2. REAL-TIME CONTROL SYSTEM AND REAL-TIME NETWORKS

2.1 Real-Time and Control

Computer based digital controllers typically have the ability to monitor a number of discrete and analog inputs, perform complex control algorithms, and drive several outputs, all at defined speeds, often very high. It is necessary that all the above operations and calculations take place at the correct moment in time. This imposes the following requirements on the hardware and software of computer-based control systems:

- sufficient processing power,
- sufficient high-speed input/output interfaces (peripheral hardware),
- operating systems fulfilling more or less hard real time requirements and handling error conditions in a predefined way.

In many cases the processing power of digital controllers is not the only priority. Some applications are classified **as safety-critical**, in the sense that computer failure may result in incorrect or disastrous behaviour of the process (for instance a threat to life, or to the Earth's environment). In this case, the list of the requirements must be extended to include control system reliability (hardware endurability and operating system stability).

In general, a computer-based digital controller must detect real-world events and respond to them by taking appropriate actions. This feature of the controller is referred to as **real-time operation**: **the operating mode of a computer system processing asynchronous inputs and producing outputs in a deterministic and bounded amount of time**. The arrival times of the data can be randomly distributed or can be determined depending on the different applications. A real-time system is one in which the correctness of the system operation depends not only on the logical results of computation, but also on the time at which the results are generated. Hardware over-capacity may satisfy the execution duration of all the applications but may not satisfy the predictability. Only a proper distribution of computer resources together with sufficient computational power allows one to build a predictable Real Time Control System (RTCS). Real-time systems are usually classified into hard real-time systems, soft real-time systems and firm real-time systems.

- A **hard real-time system** is a system, where missing the response deadline can be catastrophic.

- A **soft real-time system** is a system where deadlines are important but where the system operates properly if the deadlines are occasionally missed (system performance is degraded, but not destroyed).
- The term **firm** relates to a real-time system where late data processing results are worthless and some probability of violating a response time is tolerable.

A typical control system consists of: a controlled plant, sensors, actuators, input-output interfaces and a controller including a Real-Time Operating System (RTOS). There are three basic concurrent processes of RTOS: performing measurements, data processing and running control algorithms.

Fig.2.1 illustrates the principle of co-operation between hardware and software layers. The RTOS operates on device drivers. A **driver** is a software "image" of the I/O hardware. Access to the I/O devices is possible by device drivers. The RTOS synchronises the information flow between the hardware level and control applications.

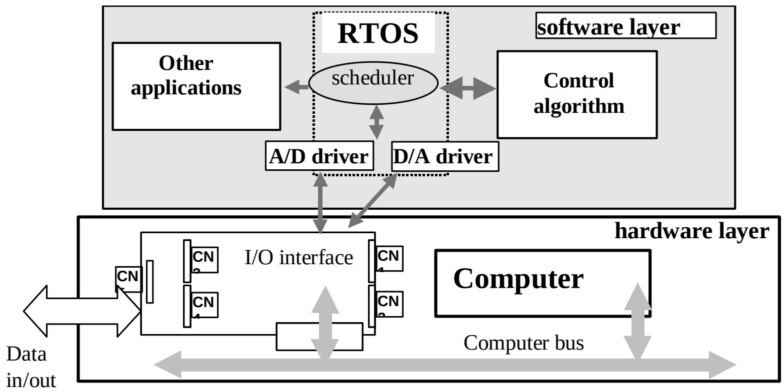

Fig.2.1. Hardware and software layers of the computer controlled system

2.2 Real-Time Distributed Control Systems

Feedback control systems wherein the control loops are closed through a communication network are referred to as distributed control systems. They are distributed in the sense that their sensors, actuators and controllers communicate via a shared data transmission network. Communication networks were introduced in control in the 1970s. They can be grouped into fieldbuses (e.g. CAN, Profibus, Modbus) and general purpose networks (e.g. IEEE standard LANs). Each type of network has its own protocol that is designed for a specific range of applications. Fieldbuses are intended for real-time applications, but in some cases general-purpose networks may have been used for control [3]. The behaviour of a networked control system depends on the performance parameters of the underlying network, which include transmission rate and access method to the network transmission medium.

Using a distributed control architecture has many advantages over a point-to-point design including low cost of installation, ease of maintenance and flexibility [4]. The introduction of distributed architecture can improve the reliability and efficiency of the control application. For these reasons distributed control architecture is widely used in industrial applications.

The introduction of a network into a feedback loop in some cases violates conventional control theory assumptions such as non-delayed sensing and actuation. These time delays mainly come from the time sharing of the communication network. **Lack of access to the communication network is an important constraint compared to lack of computer power or time errors of the RTOS.** Time delays can degrade the performance of the control system designed without considering network delays and can even destabilise the system.

