A Framework for Studying Complex Industrial Systems: An Example Based on the UMTS Infrastructure.

Simon Bliudze
Supervisor: Daniel Krob

June 22, 2006

Contents

Chapter 1

Introduction

> Fabriquées à partir du langage, les machines sont cette fabrication en acte ; elle sont leur propre naissance répétée en elles-mêmes ; entre leur tubes, leurs roues dentées, leur systèmes de métal, l'écheveau de leurs fils, elles emboîtent le procédé dans lequel elles sont emboîtées.
>
> Michel Foucault, *Raymond Roussel*

Initially, when I have started working on my thesis, its was supposed to be centred around the performance analysis in mobile communication networks. Accordingly, I have carried out — using classical approaches such as simulation or combinatorial analysis — several more or less independent studies in this area, which are presented here as Chapters 4 through 6. Two of these (corresponding to Chapters 4 and 5) were conducted in collaboration with the UMTS Architecture team at Alcatel CIT, whereas the third one, constituting Chapter 6 of this thesis and generalising some previous studies by Dornstetter, Krob, Thibon, and Vassilieva was conducted at the Laboratory for Computer Science of the École Polytechnique (LIX).

While working on these studies we have realised that they represent different levels of abstraction for the Quality of Service analysis of UMTS, each level relying with a different degree of explicitness on the lower one(s). This observation illustrated rather well one of the characteristics of Complex Industrial Systems — the subject of the project started in autumn 2004 —, namely the fact that they are decomposed recursively into a hierarchical structure of subsystems. We have decided, therefore, to use these three studies to illustrate the notion of system that we introduced for the latter project. This was, moreover, motivated by the fact that the definition of the system is partially inspired by our work on mobile communications: some examples such as sampler and modulator come directly from digital signal processing, and the idea of working with streams of data at different time scales (frequencies) is very well illustrated by a typical coding chain.

Altogether, there is a kind of "retroaction loop" between the notion of systems, which was influenced considerably by these telecommunications studies, and the presentation of the latter, which we adapt to better illustrate the systemic approach. Moreover, in Chapter 4, for example, the systemic treatment of power control allows us to better underline the similarities between double loop algorithms, and the couple outer/inner loop power control.

As a consequence of the above decision, the present thesis comprises essentially two more or less self-contained, although not independent, parts: first we present the systems as defined in the framework of the Complex Industrial Systems project, and then we illustrate some aspects of these with the three studies mentioned above.

We have abstained, therefore, from the traditional in such cases presentation, where the manuscript would be split into Part 1 and Part 2 correspondingly, in favour of a sequential presentation in order to better reflect the idea of descending through different levels of hierarchical decomposition of a given system. We start from the global definition of a system, then we present a particular one — the UMTS —, and descend subsequently to the bit level analysis through a subsystem level (power control) and frame level (hybrid ARQ).

1.1 Complex industrial systems

In the modern world, complex industrial systems are just everywhere even if they are so familiar to us that we usually forget their underlying technological complexity. Transportation systems (such as aeroplanes, cars or trains), industrial equipment (such as micro-electronic or communication systems) and information systems (such as commercial, production, financial or logistics systems) are good examples of complex industrial systems that we are using or dealing with in the everyday life.

"Complex" refers here of course first to the fact that the design and the engineering of these industrial systems are incredibly complex technical and managerial operations. Thousands of specialised engineers, dozens of different scientific domains and hundreds of millions of euro can indeed be involved in the construction of such systems. For instance, in the automobile industry, a new car project typically lasts 4 years, requires a total working effort of more than 1 500 man-years, involves around 50 different technical fields and costs from 800 up to 1 500 millions of euro! In the context of software systems, important projects have also the same kind of complexity. Recently, unification of the information systems during a merger of two important French financial companies, has required 6 months of preliminary studies followed by 2 years of work for a team of 1 000 computer engineers, in order to integrate more than 250 different business applications, leading to a total cost of around 500 million euro.

As one may imagine, such projects are extremely difficult to manage due to the fact that the underlying systems are much too complex to be totally understood in their whole by a single person. It is in this context that we speak of complex industrial systems. Although, at this point, this notion is clearly not very well defined and rather subjective, it corresponds, nevertheless, to a strong industrial reality.

Complex industrial systems are, indeed, characterised both by the intrinsic difficulty of their design and by the large number of subsystems and technologies they involve, in such a way that the global resulting system can not be anymore apprehended in all its details by one human being. One should not in particular mix up complex systems with complicated systems, the latter referring to industrial systems that are difficult to design and to construct, but that can still be completely technically understood by some brilliant engineer.

To face this complexity, engineers developed a number of methodological tools, popularised in the industry under the name of *system engineering* (see [68, 69] for general systems or [83, 86] for software systems) that fundamentally rely on one of the oldest and most popular paradigms in human history, *divide and conquer*, which translates here into the assertion that complex industrial systems can always be recursively decomposed in a series of coupled subsystems,

up to arriving to totally elementary systems that can be completely handled. [1] In such a framework, system engineering provides the techniques for assisting all stages of the analysis and development process: architecture design, progressive integration, and final validation and qualification that altogether determine the realisation of an industrial complex system.

Despite this strong methodological environment, there is still a huge lack of theoretical tools that may help engineers to face such complexity in practice. In particular, one does not find a lot of research works that study "heterogeneous" systems (i.e. complex industrial systems that result from the integration of several "homogeneous" subsystems) directly *in their whole*, though a rather important research effort has been made during the last decade to better understand several important families of homogeneous systems (such as embedded systems, software systems, etc.) that appear as typical subsystems involved within larger industrial systems.

Also, due to the fact that main categories of homogeneous industrial systems can be handled by a large variety of models and tools, there are so far neither unified models, nor unified tools that can be used to deal with complex industrial systems in all their generality. In the same way, there are no unified tools or methods for managing all the aspects of the implementation cycle of an industrial complex system (that is to say, the cycle of development of a system going from the analysis of needs and the specification phase up to the final IVVQ [2] processes).

The first stage of our project is, therefore, to reconnect all these (more or less disconnected) streams by going back to the very fundamentals, that is to say by looking for a *unified definition of an industrial system* from which all these different models could be deduced. Observe that such an approach is clearly in rupture with the usual one, which is rather oriented on local fixing of connection problems existing between the different tools that are used for designing and managing an industrial system (by transforming them into interface design questions). We think, however, that the key problem is much deeper and comes directly from the fact that there does not really exist any mathematically consistent global point of view on industrial systems (even if some interesting approaches are to be noticed — see for example [19, 85, 102]).

1.1.1 Complex industrial systems in practice

As already mentioned above, complex industrial systems are characterised by the fact that they integrate a big number of heterogeneous components. One can in particular distinguish three main categories of such homogeneous (sub-)systems that are listed below.

1. *Physical systems:* these types of systems are transforming *physical parameters*. The corresponding formal models are based on *continuous* (transfer) functions that are modelling the behaviour of such systems by means of partial derivative equations. The physical hearts of transportation systems, micro-electronics systems, telecommunication systems, etc. are for instance typical physical systems.

2. *Software systems:* these systems are characterised by the fact that they are only transforming and managing *data*. The associated formal frameworks are therefore based on *discrete functions* dealing with discrete inputs and outputs. Databases, Web oriented applications, Enterprise Resource Planning software (ERP), billing systems, etc. are again typical examples of software systems.

[1] This property can be used to construct a formal recursive definition of complex industrial systems.
[2] Integration, Verification, Validation and Qualification.

3. *Human systems :* human organisations can also be seen as systems as soon as their internal processes have reached a certain degree of normalisation. A typical example of such a system is the so-called *workflow management*, i.e. the process of managing different tasks performed by human employees as part of the operation of a given enterprise. These processes can indeed be seen as transfer functions that characterise this new type of systems. We cannot, in particular, avoid taking in consideration this non-technical type of system in the modelling of a global system as soon as the underlying human organisations are strongly interacting with its physical and/or software components.

   It is important to mention here that such a system does not represent any society as a whole, but rather a human team working in a framework of a particular industrial project. In such an organisation every single person would have a precisely defined role and a number of functions corresponding to this role, allowing to define eventually the global transfer function describing the whole organisation. However, these personal roles and functions would typically involve some kind of *decision making* and, therefore, imply a certain degree of randomness. Thus, a theoretical model capable of describing human systems should also allow random operation. For this reason, in this thesis, we do not consider this last type of systems, and only cite them here as their modelling constitutes a possible future research direction.

Observe that the main categories of inter-system couplings — that correspond to the possible interactions between different types of homogeneous systems — are immediately emerging from this last typology. On one side, one can indeed study the systems resulting from the coupling between physical and software systems, which are also called *hybrid systems* in the literature (see, for example, [9, 43, 53, 73, 101] for different point of views on such systems). On the other side, there is the problematic of *human-system interfaces*[3] that recovers the coupling of technical — that is to say physical or software — systems with human systems in the very specific meaning we adopt for this terminology.

1.1.2 Systems: a first formal definition

In a very fundamental way, a system can be seen as a transfer function $\mathcal{F}$ which is transforming — at each moment t of time — a vector $x \in I$ of *input parameters* into a vector $y \in O$ of *output parameters*. In this framework, all the entries of x (resp. of y) belong to a topological space, denoted here by I (resp. by O), which is called the *input* (resp. *output*) *space* of the system. In other words, the behaviour of a system is depicted by a classical transfer function model of the type:

$$y = \mathcal{F}(x; t)\,. \tag{1.1}$$

Of course, only the simplest *memoryless* systems can be described by an equation of this type. To include more complicated systems in this formalism, it has to be extended in the following way. First of all, a *state variable* has to be introduced. Let us reproduce here two examples from Severance [85] that illustrate rather well this situation.

Example 1.1 (Simple electrical circuit)
Consider the electrical resistive network shown in Figure 1.1, where the system is driven by

[3] Which is not connected with the classical human-machine interface (HMI) research trend, due to the fact that we are not interested here in the interaction of one person with a single machine, but clearly in the coupling of a whole organisation — analysed as a input/output system — with a physical and/or a software system considered also as a whole.

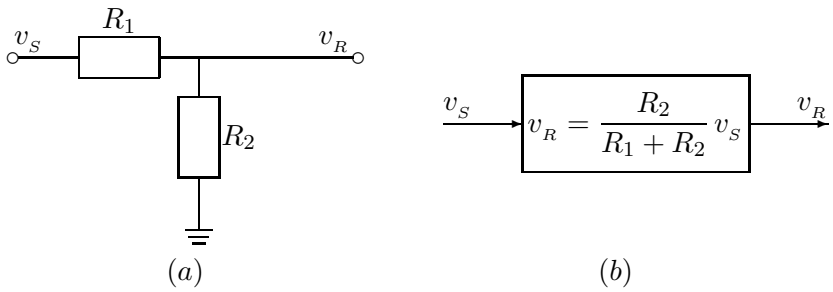


Figure 1.1: Electrical network (a) and its systemic representation (b).

an external voltage source $v_{_S}(t)$. The output is taken as the voltage $v_{_R}(t)$ across the second resistor R_2.

From the basic laws of electronics, it is clear that at any time t we have

$$v_{_R}(t) = \frac{R_2}{R_1 + R_2}\, v_{_S}(t)\,. \tag{1.2}$$

Thus the output of the system only depends on its current input, and therefore it can be fully described as in (1.1), without using any additional state variable. Moreover, one can observe also that this system is stationary (i.e. its action is independent of time t), which allows us, in particular, to drop the time parameter in Figure 1.1, and eventually in equation (1.2). ■

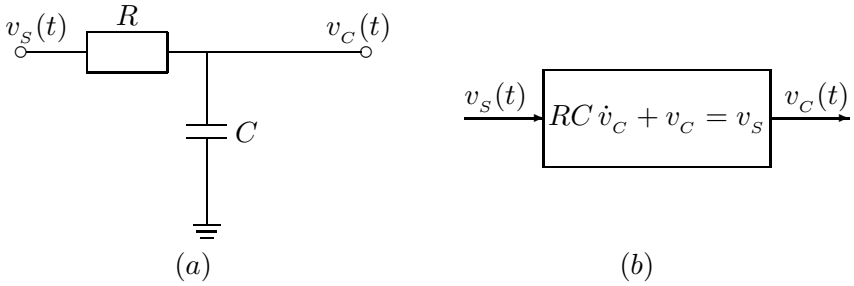


Figure 1.2: Electrical RC network (a) and its systemic representation (b).

Example 1.2 (Simple resistor-capacitor network)
Consider now the resistor-capacitor network shown in Figure 1.2. Since the capacitor is an energy storage element, this system is no longer stationary, and its output (the voltage across the capacitor) is described in terms of differential equations:

$$RC\,\dot{v}_{_C} + v_{_C} = v_{_S}\,.$$

This equation contains $v_{_C}$, which also implies that the system's operation depends on its own output, which is only possible in two situations: either the output is available to the system before it is produced,[4] or it has to be reintroduced as part of the system's input. We shall discuss the latter phenomenon further below, whereas the former consists simply in using a

[4] One can be easily convinced that there is no *causality* conflict in this particular example. More generally, the problem of causality is discussed in Section 2.2.1.

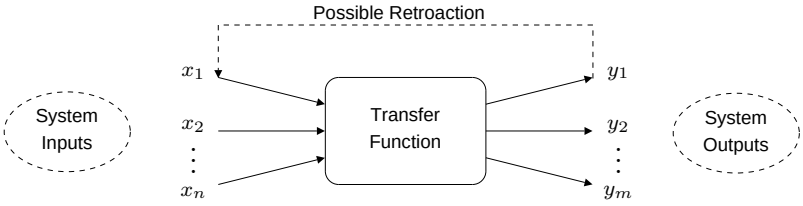


Figure 1.3: Functional representation of a system

state variable $q(t)$ to redefine the system in functional terms by setting

$$\begin{cases} \dot{q}(t) &= \frac{1}{RC}\Big(v_S(t) - q(t)\Big) \\ v_C(t) &= q(t) \end{cases} .$$

■

Observe that, rather than (1.1), the system of the last above example is more properly described by a functional relation of the form

$$(y; q) = \mathcal{F}(x; q, t)\,, \tag{1.3}$$

where q is the state variable, and, moreover, all three variables x, y, and q are considered to be functions of time t.

From the syntactic point of view, this situation is a special case of the *retroaction* or *feedback* phenomenon that we have also mentioned in Example 1.2. Indeed, when some output parameters can retroact on the inputs of a system (see Figure 1.3), the corresponding functional relation is defined by an equation of the following form, which can eventually be abbreviated to (1.3),

$$\begin{cases} (y_1, y_2, \ldots, y_m) &= \mathcal{F}(x_1, x_2, \ldots, x_n; t) \\ x_{i_1} &= y_{j_1} \\ &\vdots \\ x_{i_k} &= y_{j_k}\,, \end{cases} \tag{1.4}$$

where the last k equations define the feedback couplings.

From the semantic point of view, however, it is preferable to distinguish state variables from the input and output ones as we do it in equation (1.3). The reason for this is that state variables, as opposed to the input ones, should be perceived as internal to the system, and thus not subject to external influences. This becomes even clearer when one considers a system as a black box transforming the given inputs into corresponding outputs. Indeed, from this point of view the notion of state variable does not have any meaning, and the output of the system corresponds to a fixed point of the operator $\mathcal{F}$ in (1.4). [5]

1.1.3 Industrial systems: an architectural approach

Industrial systems are now characterised by the fact that they result from an integration process. These systems are, indeed, systems of systems that can be recursively decomposed into subsystems, up to arriving to elementary components that are simple enough to be handled

[5] This is only applicable when such a fixed point exists, which is, of course, the case of all realistic systems (as opposed, for example, to the system defined by the equation $x = x + 1$).

entirely. This decomposition — also called *system's architecture* — is, of course, not à priori determined by the system (i.e. the system's required functionality) and, moreover, in most cases it is not unique. Designing a good architecture for a given system constitutes one of the most important tasks in system engineering and can considerably influence the the rest of the development and analysis process. Figure 1.4 illustrates this situation on the example of the aeroplane transportation system considered at the global world level: this system can be, indeed, analysed as the interaction between several very different types of systems (i.e. physical systems such as aeroplanes, information systems such as reservation systems, software prevalent systems such as air traffic management systems, human systems such as airport organisations, etc.).

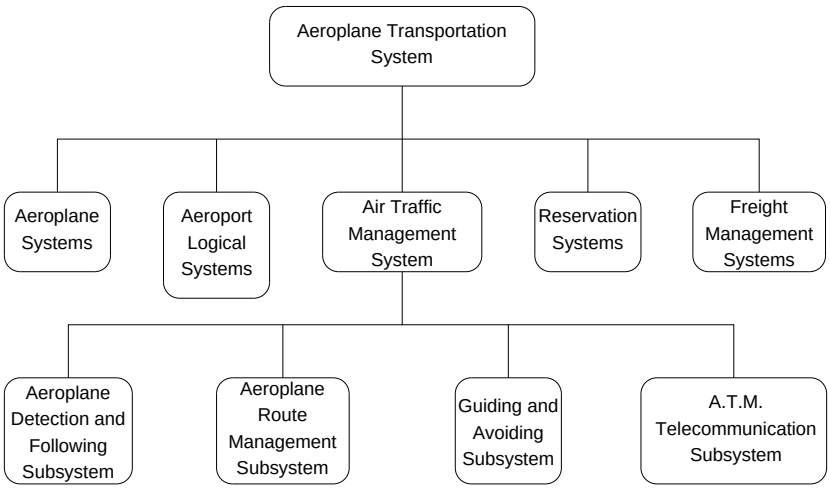


Figure 1.4: A hierarchy of complex systems.

This recursive representation of an industrial system is, of course, absolutely not independent of the process that leads to its construction. Recursively decomposing a system into its main subsystems allows to separate the different realisation tasks in a natural way, resulting therefore in a relatively rational industrial organisation for developing such industrial systems, Incidentally, this approach can influence the very understanding of the global behaviour of the resulting system.

In particular, the famous "V cycle" that corresponds to the development process used in practice for developing industrial systems can be seen as a direct consequence of the recursive modelling of a system that we introduced above. This "V cycle" can, indeed, be decomposed into three major steps that are all connected with this recursive approach as presented below.

1. *Step 1: Engineering.* This first development period is devoted to the *specification of the system*, that is to say to the construction of its recursive decomposition. This phase is then followed by the *technical design of the system* that can be analysed as an exploration from top to bottom of the associated recursive tree (i.e. from the root which represents the system in its whole, up to the leafs which correspond to the elementary components involved in the particular system) in order to technically design all the resulting components of a given system.

2. *Step 2: Realisation of the elementary subsystems.* When the engineering of a system is finished, one can begin to practically realise its different elementary parts (which correspond to the leafs of the tree associated with the system).

3. *Step 3: Integration.* The final step of the realisation of an industrial system is the integration step. It corresponds to a progressive assembling of the different pieces that form such an industrial system, followed by a recursive validation of the different resulting systems. Note that this integration process can also be analysed as an exploration from bottom to top of the associated recursive tree.

This typical "V development cycle" is illustrated in Figure 1.5.

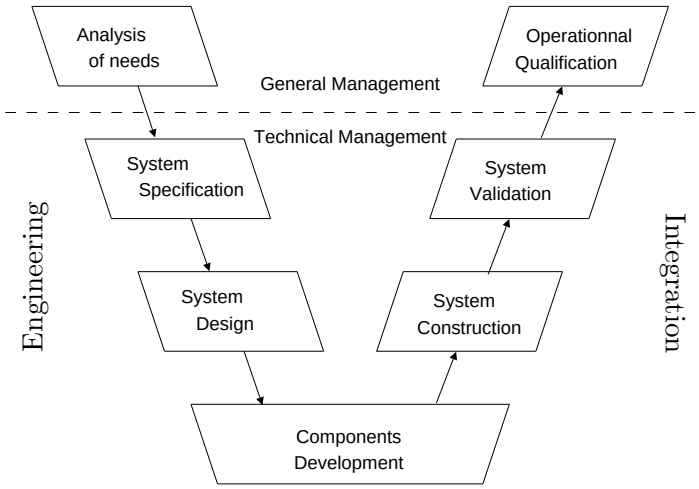


Figure 1.5: The development cycle of a system.

1.1.4 Complex industrial systems: a tentative definition

We are now in position to propose a definition for complex industrial systems, which is just unifying the two points of view that were presented in the last two subsections. A *complex industrial system* is indeed a recursively integrated system of (sub)-systems that can be modelled by a set of transfer functions coupled together. The global resulting system is then composed of a series of transfer functions whose inputs and outputs are mutually interconnected. Observe that global properties of such systems are typically rather difficult to study when the underlying subsystems are modelled by transfer functions of different mathematical nature [6] (see again Section 1.1.1).

In this definition, we are therefore mixing both a functional and an architectural point of view on industrial systems. This approach is also illustrated by Figure 1.6 which represents — in a very sketchy way — the transfer function associated with a "car" system. In this example, the behaviour of the resulting system — which is a mix between a purely physical system and an (embedded) software system — depend therefore on these physical and software subsystems and on an human system reduced here to a single driver, which reflects in the structure of its inputs (that are the sum of the inputs of these last two technical systems and of the actions of the driver). On the other hand, the outputs are here mainly of physical nature (wheel orientation, position and speed of the car, etc.) and they induce an important retroaction on the inputs (speed modification, attitude of the driver, etc.).

[6] Such as partial derivative equations and finite automata for example.

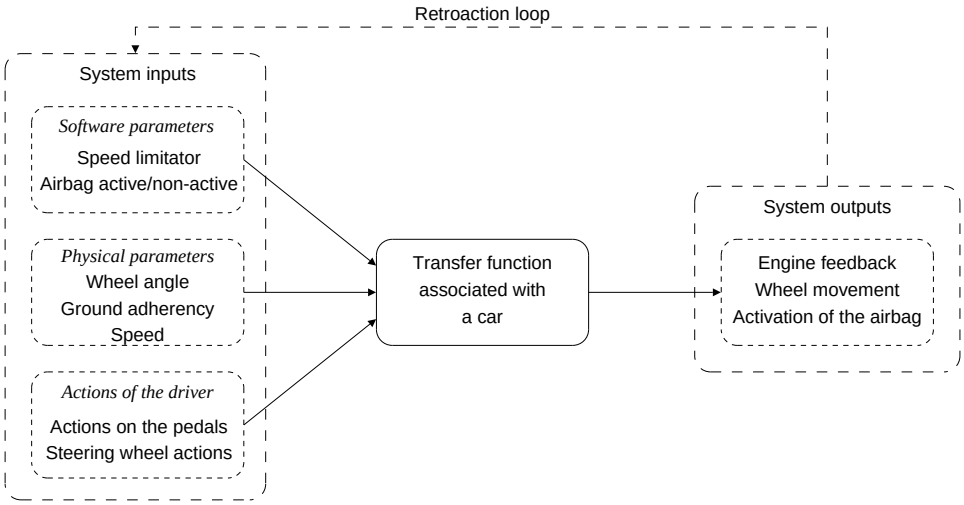


Figure 1.6: Simplified functional representation of a car system.

1.2 Universal Mobile Telecommunications System

1.2.1 Evolution of mobile communications

Electromagnetic waves were shown to be capable of transmitting information in the end of 19th century, not long after the invention in 1876 by Alexander Graham Bell (1847–1922) of wire telephone. The first radio communication is so far disputed by an Italian inventor Guglielmo Marconi (1874–1937) and a Russian physicist Alexander Stepanovitch Popov (1859–1906), both having performed a successful transmission in 1895. [7] However, further developments in wired and wireless communications did not advance at the same pace. While the total number of messages per year only for Bell company is cited to be 5,305,900,000 already in 1908 (see [18]), reports of a "first two way radio communication" can be found for the dates in the range between 1914 and 1929, and the first commercial mobile service (car phone) was not to be introduced before late 1940s in the USA and early 1950s in Europe.

The equipment in this systems was heavy and bulky, and the reception quality rather poor, especially considering the elevated price of the service. A number of technological advances around the end of 1970s such as, in particular, the invention of microprocessors, as well as the introduction of cellular systems allowed to render mobile communications accessible to a considerably wider group of users. Communication systems introduced during this period are termed 1G — for First Generation —, and they were still only capable of transmitting analog voice information.

The emergence of Second Generation (2G) cellular systems, in the beginning of 1990s, allowed to improve transmission quality, system capacity, and coverage by use of digital transmission technologies. A number of services other than speech transmission, such as short message service, fax and data transmission and roaming have appeared.

GSM,[8] the most widely spread 2G standard, was born, in 1991, as a result of a collaboration between France and Germany, which was at the origin, in 1984, of a public tender for a

[7] Marconi's patent has been overturned in 1943 in favour of Nikola Tesla (a Croatian-born American, 1856–1943), who has conducted similar experiments two years earlier, in 1893.

[8] GSM initially stood for "Groupe Spécial Mobile" (french). Later it was decided to keep the acronym while changing the name to "Global System for Mobile communications".

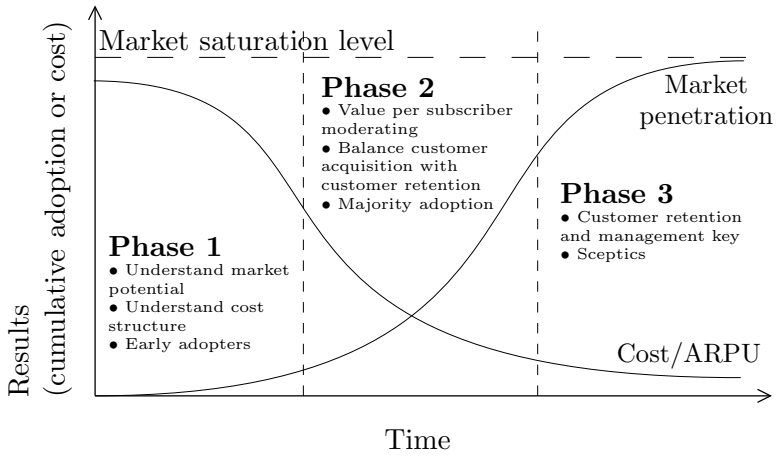


Figure 1.7: The S-curve — product cost and adoption evolution. (ARPU = Average Revenue Per User)

common mobile communications system. The original tender was conceived in terms of analog transmission techniques, and was subsequently modified to accommodate three proposals based on digital signal processing.

Second generation systems, being much more robust and accessible, provoked a sharp increase in popularity of wireless communications. In the end of 2005, the number of GSM subscribers on its own has passed the cap of 1.5 milliard users,[9] as opposed to some 20 million in 1990.

Similarly to the majority of consumer-oriented businesses governed by a diffusion process, the economic evolution of mobile communications follows, in the first approximation, the so-called *S-curve* (see for example [61, 67, 87, 95]). This curve, illustrated in Figure 1.7, represents the process of market penetration broken up in the following three phases.

Phase 1 (*Innovation*) At the initial stage of the product's introduction to the market, it is only adopted by the minority constituted by the tech-savvy users who provide the first evaluation of the products utility. At this stage, no practical data is available regarding the cost structure for the product, its market potential, and other similar characteristics such as, for example, the churn rate. Customer acquisition is primordial for the operator, as it both contributes to the product's credibility and, eventually, to the so-called *network effect*, which allows a transition to the second phase. During the innovation period, the cost of running the service is comparatively high, but so is the average revenue per user (ARPU), as customers are motivated by curiosity rather than by the price of the service.

Phase 2 (*Growth)* Once the customer base attains the critical mass, the product is adopted by the majority of the target population. This is normally accompanied by the reduction of both operating costs and ARPU, the total revenue being primarily determined by the number of customers. At this stage, more reliable statistics are available for various market parameters, which allows to better determine the pricing policies, and to balance the customer acquisition and retention.

[9] According to Wireless Intelligence [94], GSM accounted at that point for approximately 77% of world mobile communications market.

Phase 3 (*Maturity*) Finally, at the last stage of deployment, the marketing sector is nearly exhausted, and most of the target population has adopted the product. Customer acquisition, therefore, is no longer a priority as only a particularly sceptical minority is liable to adapt the product at this stage. Instead, customer retention and management becomes a key issue. At this stage, the service infrastructure is well established and the cost of running the service is at its lowest. On the other hand, this stage is characterised by the mounting pressure from the competitors.

Once the product deployment has reached the last phase above, one can isolate two particular problems:

- the saturated market provides steady cash flow for the service operator, but the equipment suppliers experience diminishing of revenue;
- in order to retain customers the operator has to maintain high value-for-money ratio as compared to that of the competitors, which means either lowering prices, or providing additional services.

One of the possible solutions is to restart the S-curve cycle by introducing new products, which might be either revolutionary or evolutionary. The former renders the previous product obsolete, and thus opens up a new market providing an opportunity to gain part of the competitors' customer base but undermines the operator's own customer base, whereas the latter evolves from the previous one and thus allows the customers to pass to the newer service with the same operator.

In the context of mobile communications, the Third Generation (3G) systems, and in particular the Universal Mobile Communications System (UMTS), are being introduced as this innovation. The new services are primarily oriented towards multi-media communications: private or business messaging services, video conferencing, video and audio streaming, gaming etc.; and location-based services: enable users to find other people, resources, or services; also enable others to find users, as well as enabling users to identify their location in the context of rescue operations or, for example, UMTS assisted GPS (see [17] for an in-depth morphological analysis of 3G systems).

The 3G systems are intended to develop into a single standard that would guarantee worldwide mobility and high data rates. The initial (Release 99) maximum achievable data rate for UMTS was 2 Mb. With the introduction of High Speed Downlink Packet Access (HSDPA) in Release 5 this has advanced to 10.8 Mbs and still growing...

At present, 3G systems are in the first phase of the S-curve. According to the UMTS Forum the worldwide total number of UMTS subscribers is 50 millions with 51% in Europe and 48% in Asia Pacific.[10] The market penetration is most important in such tech-savvy countries as Japan and Hong-Kong, while, in Europe, 3G systems are faced with an apparent lack of a "killer application" and with important competition from new emerging technologies such as, for instance, Wi-Fi or WiMax. Thus the transition of UMTS into the second phase of the adoption by mass market depends substantially on rapid development of high quality service base.

1.2.2 UMTS infrastructure: a systemic view

As with any communication system, the purpose of UMTS is simply to serve as a relay between its subscribers and service providers. Although this two communities often intersect as, for

[10] UMTS Forum Fast Facts data at 09 February 2006.

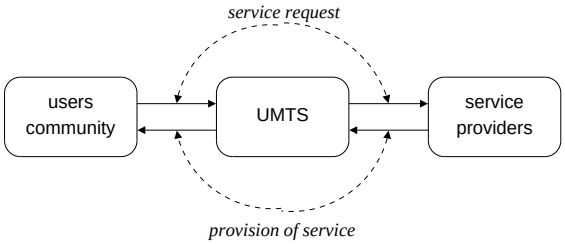


Figure 1.8: UMTS as a relay between subscribers and service providers.

example, in the case of the most fondamental speech service, the global situation can be very well illustrated as in Figure 1.8, i.e. UMTS as a system receives on input a set of demands for services from user community, transmits these to the domain of service providers, and then relays back the corresponding information.

At the first sight, this does not constitute a problem from the systemic point of view, as one can easily imagine a system that listens *conitnuously* on its incoming channels, and, whenever there is a service request from a user, transmits necessary information over its output channel, while again *continuously* listening on its second input channel for service providers' responses. We obtain, in this manner, a perfectly normal *physical* system as described in Section 1.1.1.

The problems start to appear when one examines the next level of decomposition of the UMTS infrastructure. Indeed, on this next level, UMTS can be divided in three domains communicating with each other:

1. *Core network:* comprises the network part inherited from second generation systems that connects the UMTS network to the existing ones such as the Internet and the Public Switched Telefone Network (PSTN);

2. *UMTS Terrestrial Radio Access Network (UTRAN):* contains the two new network elements introduced in UMTS, i.e. Radio Network Controller (RNC) and Node B, and provides a point of access to Core Network and consequently all the services over the radio channel;

3. *User Equipment (UE):* includes all the devices on the user side of the communication such as mobile phones, PC cards, credit card readers for rural areas without landline access, etc.

Indeed, in the diagram of Figure 1.8, does the UE, for example, belong to the UMTS or rather to the users community? On one hand, the specifications of the UE depend tightly on those of UMTS, and it is quite impossible to imagine the latter without the UE. On the other hand, if we consider user equipement to be a subsystem of UMTS, then it becomes difficult to establish a relation between different users and their respective equipment. Moreover, this would imply that the system modelling the UMTS has to change every time a mobile is switched on or off, or, even worse, moves from one cell to another!

The discussion above suggests that UMTS is probably better represented by a network of systems rather than by a single system that comprises the whole infrastructure (see Figure 1.9). In other words, one would consider UMTS as a graph, where each node (UE, Node B, etc.) is a system — both in generic and our particular meaning of the word — participating in the communication, and the edges correspond to available links.

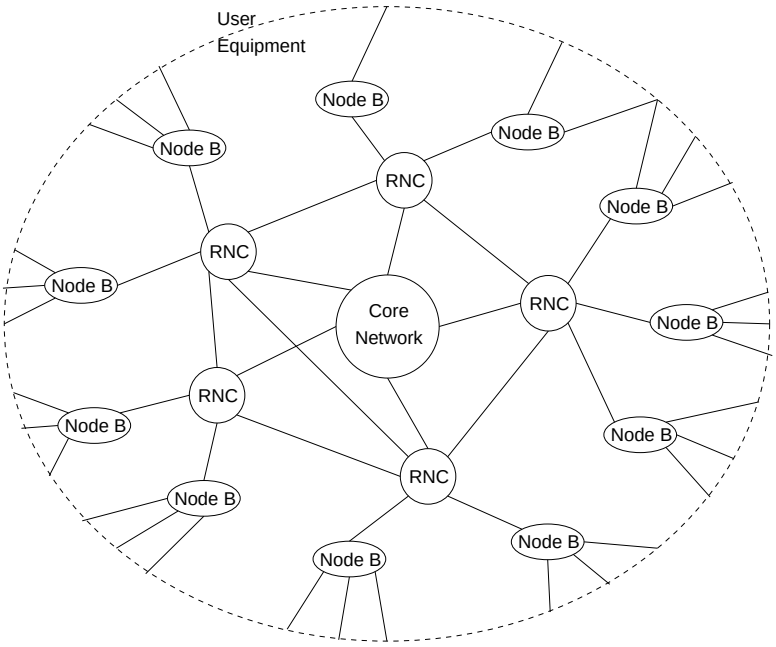


Figure 1.9: Network representation of UMTS.

Although, as we shall see in Chapter 3, we can still model this kind of systems (networks) of systems in our framework, it is clearly not very well adapted to this context. This allows us to give a sketch of definition for the target range of application of our model. Our framework is designed to model complex industrial systems with *finite description*, as opposed, for instance, to networks, where the full description of a system can *potentially* grow to infinity as new elements are added all along the system's life-time.

Nevertheless, the UMTS infrastructure allows us to develop a number of interesting examples for our model (see, for instance, Chapter 4), due exactly to the fact that it represents not a single system, but a collection of intercommunicating ones.

1.3 Structure of the thesis

As it has been mentioned above, this thesis is a product of the research performed as a part of the project that aims to define and develop the emerging field of complex industrial systems. Our goal, therefore, was first of all to propose an initial formal definition of such a system, and also to illustrate the context and the methodology that we would like this definition to reflect.

Consequently, the very structure of this thesis reflects the "V development cycle" that we have described above, and, more precisely, its descending branch, which consists of system specification (hierarchical decomposition) and analysis.

We start, in Chapter 2, by introducing a formal definition integrating at the same time the principle of recursive decomposition of a system and its functional (input/output) aspect. Also, in the same chapter, we provide some basic examples, which allow us to illustrate a large spectrum of systems that can eventually be modelled with the definition we introduced.

In the following chapters, we illustrate our approach on a case study of Universal Mobile Telecommunications System (UMTS). Clearly, a complete decomposition of UMTS is way out of scope of this thesis — in particular, due to the inherent complex nature of any system that

could serve as an example in our case![11] We proceed, therefore, using a sort of a "zoom-in" approach. We begin, in Chapter 3, by presenting a global overview of the UMTS infrastructure, providing also several high-level decompositions in major elements and domains of the network (e.g. UTRAN, Node B, etc.). In the subsequent chapters, we continue by selecting, for each chapter, a subsystem of lower level than that analysed in the previous one. Each time, we consider a particular problem characteristic of that level, and we show how this problem can be solved using the appropriate methods.

In Chapter 4, we select a number of subsystems, which have appeared in the high-level decomposition, to re-assemble them into a "virtual" subsystem representing one particular functionality of UMTS — the power control. More precisely, we discuss several algorithms for Uplink Outer Loop Power Control (UL OLPC), and we show how these algorithms can be parametrised in order to obtain the optimal performance.

We proceed then to the next — frame or block — level, by devoting Chapter 5 to a study of High Speed Downlink Packet Access (HSDPA) — a service that constitutes one of the latest additions to UMTS. We consider on this occasion one particular technique — Hybrid Automatic Repeat Request (H-ARQ) — utilised in HSDPA to improve the performance of decoding at the receiver by adapting the information transmitted. We present the optimal scheme to control the way this transmitted information is adapted. Contrary to Chapter 4, where the analysis is performed by means of stochastic methods, in Chapter 5, we obtain our results by simulation, thus illustrating the two approaches dominant in the industrial analysis of complex systems.

We conclude our descent through the levels of decomposition of the UMTS infrastructure by considering, in Chapter 6, a transmission of a single bit over a radio channel with spatial diversity, that is in presence of multiple paths (or trajectories) between the transmitting and receiving antennae. More precisely, we consider the probability that on reception this bit is not correctly demodulated, i.e. the Bit Error Rate (BER), which is a very important statistic in a telecommunications network. This time, we apply combinatorial methods to analyse the probability in question, which allows us to conclude on a major note by showing how an originally technical problem can give rise to interesting mathematical models.

[11] Here, we allow ourselves a luxury of a self-quote, by reminding that complex industrial systems that we are considering are, in particular, characterised by the fact that they "are much too complex to be totally understood in their whole by a single person".

Chapter 2

Global Approach: Functional Modelling of Complex Industrial Systems

> I don't own a watch or clock. I think of time in other totalities now. I think of my personal time-span set against the vast numerations, the time of the earth, the stars, the incoherent light-years, the age of the universe, etc.
>
> Don DeLillo, *Cosmopolis*
> (The Confessions of Benno Levin)

As we have already mentioned it in the previous chapter, a full panoply of technologies is used at present to model industrial systems. A direct consequence of this diversity of technologies is the equivalent diversity of tools implementing them. A majority of these tools are developed to perform some particular task, such as architectural design and analysis, verification or simulation, and therefore the underlying models are often incompatible. Thus designing a given system from scratch becomes extremely effort-consuming, as the results of the analysis by one of the available tools have to be converted manually to fit the requirements of another one, which is to be used at the next stage of the design process.

This diversity is not so much a result of the fundamental differences between the problems to be solved (systems to be modelled), but is rather a consequence of political decisions, and — what is even more important — the circumstances conditioning the evolution of the technologies in question. Indeed, most advanced contemporary modelling tools come as a result of a convergence process between different basic ones that were developed for some very specific purpose, such for example as designing electrical circuits, stress analysis or that of the hydro- or aerodynamic properties of an object. The most striking examples are probably coming from the systems involving some kind of control, where software subsystems have to be integrated with mechanical ones.

The question arises naturally: is it possible to unify these technologies by developing a common model? Considering the above argument, a model aspiring to realise such unification has to be defined on a very low level. Indeed, it suffices to consider, for example, hybrid systems, which constitute a subject of a relatively young and ever growing in popularity research field, to realise that the intrinsic heterogeneity of such systems is due essentially to the fondamental

difference between the underlying visions of time: continuous time governing some physical process' evolution, and discrete operation of the controlling — usually software — one.

In this chapter, we propose a formal definition of a system which intends to capture both continuous and discrete systems — the two major types of technical subsystems that compose a given industrial system (see Section 2.2). [1] The key point, on which our approach relies, is a common (discrete) model of time, based on the use of a non-standard model of real numbers (see Section 2.1). Making this (very strong) change allows, indeed, to take in account in the same way both conservative physical systems and computer systems (see again Section 2.2 for several examples). Moreover, our systemic models are always causal (see Section 2.2.1): non-causality appears indeed in our approach as the consequence either of abstraction (i.e. simplifying our model) or standardisation (i.e. going back to the usual model of time). Another advantage of our approach is that it integrates a number of already classical system types, as for example synchronous, Hamiltonian or dynamical systems (see Section 2.3 for some insights on these questions).

2.1 Time

In order to model all industrial systems in a unified way, the first key problem — as already mentioned above — is to develop a common functional [2] framework that takes into account both continuous and discrete systems, that is to say, systems whose time evolution is represented either continuously or discretely (typically physical and computer systems). At this point, two directions can be chosen to construct an unified theory, depending on whether one prefers to develop a continuous or a discrete point of view on time. In this thesis, we develop the discrete approach in order to keep the usual intuitions originating from computer systems. The price to pay is then the change of model of real numbers, on which relies the concept of time. This allows to capture continuous systems in the same global framework as the discrete ones. Note, however, that one could also do the opposite by always dealing with the usual continuous model of time. This approach would lead to developing a distribution point of view (see [84]) on computer systems, typically using Dirac combs to model discrete entries of a given software system.

2.1.1 Non-standard analysis vs. the classical one

To develop a global (discrete) unified framework for dealing both with continuous and discrete industrial systems, we go back to the 18th century representation of real numbers (see [28, 56, 55]).

Indeed, in the 17th and 18th centuries the common idea of real numbers was different from the modern one. The reasoning of the Differential Calculus was carried out with the help of the quantities called "infinitesimals". For instance, to compute the derivative of a given function $f(x)$, one would consider the *increment* of this function given an *infinitesimal* increment dx to x. Thus the derivative $f'(x)$ was defined by setting

$$f'(x) = \frac{f(x+dx) - f(x)}{dx}\,.$$

[1] In particular, we do not try to integrate human systems — such as company organisations — in our modelling even if such systems are also fundamental for some type of applications (typically information systems).

[2] Here, "functional" refers to the fact that systems will only be considered here as input/output functions. We do not model the architecture, on which relies a system.

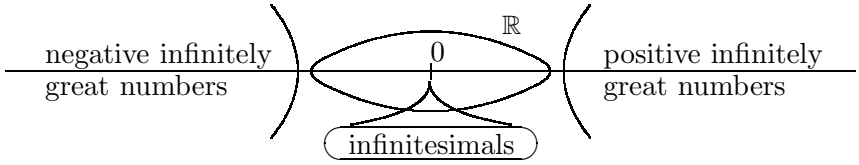


Figure 2.1: Graphical representation of non-standard real numbers.

For example, applying this reasoning to $f(x) = x^2$, one obtains the following computation

$$f'(x) = \frac{(x+dx)^2 - x^2}{dx} = \frac{2x\,dx + dx^2}{dx} = 2x + dx \approx 2x\,,$$

where the last relation signifies that, dx being infinitesimal, it *vanishes* in the final expression.

Although the notion of infinitesimals is extensively discussed in the article "différentiel" of the encyclopedia edited by Diderot, d'Alembert, and Le Rond [28], its status was clearly not very solid due to the absence of a strict mathematical formalism to support it. This led in the beginning of 19th century, to the development by Cauchy (1812) and Weierstrass (1820) of the modern analysis, based on the epsilon-delta reasoning.

In the beginning of 1960s Robinson has developed a constructive theory that he called Non-standard Analysis "since it involves and was, in part, inspired by the so-called Non-standard models of Arithmetic" (see [81]). This theory constructs a set ${}^*\mathbb{R}$ of *non-standard reals*, [3] which is a real-closed field that contains all usual real numbers, but also the infinitesimal reals (i.e. the non-zero non-standard real numbers that have their absolute value strictly less than any $r \in \mathbb{R}_*^+$) — we denote the set of infinitesimal reals by $\mathbb{I}$ — and their inverses, which are the infinitely great reals, i.e. those with an absolute value strictly greater than any usual real number $r \in \mathbb{R}$ (see Figure 2.1).

In particular, two non-standard real numbers x and y are said to be *infinitely close* (denoted by $x \approx y$) if and only if $x - y$ is infinitesimal.

The field ${}^*\mathbb{R}$ is elementarily equivalent to $\mathbb{R}$, which means that the first order logical properties of $\mathbb{R}$ and ${}^*\mathbb{R}$ (expressed in the logical theory of ordered fields) are exactly the same (see [13] and [48] for more model theoretical fundamentals of non-standard analysis). Observe also that, among all non-standard real numbers, one can of course consider the set ${}^*\mathbb{Z}$ of *non-standard integers* that, on top of standard integers, contains infinitely great ones, having absolute value greater than any $n \in \mathbb{N}$.

Contrary to the usual analysis, the Non-standard Analysis developed by Robinson is based on a weak form of the Axiom of Choice (see Appendix A). [4] This introduces a problem of philosophical order concerning the constructibility of non-standard real numbers and, consequently, the possibility of numerical calculation.

However, the elementary equivalence between $\mathbb{R}$ and ${}^*\mathbb{R}$ guarantees that most of the usual

[3] A star on the left of a symbol as in ${}^*\mathbb{R}$ stands always for "non-standard", whereas on the right, it means either that zero is not included, as in $\mathbb{R}^*$ or ${}^*\mathbb{R}^*$ (which denote respectively the usual and the non-standard sets of non-zero real numbers) or that we speak of the set of words over a given alphabet, as in A^* (which one of the latter two notations is used shall be clear from the context).

[4] Regarding the usual analysis, Dedekind has shown that it is independent from the Axiom of Choice by providing in 1872 a construction of $\mathbb{R}$, which reinterprets in modern mathematical language the Book V of Euclid's Elements. This construction does not make use of any form of Axiom of Choice, and is based on what is now called *Dedekind cuts*.

algorithms can also be applied to non-standard numbers. [5] On the other hand, we do not see any specific obstacle to developing a non-standard formal calculus. Thus, one can argue that, while a model based on non-standard analysis is not (or, at least, not directly) applicable to simulating systems, it can very well be in the foundation of a modelling and verification tool. Moreover, one can easily imagine that, once a model of a given system is complete, it should be possible to determine the "practical infinity", i.e. a sufficiently great usual real number $R \in \mathbb{R}$ such that $\varepsilon = 1/R$ can be considered infinitesimal for the purposes of simulating with a given precision this particular system.

The above arguments, together with the attractive idea of developing a unified framework for systems operating in both discrete and continuous time (see next section), convinces us of the viability of our non-standard approach to systems' modelling.

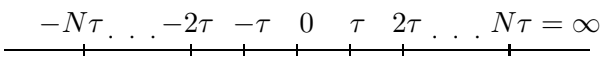
A comprehensive introduction to non-standard reals can be found in the paper of Lindstrøm in [65], whereas an axiomatic approach is developed in [29]. We also provide a basic synthesis of this theory in in Appendix A).

2.1.2 Time scales

From now on, we will suppose that the time is modelled by ${}^*\mathbb{R}$ throughout all this chapter. Let us therefore give the following first fundamental definition.

Definition 2.1 *Let $\tau \in {}^*\mathbb{R}_*^+$ be a strictly positive non-standard real number. The set $\mathbb{T}_\tau = {}^*\mathbb{Z}\,\tau$ will then be called the* time scale *of step $\tau > 0$. Any element $t \in \mathbb{T}_\tau$ is said to be a* moment *on this time scale.*

A time scale can therefore be seen as a discrete [6] series of clock ticks occurring at times $n \cdot \tau$, with $n \in {}^*\mathbb{Z}$ being a non-standard integer eventually infinitely great.



Note 2.2 It is important to notice here that not only does ${}^*\mathbb{Z}$ contain infinitely great integer numbers, but it contains, indeed, an infinite number of different infinitely great integers! Moreover, the cardinality of this set is continuum, which is exactly the property which allows us to use ${}^*\mathbb{Z}$ to model $\mathbb{R}$.

The following lemma, which is a direct consequence of Corollary A.27 in Appendix A, shows that we can recover usual continuous time within this model by considering time scales with infinitesimal steps (i.e. with $\tau \approx 0$).

Lemma 2.3 *Let $\mathbb{T}_\tau$ be a time scale with an infinitesimal step τ and let $r \in \mathbb{R}$ be any standard real number. Then there always exists a moment $t \in \mathbb{T}_\tau$ which is infinitely close to r.*

More generally, we can classify all possible time scales $\mathbb{T}_\tau$ in three following groups according to the nature of their corresponding step τ:

- *continuous* time scales have an infinitesimal time step, that is $\tau \approx 0$,

[5] One of the typical examples of such algorithms is the Euclidean algorithm for computing the greatest common divisor of two [non-standard] integers.

[6] Here the word discrete primarily implies that for each moment on a time scale the *next* moment is uniquely defined (this is not true, however, for the *previous* one).

- *discrete* time scales have a time step that is a non-infinitesimal bounded non-standard real number, i.e. $\tau \approx r$ for some strictly positive usual real number $r \in \mathbb{R}_*^+$,
- *infinite* time scales have an infinitely great non-standard real number as a time step.

Note that the latter case is not of practical interest as there are essentially only three "standard" moments on an infinite time scale, that is to say $-\infty$, 0, and $+\infty$. Therefore we will only consider the time scales of the first two types.

Let us finally give the following three definitions that we will use in our model of complex industrial systems.

Definition 2.4 *A time scale $\mathbb{T}_\tau$ is said to* refine *another time scale $\mathbb{T}_{\tau'}$ — which is denoted by $\mathbb{T}_{\tau'} \preceq \mathbb{T}_\tau$ – if and only if one of the two following equivalent properties holds:*

- $\mathbb{T}_{\tau'} \subset \mathbb{T}_\tau$
- $\exists\ N \in {}^*\mathbb{N},\ \tau' = N\tau$

Definition 2.5 *Given two time scales $\mathbb{T}_\tau$ and $\mathbb{T}_{\tau'}$, we shall call* synchronisation points *all the moments that belong to both of these time scales, i.e. to $\mathbb{T}_\tau \cap \mathbb{T}_{\tau'}$.*

Observe that, if $\mathbb{T}_\tau$ refines $\mathbb{T}_{\tau'}$, any moment on $\mathbb{T}_{\tau'}$ is a synchronisation point.

Definition 2.6 *A* temporal filter *is a set of time scales $\mathcal{F} = \{\mathbb{T}_\tau\}_{\tau \in T}$ such that $\preceq$ is a total order on $\mathcal{F}$. In other words, $\mathbb{T}_\tau$ and $\mathbb{T}_{\tau'}$ are always comparable by $\preceq$ for any $\tau, \tau' \in T$.*

2.2 Systems

2.2.1 Definition

In this section, we give a formal definition of the notion of system which tries to capture the reality of industrial complex systems. Most systems — be that industrial scale technological systems or networks of information processing machines (and probably also certain biological systems) — are too complex to be modelled or analysed as a whole, but can be treated as the result of the integration of several components. These components tend to be simpler systems that can in their turn be decomposed in the same way. We arrive eventually at the level where all such components are *elementary systems*, i.e. sufficiently simple to be considered independently of their structure. The following key definition is an attempt to provide a theoretical model for this as yet intuitive and informal process (on which however relies system engineering in the industry).

