Chapter 2
Background

This chapter briefly reviews the fundamentals of the main topics treated in this book. Three sections cover concepts, definitions and schemes relating to sewer networks, model predictive control (including hybrid models), and fault tolerance mechanisms. Moreover, bibliographical references to relevant scientific contributions in journals and important congress and research reports are given for each topic, and their contents are briefly presented and discussed.

2.1 Sewer Networks: Definitions and Real-time Control

2.1.1 Description and Main Concepts

First of all, this section introduces some important concepts and relevant definitions relating to sewer networks. The basic concepts to define are what a sewer network is and what its objective is. In general, *sewers*[1] are pipelines that transport wastewater and rain drains from city buildings and streets to treatment facilities. Sewers connect these items to the horizontal mains. The sewer mains often connect to larger mains, and these are linked to a wastewater treatment plant. Vertical pipes called *manholes* connect the mains to the surface. Sewers are generally gravity powered, although pumps may be used if necessary.

The main type of wastewater that is collected and transported by a sewer network is generally *sewage*, which is defined as the liquid waste produced by humans. This typically contains washing water, faeces, urine, laundry

[1] The word *sewer* comes from the old French *essouier* (to drain), which in turn derives from the Latin *exaquaria* (*ex*-, meaning "out", and *aquaria*, the feminine of *aquarius*, meaning "pertaining to water".

waste, and other liquid or semi-liquid wastes from households and industry. Such a sewer network is known as a *sanitary sewer network*.[2]

Similarly, there are the *storm sewer mains*, which are large pipes that transport stormwater run-off from streets to natural sewage bodies in order to avoid street flooding. When a given network collects not only sewage from houses and industry but also stormwater run-off, it is called a *unitary network* or a *combined sewage system* (CSS). Such sewer networks were built in many older cities, as a mixed system was cheaper to build, but they encounter problems during heavy rains. Hence, these combined systems were designed to handle storms of a certain size. When the sewer mains were overloaded, the sewage would pass from the sewer system into a nearby body of water through a relief main to prevent flooding in the streets or in houses and buildings. This book considers the case of unitary networks, so all concepts and descriptions presented hereafter relate to such systems.

According to the literature, sewer networks can be considered a collection of elements that each provide a specific function. The discussion below focusses on a few typical elements found in a sewer network, while Figure 2.1 gives an idea of how they are interrelated using a scheme for a very small and simple sewer network. Some of the figures presented here relate to the sewer network of Barcelona, which is described in Chapter 3 and forms the case study of this book.

Transport Links

These elements are used not only to connect different parts of the network but also as storage elements when they become sufficiently extensive (*i.e.*, when the total capacity of these links becomes large enough). Regarding their hydrodynamics, This fact also means that hydrodynamic phenomena relating to the manipulation of the sewer network inflow using throttle gates must be considered when this approach is used. In such cases, the so-called *backwater effect*[3] may occur, which adds an extra degree of complexity when attempting to model and simulate the dynamical behaviours of such elements. Moreover, due to the size of the network, transport delays and other nonlinear dynamics should be taken into account when the dynamics of these elements are described. Within a sewer network, there are a variety of links that vary in size, geometric shape, specific function, *etc*. Figure 2.2 shows a large-diameter sewer main found in a real sewer network.

---

[2] It is also called a *foul sewer*, especially in the UK.

[3] *Backwater* is water that is held or pushed back; for example by a dam or current.

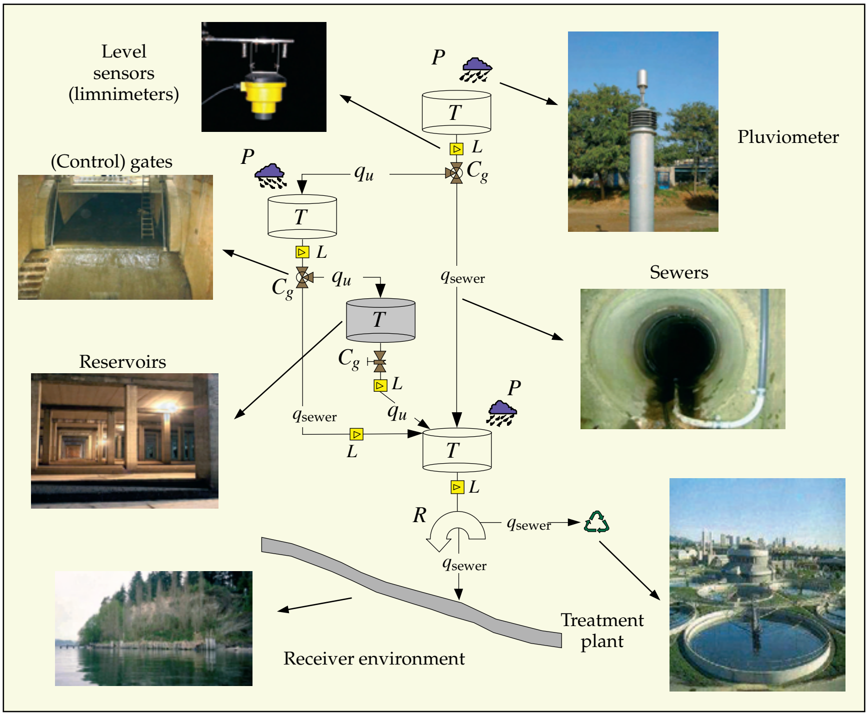

Figure 2.1 Typical components found in the scheme of a simple sewer network

Figure 2.2 Large-diameter sewer. Taken from CLABSA (2007)

Tanks or Reservoirs

These elements are used as dual-function storage devices. Their first function is to ensure that their outflow is laminar, which means that the inflow

Figure 2.3 Inside a retention tank. Taken from CLABSA (2007)

is greater than the outflow. This facilitates easier manipulation of the flows in elements downstream, particularly during heavy rain episodes. Second, these elements have an environmental function in the sense that they retain highly contaminated sewage. This retention prevents the spillage of sewage onto beaches, rivers and ports, thus allowing it to be treated by wastewater treatment plants (WWTPs). On the other hand, the degree of contamination of the sewage retained in a tank decreases due to the sedimentation caused by the retention process.

In terms of phenomenological models, these elements can include overflow capabilities, which means that when the volume of sewage exceeds the maximum capacity of the tank, a new flow can appear. This flow is the sewage that cannot be stored in the tank. However, some models incorporate the manipulation of a redirection gate located in the tank's input as an alternative strategy to cope with overflows. In this case, an overflow is not a nominal mode of operation; it becomes a security mechanism. The maximum capacity of the tank would be a operational constraint on the input gate management policy. The usefulness of each of these approaches depends on the model and the control strategy applied to the sewer system. The inside of a retention tank is shown in Figure 2.3.

Gates

In a sewer network, gates are used as control elements, as they can be used to change the flow downstream. Depending on the actions they perform, gates can be classified as follows:

Figure 2.4 Typical retention gate. Taken from CLABSA (2007)

Redirection gates. These gates are used to change the direction of sewage flow. They can be located before a reservoir or at any position in the network where sewage redirection may be required.

Retention gates. These are used to retain the sewage flow at a certain point in the network. They are generally located at the outputs of reservoirs, where they can be used to control whether sewage is kept in the tank and thus whether it undergoes the beneficial wastewater sedimentation process.

In sewer network control, when a system model based on sewage flows is considered, the control variables may correspond to the manipulated outflows from the network gates. Taking into account the scheme in Figure 1.4, where the global control level computes these outflows, local controllers handle the mechanical actions of the physical gates (actuators) by using these computed outflows as the set-points of those controllers. This procedure avoids the need to consider inherent nonlinearities associated with the dynamic behaviour of the gate. Figure 2.4 shows a large retention gate.

