Proceedings of the 5th WSEAS Int. Conf. on APPLIED INFORMATICS and COMMUNICATIONS, Malta, September 15-17, 2005 (pp192-197)

Discredibility Detection of a Sensor in a Control Loop via Computational Intelligence

DAVID KLIMÁNEK
BOHUMIL ŠULC
Department of Instrumentation and Control Engineering
Czech Technical University in Prague, Faculty of Mechanical Engineering
166 07 Prague, Technická 4
CZECH REPUBLIC

Abstract: When operating a control loop, the correct function of the controller depends on data usually acquired from a sensor. A problem may occur when the output information from the sensor is biased. Then malfunction detection of the sensor used for measuring the controlled variable can become a very important task. It is sometimes difficult to detect changes in the properties of the sensors, because they are not apparent from the control loop behaviour. Although biased sensor output information does not lead to a failure of the control function, we are faced with the problem that, although the control loop seems to be working properly, the consequences of a "small" malfunction (sensor discredibility) can be substantial and expensive. It is easy to imagine, e.g. a combustion ratio control where deviations from an optimal ratio value have no principal influence on the operation of the device, but late discovery of an increase in harmful emissions may be very costly. Sensor redundancy may not be an acceptable solution if it requires expensive measuring equipment. This paper describes experiments on software sensor discredibility detection as a way that replaces usual hardware redundancy and saves the costs of additional measurements. It also shows the possibility of improving a controlled system by avoiding hidden impreciseness in the control loop operation. The used tools are methods of computational intelligence that are adapted and evaluated in an example application of a level control in a two-tank cascade.

Keyords: malfunction detection, evolutionary algorithm, simulated annealing algorithm, software redundancy

1 Introduction

The notion of "sensor discredibility" refers to a type of sensor faults that do not involve a total sensor failure, but only a small deviation from its correct function. These changes are not very apparent. For example, to the outside observer, a control loop will seem to be working properly, because the fact that the sensor provides a biased measurement of the controlled variable, implying an inaccurate control process, cannot be detected without some additional measuring. Sensor discredibility detection is usually achieved with the use of measuring equipment, i.e., by hardware redundancy. This paper attempts to show new ways toward discredibility detection that differ from the usual hardware redundancy. To avoid additional costs, we are working on a way to detect sensor discredibility with the use of software tools. Software detection, or the software redundancy, can substitute one or two redundant pieces of measuring equipment that are sometimes expensive. Thus, the task is to improve the function of the controller so that it is able to indicate biased sensor properties. Only if an operator is warned, will he or she be able to asses the harmful effects resulting from an incorrect sensor function and a make decision about replacing the sensor.

According to [1], fault detection methods are classified in three general categories:

- quantitative model-based methods,
- qualitative model-based methods,
- process history based methods.

In contrast to model-based approaches, where a priori knowledge about the process is needed, it seems to be useful to use process history based methods and to employ methods of computational intelligence. The main advantage of such a solution is that necessary information about changes in sensor properties can be obtained from the standard process data that is in any case acquired and recorded for the process control.

2 Sensor discredibility detection via computational intelligence

The key to sensor discredibility detection lies in determining the estimated output of a sensor $y_{est}(t)$ and the residual variable $e(t)$ as the difference between the estimated (y_{est}) and the real sensor (y_{real}) outputs,

$$e(t) = y_{est}(t) - y_{real}(t), \qquad (1)$$

where the residual variable $e(t)$ indicates the rate of variance between the output estimated via a sensor model and the value acquired by the real sensor. The sensor model can have various forms: an equation, a table, a graph, etc. In the sensor model, some other physical quantities may be considered as inputs if they influence the sensor output. Through such an intermediating sensor, model measurement of some other quantities can be used in order to obtain the estimated output of the sensor under consideration. If we limit our reflections to the case when only the non-dynamical behavior of the sensor is taken into account by the model, then the general sensor model can be described by a scalar function

$$y_{est} = f(\boldsymbol{p}, \boldsymbol{x}), \tag{2}$$

where $\boldsymbol{x}$ is a vector of the considered sensor inputs and $\boldsymbol{p}$ represents a parameter vector of the sensor model. The relations between the inputs of the sensor model and the estimated value y_{est} as the output of the sensor model are of a different type. Most sensor models assume that the sensor output is proportional only to one input

$$y_{est} = p_1 x + p_2, \tag{3}$$

where parameter p_1 represents the gain of the sensor model (often assumed to be equal to 1), and parameter p_2 expresses the shift factor (often assumed to be equal to 0), and x is the sensor model input. Further considerations refer to this type of sensor model.

