A Benchmark Model to Generate Batch Process Data for Machine Learning Testing and Comparison

Margarida L. C. Vicente
margarida.s.vicente@tecnico.ulisboa.pt

Instituto Superior Técnico, Lisbon, Portugal

September 2021

Abstract

During production, batch processes generate batch data and time-series values. The former contains start/end dates of production from the archived Manufacturing Execution or Batch Control systems. Furthermore, it may contain several quality-related or custom attributes. The time-series values are gathered by sensors and controllers containing information about production delays/variability. These data are valuable to build process models to support decision-making in planning and scheduling. Due to the repetitiveness nature of batch processes, there are reproducible behaviors among the data. Thereby, Machine Learning (ML) algorithms have been exploited for generating batch statistics. However, the practical use of these techniques is arduous by the noise in the data (transient, stochastic, and discrete), scarcity of quality data to train the algorithm, or process disturbances.

The main ambition of this work is to develop a benchmark model of a batch process to generate data where active challenges, disturbances, and noise are fully controlled. The process is viewed at the unit level with filling, processing, draining, and cleaning operations; the raw-materials loads are valve-controlled; and, served by hot and cold utilities. The simulation is limited to mass balances with chemical reactions and kinetics being omitted (not relevant for the intended ML application). Several scenarios were generated where 21 disturbances of different types, causes, or fault origins are injected (isolated or combined) in the implementation for testing and comparison of ML algorithms. The scenarios were grouped into six benchmark cycles with increasing levels of complexity in terms of intensity and duration tackling major challenges in ML.

Keywords: Benchmark Batch Process Model, MATLAB Simulink Stateflow, Hybrid Dynamic Simulation, Process Modelling, Machine Learning

1. Introduction

Batch processes are ubiquitous in pharmaceutical and fine-products industries [4]. They are characterized by cyclic operations of one or several units that are filled with material, perform their desired task for a given duration, shut down, drained, and cleaned before the next cycle starts [8]. The operations sequence and parameters follow a specific recipe or *production schedule*, typically based on heuristics and experience [7, 23]. In practice, disturbances and noise distress the execution of the production schedule [16]. Consequently, *production planning* needs to make **robust production plans** which allows for **variability** without having delays at the customer end [6].

Decisions in planning and scheduling are based on batch data (also designated as event-data) which generally contain start and end dates of production, and may contain further quality-related or custom attributes (such as product content). The time-series values (gathered by the sensors and controllers during production) of the process contain information about the start, end, delays, and variability of the process. For that reason, time-series values are of interest to the planner, although not ordinarily used in planning without previous processing and analysis [4, 19]. Batch data are needed to evaluate and optimize production planning, scheduling, and a wide range of retrofit optimization projects which can lead to better economic and environmental results. This can be done in several ways (e.g., by experts and Microsoft Excel calculations) but *process modelling and discrete-event simulation* is one method that can be used [1, 7].

The development and application of mechanistic models are often not economically practical for complex batch processes especially if involving a large number of equations with a significant number of process variables. For that principle, it is required several experiments for model parameterization and, the time necessary to develop the model can be quite long. As such, systems that ordi-

narily contain non-linear continuous equations and need a discrete-event handler require a specialized solver [6]. Hybrid simulation coupled with empirical models allows to train the model in an easier way (parameter estimation); to update/adapt more frequently; and, to solve it in real-time. Combining both time-driven and event-time environments still allows for fast execution if beyond the simple mass balance (reduced to a single component)—no further heat, or mass transfer phenomena, or reaction kinetics are included. The implementation of a continuous-discrete simulation can be modeled by combining Stateflow® and Simulink® [27].

Information about the start and end points of a production schedule is not always generated during production which turns into an obstacle to data analysis and process optimization (data-driven methods as well as mechanistic models). If these data are not generated during production, they need to be generated later on. One possibility to accomplish that goal is to rely on time-series data of process measurements which are much more frequently archived than batch data.

ML describes the ability of an algorithm to learn from data [9]. Due to the inherent repetitiveness of batch processes which are based on standardized recipes, there are reproducible patterns among the data. It is expected that ML can assist humans in generating predictions based on time-series data, which can be a challenging task. For this work, supervised methods of ML were used for the generation of interpretable batch data—the system learns by already being provided labels. ML performance is affected by several factors, decreasing when there is noise within the data (transient, stochastic and discrete), low amount of quality data to train the algorithm, and disturbances in the process (wrong operator interventions, breakdowns, unknown specifications of raw materials, among others) [9].

A comparatively large effort is needed for model-based process optimization on batch processes given its dynamic character and complexity (sequence of operations with possibly some occurring in parallel). As a result, the development and validation of dynamic models are often expensive or even impossible for batch processes in contrast with continuous processes, which have been plainly studied [2]. Other challenges might appear in model-building not just due to the complexity of batch process plants. Incomplete monitoring, model-uncertainties, and complex manual operations can create reasonable doubts about the quality of data acquired after implementation [4]. Batch systems are also computationally more expensive to study since it involves solving numerical differential equations due to the absence of steady-state [3]; and, chemical processes are non-linear which also makes it more difficult for an ML algorithm to predict behavior. In practice (in real production data), it is very difficult to know which challenges a data set contains. There are too many disturbances, noise in parallel, and the origin of the data is not fully known. Hence, developing a batch process that serves as a benchmark model for the development and study of data-driven techniques, in particular, ML.