Nodes

According to Marinaki and Papageorgiou (2005), these elements correspond to points where sewage flows are either propagated or merged. Propagation means that the node has one inflow and one outflow, so the objective of this point is – for example – to connect sewer mains with different geometric shapes. On the other hand, merging means that two or more inflows are merged into a greater outflow. Thus, there are two types of nodes:

- Nodes with one inflow and multiple outputs (splitting nodes).
- Nodes with multiple inputs and one output (merging nodes).

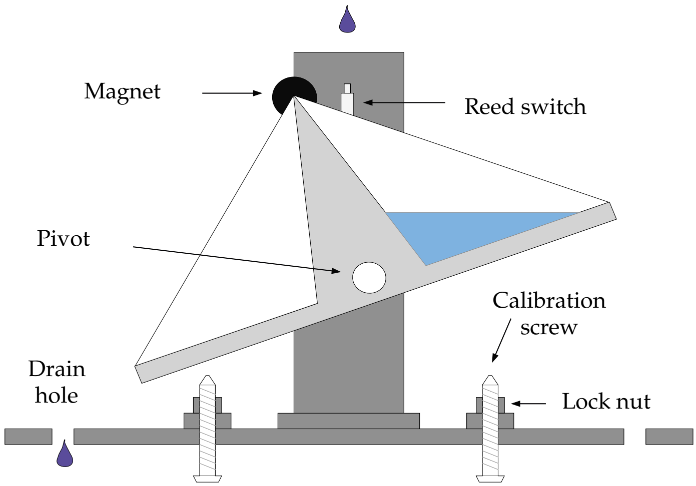

Figure 2.5 Scheme of a tipping bucket rain gauge

Each of these elements inherently has a maximum outflow capacity, which leads to an overflow situation under given conditions. Hence, each node element has a sewage outflow output as well as a possible output for overflow. Such elements are called *weirs*. These exhibit switching behaviour in their dynamics, and such behaviour can be difficult to take into account in many system models.

Instrumentation

In a sewer network, many variables must be measured in order to implement an RTC scheme. The main devices that are used to fulfil this goal are outlined below.

Rain gauges. Rain can be considered to be the main exogenous input to a sewer network. Hence, it is necessary to measure rain intensity in order to compute rain inflow. Rain intensity is measured with a *tipping bucket* rain gauge (see Figure 2.5). This gauge technology uses two small *buckets* mounted on a fulcrum (balanced like a seesaw). The tiny buckets are manufactured with tight tolerances to ensure that they hold an exact amount of precipitation. The tipping bucket assembly is located underneath the rain sewer, which funnels the precipitation to the buckets. As rainfall fills one of the tiny buckets, it overbalances and tips, emptying itself, while the other bucket pivots into place for the next reading. Each tipping event triggers a small switch which activates electronic circuitry that transmits the count to a console indoors; this records the event as a particular amount of rainfall. Once the rain intensity has been determined, the rain inflow can be computed using the procedure proposed and explained in Chapter 3.

Limnimeters. These devices measure sewage levels within the sewer mains. They are placed at strategic points in the network, and the data they provide are related to the sewage volume and flow by means of Manning's formula; see Chapter 3. They are mainly used at locations where the slope of the sewer main allows the sewage to flow due to the effect of gravity.

Velocity sensors. Depending on the geometry and topology of the sewer main considered, the flow information can be inaccurate when it is deduced from level measurements. Therefore, velocity sensors are used to measure the sewage velocity at a specific location in the network. This information enables the sewage flow to be computed more accurately, thus ensuring that situations where the sewage level remains constant in a sewer main with almost no slope and sewage flow do not occur.

Radar. An alternative way to measure the rain intensity is via *weather radar*. Weather radar is an instrument that is used to obtain a detailed description of the spatial and temporal rainfall field. This information is needed to create a hydrological model of a certain region with sufficient resolution. However, such devices are complex instruments. They measure a specific property of raindrops. This property is related to the fraction of the radar beam power that bounces off the target and is detected by the radar. This property, known as the rainfall reflectivity, is indirectly related to the rainfall intensity (through the size distribution of the raindrops). It is also indirectly related to the intensity of the rainfall that reaches the ground (GRAHI, 2007). For more on these instruments and how they are used in sewer network management and control, see Sempere-Torres *et al*. (1999), Bringi and Chandrasekar (2001), Bech *et al*. (2006) and Velasco-Forero *et al*. (2009), among others.

Pumping Stations

Once a rain episode has finished, the part of the sewage volume left in the tanks after the rest of the sewage has exited under the influence of gravity is directed to a WWTP. Two types of elements may be needed to do this: retention gates (discussed earlier) and pumping stations. Pumping stations are needed to remove the sewage from the tanks. Hence, these pumping stations are also manipulated, allowing flow control downstream. These elements usually have complex control strategies that depend on the management policies adopted. Notice that a pumping station can consist of different groups of devices that pump the sewage in a pre-established order, according to sequences of operation. There are many types of pumping stations in sewage systems, such as wet pit pumping stations, pneumatic ejector pumping stations, and dry well pumping stations. Figure 2.6 shows a pumping station of a sewer network.

Figure 2.6 Typical pumping station for a reservoir. Taken from CLABSA (2007)

Treatment Elements

These are basically plants where physicochemical and biological processes are employed to remove organic matter, bacteria, viruses and solids from wastewaters before they are released to rivers, lakes and seas. Such elements are very important components of sewer networks since they help to preserve the ecosystem and maintain the environmental balance within the water cycle. In this context, separating storm sewer mains from waste sewers is a useful strategy, as the huge amount of rainwater inflow generated during a storm can overwhelm WWTPs, resulting in the release of untreated sewage into the environment. Some cities around the world have dealt with this issue by adding large storage tanks or ponds to hold the sewage/rainwater until it can be properly treated. Another way to deal with this is to design a control strategy that prevents all types of pollution and combined sewage overflow (CSO) from the sewer network and thus damage to the environment. Figure 2.7 shows a WWTP located at Baix Llobregat, Barcelona (Spain).

2.1.2 RTC of Sewage Systems

This section explores contributions to the literature on the RTC of sewer networks, although it also takes into account aspects of modelling sewage systems due to the close relation between modelling and control for these systems. The RTC of sewage systems plays an important role in meeting increasingly restrictive environmental regulations aimed at reducing the release of untreated wastewater or pollution into the environment. Reducing pollution often requires major investments in infrastructure within city lim-

Figure 2.7 View of the WWTP at Baix Llobregat, Barcelona (Spain). Photograph from http://www.depurbaix.cat/

its, and thus any improvements that can be made to the efficiency of the existing infrastructure (for example through the use of improved control strategies) are of great interest. The advantage of sewer network control has been demonstrated by a number of researchers over the past few years (see, *e.g.*, Gelormino and Ricker, 1994; Pleau *et al.*, 1996; Marinaki, 1999; Pleau *et al.*, 2005; Marinaki and Papageorgiou, 2005). One common control strategy for dealing with urban drainage systems is MPC, as described by Gelormino and Ricker (1994), Pleau *et al.* (2001), Marinaki and Papageorgiou (2005) and Puig *et al.* (2009).

