Downloaded from orbit.dtu.dk on: Apr 28, 2024

DTU Library

Hybrid machine learning assisted modelling framework for particle processes

Nielsen, Rasmus Fjordbak; Nazemzadeh, Nima; Sillesen, Laura Wind; Andersson, Martin Peter; Gernaey, Krist V.; Mansouri, Seyed Soheil

Published in:
Computers & Chemical Engineering

Link to article, DOI:
10.1016/j.compchemeng.2020.106916

Publication date:
2020

Document Version
Peer reviewed version

Link back to DTU Orbit

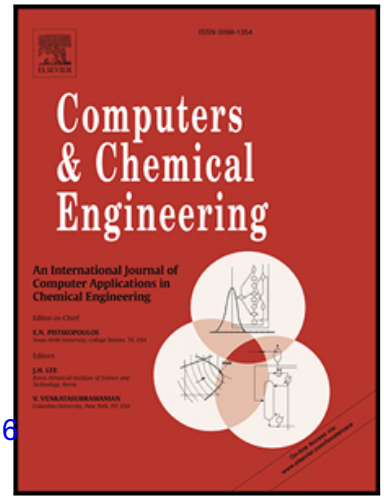
Citation (APA):
Nielsen, R. F., Nazemzadeh, N., Sillesen, L. W., Andersson, M. P., Gernaey, K. V., & Mansouri, S. S. (2020). Hybrid machine learning assisted modelling framework for particle processes. *Computers & Chemical Engineering*, *140*, Article 106916. https://doi.org/10.1016/j.compchemeng.2020.106916

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Journal Pre-proof



Hybrid machine learning assisted modelling framework for particle processes

Rasmus Fjordbak Nielsen, Nima Nazemzadeh, Laura Wind Sillesen, Martin Peter Andersson, Krist V. Gernaey, Seyed Soheil Mansouri

PII: S0098-1354(19)31082-8
DOI: https://doi.org/10.1016/j.compchemeng.2020.106916
Reference: CACE 106916

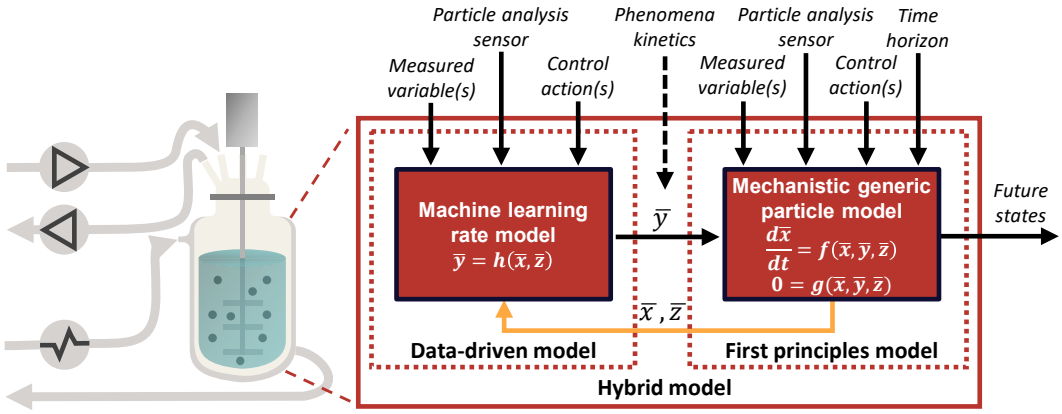
To appear in: *Computers and Chemical Engineering*

Received date: 15 October 2019
Revised date: 8 May 2020
Accepted date: 9 May 2020

Please cite this article as: Rasmus Fjordbak Nielsen, Nima Nazemzadeh, Laura Wind Sillesen, Martin Peter Andersson, Krist V. Gernaey, Seyed Soheil Mansouri, Hybrid machine learning assisted modelling framework for particle processes, *Computers and Chemical Engineering* (2020), doi: https://doi.org/10.1016/j.compchemeng.2020.106916

This is a PDF file of an article that has undergone enhancements after acceptance, such as the addition of a cover page and metadata, and formatting for readability, but it is not yet the definitive version of record. This version will undergo additional copyediting, typesetting and review before it is published in its final form, but we are providing this version to give early visibility of the article. Please note that, during the production process, errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

© 2020 Published by Elsevier Ltd.



Highlights

- A framework for hybrid modelling of particle processes has been developed
- A machine learning based soft-sensor is generated for estimation of particle phenomena kinetics
- The framework requires only limited prior process knowledge
- The framework has been applied and evaluated in both small and large scale case studies
- The framework has been implemented using automatic differentiation to speed up model training

Journal Pre-proof

Journal Pre-proof

Hybrid machine learning assisted modelling framework for particle processes

Rasmus Fjordbak Nielsen[a], Nima Nazemzadeh[a], Laura Wind Sillesen[a], Martin Peter Andersson[b], Krist V. Gernaey[a], Seyed Soheil Mansouri[a,*]

[a] *Process and Systems Engineering Centre (PROSYS), Department of Chemical and Biochemical Engineering, Technical University of Denmark, Søltofts Plads, Building 228a, DK-2800 Kgs. Lyngby, Denmark*
[b] *CHEC Research Centre, Department of Chemical and Biochemical Engineering, Technical University of Denmark, Søltofts Plads, Building 228a, DK-2800 Kgs. Lyngby, Denmark*

Abstract

Particle processes are used broadly in industry and are frequently used for removal of insolubles, product isolation, purification and polishing. These processes are challenging to control due to their complex dynamics and physical-chemical properties. With the developments in particle monitoring tools make it possible to gain real-time insights into some of these process dynamics. In this work, a systematic modelling framework is proposed for particle processes based on a hybrid modelling concept, which integrates first-principles with machine-learning approaches. Here, we utilize on-line/at-line sensor data to train a machine learning based soft-sensor that predicts particle phenomena kinetics by combining it with a mechanistic population balance model. This approach allows flexibility towards use of process sensors and the model predictions do not violate physical constraints. Application of the framework is demonstrated through a laboratory-scale lactose crystallization, a laboratory-scale flocculation, and an industrial-scale pharmaceutical crystallization, using only limited prior process knowledge.

Keywords: Hybrid modelling, Modelling framework, Machine learning based soft-sensor, Real-time training
PACS: 07.05.Mh, 07.05.Tp

1. Introduction

Particles play a key role in many industrial productions, covering products such as minerals, coatings, chemicals, food and pharmaceuticals. Particle processes are typically used for removal of particles, product isolation, product purification and/or product polishing. Some examples of these processes are precipitation, crystallization, flocculation and emulsification. In general, for all of these processes, the particle properties are critical attributes in terms of the final product quality and/or the process efficiency. Depending on the specific product and process, the critical particle attributes may vary. However, the ideal particle attributes are typically an established specification from an early stage in the product/process development.

However, it is not straightforward to operate these types of processes with low process variations in the presence of disturbances. In many instances, the process kinetics are known to be highly non-linear and multi-variable, causing fluctuations in product properties and quality. These variations may in some processes result in the requirement for re-processing or even disposal of product. Thus, to transition into a

*Corresponding author (seso@kt.dtu.dk)

Journal Pre-proof

more sustainable production and consumption, as adressed in the UN sustainable development goals [1], it is necessary to develop tools that can address these challenges.

A key step in this development is the ability to monitor particle processes. Various techniques have been applied throughout history, starting with sieving analysis and optical microscopy in the early 18th century [2], light scattering techniques in the second half of the 19th century [3, 4] and laser reflectance in the late 20th century [5]. Especially the introduction of laser reflectance as a particle analysis technology has paved the road for real-time measurements of particle properties. However, this technique is indirect. Thereby, it is prone to loose information on particle dimensions and morphology in analysis of non-spherical particles. To accommodate this, there have been significant developments within direct measurement techniques of particles over the last 20 years, where dynamic image analysis has been a major focus.

Today, it is possible to analyse larger populations of particles using real-time particle imaging. The images are typically obtained using a confocal microscope combined with a high-speed camera and afterwards processed using a segmentation algorithm that identifies each particle. Then finally, for each particle, a number of particle property features are extracted [6].

Several dynamic image analysis sensor solutions have been suggested through the last two decades, where a lot of them are also commercially available. In the field of on-line/in-line particle image analysis of particles in liquid suspension, both probe based (Mettler Toledo PVM [7]) and flow-cell based methods (Sympatec QicPic [8], ParticleTech solution [9]) have been suggested.

The availability of real-time particle analysis has a great potential in many parts of the development and operation of particle processes, and especially in the challenging task of characterizing particle process dynamics. Scientists coming from various fields of particle processing have throughout the years carried out countless studies on particle kinetics, using medium to low-frequency particle analysis. A large fraction of these studies have relied on various empirical and semi-empirical mathematical expressions to describe the nature of the kinetics. The empirical expressions have been kept simple to accommodate the low frequency of particle analysis data and thereby reducing the risk of model over-fitting. With the opportunity of obtaining higher frequency particle analysis data, it is now feasible to use more complex expressions without risking over-fitting the kinetic models.

This has also recently caught attention in other fields of engineering, resulting in an increased usage of data-driven modelling approaches, also denoted machine-learning approaches. This includes both purely data-driven approaches and hybrid approaches [10, 11]. Especially the concept of hybrid modelling, where mechanistic and machine learning models are combined into one model, has been demonstrated with great success in various applications. For instance in modelling and optimization of penicillin production [12] and in modelling of heat transfer in fixed-bed reactors [13] with many more reviewed by Glassey et al. [14]. This type of modelling has recently been pointed out to be one of the most interesting approaches to bring artificial intelligence into the field of chemical and biochemical engineering [15]. This is due to a number of advantages over purely data-driven methods, including increased robustness towards measurement

Journal Pre-proof

noise/outliers by ensuring that the model does not violate physical/chemical constraints. Furthermore, it allows for utilizing prior knowledge to the extent it is available, whereas the unknowns can be derived using mathematically flexible machine learning models.

Hybrid modelling approaches have previously been applied to a number of different particle systems, including crystallization [16, 17, 18] and milling/grinding processes [19]. A selection of these studies are listed in Table 1.

Table 1: A selection of studies on hybrid modelling of particle processes. Bold text indicates the model output that has been fitted to experimental data. Metod of moments is abbreviated as MoM, Population balance model is abbreviated as PBM

Work	System	Model	Phenomena	Model output
Lauret et al. [16] (2001)	Crystallization	Mass balance	Growth (size indep.)	**Total crystal mass**
Galvanauskas et al. [17] (2006)	Crystallization	MoM PBM, Mass balance	Nucleation, Growth (size indep.), Agglomeration (size indep.)	**Total crystal mass,** Average crystal size (mass), Coefficient of variation (%)
Akkisetty et al. [19] (2010)	Milling	Discr. PBM (four size bins)	Breakage (size dep.)	**Particle size distribution**
Meng et al. [18] (2019)	Crystallization	MoM PBM, Mass balance, Energy balance	Nucleation Growth (size indep.) Agglomeration (size indep.)	**Crystal mass** Average crystal size (mass), Coefficient of variation (%)

In 2001, Lauret et al. [16] proposed a hybrid mass balance model for a crystallization process. They used a neural network to estimate a size independent growth rate for predicting the total crystal mass. To estimate the growth rate they used an off-line measured crystal mass in their study.

In 2006, Galvanauskas et al. [17] proposed a hybrid population and mass balance model for a crystallization process. Here they showed that their approach could outperform a conventional mechanistic reference model. Galvanauskas et al. used a neural network to estimate both nucleation and a size independent growth rate. The model predictions would result in predicted total crystal mass, average crystal size (based on mass) and the coefficient of variation (based on mass). The population balance model was solved using the method of moments (MoM). To estimate the nucleation and growth rate, they used off-line measured crystal mass.

In 2010, Akkisetty et al. [19] presented a hybrid discrete population balance model for a milling process. They estimated a size-dependent breakage rate using a neural network. They discretized the particle distribution into four size bins and used off-line measured particle size distributions to fit the neural network. The reason for the low number of bins was not mentioned.

In 2019, Meng et al. [18] suggested a hybrid mass, energy and population balance model for a crystallization process. Similar to Galvanauskas et al., they used a neural network to estimate both the nucleation rate and a size independent growth rate. Furthermore, they added agglomeration to the model, and estimated a size independent agglomeration rate in the neural network as well. Apart from the energy balance, their proposed model structure was similar to the one suggested by Galvanauskas et al. [17]. To fit the neural

Journal Pre-proof

network, Meng et al. [18] also used off-line measured crystal mass.

It is evident from the aforementioned studies, that the suggested models have been strongly focused on process variables that historically have been available at a high measurement rate. Thus, in the crystallization processes [16, 17, 18], the crystal mass has been used as the observed property, and used to induce phenomena kinetics. Only in the milling application by Akkisetty et al. [19], particle size distributions have been used to fit the data-driven model.

The applied population balance models have mainly been calculated using the method of moments, reducing the dimensionality of the problem. Only in the milling case by Akkisetty et al. [19], a discretized population balance model was used, but with a small number of bins. These decisions were most likely taken due to two concerns: to prevent over-fitting of the hybrid models with fairly small training data sets and to reduce the computational complexity of training the models.

With the availability of on-line/at-line particle distribution measurements, it is possible to fit the hybrid models directly to the particle properties instead of indirect process variables such as crystal mass etc. Furthermore, due to the increased amounts of data, it is also possible to relax some of the simplifying assumptions used in the previous hybrid modelling approaches, such as low dimensional population balance models. This poses a great opportunity for obtaining new process insights and generating particle models with a greater accuracy. Additionally, methods such as automatic differentiation, have significantly reduced the computational cost of training machine learning models [20]. These developments allow for faster training of the hybrid models, opening up for training the models in real-time and potentially using them as on-line models.

