Multiscale Analysis of Complex Time Series: Integration of Chaos and Random Fractal Theory, and Beyond
by Jianbo Gao, Yinhe Cao, Wen-wen Tung and Jing Hu

CHAPTER 12

BIFURCATION THEORY

In Chapter 2, we briefly discussed elements of chaotic dynamics. Since in reality simple and complex motions coexist, it is important to consider the transitions from a simple motion to a complex one and vice versa. Insights into this issue can be gained by studying bifurcations and routes to chaos. The latter was briefly discussed in Chapter 2. In this chapter, we study bifurcation, which is the change in the qualitative character of a solution as a control parameter is varied. While the theory might seem simple, it can be extremely effective in solving difficult engineering problems. As an example, we shall discuss how to find the exact error threshold values for noisy NAND and majority gates to function reliably. This is a topic of considerable current interest, since in emerging nanotechnologies, reliable computation will have to be carried out with unreliable components as integral parts of computing systems.

12.1 BIFURCATIONS FROM A STEADY SOLUTION IN CONTINUOUS TIME SYSTEMS

12.1.1 General considerations

Assume that our dynamical system is described by a first-order ordinary differential equation (ODE),

$$\dot{y} = f(y, r),$$

where r is a control parameter. A fixed point $y = y^*$ is a solution to

$$f(y, r) = 0.$$

That is, the velocity at $y = y^*$ is zero. Typically, y^* is a function of r. To find the stability of $y = y^*$, we can consider an orbit $y(t) = y^* + u(t)$, where r is fixed and $u(t)$ is a small perturbation. We have

$$\frac{d}{dt}(y^* + u(t)) = f(y^* + u(t), r) \approx \left.\frac{\partial f}{\partial y}\right|_{(y^*, r)} u.$$

Therefore, $y = y^*$ is stable when $\left.\frac{\partial f}{\partial y}\right|_{(y^*,r)} < 0$, unstable when $\left.\frac{\partial f}{\partial y}\right|_{(y^*,r)} > 0$. At the bifurcation point $r = r_c$, $\left.\frac{\partial f}{\partial y}\right|_{(y^*,r_c)} = 0$.

We now consider classification of bifurcations at $r = r_c$. Using Taylor's expansion of $f(y, r)$ at (y^*, r_c), we have

$$\begin{aligned}\dot{y} = f(y, r) = f(y^*, r_c) + (y - y^*)\left.\frac{\partial f}{\partial y}\right|_{(y^*, r_c)} \\ +(r - r_c)\left.\frac{\partial f}{\partial y}\right|_{(y^*, r_c)} + \frac{1}{2}(y - y_*)^2\left.\frac{\partial^2 f}{\partial y^2}\right|_{(y^*, r_c)} + \cdots .\end{aligned} \tag{12.1}$$

The first term, $f(y^*, r_c)$ is zero, since at $r = r_c$, $y = y^*$ is a fixed point solution. Now rewrite terms such as $y - y^*$ as u, $r - r_c$ as a, $\dot{y}$ as $d(y - y^*)/dt = \dot{u}$, etc. We have $\dot{u}$ (which is equal to $\dot{y}$) as a summation of various terms of powers of u, a, etc.

To fix the idea, let us consider

$$\dot{u} = au + bu^3 - cu^5, \tag{12.2}$$

where $b, c > 0$. Usually, we would like the coefficients before powers of x to be 1. For this example, we can let $x = u/U$, $\tau = t/T$, choose U, T, and r, and obtain

$$\frac{dx}{d\tau} = rx + x^3 - x^5. \tag{12.3}$$

The various normal forms of bifurcations discussed below amount to a classification of this Taylor series expansion, with higher-order terms ignored.

12.1.2 Saddle-node bifurcation

The saddle-node bifurcation is the basic mechanism by which fixed points are created and destroyed. The prototypical example of a saddle-node bifurcation is given by the first-order system

$$\dot{x} = r + x^2, \tag{12.4}$$

where r is a parameter taking on real values. When $r < 0$, there are two fixed points: $-\sqrt{-r}$, which is stable, and $\sqrt{-r}$, which is unstable. The bifurcation point is at $r = 0$.

A bifurcation diagram is a plot of the fixed point solutions vs. the parameter. The bifurcation diagram for the saddle-node bifurcation is shown in Fig. 12.1(a). The solid branch, which is stable, means that if Eq. (12.4) is numerically solved, then eventually the solution will settle on the curve for the corresponding r. The dashed branch, which is unstable, cannot be observed by numerical simulations if the unstable solution is not chosen as the initial condition.

12.1.3 Transcritical bifurcation

There are situations where multiple fixed point solutions coexist. At least one of them must be stable for all values of the controlling parameter, yet the stability of a specific fixed point solution may change with the value of the parameter. Such situations are characterized by a transcritical bifurcation. The normal form is

$$\dot{x} = rx - x^2, \tag{12.5}$$

where r is a parameter taking on real values. In this situation, there are two fixed point solutions, $x = 0$ and $x = r$. When $r < 0$, $x = 0$ is stable but $x = r$ is not; when $r > 0$, $x = 0$ becomes unstable while $x = r$ becomes stable. The bifurcation diagram is shown in Fig. 12.1(b). Again, $r = 0$ is the bifurcation point.

12.1.4 Pitchfork bifurcation

Pitchfork bifurcation is common in physical problems that have symmetry. In such cases, fixed points tend to appear or disappear in symmetrical pairs. There are two different types of pitchfork bifurcation. Here $r = 0$ is also the bifurcation point. The simpler type is called supercritical. Its normal form is

$$\dot{x} = rx - x^3, \tag{12.6}$$

where r is a real parameter. When $r \leq 0$, there is only one fixed point solution, $x = 0$, which is stable. When $r > 0$, $x = 0$ becomes unstable. Two new stable fixed point solutions emerge, $x = \pm\sqrt{r}$. See Fig 12.1(c). In numerical simulations, depending on which initial conditions are used, one of the stable fixed point solutions will be observed.

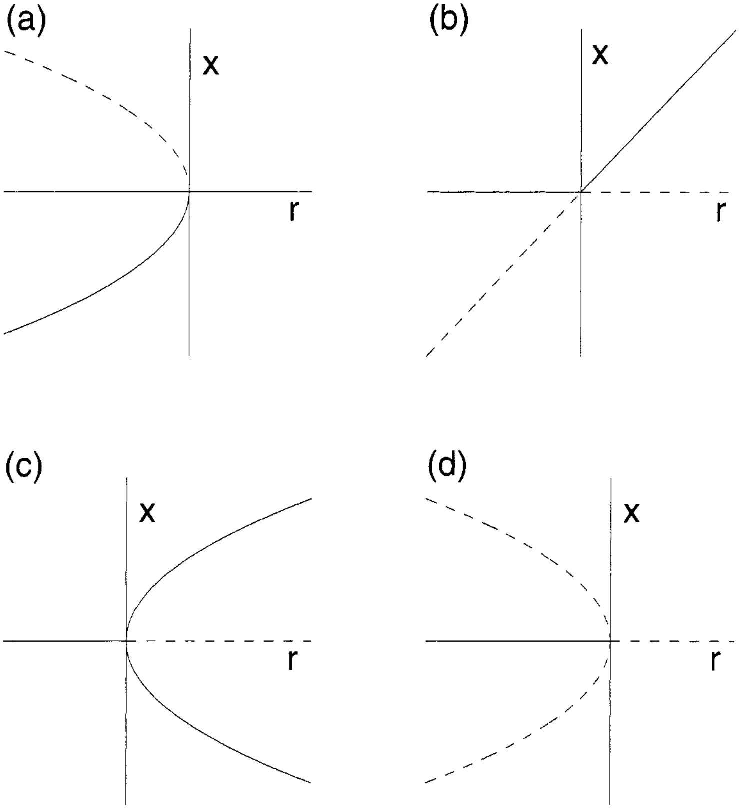

Figure 12.1. Normal forms for stationary bifurcations. Solid lines are stable solutions; dashed lines are unstable solutions. (a) Saddle-node; (b) transcritical; (c) supercritical pitchfork; (d) subcritical pitchfork.

