

YEASTsim: a Matlab-based simulator for the study of process control of fed-batch yeast fermentations

USER'S GUIDE to YEASTsim v1.62

Version 1.0

Pavel Hrnčířík, Jan Kohout

Department of Mathematics, Informatics and Cybernetics

University of Chemistry and Technology Prague

Table of contents

1	Introduction.....	3
2	Process description.....	3
3	Mathematical model of fed-batch yeast fermentation	4
3.1	Biomass growth kinetics model.....	4
3.2	Mass transfer model.....	7
4	Process quality indicators.....	8
5	Process control strategies	9
5.1	Rule-based control strategy using the dissolved oxygen concentration (DOCONT).....	9
5.2	Feedforward control strategy using biomass concentration estimates by software sensors	10
6	Process simulator (YEASTsim MATLAB app).....	11
6.1	Introduction.....	11
6.2	Installing the YEASTsim MATLAB app.....	13
6.3	Running the YEASTsim app.....	13
6.4	Adding user's own control strategies to the simulator	14
6.5	Using the simulator for advanced users (without the YEASTsim MATLAB App GUI).....	14
7	References.....	15
	Appendix A: controller_function_template m-function code.....	16
	Appendix B: RUN_YEAST_SIMCON.m script code	18
	Appendix C: Structure of the exported YeastSim data structure	20

1 Introduction

This document should serve as a basic guide for users of the YEASTsim simulator. In the introductory part including sections 2 and 3, the basic characteristics of the modelled cultivation process and the basic features of the mathematical model used in the simulator are presented. The following sections 4 and 5 describe the main process quality indicators used for quality evaluation from the point of view of the production goals of the simulated bioprocess, as well as a set of control strategies that are included in the basic installation of the simulator. The second part of the document (section 6) is devoted to practical instructions for using the simulator, from the basic installation of the simulator in the form of MATLAB app to recommendations for advanced users regarding the use of the simulator without the MATLAB app GUI. Selected code examples for creating your own control strategies and embedding the simulator into your own MATLAB scripts are presented in the appendices of the document.

2 Process description

The software simulator was created for the fed-batch aerobic cultivation process of the *Saccharomyces cerevisiae* yeast, *i.e.* baker's yeast. This process is one of the most common biotechnological processes, with a wide range of applications from the food industry to the pharmaceutical industry. In the bioprocess control laboratory at UCT Prague, this process is operated on a laboratory scale (7.5-liter bioreactor), e.g. for the purpose of experimental testing and development of various control strategies for fed-batch aerobic cultivation processes. The basic scheme of the process is shown in Figure 1 with an overview of all the process variables. The specified variables are divided into 3 groups:

- State variables – basic process variables that describe the immediate state of the cultivation process within the bioreactor (molar concentration of individual reaction components and volume of liquid in the bioreactor). These variables are considered in the mathematical model of the process that forms the basis of the simulator, but they are not measured directly in the laboratory bioreactor in the online mode.
- Manipulated variables – variables used for control of the cultivation process both during experiments in the bioreactor and in the software simulator. In general, there are three manipulated variables: feed and air flowrates, stirrer speed. Their values are calculated using individual control strategies.
- Output variables (measured) - variables that are directly measured using sensors in the laboratory bioreactor in online mode. In the software simulator, their values are calculated from the values of the state variables.

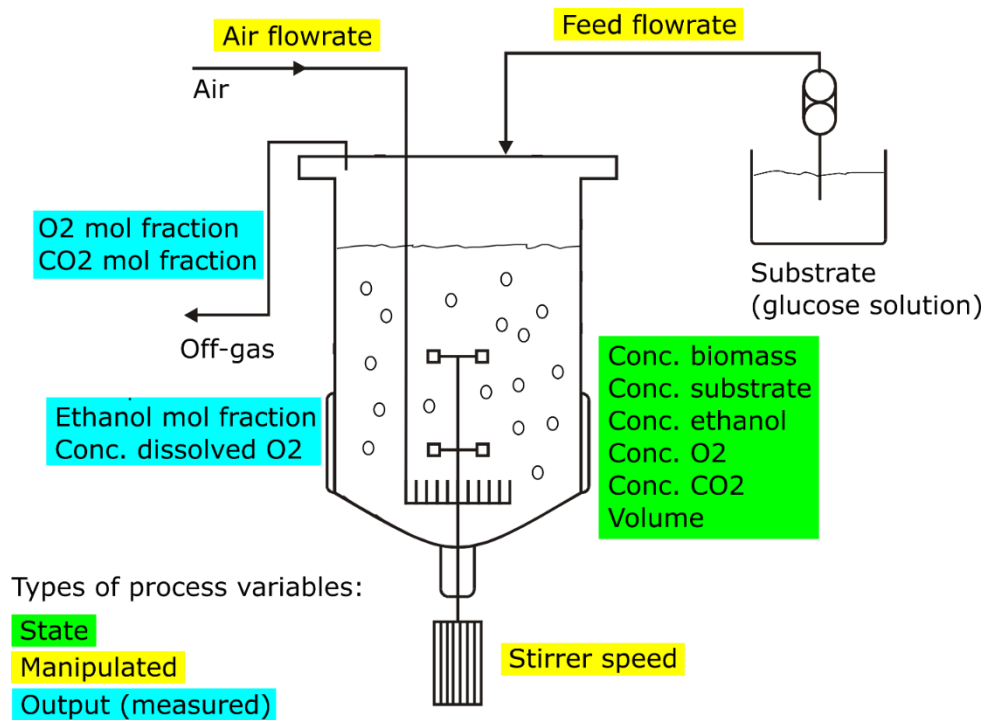


Figure 1 Process scheme

3 Mathematical model of fed-batch yeast fermentation

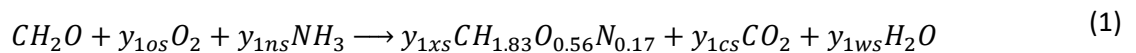
The mathematical model of the fed-batch cultivation of baker's yeast (*Saccharomyces cerevisiae*) consists of two separate parts:

- model describing biomass growth kinetics,
- model that describes mass transfer between the liquid and gas phase in the bioreactor.

