Fuzzy Supervisory Control

Jan Jantzen <jj@iau.dtu.dk>[1]

Abstract

Control problems in the process industry are dominated by non-linear and time-varying behaviour, many inner loops, and much interaction between the control loops. Fuzzy controllers have in some cases nevertheless mimicked the control actions of a human operator. For high level control and supervisory control several simple controllers can be combined in a priority hierarchy such as the one developed in the cement industry. An example of a distillation column shut-down illustrates a design procedure.

Contents

[1] Technical University of Denmark, Department of Automation, Bldg 326, DK-2800 Lyngby, DENMARK.
Tech. report no 98-H-875 (proc), November 11, 1998

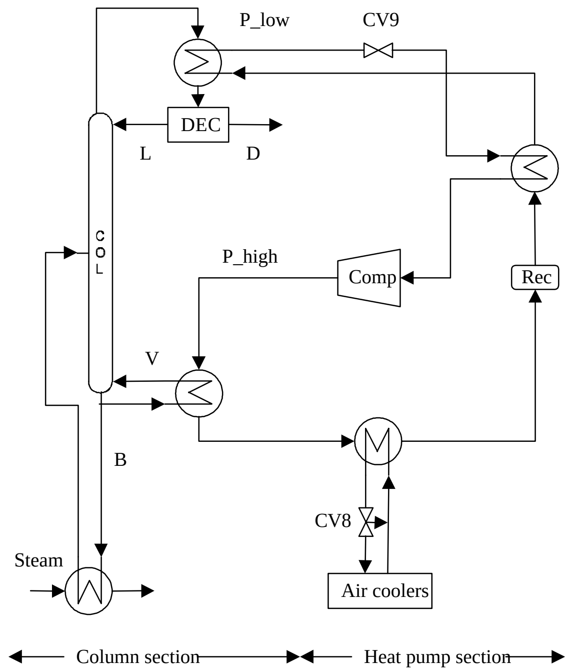

Figure 1: Distillation column equipped with a heat pump (right side of the figure) for recycling the condenser heat.

1. Introduction

By *process control* we shall understand the automation of a large-scale, industrial plant, so complex that it is impossible to achieve the complete satisfaction of a particular control specification. Typical examples are the control of distillation columns, the glass, cement, and plastic production, and electric power plants. Typical goals for a *supervisory controller* are safe operation, highest product quality, and most economic operation. All three goals are usually impossible to achieve simultaneously, so they must be prioritised; presumably, safety gets the highest priority.

Definition (Yazdi, 1997) A supervisory system is a system that evaluates whether local controllers satisfy prespecified performance criteria, diagnoses causes for deviation from the performance criteria, plans actions, and executes the planned actions.

□

Consider a distillation column with a thermosiphon *reboiler*, a total *condenser*, and an *accumulator* (Fig. 1). An indirect *heat pump* (Comp) is used to recycle energy from the condenser to the reboiler. To preheat the feed flow F and allow for heat input to the plant during start-up, a steam *preheater* is used. The feed is separated into a *top product* with the flow rate D and a *bottom product* with a flow rate of B; after reboiling some of the bottom product is fed back into the column with a vapour rate of V. Some of the top product, the *reflux*, is also fed back into the column, with a flow rate of L. The column is an experimental column, about 10 meters high, located at the Technical University of Denmark. The low and high pressure parts of the heat pump (P_{high} and P_{low}) are controlled by local PID controllers, likewise some of the liquid flow rates and the steam flow. A supervisory control system is responsible for sending proper setpoints to the PID controllers. During start-up, the column and the heat pump must be kept within certain pressure limits. The most critical decisions are when to start the heat pump, when to stop the external heat supply, and when to increase the capacity of the heat pump.

Start-up and shut-down are described in a manual (Andersen et al. in Yazdi, 1997), and normally executed by an operator. Two typical instructions from the shut-down are:

> *Feed- and product streams*. Stop the feed streams through PF1 and PF2. A high recycling rate through FM3 and FM2 flowmeters is established by opening BV48. Change the feed stage to the top of the column. Stop the bottom product outlet. Stop the steam supply to HEFS by closing the steam entry valve. Top product outlet is closed ...
>
> *Heat pump circuit*. The P_Low and P_High controllers are set to manual mode. CV9 is opened fully and CV8 is set for maximum cooling ...

Plain text instructions have several disadvantages: 1) Ambiguity is difficult to avoid, and operators may misinterpret the instructions, especially regarding task timing and synchronization; 2) keeping an overview of the running plant is difficult, and 3) it is insufficient for implementation, maintenance or documentation purposes.

The objective here is to describe a more organised procedure for supervisory control, especially with regard to start-up and shut-down of a plant.

Due to sheer complexity it is impossible, or at least very expensive, to build a mathematical model of the plant, and furthermore the control is normally a combination of sequential, parallel, and feedback control actions. Operators, however, are able to control complicated plants from experience and training, and thus fuzzy control becomes a relevant method within supervisory control. The control of cement kilns, developed by F. L. Smidth (FLS) in Denmark (Holmblad & Ostergaard, 1982), was inspired from Mamdani's early experiments with rule based control of a steam engine (Assilian & Mamdani, 1974) which, in combination with the rules of thumbs for manual control of a cement kiln (Peray & Wadell, 1972) resulted in the first high level process control strategy based on fuzzy logic. The control instructions for the cement kiln operators use the everyday terms LOW, OK, SLIGHTLY etc., which also are some of the basic primary terms that are represented by membership functions in the theory of fuzzy sets.

