Flexible distributed control of production line with the LON fieldbus technology: a laboratory study

N.P. Mahalik
Department of Electronics and Telecommunication Engineering, University College of Engineering, Burla, India
S.K. Lee
Department of Mechatronics, Kwangju Institute of Science and Technology, Kwangju, South Korea

Keywords
Industry, Control, Distribution operations, Fieldbus, Manufacturing systems, Reliability

Abstract
Almost all industrial systems are distributed with multiple control points which interact to a limited extent, for which the idea of distribution of task at local (field) level is emerging. As locally-based application tasks can reduce control delays, a fieldbus-based smart and reliable DCS solution is recognised as a leader for real-time industrial automation. Advanced control system has turned itself towards the implementation of digital distributed control systems (DCS) from centralised control systems. The phenomenon is becoming very popular because of its advantages over the whole operating system. Presents a case study for realising manufacturing systems (production lines) with fieldbus technology. The local operating network (LON) fieldbus system was chosen for this purpose because of availability of a wide range of products. Emphasises the reliability aspects of the control systems. A representative of a conveyor system, integrated with field devices, was conceived as the target platform.

Received June 2001
Revised February 2002
Accepted May 2002

Emerald

Integrated Manufacturing Systems
14/3 [2003] 268-277

[ISSN 0957-6061]
[DOI 10.1108/09576060310463217]

1. Introduction

Evolutionary changes in digital technology (DT) and industrial application portfolio are the main driving forces responsible for the development of modern control systems to be implemented in a distributed manner with fieldbus technology (Armstrong and Moore, 1994; Lutze, 1995, Mason, 2000). A new design step concerning decentralisation of control tasks through multiple processing units is required, when the implementation has multiple points of interactions. The basic problem is concerned with how to decompose an application and how to allocate the task in terms of a reliable distributed control system. A case study for realising a reliable and low cost solution for production line automation with fieldbus technology has been presented in this paper. Because of availability of a wide range of products, the local operating network (LON) fieldbus system (Kagan, 1991; Harold, 1992; Tsang and Wang, 1994) was chosen. A representative of a conveyor system, integrated with field devices such as sensors (temperature and colour), actuator (an AC motor), microswitches and a pick-and-place robot, situated in the laboratory was conceived as the target platform for the realisation of DT-based reliable DCS implementation.

2. Terminology

2.1 Smart

The sensible word intelligent, within the instrumentation and process control environment, is variously defined and therefore has different meanings at different places. There is significant difference between conventional and smart control. Smart control attempts to build upon and enhance the conventional control methodologies and design to solve new challenging control problems. Typically, in the instrumentation and process control arena the term smartness is considered as the elements in a vector defined by the following parameters (Tian *et al.*, 2000; Christian *et al.*, 1998):

- digital system;
- distributed system;
- enhanced computing power;
- real-time performance;
- field level programming;
- modularity;
- scalability;
- interchangeability and interoperability (openness);
- reliability, safety and intrinsic safety (IS);
- higher degree of freedom;
- feedback loop at field level;
- sensor, actuator, loop, and system validation;
- life cycle data acquisition, etc.

2.2 Device

Devices are: non-smart sensors, actuators, valves, switches, drives etc. used within the industrial environment.

2.3 Node

Node is defined as an autonomous processing entity, which contains the following elements:

- a VLSI circuit containing a microprocessor;
- field devices such as sensor(s), actuator(s) or both;
- a transceiver for communicating through the control network;
- memory such as RAM, PROM, EEPROM,
- I/O ports; and
- the communication port.

The ROM and EEPROM store the protocol and user's application program respectively. A node is considered as a smart device as it

The Emerald Research Register for this journal is available at **http://www.emeraldinsight.com/researchregister**

The current issue and full text archive of this journal is available at **http://www.emeraldinsight.com/0957-6061.htm**

contains a processor. Physical and logical connections among the nodes are done via transceiver and by network service tools or network management tools respectively.

2.4 Component

A component is a hardware or software entity with an identifiable boundary which is designed to perform a certain function and which can be used to compose more complex entities for a given purpose. The components, within the control system environment, are thus some sort of generic basic building blocks that contain sufficient information and are considered to be an evolving portfolio which significantly enhance portability at all levels of the control system for consequent enhancement of the interoperability of the applications. In summary, a component is defined as a node with specific task inside (Xie, 1999).

2.5 Open system

An architecture whose specifications are public is known as open systems. When dealing with industrial applications the problem is not only to find the theoretical solution for the diagnostic problem, but several constraints come from the application environment. Unlike proprietary environments, an open systems environment allows users to choose from a wider selection of platforms that fit the implementation needs. This includes officially approved standards as well as privately designed architectures. These systems also offer increased interoperability (ability to plug instruments onto a control system and have them communicate with each other) and interchangeability (ability to swap one instrument for another in a control system with no loss of functionality) permitting different systems to fit together, and software developed on one platform to run on another with minimal adaptation. In summary, open system standards define the format in which data are exchanged, remote systems are accessed and services are involved. Architecture based on open systems standards can be implemented throughout the world as global systems become the norm for organisations. Systems are mutually open by virtue of their shared adherence to the appropriate standards. The aim is to provide a common framework intersystem communication (Train, 2001).