This motivates the contribution of a benchmark model for the development and testing of ML algorithms for batch phase-detection problems. By working with simulated data where there is full control of the active challenges, disturbances, and noise, a study is made focusing on what makes an ML algorithm function and fail.

This article is structured as follows: the benchmark process is described in Sec. 3 after providing the base knowledge required for the understanding of the implementation in Sec. 2. The disturbance mapping and functionality of the simulation with and without disturbances are provided in Sec. 8 followed by a review of the ML algorithm feasibility study. The paper closes in Sec. 9 with some final remarks and suggestions for further work.

2. Background

This section provides a brief background on the typical causes for sensor noise and process disturbances that can be found in batch processes, whose inclusion in the benchmark process model is discussed in Sec. 6.

2.1. Sensors

In the chemical and biochemical industries, temperature, flow rate, pressure, level, and weight sensors need to be installed to monitor the processes and enable safe operation. Tab. 1 summarizes the typical causes of noise to be found [22].

At this point, it is convenient to distinguish between noise and disturbances. While **disturbances** are meant to be abnormal behaviors that deviate the operation from the expected course; **process noise** is naturally occurring fluctuations in instrumentation signals and in the process (e.g., agitation or solids being added through a hatch).

2.2. Disturbances

In a realistic setting, industrial processes are subject to disturbances and uncertainties affecting the normal practice of an industrial site [10]. These disturbances can result from a variety of sources, including external environmental variables, as discussed in Tab. 2; and, the occurrence can be random or have an underlying pattern (systematic).

Because the focus of this work is to model the phenomena which make process monitoring/fault detection needed, a brief introduction of fault diag-

Table 1: Causes of process and sensor noise for each process measurement [8, 22, 24].

Process Measurement	**Causes of Noise**
Temperature	- Time delays - Multiple thermal capacities - Compression
Flow	- Pump vibration/turbulence - Valve malfunction - Slugging - Foam formation
Pressure	- Flow changes across a junction - Cavitating pumps - Slugging - Air leakage and air blockage
Level	- Splashing and turbulence of liquid entering the tank; or, agitation - Changes on/above the surface of the liquid, p.e. foam formation
Composition	Time delays

nosis is introduced. **Fault diagnosis** is most useful, not only for the detection of faults as a batch progresses but also for revealing whether or not a specific batch belongs to the desired or normal behavior [16]. In essence, provides the knowledge to identify disturbed or normal batches. Many of the issues with sensor faults are related to the process data used for the software sensor building. The challenges with process industry data are mainly from **missing values, data outliers, drifting data, data co-linearity, sampling rates, and measurement delays** [14]. Modeling faults improves the reliability and robustness of the plant by making new decisions based on such variability [24].

Table 2: Type of disturbances and possible causes associated with them [15, 16, 21].

Type of Disturbances	**Possible causes**
Deviations in production, changeover and cleaning times	- Unknown specifications of input material - Changes in operating conditions - Unavailability of resources - Equipment malfunction - New raw material on site
Deviations in product specification	- Varying consumer demand - Unknown specifications of input material - Unknown side reactions - Tuning between shifts
Equipment failure or malfunctioning	- Aging/Clogging of equipment - Wrong operator interventions - Fouling
Erroneous sensor readings	- Faulty calibration - Presence of solid material, ice, or bubbles in a line

3. Benchmark Process Model

The benchmark case study consists of a batch process with *filling, processing, draining, and cleaning* operations wherein the entire process is viewed at the unit level. That is, all operations occur in a single vessel called Unit1_Vessel1, as depicted in the Process Flow Diagram (PFD) of Fig. 1. Three liquid raw materials (Educt1, Educt2, Educt3) and one solid input are used in the reaction, whose loads are valve-controlled; and, solids are added through a hatch placed on top of the vessel. The process model here presented is purely conceptual wherein the goal is to have a working benchmark that captures the complexity of batch operations, featured by time-dependent operations, phase transitions, and disturbances of different nature.

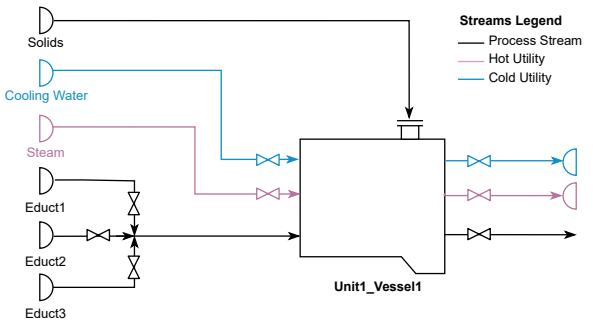

Figure 1: Simplified process flowsheet diagram of the benchmark process.

Besides raw materials, utilities are necessary to obey the process design parameters and batch recipe. It is considered that the reaction requires a hot utility to achieve the desired gradient of temperature. Although no numerical values of gradients are given, medium pressure steam is applied (since being one of the most common hot utilities in a chemical site). Following the reaction, the vessel is assumed to be cooled using water as the utility. Because most operations follow a low-temperature gradient, it is necessary to decrease the temperature of the vessel to assure the proper operating conditions. These are abstract streams where utilities are used for the simulation to be as close to reality.

