Chapter 1
Introduction

María Guinaldo, Francisco R. Rubio, Sebastián Dormido, Pablo Millán, Carlos Vivas and Luis Orihuela

1.1 Historical Perspective: From Digital Control to Networked Control Systems

The idea of using digital computers for control purposes started to emerge in the 1950s. In those times, however, computers were slow and unreliable, very limited in memory and computation capabilities, and were generally restricted for use as data loggers or performing computations for managing information. As reliability improved, computers were gradually integrated, first in supervisory control operations, then as controllers themselves. In 1962, a radical breakthrough was introduced by Imperial Chemical Industries (ICI) Ltd. in the UK, installing a Ferranti Argus Computer at Burnaze Works to measure 224 variables and manipulate 129 valves

M. Guinaldo (✉) · S. Dormido
Dpto. de Informática y Automática, Escuela Técnica
Superior de Informática, UNED, Madrid, Spain
e-mail: mguinaldo@dia.uned.es

S. Dormido
e-mail: sdormido@dia.uned.es

F.R. Rubio · C. Vivas
Dpto. de Ingeniería de Sistemas y Automática,
Escuela Superior de Ingenieros, Universidad de Sevilla, Seville, Spain
e-mail: rubio@us.es

C. Vivas
e-mail: vivas@us.es

P. Millán · L. Orihuela
Dpto. de Matemáticas e Ingeniería, Escuela Técnica
Superior de Ingeniería, Universidad Loyola Andalucía, Seville, Spain
e-mail: pmillan@uloyola.es

L. Orihuela
e-mail: dorihuela@uloyola.es

M. Guinaldo Losada et al. (eds.), *Asynchronous Control for Networked Systems*,
DOI 10.1007/978-3-319-21299-9_1

directly. This is considered as the first time a computer was directly interfaced to and controlled a particular system, and the beginning of the era of direct digital control (DDC).

The growth of DDC was explosive since then, helped by lower costs, increasing performance, and reliability of digital technology. While the first implementations of DDC were restricted to dedicated links between controller and actuators/sensors, user needs and technological advances in communications paved way for the introduction of digital multiplexing in serial communication in the early 1970s and the first decentralized computer control systems (DCCS) in the middle and late 1970s. At this period of time, research interests shifted somehow to the new paradigm, as it is evident from the fact that IEEE and IEE conferences on distributed processing and distributed computer control systems were started.

Decentralized control systems were soon thereafter applied in integrated manufacturing and industrial applications in general. The first works treating the use of decentralized control in machinery also appeared at this time, see [63]. Excellent work dealing with some of the fundamentals of decentralized control systems was produced in the early days of decentralized processing. For example, elements for a global clock as a fundamental base for decentralized applications was put forward by [136], together with the use of datagrams for real-time applications instead of conventional positive acknowledgment and retransmission protocols. In an early work on decentralized processing by [118], the partitioning and allocation phases were also discussed. In [67, 117], the levels and degrees of decentralization were clarified. These ideas gave rise to a whole new branch of control theory whose most prominent applications came in the form of the field bus technology (e.g., FIP and PROFIBUS) and automotive buses (e.g., CAN), successfully employed for decades in the process and automation industry.

The astonishing growth of communication technologies over the past decades—reflected by available protocols, coding, and modulation algorithms and the switching/routing technologies for packet-based networks—rapidly attracted the interest of the control community. The use of a multipurpose shared network to connect decentralized control elements promised improvements in terms of more flexible architectures, reduced installation and maintenance costs, and higher reliability than traditional bus-based communication technologies. The problems associated to such a change of paradigm also proved to be challenging [188].

Networked control systems (NCSs) are decentralized systems in which the communication of the different elements of the control loop (sensors, actuators, and controllers) employs a shared digital communication network. NCSs is thus an interdisciplinary field, lying at the intersection of control and communication theories. Favored by the large number of applications and difficulties involved, in the last few years NCS has become a common issue for many control research groups all around the world (see the expert panel report on *Future Directions in Control, Dynamics, and Systems*[1]). Indeed, at least two of the technical areas of the International Federation of Automatic Control (IFAC) are devoted to this field, a new IEEE Transactions on

[1] http://www.cds.caltech.edu/murray/cdspanel.

the topic (TCNS) was launched in 2014, and there also exists an increasing number of specialized conferences and workshops, such as the IFAC Workshop on Distributed Estimation and Control in Networked Systems (NecSys) or the SICE International Symposium on Control Systems.

1.2 Overview of Networked Control Systems and Asynchronous Systems

1.2.1 Emergence and Advantages of Networked Control Systems

Typically, a control system is composed of the following elements: system or plant to be controlled; sensors measuring plant outputs, and transmitting them; automatic controllers receiving plant outputs and making decisions on the control signals to be applied to the plant; and actuators receiving the inputs sent by the controller and applying these inputs to the plant. Point-to-point communication links between the different devices make it possible to implicitly consider the *perfect communication channel approach*: absence of transmission delays, information integrity and unlimited bandwidth (Fig. 1.1).

Needless to mention, the feature that distinguishes an NCS from a classical control system is the presence of a communication network affecting inside the loop (Fig. 1.2). The perfect communication channel assumption does not hold when a network mediates the connection among the different elements, at least generally speaking. Even when dedicated, standard communication networks are usually designed to preserve data integrity and do not suit the stringent real-time requirements of closed-loop control. These problems become particularly apparent when wireless or non-dedicated networks are used. A large number of systems may be using the communication channel concurrently sharing the available bandwidth.

Hence, the following questions arise: Why is it better to use this type of technology for control purposes? In which situations are these solutions more suitable?

On the one hand, there are a number of generic advantages when using digital communication networks. Namely,

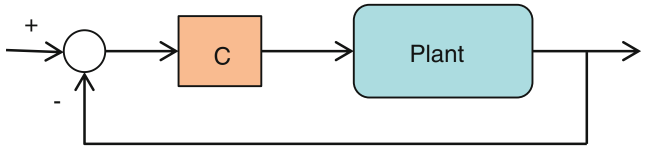


Fig. 1.1 Classic control scheme with the assumption of *perfect communication channel*

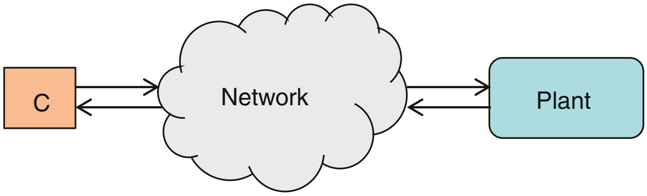


Fig. 1.2 Networked control scheme

- **Low cost** Using a point-to-point communication in large-scale systems or geographically distributed plants is generally a costly and impractical solution. Wireless or even wired networks, however, reduce the connections and the wire length. Concomitantly, the deployment and maintenance costs are shortened.
- **Reliability** In addition to the acknowledgment-retransmission mechanism of conventional communication protocols, a meshed network topology intrinsically improves reliability as dynamic routing allows to find alternative routes in the case that broken links are present. Additionally, fault detection algorithms can be easily implemented.
- **Maintenance** The reduction of wiring complexity facilitates the diagnosis and maintenance of the system.
- **Flexibility** Network structured systems offer flexible architectures, making easier the reconfiguration of the system parts and allowing a simpler addition of new devices.
- **Accessibility** Traditional centralized point-to-point control systems are no longer suitable to meet new requirements, such as modularity, control decentralization, or integrated diagnostics.

