ContcarSim User Manual

v1.1, Francesco Destro

April 11, 2023

f.destro.10@gmail.com

Index

To accompany the article:

Destro, F., Nagy, Z. K. and M. Barolo (2022). A benchmark simulator for quality-by-design and quality-by-control studies in continuous pharmaceutical manufacturing – Intensified filtration-drying of paracetamol/ethanol slurries. *Comput. Chem. Eng.*, **163**, 107809.

The simulator has been developed on MATLAB R2019a for Windows.

Overview

This User Manual provides indication on how to use the ContCarSim simulator. Section 1 briefly describes the process. Section 2 contains a detailed explanation on how the simulator can be used and edited for different purposes. The Appendix provides the simulation inputs to be used to reproduce the sample case study contained in Destro *et al.* (2022). More information on the process and on the possible applications that are envisioned for ContCarSim is provided in Destro *et al.* (2022).

1. Process description

The carousel reproduced by the simulator is sketched in Figure 1. A schematic P&ID of the process is provided in Figure 2, with the legend of the equipment reported in Table 1. Figures 1-2 and Table 1 also report the sensors and controllers network implemented in the carousel simulator.

The unit can continuously process an inlet slurry stream into a dry crystals cake. The slurry system considered in the simulator is composed by pure paracetamol crystals in a mother liquor composed by pure ethanol. The carousel features five cylindrical ports, each one of 15 mm diameter, which allow a maximum hold-up of 10 mL. The ports are embedded in a main cylindrical body, aligned to five processing stations (Stations 1-5). For illustrative purposes, in Figure 2 the stations are represented as vessels in series (V-101-V-105), although the actual layout of the carousel is as in Figure 1. Stations 1-4 present a filter mesh at the bottom (F101-104). Station 5 is, instead, open at the bottom for cake discharge, which is enabled by the action of a pneumatic piston (not shown). The pressure gradient for filtration and drying is provided by a compressor (P101), connected to the top section of all the stations, whereas all stations are maintained at atmospheric pressure on the bottom section.

The carousel operates in cyclic mode: processing cycles, during which every port processes batch-wise the material therein contained, are alternated to carousel rotations, during which the ports containing the material being processed are moved to the following station. Carousel rotations are logically represented in the P&ID as material streams, whose flows are controlled by FC-101. The alternating processing cycles and carousels rotations are interrupted when significant mesh fouling is

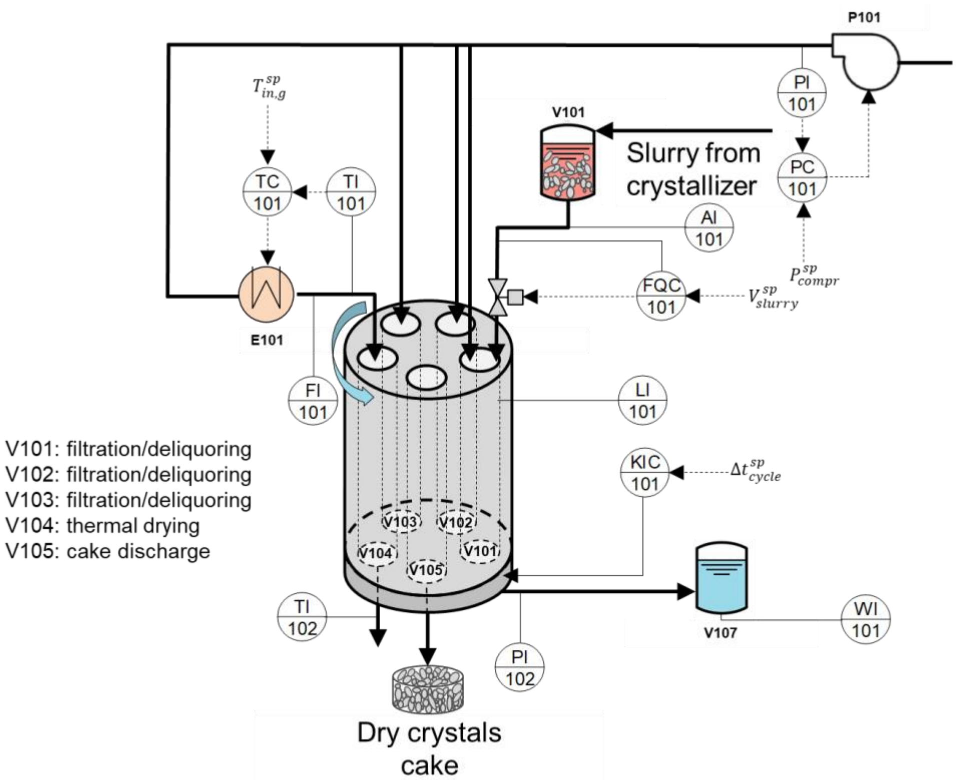

Figure 1. *Schematic drawing of the carousel for continuous integrated filtration-drying of crystallization slurries mimicked by the simulator. Filter meshes are placed at the bottom of V-101-V04 (i.e., processing stations 1–4). Station 5, instead, is open for cake discharge. In physical carousels, controllers FQC-101 and KIC-101 are routines of the programmable logic controllers of the unit.*

