

Article

A Software Emulator for the Modelling and Control of an Activated Sludge Process in a Wastewater Treatment Plant

Dan Selișteanu ^{1,*} , Ion-Marian Popescu ¹, Monica Roman ¹, Constantin Șulea-Iorgulescu ¹ 
and Sorin Mehedințeanu ²

¹ Department of Automatic Control and Electronics, University of Craiova, A.I. Cuza 13, 200585 Craiova, Romania; marian.popescu@edu.ucv.ro (I.-M.P.); monica.roman@edu.ucv.ro (M.R.); constantin.sulea@edu.ucv.ro (C.Ș.-I.)

² Wastewater Treatment Plant Făcăi, Water Company Oltenia, 200177 Craiova, Romania; sorinmehedinteanu@yahoo.com

* Correspondence: dan.selisteanu@edu.ucv.ro

Abstract: The design and implementation of a simulator, as a real-time application, for a complex process from the biological treatment stage of a wastewater treatment plant (WWTP), is addressed. More precisely, this emulator was achieved as a software tool that can be later integrated into a more complex SCADA (supervisory control and data acquisition) system of the WWTP Făcăi, Romania. The basic idea is to implement and validate a reduced-order model of the activated sludge process (ASP), initially simulated in the Matlab/Simulink environment (The MathWorks, Inc., Natick, MA, USA). Moreover, an advanced multivariable adaptive control scheme of the ASP is addressed. This software tool can be made to work in parallel with the evolution of the process and can have as input signals measured directly at the process level, possibly following parametric or model adaptations. The software emulator is developed in the LabWindows/CVI programming environment (National Instruments), which offers low-level access to hardware or software systems that have minimal open-architecture facilities. This environment provides versatile drivers and software packages that can facilitate the interaction with software tools developed within some earlier SCADA systems. The structure and the graphical interface of the emulator, some functionalities, experiments, and evolution of main variables are presented.

Keywords: software simulators; wastewater treatment plants; activated sludge process; bioprocess control; adaptive control; supervisory control and data acquisition



Citation: Selișteanu, D.; Popescu, I.-M.; Roman, M.; Șulea-Iorgulescu, C.; Mehedințeanu, S. A Software Emulator for the Modelling and Control of an Activated Sludge Process in a Wastewater Treatment Plant. *Processes* **2021**, *9*, 2054. <https://doi.org/10.3390/pr9112054>

Academic Editor: Mohd Azlan Hussain

Received: 18 October 2021

Accepted: 15 November 2021

Published: 16 November 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

A WWTP is a complex plant typically used for the removal of organic matter, nitrogen, and phosphorus from the polluted water cleared from domestic, industrial, and agricultural activities. The treatment processes in a WWTP are commonly organized into three stages: primary (or mechanical) treatment, secondary (or biological) treatment, and tertiary (chemical) treatment. This work will address issues related to the modelling and control of biotechnological processes (bioprocesses) within WWTPs, i.e., from the secondary treatment stage. These topics have been important research subjects in the last decades [1,2].

The control structure of a WWTP is usually organized as a distributed control system/supervisory control and data acquisition (DCS/SCADA). This architecture is a hierarchical and distributed control system, with the differences between several implementations residing in the number of hierarchical levels, the software and hardware equipment, and the control algorithms. However, a typical DCS has four to five hierarchical levels [3,4]. For example, the DCS–SCADA system from WWTP Făcăi, Romania (which is a big station of the Water Company Oltenia WCO, designed for the Craiova city, 300,000 inhabitants) has four levels—Figure 1 [5].

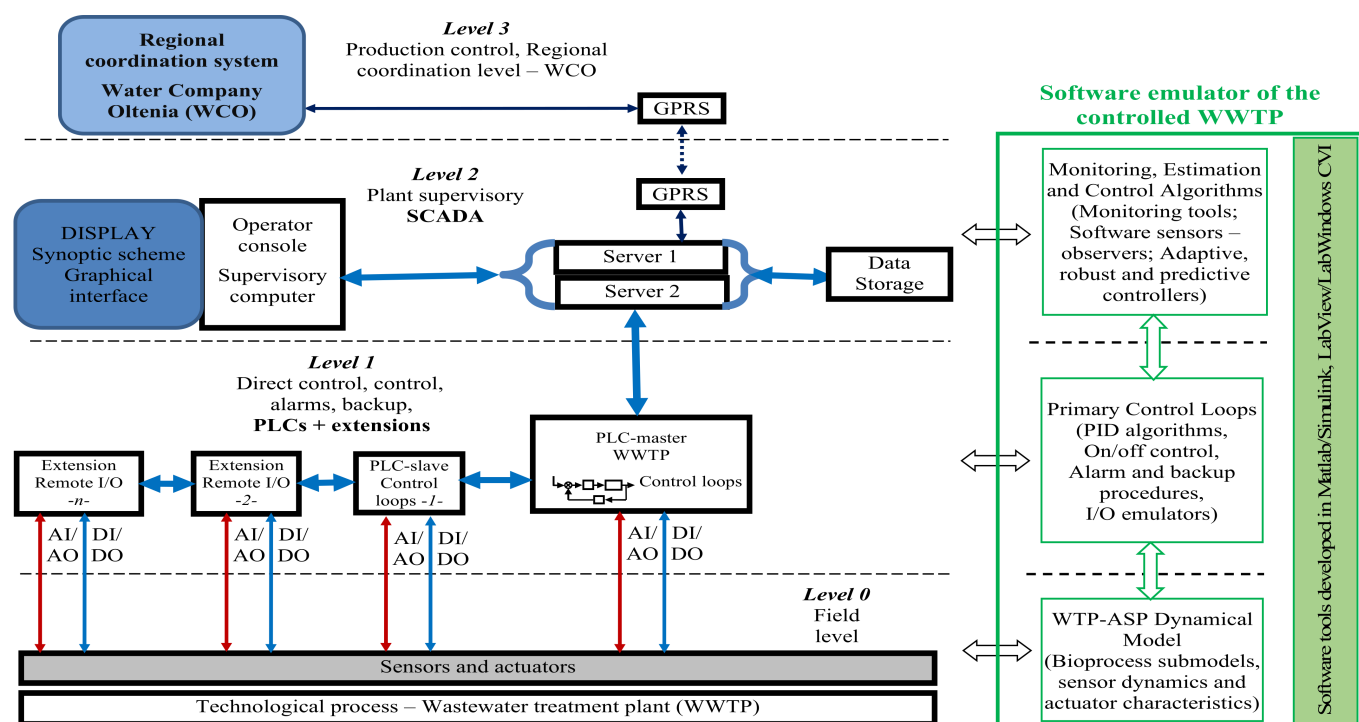


Figure 1. The distributed control system/SCADA architecture of a WWTP and the simulator of the controlled plant (WWTP Făcăi, Romania) [5].

This work is focused on the development of an emulator dedicated to a specific process within the biological treatment stage of a WWTP—the activated sludge process. The bioprocess emulators are useful to understand the operation of real plants [6,7]. In bioengineering, specifically in WWTPs, further reasons for the use of simulators are the large data sets and the biological experimental risks. Among the first software tools developed for bioprocess modelling, we can mention CAMBIO [7,8] and DESASS [9], designed for WWTPs (more specifically, for the Biological nutrient removal model no. 1). Another known simulator is SIMBA (ifak e.V., Magdeburg, Germany). The activated sludge process description in SIMBA is based on the so-called ASMs (activated sludge models), with software tools that use the models ASM1, ASM2d [10], and ASM3. As Matlab/Simulink is a software environment very useful for modelling, but also for control design, there are several bioprocess simulators developed in this environment [11,12]. However, as the efficient operation of a WWTP needs useful online information from the processes and, at the same time, useful real-time control decisions, it is quite important to design and to implement a simulator able to deal with these specificities, allowing a good connectivity with the process and with the SCADA software and devices. Thus, the LabWindows/CVI environment from National Instruments (Austin, TX, USA) [13] is an appropriate choice.

The mathematical models of bioprocesses have a certain degree of generalization, especially those models derived from the mass balance equations of the biochemical reactions. Thus, these models can describe a wide range of biotechnological processes [14], such as fermentation process in the food industry (wine, beer, baker's yeast, and so on), biological processes in the pharmaceutical industry, bioprocesses in wastewater treatment plants, and so on. Moreover, even the control techniques and strategies may differ for various bioprocesses, and there is a common framework specific for control solutions [14]. Regarding the design and implementation of simulators for food bioprocesses, one of the first was CAMBIO, a software package devoted to the process of kinetic modelling [8], with application to yeast fermentation. Further approaches used specific software environments (especially Matlab/Simulink, The MathWorks, Inc., Natick, MA, USA) in order to design bioprocess simulators. For example, in [7], an interactive user-friendly software system

for the identification of bioprocesses models is proposed. It is based on optimization programmes built in the graphical user interface development environment (GUIDE) of Matlab, and the kinetics identification of an ethanol production process is addressed. In [15], the design of integrated biopharmaceutical processes using process simulators, discrete event simulators, or finite capacity scheduling tools is approached. More precisely, the modelling of a single batch bioprocess for the production of therapeutic monoclonal antibody (MAb) is addressed. Another simulator dedicated to the parameter estimation of MAbs, based on the Matlab environment, is presented in [16].

