Simulation of Power Plant Superheater Using Advanced Simulink Capabilities

Pavel Nevriva, Stepan Ozana, Martin Pies

Abstract — The paper deals with simulation of both dynamics and control of power plant superheaters by means of Simulink S-functions. Superheaters are heat exchangers that transfer energy from flue gas to superheated steam. A composition of superheater, its input and output pipelines, and fittings is called a superheater assembly. Inertias of superheater assembly are often decisive for design of a steam temperature control system. Mathematical model of a superheater assemble is described by sets of nonlinear partial differential equations. To analyze accuracy of the mathematical model, the system was agitated by test signals. Experiments carried out at the power plant were simulated mathematically. Data obtained by the measurement was compared with simulation results**.**

***Keywords* —** modeling, simulation, Matlab&Simulink, S-Functions, superheater.

I. Introduction

New high-temperature heat exchangers are being developed nowadays. They are built using new constructions, they may operate under new conditions or may use new materials and working media, such as mixture of CO2/water vapor or air/water vapor. New facilities and blocks for transformation of primary fossil energy into electrical energy are being developed, with a focus on reducing CO2 as a greenhouse gas. There are several branches in development of such blocks. One of them is concentrated on reducing CO2 in burning process; this category covers development of systems with as high efficiency of primary energy transformation as possible. One of the examples can be energy blocks using Rankin – Clausius cycle with supercritical parameters that have about 45% efficiency, or energy blocks working on so called vapor-gas principals using combination of Brayton and Rankin – Clausius cycles that can reach up to 60% efficiency. The interchange of energy from chemical to electrical one made in fossil thermal power plant is a complex process. Mathematical model of this process enables operator to optimize the control of the actual plant and the designer to optimize the design of the future plants.

There are many units that are situated in the main technological chain of the thermal power plant. All of them can be described mathematically and included in the mathematical model of the plant. This paper deals with power plant heat exchangers, particularly with superheaters. Superheaters are parts of the power plant boiler. They transfer heat energy from flue gas to superheated steam. Superheaters are connected to the other parts of the boiler by pipelines and headers. Inertias of heat exchangers and their pipelines are often decisive for the design of the power plant steam temperature control system.

Mathematical model of the steam exchanger was developed in [1]. Mathematical model of a pipeline or a header can be developed from the mathematical model of the heat exchanger by introducing zero heat transfer coefficient from the surrounding (making the pipeline isolated). The models comprise many coefficients. Coefficients of pipelines and headers are usually known with the relatively good accuracy. Let us consider the mathematical model of the superheater assembly comprising superheater, its associated pipelines and pipe fittings. The accuracy of the model would depend on both the accuracy and correctness of coefficients of the model of the superheater. In this paper, the deterministic verification of the mathematical model of the superheater and its associated parts is presented. The verification process was as follows. The superheater assembly of operating 200 MW power plant was agitated by the set of long term forced input signals. The dynamic responses were both measured and simulated. The measured and calculated results were compared. The paper presents results of selected experiments.

II. SIMULINK S-FUNCTION

S-functions (system-functions) provide a powerful mechanism for extending the capabilities of Simulink. This paragraph describes what S-function is and when and why it is convenient to use one, see Fig.1.

S-functions make it possible to add customized algorithms to Simulink models, either written in MATLAB or C. By following a set of simple rules it possible to implement the algorithms in an S-function. After S-function has been written and placed its name in an S-Function block (available in the User-defined Functions sublibrary), it’s time to customize the user interface by using masking. An S-function is a computer language description of a dynamic system. S-functions can be written using MATLAB or C. C language S-functions are compiled as MEX-files using the mex utility described in the Application Program Interface Guide. As with other MEX-files, they are dynamically linked into MATLAB when needed. S-functions use a special calling syntax that enables you to interact with Simulink’s equation solvers. This interaction is very similar to the interaction that takes place between the solvers and built-in Simulink blocks. The form of an S-function is very general and can accommodate continuous, discrete, and hybrid systems. As a result, nearly all Simulink models can be described as S-functions. The most common use of S-functions is to create custom Simulink blocks. S-functions can be effectively used for a variety of applications, such as adding new general purpose blocks to Simulink, incorporating existing C code into a simulation, describing a system as a mathematical set of equations, using graphical animations. An advantage of using S-functions is that it is possible to build a general purpose block that can be used many times in a model, varying parameters with each instance of the block.

An M-file or a CMEX-file that defines an S-Function block must provide information about the model; Simulink needs this information during simulation. As the simulation proceeds, Simulink, the ODE solver, and the M-file interact to perform specific tasks. These tasks include defining initial conditions and block characteristics, and computing derivatives, discrete states, and outputs. Simulink provides a template M-file S-function that includes statements that define necessary functions, as well as comments to help with writing the code needed for a particular S-function block. C MEX-file S-functions have the same structure and perform the same functions as M-file S-functions. In addition, C MEX S-functions provides more functionality than M-file S-functions.

III. SUPERHEATER ASSEMBLY

The mathematical model of the heat exchanger was specified for the parallel flow output superheater of the 200 MW block of Detmarovice thermal power station, EDE. The EDE is the 800 MW coal power plant of CEZ joint-stock company, it is equipped with very modern digital controllers and computer control system. The specification of the model was made with the assistance of the thermal and hydraulic boiler calculation that defines operating parameters of the superheater. It also defines various operating steady-state values of state variables at both the input and the output of the superheater. It does not cover all parameters of the model and functional dependences of parameters.