2.6 Agile control systems (ACS)

As a global competition for greater productivity, control systems are being evolved which are being considered as agile control systems. Many features cannot be addressed by the existing conventional control system successfully. There are cases where we need to significantly increase the opportunity range. New control architectures to perform novel control functions to meet new objectives are being designed while the system is in operation. In the industrial and manufacturing environments, control systems take many different forms for which the control architecturers needed to cover many different aspect of operation. Some of the important characteristics of the new generation of control architecture are optimal flexibility, scalability, automation flexibility, extendibility, portability, configurability, operational flexibility, modularity, re-usability, re-cyclability, sustainability, and most importantly, the reliability (Fung *et al.*, 2000).

2.7 Network access protocol

Generally, two types of networks are seen; data-network and control network. Enterprise-wide, global networking systems such as LAN, WAN, Internet etc. are referred to as data networking systems, whereas fieldbuses are considered as the control networking system or simply inheriting a control protocol. Fieldbus is a digital serial communication system, which includes a protocol, hardware and supporting software for multiplexing control signals over a single communication channel. The general definition of the protocol is that it is a set of rules, which governs the formatting of real data and scheduling the transactions of formatted data in the network. Since the fieldbus multiplexes any kind of control signals, it is suitable to implement DCS strategy. In other words, a fieldbus can be used for the realisation of distributed control. The fieldbus systems are nothing but the redundant version of the complete networking protocol. A complete networking protocol could be a manufacturing automation protocol (MAP) or International Standards Association/open system interconnection reference, model (ISO/OSI RM). Some of the services of many layers and even some of the layers of these complete networking protocol have deliberately been eliminated in the fieldbus protocol in order to meet real-time requirements. The existence of two kinds of networking systems is due to the reasons that the underlying requirement of bulk data communication and control data communication are very different. There has been some confusion between fieldbus technology and the data-network as they adopt the similar concept of digital signal communication. Data-networks usually are computer networks providing for communication between them in a geographically larger area. The access to the data-networks is achieved with a microprocessor based controller and a medium access unit (MAU). In a data-networks configuration, large quantities of data (in the order of megabits per second) are transmitted from one node to another. The timing of messages and their transfer is not critical. A control network, on the other hand, interconnects field devices (sensors, actuators and switches) in rugged industrial environments. Message based data, however, in small quantities, are transmitted in a time that is pre-determined. As already mentioned, fieldbus conforming to the OSI model typically omit some of the OSI layer functionality, in order to meet the real-time requirement.

2.8 Network management tool

Network management, as a term, has many definitions, dependent on whose operational function is in question. Network management functions such as fault management, configuration management, accounting, performance management, security, chargeback, systems management, cost management etc. originally emerged from the data communication network. However, network management functions within the DCS environment are somehow different. Managed objects are the field devices, routers, network and/or anything else requiring some form of generating codes, installing, configuring, testing, monitoring and repairing. Thus network management tools are off-the-shelf or newly generated software applications designed for asserting computer integrated activities for the design of DCS network.

3. Surveys on industrial control systems

3.1 Traditional scenario

Control systems can be categorised as centralised control systems (CCS) and distributed control systems (DCS) (Sansom, 1999). Typical example of conventional centralised monitoring and control systems with a central processing unit that communicates with number of field devices can be seen in Chu (1986). Such systems require individual point-to-point links between each field device and the processing unit. In other words, in a centralised system all the sensors and actuators are directly wired back to a central monitoring system. In the past, a manufacturing cell (MC) consist of a group of computer numerical controllers (CNC) capable of controlling machine tools, robot, conveyor system, automated guided vehicle (AGV) within the rugged industrial and manufacturing environments (Yeung and Moore, 2001).

3.2 Trends

The increase in both quantity and quality of sensor information coupled with increased performance demands have proven to be a problem for an overloaded centralised control strategy (Reza, 1994). For a large system the total number of I/O signals could total thousands, and this number is beyond the capacity of computer hardware. The computer, however powerful, has difficulty in polling round all the inputs within the time limit of the system (Mahalik *et al.*, 1995). Other disadvantages in centralised systems are (Tomgren *et al.*, 1993; Warwick, 1996):

- system itself is not flexible;
- the system will be totally paralysed if central processing unit fails to function;
- unable to make best use of on-line technique;
- slow to take advantage of improved technology; and
- the installation cost tends to be high.

Hence there has been a move towards distribution of control tasks throughout the system (Dunlop, 1989; Parkin *et al.*, 1993; Yaxley and Parkin, 1994; Blair, *et al.*, 1996). Furthermore, control systems for real-time applications are characterised by their ability to support strong time constraints (Bennet and Linkens, 1984; Tomgren *et al.*, 1993; Cooling, 1994). Industrial automation systems have been growing speedily with the widespread availability of digital technology, information systems, parallel processing and computing environment.

