

2 Structures, Graphs and Networks

2.1 Directed Graphs

An important graph in process technology is the directed graph (Gozinto graph) that specifies the interconnection of components, assemblies and modules within a plant or a device. An example demonstrates the procedure.

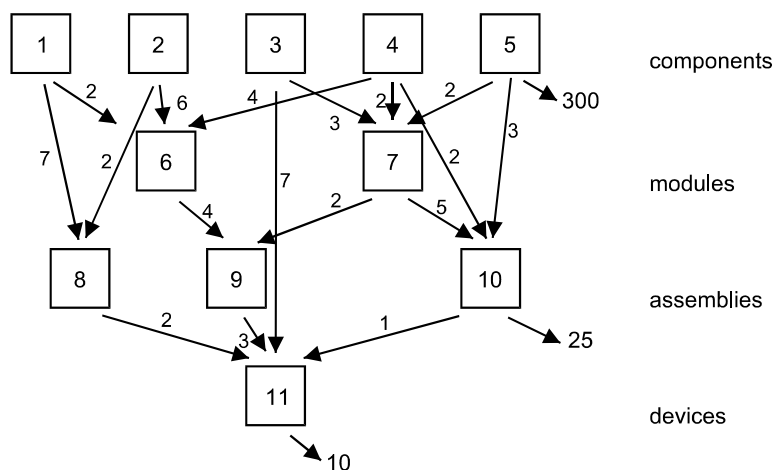


Figure 2.1. Gozinto graph

The nodes are numbered in increasing sequence and define a cost coefficient a_{ij} that specifies which set of products E_i is required to produce just *one* item of the product E_j .

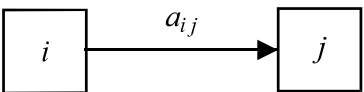


Figure 2.2. Coefficient of costs

When all the coefficients in a matrix are noted, this produces the costs matrix.

$$\mathbf{A} = (a_{ij}) \quad (2.1)$$

Because no closed-arrow sequences are possible in directed graphs, a numbering of the nodes can always be achieved in which each arrow points from a node with a smaller number to a node with a larger number. This means coefficients of costs are only possible outside the leading diagonals. For example, $a_{5,7} = 2$ in Figure 2.1 means that exactly 2 components of type E_5 are required to manufacture 1 item of the E_7 module.

The costs matrix obviously always has the value zero in the leading diagonals. A zero must be entered in all the empty fields.

However, for clarity, this is not done. For the example, the costs matrix is shown in Figure 2.3.

$$\mathbf{A} = \begin{pmatrix} 0 & & & & & & & & \\ & 0 & & & & & & & \\ & & 0 & & & & & & \\ & & & 0 & & & & & \\ & & & & 0 & & & & \\ & & & & & 0 & & & \\ & & & & & & 0 & & \\ & & & & & & & 0 & \\ & & & & & & & & 0 \end{pmatrix}$$

Figure 2.3. Costs matrix (example)

The calculation of the sets z_i of products E_i ($i = 1, \dots, N$) is now performed as follows.

Add all set requests made to node i :

$$z_i = a_{ij} \cdot z_j + a_{ik} \cdot z_k + a_{il} \cdot z_l + \dots + a_{iN} \cdot z_N + y_i \quad (2.2)$$

This equation assumes that an arrow is also possible to all subsequent nodes, otherwise the corresponding cost coefficient is zero. The set y_i is the set that leaves the system.

Example: $i = 5$

$$z_5 = 2 \times z_7 + 3 \times z_{10} + 300$$

where $a_{5,6} = 0$, $a_{5,7} = 2$, $a_{5,8} = 0$, $a_{5,9} = 0$, $a_{5,10} = 7$, $a_{5,11} = 0$ and $y_5 = 300$.

When Equation (2.2) is written for all nodes and generalised, this produces the interconnection equation

$$\mathbf{z} = \mathbf{A} \cdot \mathbf{z} + \mathbf{y} \quad (2.3)$$

with the production vector \mathbf{z}

$$\mathbf{z} = \begin{pmatrix} z_1 \\ z_2 \\ \cdot \\ \cdot \\ \cdot \\ z_N \end{pmatrix} \quad (2.4)$$

and the delivery vector (output vector) \mathbf{y} .

$$\mathbf{y} = \begin{pmatrix} y_1 \\ y_2 \\ \cdot \\ \cdot \\ \cdot \\ y_N \end{pmatrix} \quad (2.5)$$

The task now involves calculating the production vector \mathbf{z} for a given matrix \mathbf{A} and for a given output vector \mathbf{y} .

Equation (2.3) produces

$$\mathbf{z} = \mathbf{B} \cdot \mathbf{y} \quad (2.6)$$

where the demand matrix \mathbf{B}

$$\mathbf{B} = (\mathbf{E} - \mathbf{A})^{-1} \quad (2.7)$$

with the identity matrix \mathbf{E} .

However, it is not necessary to solve this task using a matrix inversion but with an algorithm designated as a mirror algorithm. To do this, the costs matrix is used to create a table. The leading diagonals represent a mirror. The algorithm runs as follows: start in row 11 and set the value for $y_{11} = 10$ in the column z , *i.e.* $z_{11} = 10$.

Continue with row 10: search for $z_{11} = 10$ and mirror this number to the cost coefficient $a_{10;11} = 1$, form the product and add the value $y_{10} = 25$, *i.e.*

$$\begin{aligned} z_{10} &= z_{11} \times a_{10;11} + y_{10} \\ &= 10 \times 1 + 25 = 35 \end{aligned}$$

	1	2	3	4	5	6	7	8	9	10	11	z	y	x
1						2		7				380		380
2						6		2				760		760
3							3				7	775		775
4						4	2			2		1,020		1,020
5							2			3		875	300	875
6									4			120		
7									2	5		235		
8											2	20		
9											3	30		
10											1	35	25	
11												10	10	

Figure 2.4. Mirror algorithm

Similarly with row 9

$$\begin{aligned} z_9 &= z_{11} \times a_{9;11} + z_{10} \times a_{9;10} + y_9 \\ &= 10 \times 3 + 35 \times 0 + 0 = 30 \end{aligned}$$

etc.

For completion: row 5

All z_6 to z_{11} are known, then it follows:

$$z_5 = 35 \times 3 + 235 \times 2 + 300 = 875$$

The vector \mathbf{x} represents the purchased parts and agrees with the corresponding product sets of the production vector \mathbf{z} .

2.2 Network Planning

2.2.1 Principles

Technological complete processes can be represented as networks. Thus, the network is a model for the manufacturing process that permits calculations

based on practice, clear graphical representations and structure analyses. Although there are many types of networks, all have the process duration d as the most important characteristic value. It is usual to represent the product flow through a network as a model. The oldest network method is known by the name CPM (critical path method). The process durations are considered as being determined, where the individual processes are represented as arrows and are generally designated in the network as *activities* or actions. The nodes between the arrows are called *events* in the network.

This network is thus a directed graph with a source and a sink. Thus, only arrows begin in the source, only arrows end in the sink. The duration of an activity is also called the *length* of an arrow. The sequence of arrows forms a *path*. The length of the path is the sum of the lengths of all arrows along the path. The critical path is the longest of all possible paths from the source to the sink.

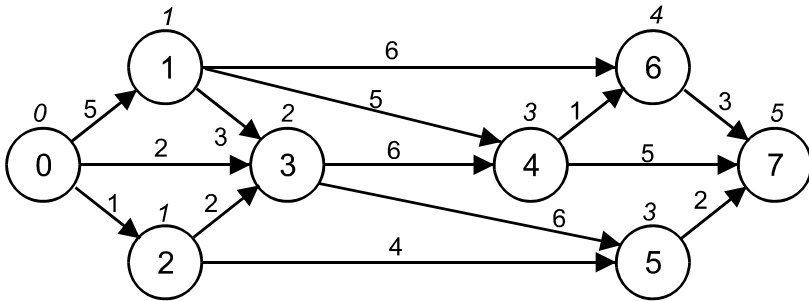


Figure 2.5. A network as a graph

A CPM network is a very understandable and simple model for a complete process. Although it does not demand any particular mathematical requirements, a good structure analysis of the product flow is necessary.

2.2.2 CPM Network and Equivalent Representations

Network as graph

The network consists of $N + 1$ nodes and at least N arrows, where a closed arrow series is not possible. This allows the nodes to always be numbered so that each arrow points from a smaller to a larger number. The source and the sink receive the number 0 and N , respectively. When all arrows that leave the source are removed, this produces (at least) one new source. This receives the number 1 (rank 1). When there are several new sources, they all receive the rank 1. Then continue and assign rank 2, etc. The numbering then continues with increasing rank (algorithm from Ford). The rank r_i of a

node (i) is equal to the maximum number of arrows from the source 0 to the node (i). In other words: the rank r_i is the length of the critical path when each arrow has the length “1”.

Figure 2.5 shows an example of a network. It has 8 nodes and 14 arrows. 17 possible paths lead from the source to the sink: the longest path is the *critical path* with length 19. The rank is shown above the nodes. It can be seen on the network whether activities can be moved or whether events have buffer times.

Network as matrix

Form a square matrix with $N + 1$ rows and write the durations between the nodes at the appropriate locations.

$t_f(i)$	j	0	1	2	3	4	5	6	7
i									
0	0	-	5	1	2				
5	1		-		3	5		6	
1	2			-	2		4		
8	3				-	6	6		
14	4					-		1	5
14	5						-		2
15	6							-	3
19	7								-
$t_s(j)$		0	5	6	8	14	17	16	19

Figure 2.6. A network as amatrix

There are *no* entries in the leading diagonal and below the leading diagonal. If one entered a zero there, this would correspond to an arrow with the duration $d = 0$, which would be equivalent to a “pseudoactivity”. Pseudoactivities are necessary when chronological and technological dependencies require them. Similarly, it is also not permitted to enter zeros at free locations above the leading diagonals. Thus, although no matrix exists in the mathematical sense, this term will be used in the discussed case.

Section 2.2.3 discusses the specified column with the designation $t_f(i)$ and row with the designation $t_s(j)$ in Figure 2.6.

The structure and other properties of the graphs can also be read from the matrix. For example, it is easy to see from a column (j) from which nodes the arrows end in the nodes (j). Similarly, take a row (i) to see where the leaving arrows lead.

If most fields above the leading diagonal are set, a chain structure is in effect present. If all fields are occupied, this is called a complete graph. Complete graphs seldom occur in process technology.

The network can be uniquely obtained from the matrix. Both representations are equivalent.

Network as GANTT diagram

The GANTT diagram is the representation of all activities on the time axis. Two rules are prescribed:

1. All activities $(i; j)$ are represented proportional to their duration $d(i; j)$ as early as possible (*i.e.* as far left in the diagram as possible).
2. All activities are drawn in reciprocal lexicographical sequence. In other words: first the activity with $j = 1$, then the activities with $j = 2$ with increasing i , then with $j = 3$, *etc.*

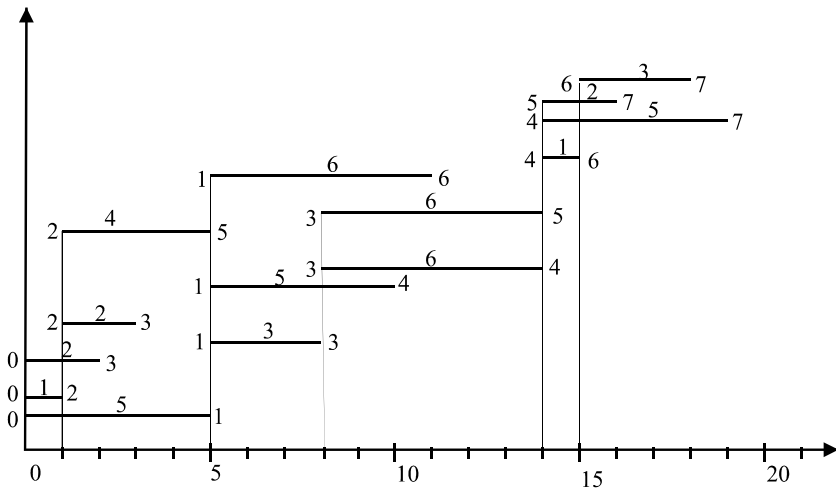


Figure 2.7. GANTT diagram

Figure 2.7 shows the GANTT diagram for the example network. The critical path is easy to see from the diagram. It consists of the 4 activities (0;1), (1;3), (3;4) and (4;7).

The GANTT diagram is also equivalent to the other two network representations. Each of the 3 representation forms can be derived from each of the other forms and also transformed into each other form.

2.2.3 Determining Dates in the Network

Event dates

Each event can be characterised by 2 dates.

$t_e(i)$ – earliest date of the event (i)

$t_l(i)$ – latest date of the event(i).

Both dates are calculated recursively (by choosing the indices appropriately).

$$\begin{aligned}
 t_e(0) &= 0 \\
 t_e(j) &= \text{Max}_i \{ t_e(i) + d(i; j) \} \\
 t_l(N) &= t_e(N) \\
 t_l(i) &= \text{Min}_j \{ t_l(j) - d(i; j) \}
 \end{aligned} \tag{2.8}$$

These calculations can be performed very simply with the matrix (see Figure 2.6). To calculate t_e of the row i go to the column i , add the values in this column to each of previously calculated values t_e and determine the maximum. $t_l(j)$ is calculated similarly.

For critical events (i_k)

$$t_l(i_k) = t_e(i_k). \tag{2.9}$$

The critical events in the example network are (0), (1), (3), (4) and (7).

