Modeling and Control of Manufacturing Systems

Erjen Lefeber

Abstract In this chapter we provide a framework within which concepts from the field of systems and control can be used for controlling manufacturing systems. After introducing some basic notions from manufacturing analysis, we start with the concept of effective process times (EPTs) which can be used for modeling a manufacturing system as a large queuing network. Next, we restrict ourselves to mass production, which enables us to model manufacturing systems by means of a linear system subject to nonlinear constraints (clearing functions). These models serve as a starting point for designing controllers for these manufacturing systems using Model-based Predictive Control (MPC). Finally, the resulting controllers can be implemented on the queuing network model, and ultimately at the real manufacturing system.

1 Preliminaries

In this section we first recall a few basic notions and the main principles from manufacturing system analysis.

1.1 Basic Notions from Manufacturing Analysis

The items produced by a manufacturing system are called *lots*. Also the words product and job are commonly used. Other important notions are throughput, flow time, wip and utilization. These notions are illustrated in Fig. 1 at factory and machine level.

E. Lefeber (✉)
Department of Mechanical Engineering, Eindhoven University of Technology,
P.O. Box 513, 5600 MB, Eindhoven, The Netherlands
e-mail: A.A.J.Lefeber@tue.nl

D. Armbruster and K. G. Kempf (eds.), *Decision Policies for Production Networks*,
DOI: 10.1007/978-0-85729-644-3_2, © Springer-Verlag London 2012

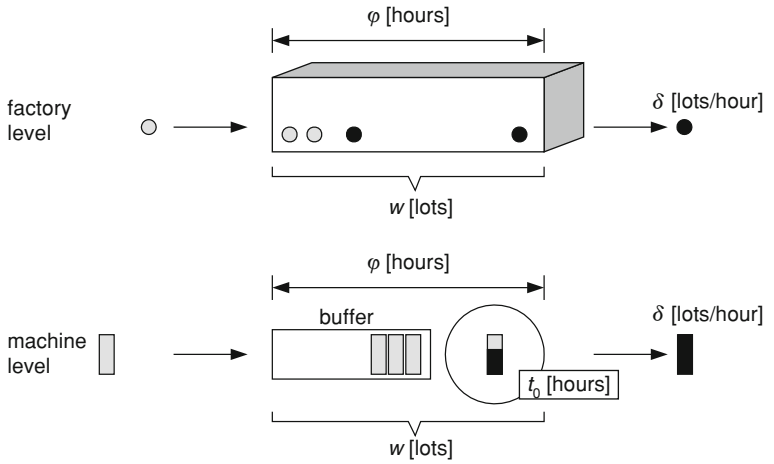


Fig. 1 Basic quantities for manufacturing systems

Raw process time t_0 of a lot denotes the net time a machine needs to process the lot. This process time excludes additions such as setup time, breakdown, or other sources that may increase the time a lot spends in the machine. The raw process time is typically measured in hours or minutes.

Throughput δ denotes the number of lots per unit time that leaves the manufacturing system. At a machine level, this denotes the number of lots that leave a machine per unit time. At a factory level it denotes the number of lots that leave the factory per unit time. The unit of throughput is typically lots/hour.

Flow time φ denotes the time a lot is in the manufacturing system. At a factory level this is the time from the release of the lot into the factory until the finished lot leaves the factory. At a machine level this is the time from entering the machine (or the buffer in front of the machine) until leaving the machine. Flow time is typically measured in days, hours, or minutes. Instead of flow time the words cycle time and throughput time are also commonly used.

Work in process (wip) w denotes the total number of lots in the manufacturing system, i.e., in the factory or in the machine. Wip is measured in lots.

Utilization u denotes the fraction of time that a machine is not idle. A machine is considered idle if it could start processing a new lot. Thus process time as well as downtime, setup-time and preventive maintenance time all contribute to the utilization. Utilization has no dimension and can never exceed 1.0.