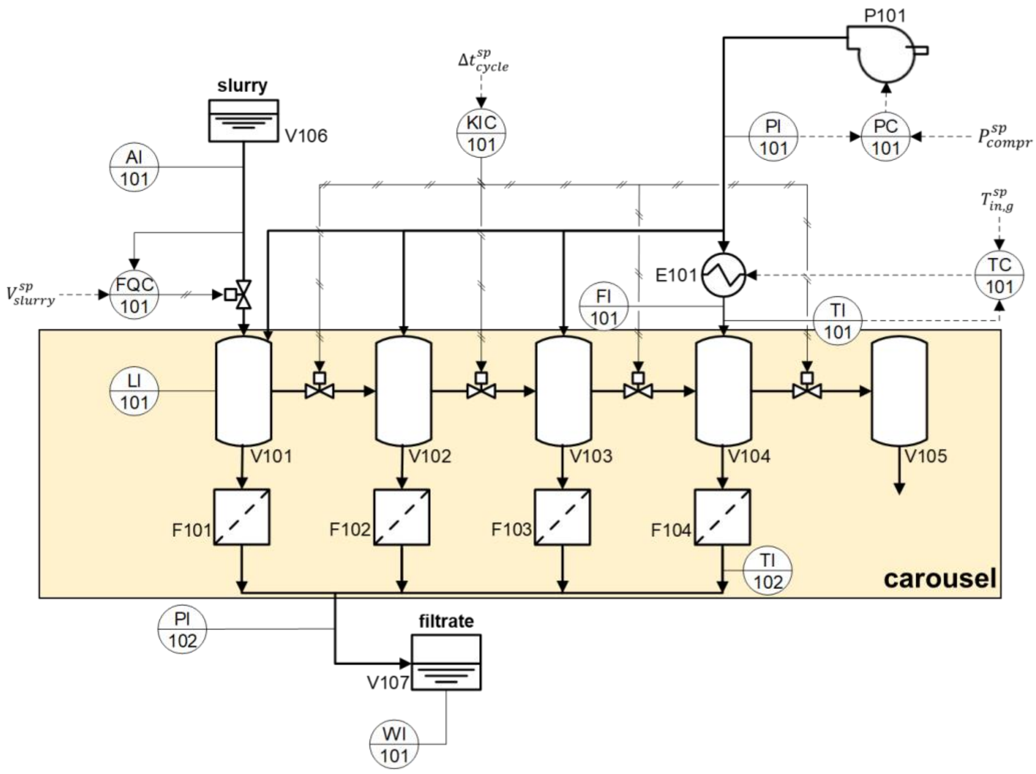

Figure 2. *Logical P&ID of the carousel illustrated in Figure 1. The equipment legend is reported in Table 1.*

detected: a cleaning-in-place cycle is then triggered. In this case, the material already loaded in the carousel ports is regularly processed during the following cycles, but no more slurry is loaded into the first station. Hence, at every cycle following cleaning-in-place initiation, an increasing number of ports will be empty. When all ports are empty, all meshes are automatically cleaned by sending a cleaning solvent into the carousel. The idle time for mesh cleaning is an input of the simulator, set to zero by default. The idle time at the end of every cycle (to sum up with the idle time for mesh cleaning when occurring), accounting for the carousel rotation and piston movement for cake discharge, is an additional input of the simulator, by default set to zero. Additional information on the alternance of processing cycles and cleaning cycles is provided in Appendix A of [1].
Stations 1-3 are dedicated to filtration and deliquoring, while in Station 4 thermal drying is carried out. In Station 5, only cake discharge occurs. Slurry processing occurs as follows. The crystallization slurry is fed to Station 1 at the beginning of every cycle, by keeping the valve between the slurry tank (V-106) and Station 1 open. After slurry feeding, a subsequent filtration step starts in Station 1, and it continues until filtration ends. If a carousel rotation is triggered before filtration finishes, filtration will continue in Station 2 (and, possibly, in Stations 3 and 4). During filtration, the liquid contained in the slurry is filtered out of the port by the action of the pressure gradient generated by P101, and stored in filtrate collector V-106, while the crystals are retained on top of the filter mesh, leading to cake formation. We distinguish between actual filtration, when there is a slurry hold-up on top of the cake being formed, and the subsequent deliquoring, during which the sole remaining liquid is the one retained inside the cake pores. Upon deliquoring, the liquid in the cake pores is mechanically displaced out of the cake by the action of the pressure gradient, until a certain pore saturation equilibrium is achieved. Filtration duration depends on the cake properties and on the pressure gradient itself. Depending on filtration duration, the cake can be partially deliquored in Stations 1-3,

or it might even enter Station 4 with some slurry hold-up (drying cannot be properly conducted in this situation, which should be avoided). Thermal drying is performed in Station 4 by flowing a hot air stream through the cake.