A slightly more complicated situation is called subcritical. Its normal form is

$$\dot{x} = rx + x^3, \tag{12.7}$$

where r is a real parameter. When $r > 0$, there is only one unstable fixed point solution, $x = 0$. When $r < 0$, three fixed point solutions coexist; $x = 0$ is stable, while $x = \pm\sqrt{-r}$ are unstable. See Fig 12.1(d).

When $r > 0$, a typical solution to Eq. (12.7) diverges to infinity very rapidly. To stabilize it, let us add a term x^5 to Eq. (12.7),

$$\dot{x} = rx + x^3 - x^5. \tag{12.8}$$

The system is still symmetric under $x \to -x$. For $r < r_s = -1/4$, there is only one stable fixed point solution, $x = 0$. When $r_s \leq r < 0$, there are five fixed point solutions; three are stable, given by $x = 0$ and $x^2 = [1 + \sqrt{1+4r}]/2$, and two are unstable, given by $x^2 = [1 - \sqrt{1+4r}]/2$. When $r \geq 0$, there are three fixed point

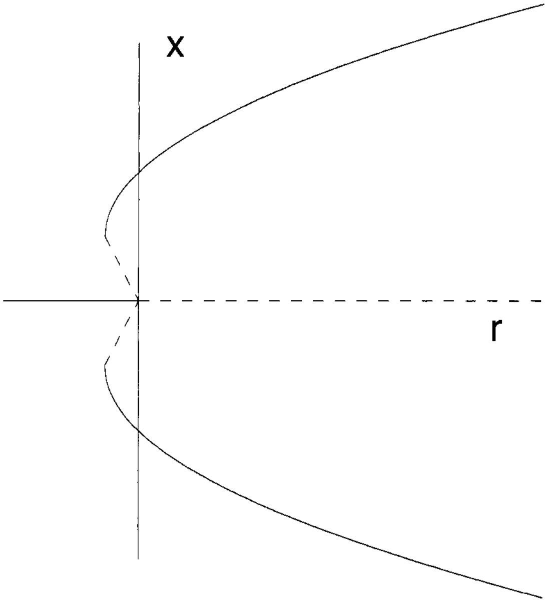

Figure 12.2. Subcritical pitchfork bifurcation with phenomenological quintic stabilizing terms.

solutions: $x = 0$, which is unstable, and $x^2 = [1 + \sqrt{1 + 4r}]/2$, which are stable. These solutions are summarized in Fig. 12.2.

Figure 12.2 provides a mechanism for hysteresis as the parameter r is varied. Suppose we start from the solution $x = 0$ and a negative r. When r is increased but remains negative, the solution will still be $x = 0$. When r passes zero, however, any tiny disturbance will cause the solution to switch to one of the stable fixed points. Let us assume that the upper branch is reached. Then the system will stay at that branch so long as r is kept equal to or larger than $r_s = -1/4$. When r is below r_s, the unique stable fixed point solution of $x = 0$ will again be reached.

12.2 BIFURCATIONS FROM A STEADY SOLUTION IN DISCRETE MAPS

We now consider a discrete map,

$$x_{n+1} = f(x_n, r).$$

Denote the fixed point solution corresponding to $r = r_c$ as x^*. It satisfies

$$x^* = f(x^*, r_c). \tag{12.9}$$

To determine the stability of x^*, we examine a trajectory, $x_0, x_1, \cdots, x_n$, where $x_n = \delta x_n + x^*$. We have

$$x_{n+1} = \delta x_{n+1} + x^* = f(\delta x_n + x^*, r_c) \approx f(x^*, r_c) + \left.\frac{\partial f}{\partial x}\right|_{x=x^*, r=r_c} \delta x_n.$$

Therefore,

$$\delta x_{n+1} \approx \left.\frac{\partial f}{\partial x}\right|_{x=x^*, r=r_c} \delta x_n.$$

For simplicity, denote

$$a = \left.\frac{\partial f}{\partial x}\right|_{x=x^*, r=r_c}.$$

When $|a| < 1$, δx_{n+1} decays to zero exponentially fast; hence, x^* is stable. Otherwise, it is unstable. The bifurcation point is given by $a = \pm 1$: when $a = 1$, the bifurcation is called pitchfork; when $a = -1$, it is called period-2.

To be concrete, we consider the logistic map, $x_{n+1} = f(x_n) = rx_n(1 - x_n)$, where $0 \le x_n \le 1,\ 0 \le r \le 4$. Using Eq. (12.9), we easily find

$$x^* = 0 \ \text{ or } \ 1 - \frac{1}{r}.$$

It is straightforward to check that $x^* = 0$ is stable when $0 \le r < 1$, while $x^* = 1 - \frac{1}{r}$ is stable when $1 < r < 3$. $r = 1$ and $r = 3$ are two bifurcation points: at $r = 1$, the bifurcation is pitchfork; at $r = 3$, the bifurcation is period-2. Beyond $r = 3$, the motion eventually becomes chaotic through the period-doubling route (see Fig. 2.4).

12.3 BIFURCATIONS IN HIGH-DIMENSIONAL SPACE

In general, a dynamical system may be described by a high-dimensional ODE or map and possibly by multiple control parameters. In the former case, the analysis would still be simple – interpreting x as a vector in the discussions of Secs. 12.1 and 12.2 would suffice. The situation is more complicated when there are multiple parameters. One good strategy is to first find fixed point solutions and then carry out numerical simulations of the dynamical system by systematically varying the control parameters.

12.4 BIFURCATIONS AND FUNDAMENTAL ERROR BOUNDS FOR FAULT-TOLERANT COMPUTATIONS

In the emerging nanotechnologies, faulty components may be an integral part of a system. For the system to be reliable, the error of the building blocks — the noisy gates — has to be smaller than a threshold. Therefore, finding exact error thresholds for noisy gates is one of the most challenging problems in fault-tolerant computations. Since von Neumann's foundational work in 1956, this issue has attracted a lot of attention, and many important results have been obtained using information theoretic approaches. Here we show that this difficult problem can be neatly solved by employing bifurcation theory.