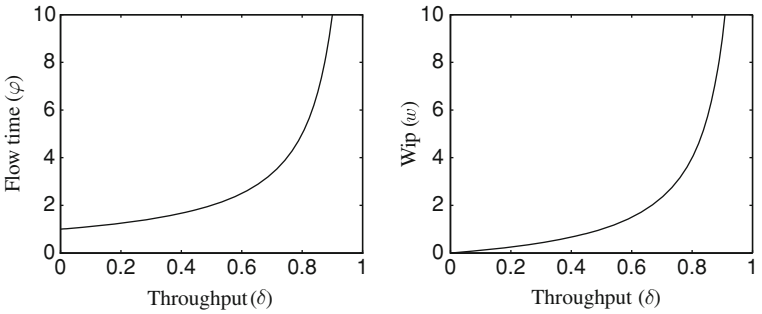


Fig. 2 Basic relations between basic quantities for manufacturing systems

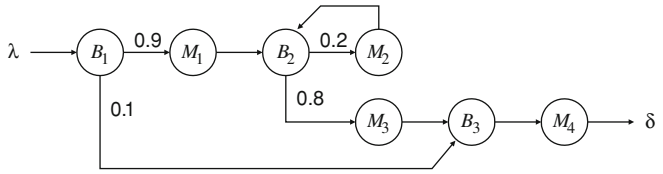
Ideally, a manufacturing system should have both a high throughput and a low flow time or low wip. Unfortunately, these goals are conflicting (cf. Fig. 2) and can not both be met simultaneously. If a high throughput is required, machines should always be busy. As from time to time disturbances like machine failures happen, buffers between two consecutive machines are required to make sure that the second machine can still continue if the first machine fails (or vice versa). Therefore, for a high throughput many lots are needed in the manufacturing system, i.e., wip needs to be high. As a result, if a new lot starts in the system it has a large flow time, since all lots that are currently in the system need to be completed first.

Conversely, the least possible flow time can be achieved if a lot arrives at a completely empty system and never has to wait before processing takes place. As a result, the wip level is small. However, for most of the time machines are not processing, yielding a small throughput.

When trying to control manufacturing systems, a trade-off needs to be made between throughput and flow time, so the nonlinear (steady state) relations depicted in Fig. 2 need to be incorporated in any reasonable model of manufacturing systems. We return to this in Sect. 4.1 when discussing clearing functions.

1.2 Analytical Models for Steady-State Analysis

In order to get some insights in the steady-state performance of a given manufacturing system simple relations can be used. We first deal with mass conservation for determining the mean utilization of workstations and the number of machines required for meeting a required throughput. Furthermore, relations from queueing theory are used to obtain estimates for the mean wip and mean flow time.



machine	t_0[hrs]
M_1	2.0
M_2	6.0
M_3	1.8
M_4	1.6

Fig. 3 Manufacturing system with rework and bypassing

1.2.1 Mass Conservation (Throughput)

Using mass conservation the mean utilization of workstations can easily be determined.

Example 1 Consider the manufacturing system with rework and bypassing in Fig. 3. The manufacturing system consists of three buffers and four machines. Lots are released at a rate of λ lots/hour. The numbers near the arrows indicate the fraction of the lots that follow that route. For instance, of the lots leaving buffer B_1 90% goes to machine M_1 and 10% goes to buffer B_3. The process time of each machine is listed in the table in Fig. 3.