Definition 2.7 *A* system *S is recursively defined as the union of the following elements:*

- *an input/output mechanism that consists respectively of:*
  - *an* input channel *x capable of receiving — only at moments that belong to a given time scale $\mathbb{T}_{\tau_i}$ called the* input time scale *— data that belong to a given set I, called the* input domain *of S (and also denoted by $In(S)$),*
  - *an* output channel *y capable of emitting — only at moments that belong to a given time scale $\mathbb{T}_{\tau_o}$ called the* output time scale *— data that belong to a given set O, called the* output domain *of S (and also denoted by $Out(S)$),*

- *two internal storage mechanisms that consist respectively of:*
  - *an* internal memory *given by a tape indexed by* ${}^*\mathbb{N}$ *— with a window that can take any value within* ${}^*\mathbb{N}$ *— which may contain any (non-standard) finite number*[7] *of values in a given set* M *(also denoted by* $Mem(S)$*), called the* memory domain*,*
  - *an* internal state set *which is just an arbitrary (non-standard) finite set* Q *(that is also denoted by* $State(S)$*),*
- *an internal time behaviour which is given by*
  - *an* internal time scale $\mathbb{T}_{\tau_s}$ *that refines both the input and the output time scales, i.e. which satisfies both* $\mathbb{T}_{\tau_s} \preceq \mathbb{T}_{\tau_i}$ *and* $\mathbb{T}_{\tau_s} \preceq \mathbb{T}_{\tau_o}$*,*
  - *an* internal state evolution function $q(t)$ *mapping each element* $t \in \mathbb{T}_{\tau_s}$ *onto some element* $q(t) \in Q$ *(called the value of the internal state at moment* t*),*
  - *an* internal memory evolution function $m(t)$ *mapping each element* $t \in \mathbb{T}_{\tau_s}$ *onto some element* $m(t) \in M^f$ *(called the value of the internal memory at moment* t*),*[8]
- *three transition mechanisms that consist respectively in:*
  - *a function* read $: Q \times I \to Q \times M^f$ *that can read a given value on the input channel and write correspondingly a series of values — depending on the status of a state* $q \in Q$ *(updated after the operation) — onto the internal memory,*
  - *a* controller function $\delta : Q \times M \times {}^*\mathbb{N} \to Q \times M \times {}^*\mathbb{N}$ *that allows — as we will see — to replace an element of the internal memory by another one,*[9]
  - *a function* write $: Q \times M^f \to Q \times O$ *that can read a set of values on the internal memory and write another value — that may depend on the status of a state* $q \in Q$ *(updated after the operation) — onto the output channel,*
- *a finite (in the usual standard meaning) set of systems* $Sub(S) = \{S_1, \ldots, S_n\}$*, which are called the* subsystems *of* S *that are equipped with:*
  - *for each* $k = 1, \ldots, n$*, a function* $\rho_k : Q \times Out(S_k) \to Q \times M^f$ *that reads the output of* S_k*, writes it into the internal memory and eventually changes the internal state,*
  - *for each* $k = 1, \ldots, n$*, a function* $G_k : Q \times M^f \times \bigotimes_{i=1}^{n} Out(S_i) \to Q \times In(S_k)$ *that defines the interactions between the subsystems,*

  and, moreover, such that the output and input time scales of all these subsystems are always refined by the internal time scale of S[10]*.*

This definition allows us to construct a symbolic model of a given system's architecture (which is, of course, conditioned by a number of choices, such as, for instance, the decomposition into subsystems — see in particular Definition 2.17 below and the discussion preceding it), but does not so far allow to consider the time behaviour of this system, i.e. the evolution of its state.

[7] Note that "finite" should be taken in the non-standard meaning. We recall that, in this context, a set is said to be finite if and only if it can be put in bijection with a set of the type $[0, N]$ where N stands for any (either usual or infinitely great) non-standard positive integer in ${}^*\mathbb{N}$.

[8] We denote here by M^f the set consisting of all non-standard finite (in the non-standard meaning) sequences over a set M (i.e. partial functions ${}^*\mathbb{N} \to M$ with non-standard finite support).

[9] Depending on the value of a given state of Q which can be also changed by the action of δ.

[10] Forming therefore altogether a temporal filter.

To progress in this last direction, we first introduce the notion of instantaneous description of a system.

Definition 2.8 *Let S be a system. An* instantaneous description *of S is then any quadruple of the type $d = (t, q, m, i) \in \mathbb{T}_{\tau_s} \times Q \times M^f \times {}^*\mathbb{N}$, where*

- *$t \in \mathbb{T}_{\tau_s}$ is a moment of the internal time scale $\mathbb{T}_{\tau_s}$ of S,*
- *$q \in Q$ is the value $q(t)$ of the internal state evolution function of S at moment t,*
- *$m \in M^f$ is the value $m(t)$ of the internal memory evolution function of S at moment t,*
- *$i \in {}^*\mathbb{N}$ is the position of the window of the internal memory of S at moment t* [11].

We are now in position to define the time behaviour of a system, by considering a series of instantaneous descriptions (completed by the time evolutions of the values of the input and output channels that we did not integrate in these descriptions) indexed by its internal time scale and submitted to some natural transition constraints.

Definition 2.9 *Let S be a given system. A* time behaviour *associated with S is then a family $d(t) = (t, m(t), q(t), i(t))$ of instantaneous descriptions of S, where t describes the internal time scale $\mathbb{T}_{\tau_s}$ of S and where one always passes from $d(t)$ to $d(t+\tau_s)$ by executing one of the following possible transition actions:*

1. *for any $k = 1, \ldots, n$ and at every synchronisation point $t + \tau_s$ between all concerned time scales, update the input of S_k and the current internal state of S by setting*

$$(q(t+\tau_s), x_k(t+\tau_s)) = G_k\Big(q(t); m(t); y_1(t), \ldots, y_n(t)\Big), \tag{2.1}$$

   where x_k denotes the input channel of the subsystem S_k and where each y_i stands for the output channel of S_i correspondingly,

2. *for any $k = 1, \ldots, n$ and at each synchronisation point t between the internal time scale $\mathbb{T}_{\tau_s}$ of S and the output time scale of S_k, read the output of S_k and update both the internal memory of S — beginning at the current position of its associated window — and the current value of its internal state by setting*

$$(q(t+\tau_s), m(t+\tau_s)_{i \geq i(t)}) = \rho_k\Big(q(t); y_k(t)\Big), \tag{2.2}$$

3. *at each synchronisation point t with the input time scale, perform a* read *operation — depending on the value of the internal state of the system — and update both the internal memory of S beginning at the current position of its window and its internal state:*

$$(q(t+\tau_s), m(t+\tau_s)_{i \geq i(t)}) = \mathrm{read}\Big(q(t); x(t)\Big), \tag{2.3}$$

4. *at each synchronisation point t with the output time scale, perform a* write *operation, taking into account both the internal state of the system (that is updated after the operation) and the values of the internal memory that begin at the current position of its window:*

$$(q(t), y(t)) = \mathrm{write}\Big(q(t); m(t-\tau_s)_{i \geq i(t-\tau_s)}\Big), \tag{2.4}$$

[11] Whose value is given by $m(t)$.

5. *at any other moment t on $\mathbb{T}_{\tau_s}$, update the internal state, the value of the current position of the internal memory of S and the current position of its associated window:*

$$(q(t+\tau_s), m(t+\tau_s)_{i(t)}, i(t+\tau_s)) = \delta\Big(q(t), m(t)_{i(t)}, i(t)\Big)\,. \tag{2.5}$$

Time behaviours may however not be unique. This remark leads us to the following definition of deterministic systems (observe that the systems consider in this thesis are mainly deterministic.)

Definition 2.10 *A system S is said to be* deterministic *if and only if S possesses exactly one time behaviour. When this is not the case, the system is said to be* non-deterministic.

The previous definitions are illustrated by the diagram in Figure 2.2. For the sake of simplicity and clarity, we will occasionally vary this representation. In particular, we will sometimes omit domains of variables or their names, if the omitted part is clear from the context.

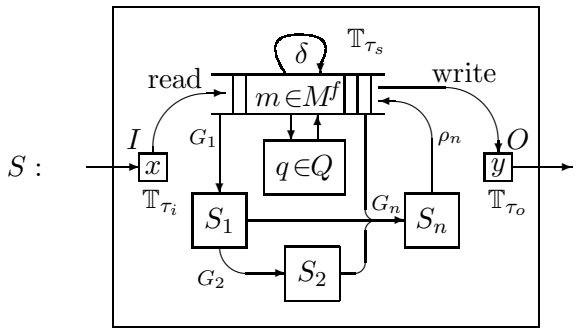


Figure 2.2: Graphical representation of a system S.

Note 2.11 One can also consider systems with more than one tape, that is with several internal memory variables, which corresponds to saying that M is a direct product of different independent sets. In this case, there has to be a window as above defined for each tape.

Note 2.12 The subsystem graph of S is the graph which is formed by taking $Sub(S)$ as vertices and with edges defined by $(\mathcal{G}_i)_{i=1,\dots,n}$, i.e. such that there is an edge going from S_i to S_j if and only if x_j depends on y_i according to Equation (2.1). This graph can contain cycles and does not necessarily have to be connected. Cycles in the subsystem graph represent feedback loops common to a large number of industrial system, particularly where some form of control is involved.

Note 2.13 Note that we suppose that each time scale $\mathbb{T}$ of an input or an output channel of a subsystem of a given system S is always refined by the internal time scale $\mathbb{T}_{\tau_s}$ of this system, i.e. that one has $\mathbb{T}_{\tau_s} \prec \mathbb{T}$, especially if such a time scale is *free from interactions* which means that the corresponding channel is never implied in one of the relations (2.1).

In order to reflect the level of complexity of a system, we now introduce the notion of *order* of a system. A system is said to be of *order* N if it is constructed using only subsystems of order $N-1$ and less. Systems of *zeroth order* are called *elementary* systems.

Definition 2.14 *We define the* order *of a system S by setting*

$$ord(S) = \begin{cases} 0 & \text{if } Sub(S) = \emptyset\,, \\ 1 + \max\Big\{ord(S') \,\Big|\, S' \in Sub(S)\Big\} & \text{otherwise}\,. \end{cases}$$

We are now in position to introduce the notion of well defined system (which will in fact be the only kind of systems that we shall consider in the sequel).

Definition 2.15 *A system S is said to be* well defined *if there exists a positive standard integer $N \in \mathbb{N}$ such that $N = ord(S)$.*

Note that each time behaviour of a system implicitly defines an input/output relation of the following type (keeping the notations of the previous definitions):

$$y(t_0 + \tau_o) = \mathcal{F}\Big(x(u), m(t), q(t) \;\Big|\, u, t \in [t_0, t_0 + \tau_o)\Big)\;, \tag{2.6}$$

where t_0 describes the output time scale $\mathbb{T}_{\tau_0}$ of the system and where $u, t \in [t_0, t_0 + \tau_o)$ signifies that each of these variables is taken between t_0 and $t_0 + \tau_o$ (excluding the upper limit) on the time scales $\mathbb{T}_{\tau_i}$ and $\mathbb{T}_{\tau_s}$ respectively. For more simplicity, we will rewrite equivalently the input/output relation (2.6) in the following simplified functional form

$$y = \mathcal{F}(x; q, m)\,. \tag{2.7}$$

It is easy to see that the function $\mathcal{F}$ is uniquely associated with a given system S if and only if S is a deterministic system. Observe also that $\mathcal{F}$ must obviously (by construction) always be a causal function, i.e. each value $y(t)$ depends only on the values $x(t')$ with $t' < t$.

Note 2.16 Observe, however, that in several classical models of systems, such as dynamical systems (see [32]) or synchronous systems (see [14]), non-causality is a real potential problem. Such a situation is, indeed, always a direct consequence of the collapse between system feedbacks and the hypothesis, implicit in usual continuous modelling (and corresponding to standardisation with respect to our approach), but totally explicit in synchronous modelling, that no time is required to realise the internal operations of a given system.

One can distinguish two main and complementary approaches to system design, namely specification and engineering. While the latter is interested in full details in the effective way a system can be constructed, the former only deals with formal requirements on the input/output relation of a system. In other words, the system specification approach considers a system as a "black box" with a precise functional behaviour, but without trying to know how such a behaviour is obtained. This approach is fundamental in the industry: a computer manufacturer will for instance be able to provide the specification of a given wireless interface to different suppliers that may realise different wireless computer interfaces from their structural point of view, as soon as the input/output relations specified by the manufacturer are exactly the same. Such situations are modelled by the following definition for equivalent systems.

Definition 2.17 *Two systems S_1 and S_2 are said to be* equivalent *if and only if one has both $In(S_1) = In(S_2)$ and $Out(S_1) = Out(S_2)$, and the following condition is satisfied*

$$\forall\; t \in \mathbb{T}_{\tau_i},\; x_1(t) = x_2(t) \;\Rightarrow\; \forall\; t \in \mathbb{T}_{\tau_o},\; y_1(t) = y_2(t)\,,$$

where (x_1, y_1) and (x_2, y_2) stand respectively for the input and output channels of S_1 and S_2.

Note 2.18 The previous definition has mainly a meaning when S_1 and S_2 are deterministic.

The above definition of equivalence allows us to make one more observation. Indeed, one can be easily convinced now that Definition 2.14 (and consequently Definition 2.15) only has sense when certain restrictions are applied to the elements of which a system is composed. Indeed, the following theorem implies that, when no such restrictions apply, the hierarchy defined by the notion of order effectively collapses.

Theorem 2.19 *For any given system S defined as above, there exists an equivalent zeroth order system S', that is having $Sub(S') = \emptyset$.*

Proof. For a system S of first order, it is sufficient to construct a system S' with the same input and output time scales as those of S, the internal time scale refining that of S and of all its subsystems (such time scale obviously exists as $Sub(S)$ is standard finite), and both internal domains (memory and state) defined as direct products of those of S and its subsystems.

To prove this theorem for systems of arbitrary order, one then proceeds recursively by integrating the elementary components into those of first order (which effectively diminishes the order of the system and that of all its subsystems by 1). ■

As it has been mentioned above, this theorem shows that the notion of order does not really have any "objective" meaning, but rather reflects the level of abstraction at which a system is modelled. This corresponds well to the nature of the hierarchical decomposition process, where one starts from a general specification of a system's global behaviour, and then refines until subsystems of a certain level of simplicity are obtained. Moreover, as we shall see in Chapter 4, the hierarchical decomposition in subsystems of lesser order has certain advantages for the analysis of a system, as individual subsystems can be eventually extracted from the global context to be analysed separately.

2.2.2 Elementary systems

We will now study in more detail some important classes of elementary systems, that is of systems of order zero. In particular, we will show that our framework already allows us to capture several interesting classical types of systems of very different nature. Note finally that we will concentrate in this subsection on elementary systems of the following three types (using here a general terminology which is not specially reserved to elementary systems):

1. *software systems* model phenomena observed mostly in information technologies: they are characterised by the fact that their three defining time scales are all discrete,

2. *physical systems* are characterised by the fact that their three defining time scales are all continuous: they are generally used to model real-life physical systems,

3. finally, *hybrid systems* mix — by definition — both discrete and continuous time scales.

Elementary software systems

Elementary software systems have only discrete time scales. Their input and output spaces will be called alphabets (a set of letters or symbols). We assume that an elementary software system is equipped with a tape and a corresponding window that indicates the position of its head (this definition can be generalised to take into account several tapes).

As depicted on Figure 2.3, elementary software systems are receiving — on the input channel — data within some input alphabet I at a rate given by the input time scale $\mathbb{T}_{\tau_i}$. They are also emitting — on the output channel — data that belong to some output alphabet O at a rate given by the output time scale $\mathbb{T}_{\tau_o}$. Moreover, any elementary software system has also the right to perform — at a rate given by its internal time scale $\mathbb{T}_{\tau_s}$ — a number of internal actions controlled by the value of an internal state $q \in Q$, that is to say:

- read an input data x, transform it into a word (depending only on x) on the tape alphabet and write it finally on the tape beginning at the current position of its window,
- change the value of the element of the tape that is obtained by looking on the current position of its window (that can be updated after the operation),
- write an output data y by taking a word w on the tape beginning at the current position of its window and transforming it (depending only on w) into y.

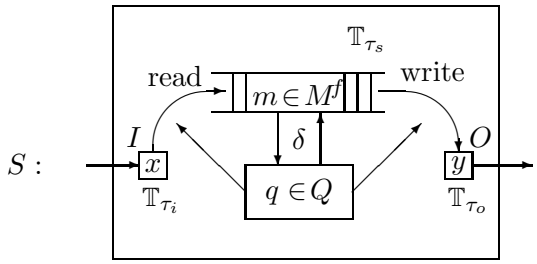


Figure 2.3: Graphical representation of an elementary software system S.

As we can see, elementary software systems are therefore just a slight generalisation of usual Turing machines, obtained by adding to this classical model a permanent input/output temporal behaviour. The reader can indeed easily check that Turing machines (or equivalently recursive functions if one prefers to stay within a functional approach) correspond to the degenerated case of our model, where one considers elementary software systems that can only perform a unique read action (or whose input channel will only receive a single input data during all possible moments of time). Note also that as an immediate consequence of this simple observation, we obtain the undecidability of the existence of a system's output!

Example 2.20 (One element buffer)
Let us now present an example of elementary deterministic software system that we will use in the sequel (as a subsystem of a higher order system). Our example consists of a buffer capable of storing only — at each moment of time — one single message out of a message set A. We assume that this buffer has two input channels. On the first input channel, the buffer can only receive either a message $m \in A$ or a distinguished empty message ε. On the second one, it can receive either a write request, that we will denote by '$\uparrow$', or again the distinguished empty message ε. The buffer stores each message it receives on the first channel in a fixed memory cell, overwritten each time a new non-empty message arrives on the same channel. When the buffer receives a write request, it sends the currently stored message on its output channel. A representation of such a buffer is shown in Figure 2.4.

Let us now describe how to model this simple buffering mechanism by an elementary deterministic software system, denoted by Buf. The input, internal and output time scales of such a system are all discrete with respective time steps $\tau_i = \tau$ and $\tau_s = \tau_o = \tau/2$. In other words,

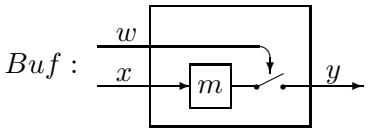


Figure 2.4: Graphical representation of a one-element buffer.

we require the buffer to operate internally on a rate which is twice as fast as its input rate. The input domain of Buf is clearly modelled by $(A \cup \{\varepsilon\}) \times \{\varepsilon, \uparrow\}$ in order to take into account the two entry channels, whereas the output domain is just equal to $A \cup \{\varepsilon\}$. The memory domain will be equal to A. Finally the internal state set of Buf is defined as $\{r, \varepsilon, \uparrow\}$ (the first state models the reading of the input channel, when the two last ones correspond to the two possible writing decisions on the output channel). Therefore we have:

$$In(Buf) = (A \cup \{\varepsilon\}) \times \{\varepsilon, \uparrow\}, Mem(Buf) = A, Out(Buf) = A \cup \{\varepsilon\}, State(Buf) = \{r, \varepsilon, \uparrow\},$$

The control mechanisms of Buf are now given by the following transition functions:

$$\delta = -, \quad \texttt{read}(r; (x, w)) = \begin{cases} (w; x) & \text{if } x \neq \varepsilon, \\ (w; -) & \text{otherwise}, \end{cases} \qquad \texttt{write}(q; y) = \begin{cases} (r; y) & \text{if } q = \uparrow, \\ (r; \varepsilon) & \text{if } q = \varepsilon, \end{cases} \tag{2.8}$$

where $-$ means not defined or no action (depending on the situation). Observe that the choice of δ just reflects the fact that the input message is stored in a single cell of the internal memory on which no action can be made. The unique possible time behaviour of the buffer consists then just in alternating a `read` and a `write` action at each moment of its internal time scale. ■

Elementary physical systems

Before presenting the notion of elementary physical system, let us first introduce the general framework on which relies this concept. We will, indeed, suppose that each instance p of a given physical parameter φ (such as mass, distance, kinetic energy, potential energy, torsion energy, temperature, kinetic momentum, etc.) that we will deal with, can always be both

1. *measured* using a measure function m_φ, which means that one can associate to each such physical quantity p of type φ its measure $m_\varphi(p) \in {}^*\mathbb{R}$,

2. *decomposed into a (non-standard) finite sum of infinitesimal quantities*, i.e. a sum of physical quantities p of type φ that have an infinitely small measure $m_\varphi(p) \approx 0$.

The set of all physical quantities of a given type φ (e.g. energy) is then called the physical domain associated with φ and denoted by $\mathbb{P}_\varphi$. In the same way, the physical infinitesimal domain of type φ (e.g. energy quanta) — which is denoted by $\mathbb{I}_\varphi$ — consists of all infinitesimal physical quantities of type φ.

Elementary physical systems can now be described exactly in the same way as elementary software systems, i.e. by a mechanism similar to the one given by Figure 2.3, the only (but fundamental) difference being here that such systems manipulate physical quantities using continuous time scales. An elementary physical system is indeed characterised by the fact that it has

1. continuous input, internal and output time scales,

2. input and output domains that are both equal to the same finite (in the usual meaning) product of physical domains, i.e. both equal to $\otimes_{i=1}^{n} \mathbb{P}_i$ for some finite (standard) positive integer $n \in \mathbb{N}$, where each $\mathbb{P}_i$ stands for a physical domain of a given type,

3. an internal domain which is necessarily equal to $\otimes_{i=1}^{n} \mathbb{I}_i$ where each $\mathbb{I}_i$ stands for the infinitesimal physical domain associated with the physical domain $\mathbb{P}_i$ which is involved in either the corresponding input or output domain.

For the sake of simplicity, we can of course consider — without any extension of the representation power of our model — that an elementary physical system has a finite (in the usual sense) number of tapes, each devoted to some particular infinitesimal physical domain.

Such elementary systems are intended to model real physical systems, considered as transformers of infinitesimal physical quantities. Indeed, in our framework, the behaviour of an elementary physical system S has to be physically interpreted as follows:

- S receives at each moment on its input time scale (hence infinitely often) a vector x that consists of different physical quantities of given types, i.e. a vector $x \in \otimes_{i=1}^{n} \mathbb{P}_i$, where each $\mathbb{P}_i$ stands for some physical domain; it transforms then each component $x_i \in \mathbb{P}_i$ of x into a (non-standard) finite sequence $(x_i^j)_{j=1...N}$ — written on a specific tape of the internal memory of S — of infinitesimal physical quantities within $\mathbb{I}_i$ (i.e. of the same type as $\mathbb{P}_i$) whose sum has the same measure as that of x_i, i.e. such that

$$\sum_{j=1}^{N} m_i(x_i^j) = m_i(x_i)\,, \tag{2.9}$$

  where m_i stands for the measure function associated with the physical domain $\mathbb{P}_i$,

- S can transform infinitely often, at the rate given by its internal time scale, any infinitesimal physical quantity of a given type written on one of its tapes into another infinitesimal physical quantity of another given type (that can also be stored on another tape),

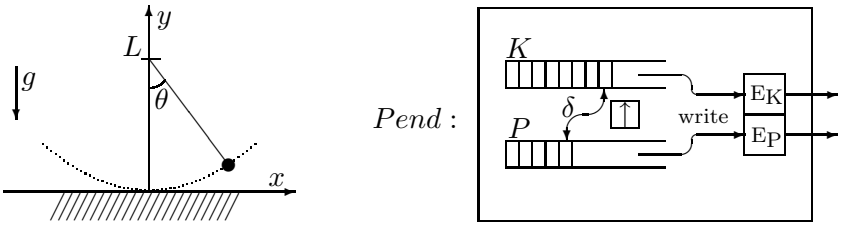
- S emits at each moment on its output time scale (hence again infinitely often) a vector $y \in \otimes_{i=1}^{n} \mathbb{P}_i$ whose components are obtained by "gluing" together sequences of infinitesimal physical quantities (of compatible types) coming from the internal memory of S, by using the reverse process of the initial writing mechanism as described above.

One can prove that large classical classes of physical systems — such as Hamiltonian systems (cf. [70]) — can be recovered (as higher order systems) in our model. We will however not prove here this last result which is quite technical, but rather illustrate on a simple example how to analyse a classical mechanical system as an elementary deterministic physical system.

Example 2.21 (Simple pendulum)
Let us now consider a simple pendulum as shown in Figure 2.5-*a*. The pendulum consists of a point mass m attached to the point $(0, L)$ by a rigid string of negligible mass and of length L (when hanging freely the pendulum touches the ground).
The motion of such a pendulum can be described by applying Newton's second law of motion, leading immediately to the following differential equation

$$m\,L\,\dot{\varphi} = -m\,g\,\sin\theta\ , \tag{2.10}$$

where θ and $\varphi = \dot{\theta}$ stand respectively for the angle formed by the string and the y axis and for the corresponding angular speed. In its turn, Equation (2.10) is clearly equivalent to the



(a) (b)

Figure 2.5: Simple pendulum: mechanical (a) and systemic (b) representations.

following energy preservation equation (obtained by integrating the former and multiplying both sides by L)

$$\frac{1}{2}\, m(L\varphi)^2 - mgL\cos\theta = C\,, \tag{2.11}$$

where the first and the second summand in the left hand side represents respectively the kinetic and the potential energy of the pendulum, and where C stands for the initial potential energy of the pendulum, when it is in the point farthest from the y axis with zero angular speed.

An elementary deterministic physical system $Pend$ modelling such a pendulum is shown in Figure 2.5$-b$. This system takes no input and provides on the output channel a pair of physical quantities that consist respectively of the pendulum's current kinetic and potential energy. Its internal and output time scales are supposed to be the same continuous time scale (with dt as common infinitesimal time step). Finally, we define the output domain, the memory domain and the internal state set of the system to be respectively equal to

$$Out(Pend) = \mathbb{E}_K \times \mathbb{E}_P \ ,\ Mem(Pend) = \{0, de_K\} \times \{0, de_P\} \ ,\ State(Pend) = \{x, s\} \times \{\uparrow, \downarrow\}\,,$$

where $\mathbb{E}_K$ and $\mathbb{E}_P$ denote respectively the two physical domains which are associated with kinetic and potential energy, where de_K and de_P stand respectively for two infinitesimal quanta of kinetic and potential energy — with a common measure $m_K(de_K) = m_P(de_P) = de \in \mathbb{I}$ — and where the x, s and the arrow states should respectively be interpreted as the two possible internal actions of the pendulum (exchanging energy on its two tapes — see below — or sending physical quantities to the output channel) and as the two possible directions of the pendulum's motion. This system has therefore two tapes K and P, each of them containing a (non-standard) finite number of copies of the corresponding infinitesimal quantum of energy (the other parts of the two tapes being equal to 0). Moreover the memory of our system evolves in such a way that it always contains the same global number (necessarily infinitely great) $N \in {}^*\mathbb{N}$ of energy quanta (which satisfies the energy conservation condition $N\,de = C$). The system's behaviour consists then essentially in taking at each moment of time one quantum of energy from one tape, depending on the direction the pendulum is moving, and putting it on the other one, until the working tape is empty. The corresponding controller function δ is given below.

$$\delta((x,\uparrow),(k,p),(i_K,i_P)) = \begin{cases} \Big((s,\uparrow),(0,de_P),(i_K-1,i_P+1)\Big) & \text{if } (k,p)=(de_K,0) \text{ and } i_K>1, \\ \Big((s,\downarrow),(0,de_P),(0,i_P+1)\Big) & \text{if } (k,p)=(de_K,0) \text{ and } i_K=1, \end{cases}$$

$$\delta((x,\downarrow),(k,p),(i_K,i_P)) = \begin{cases} \Big((s,\downarrow),(de_K,0),(i_K+1,i_P-1)\Big) & \text{if } (k,p)=(0,de_P) \text{ and } i_P>1, \\ \Big((s,\uparrow),(de_K,0),(i_K+1,0)\Big) & \text{if } (k,p)=(0,de_P) \text{ and } i_P=1, \end{cases}$$

where i_K and i_P are the cursors of tapes K and P. The `write` function is then defined as the constructor of the two global physical quantities (i.e. kinetic and potential energy) that can be obtained by summing all infinitesimal quanta that respectively exist on tapes K and P. [12] This last function works only in state $(s, \updownarrow)$ and ends in state $(x, \updownarrow)$ (where $\updownarrow$ stands for any type of arrow). The unique possible time behaviour of our (deterministic) system consists hence just in alternating permanently a tape exchange operation with a write operation.

It is then immediate to see that the elementary physical system that we just defined, always satisfies — by construction — the energy conservation equation

$$E_K + E_P = C \ , \tag{2.12}$$

where E_K and E_C stand for the measures of the kinetic and the potential energy of the pendulum, that is exactly equivalent to Equation (2.11). Note, however, that our approach does not connect the physical quantities that we manipulated (here kinetic and potential energy) with the high level parameters φ and θ that were used in writing down Equation (2.11).

To fill this gap, we must interpret the pendulum as a new deterministic system $Newpend$ of higher order that contains the previous elementary physical system $Pend$ as a subsystem. This new system has another subsystem which makes alternatively the two only operations:

- it reads the outputs of $Pend$ and transforms them — by applying the two associated measure functions — into (non-standard) real values k and p that are stored in its internal memory,
- it takes the above two non-standard real values k and p and writes them on its output channel by applying the following transformation

$$\texttt{write}\,(k,p) = \left(\frac{1}{L}\sqrt{\frac{2\,k}{m}}\,,\, \arccos\frac{p}{mgL} \right) \ ,$$

  which can be expressed within our model by making use of adapted subsystems, due to the fact that we are only dealing here with analytic transformations (see Section 2.3).

If one identifies the output channels of this last subsystem to the output channels of $Newpend$, it is then obvious to see that $Newpend$ produces on its own output channels the pair (φ, θ) in such a way that the energy conservation Equation (2.11) is always fulfilled. ■

Elementary hybrid systems

Elementary hybrid systems are systems of order 0 which can transform continuous behaviours into discrete ones (or vice-versa). Therefore, they can naturally be applied to model interfaces between software and physical systems. The two following examples illustrate how to interpret within our approach two classical interfaces of this kind — here a sampler and a modulator — that are totally fundamental in practice.

Example 2.22 (Sampler)
A sampler is a mechanism that takes a continuous time input function and produces a discrete sequence of samples of its values. It can be modelled by an elementary deterministic hybrid

[12] We can easily assume that the cardinality of these two families of infinitesimal quanta is respectively given by i_K and i_P. Under this hypothesis, the `write` function can be more precisely defined — independently of the value of the current internal state of the system — by setting $\texttt{write}(K, P) = (i_K\, de_K, i_P\, de_P)$.

system H_τ which is parameterised by the time step $\tau > 0$ of its discrete output time scale. The input and the internal time scales of such a system are both continuous with the same infinitesimal time step $dt = \tau/N$ (where $N \in {}^*\mathbb{N}$ is a given infinitely great non-standard integer). The input, output and memory domains of H_τ are all equal to ${}^*\mathbb{R}$, whereas its internal state set is equal to $[0, N]$, i.e. we put:

$$In(Sampler) = Out(Sampler) = Mem(Sampler) = {}^*\mathbb{R}\,,\ State(Sampler) = [0, N],.$$

The control mechanisms of H_τ are now given by the following transition functions:

$$\begin{array}{rcl} \texttt{read}\,(i;x) & = & \left\{ \begin{array}{cc} (i+1;x) & \text{if } 1 \leq i < N\,, \\ (0;x) & \text{if } i = N\,, \end{array} \right. \\ \texttt{write}\,(0;y) & = & (1;y)\,, \\ \delta & = & -\,, \end{array}$$

where $-$ means that no action should be taken (on the tape). The unique temporal behaviour of this deterministic hybrid system is now obvious: at each moment of its internal time scale, which is not a synchronisation point with the time scale of the output channel, the system uses the `read` function to memorise the value on the input channel inside a fixed cell of its internal memory. On the other hand, the system outputs — with the `write` function — the value currently stored in this cell at each synchronisation point with the output time scale, i.e. at every τ, thus producing a discrete sequence out of a continuous one. ■

Example 2.23 (Modulator)
The action of a modulator is reciprocal to that of a sampler and consists in converting a discrete sequence of real numbers into a continuous function by making use — at a discrete rate — of a pulse shape $p_\tau(t)$, which will be considered here — for the sake of simplicity (see below for more details on this hypothesis) — as a given function with interval $[0, \tau]$ as support. A modulator can then be modelled by an elementary deterministic hybrid system Mod_{p_τ} parameterised by this last function. The input, output and memory domains of such a system are equal to ${}^*\mathbb{R}$, whereas its internal state set is equal to $[0, 2\,N-1]$ where N stands for a fixed infinitely great positive integer within ${}^*\mathbb{N}$, i.e. we put:

$$In(Mod) = Out(Mod) = Mem(Mod) = {}^*\mathbb{R}\,,\ State(Mod) = [0, 2\,N-1]\,.$$

The input time scale of Mod_{p_τ} is then a discrete time scale of time step τ, whereas its internal and output time scales are both continuous with respective time steps $dt_s = \tau/(2\,N)$ and $dt_o = \tau/N$. The control mechanisms of this system are now given by the following transition functions:

$$\begin{array}{rcl} \texttt{read}\,(0\,;x) & = & (1\,;x)\,, \\ \texttt{write}\,(2\,k\,;y) & = & (2\,k{+}1; y \cdot p_\tau(k\,dt_o)) \quad \text{for every } k \in [1, N{-}1]\,, \\ \delta(k;m;0) & = & \left\{ \begin{array}{cc} (2;m;0) & \text{if } k = 1\,, \\ (k{+}1;m;0) & \text{if } k = 2\,i+1 \text{ with } i \in [1, N{-}2]\,, \\ (0;m;0) & \text{if } k = 2\,N{-}1\ . \end{array} \right. \end{array}$$

The unique temporal behaviour of such a system consists then in reading, at each possible synchronisation point, a value on its input channel and storing it in a fixed cell of its internal memory, a blank operation (i.e. doing nothing during one single internal clock tip) and then applying alternatively a `write` operation and a δ controlled transition.

Note also that though the pulse shape p_τ in the above example gives us a degree of freedom in the way we can modulate the continuous output, the most realistic choice is unfortunately just to take $p_\tau(t) = 1$ for any $t \in [0, \tau]$. The examples that correspond to practical situations are indeed obtained if the pulse shape has support bigger than $[0, \tau]$, which requires to add several consecutive input values. This technique would however lead to a more complicated modelling, and therefore we restrained ourselves to the simpler system presented above. ■

2.2.3 Addition and multiplication of reals

As it has been mentioned above, it is essential for the development of any theory of systems to show that the main classes of known systems can be modelled in it. In the spirit of this chapter, this requires one to show first of all that it is possible to model elementary components of these systems.

One of such important classes consists of the so-called *dynamical systems* (see [32]), which are defined in terms of analytic functions (see Section 2.3 fore more detail). A function f is said to be *analytic on an open set* $D \subset \mathbb{R}$, if for any $x_0 \in D$, the function f can be represented by a real series

$$f(x) = \sum_{n=0}^{\infty} a_n (x - x_0)^n \,,$$

convergent in a neighbourhood of x_0.

When transferred to non-standard domain, the definition of analytic functions is expressed in terms of non-standard finite sums and products. Thus, the first step to modelling dynamical systems is, indeed, to show that addition and multiplication of standard reals can be effectively modelled in our definition.[13]

In the following two examples, we assume that we are capable of performing addition and multiplication on ${}^*\mathbb{N}$. Indeed, these operations are computed in the same way as for standard $\mathbb{N}$, and in this case corresponding Turing machines can be constructed (see Section 2.2.2 for a way to model Turing machines). We then show that we can perform the same kind of operations on standard reals.

We consider a quantum $\varepsilon \in {}^*\mathbb{R}$ such that $\varepsilon = 1/E > 0$, where $E \in {}^*\mathbb{N}$ is an infinitely great integer. In the same manner as for physical systems, described in the previous section, we assume that the function `read` consists in quantifying the real number on input, i.e. transforming it into a sequence of quanta. Similarly, we assume that `write` is capable of reconstituting the real from this sequence of quanta.

Example 2.24 (Addition of standard reals)
Considering all that has been said above, it is quite simple to construct a system that calculates the sum of two given reals. Indeed, consider the system *Add* defined as follows

$$In(Add) = \mathbb{R}^2 \,, \quad Mem(Add) = \{0, \varepsilon\} \,, \quad State(Add) = \{r, w\} \,, \quad Out(Add) = {}^*\mathbb{R} \,,$$

$$\begin{aligned} \text{read}\,(r; x, y) &= (w\,; \overline{n} \cdot \overline{m}), \text{ such that } n\,\varepsilon \approx x\,,\ m\,\varepsilon \approx y\,, \\ \delta &= -\,, \\ \text{write}\,(w\,; \overline{l}\,) &= (r\,; l\,\varepsilon)\,, \end{aligned}$$

[13] The importance of modelling these two operations is further underlined by the consideration that a theory of modelling complex systems calls immediately for the corresponding computability theory. In our case, this would necessarily be computability of real functions, based inherently on addition and multiplication.

where, for any $k \in {}^*\mathbb{N}$, we denote by $\overline{k}$ the sequence of k quanta of measure ε (thus $\overline{k} \in \{0, \varepsilon\}^f$, with A^f denoting, as before, the set of non-standard finite sequences over the alphabet A), the symbol $\cdot$ (central dot) represents concatenation, and $-$ (dash) represents, as in several previous examples, an empty operation. Thus the unique possible time behaviour of the system consist in alternatively reading two real numbers on the input, and writing their sum $(n+m)\,\varepsilon \approx x+y$ on the output. ■

Example 2.25 (Multiplication of real numbers)
In the same manner as in the previous example, we assume two reals x and y on input. Taking n and m such that $n\,\varepsilon \approx x$ and $m\,\varepsilon \approx y$, we observe that

$$x \cdot y \approx (n\varepsilon) \cdot (m\varepsilon) = (n \cdot m)\,\varepsilon^2 = \frac{n \cdot m}{E}\,\varepsilon\,. \tag{2.13}$$

We now run the Euclidean algorithm on $n \cdot m$ and E to obtain q and r such that $n \cdot m = q \cdot E + r$ with $0 \leq r < E$. We then have

$$\frac{n \cdot m}{E}\,\varepsilon = \frac{q \cdot E}{E}\,\varepsilon + \frac{r}{E}\,\varepsilon \approx q\,\varepsilon\,. \tag{2.14}$$

Thus we can design our system with two working memory bands and the Euclidean algorithm as a subsystem. Indeed, setting

$$\begin{array}{ccc} In(Mult) = \mathbb{R}^2\,, & Mem(Mult) = \{0,\varepsilon\}^2\,, & Out(Mult) = \mathbb{R}\,, \\ State(Mult) = \{r, c, ei, eo, w\}\,, & & Sub(Mult) = \{Eucl\}\,, \end{array}$$

$$\begin{array}{rcl} \text{read}\,(r\,;x,y) & = & (c\,;\overline{n},\overline{m}\,),\ \text{such that}\ n\,\varepsilon \approx x\,,\ m\,\varepsilon \approx y\,, \\ \delta(c\,;\overline{n},\overline{m}\,) & = & (ei\,;\overline{n\cdot m}\,,\overline{m}\,)\,, \\ G(ei\,;\overline{n},\overline{m}\,) & = & (eo\,;n,E)\,, \\ \rho(eo\,;q,r) & = & (w\,;\overline{q},\overline{r}\,)\,, \\ \text{write}\,(w\,;\overline{q},\overline{r}\,) & = & (r\,;q\,\varepsilon)\,, \end{array}$$

where the subsystem $Eucl$ takes two non-standard integers on input, and produces on output the quotient q and the residue r of integer division of the first one by the second one. Equations (2.13) and (2.14) prove that the output of our system is infinitely close to the desired multiple of the two reals on input. ■

Note 2.26 Observe that using the above two example as components, it is relatively easy to construct a system to compute any analytic function.

2.2.4 An example of higher order system

In order to show the potential of our approach in practical situations, we will now try to study how to model within our framework a simplified version of a radio transmission system which immediately appears as a higher order system (in the sense of Definition 2.14).

Example 2.27 (Simplified radio transmission)
We will now show how to model a communication transmitter taking messages from a buffer Buf — as described in Example 2.20 (we also keep the notations of this example) — and transmitting them over a radio channel. In order to achieve this goal, we must first introduce an

encoder component *Enc* which reads messages from the buffer, converts them into binary form and encodes the resulting sequence of bits — by blocks of N_b bits — into complex symbols. [14] Whenever the encoder has less than N_b bits to work on, it sends a write request to the buffer. A description of the systemic organisation of this component is given just below in Figure 2.6.

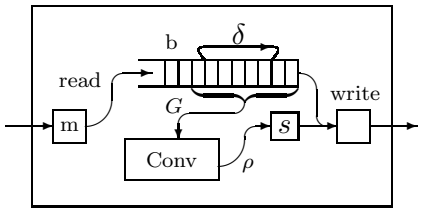


Figure 2.6: Description of the encoder component.

The corresponding (non-deterministic) system can indeed be defined as follows. We first suppose that the internal time scale of *Enc* has a discrete time step which is N_b times smaller than the rest of the global radio transmission system to which it will belong. The input domain of the encoder is just A, the same set of messages as that of the buffer to which it is connected. The output domain is taken to be $\mathbb{C} \times \{\varepsilon, \uparrow\}$ in order to model the fact that two types of output information can be sent on two different channels (i.e. a complex encoded symbol or a write request). The memory domain is defined as $\{0,1\}^* \times \mathbb{C}$ since one must be able to store (on two different tapes) both sequences of bits and complex numbers. Finally, the internal state set is reduced to two elements q_0 and q_1 (see the explanations below). These elements can be summarised by setting:

$$In(Enc) = A\,,\ Out(Enc) = \mathbb{C} \times \{\varepsilon, \uparrow\}\,,\ Mem(Enc) = \{0,1\}^* \times \mathbb{C}\,,\ State(Enc) = \{q_0,\, q_1\}\ .$$

We also suppose that *Enc* possesses a unique subsystem *Conv* (i.e. that $Sub(Enc) = \{Conv\}$), which can convert blocks of N_b bits into complex symbols according to a given table and which has the same input time scale as *Enc* (the second state of *Enc* is in particular used to interconnect the reading of a bit on the tape of *Enc* with its sending to *Conv*). The control mechanisms of *Enc* can now be given by means of the following transition functions:

$$\begin{aligned}
\texttt{read}\,(q_0; m) &= (q_0; b \cdot \overline{m})\,, \\
G(q_0; (b, s)) &= (q_1; b_1)\,, \\
\delta(q_1; (b, s); 0) &= (q_0; (b \ll 1, s); 0)\,, \\
\rho(q_0; (b, s); y) &= (q_0; (b, y))\,, \\
\texttt{write}\,(q_0; (b, s)) &= \begin{cases} (q_0; (s, \uparrow)) & \text{if } |b| < N_b\ , \\ (q_0; (s, \varepsilon)) & \text{otherwise}\ , \end{cases}
\end{aligned} \tag{2.15}$$

where $m \in A$ and $\overline{m} \in \{0,1\}^*$ stand respectively for the message at the input of the encoder and for its binary form, and where $(b, s) \in \{0,1\}^* \times \mathbb{C}$ is the current value of the internal memory of the system, with b denoting the sequence of bits that is currently stored on the main tape of the encoder and s being the complex symbol produced by the *Conv* subsystem. We also denoted above the concatenation product by '$\cdot$', the first bit of the sequence b by b_1 and the shift of b by one bit by $b \ll 1$. Note finally that we did not represent here — for the sake of clarity —

[14] The simplest example of such a process is the Binary Phase Shift Keying (BPSK) modulation protocol that converts every bit $b \in \mathcal{B} = \{0,1\}$ into the real number $(-1)^b$ (for more details, see for example [75] or [54]).

the precise behaviour of δ on the main tape of *Enc* (that we modelled just as a single cell that can contain sequences of bits on which concatenation products can be directly applied).

The transmitter in its turn receives on the input a sequence of complex symbols $(s_k)_{k\geq 0}$ at a discrete rate and generates on the output a radio signal of the form

$$u(t) = \sum_{k=0}^{\infty} s_k \, p(t - t_k) \, ,$$

where $p(t)$ is a pulse shape function and t_k is the moment when the k-th complex symbol is sent. This new operation can be realised by a modulator similar to that of Example 2.23. Composing now, as shown in Figure 2.7, the three components we introduced, one obtains the simplified radio transmitter system that we wanted to model.

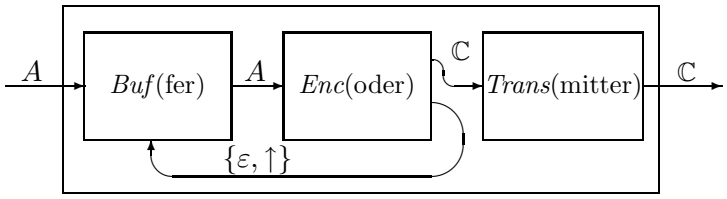


Figure 2.7: Graphical description of a simplified radio transmission chain.

Observe that in the above example, the nature of the first two components (that is to say *Buf* and *Enc*) is completely discrete, whereas that of the third component (*Trans*) is hybrid (discrete input with continuous output). This should be interpreted within the classification that we introduced in Section 2.2.2, as Buffer and Encoder are purely software, whereas Transmitter serves as an interface between software and physical environments. ■

2.3 Discussion

In this chapter, we tried to show that it is possible to construct a general unified theory of systems which may give a common framework to deal both with continuous and discrete systems (that are the two core kinds of systems used in engineering modelling). Note that our theory allows to take into account large classes of classical systems.

For example, consider a (controlled) causal dynamical system defined as in [32] by the following system of equations

$$\begin{cases} \dot{q}(t)(= dq/dt) = A_0(q) + \sum_{i=1}^{n} u_i(t) A_i(q), \\ y(t) = h(q), \end{cases} \tag{2.16}$$

where the state q belongs to an analytic real manifold Q of (standard) finite dimension with a given value $q(0)$, vector fields $A_0, A_1, \ldots, A_n$ and function $h : Q \to \mathbb{R}$ are analytic, defined in a neighbourhood of $q(0)$, and the (control) inputs $u_1, \ldots, u_n : [0, T] \to \mathbb{R}$ are piecewise continuous. Such systems can be recovered within our framework as the standardisation of appropriate physical-like deterministic systems.

Indeed, to recover this family of systems, one should transform the differential equation (2.16) into a finite difference equation involving an infinitesimal time step. To see that this last equation expresses the functional behaviour of a system, the key problem is then just to

prove that any $A(q)$ can be computed by a system when A is an analytic vector field. Under appropriate calculability assumptions, this last property can be reduced to proving that (non-standard) finite sums and products of non-standard real numbers can be realised by a system, and this can be done with the help of the two components from the examples of Section 2.2.3.

In the same way, synchronous systems — which are the discrete equivalents of causal dynamical systems — can also be modelled in our framework by software deterministic systems with the same input and output time scales.

However, our approach does not reduce to re-interpret already existing classes of systems. It also introduces new classes of systems (such as the elementary software and physical systems that were discussed in Section 2.2.2) and to lots of new questions (can one develop a Λ-calculus formalism for systems? what are the "good" system sub-families? can one construct a complexity theory for systems? what are really the differences existing between deterministic and non-deterministic systems? etc.) that should now be studied more in details in order to understand more deeply the notion of system.

To summarise the above discussion one can say that, in the global perspective, we are looking to formulate the foundations of a theory of calculability for complex industrial systems, where the systems, as we have defined them in this chapter, will serve the same role as Turing machines have done in classical theory: they are not to be employed for real-life computing, but as a theoretical tool allowing to reason on systems.

Similarly, we expect our model to satisfy a thesis in the spirit of the Church's one, i.e. we expect it to be possible to reformulate any formalism describing temporised "reasonably computable" systems in terms of our model.

Chapter 3

An Example on System Level: UMTS Infrastructure

In this chapter we illustrate the notion of a system that we have introduced in Chapter 2. More precisely, we proceed in the spirit of the *V development cycle* mentioned in Section 1.1.3, by giving here a systemic representation of a UMTS infrastructure, starting from a global view of the system, and descending in the hierarchical decomposition to single out the subsystems that will be necessary for the following chapters.

3.1 The predecessor: a quick look at the GSM

The Universal Mobile Communications System has inherited a number of its features and architectural elements from its Second Generation predecessor — the Global System for Mobile Communications.[1] Before moving on to the overview of the UMTS, let us therefore present a quick summary of GSM architecture.

3.1.1 Network elements

The GSM network can be divided into three domains: Mobile Station (MS), Base Station Subsystem (BSS), and Network and Switching Subsystem (NSS) (see Figure 3.1).

Mobile Station

The Mobile Station is a user-side component of a GSM network. It consists of a Mobile Equipment (ME) and Subscriber Identification Module (SIM).

Mobile equipment can be either portable (phone, PDA) or fixed (computer, car phone), and is commonly referred to as *terminal* or *handset*. It is uniquely identified by its International Mobile Equipment Identity (IMEI) number, which is primarily used for security purposes.

A SIM is a smart card that is inserted into the ME. Each SIM card contains an International Mobile Subscriber Identity (IMSI) number that uniquely identifies the subscriber to the network thereby allowing access to subscribed services. Along with the subscriber identity and some basic services, a SIM card can store, in particular, the user's phone book. Altogether, this

[1] As it has already been mentioned in Chapter 1, originally the acronym GSM stood for *Groupe Spécial Mobile*.

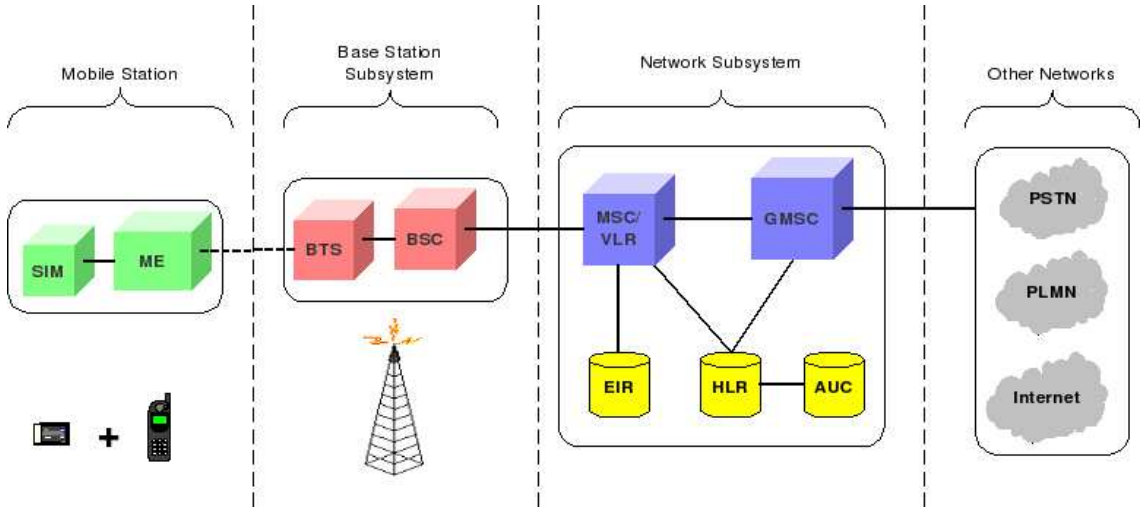


Figure 3.1: GSM network architecture.

information allows both to easily change the handset, and, on the contrary, to use the same handset with different SIM cards (for example when the user's home operator does not have a roaming agreement with the operator of the network in the area, where the user is currently located, or simply when the user is unwilling to pay the extra charge for roaming).

Base Station Subsystem

On the other side of the radio connection, the Base Station Subsystem is composed of two parts: the Base Transceiver Station (BTS) and the Base Station Controller (BSC). A GSM network is composed of many BSSs, each controlled by one BSC, but eventually containing multiple BTSs. The BSS monitors the radio connections to the MS, and performs the necessary channel coding and decoding.

