Ratio, Override, and Selective Control

Alessandro Anzalone, Ph.D., CPT
Hillsborough Community College, Brandon Campus

Agenda

1. Introduction
2. Signals and Computing Algorithms
    - ✓ Signals
    - ✓ Programming
    - ✓ Scaling Computing Algorithms
    - ✓ Significance of Signals
3. Ratio Control
4. Override, or Constraint, Control
5. Selective Control
6. Designing Control Systems
7. Summary
8. References

Introduction

In the previous chapter we began the presentation of control techniques that aid simple feedback to provide improved control performance. In the present chapter we continue this presentation with three other techniques: ratio, override, and selective control; override control is also sometimes referred to as constraint control. Ratio control is commonly used to maintain two or more streams in a prescribed ratio. Override and selective control are usually implemented for safety and optimization considerations. These two techniques often deal with *multiple control objectives* (controlled variables) and a single manipulated variable; up to now we have dealt only with processes with *one control objective*. The chapter begins with a presentation of distributed control systems (DCSs), how they handle signals, and some computing algorithms and programming needed for implementing control techniques.

Signals and Computing Algorithms

Many of the control techniques presented in this and subsequent chapters require some amount of computing power. That is, many of these techniques require the multiplication, division, addition, subtraction, and so on, of different signals. Several years ago all of these calculations were implemented with analog instrumentation. Computers allow for a simpler, more flexible, more accurate, more reliable, and less expensive implementation of these functions.

Signals

There are two different ways that field signals are handled once they enter the DCS. The first way is to convert the signal received by the computer into a number with engineering units. For example, if a signal is read from a temperature transmitter, the number kept in memory by the computer is the temperature in degrees. The computer is given the low value of the range and the span of the transmitter, and with this information it converts the raw signal from the field into a number in engineering units. A possible command in the DCS to read a certain input is

variable = AIN(input channel #, low value of range, span of transmitter)

or T = AIN(3, 50,100)

Signals

This command instructs the DCS to read an analog input signal (AIN) in channel 3, it tells the DCS that the signal comes from a transmitter with a low value of 50 and a span of 100, and it instructs the DCS to assign the name T to the variable read (possibly a temperature from a transmitter with a range of 50 to 150°C). If the signal read had been 60%, 13.6mA, then T = 110°C.

Signals

The second way of handling signals, and fortunately the least common, is not by converting them to engineering units but by keeping them as a percentage, or fraction, of the span. In this case the input command is something like

variable = AIN(input channel)

or T = AIN(3)

and the result, for the same example, is T = 60% (or 0.6)

Signals

In DCSs that work in engineering units, the range of the transmitter providing the controlled variable must be supplied to the PID controller (there are different ways to do so).With this information, the controller converts both the variable and the set point to percent values before applying the PID algorithm. This is done because the error is calculated in %TO. Remember, the KC units are %CO/%TO. Thus the controller output is then %CO. A possible way to “call” a PID controller could be

OUT = PID(controlled variable, set point, low value of range, span of transmitter)

or OUT = PID(T, 75, 50,100)

Programming

Block-Oriented Programming. Block-oriented programming is software in a subroutine-type form, referred to as computing algorithms or computing blocks. Each block performs a specified mathematical manipulation. Thus, to develop a control strategy, the computing blocks are linked together, the output of one block being the input to another block. This linking procedure is often referred to as configuring the control system.

Programming

Some typical calculations (there are many others) performed by computing blocks are:

1. Addition/subtraction. The output signal is obtained by adding and/or subtracting the input signals.
2. Multiplication/division. The output signal is obtained by multiplying and/or dividing the input signals.
3. Square root. The output signal is obtained by extracting the square root of the input signal.
4. High/low selector. The output signal is the highest/lowest of two or more input signals.
5. High/low limiter. The output signal is the input signal limited to a preset high/low limit value.

Programming

6. Function generator, or signal characterization. The output signal is a function of the input signal. The function is defined by configuring the x, y coordinates.
7. Integrator. The output signal is the time integral of the input signal. The industrial term for integrator is totalizer.
8. Lead/lag. The output signal is the response of the transfer function given below. This calculation is often used in control schemes, such as feedforward, where dynamic compensation is required.

$$\text{Output} = \frac{\tau_{\text{ld}} s + 1}{\tau_{\text{lg}} s + 1} \text{ x input}$$

9. Dead time. The output signal is equal to a delayed input signal. This calculation is very easily done with computers but is extremely difficult to do with analog instrumentation.

Programming

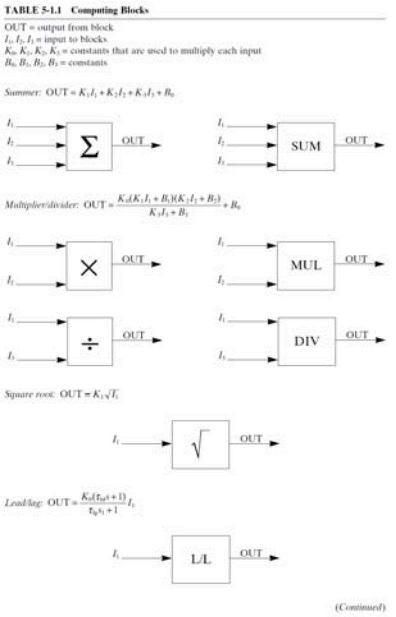


TABLE 5-1.1 Computing Blocks

OUT = output from block
I_1, I_2, I_3 = input to blocks
K_0, K_1, K_2, K_3 = constants that are used to multiply each input
B_0, B_1, B_2, B_3 = constants

Summer: $\text{OUT} = K_1 I_1 + K_2 I_2 + K_3 I_3 + B_0$

Multiplier/divider: $\text{OUT} = \frac{K_0(K_1 I_1 + B_1)(K_2 I_2 + B_2)}{K_3 I_3 + B_3} + B_0$

Square root: $\text{OUT} = K_1 \sqrt{I_1}$

Lead/lag: $\text{OUT} = \frac{K_0(\tau_{ld} s + 1)}{\tau_{lg} s_1 + 1} I_1$

(Continued)

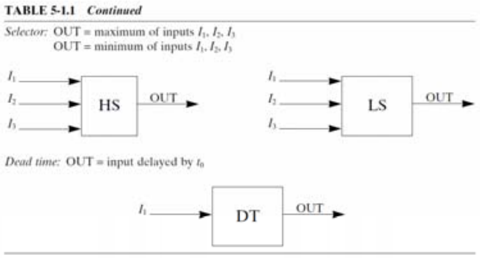


TABLE 5-1.1 *Continued*

Selector: OUT = maximum of inputs I_1, I_2, I_3
OUT = minimum of inputs I_1, I_2, I_3

Dead time: OUT = input delayed by t_0

Programming

Software-Oriented Programming. Manufacturers have developed their own programming languages, but they are all similar and resemble Fortran, Basic, or C. Table 5-1.2 presents the programming language we use in this book; this language is similar to those used by different manufacturers.

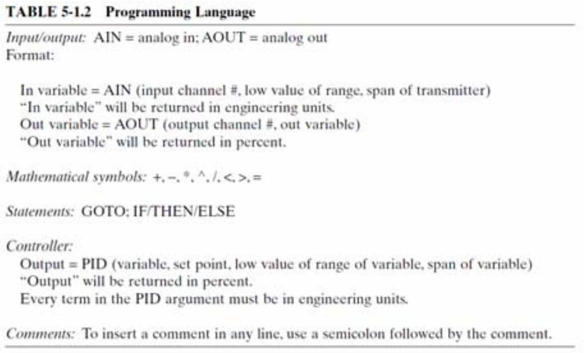


TABLE 5-1.2 Programming Language

Input/output: AIN = analog in; AOUT = analog out
Format:

In variable = AIN (input channel #, low value of range, span of transmitter)
"In variable" will be returned in engineering units.
Out variable = AOUT (output channel #, out variable)
"Out variable" will be returned in percent.

Mathematical symbols: +, −, *, ^, /, <, >, =

Statements: GOTO; IF/THEN/ELSE

Controller:
Output = PID (variable, set point, low value of range of variable, span of variable)
"Output" will be returned in percent.
Every term in the PID argument must be in engineering units.

Comments: To insert a comment in any line, use a semicolon followed by the comment.

Scaling Computing Algorithms

When signals are handled as a percent, or fraction, of span, additional calculations must be performed before the required mathematical manipulations can be implemented. The necessity and meaning of the additional calculations are explained by the following. Consider a tank, shown in Fig. 5-1.1, where temperature transmitters with different ranges measure temperatures at three different locations in the tank.

Scaling Computing Algorithms

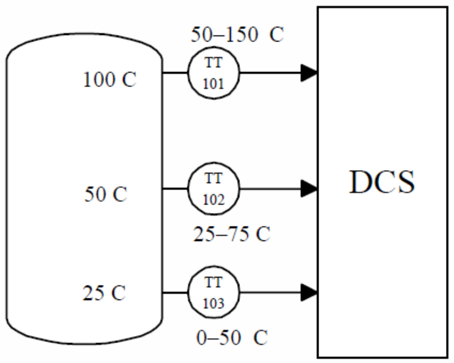


Figure 5-1.1 Tank with three temperature transmitters.