On the other hand, in a large number of practical situations the application or process advises engineers to use communication networks for control:

- **Space and weight limitation** Stringent limitations of this type need to be accomplished, for instance, in avionics (commercial aircrafts, unmanned aerial vehicles) or embedded systems in the automotive industry.
- **Coverage of considerable distances** chemical plants, large-scale factories, and automation systems.
- **Control applications where wiring is not possible** fleet of autonomous vehicles, safe driving control systems involving inter-vehicle communications, teleoperated systems, etc.

1.2.2 Communication Drawbacks

Communication through a shared network is imperfect and may be affected by some of the following problems (see Fig. 1.3):

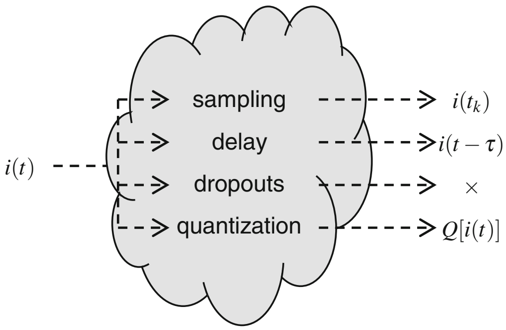


Fig. 1.3 The various problems affecting information $i(t)$ transmitted through a network

- **Sampling** In most digital networks, data are transmitted in atomic units called *packets*. These packets are sent at a finite rate, therefore continuous models must be discretized with an adequate sampling time. Since the available bandwidth is limited, sampling appears as a problem of the channel. In some network protocols, such as WiFi or Ethernet, this sampling time is not constant, as it strongly depends on the network traffic and congestion. A correct choice of the sampling periods will help to maximize the available bandwidth in those cases.
- **Delay** The overall delay between sampling and decoding at the receiver can be highly variable because both the network access delays (i.e., the time it takes for a shared network to accept data) and the transmission delays (i.e., the time during which data are in transit inside the network) depend on highly variable network conditions such as congestion and channel quality. Consequently, packets traveling through a network are received belatedly. For example, it is certainly common to receive one packet before another released earlier. Some protocols, such as TCP/IP, implement mechanisms accounting for this, but at the cost of increasing the delay. Even so, the reordering might be useless in control applications.
- **Packet dropouts** Some packets may also be lost, mainly because of the capacity of the reception buffer. If an element is receiving packets at a higher rate than it can process them, the buffer could overflow at any instant. Even, errors in physical links may cause the loss of information, as the packet must be discarded. Though some protocols guarantee data integrity through retransmission mechanisms, this is often useless in real-time control as old data packets cannot be used for control purposes. Indeed, many networked control algorithm discard and treat as losses those packets received with excessive delays.
- **Quantization** A quantizer is a function that maps a real-valued function into a piecewise constant function taking on a finite set of values. This mapping typically introduce inaccuracies inversely proportional to the cardinality of the representation alphabet. One of the basic choices in quantization is the number of discrete quantization levels to use. The fundamental tradeoff in this choice is the resulting signal quality versus the amount of data needed to represent each sample.

1.2.3 Research Trends

In the late 1990s, researchers began to identify the key distinctive issues of NCSs, driving the main research topics of the next decade.

- **Delays and packet dropouts** The *control-induced* delay, that is, the delay caused by the control scheme adopted, was first studied in the 1970s, when digital controllers were introduced to replace analog controllers. It was noticed that this kind of delay may induce by itself system instability, as was shown in a simple example in [271]. Digital controller design taking into account the computational delay has also been extensively studied, generally as extensions or applications of results developed for *time-delay systems* (TDSs) [13].
  Another source of delay is however present in NCSs, and it is caused by the transmission of the information through the network to the different components of the system. This kind of *control-induced* delay is commonly known in the literature as *network-induced delay* [242]. Network-induced delays, because of their discrete and distributed nature, are quite different from the plant delays and computational delays that have been studied in the past. However, in some cases, it is possible to use tools for linear sampled-data systems for the analysis and design of certain classes of linear NCSs [193]. Some problems also admit dealing with networked-induced delays in a similar way as traditional TDSs. This is the approach in some recent studies of time-delayed-system analysis and design [40, 122, 155, 168, 222, 280], which though not specific to NCSs, provide results that are applicable to NCSs.
  From a historical perspective, first results in the topic of network-induced delays in control systems were developed for the assessment of systems performance and design of improved communication protocols [95, 244]. Network time delays have since then been tackled in a variety of forms. In general, there are two methods to handle the networked-induced delays. One method is to design control algorithms considering the delays, such as in [159, 270]; the other is to reduce the delays by sharing a common network resource. Recently, part of the research [121, 144] on NCSs has focused on how to schedule network resources to make the network-induced delays as small as possible. These research results have also shown that network scheduling plays a subordinate, but very important role in NCSs. Other approaches tackle the problem from a robust control perspective, guaranteeing stability and performance in spite of the presence of delays. In many of these designs, the so-called maximum allowable delay bound (MADB) is established. The MADB can be defined as the maximum allowable interval from the instant when the sensor nodes sense the data from a plant, to the instant when actuators apply to the plant the corresponding control actions. For guaranteeing an NCS being stable, the sampling periods must be less than the corresponding maximum allowable delay bounds (MADBs) [38, 83, 145, 173, 268, 275, 277].
  Another significant difference between NCSs and standard digital control is the possibility that data packets may be lost while in transit through the network. For

a given sampling frequency, implementing estimation methods in an NCS would reduce the network traffic increasing the effective bandwidth of the system [25].

- **Band-Limited Channels** Any communication network can only carry a finite amount of information per unit of time. In many applications, this limitation poses significant constraints on the operation of NCSs. First incursions in the topic came from well-established results of information theory. A significant research effort has been devoted to the problem of determining the minimum bit rate that is needed to stabilize a linear system through feedback [26, 65, 105]. Recently, some progress has also been made in solving the finite-capacity stabilization problem for nonlinear systems [150, 191], derivation of stability conditions based on anytime information [223], or the study of performance limitations of feedback over finite capacity memoryless channels [161], with Bode-like extension limits of performance.
- **Stability of NCS** Unlike regular control systems, in NCSs the synchronization between different sensors, actuators, and control units is not guaranteed. Furthermore, there is no guarantee for zero delay or even constant delay in sending information from sensors to the control units and control units to the actuators. In real-time systems, particularly control systems, delays or dropped packets may be catastrophic and may cause instability in the process. Moreover, the time-varying nature of delays in NCSs may induce instability for time-varying delays in a bounded set; even when the NCS with any constant delay taken from this set is asymptotically stable [264].

