

# Revision of the Tennessee Eastman Process Model

Andreas Bathelt\* N. Lawrence Ricker\*\* Mohieddine Jelali\*\*\*

\* *Laboratory of Control Engineering and Mechatronics, Cologne University of Applied Science, Betzdorfer Straße 2, 50679 Cologne, Germany (e-mail: andreas.bathelt@fh-koeln.de)*

\*\* *Professor Emeritus, Department of Chemical Engineering, University of Washington, Seattle, WA 98195-1750, USA (e-mail: ricker@u.washington.edu)*

\*\*\* *Laboratory of Control Engineering and Mechatronics, Cologne University of Applied Science, Betzdorfer Straße 2, 50679 Cologne, Germany (e-mail: mohieddine.jelali@fh-koeln.de)*

**Abstract:** In this paper, a closer look at the underlying computer code of the well-known Tennessee Eastman Process model is taken. Since its introduction in the 1990s typical simulation software, e.g. MATLAB, which is used to simulate the process model, has changed. Thus the original program flow devised by Downs & Vogel no longer holds. This results in problems regarding the repeatability of simulation results. This problem and its cause are considered in the following and a solution in terms of a modified code is presented. Furthermore, some additional changes are discussed, widening the useability of the simulation model (e.g. lower simulation runtime, additional process measurements).

© 2015, IFAC (International Federation of Automatic Control) Hosting by Elsevier Ltd. All rights reserved.

**Keywords:** Benchmark example; Chemical industry; Process models; Process simulators; Process Control.

## 1. INTRODUCTION

Although being a rather old process model, the Tennessee Eastman Process model of Downs and Vogel [1993] remains an important tool throughout all disciplines of the system theory for the purpose of comparative studies or validation of algorithms (e.g. fault diagnosis in Yin et al. [2012], system identification in Bathelt and Jelali [2014]). Its strength is the fact that it was modeled based on a real process. This led to a non-linear model of a rather complex multicomponent system. Due to the model's frequent usage, it is advantageous that this model, or more precisely its code, runs flawlessly. However, the authors' attention was caught by irregularities regarding the consistency of the model's results during simulation runs under MATLAB. The model's implementation was the C-mex S-function of the Tennessee Eastman Archive of Ricker [2005]. It turned out that these inconsistencies in the simulation results were dependent of the choice of the solver. That is, the simulation results could differ significantly from one solver to another solver. In terms of reproducibility of results, this effect is undesirable since it is consequently necessary to specify the simulation settings along with the disclosure of the results. Hence, the program was analyzed in order to track down the cause of this inconvenience. This cause and the resulting alteration are presented in the subsequent sections.

The paper is structured as follows. In Section 2, the problem and its cause are explained. Along with some additional changes to the program, the remedy for this dependency on the solver is depicted in Section 3. The

fourth section is used to present results of two small comparative studies regarding the solver (in)dependence and the simulation duration (in comparison with the original model) of the revised model. In the concluding section, a summary is given.

The files of the revised model are available at [depts.washington.edu/control/LARRY/TE/download.html#Updated\\_TE.Code](http://depts.washington.edu/control/LARRY/TE/download.html#Updated_TE.Code). Following Ricker [2005], there are two models. The activation vector of the disturbances of one model is provided using the model parameters (temex\_mod), whereas the other model treats the disturbance vector as an additional input (temexd\_mod).

## 2. PROBLEM DESCRIPTION

As mentioned in the introductory section, results from the existing code of the Tennessee Eastman Process model vary with choice of the solver used for the simulations under MATLAB/Simulink. Furthermore, a change of the simulation's time increment has the same effect. Fig. 1 illustrates the significant influence the choice of the solver has on the simulation results. Except for the different solvers, the settings of the simulations were the same. The simulations ran open-loop using Mode 1 and disturbances of the type "random variation" affecting the reactor cooling water inlet temperature and the composition of the components of stream 4 (see Downs and Vogel [1993]). In order to keep the reactor pressure as long as possible below its shut-down constraint of 3000 kPa, the position of the reactor cooling water outlet valve was given a slope of  $-8\%/h$ . The applied solvers were the fixed-step solvers of

