Communication Architectures for Electrical Drives

Francesco Benzi, *Member, IEEE*, Giuseppe S. Buja, *Fellow, IEEE*, and Max Felser, *Member, IEEE*

***Abstract*—The paper gives an overview of the communication architectures adopted in the industrial automation for the electrical drives, ensuring a fast data exchange and high performance control. An attempt is made at defining real-time operation for this application field, at reviewing the standardization work done to unify the electrical drive interfaces, and at encompassing the recently accepted solutions, including those based on the Industrial Ethernet.**

***Index Terms*—Communication architectures, electrical drives, fieldbuses, real-time systems.**

I. Introduction

THE EXTENSIVE integration of industrial processes, with the corresponding increase in the volume of exchanged data, has produced significant novelties in the factory communication equipment, which has moved from the point-to-point lines transmitting analogue quantities toward serial buses transmitting digital signals, i.e., the fieldbuses. With their advent, configuration and operation of the control systems have become strictly dependent on the architectural solutions chosen for integrating the industrial process.

The relevant international standard committees had the quite unrewarding task of unifying several industrial solutions, each strongly supported by established companies. As a matter of fact, the International Committees involved in the fieldbus protocols have only partially succeeded, ending up with a compromise solution. In practice, the idea of "variant within a standard" has been accepted, which actually means to acknowledge an equal status for a number of protocols that have a few common elements, with the consequence of threatening the interoperability among components that, in principle, conform to the same standard. As a matter of fact, under the standard IEC61158/ISA SP50.02, different protocols coexist, namely, ControlNet, Profibus, P-Net, Fieldbus Foundation HSE, Swiftnet, WorldFIP, and Interbus [1]. The European EN 50170, in turn, includes as variants P-Net, Profibus, WorldFIP, Foundation Fieldbus, Profibus-PA, and ControlNet [2].

The efforts of the committees—even if not concluded with a real unification—had the merit of restraining the different solutions within a limited range, thus addressing users, device manufacturers, and integration experts toward a significant (even if not complete) number of alternative choices among components and solutions [3].

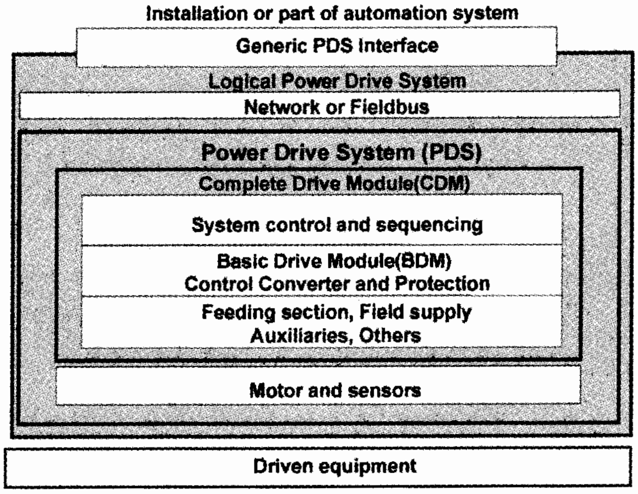


Fig. 1. Definition of PDS.

Standardization activities in integrating electrical drives have brought to IEC 61 800-7 [4], a standard of IEC 61 800 series intended to describe a generic interface between automation system and Power Drive Systems (PDSs), as shown in Fig. 1. The interface also can be embedded in the control system on board of the drive, and the resulting PDS is sometime known as a "smart drive" or "intelligent drive."

A variety of physical interfaces are available for PDS: analogue and digital, serial and parallel, fieldbuses, and networks. Profiles based on specific physical interfaces do already exist for some application areas (e.g., motion control) and some device classes (e.g., standard drives, positioners), but the associated drivers with the application programs are proprietary and vary widely. Thus, a number of troubles arise specially for system integrators. For example, some control devices support only a single interface, which cannot be implemented in a specific drive; on the other hand, functions and data structures can be specified with incompatibilities, and it is up to the system integrator to write the interfaces to the application, although this task should not be part of his duty. Moreover, some applications need exchangeability of the devices or integration of new devices; if they do not fit each other and the effort to adapt a solution to a drive profile or to manufacturer-specific extensions is unacceptable, the integrator is forced to adopt the physical interface supported by a given controller.

