

# Intelligent Network System for Process Control: Applications, Challenges, Approaches

Qurban A Memon,  
*UAE University,  
United Arab Emirates*

## 1. Introduction

The ever increasing size, sophistication and complexity of sorting and handling systems, as industries strive to increase efficiency and offer greater consumer choice, is a constant problem for system controllers and integrators who build such systems. In particular, the growing trend towards larger capacity systems, which increase throughput rates and provide greater flexibility to deal with product variation and changing supply demand, place increased pressure on system controllers to get things right on time and within budget, often in the face of loose specifications resulting in large and complex programs. In view of these demands it comes as no surprise that, when designing and integrating a control scheme, control engineers prefer to use past practice to minimize risk. Current practice includes distributed and modular systems like programmable logic controllers (PLC's) to break down complex control.

The need for modularity, real timeliness, integrated diagnostics, decentralized control, expanding physical setups, and functionality has resulted into various types of distributed control system (DCS). Presently, the DCS concept is not only applied to the process control and manufacturing automation but it has targeted many areas like communications engineering (Yang, 2006), physics and nuclear science (Cavineto et al., 2006; Kleines et al., 2004), power system engineering (Stewart, et al., 2003; Ioannides, 2004), a lighting system design (Alonso, et al., 2000) etc. The innovative ideas are being applied in field, for example in (Alonso, et al., 2000), the authors have made use of the power line as the communication medium for developing the DCS of the lighting system, based on a custom built large scale integrated Neuron chip incorporating both the control and communication features in a three microprocessors based system. Another domain of application is industrial plant automation to reduce operational cost, reduce product development time to market and improve flexibility of the plant in manufacturing a range of products. The decentralization in the control systems has brought communication and coordination amongst various sub-control systems to surface as an important aspect in building an optimal control system design. The research and development in the area of DCS network design is quite active because of developments in the communications and information technology field. As an example, the authors in (O'Hearn et al., 2002) discuss an approach for combination of

distributed control system with motor control center involving a number of remote terminal units within an industrial facility.

The chapter is organized as follows: In the next section, literature review of related work is presented to highlight existing approaches related to distributed process control and corresponding applications. The customary environment is discussed in detail along with performance specification. Additionally, a set of relevant standards are highlighted that set the direction for common approach using distributed control. In section 3, a set of approaches are described that maximize the performance of the central process controller. The section 4 presents recent advancements in communications and information technology that are proving to be better candidates to yield improved performance for process control. A set of approaches are described, which minimizes the communication delay, improve performance, project reconfigurability and interoperability amongst various process devices. The section 5 provides recent standardization efforts in this area. In section 6, conclusions are presented followed by references in section 7.

## 2. Process Control

The DCS consists of a network of small, compact and intelligent nodes placed throughout the environment and linked via a network to main process controller. Identical hardware simplifies installation and field replacement. The nodes interface with and control a variety of device types, while main controller oversees the operation. In DCS, multipoint and network based process control has replaced the traditional point-to-point architecture which had been used by the industry for decades. The new architecture has smaller volume of wiring and distributed processing resulting into quick and easy maintenance and low cost. These developments have initiated research in two main directions: (i) Network design, configuration and communication protocols for distributed process control (ii) DCS system hardware and software architectures. To abreast our selves of first direction of research, a research work in (Lian et al, 2001) is studied that carries out simulation and experimental evaluation of three commonly used control networks in the industry. The control networks evaluated were: Ethernet, ControlNet, and DeviceNet. The evaluation procedure considered how each control network is used as a communication backbone for a network controlled system connecting sensors, actuators, and controllers with a detailed discussion on how medium access control (MAC) protocol works for each network. The performance parameters addressed include access delay, transmission time, message delay, message collisions, message throughput, packet size, network utilization, and determinism boundaries. Timings analysis of each control network presented addresses blocking time, frame time and propagation time. Using some case studies, the authors concluded that DeviceNet outperforms the other two networks in case of shorter or prioritized messages, where as ControlNet is suitable in case of time critical and non-time-critical messages. Summarily, this research work provided insight in setting performance limits offered by respective networks.

The Profibus is the standard protocol for communication which is vendor independent so that the communication between devices from different manufacturers can be implemented without any special interface adjustment. Regarding the Profibus networks, a comprehensive study has been carried out in (Tovar & Francisco, 1999), where authors discuss how Profibus fieldbus networks can be used to support real time communication for

industrial systems. The major contribution noted in this research is that the real time behavior is guaranteed even in the absence of synchronous bandwidth allocation. The proposed approaches are based on an assumption based on which real time traffic is guaranteed. The author in (Hong, 2000) has performed experimental evaluation of Profibus-FMS. In this work the experimental model of a Profibus-based industrial application is developed. The experimental environment consists of devices connected in a Profibus network involving data exchange through the services provided by FMS. The model evaluates the delay characteristics of a message once it passes through FMS service. Based on the experimental results, the author suggests an optimal value of target token rotation and an appropriate segment size to reduce the delay due to variable messages and in domain management service of FMS respectively. This also concludes that Profibus-FMS service is greatly affected by conventional values of target rotation time and segment size.

Apart from consideration of network protocols or control networks with associated delay characteristics, a study has been carried out in (Lian et al, 2002) about the design considerations for a specific network like DCS. This research work highlights the impact of network architecture in Network Control System (NCS) class of DCS. Design considerations include network parameters, control parameters, and network controlled system performance. The main evaluation measures for the network quality of service (QoS) are: time delay statistics, network efficiency, network utilization, and the number of lost or unsent messages. The authors have used a simulation model and an experimental study to validate the performance characteristics shown by theoretical analysis. Based on results, the authors demonstrate that the performance characteristics are useful guidelines to choose the network and control parameters while designing an NCS. The authors suggest that the performance characteristics are dependent upon the sampling period. However, choosing a sampling period has not been addressed in the research.

The wireless Profibus integrating wireless and wired devices is also investigated for process control applications. A study has been carried out in (Willig, 2003), where the parameters of wireless Profibus protocol standard have been discussed. The author introduces two approaches to wireless Profibus and compares them with respect to real time performance in the presence of transmission errors. The work demonstrates that the performance of real time Profibus protocol in IEEE 802.11 type PHY environment or in presence of bursty errors is unsatisfactory. The author suggests a specifically tailored protocol for wireless stations in a situation where integration of wired and wireless stations is necessary in a common Profibus Local Area Network (LAN). The tailored protocol works on top of IEEE 802.11 PHY and offers the same link layer interface to upper layers as does the Profibus protocol. The author has shown that round robin protocol with three add-ons namely simple data relaying, simple poll relaying, and priority restriction outperforms Profibus protocol under bursty conditions, and a stable ring. However, it was also concluded that keeping logical ring stable in presence of transmission errors is difficult to achieve. In a similar effort using network based control, the authors (Almeida, et al., 2002) highlight the importance of operational flexibility as well as event and time triggered paradigms in fieldbus communication systems, and propose a protocol to support time-triggered communication in a flexible way. The authors also discuss an efficient combination of event and time-triggered traffic with temporal isolation, maintaining the desired properties of both types of the traffic.

Other than the domain discussed above, there exists a set of approaches to the same problem. In one of the approaches, the distributed real time control application development for task allocation in heterogeneous systems is also reported in (Prayati et al., 2004), where authors address real time requirements in presence of interoperability of networked devices. The approach uses functional block allocation (FBALL) algorithm (International Electrotechnical Commission, 2007) to guarantee real time requirements. However, this work does not consider network situation during re-configurability of the process itself or its field devices during real time. Motivated by advent of pervading communication and computing, the authors in (Recht & D'Andrea, 2004) consider spatially interconnected distributed systems with arbitrary discrete symmetry groups to discuss distributed controller design applicable to a much larger class of structure topologies. In another approach, the application domains and tasks faced by multi-robot systems of increasing complexity have been investigated. The authors (Farinelli et al., 2004) describe various forms of cooperation and coordination realized in the domain of multi-robot systems.

