9 PROCESS CONTROL BY MINI COMPUTERS

by J.O. Liljenzin

Increasing demands on reliability, work environment and increasing labor costs, etc. results in computer control of different processes and processes becoming more common. At the same time, developments in the electronics sector mean that the price of mini-computers decreases while their performance increases. The boundaries between the former "mainframes" and today's mini-computers are erased. Similarly, the boundary between mini- and micro-computers is expelled. Each computer management application has a lot of individual problems and requirements, so this presentation must be of a very general nature. The purpose is therefore to provide a systematic description of the signals, functions and program parts that usually occur in minidator management of different processes. Details that depend on particular characteristics of the controlled process will be omitted.

General views on computer and software

Computer and software are often a very small part of total construction costs. However, the function downgrade of this part greatly affects the entire operation reliability of the process. The cost of the entire computer system may, for example, be less than the value of a day's operation. Accordingly, operational reliability is usually more important than a low price. Other general considerations can be summarized as follows:

1. Computer

Appropriate word length, Arithmetic goes faster with large word length, while bit management becomes more memorable and time consuming with great word length. Enough large memory capacity. The speed increases and the risk of error decreases, the more programs that can be permanently stored in memory. In / out scrolling to t, eg, disk memory involves mechanical equipment with moderate reliability. Registry based addressing facilitates multiprogramming and, in combination with automatic interrupt-generated registry swapping, provides short response time and shorter applications.

2. Hardware

High reliability. Preferably> 1 year MTBF. High environmental resistance. Dust, moisture, chemicals, temperature variations common. High Sensitivity - Contactors, Thyristors, Large Motors etc, Common. Mechanically robust. Short repair time. Drift stops usually expensive.

3. Software

Program may not terminate in case of error. Different from standard computing. Frequently programmed by "re-entrant" type. Common subroutines and features must be of "re-entrant" type. High-level language less common. Depending on the bit management, etc., most machine languages are optimal.

4. Flexibility

Experience shows that when a system has been run for some years, there is seldom any change except when the equipment changes. Development of programs "on-line" is usually inappropriate. There is a risk of interference in the system when new programs are run,

5. Information and signal flow

The paths that the information follows in a computer-controlled system is given schematically in Figure 1. Signals can be divided into analog and logic signals depending on whether they are continuously variable or discrete. All signal processing occurs in real time. Time is an important major in process control because a certain signal may have different meanings depending on when it occurs. Signal paths are also affected by the system's condition, see below. A common fake is to maintain some older interference on control signals when the computer system is introduced. This can lead to unexpected locks in the system.

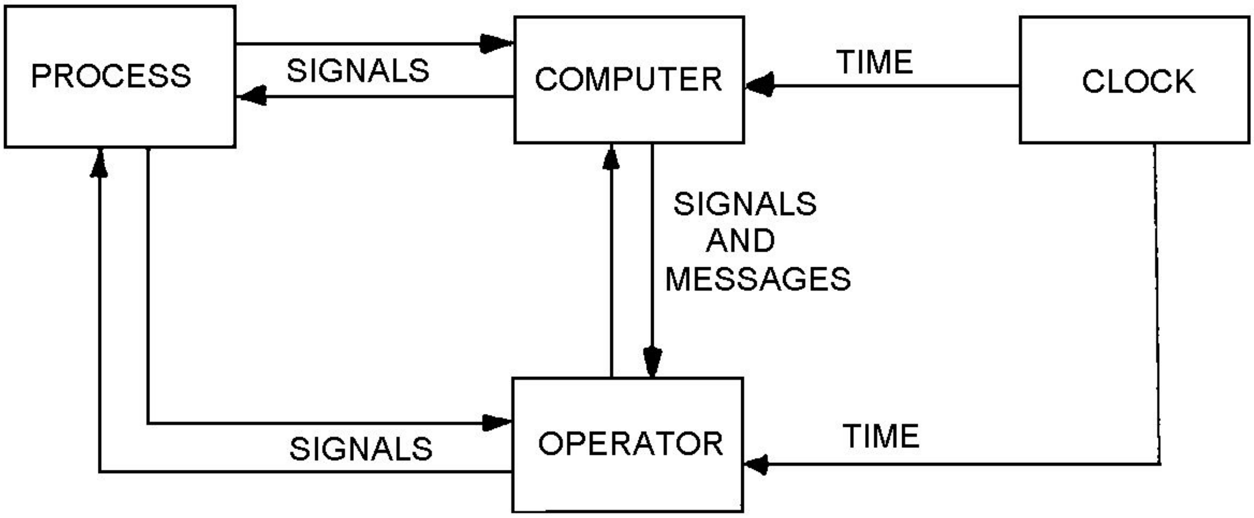

Figure 1. Information flow in computer control of a process.