The basic useful method to check the model accuracy is to compare selected steady-state values of physical variables obtained by simulation with values specified by the thermal and hydraulic boiler calculation. The better method to check the model accuracy is to compare selected characteristics and time responses obtained by superheater simulation with characteristics and time responses obtained by measurement on the actual power plant superheater. Such quantification of accuracy needs the suitable selection of characteristics and responses. This paper compared the responses of both the actual superheater plus associated piping and its mathematical model to forced input signals perturbations. The closed loop control system is not suitable for this purpose, because a desired change of the setpoint cannot be performed in real operation due to the technological safety reasons, thus this only simulation is provided here.

The effect of accuracy of coefficients of mathematical model of superheater on the resulting transients is due the feedback very small. To assess the accuracy of the mathematical model, experiments have to be done on the open loop system, see paragraph 5. Fig. 2 shows the scheme comprising the superheater, piping, and the basic controllers that stabilize the temperature of steam at the output of superheater assembly.

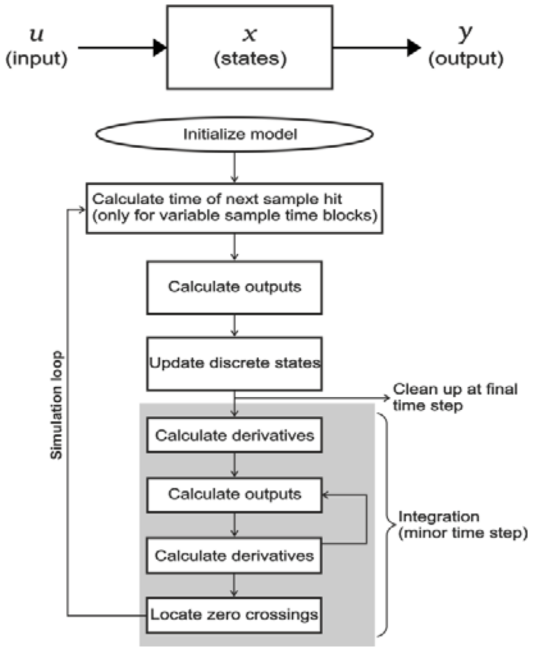


Fig. 1 scheme of the superheater assembly

The inlet superheated steam enters the mixer. The outlet superheated steam leaves the last pipeline. The control circuit includes two control loops. The fast loop with PI controller regulates the water flow rate by the valve injection to balance the temperature behind the mixer. The main loop with PID controller stabilizes superheater assembly outlet steam temperature. Superheater assembly being controlled consists of the input section, parallel flow superheater (SH), and the output section.

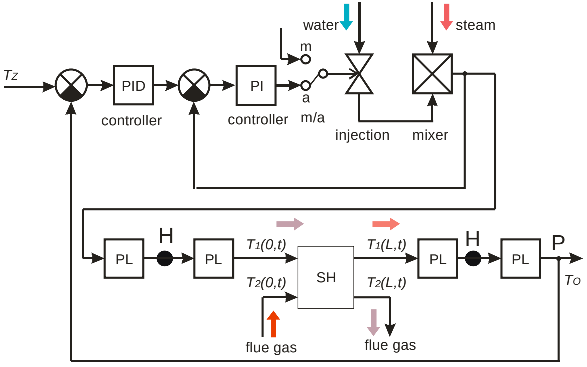


Fig. 2 scheme of the superheater assembly

Both input and output section consists from two pipelines (PL) separated with a header (H). The manual to automatic control switch m/a is set to the automatic control mode, and the assembly outlet steam temperature measured at point P is stabilized to the set point value Tz = 540°C.

The closed loop control loop process was simulated in MATLAB&Simulink. Data for simulation were accumulated by measurement on EDE. The basic scheme is shown in Fig. 3.

Fig. 4 shows one typical simulation task. This experiment cannot be carried out on the operating power plant. It is not possible to enter such a set point difference to the power plant equipped with the actual closed loop control system. Fig. 4 relates to the superheater that is operating under standard operating conditions. Superheater and its feedback control system are shown in Fig. 2. At time t = 0, the superheater is in its steady state, and the set point value Tz is changed from 520°C to 540°C. The simulated time response of the outlet temperature of the superheater assembly initiated by both the outlet temperature set point step change of 20°C and actual deviations of input signals is displayed in Fig. 4. Positions of signals are shown in Fig. 2.

IV. VERIFICATION OF THE MATHEMATICAL MODEL

There are six input variables in the mathematical model. The output variables of interest are temperatures of steam and flue gas and pressure of steam. The change of each input variable results in time responses all of output variables. It would be advantageous to set all but one input signals constant and study the responses of the system item-by-item. At the operating power plant, it is not a simple problem.

As listed above, there are eighteen principal combinations of choice of the input to output pair of a superheater. There is also possible to insert some input signals and measure some output signals in different points of superheater assembly. It is beyond the scope of this paper to present here all possible combinations of responses. To discuss the quality and accuracy of the mathematical model, the example of presentation has been selected as follows. The input was the disturbance of the water flow rate at the controlling water injection. Note that the change of the water flow rate results in a change of both steam velocity and steam temperature at the output of the mixer. The output was the superheater assembly outlet steam temperature To.

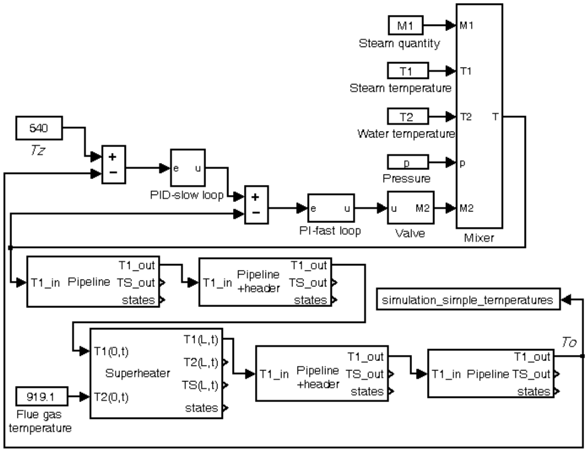


Fig. 3 closed loop temperature control MATLAB&Simulink

Layout of the experiment is shown at Fig. 5. To obtain sufficiently large values of deviations of state values and output signals, the superheater's automatic feedback control loops were disconnected during experiments.