The advantages of defining a generic PDS interface are then well understood. For a drive device manufacturer, it implies the facilitation of the task of the system integrators in using drive functions because of a common frame and terminology. Moreover, the selection of drives does not depend on the availability of specific supports; in fact, the need for adaptation is reduced

Manuscript received September 23, 2004; revised January 18, 2005.

F. Benzi is with the Department of Electrical Engineering, University of Pavia, Pavia, Italy (e-mail: francesco.benzi@unipv.it).

G. S. Buja is with the Department of Electrical Engineering, University of Padova, 35131 Padova, Italy (e-mail: g.buja@ieee.org; buja@die.unipd.it).

M. Felser is with the School of Engineering and Information Technology, Division of Electrical and Communication Engineering, Berne University of Applied Sciences, Berne, Switzerland (e-mail: max.felser@hti.bfh.ch).

Digital Object Identifier 10.1109/TII.2005.844428

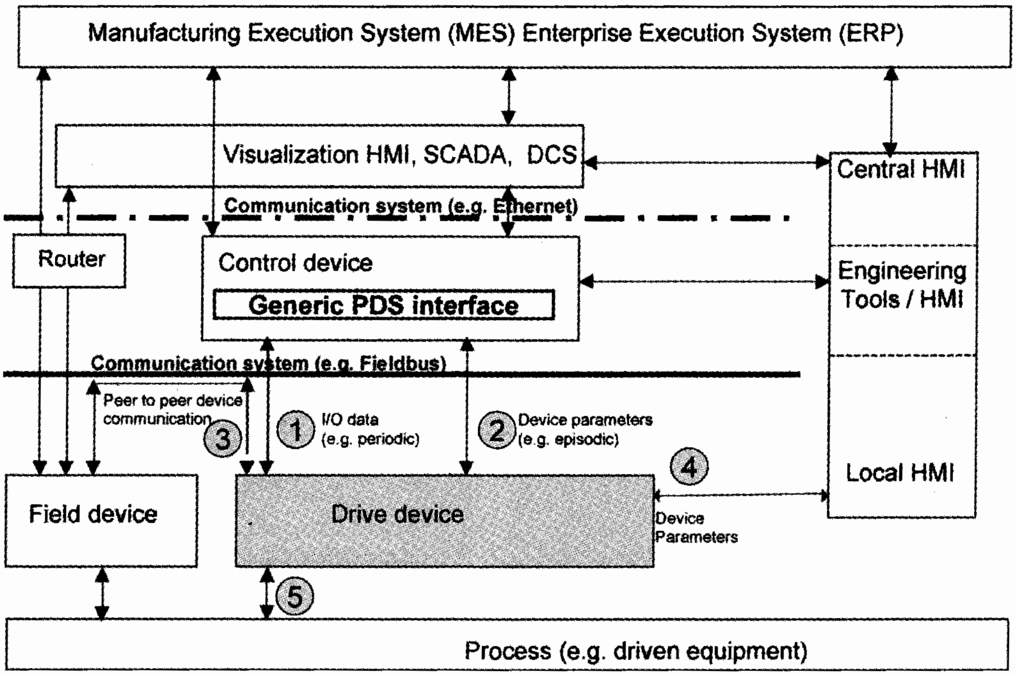


Fig. 2. General architecture (adapted from [2]).

when only the mapping has to be exchanged instead of implementing in the operating system the whole communication services together with the drivers and the supports for the configuration tools. For a control device manufacturer, there is no dependence on the bus technology and the drive supplier, leading to an easier design and implementation. For a system integrator, who builds modules, machines, plants, etc., there is less integration effort because there is only one reasonable way of modeling the connection; the devices mostly share the same physical interface, and a substantial independence on the bus technology is ensured.

From this perspective, a significant issue is the definition of architectures that are able to support the process control right down to the field level, by offering solutions with different complexity and performance for both single electrical drives, typically utilized in the process industry, and multiple electrical drives, typically utilized in the manufacturing industries.

In this paper, recent advancements in integrating and networking electrical drives are presented [5]. The presentation is focused on the fieldbus solutions developed for the real-time control of both single-drive and multiple-drive systems. Beside fieldbuses, an approach of increasing interest is the extension of the protocols underlying the Internet world (e.g., Ethernet and TCP/IP) to the industrial automation. Such an approach is also addressed in the paper.