Scaling Computing Algorithms

The figure shows the transmitter ranges and the steady-state values of each temperature, which are at midvalue of each range. It is desired to compute the average temperature in the tank. This computation is straightforward for the control system that reads each signal and converts it to engineering units. The three values are added together and divided by 3; the program in Fig. 5-1.2 does just that. The first three lines, T101, T102, and T103, read in the temperature, and the fourth statement calculates the average temperature, TAVG.

Scaling Computing Algorithms

The figure shows the transmitter ranges and the steady-state For control systems that treat each signal as a percent of span, this simple computation would result in an answer without much significance; Fig. 5-1.3 shows this program. That is, because each signal is 50% of its range, the computation result would also be 50%. However, 50% of what range? How do we translate this answer into a temperature? Furthermore, notice that even though every input signal is 50%, their measured temperatures are different because the ranges are different. Thus, for the computation to “make sense,” the range of each input signal, and a chosen range for the output variable, must be considered. The consideration of each range will ensure compatibility between input and output signals, and it is called scaling. Reference 1 presents the method to scale the computations.

Significance of Signals

During the presentation of the types of field signals in Chapters 1 and 4, and in the discussion earlier in this section, it was mentioned that signals are used by the instruments to convey information and that, therefore, every signal has physical significance; that is, every signal used in the control scheme has some meaning. Signals are in percent, but percent of what (pressure, temperature, flow, etc.)? The what is the meaning of the signal. It is now important to stress this fact again as we embark on the design of complex strategies to improve control performance.

Significance of Signals

As mentioned earlier in this chapter, the new strategies frequently require the manipulation of signals in order to calculate controlled variables, set points, or decide on control actions. To perform these calculations correctly, it is most important to understand the significance of the signals. Very often, the first step in the design of a control strategy is to give a signal, sometimes referred to as the master signal, a physical significance. Then, based on the given significance, the strategy is designed. Currently, this presentation may seem somewhat abstract; however, as we continue with the study of different control strategies, the presentation will become clear and realistic.

Ratio Control

A commonly used process control technique is ratio control, which is the term used to describe the strategy where one variable is manipulated to keep it as a ratio or proportion of another. In this section we present two industrial examples to show its meaning and implementation. The first example is a simple and common one and explains clearly the need for ratio control.

Ratio Control

Example 5-2.1. Assume that it is required to blend two liquid streams, A and B, in some proportion, or ratio, R; the process is shown in Fig. 5-2.1.The ratio is R = F_B/F_A, where F_A and F_B are the flow rates of streams A and B, respectively. An easy way of accomplishing this task is shown in the figure.

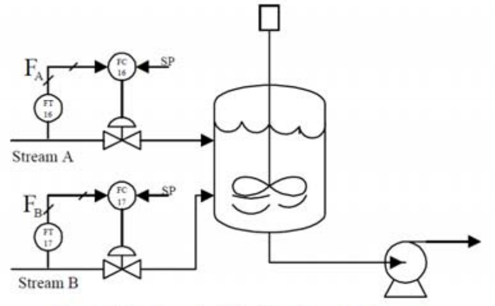


Figure 5-2.1 Control of blending of two liquid streams.

Ratio Control

Each stream is controlled by a flow loop in which the set points to the controllers are set such that the liquids are blended in the correct ratio. However, suppose now that one of the streams, stream A for example, cannot be controlled, just measured. The flow rate of this stream, often referred to as *wild flow*, is usually manipulated to control something else, such as level or temperature, upstream. The controlling task is now more difficult. Somehow the flow rate of stream B must vary, as the flow rate of stream A varies, to maintain the blend in the correct ratio. Two possible ratio control schemes are shown in Fig. 5-2.2.

Ratio Control

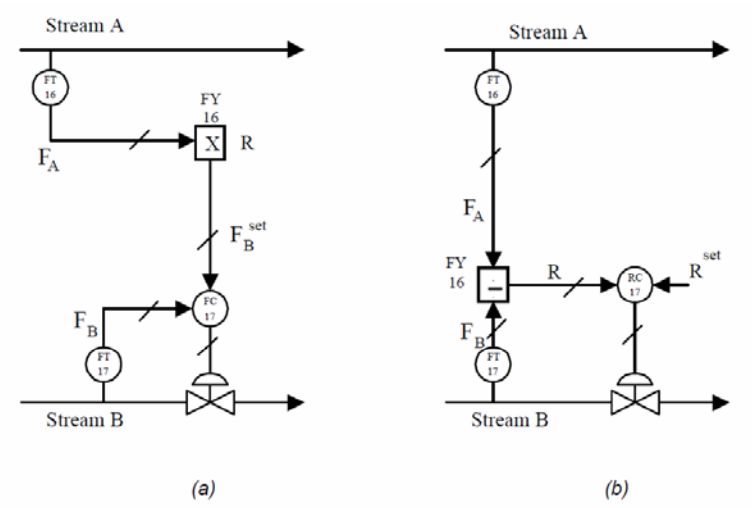


Figure 5-2.2 Ratio control of blending system.

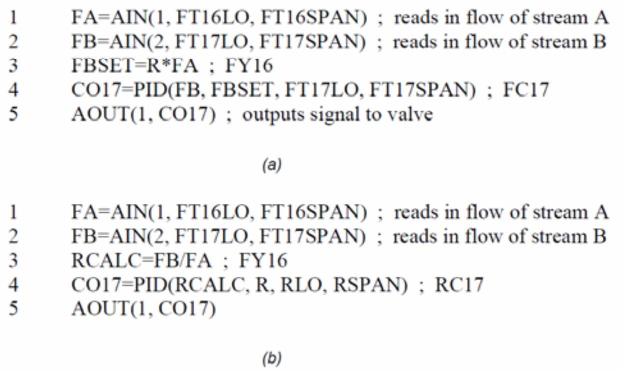
Ratio Control

The scheme shown in Fig. 5-2.2a consists of measuring the wild flow and multiplying it by the desired ratio, in FY16, to obtain the required flow rate of stream B; that is, $F_B^{set} = RF_A$. The output of FY16 is the required flow rate of stream B, and it is used as the set point to the flow controller of stream B, FC17. So as the flow rate of stream A varies, the set point to the flow controller of stream B will vary accordingly to maintain both streams at the ratio required. If a new ratio between the two streams is required, the new ratio is set in the multiplier. The set point to the flow controller of stream B is set from a computation, not locally. Figure 5-2.3a shows the software equivalent to Fig. 5-2.2a and assumes that the control system works in engineering units. FT16LO, FT16SPAN, FT17LO, and FT17SPAN are the low value and span of FT16 and FT17.

Ratio Control

The ratio control scheme shown in Fig. 5-2.2b consists of measuring both streams and dividing them, in FY16, to obtain the actual ratio flowing through the system. The calculated ratio is then sent to a controller, RC17, which manipulates the flow of stream B to maintain the set point. The set point to this controller is the required ratio and it is set locally. Figure 5-2.3b shows the equivalent scheme using software. Note that in the controller it is necessary to specify RLO and RSPAN, which are the low value and span you expect the ratio to change. This is the same as selecting a ratio transmitter range.

Ratio Control



```
FA=AIN(1, FT16LO, FT16SPAN) ; reads in flow of stream A
FB=AIN(2, FT17LO, FT17SPAN) ; reads in flow of stream B
FBSET=R*FA ; FY16
CO17=PID(FB, FBSET, FT17LO, FT17SPAN) ; FC17
AOUT(1, CO17) ; outputs signal to valve
```

(a)

```
FA=AIN(1, FT16LO, FT16SPAN) ; reads in flow of stream A
FB=AIN(2, FT17LO, FT17SPAN) ; reads in flow of stream B
RCALC=FB/FA ; FY16
CO17=PID(RCALC, R, RLO, RSPAN) ; RC17
AOUT(1, CO17)
```

(b)

Figure 5-2.3 Software equivalent of Fig. 5-2.2.

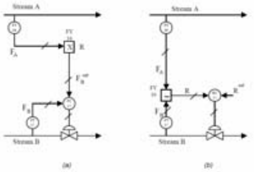


Figure 5-2.2 Ratio control of blending system.

Ratio Control

From a practical point of view, even if both streams can be controlled, the implementation of ratio control may still be more convenient than the control system shown in Fig. 5-2.1. Figure 5-2.4 shows a ratio control scheme for this case. If the total flow must be changed, the operator needs to change only one flow, the set point to FC16; then the set point to FC17 changes automatically once the flow rate of stream A changes. In the control system of Fig. 5-2.1 the operator needs to change two flows, the set points to FC16 and FC17.

Ratio Control

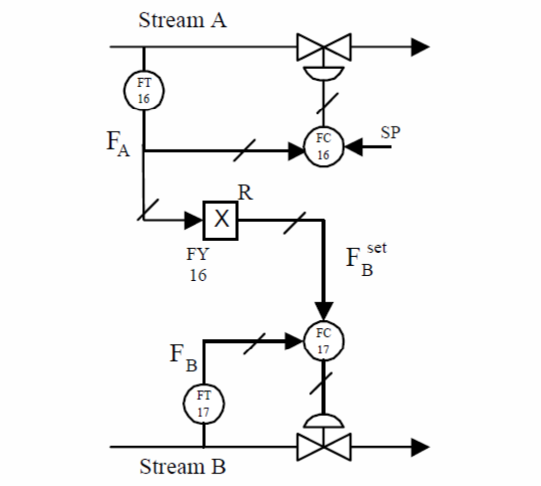


Figure 5-2.4 Ratio control of blending system.