As in the case of WWTP simulators, most of these simulators dedicated to several bioprocesses from food and pharmaceutical industry lack connectivity with the real processes, with respect to data acquisition and control implementation.

In most WWTPs, the activated sludge process is a key component of the biological treatment stage. Several works indicated that there are numerous hurdles in the ASP control design and implementation, such as kinetics nonlinearity and uncertainty, unavailability of online data, and noisy measurements [17–20]. To handle these problems, various control techniques were designed: adaptive and robust-adaptive approaches [19,21], sliding mode and neural networks-based control [22], event-based control [23], optimal and predictive control [24], and so on. The present work will use a novel control strategy for the ASP, proposed in [25]. More precisely, an adaptive multivariable decoupled control scheme tailored for the ASP from the WWTP Făcăi is considered. The design of this control strategy was based on several practical and realistic considerations, such as the unavailability of state variables, the kinetics uncertainty, and the lack of knowledge regarding the influent concentration.

To design control algorithms in WWTPs, it is necessary to obtain useful information about the processes, using the available sensors as well as software tools. These tools are related to several levels of the DCS, involving data acquisition solutions (level 0), primary control loops (level 1), as well as estimation (software sensors) and advanced control algorithms (level 2)—Figure 1. The need for such software tools and emulators is obvious as the implementation of advanced control techniques in WWTPs is quite poor [26,27]. In many cases, only empirical rules and primary controllers (PID, on/off) are used. The use of advanced control would lead to an improvement in efficiency and performance.

In this paper, which is an extended work based on [5,25], the design and implementation of a simulator, as a real-time application, for the ASP residing in the WWTP Făcăi, Romania is approached. The proposed emulator is a software tool that can be later integrated into a more complex SCADA. Firstly, a reduced-order model of the ASP (initially simulated in the Matlab/Simulink environment) is implemented. This is related to level 0 of the DCS and contains the nonlinear dynamical model of the ASP adapted to the WWTP Făcăi. Secondly, the adaptive multivariable control algorithms developed in [25] are integrated into the simulator. These are related especially to level 2 of the DCS. The emulator is developed in the LabWindows/CVI programming environment (National Instruments) and can work in parallel with the evolution of the process, having as inputs the signals measured directly at the process level. This environment provides versatile drivers and software packages that can facilitate the interaction with software tools developed within some earlier SCADA systems. The structure and the graphical interface of the emulator, some functionalities, experiments, and evolution of main variables are presented. The main advantages of the proposed emulator are as follows:

- It is a real-time software application, as the processing is implemented using the interrupts generated by a timer, under a strict time interval—the sampling period.
- It can be integrated into existing SCADA systems and can ensure the operation in parallel with the process, having the same input signals. This allows the calibration of the parameters of the mathematical model and the validation of the observers.
- Useful comparisons of the advanced control structure performance can be achieved, in relation to the existing classic control system at a WWTP.

The structure of the paper is as follows. Section 2 presents the typical organization of a WWTP and the description of the WWTP Făcăi. Moreover, the activated sludge process and its dynamical model are described. Section 3 approaches the adaptive multivariable decoupled control of the ASP. In Section 4, the graphical interface of the emulator, its functionalities, and some experiments are widely presented. Finally, Section 5 concludes the paper.

2. Structure of the WWTP Făcăi and Activated Sludge Process Model

The treatment plant Făcăi (WCO) was designed for the treatment of polluted water coming from the city of Craiova (about 300,000 inhabitants), Romania. An aerial view is presented in Figure 2. This station consists of three stages: the primary or mechanical stage, secondary/biological stage, and tertiary/chemical stage. The WWTP has four independent operating lines for the primary and secondary stages. The total treatment capacity is $5 \text{ m}^3/\text{s}$ in the primary stage, and $2.5 \text{ m}^3/\text{s}$ can be processed in the secondary stage. The general block scheme of the WWTP Făcăi is provided in [5]. The DCS/SCADA structure of the WWTP is presented in Figure 1. The plant is equipped with various data acquisition and control devices specific for WWTPs, produced by Siemens, National Instruments, Rockwell, Schneider, and Endress + Hauser, among others. The main process from the biological stage is the ASP, an aerobic process that typically is operated in two sequenced units: a bioreactor (aeration tank) and a sedimentation tank (settler). The biological degradation of the pollutants is achieved in the aeration tank, while in the settler, the liquid is clarified, and the sludge is separated. A part of the settled biomass is recycled into the reactor, while the excess is removed [19]. In industrial practice, the ASP uses aeration reactors with anoxic and aerobic tanks. The number of tanks is determined by the plant size.

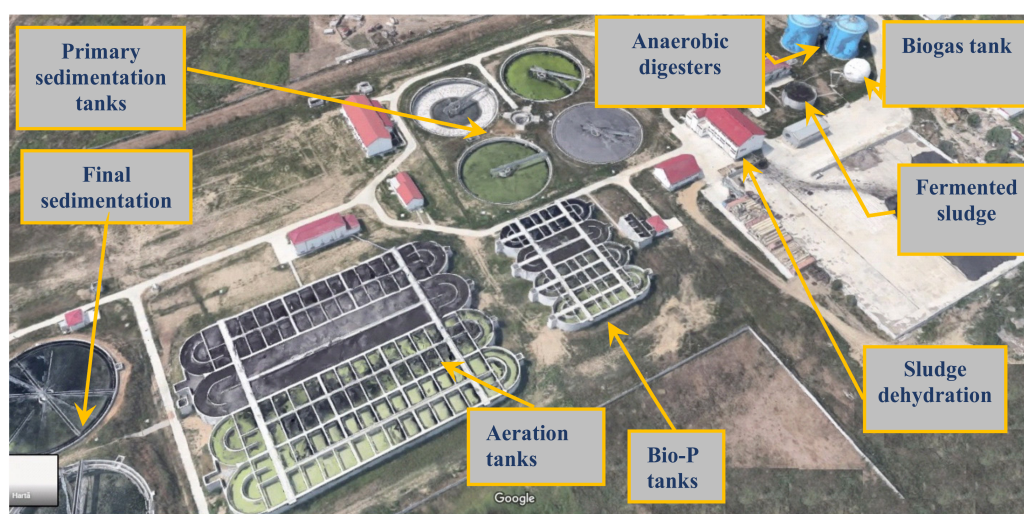


Figure 2. An aerial view of the WWTP Făcăi, Romania [5].

In this work, the ASP model and parameters correspond to WWTP Făcăi, and the ASP block scheme is presented in Figure 3 [25]. The volume of every anaerobic/anoxic tank (four tanks for four operating lines) is 3375 m^3 . These tanks are connected with aerobic tanks with a volume of $15,000 \text{ m}^3$ each.

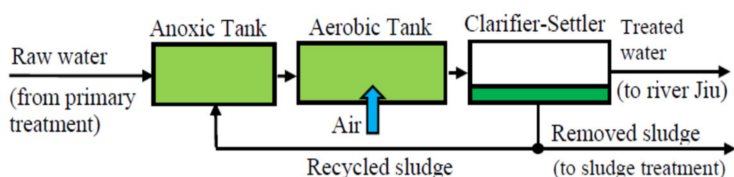


Figure 3. The ASP structure—WWTP Făcăi [25].

The dissolved oxygen (DO) concentration in the aeration tanks, which is a key parameter, should be over 1.5 mg/L; if not, the bacteria growth decreases or, even worse, is stopped. When the polluted water is characterized by a less organic load, the DO concentration may increase to 6–8 mg/L. The flow of recycled sludge from the clarifier-settler is the major parameter that allows the adjustment of the activated sludge concentration. This succinct description is quite generic; in fact, the ASP is a complex, nonlinear, and uncertain process [20,23].

