A Continuous Stirred Tank Reactor Simulator for Fault Diagnosis Experiments

Abstract

Process simulation benchmark processes are available for fault diagnosis experiments, for instance, the Tennessee Eastman chemical process simulation software. These lack however, in general, a comprehensive analytic description understandable to researchers outside a specialized area, especially fluid dynamics from mechanical engineering, stoichiometry from chemistry and reaction kinetics from chemical engineering. In this work a model of a benchmark process based on two chemical engineering doctoral theses from the Massachusetts Institute of Technology is extensively explained and improved to allow a sufficiently profound understanding of the chemical reaction and fluid dynamics, reducing the simulation model to its necessary complexity, and allowing a consistent reasoning between faults and their manifestations in the system. The process is a continuous stirred tank reactor (CSTR), involving hydraulics, chemistry, chemical engineering and control theory. The general goal of this work is to improve the partially heuristic framework of the CSTR by consolidated theory of the components of the model. Moreover, the aim is to provide a tutorial-like experimental platform for model-based and model-free fault diagnosis where the model faults are easily configured and analyzable.

Keywords: Fault diagnosis, chemical engineering, simulation, CSTR.

1. Introduction

The importance of the availability of chemical process simulators in order to study fault diagnosis methods is permanently confirmed. The Tennessee-Eastman (TE) chemical process (Downs and Vogel, 1993; Chiang et al., 2001) is a prominent example. The software is about 25 years old, but is continuously used as the experimental benchmark of choice in sophisticated data-driven process condition detection techniques (Rato and Reis, 2013; Chen et al., 2014; Askarian et al., 2016; Yin et al., 2012, 2014; Aldrich and Auret, 2013; Gao and Hou, 2016; Yin et al., 2015; Chiang et al., 2015; Monroy et al., 2010, 2012; Maestri et al., 2010; Robertson et al., 2015). However, it deliberately lacks the underlying description of the mass and heat balances, the control algorithms and the simulation framework. The TE source code is available, however in an obfuscated way. Researchers naturally prefer unconditional access to the mathematical groundwork of the simulation. **?** confirm that an explicit mathematical representation of the process is not given and instead the simulation is distributed as purposely convoluted FORTRAN code. In **?** the necessity for a more transparent and flexible TE model is postulated, proposing several improvements to the code, deciphering the source in a certain way, but the analytic description of the TE simulator is still missing.

An alternative chemical process simulator of comparable complexity is a Continuous Stirred Tank Reactor (CSTR), created in the context of two doctoral theses (Finch, 1989; Oyeleye, 1990; Oyeleye and Kramer, 1988), elaborated at the Chemical Engineering Department of the Massachusetts Institute of Technology. The simulator will be denominated as the MIT-CSTR. Its analytic description provides complete transparency of how the simulation model works and consequently allows the researcher a deeper understanding of the process dynamics, leading to more consolidated affirmations about his fault diagnosis methods. Similar simulators derived from this work were later used, e.g. Sorsa et al. (1991), Sorsa and Koivo (1993), Zhang and Morris (1994), Zhang et al. (1996), Venkatasubramanian et al. (1990). The MIT-CSTR simulator probably was not used more extensively because it stems from a pre-internet period with no online availability of the source code, hence the TE simulator dominated as a benchmark platform for fault diagnosis experiments. The complete understanding of the MIT-CSTR requires knowledge from the area of chemistry, chemical engineering and hydraulics from mechanical engineering. The MIT-CSTR provides an elaborated model of the involved components and dynamic behaviour of the process variables. There are however two important aspects that are improved in this work. The description of the process is expanded to permit a self containing tutorial for researchers not familiar with the chemical and hydraulic theory. Moreover, the heuristic parts of the simulator are replaced by theoretically more sound methods with a low degree of complexity. For instance a simple model for the control valves is proposed and the simultaneous nonlinear system of the process variables is solved by a Newton-Raphson iterative algorithm, substituting the model inspired in Kirchhoff laws from electricity. The main contribution of this paper is a complete and improved theory of all components of a CSTR simulator with just the right amount of complexity, the substitution of heuristic methods and the complete re-implementation of the simulator in order to provide a experimental fault diagnosis benchmark. The rest of the paper is organized in the following manner: Section 2 introduces the proposed object of study, the CSTR simulator. In section 3 the necessary hydraulic groundwork is laid to model the fluid dynamics, section 4 explains the chemical reaction that is performed

in the CSTR and section 5 instantiates the theory in the concrete case of the CSTR simulator. Case studies are presented in section 6 and finally the conclusions are drawn in section 7.

2. CSTR Simulator Model for Fault Diagnosis Experiments

Fig. 1 shows the general process flow diagram of the MIT-CSTR simulator. Its core is a CSTR where two globally exothermic, competing reactions are transforming reactant A into products B and C. Species A with concentration c_{A0} at temperature T_1 is entering the reactor with flow rate Q_1. The products and rest of the reactant are pumped out of the tank, with flow rate Q_2 which is bifurcating to an effluent pipe with flow rate Q_4 out of the system and an eventual leak with flow rate Q_3. The level L of the CSTR is controlled by a valve as an actuator. A jacket around the CSTR is responsible for keeping its temperature T_2 constant. A cascade controller, actuating on a second valve varies the inflow of the cooling liquid at temperature T_3 and flow rate Q_5. Leaks from the jacket to the exterior at flow rate Q_7 and from the jacket into the CSTR at flow rate Q_6 can be introduced into the simulation as faults. The coolant inside the jacket has temperature T_4 and is leaving the system with flow rate Q_8. There is a total of 14 measured variables (in circles), used as features x_i for fault diagnosis. Moreover, four constraints are calculated as additional features, c.f. table 1. White Gaussian noise, adjusted to the dimension of each variable was added independently. In order to allow reproducibility of the experiments, a fixed seed for the random number generator is provided in the simulation configuration. All parameters of the simulator are resumed in table 2. Each of the involved hydraulic units and the chemical reaction model will be described in detail in section 3 and section 4 respectively.

Table 1: Measured variables and constraints of the CSTR simulator

#	Variable/Constraint name	Acronym	Nominal value	Units
1	Feed concentration	c_{A0}	20.0	mol/m^3
2	Feed flowrate	Q_1	0.25	m^3/s
3	Feed temperature	T_1	30.0	°C
4	Reactor level	L	2.0	m
5	Product A concentration	c_A	2.85	mol/m^3
6	Product B concentration	c_B	17.114	mol/m^3
7	Reactor temperature	T_2	80.0	°C
8	Coolant flowrate	Q_5	0.9	m^3/s
9	Product flowrate	Q_4	0.25	m^3/s
10	Coolant inlet temperature	T_3	20.0	°C
11	Coolant inlet head	h_7	10.0	m
12	Level controller output	m_1	10.16	%
13	Coolant flow controller output	m_2	61.0	%
14	Coolant flow controller setpoint	u_2	0.907	m^3/s
15	Inventory	z_1	0.0	m^3
16	Mol balance	z_2	0.0	mol
17	Cooling water head loss	z_3	0.0	m
18	Effluent head loss	z_4	0.0	m

3. Fluid Dynamics for a Chemical Reactor

For the purpose of this simulator, a sufficient analysis of the fluid dynamics in a chemical reactor is the knowledge of all involved *hydraulic heads* h and all *volumetric flows* Q. The considered components in the system are reactor tanks, pipes, pumps, valves and leakages which define a simultaneous system of nonlinear equations in which the heads and flows are the unknown variables. Tanks and pumps add energy to the fluid (head gain), pipes, valves and leakages subtract it from the fluid (head loss). The head difference $\Delta h = h_{\text{before}} - h_{\text{after}}$ between a head before and after a component, is positive in the case of a head loss, and negative in the case of a head gain. The hydraulic head[1] h[m] for an incompressible fluid is an appro-

Table 2: CSTR simulator system variables, units and nominal values in square brackets, c.f. fig. 1

$\Delta H_B, \Delta H_C$	Primary and secondary heat of reaction [kJ/kmol] [30000.0] [-10000.0]
$\Delta t = t_{k+1} - t_k$	Sample interval of the simulator [0.02 min]
ρ, ρ_c	Density of reactor liquid and coolant [1000 kg/m^3]
A_P	Cooling circuit pipe cross sectional area [0.8 m^2]
A_H	Heat exchange area between jacket and reactor [m^2]
A_P	Pipe cross sectional area [0.011 m^2]
A_R	Floor area of reactor [1.5 m^2]
c_{A0},c_A,c_B,c_C	Concentrations [mol/m^3]: Feed [20.0], Reactant A [2.85], Product B [17.114], Product C [0.0226]
C_p, C_{p_c}	Specific heat capacity of reactor liquid and coolant [4.2 kJ/(kg °C)]
E_B, E_C	Primary and secondary activation energy [kJ/kmol] [25000.0] [45000.0]
h_0, h_1, h_2, h_3	Heads [m]: Pump differential [47.0], reactor outlet [2.0], pump exit [5.0], outlet pipe [2.0]
h_4, h_5, h_6, h_7	Outlet valve [1.0], effluent pipe [1.0], pump leak [0.1], coolant feed [10.0]
h_8, h_9, h_{10}, h_{11}	Coolant pipe [5.0], Coolant valve [2.0], jacket [1.0], jacket effluent [1.0]
k_{0_B}, k_{0_C}	Primary and secondary Arrhenius rate constant pre-exponential factor [1/min] [2500.0], [3000.0]
K_1, K_2, K_3, K_4	Minor loss coefficients: reactor exit pipe [100.0], pump leak [10^6], level control valve [19.85], effluent pipe [500.0]
K_5, K_6, K_7, K_8	coolant feed pipe [72.0], coolant feed valve [45.95], jacket to tank leak [10^6], jacket to exterior leak [10^6],
K_9, K_{10}	blockage in jacket [10^6], coolant effluent pipe [65.0]
K_{11}, K_{12}	pump flow [1.0], pump major [2.0]
$L = V/A_R$	Liquid level of reactor [2.0 m]
m_1,r_3,m_2	Controller output signals [0.1016], [0.61], [0.907]
$\dot{q}_c$	Heat transfer from reactor to jacket [38020 W]
$\dot{q}_{\text{ext}}$	External heat source (sink) [0 W]
Q_1, Q_2, Q_3, Q_4	Flows [m^3/min]: reactant feed [0.25], reactor exit [0.25], effluent leak [0.0], effluent [0.25]
Q_5, Q_6, Q_7, Q_8	coolant feed [0.9], jacket to reactor leak [0.0], jacket to environment leak [0.0], jacket effluent [0.9]
r_A, r_B, r_C	Reaction rates [1/min]
r_1, r_2, r_3	Set points of the three controllers: reactor level, reactor temperature, coolant flow rate [2.0,80.0,0.9]
T_1, T_2, T_3, T_4	Temperatures [°C] of reactant feed [30], reactor [80], coolant feed [20], coolant in jacket [40]
K_p, T_i, T_d	Controller proportional gain [-], integral time [s], derivative time [s]: Level [0.3], [0.1], [0.15]; Temperature [0.5], [2.0], [0.25]; Flow [0.15], [0.01], [0.035]
UA_H	heat transfer coefficient U, multiplied by A_H [1901 kJ/(min °C)]
V	Reactor volume [3.0 m^3]

[1] When a variable is introduced, its units are written in brackets.