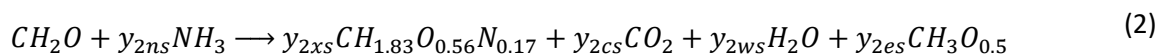
3.1 Biomass growth kinetics model

This sub-model assumes that the yeast *Saccharomyces cerevisiae* grows on the two main substrates of glucose and ethanol using three different metabolic pathways: Eqs. (1, 2) for growth on glucose and eq. (3) for growth on ethanol. The kinetic model used is based on the well-known theory of limited respiratory capacity developed by (Sonnleitner and Käppeli, 1986) and (Sonnleitner and Hahnemann, 1994).

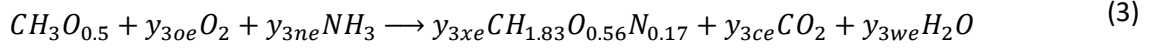
Oxidative growth on glucose (reaction 1)



Reductive growth on glucose (reaction 2)



Oxidative growth on ethanol (reaction 3)



Furthermore, it is assumed that biomass growth is governed by Monod kinetics, *i.e.* substrate uptake is described by Eq. (4).

$$q_i = q_{i,max} \cdot \frac{c_i}{ak_i + c_i} \quad i = s, e, o \quad (4)$$

However, eq. (4) by itself is not sufficient to determine the actual consumption rates of individual substrates in reactions 1–3, because these consumptions are ultimately determined by the currently limiting substrate. The actual rates of consumption of substrates in individual metabolic pathways are thus finally defined as follows:

Oxidative growth on glucose - reaction 1

$$q_{s1} = \min(q_o/y_{1os}, q_s) \quad (5)$$

$$r_{s1} = -q_{s1} \cdot c_x \quad (6)$$

$$r_{x1} = y_{1xs} \cdot r_{s1} \quad (7)$$

$$r_{o1} = y_{1os} \cdot r_{s1} \text{ where } y_{1os} = 1 + 1.05y_{1xs} \quad (8)$$

$$r_{c1} = y_{1cs} \cdot r_{s1} \text{ where } y_{1cs} = -1 - y_{1xs} \quad (9)$$

Reductive growth on glucose - reaction (2)

$$q_{s2} = q_s - q_{s1} \quad (10)$$

$$r_{s2} = -q_{s2} \cdot c_x \quad (11)$$

$$r_{x2} = y_{2xs} \cdot r_{s2} \quad (12)$$

$$r_{e2} = y_{2es} \cdot r_{s2} \text{ where } y_{2es} = -2/3 - 0.7y_{2xs} \quad (13)$$

$$r_{c2} = y_{2cs} \cdot r_{s2} \text{ where } y_{2cs} = -1/3 - 0.3y_{2xs} \quad (14)$$

Oxidative growth on ethanol - reaction (3)

$$q_{e3} = \min((q_o - y_{1os} \cdot q_{s1})/y_{3oe}, q_e) \quad (15)$$

$$r_{e3} = -q_{e3} \cdot c_x \quad (16)$$

$$r_{x3} = y_{3xe} \cdot r_{e3} \quad (17)$$

$$r_{o3} = y_{3oe} \cdot r_{e3} \text{ where } y_{3oe} = 1.5 + 1.05y_{3xe} \quad (18)$$

$$r_{c3} = y_{3ce} \cdot r_{e3} \text{ where } y_{3ce} = -1 - y_{3xe} \quad (19)$$

The total rates of production or consumption for all reactants can then be defined as the sums of the corresponding individual rates in each of the three metabolic pathways:

$$r_x = r_{x1} + r_{x2} + r_{x3} \quad (20)$$

$$r_s = r_{s1} + r_{s2} \quad (21)$$

$$r_e = r_{e2} + r_{e3} \quad (22)$$

$$r_o = r_{o1} + r_{o3} \quad (23)$$

$$r_c = r_{c1} + r_{c2} + r_{c3} \quad (24)$$

Based on eqs. (5) and (15), three different alternative growth models can be defined as a result, each with its own growth kinetics and physiological interpretation. Eqs. (5) and (15) thus control the switching between these 3 alternative models during the simulation calculations.

Model 1

$$q_o/y_{1os} < q_s$$

Oxygen is the main limiting substrate, and therefore the respiratory capacity of biomass is fully utilized for the consumption of glucose through the oxidative metabolic pathway (1). Any remaining glucose is then consumed by a reductive metabolic pathway (2). The entire biomass capacity is used for glucose processing and therefore the third metabolic pathway (3) is not active in this situation. The rates of consumption of individual substrates are then defined as follows:

$$q_{s1} = \frac{1}{y_{1os}} \cdot q_{o,max} \cdot \frac{c_o}{ak_o + c_o} \quad (25)$$

$$q_{s2} = q_{s,max} \cdot \frac{c_s}{ak_s + c_s} - q_{s1} \quad (26)$$

$$q_{e3} = 0 \quad (27)$$

Model 2

$$q_o/y_{1os} \geq q_s \text{ and } (q_o - y_{1os} \cdot q_{s1})/y_{3oe} < q_e$$

The main limiting substrate is glucose, and the respiratory capacity of biomass is fully utilized for both glucose and ethanol consumption via oxidative metabolic pathways (1) and (3). The metabolic pathway (2) is not active because all glucose is consumed oxidatively (1). The rates of consumption of individual substrates are then defined as follows:

$$q_{s1} = q_{s,max} \cdot \frac{c_s}{ak_s + c_s} \quad (28)$$

$$q_{s2} = 0 \quad (29)$$

$$q_{e3} = \left(q_{o,max} \cdot \frac{c_o}{ak_o + c_o} - y_{1os} \cdot q_{s1} \right) / y_{3oe} \quad (30)$$

