em todo o volume de controle utilizado, enquanto que um modelo com parâmetros distribuídos leva em conta variações detalhadas em cada ponto do sistema em estudo. Todo processo real apresenta parâmetros distribuídos, uma vez que algumas variações dos estados das variáveis sempre ocorrerão em algum ponto de sua extensão; contudo, se essas variações forem pequenas o suficiente, usualmente considera-se como hipótese que é possível a aproximação para um modelo de parâmetros concentrados. Principalmente na simulação e otimização, é muito raro que um processo mais complexo seja modelado ao mesmo tempo como transiente e com parâmetros distribuídos, uma vez que isso gera equações diferenciais parciais que nem sempre têm soluções possíveis (EDGAR; HIMMELBLAU; LASDON, 2001).

Por fim, as variáveis que o modelo possui podem ser discretas ou contínuas: variáveis contínuas podem assumir qualquer valor real num dado intervalo, enquanto que variáveis discretas possuem valores bem determinados; por exemplo, apenas números inteiros. Na engenharia química não é incomum que essas duas variáveis apareçam juntas num mesmo equipamento ou processo, como acontece num compressor, por exemplo. Ressalta-se que variáveis discretas apresentam dificuldades em termos de otimização e simulação (EDGAR; HIMMELBLAU; LASDON, 2001).

Feitas essas distinções, pode-se definir o modelo apresentado na sequência como determinístico, altamente não-linear, em estado transiente, com parâmetros concentrados e com variáveis contínuas, o que caracteriza uma modelagem ainda bem complexa, por mais que algumas hipóteses e simplificações tenham sido adotadas.

2.4.2. MODELO MATEMÁTICO DA SECAGEM DE BAGAÇO DE LARANJA EM LEITO DE JORRO

O modelo aqui apresentado foi retirado de (FREIRE et al., 2017). Esse é um modelo adaptativo, ou seja, baseado em medições de umidade e temperatura dos sólidos e do gás em determinados instantes de tempo, busca, por uma otimização que contém dentro de si uma integração numérica, estimar os parâmetros k , h_A e Q_{loss} , que são representativos dos fenômenos de transferência de massa e energia. Sua dedução foi feita por balanços de massa global sob as seguintes hipóteses: (1) o leito de jorro se comporta como um tanque perfeitamente agitado; (2) a fase gás se comporta como um gás ideal; (3) o bagaço de laranja já seco inserido no leito é inerte, ou seja, não interfere no processo; (4) não há mudanças abruptas nem nas variáveis de estado nem nos parâmetros físicos. Feitas as simplificações, as Equações 3-7 determinam o resultado obtido.

Balanço de massa

$$R_w = k(X_s - X_{eq}) \quad (3)$$

$$\frac{dX_s}{dt} = -R_w \quad (4)$$

$$\frac{dY_g}{dt} = \frac{\dot{Q}}{M_g} \rho_{gi} (Y_{gi} - Y_g) + \frac{M_s}{M_g} R_w \quad (5)$$

Balanço de energia

$$\frac{dT_s}{dt} = \frac{1}{C_{p_s} + X_s C_{p_w}} \left[\frac{hA(T_g - T_s)}{M_s} - R_w Q_{latent} \right] \quad (6)$$

$$\frac{dT_g}{dt} = \frac{\dot{Q}}{M_g} \rho_{gi} (T_{gi} - T_g) + \frac{1}{C_{p_g} + Y_{gi} C_{p_w}} \left[-\frac{hA(T_g - T_s)}{M_g} - R_w \left(\frac{M_s}{M_g} \right) C_{p_w} (T_g - T_s) - Q_{loss} \right] \quad (7)$$

A fim de calcular alguns dos parâmetros necessários para essas equações, o conjunto de Equações 8-15, assim como a Equação 2, são de fundamental importância, e foram retiradas de (NARIMATSU, 2004) e (PERAZZINI et al., 2016).

Calor específico do sólido

$$C_{p_s} = 0.72 + 6.38 \times 10^{-3} T_s \quad (8)$$

Calor específico do gás

$$C_{p_g} = \frac{1}{MW_g} (28.94 + 0.41 \times 10^{-2} T_g + 0.32 \times 10^{-5} T_g^2 - 2.00 \times 10^{-9} T_g^3) \quad (9)$$

Calor específico da água líquida

$$C_{p_w} = 75.4 / MW_w \quad (10)$$

Calor específico do vapor de água

$$C_{p_v} = \frac{1}{MW_w} (33.46 + 0.69 \times 10^{-2} T_g + 0.76 \times 10^{-5} T_g^2 - 3.59 \times 10^{-9} T_g^3) \quad (11)$$

Densidade do gás

$$\rho_g = 2.25 + 3.59 \times 10^{-3} T_g \quad (12)$$

Umidade no equilíbrio (equação de Halsey modificada)

$$X_{eq} = \left[-\frac{\exp(-0.24 T_g + 10)}{\ln(RH)} \right]^{1/2.25} \quad (13)$$

Pressão do vapor saturado

$$P_v = 6.00 \times 10^{\frac{7.34T_g}{T_g + 229.40}} \quad (14)$$

Pressão de vapor

$$P_w = \frac{P_v RH}{100} \quad (15)$$

Por mais que tenha sido consideravelmente simplificado em relação ao que acontece no processo real, o modelo aqui apresentado ainda é bastante complexo, e precisará de métodos numéricos bastante específicos para que consiga atualizar os parâmetros de transferência de massa e energia em um intervalo de tempo menor que aquele no qual uma nova medição estará disponível para ajuste.

Nota-se, por fim, que segundo (FREIRE et al., 2017), a hipótese de mistura perfeita durante todo o tempo de secagem é a mais fraca das assumidas, o que leva à possibilidade de que isso prejudique a extensão desse modelo a outros sistemas.

2.5. MÉTODOS NUMÉRICOS

Como já foi pontuado, esse é um modelo matemático considerado como adaptativo, e isso significa que, baseado em valores experimentais de temperatura e umidade do gás e do sólido tomados em pontos discretos de tempo, serão feitos ajustes de três parâmetros, a saber: k , h_A e Q_{loss} .

Computacionalmente, isso pode ser feito através de dois métodos aninhados: (1) em primeiro lugar, é preciso definir no software de escolha os quatro balanços de massa e energia, que são integrados numericamente com uma estimativa inicial dos parâmetros buscados; (2) depois, define-se uma função desvio entre os valores integrados e os medidos em certo tempo, fazendo, por fim, uma otimização objetivando minimizar esse desvio.

Contudo, tanto a integração como a otimização são custosos a nível computacional; por isso, optou-se por desacoplar as equações de balanço de massa e energia do tempo, sob a hipótese bastante razoável de que nem as variáveis de estado nem os parâmetros físicos variam bruscamente com o tempo.

A seguir, cada um dos métodos supracitados será explicado de forma sucinta.

2.5.1. SOLUÇÃO NUMÉRICA DE EQUAÇÕES DIFERENCIAIS

É de cunho introdutório nos cursos de cálculo integral e equações diferenciais que nem toda função pode ser integrada analiticamente, seja pela não existência de uma expressão primitiva a ela, seja pela falta de técnicas capazes de fazê-lo, e o mesmo é verdade para equações diferenciais que, via de regra, necessitam de uma integração para serem solucionadas.

Principalmente para o caso de não haverem técnicas analíticas capazes de fazê-lo, o advento dos computadores permitiu que fossem usados outros métodos para estimar a integral de expressões mais complexas, uma vez que, para um computador, métodos que antes seriam extremamente exaustivos a humanos pela sua mecanicidade e repetição são feitos em milésimos de segundo, desde que exista um algoritmo adequado para isso.

Quando melhor caracterizadas, as expressões que necessitam de integração para o balanço de massa e energia são equações diferenciais não-lineares com condição inicial conhecida, o que permitirá a formulação de um problema de valor inicial; ou seja, um problema no qual o valor da variável dependente é conhecido no início do intervalo de tempo que se empreende a integração.

2.5.2. MÉTODOS DE RUNGE-KUTTA

Evidentemente, existem inúmeros métodos e adaptações de métodos possíveis para resolver um problema de valor inicial numericamente; contudo, nos dias de hoje os métodos de Runge-Kutta e suas derivações são os predominantes na maior parte dos algoritmos pré-instalados em pacotes computacionais.

A ideia básica destes métodos é aproveitar das qualidades dos métodos de série de Taylor, ou seja, permitir a solução para qualquer problema de valor inicial e ser de passo um (no cálculo numérico, passo um significa que um algoritmo depende apenas de um conjunto de valores anteriores para as variáveis dependentes e independentes a fim de prosseguir com a integração numérica; evidentemente, o primeiro conjunto de valores é aquele dado pela condição inicial, que é conhecida), ao mesmo tempo que não calcula derivadas das funções, como seria feito caso se utilizasse o método da série de Taylor (em que cada ordem exigiria a estimativa de uma derivada de grau cada vez maior, que muito possivelmente seria feita utilizando um outro método numérico em um caso de resolução por computador), o que os fazem computacionalmente aceitáveis; contudo, é necessário calcular a função em uma maior quantidade de pontos quando comparado à resolução por séries de Taylor, justamente por não empreender o cálculo de derivadas (RUGGIERO; DA ROCHA LOPES, 1996).

Uma desvantagem dos métodos de Runge-Kutta é que não existe para eles uma estimativa simples do erro e, como métodos de resolução numéricos de PVI discretizam um passo de tempo para avançar no cálculo da função e/ou de suas derivadas, essa desvantagem pode gerar em dificuldades no momento de estimar o passo adequado (de fato, isso aconteceu algumas vezes durante a fase de testes da simulação realizada) (RUGGIERO; DA ROCHA LOPES, 1996).

Tanto a biblioteca SciPy, do Python, que foi o pacote computacional utilizado para a resolução do problema no presente trabalho, como os métodos intrínsecos ao MATLAB, no qual o script de (FREIRE et al., 2017) foi desenvolvido, se utilizam de métodos de Runge-Kutta de ordem 4 para resolver os PVI's presentes. A Equação 16 mostra o método iterativo pelo qual é feito esse cálculo, lembrando-se que o passo h pode ser pré-determinado, ou estimado automaticamente pelo software de escolha.

Método de Runge-Kutta de Quarta Ordem

$$y_{n+1} = y_n + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) \quad (16)$$

na equação acima,

$$k_1 = h \cdot f(x_n, y_n)$$

$$k_2 = h \cdot f(x_n + h/2, y_n + k_1/2)$$

$$k_3 = h \cdot f(x_n + h/2, y_n + k_2/2)$$

$$k_4 = h \cdot f(x_n + h, y_n + k_3)$$

2.5.3. OTIMIZAÇÃO

Como já exposto, a cada nova integração numérica realizada, serão obtidos valores de temperatura e umidade, que precisam ser comparados com os experimentais, e ter seus respectivos desvios minimizados com a alteração das variáveis que controlam a cinética da secagem.

Para que ocorra essa minimização dos desvios, uma otimização é necessária, ou seja, é preciso implementar algum algoritmo que seja capaz de deslocar a função determinada, que é comumente chamada de função objetivo no jargão da otimização, para um de seus pontos de mínimo, seja ele local ou global, dentro de um intervalo de valores coerentes e razoáveis para os fenômenos acontecidos no leito.