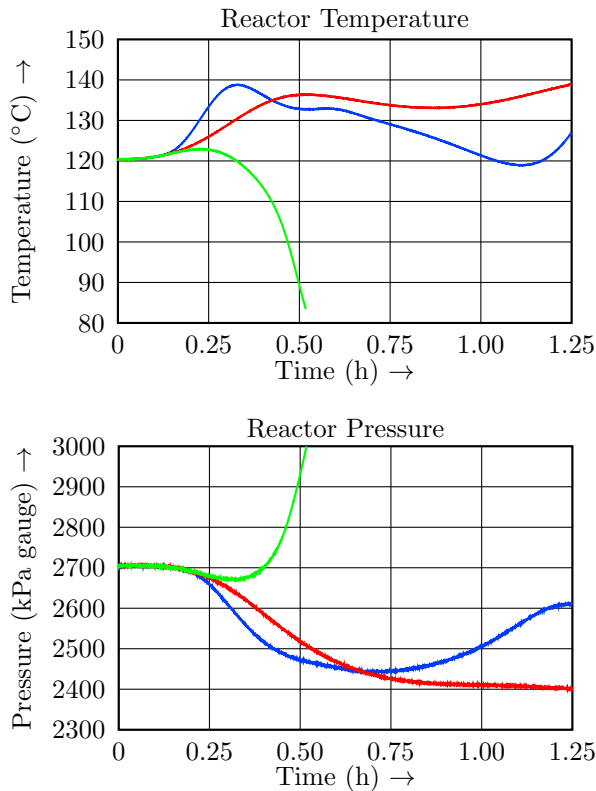


Fig. 1. Simulation results of the reactor pressure and reactor temperature using the original model (blue: Euler/ode1; red: Bogacki-Shampine/ode3; green: Dormand-Prince/ode45)

the Euler method (ode1; in blue) and Bogacki-Shampine method (ode3; in red) as well as the variable-step solver of the Dormand-Prince method (ode45; in green). The increment of the simulation time was set to 1 second (for Dormand-Prince maximum increment). It is possible to increase the time increment of the latter two solvers. But with regard to the Euler solver, an increment of one second marks approximately the upper bound, as the simulation potentially tends to become numerically unstable, if the increment is further increased.

Although simulations with activated disturbances of the type “random variation” (in the following referred to as *process disturbances*) were affected, a simulation without any disturbance showed no dependency on the solver. Thus, the cause of the problem was isolated within the simulation and generation of the process disturbances and even more precisely within the generation of the random numbers<sup>1</sup> used for this purpose. The random generator of the process model is a sub-function of the main-function *tefunc()*. The state of this random generator changes with every call to this sub-function, which happens in conjunction with the generation of the process disturbances and likewise with the generation of the measurement noise. The

<sup>1</sup> It is important to bear in mind that this is not an error or bug in the program of Downs & Vogel or the C-implementation of Ricker in the usual sense. This effect is rather due to the development of the simulation environment, *i.e.* MATLAB/Simulink, since the first publication of the model, which thereby causes a mismatch between the original intended usage of the model and the program flow of the simulation.