At a 200 MW superheater, it is a rather challenging task. To deal with this problem, the presented experiments were realized at the derated power of 180 MW.

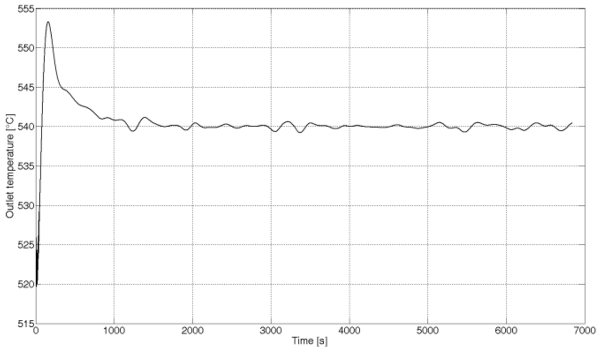


Fig. 4 simulated outlet temperature at the feedback control system

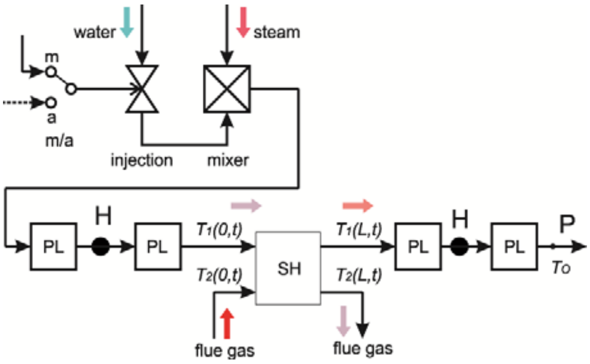


Fig. 5 layout of the open loop experiment

Note that at the output superheater the outlet steam is technologically stabilized and lead to the high-pressure part of turbine. The discussion of technological stabilization is beyond the scope of this paper. To disconnect the feedback loops, the control of the controlling water injection was set to the manual mode. The superheater assembly outlet temperature To was not controlled, only kept within safety limits by the operator. The open loop temperature control process was simulated in MATLAB&Simulink. The basic scheme is shown in Fig. 6.

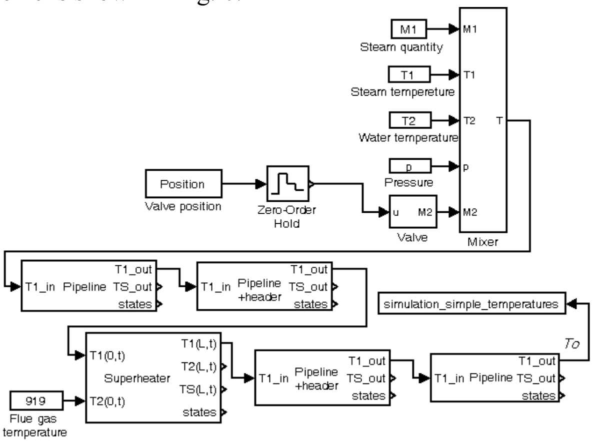


Fig. 6 open loop temperature control scheme

Every measurement was approximately for two hours in length. All necessary input and output variables were measured automatically and processed and evaluated by the model. Data were measured in three second sampling interval.

V. Measuring of Heat exchanger by stochastic signals

The Detmarovice power station made it possible to carry out experiment concerning verification of the mathematical model with the measurements on real block (under full operation) by methods of stochastic linear analysis. Steam temperature $T_1(0,t)$ was chosen to be the input signal and $T_1(L,t)$ stands for output signal.

What speaks for usage of stochastic analysis is firstly the fact that it can be used for systems under real operation. Secondly, in our case, both signals contain superposed noise with frequency characteristics proximate to white noise and the plant can be considered as system with constant parameters that can be quite precisely approximate in operating point. To be able to use the methods of stochastic analysis of linear systems it is necessary to dispose with a huge and very time-consuming measurements. At the same time the output and input signals are correlated.

Fig. 7 shows the setup of the first measurement when the input temperature $T_1(0,t)$ and output temperature $T_1(L,t)$ were measured. The flue gas $T_2(0,t)$ was computed backward based on knowledge of signals $T_1(0,t)$, $T_1(0,t)$ and mathematical model of superheater with parameters corresponding to a real superheater in Detmarovice power station.

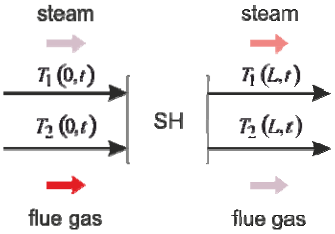


Fig. 7 Measurement of the superheater by stochastic signals

Fig. 8 shows the measured steam temperature at the output of superheater $T_1(L,t)$ compared to the superheater output temperature computed by mathematical model (referred to as $T_{1sim}(L,t)$).