A fim de evitar respostas com parâmetros fora da realidade física, como, por exemplo, uma constante cinética k de secagem inferior a 0, é conveniente limitar a região de busca, sendo, no caso em estudo, sempre o valor de 0 utilizado como limite inferior, e o limite superior depende de cada variável (h_A , por exemplo, tem valores várias ordens de grandeza acima de k , pois a transferência de calor por convecção é muito mais significativa que os movimentos de umidade dentro do sólido).

Em termos de método, o problema em voga exige algum que seja capaz de otimizar funções não-lineares (comumente referido como programação não-linear em textos de otimização), e que não use de suas derivadas, uma vez que a expressão das funções é bastante complexa, principalmente no balanço de energia, e o tempo computacional empregado para derivá-las seria um impeditivo para a execução da otimização antes da aquisição do próximo conjunto de dados pelos medidores.

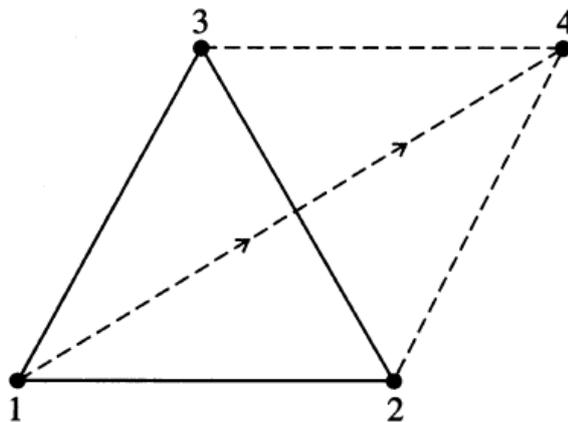
2.5.4. MÉTODO DE NELDER-MEAD

O método escolhido para o problema em questão foi o de Nelder-Mead, proposto pela primeira vez em 1965, mas que desde lá recebeu várias alterações e modificações até ser implementado, nativamente ou como complemento, em diversos softwares disponíveis comercialmente e de forma gratuita.

Esse método se baseia no método de busca simplex, que foi formulado em 1962 e usa um polígono regular, chamado de simplex, com $(n + 2)$ vértices e dimensão $(n + 1)$ para uma função com (n) variáveis dependentes. Em cada ponto desse polígono, a função é avaliada, e aquele que resultar no maior valor dentre todos os selecionados é descartado, sendo que a nova direção de busca é refletida a partir dele. A partir do momento que nenhum ponto possa ser descartado, a dimensão desse polígono precisa ser reduzida ou aumentada, de modo que se reduza a região de busca e, após uma certa tolerância definida, o valor ótimo possa ser alcançado. Nelder e Mead complementaram esse algoritmo ao adicionar a possibilidade da região de busca se contrair ou expandir continuamente.

Evidentemente, pelo fato de não se utilizar de derivadas, esse método é bastante adequado para formulações mais complexas; contudo, ele não garante que o mínimo possa ser encontrado e é muito sujeito a tender para outros mínimos locais caso uma região de busca inadequada seja definida (EDGAR; HIMMELBLAU; LASDON, 2001). A Figura 2.5 ilustra a reflexão de um simplex em duas dimensões (cujo polígono é um triângulo equilátero), no caso em que o ponto 1 apresenta o maior valor da função.

Figura 2.5 – Reflexão de pontos no método de busca simplex em duas dimensões.



Fonte: (EDGAR; HIMMELBLAU; LASDON, 2001)

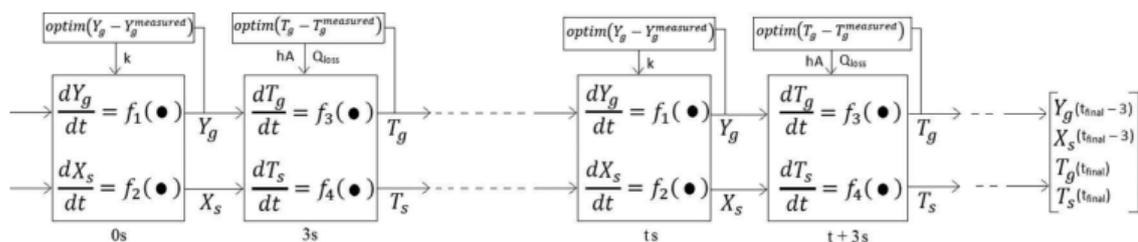
2.5.5. MODELAGEM EM CASCATA

O tempo de execução de cada otimização feita pelo programa é de fundamental importância para suas aplicações futuras, e fazer simultaneamente a integração e otimização de 4 equações diferenciais altamente não-lineares não é simples em termos computacionais. Percebendo esse gargalo, e se baseando em tentativas bem-sucedidas com outros sistemas no passado, (FREIRE et al., 2017) decidiram aplicar o método de simulação em cascata.

Esse método usa como hipótese justificadora o fato que nem as variáveis de estado nem os parâmetros físicos variam de forma abrupta durante a secagem e, como o intervalo de tempo entre a retirada das amostras de umidade e temperatura é bem pequeno, de apenas 3 s, é razoável considerar que, quando acontece variação no parâmetro de transferência de massa, ambos os parâmetros de transferência de energia se mantêm constantes, e vice-versa.

Como ilustração, a Figura 2.6 traz um diagrama de como esse método se comporta. No intervalo que vai de 0 a 3 s, assume-se que não ocorre alteração das variáveis de transferência de calor; já no intervalo de 3 a 6 s, assume-se que a transferência de massa se mantém com o mesmo valor de variável que resultou da otimização do intervalo de 0 a 3 s. Esse processo é repetido de maneira alternada até o tempo final das medições, no caso de se manipular um banco de dados, ou até ser interrompido, para uma possível aplicação futura de medidas on-line.

Figura 2.6 – Representação da modelagem em cascata.



Fonte: (FREIRE et al., 2017)

2.6. LINGUAGENS DE PROGRAMAÇÃO

Hoje em dia existem mais linguagens de programação disponíveis do que nunca, algumas mais simples, outras mais complexas; algumas voltadas ao desenvolvimento web, outras ao back-end. Com tantas opções, fica a cargo do usuário escolher a qual irá se dedicar mais; contudo, alguns fatores podem favorecer certas linguagens em relação às outras. Por exemplo, poucas pessoas buscam aprender Assembly, ainda que seja importante essa linguagem ter seus especialistas, uma vez que, malgrado seja de baixo nível (na realidade, muito próxima do nível das máquinas), ainda figura em aplicações importantíssimas; por outro lado, observa-se um crescimento cada vez maior do uso de linguagens de alto nível (que são mais compreensíveis ao usuário), sobretudo Python, que já há muitos anos figura entre as mais utilizadas, variando sua posição conforme a pesquisa, mas nunca fora do top 5.

Como Python foi a linguagem escolhida para a elaboração do presente trabalho, por motivos, dentre outros, de uma maior familiaridade do autor com ela, ampla comunidade online disponível para dúvidas, detecção de erros e desenvolvimento de módulos muito úteis aos mais diversos fins e a gratuidade do software, algum aprofundamento nessa linguagem será feito a seguir.

2.6.1. PYTHON

A construção de estruturas de dados e algoritmos requer que comuniquemos instruções detalhadas a um computador, e uma excelente maneira de executar tais comunicações é através de uma linguagem de alto nível, como o é o Python. Essa linguagem de programação foi desenvolvida inicialmente por Guido van Rossum no início da década de 1990, e desde lá ocupa uma posição de destaque na indústria e na educação, pela sua simplicidade ao usuário e ampla gama de aplicações. A segunda grande atualização, denominada Python 2, foi lançada em 2000, sendo que a terceira maior versão, Python 3, foi pela primeira vez disponibilizada em 2008. Por ser um software de acesso livre, a última versão, tanto estável como de testes está disponível no site www.python.org, juntamente com tutoriais e toda a documentação necessária para a compreensão da linguagem (GOODRICH; TAMASSIA; GOLDWASSER, 2013).

Existem dois tipos de linguagens de programação quanto à forma que o código é executado: as interpretadas e as compiladas, sendo que Python se enquadra no primeiro tipo, uma vez que todos seus comandos são executados num software denominado de interpretador, que recebe um comando, o avalia e retorna seus resultados. Usualmente, linguagens interpretadas são consideradas mais lentas, pelos motivos que o próximo subtópico trará;

contudo, para os fins desse trabalho, esse tempo de execução é quase que insignificativo, e só será perceptível em bancos de dados com diversos milhares a milhões de linhas; o que permite uso de uma linguagem relativamente mais simples.

2.6.2. LINGUAGENS DE PROGRAMAÇÃO COMPILADAS E INTERPRETADAS

Uma linguagem compilada tem seu conjunto de instruções, ou código, compilado diretamente em instruções para máquinas, ou seja, instruções em linguagem do mais baixo nível possível, conhecida popularmente como código binário, que é indecifrável por humanos e executada na máquina alvo. Exemplos das linguagens mais famosas que se utilizam desse paradigma são C, C++, C# e Java.

Por outro lado, uma linguagem interpretada não tem esse processo de tradução para o baixo nível, uma vez que não é executada diretamente pela máquina alvo, e sim por uma máquina virtual, que é o interpretador, e justamente por isso têm um tempo de execução mais lento que suas contrapartes compiladas. Seus exemplos mais notáveis são Python, JavaScript e PHP.

2.6.3. PROGRAMAÇÃO ORIENTADA AO OBJETO

Ainda que o uso de uma linguagem interpretada seja justificável, sempre que possível é de bom grado se utilizar de técnicas e paradigmas que permitam uma execução mais acelerada do código, uma vez que esse é o aspecto de mais fundamental importância para a aplicação aqui estudada; dessa forma, optou-se por se utilizar do paradigma da programação orientada ao objeto (POO), ainda que se tenha perdido certa fidelidade ao código em MATLAB de (FREIRE et al, 2017), que foi feito de forma procedural, ou seja, usando apenas da lógica e de um conjunto de instruções sequenciais.

Sendo um paradigma da programação, existem alguns princípios gerais da POO, que serão sintetizados a seguir, mas ressalta-se que não necessariamente existe a necessidade de todos esses princípios estarem contidos num código que segue a POO, servindo, mais que qualquer coisa, apenas como pontos de referência.

O primeiro princípio, e talvez o mais básico de todos, é o de múltiplas representações, que determina que, quando uma operação é evocada num objeto, o próprio objeto determina qual código será executado, tanto que dois objetos que respondem ao mesmo conjunto de operações (ou seja, com a mesma interface) podem usar representações completamente diferentes, desde que cada um traga consigo uma implementação das operações que funcione

com sua representação específica. Essas implementações são denominadas métodos (PIERCE, 2002).

O segundo princípio, que foi extensamente utilizado no código produzido nesse trabalho, é o da encapsulação, que determina que a representação interna de um objeto é separada de tudo aquilo que está fora de sua definição, ou seja, apenas os métodos próprios de um objeto podem manipulá-lo e inspecioná-lo diretamente. Isso significa que mudanças na representação interna de um objeto podem afetar apenas uma região pequena e facilmente identificável do código, de forma que seja mais fácil diagnosticar fontes de erros, ler e manter códigos maiores e/ou mais complexos (PIERCE, 2002).