System Condition
A computer-controlled process usually has two states: manual operation and automatic operation, which differ by the operator and the computer's ability to influence the controlled process.

1. Manual operation
The computer can not affect the process. This is prevented by appropriate external interference on the control signals. The operator can freely influence the process. He must be able to read the necessary signals, regardless of the computer. Manual operation usually requires control panel and spare system.

2. Automatic operation
The operator can not directly affect the process control signals. Prevented by appropriate interfaces. The computer now has sole control of the control signals. However, the operator can indirectly influence the process via signals to the computer.
The need for manual operation help causes t, e.g. Analog signals from the process are often converted to digital outside the computer and sometimes the computer may control analogue controllers. This avoids certain duplication of equipment.

Functional Classification
In order to better understand, review and discuss various functions required by a computer-controlled system, a logical system structure can be made according to Figure 2. The features may be separated on different modules or may be part functions in larger modules depending on what appears most appropriate in the individual case. The following gives a brief account of the various system functions,

1. Monitor
The program system usually operates during and with the help of a monitor program. In this, one or more of the following tasks are performed:

a. Interruption management by priority
b.Time sharing between programs by priority
c. Memory allocation for applications

d. Start of program:
- At the request of the operator.
- At the request of another program.
- At a given time.
- after the given time interval

e. Monitor programs during execution
f. Communication of data between programs
g. Coordination of programs (to avoid conflicts)
h. Allocating in / out devices to the programs
i. Feldiagnos
j. System time (year, month, day, hour, minute, etc.)

A process usage monitor often differs in some respects from those used in other real-time applications. These differences depend on the slightly different requirements that are imposed on the monitor, ie:

a. Minimal possible "over-head" time
b. Minimum memory requirements (competing for memory with process programs)
c. Be able to handle temporary overloads by queueing tasks (including some of the monitor's own data that have lower priority)