This control methodology is suitable due to the multi-input, multi-output and multi-objective nature of the urban drainage control problem. Additionally, one of the main management goals is to exploit the existing infrastructures of these networked systems, accounting for their size and operational limits. All of these system characteristics are conveniently handled by MPC and its capacity to deal with constraints, multiple control objectives, disturbances, delays, *etc*.

One very important aspect of sewer network management is network modelling. Several modelling approaches have been presented in the literature (see, among others, Ermolin, 1999; Marinaki, 1999; Duchesne *et al.*, 2001; Marinaki and Papageorgiou, 2005; Dellana and West, 2009). Due to the complex nature of the problem, several hydrological models have been proposed (Pleau *et al.*, 1996; Zhu *et al.*, 2001; Vanrolleghem *et al.*, 2005). Sewer networks are systems with complex dynamics, since the sewage flows through sewer mains in open canals. As will be discussed later, flows along open canals can be described by the Saint-Venant partial differential equations, which can be used to perform simulation studies but are highly complex to solve in real time. Control-oriented modelling techniques have also been reported in the literature (see Duchesne *et al.*, 2001; Ocampo-Martinez *et al.*, 2006b). However, when an RTC strategy is implemented, the complexity of the model could imply a high computational burden and thus difficulties in computing a control sequence for a desired performance (Zhu *et al.*, 2001; Marinaki and Papageorgiou, 2005). Additionally, a high computational burden can arise when models with hundreds or even thousands of dynamic variables are considered. Such models of huge dimensions are commonly associated with large-scale systems. The purpose of a given dynamical model is often to perform simulation studies, so they can vary from highly complex partial differential equations to simpler conceptual models.

In an early MPC-related work (Gelormino and Ricker, 1994), a linear model of a sewer network was used for prediction. Linear models that were identified for simulating flows in urban drainage networks using rain measurements as known input were also reported to give good results in Previdi *et al.* (1999). The use of nonlinear models for the predictive control of urban drainage systems has also been reported (see, among others, Ricker and Lee, 1995; Marinaki and Papageorgiou, 1998; Shen *et al.*, 2009; Dellana and West, 2009).

Improvements in prediction achieved by using nonlinear models need to be compared to the uncertainty introduced due to the error in predicting the rain over the horizon. Short-term rain prediction or *nowcasting* is an active field of research (see, among many others, Smith and Austin, 2000; Xu *et al.*, 2005; Chan and Tam, 2005; Suresh, 2007). Using a combination of radar and rain gauge measurements as well as advanced data processing, rain prediction has improved greatly in recent years, and the potential for its use in the predictive control of urban drainage systems has been highlighted in Yuan *et al.* (1999), Elliott and Trowsdale (2007), Thorndahl and Willems (2008) and Villarini *et al.* (2010).

An operational model of an urban drainage system is therefore a set of equations that can rapidly but approximately evaluate the hydrological variables of the network and their responses to gate control actions. In Ricker and Lee (1995), nonlinear MPC was implemented across a large-scale system with 26 states and ten manipulated inputs. It was shown that a complex nonlinear model is always better, but differences from the results obtained with linear MPC may be too small to justify the extra effort required for NMPC. Marinaki and Papageorgiou (1997) state that the use of simpler models for the optimisation-based control of sewer networks is justified because:

- The impact of model inaccuracies is reduced by solving the control problem iteratively and updating inflow predictions and initial conditions
- The details of the local elements and catchments are considered in local control loops.

Regarding control strategies, extensive research has been carried out on the RTC of urban drainage systems. Comprehensive reviews that include a discussion of some existing implementations are given in Schilling *et al.* (1996) (and references therein), Zug *et al.* (2001) and Schütze *et al.* (2008), while practical issues are discussed by Lahoud *et al.* (1998), Schütze *et al.* (2002a) and Akridge and Carty (2006), among others. The common idea here is to use optimisation techniques to improve system performance in terms of avoiding street flooding, preventing CSO discharges, minimising pollution, obtaining uniform utilisation of the sewer system's storage capacity, and, in most cases, minimising running costs, among another objectives (see Ermolin, 1999; Weyand, 2002; Schütze *et al.*, 2002b, 2004). In this way, Gelormino and Ricker (1994) proposed the implementation of MPC of the Seattle urban drainage system. In this work, the authors laid out the fundamental concepts involved in applying these techniques to sewer networks: defining appropriate cost functions, creating and maintaining models, and using prediction to minimise the effect of uncertainty in rain estimation – an aspect that is crucial to the proper operation of such systems in closed loops.

Marinaki and Papageorgiou (1997) proposed the application of optimal control in a previously proposed hierarchical structure (Papageorgiou, 1985). This methodology suggests an RTC structure that combines high efficiency and low implementation cost, and has the following three layers:

- An adaptation layer, where the inflow is predicted (rain) and state estimation is performed in real time
- An optimisation layer, which is responsible for global control and the computation of reference trajectories
- A decentralised control layer, which is responsible for realising the control trajectories.

A similar idea of hierarchical control for RTC can be found in Schütze *et al.* (2004) and Brdys *et al.* (2008). In Marinaki and Papageorgiou (2001), the authors combine the work presented in Marinaki and Papageorgiou (1997) with the receding horizon philosophy; that is, optimal control with a finite time horizon and prediction with a sliding time window. Duchesne *et al.* (2004) implements the global control level introduced in Figure 1.4 within the framework of MPC to minimise the overflow from combined sewer mains during rainfall in the urban area drained by the Marigot interceptor in Laval, Canada. The results showed that allowing surcharged flows in the interceptor during rainfall leads to a significant reduction in overflows.

Although the optimisation methods are applied and (more generally) control procedures are developed in order to determine the optimal (best possible) control action under the given conditions, a suboptimal control decision can be sufficient for RTC so long as it can be ensured that this decision does not cause the system to perform less optimally than in the no-control scenario. However, by applying specific model conditions and MPC strategies, it is possible to ensure that the best solution is obtained.

2.2 MPC and Hybrid Systems

2.2.1 MPC Strategy Description

MPC, also referred to as *model-based predictive control*, *receding horizon control*, or *moving horizon optimal control*, is one of the few advanced methodologies that has had a significant impact on industrial control engineering. MPC is applied in the process industry since it can handle multi-variable control problems in a natural form, it can take into account actuator limitations, and it allows constraints to be considered. Predictive control methods are developed around certain common ideas that are discussed in, *e.g.*, Maciejowski (2002), Camacho and Bordons (2004), Goodwin *et al.* (2005) and Rawlings and Mayne (2009), which are basically:

- The explicit use of a model to predict the process output in a time horizon
- The production of a control sequence that minimises a cost (objective) function
- The application of the first control signal from the computed sequence and the displacement of the horizon towards the future.

MPC is a wide field of control methods that are developed around a set of basic elements that the methods have in common. Its parameters can be modified to give different algorithms. These main elements are as follows:

- A prediction model, which should capture all of the process dynamics and can be used to predict the future behaviour of the system output.
- An objective (cost) function, which is, in its general form, the element that penalises deviations of the predicted controller outputs from a reference trajectory. It combines a set of performance indices for the dynamical system considered.
- Control signal computation.

This control strategy presents important advantages over other control methods. Some of these advantages are outlined below (Bordóns, 2000):

- It is very easy to for people lacking a deep knowledge of control to use. Its concepts are quite intuitive and it is relatively easy to tune.
- Can be used to control a wide variety of processes, from those with simple dynamics to systems with big delays, unstable systems, and non-minimum phase systems.
- It is very useful for multi-variable systems.
- It has built-in delay compensation.
- It allows the use of constraints, which can be added during the design process.