O terceiro princípio é o de subtipos, que enuncia, resumidamente, que o tipo de um objeto – ou sua interface - é apenas o conjunto de nomes e tipos de suas operações. A representação interna do objeto não aparece em seu tipo, pois não afeta o conjunto de coisas que podemos fazer diretamente com o objeto (PIERCE, 2002).

O penúltimo dos principais princípios da POO é o de herança: objetos que compartilham partes de sua interface também, usualmente, compartilharão alguns dos mesmos comportamentos, e é desejável que esses comportamentos comuns sejam implementados apenas uma vez, o que pode ser atingido pelo uso de classes, que são modelos dos quais os objetos podem ser instanciados, sendo que o mecanismo de subclasses permite que novas classes sejam derivadas de antigas pela adição de implementações de novos métodos, ou, quando necessário, a sobrescrita de um método da classe que deu a origem (PIERCE, 2002).

Por fim, a recursão aberta é uma ferramenta que a maioria das linguagens com objetos e classes possui, e que dá a habilidade de um método evocar outro no mesmo objeto com uma variável especial, que usualmente é a variável `self` ou, em alguns casos, a variável `this` (PIERCE, 2002).

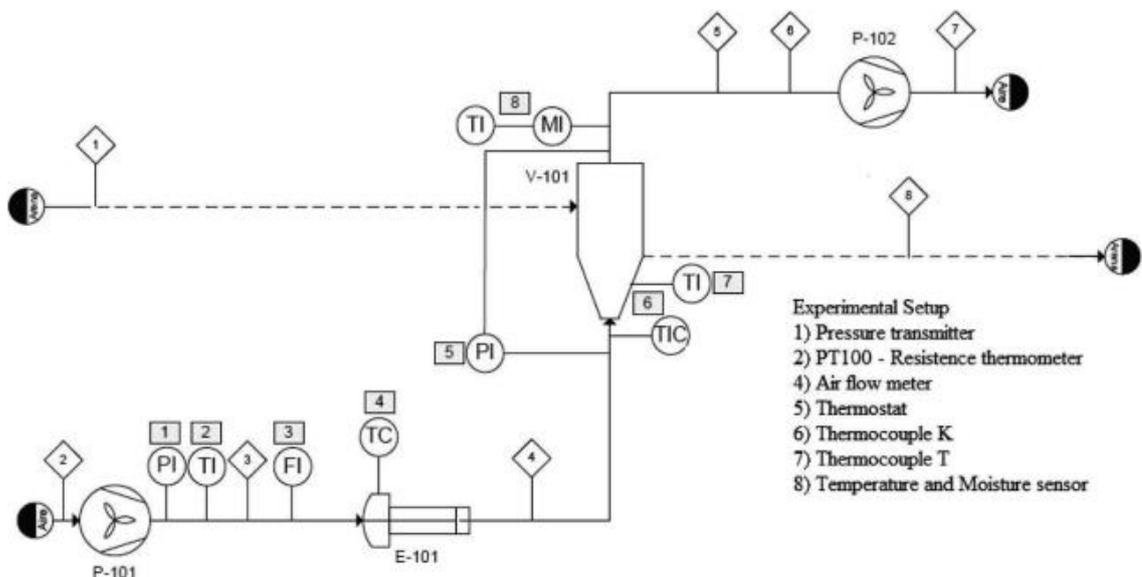
3 – MATERIAIS E MÉTODOS

3.1. BANCO DE DADOS

3.1.1. OBTENÇÃO DOS DADOS

Para que fosse possível a simulação em um software livre, foi necessário que o mesmo banco de dados antes usado para desenvolver o modelo no MATLAB por (FREIRE et al., 2017) fosse utilizado, e esse banco de dados foi obtido a partir do aparato experimental representado na Figura 3.1. Nela, o vaso de aço inoxidável AISI-304 é composto por uma base inferior cônica de aço inoxidável com diâmetro de entrada de 6,0 cm, ângulo do cone α de 32° , altura do cone de 50 cm e uma parte cilíndrica superior (diâmetro do cilindro de 36 cm, altura do cilindro de 60 cm). A altura total, desde a entrada na parte cônica inferior até a saída na parte cônica superior, é de 160,5 cm. O fluxo de ar é regulado manualmente através de um servidor. Um transmissor de pressão mede a queda de pressão através do leito. Um aquecedor elétrico de temperatura, controlado pelo termopar colocado na entrada do leito, aquece o ar antes de entrar no leito. A umidade relativa do ar e a temperatura são medidas na saída do leito por higrômetros capacitivos e um termopar tipo K conectado ao sistema de aquisição de dados (FREIRE et al., 2017).

Figura 3.1 – Fluxograma do aparato experimental utilizado.



Fonte: (FREIRE et al., 2017)

O soprador utilizado foi um RODE 500 fornecido pela MOLVISA, com potência de 4 kW (460 mm de comprimento x 380 mm de largura x 525 mm de altura), responsável por fornecer a pressão necessária para vencer a queda de pressão do processo. O sistema de aquecimento do ar foi realizado por um aquecedor de uma etapa RXP E32348 de 7,5 kW da ELECTRIC INDUSTRIES SA SOLER, e a temperatura foi regulada por um controle PID que atua sobre um SSR. O aquecedor é composto por três resistências em forma de U alimentadas por uma corrente trifásica de 380 V e 50 Hz. A instrumentação necessária para a unidade de aquisição de dados incluiu um transmissor de pressão absoluta, um medidor de vazão, um termostato para o aquecedor, dois termopares, duas sondas PT100, um manômetro diferencial e um medidor de temperatura e umidade com medição conjunta das variáveis (FREIRE et al., 2017).

Maiores informações sobre como o bagaço foi caracterizado e em que formato foi alimentado, qual a alimentação de inertes e em qual proporção, assim como informações gerais sobre o procedimento experimental podem ser encontradas em (FREIRE et al., 2017).

3.1.2. DISPOSIÇÃO DOS DADOS

O banco de dados obtido por (FREIRE et al., 2017) é constituído de um conjunto de 1117 pontos, com a medição de 9 variáveis em cada um deles, em intervalos de tempo de 3 s entre cada uma. A disposição do banco de dados é, então, feita em 10 colunas e 1117 linhas, sendo as colunas correspondentes a: (i) tempo; (ii) vazão volumétrica de ar, em m³/h; (iii) temperatura de entrada do ar, em °C; (iv) temperatura de saída do ar, em °C; (v) temperatura do leite, em °C; (vi) temperatura do filtro, em °C; (vii) umidade relativa, em (%); (viii) diferença de pressão, em hPa; (ix) temperatura do soprador, em °C e (x) pressão total, em hPa.

Para os fins do modelo utilizado, apenas as colunas (i), (ii), (iii), (iv) e (vii) são de interesse, uma vez que, pelas considerações feitas e equações disponíveis, estas serão suficientes para determinar tudo o que é necessário durante a execução do algoritmo. Dessa forma, nas etapas iniciais de execução, as demais medidas foram ignoradas para evitar quaisquer erros decorrentes de mantê-las sem necessidade de sua futura utilização.

Buscando ilustrar a disposição original desse banco de dados, a Tabela 3.1 mostra seus primeiros e últimos pontos.

Tabela 3.1 – Medidas iniciais e finais do banco de dados obtido no artigo de (FREIRE et al., 2017).

t (s)	Q (m³/h)	T_{in} (°C)	T_{out} (°C)	T_{bed} (°C)	T_{filter} (°C)	RH (%)	ΔP (hPa)	T_{soprador} (°C)	P_{total} (hPa)
0	145,2	87,4	58,4	80,5	38	5,0	68	23,2	287
3	145,2	87,8	58,3	80,1	38	5,1	70	23,3	231
6	145,4	87,8	58,3	80,5	38,1	5,1	64	23,2	256
9	145,4	88,2	58,2	80,5	38,1	5,2	64	23,2	237
12	145,4	88,1	58	80,4	38,1	5,3	64	23,3	225
...
3318	177,6	87,3	67	80,8	46,9	3,6	68	27,6	337
3321	178,2	87,5	67,1	80,8	46,9	3,6	62	27,6	368
3324	177,8	87,5	67,1	80,8	46,9	3,6	62	27,6	368
3327	177,6	87,4	67	80,8	46,9	3,6	64	27,5	418
3330	177,2	87,2	67,1	80,8	46,9	3,6	64	27,6	381

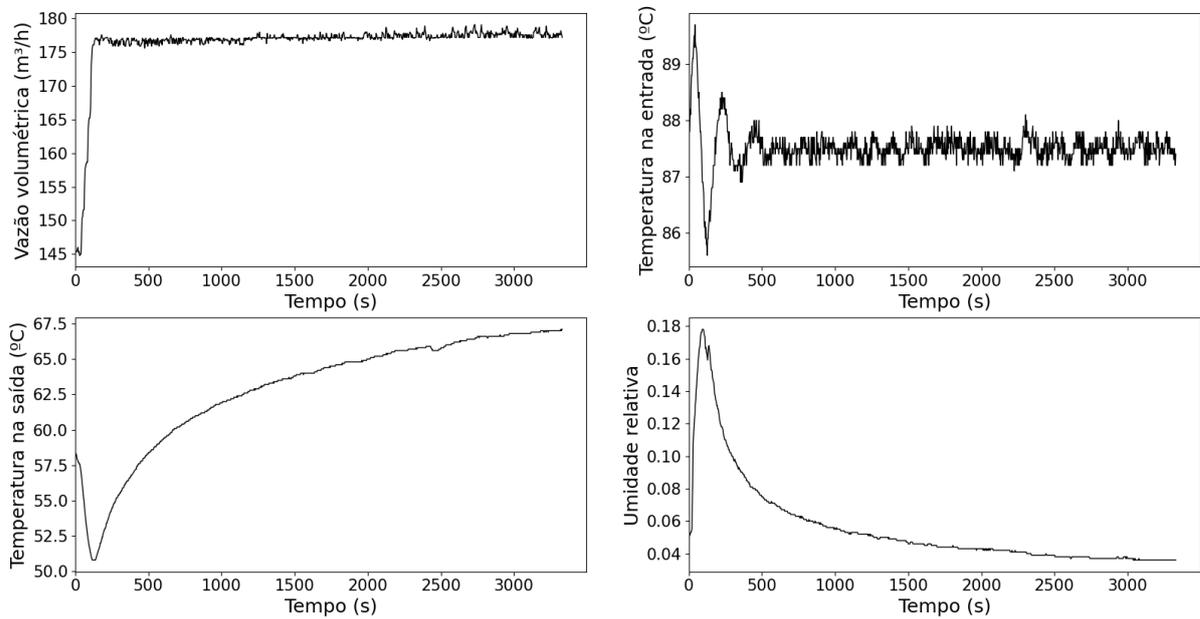
Fonte: Elaborado pelo autor.

3.1.3 TRATAMENTO DO BANCO DE DADOS

Como o algoritmo precisará resolver um problema de valor inicial a cada iteração de uma otimização pelo método de Nelder-Mead, é de importância fundamental que os dados sejam tão pouco ruidosos quanto possível, uma vez que os passos de tempo dados pelo algoritmo de Runge-Kutta na resolução do PVI são definidos de forma automática e, para um banco de dados ruidoso, esses passos podem ser correspondentes a um tempo maior os presentes no seu intervalo de trabalho, gerando um erro e impedindo que o código continue a ser executado.