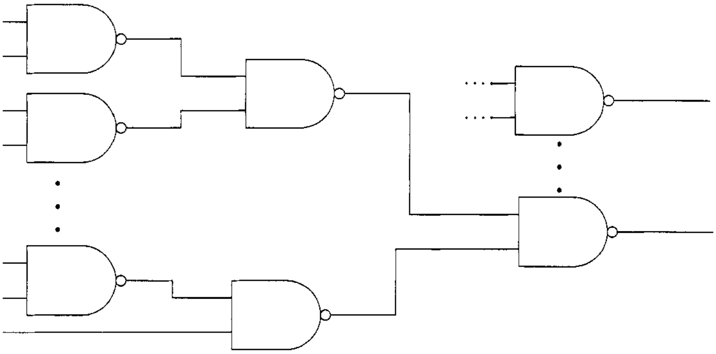

Figure 12.3. Schematic of a circuit built out of two-input NAND gates.

12.4.1 Error threshold values for arbitrary K-input NAND gates

In this subsection, we first derive the error threshold value for a two-input NAND gate and then extend the analysis to an arbitrary K-input NAND gate.

12.4.1.1 Two-input NAND gates The basic computation in the von Neumann multiplexing scheme involves the circuit schematic shown in Fig. 12.3, where it is assumed that there are no feedback loops and the output of each gate is connected to an input of only one other gate in the circuit. For the basic gate error, we adopt the simple von Neumann model, which assumes that the gate flips the output with a probability $\epsilon \leq 1/2$, while the input and output lines function reliably. For a single NAND gate, let us denote the probabilities of the two inputs being 1 by X and Y. Since the circuits considered have neither closed loops nor fan-out, the two inputs can be treated as independent. Thus the probability Z of the output being a 1 is

$$Z = (1-\epsilon)(1-XY) + \epsilon XY = (1-\epsilon) + (2\epsilon - 1)XY. \qquad (12.10)$$

We observe a few interesting properties of Z: (1) When $\epsilon = 0$ and X, Y, Z take on values of either 1 or 0, Eq. (12.10) reduces to the standard definition of an error-free NAND gate, $Z = 1 - XY$. (2) $Z_{\text{max}} = 1 - \epsilon$, $Z_{\text{min}} = \epsilon$. (3) When X (or Y) is fixed, Z linearly decreases with Y (or X). (4) Z decreases most rapidly along $X = Y$, and hence $X = Y$ constitutes the worst-case scenario.

In von Neumann's multiplexing scheme, duplicates of each output randomly become the inputs to another NAND gate. This motivates us to first consider the case $X = Y$. Furthermore, we label a sequence of NAND gates by index i, $i = 1, 2, \cdots, n, \cdots$, where the output of gate i becomes the input to gate $i+1$. Noticing that a NAND gate is a universal gate, in an actual computation *i may also* be considered equivalent to the ith step of the computation. Equation (12.10) thus reduces to a simple nonlinear map,

$$X_{n+1} = (1 - \epsilon) + (2\epsilon - 1)X_n^2. \tag{12.11}$$

Bifurcation analysis of Eq. (12.11) reveals that a period-doubling bifurcation occurs at

$$\epsilon_* = (3 - \sqrt{7})/4 = 0.08856. \tag{12.12}$$

See Fig. 12.4. When $\epsilon_* < \epsilon \le 1/2$, the system has a stable fixed point solution,

$$x_0 = \frac{-1 + \sqrt{4(1-\epsilon)(1-2\epsilon) + 1}}{2(1 - 2\epsilon)}. \tag{12.13}$$

Exactly at $\epsilon = \epsilon_*$, a period-doubling bifurcation occurs, since $f'(x_0) = -1$. From Eq. (12.13), one can readily find ϵ_*, as given by Eq. (12.12). When $0 \le \epsilon < \epsilon_*$, x_0 loses stability, and the motion is periodic with period 2. These two periodic points have been labeled x_+ and x_- in Fig. 12.4. For convenience in our later discussions, we explicitly write down the following three equations:

$$x_0 = (1 - \epsilon) + (2\epsilon - 1)x_0^2, \tag{12.14}$$

$$x_+ = (1 - \epsilon) + (2\epsilon - 1)x_-^2, \tag{12.15}$$

$$x_- = (1 - \epsilon) + (2\epsilon - 1)x_+^2. \tag{12.16}$$

Noting that x_0 is also a solution to Eqs. (12.15) and (12.16), after factoring out the terms involving x_0, one obtains the two periodic points on the limit cycle,

$$x_\pm = \frac{1 \pm \sqrt{4(1-\epsilon)(1-2\epsilon) - 3}}{2(1 - 2\epsilon)}. \tag{12.17}$$

For a NAND gate to function reliably, two identical inputs of 1 or 0 should output a 0 or 1, respectively. We thus see that $0 \le \epsilon < \epsilon_*$ is the parameter interval where the NAND gate can function. When $\epsilon > \epsilon_*$, the output x_0 can be interpreted as either 1 or 0 and hence, is what von Neumann called a state of irrelevance. Equations (12.12), (12.13), and (12.17) are identical, with expressions derived by Pippenger [348] using a much more complicated approach. In what follows, it will become clear that bifurcation analysis provides additional insights and generalizes to K-input NAND gates.

12.4.1.2 Arbitrary K-input NAND gate First, we note that with current technology, an arbitrary K-input NAND gate can be readily built, independent of a two-input NAND gate. A circuit built of arbitrary K-input NAND gates may be more reliable and less redundant than a circuit built of two-input NAND gates.

For a K-input NAND gate, we can generalize Eqs. (12.10) and (12.11) to come up with the following two basic equations:

$$Z = (1 - \epsilon) + (2\epsilon - 1)Y_1 Y_2 \cdots Y_K, \tag{12.18}$$

$$\begin{aligned} X_{n+1} &= (1 - \epsilon) + (2\epsilon - 1)X_n^K \\ &= 1 - X_n^K + \epsilon(2X_n^K - 1) = f(X_n). \end{aligned} \tag{12.19}$$

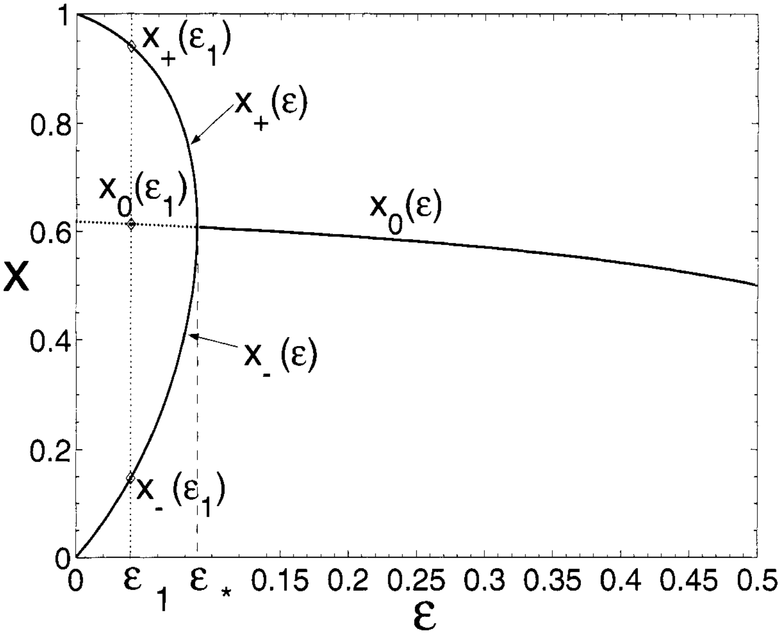

Figure 12.4. The bifurcation diagram for Eq. (12.11). ϵ is the individual gate error probability and X is the probability of a gate output being 1. $x_0(\epsilon)$, given by Eq. (12.13), is unstable when $0 < \epsilon < \epsilon_*$ and stable when $\epsilon_* < \epsilon < 1/2$. In the interval $0 < \epsilon < \epsilon_*$, $x_\pm(\epsilon)$ are given by Eq. (12.17) and form the upper and lower branches of the bifurcation.