Model 3

$$q_o/y_{1os} \geq q_s \text{ and } (q_o - y_{1os} \cdot q_{s1})/y_{3oe} \geq q_e$$

Glucose is again the main limiting substrate, but the respiratory capacity of biomass is only partially used for the consumption of glucose and ethanol through oxidative metabolic pathways (1) and (3). The metabolic pathway (2) is no longer active because all glucose is consumed oxidatively (1). The rates of consumption of individual substrates are then defined as follows:

$$q_{s1} = q_{s,max} \cdot \frac{c_s}{ak_s + c_s} \quad (31)$$

$$q_{s2} = 0 \quad (32)$$

$$q_{e3} = q_{e,max} \cdot \frac{c_e}{ak_e + c_e} \quad (33)$$

3.2 Mass transfer model

The mass transfer model describes the exchange of mass between the gas and liquid phases. It is assumed that oxygen, carbon dioxide, and ethanol are the only components that can be transported between the two phases and that these are ideally mixed (Kristiansen, 1994).

Equilibrium liquid - outlet gas is given by Eq. (34) as:

$$(c_{jw}/c_L) \cdot H_j = (c_{jog}/c_G) \cdot p_{atm} \quad j = e, o, c \quad (34)$$

From the pseudostationary balance in gas (Eq. 35) and the liquid - outlet gas equilibrium (Eq. 34) the equilibrium concentrations in the liquid phase (Eq. 36) and consequently, the outlet gas concentrations (Eq. 37) can be derived.

$$V \cdot K_L a \cdot (c_{jw} - c_j) = F_{Air} \cdot (c_{jig} - c_{jog}) \quad j = e, o, c \quad (35)$$

$$c_{jw} = (F_{Air} \cdot c_{jig}/V + K_L a \cdot c_j) / (F_{Air} \cdot c_G \cdot H_j / (V \cdot c_L \cdot p_{atm}) + K_L a) \quad j = e, o, c \quad (36)$$

$$c_{jog} = c_{jw} \cdot c_G \cdot H_j / (c_L \cdot p_{atm}) \quad j = e, o, c \quad (37)$$

For stirred bioreactors, the volumetric mass transfer coefficient $K_L a$, which is generally considered to be dependent on the velocity of the aeration gas V_{gas} (Eq. 38) and the mechanical agitation power P (Eq. 39), can be approximated by Eq. (40).

$$V_{gas} = \frac{4F_{Air}}{3600D_r^2} \quad (38)$$

$$P = P_{no} \cdot \rho \cdot \left(\frac{N}{3600} \right)^3 \cdot D_s^5 \quad (39)$$

$$K_L a \cong 3600 \cdot \left(0.026 \cdot \left(\frac{P}{V} \right)^{0.4} \cdot V_{gas}^{0.5} \right) \quad (40)$$

By combining the biomass growth and mass transfer models, we finally obtain a set of ordinary differential equations for the gaseous and nongaseous reactants, as well as the volume of liquid, which ultimately form the overall process model (Equations 41 to 43).

$$\frac{dc_j}{dt} = r_j + K_L a \cdot (c_{jw} - c_j) + \frac{Feed}{V} \cdot (c_{ji} - c_j) \quad j = e, o, c \quad (41)$$

$$\frac{dc_j}{dt} = r_j + \frac{Feed}{V} \cdot (c_{ji} - c_j) \quad j = x, s \quad (42)$$

$$\frac{dV}{dt} = Feed \quad (43)$$

Table 1: The numerical values of the model parameters

<i>Parameter</i>	<i>Value</i>	<i>Parameter</i>	<i>Value</i>
C_L	55500 mol.m ⁻³	C_{oig}	8.43 mol.m ⁻³
C_G	40.4 mol.m ⁻³	C_{cig}	0.004 mol.m ⁻³
H_e	100 MPa	$q_{s,max}$	2.94 h ⁻¹
H_o	4810 MPa	$q_{e,max}$	0.22 h ⁻¹
H_c	188 MPa	$q_{o,max}$	0.20 h ⁻¹
D_r	0.152 m	ak_s	5.67 mol.m ⁻³
D_s	0.07 m	ak_e	10.87 mol.m ⁻³
P_{no}	5.5	ak_o	0.0094 mol.m ⁻³
ρ	1000 kg.m ⁻³	y_{1xs}	-0.584 mol X/mol S
P_{atm}	101.325 kPa	y_{2xs}	-0.083 mol X/mol S
C_{xi}	0 mol.m ⁻³	y_{3xe}	-0.658 mol X/mol E
C_{si}	9500 mol.m ⁻³	M_e	23 g.mol ⁻¹
C_{ei}	0 mol.m ⁻³	ρ_e	781 kg.m ⁻³
C_{oi}	0.22 mol.m ⁻³	V_f	7.5 dm ³
C_{ci}	0 mol.m ⁻³	t_f	10 h
C_{eig}	0 mol.m ⁻³		