Table 1. *Legend of Figures 1-2, including unit operations and ancillary equipment.*

Name	Description
Unit ID	
F101-F104	Filter mesh below Stations 1-4 (respectively)
P101	Compressor
E101	Drying air electrical heater
V101	Carousel Station 1
V102	Carousel Station 2
V103	Carousel Station 3
V104	Carousel Station 4
V105	Carousel Station 5
V106	Filtrate collector
V107	Slurry tank
Controllers and sensors	
AI-101	Slurry concentration indicator
FC-101	Fed slurry volume controller
FI-101	Flowmeter for drying air entering carousel ports
KIC-101	Carousel rotation controller
LI-101	Camera system measuring volume of fed slurry and cake height
PC-101	Air pressure controller
PI-101	Compressor delivery pressure indicator
PI-102	Filtrate pressure indicator
TC-101	Drying air inlet temperature controller
TI-101	Thermocouple for drying air inlet temperature
TI-102	Thermocouple for drying air outlet temperature
WI-101	Scale for inferring filtrate flowrate

Table 2. Simulator inputs.

Variable name in `run_carousel.m`	Variable	UOM	Admissible values
`control_mode`	Flag for selecting control strategy: 0: open-loop 1: closed-loop controller of sample case study from Destro et al. (2022) Other modes can be set up by the user	-	0, 1, other values set up by user
`disturbance_scenario`	Flag for selecting disturbance scenario: 0: normal operating conditions 1: slurry concentration ramp change 2: specific cake resistance step change	-	0, 1, 2
`total_duration`	Simulation duration	s	$[0, +\infty)$
`u_nominal.t_rot`	Nominal set-point cycle duration	s	$[5, +\infty)$*
`u_nominal.V_slurry`	Nominal set-point fed slurry volume	m^3	$[5\times10^{-7}, 1\times10^{-6}]$
`u_nominal.P_compr`	Nominal set-point compressor delivery pressure (gauge)	Pa_g	$[1\times10^4, 2\times10^5]$
`u_nominal.Tinlet_drying`	Nominal set-point drying air inlet temperature	K	[293, 353]
`cryst_output.conc_slurry`	Nominal slurry concentration	kg/ m^3	[50, 500]
`control_interval`	Time interval at which `controller_online.m` is called	s	Multiples of 1
`sampling_interval`	Sampling interval for measurements and states in simulation output	s	Submultiples of 1
`inter_cycle_Dt`	Idle time at the end of every cycle	s	$[0, +\infty)$*; default: 0
`mesh_clean_Dt`	Idle time at mesh cleaning	s	$[0, +\infty)$*; default: 0

***MUST BE AN INTEGER**

2 How to use the simulator

2.1 Inputs

The simulator inputs are reported in Table 2, and have to be set up in the script `run_carousel.m`.

2.2 Simulation execution

Figure 3 shows the logical structure of the simulator, while Figure 4 elucidates the set of scripts and functions that make up the simulator, together with the order and the logics with which they are called. The simulation is initialized by running the script `run_carousel.m`, after having set up the desired inputs. The simulation is carried out as shown in Figure 4. Function `run_simulation.m`, handling the simulation routine, is automatically called. The parameters of the model are automatically retrieved from function `carousel_parameters.m`, and the resistances of the filter meshes for the first 1200 processing cycles are loaded from `resistances.m` (for simulating more than 1200 processing cycles, the relevant number of additional values of filter mesh resistances have to be added to `resistances.m`).

Then, the simulation of the first processing cycle begins. The `carousel_simulator.m` function is called for simulating 1 s of carousel operation. The `estimator_online.m` function is then executed. By default, `estimator_online.m` is an empty function, but it can be modified by the user for setting up parameter and state estimation routines.

Afterwards, carousel operation is simulated again for a duration of 1 s with `carousel_simulator.m`, or, if the control interval specified by the user has been achieved, the `controller_online.m` function is called. Function `controller_online.m` is the high-level controller that updates the value of the set-points of the operating variables at every control interval and, together with function `controller_switch.m` (called at every cycle switch), forms the "High-level controller" block of Figure 3.

The set-point of the low-level controllers of the operating variables are (Figures 1-2):

- The set-point of the inlet drying air temperature $T_{in,g}^{sp}$;
- The set-point of the slurry volume fed to the carousel at every cycle V_{slurry}^{sp};
- The set-point of the cycle duration Δt_{cycle}^{sp};
- The set-point of the pressure provided by the compressor P_{compr}^{sp}.