The Base Transceiver Station, or simply the Base Station, is the interface for the MS to the network. It handles all communications with the MS via the air interface (technically referred to as the Um interface in the GSM specifications). The available transmitting power of a BTS essentially defines the potential cell size, i.e. its coverage area. However, the necessary cell size depends rather on the subscriber density in the area: in large urban areas, the number of BTSs deployed is large and the corresponding cell size is small; in contrast, there is usually a far smaller number of BTSs deployed in rural areas, and consequently the cell size has to be quite large to provide sufficient coverage.

The Base Station Controller manages the radio resources for multiple BTSs, the number of which varies but could be up to several hundred. As well as the allocation and release of radio channels, the BSC is responsible for handover management when the MS moves over to an area covered by a different BSC. Similar to all other interfaces in GSM, the interface between the BSC and a BTS is standardised and is referred to as the Abis interface.

Network and Switching Subsystem

While MS and BSS are responsible for the *physical layer* of the GSM system, the Network and Switching Subsystem (NSS) is in the heart of the entire GSM network. It contains the core

switching component — the Mobile Switching Centre (MSC), as well as the four databases below.

- Home Location Register (HLR),
- Visitor Location Register (VLR),
- Authentication Centre (AuC),
- Equipment Identity Register (EIR).

The Mobile Switching Centre is a digital ISDN switch that sets up connection to other MSCs and to the the BSCs (via the so-called A interface). In addition to the functions of a normal switching node in a fixed network, MSC handles mobile subscribers, which includes registration, authentication, location updating etc. Each GSM network must have at least one MSC. The totality of the MSCs in the network form its wired (fixed) backbone, and connect it to the Public Switched Telecommunications Network (PSTN). An MSC performing the interconnection with other networks is called a Gateway MSC (GMSC).

3.1.2 Frequency reuse

GSM uses a combination of both the Time Division Multiple Access (TDMA) and Frequency Division Multiple Access (FDMA) technologies. In this section, we shall concentrate on FDMA for the following reason. When Frequency Division Multiple Access is used, each user is assigned a different carrier frequency, which allows him to transmit information while reducing to minimum interference with other users' signals. Thus, available frequencies constitute a valuable resource, essentially determining the network capacity.

Making use of the inherent property of radio waves to attenuate with distance, cellular networks are fundamentally based on the principle of *frequency reuse*, that is assigning the same frequency to users sufficiently distant one from another to assume that their signals do not interfere.

Example 3.1
Let R be the radius of a circular zone A to cover with $N = 7$ available frequencies. Suppose for simplicity that each frequency allows to establish exactly one communication link. Thus, without frequency reuse, a maximum of 7 links can be established in the whole zone A.

Splitting the zone A into a number of smaller zones of radius $r < R$ (that is by using less powerful transmitters; see Figure 3.2), we can reuse frequencies by assigning one to each of these smaller zones to obtain the maximal number of simultaneous communications of the order of

$$n = \frac{\pi R^2}{\pi r^2} = \left(\frac{R}{r}\right)^2 ,$$

and taking, for example $R = 10\,km$ and $r = 500\,m$, we obtain the maximal number of simultaneous communication links of the order of 400 instead of 7! ■

In the example above, all frequencies are distributed over seven cells, and the arrangement of these cells is then repeated to cover the whole zone A. Such arrangements of cells are called *reuse patterns* or *clusters*. A cluster defines the network capacity and determines the level of interference, which can be of two kinds:

- *co-channel interference* is observed between two signals of the same frequency with different phase and amplitude conditions,

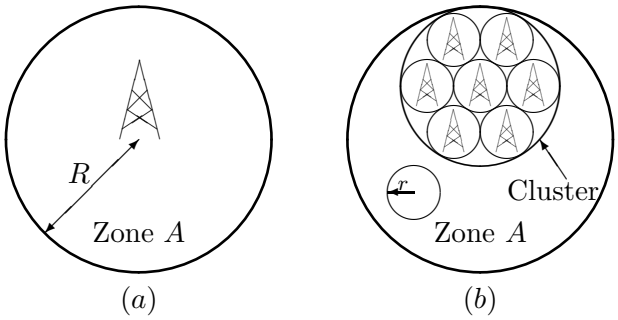


Figure 3.2: Illustration of the frequency reuse principle.

- *adjacent channel interference* is observed between two signals on frequencies close in the sense of the spectral distance.

In a more general case, one assumes that a cell can consist of one or several sectors depending on the type of the antennae (omnidirectional or directed) used in this particular cell. A (p, n)-*cluster* is then a cluster consisting of p cells and n sectors (the cluster in Figure 3.2-b is therefore a (7,7)-cluster).

The problem of assigning frequencies to sectors in a cluster can be modelled in the following manner. First of all, assume that

1. a regular pavement of a plane and a reuse pattern with n sectors are given,
2. each sector has one transceiver antenna,
3. n consecutive frequencies are available for assignment.

We have to find a frequency allocation such that frequencies in neighbouring sectors respect some given spectral distance constraints. This can be expressed in terms of permutations of order n.

Problem 3.2 *Find a permutation σ of order n such that for all $i, j \in [1, n]$ we have $|\sigma(i) - \sigma(j)| \geq M_{i,j}$, where σ is the permutation that assigns a frequency $\sigma(i)$ to sector i, and M is a symmetric square matrix with zero diagonal defining the spectral distance constraints.*

This matrix M, called *compatibility matrix* or *interference matrix*, is essentially determined by

- the reuse pattern type,
- the inter-pattern arrangement (the way the cluster is repeated to pave the plane),
- the spectral distance imposed between the frequencies assigned to
  - two sectors of the same cell,
  - two sectors of adjacent cells,
  - two sectors having similarly oriented antennae.

Even in this simplified model, the problem of frequency assignment happens to be NP-complete.

In practice, frequency assignment depends on the projected traffic density (e.g. urban vs. rural areas) and landscape considerations. The location of cells and the corresponding attributed frequencies constitute one of the operators' most vigorously guarded trade secrets.

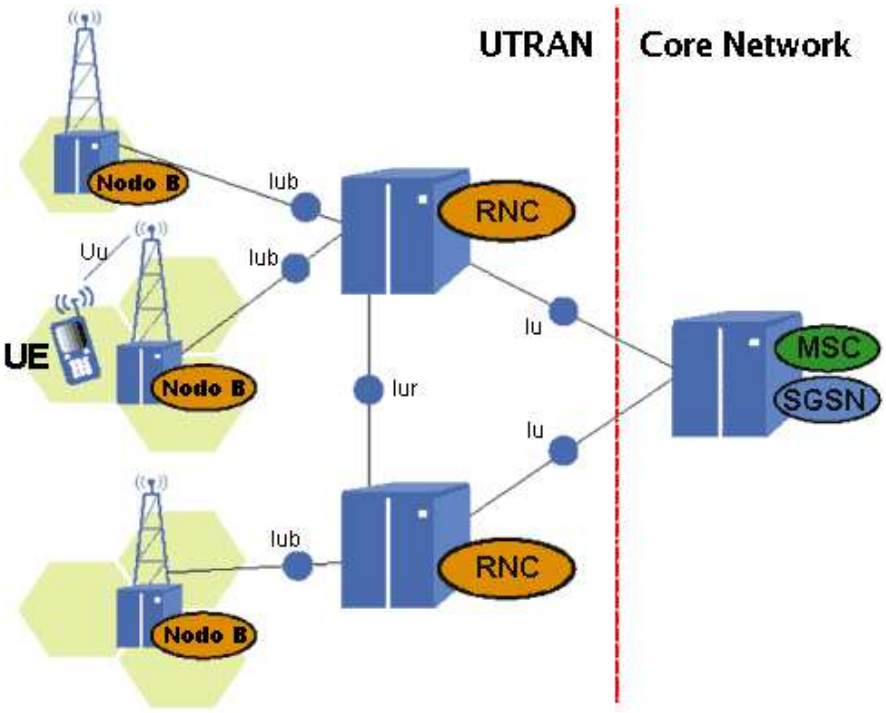


Figure 3.3: UMTS network architecture (from [47])

3.2 Overview of the UMTS architecture

By now, numerous descriptions of UMTS can be found in the literature (see for example [22, 44, 47, 93]). We shall therefore limit ourselves here to a brief summary of its aspects that we refer to in the sequel.

Universal Mobile Telecommunication System (UMTS) is one of the most significant advances in the evolution of telecommunications into third generation (3G) networks. It is based on the Wideband Code Division Multiple Access (W-CDMA) air interface, and its main advantage, compared to second generation (2G) systems, is the increased achievable data rate, allowing a number of new applications.

The UMTS standard can be seen as an extension of existing networks. Similarly to GSM, it consists of three interacting domains: User Equipment (UE), UMTS Terrestrial Radio Access Network (UTRAN) and Core Network (see Figure 3.3). The Core Network is the part of the existing infrastructure integrated into UMTS and providing access to services. Its elements can be extended to adopt the UMTS requirements, and do not have to be redesigned completely. At the same time, the air interface used in UMTS is radically different from that of 2G systems.[2] Therefore radio access components, i.e. UTRAN and UE, have to be completely new designs. The UTRAN in its turn is subdivided into individual Radio Network Systems (RNS), each controlled by a *Radio Network Controller* (RNC), which inherits considerably from the Base Station Controller in GSM networks. The RNC is connected to a number of *Nodes B* — 3G equivalent of Base Stations —, each of which can serve one or several cells.

As it has already been mentioned in the introduction, the UMTS infrastructure as a whole is not exactly in the range of complex industrial systems that we define, due to its intrinsic

[2] 2G systems are mainly based on Frequency Division or Time Division Multiple Access (FDMA and TDMA; see Section 3.1.2)

continuous structure evolution. Indeed, UMTS is a typical example of a network connecting an ever changing number of elements between themselves.[3]

Considered as a network, UMTS presents a clear separation in two types of objects:

1. the *vertices* of the underlying graph, i.e. the nodes of the network, communicating with each other, such as UE, Nodes B, and RNC; and
2. the *edges* of this graph, which represent the connections.

The objects of the first type, that is the ones that actively participate in communication can be perfectly well treated as systems in our sense, either software (as, for example, in the case of the RNC, which can be considered a purely software system as long as we do not descend in the analysis to the level of electric circuits), or hybrid (this is the case of UE and Node B, which combine both software and, due to the presence of radio signal, physical properties).[4]

The objects of the second type, i.e. those representing the connections between different nodes, are termed *interfaces*. The UMTS defines four new interfaces: Uu (connecting UE to Node B), Iub (Node B to RNC), Iu (RNC to Core Network), and Iur (RNC to RNC connection). From the systemic perspective, the interfaces can be essentially reduced to a collection of protocols that define what data and in which form has to be transmitted from one node to another. The subsystems that actually realise these interfaces belonging to corresponding systems of the first type above, we can say that the interfaces do not constitute systems in our sense of the word.[5]

3.2.1 Hardware network elements

As it can be seen from the previous section, the most important network elements specific to UMTS are the Radio Network Controller and the Node B that compose the UTRAN domain.

Radio Network Controller

The RNC enables autonomous radio resource management (RRM) by UTRAN. It performs the same functions as the GSM BSC, providing central control for the RNS elements (RNC and Nodes B). It also handles protocol exchanges between Iu, Iur, and Iub interfaces and is responsible for centralised operation and maintenance (O&M) of the entire RNS.

The functions of RNC are among others:

- radio resource control (RRC)
- admission control
- channel allocation
- handover control
- macro diversity
- uplink outer loop power control (UL OLPC)

[3] As we shall see in Section 3.3.2, this kind of a structure still can be modelled as a system in our sense. Although such a model does have certain advantages, it is nevertheless not particularly natural.

[4] It is, indeed, difficult to find a purely physical system in UMTS, as it is after all a *digital* communications system. A rare exception would be the model of the radio channel, over which the continuous signal is transmitted.

[5] Once again, observe here that even though the interfaces do not represent systems in our perspective, the corresponding channels that provide the communication medium (radio channel or electric wire) can very well be modelled as physical systems.

Node B

Node B is the physical unit for radio transmission/reception within cells. Depending on sectoring type, one or more cells may be served by a single Node B. It connects with the UE via the W-CDMA Uu radio interface and with the RNC via the Iub interface.

The main task of Node B is the conversion of data to and from the Uu radio interface, which includes

- forward error correction (FEC) coding,
- rate adaptation,
- W-CDMA spreading/despreading,
- modulation (quadrature phase shift keying (QPSK) or 16 symbol Quadrature Amplitude Modulation (16QAM)),

It measures quality and strength of the connection and determines various error rates such as Frame Error Rate (FER), Block Error Rate (BLER), and Bit Error Rate (BER), transmitting these to the RNC.

The Node B also participates in the Uplink Closed Loop Power Control (UL CLPC). It enables the UE to adjust its power using downlink (DL) Transmission Power Control (TPC) commands. The predefined values for UL CLPC are derived by the RNC via the Outer-Loop Power Control. (See Chapter 4 for a more detailed description of power control.)

3.2.2 Wideband CDMA

Wideband CDMA (W-CDMA) technology is used for the UTRAN air interface. It consists in spreading the user information bits over a wide bandwidth by multiplying the data sequence for each user with a different code (a different sequence of *chips*). All users' signals are transmitted in the same frequency bandwidth, one for uplink and one for downlink. This is called the Frequency Division Duplex mode (FDD mode). Moreover, another code is applied to separate different channels utilised by the same user. This is performed by a process called spreading that consists of the following two operations (see [5] for complete specifications).

- The first operation is called *channelisation*. It transforms each data symbol into a number of chips (thus increasing the bandwidth of the signal), by multiplying the data symbol by an Orthogonal Variable Spreading Factor [6] (OVSF) code. These orthogonal codes help in distinguishing between the transport channels. Figure 3.4 shows the process of generating a CDMA signal with an 8-chip channelisation code.
- The second operation is called *scrambling*. The spread signal is modulated with pseudo-random (also called pseudo-noise, PN) complex sequences. A PN-sequence allows to distinguish between different users in uplink and between different cells in downlink connection.

The PN code used to scramble the data, can be of two main types. A short PN code (typically 10–128 chips in length), can be used to modulate each data symbol. The short PN code is then repeated for every data symbol allowing for quick and simple synchronisation of

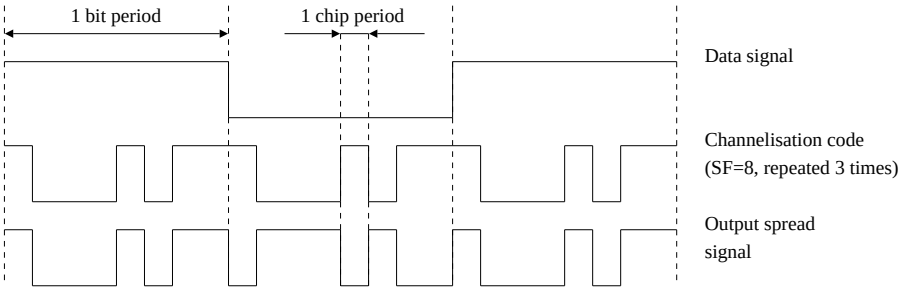


Figure 3.4: Spreading a data signal with an 8 chip channelisation code (data=010, code=01110100, output=01110100 10001011 01110100).

the receiver. Alternatively, a long PN code can be used. Long codes are generally thousands to millions of chips in length, and thus are only repeated infrequently.

At the reception, the overall signal is demodulated by multiplying each user's signal with its allocated code. The amplitude of the user signal increases on average by the spreading factor relative to that of the other interfering users' signals, i.e. this correlation detection extracts the user signal from the noise. This effect is termed *processing gain* and is a fundamental aspect giving all CDMA systems the robustness against interference. Let's take the example of the Speech service with a bit rate of 12.2 kbps. Assuming that QPSK modulation is used,[7] which is the usual case in practice, the processing gain is computed by taking

$$PG = 10 \log_{10} \left(2 \cdot \frac{3.84 \cdot 10^6}{12.2 \cdot 10^3} \right) \approx 28 \text{ dB},$$

where $3.84 \cdot 10^6$ is the carrier chip rate. After despreading, the signal power needs to be typically a few decibels above the interference and noise power. For Speech service in uplink with 2 antennas, the required energy per information bit to noise ratio Eb/No is typically in the order of 5 dB, and the required wideband signal-to-interference ratio is therefore $5 - 28 = -23$ dB. In other words, the signal power can be 23 dB under the interference or thermal noise power, and the W-CDMA receiver can still detect the signal.

In general, a CDMA network limiting factor is first the total available power then the number of codes. Thus, a good criterion to evaluate the capacity of a cell (the maximum number of active users in a cell) is the transmitted power and the power saturation rate at the base station.

3.2.3 Quality of Service and performance evaluation

As we have already mentioned in Section 1.2.1, wireless networks have by now attained an extremely high rate of market penetration, and therefore the operators' revenues, and consequently their market value, are determined by the average revenue per user (ARPU), and no longer by the number of users. An increase of the ARPU can hardly be realised from SMS and MMS based messaging services, and therefore new attractive services have to be introduced.

However, it is not sufficient to introduce a service, it is no less important to assure that customers do, indeed, use it. For this to happen, the following three conditions must be satisfied.

[6] Spreading factor (SF) is the length of the code, i.e. the number of chips corresponding to one data symbol. It can be any power of 2 between 4 and 512 depending on the transmission context.

[7] QPSK stands for Quadrature Phase Shift Keying modulation; it is based on a four symbol constellation with a combination of 2 bits being encoded by each complex symbol.

Table 3.1: Classes of services.

Error tolerant	Conversational voice and video	Voice messaging	Streaming audio and video	Fax
Error intolerant	E-commerce, interactive games	Telnet, Web browsing	FTP, paging, still image	E-mail arrival notification
	Conversational (delay $<<$ 1 s)	Interactive (delay $\approx$ 1 s)	Streaming (delay $<$ 10 s)	Background (delay $>$ 10 s)

1. The service must be adapted to the terminals available on the market, and, vice-versa, the terminals have to be conceived with the applications in mind.

2. The quality of the user experience must be sufficiently high.

3. The price of the service must correspond to user expectations, i.e. the value-for-money ratio must be sufficiently high.

The existing terminals have been considerably improved in the last years, and are approaching the second generation ones in terms, for example, of weight and battery life, while having evolved to accomodate the needs of 3G applications, in particular by the increase in the display size.

The operators' pricing policies being well beyond the scope of this thesis, we shall concentrate here on the quality of the user experience and on the network resources management. The 3GPP specifications include four classes for Quality of Service: Conversational, Streaming, Interactive, and Best Effort. These classes correspond to different levels of error and delay tolerance (see Table 3.1).

Let us now emphasise some of the criteria that allow to quantify the performances of a UMTS network. Generally speaking, these criteria can be divided in two groups, one for each side participating in the network operation, i.e. the subscriber and the operator. These two groups are

1. *User satisfaction criteria.* This group concerns the measures of the subscriber's experience such as

   - *call acceptance rate:* the probability that a connection is available at the moment when the user wants to gain access to a given service; this rate is influenced by a number of network elements, and in particular Access Control algorithms and handover management;
   - *amount of noise:* for Circuit Switched (CS) services such as voice and video communications, the quality of the received signal; this is generally influenced by the error correction coding, and by the signal to noise rate, and therefore by the efficiency of the power control both in up- and downlink;
   - *data rate:* this measure concerns the Packet Switched (PS) services such as file transfer or web browsing; it is affected, among others, by power control, channel adaption, and also by scheduling algorithms.

Table 3.2: End-user performance expectations for interactive services (from [95]).

Medium	Application	Degree of symmetry	Key performance parameters and target values		
			One-way delay	Delay variation	Information loss
Audio	Voice messaging	Primarily one-way	< 1 s for playback < 2 s for record	< 1 ms	$< 3\%$ FER
Data	Web-browsing (HTML)	Primarily one-way	< 4 s/page	N/A	Zero
Data	Transaction services — high priority (e-commerce)	Two-way	< 4 s	N/A	Zero
Data	E-mail (server access)	Primarily one-way	< 4 s	N/A	Zero

Note that both the amount of noise for CS services and the data rate for PS ones depend directly on the same parameter, which is the Block Error Rate (BLER), i.e. the probability that at the reception a transport block is not decoded correctly. [8] Observe that in case of CS services, the false block is simply discarded, which is the source of noise in the reproduced sound or video signal, whereas for PS services, it is retransmitted thus decreasing the overall data rate and correspondingly increasing the delays experienced by the user. A summary of user performance expectations for some services is shown in Table 3.2 (source [95]).

2. *Network resource criteria.* This group reflects the costs incurred by the operator of the network, both those that are connected with the resources such as power required to maintain the radio signal, and the capacity of network to accept users, as low performance of the latter implies a loss of profit. Some of the related measures are therefore

   - *network coverage:* this measure obviously indicates the amount of the territory covered by the operator's network, and it is mostly affected by the way the Nodes B are placed;
   - *network saturation:* the percentage of the network capacity effectively utilised for subscriber communication; maximum revenue can be obtained when the network is saturated, that is when all resources are utilised to generate profit; this measure is tightly linked with the call acceptance rate above, and similarly is influenced among others by such elements as access and power control;
   - *energy per bit:* this more low level measure gives a clue to the amount of energy spent to communicate one bit (this can be either an information bit or a transmitted one); it is, in particular, influenced by the propagation conditions for the link in question.

[8] Sometimes another equivalent measure is considered — the Frame Error Rate (FER). Typically, for a given service, a transport block consists of a fixed number of frames, in which context the connection between these two measures is obvious.

Observe, that one of the most recurrent influences on the above cited measures is the power control. It is, indeed, a very important feature of the UMTS, not only because it allows to limit the interferences, but also, for example, due to the fact that it considerably influences both the uplink and downlink capacity of the network (see Section 4.1 of Chapter 4 for a more detailed discussion).

A number of performance optimisation procedures implies a trade-off between the user satisfaction criteria and the network resource ones. For instance, the goal of power control consists in finding the minimal transmit power that allows to maintain the BLER just under the maximum rate required for a given service.

3.3 Two systemic approaches to UMTS

3.3.1 Single user case

As it has been mentioned in the opening of the chapter, we proceed in the spirit of the *V development cycle* described in the introduction. More precisely, we follow its descending branch, which goes from system specification to development of elementary components through system design or modelling. Therefore, omitting the components' development, and taking the description in the previous section as a summary specification, we shall now illustrate the modelling stage.

Before we attempt to give a systemic decomposition of the full UMTS infrastructure as it has been briefly described in the previous section, let us first consider the simplest possible case with only one user in the system, who is served by the same Node B all the time. The reasons for this are twofold. First of all, starting by this simpler case provides us with building blocks for the general model, and, equivalently, this simplified system can be used to analyse the performance of a single link. The results of this analysis can then serve as an input to the analysis of the complete system.

Figure 3.5 shows an hierarchical decomposition of a UMTS network based on the description above. Indeed, as it has been stated in Section 3.2, we separate the network in three domains: Core network, UTRAN, and User Equipment. For now, we do not decompose the UE and Core network any further, and we split UTRAN in its turn into Node B and RNC. This is justified by the assumption that, in the considered case where there is only one user, the whole network can be limited to one Node B and one RNC.

In this decomposition, Node B and RNC have subsystems responsible for their respective functionalities. Both UE and Core network could be decomposed in a similar manner, however we omit this for clarity as we will not consider these subsystems in the sequel.

Let us now consider a systemic representation of our network, at the abstraction level corresponding to the part above the dashed line in Figure 3.5. In addition to the five subsystems introduced in this figure, the system shown in Figure 3.6 has three subsystems modelling the interaction interfaces.

It is important to observe here that the component representing the radio channel (connecting UE with UTRAN via its Node B subsystem) is best modelled as a physical component, due to the continuous nature of a radio signal. This implies that, at the considered level of abstraction, both UE and Node B (and consequently UTRAN) are hybrid systems combining discrete and continuous time scales. All other subsystems in Figure 3.6 can be modelled as software ones.

Let us now focus our attention on the Node B subsystem. As it has been mentioned in Section 3.2.1, its key functionalities (cf. Figure 3.5) are channel coding and modulation (and their

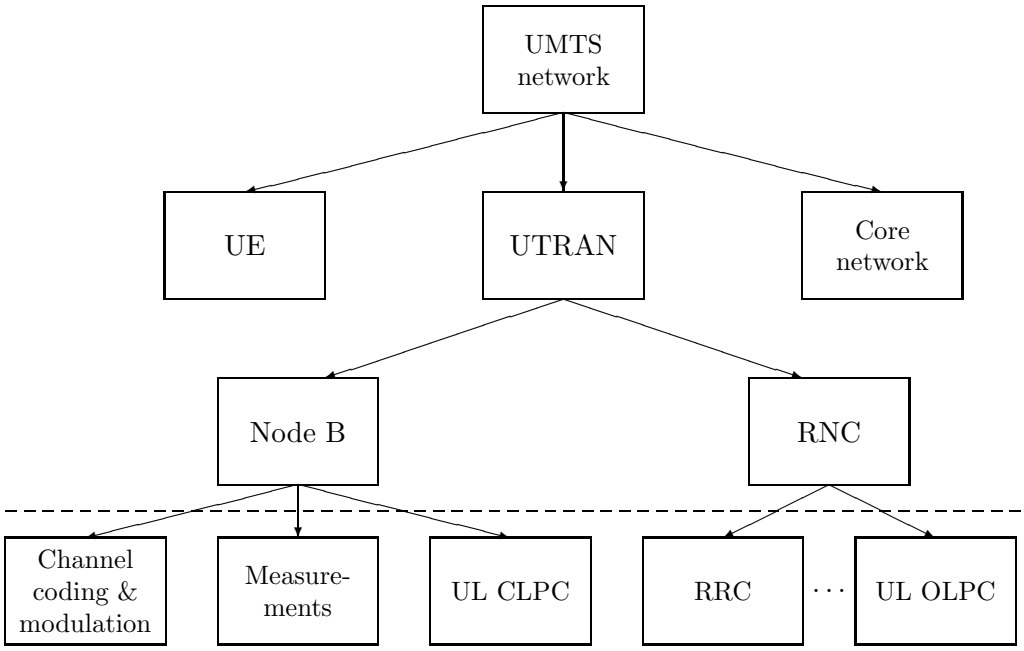


Figure 3.5: Hierarchical decomposition of the UMTS network with one user.

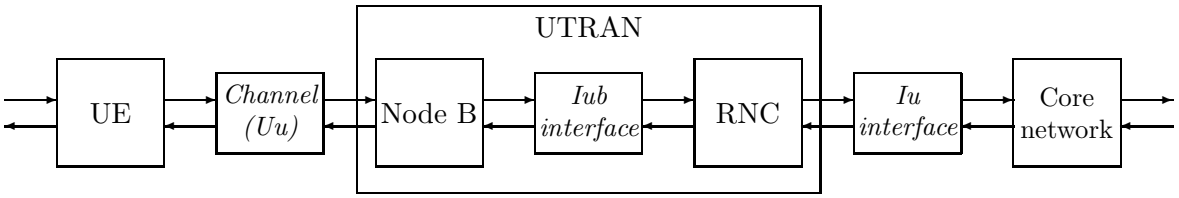


Figure 3.6: Systemic representation of the UMTS network with one user.

inverse in uplink, decoding and demodulation), measurement of various connection statistics, and part of the uplink power control.

A possible model for a Node B in the spirit of Chapter 2 is shown in Figure 3.7. [9] In this model, we isolate *Receiver* and *Modulator* from the rest of the system in order to emphasise once again that those are hybrid systems that transform the continuous radio signal into a discrete one and vice-versa. We assume that the component responsible for performing various measurements communicates directly with the CLPC component, as the latter utilises directly one of the basic channel measurements provided by the former, which is the Signal to Interference Ratio (SIR). This shall be further explained in Chapter 4. All other interactions pass by the system's internal memory, and all the operations are managed by a controller Q.

Finally, observe also that the two subsystems *Coding* and *Decoding* represent, in fact, a number of parallel coding (respectively decoding) chains — one for each transport channel. An example of such a chain for the High Speed Downlink Packet Access (HSDPA) service is given in Chapter 5.

[9] Observe that this model does neither rigorously model a Node B, nor perfectly comply with the definition of a system as given in Chapter 2. One should, however, keep in mind that the goal here is to illustrate the latter while reflecting as close as possible the *functional* nature of a real system, which is the Node B.

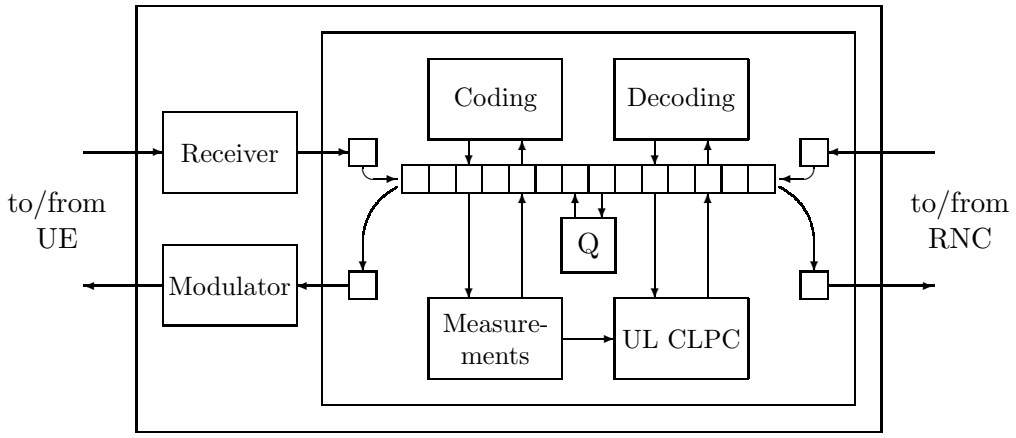


Figure 3.7: A systemic model of Node B.

3.3.2 Multiple users case

Let us now turn to the general case where the UTRAN consists of several RNS, which in their turn can have several Nodes B each, etc. This situation is illustrated in Figure 3.8. Although UE do not normally belong to the UTRAN, here we include them into the hierarchical decomposition for reasons that will become apparent when we consider a systemic model.

Indeed, contrary to the single user case that we have considered in the previous section and where we assumed one instance of each type of subsystem (RNS, Node B, etc), here we have a set of such components that can vary in size. This is especially clear when one considers UE present in a given cell. The set of UE in a cell is affected by two processes: UE connecting to the network or disconnecting from it on one hand, and various types of handover, when a given UE migrates from one cell to another. Thus our model has to take in account this varying nature of the network.

Figure 3.9 shows a backbone for a model of the part of the network comprising the UTRAN and the UE domain. In this model each subsystem is representing a type of network nodes, rather than a particular physical object. [10] We suppose here that the system in a higher level of hierarchy (for example *Cell* for *UE* here) keeps in memory a list of objects in the lower level, and calls the corresponding subsystem when action has to be taken on a particular physical — in a usual sense of the word, rather than that of our classification of systems — instance of the latter. In reality, the picture is of course slightly different. For example, even though the UTRAN does, indeed, have a database with all the RNC in the network, it only has the information about these RNC, while the information about the Nodes B is kept in the RNC, etc.

Another important difference between this model and a real-life system is that the former contains several purely logical components (*RNS*, *Cell*, ...) equipped with memory, which is obviously impossible in the latter. In particular, we assume a logical component *UE domain* that keeps a list of all cells. This assumption leads us to introduce the *Controller* component that allows to update this list whenever a new cell is added to the system. Instead, in real-life systems, when a new Node B is installed all the corresponding processing is performed in the UTRAN, as the corresponding UE domain does not exist yet.

In spite of all the differences that have been discussed above, this approach has two important advantages. First of all, it allows us to have a static model for a dynamic network, i.e. we can

[10] Here, one could draw a parallel with the class/instance duality in object oriented programming.

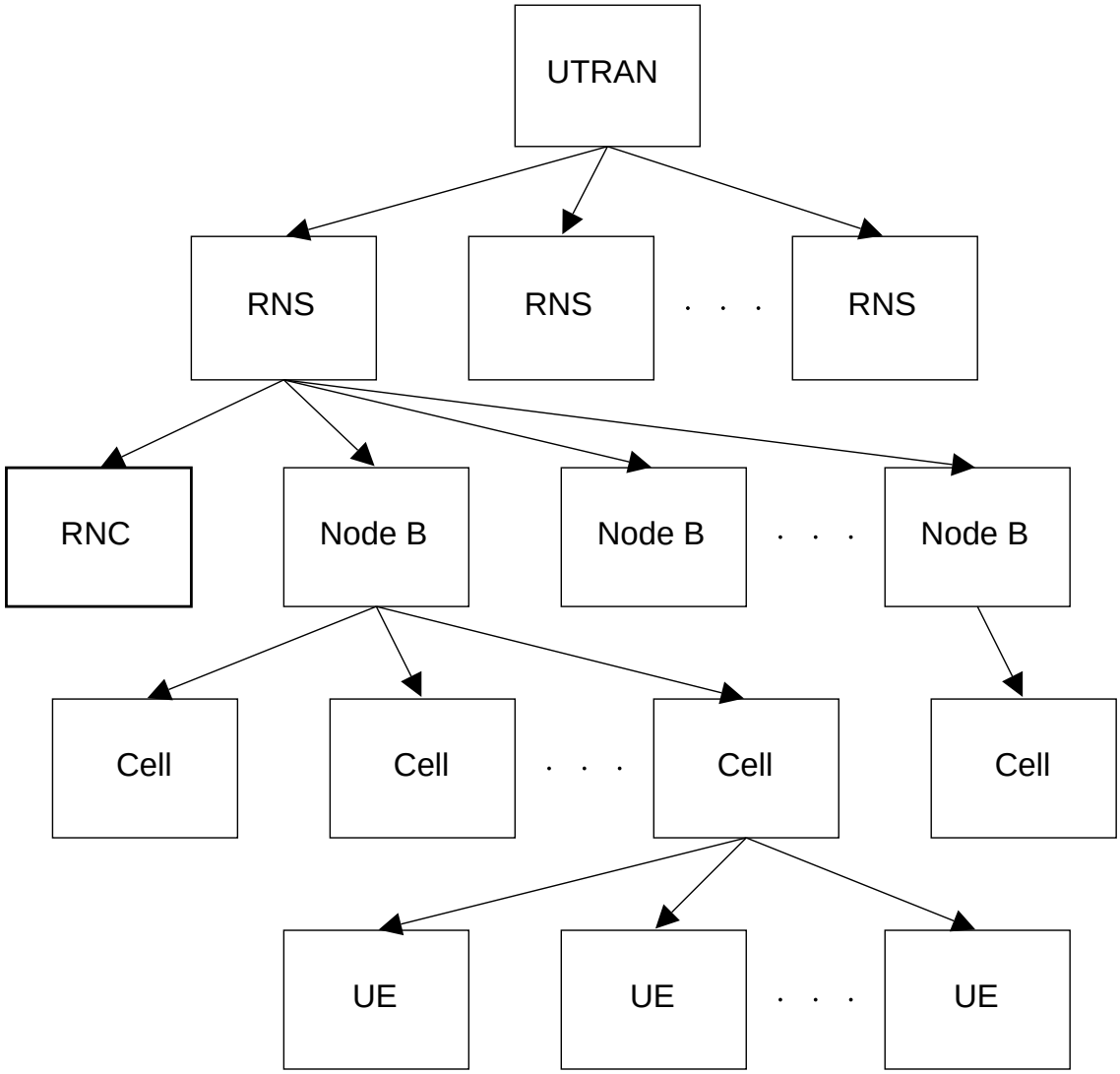


Figure 3.8: Hierarchical decomposition of a general UMTS network.

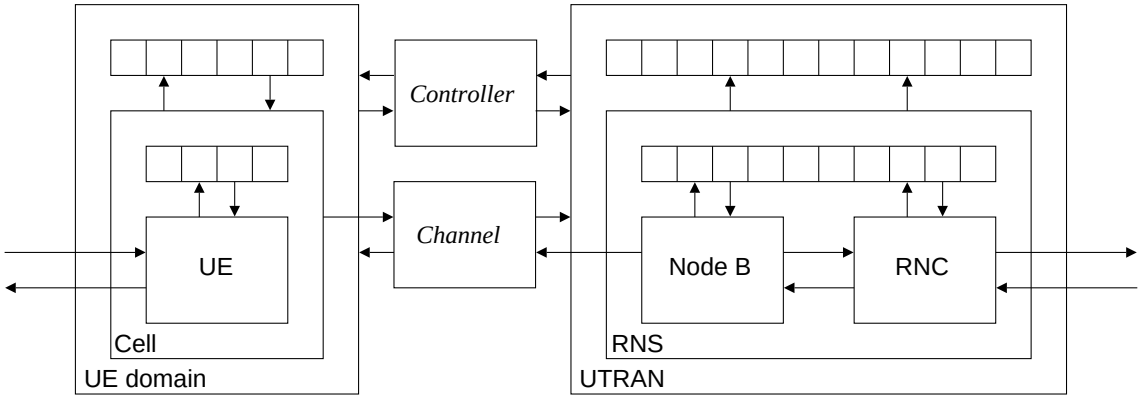


Figure 3.9: A backbone of a systemic model of UMTS in a general case.

dynamically add and remove nodes from the network without having to change the underlying model. On the other hand, and partly as a consequence of the previous observation, if we were to consider a network simulator, this would be essentially the only possible choice, as all the system's memory would have to be contained in that of the simulating device! Finally, we have to stress once again that, provided that this model is carefully developed, it can be *functionally* equivalent (see Definition 2.17 in Section 2.2.1) to (a part of) the real-life system.

3.4 Discussion

In this chapter, we have presented an overview of the Universal Mobile Telecommuncations System (UMTS), which will serve as a framework for the examples of the following chapters of this thesis. In order to better situate this system in a general telecommunications context, we have also accompanied this overview by a brief introduction to the Global System for Mobile Communications (GSM), by which the UMTS has been largely inspired.

We have concentrated on two particular aspects of both of these systems, which we refer to in the sequel: their hardware architecture and the multiple access principles utilised.

In Section 3.3, we have considered two approaches to modelling UMTS in the spirit of Chapter 2. The theory presented in this chapter being in its rudimentary stage, these approaches were limited to the decomposition in subsystems and the analysis of different types of subsystems involved in this decomposition (that is physical, logical, and hybrid ones).

The discussion of Section 3.3 allows us to define the range of the systems to which our model is applied. Indeed, as we have observed in this section, the main problems of modelling the UMTS infrastructure by a system in the sense we adopt comes from its network nature, that is from the fact that elements of the system can be added and removed at will. This observation suggests the following restriction: we shall consider in our model only *complex industrial systems that admit a finite description.*

Chapter 4

Subsystem Level: Power Control

Let us emphasise now one particular functionality of the UMTS networks — the power control. We have seen in the previous chapter that this functionality is mainly shared between the RNC, the Node B, and the User Equipment (UE). Thus in the straightforward hierarchical decomposition, components responsible for various aspects of power control form parts of the decomposition of different subsystems. This reflects well the aspect of the industrial systems' engineering that is concerned firstly with determining the components necessary to construct a system (here RNC or Node B) and, secondly, with constructing each separate component. However, the latter implies also another not less important process, which is analysing a particular functionality in order to calibrate the subsystems that realise it. Indeed, a functionality that is performed by several inter-operating components — and the power control provides here a perfect example — can nevertheless be sufficiently independent from the rest of the system for it to be studied as a separate system.

This brings forward one of the advantages of the recursive model of systems as introduced in Chapter 2 (see in particular Theorem 2.19 and the discussion immediately after). Indeed, suppose we have several systems each decomposed as in Section 2.2.1 into a number of subsystems, and co-operating to provide, among others, a given functionality. For each of these systems, we can isolate the components participating in this particular functionality and put them all together to obtain a virtual system that only models this functionality in question. This virtual system can then be analysed independently of the rest of the original systems, thus allowing for better understanding and calibration of the components involved.

In this chapter we will illustrate this idea on the example of the power control mechanism in UMTS. We start, in Section 4.1, by giving an overview of this mechanism, then, in Section 4.5, we describe a virtual system corresponding to the uplink power control, and finally, in the following sections, we give an example of a detailed analysis of one of the aspects of the latter, namely we consider the Uplink Outer Loop Power Control (UL OLPC).

4.1 Overview of the power control

From a very general point of view, power control consists simply in adapting the transmit power of either a UE or a Node B depending on current radio transmission environment. It can be, therefore, considered as a system taking on input a suite of measures corresponding to the current link state and provides on output a value of transmit power sufficient to maintain the required quality of service (QoS).

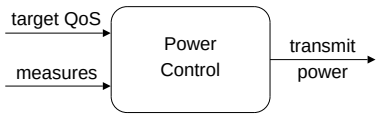


Figure 4.1: Power control: high-level system.

The importance of power control in W-CDMA networks is difficult to overestimate. Indeed, contrary to the second generation systems, such as, for example, the widely implemented GSM system, all UMTS users share the same frequency band. Signals from and to different users are separated by codes (see Section 3.2.2 of Chapter 3). In downlink, these codes are synchronised, and consequently orthogonal. Thus, the downlink capacity is no longer limited by the number of frequencies that can be allocated in a given cell. Instead, in good radio conditions (ideal scenario here corresponds to all mobiles being in the *line of sight* from the Node B), it is primarily determined by the power available for transmission at the Node B. In general, multiple propagation paths desynchronise the codes, which results in their loosing orthogonality. However, available transmit power stays one of the important limiting factors of the downlink network capacity. It is also important to notice that good power control in downlink allows to minimise the interference to adjacent cells.

Considering the uplink connection from the mobile to the Node B, one can observe that, once again, efficient power control is absolutely indispensable. First of all, the battery life of a mobile is a universally accepted critical issue related directly to the end-user experience, whereas the most power consuming activity of a mobile handset is transmitting the radio signal. Thus in order to optimise the battery life, it is essential to maintain the transmitting power at the lowest possible limit.

At the same time the uplink capacity of a given cell is most often determined by a limit imposed on the so-called *noise rise*, which is defined by $NR = P_r/I_0$, where P_r is the total received power at the Node B, and I_0 is the floor noise level. Thus keeping the mobiles' transmit power at minimum is one of the important factors allowing to improve the network capacity.

In uplink, it is impossible to perfectly synchronise PN sequences separating different users and especially so in presence of multiple propagation paths. Therefore, a certain amount of interference between different users cannot be avoided. The interference of a signal from the mobile close to the Node B can prevent a signal from another mobile further in the cell from being decoded correctly. This is called a *near-far* effect.

This effect can be clearly understood by imagining two mobiles transmitting in the same cell and at the same power level. Suppose, however, that one of these mobiles is very close to the Node B, while the other one is located at the border of the cell (see Figure 4.2). It is clear that the signal received by the Node B from the first mobile is much stronger than that of the second one, and thus the level of the produced interference can be too high for the second mobile's signal to be decoded correctly. More generally, one mobile close to the Node B can effectively prevent all other mobiles in the cell from transmitting any information. It is now sufficient, in order to understand the importance of an efficient power control mechanism, to observe that in the example above the near-far effect can be eliminated if the first mobile reduces sufficiently its transmit power. In the general case, this corresponds to each mobile's transmitting at the least power level sufficient to maintain the quality of service necessary for its connection.

Figure 4.2: Illustration of the *near-far* effect.

4.2 Measures involved in the Power Control

There is a large variety of measures that can be used to perform the power control. These can be generally separated in two groups: measures reflecting the quality of service, such as *bit*, *block*, or *frame error rates* (BER, BLER, and FER correspondingly); and those reflecting the channel quality, such as *signal to interference ratio* (SIR) or *bit energy to interference spectral density ratio* (E_b/I_0).

The following relation holds between the two last measurements

$$\frac{E_b}{I_0} = SIR\,\frac{W}{r}\,,$$

where W is the transmission bandwidth in Hertz and r is the data rate in bits per second. Thus, assuming that the data rate is fixed, this relation becomes a simple linear dependency.

On the other hand, the relation between BER, BLER, and FER is much more complicated, and we shall only state here that it is strictly monotonic. That is, when for example the BER increases, so do the BLER and the FER. The relation between any of these three rates and the channel quality measurements is inverse monotonic. For instance, increasing the SIR decreases the BER and, consequently, the BLER.

It is important to observe here that these different measures correspond to different "levels of abstraction" as to the amount of information they convey. Indeed, when one consideres one block, which is most often the case as far as power control is concerned, the BLER corresponds to the binary status of CRC check (block received correctly or not), whereas BER indicates how many bits in this block were decoded with errors. In its turn, the BER summarises the low level information available to the decoder (such as, for instance, the log-likelihood ratio of each bit), and so on.

Thus, it is clear that the performance of power control algorithms depends on the particular quality indicator they refer to: the more low level is the quality indicator, the better performance can be expected from the corresponding algorithm. The downside is, however, that obtaining lower level quality indicators is more complex, and so are the algorithms that utilise them. This results in the implementation of such algorithms being more expensive in termes of both space and time complexity, resulting in more expensive circuits and, *a fortiori*, equipement.

4.3 3GPP power control

A number of approaches to power control problem have been studied, which can be generally separated in two groups: centralised algorithms, and distributed ones. The former assume full knowledge of the system and derive an optimal power assignment vector, i.e. an optimal value of transmit power for each mobile in the cell, provided, of course, that such a vector exists. [1] In the latter approach, an independent algorithm runs on each mobile unit and adjusts its transmitting power in such a way as to converge to the above optimal value without requiring any global knowledge of the system.

An important advantage of distributed algorithms is that they induce much less signalling overhead in the network and, in particular, can adapt more efficiently than the global solutions to the changes in the environment.

Starting from Section 4.5, we shall limit ourselves to the uplink power control as it is implemented in the actual 3GPP specifications. In this approach, the only information required is that concerning the link between the Node B and the moble in question, and therefore it should rather be classified as a distributed one, even though some algorithms involved are running on the serving RNC of the corresponding cell.

Power control in 3GPP consists essentially of three so-called *loops*. Figure 4.3 illustrates

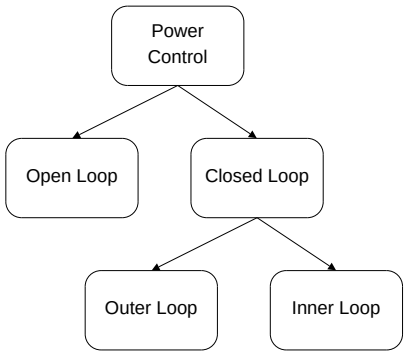


Figure 4.3: Types of power control in 3GPP.

this typology, although it might seem to be in discord with the previous statement. First of all, there are two main types of power control: Open Loop Power Control and Closed Loop Power Control (CLPC). Both these types have the same systemic structure as the one shown in Figure 4.1 of the previous section, and one should consider them as connected "in parallel" (see Figure 4.4). The difference is that Open Loop does not rely on any feedback information from the system, and therefore can be utilised while the connection is being initialised. Closed Loop requires feedback information on the reception quality, which allows it to provide better performance.

In its turn, Closed Loop is subdivided into Outer Loop (OLPC [2]) and Inner Loop. The former sets a target reception quality according to the required quality of service, while latter adapts the transmit power in order to match the target set by the Outer Loop (see the following sections for more details). Keeping the above terminology, one can say that these two loops are connected "in series" (see again Figure 4.4).

[1] In this context, it is said that power control problem is *feasible*, if there exists such an optimal power assignment vector that every mobile can effectively support its assigned transmit power (see Section 4.4).

[2] Observe that OLPC is an abbreviation for Outer Loop Power Control, and not for Open Loop Power Control.

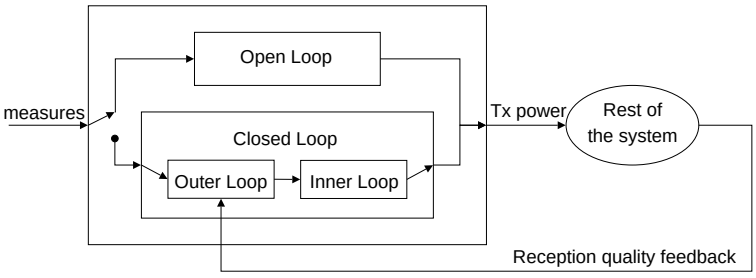


Figure 4.4: Open Loop vs. Closed Loop.

From the above discussion, one can see that there are only three algorithms ivolved. Moreover, Closed Loop is often identified with Inner Loop. One speaks then of Closed Loop *controlled* by the Outer Loop. This is also the terminology that we shall use in the rest of the chapter.

4.3.1 Open Loop Power Control

The *Open Loop Power Control* is used in the Frequency Division Duplex (FDD) mode, and only during the transmission initialisation phase. To perform this type of power control, the transmitting entity measures the Signal to Interference Ratio (SIR) of the signal received from the receiving one, compares it to a given target, and adjusts its proper transmit power accordingly. In other words, each participating entity estimates the propagation conditions of its connection to the other one from the signal received on the reverse link.

This procedure allows to quickly estimate the required level of transmit power when the connection is being initialised. During this phase, the connection is only being established, and thus the mobile and the Node B do not have the possibility to exchange information about quality the received signal.

The main inconvenience of this method is that the uplink and downlink transmissions are performed on different frequencies, implying a low correlation of fading effects for these two connections. Thus the power control decision is not sufficiently reliable.

4.3.2 Closed Loop Power Control

Once the connection has been established, both the mobile and the Node B can report their corresponding received signals' qualities, and thus the power control decisions based on this reporting reflect the actual channel situation much more accurately than those of the Open Loop PC. For this reason the *Closed Loop Power Control* (CLPC) is used at all moments other than connection initialisation.

Every time slot (TS):
  $SIR_{TS} \leftarrow$ mean of the received SIR over TS;
  Modify SIR_{TS} with TPC commands not yet taken into account;
  if ($SIR_{TS} > SIR_{target}$)
    send a "down" TPC command,
  else
    send an "up" TPC command.

Algorithm 4.1: 3GPP algorithm for Closed Loop Power Control.

a b

Figure 4.5: Closed Loop (a) and Outer Loop (b) components of the Power Control system.

The CLPC proceeds as following (see Algorithm 4.1). The receiving entity measures the SIR level, compares it to the *target SIR* and, according to the result of this comparison, sends a power control command to the transmitting entity to either increase or decrease it's transmit power (this is summarised in Figure 4.5-a). The latter, on reception of such a command, modifies its transmit power by a fixed step Δ, which according to 3GPP specifications can be chosen between 1, 1.5, and 2 dB [3].

The Closed Loop PC is sometimes called *Inner Loop* in order to emphasise its relation to the Outer Loop PC (see Section 4.3.3), and also *Fast Power Control* as opposed to the one used in GSM. Indeed, to counteract fast fading effect the CLPC has to be applied sufficiently often. In practice it is applied after transmission of each slot, that is at the rate of 1.5 kHz, whereas in GSM power control is applied at the rate of 2 Hz.[4]

4.3.3 Outer Loop Power Control

As it has been mentioned in the previous sections, the role of the *Outer Loop Power Control* is to maintain the target SIR at the appropriate level to provide the required quality of service [5] (see Figure 4.5-b for the corresponding systemic diagram). Let us first of all explain why this loop is necessary, that is why does the target SIR have to be adjusted at all.

We have seen, in Section 4.3.2, that the transmitting entity uses a fixed step to adjust its transmit power under the Closed Loop PC. Thus its ability to converge to the ideal power level that would provide the necessary received SIR depends considerably on the propagation conditions. This property is reflected by the standard deviation of the transmit power over the transmission period.

It is clear that, when the mobile's speed is low, so is also the channel variation. Therefore, in this case, the CLPC efficiently compensates for the fading dips, and the standard deviation of the transmit power is low. At high speeds, channel conditions vary rapidly, and the CLPC compensates for fading dips less efficiently, which degrades the quality of service. To compensate for this effect the target SIR has to be increased correspondingly.