Although the literature was surveyed at length and considerable approaches were found related to network based control but the ones addressed in previous section are the most notable among many others. The issues addressed in literature cover network considerations (i.e., protocol change or delay characteristics), interoperability of devices; customization and reconfiguration of the control system. The approaches that address the network environment analyzed timing characteristics or protocols in order to reduce communication delays at various stages of information processing.

The process under our consideration is customary i.e., it comprises devices and nodes that are practically available today and mostly used. The environment can be visualized as shown in Figure 1. In this Figure, the controller is the main device for which distributed control environment is to be investigated. The complete system involves interconnection of control devices, some distributed field and intelligent devices connected to Profibus, and few intelligent control devices operating under legacy environment. The control devices at lower level of the system may involve standard PLCs for motion control of a typical drive, for example. The intelligent devices are normally needed for a specific application where some of the decisions and local information gathering is done at that level and for onward transmission to the main controller for final network control decision. The devices are assumed to be reprogrammable and reconfigurable to suit changing needs of the process control. The interface shown in the Figure 1 is for legacy intelligence devices which are now part of the main distributed control system. The network considered is typical with Profibus compatible network devices with wired or wireless connectivity to a remote device or a process controller. The PC shown is for management purpose, especially for SCADA system. All the field devices considered are IEC 61804 compliant, and PLCs are IEC 61131 compliant. Based on our discussion above, the requirements for DCS development for this process control include consideration of reconfigurability and intelligence in the form of framework within a Profibus compatible network.

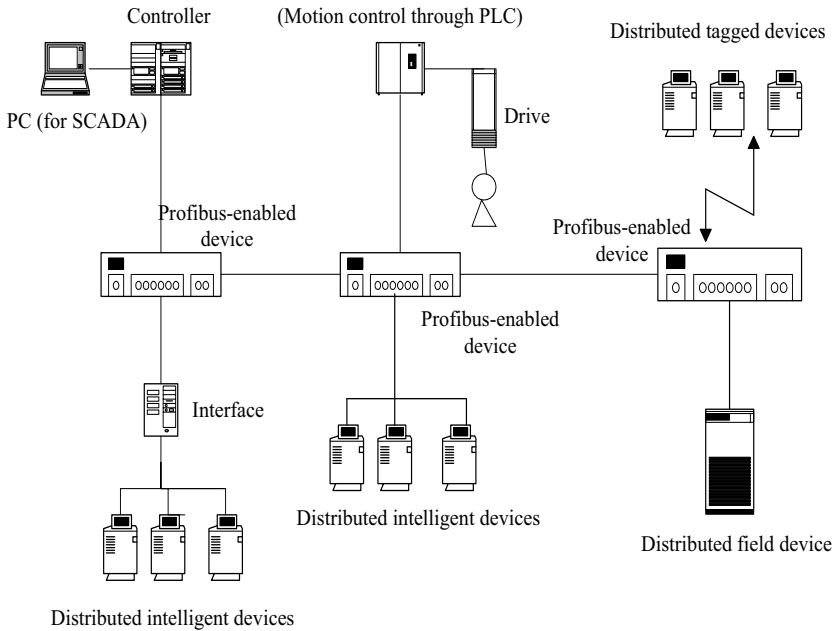


Fig. 1. Typical Process Control Network

For comparative purposes, we need to create a model that sets a baseline for performance. As the configuration of the network is distributed over a wide area, hence we consider a multiple input multiple output (MIMO) baseline that requires performance, which matches to that of the centralized MIMO (without network induced delays). A discrete time, causal and linear time-invariant systems with  $n$  states and  $m$  outputs can be described as (Goodwin et al., 2001):

$$X(k + 1) = A X(k) + B [U(k) + D(k)]; Y(k) = C X(k) \tag{1}$$

with  $U(z) = K(z) E(z) = K(z) [R(z) - (C X(z) + N(z))]$

where  $X, D, K, E, R, N$  are states of actual MIMO system in centralized control, disturbance, closed loop controller, error, reference, and sensor noise respectively. It should be noted that the MIMO system is assumed to be well designed meaning that its performance is exponentially stable (Goodwin et al., 2001), and that any MIMO control design technique can be used to develop an appropriate controller  $K$ . The parameters  $A, B, C$  are minimal realizations of the actual plant (Goodwin et al., 2001). Using equation (1), various states of the system can also be calculated:

$$X'(k + 1) = A' X'(k) + B' U'(k); Y'(k) = C' X'(k) \tag{2}$$

with  $U'(z) = K(z) E'(z) = K(z) [R(z) - C' X'(z)]$

where  $A', B', C'$  represent minimal realizations of the model. For Figure 1 (as a distributed implementation of MIMO) to achieve same performance to that of the centralized MIMO, each error or reconfiguration request needs to be communicated over the network at any sample time, and that there are no communication delays. In order to categorize time delay  $d(t)$  in the network is divided into three categories:

$$d(t) = d_1(t) \vee d_2(t) \vee d_3(t) \quad (3)$$

Thus, the total time delay  $d(t)$  is either of the time delays, depending on the prevailing situation in the environment. The  $d_1$  corresponds to a normal situation (Green level), when only one field device communicates with the central process controller for a possible update or reconfiguration at any time. The  $d_2$  corresponds to a situation (Yellow level), when more than one device (with number set as a threshold) communicate with central process controller for possible problem getting out of hand. The  $d_3$  corresponds to a situation (Red level), when a set of devices (with number set as a threshold) tries to communicate with the central process controller. One obvious direction could be to minimize the delays described in equation (3) to optimize the performance to match a centralized MIMO system with no communication delays.

Since field devices may include nodes or sensors that are non-stationary, protocol invariant and reconfigurable, the performance requirements stated by centralized MIMO would require approaches those are not based on legacy network control but should include recent advancements in sensor, information and communication technologies to match optimized performance. The combination of distributed control up to edges of the process containing sensors and tags, with recent information technology concepts like layered software and agents is an active area of research these days.

## 2.1 Relevant Standardization

The PLCs are the most widely used processing units and fieldbus is most widely used to interconnect the process controllers, sensors and actuators in a DCS system. The Profibus is the standard protocol for communication which is vendor independent so that the communication between devices from different manufacturers can be implemented without any special interface adjustment. To realize the plug and play type of operation, the DCS vendors ought to adopt certain standards both for PLCs and the communication. The International Electrotechnical Commission (IEC) and PLC manufacturing companies are actively involved in this development and establishment of standards (International Electrotechnical Commission, 2007). These standards define basic concepts for the design of modular, reusable, distributed components for the distributed control systems. Some of the most popular IEC standards are IEC 61131, IEC 61499 and IEC 61804. Because of the importance of these standards a brief introduction is presented in the next section.

**IEC 61804:** This standard addresses the need of common device description for representing the particular features of the field devices used in the DCS system. The standard is applicable to Function Blocks (FB) for process control and specifies the Electronic Device Description Language (EDDL) thus improving the device interoperability.

**IEC 61131:** This standard has different versions ranging from 61131-1 to 61131-8. The whole range covers general information about just the PLCs, the equipment requirement and tests, defines syntax and semantics of programming languages including the function block (FB) portion of standard for PLC languages, and sets the standards for the communication. To conclude IEC 61131 was first attempt to provide common terminology and reference model about hardware (HW) and software (SW) architectures, communication, and languages for a special class of control devices i.e., the PLCs.