Within the default control mode 0, the process is operated at open-loop. Therefore, the set-points of the lower-layer controllers (FQC-101, PC-101, TC-101, and KIC-101) coincide with the nominal values set by the user (Table 2). If closed-loop routines are implemented, the lower-layer controllers set-points are instead adjusted during carousel operation, based on the control laws implemented in `controller_online.m`.
The drying air inlet temperature ($T_{in,g}$), the pressure provided by the compressor (P_{compr}), and the cycle duration (Δt_{cycle}) are assumed to be perfectly controlled, namely the actual responses perfectly track the relevant setpoints. The fed slurry volume (V_{slurry}) is instead subject to Gaussian fluctuations around the set-point, to reflect the behavior of real life carousels, as outlined in Destro et al. (2022). Moreover, for cycles during which V-101 is empty due to the cleaning-in-place routine, V_{slurry} is automatically set equal to zero. Function `controller_online.m` also contains a sample closed-loop control routine (control mode 1), illustrated in the sample case study from Destro et al. (2022).

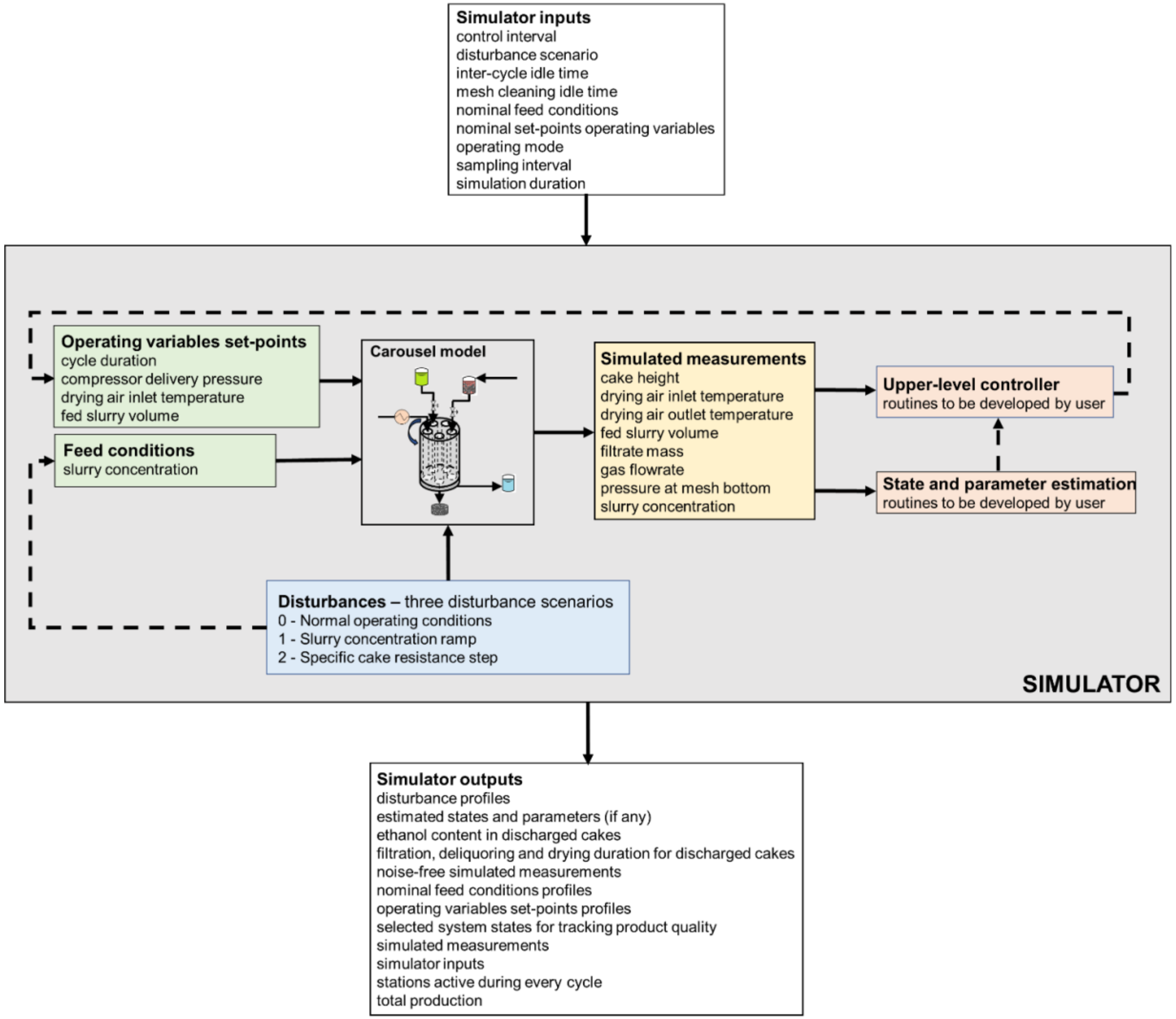

Figure 3. *ContCarSim: schematic elucidating the simulator structure. The "Carousel model" block contains the lower-level controllers of the operating variables, whose set-points are fixed at open-loop, while they are adjusted by the "Upper-level controller" block at closed-loop.*