In this work, a systematical hierarchical framework is presented for modelling particle processes. Provided that on-line/at-line particle analysis measurements are available, the framework offers a fast modelling approach that can provide insights into the kinetics of the physical particle phenomena taking place in a particle process. The modelling framework targets particle processes, where particle phenomena mechanisms are not well established and the particle phenomena kinetics are expected to have a multivariable nature. The framework is based on the concept of hybrid modelling, where the individual strengths of mechanistic modelling and machine learning modelling are combined. The current approach simplifies a particle process to consist of up to five general physical particle phenomena; nucleation, growth, shrinkage, agglomeration and breakage. The kinetic rates of these phenomena are estimated using a deep neural network, given a number of on-line process sensor inputs. By combining the kinetic model with a mechanistic discretized population balance model, future particle attributes can be predicted, without the need for investigating the underlying phenomena kinetics. The application of the framework is demonstrated through three case studies, including a laboratory scale food ingredient crystallization, a laboratory scale flocculation/breakage process, and an industrial scale pharmaceutical crystallization.

The paper is organized as follows: In Section 2, a generic and hybrid particle model is presented based on the five general particle phenomena; nucleation, growth, shrinkage, agglomeration and breakage. In Section 3,

Journal Pre-proof

a hybrid modelling framework is presented and described in detail, going from limited prior process knowledge to model predictions. In Section 4, the application of the framework is illustrated on a laboratory scale food ingredient crystallization, a laboratory scale flocculation/breakage of inorganic particles and an industrial scale pharmaceutical crystallization. Following this, in Section 5, a discussion is given on the presented modelling approach, comparing it to previous approaches. Furthermore, its strengths and weaknesses are identified, followed by highlighting a number of future opportunities. Finally, the conclusions are summarized in Section 6.

2. Generic particle model

In this section, a generic hybrid particle process model is presented for a continuous stirred tank reactor (CSTR). First, the modelled system is described. Afterwards, the overall model structure is presented and followed by a summary of the model equations. Finally, a number of considerations addressing over-fitting and under-fitting are discussed.

2.1. System description

In the forthcoming text, a hybrid particle process model will be presented containing a solid phase (s) and a liquid phase (l). The system is simplified to be behave as a CSTR. As the model is partly data-driven, there are a number of requirements related to data-sources that need to be met:

First of all, an on-line/at-line particle analysis technique is required for obtaining discrete time measurements of one or more solid phase state variables related to the particle population. These state variables are denoted $\overline{x}^{(s)}$. Furthermore, it is required to have discrete time measurements of one or more liquid state variables measured using one or more on-line/at-line/soft sensor(s). The liquid state variables are denoted as $\overline{x}^{(l)}$. Lastly, a number of the measured state variables may be controllable, where one may apply certain control actions. The control actions are denoted as $\overline{z}$. Note that the measured state variables $\overline{x} = [\overline{x}^{(s)}, \overline{x}^{(l)}]$ are assumed to be the only variables that impact the process kinetics. Thus, to model the process to a satisfactory description of the system, one has to be able to measure the key process variables either directly or indirectly. If not, this will adversely affect the predictive qualities of the model.

2.2. Model structure

In this framework, a recursive hybrid model structure is used. The overall structure can be seen in Figure 1. A machine learning model, h, in this case a deep neural network, takes in the currently measured state variables $\overline{x}$ and the applied control actions $\overline{z}$ and calculates a number of kinetic rates $\overline{y}$. These rates are continuously fed to a set of mechanistic models for the solid and liquid state variables respectively. Using a variable-step ODE solver, the kinetic rates are calculated for integration time-steps $\mathrm{d}t$ until the model has been integrated into the future time horizon Δt.

To carry out a supervised training of the machine learning model, h, a set of inputs and outputs (also called labels in machine learning) are needed. This set can either be obtained by a direct or an indirect

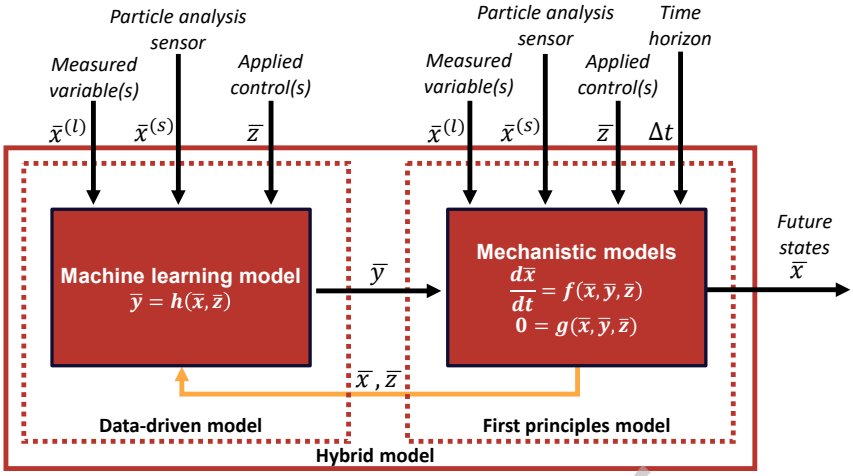


Figure 1: Hybrid model structure, where the orange arrow indicates the circular dependency used during model predictions

approach. The direct approach would be to measure the phenomena kinetics of individual particles directly. The indirect approach would be to measure the evolution of properties for a population of particles, and solve the inverse problem to estimate the kinetics. The direct approach has some practical limitations, as it is difficult to observe the properties of a specific particle without altering its processing environment. In this work, the indirect approach is used as also done previously in other works [17, 19].

Given the process states at two time stamps, $\overline{x}(t)$ and $\overline{x}(t + \Delta t)$, these can be used as input and label respectively for fitting/training the overall process model. For training of the machine learning model, h, it is assumed that the state variables $\overline{x}$ and the applied control actions $\overline{z}$ can be considered constant over the given time horizon Δt. This produces a sequential model structure, where $\overline{y}$ is constant, which yields a training model structure similar to the one used by Galvanauskas et al. [17]. For this assumption to be true, it is required that the time-gap Δt between two samples does not exceed a process specific critical time horizon, Δt_{crit}. This time horizon is based on the rate of change of the measured state variables $\overline{x}$ and the process kinetics $\overline{y}$. On the other hand, the change between the two distributions should also be significant enough to detect the changes. This can especially be difficult if the measurements are noisy.

Note, that this puts a restriction on the applicable sensors that can be used in this framework. First of all, they have to operate with a sampling frequency higher than the critical sampling rate $v_{crit} = 1/(\Delta t_{crit})$. Furthermore, the particle analysis method should be capable of measuring a statistically sufficient number of particles for each sampling to ensure that the sampling uncertainty is low enough to detect changes from sample to sample. Most commercially available on-line/at-line/in-line particle analysis methods are capable of fulfilling both criteria for industrial particle processes.

2.3. Model equations

With the overall modelling approach presented, the dynamic model equations are now presented. The model equations are divided into three sections; the balance equations, conditional equations and constitutive

Journal Pre-proof

equations, as also shown in Figure 2. The balance and conditional equations related to the particle properties are defined by first principles, whereas the constitutive equations rely on empirical relations, derived using machine learning methods. The three categories of equations are now presented individually for the proposed particle model.

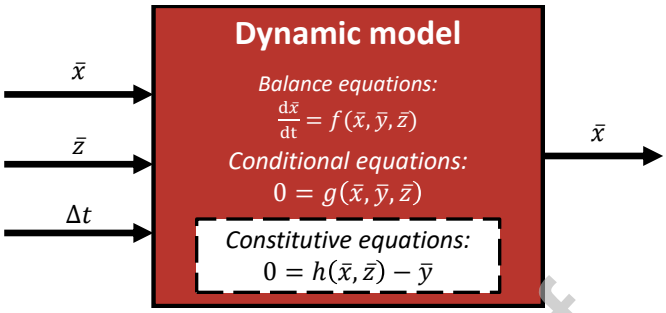


Figure 2: Dynamic model structure. Note that the constitutive equations may be calculated inside or outside the dynamic model depending on whether the model is trained or used for predictions.

2.3.1. Balance equations

To model the solid phase properties, a set of population balance equations are set up. In this framework, the main particle population state of interest, is the particle size density distribution. Thus, the particle state variable $\overline{x}^{(s)}$ is equal to the particle density variable $\overline{N}$. Using a discrete 1-dimensional population balance model, discretized into m size-bins, the accumulation of particles in each size-bin i can be described as follows, with a corresponding generation rate expression $r_i^{(s)}$:

$$\frac{\mathrm{d}\overline{x}_i^{(s)}}{\mathrm{d}t} = f_i^{(s)} + z_i^{(s)} = \frac{\mathrm{d}N_i}{\mathrm{d}t} + z_i^{(s)} = r_i^{(s)} + z_i^{(s)} \qquad \text{for} \quad i = [1; m] \tag{1}$$

Here, the control action $z_i^{(s)}$ accounts for a possibly controlled addition/removal of particles.

Furthermore, a number of balances (possibly pseudo balances) must be set up for the remaining state variables $\overline{x}^{(l)}$. These are highly system specific and require prior process knowledge to set up. A first principles model is preferred if available. If this is not possible, one can use one of the following simplications:

- In case the state-variable j is a tightly controlled variable, where one in practice can approach an ideal control: Here the applied control action can be accurately set. This means that the time-derivative of the state-variable $\mathrm{d}\overline{x}_j^{(l)}/\mathrm{d}t$ will also be known. This control action information is already supplied in the control action vector z_j, thus, the pseudo balance becomes:

$$\frac{\mathrm{d}\overline{x}_j^{(l)}}{\mathrm{d}t} = f_j^{(l)} = z_j \tag{2}$$

- In case no model is available for the state-variable j and the variable is not tightly controlled: assume the variable to be a disturbance variable and assume it to be constant, i.e. the time-derivative of the state-variable will be given as follows:

$$\frac{\mathrm{d}\overline{x}_j^{(l)}}{\mathrm{d}t} = f_j^{(l)} = 0 \tag{3}$$

Journal Pre-proof

Note that these two approaches are rather simplifying assumptions, and should only be used in absence of an accurate mechanistic or data-driven model. However, in case that the presented hybrid model is intended for on-line modelling purposes, wrongly predicted states can be substituted with new measurements continuously during operation, and can thereby reduce the effects of these errors.

2.3.2. Conditional equations

The conditional equations here deal with the generation rate expressions $r_i^{(s)}$. To define the balance rate expressions, $r_i^{(s)}$, a phenomenological approach is used. The overall generation rate is assumed to be described as the difference between the sum of particle birth and death rates, respectively, where there is no spatial dependency due to the assumption of ideal mixing. These originate from up to five physical particle phenomena including nucleation, growth, shrinkage, binary agglomeration and breakage:

$$g_i^{(s)} = \sum_{phenomena} (B_{phenomena,i} - D_{phenomena,i}) - r_i^{(s)} = 0 \quad \text{for} \quad i = [1; m] \tag{4}$$

The birth and death rates for each of these phenomena can be derived from number and mass balances, and can be generalized as shown in Table 2. Here, the number and mass balances have been kept as generic as possible to ensure that potentially all kinetic dynamics can be captured. All kinetic related variables are lumped together into the overall kinetic rate variable denoted $\overline{y}$ which is calculated using the constitutive equations. The individual kinetic rate variables are listed in Table 3. Here, each individual size-bin will have its own kinetic expression, which allows for the highest modelling flexibility. Note that the number of kinetic rates in $\overline{y}$ is depending on which phenomena that are included in the model. Also note that it is possible to reduce complexity of the model by assuming bin independent kinetics, where the dimensions of the rate variables can be reduced to unity.

Table 2: Birth and death terms for each of the five particle phenomena: nucleation, growth, shrinkage, agglomeration and breakage. Description of kinetic rate variables can be found in Table 3.

	Birth rate	Death rate
Nucleation	$B_{i=1} = \alpha$ $B_{i\neq 1} = 0$	$D_i = 0$
Growth	$B_i = \beta_{i-1} \cdot \frac{1}{2 \cdot \Delta L_{i-1}} \cdot N_{i-1}$	$D_i = \beta_i \cdot \frac{\beta_i}{2 \cdot \Delta L_i} \cdot N_i$
Shrinkage	$B_i = \gamma_{i+1} \cdot \frac{1}{2 \cdot \Delta L_{i+1}} \cdot N_{i+1}$	$D_i = \gamma_i \cdot \frac{1}{2 \cdot \Delta L_i} \cdot N_i$
Agglomeration (binary) [21]	$B_i = \sum_{j,k}^{j \geq k} \left(1 - \frac{1}{2} \cdot \delta_{j,k}\right) \cdot \eta_{j,k,i} \cdot N_j \cdot \frac{N_k}{\sum N} \cdot \epsilon_{j,k}$	$D_i = N_i \cdot \sum_k \frac{N_k}{\sum N} \cdot \epsilon_{i,k}$
Breakage [21]	$B_i = \sum_k p_k \cdot \kappa_k \cdot \theta_{i,k} \cdot N_k$	$D_i = \kappa_i \cdot N_i$

For the particle phenomena agglomeration and breakage, supplementary mass balance conditional equations are required for calculating the agglomeration contribution constant η and the breakage related number of daughter particles p. These balances can be found in Appendix A.

Journal Pre-proof

2.3.3. Constitutive equations

Lastly, the constitutive equations are defined. These come from the machine learning model h. In this work, a deep neural network which takes in the state variables $\overline{x}$ and the control actions $\overline{z}$ and calculates the kinetic rates $\overline{y}$ is employed. Depending on the choice of phenomena to include in the model, the kinetic rates $\overline{y}$ may include one or more of the variables that are listed in Table 3. The overall kinetic model structure is illustrated in Figure 3.

Table 3: Rate variables for the five particle phenomena assuming bin dependent kinetics. The dimensions of the kinetic rates can be reduced to unity in case of bin independent kinetics.