**IEC 61193-2:** This standard IEC applies to the inspection of electronic components, packages, and modules for use in electronic and electric equipment, manufactured under

suitable process control, which prevents the outflow of nonconforming products. The objective is to provide a method for stabilizing, monitoring, and improving processes.

**IEC 61987-1:** It defines a generic structure in which product features of industrial- process measurement and control equipment with analogue or digital output should be arranged, in order to facilitate the understanding of product descriptions when they are transferred from one party to another. It applies to the production of catalogues of process measuring equipment supplied by the manufacturer of the product and helps the user to formulate requirements.

**IEC 61499:** This standard defines a generic architecture and presents guidelines for the use of function blocks as the main building block for industrial process measurement and control system applications. It extends the FB language (defined in IEC 61131-3) to more adequately meet the requirements of the distributed control in a format that is independent of implementation. The architecture is presented in terms of implement-able reference models, textual syntax and graphical representations. The standard addresses the need for modular software that can be used for distributed industrial process control by defining the basic concepts for the design of modular, reusable, distributed components for industrial control systems. It also specifies the software tool requirements.

A variety of fieldbus technologies and digital fieldbus devices have been introduced within the process industries over the last fifteen years. There has been a gradual acceptance of the fact that a variety of communication technologies are needed to fully address the application requirements of a manufacturing facility. Process control systems that use fieldbus, function blocks, and device descriptive languages already exist in the market. It is clear that the majority of these systems will continue to use function blocks into the foreseeable future, as they are modular and reusable. The function blocks hide the changing details and thus provide a stable interface for the application developers.

### 3. Sensors and Tagging Technology for Automation

The RFID (Radio Frequency Identification) technology enables events that are in line with the vision of pervasive computing: using digital media to augment physical and social spaces rather than replacing them with disembodied virtual spaces. The diversification of consumer needs has made RFID and other sensor systems an important means within the manufacturing industry to achieve more efficient multi-product production. Within the RFID tag process, for example, control functions may be stored and updated as the product progresses through various stages of the operation. Specific uses of the data within the tag for process control include part routing, program operations, build orders, component needs and specific operations which may be unique to only a few products. While operating in a process control function, the RFID tag can also be used as a portable data base providing station to station data transfers, repair information, operation status, operation values, pass/fail data, host data transfers, part history and raw SPC data. One of the major advantages of using an RFID system in manufacturing is a major reduction in the size and complexity of the PC or PLC program reducing the cost associated with programming and also reduces the scan time of the controller.

The object's RFID tag may be integrated with a large object database to integrate object community into the space where social interaction occurs (Konomi et al., 2006). The Researchers at the University of Washington are curious to see what effects RFID technology

could have on social networking (The RFID Ecosystem, University of Washington, 2008). To see what happens when the tags become ubiquitous, they installed two hundred antennae in and around a campus building and gave tags to twelve researchers. The result was such that their every move is recorded by computer. The system responds to nearby participants and uses algorithms to dynamically derive interconnected social clusters from a publication database - highlighting new opportunities and key design challenges. The Department of Computer Science and Engineering at the University of Washington has initiated a large scale RFID Ecosystem project to investigate user-centered RFID systems in connection with technology, business, and society and determine balance between privacy and utility (The RFID Ecosystem, University of Washington, 2008).

For vast field deployment, the authors (Bohn & Mattern, 2004) discuss distribution schemes where passive RFID tags are deployed in vast quantities and in random fashion. The authors suggest that certain deployment patterns help create novel RFID based services and applications. The density, structure of tag distributions, tag typing and clustering is also investigated by authors. For distributed business like process applications, the author (Palmer, 2007) discusses effective ways handling vast volume of RFID data for enterprise application management in real time. The steps applied together help enterprise manage volumes of raw RFID data in real time, provide reliability and accurate decisions to tackle the challenges, and exploit opportunities offered by RFID.

It is impossible to separate RFID technology from its issues and concerns surrounding its deployment, especially with respect to its potential for privacy infringement and identity theft. The RFID systems are designed to be asymmetric: tags are cheap and require low power while readers are expensive and power hungry. The various types and functionalities of these devices have been defined in an RFID class structure by the Auto-ID center, and later through Electronic Product Code (EPC) global (The EPC Global Standard, 2009), which has been subsequently refined and built-on. The number and use of standards within RFID and its associated industries is quite complex, involves a large number of bodies (ISO standard, 2009; ETSI standard, 2009; RFID Journal, 2009) and are in a continuous process of development. The various standards produced cover four key areas of RFID technology: its air interface (tag-reader communications) standards, data content and encoding, conformance and interoperability between different applications and RFID systems (RFID Journal, 2009). The various technological issues including its frequency range, standards, adoption and innovation has also been addressed at length in (Ward et al., 2006).

The RFID's have also been investigated in conjunction with software technology to optimize its usability. Integrating software agents into RFID architectures to accumulate information from tags and process them for specific use at different layers of the control plane has attracted many operational managers to investigate their deployment for real time use in industrial applications. The authors in (Naby & Giorgini, 2006) propose a possible integration between a Multi Agent framework and an RFID based application (back end). This is enabled by creating an agent for each person to develop its profile and correlating it with RFID tags movement inside a working environment. The accumulated information, after processing may be used to facilitate a concurrent mission. The RFID has also been explored for product identification in agent-based control applications to gain full control over all products and resources at shop floor level - aiming at installing a completely distributed agent-based production control system with full information control (Bratukhin & Treytl, 2006). The summary of literature survey for RFID deployment suggests



investigating a multi-agent architecture as an application layer within existing RFID layered architecture that includes tag layer, RFID-reader layer, and back-end layer (Naby & Giorgini, 2006).

#### 4. Agents for Intelligent Control

Recently there have been a number of advances in distributed artificial intelligence that provide the tools to move away from the traditional centralized, scan based programmable logic control (PLC) architecture towards a new architecture for the real-time distributed intelligent control. The Industrial control is typically implemented using large and often expensive hardware platforms that support monolithic computer control applications. The authors in (Bernnan et al., 2002) propose a general approach for dynamic and intelligent reconfiguration of real-time distributed control systems that utilizes the IEC 61499 function block model (International Electrotechnical Commission, 2007) to achieve shorter up-front commissioning time and significantly more responsive to changes, and is based on object oriented and agent based methods. Similarly, the authors in (Heck et al., 2003) discuss software technology for implementing reusable, distributed control components to make corresponding systems more flexible and dynamically re-configurable, so that they can be adapted in an easy way. Intelligent control and management of traffic and transportation systems in connected environments has been investigated in (Wang, 2005) that requires agent based technology to develop cheap, reliable and flexible control system. The authors in (Maturana et al., 2005) present distributive intelligent control architecture using agents by developing tools and methodologies for agent based solutions to a wide variety of industrial, financial and security problems, as classical control does not adapt dynamically well to the variability of the process. In another work, the authors in (Fregene et al., 2005) present a systems-and-control-oriented intelligent agent framework, as well as its composition into specific kinds of multi-agent systems to achieve the coordinated control of multiple multimode dynamical systems. The author in (Vyatkin, 2008) discusses distributed intelligent control using IEC 61499 (International Electrotechnical Commission, 2007) programming architecture for distributed automation systems in reconfigurable manufacturing systems. The testing environment consists of a network of identical software components for control of baggage handling system at the airport. The work in (Gingko Networks, 2008) provides simple decentralized ways to deal with a growing number of modern network requirements, by using distributed intelligent agents to deal with local situations in a more responsive way like sensing and observing events and changes occurring locally. In order to understand the potential use of agents in process industry, a survey has been presented in (Yang & Vyatkin, 2008) on design and validation of distributed control in process industry. The respective authors discuss distributed control benefits such as flexibility, reconfigurability and software reusability. The process control system is generally considered as a hybrid system as it usually contains both discrete and continuous dynamics. The authors highlight the importance of hybrid verification and simulation while handling validation of distributed systems.