Ratio Control

The schemes shown in Figs. 5-2.2a and 5-2.4 are quite common in the process industries. Recalling what was presented about computing blocks in section 5-1, we realize that the implementation of the ratio stations can simply be accomplished with the use of a unit such as the one shown in Table 5-1.2. Most computer control systems offer a controller, referred to as PID-RATIO, that accepts a signal, applies the same algorithm as the ratio unit, FY16, in Fig. 5-2.2a, and uses the internal result as its set point. Thus, if a PID-RATIO is used, the calculations done by FY16 and FC17 in Fig. 5-2.4 are performed in only one block.

Ratio Control

Example 5-2.2. Another common example of ratio control used in the process industries is control of the air/fuel ratio to a boiler or furnace. Air is introduced in a set excess of that required stoichiometrically for combustion of the fuel; this is done to ensure complete combustion. Incomplete combustion results not only in inefficient use of the fuel, but may also result in smoke and the production of pollutants. In addition, if not enough air is introduced, this may result in pockets of pure fuel inside the combustion chamber—not a very safe condition. The excess air introduced is dependent on the type of fuel, fuel composition, and equipment used. However, the greater the amount of excess air introduced, the greater the energy losses through the stack gases. Therefore, control of the air and fuel flows is most important for safe and economical operation.

Ratio Control

The flow of combustibles is generally used as the manipulated variable to maintain the pressure of the steam produced in the boiler at some desired value. Figure 5-2.5 shows one way to control the steam pressure as well as the air/fuel ratio control scheme. This scheme is called parallel positioning control with manually adjusted fuel/air ratio. The steam pressure is transmitted by PT22 to the pressure controller PC22, and this controller manipulates a signal, often referred to as the boiler master signal, to the fuel valve. Simultaneously, the controller also manipulates the air damper through the ratio unit FY24. This ratio station sets the air/fuel ratio required.

Ratio Control

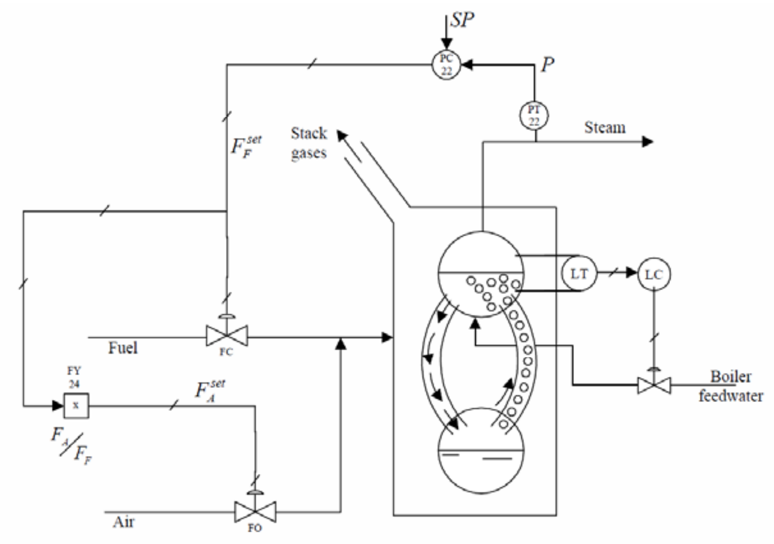
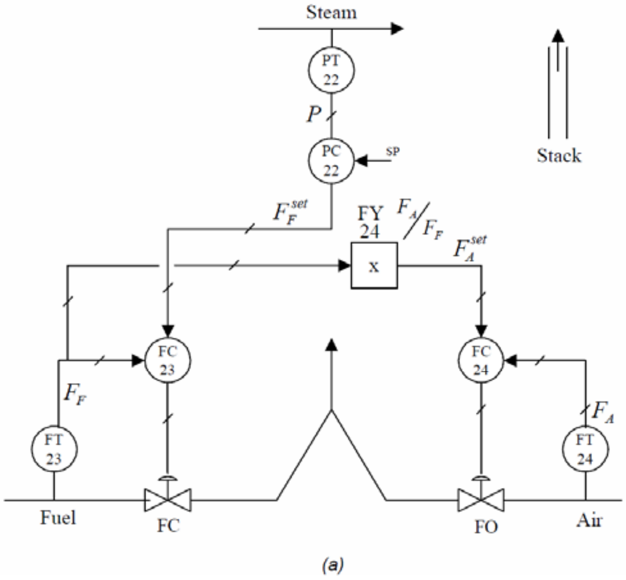


Figure 5-2.5 Parallel positioning control with manually adjusted air/fuel ratio.

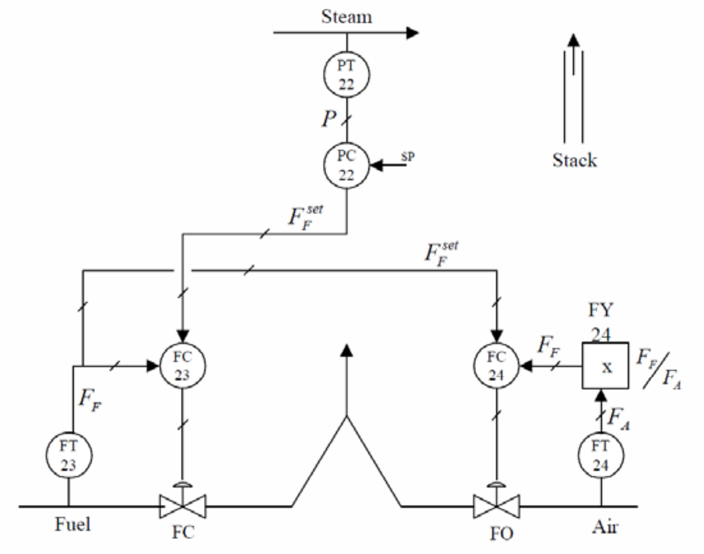
Ratio Control

The control scheme shown in Fig. 5-2.5 does not actually maintain an airflow/fuel flow ratio, but rather, maintains only a ratio of signals to the final control elements; the actual flows are not measured and used. The flow through the valves depends on the signals and on the pressure drop across them. Consequently, any pressure fluctuation across the valve or air damper changes the flow, even though the opening has not changed, and this in turn affects the combustion process and steam pressure. A better control scheme to avoid this type of disturbance, shown in Fig. 5-2.6, is referred to as full metering control. In this scheme the pressure controller sets the flow of fuel, and the airflow is ratioed from the fuel flow. The flow loops correct for any flow disturbances. The fuel/air ratio is still adjusted manually.

Ratio Control



Ratio Control



(b)

Ratio Control

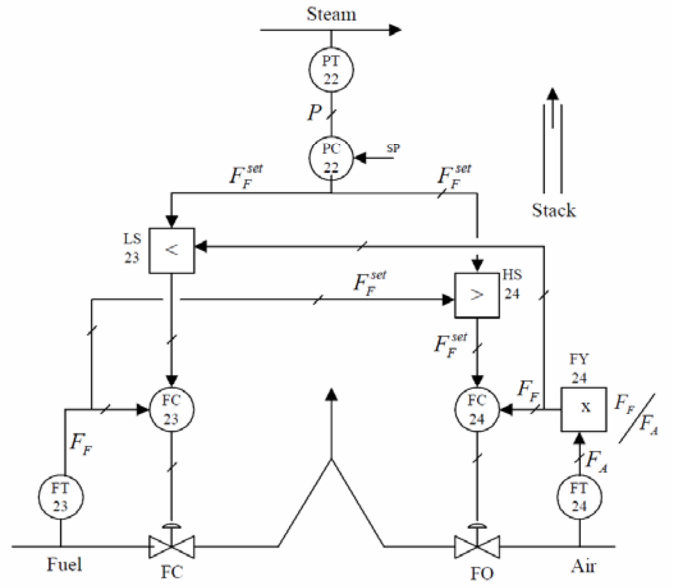


Figure 5-2.7 Cross-limiting control.

Ratio Control

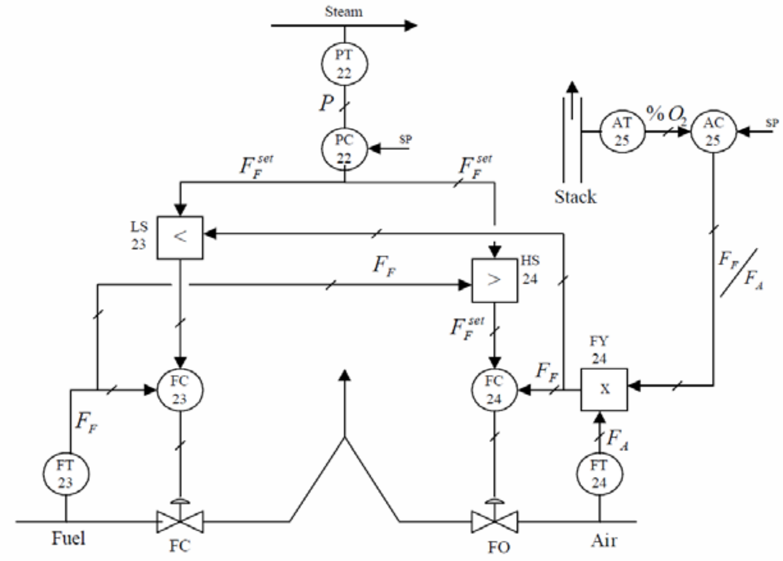


Figure 5-2.8 Cross-limiting with O_2 trim control.