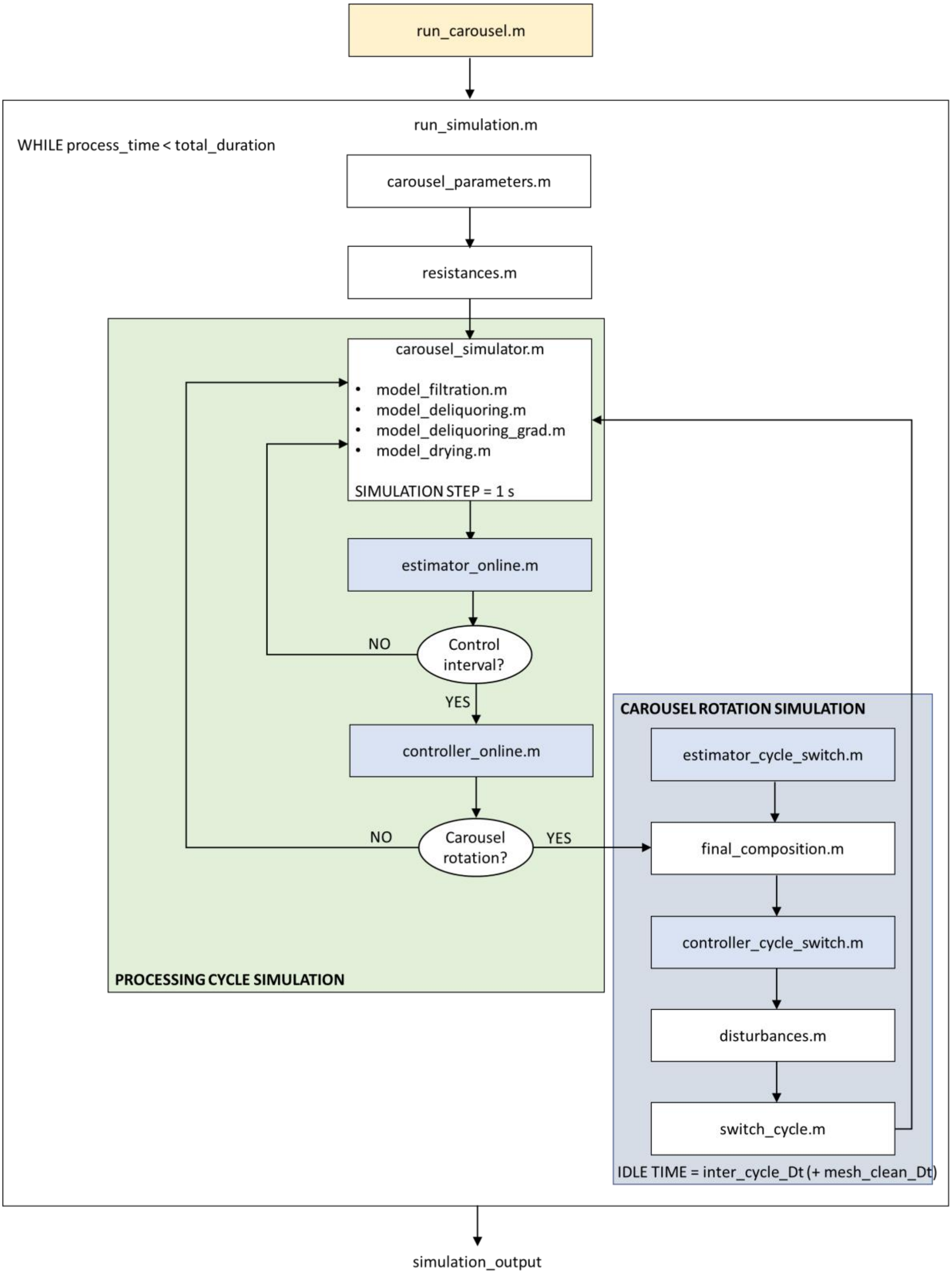

Figure 4. *ContCarSim: computational structure of the simulator, showing the logics and the order with which the functions and scripts of the simulator are called. Yellow: script to be (optionally) edited with desired operating settings and then run. Blue: functions that can be edited for modifying the control strategy. At the end of the simulation of a carousel rotation, the process time* (`process_time`) *is increased of an idle time corresponding to* `inter_cycle_Dt` *when no mesh cleaning occurs, and of* `inter_cycle_Dt+mesh_clean_Dt` *when mesh cleaning occurs. Default settings:* `inter_cycle_Dt=mesh_clean_Dt=0`.

