Journal of King Saud University, ***17***, Eng. Sci. *(1), 25-45, 2004.*

**Fuzzy Control for the Start-up of a Non-isothermal CSTR**

Emad M. Ali* and Abu Khalaf, A. M.
Chemical Engineering Department
King Saud University
POBox 800, Riyadh 11421
Saudi Arabia
Fax: ++(9661)467-8770, Phone: ++(9661)467-6871
Amkamal@ksu.edu.sa

**Abstarct**

This paper addresses the control problem of starting up a non-isothermal CSTR. The startup problem is of dual objectives. One objective is to fill up the tank and the other is to achieve certain reaction conversion. The two objectives compete with each other because each has a different gain sign. Consequently, standard PI algorithms may not perform well. In fact, nonlinear control algorithms can be tested for the startup control problem. However, to avoid computational complexity brought in by such nonlinear controllers, Fuzzy Logic Control (FLC) can be a simple and suitable alternative. Specifically, FLC algorithm is applied for the start-up problem and its performance is compared to that of PI controller with gain scheduling.

**Introduction**

Chemical processes show dynamic behavior during start-up, shutdown, and when upsets occur under steady state conditions. Mathematical modeling, simulation, and control of these processes are relatively difficult, because of the nonlinear nature of these processes and the activation and tuning difficulties of the controllers. Most of the reported control problems in the literature deal with disturbance rejection or changing set point. Only few efforts were reported dealing with the automatic start-up of continuous chemical processes from zero initial condition. Automatic start-up is similar in concept to finding the optimal input profile for batch processes. The start-up problem considered here introduces an integrator state, which makes the control problem more challenging. Moreover, the control problem has one degree of freedom, therefore, the control design must handle all objectives or ignore some of them and suffer from the consequences.

Standard proportional plus integral (PI) control to start-up a non-isothermal CSTR was investigated[1]. Simple and straightforward schemes of activation were tried, in an attempt to achieve maximum possible conversion and to attain smooth operation during start-up[1]. The importance of this control problem lies in the difficulty of triggering the controller and the re-tuning of the PI settings during start-up from zero initial condition. The startup operation consists of filling the reactor starting from an empty condition to a maximum level. During the startup operation it is also required that the chemical reaction reaches maximum conversion manifested by the concentration of the product. Therefore, the control objective for the CSTR is dual. In this case, the feed flow, which is the only available manipulated variable, needs to be regulated to achieve both objectives. The first control objective, i.e. reactor filling, requires a positive controller gain, while the second control objective, i.e. reaction conversion, requires a negative controller gain. The difficulty thus relies on the changing sign of the process gain for the same manipulated variable.

Most of the research work in the literature deals only with the control problems of disturbance rejection and set point tracking. However, the control problem of starting up a process from zero initial condition is a difficult task. Due to the non-linearity of the process behavior during start up, linear PI controllers were found inadequate without the aforementioned strategies. Therefore, the main objective of the paper is to investigate application of nonlinear control algorithms for the start-up problem. Specifically, Fuzzy Logic Control[2,3] (FLC) will be examined in this paper. This type of control algorithm is chosen because it retains some of the characteristics of the PI controllers. This is manifested in the sense of independence of FL control law on the model equations. Moreover, FLC is suitable for controlling processes requiring human knowledge. For example, the knowledge of the process changing gain is required to perfectly start-up the reactor.

FLC is based on the original work of Zadeh[4] on fuzzy set theory. Its first implementation to control physical processes was proposed by Mamdani [5,6]. Since then, several other applications were reported[7-12]. Recently, FLC received more interest due to its successful application to important industrial systems[13]. Since FLC can adapt itself to changing situations, it can outperform conventional PI controllers for unstable dynamics and nonlinear systems. In contrast to model-based controllers, FLC is known as knowledge based controllers, that does not require a mathematical model of the process at any stage of the controller design and implementation. In many cases, the phenomenological model of the control process may not exist or may be too expensive in terms of computer processing power and memory, and a system based on rules of human knowledge may be more effective. In this case, FLC is a simple alternative to model-based advanced controllers.

However, there are many factors that limit the spread of such controller such as a lack of understanding of the technology, limited application experience[14]. Many Engineers may not understand the mathematics involved. Moreover, its tuning requires adjustment of many parameters.

The objective of this paper is to investigate the feasibility of applying FLC to the control problem of automatic start-up of a non-Isothermal CSTR. Comparison of the FLC with that of conventional gain-scheduling algorithm will be investigated. Ease of implementation and practicality will be addressed as the basis for evaluation. The start-up control problem is particularly suitable for FLC implementation. The latter is known to be useful where specific knowledge about the process behavior is available. For example, the nonlinearity of the process arising from the gain sign change can thus be transformed into fuzzy sets. The latter can be easily incorporated in the fuzzy logic control law. Therefore, incorporating the known nonlinearity into the FLC framework is the main contribution of this work.

The paper is organized as follows. Next section covers the process description and its mathematical model. The following section presents the design procedure of FLC algorithm. Then, simulation tests will be shown in a following section to investigate the performance of the proposed control algorithms. The final section outlines the concluding remarks.

**Non-Isothermal CSTR Model**

A liquid phase chemical reaction of the form of: A+B → C+D with known kinetics[2,3] is taking place in a non-isothermal CSTR as shown in figure 1. The CSTR is equipped with an overflow and a total feed (pure A and pure B). The two feeds are equi-molar at 0.1M. The reactor represents an existing lab-scale process at the departmental laboratory. Various design parameters are listed in Table 1. This process is chosen as a test example for its varying dynamics character as it operates in two different stages. The first stage is the start-up where the process operates in a semi-batch mode during which the liquid holdup behaves like an integrator system. The second stage is the steady state where the process operates in a continuous mode during which the liquid holdup is constant at its maximum. The complete dynamic model for the process is given as follows[1]:

$$\frac{dV}{dt} = F_1 + F_2 - F \quad (1)$$

$$\frac{dVC_A}{dt} = F_1 C_{Af} - FC_A - Vk_r C_A^2$$
$$\frac{dVC_c}{dt} = -FC_c + Vk_r C_A^2 \quad (2)$$

$$\frac{d}{dt} V(\sum_{i=1}^{4} C_i Cp_i)T = (F_1 C_{Af} Cp_A + F_2 C_{Bf} Cp_B)(T_f - T_{ref}) - F(\sum_{i=1}^{4} C_i Cp_i)T - Vk_r C_A^2 \Delta H - Q \quad (3)$$

Where:

$$k_r = k_0 \exp(-E/RT)$$

$$Q = A_s h_{air} (T - T_{amb})$$

$A_s$ denotes the reactor surface area. The material balances for component *B* and *D* are omitted since they are identical to those for component *A* and *C* respectively. Equations 1-3 simulate the CSTR dynamics during the start-up stage. The original model[15,16], which is validated against the lab data, consists of three sets of model equations. Each set corresponds to a specific stage of the process startup operation.

For example, the first stage is filling up, the second is approaching steady state and the third is operating steadily. In this paper, the process model equations are lumped as given by Eqs.1-3 for simplicity. However, to simulate the dynamics of the first stage, the outlet flow, $F$, is set to zero and therefore, the reactor holdup varies with time. In the second and third stages when the holdup reaches its maximum value, the outlet flow is set equal to the sum of feed flow rates.

Due to this interesting dynamics, the process gain and time constant change substantially with operating conditions as shown in figure 2. The figure demonstrates the open-loop response of the product concentration for two different step changes in the inlet flows starting from zero initial conditions. The corresponding steady state operating conditions of the process are listed in Table 2. Note that the reaction temperature at the zero initial steady state is taken as the room temperature. As figure 2 shows, at large step change in the feed flow the product concentration reaches a small value, while at a smaller step change in the feed flow rate, the product concentration reaches a higher value. This means, the product concentration decreases with increasing the *start-up* flow rate. Therefore, the static gain relating the product concentration with feed flow rate is negative. On the other hand, the static gain relating the reactor holdup to the feed flow rate is positive, which is obvious. The phenomena of varying process gain with operating conditions and the fact that the controller has dual objectives with different gain signs during startup are the main motives to implement and test the fuzzy control algorithm. The process static gain and time constant at different operating conditions are listed in Table 2. These process parameters were estimated by reaction curve tests except those at the initial condition. Since the process responds as a ramp function to step changes induced at the zero initial condition, reaction curve method can not be used to identify the process parameters. Instead, the process gain is identified by impulse test[1]. Alternatively, the process gain can be calculated from a linearized version of the model equations 1-3. The model equations will be used for simulation purposes and not in the design of the FLC algorithm.

## Fuzzy Logic Control Algorithm

The basic FLC loop is shown in Figure 3. It consists of three major sequential steps, namely Fuzzification, Inference engine and Defuzzification. Fuzzification transforms a crisp value (real-value) into a member of fuzzy sets, while defuzzification transforms the fuzzy output determined by the inference engine into a crisp value. The inference engine is the decision-making engine (control law). In the following, the development and design of each step is discussed in detail. Hereafter, by input we mean controller input, i.e. error and/or error velocity signal and by output we mean the controller output, i.e. manipulated variable.