If the target SIR were to be constant, then it has to be calculated for the worst case scenario, and thus a mobile experiencing comparatively good radio conditions would be transmitting at an excessively high power level. Sampath, Kumar and Holtzman in [82] showed, for example, that for a frame error rate (FER) of 1%, if the target SIR was calculated on a basis of 2 dB standard deviation for the CLPC, then for a user experiencing a standard deviation of 1 dB the loss in transmitting power would be approximately 5.4 dB.

Another reason for adjusting the target SIR is that the received SIR is measured before channel decoding and therefore it reflects the total energy over all propagation paths. However,

[3] 3GPP specifications also contain a slight modification of the described algorithm that allows to emulate smaller steps by applying power control less often. However, this algorithm is beyond the scope of this dissertation.

[4] Such an important difference is explained by the fact that fast and efficient power control is essential in W-CDMA networks to distinguish different users' signals (see Section 4.1), whereas in GSM these signals are modulated on different frequencies, and thus the impact of power control is less crucial.

[5] This quality of service is most often defined in terms of BLER.

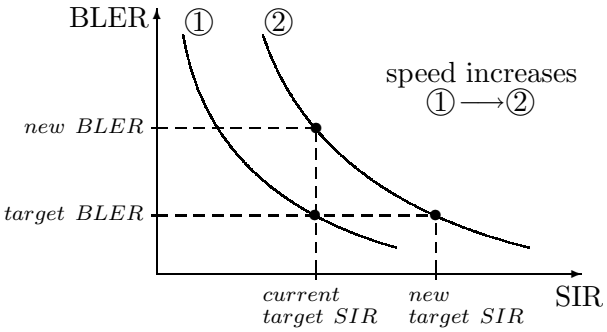


Figure 4.6: Effect of increasing UE speed on target SIR.

as the delays are not the same on all paths, there is a non-zero inter-symbol interference (some times also called *auto-interference*; see Appendix B, cf. also [35, 40]), which implies that only a certain amount of received power can be effectively used for decoding. When the characteristics of the multipath environment change so does the amount of the inter-symbol interference. The "useful" proportion of power is difficult to estimate, and therefore it is easier to adapt the target SIR accordingly.

To illustrate the above argument, let us consider a single cell system with N mobile units. The mobile i transmits at power p_i, and the channel attenuation to the Node B is denoted by g_i. The uplink SIR requirement is then expressed by

$$\frac{\xi_k g_k p_k}{\sum_{i=1, i\neq k}^{N} g_i p_i + \eta_k} = \xi_k \gamma_k \geq \gamma_k^t \,, \tag{4.1}$$

where γ_k is the measured received SIR, η_k is the Gaussian noise at the receiver, $\xi_k \in (0,1)$ represents the influence of the inter-symbol interference, and γ_k^t is the threshold value required to maintain the QoS. If we discard the inter-symbol interference, the relation (4.1) states simply that the received SIR has to be superior to a given target SIR. In general case, this constraint can be translated as

$$\gamma_k \geq \gamma_k^t / \xi_k \,, \tag{4.2}$$

and the target SIR is now defined by the right-hand side of (4.2), which varies when the channel conditions change.

Example 4.1 (Typical application of OLPC)
We consider a situation where a mobile in communication with a Node B rapidly increases its speed, thus causing a deterioration of radio conditions. Figure 4.6 illustrates schematically the typical BLER to target SIR dependencies before and after acceleration. One can see that, if the target SIR was to remain constant, this would entail an increase in the current BLER. Thus to maintain the BLER level, it is necessary to adapt (increase in this particular case) the target SIR, which is exactly the role of OLPC. ■

4.4 State of the art

As we have seen in the previous sections, power control is one of the key issues in network resource management. Consequently, it is extensively studied in the literature, and particularly so in the past two decades.

A vast majority of this research is, however, concentrated on the so-called SIR-balancing problem [7, 98] and its variants. Here, one takes a "snapshot" approach to power control, by assuming that the derived algorithm provides an acceptable power assignment faster than the the corresponding link attenuations evolve to invalidate it.

For convenience, we shall formulate this problem for uplink direction (for downlink the formulation is very similar). In a rather general case this can be done as below (see for example [77]). A variant, which we have already mentioned in Section 4.3.3 (cf. (4.1)), accounts for the auto-interference effect [35], i.e. for the fact that not all received power can be effectively utilised for decoding.

Consider N mobile stations transmitting over the same channel including the intracell and intercell users. Define a base station assignment function $b(i)$ so that $k = b(i)$ if mobile i is served by base station k. Denoting by g_{ji} the link attenuation from mobile i to base station j, and by p_i its transmit power, the uplink SIR requirement for user i can then be expressed as

$$\gamma_i = \frac{g_{b(i)i}\, p_i}{\sum_{j=1, j\neq i}^{N} g_{b(i)j}\, p_j + n_i} = \frac{p_i}{\sum_{j=1, j\neq i}^{N} \frac{g_{b(i)j}}{g_{b(i)i}}\, p_j + \frac{n_i}{g_{b(i)i}}} \geq \gamma_i^t \,, \tag{4.3}$$

where γ_i is the SIR at the receiver for the signal of this user, γ_i^t is his uplink SIR requirement, and n_i is the receiver noise power. By defining the vectors $\mathbf{p} = \{p_i\}$ and $\mathbf{n} = \{\gamma_i^t\, n_i / g_{b(i)i}\}$ and the matrix $\mathbf{H} = \{H_{ij}\}$ with elements $H_{ij} = \gamma_i^t\, g_{b(i)j} / g_{b(i)i}$ when $i \neq j$ and $H_{ii} = 0$, we can rewrite (4.3) in matrix form:

$$(\mathbf{I} - \mathbf{H})\mathbf{p} \geq \mathbf{n} \,, \tag{4.4}$$

where $\mathbf{I}$ denotes the identity matrix, and the inequality holds componentwise. A minimum-power solution corresponds to the case where (4.4) is satisfied with equality.

Definition 4.2 *The SIR-balancing problem above is said to be* feasible *if there exists a non-negative power vector* $\mathbf{p}$ *satisfying (4.4).*

The optimal power vector p^*, such that the equality is met in (4.4), exists if the largest eigenvalue of the matrix $\mathbf{H}$, denoted by $\rho(\mathbf{H})$, is less than or equal to one [99, 100]. Although power assignment provided by p^* ensures optimal performance for all users,[6] it entails an extraordinary signalling overhead in the network and therefore is not practical. This *global* solution of SIR-balancing problem serves rather to obtain a theoretical bound for other algorithms.

A number of iterative *distributed* algorithms have been studied that allow each user to update his transmit power based only on local measurements and his own channel attenuation [21, 33, 72, 92, 99]; Yates, in [96], presents a framework for studying such algorithms, also considering different possible user-to-base station assignment policies. Another good overview is proposed in [40] with a rich reference base, whereas the latest contribution to the bibliography is probably the Rintamäki's PhD thesis [78].

As we have already presented in Section 4.3, the power control in UMTS is organised in two cascading loops: the inner loop using Algorithm 4.1 to align the received SIR of a given user to target one (see Section 4.3.2), while the outer loop updates this target according to the channel conditions.

Algorithm 4.1, initially proposed in [11] and utilised in the inner loop, reduces to minimum the signalling between the base station and the mobile: one bit is used per transmitter power

[6] In case, where the SIR balancing problem is not feasible, a removal strategy must be employed to remove users for which the necessary QoS cannot be achieved [10, 36, 97, 99].

control (TPC) command, which is, however, repeated in some systems in order to increase protection.[7] Recall that this algorithm is essentially reduced to the transmitter's increasing or decreasing the transmit power by a fixed amount according to whether the received SIR is above or below the target one, which information is signalled from the receiver.

A number of modifications of this algorithm can be found in the literature optimising its performance, in particular, by reducing the oscillations around the target SIR or adapting the SIR modification step to improve convergence [8, 40, 74, 79, 80].

Regarding the outer loop, the conventional algorithm that we present in Section 4.6.1 has been proposed in [82], and consists in increasing the target SIR by a large step, when a block transmission error is detected, and decreasing it by a smaller one on correct transmission. The relation between the two steps defines the average resulting BLER.

This algorithm will be discussed in more detail in Section 4.6, where we argue that, although rather efficient for services with target BLER of order 10^{-2} and higher, for those with higher QoS requirements it becomes inadequate due to the inherently limited information conveyed by the simple CRC check. In other words, for high quality services, too few errors occur for a reliable estimation of BLER to be possible.

The outer loop power control is not fixed in 3GPP specifications, which makes it into a field where different equipment suppliers may compete. For this reason, most algorithms for outer loop are confidential. Nevertheless, some contributions can also be found in the public domain [38, 41, 49, 52]. One of the possible improvements is represented by the so-called double-loop algorithms, which are discussed to some extent in Section 4.6.3.

Finally, another trend in QoS optimising is the *multi-user detection*. Although this technique concerns primarily the interference suppression by way of improving receiver structure, it can be combined for best results with efficient power control, as for example in [91].

4.5 Systemic view of the uplink power control

As it has been already mentioned in the opening of this chapter, the power control functionality is distributed over several UMTS subsystems. On the network side, for example, the Open Loop and the Inner Loop PC are performed in the Node B, whereas the Outer Loop PC is implemented in the RNC. The latter allows, for example, the OLPC to be active even during soft handover when mobile is switched from one Node B to another, thus avoiding unnecessary instances of the initial convergence phase of the algorithm, which is rather slow. The Closed Loop PC, on the contrary, converges sufficiently fast — justifying the name of Fast Power Control —, and therefore can be realised in Node B, which increases its efficiency by avoiding the signalling delay between the RNC and the Node B. (The target SIR determined by the OLPC is signalled to the Node B over the so-called Iub interface, thus introducing a considerable delay before it is applied by the CLPC.)

More generally speaking, various aspects of power control are distributed between the RNC (Uplink OLPC), Node B (Downlink Open Loop, Uplink CLPC), and User Equipment (Downlink PC). However, in order to properly model the whole functionality, one has to consider also the delay introduced by the Iub interface when signalling the target SIR from the RNC to the Node B, and the error in the TPC command due to its being transmitted over the radio channel without any error protection coding, as well as other components' influence — as always, the model grows with the precision of the results required.

[7] No error correction coding is applied to the TPC command in order to increase processing speed and, consequently, reduce the power control delay.

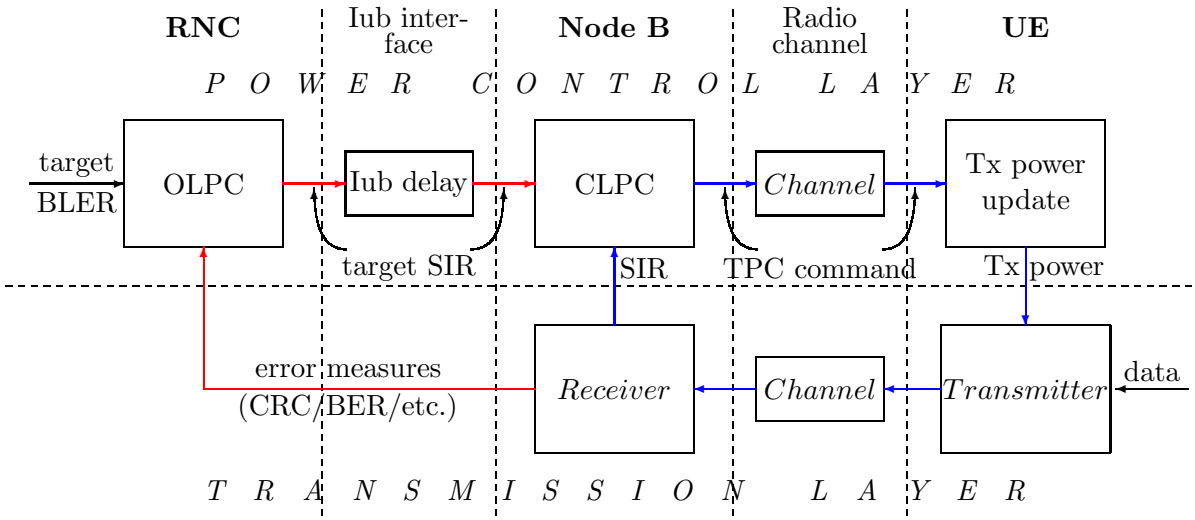


Figure 4.7: A systemic diagram of Uplink Power Control.

For the sake of clarity and simplicity, we shall now restrict ourselves even further by dropping the Outer Loop Power Control out of our global picture. Figure 4.7 shows a diagram representing a possible virtual system modelling the Closed Loop Power Control (here in the sense including the OLPC).

The diagram is separated, first of all, in two layers: power control layer, and transmission layer. Indeed, as it can be seen from the discussion above, power control from our point of view is a reactive system heavily dependent on the surrounding environment. When one speaks of a *loop* in our context, it is necessarily an action-reaction loop, where the action consists in adjusting continuously a particular set of parameters according to the environment measurements, constituting the corresponding reaction. For example, one can see that the blue arrows on the diagram, corresponding to the interactions involved in the Inner Loop PC, do, indeed, form a loop. Replacing the arrow, going from the *Receiver* to the CLPC component in the Node B, by the set of red arrows we obtain the Outer Loop PC.

Transversly, this diagram is broken up into five blocks corresponding to different subsystems of the original system in Chapter 3.

Let us briefly discuss various components shown in Figure 4.7. First of all, observe that the system is completely decomposed in subsystems, and therefore there is no need in internal memory or controller. The system can be viewed as having no output, and its input is essentially restricted to the target BLER. However, the data to be transmitted has to be provided on input for the sake of completeness. Similarly, one can consider the actual BLER to be the output of our system, although its only purpose in this context is to allow the analysis of the power control performance by comparing it to the target one. In any case, all the corresponding time scales are discrete, and thus on a higher abstraction level the system should be considered as a software one.

Recall now that the classification in Section 2.2.2, separating all systems in three classes — software, physical, and hybrid ones —, is entirely based on the nature of system's time scales. Thus we can conclude that the system in Figure 4.7 would be a software one, if it was not for the *Channel* component in the Transmission layer, and the two corresponding connections between *Radio Channel* on one hand, and *Transmitter* and *Receiver* correspondingly on the other. As it has already been mentioned in Chapter 3, the radio channel is best modelled by a continuous

system, as most of the analysis in the litterature (as for example in [75]) is based on continuous functions. Thus the *Transmitter* component is here a hybrid subsystem similar, although more complex, to the Simplified Radio Transmission chain in Example 2.27 of Section 2.2.4. The *Receiver* component is also a hybrid subsystem, symmetrical to the *Transmitter* one.

Observe that the component modelling the radio channel in the Power Control layer of the diagram has a different nature from that in the Transmission layer. Indeed, its only purpose is to model the possible error on the TPC command. This error is most often assumed to be negligible, and the TPC command can only take a small number of values (two or three depending on the particular system), therefore the underlying radio channel can be easily modelled by a purely software system that would modify the value of the transmitted command with a given probability. Both input and output time scales of this system are therefore discrete with a time step $0.67 \cdot 10^{-3} s$, corresponding to a rate of 1 500 Hz, at which the CLPC is performed.

Properly speaking, the Power Control as such is modelled by the components in the upper layer of the diagram. The separation in two layers illustrates one of the fundamental differences between the standard approaches to system analysis: theoretical analysis and simulation. Theoretical analysis assumes a certain pattern for all external influences, expressed by a system of differential equations, a probability distribution, or other similar means, in order to produce an analytical expression for the behaviour of the system in question. Simulation, at the same time, consists in approximating the behaviour of all participating systems, including the external influences, to obtain purely numerical results predicting the expected performance. In the following sections we illustrate the former by analysing several algorithms for the Uplink Outer Loop Power Control.

4.6 Outer loop power control analysis

The algorithm for CLPC is, by now, rigorously specified on the User Equipment (UE) side. Although, on the network side, there is more choice as to the decisions a Node B has to take on receiving a power control command, the general principle is given by Algorithm 4.1. At the same time the OLPC algorithm is left completely open for RNC manufacturers. Therefore, most industrial research is concentrated on the Outer Loop Power Control.

4.6.1 Sawtooth algorithm

Recall from Section 4.1 that the role of the OLPC is to maintain the target SIR at correct level depending on the current radio conditions. The basic algorithm that serves as a reference for most studies of OLPC is the well known *Sawtooth* algorithm (see Algorithm 4.2). [8] The principle of Sawtooth is very basic and resembles to that of the CLPC algorithm. It consists in evaluating the quality of the received signal, and adjusting the target SIR accordingly.

For each received block:
  if (CRC fail)
    $SIR_t := SIR_t + \delta_{up}$
  else
    $SIR_t := SIR_t$ - δ_{down}

Algorithm 4.2: Sawtooth algorithm.

[8] This algorithm in a slightly different form was proposed in [82].

In general, an OLPC algorithm is governed by a set of parameters that can be split in the same manner as in Section 4.1 into QoS parameters (BER, BLER, etc) and channel quality parameters (SIR, E_b/I_0, etc). It can be said that the goal of the OLPC is to implicitly *measure* the channel quality and set the target SIR correspondingly. Therefore, the algorithm's efficiency depends particularly on its channel quality related parameters. Thus, an important problem in analysing an OLPC algorithm is to establish a relation that, given the QoS requirements, allows to determine the correct values for the latter.

In the case of Sawtooth, the QoS is determined by the BLER with the corresponding requirement expressed in terms of a so-called *target BLER*. This algorithm has two parameters δ_{up} and δ_{down} that directly affect its performance. However, neither of these parameters has an explicit relation with the quality of service. It is imperative therefore to establish a relation between δ_{up}, δ_{down}, and target BLER. This can be done with the help of the following theorem that we prove in Section C.1 of Appendix C.

Theorem 4.3 *Let $\{X_n\}$ be a stochastic process on $\mathbb{R}$ defined by setting $X_{n+1} = F(X_n)$, where*

$$F(X) = \begin{cases} X + a & \text{with probability } p(X) \\ X - b & \text{with probability } q(X) \\ X & \text{with probability } 1 - p(X) - q(X)\,, \end{cases} \tag{4.5}$$

with $a, b > 0$ and $p(x) + q(x) \leq 1$ for all $x \in \mathbb{R}$. Suppose also that this process converges to a stationary distribution π. Then, denoting by $\mathbb{E}[p(x)]$ and $\mathbb{E}[q(x)]$ the expectations in stationary distribution of the probabilities of an upwards and downwards steps correspondingly, we have the following relation

$$a\,\mathbb{E}[p(x)] = b\,\mathbb{E}[q(x)]\,. \tag{4.6}$$

Proposition 4.4 *In order for Sawtooth algorithm to converge to a given target BLER p, it is necessary that its parameters δ_{up} and δ_{down} satisfy the following relation*

$$\delta_{up} \cdot p = \delta_{down}(1 - p)\,. \tag{4.7}$$

Proof. Observe that the evolution of target SIR controlled by Sawtooth algorithm can be described as a stochastic process satisfying the conditions of the above theorem. Indeed, let us denote by $X_n \in \mathbb{R}$ the target SIR on reception of the block $n + 1$. Sawtooth then defines the next value of target SIR by setting

$$X_{n+1} = \begin{cases} X_n + \delta_{up} & \text{with probability } p(X_n) \\ X_n - \delta_{down} & \text{with probability } 1 - p(X_n)\,, \end{cases}$$

where $p(X_n)$ is the value of BLER when the received SIR is equal to X_n, i.e. the probability that block $n+1$ is not decoded correctly. Assuming that this algorithm converges, we can easily deduce that, in terms of Theorem 4.3, we have $\mathbb{E}[q(x)] = \mathbb{E}[1-p(x)] = 1-\mathbb{E}[p(x)]$. Substituting target BLER for $\mathbb{E}[p(x)]$ in (4.6) we obtain the desired relation. ∎

In [82] and often in practice, equation (4.7) is simplified to

$$\delta_{down} = \delta_{up} \cdot BLER_{target}\,, \tag{4.8}$$

as target BLER is often negligible compared to 1. This provides a simple relation between two parameters. The task of optimising the algorithm's performance is now reduced to determining the optimal value for δ_{up}, which can be performed by simulation.

– o –

In Sawtooth, BLER serves to evaluate the quality of the received signal. This approach has an advantage of using the same measure for both controlling the OLPC algorithm, and defining the Quality of Service requirements; it is a very good choice for services with high error tolerance such as packet services or voice connection. Indeed, for services requiring a BLER of 1% and higher, errors occur sufficiently often for the algorithm to be able to adapt to channel variations, and also to converge rapidly at the initial phase of the connection. On the other hand, for services requiring lower BLER, Sawtooth algorithm is no longer capable of producing satisfactory performances. For example, taking $\delta_{up} = 0.25\ dB$, and assuming a target BLER of 10^{-4} (typical target BLER for video streaming), we obtain with (4.8) a value of $2.5 \cdot 10^{-5}$ for δ_{down}. This means that in order to compensate for an excess of 1 dB in the initial transmit power one would have to wait for forty thousand blocks, i.e. at least eighty seconds of connection time![9] Morover, the following example shows that, even at the ideal value of SIR, Sawtooth faces another problem — that of stability.

Example 4.5
Suppose again that we use Sawtooth to perform the power control on a service with target BLER $p = 10^{-4}$, and that the algorithm has converged to an ideal value s of the target SIR. Suppose further that we fail to decode the block number n. According to Algorithm 4.2, the target SIR for the following transmission is set to $s + \delta_{up}$. Thus the target SIR becomes higher than the required value, and approximately $1/p = 10\,000$ blocks have to be received correctly in order for the target SIR to get back to the ideal value (cf. equation (4.7)). ■

Note 4.6 Observe that the use we make here of the word "stability" is somewhat abusive. Indeed, if we assume constant radio environment, the trajectory of the transmit power defined by Sawtooth is rather stable in the sense that it oscillates in a certain fork around the ideal value. When we decrease the target BLER, this affects this fork and, more importantly, the cycle of these oscillations (cf. Example 4.5): the less is the target BLER, the longer is this cycle and thus less realistic the assumption of constant environment. Therefore, in a realistic situation with variable radio conditions the transmit power will have a stronger tendency to diverge from the ideal value. It is in reference to the latter phenomenon that we speak of instability.

The problem described in the example above arises from the fact that the presence of a single error does not provide any information as to the current BLER and, consequently, target SIR. The design of Sawtooth depends entirely on the statistical reasoning provided by Proposition 4.4.

To summarise, when using Sawtooth algorithm with services requiring lower BLER, we encounter two particular problems:

- the information, provided by the fact that a block is decoded correctly or not, is not sufficient to allow fast convergence to the ideal SIR value;
- even assuming that the algorithm has converged to (or was initialised with) this ideal value, its stability cannot be assured.

The next section presents a modification of Sawtooth that allows to resolve the latter problem without recurring to additional measurements, however it does not resolve the former one. Thus, other measures of the received signal's quality, providing more information on the current link state than the simple CRC check, are necessary for services with low target BLER and also more sophisticated OLPC algorithms.

[9] Assuming that the whole block is sent in one frame, and that the frame duration is the standard 2ms.

4.6.2 Adapting Sawtooth to increase stability

As it has been mentioned in the previous section, Sawtooth is a good choice of power control algorithm when the target BLER is of 1% and higher. One of the problems of applying it to services with lower target BLER is that a very long sample is necessary to estimate the current BLER, and thus it is rather difficult to ensure the algorithm's stability (cf. Example 4.5).

One way of addressing this problem is, therefore, to gather more information about the current BLER before taking a decision to increase or decrease the target SIR. In other words, we define a second power control algorithm (see Algorithm 4.3) that increases the target SIR when a sufficient number of block errors occur during a given sampling period, and inversely it decreases the target SIR when a sufficient number of blocks have been decoded correctly.

At the initialisation phase:
  n := 0
  N_{error} := 0

For each received block:
  n := n + 1
  if (CRC fail) $N_{error} := N_{error} + 1$

  if (n = N_b)
    n := 0
    if ($N_{error} > N_{up}$) $SIR_t := SIR_t + \delta_{up}$
    if ($N_{error} < N_{down}$) $SIR_t := SIR_t$ - δ_{down}
    N_{error} := 0
  end if

Algorithm 4.3: Sawtooth algorithm adapted to increase stability at lower values of target BLER.

In Algorithm 4.3, this is implemented by defining two thresholds N_{up} and N_{down}. The target SIR is updated every N_b blocks: it is increased if the number of block errors (N_{error}) exceeds N_{up}, and decreased if it is below N_{down}. Compared to Sawtooth, this introduces three new parameters that have to be taken in account by a relation equivalent to the one provided for Sawtooth by Proposition 4.4.

First of all, observe that, supposing that at step n the received SIR is equal to x, and the corresponding BLER is defined by a function $f(x)$, the probabilities $p(x)$ and $q(x)$ of increasing or decreasing the target SIR (cf. Equation (4.5) in Theorem 4.3) are defined correspondingly by

$$\begin{aligned} p(x) = P(N_{error} > N_{up}) &= \sum_{k=N_{up}+1}^{N_b} P(N_{error} = k)\,, \\ q(x) = P(N_{error} < N_{down}) &= \sum_{k=0}^{N_{down}-1} P(N_{error} = k)\,, \end{aligned} \tag{4.9}$$

where

$$P(N_{error} = k) = \binom{N_b}{k} f(x)^k \Big(1 - f(x)\Big)^{N_b - k}. \tag{4.10}$$

Substituting (4.10) into (4.9) we obtain a proposition equivalent to Proposition 4.4.

Proposition 4.7 *In order for Algorithm 4.3 to converge to a given target BLER p, it is necessary that parameters δ_{up} and δ_{down} satisfy the following relation*

$$\delta_{up}\,\mathbb{E}\left[\sum_{k=N_{up}+1}^{N_b}\left(\begin{array}{c}N_b\\k\end{array}\right)f(x)^k\Big(1-f(x)\Big)^{N_b-k}\right] = \tag{4.11}$$

$$= \quad \delta_{down}\,\mathbb{E}\left[\sum_{k=0}^{N_{down}-1}\left(\begin{array}{c}N_b\\k\end{array}\right)f(x)^k\Big(1-f(x)\Big)^{N_b-k}\right].$$

However, the relation in proposition Proposition 4.7 is too complex to be applied in practice and has to be replaced by a suitable approximation. First of all, we substite the expectations in (4.11) by the values of the corresponding probabilities in $\mathbb{E}[X]$ (see the discussion in Section C.1 [10]). Assuming that the algorithm converges to a stationary distribution such that the mean value of BLER is equal to the target p, we substitute p for $f(x)$ to obtain the following relation

$$\delta_{up}\sum_{k=N_{up}+1}^{N_b}\left(\begin{array}{c}N_b\\k\end{array}\right)p^k(1-p)^{N_b-k} = \delta_{down}\sum_{k=0}^{N_{down}-1}\left(\begin{array}{c}N_b\\k\end{array}\right)p^k(1-p)^{N_b-k}.$$

Algorithm 4.3 has been studied extensively in [16]. Several possible approximations for this latter relation have been exhibited, as well as a study of this algorithm's performance. Due to confidentiality restrictions, we can only present here the simplest case.

Example 4.8 ($N_{down}=1$, $N_{up}=0$)
This case represents an emulation of Sawtooth's behaviour. That is, target SIR is increased as soon as there is at least one error, and it is only decreased when no errors occur during the period of N_b blocks. This configuration is mostly useful for comparing the stability of this algorithm against that of Sawtooth.

On the other hand, determining the optimal relation between δ_{up}, δ_{down}, and target SIR in this case is relatively simple. Indeed, denoting the target BLER as above by p and assuming $p \ll 1$, we obtain

$$p(SIR_t) = P(N_{error}=0) = (1-p)^{N_b} = \Big((1-p)^{1/p}\Big)^{N_b\,p} \approx (1/e)^{N_b\,p},$$

and consequently

$$q(SIR_t) = P(N_{error}>0) = 1 - P(N_{error}=0) \approx 1-(1/e)^{N_b\,p}.$$

Substituting the two equations above into (C.4), we obtain the following relation

$$\delta_{down} = \delta_{up}\Big(e^{N_b\cdot BLER_{target}}-1\Big). \tag{4.12}$$

Restricting the target BLER even further by assuming $N_b \ll p^{-1}$, we can obtain an even simpler relation. Let us consider again the probability of increasing the target SIR.

$$q(SIR_t) = 1-(1-p)^{N_b} = 1-(1-N_b\,p+o(p^2)) = N_b\,p+o(p^2),$$

and thus, assuming that $e^{N_{tb}p}\approx 1$, we obtain the following relation

$$\delta_{down} \approx e^{N_{tb}p}\cdot\delta_{up}\cdot N_b\cdot BLER_{target} \approx \delta_{up}\cdot N_b\cdot BLER_{target}. \tag{4.13}$$

[10] It can be verified that both probabilities concerned satisfy necessary conditions.

4.6.3 Double loop algorithms

As it has been mentioned above, compared to Sawtooth, the algorithm described in the previous section mainly improves stability at low values of target BLER. Indeed, its main characteristic feature is the ability to delay target SIR update until a better estimate of current BLER is available to avoid unnecessarily changing target SIR when it is already close to the ideal value. On the other hand, as the target SIR is updated less frequently, the algorithm converges even slower than Sawtooth.

In order to improve OLPC algorithm's convergence, a different QoS metric has to be chosen instead of BLER, such that it can be well estimated in a shorter period of time. One of such metrics is, for example, the Bit Error Rate (BER). The obvious advantage of BER is that it takes the same time to transmit up to several thousands of bits (depending on the transport channel characteristics) as to transmit one block. The approximate number of bits or blocks necessary to estimate the corresponding error rate can be obtained with the De Moivre-Laplace theorem (see Section C.2), and is given by

$$n = \left\lceil \left(\frac{b}{\varepsilon}\right)^2 p(1-p) \right\rceil . \tag{4.14}$$

Example 4.9 (Voice communication)
Consider a voice communication. Typically, for this service, the target BLER is taken to be 0.01, and the corresponding BER is approximately 0.1288.

We start by computing the number of blocks necessary to estimate the BLER. Let us also require, for example, a precision of 0.01. Substituting these values into equation (4.14) we deduce that, taking the proportion of errors over a sequence of $n = \lceil 99\, b^2 \rceil$ blocks as an estimate of the BLER, the probability that the estimation error is less than 0.01 is approximately $G(b)$, where $G(b)$ is the probability of the interval $[-b, b]$ under standard normal distribution (see again Section C.2). [11] The values of $G(b)$ can be obtained from tables in numerous textbooks, and in particular it is known that $G(1.96) \approx 0.95$ and $G(2.6) \approx 0.99$. Therefore, taking $b = 2.6$, we can assume in the above context that with 99% probability the observed average BLER over 258 blocks is within 0.01 of the actual one.

Applying the same reasoning to the BER estimation, we conclude that, if the actual BER is close to the target of 0.1288, then to have, for example, a 99% probability of estimating the BER with precision 0.003 [12] we have to observe a sequence of approximately 84 283 bits.

In order to compare the duration of the two corresponding transmissions, we have to compute the number of bits that are sent per block. To do so, observe that typically the bit-rate for voice communication would be 12.2 kbps, and the channel would be coded with the convolutional coding of rate 1/3 with prior addition of 12 bits for the Circular Redundancy Check (CRC). Also, one block corresponds to two 10ms frames, and therefore, to achieve the 12.2 kbps bit-rate, we have to send $2 \times 12\,200/100 = 244$ information bits per block. Adding the CRC bits, and multiplying by 3, we see that 792 bits are actually transmitted per block in this context. We can now deduce that approximately 107 blocks should be sufficient to estimate the BER against 258 for the BLER.

Note 4.10 In the example above, the target BLER is 0.01. Several observations can be made regarding services with lower target BLER.

[11] This, of course, assuming that the actual BLER is close to the target 0.01.
[12] Simulations show that further increasing precision for BER estimation does not improve OLPC performance.

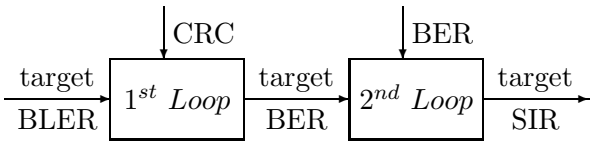


Figure 4.8: A systemic diagram of a double loop algorithm.

First of all, if we decrease the target BLER, the precision of BLER estimation has to increase. Typically, it should be at most the same as the target BLER. In this case, equation (4.14) can be rewritten as

$$n = \left\lceil \left(\frac{b}{p}\right)^2 p(1-p) \right\rceil = \left\lceil b^2 \, \frac{1-p}{p} \right\rceil .$$

Thus, the number of blocks required to obtain a good estimate of BLER increases when the latter goes to zero.

On the other hand, the necessary precision of BER estimation does not vary considerably, and the number of blocks necessary to estimate the actual BER is mostly determined by the number of bits per block. The latter, however, increases, as the services requiring low BLER would typically have a bit-rate of 64 kbps and higher.

Thus, for the services in question the time necessary to obtain an estimate of the BLER is higher than that of the example above, whereas the time necessary to estimate the BER is even less!

Note 4.11 De Moivre-Laplace theorem requires the trials to be independent. In real-life radio channels this can be assumed to be true for blocks, but not, a priori, for bits composing these blocks. Indeed, bit errors occur by bursts, that is the errors in the adjacent bits are not independent events. Nevertheless, we can apply this theorem thanks to the channel interleaving that is used to compensate for this burstyness. With channel interleaving, bits are rearranged before transmission in a pseudo-random manner, and arranged back in order on reception. As the transport channel BER is estimated after the deinterleaving, bit errors can also be considered to be independent.

The discussion above shows that BER can be estimated faster than BLER, and therefore it is more suitable when controlling a transmission with high QoS requirements. However, these requirements are still expressed in terms of target BLER, whereas corresponding target BER is often a priori unknown. Thus an OLPC algorithm that uses BER to determine target SIR, also has to determine the target BER corresponding to a given BLER.

This gives rise to the so-called *double loop* algorithms split — as the name suggests — in two loops: the first one determines the target BER observing the block errors, while the second one updates the target SIR depending on whether the actual BER is higher or lower than this target. Figure 4.8 illustrates this construction from a systemic point of view. One should notice, in particular, the similarity between the systemic decompositions of a double loop algorithm and the Outer Loop–Inner Loop pair (compare Figure 4.8 with Figure 4.5). Indeed, in both cases, the system is divided in two parts: one provides the output depending on a certain quality statistic, and the other one sets the target for the former in order to provide the required value of some higher level statistic. The fact that the nature of the two systems is similar, suggests also that their analyses should also be alike.

Double loop algorithms being more advanced than those described in the previous two sections, confidentiality restrictions apply once again, so we cannot give more detail of neither

the algorithms, nor the corresponding analysis. However, the study of one such algorithm can be found in [15, 16].

4.7 Discussion

In this chapter, we have presented the power control subsystem of UMTS, giving particular attention to the Outer Loop Power Control (OLPC) in uplink connection.

We have, in particular, presented two simpler algorithms for OLPC — Sawtooth and one of its extensions — and a common method, based on stochastic processes approach, for determining the necessary relation between the parameters of such an algorithm that would garantee optimal performance.

We have observed also that the common problem of both of these algorithms was intrinsic to the usage of CRC test as reference statistic, which happens to be insufficient, for services with low target error rate, as to the amount of information it conveys on the radio conditions. This observation leads to a conclusion that more sofisticated algorithms, using lower level statistics are required to obtain better performance in such cases.

One particular group of such more advanced algorithms consists of double-loop algorithms, which we did not present explicitly, in particular, due to confidentiality restrictions. We have presented, however, a systemic representation of such an algorithm, which happens to be structurally very similar to that of the Closed Loop Power Control (comprising both Outer Loop and Inner Loop), suggesting that the analysis of a double loop algorithm should eventually be performed in the same way as that of the Closed Loop Power Control.

Finally, the systemic representation, in Figure 4.7, of the uplink power control allowed us to underline once more the importance of a model for studying systems of heterogenous nature, exhibiting at the same time physical, logical, and hybrid components, with both continuous and discrete time evolution.

Chapter 5

Frame Level: Hybrid ARQ Control Schemes

We shall now descend even further in the hierarchical decomposition of the UMTS network by considering a particular subsystem of the Node B. Indeed, one of the latest features of the UMTS is the High Speed Downlink Packet Access (HSDPA), a service allowing packet access at a very high bit-rate compared to previous releases. It is based on a new channel called High Speed Downlink Shared Channel (HS-DSCH). Shared between several users, this channel is dedicated to downlink traffic and supports high data rates.

Although not the only modification required to introduce the HSDPA, the addition of HS-DSCH is probably the most important one. From the systemic point of view this corresponds to an addition of a new parallel coding chain to the coding component of Node B (cf. Section 3.3.1) and, of course, the corresponding decoding chain in the UE.

In this chapter, we concentrate on one particular technique introduced as part of HS-DSCH coding chain, which is the Hybrid ARQ (H-ARQ), and more precisely on the corresponding control schemes (see Section 5.1 for more details).

Contrary to the previous chapter, where we emphasised rather analytical methods for our study of the power control, in this chapter, we use the second dominant method of analysis of complex industrial systems, that is simulations. We proceeed as following. In Section 5.1, we give an overview of H-ARQ as well as the different techniques it consists of. In Section 5.2, we present the link level simulations that we have performed to compare several control schemes for H-ARQ. Finally, we conclude, in Section 5.3, by indentifying an optimal control scheme in terms of quality of service, as well as two suboptimal ones showing slightly worse performance but allowing to reduce the UE buffer requirements.

5.1 Overview of Hybrid ARQ

The HS-DSCH channel, which is the base of the new HSDPA service, is the main evolution of Release 5 (R5) of 3rd Generation Partnership Project (3GPP) specifications. Release 5 introduces for this channel some new advanced radio technologies both in the physical and in the Medium Access Level (MAC) layer. The main techniques are: a new modulation scheme, 16-state Quadrature Amplitude Modulation (16QAM); adaptive modulation and coding (AMC); and Hybrid Automatic Repeat Request (H-ARQ), an improved method of retransmission of false blocks. These new technologies allow to achieve data rates of up to 10.8 Mbps.

In downlink, H-ARQ allows a User Equipement (UE) to automatically request a retransmission of a block it didn't manage to decode correctly. In previous releases of UMTS (R99),

with H-ARQ type 1, on reception of a false block the UE discarded it and waited for a retransmission of the block from the Radio Network Controller (RNC) hoping to decode the new copy. In HSDPA fast H-ARQ is applied by retransmitting directly from Node B in the physical layer, thus enabling quicker retransmissions.

In Release 5, H-ARQ Type 2/3 is added, whose aim is to enable combining a retransmission with previous transmissions to increase the chances of correct decoding. The ensuing disadvantage of H-ARQ is that the UE needs to store the false blocks and add the new set of the soft decision bits to the previous ones it couldn't decode correctly, which requires additional memory and processing.

Although some research has been done to determine the optimal parameters for ARQ as such (see for example [71]), none is so far available in the context of H-ARQ for HSDPA as defined by 3GPP specifications.

H-ARQ is described in detail in [6] and consists of the following three techniques:

- *Chase combining (CC)* — if the received block doesn't have the correct Circular Redundancy Check (CRC) sequence, it is retransmitted and new values of soft decision bits are added to those of the first transmission.
- *Incremental redundancy (IR)* — incorrect block is retransmitted with different redundancy version parameters (different systematic over parity bits priority and/or rate matching parameters).
- *16QAM constellation rearrangement (CoRe)* — different mapping of blocks of bits to symbols.

5.1.1 Chase combining

Chase combining was originally proposed in [20]. It provides a considerable gain in transmission power (3 dB when two transmissions are used in Gaussian environment) at the cost of slightly increased processing complexity and a buffer in the UE that is required to store the received values.

To understand the idea behind this technique, one should first of all observe that all the way up to the decoding stage the receiver works with so-called *soft decision bits* rather than with logical ones, i.e. 0 or 1. Each soft decision bit represents the log-likelihood ratio of the corresponding bit, defined by

$$\Lambda = \log \frac{P(b = 0 \,|\, \hat{s})}{P(b = 1 \,|\, \hat{s})}$$

where b is the original transmitted bit, and $\hat{s}$ is the received signal. In other words, the log-likelihood ratio indicates if, given the received signal $\hat{s}$, the original transmitted bit is more likely to be 0 or 1. Figure 5.1 shows the log-likelihood ratio depending on the conditional probability $P(b = 0 \,|\, \hat{s})$. One can clearly see that the bigger the log-likelihood ratio, the higher is the probability that given the received signal $\hat{s}$ the original transmitted bit was equal to 0, and vice-versa.

Assuming that bits on the output of source encoder are identically distributed (that is $P(b = 0) = P(b = 1) = 1/2$) we can apply the Bayes' theorem to obtain

$$\Lambda = \log \frac{P(\hat{s} \,|\, b = 0)\frac{P(\hat{s})}{P(b=0)}}{P(\hat{s} \,|\, b = 1)\frac{P(\hat{s})}{P(b=1)}} = \log \frac{P(\hat{s} \,|\, b = 0)}{P(\hat{s} \,|\, b = 1)}\,.$$

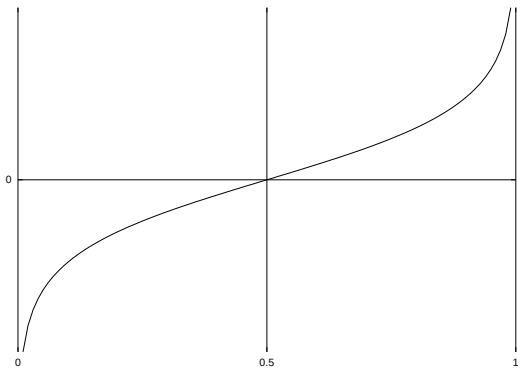


Figure 5.1: Log-likelihood ratio corresponding to a bit b as a function of $P(b = 0 \,|\, \hat{s})$.

The same ideas can be applied in case of multiple transmissions. Indeed, assume that a given block is not decoded correctly after the first transmission and is subsequently retransmitted. For each bit in this block, we obtain therefore two values of soft decision bits corresponding respectively to both of these transmissions. Consequently, the log-likelihood ratio can be defined, for any given bit, based on the two received signals $\hat{s}_1$ and $\hat{s}_2$. One can assume that the channel characteristics at the moments of these two transmissions are independent (for example, the first transmission might have occured during a fading dip, which would have been over before the second transmission was initiated). Applying Bayes' theorem as above we obtain the following relation.

$$\Lambda = \log \frac{P(\widehat{s_1}, \widehat{s_2}|b = 0)}{P(\widehat{s_1}, \widehat{s_2}|b = 1)} = \log \frac{P(\widehat{s_1}|b = 0)P(\widehat{s_2}|b = 0)}{P(\widehat{s_1}|b = 1)P(\widehat{s_2}|b = 1)} = \Lambda_1 + \Lambda_2 \,, \tag{5.1}$$

where Λ_1 and Λ_2 are the log-likelihood ratios of the bit in question corresponding to the first and the second transmission respectively.

However, in practice consequent transmissions are not entirely independent, and therefore the equality does not hold in (5.1). Nevertheless, the correlation is low, and putting $\Lambda \approx \Lambda_1 + \Lambda_2$ in the above notations happens to be a sufficiently good approximation of the log-likelihood ratio with respect to the two transmissions.

5.1.2 Incremental redundancy

Incremental redundancy provides yet another improvement by allowing to send additional information in case were retransmission is needed. The channel coding in HSDPA is based on the rate 1/3 Turbo encoding. This means that to every block of information bits, two blocks of parity bits of the same size are added at the encoding stage. Consequently, some bits have to be punctured before transmission to obtain a given bit rate. The incremental redundancy consists in puncturing different bits at consequent transmissions. In other words, bits which are punctured at the rate matching step of the first transmission can be sent at the second one (see Figure 5.2).

In HSDPA, one can prioritise sending systematic or parity bits, and at the same time vary the parameters of the rate matching algorithm, thus choosing not to puncture the same bits as at previous transmissions. Whatever the choice of priority (sending systematic or parity bits), there are two possible values of rate matching algorithm's parameters, which provides altogether four *redundancy versions*. [1].

[1] Observe that choosing the same redundancy version at each transmission, we obtain Chase combining. Thus

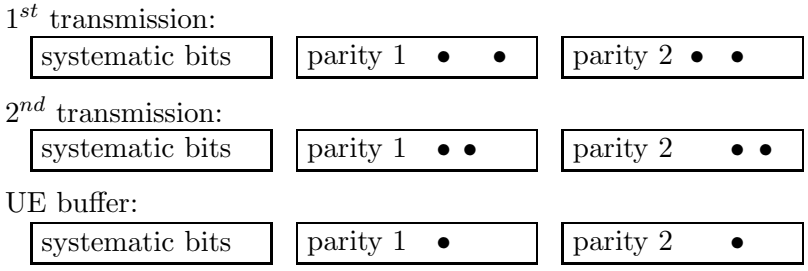


Figure 5.2: Illustration of the incremental redundancy principle (punctured bits are indicated by •; by combining the two transmissions, the UE has information about more bits than has been sent at any single transmission).

Incremental redundancy consists, therefore, in changing the redundancy version utilised at subsequent transmissions of the same block. This technique greatly improves Turbo decoder's performance (see [1, 2] for a comparison of IR vs CC). The disadvantage is that the buffer size in the UE has to increase considerably, as well as processing complexity.

5.1.3 16QAM constellation rearrangement

16QAM constellation rearrangement is a technique proposed in [3, 4] that allows to increase performance as compared to Chase combining while keeping processing complexity and buffer requirements comparatively low. Thus, constellation rearrangement can be viewed as a low complexity alternative to incremental redundancy. As implied by its name, this technique is only applicable when 16QAM modulation is used [2], and consists in changing the mapping of blocks of bits to complex symbols.

16QAM is a quadrature amplitude modulation based on a constellation of 16 symbols depicted in Figure 5.3. The bits to be transmitted are grouped in blocks of four. Each one of these blocks defines a constellation symbol that is then transmitted over a communication channel. More precisely, denoting the four bits by $i_1\, q_1\, i_2\, q_2$ correspondingly, the complex-valued symbol is obtained with the following formula:

$$s = \frac{\widetilde{i_1}(2 - \widetilde{i_2}) + j \cdot \widetilde{q_1}(2 - \widetilde{q_2})}{\sqrt{5}},$$

where $j = \sqrt{-1}$, and $\widetilde{b} = (-1)^b$ is the real-valued bit corresponding to the logical bit b. One can observe that first and third bits (i_1 and i_2) define the real part of the symbol, and second and fourth (q_1 and q_2) — the imaginary one, and therefore demodulation of the received signal $\widehat{s}$ would consist, firstly, in comparing its real and imaginary parts to zero in order to determine i_1 and q_1, and, secondly, in comparing absolute values of its real and imaginary parts to the threshold $2C/\sqrt{5}$ to determine i_2 and q_2, where C depends on the radio conditions and transmit power.

The advantage of 16QAM is that 4 bits are transmitted per single complex-valued symbol, as opposed to 2 in QPSK — the modulation used for all channels in UMTS transmitting user data, except for HS-DSCH —, thus doubling the possible bit-rate. On the other hand, its

the latter is a trivial case of Incremental redundancy.

[2] Here we place ourselves in the context of R5 UMTS where the only modulations in use are BPSK, QPSK, and 16QAM. In a different context constellation rearrangement could be used with any amplitude modulation (e.g. 64QAM, 128QAM, etc.)

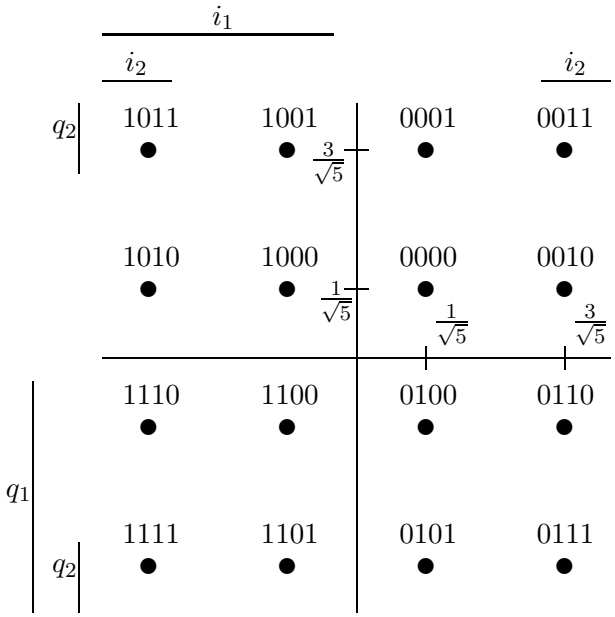


Figure 5.3: 16QAM symbol constellation.

Constellation version	Output bit sequence	Operation
0	$i_1\, q_1\, i_2\, q_2$	None (mapping as in Figure 5.3)
1	$i_2\, q_2\, i_1\, q_1$	Swapping MSBs with LSBs
2	$i_1\, q_1\, \overline{i_2\, q_2}$	Inversion of LSBs' logical values
3	$i_2\, q_2\, \overline{i_1\, q_1}$	Both swapping and inversion

Table 5.1: Constellation rearrangement for 16QAM (see [6], p. 65).

disadvantage is a more complex modulation and demodulation procedure, as well as increased sensitivity to radio conditions.

It is well known (see for example [75]) that among the four bits forming a symbol in 16QAM the probability of error can be considerably less for the most significant bits (MSBs) than for the less significant bits (LSBs). For example, if we consider symbol 2 (0010) of the constellation in Figure 5.3, for the first bit to be demodulated erroneously the perturbation of the real part of the transmited signal has to be three times that necessary to induce an error in the third bit.

In order to compensate for this effect, bits can be rearranged before retransmission in such a manner that some less protected bits become more protected. More precisely, denoting the four bits by $i_1\, q_1\, i_2\, q_2$, one of the four transformations in Table 5.1 is applied before they are mapped to a constellation symbol (see [6]).

Assuming that each symbol of the constellation is transmitted with equal probability, averaging of the probability of error over a long chain of bits is equivalent to averaging it over the symbol constellation. Thus, in order to better understand the principle of constellation rearrangement, we can consider a transmission where each constellation symbol is sent exactly once over an ideal channel (no fading, no noise).

Suppose we use standard bits-to-symbols mapping, i.e. constellation version 0. We then have a chain of 64 bits, out of which 16 are better protected than the other 48. These 16 bits are the

X_{rv}	0	1	2	3	4	5	6	7
s	1	0	1	0	1	1	1	1
r	0	0	1	1	0	0	0	1
b	0	0	1	1	1	2	3	0

Table 5.2: Encoding of redundancy version parameters for 16QAM.

two MSBs of each of four symbols in the corners of the constellation, and one of the MSBs for the eight other symbols on the constellation's exterior. Each of the four transformations in Table 5.1 provides better protection for a different set of 16 bits. Thus consequent retransmissions with different constellation arrangements would considerably improve Turbo decoder's performance.

One can make the following observations regarding constellation rearrangement.

- Constellation rearrangement does not require additional buffer in the UE. The only space required is that, necessary to store three additional tables for bits-to-symbols mapping, and it is negligible compared to the size of buffer used to store transmitted bits. There is no additional processing to be done.
- When only one transmission is performed, all four constellation rearrangement techniques are equivalent. Similarly, if several retransmissions are needed, whatever is the rearrangement sequence, there is always an equivalent one with first transmission using standard mapping. Maximum benefit from constellation rearrangement can be obtained with four retransmissions using different rearrangement techniques.