4 Process quality indicators

$Y_{x/s}$ - biomass yield per substrate (g of biomass produced per g substrate consumed):

$$Y_{x/s} = \frac{(c_{x,end} \cdot V_{end} - c_{x,start} \cdot V_{start}) \cdot M_x}{V_S \cdot S_0} \quad (44)$$

$Prod$ – bioprocess productivity (g biomass produced per litre of bioreactor volume per hour):

$$Prod = \frac{(c_{x,end} - c_{x,start}) \cdot M_x}{t_f} \quad (45)$$

where $c_{x,end}$ is the final biomass concentration in the bioreactor, V_{end} is the final volume of the cultivation broth in the bioreactor, $c_{x,start}$ is the initial biomass concentration in the bioreactor, V_{start} is the volume of the initial cultivation broth in the bioreactor, M_x is the molar mass of biomass, V_s is the total volume of the substrate solution supplied to the bioreactor over the entire cultivation duration, S_0 is the substrate concentration in the substrate solution and t_f is the total cultivation time.

The operational goal of the cultivation process is to maximise the values of both indicators. However, it should be considered that increasing the value of one indicator leads to a decrease in the value of the other indicator. For this reason, it is necessary to choose a suitable compromise or trade-off between the two possible extremes according to the specific production requirements, typically determined by the costs of substrates, products, etc.

5 Process control strategies

The simulator installation includes two control strategies for the cultivation process. In both cases, these are strategies aimed at influencing the growth of yeast biomass by adjusting the substrate feedrate to the bioreactor. The first strategy is a rule-based control strategy, which is based on continuous adjustments of the substrate feedrate according to changes in the dissolved oxygen concentration in the bioreactor. The second type of expert control strategy is based on continuous estimation of the amount of yeast biomass in the bioreactor using software sensors and subsequent calculation of the corresponding substrate feedrate. Additional control strategies can be created and added to the simulator according to the user's needs and preferences (see section 6.4).

5.1 Rule-based control strategy using the dissolved oxygen concentration (DOCONT)

The DOCONT strategy represents an example of direct expert process control in the form of a rule-based strategy that adjusts the substrate feedrate according to changes in the concentration of dissolved oxygen in the bioreactor (DO). At the beginning of the cultivation process, the strategy begins to feed the substrate into the bioreactor with a suitably set initial flow rate. The feedrate is then continuously increased during cultivation whenever the change in the dissolved oxygen concentration value during 1 control period (set at 1 min) exceeds a suitably set threshold value. A positive change in the concentration of dissolved oxygen (at a constant level of aeration) namely indicates that the current level of feeding is no longer sufficient with respect to the current amount of biomass, and it would be appropriate to adequately increase the feed flowrate. This increase can take place according to different scenarios. In the current version of the simulator, there are a total of 4 different variants of the DOCONT strategy (denoted as v1, v2, v3, v4), differing in the method of calculating the feedrate increase, or in the method of determining the threshold values for activating the increase, respectively:

DOCONT strategy

if $DO_change_i > k_1 \cdot DO_change_threshold$ then

$$Feed_i = k_2 \cdot Feed_{i-1} + k_3 \cdot Feed_increment$$

end

Table 2: DOCONT strategy variants

DOCONT strategy variant	k_1	k_2	k_3
v1	1	1	1
v2	1	1	$DO_change_i/DO_change_threshold$
v3	0.5	1	$DO_change_i/DO_change_threshold$
v4	1	1.1	0

5.2 Feedforward control strategy using biomass concentration estimates by software sensors

In the case of this control strategy, the feedrate of the substrate to the bioreactor is continuously calculated according to Eq. (46):

$$Feed(t) = \frac{k_{desired} \cdot c_{x,est}(t) \cdot V(t)}{Y_{X/S,desired} \cdot S_0} \quad (46)$$

where t is the cultivation time, $Feed(t)$ is the substrate feedrate to the bioreactor at the current cultivation time, $k_{desired}$ is the desired biomass growth rate, $c_{x,est}(t)$ is the current biomass concentration estimated using a software sensor, $V(t)$ is the current volume of the cultivation broth in the bioreactor, $Y_{X/S,desired}$ is the desired biomass yield per substrate and S_0 is the glucose concentration in the substrate solution fed to the bioreactor.

A key input for this strategy is the instantaneous concentration of yeast biomass in the bioreactor, which is a process variable that is not normally measurable online under operating conditions. For this reason, this strategy uses so-called software sensors, which estimate the instantaneous biomass concentration from commonly online measurable quantities, typically the concentrations of CO_2 and O_2 in the gas leaving the bioreactor. This simulator includes a total of 2 software sensors to estimate biomass concentration. The first sensor is based on estimating the biomass concentration according to the overall stoichiometric equation of biomass growth and uses as values the values of instantaneous rates of O_2 consumption, CO_2 production and substrate feedrate to the bioreactor assuming negligible substrate accumulation in the bioreactor, see Eq. (47):

$$\Delta x_{est}(t + \Delta t) = \Delta t \cdot (k_{O_2} \cdot OUR(t) + k_{CO_2} \cdot CPR(t) + k_{feed} \cdot Feed(t) \cdot S_0) \quad (47)$$

where $\Delta x_{est}(t+\Delta t)$ is the increase in the amount of biomass during the Δt period, $OUR(t)$, $CPR(t)$, $Feed(t)$, S_0 are the instantaneous rates of O_2 consumption, CO_2 production and substrate feedrate to the bioreactor and glucose concentration, respectively, and k_{O_2} , k_{CO_2} , k_{feed} are calibration constants related to the the overall stoichiometric equation of biomass growth. (Johnson, 1987)