Communication protocols that minimise data delay and make the system time invariant are widely introduced. There are other approaches for accommodating all network effects in control system design. One way is to treat a distributed control system as a real-time time system, where data transfer from/to the plant and control of communication lines are real-time tasks. For such a task one can use all the standard methods for real-time system design: define priorities for task scheduling methods and algorithms and time constraints etc. Finally, a distributed RTOS system is a design which enables the flow of information in a limited amount of time.

Another approach is to treat the network properties as given conditions, and to design a control algorithm that takes network effects into account. The effect of this on performance degradation of the distributed control system is reduced by a proper selection and tuning of the control algorithm.

Current communication systems for automation implement different protocols. This is a substantial disadvantage, leading to the need to use vendor-specific hardware and software components, which increase installation and maintenance costs. Moreover, presently used fieldbus technologies make vertical communication across all levels of the automation systems difficult. Gateways need to be used to establish connections between different kinds of fieldbus systems used in the lower levels, and Ethernet used in the upper levels.

Ethernet provides unified data formats and reduces the complexity of installation and maintenance, which, together with the substantial increase in transmission rates and communication reliability over the last few years, results in its popularity in the area of industrial communications.

Ethernet, as defined in IEEE 802.3, is non-deterministic and, thus, is unsuitable for hard real-time applications. The media access control protocol, CSMA/CD with its backoff algorithm, prevents the network from supporting hard real-time communication due to its random delays and potential transmission failures. In real-time systems, delays and irregularities in data transmission can very severely affect the system operation. Therefore, various techniques and communication protocol modifications are employed in order to eliminate or minimise these unwanted effects.

To employ Ethernet in an industrial environment, its deterministic operation must first be assured. This can be accomplished in several ways. Coexistence of real-time and non-real time traffic on the same network infrastructure remains the main problem. This conflict can be resolved in several ways by:

- embedding a fieldbus or application protocol on TCP/IP – the fieldbus protocol is tunneled over Ethernet, and full openness for "office" traffic is maintained,
- using a special Data Link layer for real-time devices – special protocol is used on the second OSI Layer, implemented in every device. The real-time cycle is divided into slots, one of which is opened for regular TCP/IP traffic, but the bandwidth available is heavily limited (i.e., minimized),
- using application protocol on TCP/IP, direct MAC addressing with prioritization for real-time, and hardware switching for fast real-time,
- maintaining real-time on TCP/IP is achieved by prioritized messaging and time synchronization – the synchronized devices assign higher priority and timestamp real-time messages,

All these specific techniques allow a considerable improvement in terms of determinism. The desire to incorporate a real-time element into this popular single-network solution has led to the development of different real-time Industrial Ethernet solutions, called Real-time Ethernet, such as PROFINET, EtherCAT, Ethernet/IP and many more. The conditions for the industrial use of Ethernet are described by international standard IEC 61 784-2 *Real Time Ethernet* (See Fig.2.2). IEC stands for *International Electrotechnical Commission*.

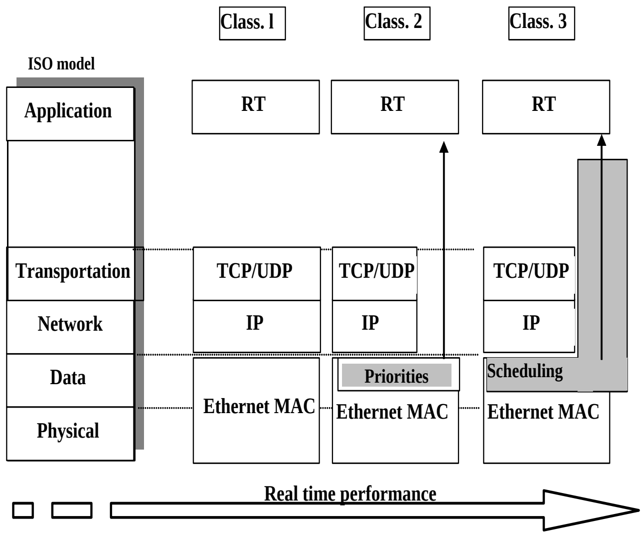

Fig.2.2 Classification of industrial Ethernet (IEC 61 784-2)

Class 1 describes the use of standard Ethernet TCP/IP as it is. In this case the different real time protocols and the best-effort protocols, like HTTP, SNMP, FTP etc., uses the services of the TCP/IP protocol suite. This includes examples such as CIP Sync (Ethernet/IP, ModBus/TCP). The class 1 has the largest conformity to the Ethernet TCP/IP standard and can thereby use standard hardware and software components.
Class 2 introduces optimizations, whereby the realtime data bypasses the TCP/IP stack and thus considerably reduces the dwell time in the node and increases the achievable packet rate. The dwell time of the node is one of the substantial influence factors for the realtime performance and has for embedded devices typical values of 1*ms*. In Classes 1 and 2, the priority support described in IEEE 802.1Q can also be used depending on the approach. In Class 3 the scheduling on the MAC level is again modified through the introduction of a TDMA method. Class 3 can be used in applications that require maximum latency in the range 1*ms* and a maximum jitter of < 1micro*s*. In this class there are strong restrictions for the use of standard components or the necessity for special components, like switches. Generally conformance with the Ethernet standard decreases when ones increase the Class number, while the achievable realtime performance increases.