The idea underlying discredibility detection is then as follows: find a parameter vector of the sensor model $\boldsymbol{p}$ for which the residual function is minimal. It is assumed that at the time when the sensor provides correct data the parameter vector of the sensor model represents the correct parameters and the residual variable is equal or close to zero. When a sensor discredibility occurs, a parameter optimization algorithm is applied and searches new values of the parameters that will again achieve a minimum of the residual variable. In this way, a discredibility detection task is transformed to an optimization task. An extreme increase of any of the parameters may signal the beginning of sensor discredibility.

In principle any optimization methods could be used for the optimization task. The problem is that the sensor model input is an unknown dynamically changing variable. Therefore, the choice and the parameter selection must include an element of random selection from many alternatives. This is fulfilled by the methods of so called computational intelligence [3]. The high computational time requirements do not matter in the case of sensor discredibility detection, because it follows from the character of sensor discredibility that the loss of credibility is a gradual development. It differs in this way from standard fault detection, where the discovery of a fault is required to be as quick as possible.

Computational intelligence [3] includes genetic algorithms, evolution strategies, genetic programming and simulated annealing. All the considered methods contain a random component. In the following sections we will focus on an explanation of discredibility detection via the genetic algorithm method and the simulated annealing method. The usual general presentation of the methods has been transformed in a way that uses terms from the field of sensor discredibility and facilitates understanding of the used procedures.

2.1 Discredibility detection via the Standard Genetic Algorithm method

Among the large variety of known genetic algorithms, there is one, called the Standard Genetic Algorithm, that has established the basis for many versions of optimization tasks [3].

The Standard Genetic Algorithm (SGA) works with the notion of a population. In terms of sensor discredibility detection, the population represents a developing set of the parameter vectors of sensor model with which the sensor model (e.g. (3)) has a chance to approach the minimum of the residual function.

The process of discredibility detection via SGA starts by creating an initial randomly generated set of parameter vectors of the sensor model. The number of vectors of the set refers to the population size n_p. An iterative process then follows. It consists of the following steps: evaluation, selection, and crossover. This iteration runs until a stopping rule is met. The stopping rule can be specified in terms of a minimal value of the residual function, evaluation time, etc. Fig. 1 shows a pseudo-code of the genetic algorithm as a part of the optimization algorithm subsystem shown in Fig. 5. For programming this code, Matlab/Simulink programming tools have been applied.

In order to carry out sensor discredibility detection using genetic algorithm methods, the following steps are required (the procedure from [5] has been adapted for the needs of sensor discredibility detection):

1. When an initialization part starts, the evolution time is set to $t = 0$ and an initial set of parameter vectors of the sensor model $\boldsymbol{p}_t$ is randomly generated within an expected range of reasonable values for each of the parameters. For each of the parameter vectors of the sensor model, so called individuals, the value of its residual function is evaluated. Also the average value of the residual function values is also evaluated.
2. After the start of the iteration process, the selection operator is employed. The selection is used to create a new set of the parameter vectors for the sensor model from the old set. Wheel selection [5] is based on the principle of turning a

roulette wheel with many holes, and the number of holes for each of the parameter vectors of the sensor model is proportional to its residual value. The wheel is spun n_p times and each time one member of the new set is picked, where n_p represents the population size. The parameter vector of the sensor model that provides corresponding residual values lower than the average value will be replicated into the next subset for generating new parameter candidates in more copies than in the original set, and the individuals with the residual value below average will be lost.

3. Next comes the crossover operation over the randomly selected pairs of the parameter vector of the sensor model of the topical set. The probability of crossover is P_{cross}, which represents one of the SGA parameters. The selected sensor model parameters are coded into binary strings. The standard one point crossover operator takes two parameters, randomly chooses a crossing point and interchanges strings from this point onward.
4. The mutation operator mimics random mutations. The newly created parameters are coded into binary strings and one bit of each string is switched with random probability P_{mut}.
5. If the stopping criterion is not met a return to step 3 repeats the process. The stopping criterion is met when the difference between the average of the residual values from the current run and the previous run is lower than 0.5.

procedure SGA ()
begin
set the evolutionary time $t = 0$;
generate randomly the initial set of parameter vectors of the sensor model $\boldsymbol{p}_0$ (initial generation)**;**
evaluate a vector of the residual values over the generated initial generation $\boldsymbol{p}_0$;
while the stopping rule is **not** satisfied
increase the evolution time t
select a new population $\boldsymbol{p'}_t$ from $\boldsymbol{p}_{t-1}$;
mutate the selected population $\boldsymbol{p'}_t$
evaluate the vector of the residual function values for the generation $\boldsymbol{p}_t$;
end

Fig. 1 Pseudo-code of the Standard Genetic Algorithm

2.2 Discredibility detection via the Simulated Annealing method

The method of Simulated Annealing (SA) is based on an analogy with thermodynamics, according to which solids are heated and cooled gradually to a crystalline state with minimum energy. This process is known as annealing. If a solid is heated above its melting point and then cooled down, the structural properties of the solid will depend partly on the speed of cooling.