Ratio Control

```
P = AIN(1, $P_{low}$, $P_{span}$)  ;  reads in pressure
FA = AIN(2, $FA_{low}$, $FA_{span}$)  ;  reads in air flow
FF = AIN(3, $FF_{low}$, $FF_{span}$)  ; reads in fuel flow
%O2 = AIN(4, $\%O2_{low}$, $\%O2_{span}$)  ; reads in $\%O_2$
FOUT = PID(P, $P^{set}$, $P_{low}$, $P_{span}$)  ;  PC22
ROUT = PID(%O2, $\%O2^{set}$, $\%O2_{low}$, $\%O2_{span}$)  ;  AC25
$PFF^{set}$ = ($FF_{span}$/100)*FOUT + $FF_{low}$  ;  converts output of PC22 to
                                  ;  fuel flow set point in engineering units
RATIO = ($RATIO_{span}$/100)*ROUT + $RATIO_{low}$  ;  converts output
                                  ;of AC25 to FA/FF ratio in engineering units
RFF = FA*RATIO  ;  FY24
IF $PFF^{set}$ < RFF THEN $FF^{set}$ = $PFF^{set}$  ELSE $FF^{set}$ = RFF  ;  LS23
COFUEL = PID(FF, $FF^{set}$, $FF_{low}$, $FF_{span}$)  ;  FC23
IF $PFF^{set}$ > FF THEN $FF^{set}$ = $PFF^{set}$  ELSE $FF^{set}$ =FF  ;  HS24
COAIR = PID(RFF, $FF^{set}$, $FF_{low}$, $FF_{span}$)  ;  FC24
AOUT(1, COFUEL)  ;  outputs signal to fuel valve
AOUT(2, COAIR)  ;  outputs signal to air valve
```

Figure 5-2.9 Software program equivalent to Fig. 5-2.8.

Override, or Constraint, Control

Override, or constraint, control is a powerful yet simple control strategy generally used as a protective strategy to maintain process variables within limits that must be enforced to ensure the safety of personnel and equipment and product quality. As a protective strategy, override control is not as drastic as interlock control. Interlock controls are used primarily to protect against equipment malfunction. When a malfunction is detected, the interlock system usually shuts the process down. Interlock systems are not presented, but Refs. 5 and 6 are provided for their study. Two examples of constraint control are now presented to demonstrate the concept and implementation of the strategy.

Override, or Constraint, Control

Example 5-3.1. Consider the process shown in Fig. 5-3.1. A hot saturated liquid enters a tank and from there is pumped under flow control back to the process. Under normal operation the level in the tank is at height h_1. If, under any circumstances, the liquid level drops below the height h_2, the liquid will not have enough net positive suction head (NPSH), and cavitation at the pump will result. It is therefore necessary to design a control scheme that avoids this condition. This new control scheme is shown in Fig. 5-3.2.

Override, or Constraint, Control

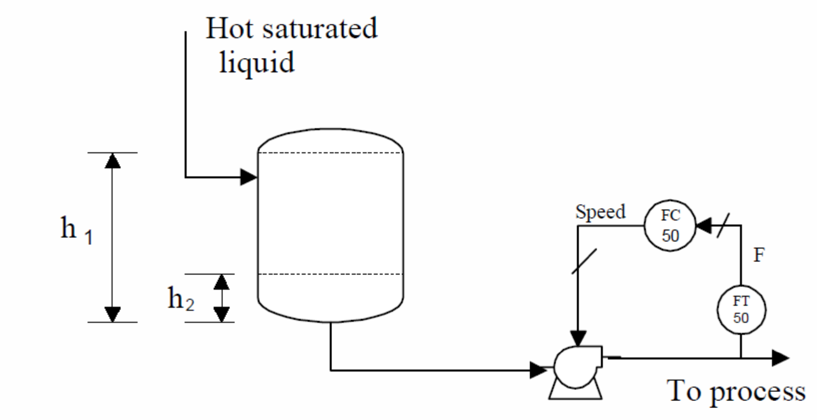


Figure 5-3.1 Tank and flow control loop.

Override, or Constraint, Control

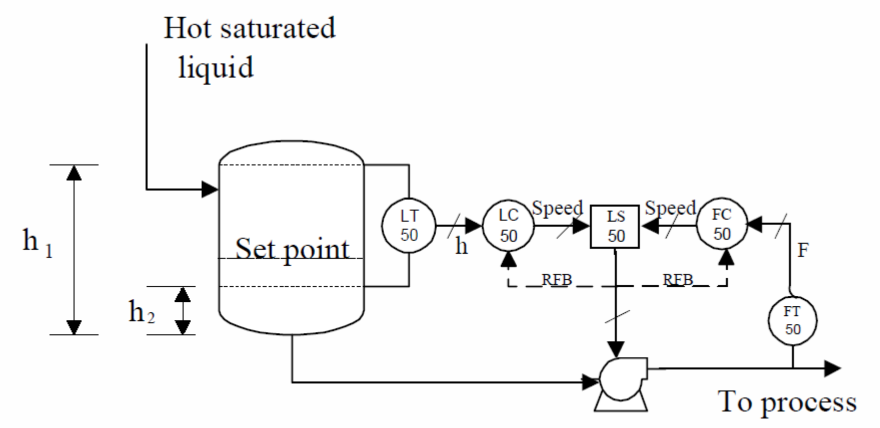


Figure 5-3.2 Override control scheme.

Override, or Constraint, Control

Example 5-3.2. A fired heater, or furnace, is another common process that requires the implementation of constraint control. Figure 5-3.3 shows a heater with temperature control manipulating the gas fuel flow. The manipulation of the combustion air has been omitted to simplify the diagram; however, it is the same as discussed in a previous example. There are several conditions in this heater that can prove quite hazardous. Some of these conditions are higher fuel pressure, which can sustain a stable flame, and higher stack, or tube, temperature than the equipment can safely handle. If either of these conditions exist, the gas fuel flow must decrease to avoid the unsafe condition; at this moment, temperature control is certainly not as important as the safety of the operation. Only when the unsafe conditions disappear is it permissible to return to straight temperature control.

Override, or Constraint, Control

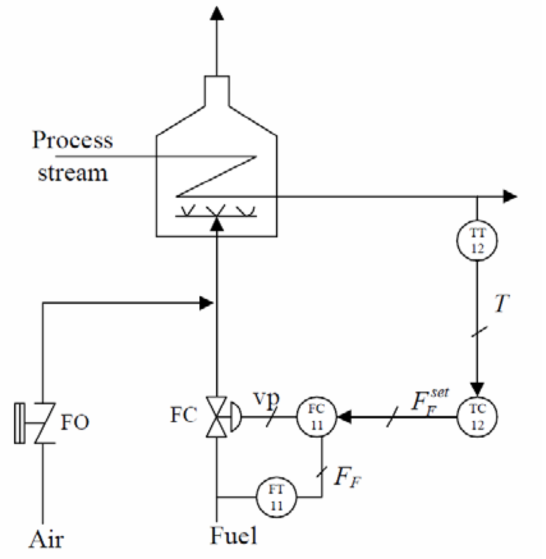


Figure 5-3.3 Heater temperature control.

Override, or Constraint, Control

Figure 5-3.4 shows a constraint control strategy to guard against the unsafe condition described above. The gas fuel pressure is usually below the set point to PC14, and consequently, the controller will try to raise the set point to the fuel flow controller. The stack temperature will also usually be below the set point to TC13, and consequently, the controller will try to raise the set point to the fuel flow controller. Thus, under normal conditions the exit heater temperature controller would be the controller selected by the low selector because its output will be the lowest of the three controllers. Only when one of the unsafe conditions exist would TC12 be “overridden” by one of the other controllers.

Override, or Constraint, Control

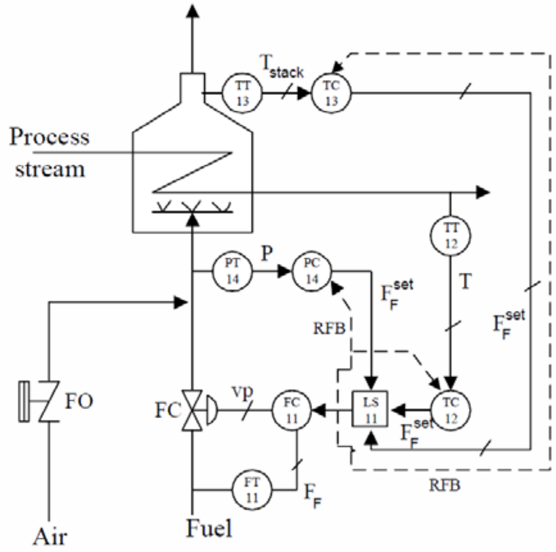


Figure 5-3.4 Heater temperature control, constraint control.