	Phenomena rates			
	Symbol	Unit	Number of variables	Scaling factor
Nucleation	α	1/(time · volume)	1	10^{-3} [-/(μL s)]
Growth	$\overline{\beta}$	size/time	m	10^{-4} [μm/s]
Shrinkage	$\overline{\gamma}$	size/time	m	10^{-4} [μm/s]
Agglomeration	$\overline{\epsilon}$	1/time	$m \cdot (m+1)/2$	10^{-4} [-/s]
Breakage	$\overline{\kappa}$	1/time	m	10^{-5} [-/s]
	$\overline{\theta}$	1	$m \cdot (m+1)/2$	N/A

The machine learning model h will have the following input and output sizes, where $n_{sensor,dim}$ is the dimension of a given sensor measurement and $n_{phenomena,dim}$ is the dimension of a given set of selected phenomena specific kinetic variables (see Table 3):

$$\text{input size} = 2 \cdot \sum_{sensor} n_{sensor,dim} + m \tag{5}$$

$$\text{output size} = \sum_{phenomena} n_{phenomena,dim} \tag{6}$$

First, the particle distribution $\overline{x}^{(s)}$ is converted into a relative distribution, by normalizing the count of particles in each bin with the total count of particles. Then, to ease the network training, a batch-normalization is applied to the relative distribution, alongside the remaining process variables $\overline{x}^{(l)}$ and the control actions $\overline{z}$. This normalizes the input data to have a batch average of zero and a batch variance that equals unity. This ensures that the initial machine learning structure has a non-biased weighting of the various process variables, which allows the training algorithm to easily select the most important features.

The construction of the neural network itself is carried out by using the following rule-of-thumb methods by Heaton [22], that relate the network structure to the above calculated input and output sizes:

- To benefit from automatic feature selection, the number of hidden layers should be more than 2
- The number of hidden neurons in each layer should be between the size of the input and the size of the output
- The number of hidden neurons should be less than 2 times the input size

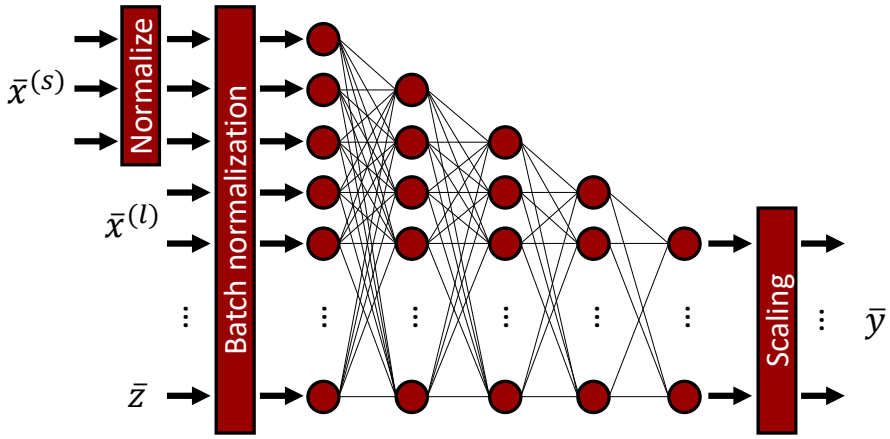


Figure 3: Example of kinetic model structure

Finally, to reduce training time and ease the convergence, the outputs from the network are multiplied with the magnitude of the order of the various phenomena. This step is crucial as the non-scaled neural network output will be approximately unity when the dense layers are initialized with random weights. By scaling the phenomena rates to their right scale from the start of the training, one can reduce training time significantly and prevent early over-fitting. Suggestions for scaling factors can be found in Table 3.

2.4. Trade-off between under-fitting and over-fitting

When constructing a partly data-driven model, one always needs to find a reasonable trade-off between model simplicity and flexibility. This is the case for conventional mechanistic models, and even more for machine learning models. Inverse problems, such as the one that needs to be solved in this modelling framework, are in many cases ill-posed, which cause an increased risk of over-fitting. To prevent this, one may apply a number of assumptions/simplifications to reduce the problem complexity. However, overly simplifying the model will result in under-fitting, where the model is not capable of capturing the observed dynamics.

The major causes for possible over-fitting and under-fitting in the presented hybrid model are summarized in Table 4. The number of phenomena and the complexity of these phenomena (bin-dependent or bin-independent) are some of the most crucial decisions concerning the mechanistic model. For small volumes of training data, one should restrict the choice of modelled phenomena to the most important ones, and assume bin-independent phenomena kinetics. For larger volumes of data and/or an on-line data-source, it is possible to use a mechanistic model with higher complexity, meaning more phenomena and/or bin-dependent phenomena kinetics.

The same goes for the machine learning model, where the number of model weights (model parameters) severely impacts the flexibility of the model, and also the risk of over-fitting and under-fitting. Also, the model flexibility/expressivity, determined by activation functions etc., has an impact. The optimal configuration needs to be determined on a case by case basis, and may be optimized for each given case either manually

Journal Pre-proof

Journal Pre-proof

Table 4: Selection of factors that impact the risk of over-fitting and under-fitting with the suggested model

	Model	Data
Mechanistic model	- Number of phenomena	- Data quantity
	- Bin-dependent/bin-independent phenomena rates	- Data uncertainty
Machine learning model	- Number of model weights	- Data quantity
	- Expressivity of the model structure	- Data uncertainty
	- Number of input variables	

or using optimization algorithms to generate the optimal network structure [23].

Finally, the number and nature of input variables (features) to the machine learning model also needs to be carefully selected. However, it has multiple times been shown in literature that newer and more complex machine learning models are now capable of carrying out this feature-selection automatically [24]. This has especially been illustrated using deep neural networks (DNN) that are capable of extracting the most important features from raw data sources like sound spectra [25] and images [26].

Apart from simplifying the problem, there are other techniques to reduce possible over-fitting. This can be done using specific techniques for training/fitting of the hybrid model, including regularization and ensembling. In this paper, only regularization will be applied, but in various forms, favouring models with smaller model weights, which typically results in smoother functions and less tendency to over-fit in the presence of data uncertainty.

Regularization terms are indirectly added to the loss function, by introducing dynamic zero-mean noise during training. This has previously been shown to have the same effect as a Tikhonov regularization term in the loss function [27]. The noise is added to the training data, including both the inputs and labels. Furthermore, regularization is applied by means of early stopping. Here, the validation error is continuously monitored during training and used as a terminating criterion when it starts to increase. This prevents the machine learning model from finding overly complex relations, and stops it from over-fitting further.

Journal Pre-proof

3. Modelling framework

In the following section, the proposed modelling framework is presented step by step. The framework consists of six main steps, which are illustrated in Figure 4 alongside the overall data-flow. The six main steps are: system specification, setting up model structure, data acquisition, data pre-processing, model training/validation and model prediction. There is a possibility of continuously refining the model by cycling between data acquisition, data pre-processing and model training, enabling on-line modelling applications. The steps are described individually in the following sub-sections.

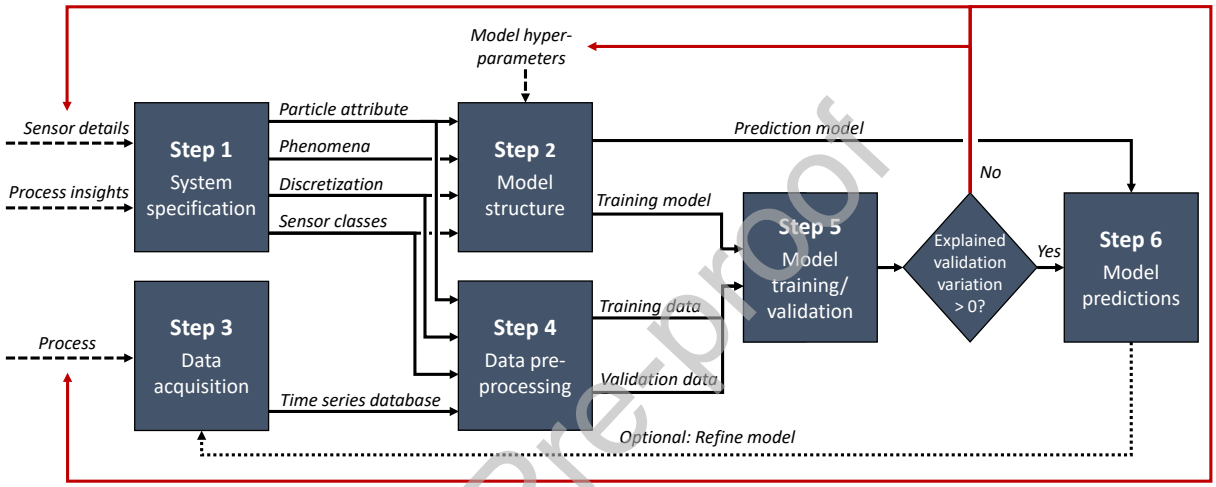


Figure 4: Outline of generic modelling framework with overall data and workflow.

The presented modelling framework has been implemented and tested in the open-source tool, Tensorflow [28], developed by the Google Brain Team. This tool was chosen as it is one of the most comprehensive open-source tools available for building customized neural networks. Furthermore, it supports implementing customized differential equation models, where automatic differentiation can be used for backpropagation when training. This increases the computational speed significantly, allowing for real-time training during process operation. Source code for the whole modelling framework is available from GitHub [29].

3.1. Step 1: System specification

The first step is to make the overall system specification. This includes specifying the particle attribute of interest and the attribute discretization. This is followed by selection of the dominating particle phenomena. Lastly, all the relevant and experimentally available process sensors (at-line-, on-line- and/or soft-sensors) are screened and classified.

3.1.1. Task 1.1: Specify particle attribute of interest

In this task, the particle size attribute of interest is to be specified. The selected particle attribute needs to be measurable using an on-line/at-line particle analysis method, with a measurement frequency

Journal Pre-proof

higher than v_{crit} (see definition in Section 2.2). For FBRM measurements, this may be chord-length, and for image analysis methods, it may be a Feret diameter. Furthermore, the on-line/at-line particle analysis method should be capable of measuring particles in the size range of interest with a resolution that allows for tracking the size-evolution. If the available particle analysis method is not applicable, then this framework will not be applicable and cannot be used for this case.

3.1.2. Task 1.2: Specify dominating particle phenomena

In this task, the dominating particle phenomena must be selected amongst the following five phenomena: nucleation, growth, shrinkage, agglomeration and breakage. The decision should be based on prior process knowledge on the particle process of interest. Enabling everything from a single particle phenomenon to all five phenomena is possible. However, one should note that the problem becomes increasingly ill-posed when more particle phenomena are considered, which increases the risk of over-fitting. Thus, the recommendation is to only select the most important particle phenomena. As a guideline, the typical dominating particle phenomena for a number of industrial particle processes can be found in the supplementary material, section C.

3.1.3. Task 1.3: Specify particle attribute discretization

In this task, a discretization scheme must be specified for the chosen particle attribute in Task 1.1. An algorithm for scheme selection and calculation is shown in Algorithm 1.1. Two different discretization schemes, one linear and one non-linear, and their corresponding equations for calculation of the discretization characteristics can be found in the supplementary material, section D.

Algorithm 1.1: Selection and calculation of discretization scheme

1. Specify the typical attribute bounds based on prior process insights: L_{min} and L_{max}
2. Specify the number of discretizations m using Rice's rule [30] in a slightly modified edition, where N is the approximate number of analysed particles in a single sample:

$$m \approx 3 \cdot N^{1/3}$$

3. Select the type of discretization scheme for the particle attribute.
   - If dominated by growth and/or shrinkage: use a linear grid
   - If dominated by agglomeration and breakage: use a nonlinear grid
4. Calculate bin mid-points (L_i), bin widths (ΔL_i) and bin edges (X_i) using the above chosen discretization scheme

   Note: *Note that the bin widths should here be coarser than the resolution of the chosen particle analysis technique, if not, revise and adjust the number of discretizations m in step 2 accordingly*

3.1.4. Task 1.4: Specify process sensors

In this task, the process sensors need to be selected and categorized. An algorithm for this procedure is summarized in Algorithm 1.2. A more thorough sensor selection is not necessary at this point, as an automatic feature extraction is used in the presented modelling approach.

Journal Pre-proof

Algorithm 1.2: Process variable screening and categorization algorithm

1. List all experimentally relevant and available process variables measurable from on-line/at-line/soft-sensors
2. For each available process variable:
   (a) Is the measurement frequency higher than v_{crit}?
   Note: *a typical v_{crit} for a particle process is within the range of 0.05 min^{-1} to 1 min^{-1}, corresponding to a measurement every 1 to 20 minutes*
   - If yes, proceed to next step
   - If no, the process variable measurement is not applicable. Return to 2. for next sensor
   (b) Can the measured process variable be tightly controlled?
   Note: *The process state is only tightly controlled if it is possible to control the state close to the ideal setpoint*
   - If yes, categorize the sensor as manipulated process variable
   - If no, proceed to next step
   (c) Can the measured process variable be predicted using an available mechanistic model?
   - If yes, categorize the sensor as modelled process variable
   - If no, categorize the variable as a disturbance variable

3.2. Step 2: Set up model structure

The particle model is now set up using the specifications supplied in step 1. First, the kinetic model is set up, followed by setting up the dynamic model, and finally connected to the training and prediction models.

3.2.1. Task 2.1: Set up kinetic rate model

In this task, the neural network kinetic rate model, denoted $\overline{y} = h(\overline{x}, \overline{z})$, is set up. The procedure is summarized in Algorithm 2.1.

Algorithm 2.1: Setting up the neural network kinetic rate model model

1. Calculate input and output dimensions of the neural network using Equation (5) and Equation (6), based on selected phenomena and process variables in Task 1.2 and 1.4 respectively
2. Specify hyper-parameters using the following guidelines:
   - Use a minimum of 3 neural network layers to benefit from automatic feature selection
   - The number of neurons in a layer should be between the size of the input and the size of the output
   - The number of neurons in a layer should be less than 2 times the input size
   Note: *The hyper-parameters need to be determined by heuristics, and can be subject to fine-tuning based on the model performance observed in the model validation/evaluation step. This includes choice of activation functions and scaling factors of the various kinetic rates*
3. Set up the network structure based on the hyper-parameters, using a batch-normalization layer as the first layer in the network

3.2.2. Task 2.2: Set up training and prediction model

In this task, the dynamic model is set up for the state variables $\overline{x}$. This includes the model for the particle density variables $\overline{N}$ and the measured liquid state variables $\overline{x}^{(l)}$. Algorithm 2.2 shows the process of setting

Journal Pre-proof

up the equations for calculating the right hand side of the ordinary differential equation system for the state variables $\overline{x}$.