Fuzzification:

The input signal of the controller, which is a real-value variable also known as crisp value is fed to the fuzzifier. In the fuzzifier, the crisp value is converted as a member of a certain fuzzy set. The fuzzy set is usually represented as a membership function as shown in figure 4. The membership function can have any symmetrical geometric shape such as triangles, trapezoidal, Π function Gaussian function, etc. The fuzzy set can also by represented by non-symmetric shapes. Regardless of its shape, the membership function is graded between 0 and 1. Usually, finite number of overlapping membership functions (fuzzy sets) can be used to span the possible range of the process variable. The overall span (domain of a specific variable) is known as the universe of discourse. To identify these sets, each fuzzy set is given a linguistic name such as Large Positive (LP), Small Positive (SP), Zero (ZE), Small Negative (SN), Large Negative (LN), etc.

Therefore, the process of fuzzification is simply checking the value of the input signal (member) against each fuzzy set to determine its degree of membership (belongingness). The member can have a membership value of zero (non member), a value of one (full member), or an intermediate value (partial members). An input signal can be a member of more than one membership function. Therefore, when the member has a membership value other than zero, the corresponding fuzzy set is considered fired or triggered.

Common difficulties exist in this step. The selection of the shape and number of the membership functions, the location of their center, i.e. where fuzzy set has a maximum value, and the size of the universe of discourse are not clear. Moreover, the common FLC design involves at least three different groups of fuzzy sets, each of which corresponds, to a different process variable. For example one group is used for the error signal, *e*, another for the velocity of error signal, Δ*e*, and another for the controller's output (manipulated variable), *u*. The latter is used in the defuzzification step. In this paper we try to overcome the above problems. First we use only one group of fuzzy sets for all the three process variables. To achieve this, the universe of discourse is unified so that it spans the interval [-1,1]. In this case, the value of each process variable should be scaled properly to fit the specific interval. Discussion of the scaling issue is given under the tuning section. Furthermore, gaussian and sigmoidal shapes are considered for the membership functions. Five such functions are used with the locations of their centers is as shown in Figure 5. Gaussian shape is selected because it is continuous function and can be easily coded in a digital computer. The number of fuzzy sets is chosen arbitrary, however increasing it will increase the number of control rules at the benefit of little improvement. The relative location of their center will be adjusted automatically using our proposed tuning method as discussed later.

Specifically, the fuzzy sets shown in figure 5 are assigned the following mathematical functions:

$$\mu_{LP}(x) = \frac{1}{1+e^{-20(x-0.75)}} \equiv \mu_1 \qquad (4)$$

$$\mu_{SP}(x) = \mathbf{exp}(-20(x-0.5)^2) \equiv \mu_2 \qquad (5)$$

$$\mu_{ZE}(x) = \mathbf{exp}(-20x^2) \equiv \mu_3 \qquad (6)$$

$$\mu_{SN}(x) = \mathbf{exp}(-20(x+0.5)^2) \equiv \mu_4 \qquad (7)$$

$$\mu_{LN}(x) = \frac{1}{1+e^{20(x+0.75)}} \equiv \mu_5 \qquad (8)$$

Where *x* is the process variable. Note that these functions are unified for all process variables used in the work, namely the error (*e*), the error velocity (Δ*e*) and the manipulated variable velocity (Δ*u*). Therefore, in this controller phase, the membership degree of a specific input value, i.e. *e* or Δ*e*, for all fuzzy sets can be numerically computed via direct substitution in equations 4-8.

*Inference Engine*:

Inference engine is the heart of the FLC algorithm where the control action is formulated. Specifically, it describes the output of the controller for all input signals combination. It consists of several fuzzy set rules represented by conditional statement in the form of IF-Then rules as shown in Table 3[17]. The collection of all rules is called Rule Base. Generation of such rules is the difficult part of the FLC design.

In general, deriving the rule base can be approached by:

- Empirical knowledge of a skilled human operator
- Desired response of the process
- Mimic a conventional PI controller in the velocity mode.

In this paper, we choose to design the rule base according to desired response of the process because it the most intuitive for many control practitioners. The description and reasoning of each rule is explained in Table 3[17], which basically describe a generic feedback response. Note that the AND command is a common fuzzy rule operation, which mathematically implies[17]:

$$\mu_A(x)\ \text{AND}\ \mu_B(x) = \mathbf{min}(\mu_A(x), \mu_B(x)) \qquad 9$$

At this phase of the controller algorithm, given a value for the input signal, the degree of fulfillment of each rule in the rule base set is determined. The rule base in Table 3 is given in fuzzy logic terminology. Using equation 9, the degree of fulfillment of the base rules can be computed mathematically as shown in the fourth column in Table 3. Note that the first index of μ is the label for the membership function as in equations 4-8. The second index indicates the rule number. The degree of fulfillment of the rule base is known as the conclusion or the result of the rule base. The process in which these conclusions are calculated is known as inference. Due to overlapping membership functions, some of the rule conclusions may have a zero value and some a non-zero value. Membership function for the output with a non-zero degree of fulfillment is considered fired. In standard FLC algorithms, the fired functions are clipped or scaled and then copied to a temporary template. All fired sets are then combined using superimposing technique[17]. The combined set is known as the inferred controller output. For example, if three membership sets were fired, clipped and combined, then an inferred output with new geometrical shape is obtained as shown in Figure 6. The inferred output (new geometrical shape) is then converted into a crisp value using the defuzzifier. The calculated crisp value is the numerical value for the manipulated variable. In this paper, the process of clipping, copying and combining is overlooked. Alternatively it is replaced by direct numerical method as discussed in the next section.

It should be noted that the rule base in Table 3 is a generic one that works for most positive feedback loops. One can further modify or can add more rules to capture certain known behavior of the process such as non-linearity or constraints. For example, the results of the rules are reversed when implemented. The reason for reversing is that the process has negative gain. After reversing the rule results, the process can not be started up from zero initial condition since the process requires a positive gain at that point. For this reason, the following rules are added:

R26: If $e$ is LP and $V$ is ZE then $\Delta u$ is LP

R27: If $e$ is SP and $V$ is ZE then $\Delta u$ is LP

R28: If $\Delta e$ is LP and $V$ is ZE then $\Delta u$ is LP

Here $V$ is the liquid holdup. Note that a single membership function, i.e. ZE, is used to fuzzify the process variable $V$. The ZE fuzzy set for $V$ is designed to span the interval [0,0.01]. The latter region means that a value for $V$ in that region indicates empty tank conditions. Note that it is necessary for start-up to use these three additional rules so that at the beginning of the simulation the total fired positive rules outweigh the total fired negative rules. When the holdup value comes out of the ZE region, the negative rules start outweighing the positive rules producing lower feed flow, but leading to the desired product concentration.

Defuzzification:

In this step, the combined output fuzzy sets are then converted into a single crisp value. Usually it is equivalent to finding the weighted average value for the combined sets. In standard FLC applications, the combined set is a new geometric shape, say $\mu_{out}$. Hence, finding a weighted average is similar to determining the geometric center. One way is by calculating the center of area (COA)[17]:

$$u = \frac{\sum_{i=1}^{N} u_i \mu_{out}(u_i)}{\sum_{i=1}^{N} \mu_{out}(u_i)} \tag{10}$$

The summation is carried out over discrete values for the universe of discourse $u_i$ for the specific fuzzy set $\mu_{out}$ sampled at $N$ points. Another way is to calculate the mean of the maxima (MOM)[17]

$$u = \sum_{m=1}^{M} \frac{u_m}{M} \tag{11}$$

where $u_m$ is the $m$th element in the universe of discourse at which the membership function $\mu_{out}$ has a maximum value. $M$ is the total number of such elements.

COA or MOM as given above is implemented on the output membership function $\mu_{out}$. Thus, the output membership function should be inferred first, i.e. carrying out the clipping, copying and combining procedure. In this paper, we alternatively calculate the weighted average numerically using the results of Table 3:

$$u_k = \frac{\sum_{j=1}^{n_R} \sum_{i=1}^{n_f} \mu_{j,i}(e, \Delta e) \delta_i A_{j,i}}{\sum_{j=1}^{n_R} \sum_{i=1}^{n_f} \mu_{j,i}(e, \Delta e) A_{j,i}} \tag{12}$$

Where $n_R$ is the number of rules and equals 26 in this paper, $n_f$ is the number of membership functions and equals 5 in this paper, $\delta_i$ is value for the location of the center of $\mu_i$. The value of $\delta_i$ is pre-calculated and fixed as shown in figure 5. $A$ is $n_R$ x $n_f$ pre-calculated matrix, which identifies which membership function is included in each Rule. For example, row 1 of matrix *A,* which is assigned for Rule 1, contains 1 at the first column and zeros elsewhere. The same logic is carried out over the remaining rows. The formula given by equation 12 is a combination of the COA and MOM formulas and is used in here to avoid the process of aggregating the fired fuzzy sets.

It should be emphasized that the control output, $u$ computed by equation 12 at certain sampling time $k$ is taken to be in the velocity form. Velocity form is more suitable for non-linear systems as discussed next. In non-linear systems, the new equilibrium value for $u_{ss}$ that brings the output to the desired steady state value may not be known beforehand. Thus, it is difficult to locate $u_{ss}$ in the universe of discourse as the center for the ZE membership function. However, when $\Delta u$ is used, zero value will always be the equilibrium point around which ZE can be built.