5.1.4 Control schemes

Both incremental redundancy and 16QAM constellation rearrangement are controlled by a set of so-called redundancy version (RV) parameters: r, s, and b that are in their turn encoded by a single parameter X_{rv}. The parameters r and s control the rate matching step, which is the base of incremental redundancy technique, while b controls the way 16QAM constellation is rearranged.

The value of s can be either 0 or 1 and indicates if, at rate matching step, the systematic bits are prioritised ($s = 1$) or not ($s = 0$). Once we know what flows are to be punctured in priority — systematic or parity bits — the value of r determines the exact puncturing pattern within these flows. The range of r is 0 to 3 for the QPSK [3] modulation or 0 to 1 for 16QAM. For 16QAM these parameters are encoded according to Table 5.2 (see also [6], p. 67).

At each transmission, the value of the X_{rv} parameter is chosen in the following manner. We fix a list $X = \{x_0, x_1, \ldots, x_{l-1}\}$ of values between 0 and 7, where l is arbitrary. We shall now set $X_{rv} = x_{n-1 \bmod l}$ at n-th transmission. In other words, for each given block the value of X_{rv} cycles through the list X that we shall call the *H-ARQ control scheme*.

For example, a list consisting of a single value $\{0\}$ defines the scheme that only uses Chase combining (the same redundancy version is sent at each retransmission). A list with two elements $\{0, 1\}$ implies that we send alternatively systematic and parity bits. If, in the latter case, a third or fourth retransmission is required the value of X_{rv} shall again be 0 and 1 correspondingly.

The question arises naturally: what H-ARQ control scheme is optimal in terms of the quality of service (QoS) (lowest I_{or}/I_{oc} [4] for a given BLER) and complexity (UE buffer and processing)?

[3] Quadrature Phase Shift Keying

[4] The ratio of the total user power to noise (dB)

Name	Scheme	Buffer	Description
CC	{0}	1	Chase combining
CoRe	{0,4,5,6}	1	Constellation rearrangement
IR	{0,1,7,3}	4	Incremental redundancy
IR+	{0,1,2,3}	4	Incremental redundancy with constellation rearrangement
IR/2–a	{0,7}	2	Systematic bits only with two rate matching patterns
IR/2–b	{0,1}	2	Systematic then parity bits; same rate matching pattern
CoReIR–	{0,1,4,1}	2	Hybrid sub-optimal
CoReIR	{0,1,4,**8**}	2	Hybrid (CoRe + IR)

Table 5.3: Compared H-ARQ control schemes. A scheme here is a list of values for the X_{rv} parameter to be used at consequent retransmissions (cf. Table 5.2). The third column shows the UE buffer space complexity index, i.e. the number of different redundancy version sent with a given control scheme.

$-\ o\ -$

As it has been mentioned above, the maximum benefit from constellation rearrangement can be obtained with four retransmissions. The same can be said about incremental redundancy in 16QAM as there are four different redundancy versions: systematic or parity bits, and two rate matching patterns for each of them. We shall therefore compare different H-ARQ control schemes of length four. The schemes we compare are shown in Table 5.3.

Along with the list defining a scheme and its short description we give in this table the scheme's space complexity index. This index represents the number of redundancy versions (systematic/priority bits, rate matching pattern) used in the scheme. Each additional redundancy version increments the number of bits that have to be stored in the UE buffer. Therefore the higher is the index in question, the more expensive the scheme is in terms of buffer requirements. [5] Let us briefly discuss the proposed schemes.

CC as indicated in Table 5.3 this scheme represents Chase combining. Indeed, the same bits are sent at all retransmissions.

CoRe this scheme makes full use of constellation rearrangement, but none of incremental redundancy: we send the same bits at each transmission using all possible bits-to-symbols mappings.

IR in this scheme we send alternatively systematic and parity bits. Two last retransmissions use a different rate matching pattern compared to the first two, thus making full use of incremental redundancy.

IR+ incremental redundancy is enhanced here by using constellation rearrangement on the last two retransmissions. This increases protection of bits that are not punctured in both rate matching patterns.

IR/2–a in this scheme only systematic bits are sent. Thus, comparing its performance with that of other schemes we can see if alternating systematic bits with parity ones is preferable to alternating rate matching patterns.

[5] One should keep in mind, however, that a scheme with space complexity index 4, for example, does not use four times more space than that with this index equal to 1.

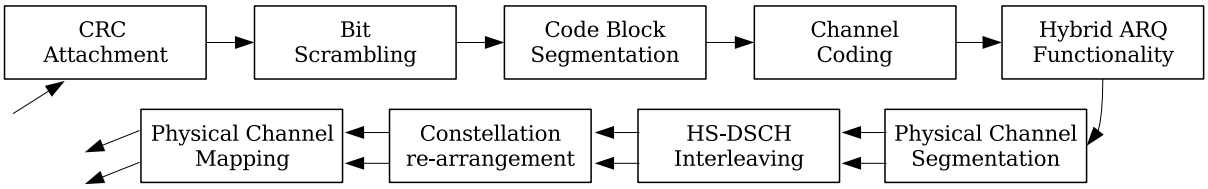


Figure 5.4: HSDPA coding chain.

IR/2–b this scheme is complementary to IR/2–a: we send systematic and parity bits alternatively maintaining the same rate matching pattern.

CoReIR– this scheme is an attempt at a compromise between the high performance and high complexity of IR on one hand, and lower performance and low complexity of CoRe on the other: we alternate systematic and parity bits, but instead of using different rate matching patterns, we use different bits-to-symbols mapping. As there is no available value of X_{rv} that would have $s = 0$, $r = 0$, and $b \neq 0$ (as $X_{rv} = 1$, but with a different bits-to-symbols mapping), we use the same value of X_{rv} for second and fourth transmissions.

CoReIR same as CoReIR–, but we introduce for testing purposes $X_{rv} = 8$ that is not in the 3GPP specifications (Table 5.2; also [6]) and encodes the following combination of RV parameters: $s = 0$, $r = 0$, $b = 1$, i.e. prioritising parity bits with the same rate matching pattern as for $X_{rv} = 1$ but with a different bits-to-symbols mapping.

The roles of these schemes are as follows. CC provides us with reference performances. IR+ being the scheme that ensures most diversity (different redundancy versions plus some use of constellation rearrangement), is the candidate for best performance, however it also requires the largest UE buffer. The goal of CoRe and IR is to enable a comparison between the two techniques, as well as to verify if sufficiently good results can be obtained using only one of them (initially constellation rearrangement was proposed in [3] to completely replace incremental redundancy). Furthermore, IR/2–a and IR/2–b allow us to single out the aspect of incremental redundancy that ensures the most gain in performance: IR/2–a does not send parity bits only varying the rate matching pattern, whereas IR/2–b does the inverse by alternating systematic and parity bits. Finally, as it has been mentioned above, CoReIR and CoReIR– are constructed by merging together incremental redundancy and constellation rearrangement in order to obtain good performances while keeping low UE buffer requirements.

5.2 Simulations

5.2.1 Simulation conditions

For all of the above H-ARQ schemes we perform link level simulations in Gaussian environment. We consider an HSDPA connection using 16QAM with one channelisation code at coding rate 3/4, and we consider BLER to I_{or}/I_{oc} ratio at different retransmissions. We implement the full HSDPA processing chain, of which the coding part is shown in Figure 5.4. The list of simulation parameters is given in Table 5.4.

Table 5.4: List of simulation parameters.

Parameter	Value
Channel model	Additive White Gaussian Noise (AWGN)
Chip-rate	3.84 Mcps
Power control	Off
Channel estimation	Ideal
Allocated power for HS-DSCH	80% (-1 dB)
Spreading factor	16
Number of codes for HS-DSCH	1
Number of slots per TTI	3
Frame length	2 ms
Number of transport blocks per TTI	1
Channel coding	Turbo code (rate 3/4)
CRC	24 bits
Tail bits	12
Turbo decoder	Log-MAP
Number of decoder iterations	8
Max number of retransmissions	4
Modulation	16QAM
Accuracy	50000 — 250000 slots per I_{or}/I_{oc} value; At least 100 block errors

5.2.2 Results

Figure 5.5 shows BLER performance of all considered H-ARQ control schemes after second transmission. In solid lines are plotted the curves for schemes that do not send parity bits on the second transmission, while those that do are plotted in dashed lines [6].

One can observe in the first place that sending parity bits provides a gain of approximately 1.8 dB at 10% BLER over resending systematic ones. Another observation to be made is that sending systematic bits with the same rate matching pattern but a different bits-to-symbols mapping (CoRe) provides a gain of approximately 0.4 dB over using a different rate matching pattern with the same mapping (IR/2–a).

Comparing performances after the third transmission (see Figure 5.6 [6]) allows us to differentiate between the schemes that produced similar results after second transmission. As expected, IR+ provides best performance vis-à-vis the BLER to I_{or}/I_{oc} ratio.

Both CoReIR and CoReIR– perform approximately 0.4 dB better than IR (cf. the second observation on the performances after two transmissions). The difference between IR+ and CoReIR (CoReIR–) is very slight (less than 0.1 dB). Thus, up to this point both CoReIR and CoReIR– provide performances close to optimal, while maintaning lower UE buffer requirements (see again Table 5.3).

Let us finally compare the performances after the fourth transmission (Figure 5.7). We can observe that due to retransmitting the same redundancy version as at the second transmission CoReIR– performs here worse than IR. At the same time CoReIR, which has a much lower requirements for UE buffer, provides a considerably better performance. Indeed, its performance is approximately 0.2 dB better than that of IR, and very close to IR+.

[6] Note that after second and third retransmissions performances are the same for some schemes, therefore all dashed curves in Figure 5.5 and Figure 5.6 coincide.

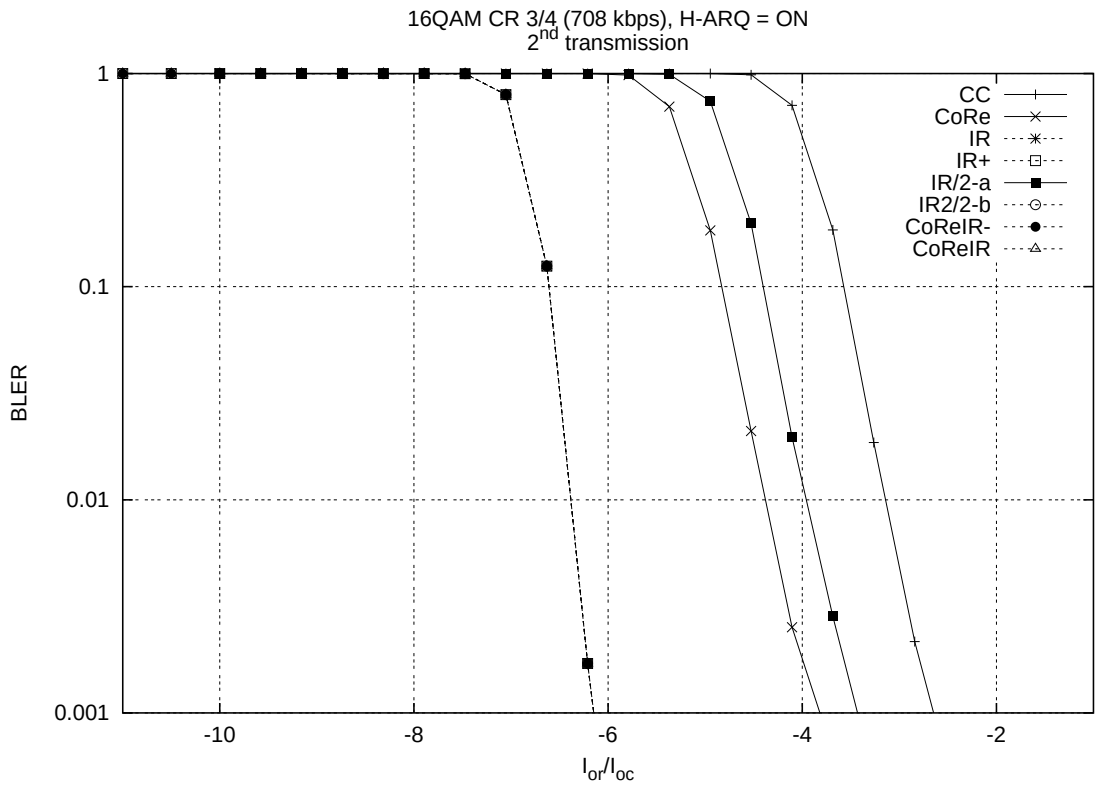


Figure 5.5: Performance of different H-ARQ control schemes after 2^{nd} transmission.

To finalise the presentation of simulation results we give in Table 5.5 the ratings of all schemes considered, and in Figure 5.8 the bit-rate achieved by the best candidates compared to that of Chase combining.

5.3 Discussion

The simulations presented above allow us to conclude that the best performance in terms of BLER to I_{or}/I_{oc} ratio is provided by the scheme we have denoted IR+ that makes full use of both incremental redundancy and constellation rearrangement. However — and for that reason —, this scheme requires the biggest UE buffer among all possible schemes.

Table 5.5: BLER to I_{or}/I_{oc} ranking of all schemes after each retransmission.

Name	UE buffer	2^{nd}	3^{rd}	4^{th}
CC	1	8	8	8
CoRe	1	6	6	5
IR	4	1–5	4	3
IR+	4	1–5	1	1
IR/2–a	2	7	7	7
IR/2–b	2	1–5	5	6
CoReIR–	2	1–5	2–3	4
CoReIR	2	1–5	2–3	2

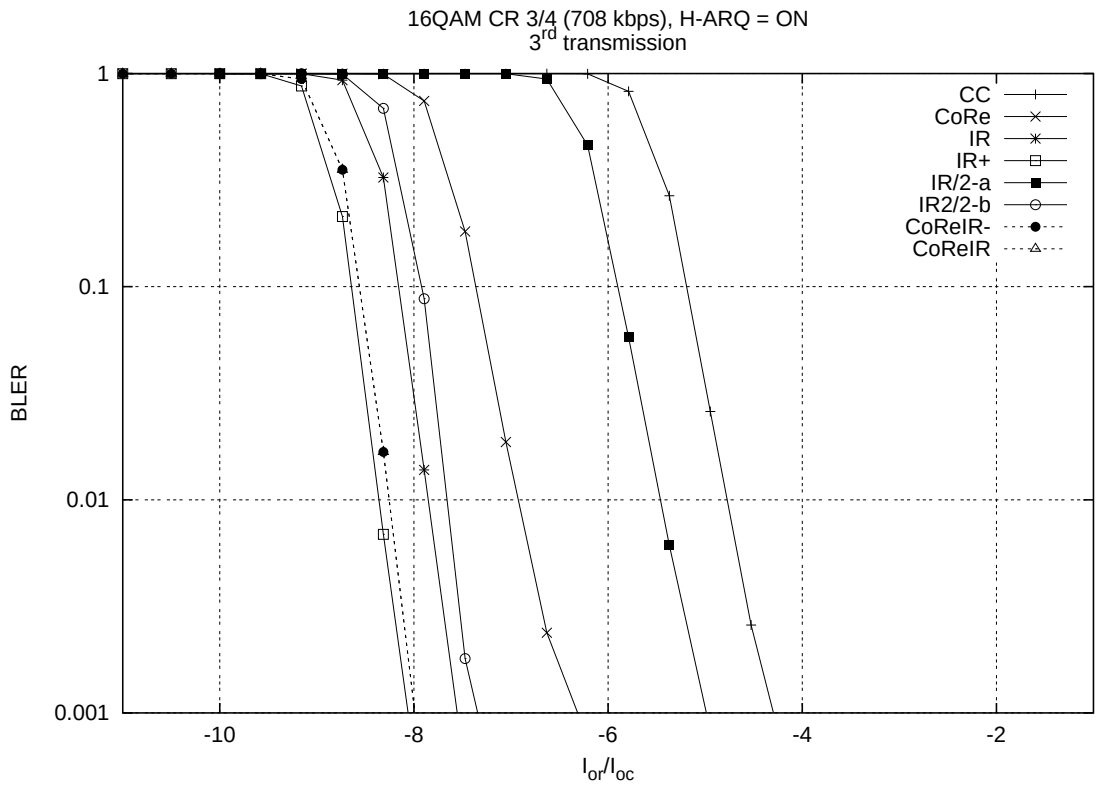


Figure 5.6: Performance of different H-ARQ control schemes after 3^{rd} transmission.

At the same time another scheme denoted CoReIR provides performance that is only about 0.1 dB worse than that of IR+ while considerably reducing UE buffer requirements[7].

The disadvantage of this scheme is that at the fourth transmission it uses a combination of redundancy version parameters that is not found in the 3GPP specifications, i.e. $s = 0$, $r = 0$, and $b = 1$ (transmitting parity bits with the same rate matching pattern as for $X_{rv} = 1$, but with a different bits-to-symbols mapping).

Therefore there are two possible lines of action:

1. Without changing the specifications (see Table 5.2), one should decide between IR+ and CoReIR– according to what is being prioritised: performance or UE buffer size.

2. One of the entries of the Table 5.2 ($X_{rv} = 5$, 6, or 7) should be modified to match the parameters specified above, and CoReIR should be selected as optimal H-ARQ control scheme.

[7] UE supporting all coding rates from 1/3 to 1 would require approximately 50% more buffer with IR+ than with CoReIR.

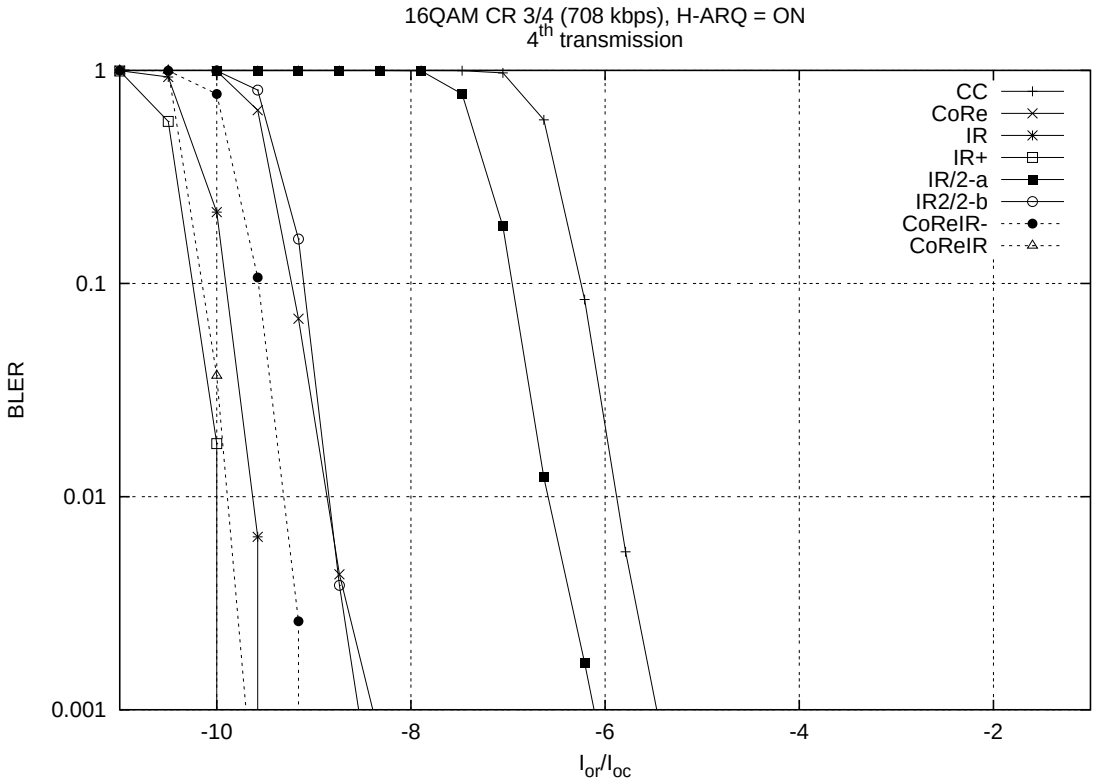


Figure 5.7: Performance of different H-ARQ control schemes after 4^{th} transmission.

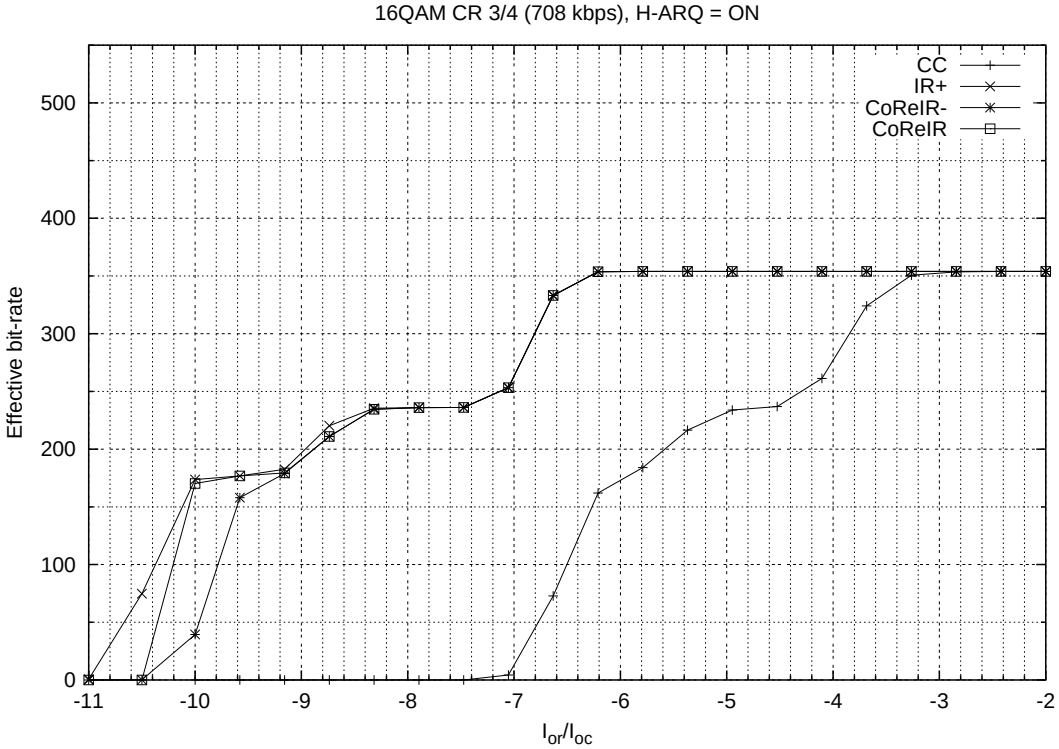


Figure 5.8: Effective bit-rate achieved with best H-ARQ control schemes.

Chapter 6

Bit Level: Analysis of BPSK Modulation with Spatial Diversity

In this chapter, we conclude our descent through the levels of the system modelling the UMTS network, which we have started in Chapter 3, by considering a transmission of a single bit over a radio channel with spatial diversity, which implies a presence of multiple paths (or trajectories) between the transmitting and receiving antennae. In this context, we study the probability that on reception this bit is not correctly demodulated, i.e. the Bit Error Rate (BER), which, as we have seen in Chapter 4, is a very important statistic in a telecommunications network.

An expression for this error rate has been studied in [30], [31], and [54], where, on one hand, a stable algorithm was developed to compute it, and, on the other hand, a combinatorial interpretation of the underlying expression was obtained by means of Young tabloids of $N \times N$ square shape.

We devote this chapter to generalising the results of these studies to the conditional probability that the log-likelihood is below a certain threshold ε when the transmitted bit is 0. [1]

In Section 6.1, we briefly present the model describing the Binary Phase Shift Keying (BPSK) modulation that we use for our studies. In Section 6.2, we consider the probability that a bit's log-likelihood is less than a given threshold ε and deduce two expressions in terms of symmetric functions for first coefficients of its Taylor expansion. One of these expressions leads to a stable and efficient algorithm computing these coefficients, whereas the second one allows an interesting combinatorial interpretation that we develop in Section 6.3.

This combinatorial interpretation involves a class of objects that we call square tabloids with ribbons. We show that a Robinson-Schensted-Knuth correspondence can be naturally extended to associate a {0,1}-matrix to each square tabloid with ribbon, and we conclude by providing a complete and independent characterisation of the class of {0,1}-matrices that arise in this context.

6.1 Signal processing background

Modulating numerical signals means transforming them into wave forms. Due to their importance in practice, modulation methods were widely studied in signal processing (see, for instance,

[1] We have already mentioned log-likelihood in Chapter 5. However, for consistency of the chapter, we repeat this definition in the next section. Here, it is sufficient to notice that the probability of a bit error is equal to the conditional probability that the the log-likelihood is negative given that the transmitted bit is 0.

Chapter 5 of [75]). Among the different modulation protocols used in practical contexts, an important class consists in methods where the modulation reference (i.e. a fixed numerical sequence) is transmitted over the same channel as the usual signal. The demodulation decision is based then on at least two noisy informations, i.e. the transmitted signal and the transmitted reference. It happens, however, that one can also take in account in the demodulating process several noisy copies of these last two signals: one speaks then of demodulation with diversity. It appears that the probability of errors in such context is of the following form:

$$P(U < V) = P\left(U = \sum_{i=1}^{N} |u_i|^2 \ < \ V = \sum_{i=1}^{N} |v_i|^2 \right), \tag{6.1}$$

where N is the number of paths (and consequently the number of copies of the information-reference pair), and the u_i's and v_i's denote independent centered complex Gaussian random variables with variances equal to $E[\,|u_i|^2\,] = \chi_i$ and $E[\,|v_i|^2\,] = \delta_i$ for every $i \in [1, N]$ (see also Section 6.1.2).

The problem of computing explicitly probabilities of this last type was studied in signal processing by several researchers (see [12, 46, 75, 90]). The most interesting result in this direction is due to Barett ([12]) who obtained the following expression for the probability given by formula (6.1).

$$P(U < V) = \sum_{k=1}^{N} \left(\prod_{i \neq k} \frac{1}{1 - \delta_k^{-1}\delta_i} \prod_{i=1}^{N} \frac{1}{1 + \delta_k^{-1}\chi_i} \right) \tag{6.2}$$

Observe, however, that, when for some $i \neq k$ the values of δ_k and δ_i are close, the denominator $1 - \delta_k^{-1}\delta_i$ in the right-hand part of this expression is close to zero, and thus the equation is computationally unstable.

Equation (6.2) provides us the conditional probability that the *log-likelihood* of a bit is less than zero under the condition that the transmitted bit was $+1$. [2] We remind the reader that log-likelihood is defined by setting

$$\Lambda = \log \frac{P(b = +1|X)}{P(b = -1|X)},$$

and allows to decide what was the value of the transmitted logical bit. It is also essential for various decoding algorithms such as MAP and its variants, and Soft Output Viterbi Algorithm (SOVA) (see for example Chapter 4 of [42]).

6.1.1 Multipath channel model

We consider a model where one transmits an information $b \in \{-1, +1\}$ on a noisy channel [3]. A reference $r = 1$ is also sent on the noisy channel at the same time as b. We assume that we receive N pairs $(x_i(b), r_i)_{1 \leq i \leq N} \in (\mathbf{C} \times \mathbf{C})^N$ of data (the $x_i(b)$'s) and references (the r_i's) [4] that have the following form

$$\begin{cases} x_i(b) &= a_i\, b + \nu_i & \text{for every } 1 \leq i \leq N, \\ r_i &= a_i \sqrt{\beta_i} + \nu_i' & \text{for every } 1 \leq i \leq N, \end{cases}$$

[2] Real-valued bits are often considered in signal processing theory, which are defined as $(-1)^b$ with b being a logical 0–1 bit.

[3] This is the case for example when BPSK modulation is used. For a large number of other modulation methods the information transmitted is more complex, and contains more than one bit. However, performance analysis for these modulations can be reduced to that of BPSK (see [75]).

[4] One speaks in this case of spatial diversity, i.e. when more than one antenna is available, but also of multipath reflexion contexts. Both of these situations are very common in mobile communications.

where $a_i \in \mathbf{C}$ is a complex number that models the channel fading associated with $x_i(b)$ [5], where $\beta_i \in \mathbf{R}^+$ is a positive real number that represents the signal to noise ratio (SNR) which is available for the reference r_i and where $\nu_i \in \mathbf{C}$ and $\nu_i' \in \mathbf{C}$ denote finally two independent complex white Gaussian noises. We also assume that every a_i is a complex random variable distributed according to a centered Gaussian density of variance α_i for every $i \in [1, N]$.

According to these assumptions, all observables of our model, i.e. the pairs $(x_i(b), r_i)$ for all $1 \leq i \leq N$, are complex Gaussian random variables. We finally also assume that these N observables are mutually independent random variables in $\mathbb{C}^2$. Under these hypotheses we have the following expression for the log-likelihood.

$$\Lambda = \log\left(\frac{P(b=+1|X)}{P(b=-1|X)}\right) = \sum_{i=1}^{N} \frac{4\,\alpha_i\,\sqrt{\beta_i}}{1+\alpha_i\,(\beta_i+1)}\,(x_i(b)|r_i) \tag{6.3}$$

with $X = (x_i(b), r_i)_{1\leq i\leq N}$ and where $(\star|\star)$ denotes the Hermitian scalar product. One indeed decides that b was equal to 1 (resp. to -1) when the right hand side of (6.3) is positive (resp. negative) [6]. One obtains (6.1) now applying the parallelogram identity to (6.3).

The situation undesirable for both demodulation (increased chances of taking incorrect decision) and soft decoding algorithms (unreliable input) is when the log-likelihood is close to zero, i.e. $|\Lambda| < \varepsilon$. We shall therefore study the probability $P(U - V < \varepsilon)$ [7] generalising (6.1) where $P(U - V < 0)$ is considered instead.

6.1.2 The analogue of Barret's formula

Let us consider two real random variables U and V defined, as in [31] by setting

$$U = \sum_{i=1}^{N} |u_i|^2 \quad \text{and} \quad V = \sum_{i=1}^{N} |v_i|^2$$

where u_i's and v_i's are independent centered complex Gaussian random variables with variances $\mathbf{E}[|u_i|^2] = \chi_i$ and $\mathbf{E}[|v_i|^2] = \delta_i$ for every $i \in [1, N]$. It is then easy to prove by induction on N that the probability distribution functions of U and V are equal to

$$d_U(x) = \sum_{j=1}^{N} \frac{\chi_j^{N-2}}{\prod\limits_{1\leq i\neq j\leq N}(\chi_j - \chi_i)}\, e^{-\frac{x}{\chi_j}} \quad \text{and} \quad d_V(x) = \sum_{k=1}^{N} \frac{\delta_k^{N-2}}{\prod\limits_{1\leq i\neq k\leq N}(\delta_k - \delta_i)}\, e^{-\frac{x}{\delta_k}} \tag{6.4}$$

when all variances χ_i and δ_i are distinct. One can then easily obtain

$$P(V > x) = \int_x^{+\infty} d_V(t)\,dt = \sum_{k=1}^{N} \frac{\delta_k^{N-1}}{\prod\limits_{1\leq i\neq k\leq N}(\delta_k - \delta_i)}\, e^{-\frac{x}{\delta_k}}. \tag{6.5}$$

We then have the following expression for $P(U - V < \varepsilon)$

$$P(U - V < \varepsilon) = \int_0^{+\infty} d_U(x) P(V > x - \varepsilon)\,dx.$$

[5] Fading is typically the result of the absorption of the signal by buildings. Its complex nature comes from the fact that it models both an attenuation (its modulus) and a dephasing (its argument).

[6] In the case when Turbo codes are used for channel coding the actual value of log-likelihood represents the reliability of the input.

[7] Probability $P(U - V < \varepsilon)$ can be studied independently as the distribution function of the random variable $U - V$ (cf. [39]).

Substituting relations (6.4) and (6.5), this last identity leads to the expression

$$P(U-V<\varepsilon)=\int_0^{+\infty}\sum_{j,k=1}^{N}\frac{\chi_j^{N-2}\,\delta_k^{N-1}}{\prod\limits_{1\leq i\neq j\leq N}(\chi_j-\chi_i)\prod\limits_{1\leq i\neq k\leq N}(\delta_k-\delta_i)}\,e^{-\frac{x}{\chi_j}}\,e^{-\frac{x}{\delta_k}}\,e^{\frac{\varepsilon}{\delta_k}}\,dx,$$

from which we immediately obtain the relation

$$P(U-V<\varepsilon)=\sum_{j,k=1}^{N}\frac{\chi_j^{N-1}\,\delta_k^{N}}{(\delta_k+\chi_j)\prod\limits_{1\leq i\neq j\leq N}(\chi_j-\chi_i)\prod\limits_{1\leq i\neq k\leq N}(\delta_k-\delta_i)}\,e^{\frac{\varepsilon}{\delta_k}}.$$

This last formula can now be rewritten as follows

$$P(U-V<\varepsilon)=\sum_{k=1}^{N}\frac{\delta_k^N e^{\frac{\varepsilon}{\delta_k}}}{\prod\limits_{1\leq i\leq N}(\delta_k+\chi_i)\prod\limits_{1\leq i\neq k\leq N}(\delta_k-\delta_i)}\left(\sum_{j=1}^{N}\frac{\prod\limits_{1\leq i\neq j\leq N}(\delta_k+\chi_i)}{\prod\limits_{1\leq i\neq j\leq N}(\chi_j-\chi_i)}\,\chi_j^{N-1}\right),\quad(6.6)$$

Finally we can deduce the analogue of Barret's formula (cf. [12, 31]):

$$P(U-V<\varepsilon)=\sum_{k=1}^{N}\frac{\delta_k^{2N-1}e^{\frac{\varepsilon}{\delta_k}}}{\prod\limits_{1\leq i\leq N}(\delta_k+\chi_i)\prod\limits_{1\leq i\neq k\leq N}(\delta_k-\delta_i)}\quad(6.7)$$

due to the fact that the internal sum in relation (6.6) is just the Lagrange interpolation expression taken at the points $(-\chi_j)_{1\leq j\leq N}$ for the polynomial δ_k^{N-1} (considered here as a polynomial of $\mathbf{C}[\chi_1,\ldots,\chi_N][\delta_k]$).

6.2 Symmetric functions expression

In this section, we use a number of facts and notations related to symmetric functions in general, and Schur functions in particular. These facts and notations can be found in Section D.2 of Appendix D.

We shall try to represent the probability $P(U-V<\varepsilon)$ in terms of Schur functions. In order to do so we have to get rid of the exponential in the numerator of the right hand side of (6.7). Replacing it by its Taylor decomposition, we obtain

$$P(U-V<\varepsilon)=\sum_{m=0}^{+\infty}\sum_{k=1}^{N}\frac{\delta_k^{2N-m-1}}{\prod\limits_{1\leq i\leq N}(\delta_k+\chi_i)\prod\limits_{1\leq i\neq k\leq N}(\delta_k-\delta_i)}\times\frac{\varepsilon^m}{m!}.$$

We will now concentrate our efforts on the m-th coefficient of this exponential series, i.e.

$$P_m^{(N)}=P_m^{(N)}(\Delta,X)=\sum_{k=1}^{N}\frac{\delta_k^{2N-m-1}}{\prod\limits_{1\leq i\leq N}(\delta_k+\chi_i)\prod\limits_{1\leq i\neq k\leq N}(\delta_k-\delta_i)}.\quad(6.8)$$

This formula can be expressed using the Lagrange operator L. Let us indeed set $\delta_k=x_k$ and $\chi_k=-y_k$ for every $k\in[1,N]$. Then one can rewrite (6.8) as

$$P_m^{(N)}=\sum_{k=1}^{N}\frac{x_k^{2N-m-1}}{R(x_k,Y)R(x_k,X\backslash x_k)},$$

where we denoted $X = \{ x_1, \ldots, x_N \}$ and $Y = \{ y_1, \ldots, y_N \}$ and where

$$R(A,B) = \prod_{a \in A, b \in B} (a-b)$$

is the resultant of two polynomials having A and B as sets of roots. Hence, we have

$$P_m^{(N)} = \sum_{k=1}^{N} \frac{g(x_k, X\backslash x_k)}{R(x_k, X\backslash x_k)} = L(g) \tag{6.9}$$

where g stands for the element of $Sym(x_1) \otimes Sym(X\backslash x_1)$ defined by setting

$$g(x_1, X\backslash x_1) = g(x_1) = \frac{x_1^{2N-m-1}}{R(x_1, Y)}\,.$$

Observe now that one has

$$g(x_1, X\backslash x_1) = \frac{1}{R(X,Y)}\, x_1^{2N-m-1} f(x_1, X\backslash x_1)$$

where f stands for the element of $Sym(x_1) \otimes Sym(X\backslash x_1)$ defined by setting

$$f(x_1, X\backslash x_1) = R(X\backslash x_1, Y) = s_{(N^{N-1})}((X\backslash x_1) - Y) \tag{6.10}$$

(the last above equality comes from the expression of the resultant in terms of Schur functions). Note now that the resultant $R(X,Y)$, being symmetric in the alphabet X, is a scalar for the operator L. It follows therefore from relation (6.9) that one has

$$P_m^{(N)} = \frac{L(x_1^{2N-m-1} f(x_1, X\backslash x_1))}{R(X,Y)}. \tag{6.11}$$

Let us now study the numerator of the right-hand side of relation (6.11) in order to give another expression for $P_m^{(N)}$. Note first that Cauchy formula leads to the development

$$s_{(N^{N-1})}((X\backslash x_1) - Y) = \sum_{\lambda \subset (N^{N-1})} s_\lambda(X\backslash x_1) s_{(N^{N-1})/\lambda}(-Y)\,. \tag{6.12}$$

According to the identities (6.10) and (6.12), we now obtain for $0 \leq m < 2N$ the relations

$$\begin{aligned} L(x_1^{2N-m-1} f(x_1, X\backslash x_1)) &= \sum_{\lambda \subset (N^{N-1})} L(x_1^{2N-m-1} s_\lambda(X\backslash x_1)) s_{(N^{N-1})/\lambda}(-Y) \\ &= \sum_{\lambda \subset (N^{N-1})} s_{(\lambda, N-m)}(X) s_{(N^{N-1})/\lambda}(-Y), \end{aligned}$$

the last above equality being an immediate consequence of Theorem D.11. Using the equality

$$s_{\lambda/\mu}(-X) = s_{\tilde{\lambda}/\tilde{\mu}}(X),$$

where $\mu \subset \lambda$ (see [66]) and the definition of skew Schur functions, we can rewrite the last above expression as

$$L(x_1^{2N-m-1} f(x_1, X\backslash x_1)) = \sum_{\lambda \subset (N^{N-1})} (-1)^{|\lambda|} s_{(\lambda, N-m)}(X) s_{\overline{(\lambda, N)}}(Y),$$

where $0 \leq m < 2N$, and $\overline{(\lambda, N)}$ denotes the complementary partition of (λ, N) in the square N^N. Going back to the initial variables, the signs disappear in the previous formula by homogeneity of Schur functions. Reporting the identity obtained in such a way into relation (6.11), we finally obtain an expression for $P_m^{(N)}$ in terms of Schur functions, i.e.

$$P_m^{(N)} = \frac{\sum_{\lambda \subset (N^{N-1})} s_{(\lambda, N-m)}(\Delta) s_{\overline{(\lambda,N)}}(X)}{\prod_{1 \leq i,j \leq N} (\chi_i + \delta_j)} \tag{6.13}$$

where $X = \{\chi_1, \ldots, \chi_N\}$, $\Delta = \{\delta_1, \ldots, \delta_N\}$.

6.2.1 A determinantal approach

Let us now go back to the alphabets X and Y defined in Section 6.2. We saw there that

$$P_m^{(N)} = \frac{f_m^{(N)}(X, Y)}{R(X, Y)} \tag{6.14}$$

where $0 \leq m < 2N$, and $f_m^{(N)}(X, Y)$ is a symmetric function of $Sym(X) \otimes Sym(Y)$ given by

$$f_m^{(N)}(X, Y) = \sum_{\lambda \subset (N^{N-1})} s_{(\lambda, N-m)}(X) s_{(N^{N-1})/\lambda}(-Y).$$

Let's now compute the action of the vertex operator $\Gamma_z(X)$ (see Section D.2.2) on the rectangle Schur function $s_{(N^{N-1})}(X - Y)$. Recall first that Cauchy formula shows that one has

$$s_{(N^{N-1})}(X - Y) = \sum_{\lambda \subset (N^{N-1})} s_\lambda(X)\, s_{(N^{N-1})/\lambda}(-Y).$$

Applying the vertex operator $\Gamma_z(X)$ to this expansion, we now get

$$\begin{aligned} \Gamma_z(X)(s_{(N^{N-1})}(X - Y)) &= \sum_{\lambda \subset (N^{N-1})} \Gamma_z(X)(s_\lambda(X))\, s_{(N^{N-1})/\lambda}(-Y) \\ &= \sum_{k=-\infty}^{+\infty} \Big(s_{(\lambda,k)}(X)\, s_{(N^{N-1})/\lambda}(-Y) \Big) z^k. \end{aligned}$$

Hence, $f_m^{(N)}(X, Y)$ is equal to the coefficient of z^{N-m} in the image of $s_{(N^{N-1})}(X - Y)$ by $\Gamma_z(X)$. On the other hand, using Cauchy formula in connection with relation

$$\Gamma_z(X)(s_\lambda(X)) = \sigma_z(X)\, s_\lambda(X - 1/z).$$

given by Thibon in [89], one can also write

$$\begin{aligned} \Gamma_z(X)(s_{(N^{N-1})}(X - Y)) &= \sigma_z(X) s_{(N^{N-1})}(X - Y - 1/z) \\ &= \sigma_z(X) \Big(\sum_{j=0}^{N-1} s_{(N^{N-1})/(1^j)}(X - Y)\, s_{(1^j)}(-1/z) \Big) \\ &= \Big(\sum_{i=0}^{+\infty} s_i(X) z^i \Big) \Big(\sum_{j=0}^{N-1} s_{(N^{N-1})/(1^j)}(X - Y)(-1/z)^j \Big) \end{aligned}$$

due to the fact that the only non zero Schur functions of the alphabet $-1/z$ are indexed by column partitions of the form 1^k (and are equal to $(-1/z)^k$). The coefficient of z^{N-m} in the above product gives us then a new expression for $f_m^{(N)}(X,Y)$, i.e.

$$f_m^{(N)}(X,Y) = \sum_{k=0}^{N-1} (-1)^k s_{N-m+k}(X) s_{N^{N-1}/1^k}(X-Y). \tag{6.15}$$

But this last expression is just the development along the last colomn of the determinant

$$\begin{vmatrix} s_N(X-Y) & s_{N+1}(X-Y) & \dots & s_{2N-2}(X-Y) & s_{2N-m-1}(X) \\ s_{N-1}(X-Y) & s_N(X-Y) & \dots & s_{2N-3}(X-Y) & s_{2N-m-2}(X) \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ s_2(X-Y) & s_3(X-Y) & \dots & s_N(X-Y) & s_{N-m+1}(X) \\ s_1(X-Y) & s_2(X-Y) & \dots & s_{N-1}(X-Y) & s_{N-m}(X) \end{vmatrix}, \tag{6.16}$$

which is an expression of the multi-Schur fonction $s_{(N^{N-1},N-m)}(X-Y,\dots,X-Y,X)$. Hence, relation (6.16) gives us both a determinantal and a multi-Schur expression for the denominator of the right hand side of formula (6.14). Using the interpretation of the resultant $R(X,Y)$ as a multi-Schur function, we can conclude that for $0 \leq m < 2N$

$$P_m^{(N)} = \frac{s_{(N^{N-1},N-m)}(X-Y,\dots,X-Y,X)}{s_{(N^N)}(X-Y,\dots,X-Y)} \tag{6.17}$$

where the alphabet $X-Y$ appears $N-1$ times in the numerator and N times in the denominator of the right hand side of the above formula.

6.2.2 A Toeplitz system and its solution

Using the determinantal expression of the multi-Schur function $s_{(N^N)}(X-Y,\dots,X-Y)$, we can now observe that for all $0 \leq m < 2N$ relation (6.17) shows that $P_m^{(N)}$ is equal to the quotient of the determinant (6.16) by the determinant

$$\begin{vmatrix} s_N(X-Y) & s_{N+1}(X-Y) & \dots & s_{2N-1}(X-Y) \\ s_{N-1}(X-Y) & s_N(X-Y) & \dots & s_{2N-2}(X-Y) \\ \vdots & \vdots & \ddots & \vdots \\ s_1(X-Y) & s_2(X-Y) & \dots & s_N(X-Y) \end{vmatrix},$$

which is obtained by replacing the last column of the determinant (6.16) by the N-dimensional vector $(s_{2N-1}(X-Y), s_{2N-2}(X-Y), \dots, s_N(X-Y))$. Hence, the right hand side of relation (6.17) can be interpreted as the Cramer expression for the last component p_0 of the linear system

$$\begin{pmatrix} s_N(X-Y) & s_{N+1}(X-Y) & \dots & s_{2N-1}(X-Y) \\ s_{N-1}(X-Y) & s_N(X-Y) & \dots & s_{2N-2}(X-Y) \\ \vdots & \vdots & \ddots & \vdots \\ s_1(X-Y) & s_2(X-Y) & \dots & s_N(X-Y) \end{pmatrix} \begin{pmatrix} p_{N-1} \\ p_{N-2} \\ \vdots \\ p_0 \end{pmatrix} = \begin{pmatrix} s_{2N-m-1}(X) \\ s_{2N-m-2}(X) \\ \vdots \\ s_{N-m}(X) \end{pmatrix}.$$

Let us now set $\pi_m(t) = p_0 + p_1\, t + \dots + p_{N-1}\, t^{N-1}$. The above linear system implies that the coefficients of order N to $2N-1$ in the series $\pi_m(t)\, \sigma_t(X-Y)$ are equal to the coefficients

of the same order in $t^m\sigma_t(X)$. This means equivalently that there exists a polynomial $\mu_m(t)$ of degree less than or equal to $N-1$ such that one has

$$\pi_m(t)\,\sigma_t(X-Y) - t^m\sigma_t(X) + \mu_m(t) = O(t^{2N})\ .$$

Going back to the definition of $\sigma_t(X-Y)$, one can notice that this property can be rewritten as

$$(\pi_m(t)\lambda_{-t}(Y) - t^m)\sigma_t(X) + \mu_m(t) = O(t^{2N})$$

that is itself clearly equivalent to

$$\pi_m(t)\lambda_{-t}(Y) + \mu_m(t)\lambda_{-t}(X) = t^m + O(t^{2N}).$$

Since the left hand side of the above identity is a polynomial of degree at most $2N-1$, it follows that its right hand side must be equal to t^m, keeping in mind that we consider $0 \leq m < 2N$. Hence, we showed that one has

$$\pi_m(t)\lambda_{-t}(Y) + \mu_m(t)\lambda_{-t}(X) = t^m. \tag{6.18}$$

Thus, for $0 \leq m < 2N$, $P_m^{(N)}$ is the constant term $\pi_m(0)$ of the polynomial $\pi_m(t)$, where $\pi_m(t)$ and $\mu_m(t)$ are the polynomials of degree $\leq N-1$ defined by (6.18).

6.2.3 A Bezoutian algorithm

Algorithm 6.1 shown on page 91 computes π_m and μ_m iteratively, starting with $m = 0$, and then consequetively deriving π_m and μ_m from π_{m-1} and μ_{m-1} for $m = 1 \ldots 2N-1$.

Let us now prove the consistency of this algorithm.

Proposition 6.1 *The polynomials $\pi_m(t)$ and $\mu_m(t)$, produced by Algorithm 6.1, satisfy relation (6.18).*

Proof. We argue by induction on m. The case $m = 0$ being obvious, we can consider only $m \geq 1$.

Suppose that at `Step` $m-1$ we have found the two polynomials $\pi_{m-1}(t)$ and $\mu_{m-1}(t)$ of degrees $\leq N-1$ satisfying the relation (6.18) for $m-1$. First of all observe that it follows immediately from (6.18) and the fact that $m < 2N$ that, $d(\pi_{m-1}) = N-1$, then also $d(\mu_{m-1}) = N-1$ and vice versa.

If we have $d(\pi_{m-1}), d(\mu_{m-1}) < N-1$, then the two polynomials $\pi_m(t) = t\pi_{m-1}(t)$ and $\mu_m(t) = t\mu_{m-1}(t)$ satisfy (6.18) for m. Observe that in this case the coefficient c calculated in `Step` m.1 is zero, and hence relation (6.20) holds.

Suppose now that $d(\pi_{m-1}) = d(\mu_{m-1}) = N-1$. Then we can set

$$\begin{cases} \pi_{m-1}(t) = a\,t^{N-1} + \pi'(t) \\ \mu_{m-1}(t) = b\,t^{N-1} + \mu'(t) \end{cases} \quad \text{with } d(\pi'), d(\mu') < N-1\,. \tag{6.21}$$

At the same time one can easily see from the definition of $X(t)$ and $\Delta(t)$ that

$$\begin{cases} X(t) = (-1)^N\chi_1 \ldots \chi_N \cdot t^N + X'(t) \\ \Delta(t) = \delta_1 \ldots \delta_N \cdot t^N + \Delta'(t) \end{cases} \quad \text{with } d(X'), d(\Delta') = N-1\,. \tag{6.22}$$

Input: Alphabets $\Delta = \{\delta_1, \ldots, \delta_N\}$ and $X = \{\chi_1, \ldots, \chi_N\}$.
Output: For all $m = 0 \ldots 2N-1$, a pair of polynomials (π_m, μ_m) satisfying (6.18).

For $m = 0$, the right hand side of the equality (6.18) is 1, i.e. the greatest common divisor of $\lambda_{-t}(Y)$ and $\lambda_{-t}(X)$. This implies that we can use the Generalised Euclidean algorithm as first step of our algorithm.

- **Step 0.1.** Consider the two polynomials $X(t)$ and $\Delta(t)$ of $\mathbf{R}[t]$ defined by setting

$$X(t) = \prod_{i=1}^{N} (1 - \chi_i\, t) \quad \text{and} \quad \Delta(t) = \prod_{i=1}^{N} (1 + \delta_i\, t).$$

- **Step 0.2.** Compute the unique polynomial $\pi_0(t)$ of $\mathbf{R}[t]$ of degree $d(\pi_0) \leq N-1$ such that

$$\pi_0(t)X(t) + \mu_0(t)\Delta(t) = 1$$

where $\mu_0(t)$ stands for some polynomial of $\mathbf{R}[t]$ of degree $d(\mu_0) \leq N-1$.

Suppose now that at `Step` $m-1$ we have found the polynomials $\pi_k(t)$ and $\mu_k(t)$ for all $k < m$. Then the following `Step` m provides us the next pair of polynomials $\pi_m(t)$ and $\mu_m(t)$ of degrees $\leq N-1$, satisfying the relation (6.18).

- **Step *m*.1.** We suppose that $0 < m < 2N$. Let then

$$c = \frac{[t^{N-1}](\mu_{m-1})}{\chi_1 \cdots \chi_N} = (-1)^{N-1} \frac{[t^{N-1}](\pi_{m-1})}{\delta_1 \ldots \delta_N}\ , \tag{6.19}$$

where $[t^{N-1}](\pi)$ stands for the coefficient of t^{N-1} in the polynomial $\pi(t)$.

- **Step *m*.2.** We then define

$$\begin{cases} \pi_m(t) = t\,\pi_{m-1}(t) + (-1)^N c\,\Delta(t) \\ \mu_m(t) = t\,\mu_{m-1}(t) - (-1)^N c\,X(t) \end{cases} \tag{6.20}$$

to obtain the required polynomials.