There is theoretical work, the so-called *Supervisory Control Theory* (Ramadge & Wonham in Yazdi, 1997), based on finite state machines. The supervisor is here viewed as a controller that restricts the plant behaviour to a subset of all possible behaviours by starting or stopping events. To avoid generating the (large) space of possible behaviours, a more compact logic based approach can be applied. *Petri nets* for discrete event systems allow intuitive modelling of the plant, see the survey by David & Alla (1992). The Petri net is the basis for the so-called *Grafcet*, which is a graphical language for modelling sequential controllers (David, 1995). A version of Grafcet, *Sequential Function Charts*, has been adopted as an international standard (IEC, 1988) widely used in connection with programmable logic controllers (PLCs). Another modelling technique also associated with a graphical language, *Multi-level Flow Modelling*, requires defining explicit goals for each task and the means to reach those goals (Lind in Yazdi, 1997). A recent PhD project (Yazdi, 1997) resulted in a procedure for designing supervisory control systems by combining ideas from Grafcet, Multi-level Flow Modelling, and fuzzy rules. Much of the following is based on that thesis.

During plant start-up the operator performs actions based on his knowledge of the components and how they interact. Some operator actions can be categorised as follows.

Binary actions. Changes in the structure of the plant and switching to other plant configurations. Examples are on/off valves and providing power to a pump drive.

Prepare actions. Prepare the whole plant, or part of the plant, for closed loop control, with setpoints selected by the operator. An example is to start a pump in order to obtain a minimum flow rate of steam before switching to automatic control.

Control actions. Closed loop control around proper setpoints. An example is control of a steam flow rate, by automatically adjusting pump speed.

Corrective actions. Take action when a (foreseeable) malfunction occurs. An example is a servo valve, that does not function, because it sticks.

In order to design an automated start-up or shut-down, the problem is 1) to acquire the necessary operator's knowledge, 2) to structure the knowledge, and 3) to identify suitable descriptive tools. We will assume here that item 1) has been taken care of already, and concentrate on the last two items.

2. Design approach

Even though the complexity cannot be reduced, it is helpful to decompose a supervisory control system into hierarchical sub-systems. This may in fact be the only practical way to deal with complexity in large scale systems.

2.1 Control tasks

A start-up procedure consists of a number of sub-procedures or *tasks*, each having a specific purpose such as reaching a specific state of the plant. In general a complex start-up procedure consists of distinct *phases*, where the plant configuration remains unchanged, and each phase may contain several tasks. The changeover from a phase to the next requires terminating certain tasks of the one phase and initiating certain tasks of the following phase. Yazdi (1997) has defined a standard *control task* in terms of a set of necessary properties:

- *Name*. A name describing a task goal.

Case	Condition	Action to be taken	Reason
10	BZ OK	a. Increase I.D. fan speed	To raise back-end temperature and increase
	OX low		oxygen percentage for action 'b'
	BE low	b. Increase fuel rate	To maintain burning zone temperature
11	BZ OK	a. Decrease fuel rate slightly	To raise percentage of oxygen
	OX low		
	BE OK		
12	BZ OK	a. Reduce fuel rate	To increase percentage of oxygen for action 'b'
	OX low	b. Reduce I.D. fan speed	To lower back-end temperature and maintain
	BE high		burning zone temperature
13	BZ OK	a. Increase I.D. fan speed	To raise back-end temperature
	OX OK	b. Increase fuel rate	To maintain burning zone temperature
	BE low		
14	BZ OK	NONE. However, do not get	
	OX OK	overconfident, and keep all con-	
	BE OK	ditions under close observation.	

Table 1: Extract from textbook (Peray & Wadell, 1972) for kiln operators.

- *Goal*.
- *Strategic conditions*. A condition set which only has to be valid for initiating the task. The strategic condition set is normally a process requirement before the task starts.
- *Execution conditions*. A set of conditions which has to be valid during task execution. The condition set is normally related to the operational constraints, e.g. physical limitations and safety conditions. Information about these properties is derived by combining operator knowledge of the operational conditions with design knowledge.
- *Initial actions*. A set of actions which has to be carried out to prepare the task for control structuring. This property contains most binary actions (on/off, start/stop) during start-up.
- *Control actions*. A set of sensors and actuators through which proper control can be designed in order to achieve the task objective. This property describes the resources of the control function.
- *Achievement indicator*. An indicator for the degree of goal achievement during the task operation.
- *Final action*. A task can end with a set of final actions, according to the start-up procedure. Final action is initiated when the task goal has been achieved.

A control task, and a control task should be defined for each (fuzzy) controller. Each control task describes a detailed strategy for reaching a specific sub-goal. Execution of a control task is only allowed if *execution* conditions are valid at each sampling instant, while *strategic* conditions are checked only at the first sampling of the task.

2.2 High level fuzzy control

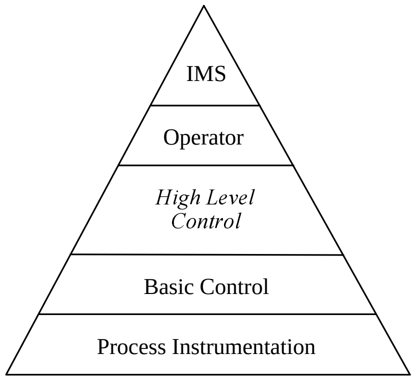

Figure 2: Process control pyramid.

It can be difficult, however, to combine all control tasks in order to achieve a total goal structure. The FL Smidth design procedure, associated with their second generation fuzzy controller (Fuzzy II, see Østergaard, 1990; 1996), includes for this reason a priority management system.

The obvious link between the rule based approach of fuzzy logic and supervisory control is illustrated by Table 1. The control instructions for the cement kiln operators use the everyday terms LOW, OK, SLIGHTLY etc., which also are some of the basic primary terms that are represented by membership functions in the fuzzy controller.

A *high level* controller works on the same level as the human operator. It takes over a part of, or all of the operator's job of controlling the process.

Definition (Østergaard, 1996) High level process control is the coordination of control loop set points, which normally is done by a human operator.

□

High level control thus introduces an extra layer between the operator and the conventional control system in the process control pyramid as shown in Fig. 2. The character of high level control is intervention, and as such a sub-set of supervisory control.