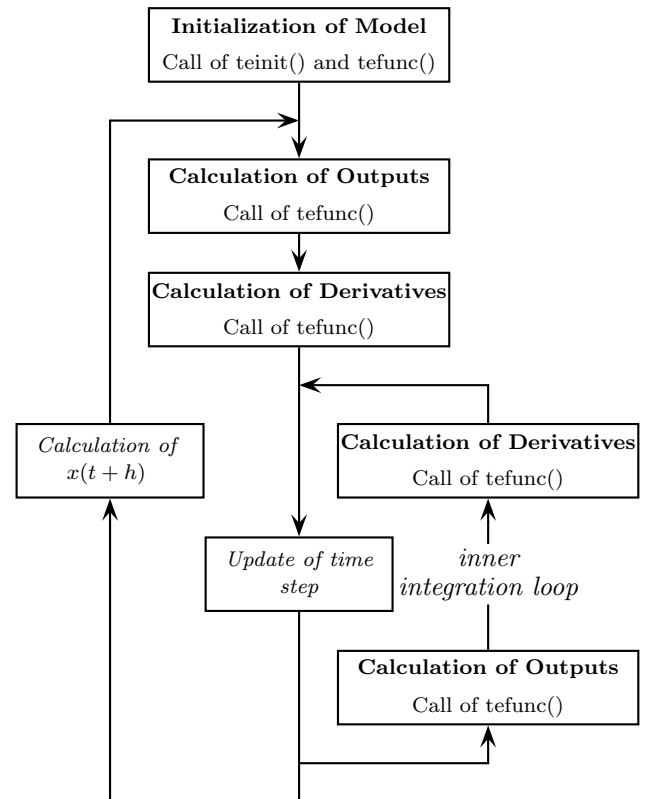


Fig. 2. Flow chart of the simulation loop in principle

model of the process disturbances itself is a polynomial, whose coefficients are updated periodically based on the generated random numbers. The generation of the measurement noise, however, takes place whenever the outputs of the process (*i.e.* outputs of the model) are calculated. This is done with every call of the main-function of the process, which not only happens once the outputs (of the simulation as a whole) are determined, but also once the derivatives are calculated and during the execution of the inner integration loop of solvers using minor time steps (see Fig. 2). The number of cycles of this inner integration loop depends on the solver. Thus the number of calls of the random generator between two (major) time steps is directly dependent upon the choice of the solver. That is, when the recalculation of the polynomials of the process disturbances takes place, the state of random generator varies with the chosen solver. This again causes the differences of the simulation results. The original structure of the model, devised by Downs & Vogel, was intended for a less structured simulation loop. The main-function was supposed to be called only once per cycle. The calculation of the derivatives and outputs were to happen together within one step; see Fig. 2 and referring description in Downs and Vogel [1993] p. 254.

In spite of this problem, existing results based on the original code remain correct, since the physical-chemical model is not affected. The outlined problem takes effect only during the generation of disturbances of the type “random variation” (see Table 8 in Downs and Vogel [1993] p. 250) and changes the progression of the disturbance unless the settings of the solver are the same.