II. BUS PERFORMANCE FOR REAL-TIME CONTROL

A general and comprehensive architecture of an automation system with PDS, which was adapted from the recent documents released by IEC 61 800-7, is shown in Fig. 2 (an explanation of the circled numbers is given later). It includes a management level with MES and ERP functions, an intermediate distribution level mostly implemented by means of SCADA and DCS systems, a controller level with PLC, PC, or other types of controllers, and a field level next to the process. The Human Machine Interface (HMI) is distributed at different levels.

Different communication systems are involved in the architecture: At the higher level, a *system or supervisory bus* with an extended framework length suitable for general-purpose exchange of structured messages, neither cyclic nor time-critical, is useful for factory management and supervision (e.g., Standard Ethernet). Going down toward the field level, there are buses with intermediate or short framework length employed as an interface between the control units at the cell level and the devices at the field level, which are endowed with cyclic and acyclic functions and used for control, parameterization, initialization, and alarms. All the fieldbus types belong to this category, which can be considered as the keystone of the modern manufacturing and process industry. Depending on their cycle time and protocol features, fieldbuses can be used to network fast-operating devices with time-critical performance, such as PDSs.

A significant issue regarding the choice of a fieldbus for the integration of an industrial process arises from the variety of protocols, which stimulate the request of criteria for performance evaluation and comparison. The serial transmission is a common property of almost all the protocols, but the raw value of the bit rate, which is defined in bits per second, is hardly sufficient to suggest the transmission time. First of all, the bit overhead required by the protocol has to be accounted for; then, some application-related elements affect the transmission efficiency, such as the number of connected nodes and the demanded applicative functions. When the bus is asked to network actuators, sensors, and other devices with relatively slow dynamics, it is likely that the transmission speed is fast enough to make the connected devices a "transparent" module (process industry). On the other hand, when the number of networked devices grows or their dynamics become faster (PDSs for motion

