Intelligent control of the belt calcinator for calcining zeolite catalysts based on the multidimensional logic controller with interval uncertainty

M. B. Guzairov
Department of computer science and robotics
Ufa state aviation technical university
Ufa, Russia
e-mail: GuzairovMB@mail.ru

E.A. Muraveva
Ufa State Petroleum Technical University
Sterlitamak branch
Sterlitamak, Russia
e-mail: muraveva_ea@mail.ru

Abstract[1]

In the paper it is suggested to resolve problems and process contradiction in zeolite catalysts production using multi-dimensional (in this case a six-dimensional) logical controller with interval uncertainty (MLCwIU) to control temperature in each of the zones of the belt calcinator.

1. Introduction

Calcining zeolite catalysts (CZC) are widely used to improve the following reactions [11]: cracking, hydrocracking, isomerization, alkylation, hydrogenation, dehydration, oxidation. They are thermally stable, resistant to sulfurous and nitrogenous compounds, being noncorrosive with respect to equipment. The developed surface area (up to 800 m^2/g), the cation-exchange capacity, as well as high mechanical strength make it possible to apply them as the catalytical active mass carrier.Most CZC catalysts calcination installations are designed for production of both obsolete and inefficient catalysts brands with a matrix-supported on silica gel and an alumina sol binding agent which are not critical to the temperature condition of the belt calcinator (BC). However, the catalysts brands that provide higher yield of finished products (e.g. light motor fuels) have a cohesive framework pseudoboehmite and aluminum oxychloride. But the technical process of production of CZC is more sensitive to deviations from the optimal calcination condition. It is suggested to resolve the above mentioned process contradiction with the band seven zones belt calcinator (Fig. 1), using multi-dimensional (in this case a six-dimensional) logical controller with interval uncertainty (MLCwIU) [3, 10] to control temperature in each of the zones. The main arguments in favor of this solution are as follows: BC is a classic multiple nonlinear control object that does not have reliable and adequate mathematical model; owing to the MLCwIU the loops interference is compensated more readily and with a less error as compared with multidimensional PID and fuzzy controllers.

2. The logic operation algorithm of multi-dimensional logical controller with interval uncertainty

Fig. 1 shows the BC graphical interface, consisting of seven sections. At the top of the first six of which mounted are electric heaters. For the heated air circulation in order to save energy in the working space of the calcinator there are blow fans provided above heating elements electric heaters. CZC pellets are calcined on the belt L-301 at temperatures of 630-650 °C in the middle sections, 500 °C in the last section, and 150 °C in the cooling zone.

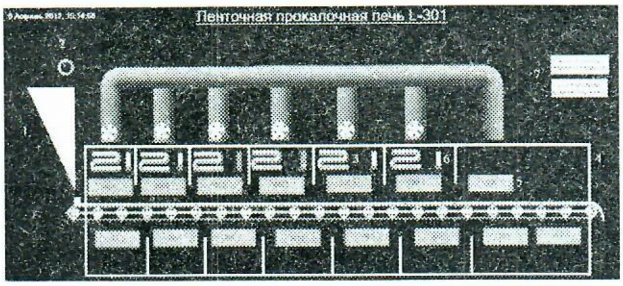


Fig. 1. Belt calcinator graphical interface:

1 – histogram showing the amount of catalyst fed into the calcinator, 2 – the catalyst maximum flow rate alarm, 3 – electric heaters, 4 – belt calcinator L-301; 5 – temperature indicators, 6 – electric heaters temperature controllers, 7 – selection buttons of section's heating either with loops interference compensation or without it

---

[1] *Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage, the CSIT copyright notice and the title of the publication and its date appear, and notice is given that copying is by permission of the Institute for Contemporary Education JMSUICE. To copy otherwise, or to republish, requires a fee and/or special permission from the JMSUICE.*

Proceedings of the 14th international workshop on computer science and information technologies CSIT'2012, Ufa – Hamburg – Norwegian Fjords, 2012