Let δ_{M_i} and δ_{B_i} denote the throughput of machine M_i $(i = 1, 2, 3, 4)$ and buffer B_i $(i = 1, 2, 3)$, respectively. Using mass conservation we obtain

$$\begin{aligned} \delta_{M_1} &= 0.9\delta_{B_1} & \delta_{B_1} &= \lambda \\ \delta_{M_2} &= 0.2\delta_{B_2} & \delta_{B_2} &= \delta_{M_1} + \delta_{M_2} \\ \delta_{M_3} &= 0.8\delta_{B_2} & \delta_{B_3} &= \delta_{M_3} + 0.1\delta_{B_1} \\ \delta_{M_4} &= \delta_{B_3} & \delta &= \delta_{M_4}. \end{aligned}$$

Solving these linear relations results in:

$$\begin{aligned} \delta_{M_1} &= 0.9\lambda & \delta_{B_1} &= \lambda \\ \delta_{M_2} &= 0.225\lambda & \delta_{B_2} &= 1.125\lambda \\ \delta_{M_3} &= 0.9\lambda & \delta_{B_3} &= \lambda \\ \delta_{M_4} &= \lambda & \delta &= \lambda. \end{aligned}$$

Using the process times of the table in Fig. 3, we obtain for the utilizations:

$$\begin{aligned} u_{M_1} &= 0.9\lambda \cdot 2.0/1 = 1.8\lambda & u_{M_3} &= 0.9\lambda \cdot 1.8/1 = 1.62\lambda \\ u_{M_2} &= 0.225\lambda \cdot 6.0/1 = 1.35\lambda & u_{M_4} &= \lambda \cdot 1.6/1 = 1.6\lambda. \end{aligned}$$

Machine M_1 has the highest utilization, therefore it is the bottleneck and the maximal throughput for this line is $\lambda = 1/1.8 = 0.56$ lots per hour. □

Using mass conservation, utilizations of workstations can be determined straightforwardly. This also provides a way for determining the number of machines required for meeting a given throughput. By modifying the given percentages the effect of rework or a change in product mix can also be studied.

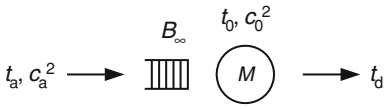


Fig. 4 Single machine workstation

1.2.2 Queueing Relations (Wip, Flow time)

For determining a rough estimate of the corresponding mean flow time and mean wip, basic relations from queueing theory can be used.

Consider a single machine workstation that consists of infinite buffer B_∞ and machine M (see Fig. 4). Lots arrive at the buffer with a stochastic interarrival time. The interarrival time distribution has mean t_a and a standard deviation σ_a which we characterize by the coefficient of variation $c_a = \sigma_a/\mu_a$. The machine has stochastic process times, with mean process time t_0 and coefficient of variation c_0. Finished lots leave the machine with a stochastic interdeparture time, with mean t_d and coefficient of variation c_d. Assuming independent interarrival times and independent process times, the mean waiting time φ_B in buffer B can be approximated for a stable system by means of Kingman's equation [10]:

$$\varphi_B = \frac{c_a^2 + c_0^2}{2} \cdot \frac{u}{1-u} \cdot t_0 \tag{1}$$

with the utilization u defined by: $u = t_0/t_a$. Equation 1 is exact for an $M/G/1$ system, i.e., a single machine workstation with exponentially distributed interarrival times and any distribution for the process time. For other single machine workstations it is an approximation.

For a stable system, we have $t_d = t_a$. We can approximate the coefficient of variation c_d by Kuehn's linking equation [11]:

$$c_d^2 = (1-u^2) \cdot c_a^2 + u^2 \cdot c_0^2. \tag{2}$$

This result is exact for an $M/M/1$ system. For other single machine workstations it is an approximation. Having characterized the departure process of a workstation, the arrival process at the next workstation has been characterized as well. As a result, a line of workstations can also be described.

Example 2 (*Three workstations in series*) Consider the three workstation flow line in Fig. 5. For the interarrival time at workstation 0 we have $t_a = 4.0$ h and $c_a^2 = 1$. The three workstations are identical with respect to the process times: $t_{0,i} = 3.0$ h for $i = 0, 1, 2$ and $c_{0,i}^2 = 0.5$ for $i = 0, 1, 2$. We want to determine the mean total flow time per lot.