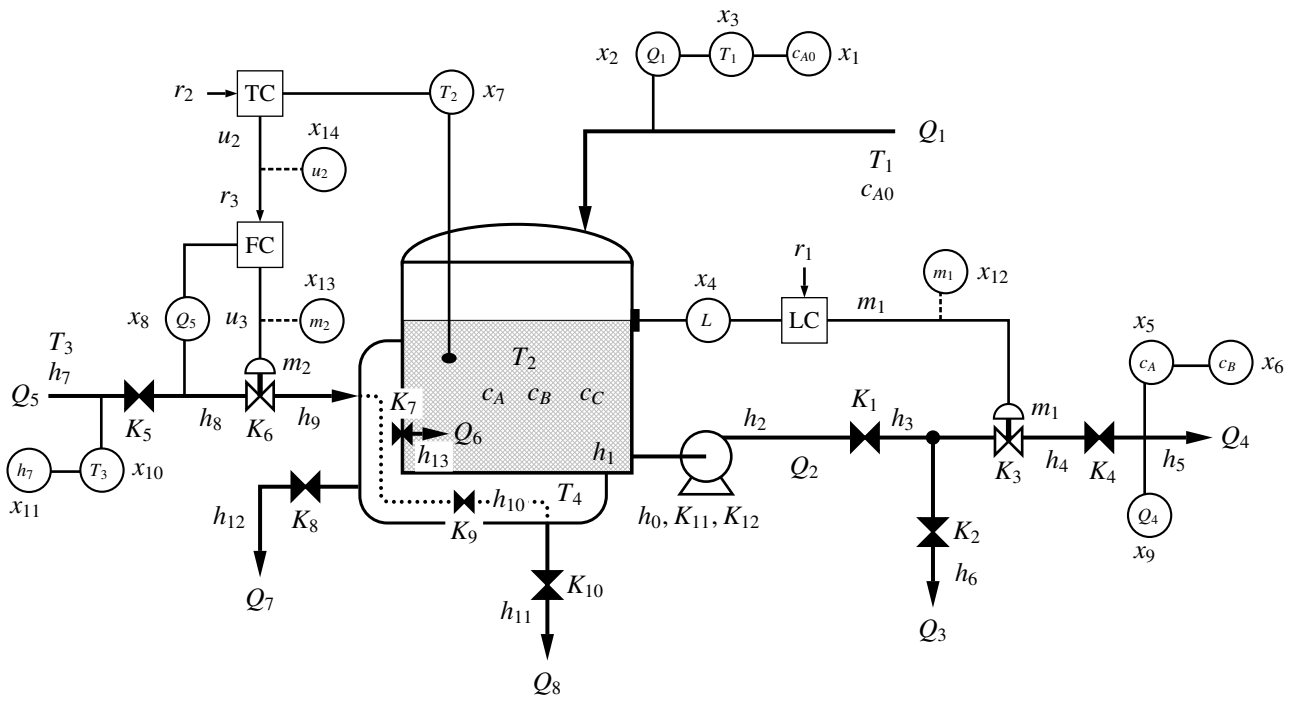

Figure 1: The Process Flow Diagram of the CSTR simulator defined in Finch (1989); Oyeleye (1990). The original model with flow resistances 'R' was improved by a model with minor loss coefficients 'K' of the hydraulic components. Pressures were substituted by heads 'h'. There are 14 measured variables, circled in the diagram, which form the first 14 components $x_i, i = 1, \ldots, 14$ of the feature vector $\mathbf{x}$ used for condition monitoring.

priate measure of energy per unit weight. The Bernoulli equation for incompressible smooth flow (Young et al., 2006) under adiabatic[2] conditions states that the total energy of the fluid is constant. It is composed of three components, the energy due to its pressure, the kinetic energy of the fluid, and the static energy due to its elevation relative to a reference level. When the Bernoulli equation is divided by a fixed local gravitation $g\left[\frac{\mathrm{m}}{\mathrm{s}^2}\right]$, the equivalent total energy head h, composed by the pressure head h_p, velocity head h_v, and static head h_z is

$$h = h_p + h_v + h_z = \frac{P}{\rho g} + \frac{v^2}{2g} + z \tag{1}$$

where $P\left[\mathrm{Pa} = \frac{N}{\mathrm{m}^2} = \frac{\mathrm{kg}}{\mathrm{ms}^2}\right]$ is the pressure of the fluid, $\rho\left[\frac{\mathrm{kg}}{\mathrm{m}^3}\right]$ its the density, $v\left[\frac{\mathrm{m}}{\mathrm{s}}\right]$ its mean velocity and z[m] its elevation. In the context of the chemical process simulator, the velocity and static heads are ignored before and after a hydraulic component. In a pump, for instance, it is assumed that the inlet and outlet are on the same level and that the fluid velocity after entering and exiting the pump are the same. Hence the remaining relevant head is the pressure head $h = h_p$. This allows the equivalent expression of energy for a particular fluid at constant gravitational conditions as either head or pressure, connected as

$$P = \rho g h.$$

The task is now to analyze the property of each of the involved components that interacts with the fluid. Either a fixed head must be available, for instance that of a tank, or a functional relationship between the head difference and the corresponding flow has to be defined, for instance at both ends of a valve or a pipe.

3.1. Hydraulic Components

A model that incorporates all aspects of how a fluid transits through all involved entities in a chemical reactor is far too complex for the purpose of fault diagnosis. A pipe, for instance, had to consider its shape and length, its material, obstacles like bends, or the density and viscosity of the fluid, which itself depend on the temperature. Besides one must know if the flow is laminar or turbulent, defined by the Reynolds number, for which no general closed form formula exists. In fact, the Reynolds number is the functional relation of the inertial forces of a liquid to its viscous forces. This function again involves the velocity along an equivalent distance, and viscosity and density of the fluid. Researchers from outside the field of fluid dynamics, especially electrical engineering and computer science, should have access to a low abstraction level that permits an easy understanding of the involved component, but simultaneously flexible enough to capture with sufficient fidelity the function of that component. In the following, a consistent model is defined that provides the necessary abstraction to calculate the heads and flows within the system. Even with this limited complexity, a highly nonlinear, sophisticated simulator can be build.

[2]No transfer of heat or matter

3.1.1. Reactor Tank

The liquid in the CSTR adds head to the system based on the potential energy of the tank contents (head gain). It is simply

$$h^{\text{tank}} = -L, \tag{2}$$

where $L\,[m]$ is the height of the liquid column of the reactor.

3.1.2. Pipe Flow

For horizontal, incompressible flow of a liquid with volume $V\left[\text{m}^3\right]$ within a pipe, the temporal[3] relation of volumetric flow $Q\left[\frac{\text{m}^3}{\text{s}}\right]$, defined as

$$Q = \dot{V} = \frac{dV}{dt} \tag{3}$$

to the difference of head $\Delta h = h_i - h_j$ at both ends of the pipe segment under consideration is derived from the Darcy-Weisbach equation (Çengel and Cimbala, 2013; Young et al., 2012, 2006)

$$\Delta h = f\frac{L}{D}\frac{v^2}{2g}, \tag{4}$$

where f is the dimensionless Darcy friction factor, $L\,[\text{m}]$ and $D\,[\text{m}]$ are the length and diameter of the pipe and $v\left[\frac{\text{m}}{\text{s}}\right]$ is the average velocity of the fluid. Considering the cross sectional area $A\left[\text{m}^2\right]$ of the pipe, (4) establishes the necessary connection between the head difference Δh and the flow

$$Q = vA \tag{5}$$

as

$$\Delta h = f\frac{L}{D}\frac{Q^2}{2gA^2}. \tag{6}$$

This head loss in a pipe falls into the category of *major losses* and is incorporated into the dimensionless *loss coefficient* K_L (Young et al., 2006), aka *resistance coefficient* (Idelchik and Steinberg, 1996)[4] which is defined as

$$K_L = f\frac{L}{D}. \tag{7}$$

The index of K emphasizes the fact that the major loss of a pipe is directly proportional to its length L and inversely proportional to its diameter. When the Darcy-Weisbach equation (4) is specialized for a pipe with fully developed laminar flow, i.e. with negligible turbulence, it turns into the Hagen-Poiseuille equation which basically states that the head loss has a *linear* relationship $\Delta h \propto Q$ to the flow. This is due to the fact that the Darcy friction factor f is inversely proportional to the Reynolds number Re which by itself is proportional to the velocity and hence the flow of the fluid. This ideal linear relationship is not appropriate for a simplified and robust model. For the use in a simulator, the pipe is considered as a system where only the influence of its geometry is considered, not the influence of the fluid dynamics (Young et al., 2006). This permits to include geometric elements like elbows, bends and tees. The head loss falls now into the category of *minor losses*. Hence for a particular piece of pipe a constant loss coefficient (7) can be assumed as

$$K^{\text{pipe}} = \text{const.}$$

Obviously, considering (7), longer pipes (high value of L) or narrower pipes (small value of D) should have a higher value of K^{pipe} than short or large[5] pipes. Also, an obstacle to flow, like a bend should elevate the loss coefficient. Moreover, obstructions due to faults along a flow direction can also be modelled in this way. In fact, in this simulator an obstruction fault in the cooling jacket is simulated by elevating a loss coefficient K. Since the flow of the liquid is often controlled by a valve, and hence imperfections of the pipe model are compensated, the only remaining task of the simulator designer is the definition of a fixed 'reasonable' value of K^{pipe} for a certain pipe element of a system, since its geometry parameters (L, D, A) are invariant. Therefore the head loss (6) of a pipe segment can be abstracted as

$$\Delta h^{\text{pipe}} = K^{\text{pipe}}Q^2. \tag{8}$$

It should be noted that the omission of the term $1/(2gA^2)$ from (6) to define (8) creates an inconsistency in the units, i.e. K would be in $\left[\frac{\text{s}^2}{\text{m}^5}\right]$. This does however not make much sense, such that the right hand side of (8) is always multiplied implicitly with a unit value of $1\text{s}^2/\text{m}^5$ and the loss coefficient K^{pipe} is kept dimensionless.

3.1.3. Effluent Pipe Loss Coefficient

At the exit of a pipe, an additional head loss occurs. Considering (1) and (5), the head loss due to the kinetic energy of the fluid is

$$h_v = \frac{1}{2gA^2}Q^2.$$

Hence at the exit of a pipe with area A, the effluent loss factor can be defined as

$$K^{\text{eff}} = \frac{1}{2gA^2},$$

which relates the head loss and flow at the exit of a pipe as

$$\Delta h^{\text{eff}} = K^{\text{eff}}Q^2.$$

A fault that would model an obstruction at the end of a pipe can consistently be implemented by raising this loss factor, equivalent to a reduced pipe area.

3.1.4. Flow Obstruction

Generally, obstacles that diminish the head along a certain flow direction can easily be modeled as minor losses. For instance a fault caused by an obstruction within the cooling jacket falls into this fault category. Hence

$$\Delta h^{\text{obstruc}} = K^{\text{obstruc}}Q^2.$$

[3] For any variable y, the time derivative in Leibniz's notation dy/dt is written in Newton's 'over-dot' notation $\dot{y}$ from now onward.

[4] The resistance coefficient in Idelchik and Steinberg (1996) is denoted as *coefficient of fluid resistance* with the symbol ζ and the *pressure loss* $\Delta H_{\text{sum}}\left[\frac{\text{kg}}{\text{m}^2}\right]$ is equivalent to $\Delta P/g = \rho\Delta h$.

[5] In fact, considering the denominator A^2 in (6), the dependence on the diameter D is to the fifth degree.

3.1.5. Control Valve

In the envisioned simulator application, it does not aggregate useful information when the valve is specified in all its details (Dickenson, 1999). As an example, the hydraulic simulation environment of MATLAB, when choosing a ball valve, three geometric parameters and several flow related parameters must be provided, which are partially very difficult to specify reliably. In a real world scenario, the control valve is chosen by its *flow coefficient* C_v (*flow factor* K_v when dealing with metric units) (Green and Perry, 2007; Controls, 2005; Nayyar, 2000) which depends on the type of valve, and more importantly, on the *valve travel range*, i.e. the percentage of its opening.