However, MPC also has some disadvantages, such as a high computational burden when the control law is calculated. In any case, the main potential problem of this strategy is its strong dependence on the accuracy of the system model. The algorithm for MPC controller design is based on the previous knowledge of the system's behaviour, so its performance is related to the quality of the representation of the plant.

MPC Formulation

In most of the cases presented in the research literature, the MPC formulation is expressed in state space. However, in order to present the strategy in a compact and simple but clear way, let

$$x_{k+1} = g(x_k, u_k) \tag{2.1}$$

be the mapping of states $x_k \in \mathbb{X} \subseteq \mathbb{R}^n$ and control signals $u_k \in \mathbb{U} \subseteq \mathbb{R}^m$ for a given system, where $g : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ is an arbitrary system state function and $k \in \mathbb{Z}_+$. Let

$$\mathbf{u}_k(x_k) \triangleq \left(u_{0|k}, u_{1|k}, \ldots, u_{H_p-1|k}\right) \in \mathbb{U}^{H_p} \tag{2.2}$$

be the input sequence over a fixed-time prediction horizon H_p. Moreover, the *admissible input sequence* with respect to the state $x_k \in \mathbb{X}$ is defined by

$$\mathscr{U}_{H_p}(x_k) \triangleq \left\{\mathbf{u}_k \in \mathbb{U}^{H_p} | \mathbf{x}_k \in \mathbb{X}^{H_p}\right\}, \tag{2.3}$$

where

$$\mathbf{x}_k(x_k, \mathbf{u}_k) \triangleq \left(x_{1|k}, x_{2|k}, \ldots, x_{H_p|k}\right) \in \mathbb{X}^{H_p} \tag{2.4}$$

corresponds to the state sequence generated by applying input sequence (2.2) to system (2.1) from initial state $x_{0|k} \triangleq x_k$, where x_k is the measured or estimated current state (initial condition). Hence, the receding horizon approach is based on the solution of the open-loop optimisation problem (OOP)

$$\min_{\{\mathbf{u}_k \in \mathscr{U}_{H_p}\}} J\left(\mathbf{u}_k, x_k, H_p\right), \tag{2.5a}$$

subject to

$$H^{\mathrm{u}}_{iq}\mathbf{u}_k \leq b^{\mathrm{u}}_{iq}, \tag{2.5b}$$

$$G_{iq}\mathbf{x}_k + H_{iq}\mathbf{u}_k \leq b_{iq}, \tag{2.5c}$$

$$H^{\mathrm{u}}_{eq}\mathbf{u}_k = b^{\mathrm{u}}_{eq}, \tag{2.5d}$$

$$G_{eq}\mathbf{x}_k + H_{eq}\mathbf{u}_k = b_{eq}, \tag{2.5e}$$

where $J(\cdot) : \mathbb{X}_f(H_p) \mapsto \mathbb{R}$ is the cost function with its domain in the *set of feasible states* $\mathbb{X}_f(H_p) \subseteq \mathbb{X}$ (Lazar *et al.*, 2006), H_p denotes the *prediction horizon* or *output horizon*, and G_{iq}, G_{eqe}, H_{iq}, H_{eq}, H^{u}_{iq}, H^{u}_{eq}, b_{iq}, b_{eq}, b^{u}_{iq}, and b^{u}_{eq} are matrices with suitable dimensions. In sequence (2.4), $x_{k+i|k}$ denotes the prediction of the state at time $k+i$ performed at k, starting from $x_{0|k} = x_k$. When $H_p = \infty$, the OOP is called the *infinite horizon problem*; when $H_p \neq \infty$, the OOP is called the *finite horizon problem*. Constraints employed to guarantee the system's stability in a closed loop would be added in (2.5b)–(2.5e). In particular, constraints (2.5d)–(2.5e) are related to static elements where an equality condition must hold.

Assuming that the OOP (2.5) is feasible for $x \in \mathbb{X}$, *i.e.*, $\mathscr{U}_{H_p}(x) \neq \emptyset$, there is an optimal solution given by the sequence

$$\mathbf{u}^*_k \triangleq \left(u^*_{0|k}, u^*_{1|k}, \ldots, u^*_{H_p-1|k}\right) \in \mathscr{U}_{H_p}, \tag{2.6}$$

and then the receding horizon philosophy sets

$$u_{\mathrm{MPC}}(x_k) \triangleq u^*_{0|k}, \tag{2.7}$$

and disregards the computed inputs from $k = 1$ to $k = H_p - 1$, with the whole process repeated at the next time instant $k \in \mathbb{Z}_+$. Expression (2.7) is known in the MPC literature as *the MPC law*.

Remark 2.1. The concept of a *control horizon*, denoted by H_u, was not considered in this statement of the MPC strategy. This concept implies the determination of a number of control actions that may be less than or equal to the number of time predictions (H_p), *i.e.*, $H_u \leq H_p$. The case $H_u > H_p$ is unusual but can be achieved (Maciejowski, 2002). ◊

Summarising, Algorithm 2.1 briefly describes the basic computation procedure for an MPC law.

Algorithm 2.1 Basic procedure for MPC law computation

1: $k = 0$
2: **loop**
3: $\quad x_{k+0|k} = x_k$
4: $\quad \mathbf{u}^*_k(x_k) \Leftarrow$ solve OOP (2.5)
5: $\quad$ Apply only $u_k = u^*_{k+0|k}$
6: $\quad k = k+1$
7: **end loop**

The way that MPC works is often compared to playing chess. Both follow the philosophy of planning the best sequence of future moves in order to achieve a pre-established objective (in the case of chess: to win the game). At the end, just one movement can be performed.

2.2.2 *Hybrid Systems*

Dynamical systems exhibit several phenomena produced by the interactions of signals of different kinds. In general, systems consist of both continuous and discrete components; the former are typically associated with physical first principles, and the latter with logic devices such as switches, digital circuitry, software code, *etc*. This mixture of logical conditions and continuous dynamics gives rise to a *hybrid system*.

In sewer networks, there are several phenomena (overflows in sewer mains and tanks) and elements in the system (redirection gates and weirs) that exhibit different behaviours depending on the flow/volume present within the network. This fact leads naturally to the use of hybrid models to describe such behaviours. The hybrid models considered here belong to the class of discrete-time linear hybrid systems. The condition of discrete time avoids certain mathematical problems, such as Zeno behaviour (Heymann *et al.*, 2002; Ames and Sastry, 2005), and allows to derive models that can be employed for analysis and to explore optimal/predictive control problems.

Mixed Logical Dynamical Systems

The *mixed logical dynamical* (MLD) modelling framework, introduced by Bemporad and Morari (1999), is one way (among others) of representing hybrid systems that can be described by interdependent physical laws, logical rules, and operating constraints. MLD models have recently been shown to be equivalent to representations of hybrid systems such as *linear complementarity* (LC) systems, *min-max-plus-scaling* (MMPS) systems and *piecewise affine* (PWA) systems (among others) under mild conditions (see Heemels *et al.*, 2001). MLD forms are described by linear dynamical equations that are subject to linear mixed-integer inequalities, *i.e.*, inequalities involving both continuous and binary (or logical, or 0–1) variables. These include physical/discrete states, continuous/integer inputs, and continuous/binary auxiliary variables. The ability to include constraints, constraint prioritisation, and heuristics is a powerful feature of the MLD modelling framework.