Algorithm 2.2: Setting up the dynamic mechanistic model of the state variables

1. Obtain the kinetic rate variables $\overline{y}$.
   - *Training phase: Obtain as a constant parameter, based on initial conditions by calling the machine learning model* $h(\overline{x}_0, \overline{z}_0)$.
   - *Prediction phase: Obtain the parameter for each ODE-solver time-step, by calling the machine learning model* $h(\overline{x}, \overline{z})$.
2. Calculate the birth and death rates, $B_{i,phenomena}$ and $D_{i,phenomena}$.
   Note: *Only calculate for the selected phenomena in Task 1.2 for each size-bin defined in Task 1.4, using the kinetic rate variables* $\overline{y}$. *Use the generalized birth and death rates presented in Table 2.*
3. Calculate the generation rate expression $r_i^{(s)}$ using Equation (4):
4. Calculate the overall number balances $\frac{\mathrm{d}N_i}{\mathrm{d}t}$ using Equation (1):
5. For each remaining state variable j, calculate $\frac{\mathrm{d}\overline{x}_j^{(l)}}{\mathrm{d}t} = f_j^{(l)}$:
   - If manipulated process variable: assume $f_j^{(l)} = z_j^{(l)}$
   - If modelled process variable: provide system specific model for $f_j^{(l)}$
   - If disturbance process variable: assume constant $f_j^{(l)} = 0$.

The dynamic model obtained from Algorithm 2.2 is solved using a variable step ODE-solver. The input variables for the dynamic model ODE-solver, in both the training phase and prediction phase, are the initial states $\overline{x}(t)$, the control actions $\overline{z}$ and a prediction horizon Δt, where the output consists of the future states of $\overline{x}(t + \Delta t)$. Furthermore, when used for training, the dynamic model will be fed with a set of constant kinetic rates $\overline{y}$, coming from outside the dynamic model. When used for predictions, the kinetic model is evaluated for each internal time-step generated by the ODE-solver.

3.3. Step 3: Acquire time series data from process

In this step, time series data are gathered from the process sensors that were specified in Task 1.1. As the quantity of data here can end up being rather large, and in some cases needs to be accessed from multiple clients (e.g. for simultaneous model training, predictions, optimization etc.), it is advised to store this in a specifically allocated database. A suggestion for such database structure can be seen in the supplementary material, section E. From now on, this collection of data will be denoted $\mathcal{D}$.

During process operation the current process sensor readings are collected with a given frequency v, higher than the critical frequency v_{crit}. All process sensor readings, including the particle analysis, have to be taken at the same time, thus it is recommended that the process sensor readings are available on-line/at-line and electronically accessible. Particle analysis data should be stored in particle-wise manner, where each detected particle is recorded with its corresponding measured atrribute(s), where the available measured atrribute(s) depends on the chosen particle analysis method.

Journal Pre-proof

For every measurement point, all sensor measurements are saved alongside the time-stamp of the measurement. Note that the modelling framework allows for varying measurement frequency v throughout the data acquisition in case of various measurement delays e.t.c. Also note that the highest possible measurement frequency v_{max} is determined by the sensor with the lowest measurement frequency, as all measurements should be taken at the same time. This procedure can be repeated for multiple batch operations, providing more data to the model.

3.4. Step 4: Prepare time series data for model training/validation

In this step, the time series data are now pre-processed for training and validating the model. First, for each batch of data acquired in step 3, the corresponding time series data needs to be transformed into binary pairs between the measurements with time stamps t_i and t_j, where $t_i < t_j$. Algorithm 4.1 describes this process. The training data set is here denoted $\mathcal{T}$ and the validation data set is denoted $\mathcal{V}$. The subscripts *input* and *labels* denote the model inputs and output/labels respectively.

Algorithm 4.1: generating training and validation data

1. For each batch operation stored in $\mathcal{D}$:
   (a) Is the batch used for training or validation?
   - *For training, save data in* $\mathcal{T}$
   - *For validation, save data in* $\mathcal{V}$
   (b) For each binary measurement pair k and l, where $t_k < t_l$:
   i. Calculate $\Delta t = t_l - t_k$
   ii. If $v = \frac{1}{\Delta t}$ is higher than v_{crit} proceed to next step. If not, return to (b) with a new binary measurement pair.
   iii. Save Δt in $\mathcal{T}_{input}$ or $\mathcal{V}_{input}$
   iv. Calculate the particle size density distribution $\overline{N}$ for time point t_i, using the chosen discretization scheme for particle attributes from step 1, the particle feature data and the sample volume V_{ref}. Save $\overline{N}$ in $\mathcal{T}_{input}$ or $\mathcal{V}_{input}$.
   v. For all process variables in $\overline{x}^{(l)}$, save the current value at the time point t_k in $\mathcal{T}_{input}$ or $\mathcal{V}_{input}$.
   vi. For all process variables in $\overline{x}^{(l)}$, calculate the corresponding control action vector $\overline{z}$ and save this control action vector in $\mathcal{T}_{input}$ or $\mathcal{V}_{input}$
   - If manipulated variable: use the time-derivative of the variable:

$$z_j^{(l)} = \frac{\overline{x}_j^{(l)}(t_l) - \overline{x}_j^{(l)}(t_k)}{\Delta t}$$

   - If modelled or disturbance variable: set $z_j^{(l)} = 0$.
   vii. Repeat iv. for time-point t_k and save this value in $\mathcal{T}_{labels}$ or $\mathcal{V}_{labels}$.

3.5. Step 5: Model training

In this step, the training model from step 3, is fitted to the generated training data obtained from step 4. An algorithm for carrying out this training is presented in Algorithm 5.1. Formally, the training is an

optimization problem that can be described as follows, where w is the vector of machine learning model parameters in the machine learning model h:

$$\min_{w} loss(\text{model}(\mathcal{T}_{input}), \mathcal{T}_{labels}) \tag{7}$$

During model training, the model performance is validated/evaluated based on a validation data-set, as described in Algorithm 5.2.

Algorithm 5.1: training the machine learning model h.

1. Select an optimization method for training the machine learning model

   Note: *It is here advised to use a gradient descent based method, like the Adam optimizer (adaptive moment estimation)*

2. Specify a loss function *loss* which transforms the multi-objective optimization to a single-objective optimization.

   Note: *One can use one of the following objective functions, based on the amount of particle analysis measurement noise*

   - Particle analysis with low noise: Use a scaled L_1 norm of the absolute particle density N_i, where $n_{samples}$ is the number of samples in the training set:

   $$loss \equiv \sum_{k=1}^{n_{samples}} \sum_{i=1}^{m} \left| \frac{N_i^{prediction}}{\sum_{k=1}^{m} N_k^{target}} - \frac{N_i^{target}}{\sum_{k=1}^{m} N_k^{target}} \right| \Big/ n_{samples} \tag{8}$$

   - Particle analysis with medium noise: Use a L_1 norm of the relative particle density $N_i / \sum_i N_i$, where $n_{samples}$ is the number of samples in the training set. Note that this method will only provide reliable predictions of the relative particle density distribution, as the overall particle density $\sum_i N_i$ is not trained in this approach:

   $$loss \equiv \sum_{k=1}^{n_{samples}} \sum_{i=1}^{m} \left| \frac{N_{i,k}^{prediction}}{\sum_i N_{i,k}^{prediction}} - \frac{N_{i,k}^{target}}{\sum_{i,k} N_i^{target}} \right| \Big/ n_{samples} \tag{9}$$

3. **Optional:** Apply one or more regularization methods to reduce the risk of over-fitting and improve generalization.
4. Run training until convergence

Two regularization methods are here recommended for reducing over-fitting during model training, including early stopping and applying noise during training. For early stopping, the validation loss must be calculated for each epoch (see Algorithm 5.2 for procedure), where the training is stopped when the loss on the validation dataset starts to increase.

Furthermore regularization is introduced by generating gaussian zero-mean noise for each optimization iteration and introduce it to the input and output training data, corresponding to measurement uncertainties, if these can be measured or estimated. For the measured particle distributions, the inherent sampling uncertainty related to sample-based particle analysis techniques is used to estimate the uncertainty. The uncertainty here corresponds to the statistical error of sampling with replacement. Assuming an unbiased measured particle size distribution, this uncertainty becomes [31]:

$$\sigma_i = \sqrt{\left(N_i \cdot (1 - N_i / \sum_i N_i \right)} \tag{10}$$

Journal Pre-proof

Journal Pre-proof

The noise introduced to both inputs and labels, has a standard deviation equal to this uncertainty.

Note that when using gradient descent based methods, the gradient $\nabla error(w)$ is likely to be too computationally heavy to evaluate using classical numerical approaches, which makes it infeasible to carry out model training if the number of machine learning parameters is too high. This problem can be resolved if the model structure is implemented in a programming framework that supports automatic differentiation, where backpropagation can be used to significantly speed up these evaluations. I.e. the computational cost of calculating the gradient of a single value loss function using a numerical finite-differences method, is at least $O(2 \cdot n)$, where n is the number of weights. With backpropagation, using automatic differentiation, this cost is $O(1)$. The improvement over numerical approaches is especially evident in this framework, as solving the set of ODEs can be computationally heavy by itself.

Algorithm 5.2: model validation/evaluation during model training

1. Calculate the loss for the model prediction and the reference loss when using a persistence method (no-change model), using the validation data, $\mathcal{V}$:

$$loss_{model} = \text{loss}(x_{prediction} = \text{model}(\mathcal{V}_{input}), x_{true} = \mathcal{V}_{labels})$$
$$loss_{reference} = \text{loss}(x_{prediction} = \mathcal{V}_{input}, x_{true} = \mathcal{V}_{labels})$$

2. Calculate the explained variation metric, based on calculated persistence method and model loss:

$$\text{explained variation} = \frac{loss_{reference} - loss_{model}}{loss_{reference}}$$

3. If explained variation > 0, the model can be said to be predictive, and suitable for predictions. If not, there can be multiple causes, including lack of information coming from the chosen sensors in step 1, wrong choice of phenomena in step 1, under-fitting/over-fitting due to chosen model hyper-parameters in step 2, and last but not least, lack of data in step 3. One should here reconsult steps 1, 2 and/or 3, following this order.

3.6. Step 6: Generate process predictions

In this final step, process predictions are carried out. Using the trained machine learning model, the prediction model implemented in Step 2 can be used directly to make predictions into a future time horizon Δt, where the only needed specifications are the initial states $\overline{x}$ and the control actions as a function of time $\overline{z}$(t).

In case that on-line data is available, it is possible to continuously refine the hybrid model, and thereby transforming it into an on-line model. This can be done by using incremental learning, which is also a rising topic when using machine learning for classification problems [32]. Here, every time a new data-point becomes available, steps 3 to 5 are repeated. This results in a constantly growing dataset $\mathcal{T}_{input}$ which the model can be trained upon. From a computational power point of view, it may be desirable to limit the size of the training-data by continuously removing older data-points as new data become available. This will at the same time introduce a 'forgetting' behaviour, suitable for particle processes where batch-to-batch drifting

Journal Pre-proof

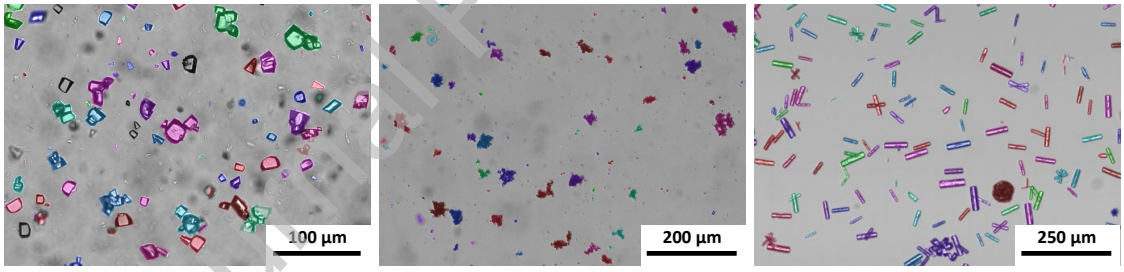
may be present.

4. Case studies

In this section, three case studies of particle processes are examined using the framework presented in Section 3. The three case studies consist of a laboratory scale crystallization of lactose, a laboratory flocculation and breakage of silica particles suspended in water and an industrial scale pharmaceutical crystallization. Examples of segmented microscopy images from the three case studies can be seen Figure 5. An overview of the case studies is presented in Table 5.

Table 5: An overview over the three case studies that have been examined using the presented modelling framework

	Lactose crystallization	Silica flocculation	Pharmaceutical crystallization
Type	Cooling crystallization	pH induced flocculation	Anti-solvent crystallization
Scale	Lab	Lab	Industrial
Particle analysis method	Flow cell (on-line)	Titer plate (at-line)	Titer plate (at-line)
Measured variables	Temperature	pH	Temperature, pH, Conductivity
Controlled variables	Temperature	pH	Temperature, pH
Number of batches [-]	2	16	5
Approx. batch duration [hr]	4	1	5
Measurement frequency [min^{-1}]	[0.17; 0.33]	0.2	[0.15; 0.23]



(a) Lactose crystals (b) Silica flocks (c) Pharmaceutical crystals

Figure 5: Segmented microscopy images from the three case studies. The coloring of the particles is random and only there to illustrate the segmentation.

In all case-studies, a non-invasive image-analysis solution (developed by ParticleTech Aps, Farum, Denmark) was used for particle analysis. The solution consists of a microscope imaging unit, a sampling unit and a software that includes both segmentation and feature-extraction algorithms. The equipment can be used for on-line, at-line and off-line applications. It uses the FluidScope$^{\text{TM}}$ technology [33] that allows for scanning particles in a volume of liquid. This is done by capturing and z-stacking a number of 6.25° tilted images.