Activity dates

Four dates can be specified for each activity ($i; j$)

$t_{es}(i; j)$ – earliest start date of the activity ($i; j$)

$t_{ee}(i; j)$ – earliest end date of the activity ($i; j$)

$t_{ls}(i; j)$ – latest start date of the activity ($i; j$)

$t_{le}(i; j)$ – latest end date of the activity ($i; j$)

with the following computing rules

$$\begin{aligned}
 t_{es}(i; j) &= t_f(i) \\
 t_{ee}(i; j) &= t_f(i) + d(i; j) \\
 t_{ls}(i; j) &= t_s(j) - d(i; j) \\
 t_{le}(i; j) &= t_s(j)
 \end{aligned} \tag{2.10}$$

Figure 2.8 shows the dates for the activity (2, 5):

$$\begin{array}{ll}
 t_e(2) = 1 & t_{es}(2; 5) = 1 \\
 t_l(2) = 6 & t_{ee}(2; 5) = 5 \\
 t_e(5) = 14 & t_{ls}(2; 5) = 13 \\
 t_l(5) = 17 & t_{le}(2; 5) = 17
 \end{array}$$

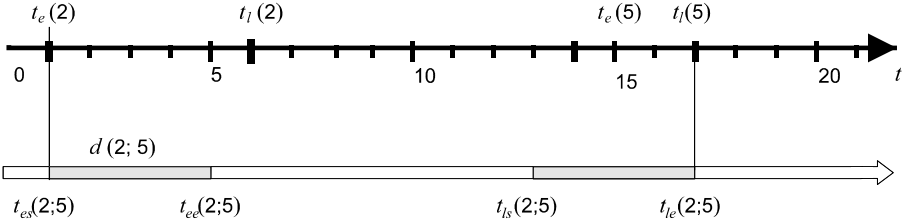


Figure 2.8. Date calculations for the activity (2;5)

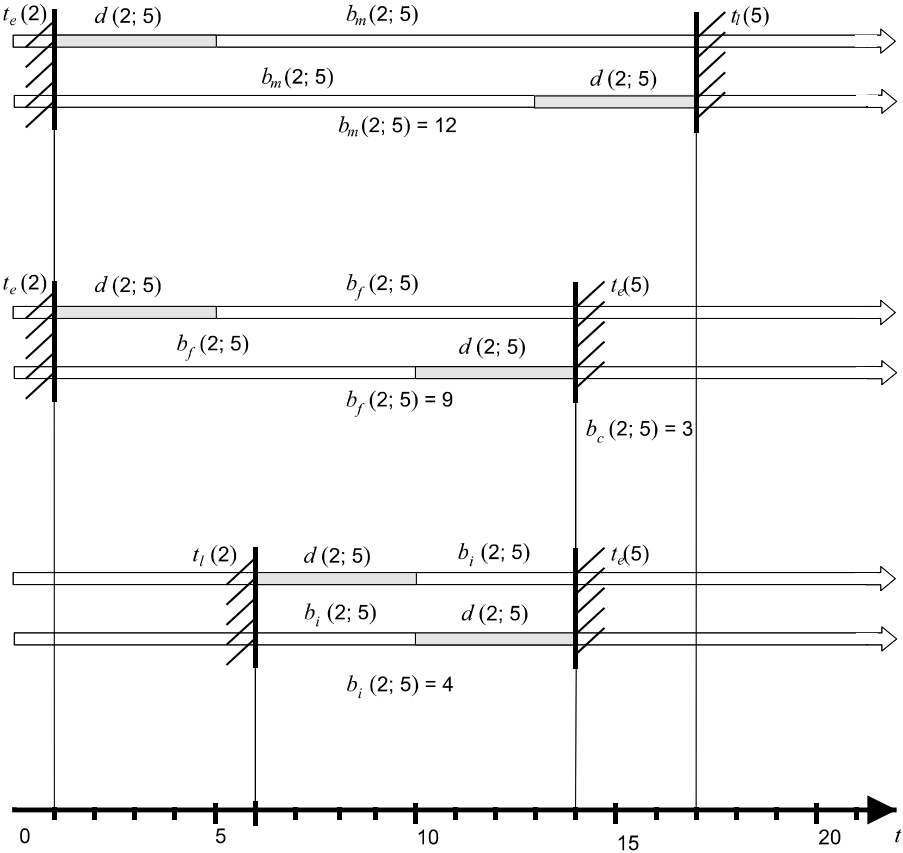


Figure 2.9. Buffer times for the activity (2;5)

2.2.4 Calculation of Buffer Times in the Network

Buffer time of the event (i)

The buffer time $b(i)$ of the event (i) is calculated as

$$b(i) = t_l(i) - t_e(i) \quad (2.11)$$

The buffer time is zero for critical events i_k .

$$b(i_k) = 0 \quad (2.12)$$

The actual event date $t(i)$ always lies between $t_e(i)$ and $t_l(i)$. This means that critical events do not have any freedom for date displacements. Buffer times are also called slack times.

Buffer times of the activities ($i; j$)

There are 4 possible buffer times:

$b_m(i, j)$ – maximum buffer time of the activity ($i; j$)

$b_c(i, j)$ – conditional buffer time of the activity ($i; j$)

$b_f(i, j)$ – free buffer time of the activity ($i; j$)

$b_i(i, j)$ – independent buffer time of the activity ($i; j$)

The calculation equations are

$$b_m(i, j) = t_l(j) - t_e(i) - d(i, j) \quad (2.13)$$

$$b_c(i, j) = t_l(j) - t_e(j) = b(j)$$

$$b_f(i, j) = t_e(j) - t_e(i) - d(i, j)$$

$$b_i(i, j) = \text{Max}\{0; t_e(j) - t_l(i) - d(i, j)\}$$

The activity (2; 5) is again used as an example for calculating the buffer times (Figure 2.9).

2.2.5 Strategies in Network Planning

The critical path determines the duration of a complete process. The critical activities (i, j)_k cannot be displayed and must begin exactly at their start date

$$t_s(i, j)_k = t_{es}(i, j)_k = t_{ls}(i, j)_k \quad (2.14)$$

Each network has (at least) one critical path. The example has 4 critical activities: (0; 1), (1; 3), (3; 4), (4; 7). The other 10 activities are not critical. The actual start date can be freely chosen between the latest (latest permitted) and the earliest (earliest possible) start date. The free choice of the start date of a critical activity, however, is subject to restrictions:

Choice of the maximum buffer time (Figure 2.9, top)

If the activity $(i; j)$ is started at time $t_e(i)$, all activities that end at (i) will be forced to be completed by time $t_e(i)$. Thus you do not have the possibility, which you actually have, to end at $t_l(i)$. Consequently, you must intervene in the “freedom” of the preceding activities. The same thing occurs if you allow an activity to end at time $t_l(j)$. The subsequent activities that begin at (j) must wait and cannot, which would be possible, begin already at $t_e(j)$. Thus, if an activity (i, j) uses the maximum possible buffer time and occupies the complete interval $t_e(i)$ to $t_l(j)$, both the preceding activities and the subsequent activities will be hindered.

Thus, laying claim to the maximum buffer time is the *most inconsiderate behaviour* of an activity in the network.

Choice of the free buffer time (Figure 2.9, centre)

In this case, the activity $(i; j)$ claims the interval between $t_e(i)$ and $t_e(j)$. This takes account of all subsequent activities that are fully free in deciding to start as early as possible. This is a *desirable* behaviour in the network, but, however, requires all previous activities to complete at the earliest possible date.

Choice of the independent buffer time (Figure 2.9, bottom)

The activity $(i; j)$ occupies only the interval $t_l(i)$ to $t_e(j)$. This allows each subsequent activity to freely decide on the start and end, and will not be hindered. This is the absolutely *most considerate* behaviour of the activity $(i; j)$ in the network.

Choice of independent and free buffer time

Figure 2.9 shows that the activity $(i; j)$ in this case occupies the interval between $t_l(i)$ and $t_e(j)$. This strategy allows all preceding activities to freely decide but, if necessary, be complete at the latest permitted date. This behaviour is *conditionally sensible*.

Thus, the choice of the actual start dates $t_s(i; j)$ for an activity $(i; j)$ is always associated with a strategy. In the case of the independent buffer time, it is, however, possible that such a strategy is not available because the computed value yields

$$t_e(j) - t_l(i) - d(i; j) < 0$$

namely, it is negative. Then, the definition states $b_i = 0$, because the maximum formed between 0 and a negative number gives the value zero. Thus, in this case, the “possible” strategy cannot be achieved.

Thus, one should always perform an exact network analysis and control the true process execution using a good strategy.

2.2.6 CPM Cost

Previously, the duration $d(i; j)$ was always considered to be an invariant quantity, only the start of the activity $t_s(i; j)$ was considered to be variable within certain limits. To better conform to the practical conditions, the costs are also included in the CPM network planning as follows.

The duration is considered to be a quantity that possibly may be shortened, although such a shortening of the duration causes higher costs. When one creates a cost–duration diagram, this produces the general behaviour shown in Figure 2.10.

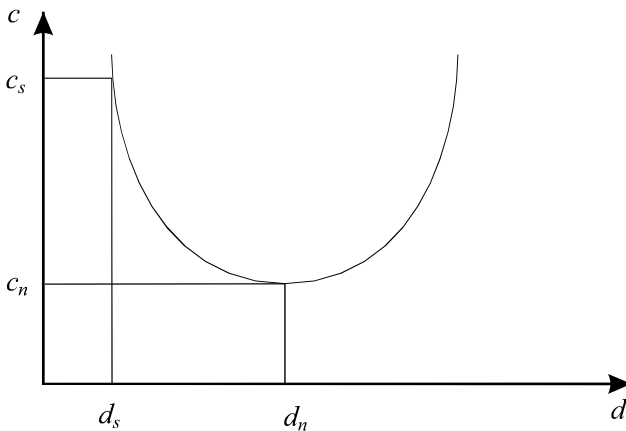


Figure 2.10. General cost–duration diagram

The duration of an activity can in general only be reduced to d_s , even if the costs further increase. Conversely, an arbitrary reduction of the costs, despite the lengthening of the duration, also cannot be achieved. Quite the contrary, the costs will further increase for a longer duration. The actual “stable” costs–duration characteristic curve lies between d_s and d_n . If this part is replaced with a straight line, this produces the linear cost–duration diagram shown in Figure 2.11.

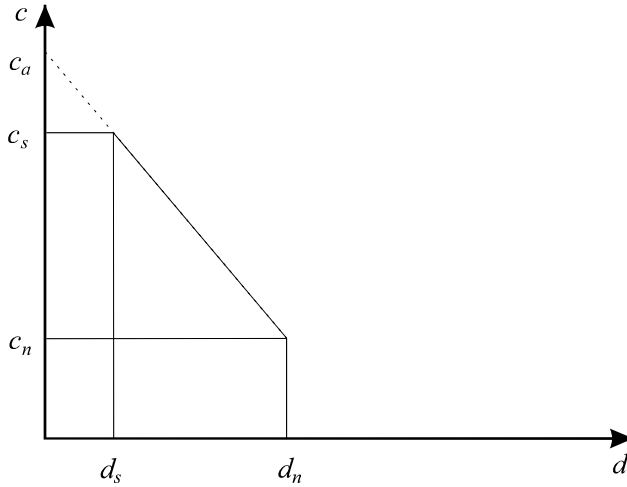


Figure 2.11. Linear cost–duration diagram

The complete duration D of the network and the complete costs C lie in an area (Figure 2.12).

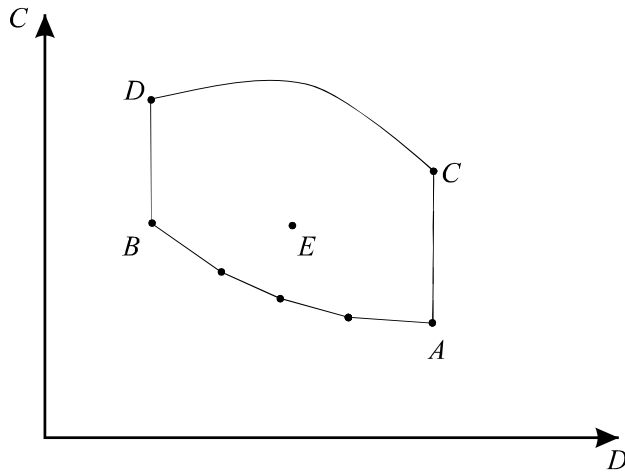


Figure 2.12. Cost–duration area of the complete network

Operating point A: *cheapest* and *longest* network. All activities (i, j) have the duration $d_n(i, j)$, i.e. each with the longest duration and the least cost.

Curve A–B: the reduction of the critical path starting at A. Initially starting at *that* activity that when shortened shows the smallest cost increase, etc. The path is formed from those straight lines that always show the largest increase.

Operating point B: the *cheapest* and *shortest* network.

Section B–D: reduction of the non–critical activities starting at B; this makes the network more expensive but not shorter.

Operating point D: all activities for d_s and c_s ; the *shortest* and *most expensive* network.

Section A–D: starting at A. Whereas non–critical activities will be shortened, critical ones will not be shortened. The critical path remains, but all non–critical activities are at d_s and c_s .

Operating point C: the *longest* and *most expensive* network.

Operating point E: the normal, not optimised, network.

2.3 Product-flow Graphs

2.3.1 Weighting Function for Technological Processes

Introduction

The duration of technological processes is described by a random variable with the probability density $f(t)$. In practice, the process duration always lies only between a smallest value a and a largest value b . Consequently, Section 1.3.2 specifies the integration limits with a and b (see Figure 1.22). Obviously the integration limits can also be extended $(-\infty, +\infty)$ when one takes into consideration that the following equation applies

$$f(t) = 0 \quad \text{for } t < a \quad \text{and } t > b \quad (2.15)$$

For example, using Equation (2.15), it can also be written as:

$$\int_a^b f(t) dt = \int_{-\infty}^{+\infty} f(t) dt = 1 \quad (2.16)$$

The following considerations do not assume a specific distribution. Other than the process duration, a probability w is assigned to each technological process as an additional characteristic value. w should be called the

realisation probability and specifies the probability with which a technological process occurs.

$$0 < w \leq 1 \quad (2.17)$$

If, for example, defect products with a defect rate p are determined in an inspection process and these products are subsequently repaired, the repair process then has the realisation probability $w = p$.

The product of w and $f(t)$ is called the *weighting function* $g(t)$.

$$g(t) = w \cdot f(t) \quad (2.18)$$

This allows many relationships from the linear system theory [7] to be used for technological processes. Several examples are used to demonstrate this procedure.

Weighting Function for Constant Process Duration

The technological process is not stochastic, but rather determined with the constant duration d . In this case, the density $f(t)$ becomes the *Dirac function* at position d , and the weighting function reduces to

$$g(t) = w \cdot \delta(t - d) \quad (2.19)$$

With regard to the distribution theory, the Dirac function is also a probability density here with the area 1, although it assumes the zero value and for $t = d$ the infinite value for $t \neq d$. It is an infinitely narrow and an infinitely high *pulse* and thus a special or boundary case for ordinary functions.

One can illustrate the occurrence of the Dirac function as follows: for constant area and constant expected value μ , the ordinary distribution $f(t)$ is made ever narrower (and thus ever higher) until finally the Dirac pulse occurs in the boundary case. The weighting function (2.19) also characterises the technological process so that a product with the probability w (in practical terms: the *proportion* w of a *product set*) flows through the technological process and requires the duration d from start to end, *i.e.* appears at the output after duration d . The system theory assigns an input x_1 and an output x_2 to the technological process. The technological process takes place between this input and output. This can be interpreted as a delay system for a constant process duration.

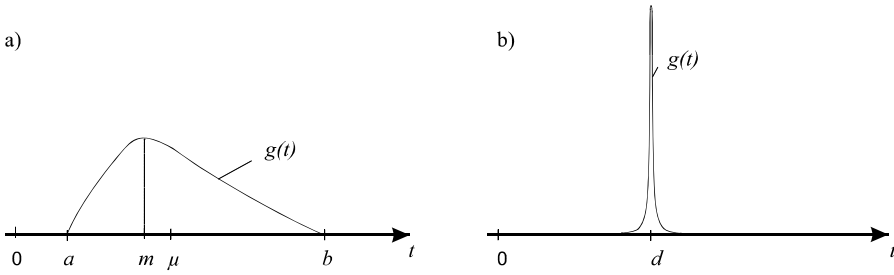


Figure 2.13. Weighting function for technological processes – (a) stochastic, (b) determined

Weighting Function and Pulse Response

If a Dirac pulse $\delta(t)$ is applied at the input $x_1(t)$ of a technological process at time $t = 0$, the so-called pulse response appears at the output $x_2(t)$ as a response (see Figure 2.14). The system theory proves that the pulse response is always identical to the weighting function $g(t)$, *i.e.*

$$x_1(t) = \delta(t) \Leftrightarrow x_2(t) = g(t) \quad (2.20)$$

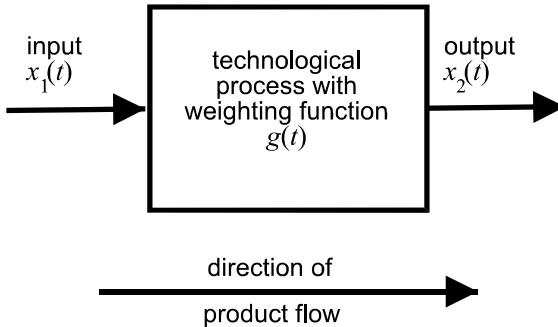


Figure 2.14. System representation of the technological process

Weighting Function and System Equation

The system equation describes the general relationship between $x_1(t)$, $g(t)$, $x_2(t)$, where $x_1(t)$ and $x_2(t)$ are also probability densities. The system equation is a convolution equation defined by the following integral:

$$\begin{aligned}
x_2(t) &= g(t) * x_1(t) \\
&= \int g(u) \cdot x_1(t-u) du \\
&= \int g(t-u) \cdot x_1(u) du
\end{aligned} \tag{2.21}$$

Calculation of the Process Parameters Using the Weighting Function

Parameters, such as expected value, deviation, standard deviation and variation coefficient, process duration, but also the realisation probability of technological processes, have major significance for practical investigations and for configuring. We now show here how some of these parameters can be calculated using the weighting function $g(t)$.