Fig. 2 illustrates the path, operations (dark blue), and phases (light blue) for the chosen reference process following the nomenclature according to the ISA-88 norm (procedural-perspective class) [12]. From ID 1 to 10, a batch is produced with all the filling, processing, and draining operations. The cleaning procedure, after each batch, is given by the last operation (ID 11.1 to 11.5).

Since the focus of this work is batch phase identification, it is not the task of the ML algorithm to model heat and mass balances. For that reason, the simulation is limited to mass balances where **chemical reactions, and kinetics are omitted** (modeled variables are therefore vessel contents and their sensors, as well as valves, flows, and the hatch position). While the process just described is apparently simple, complexity rises when mathemat-

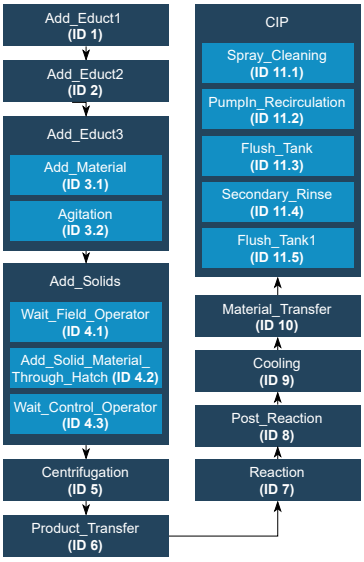

Figure 2: Benchmark Model Recipe.

ical expressions are added to emulate different behaviors at the vessel level. Instead of having only linear profiles, the vessel level can follow an exponential, step, and stair function. After establishing the recipe, it is necessary to specify duration and volume conditions for the model implementation. For this purpose, the flow rate is adjusted as a degree of freedom to connect volume with time. Characteristics of the process such as level profile and conditions can be found in Tables 3 and 4.

4. Simulation Computational Framework

Simulink® and Stateflow® are both visual programming languages. While Stateflow® is based on finite state machines and suited for discrete-event simulation; Simulink's primary applications are dynamic systems modeling and simulation. Combining discrete and continuous environments has proven to be effective and resourceful when dealing with batch scheduling [5]. As such, the process sequence as well logical behavior of operations or phases are implemented in Stateflow® and the continuous subsystem in Simulink®.

The simulation framework consists of continuous and discrete systems linked to a scenario control provided by a MATLAB script. The simulation is initialized in Simulink®/Stateflow® with the input parameters loaded in the MATLAB workspace. The simulation is then executed, and the output data is written in both environments. If chosen, the output can be written back to MATLAB unless the user interrupts the simulation in-between. Using the script allows to automate the execution of the simulation; provide relevant instructions for the simulation, and manage variables input and output. Simulink® exports the sensor values (floating values) while Stateflow® exports time start/end points of operations (integer and floating), valve positions (binary), and event labels (strings). Tab. 5 presents all the output variables of the simulation.

Table 3: Transition trigger conditions, nominal durations, and level profiles of the benchmark process.

Event ID	Transition Trigger Condition †	Nominal Duration	Level Profile
1	L1_PV >= 30%	3 min	Linear
2	L1_PV >= 60%	3 min	Linear
3.1	L1_PV >= 65% or after(5,min)	5 min	Linear
3.2	L1_PV >= 70%	5 min	Noise is added to the signal
4.1	after(15,min)	15 min	Linear
4.2	L1_PV >= 85%	5 min	Noise is added to the signal
4.3	after(5,min)	5 min	Linear
5	L1_PV >= 90% Design: 5 steps (each step with a fixed flow rate)	increase: 3 min inactivity: 7 min	Step
6	L1_PV <= 60%	3 h	Linear
7	L1_PV >= 85%	12 h	Exponential
8	after(2,h)	2 h	Linear
9	L1_PV <= 70%	1 h	Exponential
10	L1_PV <= 0%	5 min	Linear
11.1	L1_PV >= 10%	30 min	Noise is added to the signal ‡
11.2	after(14,min)	14 min	
11.3	L1_PV <= 0%	5 min	
11.4	L1_PV >= 5%	30 min	
11.5	after(2.5,min)	2.5 min	

† L1_PV corresponds to the level sensor, as a percentage of how much the vessel is filled.

‡ It is to be understood that the behavior of these phases can not be simulated only on the discrete environment. Noise is not meant to be corrected, it is part of the nominal profile of the process.

Table 4: Characteristics of the simulation.

Property	Value	Unit
Vessel volume	12.5	m^3
Threshold volume †	6.25	m^3
Batch duration	19.5	h
Cleaning procedure duration	1.4	h
Simulation Time ‡	333	days
Inactivity Period † ‡	5 min—2 h	-

† The vessel is curved, and as a result, the height has a logarithmic profile until a threshold value, followed by a linear behavior. The height is given as a function of the vessel volume with the above-mentioned threshold volume.

‡ Operation year has 333 days, already leaving a month for cleaning and maintenance.