In intelligent control design, scalability of a control system, active system reconfiguration, distributed intelligence and reduced communications are targeted benefits. In these situations, the issues like customization, decentralization and modularity at various levels of control are the main challenges in respective DCS design to facilitate distributed field

operations in a process control. Intelligent network control together with active remote monitoring of a scalable distributed system has turned out to be an active area of research. The obvious choice seems to break the complex control process into two distinct processes (Memon, 2008): Local Process for simple task execution at field device level using social interaction of field process entities and generating operational parameters at the device level; Central Process for estimation of new characteristics of social entities only when entities need to be created or modified during operational stages. The independence at local process level minimizes communication between devices to avoid bottlenecks arising due to interoperability of devices, network protocol change or simple operational requirements at the device level; and provide a degree of re-configurability of field devices at the same time. Using local intelligence collected through interactions of local entities and combined with controller requirements at the central level helps solve combinatorial complexity present at any time during the operation. Thus, two areas are targeted: social interactions of entities with domain intelligence, and effective decision making set by the central process. The following is the discussion on both processes.

#### 4.1 Local Process

Knowledge integration at central level requires balanced information from local entities. These entities tend to be distributed throughout the operational environment to support all operations. The job of these entities can be done effectively by distributed agents. The agents collaborate socially, learn and adjust their abilities within the constraints of the global process. Thus the agents are sophisticated software entities set by central process to execute trained intelligence at the local level. The functions within this process include: how agents collaborate and clusters are formed to accumulate intelligence and enable decision making; how access to another agent is facilitated; and the procedure for inter-cluster collaboration. To make sure that these agents form an intelligent system, the framework has to conform to four requirements (Maturana et al., 2005): that the system is decentralized; agents follow centrality; they use common language; and that the system is scalable.

##### 4.1.1 Agent design & management

The agents deployed in existing infrastructures are designed to be manageable as other components with tools and procedures already in place at central process level. Each agent has software components adaptable to environmental changes. Each of these can be considered as a specialized function with some built-in capabilities, like sense→decide→act loop. The various categories of these specialized functions include:

- Producing knowledge in cooperation with other agents
- Evaluating the situation and decide to apply an appropriate action individually or through collaboration
- Acting onto the network element parameters, like fine tuning quality of service
- Forward useful information to the central process

Agents provide simple decentralized ways to deal with a growing number of modern process control network requirements. Agents can provide a set of management functions:

- Configuration functions for downloading of configuration parameters
- Control functions for monitoring the process environment and providing on-line activation and inspection of agents

- Data collection functions for enabling uploading of useful information collected by agents

- Deployment functions enable to add or remove agents dynamically

Agents can also request for additional capabilities once they discover that the task at hand cannot be fulfilled with existing ones. The programming of these agents is done at the central level where a set of heuristics is used for reasoning at the local level, and is stored as a function block diagram (like an internal script). The life cycle of the cluster is pre-set at the central process during agent design, and which can be re-negotiated upon an agent's request. The agents know about their equipment, and continuously monitor their state, as shown in Figure 2. The Figure 2 shows agents residing on tagged network elements, known as intelligent devices that can compute and communicate. Based on the intelligence, they can decide whether to participate in a mission or not, as shown in Figure 3. Thus the design of an agent is normally made open ended to add flexibility during operation.

#### 4.1.2 Agent Collaboration

Each agent maintains its own view of the environment on the basis of the information obtained directly from its sensors and indirectly from the network through collaboration with its neighboring agents. As the local events are known and are properly documented in its view, the agent responds quickly and appropriately. Based on its view, agent may decide to automatically adapt its certain parameters of its own network element. The environmental view of the agent drives implicit cooperation amongst agents. This mode of cooperation is simple, robust and well situated for dynamically changing environments. The collaborating agents join (on their own will) and thus form a cluster in order to enable a decision making. Within their constraints, they accomplish their task. In addition to agents, there are other computing units that exist at the local level and which help to form a cluster. These are known as cluster directory (CD) and cluster facilitator (CF) respectively. The following steps describe the operational scene of agent collaboration in clusters:

- Agent N receives a request from central process for a task planning.
- It checks its internal scripts, if it can participate then it solves the local steps.
- For external steps, it contacts cluster directory to check for other agents if they also have external capability.
- In that case, CD provides contact details.
- Upon receipt of these details, agent N creates  $CF_N$  and passes on these details.
- $CF_N$  understands coordination context.
- $CF_N$  passes the request to specified agents, and thus cluster is formed.

For efficient collaboration, CD must remain updated for recording the information of its members, such as agent name, agent locator, service name, service type, and so on. Upon joining or leaving the cluster, an agent must register or cancel registration respectively through CF. Through a query, an agent can find out other members' services and locators. Through these steps, a trust is developed amongst agents.

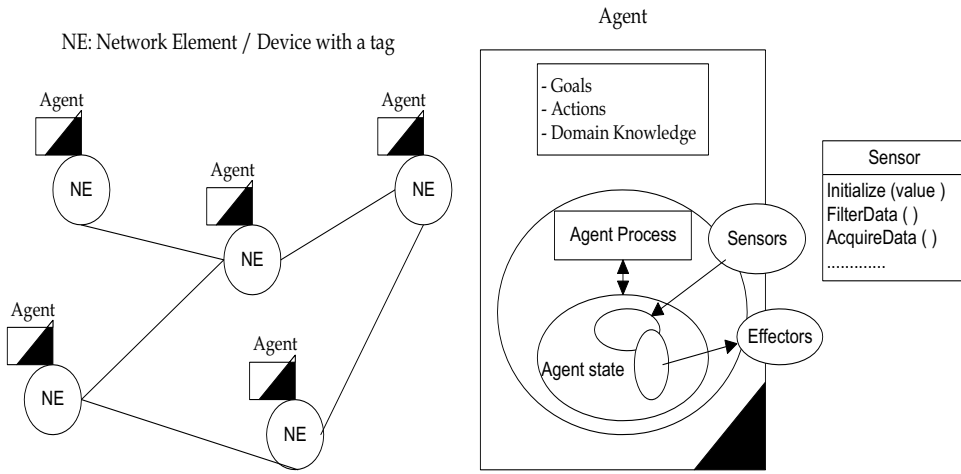
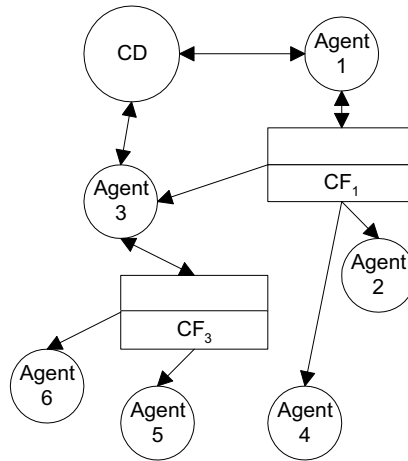


Fig. 2. Agents in a process network

Cluster 1: Agents (1,2,3,4); Cluster 2: Agents (3,5,6)



CD: Community Directory; CF: Community Facilitator

Fig. 3. Agent Collaboration

### 4.1.3 Agent Technology

Recently developed tools may be used to help design cluster facilitator (CF) and domain ontology, using for example DARPA Markup Language (DAML) (DAML website, 2000). The DAML extends XML (Extensible Markup Language) and RDF (Resource Description Framework) to include domain ontology. It provides rich set of constructs to create ontology and to markup information for attaining machine readability and understandability; it has capability of inference so that membership or service of cluster can be precisely defined. In

case of XML and RDF, there seems to be no significant literature to manage ontology, and this is why, DAML is chosen to build cluster ontology. A number of agent standards have been in practice like Knowledge Query Manipulation Language (KQML), OMG's mobile agent system interoperability facility (MASIF) and the Foundation for Intelligent Physical Agents (FIPA). The Foundation for Intelligent Physical Agent (FIPA) Agent Management Specification (FIPA specification, 2002) is extended to develop the agent role called CF to manage cluster directory (CD) and cluster ontology. Using assistance from DAML-based ontology, the members of the cluster are able to form cluster and communicate with other agents. The interaction among domain ontology, CD and CF can be best understood using Figure 4. The Figure 4 shows how CF gets access to DAML files and facilitates the common goal of the cluster. There are tools available like Jena semantic web (HP Labs toolkit, 2003) that can be used to handle the cluster directory (CD) built using DAML, and to develop a Java class "Directory".