1. Calculation of the realisation probability

Integrating Equation (2.18) and using Equation (2.17) it is easy to show:

$$w = \int g(t) dt \tag{2.22}$$

To simplify the description, we omit the specification of the integration limits a and b or $-\infty$ and $+\infty$ here and in the following section, *i.e.* if *no* limits are specified, the integral should always be performed over the complete domain.

2. Calculation of the expected value μ of the process duration

Equation (1.18) with Equation (2.18) yields

$$\mu = \frac{1}{w} \cdot \int t \cdot g(t) dt \tag{2.23}$$

3. Calculation of the variance σ^2 of the process duration

Equation (1.19) with Equation (2.18) yields

$$\sigma^2 = \frac{1}{w} \cdot \int t^2 \cdot g(t) dt - \mu^2 \tag{2.24}$$

2.3.2 Operator Technological Processes

Introduction

Using the Laplace transformation on the weighting function $g(t)$ produces the transfer function $G(s)$, also called the *technological operator*.

$$G(s) = \int_{-\infty}^{+\infty} g(t) \cdot e^{-st} dt = L\{g(t)\} \quad (2.25)$$

The symbol L is used as a short form for the multiplication of the function to be transformed with the so-called kernel e^{-st} and subsequent integration. This makes the character of a representation more apparent, *i.e.* the original domain (or t -domain, time domain) the function $f(t)$ is “represented” in the range (or s -domain, operator domain), *i.e.* transformed and called $G(s)$ there.

The Laplace transformation is a very powerful and popular method not only for solving differential equations but also for calculating time activities in systems used in the automatic control theory, in system theory and in electrical engineering.

The Laplace reverse transformation can be used to obtain the weighting function from any operator. This means $g(t)$ and $G(s)$ are fully equivalent.

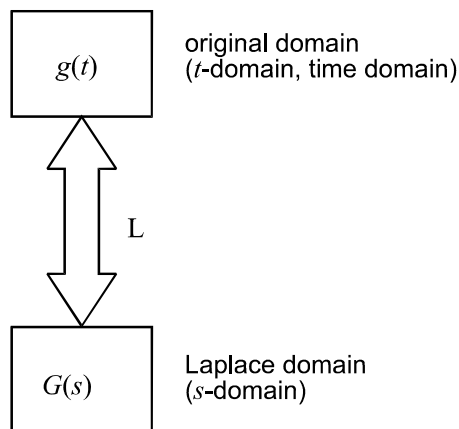


Figure 2.15. Laplace transformation

Consequently, a function in the range can be assigned to any function in the original domain and *vice versa*.

Operator for Constant Process Duration

For determined processes with a constant duration d , the operator can be calculated using Equation (2.16) in Equation (2.25).

$$G(s) = w \cdot e^{-sd} \quad (2.26)$$

This is the transform of the Dirac function (Equation (2.19)) in the range (using a realisation probability w and duration d). For the limit case $w = 1$ and $d = 0$, this yields $G(s) = 1$ and $g(t) = \delta(t)$.

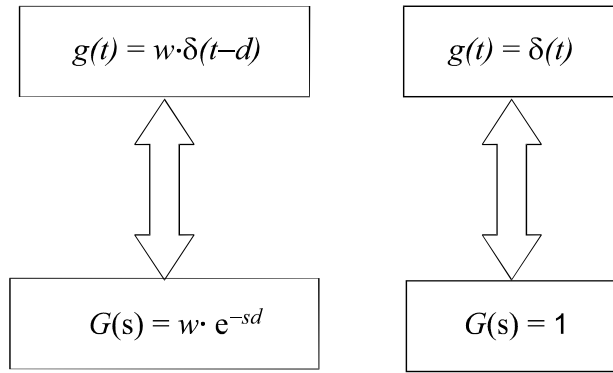


Figure 2.16. Operator for determined technological processes

System Equation in the Range

The system Equation (2.18) can be fully transformed in the range

$$X_2(s) = G(s) \cdot X_1(s) \quad (2.27)$$

where $X_1(s)$ is the Laplace transform of $x_1(t)$; $X_2(s)$ is formed similarly. Consequently, the convolution in the t -domain (2.21) corresponds to multiplication in the s -domain. This allows $G(s)$ to be determined very easily by forming the quotient

$$G(s) = \frac{X_2(s)}{X_1(s)}$$

The technological process is represented as an arrow, where $X_1(s)$ and $X_2(s)$ are interpreted as the node potential and $G(s)$ is interpreted as the transfer function.

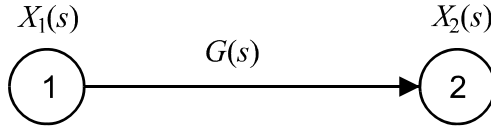


Figure 2.17. Element of a product-flow graph

Calculating Parameters from the Technological Operator

The realisation probability w , the expected value μ and the variance σ^2 of the process duration cannot only be calculated from the weighting function $g(t)$, but also from the technological operator $G(s)$.

1. Calculation of the realisation probability w

Using Equation (2.25) with Equation (2.18) yields

$$G(s) = w \cdot \int f(t) \cdot e^{-st} dt \quad (2.28)$$

Setting $s = 0$ (mathematically, one should actually form a limit value: $s \rightarrow 0$), this produces from Equation (2.26) with Equation (2.17)

$$w = G(s) \Big|_{s=0} = G(0) \quad (2.29)$$

If $G(s)$ is known, then it is very easy to calculate the value $G(0)$.

2. Calculation of the expected value

Differentiating with s on Equation (2.28) yields

$$\frac{dG(s)}{ds} = G'(s) = -w \cdot \int t \cdot f(t) \cdot e^{-st} dt \quad (2.30)$$

Setting $s = 0$, Equations (1.18) and (2.29) yield

$$\mu = - \frac{G'(s)}{G(s)} \Big|_{s=0} = - \frac{G'(0)}{G(0)} \quad (2.31)$$

This expression can be written as

$$\mu = - \frac{d}{ds} \{ \ln G(s) \} \Big|_{s=0} = - (\ln G(s))' \Big|_{s=0} \quad (2.32)$$

3. Calculation of the variance σ^2

Further differentiation of Equation (2.30) with Equation (1.19) yields

$$\begin{aligned}\frac{d^2 G(s)}{ds^2} = G''(s) &= w \cdot \int t^2 f(t) \cdot e^{-st} dt \\ &= w \cdot (\sigma^2 + \mu^2)\end{aligned}\quad (2.33)$$

and then with Equations (2.29) and (2.30)

$$\begin{aligned}\sigma^2 &= \left[\frac{G''(s)}{G(s)} - \left(\frac{G'(s)}{G(s)} \right)^2 \right]_{s=0} \\ &= \left[\frac{G'(s)}{G(s)} \right]_{s=0}' = \left[\frac{G'(0)}{G(0)} \right]' \\ &= \left[\ln G(s) \right]'' \Big|_{s=0}\end{aligned}\quad (2.34)$$

2.3.3 Basic Structures of Technological Processes

We demonstrate the procedure using the three basic structures: chain structure, parallel structure and feedback structure.

Chain Structure

Two technological processes T_1 and T_2 with operators $G_1(s)$ and $G_2(s)$ form a chain structure

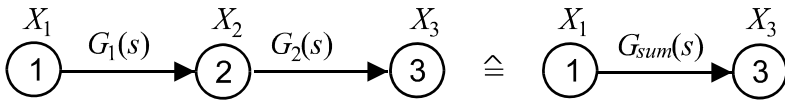


Figure 2.18. Chain structure consisting of 2 technological processes

This gives for the process

$$X_2(s) = G_1(s) \cdot X_1(s) \quad (2.35)$$

$$X_3(s) = G_2(s) \cdot X_2(s)$$

with

$$G_{sum} = \frac{X_3(s)}{X_1(s)} \quad (2.36)$$

this then yields

$$G_{sum}(s) = G_1(s) \cdot G_2(s) \quad (2.37)$$

Using Equations (2.29), (2.32) and (2.34) yields the following calculation equations

$$\begin{aligned} w_{sum} &= w_1 \cdot w_2 \\ \mu_{sum} &= \mu_1 + \mu_2 \\ \sigma_{sum}^2 &= \sigma_1^2 + \sigma_2^2 \end{aligned} \quad (2.38)$$

It is easy to extend to more than 2 process elements using the above equations. The chain structure of technological processes is the simplest and most common form used in real manufacturing systems.

Parallel Structure

Two technological processes T_1 and T_2 with operators $G_1(s)$ and $G_2(s)$ form a parallel structure.

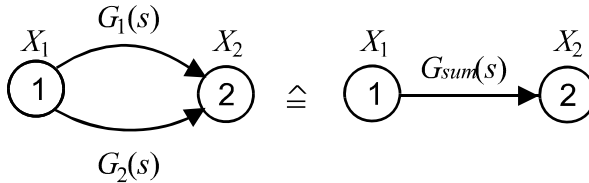


Figure 2.19. Parallel structure formed from 2 technological processes

This gives for the complete process

$$G_{sum} = G_1(s) + G_2(s) \quad (2.39)$$

which then gives

$$\begin{aligned} w_{sum} &= w_1 + w_2 \\ w_{sum} \cdot \mu_{sum} &= w_1 \cdot \mu_1 + w_2 \cdot \mu_2 \\ w_{sum} \cdot (\sigma_{sum}^2 + \mu_{sum}^2) &= w_1 \cdot (\sigma_1^2 + \mu_1^2) + w_2 \cdot (\sigma_2^2 + \mu_2^2) \end{aligned} \quad (2.40)$$

It is relatively easy to extend to more than 2 process elements results using Equations (2.40). The parallel structure of technological processes always occurs in real manufacturing processes when the products at the end of an evaluation process are split into two (or more) product flows, which then pass through different processes and are recombined to form a single flow.

Feedback Structure

To derive the overall operator, a structure is formed from 4 individual processes.

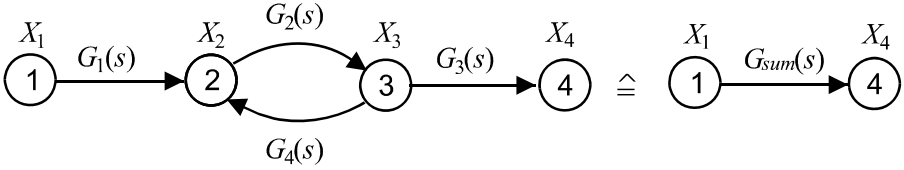


Figure 2.20. Feedback structure

The node equations are:

$$X_2(s) = G_1(s) \cdot X_1(s) + G_4(s) \cdot X_3(s) \quad (2.41)$$

$$X_3(s) = G_2(s) \cdot X_2(s)$$

$$X_4(s) = G_3(s) \cdot X_3(s)$$

The overall operator G_{sum} is calculated using:

$$G_{sum}(s) = \frac{X_4(s)}{X_1(s)} = \frac{G_1(s) \cdot G_2(s) \cdot G_3(s)}{1 - G_2(s) \cdot G_4(s)} \quad (2.42)$$

The feedback in real manufacturing systems is often caused by $G_2(s)$ and $G_4(s)$ representing an inspection process and a reworking or rejection process, respectively. This is sometimes also known as a “repair loop”. Such structures occur frequently in electronics production. From the viewpoint of process modelling, the considered feedback structure can be interpreted as being a “pure chain” with 4 elements.

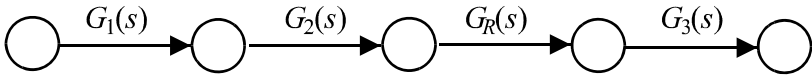


Figure 2.21. Modelling of the feedback structure as a chain structure

$$G_{sum}(s) = G_1(s) \cdot G_2(s) \cdot G_R(s) \cdot G_3(s) \quad (2.43)$$

Using the equations from Section 1.3.1, this gives

$$w_{sum} = w_1 \cdot w_2 \cdot w_R \cdot w_3 \quad (2.44)$$

$$\mu_{sum} = \mu_1 + \mu_2 + \mu_R + \mu_3$$

$$\sigma_{sum}^2 = \sigma_1^2 + \sigma_2^2 + \sigma_R^2 + \sigma_3^2$$

If one designates the operator for the chain in the feedback loop as

$$G_0(s) = G_2(s) \cdot G_4(s) \quad (2.45)$$

this produces the following operator for the feedback process:

$$G_R(s) = \frac{1}{1 - G_0(s)} \quad (2.46)$$

This is used to calculate

$$\begin{aligned} w_R &= \frac{1}{1 - w_0} & \mu_R &= \frac{w_0}{1 - w_0} \cdot \mu_0 \\ \sigma_R^2 &= \frac{w_0}{1 - w_0} \left(\sigma_0^2 + \frac{1}{1 - w_0} \cdot \mu_0 \right) \end{aligned} \quad (2.47)$$

with

$$\begin{aligned} w_0 &= w_2 \cdot w_4 \\ \mu_0 &= \mu_2 + \mu_4 \\ \sigma_0^2 &= \sigma_2^2 + \sigma_4^2 \end{aligned} \quad (2.48)$$

2.3.4 General Structures, Regularities, Relationships

Operator Equation

To calculate the overall operator of a manufacturing process with K nodes, the nodes are numbered successively. A potential $X_k(s)$ is assigned to each node with the number k . Each arrow is given an operator $G_n(s)$.

$K-1$ node equations are then prepared. They use all the arrows leading to the associated node and add the transferred potential proportions.

The overall operator is obtained by eliminating the potentials of the intermediate nodes. The following example demonstrates the procedure.