3.3 An enabling platform for implementing DCS: the fieldbus technology

Fieldbus is now becoming a buzz word within the instrumentation and process control environment (Ericssion, 1993; Goldsmith, 1994; Hodgkison, 1994; Beeston, 1995; Hoffmann *et al.*, 1995; Greek, 1996). Fieldbus is a generic term which describes a digital networking system for industrial automation solutions. Some of the industrial automation applications (target platforms) are, robotics, production line, packaging plant, autonomous guided vehicles (AGV), assembly automation, supervisory control and data acquisition (SCADA) applications etc. These are communications networks which will be used in industry to replace the existing centralised based 4-20mA analogue systems (Wood, 1995). The fieldbus technology includes hardware, software and other necessary ingredients for the design of digital, bi-directional, serial-bus control network by linking isolated field devices within the target platform described above. Each field device is connected to a low cost VLSI processor, making each device a "smart" device in terms of a node. Each node will be able to execute simple functions on its own such as control, validation, and subsistence functions. In basic nature fieldbuses will replace centralised control systems with distributed control networks. Therefore a fieldbus is much more than an alternative for the 4-20mA analogue standard.

Although fieldbus technology has been available for the past 12 years it is still not widely used. The reason for this delay is due to the lack of an open international standard which would enable interchangeability and interoperability between different fieldbus manufactures and suppliers. For example, the personal computer (PC) platform is an open standard and its architecture conforms to seven layers of ISO/OSI RM (International Standard Organisation/open system interconnection reference model). A user can buy the physical interfacing from different manufactures and can link them together to build his/her own computer. The documentation for open fieldbus architecture has not yet been forecasted. This indicates there is still a long time to wait. However, with endusers/consumers becoming impatient many big companies have decided to release there own fieldbus systems, which work off different standards and architecture. It is worth mentioning that, recently, there are more than 50 different fieldbus standards available in the technology marketplace and some of them are national standards (e.g. Profibus is one). Out of them, the name of some of the leading standards, in alphabetical order, are; Actuator sensor interface (ASI), Bitbus, Controller area network (CAN), Device network (DeviceNet), European installation fieldbus (EIB), Fieldbus Foundation (FF), HART, InterBus, Interoperable system project (ISP SP50), Local operating network (LonWorks), Process fieldbus (Profibus), Process net (P-NET), Smart distributed systems (SDS), World factory instrumentation protocol (WorldFIP) etc. For more information on these standards and others see www.synergetic.com/compare.htm and Mahalik *et al.* (1995) and Mahalik and Moore (1997).

4. A case study

In response to ACS, the work presented in this paper is based on a case study carried out with a representative PLCS (production line conveyor system) platform, present in the laboratory. The PLCS is integrated with an AC motor, movable belt, other components such as two microswitches, one colour sensor and a pick-and-place robot. As the name suggests, the role of the pick-and-place robot is to pick up an object when it arrives from the pre-defined place and to place the object at the appropriate place which is also pre-defined. The conveyor belt is fixed with an object holder, which holds the objects during run-time. The role of the colour sensor is to detect the colour of the objects. And finally, the function of two switches are to provide switch signals at the appropriate time when the associated events occur.

4.1 Target platform: detailed description

4.1.1 Conveyor belt

The conveyor belt has two sides. One side of the belt is considered as supply side and another side behaves as the delivery side. The supply side of the conveyor belt is fixed with 12 object holders and the belt is driven by the single phase AC motor. The motor is tightly connected to the belt through many shafts and gears so as to keep the speed of the belt within the operation speed. The delivery side belt holds empty boxes where the objects are to be placed by the robot. The empty box can accommodate 12 objects in the matrix 3*4. Each column has separate colour code (i.e. red, green and blue). If the colour of the object is red then the robot will place the object at the red column of the object box accordingly.

4.1.2 AC motor

A single phase AC motor was used to drive the conveyor belt. Normal speed of operation is 1,500 rev./min. The motor drives both sides of the conveyor belt.

4.1.3 Pick-and-place robot

The pick-and-place robot used in this study is just a robot arm consisting of six sections such as column, shoulder, upper arm, lower arm, wrist unit and gripper. The arm is controlled from a personal computer through an RS232 serial link. The control software contains the library functions as commands, which can be used to program the arm as per users requirement. Either Turbo Pascal or Forth can be used as the programming language. The author used Turbo Pascal to program the arm to do the necessary job. The arm is driven by three servo-motors and the wrist and gripper is driven by another three motors. The zed motor drives a belt which propels the arm up and down the column. The other motors make it swivel from side to side at the shoulder and elbow joints. Normally each motor works individually. One motor is responsible for driving the wrist from side to side (yaw) and other two within the wrist unit enable the gripper to move up and down (pitch). The gripper has its own motor which opens and closes the jaws.