The bifurcation diagrams for Eq. (12.19) with $K = 3, 6, 9$, and 12 are shown in Figs. 12.5(a–d). We observe that the fixed point x_0 monotonically increases with K and that the bifurcation involved is always period-doubling. The latter makes perfect sense, since for a sequence of NAND gates to function, the states of the outputs must flip. Such behavior can only be described by a period-2 motion.

Let us now find the bifurcation point ϵ_*. Let x_0 again be a fixed point of the map. We have

$$x_0 = 1 - x_0^K + \epsilon(2x_0^K - 1). \tag{12.20}$$

It is stable when $\epsilon > \epsilon_*$. When K is large, the explicit expression for x_0 may be hard to obtain. Let us not bother to do so. Instead, let us examine when x_0 may lose stability when ϵ is varied. This can be found by requiring the derivative of $f(x)$ evaluated at x_0 to be -1, $f'(x_0) = -1$. Hence

$$(2\epsilon_* - 1)Kx_0^{K-1} = -1. \tag{12.21}$$

Combining Eqs. (12.20) and (12.21), we have

$$\left(1 + \frac{1}{K}\right)\left[\frac{1}{K(1 - 2\epsilon_*)}\right]^{\frac{1}{K-1}} = 1 - \epsilon_* \,. \tag{12.22}$$

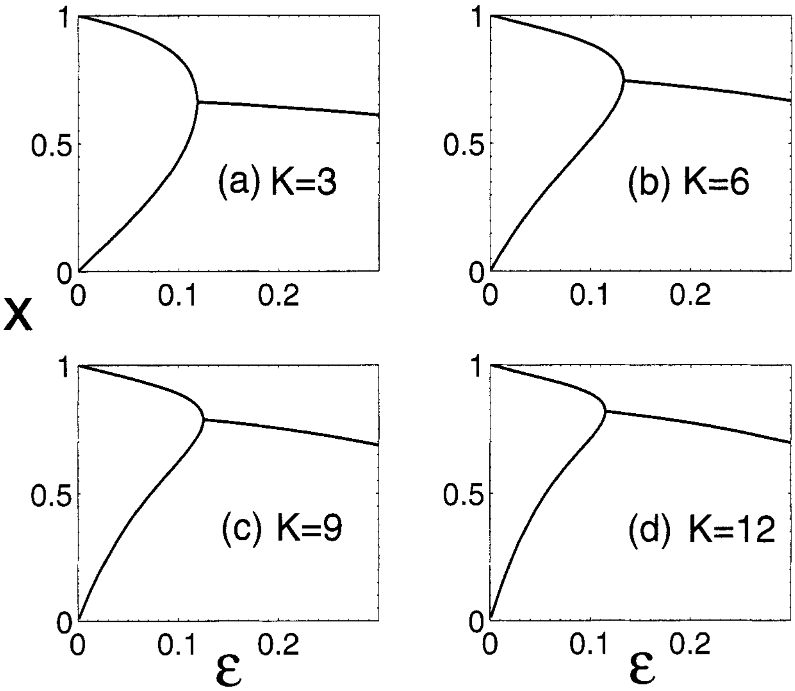

Figure 12.5. Bifurcation diagrams for Eq. (12.19) with $K = 3, 6, 9$, and 12.

For example, let us solve Eq. (12.22) for $K = 2$ and 3. We obtain Eq. (12.12) for a two-input NAND gate and

$$\epsilon_* = \frac{5}{6} - \frac{1}{6(33 - 8\sqrt{17})^{\frac{1}{3}}} - \frac{1}{6}(33 - 8\sqrt{17})^{\frac{1}{3}} \approx 0.1186$$

for a three-input NAND gate. For $K \geq 4$, analytical expressions for ϵ_* are messy or hard to obtain; however, we can still obtain the threshold values by numerically solving Eq. (12.22). Figure 12.6 shows the variation of ϵ_* with K for $K \leq 20$. We observe that the error threshold value ϵ_* of 0.1186 for $K = 3$ is almost 34% larger than that for $K = 2$. We also note that ϵ_* assumes a maximum value of 0.1330 at $K = 5$, which is more than 50% larger than that for $K = 2$. We thus see that in situations where a two-input NAND gate is too noisy to compute reliably, it may be advantageous to use gates with more than two (but fewer than six) inputs.

12.4.2 Noisy majority gate

We now consider noisy majority gates with $2K + 1$ inputs. Let ϵ be the probability that the noisy gate makes a von Neumann error (flipping the output logic value). Assume that the inputs are independent and that their probabilities of being 1 are all

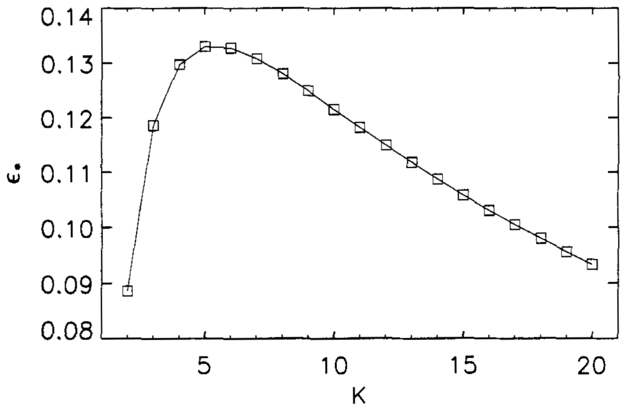

Figure 12.6. Variation of ϵ_* vs. K.

the same, which we shall denote by x. Let θ represent the probability of the output being 1 when the gate is fault-free. Notice that the majority gate outputs 1 when the number of excited inputs is larger than the number of unexcited inputs. Otherwise, the gate outputs 0. Thus θ is given by

$$\theta = \sum_{i=K+1}^{2K+1} \binom{2K+1}{i} x^i (1-x)^{2K+1-i}. \tag{12.23}$$

It follows that the probability Z that the output is 1 is

$$\begin{aligned} Z &= \epsilon(1-\theta) + (1-\epsilon)\theta \\ &= \epsilon + (1-2\epsilon)\left[\sum_{i=K+1}^{2K+1} \binom{2K+1}{i} x^i (1-x)^{2K+1-i}\right]. \end{aligned} \tag{12.24}$$

Often there are situations (such as in the case of a network of gates) where one needs to consider a sequence of gates, where the output from one gate becomes the input to the next gate. This motivates us to denote the probability of the output from gate n being 1 as x_n, while the probability of the output from gate $n+1$ being 1 is x_{n+1}. We thus obtain the following map:

$$x_{n+1} = \epsilon + (1-2\epsilon)\left[\sum_{i=K+1}^{2K+1} \binom{2K+1}{i} \cdot x_n^i \cdot (1-x_n)^{2K+1-i}\right], \tag{12.25}$$

where ϵ is identified as a bifurcation parameter. Alternatively, we may interpret x_n as the probability of the input to gate n being 1 and x_{n+1} as the probability of the input to gate $n+1$ being 1.