A basic model of how the liquid is controlled within the system is sufficient to satisfactorily consider the conditions of the process. For instance when a valve is used to control the flow rate at a certain position of the process, it is not essential for the fault simulation if it is a gate valve, globe valve, or any other instantiation of this flow regulator. In order to build a reasonable model, it is sufficient to know how the flow of the liquid is affected when the valve is gradually opened or closed by varying the output parameter of the controller. Hence the ideal model for the simulator designer is a valve that is controlled by a single parameter, the percentage of how much the valve is open, the travel. Considering the heuristics elaborated for different types of valves (Idelchik and Steinberg, 1996), it can be observed that the basic relationship of the loss coefficient to the valve's contraction diameter is expressed by the proportionality

$$K^{\text{valve}} \propto \frac{1}{\left(\frac{h}{D_0}\right)^2},$$

where $D_0\,[\text{m}]$ is the diameter of the passage cross-section of the valve and $h\,[\text{m}]$ is the lift of the valve. Here the valve travel range $m\,[\%]$ can be defined as

$$m = \frac{h}{D_0}.$$

The range of loss coefficients for fully opened valves in standard textbooks (Green and Perry, 2007; Young et al., 2006) is highly dependent on the type of valve. Values of 0.05 for a ball valve, 0.24 for a butterfly valve and 10.0 for a globe valve suggest that for the purpose of a simple model of a valve in a fault diagnosis simulation, the loss coefficient for an unspecified valve can plausibly be set to

$$K^{\text{valve}} = \frac{1}{m^2}, \tag{9}$$

and hence the head loss caused by a valve is

$$\Delta h^{\text{valve}} = K^{\text{valve}} Q^2. \tag{10}$$

The choice of (9) further implies a linear inherent flow characteristic of the valve (Controls, 2005; Green and Perry, 2007), i.e. the travel parameter m is linearly correlated to the flow rate, c.f. (10). What was said about the inconsistency of the units in the case of a pipe is adopted also for the case of a valve.

3.1.6. Leaks

Especially for the purpose of fault simulation, the occurrence of a leak in a process component is necessary. This defect is easily modelled by introducing an additional zero-length pipe with a virtual valve that simulates no leak with a fully closed valve and total leaking by a fully opened valve, hence

$$K^{\text{leak}} = K^{\text{valve}}.$$

3.1.7. Pump

As the basic relationship between the liquid before ('in') and after ('out') the pump, consider the energy conservation based on the Bernoulli equation for incompressible smooth flow (Young et al., 2006) under adiabatic conditions. The total head h of the liquid can be separated into four parts, the velocity head of the fluid h_v (its kinetic energy), the elevation head h_z due to differences in altitude, the pressure head h_p which is the net contribution of the pump and the head loss h_L which is the energy absorbed within the pump, hence

$$h = h_v + h_z + h_p - h_L = \frac{v_{\text{in}}^2 - v_{\text{out}}^2}{2g} + (z_{\text{in}} - z_{\text{out}}) + \frac{P_{\text{in}} - P_{\text{out}}}{\rho g} - h_L, \tag{11}$$

where $\rho\left[\frac{kg}{m^3}\right]$ is the density of the fluid, $v\left[\frac{m}{s}\right]$ its velocity and $z\,[m]$ its elevation. If the dynamic pressure difference ($v_{\text{in}} \approx v_{\text{out}}$) and the elevation difference ($z_{\text{in}} \approx z_{\text{out}}$) are neglected, the head difference, based on (11) is

$$\Delta h^{\text{pump}} = -h_p + h_L.$$

The most common pump in chemical engineering is the centrifugal pump. It is assumed that the pump is not controlled, i.e. running at a constant speed. Based on the Euler turbomachine equation and under the simplifying condition of no tangential fluid velocity when entering the centrifugal pumps impeller, the pressure head of the pump is a linear function of the flow rate Q of the liquid (Young et al., 2006), with only two parameters h_0, K_1, basically defined by the rotational speed of the pump and its geometry as

$$h_p = h_0 - K_1 Q,$$

where h_0 is a constant energy injection to the fluid is form o a head gain and K_1 is a loss coefficient proportional to the flow. This linear relationship between the head and the flow is turned into a nonlinear relationship by all the minor losses that occur within the pump due to the mechanical and hydraulic friction, expressed in the head loss h_L. An exact relationship for the total head, ignoring velocity and elevation differences that the pump will add to the system as a function of the flow Q, is the pump head-flowrate curve (Young et al., 2006), aka pump performance curve (Mackay, 2004).

Again, for the purpose of fault diagnosis, a model that considered all types of losses, originating from mechanical and hydraulic losses is too complex. In the same sense as the losses for pipes and valves, a minor loss coefficient for a pump could

be defined that establishes a quadratic relationship between the head loss and the flow rate as

$$\Delta h = K_2^{\text{pump}} Q^2.$$

Resuming, the head gain of the pump, assuming that $-h_0$ is the dominant contribution, can be modeled as

$$\Delta h^{\text{pump}} = -h_0 + K_1^{\text{pump}} Q + K_2^{\text{pump}} Q^2.$$

Table 3 compiles the relationships between the head difference and the flow of all hydraulic units involved in the simulator. It can be observed that the equations at at most quadratic.

Table 3: Head gains and losses of all hydraulic units used in the fault simulator. Superscripts: R=Reactor tank, P=Pipe, E=Pipe effluence, O=Obstruction, PU=Pump, V=Valve, L=Leak.

Unit	Function	Domain, constraints
Reactor Tank	$h^{\text{R}} = -L$	
Pipe	$\Delta h^{\text{P}} = K^{\text{P}} Q^2$	$K^{\text{P}} > 0$
Pipe Effluence	$\Delta h^{\text{E}} = K^{\text{E}} Q^2$	$K^{\text{E}} > 0$
Flow Obstruction	$\Delta h^{\text{O}} = K^{\text{O}} Q^2$	$K^{\text{O}} \geq 0$
Valve	$\Delta h^{\text{V}} = K^{\text{V}} Q^2, K^{\text{V}} = \frac{1}{m^2}$	$m \in [0\%, 100\%]$
Pump	$\Delta h^{\text{PU}} = -h_0^{\text{PU}} + K_1^{\text{PU}} Q + K_2^{\text{PU}} Q^2$	$K_1^{\text{PU}} > 0, K_2^{\text{PU}} > 0$
Leak	$\Delta h^{\text{L}} = K^{\text{L}} Q^2, K^{\text{L}} = \frac{1}{[m^{\text{L}}]^2}$	$m^{\text{L}} \in [0\%, 100\%]$

3.2. Hydraulic model

It is assumed that the dynamics of the liquid within the system is a quasi-steady closed-conduit flow (Novak et al., 2010). The liquid is circulating in the system, eventually branching to different parts or joining in a confluence, thus changing flow values and heads.

3.2.1. Serial Flow and Parallel Flow

Additional information about the system is obtained by distinguishing how the material is moving. In a sequence of units, the flow Q does not change. For instance, leaving a tank, passing through a pipe, then a pump, then a valve, the flow is constant along this path. What changes are all involved heads h before and after a unit. Whenever a branching of the current flow Q occurs, an additive separation into two or more new flows $Q_1, Q_2, \ldots$ must be considered. Hence, in a bifurcation

$$Q = Q_1 + Q_2.$$

On the other hand, the head h stays invariant in a branching of the flow. For instance in fig. 1, the coolant flow Q_5 stays the same until entering the jacket, but head losses occur for the pipe K_5 and valve K_6. Another example is that Q_2 when leaving the tank is split into the leak flow Q_3 and effluent flow Q_4, so $Q_2 = Q_3 + Q_4$.

3.2.2. Simultaneous System Variables

In the simulator model, within the limits of a independent subsystem, the constraints among all heads and flows must be satisfied *simultaneously*. An appropriate method to obtain all desired parameters is to establish a system of simultaneous equations and solve it. Since the functions that connect heads and flows are nonlinear (quadratic at most in this case, cf. table 3), a nonlinear system of equations must be solved. This problem is a simplified version of a quasi-steady closed-conduit flow analysis (Jeppson, 1974; Novak et al., 2010), for instance, in a water supply network where extensive branching of the system can occur and even the change of flow direction in pipe loops is possible. For the studied simulator, flow directions are invariable, branching is rare and the hydraulic network is not too complex.

All p flows $Q_1, Q_2, \ldots, Q_p$ and all q heads $h_1, h_2, \ldots, h_q$ are merged into a vector of $n = p + q$ unknown variables $\mathbf{x} = [x_1 := Q_1, \ldots, x_n := h_q]$. In order to solve this system, n independent equations $f_i, i = 1, \ldots, n$ of n variables $x_i, i = 1, \ldots, n$ must be established

$$F(\mathbf{x}) = \begin{pmatrix} f_1(x_1, \ldots, x_n) \\ \cdots \\ f_n(x_1, \ldots, x_n) \end{pmatrix}. \tag{12}$$

All equations have to be brought into a canonical form $f_i(\mathbf{x}) = 0, i = 1, \ldots, n$, hence the desired solution is the root of the system

$$F(\mathbf{x}) = \mathbf{0}. \tag{13}$$

The unconstrained optimization method of Newton-Raphson is an appropriate tool to solve the nonlinear system (13). The first order derivative of the multidimensional function of multidimensional arguments (12) is the Jacobian matrix

$$J(\mathbf{x}) = \frac{dF(\mathbf{x})}{d\mathbf{x}} = \left[\begin{array}{ccc} \frac{\partial F(\mathbf{x})}{\partial x_1} & \cdots & \frac{\partial F(\mathbf{x})}{\partial x_n} \end{array} \right] = \begin{pmatrix} \partial f_1(\mathbf{x})/\partial x_1 & \ldots & \partial f_1(\mathbf{x})/\partial x_n \\ \vdots & \ddots & \vdots \\ \partial f_n(\mathbf{x})/\partial x_1 & \ldots & \partial f_n(\mathbf{x})/\partial x_n \end{pmatrix}. \tag{14}$$

The ith line is the transposed gradient $[\nabla f_i(\mathbf{x})]^{\mathsf{T}}$ of the ith function f_i. Using a Taylor series expansion near point $\mathbf{a}$, a linear approximation of (12) is

$$F(\mathbf{x}) \approx P_1(\mathbf{x}) = F(\mathbf{a}) + J(\mathbf{a})(\mathbf{x} - \mathbf{a}),$$

so the approximated root $P_1(\mathbf{x}) = \mathbf{0}$ of (13) is equivalent to

$$J(\mathbf{a})(\mathbf{x} - \mathbf{a}) = -F(\mathbf{a}).$$

The direct inversion of the Jacobian in case of a matrix of rank n, or by the pseudoinverse, in the case of a rectangular Jacobian (more equations than unknowns), delivers the desired solution $\mathbf{x} = \mathbf{a} - J(\mathbf{a})^{-1} F(\mathbf{a})$. When the Jacobian is calculated numerically at each new time instance, the direct inversion should be avoided due to the high computational cost A more efficient method is the solution of the linear system of form $A \cdot \mathbf{x} = \mathbf{b}$, in this case the purpose is to find the unknown $\mathbf{x}^{(k+1)}$ during the kth iteration

$$J(\mathbf{x}^{(k)}) \cdot \mathbf{x}^{(k+1)} = J(\mathbf{x}^{(k)}) \cdot \mathbf{x}^{(k)} - F(\mathbf{x}^{(k)}),$$

given the current estimate $\mathbf{x}^{(k)}$, with $A := J(\mathbf{x}^{(k)})$, $\mathbf{b} := J(\mathbf{x}^{(k)}) \cdot \mathbf{x}^{(k)} - F(\mathbf{x}^{(k)})$. At each discrete time step t the variable vector is initialized as $\mathbf{x}^{(0)}$ with the current simulation state of the flows $Q_1(t), Q_2(t), \ldots, Q_p(t)$ and heads $h_1(t), h_2(t), \ldots, h_q(t)$. The Newton-Raphson root finding usually shows a fast convergence after a few iterations, not imposing any computational burden on the real-time simulation[6].