  Stability under such circumstances has been investigated by a number of researchers. First results were obtained from the application of classical tools, as in [125] where a frequency-domain stability criterion, based on the small gain theorem, is proposed to investigate the stability of SISO NCS plants. A different modeling approach is used in [189, 274], where a continuous-time description, with a zero-order-hold controller, is proposed. Other relevant results regarding stability of NCSs can be found in [45, 102, 146, 151, 253, 273, 282, 283].
- **Energy aware** In all fields of engineering, energy-efficiency is becoming very important due to economical and environmental concerns. In networked control systems—especially if battery-powered devices are employed over wireless—energy-saving is key to increasing the lifespan of the system and, indirectly, reducing costs. Moreover, in some applications, the network devices can be deployed over hazardous or unreachable locations, and replacing the batteries may be expensive or impractical. This motivates current interests in developing energy-aware NCS methodologies, in particular, on protocols to reduce the average media access rate, as it is well known that wireless devices consume most energy when the radio is on.
- **Wireless Sensor Networks** A technological factor that has definitely amplified the impact of NCSs, both in industry applications and interest from academia, has been the rapid developments of wireless technologies in the past decade. Recent achievements in miniaturization, such as MEMS- and nano-technologies, have enabled the development of low-power, reduced-cost wireless devices with the capacity of establishing meshed networks in the so-called wireless sensor networks

(WSN). It is widely believed that this type of pervasive networking technology will be transparent to the user, but at the same time will allow monitoring and automation to an unprecedented scale.

- **Distributed systems** The challenge to the field is to go from the traditional view of control systems as a single process with a single controller, to recognizing control systems as a heterogeneous collection of physical and information systems with intricate interconnections and interactions. In addition to inexpensive and pervasive computation, communication, and sensing—and the corresponding increased role of information-based systems—an important trend in control is the move from low-level control to higher levels of decision making.

New possibilities and challenges arise in this context, and issues as distributed estimation and control over WSN, energy-aware NCS control, or multi-agent control are hot topics nowadays. Particularly, distributed estimation has been devised as a potentially useful strategy since the early 1990s [87], though it has found a renewed interest in the past few years with the development of WSNs. Distribution estimation techniques has been developed under different levels of imperfect channel assumptions in [128, 266, 281], and more recent unified control and estimation approaches can be found in [171, 206].

As we look forward, the opportunities for new applications that will build on advances in control expand dramatically. The advent of ubiquitous, distributed computation, communication, and sensing systems has begun to create an environment in which we have access to enormous amounts of data and the ability to process and communicate that data in ways that were unimagined 20 years ago. This will have a profound effect on military, commercial, and scientific applications, especially as software systems begin to interact with physical systems in more and more integrated ways.

1.2.4 Asynchronous Control

Traditionally, the information between sensors, actuators, and controllers is exchanged at constant rates. The sampling frequency has to guarantee the stability of the system under all possible scenarios, and this can sometimes yield a conservative choice of the sampling period. Moreover, all tasks are executed periodically and independently of the state of the plant.

In recent years, the idea of taking into account the plant state to decide when to execute the control and sampling tasks has received renewed interest. In general, in this non-conventional sampling paradigm, information is exchanged in the control loop when a certain condition depending on the state is violated. Hence, there is an adaptation to the needs of the process at any time.

However, there is no uniform terminology when referring to this concept. One can find in the literature the terms event-based control, event-triggered control, send-on-delta control, level-crossing control, self-triggered control, minimum attention control, anytime attention control, and many more. All of them have basically the same idea, but vary in implementation. We will refer to *asynchronous control* or *asynchronous sampling* to cover all these approaches.

Despite its recent popularization, asynchronous sampling is not actually a new concept, and its origins date back to the late 1950s when it was argued that the most appropriate sampling method is to transmit data when there is a significant change in the signal [66]. Later, in the 1960s and 1970s, a heuristic method called *adaptive sampling* [60] was popularized. The objective was to reduce the number of samplings without degrading the system performance, evaluating in each interval the sampling period.

More recently, an event-based PID controller was implemented in [12] showing that the number of control updates was reduced without degrading the performance of the system. In [98], level-crossing control was applied to control the angular position of a motor with a low-resolution sensor.

The first analytical results were for first-order linear stochastic systems in [214], showing that under certain conditions the event-based control outperforms the periodic control. But the real impulse to the asynchronous control came out a few years later when many researchers realized the benefits of applying this theory to networked control systems. Section 1.4.2 will present a literature review of asynchronous control applied to NCSs as well as the main concepts used in this formalism.

1.3 Applications and Industrial Technology Over Network

Networked control systems have been finding application in a broad range of areas. Because of the attractive benefits detailed in Sect. 1.2.1, many industrial companies and institutes have shown interest in applying networks for remote industrial control purposes and factory automation [242]. The fact that many infrastructures and service systems of present-day society can naturally be described as networks of a huge number of simple interacting units increases the areas where NCSs can be applied.

For these reasons, these systems have a lot of potential applications, including environmental and pollution monitoring [113], control of water distribution networks [113], surveillance [16, 43], remote surgery [167], distributed power systems and smart grids [5, 24], mobile sensor networks [111, 198], formation control of autonomous vehicles [86, 229], haptics collaboration over the Internet [106], intelligent transportation systems [178], unmanned aerial vehicles [116] and chemical and petrochemical plants [267], just to name a few. Next, some of these NCS applications are detailed.

Wireless Sensor Networks

Built on nodes, are gaining a role of importance taking part of embedded systems. Embedded systems, by definition, interact with the physical world as sensors, actuators, and controllers that are programmed to perform specified actions. As the range of applications grows, the demand to perform incrementally complex tasks on the nodes also increases.

In general, each node has four main parts: I/O ports connected with sensors and actuators, a radio transceiver to transmit the information, a microcontroller, and an energy source, usually a battery. Each node can monitor physical or environmental conditions such as humidity, temperature, lighting, and so on.

The advantage of WSNs with respect to traditional technologies is enormous, as deploying and maintaining a geographically distributed wired network of thousands of nodes is impractical considering the distances among nodes. WSNs themselves have several applications such as surveillance, health care, air pollution, water quality, or industrial monitoring, some of which will be commented later on. WSNs are characterized by the mobility of nodes, power consumption constraints, or node failures, all of them challenges the control design has to deal with.

Biological Systems

Renewable energy-based systems and mitigation of the greenhouse effect are two of the main concerns in the present century. Large efforts are being done around the world trying to look for clean resources and new technologies to face these issues [243]. Also, the problem of quality and quantity of water resources is a global challenge for the upcoming years. Both, an adequate amount and quality of water are essential for public health and hygiene [113].

Bioprocesses technology or biotechnology is one of the emerging areas that can highly contribute to the challenging aspects mentioned above as well as to produce high-value products. Bioprocess operations make use of microbial, animal, and plant cells and components of cells, such as enzymes, to manufacture new biotechnological products (food industry, pharmaceutical products, biofuel), destroy harmful wastes (CO_2 mitigation) [59], or obtain large quantities of water with good quality.

For example, finding suitable biofuel crops so that the oil production could replace fossil fuel usage is a trendy line of research. In this regard, microalgae are seen as the bioprocess with great potential for biofuel production in the future. Microalgal biomass can reach up to 80 % of dry weight under certain stress conditions; they can be cultivated in high area yields compared to other crops; they have high oil content in some strains, low-water consumption is required, and it is possible to produce them on arid lands [20, 196].

As far as water scarcity is concerned, water treatment and desalination plants seem a solution to provide the possibility to use water everywhere. Recycled water is most commonly used for non-potable purposes, such as agriculture, landscape, public parks, and industrial applications, among others (Fig. 1.4).