The idea of SA appeared in a paper published by Metropolis et al. in 1953 [2]. When this idea is applied to the discredibility detection problem, a crystal represents a parameter vector of the sensor model $\boldsymbol{p}$, and the energy of the crystal is declared as an absolute value of the residual variable $z = |e(\boldsymbol{p})|$. The task is to find the parameter vector of the sensor model minimizing the residual function z. The algorithm runs in iterations. In each of the iterations, a potential parameter vector of the sensor model $\boldsymbol{p}_i$ is randomly generated, and a residual value z_i is evaluated, where i is the iteration index. The values are again generated within chosen limits of the expected values for each of the parameters. If $\boldsymbol{p}_i$ provides a residual value z_i lower or equal to z_{i-1}, the vector $\boldsymbol{p}_i$ is automatically accepted as a current parameter vector of the sensor model. In the opposite case, the randomly generated vector of parameters $\boldsymbol{p}_i$ may be randomly accepted according to the Boltzmann criterion (4), as shown in Fig. 2**.**

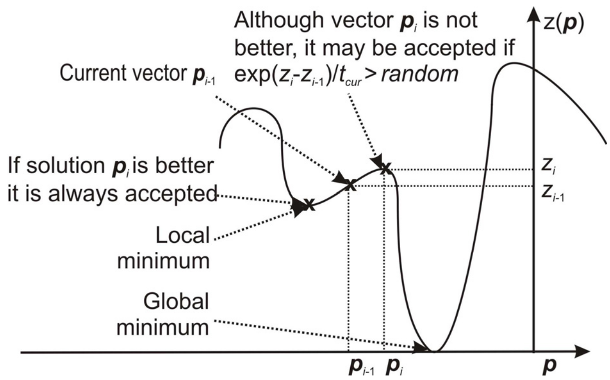

Fig. 2 Principle of finding the global minimum of the residual variable z via the SA method

Fig. 3 shows the pseudo-code of the Simulated Annealing algorithm. This is a symbolic representation of the content of the block called the Optimization Algorithm Subsystem in Fig. 5 if the variant for SA is chosen. The sensor discredibility detection via SA can be described by the following steps:

1. An initial control parameter t_{cur} and a final control parameter t_{final} are set (analogy of initial and final temperature). The control parameter t_{cur} is used to test the stop criterion and to evaluate the Boltzmann criterion (4), which affects the acceptance of the current parameter vector of the sensor model in Step 4.
2. A random parameter vector of the sensor model $\boldsymbol{p}_{i-1}$ is selected and the value of the residual variable z_{i-1} is evaluated.
3. The iteration index i is increased by 1 and, using a stochastic strategy, a new parameter vector of the sensor model is randomly generated and the corresponding value of the residual variable z_i is evaluated.

4. The difference $z_{i-1} - z_i$ between the current residual value z_i and the residual value z_{i-1} from the previous iteration step is evaluated. If $z_{i-1} - z_i > 0$, then the parameter vector is accepted (Fig. 2). If $z_{i-1} - z_i < 0$ the algorithm may accept the parameter vector according to the probability defined by the formula (Boltzmann criterion)

$$P = \mathrm{e}^{-\frac{z_{i-1}-z_i}{t_{cur}}}, \qquad (4)$$

where t_{cur} is a control parameter for the current interaction.

5. The current control parameter t_{cur} is weighted with a coefficient λ, where $0 < \lambda < 1$.
6. If t_{cur} is lower or equal to the final control parameter t_{final}, then the stop criterion is met and the current parameter vector of the sensor model is accepted. Otherwise a return to Step 3 repeats the process.

3 Application of the presented methods

The described methods of computational intelligence were applied to a simple example of a control loop, the Simulink block scheme of which is presented in Fig. 5. As an object where is measured the control variable by a sensor, is a cascade of two tanks. The level in tank 2 is the control (process) variable. Fig. 4 represents the scheme for discredibility detection realization in a block form. The general aim of this control scheme is to enhance the function of a standard PI controller so that the controller will be able to warn the operator about changes in the properties of the control variable sensor.