4. Chemical Reaction Kinetics of the Continuous Stirred Tank Reactor

The chemical reaction is taking place in a well known device known as the Continuous Stirred Tank Reactor (CSTR). In the particular case of the proposed simulator, a chemical reactant A is fed constantly into the reactor, producing two products, B and C in a parallel (aka multistep)[7], irreversible[8], first-order reaction. Two catalysts are present in the reaction. The two products B and C are competing over the reactant A. The fault of a catalyst deactivation is simulated by an increase of the necessary reaction energy (E_B and/or E_C). The main reaction $A \rightarrow B$ is exothermic, i.e. transforming chemical to thermal energy, thus making it necessary to cool down the liquid within the tank by a controlled cooling circuit that passes through the reactor jacket in order to keep the reactor temperature constant. In chemical engineering, this production infrastructure is usually called nonisothermal CSTR. A secondary endothermic reaction, i.e. thermal to chemical energy, $A \rightarrow C$ takes place with a small proportion with respect to the total reaction. The basic theoretical groundwork used in this simulator can be found for instance in Schmidt (2005); Fogler (2011). However, exceptional fault situations in a CSTR increase the complexity of the mathematical model of the CSTR. For instance in steady-state normal operation and even in transient state startup operation, a constant reactor volume is assumed as the standard case. This constraint is invalidated, e.g. by a leak or malfunctioning level controller of the reactor fluid which would cause an increasing or decreasing reactor level, and hence faulty volume difference relative to the healthy state. The necessary knowledge to understand the chemical reaction is explained in the following.

4.1. Arrhenius Rate Equation and First-Order Reaction

The concentration of a chemical substance, aka species S, in this case reactant A and products B, C, is usually written using brackets as $[S]$. In the considered literature the symbol c_j prevails to denominate the concentration of the jth species. Concentrations of all substances are assumed to be homogeneously distributed within the volume of the reactor due to the relatively rapid mixing of the substances by stirring. The concentration c of a reactant is governed by an exponential decay law, the Arrhenius Rate Equation

$$k = k_0 e^{\frac{-E}{RT}}, \tag{15}$$

where the rate constant k depends on the pre-exponential factor k_0, the activation energy $E\left[\frac{\mathrm{J}}{\mathrm{mol}}\right]$ and the temperature $T\,[K]$ of the reaction[9]. The gas constant is $R = 8.31446\ \mathrm{J/(mol \cdot K)}$.

The reaction rate r that depends only on a single species with concentration $c_j\left[\frac{\mathrm{moles}}{\mathrm{m}^3}\right]$ is defined as

$$r_j = k_j c_j^n, n \in \mathbb{Q}, \tag{16}$$

where n is the order of the reaction and k is the rate constant of (15). With $n = 1$, a first-order reaction is defined as

$$r_j = k_j c_j = c_j k_{0_j} e^{\frac{-E}{RT}}. \tag{17}$$

For first-order reactions, the pre-exponential factor k_0 is also called the frequency factor, since its unit is $\left[\frac{1}{\mathrm{s}}\right]$ and r has units $\left[\frac{\mathrm{moles}}{\mathrm{s \cdot m}^3}\right]$ which can be interpreted as how many molecules are consumed for a reactant, or generated for a product, relative to a unit volume and unit time. Reaction rates of reactants have negative signs, that of products positive signs.

For the parallel, competing, first-order reactions $A \rightarrow B$ and $A \rightarrow C$, the rate of disappearance (i.e. negative generation) of species A is the sum of the rates of generation of the two products B and C, (Fogler, 2011), so

$$-r_A = r_B + r_C = k_B c_A + k_C c_A = (k_B + k_C) c_A. \tag{18}$$

The current reaction rates r_B and r_C are calculated in the simulation from (17) with the current values of the involved variables, hence

$$r_B = c_A k_{0_B} e^{\frac{-E_B}{RT}}, \qquad r_C = c_A k_{0_C} e^{\frac{-E_C}{RT}}, \tag{19}$$

where two pre-exponential factors k_{0_B}, k_{0_C} are constants, and T is the reactor temperature. The two activation energies E_B, E_C involved in the generation of species B and C are considered as variables, associated to a fault in the catalysts, i.e. if a catalyst concentration[10] is reduced, the free energy to cause the reaction by the same rate must be increased. This fault can be simulated by modifying the energies directly.

4.2. Reactor Material Balance

A preliminary task is the analysis of the reactor volume. Firstly, all flows related to a potential change of the reactor volume V are analyzed. A reactant flow into the CSTR opposes the product flow out of the CSTR. A coolant flow into the jacket opposes the warmed coolant flow out of the jacket. Moreover, leaks are considered as possible faults. A leak of the

[6] Using modern symbolic math processing software like Mathematica or Maxima, it is even possible to find a closed form expression for the right-hand side of iteration function $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - J(\mathbf{x}^{(k)})^{-1} F(\mathbf{x}^{(k)})$, thus avoiding the need to solve the linear system. The expressions for the updated components of $\mathbf{x}^{(k+1)}$ however are quite complex. See the additional material.

[7] The same reactant A is involved in the production of two different, competing products B and C.

[8] There is no inverse reaction that produces A from B and/or C.

[9] The capital 'A' is usually used to denominate the pre-exponential Arrhenius factor k_0, not to be confused with the name A of the reactant.

[10] Strictly speaking, the two parallel, competing reactions with two different, non disappearing catalysts Z_B, Z_C, using stoichiometric equations would be $\nu_A A + \nu_Z Z_B \rightarrow \nu_B B + \nu_Z Z_B$ and $\nu_A A + \nu_Z Z_C \rightarrow \nu_C C + \nu_Z Z_C$, with the stoichiometric coefficients $\nu_A = -1, \nu_B = \nu_C = 1, \nu_Z = 0$.

products, after they have left the CSTR is possible. A leak of the coolant from the jacket to the exterior and an internal leak of the coolant from the jacket into the CSTR are additional fault scenarios. The volume variation balance for the reactor contents, considering (3) are the inflows of the reactant feed and the leak from the cooling jacket, subtracting the flow out of the reactor, hence

$$\dot{V} = Q_1 + Q_6 - Q_2. \tag{20}$$

(Eq. 20) is a first order ordinary differential equation (ODE), for its solution and the corresponding update rule for the reactor volume V cf. Appendix A.1.

The fundamental relationship for the moles[11] (and later the heat in the energy balance) of the jth species in a closed system, in this case the CSTR, is described by the balance equation

$$[\text{Accumulation}] = [\text{Flow in}] - [\text{Flow out}] + [\text{Generation}]$$

$$\dot{N}_j = F_{j0} - F_j + G_j, \tag{21}$$

where each term has units $\left[\frac{\text{moles}}{\text{s}}\right]$. As mentioned before, "Generation" can also mean consumption of a species as a reactant. During the reaction with rate r_j, the generation $G_j \left[\frac{\text{moles}}{\text{s}}\right]$ of the jth product, or the consumption of the jth reactant, assuming that the species are homogeneously distributed within the reactor of volume $V\left[\text{m}^3 = \text{L}\right]$, is

$$G_j = r_j V. \tag{22}$$

The dependence of the reaction rate on the reactor volume is neglected (Hill, 1977), even if a variable volume is a realistic scenario in for a volume related fault. The molar flow into the tank F_{j0} and the molar flow out of the tank F_j are related to the volumetric flow $Q\left[\frac{\text{m}^3}{\text{s}}\right]$ as

$$F_{j0} = c_{j0} Q_1, \qquad F_j = c_j Q_2. \tag{23}$$

The moles are related to the concentration and volume of the tank as

$$N_j = V c_j. \tag{24}$$

Usually N_j is independent of the volume, since this is assumed to be constant in a CSTR. In a variable volume (Lund, 1970), presumably faulty CSTR, the moles of the jth species in (24) depend both on time varying volume $V(t)$ and time varying concentration $c_j(t)$, hence the total derivative of the accumulation term in (21) must be calculated. Consequently, omitting the index for the species j, the balance (21), with variable volume, becomes

$$[\dot{V(t)c(t)}] = c\dot{V} + V\dot{c} = F_0 - F + G. \tag{25}$$

Plugging (20) into (25), and considering (17), (22), (23), yields

$$\dot{c} = kc + \frac{1}{V}\left[(c_0 - c)Q_1 - cQ_6\right]. \tag{26}$$

Considering (18), for the concentrations of the reactant and products, and that the initial concentrations of the products are zero, i.e. $c_{B0} = 0, c_{C0} = 0$, (26) instantiates to

$$\begin{aligned}
\dot{c_A} &= -r_B - r_C + \frac{1}{V}\left[(c_{A0} - c_A)Q_1 - c_A Q_6\right] \\
\dot{c_B} &= r_B - \frac{c_B}{V}(Q_1 + Q_6) \\
\dot{c_C} &= r_C - \frac{c_C}{V}(Q_1 + Q_6)
\end{aligned} \tag{27}$$

For the solution of the ODE (27) cf. Appendix A.2.

4.3. Reactor Energy Balance

The nonisothermal CSTR is an open system where the total energy $E^{\text{tot}}[J]$ can change over time due to the following reasons: 1) An inflow of mass with a certain heat capacity, density and temperature, 2) One or more[12] outflows with the same characteristics, 3) Energy generation in the form of heat by a globally exothermic chemical reaction, 4) Heat loss by a cooling system and 5) An external heat sink or source to or from the environment, for instance by a bad insulation. In this model, based on the first law of thermodynamics, kinetic energy, potential energy and shaft work within the system are ignored, such that the total energy equals the internal energy $E^{\text{tot}} = U$ of the system. Since the system is dynamic, the interest lies in the *change* of energy over time $\dot{E}^{\text{tot}} = \dot{U}\left[\frac{\text{J}}{\text{s}}\right]$ Moreover the reaction is occurring at constant pressure, in a liquid-phase, such that the energy change balance is also an enthalpy change $\dot{H}$ balance. Enthalpy $H = U + pV$ is defined as the sum of the internal energy U and the product of the pressure and volume of the system. Under constant pressure, and neglecting the pressure effect on enthalpy, which is normally small for a liquid (Rawlings and Ekerdt, 2002), a further simplification can be made. The differential of the enthalpy becomes identical to the differential of the internal energy and all energy change terms are equivalent to heat change $\dot{q}\left[\frac{\text{J}}{\text{s}}\right]$ terms, hence

$$\dot{H} = \dot{U} = \dot{q}.$$

The energy change (Rawlings and Ekerdt, 2002; Hill, 1977), for a CSTR under constant pressure and ignoring pressure effect on the enthalpy can now be established as

$$\begin{bmatrix}\text{Heat accumulation}\\ \text{within CSTR}\end{bmatrix} = \begin{bmatrix}\text{Heat flow}\\ \text{into CSTR}\end{bmatrix} - \begin{bmatrix}\text{Heat flow(s)}\\ \text{out of CSTR}\end{bmatrix} + \\ + \begin{bmatrix}\text{Heat generation within}\\ \text{CSTR by reaction}\end{bmatrix} - \begin{bmatrix}\text{Heat removal from}\\ \text{CSTR by cooling}\end{bmatrix}$$

$$\dot{H} = \dot{q}_{\text{acc}} = \dot{q}_{\text{in}} - \dot{q}_{\text{out}} + \dot{q}_{\text{rxn}} - \dot{q}_{\text{c}}. \tag{28}$$

This balance is basically used to update the two variable temperatures of the system, namely those of the reactor and cooling jacket.

[11] Amount of basic chemical entities with the Avogadro constant $6.022 \cdot 10^{23}$ as a divisor. Different species with the same moles generally have distinct masses.

[12] A leaking must be considered as an additional outflow in a process. There are even faulty flows between the cooling jacket and the rector, in case of a hole in the reactor wall.