The development of new technologies has made possible the monitoring and control of such biological processes. The integration of specific sensors and actuators in motes and the adaptation of the network function to the specific requirements that this type of application impose are identified as key features. They allow the distributed monitoring and control that improve the efficiency, productivity, and optimization of these large-scale systems.

Fig. 1.4 NCS applications to agriculture

Remote Surgery

This enables the surgeon to remotely operate on the patient with the help of a medical telerobot. Theoretically, it frees the surgeon from the operation room, protects the surgeon from radiation, and provides rescue for patients in areas of difficult access [167]. Hence, the new developed technology will help to remove distance barriers from surgery.

This ability can benefit patients who would otherwise go untreated, improve the quality of care since expert surgeons can proliferate their skills more effectively, and reduce costs by avoiding unnecessary patient and surgeon journeys [27]. Yet, other obstacles such as licensing, reimbursement, liability, etc., cannot be ignored.

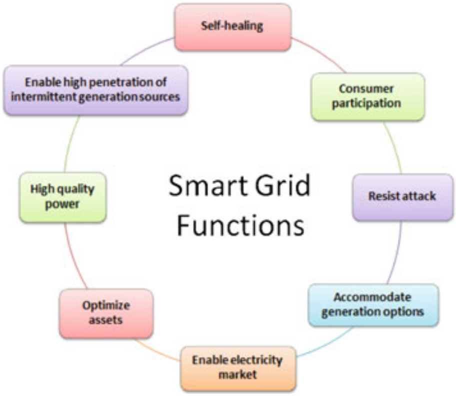
The first telesurgery prototypes were through wired connections [27, 167], but there are also some recent results on wireless remote surgery [158].

Smart Grids and Distributed Power Systems

We define a smart microgrid as a portion of the electrical power distribution network that connects to the transmission grid in one point and that is managed autonomously from the rest of the network [24]. The objective of transforming the current power grid into a smart grid is to provide reliable, high quality electric power in an environmentally friendly and sustainable way. To achieve this, a combination of existing and emerging technologies for energy efficiency, renewable energy integration, demand response, wide-area monitoring, and control is required (Fig. 1.5).

For instance, the so-called flexible AC transmission systems (FACTS) technology would allow to find the most efficient paths and better power production mixes and schedules. Additionally, the massive use of deployed sensors would make possible the measurement of the consumption of the end users at any time, weather data, or equipment condition. Monitoring, optimization, and control applications would

Fig. 1.5 Smart grids function diagram



increase the energy delivery efficiency and security by means of the dynamical computation of ratings and balance load and resources [227].

Intelligent Transportation Systems (ITS)

These are defined as those that utilize synergistic technologies and system engineering concepts to develop and improve transportation systems of all kinds. They provide innovative services related to different means of transport and traffic management. This will definitely achieve a *smarter* use of transport networks, making them safer and more coordinated.

Intelligent transportation technologies are based on wireless communications. The current trend is to develop new embedded system platforms that allow for more sophisticated software applications to be implemented, including model-based process control, artificial intelligence, and ubiquitous computing.

Applications of ITS are, for example, emergency vehicle notification systems, variable speed limits to control the traffic flow [263], travel time predictions [199, 221], collision avoidance systems, or dynamic traffic light sequence.

Formation Control

In many applications, a group of autonomous vehicles are required to follow a predefined trajectory while maintaining a desired spatial pattern [42]. Formation control has many applications. For example, in small satellite clustering, formation helps to reduce the fuel consumption and expand sensing capabilities. In military missions, a group of autonomous vehicles keeps a formation for exploratory purposes. Other examples include search and rescue missions, automated highway systems, detect, locate, and neutralize undersea mines by underwater vehicles, or mobile robotics [73].

Such autonomous vehicles can be coupled physically or through the control task to accomplish the specific task. Information is usually shared through a network to achieve the mission, and vehicles have only access to partial information when making decisions. Hence, new challenges arise in the control problem. For instance, communication is really weak in some scenarios, such as for underwater vehicles, where delays, reliability, and data rate constraints are very demanding (Fig. 1.6).

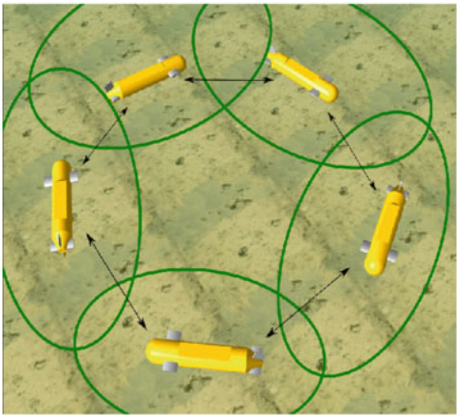


Fig. 1.6 Submarines in Formation

1.4 Networked Schemes: From Centralized to Distributed Techniques

Networked control systems are characterized by the transmission of sensor and/or control data through a shared network. Due to the finite bandwidth of the network, the flow of information is at discrete instances of time. This discontinuous flow is represented by dashed lines, whereas solid lines correspond to continuous signals. The flexibility that NCSs offer yields multiple possible architectures. In this book, we focus on the three most common configurations: centralized, decentralized, and distributed models.

1.4.1 Centralized and Decentralized Schemes

Since their inception, practically all the existing control and estimation techniques have been devised and developed for centralized schemes. In these schemes, every sensor or actuator of the plant is connected to a central agent that gathers all the data.

The advantages of centralized implementations have been widely exploited by systems engineers for decades. When a central agent collects all the available information of a system, monitoring and control tasks can potentially achieve high performances. In addition, there is a wide body of knowledge and a huge variety of techniques developed for centralized implementation, which means that the experimented practitioner can select the one that fits the system needs over a number of different possibilities.

In a centralized scheme (Fig. 1.7), the central unit receives the measurements $\{y_i(t)\}$ taken by the sensors in the plant and sends the control actions $\{u_i(t)\}$ back to the system.

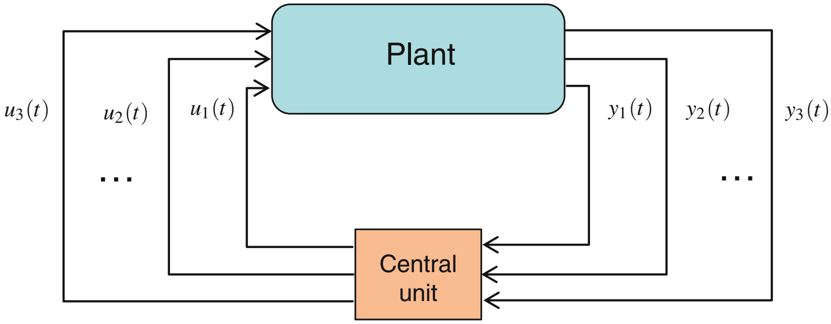


Fig. 1.7 Centralized architecture

In centralized NCS, there are different configurations depending on how the sensors (S), the actuators (A), and the controller (C) are located with respect to the network (see Fig. 1.8). Thus, the controller can be co-located with the sensor nodes (Fig. 1.8a), co-located with the actuators (Fig. 1.8b), or work as a remote controller (Fig. 1.8c):