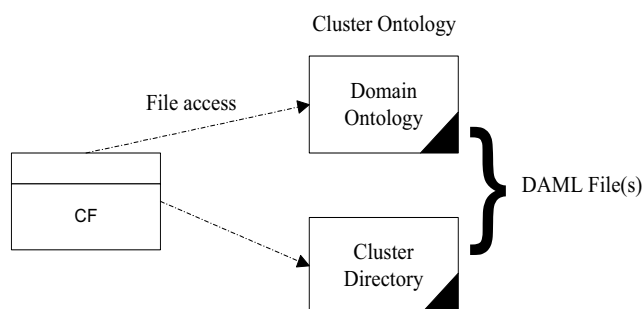


Fig. 4. Linking CF with DAML

From discussions above, the main functions of CD can be summarized, as:

- Add and Remove the information of an agent
- Get the list of agent names of all members
- Get the information of individual agent by name
- Get ontology used by members in the cluster
- Add external ontology if provided by an agent

Using local process mechanism and main functions of CD, the partial directory can be described as shown in Figure 5. It shows information of CF (lines 1-9) and members of cluster (lines 20-22), the cluster directory also records meta-data about cluster such as cluster name (line 12), cluster description (lines 13-15), ontology used in cluster (lines 16-18), and so on.

```

1. <cluster:CF rdf:ID="theCF">
2. <cluster:agentName>"CF"</cluster:agentName>
3. <cluster:agentDescription>
4. "DCS Cluster Facilitator"
5. </cluster:agentDescription>
6. <cluster:locator>
7. "http://dcs.ee.uae.ac.ae/DCS/agent/CF"
8. </cluster:locator>
9. </cluster:CF>
10.
11. <cluster:Cluster rdf:ID="DCScluster">
12.<cluster:clusterName>"DCS"</cluster:clusterName>
13. <cluster:clusterDescription>
14. "Distributed Control System"
15. </cluster:clusterDescription>
16. <cluster:ontology>
17. "http://dcs.ee.uae.ac.ae/DCS/ontology/dcs.daml"
18. </cluster:ontology>
19.
20. <cluster:hasCF rdf:Resource="#theCF"/>
21. <cluster:consistOf rdf:Resource="#agent1"/>
22. <cluster:consistOf rdf:Resource="#agent2"/>
23. </cluster:Cluster>

```

Fig. 5. DCS Cluster Directory

#### 4.1.4 Example

An example can be illustrated to show how ontology may be updated (Fig. 6(b)) and that how interactions may develop in a local process. It should be noted here that basic cluster ontology (the knowledge of the local process) provided by CF remains the same but all members' domain knowledge (ontology) may not be the same. For example, user agent holds basic knowledge of the local process but does not understand the knowledge that a distributed field device holds. Through DAML-based ontology, members can communicate with each other to acquire requested service, as shown in Figure 6. It is clear from the Figure 6 that when distributed field device agent joins the cluster, it informs CF about corresponding ontology it provides (Figure 6(a)). Thus the CF maintains local process ontology plus the distributed field device ontology. When a user agent wants to perform a task, it asks CF about domain ontology and the agents that provide external capability. In response, CF informs the user agent if ontology is to be acquired (Figure 6(c)). Thus, the user agent can communicate with the distributed field device agent (Figure 6(d)).

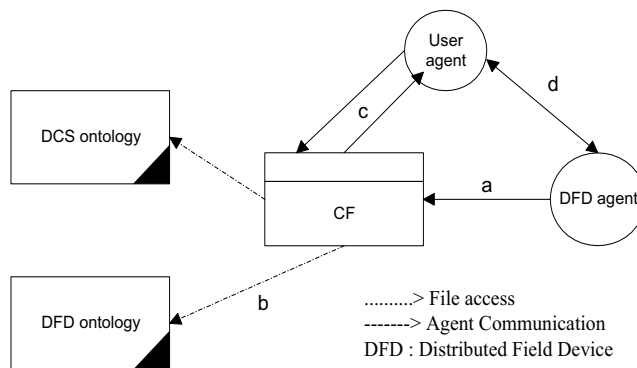


Fig. 6. Update in ontology provided by distributed field device agent

#### 4.2 Central Process

This process handles core mechanism that glues organization's local processes to the central process. Some of the functions for example library of agents, their job description, definition of controller tasks, and domain ontology of each cluster can be defined offline before the implementation actually starts. The dynamic components are removing of agent deadlocks, security of agents, estimation of characteristics and relationships and decision making in cases of emergencies and when situation develops beyond the capabilities of agent clusters. It seems that all of these dynamic functions together may require computations, but the advantages gained are many: (i) reduced communications between central process controller and the device(s) (ii) provide simplicity to enable better interoperability (iii) intelligence gathering to build a degree of reconfigurability in a case estimated parameters exceed beyond a limit (iv) reduced human supervision. It can be also argued that complexity of this process is only a technology mismatch, and that if only small scale changes are to be decided at the central process like reconfiguration of device parameters, security of agents, then intelligence can further be distributed to the agents at the local level. Based on presented work in section 3 and in 4.1, agents can be embedded in tagged devices within a layered architecture to support business operations and services in real time. In Figure 7, the model architecture of four tiers is drawn to implement objectives of the central process. At the bottom layer (Tier 1), active readers or Profibus/Profinet enabled devices collect data, often collected on a trigger similar to a motion sensor. These readers should be controlled by one and only one edge server to avoid problems related to network partitioning. In addition, this layer supports the notion that intelligence be introduced at the edges to reduce data traffic and improve reaction at the next layer. This layer also provides hardware abstraction for various Profibus/Profinet compatible hardware and network drivers for interoperability of devices. The edge sever (Tier 2) regularly poll the readers for any update from device agents, monitors tagged devices and distributed devices through readers, performs device management, and updates integration layer. This layer may also work with system through controls and open source frameworks that provide abstraction and design layer. The integration layer (Tier 3) provides design and engineering of various objects needed for central controller as well as for field processes and for simulation levels of reconfigurability. This layer is close to business application layer (Tier 4). The monitoring of

agents behavior, its parameters and cluster characteristics are done at this layer to assess the degree of reconfigurability. This layer also takes care of parameters like handling device processes, applications, security of agents, resource allocation and scheduling of processes.