2.3 Computer Implementations of Real-Time Control Systems

Real-time control computers equipped with RTOSs are applied as:

Embedded Systems, when the computer becomes a component of a larger system:

- microcontrollers based on dedicated processors,
- miniature WEB servers hosting I/O circuits,
- controllers based on software-configurable FPGA technology, e.g. XILINX chips.

Industrial Control Systems, when the computer creates a self-contained control configuration:

- Programmable Logic Controllers (PLCs), soft-PLCs,
- open-standard industrial computers (e.g. VME or PC104 bus industrial computers holding appropriate I/O interfaces),
- distributed control systems, organized according to some hierarchy, industrial computers (IPC) supervising a more or less intelligent input/output boards.

Most embedded systems are dedicated. This means that their functionality is tied to hardware and software for ever. They are widely used in everyday electronic equipment, e.g. transaction systems or multimedia equipment. They fulfil the essential requirement of a real-time application such as a deterministic response time and operating system stability. However these solutions are not flexible.

In industry, PLCs are still a standard control solution for both continuous and sequential processes. The VME industrial computers offer hardware and software applications for harsh conditions. They are flexible i.e. the computer is used as a development and implementation platform. Their main drawback is the price. The guaranteed cycle times of RTOS applied to different hardware platforms is shown in Table 2.1.

Table 2.1. Industrial RT solutions

Hardware platform	**Minimal cycle**)**	**Examples of RTOS**
Microcontrollers	10- 50 μs	-
DSP controllers	1 μs	dSPACE
PLC	1-20 ms	dedicated
FPGA controllers	10 ns	-
FPGA controllers	10 μs	EDA
VME industrial controllers	100 μs	QNX, VxVorks
IPC*)	100 μs	Extended Windows, RTLinux

*) industrial PC, **) time interval between input reading and output signal generation

2.4 RTOS and Control System Performance

Currently, there are three different approaches to the problem of designing (or selecting) the RTOS for a computer-based control system:

- Real-time system designer approach,
- Control system designer approach,
- Mixed approach.

Real- time system designers usually try to decouple real-time aspects and dynamics of the control system. They develop controllers that guarantee all task deadlines under worst case scenarios such as heavy load and external interruptions or breakdowns. The design of safety-critical controllers is based on this approach. Plant can be suitably controlled, but at the cost of poor computer resource utilisation. Timing flexibility in real-time systems can be provided by introducing concepts of graceful degradation, of control system performance, or imprecise computation methods. This last approach relies on making available the results of poorer quality calculations in situations when precise results can not be delivered in time.

Control system designers use mathematical models and simulation tools to analyze the influence of real-time control parameters on stability and performance of the controlled system. The main parameters of real-time control are: sampling period and jitter in sampling periods. The contribution of sampling periods and timing delays to the instability of control systems is well known.

A great number of control processes can tolerate some timing errors of the computer operating system without serious failure and return to normal operation thanks to feedback correction commands. This robustness against RTOS timing errors is especially strong if a large safety margin in sampling rate has been chosen; i.e. the time interval between input reading and output signal generation is much shorter than the minimum plant sampling rate necessary for proper operation. If the sampling rate is bound by the slow speed of the `control computer and is close to the minimum plant sampling rate, the control system becomes **time critical** (Fig.2.3). For such a system, the performance of the RTOS and of the communication network is essential for the correct operation of the control system.

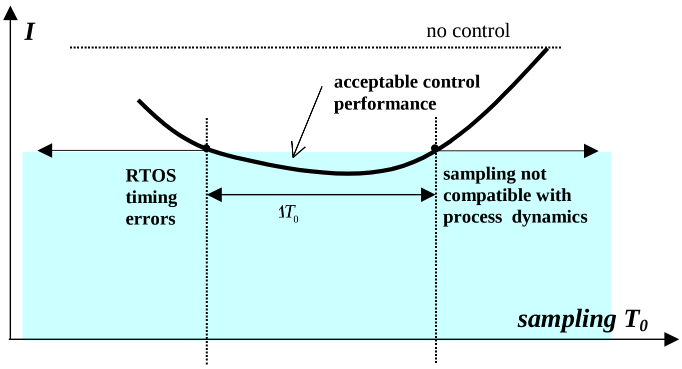

Fig.2.3. RTOS and process dynamic relations. For small ΔT_0 the control system becomes time critical. *I* – value of an objective function

Author: W. Grega