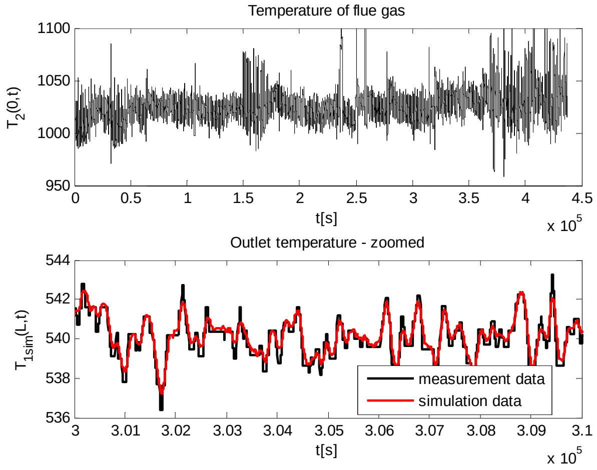


Fig. 8 Calculated flue gas temperature $T_2(0,t)$ together with comparison of output superheater temperatures $T_1(L,t)$ and $T_{1sim}(L,t)$

Measured signals from real operation make up ten-day record from July/August 2009. The records are separated from daily periods when the power plant's wattage was 180MW, with sampling period of Ts = 3 seconds.

This paper describes identification of a superheater assembler in a control circuit. Firstly, it is the identification of impulse and step response of the superheater, whose input signal $T_1(0,t)$ is affected by other signals present in control circuit. Secondly, it is the identification of stem temperature response at the superheater's output $T_1(L,t)$ to the temperature T_p, that is brought to the mixer. In fact it is determination of impulse and step response of disturbance transfer function.

When computing numerical deconvolution, Wiener – Hoppf equation

$$R_{uy}(\tau)=\int_0^\infty h(t)R_{uu}(\tau-t)dt \qquad (1)$$

represents stochastic formulation of dynamic system.

Under a special condition, in case of bringing white noise into input of the system having the following autocorrelation function

$$R_{uu}(\tau)=\delta(\tau)\ , \qquad (2)$$

we get

$$R_{uy}(\tau)=\int_0^\infty h(t)\delta(\tau-t)dt = h(\tau) \qquad (3)$$

Numerical calculation of weighting function is based on replacing integration process by summation and numeric deconvolution. Discretizing equation (3) leads to

$$R_{uy}(\tau)\approx\sum_{k=0}^{N}R_{uu}(\tau-kT_S)h[kT_S]T_S \qquad (4)$$

If time shift τ is expressed as multiple of time step Ts, that is τ = 0, Ts, 2 Ts, …, N, it is possible, using the last equation, a set of (N + 1) linear algebraic equations, from which it is possible to compute unknown values of weighting function h(0), h(Ts), …, h(NTs):

$$R_{uy}[0]=(R_{uu}[0]h[0]+R_{uu}[-T_s]h[T_s]+\ldots+R_{uu}[-NT_s]h[NT_s])T_s$$
$$R_{uy}[T_s]=(R_{uu}[T_s]h[0]+R_{uu}[0]h[T_s]+\ldots+R_{uu}[T_S-NT_s]h[NT_s])T_s$$
$$\vdots$$
$$R_{uy}[NT_s]=(R_{uu}[NT_s]h[0]+R_{uu}[NT_s-T_s]h[T_s]+\ldots+R_{uu}[0]h[NT_s])T_s$$

(5)

Using following feature of autocorrelation function,

$$R_{uu}(\tau)=R_{uu}(-\tau) \qquad (6)$$

and after introduction of shortened notation of weighting function $h_k=h[kT_s]$

The set of equation can be rewritten into matrix form:

$$\begin{bmatrix}\frac{R_{uy}[0]}{T_s}\\ \frac{R_{uy}[T_s]}{T_s}\\ \vdots\\ \frac{R_{uy}[N\cdot T_s]}{T_s}\end{bmatrix}=\begin{bmatrix}R_{uu}[0] & R_{uu}[T_s] & \cdots & R_{uu}[NT_s]\\ R_{uu}[T_s] & R_{uu}[0] & \cdots & R_{uu}[(N-1)T_s]\\ \vdots & \vdots & & \vdots\\ R_{uu}[NT_s] & R_{uu}[(N-1)T_s] & \cdots & R_{uu}[0]\end{bmatrix}\cdot\begin{bmatrix}h_0\\ h_1\\ \vdots\\ h_N\end{bmatrix} \qquad (7)$$

Or in the matrix form

$$\mathbf{r} = \mathbf{R} \cdot \mathbf{h} \tag{8}$$

Solution of weighting function can be reached by use of inverse matrix as follows:

$$\mathbf{h} = \mathbf{R}^{-1} \cdot \mathbf{r} \tag{9}$$

This numerical solution of deconvolution in Matlab is limited by matrix until approximately elements.

Concerning that the length of measured data exceeds the size of the matrix that would be created during numerical solution of deconvolution, it is suitable to split the record into several same sections and compute particular impulse characteristics. The second reason for splitting is the fact that time constant of superheater is smaller than time of calculated impulse response that would be computed in case of maximal possible solution of numeric deconvolution (3000 x Ts = 9000 seconds). Due to this reason, the ergodic hypothesis was used for estimation of impulse characteristic. By means of Wiener – Hopf equation (1) was estimated the impulse characteristic (see Fig. 9) of the superheater assembled in a control circuit, and impulse characteristic of disturbance transfer function (see Fig. 10). In equation (1) signal *u* denotes temperature $T_1(0,t)$, signal *p* denotes temperature of steam, entering into mixer and signal *y* stands for temperature $T_1(L,t)$, resp. $T_{1\mathrm{sim}}(L,t)$. To get worked this method in proper way it is necessary to detrend the temperatures.