When used as an on-line sensor, the particles are analysed using a flow cell, which is situated in the microscope imaging unit. To analyse the particles, a liquid sample is pumped from the process tank to

a flow cell, using the ParticleTech sampling unit. When the sample has reached the flow cell, the flow is stopped, and the images are captured. After imaging, the sample can potentially be sent back to the process tank, if there are no concerns regarding contamination etc. In an at-line or off-line setting, a sample would be manually taken from the process tank and pipetted to a microtiter plate, which is then placed in the microscope imaging unit and the images are captured hereafter.

The images are processed in the associated software, where the images are stitched together forming a best-focus microscopy image, ensuring that the floating particles are in focus. Afterwards, the best-focus image is analysed using the included segmentation algorithm, identifying each individual particle, which is then processed using a feature-extraction algorithm. This results in a table with the recorded features for each individual particle. The process is shown in Figure 6.

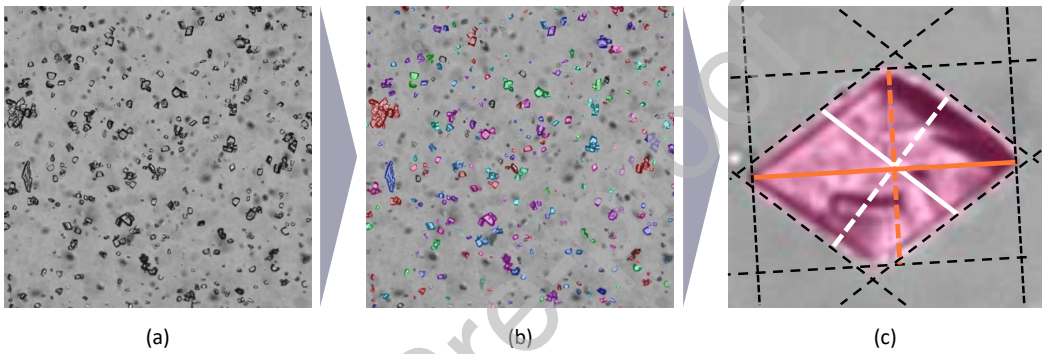


Figure 6: Particle analysis route using image analysis, consisting of (a) imaging, (b) segmentation and (c) feature extraction. A small selection of the particle features extracted by the ParticleTech software are shown in (c), where the solid white and orange lines are indicating the FeretMin and FeretMax diameters respectively. The dashed lines are the corresponding FeretMin90 and FeretMax90 diameters.

At the time of writing, it takes approximately 30-60 seconds to sample, record images, segment and extract the particle features in on-line applications using this equipment, depending on the analyzed volume and the number of particles present. For at-line applications, this may take up to 5 minutes due to the manual handling of the sample.

In the following sections, the three case studies are presented individually, including details on experimental setup, sampling method, measurement frequencies, a summary of the kinetic model structure, model training and model predictions. The details of the case studies can furthermore be found summarized in the supplementary material, section B. Note that in the following, the dynamic model has been solved using the 5th order Runge-Kutta method with adaptive step size control [34]. Furthermore, the Adam method [35] has been used for training in all cases, using a batch-size of 50 training entries per optimization iteration. Source code for the whole modelling framework, including Tensorflow models, can be found on GitHub [29]. All models have been trained and benchmarked using a 1.6 GHz quadcore CPU (Intel i5-8250U).

Journal Pre-proof

4.1. Laboratory scale lactose crystallization

In this section, a study of a laboratory scale cooling crystallization of lactose in water is presented, where only temperature and particle size distribution were measured during the process at an average sample frequency of 0.3 min^{-1}, corresponding to a measurement every 3-6 minutes. This particular case study serves the purpose of comparing the performance of the hybrid modelling framework with conventional modelling approaches.

To do so, additional prior process knowledge is needed, such as knowledge on the solubility of lactose in water as a function of temperature and the crystal density. The solubility and crystal density has previously been examined and reported in the litterature [36, 37], making it possible to calculate the super saturation in a given experiment using a first principles model. This enables the use of traditional kinetic expressions for modelling of particle phenomena such as nucleation and growth. In other words, the amount of prior process knowledge allows for using conventional modelling methods.

The crystallization was carried out in a stirred beaker, where the crystals were analysed on-line using the ParticleTech flow cell system. After analysing the liquid samples, with crystals in suspension, they were returned to the beaker to keep the volume constant. The vessel temperature was monitored using an in-line thermometer. The cooling was carried out by natural convection to the surroundings from approximately 65 °C to approximately 20 °C. The initial lactose concentrations were 0.37 g/g water in both crystallization experiments. A segmented microscopy image of the produced lactose crystals can be seen in Figure 5a.

In total, two batch operation experiments were carried out. Batch 1 contained 41 data-points and batch 2 contained 82 data-points, yielding in total 123 data-points of corresponding particle distributions and temperature measurements. As both experiments lasted approximately 4 hours, the measurement frequencies were here $v = 0.17$ min^{-1} and $v = 0.33$ min^{-1} in batch 1 and 2 respectively.

In this case study, it is decided in Task 1.1 to model the particle size distribution based on the FeretMean diameter, corresponding to the distribution of the mean diameter of the crystals. Temperature and FeretMean measurements can be seen in Figure 7, where only the median FeretMean diameter measurements, also called D50 FeretMean, have been plotted for illustration purposes.

From Figure 7, it should be noted that the temperature profiles for the two experiments are somewhat similar. However, the D50 FeretMean measures end up being rather different in the end of the two experiments. This already now indicates that the temperature measurement may not be enough to fully explain the kinetic phenomena. Also note that the rapid fluctuations in measured particle sizes in the last part of the batches are due to a high density of crystals. This makes image segmentation difficult and makes it more likely to detect multiple crystals as one.

In Task 1.2, it is assumed that the crystallization is dominated by only two particle phenomena; nucleation and bin-dependent growth. The exclusion of shrinkage is justified by the fact that the temperature is only decreasing during the experiments, which will only decrease the lactose solubility, and therefore not promote dissolution/shrinkage of the crystals.

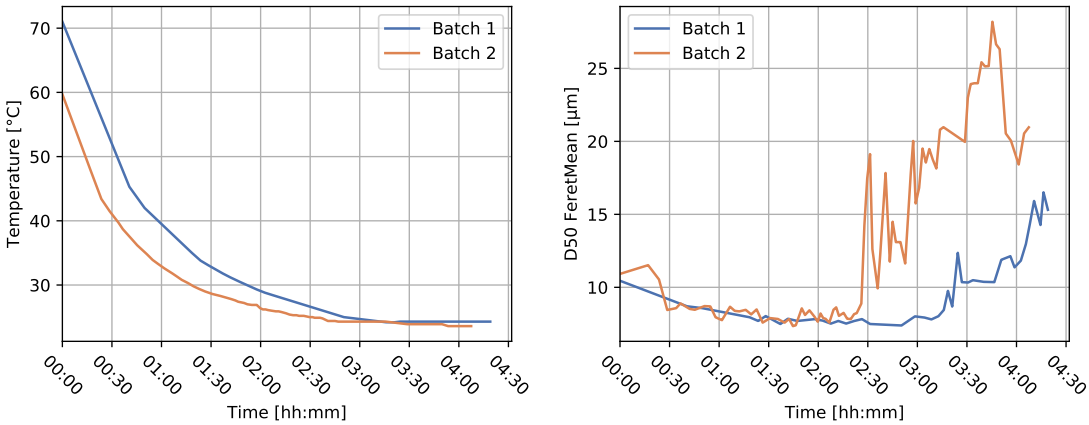


Figure 7: Overview of the sensor readings from the two lactose batch crystallizations

In Task 1.3, the particle size distribution is discretized using a linear discretization scheme (see supplementary material, section D) as the process is mainly dominated by particle nucleation and growth. The upper and lower size bins are set to be $L_{max} = 80$ μm and $L_{min} = 5$ μm respectively based on prior process knowledge. Note that all detected particles with a FeretMean diameter below 5 μm have not been accounted for, as most of the detected particles of this size were impurities and not crystals. The impurities were here mainly fat and other residues from the milk where the lactose was originally isolated from. As the image analysis equipment is set to provide measurements of approximately 1000 particles from a single image, the number of discretizations m is chosen to be $m = 3 \cdot 1000^{1/3} = 30$ following the modified Rice-equation.

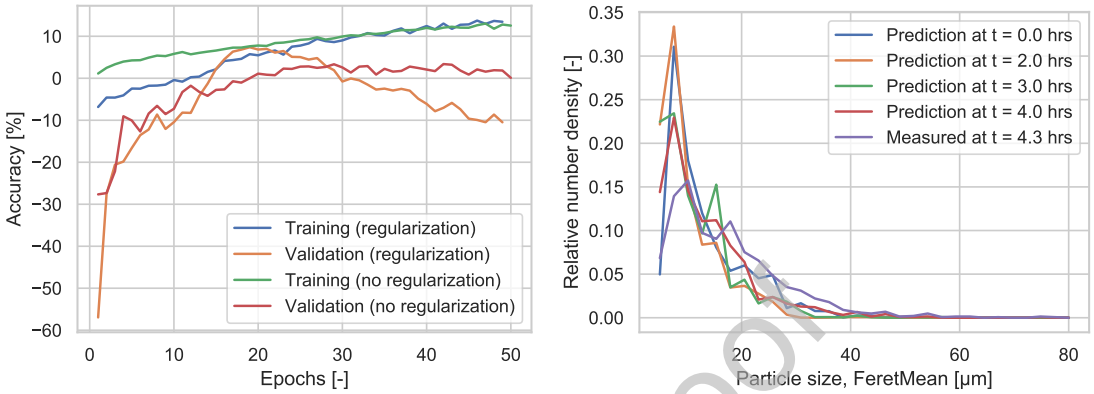
It is assumed that the temperature can be controlled tightly, qualifying it to be classified as a manipulated variable in Algorithm 1.2, Task 1.4. This was not the case with this particular setup. However, in an industrial setting of the same crystallization, one would most likely be able to apply a relatively tight temperature control.

The kinetic model is generated using Algorithm 2.1 in Task 2.1, resulting in a neural network model that consists of 4 dense neural layers in total, where the two first layers have exponential linear units (eLU) as activation functions, and the last two have linear activation functions. The number of units for each of the four layers are here chosen to be 32, 32, 32, and 31, resulting in 4,255 machine learning model parameters w_i in total. The rest of the model equations are set up following Algorithm 2.2 in Task 2.2, by using the Tensorflow implementation by [29].

Training and validation data are generated using Algorithm 4.1 with a critical measurement frequency of $v_{crit} = 0.05$ min^{-1}. Batch 2 is here used for training and batch 1 for validation. This forms in total 546 samples in the training data-set and 142 validation samples. The reason for the large difference in sample sizes in the two batch operations, is due to a higher measurement frequency in batch 2. It should here be noted that only one batch operation for training and one for validation is rather low, which also may impact the predictive qualities of the hybrid model.

Journal Pre-proof

The L_1 norm (Equation (9)) of the relative particle number density is used as the objective function in the training, when applying Algorithm 5.1. The explained variation metric is plotted in Figure 8a for training with and without regularization respectively to illustrate the effect of the regularization methods by applying Algorithm 5.2.



(a) Explained variation metric during training

(b) End-of-batch predictions using hybrid model

Figure 8: Model training and predictions for laboratory lactose crystallization

From the training graph in Figure 8a, it can be seen that the model training without regularization at first captures the correct process dynamics, but after approximately 30 epochs, the validation loss stagnates and starts to over-fit the training data. This is partly mitigated with regularization methods where noise is added during the model training and early-stopping is utilized. Here it is evident that the maximum explained variations for the validation data is higher than without regularization, indicating a better generalization. Using the implementation available on GitHub [38], model training takes approximately 7 seconds for processing the 546 training entries for one iteration (one epoch). In total, it takes 7 minutes to train the model from scratch, using early stopping and regularization.

The predictive capabilities of the hybrid model are illustrated in Figure 8b. Here, a number of end-of-batch simulations have been run for predicting the particle size distribution of batch 1 (validation batch) using initial conditions from four different time-points during the batch operation. Overall the predicted particle size distributions fit relatively well with the measured final distribution. However the rate of nucleation seems to be overestimated, resulting in wrong predictions of the number of particles in the lower size-bins. Furthermore, the growth rate seems to be underestimated slightly, resulting in an underestimation of the number of particles in the larger size-bins.

To compare the hybrid model performance, the corresponding predictions based on a conventional mechanistic crystallization model can be seen in Figure 9. The conventional model consists of a population balance model equal to the hybrid model, but using conventional simpler kinetic models for primary nucleation, secondary nucleation and growth. Furthermore, a mass balance and solubility model have been implemented, using solubility parameters and density factors obtained from literature [36, 37]. In total 6 kinetic model

Journal Pre-proof

parameters and one shape parameter were fitted using the same data as the hybrid model. Model equations for the reference model and the estimated parameters can be found in the supplementary material, section A. The parameter fitting here took 38 seconds.

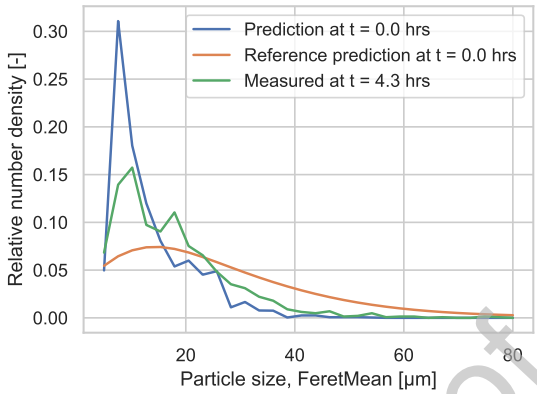


Figure 9: Conventional crystallization model predictions vs hybrid model predictions for lactose crystallization

From Figure 9, it can be seen that the quality of the hybrid model prediction of the final particle size distribution performs only slightly better than the reference prediction using the conventional mechanistic crystallization model.