The basic idea of high level process control based on fuzzy logic is to formulate sets of rules for automatic operation based on practical experience and knowledge about manual control of the process. The terms of the rules, say, SMALL, OK, and HIGH, are represented by membership functions, and the rules are evaluated by an inference engine. Examples of fuzzy control rules are:

IF Temp is High and Pressure is Ok THEN Medium Flow

IF Temp is Ok and Pressure is Ok THEN Small Flow

In fuzzy logic, the condition of a rule is fulfilled to a certain degree, and each rule will influence the result of the set of rules in accordance with its grade of fulfillment..

This heuristic approach for design of control strategies for automatic process control is useful, and probably the only practical solution, when the process is only partly known, difficult to describe by a mathematical model, if few measurements are available, or if the process is highly nonlinear.

High level control expectations The overall expectations to the benefits are that the automated operation will result in better process performance than produced by manual control. It is, however, not always obvious what the meaning of improved process performance is, and it may even differ within the same process industry depending on local conditions such as the present market situation, raw material costs and overall strategic goals. In some countries reduced fuel consumption, for instance, is not as attractive as it is in countries where the price of fuel is high. In general, the expectations to a high level control system will be in comparison with manual control by an operator, or perhaps in comparison with an existing and more conventional control scheme.

In most cases, expectations will relate to profit in terms of reduced costs or increased productivity. Many industrial processes are very energy and raw material consuming. The glass industry, the cement industry, and power plants are examples of industries which consume large amounts of coal, oil, natural gas, or electrical energy. The plastic industry, for instance, and the processes for cleaning exhaust gases and waste water, use large amounts of raw materials which has a significant impact on the overall plant economy.

It is characteristic that the consumption of energy and raw materials depend very much on how the processes are being controlled. In general, improved operation can be defined as:

- More stable (i.e., steady) operation,
- running closer to the limits for acceptable product quality, and
- running closer to the environmental emission limits.

Stability, in fact, is the most important key to improve the operation of a process. Oscillations will always increase the consumption of energy and raw materials, and it will reduce the quality of the product. If the process is unsteady, it must operate with a safety margin to keep average values within the limits. Emission figures are sensitive to unsteady process operation.

Example 1 (TVD) *A simple way to measure stability is through the standard deviation STD of the key measurements and quality parameters. Depending on the type of industrial process, the standard deviation may be calculated on a daily basis, and an average standard deviation is then calculated for a period which is representative for the performance evaluation. If the measurements and the quality parameters have target values, then a more feasible measure of stability is obtained by calculating the target value deviation TVD around the target value SP, instead of the variations around the average value AVR as done by STD. TVD may be calculated from the standard deviation, the setpoint, and the average value by:*

$$TVD = \sqrt{STD^2 + (SP - AVR)^2} \tag{1}$$

Improved control of the process is one way to make the process run more stable. It is

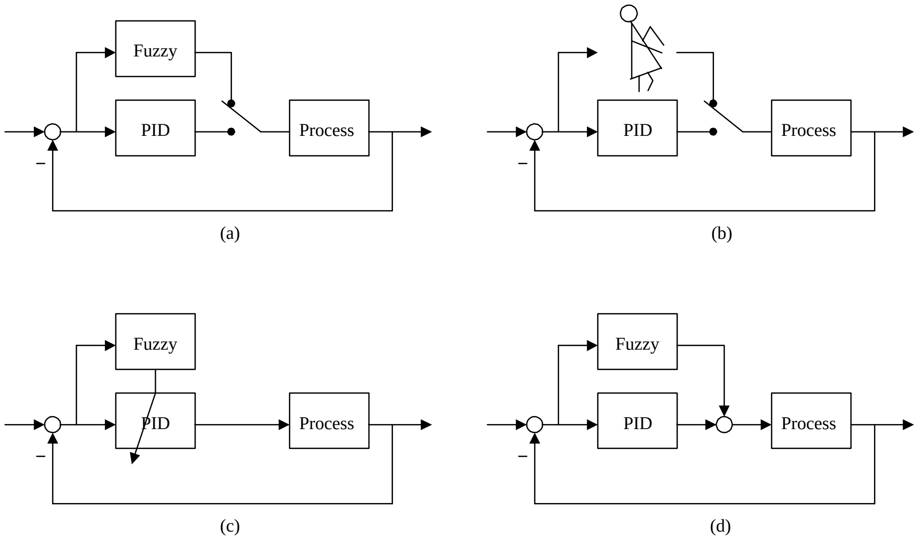

Figure 3: Fuzzy controller configurations. Fuzzy replaces PID (a), fuzzy replaces manual control (b), fuzzy adjusts PID parameters (c), and fuzzy adds to PID output.

realistic to expect reductions in STD *and/or* TVD *of up to 50% or more as the result of a good working high level control system. The improved stability may be utilized to operate the process at a higher production level, and to run it closer to the limits for safe operation and also closer to the limits for emissions to the environment.*

□

Example 2 (waste incineration plant) *A waste incineration furnace is an example of a process where stability has a direct impact on profitability. In a waste incineration furnace, the temperature levels must not exceed certain values as that will cause build-up of material on the furnace walls, and result in unnecessary wear on the brick lining, and reduce the amount of burned waste. In case of severe build-up, the furnace has to be stopped for cleaning. However, the more stable the control strategy can keep the temperatures, the closer it is possible to run the temperatures at the upper limits. Higher temperatures result in more waste to be burned, which directly will increase profit.*

□

High level control configurations Fuzzy controllers are integrated with other controllers in various configurations as shown in Fig. 3, where PID stands for a conventional control scheme which in most cases consists of independent or coupled PID loops. Fuzzy in Fig. 3 refers to a high level control strategy. Normally, both the PID- and the Fuzzy blocks have more than one input and one output.