- **Co-located with sensors** This architecture offers the advantage of providing the unaltered outputs instantaneously to, if necessary, reconstruct the state of the system. Thus, the synchronization of the controller with the sensors is a fair assumption in this case. The controller computes the control inputs that are transmitted

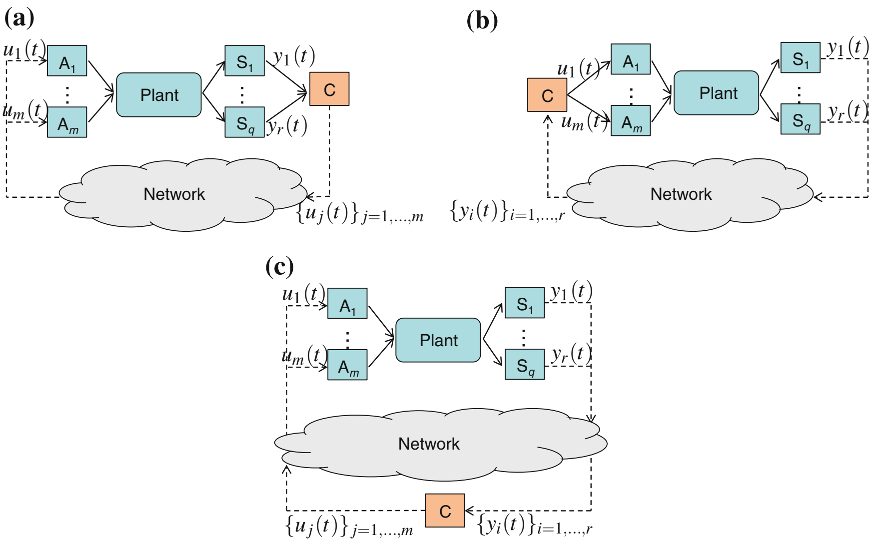


Fig. 1.8 Centralized models in NCSs

through the network at discrete instances of time (equidistant from each other or not) to the actuators, which might not have clocks' synchronization with other nodes.

- **Co-located with actuators** Information about the state of the system is transmitted from the sensor nodes to the controller through the imperfect channel. The controller will gather this information to calculate the control signals that are delivered to the actuators immediately.
- **Remote controller** This is the most general framework and the network is on both sides of the controller, which in general will not be synchronized with the other nodes in the network. Transmission of both sensor measurement and control inputs will suffer from the network imperfections.

In general, the control law is given as

$$u(t) = k(y(t)),$$

where $u(t) = (u_1(t) \ldots u_m(t))^T$ and $y(t) = (y_1(t) \ldots y_r(t))^T$.

Centralized architectures require to connect every device to a central node. This can be unsuitable in some applications, especially in the context of large-scale systems as, for instance, some of the applications detailed in Sect. 1.3. The implementation of centralized architectures in these kinds of systems may be challenging as important problems usually arise: technical difficulties to transmit all the system signals in real time, security issues, robustness against connection failures, high wiring costs, or excessive computational burden in the central controller.

In contrast, in decentralized schemes (Fig. 1.9), the tasks over the system are performed by a set of independent controllers suitably deployed [15, 213]. This way, each controller has access to local data and manages specific input/output channels. In decentralized architectures the computations can be carried out in parallel and the wiring costs are minimized, which also means reduced danger of breaking cables, less hassle with connectors, etc. Nonetheless, an important disadvantage of this approach is that the absence of communication between agents limits the achievable performance.

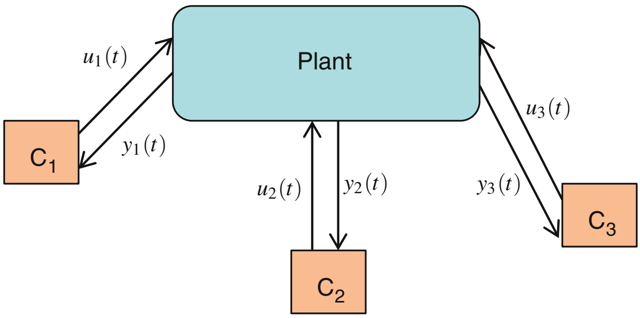


Fig. 1.9 Decentralized architecture

Each control unit C_i computes the control input $u_i(t)$ based on the local measurement $y_i(t)$. In general, the control law is

$$u_i(t) = k_i(y_i(t)).$$

1.4.2 The Middle Ground: Distributed Systems

Distributed systems are the middle ground that lies between decentralized and centralized solutions. As in decentralized architectures, in distributed systems the agents have access to local plant data. Thus, distributed architectures (Fig. 1.10) require lower levels of connectivity and less computational burden than centralized approaches.

However, as opposed to decentralized schemes, in this framework the controller nodes are endowed with communication capabilities and they can share information with a limited set of neighboring controllers (agents), which allows this approach to improve the performance. Therefore, distributed control systems (DCSs) are networked control systems where it is possible to trade-off between communication burden and control performance.

Distributed control and estimation techniques are becoming more and more popular with the development of wireless sensor networks, which has made easier the implementation of distributed control systems and has simplified deployment, migration, and decommissioning of networks, among other elements, see [225], or [2].

Nowadays, most vendors offer wireless-enabled product lines with different technologies (WSAN from ABB or OneWireless Network from Honeywell, to give a couple of examples). Although further efforts must be made to improve interoperability, computation capabilities, and connectivity of present devices, the scenario

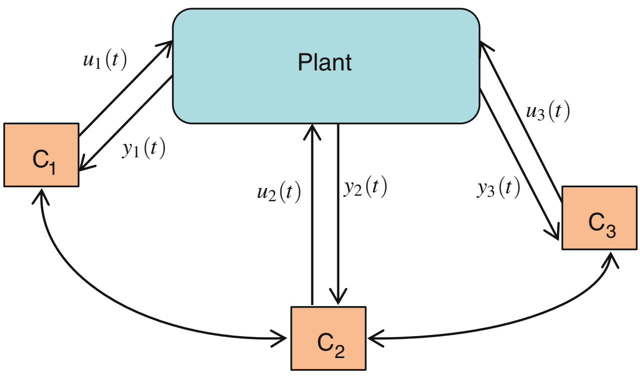


Fig. 1.10 Distributed architecture

where off-the-shelf components with the attributes required to implement sophisticated collaborative control/estimation schemes are available, is not so far in the future.

In contrast, compatibility, standardization, and integration of DCSs with other aspects of process control (human–machine interface, alarm systems, historical records, etc.) are still important issues to be resolved for a wider implementation of these systems. Besides, due to various design considerations, such as small size battery, bandwidth and cost, from the control design point of view, two types of interconnections between subsystems that compose the overall plant are distinguished. The first one is the physical interconnection, i.e., the state of a subsystem i directly drives the dynamics of another subsystem j. This fact can be used in the control design of the subsystem j to compensate this interconnection if the state of the subsystem i is available at j. The second type of interconnection is when the need for communication between the controllers comes from the fact that the system tries to achieve a common objective, such as for example, consensus. This leads to cooperative control. The usual terminology to refer to these systems in which the gathering of information from individual parts is used to control the global behavior of the networked system is *multi-agent systems*.