Functions `carousel_simulator.m`, `estimator_online.m`, and `controller_online.m` are subsequently called until the cycle time reaches the current cycle duration Δt_{cycle}. At that point, the following functions are called:

- `estimator_cycle_switch.m`: by default an empty function, that can be modified by the user for setting up parameter and state estimation routines;
- `final_composition.m`: calculates the composition of the discharged cake, if there is any, and stores other variables that are outputs of the simulator;
- `controller_cycle_switch.m`: high level controller updating the set-points of the operating variables, following the control laws set up by user. The default implementation acts only on V_{slurry}^{sp}, setting it to zero for cycles in which V-101 is empty, and to the nominal set-point of fed slurry volume specified in `run_carousel.m` in all the other cases;
- `disturbances.m`: updates the value of the variability sources for the following cycle, based on the selected disturbance scenario;
- `switch_cycle.m`: handles the carousel rotation routine.

The process time (`process_time`) is then increased of an idle time corresponding to `inter_cycle_Dt` when no mesh cleaning occurs, and of `inter_cycle_Dt+mesh_clean_Dt` when mesh cleaning occurs. Simulation of a new processing cycle is then initiated. Note that, unless otherwise specified in `run_carousel.m`, `inter_cycle_Dt` and `mesh_clean_Dt` are, by default, both set to zero.

Subsequent processing cycles and carousel rotation routines are simulated, until the set total process duration is reached. At that point, the simulation is terminated, and the simulation output object (`simulation_output.mat`) is generated.

2.3 Outputs

The structure of `simulation_output.mat` is elucidated in Table 3. Note that `simulation_output.mat` is generated only if at least one cake has been discharged by the carousel, namely if a large enough total process duration has been specified, compared to the set cycle duration.

The fields of `simulation_output.mat` correspond to the simulator outputs in Figure 3:

- `states`, storing the value assumed by selected system states during the simulation;
- `measurements`, storing the process measurements collected from the sensors reported in Figures 1-2 and Table 1;
- `measurements_nf`, storing the noise-free values of the process measurements;
- `disturbances`, containing the values assumed by the variability sources during the simulation;
- `operating_vars`, storing the profiles of the set-points of the operating variables;
- `x_estim`, containing the states and parameters estimated through `estimator_online.m` and `estimator_cycle_switch.m`;
- `feed`, storing the nominal slurry concentration profile during the simulation;

Table 3. Structure of simulation_output.mat. n_cycles = total number of initiated carousel cycles.

Field	Sub-field	Sub-sub-field	Variable	Description	UOM
states	station1 station2 station3	cake_x	t	Readings of timer reinitialized at every carousel rotation – vector, step: sampling_interval	s
			S	Average cake saturation time profile for cake_x in Station 1/2/3 – vector [1×length(t)]	-
			w_EtOH_cake	Time profile of average ethanol mass fraction in cake for cake_x in Station 1/2/3 – vector [1×length(t)]	-
	station4	cake_x	t	Readings of timer reinitialized at every carousel rotation – vector – step: sampling_interval	s
			S	Average cake saturation time profile for cake_x in Station 2/3/4 – vector [1×length(t)]	-
			w_EtOH_cake	Time profile of average ethanol mass fraction in cake for cake_x in Station 2/3/4 – vector [1×length(t)]	-
			Tg_top	Temperature of drying air at top of cake in Station 4 – vector [1 × length(t)]	K
measurements			t_meas	Readings of timer initialized at process onset - vector: step = sampling_interval	s
			m_filt_WI101	Readings of WI101 – vector [1 × length(t_meas)]	kg
			P_PI101	Readings of PI101– vector [1 × length(t_meas)]	Pa_g
			P_PI102	Readings of PI102– vector [1 × length(t_meas)]	Pa_g
			c_slurry_AI101	Readings of AI101– vector [1 × length(t_meas)]	kg/m^3
			L_cake_LI101	Readings of LI101– vector [1 × length(t_meas)]	m
			V_slurry_LI101	Readings of LI101– vector [1 × length(t_meas)]	m^3
			Tg_in_TI101	Readings of TI101– vector [1 × length(t_meas)]	K
			Tg_out_TI102	Readings of TI102– vector [1 × length(t_meas)]	K
			Vdryer_FI101	Readings of FI101– vector [1 × length(t_meas)]	L/min
measurements_nf	*same structure of* measurements				
disturbances			resistances	Vector [n_cycles × 4] – element (i, j) = resistance of mesh in position j during processing cycle i, for i = 1, 2, …, n_cycles (= total number of simulated cycles)	1/m
			c_slurry	Multiplicative coefficients to nominal slurry concentration – vector [1 × n_cycles]	-
			V_slurry	Multiplicative coefficients to current fed slurry set-point – vector [1 × n_cycles]	-
			E	Multiplicative coefficients to nominal cake porosity – vector [1 × n_cycles]	-
			alpha	Multiplicative coefficients to nominal specific cake resistance – vector [1 × n_cycles]	
			hM	Multiplicative coefficients to nominal mass transfer parameter – vector [1 × n_cycles]	-
			hT	Multiplicative coefficient to nominal heat transfer parameter – vector [1 × n_cycles]	-
operating_vars			t_vector	Readings of timer initialized at process onset – vector: step = control_interval	s