The second software sensor, on the other hand, is based on the assumption of a linear relationship between the increase in biomass in the bioreactor since the beginning of cultivation and the amount of oxygen consumed over the same period; see Eq. (48):

$$\Delta x_{est}(t) = k_{cal} \cdot \int_{t_{start}}^t OUR(\tau) d\tau \quad (48)$$

where $\Delta x_{est}(t)$ is the increase in the amount of biomass from the beginning of cultivation (time t_{start}) to the current time t , OUR is the rate of O_2 consumption, and k_{cal} is a calibration constant determined by simple linear regression from historical process data. (Hrnčířík, 2021)

For their correct operation, both software sensors need an estimate of the initial biomass concentration at the beginning of cultivation after inoculation. In the simulator, this estimate is realized by a calculation based on the simulation of the uncertainty of the estimation of the initial amount of biomass with an error of $\pm 50\%$ (using a random number generator) compared to the actual real value of the biomass concentration.

6 Process simulator (YEASTsim MATLAB app)

6.1 Introduction

The software simulator for the fed-batch baker yeast fermentation process was created in the MATLAB environment as a modular application in the form of a set of MATLAB user-defined functions (see Figure 2). Thanks to this modular structure, the simulator can be used in two basic ways. Less advanced users can run the simulator through a GUI created in the form of a MATLAB app. This GUI is made up of the simulator display with basic controls. Advanced users can run the simulator directly through the MATLAB user function *yeast_simcon*, either from the MATLAB command line or by embedding this function into their MATLAB script or app. Advanced users can also easily expand the simulator with other modules of controllers or soft sensors in the form of their own MATLAB user functions.

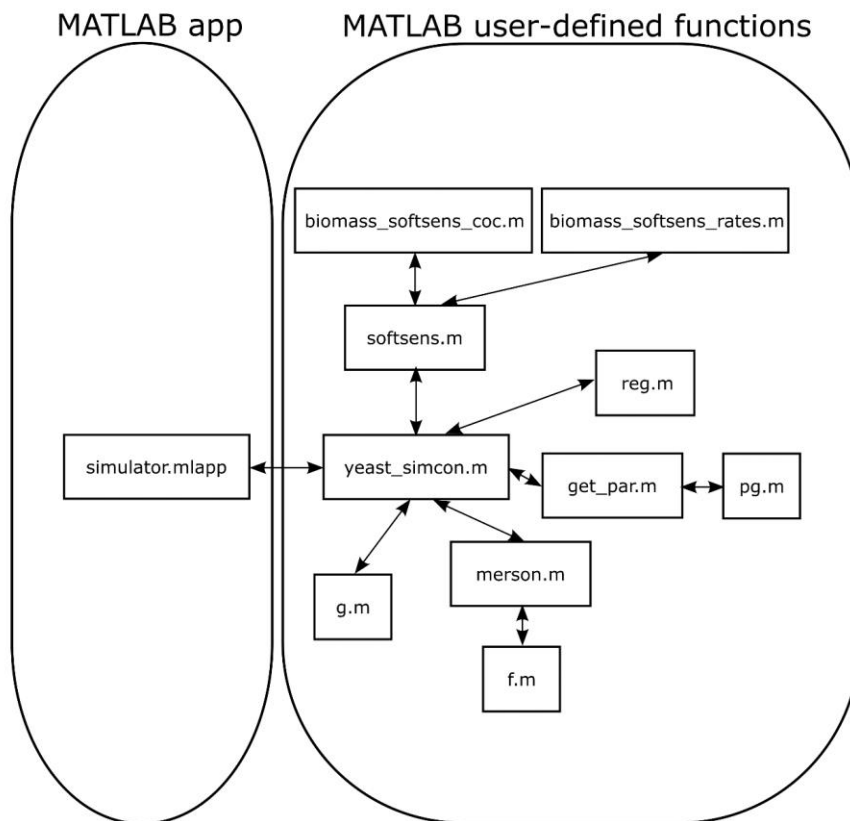


Figure 2 The YEASTsim simulator structure

Regardless of which of the above-described methods the simulator is started with, the actual simulation calculation always takes place according to the diagram in Figure 3. The connection

between individual simulation steps and their corresponding user-defined MATLAB functions is described in Table 3.