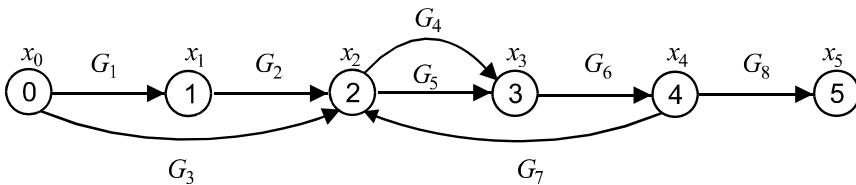


Figure 2.22. Product-flow graph for a manufacturing system

The 5 node equations are

$$X_1 = G_1 \cdot X_0 \quad (2.49)$$

$$X_2 = G_2 \cdot X_1 + G_3 \cdot X_0 + G_7 \cdot X_4$$

$$X_3 = G_4 \cdot X_2 + G_5 \cdot X_2$$

$$X_4 = G_6 \cdot X_3$$

$$X_5 = G_8 \cdot X_4$$

We use

$$X_5 = G_{sum} \cdot X_0$$

to obtain the overall operator

$$G_{sum} = \frac{(G_1 \cdot G_2 + G_3) \cdot (G_4 + G_5) \cdot G_6 \cdot G_8}{1 - (G_4 + G_5) \cdot G_6 \cdot G_7} \quad (2.50)$$

A symbolic method also allows G_{sum} to be specified when the arithmetic rules and the hierarchical system form are used for the chain, parallel and feedback structure:

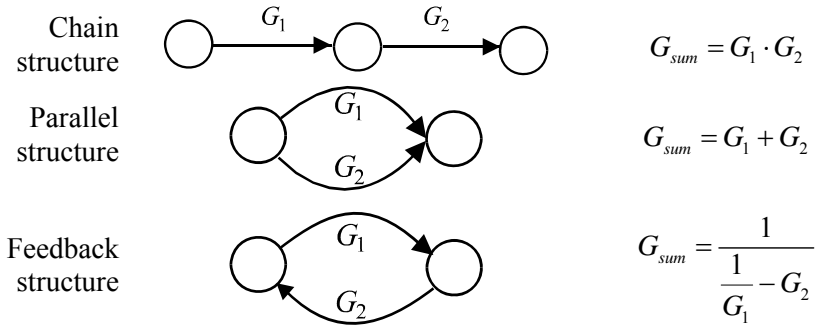


Figure 2.23. Basic structures and their operator equations

Realisation Probability

The sum of the realisation probabilities for all arrows *leaving* a node is always 1.

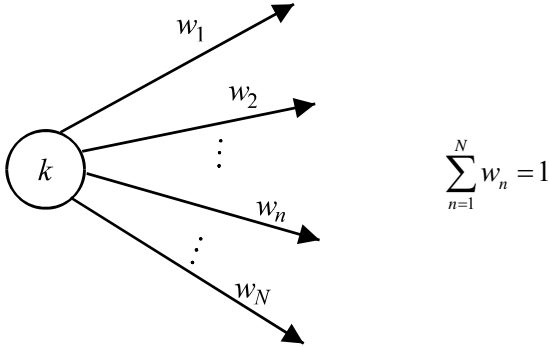


Figure 2.24. Law of conservation

This means that $w = 1$ only when *one* single arrow leaves a node. Every structure with K nodes always has $K-1$ conservation equations for the realisation probabilities that directly correspond with the structure.

As a first example, the conservation equations are specified for the feedback structure (Figure 2.20):

$$\begin{aligned} w_1 &= 1 & w_3 + w_4 &= 1 \\ w_2 &= 1 \end{aligned} \quad (2.51)$$

The second example concerns Figure 2.22

$$\begin{aligned} w_1 + w_3 &= 1 & w_4 + w_5 &= 1 & w_7 + w_8 &= 1 \\ w_2 &= 1 & w_6 &= 1 \end{aligned} \quad (2.52)$$

The overall realisation probability is calculated from the structure using Equation (2.29). This is done most easily by replacing $G_n(s)$ or G_n by w_n for all n in the operator equation.

This is demonstrated using the feedback structure as an example. Equation (2.42) then yields

$$w_{sum} = \frac{w_1 \cdot w_2 \cdot w_3}{1 - w_2 \cdot w_4} \quad (2.53)$$

Set the conservation Equations (2.51) in Equation (2.53) to obtain $w_{sum}=1$. This relationship obviously results for all systems provided they have “completed”, *i.e.* they must have a node as source (only leaving arrows) and a node as sink (only entering arrows).

The relationships are *independent* of the associated operators, *i.e.* they are also independent of μ , σ^2 and other duration parameters!

They characterise only the structure of the manufacturing system.

Realisation probabilities characterise the quality behaviour of technological processes.

Expected Value of the Duration

The relationships of the process durations and the calculation of the complete duration are very important in electronic process technology. The method of product-flow graphs can now use the basis of manufacturing structures to derive the general equations that can be used to perform the dimensioning in an actual case for real manufacturing systems.

For a manufacturing system with N separate technological processes, the expected value of the complete duration μ_{sum} is always a sum of all expected values of the individual technological processes, where an intensity coefficient c_n must be considered for each process T_n

$$\mu_{sum} = \sum_{n=1}^N c_n \cdot \mu_n \quad (2.54)$$

The feedback structure (Figure 2.20) is used as an example to explain this relationship.

Setting Equations (2.47), (2.48) and (2.51) into Equation (2.44) yields

$$\mu_{sum} = \mu_1 + \frac{1}{1 - w_4} \cdot \mu_2 + \mu_3 + \frac{w_4}{1 - w_4} \cdot \mu_4 \quad (2.55)$$

The intensity coefficients depend only on the realisation probabilities. This is true for all manufacturing systems.

In particular

$$\begin{aligned} c_1 &= 1 & c_3 &= 1 \\ c_2 &= \frac{1}{1 - w_4} & c_4 &= \frac{w_4}{1 - w_4} \end{aligned} \quad (2.56)$$

The model is explained using a numerical example.

Assuming a repair probability of $p = w_4 = 0.2$, one obtains $c_2 = 1.25$ and $c_4 = 0.25$. This means: the process T_1 has a defect rate of 20% and a yield $FPY = 0.8$ (first-pass yield). The repair process T_4 is loaded with 25% of the manufacturing flow because the repair loop is continually “repeated” in the model!

In other words: 20% of the originally manufactured products are repaired and retested; also afterwards 20% pass through the repair loop a second

time, *i.e.* 4% of the originally manufactured products have two repair loops; 0.8% three repair loops, *etc.* This means the inspection process is loaded in total with 125%.

The intensity coefficients c_n can be calculated from the process equation $G_{sum}(s)$, also with total differentiation:

From

$$G_{sum}(s) = G_{sum}(G_1(s), G_2(s), \dots, G_n(s), \dots, G_N(s)) \quad (2.57)$$

yields

$$G'_{sum}(s) = \sum_{n=1}^N \frac{\partial G_{sum}}{\partial G_n} \cdot G'_n(s) \quad (2.58)$$

and

$$\mu_{sum} = \sum_{n=1}^N \frac{\partial G_{sum}}{\partial G_n} \cdot \frac{G_n(s)}{G_{sum}(s)} \Big|_{s=0} \cdot \mu_n \quad (2.59)$$

thus

$$c_n = \frac{w_n}{w_{sum}} \cdot \frac{\partial w_{sum}}{\partial w_n} \quad (2.60)$$

The expected value of the duration of the complete process thus depends neither on the deviations nor on the associated density change, but only on the individual expected values and their intensity coefficients. Consequently, the expected value can also be calculated when the process model is assumed to be deterministic (see Section 1.3.2), *i.e.*

$$G_n(s) = w_n \cdot e^{-\mu_n \cdot s} \quad (2.61)$$

Duration Deviation

The further differentiation for s on Equation (2.58) yields

$$G''_{sum}(s) = \sum_{n=1}^N \frac{\partial G_{sum}}{\partial G_n} \cdot G''_n(s) + \sum_{m=1}^N \sum_{n=1}^N \frac{\partial^2 G_{sum}}{\partial G_m \partial G_n} \cdot G'_m(s) \cdot G'_n(s) \quad (2.62)$$

Dividing this $G_{sum}(s)$ with Equations (2.29), (2.31) and (2.34)

$$\sigma_{sum}^2 + \mu_{sum}^2 = \sum_{n=1}^N c_n \cdot (\sigma_n^2 + \mu_n^2) + \sum_{m=1}^N \sum_{n=1}^N d_{mn} \cdot \mu_m \cdot \mu_n \quad (2.63)$$

with c_n according to Equation (2.60) and

$$d_{mn} = \frac{w_m \cdot w_n}{w_{sum}} \cdot \frac{\partial^2 w_{sum}}{\partial w_m \cdot \partial w_n} \quad (2.64)$$

In practice, however, the deviation equation is almost never used.

2.4 Queue Models

Although queuing theory is generally suitable for the analytical modelling of manufacturing processes, its use also requires good knowledge of mathematics, in particular, probability theory. Unfortunately, the high abstraction level and structural reasons mean narrow limits are set on its practical use. Thus, complex systems with complicated controls can, at best, only be approximated. Queue models are infinite time-continuous models controlled stochastically by events. Furthermore, the models are both event and state-discrete, and consider not only the finiteness of the resources but also their interactions.

Queuing systems consist essentially of three basic elements

- service stations
- queues
- event or demand flows.

The stations and queues are connected with each other with a network consisting of directed edges. The demand flows run along these edges; the demand flows can be separated or combined at special nodes of the graphs (Figure 2.25).

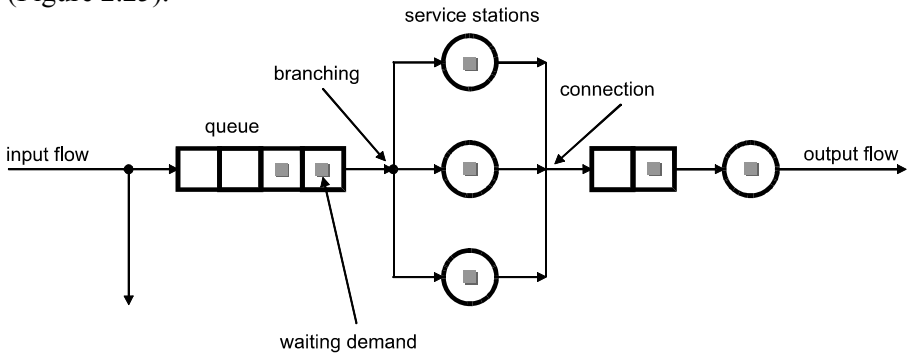


Figure 2.25. Example of a queuing system

The demand represents a very complex entity in queuing theory that is not always easy to interpret and that can best be compared with a product or produce in the real manufacturing. However, it is also possible to consider

other tasks, such as the repair of a machine, as a demand. Generally, the demand contains a whole complex of tasks to be performed at the various service stations. This work schedule is however not explicitly linked with the demand object itself, but rather contained implicitly in the structure graphs of the queuing system.

A specific demand passes through the queuing system along the previously defined flow edges. When the demand arrives at a service station, it remains for the duration of the servicing in the station. The servicing duration is a property of the station and is generally considered to be a stochastic quantity. Service stations also have a limit capacity that specifies how many demands can be served concurrently. The following section assumes that this limit capacity always has the value 1. There are two possibilities for demands that arrive at an occupied station:

1. The demand waits in front of the station in a specially provided queue.
2. The demand will be rejected and diverted to a so-called loss flow when the acceptance capacity of the queue is reached.

Systems for which loss flows can occur are also designated as loss systems. Systems that do not accept or issue any external demands are called closed queuing systems. The number of demands is constant in closed systems. Such systems can be used for such things as the modelling of the manufacturing control.

2.4.1 Demand Flows

Because the demand flows form the backbone of the queuing theory, we will consider them in more detail here. We will make the basic assumption that a demand e_k arrives at the system at some random time (date) T_k . The dates must generally obey the relation: $T_k \leq T_{k+1}$. Because the arrival of a demand is normally equivalent to an event, the terms demand and event are used synonymously. The term event flows is often used instead of demand flows.

The system analysis is greatly simplified when simultaneous events are excluded, namely $T_k < T_{k+1}$ is assumed. This is no significant restriction from the practical point of view, because the refinement of the time grid means simultaneous events can always be transformed into sequential events. Event flows with this property are also called ordinary event flows. As for the event times themselves, the demands are linearly ordered and form the sequence $\{e_k\}$.

The dates can be interpreted as being a physical time when they relate to a common start date. The date T_0 of the first demand e_0 is a natural candidate here. Thus, the subsequent discussion will not differentiate

between the random event date T_k and the random physical event time $T_k - T_0$.

The event flow is characterised by the so-called intermediate arrival time $\Delta T_k = T_k - T_{k-1}$ that results from the time difference between two consecutive events. The intermediate arrival time is a random quantity whose distribution function largely determines the properties of the event flow.

As Figure 2.26 shows, there are several equivalent representations for event flows:

1. sequence of event times (dates) $\{T_k\} = T_0 < T_1 < \dots < T_{k-1} < T_k < \dots$
2. sequence of intermediate arrival times $\Delta T_k = T_k - T_{k-1}$ with $k \geq 1$
3. continuously increasing counting process $N(t) = k$ for $T_k \leq t < T_{k+1}$

The intensity λ can be defined for a stationary event flow that specifies the average number of arriving demands until time t .

$$\lambda = \frac{E(N(t))}{t} \quad (2.65)$$

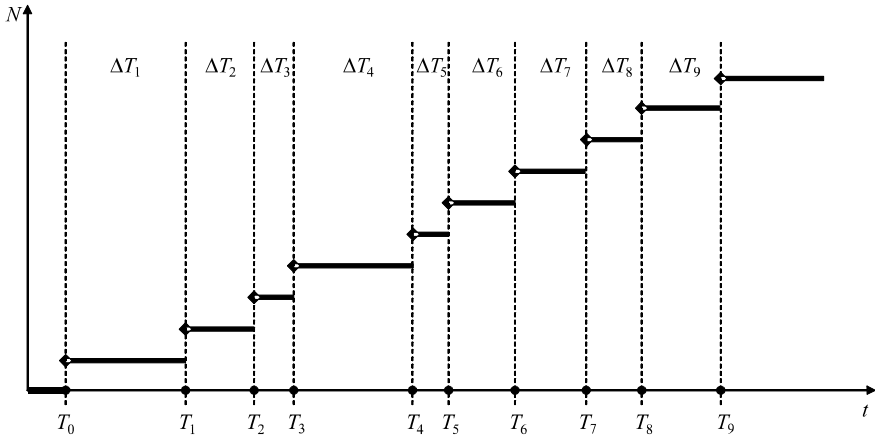


Figure 2.26. Representation of event currents

Event currents whose intermediate arrival times ΔT_k are independent exponentially distributed random values play a special role in queuing theory. In this case: $P(\Delta T_k > \Delta t) = e^{-\alpha \Delta t}$. The growths $\Delta N(t) = N(t + \Delta t) - N(t)$ are then also independent random quantities. However, because they obey the Poisson distribution, these flows are also designated as Poisson flows:

$$P(\Delta N(t) = k) = \frac{(\alpha \cdot \Delta t)^k \cdot e^{-\alpha \cdot \Delta t}}{k!} \quad (2.66)$$

The Poisson flows have properties that significantly simplify the mathematical analysis. To understand this, we must first consider the exponential distribution and its special properties.

A continuous random value X has an exponential distribution when the following equation governs its distribution density:

$$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ \alpha \cdot e^{-\alpha x} & \text{for } x \geq 0 \end{cases} \quad (2.67)$$

Figure 2.27 shows the associated distribution function $F(x)$, for example, for three parameters: $\alpha = 2$, $\alpha = 1$ and $\alpha = 0.5$. Exponentially distributed random values produce small values particularly frequently. The larger an observed value is, the more seldom it occurs. Negative values are not created. If all values of an observation series are assigned to interval classes with width 0.2, this produces for $\alpha = 1$ the histogram shown in Figure 2.28.