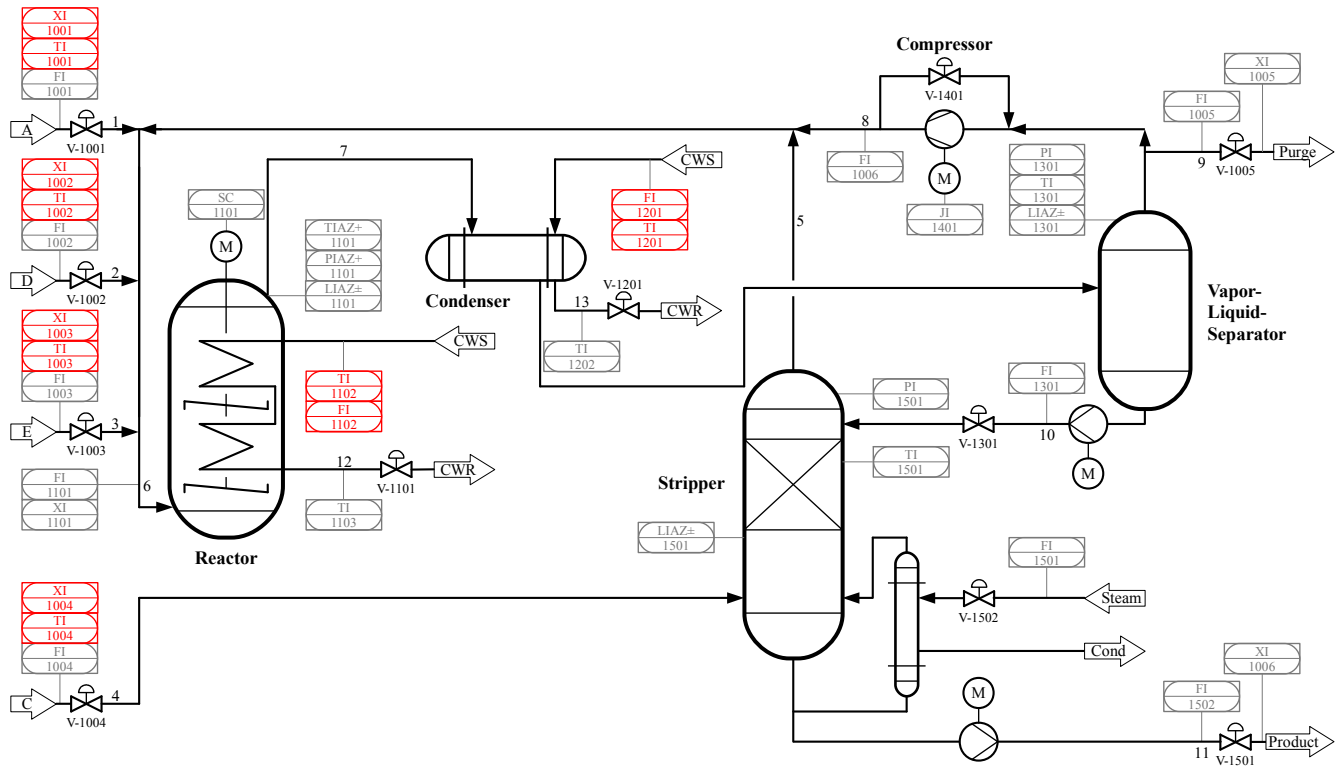


Fig. 3. P&ID of the revised process model; additional measurements in red

### 3. REVISION AND EXTENSION OF THE PROCESS MODEL

During the revision, two major aspects were considered. On the one hand the algorithms and structure of the code were revised, so as to adapt it to the structure of the simulation loop. On the other hand the model was extended in terms of additional outputs, new process disturbances and so forth.

#### 3.1 Changes to the simulation model

With regard to the code the following three objectives were accomplished:

- Revision of the generation of random numbers and the update mechanism of the process disturbances
- Revision of the execution of *tefunc()* in order to adapt the program to the needs of the respective step of the simulation loop
- Revision of the model's data structure

**Revision of the disturbance generation** As stated in the preceding section, the differences of the simulation results can be traced back to the usage of the random number generator. Thus the revised model is equipped with two random number generators for both the measurement noise and the process disturbances by adding a second state to the existing random state. Now, there is one state for the generation of the measurement noise and another for the generation of the process disturbances. Hence, the number of calls of the function *tefunc()* will no longer influence the generation of the process disturbance directly. The second modification, which had to be made in this context,

concerns the recalculation of the coefficients of the disturbance polynomials themselves. Since the time points of the recalculation, which are determined during the previous calculations, are not synchronized with the simulation calculations, the actual recalculation will be done during the subsequent call of *tefunc()* (relative to the determined time of the recalculation). Due to the program flow of each solver, it is possible that one solver recalculates the coefficients during the inner loop, whereas another solver does this recalculation only when the next major time step has been reached. As long as only one disturbance process needs to be recalculated, there are no problems. But as soon as two or more recalculations are needed between two major time steps, the recalculation order may vary. This again results in different simulation results. Thus the recalculation algorithm was altered to ensure that the order is based on the time of the recalculation and not on the index of the process disturbance. With this second modification the generation of the process disturbances is completely independent of the choice of the solver.

**Execution of the main-function** As shown in Fig. 2, every time the outputs or derivatives of the simulation model are determined, the whole model (*i.e.* code) is executed. However, this is unnecessary since the derivatives are not needed when the outputs are determined and vice versa. Furthermore, the process data, which describe the process's current state by means of temperatures, levels, etc. of the process units, need to be calculated only once during a (minor/major) time step. The process data are determined based on the states of the model. That is, only the essential values are used as the states of the model, while the remaining ones are calculated using these state values. Therefore, *tefunc()* is divided into the calculation