A scheme of a distributed NCS is depicted in Fig. 1.11. Each node i has a local controller C_i, which receives the local information $y_i(t)$ and also some but not all other information $y_j(t)$ from other subsystems (also called agents) measured at different instances of time. The agents that transmit information to i are known as its neighborhood (denoted by $\mathscr{N}_i$) and correspond to the ones that are interconnected with agent i. Hence, the control input $u_i(t)$ of the ith subsystem is

$$u_i(t) = k_i(y_i(t), \{y_j(t),\ j \in \mathscr{N}_i\}).$$

This scheme can be extended to more general frameworks. For instance, agents can exchange state estimations, different representation of sets, or other parameters.

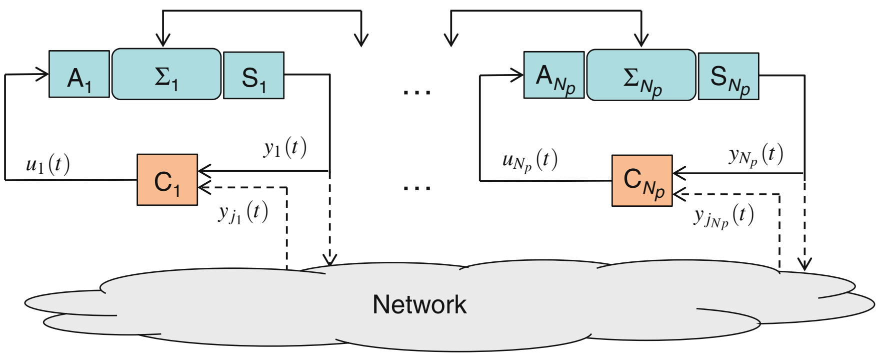


Fig. 1.11 Distributed NCS

1.5 Communication Through a Non-reliable Network

The main limitations imposed by an imperfect communication channel have been introduced in Sect. 1.2. To illustrate these concepts, let us consider the situation depicted in Fig. 1.12. There are two nodes in the network: the sender and the receiver. The first one wants to transmit some data to the other. The sender can be a controller, a sensor, or a subsystem of a distributed network, and the receiver can be an actuator, a controller, or another subsystem.

The first issue that makes different a networked system from a conventional control system is that the components are, in general, spatially distributed. As a consequence, the synchronization of the clocks of these components cannot be assumed normally, that is, measures of time are not equal. This phenomenon is illustrated in Fig. 1.13. On the left, sender and receiver have synchronized clocks. On the right, the measures of time differ from a value Δ, which is unknown by the nodes and is hard to compute. This makes difficult, for example, the measurement of delays.

The limited bandwidth that characterizes the network imposes that the amount of information transmitted per unit of time must be finite. Thus, on the one hand, analog signals must be transformed to be transmitted in a finite number of bits, which yields to quantization. The maximum amount of information that can be sent at once is given by the size of the packet, which depends on the network protocol. For instance, a packet can be divided into the control information, which provides the network needs to deliver the packet, and the user data, also known as *payload*. The size of the payload goes from 1500 bytes in Ethernet to 8 bytes in some Radio Frequency protocols used to communicate small devices.

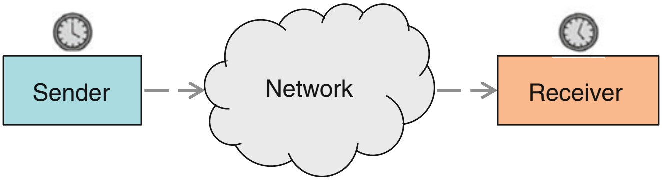


Fig. 1.12 Two nodes connected through the network

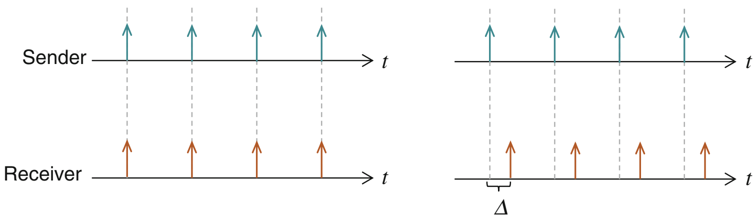


Fig. 1.13 Synchronization

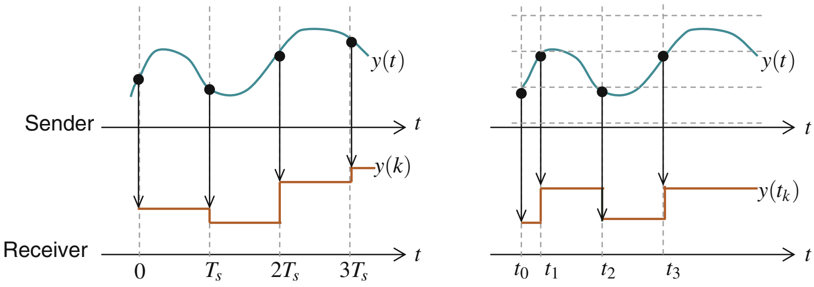


Fig. 1.14 Periodic and event-based sampling

On the other hand, the values of these signals can only be transmitted at discrete time instants. In this regard, there exist two alternatives as shown in Fig. 1.14. On the left, the measurements of the signal $y(t)$ in the sender node are sent to the receiver at equidistant instances of time given by a period T_s. Hence, the data received is $y(kT_s)$, $k \in \mathbb{N}$. For example, if we think that $y(t)$ is the output measured by a sensor, this technique corresponds to periodic or time-driven sampling, in the sense that the actions are taken based on the passing of time.

By contrast, when the transmission of data are not equidistant in time, and it is the value of the signal that matters in the decision of when to send the samples, we talk about event-driven or event-triggered sampling. Note that, for instance $t_1 - t_0 \neq t_2 - t_1$ on the right-hand side of Fig. 1.14. For a general value $k \in \mathbb{N}$, the difference between $t_{k+1} - t_k$ is called *inter-event time* and is denoted by T_k. Other authors also refer to this magnitude as broadcasting period [259].

The last concepts we want to illustrate are the network delays and the data dropouts. The reasons why these problems occur in a networked system have been discussed in Sect. 1.2. As stated there, some network protocols implement mechanisms to control the flow of packets. For instance, one common approach is to use *acknowledgment* (ACK), that is, the transmission of a small packet to confirm the reception of data. If ACK is not received after some waiting time (T_W), the sender deduces that the packet must have got lost and will try to retransmit the packet.

Let us illustrate these concepts with an example. For simplicity, assume that the sender and the receiver have synchronized clocks and periodic transmission of information as in Fig. 1.15. First, some data are sent at $t = 0$, which is received after some time τ_1 due to some delay in the transmission. Secondly, at $t = T_s$ new data are transmitted and dropped, for example, for some error in physical links. Data are retransmitted according to the protocol described above at the next sampling time. This causes the information to be finally received after some time τ_2. Hence, data dropouts and delays are related. In general, if n_p denotes the number of consecutive data dropouts and τ is the transmission delay, the effective delay is $n_p T_s + \tau$. For instance, if a control input $u(t)$ is computed by some controller node (*sender*), sent to an actuator node (*receiver*), and directly applied when received, the dynamics of the plant are in the continuous time

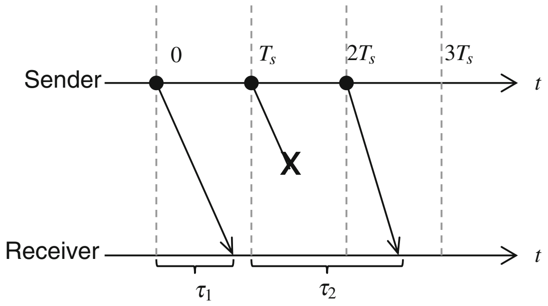


Fig. 1.15 Example of delays and data dropouts

$$\dot{x}(t) = f(x(t), u(t - (n_p T_s + \tau))),$$

or in the discrete time

$$x(k+1) = f(x(k), u(k - (n_p T_s + \tau))).$$

There usually exists an upper bound on the effective delay over which the system is unstable. Time-delay systems are by themselves an extensive research area in control theory. In this book, different strategies are proposed to deal with these kinds of problems and to compute bounds on the effective delay.