A liquid with density $\rho\left[\frac{\text{kg}}{\text{m}^3}\right]$, with specific heat capacity at constant pressure $C_P\left[\frac{\text{J}}{\text{kg}\cdot\text{K}}\right]$, with volume $V\left[\text{m}^3\right]$ and at temperature $T[K]$ stores the amount of heat q [J], defined as

$$q = \rho c_P V T. \tag{29}$$

4.3.1. Total Enthalpy Change

The specific heat capacity c_P is dependent on temperature and the composition of the species in the reaction, but within this model it is assumed to be constant (Fogler, 2011). The time derivative of (29) can be used to explain the left hand side $\dot{q}_{\text{acc}}$ of (28) and the inflow $\dot{q}_{\text{in}}$ and outflow $\dot{q}_{\text{out}}$ terms on the right hand side. Considering that the heat within the tank changes with temperature and volume (especially in a fault scenario), one has

$$\dot{q}_{\text{acc}} = \rho c_P[\dot{VT}] = \rho c_P\left[V\dot{T} + T\dot{V}\right]. \tag{30}$$

4.3.2. Enthalpy Flow Balance

Considering (3), the flow terms derived from (29) become

$$\begin{aligned}\dot{q}_{\text{in}} &= \rho c_P Q_1 T_1\\ \dot{q}_{\text{out}} &= \rho c_P Q_2 T_2.\end{aligned} \tag{31}$$

4.3.3. Heat of Reaction

The change of enthalpy $\Delta H_j\left[\frac{\text{J}}{\text{mol}}\right]$ for the jth species of a chemical reaction under constant pressure is called heat of the reaction. It is energy normalized for one mole of the species. In order to obtain the total heat energy per unit volume, is must be multiplied by the current concentration $c_j\left[\frac{\text{mol}}{\text{m}^3}\right]$ of that species. Consequently the total heat for the total volume becomes

$$q_{\text{rxn}_j} = \Delta H_j c_j V.$$

The time derivative of this heat is introduced by the reaction rate of (16) and (17), due to the Arrhenius Rate Equation (15), hence

$$\dot{q}_{\text{rxn}_j} = \Delta H_j c_j V k = \Delta H_j r_j V.$$

Since there are two reactions in parallel, one $A \rightarrow B$ exothermic with $\Delta H_B < 0$ and one $A \rightarrow C$ endothermic with $\Delta H_C > 0$, the total heat change due to the two first order, parallel, competing reactions is

$$\dot{q}_{\text{rxn}} = \dot{q}_{\text{rxn}_B} + \dot{q}_{\text{rxn}_C} = (\Delta H_B r_B + \Delta H_C r_C)V. \tag{32}$$

Globally, the reaction is exothermic since the first reaction is dominant, i.e. $(\Delta H_B r_B + \Delta H_C r_C) < 0$.

4.3.4. Heat Exchanger

Under normal steady state operation, the heat produced by the exothermic reaction must be absorbed by the coolant fluid. This *heat flux* between the CSTR and the cooling system is a one-dimensional model defined as

$$\dot{q}_{\text{c}} = UA_H(T_2 - T_4) = \rho_c C_{P_c} Q_5(T_4 - T_3), \tag{33}$$

where $U\left[\frac{\text{J}}{\text{m}^2\,\text{K}\cdot\,\text{s}}\right]$ is an overall heat transfer coefficient[13], A_H is the heat transfer area between the cooling jacket and the reactor, $T_4, T_3, \rho_c, C_{P_c}$ and Q_5 are the jacket and coolant feed temperature, density, specific heat capacity at constant pressure and the flow rate of the coolant, respectively. The right hand side term can be recognized as the difference of two heat flow terms in and out of the jacket ($\rho_c C_{P_c} Q_5 T_4 - \rho_c C_{P_c} Q_5 T_3$) in the cooling circuit, similar to those in and out of the CSTR in (31). This balance can be used to update the jacket coolant temperature in a simulator, solving for T_4 even in a dynamic scenario, since the temperature exchange is slow. So, using all current values of the simulation on the right hand side, the new value of the coolant temperature is obtained from (33) as

$$T_4^{\text{new}} = \frac{UA_H T_2 + \rho_c C_{P_c} Q_5 T_3}{\rho_c C_{P_c} Q_5 + UA_H}. \tag{34}$$

4.3.5. Reactor Temperature Update

A major task in the update process of the system variables is the calculus of the reactor temperature T_2^{new}. The volume variation and its approximation have been defined in (20). In order to update the CSTR temperature, the energy balance (28) must be slightly modified in case a faulty leak flow $\dot{q}_{\text{jrleak}}$ of coolant from the jacket into the reactor and a heat exchange with the exterior environment $\dot{q}_{\text{ext}}$ are included into the simulator model. Hence, extending (28), the CSTR energy balance is

$$\dot{q}_{\text{acc}} = \dot{q}_{\text{in}} - \dot{q}_{\text{out}} + \dot{q}_{\text{rxn}} - \dot{q}_{\text{c}} + \dot{q}_{\text{jrleak}} + \dot{q}_{\text{ext}}. \tag{35}$$

It is assumed that the density ρ and the specific heat capacity C_P of the coolant and that of the CSTR solution are identical.

The heat flow due to the leak from the cooling circuit through the jacket into the reactor, considering (31) is

$$\dot{q}_{\text{jrleak}} = \rho c_P Q_6 T_4. \tag{36}$$

The heat flow $\dot{q}_{\text{ext}}$ to or from the exterior from or to the CSTR is a explicitly defined single fault variable that changes conforming to a fault evolution function (immediate, gradually) defined later. Consequently, the heat flow balance, considering (30) and (35) is

$$\rho c_P\left[V\dot{T}_2 + T_2\dot{V}\right] = \dot{q}_{\text{in}} - \dot{q}_{\text{out}} + \dot{q}_{\text{rxn}} - \dot{q}_{\text{c}} + \dot{q}_{\text{jrleak}} + \dot{q}_{\text{ext}}. \tag{37}$$

Solving for the reactor temperature temporal gradient $\dot{T}_2$, considering (31), (32), (33), (36), yields

$$\dot{T}_2 = \frac{\dot{q}_{\text{rxn}} + \dot{q}_{\text{ext}} - \dot{q}_{\text{c}}}{\rho c_P} + Q_1(T_1 - T_2) + Q_6(T_4 - T_2). \tag{38}$$

Remarkably, the reactor temperature change does not contain the outflow term Q_2 directly. On the other hand, Q_2 has an influence on the reactor volume V via (20), and hence, via (32) on the reaction heat term $\dot{q}_{\text{rxn}}$. For the solution of the ODE (38) cf. Appendix A.3.

5. The MIT-CSTR

5.1. CSTR Control

For the uniformly distributed discrete time instants $t_0, t_1, \ldots, t_k, \ldots, t_n$, with the time interval $\Delta t = t_k - t_{k-1}$ between

[13] Not to be confused with the internal energy U.

two consecutive instants, a sample of a process variable $y(t_k)$ to be controlled is acquired at t_k. The desired value of this variable (set point) is $r(t_k)$. The error at t_k is defined as the difference between the desired and measured values

$$e(t_k) = r(t_k) - y(t_k).$$

This discrepancy serves as the basis for a PID controller (Green and Perry, 2007), with three components that contribute to the error correction, proportional, integral and derivative. The controlled variable that is sent to the plant is $u(t_k)$, for instance a valve opening percentage, and is calculated as

$$u(t_k) = u(t_{k-1}) + K_p \left[\left(1 + \frac{\Delta t}{T_i} + \frac{T_d}{\Delta t}\right) e(t_k) + \left(-1 - \frac{2T_d}{\Delta t}\right) e(t_{k-1}) + \frac{T_d}{\Delta t} e(t_{k-2}) \right], \quad (39)$$

where K_p, T_i, T_d are constants of the three components (proportional gain, integral time, derivative time), $u(t_{k-1})$ is the last value of the controlled variable, and $e(t_{k-1})$ e $e(t_{k-2})$ are the last and penultimate errors. With $T_i = \infty$, i.e. the integral gain $K_i := K_p/T_i = 0$, the PID controller specializes to a PD controller, with $T_d = 0$, i.e. the derivative gain $K_d := K_p T_d = 0$, the PID controller specializes to a PI controller (used in the original simulator), and with both gains set to zero $K_i = K_d = 0$, to a P controller.

There are three PID controllers[14] present in the CSTR simulator, the first standalone, responsible for the reactor volume V, via the reactor level L, the second as a cascade controller with two control loops responsible for the reaction temperature T. The primary controller of the cascade calculates as the error the difference between the set point of the temperature and the measured value, i.e. $e_T = r_T - y_T = r_2 - T_2$, and as its output the set point of the secondary controller, i.e. $u_2 = r_3$. Then the error between this set point and the measured inflow rate of the cooling circuit $e_{\text{cin}} = r_{\text{cin}} - y_{\text{cin}} = r_3 - Q_5$ determines the output $u_3 = m_2$ used to control the cooling water valve position.

5.2. Fault Model

The introduction of a fault is realized by changing some of the simulation system parameters. Table 4 and table 5 compile the possible faults available for simulation and the respective parameter that is changed. Table 4 shows the process faults where a simulation parameter is changed. Note that these simulation parameters are internal variables that only partially are identical to the measured variables in table 1. Fault zero is the normal operating condition. Table 5 shows the sensor faults where the 14 measured variables of table 1 deliver erroneous readings, either at a fixed value or with a fixed bias.

5.3. Fault Evolution Trajectories

The original simulator of Finch (1989) defined the trajectory of an evolving fault as a function for the value $v(t)$ of the affected simulation parameter at time instance t as

$$v(t) = v_{\text{lim}} - (v_{\text{lim}} - v_{\text{nominal}}) \exp\left[\tau\theta(t_{\text{delay}} - t)\right], \quad \theta(x) = \begin{cases} 0, & \text{if } x \geq 0 \\ x, & \text{otherwise} \end{cases} \quad (40)$$

where v_{nominal} is the value of the affected parameter in steady state for a certain normal process condition, v_{lim} is the maximum extent value that the affected parameter can reach asymptotically in case of a fault, below or above the nominal value, t_{delay} is the time after which the fault initiates and $\tau \in \{0.01, 0.1, 1, 10, 100\}$ is a time constant that controls how fast the maximum value v_{lim} of the affected parameter is reached. A high value of τ is almost equivalent to a sudden considerable change of a fault inducing parameter. This trajectory model allows to simulate plausibly almost all possible evolution cases. An elimination of an exiting fault can be achieved by interchanging the roles of v_{lim} and v_{nominal} in (40).