The general MLD form can be written as (Bemporad and Morari, 1999)

$$x_{k+1} = Ax_k + B_1 u_k + B_2 \delta_k + B_3 z_k, \tag{2.8a}$$

$$y_k = Cx_k + D_1 u_k + D_2 \delta_k + D_3 z_k, \tag{2.8b}$$

$$E_2 \delta_k + E_3 z_k \leq E_1 u_k + E_4 x_k + E_5, \tag{2.8c}$$

where:

- The x variables are the continuous and binary states

$$x = \begin{bmatrix} x_c \\ x_\ell \end{bmatrix}, \quad x_c \in \mathbb{X} \subseteq \mathbb{R}^{n_c}, \quad x_\ell \in \{0,1\}^{n_\ell} \tag{2.9}$$

- The y variables are the continuous and binary outputs

$$y = \begin{bmatrix} y_c \\ y_\ell \end{bmatrix}, \quad y_c \in \mathbb{Y} \subseteq \mathbb{R}^{p_c}, \quad y_\ell \in \{0,1\}^{p_\ell} \tag{2.10}$$

- The u variables are the continuous and binary inputs

$$u = \begin{bmatrix} u_c \\ u_\ell \end{bmatrix}, \quad u_c \in \mathbb{U} \subseteq \mathbb{R}^{m_c}, \quad u_\ell \in \{0,1\}^{m_\ell} \tag{2.11}$$

- The δ variables are auxiliary Boolean variables with $\delta \in \{0,1\}^{r_\ell}$
- The z variables are auxiliary continuous variables with $z \in \mathbb{R}^{r_c}$.

Note that, by removing (2.8c) and setting δ and z to zero, (2.8a) and (2.8b) reduce to an unconstrained linear discrete-time system in state space. The variables δ and z are introduced when translating logic propositions into linear inequalities. All constraints are collected in the inequality (2.8c).

The transformation of certain hybrid system descriptions into the MLD form requires the application of a given set of rules. To avoid the tedious procedure of deriving the MLD form by hand, a compiler called HYSDEL (HYbrid System DEscription Language) was developed in Torrisi and Bemporad (2004) to generate the matrices A, B_i, C, D_i, and E_i in (2.8).

2.2.3 MPC Problem and Hybrid Systems

Different methods for the analysis and design of hybrid systems have been proposed in the literature over the last few years (see, among many others, Bemporad and Morari, 1999; Lygeros *et al.*, 1999; Branicky and Zhang, 2000; Borrelli, 2003; Teel, 2007; Lunze and Lamnabhi-Lagarrigue, 2009). The implementations of these methods directly depend on the hybrid system representation used. One of the most well-studied methods involves the class of optimal controllers, which may use the MLD form in order to compute the corresponding control law according to the system's performance objectives.

The formulation of the optimisation problem in the hybrid MPC (HMPC) framework follows the approach of the standard MPC design (see Maciejowski, 2002). The desired performance indices are expressed as affine functions of the control variables, initial states, and predicted disturbances. However, due to the presence of logical variables, the resulting optimisation problem is a *mixed-integer quadratic* or *linear programming* problem (MIQP or MILP, respectively). The computed control law is referred to as *mixed-integer predictive control* (MIPC).

In general, the structure of the MIPC is defined by the OOP (2.5), with the part related to the switching dynamics added. Hence, the new OOP that considers the hybrid system framework is presented as follows. Assume that the system output should track a reference signal y_r, and that x_r, u_r and z_r are desired references for the states, inputs and auxiliary variables, respectively. For a fixed prediction horizon $H_p \in \mathbb{Z}_{\geq 1}$, the input sequence (2.2) is applied to the system (2.8), resulting in sequence (2.4) and the sequences

$$\Delta_k(x_k, \mathbf{u}_k) \triangleq \left(\delta_{0|k}, \delta_{1|k}, \dots, \delta_{H_p-1|k}\right) \in \{0,1\}^{r_\ell \times H_p}, \tag{2.12}$$

$$\mathbf{z}_k(x_k, \mathbf{u}_k) \triangleq \left(z_{0|k}, z_{1|k}, \dots, zH_p - 1|k\right) \in \mathbb{R}^{r_c \times H_p}, \tag{2.13}$$

under the same conditions as in (2.5).

Hence, the OOP for hybrid systems in MLD form is now defined as

$$\begin{aligned}\min_{\{\mathbf{u}_k \in \mathscr{U}_{H_p}\}, \Delta_k, \mathbf{z}_k} \; & J\left(\mathbf{u}_k(x_k), \Delta_k, \mathbf{z}_k, x_k\right) \triangleq \left\| Q_{x_f}\left(x_{H_p|k} - x_f\right)\right\|_p \\ & + \sum_{i=1}^{H_p-1} \left\| Q_x\left(x_{k+i|k} - x_r\right)\right\|_p + \sum_{i=0}^{H_p-1} \left\| Q_u\left(u_{k+i|k} - u_r\right)\right\|_p \\ & + \sum_{i=0}^{H_p-1} \left(\left\| Q_z\left(z_{k+i|k} - z_r\right)\right\|_p + \left\| Q_y\left(y_{k+i|k} - y_r\right)\right\|_p\right),\end{aligned} \tag{2.14a}$$

subject to

$$x_{k+i+1|k} = Ax_{k+i|k} + B_1 u_{k+i|k} + B_2 \delta_{k+i|k} + B_3 z_{k+i|k}, \tag{2.14b}$$

$$y_{k+i|k} = Cx_{k+i|k} + D_1 u_{k+i|k} + D_2 \delta_{k+i|k} + D_3 z_{k+i|k}, \tag{2.14c}$$

$$E_2 \delta_{k+i|k} + E_3 z_{k+i|k} \leq E_1 u_{k+i|k} + E_4 x_{k+i|k} + E_5, \tag{2.14d}$$

$$x_f = x_{r H_p|k}, \tag{2.14e}$$

for $i = 0, \dots, H_p - 1$, where x_f corresponds to the desired value of the state variable at the end of the prediction horizon, and p is related to the selected norm (1-norm, quadratic norm or infinity norm). Q_{x_f}, Q_x, Q_u, Q_δ, Q_z and Q_y are the weight matrices of suitable dimensions that fulfil the following conditions:

$$\begin{aligned}& Q_{x_f,x,u} = Q_{x_f,x,u}^T \succ 0, \quad Q_{\delta,z,y} = Q_{\delta,z,y}^T \succeq 0 \qquad (p = 2), \\ & Q_{x_f}, Q_x, Q_u, Q_\delta, Q_z, Q_y \quad \text{nonsingular} \qquad (p = 1, p = \infty).\end{aligned} \tag{2.15}$$

Assuming that MIPC problem (2.14) is feasible for $x \in \mathbb{X}$, there is an optimal solution that is given by the sequence

$$\left(u_{0|k}^*, u_{1|k}^*, \cdots, u_{H_p-1|k}^*, \delta_{0|k}^*, \delta_{1_k}^*, \cdots, \delta_{H_p-1|k}^*, z_{0|k}^*, z_{1|k}^*, \cdots, z_{H_p-1|k}^*\right), \tag{2.16}$$

which, upon applying the receding horizon strategy, yields the MPC law in (2.7). Notice that the described procedure corresponds to the extension of the MPC formulation in Section 2.2.1 to hybrid systems, but that the solution is obtained by solving the OOP (2.14).