1.6 Asynchronous Control in NCSs

1.6.1 Event-Based Control Approaches in the Literature

In most implementations, an event is triggered when some error function exceeds a tolerable bound. How this error function and this bound are defined distinguishes the different approaches in the literature that are discussed next.

Deadband Control

If the error is defined as the difference between the state of the last event occurrence and the current state, and the bound is defined as a constant, an event is triggered whenever

$$\|\varepsilon(t)\| = \|x(t) - x(t_k)\| \leq \delta,$$

becomes positive, where t_k refers to the instant of the last event and t is the current instant of time. The value of δ determines, on one hand, the performance of the system and the ultimate set in which the state of the plant is confined around the equilibrium, and on the other hand, the average frequency of communication. Figure 1.16a, b depict two examples of deadband control for a first-order and a second-order system, respectively. Some works related to deadband control are [99, 226].

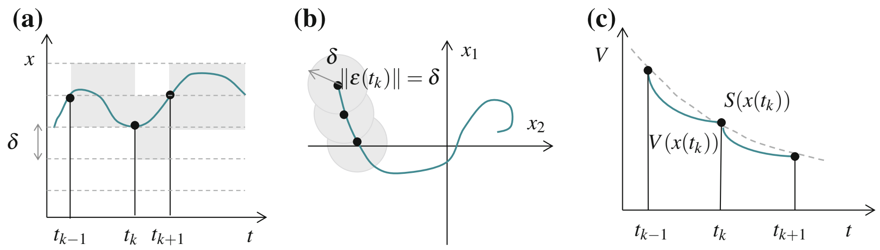


Fig. 1.16 Examples of triggering rules

Lyapunov Approaches to Asynchronous Control

Deadband control does not generally yield asymptotic stability. And so, some researchers have investigated triggering rules to fulfill this. One example is presented in [239] where the error is bounded by the state at the current time

$$\|\varepsilon(t)\| = \|x(t) - x(t_k)\| \leq \sigma\|x(t)\|.$$

This approach yields the asymptotic stability of the system but the inter-event times become shorter when the system reaches equilibrium. In [239] it is shown that a minimal inter-event time is guaranteed to exist only under suitable assumptions.

Other authors have exploited the idea of using Lyapunov methods to define the triggering rule [163]. An event is triggered when the value of the Lyapunov function of the closed-loop system for the last broadcast state reaches a certain threshold of performance $S(x, t)$ (see Fig. 1.16c):

$$V(x, t) \leq S(x, t).$$

This condition also guarantees that equilibrium is reached asymptotically.

Time-Dependent Event-Triggering

Recently, time-dependent triggering rules have been proposed to reach the equilibrium point asymptotically. In [89, 232], the trigger functions for linear interconnected systems and multi-agent systems, respectively, bound the error as

$$\|\varepsilon(t)\| \leq \delta e^{-\beta t}, \quad \delta, \beta > 0,$$

which has the aforementioned property, guaranteeing a lower bound for the inter-execution times. Note that this bound approaches to zero when $t \to \infty$, but still the Zeno behavior is avoided even in non-ideal network conditions.

Self-triggering

Sensor networks are a special case of networked control systems in which the energy consumption plays a crucial role. Thus, event-triggering approaches are convenient in sensor networks since the number of transmissions can be decreased. However, it has been discussed [8, 10, 165] that most of the energy consumed in a sensor node comes from the task of monitoring the measured variable(s) rather than the transmission. The asynchronous control strategies discussed above require the continuous monitoring of the state. For this reason, a new approach known as self-triggered control has emerged in the recent years.

Self-triggering policies determine the next execution time t_{k+1} by a function of the last measurement of the state x_k. The sensor nodes do not monitor the process until they are woken up at time t_{k+1}, they take the measurement and transmit it, and the next execution time is computed again. The concept of self-triggering was first suggested by [251]. Self-triggered control can be regarded as a software-based emulation of event-triggered control. It has been studied for linear systems [164, 256], nonlinear systems [8, 241], and applied to sensor and actuator networks in [9, 28, 165, 240].

A general problem of this scheme is the consideration of unknown effects, such as model uncertainties or unknown exogenous disturbances. To cope with all these effects conservative results have to be derived to guarantee the stability of the self-triggered control loop which may lead to relatively short sampling intervals in practice [258].

Minimum Attention Control (MAC) and Anytime Attention Control (AAC)

Minimum attention control maximizes the time interval between executions of the control task, while guaranteeing a certain level of closed-loop performance [7, 58]. It is similar to self-triggered control in the sense that the objective is to have as few control task executions as possible but it is typically not designed using emulation-based approaches. In [58] an approach based on extended control Lyapunov functions allows to solve the problem online alleviating the computational burden as experienced in [7]. However, MAC is by far less robust against delays or disturbances than event-based control. Similar problems present the so-called 'anytime control' methods, which are an alternative way to handle limited computation and communication resources [84, 93, 94]. The AAC proposed in [7] assumes that after each execution of the control task, the control input cannot be recomputed for a certain amount of time that is specified by a scheduler, and finds a control input that maximizes the performance of the closed-loop system.

Periodic Event-Triggered Control

Periodic event-triggered control strikes a balance between periodic control and event-based control. As self-triggered control, it avoids continuous monitoring of the system outputs while preserving the reduction in resource utilization. So, instead of checking the trigger condition continuously, this is only evaluated at instances of time defined by a period T_s.

The design methods that have been proposed [97] use Lyapunov-based trigger functions and provide the tools to check stability and performance for a given control gain and a sampling period. One additional advantage is that it guarantees a minimum inter-event time of (at least) the sampling interval of the event-triggering condition.

Model-Based Event-Triggered Control

All the approaches described above consider zero-order hold at the actuator, i.e., the control input computed at event times is held constant till the next event occurrence. Although this consideration of *doing nothing* between events simplifies the analysis, it has been shown that if a precise model of the plant is available, a control input generator can emulate the continuous-time state feedback loop and under certain constraints get a better performance than a zero-order hold [154]. The idea of taking advantage of a model in NCS and working in open loop is not new and was introduced in [181, 183], though the updates from the system are periodic, not event-triggered. However, emulation approaches such as [154] require synchronization of all the elements in the control loop, and this constraint is difficult to meet in the case of remote controllers or in distributed paradigms.

Asynchronous Control and Output Measurement

The triggering rules presented previously are all based on full state measurement, although in practice the full state is not often available. If the same setups are tried to be used for output feedback controllers, the Zeno behavior might occur, as pointed out in [57].