Selective Control

Selective control is another interesting control scheme used for safety considerations and process optimization. Two examples are presented to show its principles and implementation.

Selective Control

Example 5-4.1. Figure 5-4.1 shows a plug flow reactor where an exothermic catalytic reaction takes place; the figure also shows the reactor temperature control. The sensor providing the temperature measurement should be located at the "hot spot." As the catalyst in the reactor ages, or conditions change, the hot spot will move. It is desired to design a control scheme so that its measured variable "moves" as the hot spot moves. A control strategy that accomplishes the desired specifications is shown in Fig. 5-4.2. The high selector in this scheme selects the transmitter with the highest output, and in so doing the controlled variable is always the highest, or closest to the highest, temperature.

Selective Control

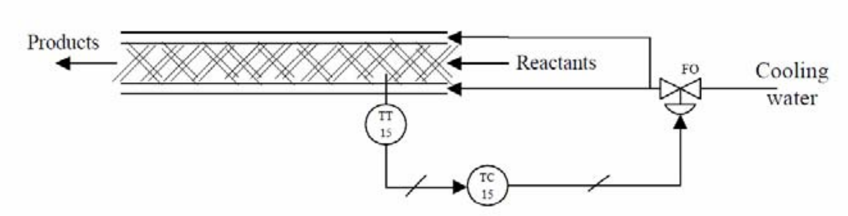


Figure 5-4.1 Temperature control of a plug flow reactor.

Selective Control

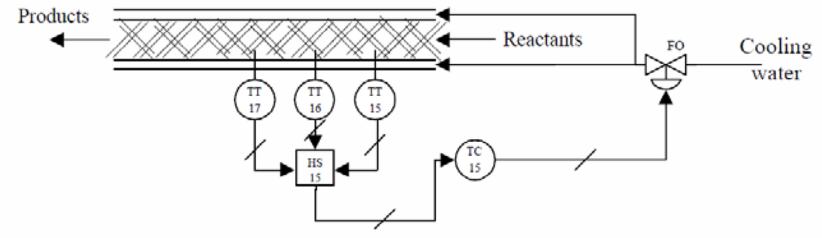


Figure 5-4.2 Selective control for a plug flow reactor.

Selective Control

Example 5-4.2. An instructive and realistic process where selective control can improve the operation is shown in Fig. 5-4.3. A furnace heats a heat transfer oil to provide an energy source to several process units. Each individual unit manipulates the flow of oil required to maintain its controlled variable at set point. The outlet oil temperature from the furnace is also controlled by manipulating the fuel flow. A bypass control loop, DPC16, is provided.

Selective Control

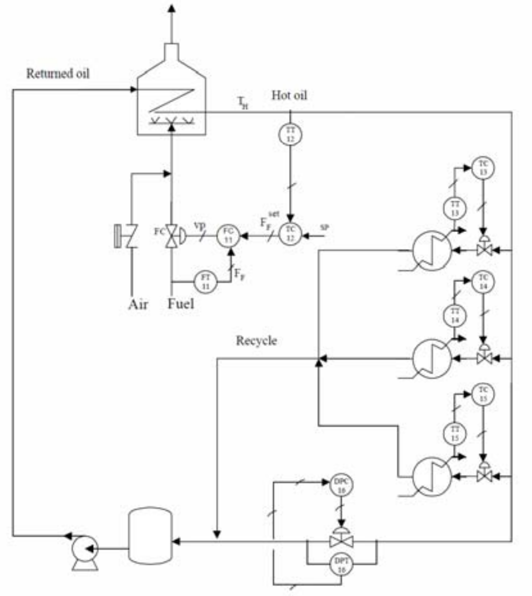


Figure 5-4.3 Hot oil system.

Selective Control

Suppose that it is noticed that the control valve in each unit is not open very much. For example, suppose that the output of TC13 is only 20%, that of TC14 is 15%, and that of TC15 is only 30%. This indicates that the hot oil temperature provided by the furnace may be higher than required by the users. Consequently, not much oil flow is necessary and much of it will bypass the users. This situation is energy inefficient since to obtain a high oil temperature, a large quantity of fuel must be burned. Also, a significant amount of the energy provided by the fuel is lost to the surroundings in the piping system and through the stack gases.

Selective Control

A more efficient operation is the one that maintains the oil leaving the furnace at a temperature just hot enough to provide the necessary energy to the users, with hardly any flow through the bypass valve. In this case the temperature control valves would generally be open. Figure 5-4.4 shows a selective control strategy that provides this type of operation. The strategy first selects the most open valve using a high selector, HS16. The valve position controller, VPC16, controls the valve position selected, say at 90% open, by manipulating the set point of the furnace temperature controller. Thus this strategy ensures that the oil temperature from the furnace is just “hot enough.” Note that since the most open valve is selected by comparing the signals to each valve, all the valves should have the same characteristics.

The selective control strategy shows again that with a bit of logic, a process operation can be improved significantly.

Selective Control

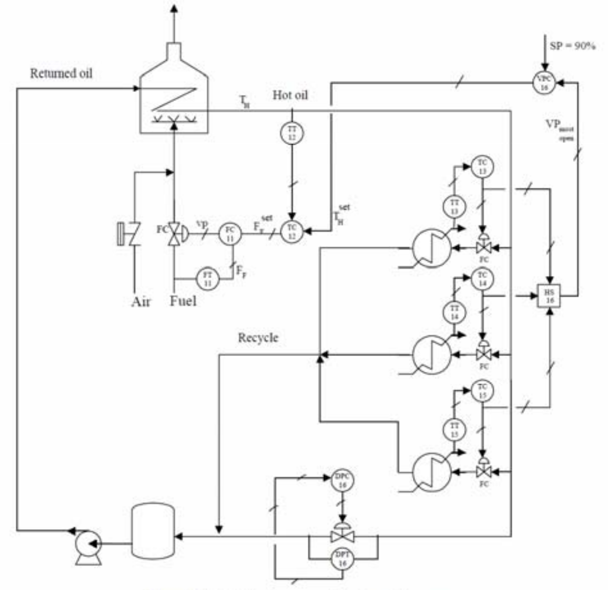


Figure 5-4.4 Selective control for hot oil system.

Designing Control Systems

In this section we present three examples to provide some hints on how to go about designing control schemes. To obtain maximum benefit from this section, we recommend that you first read the example statement and try to solve the problem by yourself. Then check with the solution presented.

Example 5-5.1. Consider the reactor shown in Fig. 5-5.1, where the exothermic reaction A + B → C takes place. The diagram shows the control of the temperature in the reactor by manipulating the cooling water valve.

(a) Design a control scheme to control the flow of reactants to the reactor. The flows of reactants A and B enter the reactor at a certain ratio R; that is, $R = F_B/F_A$. Both flows can be measured and controlled.

Designing Control Systems

(b) Operating experience has shown that the inlet cooling water temperature varies somewhat. Because of the lags in the system, this disturbance usually results in cycling the temperature in the reactor. The engineer in charge of this unit has been wondering whether some other control scheme can help in improving the temperature control. Design a control scheme to help him.

(c) Operating experience has also shown that under infrequent conditions the cooling system does not provide enough cooling. In this case the only way to control the temperature is by reducing the flow of reactants. Design a control scheme to do this automatically. The scheme must be such that when the cooling capacity returns to normal, the scheme of part (b) is reestablished.

Designing Control Systems

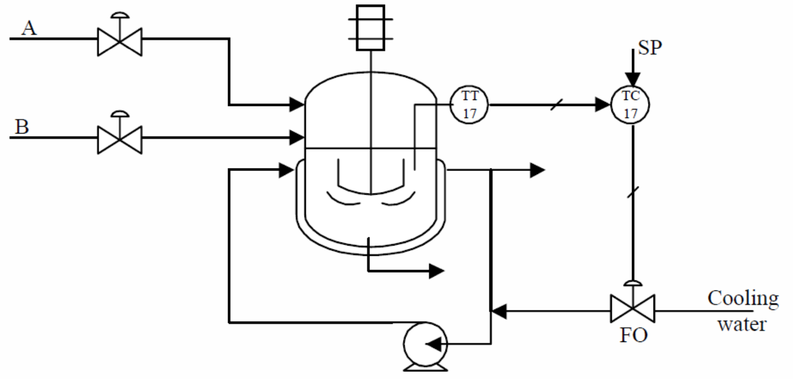


Figure 5-5.1 Reactor for Example 5-5.1.