Some theoretical aspects of the control of hybrid systems have been significant research topics during the last few years. For instance, notice that H_p should be finite. Infinite horizon formulations are not pragmatic theoretically or in practical implementations. The statement that H_p is as large as possible implies a high number of logical variables in the MIPC problem, which makes a computational treatment almost impossible (Bemporad and Morari, 1999). Thus, the assumption that H_p tends to infinity is even worse in the case of large-scale systems. On the other hand, the constraint $x_f = x_{rH_p|k}$, related to the final state in (2.14), can be relaxed as $x_{H_p|k} \in \mathbb{X}_T \subseteq \mathbb{X}$, where $\mathbb{X}_T$ is defined as the *target state set* (Lazar *et al.*, 2006). According to this assumption, the sequence $\mathscr{U}_{H_p}(x_k)$ in (2.3) is redefined with respect to $\mathbb{X}_T$ as

$$\mathscr{U}_{H_p}(x_k) \triangleq \{\mathbf{u}_k \in \mathbb{U}^{H_p} \,|\, x_k \in \mathbb{X}^{H_p}, x_{H_p|k} \in \mathbb{X}_T\}. \tag{2.17}$$

All of the concepts, formulations and definitions presented so far are used in the following chapters to present MPC formulations for linear and hybrid systems. Chapters 4 and 8 consider the definition of a OOP where the model is purely linear, while Chapters 5, 6 and 7 consider the OOP for a hybrid system.

2.3 Fault-tolerance Mechanisms

The aim of RTC of sewer networks is to improve their dynamical performance during extreme meteorological conditions. Under these conditions, it is very likely that a fault will occur in a constitutive element of the network, which will result in a reduction in control effectiveness, degrading system performance and even causing dangerous situations such as severe flooding or pollution. Therefore, it is very important to incorporate fault tolerance mechanisms that reduce the effects of such faults, thus ensuring that the control objectives are at least partially fulfilled.

Fault-tolerant control (FTC) is a relatively new idea, introduced in the research literature (Patton, 1997), which allows the creation of a control loop that fulfils its objectives (albeit possibly with a degree of degradation) when faults occur in components of the system (instrumentation, actuators and/or a plant). A control loop can be considered fault tolerant if:

- Adaptation strategies for the control law are included in the closed-loop scheme, and/or
- Mechanisms that introduce redundancy in sensors and/or actuators are incorporated.

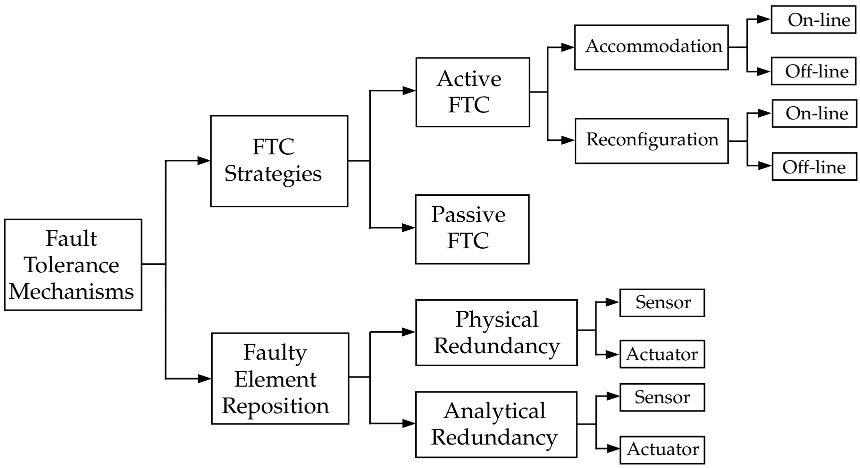

Figure 2.8 Taxonomy of fault-tolerance mechanisms

Figure 2.8 proposes a classification of the fault-tolerance mechanisms considered in this section.

2.3.1 Fault Tolerance by Adapting the Control Strategy

The literature considers two main groups of fault-tolerant control strategies: *active* and *passive* techniques. Passive techniques are control laws that account for the occurrence of a fault as a system perturbation. Thus, within certain margins, this type of control law has inherent fault tolerance capabilities that allow the system to cope with the presence of a fault. Complete descriptions of passive FTC techniques can be found in Liang *et al.* (2000), Qu *et al.* (2001), Liao *et al.* (2002), Qu *et al.* (2003), Niemann and Stoustrup (2005), Benosman and Lum (2008) and Steffen *et al.* (2009), among many others.

On the other hand, active FTC techniques adapt the control law based on the information from the FDI block. Using this information, some automatic adjustments are made in order to attempt to achieve the given control objectives.

The scheme of Figure 2.9 proposes a particular architecture of an active FTC loop introduced by Blanke (1999), which contains three design levels: the *control loop* (level 1), the *fault diagnosis and isolation* (FDI) system (level 2) and the *supervisor system* (level 3), which closes the outer loop and adds the fault-tolerance capabilities.