Algorithm 6.1: Calculating the polynomials π_m and μ_m

Substituting these four equations into (6.18) we obtain:

$$\begin{aligned} t^{m-1} &= \pi_{m-1}(t)X(t) + \mu_{m-1}(t)\Delta(t) \\ &= \Big((-1)^N a\,\chi_1 \dots \chi_N + b\,\delta_1 \dots \delta_N\Big)t^{2N-1} + O(t^{2N-2}) \end{aligned}$$

As the degree of the left-hand side of this equation is $m-1 < 2N-1$, we conclude that

$$(-1)^N a\,\chi_1 \dots \chi_N + b\,\delta_1 \dots \delta_N = 0.$$

Therefore we can define the coefficient c as

$$c = (-1)^N \frac{a}{\delta_1 \dots \delta_N} = \frac{b}{\chi_1 \cdots \chi_N}\ . \tag{6.23}$$

Let us now put $\pi_m(t) = t\,\pi_{m-1}(t) - d\Delta(t)$, where d is a coefficient such that $d(\pi_m) \leq N-1$. Indeed, substituting (6.21), (6.22), and (6.23) into this formula we obtain:

$$\begin{aligned} \pi_m(t) &= (-1)^{N-1} c\,\delta_1 \dots \delta_N\, t^N + t\pi'(t) - d\,\delta_1 \dots \delta_N t^N - d\,\Delta'(t) \\ &= \Big((-1)^{N-1}c - d\Big)\delta_1 \dots \delta_N\, t^N + \underbrace{t\,\pi'(t) - d\,\Delta'(t)}_{\deg\ \leq\ N-1}\ , \end{aligned}$$

thus it is sufficient to take $d = (-1)^{N-1}c$ in order to have $d(\pi_m) \leq N-1$. Applying the same reasoning to $\mu_m(t)$ we obtain the following two expressions:

$$\left\{ \begin{array}{rcl} \pi_m(t) &=& t\,\pi_{m-1}(t) - (-1)^{N-1} c\,\Delta(t) \\ \mu_m(t) &=& t\,\mu_{m-1}(t) - (-1)^N c\,X(t) \end{array} \right. , \tag{6.24}$$

where $d(\pi_m), d(\mu_m) \leq N-1$. In order to complete our proof we have to check that these two polynomials satisfy (6.18):

$$\begin{aligned} t^m &= t\,\pi_{m-1}(t)X(t) + t\,\mu_{m-1}(t)\Delta(t) \\ &= \Big(\pi_m(t) + (-1)^{N-1} c\,\Delta(t)\Big)X(t) + \Big(\mu_m(t) + (-1)^N c\,X(t)\Big)\Delta(t) \\ &= \pi_m(t)X(t) + \mu_m(t)\Delta(t) \end{aligned}$$

This ends our proof. ∎

Note 6.2 *We recall that* $\pi_m(0) = P_m^{(N)}$.

6.3 Combinatorial interpretation

6.3.1 A special case

Recall that originally $P_m^{(N)}$ has been defined as the m-th coefficient of the decomposition of $P(U - V < y)$ into an exponential series (cf. Section 6.2):

$$P(U - V < \varepsilon) = \sum_{m=0}^{\infty} P_m^{(N)} \times \frac{\varepsilon^m}{m!}\,,$$

and therefore, taking $y = 0$, we obtain (see also (6.13))

$$P(U < V) = P_0^{(N)} = \frac{\sum\limits_{\lambda \subset (N^{N-1})} s_{(\lambda,N)}(\Delta) s_{\overline{(\lambda,N)}}(X)}{\prod\limits_{1 \le i,j \le N} (\chi_i + \delta_j)} \,. \tag{6.25}$$

This expression for the probability $P(U < V)$ has been obtained in [31], while in [54] it has been given a combinatorial interpretation. Indeed, as $\lambda \subset (N^{N-1})$ we have $\lambda_i \le N$ for all i, and thus (λ, N) is also a partition.

It is well known that a Schur function over an alphabet A indexed by some partition λ can be expressed as a sum

$$s_\lambda(A) = \sum_{t_\lambda} m(t_\lambda) \,,$$

where t_λ runs through all possible Young tableaux over A of shape λ, and $m(t_\lambda)$ is the monom obtained by taking the product of all elements of A contained in t_λ. For example,

$$s_{(1,2)}(\{\chi_1, \chi_2\}) = \chi_1^2 \chi_2 + \chi_1 \chi_2^2 \,,$$

which corresponds to

$$\begin{array}{|c|c|}\hline \chi_2 & \multicolumn{1}{c}{} \\ \hline \chi_1 & \chi_1 \\ \hline \end{array} \quad + \quad \begin{array}{|c|c|}\hline \chi_2 & \multicolumn{1}{c}{} \\ \hline \chi_1 & \chi_2 \\ \hline \end{array} \,.$$

In this manner one can represent the numerator of the fraction in (6.25) as a sum of the monoms corresponding to (N^N) square tabloids consisting of a Young tableau over the alphabet Δ and a complimentary one over X. For example,

$$P_0^{(2)} = \frac{\chi_1\chi_2(\delta_1^2 + \delta_1\delta_2 + \delta_2^2) + (\chi_1 + \chi_2)(\delta_1^2\delta_2 + \delta_1\delta_2^2) + \delta_1^2\delta_2^2}{(\chi_1 + \delta_1)(\chi_1 + \delta_2)(\chi_2 + \delta_1)(\chi_2 + \delta_2)}$$

corresponds to

$$\begin{array}{|c|c|}\hline \chi_2 & \chi_1 \\ \hline \delta_1 & \delta_1 \\ \hline \end{array} + \begin{array}{|c|c|}\hline \chi_2 & \chi_1 \\ \hline \delta_1 & \delta_2 \\ \hline \end{array} + \begin{array}{|c|c|}\hline \chi_2 & \chi_1 \\ \hline \delta_2 & \delta_2 \\ \hline \end{array} + \begin{array}{|c|c|}\hline \delta_2 & \chi_1 \\ \hline \delta_1 & \delta_1 \\ \hline \end{array} + \begin{array}{|c|c|}\hline \delta_2 & \chi_2 \\ \hline \delta_1 & \delta_1 \\ \hline \end{array} + \begin{array}{|c|c|}\hline \delta_2 & \chi_1 \\ \hline \delta_1 & \delta_2 \\ \hline \end{array} + \begin{array}{|c|c|}\hline \delta_2 & \chi_2 \\ \hline \delta_1 & \delta_2 \\ \hline \end{array} + \begin{array}{|c|c|}\hline \delta_2 & \delta_2 \\ \hline \delta_1 & \delta_1 \\ \hline \end{array} \,.$$

For an arbitrary m such that $0 < m < 2N$ it is possible that $(\lambda, N - m)$ (see again (6.13)) is not a partition. In order to obtain an analogous representation of $P_m^{(N)}$ we will have to introduce a more complex combinatorial object — *square tabloid with ribbon*.

6.3.2 Square tabloids with ribbons

Definition 6.3 *A* ribbon *in a Young diagram is a connected chain of boxes not containing a* 2×2 *square such that any box has at most two neighbours (see Figure 6.1). The number of boxes in a ribbon is its* length.

The examples *a–c* in the Figure 6.1 are correct ribbons, while *d–f* are not. In this note we will only consider those that start in the lower right-hand corner of the square, and go to the North-West (examples *b*, *c*). For the sake of simplicity we will omit these two conditions when refering to ribbons.

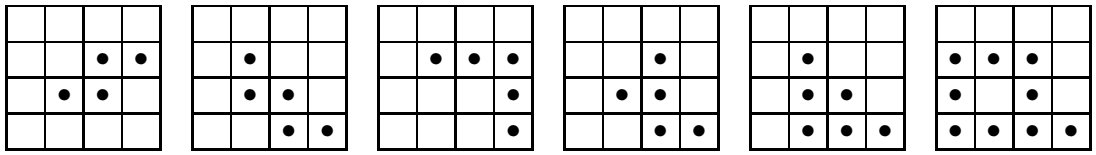


Figure 6.1: Several examples and counter-examples for the Definition 6.3 of ribbons.

We shall denote by $\mathcal{R}^{(N)}$ the set of all such ribbons in (N^N). Denoting by $\mathcal{R}_m^{(N)}$ the subset of $\mathcal{R}^{(N)}$ consisting of ribbons of a given length m, one obtains the following obvious decomposition:

$$\mathcal{R}^{(N)} = \bigcup_{m=1}^{2N-1} \mathcal{R}_m^{(N)}.$$

In a way analoguous to the one used to represent Young diagrams as a partition of an integer, a ribbon is fully described by a sequence $(r_1, \ldots, r_k)$, where r_i is the number of boxes in its i-th row. For example, the ribbons b and c from Figure 6.1 are (2, 2, 1) and (1, 1, 3) correspondingly.

For any given $r \in \mathcal{R}^{(N)}$ we shall consider all Young diagrams $\lambda \subset (N^N)$, such that $\lambda \cup r$ is also a Young diagram, together with $\mu = \overline{\lambda \cup r}$ — diagram complementary to the latter. Here the union '$\cup$' is taken in its geometrical sense — the union of two sets of boxes.

We can introduce one of our bijection's domains by considering all possible triplets consisting of a ribbon and two Young tableaux of shapes λ and μ on the alphabets $\Delta = \{\delta_1, \ldots, \delta_N\}$ and $X = \{\chi_1, \ldots, \chi_N\}$ correspondingly, such that put together they form a complete square:

$$\mathcal{T}_m^{(N)} = \{(t_\lambda, t_\mu, r) | r \in \mathcal{R}_m^{(N)}, \lambda \cup r \subset (N^N), \mu = \overline{\lambda \cup r}\},$$

where $1 \leq m \leq 2N-1$, λ and μ are the shapes of t_λ, and t_μ (see Figure 6.2).

From the combinatorial point of view this construction means that we take a square N by N, cut out a ribbon of length m, then split the rest into two Young tableaux over alphabets Δ and X.

δ_4	χ_4	χ_3	χ_1
δ_3	•	χ_3	χ_1
δ_2	•	•	χ_4
δ_1	δ_2	•	•

Figure 6.2: A typical element of $\mathcal{T}_4^{(5)}$.

Proposition 6.4 *We have the following representation for* $P_m^{(N)}$

$$P_m^{(N)} = \frac{\sum_{t \in \mathcal{T}_m^{(N)}} (-1)^{h(t)-1} m(t)}{\prod_{1 \leq i,j \leq N} (\chi_i + \delta_j)},$$

where $m(t)$ is the monom corresponding to t, and $h(t)$ is the height of the ribbon of t defined as follows. If $t = (t_\lambda, t_\mu, r) \in \mathcal{T}_m^{(N)}$ is a square tabloid with ribbon, and $r = (r_1, \ldots, r_N)$ then $h(t) = \max\{i | r_i > 0\}$.

Proof. This proposition becomes an obvious consequence of (6.13), if we consider the fact that

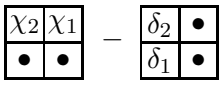
$$s_{(\ldots,i,j,\ldots)}(A) = -s_{(\ldots,j+1,i-1,\ldots)}(A)\,.$$

■

Example 6.5

$$P_2^{(2)} = \frac{\chi_1\chi_2 - \delta_1\delta_2}{(\chi_1+\delta_1)(\chi_1+\delta_2)(\chi_2+\delta_1)(\chi_2+\delta_2)}\,,$$

can be represented as the following difference



■

6.3.3 Description of the bijection

In the following we shall construct a bijection between the square tabloids with ribbons introduced in the previous section, and a subset $\mathfrak{M}_m^{(N)}$ of $\mathcal{M}_{N\times(N+m)}$ [8] that we will define later on. We shall split a matrix from this set into two parts: the one on the left-hand side containing N columns, and the one on the right-hand side — m columns. The right part is the one that will eventually generate the ribbon in the corresponding tableau, and has only one 1 in each of its columns. Meanwhile the left part will be responsible for the tableau t_λ, and has one 1 for each of its boxes. Going back to the example of Figure 6.2, the corresponding matrix would be

$$\left(\begin{array}{cccc|ccccc} 0&0&0&0&0&0&1&0&1\\ 1&1&0&0&1&0&0&1&0\\ 0&1&0&0&0&1&0&0&0\\ 0&0&1&1&0&0&0&0&0 \end{array}\right)\,.$$

Algorithm 6.2 takes a square tabloid $T \in \mathcal{T}_m^{(N)}$ on input and constructs the corresponding matrix from $\mathcal{M}_{N\times(N+m)}$. As this algorithm is based on the Knuth correspondence (cf. [34, 51, 54]), [9] it can be easily seen that it is reversible, thus, denoting by $\mathfrak{M}_m^{(N)}$ the image of the mapping defined by this algorithm, we obtain a bijection between $\mathfrak{M}_m^{(N)}$ and $\mathcal{T}_m^{(N)}$.

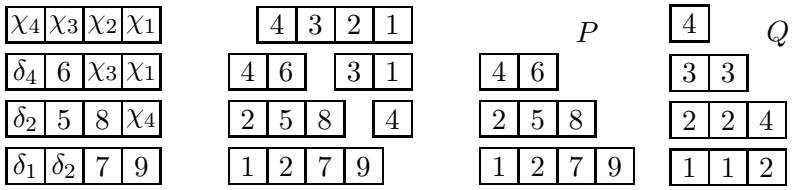
[8] In the following we will only consider $\{0,1\}$-matrices, therefore we use $\mathcal{M}_{m\times n}$ as a shorthand notation for $\mathcal{M}_{m\times n}(\{0,1\})$.

[9] This correspondence, as well as its extension that we use in Step 3 of Algorithm 6.2 (and also in Step 2 of Algorithm 6.3) is presented in Section D.1.1 of Appendix D.

Input: $T = (t_\lambda, t_\mu, r)$ — a square tabloid in $\mathcal{T}_m^{(N)}$.
Output: A $\{0,1\}$-matrix $M \in \mathcal{M}_{N\times(N+m)}$.

- **Step 1.** Number each box of r starting with $N+1$ and up to $N+m$ from bottom to top and from left to right (see Figure 6.3-*a*).
- **Step 2.** Replace all δ_i in t_λ and χ_i in t_μ by i, and join t_λ with r to obtain two Young tableaux P and $\overline{Q}$ of shapes $\lambda \cup r$ and $\mu = \overline{\lambda \cup r}$ correspondingly (Figure 6.3-*b*).
- **Step 3.** Apply the same procedure as in [54] to obtain a tableau Q of a shape conjugated to $\lambda \cup r$ (Figure 6.3-*c*; see Example 6.6 below, Section D.1.1 or [54] for a formal description).
- **Step 4.** Finally apply Knuth's bijection based on column bumping to the pair (P, Q) of Young tableaux of conjugate forms to obtain a matrix from $\mathcal{M}_{N\times(N+m)}$ (Figure 6.3-*d*).

Algorithm 6.2: Construction of a $\{0,1\}$-matrix from a square tabloid



a b c

$$\left(\begin{array}{cccc|ccccc} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{array}\right)$$

d

Figure 6.3: Applying the algorithm to an element of $\mathcal{T}_4^{(5)}$.

Example 6.6
Let us elaborate on the example given along the algorithm. We start with the following square tabloid from $\mathcal{T}_4^{(5)}$:

χ_4	χ_3	χ_2	χ_1
δ_4	•	χ_3	χ_1
δ_2	•	•	χ_4
δ_1	δ_2	•	•

As the size of the square is 4, and the length of the ribbon is 5, the classical numbering for the ribbon goes from 5 to 9 and is shown in Figure 6.3-*a*. Re-labelling the rest of the tabloid and re-arranging it as indicated in the Step 2 of Algorithm 6.2 we obtain the two Young tableaux shown in Figure 6.3-*b*.

In Step 3 we take the upper right-hand corner tableau

4			
3			
2	3		
1	1	4	

and we transform it into another one of complementary shape by applying the following procedure.

First of all we consider this tableau as having four columns — the last one being empty. Then we form another tableau by putting in its first column all the numbers between 1 and 4 that are not in the last column of this one. As the last column of the original tableau is empty, we put in the first column of the new one all numbers 1–4. In the second column we put all the numbers that are not in the third one of the original tableau (i.e. 1–3), etc. Thus we obtain the tableau Q from Figure 6.3-*c*.

The only thing left to do now is to apply Knuth's bijection to the pair (P, Q) to obtain the matrix in Figure 6.3-*d*. ■

Note 6.7 We denote by $\mathfrak{M}^{(N)} = \bigcup_{m=1}^{2N-1} \mathfrak{M}_m^{(N)}$ the set of all matrices that can be obtained by applying this algorithm.

As it has been mentioned in the beginning of the section, it can be easily seen that Algorithm 6.2 is reversible. More precisely, we have the following reciprocal algorithm.

Input: A $\{0,1\}$-matrix $M \in \mathfrak{M}_m^{(N)}$.
Output: $T = (t_\lambda, t_\mu, r)$ — a square tabloid in $\mathcal{T}_m^{(N)}$.

- **Step 1.** Applying Knuth's bijection in the opposite direction we can transform any matrix from $\mathcal{M}_{N\times(N+m)}$ into a pair of Young tableaux P and Q of conjugated shapes: on the alphabets $\{1,\dots,N+m\}$ and $\{1,\dots,N\}$ correspondingly.
- **Step 2.** The fact that M belongs to $\mathfrak{M}_m^{(N)}$ implies that both P and Q fit into (N^N), and thus we can again apply to Q the same procedure as in [54] to obtain a new Young tableau t_μ of the shape complementary to that of P.
- **Step 3.** Once again referring to the fact that M belongs to $\mathfrak{M}_m^{(N)}$, we can state that in P there is exactly one occurence of each one of $N+1,\dots,N+m$, and that the corresponding boxes form a ribbon r numbered from bottom to top and from left to right. Moreover, this ribbon can be cut out of P leaving a Young tableau t_λ on the alphabet $\{1,\dots,N\}$. (See Section 6.3.4 for conditions on $\{0,1\}$-matrices defining $\mathfrak{M}_m^{(N)}$ explicitely.)
- **Step 4.** To finalise our algorithm it is sufficient to replace all entries i in t_λ with δ_i, and in t_μ — with χ_i.

Algorithm 6.3: Construction of a square tabloid from a $\{0,1\}$-matrix

Example 6.8
To reverse Example 6.6 we start with the matrix

$$\left(\begin{array}{cccc|ccccc} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{array}\right)$$

and we transform it into a two-row array representing the positions of 1's: in each column of the array the element in the first row is the row number, and the one in the second row — the column number of a position containing 1. We obtain therefore the following array:

$$\left(\begin{array}{ccccccccc} 1 & 1 & 2 & 2 & 2 & 3 & 3 & 4 & 4 \\ 7 & 9 & 2 & 5 & 8 & 1 & 6 & 2 & 4 \end{array}\right)$$

Applying Knuth's bijection consists now in forming one Young tableau by column-bumping in the elements of the second row of the array from left to right, and placing the corresponding elements of the first row into a Young tableau of the conjugated form. This results exactly in the pair of tableaux shown in Figure 6.3-*c*.

Applying the procedure described in Example 6.6 we obtain a pair of tableaux of complimentary shapes

$$\begin{array}{|c|c|c|c|}\hline 4 & 6 & & \\ \hline 2 & 5 & 8 & \\ \hline 1 & 2 & 7 & 9 \\ \hline \end{array}\ , \qquad \begin{array}{|c|c|c|c|}\hline 4 & & & \\ \hline 3 & & & \\ \hline 2 & 3 & & \\ \hline 1 & 1 & 4 & \\ \hline \end{array}\ .$$

Finally, re-arranging and re-numbering these two accordingly we end up with the tabloid

χ_4	χ_3	χ_2	χ_1
δ_4	$\bullet$	χ_3	χ_1
δ_2	$\bullet$	$\bullet$	χ_4
δ_1	δ_2	$\bullet$	$\bullet$

that was the starting point of Example 6.6. ■

We will denote the square tabloid obtained by applying Algorithm 6.3 to a matrix M by $\Phi(M)$. Note that $\Phi(M)$ is also defined on some matrices that do not belong to $\mathfrak{M}^{(N)}$, but in that case $\Phi(M) \notin \mathcal{T}^{(N)}$.

6.3.4 Characterisation of matrices in $\mathfrak{M}^{(N)}$

In the previous section, we presented a bijective mapping from $\mathfrak{M}^{(N)}$ to $\mathcal{T}^{(N)}$. This mapping being defined by an algorithm, we can explicitely calculate its image given an element of $\mathfrak{M}^{(N)}$. However, this set is only defined implicitely as the image of the mapping induced by the Algorithm 6.3. This section is therefore devoted to providing explicit conditions on a matrix from $\mathcal{M}_{N\times(N+m)}$ to be an element of $\mathfrak{M}_m^{(N)}$.

We will use an equivalent of Green's theorem that gives us a way of calculating the shape of the Young tableau obtained by the Robinson-Schensted correspondence from a word on the corresponding alphabet. As Robinson-Schensted correspondence is the base of Knuth's bijection, this theorem can be reformulated in terms of $\{0,1\}$-matrices to be applied to the latter.

First of all, let us introduce a few notations. Let M be a $\{0,1\}$-matrix as considered above. We shall denote by $R(M,k)$ the largest possible number of 1's in M that can be arranged in k disjoint (possibly empty) sequences going North-east. Here, as in [34], we will begin each word indicating a direction with a capital letter if the sequence goes strictly in that direction, and with a small one if it does so weakly. Here, for example, "*N*orth-*e*ast" stands for "strictly North and weakly East", i.e. if two 1's in positions (i_1,j_1) and (i_2,j_2) (where $i_2 \leq i_1$) belong to the same sequence then we have $i_2 < i_1$ (strictly North), and $j_1 \leq j_2$ (weakly East) (see Figure 6.4). By convention $R(M,0)=0$. Taking $\lambda=(\lambda_1,\ldots,\lambda_N)$ to be the shape of the tableau P obtained

$$\begin{array}{ccc} \left(\begin{array}{ccc} 1 & 0 & \boxed{1} \\ 1 & \boxed{1} & 0 \\ 0 & \boxed{1} & 0 \end{array}\right) & \qquad & \left(\begin{array}{ccc} 1 & 0 & \boxed{1} \\ \boxed{1} & \boxed{1} & 0 \\ 0 & 1 & 0 \end{array}\right) \\ \text{a} & & \text{b} \end{array}$$

Figure 6.4: The boxed sequence of 1's goes North-east on (a), but not on (b).

from M by Knuth correspondence, we can state the following theorem:

Theorem 6.9 (Green) *In the above notations, one has:*

$$\forall k = 1 \ldots N, \quad R(M,k) - R(M,k-1) = \lambda_k.$$

Now, if we denote by $(\lambda_1,\ldots,\lambda_N)$ and $(r_1,\ldots,r_N)$ the shapes of t_λ, and r correspondingly, where $\Phi(M)=(t_\lambda,t_\mu,r)$, and taking M' to be the left-hand $N\times N$ square part of M, we obtain automatically

$$\forall k = 1 \ldots N, \quad \left\{ \begin{array}{lcll} R(M',k) & - & R(M',k-1) & = & \lambda_k \\ R(M,k) & - & R(M,k-1) & = & \lambda_k + r_k \end{array} \right. . \tag{6.26}$$

In other words, (6.26) provides us a way of calculating the shapes of t_λ and r given a $\{0,1\}$-matrix M. This immediatly delivers the first condition to be satisfied in order for M to be in $\mathfrak{M}^{(N)}$.

Condition 6.10 *Let $\Phi(M)=(t_\lambda,t_\mu,r)$ and $(\lambda_1,\ldots,\lambda_N)$ and $(r_1,\ldots,r_N)$ be the values provided by (6.26), then for r to be a correct ribbon as described by the Definition 6.3, it is necessary that*

- *there exists $h \in [0,N]$ such that $r_k > 0$ for any $k \in [1,h]$, and $r_k = 0$ when $k > h$;*
- *$\lambda_k + r_k = \lambda_{k-1} + 1$ for all $k \in [2,h]$;*
- *$\lambda_1 + r_1 = N$.*

The above condition, when fulfilled, guarantees that the shape of the ribbon is correct. It rests therefore to ensure that it's numbering is the required one, i.e. all boxes forming the ribbon must be numbered from bottom to top, and from left to right by the sequence $N+1,\ldots,N+m$.

First of all, there has to be exactly one box in tableau P for each number between $N+1$ and $N+m$. This is obviously guaranteed by the following condition.

Condition 6.11 *For any $k \in [N+1, N+m]$ there is exactly one 1 in the k-th column of M.*

Example 6.12
Let us consider the following matrix

$$M = \left(\begin{array}{cccc|ccccc} 0 & 0 & 0 & 0 & 0 & 0 & 0 & \boxed{1} & \textcircled{1} \\ \boxed{1} & 0 & 0 & 0 & 1 & 0 & \textcircled{1} & 0 & 0 \\ \boxed{1} & 1 & 0 & 0 & 0 & \textcircled{1} & 0 & 0 & 0 \\ \boxed{1} & 0 & \textcircled{1} & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right)$$

Considering the left-hand side of the matrix we can see that

$$\begin{array}{ll} R(M',1) = 3 & \text{(boxed sequence)}, \\ R(M',2) = 4 & \text{(boxed and circled sequences)}, \\ R(M',3) = 5 & \text{(boxed, circled, and unmarked sequences)}. \end{array}$$

As there are no more 1's left we conclude that $\lambda = (3,1,1)$. Now if we consider the whole matrix we obtain

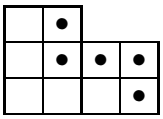
$$\begin{array}{ll} R(M,1) = 4 & \text{(boxed sequence)}, \\ R(M,2) = 8 & \text{(boxed and circled sequences)}, \\ R(M,3) = 10 & \text{(boxed, circled, and unmarked sequences)}. \end{array}$$

and therefore

$$\lambda_1 + r_1 = 4, \quad \lambda_2 + r_2 = 4, \quad \lambda_3 + r_3 = 2,$$

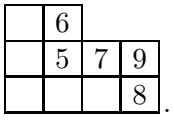
i.e. . $r = (1,3,1)$. It is easy to see now that both Condition 6.10 and Condition 6.11 are verified.

We can deduce that the lower left-hand side tableau and the ribbon in the image of M will have the following shapes:



■

In the following we will require some additional notions. As we have seen above, given a matrix $M \in \mathcal{M}_{N\times(N+m)}$ and provided that Condition 6.10 is satisfied, we can calculate the shape of all elements of $\Phi(M)$. Thus, in particular, we know the desired classical numbering of the ribbon. For instance the numbering of the ribbon in the Example 6.12 should be as follows:



Definition 6.13

- *We will refer as* columns *of the ribbon to the sequences of numbers in each column of boxes of the ribbon in the classical numbering ((5,6), (7), and (8,9) in the example above).*
- *Two numbers $i, j \in [N+1, \ldots, N+m]$ are said to be in the* same level l*, if each one of them is exactly l boxes down from the top of its column in the classical numbering of the ribbon. We will refer as* levels *to maximal sets of numbers being in the same level. We say that level l_1 is* higher *than level l_2 if $l_1 < l_2$ — in other words if level l_1 is closer to the top.*

Example 6.14
Taking on the previous example, we can say that numbers 6, 7, and 9 form level 0 in this ribbon, and 5 and 8 — level 1. ■

Example 6.15
More generally, if we have a ribbon numbered as follows,

	9		
	6	7	8
			5

we shall say that its *columns in the classical numbering* are (5,6), (7), and (8,9); its *levels [in the classical numbering]* are (6,7,9) and (5,8); and its *columns in the actual numbering* are (6,9), (7), and (5,8). ■

Our goal now is to ensure that the numbering of the ribbon, obtained by applying Algorithm 6.3, is the same as the classical one.

Condition 6.16

1. *The 1's corresponding to each column of the ribbon form a sequence going south-East.*
2. *The 1's corresponding to each level of the ribbon form a sequence going North-East.*

We can now state the following theorem.

Theorem 6.17 (Caracterisation of $\mathfrak{M}_m^{(N)}$) *Let $M \in \mathcal{M}_{N\times(N+m)}$ be a $\{0,1\}$-matrix. $M \in \mathfrak{M}_m^{(N)}$ if and only if M satisfies all three conditions 6.10–6.16.*

Proof of the main theorem

It is obvious that $M \in \mathcal{M}_{N\times(N+m)}$ satisfies both Condition 6.10 and Condition 6.11 if and only if there is exactly one box in $\Phi(M)$ numbered with each one of $N+1,\dots,N+m$ and these boxes form a correct ribbon in the sense of Definition 6.3. Thus, we only have to show that, when these two conditions are fulfilled, the Condition 6.16 is equivalent to the ribbon in $\Phi(M)$ being numbered correctly.

Recall that Algorithm 6.3 is based on Robinson-Schensted-Knuth correspondence, which has column bumping as its building block. Therefore, when applying this algorithm to M, we perform a certain number Θ of column bumpings. Thus, for each $\theta \in [0,\Theta]$, one can consider a Young tableau T_θ obtained after bumping in θ boxes.

Definition 6.18 *For $a \in [N+1, N+m]$, we shall denote by $d_\theta(a)$ the column of T_θ containing the box numbered a. We take $d_\theta(a) = 0$ if a is not in T_θ.*

Note 6.19 There is no ambiguity in the definition of $d_\theta(a)$ due to Condition 6.11.

Example 6.20
Consider the matrix

$$M = \begin{array}{c} \begin{matrix} 1 & 2 & 3 & 4 & 5 & 6 \end{matrix} \\ \left(\begin{array}{ccc|ccc} 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \end{array}\right) \end{array}.$$

To obtain the lambda part and the ribbon of $\Phi(M)$ we have to successively perform column bumping on all letters of the word 562413. This creates the following chain of Young tableaux:

$$\begin{array}{|c|}\hline 5\\\hline\end{array} \rightarrow \begin{array}{|c|}\hline 6\\\hline 5\\\hline\end{array} \rightarrow \begin{array}{|c|c|}\hline 6 & \\\hline 2 & 5\\\hline\end{array} \rightarrow \begin{array}{|c|c|}\hline 4 & 6\\\hline 2 & 5\\\hline\end{array} \rightarrow \begin{array}{|c|c|}\hline 4 & 6\\\hline 1 & 2 & 5\\\hline\end{array} \rightarrow \begin{array}{|c|c|c|}\hline 3 & 4 & 6\\\hline 1 & 2 & 5\\\hline\end{array} \rightsquigarrow \begin{array}{|c|c|c|}\hline 3 & \bullet & \bullet\\\hline 1 & 2 & \bullet\\\hline\end{array}$$

We have therefore

$$d_1(4) = d_2(4) = d_3(4) = 0 \quad d_4(4) = d_5(4) = 1 \quad d_6(4) = 2\,.$$

■

We will show now that Condition 6.16.1 is equivalent to the following proposition: at any stage of the column bumping process, the box containing a number from any column of the ribbon in the classical numbering will be no further in the tableau than any box containing another number from the same column but a lower level.

Example 6.21
Taking on the previous example, one can easily verify that, for any $0 \leq \theta \leq 6$, we have $d_\theta(5) \geq d_\theta(6)$.

Lemma 6.22 *Suppose that Conditions 6.10 and 6.11 are satisfied, and let r be the ribbon of $\Phi(M)$. Then for any $a \geq N+1$, such that a and $a+1$ belong to the same column of r in its classical numbering, the relation*

$$d_\theta(a) \geq d_\theta(a+1)$$

is invariant over $\theta \in [0, \Theta]$ such that $d_\theta(a) > 0$.

Proof. Suppose that we have $d_\theta(a) < d_\theta(a+1)$ and $d_{\theta+1}(a) \geq d_{\theta+1}(a+1)$. This means that during the $(\theta+1)^{st}$ column bumping a bumps out some c to take its place in the same column where $a+1$ is (see Figure 6.5-*a*). In this case we have $a \leq c$ (by definition of column bumping

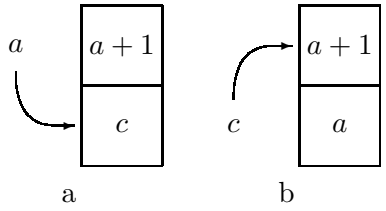


Figure 6.5: Snapshots of column-bumping process for Lemma 6.22.

process), and $c < a+1$ (by definition of Young tableaux), which means that $a = c$. However this is impossible because of the Condition 6.11.

On the other hand, if at some point we have $d_\theta(a) \geq d_\theta(a+1)$ and $d_{\theta+1}(a) < d_{\theta+1}(a+1)$, it means that a and $a+1$ are in the same column, and $a+1$ is bumped out by some c (see Figure 6.5-*b*). Again, we can notice that, by definition of column bumping, $c \leq a+1$ and $c > a$ implying that $c = a+1$, which contradicts Condition 6.11. ■

Note 6.23 It can be easily observed that Condition 6.16.1 is equivalent to saying that, for any a as in Lemma 6.22, a is bumped in earlier than $a+1$, i.e. $d_{\theta_a}(a) > d_{\theta_a}(a+1) = 0$, where $\theta_a = \min\{\theta | d_\theta(a) > 0\}$.

Corollary 6.24 *In the conditions of Lemma 6.22, Condition 6.16.1 implies that* $d_\theta(a) \geq d_\theta(a+1)$ *for all* $\theta \in [0, \Theta]$.

Proof. This corollary is a trivial consequence of Lemma 6.22 and Note 6.23. ∎

Corollary 6.25 $M \in \mathfrak{M}^{(N)}$ *implies Condition 6.16.1.*

Proof. Obviously, if $M \in \mathfrak{M}^{(N)}$, we have $d_\Theta(a) = d_\Theta(a+1)$ for any a as in Lemma 6.22. As Conditions 6.10 and 6.11 also hold in this case, we can deduce from Lemma 6.22 and Note 6.23 that Condition 6.16.1 is verified. ∎

We have shown therefore that Condition 6.16.1 is necessary for M to belong to $\mathfrak{M}^{(N)}$. The next step is to prove that Condition 6.16.2 is also necessary. We can remark the following by analogy with the Note 6.23.

Note 6.26 Condition 6.16.2 is equivalent to saying that, for any a and b ($a < b$) belonging to the same level in the ribbon, b is bumped in earlier than a, i.e. $0 = d_{\theta_b}(a) < d_{\theta_b}(b)$, where $\theta_b = \min\{\theta | d_\theta(b) > 0\}$.

Let us now introduce a few notations. We consider all columns of the ribbon in the sense of Definition 6.13 numbered from left to right. The conjugated shape of the ribbon is a sequence $(c_1, \ldots, c_N)$, where c_i is the number of boxes in the ribbon's i-th column. In the following discussion we will only consider those columns i that have $c_i > 0$.

For each column i we will denote by t_i its top box in the classical numbering. In Example 6.12 the conjugated shape of the ribbon is (2,1,2), and $t_1 = 6$, $t_2 = 7$, and $t_3 = 9$.

Lemma 6.27 *Suppose that Conditions 6.10, 6.11, and 6.16.1 are satisfied, and let* r *be the ribbon of* $\Phi(M)$. *Suppose also that for any* $t_{i-1} - k$, *and* $t_i - k$ — *two numbers in the adjacent columns and the same level of* r *in its classical numbering — we have*

$$d_{\theta+1}(t_{i-1} - k) < d_{\theta+1}(t_i - k)\,,$$

then the same is correct if we replace $\theta + 1$ *by* θ.

Proof. Suppose this is not the case. Then we have at the same time

$$\begin{array}{c} d_\theta(t_{i-1} - k) \geq d_\theta(t_i - k) \\ d_{\theta+1}(t_{i-1} - k) < d_{\theta+1}(t_i - k) \end{array}\quad ,$$

which means that $t_i - k$ and $t_{i-1} - k$ are in the same column of T_θ, and during the $(\theta+1)$-st column bumping some t bumps $t_i - k$ out (see Figure 6.6). From the definition of Young tableaux and column bumping we can conclude then that $t \in [t_{i-1} - k + 1, \ldots, t_i - k - 1]$. However, $t \in [t_{i-1} + 1, \ldots, t_i - k - 1]$ implies that t is in the column i of the final ribbon, and thus, by Condition 6.16.1 and Lemma 6.22, we have $d_\theta(t) \geq d_\theta(t_i - k)$, which is impossible, as t bumps out $t_i - k$ at step $\theta + 1$. Thus we can deduce that

$$t \in [t_{i-1} - k + 1, \ldots, t_{i-1}]\,. \tag{6.27}$$

Notice here that, for $k = 0$, this interval is empty and we obtain a contradiction. We can therefore continue our proof inductively.

Let us suppose that we have proven the assertion of the lemma for all levels higher than k. To prove it for k notice that (6.27) implies that t is in the column $i - 1$ of the classical

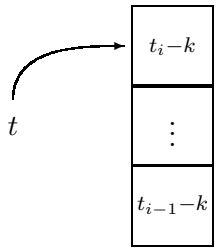


Figure 6.6: Snapshot of column-bumping process for Lemma 6.27.

numbering of the ribbon, and thus $t = t_{i-1} - k'$, where $0 \leq k' < k$. By the assumption of the induction, we then have

$$d_\theta(t_i - k) - 1 = d_\theta(t) = d_\theta(t_{i-1} - k') < d_\theta(t_i - k') \leq d_\theta(t_i - k) \ , \tag{6.28}$$

and therefore

$$d_{\theta+1}(t) = d_{\theta+1}(t_{i-1} - k') < d_{\theta+1}(t_i - k') = d_\theta(t_i - k') = d_\theta(t_i - k) = d_{\theta+1}(t) \ . \tag{6.29}$$

The inequality in (6.29) is the assumption of the lemma, the first equality is trivial, the second is due to $(t_i - k')$'s not changing its position during $(\theta + 1)$-st column bumping, the third is a consequence of (6.28), and the last represents the fact that t bumps out $t_i - k$. The relation (6.29) being contradictory proves the lemma. ∎

Let us now generalise Lemma 6.27 to any pair of numbers in the same level of the ribbon.

Lemma 6.28 *Suppose that Conditions 6.10, 6.11, and 6.16.1 are satisfied, and let r be the ribbon of $\Phi(M)$. Suppose also that for any $i < j$ and any $t_i - k$, and $t_j - k$ — two numbers in the same level of r — we have*

$$d_{\theta+1}(t_i - k) < d_{\theta+1}(t_j - k) \, ,$$

then the same is correct if we replace $\theta + 1$ by θ.

Proof. First of all notice that without loss of generality we can assume that $t_j - k$ and $t_i - k$ are adjacent in the level k of r, i.e. for any h, such that $i < h < j$, there is no box in the level k of the column h of r. Observe also that Lemma 6.27 is a special case of this one with $j = i + 1$, and therefore we only have to discuss the case were $j - i > 1$.

As before we suppose that we have proven the assertion of the lemma for all levels higher than k. We shall suppose that it is not satisfied for level k, and obtain a contradiction. Indeed, using the same reasoning as for (6.27), we can show that

$$t \in [t_i - k + 1, \ldots, t_{j-1}] \, , \tag{6.30}$$

which implies that t belongs to some column i' where $i \leq i' < j$, and therefore $t = t_{i'} - k'$ with some $k' \in [0, \ldots, k-1]$. Then we have the following chain of relations:

$$d_\theta(t_j - k) - 1 = d_\theta(t) = d_\theta(t_{i'} - k') < d_\theta(t_j - k') \leq d_\theta(t_j - k) \tag{6.31}$$

that is analogous to (6.28). We obtain a contradiction in the same manner as in (6.29), which proves the lemma. ∎

Corollary 6.29 *$M \in \mathfrak{M}^{(N)}$ implies the Condition 6.16.2.*

Proof. As we have seen before, $M \in \mathfrak{M}^{(N)}$ implies both Condition 6.10 and Condition 6.11, and by Corollary 6.25 also the Condition 6.16.1. It can also be easily observed that the assumption of Lemma 6.28 holds for $\theta + 1 = \Theta$, and therefore applying this lemma inductively, we prove the validity of Condition 6.16.2 (cf. Note 6.26). ∎

Collecting together Corollaries 6.25 and 6.29, we obtain the following proposition.

Proposition 6.30 *$M \in \mathfrak{M}^{(N)}$ implies all the conditions 6.10–6.16.*

$$- o -$$

To show that Conditions 6.10–6.16 are sufficient for $M \in \mathfrak{M}^{(N)}$, we make use of the notion of *plactic equivalence* described in Section D.1.2 of Appendix D.

Let us again consider a matrix $M \in \mathcal{M}_{N\times(N+m)}$, and $\Phi(M) = (t_\lambda, t_\mu, r)$ — the square tabloid obtained by applying Algorithm 6.3 to M.

If we take u to be the word obtained by reading the positions (columns) of 1's in M, then, denoting by T_u the Young tableau obtained by applying column bumping to the word u, we have by definition of Robinson-Schensted-Knuth correspondence

$$T_u = t_\lambda \cup r\,,$$

which implies by Proposition D.5 that

$$\overline{u} \equiv w(T_u) = w(t_\lambda \cup r)\,,$$

where $\overline{u}$ is the mirror image of u. Applying now Theorem D.8 to this equivalence and the interval $[N+1, N+m]$ we obtain

$$\overline{u''} \equiv w(r)\,,$$

where u'' is the word obtained by reading the positions of 1's in M'' — the right-hand part of M —, and $w(r)$ is the the restriction of the tableau word corresponding to $t_\lambda \cup r$ to the ribbon r. Therefore, by Theorem D.7, we have for any $k \geq 0$

$$R(\overline{u''}, k) = R(w(r), k) \tag{6.32}$$

Observe that the increasing subsequences in $\overline{u''}$ correspond exactly to the sequences of 1's going North-East in M'', while the decreasing ones correspond to the sequences going south-East. Condition 6.16.1 implies therefore that no two 1's corresponding to boxes of the same column of r in the classical numbering can be part of the same increasing subsequence. Thus the increasing subsequences can only have one 1 per column of r in the classical numbering, which means that they cannot be longer than the subsequences corresponding to levels of r in the classical numbering. More generally, we can conclude that we obtain $R(\overline{u''}, k)$ considering the subsequences of $\overline{u''}$ corresponding to the k top levels of r.

On the other hand, Corollary 6.24 implies that an increasing subsequence in $w(r)$ can only have one number per column of r in the classical numbering. At the same time it follows trivially from the definition of a Young tableau that such a sequence can only have one number per column of r in its actual numbering (the one constructed by Algorithm 6.3).

Let us now substitute $k = \min\{c_i | c_i > 0\}$ in (6.32). In this case k is equal to the number of levels of r of maximum length. Combining the two observations above, we can deduce that

in $w(r)$ there are exactly k subsequences each having one letter per column of r in its classical numbering as well as in its actual numbering. Thus, for any i such that $c_i = k$ we can deduce that all numbers $t_i - c_i + 1, \ldots, t_i$ are in the column i of r in its actual numbering, in other words, column i is numbered correctly.

Repeating the same argument for subsequent levels of r we conclude that all columns of r are numbered correctly, which proves that $M \in \mathfrak{M}^{(N)}$, and thus finalises the proof of Theorem 6.17. ■

6.4 Discussion

In this chapter, we have turned to the very fundamentals of the performance analysis of UMTS in particular and any mobile communications system in general, by placing ourselves in the context of Bit Error Rate (BER) estimation. More precisely, we have considered soft demodulation of a digital signal modulated with Binary Phase Shift Keying (BPSK) technique and in presence of spatial diversity.

This subject has already been studied in [30, 31] and [54], where interesting combinatorial results were obtained, as well as an efficient algorithm for calculating BER based on the characteristics of different propagation paths. In these papers, BER is expressed as a conditional probability that a difference of two quadratic forms U and V is less than 0 under the assumption that the value of the transmitted bit was 0. This is denoted $P(U - V < 0)$, where U and V are two real random variables such that $U = \sum_{i=1}^{N} |u_i|^2$ and $V = \sum_{i=1}^{N} |v_i|^2$ with u_i's and v_i's being independent centred complex Gaussian variables with variances $\mathbf{E}[\,|u_i|^2\,] = \chi_i$ and $\mathbf{E}[\,|v_i|^2\,] = \delta_i$.

The material of this chapter was devoted to generalising the results and algorithms of these studies to a computation of the corresponding conditional probability distribution function $P(U - V < \varepsilon)$, which can also be given an interpretation in terms of mobile communications. Indeed, the difference $U - V$ represents the log-likelihood ratio of the bit in question, and, therefore, the probability $P(U - V < \varepsilon)$ indicates in way "how close we were to making an error".

We gave two expressions in terms of symmetric functions over the alphabets $\Delta = (\delta_1, \ldots, \delta_N)$ and $X = (\chi_1, \ldots, \chi_N)$ for the first $2N-1$ coefficients of the Taylor expansion of $P(U-V < \varepsilon)$ in terms of ε. The first one is a quotient of multi-Schur functions involving two alphabets derived from alphabets Δ and X, which allows us to give an efficient algorithm for the computation of these coefficients.

The second expression involves a certain sum of pairs of Schur functions $s_\lambda(\Delta)$ and $s_\mu(X)$ where λ and μ are complementary shapes inside an $N \times N$ rectangle. We showed that such a sum has a natural combinatorial interpretation in terms of what we call square tabloids with ribbons and that there is a natural extension of the Knuth correspondence that associates a (0,1)-matrix to each square tabloid with ribbon. We then presented a complete characterisation of the (0,1)-matrices that arise from square tabloids with ribbons under this correspondence.

Clearly, the results of this chapter are not directly applicable to practical performance evaluation in communication networks, but present more interest in the domain of combinatorics. They allow us, however, to show how interesting combinatorial objects can be obtained in a rather practical context.

Conclusion

The main goal of this thesis was twofold: firstly, to develop a formal mathematical model for a concept that we call *Complex Industrial Systems* and, secondly, to illustrate this model and the underlying analysis on a real-life industrial system.

The notion of complex industrial system refers principally to products of industrial development, such as, for example, cars and aeroplanes, of which one can say that their design and engineering are complex technical and managerial operations. Moreover, this notion can also imply some more global entities as, for example, an industrial plant, which involves, at the same time, a number of technological systems such as production chains, software systems controlling these chains, and eventually a team of human operators, which can also be considered as a system in this context.

Typically, such systems are a result of integration of a multitude of subsystems or components that, in their, turn can also be complex systems, although with a more precise functionality and easier to conceive. Thus, the underlying development process can be separated in two phases: engineering and integration. The former consists in recursively decomposing the system into smaller subsystems, until each component can be completely specified, and the latter involves assembling these *elementary* components to produce the final complex system. The complexity of the final design and the multitude of the involved subsystems are the two qualities that characterise the resulting system. In particular, a system in the target range of our model can be described by an affirmation that it *cannot be apprehended in all its details by one human being.*

We have introduced, in this thesis, a theoretical model that allows to formally define a system as described above, and reflects its complex nature by integrating the principle of recursive decomposition.

As it has been mentioned above, a complex industrial system consists of a large number of subsystems that are very often heterogeneous in their nature. Indeed, one encounters, in practically all situations involving some kind of control, interactions between *physical* phenomena, which evolve in a continuous manner described most often by differential equations, and *logical* (or, as we call them, *software*) ones that exhibit a discrete behaviour. This situation constitutes the research domain of a field facing a steadily increasing popularity, termed *Hybrid Systems*. However, all models proposed so far do not eliminate this inherent duality but rather concentrate their efforts on developing a good interface.

A complex system in our context can, for example, fabricate a discrete output depending on some input information provided continuously, and similarly for all other combinations of discrete and continuous behaviours. One of the fundamental aspects of our model is, therefore, the notion of time. Our model comprises, at each level of the recursive decomposition, three *time scales*: the input, output, and internal ones, which reflect the corresponding evolutions. These time scales are based on the notion of non-standard real numbers that allows to describe

in the same way both discrete and continuous behaviour. Consequently, our model uniformly integrates both types of systems.

Moreover, we have also shown that a number of classical systems ranging from mechanical (as in the Pendulum example) to purely mathematical (as in the discussion of the dynamical systems as introduced in [32]) can be naturally represented in our model.

$$- \; o \; -$$

The model defined above has served us, in the rest of this thesis, as a guiding thread for a progressive descent through different levels of a particular industrial system, namely the Universal Mobile Telecommunications System — a third generation mobile communications standard.

Thus, in Chapter 3, after a brief overview of the evolution of communication systems and the architecture of UMTS, we have provided two high-level systemic decompositions of the network: one in a simplified case with only one user, and another in a more general case. In particular, the discussion of the latter allowed us to propose a definition of the range of systems that can be naturally treated by our model as *systems admitting a finite description*.

In Chapters 4, 5, and 6, we have pursued the descent started in Chapter 3, by considering three problems encountered on different levels of decomposition: starting from a rather high level modelling a virtual subsystem in Chapter 4, and descending to a comparatively low level in Chapter 6, where a single bit transmission was analysed.

More precisely, in Chapter 4, we have studied the Uplink Power Control. We started by showing how a virtual system can be assembled from separate components of the global system in order to study a particular functionality, here the power control in the ascending link between a mobile and a base station. We have presented several stochastic algorithms for the Outer Loop Power Control, and exhibited thereby a concrete method for determining the appropriate parameters for such algorithms, which is an important problem influencing the network capacity as well as the performance of a single link.

In Chapter 5, we have descended to the next level of our system, by considering the 16QAM modulation used for the High Speed Downlink Packet Access feature. In particular, our goal, in this chapter, was to determine the optimal control scheme for Hybrid ARQ. As opposed to Chapter 4, where the analysis was realised by a rather theoretical approach, that of Chapter 5 was conducted by simulation, thus illustrating the use of the two most prominent techniques in industrial development.

Finally, in Chapter 6, we have descended to practically the lowest possible level in the analysis of a telecommunications network, by placing ourselves in the context of a single bit transmission over a radio channel with spatial diversity, i.e. in presence of multiple propagation paths, to study the problem of estimating the bit error rate. The results of this chapter generalise those obtained in [30, 31, 54], and introduce an interesting class of combinatorial objects.

$$- \; o \; -$$

To conclude, this work allowed us to lift the curtain on what might become one day a uniform theory of complex industrial systems and to gain a certain understanding of its possibilities and limitations. Nevertheless, it leaves also a number of open problems and research directions both in the systemic part and for each individual problem considered in the last chapters. In particular, it would be interesting to consider possible hierarchies of systems, and to build the corresponding complexity and calculability theories based on this model; but also one can continue the exploration of power control algorithms, look for a generalisation and analytical

analysis of H-ARQ control schemes, not to speak of the properties of square tabloids with ribbons, which is the class of combinatorial objects introduced in the last chapter.

Appendix A

Non-standard analysis

In this appendix, we provide the fundamentals of the non-standard analysis that we use in Chapter 2 to define the notion of time scales used in our definition of systems. We start by presenting the Zermelo-Fraenkel axiomatic system of set theory, as well as its main extensions, namely the axiom of choice and the continuum hypothesis. We then proceed by introducing some elements of model theory, and in particular the notions of ultrafilters and ultraproducts, followed by that of elementary equivalence. Finally, making use of these notions we give a formal construction of the field ${}^*\mathbb{R}$ of non-standard reals and show that it is elementary equivalent to the standard real field $\mathbb{R}$.

Except for a couple of minor additions, the body of this appendix is a condensed compilation of [60] — for the set theory presentation (Section A.1) — and [48] and [65] — for what concernes ultrafilters and ${}^*\mathbb{R}$ (Sections A.2 through A.4).

A.1 Elements of set theory

A.1.1 Zermelo-Fraenkel system (ZF) or common mathematics

The ZF system is given by the universal closure of the nine axioms below. This system provides axiomatic foundation for what is commonly considered *usual mathematics*. More precisely, it introduces the notion of set that allows to define constructively most of the objects the usual mathematics manipulate.