To be able to design and implement the control algorithm, it is necessary to obtain useful ASP mathematical models [17,28,29]. The most known ASP models are the ASMs (activated sludge models): ASM1, ASM2, ASM2d, and ASM3 [10]. Unfortunately, these are complicated, being unusable for control purposes. Therefore, reduced-order models were developed and used for control design. In this work, we will use a simplified model, as in [25], initially proposed by Nejari et al. in [19], but adapted for the ASP from the WWTP Făcăi:

$$\begin{aligned}\dot{C}_X(t) &= \mu C_X - dC_X - r dC_X + r dC_{Xr}, \\ \dot{C}_S(t) &= -\frac{1}{Y} \mu C_X - (1+r)dC_S + dC_{Sin}, \\ \dot{C}_{DO}(t) &= -\frac{K_O}{Y} \mu C_X - (1+r)dC_{DO} + \alpha F_O(C_{DO}^{\max} - C_{DO}) + dC_{Oin}, \\ \dot{C}_{Xr}(t) &= (1+r)dC_X - r dC_{Xr} - \beta dC_{Xr},\end{aligned}\quad (1)$$

where C_X , C_S , C_{DO} , and C_{Xr} are the concentrations of the activated sludge (the biomass), the substrate (the pollutant), DO, and the recycled biomass, respectively.

In Equation (1) d is the dilution rate ($d = F_{in}/V$, with F_{in} being the input pollutant flow rate and V the bioreactor volume), K_O is a constant, Y is a yield factor, α is the oxygen transfer rate, F_O is the aeration rate, C_{DO}^{\max} is the saturation DO concentration, and β is the ratio of removed sludge. C_{Oin} and C_{Sin} are the influent oxygen and substrate concentrations, respectively. The kinetics is expressed by μ , the specific growth rate, and r is the rate of recycled sludge.

One of the most important kinetic variables is the growth rate μ , expressed in the case of our particular ASP by a dual Monod model, as follows:

$$\mu(t) = \mu_{\max} \frac{C_S(t)}{K_S + C_S(t)} \cdot \frac{C_{DO}(t)}{K_{DO} + C_{DO}(t)} \quad (2)$$

where μ_{\max} is the maximum specific growth rate, while K_S and K_{DO} are constants that express the saturation phenomenon.

The reduced-order dynamical model, Equations (1) and (2), was simulated in the Matlab environment in some previous works [5,25]. In the present paper, the dynamical model is integrated inside a simulator developed in LabWindows/CVI, whose structure will be presented in Section 4.

Owing to the complexity of the WWTP, the proposed simulator addresses only a part of the process, more precisely the ASP structure presented in Figure 3. It should be mentioned that the processes that take place in the WWTP contain several other technological components, in all the stages of the station: mechanical, biological, and chemical, as can be seen, for example, in Figure 2. Moreover, the existing SCADA of the plant, succinctly described in Figure 1, covers in the present moment only some of these processes.

3. An Adaptive Multivariable Decoupled Control Strategy for the ASP

3.1. Control Objective

The control goal for the ASP within the WWTP is to obtain an imposed small level for pollution, with a proper aeration, in severe conditions owing to the unavailability and uncertainty of several key variables and parameters [19–21]. According to the particularities of the WWTP Făcăi, the controlled variables are C_S and C_{DO} , while the control inputs are the recycled sludge rate r and the aeration rate F_O [25]. This results a multivariable control

scheme, which should work such that the controlled outputs to asymptotically track the imposed references.

In [25], an advanced decoupled multivariable adaptive control strategy was developed for the ASP from the WWTP Făcăi, as shown in Figure 4, where C_{DO}^* and C_S^* are desired piece-wise constant outputs (setpoints). This control was designed under several realistic conditions: the inflow rate F_{in} varies in wide limits (but is measurable), C_{Sin} is time varying and unknown, the kinetics μ is unknown, and the concentrations C_X and C_{Xr} are not available. The existing online measurements are C_S , C_{DO} , and C_{Oin} , and all the other parameters are known. The removal of interactions between inputs and outputs was achieved using feedback actions supplied by a properly designed multivariable controller.

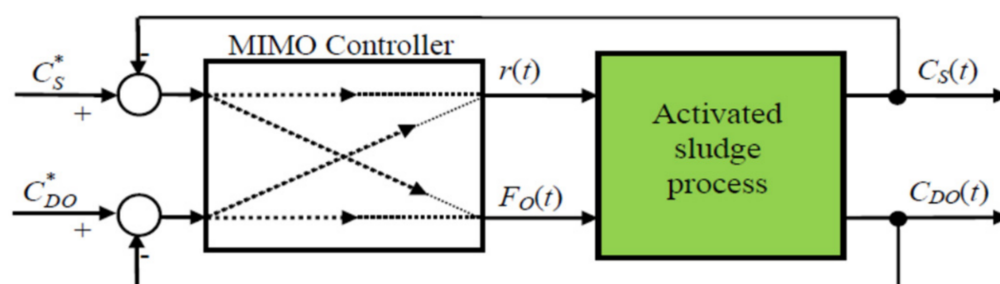


Figure 4. The decoupling multivariable adaptive control structure of the ASP [25].

3.2. Multivariable Adaptive Control Structure

The full design of the control strategy of the ASP is provided in [25]. The adaptive control strategy is developed as a linearizing controller coupled with a sliding mode observer (SMO) used to obtain estimates of the unknown influent pollutant concentration, with an asymptotic state observer, and with a suitable parameter estimator dedicated to kinetics estimation.

More exactly, the adaptive control structure is achieved using two linearizing control laws. As these control laws contain the states C_X , C_S , and C_{DO} (of which only the last two are measurable), as well as the unknown growth rate, these exact controllers become two adaptive control laws if μ and C_X are replaced by their online estimates provided by a parameter estimator (an observer-based estimator OBE), and respectively by an asymptotic observer. Furthermore, as C_{Sin} is unmeasurable, it is estimated using a generalized super-twisting SMO [25].

Thus, the proposed adaptive control layout is made up by combining the sliding mode observer, the asymptotic observer, and the OBE with the linearizing controllers. The structure and the information flow inside the controlled process can be seen in the overall adaptive control scheme presented in the main graphical interface of the proposed emulator. The control algorithms were integrated within the emulator developed in LabWindows/CVI, whose structure will be presented in the following section.

4. Emulator Design and Functionalities

4.1. Main Graphical Interface of the Emulator

The proposed simulator will be useful as a software tool that can be later integrated into a more complex SCADA system, and thus will allow the implementation of advanced control algorithms and the improvement in the WWTP operation. Moreover, the graphical interfaces of the simulator are useful as virtual instrumentation and synoptic schemes of the process. In the first stage, the simulator can generate control actions using the real input data from the process, the plant model, and the proposed advanced control scheme. These commands are not applied to the WWTP, but are used in several comparisons with the actual classical control solution. Further research will be focused on the hard real-time implementation of the advanced control solutions.

The software emulator is developed in the LabWindows/CVI programming environment (National Instruments, Austin, TX, USA), as this environment offers low-level access to hardware or software systems that have minimal open-architecture facilities. Moreover, it provides versatile drivers and software libraries that can facilitate the interaction with software tools developed within the existing SCADA from the WWTP Făcăi. This is a development environment for real-time applications based on the ANSI C standard, providing complex libraries for control applications, and a Run-Time Engine software module with interrupt handler facilities. Moreover, through the “Open architecture” feature, it allows the integration in the same project of various software modules, achieved in other development environments, which will be link-edited together with the LabWindows/CVI source code files. Thus, the pre-compiled object code files can be included in the project (these have been developed in the assembly language). Moreover, various *.lib or *.dll library files can be added to the LabWindows project. Further, new *.lib and *.dll files that contain various software functions can be developed and can be used in other development environments.

It should be mentioned that the LabWindows/CVI—NI programming environment is based on the ANSI C standard but contains libraries of functions oriented on complex processing as well as real time through the Run-Time Engine (RTE) module that interfaces the application developed with the Windows operating system, more precisely with the kernel of the operating system. At the same time, because a real-time system is a dedicated hardware-software unit, the NI hardware systems can work using, in addition to RTE, a dedicated operating system kernel that uses real-time principles.

The LabWindows/CVI development environment can be considered of the open architecture type, in the sense that the developed application can integrate modules written in other languages (including .obj modules, written in assembly language) or can be integrated into other development environments.

Figure 5 shows the main graphical interface of the real-time simulator, which has two instances related to the control system proposed for the biological stage: open-loop and closed-loop. As can be seen, this emulator is based on the multivariable adaptive control system succinctly described in the previous section. The mathematical model used for the ASP is the reduced-order one, presented in Section 2.