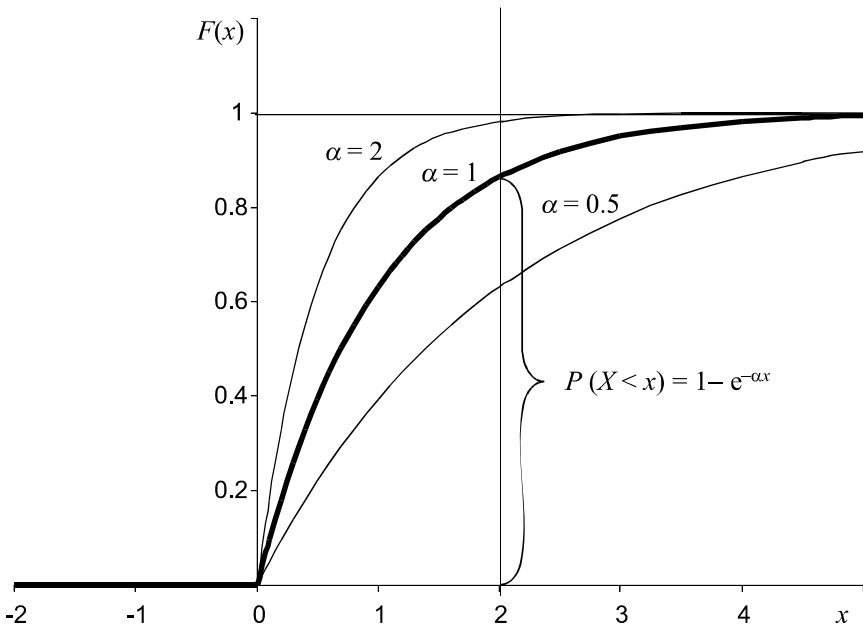


Figure 2.27. $F(x)$ distribution function of the exponential distribution (various values of parameter α)

The probability that a value x lies in the interval $[0; 0.2)$ is approximately 0.18. However, in the interval $[1.8; 2.0)$ it has already fallen to less than

0.03. The following equation governs the expected value and the deviation of the exponential distribution:

$$E(X) = \sqrt{D^2(X)} = \frac{1}{\alpha} \quad (2.68)$$

The intensity of the Poisson event flow can be determined from the counting process:

$$E(N(t)) = \sum_{k=1}^{\infty} k \cdot P(N(t) = k) \quad (2.69)$$

The intensity $\lambda = \alpha$ is obtained from Equation (1.36). Namely, the intensity of the Poisson event flow is identical to the reciprocated expected value of the intermediate arrival times.

However, another property, in particular, exclusively characterises the exponential distribution and thus the Poisson event flows and makes it important for queuing theory. When one queries the probability that $X \geq y + x$ subject to the condition that $X \geq y$ applies, this produces the following equation:

$$P(X \geq y + x | X \geq y) = P(X \geq x) \quad (2.70)$$

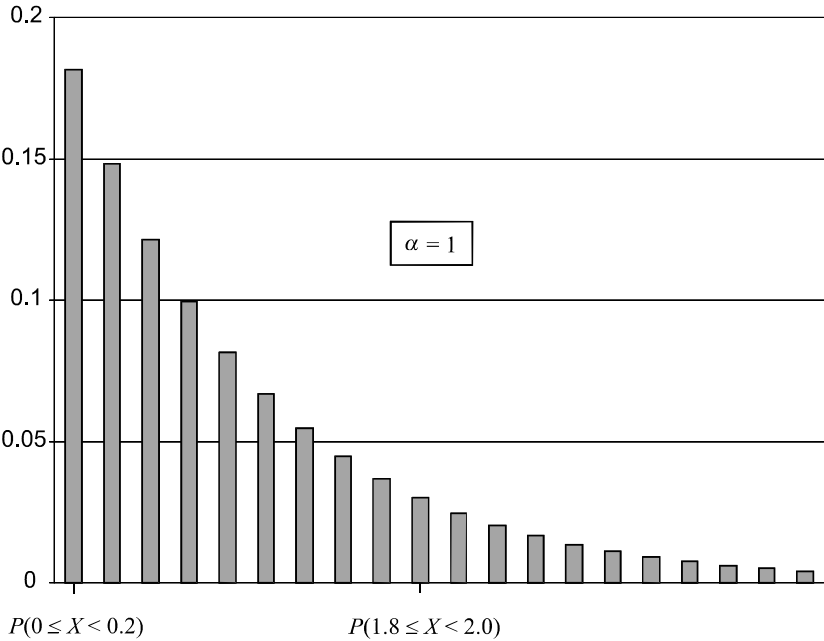


Figure 2.28. Histogram of an exponentially distributed random value for $\alpha = 1$

This surprising property is easy to prove:

$$P(X \geq y + x | X \geq y) = \frac{P((X \geq y + x) \wedge (X \geq y))}{P(X \geq y)} = \frac{P(X \geq y + x)}{P(X \geq y)} \quad (2.71)$$

When $P(X \geq x) = e^{-\alpha x}$, then:

$$P(X \geq y + x | X \geq y) = \frac{e^{-\alpha(y+x)}}{e^{-\alpha y}} = e^{-\alpha x} = P(X \geq x) \quad (2.72)$$

Transferred to the exponentially distributed intermediate arrival times of a Poisson event flow, this means: it is assumed that an event e_{k-1} has arrived by time T_{k-1} . Although the flow since this time was then observed constantly during the duration Δt_{k1} , the subsequent event e_k has still not yet arrived. The probability that the remaining wait time is greater than or equal to Δt_{k2} does not change, irrespective of how far the previous event occurred in the past. This independency of the remaining observation time of the previously elapsed observation time often conflicts with our experience.

Example 1:

The failure of an incandescent lamp becomes more probable with increasing operation duration.

Example 2:

The remaining time until the arrival of the long-expected bus should be shorter, the longer that the passenger has already waited at the bus stop. No Poisson event flow is apparently present in all these cases.

This can be deduced using a simple consideration without needing to subject the event flow and the distribution of its intermediate arrival times to a comprehensive statistical test. A counterexample, however, can be easily found.

Example 3:

It is assumed that a lottery participant regularly makes his bet every weekend. The probability that the time until he makes a large win is longer than one year is P_1 . This probability is usually very high and it is easy to appreciate that it does not change after a year.

The cause for the independency of the remaining wait time (waiting for the win) is the number-picking machine. The individual draws are completely independent. The drawing from the previous week has no effect on the new draw. If this was not the case, you could make use of this property to increase your chance of winning, which certainly is not in the interest of the lottery company.

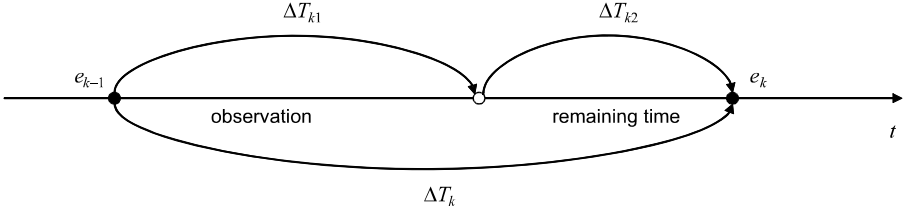


Figure 2.29. Observation duration and the remaining observation duration

The examples from daily life show how simple considerations can often be used to test whether or not Poisson event flows are involved. These considerations also clarify the nature of Poisson flows. Not only are the intermediate arrival times independent random variables, but the individual events arrive independently and are not coupled with each other with any internal structure other than their membership of the same event flow.

The exponential distribution has in the queuing theory a comparable role as the normal distribution has in mathematical statistics. In both cases, the reasons lie in the independency of the events or effects and in the possible simplification of the associated equation apparatus.

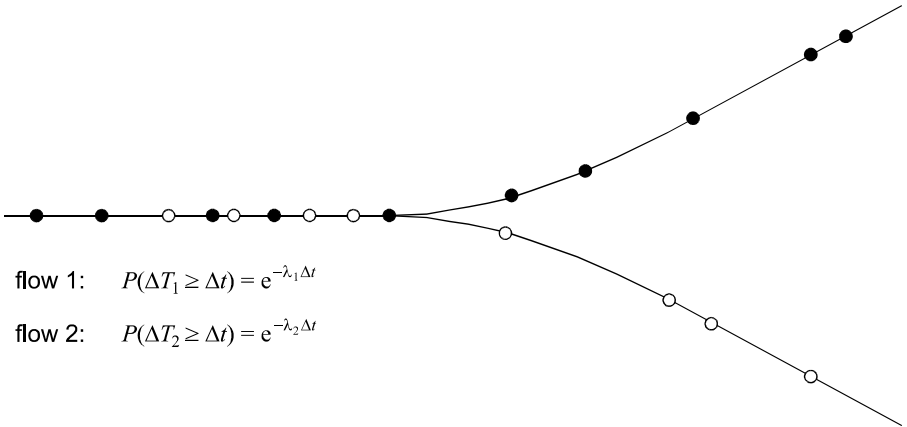


Figure 2.30. Overlaying of Poisson event flows

Figure 2.30 shows two event flows that are separated or combined. The following general equations apply to the complete flow

$$P(\Delta T \geq \Delta t) = P((\Delta T_1 \geq \Delta t) \wedge (\Delta T_2 \geq \Delta t)) \quad (2.73)$$

or for independent flows

$$P(\Delta T \geq \Delta t) = P(\Delta T_1 \geq \Delta t) \cdot P(\Delta T_2 \geq \Delta t) \quad (2.74)$$

For the Poisson event flows case, in particular:

$$P(\Delta T \geq \Delta t) = e^{-\lambda_1 \cdot \Delta t} \cdot e^{-\lambda_2 \cdot \Delta t} = e^{-(\lambda_1 + \lambda_2) \cdot \Delta t} \quad (2.75)$$

Equation (2.75) shows that separated or combined Poisson event flows are themselves Poisson flows [6].

2.4.2 Classification of Queuing Systems

The designation key introduced by the English mathematician D. G. Kendall is generally used for the characterisation of queuing systems [2]. The key contains the character of the event flow, the property of the service station, the number of service stations and the length of the queues; the individual items are usually separated with a slash. These designations are found very often in modified form in the literature (see Table 2.1).

Systems with general distribution, multiple service stations m and limited queue length n : $G/G/m/n$ represent some of the largest challenges to the queuing theory. However, only the simpler systems with exponential distribution are investigated in the following section.

Table 2.1. Classification of queuing systems

	Explanation
M	Markov – exponentially distributed intermediate arrival times or serving times (Poisson event flow)
D	Deterministic – deterministic intermediate arrival times or serving times
GI	General Independent – The intermediate arrival times or the serving times are independent of each other. In addition, no demands are placed on the distribution function.
G	General – as GI, the interdependency between the random variables is however not demanded.

2.4.3 Loss System M/M/1/0

The M/M/1/0 system is the simplest possible loss system. It consists of just a single service station (machine) and does not have any queue. If the service station is free, the arriving demand will be processed immediately, otherwise it will be rejected. Rejected demands will be lost and form the loss flow. Without doubt, the best-known loss system is the telephone. If the dialled number is busy, one must hang up and redial later. However, even manufacturing systems contain substructures that can be described by a simple loss system.

Example:

A machine group contains a preferred machine that processes all incoming jobs. If the machine is occupied, incoming jobs will be diverted to the other machines. The problem situations could be described as: What is the probability that a job will not be processed on the preferred machine? How high is the loading on the preferred machine?

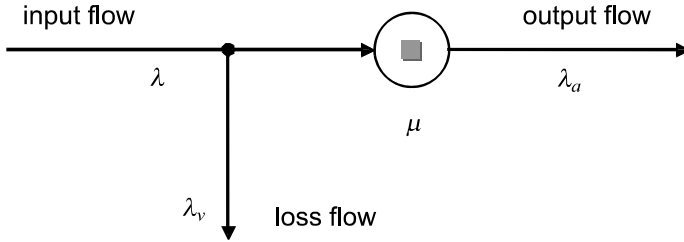


Figure 2.31. M/M/1/0 loss system

The following section assumes a stationary Poisson input flow with intensity λ . The intermediate arrival time of two successive events is T_A . Simultaneous events are ignored or excluded. The service station can always process just one demand for which it requires time T_B . Both the servicing time and the intermediate arrival time have an exponential distribution with servicing intensity μ .

$$P(T_A \geq \Delta t) = e^{-\lambda \cdot \Delta t} \quad (2.76)$$

$$P(T_B \geq \Delta t) = e^{-\mu \cdot \Delta t}$$

The loss system knows two discrete states. The task involves determining the state probabilities for an arbitrary time t .

State	Probability	
Z_0	$P(Z_0, t) = p_0(t)$	System is empty
Z_1	$P(Z_1, t) = p_1(t)$	System is busy

The probability that the system within the time interval Δt switches from state Z_i to state Z_j is the transition probability $p_{i,j}(\Delta t)$.

$$p_{i,j}(\Delta t) = P((Z_j, t + \Delta t) | (Z_i, t)) \quad (2.77)$$

This allows the state probabilities $p_i(t + \Delta t)$ to be calculated depending on the state probabilities at time t as follows.

$$p_0(t + \Delta t) = p_0(t) \cdot p_{0,0}(\Delta t) + p_1(t) \cdot p_{1,0}(\Delta t) \quad (2.78)$$

$$p_1(t + \Delta t) = p_0(t) \cdot p_{0,1}(\Delta t) + p_1(t) \cdot p_{1,1}(\Delta t)$$

The state graph is shown in Figure 2.32. Systems for which the following state is determined exclusively by the previous state are also called Markov systems.