Table 1. Outputs of the revised model

Number	Description	Base Value	Unit
1 - 41	see Table 4 and Table 5 in [Downs and Vogel, 1993, p. 249]	–	–
42	Temperature A feed (stream 1)	45	°C
43	Temperature D feed (stream 2)	45	°C
44	Temperature E feed (stream 3)	45	°C
45	Temperature A and C feed (stream 4)	45	°C
46	Reactor cooling water inlet temperature	35	°C
47	Reactor cooling water flow	93.37	m <sup>3</sup> /h
48	Condenser cooling water inlet temperature	40	°C
49	Condenser cooling water flow	49.37	m <sup>3</sup> /h
50 - 55	Composition of A feed (stream 1); components A through F	base values of outputs	mol%
56 - 61	Composition of D feed (stream 2); components A through F	52 - 75 are given in	mol%
62 - 67	Composition of E feed (stream 3); components A through F	Table 1 of	mol%
68 - 73	Composition of A and C feed (stream 4); components A through F	[Downs and Vogel, 1993, p. 247]	mol%

of the process data and outputs, which is based on the states of the model, and the calculation of the derivatives, which is based on the process data. These two parts are called subject to the current position within the simulation loop.

**Data structure** The original model has one global data set (with respect to the whole simulation model, *i.e.* mdl-file), containing all the information of the process's current state. Thus it is not possible to simulate more than one model at a time, since two models would override each others process data. In order to overcome this problem, the data set is encapsulated in one model-specific data record. During the initialization of the simulation model, memory is allocated for each data record, allowing for multiple process models to run within one simulation model without interfering with each other.

### 3.2 Additional changes

The second group of changes were centered around the model. The main intent was the widening of the abilities of the process model. This group can be subdivided into the extensions of the outputs and the extensions of the parameter set.

The piping and instrumentation diagram (P&ID) of the process with extended measurements is shown in Fig. 3. The additional measuring points are shown in red. The measuring points and valves are numbered using four-digit numbers whose values are assigned with respect to the five units of the process. The first two digits specify the unit; 11 is for the reactor, 12 for the condenser, 13 for the separator, 14 for the compressor, 15 for the stripper and 10 is for the overall equipment. The description of the measuring points is given in Table 1. In terms of the program, the additional measurements are appended to the existing output array (output vector in MATLAB's terminology).

Furthermore, three new output groups (in terms of the model three output arrays) serving the purpose of super-

Table 2. Monitoring output of random-variation-disturbances

Number	IDV	Description	Unit
1	8	A proportion in stream 4	mol%
2	8	B proportion in stream 4	mol%
3	8	C proportion in stream 4	mol%
4	9	D feed temperature (stream 2)	°C
5	10	A and C feed temperature (stream 4)	°C
6	11	Reactor cooling water inlet temperature	°C
7	12	Condenser cooling water inlet temperature	°C
8	13	Variation coefficient of reaction kinetics of $A + C + D \rightarrow G$	1
9	13	Variation coefficient of reaction kinetics of $A + C + E \rightarrow H$	1
10	16	Variation coefficient of the steam supply of the heat exchanger of the stripper (originally specified as unknown)	1
11	17	Variation coefficient of heat transfer in reactor (originally specified as unknown)	1
12	18	Variation coefficient of heat transfer in condenser (originally specified as unknown)	1
13	20	unknown	1
14	21	A feed temperature (stream 1)	°C
15	22	E feed temperature (stream 3)	°C
16	23	A feed flow (stream 1)*	kmol/h
17	24	D feed flow (stream 2)*	kmol/h
18	25	E feed flow (stream 3)*	kmol/h
19	26	A and C feed flow (stream 4)*	kmol/h
20	27	Reactor cooling water flow*	m <sup>3</sup> /h
21	28	Condenser cooling water flow*	m <sup>3</sup> /h

\*Nominal flow (if the position of the respective valve is 100%)

vision and monitoring of the simulation were added to the existing output. The first one contains the (partly internal) values which can be subjected to process disturbances (disturbances of the type “random variation”). The structure of this output is given in Table 2. The second group of outputs are monitoring outputs of the process and is composed of internal values of the reactor and the process as well as of cost measurements; see Table 3. These values might be also used for the purpose of validation of observer strategies. In the practice, these values aren't available by normal measurements. The basis for this output stems from the FORTRAN source of Ricker [2005] (values 5-8, 10-12, 59) and from Downs and Vogel [1993] (value 61). The third output group consists of the substance concentration of the process stages of the model. For a description of these outputs please refer to the annotations at the beginning of the source code of the revised model.