It should be noted that the model performances are not completely comparable, as the validation data was used in the hybrid model to determine the optimal stopping of the model training, whereas the parameter fitting in the mechanistic model did not benefit from the validation data. On the other hand, the hybrid model did not utilize any prior information on solute concentration and solubility.

4.2. Laboratory scale flocculation and breakage

In this section, a case study is presented of a laboratory scale shear-induced flocculation of silica particles, where pH and particle size distributions were measured during the experiments. It is in this case harder to model the process with a conventional mechanistic model due to the lack of fundamental understanding of the kinetic phenomena. This does however not hinder the use of the hybrid model, where the kinetic rate expressions relies less on prior process knowledge and more on the available process time-series data.

The flocculation was carried out in a stirred 200 mL reactor by Applikon biotechnology, where the flocculation was carried out at various pH values, adjusted at the start of each batch. All the experiments were carried out with a constant stirring speed of 200 RPM. The experiments were carried out on 0.02 wt/wt % silica-water suspensions that had been ultra-sonicated for 15 minutes. The dry silica particles (size 0.5 μm to 10 μm) used for these experiments had here been washed beforehand to remove potential impurities. The washing procedure here included three rounds of washing with demineralized water, ethanol, and demineralized water for respectively 10 minutes, two hours, and 10 minutes. At last, the silica particles had been dried for 24 hours at 95 ° C in an oven.

Journal Pre-proof

During experiments, particle samples were analysed with a frequency of 0.2 min^{-1}, corresponding to every 5 minutes for one hour. The particle analysis was carried out at-line, by using a syringe to transfer liquid suspension to a titer plate and subsequently analysed in the image analysis equipment. The pH was furthermore monitored using an in-line pH probe. In total 11 batch operations were carried out. The stirring speed has been kept constant throughout all experiments. A segmented microscopy image of flocculated silica particles can be seen in Figure 5b.

In Task 1.1, it is decided to model the diameter of the equivalent circle (EQPC) distribution. To do this, it is assumed that the volume of the silica flocs corresponds to a sphere with the diameter of the measured EQPC. pH and D90 EQPC measurements are illustrated in Figure 10, where only the 90% fractile EQPC diameter measurements (D90 EQPC) have been plotted for illustration purposes. It can be seen that only two of the batches were dominated by flocculation (batch 1 and 2) indicated by an increasing D90, whereas the rest of the experiments were dominated by breakage indicated by a slightly decreasing D90. The presented data-set for this study is significantly larger with respect to number of batches when compared to the lactose crystallization study, which increases the likelihood that the hybrid model can capture the process kinetics.

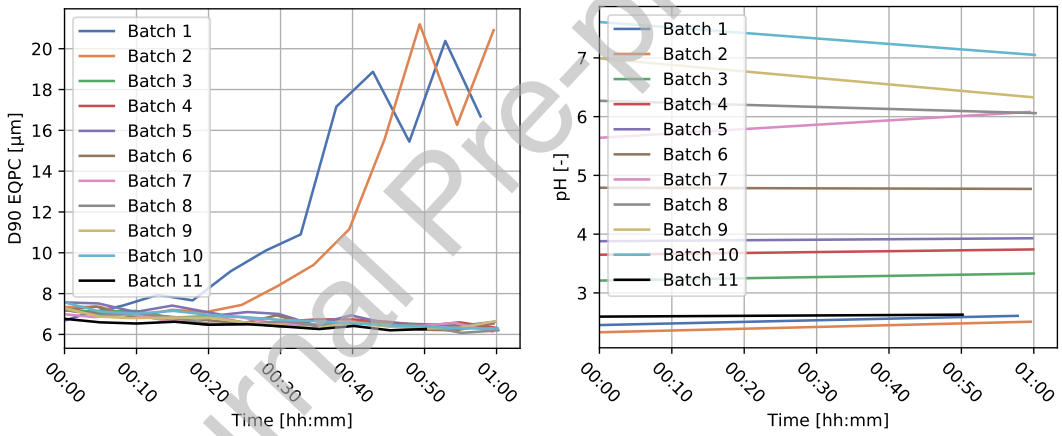


Figure 10: Overview of the sensor readings from the 11 batch flocculations.

To model this process, it is assumed in Task 1.2 that the process is dominated by breakage and agglomeration, where both phenomena are set to be bin-dependent.

In Task 1.3, the particle size distribution is discretized using a non-linear discretization scheme (see supplementary material, section D) as the process is mainly dominated by particle agglomeration and breakage. The upper and lower size bins are set to be L_{max}=30 μm and L_{min}=2 μm respectively. As the image analysis equipment has been set to provide measurements of approximately 1000 particles per image, the number of discretizations m has been chosen to be $m = 3 \cdot 1000^{1/3} = 30$ following the modified Rice-equation.

It is assumed that the pH can be controlled tightly, qualifying it to be classified as a manipulated variable in Algorithm 1.2, Task 1.4. It should here be noted that the volume effects due to pH adjustments are not included in the model.

Journal Pre-proof

By applying Algorithm 2.1, the kinetic model is set up to consists of 4 dense layers, where the first two have exponential linear units (eLU) as activation functions, and the last two have linear activation functions. The number of units for each of the four layers is set to 32, 40, 50, and 960 respectively, resulting in 51,984 machine learning model parameters w_i. As in the previous case study, the rest of the model equations are set up following Algorithm 2.2 in Task 2.2, by using the Tensorflow implementation by [29].

Training and validation data are generated using Algorithm 4.1, where the critical measurement frequency of the process is assumed to be $v_{crit} = 0.0625$ min^{-1}. This forms in total 263 samples in the training data-set and 93 validation samples, where batch 2, 6 and 11 are used as validation data and the rest for training. The L_1 norm of the relative volume density (Equation (9)) is chosen in Algorithm 5.1 as the objective function. The relative volume density is here used instead of the relative number density to better capture both agglomeration and breakage. The explained variation metric can be seen plotted in Figure 11 for training with regularization, where a maximal validation accuracy score is 32% on a volume basis. It should be noted that the sampling uncertainty of the particle analysis measurements is rather high for this case study, as the changes from measurement to measurement are relatively subtle.

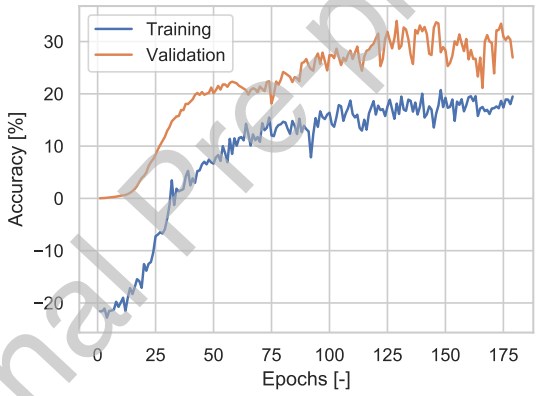
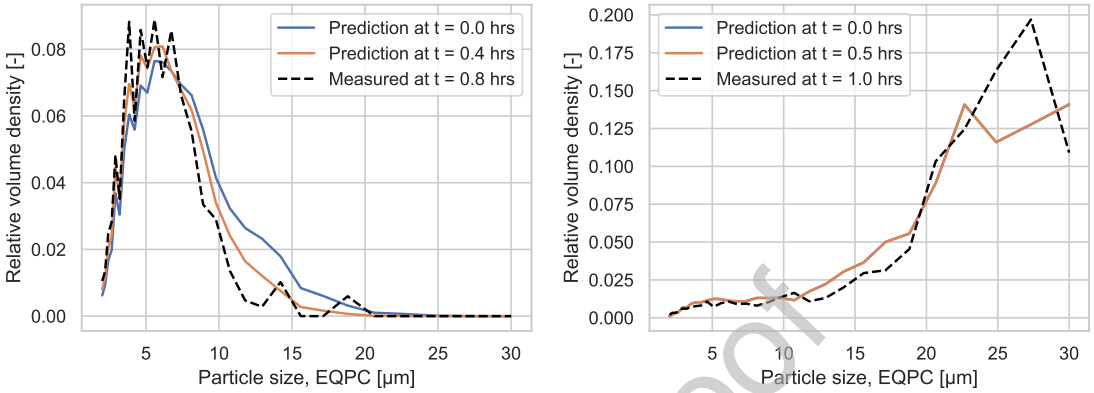


Figure 11: Explained variation metric during training

Using the implementation available on GitHub [38], model training here takes approximately 15.6 minutes to train the model from scratch (134 epochs), using early stopping and regularization. The predictive capabilities of the hybrid model are illustrated in Figure 12a, where end-of-batch simulations for batch 11 have been carried out, using initial conditions from two different time points during the batch operation. Note that batch 11 was mainly dominated by breakage and not flocculation. This is also accurately predicted by the hybrid model.

Contrary to batch 11, batch 2 did exhibit flocculation. To illustrate the quality of the estimation of the agglomeration, end-of-batch simulations for batch 2 are plotted in Figure 12b. Note that the two end-of-batch predictions are practically identical, where the prediction at t=0.5 hours is predicting slightly lower density of the highest size-bin. It can be seen that the hybrid model predicts the particle size distribution to a good extent, however with a slight underestimation of particles in the upper size bins. The prediction error

may be explained by two factors; measurement uncertainty as the number of larger particles is relatively low compared to smaller particles, and the fact that pH may not be able to solely explain the phenomena. One could therefore consider to include additional process variables that may have an effect on the phenomena rates.



(a) End-of-batch predictions for batch 11

(b) End-of-batch predictions for batch 2

Figure 12: Model predictions for laboratory flocculation and breakage. Note that the relative density plots are on a volume basis

It should be noted that no flocculants/coagulants were dosed to the system. In order to provide appropriate surfaces for the particles to agglomerate reliably, it is most likely necessary to use a flocculant, for instance a polymer. For future experiments, it could therefore be interesting to use polymer dosage as an additional process variable.

4.3. Industrial scale pharmaceutical crystallization

In this section, a study of a an industrial scale pharmaceutical crystallization is presented. The name of the compound is not to be mentioned here due to confidentiality. The crystallization is carried out in an industrial scale crystallization tank, where pH and temperature are measured and regulated to facilitate the crystallization of the rod-shaped crystals which can be seen in the segmented microscopy image in Figure 5c. Furthermore, conductivity was measured during process operation.

In this case, the solubility of the pharmaceutical is not fully characterized as it is in the lactose case study. Without this prior process knowledge, it is not possible to use conventional expressions using the degree of saturation to estimate the kinetic rates of the various particle phenomena. However, as also mentioned in the flocculation case study, the lack of prior process knowledge does not hinder the use of the hybrid model, where the kinetic rate expressions relies less on prior process knowledge and more on the available process time-series data.

At-line sampling of the crystals was used due to GMP regulations, where samples where taken from a sample valve located at the bottom of the tank. The samples were transferred to a titer plate and diluted

Journal Pre-proof

with crystal-free mother liquor to ensure a good image analysis segmentation. In total, five batch experiments were carried out (4-5 hours each), forming in total 272 data-points of corresponding particle distributions, temperature, pH and conductivity. Due to the manual sampling required in the at-line setup, small samples (≤ 0.5 mL) were withdrawn with an average frequency of 0.2 min^{-1}, corresponding to a measurement every 4-5 minutes.

In Task 1.1, it is decided to study the FeretMax diameter is to be modelled, which corresponds to the length of the crystal. pH, temperature, conductivity and D50 FeretMax measurements (median FeretMax diameters) are illustrated in Figure 13.

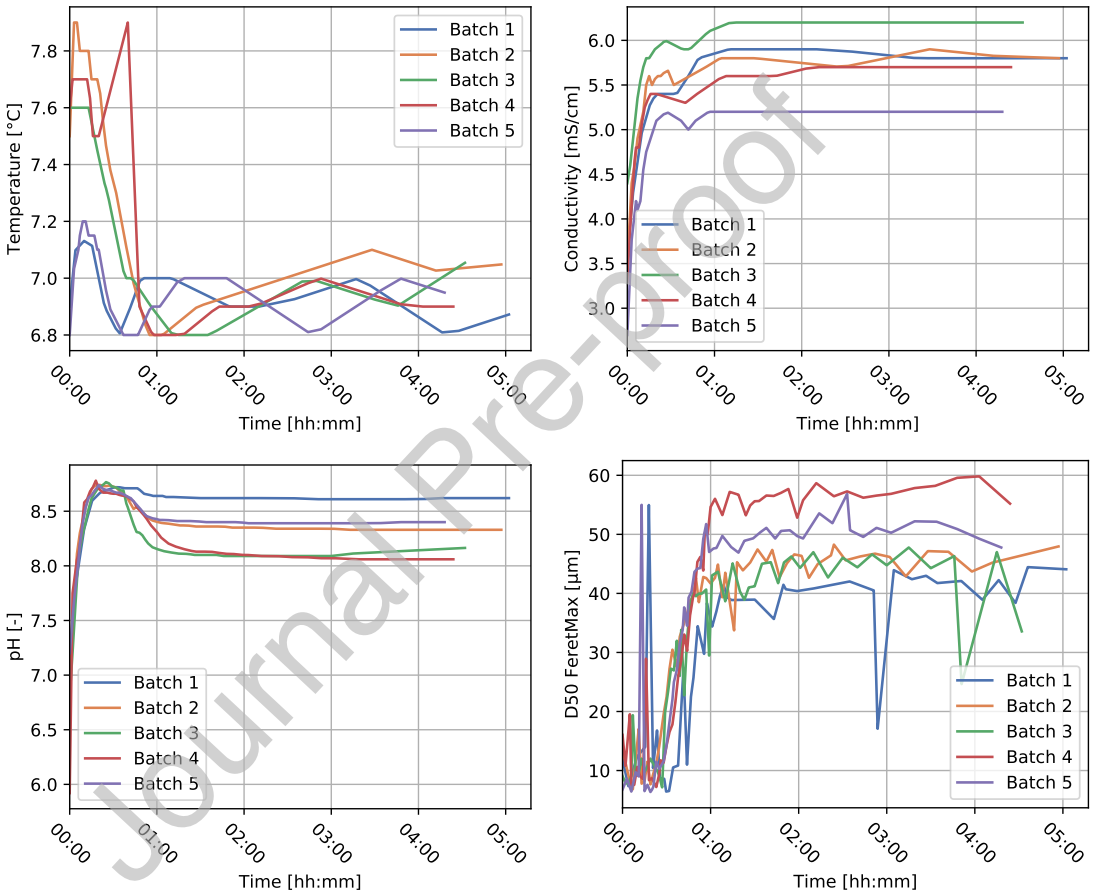


Figure 13: Overview over the sensor readings from the particle sensor and the three other process sensors

Compared to the case study of lactose crystallization, this dataset contains more batches with varying process conditions. Furthermore, this case has multiple measured process variables compared to only one in both the lactose crystallization case and the flocculation/breakage case. It is therefore expected that it will be possible to explain the process dynamics in this study relatively well.