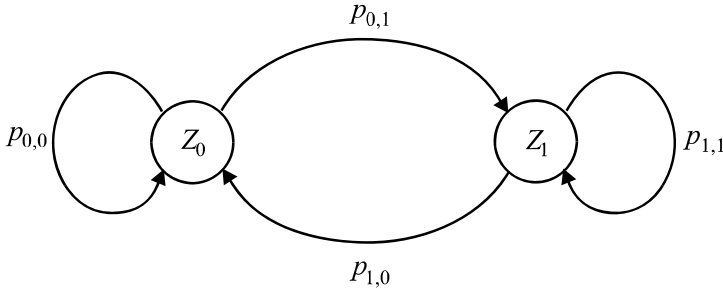


Figure 2.32. State graph of the M/M/1/0 loss system

This yields for the transition probabilities in a time interval Δt :

$$p_{0,0}(\Delta t) = P(T_A \geq \Delta t) = e^{-\lambda \cdot \Delta t} \quad (2.79)$$

$$p_{1,0}(\Delta t) = P(T_B < \Delta t) = 1 - e^{-\mu \cdot \Delta t}$$

$$p_{0,1}(\Delta t) = P(T_A < \Delta t) = 1 - e^{-\lambda \cdot \Delta t}$$

$$p_{1,1}(\Delta t) = P(T_B \geq \Delta t) = e^{-\mu \cdot \Delta t}$$

One can now generate the individual functions in Taylor series and so obtain the probability polynomials in Δt for the transition. All elements with powers greater than 1 have each been combined in a remainder element $o(\Delta t)$ in Equation (2.80).

$$p_{0,0}(\Delta t) = 1 - \lambda \cdot \Delta t + o(\Delta t) \quad (2.80)$$

$$p_{1,0}(\Delta t) = \mu \cdot \Delta t + o(\Delta t)$$

$$p_{0,1}(\Delta t) = \lambda \cdot \Delta t + o(\Delta t)$$

$$p_{1,1}(\Delta t) = 1 - \mu \cdot \Delta t + o(\Delta t)$$

Now setting the transition probabilities in Equation (2.78) yields the differential equation (2.81) for the state probabilities.

$$\begin{aligned} \frac{p_0(t + \Delta t) - p_0(t)}{\Delta t} &= -\lambda \cdot p_0(t) + \mu \cdot p_1(t) + \frac{o(\Delta t)}{\Delta t} \\ \frac{p_1(t + \Delta t) - p_1(t)}{\Delta t} &= \lambda \cdot p_0(t) - \mu \cdot p_1(t) + \frac{o(\Delta t)}{\Delta t} \end{aligned} \quad (2.81)$$

Because the term $o(\Delta t)$ contains only powers of Δt^2 and higher, the quotient $o(\Delta t)/\Delta t$ vanishes for the limit transition $\Delta t \rightarrow 0$ in both equations. The result is the differential equations for the state probabilities.

$$\dot{p}_0(t) = -\lambda \cdot p_0(t) + \mu \cdot p_1(t) \quad (2.82)$$

$$\dot{p}_1(t) = \lambda \cdot p_0(t) - \mu \cdot p_1(t)$$

Because the loss system assumes just one of the two states p_0 or p_1 at any time:

$$p_0(t) + p_1(t) = 1 \quad (2.83)$$

This allows the system of differential equations (2.82) to be separated into two independent differential equations for p_0 and p_1 .

$$\dot{p}_0(t) = -(\lambda + \mu) \cdot p_0(t) + \mu \quad (2.84)$$

$$\dot{p}_1(t) = -(\lambda + \mu) \cdot p_1(t) + \lambda$$

Equation (2.84) shows linear differential equations of first degree with constant coefficients that can be solved without difficulty. The result depends on the initial conditions. For the case that the loss system is initially empty, namely $p_0(0)=1$, this yields the following state equations.

$$p_0(t) = \frac{\mu}{\lambda + \mu} + \frac{\lambda}{\lambda + \mu} \cdot e^{-(\lambda + \mu)t} \quad (2.85)$$

$$p_1(t) = \frac{\lambda}{\lambda + \mu} - \frac{\lambda}{\lambda + \mu} \cdot e^{-(\lambda + \mu)t}$$

Figure 2.33 shows the transient response for $\lambda/\mu = 2$. The quotient from the input intensity λ and servicing intensity μ is also designated as the *traffic value* for which the Greek letter ρ is used,

$$\rho = \frac{\lambda}{\mu} \quad (2.86)$$

The traffic value is a measure for the service-station loading. A high traffic value also means a high relative loading and consequently more demands enter the loss flow. The loss system quickly stabilises itself to the stationary end state. This can be calculated with the traffic value as follows:

$$p_0(t = \infty) = \frac{1}{1 + \rho} \quad (2.87)$$

$$p_1(t = \infty) = \frac{\rho}{1 + \rho}$$

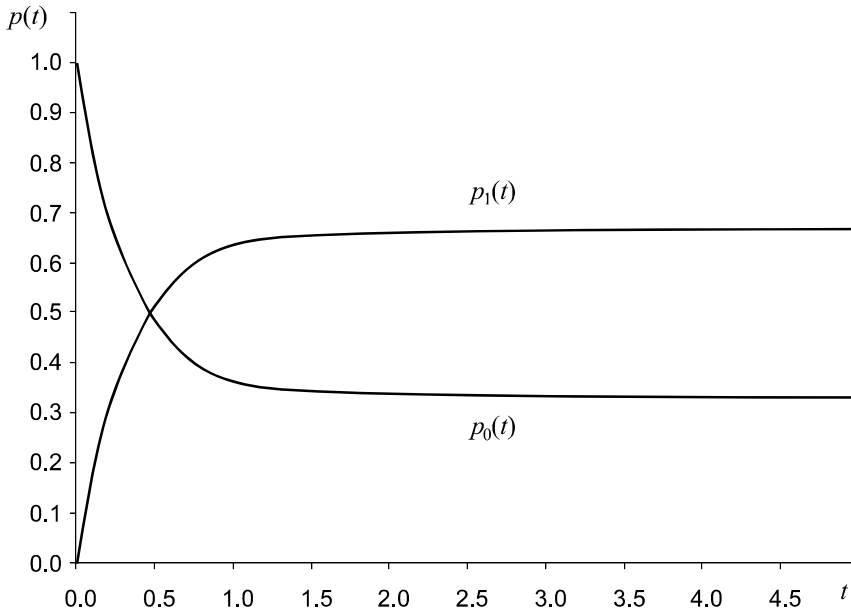


Figure 2.33. Transient response of the M/M/1/0 loss system for $\rho = 2$

Although the loss system is comparatively simple, it provides interesting and generally usable results for practical use.

Let us assume that a testing position is occupied by a single person and the duration required for the inspection of a product has an exponential distribution with testing intensity μ . The products to be inspected form a Poisson input flow with intensity λ . A product will be tested only when the testing person has currently nothing to do, otherwise the product will be forwarded. For this type of sample test, one is now interested in the number of tested products compared with the number of untested products.

The testing position separates the product flow into two flows: the flow of tested products with intensity λ_a and the flow of untested products with intensity λ_v . Obviously intensity λ_a cannot be larger than the testing intensity μ . In practice, a capacity consideration is often used for the calculation, for which the output intensity λ_a remains equal to the input intensity λ while $\lambda < \mu$. Afterwards, the output intensity is equal to the testing intensity μ . Figure 2.34 shows the relationship in its general form, in which both the input intensity and the output intensity are shown relative to μ . Formally, in addition to the traffic value ρ at the entry to the system, a traffic value $\rho_a = \lambda_a / \mu$ at the exit is also obtained. The representation is thus normalised and applies independently of the absolute value of the intensities.

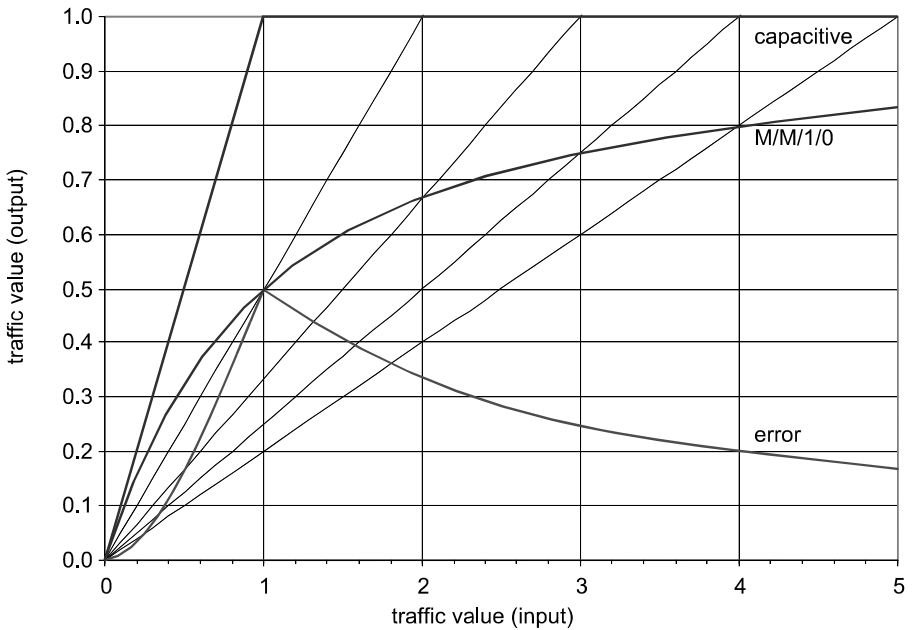


Figure 2.34. Loss system – capacitive and stochastic analysis

Actually, the M/M/1/0 loss system behaves quite differently. Between the simplified capacity consideration and the more accurate calculation using queuing theory there is an error that attains its maximum for the traffic value $\rho = 1$.

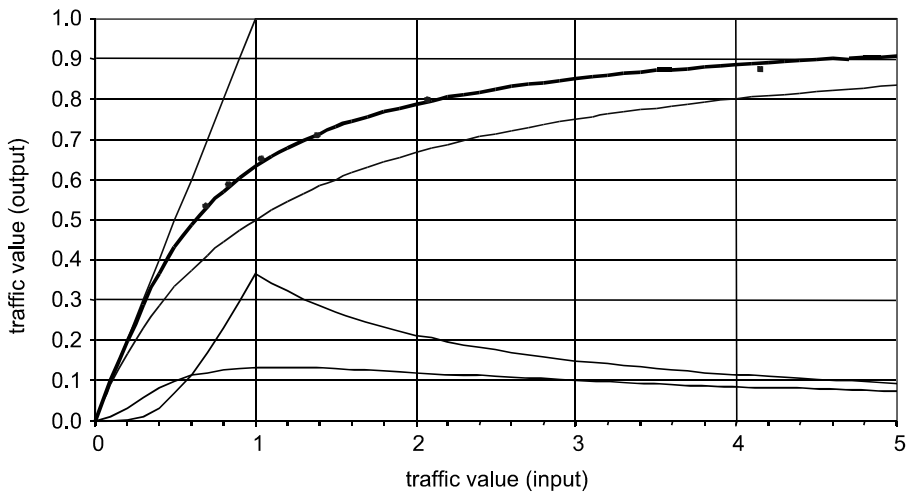


Figure 2.35. Comparison of the D/M/1/0, M/M/0/1 and capacitive model

The error arises from the fact that the simple capacity models do not take into consideration that phases of inactivity also occur for overloading, so that the theoretical intensity limit value can never be achieved. A rough estimate of the capacity is actually only suitable in two cases: for chronic underloading or for extreme overloading of the system. Otherwise, it can be shown that the M/M/1/0 loss system marks a limit curve. The system behaviour of an arbitrary queue system is characterised by a characteristic curve that lies in the area between the curve of the simple loss system as lower limit and the curve of the capacitive model as upper limit.

We use the D/M/1/0 system as an example in which the Poisson input flow has been replaced with a deterministic input flow. The traffic value at the system exit can be calculated as follows:

$$\rho_a = \rho \cdot (1 - e^{-1/\rho}) \quad (2.88)$$

The function $\rho_a = \rho(\rho)$ lies in the expected range. The marked points are the result of a simulation experiment and confirm the calculation. As the error curves below show, both the capacitive model and the M/M/1/0 loss system yield significant deviations.

2.4.4 Systems with Queue

The M/M/1/0 loss system is surely the simplest of all server systems and, in the true sense of the word, not really a queue system. Of more interest for manufacturing processes are systems with queue in which no demands or manufacturing jobs can get “lost”. Systems whose queues have infinite capacity are easier to calculate than those with limited capacity. The M/M/1/ ∞ queue system is considered as a typical representative here.

Figure 2.36 shows a simple server system with an infinite queue and a service station. The intensity of the input flow is λ and the servicing intensity is μ . Both the intermediate arrival times and the serving times are mutually independent and have an exponential distribution.

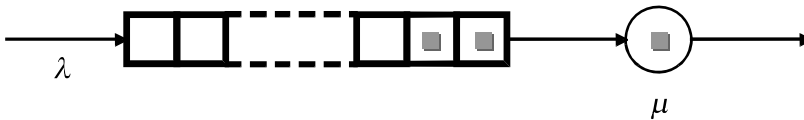


Figure 2.36. Server system with infinite queue

The system can, depending on the maximum permitted queue length n , assume exactly $n+1$ discrete states, which are defined as follows:

State	Probability	
Z_{n-1}	$P(Z_{n-1}, t) = p_{n-1}(t)$	System contains $n - 1$ demands
Z_n	$P(Z_n, t) = p_n(t)$	System contains n demands
Z_{n+1}	$P(Z_{n+1}, t) = p_{n+1}(t)$	System contains $n + 1$ demands

As for the simple loss system, the demand in the service station is also counted. Figure 2.37 shows a section of the state graph that can be arbitrarily extrapolated at the left and the right.

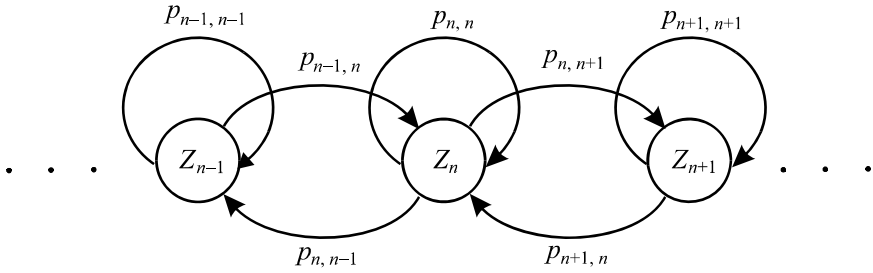


Figure 2.37. State graph of the M/M/1/∞ queue system (section)

Assuming that simultaneous events do not occur, a time interval Δt can always be found in which just one and only one state change takes place. Using the limit transition $\Delta t \rightarrow 0$ described later, the probability for the Z_n state can be calculated from the probabilities of the two neighbouring states Z_{n-1} and Z_{n+1} and the associated transition probabilities.

$$p_n(t + \Delta t) = p_{n-1}(t) \cdot p_{n-1, n}(\Delta t) + p_n(t) \cdot p_{n, n}(\Delta t) + p_{n+1}(t) \cdot p_{n+1, n}(\Delta t) \quad (2.89)$$

The transition probabilities in the time interval Δt yields:

$$\begin{aligned} p_{n-1, n}(\Delta t) &= P(T_A < \Delta t) \\ p_{n, n}(\Delta t) &= P((T_B \geq \Delta t) \wedge (T_A \geq \Delta t)) \\ p_{n+1, n}(\Delta t) &= P(T_B < \Delta t) \end{aligned} \quad (2.90)$$

Assuming that all random quantities are mutually independent and have an exponential distribution, this yields:

$$\begin{aligned} p_{n-1, n}(\Delta t) &= 1 - e^{-\lambda \cdot \Delta t} = \lambda \cdot \Delta t + o(\Delta t) \\ p_{n, n}(\Delta t) &= e^{-\mu \cdot \Delta t} = (1 - \mu \cdot \Delta t) \cdot (1 - \lambda \cdot \Delta t) + o(\Delta t) \\ p_{n+1, n}(\Delta t) &= 1 - e^{-\mu \cdot \Delta t} = \mu \cdot \Delta t + o(\Delta t) \end{aligned} \quad (2.91)$$

When we now set the transition probabilities in Equation (2.89), after using the limit transition $\Delta t \rightarrow 0$ we obtain the differential equations of the system. The special case $n = 0$ (empty system) can be derived from the simple loss system.

$$\begin{aligned}\dot{p}_n(t) &= \lambda \cdot p_{n-1}(t) - (\mu + \lambda) \cdot p_n(t) + \mu \cdot p_{n+1}(t) \quad \text{for } n > 0 \\ \dot{p}_0(t) &= -\lambda \cdot p_0(t) + \mu \cdot p_1(t)\end{aligned} \quad (2.92)$$