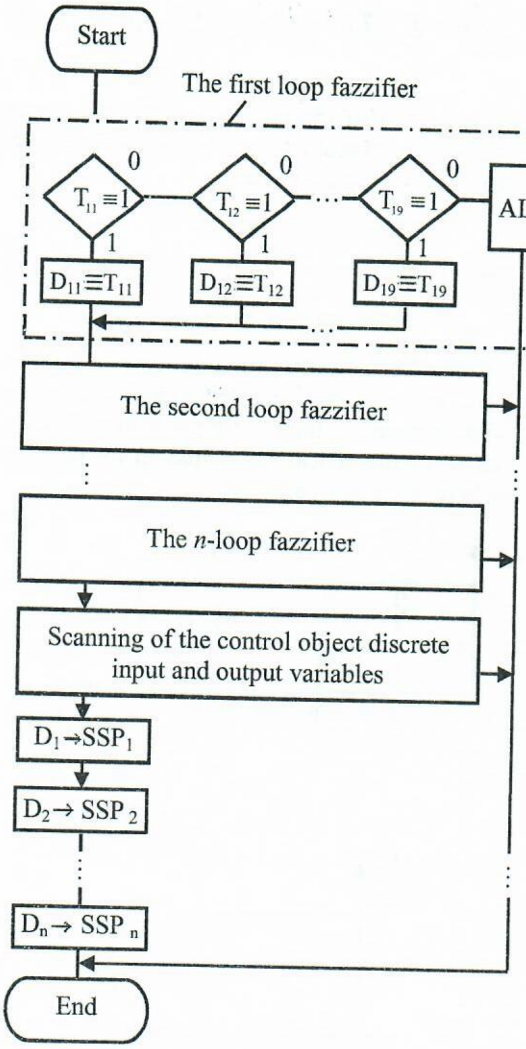


Fig. 2. The logical diagram of the MLCwIU functioning algorithm

Fig. 2 shows a logic operation algorithm of n-dimensional MLCwIU, consisting of the following program blocks: n fazzifiers (one for each MLCwIU control loop); scanning of the control object discrete input and output variables; situational subprograms $((D_1 \to SSP_1) \div (D_n \to SSP_n))$.

The fazzifier of each of the n control loops includes: conditional transfer operators $((T_{i1} \equiv 1) \div (T_{i9} \equiv 1))$, where i is the MLCwIU current control loop; $((D_i \equiv T_{i1}) \div (D_i \equiv T_{i9}))$ – situational subprograms access registers; AL_{if} – operator of emergency situations execution. Without loss of reasoning generality and for the sake of certainty the MLCwIU input and output variables are taken as identified by nine precise terms. Experience has shown [1] that in most cases, increasing the number of terms does not significantly improve the control quality.

Let us consider the algorithm whose scheme is shown in Fig. 1. Scanning cycle of the program that implements MLCwIU begins with the conditional transfer operators execution, comprising MLCwIU fazzifier control loops. For example, if in the first loop precise term T_{11} is equal to a logical unit, then the microprocessor following branch "1" goes to the operator $(D_1 \equiv T_{11})$ and into the register D_1 to record the address of the situational subprogram SSP_1 startup. If the term T_{11} is equal to a logical zero, then control is transferred to the operator $(T_{12} \equiv 1)$, which performs actions similar to that one's of the operator $(T_{11} \equiv 1)$ and etc. similarly up to the operator $(T_{19} \equiv 1)$.

If all the terms $(T_{11} \div T_{19})$ appear to be equal to the logical zero, this indicates fazzification range incorrect setting so that control is transferred to the operator AL_{1f} with the following error message of the MLCwIU operation. In the case that one of the precise terms $(T_{11} \div T_{19})$ is equal to the logical unit the microprocessor proceeds to a similar execution of the second loop fazzifier, and then third and etc. up to the n-loop fazzifier.

Next, the current logic value of discrete input and output variables of the control object is determined (travel sensors, controls, turning on and off of the actuators, etc.). There and then the logical state of the variables that identifies an emergency situation is also verified.