Before putting the arm on-line, it is required to teach the robot to do the requisite actions. The arm was programmed and was put into place. The integration between the robot controller (PC) and LON system was achieved through game port (a game port is a port that is incorporated into a computer conventionally using a 15-pin D-style connector) of the PC.

4.1.4 Colour sensor
The role of the colour sensor is to detect the colour of the object and inform the robot by assigning a colour code to it. According to the colour code the pick-and-place robot will make a decision where to place the object in the empty box already there on the delivery side. There are three types of coloured objects (green, red and blue) used in this case study as mentioned earlier.

4.1.5 Temperature sensor
A temperature sensor is connected to the coil of the AC motor. A threshold value has been assigned to the temperature sensor. If the temperature motor coil exceeds the threshold value and the certainty value (CV); explained latter is greater than 75 per cent, then the motor will be stopped.

4.1.6 Microswitches
The function of two switches is to provide switch signals at the appropriate time when the associated events occur. One is attached to the conveyor belt and the second switch has been fixed at the robot zed. The first switch is activated when an object arrives. That means if the object holder touches the switch, the switch provides a switch signal to the system for next action to be performed. This switch also indirectly counts the number of objects already packed into the box so that if the box is full, the motor can be instructed to move the belt for next operation. The second switch provides a signal when the robot is brought to the initial position.

4.2 Introducing LON platform and supporting tools
LON fieldbus called LonWorks technology was selected, because of availability of a wide range of products (e.g. LonMaker and LNS (LonWorks Network Service) based network kernel are both tools for serving network management operations. However, LonMaker is a DOS based product and the other one is a WindowsNT/95 based product), to configure and synchronise various actions within the PLCS system.

4.2.1 Introducing LON
LonWorks is a proprietary fieldbus system developed by Echelon Corporation. This fieldbus system includes all the elements required to design, deploy and support distributed control. It is worth mentioning that LonWorks is an underlying architecture with which fully-fledged DCS can be built as per control requirement. Those basic elements are: protocol (called LonTalk), processor (Neuron chip), transceiver (different types of transceiver for different media are available), I/O interfacing (A/D, D/A converter), tool for developing control code (NodeBuilder) and tool for carrying out network management (LonMaker, 32-bit LNS network kernel based management tool). The field devices such as sensors, actuators etc. within the control environment can communicate with each other for control level interaction via Neuron chip-based node by the help of LonTalk protocol. Neuron chip is an 8-bit VLSI processor that can execute the control task at the field level and can interact with other Neuron chip-based nodes within the control network. The connections among the Neuron chips are made with the help of network management tools. The control code to be assigned (must be written in Neuron C language; a derivative of ANSI C) to any node is downloaded (after being developed, complied and debugged) by a software called NodeBuilder (Echelon Corporation, 1995).

The author used NodeBuilder, LNS network kernel as the LON compatible software tools and Neuron chip-based LTM-10 nodes and Gizmo-3 I/O board as hardware tools. LTM-10 nodes are fitted with transceivers for twisted pair wire and Gizmo-3 I/O board (from Motorola), that holds the A/D converter, seven segment displays and their drivers and a clock, were used for I/O interfacing to the Neuron chip based node. Echelon's PCLTA network interface card was used for the host PC. As a requirement, a pentium-based PC with WindowsNT OS was used as host.

4.3 Reliability within the components
Providing control tasks at the local points does not manifest that DCS system is intelligent. In some generalised sense it can be argued that such implementation can be treated as "smart DCS systems" (as it can

meet the real-time requirement), however, in DCS a device is said intelligent if and only if it has some degree of self-diagnostic capability (Evans and Underwood, 1994; Henry, 1995; Korbicz and Kus, 2000). Self-diagnosis concepts within the control system environment have long been a major research area. The original idea with regard to validation of devices has come from the research work at Sensor Validation (SEVA) Group at Oxford University. Potential research work is a prime requisite to answer how a DCS system can accommodate the concept of self-diagnosis features. On this issue, there is a need to introduce and implement self-diagnostic features within the components such as actuators, sensors, valves, switches etc. and this feature should be an integral part of DCS.

4.3.1 Validation components

As pointed out, a device is visualised as smart/intelligent if it has the capability of producing some sort of validation pointer (self-diagnostic feature). A device designed with validation property can be considered as a validation component as defined in the terminology section. The validation components with regard to temperature sensor, microswitches and DCS network have been designed and implemented. However, the validation component for the actuator system needs considerable attention as they are of many kinds, depending on their principle of operation. For example, basic actuators are usually a power switch, a valve, an electrical, pneumatic or hydraulic amplifier or motor. A classification and evaluation can be concentrated on two major groups; electromechanical actuator and fluid power actuator (Isermann, 1993).

Sensor device was validated by identifying five validation modes:

1 *Abrupt step change*. This type of sensor validation mode is called step or spike (being a delta function) failure. The signal change is quicker than its normal operation.
2 *No change fault*. For all time to come the device output hangs on to a constant value.
3 *Significant change*. When the measurement noise level of the device output changes significantly.
4 *Threshold fault*. When the device's output crosses the threshold limit.
5 *Additive*. Continuous increment or decrement of output signal.