*Tuning method:*

Tuning a fuzzy linguistic controller to changing process and environment dynamics can be accomplished in several different ways:

- Adjusting the membership functions.
- Changing the finite set of values describing the universe of discourse.
- Reformulating the finite set of control rules in the knowledge base (inference engine).

However, these procedures are cumbersome. In addition, there are no clear guidelines on how these procedures affect the closed-loop response. In this paper, we adopt a simpler method. The scaling factors for the input and output signals are used as the tuning parameters. As will be seen in the examples, these factors have direct and clear effect of the closed-loop response. These factors are used to scale the process variables so that they fit the universe of discourse domain used in figure 5. Specifically, the scaling factors for the error, error velocity, and output velocity are $se = a/sp$, $sde = b/sp$ and $sdu = c\Delta u_m$ respectively. For servo control problems, $sp$ is the difference between the set point and the steady state value for the controlled variable. For regulatory control problem, $sp$ is the set point value. $\Delta u_m$ is the difference between the maximum and minimum allowable values for the manipulated variable. Therefore, *a*, *b*, and *c* are the tuning parameters. Therefore, changing the value for *a*, *b*, or *c* is equivalent to stretching or expanding the universe of discourse of the fuzzy sets shown in figure 5. Conceptually, this is similar to the first two tuning guidelines mentioned above. This idea of normalizing the fuzzy set domain is initially proposed by Qin and Borders[18]**.**

FLC algorithm:

The following steps explain the FLC control algorithm used in this paper.

Set $a=b=c=1$.At any sampling time, *k* do:

**Step1**: scale the error and the error velocity signals ($e$ ($k$), $\Delta e$ ($k$)) via multiplying them with *se* and *sde*.

**Step2**: Compute the degree of membership of $e(k)$ and $\Delta e(k)$ to the five membership functions using equations 4-8.

**Step3**: Calculate the conclusions of the Rule base as given in Table 3.

**Step4**: Calculate the control action using equation 12. Scale the computed value by multiplying with *sdu*.

**Step 5**: Implement the control action, set $k=k+1$ and go back to step 1

If the control performance is poor, adjust the value of *a*, *b*, or *c*. We have found that increasing the value of *a* increases the speed of response and eliminates offset. Increasing the value of *c* penalizes the manipulated variable moves, thus produces sluggish response. The parameter *b* has almost similar effect as *a* has, but with less magnitude. However, tuning *b* should be avoided because large values for it make the controller sensitive to steady state noise (numerical error in case of simulations).

The FLC implemented here has a single control loop with dual objectives. By dual objectives we mean two controlled variables connected to one manipulated variable. The controlled variables are the product concentration and the reactor volume. Conceptually, this dual objective procedure is similar to split range control scheme.

**Gain scheduling scheme**

Adapting the controller gain with a changing process key (auxiliary) variable is known in general as gain scheduling (GS). The conventional way of gain scheduling is switching between different sets of local linear controllers, each of which is designed for a specific region of the process operating condition space. Gain scheduling can have different formulations, one of which can be given as an interpolation of two given extreme values for the controller gain[19]:

$$k_c(\mathrm{t}) = (1- q)\, k_c^u + q\, k_c^l \tag{13}$$

where $k_c^u$ and $k_c^l$ are the upper and lower values for the controller gain respectively and $q$ is an interpolation parameter, which can be given as:

$$q = exp\,(-\kappa\, |e(t)|\,)$$

where κ is a design parameter. This interpolation criterion is used to provide smooth transition between two different values for the gain. In the present work, the extreme values for $k_c$ are determined by their magnitude irrespective to their signs. If the process model is available or its gain can be estimated on-line, a programmable or model gain scheduling (MGS) can be formulated as follows[19]:

$$k_c(t) = \frac{k_{co} k_{po}}{k_p(t)} \tag{14}$$

Here $k_{co}$ and $k_{po}$ are the reference values for the controller and process gains respectively. The idea is to keep the overall gain of the closed-loop system, i.e. the product $k_{co}$ $k_{po}$, constant. It is necessary to keep the overall gain less than unity, particularly at the crossover frequency, to ensure stability[20]. The gain scheduling technique is repeated here for comparison purposes. Note that the gain scheduling method will be implemented on the standard PI controller. The GS function is thus to update the PI controller gain, $kc$, online according to Eq. 13 or 14. Unlike the FLC algorithm, the gain-scheduling method has a single controlled output, which is the product concentration, $C_c$. The other output, i.e. reactor holdup, will be handled implicitly through gain scheduling. For example, the controller gain obtained from either Eq. 13 or 14 will be positive initially, which will allow reactor filling. The controller gain will then change sign as the reaction proceeds.

**Simulation Results**

The control objective of the process is to fill up the reactor and the same time to bring the controlled variable $C_c$ at the desired value of 0.0326 mole/l. This will be approached during start-up, i.e., starting the reaction from zero initial conditions, and during disturbance rejection. The manipulated variable in this case is the inlet flow rate $F_1$. The second inlet flow, $F_2$ will be maintained through ratio control of obvious ratio of $F_1$:$F_2$ = 1:1[1]. Both $F_1$ and $F_2$ are constrained between 0 and 1 *l/min*. Therefore, in this paper $\Delta u_m$ =1 *l/min*. A sampling time of 1 *min* is used in the simulations hereafter.

The closed-loop response for set point change for $C_c$ from 0.0 to 0.0326 mole/l is shown in figure 7. The figure compares the performance of GS, MGS and FLC algorithms. For GS method, $\kappa$=50, $\tau_I$ = 2 min and $50 \leq k_c \leq -120$ are used. The PI parameters are determined previously[1]. The value of $\kappa$ is determined via trial and error. It is found that $\kappa$ has to be at least equal 50 for the PI algorithm with GS to work well. For MGS method, $k_{co}$ = 50, $k_{po}$ = 0.011 and $\tau_I$ = 2 min are used. Note that the PI settings are determined previously[1], while the value for $k_{po}$ is taken from Table 2 at zero initial condition. The value of $k_p$ is computed online using a predefine correlation[1]. In that correlation, $k_p$ is made a function of $F_1$. For FLC method, $a$=$b$=1 and $c$ =0.5 are used. These values were sufficient to provide acceptable feedback performance. As figure 7 demonstrated, all the control algorithms managed to start-up the reactor from zero initial conditions. Apparently, the GS and MGS managed to fill the tank readily. The gain-scheduling methods managed to fill up the reactor rapidly because they possess relatively large positive controller gain at the beginning of the startup point. This positive gain resulted in plentiful feed flow rate. On the other hand, the FLC has two competing objectives, one with positive gain and the other one with negative gain. This situation created smaller positive action manifested by modest feed flow rate and consequently slower filling of the tank. However, as far as the primary controlled variable is concerned, the three methods demonstrated somewhat similar performance. In fact, the FLC delivered relatively faster response with minor overshoot. Figure 7 also illustrates identical responses for $F_1$ and $F_2$. This is because of the obvious 1:1 ratio control.

It is interesting to test the effectiveness of the proposed control algorithms for the startup problem with the presence of process upsets. Figure 8 shows the startup response from zero initial condition. In this case, step disturbance in $C_{Af}$ of magnitude of –0.025 is introduced at time = 10 minutes from the start of the simulation. The plot of $F_2$ is omitted here because it is identical to $F_1$. The tuning parameters for all controllers are the same as before except the parameter *c* for the FLC algorithm, which is adapted to 0.75. All the proposed algorithms managed to fill up the reactor and bring the product concentration to the desired value even in the presence of upset. The GS method provided the best closed-loop performance in terms of no overshoot. It is clear from Figure 8 that the feed flow rate ($F_1$) had to be increased after the upset was introduced to compensate for the loss of moles of reactant *A*. Eventually both feed flow rates ($F_1$ and $F_2$) go to zero. In this operation, the second feed flow is equal to the first one because of the 1:1 ratio controller and the reaction stoichiometry is 1:1 for all reactants and products. Therefore, the reduction in $C_{Af}$ will create abundance of unreacted species *B*, which will dilute the concentration of the product, i.e. species C. As a result, both feed flow rates has to be zero to maintain the desired concentration of the product.

Figure 9 depicts the feedback response for the start-up operation while the first fed flow undergoes a sudden drop of –0.05 l/min at time = 10 minutes after the start-up of the simulation. This type of upset is common for our experimental setup and may occur due to drop in head pressure of the head tanks. It can be handled through feed-forward controller, however it is considered here to examine the performance of the proposed control algorithms. Figure 9 shows the closed-loop response for the GS and FLC methods only. The MGS response is shown in Figure 10. In practice, the control algorithms manipulate the feed flow indirectly through adjusting the valve opening of the specific feed stream. Therefore, Figure 9, shows also the plot for the valve opening of the first feed stream ($V_1$) in order to make the effect of upset on $F_1$ clearer. Here, the flow-valve relation is considered spontaneous with unity gain. The tuning parameters for all controllers are the same as those used in Figure 8. As shown in Figure 9, the valve opening is increased after the flow reduction due to pressure drop has been introduced. The expansion in valve opening is necessary to compensate for feed flow losses. However, to avoid diluting the product, the feed flow rates are turned off. Figure 10 shows the performance of the MGS algorithm for the same control problem. The figure shows how the MGS fails to maintain the desired concentration and creates oscillatory response at steady state. At steady state, the feed flow becomes zero at which the gain correlation produces a positive static gain. The gain sign is incorrect especially when the reactor holdup reached its maximum value. This situation created unstable feedback behavior. MGS requires the continuous measurement of an auxiliary variable to update the gain scheduler. In this case $F_1$ is used to identify the zero initial condition, which is misleading because zero value for $F_1$ dose not necessarily identifies the zero initial condition. In fact, $V$=0 accurately indicates the zero initial condition. However, the liquid holdup, $V$, can not be used as the auxiliary variable because it has only two steady state values, i.e. 0 or 2.8 *l*. This result demonstrates clearly how the MGS performance can deteriorate in the presence of modeling error or uncertainty in the gain prediction.