† ‡ The inactivity period corresponds to the time between the end of a cleaning procedure and the beginning of a new batch. It is set as a random duration between 5 min and 2 h differing on each new batch production.

5. Process Model Implementation

5.1. Dynamic Model of the Holding Vessel

For this particular case, the volume of the tank is calculated from a mass balance where incompressible flow and constant density at the vessel entry and exit are assumed:

$$\frac{\text{d Volume}}{\text{d}t} = valve_{out} \cdot q_{out} - valve_{in} \cdot q_{in} \quad (1)$$

By changing the binary value of $valve_{\text{in}}$ and $valve_{\text{out}}$, it is possible to control the liquid flow direction. For simplicity's sake,, the valve positions were implemented in Stateflow® instead of Simulink®. Valves, pumps, and pipes dynamics were not considered in this work, but these features can be later implemented for additional complexity.

Table 5: Output variables of the simulation.

Variable Label	**Description**
BatchID	Number of batches produced
CleaningID	Number of cleaning operations
EventFrame ID_Reference	ID of each phase/operation
EventFrame Label_Current	Label of each phase/operation
Y1_Digital	Valve is only opened on Event ID1
Y2_Digital	Valve is only opened on Event ID2
Y3_Digital	Valve is only opened on Event ID3
Y4_Digital	Valve is only opened on Event ID5
Hatch_Digital	Hatch opens on Event ID4.2
L1_PV	Level Sensor. Percentage of how much the vessel is filled
H1_PV	Height Sensor (m)
N1_PV	Motor Sensor (rpm) turned on from Event ID3.2 to ID9
F1_PV	Cooling Water Flow Sensor (m^3/h) active for Event ID9
F2_PV	Steam Flow Sensor (m^3/h) active for Event ID7

Because the trigger transitions variables are mainly volumetric, the volume is transformed to a level signal representing the vessel filling level (already stated in Tab. 3 as $L1\_PV$). This variable becomes the sole input of the discrete environment and considers a vessel maximum volume of 12.5 m^3—consult Eq. 2.

$$Volume_P = \frac{Volume}{Volume_{vessel}} \cdot 100 \quad (2)$$

The solver used in Simulink® to integrate Eq. 1 was **ode1 (Euler)** with a fixed-step size of 10^{-3} which provides the adequate speed to simulate given its simplicity. Even providing fast calculations, ode1 is less accurate and stiff when comparing to other solvers [5]. However, the solver can be changed by the user and studied if whether it makes a difference for the ML algorithm. This solver leads to a numerical error within an acceptable tolerance given the model's level of abstraction.

5.2. Heating and Cooling

The flow rates of heating/cooling agents are assumed to increase/decrease exponentially. The implementation in Simulink® lies on the combination of two blocks: *Transport Delay* and *Transfer Fcn*. Both flows have a delay of 0.05 h but with different second-order functions parameters. The steam and cooling water flows second-order functions are given, respectively by: $\frac{10}{s^2+18s+15}$, and $\frac{5}{0.001s^2+0.2s+3}$. The parameters of these functions were driven on a basis of trial-and-error.

5.3. Noise Enabler

As mentioned in Tab. 3, during specific states, noise is added to the level signal to accomplished a specific level profile. The implementation of the Noise Enabler is the last step of the overall implementation. Nevertheless, to simplify the reading, the section is now presented since it appears in a continuous environment. For the phases being affected:

- *Agitation:* Turning the motor on for mixing to start leads to recurrent oscillations in the level sensor. For the implementation, it is assumed the volume fluctuates between -5% and +5%. The *Repeating Sequence* block provides the model with the intended behavior.
- *Adding Solids:* ID4.2 profile lies on the combination of two blocks: *Transport Delay* and *Transfer Fcn*. With the current parameters on *Transfer Fcn* ($omega = 4\ min$ and $dampingfactor = 0.4$), the behavior due to splashing and such leads to an uneven level with diminishing spikes.
- *CIP:* Cleaning procedures are often masked with heavy sensor noise due to the filling and draining of water which induces agitation and oscillation behaviors. For a heavy signal output behavior, two *Random Blocks* are used with different variations, mean values, and sample times.
- *Post Reaction:* Often these operations have small variations in volume either due to expansion/compression of volume or to changes in the vessel temperature (p.e. using utilities that increases the vessel temperature) [15]. The noise-base signal covers this physical behavior by having small spikes of 1% on top of the volume measurement. The noise implementation results in equal variances and mean values from batch to batch, but with spikes in different points in time.
- *Inactivity Period:* Small spikes on the level sensor of +10% of volume might appear. By adding the *Uniform Random Number* block to the signal, the desired behavior is obtained.

6. Disturbances Mapping and Implementation

Disturbances can be either classified by their occurrence or environment implementation. For systematic disturbances are to be considered: **step:**

mirror time effects such as a recipe change; **seasonality:** oscillation emulating time effects such as season of the year; **stair:** for example, consecutive steps (changing supplier multiple times); and, a **drift:** such as a linear term overlay. Tab. 6 lists key disturbances scenarios studied based on the information already provided in Subsection 2.2.