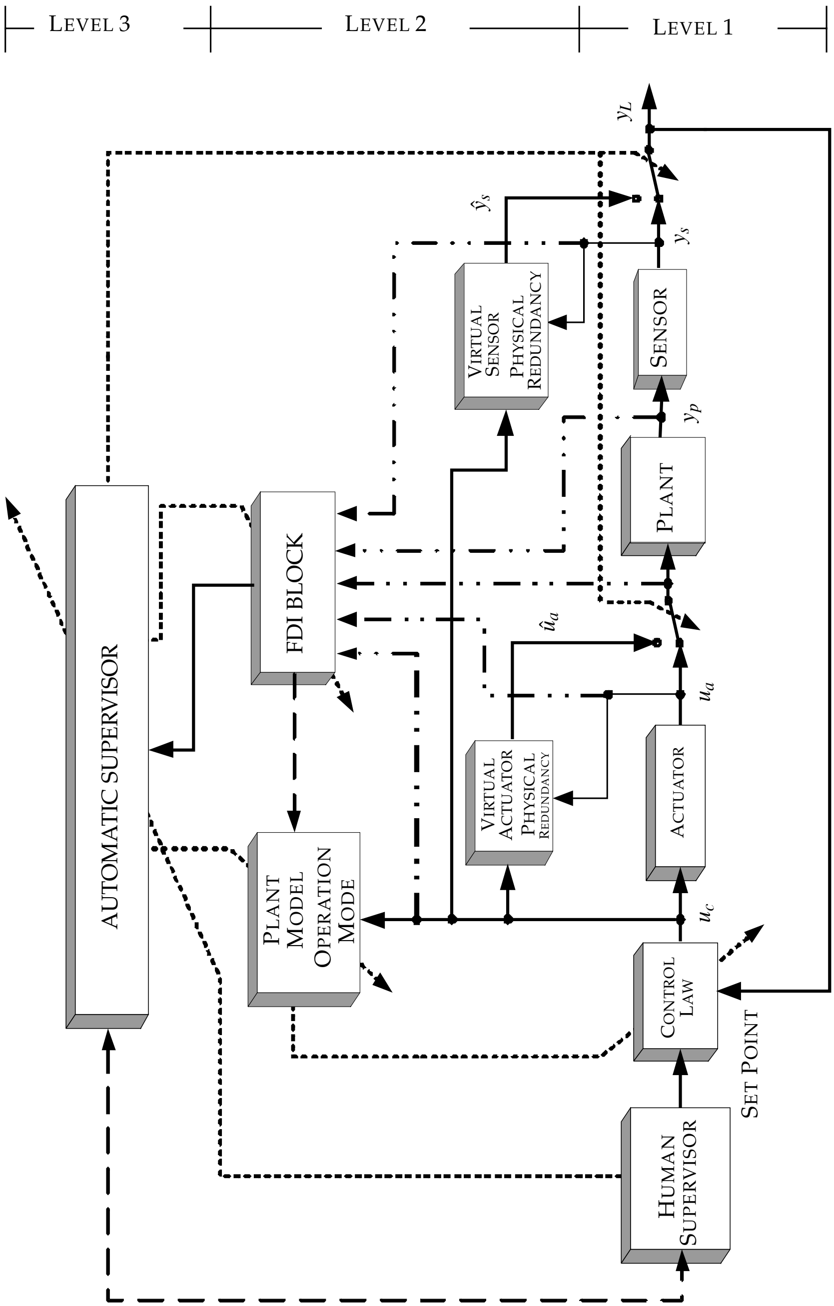

Figure 2.9 Proposed architecture of an FTC system

The feedback control loop shown in Figure 2.9 consists of a *control law*, an *actuator* or a *set of actuators*, the *plant*, and a *sensor* or an *set/array of sensors*. In parallel with the sensor and the actuator blocks, other hardware or software blocks are used to provide *redundancy* during signal measurement and the application of control actions. This redundancy can be introduced in physical form (as redundant sensors or actuators) or in analytical form (by employing mathematical models). Using the input and output signals of sensors, actuators and the plant, the FDI system detects and isolates faults, quantifies the magnitude of the fault, and identifies the faulty elements (if possible). Next, the FDI system sends this information to the *automatic supervisor* (AS), which takes the decisions needed to keep the control loop operative despite the occurrence of the fault.

Note that the AS block is a discrete-event system, while the supervised system is defined in continuous time. Information exchange between these systems is performed through the FDI block. Since the whole system is mixed in nature, its corresponding analysis and design can be done using *hybrid systems theory* (see, among many others, Cassandras *et al.*, 1995; Bemporad and Morari, 1999; Attouche *et al.*, 2001; Morari *et al.*, 2003; Lunze and Lamnabhi-Lagarrigue, 2009), which is an area that is currently being explored and developed in the research literature.

Once the AS block receives the information from the FDI module, it evaluates the admissibility of the system performance, considering the occurrence of the fault. To do this, AS judges whether the control objectives:

- Are fulfilled, allowing for a certain degree of degradation (region of degraded performance), or
- Are not fulfilled, but it is still possible to activate corrective actions (region of unacceptable performance).

Otherwise, the process should be stopped (region of danger). Figure 2.10 shows the abovementioned regions of operation for a two-state system. Chapter 8 presents a methodology that evaluates the admissibility of a given fault configuration.

Accommodation and Reconfiguration Strategies

In order to understand the operation of the different strategies within the active FTC philosophy, the standard feedback control problem is defined by (see Blanke *et al.*, 2006)

$$\langle \mathscr{O}, \mathscr{C}(\theta), \mathscr{U} \rangle, \tag{2.18}$$

where $\mathscr{O}$ is the set of *control objectives*, $\mathscr{C}$ is the set of *system constraints*, θ is the vector of *system parameters* and $\mathscr{U}$ is the *control law*. Hence, the impact of the fault is considered relative to the problem expressed in (2.18), where $\mathscr{C}(\theta)$ indicates how the system constraints depend on the parameters that

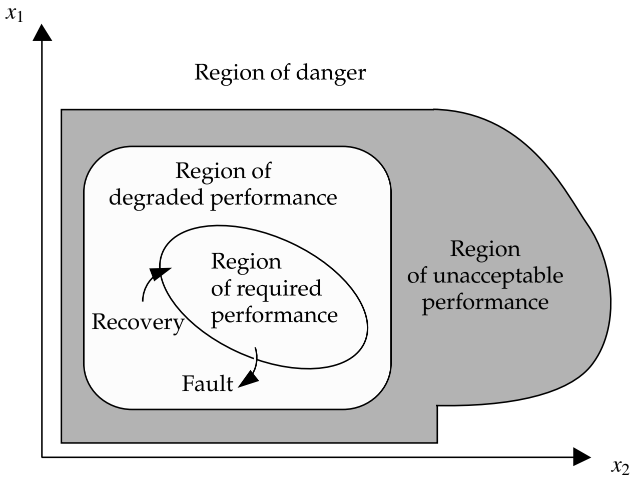

Figure 2.10 Regions of operation based on system performance. Taken from Blanke *et al.* (2006)

may by affected by the faults. The FDI block detects and isolates the fault with or without estimating its magnitude.

Depending on the information provided by the FDI module regarding the magnitude of the fault, two main strategies to adapt the control loop in order to introduce fault tolerance are possible. The first strategy consists of modifying the control law without changing the other elements within the control loop. This is known as *accommodating the system to the effect of the fault*, and it can be done in the case where all changes in system structure and parameters resulting from the fault can be accurately estimated. This concept is introduced formally in Definition 7.1.

The second strategy used to adapt the control loop is based on changing the control law and complementary elements of the closed loop as needed. This is known as *refiguring control due to the presence of a fault*, and it can be applied when no data on fault estimation is available. In this case, faulty components will be unplugged by the FDI block and an attempt will be made to achieve the control objectives using the remaining (fault-free) components. This concept is formally introduced in Definition 7.2.

On-line/Off-line Control Law Adaptation

Once the adaptation approach has been selected for the control law, there are two main ways of implementing this adaptation within the control loop. The basic difference between them is that a pre-computation of the control law parametrised with respect to the faults is performed off-line in one case (off-line adaptation), whereas the control law is recomputed on-line while taking the faults into account in the other case (on-line adaptation). These approaches are described in more depth below:

Off-line adaptation. Also known as adaptation using a pre-calculated controller. In this case, the control law can be written as $\mathscr{U}_f = U(f)$, where f corresponds to the determined fault. Thus, within the FTC architecture, there is a block that determines the mode of operation of the system once the fault occurs, which allows $\mathscr{U}_f$ to be computed (see Figure 2.11 (a)). One possible characterisation of the control laws in this framework according to the nature of the plant (the mathematical model) is given in Theilliol (2003) as follows:

- *Control laws for LTI models*: techniques based on LTI system models, such as *model matching* (Kung, 1992), *model following* (Jiang, 1994), *LQR* and *eigenstructure assigment* (Jiang, 1994; Zhang and Jiang, 2002), among others.
- *Control laws for a family of LTI models*: techniques based on LTI models obtained by linearisation around a set of equilibrium points covering a certain portion of the whole state space. Some examples are *multi-models*, *gain scheduling* and *LPV*, among others.
- *Control laws for nonlinear models*: techniques based on the nonlinear model of the system. In this case, *soft computing* techniques are usually employed to design the controller. Examples of these laws include *fuzzy control*, *neural control* and *neuro-fuzzy control*, among others (Diao and Passino, 2001).

On-line adaptation. Also known as adaptation using a controller computed on-line. In this case, the control law $\mathscr{U}$ is obtained on-line from an estimate of the actual system restrictions $\hat{\mathscr{C}}_f(\hat{\theta}_f)$ once the fault occurs. Figure 2.11 (b) shows the basic operational scheme for this case. There are two ways to estimate the effect of the fault on the system constraints:

- *Off-line estimation*: The effect of the fault on the system constraints is considered off-line. This allows the constraints to be expressed as a function of the fault, and the control law to be changed according to the fault information provided by the FDI module. In this way, the controller is always recomputed while taking into account the effect of the fault on the system constraints. Examples of control techniques from this group are *MPC* (Maciejowski and Ramirez, 1993; Maciejowski and Lemos, 2001) and *static feedback linearisation* (Zhang and Jiang, 2003, 2008).
- *On-line estimation*: The effect of the fault on the system constraints is computed on-line, so the controller changes on-line too. Examples of control techniques from this group are *adaptive control* (Ikeda and Shin, 1995; Diao and Passino, 2002), *dynamic feedback linearisation*

(Zhang and Jiang, 2003, 2008) and *dual predictive control* (Veres and Xia, 1998).