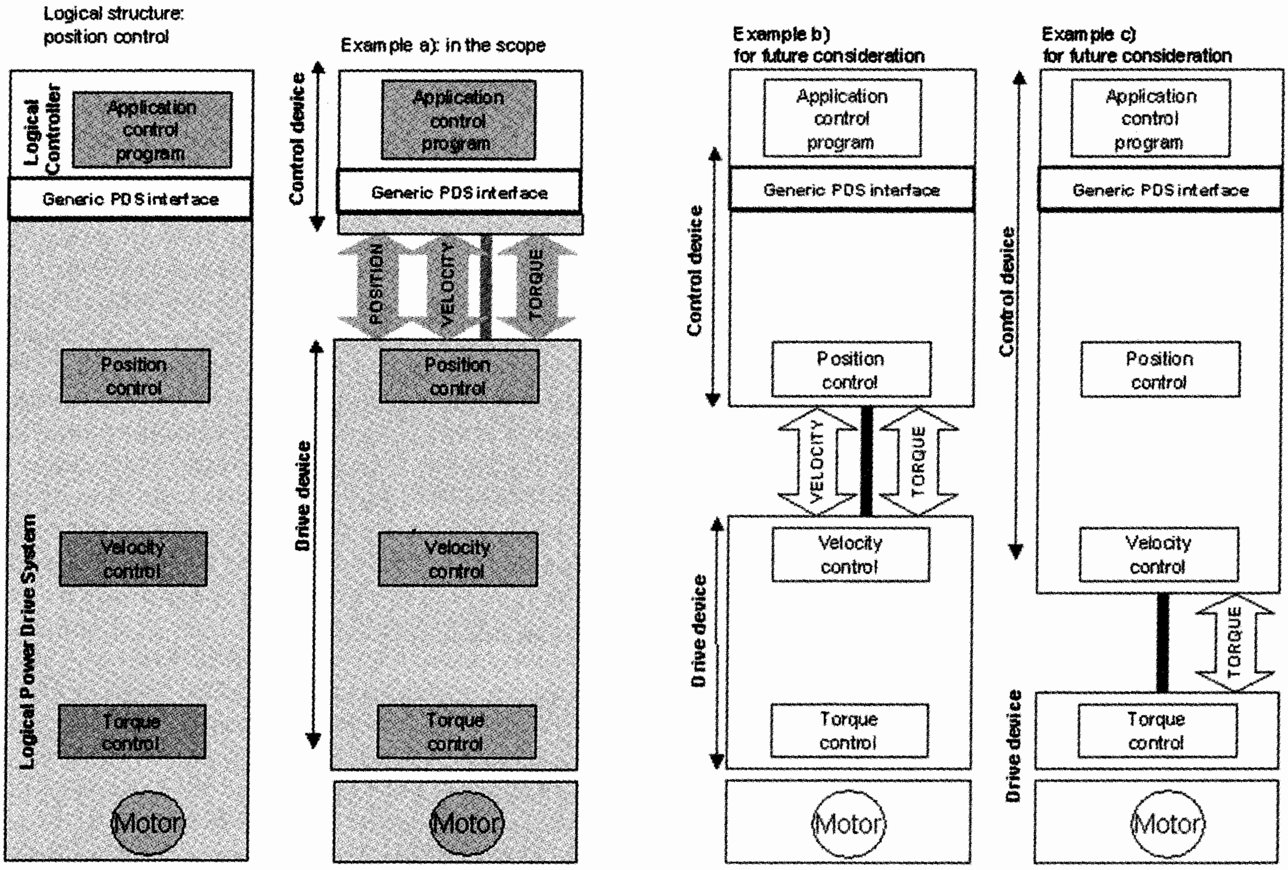


Fig. 3. Logical structure and examples of position control architecture for a single drive, according to IEC 61 800-7.

control systems and machine tools), protocol characteristics regarding determinism, synchronism, and repeatability of the control cycle must be considered, in addition to the coordination and supervision capabilities [6]. Finally, the assessment of an evaluation procedure of the fieldbus performance must reckon with the application.

For the sake of discussion, basic terms relevant to a networked process are explained as follows.

1) *Real-Time operation* of a fieldbus can be defined as the condition for the data exchanged on the bus to be directly used to accomplish the control tasks of the networked process. There is not a given speed specification ensuring that a fieldbus operates in real time, but it is necessary that the bus conveys the data in a time interval that is consistent with the dynamics of the process. Therefore, a fieldbus operating in real time for the process industry, where cycles of tens of milliseconds are common, may not be able to meet the real-time requirements for the control of a PDS in the manufacturing industry, where the cycles are faster. In a less strict meaning, real-time operation is ensured when the protocol can certify the maximum time spent to execute a fixed communication task that, in the manufacturing environment, is in the order of one millisecond or less.
2) *Determinism* is the property of a protocol to associate, at any instant, a comprehensive knowledge of the network status, which is to know which node has the right to access the bus and, more generally, the type of message the node conveys on the bus.
3) *Synchronism* is the property of a fieldbus to have a clock common to all the process events. This implies the periodical broadcasting of clock data telegrams on the bus. Synchronism is important for the implementation of any discrete control algorithm.

III. STANDARD INDUSTRIAL INTERFACES FOR PDS

A. *Interfaces for PDS*

According to IEC 61 800-7, the drive device has different logical interfaces to the outside world, which are circled from 1 to 5 in Fig. 1.

I/O data are transferred through interface 1 on a regular or scheduled time base. They include set points, commands from the control device to the drive device, and status and monitor values from the drive device to the control device. In a typical application, this transfer occurs in a cyclic I/O data exchange. In some application modes, it is possible to transmit this I/O data in a synchronous mode, meaning that the cycles of the control device, the cycles of the network, and the cycle of the drive device stay in a fixed time relation.

Device parameters are transferred through interface 2 on an unscheduled time base. This acyclic communication is typically used as an engineering tool for technological set up and parameterization purposes, but the