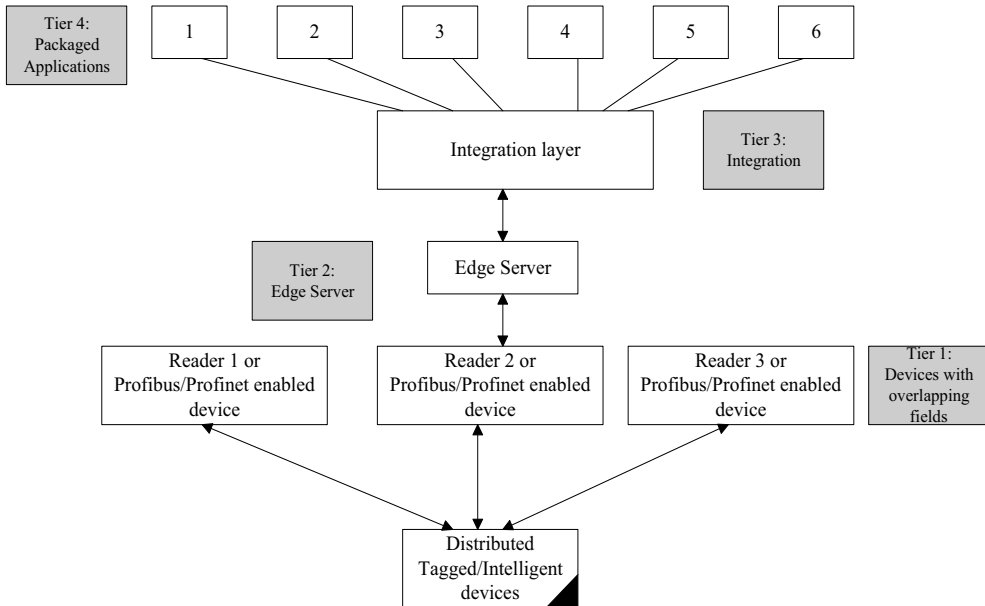


Fig. 7. 4-Tier Reference Architecture

The separation of edge server and integration layer improves scalability and reduces cost for operational management, as the edge is lighter and less expensive. The processing at the edge reduces data traffic to central point and improves reaction time. Similarly, the separation of integration from business applications helps in abstraction of process entities. The Tier 3 also enables it as self-healing and self-provisioning service architecture to increase availability and reduce support cost. Control messages flow into the system through business application portal to the integration layer, then on to the edge and eventually to the reader. Provisioning and configuration is done down this chain, while reader data is filtered and propagated up the chain.

The equation (3) may now be investigated again further, and evaluated using the model architecture, shown in Figure 7. The objective is to minimize the communication between field devices and the central process controller, and bring most of the local decision making intelligence at the local field level. Only when certain parameters need to be changed at local level device, then the values  $d_1$ ,  $d_2$ , and  $d_3$  need to be estimated. In order to evaluate these delay parameters, it is sufficient to estimate one communication between an RFID reader and the central process controller (i.e.,  $d_1$ ), as others just accumulate these delays over a number of communications. The communication between various nodes in Figure 7 may be abstracted using queuing models. In order to evaluate the performance modeling of this



architecture, queuing model can be used, since communication traffic may be considered as datagram that involve traversing multiple paths, which means M/M/1 queuing system can be used. Assume that the service rate between a reader and edge server is  $\mu_{RE}$ , between an edge and integration layer is  $\mu_{EL}$ , and that between integration and central process at packaged application is  $\mu_{IC}$  respectively. Assume also that datagram arrival rate from an RFID reader is  $\lambda_{RE}$  and  $T_{ij}$  is the average propagation delay between nodes  $i$  and  $j$ , where nodes  $i$  and  $j$  belong to one hop delay between any two nodes. Using these parameters, the RFID message delay from reader to central process controller may be written as accumulated delays across the entire path:

$$d_1 = d(R, \text{controller}) = \sum_{i,j} \left( \frac{\lambda_{ij}}{\mu_{ij}(\mu_{ij} - \lambda_{ij})} + \frac{1}{\mu_{ij}} + T_{ij} \right) \quad (4)$$

With Giga bits per second wireless transmission rates available today,  $T_{ij}$  may be assumed negligible for one datagram traversing from one node to the other, along the entire path. The only terms left are the arrival rates and service rates along all node hops. Since 100 Mbps system (i.e., 0.014ms/message) is commonly available for RFID switches, network switches and servers (with exponential service waiting time), we may assume the corresponding values in equation (4) very easily. It may also be assumed that central process controller server message receiving rate at is 0.014ms/message, based on the same criterion. Thus,  $d_1$  may be estimated once we insert  $\lambda_{ij}$  in the equation (4). It turns out that for given typical value of  $\lambda$  from reader as (say) 0.2, the estimated delay from a reader to central process controller is less than 0.1msec, which is acceptable for a central process controller that is waiting to update a set of parameters for agents down the local process. For a set of RFID tags (say 50) generating communication signals to central process controller (i.e., Yellow level situation), the estimated delay is still few milliseconds. For a situation involving 1000 tags generating messages (i.e., Red level), the total estimated delay is still less than a second.

### 4.3 Performance Gains

As presented in section 4.2, the communication delay has largely been reduced at the cost of increased intelligence at the local level. In fact, if we look at equation (3) we see that  $d_1(t)$ ,  $d_2(t)$  and  $d_3(t)$  minimize to a level when problem of the node device exceeds the threshold level of the agent intelligence. Thus this approach sets practical performance limits. However, again this is just a technology mismatch. If agent design technology reaches its maturity i.e., if the collaborative intelligence within agents exceeds combinatorial complexity of device problems then there is no need of communication between devices and the controller. Thus the requirements of the central process reduce to that of the customized design of the agents only, and its performance matches to that of the centralized MIMO system. The mechanism set at the local process provides self-healing, reliability and scalability. If a reader or service goes down, additional units can take up the workload automatically. If bottlenecks develop, the RFID system software can dynamically provision new service agents to manage increased requirements. The scalability is assured by a design at the central process that grows horizontally and vertically - like a single-CPU, tag-and-ship pilot through N-way and multi-purpose device deployments, smoothing the growth path. At the central process, design and reconfigurability can help introduce features in agents to thwart external and intrusive agents, and thus help boost security of operational devices and processes during real time. This set of gains has not been addressed in either of

the approaches described in (Konomi et al., 2006; Lian et al., 2002; Maturana et al., 2005; Prayati et al., 2004). The combination of agents and tagging technology uses programming and standardized components, which adds versatility to the process control. This type of process control is suited to a wide range of applications that need wide area sensing, and control points. The exploitation of agents is expected to rise over time as other enabling technologies grow in prominence.

## 5. Recent Standardization

There has been a large standardization effort conducted towards process control communications and systems, covering a range of industries. It is not possible to describe all of them here, but most recent, relevant to this work is presented below (International Electrotechnical Commission standard, 2006-2009; Hart Communication Foundation standard, 2009; Aim Global RFID Guideline, 2009):

**IEC 60770-3 (2006):** The standard specifies the methods for reviewing the functionality and the degree of intelligence in intelligent transmitters, for testing the operational behavior and dynamic performance of an intelligent transmitter as well as methodologies for determining the reliability and diagnostic features used to detect malfunctions; and determining the communication capabilities of the intelligent transmitters in a communication network.

**IEC 69870-5-104 (2006):** The standard defines telecontrol companion standard that enables interoperability among compatible telecontrol equipments. It applies to telecontrol equipment and systems with coded bit serial data transmission for monitoring and controlling geographically widespread processes.

**IEC 61784-1-3 (2007):** It defines a set of protocol specific communication profiles based primarily on the IEC 61158 series, to be used in the design of devices involved in communications in factory manufacturing and process control. It contains a minimal set of required services at the application layer and specification of options in intermediate layers defined through references.

**IEC 62264-3 (2007):** It defines activity models of manufacturing operations management that enable enterprise system to control system integration. The activities defined are consistent with the object models definitions given in IEC 62264-1. The modeled activities operate between business planning and logistics functions, defined as the Level 4 functions and the process control functions, defined as the Level 2 functions of IEC 62264-1. The scope of this standard is limited to: - a model of the activities associated with manufacturing operations management, Level 3 functions; - an identification of some of the data exchanged between Level 3 activities.