Fig. 3 *(a)*. In this configuration, the operator may select between a high level control strategy and conventional control loops. Often the conventional loops represent an existing control scheme, which has been controlling the process before installation of the high level strategy. The operator has to decide which of the two alternatives is the most likely to produce the best control performance. Waste incineration furnaces are examples of processes which are equipped with an existing system of coupled PID loops for control of the charging of waste and the amount of combustion air, and here the high level control system by-passes the existing system when activated.

Fig. 3 *(b)*. This configuration represents the original high level control idea, where manual control carried out by a human operator is replaced by automatic control. Normally, the existing control loops are still active, and the high level control strategy makes adjustments of the controller set points in the same way as the operator does. Again it is up to the operator to decide whether manual or automatic control will result in the best possible operation of the process which, of course, may create conflicts.

Fig. 3 *(c)*. In this configuration, the high level strategy is used for adjustments of the parameters of the conventional control loops. A common problem with linear PID controllers used for control of highly nonlinear processes is that the set of controller parameters produces satisfactory performance only when the process is within a small operational window. Outside this window, other parameters or set points are necessary, and these adjustments may be done automatically by a high level strategy.

Fig. 3 *(d)*. Normally, conventional control systems which are based on PID controllers are capable of controlling the process when the operation is steady and close to normal conditions. However, if sudden changes occur or if the process enters abnormal situations, then this configuration may be useful to bring the process back to normal operation as fast as possible. For normal operation, the fuzzy contribution is zero, whereas the PID outputs are compensated in abnormal situations, often referred to as Abnormal Situation Management (ASM).

For all four configurations, the high level control system has an impact on the work done by the operator. Configuration (a) and (b) change directly the routines of the operator, which is a crucial point to take into account when the system is developed and installed.

The FLS design procedure For an operator, control of the process consists in achieving various goals, more or less precisely defined, such as maximum output, minimum consumption of raw materials and energy, high product quality, safe process operation. Different processes have different control objectives but, in general, good process control may be defined through a list of control objectives which should be fulfilled as much as possible. The concept of control objectives is a key element in a high level control strategy.

For a cement kiln, typical control objectives are:

- stable operation,
- good cement clinker quality,
- high production,
- complete combustion, and

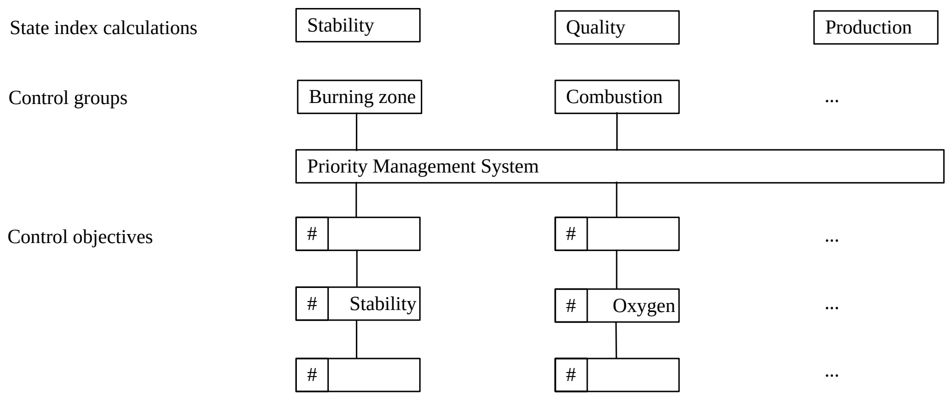

Figure 4: The FLS high level control strategy structure.

- low fuel consumption.

As control objectives are frequently in conflict, high level co-ordination means approaching optimal conditions in succession of importance. Priorities, in other words, have to be assigned to the various control objectives, specifying which objectives are considered the most important to fulfill. Control objectives with different priorities are thus the basic elements of the design of a high level control strategy.

The elements of the FLS design procedure for a process control strategy are the following (Østergaard, 1990; 1996):

State indices. Calculations concerning the actual process condition.
Control groups. Arrangements of the overall control strategy into groups of control objectives.
Priority management. Determines the extent to which the control actions should be executed to fulfil the individual objectives.
Control objectives. Specifications of the goals of the fuzzy control strategy.

Normally, a state index combines various measurements into a single figure. The degree of process stability, the product quality, and the production level are all typical examples of state indices for a kiln control strategy. The state indices are important to the structure of the FLS design scheme for a high level control strategy, as they form the basis for dividing the overall strategy into control groups which can be treated independently. The state indices are used to co-ordinate control actions from the various control groups, see Fig. 4.

The *state index calculations* are calculations of index values by which process measurements are combined into state indices for the actual process stability, product quality, production level, etc. *Control groups* form a subdivision of the control strategy into groups of objectives which are related through priority numbers. A *priority management system* manages the scheduling of control actions in order of importance. *Control objectives* specify the individual goals for the fuzzy control strategy.

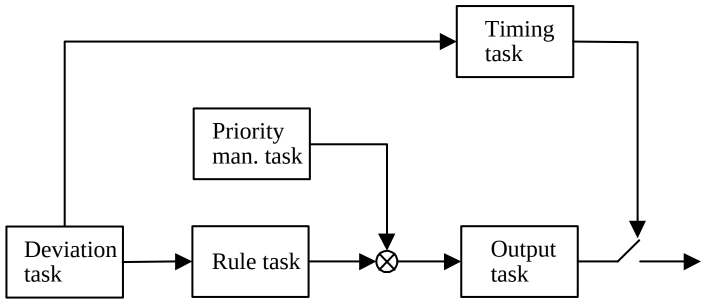

Figure 5: Control objective module in Fuzzy II.

Design of a control strategy consists of filling in the structure in Fig. 4 with specific state index calculations, control groups and control objectives, whereas the priority management system is a fixed module in the system. For a new process, it is perhaps not so clear which are the most appropriate indices and control groups. This is not a major problem, however, as the structure facilitates a stepwise implementation of control objectives and index calculations concurrent with increasing process control knowledge being available.