Since $t_a > t_{0,i}$ for $i = 0, 1, 2$, we have a stable system and $t_{a,i} = t_{d,i} = 4.0$ h for $i = 0, 1, 2$. Subsequently, the utilization for each workstation is $u_i = 3.0/4.0 = 0.75$ for $i = 0, 1, 2$.

Using (1) we calculate the mean flow time for workstation 0

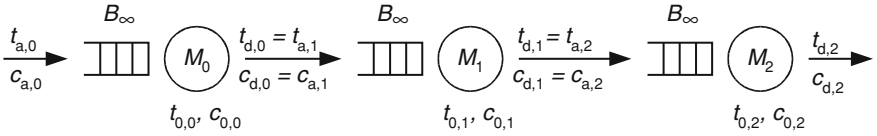


Fig. 5 Three workstation flow line

$$\varphi_0 = \varphi_B + t_0 = \frac{c_a^2 + c_0^2}{2} \cdot \frac{u}{1-u} \cdot t_0 + t_0 = \frac{1+0.5}{2} \cdot \frac{0.75}{1-0.75} \cdot 3.0 + 3.0 = 9.75\,\text{h}.$$

Using (2), we determine the coefficient of variation on the interarrival time $c_{a,1}$ for workstation W_1

$$c_{a,1}^2 = c_{d,0}^2 = (1-u^2) \cdot c_a^2 + u^2 \cdot c_0^2 = (1-0.75^2) \cdot 1 + 0.75^2 \cdot 0.5 = 0.719$$

and the mean flow time for workstation 1

$$\varphi_1 = \frac{0.719 + 0.5}{2} \cdot \frac{0.75}{1-0.75} \cdot 3.0 + 3.0 = 8.49\,\text{h}.$$

In a similar way, we determine that $c_{a,2}^2 = 0.596$, $\varphi_2 = 7.93\,\text{h}$. We then calculate the mean total flow time to be

$$\varphi_{\text{tot}} = \varphi_0 + \varphi_1 + \varphi_2 = 26.2\,\text{h}.$$

Note that the minimal flow time without variability ($c_a^2 = c_{0,i}^2 = 0$) equals 9.0 h. □

Equations 1 and 2 are particular instances of a workstation consisting of a single machine. For workstations consisting of m identical machines in parallel the following approximations can be used [8, 16]:

$$\varphi_B = \frac{c_a^2 + c_0^2}{2} \cdot \frac{u^{\sqrt{2(m+1)}-1}}{m(1-u)} \cdot t_0 \tag{3}$$

$$c_d^2 = (1-u^2) \cdot c_a^2 + u^2 \cdot \frac{c_0^2 + \sqrt{m} - 1}{\sqrt{m}}, \tag{4}$$

where the utilization $u = t_0/(m \cdot t_a)$. Notice that in case $m = 1$ these equations reduce to (1) and (2).

Once the mean flow time has been determined, a third basic relation from queueing theory, Little's law [14], can be used for determining the mean wip level. Little's law states that the mean wip level (number of lots in a manufacturing system) w is equal to the product of the mean throughput δ and the mean flow time φ, provided the system is in steady state:

$$w = \delta \cdot \varphi. \tag{5}$$

An example illustrates how Kingman's equation and Little's law can be used.

Example 3 Consider the system of Example 2 as depicted in Fig. 5. From Example 2 we know that the flow times for the three workstations are respectively

$$\varphi_0 = 9.75 \text{ h}, \varphi_1 = 8.49 \text{ h}, \varphi_2 = 7.93 \text{ h}.$$

Since the steady-state throughput was assumed to be $\delta = 1/t_a = 1/4.0 = 0.25$ lots/hour, we obtain via Little's law

$$\begin{aligned} w_0 &= 0.25 \cdot 9.75 = 2.44 \text{ lots}, \\ w_1 &= 0.25 \cdot 8.49 = 2.12 \text{ lots}, \\ w_2 &= 0.25 \cdot 7.93 = 1.98 \text{ lots}. \end{aligned}$$