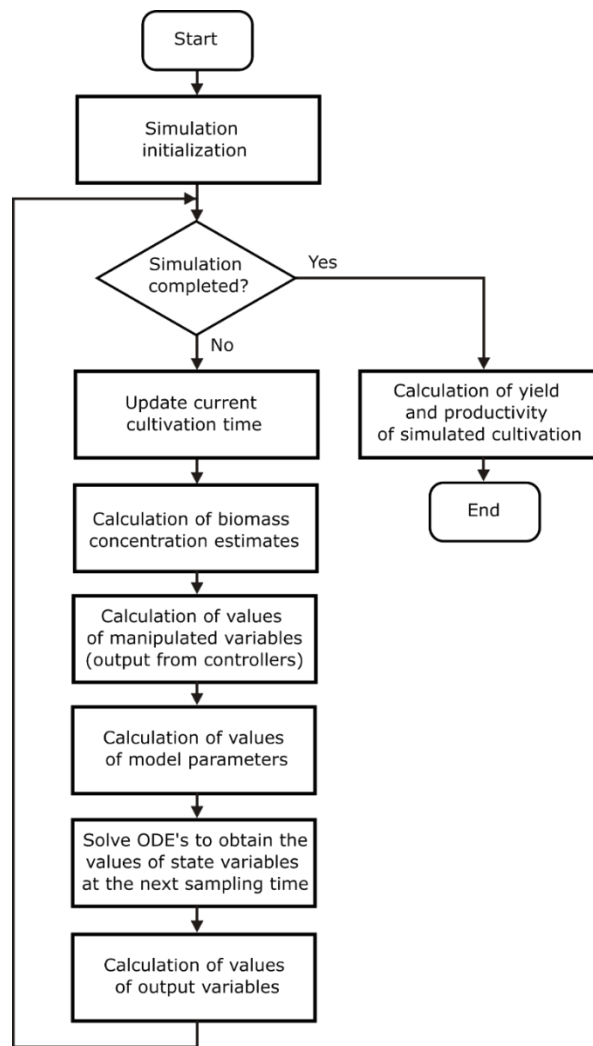


Figure 3 Flowchart of simulation steps

Table 3: List of YEASTSim functions

Function	Task
<i>yeast_simcon.m</i>	Main function of the simulator
<i>softsens.m</i>	Calculation of biomass concentration estimates
<i>biomass_softsens_???.m</i>	Calculation of biomass concentration estimates (specific softsensors)
<i>reg.m</i>	Calculation of the values of manipulated variables (specific controllers)
<i>get_par.m, pg.m</i>	Calculation of values of model parameters
<i>merson.m</i>	ODE solver (Merson's modification of the Runge-Kutta method)
<i>f.m</i>	Calculation of ODE's right-hand sides
<i>g.m</i>	Calculation of the values of output variables

6.2 Installing the YEASTsim MATLAB app

Installing the YEASTsim MATLAB app is very simple and can be done by following the steps below:

1. Copy the YEASTsim installation zip archive to the folder where you wish to have the application located.
2. Unzip the YEASTsim installation zip archive.
3. Start MATLAB¹.
4. Change the current folder to the main folder where the YEASTsim simulator is installed (or optionally add this folder to the MATLAB Search Path via Set Path in the Environment section on the Home tab).
5. Now you are ready to run the YEASTsim simulator app by entering the command `run YEASTsim.mlapp` or `run('YEASTsim.mlapp')` in the MATLAB command window.

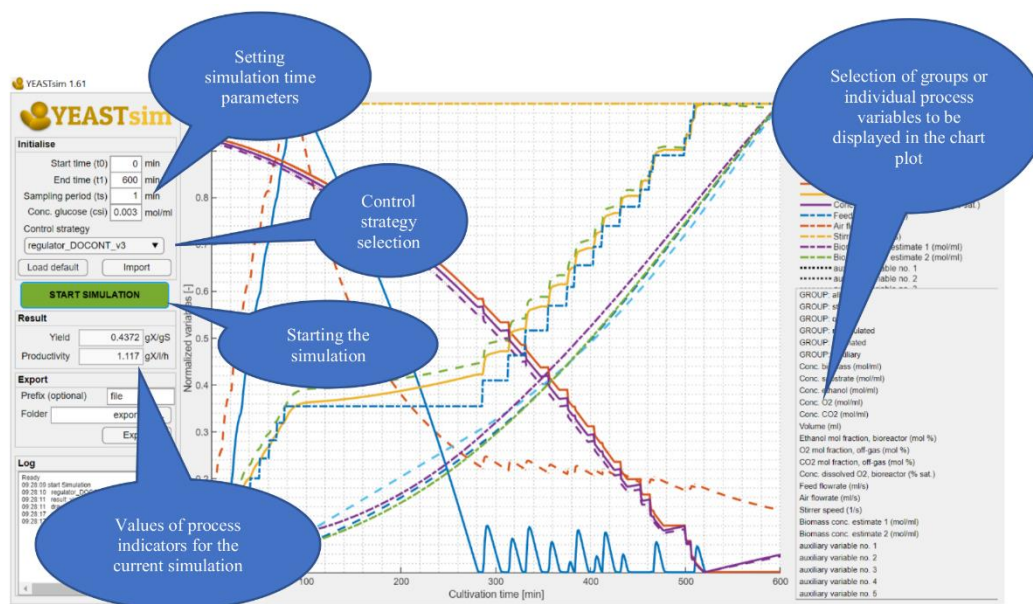
6.3 Running the YEASTsim app

The basic overview orientation of the GUI control elements of the YEASTsim app is shown in Figure 4 with labels for individual control elements. Before starting the simulation, the values of the basic parameters must be set in the left part of the window (start time, end time, sampling period, glucose concentration in the concentration of glucose in the nutrient solution, *i.e.* feed). Their preset default values correspond to standard laboratory cultivations carried out in a 7.5-l fermenter. Furthermore, a specific strategy for the control of simulated cultivation must be selected from the list of available control strategies (see section 5 for the description of the available control strategies). The simulation can then be started using the green "START SIMULATION" button.

After all calculations are completed, the values of both main process indicators (yield, productivity, see section 4) characterizing the simulated cultivation will be displayed in the left part of the window. At the same time, the graph in the middle of the window displays the time courses of all process variables for the given cultivation. Subsequently, the selective display of process variables can be set by groups or individually (item lists on the right). If multiple variables are shown together in the graph, then the y-axis has a normalized scale (range from 0 to 1). If only 1 variable is displayed in the graph, then its actual range (from min to max value) is displayed on the y-axis.

All data can also be exported to a MATLAB .mat data file ("Export" button) and then reloaded into the simulator as needed ("Import" button). The resulting .mat file with the data can be loaded into Matlab even without using the simulator. The data is stored in a data structure whose description is in Appendix C.

¹ YEASTsim app GUI requires MATLAB version 2016 and later. If you have an older version of MATLAB, then the simulator can be run without the GUI from a script (see section 6.5 for more details).