Every control objective is implemented in accordance with the so-called *objective module*, which consists of four tasks. Figure 5 shows the objective module diagram, which is used for implementation of all types of control objectives.

Deviation task. This task calculates and evaluates the degree to which the objective is fulfilled. Normally, the calculation results in a "fuzzy value" $e_i \in [-1, 1]$ which expresses how far the actual process situation is from fulfillment of the objective; a value of 0 means that the objective is fulfilled.

Rule task. This block holds the set of control rules for fulfillment of the specific objective. Normally, this block is formulated as a set of fuzzy control rules, and the output of the rule block is normally a change in action in the interval $[-1, 1]$. Other techniques may also be used, such as PID, neural nets, and mathematical models.

Priority management task. The rule block for each control objective results in control actions which are multiplied by a weight factor between 0 and 1. The weight factor w_i associated with objective i is calculated as

$$w_i = 1 - |e_i| \tag{2}$$

The weight factor is thus a function of the deviation e_i of the objective. The smaller the weight factor, the more the lower control actions are suppressed. The total weighting of an objective's output is the product of all higher priority weight factors (Fig. 6); it will be close to 1 if all objectives with a higher priority are highly fulfilled. The priorities represent built-in knowledge about optimal interaction of rule blocks.

Output task. The output program evaluates process constraints and selects among alternative control actions based upon the actual index values; it converts fuzzy out-

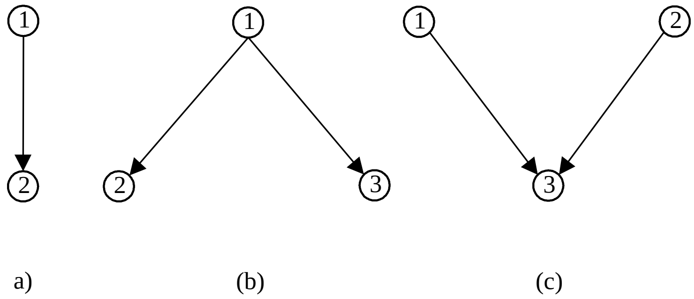

Figure 6: Priority management: a) objective 1 higher than 2; b) objective 1 affects two objectives on a lower level; c) objectives 1 and 2 both affect an objective on a lower level.

put to engineering units, that is, denormalised physical units. The logic for selecting alternative adjustments may be fuzzy or non-fuzzy depending on whether a gradual or a hard switch is the most appropriate. In most cases, the fuzzy logic approach gives the best control performance, simply because no process operates with sudden changes between alternative control actions.

Timing calculation task. The timing calculation determines when and how often control actions are to be executed. It is just as important as the rule block for determining the proper function of the control strategy. Also the timing calculation is normally fuzzy in the sense that the time interval between control actions changes gradually as a function of the deviation value. The larger the deviation, the more frequent the control actions.

Each objective has several tuning parameters: an output gain, input normalisation, and tuning of the timing calculation.

2.3 Grafcet

The priority system can be restrictive, if a design requires information about parallel and sequential execution as well as synchronisation. *Grafcet* stands for Graphe de Commande Etape / Transition-Step / Transition Control Chart. From a theoretical viewpoint Grafcet is a Petri net, that enables transition firings synchronised on external events (David & Alla, 1992). It has been proposed as an alternative to Ladder Relay Logic in PLC programming. A grafcet describes the top level control, and when the plant is in a particular state, a subprogram can decide what has to be done. The subprogram can itself be a grafcet.

A grafcet can contain both parallel and sequential paths (Fig. 7). Basic elements are *steps* and *transitions*. A step, drawn as a box with a number, represents a state and can be associated with actions. A block in the top of the diagram, associated with a zero, is the initiating step. A step can either be in *active* or *inactive* mode, a Boolean 0/1 evaluation determining the placement of a *token,* which shows that the step is *active*. The associated

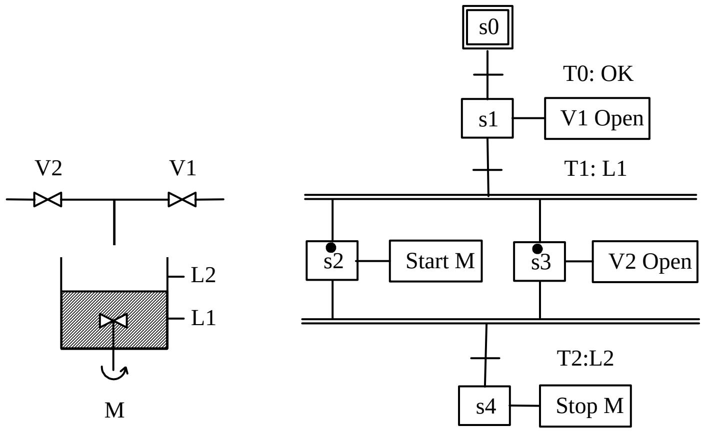

Figure 7: Mixing process described in a grafcet.

action is executed when the step is active. Actions are classified in two main types namely *level* and *impulse* actions.

> *Level Action.* An action that remains true for the active duration of the associated step. In other words, an actuator that is changed by a level action is changed back automatically to its initial value when the associated step become inactive.
> *Impulse Action.* A (command) action which is executed once the associated step state is changed form inactive to active. An actuator value that is changed by an impulse action, can only be changed back by an appropriate reverse action.

An impulse action may be said to be a *command,* e.g., 'close valve1', whereas a level action indicates a state, e.g., 'valve1 closed'. *Transitions* connect steps together and represent the conditions for activating/deactivating steps. A transition is *firable* if and only if all the steps preceding the transition are active and the *transition condition* (called *receptivity*) is true. An important rule about transitions is that several simultaneously firable transitions are fired simultaneously. The *marking* of a grafcet is a boolean vector representing the state of the grafcet, i.e., the activity of steps represented by tokens.