The simulations in Figs. 8 and 9 illustrate that filling the reactor tank and keeping the product concentration during upset in the feed conditions can be achieved only at zero feed flow rates. This means no production rate, which is undesirable. To resolve this problem, the control objective may need to be revised. However, this can not be handled in the case of GS and MGS algorithms. In the FLC case, meeting an additional control requirement can be enforced by inserting more rules to the rule base given in Table 3. For example, the rule base is augmented by the following rule:

R29: If $e$ is ZE, $V$ is ZE and $F_1$ is ZE then $\Delta u$ is SP

The simulation of Figs. 8 and 9 are repeated using the FLC algorithm with the above additional rule. The result is shown in Figure 11. It is clear that, using the additional rule, the controller managed to bring the feed flow to 0.1 l/min at steady state. However, the control objectives, i.e. achieving desired product concentration and maintaining positive feed flow at steady state, competes each other. Because of this competition, the attained steady state value for the product concentration is lower than its desired set point. Nevertheless, the FLC tuning parameters can be adjusted to trade between the two variables. This is not sought here because the main idea is simply to show the flexibility of the FLC algorithm to incorporate additional requirements.

The closed-loop response for rejecting a disturbance of magnitude of –0.02 mole/l in $C_{Af}$ while the process is operating in the maximum yield condition and fully filled tank is shown in Figure 12. Although this simulation does no belong to the startup problem investigated here, it is included for the sake of completeness. Note that the plot for the hold-up response, *V*, is omitted because it is kept constant at the maximum. The control parameters for each algorithm are the same as before except that $-50 \le k_c \le -150$ is used for GS method. If the same bounds on $k_c$ as previously shown are used, GS method can not perform well. It is clear that GS method had the best performance among the others because it delivers the smallest down-shoot and fastest recovery to steady state. However, our evaluation for these methods will not be based on their performance because the latter can be improved through tuning. Nevertheless, our evaluation will be based on their implementation issues. For example, MGS method has no manual tuning parameter or feedback to compensate for modeling error. On the other hand, GS method has another weakness. For example, it requires prior knowledge of the reasonable range for the controller gain. Moreover, two different sets for the controller gain were necessary to run the GS controller. For example, one set is used for set point change and the other for disturbance rejection. In addition, changing the value of $\kappa$ has an unpredictable effect on the GS performance.

FLC requires no pre-calculation of reasonable values for specific tuning parameters like in the GS case or specific process parameter like in the MGS case. FLC used here is found easy to implement and to tune. One group of fuzzy sets was used for $e$, $\Delta e$, and $\Delta u$ and for both the servo and regulatory problems. The unification of the fuzzy set domain is achieved through normalizing the process variable. The scaling factor for the error velocity, $b$, is found to be of little effect on the closed-loop performance. While $a$, the scaling factor for the error, is found to play the same rule that $k_c$ plays in a standard PID controllers. Specifically, increasing the value of $a$, speeds up the response and eliminates offset. Similarly, increasing the value of $c$, the scaling factor for $\Delta u$, increases the controller aggressiveness. The synthesis of the control law based on the base rules can be formulated to incorporate any additional requirements or objectives. For example in Figure 11, three objectives were merged in one control loop. One disadvantage of the FLC is the length of fuzzy rules to be developed for each control loop. Although developing such rules can be straightforward, it needs to be comprehensive to provide correct and smooth control action.

## Conclusions

The control problem of starting up a non-isothermal CSTR from zero initial conditions is addressed. Due to the non-linearity of the process during start up, the control problem has a dual objective, each of which requires different gain sign. For this reason, a standard PID algorithm may fail. Therefore, fuzzy control algorithm was tested and compared to PI controller with two gain-scheduling methods. Specifically, standard and model-based gain scheduling methods were investigated. The proposed control methods delivered acceptable feedback performance. Overall, the standard gain-scheduling method delivered the best feedback performance. However, several implementation issues can differentiate between these methods. Although the standard gain scheduling method is simple, it requires prior information about the limiting values for the proportional gain. Moreover, two different sets for the limit of the controller gain should be used for different control objectives. In addition, the controller is tuned by trial-and-error procedure without apparent guidelines. The model-based scheduling method relies on a process model or programmed correlation based on an auxiliary measurement to update its gain. Therefore, it is very sensitive to the model accuracy or uncertainty in the programmed correlation. The fuzzy control algorithm presented here has a simplified design and tuning procedures through using a unified domain for the fuzzy sets. In addition, tuning is achieved through adjusting two parameters based on apparent general guidelines. Furthermore, the synthesis procedure of the FLC algorithm is more flexible and consequently any additional known process knowledge or nonlinearity can be incorporated easily in the controller law.

NOMENCLATURE

| | |
|---|---|
| $A_s$ | Surface area, $m^2$ |
| $A$ | Constant predefined matrix for the FLC algorithm |
| $a$ | Tuning parameter for FLC algorithm |
| $b$ | Tuning parameter for FLC algorithm |
| $c$ | Tuning parameter for FLC algorithm |
| $C_i$ | Concentration of species *i*, mole/l |
| $C_{Af,}$ | Feed concentration of species *A*, mole/l |
| $C_A$, $C_c$ | Concentration of species *A* and *C* respectively, mol/l |
| $Cp_A$, $Cp_B$ | Heat capacity of specie *A and B*, J/mole $^oC$ |
| $D$ | Reactor diameter, m |
| $E$ | Activation energy |
| $e$ | Error signal |
| $F_1$ | Feed flow rate of pure component *A*, l/min |
| $F_2$ | Feed flow rate of pure component *B*, l/min |
| $F$ | Total outlet flow rate, l/min |
| $h_{air}$ | Heat transfer coefficient for air, kJ/$m^2$ $^oC$ min |
| $k_r$ | Reaction rate constant, l/mole min |
| $k_c$, $k_{co}$ | Controller gain and its reference value |
| $k_c^u$, $k_c^l$ | Upper and lower values for $k_c$ |
| $k_p$, $k_{po}$ | Process gain and its reference value |
| $k_o$ | Pre-exponential factor, l/mol min |
| $n_R$, $n_f$ | Number of rules and number of fuzzy sets, respectively |
| $Q$ | Rate of heat loss to the surrounding, kJ/min |
| $R$ | Gas constant, 0.008314 kJ/mole K |
| $T$ | Reactor temperature, $^oC$ |
| $T_{amb}$ | Ambient temperature, $^oC$ |
| $T_f$ | Feed Temperature, $^oC$ |
| $T_{ref}$ | Reference temperature, $^oC$ |
| $t$ | Time, min |
| $u$ | Manipulated variable |
| $V$ | Fluid volume (holdup), l |
| $V_r$ | Reactor volume, l |
| $\tau$ | Process time constant |

**Greek letter**

| | |
|---|---|
| $\delta$ | Location of the membership function center |
| $\kappa$ | Design parameter for the GS method |
| $\Delta H$ | Standard heat of reaction, kJ/mole |
| $\mu$ | Membership function |

**References:**


1. Abu Khalaf, A.M. and Ali, E., Automatic Start-up of A Non-Isothermal CSTR. *Chem. Eng. Ed*., 34(3), 246-251 (2000)
2. Lee, C.C., Fuzzy Logic in Control Systems: Fuzzy Logic Controller – Part I, *IEEE Trans. on Sys. Man. And Cybernetics*, 20(2), 1990, 404-418.
3. Lee, C.C., Fuzzy Logic in Control Systems: Fuzzy Logic Controller – Part II, *IEEE Trans. on Sys. Man. and Cybernetics*, 20(2), 1990, 419-435.
4. Zadeh, L. A., Fuzzy Sets, *Information Control*, 8, 1965, 330-353.
5. Mamdani, E. H., Application of Fuzzy Algorithms for Control of Simple Dynamic Plants, *IEE Proceed*., 121(3), 1974, 585-588.
6. King. P. J., and Mamdani, E. H. The Application of Fuzzy Control Systems to Industrial Processes, *Automatica*, 13(3), 1977, 25-242.
7. Bernard, J. A. Use of Rule-Based System for Process Control, *IEEE Control Systems Magazines*, 8(5), 1988, 3-13.
8. Kishimoto, M., Yoshida T. and Young M., Application of Fuzzy Expert System to Fermentation Process, *IFAC Workshop PCPI*, Osaka, Japan, 2, 1989.
9. Parekh, M. and Desai M., H. H. Lee and R. R. Rhinehart, Inline Control of Nonlinear pH Neutralization Based on Fuzzy Logic, *IEEE Transaction n Components, Packaging and Mfg. Tech*. Part A 17, 1994, 192-201.
10. Rhinehart, R. R., H. H. Lee and P. Murugan, Improve Process Control Using Fuzzy Logic, *Chem. Eng. Progress*, 91, 1996, 60-65.
11. S. R. Inamdar and M S. Chiu, Fuzzy Logic Control of an Unstable Biological Reactor, *Chem. Eng. Technol*. 20, 1997, 414-418.
12. Garrido, R., Adroer, M., and Pocj, M., Wastewater Neutralization Control Based on Fuzzy Logic Simulation Results, *Ind. Eng. Hem. Res*., 36, 1997, 1665-1674.
13. Rao, A. V.K., Jayaraman, B.D. Kulkarni, S. Japanwala and P. Shevgaonkar, Improved Controller Performance with Simple Fuzzy Rules, *Hydrocarbon Processing*, May 1999, 97-100.
14. Colcazier, J. D., Fuzzy Logic -An Effective Alternative to PID Control, ISA paper #96-077, 1996, 729-738.
15. Abu-Khalaf, A.M., Mathematical Modeling of an Experimental Reaction System, *Chem. Eng. Ed*., 28(1), 1994, 48-51.
16. Abu-Khalaf A. M., Start-Up of a Non-Isothermal CSTR: Mathematical Modeling, *Chem. Eng. Ed*., 31(4), 1997, 250-253.
17. Yen, J. and Langari, R. *Fuzzy Control*, Prentice Hall, New Jersey, 1999.
18. Qin, S.J. and G. Borders, A Multiregion Fuzzy Logic Controller for Nonlinear Process Control, IEEE Trans. on Fuzzy Systems, 2, 1994, 74-81.
19. Ogunnaike, B. and W. Ray, “Process Dynamics, Modeling, and Control”, Oxford University Press (1994)
20. Shinsky, F.G., *Process Control Systems*, McGraw-Hill, New York, 1988.