6.4 Adding user's own control strategies to the simulator

The YEASTsim simulator is designed to be easily extended to include other cultivation control strategies. To create new strategies, the user should use a template in the form of the *controller_function_template* m-function, which is stored in the *templates* folder of the simulator installation. You can also find the *controller_function_template* m-function code in Appendix A of this manual. The template contains a detailed description of all parts of the function, which makes it relatively easy for the user to implement their own cultivation control strategy. The resulting m-function file containing the user control strategy must then be located in the *regs* folder of the simulator installation and the simulator must be restarted before using the newly added control strategy.

6.5 Using the simulator for advanced users (without the YEASTsim MATLAB App GUI)

The YEASTsim simulator was created as a modular application in the form of a set of user-defined MATLAB functions (see Figure 2). Due to this modular structure, the simulator can be used not only through the GUI (*i.e.* YEASTsim MATLAB App), but advanced users can run the simulator directly through the MATLAB user function *yeast_simcon*, either from the MATLAB command line or by embedding this function in their script or MATLAB application. The *RUN_YEAST_SIMCON.m* sample script located in the scripts folder of the simulator installation can be used as a possible template for this purpose. You can also find the *RUN_YEAST_SIMCON.m* script code in Appendix B of this manual.

7 References

- Hrnčířík, P., 2021. Software Sensors for the Monitoring of Bioprocesses. Adv. Intell. Syst. Comput. 1268 AISC, 207–215. https://doi.org/10.1007/978-3-030-57802-2_20/COVER
- Johnson, A., 1987. The control of fed-batch fermentation processes—A survey. Automatica 23, 691–705. [https://doi.org/10.1016/0005-1098\(87\)90026-4](https://doi.org/10.1016/0005-1098(87)90026-4)
- Kristiansen, B., 1994. Integrated design of a fermentation plant. The production of Baker's yeast, The Chemical Engineering Journal and the Biochemical Engineering Journal. VCH Verlagsgesellschaft, Weinheim.
- Sonnleitner, B., Hahnemann, U., 1994. Dynamics of the respiratory bottleneck of *Saccharomyces cerevisiae*. J. Biotechnol. 38, 63–79. [https://doi.org/10.1016/0168-1656\(94\)90148-1](https://doi.org/10.1016/0168-1656(94)90148-1)
- Sonnleitner, B., Käppeli, O., 1986. Growth of *Saccharomyces cerevisiae* is controlled by its limited respiratory capacity: Formulation and verification of a hypothesis. Biotechnol. Bioeng. 28, 927–937. <https://doi.org/10.1002/BIT.260280620>

Appendix A: controller_function_template m-function code

```
function [ut,auxt,auxt_labels]=controller_function_template(i,t,x,y,u,ts,csi,o,aux)

% File template for creating m-functions implementing control strategies
% for the Yeastsim bioprocess simulator
%
%[ut,auxt,auxt_labels]=controller_function_template(i,t,x,y,u,ts,csi,o,aux)
%
% OUTPUT
% ut ... row vector of values of manipulated variables at time t
% aux ... row vector of auxiliary variables for optional storage of time-
% varying variables and parameters of control strategies (e.g. setpoints)
%
% ut(1) ... feed flowrate (F, ml/s)
% ut(2) ... air flowrate (Fv, ml/s)
% ut(3) ... stirrer speed (n, 1/s)
%
% aux(1) ... auxiliary variable no. 1
% aux(2) ... auxiliary variable no. 2
% aux(3) ... auxiliary variable no. 3
% aux(4) ... auxiliary variable no. 4
% aux(5) ... auxiliary variable no. 5
%
% aux_labels(1) ... text label for auxiliary variable no. 1
% aux_labels(2) ... text label for auxiliary variable no. 2
% aux_labels(3) ... text label for auxiliary variable no. 3
% aux_labels(4) ... text label for auxiliary variable no. 4
% aux_labels(5) ... text label for auxiliary variable no. 5
%
%
% INPUT
% i ... integer index value passed from the parent FOR loop
% t ... current time (s)
% x ... matrix of state variables
% y ... matrix of output (measured) variables
% u ... matrix of manipulated variables
% ts ... sample time (s)
% csi ... glucose concentration in the nutrient solution (feed) (mol/ml)
% o ... matrix of estimated variables by soft sensors
% aux ... matrix of auxiliary variables for optional storage of time-
% varying variables and parameters of control strategies (e.g. setpoints)
% Note: for the composition of the x,y,u,o,aux matrices, see help for
% yeast_simcon.m function

% Auxiliary global variable for recording the number of controller m-
% function executions
global citac_spusteni_reg

% Initialization of function output variables
ut=zeros(1,3);
auxt=zeros(1,5)*NaN;
auxt_labels={'label_auxvar1','label_auxvar2','label_auxvar3','label_auxvar4',
'label_auxvar5'};

% Calculation of feed flowrate (ml/s)
```



```

% admissible range for model laboratory bioreactor: 0-0.16 ml/s (0-10
ml/min), in addition, the total limit for the volume of the bioreactor must
be observed, see below
% ut(1)=<define calculation according to your control strategy>;

% Calculation of air flowrate (ml/s)
% admissible range for model laboratory bioreactor: 0-250 ml/s (0-15 l/min)
% typically set to a constant value of 10 l/min, i.e. 167 ml/s
% ut(2)=167;
% or
% ut(2)=<define air flowrate calculation according to your control
strategy>;

% Calculation of stirrer speed (1/s)
% admissible range for model laboratory bioreactor: 0-16 1/s (0-1000 1/min)
% typically set to a constant value of 600 1/min, i.e. 10 1/s
% ut(3)=10;
% or
% ut(3)=<define stirrer speed calculation according to your control
strategy>;

% calculation of values of manipulated variables at time t completed
citac_spusteni_reg=citac_spusteni_reg+1;

% safety check against overfilling of the bioreactor

% maximum total allowable volume of liquid in the bioreactor (ml)
vol_max=7500;

cur_vol=x(i,6);
new_vol=cur_vol+ut(1)*60;

if new_vol>vol_max
    ut(1)=(vol_max-cur_vol)/60;
end