The steady state has a particularly simple solution. The system is stationary in this case, namely, the state probabilities P_n no longer change: $\dot{p}_n(t) = 0$. The system of differential equations (2.92) changes to a system of homogeneous linear equations that can be solved iteratively, which is easy to prove.

$$\begin{aligned}0 &= \lambda \cdot P_{n-1} - (\mu + \lambda) \cdot P_n + \mu \cdot P_{n+1} \quad \text{for } n > 0 \\ 0 &= -\lambda \cdot P_0 + \mu \cdot P_1\end{aligned}$$

First solve for P_1 by setting $n = 0$ in the equation, and then set the result in the equation for $n = 1$, *etc.* The result is a general equation for the state probabilities P_n depending on the probability P_0 .

$$P_n = \rho^n \cdot P_0 \quad (2.93)$$

Once again, the traffic value ρ plays a decisive role here, although note that Equation (2.93) is true only under the condition that the traffic value is less than 1. Otherwise the queue would grow continually and thus no stationary state could occur. The state probability P_0 can be easily eliminated using the total probability and the partial sum for the geometric series.

$$1 = \sum_{n=0}^{\infty} P_n = P_0 \cdot \sum_{n=0}^{\infty} \rho^n = P_0 \cdot \frac{1}{1 - \rho} \quad (2.94)$$

The result is finally a usable equation for the state probability P_n that depends only on the traffic value ($\rho < 1$).

$$P_n = \rho^n (1 - \rho) \quad (2.95)$$

The practitioner is normally less interested in the actual state probability than the statistical characteristic values of the systems. If, for example, the average queue length is known \bar{n} , one has an important starting point for the dimensioning of the buffer storage.

$$\bar{n} = \sum_{n=0}^{\infty} n \cdot P_n = \frac{\rho}{1 - \rho} \quad (2.96)$$

Based on the differential equations that describe the system, obviously a number of further characteristic values, such as the average production time of a job depending on the time, the loading level of the service stations or the average wait time for manufacturing jobs can be calculated. The interested reader should consult the relevant and copious literature for queuing theory. [2–4]

2.5 Petri Nets

The idea of the Petri net is based on work by C. A. Petri, who submitted in 1962 at the Institute for Instrumental Mathematics at the Bonn University a dissertation with the title “Communication with Automats”. The concept developed by Petri is ideally suited for the modelling and analysis of discrete systems, thus also for manufacturing systems. In no other model are such different categories, such as resources and events, linked structurally with each other. Although the model modules are elementary, the mapping possibilities are almost unlimited.

As the name suggests, the theory of the Petri nets is based on graphical theory. Depending on the use, they can be assigned both to analytical models and to the simulation models. Indeed, simulations for manufacturing processes based on Petri nets exist, although these are far less common than those based on queuing theory. The reason may lie in the abstract importance of the system modules that the less experienced user cannot always easily grasp. The relationship between model and reality is less apparent than, for example, for the queue models.

2.5.1 Definition

Petri nets are directed graphs, which, in contrast to ordinary graphs, have two different classes of nodes:

- Places
- Transitions

It is assumed that both the set of places $S = \{s_1, s_2, s_3, \dots\}$ and the set of the transitions $T = \{t_1, t_2, t_3, \dots\}$ are nonempty, namely, the Petri net contains always both – transitions and places. The set of directed edges is defined by a *flow relation* F as follows:

$$F \subseteq S \times T \cup T \times S \quad (2.97)$$

Namely, an edge in the graph always connects a place with a transition or *vice versa*, never, however, places or transitions with each other. Using the

mappings K and W , a natural number greater than 0 is assigned to each place s_i and each transition t_j of the Petri net:

$$\begin{aligned} K: S &\rightarrow N \setminus \{0\} \\ W: F &\rightarrow N \setminus \{0\} \end{aligned} \quad (2.98)$$

The value $K(s_i)$ is called the capacity of the place s_i and $W(s_i, t_j)$ is the weight of the edge from s_i to t_j . The importance of the capacity for a place is that it specifies the maximum number of markings that this place can accept. When one forms a relation between the places and the resources of the abstract manufacturing system from Section 1.2, the markings represent the number of references to the corresponding resource element. Markings form the dynamic element of the Petri net. Driven by the transitions, they move from place to place along and in the direction of the edges. Petri nets, however, differ from flow graphs through the absence of a conservation set. Markings are immaterial objects that may suddenly disappear or appear.

Preset and postset

A Petri net has $x, y \in S \cup T$ nodes. The set of all nodes from which edges lead to the nodes x is called the *preset* for x :

$$\bullet x = \{y \mid (y, x) \in F\} \quad (2.99)$$

Correspondingly, the set of all nodes to which the edges leaving the nodes x lead is called the *postset* for x :

$$x\bullet = \{y \mid (x, y) \in F\} \quad (2.100)$$

If a transition t_j fires, markings will be removed from every place of the preset and markings added to every place of the postset. The number of markings moved here corresponds exactly to the weight of associated edges. The transition itself does not decide when a transition fires, but the initiating event must arrive externally. Thus, the transitions indicate only the points in the system at which the events can act. The condition under which a transition activates is determined by the marking assignment of the preset and postset of the transition. Any missing markings in the preset or missing capacity in the postset hinder the activation of the transition. Places consequently perform the function of conditions under which the determined events can act and represent the state of the system.

State and marking

The initial state M_0 of a Petri net is determined by its marking – a mapping that each place s_i shows a number $M_0(s_i)$ of markings,

$$M_0: S \rightarrow N \quad \text{with} \quad 0 \leq M_0(s_i) \leq K(s_i) \quad (2.101)$$

A Petri net can be represented as a graph where the places and the transitions are indicated as a circle and as a rectangle (or often simply as a bar), respectively. In addition to the edge evaluation and the capacity, the marking of the places is entered. Appropriate simulation systems also allow the animation of the marking flow with the significant increase in the clarity of the method. Figure 2.38 shows a simple example of a Petri net that consists of two places and transitions. Whereas the place s_1 has a marking, the place s_2 is empty.

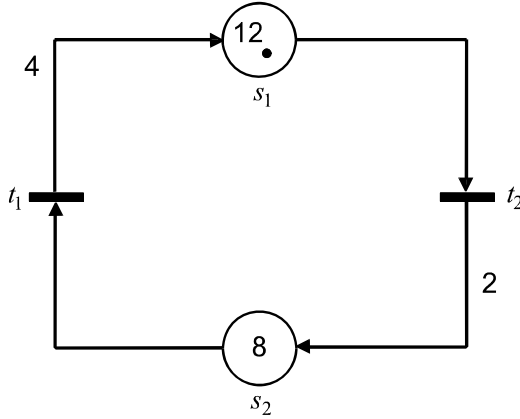


Figure 2.38. Example of a simple Petri net

Vector notation

Let us assume that the Petri net has $|S|$ places, then a *capacity vector* \mathbf{k} and a *marking vector* \mathbf{m}_0 can be defined as follows:

$$\mathbf{k} = \begin{pmatrix} K(s_1) \\ K(s_2) \\ \vdots \\ K(s_{|S|}) \end{pmatrix} \quad \text{and} \quad \mathbf{m}_0 = \begin{pmatrix} M_0(s_1) \\ M_0(s_2) \\ \vdots \\ M_0(s_{|S|}) \end{pmatrix} \quad (2.102)$$

Similarly, for each transition $t_j \in T$ a vector \mathbf{t}_j with $|S|$ components can also be defined in which the component $t_{j,i}$ specifies how many markings the transition t_j saves at the place s_i (positive value) or takes from the place s_i (negative value).

$$t_{j,i} = \begin{cases} -W(s_i, t_j) & \text{if } (s_i, t_j) \in F \\ +W(t_j, s_i) & \text{if } (t_j, s_i) \in F \\ 0 & \text{else} \end{cases} \quad (2.103)$$

$$\mathbf{t}_j = \begin{pmatrix} t_{j,1} \\ t_{j,2} \\ \vdots \\ t_{j,|S|} \end{pmatrix} \quad (2.104)$$

A *net matrix* (incidence matrix) \mathbf{N} that describes the graph structure then can be created from the transition vectors.

$$\mathbf{N} = (\mathbf{t}_1, \mathbf{t}_2, \dots, \mathbf{t}_{|T|}) \quad (2.105)$$

The net matrix, the capacity vector and the marking vector provide a complete description of a Petri net. For the example from Figure 2.38, then:

$$\mathbf{N} = \begin{pmatrix} 4 & -1 \\ -1 & 2 \end{pmatrix}, \mathbf{k} = \begin{pmatrix} 12 \\ 8 \end{pmatrix}, \mathbf{m}_0 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

Firing rules

A firing transition t_j takes from all places $s_i \in \bullet t_j$ markings each with $W(s_i, t_j)$ and takes in all places $s_k \in t_j \bullet$ markings each with $W(t_j, s_k)$. The so-called strong firing rule says that a transition is only activated when after firing the transition a negative marking assignment is not expected at any place and the capacity of the places is not exceeded anywhere. The firing of a transition in the net changes the marking \mathbf{m} into the marking \mathbf{m}' . This can be formulated elegantly using the vector notation:

$$\mathbf{m}' = \mathbf{m} + \mathbf{t}_j \quad (2.106)$$

A transition is activated only when the following equation applies to all places:

$$0 \leq M'(s_i) \leq K(s_i) \quad \text{with } i = 1 \dots |S| \quad (2.107)$$

If several transitions are fired successively, this is called a *firing sequence*. The firing sequence $\sigma = t_1, t_2, \dots, t_n$ is called usable for the marking M if all transitions of the sequence are activated.

Nets in which all edges and all places have the weight 1 and the capacity, respectively, are called *condition/event nets* or simply *C/E nets*.

Reachable set

A marking M of a Petri net is called reachable when a usable firing sequence σ exists that changes the initial marking M_0 into M . The set of all reachable markings is also called the reachable set $R_N(M_0)$.

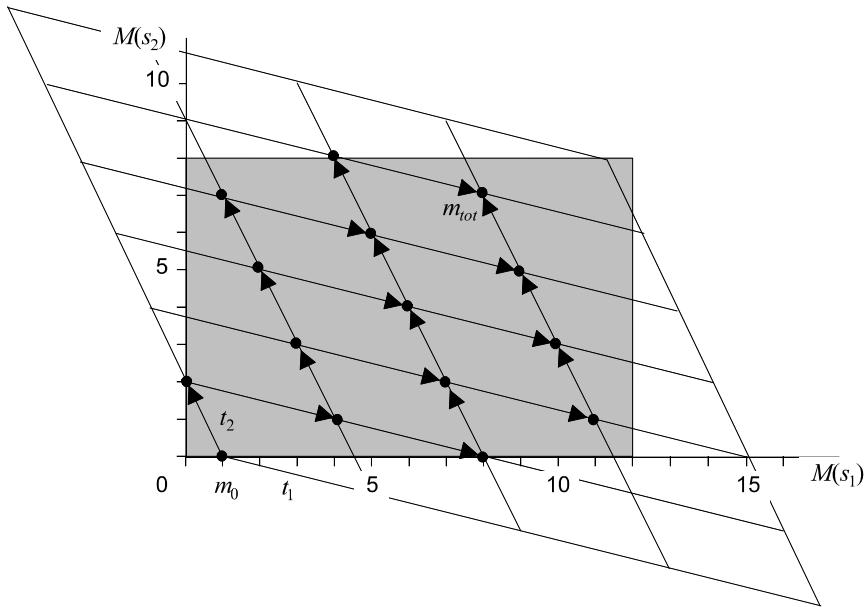


Figure 2.39. Vector representation of the reachable set

The vector representation (see Figure 2.39) can be used to derive the directed reachability graph, when the complete marking of each state, as shown in Figure 2.40, rather than the vector end points, are entered. The reachability graphs can be used to prove the important properties of the Petri net, such as reversibility or liveness. Major parts of the net analysis are based on graph-theory methods. However, in some cases, linear algebra methods can be used. Thus, Petri nets build a bridge between the purely simulative methods and the analytical methods.

Petri nets in the previously described form do not have any time properties, but restrict themselves only to events and the resulting system states. However, such nets can provide important conclusions about the behaviour of manufacturing systems, in particular, with regard to their control and the use of shared resources. The net analysis here concentrates on the reproducibility of system states and on the detection or avoidance of system deadlocks.

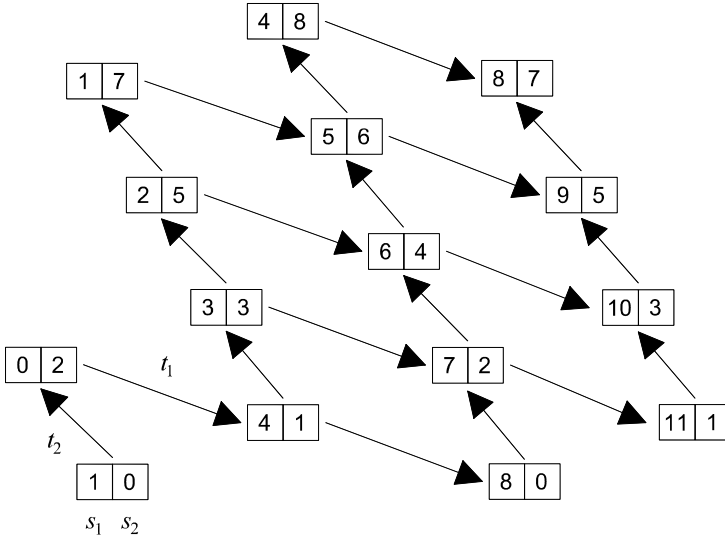


Figure 2.40. Reachability graph

Reversibility

A Petri net is reversible when:

$$\forall M_1, M_2 \in R_N(M_0): M_1 \in R_N(M_2) \quad (2.108)$$

Any system state can be reproduced in reversible Petri nets. In contrast, a marking M_{tot} , from which no transition is activated, indicates a *total deadlock*. If a subset of the transitions still remains activated, this is called a *partial deadlock*.

The simple Petri net from Figure 2.38 shows a total deadlock for the marking (8,7). Generally, deadlocks in manufacturing systems have severe consequences. Their correction requires intervention in the execution control with the consequent significant potential for malfunction. The early detection and avoidance of such system states is thus very important.

Liveness

A transition $t_j \in T$ is called living when at any marking $M_1 \in R_N(M_0)$ always just one firing sequence σ exists that leads to a marking $M_2 \in R_N(M_0)$ for which t_j is activated. A living network thus describes systems in which none of the previously defined events can be excluded long term. With regard to manufacturing systems, reversibility and liveness of the net are fundamental requirements that the system design must meet.

2.5.2 Example

The modelling of a manufacturing system with a Petri net requires a very high degree of abstraction. This may well be one of the reasons why Petri nets are found comparatively seldom in industrial practice. If, however, the abstraction hurdle is tackled, this provides a very powerful instrument for the analysis, simulation and, to a limited extent, also for the calculation of discrete, event-driven systems. Furthermore, the Petri nets provide an insight into the structure of the interprocess communication, with a clarity that no other model concept permits. The following example from electronics production shows how a simple C/E net can be used to model a manufacturing process and visualise any conflict potential.