To better understand the dynamic behavior of Eq. (12.25), let us recall a few facts on how faulty majority gates function. When the gate is faulty, there are two distinct cases: (1) when the error is small enough, we anticipate that we can assign unambiguously a logic 1 or 0 to the output of the gate, depending on whether the number of excited inputs is larger than that of the unexcited inputs; (2) when the error is very large, we will no longer be able to decide what logic value of the output of the gate is in. This was referred to as a state of irrelevance by von Neumann. This simple discussion makes us anticipate that when the gate is reliable enough, the map has two stable fixed point solutions: one larger than $1/2$, the other smaller than $1/2$, corresponding to logic 1 and 0, respectively. Depending on whether the initial condition is larger or smaller than $1/2$, the fixed point with value larger or smaller than $1/2$ will then be approached in a circuit with sufficient depth. Mathematically, the gate being reliable enough can be described by $\epsilon < \epsilon_*$, where ϵ_* is a certain threshold value. Our purpose is to find the exact error threshold values ϵ_* and the two stable fixed point solutions when $\epsilon < \epsilon_*$ for an arbitrary number of inputs $2K+1$. To simplify the subsequent analysis, let us make a simple transformation:

$$x = \frac{1-y}{2}.$$

Equation (12.25) then becomes

$$\begin{aligned} y_{n+1} &= f(y_n, \epsilon) \\ &= (1-2\epsilon)\left\{1 - 2^{-2K} \sum_{i=K+1}^{2K+1} \binom{2K+1}{i} \cdot (1-y_n)^i \cdot (1+y_n)^{2K+1-i}\right\}. \end{aligned} \tag{12.26}$$

We have the following simple but important lemma.

> **Lemma 1** $y = 0$ is a fixed point solution to Eq. (12.26) for $0 \le \epsilon \le 1/2$. Equivalently, $x = 1/2$ is a fixed point solution to Eq. (12.25).

Proof: This amounts to proving that

$$1 - 2^{-2K} \sum_{i=K+1}^{2K+1} \binom{2K+1}{i} = 0. \tag{12.27}$$

This is indeed so, since, by symmetry,

$$\left(\frac{1}{2}+\frac{1}{2}\right)^{2K+1} = 1 = 2 \cdot \left\{\sum_{i=K+1}^{2K+1} \binom{2K+1}{i} \cdot \left(\frac{1}{2}\right)^i \cdot \left(\frac{1}{2}\right)^{2K+1-i}\right\}.$$

Lemma 1 says that when $x = 1/2$ or $y = 0$, a meaningful logic 1 or 0 cannot be assigned to the output of the gate. The problem is thus to determine when this state dominates and when it cannot be reached. These two questions are answered by the following theorem.

Theorem 1 Equation (12.26) undergoes a pitchfork bifurcation at

$$\epsilon_* = \frac{1}{2} - \frac{2^{2K-1} \cdot (K!)^2}{(2K+1)!}. \tag{12.28}$$

When $\epsilon \geq \epsilon_*$, the fixed point solution $y = 0$ is stable, and the gate is in a state of irrelevance. When $0 < \epsilon < \epsilon_*$, however, $y = 0$ becomes unstable, and two new stable fixed point solutions are created, corresponding to the scenario in which the gate functions reliably.

Proof: Our discussion so far has pointed to the possibility that the bifurcation pertinent to Eq. (12.26) is a pitchfork bifurcation. Mathematically, a pitchfork bifurcation means that the derivative of the right-hand side of Eq. (12.26) evaluated at ϵ_* is 1, $\partial f(y,\epsilon)/\partial y|_{y=0,\epsilon=\epsilon_*} = 1$. We thus have a simple equation:

$$(1-2\epsilon_*)\left(-2^{-2K}\right)\left\{\sum_{i=K+1}^{2K+1} (2K+1-2i)\binom{2K+1}{i}\right\} = 1. \tag{12.29}$$

To simplify Eq. (12.29), we first note, using Eq. (12.27), that

$$\left\{\sum_{i=K+1}^{2K+1} (2K+1)\frac{(2K+1)!}{i! \cdot (2K+1-i)!}\right\} = (2K+1)\, 2^{2K}.$$

The term

$$\left\{\sum_{i=K+1}^{2K+1} (-2i)\frac{(2K+1)!}{i! \cdot (2K+1-i)!}\right\},$$

on the other hand, can be rewritten as

$$\left\{\sum_{j=i-1=K}^{2K} (-2)\frac{(2K+1)\cdot(2K)!}{j! \cdot (2K-j)!}\right\}.$$

By expanding

$$\left(\frac{1}{2}+\frac{1}{2}\right)^{2K}$$

and again using symmetry, one finds that

$$\left\{\sum_{j=K}^{2K} (-2)\frac{(2K+1)\cdot(2K)!}{j! \cdot (2K-j)!}\right\} = -(2K+1)\left(2^{2K} + \frac{(2K)!}{K! \cdot K!}\right).$$

Equation (12.28) then follows. Next, let us prove our last statement: when $0 < \epsilon < \epsilon_*$, the gate functions reliably. For this purpose, we define the logic value of the output of the gate to be 1 if $1/2 < x \leq 1$, and 0 if $0 \leq x < 1/2$. Equivalently, this can be expressed as $-1 \leq y < 0$ and $0 < y \leq 1$. To prove the statement, it is sufficient to prove that when $-1 \leq y_0 < 0$ or $0 < y_0 \leq 1$, then $-1 \leq y_1 < 0$ or $0 < y_1 \leq 1$, where y_0 and y_1 are related by Eq. (12.26). For concreteness, let us focus on the case of $-1 \leq y_0 < 0$. We need to prove

$$\sum_{i=K+1}^{2K+1} \binom{2K+1}{i} \cdot (1-y_0)^i \cdot (1+y_0)^{2K+1-i} > 2^{2K}.$$

This is indeed the case, noting that

$$(1 - y_0 + 1 + y_0)^{2K+1} = 2^{2K+1}$$

and

$$(1-y_0)^i \cdot (1+y_0)^{2K+1-i} > (1-y_0)^{2K+1-i} \cdot (1+y_0)^i \qquad (i \geq K+1).$$

Note that this part of the proof can be made more formal by defining the logic value of the output to be 1 if $1/2 + \eta \leq x \leq 1$ and 0 if $0 \leq x \leq 1/2 - \eta$, where η is an arbitrarily small positive number.

It is interesting to examine whether the gate can function reliably when $\epsilon = \epsilon_*$. In a circuit with sufficient depth, where n can be arbitrarily large, y_n can be arbitrarily close to 0. Hence, for any prescribed η, sooner or later, y_n will be within $(-\eta, \eta)$. This means $\epsilon = \epsilon_*$ has to be excluded.