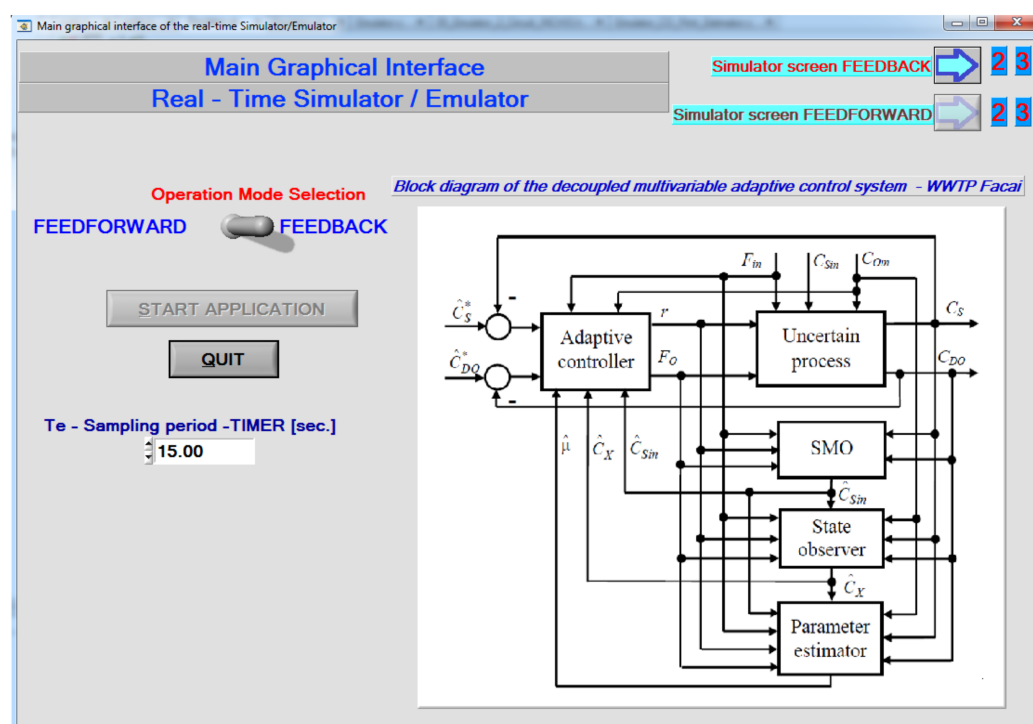


Figure 5. Main graphical interface of the software emulator of the controlled ASP.

The main operating modes of the emulator are as follows:

1. Simulator of the open-loop system, which includes the model of the ASP and the estimator of the influent substrate concentration. This is useful in the analysis, testing, and validation of the mathematical model of the process and of the estimator.
2. Real-time emulator of the closed-loop system, based on the proposed multivariable adaptive control structure of the ASP.

Thus, from the main graphical interface, two options can be selected: open loop or closed loop, by means of a selection switch. Switching between the two operating modes can be done interactively during operation, and thus can handle the possible shocks from the evolution of the process, as they are equivalent to the known problems of automatic/manual switching.

By pressing the *Start Application* button, the timer of the real-time software tool is actually started, which will generate the interrupts related to the sampling period used in the processing of the mathematical model. Through this mechanism, the simulator is implemented in real time, because it can have a “continuous” running in time, with the very strict implementation of the sampling period. This running eliminates the problems of standard simulations over a finite period of time, where the problem of initial conditions is very important. Thus, after starting the application, it is expected to reach a stationary operating mode, after which various input signals (references, disturbances) can be modified online to view their effects.

The software function (Program Block), which is a C function, is attached to the timer interrupt and encapsulates all the tasks necessary to be performed with a certain sampling period:

- Tasks dedicated to the processing of the mathematical process model;
- Tasks dedicated to the implementation of the estimators (three estimators 1, 2, and 3);
- Tasks for the implementation of open-loop/closed-loop control architectures;
- Tasks for displaying the evolution of various signals at the level of the approached architecture.

The proposed simulator acts as a soft real-time system, and in the next stage, it will be implemented on an NI hardware system in order to be a hard real-time system [30,31]. The bioprocess taken into consideration is a slow process; therefore, the real-time constraints should not be critical; however, slow processes are often difficult to handle from the control engineering point of view. In this context, the time instants in which a modification of a certain stationary regime or the appearance of a disturbance are detected are very important. The process response will be influenced by such disturbances. Thus, it is important to detect in real time the evolution of causes and effects.

4.2. Simulator of the Open-Loop System

After starting the real time application and selecting the open-loop operating mode, it is possible to navigate in the next two related screens—Figures 6 and 7, where real-time signals are presented as graphs resulting from the simulator. The selection switch Simulation/Measured can be used to select the simulation mode or the mode with real input signals, measured from the process:

- “Simulation” is a simulated working mode, where all the necessary parameters are taken from the virtual knobs of the graphical interface, placed at the top and side of the screen.
- “Measured” is a working regime with real (measured) data/parameters taken from the process, as far as this is technically possible, or there are sensors available for measurements.

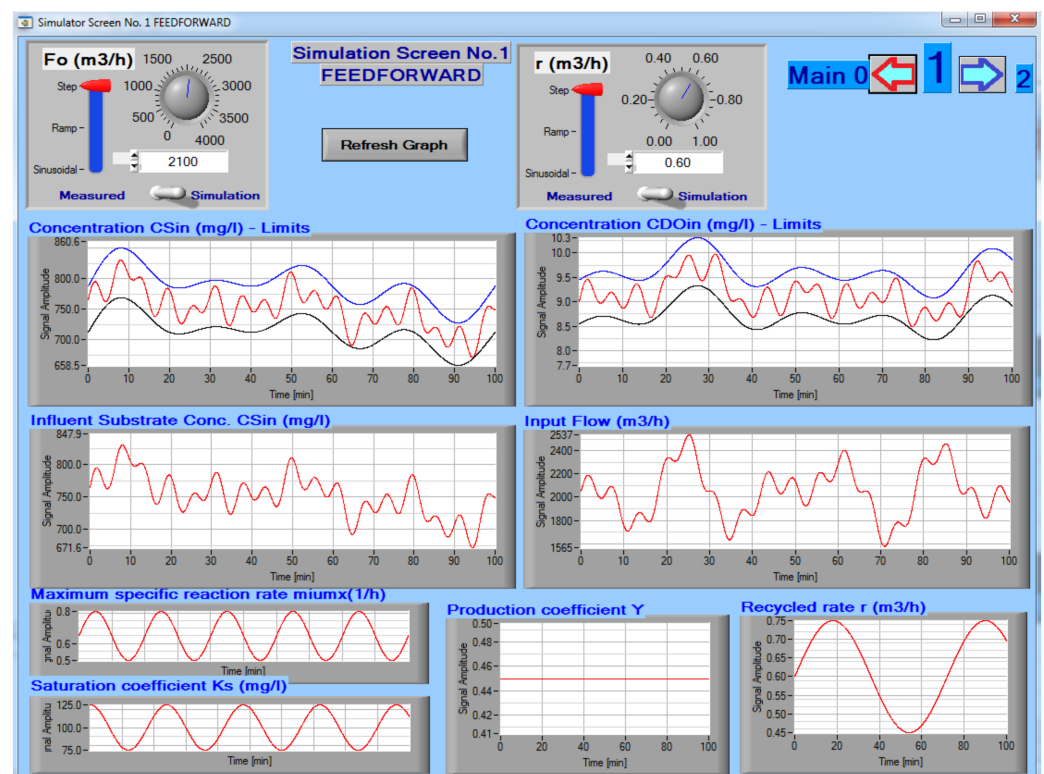


Figure 6. Open-loop simulation of the ASP, the biological treatment stage, with the visualization of the evolution of the main biological variables (the first screen).

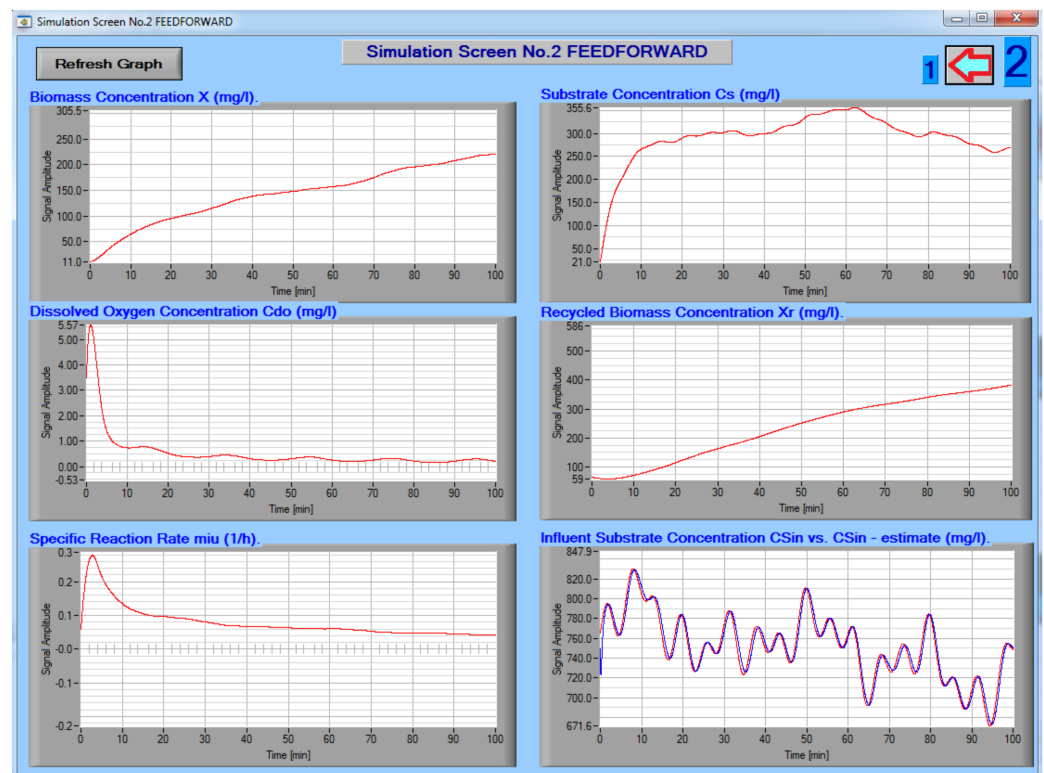


Figure 7. Open-loop simulation of the ASP, the biological treatment stage, with the visualization of the evolution of the main biological variables (the second screen).