		P_compr_vector	Profile of set-points of compressor delivery pressure (gauge)– vector [1 × length(t_vector)]	Pa_g
		Tin_drying_vector	Profile of set-points of drying gas temperature – vector [1 × length(t_vector)]	K
		n_cycle_vector	Initiated cycles counter - vector [1 × n_cycles]	-
		t_rot_vector	Completed cycles duration – vector [1 × number of completed cycles)]	
		V_slurry_vector	Set-point of fed slurry volume – vector [1 × n_cycles]	m^3
x_estim	*structure depends on routines set up in* estimator_online.m and estimator_cycle_switch.m			
feed		c_slurry_nom_vector	Profile of nominal slurry concentration – vector [1 × n_cycles]	kg/m^3
cakes_proc_times	cake_x	filtration_duration	Duration of filtration undergone by cake_x during carousel processing	s
		deliquoring_duration	Duration of deliquoring undergone by cake_x during carousel processing	S
		drying_duration	Duration of drying undergone by cake_x during carousel processing	s
final_content			Mass fraction of ethanol content in discharged cakes [1 x number discharged cakes]	-
active_stations			Vector [n_cycles x 4] – if port *j* processes material during cycle *i*, active_stations(*i*, *j*) = 1. Otherwise, active_stations (*i*, *j*) = 0.	-
total_production			cumulative mass of cakes of acceptable quality obtained during the simulation	kg
settings		control_mode	Scalar	-
		disturbance_scenario	Scalar	-
		control_interval	Time interval at which control routines are called - scalar	s
		sampling_interval	Sampling interval for all sensors – scalar	s
		total_duration	Simulation duration – scalar	s
		inter_cycle_Dt	Idle time at the end of every cycle	s
		mesh_clean_Dt	Idle time at mesh cleaning	s
	cryst_out_nom	conc_slurry	Nominal slurry concentration in feed – scalar	kg/m^3
		x	Crystal size distribution – particles diameters	m
		CSD_perc	Volumetric crystal size distribution	%
		T	Inlet slurry temperature (equal to room temperature)	K
	u_nom	t_rot	Nominal set-point cycle duration – scalar	s
		V_slurry	Nominal set-point fed slurry volume - scalar	m^3
		P_compr	Nominal set-point pressure provided by compressor (gauge) – scalar	Pa_g
		Tinlet_drying	Nominal set-point inlet drying gas temperature - scalar	K

- `cakes_proc_times`, reporting the filtration, deliquoring and drying duration undergone by all the cakes discharged from the carousel;
- `final_content`, listing the ethanol mass fraction in all the discharged cakes;
- `active_stations`, summarizing which stations where active during which carousel cycle (as outlined in Section 1, certain stations are empty during carousel operation, due to the cleaning-in-place routine);
- `settings`, containing the settings that were specified in `run_carousel.m` before initiating the simulation.
- `total_production`, cumulative mass of cakes of acceptable quality obtained during the simulation

Of all the simulation outputs, the only ones available in physical carousels are those contained in the `measurements` **field**.

To retrieve the actual value of the variables affected by disturbances from the simulation output:

- **fed slurry concentration**: the actual concentrations of the slurries fed to the carousel in all cycles are obtained from the product of `simulation_output.disturbances.c_slurry` with `simulation_output.feed.c_slurry_nom_vector`;
- **drying kinetic parameter and heat transfer coefficient** between cake and air during drying: the nominal values of the parameters vary during every cycle, following the drying model presented in [1], and are not provided in the simulation output. The multiplicative coefficients acting as variability source, varying at each cycle, are accessible from `simulation_output.disturbances`;
- **fed slurry volume**: the actual slurry volumes fed to the carousel in all cycles are obtained from the product of `simulation_output.operating_vars.V_slurry_vector` with `simulation_output.disturbances.V_slurry`;
- **specific cake resistance**: the actual resistances of all the processed cakes are calculated bymultiplying `simulation_output.disturbances.alpha` by the nominal value set in `carousel_parameters.m`;
- **cake porosity**: the actual porosities of all the processed cakes are calculated multiplying `simulation_output.disturbances.E` by the nominal value set in `carousel_parameters.m`.