Table 4: Process Faults

#	Fault Type	Affected Parameter	Nominal Value [Fault Range]
1	No fault	-	-
2	Blockage at tank outlet	K_1	10.0 (10.0,300.0]
3	Blockage in jacket	K_9	1.0 (0.0,1.0]
4	Jacket leak to environment	m_8^L, K_8	0.0 (0.0,1.0]
5	Jacket leak to tank	m_7^L, K_7	0.0 (0.0,1.0]
6	Leak from pump	m_2^L, K_2	0.0 (0.0,1.0]
7	Loss of pump head	h_0	47.0 [0.0, 47.0)
8	Jacket exchange surface fouling	UA_H	1901.0 [1600.0,1901.0)
9	External heat sink/source ×1000	$\dot{q}_{\text{ext}}$	0 [-10,10]
10	Primary reaction activation energy ×1000	E_B	25 (25,30]
11	Secondary reaction activation energy ×1000	E_C	45 (45,54]
12	Abnormal feed flowrate	Q_1	0.25 [0.0,0.35]
13	Abnormal feed temperature	T_1	30.0 [10.0,50.0]
14	Abnormal feed concentration	c_{A0}	20.0 [0.0,30.0]
15	Abnormal cooling water temperature	T_3	20.0 [0.0,40.0]
16	Abnormal cooling water head	h_7	10.0 [0.0,15.0]
17	Abnormal jacket effluent head	h_{10}	≈1.4 [-100.0,100.0]
18	Abnormal reactor effluent head	h_1	≈26.0 [-200.0,200.0]
19	Abnormal level controller setpoint	r_1	2.0 [1.5,2.5]
20	Abnormal temperature controller setpoint	r_3	80.0 [70.0,90.0]
21	Control valve 1 stuck	m_1	0.1059 [0.0,1.0]
22	Control valve 2 stuck	m_2	0.9 [0.0,1.0]

5.4. Simultaneous Update of Heads and Flows

The CSTR circuit from the tank exit until the effluence of the products, including a leak is described by the instantiation of (13) of the system $F(\mathbf{h}^R, \mathbf{Q}^R) = \mathbf{0}$, defined by the variable vector $\mathbf{x}^R = \left[\mathbf{h}^R, \mathbf{Q}^R\right]$ composed of eight variables, five heads and three flows as $\mathbf{h}^R = [h_2, h_3, h_4, h_5, h_6]$, $\mathbf{Q}^R = [Q_2, Q_3, Q_4]$,

[14] PI controllers in Finch (1989)

Table 5: Process Sensor Faults

#	Fault Type	Affected Sensor	Nominal Value [Fault Range]
	Sensor faults with fixed bias		
23	Feed concentration	x_1	20.0 [-20.0,30.0]
24	Feed flowrate	x_2	0.25 [-0.25,0.35]
25	Feed temperature	x_3	30.0 [-30.0,50.0]
26	Reactor level	x_4	2.0 [-0.8,2.75]
27	Product A concentration	x_5	2.85 [-2.85,30.0]
28	Product B concentration	x_6	17.114 [-17.114,30.0]
29	Reactor temperature	x_7	80.0 [-80.0,130.0]
30	Coolant flowrate	x_8	0.9 [-0.9,2.0]
31	Product flowrate	x_9	0.25 [-0.25,0.35]
32	Coolant inlet temperature	x_{10}	20.0 [-20.0,40.0]
33	Coolant inlet head	x_{11}	10.0 [-10.0,140.0]
34	Level controller output	x_{12}	0.1016 [-0.1016,1.0]
35	Coolant flow controller output	x_{13}	0.61 [-0.61,1.0]
36	Coolant flow controller setpoint	x_{14}	0.9 [-0.9, 1.0]
	Sensor faults with fixed value		
37	Feed concentration	x_1	20.0 [0.0,30.0]
38	Feed flowrate	x_2	0.25 [0.0,35.0]
39	Feed temperature	x_3	30.0 [10.0,50.0]
40	Reactor level	x_4	2.0 [1.2,2.75]
41	Product A concentration	x_5	2.85 [0.0,30.0]
42	Product B concentration	x_6	17.114 [0.0,30.0]
43	Reactor temperature	x_7	80.0 [0.0,130.0]
44	Coolant flowrate	x_8	0.9 [0.0,2.0]
45	Product flowrate	x_9	0.25 [0.0,0.35]
46	Coolant inlet temperature	x_{10}	20.0 [0.0,40.0]
47	Coolant inlet head	x_{11}	10.0 [0.0,140.0]
48	Level controller output	x_{12}	0.1016 [0.0,1.0]
49	Coolant flow controller output	x_{13}	0.61 [0.0,1.0]
50	Coolant flow controller setpoint	x_{14}	0.9 [0.0,1.0]

and defined as

$$F^{\mathrm{R}}(\mathbf{h}^{\mathrm{R}}, \mathbf{Q}^{\mathrm{R}}) = \begin{cases} Q_2 - Q_3 - Q_4 \\ -h_1 - h_0 + K_{11} Q_2 + K_{12} Q_2^2 + h_2 \\ -h_2 + h_3 + K_1 Q_2^2 \\ -h_3 + h_4 + K_3 Q_2^2 \\ -h_3 + h_6 + K_2 Q_3^2 \\ -h_4 + h_5 + K_4 Q_4^2 \\ -h_6 + K_2 Q_3^2 \\ -h_5 + K_4 Q_4^2 \end{cases} \tag{41}$$

Note that the head of the CSTR h_1 as an input variable that is not mutually influenced by the other heads or flows, does not define a proper equation in the system (41), since it would create a trivial, one variable equation $h_1 + L = 0$ from (2), where L is the current reactor level. The Jacobian (14) of the system (41) is quite sparse, since in each of the eight equations, only a few of all eight variables appear. It is calculated from (41) as

$$J^{\mathrm{R}} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 1 & -1 & -1 \\ 1 & 0 & 0 & 0 & 0 & K_{11} + 2K_{12} Q_2 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 & 2K_1 Q_2 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & 2K_3 Q_2 & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 & 0 & 2K_2 Q_3 & 0 \\ 0 & 0 & -1 & 1 & 0 & 0 & 0 & 2K_4 Q_4 \\ 0 & 0 & 0 & 0 & -1 & 0 & 2K_2 Q_3 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & 2K_4 Q_4 \end{pmatrix}. \tag{42}$$

Similarly the cooling circuit with six heads (h_7 is again omitted since it is not influenced by other variables) and four flows $\mathbf{h}^{\mathrm{C}} = [h_8, h_9, h_{10}, h_{11}, h_{12}, h_{13}]$, $\mathbf{Q}^{\mathrm{C}} = [Q_5, Q_6, Q_7, Q_8]$, is described by the system

$$F^{\mathrm{C}}(\mathbf{h}^{\mathrm{C}}, \mathbf{Q}^{\mathrm{C}}) = \begin{cases} -Q_5 + Q_8 + Q_6 + Q_7 \\ -h_7 + h_8 + K_5 Q_5^2 \\ -h_8 + h_9 + K_6 Q_5^2 \\ -h_9 + h_{13} + K_7 Q_6^2 \\ -h_9 + h_{12} + K_8 Q_7^2 \\ -h_9 + h_{10} + K_9 Q_8^2 \\ -h_{10} + h_{11} + K_{10} Q_8^2 \\ -h_{11} + K_{10} Q_8^2 \\ -h_{13} + K_7 Q_6^2 \\ -h_{12} + K_8 Q_7^2 \end{cases} \tag{43}$$

with the Jacobian (14)

$$J^{\mathrm{C}} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & -1 & -1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 2K_5 Q_5 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 & 0 & 2K_6 Q_5 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 1 & 0 & 2K_7 Q_6 & 0 & 0 \\ 0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 2K_8 Q_7 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 2K_9 Q_8 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 2K_{10} Q_8 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 2K_{10} Q_8 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 2K_7 Q_6 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 2K_8 Q_7 & 0 \end{pmatrix}. \tag{44}$$

5.5. Original Simulator Improvements

In the work of Finch (1989), the calculus of the flow rates in the production and cooling circuit is a heuristic inspired by Kirchhoff's rules of equivalent resistances in serial and parallel electrical circuits.

Moreover, additional features in the form of constraints are available that can aid the fault diagnosis. These four parameters are incorporated into the 14 sensed variables vector of table 1 to form the final 18-dimensional feature vector $\mathbf{x}$. The inventory constraint ensures that the volume stays close to its initial value and that the volume difference caused by unequal inflow and outflow is kept small. Integrals are approximated by summing up finite values $[.]\Delta t$. The mol constraint measures the difference between the initial and current values (24) and the accumulated difference between the incoming and outgoing moles. The head loss constraints just sum up the head gains and losses $\Delta h = KQ^2$ of table 3 in the two systems (41) and (43) along the reactor and cooling circuits, assuming that there are no leaks.

Table 6: Security constraints of the CSTR simulator. Superscript 't_0' mean initial value at $t = 0$, superscript 'N' means nominal value in normal operation, c.f. table 1.

Constraint name	Definition
Inventory	$z_1 = V - V^{t_0} - \int_0^t [Q_1 - Q_4]\, d\tau$
Mol balance	$z_2 = (c_A + c_B + c_C^N)V - (c_A^{t_0} + c_B^{t_0} + c_C^{t_0})V^{t_0} - \int_0^t \left[c_{A0} Q_1 - (c_A + c_B + c_C^N) Q_4\right] d\tau$
Cooling water head loss	$z_3 = (h_7 - h_{11}) - (K_6 + K_5^N + K_9^N + K_{10}^N) Q_5^2$
Effluent head loss	$z_4 = (h_1 + h_0 - h_5) - (K_{11} + K_{12} + K_3 + K_1^N + K_4^N) Q_4^2$

5.6. Simulation algorithm

The main simulation loop, after the initialization of the system, can be resumed as the following calculations

1. Output of the controllers (level, reactor temperature coolant flow), (39)
2. Valve positions and valve resistances, (9)
3. Heads and flow rates (41), (43)
4. Heat flux from reactor to jacket and jacket temperature, (33), (34)
5. Reactor volume, (20)
6. Reaction rate, using Arrhenius equation, (19)
7. Concentration of reactant A and products B and C, (27)
8. Reactor temperature, (38)

6. Case study

The aim of this work is to promote an experimental platform for fault diagnosis experiments, and not to present a particular fault diagnosis technique. Therefore commonly used methods are used to illustrate the possibilities of the CSTR simulator to perform experimental studies for fault diagnosis. Fig. 2 shows the 11 mainly affected, out of the 18 available parameters for fault diagnosis in a normalized graph where each signal is scaled to a unit nominal value and distributed along the y-axis. The scenario is a succession of two faults. The simulation time is 100 min with $\Delta t = 0.2s$. Each 1 min the 18-dimensional feature vector is dumped, forming the $n \times m$ data matrix X with $n = 100$, $m = 18$. The random number seed is set to 554376 in order to permit an exact reproduction of this scenario. The first fault (#3 in table 4, blockage in jacket) occurs after 20 min, reaching its limit value 0.05 in (40) with $\tau = 1.0$. The second fault (#5 in table 4, jacket leak to tank) occurs after 40 min, reaching its limit value 0.05 in (40) with $\tau = 0.01$. After 63.7 min, an emergency shutdown is issued since the reactor temperature T_2 surpasses the limit of 130 °C. Some signals show an oscillatory behavior from this moment on and are practically useless for diagnosis.

After the first fault is triggered, the reactor temperature T_2 is gradually raising, augmenting even more after the second fault. Considering this fault combination, the reactor temperature is an example where different faults affect the same variable. Other variables seem to be affected only by one of the two faults, for instance the level controller output m_1 by fault #5 only.

Principal Component Analysis (Jolliffe, 2002) is an extremely important dimensionality reduction and visualization tool in fault diagnosis, e.g. (Chiang et al., 2001; Rato and Reis, 2013; Chiang et al., 2015; Gao and Hou, 2016; Maestri et al., 2010). Prior to the next steps, all features were individually scaled into the interval $[-1, 1]$. From the data matrix X, its centralized version $\mathsf{X_c} = [\mathbf{x}_1 - \boldsymbol{\mu}_x, \ldots, \mathbf{x}_n - \boldsymbol{\mu}_x]^\mathsf{T}$ is obtained by subtracting from each sample (row) $\mathbf{x}_i$, $i = 1, \ldots, n$ the mean of all samples $\boldsymbol{\mu}_x$. The eigenvalue analysis of the sample covariance matrix $\mathsf{\Sigma} = \mathsf{X}^\mathsf{T}\mathsf{X}/(n-1)$ produces the eigenvector matrix V and the correspondent diagonal eigenvalue matrix $\mathsf{\Lambda}$. The scores $\mathbf{y}_i = \mathsf{V}^\mathsf{T}(\mathbf{x}_i - \boldsymbol{\mu}_x)$ are the linearly decorrelated patterns. If only the $a < m$ first components of $\mathbf{y}$ are retained, that correspond to the a highest eigenvalues in $\mathsf{\Lambda}$, the principal components can be determined. Visualization is possible for $a = 2$ or $a = 3$.