**Hart 7.0 (2007):** The Hart Communication Foundation (HCF) has released the HART 7 specification, enabling more capabilities for communication with intelligent field devices, and targeting wireless communication in industrial plant environment. The specification allows building on established and field-proven international standards including IEC 61158, IEC 61804-3, IEEE 802.15.4 radio and frequency hopping, spread spectrum and mesh networking technologies.

**IEC 61298-1-4 (2008):** The specification defines general methods and procedures for conducting tests, and reporting on the functional and performance characteristics of process measurement and control devices. The methods and procedures specified in this standard are applicable to any type of process measurement and control device. The tests are

applicable to any such devices characterized by their own specific input and output variables, and by the specific relationship (transfer function) between the inputs and outputs, and include analogue and digital devices.

**IEC 62424 (2008)E:** It specifies how process control engineering requests are represented in a P&ID for automatic transferring data between P&ID and PCE tool and to avoid misinterpretation of graphical P&ID symbols for PCE. It also defines the exchange of process control engineering data between a process control engineering tool and a P&ID tool by means of a data transfer language (called CAEX). These provisions apply to the export/import applications of such tools.

**IEC/PAS 62443-3 (2008):** It establishes a framework for securing information and communication technology aspects of industrial process measurement and control systems including its networks and devices on those networks, during the operational phase of the plant's life cycle. It provides guidance on a plant's operational security requirements and is primarily intended for automation system owners/operators (responsible for ICS operation).

**IEC 61850 (2009):** This is a standard for the design of electrical substation automation. Multiple protocols exist for substation automation, which include many proprietary protocols with custom communication links. The objectives set for the standard are: a single protocol for complete substation, definition of basic services required to transfer data, promotion of high interoperability between systems from different vendors, a common method/format for storing complete data, and define complete testing required for the equipments which confirms to the standard.

**IEC/PAS 62601 (2009):** It specifies WIA-PA system architecture and communication protocol for process automation based on IEEE 802.15.4. WIA-PA network is used for industrial monitoring, measurement and control applications.

**AIM Global RFID Guideline 396 (2008):** This guideline describes RFID chips and transponders, verification and qualification of design and manufacture of chips. This guideline targets item level tagging where the RFID tag may be present in various formats including a label, incorporated into a patch, which then becomes permanently affixed to the inner or outer surface of a tire or incorporated during manufacture into the structure of the tire as an integral part of the tire.

## 6. Conclusions

The main idea behind two processes is decentralization. The communication delay is reduced at the cost of increased intelligence at the local level. In fact, by looking at equation (1) it is clear that  $d_1(t)$ ,  $d_2(t)$  or  $d_3(t)$  minimize to a level when problem of the node device exceeds the threshold level of the agent intelligence. If collaborative intelligence exceeds combinatorial complexity then there is no need of communication between devices and the controller and requirements of the central process reduce to that of the design of agents only. Thus, the performance matches to that of the centralized MIMO system. The four-tier modular architecture at central level helps in implementation of distributed intelligence at field level and in designing of agents. The functionality more appropriate to the layer has been fit into respective tiers at central level. Additionally, design and reconfigurability can help introduce features in agents to thwart intrusive agents, during real time. It was also shown that the estimated delay due to communication from a tagged device to a central

process controller is less than a second when one thousand tagged devices pass on their communication signal to central process controller at the same time. This set of gains has not been claimed in either of the approaches for distributed control system widely discussed in the literature.

## 7. References

- Almeida, L., Pedreiras, P., & Fonseca, J. (2002). The FFT-CAN Protocol: Why and How, *IEEE Transactions on Industrial Electronics*, Vol. 49, No. 6, pp. 1189-1201, December, 2002
- Alonso, J., Ribas, J., Coz, J., Calleja, A., & Corominas, E. (2000). Development of a Distributive Control Scheme for Fluorescent Lighting based on LonWorks Technology, *IEEE Transactions on Industrial Electronic*, Vol. 47, No. 6, pp. 1253-1262, December, 2000
- Association for Automatic Identification and Mobility, AIM Global radio frequency identification (RFID) Guideline REG 396, [www.aimglobal.org/](http://www.aimglobal.org/), [last accessed 04/25/2009]
- Bernnan, Fletcher, M., & Norrie, D. (2002). An Agent-Based Approach to Reconfiguration of Real-time Distributed Control Systems, *IEEE Transactions on Robotics and Automation*, Vol. 18, No. 4, pp. 444-451, August, 2002
- Bohn, J., & Mattern, F. (2004). Super-Distributed RFID Infrastructures, *Lecture Notes in Computer Science (LNCS)* No. 3295, pp. 1-12, Eindhoven, Netherlands, November 8-10, Springer-Verlag, 2004
- Bratukhin, A., & Treytl, A. (2006). Applicability of RFID and Agent-Based Control for Product Identification in Distributed Production, *Proceedings of IEEE Conference on Emerging Technologies and Factory Automation*, Vol. 20, Issue 22, pp. 1198-1205, Prague, 2006
- Cavinato, M, Manduchi, G., Luchetta, A., & Taliercio, C. (2006). General-Purpose Framework for Real Time Control in Nuclear Fusion Experiments, *IEEE Transactions on Nuclear Science*, Vol. 53, No. 3, pp. 1002-1008, June, 2006
- DAML website, DARPA Agent Markup Language (2000) [Online], <http://www.daml.org/>; [last accessed 04/22/2006]
- ETSI RFID standards by ETSI, <http://www.etsi.org/WebSite/Standards/Standard.aspx>, Available online [last accessed on 04/25/2009]
- Farinelli, A., Iocchi, L., & Nardie, D. (2004). Multirobot Systems: A Classification Focused on Coordination, *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 34, No. 5, pp. 2015-2028, October, 2004
- FIPA website, Foundation for Intelligent Physical Agents (FIPA) Agent Management Specification (2002) [Online], <http://www.fipa.org/specs/fipa00023/>; [last accessed 09/21/2006]
- Fregene, K., Kennedy, D., & Wang, D. (2005). Toward a Systems- and Control-Oriented Agent Framework, *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 35, No. 5, pp. 999-1012, October, 2005
- Function Blocks (FB) for industrial-process measurement and control systems, IEC 61804, IEC 61131, and IEC 61499. [http://www.iec.ch/searchpub/cur\\_fut.htm](http://www.iec.ch/searchpub/cur_fut.htm), [last accessed 02/26/2009]