Figure 6 presents the first screen with the graphical interface for the implementation of the software tool for the real-time simulation of the evolution of the ASP mathematical model. The architecture is of the feedforward type, and estimates some kinetic parameters of the unknown process. More precisely, the concentration of the influent substrate is estimated using a sliding mode observer, because of the fact that this input variable can be considered as a part of the ASP model.

The simulator provides, via the specific graphical window, the evolution of several biological variables of the activated sludge process, such as the concentrations of influent DO and of the influent substrate, as well as the maximum specific growth rate, the input flow of polluted water, the recycled flow, and other kinetic parameters of the process (Figure 6). It should be mentioned that the influent DO and substrate concentrations evolve into some pre-specified bounds. Several aspects about the time evolution and the profiles of biological variables are widely discussed in [25]. Figure 7 depicts the second screen of the graphical interface corresponding to the open-loop operating mode of the simulator.

Here, the evolution of other key variables of the ASP can be plotted, such as DO, substrate, biomass and recycled biomass concentrations, specific growth rate, and the concentration of the influent substrate versus its estimation.

The developed application allows the calibration/testing/validation of the mathematical model with the real process parameters and, at the same time, the operation in parallel with the process, making it possible to assign the same input data for the mathematical model as for the real process. Therefore, the graphical interface related to the open-loop option allows the selection of the simulator working mode in SIMULATED or REAL mode, by generating input signals/step, ramp, sinusoidal reference from the graphical interface, and input signals obtained by measuring real signals at the WWTP, respectively. The graph objects in these two screens are populated with the signal samples in parallel and simultaneously in both related windows, even if the window in which it is located is not visible at a given time.

As a general rule of implementing a real-time application, this should work by using interrupts, so that the event is treated when it occurs and not later. For example, for the timer object, an important field is available in the corresponding properties box. In this field, the “callback” software function is attached. Here will be inserted the name of the developed function “Sampling_Action ()”, which will contain the software block that will be called at each sampling period, when the interrupt from the timer appears. The CVICALLBACK functions, which deal with interrupts, can be attached to any graphic object, as well as to other tasks, even virtual ones, such as communication on interrupts, alarms, priorities, and so on. Three CVICALLBACK type functions constitute the core of the developed real-time application. These three functions are as follows:

- *Sampling_Action ()*, attached to the timer object, which implements the sampling period and will contain all the processing performed by our application. [Timer] # [EVENT_TIMER_TICK—event];
- *Start_Application ()*, attached to a button type graphic object, more precisely to the button labelled “Start Application” in the graphical interface. After the run of the software application, the first window will appear on the screen. This will wait for the arrival of events of all types on the screen, by continuously running the Run-Time Engine module. The presented function detects the starting event of the application and will make the necessary settings for the operation of the program, after which the application timer will actually start. [APPLICATION START button] # [EVENT_COMMIT-event];
- *QuitCallback ()*, attached to a button graphic object, i.e., the QUIT button. This will stop the application, free up memory, and give control of the operating system. [Quit button] # [EVENT_COMMIT-event].

The *WinMain ()* function includes the five instructions as a kind of standard, because any LabWindows/CVI application must minimally load a window into memory, display it, launch Run-Time Engine, and then will wait for events in an endless mode. Other

code modules can be added to this function (in C Language), such as installing multiple windows, installing priority for interrupts, and other various tasks.

To illustrate how the various estimation and control algorithms are implemented, Figure 8 presents the source code related to the implementation of the SMO used for the estimation of influent substrate concentrations. The software tool based on LabWindows/CVI implements the sampling period by means of an interrupt generated by the timer available at the level of the computer system, which gives it the real-time feature and not just a simulation over a predefined time interval.

```

//##### Iterative relationships##### Implementing Estimator for Sin #####
Dk=Fink/V; Dsk=Fink/Vs;
miuk=mmask*sk/(Ksk+sk)*dok/(Kdo+dok);
xkv1=xk+T*((miuk-miuS0)*xk-Dk*(1+rk)*xk+rk*Dk*xrk);
skv1=sk+T*(-1/Yk*(miuk-miuS0)*xk-Dk*(1+rk)*sk+Dk*Sink);
dokv1=dok+T*(-Ko/Yk*(miuk-miuS0)*xk-Dk*(1+rk)*dok+alpha*Wdo0*(D0max-dok)+Dk*D0ink);
xrkv1=xrk+T*(Dk*(1+rk)*xk-(beta+rk)*Dk*xrk);
//estimator equations
zSink=-Ko*sk+dok;
eSink=zSink-zSink;
Fi1Sink=gama1Sin*(sqrt(abs(eSink)))*semm(eSink)+gama2Sin*eSink;
Fi2Sink=gama1Sin*gama1Sin/2*semm(eSink)+3/2*gama1Sin*gama2Sin*(sqrt(abs(eSink)))*semm(eSink)+gama2Sin*eSink;
zSinkv1=zSink+T*(-Dk*(1+rk)*zSink-Ko*Dk*Sink+alpha*Wdo0*(D0max-dok)+Dk*D0ink-LSin*alfa1Sin*Fi1Sink);
Sinkv1=Sink+T*LSin*LSin*alfa2Sin*Ko*Dk*Fi2Sink;
ak=zSinkv1;
if (Sinkv1<0) Sinkv1=0;
if (cnt_autozoom) { xkv1=xk; skv1=sk; dokv1=dok; xrkv1=xrk; miuk_1=miuk; Sink_1=Sink; Sinkv1=Sink; }
cnt_autozoom=0;
//Graphic display of ESTIMATOR signals >>>FEEDFORWARD
PlotLine (CDpanelHandle, PANEL_CD_1_GX, tk_1, xk, tk, xkv1, VAL_RED);
PlotLine (CDpanelHandle, PANEL_CD_1_GS, tk_1, sk, tk, skv1, VAL_RED);
PlotLine (CDpanelHandle, PANEL_CD_1_GDO, tk_1, dok, tk, dokv1, VAL_RED);
PlotLine (CDpanelHandle, PANEL_CD_1_GXR, tk_1, xrk, tk, xrkv1, VAL_RED);
PlotLine (CDpanelHandle, PANEL_CD_1_GMIU, tk_1, miuk_1, tk, miuk, VAL_RED);
PlotLine (CDpanelHandle, PANEL_CD_1_GSINSINE, tk_1, Sink_1, tk, Sink, VAL_RED);
PlotLine (CDpanelHandle, PANEL_CD_1_GSINSINE, tk_1, Sink, tk, Sinkv1, VAL_BLUE);
//Update a new software cycle, for the next interruption from Timer
tk_1=tk; tk=tk+Te; Refk_2=Refk_1; Refk_1=Refk;
mmask_1=mmask; Ksk_1=Ksk; Yk_1=Yk; rk_1=rk; Sink_1=Sink; Fink_1=Fink; Sinmk_1=Sinmk;
SinMk_1=SinMk; D0inmk_1=D0inmk; D0inMk_1=D0inMk; D0ink_1=D0ink;
xk=xkv1; sk=skv1; dok=dokv1; xrk=xrkv1; miuk_1=miuk; zSink=zSinkv1; Sink=Sinkv1;

```

Figure 8. The source code for the implementation of the SMO super-twisting algorithm used for the estimation of influent substrate concentration.

4.3. Closed-Loop System Emulator

The implementation of the closed-loop simulator application is based on the same environment and ideas. Here, the multivariable adaptive control structure described in Section 3 is implemented. Some of the main components are as follows:

- The first screen of the graphical interface for the closed-loop operating mode is depicted in Figure 9;
- The second screen of the graphical interface is given in Figure 10;
- Figure 11 presents a sample of the source code (LabWindows/CVI) in which the closed-loop nonlinear adaptive control inputs are implemented.