The extension of the parameter set came along with the changes to the program and the model. In order to generate different disturbance characteristics, the random generator was equipped with a parameter to set the initial value of the states. All integer values except for 0 are valid. Furthermore, a second parameter was introduced serving the purpose to switch the changes made to the original model on or off. This integer parameter is evaluate based on its binary value. Each bit is assigned to one program adjustment. Table 4 sums up the parameters of the two models and the assignment of the bits. Thus the model

Table 3. Monitoring output of reaction and process; values 1 - 15 are related to the reactor

Number	Description	Unit
1	Conversion rate component A	kmol/h
2	Conversion rate component C	kmol/h
3	Conversion rate component D	kmol/h
4	Conversion rate component E	kmol/h
5	Conversion (production) rate component F	kmol/h
6	Conversion (production) rate component G	kmol/h
7	Conversion (production) rate component H	kmol/h
8	Partial pressure of component A	kPa abs
9	Partial pressure of component B	kPa abs
10	Partial pressure of component C	kPa abs
11	Partial pressure of component D	kPa abs
12	Partial pressure of component E	kPa abs
13	Partial pressure of component F	kPa abs
14	Partial pressure of component G	kPa abs
15	Partial pressure of component H	kPa abs
16 - 21	Delay-free and disturbance-free measurements of reactor feed analysis	mol%
22 - 29	Delay-free and disturbance-free measurements of purge gas analysis	mol%
30 - 34	Delay-free and disturbance-free measurements of product analysis	mol%
35 - 40	Delay-free and disturbance-free measurements of feed A analysis	mol%
41 - 46	Delay-free and disturbance-free measurements of feed D analysis	mol%
47 - 52	Delay-free and disturbance-free measurements of feed E analysis	mol%
53 - 58	Delay-free and disturbance-free measurements of feed C analysis	mol%
59	Production costs related to product quantity based on measurements	ct/(kmol product)
60	Production costs related to product quantity based on (disturbance free) internal values	ct/(kmol product)
61	Production costs related to time based on measurements	USD/h
62	Production costs related to time based on (disturbance free) internal values	USD/h

can be adjusted to the needs of the simulation, *e.g.* the corrected disturbance generation is activated by setting its value to  $96_{10} = 1100000_2$ . The revised execution structure can only be switched off by using Bit 15 of the structure parameter, otherwise the modified execution structure is always active.

Moreover, the disturbances are augmented with eight disturbances of the type “random variation”; see Table 5. The parameter vector of the disturbance activation is further used to scale the amplitudes of the disturbances between 0% and 100%. The respective values of the activation parameter are required to lie between 0 and 1.

#### 4. COMPARISON OF MODELS

In the following, the results of two comparative studies are presented. The first comparison is the solver comparison of Section 2 using the revised model. The second comparison considers the simulation duration of the original and revised model.

Table 4. Parameter list of the s-functions `temex_mod` and `temexd_mod` (abbreviated with `te_m` and `ted_m`)

Parameter		Description																				
te_m	ted_m																					
1	1	Array of the initial values of the 50 states of the model; if empty the default values of Mode 1 (see Downs and Vogel [1993]) are used																				
2	-	[only for temex_mod] Array of the activation of the 20 disturbances (see Table 8 in Downs and Vogel [1993] p. 250); if empty the disturbances are switched off																				
3	2	Initial value (seed) of the state of the random generator																				
4	3	Model structure flag																				
		<table><tr><th>Bit</th><th>Description</th></tr><tr><td>0</td><td>Additional measurement points (in Fig. 3 red)</td></tr><tr><td>1</td><td>Monitoring outputs of the values subjected to random variations</td></tr><tr><td>2</td><td>Monitoring outputs of the reaction and process</td></tr><tr><td>3</td><td>Monitoring outputs of the component's concentration</td></tr><tr><td>4</td><td>Deactivation of measurement noise</td></tr><tr><td>5</td><td>Random generator uses different state variables for the process disturbances and measurement noise</td></tr><tr><td>6</td><td>Solver-independent calculation of the process disturbances</td></tr><tr><td>7</td><td>Disturbances are scaled by the value of the activation flags</td></tr><tr><td>15</td><td>Reset model structure to original structure of Ricker [2005]</td></tr></table>	Bit	Description	0	Additional measurement points (in Fig. 3 red)	1	Monitoring outputs of the values subjected to random variations	2	Monitoring outputs of the reaction and process	3	Monitoring outputs of the component's concentration	4	Deactivation of measurement noise	5	Random generator uses different state variables for the process disturbances and measurement noise	6	Solver-independent calculation of the process disturbances	7	Disturbances are scaled by the value of the activation flags	15	Reset model structure to original structure of Ricker [2005]
Bit	Description																					
0	Additional measurement points (in Fig. 3 red)																					
1	Monitoring outputs of the values subjected to random variations																					
2	Monitoring outputs of the reaction and process																					
3	Monitoring outputs of the component's concentration																					
4	Deactivation of measurement noise																					
5	Random generator uses different state variables for the process disturbances and measurement noise																					
6	Solver-independent calculation of the process disturbances																					
7	Disturbances are scaled by the value of the activation flags																					
15	Reset model structure to original structure of Ricker [2005]																					