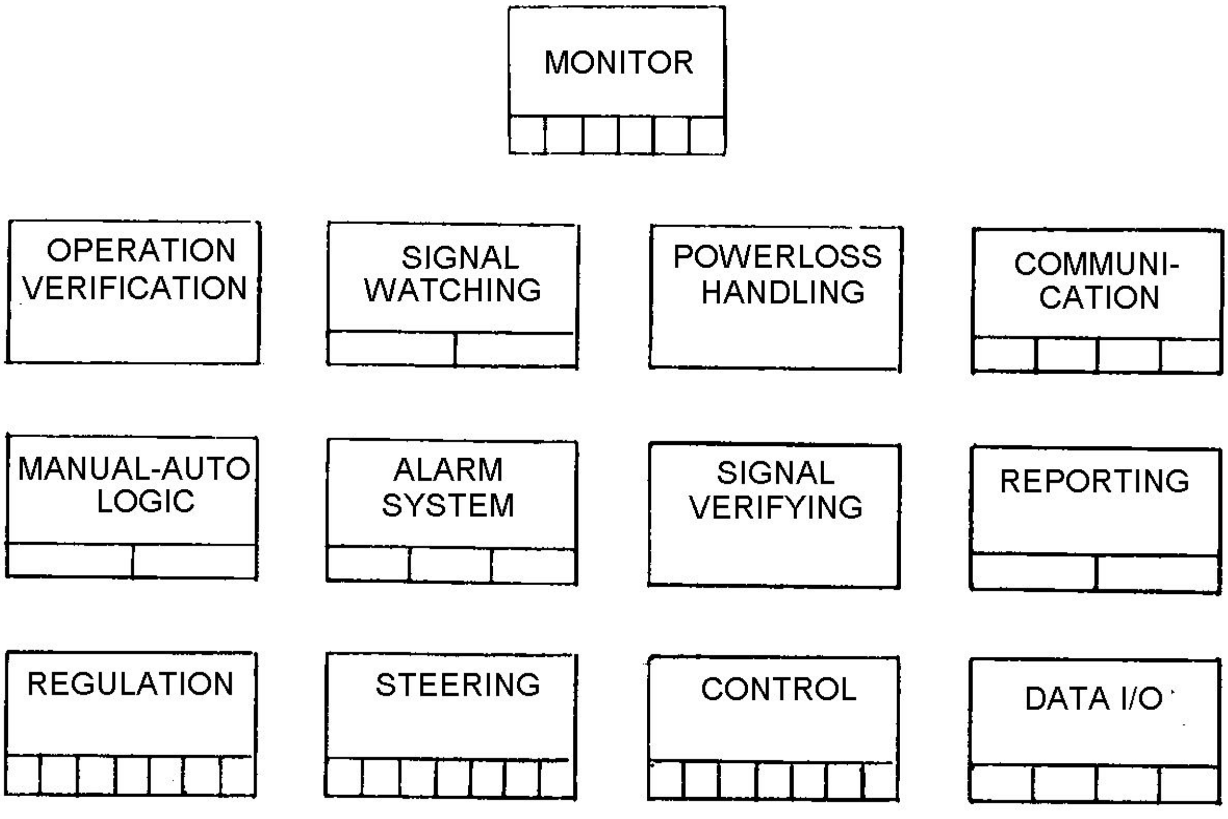

Figure 2. Logical system structure by function

1. System permission
The subordinate programs always have one of five different states:
a. Requested (The program will start as soon as possible)
b. Started (memory area busy etc.)
c. Interrupted (accidental interruptions)
d. Waiting (voluntary interruption)
e. Stopped (memory available etc.)

2.Drift verifcation
In order to prevent malfunctions and accidents, continuous monitoring of the functioning of the computer system is required. This usually happens through special signaling to an external monitoring device. Can be done by eg a small priority with the lowest priority is executed cyclically and thereby pulses an output signal. If the pulse rate drops or the signal stops in any mode, the external logic must cause alarms and transmit the system in manual mode. For systems with multiple concurrent computers, corresponding signaling between the computers is required. The program that monitors other computer operating indicators must have high priority,

3. Signal monitoring
This function means that incoming signals are monitored and programs are started or continued when certain external events occur. The signal monitoring programs are often executed cyclically, for example, every 1/10 second. Must have high priority and short execution time. Two functional possibilities are available:

a. Fixed connection between signal and program.
b. Optional connection between signal and program. Monitoring is requested in this case by the individual programs. Note: more than one application may request monitoring of the same signal.

4. Power failure management
In case of power failure, the hardware must cause program failures and start a special routine in time before operating voltages have fallen too low. This routine must take action during power failure and, on the other hand, handle the problems that may occur when power is returned and the computer restarts automatically.

4.1. Measures for power failure
The time for action is usually limited to a few milliseconds. For Honeywlll DDP-516, H-316 and H-716, power is available for normal operation for at least 20 milliseconds after power outage. The most important actions are summarized in the following list:

i. Save immediately status words and registry contents (if this does not happen automatically)
ii. Manage queued interruptions (store information because hardware is probably reset)
iii. Stop operating signal (may be local power failure for the computer)
iv. Set up and save the restart sequence that will run when the computer automatically starts working when the power returns.
v. Stop the computer

4.2 Action when power returns
The computer should normally start automatically in a fixed memory location, so that the modified and saved restart program automatically starts.
Two different series of actions are then possible, depending on the type of power failure (computer or computer and process only) and how the connected process works after a power failure.

i. The process can not be started
- Switch to manual operation.
- Power failure.
- Register the interruption.

ii. The process may be started
- Verify operational integrity. If not OK, switch to i. Above.
- Start operating signal

- Start restart sequence
- Fix saved interrupted indications. For example. simulate interruptions from external hardware.
- Register the interruption
- Alarm, then system time may be incorrect.