Table 1: Process parameters

| Parameter | Value | Parameter | Value |
|---|---|---|---|
| $T_f$ | 24 $^oC$ | $R$ | 0.008314 J/mol $^oK$ |
| $T_{amb}$ | 29 $^oC$ | $Cp_A$ | 75.25 J/mol $^oC$ |
| $V_r$ | 2.8 l | $Cp_B$ | 175.3 J/mol $^oC$ |
| $h_{air}$ | 2.5 J/m$^2$ $^oC$ min | $Cp_C$ | 78.2 J/mol $^oC$ |
| $\Delta H$ | -1.5 kJ/mol | $Cp_D$ | 103.8 J/mol $^oC$ |
| $D$ | 15 cm | $C_{Af}$ | 0.1 mol/l |
| $T_{ref}$ | 24 $^oC$ | $C_{Bf}$ | 0.1 mol/l |
| $E$ | 48.32 | $C_{Cf}$ | 0.0 mol/l |
| $k_o$ | 109.31l/mole min | $C_{Df}$ | 0.0 mol/l |

Table 2: Steady state operating conditions

| Variable | Initial | Min. throughputs | Max. yield | Max. throughputs |
|---|---|---|---|---|
| $F_1$ | 0.0 l/min | 0.01 l/min | 0.1 l/min | 1 l/min |
| $F_2$ | 0.0 l/min | 0.01 l/min | 0.1 l/min | 1 l/min |
| $C_A$ | 0.1 mol/l | 0.0062 mol/l | 0.0174 mol/l | 0.03676 mol/l |
| $C_B$ | 0.1 mol/l | 0.0062 mol/l | 0.0174 mol/l | 0.03676 mol/l |
| $C_C$ | 0.0 mol/l | 0.044 mol/l | 0.0326 mol/l | 0.01324 mol/l |
| $C_D$ | 0.0 mol/l | 0.044 mol/l | 0.0326 mol/l | 0.01324 mol/l |
| $V$ | 0.0 l | 2.8 l | 2.8 l | 2.8 l |
| $T$ | 24 $^oC$ | 27.7 $^oC$ | 26.5 $^oC$ | 25.04 $^oC$ |
| $k_p$ | 0.011 | - 0.1382 | - 0.0338 | - 0.0039 |
| $\tau$ | - | 8 | 5 | 1.35 |

Table 3: Rule base for the FLC algorithm

| no. | Fuzzy rule | Description | Results (Conclusion) |
|---|---|---|---|
| R1<br>R2<br>R3<br>R4 | If e is LP & Δe is LP then Δu is LP<br>If e is SP & Δe is SP then Δu is SP<br>If e is SN & Δe is SN then Δu is SN<br>If e is LN & Δe is LN then Δu is LN | Starting up, change the input in response to the set point change | $\mu_{1,1}(\Delta u) = \min(\mu_1(e),\mu_1(\Delta e))$<br>$\mu_{2,2}(\Delta u) = \min(\mu_2(e),\mu_2(\Delta e))$<br>$\mu_{4,3}(\Delta u) = \min(\mu_4(e),\mu_4(\Delta e))$<br>$\mu_{5,4}(\Delta u) = \min(\mu_5(e),\mu_5(\Delta e))$ |
| R5<br>R6<br>R7<br>R8 | If e is LP & Δe is ZE then Δu is LP<br>If e is SP & Δe is ZE then Δu is SP<br>If e is SN & Δe is ZE then Δu is SN<br>If e is LN & Δe is ZE then Δu is LN | Error not changing, change input accordingly. | $\mu_{1,5}(\Delta u) = \min(\mu_1(e),\mu_3(\Delta e))$<br>$\mu_{2,6}(\Delta u) = \min(\mu_2(e),\mu_3(\Delta e))$<br>$\mu_{4,7}(\Delta u) = \min(\mu_4(e),\mu_3(\Delta e))$<br>$\mu_{5,8}(\Delta u) = \min(\mu_5(e),\mu_3(\Delta e))$ |
| R9<br>R10<br>R11<br>R12 | If e is LP & Δe is SN then Δu is ZE<br>If e is SP & Δe is SN then Δu is ZE<br>If e is SN & Δe is SP then Δu is ZE<br>If e is LN & Δe is SP then Δu is ZE | Moving along, maintain input | $\mu_{3,9}(\Delta u) = \min(\mu_1(e),\mu_4(\Delta e))$<br>$\mu_{3,10}(\Delta u) = \min(\mu_2(e),\mu_4(\Delta e))$<br>$\mu_{3,11}(\Delta u) = \min(\mu_4(e),\mu_2(\Delta e))$<br>$\mu_{3,12}(\Delta u) = \min(\mu_5(e),\mu_2(\Delta e))$ |
| R13<br>R14<br>R15<br>R16 | If e is LP & Δe is SP then Δu is LP<br>If e is SP & Δe is LP then Δu is LP<br>If e is SN & Δe is LN then Δu is LN<br>If e is LN & Δe is SN then Δu is LN | Getting worse, reverse input somewhat | $\mu_{1,13}(\Delta u) = \min(\mu_1(e),\mu_2(\Delta e))$<br>$\mu_{1,14}(\Delta u) = \min(\mu_2(e),\mu_1(\Delta e))$<br>$\mu_{5,15}(\Delta u) = \min(\mu_4(e),\mu_5(\Delta e))$<br>$\mu_{5,16}(\Delta u) = \min(\mu_5(e),\mu_4(\Delta e))$ |
| R17<br>R18<br>R19<br>R20 | If e is LP & Δe is LN then Δu is SN<br>If e is SP & Δe is LN then Δu is SN<br>If e is SN & Δe is LP then Δu is SP<br>If e is LN & Δe is LP then Δu is SP | Error changing too fast, adjust input somewhat | $\mu_{4,17}(\Delta u) = \min(\mu_1(e),\mu_5(\Delta e))$<br>$\mu_{4,18}(\Delta u) = \min(\mu_2(e),\mu_5(\Delta e))$<br>$\mu_{2,19}(\Delta u) = \min(\mu_4(e),\mu_1(\Delta e))$<br>$\mu_{2,20}(\Delta u) = \min(\mu_5(e),\mu_1(\Delta e))$ |
| R21 | If e is ZE & Δe is ZE then Δu is ZE | Reached equilibrium | $\mu_{3,21}(\Delta u) = \min(\mu_3(e),\mu_3(\Delta e))$ |
| R22<br>R23 | If e is ZE & Δe is LN then Δu is SN<br>If e is ZE & Δe is LP then Δu is SP | Error is nil but changing, take action | $\mu_{4,22}(\Delta u) = \min(\mu_3(e),\mu_5(\Delta e))$<br>$\mu_{2,23}(\Delta u) = \min(\mu_3(e),\mu_1(\Delta e))$ |
| R24<br>R25 | If e is ZE & Δe is SN then Δu is ZE<br>If e is ZE & Δe is SP then Δu is ZE | Error is nil and changing insignificantly, no action- wait and see | $\mu_{3,24}(\Delta u) = \min(\mu_3(e),\mu_4(\Delta e))$<br>$\mu_{3,25}(\Delta u) = \min(\mu_3(e),\mu_2(\Delta e))$ |