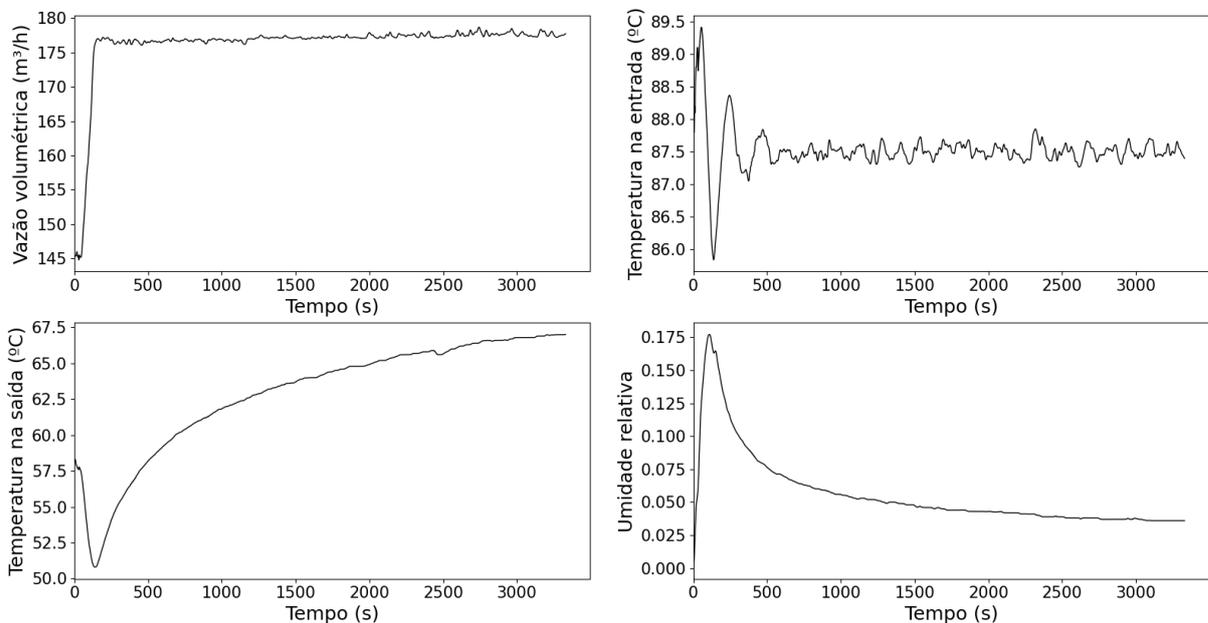
Ao plotar os dados experimentais brutos das variáveis de interesse em função do tempo, observou-se que eram bastante ruidosos, como mostra a Figura 3.2. Dessa forma, foi preciso lançar mão de recursos de filtragem de sinal, que estão disponíveis em grande número e oferecidos por diversos módulos no Python. Por termos de simplicidade de implementação e por atender bem aos objetivos de filtragem aqui pretendidos, escolheu-se um filtro do tipo FIR, oferecido pelo módulo SciPy, com número de coeficientes igual a 10 e frequência de corte de 0,005.

Figura 3.2 – Gráficos das variáveis de interesse para o algoritmo, a partir dos dados experimentais sem tratamento, com destaque para o ruído que apresentam.



Fonte: Elaborado pelo autor.

Figura 3.3– Gráficos das variáveis de interesse para o algoritmo, utilizando um filtro FIR para tratar o ruído nelas presente.



Fonte: Elaborado pelo autor.

Alguns tipos de filtro, como é o caso do FIR ou da filtragem por média móvel, apresentam um significativo efeito de borda no início e/ou no final do banco de dados, que corresponde ao número de coeficientes para eles escolhido. Por esse motivo, os dez primeiros

valores retornados pelo filtro foram substituídos por aqueles presentes no banco de dados original. Para fins de visualização, a Figura 3.3 traz a filtragem das principais variáveis já com o tratamento desses efeitos de borda, já que, em um gráfico, eles têm difícil visualização. Nela, observa-se como foi efetiva a filtragem, tornando os dados muito menos ruidosos e, por conseguinte, facilitando sua utilização no programa computacional desenvolvido.

3.2. MATERIAIS

3.2.1 HARDWARE

Como o tempo de simulação é uma importante variável, principalmente no que tange à determinação do intervalo ideal para retirada de dados pelos medidores presentes no aparato experimental, o hardware em que foi desenvolvido o algoritmo é relevante, pois, pensando numa futura aplicação desse mesmo modelo a um controlador e a medidores online, pode ser que esse tempo varie para mais ou para menos.

Levando em conta que em nenhum momento se lançou mão do recurso de aceleração por GPU no IDE de Python utilizada, os parâmetros mais importantes de se especificar são o processador e a memória RAM do desktop utilizado, que são, respectivamente: Intel® Core™ i5-3330 de 3.00GHz; com 16,0 GB de RAM.

3.2.2 SOFTWARES

De antemão, estava disponível o código utilizado por (FREIRE et al., 2017) para a simulação do leito de jorro, desenvolvido em MATLAB; contudo, como esse é um software pago, ele não pôde ser utilizado no trabalho, já que não havia uma licença disponível.

Mesmo assim, foi necessário encontrar alguma forma de tornar esse código possível de ler através de softwares livres, demanda que foi atendida pelo Visual Studio Code (ou, como é comumente abreviado, VS Code), que é um aplicativo que permite leitura de inúmeras linguagens de programação diferentes sem a necessidade de instalação dos IDEs específicos de cada uma, além de oferecer opções para edição dos respectivos códigos (inclusive, mais de uma opção para uma mesma linguagem a depender da finalidade – por exemplo, desenvolvimento web, desenvolvimento de jogos, etc) por meio de extensões disponíveis em seu “mercado”, que podem ser gratuitas ou pagas, a depender da opção dos seus desenvolvedores.

Ainda que também seja possível editar um código em Python, do início ao fim, pelo VS Code, optou-se por utilizar um IDE específico pela disponibilidade de um maior número de recursos e ferramentas facilitadoras nele. O IDE escolhido foi o PyCharm Community Edition 2022.3.2. Para clarificar um pouco, um IDE significa, em tradução livre, um ambiente integrado de desenvolvimento, e é uma ferramenta bastante útil no desenvolvimento de algoritmos, uma vez que deixa mais visível e dá sugestões de correções e/ou alterações em trechos de código, permite a visualização de diversos arquivos simultaneamente (o que é importantíssimo, sobretudo quando se utiliza dos princípios da POO), e também uma organização de projetos

sem igual, ao definir um ambiente separado para cada um, com suas próprias exigências de módulos e memória.

3.3. DESENVOLVIMENTO DO ALGORITMO

Ao final da adaptação, foi possível desenvolver um algoritmo separado em 4 módulos, sendo que dentre esses há 3 classes, sendo elas: (i) `DataProcessing`, responsável pela leitura, filtragem e processamento dos dados experimentais brutos, contendo funções de conversão de variáveis e retornando vetores com todos os parâmetros adaptados à execução dos balanços de massa e energia; (ii) `Model`, que tem como principal função calcular os balanços de massa e energia, mas também possui funções auxiliares para permitir uma melhor capacidade da leitura desses balanços e tornar a classe como um todo mais encapsulada e com correções mais facilitadas, por conseguinte; (iii) `Simulation`, que, em suma, é responsável pela resolução dos PVI a partir das condições iniciais e estimativa do erro entre os valores experimentais e os calculados; e uma função principal, cujo módulo foi nomeado `CascadeModel`, que é responsável pela simulação em cascata e definição de objetos pertencentes às 3 classes supracitadas.

Pensando em possíveis desenvolvimentos futuros a partir do progresso aqui feito, a separação em módulos permite que, com uma melhor generalização e encapsulação, assim como o desenvolvimento de uma interface para o usuário final, seja possível desenvolver um aplicativo sem grandes dificuldades. Caso seja validado o modelo, tal aplicativo seria de grande ajuda para a implementação de um sistema de controle a esse processo.

Para maiores detalhes, o Apêndice A traz na íntegra o algoritmo, assim como foi usado para determinar os resultados vistos na seção 4 desse documento.

4 – RESULTADOS E DISCUSSÃO

4.1. TEMPO DE EXECUÇÃO

Como algumas vezes ressaltado no decorrer do texto, o tempo de execução é um fator de desempenho importantíssimo, já que o modelo sendo trabalhado é adaptativo, ou seja, atualiza seus parâmetros conforme os dados experimentais fornecidos, e por isso é desejável que, antes da transmissão pelos medidores de um novo ponto experimental, o ponto anterior tenha tido seus parâmetros devidamente ajustados; caso contrário, uma fila incrementalmente maior será estabelecida, de forma que o ajuste de parâmetros em certo tempo não seja representativo dos fenômenos de transporte de massa e energia ocorrendo nesse tempo.

Dito isso, foi proposta uma possível simplificação do modelo a fim de avaliar como isso afetaria o tempo de execução: ao invés de se utilizar da variação de 3 parâmetros, sendo 1 deles no balanço de massa (k) e 2 no balanço de energia (h_A e Q_{loss}), optou-se por assumir Q_{loss} como constante e igual a 0, o que é razoável levando em conta os resultados apresentados por (FREIRE et al, 2017), já que a energia perdida pelo sistema é bastante baixa se comparada com a energia total fornecida a ele; ademais, a ausência desse parâmetro pode ser compensada por uma diminuição em h_A , ou seja, que passaria a ser um coeficiente de transferência de energia aparente.

Além do número de parâmetros variados, a tolerância para finalização da otimização pelo algoritmo de Nelder-Mead exerce uma influência considerável no tempo de execução, ao mesmo tempo que é uma variável que influi diretamente na qualidade dos ajustes feitos; assim, é outro fator importante a ser analisado nesses testes.

Ressalta-se, aqui, que esse tempo tem forte influência do hardware utilizado, sobretudo considerando que aquele descrito na seção 3 já é bastante ultrapassado, já que pertence a um desktop de 10 anos atrás; contudo, a proporção entre um tempo e outro tende a ser a mesma, independente do hardware, e isso é que foi analisado nas simulações feitas.

Para fins de comparação, considerar-se-á que um tempo plausível é, no mínimo, menor que o último tempo presente no banco de dados (3330 s, ou 0,925 h). A partir desses esclarecimentos, pode-se compreender os dados trazidos pela Tabela 4.1.

Tabela 4.2 – Tempos de execução dependendo da tolerância para finalização do algoritmo de Nelder-Mead e número de parâmetros ajustados.

Tolerância	Tempo	Tempo
(-)	(2 parâmetros)	(3 parâmetros)
10^{-2}	0,15 h	1,1 h
10^{-3}	0,19 h	2,3 h
10^{-4}	0,23 h	5+ h
10^{-5}	0,36 h	-

Fonte: Elaborado pelo autor.

Como fica bem claro a partir da Tabela 4.1, ao se utilizar do modelo de 3 parâmetros, o tempo de execução começa a ser tão alto que é um impeditivo para que haja ajuste plausível do processo em tempo real, pelo menos para um intervalo entre a coleta de dados de 3 segundos; assim, há duas opções possíveis para um uso adequado do modelo: (i) a simplificação do ajuste, utilizando-se de apenas 2 parâmetros, o que, como dito anteriormente, é uma hipótese bastante razoável, principalmente se o leito utilizado para obtenção dos dados for isolado do ambiente, o que diminuiria ainda mais a sua perda de calor; (ii) utilização de um intervalo maior para a coleta de dados, ainda utilizando-se do modelo com ajuste de 3 parâmetros, sendo esse intervalo tão maior quanto menor a tolerância desejado na otimização.