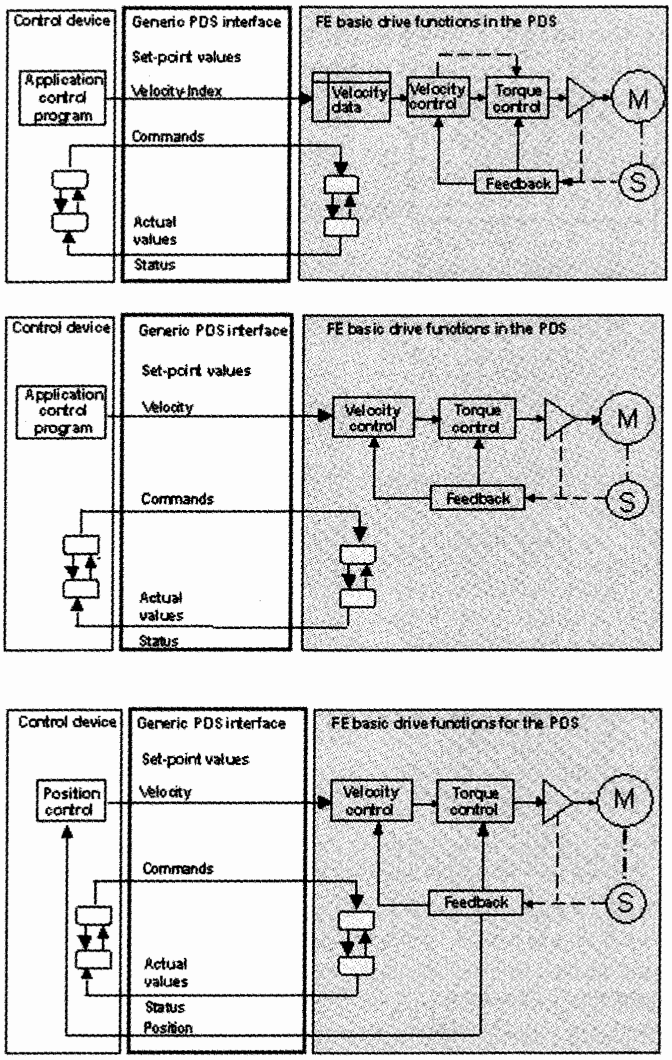


Fig. 4. Three interfaces for velocity control (*from top down*) 1. Preset velocity control through an index table. 2. Direct velocity control. 3. Velocity control with position feedback.

control device is also able and allowed to access the drive device over the network through this interface for diagnostic and monitoring purposes.

Peer to peer device communication trough interface 3 allows one drive device to exchange I/O data with other drive devices. This feature is exclusive to some of the communication profiles.

Local interface #4 is used as an access for engineering tools via HMI to the drive device. HMI may be conveyed by means of the fieldbus communication services supporting the PDS interface.

Process Interface #5 depends on the technology of the driven equipment, e.g., the motor or the actuator.

Interfaces #3, #4, and #5 are essentially manufacturer specific and are not covered by the specifications of IEC 61 800-7.

Most drive profiles describe different modes of operation, otherwise termed *operating modes*. Notice that the term "operating mode" is defined in IEC [7] as follows: "Characterization of the way and the extent to which the human operator intervenes in the control equipment." Therefore, for a drive device, it is more appropriate to use the term *application mode*, which is defined as the type of application that can be requested from a drive device. The different application modes reflect the control-loop arrangement for torque control, velocity control, position control, or other applications like homing. Depending on the application mode, the generic PDS interface uses different commands, set points, actual, and status values.

B. Architectures for PDS

Logical structure and examples of architectures for single PDSs have been provided for by IEC 61 800-7, as shown in Fig. 3, where the position control application mode has been reported . Here, the control loops are nested according to the following sequence: position, velocity, and torque. In the proposed examples, the interface is based on the assumption that all the control loops are implemented in the drive device. Other solutions are foreseeable, with a more relevant role of the controller that can implement some control loops, but such a centralized solution does not seem very popular, and at the moment, it is provided only for future consideration by IEC 61 800-7. Simpler control architectures, including only velocity (see Fig. 4) or torque control, are also available using a similar architecture.

C. Drive Interface Profiles

A number of industrial profiles, which are well established on the market and dedicated to PDS, are now shortly reviewed and compared, since only from their amalgamation can we hopefully derive some sort of convergence toward a generic interface.

1) CANopen: The organization CAN in Automation (CiA) has defined various versions of profiles for Drives and Motion Control. The CANopen profile differentiates between Process Data Objects (PDOs) and Service Data Objects (SDOs). PDOs are transmitted in a fast and efficient way but without confirmation. SDOs are transmitted with confirmed services and are used to access parameters. The different Qualities of Service (QoS) for PDOs are synchronous and asynchronous transmissions with event- or time-driven triggering modes. In the latest version [8], the CANopen profile is very similar to the DRIVECOM profile. Several sections and drawings are even identical. CANopen was developed to be used over a CAN network, but in recent publications, it is also proposed for different networks (e.g., for Powerlink [9]).

2) CIP: The Open DeviceNet Vendor Association and ControlNet International defined a common application layer protocol, which is termed Control and Information Protocol (CIP). Of all the device profiles defined in Chapter 6 of the document [10], only the subset that builds a "Hierarchy of Motor Control Devices" is of interest here. The CIP Protocol is designed to be downwards compatible from the more complex servo drives to the simpler devices. The CIP protocol also can be used over Ethernet/IP. In CIP, the periodic I/O data is mapped to the assembly I/O object. All the parameters of all the objects can be accessed over an acyclic communication.