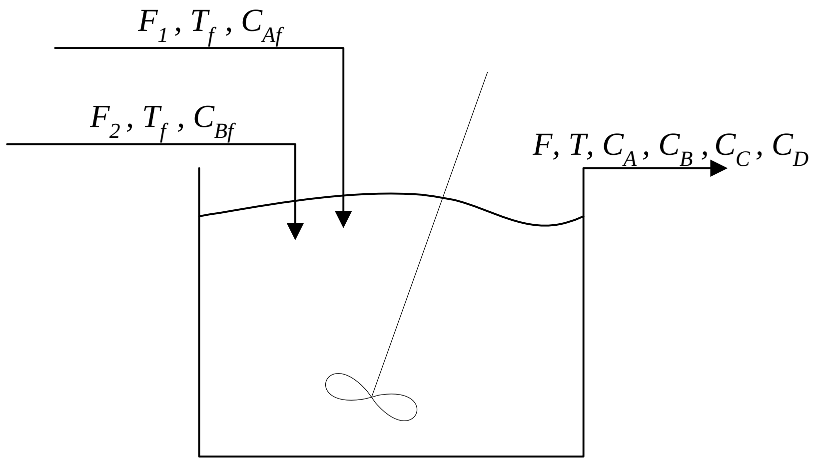


**Figure 1: Schematic of the non-isothermal CSTR**

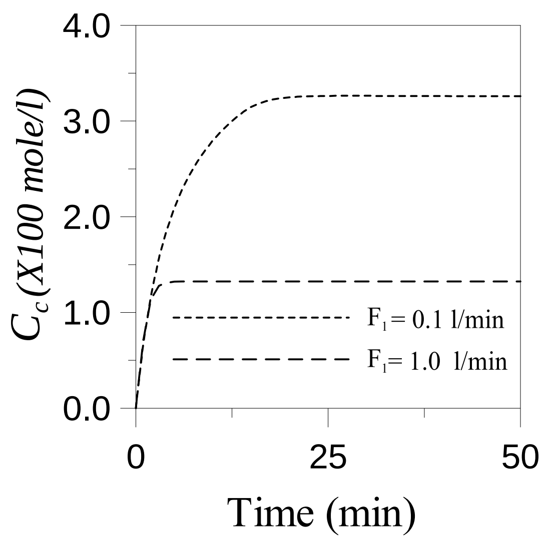


Figure 2: Open loop response to two step changes in the feed flow rate.

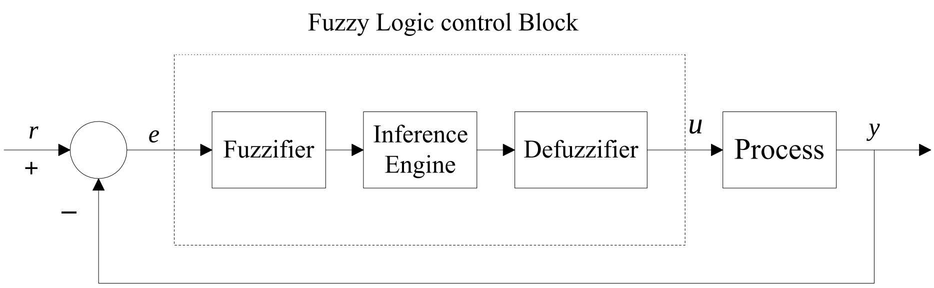


Figure 3: Block diagram for the FLC algorithm.

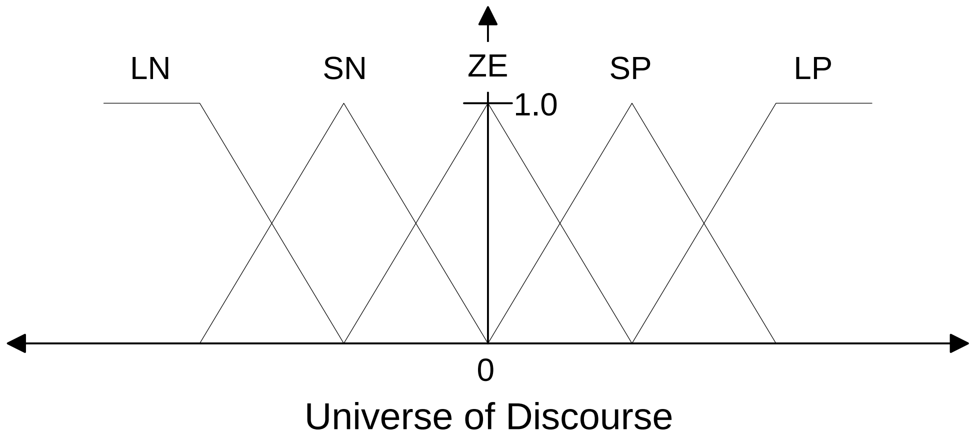


Figure 4: Example of fuzzy set.

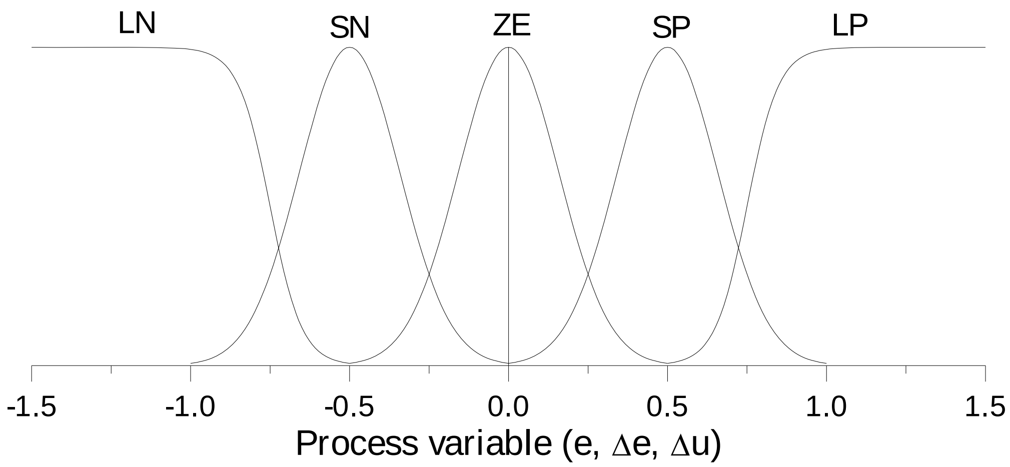


Figure 5: Typical fuzzy set used.

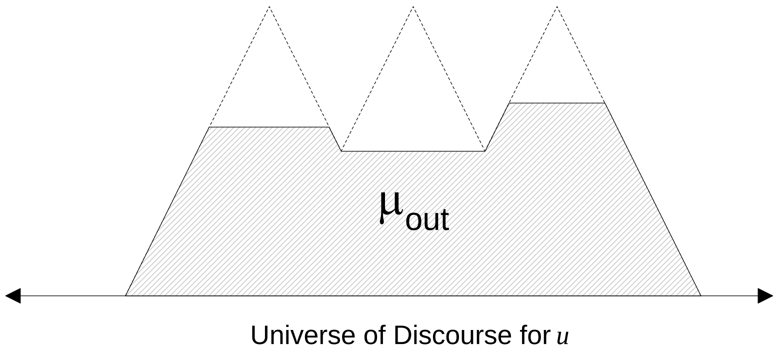


Figure 6: Example of output fuzzy set.

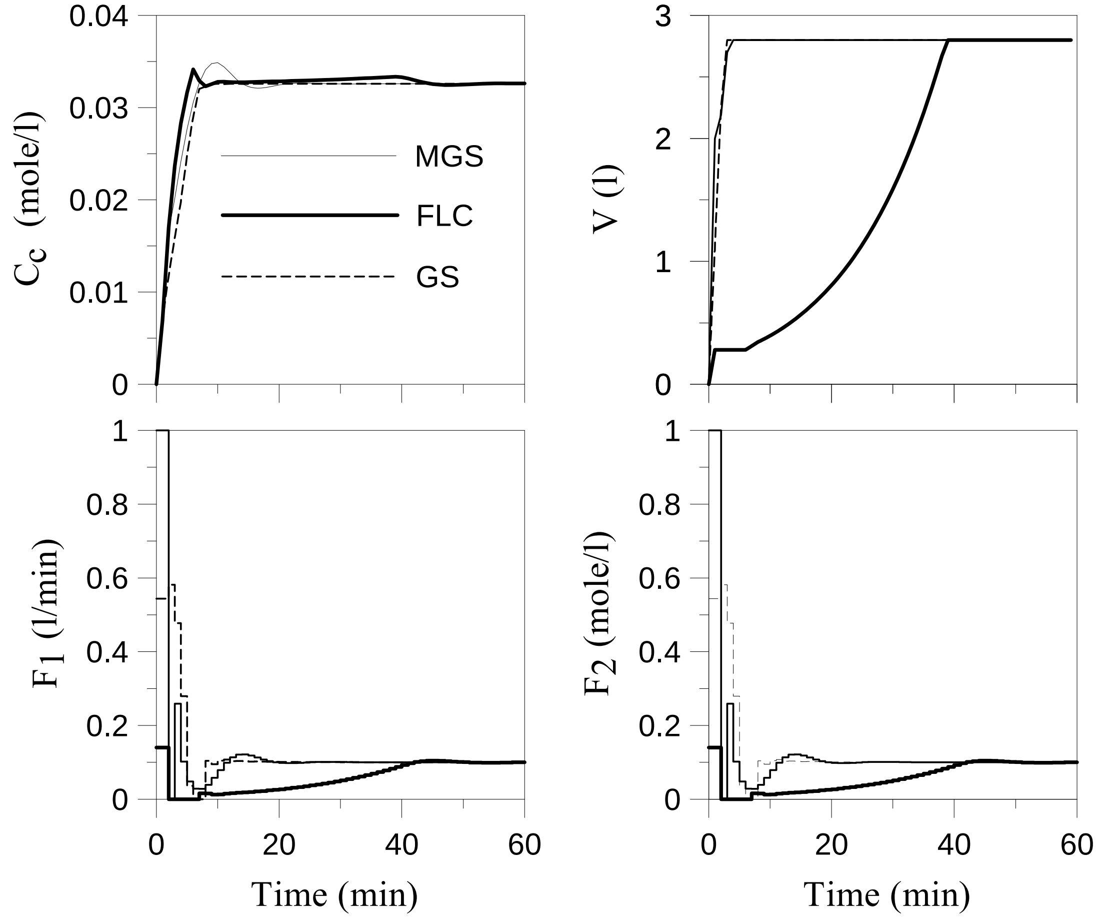


Figure 7: Feedback response for Startup operation.