The program scanning cycle implementing MLCwIU, completes by execution of the situational subprogram (SSP) block. If one is to assume that the same number of precise terms m is used to interpret all the MLCwIU controlled variables, the total number of SSPs for the given controller will be equal to $(n \times m)$.

3. Block diagram of the logical controller with interval uncertainty

According to their logical nature, each SSP is a software implementation of a production rule, which must be executed when one of the precise terms of each MLCwIU controlled parameter is equal to the logical unit. From which it follows that the correct assignment of MLCwIU controlled output variables ranges the microprocessor executes n production rules in each scanning cycle, that is one rule for each MLCwIU loop. It should be noted that in the standard multivariate fuzzy logic controllers [1] the entire system is executed, comprising out of tens or even hundreds of production rules in similar situations in each loop.

However, the possibilities of the scheme of the algorithm in Fig. 2 for MLCwIU speed improvement are much wider, due to the fact that in each scanning cycle not the entire system of controller production rules is executed, but only one rule, whose antecedent is currently equal to the logical unit. In this connection for this rule to found a standard access procedure to the subprogram is used, which is now well minimized by CPU time and storage capacity.

On the basis of the MLCwIU execution algorithm the block diagram is developed which consists from four main blocks [10] (Fig. 3),: F – fazzifier, SDIO MCO – scanning of discrete input $(X_1 \div X_s)$ and output $(Y_1 \div Y_k)$

variables of the multiple control object, where s and k – their number respectively; SSPE – situational subprogram execution; MCO – multiple control object with n controlled parameters.

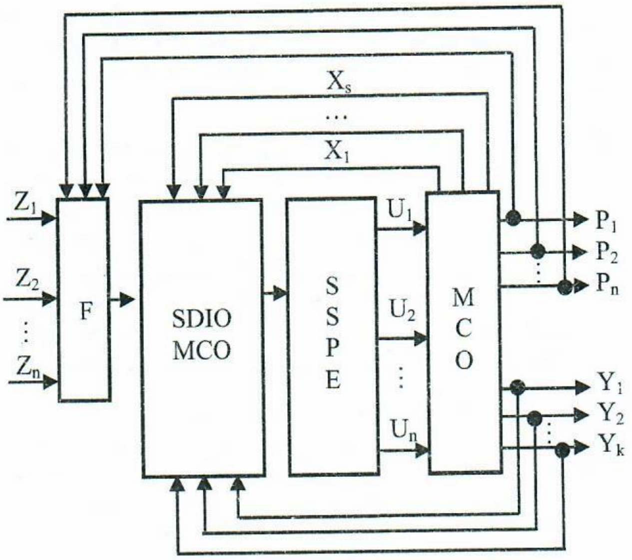


Fig. 3. MLCwIU block diagram

Fazzifier F has inputs of presetting actions ($Z_1 \div Z_n$) and feedbacks ($P_1 \div P_n$). Its output is connected to the SSPE MCO block, to whose inputs discrete input ($X_1 \div X_s$) and output ($Y_1 \div Y_k$) variables of the control object are fed. The SSPE MCO block output is connected to the SSPE block input, whose output signals ($U_1 \div U_{1n}$) in an analog (precise) format is submitted to the actuator of the multiple control object.

4. The theoretical basis for speed increasing and reducing the MLCwIU error

Interpretation of the i controlled (P_i) and presetting (Z_i) variables by the set of m precise terms is shown in Fig. 4, a. From this it follows that at any given instant only one term is equal to the logical unit, and exactly the one inside which there is currently P_i and Z_i precise value which is in keeping to common sense. In its turn, it is owing to this that only one rule has the antecedent which is equal to the logical unit in the production rule system, operating with precise terms, at any given instant. It is precisely these fundamental properties of the above mentioned precise terms set and the production rule system which are the theoretical basis for speed increasing and reducing the MLCwIU error.