To solve this problem, the existing approaches to output-based asynchronous controllers can be categorized as observer-based or not. To the first category belong [141, 143]. The measured state is replaced in the trigger function by the estimated state provided by the observer [141] or the filter [143]. The second direction is to use a different structure in the controller. A dynamical output-based controller is proposed in [56]. Using mixed event-triggering mechanisms, the ultimate boundedness can be guaranteed while excluding the Zeno behavior. A level crossing sampling solution with quantization in the control signal is presented in [135], where an LTI continuous-time controller is emulated.

1.6.2 Event Definitions

We have just introduced the idea of event-based or event-triggered sampling (control). Let us formulate it in a formal way.

For simplicity, let us state full state measurement. If $x(t)$ is the state of the system and $x_b(t)$ accounts for the information available at the controller k, the error can be defined as

$$\varepsilon(t) = x_b(t) - x(t).$$

Then, the system is described as

$$\begin{aligned}\dot{x}(t) &= f(x(t), u(t)) \\ u(t) &= k(x(t) + \varepsilon(t))\end{aligned}$$

To formulate a general setup we assume that the triggering condition is given by some function $F_e(x(t), \varepsilon(t), t)$, which is jointly continuous in x and ε.

The sequence of event or *broadcasting* times t_k is determined recursively by the trigger function F_e as

$$t_{k+1} = \inf\{t : t > t_k, F_e(x, \varepsilon, t) > 0\}.$$

Most of the triggering conditions set a bound on the error function and, hence, the trigger function can be written as

$$F_e(x(t), \varepsilon(t), t) = ||\varepsilon(t)|| - \delta(x(t), t).$$

Of course, this includes the case of δ being a constant.

We say that the triggering scheme induces *Zeno behavior*, if for a given initial condition x_0 the event times t_k converge to a finite t^*. This means that $T_k = t_{k+1} - t_k$ tends to zero. This is, of course, an undesirable behavior since it requires the detection of events and transmission of data infinitely fast. Hence, the design of trigger functions F_e has to guarantee the existence of a lower bound for the inter-event time T_k.

Equivalent definitions can be given for discrete time systems. In this case, the event times are a multiple of the sampling period and, therefore, the Zeno behavior is excluded by construction.

This formalism is based on the continuous monitoring of the state $x(t)$, which requires waste of computational resources and, as a consequence, of energy. A self-triggered implementation is given by a map $F_h : \mathbb{R}^n \to \mathbb{R}$ determining the next sampling time t_{k+1} as a function of the state $x(t_k)$ at the time t_k, i.e.,

$$t_{k+1} = t_k + F_h(x(t_k)).$$

The most common implementation of F_h consists of predicting future states of the plant based on a model of the system:

$$\begin{aligned} \dot{x}_m(t) &= f(x_m(t), u(t)),\ x_m(t_k) = x(t_k) \\ u(t) &= k(x(t_k)),\ t_k < t < t_{k+1}. \end{aligned}$$

Then, the Lyapunov function at the current event time t_k, $V(x(t_k))$, and at the future times t, $V(x_m(t))$, are evaluated. The next event time t_{k+1} will be the first value of t such that

$$\Delta V(x_m(t), x(t_k)) = V(x_m((t)) - V(x(t_k))\gamma(t, t_k) \geq 0.$$

The function $\gamma(t, t_k)$ can take the value 1 and, therefore, the next sampling time will be when the computed Lyapunov function $V(x(t))$ exceeds the current value $V(x(t_k))$. An alternative that ensures the exponential decrease of the Lyapunov function is $\gamma(t, t_k) = e^{-\alpha(t-t_k)}$, $\alpha > 0$.

1.7 Stability and Performance Measurements

As in conventional systems, guaranteeing the stability of a networked system is essential. The two main approaches for verifying stability found in this book are spectral theory for linear systems and Lyapunov functions for both linear and non-linear systems. Additionally, in order to address delays, extensions of the Lyapunov function concept in the sense of Krasovskii or Razumikhin can be used. In general, the Lyapunov–Krasovskii theory yields less conservative results, and this will be the preferred approach in this monograph.

We next introduce some concepts that will be used throughout the book.

Definition 1.1 The state of the system $x(t)$ is asymptotically stable if

$$\lim_{t\to\infty} ||x(t)|| = 0.$$

where $||\cdot||$ denotes an arbitrary matrix or vector norm.

Definition 1.2 A square matrix A is said to be Hurwitz if every eigenvalue of A has strictly negative real part, that is,

$$\mathbb{R}e[\lambda_i(A)] < 0,$$

for each eigenvalue λ_i.

It is also called the stability matrix, because then the differential equation $\dot{x}(t) = Ax(t)$ is asymptotically stable.

Analogous definitions can be given for a discrete-time system $x(k+1) = Ax(k)$. In that case, the condition over the eigenvalues is to lie inside the unit circle.

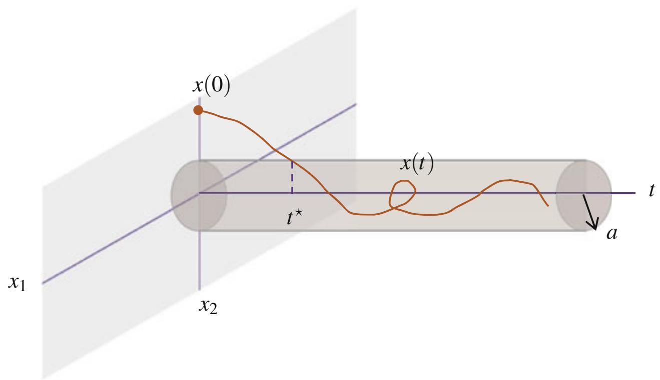


Fig. 1.17 Ultimate boundedness

For some triggering conditions in event-based control (deadband control), the asymptotic stability of the system cannot be guaranteed. A more appropriate stability definition is given by ultimate boundedness which is illustrated in Fig. 1.17 and is defined next.

Definition 1.3 The solution $x(t)$ of a continuous-time system $\dot{x}(t) = f(x(t), u(t))$ is globally uniformly ultimately bounded (GUUB) if for every $x(0) \in \mathbb{R}^n$ there exists a positive constant a and a time $t^\star$ such that the following holds:

$$x(t) \in \Omega_t \triangleq \{x : ||x|| < a\}, \forall t \geq t^*.$$

An interesting phenomenon that has been observed in some event-based control systems is the occurrence of oscillatory behaviors around the equilibrium point, and more specifically, of limit cycles. A limit cycle is defined as an isolated closed curve.

The stability of limit cycles can be defined in similar terms as for equilibrium points. For instance, if Γ is the closed orbit (limit cycle), we say that Γ is globally asymptotically stable if for any $x(0)$

$$\lim_{t\to\infty} \inf_{y\in\Gamma} ||x(t) - y|| = 0.$$

Proving global asymptotic stability is hard and most of the existing results are only for local stability [35, 82].

Limit cycles are inherent properties of nonlinear systems, and the fact that event-based control/sampled systems are nonlinear even though the dynamics of the original system is linear, is the reason why this phenomenon occurs in some types of deadband controllers. This will be studied in Chap. 2.