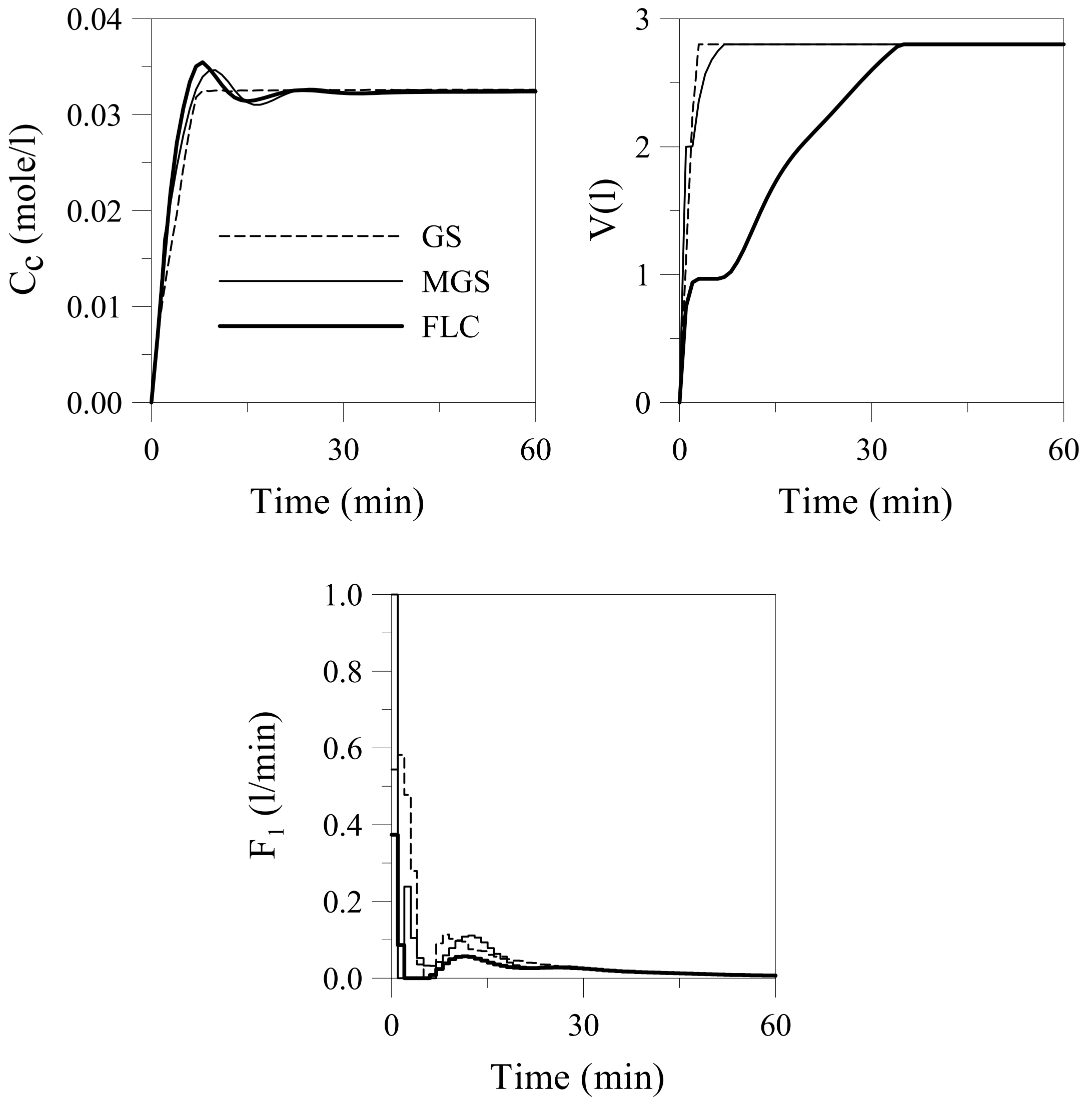


Figure 8: Feedback response to startup operation with upset in $C_{Af}$

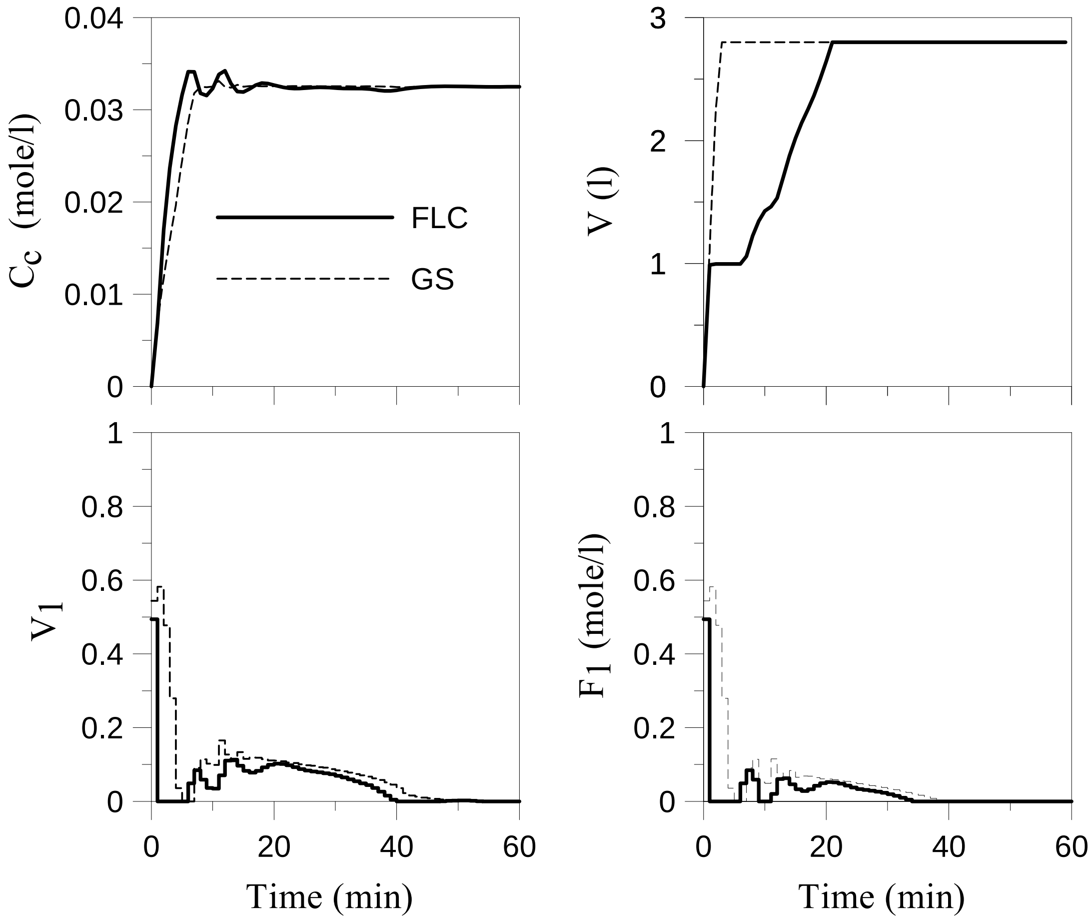


Figure 9: Feedback response to startup operation with upset in $F_1$.