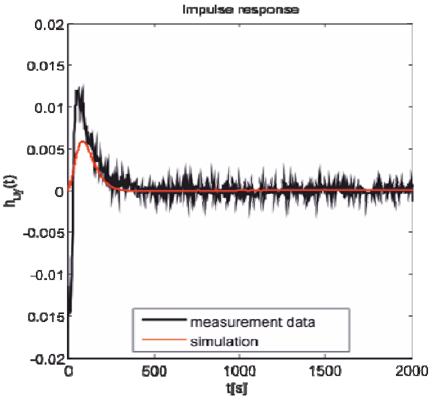


Fig. 9 Comparison of estimations of impulse characteristics of heat exchanger, assembled in a control circuit

Integrating impulse characteristic we get estimation of the step response of disturbance transfer function (see Fig. 10).

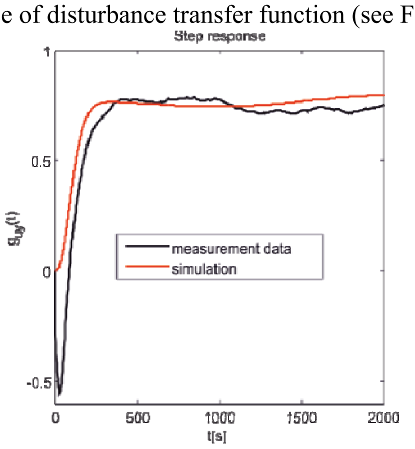


Fig. 10 Comparison of estimations of heat exchanger, assembled in a control circuit

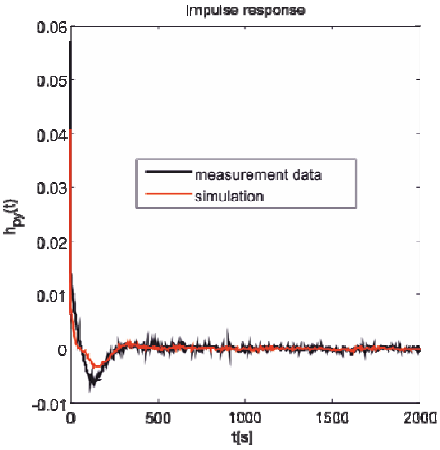


Fig. 11 Comparison of estimations of impulse characteristics of disturbance transfer function

Integrating impulse characteristic we get estimation of the step response of disturbance transfer function (see Fig. 12).

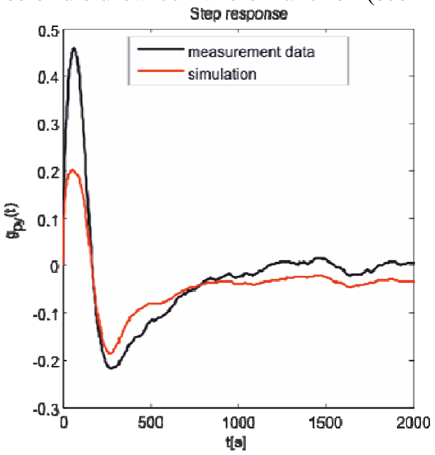


Fig. 12 Comparison of estimations of step characteristics of disturbance transfer function

Disturbance transfer function figuratively means "black box" systems representing modeled control circuit, see Fig. 2. This control circuit compensates error corresponding to steam temperature at the inlet of mixer T_p. Fig. 12 shows comparison of estimated step responses. The first one is computed from measured data while the second one is calculated from simulation of modeled control circuit driven by measured data. Evaluating dynamics of these responses, it is possible to conclude that real control circuit has very similar dynamic as its model. The difference overshoot can be resolved by the fact that temperatures T_p and $T_1(L,t)$ are correlated to a certain degree.

VI. LOADING AND SAVING COMPUTED STEADY STATE IN SIMULINK

When solving this project the need of saving the final (steady) state arose. Particularly, we want to compute the steady state of simulation, save it and use it as initial state for further experiments. Since 2009 Simulink provides such a tool, while the final it does not only save the states in the sense of state variables, but the final state of the whole simulation, including all inputs, outputs and parameters. The procedure will be shown in a simple example, see Fig. 13.

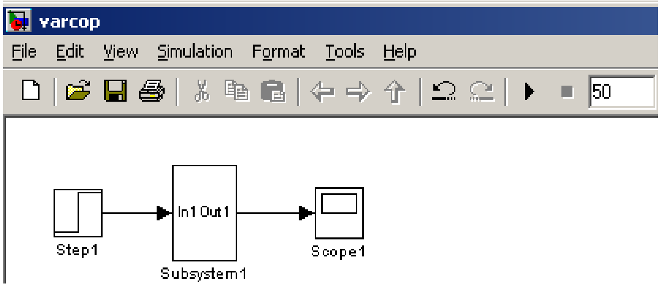
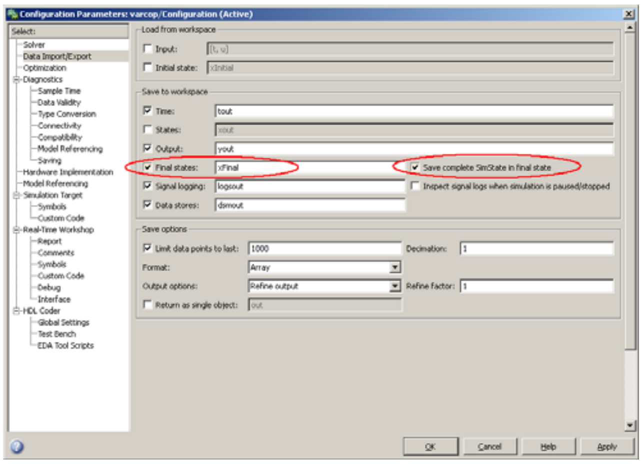


Fig. 13 Basic scheme for computing steady state

Next step is to choose Simulation, Configuration Paramaters (from the top menu), see Fig. 14.