3) DRIVECOM: The DRIVECOM profile is used by Interbus communication networks. The document describing the profile of DRIVECOM is the oldest one dealing with drive interface [11]. An I/O data channel for the cyclic data and a communication channel for device parameter data are defined.

4) Profidrive: The Profibus Organization has defined several versions of profiles for variable-speed drives. The last version 3.1 [12] is called *Profi*drive and includes new features grouped in classes of applications: Application Class 1: Standard Drive; Application Class 2: Standard drive with distributed technology controller; Application Class 3: Positioning drive, single axis with distributed position control and interpolation; Application Class 4: Positioning with central interpolation and position control; Application Class 5: Positioning with central interpolation and distributed closed-loop position control; Application Class 6: Motion control for clocked processes or distributed angular synchronism.

The I/O data for the different application modes are defined together with the standard telegrams. Cyclic communications can run free or can be synchronized with a special telegram (i.e., the global control telegram). Peer-to-peer communication with broadcast messages is supported. A special communication channel to read and write parameter data is included. The definition of the parameters, however, is considered as technology dependent and, therefore, is not defined in the *Profi*drive profile. Actually, the work has started to define *Profi*drive also for the Eternet-based *Profi*NET protocol.

5) SERCOS: In 1995, the SERCOS Interface was standardized by IEC [13]. The actual version 2 was released in 1999. At the present time, a version 3 based on Ethernet technology is under development.

The SERCOS Specification is very much oriented to the communication technology used to synchronize the drives. The so-called Interface Compliance Classes (ICCs) are used to specify the different operating modes. This has been chosen as the only way of giving a structure to the variety of parameters defined in the profile. The SERCOS interface contains hundreds of identification numbers, which are gathered in Annex A of [13]. The profile's parameters provide access to all the signals in the drive system, including controller-internal signals. The SERCOS Profile is split in three Classes: A, B, and C. Class A supports only the exchange of general parameters, while the parameters relevant to an application mode are grouped in Class B. Class C contains extensions in the functionalities.

The SERCOS Profile has a cyclic data exchange. A Master Synchronization Telegram defines the cycle limit. The Master Data Telegram allows the master to send data to all the drives (broadcast). The Drive Telegram (AT) is used by a drive to send data to the master. The noncyclic data exchange takes a time slot in the service channel to modify the drive parameters. A special Master Data Telegram and AT are in charge of these tasks. Within one of the ICCs, file uploading and downloading are also supported.

D. Profiles Comparison

Fig. 5 reports an example of differences and common features of the various profiles. A significant issue of comparison is the content of the command word (or byte) inside the I/O data that the control device utilizes to control the drive device. The different assignation of the bits used by the various profiles is specified in Fig. 5. The meaning is almost identical, except for SERCOS, where the numbering is upside down. The mechanism to reset a fault condition in the drive is the same in all the profiles. The switch-on and quick-stop features of the drive are very similar. The most important difference is with the CIP profile, where only one byte of command is defined, and most bits are optional, thus originating a quite simple solution consisting in just switching the drive on and off. The other drive models include more or less complex state machines, which depend on the application mode.

IV. REAL-TIME ETHERNET FOR PDS

Ethernet was born to support the general-purpose communication tasks in the office environment, and since then, it has been expanded to a universal domain thanks to TCP/IP interface, which stands as the basis of the World Wide Web. As such, it looks rather distant from the communication needs of the industrial processes.

Nonetheless, Ethernet networks have been recently developed for the industry [14]. Their usage in the cell and field levels is particularly appealing for many reasons: consistency with the communication systems installed at the higher levels, short cycle-times due to the high transmission speed, large availability of chips implementing the Ethernet protocol, and use of

	DRIVECOM	SERCOS	CANopen	PROFIdrive	CIP
Bit 0	Switch-on	Master Handshake	Switch on	ON/OFF	Run1 (Fwd)
Bit 1	Disable voltage	Read/Write	Enable voltage	Coast Stop	Run2 (Rev) 3)
Bit 2	Quick stop	Progress	Quick stop	Quick Stop	Fault Reset
Bit 3	Enable operation	Datablock	Enable operation	Enable Operation	Reserved
Bit 4	1)	Datablock	1)	1)	Reserved
Bit 5	1)	Datablock	1)	1)	NetCtrl 3)
Bit 6	1)	Real-time	1)	1)	NetRef 3)
Bit 7	Reset-malfunction	Real-time	Fault reset	Fault Acknowledge	NetProc 3)
Bit 8	Reserved	Operation mode 3	HALT	Jog 1	—
Bit 9	Reserved	Operation mode 2	Reserved	Jog 2	—
Bit 10	Reserved	Synchronisation	Reserved	Control by PLC	—
Bit 11	2)	Operation mode1	2)	2)	—
Bit 12	2)	Reserved	2)	2)	—
Bit 13	2)	Halt/restart Drive	2)	2)	—
Bit 14	2)	Enable Drive	2)	2)	—
Bit 15	2)	Drive ON/OFF	2)	2)	—