Disturbances are modeled stochastic, with the user controlling likelihood and severity. For each disturbance identification number, a different scenario is simulated and by class, implementation is either in Stateflow® (ID1–10 and, 13) or Simulink® (ID11, 12, and 14–21). For presentation sake, each disturbance has an ID which is the same as in Tab. 6.

7. Evaluation of ML algorithms

The most common and efficiently proven algorithms to work with time-series data in supervised learning are **Random Forests** (RF) and the **Hidden Markov Model** (HMM) [17, 26]. The following sections briefly analyze both algorithms, including their differences in implementation, accuracy, and limitations. While the HMM deals with categorical features, RF uses continuous features [17]. An introduction of the HMM and RF are provided in [20], and [20, 25], respectively.

Several studies compare these algorithms in different categories: overall accuracy on labeling, classification speed, memory consumption, feature computation, and model complexity. A comparison between both algorithms is provided by comparing their performance based on a study regarding internet traffic [17]. In this particular study, the user could be inclined to choose the RF since the overall accuracy is higher. Nonetheless, not even one of the literature studies was regarding time-series data and labels sequence recognition—which becomes an important aspect for the feasibility study of the benchmark model. In other studies, when using sequence labeling with the HMM, accuracy levels of 96% are achieved [11], and neural networks, 97% [18]. Since the article task is identical to the studies above mentioned, one could also expect similar results. Even though the RF is a great classifier, the simulation of the batch process will run with several labels and specific transitions/relations between them. This dependency between states can not be recognized with a RF.

According to [13], three criteria can be used to evaluate the algorithm learning performance: Accuracy, Confusion Matrix, and Mean Absolute Error (MAE). Firstly, **accuracy** which is given by the number of correct labels divided by the total number of labels. Secondly, a more in-depth analysis of the results obtained is possible if a **truth matrix** (confusion matrix) is established. It allows depicting the states that frequently are confused and wrongly labeled. However, this evaluation is more time-consuming than with accuracy. At last, the **MAE** is introduced. Whether a time step represents a change point or not leads to a binary class decision. The evaluation based on parameters such as accuracy and precision can be troublesome since even an error of a time step leads to a worse result. It is, therefore, easier to evaluate errors in the identification of change points using the distance between true and estimated change points. Within the scope of this work, the MAE should be considered, where:

$$MAE = \frac{\sum_{i=1}^{\#CP} \mid Predicted\ CP - Real\ CP \mid}{\#CP} \tag{3}$$

8. Results

After establishing the nominal behavior for the reference model (Sec. 8.1), by enabling the control center of each disturbance, it comes the possibility of studying non-nominal behaviors. Because more than 21 disturbances are given to the user, the presentation of each scenario would be extensive. As such, only the results for the ML training are given highlighting the most challenging profiles, disturbances, and noise for the algorithm.

8.1. Nominal Reference Model

Most disturbances are implemented within batches, being relevant to share the level measurement of a disturbance-free batch—Fig. 3. Regardless, the profile of the CIP is also introduced on the same plot.

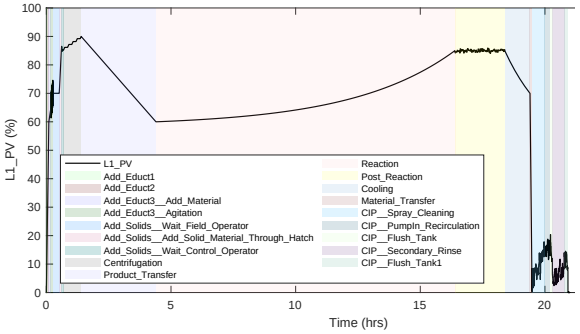

Figure 3: Nominal behavior of a batch and a cleaning procedure for the reference model with batch phase identification.

Because there is a clear dependency between continuous and discrete variables (p.e. a valve being triggered when a state is active), Fig. 4 plots the relevant output variables and allows to identify such type of relation.

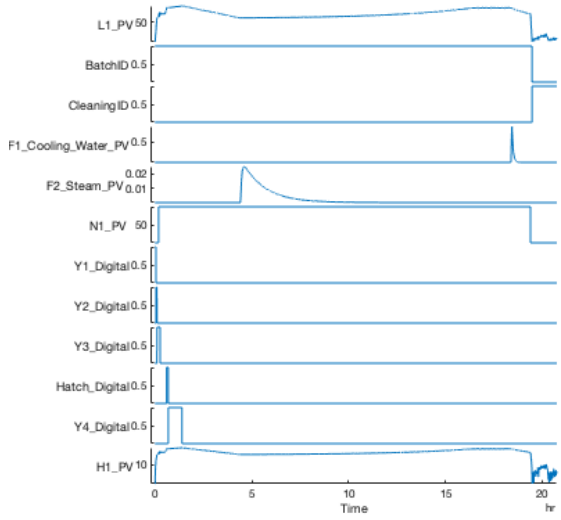

Figure 4: Nominal behavior for the relevant output variables of the reference model.

8.2. ML Feasibility Study

For this study, the simulation runs were for an entire batch-operational year.