Designing Control Systems

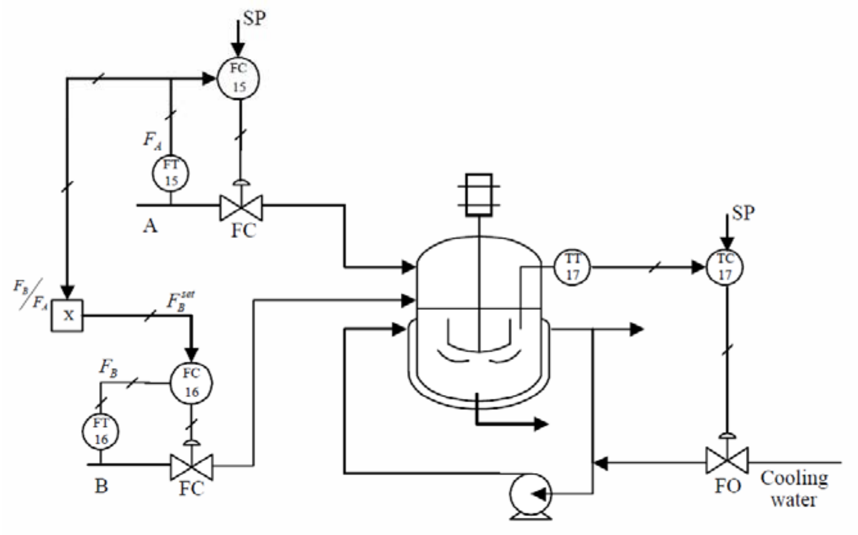


Figure 5-5.2 Ratio control scheme for part (a) of Example 5-5.1.

Designing Control Systems

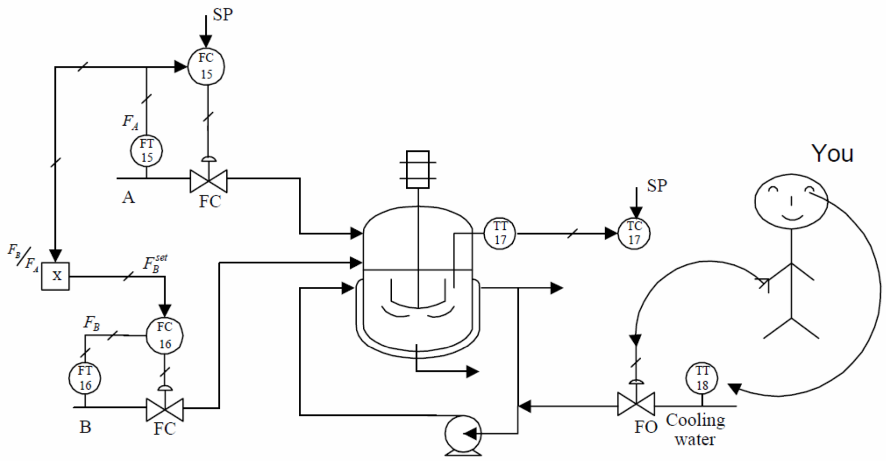


Figure 5-5.3 Proposed manual control scheme (first draft) to compensate for changes in inlet cooling water temperature.

Designing Control Systems

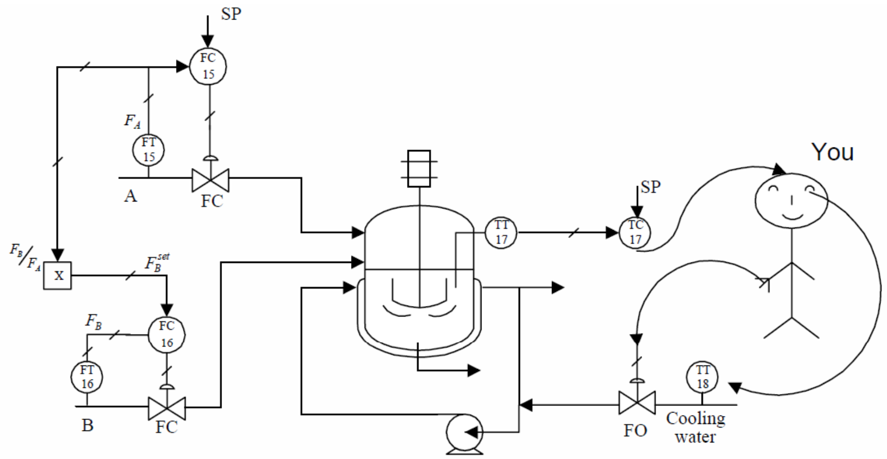


Figure 5-5.4 Proposed manual control scheme (second draft) to compensate for changes in inlet cooling water temperature.

Designing Control Systems

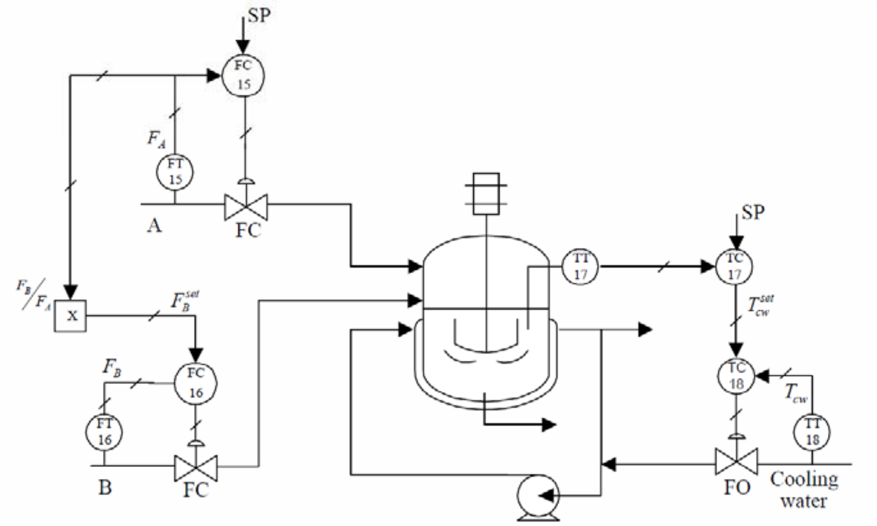


Figure 5-5.5 Proposed automatic control scheme (first draft) to compensate for changes in inlet cooling water temperature.

Designing Control Systems

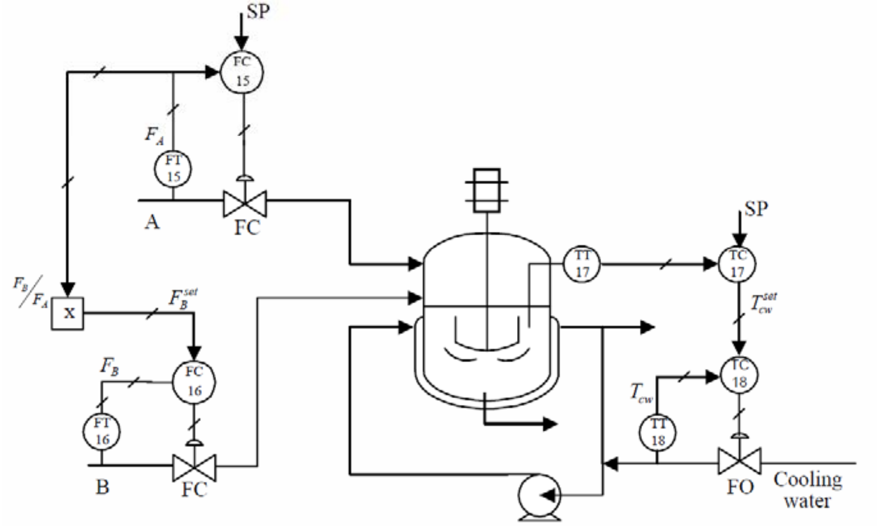


Figure 5-5.6 Proposed automatic control scheme (second draft) to compensate for changes in inlet cooling water temperature.

Designing Control Systems

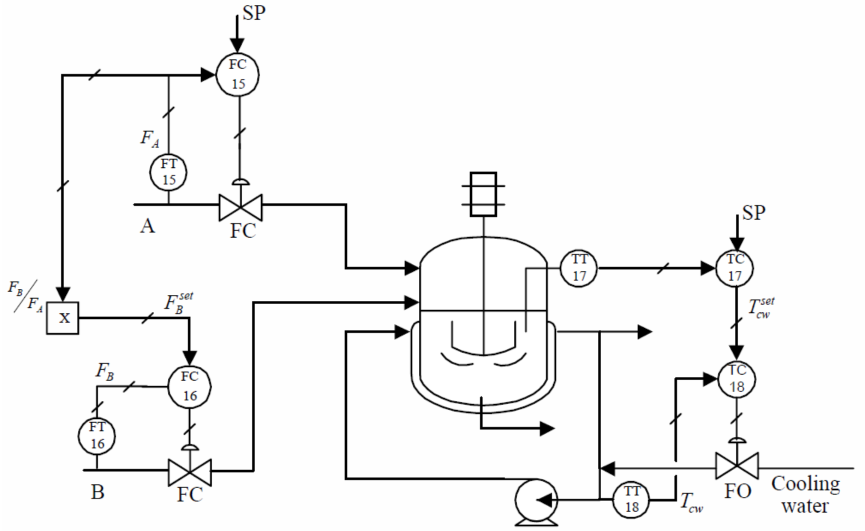


Figure 5-5.7 Cascade control scheme.