For complex systems *macrosteps* are available. A macrostep represents a grafcet structure, in one step, that is part of another grafcet. In addition, macroactions can be used to model the influence of one grafcet on another grafcet.

Example 3 (mixer) *Two components are mixed in a container (Fig. 7). A grafcet model of the control is shown in the right half of the figure. The mixer starts when the OK signal is generated by an operator. Step $s1$ becomes active and the level action $V1$ $Open$ becomes true. When the level of the tank reaches $L1$, step $s1$ becomes false and the value of $V1$ $Open$ is changed to false. Simultaneously, steps $s2$ and $s3$ are activated, shown by two tokens (black dots). The double lines are parallel branches or junctions, such that $s2$ and*

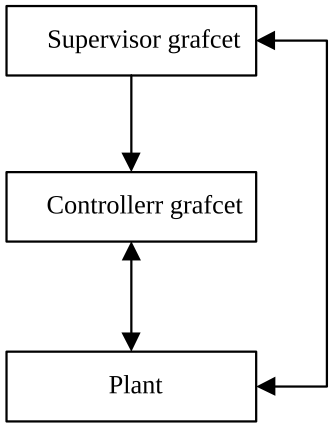

Figure 8: The supervisor affects the plant and the controller in case of abnormal operation.

$s3$ are active at the same time. The impulse action Start M *changes the value of the mixer M while the level action* V2 Open *associated with step $s3$ takes place. Note the difference between the impulse and level actions. When the tank level reaches $L2$, $s2$ and $s3$ are deactivated. This implies that $V2$ automatically is closed while the mixer M still is running. The difference is due to the different functions of level and impulse actions. When $s4$ is activated, another impulse action,* Stop M, *will change the state of the mixer from* running *to* stop.

□

Yazdi (1997) recommends to separate control and supervision (Fig. 8). A *controller grafcet* should represent the normal operation of the plant, while a *supervisor grafcet* should represent the corrective actions for the abnormal operation of the plant. The occurrence of an abnormal event triggers the supervisor.

3. Shut-down of a distillation column

Returning to the objective set out in the introductory section, this section concerns only the shut-down procedure for the distillation column, because the shut-down is easier and less complex than the start-up (see Yazdi, 1997).

The distillation plant consists of three sections, 1) the column, 2) the heat pump, and 3) a tank park. Figure 9 shows a diagram of the two first sections. The tank park is storage, and it can be set up with various types of piping configurations. The column separates a mixture of methanol and isopropanol with a slight water impurity. The heat pump recycles the condensation energy in HECOND to the reboiler HERB. The number of active cylinders in the compressors COMP can be adjusted in steps of 2 with a minimum of 4 working at a time and a maximum of 16. A number of PID regulators (Programmable Logic Controllers)

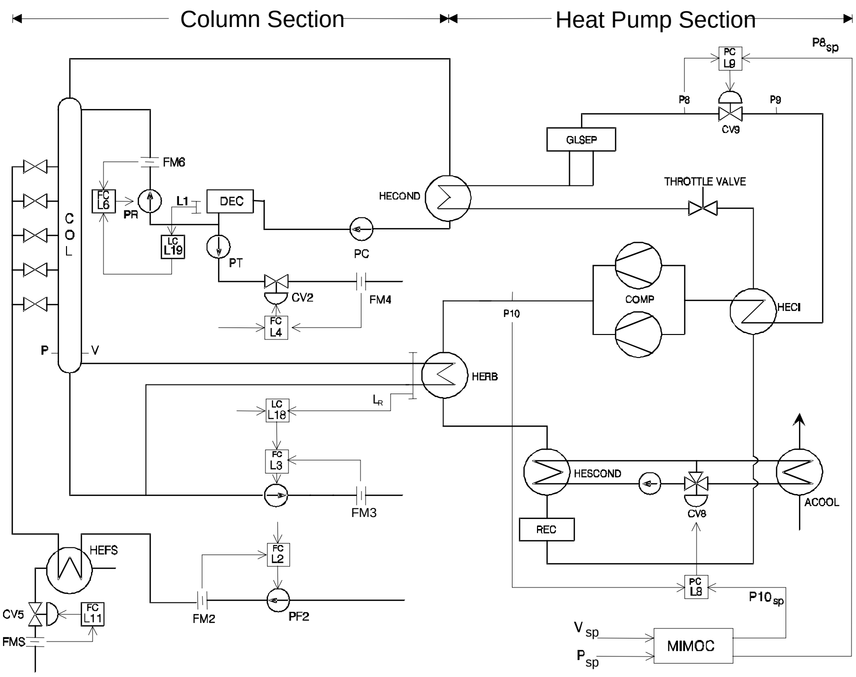

Figure 9: Process diagram of components and controllers.

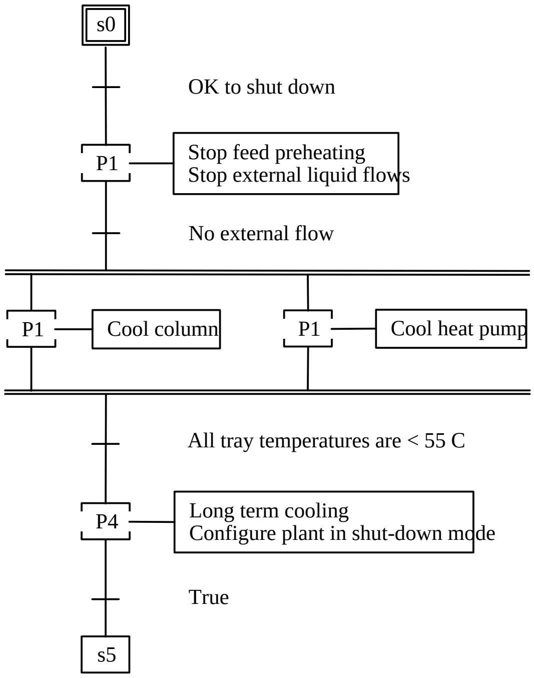

Figure 10: Skecth of shut-down using pseudomacrosteps.

take care of local flow control (FC) and pressure control (PC). For the overall control a 6 inputs – 2 outputs controller (MIMOC) manipulates the setpoints for the low pressure (LP) side and the high pressure side (PH) of the heat pump. The controller setpoints are column bottom pressure (Psp) and boil-up rate (Vsp). In the following, the plant shut-down procedure is modelled using a Grafcet formalism.