By following a sequence with an increasing complexity level of disturbances scenarios, starting with "single" disturbances (each one implemented separately), each point in the sequence where the algorithm fails, is a good metric for its performance. Such arrangement is made regarding the disadvantages and limitations of the HMM [28]:

- **Correlation with non-linear behaviors:** There is some difficulty for ML algorithms to recognize non-linear behaviors. For that reason, different types of level profiles were added: step, stair, exponential, and noisy. Even though it is not a disturbance, these profiles add model complexity and should be recognized in a disturbance-free environment.
- **Discretization of continuous variables:** for the HMM, in particular, it is not possible to handle a combination of discrete and continuous variables. For the algorithm's training, one type of variable needs to be chosen—in this case, the user chose the discretization of continuous variables. Since discretization is accomplished by dividing time into smaller step-sizes, dynamic behaviors might be lost from step to step.
- **Dependence between two consecutive states:** adding label variability which affects the probability of an event to be recognized can be a challenge. To clarify, one example is provided: if the training data provided has missing labels (because one operation is masked as the previous one), the most difficult is the sequence to be correctly identified. This is probably the most challenging disturbance—an operation/phase label simply does not appear for the disturbed batch. As well as ID7, ID4 follows a similar line of thinking: specific phases occur irregularly affecting how the algorithm reads the recipe.
- **Not recommended to represent multiple overlapping features:** since it is a likelihood-based method, having multiple features linked to a singular state might affect the density estimation. One example might be ID11 where a new valve is switched on at random points in time. Disturbances with level variability might also affect the identification of states. If in a scenario, the volume of a batch increases by 20% at a random number of batches, the probability density of each operation will have a different minimum and maximum level measurement. This can lead to a misconception of labels leading to wrong identification.

In conclusion, the most problematic disturbances scenarios will be the ones where **likelihood and dependency between consecutive states** are affected, meaning ID4, ID5, ID7, and ID11.

8.2.1 Benchmark Cycles with Multiple Disturbances Scenarios and Algorithms Evaluation

After verifying the algorithm works properly with singular scenarios, disturbances are paired up in groups with numerous degrees of difficulty: low to high frequencies, durations, and amplitudes. By providing 21 singular disturbances scenarios, a large number of multiple scenarios can be made and it is up to the user to decide which. However, 6 cycles are presented as a challenge for the ML algorithm with increasing levels of complexity from cycles A to F. Cycle F has the same parameters as Cycle E but, with an additional disturbance: ID7 (incorrect label). Since variability and delays are the most common types of disruptions in a real process, **ID1/ID2, ID5, ID12, and ID21 are enabled for all cycles**—only variability, duration, and affected operations change.

The training and evaluation of the HMM and RF are not the focus, having this complementary study made in [13]. As this section is only to discuss those results, an overview of the achieved accuracy and MAE is given in Fig. 5 with following conclusions:

- Cycles A and B, as expected, present the best results for accuracy with the MAE decreasing slightly for the HMM on Cycle B. The overall results indicate the RF is less able to generalize

Table 6: Disturbances mapping and description, classified by cause. ID identification for each benchmark cycle with multiple disturbances scenarios (second column).

ID	Cycles	Cause of Noise/Fault	Behavior Description
1	A-F	Delay of single-phase end	Phase end is delayed, no change in actuation
2	A-F	Delay of multiple phases ends	occurs during the phase
3	C-F	An irregular single-phase occurs	e.g., if the yield was found to be insufficient after quality sample, an additional reaction step is provided
4	C-F	An irregular series of phases occur	The CIP does not always occur at the batch end
5	A-F	Tank is filled to different final fill levels	After a specific number of batches, the volume decreases between 30-70% of the batch size. Also, small variability of 2% to 5% can be added to final filling stages
6	†	Start/End points of the label(s) are shifted	End of phase 1 happens, but logged time of phase 1 is at the beginning of phase 2, the label of which is delayed
7	F	Phase wrongly labeled	It appears wrongly labeled as phase 1 lasts for the time of the 2 phases, but the sensor measurements remain the same
8	†	Two consecutive phases have the name of the first phase	
9	A-F	A valve is opened and closed several times, and nothing happens before the material transfer starts	Instead of being opened and closed once, it happens more times. Valve starts open
10	A-B, D-F	Valve opens and closes several times, and nothing happens after material transfer ends	Instead of being opened and closed once, it happens more times. Valve starts closed
11	B-F	Phase name remains the same, but the actuation changes	A different valve is opened at several instances without that behavior having meaning to the reference process
12	A-F	A pump slowly supplies less throughput	The flow rate decreases in time. To reach the same level of liquid in the tank, a certain task takes more time to complete
13	A, D-F	The motor provides less agitation	Rotation number decreases randomly, resetting after a batch
14	C-F	Utility Flow masked with noise	Incremental changes in the flow with increasing spikes
15	C	Loss of signal	A series of data points are not written for the sensors
16	†	Value outside of sensor range	If the level measurement deviates from normal values (from 0 to 100), an error message appears
17	C, D	A sensor suffers from a gradual drift for a period of time	Drift lasting until the end of the simulation, resulting in volume shift, as much as the slope chosen
18	B, E-F	A sensor suffers from a gradual drift and is suddenly recalibrated	Similar to ID15 but the sensor has a sudden recalibration after a specific number of batches
19	B, D-F	A sensor suddenly has an offset (recalibration or fault)	At random points in time, with small durations, the sensor has offsets. The actuation remains unaltered
20	B-F	A wide variety of sensor noise	Statistical distributions: Gaussian, uniform, and waveform
21	A-F	White-Band Noise	Added to the signal introducing spikes in measurements

† Implemented but not included in the benchmark cycles reported here.