The axioms below can be separated into three categories:

- the axioms of *simple existence* that stipulate the existence of sets: the empty set (ZF_3) and the infinity (ZF_7),
- the axioms of *restriction* that limit the spreading of sets by requiring that they possess some reasonable properties: extensionality (ZF_1) and foundation (ZF_9),
- the axioms of *conditional existence* (all other axioms of ZF) that provide ways of "constructing" sets from those already existing.

Let us now list the nine axioms discussed above. We use the symbols like $=$ (identity), $\in$ (membership), $\forall$ (universal quantifier), $\exists$ (existential quantifier), etc. in their usual sense; the letters x, y, z, and t represent variables, i.e. they are always tied up by a corresponding quantifier; and the letters a and b represent constants referring to some given set fixed in the context where they appear.

ZF_1 **the axiom of Extensionality**. Two sets x and y having exactly the same elements are equal:

$$\forall x \forall y \Big(\forall z (z \in x \Leftrightarrow z \in y) \Rightarrow (x = y) \Big) .$$

ZF_2 **the axiom of Subsets (Separation, Comprehension)**. For any property $\varphi(x)$ that does not contain y, and where x is a free variable, and for any set a, there exists a subset y of a, which contains only those elements of the latter that satisfy the property $\varphi(x)$:

$$\exists y \; \forall x (x \in y \Leftrightarrow (x \in a \wedge \varphi(x)) .$$

ZF_3 **the axiom of the Empty Set**. There exists an empty set (i.e. a set without elements):

$$\exists y \; \forall x (x \notin y) .$$

Assuming that there exists at least one set a (which is implied by the existence of an infinite set postulated by ZF_7), this axiom is a consequence of ZF_2 as we have

$$\emptyset = \{x | x \neq x\} = \{x | x \in a \wedge x \neq x\} .$$

ZF_4 **the axiom of the Unordered Pair (Pairing)**. Given two sets a and b, there exists a set which has a and b as its only elements:

$$\exists y \; \forall x (x \in y \Rightarrow x = a \vee x = b) .$$

ZF_5 **the axiom of the Power Set**. Given a set a, there exists a set which has all the subsets of a as its only elements:

$$\exists y \; \forall x (x \in y \Leftrightarrow \forall z (z \in x \Rightarrow z \in a))$$

or in an abbreviated form

$$\exists y \; \forall x (x \in y \Leftrightarrow x \subset a) .$$

ZF_6 **the axiom of the Sum Set (Union)**. For any set a there exists a set $\bigcup a$ (also denoted $\bigcup_{x \in a} x$), which is the union of all elements of a, i.e. it has the elements of all elements of a as its only elements:

$$\exists y \; \forall x (x \in y \Leftrightarrow \exists z (x \in z \wedge z \in a)) .$$

ZF_7 **the axiom of Infinity**. There exists an infinite set. More precisely, there exists a set x, which has $\emptyset$ as its first element (in the order induced by $\in$), such that $y \in x$ implies $y \cup \{y\} \in x$:

$$\exists x (\emptyset \in x \wedge \forall y (y \in x \Rightarrow \exists z (z \in x \wedge \forall t (t \in z \Leftrightarrow t \in y \vee t = y)))) ,$$

which in abbreviated form becomes

$$\exists x (\emptyset \in x \wedge \forall y (y \in x \Rightarrow y \cup \{y\} \in x)) .$$

The smallest set satisfying ZF_7 is denoted by ω or, more commonly, by $\mathbb{N}$.

ZF_8 **the axiom of Replacement**. If the definition domain of a functional relation $\varphi(x, y)$ is a set then its image is also a set. More precisely, for any property $\varphi(x, y)$ that does not contain a, and where x and y are free variables, we have

$$\forall x \forall y \Big(\forall z (\varphi(x, y) \wedge \varphi(x, z) \Rightarrow y = z) \Rightarrow \exists t \; \forall y (y \in t \Leftrightarrow \exists x (x \in a \wedge \varphi(x, y))) \Big) .$$

ZF_9 **the axiom of Foundation (Regularity)**. Any non-empty set x has an element y, which does not have any common elements with x, i.e. a minimal element in the order induced by $\in$ (that is the relation $\in$ is well founded):

$$\forall x \Big(\exists y (y \in x) \Rightarrow \exists y (y \in x \wedge \forall z \, \neg (z \in x \wedge z \in y)) \Big)$$

or, in abbreviated form,

$$\forall x \Big((x \neq \emptyset) \Rightarrow \exists y (y \in x \wedge y \cap x = \emptyset) \Big) .$$

A.1.2 Stronger theories

Even though Zermelo-Fraenkel set theory provides to an important extent the foundation of the common mathematics, other assertions can be added to enrich it. Some of these assertions are independent from the ZF system. Their admission or exclusion is made depending on what is the use one intends to make of the set theory. Postulating a given axiom can, for instance, solve some important mathematical problems. However, some assertions are incompatible with other ones, and consequently different ways of enriching ZF exist, sometimes also mutually incompatible. In this section, we shall briefly present the two most prominent of the different extensions of ZF existing at present.

Axiom of choice (AC)

The axiom of choice is a typical example of an assertion extending ZF, which has profoundly divided the mathematical community in the first half of the 20th century, but which also became a source of vigorous research aiming to clarify its nature.

In particular, in 1938, Gödel has shown that adding the axiom of choice to other axioms of set theory can not render the enriched theory contradictory, provided that this is not the case with the original one. In other words, the axiom of choice cannot be refuted in the framework of the fondamental set theory.

In return, in 1963, Cohen proved that the negation of the axiom of choice is also compatible with other axioms of set theory, thus showing that the axiom of choice can neither be derived in this framework.

The assertion below is the form of the axiom of choice as it has been included, in 1908, by Zermelo into his axiomatic system. (It was also Zermelo, who has coined the term "axiom of choice" in the same year.)

> *For any set a, such that all of its elements are non-empty and pairwise disjoint, there exists a set c (called the* choice set *of a), such that its intersection with each element of a is a set consisting of exactly one element:*

$$\forall x \Big(x \in a \Rightarrow x \neq \emptyset \wedge \forall y (y \in a \Rightarrow x \cap y = \emptyset \vee x = y) \Big) \Rightarrow \exists c \; \forall x \exists t (x \in a \Rightarrow c \cap x = \{t\}) .$$

The axiom of choice has two important characteristics:

- its *fecondity*: numerous mathematical assertions are consequences of this axiom, and, moreover, cannot be proven without some form of "choice";
- and its *stability*: an important number of the consequences of the axiom of choice, coming from various branches of mathematics, imply it in their turn, and thus represent its *equivalent forms*.

One of the most well-known equivalent forms of the axiom of choice is the Zorn's lemma.

Lemma A.1 (Zorn, 1935) *If $(x, \leq)$ is a non-empty partially ordered set, such that any chain (totally ordered subset) in x admits an upper bound, then x has a maximal element.*

Continuum hypothesis

Another important assertion that is often considered as an addition to the fondamental set theory is the so-called *continuum hypothesis*.

For a given set a, let us denote by $P(a)$ the set of all of the subsets of a (cf. ZF_5). Any set a is then equipotent to some subset of $P(a)$ but not to $P(a)$ itself, the cardinality of the latter is strictly superior to that of a (Cantor's theorem). The question arises naturally: whether the are cardinal numbers between those of a and $P(a)$. The answer is obviously positive in case when a is finite. However, in the case when a is infinite, the question is undecidable in ZF. Equivalently, the axiom of choice implies that, for any ordinal number α, we have $\aleph_{\alpha+1} \leq 2^{\aleph_\alpha}$, but $\aleph_{\alpha+1} = 2^{\aleph_\alpha}$ is undecidable in ZFC (ZF+AC). The following assertions postulating the corresponding equalities are called respectively *continuum hypothesis* and *aleph hypothesis*:

Hypothesis A.2 (Continuum hypothesis [CH]) *There is no cardinal comprised strictly between the first infinite cardinal $\aleph_0$ and $2^{\aleph_0}$.*

This can be summarised by the affirmation that the power of continuum $2^{\aleph_0}$, i.e. the cardinality of the set $\mathbb{R}$ of real numbers, is equal to the first non-denumerable cardinal $\aleph_1$.

Hypothesis A.3 (Aleph hypothesis [AH]) *For any ordinal α, we have $2^{\aleph_\alpha} = \aleph_{\alpha+1}$.*

Observe that the aleph hypothesis is a generalisation of the continuum hypothesis. In ZF it is equivalent to the *generalised continuum hypothesis* below.

Hypothesis A.4 (Generalised continuum hypothesis [GCH]) *For any transfinite cardinal κ, there is no cardinal comprised strictly beteen κ and 2^κ.*

Gödel has shown that GCH cannot be refuted in ZFC, whereas Cohen has shown subsequently that CH (and *a fortiori* GCH) cannot be proven in ZFC (both under the assumption that ZFC is consistent).

A.2 Ultrafilters and ultraproducts

A.2.1 Ultrafilters and measures

Definition A.5 *If I is an arbitrary set, a* filter *$\mathcal{F}$ over I is a family of subsets of I satisfying the following conditions:*

1. *the empty set does not belong to* $\mathcal{F}$,
2. *for any two* $P, Q \in \mathcal{F}$ *we also have* $P \cap Q \in \mathcal{F}$,
3. *for any* $P \in \mathcal{F}$ *and any* $Q \subset I$ *such that* $P \subset Q$ *we also have* $Q \in \mathcal{F}$.

Observe that conditions 1 and 2 imply that a filter cannot contain both a given set $P \subset I$ *and its complementary* $(I \setminus P)$. *A filter that, for any* $P \subset I$ *contains at least one of these is called an* ultrafilter.

Example A.6
A family $\mathcal{F}_a = \{P \subset I \,|\, a \in P\}$ of all subsets of I containing a given element $a \in I$ is called a *principal* (or *trivial*) ultrafilter generated by a. ■

Example A.7
If J is infinite the family of all $P \subset I$, such that $I \setminus P$ is finite, forms a filter. Filters of this kind which don't contain any finite sets are called *free*. ■

The filters over a given set I are partially ordered by set inclusion. The following proposition stipulates that ultrafilters are exactly filters that are maximal with respect to this ordering.

Proposition A.8 *A filter* $\mathcal{F}$ *over* I *is an ultrafilter iff it is maximal with respect to set inclusion.*

Proof. The fact that a filter that, for any given $P \subset I$, contains either P or $I \setminus P$ is maximal is trivial. Hence we only have to show that any maximal filter satisfies this property.

Suppose that $\mathcal{F}$ is a maximal filter, and let $P \subset I$ be a subset of I such that neither $P \in \mathcal{F}$ nor $(I \setminus P) \in \mathcal{F}$.

Observe that there cannot exist two sets $Q_1, Q_2 \in \mathcal{F}$ satisfying $Q_1 \cap P = Q_2 \cap (I \setminus P) = \emptyset$ since this would imply that $\emptyset = Q_1 \cap Q_2 \in \mathcal{F}$. Therefore we can assume without loss of generality that, for all $Q \in \mathcal{F}$, we have $Q \cap P \neq \emptyset$. It is easy to check then that

$$\mathcal{F}' = \left\{ Q \subset I \,\middle|\, \exists F \in \mathcal{F}\,(F \cap P \subset Q) \right\}$$

is a filter and, moreover, a proper extension of $\mathcal{F}$, which contradicts the latter's maximality. ■

Note A.9 Corollaries A.10 and A.11 below assume Zorn's lemma.

Corollary A.10 *Any filter* $\mathcal{F}$ *over some set* I *can then be extended to an ultrafilter over* I.

Proof. Let $\mathcal{S}$ be the set of all filters over I extending $\mathcal{F}$, and order $\mathcal{S}$ by inclusion. By Zorn's lemma, $\mathcal{S}$ has a maximal element $\mathcal{U}$. Proposition A.8 implies that $\mathcal{U}$ is necessarily an ultrafilter. ■

Applying the above corollary to the free filter defined in Example A.7 we obtain the existence of free ultrafilters under the hypothesis of the Zorn's lemma validity.

Corollary A.11 (Weak ultrafilter theorem) *Let* I *be an infinite set. There exists a free ultrafilter over* I.

The proof of the existence of non-principal ultrafilters that we have provided above is based on the Zorn's lemma, and thus depends on a form of the axiom of choice. It is known, however, that it is strictly weaker than AC. Moreover, assuming the consistence of ZF system, the dependencies in Figure A.1 can be shown. The statements in Figure A.1, which were not yet mentioned, are listed below.

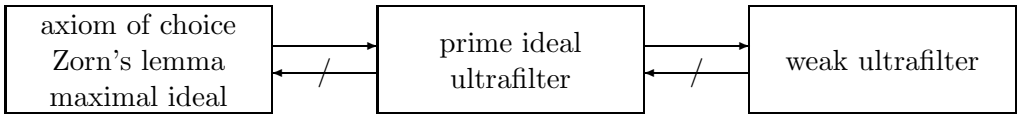


Figure A.1: Dependencies between some statements in set theory (an arrow indicates that one notion implies another, a crossed arrow — that it does not; statements in the same box are equivalent).

Theorem A.12 (Maximal ideal theorem [Krull, 1929 [1]]) *Every commutative ring, such that $1 \neq 0$, contains a maximal ideal.*

Theorem A.13 (Prime ideal theorem [Stone, 1936]) *Every Boolean algebra contains a prime ideal.*

Theorem A.14 (Ultrafilter theorem [Cartan, 1937 [2]]) *Every Boolean algebra contains an ultrafilter.*

$$-\ o\ -$$

Observe that, if $\mathcal{F}$ is a free ultrafilter on I, the function $m : 2^I \to \{0, 1\}$ defined by setting

$$m(P) = \begin{cases} 1 & \text{if } P \in \mathcal{F} \\ 0 & \text{if } P \notin \mathcal{F}\,, \end{cases}$$

is a finitely additive measure defined on all subsets of I, such that $m(I) = 1$ and $m(P) = 0$ for all finite subsets $P \subset I$. Conversely any such measure m defines a free ultrafilter $\mathcal{F}$ on I by setting

$$\mathcal{F} = \{P \subset I \,|\, m(P) = 1\}\,.$$

Thus a free ultrafilter can be interpreted as a collection of subsets of I, such that each of the sets belonging to it contains "almost all" elements of I. Observe that this notion of "almost all" does not necessarily correspond to the intuitive one. Indeed, if one was to consider, for example, a free ultrafilter $\mathcal{U}$ over $\mathbb{N}$, it is impossible to say *a priori* whether even or odd, prime or composite numebers constitute "almost all" natural numbers, even though one and only one of each of these pairs has to member of $\mathcal{U}$. However, this ambiguity does not have any real effect on the construction of the set of the non-standard reals, which is the main goal of this appendix (see Section A.3).

A.2.2 Ultraproducts and elementary equivalence

Let $(R_\alpha)_{\alpha\in I}$ be a family of rings,[3] and let $\mathcal{F}$ be an ultrafilter over I. In the direct (Cartesian) product $\prod_{\alpha\in I} R_\alpha$ we introduce an equivalence relation by setting $(r_\alpha) \sim (r'_\alpha)$ if and only if the set $\{\alpha \in I \,|\, r_\alpha = r'_\alpha\}$ belongs to $\mathcal{F}$, which is also expressed by saying that $r_\alpha = r'_\alpha$ for $\mathcal{F}$-*almost all* α.

[1] The equivalence of the maximal ideal theorem and the axiom of choice was shown by Scott in 1954.

[2] In 1937, Cartan was unaware of the duality between a filter and an ideal in a Boolean algebra.

[3] More generally, we can consider a family of any type of algebraic structures such as, for example, groups or modules. Here, we choose to speak of rings purely for the convenience of presentation.

The equivalence class represented by an element (r_α) is denoted by $[r_\alpha]$. By obvious componentwise addition and multiplication these equivalence classes form a ring, called the *ultraproduct of $(R_\alpha)_{\alpha\in I}$ with respect to $\mathcal{F}$*, that is denoted $\prod_{\alpha\in I} R_\alpha/\mathcal{F}$. If $R_\alpha = R$ for all $\alpha \in I$, the ultraproduct is denoted by $R^I/\mathcal{F}$ and is called the *ultrapower of R with respect to $\mathcal{F}$*. In the latter case there is a canonical mapping $\Delta : R \to R^I/\mathcal{F}$ defined by setting $\Delta(r) = [r_\alpha]$, where $r_\alpha = r$ for all $\alpha \in I$.

It can be easily verified that $\prod_{\alpha\in I} R_\alpha/\mathcal{F}$ is a field if each R_α is a field. It even suffices to assume that R_α is a field for all α's in some subset $J \subset I$ belonging to $\mathcal{F}$. Conversely, if the ultraproduct $\prod_{\alpha\in I} R_\alpha/\mathcal{F}$ is a field, the set $\{\alpha \in I \,|\, R_\alpha \text{ is a field}\}$ belongs to $\mathcal{F}$. Thus, we have the following property of an ultraproduct.

Proposition A.15 *The ultraproduct $\prod_{\alpha\in I} R_\alpha/\mathcal{F}$ is a field if and only if R_α is a field for $\mathcal{F}$-almost all $\alpha \in I$.*

The property *R is a field* can be expressed by the formula $\forall x(x = 0 \vee \exists y\,(xy = 1))$. This is an example of a *first order sentence in the language $\mathcal{R}$ of rings*, that is, a formula in the language of rings, in which every variable is in the scope of a quantifier ($\forall$ or $\exists$).

The above example is a special case of a metatheorem called Łos's principle.

Theorem A.16 (Łos's principle) *Let $(R_\alpha)_{\alpha\in I}$ be a family of rings (resp. fields, modules, etc.), and $\mathcal{F}$ an ultrafilter over I. A first order sentence σ in the language of rings (resp. fields, modules, etc.) holds for the ultraproduct $\prod_{\alpha\in I} R_\alpha/\mathcal{F}$ if and only if σ holds in R_α for $\mathcal{F}$-almost all α in I.*

Definition A.17 *Two algebraic structures R and S are called* elementary equivalent *(denoted $R \equiv S$) if R and S satisfy the same first order sentences in the corresponding language.*

Corollary A.18 *Let R be an algebraic structure. The following equivalence then holds for any index set I and any ultrafilter $\mathcal{F}$ over I*

$$R^I/\mathcal{F} \equiv R\,.$$

A.3 The set ${}^*\mathbb{R}$ of non-standard reals

A.3.1 Construction of non-standard reals

Let us now consider the sequences of real numbers indexed by $\mathbb{N}$. Postulating the weak ultrafilter theorem,[4] there exists a free ultrafilter $\mathcal{F}$ over $\mathbb{N}$. The set of non-standard reals is then simply the ultrapower $\mathbb{R}^{\mathbb{N}}/\mathcal{F}$. Let us however elaborate some more on this construction. The following proposition is derived trivially from Definition A.5.

Proposition A.19 *Let $\mathcal{F}$ be an non-trivial ultrafilter over $\mathbb{N}$. A relation $\sim_{\mathcal{F}}$, defined by setting*

$$(x_n) \sim_{\mathcal{F}} (y_n) \overset{def}{\iff} \{n \in \mathbb{N} \,|\, x_n = y_n\} \in \mathcal{F}\,,$$

is then an equivalence relation on the set $\{(x_n)_{n=1}^{\infty} \,|\, \forall n \in \mathbb{N}\,(x_n \in \mathbb{R})\}$ of sequences of (standard) real numbers.

[4] I.e. the weakest of the assertions in Figure A.1.

We are now in position to define the set of *non-standard reals* or *hyperreals* by factoring the set of (standard) real sequences by the relation $\sim_{\mathcal{F}}$:

$$^*\mathbb{R} \stackrel{def}{=} \left\{ (x_n)_{n=1}^{\infty} \,\middle|\, \forall n \in \mathbb{N}\,(x_n \in \mathbb{R}) \right\} \Big/ \sim_{\mathcal{F}} .$$

Denoting by $[x_n]$ the equivalence class of the sequence (x_n), we can define addition and multiplication in $^*\mathbb{R}$ by setting

$$[x_n] + [y_n] = [x_n + y_n] \; ; \quad [x_n] \cdot [y_n] = [x_n \cdot y_n]$$

and order it accordingly by

$$[x_n] < [y_n] \stackrel{def}{\Longleftrightarrow} \{n \in \mathbb{N} \,|\, x_n < y_n\} \in \mathcal{F} . \tag{A.1}$$

It is easy to verify that these definitions are independent of the representatives (x_n) and (y_n) of the equivalence classes $[x_n]$ and $[y_n]$ correspondingly.

The standard reals can be embedded in $^*\mathbb{R}$ by identifying each $x \in \mathbb{R}$ with the equivalence class $[x]$ of the corresponding constant sequence.

Observe that this definition of non-standard real numbers bears a significant resemblance to the construction of standard reals from the rationals using Cauchy sequences. Indeed, if we denote by $\mathcal{C}$ the set of such sequences, and by $\sim$ the equivalence relation defined by

$$(p_n) \sim (q_n) \stackrel{def}{\Longleftrightarrow} \lim_{n\to\infty} (p_n - q_n) = 0 \,,$$

then the standard reals are just the set $\mathbb{R} = \mathcal{C}/\!\sim$ with the operations defined in the same manner as in (A.1).

When we construct, in this manner, the reals from the rationals, we are interested in constructing limit points for all "naturally" convergent sequences. Since the limit is all we care about, it is convenient to identify as *many* sequences as possible; i.e. all those that converge to the same "point". No attention is paid to the rate of convergence; hence the two sequences $(1/n)$ and $(1/\sqrt{n})$ are identified with the same number 0 although they converge at quite different rates. In creating $^*\mathbb{R}$ from $\mathbb{R}$, we want to construct a rich and well-organised algebraic structure that encodes not only the *limit* of a sequence but also its *mode of convergence*. To achieve this, we reverse the strategy and identify as *few* sequences as possible. The construction presented above allows to achive this goal preserving, however, some nice algebraic properties of $\mathbb{R}$.

Example A.20
Consider two sequences $(p_n) = (1, 0, 1, 0, 1, \ldots)$ and $(q_n) = (0, 1, 0, 1, 0, \ldots)$. Thus $(p_n)\cdot(q_n) = 0$, and if none of these two sequences is identified with zero, we obtain a structure with zero divisors.

To see that this problem does not appear in our construction, assume that $[p_n] \cdot [q_n] = [0, 0, \ldots]$, i.e. $\{n \,|\, p_n \cdot q_n = 0\} \in \mathcal{F}$. This implies immediately that

$$\{n \,|\, p_n \neq 0\} \cap \{n \,|\, q_n \neq 0\} = \{n \,|\, p_n \cdot q_n \neq 0\} \notin \mathcal{F} , \tag{A.2}$$

and, consequently (by the property 2 in the Definition A.5 of filters), at least one of the sets in the left-hand side of (A.2) does not belong to $\mathcal{F}$. Therefore either $\{n \,|\, p_n = 0\} \in \mathcal{F}$ or $\{n \,|\, q_n = 0\} \in \mathcal{F}$, which translates to the statement that either $[p_n] = [0, 0, \ldots]$ or $[q_n] = [0, 0, \ldots]$. ∎

Although the constructions of $\mathbb{R}$ and ${}^*\mathbb{R}$ are very similar, there is an important difference between the two sets: the dependence on the ultrafilter $\mathcal{F}$ makes ${}^*\mathbb{R}$ "less canonical" than $\mathbb{R}$. Indeed, looking back at Example A.20, one observes that in ${}^*\mathbb{R}$ one of the two sequences $(1,0,1,0,1,\ldots)$ and $(0,1,0,1,0,\ldots)$ is identified with 0 and the other one with 1; and which is which depends on the ultrafilter $\mathcal{F}$. If we stick to the philosophy above and consider $\mathbb{R}$ and ${}^*\mathbb{R}$ as structures constructed to reflect the asymptotic behaviour of sequences, this is not too disconcerting: the difference between the two sets is just that in creating $\mathbb{R}$ from the rational Cauchy sequences we throw out the sequences that do not have a decent asymptotic behaviour at the very beginning, whereas in creating ${}^*\mathbb{R}$ we keep them and treat them in an arbitrary but coherent way instead.

Mathematically, this point of view is supported by the fact that hyperreals arising from different non-principal ultrafilters have the same analytic properties, **even though they can only be shown to be algebraically isomorphic under extra set-theoretic assumptions such as the continuum hypothesis**.

Note A.21 Observe that for any $x \in \mathbb{N}$ the factor space $\{(x_n)_{n=1}^{\infty} \,|\, \forall n \in \mathbb{N},\ x_n \in \mathbb{R}\}/{\sim_{\mathcal{F}_x}}$, where $\mathcal{F}_x$ is the trivial filter introduced in the Example A.6, is algebraically isomorphic to $\mathbb{R}$.

A.3.2 Some properties of ${}^*\mathbb{R}$

For any non-standard real $x \in {}^*\mathbb{R}$, such that $x = [x_n]$, we denote by $|x|$ the absolute value of x defined by $|x| = [|x_n|] \in {}^*\mathbb{R}$. Thus, we are in position to give the following definition.

Definition A.22 *All elements of ${}^*\mathbb{R}$ are classified in three groups:*

- *An element $x \in {}^*\mathbb{R}$ is* infinitesimal *if $|x| < a$ for all positive real numbers a (denoted $x \approx 0$).*
- *An element $x \in {}^*\mathbb{R}$ is* finite *if $|x| < a$ for some positive real number a.*
- *An element in ${}^*\mathbb{R}$ that is not finite is* infinite.

Three examples of infinitesimals are 0, $\delta_1 = [1/n]$, and $\delta_2 = [1/\sqrt{n}]$. To check that, say, δ_1 is infinitesimal, note that for any positive $a \in \mathbb{R}$, the set $\{n \,|\, -a < 1/n < a\}$ contains all but a finite number of n's and hence belongs to the free ultrafilter $\mathcal{F}$ used in the construction of ${}^*\mathbb{R}$. Observe also that since $\delta_1 \neq \delta_2$, the two sequences $[1/n]$ and $[1/\sqrt{n}]$ converging to zero at different rates are represented by different infinitesimals. Finally, note that zero is the only infinitesimal real number. Examples of infinite numbers are $[n]$ (positive) and $[-n^2]$ (negative).

It is easy to check that the intuitive arithmetic rules do hold in ${}^*\mathbb{R}$. For instance, the sum of two infinitesimals is infinitesimal, and so is the product of a finite number and an infinitesimal one.

Proposition A.23 *Any finite $x \in {}^*\mathbb{R}$ can be written uniquely as a sum $x = a + \varepsilon$, where $a \in \mathbb{R}$ and $\varepsilon \approx 0$.*

Proof. The uniqueness is obvious since, if $x = a_1 + \varepsilon_1 = a_2 + \varepsilon_2$, then $a_1 - a_2 = \varepsilon_2 - \varepsilon_1$, which is both real and infinitesimal, and therefore zero.

To prove the existence of such decomposition, let $a = \sup\{b \in \mathbb{R} \,|\, b < x\}$. Since x is finite a exists, and we have to show that $x - a \approx 0$. Suppose that this is not so. Then there exists $r \in \mathbb{R}$ such that $0 < r < |x - a|$, and therefore either we have $a + r < x$ (if $x - a > 0$) or $x < a - r$ (if $x - a < 0$), both contradicting the choice of a. ∎

We shall write $x \approx y$ for *x and y are infinitely close*, i.e. $x - y \approx 0$. Moreover, this proposition allows us to define the *standard part* of finite non-standard reals.

Definition A.24 *For each finite $x \in {}^*\mathbb{R}$, the unique real number a such that $x \approx a$ is called the* standard part *of x and denoted $st(x)$. Conversely, for each real $a \in \mathbb{R}$ the set $\{x \in {}^*\mathbb{R} \,|\, st(x) = a\}$ is called the* monad *of a.*

Finally, observe that ${}^*\mathbb{R}$ also contains the non-standard versions of natural ${}^*\mathbb{N} = \mathbb{N}^{\mathbb{N}}/\mathcal{F}$, integer ${}^*\mathbb{Z} = \mathbb{Z}^{\mathbb{N}}/\mathcal{F}$, and rational ${}^*\mathbb{Q} = \mathbb{Q}^{\mathbb{N}}/\mathcal{F}$ numbers. These sets extend their standard counterparts in the same sense as ${}^*\mathbb{R}$ extends $\mathbb{R}$, and similar elementary equivalences hold. It is important to observe, however, that neither of ${}^*\mathbb{N}$, ${}^*\mathbb{Z}$, and ${}^*\mathbb{Q}$ is countable, i.e. the cardinality of all three of these sets is continuum.

$$- \; o \; -$$

Applying Corollary A.18 to the pair $\mathbb{R}$ and ${}^*\mathbb{R}$, we obtain what is called the *transfer principle.*

Proposition A.25 (Transfer Principle) *A first order sentence in the language of fields is true in ${}^*\mathbb{R}$ if and only if it is true in $\mathbb{R}$.*

The transfer priciple allows us, in particular, to use the Archimedean property in connection with ${}^*\mathbb{R}$ and ${}^*\mathbb{N}$.

Proposition A.26 (Archimedean property) *For each $x \in {}^*\mathbb{R}$ there exists a non-standard natural $N \in {}^*\mathbb{N}$, such that $N < x \leq N + 1$.*

Corollary A.27 *Consider an infinitesimal $\tau \in {}^*\mathbb{R}$. There exists, for any standard real $x \in \mathbb{R}$, a non-standard natural $N \in {}^*\mathbb{N}$ such that $N\tau \approx x$.*

Proof. The non-standard version of Archimedean property implies that there exists a non-standard integer $N \in {}^*\mathbb{N}$ (here necessarily infinitely great) such that $N < x/\tau \leq N + 1$. Hence, we have $N\tau < x \leq (N + 1)\tau$ and consequently $0 < x - N\tau \leq \tau \approx 0$. ∎

A.3.3 Internal sets and functions

One of the first things one does having introduced a new mathematical structure is to look for the classes of "nice" subsets and functions (such as open sets and continuous functions in topology or measurable sets and functions in measure theory). In non-standard analysis the "nice" sets and functions are called internal, and they arise in the following way.

Definition A.28 *A sequence $\{A_n\}$ of subsets of $\mathbb{R}$ defines a subset $[A_n]$ of ${}^*\mathbb{R}$ such that*

$$[x_n] \in [A_n] \text{ iff } \{n \,|\, x_n \in A_n\} \in \mathcal{F}\,.$$

A subset of ${}^*\mathbb{R}$ which can be obtained in this way is called an internal set.

Internal functions are defined in similar manner.

Definition A.29 *A sequence $\{f_n\}$ of functions $f_n : \mathbb{R} \to \mathbb{R}$ defines a function $[f_n] : {}^*\mathbb{R} \to {}^*\mathbb{R}$ by setting*

$$[f_n]([x_n]) = [f_n(x_n)]\,.$$

A function on ${}^*\mathbb{R}$ which can be obtained in this way is called an internal function.

Example A.30

1. If $a = [a_n]$ and $b = [b_n]$ are two elements of ${}^*\mathbb{R}$, then the interval
$$[a, b] = \{x \in {}^*\mathbb{R} \,|\, a \leq x \leq b\}$$
is internal as it is obtained as $\Big[[a_n, b_n]\Big]$.
2. Consider $c = [c_n] \in {}^*\mathbb{R}$. The function $\sin(cx)$ is an internal function defined by
$$\sin(cx) = [\sin(c_n x_n)]\,.$$

■

Note, for instance, that two internal sets $[A_n]$ and $[B_n]$ are equal if and only if $A_n = B_n$ for $\mathcal{F}$-almost all n (and similarly for functions).

The importance of the internal sets and functions consists in their product-like structure, which allows to lift operations and properties componentwise from $\mathbb{R}$ to ${}^*\mathbb{R}$. As an example, we can define the non-standard integral $\int_A f dx$ (where $A = [A_n]$ is an internal set and $f = [f_n]$ an internal function) by setting
$$\int_A f dx = \Big[\int_{A_n} f_n dx\Big]\,.$$
This new integral inherits most of the properties of the standard one. It is easy to check that both general statements such as
$$\int_A (f + g)dx = \int_A f dx + \int_A g dx$$
and more specific ones such as
$$\int_a^b \sin(cx)dx = \frac{1}{c}\cos(ca) - \frac{1}{c}\cos(cb)$$
for all $a, b, c \in {}^*\mathbb{R}$ (with $c \neq 0$) remain true.

The simplest way to obtain internal objects are the so-called *non-standard versions* of standard sets and functions.

Definition A.31 *For each $A \subset \mathbb{R}$, the internal set ${}^*A = [A, A, A, \ldots]$ is called the* non-standard version of A. *In the same way, for any function $f : \mathbb{R} \to \mathbb{R}$, the internal function ${}^*f = [f, f, f, \ldots]$ is called the* non-standard version of f. *An internal set or function is called* standard *if it is of the form ${}^*A$ or ${}^*f$ correspondingly.*

Note that, similarly to $\mathbb{R}$ and ${}^*\mathbb{R}$, the set ${}^*A$ is much richer than A; e.g. the non-standard interval ${}^*(a, b)$ contains not only all real numbers between a and b, but also all non-standard ones with the same property. More generally speaking, holds the following proposition.

Proposition A.32 *For any $A \subset \mathbb{R}$, we have $A \subseteq {}^*A$, with equality if and only if A is finite.*

An interesting example of internal sets are the *hyperfinite* sets; they are infinite sets with all the combinatorial structure of finite ones.

Definition A.33 *An internal set $A = [A_n]$ is called* hyperfinite *if $\mathcal{F}$-almost all the A_n's are finite. The* internal cardinality *of A is the non-standard integer $|A| = [|A_n|]$, where $|A_n|$ is the number of elements in A_n.*

Example A.34
Consider an infinite number $N \in {}^*\mathbb{N}$ and a set

$$T = \left\{0, \frac{1}{N}, \frac{2}{N}, \ldots, \frac{N-1}{N}, 1\right\} ,$$

Observing that if $N = [N_n]$ then $T = [T_n]$ with

$$T_n = \left\{0, \frac{1}{N_n}, \frac{2}{N_n}, \ldots, \frac{N_n - 1}{N_n}, 1\right\} .$$

Hence $|T| = [|T_n|] = [N_n + 1] = N + 1$. ∎

Consider an internal function $f = [f_n]$ and a hyperfinite set $A = [A_n]$. We can then define the *sum* of f over A by

$$\sum_{a \in A} f(a) = \left[\sum_{a \in A_n} f_n(a)\right] .$$

If T is as in the example above and $g : \mathbb{R} \to \mathbb{R}$ is a function, we obtain

$$\sum_{t \in T} {}^*g(t)\frac{1}{N} = \left[\sum_{t \in T_n} g(t)\frac{1}{N_n}\right] .$$

If g is continuous, the sequence on the right converges to $\int_0^1 g(t)dt$, and thus

$$\int_0^1 g(t)dt = st\left(\sum_{t \in T} {}^*g(t)\frac{1}{N}\right) .$$

A.4 Some applications of NSA

A.4.1 Continuity and differentiability of standard functions

Let us now present two basic properties of functions translated in the non-standard language. By doing this we pursue a double goal: firstly, these two examples illustrate the simplicity that some classical notions acquire when exposed in these terms, and, secondly, we will implicitly refer to them in the concluding sections of this appendix.

Proposition A.35 *A function $f : \mathbb{R} \to \mathbb{R}$ is continuous in the point $a \in \mathbb{R}$ if and only if ${}^*f(x) \approx f(a)$ for all $x \approx a$.*

Proof. Assume that f is continuous in a and that $x = [x_n]$ is infinitely close to a. Given $\varepsilon \in \mathbb{R}_+^*$ we must show that $|{}^*f(x) - f(a)| < \varepsilon$. Choose $\delta \in \mathbb{R}_+^*$ such that for all $y \in \mathbb{R}$, $|y - a| < \delta$ implies $|f(y) - f(a)| < \varepsilon$. We then have

$$\left\{n \,\middle|\, |f(x_n) - f(a)| < \varepsilon\right\} \supset \left\{n \,\middle|\, |x_n - a| < \delta\right\}$$

and, since $x \approx a$, the set on the right belongs to the ultrafilter used in the construction of ${}^*\mathbb{R}$. Consequently that on the left also belongs there, which means by definition of order on ${}^*\mathbb{R}$ (cf. (A.1)) that $|{}^*f(x) - f(a)| < \varepsilon$.

If f is not continuous in a, there exist an $\varepsilon \in \mathbb{R}_+^*$ and a sequence $\{x_n\}$ of reals converging to a such that $|f(x_n) - f(a)| > \varepsilon$ for all n. But then $x = [x_n]$ is infinitely close to a and $|{}^*f(x) - f(a)| > \varepsilon$. ∎

Proposition A.36 *A function $f : \mathbb{R} \to \mathbb{R}$ is differentiable in $a \in \mathbb{R}$ if and only if there exists a number $b \in \mathbb{R}$ such that*

$$\frac{{}^*f(x) - {}^*f(a)}{x - a} \approx b$$

for all x satisfying both $x \approx a$ and $x \neq a$. Moreover, if such b exists, it equals $f'(a)$.

The proof of this second proposition is almost identical to the proof of Proposition A.35 above.

Corollary A.37 *If g is differentiable in a and f in $g(a)$, then $f \circ g$ is differentiable in a, and $(f \circ g)'(a) = f'(g(a))g'(a)$.*

Proof. Let $x \approx a$. All we have to prove is that

$$\frac{{}^*f\Big({}^*g(x)\Big) - {}^*f\Big({}^*g(a)\Big)}{x - a} \approx f'\Big(g(a)\Big)g'(a)\,.$$

But if ${}^*g(x) = {}^*g(a)$, then both sides of this equation are zero, since $g'(a) = \Big({}^*g(x) - {}^*g(a)\Big)/(x - a)$, whereas if ${}^*g(x) \neq {}^*g(a)$, we can write

$$\frac{{}^*f\Big({}^*g(x)\Big) - {}^*f\Big({}^*g(a)\Big)}{x - a} = \frac{{}^*f\Big({}^*g(x)\Big) - {}^*f\Big({}^*g(a)\Big)}{{}^*g(x) - {}^*g(a)} \cdot \frac{{}^*g(x) - {}^*g(a)}{x - a} \approx f'\Big(g(a)\Big)g'(a)$$

by Proposition A.36. ∎

A.4.2 Differential equations

Theorem A.38 (Peano) *Let $f : \mathbb{R} \times [0, 1] \to \mathbb{R}$ be a bounded continuous function. Then the initial value problem*

$$\begin{cases} y'(t) &= f(y(t), t) \\ y(0) &= y_0 \end{cases}$$

has a solution for all $y_0 \in \mathbb{R}$.

Proof. Let the set T be defined as in Example A.34, and consider a function $Y : T \to {}^*\mathbb{R}$ such that $Y = [Y_n]$ with functions $Y_n : \mathbb{R} \to \mathbb{R}$ defined inductively by setting

$$Y_n\Big(k/N_n\Big) = y_0 + \sum_{i=0}^{k-1} f\Big(Y_n(i/N_n), i/N_n\Big)\frac{1}{N_n}\,.$$

Similar equality then holds for Y:

$$Y\Big(k/N\Big) = y_0 + \sum_{i=0}^{k-1} {}^*f\Big(Y(i/N), i/N\Big)\frac{1}{N}\,. \tag{A.3}$$

Next, observe that since f is bounded by some real number M, for all $s, t \in T$ we have $|Y(t) - Y(s)| \leq M|t - s|$. Consequently, Y is continuous in the sense that $Y(s) \approx Y(t)$ whenever $s \approx t$, and a function $y : [0, 1] \to \mathbb{R}$, defined by setting $y(t) = st(Y(\tilde{t}))$ with $\tilde{t}$ being the element of T to the immediate left of t, is so in the sense of Proposition A.35.

An argument similar to that after the Example A.34 in the end of Section A.3.3 allows us to conclude that

$$\int_0^t f(y(s),s)ds \approx \sum_{i=0}^{\tilde{t}N} {}^*f\Big({}^*y(i/N), i/N\Big)\frac{1}{N}\,. \tag{A.4}$$

Finally, observe also that f, y, and Y all being continuous entails

$${}^*f\Big({}^*y(i/N), i/N\Big) \approx {}^*f\Big(Y(i/N), i/N\Big)\,, \tag{A.5}$$

and therefore, combining (A.3), (A.4), and (A.5) we obtain

$$y(t) = y_0 + \int_0^t f\Big(y(s),s\Big)ds\,,$$

which proves the theorem. ■

The proof of this theorem provides a technique of solving differential equations by reducing them to hyperfinite difference equations, which happens to be quite powerfull. In the following section we apply it to a much more complicated theory of stochastic differential equations, but before doing this let us give the following simple example.

Example A.39
Consider the following differential equation on $\mathbb{R}$

$$\left\{\begin{array}{rcl} y'(t) &=& y(t)+2 \\ y(0) &=& a\,. \end{array}\right.$$

To solve it, let us take an infinite number $N \in {}^*\mathbb{N}$. For any given $x \in \mathbb{R}$, we then have the following chain of equations

$$\begin{array}{rcll}
{}^*y\left(\frac{x}{N}\right) &\approx& {}^*y(0) + ({}^*y(0)+2)\,\frac{x}{N} & = a\left(1+\frac{x}{N}\right) + \frac{2x}{N}\,; \\
{}^*y\left(\frac{2x}{N}\right) &\approx& {}^*y\left(\frac{x}{N}\right)\left(1+\frac{x}{N}\right) + \frac{2x}{N} & \approx a\left(1+\frac{x}{N}\right)^2 + \frac{2x}{N}\left(1+\frac{x}{N}\right) + \frac{2x}{N}\,; \\
& \vdots & & \vdots \\
{}^*y\left(\frac{kx}{N}\right) &\approx& {}^*y\left(\frac{(k-1)x}{N}\right)\left(1+\frac{x}{N}\right) + \frac{2x}{N} & \approx a\left(1+\frac{x}{N}\right)^k + \frac{2x}{N}\sum_{i=0}^{k-1}\left(1+\frac{x}{N}\right)^i\,; \\
& \vdots & & \vdots \\
{}^*y(x) &\approx& {}^*y\left(\frac{(N-1)x}{N}\right)\left(1+\frac{x}{N}\right) + \frac{2x}{N} & \approx a\left(1+\frac{x}{N}\right)^N + \frac{2x}{N}\sum_{i=0}^{N-1}\left(1+\frac{x}{N}\right)^i \\
&=& a\left(1+\frac{x}{N}\right)^N + \frac{2x}{N}\cdot\frac{(1+\frac{x}{N})^N-1}{(1+\frac{x}{N})-1} & = a\left(1+\frac{x}{N}\right)^N + 2\left(1+\frac{x}{N}\right)^N - 2 \\
&=& (a+2)\left(1+\frac{x}{N}\right)^N - 2 & \\
&\approx& (a+2)e^x - 2\,. &
\end{array}$$

In a general case, to finalise this derivation we should apply the standard part operator; in this particular example this happens to be trivial. The solution of our differential equation is therefore the function $y : \mathbb{R} \to \mathbb{R}$ defined by $y(x) = (a+2)e^x - 2$. ■

A.4.3 Brownian motion

Definition A.40 *A* Brownian motion *is a stochastic process* $b : \Omega \times [0,\infty) \to \overline{\mathbb{R}}$ *such that* $b(\omega, 0) = 0$ *for all* ω *and*

1. *if* $s_1 < t_1 \leq s_2 < t_2 \leq \ldots \leq s_n < t_n$, *then the random variables* $b(\cdot, t_1) - b(\cdot, s_1), \ldots b(\cdot, t_n) - b(\cdot, s_n)$ *are independent;*

2. *if* $s < t$, *the random variable* $b(\cdot, t) - b(\cdot, s)$ *is Gaussian distributed with mean zero and variance* $t - s$*;*

3. *for almost all* ω, *the path* $t \to b(w, t)$ *is continuous.*

The definition above introduces a one-dimensional Brownian motion. Higher dimensional versions are obtained by combining independent one-dimensional copies for each of the orthogonal axes.

The non-standard version of Brownian motion is quite easy and intuitive to construct. Choose an infinite integer $N \in {}^*\mathbb{N}$, and let T be the *hyperfinite* time-line

$$T = \left\{ 0, \frac{1}{N}, \frac{2}{N}, \ldots, \frac{N^2 - 1}{N}, N \right\} .$$

The collection Ω of all internal maps $\omega : T \to \{-1, 1\}$ is a hyperfinite set with 2^{N^2+1} elements. Considering each $\omega \in \Omega$ as a sequence $\omega(0), \omega(\frac{1}{N}), \omega(\frac{2}{N}), \ldots$ of coin tosses, where the value 1 means a step to the right and -1 a step to the left, we can define the *hyperfinite random walk* $B : \Omega \times T \to {}^*\mathbb{R}$ by setting

$$B\left(\omega, \frac{k}{N}\right) = \sum_{j=0}^{k-1} \frac{\omega(j/N)}{\sqrt{N}} .$$

This walk starts at the origin and walks along the hyperreal axis with steps of length $\frac{1}{\sqrt{N}}$.

The set Ω, with the algebra $\mathcal{A}$ of all its internal subsets and the normalised counting measure defined by setting

$$P(A) = \frac{|N|}{2^{N^2+1}}$$

for all $A \in \mathcal{A}$, can be transformed into a proper measure space $(\Omega, L(\mathcal{A}), L(P))$ by applying the so-called Loeb construction (see [65] for more details).

We can now define a standard map $b : \Omega \times [0,\infty) \to \overline{\mathbb{R}}$ by setting

$$b(\omega, t) = st(B(\omega, \tilde{t}\,)) ,$$

where $\tilde{t}$, as in the previous section, is the element of T to the immediate left of t.

Theorem A.41 *b is a Brownian motion on $(\Omega, L(\mathcal{A}), L(P))$.*

One particular application of this model of Brownian motion concerns stochastic differential equations. Let $f, g : \mathbb{R} \times [0,\infty) \to \mathbb{R}$ be bounded, continuous functions, and define an internal process $X : \Omega \times T \to {}^*\mathbb{R}$ inductively by

$$X(\omega, t) = \sum_{s=0}^{t} {}^*f(X(\omega, s), s)\Delta t + \sum_{s=0}^{t} {}^*g(X(\omega, s), s)\Delta B(\omega, s) .$$

Then the standard process $x(\omega, t) = st(X(\omega, \tilde{t}))$ is a solution of the stochastic differential equation

$$x(\omega, t) = \int_0^t f(x(\omega, s), s)ds + \int_0^t g(x(\omega, s), s)db(\omega, s)\,.$$

This is proved by checking that

$$\int_0^t f(x(\omega, s), s)ds = st\left(\sum_{s=0}^{\tilde{t}} {}^*f(X(\omega, s), s)\Delta t\right)$$

and

$$\int_0^t g(x(\omega, s), s)db(\omega, s) = st\left(\sum_{s=0}^{\tilde{t}} {}^*g(X(\omega, s), s)\Delta B(\omega, s)\right)\,.$$

The proof of the first equality is very similar to the proof of Peano's existence theorem in the previous section, whereas the proof of the second one is a little more technical as it involves a stochastic integral.

This method for proving existence results is due to Keisler [50], and can be extended to more complicated situations where f and g are no longer continuous and where the processes take values in higher dimensional spaces. In this paper, Keisler obtained, in particular, new, strong existence results for stochastic differential equations of the Itô type. In a series of papers [23, 24, 25, 26, 27], Cutland mixed Keisler's ideas with innovations of his own to study optimal controls in both deterministic and stochastic settings. Hoover & Perkins [45], Lindstrøm [62, 63, 64], and Stoll [88] extended the non-standard theory of stochastic integration to include integration with respect to semi-martingales, infinite dimensional Brownian motion, and white noise.

Appendix B

A note on inter-symbol interference

In this appendix, we explain the effect of inter-symbol interference (also called auto-interference) that we have mentioned in Chapter 4. We refer to this effect in the discussion related to the Outer Loop Power Control (OLPC).

The goal of OLPC, as we have explained in Section 4.3.3, is to adapt the target SIR, used by the Inner Loop, to the changes in the radio environment. The question is why would this target ever change? Indeed, the signal to interference ratio reflects the power of the received signal as compared to the interference level, and thus it is measured at reception, that is *after* the channel. The naive assumption would be that, as long as this ratio is sufficiently great, this should be sufficient to decode the signal with the constant error rate, and there should be no reason to modify it. This would be the case if there were no multiple propagation paths. These multiple paths introduce an interference between different copies of the user's signal, which results in additional errors in the decoding process. Thus, changes in the environment, i.e. in the delays and number of different trajectories, affects the signal to interference ratio necessary to guarantee a constant error rate, which gives the main reason for the Outer Loop's existence.

Therefore, to demonstrate the auto-interference effect, we place ourselves in the context of a transmission over a multipath channel using a simplified (without loss of generality) coding chain, and we observe the interference of the transmitted data on itself, which occurs due to the difference of delays on various trajectories.

Assume that we have to transmit a chain of symbols $s_0, s_1, \ldots, s_{M-1}$. We spread every symbol with a spreading sequence $c_0, c_1, \ldots, c_{Q-1}$ to obtain a sequence $u_0, u_1, \ldots, u_{N-1}$ defined by

$$u_k = s_{\lfloor k/Q \rfloor} c_{k \bmod Q}\,, \tag{B.1}$$

where Q is the spreading factor such that $M\,Q = N$ is the length of the actual sequence of chips that will be transmitted over the channel.

Before transmitting, this sequence is scrambled with a scrambling code that we shall denote $a_0, a_1, \ldots, a_{N-1}$. Thus the final transmitted signal can be expressed as

$$u(t) = A \sum_{n=0}^{N-1} a_n\, u_n\, p(t - nT_c)\,, \tag{B.2}$$

where A is the constant transmit power, T_c is the chip duration, and $p(t)$ is the pulse shape such that $(p \star p)(nT_c) = \delta_n$ (i.e. it equals 1 if $n = 0$ and 0 otherwise).

The received signal can then be expressed by

$$s(t) = \sum_{k=0}^{N_p-1} \alpha_k \, u(t - d_k T_c) + n(t) = A \sum_{k=0}^{N_p-1} \left[\alpha_k \sum_{n=0}^{N-1} a_n \, u_n \, p\Big(t - (n + d_k) T_c\Big) \right] + n(t) \,, \quad \text{(B.3)}$$

where N_p is the number of paths, α_k and $d_k T_c$ are correspondingly the attenuation and the delay [1] of the path number k, and $n(t)$ is the additive white noise component. For our purposes, we can either assume that there is no interference from other users or, otherwise, also model it as a Gaussian noise included in this case into $n(t)$. Whatever the choice, it does not affect the subsequent calculations.