The validation modes are represented in a term called certaintiness value (CV). The operational behaviour (Henry, 1995) of the CV value can be attributable to the following:

- CV is an index of sensor signal.
- CV is the refined operational value of the sensor.
- From the control point of view, it provides the health of the sensor.
- The characteristics of the sensor are rated by CV value.
- The value describes the quality of the input which is device independent i.e. transduction mechanism.
- CV can be used for onward transmission of signal for control use.
- The control system takes operational decisions in response to CV value.

The CV value is expressed in percentage. The validation algorithm was developed in Neuron C language and all the validation modes were tested separately with a pressure sensor integrated with the Neuron chip. In this case study, later however, the same validation algorithm was implemented with a temperature sensor to meet the requirement within the PLCS.

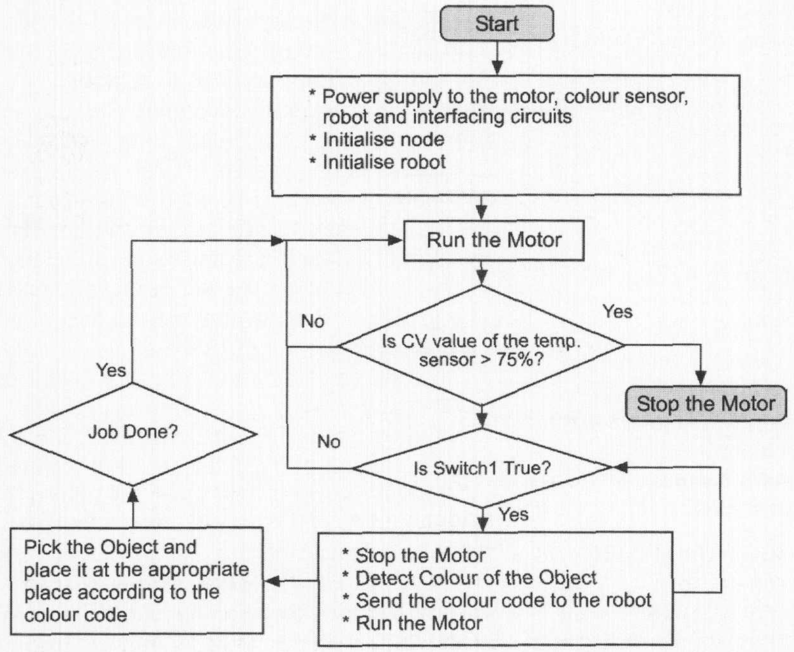
Switches are the on/off type devices and are not smart enough in the present form of automation and control applications. Operating a switch within the DCS environment with fieldbus type system does not signify that it is smart enough. In the process control environment or in robotics smarter switches are now desirable. The way a smarter switch differs from the traditional switch depends on the validated output signal. There is no guarantee that the output of a switch is real or false. The question like how it is guaranteed that the output of a switch is correct will arise, and to answer this question a means must be devised. As the control system components/devices are interacting with each other (e.g. in closed loop control system), any malfunctions with the switch signal may lead to serious disaster. For example, switches play an important role in the closed loop production line automation systems (PLAS; e.g. in a conveyor system of a packaging plant). It is not an easy task to validate the switch signal by signal processing technique as described in the context of sensor/actuator validation. The proposed scheme considered, in order to validate any kind of switch signal, is based on hardware redundancy approach (identical devices at least triplex configurations are verified by comparing the measured value at a given instant and the failures are isolated by majority voting). The idea considered is simple and flexible to implement with the fieldbus type systems.

Given validated sensors, switches, it is natural to move toward network validation in the DCS systems. That is, moving from a centralised system solution to a distributed solution introduces a different set of validation and robustness issues, in particular, issues of network liveness must be addressed. Questions like:

- How does the system know if all the nodes are working correctly?
- Whether or not the physical connection exist after installation?
- If a node becomes isolated what does it do? will arise.

A physical disconnection problem arises during control system design stages as well as run-time (after the design is complete). Even though the network is logically valid and all the logical bindings exist, any physical disconnection of any node due to human error or unexpected accident will bring another problem. As it is obvious, troubleshooting is required. In the present situation, such type of problem is being rectified by visual inspection. However, in the industrial environment where hundreds of nodes are connected to form a network, this type of human inspection is not really tolerable. For this a means should be devised in order to check continuously whether or not all the nodes are attached to the network physically. To solve this problem, a network validator component (NVC), an extra node is added and configured into the network that would keep on-line checking (if the physical connection exists) of all other nodes. That is, on-line network validation is achieved by designing a hardware component, which regularly collects information whether or not all the nodes are working correctly. It can be argued that by configuring an extra node the real-time performance on the network validated-based control network would definitely be decreased. However, the benefits can be gained in terms of implementing reliable and safety DCS systems.