To make the above discussions more informative, let us numerically investigate the bifurcations of Eq. (12.25) for a few fixed $2K + 1$. A few examples are shown in Fig. 12.7 for $2K + 1 = 3, 5, 15, 101$. We observe that the bifurcation involved is indeed a pitchfork bifurcation and that ϵ_* increases with the number of inputs $2K + 1$.

12.4.3 Analysis of von Neumann's multiplexing system

In emerging nanotechnologies, reliable computation will have to be carried out with unreliable components being integral parts of computing systems. One promising scheme for designing these systems using universal NAND gates is von Neumann's multiplexing technique. A multiplexed system consists of a chain of stages (in space or time) each of which consists of NAND gates whose outputs are duplicated and randomly connected to the inputs of the gates of the next stage in the chain. The NAND multiplexing unit schematically shown in Fig. 12.8 is comprised of a randomizing unit R and N copies of NAND gates each failing (flipping the output bit value) with gate error probability ϵ. It takes two bundles of N wires as inputs and generates a bundle of N wires as the output. Through a random permutation by the "randomizing unit," the inputs in one bundle are randomly paired with those from the other bundle to form input pairs to the duplicated NAND gates. In systems based on this construction, each signal is carried on a bundle of N wires instead of a single wire and every logic computation is done by N duplicated gates simultaneously. The

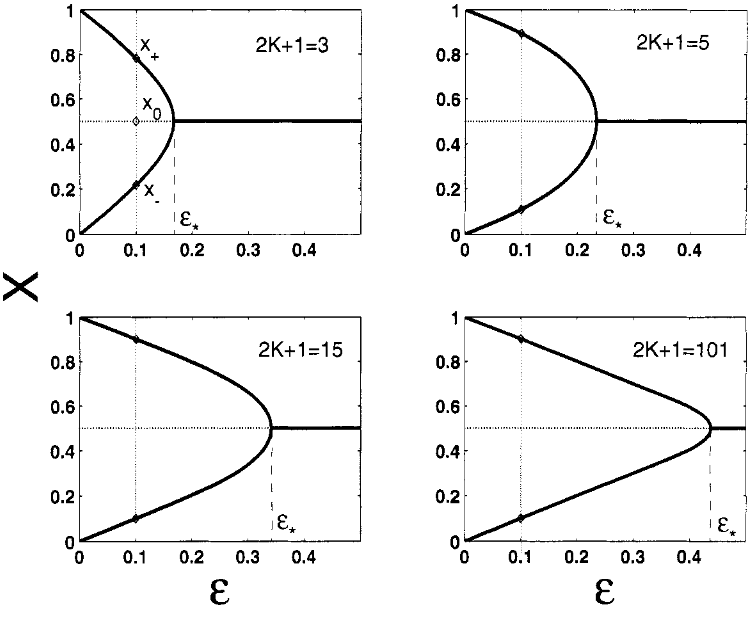

Figure 12.7. Bifurcation diagrams for $2K + 1 = 3,\ 5,\ 15, 101$.

logic state 1 or 0 is decided for a bundle when its excitation level, i.e., the fraction of excited wires, is above or below a preset threshold. In a multiplexed system, each computation node is comprised of three such multiplexing units connected in series, as illustrated in Fig. 12.9. The executive unit carries out logic computation, and the restorative unit (comprised of two NAND multiplexing units) restores the excitation level of the output bundle of the executive unit to its nominal level.

Recently, a number of research groups have used a Markov chain model to analyze the reliability of multiplexed systems. We shall show here that the bifurcation approach can effectively analyze the behavior of the multiplexing system, while the approach based on the Markov chain model is at best a very crude approximation.

12.4.3.1 Probability distributions for the excitation level of a multiplexing unit First, we note that due to the randomization operation in the multiplexing system, starting from the second unit of the multiplexing system, it is not meaningful to talk about whether the state of a wire is either in state 1 or 0. It is only meaningful to talk about the state of wires probabilistically: i.e., with probability Z being excited and probability $1 - Z$ unexcited. Let us now consider one NAND multiplexing unit depicted in Fig. 12.8. Denote the number of excited wires in a bundle of size N by a random variable L and call L/N the bundle's excitation level. Since each wire in the bundle has a probability of Z to be excited, obviously L follows a binomial

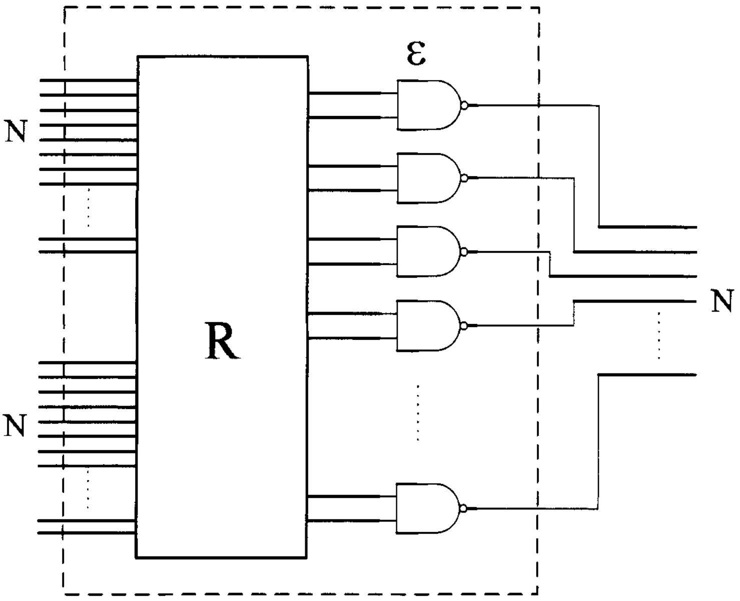

Figure 12.8. A NAND multiplexing unit.

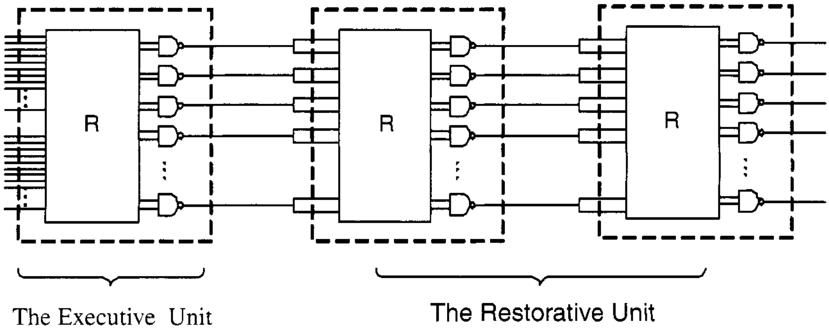

Figure 12.9. Multistage NAND multiplexing system. Each computation node in the system consists of three NAND multiplexing units shown in Fig. 12.8. The executive unit does logic computation and two NAND multiplexing units act as a restorative unit.

distribution with parameters N and Z:

$$P(l) = P(L = l) = \binom{N}{l} Z^l (1 - Z)^{N-l}. \tag{12.30}$$

The mean and variance of L are NZ and $NZ(1 - Z)$. The *excitation level* L/N is a normalized version of L with mean value Z. Von Neumann suggests that

Eq. (12.30) may be approximated by a normal distribution, $f(l)$:

$$f(l) = \frac{1}{\sqrt{2\pi}\sqrt{NZ(1-Z)}} e^{\frac{(l-NZ)^2}{2NZ(1-Z)}}. \tag{12.31}$$

Mathematically, by the central limit theorem, such an approximation is justified when Z is fixed and the bundle size N is very large. However, when building nanocomputers, we are concerned with small system error probabilities (e.g., on the order of 10^{-10} or smaller) and a small economic redundancy factor (say, $N < 1000$); hence, Gaussian approximation is not very useful. This point will be made clearer shortly. Below we show that the parameter Z can be easily found; hence, Eq. (12.30) can be readily computed without any approximations.