Configuration Parameters: varcop/Configuration (Active)

Load from workspace

□ Input: [t, u]

□ Initial state: xInitial

Save to workspace

☑ Time: tout

□ States: xout

☑ Output: yout

☑ Final states: xFinal ☑ Save complete SimState in final state

☑ Signal logging: logsout □ Inspect signal logs when simulation is paused/stopped

☑ Data stores: dsmout

Save options

☑ Limit data points to last: 1000 Decimation: 1

Format: Array

Output options: Refine output Refine factor: 1

□ Return as single object: out

Fig. 14 Configuration of simulation

Then, after simulation, Fig. 15, the steady state is computed and *xFinal* variable appears in Matlab workspace.

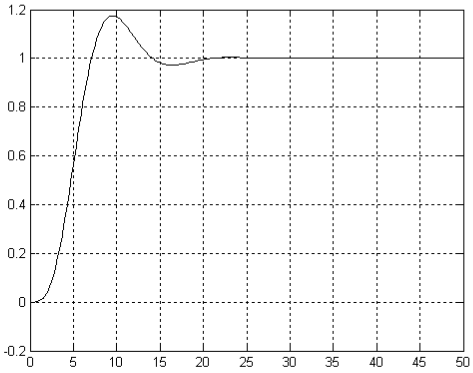


Fig. 15 Simulation result

Keep in mind that after exiting Matlab the workspace is deleted and this state kept unsaved. Saving the state can be done by storing into the file by using this syntax:

save('ustaleny_stav.mat','xFinal')

For further experiments the saved final state is considered as new initial state to be loaded from appropriate workspace variable. It can be loaded from the state using the syntax:

load('ustaleny_stav.mat')

For computing the steady state in our case it was enough to do the simulation for 50 seconds. Now the next example describes how to simulate the response to step change. The simulation final time must be set appropriately (enough), for example 100s, see Fig.16. Step has to arise at 50seconds at least, we chose 55 seconds for demonstration, see Fig. 17. Its initial value corresponds to final value in first experiments (value 1) and its value rises by one, to the value of 2.

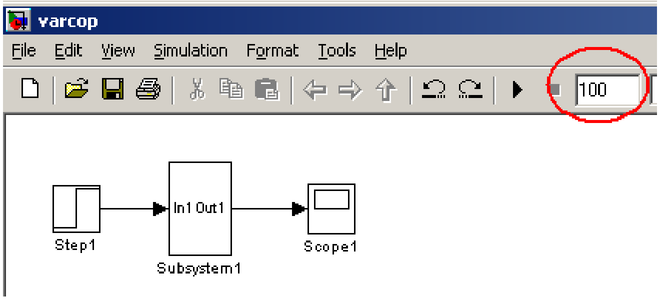
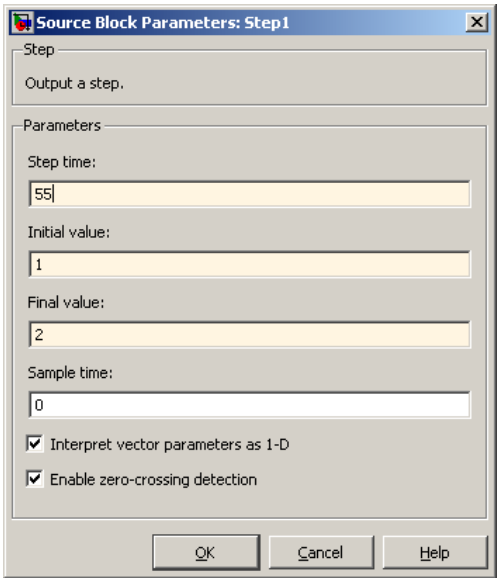


Fig. 16 Scheme for consequent experiments


Source Block Parameters: Step1

Step

Output a step.

Parameters

Step time: 55

Initial value: 1

Final value: 2

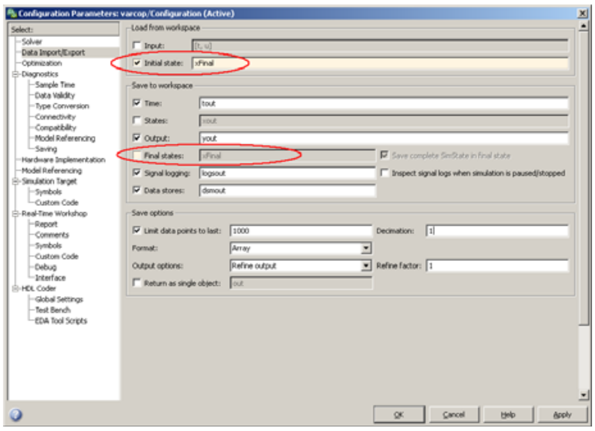
Sample time: 0

☑ Interpret vector parameters as 1-D

☑ Enable zero-crossing detection

Fig. 17 Step parameters

Now the simulation must start from previously saved state stored in *xFinal* variable. At the same time we need to determine if the new final state is about to be saved again, and if this can rewrite the previous state or not. If *Final states* stays checked, the variable would be rewritten. Of course the name can be different, can be conserved into the file or it can stay unchecked, see Fig. 18.