Analytically, the basic term-set, as shown in Fig. 4, a can be represented by the following expression:

$$T(p)=\{T_1(0 \le p < l),\ T_2(l \le p < 2l),\ T_3(2l \le p < 3l),\ \ldots,\ T_i((i-1)l \le p < il),\ T_m((m-1)l \le p < ml)\}, \tag{1}$$

where l – a precise term width.

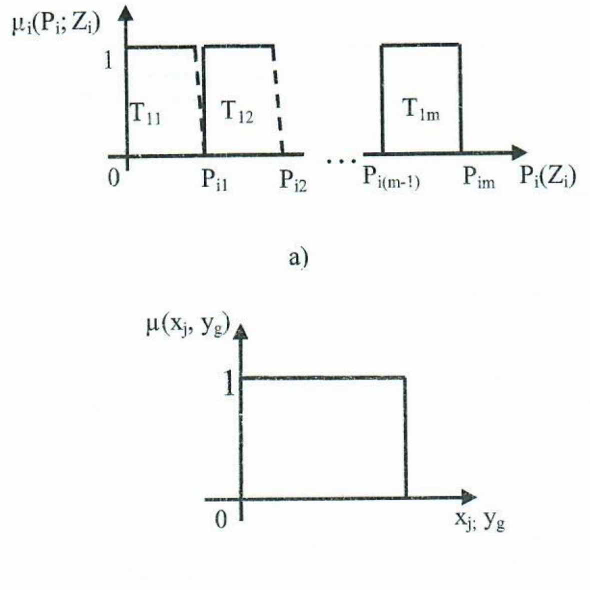
However, in practice the expression (1) is more appropriate to use in the following form:

$$T(p) = \textstyle\sum_{i=1}^{m} T_i((i\text{-}1)l \le p \le il) = \sum_{i=1}^{m} T_i(il) \tag{2}$$

Membership function of discrete input (X_j) and output (Y_g) variables of the control object is presented in Fig. 4, b. And it can take two logical values:

$$\mu(x_j \text{ or } y_g) = \begin{cases} 1, \text{ if } x_j \text{ or } y_g \text{ turn on;} \\ \\ 0, \text{ if } x_j \text{ or } y_g \text{ turn off.} \end{cases} \tag{3}$$

From Fig. 4, a and b, as well as formulas (1-3) it follows that the precise terms and discrete signals have the common logical nature – they are arguments of binary logic, which is the theoretical substantiation for them to be joint use in production rules.



a)

б)

Fig. 4. Membership functions of the precise terms (a) and discrete variables (b) of the multiple control object

Let us make a quantitative estimate of the scanning time reduction of the program that implements MLCwIU, in comparison with a typical fuzzy controller. The time required to complete one fazzifier scanning cycle of the MLCwIU being suggested is equal to

$$T_{fn} = \sum_{i=1}^{m} (m_{tif}) t^{f}_{ci} + t^{f}_{ei}), \tag{4}$$

where m_{tif} – the number of productions, executed in the current scanning cycle of the i fazzifier of the n-dimensional precise logic controller; t^{f}_{ci}, t^{f}_{ei} – the execution duration of conditional and final parts of the production rule of i fazzifier, respectively (that t^{f}_{ci} and t^{f}_{ei} are presumably constant values).

For the vast majority of programmable controllers [1], on which MLCwIU is implemented $t^{f}_{ci}=t^{f}_{ei}=$ const, whereas

$m_{tif} = 0{,}2 \cdot m_i$. Here, m_i is the production rules number in the MLCwIU i fazzifier (i=1÷n). Given the above conditions at $m_i = m =$ Const for all MLCwIU fazzifiers the expression (4) looks like:

$$T_{ssp}=\sum_{j=1}^{n}(t_{ei}^{ssp}+t_{ej}^{ssp}), \qquad (5)$$

Under the same conditions as for the expression (4) the scanning cycle duration of the situational subprogram system (SSPS) is determined by the following formula

$$T_{ssp}=\sum_{j=1}^{n}(t_{ci}^{ssp}+t_{ej}^{ssp}) \qquad (6)$$

where t_{cj}^{ssp} and t_{ej}^{ssp} are the execution duration of conditional and final parts of a production rule of the j situational subprogram, respectively (both t_{cj}^{ssp} and t_{ej}^{ssp} are presumably constant).