In order to evaluate Eq. (12.30), one only needs to find the parameter Z. To understand how Z can be identified, let us consider two consecutive multiplexing units, the ith and $(i+1)$th units. It is then clear that

$$Z = X_i \text{ for the } i\text{th stage and } Z = X_{i+1} \text{ for the } (i+1)\text{th stage},$$

respectively, where X_i and X_{i+1} are related by Eq. (12.11) or Eq. (12.19), depending on whether the building blocks are two-input NAND gates or arbitrary K-input NAND gates. When N is very large, Z can be arbitrarily close to either x_+ or x_- (see Eq. (12.17) and Fig. 12.4). Let us denote the distributions expressed by Eq. (12.30) by

$$P(l) = \vec{\pi}_+ \quad \text{when} \quad Z = x_+$$

and by

$$P(l) = \vec{\pi}_- \quad \text{when} \quad Z = x_-.$$

Figure 12.10 shows three examples of $\vec{\pi}_+$ and $\vec{\pi}_-$ for three different gate error probabilities ϵ and a fixed value of $N = 100$. We note that when transients in a multiplexing system die out, the distribution for the excitation level of the output bundle of a multiplexing unit is either $\vec{\pi}_+$ or $\vec{\pi}_-$, while the distribution for the excitation level of the output of the multiplexing unit following the previous one is either $\vec{\pi}_-$ or $\vec{\pi}_+$. In other words, the distributions oscillate between $\vec{\pi}_+$ and $\vec{\pi}_-$ along the multiplexing stages. When the gate error probability ϵ exceeds the threshold value, however, x_+ and x_- become x_0, and we only have one stationary distribution, which we shall denote by $\vec{\pi}_0$, obtained by substituting Z in Eq. (12.30) by x_0. When this is the case, the multiplexing system no longer functions.

In practice, the number of cascaded multiplexing units (or logic depth) may not be very large. Then, how relevant are $\vec{\pi}_+$ and $\vec{\pi}_-$ to reality? The answer is easiest to obtain when system reliability is considered, as shown below.

12.4.3.2 Reliability of multiplexing systems Characterizations of error and reliability are natural extensions of the notion of probabilistic computation and are particularly important for the design of fault-tolerant nanoelectronic systems

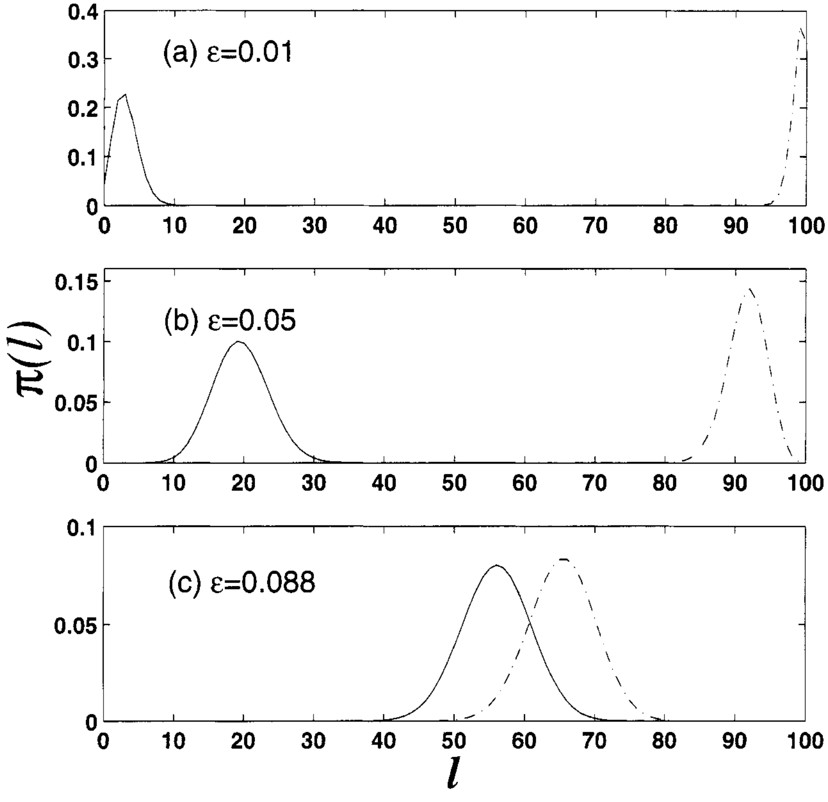

Figure 12.10. Distributions $\vec{\pi}_-$ (solid curve) and $\vec{\pi}_+$ (dash dot curve) for three different gate error probabilities for von Neumann's multiplexing system with $N = 100$. They amount to the distributions for two consecutive stages of the multiplexing system when the stage number is large.

with a large number of defective components. As pointed out, when $\epsilon > \epsilon_*$, the multiplexing system has a single stationary distribution and the logic states 1 and 0 of the system become indistinguishable. Thus only the case $\epsilon < \epsilon_*$ will be considered in the analysis that follows.

Let us now give our definitions of error and reliability. We start by noting that computation by multistage NAND multiplexing systems is essentially a binary communication problem. Errors occur at each stage as the signal propagates through the noisy channel — a cascade of NAND multiplexing units. Given the gate error probability ϵ, bundle size N, for each input instance and a particular stage n, we need to make a decision on the output logic value based on L_n, *the observed number* of excited wires in the bundle. Let us employ a simple decision rule: set a decision threshold $\gamma \in [0, 1]$; if the number of excited wires in the bundle exceeds γN, then 1 is decided; otherwise, 0 is decided. When the stage number n is odd, the desired state of its output is the opposite of the state of the input to the first stage. For instance, the desired state of the output of the nth stage is 1 when 0 is sent; hence, the output error is the total probability for it to be interpreted in state 0,

$$P_{e,0} = Pr(L_n < \gamma N) = \sum_{i=0}^{i=\lfloor N\gamma \rfloor} P_n(i), \tag{12.32}$$

where P_n denotes the distribution of the output excitation level for the nth stage given by Eq. (12.30). Similarly, when 1 is sent, the error probability is

$$P_{e,1} = Pr(L_n > \gamma N) = \sum_{i=\lceil N\gamma \rceil}^{i=N} P_n(i). \tag{12.33}$$

When n is even, the desired state of the output of the nth stage is the same as the state of the input to the first stage, and the output error probabilities can be defined similarly. With a logic depth of n, system error and reliability are then defined by

$$P_e = \max(P_{e,0}, P_{e,1}), \tag{12.34}$$

$$P_r = 1 - P_e. \tag{12.35}$$

There are two basic issues. One is to choose a suitable decision threshold $\gamma \in [0, 1]$. The other is to decide a suitable P_n to use in evaluating Eqs. (12.32) and (12.33). Let us discuss the latter first.