The graphical interface from Figure 9 presents at the top virtual potentiometers through which various types of reference (setpoint) signals will be generated (C_S —substrate concentration, C_{DO} —dissolved oxygen concentration) for the implemented control structure. The potentiometers on the left side of the graphical interface generate various signal variations for the process parameters, when working in simulation mode, even if the simulator is running in real time. In this way, it is possible to test various scenarios for the validation of the proposed architecture. To the extent that signals measured directly from the process are available, these signals will be replaced with actual samples from the process, leading to parallel operation of the emulator with the actual process. The selection between the two working modes is made by the switch Simulation/Measured. The whole operation of mode selection, modification of simulated or measured parameters, and adjustment of references is performed on interrupts attached to the corresponding graphic objects, so that the processing is minimally influenced by additional processing. Objects such as buttons with arrow icons indicate how to navigate in the application windows. It can be specified that the graphics display is built continuously, even if the window with the related graphics is not displayed at a certain time. Pressing the refresh graph button will delete the graphics and start a new representation starting from that sampling moment. At the same time, the graphics limits for the time axis can be set according to the use of the emulator, ranging from an interval in seconds, minutes, hours, and even days.

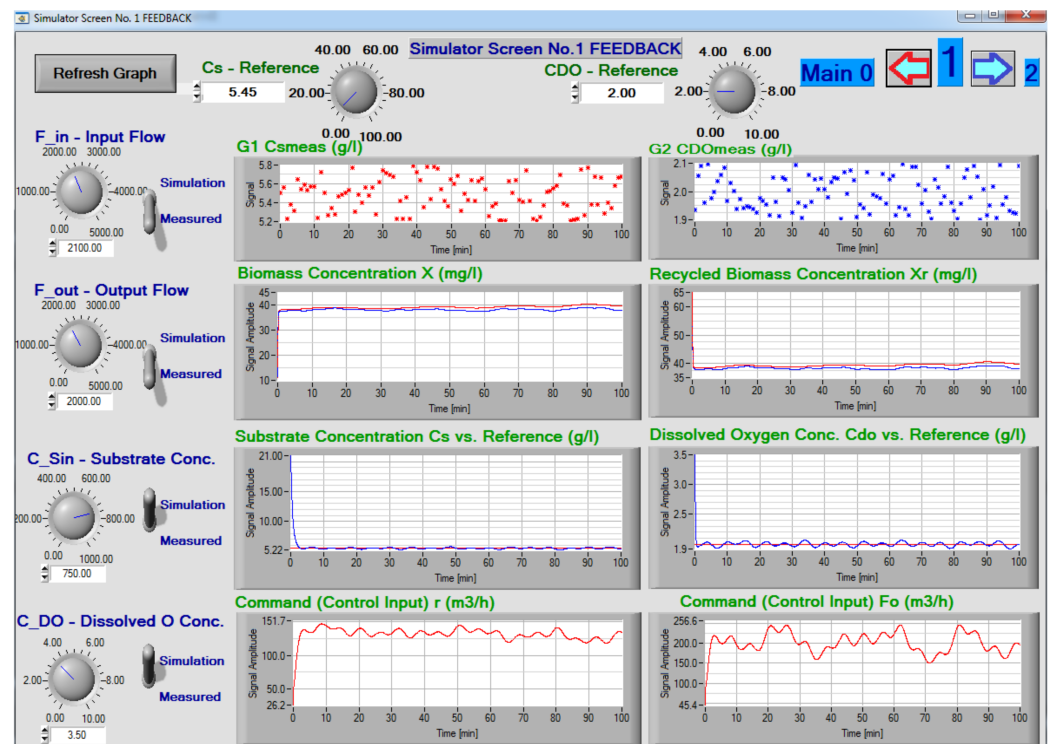


Figure 9. A sample from the source code used for the implementation of the multivariable adaptive control structure of the ASP within the WWTP.