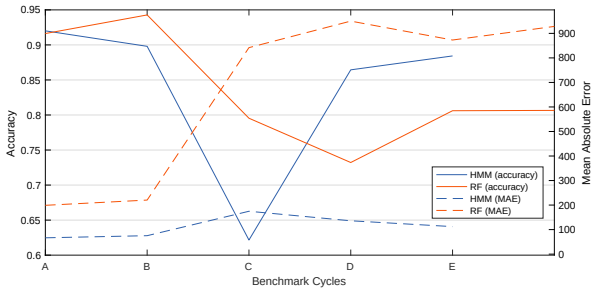

Figure 5: Evaluation of accuracy and MAE for each benchmark cycle with RF and HMM [13].

and tends to overfit. The increasing level of difficulty from Cycle A to F was almost followed for both algorithms.

- For Cycle A, the HMM correctly estimates the number of change points, and the recipe sequence is recognized and adhered to. Nonetheless, there is confusion where the change points occur being incorrectly estimated. Tables 6.1 and 6.2 of [13] allow a demonstration of the Confusion Matrix. The most problematic transitions are from $Centrifugation$ to $Product\_Transfer$ and between all phases of the cleaning procedure. Between the HMM and RF, the latter confuses these state transitions more often than the HMM. This confusion between states might be due to heavy noise being added the signal for the CIP—the overlap of volumes for consecutive states is a limitation; or, a step profile in the case of $Centrifugation$.

- Cycle C was the most challenging cycle, especially for the HMM. Because the implementation of ID15 is set to be independent on the number of batches, fewer occurrences of the CIP ended up affecting the level profile for a longer time. While the RF can order input characteristics according to their influence on the classification decision, the HMM always outputs the same phase for these batches.

- The HMM could not make any predictions for Cycle F because individual observations could not be assigned to any state. By enabling a label disturbance, this algorithm failed in recognizing states.

- Overall, the RF confuses more states (higher MAE) whereas, the HMM is still able to identify the correct sequence. This is particularly relevant when ID4 is active for Cycles C to F.

- In summary, the HMM delivers accuracies for four of the six benchmark cycles. Even though the accuracy does not reach the 96% required in Subsection 7, results are similarly reaching the most of 92% for Cycle A.

9. Conclusions

The focus of this work was to develop a benchmark model to support the testing and comparison of ML methods in their description of batch processes. The entire process is viewed at the unit level with filling, processing, draining, and cleaning operations. The model was then used to generate data where active challenges, disturbances, and data noise are fully controlled. Different scenarios were generated where up to 21 disturbances of several types, causes, or fault origins are injected into the simulated data for testing. The scenarios were grouped into six benchmark cycles with increasing levels of complexity in terms of intensity, duration, and probability.

A good balance was found by implementing the process sequence and the disturbances using both continuous and discrete environments. Simulink® and Stateflow® allow simulating faster and massive simulation scenarios with low CPU in comparison with mechanistic models. As it is not the task of the algorithms to learn heat balance and kinetics, the abstraction was made to not contemplate them. Also, a trade-off between model complexity and user-friendliness was chosen by implementing a process recipe with the current number of valves, event labels, and sensors; having such a model allows for meaningful studies which can be understood by a wide range of users.

From the algorithm's training point of view, the disturbances which caused the most damage on the algorithm training were: ID7 (purpose incorrectly labeled training data) and ID15 (loss of several signal points) combined with ID4 (phase occurring irregularly). Within the scope of the work, it is clear that HMM was able to deal with most of the implemented process disruptions and was able to make better predictions than the RF. In particular, it was shown that the HMM can learn the sequence of the recipe steps and correctly assess the number of phase transitions. Errors arise in determining the correct point in time for the phase change. For two of the six implemented test cycles, the performance of the HMM came short since it could not make any predictions (Cycle F with ID7) or only poor predictions (Cycle C with ID15). On the other hand, the RF was not affected, having accuracies of 81%/79%, respectively.

The process model here presented can be used as the basis to incorporate in the future additional features, including [24]: **1**—A valve can be open to a certain percentage or a certain degree. For example, even though the valve is opened 10%, changes might not be detected until it reaches 15% or 20% of the opening (air blockage or slugging behaviors are mimicked); **2**—Inclusion of equilibrium- and kinetically-controlled chemical reactions. If so, the complexity of the process and disturbances can be broadly increased. For instance, ID3 might have an additional reaction step if the yield is not sufficient to move forward in the operation; **3**—Temperature monitoring where effects of heat transfer on the system dynamics are taken into account; **4**—Study of the hierarchical control by fixing set points, and manipulated and controlled variables. Not many changes would be needed as Simulink® provides the necessary blocks for the implementation; **5**—More non-linear profiles can be added to the recipe as steps, stairs, or noisy profiles. As discussed in Sec. 8, these are the more problematic transitions.