Load from workspace

□ Input: [t, u]

☑ Initial state: xFinal

Save to workspace

☑ Time: tout

□ States: xout

☑ Output: yout

□ Final states: xFinal ☑ Save complete SimState in final state

☑ Signal logging: logsout □ Inspect signal logs when simulation is paused/stopped

☑ Data stores: dsmout

Save options

☑ Limit data points to last: 1000 Decimation: 1

Format: Array

Output options: Refine output Refine factor: 1

□ Return as single object: out

Fig. 18 Configuration of simulation

After simulation, we see the new response for consequent experiment, see Fig.19:

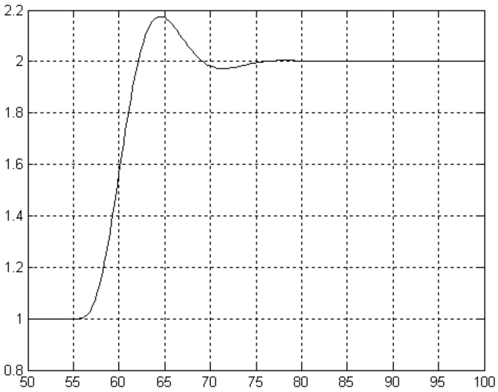


Fig. 19 Results of consequent experiment

Output starts from previous steady state and in time t=55s when there was change of the input signal we see the start of a new response.

Main disadvantage of this tool is the fact that the structure of circuit has to stay the same between computing the steady state and use it for new experiments (wiring and number of blocks must be the same. Example: We compute steady state as a response to step input signal. At the same time we know that for next work we will use other type of signal, for example square impulses. Block Step must not be replaced, but the situation can be solved by using Manual Switch in the circuit structure (in library Signal Routing), see Fig.20. Then it is possible to switch appropriate branch of the input signal and use the tool for saving and loading the intital/final state.

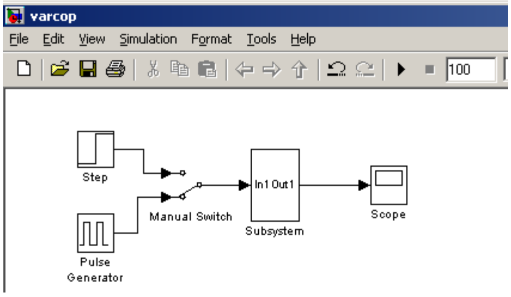


Fig. 20 Adding manual switch into the scheme

The described tool can be especially used for very large and complex schemes where computation of steady state takes a very long time and it would have to be computed every time for each experiment. This is illustrated on example of FES system (flexible energy system) that contains several blocks reheaters, pipelines and turbine with compressor blocks, see Fig.21. Ten thousands of simulation time (reaching the steady state) takes approximately 20 minutes of computation with quadruple processor PC.

There are other possibilities how to handle with the condition for wiring and structure of the circuit. For example, at time t=6000s there’s step change of flue gas temperature (increasing by 10 degrees). It can be easily disabled by setting the step time outside the interval of simulation (step change wouldn’t arise) or setting Initial a Final value at the same levels. However, the solution with Manual Switches is more elegant and universal

.

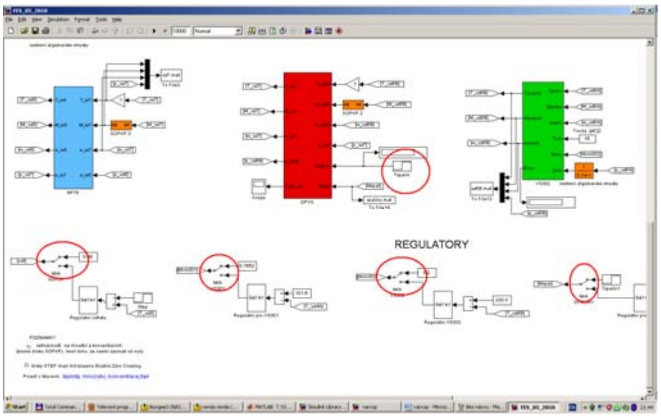


Fig. 21 Example of FES scheme

Moreover, there are 4 manual switches at the controllers to make it possible to model either manual or automatic modes of the circuit. To cope with complex circuits it is recommended to make up the structure including all blocks, then compute steady state and to the other experiments by setting up the parameters of appropriate blocks.

VII. CONCLUSION

Fig. 22 compares assembly steam outlet temperatures obtained by both measurement and simplified mathematical model. Fig. 23 presents the same measurement and compares the simulated results for the full mathematical model.

The position of output signal is shown in Fig. 5. The intensity of the forced disturbance of the water flow rate at the controlling water injection applied was the part of the experiment. The disturbance in the standard operating regime of the superheater is much smaller as well as the deviations of the outlet temperature.

As the basic model is more complex than the simplified model, it gives more precise results at both steady states and dynamics of the time responses. Comparison of Fig. 22 with Fig. 23 illustrates, that at the standard operating state the simplified model approximates the basic mathematical model very well. Outside the vicinity of the set point, the accuracy of the simplified model decreases.

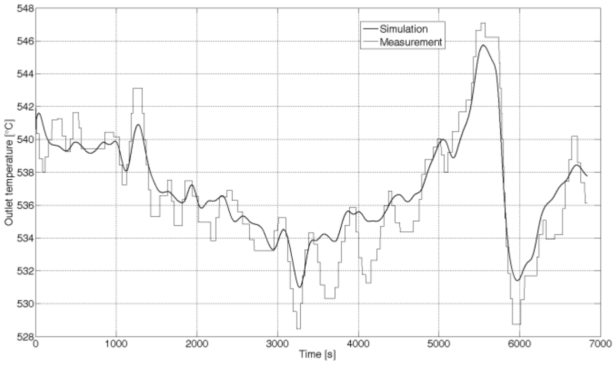


Fig. 22 Comparison of measured and simulated outlet temperatures at the open loop control system experiment (simple model)