Designing Control Systems

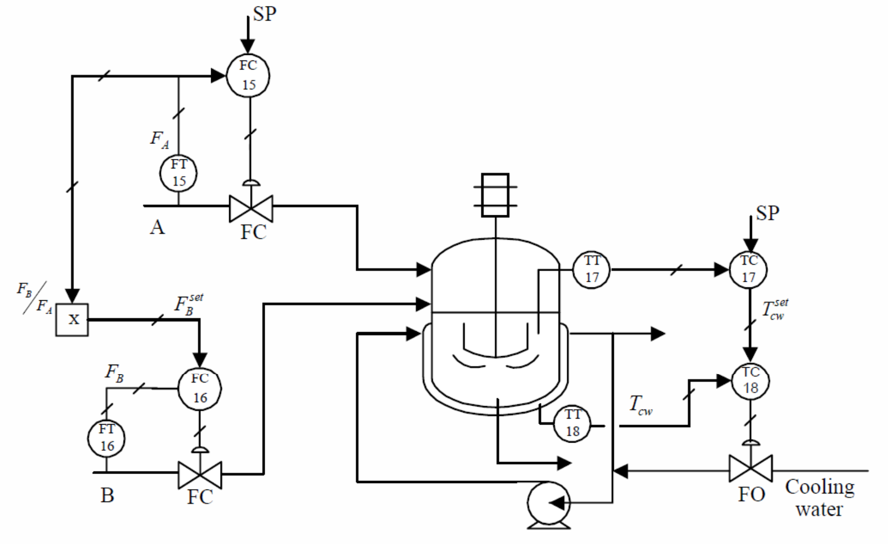


Figure 5-5.8 Cascade control scheme.

Designing Control Systems

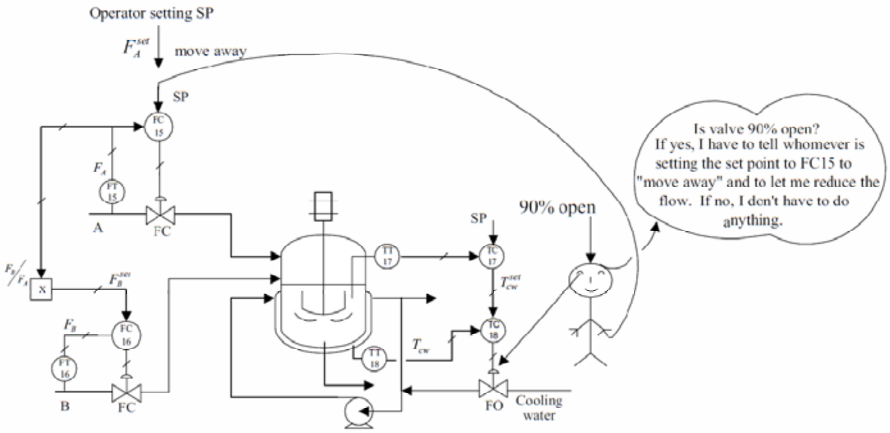


Figure 5-5.9 Avoiding the cooling valve going over 90% open, manual control.

Designing Control Systems

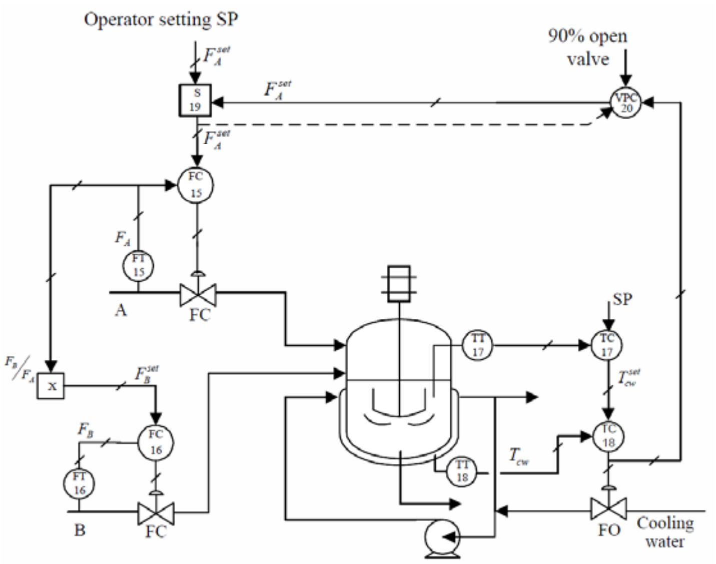


Figure 5-5.10 Override control scheme to compensate for loss of cooling capacity.

Designing Control Systems

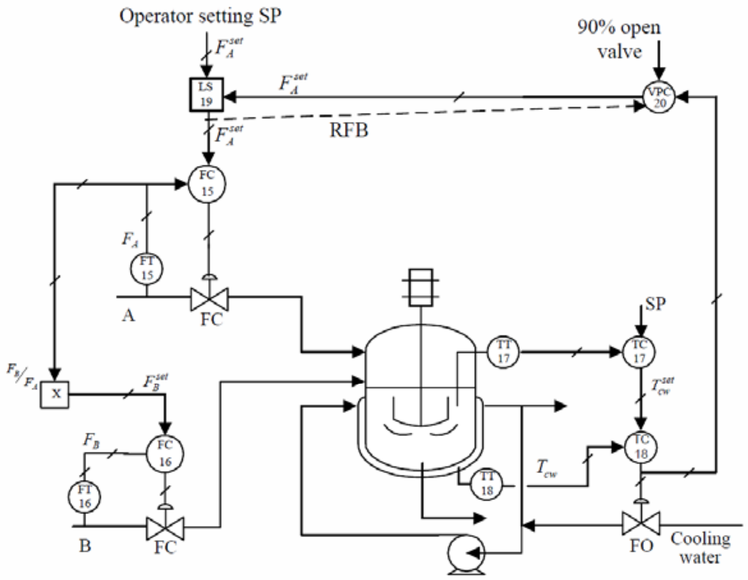


Figure 5-5.11 Override control scheme to compensate for loss of cooling capacity.

Designing Control Systems

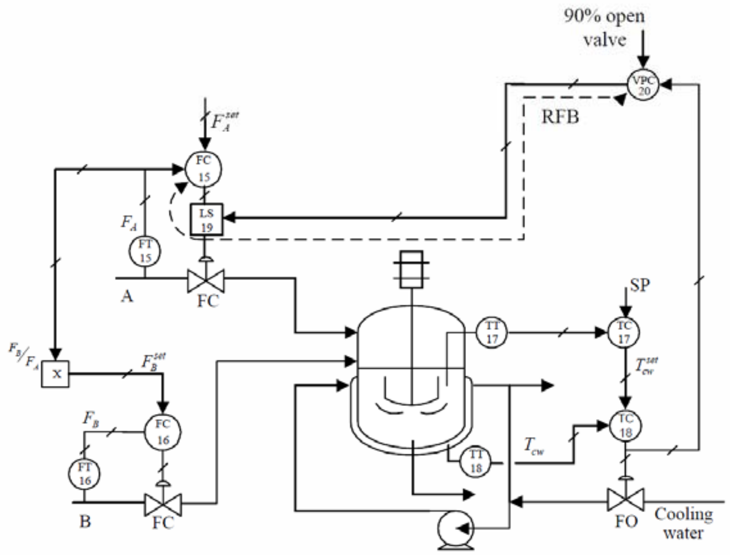


Figure 5-5.12 Another override control scheme.

Designing Control Systems

Example 5-5.2. Consider the reactor shown in Fig. 5-5.15, in which the irreversible and complete liquid reaction $A + B \rightarrow C$ occurs. Product C is the raw material for several process units downstream from the reactor. Reactant A is available from two sources. Because of a long-term contract, source 1 is less expensive than source 2. However, the contract is written with two limitations: a maximum instantaneous rate of 100 gpm and a maximum monthly consumption of 3.744 ¥ 106 gal. If either of these limitations is exceeded, a very high penalty must be paid, and thus it is less expensive to use the excess from source 2. For example, if 120 gpm of reactant A is required, 100gpm should come from source 1 and the other 20 gpm from source 2. Similarly, if on day 27 of the month 3.744 ¥ 106 gal have been obtained from source 1, from then on, until the end of the month, all of reactant A should come from source 2. You may assume that the densities of each reactant, A and B, and of product C do not vary much and therefore can be assumed constant.

Designing Control Systems

(a) Design a control system that will preferentially use reactant A from source1 and will not allow us to exceed contractual limitations. The feed ratio of A to B is 2 : 1 in gpm units.

(b) A few weeks after the control strategy designed in part (a) was put into operation, it was noticed that for some unknown reason, the supply pressure from source 2 was cut by the supplier every once in a while. Thus the flow controller manipulating the flow from source 2 would have to open the valve, and in some instances the valve would go wide open. At this moment there would not be enough flow from source 2 to satisfy the demand. It was decided that the correct action to take in this case, while the lawyers investigate—which may take a long time—is to obtain from source 1 whatever source 2 does not supply. Design a control strategy to accomplish this action. Be sure that your design is such that whenever source 2 provides the flow required, the scheme designed in part (a) is in effect.

Designing Control Systems

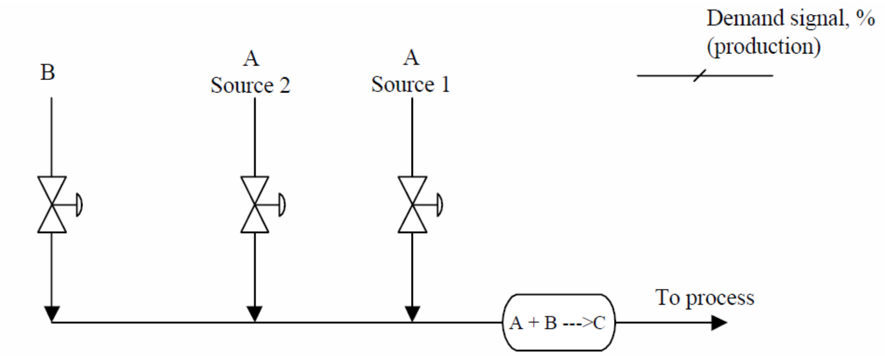


Figure 5-5.15 Reactor for Example 5-5.2.