In the following, it is assumed that the liquid volume is not changing during the process operation. In other words, the volume effects originating from pH control are neglected. Furthermore, it is assumed that

Journal Pre-proof

the reactor is well-mixed, which is not fully valid due to the size of the crystallizer. Thus, the induced kinetic expressions will be based on a tank-averaged basis instead of the local conditions. The crystal size distribution may also be different than the tank average, as the samples were only withdrawn from a single location in the tank.

In Task 1.2, it is assumed that the process is dominated by nucleation, growth and shrinkage, where growth and shrinkage are set to be bin-dependent. The reasoning behind the inclusion of shrinkage comes from the fact that no prior knowledge on solubility of the given compound is available. Furthermore, the solute concentration is not known either, and may fluctuate during the experiment. As growth and shrinkage will counteract each other, it is assumed that a specific bin can either grow or shrink - not both at the same time.

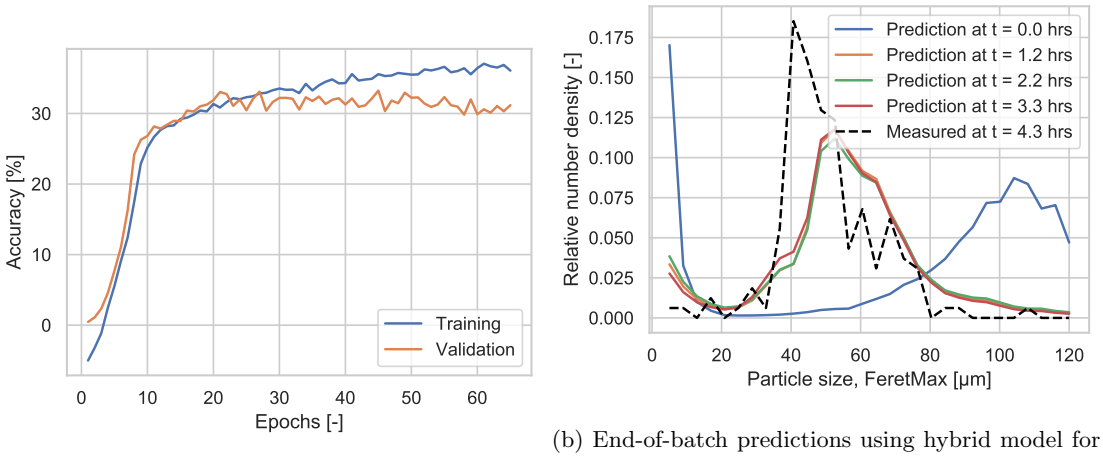
In Task 1.3, the particle size distribution is discretized using a linear discretization scheme (see supplementary material, section D) as the process is mainly dominated by particle nucleation and growth. The upper and lower size bins are chosen to be $L_{max} = 120$ μm and $L_{min} = 5$ μm respectively based on prior process knowledge. The typical number of particles analysed with the image analysis applied settings is approximately 1000, thus following the modified Rice's rule, the number of discretizations m is chosen to be $m = 3 \cdot 1000^{1/3} = 30$.

Using Algorithm 1.2, in Task 1.4, pH and temperature are classified as manipulated variables, whereas conductivity is categorized as a disturbance variable.

By using Algorithm 2.1, the kinetic model is set up to consist of 4 dense neural layers, where the first two have exponential linear units (eLU) as activation functions, and the last two have linear activation functions. The number of units for each of the four layers is 36, 40, 50, and 61 respectively. With this model structure, there are 8,117 machine learning model parameters w_i to be estimated during model training. As in the two previous case studies, the rest of the model equations are set up following Algorithm 2.2 in Task 2.2, by using the Tensorflow implementation by [29].

Training and validation data are generated using Algorithm 4.1, where the critical measurement frequency of the process is assumed to be $v_{crit} = 0.05$ min^{-1}. This forms in total 1142 samples in the training data-set and 371 validation samples, where batch 5 is used for validation data and the rest for training. The L_1 norm of the relative particle number density (Equation (9))is here used as the objective function in Algorithm 5.1. The explained variation metric can be seen plotted in Figure 14a for training with regularization using Algorithm 5.2.

From the training graph in Figure 14a, the explained variation metric can be seen to be above 30% on a number basis, which clearly indicates that the hybrid model has been able to capture a significant part of the experimentally measured process dynamics. Training of the model takes 50 epochs, corresponding to a total time for training with regularization of approximately 37.5 minutes starting from scratch. The predictive capabilities of the hybrid model are illustrated in Figure 14b, where end-of-batch particle size distribution simulations have been carried out for batch 5 (validation batch) based on initial conditions from four different



(a) Explained variation metric during training

(b) End-of-batch predictions using hybrid model for batch 5

Figure 14: Model training and predictions for industrial pharmaceutical crystallization

time points during the batch operation.

It can be seen that the trained hybrid model predicts the dynamics of the validation batch quite well for the three last predictions. This cannot be said about the initial prediction at t=0, which largely overestimates the growth rate and nucleation rate. The reason for this lies in the overly simplified modelling of the conductivity as a constant process variable. This is confirmed by the effect being present in the initial prediction where the conductivity changes rapidly and then stabilizes for the rest of the process. To avoid this model inaccuracy, it would be beneficial to either include a model for this process variable or leave it out entirely and accept a slightly lower explained variation.

5. Discussion and perspectives

It is evident from the three presented case studies, that the modelling framework suggested in this work has a minimum requirement of data with respect to both quantity and quality. This is not a surprise, as a greater part of the model is data-driven compared to first-principles approaches.

Thus, for smaller amounts of data (i.e. range 1-2 batches), as in the lactose crystallization case, it has been demonstrated that there is little to no benefit of applying the hybrid particle model framework. However, it may still be desirable to use a generic hybrid approach as the one presented in this work, in cases where it is only possible to measure process variables that have an indirect impact on the particle phenomena. For instance the pharmaceutical crystallization where solubility information and measures of solute concentration were lacking. If it is not possible to measure any process variables, one will have to resort to first principles modelling.

For larger amounts of data (i.e. range >2 batches), and in cases where the critical process variables can be measured/predicted, it is possible to apply the hybrid particle model framework with success. This was the case for the pharmaceutical crystallization where it opened up for modelling particle systems with limited

Journal Pre-proof

prior process knowledge, which is one of the major strengths of the hybrid model. It was also shown possible to predict the dynamics of the flocculation/breakage of silica particles to a good extent, using only pH as process variable.

To improve the model performance, it is however evident that one may have to introduce more knowledge to the hybrid model, by introducing more first principles models in algorithm 2.2, as also has been done in previous works on hybrid modelling of particulate systems [17, 18]. This could potentially help out faulty predictions, as seen in the case study of the pharmaceutical crystallization, where a process variable was assumed constant when in reality it is not.

It could also be relevant to introduce more first principles modelling on a molecular level, for cases such as flocculation and breakage. These processes are multiscale processes that take place from nano-scale up and beyond microscale. It is therefore reasonable to think that a more precise understanding of the nano-scale interaction between the primary particles can greatly enhance the modelling accuracy of the process beyond microscale. This could potentially transform the hybrid model presented here to a hybrid multiscale design methodology, adding more insights on the kinetics governing the system. In order to have a phenomenological understanding of the process, one could for instance use computational chemistry to estimate the surface interactions between particles and between particles and solution. The data generated from such calculations could then be used as a soft sensor to provide the neural network with more data which will enhance the kinetic predictions. As an example, for the presented flocculation case, changing pH has an impact on the surface charge of the silica particles by protonation/deprotonation of surface silanol groups. This will change the interaction between those particles. To this end, a solid/liquid interfacial tension (IFT) model can be developed to estimate the surface free energy of particle-particle and particle-solution interfaces. The method for the calculation of solid/liquid IFT uses density functional theory (DFT) calculations combined with the COSMO-RS implicit solvent model [39, 40]. The corresponding method for liquid-liquid interfaces has been published [41]. The balance of surface free energies gives the strength of attractive interaction between particles in solution and the energy required for agglomerates to break apart and make smaller particles (breakage), a property that is very difficult to measure directly. Those energies can thus be utilized as soft sensors for the estimation of kinetic parameters within the hybrid model to more precisely predict the behavior of the system.

One concern that may prohibit the use of the presented framework, lies in the cost of particle analysis equipment. Currently, it may pose a considerable capital investment, which may not be proportional to the increased accuracy/faster model development speed. However, with an increasing amount of producers of on-line/at-line particle analysis equipments, it is expected that the price will for such equipment will drop in the coming years.

It is the authors expectation that the increased availability of on-line/at-line particle analysis, and the use of hybrid model structures as the one presented in this work, poses great opportunities in several tasks in the production life-cycle. A number of examples are presented below, where a majority of these are tasks

Journal Pre-proof

where traditional modelling may be opted out due to the development cost of a mechanistic model.

This includes initial screening of optimal processing conditions and the characterization of new particulate processes. In these cases, the kinetics have not been established, which makes it infeasible to set up a mechanistic model to carry out model based design of experiments (MBDoE). As it has been shown in this work, it is computationally feasible to train such a hybrid model in real-time, by using automatic differentiation. The kinetic model can hereby continuously be adapted to new measurements, allowing modelling in the very early phases of process development, and even during process operation.

Process optimization and control is another field that could potentially benefit from adaptive modelling approaches, such as the presented hybrid particle model. Especially with the lower cost of initially applying a hybrid model, it is possible to start using a model predictive control and/or model based optimization from an earlier point in the process development.

Finally, one could potentially enable process design, process scale-up, and even integrated process design and control, based on the hybrid model if design aspects/parameters are incorporated into either the mechanistic or data-driven part of the presented hybrid model. This could potentially be applied for non-linear systems with complex dynamics, such as biotechnology and food production processes, which also recently have been pointed out as potential future fields where integrated approaches could benefit considerably to enterprise-wide sustainability [42]. One would however not use the presented hybrid model to reduce computational costs, but rather for the investigation of complex systems where the process kinetics are unknown.

In all of the above applications, model certainty, reliability and transparency are crucial for the use in real-life applications. This still remains a major frontier for application of data-driven approaches in critical decision making. Even for hybrid modelling approaches as presented here, where the back-bone consist of first-principle models. It however also requires significant changes within legislation and regulations. This is especially relevant for the use of machine learning in process control of food and pharmaceutical productions. In the current FDA (Food and Drug Administration) and EMA (European Medicines Agency) legislations, the use of machine-learning and artificial intelligence in production is still not fully supported. However, this is one of the current major focus points, where multivariate data analysis and model predictive control are highlighted as some of the crucial steps in the transition from batch to continous processing [43].

For the presented framework, in its present form, it may provide sufficient reliability and transparency for the initial process screenings and process characterizations of a given particle process. However, when it comes to process scale-up and process control, it may be necessary to add trust, for instance through uncertainty calculations. Thus, for future works, it could be interesting to look into the use of probabilistic data-driven models instead of plain neural networks. One could also think of utilizing bootstrapping methods such as neural network ensembling where multiple neural networks are trained in parallel and their individual outputs are used to estimate the certainty of the model predictions.

Journal Pre-proof

Journal Pre-proof

6. Conclusions

In this work, a modelling framework has been proposed for particle processes. The framework allows for a versatile modelling of particle processes, based on mainly qualitative process knowledge of the process phenomena and on-line/at-line sensor measurements.

The proposed modelling approach combines mechanistic modelling of the particle phenomena with a machine learning based soft-sensor for estimating particle phenomena kinetics. Here, the soft-sensor approach allows for varying the number and nature of the process sensor inputs.

The application of the framework has been demonstrated through three case studies covering a laboratory scale crystallization, a laboratory scale flocculation and breakage of inorganic particles and an industrial scale crystallization. Here it has been demonstrated that using only limited prior process knowledge and on-line/at-line particle analysis measurements, it is possible to capture and model the kinetics of various phenomena, including nucleation, growth, shrinkage, agglomeration and breakage.

The hybrid model has been evaluated and compared to conventional particle phenomena modelling, where it was demonstrated that the hybrid model performs equally good as conventional models when the amount of training data is limited, but requiring less process insights than in conventional models.

The framework has been implemented using automatic differentiation for training of the hybrid model, enabling computationally fast training, which makes it possible to train the hybrid model in real-time, which is required if the hybrid model is to be used in an on-line modelling setting.

By extending the hybrid model with model uncertainty calculations, it is expected that this model framework can be applied in various process development and operational tasks where deriving a model is opted out today. This includes model based process design, model based optimization and model based control.

Declaration of interest

The authors declare no conflict of interest.

Acknowledgements

This work partly received financial support from the Greater Copenhagen Food Innovation project (CPH-Food), Novozymes, from EUs regional fund (BIOPRO-SMV project) and from Innovation Fund Denmark through the BIOPRO2 strategic research center (Grant number 4105-00020B). Furthermore, the authors would like to thank the Otto Mønsted foundation for providing financial support covering travel expenses. Parts of the presented work have previously been presented at the 29th European Symposium on Computer Aided Process Engineering [38].