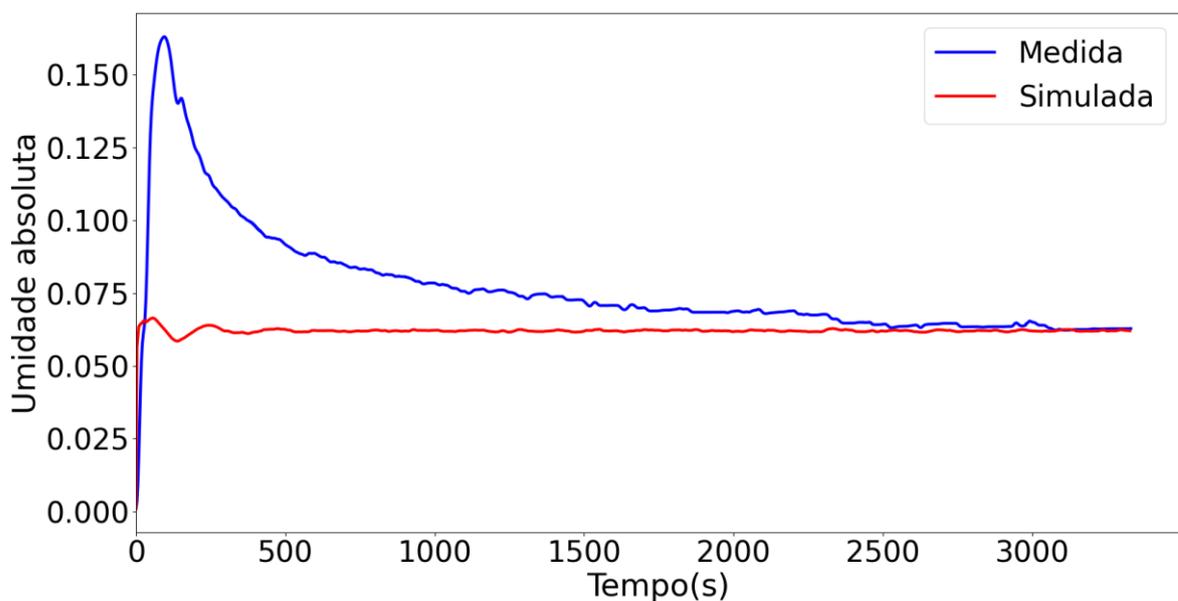
4.2. VALIDAÇÃO DO MODELO

O fato de o programa ter sido executado sem erros não significa nada além de que, matematicamente, não há nada de absurdo com o algoritmo nele presente; ainda assim, isso não tem relação alguma com a validade dos resultados por ele apresentados.

Para validar esse modelo, são necessárias duas análises independentes, sendo que a primeira delas diz respeito à correspondência entre as variáveis medidas e simuladas – estas, para o leito de jorro objeto do estudo, são a umidade absoluta do gás, que está intimamente relacionada com o balanço de massa, e a temperatura de saída desse gás, que, pela hipótese de mistura perfeita, é a mesma que este apresenta no interior do leito, e é mais ligada ao balanço de energia – e a segunda é uma análise da coerência física dos parâmetros ajustados (um exemplo de que o modelo fez um ajuste inválido é se qualquer um dos ajustes resultar num parâmetro negativo).

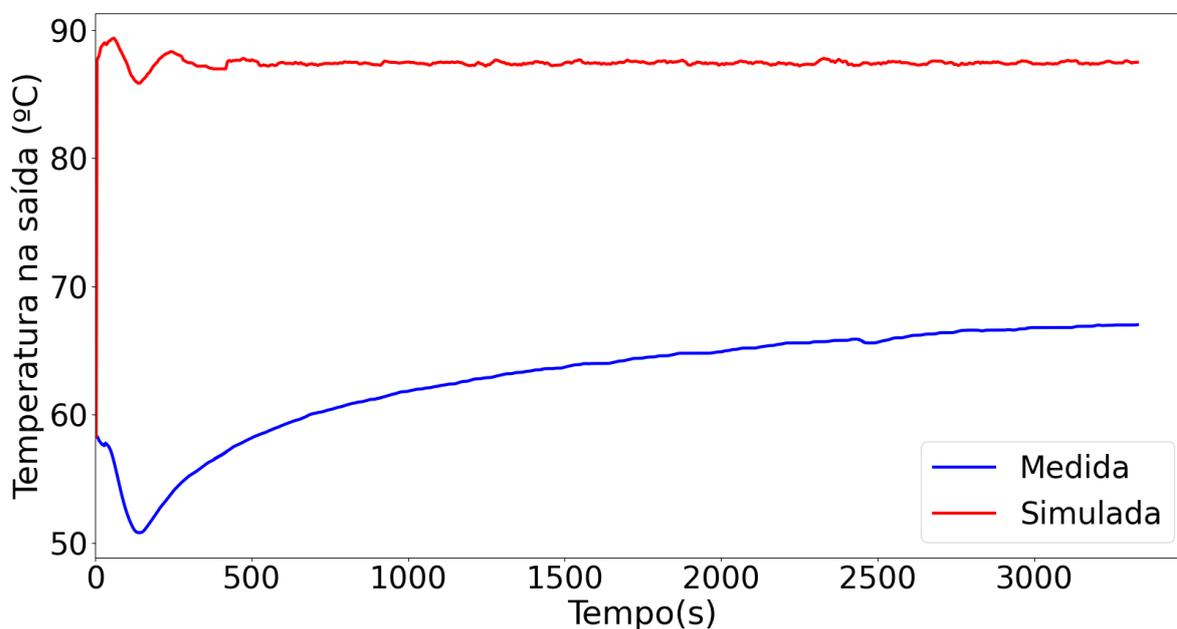
Pelos resultados referentes ao tempo de execução apresentados na seção imediatamente anterior a essa, a simulação aqui feita levou em conta o ajuste de apenas 2 parâmetros, pois isso permite o uso de uma menor tolerância na otimização, o que, em teoria, faria com que o ajuste obtido fosse melhor. Dito isso, a Figura 4.1 traz a correspondência entre as umidades absoluta medida e simulada, enquanto que a Figura 4.2 faz o mesmo para as temperaturas de saída.

Figura 4.1 – Gráfico da correspondência das umidades absolutas do gás na saída do leito de jorro medida e simulada.



Fonte: Elaborado pelo autor.

Figura 4.2 – Gráfico da correspondência das temperaturas do gás na saída do leito de jorro medida e simulada.



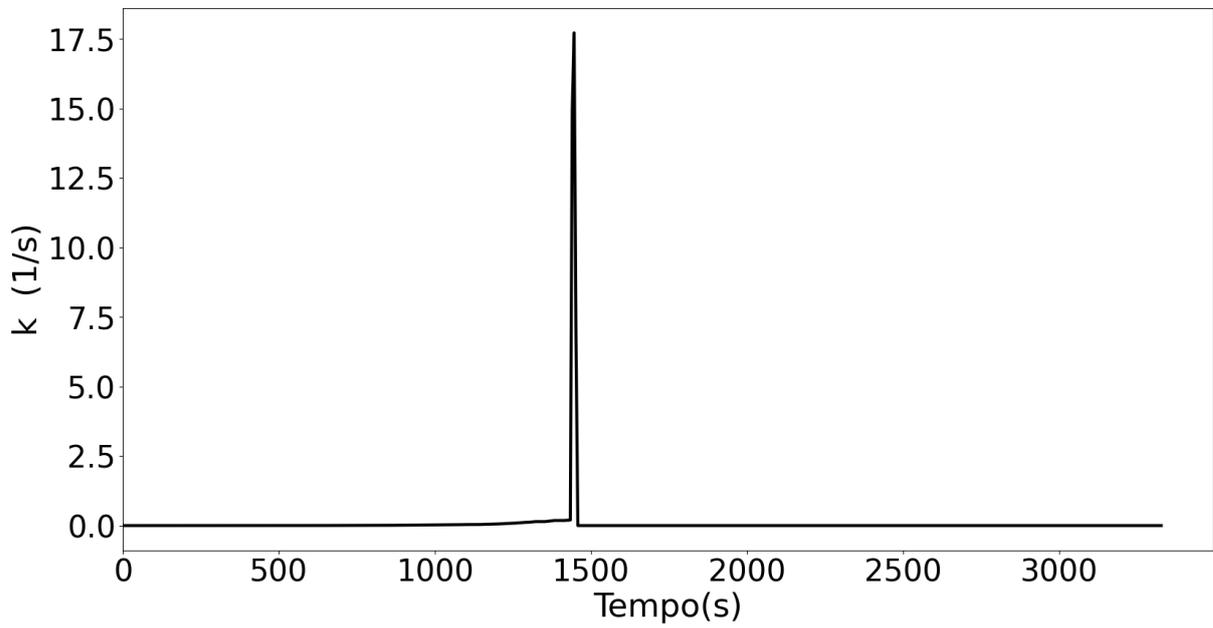
Fonte: Elaborado pelo autor.

Apenas se valendo desse critério de validação, o modelo está bastante longe de ser adequado para a realidade física do problema em estudo, uma vez que, pela Figura 4.1, observa-se uma correspondência entre as umidades medida e simulada apenas no início e ao final do tempo de simulação, enquanto a Figura 4.2 mostra perfis de temperatura completamente diferentes dos modelos simulado e medido.

Uma hipótese que pode explicar o comportamento das duas figuras é a seguinte: é sabido que o balanço de massa tem termos que dependem da temperatura, ou seja, do balanço de energia, e este último se desvia em muito do que era esperado de si já no início da simulação; mais ainda, se desvia para valores de temperatura próximas daquela da entrada do leito (um pouco menor que 90 °C). Tendo isso em vista, existe uma grande possibilidade de haver algum (ou alguns) termos programados de forma errônea na função que simula o balanço de energia, de modo que utilize da temperatura de entrada ao invés daquela na saída do leito de jorro, e gerando o desvio aqui evidenciado.