1) Meaning depends on mode of operation; 2) Manufacturer or device specific; 3) Only available in extended mode

Fig. 5. Bit assignation in the command word.

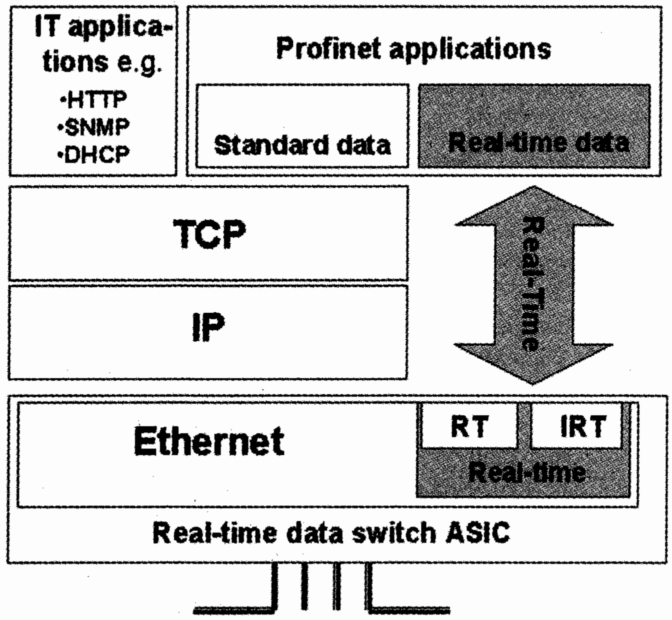


Fig. 6. Profinet layer distribution.

IT utilities for services and diagnostics. At the moment, a pure real-time Ethernet network is still not available, even if some manufacturers have incorporated Ethernet in their fieldbuses, thus creating a new set of Ethernet-based solutions. Examples are Profinet (used by Interbus and Profibus), Ethernet-IP (CIP based on DeviceNET and ControlNET), Fieldbus Foundation High Speed Ethernet (HSE), and Ethernet Powerlink (based on CANopen).

In general, Ethernet-based solutions have been delivered for general-purpose applications in the manufacturing and process industry. Nevertheless, such solutions as those provided by Profinet deserve a mention, where a special layer is arranged for real-time applications, thus constituting an architecture that is able to encompass all process levels, from the management to PDS (see Fig. 6). In particular, the standard data channel is dedicated to device parameterization and configuration as well as to the reading of diagnostics data and the negotiation of the communication channel for user data. The two ad-hoc channels Real Time (RT) and Isochronous Real Time (IRT) are intended for real-time operation; RT is available for fast cyclic transfer of user data and event-controlled signals/alarms, while IRT is available for high-performance transfer of user data with a jitter time lower than 1 μs.

V. CONCLUSIONS

Modern industrial architectures can fully exploit the performance made available by fieldbuses and communication devices for controlling and supervising industrial plants based on PDS. In fact, fieldbuses take advantage from the increasing bit rate and the dedicated communication services to exhibit distinct features committed to real-time implementation, even in the presence of demanding control specifications of PDS such as those required by machine tool and motion control applications. The crucial problem of standardization is still a significant obstacle against interoperability of the devices and freedom in the technical choices from the customer. It could find a promising answer in the regulation activities aimed at the definition of a generic interface for PDS, thus guiding manufacturers and system designers toward a standard process architecture. Another opportunity could be given by the extension of the Ethernet protocol to the field level since it makes possible a true integration within a factory from the supervisory level to the PDS one.

REFERENCES

[1] International Electrotechnical Committee, Digital Data Communications for Measurement and Control—Fieldbus for Use in Industrial Control Systems, 3rd ed., 2003.
[2] European Committee for Electrotechnical Standardization, General Purpose Field Communication System, 2002.

[3] M. Felser and T. Sauter, "The fieldbus war: history or short break between battles?," in *Proc. IEEE Int. Workshop Factory Commun. Syst.*, 2002, pp. 73–80.