We claim that in Eq. (12.32), we can choose $P_n = \vec{\pi}_+$, i.e., Z in Eq. (12.30) is given by x_+. Similarly, in Eq. (12.33), we can choose $P_n = \vec{\pi}_-$, with Z in Eq. (12.30) substituted by x_-. Such a choice is natural when the logic depth, i.e., the number of stages, is very large. When the logic depth is small, the error probabilities given by Eq. (12.32) and Eq. (12.33) based on $P_n = \vec{\pi}_+$ and $P_n = \vec{\pi}_-$, respectively, can be considered proper upper bounds: when the first stage is in a state of 1, we require that at least Nx_+ wires are excited; when the state of the first stage is 0, we require that at most Nx_- wires are excited. This way, $\vec{\pi}_+$ will be approached from the right, and $\vec{\pi}_-$ from the left; hence, $P_{e,1}$ and $P_{e,0}$ based on $\vec{\pi}_-$ and $\vec{\pi}_+$ are upper bounds. The input to the first stage can, of course, be easily controlled.

Let us now discuss how to choose γ. First, we examine whether the commonsense choice $\gamma = 1/2$ would work or not. Since a NAND multiplexing unit is purported to work as a NAND gate, we only need to consider if the threshold $\gamma = 1/2$ would work for a NAND gate. If $\gamma = 1/2$ would work, then given an input $x_n < 1/2$, we should get an output $x_{n+1} > 1/2$, and vice versa. However, by Eqs. (12.11) and (12.19), this is not the case: when an input x_n satisfies $1/2 < x_n < x_0$, where x_0 is given by Eq. (12.13) (see also Fig. 12.4), we have an output $x_{n+1} > x_0 > 1/2$. This discussion makes it clear that a natural decision rule is to set $\gamma = x_0$. We

Bundle Size N	50	100	200	300
P_e	1.29×10^{-9}	9.6×10^{-18}	4.47×10^{-33}	1.76×10^{-47}
P_g	5.29×10^{-14}	3.79×10^{-26}	2.71×10^{-50}	2.23×10^{-74}

Table 12.1 Output error probability for a NAND multiplexing system built from 2–input NAND gates. P_e is computed using binomial distribution and P_g is computed using Gaussian approximation. The gate error probability is assumed to be $\epsilon = 0.05$.

emphasize, however, that so long as $\epsilon \ll \epsilon_*$, $N \gg 1$ and γ is somewhere in the middle of x_- and x_+, $P_{e,0}$ and $P_{e,1}$ given by Eqs. (12.32) and (12.33) will be fairly insensitive to the actual choice of γ. This is because the probability distributions are sharply peaked around x_- and x_+ (see Fig. 12.10).

Now we can explicitly derive the expression for P_e or P_r. Due to the asymmetry in the shape of the bifurcation diagram of Fig. 12.4 and the shape of the distributions for $\vec{\pi}_+$ and π_- (see Fig. 12.10), we have found that $\vec{\pi}_+$ should be used to evaluate P_e. That is,

$$P_e = \vec{\pi}_+(I < Nx_0) = \sum_{i=0}^{i=\lfloor Nx_0 \rfloor} \binom{N}{i} (x_+)^i (1 - x_+)^{N-i} \ . \tag{12.36}$$

To appreciate how system error depends on redundancy, in Table 12.1 we have listed a few P_e for four different bundle sizes N when the gate error probability ϵ is fixed at 0.05. For convenience of comparison, the output error probabilities computed using Gaussian approximation are also included.

Now we are in a position to truly appreciate why normal approximation to Eq. (12.30) is not useful in practice. There are two basic reasons: (1) the distributions for the output of the multiplexing unit (see Fig. 12.10) peak around x_+ or x_- but are not symmetric about x_+ or x_-, especially when the gate error probability ϵ is close to 0; (2) the system error probability is basically contributed by the probabilities that are away from x_+ or x_-, not around x_+ or x_-. Those tail probabilities tend to be underestimated by Gaussian distribution, whose tail probabilities decay faster than binomially. As reflected by the error probabilities listed in Table 12.1, normal approximation underestimates system error probability by many orders of magnitude. The smaller the gate error ϵ or the larger the bundle size N, the more severely P_g underestimates P_e.

12.4.3.3 Failure of Markov chain modeling When we model the multiplexed system by a Markov chain, stimulated wires in an output bundle of size N are described by a stochastic process $K = \{K_n; n \in M\}$ where n is the stage number, $K_n \in J = \{0, 1, ..., N\}$ is the state of the process K at stage n, and

$M = \{0, 1, \cdots\}$. The relation between the input and output distributions of each multiplexing unit (also referred to as the stage) is described by a first-order Markov chain:

$$P(K_{n+1} = j_{n+1}|K_0 = j_0, ..., K_n = j_n) = P(K_{n+1} = j_{n+1}|K_n = j_n). \quad (12.37)$$

The probability transition matrix $\mathbf{P}$ is given by

$$\mathbf{P} = [P(j|i)],$$

where the matrix element $P(j|i) = P(K_{n+1} = j|K_n = i)$ is given by the conditional probability

$$P(j|i) = \binom{N}{j} Z(i)^j (1 - Z(i))^{N-j}$$

and

$$Z(i) = (1 - \epsilon) - (1 - 2\epsilon)(i/N)^2. \quad (12.38)$$

Note that Eq. (12.38) is in fact used only to compute the distribution for the first stage of the Markov chain. Distributions for the later stages of the Markov chain are obtained through iteration of the probability transition matrix,

$$\vec{\pi}(k+1) = \vec{\pi}(k)\mathbf{P},$$

where $\vec{\pi}(k)$ is the distribution for the kth stage of the Markov chain.

To understand why such a Markov chain model fails to describe the multiplexing system, it is sufficient to note that the stationary distribution for the Markov chain is simply $(\vec{\pi}_+ + \vec{\pi}_-)/2$. In other words, the stationary distribution is given by the average of the solid and dash-dot curves in Fig. 12.10. This is completely different from the analysis using the bifurcation approach: the solid and dash-dot curves are the distributions for two successive stages when the stage number of the multiplexing system is large. Now, by the definition of system reliability discussed earlier, we see that the error probability based on the Markov chain model eventually deteriorates to 1/2 when the stage number increases. Therefore, the Markov chain approach is not valid.

12.5 BIBLIOGRAPHIC NOTES

An excellent discussion on elementary bifurcation theory can be found in [415]. Readers interested in error thresholds of noisy gates are referred to [121, 122, 161, 211, 348, 359, 460].

12.6 EXERCISES

1. Show that the first-order system $\dot{x} = r - x - e^{-x}$ undergoes a saddle-node bifurcation as r is varied and find the value of r at the bifurcation point.

2. Show that the first-order system $\dot{x} = x(1 - x^2) - a(1 - e^{-bx})$ undergoes a transcritical bifurcation at $x = 0$. Find a combination of the parameters a and b as the control parameter.

3. Explicitly find U, T, and r that lead Eq. (12.2) to Eq. (12.3).

4. By making a simple transformation, prove that Eq. (12.11) is equivalent to the logistic map.