□

The above mentioned relations are simple approximations that can be used for getting a rough idea about the possible performance of a manufacturing system. These approximations are fairly accurate for high utilizations but less accurate for lower degrees of utilization. A basic assumption when using these approximations is the independence of the interarrival times, which in general is not the case, e.g., for merging streams of lots. Furthermore, using these equations only steady state behavior can be analyzed. For studying things like ramp-up behavior or for incorporating more details like operator behavior, more sophisticated models are needed, as described next.

1.3 Discrete Event Models

A final observation of relevance for modeling manufacturing systems is the nature of the system signals. In Fig. 6a characteristic graph of the wip at a workstation as a function of time is shown. Wip always takes integer values with arbitrary (non-negative real) duration. One could consider a manufacturing system to be a system that takes values from a finite set of states and jumps from one state to the other as time evolves. This jump from one state to the other is called an *event*. As we have a countable (discrete) number of states, the name of this class of models is explained.

Manufacturing systems can be modeled as a network of concurrent processes. For example, a buffer is modeled as a process that as long as it can store something is willing to receive new products, and as long as it has something stored is willing to send products. A basic machine is modeled as a process that waits to receive a product; upon receipt it holds the product for a specified amount of time (delay). Upon completion, the machine tries to send the product to the next buffer in the manufacturing line. The machine keeps on doing these three consecutive things. The delay used is often a sample from some distribution.

In particular in the design phase discrete event models are used. These discrete event models usually contain a detailed description of everything that happens in the manufacturing system under consideration, resulting into large models. Since in

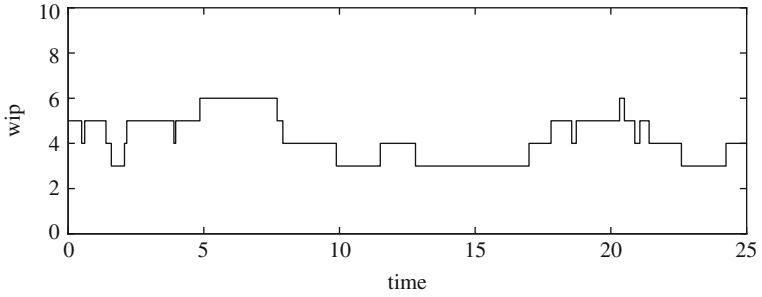


Fig. 6 A characteristic time-behavior of wip at a workstation

practice manufacturing systems are changing continuously, it is very hard to keep these discrete event models up-to-date [4].

Fortunately, for a manufacturing system in operation it is possible to arrive at more simple/less detailed discrete event models by using the concept of Effective Process Times (EPTs) as discussed in the next section.

2 Effective Process Times (EPTs)

For the processing of a lot at a machine, many steps may be required. For example, it could be that an operator needs to get the lot from a storage device, setup a specific tool that is required for processing the lot, put the lot on an available machine, start a specific program for processing the lot, wait until this processing has finished (meanwhile doing something else), inspect the lot to determine if all went well, possibly perform some additional processing (e.g., rework), remove the lot from the machine and put it on another storage device and transport it to the next machine. At all of these steps something might go wrong: the operator might not be available, after setting up the machine the operator finds out that the required recipe cannot be run on this machine, the machine might fail during processing, no storage device is available anymore so the machine cannot be unloaded and is blocked, etc.