Figure 1
Control flow chart



4.4 Realisation

The control sequence of the considered PLCS is shown in Figure 1 in flow chart form. In order to realise fieldbus based reliable DCS solution drawn upon a representative target PLCS with following control flow, six nodes were designed as shown in Figure 2. Besides Gizmo-3 I/O board, other interfacing circuits were designed to interface with the robot controller through Game Port, AC motor and colour sensor. The control code for each node was written in Neuron C language as mentioned earlier. Neuron C is designed for Neuron chip and the use is very simple and flexible due to the existence of network variable (NV); a new class of objects which simplify data sharing among nodes, and when statements; which introduces events and defines the temporal ordering of these events. There are 34 pre-defined I/O functions, which can be accessed by users for writing the application.

Regardless of the system complexity, the distributed system design methodology involves problem identification or specification, segmentation or decentralisation, development and testing. The stages involved in designing a LON based design development process include the following steps and are shown in Figure 3.

All the nodes were installed and configured (logical binding) with the help of a network management tool. It is worth mentioning that Echelon's LNS-based network kernel is a platform for developing the network management tools. The platform is called LonWorks Component Architecture (LCA) Object Server (LCAOS). Thus LCAOS is not a network management tool, rather it is an object server which provides functional objects for the development of WindowsNT/95 based network management tool. A management tool for DOS platform is readily available from Echelon in the name of LonProfiler and LonMaker. To meet the requirements for ACS, the author designed and used a LCAOS based network management tool for the installation and configuration of the said control network.

Figure 2
Production line automation with LON based reliable DCS network

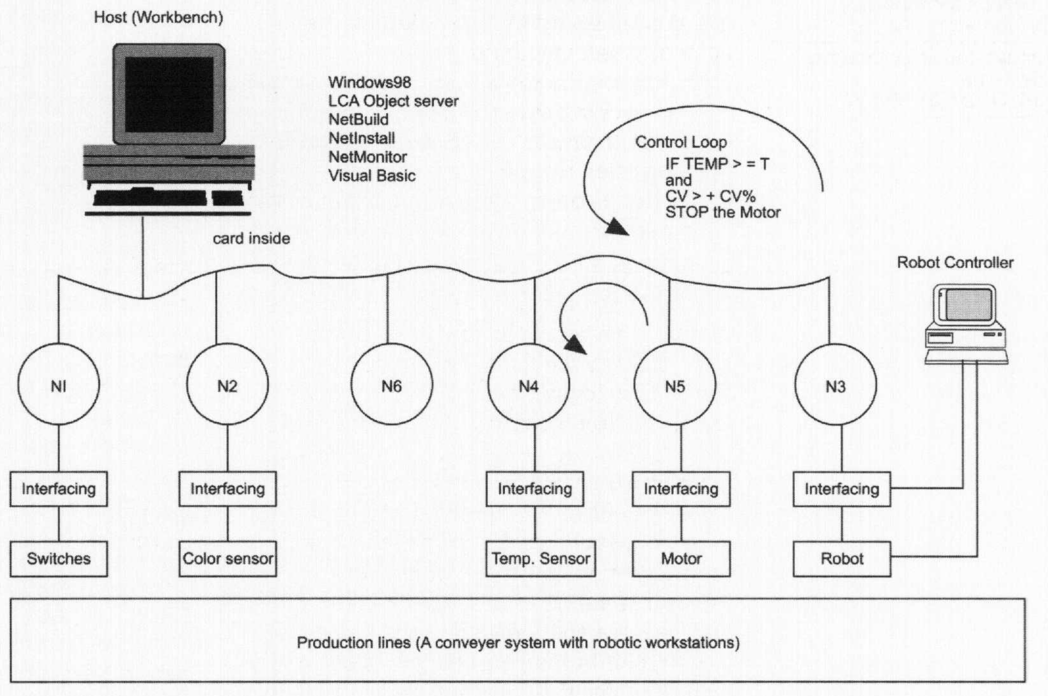
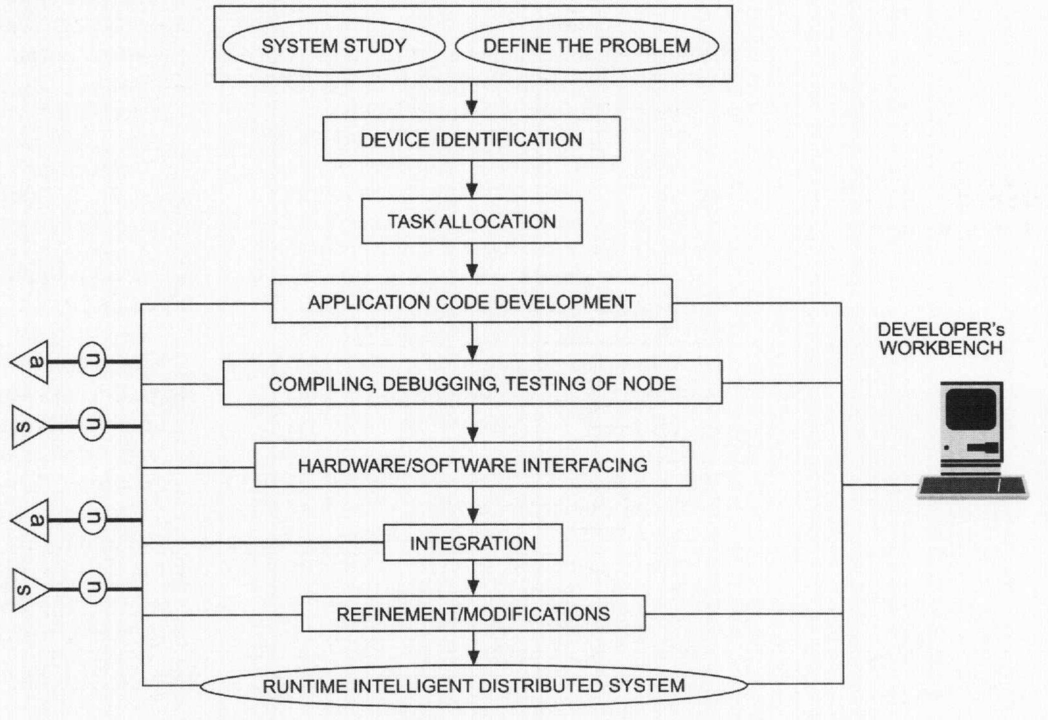