5. Communication

Since many signals (in / out words) can be composed of bits that are managed by different programs and / or require certain input and output sequences, it may be appropriate for a program (or more) to handle the input / output of such data. Non-interrupting inputs and outputs often use small programs for this purpose. The program is then executed cyclically at the appropriate time interval. Data is retrieved and stored from or on given locations in memory. Often a cycling time is sufficient in 10 seconds. Priority must be higher than for those programs that can utilize this feature. Otherwise, severe locks may occur! The signals can target several recipients or come from several sources:

a. The process
b. Operator
c. Peripheral devices to the computer
d. Sequence equipment
e. Subordinate computer, side-by-side computer or parent computer.

6. Manual to Auto Logic

This function will transfer the system from manual to automatic operation and vice versa. Often this happens through two different programs, one for each direction. These programs are preferably started by external signal or combination of signals. Note, the power cut routine may need to start the "auto to manual" routine. The data relating to these routines can be summarized as follows:

a. Manual to auto transition

This feature may have a low priority since it rarely is very urgent to switch to automatic operation,

i. Analyze the state of the process (mode, operations, activities)
ii. Add the necessary signals so that the process does not break or stop
iii. Request required information from the operator (what this has done or wants to do)
iv. Add auto signal to change the control of the computer
v. Start the required programs
we. Confirm state (message, log event)

b. Auto to manual transition

Should have a relatively high priority as something may have occurred which makes it urgent for the operator to get manual control.

i. Stop some operations (accidental if they continue)
ii. Break the auto signal so that the operator is given control
iii. Print message about ongoing operations
iv. Stop affected applications (eg via stop flag)
v. Launch Waiting Programs (then automatically stops due to the stop flag)
we. Reset any alarm and alarm queue (see Alarm below)
vii. Reset for example stop flags after the appropriate time (prevents the operator from swapping back and forth between manual and auto mode)
viii. Start monitoring of automatic request

7. Alarm system.

A well-functioning alarm system is one of the most important modules in a computer-controlled system. The alarm system often consists of several programs that interact. The following minimum data are available:

a. Larma operator or parent computer in case of malfunction or abnormal progress. May sometimes be a forecast of expected malfunction.
b. Register the event so that the operator receives the correct message (alarm queue)
c. Prevent further use of improper equipment (blocking)
d. Restore alarms on orders from the operator (receipt)
e. Reset blocking or take special actions on command from operator (receipt code, action code). The codes control how the system will handle any errors (remedies or remnants).

Because the different tasks have different priorities, it may be useful to share the data on several applications, such as the four following:

a. Registration of alarms.
- The alarm is set in queue with time, cause, program and possible receipt and blocking requirements.
- Possible blocking is performed. The alarm is also set in Kvittenskö with unique code.
- The alarm signal to the operator is triggered
- If no acknowledgment or blocking is required, the program restarts that triggered this alarm (alarm is alert)
- Lamb operation is requested

b. Alarm printing.
- Alert text, preferably plain text, is printed that gives the operator information about:
i. Time of alarm registration
ii. Cause of the alarm (fixed and variable alarm text)
iii. Certificate of receipt and, if so, current code

iv. Any blockages that have occurred through this alarm
v. If the alarm has no (blocking) requirements, it is removed from the queue.
we. Logging of alarm (sometimes required for error history) in table or on in / out device.

c. Alarm receipt.
The operator acknowledges any alarm. Two types of alarms can be current, with and without action code. If the alarm is not blocked, any kind of general receipt will suffice. If blocking is triggered, there are two possibilities:

- Only receipt (blocking remains, the alarm remains in the queue).
- Receipt with action code (blocking canceled, action code transferred to alarm program, alarm removed from queue).

d. Confirmation of receipt.
A valid alarm letter should be confirmed by printing a single message. This should include information about alarm time, quitting, quiz and any action code, eg:

1979.04.03.00.01.31 ALARM 1979.04.03.02.05.10 NO 74 ACTION 3

An invalid receipt should be followed by notification and printing of existing alarms. To allow rapid action, the alarm registration must be done with high priority. Other actions can then be done with lower priority. Appropriate order in priority can be: a >> d> c> b

Alarms should be indicated visually visually with appropriate equipment and with indication of the severity of the alarm, eg:

a. Warning without receipt requirement (indicates that something has happened)
b. Alarm with receipt requirements (protection for personnel etc.)
c. License Requirements and Action Requirements (the computer needs help with decision making)
d. Received with unresolved alarm remaining (enlightenment)

Type a. Above can be easily reset by the operator via the signal button.