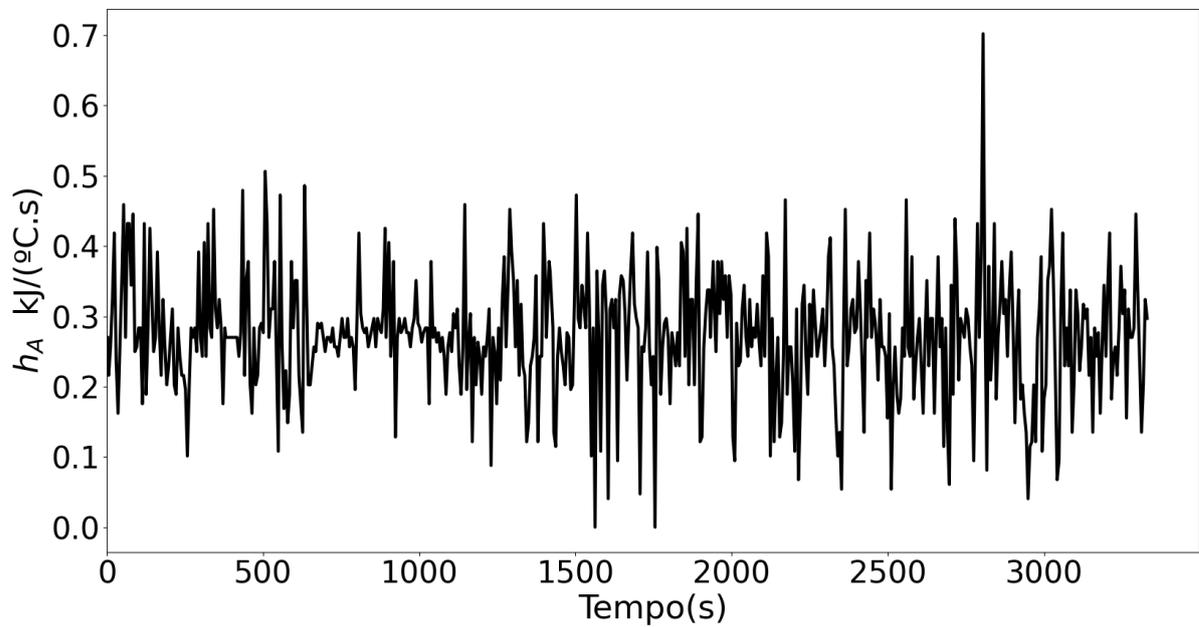
Mesmo que o modelo já tenha sido invalidado, a fim de depurar ainda mais a causa da inadequação do modelo, serão plotadas as relações dos dois parâmetros ajustados com o tempo; para o balanço de massa na Figura 4.3, e para o balanço de energia na Figura 4.4.

Figura 4.3– Gráfico da evolução da constante cinética de secagem em função do tempo.



Fonte: Elaborado pelo autor.

Figura 4.4 – Gráfico da evolução do coeficiente global de transferência de calor em função do tempo.



Fonte: Elaborado pelo autor.

Não é difícil perceber o comportamento atípico que ocorre na Figura 4.3, que apresenta um pico do ajuste paramétrico próximo a 1500 s, e depois mantém valores baixos o suficiente para não serem mostrados na plotagem (de fato, os valores esperados não deveriam ser maiores que 10^{-4}); contudo, é na Figura 4.4 que há mais uma pista para detectar a causa desse resultado completamente errôneo e inesperado, uma vez que esta mostra uma resposta completamente instável no ajuste do coeficiente global de transferência de calor; desse modo, fica mais forte a hipótese sugerida anteriormente, e, com isso, ainda mais razoável a proposição de que a causa para isso esteja em algum parâmetro passado errado na execução do balanço de energia.

Como uma hipótese adicional, nunca se pode descartar a possibilidade de que o código em MATLAB contivesse ele mesmo alguns erros; de fato, foram observadas incompatibilidades significativas com a linguagem Python, que, por falta de familiaridade com o software pago, não puderam imediatamente ser diagnosticadas como erros.

Ainda assim, falsos positivos existem na ciência, e nenhum resultado, mesmo que pareça bastante razoável e condizente com a realidade física à primeira vista, deve ser tido como incontestável, sobretudo para uma aplicação como a que foi estudada aqui, que envolve uma indentação de funções bastante complexas e altamente sensíveis a alterações de parâmetros (por exemplo, definir fronteiras equivocadas da região de otimização pode levar a um mínimo local da função objetivo que não tem correspondência com a realidade, havendo um ajuste puramente matemático).

5 – CONCLUSÕES E SUGESTÕES PARA TRABALHOS FUTUROS

Com um objetivo ambicioso, sobretudo levando em conta o tempo disponível para que fosse completado, esse trabalho buscou adaptar um modelo adaptativo a parâmetros concentrados da secagem de bagaço de laranja num leito de jorro do software pago MATLAB para o software livre Python. Mais que isso, foi feita a adaptação do paradigma de programação utilizado: ao invés de se valer de uma programação procedural, fez-se uso dos princípios da programação orientada ao objeto.

Em termos de desenvolvimento do programa computacional, pode-se considerar o que foi feito aqui como bem-sucedido, uma vez que todas suas partes funcionam em conjunto, tanto que foi permitida uma análise de tempo de execução e a plausibilidade do uso de tal algoritmo para uma estimativa de parâmetros on-line, da qual se concluiu que, através de uma simplificação razoável do modelo, considerando nula a perda de calor do leito para o ambiente, esse uso se torna plausível; contudo, nenhum caso no qual os três parâmetros estudados são variados simultaneamente pôde obter o mesmo resultado.

Mesmo que o algoritmo tenha sido executado com sucesso, a realidade física do sistema não foi validada, pois há pouca ou nenhuma correspondência entre as variáveis medidas e simuladas, além de ambos os parâmetros possuírem valores muito distantes do que era deles esperado; ainda assim, pelas respostas que foram retornadas, foi possível propor uma causa bastante pontual que, caso reparada, é capaz de tornar preciso o ajuste.

De todo modo, há um excelente ponto de partida para trabalhos futuros, já que um código bastante extenso e encapsulado com perícia razoável pôde ser desenvolvido. A maior prioridade, evidentemente, é depurar a causa precisa da divergência entre as variáveis medidas e as simuladas. Como discutido pelos melhores resultados retornados, é muito difícil que essa causa esteja em outro ponto que não o balanço de energia, de forma que seria uma boa prática refazer toda a lógica da função responsável pelo seu cálculo. Ainda no quesito de depuração, assumindo que pôde ser validado o modelo em Python, uma análise crítica do código original precisa ser feita, já que, como já citado, falsos positivos podem ocorrer, e o olhar sem viés de alguém alheio à publicação do artigo e desenvolvimento do código é de grande valia para contestar e melhorar diversos pontos deste último.

Já pensando no algoritmo atual, como o tempo de execução é um grande indicativo da sua eficácia, seria de grande valia um estudo buscando apenas otimizar esse aspecto, tendo como objetivo melhorar o encapsulamento já feito e, se necessário, aplicar outros princípios ou mesmo outros paradigmas da programação que permitam uma melhora do desempenho e diminuição da complexidade do programa. Nesse sentido, o ganho de um segundo no processamento de todo o banco de dados pode significar um grande avanço para futuras aplicações em controladores.

Ainda no âmbito computacional, uma última etapa seria o desenvolvimento de uma interface para o usuário que permita a entrada de diferentes bancos de dados e dimensões do secador, de forma que possa ser analisada a validade geral das hipóteses assumidas no modelo para outras configurações experimentais.

REFERÊNCIAS BIBLIOGRÁFICAS

- ALTZIBAR, H.; LOPEZ, G.; BILBAO, J.; OLAZAR, M. Effect of draft tube geometry on pressure drop in draft tube conical spouted beds. **Canadian Journal of Chemical Engineering**, [s. l.], v. 91, n. 11, p. 1865–1870, 2013.
- ALTZIBAR, H.; ESTIATI, I.; LÓPEZ, G.; SALDARRIAGA, J. F.; AGUADO, R.; BILBAO, J.; OLAZAR, M. Fountain confined conical spouted beds. *Powder Technology*, 312, 334–346, 2017.
- AMUTIO, M.; LOPEZ, G.; AGUADO, R.; ARTETXE, M.; BILBAO, J.; OLAZAR, M. Kinetic study of lignocellulosic biomass oxidative pyrolysis. **Fuel**, [s. l.], v. 95, p. 305–311, 2012.
- ARTEXE, M.; LÓPEZ, G.; AMUTIO, M.; BARBARIAS, I.; ARREGI, A.; AGUADO, R.; BILBAO, J.; OLAZAR, M. Styrene recovery from polystyrene by flash pyrolysis in a conical spouted bed reactor. *Waste Management*, 45, 126–133, 2015.
- BARBOSA, A. M.; ROCHA, T. A. F.; SALDARRIAGA, J. F.; ESTIATI, I.; FREIRE, F. B.; FREIRE, J. T. Alternative drying of orange bagasse in vibrofluidized bed for use in combustion. **Chemical Engineering and Processing - Process Intensification**, [s. l.], v. 152, n. May, p. 107941, 2020.
- BRITO, R. C.; BÉTTEGA, R.; FREIRE, J. T. Energy analysis of intermittent drying in the spouted bed. **Drying Technology**, [s. l.], v. 37, n. 12, p. 1498–1510, 2018.
- EDGAR, T. F.; HIMMELBLAU, D. M.; LASDON, L. S. *Optimization of Chemical Processes*. 2. ed. McGraw-Hill Science, Engineering & Mathematics, 2001.
- EPSTEIN, N.; GRACE, J. R. *Spouted and Spout-Fluid Beds*. [s.l.] Cambridge University Press, 2010.
- ERDOGAN, E.; ATILA, B.; MUMME, J.; REZA, M. T.; TOPTAS, A.; ELIBOL, M.; YANIK, J. Characterization of products from hydrothermal carbonization of orange pomace including anaerobic digestibility of process liquor. **Bioresource Technology**, [s. l.], v. 196, p. 35–42, 2015.
- EYKHOFF, P. *System Identification*. Wiley-Interscience, Nova York, 1974.
- FAO. *Citrus Fruit Statistical Compendium 2020*. Rome, 2021.
- FREIRE, Fábio B.; ATXUTEGI, A.; FREIRE, Flavio B.; FREIRE, J. T.; AGUADO, R.; OLAZAR, M. An adaptive lumped parameter cascade model for orange juice solid waste drying in spouted bed. **Drying Technology**, [s. l.], v. 35, n. 5, p. 577–584, 2017.
- GOODRICH, M. T.; TAMASSIA, R.; GOLDWASSER, M. H. *Data Structures and Algorithms in Python*. John Wiley & Sons, 2013.
- GUZMAN, G.; XIAO, D.; LISKA, D. A.; MAH, E.; SANOSHY, K.; MANTILLA, L.;

REPLOGLE, R.; BOILEAU, T. W.; BURTON-FREEMAN, B. M.; EDIRISINGHE, I. Addition of Orange Pomace Attenuates the Acute Glycemic Response to Orange Juice in Healthy Adults. **Journal of Nutrition**, [s. l.], v. 151, n. 6, p. 1436–1442, 2021.

HAMAWAND, I.; DA SILVA, W. P.; EBERHARD, F.; ANTILLE, D. L. Issues related to waste sewage sludge drying under superheated steam. **Polish Journal of Chemical Technology**, [s. l.], v. 17, n. 4, p. 5–14, 2015.

HUANG, Y. L.; MA, Y. S. The effect of extrusion processing on the physiochemical properties of extruded orange pomace. **Food Chemistry**, [s. l.], v. 192, p. 363–369, 2016.

LOPÉZ, G.; AMUTIO, M.; ELORDI, G.; ARTEXE, M.; ALTZIBAR, H.; OLAZAR, M. A CONICAL SPOUTED BED REACTOR FOR THE VALORISATION OF WASTE TIRES. ECI Digital Archives, 2010.

LOPÉZ, G.; ERKIAGA, A.; AMUTIO, M.; BILBAO, J.; OLAZAR, M. Effect of polyethylene co-feeding in the steam gasification of biomass in a conical spouted bed reactor. *Fuel*, 153, 393–401, 2015.

MADEIRA, J. V.; MACEDO, J. A.; MACEDO, G. A. A new process for simultaneous production of tannase and phytase by *Paecilomyces variotii* in solid-state fermentation of orange pomace. **Bioprocess and Biosystems Engineering**, [s. l.], v. 35, n. 3, p. 477–482, 2012.