```

Appendix B: RUN_YEAST_SIMCON.m script code

```
clear ALL;
clear global;
close all;
clc;

% SETUP SECTION - START
%
% x0 ... initial condition at time t0
% typically x0=[cx0 0 0 0.23e-6 0 v0 0.2/3600 0 0]
% where cx0 is the initial biomass concentration (e.g. 1 g/l, i.e. 4e-5
mol/ml)
% and v0 is the initial volume (typically 5 l in a lab fermenter, i.e. 5000
ml)
% note: for the meaning of other entries of x0, see help for yeast_simcon.m
function
%
x0=[4e-5 0 0 2.3e-7 0 5000 5.5556e-5 0 0];

% csi ... concentration of glucose in the nutrient solution (mol/ml) (in a
lab fermenter typically 100 g/l, i.e. 0.0033 mol/ml)
%
csi=0.0033;

% Setup of the simulation time parameters
% t0 ... start time of simulated cultivation (s)
% t1 ... end time of simulated cultivation (s)
% ts ... sampling time (s)
%
t0=0;
t1=36000;
ts=60;

% Control strategy selection
% reg=@controller_function where controller_function is the name of the m-
function implementing the selected control strategy, in the form
[ut,auxt,auxt_labels]=controller_function(i,t,x,y,u,ts,csi,o,aux), see the
regs folder for examples
%
reg=@regulator_feedforward_xest;
% reg=@regulator_DOCONT_v1;
% reg=@regulator_DOCONT_v2;
% reg=@regulator_DOCONT_v3;
% reg=@regulator_DOCONT_v4;

% auxiliary global variable for control strategy
global citac_spusteni_reg
citac_spusteni_reg=0;

% SETUP SECTION - END

% SIMULATION SECTION - START

[yc,xc,uc,oc,auxc,auxc_labels]=yeast_simcon(t0,t1,ts,x0,csi,reg);
```

```

% SIMULATION SECTION - END

% RESULTS AND VISUALIZATION SECTION - START

tim=[t0:ts:t1]';
tim=tim./60;

% Calculation of process quality indicators
process_indicators=yield_prod_calc(tim,xc,yc,uc,csi);

% biomass yield on substrate (gX/gS)
disp('Yield (gX/gS):')
disp(process_indicators(1));

% bioprocess productivity (gX/l/h)
disp('Productivity (gX/l/h):')
disp(process_indicators(2));

% Plots of state variables
figure(1);plot(tim,xc(:,1));title('Biomass concentration
(mol/ml)');xlabel('Cultivation time (min)');
figure(2);plot(tim,xc(:,2));title('Glucose concentration
(mol/ml)');xlabel('Cultivation time (min)');
figure(3);plot(tim,xc(:,3));title('Ethanol concentration
(mol/ml)');xlabel('Cultivation time (min)');
figure(4);plot(tim,xc(:,4));title('O2 concentration
(mol/ml)');xlabel('Cultivation time (min)');
figure(5);plot(tim,xc(:,5));title('CO2 concentration
(mol/ml)');xlabel('Cultivation time (min)');
figure(6);plot(tim,xc(:,6));title('Fermentation broth volume
(ml)');xlabel('Cultivation time (min)');

% Plots of output (measured) variables
figure(7);plot(tim,yc(:,1));title('Ethanol mol fraction (%)
mol.));xlabel('Cultivation time (min)');
figure(8);plot(tim,yc(:,2));title('O2 mol fraction - off-gas (%)
mol.));xlabel('Cultivation time (min)');
figure(9);plot(tim,yc(:,3));title('CO2 mol fraction - off-gas (%)
mol.));xlabel('Cultivation time (min)');
figure(10);plot(tim,yc(:,4));title('Dissolved O2 concentration (%)
sat.));xlabel('Cultivation time (min)');

% Plots of manipulated variables
figure(11);plot(tim,uc(:,1));title('Feed flowrate
(ml/s)');xlabel('Cultivation time (min)');
figure(12);plot(tim,uc(:,2));title('Air flowrate
(ml/s)');xlabel('Cultivation time (min)');
figure(13);plot(tim,uc(:,3));title('Stirrer speed
(1/s)');xlabel('Cultivation time (min)');

% Plots of estimated variables
figure(14);plot(tim,xc(:,1),'g',tim,oc(:,1),'r',tim,oc(:,2),'b');title('Bio
mass concentration and its estimates by soft sensors
(mol/ml)');xlabel('Cultivation time (min)');

% RESULTS AND VISUALIZATION SECTION - END

```

Appendix C: Structure of the exported YeastSim data structure

YeastSim struct with fields:

```
        xc: [601×9 double]
        yc: [601×4 double]
        uc: [601×3 double]
        oc: [601×2 double]
        aux: [601×5 double]
aux_labels: {1×5 cell}
        x0: [4.0000e-05 0 0 2.3000e-07 0 5000 5.5556e-05 0 0]
        t: []
        labels: [1×1 struct]
        t0: 0
        t1: 36000
        ts: 60
        csi: 0.0033
        reg: @regulator_feedforward_xest
reg_name: 'regulator_feedforward_xest'
        yield: 0.4000
productivity: 1.1303
```