3.1 Phase 1: sketch.

To begin with, sketch the shutdown procedure in Grafcet using *pseudomacrosteps*. A pseudomacrostep is a sketch of a step, which does not necessarily comply with the strict syntax of regular steps. The symbol is of a set of square brackets lying on the side. Figure 10 shows such a grafcet with three sequential operations. First, all external flow streams are stopped (P1) followed by two pseudomacrosteps P2 and P3 to cool the column and the heat pump respectively. Once the column has reached a relatively safe state, long term cooling is carried out (P4). Step *s5* indicates that the plant is in a standstill state. Note, that this grafcet represents only the main goal and the sequence of the means to achieve it.

3.2 Phase 2: eliminate pseudomacrosteps

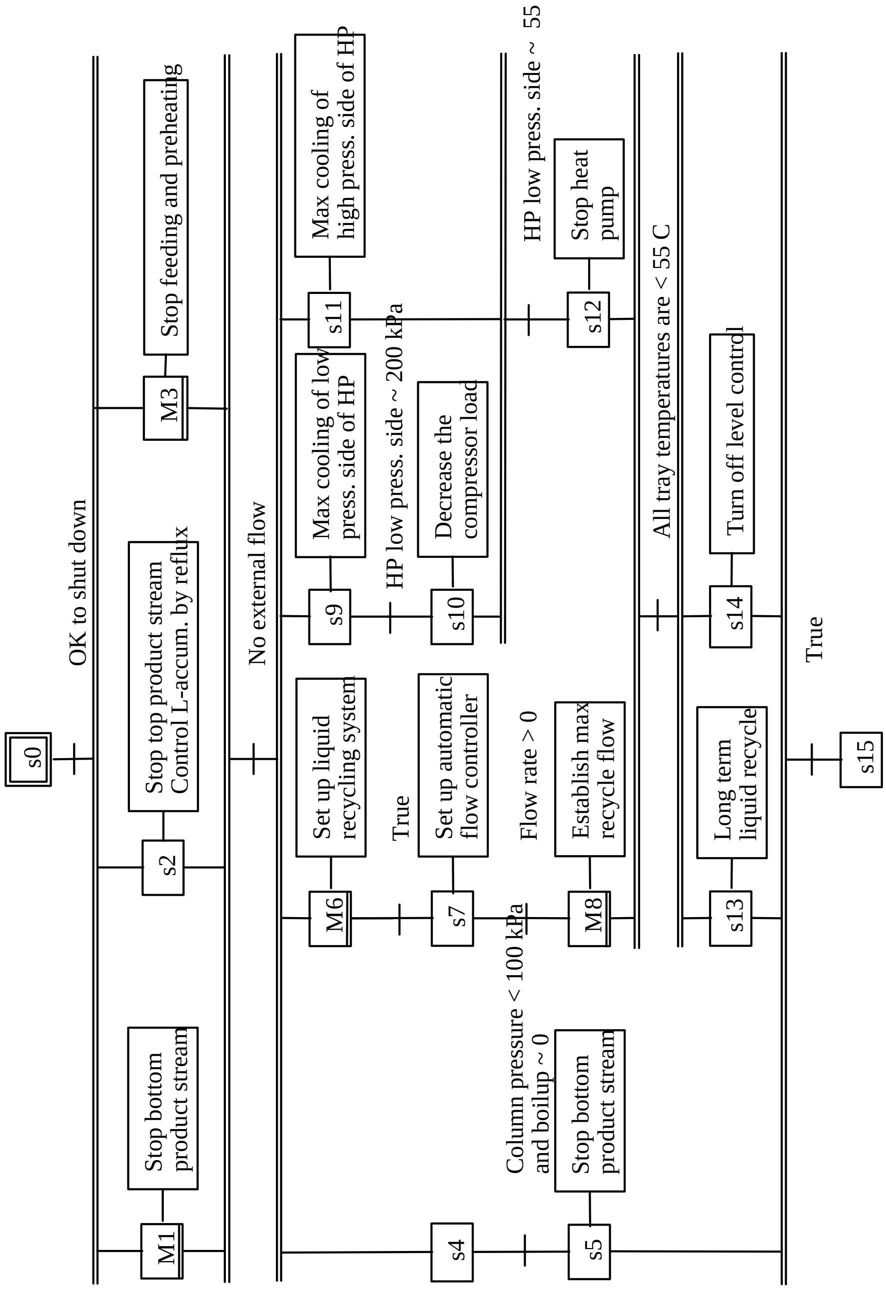

Figure 11: Expanded grafcet with only ordinary steps and macrosteps.

Describe the contents of the pseudomacrosteps in detail. Depending on complexity and experience, a number of intermediate grafcets may be necessary until a grafcet without pseudomacrosteps is obtained. Figure 11 shows a grafcet consisting of a number of (regular) steps and macrosteps. A macrostep represents a sub-grafcet as one step. Its structure is such that M1, s2 and M3 corresponds to P1 in the previous grafcet. Furthermore, the pseudomacrostep P2 is implemented as a sequence of M6, s7 and M8 to cool the column by recycling and cooling the reboiler liquid. At the same time, the cooling process of the heat pump is carried out in s9, s10 and s11 by maximum cooling of the low and high pressure parts of the system. The heat pump is completely stopped in s12. When the boil-up rate is decreased to zero and the column pressure is below the atmosphere pressure, the column is opened to the ambience for safety reasons (s5). The long term column cooling (s13) is the step where the recycle flow rate is fixed to a certain level for a few hours. In the same phase, final plant re-configurations for standstill are performed (s14). When s15 become active the plant is in standstill. In order to complete the functional grafcet above the macroactions must be specified in detail (see Yazdi, 1997)

M8 is a continuous process aiming at optimizing the cooling rate of the column. Instead of a fixed setpoint to the flow controller L3, the setpoint is changed dynamically. This task is carried out by a cascade controller with PB load and regulation error of L3 as input variables, and new setpoints of L3 as the output variable. The cascade controller is implemented as a fuzzy logic rule-based controller.