Figure 3
Fieldbus based design development phases



5. Conclusions

The fieldbus platform enhances efficiency by improving quality and reducing the overall design costs. This is a great deal more accurate than transmitting using analogue methods which were used previously. Each field device is also a "smart" device and can carry out its own control, maintenance and diagnostic functions. As a result it can report if there is a failure of the device. This increases the efficiency, productivity and availability of the system and reduces the amount of maintenance required. Fieldbus is no longer simply a communications standard but is an integrated measurement and control system which is looking to change the face of process control forever. In this paper a case study realising DT-based DCS solution for the automation application has been presented. In summary:

- As the modern automation system demands fieldbus based distributed control, a fieldbus based reliable device validation algorithm should reside close to the device to improve the plant efficiency, safety and availability. It is proved that sensors, actuators, switches, valves can be validated with the fieldbus system.
- Fieldbus based reliable DCS scheme is robust, easy to implement, economical with good performance since the validation algorithm resides at the field level i.e. close to the components. Thus it could improve the real-time requirement. The approach uses the same platform (fieldbus) for implementing self-diagnostic algorithm or fault detection and isolation (FDI) as well as control algorithm within the same node – hence a reduction in overall installation cost.

References

Armstrong, N.A. and Moore, P.R. (1994), "A distributed control architecture for intelligent crane automation", *Automation in Construction*, Elsevier, Amsterdam.

Beeston, J.W. (1995), "Fieldbus: the revolution arrives", *Chemical Engineer*, January, pp. 12-13.

Bennet, S. and Linkens, D.A. (1984), *Real-time Computer Control*, Peter Peregrinus, London, Chapter 6 and 7.

Blair, G., Coulson, G. and Davies, N. (1996), "Standards and platform for open distributed processing", *IEE Journal on Electronics Communication Engineering*, Vol. 8 No. 3, June.

Cooling, J.E. (1994). "Task scheduling in hard-real-time embedded systems using hardware co-processors", *Microprocessors and Microsystems*, Vol. 18 No. 10, December.

Christian, W., Dominik, H., Heinz, W., Jan, S., Martin, D. and Wolfgang, M. (1998), "A distributed planning and control system for industrial robots", *Proceedings, 5th IEEE International Advanced Motion Control Workshop*, pp. 1-6.

Chu, W.W. (1986), "Distributed systems", *Distributed Processing Systems*, Vol. 1, Aptech House, Inc., Dedham, MA.

Dunlop, G.J. (1989), "The industrial fieldbus", IEE Colloquium on Buses for Instrument, April, London.

Echelon Corporation (1995), *The 95-96 Echelon LonWorks Products Databook, LonWorks Engineering Bulletins, 993-5., Neuron C Programmer's Guide, Revision 3, 1990-5, NodeBuilder User's Guide.*

Ericssion, G. (1993), "A comparison of three fieldbus protocols: FIP, Profibus, and Masterfieldbus", *Proceedings of International Workshop on Mechatronical Computer Systems for Perception and Action.*

Evans, D.S. and Underwood, P. (1994), "Self-diagnostics in pneumatic systems using PLCs", *Proceedings Joint Hungarian-British Inernational Mechatronics Conference*, Budapest.

Fung S.T.R., Tam, K.H.P., Yeung, W.H.R. and Cheung, H.M.E. (2000), "A novel scheduling methodology for holonic manufacturing control system", 27th International Conference on Computer and Industrial Engineering, Beijing, China, October, p. 223 (CD-ROM version).