2.3.2 *Fault Tolerance by Repositioning Sensors and/or Actuators*

Serious faults in sensors or actuators break the control loop. In order to keep the system operating, some degree of redundancy should be present so that a new set of sensors (plant inputs) or actuators (plant outputs) can be used. To do this, an accommodation block is implemented to work together with the plant and the other fault-free elements. The main objective is to get a closed loop with almost the same performance as the fault-free closed loop, and thus to attempt to maintain the desired control objectives.

The required redundancy for sensor/actuator fault tolerance can be achieved using either physical redundancy (also called *hardware redundancy*) or analytical redundancy (also known as *software redundancy* or *redundancy by virtual element*).

Fault Tolerance in Sensors

In the case of sensors, the physical redundancy is achieved by having an odd number of measuring elements, and the outputs from these elements are multiplexed in a decision block. Such a block identifies the correct measurement by determining the most common signal value from among the multiplexed signals. On the other hand, the tolerance mechanism utilised with analytical redundancy is to employ an observer, which enables the system measurements to be reconstructed from other existing sensors. Therefore, this technique is also known as a *virtual or software sensor*. The design of a sensor network that takes fault tolerance, system observability, costs and robustness into account is currently an important research subject in the literature (see, *e.g.*, Hoblos *et al.*, 2000; Attouche *et al.*, 2001; Khanna *et al.*, 2004; Kuhn *et al.*, 2006; Michaelides and Panayiotou, 2009). In Staroswiecki *et al.* (2004), sensor network design for fault tolerance estimation is proposed. In this work, aspects like the reliability of a set of sensors, their fault tolerance capabilities and the minimum number of redundant sensors are evaluated. Applications of these mechanisms can be found in aeronautics (Lyshevski *et al.*, 1999; Huo *et al.*, 2001), in AC systems (Bennett *et al.*, 1999), and in wireless network set-ups (Cardei *et al.*, 2007; Chao and Chang, 2008; Alwan and Agarwal, 2009), among many other fields.

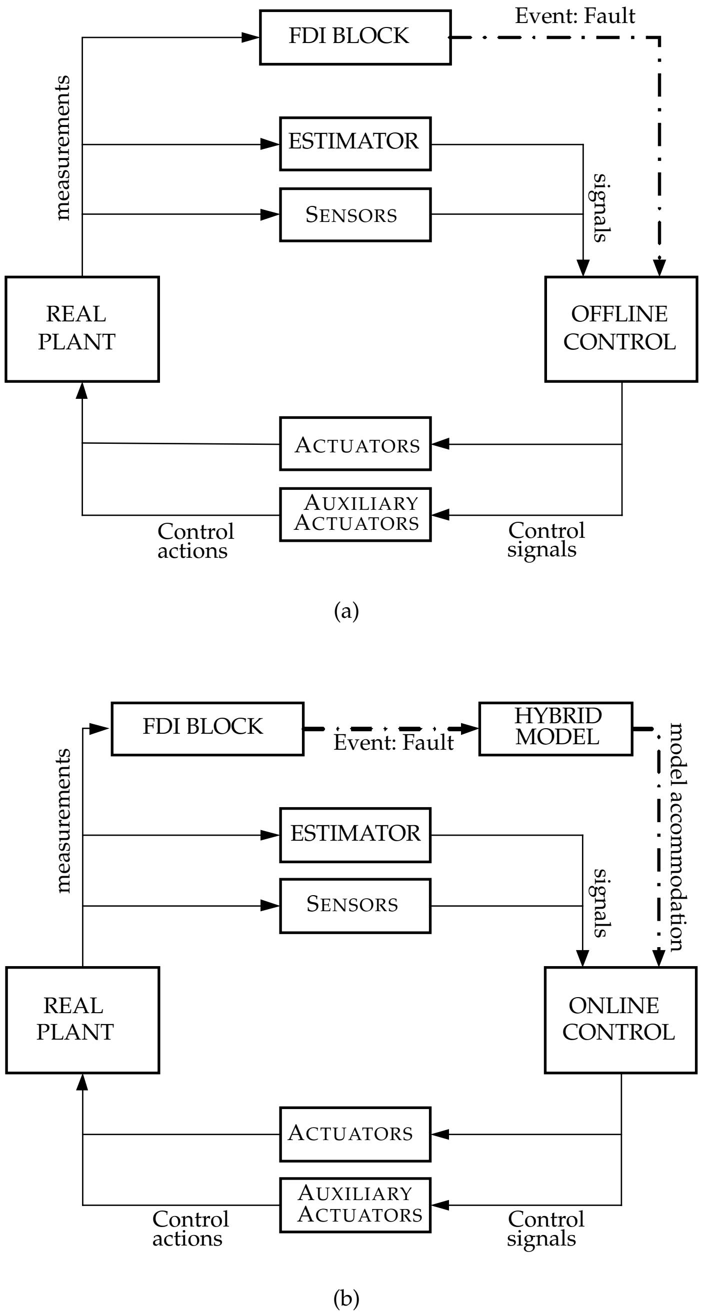

Figure 2.11 Conceptual schemes for FTC law adaptation: (a) off-line, and (b) on-line

Fault Tolerance in Actuators

Just as for sensors, physical redundancy of actuators means using additional units that can be multiplexed in a decision block by unplugging the faulty actuator and connecting an alternative one with no fault.

On the other hand, in the case of an overactuated system, some degree of physical redundancy already exists. This fact allows to adapt the control law (using either an accommodation or reconfiguration strategy) to find suitable control actions for fault-free actuators. In this way, the control objectives can be fulfilled with an acceptable level of degradation (Dardinier-Maron *et al.*, 1999). Thus, the need to incorporate new hardware into the closed loop is avoided, which makes it cheaper to implement. For instance, in the case of a large-scale water system, where there are thousands of actuators, this approach is suitable for achieving actuator fault tolerance (see Chapters 7 and 8).

The theoretical use of analytical redundancy to achieve actuator fault tolerance was recently proposed. An actuator counterpart to a virtual sensor, known as a *virtual actuator*, was reported in Lunze and Steffen (2003). This proposal was developed further in Lunze and Richter (2006) and Richter *et al.* (2007), while it was applied in Gawthrop and Ballance (2005) and Steffen (2005).

2.4 Summary

This chapter has presented the main aspects of each of the topics treated in this book. The first section encompassed definitions, concepts and discussions taken from the literature regarding sewer networks and their constitutive elements. Moreover, a brief outline of the state of the art in the RTC of networked systems was provided. In the second section, the MPC strategy, hybrid systems, the MPC formulation, and the RTC strategy for sewage systems was presented and discussed.

Finally, the third section collected together the main ideas about existing fault-tolerance mechanisms. In Chapter 3, the sewer network elements presented in Section 2.1 are described using mathematical modelling principles in order to obtain a model of the case study considered in this book. MPC formulations and concepts for linear systems are applied in Chapter 4, while those for hybrid systems are applied in Chapters 5, 6 and 7. Additionally, in Chapter 7 and Chapter 8, the descriptions and definitions relating to fault tolerance and FTC introduced in Section 2.3 are considered and their application discussed.

http://www.springer.com/978-1-84996-352-7

Model Predictive Control of Wastewater Systems
Ocampo-Martinez, C.
2010, XXX, 217 p., Hardcover
ISBN: 978-1-84996-352-7