A job-lot manufacturer for electronic printed-circuit boards produces every day a large number of lots for different module types. Two SMT lines, each with several placement machines, are available for placing components on the circuit boards. The module types A and B are often produced in parallel, with type A on line 1 and type B on line 2. Both A and B require two setup groups, SG 1 and SG 2, which, however, are each present just once. A setup group is, for example, a feeder carriage on which a specific range of electronic components can be stored in belts and, when required, coupled to an automatic placement machine. A manufacturing batch is started autonomously by the two independent line controllers. The production can be started only when both setup groups, SG 1 and SG 2, are available. We also assume for the control algorithm that first the setup group SG 1 and then SG 2 is required. If either of the setup groups is busy, the lot will be suspended until the missing setup group becomes free. The following tables summarise the principle conditions and events of the manufacturing process.

Table 2.2. Conditions for the manufacturing process

Condition (place)	Description	Condition (place)	Description
s_1	Lot A reported at line 1	s_5	SG 1 made available for line 2
s_2	SG 1 made available for line 1	s_6	SG 2 made available for line 2
s_3	SG 2 made available for line 1	s_7	SG 1 free
s_4	Lot B reported at line 2	s_8	SG 2 free

Table 2.3. Events of the manufacturing process

Event (transition)	Description	Event (transition)	Description
t_1	Request SG 1 for line 1	t_4	Request SG 1 for line 2
t_2	Request SG 2 for line 1	t_5	Request SG 2 for line 2
t_3	Type A completed on line 1	t_6	Type B completed on line 2

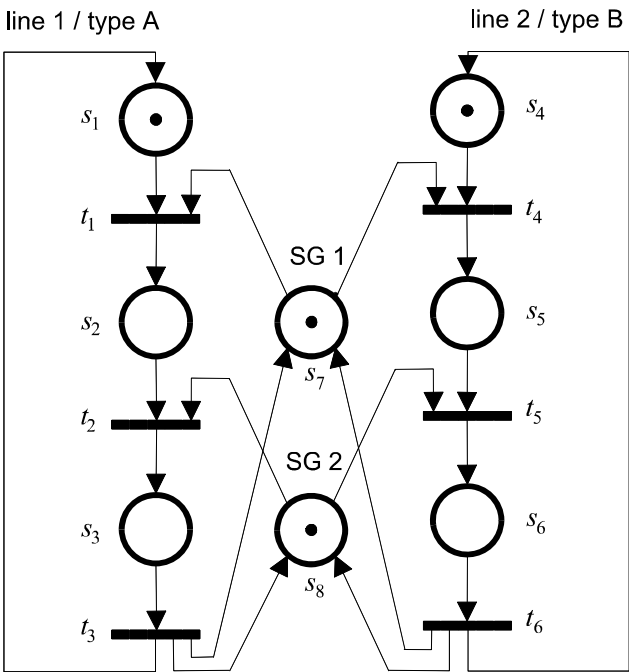


Figure 2.41. Petri net of the manufacturing system

Figure 2.41 shows the associated Petri net. The figure clearly shows the two independent processes on line 1 and line 2, and the setup groups SG 1 and SG 2 as shared resources.

The manufacturing process is simulated by the sequential activation of the individual transitions. This enables the marking flow in the Petri net. The reachability graphs from Figure 2.42 show that the two concurrent processes mutually exclude each other because of the shared use of the setup groups. Although the process that starts later must wait, the network is living.

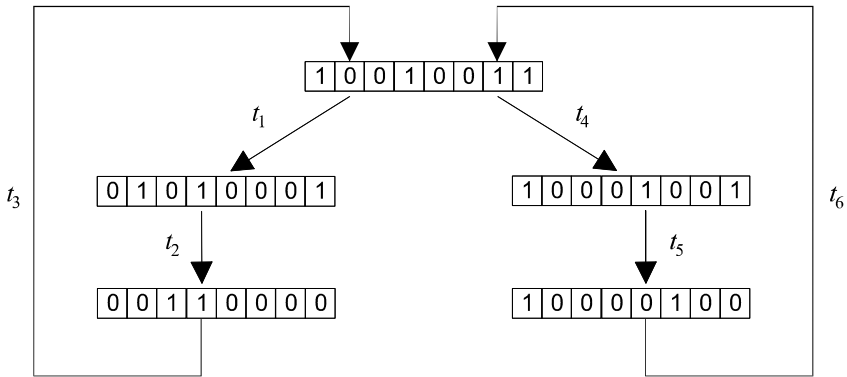


Figure 2.42. Reachability graph

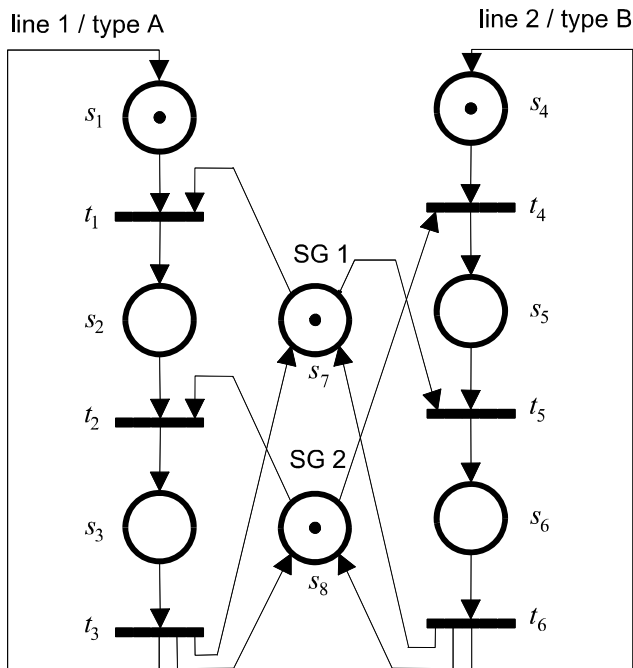


Figure 2.43. Petri net with changed strategy for the setup group request

Figure 2.43 shows a slightly changed net for the same manufacturing process. The only difference here is that line 2 first requests the setup group SG 2 and only then the setup group SG 1. This has the consequence that both processes can enter a start state and so no longer guarantee that they exclude each other.

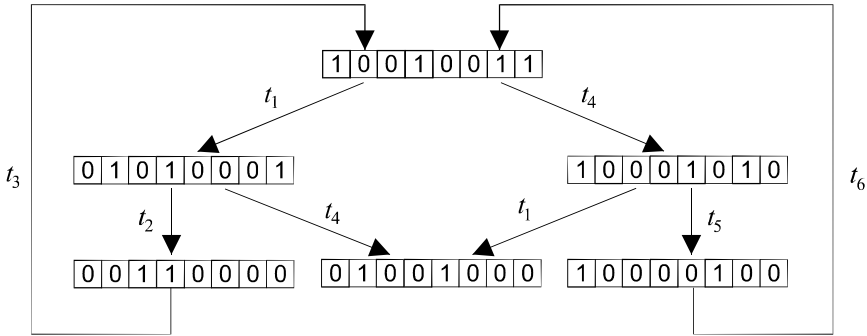


Figure 2.44. Reachability graph with deadlock

The associated reachability graph from Figure 2.44 shows that the system has a deadlock. There are two possible firing sequences that lead to a deadlock: t_1, t_4 and t_4, t_1 . In both cases, each of the two processes has one setup group and waits for the other. Because neither of the two starting processes can be returned to its original state, the system cannot leave this state.

2.5.3 Elementary Links in Petri Nets

The example from Section 2.5.2 clearly shows the nature of the Petri nets. The manufacturing system is considered to be an environment in which several manufacturing processes can be embedded. In the example, these are the processes A and B that have a relationship with the circuit-board types A and B. In contrast, the SMT lines 1 and 2 can be considered as being only passive resources that are periodically required by the active processes. Both processes are controlled autonomously. Depending on the customer demand or the manufacturing manufacturer's schedule, they could be active successively or simultaneously. This form of independency is also known as concurrent processes. If we assume there was no conflict with the setup groups, both processes would be completely asynchronous and the Petri net would divide into two separate subnets. Only the use of shared resources provides a contact area between the two processes that partially cancels the asynchronous operation. The example net shown in Figure 2.41 is a classic alternative brought about by the mutual exclusion from critical process sections. Both processes compete for the same resources. The process granted the resource first suspends its competitor by the withholding of markings. The Petri net means *a priori* no organisational sequence is forced for the shared resources, rather, it is possible that either A or B may be activated successively several times or suspended for a longer period of time.

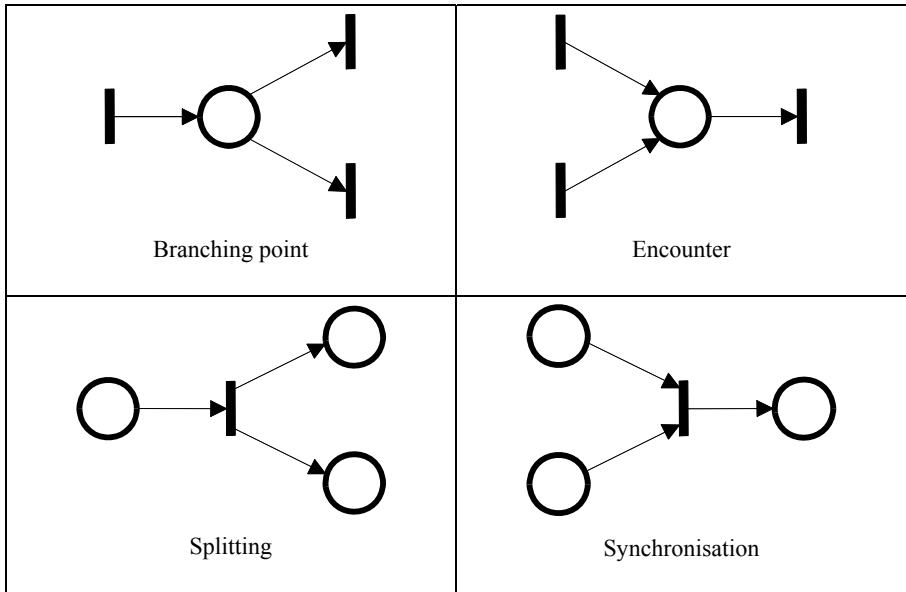


Figure 2.45. Elementary links in Petri nets

A characteristic of the example is that just one process can be active in the critical section. The manufacturing engineering obviously knows other forms of the process communication as the overview in Figure 2.45 shows. Branching point and encounter describe alternative processes. If we assume a simple C/E net, a branch produces a single process from n possible processes. An encounter links n processes to produce a single process so that the previously shared events are now performed sequentially. Branches are required, for example, to describe the distribution of products on the individual machines of a machine group. A company's central shipping department can be represented by an encounter.

As difference to the branching point, the splitting and the synchronisation are true concurrency. The splitting of a process is often found as lot splitting. In contrast to the branching point where the process instance is retained and only used for planning the use of the resources, the splitting creates new processes. Consequently, the parts of a split lot in a real manufacturing process are often managed with their own identification. The synchronisation is one of the most important elementary links of the manufacturing process. Both assembly tasks and all types of bill-of-materials relationships belong to this category. The nature of the synchronisation causes two original concurrent processes to wait for each other after which they then continue together or each separately.

The elementary links, especially, show that Petri nets are particularly suitable for representing the manufacturing process character. This becomes

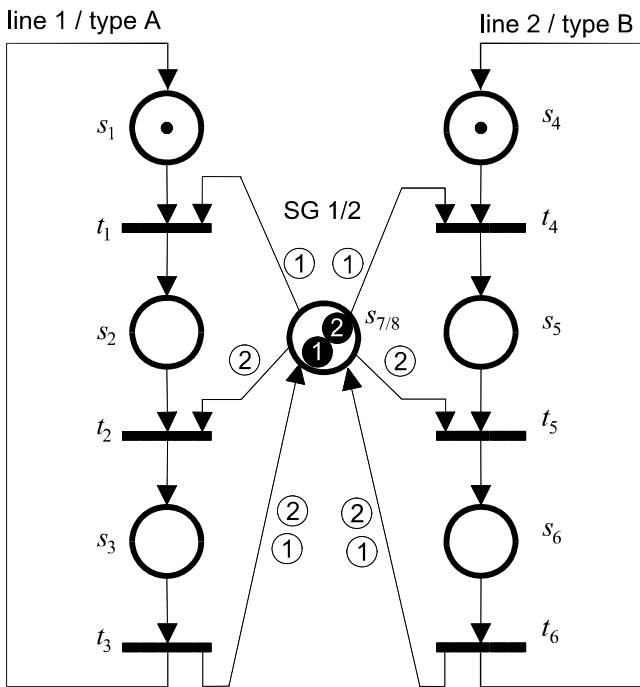


Figure 2.46. Petri net with specific markings

remains to specify which markings flow through the edges. In the example, the transition t_1 requests a marking of type “1” and the transition t_3 puts back a marking of type “1” and “2”. In the same way as the two setup groups were combined, the processing steps of the two lines could be combined. In this case, the places s_1 and s_4 would become the shared place $s_{1/4}$, etc. The different products could be represented using markings of the type “A” and “B”. Compared with normal Petri nets, the CP net requires significantly fewer places. The proximity to the real manufacturing system is also better because some of the specific markings correspond to the various manufacturing jobs [1]. Petri nets provide only information about which transitions are activated for a specified marking. In contrast, the firing of the transition itself depends on external factors.

The firing action and the associated marking transport are considered to be an event, namely it does not consume any time. However, for the operational planning it is desirable to also represent the time behaviour of the manufacturing system. References [4] and [5] introduce Petri nets with time-evaluated edges, transitions and places. However, the following general rule governs both the time-evaluated nets and the CP nets: Most extensions take some of the elegance from the theory of the Petri nets, which, in particular, impairs the analytical aspect of the theory.

References

- [1] ABEL, D.: *Petri-Netze für Ingenieure – Modellbildung und Analyse diskret gesteuerter Systeme*. Springer-Verlag, Berlin 1990
- [2] AMOSSOWA, N. N.; GILLERT, H.; KÜCHLER, U.; MAXIMOW, J. D.: *Bedienungstheorie – Eine Einführung*. BSB B.G. Teubner Verlagsgesellschaft, Leipzig 1986
- [3] CHURCHMAN, C.W.; ACKOFF, R.L.; ARNOFF, E.L.: *Operations Research – Eine Einführung in die Unternehmensforschung*. Verlag die Wirtschaft Berlin, Berlin 1968
- [4] KIENCKE, U.: *Ereignisdiskrete Systeme - Modellierung und Steuerung verteilter Systeme*. R. Oldenbourg Verlag, Munich, Vienna, 1997
- [5] KÖNIG, R.; QUÄCK, L.: *Petri-Netze in der Steuerungstechnik*. Verlag Technik, Berlin 1988
- [6] MÜLLER, P. H.: *Wahrscheinlichkeitsrechnung und Mathematische Statistik - Lexikon der Stochastik*. Akademie Verlag, Berlin 1991
- [7] WUNSCH, G.; SCHREIBER, H.: *Analoge Systeme (Grundlagen)*. Springer-Verlag Berlin Heidelberg 1993

Electronics Process Technology

Production Modelling, Simulation and Optimisation

Sauer, W.; Oppermann, M.; Weigert, G.; Werner, S.;

Wohlrabe, H.; Wolter, K.-J.; Zerna, Th.

2006, XX, 482 p. 281 illus., Hardcover

ISBN: 978-1-84628-353-6