The relation between the physical quantity measured by the sensor and the value that is outputted by the sensor for further processing, can, in the case of a sensor for level measurement, be assumed to be linear in form

$$h_{est} = kh_2 + q, \qquad (5)$$

where k and q are the sensor model parameters

Initialization()
set initial and final control parameter t_{cur} respectively t_{final}
generate randomly parameter vector of sensor model p_0;
evaluated residual variable z_0 for the initial parameter vector;
set index of iteration $i = 1$;
initialize weighting coefficient λ;
if $t_{cur} > t_{final}$
randomly select parameter vector of the sensor model;
obtain residuum value z_i
evaluate Boltzmann criterion $P = \exp(-(z_{i-1} - z_i) / t_{cur})$
if $z_i < z_{i-1}$ or random $< P$ then
set the quality index equal to the new value $z_{i+1} = z$;
accept the current parameter vector of the sensor model;
else
keep the old quality index $z_{i+1} = z_i$;
keep the old parameter vector of the sensor model;
end
decrease control parameter $t_{cur} = \lambda\, t_{cur}$;
save parameter vector of the sensor model (k, q) into vector p_{i+1};
increment iteration index ;
end

Fig. 3 Pseudo-code of simulated annealing algorithm

according to (3). The value h_2 - the sensor model input, which as a value of a real physical quantity is not available to us, can be determined indirectly from the steady-state characteristics. These express the dependence of the level h_2 on the values of the manipulated variable u (opening of the inlet valve) and the load represented by the values of the steady state flow rate through the tank cascade. It is possible to express this relation by a formula or by a table.

It is clear that the level sensor, as the object whose malfunction should be detected, is not an expensive device. Discredibility detection becomes important in more complex sensors than a level sensor; this application has been used as an example where it is easy to model

the controlled process and to verify the obtained results. Experiments on discredibility detection via both simulated annealing and the genetic algorithm have been carried out.

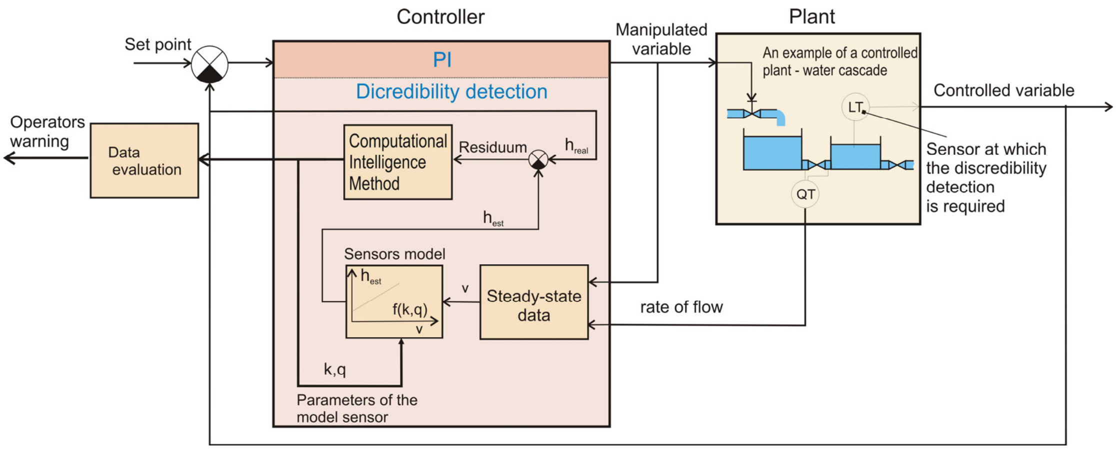

Fig. 4 Extension the controller function for discredibility detection of the control variable sensor