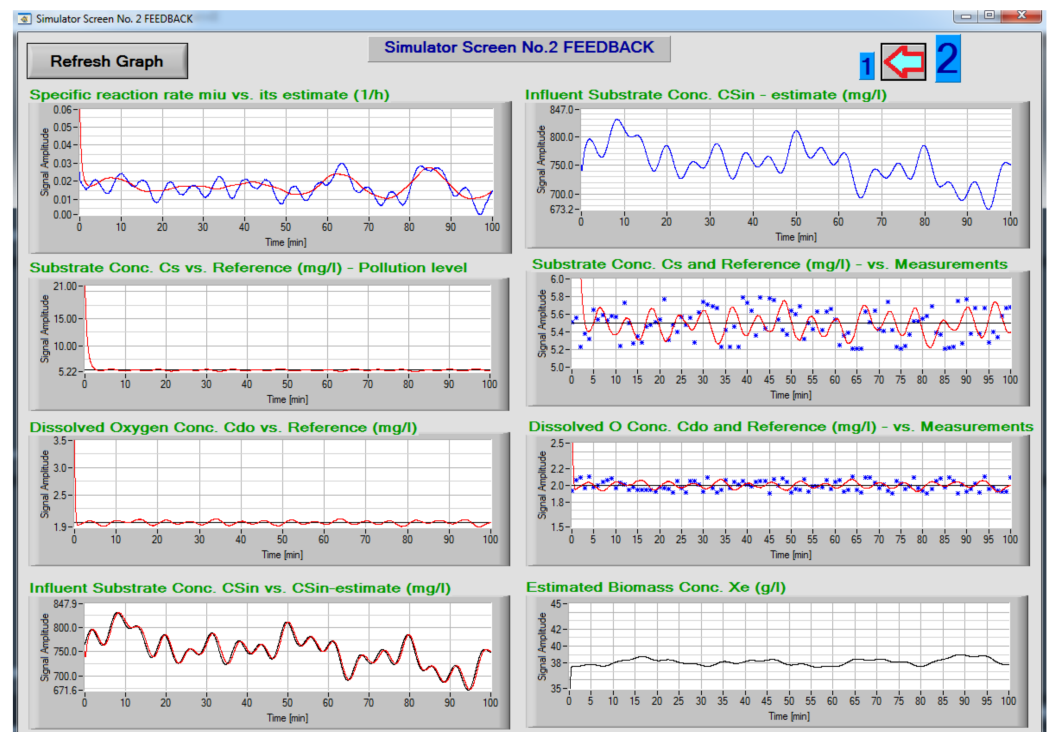


Figure 10. Closed-loop emulator of the adaptive controlled ASP—first screen, with several knobs and graphs for the visualization of the main variables' evolution.


```

##### Emulator software implementation, FEEDBACK working mode / Decoupled multivariable control #####
int Emulator_FEEDBACK()
{
    // Format input signals
    mmxk=mmxk0*(1+km1u*sin((tk*pi)/10)); Ksk=Ks0*(1+Ks*sin(tk*pi/12+pi/2)); Yk=Y0*(1+Y*cos(tk*pi/20));
    Ykol=Y0*(1+Y*cos((tk*T)/pi/20)); Fink=Fin0*(1+Fin1*cos(tk*pi/12)+sin(tk*pi/50)+kFin2*sin(tk*pi/3));
    Sink=Sin0*(1+0.1*sin(tk*pi/20)*cos(tk*pi/80)*cos(tk*pi/50)+0.025*sin(tk*pi/3)+0.02*cos(tk*pi/5));
    Sinmk=0.95*Sin0*(1+0.1*sin(tk*pi/20)*cos(tk*pi/80)*cos(tk*pi/50));
    SinMk=1.05*Sin0*(1+0.1*sin(tk*pi/20)*cos(tk*pi/80)*cos(tk*pi/50));
    D0ink=D0in0*(1+0.1*cos(tk*pi/25)*sin(tk*pi/60)*cos(tk*pi/30)+0.025*sin(tk*pi/3)+0.02*sin(tk*pi/5));
    D0inmk=0.95*D0in0*(1+0.1*cos(tk*pi/25)*sin(tk*pi/60)*cos(tk*pi/30));
    D0inMk=1.05*D0in0*(1+0.1*cos(tk*pi/25)*sin(tk*pi/60)*cos(tk*pi/30));
    Dk=Fink/V; Dsk=Fink/Vs; sak=sk; //Process >>> Estimator parameters >>> Sin
    // Here the values of the measurable quantities Cs and Cdo are "built"
    Csmask=Csmas0+0.6*(-0.5+randomul()); D0mask=D0mas0+0.2*(-0.5+randomul()); tmaskol=(tk/T)*Tmas;
    // ADAPTIVE command >>> S pollutant and DO oxygen are controlled >>> //Commands: r and Fo (WDO)
    //The X and Xr biomasses are estimated, with an asymptotic observer, and the specific reaction rate miu.
    if ((tk/T) > p1) D0stea=D0stea2; if ((tk/T) > p2) D0stea=D0stea3; if ((tk/T) > p3) sstea = sstea2;
    srefk=sstea; dorefk=D0stea;
    //Calculation of control signals, in FEEDBACK mode
    rk=-1/(Dk*sk*(1+czs*randomul()/2.5))*(11*(sstea-sk*(1+czs*randomul()/2.5))-
    -Dk*(Sink-sk*(1+czs*randomul()/2.5))+1/Yk*(miuek-miuS0)*xek);
    Df1=(1-T/Tfd)*Df0+T/Tfd*rk; rk=Df1; if (rk<0) rk=0;
    Wdok=-dok*(1+czo*randomul()/2.5)/(alpha*sk*(1+czs*randomul()/2.5)*(D0max-dok*(1+czo*randomul()/2.5)))+(11*(sst
    +1/(alpha*(D0max-dok*(1+czo*randomul()/2.5)))*(12*(D0stea-dok*(1+czo*randomul()/2.5))+Ko/Yk*(miuek-miuS0)*xek-
    Wdof1=(1-T/Tfo)*Wdof0+T/Tfo*Wdok; Wdok=Wdof1; if (Wdok<0) Wdok=0;
    //Process in FEEDBACK mode
    miuk=mmxk*sk/(Ksk+sk)*dok/(Kdo+dok);
    xkol=xk+T*((miuek-miuS0)*xk-Dk*(1+rk)*xk+rk*Dk*xrk); skol=sk+T*(-(1/Yk)*(miuek-miuS0)*xk-Dk*(1+rk)*sk+Dk*Sink);
    dokol=dok+T*(-Ko/Yk*(miuek-miuS0)*xk-Dk*(1+rk)*dok+alpha*Wdok*(D0max-dok)+Dk*D0ink);
    xrkol=xrk+T*(Dk*(1+rk)*xk-(beta+rk)*Dk*xrk);
    //Estimator for Sin, in FEEDBACK mode
    zSink=-Ko*sk*(1+czs*randomul()/2.5)+dok*(1+czo*randomul()/2.5); eSink=zSink-Sink;
    F1Sink=gama1Sin*sqrt(absolutul(eSink))*semm(eSink)+gama2Sin*eSink;
    F2Sink=gama1Sin*gama1Sin/2*semm(eSink)+3/2*gama1Sin*gama2Sin*sqrt(absolutul(eSink))*semm(eSink)+gama2Sin*
    zSinkol=zSink+T*(-Dk*(1+rk)*zSink-Ko*Dk*Sink+alpha*Wdok*(D0max-dok*(1+czo*randomul()/2.5))+Dk*D0ink*(1+czo*rand
    Sinkol=Sink+T*LSin*LSin*alpha2Sin*Ko*Dk*F2Sink; if (Sinkol<0) Sinkol=0;
    //Estimator for biomass X and Xr, in FEEDBACK mode
    z1kol=z1k+T*(-Dk*(1+rk)*z1k+1/Yk*rk*Dk*z3k+Dk*Sink);
    z2kol=z2k+T*(-Dk*(1+rk)*z2k+Ko/Yk*rk*Dk*z3k+alpha*Wdok*(D0max-dok*(1+czo*randomul()/2.5))+Dk*D0ink*(1+czo*randomul
    z3kol=z3k+T*(Dk*(1+rk)*Yk*(z1k-sk*(1+czs*randomul()/2.5))-(beta+rk)*Dk*z3k);
    xekol=Ykol*(z1kol-skol);
    //Reaction speed estimator for miu, in FEEDBACK mode
    sekol=sek+T*(-1/Yk*(miuek-miuS0)*xek-Dk*(1+rk)*sk*(1+czs*randomul()/2.5)+Dk*Sink+omega1*(sk*(1+czs*randomul()/2.5
    doekol=doek+T*(-Ko/Yk*(miuek-miuS0)*xek-Dk*(1+rk)*dok*(1+czo*randomul()/2.5)+alpha*Wdok*(D0max-dok*(1+czo*randomul
    miuekol=miuek+T*(-1/Yk*xek*gama1*(sk*(1+czs*randomul()/2.5)-sek)-Ko/Yk*xek*gama2*(dok*(1+czo*randomul()/2.5)-doek)
    Df0=Df1; Wdof0=Wdof1;
    //.....Graphic display, other tasks, updating a new cycle to Timer

```

Figure 11. Closed-loop emulator of the adaptive controlled ASP—second screen, with several graphs for the visualization of the main variables' evolution.

The second screen related to the feedback mode is presented in Figure 10. Here, the time profiles of several biological variables can be visualised (and stored), such as the specific reaction rate (versus its estimate), the substrate, and DO concentrations in comparisons with their references and with some measurements, estimated biomass and influent substrate concentrations, respectively, among others.

To design a proper and practical control system, it is important to have an accurate model of the real ASP. This model is fully defined when its parameters are obtained via analytical modelling, measurements, or identification procedures. Moreover, some online estimation algorithm can be used to reconstruct the kinetics and the state variables [14,21].

In the case of the ASP from the WWTP Făcăi, the available measurements are the influent flow rate and two state variables of the model (1): the pollutant concentration C_S and the DO concentration C_{DO} , both measured inside the aerobic tank. The polluted water is injected into the reactor with the flow rate F_{in} (m^3/h), with the influent pollutant concentration being C_{Sin} (mg/L). The air is blown in the reactor with the flow rate F_O (m^3/h), with the concentration of the contained oxygen being C_{DOin} (mg/L).

In Figure 10, some of the obtained results, which allow a partial validation of the simplified model, are presented. Thus, the simulated evolution of the most important variables of the bioprocess, i.e., the outputs C_S and C_{DO} , is shown against the measurements obtained from the process. The estimates of the biomass concentration are obtained using the state observer. Several other profiles of the process variables can also be obtained using the simulator, as can be seen in Figure 10. The parameters of the bioprocess (yield and kinetics) used in the simulator and calibrated from the available measurements are as follows: $\mu_{\max} = 0.56 \text{ h}^{-1}$, $K_S = 100 \text{ mg/L}$, $K_{DO} = 2 \text{ mg/L}$, $Y = 0.65$, $K_O = 0.75$, $\alpha = 0.018$, $\beta = 0.2$, $F_{in}^0 = 2040 \text{ m}^3/\text{h}$, $C_{Sin}^0 = 760 \text{ mg/L}$, $C_{Oin} = 9 \text{ mg/L}$, $C_{DO}^{\max} = 8 \text{ mg/L}$, and $V = 15,000 \text{ m}^3$.

However, it is necessary to perform additional tests and to improve the instrumentation (sensors) in order to obtain more satisfactory results regarding the full calibration of the model.

Figure 11 presents the source code used for the implementation of discretized equations, in the form of iterative relations, which were designed for the multivariable adaptive control structure of the ASP within the WWTP. These iterative relationships were included in the type C function *Emulator_2_Closed_Loop* () and will be processed at each interrupt generated by the timer, according to the set sampling period.

The advantage of this real-time implementation is that it can be connected so as to work in parallel with the process, allowing important parametric corrections. At the same time, real-time signal samples, such as input flow, output flow, dissolved oxygen concentration, or other measured parameters, can become inputs in the application, which allows not only testing the correctness of the model considered for the process, but also the validation of the estimation and control solutions designed for the WWTP Făcăi.

The implementation of the emulator as a real-time application, using the development environment LabWindows/CVI, based on the ANSI C standard, allows the integration on the most diverse hardware equipment available at SCADA level, but also as embedded systems. The migration of the application to an embedded system can be done by directly taking over the source code of the algorithms and creating new software projects or by creating files in compiled form that can be integrated into new software projects and compiled together.

5. Conclusions

This work addressed the design and implementation of an emulator for a controlled bioprocess from a complex WWTP. This simulator was achieved as a software tool that can be integrated into a more complex DCS-SCADA system of the WWTP Făcăi, Romania. This software tool was tailored on the WWTP Făcăi but can be used with some adjustments to other WWTPs.

The activated sludge process dynamical model and a multivariable adaptive control system were integrated in this simulator. The software tool can work in parallel with the evolution of the process and can have as input signals measured directly at the process level. The software emulator was developed in the Labwindows/CVI programming environment. Two main operating modes of the emulator were implemented: (1) simulator of the open-loop system, which includes the model of the ASP and the estimator of the influent substrate concentration; (2) real-time emulator of the closed-loop system, based on the proposed multivariable adaptive control structure of the ASP.

The emulator provides, via specific graphical windows, the evolution of several biological variables of the ASP, such as concentrations of various components, specific rates, input and output flows, and other kinetic parameters of the process. Some of these evolutions can be compared with real data from the process, and several data provided by the sensors can be used for monitoring or control purposes. The reference profiles can be modified depending on the technological requirements. Moreover, the values of process and control parameters could be adjusted within the operation modes of the simulator.

This work can be useful for researchers and practicing engineers, as well as for postgraduate and PhD students interested in bioprocess modelling and control related to wastewater treatment plants.