- Gingko Networks (2008). White Paper: Gingko Distributed Network Piloting System, *Gingko Networks*, September, 2008
- Goodwin, G., Graebe, S., Salgado, M., (2001), *Control System Design*, New Jersey, Prentice Hall, 2001
- Hart Communication Foundation website, Hart 7.0 Wireless Communication Protocol, [http://www.hartcomm2.org/hart\\_protocol/wireless\\_hart/hart7\\_overview.html](http://www.hartcomm2.org/hart_protocol/wireless_hart/hart7_overview.html), [last accessed 04/27/2009]
- Heck, B., Wills, L., & Vachtsevanos, G. (2003). Software Technology for Implementing Reusable, Distributed Control Systems, *IEEE Control Systems Magazine*, pp. 21-35, February, 2003
- Hong, S. (2000). Experimental Performance Evaluation of Profibus-FMS, *IEEE Robotics and Automation Magazine*, pp. 64-72, December, 2000
- HP Labs website, HP Labs (2003), [Online], Jena Semantic Web Toolkit Available: <http://www.hpl.hp.com/semweb/jena.htm/>; [last accessed September 2006]
- International Electrotechnical Commission (IEC, 2007) Webstore, <http://webstore.iec.ch/webstore/webstore.nsf/artnum/> [last accessed 04/28/2009]
- Ioannides, M. (2004). Design and Implementation of PLC based Monitoring Control System for Induction Motor, *IEEE Transactions on Energy Conversion*, Vol. 19, No. 3, pp. 469-476, September, 2004
- ISO RFID standards, <http://www.iso.org/rfid>, Available online [last accessed on 04/25/2009]
- Kleines, H., Sarkadi, J., Suxdorf, F., & Zwoll, K. (2004). Measurement of Real Time Aspects of Simatic PLC Operation in the Context of Physics Experiments, *IEEE Transactions on Nuclear Science*, Vol. 51, No. 3, pp. 489-494, June, 2004
- Konomi, S., Inoue, S., Kobayashai, T., Tsuchida, M., & Kitsuregawa, M. (2006). Supporting Colocated Interactions Using RFID and Social Network Displays, *IEEE Pervasive Computing Magazine*, Vol. 5, Issue 3, pp. 48-56, July-September, 2006
- Lian, F., Moyne, J., & Tilbury, D. (2001). Performance Evaluation of Control Networks, *IEEE Control Systems Magazine*, pp. 66-83, February, 2001
- Lian, F., Moyne, J., & Tilbury, D. (2002). Network Design Consideration for Distributed Control Systems, *IEEE Transactions on Control Systems Technology*, Vol. 10, No. 2, pp. 297-307, March 2002
- Maturana, F., Staron, R., & Hall, K. (2005). Methodologies and Tools for Intelligent Agents in Distributed Control, *IEEE Intelligent Systems*, pp. 42-49, February, 2005
- Memon, Q. (2008). A Framework for Distributed and Intelligent Process Control, *Proceedings of 5<sup>th</sup> International Conference on Informatics in Control, Automation and Robotics*, pp. 240-243, May 12-15, Madeira, Portugal, 2008
- Naby, A. & Giorgini, S. (2006). Locating Agents in RFID Architectures, Technical Report No. DIT-06-095, University of Trento, Italy, 2006
- O'Hearn, T., Cerff, J., & Miller, S. (2002). Integrating Process and Motor Control, *IEEE Industry Applications Magazine*, pp. 61-65, August, 2002
- Palmer, M. (2007). Seven Principles of Effective RFID Data Management, *A Technical Primer, Real Time Division, Progress Software, Inc.*, <http://www.progress.com/> [last accessed 03/21/2009]
- Prayati, A., Koulamas, C., Koubias, S., & Papadopoulos, G. (2004). A Methodology for the Development of Distributed Real-time Control Applications with Focus on Task

- Allocation in Heterogeneous Systems, *IEEE Transactions on Industrial Electronics*, Vol. 51, No. 6, pp. 1194-1206, December, 2004
- Recht, B. & D'Andrea, R. (2004). Distributed Control of Systems over Discrete Groups, *IEEE Transactions on Automatic Control*, Vol. 49, No. 9, pp. 1446-1452, September, 2004
- RFID Journal, A Summary of RFID Standards, <http://www.rfidjournal.com/article/articleview/609/1/1/> [last accessed on 04/25/2009]
- Stewart, G., Gorinevsky, D., & Dumont, G. (2003). Feedback Controller Design for a Spatially Distributed Systems: The Paper Machine Problem, *IEEE Transactions on Control Systems Technology*, Vol. 11, No. 5, pp. 612-628, September, 2003
- The EPC global, <http://www.epcglobalinc.org>, Available online [last accessed on 04/25/2009]
- The RFID Ecosystem, <http://rfid.cs.washington.edu/>, [last accessed on 04/24/2009]
- Tovar, E., & Francisco, F. (1999). Real-time field bus communications using Profibus Networks, *IEEE Transactions on Industrial Electronics*, Vol. 46, No. 6, pp. 1241-1251, December, 1999
- Vyatkin, V. (2008). Distributed IEC 61499 Intelligent Control of Reconfigurable Manufacturing Systems, *Technical Report*, University of Auckland, 2008
- Wang, F. (2005). Agent-Based Control for Networked Traffic Management Systems, *IEEE Intelligent Systems*, pp. 92-96. October, 2005
- Ward, M., Kranenburg, R., & Backhouse, G. (2006). RFID: Frequency, Standards, Adoption and Innovation, *Technical Report in JISC Technology and Standards Watch*, May, 2006
- Willig, A. (2003). Polling based MAC protocols for improving real-time performance in a wireless Profibus, *IEEE Transactions on Industrial Electronics*, Vol. 50, No. 4, pp. 806-817, August, 2003
- Yang, C., Vyatkin, V. (2008). Design and validation of distributed control with decentralized intelligence in process industries: A survey, *IEEE International Conference on Industrial Informatics*, pp. 1395 - 1400, Daejeon, July, 2008
- Yang, Y. (2006). Attenuation Splice Control in the Manufacture of Fiber Optical Communication System, *IEEE Transactions on Control Technology*, Vol. 14, No. 1, pp. 170-175, January, 2006



## **Automation Control - Theory and Practice**

Edited by A D Rodi

ISBN 978-953-307-039-1

Hard cover, 350 pages

**Publisher** InTech

**Published online** 01, December, 2009

**Published in print edition** December, 2009

The present edited book is a collection of 18 chapters written by internationally recognized experts and well-known professionals of the field. Chapters contribute to diverse facets of automation and control. The volume is organized in four parts according to the main subjects, regarding the recent advances in this field of engineering. The first thematic part of the book is devoted to automation. This includes solving of assembly line balancing problem and design of software architecture for cognitive assembling in production systems. The second part of the book concerns different aspects of modelling and control. This includes a study on modelling pollutant emission of diesel engine, development of a PLC program obtained from DEVS model, control networks for digital home, automatic control of temperature and flow in heat exchanger, and non-linear analysis and design of phase locked loops. The third part addresses issues of parameter estimation and filter design, including methods for parameters estimation, control and design of the wave digital filters. The fourth part presents new results in the intelligent control. This includes building a neural PDF strategy for hydroelectric saturation simulator, intelligent network system for process control, neural generalized predictive control for industrial processes, intelligent system for forecasting, diagnosis and decision making based on neural networks and self-organizing maps, development of a smart semantic middleware for the Internet, development of appropriate AI methods in fault-tolerant control, building expert system in rotary railcar dumpers, expert system for plant asset management, and building of a image retrieval system in heterogeneous database. The content of this thematic book admirably reflects the complementary aspects of theory and practice which have taken place in the last years. Certainly, the content of this book will serve as a valuable overview of theoretical and practical methods in control and automation to those who deal with engineering and research in this field of activities.

### **How to reference**

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Qurban A Memon (2009). Intelligent Network System for Process Control: Applications, Challenges, Approaches, Automation Control - Theory and Practice, A D Rodi (Ed.), ISBN: 978-953-307-039-1, InTech, Available from: <http://www.intechopen.com/books/automation-control-theory-and-practice/intelligent-network-system-for-process-control-applications-challenges-approaches>

**INTECH**  
open science | open minds

### **InTech Europe**

University Campus STeP Ri

### **InTech China**

Unit 405, Office Block, Hotel Equatorial Shanghai

Slavka Krautzeka 83/A  
51000 Rijeka, Croatia  
Phone: +385 (51) 770 447  
Fax: +385 (51) 686 166  
[www.intechopen.com](http://www.intechopen.com)

No.65, Yan An Road (West), Shanghai, 200040, China  
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元  
Phone: +86-21-62489820  
Fax: +86-21-62489821



© 2009 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the [Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License](#), which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.