Acknowledgments

The author would like to thank Dr. Franz Bähner, Dr. José Granjo, and Dr.ª Ruomu Tan.

References

[1] S. Amaran, B. Sharda, and S. J. Bury. Targeted incremental debottlenecking of batch process plants. In *2016 Winter Simulation Conference (WSC)*, pages 2924–2934, 2016.

[2] M. Barrera and L. Evans. Optimal Design and Operation of Batch Processes. *Chem. Eng. Commun.*, 82(1):45–66, 1989.

[3] D. Bonvin and D. Hunkeler. Control and optimization of batch processes: Improvement of process operation in the production of specialty chemicals. *IEEE Control Systems Magazine.*, pages 1–6, 2006.

[4] F. D. Bähner and J. K. Huusom. A Debottlenecking Study of an Industrial Pharmaceutical Batch Plant. *Ind. Eng. Chem. Res.*, 58(43):20003–20013, 2019.

[5] F. D. Bähner, O. A. Prado-Rubio, and J. K. Huusom. Discrete-continuous dynamic simulation of plantwide batch process systems in matlab. *Chem. Eng. Res. Des.*, 159:66–77, 2020.

[6] C. A. Floudas and X. Lin. Continuous-time versus discrete-time approaches for scheduling of chemical processes: A review. *Computers & Chemical Engineering*, 28:2109–2129, 2004.

[7] G. Georgiadis, A. Elekidis, and M. Georgiadis. Optimization-Based Scheduling for the Process Industries: From Theory to Real-Life Industrial Applications. *Processes*, 7(7), 2019.

[8] D. Green and M. Z. Southard. *Perry's Chemical Engineers' Handbook*. McGraw-Hill Education, New York, 9 edition, 2018.

[9] A. Gron. *Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. O'Reilly Media, Inc., 2017.

[10] J. Hahn and T. F. Edgar. *Process Control*, pages 1–44. John Wiley & Sons, 2014.

[11] Z. He, Z. Wang, W. Wei, S. Feng, X. Mao, and S. Jiang. A Survey on Recent Advances in Sequence Labeling from Deep Learning Models. 2017.

[12] ISA. ISA-88: Batch Control Part 1: Models and Terminology. 1995.

[13] G. Just. Zeitreihenanalyse zur erkennung von batchphasen in der prozessindustrie. *Technische Universität Dresden*, 2021.

[14] P. Kadlec, B. Gabrys, and S. Strandt. Data-driven Soft Sensors in the process industry. *Comput. Chem. Eng.*, 33:795–814, 2009.

[15] K. Kidam and M. Hurme. Analysis of equipment failures as contributors to chemical process accidents. *Process Safety and Environmental Protection*, 91(1):61–78, 2013.

[16] E. Korovessi and A. A. Linninger. *Batch Processes*. Taylor & Francis, 2005.

[17] A. Munther, R. R. Othman, A. S. Alsaadi, and M. Anbar. A performance study of hidden markov model and random forest in internet traffic classification. In K. J. Kim and N. Joukov, editors, *Information Science and Applications (ICISA) 2016*, pages 319–329, Singapore, 2016. Springer Singapore.

[18] M. K. Mustafa, T. Allen, and K. Appiah. A comparative review of dynamic neural networks and hidden markov model methods for mobile on-device speech recognition. *Neural Computing and Applications*, 31:891–899, 2020.

[19] D. Petrides, D. Carmichael, C. Siletti, and A. Koulouris. Biopharmaceutical process optimization with simulation and scheduling tools. *Bioengineering*, 1(4):154–187, 2014.

[20] L. Rabiner and B. Juang. An introduction to hidden markov models. *IEEE ASSP Magazine*, 3(1):4–16, 1986.

[21] J. Schumacher. A framework for batch-operation analysis within the context of disturbance management. *Computers & Chemical Engineering*, 24:1175–1180, 2000.

[22] E. D. Seborg, T. F. Edgar, and D. A. Mellichamp. *Process Dynamics and Control*. John Wiley & Sons, 4 edition, 2017.

[23] L. L. Simon, N. Osterwalder, U. Fischer, and K. Hungerbuehler. Systematic Retrofit Method for Chemical Batch Processes Using Indicators, Heuristics, and Process Models. *Ind. Eng. Chem. Res.*, 47(1):66–80, 2008.

[24] A. Stief, R. Tan, Y. Cao, J. R. Ottewill, N. F. Thornhill, and J. Baranowski. A heterogeneous benchmark dataset for data analytics: Multiphase flow facility case study. *Journal of Process Control*, 79:41–55, 2019.

[25] R. T. Trevor Hastie and J. Friedman. The Elements of Statistical Learning Data Mining, Inference, and Prediction. *Springer Science & Business Media*, 2009.

[26] R. S. Tsay and R. Chen. *Nonlinear time series analysis*. John Wiley & Sons, Inc., 2019.

[27] D. van Beek and J. Rooda. Languages and Applications in Hybrid Modelling and Simulation: Positioning of Chi. *Control Engineering Practice*, 8(1):81–91, 2000.

[28] I. Visser. Seven things to remember about hidden markov models: A tutorial on markovian models for time series. *Journal of Mathematical Psychology*, 55(6):403–415, 2011.