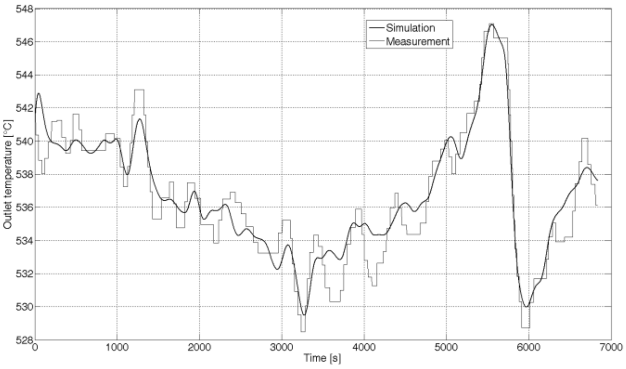
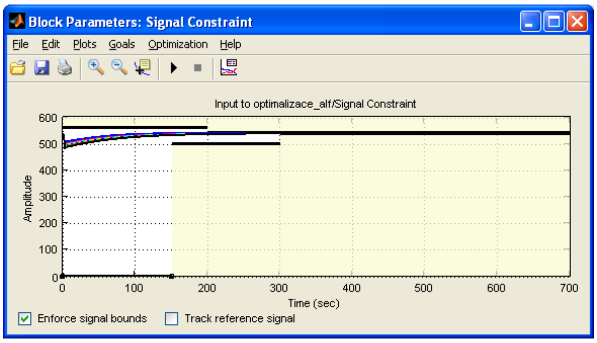


Fig. 23 Comparison of measured and simulated outlet temperatures at the open loop control system experiment (full model)

There are some reasons for differences between measured and real data. The main one is that the two coefficients of heat transfer in the model are not determined precisely. Therefore, based on input and output steady states of temperatures, the coefficients were optimized to fit these steady states using Simulink Response Optimization toolbox. There are two signal constraints for steam temperature and flue gas temperature, defining intervals for two temperatures according measured values to fit, see Fig. 24. This optimization is one of the most crucial future plans of this project.



☑ Enforce signal bounds □ Track reference signal

Fig. 24 Defining the steam temperature interval to fit the steady states according the measurements

Mentioned results and analysis of the problem show that used methodology can be successfully applied to power unit and similar plants where the wanted signal is immersed in a noise and thus a very long measurement must be done in such cases. Presented results lead to the conclusion that accuracy of stochastic identification could be increased by introducing synchronous measurement of the flue gas temperature, at least at the point $T_2(0,t)$. This paper introduced backward computation of the flue gas temperature based on knowledge of temperatures $T_1(0,t)$ and $T_1(L,t)$ and knowledge of mathematical model of the superheater including its parameters. In the frame of this mentioned sense the further research and development of the project is planned for a near future.

Acknowledgment

The work was supported by the grant “Simulation of heat exchangers with the high temperature working media and application of models for optimal control of heat exchangers”, No.102/09/1003, of the Czech Science Foundation.

References

[1] Nevriva, P., Ozana, S., Pies, M., Vilimec, L. Dynamical model of a power plant superheater. WSEAS Transactions on Systems 9 (7), pp. 774-783

[2] Nevriva, P., Ozana, S., Vilimec, L. Simulation of the heat exchangers dynamics in MATLAB&Simulink. WSEAS Transactions on Systems and Control 4 (10), pp. 519-530

[3] Ožana, Š., Macháček, Z. Implementation of the mathematical model of a generating block in Matlab&Simulink using S-functions. 2009 International Conference on Computer and Electrical Engineering, ICCEE 2009 1, art. no. 5380446, pp. 431-435

[4] Nevriva, P., Ozana, S., Vilimec, L. The finite difference method applied for the simulation of the heat exchangers dynamics. Proceedings of the 13th WSEAS International Conference on Systems - Held as part of the 13th WSEAS CSCC Multiconference , pp. 109-114

[5] Ozana, S., Pies, M. Using simulink S-functions with finite difference method applied for heat exchangers. Proceedings of the 13th WSEAS International Conference on Systems - Held as part of the 13th WSEAS CSCC Multiconference , pp. 210-215

[6] Hanuš B., Regulační charakteristiky přihřívačů páry u kotlů československé výroby. Strojírenství 11, 1961.

[7] IHI, Steam Generators, Technical Note of Ishikawajima-Harima Heavy Industries Japan, 2004.

[8] NEVŘIVA, Pavel, OŽANA, Štěpán, PIEŠ, Martin.Identification of mathematical model of a counterflow heat exchanger by methods of statistic dynamics. ICSE 2009 : Proceedings. Coventry University, ISBN 978-1-84600-0294.

[9] Kattan. P.I., MATLAB Guide to Finite Elements: An Interactive Approach. Seconnd Edition. *Springer New York 2007. ISBN-13 978-3-540-70697-7*

[10] Cook R. D. at al., Applications of Finite Element Analysis, John Wiley and Sons, 2002, ISBN 0-471-350605-0

[11] Haberman R., Applied Partial Differential Equations with Fourier Series and Boundary Value Problems, 4th Edition, Pearson Books, 2003, ISBN13: 9780130652430 ISBN10: 0130652431

[12] Dukelow S. G., The Control of Boilers, 2nd Edition, *ISA 1991, ISBN 1-55617-300-X*

[13] Jaluria Y., Torreance K.E., Computational Heat Transfer. Taylor and Francis, New York, 2003 ISBN 1-56932-477-5

[14] Chen Chi-Tsong, System and Signal Analysis, Second Edition, Sundere College Publishing, 1989

[15] Dmytruk I, Integrating Nonlinear Heat Conduction Equation with Source Term, WSEAS Transactions on Mathematics, Issue 1, Vol.3, January 2004, ISSN 1109-2769