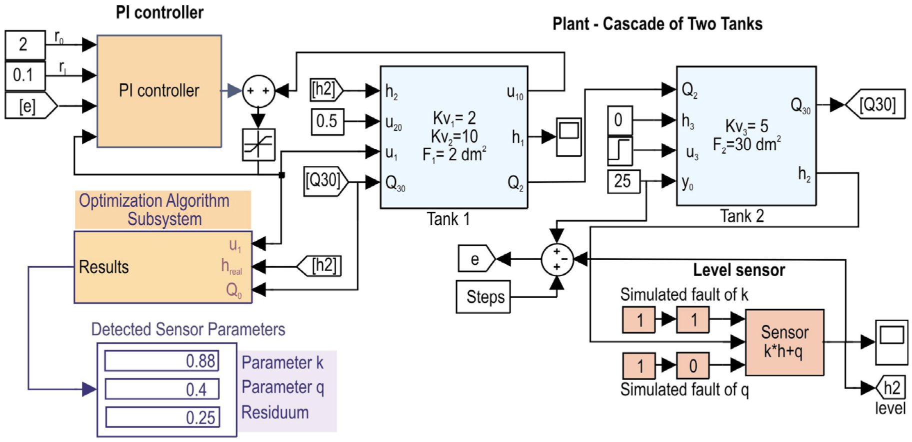

Fig. 5 Simulink block scheme for testing sensor discredibility detection

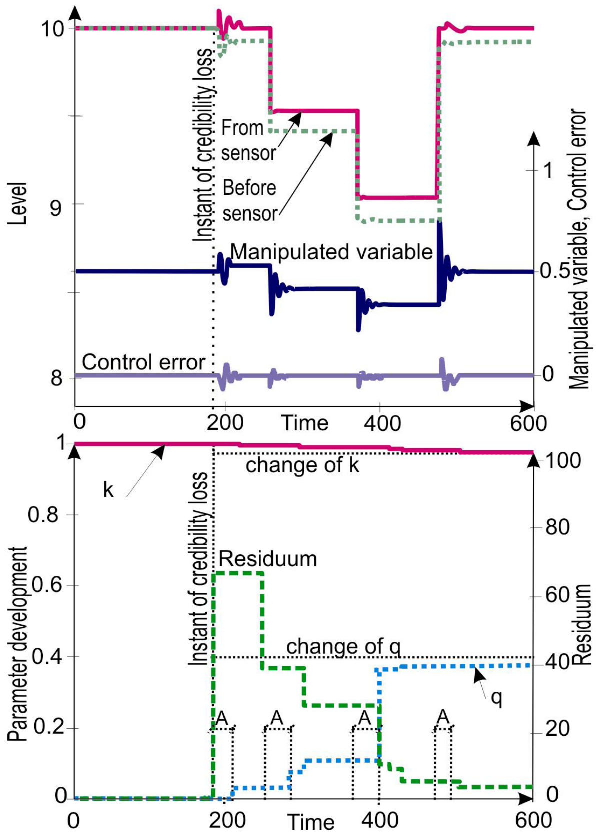

Fig. 6 Impact of the sensor credibility loss on the control loop operation

- parameters before change $k = 1$, $q = 0$
- parameters after change $k = 0.8$, $q = 0.4$
- *A* indicates time intervals with stopped detection due to unsteady state

As it is shown in Fig. 6, until a sensor discredibility occurs, the sensor provides exact data for further processing in the controller. The control loop operates normally. However, after a change in the sensor parameters was simulated, the normal (correct) function of the control loop has been disturbed. As mentioned above, the simulated malfunction should not represent a total failure of the sensor, but only a deviation from the expected features.

In testing the detection of a sensor credibility loss, the block called Optimization Algorithm Subsystem in Fig. 5 attempts to minimize the residual variable which, in this case, is the deviation between the real level value (the controlled variable h_{real}) that is transmitted by the sensor with the changed parameters, and the estimated level h_{est}. In this presented case the estimated level value is obtained from the steady-state characteristic chart based on the measurement of the actuating signal and the volume rate of flow. The steady-state characteristics were obtained with the support of MS Excel via computing the balance of the volume flow rates [4]. If the steady-state characteristics of the controlled system are known, it is not a problem to employ them for finding an estimated water level value. The steady-state characteristic can be computed via balance equations. In a more complex structure, it is necessary to obtain them by a measurement performed while the sensors are providing correct data. The suggested solution is reasonable when the additional experiments do not impose excessively high extra costs in comparison to the cost of the added sensors for hardware redundancy.

3.1 Discredibility detection via the Genetic Algorithm

For the set of parameter vectors of the sensor model a bit-string representation was used. Optimal results were obtained while the parameter vectors of the sensor model were coded into 10bit strings. In our experiment, set of the parameter vectors consisted of 30 pairs of parameters *k, q*, which represents a population of n_p=30 individuals.

For each pair from the set of parameter vectors of the sensor model a residual value was evaluated. Roulette wheel selection was used in the experiment.