3.3 Phase 3: identify performance criteria and corrective actions

According to experience two macrosteps namely M3 and M8 can present potential problems during plant shut-down. A number of supervision tasks must be carried out during these steps. Table 7.1 summarizes criteria for measuring the degree of fulfillment of goals. The elapsed time is interpreted as the performance of the scheduling. The failure degree of L3 indicates the degree $e_i \in [0, 1]$ in which L3 is responsible for the lack of performance of the flow rate control. In the data consistency part, the closing of the binary valve 46 is confirmed by the liquid flow rate passing through the valve (0.01 is the maximal flow measure calibration error). Finally, the confirmation of the valve being closed must be received (within 2 minutes). Feasible plant actions upon the failure of the criteria are also suggested in the table 7.1. Note, that the choice of actions is dependent on what options are available.

The supervisors are grafcets triggered by a failure to meet a performance criterion from the supervision task. In step M8 the recovery action in case of pump cavitation (building of vapour bubbles due to fast rotation) is to stop the activity of M8 to release the actuators, set L3 in manual mode, and decrease the PB load to a relatively low value (here 0.1). After a wait period of 60 seconds, the controller is set back to automatic mode and the activity of step M8 is resumed. The remaining supervision tasks can similarly be represented in Grafcet.

Summary An advantage of the approach is that only one formalism, the Grafcet, is applied to represent the supervisory control system. At the same time, Grafcet is used to

Task	Criteria	Actions
Management and planning performance	None	None
Scheduling performance	Activity time $<$ 5 min	Increase timing limit 10%
Optimisation performance	None	None
Cascade/MIMO control performance	None	None
Direct control performance	None	None
Operational limits fulfilment	None	None
Safety limits fulfilment	None	None
Data consistency	If BV46 is closed then FM2 $<$ 0.01	Warn operator: possible failure of BV46 closing
Data validity	The closing of BV46 is confirmed within 2 min	Warn operator: failure to confirm command

Table 2: Supervision tasks for step M3

Task	Criteria	Actions
Management and planning performance	None	None
Scheduling performance	Activity time $<$ 30 min	Increase timing limit 10%
Optimisation performance	Recycle pump load is around 80%	Tuning needed
Cascade/MIMO control performance	Recycle pump load is $70 - 90$ %	Tuning needed
Direct control performance	Failure degree $<$ 0.2	Tuning needed
Operational limits fulfilment	No cavitation: not (PB$>$0.7 and FM3$<$0.1)	Restart the pump
Safety limits fulfilment	None	None
Data consistency	None	None
Data validity	None	None

Table 3: Supervision tasks for step M8

represent the controller actions, and it can be used all the way down to the level of the PLCs. The distinction between the supervisory grafcet and the controller grafcet provides clarity in the design. The somewhat restricted priority management in Fuzzy II can be removed, by replacing it with a Grafcet type of priority manager.

4. Conclusions

High level control projects are challenging from both a technical and a project management point of view. The technical aspects include signal processing, control strategy design, test and installation, and these tasks, of course, have to addressed properly. Good technical projects may fail if the end users have not been properly involved in all the phases of the project. The project management has to ensure that the end users consider the new high level control system a help and not a threat.

5. Acknowledgments

The original work was mainly done by Hassan Yazdi and Jens-Jørgen Østergaard, FLS Automation. Hassan Yazdi developed the design procedure during his PhD project, supervised among others by myself and Jens-Jørgen Østergaard. The latter contributed a section about high level control for another publication; I edited an excerpt and included it in the present document.

References

Assilian, S. and Mamdani, E. (1974). Learning control algorithms in real dynamic systems, *Proc. Fourth Int. Conf. On Digital Computer Applications to Process Control, Zürich*, IFAC/IFIP, Springer, Berlin, pp. 13–20.

David, R. (1995). Grafcet: A powerful tool for specification of logic controllers, *IEEE Transactions on Control Systems Technology* **3**(3): 253–268.

David, R. and Alla, H. (1992). *Petri Nets and Grafcet: Tools for Modeling Discrete Event Systems*, Prentice Hall.

Holmblad, L. P. and Østergaard, J.-J. (1982). Control of a cement kiln by fuzzy logic, *in* Gupta and Sanchez (eds), *Fuzzy Information and Decision Processes*, North-Holland, Amsterdam, pp. 389–399. (Reprint in: FLS Review No 67, FLS Automation A/S, Høffdingsvej 77, DK-2500 Valby, Copenhagen, Denmark).

IEC (1988). Preparation of function charts for control systems, *Technical Report 848*, International electrotechnical Commission.

Østergaard, J.-J. (1990). Fuzzy II: The new generation of high level kiln control, *Zement Kalk Gips (Cement-Lime-Gypsum)* **43**(11): 539–541.

Østergaard, J.-J. (1996). High level control of industrial processes, *in* L. Yliniemi and E. Juuso (eds), *Proc. TOOLMET'96*, University of Oulu, Control Engineering Laboratory, Linnan-

maa, FIN-90570 Oulu, Finland, pp. 1–12.
Peray, K. E. and Wadell, J. J. (1972). *The Rotary Cement Kiln*, Chemical, New York.
Yazdi, H. (1997). *Control and Supervision of Event-Driven Systems*, PhD thesis, Technical University of Denmark, Dept. of Chemical Engineering.