The obtained results are promising and will be used in future research for the implementation of advanced control systems in the DCS-SCADA of the WWTP.

Author Contributions: Conceptualization, D.S., I.-M.P. and M.R.; data curation, C.Ș.-I. and S.M.; formal analysis, D.S. and M.R.; funding acquisition, D.S. and C.Ș.-I.; investigation, I.-M.P. and C.Ș.-I.; methodology, D.S., I.-M.P. and M.R.; resources, I.-M.P., C.Ș.-I. and S.M.; software, D.S. and I.-M.P.; supervision, D.S.; validation, I.-M.P. and C.Ș.-I.; visualization, D.S., I.-M.P. and M.R.; writing—original draft, D.S. and I.-M.P.; writing—review and editing, D.S., I.-M.P. and M.R. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Competitiveness Operational Program, TISIPRO project, ID: P_40_416/105736, 2016–2021 (European Regional Development Fund).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Iratni, A.; Chang, N.B. Advances in control technologies for wastewater treatment processes: Status, challenges, and perspectives. *IEEE/CAA J. Autom. Sin.* **2019**, *6*, 337–363. [CrossRef]
2. Nascu, I.; De Keyser, R.; Vlad, G. Modelling and control aspects of wastewater treatment processes. *Ecoterra* **2008**, *18*, 27–30.
3. Galloway, B.; Hancke, G.P. Introduction to industrial control networks. *IEEE Commun. Surv. Tutor.* **2013**, *15*, 860–880. [CrossRef]
4. Knapp, E.D.; Langill, J.T. *Industrial Network Security: Securing Critical Infrastructure Networks for Smart Grid, SCADA and Other Industrial Control Systems*, 2nd ed.; Syngress, Elsevier: Waltham, MA, USA, 2015.
5. Selișteanu, D.; Petre, E.; Prejbeanu, R.; Popescu, I.M.; Mehedințeanu, S. Software solutions for simulation, monitoring and data acquisition in wastewater treatment plants. In Proceedings of the 21st International Carpathian Control Conference, High Tatras, Slovak Republic, 27–29 October 2020; IEEE: Piscataway, NJ, USA, 2020.
6. Roman, M.; Popescu, D.; Selișteanu, D. An interactive teaching system for bond graph modeling and simulation in bioengineering. *J. Educ. Technol. Soc.* **2013**, *16*, 17–31.
7. Lyubenova, V.; Ignatova, M.; Kostov, G.; Shopska, V.; Roman, M.; Petre, E. An Interactive teaching system for kinetics modelling of biotechnological processes. In Proceedings of the 22nd International Conference on System Theory, Control and Computing, Sinaia, Romania, 10–12 October 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 366–371.
8. Farza, M.; Chérut, A. CAMBIO: Software for modelling and simulation of bioprocesses. *Bioinformatics* **1991**, *7*, 327–336. [CrossRef] [PubMed]
9. Ferrer, J.; Seco, A.; Serralta, J.; Ribes, J.; Manga, J.; Asensi, E.; Morenilla, J.J.; Llavador, F. DESASS: A software tool for designing, simulating and optimising WWTPs. *Environ. Model. Softw.* **2008**, *23*, 19–26. [CrossRef]
10. Henze, M.; Gujer, W.; Mino, T.; van Loosdrecht, M.C.M. *Activated Sludge Models ASM1, ASM2, ASM2d and ASM3*; IWA Task Group on Benchmarking on Mathematical Modeling for Design and Operation of Biological Wastewater Treatment; IWA Publishing: London, UK, 2000.
11. Luca, L.; Barbu, M.; Ifrim, G.; Caraman, S. Analysis of phosphorus removal performances in a municipal treatment plant. In Proceedings of the International Conference on System Theory, Control and Computing, Cheile Gradistei, Romania, 14–16 October 2015; IEEE: Piscataway, NJ, USA, 2015; pp. 415–420.
12. Sevela, B.; Bertalan, G. Development of a MATLAB based bioprocess simulation tool. *Bioproc. Eng.* **2000**, *23*, 621–626. [CrossRef]
13. [www.ni.com: LabWindows/CVI—National Instruments](https://www.ni.com/labwindows/cvi—National Instruments). Available online: <https://www.ni.com/ro-ro/support/downloads/software-products/download.labwindows-cvi.html#353603> (accessed on 14 October 2021).
14. Dochain, D. *Automatic Control of Bioprocesses*; ISTE and John Wiley & Sons: London, UK, 2008.
15. Petrides, D.; Carmichael, D.; Siletti, C.; Koulouris, A. Biopharmaceutical process optimization with simulation and scheduling tools. *Bioengineering* **2014**, *1*, 154–187. [CrossRef] [PubMed]
16. Selișteanu, D.; Sendrescu, D.; Georgeanu, V.; Roman, M. Mammalian cell culture process for monoclonal antibody production: Nonlinear modelling and parameter estimation. *BioMed Res. Int.* **2015**, *2015*, 598721. [CrossRef] [PubMed]
17. Dochain, D.; Vanrolleghem, P. *Dynamical Modelling and Estimation in Wastewater Treatment Processes*; IWA Publishing: London, UK, 2001.
18. Solon, K.; Volcke, E.I.P.; Spérandio, M.; van Loosdrecht, M.C.M. Resource recovery and wastewater treatment modelling. *Environ. Sci. Water Res. Technol.* **2019**, *5*, 631–642. [CrossRef]
19. Nejari, F.; Dahhou, B.; Benhammou, A.; Roux, G. Non-linear multivariable adaptive control of an activated sludge waste-water treatment process. *Int. J. Adapt. Control Signal Proc.* **1999**, *13*, 347–365. [CrossRef]
20. Norzain, N.A.; Sahlan, S.; Halim, F.A.; Wahab, N.A. Empirical modelling of activated sludge process via system identification. *J. Teknol.* **2016**, *78*, 155–162. [CrossRef]
21. Petre, E.; Selișteanu, D. A multivariable robust-adaptive control strategy for a recycled wastewater treatment bioprocess. *Chem. Eng. Sci.* **2013**, *90*, 40–50. [CrossRef]
22. Pisa, I.; Santin, I.; Vicario, J.L.; Morell, A.; Vilanova, R. ANN-based soft sensor to predict effluent violations in wastewater treatment plants. *Sensors* **2019**, *19*, 1280. [CrossRef] [PubMed]
23. Vilanova, R.; Santin, I.; Pedret, C.; Barbu, M. Event-based control for dissolved oxygen and nitrogen in wastewater treatment plants. In Proceedings of the 22nd International Conference on System Theory, Control and Computing, Sinaia, Romania, 10–12 October 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 212–217.
24. Tebbani, S.; Lopes, F.; Filali, R.; Dumur, D.; Pareau, D. Nonlinear predictive control for maximization of CO₂ bio-fixation by microalgae in a photobioreactor. *Bioproc. Biosyst. Eng.* **2014**, *37*, 83–97. [CrossRef] [PubMed]
25. Petre, E.; Selișteanu, D.; Șulea-Iorgulescu, C.; Mehedințeanu, S. Multivariable adaptive control strategy for an activated sludge process inside a wastewater treatment plant. In Proceedings of the 21st International Carpathian Control Conference, High Tatras, Slovak Republic, 27–29 October 2020; IEEE: Piscataway, NJ, USA, 2020.

-
26. Morales-Rodelo, K.; Francisco, M.; Alvarez, H.; Vega, P.; Revollar, S. Collaborative Control Applied to BSM1 for Wastewater Treatment Plants. *Processes* **2020**, *8*, 1465. [[CrossRef](#)]
 27. Lyubenova, V.; Kostov, G.; Denkova-Kostova, R. Model-Based Monitoring of Biotechnological Processes—A Review. *Processes* **2021**, *9*, 908. [[CrossRef](#)]
 28. Robles-Rodriguez, C.E.; Ben-Ayed, A.; Bernier, J.; Rocher, V.; Dochain, D. Management of an integrated network of wastewater treatment plants for improving water quality in a river basin. *IFAC PapersOnline* **2019**, *52*, 358–363. [[CrossRef](#)]
 29. Selișteanu, D.; Popescu, I.M.; Petre, E.; Roman, M.; Șendrescu, D.; Popa, B. Distributed control systems for a wastewater treatment plant: Architectures and advanced control solutions. In *Wastewater and Water Quality*; Yonar, T., Ed.; InTech Publishing: London, UK, 2018; pp. 153–181.
 30. Buttazzo, G.C. *Hard Real-Time Computing Systems: Predictable Scheduling Algorithms and Applications*, 3rd ed.; Springer: Boston, MA, USA, 2011.
 31. Leveson, N.G. *Safeware: System Safety and Computers*, 5th ed.; Addison-Wesley: Boston, MA, USA, 2001.