Designing Control Systems

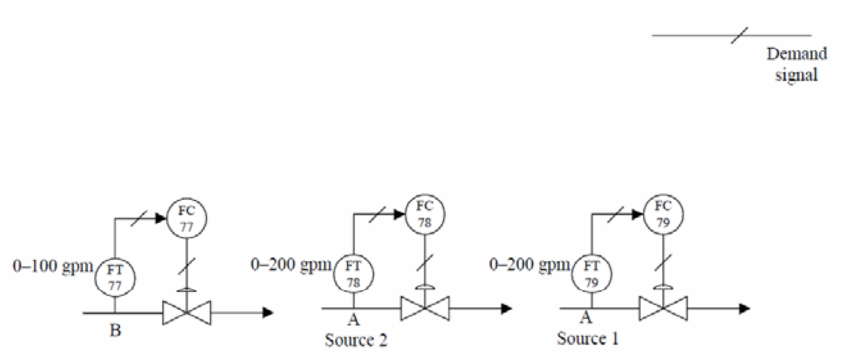


Figure 5-5.16 Flow loop installed in each stream.

Designing Control Systems

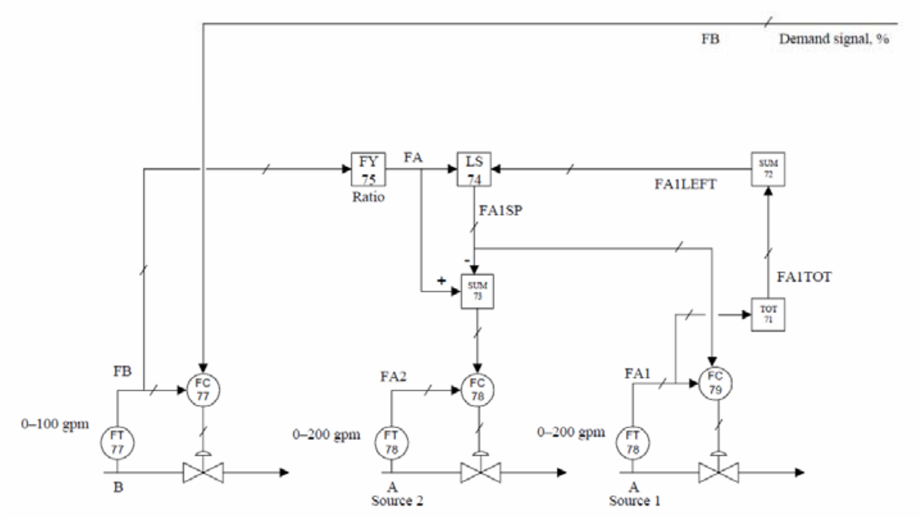


Figure 5-5.17 Control scheme for Example 5-5.2.

Designing Control Systems

Example 5-5.3. Consider the reactor shown in Fig. 5-5.18, where stream A reacts with water. Stream A can be measured but not manipulated. This stream is the byproduct of another unit. The water enters the reactor in two different forms, as liquid and as steam. The steam is used to heat the reactor contents. It is necessary to maintain a certain ratio R between the total water and stream A into the reactor. It is also necessary to control the temperature in the reactor. It is important to maintain the ratio of total flow of water to flow of stream A below a value Y; otherwise, a very thick polymer may be produced, plugging the reactor.

Designing Control Systems

A situation has occurred several times during extended periods of time in which the flow of stream A is reduced significantly. In this case the control scheme totally cuts the liquid water flow to the reactor to maintain the ratio. However, the steam flow to the reactor, to maintain temperature, still provides more water than required, and thus the actual ratio of water to stream A entering the reactor dangerously approaches Y. Design a control scheme to control the temperature in the reactor and another scheme to maintain the ratio of total water to stream A, while avoiding reaching the value of Y even if it means that the temperature deviates from the set point.

Designing Control Systems

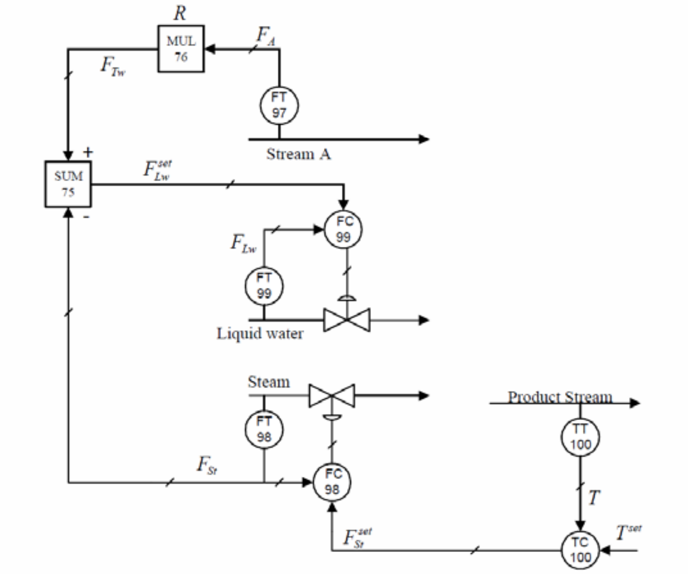


Figure 5-5.19 Temperature and ratio control for reactor of Example 5-5.3.

Designing Control Systems

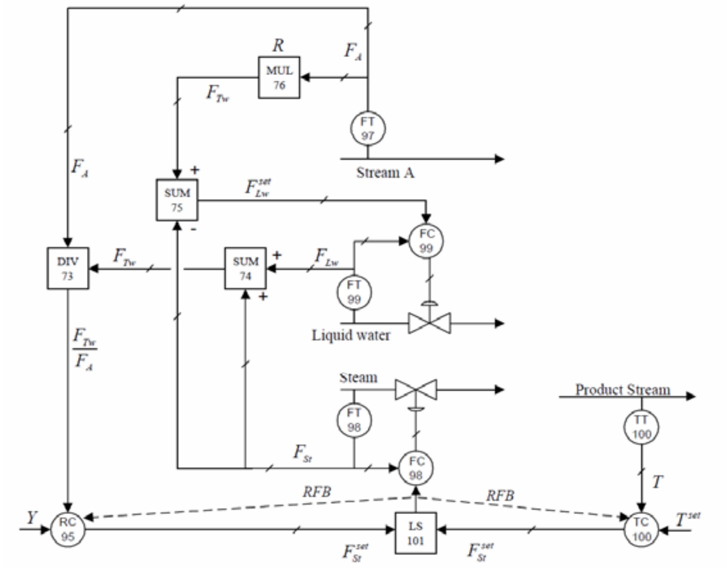


Figure 5-5.20 Complete control scheme for reactor of Example 5-5.3.

Designing Control Systems

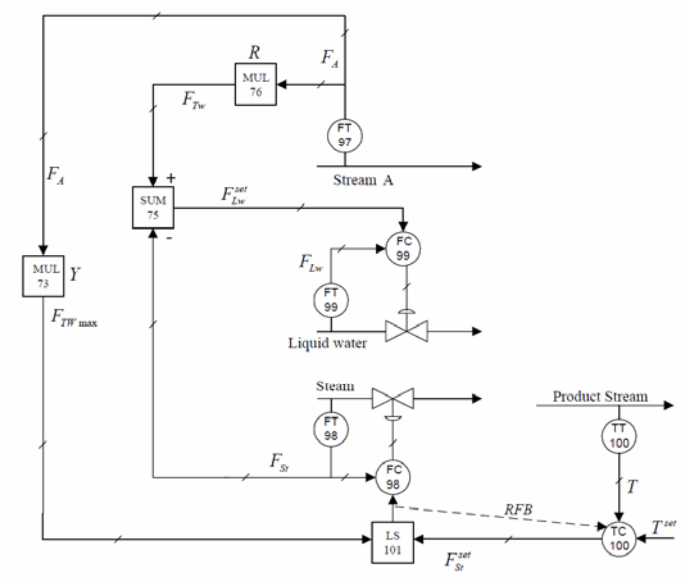


Figure 5-5.21 Another complete control scheme for reactor of Example 5-5.3.

Summary

In this chapter we have introduced the computation tools provided by manufacturers. An explanation for the need for scaling was given. A brief discussion of the significance, and importance, of field signals was also presented. We also presented the concepts, and applications, of ratio control, override control, and selective control. These techniques provide a realistic and simple method for improving process safety, product quality, and process operation. Finally, the chapter concluded with three examples, to provide some hints on the design of control schemes.

References

1. Automated Continuous Process Control, Carlos A. Smith, 2002, Wiley-Interscience, ISBN: 978-0471215783.
2. C. A. Smith and A. B. Corripio, Principles and Practice of Automatic Process Control, 3rd ed., Wiley, New York, 2006.