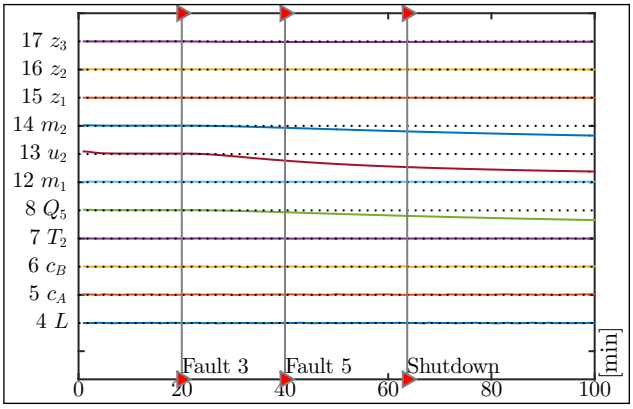

Figure 2: Signal evolution for a fault scenario with two successive faults. A sample $\mathbf{x}$ is composed of the 14 measured variables acquired each minute and the four constraints, c.f. table 1. All 18 variables have been adjusted for a unity nominal value.

Fig. 3 shows the first two principal components for the $n = 100$ samples. Since there is a temporal relationship among the samples, the trajectory can be traced (arrows are shown where the Euclidean distance between two successive scores in the Principal Component Space of the first two principal components is more than 0.45). The transition from the normal class (blue dots) to the first fault (red dots) is faster than the transition to the two simultaneous fault class (green dots), due to the higher time parameter τ.

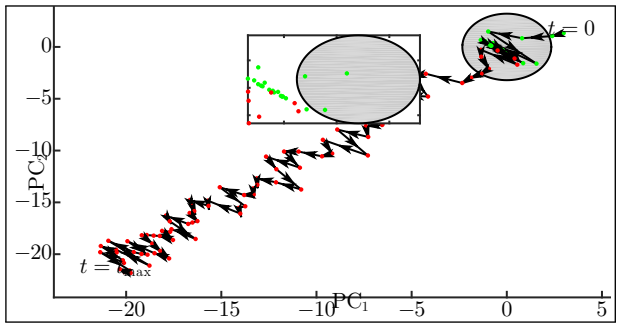

Figure 3: Temporal signal evolution of the first two principal components of all signals of fig. 2 for a fault scenario with two successive faults. Arrows are shown between patterns at two successive time instances where the Euclidean distance surpasses a threshold. The grey ellipse around $t = 0$ (shown also with a zoom in the middle of the figure) represents a Hotelling's T^2 statistic based fault hypothesis rejection region relative to the normal class.

An important line of research is fault detection where only the deviation of the process conditions from the normal operating state is considered (Chiang et al., 2001). For this purpose the eigenvector matrix V and eigenvalue matrix $\mathsf{\Lambda}$ are calculated only for those patterns that belong to the normal class. For the scenario of fig. 2 these are the first 20 data samples and the blue dots in fig. 3. The number of principal components has been fixed to $a = 2$ for this illustrative example. The occurrence of a fault can now be detected by calculating the Hotelling's T^2 statistic $T^2 = (\mathbf{x}-\boldsymbol{\mu}_x)^\mathsf{T}\mathsf{P}\mathsf{\Lambda}_a^{-1}\mathsf{P}^\mathsf{T}(\mathbf{x}-\boldsymbol{\mu}_x) = \mathbf{y}_a^\mathsf{T}\mathsf{\Lambda}_a^{-1}\mathbf{y}_a$ for an incoming process pattern $\mathbf{x}$ and executing a hypothesis test $T^2 > T_\alpha^2$ against a threshold. For an appropriate significance level α, the

threshold for the test can be defined as $T_\alpha^2 = \frac{a(n-1)}{n-a} F_{\alpha(a,n-a)}$, using a F-distribution $F_{(a,n-a)}$ with a and $n-a$ degrees of freedom. The grey ellipse around $t = 0$ in fig. 3 does exactly show the patterns considered normal for a significance of 95%. The majority within this fault rejection region are the normal patterns (blue dots). Only one red dot that is already labeled as a fault is present in the $T^2 \leq T_\alpha^2$ region, probably because the fault does not yet sufficiently manifest itself in the signal values. In Monroy et al. (2012) the evolution of the T^2 statistic is shown for a selected set of faults in the context of the Tennessee Eastman simulator. Another interesting visualization of the normal process conditions and faults was done for the TE simulator in Robertson et al. (2015), using a combination of Gaussian Mixture Models (GMM) and the Self Organizing Map (SOM) (Kohonen, 2001).

In table 7 three different standard classifiers are tested on fault scenarios with different degrees of difficulties with respect to the discernability of the classes. As a representative classifier from the area of artificial neural networks, the Multilayer Perceptron (MLP) is proposed, trained by the scaled conjugate gradient algorithm (Møller, 1993). This is the default Artificial Neural Network architecture and training algorithm of Matlab and was also used in the context of the Tennessee Eastman simulator in Monroy et al. (2012). The number of hidden neurons was set to twice the number of inputs, plus one. The 1-Nearest Neighbor classifier (Cover and Hart, 1967) with Euclidean distance metric represents the category of simple non-parametric methods. It has the advantage that no hyperparameters have to be determined first, for instance the type of kernel and associated kernel parameters in the support vector machine (SVM) classifier (Cortes and Vapnik, 1995). A SVM was used as the classifier architecture for the Tennessee Eastman process in Monroy et al. (2010, 2012). The `libsvm` library of Chen et al. (2005) with default parameters (C-SVM with $C = 1.0$, radial basis function kernel with spread $\gamma = 1.0$) was used here for the SVM implementation. K-fold cross validation (CV) is used to estimate the performance. The total data set of n patterns is divided into K subsets, each subset is used once for test and $K-1$ times for training. When the training time of the classifier is not excessive, the leave-one-out (LOO) cross validation with $K = n$ can be used as a special case of K-fold cross validation. LOO gives a more reliable estimate than simple train-test data division, since implicitly each available pattern is tested. In the experiments the LOO and K-fold CV are used, depending on the training costs of the classifier architecture. The performance criteria are the classification accuracy and the area under the ROC curve (Fawcett, 2006). The class separation difficulty is increased by diminishing the τ parameter in the fault trajectories (40), since the transition between two process conditions is slowed down. The consequence is that the classes are less discernible and hence the classification accuracy and AUC-ROC are diminishing.

Table 7: Performance estimation of the accuracy (ACC) and area under the ROC curve (AUC-ROC) for three classifiers using selected faults with different transition time parameters τ between the normal condition and the fault.

	ACC			AUC-ROC		
	Classifier Model					
Fault #, τ	1-NN	SVM	MLP	1-NN	SVM	MLP
fault_4_00001	0.8300	0.5833	0.9033	0.8269	0.8570	0.9869
fault_4_0001	0.8367	0.6733	0.9500	0.8269	0.9891	0.9959
fault_4_001	0.9567	0.8733	0.9700	0.9556	0.9952	0.9981
fault_4_01	0.9900	0.9500	0.9933	0.9902	0.9999	0.9997
fault_4_1	0.9967	0.9733	0.9967	0.9972	0.9997	0.9988
fault_4_10	0.9999	0.9833	0.9999	1.0000	1.0000	1.0000
fault_5_00001	0.8133	0.6233	0.9133	0.8000	0.9478	0.9944
fault_5_0001	0.8867	0.6767	0.9533	0.8877	0.8991	0.9874
fault_5_001	0.9733	0.9200	0.9767	0.9748	0.9921	0.9834
fault_5_01	0.9867	0.9800	0.9967	0.9876	0.9963	0.9918
fault_5_1	0.9933	0.9833	0.9967	0.9947	0.9999	0.9948
fault_5_10	0.9967	0.9900	1.0000	0.9973	1.0000	1.0000
fault_6_00001	0.8767	0.8433	0.9300	0.8679	0.9993	0.9942
fault_6_0001	0.9400	0.9200	0.9733	0.9416	0.9989	0.9967
fault_6_001	0.9767	0.9767	0.9833	0.9776	0.9998	0.9999
fault_6_01	0.9867	0.9900	0.9933	0.9875	0.9991	0.9952
fault_6_1	0.9933	0.9933	0.9999	0.9932	0.9991	0.9999
fault_6_10	0.9933	1.0000	0.9999	0.9931	1.0000	0.9999
fault_7_00001	0.8333	0.9600	0.9467	0.8270	0.9438	0.9935
fault_7_0001	0.8667	0.5867	0.9567	0.8619	0.9830	0.9935
fault_7_001	0.9400	0.6200	0.9800	0.9340	0.8921	0.9962
fault_7_01	0.9900	0.7400	0.9833	0.9917	0.9791	0.9990
fault_7_1	0.9933	0.8100	0.9900	0.9919	0.9835	0.9903
fault_7_10	0.9967	0.8300	0.9967	0.9963	0.9847	0.9980
fault_12_00001	0.8333	0.6233	0.9000	0.8267	0.9238	0.9819
fault_12_0001	0.8700	0.9567	0.9533	0.8670	0.9161	0.9890
fault_12_001	0.9667	0.9300	0.9867	0.9666	0.9614	0.9941
fault_12_01	0.9900	0.9500	0.9833	0.9903	0.9912	0.9986
fault_12_1	0.9933	0.9500	0.9900	0.9931	0.9932	0.9997
fault_12_10	0.9933	0.9567	0.9967	0.9929	0.9983	1.0000
fault_15_00001	0.8867	0.7433	0.9200	0.8846	0.8357	0.9713
fault_15_0001	0.8933	0.9333	0.9267	0.8879	0.9897	0.9880
fault_15_001	0.9733	0.9833	0.9800	0.9724	0.9978	0.9988
fault_15_01	0.9800	0.9900	0.9900	0.9803	0.9985	0.9992
fault_15_1	0.9900	0.9967	0.9967	0.9903	0.9978	0.9960
fault_15_10	0.9933	1.0000	1.0000	0.9944	1.0000	1.0000

7. Conclusion

The main goal of this work is to promote a simulation platform of a chemical reactor that can be used as a benchmark for fault diagnosis experiments. It is suitable as an experimental basis for a extensive variety of fault diagnosis methodologies, both model-based and model-free. The original system of Finch (1989); Oyeleye (1990) was improved by proposing a simple, however sufficiently sophisticated model of the hydraulic components and solution of the simultaneous nonlinear equations systems by Newton-Raphson. On the contrary to the popular Tennessee-Eastman chemical process simulator, the MIT-CSTR provides a completely documented theory which gives the researcher access to the complex causal interactions of the system variables, thus allowing founded reasoning about the analyzed faults.

Appendix A. Solution of First-Order Ordinary Differential Equations

A first order ordinary differential equation (ODE) with time t as the independent variable, and y as the dependent variable, can be defined in the form

$$\frac{dy}{dt} = \dot{y} = f(t, y). \tag{A.1}$$

When the independent variable is missing on the right hand side, the ODE simplifies to an *autonomous* ODE $\dot{y} = f(y)$. The most simple approximate solution is the Euler method with the current value y_n of the dependent variable as the given initial value. When the differentials of time dt and variable dy are limited to the linear contribution in the Taylor series expansion at t_n, with $\Delta t := t_{n+1} - t_n$ and $\Delta y := y_{n+1} - y_n$, then (A.1) is approximated by

$$\frac{\Delta y}{\Delta t} = f(t, y),$$

and hence the dependent variable can be updated from the current time instance t_n to the next t_{n+1} as

$$y_{n+1} = y_n + \Delta t f(t_n, y_n). \tag{A.2}$$

Depending on the sophistication of the simulation, the Euler method might be sufficiently precise. In Matlab for instance, the standard solver is the function `ode45` which implements a single-step Runge-Kutta, embedded method with adaptive step size, using a forth-order and a fifth-order method pair, the Dormand-Prince pair. For C, C++ code, the standard routine for the Numerical Recipes library is `odeint`. In the original simulator of Finch (1989), only the Euler method was used. The updates rules for the reactor volume, species concentration and heat balances are explicated in the following.

Appendix A.1. Reactor Volume

The update rule of the autonomous ODE of (Eq. 20) for the reactor volume V, using the Euler method (A.2) becomes

$$V^{\text{new}} = V^{\text{old}} + \Delta t \left[Q_1 + Q_6 - Q_2 \right].$$

Appendix A.2. Species Concentrations

The updated values for the concentrations of the three involved species in (27) can be calculated by Runge-Kutta formulas or simple Euler methods, using (A.2).