8. Signal control
In order to achieve satisfactory security in a computer-controlled system, redundancy or reasonability testing is required on both inbound and outbound signals. Signals that do not meet these conditions shall not be used by computer or controlled equipment. If the error persists after a given time, an alarm is given and the signal is blocked. The signals can be checked in the individual programs, but it is often more beneficial to collect this activity in special programs. The status of the signals can, for example, be put up in special "status" words.
In special cases, all outgoing signals are returned from the computer to appropriate inputs on this (or answers to the signals from external equipment). The computer will then be able to verify its own outputs and their effects. NOTE! Important that the reading of outgoing signals occurs on the equipment that is to be signaled (see the cause of the Three Mile Island accident).

9. Reporting
Some form of reporting occurs in most computer-controlled systems. This function can be divided into two tasks
a. Data storage
This usually happens through appropriate sequences in the programs that handle the desired data. Data can be stored in data area in memory or sequentially on external device,

b. Processing and printing. This feature has very low priority.
 - On command from the operator
 - After signal from process or
 - At a given time

The function processes data appropriately and a report is printed or stored on the external device. The report should begin with text that defines the content, the reason for the report, the time, and the time interval that the report refers to. Normally, the data area (data file) is not reset except when printing a special report, such as day report,
Reporting is the part of the system where local requests are strong and very different.

10. Regulation
By regulation (cf. control below), here is meant the task of achieving and maintaining a prescribed state (setpoint) of all or part of the process. A simple typing of the control function can be done as follows:

a. On / Off control (eg thermostat type)
b. Simulation of analogue controller (eg PID), This is commonly referred to as DDC (DDC) Direct Control (DDC) control,
c. Adaptive regulation. Usually occurs through adjustable control constants. Hereby, the "learning" routine will determine which values ??of the rule constants give the best results. Difficult sometimes to provide reasonable starting values.
d. Optimizing regulation, Purpose here to get a better setpoint, Several different methods exist. Difficulty often exists to uniquely (a number) indicate what is optimal (or what is better than

anything else). Especially difficult at a high noise level.

Computer controlled regulation is often no better than conventional analogue regulation except in some cases. These are:

a. Slow progress with long time between cause and effect
b. High level of noise
c. Varying setpoint
d. Optimization needs
e. Varying regulatory characteristics of the process

11. Steering
With control (cf. regulation), here is meant the task of controlling the process through a series of states, e.g. Moving an object from one location to another, characteristic is that control usually deals with sequences of actions and events. A primitive division of the control function in different types can be done as follows:

a. Sequence control with steady progress, eg traffic light.
b. Sequence control with alternative choices, Selection takes place between several fixed sequences due to some form of input, e.g. time.
c. Adaptive sequence control. Here, sequence selection takes place by controlled magnitude and time, etc.
d. Optimizing sequence control, the sequence varied to provide optimal effect of controlled equipment. For example. fastest possible traffic.

12. Control
Control is the task of monitoring the function of a complicated process. This task can be typed as follows:

a. Data Registration. Mostly the base for other control functions.
b. Border Monitoring.
c. Checking the mode value time so that the controlled process is not subjected to abnormal stresses or malfunctions, such as: monitoring of machinery,
d. Feldiagnos. Check that the prescribed functions of the process work, for example electrical equipment testing.
e. Error statistics. Check that the prescribed error rate is not exceeded, e.g. product control.
f. Function forecast. Calculation of future performance of the process and verification that this will then meet specified requirements, eg collision warning for boats and aircraft.

13. Data I/O
This feature delivers data and constants to and from the process. For example, update after manual operation or reports.

(2018-01-10. The paper was translated into english from a scan of the swedish original in paper form, converted via OCR to text, corrected and improved in some small details by J.O. Liljenzin)