2.4 Setting up a control strategy

Control strategies can be set up by the user modifying one or more of the following functions:

- `controller_cycle_switch.m`
- `controller_online.m`
- `estimator_cycle_switch.m`
- `estimator_online.m`

The input/output structure is thoroughly documented in each function. Although the functions can be freely modified, the bottom part of `controller_cycle_switch.m` and `controller_online.m`, where the updated values of the set-points of the operating variables are stored and a null slurry volume for cycles in which Station 1 is empty, should not be edited.

Note that, among all the simulation outputs, **only the process measurements are available for state and parameter estimation routines** implemented in `estimator_cycle_switch.m` and `estimator_online.m`. At the same time, **only the process measurements and potential estimated states and parameters for control routines** implemented in `controller_cycle_switch.m` and `controller_online.m`.

The default control strategies implemented in the simulator are:

- control mode = 0: open-loop operation, no estimated parameters/states;
- control mode = 1: automatic adjustment of the cycle duration, no estimated parameters/states;

2.5 Simulator scripts and functions

2.5.1 Summary

`run_carousel.m`	Script for initiating carousel simulation
`run_simulation.m`	Function handling carousel simulation schedule
`carousel_parameters.m`	Function containing simulation and model parameters
`resistances.mat`	Matrix containing the values of the filter mesh resistances for the first 1200 processing cycles (size = 1200 × 4).
`carousel_simulator.m`	Function simulating carousel operation using filtration, deliquoring and drying models
`estimator_online.m`	Function that can be written by the user for online state/parameter estimation
`controller_online.m`	Function that can be written by the user, containing online control routines. Together with `controller_switch.m`, forms the “High-level controller” block of Figure 3.
`estimator_cycle_switch.m`	Function that can be written by the user for state/parameter estimation routines to be executed at every carousel rotation
`final_composition.m`	Function executed at the end of every cycle to calculate the composition of the discharged cake, if there is any, and for storing the value of other variables contained in the simulation output
`controller_cycle_switch.m`	Function that can be written by the user, containing control routines to be executed at every carousel rotation. Together with `controller_online.m`, forms the “High-level controller” block of Figure 3.
`disturbances.m`	Function that sets the value of variability sources #1-10 for the following cycle (e.g., filter mesh resistance, Gaussian fluctuations, …), based on the disturbance scenario

`switch_cycle.m`	Function containing carousel rotation simulation routines, such as material transfer from one port to the following one
`model_filtration.m`	Function simulating filtration (ODE model)
`model_deliquoring.m`	Function simulating deliquoring with design charts (approximate method called when cake is very small, i.e. with height below 0.3 mm)
`model_deliquoring_grad.m`	Function simulating deliquoring (PDE model)
`model_drying.m`	Function simulating drying (PDE model)

2.5.2 Which scripts and functions can be edited?

In principle, all the functions and scripts in the simulator are openly accessible and can be edited, including all the models and the routine calling the models and the sub-functions for handling the carousel simulation (`run_simulation.m`). However, numerical stability and physical consistency are guaranteed only when properly editing the following files, which have been conceived and thoroughly commented for this purpose:

- Script to edit and run for starting the simulation with the desired settings:

  `run_carousel.m`

- Functions to edit for changing the control strategy:

  `controller_cycle_switch.m`, `controller_online.m`

  `estimator_cycle_switch.m`, `estimator_online.m`

- Function to edit for modifying the physical properties of the system:
  `carousel_parameters.m`

- Files to edit for implementing additional disturbance scenarios or to modify the existing ones:

  `disturbances.m` and `resistances.m`

References

[1] Destro, F., Nagy, Z. K. and M. Barolo (2022). A benchmark simulator for quality-by-design and quality-by-control studies in continuous pharmaceutical manufacturing – Intensified filtration-drying of paracetamol/ethanol slurries. *Comput. Chem. Eng.*, **163**, 107809.

Appendix

The simulator inputs used for generating the data of the sample case study in Destro et al. (2022) are reported in Table 5. The control strategies described in the case study are coded in the default `controller_cycle_switch.m` and `controller_online.m` functions of the simulator.

Table 4. *Case study: simulator inputs set in* `run_carousel.m` *for obtaining the results of the sample case study presented in Destro et al. (2022)*

Variable name in `run_carousel.m`	**Value**	**UOM**
`control_mode`	Control strategies #1,2: 0 Control strategy #3: 1	-
`disturbance_scenario`	0, 1, 2 (*every control strategy tested in every disturbance scenario*)	-
`total_duration`	1800	s
`u_nominal.t_rot`	Control strategies #1,3: 30 Control strategy #2: 45	s
`u_nominal.V_slurry`	3.0×10^{-6}	m^3
`u_nominal.P_compr`	1.0×10^5	Pa_g
`u_nominal.Tinlet_drying`	323.0	K
`cryst_output.conc_slurry`	250.0	kg/m^3
`control_interval`	1	s
`sampling_interval`	0.1	s
`inter_cycle_Dt`	0	s
`mesh_clean_Dt`	0	s