[4] Adjustable Speed Electrical Power Drive Systems, Part 7: Generic Interface and Use of Profiles for Power Drive Systems, International Electrotechnical Commission, Jun. 2004.

[5] E. Bassi, F. Benzi, G. Buja, and L. Lusetti, "Communication protocols for electrical drives," in *Proc. IECON*, 1995, pp. 706–711.

[6] G. C. Buttazzo, *Hard Real-Time Computing Systems*. Norwell, MA: Kluwer, 1997.

[7] International Electrotechnical Committee, Control Technology, Industrial Process Measurement and Control, 3rd ed., 2002.

[8] CANopen Device Profile Drives and Motion Control, Erlangen, Jul. 2002.

[9] P. Duchemin, "Ready-to-go hardware/software subsystem for Ethernet Powerlink," *Embedded Contr. Eur. Mag.*, Feb. 2004.

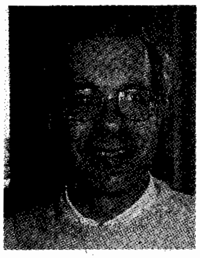
[10] Open DeviceNet Vendor Association and ControlNet International, CIP Common Specification, Jun. 2001.

[11] Profile Drive Engineering/Servo, Blomberg, Sep. 1994.

[12] Profibus Profile: Profidrive—Profile Drive Technology, Karlsruhe, Germany, Nov. 2002.

[13] Specification SERCOS Interface Standard IEC 61491, Electrical Equipment of Industrial Machines, Stuttgart, Germany, Nov. 2001.

[14] M. Bertoluzzo, G. Buja, and S. Vitturi, "Ethernet networks for factory automation," *IEEE Indust. Electron. Soc. Newslett.*, pp. 5–10, Dec. 2003.



Francesco Benzi (M'88) received the laurea degree in electronic engineering with honors from the University of Pavia, Pavia, Italy, in 1978.

He joined the WEMPEC Lab at the University of Wisconsin, Madison, in 1984–1985 as a Visiting Researcher and the Department of Electrical Engineering of the University of Pavia as an Assistant Researcher, where he is currently Full Professor of Electrical Drives. His research interests lie in the application of adjustable speed drives and actuators in industrial automation and robotics. He is also particularly involved in issues concerned with process integration, including communication protocols for industrial and domestic environments and building automation.

Prof. Benzi is a member of the Italian Electrical and Electronics Association (AEI), serving as a Chairman of the Committee for Automation.

Giuseppe S. Buja (M'75–SM'84–F'95) received the laurea degree in electronic engineering with honors from the University of Padova, Padova, Italy, in 1970.

Upon graduation, he joined the Engineering Faculty of the University of Padova. Since 1986, he has been a Full Professor of Power Electronics, first at the University of Trieste, and then at the University of Padova. He has carried out research in the fields of electric energy static converters, electric drives, motion control systems, and fieldbuses and has authored or coauthored more than 150 papers published in refereed journals and international conference proceedings. He started up the Laboratory of Electric Drives at the University of Trieste, Trieste, Italy, and the Laboratory of Industrial Automation at the University of Padova, the latter of which he is currently the Head. He has directed several research projects granted by the university and by private companies.

Prof. Buja has served the IEEE in several capacities, including General Chairman of the 20th Annual Conference of the IEEE Industrial Electronics Society (IEEE IECON-1994) and the University of Padova as the Coordinator of the Ph.D. course in electrical engineering. He was a cofounder of the International Symposium on Diagnostics for Electric Machines, Power Electronics, and Drives (SDEMPED). Currently, he is a Senior Member of the Administrative Committee of the IEEE Industrial Electronics Society. His biography has been included in the last four editions of *Who's Who in the World*.

Max Felser (M'81) received the B.Sc. degree in electrical engineering from the Ecole d'ingénieurs et d'architectes de Fribourg, Fribourg, Switzerland, in 1980 and the M.Sc. in electrical engineering from the Swiss Federal Institute of Technology Zurich, Zurich, Switzerland, in 1983.

He joined the Berne University of Applied Sciences as a Professor in 1991 after some development engineering at Ascom AG (data communication systems) and Saia-Burgess AG (PLC development). He runs the fieldbus laboratory at the School of Engineering and Information Technology, Burgdorf, Switzerland. His research interest includes fieldbuses and industrial, real-time networks.