Table 5. Extended process disturbances (21 - 28 are random variations)

Number	Description
IDV(1) - IDV(20)	see Table 8 in [Downs and Vogel, 1993, p. 250]
IDV(21)	A feed temperature (stream 1)
IDV(22)	E feed temperature (stream 3)
IDV(23)	A feed pressure (stream 1)
IDV(24)	D feed pressure (stream 2)
IDV(25)	E feed pressure (stream 3)
IDV(26)	A and C feed pressure (stream 4)
IDV(27)	pressure fluctuation in the cooling water re-circulating unit of the reactor
IDV(28)	pressure fluctuation in the cooling water re-circulating unit of the condenser

The results of the solver comparison are shown in Fig. 4. The settings of the simulations are the same as stated in Section 2. In order to use the modified disturbance generation, the model structure flag was set to  $96_{10}$ . As Fig. 4 shows, the simulation results remain the same irrespective of the chosen solver.

In order to facilitate open-loop simulations with a length of 15 h (instead of the maximal possible 3.16 h, if Mode 1 of Downs and Vogel [1993] is used), the initial values and input values of Mode 1 as given by Ricker [1995] (see Ricker [2005]) were used for the comparison of the simulation durations. The comparison of run-times is done with respect to the original model, the revised model using a model structure flag with the value 0 (only modified program execution) and the revised model using a model

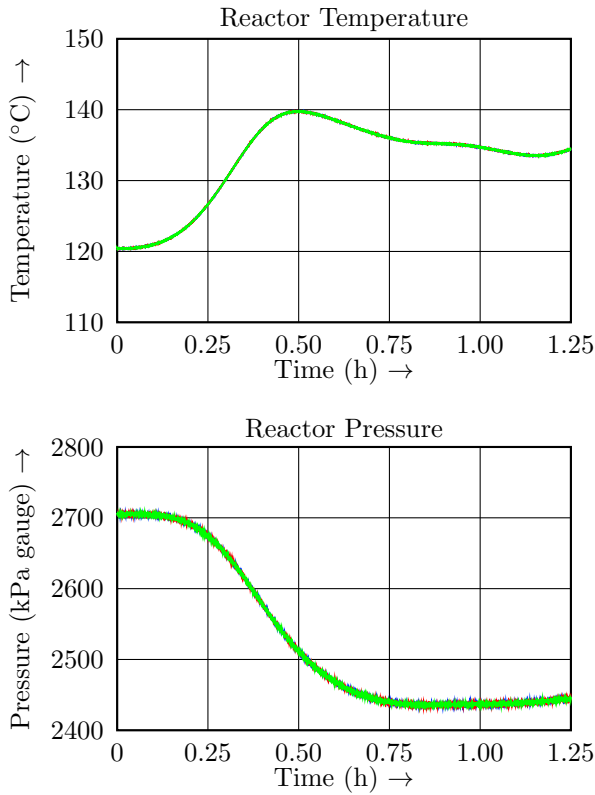


Fig. 4. Simulation results of the reactor pressure and reactor temperature using the revised model (blue: Euler/ode1; red: Bogacki-Shampine/ode3; green: Dormand-Prince/ode45)

Table 6. Comparison of simulation durations (mean and standard deviation of 250 runs)