Even though one might build a discrete event model including all these details, it is impossible to measure all sources of variability that might occur in a manufacturing system. One might measure some of them and incorporate these in a discrete event model. The number of operators and tools can be modeled explicitly and it is common practice to collect data on mean times to failure and mean times to repair of machines. Also schedules for (preventive) maintenance can be incorporated explicitly in a discrete event model. Nevertheless, still not all sources of variability are included. This is clearly illustrated in Fig. 7, obtained from [9]. The left graph contains actual realizations of flow times of lots leaving a real manufacturing system, whereas the right graph contains the results of a detailed discrete event simulation model including stochasticity. It turns out that in reality flow times are much higher

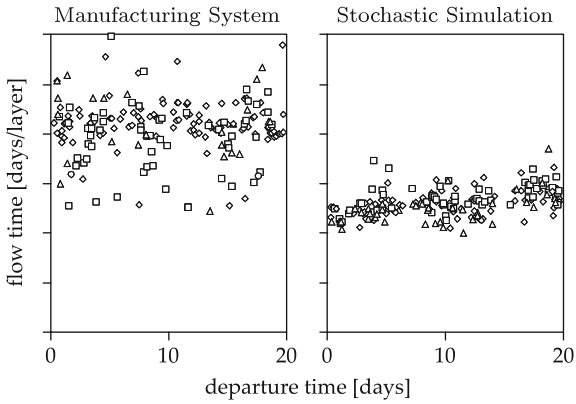


Fig. 7 A comparison

and much more irregular than simulation predicts. So, even if one endeavors to capture all variability present in a manufacturing system, still the outcome predicted by the model is far from reality.

Hopp and Spearman [8] use the term *Effective Process Time* (EPT) as the time seen by lots from a logistical point of view. In order to determine this Effective Process Time, Hopp and Spearman assume that the contribution of the individual sources of variability is known.

Instead of taking the bottom-up view of Hopp and Spearman, a top-down approach can also be taken, as shown by Jacobs et al. [9], where algorithms have been introduced that enable determination of Effective Process Time realizations from a list of events. For these algorithms, the basic idea of the Effective Process Time to include time losses was used as a starting point.

To illustrate this approach, we first deal with a workstation consisting of a single machine, serving one lot type, using a First In First Out (FIFO) policy. Then we deal with the more general case.

2.1 Single Machine, One Lot Type, FIFO Policy

Consider a workstation consisting of a single machine, serving one lot type, using a First In First Out (FIFO) policy. Let the Gantt chart of Fig. 8 depict what happened at this workstation during a certain time interval. At $t = 0$ the first lot arrives at the workstation. After a setup, the processing of the lot starts at $t = 2$ and is completed at $t = 6$. At $t = 4$ the second lot arrives at the workstation. At $t = 6$ this lot could have been started, but apparently no operator was available, so only at $t = 7$ the setup for this lot starts. Eventually, at $t = 8$ the processing of the lot starts and is completed at $t = 12$. The fifth lot arrives at the workstation at $t = 22$, processing starts at $t = 24$, but at $t = 26$ the machine breaks down. It takes until $t = 28$ before the machine has been repaired and the processing of the fifth lot continues. The processing of the fifth lot is completed at $t = 30$.