Journal Pre-proof

References

[1] Transforming our world: the 2030 Agenda for Sustainable Development, UN General Assembly, 2015.

[2] A. V. Leeuwenhoek, Part of a letter from Mr. Anthony Van Leeuwenhoek, F.R.S., concerning the figures of sand, Philos. Trans. 24 (1704) 1537–1555.

[3] J. Tyndall, Philos. Mag. 37 (1869) 384.

[4] B. J. W. S. Rayleigh, On the scattering of light by small particles, 1871.

[5] Operations Manual for PAR-TEC 100, Lasentec, 1991.

[6] K. Patchigolla, D. Wilkinson, Crystal Shape Characterisation of Dry Samples using Microscopic and Dynamic Image Analysis, Part. Part. Syst. Charact. 26 (4) (2010) 171–178, ISSN 1521-4117, doi: 10.1002/ppsc.200700030.

[7] View Particles in Real Time - Ensure Comprehensive Understanding, Mettler Toledo, 2019.

[8] J. List, U. Kohler, W. Witt, Dynamic Image Analysis extended to Fine and Coarse Particles .

[9] ParticleTech Solution, ParticleTech, URL `https://particletech.dk/particletechsolution/`, 2019.

[10] A. Sundaram, P. Ghosh, J. M. Caruthers, V. Venkatasubramanian, Design of fuel additives using neural networks and evolutionary algorithms, AIChE J. 47 (6) (2001) 1387–1406, doi:10.1002/aic.690470615.

[11] D. Chaffart, L. A. Ricardez-Sandoval, Optimization and control of a thin film growth process: A hybrid first principles/artificial neural network based multiscale modelling approach, Comput. Chem. Eng. 119 (2018) 465–479, doi:10.1016/j.compchemeng.2018.08.029.

[12] V. Galvanauskas, R. Simutis, A. Lübbert, Hybrid process models for process optimisation, monitoring and control, Bioprocess. Biosyst. Eng. 26 (6) (2004) 393–400, doi:10.1007/s00449-004-0385-x.

[13] H. Qi, X.-G. Zhou, L.-H. Liu, W.-K. Yuan, A hybrid neural network-first principles model for fixed-bed reactor, Chem. Eng. Sci. 54 (13-14) (1999) 2521–2526, ISSN 18734405, 00092509, doi:10.1016/S0009-2509(98)00523-5.

[14] J. Glassey, M. V. Stosch, Hybrid Modeling in Process Industries, CRC Press, ISBN 9781498740869, doi:10.1201/9781351184373, 2018.

[15] V. Venkatasubramanian, The promise of artificial intelligence in chemical engineering: Is it here, finally?, AIChE J. 65 (2) (2019) 466–478, doi:10.1002/aic.16489.

[16] P. Lauret, H. Boyer, J. Gatina, Hybrid Modelling of the Sucrose Crystal Growth Rate, Int. J. Model. Simul. 21 (1) (2001) 23–29, doi:10.1080/02286203.2001.11442183.

Journal Pre-proof

[17] V. Galvanauskas, P. Georgieva, S. F. D. Azevedo, Dynamic Optimisation of Industrial Sugar Crystallization Process based on a Hybrid (mechanistic+ANN) Model doi:10.1.1.124.7855.

[18] Y. Meng, S. Yu, J. Zhang, J. Qin, Z. Dong, G. Lu, H. Pang, Hybrid modeling based on mechanistic and data-driven approaches for cane sugar crystallization, J. Food Eng. 257 (2019) 44–55, ISSN 0260-8774, doi:10.1016/j.jfoodeng.2019.03.026.

[19] P. K. Akkisetty, U. Lee, G. V. Reklaitis, V. Venkatasubramanian, Population Balance Model-Based Hybrid Neural Network for a Pharmaceutical Milling Process, J. Pharm. Innov. 5 (4) (2010) 161–168, ISSN 1939-8042, doi:10.1007/s12247-010-9090-2.

[20] A. G. Baydin, B. A. Pearlmutter, A. A. Radul, J. M. Siskind, Automatic differentiation in machine learning: a survey, J. Mach. Learn. Res. 18 (1) (2017) 5595–5637, doi:10.5555/3122009.3242010.

[21] S. Kumar, D. Ramkrishna, On the solution of population balance equations by discretization - I. A fixed pivot technique, Chemical Engineering Science 51 (8) (1996) 1311–1332, doi:10.1016/0009-2509(96)88489-2.

[22] J. Heaton, Artificial Intelligence for Humans, Volume 3: Deep Learning and Neural Networks, Createspace Independent Publishing Platform, ISBN 1505714346, 2015.

[23] M. Hüsken, Y. Jin, B. Sendhoff, Structure optimization of neural networks for evolutionary design optimization, Soft Comput. 9 (1) (2005) 21–28, doi:10.1007/s00500-003-0330-y.

[24] V. Sze, Y.-H. Chen, T.-J. Yang, J. S. Emer, Efficient Processing of Deep Neural Networks: A Tutorial and Survey, P. IEEE 105 (12) (2017) 2295–2329, doi:10.1109/JPROC.2017.2761740.

[25] L. Deng, J. Li, J.-T. Huang, K. Yao, D. Yu, F. Seide, M. Seltzer, G. Zweig, X. He, J. Williams, et al., Recent advances in deep learning for speech research at Microsoft, in: 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, IEEE, 8604–8608, 2013.

[26] A. Krizhevsky, I. Sutskever, G. E. Hinton, Imagenet classification with deep convolutional neural networks, in: Advances in neural information processing systems, 1097–1105, 2012.

[27] C. M. Bishop, Training with Noise is Equivalent to Tikhonov Regularization, Neural Comput. 7 (1) (1995) 108–116, doi:10.1162/neco.1995.7.1.108.

[28] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mane, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viegas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, X. Zheng, TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems, URL `http://tensorflow.org/`, 2015.

Journal Pre-proof

[29] R. F. Nielsen, Hybrid modelling framework for particle processes, https://github.com/rfjoni/particlemodel, 2019.

[30] P. Velleman, Interactive computing for exploratory data analysis i: Display algorithms, 1975 Proceedings of the Statistical Computing Section. Washington, DC: American Statistical Association .

[31] S. Singh, Simple Random Sampling, Springer Netherlands, Dordrecht, ISBN 978-94-007-0789-4, 71–136, doi:10.1007/978-94-007-0789-4_2, 2003.

[32] F. M. Castro, M. J. Marín-Jiménez, N. Guil, C. Schmid, K. Alahari, End-to-end incremental learning, in: Proceedings of the European Conference on Computer Vision (ECCV), 233–248, 2018.

[33] oCelloScope technology, BioSense Solutions ApS, URL https://biosensesolutions.dk/wp-content/uploads/2017/05/oCelloScope-technology.pdf, 2017.

[34] L. F. Shampine, Some practical runge-kutta formulas, Mathematics of computation 46 (173) (1986) 135–150.

[35] D. P. Kingma, J. Ba, Adam: A Method for Stochastic Optimization .

[36] R. A. Visser, Supersaturation of alpha-lactose in aqueous solutions in mutarotation equilibrium., Neth. Milk Dairy J. .

[37] R. J. Lewis, Hawley's Condensed Chemical Dictionary, John Wiley & Sons, Inc., 2007.

[38] R. F. Nielsen, N. A. Kermani, L. la Cour Freiesleben, K. V. Gernaey, S. S. Mansouri, Novel strategies for predictive particle monitoring and control using advanced image analysis, in: A. A. Kiss, E. Zondervan, R. Lakerveld, L. zkan (Eds.), 29th European Symposium on Computer Aided Process Engineering, vol. 46 of *Comput. Aided Chem. Eng.*, Elsevier, 1435–1440, doi:10.1016/B978-0-12-818634-3.50240-X, 2019.

[39] A. Klamt, Conductor-like screening model for real solvents: a new approach to the quantitative calculation of solvation phenomena, J. Phys. Chem. 99 (7) (1995) 2224–2235, doi:10.1021/j100007a062.

[40] A. Klamt, V. Jonas, T. Bürger, J. C. Lohrenz, Refinement and parametrization of COSMO-RS, J. Phys. Chem. A 102 (26) (1998) 5074–5085, doi:10.1021/jp980017s.

[41] M. P. Andersson, M. V. Bennetzen, A. Klamt, S. S. Stipp, First-principles prediction of liquid/liquid interfacial tension, J. Chem. Theory Comput. 10 (8) (2014) 3401–3408, doi:10.1021/ct500266z.

[42] M. Rafiei, L. A. Ricardez-Sandoval, New frontiers, challenges, and opportunities in integration of design and control for enterprise-wide sustainability, Comput. Chem. Eng. 132 (2020) 106610, doi:10.1016/j.compchemeng.2019.106610.

[43] Quality Considerations for Continuous Manufacturing, Food and Drug Administration, 2019.

Journal Pre-proof

Nomenclature

Symbol	Description	Unit
B	Birth rate	[1/(s μL)]
D	Death rate	[1/(s μL)]
f	Balance equations	-
F	Dynamic ODE model	-
g	Conditional equations	-
h	Machine learning model	-
i	Bin index	[-]
j	Sensor index	[-]
k, l	Sample index	[-]
L	Characteristic dimension	μm
$loss$	Model loss/error function defined in Algorithm 5.1	-
m	Number of discretizations	[-]
N	Particle property density distribution	[1/(μL)]
n	Vector size	[-]
p	Number of daughter particles	[-]
r	Generation rate	[1/(s μL)]
t	Time	[s]
V	Volume	[μL]
v	Frequency	[1/s]
w	Machine learning parameter	-
X	Bin edges	μm
x	State variable	-
y	Kinetic phenomena rate	-
z	Control action	-
$\mathcal{D}$	Database of time series measurements	-
$\mathcal{T}$	Training data	-
$\mathcal{V}$	Validation data	-
Epochs	Iterations a dataset has been processed in optimization	-
MoM	Method of moments	-

Journal Pre-proof

Journal Pre-proof

Symbol	Description	Unit
α	Nucleation rate	[1/(s μL)]
β	Growth rate	[μm/s]
γ	Shrinkage rate	[μm/s]
δ	Kronecker delta	[-]
η	Agglomeration contribution constant	[-]
ϵ	Agglomeration rate	[1/s]
θ	Relative daughter particle distribtution	[-]
κ	Breakage rate	[1/s]
σ	Standard error	[-]

Journal Pre-proof

Journal Pre-proof

Appendix A. Supplementary mass balances

In this section, supplementary mass/volume balances are supplied. For both agglomeration and breakage, the particle volume needs to be estimated for each particle bin. Thus, one has to specify an expression that links the characteristic measured size L_i to the particle volume V_i.

Appendix A.1. Agglomeration

For agglomeration, an overall mass balance/volume balance is required. Here, in case of particles belonging to size bins j and k, with volume of V_j and V_k form an agglomerate that has a volume $V = V_j + V_k$. This agglomerate may lie in-between two size bins i and $i+1$. As it is a requirement that the total particle volume stays constant during such phenomena, we use a contribution constant $\eta_{j,k,i}$ that ensures that if the agglomerate lies between two size bins, the total particle volume is divided out to the two closest size bins, such that the total volume balance is fulfilled:

$$V = V_j + V_k = \eta_{j,k,i} \cdot V_i + \eta_{j,k,i+1} \cdot V_{i+1} \tag{A.1}$$

where the following summation criterion is required:

$$\eta_{j,k,i} + \eta_{j,k,i+1} = 1 \tag{A.2}$$

Thus, the contribution constant $\eta_{j,k,i}$ can be calculated as follows:

$$[\eta_{j,k,i}, \eta_{j,k,i+1}] = \begin{cases} \left[\frac{V_j+V_k-V_{i+1}}{V_i-V_{i+1}}, \frac{V_i-V_j-V_k}{V_i-V_{i+1}}\right], & V_i < V < V_{i+1} \\ [0,0]\,, \text{otherwise} \end{cases} \tag{A.3}$$

For agglomerations that results in particles larger than the size bin $i = m$, the following equation is used, to ensure a constant volume/mass:

$$\eta_{j,k,i} = \frac{V_j + V_k}{V_i}, \quad V_i \leq V_j + V_k \quad \text{AND} \quad i = m \tag{A.4}$$

Appendix A.2. Breakage

For particle breakage, an overall mass balance/volume balance is also required. When a particle from size bin i, with a volume of V_i breaks into a number of daughter particles belonging to size bins $k = [1, i]$, the total particle volume should remain constant. In other words, the sum of daughter volumes should add up to the original particle volume. This can be written as the following volume balance, where $\theta_{k,i}$ is the fraction of the volume of particles from size-bin i that goes to the formation of particles in size-bin k, and p_i is the number of daughter particles formed during the breakage of the particle in size-bin i:

$$V_i = p_i \cdot \sum_k \theta_{k,i} \cdot V_k \tag{A.5}$$

By solving for the number of daughter particles p_i, the following expression is obtained:

$$p_i = \frac{V_i}{\sum_k \theta_{k,i} \cdot V_k} \tag{A.6}$$

Journal Pre-proof

Note that $\theta_{k,i}$ here represents the volumetric daughter particle distribution, and it is subject to two constraints. First of all, the fractions distributed to the size bins k should add up to unity for a given breakage of a particle of size i:

$$\sum_{k} \theta_{k,i} = 1 \qquad \text{for} \quad i = [1; m] \tag{A.7}$$

and in the case of $k > i$, the fractions should be zero, as it is not possible to have a particle breakage forming a larger particle than the original particle:

$$\theta_{k,i} = 0 \quad \text{for} \quad k > i \tag{A.8}$$

Thus, the matrix $\theta_{k,i}$ is a lower triangular matrix, consisting of $\frac{m \cdot (m+1)}{2}$ values.

Journal Pre-proof