Solver	Test case	Time [sec]
Euler (ode1) $t_{step} = 1$ s $t_{length} = 15$ h	Original model by Ricker [2005]	$3.249 \pm 0.003$
	Revised model	$1.836 \pm 0.005$
	Revised model without disturbance correction	$1.832 \pm 0.001$
Bogacki-Shampine (ode3) $t_{step} = 1$ s $t_{length} = 15$ h	Original model by Ricker [2005]	$9.286 \pm 0.004$
	Revised model	$5.086 \pm 0.005$
	Revised model without disturbance correction	$5.082 \pm 0.011$
Dormand-Prince (ode45) $t_{step,max} = 1$ s $t_{step,min} = \text{auto}$ $t_{length} = 15$ h	Original model by Ricker [2005]	$19.900 \pm 0.170$
	Revised model	$11.538 \pm 0.005$
	Revised model without disturbance correction	$11.535 \pm 0.004$

Intel Core i7-3770 CPU @ 3.4 GHz, 16 GB RAM  
MATLAB Version 7.10.0.499 (2010a), 32-bit  
Simulink Version 7.5  
temexd\_mod Version 1.3.3

structure flag with the value 96<sub>10</sub> (modified disturbance generation). Since the program execution is, however, independent of the activation of the process disturbances, no disturbances were used. The results in Table 6 are based on the same time increment (*e.g.* as it would be in the case of system identification) and the results in Table 7 are based on time increments which are approximately the upper bound for each method. In fact, the time increment should not exceed 0.1 h, since the correct execution of

Table 7. Comparison of simulation durations with changed increments (mean and standard deviation of 250 runs)

Solver	Test case	Time [sec]
Bogacki-Shampine (ode3) $t_{step} = 4$ s $t_{length} = 15$ h	Original model by Ricker [2005]	$2.338 \pm 0.003$
	Revised model	$1.287 \pm 0.002$
	Revised model without disturbance correction	$1.286 \pm 0.002$
Dormand-Prince (ode45) $t_{step,max} = 0.1$ h $t_{step,min} = \text{auto}$ $t_{length} = 15$ h	Original model by Ricker [2005]	$3.113 \pm 0.006$
	Revised model	$1.828 \pm 0.012$
	Revised model without disturbance correction	$1.825 \pm 0.010$

Intel Core i7-3770 CPU @ 3.4 GHz, 16 GB RAM  
MATLAB Version 7.10.0.499 (2010a), 32-bit  
Simulink Version 7.5  
temexd\_mod Version 1.3.3

the code of the Tennessee Eastman Process model limits the increment to this value. The altered execution regime shows a decrease in the simulation duration of at least 40% in each case. With respect to longer simulations (10 h and beyond), the extension of the simulation duration due to the modified disturbance generation is negligible.

## 5. SUMMARY

A modified version of the program of the Tennessee Eastman Process model is presented. This version features the following properties:

- Simulation results (with active process disturbances) are independent of solver and value of time increment.
- Faster simulation: time consumption down by approx. 45% compared to original model
- Model with local data set enables the usage of several models within one simulation without mutual interference.
- Additional process measurements and process disturbances
- Monitoring outputs of the values of the process disturbances and internal values

## REFERENCES

- A. Bathelt and M. Jelali. Comparative study of subspace identification methods on the Tennessee Eastman Process under disturbance effects. In *Proceedings of the 5th International Symposium on Advanced Control of Industrial Processes*, 2014.
- J. J. Downs and E. F. Vogel. A plant-wide industrial process control problem. *Computers & Chemical Engineering*, 17(3):245–255, 1993.
- N. L. Ricker. Optimal steady-state operation of the Tennessee Eastman Challenge Process. *Computers & Chemical Engineering*, 19(9):949–959, 1995.
- N. L. Ricker. Tennessee Eastman Challenge Archive, 2005. URL <http://depts.washington.edu/control/LARRY/TE/download.html>.
- S. Yin, S. X. Ding, A. Haghani, H. Hao, and P. Zhang. A comparison study of basic data-driven fault diagnosis and process monitoring methods on the benchmark Tennessee Eastman Process. *Journal of Process Control*, 22(9):1567–1581, 2012.