The probability of crossover P_{cross} expresses the probability that two selected parameters will actually be recombined. According to [5] the usual value of the probability of crossover P_{cross} is 0.3–0.8. In the considered application the crossover probability was tested and the best results were obtained for probability $P_{cross} = 0.6$.

The mutation parameter expresses the probability of mutation of newly recombined parameters. The influence of the mutation parameter, which expresses the probability of the results, was tested. Its influence on the value of the residual function is shown in Fig. 7. The best results were obtained for probability of mutation $P_{mut} = 0.6$.

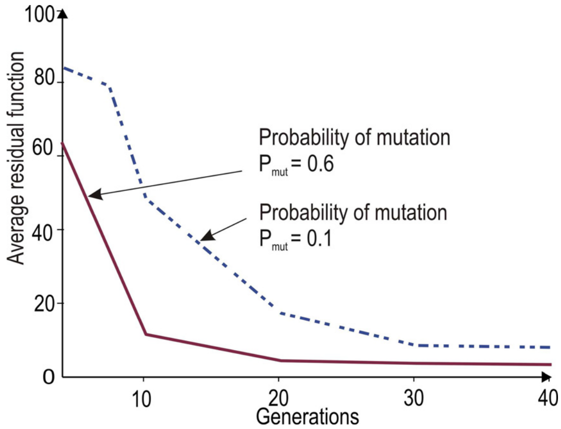

Fig. 7 Influence of the mutation parameter on the average value of the residual function

4 Conclusions

The described software discredibility detection of a sensor via computational intelligence has proved to be a suitable tool for detecting simulated changes of sensor properties. Both the standard genetic algorithm method and simulation annealing methods were described. While testing discredibility detection via the standard genetic algorithm method, the suggested algorithm found new sensor parameters in 40 generations. The time needed for evaluation was several minutes. It is obvious that in a more complex structure it would be necessary to carry out more iterations and this would consume more evaluation time. However, in the case of the application for discredibility detection, this disadvantage does not matter, because small malfunctions do not lead to fatal errors in control loop operation. However, by early detection we can avoid some harmful side effects that may result from sensor malfunction.

Comparing our results with those in [6], which were aimed at detection via simulated annealing, both algorithms proved to be useful utilities. There is no significant difference between the algorithms; their good convergence depends mainly on the algorithm settings.

Our experiments focused on discredibility detection for a single sensor. In more complex devices with more than one controlled loop it will be necessary to design discredibility evaluation from the viewpoint of the function of the whole unit.

Future research will be directed at optimizing the combustion process of a real small stoker-fired boiler for burning biomass to heat up a central heating system. It is planned that an oxygen probe will be applied to support optimization of the combustion process. In this realization the task is to verify whether the probe is providing correct data. This experiment should discover possibilities for probe credibility verification, and it is an important step toward non-simulated applications.

Acknowledgement

This research has been supported by Czech Technical University in Prague grant No. CTU0504612 and by grant No. MSM 680770009 of the Ministry of Education, Youth and Sports of the Czech Republic.

References:

[1] V. Venkatasubramanian, R. Rengaswamy. A Review of Process Fault Detection and Diagnosis. Quantitative Model-based Methods. *Computers & Chemical Engineering.* Vol 27, No 3, pp 293–311, March 2003.

[2] A. Metropolis, M. Rosenbluth. Equation of State Calculations by Fast Computing Machines. *Journal of Chemical Physics*. Vol 21, No 6, pp 1087–1092, May 1999. SSN 0098–1354.

[3] R. King. *Computational Methods in Control Engineering*. Basel, New York, 1999. ISBN 0–8247–1993–8.

[4] B. Šulc, M. Vítečková. *Theory and Practice of Control Loop Design*. CTU in Prague, Czech Republic, 2004. ISBN 80–01–03007–5. (in Czech).

[5] P. Fleming, C. Purshouse. The Matlab Genetic Algorithm Toolbox. *IEE Colloquium on Applied Control Technology Using Matlab*. Sheffield, UK, 1995.

[6] D. Klimánek, B. Šulc. Auto-Detection of Sensors Discredibility in a Control Loop via Evolutionary Algorithms. *Transactions of the VŠB–Technical University of Ostrava,* Vol 49, No 1, Part 2, pp 121–128, VŠB Ostrava, Czech Republic, June 2004. ISSN 1210-0471.

[7] D. Klimánek, B. Šulc. Evolutionary Detection of Sensor Discredibility in Control Loops. *Accepted for the IECON'05, 31st Annual Conference of IEEE on Computer and Control Systems.* November 2005, Raleigh, North Carolina, USA.