MATHUR, K. B.; EPSTEIN, N. **Spouted Beds**. [S. l.: s. n.], 1974.

MORAES, J. C.; CRESTANI, C. E. Caracterização Química do Bagaço de Laranja. In: XXII CONGRESSO BRASILEIRO DE ENGENHARIA QUÍMICA, 2018, [s. l.], . **Anais [...]**. [S. l.: s. n.], 2018. p. 257–260.

MUJUMDAR, A. S. (org.). **Handbook of Industrial Drying**. Third Edited. [S. l.]: Taylor & Francis, 1988.v. 6

NARIMATSU, C. P. Contribuições para o estudo da secagem em leito pneumático vertical. Tese (Doutorado). Universidade Federal de São Carlos, 2004. São Carlos, São Paulo, Brasil.

OLAZAR, M.; LOPEZ, G.; ALTZIBAR, H.; AMUTIO, M.; BILBAO, J. Drying of Biomass in a Conical Spouted Bed with Different Types of Internal Devices. **Drying Technology**, [s. l.], v. 30, n. 2, p. 207–216, 2012.

OLAZAR, M.; AGUADO, R.; JOSÉ, M.; ÁLVAREZ, S.; BILBAO, J. Minimum spouting

velocity for the pyrolysis of scrap tyres with sand in conical spouted beds. *Powder Technology*, 165(3), 128–132, 2006.

OTAKE, S.; IZUTSU, K.; LECHURA-BALLESTEROS, D. (org.). **Drying Technologies for Biotechnology and Pharmaceutical Applications**. 1st. ed. [*S. l.*]: Wiley-VCH Verlag GmbH & Co. KGaA, 2020. 2020.

PERAZZINI, H.; BENTES FREIRE, F.; FREIRE, F. B.; FREIRE, J. T. Thermal treatment of solid wastes using drying technologies: a review, *Drying Technology*, v. 34, p. 39-52, 2016.

PIERCE, B. C. *Types and Programming Languages*. MIT Press, 2002.

QIN, H.; YANG, K.; SUN, B.; WANG, Q.; XU, C.; QI, Y. Experimental Study on Spouted Bed Hydrodynamics for Oil Shale Semi-Coke. **Energy Procedia**, [*s. l.*], v. 17, p. 1740–1746, 2012.

RUGGIERO, M. A. G.; DA ROCHA LOPES, V. L. *Cálculo numérico: aspectos teóricos e computacionais*. São Paulo: Pearson, 1996.

SAHIN, S.; SUMNU, G.; TUNABOYU, F. Usage of solar-assisted spouted bed drier in drying of pea. **Food and Bioproducts Processing**, [*s. l.*], v. 91, n. 3, p. 271–278, 2013.

SHAHRAM, H.; TAGHIAN DINANI, S. Influences of electrohydrodynamic time and voltage on extraction of phenolic compounds from orange pomace. **Lwt**, [*s. l.*], v. 111, n. July 2018, p. 23–30, 2019.

SOUSA, R. C.; FERREIRA, M. C.; ALTZIBAR, H.; FREIRE, F. B.; FREIRE, J. T. Drying of pasty and granular materials in mechanically and conventional spouted beds. **Particuology**, [*s. l.*], v. 42, p. 176–183, 2020.

SPREUTELS, L.; CHAOUKI, J.; BERTRAND, F.; HAUT, B.; LEGROS, R. Gas residence time distribution in a conical spouted bed. **Powder Technology**, [*s. l.*], v. 290, p. 62–71, 2016.

SPREUTELS, L.; HAUT, B.; CHAOUKI, J.; BERTRAND, F.; LEGROS, R. Conical spouted bed drying of Baker's yeast: Experimentation and multi-modeling. **Food Research International**, [*s. l.*], v. 62, p. 137–150, 2014.

TETRAPAK. Fruit processing. *In*: THE ORANGE BOOK. [*S. l.: s. n.*], 2017. p. 198. Disponível em: <https://orangebook.tetrapak.com/chapter/fruit-processing>. Acesso em: 7 janeiro 2024.

TOLKIEN, J.R.R The Fellowship of the Ring: Being the First Part of The Lord of the Rings. 1954.

UCUNCU, C.; TARI, C.; DEMIR, H.; BUYUKKILECI, A. O.; OZEN, B. Dilute-acid hydrolysis of apple, orange, apricot and peach pomaces as potential candidates for bioethanol production. **Journal of Biobased Materials and Bioenergy**, [s. l.], v. 7, n. 3, p. 376–389, 2013.

YUE, Y.; WANG, S.; BAHL, P.; DE SILVA, C.; SHEN, Y. Experimental investigation of spout incoherence in a spouted bed. **Chemical Engineering Journal**, [s. l.], v. 418, n. March, p. 129320, 2021.

APÊNDICE A – CÓDIGO NA ÍNTEGRA

APÊNDICE A.1 MÓDULO DATAPROCESSING

```
from numpy import ones, array
from pandas import DataFrame
from scipy.io import loadmat
from scipy.signal import lfilter, firwin

class DataProcessing:
    def __init__(self, file, filter_params):
        self.file = file
        self.filter_params = filter_params

    @staticmethod
    def ctok(temp):
        tconv = temp + 273.15

        return tconv

    @staticmethod
    def transform_to_array(vector):
        ar = array(vector)

        return ar

    @staticmethod
    def load_data(file):
        extract = loadmat(file)
```

```
data = extract['data']

return data

@staticmethod
def pv_transform(tempc):
    pv = 6.004918 * 10 ** (7.337936 * tempc / (tempc + 229.3975))

    return pv

@staticmethod
def ph2_transform(pv, hr):
    ph2 = pv * hr / 100

    return ph2

@staticmethod
def y_transform(ph2, tempk):
    y = 2.16679 * ph2 * 100 / tempk

    return y

@staticmethod
def filter(filter_coeffs, a, arr):
    filtered_array = lfilter(filter_coeffs, a, arr)

    return filtered_array

def make_dataframe(self):
    data = self.load_data(self.file)
```

```

header = 'Time(s) Flow-rate(m³/min) Tin Tout Tbed T_filter Relative_moisture
Pressure_difference T_blower ' \
        'Total_pressure'
header = header.split()

dataframe = DataFrame(data, columns=header)

return dataframe

def filter_data(self, dataframe):

    filter_params = self.filter_params

    tin = dataframe['Tin'].values
    tout = dataframe['Tout'].values
    flowrate = dataframe['Flow-rate(m³/min)'].values
    hrout = dataframe['Relative_moisture'].values

    num_taps = filter_params[0]
    cutoff_freq = filter_params[1]
    a = filter_params[2]

    filter_coefficients = firwin(num_taps, cutoff_freq)

    t_in = self.filter(filter_coefficients, a, tin)

    t_out = self.filter(filter_coefficients, a, tout)

    flow_rate = self.filter(filter_coefficients, a, flowrate)

```

```
hr_out = self.filter(filter_coefficients, a, hrout)
```

```
border_effect = num_taps + 1
```

```
t_in[0:border_effect] = tin[0:border_effect]
```

```
t_out[0:border_effect] = tout[0:border_effect]
```

```
flow_rate[0:border_effect] = flowrate[0:border_effect]
```

```
return t_in, t_out, flow_rate, hr_out
```

```
def transform_data(self):
```

```
    dataframe = self.make_dataframe()
```

```
    tin, tout, flowrate, hr_out = self.filter_data(dataframe)
```

```
    t_inc = self.transform_to_array(tin)
```

```
    t_ink = self.ctok(t_inc)
```

```
    t_outc = self.transform_to_array(tout)
```

```
    t_outk = self.ctok(t_outc)
```

```
    flowrate = self.transform_to_array(flowrate)
```

```
    hr_out = self.transform_to_array(hr_out)
```

```
    hr_out = hr_out / 100
```

```
    length = len(hr_out)
```

```
    hr_in = ones(length)
```

```
    reference = hr_out[-1]
```

```
hr_in = .45 * reference * hr_in
```

```
pv_in = self.pv_transform(t_inc)
```

```
pv_out = self.pv_transform(t_outc)
```

```
ph2_in = self.ph2_transform(pv_in, hr_in)
```

```
ph2_out = self.ph2_transform(pv_out, hr_out)
```

```
y_in = self.y_transform(ph2_in, t_ink)
```

```
y_out = self.y_transform(ph2_out, t_outk)
```

```
time = dataframe['Time(s)'].values
```

```
return t_inc, t_outc, t_ink, t_outk, flowrate, hr_in, hr_out, pv_in, pv_out, ph2_in,
ph2_out, y_in, y_out, time
```

APÊNDICE A.2 MÓDULO MODEL

```
from numpy import pi, exp, log, inf, ndarray
```

```
class DryingModel:
```

```
def __init__(self, dimensions, freewalk, model_parameters, equilibrium_parameters,
t_sample, key_variables):
```

```
self.dimensions = dimensions
```

```
self.k = freewalk[0]
```

```
self.ha = freewalk[1]
```

```
self.qloss = freewalk[2]
```

```

self.t_inc = model_parameters[0]
self.t_ink = model_parameters[1]
self.t_outk = model_parameters[2]
self.flowrate = model_parameters[3]
self.pv_out = model_parameters[4]
self.y_in = model_parameters[5]
self.equi = equilibrium_parameters
self.t_sample = t_sample
self.X_solid = key_variables[0]
self.Y_gas = key_variables[1]
self.T_solid = key_variables[2]
self.T_gas = key_variables[3]

```

```

def set_k(self, k):

```

```

    self.k = k[0] if isinstance(k, ndarray) else k

```

```

def set_ha(self, ha):

```

```

    self.ha = ha[0] if isinstance(ha, ndarray) else ha

```

```

def set_qloss(self, qloss):

```

```

    self.qloss = qloss[0] if isinstance(qloss, ndarray) else qloss

```

```

@staticmethod

```

```

def conetrunk_volume(height, d_1, d_2):

```

```

    v = pi / 12 * (d_1 ** 2 + d_2 ** 2 + d_1 * d_2)