Goldsmith, P. (1994), "Fieldbus gets radio active", *Process Engineering*, November.

Greek, D. (1996), "All aboard the fieldbus", *Professional Engineering*, December.

Harold, M.R. (1992), "Distributed processing using local operating networks", *Assembly Automation*, Vol. 12 No. 1, pp. 14-9.

Henry, M. (1995), "Sensor validation and fieldbus", *Computing and Control Journal*, IEE, Vol. 6 No. 6, pp. 263-9.

Hodgkison, G. (1994), "Fieldbus – make a studies choice", *OEM Design Report.*

Hoffmann, F.M.D., Butler, M. and Doggett (1995), "A fieldbus selection methodology for manufacturing systems", *Proceedings of the FieldComms'95 Conference*, Hinckley.

Isermann, R. (1993), "Intelligent actuators – ways to autonomous actuating systems", *Automatica*, Vol. 29 No. 5.

Kagan, R.S. (1991), "Distributed processing using local operating network", *Proceedings of FIELDBUS '91*, The Cavendish Centre, London.

Korbicz, J. and Kus, J. (2000), "Diagnosis of sugar factory processes using GMDH neural networks", *Proc. 4th IFAC Symposium Fault Detection Supervision and Safety for Technical Processes, SAFEPROCESS 2000*, Budapest, Hungary, 14-16 June, Vol. 1, pp. 343-7.

Lutze, U. (1995), "Reduce control system complexity with fieldbus technology", *Control Engineering*, Vol. 42 No. 12, pp. 87-92, November.

Mahalik, N.P. and Lee, S.K. (2001), "Some studies on modeling and designing sensory systems for industrial applications", *Journal of Scientific and Industrial Research*, Vol. 61, January, India, pp. 34-47.

Mahalik, N.P., Moore, P.R. and Pu, J. (1995), "Fieldbus technology: a solution for multiplexing plant signal?", *IEE Colloquium Digest*, November.

Mahalik, N.P., Moore, P.R. and Pu, J. (1997), "Emerging real-time control networking systems for manufacturing and process plant: design methodology", *Journal of Production Engineering*, IE(I).

Mason, A. (2000), "Fieldbus is rapidly revolutionising the architecture of control systems", *A Newsletter of Advanced Control Technology Transfer (ACTT) Programme*, Reading University, No. 13, 16 February.

Parkin, R.M., Yaxley, S.J. and Bearman, R.A. (1993), "Intelligent operation of rock crushers using real-time distributed control", *International Workshop on Mechatronical Computer Systems for Perception and Action*, Hamstad University, Sweden.

Reza, S.R. (1994), "Smart network for control", IEEE Spectrum, *Data Communications, Special Report*, June.

Sansom, P. (1999), "Distributed control systems in cement processing plants", *World Cement*, November, Australia, pp. 323-5.

Tian, G.Y., Zhao, Z.X. and Baines, R.W. (2000), "A fieldbus-based intelligent sensor", *Mechatronics*, Vol. 10 No. 8, pp. 835-49.

Tomgren, M., Garbergs, B. and Berggren, H. (1993), "A distributed computer testbed for real-time control of machinery", *Proceedings of 5th Euromicro Workshop on Real-time Systems*, IEEE Computer Society Press, Oulu, Finland.

Train, R. (2001), "Field-based systems and advanced diagnostics", *Journal of Control Engineering Europe*, September.

Tsang, P.W.M. and Wang, R.W.C. (1994), "Development of a distributive lighting control system using local operating network", *IEEE Transaction on Consumer Electronics*, Vol. 40 No. 4, November.

Warwick, K. (1996), "Flexible distributed control of manufacturing systems using local operating network", EPSRC CDP Project GR/J64047, *Final Report Summary*, Department of Cybernetics, University of Reading, Reading.

Wood, G. (1995), "Fieldbus and intelligent devices", Editorial, fieldbus special issue, *IEE Journal of Computing and Control*, Vol. 6 No. 6.

Xie, C. (1999), "Component-image based methodology for the design of distributed control systems", PhD thesis, De Montfort University, Leicester.

Yaxley, S.J. and Parkin, R.M. (1994), "Distributed control systems within a mechatronic environment", *Proceedings of Joint Hungarian-British Mechatronics Conference*, Hungary.

Yeung, W.H.R. and Moore, P.R. (2001), "Steps towards advanced cell control systems: an application in flexible assembly", *International Journal of Intelligent Automation and Soft Computing*, Vol. 7 No. 1, pp. 3-21.

The work is partly supported by All India Council of Technical Education, India, under the research project "Flexible Distributed Control of Manufacturing Systems Using Local Operating Network" undergone at the University College of Engineering, Burla, Orissa, India and partly by BrainKorea-21 research program at Kwangju Institute of Science and Technology, Kwangju, South Korea. The authors also acknowledge Prof. P.R. Moore and Dr J. Pu in the context of technical discussion.