The original simulator of Finch (1989) approximating the differentials $d(Vc), dt$ in (25) by finite differences $\Delta(Vc) = V^{\text{new}}c^{\text{new}} - V^{\text{old}}c^{\text{old}}$, $\Delta t = t^{\text{new}} - t^{\text{old}}$, defined the update rule for the concentration from one discrete time instance t^{old} of the simulation to the next time instance t^{new} with the Euler rule as

$$c^{\text{new}} = \frac{V^{\text{old}}}{V^{\text{new}}} c^{\text{old}} + \frac{\Delta t}{V^{\text{new}}} \left[c_0 Q_1 - c Q_2 + r V^{\text{old}} \right].$$

Since the initial concentrations of the products are zero, i.e. $c_{B0} = 0, c_{C0} = 0$, considering (18), the update rules for the concentrations of the involved species from one step of the simulation to the next step become

$$\begin{aligned}
c_A^{\text{new}} &= \frac{V^{\text{old}}}{V^{\text{new}}} c_A^{\text{old}} + \frac{\Delta t}{V^{\text{new}}} \left[c_{A0} Q_1 - c_A Q_2 - (r_B + r_C) V^{\text{old}} \right] \\
c_B^{\text{new}} &= \frac{V^{\text{old}}}{V^{\text{new}}} c_B^{\text{old}} + \frac{\Delta t}{V^{\text{new}}} \left[-c_B Q_2 + r_B V^{\text{old}} \right] \\
c_C^{\text{new}} &= \frac{V^{\text{old}}}{V^{\text{new}}} c_C^{\text{old}} + \frac{\Delta t}{V^{\text{new}}} \left[-c_C Q_2 + r_C V^{\text{old}} \right].
\end{aligned}$$

Appendix A.3. Reactor Temperature

Again, (38) is solved by Runge-Kutta or Euler methods. The original simulator of Finch (1989) establishes a different formula to update the CSTR temperature T. Starting from (37), like in the case of the material balance (25), approximating the differentials dV, dt by finite differences $\Delta V = V^{\text{new}} - V^{\text{old}}$, $\Delta t = t^{\text{new}} - t^{\text{old}}$, with the Euler method, an update rule for the CSTR temperature from one discrete time instance t^{old} of the simulation to the next time instance t^{new}, considering (31), (32), (33), (36) can be found as

$$\begin{aligned}
T^{\text{new}} = \frac{V^{\text{old}}}{V^{\text{new}}} T^{\text{old}} + \frac{\Delta t}{V^{\text{new}}} \cdot \\
\left\{ \frac{1}{\rho c_P} \left[\dot{q}_{\text{rxn}} + \dot{q}_{\text{ext}} - \dot{q}_{\text{c}} \right] + Q_1 T_1 + Q_6 T_4 - Q_2 T_2 \right\}.
\end{aligned} \tag{A.3}$$

The numerical equivalence of the original temperature update (A.3) and the update based on the Euler solution of (38) was verified by extending the original code, resulting in a relative difference of approximately 0.001%.

Appendix B. CSTR Implementation

One of the reasons why e.g. the Tennessee Eastman simulator is popular and the MIT-CSTR practically unknown is that the first published the software online and the second is a pre-internet work where the original Fortran code is only available as a bitmap on the scanned Ph.D thesis of Finch (1989). The code was transcribed from there and missing parts were recovered (for instance the Fortran random number routines `GGUBS, GGNML, MDNRIS, MERFI` were originally not included). The missing variable plotting routine `SIMPLOT` was substituted by a `gnuplot` interface that allows to visualize the temporal evolution of the measured variables and constraints of table 1. The original software[15] consists of a single file that can be easily compiled by a standard Fortran compiler, e.g. `gfortran` in Unix systems. A simple configuration file that emulates the manual input of the system parameters is sufficient to run the simulator in a single command. A revised implementation with the new methods proposed in this paper was written in C.

In order to provide a common database for fault experiments, for all faults a training and test set is provided, similar to the organization of the Tennessee Eastman datasets.

[15] Code and additional material at `sites.google.com/site/cstrsimulator`

Aldrich, C., Auret, L., 2013. Unsupervised process monitoring and fault diagnosis with machine learning methods. Springer.

Askarian, M., Escudero, G., Graells, M., Zarghami, R., Jalali-Farahani, F., Mostoufi, N., 2016. Fault diagnosis of chemical processes with incomplete observations: A comparative study. Computers & Chemical Engineering 84, 104 – 116.

Çengel, Y., Cimbala, J., 2013. Fluid Mechanics Fundamentals and Applications: Third Edition. McGraw-Hill Higher Education.

Chen, H., Tiňo, P., Yao, X., 2014. Cognitive fault diagnosis in Tennessee Eastman process using learning in the model space. Computers & Chemical Engineering 67, 33 – 42.

Chen, P. H., Lin, C. J., Schölkopf, B., 2005. A tutorial on ν-support vector machines. Applied Stochastic Models in Business and Industry 21, 111–136.

Chiang, L., Braatz, R., Russell, E., 2001. Fault Detection and Diagnosis in Industrial Systems. Advanced Textbooks in Control and Signal Processing. Springer London.
URL http://web.mit.edu/braatzgroup/links.html

Chiang, L. H., Jiang, B., Zhu, X., Huang, D., Braatz, R. D., 2015. Diagnosis of multiple and unknown faults using the causal map and multivariate statistics. Journal of Process Control 28, 27–39.

Controls, F., 2005. Control Valve Handbook, 4th Edition. Fisher Controls.
URL http://www.documentation.emersonprocess.com/groups/public/documents/book/cvh99.pdf

Cortes, C., Vapnik, V., 1995. Support-vector networks. Mach. learn. 20 (3), 273–297.

Cover, T., Hart, P., Jan. 1967. Nearest neighbor pattern classification. IEEE Inform. Theory 13 (1), 21–27.

Dickenson, T., 1999. Valves, Piping, and Pipelines Handbook. Elsevier advanced technology. Elsevier Advanced Technology.

Downs, J. J., Vogel, E. F., 1993. A plant-wide industrial process control problem. Comput. Chem. Eng. 17 (3), 245–255.

Fawcett, T., 2006. An introduction to ROC analysis. Pattern recogn. lett. 27 (8), 861–874.

Finch, F. E., 1989. Automated fault diagnosis of chemical process plants using model-based reasoning. Ph.D. thesis, Massachusetts Institute of Technology, Dept. of Chemical Engineering.
URL http://hdl.handle.net/1721.1/14194

Fogler, H., 2011. Essentials of Chemical Reaction Engineering. Prentice Hall international series in the physical and chemical engineering sciences. Prentice Hall.

Gao, X., Hou, J., 2016. An improved svm integrated gs-pca fault diagnosis approach of tennessee eastman process. Neurocomputing 174, 906–911.

Green, D., Perry, R., 2007. Perry's Chemical Engineers' Handbook, Eighth Edition. McGraw Hill professional. McGraw-Hill Education.

Hill, C., 1977. An introduction to chemical engineering kinetics & reactor design. Wiley series in chemical engineering. Wiley.

Idelchik, I., Steinberg, O., 1996. Handbook of Hydraulic Resistance. Begell House.

Jeppson, R. W., 1974. Steady flow analysis of pipe networks: an instructional manual. Tech. rep., Utah State University, Utah Water Research Laboratory.

Jolliffe, I. T., 2002. Principal Component Analysis, Second Edition. Springer.

Kohonen, T., 2001. Self-organizing maps, ser. Information Sciences. Berlin: Springer 30.

Lund, M. M., 1970. Variable-volume operation of a stirred tank reactor. Ph.D. thesis, Iowa State University.
URL http://lib.dr.iastate.edu/rtd/4246

Mackay, R., 2004. The Practical Pumping Handbook. Elsevier Science.

Maestri, M., Farall, A., Groisman, P., MCassanello, Horowitz, G., 2010. A robust clustering method for detection of abnormal situations in a process with multiple steady-state operations mode. Computers & Chemical Engineering 34, 223 231.

Møller, M. F., 1993. A scaled conjugate gradient algorithm for fast supervised learning. Neural networks 6 (4), 525–533.

Monroy, I., Benitez, R., Escudero, G., Graells, M., 2010. A semi-supervised approach to fault diagnosis for chemical processes. Computers & Chemical Engineering 34, 631 642.

Monroy, I., Benitez, R., Escudero, G., Graells, M., 2012. Enhanced plant fault diagnosis based on the characterization of transient stages. Computers & Chemical Engineering 37, 200 213.

Nayyar, M., 2000. Piping Handbook. McGraw-Hill Handbooks. McGraw-Hill Education.

Novak, P., Guinot, V., Jeffrey, A., Reeve, D., 2010. Hydraulic Modelling – An Introduction: Principles, Methods and Applications. CRC Press.

Oyeleye, O. O., 1990. Qualitative modeling of continuous chemical processes and applications to fault diagnosis. Ph.D. thesis, Massachusetts Institute of Technology, Dept. of Chemical Engineering.
URL http://hdl.handle.net/1721.1/14281

Oyeleye, O. O., Kramer, M. A., 1988. Qualitative simulation of chemical process systems: steady-state analysis. AIChE J. 34 (4), 1441–1454.

Rato, T. J., Reis, M. S., 2013. Fault detection in the Tennessee Eastman benchmark process using dynamic principal components analysis based on decorrelated residuals (dpca-dr). Chemometrics and Intelligent Laboratory Systems 125, 101 – 108.

Rawlings, J. B., Ekerdt, J. G., 2002. Chemical reactor analysis and design fundamentals. Nob Hill Pub, Llc.

Robertson, G., Thomas, M., Romagnoli, J., 2015. Topological preservation techniques for nonlinear process monitoring. Computers & Chemical Engineering 76, 1 16.

Schmidt, L., 2005. The Engineering of Chemical Reactions. Topics in Chemical Engineering - Oxford University Press. Oxford University Press.

Sorsa, T., Koivo, H. N., 1993. Application of artificial neural networks in process fault diagnosis. Automatica 29 (4), 843–849.

Sorsa, T., Koivo, H. N., Koivisto, H., 1991. Neural networks in process fault diagnosis. IEEE Trans. on Systems, Man, and Cybernetics 21 (4), 815–825.

Venkatasubramanian, V., Vaidyanathan, R., Yamamoto, Y., 1990. Process fault detection and diagnosis using neural networksi. steady-state processes. Computers & Chemical Engineering 14 (7), 699–712.

Yin, S., Ding, S. X., Haghani, A., Hao, H., Zhang, P., 2012. A comparison study of basic data-driven fault diagnosis and process monitoring methods on the benchmark Tennessee Eastman process. J. Process Contr. 22 (9), 1567 – 1581.

Yin, S., Luo, H., Ding, S., May 2014. Real-time implementation of fault-tolerant control systems with performance optimization. IEEE Trans. Ind. Electron. 61 (5), 2402–2411.

Yin, S., Zhu, X., Kaynak, O., 2015. Improved pls focused on key-performance-indicator-related fault diagnosis. Industrial Electronics, IEEE Transactions on 62 (3), 1651–1658.

Young, D. F., Munson, B. R., Okiishi, T. H., 2006. Fundamentals of Fluid Mechanics, 5th Edition. John Wiley & Sons, Inc.

Young, D. F., Munson, B. R., Okiishi, T. H., Huebsch, W. W., 2012. A Brief Introduction to Fluid Mechanics, 5th Edition. Wiley.

Zhang, J., Martin, E. B., Morris, A. J., 1996. Fault detection and diagnosis using multivariatesta tistical techniques. Trans. IChemE 74, 89–96.

Zhang, J., Morris, A. J., 1994. On-line process fault diagnosis using fuzzy neural networks. IEE Intelligent System Engineering, 37–47.