```

```

    return v

```

```

@staticmethod

```

```
def cylinder_volume(height, diameter):
```

```
    v = pi / 4 * diameter ** 2 * height
```

```
    return v
```

```
@staticmethod
```

```
def gas_density(tempk):
```

```
    rho = 2.2538 - 0.003588 * tempk
```

```
    return rho
```

```
@staticmethod
```

```
def ctok(temp):
```

```
    tconv = temp + 273.15
```

```
    return tconv
```

```
@staticmethod
```

```
def pv_transform(tempc):
```

```
    pv = 6.004918 * 10 ** (7.337936 * tempc / (tempc + 229.3975))
```

```
    return pv
```

```
@staticmethod
```

```
def cp_h2o(mw_h2o):
```

```
cp = 75.42 / mw_h2o
```

```
return cp
```

```
@staticmethod
```

```
def ms():
```

```
ms = 0.5 / (1 + 3.91)
```

```
return ms
```

```
@staticmethod
```

```
def rh(y_gas, tempc, pv):
```

```
rh = y_gas * tempc / pv / 2.16679
```

```
return rh
```

```
def x_equilibrium(self, tempc, rh):
```

```
a, b, c = self.equi[0], self.equi[1], self.equi[2]
```

```
xeq = (-exp(a * tempc + c) / log(rh)) ** (1 / b)
```

```
return xeq
```

```
@staticmethod
```

```
def cp_solid(tempc):
```

```
cp = .71939 + .00638 * tempc
```

```
    return cp

    @staticmethod
    def cp_gases(params, tempc, mw):

        a, b, c, d = params[0], params[1], params[2], params[3]

        cp = 1 / mw * (a + b * 10 ** -2 * tempc + c * 10 ** -5 * tempc ** 2 + d * 10 ** -9 *
tempc ** 3)

        return cp

    def calculate_volumes(self):

        cylinder_height, cylinder_diameter, outlet_height, outlet_diameter, cone_height,
cone_diameter = self.dimensions

        v_outlet = self.conetrunk_volume(outlet_height, outlet_diameter, cylinder_diameter)

        v_cylinder = self.cylinder_volume(cylinder_height, cylinder_diameter)

        v_cone = self.conetrunk_volume(cone_height, cone_diameter, cylinder_diameter)

        v_rest = v_cylinder + v_outlet

        v_total = v_rest + v_cone

        return v_total

    def mass_balance(self, t, ym):
```

```

i = round(t / self.t_sample)

k, volume, x_solid, y_gas, ms = self.k, self.calculate_volumes(), ym[0], ym[1], self.ms()

t_gas, t_solid = self.T_gas, self.T_solid

rh = self.rh(y_gas, t_gas, self.pv_transform(t_gas))

x_equi = self.x_equilibrium(t_gas, rh)

rho_gas, rho_in, rho_out = (self.gas_density(self.ctok(t_gas)),
self.gas_density(self.t_ink[i]),
self.gas_density(self.ctok(t_gas)))

gas_mass, rw = rho_gas * volume, k * (x_solid - x_equi)

dxdt = - rw

dydt = 1 / gas_mass * (self.flowrate[i] * (rho_in * self.y_in[i] - rho_out * y_gas) + ms *
rw / volume)

return dxdt, dydt

def energy_balance(self, t, yt):

i = round(t / self.t_sample)

t_solid = yt[0]

t_gas = yt[1]

```

```
k, ha, qloss, x_solid, lambdda, mw_h2o, mw_gas, ms, volume = (self.k, self.ha,
self.qloss, self.X_solid, 2257,
```

```
18, 29, self.ms(), self.calculate_volumes())
```

```
cp_h2o, pvo = self.cp_h2o(mw_h2o), self.pv_transform(t_gas)
```

```
rh = self.rh(self.Y_gas, t_gas, pvo)
```

```
xeq = self.x_equilibrium(t_gas, rh)
```

```
rho_gas, rho_input = self.gas_density(self.ctok(t_gas)), self.gas_density(self.t_ink[i])
```

```
gas_mass, rw = rho_gas * volume, k * (x_solid - xeq)
```

```
cp_solid, cp_gas, cp_vapor = (self.cp_solid(t_solid),
self.cp_gases([28.94, .4147, .3191, -1.965], t_gas, mw_gas),
self.cp_gases([33.46, .6880, .7604, -3.593], t_gas, mw_h2o))
```

```
dstdt = (1 / (cp_solid + (75.4 * self.X_solid / mw_h2o)) * (ha * (t_gas - t_solid) / ms - rw
*
```

```
(cp_vapor * (t_gas - t_solid) + lambdda)))
```

```
dgtdt = (- self.flowrate[i] / gas_mass * (rho_gas * t_gas - rho_input * self.t_inc[i]) +
1 / (cp_gas + volume / gas_mass * (1.81 * self.Y_gas + self.y_in[i] * cp_vapor))
* (- ha * (t_gas - t_solid) / gas_mass + rw * ms / gas_mass * cp_vapor * t_gas -
qloss))
```

```
return dstdt, dgtdt
```

APÊNDICEA.3 MÓDULO SIMULATION

```
from scipy.integrate import solve_ivp
```

```
class Simulation:
```

```
    def __init__(self, y0m, y0t, model):
```

```
        self.y0m = y0m
```

```
        self.y0t = y0t
```

```
        self.model = model
```

```
    def set_y0m(self, y0m):
```

```
        self.y0m = y0m
```

```
    def set_y0t(self, y0t):
```

```
        self.y0t = y0t
```

```
    def integrate_mass(self, time_spam):
```

```
        y0m = self.y0m
```

```
        results = solve_ivp(fun=self.model.mass_balance, t_span=time_spam, y0=y0m)
```

```
        simulated_y = [results.y[1][0], results.y[1][-1]]
```

```
        simulated_x = [results.y[0][0], results.y[0][-1]]
```

```
        return simulated_y, simulated_x
```

```
    def integrate_energy(self, time_spam):
```

```
        y0t = self.y0t
```

```
        results = solve_ivp(fun=self.model.energy_balance, t_span=time_spam, y0=y0t)
```

```
        simulated_tgas = [results.y[1][0], results.y[1][-1]]
```

```
simulated_tsolid = [results.y[0][0], results.y[0][-1]]
```

```
return simulated_tgas, simulated_tsolid
```

```
@staticmethod
```

```
def error(measured_value, estimated_value):
```

```
    first_term = ((estimated_value[0] - measured_value[0]) / (measured_value[0])) ** 2
```

```
    second_term = ((estimated_value[1] - measured_value[1]) / (measured_value[1])) ** 2
```

```
    error = first_term + second_term
```

```
    return error
```

APÊNDICE A.4 MÓDULO CASCADEMODEL

```
# Imports
```

```
from scipy.optimize import minimize, Bounds
```

```
from DataProcessing import DataProcessing
```

```
from Model import DryingModel
```

```
from Simulation import Simulation
```

```
from time import time
```

```
from matplotlib.pyplot import plot, savefig
```

```
from pandas import DataFrame
```

```
from scipy.signal import medfilt
```

```
def main():
```

```
    file = 'data.mat'
```

```
    filter_params = [10, .005, 1]
```

```
thing = DataProcessing(file, filter_params)
dataframe = thing.make_dataframe()

transformed_data = thing.transform_data()
t_inc, t_outc, t_ink, t_outk, flowrate, hr_in, hr_out, pv_in, pv_out, ph2_in, ph2_out, y_in,
y_out, time \
    = transformed_data

cylinder_height = .335 + .22 + .05
cylinder_diameter = .36

outlet_height = .28
outlet_diameter = .05

cone_height = .5
cone_diameter = .068

dimensions = [cylinder_height, cylinder_diameter, outlet_height, outlet_diameter,
cone_height, cone_diameter]

k = 0.000048
ha = 0.27
qloss = 5.73

freewalk = [k, ha, qloss]

model_parameters = [t_inc, t_ink, t_outk, flowrate, pv_out, y_in]

equilibrium_parameters = [-0.2428, 2.2514, 10.0012]

t_sample = 3
```

```
x_solid, y_gas, t_solid, t_gas = 3.91, y_out[0], 25, t_outc[1]
```

```
key_variables = [x_solid, y_gas, t_solid, t_gas]
```

```
model = DryingModel(dimensions, freewalk, model_parameters, equilibrium_parameters,  
t_sample, key_variables)
```

```
time_length = len(time)
```

```
time_mass = [0]
```

```
time_energy = [3]
```

```
optimized_ks = [model.k]
```

```
optimized_ha = [model.ha]
```

```
optimized_qlosses = [model.qloss]
```

```
Xs = [3.91]
```

```
Yg = [y_out[0]]
```

```
Ts = [25]
```

```
Tg = [t_outc[1]]
```

```
y0m = [3.91, y_out[0]]
```

```
y0t = [25, t_outc[1]]
```

```
simulation = Simulation(y0m, y0t, model)
```

```
filter_length = 15
```

```
y_out = medfilt(y_out, filter_length)
```

```
t_outc = medfilt(t_outc, filter_length)
```

```
def objective_function_mass(k, time_span, simulation, y):
```

```
    model.set_k(k)
```

```
total_error = 0

simulated_y, simulated_x = simulation.integrate_mass(time_span)

error = simulation.error(y, simulated_y)

total_error += error

return total_error

def objective_function_energy(params, time_span, simulation, tgas):
    ha, qloss = params
    model.set_ha(ha)
    model.set_qloss(qloss)

    total_error = 0

    simulated_tgas, simulated_tsolid = simulation.integrate_energy(time_span)

    error = simulation.error(tgas, simulated_tgas)

    total_error += error

    return total_error

initial_k_guess = model.k
initial_param_guess = [model.ha, model.qloss]

for i in range(0, time_length - 1, 2):
```

```
time_span = (time[i], time[i + 1])
time_mass.append(time_span[-1])
```

```
measured_y = [y_out[i], y_out[i + 1]]
```

```
bounds_mass = Bounds(lb=1e-7)
```

```
result_mass = minimize(objective_function_mass, initial_k_guess, method='Nelder-
Mead',
```

```
2)          args=(time_span, simulation, measured_y), bounds=bounds_mass, tol=1e-
```

```
optimized_k = result_mass.x[0]
```

```
model.set_k(optimized_k)
```

```
initial_k_guess = optimized_k
```

```
new_y, new_x = simulation.integrate_mass(time_span)
```

```
model.X_solid = new_x[-1]
```

```
model.Y_gas = new_y[-1]
```

```
simulation.y0m = [new_x[-1], new_y[-1]]
```

```
optimized_ks.append(optimized_k)
```

```
Yg.append(new_y[-1])
```

```
Xs.append(new_x[-1])
```

```
j = i + 1
```

```
time_span = (time[j], time[j + 1])
```

```
time_energy.append(time_span[-1])
```

```
measured_tgas = [t_outc[j], t_outc[j + 1]]
```

```

bounds_energy = ((0, None), (5, None))

result_energy = minimize(objective_function_energy, initial_param_guess,
method='Nelder-Mead',
                        args=(time_span, simulation, measured_tgas), bounds=bounds_energy,
tol=1e-2)

optimized_ha, optimized_qloss = result_energy.x[0], result_energy.x[1]
model.set_ha(optimized_ha)
model.set_qloss(optimized_qloss)
new_tgas, new_tsolid = simulation.integrate_energy(time_span)
model.T_solid = new_tsolid[-1]
model.T_gas = new_tgas[-1]
simulation.y0t = [new_tsolid[-1], new_tgas[-1]]

optimized_has.append(optimized_ha)
optimized_qlosses.append(optimized_qloss)

Tg.append(new_tgas[-1])
Ts.append(new_tsolid[-1])

fig1 = plot(time_mass, optimized_ks)
savefig('1.png')
fig2 = plot(time_energy, optimized_has)
savefig('2.png')
fig3 = plot(time_energy, optimized_qlosses)

optim_dict_mass = {'time': time_mass, 'k': optimized_ks, 'Yg': Yg, 'Xs': Xs}
optim_dict_energy = {"time": time_energy, "ha": optimized_has, "qloss": optimized_qloss,
"Tg": Tg, "Ts": Ts}

```

```
df_mass = DataFrame(optim_dict_mass)
df_energy = DataFrame(optim_dict_energy)

df_mass.to_excel("optimized_parameters_mass.xlsx", sheet_name="Mass")
df_energy.to_excel("optimized_parameters_energy.xlsx", sheet_name="Energy")

if __name__ == '__main__':
    start = time()
    main()
    end = time()

    print("Elapsed time:\n{:.0f} s".format(end - start))
```