For the vast majority of programmable controllers [1, 4], on which the MLCwIU is implemented $t_{cj}^{ssp}=2t_{ej}^{ssp}=$ const, so the expression (6) can be reduced to the following form:

$$T_{ssp}=1{,}5nt_{cj}^{ssp} \qquad (7)$$

The total scanning time of the fazzifier and the SSPS of the MLCwIU being suggested is equal to the sum of the expressions (5) and (7) right-hand sides:

$$T_{sn}=T_{fn}+T_{ssp}=n(0{,}2m+2{,}5)t_{cn}, \qquad (8)$$

where $t_{cn}=t_{ci}^{f}=t_{cj}^{ssp}=const.$

Similarly, under the same conditions the expression for the scanning duration of the fazzifiers and production rules systems in a typical multi-dimensional fuzzy logic controller is as follows:

$$T_{sm}=4nmt_{cm}, \qquad (9)$$

where the t_{cm} is the scanning duration of the production rule conditional part of the typical fuzzy logic controller.

Let us define how low has the scanning duration decreased in the MLCwIU being suggested as compared to the typical multi-dimensional fuzzy controller, in which as is known [1], in each scanning cycle the entire program, which implements the fazzification and controlling is executed completely. For this purpose on condition $t_{up} = t_{ut}$ let us divide right-hand side of the expression (9) by the counterpart of expression (8):

$$K_{si}=4m/(0{,}2m+2{,}5), \qquad (10)$$

where the K_{si} is the coefficient of the speed increase of the MLCwIU being suggested.

5. Conclusions

As follows from expression (10) K_{si} does not depend on n and, for example, when $m = 9$ MLCwIU speed in comparison to the standard multi-dimensional fuzzy logic controller is increased 8,37 times.

References

1. Gostev V.I. Design of fuzzy controllers for automatic control systems. – St. Petersburg.: BHV-Petersburg, 2011 – 411 p. Ill.
2. Muravyova E.A. Discrete-logic controller with any-time algorithm to minimize the response time. // Electrical engineering, electric drive and companies electric equipment. Ufa: Gillem, 2007. S. 63-65.
3. Muravyova E.A., Antipin A.F. Multidimensional discrete-logic controller of air flow from the steam boiler using minimization of the response time // Vestnik USATU. Series "The Control, Computer Mashines and Computer Sciences, 2009, T.13, № 2 (35). P.83-87.
4. Kruglov V.V., Dli M.I. Intelligent information systems: computer support of fuzzy logic and fuzzy inference systems. Moscow: Fizmatlit, 2002.
5. Leonenkov A.V. Fuzzy Simulation in MATLAB and Fuzzy TECH.-SPb.: BHV-Petersburg, 2005. – 736p.
6. Muravyova E.A., Baimanov I.F., Kayasheva G.A. The fuzzy system of pH value control in electrolyte in chlorine manufacture via the electrolysis route // Automation, telemetry and communication in oil industry. – 2008. – №1. – p. 10-13.
7. Verevkin A.P., Kiryushin O.V. Automation of technological processes and industries in oil refinary and petrochemical industries. – Ufa: publishing office UGNTU, 2005. – 171 p.
8. Vasilyev V.I, Ilyasov B.G. Intelligent control systems. Theory and Practice: Textbook. – M.: Radiotechnika, 2009. - 392.
9. Demenkov N.P. Fuzzy control in engineering systems: .Teaching guide.-M.: MGTU named afterN.E. Bauman, 2005. – 200 p.
10. Muravyov E.A., Kayashev A.I., Gabitov R.F. The precise logic controller for process control. RF patent № 2445669. Published on: 3/20/2012. Bulletin. Number 8.
11. Khadzhiev S.N. Cracking of petroleum fractions on zeolite catalysts. - M.: Chemistry. 1982. – 277 p.