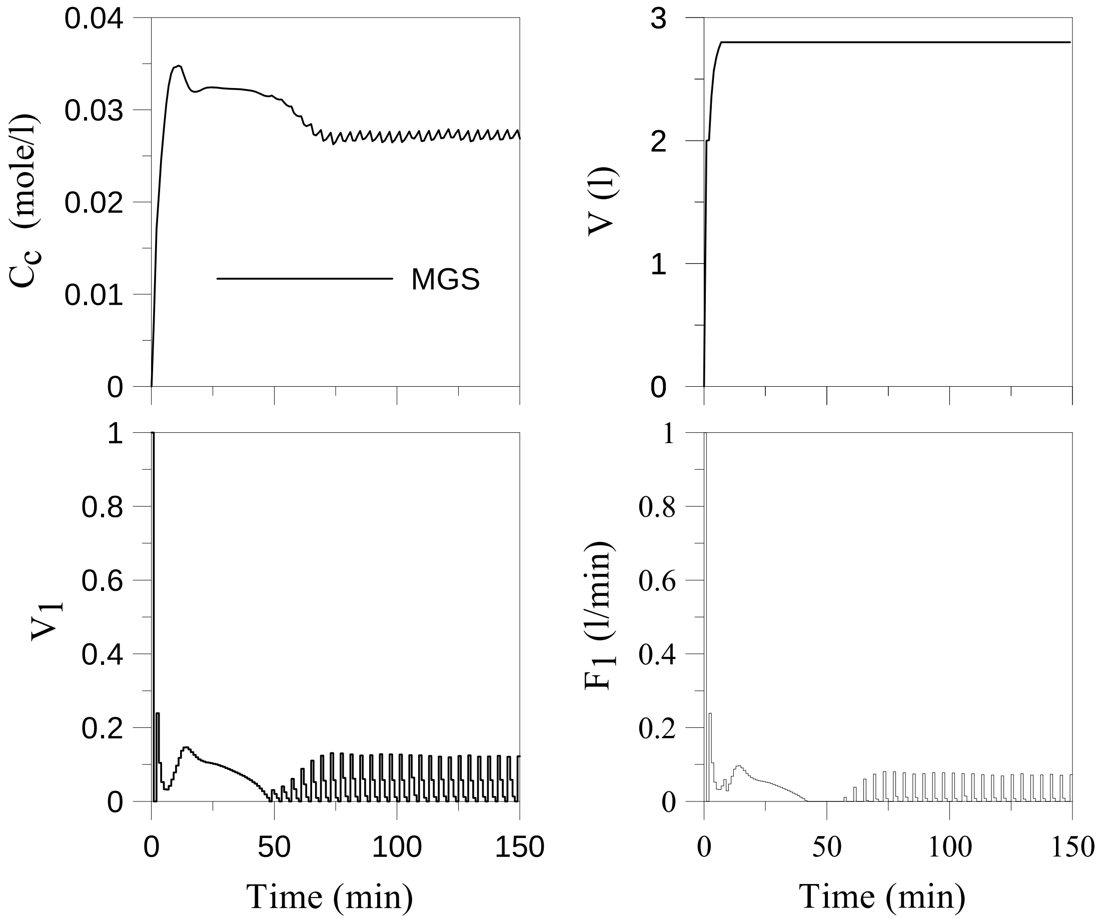


Figure 10: Feedback response for startup operation with upset in $F_1$ using MGS method.

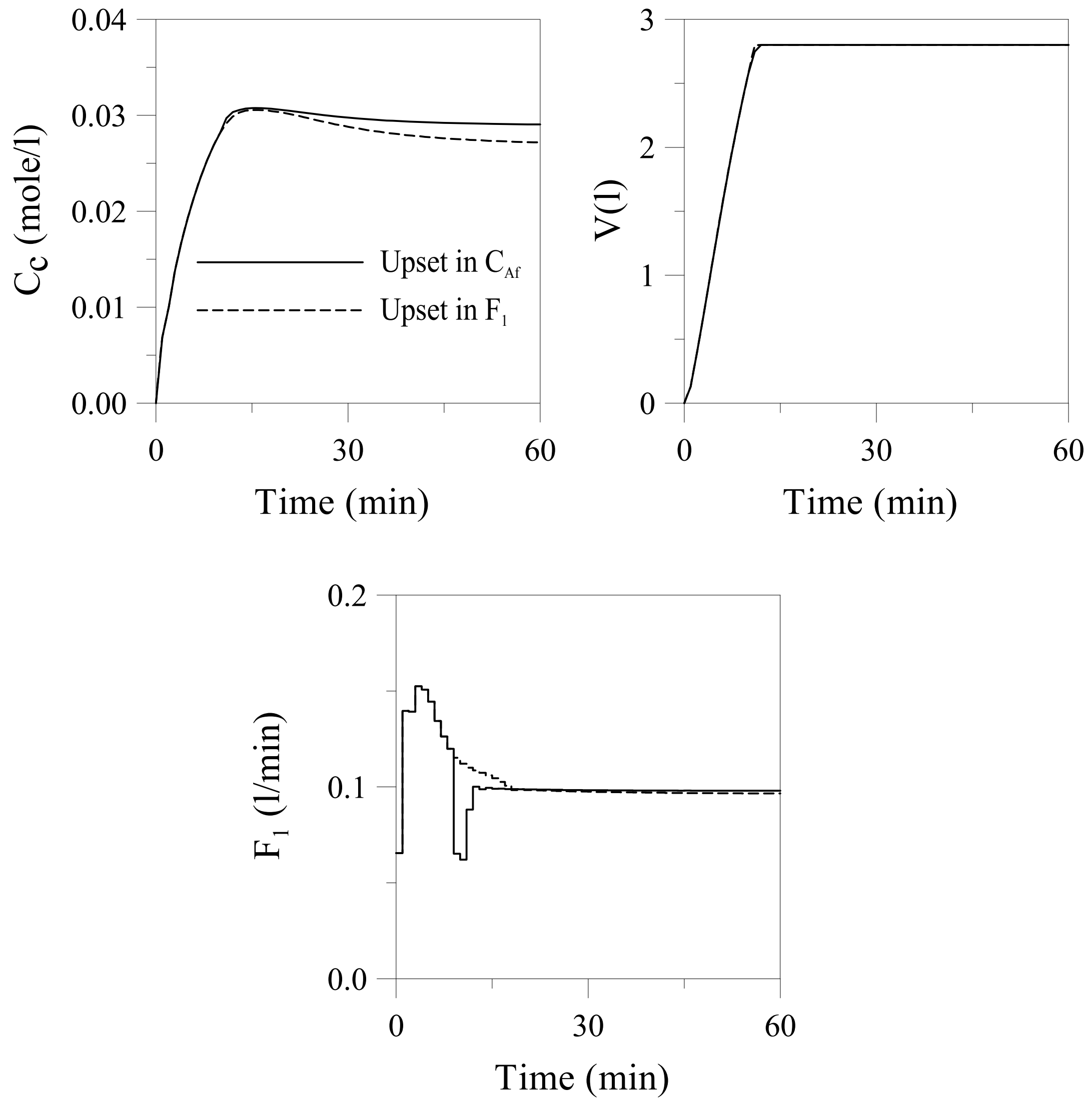


Figure 11: Feedback response for startup with disturbances using modified rules for FLC.

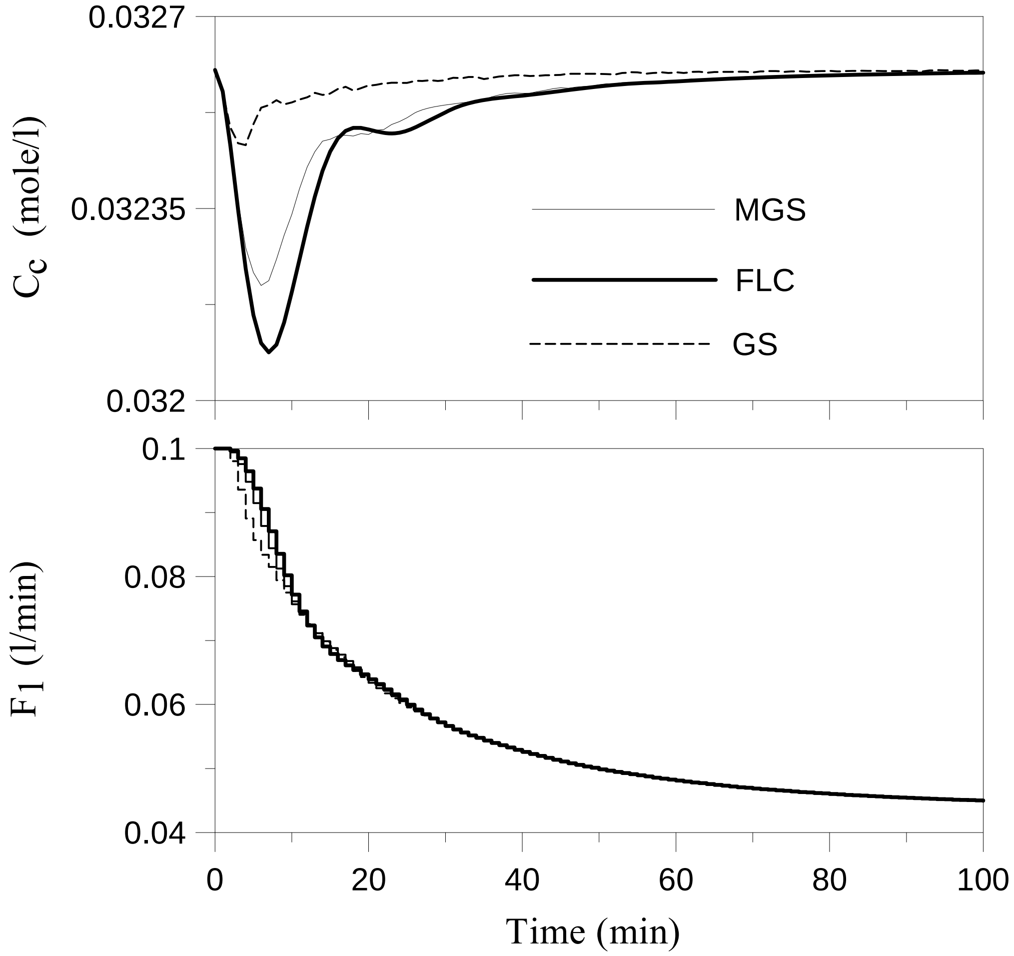


Figure 12: Feedback response for disturbance step change at fully filled tank.