At the receiver, the estimated symbols are then obtained by taking a convolution of the recieved signal with the pulse shape, multiplying by the conjugated scrambling code, and despreading:

$$\hat{s}_m = \sum_{i=0}^{Q-1} c_i \, a^*_{m\,Q+i} (s \star p)\Big((mQ + i)\, T_c\Big) \,. \quad \text{(B.4)}$$

Substituting (B.3), and noticing once again that $(p \star p)(nT_c) = \delta_n$, we can express the right-hand part as follows.

$$\hat{s}_m = \sum_{i=0}^{Q-1} c_i \, a^*_{m\,Q+i} \left[A \sum_{k=0}^{N_p-1} \alpha_k \, a_{m\,Q+i-d_k} \, u_{m\,Q+i-d_k} + \int_{-\infty}^{+\infty} p\Big(t - (mQ + i)\, T_c\Big) n(t) dt \right] \quad \text{(B.5)}$$

Substituting here the expression (B.1) for $u_{m\,Q+i-d_k}$ provides

$$\begin{aligned} \hat{s}_m \;\; = \;\; & A \sum_{i=0}^{Q-1} c_i \, a^*_{m\,Q+i} \sum_{k=0}^{N_p-1} \alpha_k \, a_{m\,Q+i-d_k} \, s_{m+\lfloor \frac{i-d_k}{Q} \rfloor} c_{(i-d_k) \bmod Q} \\ & + \int_{-\infty}^{+\infty} n(t) \left[\sum_{i=0}^{Q-1} c_i \, a^*_{m\,Q+i} \, p\Big(t - (mQ + i)\, T_c\Big) \right] dt \,. \end{aligned} \quad \text{(B.6)}$$

Without loss of generality we can suppose now that the receiver is synchornised with the path number 0 (i.e. $d_0 = 0$). This allows us to further develop expression (B.6) (we denote the integral in the right-hand side by $N(t)$):

$$\hat{s}_m = A \sum_{i=0}^{Q-1} c_i \, a^*_{m\,Q+i} \left[\alpha_0 \, a_{m\,Q+i} \, s_m \, c_i + \sum_{k=1}^{N_p-1} \alpha_k \, a_{m\,Q+i-d_k} \, s_{m+\lfloor \frac{i-d_k}{Q} \rfloor} c_{(i-d_k) \bmod Q} \right] + N(t) \,, \quad \text{(B.7)}$$

which is transformed by opening the parenthesis into

$$\hat{s}_m = A \alpha_0 \, s_m + A \sum_{i=0}^{Q-1} c_i \, a^*_{m\,Q+i} \sum_{k=1}^{N_p-1} \alpha_k \, a_{m\,Q+i-d_k} \, s_{m+\lfloor \frac{i-d_k}{Q} \rfloor} c_{(i-d_k) \bmod Q} + N(t) \,. \quad \text{(B.8)}$$

The first component in the right-hand part of equation (B.8) is the original transmitted symbol multiplied by the transmit power and the attenuation of the first path; the third one corresponds to the white noise and is accounted for in the SIR estimation; the second — not necessarily zero — component, however, only appears at the channel decoding stage, and represents exactly the inter-symbol interference.

[1] We follow here the common practice by assuming that the delays on all paths are multiples of chip duration.

Appendix C

Probabilistic background

In this appendix, we present two elementary facts from the probability theory, namely the notion of stochastic processes and the De Moivre-Laplace theorem. The former is referred to in Section 4.6 of Chapter 4, where it serves to establish the relations between parameters of two algorithms for the outer loop power control that guarantee [close to] optimal performances of these algorithms. The latter allows us to determine, in the discussion of Section 4.6.3 of the same chapter, the number of bits or blocks necessary to estimate the corresponding error rates.

C.1 Discussion of stochastic processes

Definition C.1 *A* discrete stochastic process *is a map $X : \Omega \times \mathbb{N} \to \mathbb{R}$ on some probability space $(\Omega, \mathcal{A}, P)$, such that $\omega \to X(\omega, n)$ is measurable for each n. This last map, which we denote by X_n, is said to represent the state of the process at the moment n.*

Theorem C.2 *Let $\{X_n\}$ be a stochastic process on $\mathbb{R}$ defined by setting $X_{n+1} = F(X_n)$, where*

$$F(X) = \begin{cases} X + a, & \text{with probability} \quad p(X) \\ X - b, & \text{with probability} \quad q(X) \\ X, & \text{with probability} \quad 1 - p(X) - q(X)\,, \end{cases} \tag{C.1}$$

with $a, b > 0$ and $p(x) + q(x) \leq 1$ for all $x \in \mathbb{R}$. Suppose also that this process converges to a stationary distribution π. Then, denoting by $\mathbb{E}_\pi[p(x)]$ and $\mathbb{E}_\pi[q(x)]$ the expectations in stationary distribution of the probabilities of an upwards and downwards steps correspondingly, we have the following relation

$$a\,\mathbb{E}_\pi[p(x)] = b\,\mathbb{E}_\pi[q(x)]\,. \tag{C.2}$$

Proof. First of all observe that, as π is the stationary distribution, it satisfies the following equation

$$\pi(x) = \pi(x+b)q(x+b) + \pi(x-a)p(x-a) + \pi(x)(1 - p(x) - q(x))\,, \tag{C.3}$$

which means basically that the probability of the process' being in in a given point x is equal to the sum over each point, from where one can get to x, of probabilities of the process' being there in the first place multiplied by the probability of the corresponding transition.

Let us now compute the expectation of x in the stationary distribution

$$\mathbb{E}_\pi[x] = \int_{-\infty}^{+\infty} x\pi(x)dx.$$

Substituting (C.3) into this equation we obtain

$$\begin{aligned}
&\mathbb{E}_\pi[x] = \\
&= \int_{-\infty}^{+\infty} x\pi(x+b)q(x+b)dx + \int_{-\infty}^{+\infty} x\pi(x-a)p(x-a)dx + \int_{-\infty}^{+\infty} x\pi(x)(1-p(x)-q(x))dx \\
&= \int_{-\infty}^{+\infty} (x-b)\pi(x)q(x)dx + \int_{-\infty}^{+\infty} (x+a)\pi(x)p(x)dx + \int_{-\infty}^{+\infty} x\pi(x)(1-p(x)-q(x))dx \\
&= \int_{-\infty}^{+\infty} x\pi(x)q(x)dx - b\int_{-\infty}^{+\infty} \pi(x)q(x)dx + \int_{-\infty}^{+\infty} x\pi(x)p(x)dx + a\int_{-\infty}^{+\infty} \pi(x)p(x)dx \\
&\quad + \int_{-\infty}^{+\infty} x\pi(x)(1-p(x)-q(x))dx \\
&= a\int_{-\infty}^{+\infty} \pi(x)p(x)dx - b\int_{-\infty}^{+\infty} \pi(x)q(x)dx + \int_{-\infty}^{+\infty} x\pi(x)dx \\
&= a\,\mathbb{E}_\pi[p(x)] - b\,\mathbb{E}_\pi[q(x)] + \mathbb{E}_\pi[x]\,.
\end{aligned}$$

Finally, eliminating $\mathbb{E}_\pi[x]$ in both left- and right-hand side of the equation above, we obtain the desired relation. ■

Lemma C.3 *Consider a differentiable function $f \in C^1(\mathbb{R})$, such that its first derivative satisfies the Lipschitz condition: for all $x, y \in \mathbb{R}$ we have $|f'(x) - f'(y)| \leq c|x-y|$, where c is a constant real number. Let X be a random variable on $\mathbb{R}$ with probability density π, and finite expectation $\mathbb{E}[X] = m$ and variation $\mathbb{E}[(X-m)^2] = \sigma^2$. Then one has*

$$\Big|\mathbb{E}[f(X)] - f(m)\Big| \leq c\,\sigma^2\,.$$

Proof. First of all observe that in the conditions of the lemma

$$f(x) = f(m) + \int_m^x f'(x)dx\,.$$

Denote by I_x the interval $[m, x]$ for $x \geq m$, and $[x, m]$ for $x \leq m$. As f' is Lipschitz it is also continuous, and therefore for any x there exists $x' \in I_x$ such that the integral in the right-hand part of the equation above equals $f'(x')(x-m)$. We can now write

$$\begin{aligned}
&\Big|\mathbb{E}[f(X)] - f(m)\Big| = \\
&= \left|\int_{\mathbb{R}} \Big(f(x) - f(m)\Big)\pi(x)dx\right| \\
&= \left|\int_{\mathbb{R}} f'(x')(x-m)\pi(x)dx\right| \\
&= \left|\int_{\mathbb{R}} \Big(f'(x') - f'(m) + f'(m)\Big)(x-m)\pi(x)dx\right| \\
&= \left|\int_{\mathbb{R}} \Big(f'(x') - f'(m)\Big)(x-m)\pi(x)dx + f'(m)\int_{\mathbb{R}} (x-m)\pi(x)dx\right|
\end{aligned}$$

$$
\begin{aligned}
&= \left| \int_{\mathbb{R}} \Big(f'(x') - f'(m) \Big)(x-m)\pi(x)dx \right| \\
&\leq \int_{\mathbb{R}} \left| \Big(f'(x') - f'(m) \Big)(x-m) \right| \pi(x)dx \\
&\leq c \int_{\mathbb{R}} (x-m)^2 \pi(x)dx = c\,\sigma^2 \,,
\end{aligned}
$$

which proves the lemma. ■

Note C.4 Lemma C.3 implies that, provided the probabilities $p(x)$ and $q(x)$ in (C.1) satisfy the appropriate conditions, a and b can always be chosen in such a way, as to obtain an arbitrarily small error in the following approximation of (C.2)

$$a\,p(m) = b\,q(m)\,, \tag{C.4}$$

where $m = \mathbb{E}_\pi[x]$. This is also confirmed by the intuition that the mean value of the process defined by (C.1) in the stationary distribution is the one where the conditional expectation of the next step is zero, i.e.

$$\mathbb{E}\Big[\Delta X \Big| X = m\Big] = a\,p(m) - b\,q(m) = 0\,.$$

C.2 Theorem of De Moivre-Laplace

The following theorem is a special case of the central limit theorem. It states that the binomial distribution of the number of "successes" in a series of independent identically distributed Bernoulli trials with probability p of success on each trial is approximately a normal distribution if n is large, or, more precisely, that after standardizing, the probabilities converge to those assigned by the standard normal distribution.

Theorem C.5 (De Moivre-Laplace) *Let $\{X_k\}_{k\in\mathbb{N}}$ be a series of independent identically distributed Bernoulli trials, i.e. for all k one has $X_k \in \{0,1\}$ and $P(X_k = 1) = p$. Let $N_n = \sum_{k=1}^n X_k$ be the number of occurences of 1 in $\{X_k\}$ up to the moment n. Then the following limit equation holds for all $a, b \in \mathbb{R}$.*

$$\lim_{n\to\infty} P\left(a < \frac{N_n - np}{\sqrt{np(1-p)}} < b \right) = \int_a^b g(y)dy\,, \tag{C.5}$$

where $g()$ is the density of the standardized normal probability distribution on $\mathbb{R}$ with zero mean and variance 1.

Let us denote by $G(b)$ the integral in the right part of (C.5) with an assumtion that $a = -b$ and $b > 0$. The following corollary then provides an estimate of the number of trials necessary to obtain a good approximation of p.

Corollary C.6 *Consider $\{X_k\}_{k\in\mathbb{N}}$ and N_n defined as in Theorem C.5, and take*

$$n = \left\lceil \left(\frac{b}{\varepsilon}\right)^2 p(1-p) \right\rceil\,, \tag{C.6}$$

where ε and b are two positive integers, and $\lceil x \rceil$ is the smallest integer greater than x. If ε is sufficientely small, then with probability close to $G(b)$ the proportion N_n/n of successes among the first n trials approximates p with precision ε, i.e.

$$P\left(\left|\frac{N_n}{n} - p\right| < \varepsilon\right) \approx G(b)\,. \tag{C.7}$$

Proof (corollary). Using the notations introduced above, one can rewrite (C.5) as

$$\lim_{n\to\infty} P\left(-b < \frac{N_n - np}{\sqrt{np(1-p)}} < b\right) = G(b)\,.$$

Dividing both inegalities by $\sqrt{\frac{n}{p(1-p)}}$, we obtain

$$\lim_{n\to\infty} P\left(\left|\frac{N_n}{n} - p\right| < b\sqrt{\frac{p(1-p)}{n}}\right) = G(b)\,.$$

Thus, to obtain (C.7) we have to take n such that

$$\varepsilon \geq b\sqrt{\frac{p(1-p)}{n}}\,,$$

and therefore

$$n \geq \left(\frac{b}{\varepsilon}\right)^2 p(1-p)\,.$$

■

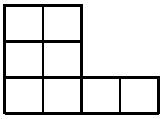
Appendix D

Combinatorial background

In this appendix, we present the combinatorial background related to the notions that we use in Chapter 6, i.e. partitions, Young tableaux, symmetric functions, and plactic monoid, as well as some operations on them.

D.1 Partitions and Young tableaux

A *partition* is a finite non-decreasing sequence $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_m)$ of positive integers. The number m of elements of λ is called the *length* of the partition λ. One can represent each such partition λ by a *Ferrers diagram* of *shape* λ, that is to say by a diagram of $\lambda_1 + \ldots + \lambda_m$ boxes whose i-th row contains exactly λ_i boxes for every $1 \leq i \leq m$. The Ferrers diagram associated with the partition $\lambda = (2, 2, 4)$ is given below.



The *conjugate* partition $\lambda\tilde{}$ of a given partition λ is the partition obtained by reading the heights of the columns of the Ferrers diagram associated with λ. For instance, the conjugate partition of $\lambda = (2, 2, 4)$ is $\lambda\tilde{} = (1, 1, 3, 3)$.

When λ is a partition whose Ferrers diagram is contained in the square represented by the partition $N^N = (N, \ldots, N)$ with N rows of length N, one can also define the *complementary* partition $\overline{\lambda}$ of λ which is the conjugate of the partition ν whose Ferrers diagram is the complement (read from bottom to top) of the Ferrers diagram of λ in the square (N^N). Note that this definition is relative to a given size N and that the square does not have to be the smallest one containing λ. For instance, for $N = 6$ and $\lambda = (1, 1, 2, 3)$, we have $\nu = (3, 4, 5, 5, 6, 6)$ and $\overline{\lambda} = (2, 4, 5, 6, 6, 6)$ (see Figure D.1).

Let A be a totally ordered alphabet. A *tabloid* of shape λ over A is a filling of the boxes of a Ferrers diagram of shape λ with letters of A. A tabloid is called a *Young tableau* when its rows and its columns consist respectively of non-decreasing and strictly increasing sequences of letters of A. One can see below a Young tableau of shape $(2, 2, 4)$ over $A = \{ a_1 < \ldots < a_5 \}$.

a_3	a_5		
a_2	a_2		
a_1	a_1	a_1	a_4

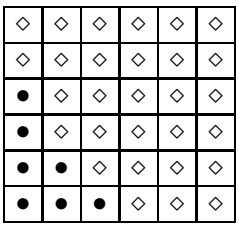


Figure D.1: Two complementary partitions : $\lambda = (1, 1, 2, 3)$ and $\overline{\lambda} = (2, 4, 5, 6, 6, 6)$.

One associates with any Young tableau T over A the monomial A^T which is the product of all letters of A that occur in the different boxes of T. One has for instance $A^T = a_1^3\, a_2^2\, a_3\, a_4\, a_5$ for T the Young tableau of the last example. The *Schur function* $s_\lambda(A)$ associated with the partition λ is then defined as the sum of all monomials A^T for T running over all Young tableaux of shape λ. We recall that the Schur functions are symmetric polynomials that form a linear basis of the algebra of symmetric polynomials over A (see Section D.2).

D.1.1 Knuth's bijection

Knuth's bijection is a famous one-to-one correspondence between $\{0, 1\}$-matrices and pairs of Young tableaux of conjugate shapes (cf. [51]). It is based on the *column insertion* process which is a classical combinatorial construction that we shall present now. Let A be a totally ordered alphabet. The fundamental step of the column insertion process associates with a letter $a \in A$ and a Young tableau T over A a new Young tableau $T(a)$ over A defined as follows.

1. If a is strictly larger than all the entries of the first column of T, the tableau $T(a)$ is obtained by putting a in a new box at the top of the first column of T.

2. Otherwise, one can consider the smallest entry b of the first column of T which is greater than or equal to a. The tableau $T(a)$ is then obtained by replacing b by a and by applying recursively our insertion scheme, starting now by trying to insert b in the second column of T. Our process continues until a replaced entry can go at the top of the next column or until it becomes the only entry of a new column.

One can easily check that $T(a)$ is always a Young tableau. Moreover, each step of our process can be reversed if one knows which new box it has created. Let now $w = a_1 \dots a_N$ be a word over A. The result of the column insertion process applied to w is the Young tableau obtained by column inserting successively $a_1, \dots, a_N$ as described above, starting from the empty Young tableau.

Note D.1 The Young tableau which is obtained by applying the column insertion process to a word $w = a_1 \dots a_N$ over A is the same as the tableau obtained by applying the row insertion process (i.e. Schensted's algorithm) to its mirror image $\tilde{w} = a_N \dots a_1$ (see [34] for more details).

We are now in the position to present Knuth's construction. Let M be a matrix from the set $\mathcal{M}_{N\times N}(\{0, 1\})$ of square $\{0, 1\}$-matrices of order N. Knuth's bijection associates with M a pair (P, Q) of Young tableaux with conjugate shapes over the alphabet $[1, N]$ as described below.

1. Construct the 2-row array A_N which results by listing the N^2 pairs (i,j) of $[1,N]\times[1,N]$ in lexicographic order, i.e.

$$A_N = \begin{pmatrix} 1 & \dots & 1 & 2 & \dots & 2 & \dots\dots & N & \dots & N \\ 1 & \dots & N & 1 & \dots & N & \dots\dots & 1 & \dots & N \end{pmatrix}.$$

2. Take in this array all the entries corresponding to the 1's of M in order to get an array

$$\mathcal{A}(M) = \begin{pmatrix} u_1 & u_2 & \dots\dots & u_r \\ v_1 & v_2 & \dots\dots & v_r \end{pmatrix}.$$

3. Form the word $w_1(M) = v_1 \dots v_r$ obtained by reading from left to right the bottom entries (the entries of the second row) of $\mathcal{A}(M)$. The column insertion process applied to $w_1(M)$ gives the Young tableau P.

4. Form finally the second Young tableau Q by placing for every $i \in [1,r]$ the i-th element u_i of the first row of $\mathcal{A}(M)$ in the box which is conjugate to the i-th box created during the column insertion process that led to P.

By reversing the steps of the described construction, we can recover the array $\mathcal{A}(M)$ (and hence our matrix M) from the pair (P,Q). We find the box in which Q has the largest entry; if there are several equal entries, the box that is farthest to the right is selected. Then we perform the reverse column insertion to P starting with the conjugate of the selected box and remove the selected box from Q. We obtain a new pair of Young tableaux with conjugate shapes and perform the same procedure up to the moment when we get two empty Young tableaux.

Example D.2
Let us consider the matrix

$$M = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 1 \end{pmatrix}.$$

Then the arrays $\mathcal{A}_3$ and $\mathcal{A}(M)$ are respectively equal to

$$\mathcal{A}_3 = \begin{pmatrix} 1 & 1 & \boxed{1} & \boxed{2} & 2 & 2 & 3 & \boxed{3} & \boxed{3} \\ 1 & 2 & \boxed{3} & \boxed{1} & 2 & 3 & 1 & \boxed{2} & \boxed{3} \end{pmatrix} \quad \text{and} \quad \mathcal{A}(M) = \begin{pmatrix} 1 & 2 & 3 & 3 \\ 3 & 1 & 2 & 3 \end{pmatrix},$$

where in $\mathcal{A}_3$ we boxed the entries corresponding to the 1's of M. Thus $w_1(M) = (3,1,2,3)$. Knuth's bijection associates with M the following pair of Young tableaux of conjugate shapes:

$$(P,Q) = \left(\begin{array}{|c|} \hline 3 \\ \hline 2 \\ \hline 1 \\ \hline \end{array} \begin{array}{|c|} \hline \\ \hline \\ \hline 3 \\ \hline \end{array} \;,\; \begin{array}{|c|c|} \hline 2 & \\ \hline 1 & 3 \\ \hline \end{array} \begin{array}{|c|} \hline \\ \hline 3 \\ \hline \end{array} \right).$$

■

Let us now present an extension of this bijection, presented in [54], that allows us to obtain a square tabloid of shape N^N over two alphabets $\Delta = \{\delta_1, \dots, \delta_N\}$ and $X = \{\chi_1, \dots, \chi_N\}$ from the pair of Young tableaux of conjugated form obtained by Knuth's bijection from a $N \times N$ square {0,1}-matrix. The additional transformation consists then in the following two simple steps.

1. Construct a Young tableau $\overline{Q}$ from Q column by column, by taking for each $i = 1, \ldots, N$ all numbers from $[1, \ldots, N]$ that do not belong to the column $N-i+1$ of Q (if this column is empty then that implies all numbers $1, \ldots, N$), and placing them into column i of $\overline{Q}$.
2. Replace all entries i in P by δ_i, and in $\overline{Q}$ by χ_i.
3. Observe now, that the shape of $\overline{Q}$ is complementary to that of Q in the square N^N, and therefore to finalise our construction it is sufficient to place $\overline{Q}$ in such a way that its corner is in the upper right-hand corner of N^N.

Example D.3
Let us continue with the pair (P, Q) of the previous example. Applying the first step of the procedure described above, we obtain, indeed, the following tableau $\overline{Q}$ of shape $(2, 3)$ complementary to that of P (i.e. $(1, 1, 2)$) in the 3 by 3 square.

$$\overline{Q} = \begin{array}{|c|c|c|}\hline 2 & 2 & \multicolumn{1}{c}{} \\ \hline 1 & 1 & 3 \\ \hline \end{array}\,.$$

Replacing now numbers by letters from Δ and X and combining P and $\overline{Q}$ we obtain the following square tabloid.

$$\begin{array}{|c|c|c|}\hline \delta_3 & \chi_2 & \chi_1 \\ \hline \delta_2 & \chi_2 & \chi_1 \\ \hline \delta_1 & \delta_3 & \chi_3 \\ \hline \end{array}\,.$$

■

It is clear that this procedure can be inversed, and therefore, by combining it with the Knuth's bijection, we obtain a bijection between $N \times N$ square {0,1}-matrices and square tabloids of shape N^N over alphabets Δ and X.

D.1.2 Plactic equivalence

The column insertion process can also be described algebraically by the plactic formalism developed by Lascoux and Schützenberger (cf. [58]). Let A be a totally ordered alphabet. The *plactic monoid* is the monoid constructed over A and subject to the following relations (discovered by Knuth (cf. [51])):

$$\begin{cases} acb \equiv cab\,, & \text{for every } a \leq b < c \text{ in } A \\ bca \equiv bac\,, & \text{for every } a < b \leq c \text{ in } A\,. \end{cases}$$

Two words over A are identified under the plactic relations if and only if the Young tableaux obtained by applying the column insertion process to their mirror images are equal (cf. [34, 58]).

One can associate with a Young tableau T a word $w(T)$ over A by reading the columns of T from top to bottom and left to right. The words associated with Young tableaux in such a way are called *tableau words*.

Example D.4
The tableau word associatied with the Young tableau

$$\mathrm{T} = \begin{array}{|c|c|}\hline a_3 & a_5 \\ \hline a_2 & a_2 \\ \hline a_1 & a_1 \\ \hline \end{array}\!\!\begin{array}{|c|c|}\hline \multicolumn{1}{c}{} & \multicolumn{1}{c}{} \\ \hline \multicolumn{1}{c}{} & \multicolumn{1}{c}{} \\ \hline a_1 & a_4 \\ \hline \end{array} \qquad \text{is } w(T) = a_3\, a_2\, a_1\, a_5\, a_2\, a_1\, a_1\, a_4\,.$$

■

Note that applying the column insertion to the mirror image of a tableau word $w(T)$ yields the tableau T. Observe also (see [34, 58]) that a word over A is equivalent with respect to the plactic relations to a unique tableau word (which is therefore associated with the Young tableau given by the column insertion process applied to the mirror image of w). In other words, we can state the following proposition.

Proposition D.5 *If u is a word over a totally ordered alphabet A, and $\overline{u}$ is its mirror image, then*

$$w(T_{\overline{u}}) \equiv u\,,$$

where $T_{\overline{u}}$ is the Young tableau obtained by applying column insertion to $\overline{u}$.

Note D.6 If u and v are two words over A such that $u \equiv v$ then by the previous proposition we have

$$w(T_{\overline{u}}) \equiv u \equiv v \equiv w(T_{\overline{v}})\,,$$

and therefore $T_{\overline{u}} = T_{\overline{v}}$.

Denoting by $R(w,k)$ the maximum total length of k increasing subsequences of w, we can state the following theorem, which is a consequence of Green's theorem (see [37]).

Theorem D.7 *If u and v are two words over a totally ordered alphabet A, such that $u \equiv v$, then for any $k \geq 0$ we have*

$$R(u,k) = R(v,k)\,.$$

Finally, we shall mention another important property of plactic equivalence. Let $w = w_1\, w_2\, \ldots w_n$ be a word over a totally ordered alphabet A, and let $a < b$ be two letters of A. We shall denote by $w_{a,b}$ the subword of w consisting of all its letters w_i, such that $a \leq w_i \leq b$. In other words, $w_{a,b}$ is the projection of w on the interval $[a,b] \subset A$.

Theorem D.8 *Let u and v be two words over a totally ordered alphabet A. Let $a < b$ be two letters of A. Then $u \equiv v$ implies $u_{a,b} \equiv v_{a,b}$.*

D.2 Symmetric functions background

In this section, we briefly present the notion of symmetric functions and a number of related concepts. For a more comprehensive study we refer the reader to the classical textbook [66] by I. G. MacDonald.

Definition D.9 *A function of multiple variables is said to be* symmetric *if it is invariant under any permutation of its variables.*

Let X be a set of indeterminates. Symmetric functions over X form an algebra that is denoted by $Sym(X)$. We define the *complete symmetric functions* $S_k(X)$ by their generating series

$$\sigma_z(X) = \sum_{n=0}^{+\infty} S_n(X)\, z^n = \prod_{x \in X} \frac{1}{1 - x\, z}\,. \tag{D.1}$$

We also define in the same way the *elementary symmetric functions* $\Lambda_k(X)$ by their generating series (which is a polynomial when X is finite)

$$\lambda_z(X) = \sum_{n=0}^{+\infty} \Lambda_n(X)\, z^n = \prod_{x \in X} (1 + x\, z)\ . \tag{D.2}$$

In order to use complete and elementary symmetric functions indexed by any integer $k \in \mathbf{Z}$, we also set $S_k(X) = \Lambda_k(X) = 0$ for every $k < 0$. Every symmetric function can be expressed in a unique way as a product of complete or elementary symmetric functions. For every n-uple $I = (i_1, \ldots, i_n) \in \mathbf{Z}^n$, we now define the *Schur function* $s_I(X)$ as the minor taken over the rows $1, 2, \ldots, n$ and the columns $i_1+1, i_2+2, \ldots, i_n+n$ of the infinite matrix $\mathbf{S} = (S_{j-i}(X))_{i,j \in \mathbf{Z}}$, i.e.

$$s_I(X) = \begin{vmatrix} S_{i_1}(X) & S_{i_2+1}(X) & \ldots & S_{i_n+n-1}(X) \\ S_{i_1-1}(X) & S_{i_2}(X) & \ldots & S_{i_n+n-2}(X) \\ \vdots & \vdots & \vdots & \vdots \\ S_{i_1-n+1}(X) & S_{i_2-n+2}(X) & \ldots & S_{i_n}(X) \end{vmatrix}\ . \tag{D.3}$$

We also define more generally for every $I = (i_1, \ldots, i_n) \in \mathbf{Z}^n$ and $J = (j_1, \ldots, j_n) \in \mathbf{Z}^n$, the *skew Schur function* $s_{J/I}(X)$ as the minor of $\mathbf{S}$ taken over the rows $i_1 + 1, i_2 + 2, \ldots, i_n + n$ and the columns $j_1 + 1, j_2 + 2, \ldots, j_n + n$. The importance of Schur functions comes from the fact that the family of the Schur functions that are indexed by partitions form a classical linear basis of the algebra of symmetric functions.

Let us finally introduce the notion of multi-Schur function (see [59]) which is another natural generalization of usual Schur functions. Let $(X_i)_{1 \leq i \leq n}$ be a family of n sets of indeterminates. For every n-uple $I = (i_1, \ldots, i_n) \in \mathbf{Z}^n$, one defines then the multi-Schur function $S_I(X_1, \ldots, X_n)$ by the determinantal formula

$$s_I(X_1, \ldots, X_N) = \begin{vmatrix} S_{i_1}(X_1) & S_{i_2+1}(X_2) & \ldots & S_{i_n+n-1}(X_n) \\ S_{i_1-1}(X_1) & S_{i_2}(X_2) & \ldots & S_{i_n+n-2}(X_n) \\ \vdots & \vdots & \vdots & \vdots \\ S_{i_1-n+1}(X_1) & S_{i_2-n+2}(X_2) & \ldots & S_{i_n}(X_n) \end{vmatrix}\ . \tag{D.4}$$

Hence the usual Schur function $s_I(X)$ is exactly the multi-Schur function $s_I(X, \ldots, X)$.

D.2.1 Transformations of alphabets

Let X and Y be two sets of indeterminates. The complete symmetric functions of the formal set $X+Y$ are then defined by their generating series

$$\sigma_z(X+Y) = \sum_{n=0}^{+\infty} S_n(X+Y)\, z^n = \sigma_z(X)\, \sigma_z(Y)\ . \tag{D.5}$$

One also defines the complete symmetric functions of the formal set $X-Y$ by setting

$$\sigma_z(X-Y) = \sum_{n=0}^{+\infty} S_n(X-Y)\, z^n = \sigma_z(X)\, \lambda_{-z}(Y)\ . \tag{D.6}$$

A symmetric function F of the alphabet $X+Y$ or $X-Y$ is then an element of $Sym(X) \otimes Sym(Y)$ whose expression in this last algebra can be obtained by developping F as a product of complete symmetric functions of $X+Y$ or $X-Y$ that are elements of $Sym(X) \otimes Sym(Y)$ according to

the two defining relations (D.5) and (D.6). Note also that the complete symmetric functions of the formal set $-X$ are in particular defined by setting

$$\sigma_z(-X) = \sum_{n=0}^{+\infty} S_n(-X)\, z^n = \lambda_{-z}(X)\ . \tag{D.7}$$

In other words, if $F(X)$ is a symmetric function of the set X, the symmetric function $F(-X)$ is obtained by applying to F the algebra morphism that replaces $S_n(X)$ by $(-1)^n\,\Lambda_n(X)$ for every $n \geq 0$. Observe that the formal set $X-Y$ can also be defined by setting $X-Y = X+(-Y)$.

The expression of a Schur function of a formal sum of sets of indeterminates is in particular given by the Cauchy formula, which states that one has

$$s_\lambda(X+Y) = \sum_{\mu\subset\lambda} s_\mu(X)\, s_{\lambda/\mu}(Y) \tag{D.8}$$

for every partition λ. One must also point out that, for all partitions μ and λ such that $\mu \subset \lambda$, one has

$$s_{\lambda/\mu}(-X) = s_{\lambda\tilde{}/\mu\tilde{}}(X) \tag{D.9}$$

where $\lambda\tilde{}$ and $\mu\tilde{}$ are the conjugate partitions of λ and μ correspondingly. Note finally that the resultant of two polynomials can in particular be expressed as a rectangular Schur function of a difference of alphabets. Let X and Y be two sets of respectively N and M indeterminates. The expression

$$R(X,Y) = \prod_{x\in X,\, y\in Y} (x-y)$$

is then the resultant of the polynomials that have X and Y as sets of roots and one can prove that one has $R(X,Y) = S_{N^M}(X-Y)$ (see [59]).

D.2.2 Vertex operators

The vertex operator $\Gamma_z(X)$ transforms every symmetric function of $Sym(X)$ into a series of $Sym(X)[[z,z^{-1}]]$. As the Schur functions indexed by partitions form a linear basis in $Sym(X)$, it is sufficient to define $\Gamma_z(X)$ only on the elements of the latter. We put

$$\Gamma_z(X)(s_\lambda(X)) = \sum_{m=-\infty}^{\infty} s_{(\lambda,m)}(X)\, z^m$$

for every partition $\lambda = (\lambda_1,\ldots,\lambda_n)$, with $(\lambda,m) = (\lambda_1,\ldots,\lambda_n,m) \in \mathbf{Z}^{n+1}$ for every $m \in \mathbf{Z}$. The following formula due to Thibon (cf. [89]) gives then another explicit expression of the action of a vertex operator on a Schur function.

Proposition D.10 (Thibon; [89]) *Let λ be a partition. Then one has*

$$\Gamma_z(X)(s_\lambda(X)) = \sigma_z(X)\, s_\lambda(X-1/z)\ . \tag{D.10}$$

D.2.3 Lagrange's operators

Let $X = \{x_1,\ldots,x_N\}$ be a finite alphabet of N indeterminates. The *Lagrange interpolating operator* L is the operator that maps every polynomial f of $\mathbf{C}[X]$ symmetric in the last $N-1$

indeterminates, i.e. every element $f(x_1, X\backslash x_1)$ of $Sym(x_1) \otimes Sym(X\backslash x_1)$, onto the symmetric polynomial $L(f)$ of $Sym(X)$ defined by setting

$$L(f) = \sum_{k=1}^{N} \frac{f(x_k, X\backslash x_k)}{R(x_k, X\backslash x_k)}$$

where $R(A, B)$ stands again for the resultant of the two polynomials that have respectively the two sets of indeterminates A and B as sets of roots (cf. Section D.2.1). The following result, corresponding to the special case of Bott's formula for fibrations in projective lines (see [57, 58] for more details), gives then an interesting property of the Lagrange interpolation operator.

Theorem D.11 (Lascoux; [57]) *Let $X = \{ x_1, \ldots, x_N \}$ be an alphabet of N indeterminates and let $\lambda = (\lambda_1, \ldots, \lambda_n)$ be a partition that contains $\rho_{N-1} = (N-2, \ldots, 2, 1, 0)$. Then one has*

$$L(x_1^k \, s_\lambda(X\backslash x_1)) = s_{(\lambda, k-N+1)}(X) \tag{D.11}$$

for every $k \geq 0$, where the Schur function involved in the right hand side of relation (D.11) is indexed by the sequence $(\lambda, k-N+1) = (\lambda_1, \ldots, \lambda_n, k-N+1)$ of $\mathbf{Z}^{n+1}$.

Bibliography

[1] 3rd Generation Partnership Project (3GPP). *Performance Comparison of Chase Combining and Incremental Redundancy for HSDPA*, Nov. 2000. Technical Specification TSGR1 #17(00)1428.

[2] 3rd Generation Partnership Project (3GPP). *Performance Comparison of Hybrid-ARQ Schemes*, Oct. 2000. Technical Specification TSGR1 #17(00)1396.

[3] 3rd Generation Partnership Project (3GPP). *Enhanced HARQ Method with Signal Constellation Rearrangement*, Mar. 2001. Technical Specification TSGR1 #19(01)0237.

[4] 3rd Generation Partnership Project (3GPP). *Further Simulation Results on HARQ with Signal Constellation Rearrangement*, May 2001. Technical Specification TSGR1 #20(01)0537.

[5] 3rd Generation Partnership Project (3GPP). *Spreading and modulation (FDD)*, Sept. 2002. Technical Specification TS 25.213 v5.2.0.

[6] 3rd Generation Partnership Project (3GPP). *Multiplexing and channel coding (FDD)*, June 2004. Technical Specification TS 25.212 v5.9.0.

[7] J. M. Aein. Power balancing in systems employing frequency reuse. *COMSAT Technical Review*, 3(2):277–299, Fall 1973.

[8] M. Aldajani and A. Sayed. Adaptive predictive power control for the uplink channel in DS-CDMA cellular systems. *IEEE Transactions on Vehicular Technology*, 52(6):1447–1462, Nov. 2003.

[9] R. Alur, C. Courcoubetis, N. Halbwachs, T. A. Henzinger, P. H. Ho, X. Nicollin, A. Olivero, J. Sifakis, and S. Yovine. *The Algorithmic Analysis of Hybrid Systems*. Number 138 in Theoretical Computer Science. 1995.

[10] M. Andersin, Z. Rosberg, and J. Zander. Gradual removals in cellular PCS with constrained power control and noise. *Wireless Networks*, 2(1):27–43, Mar. 1996.

[11] S. Ariyavisitakul. SIR-based power control in a CDMA system. In *Proceedings of Global Telecommunications Conference, GLOBECOM'92*, volume 2, pages 868–873, Orlando, FL, USA, Dec. 1992.

[12] M. Barrett. Error probability for optimal and suboptimal quadratic receivers in rapid Rayleigh fading channels. *IEEE J. Select. Areas Commun.*, 5(2):302–304, Feb. 1987.

[13] J. Barwise. *Handbook of Mathematical Logic*. Number 90 in Studies in Logic and the Foundations of Mathematics. North Holland, 1977.

[14] A. Benveniste, P. Caspi, S. A. Edwards, N. Halbwachs, P. L. Guernic, and R. de Simone. The synchronous languages twelve years later. *Proc. of the IEEE, Special Issue on Embedded Systems*, 91(1):64–83, 2003.

[15] N. Billy. Outer loop power control. Internship report, Alcatel CIT, July 2000. confidential.

[16] S. Bliudze. Outer loop power control. Internship report, Alcatel CIT, Oct. 2001. confidential.

[17] G. Bloch-Morhange and E. Fontela. Mobile communication from voice to data: A morphological analysis. *The journal of policy, regulation and strategy for telecommunications*, 5(2):24–33, Feb. 2003.

[18] R. Burt. The wireless telephone. In *The Aerogram*, pages 139–141. The Aerogram Publishing Company, Nov. 1908. `http://earlyradiohistory.us/1908uwwt.htm`.

[19] D. Cha, J. Rosenberg, and C. Dym. *Fundamentals of Modeling and Analyzing Engineering Systems*. Cambridge University Press, 2000.

[20] D. Chase. Code combining: A maximum-likelihood decoding approach for combining an arbitrary number of noisy packets. *IEEE Trans. Commun.*, 33:593–607, May 1985.

[21] S. C. Chen, N. Bambos, and G. J. Pottie. On distributed power control for radio networks. In *Proceedings of the IEEE International Conference on Communications ICC'94*, volume 3, pages 1281–1285, May 1994.

[22] Cisco Systems. Overview of GSM, GPRS, and UMTS. In *Cisco Mobile Exchange (CMX) Solution Guide*, chapter 2.

[23] N. J. Cutland. Internal controls and relaxed controls. *Journal of London Mathematical Society*, 27:130–140, 1983.

[24] N. J. Cutland. Optimal controls for partially observed stochastic systems: an infinitesimal approach. *Stochastics*, 8:239–257, 1983.

[25] N. J. Cutland. Partially observed stochastic controls based on cumulative digital readouts of the observations. In *Proc. 4th IFIP Work. Conf. Stochastic Differential Systems*, pages 261–269, Marseille-Luminy, 1984. Springer-Verlag.

[26] N. J. Cutland. Infinitesimal methods in control theory: Deterministic and stochastic. *Acta Applicandae Mathematicae*, 5:105–136, 1986.

[27] N. J. Cutland. Infinitesimals in action. *Journal of the London Mathematical Society*, 35:202–216, 1987.

[28] D'Alembert and J. d. Le Rond. Article " différentiel ". In D. Diderot, D'Alembert, and J. d. Le Rond, editors, *Encyclopédie ou Dictionnaire raisonné des sciences, des arts et des métiers*, volume 4, pages 985–989. Briasson, David, Le Breton et Durand, Paris, 1754.

[29] F. Diener and G. Reeb. *Analyse non standard*. Hermann, Éditeurs des Sciences et des Arts, 1989.

[30] J.-L. Dornstetter, D. Krob, and J.-Y. Thibon. Fast and stable computation of error probability in rapid Rayleigh fading channels. In *Proceedings of AlgoTel*, 2000.

[31] J.-L. Dornstetter, D. Krob, J.-Y. Thibon, and E. A. Vassilieva. Performance evaluation of demodulation with diversity — a combinatorial approach I: Symmetric function theoretical methods. *Discrete Mathematics and Theoretical Computer Science*, 5:191–204, 2002.

[32] M. Fliess. Fonctionnelles causales non linéaires et indéterminées non commutatives. *Bull. Soc. Math. France*, 109:3–40, 1981.

[33] G. J. Foschini and Z. Miljanic. A simple distributed autonomous power control algorithm and itsconvergence. *IEEE Transactions on Vehicular Technology*, 42(4):641–646, nov 1993.

[34] W. Fulton. *Young Tableaux*. Cambridge University Press, 1997.

[35] P. Godlewski and L. Nuaymi. Auto-interference analysis in cellular systems. In *Proceedings of the IEEE Vehicular Technology Conference*, Houston, TX, USA, May 1999.

[36] S. A. Grandhi, J. Zander, and R. Yates. Constrained power control. *Wireless Personal Communications*, 1(4):257–270, Dec. 1994.

[37] C. Greene. Some partitions aassociated with a partially ordered set. *J. of Combin. Theory, Ser. A*, 20:69–79, 1976.

[38] J. Gu, X. Che, S. Nie, and D. Wang. QoS based outer loop power control for enhanced reverse links of CDMA systems. *Electronics Letters*, 41(11):659–661, may 2005.

[39] M. Guenach and L. Vandendorpe. Downlink performance analysis of a BPSK-based WCDMA using conventional rake receivers with channel estimation. *IEEE J. Select. Areas Commun.*, 19(11):2165–2176, Nov. 2001.

[40] F. Gunnarsson and F. Gustafsson. Power control in wireless communication networks — from a control theory perspective. In *IFAC World Congress*, Barcelone, 2002.

[41] F. Gunnarsson, F. Gustafsson, and J. Blom. Estimation and outer loop power control in cellular radio systems. Technical Report LiTH-ISY-R-2332, Division of Communication Systems, Linköpings universitet, SE-581 83 Linköping, Sweden, Feb. 2001. `ftp.control.isy.liu.se` as `2332.pdf`.

[42] L. Hanzo, T. H. Liew, and B. L. Yeap. *Turbo Coding, Turbo Equalisation and Space-Time Coding for Transmission over Fading Channels*. John Wiley & Sons Inc., 2002.

[43] T. A. Henzinger. The theory of hybrid automata. In *Proceedings of the 11th Annual IEEE Symposium on Logic in Computer Science, LICS'96*, pages 278–292. IEEE Society Press, 1996.

[44] H. Holma and A. Toskala, editors. *WCDMA for UMTS: Radio Access for Third Generation Mobile Communications*. John Wiley & Sons, Ltd, 2001.

[45] D. Hoover and E. Perkins. Nonstandard constructions of the stochastic integral and applications to stochastic differential equations I, II. *Transactions of the American Mathematical Society*, 275:1–58, 1983.

[46] J. P. Imhof. Computing the distribution of quadratic forms in normal variables. *Biometrika*, 48:419–426, 1961.

[47] The International Engineering Consortium. *UMTS Protocols and Protocol Testing*. Web ProForum Tutorials.

[48] C. U. Jensen and H. Lenzing. *Model Theoretical Algebra*. Gordon and Breach, 1989.

[49] H. Kawai, H. Suda, and F. Adachi. Outer-loop control of target SIR for fast transmit power control in turbo-coded w-cdma mobile radio. *Electronics Letters*, 35(9):699–701, Apr. 1999.

[50] H. Keisler. An infinitesimal approach to stochastic analysis. *Memoirs of the American Mathematical Society*, 297, 1984.

[51] D. E. Knuth. Permutations, matrices and generalized Young tableaux. *Pacific J. Math.*, 34:709–727, 1970.

[52] C.-S. Koo, S.-H. Shin, R. A. DiFazio, D. Grieco, and A. Zeira. Outer loop power control using channel-adaptive processing for 3G WCDMA. In *Proceedings of the IEEE Vehicular Technology Conference*, volume 1, pages 490–494, Apr. 2003.

[53] S. Kowalewski, O. Stursberg, M. Fritz, H. Graf, I. Hoffmann, J. Preußig, M. Remelhe, S. Simon, and H. Treseler. A case study in tool-aided analysis of discretely controlled continuous systems: The two tanks problem. In *Proceedings of the 5th International Workshop on Hybrid Systems*, pages 165–177, 1997.

[54] D. Krob and E. A. Vassilieva. Performance evaluation of demodulation with diversity — a combinatorial approach II: Bijective methods. *Discrete Applied Mathematics*, 145(3):403–421, 2005.

[55] I. Lakatos. Proofs and refutations. *The British Journal for the Philosophy of Science*, 14:1–25, 120–139, 221–45, 296–342, 1963–1964.

[56] I. Lakatos. *Preuves et réfutations*. Hermann, 1984.

[57] A. Lascoux. Inversion des matrices de Henkel. *Lin. Alg. and its Appl.*, 129:77–102, 1990.

[58] A. Lascoux and M.-P. Schützenberger. Le monoïde plaxique. *Quaderni de la Ricerca Scientifica, A*, 109:129–156, 1981.

[59] A. Lascoux and M.-P. Schützenberger. Formulaire raisonné de fonctions symétriques. *Public. LITP, Paris 7*, 1985.

[60] G. Lelievre. Aperçu de théorie axiomatique des ensembles. In *Compléments d'analyse*, volume 1 *Topologie* (appendice). Ouvrage hors collection des cahiers de Fontenay, June 1985.

[61] S. Lindmark. Coordinating the early commercialization of general purpose technologies: The case of mobile data communications. *Innovation: Management, Policy & Practice*, 7(1):39–60, Feb. 2005.

[62] T. Lindstrøm. The structure of hyperfinite stochastic integrals, 198?

[63] T. Lindstrøm. Hyperfinite stochastic integration I, II, III. *Mathematica Scandinavica*, 46:265–333, 1980.

[64] T. Lindstrøm. Stochastic integration in hyperfinite dimensional linear spaces. In A. Hurd, editor, *Nonstandard Analysis — Recent Developments*, number 983 in Lecture Notes in Mathematics, pages 134–161. Springer-Verlag, Berlin and New-York, 1983.

[65] T. Lindstrøm. An invitation to nonstandard analysis. In N. Cutland, editor, *Nonstandard Analysis and its Applications*, number 10 in London Mathematical Society Student Texts. Cambridge University Press, 1988.

[66] I. G. MacDonald. *Symmetric Functions and Hall Polynomials*. Oxford University Press, 2nd edition edition, July 1999.

[67] M. J. Mauboussin. The economics of customer business. Technical report, Legg Mason Capital Management, Dec. 2004.

[68] J.-P. Meinadier. *Ingénierie et intégration des systèmes*. Hermès Science Publications, 1998.

[69] J.-P. Meinadier. *Le métier d'intégration de systèmes*. Lavoisier, Dec. 2002.

[70] K. R. Meyer and G. R. Hall. *Introduction to Hamiltonian Dynamical Systems and the N-Body Problem*. Number 90 in Applied Mathematical Sciences. Springer Verlag, 1992.

[71] V. Mitlin. Optimal selection of ARQ parameters in QAM channels. *Wireless Communications and Mobile Computing*, to appear. DOI 10.1002/wcm.206.

[72] D. Mitra. An asynchronous distributed algorithm for power control in cellular radio systems. In J. M. Holtzman, editor, *Proceedings of 4th WINLAB Workshop on Third Generation Wireless Networks*, Kluwer International Series in Engineering and Computer Science, pages 249–259, New Brunswick, NJ, Oct. 1993.

[73] P. J. Mosterman. An overview of hybrid simulation phenomena and their support by simulation packages. In *Hybrid Systems: Computation and Control*, Lectures Notes in Computer Science, pages 165–177. Springer Verlag, 1999.

[74] J. Nasreddine, L. Nuaymi, and X. Lagrange. Adaptive power control algorithm for 3G cellular CDMA networks. In *Proceedings of the IEEE Vehicular Technology Conference*, volume 2, pages 984–988, May 2004.

[75] J. G. Proakis. *Digital Communications*. McGraw-Hill, 3rd edition edition, 1995.

[76] J. G. Proakis, editor. *Wiley Encyclopedia of Telecommunications*. John Wiley & Sons, Ltd, Jan. 2003.

[77] M. Rintamäki. Power control in CDMA cellular communication systems. In Proakis [76], pages 1982–1988.

[78] M. Rintamäki. *Adaptive Power Control in CDMA Cellular Communication Systems*. PhD thesis, Helsinki University of Technology, Nov. 2005.

[79] M. Rintamäki, I. Virtej, and H. Koivo. Two-mode fast power control for WCDMA systems. In *Proceedings of the IEEE Vehicular Technology Conference*, volume 4, pages 2893–2897, May 2001.

[80] M. Rintamäki, K. Zenger, and H. Koivo. Self-tuning adaptive algorithms in the power control of WCDMA systems. In *Proceedings of the 5th Nordic Signal Processing Symposium (NORSIG)*, Norway, oct 2002.

[81] A. Robinson. *Non Standard Analysis*. North Holland, 1966.

[82] A. Sampath, P. S. Kumar, and J. M. Holtzman. On setting reverse link target SIR in a CDMA system. *IEEE Transactions on Vehicular Technology*, 47(2):929–932, 1997.

[83] J. W. Satzinger, R. B. Jackson, S. Burd, M. Simond, and M. Villeneuve. *Analyse et conception de systèmes d'information*. Les éditions Reynald Goulet, 2003.

[84] L. Schwartz. *Théorie des distributions*. Hermann, 1966.

[85] F. L. Severance. *System Modeling and Simulation: An Introduction*. John Wiley & Sons, Aug. 2001.

[86] I. Sommerville. *Software Engineering*. Addison-Wesley, 6th edition edition, 2001.

[87] V. Sridhar. Can we beat the S-curve syndrome? `http://www.financialexpress.com/`, Aug. 2005.

[88] A. Stoll. *Self-repellent random walks and polymer measures in two dimensions*. PhD thesis, Ruhr-Universität Bochum, Universitätsstraße 150, D-4630 Bochum 1, Germany, 1985.

[89] J.-Y. Thibon. Hopf algebras of symmetric functions and tensor products of symmetric group representations. *Int. J. of Alg. and Comput.*, 1(2):207–221, 1991.

[90] G. L. Turin. The characteristic function of Hermitian quadratic forms in complex normal variables. *Biometrika*, 47:199–201, 1960.

[91] S. Ulukus and R. D. Yates. Adaptive power control and MMSE interference suppression. *Wireless Networks*, 4(6):489–496, Nov. 1998.

[92] S. Ulukus and R. D. Yates. Stochastic power control for cellular radio systems. *IEEE Transactions on Communications,*, 46(6):784–798, June 1998.

[93] UMTS World. Overview of the universal mobile telecommunication system. `http://www.umtsworld.com/`, July 2002.

[94] Wireless Intelligence. GSM subscriber statistics. `http://www.gsmworld.com`, Q3 2005.

[95] Wireless Net DesignLine. How to tune UMTS network QoS to match user expectations. `http://www.wirelessnetdesignline.com/howto/`, Jan. 2006.

[96] R. D. Yates. A framework for uplink power control in cellular radio systems. *IEEE Journal on Selected Areas in Communications*, 13(7):1341–1348, Sept. 1995.

[97] J. Zander. Optimum power control in cellular radio systems. Technical report, Royal Institute of Technology (KTH), Jan. 1991.

[98] J. Zander. Performance of optimum transmitter power control in cellular radio systems. *IEEE Transactions on Vehicular Technology*, 41(1):57–62, Feb. 1992.

[99] J. Zander. Transmitter power control for co-channel interference management in cellular radio systems. In J. M. Holtzman, editor, *Proceedings of 4th WINLAB Workshop on Third Generation Wireless Networks*, Kluwer International Series in Engineering and Computer Science, pages 241–247, New Brunswick, NJ, Oct. 1993.

[100] J. Zander and S.-L. Kim. *Radio Resource Management for Wireless Networks*. Number 2 in Artech House Mobile Communications Series. Artech House Publishers, Mar. 2001.

[101] J. Zaytoon, editor. *Systèmes dynamiques hybrides*. Hermès, 2001.

[102] B. P. Zeigler, H. Praehofer, and K. T. Gon. *Theory of Modeling and Simulation — Integrating Discrete Event and Continuous Complex Dynamic Systems*. Academic Press, 2000.

List of Figures

List of Tables

List of Algorithms