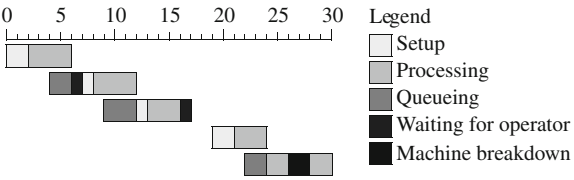


Fig. 8 Gantt chart of 5 lots at a single machine workstation

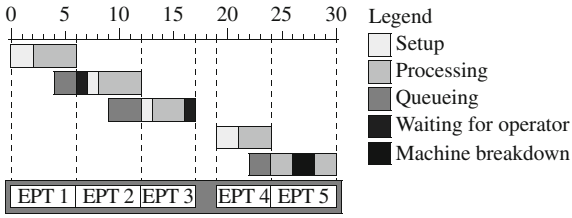


Fig. 9 EPT realizations of 5 lots at a workstation

If we take the point of view of a lot, what does a lot see from a logistical point of view? The first lot arrives at an empty system at $t = 0$ and departs from this system at $t = 6$. From the point of view of this lot, its processing took 6 time-units. The second lot arrives at a non-empty system at $t = 4$. Clearly, this lot needs to wait. However, at $t = 6$, if we forget about the second lot, the system becomes empty again. So from $t = 6$ on the second lot does not need to wait anymore. At $t = 12$ the second lot leaves the system, so from the point of view of this lot, its processing took from $t = 6$ till $t = 12$; the lot does not know whether waiting for an operator and a setup is part of its processing. Similarly, the third lot sees no need for waiting after $t = 12$ and leaves the system at $t = 17$, so it assumes to have been processed from $t = 12$ till $t = 17$. Following this reasoning, the resulting Effective Proces Times for lots are as depicted in Fig. 9. Notice that only arrival and departure events of lots to a workstation are needed for determining the Effective Process Times. Furthermore, none of the contributing disturbances needs to be measured.

In highly automated manufacturing systems, arrival and departure events of lots are being registered, so for these manufacturing systems, Effective Process Time realizations can be determined rather easily. Next, these EPT realizations can be used in a relatively simple discrete event model of the manufacturing system. This discrete event model only contains the architecture of the manufacturing system, buffers and machines. The process times of these machines are samples from their EPT-distribution as measured from real manufacturing data. Machine failures, operators, etc., do not need to be included as this is all included in the EPT-distributions. Furthermore, the algorithms as provided in [9] are *utilization independent*. That is, data collected at a certain throughput rate is also valid for different throughput rates. Furthermore, since EPT-realizations characterize operational time variability, they can be used for performance measuring. For more on this issue, the interested reader is referred to [9].

Recently, the above mentioned EPT-model has been generalized. This generalization is presented next.

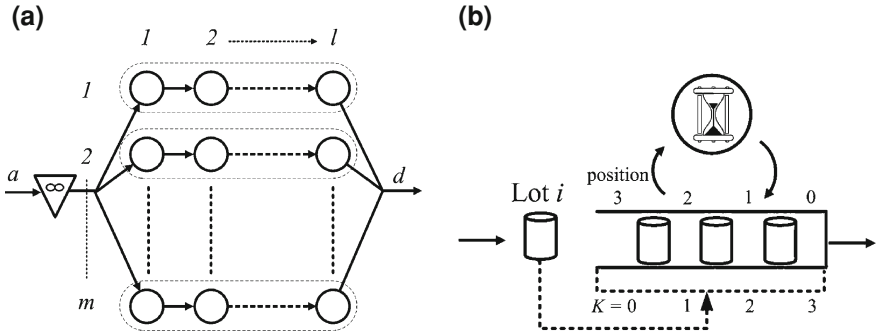


Fig. 10 **a** An example of a workstation. **b** The proposed aggregate model

2.2 Integrated Processing Workstations

Consider an integrated processing workstation consisting of m identical parallel machines, each of which have l sequential integrated processes, cf. Fig. 10. We replace the model of this workstation with a much simpler model, which is not a true physical server anymore, i.e., the structure of the aggregate model differs significantly from the real workstation. Nevertheless, the input/output behavior of the aggregate model closely resembles the input/output behavior of the workstation it models. Lots arrive according to some arrival process to the queue of the aggregate model. Lot i is defined as the ith *arriving* lot in this queue. The queue consists of *all* lots that are currently in the system, including lots that are (supposed to be) in process. Therefore, the queue is *not* a queue as in common queue-server models. Lots are not physically processed, i.e., during "processing" lots stay in the queue. Processing is modeled as a timer that determines when the next lot leaves the queue. When the timer expires, i.e., the "process time" has elapsed, the lot that is currently first in the queue leaves the system. Upon arrival of a new lot i, it is determined how many of the lots already present in the queue w are overtaken by lot i. The number of lots to overtake $K \in \{0, 1, \ldots, w\}$ is sampled from a probability distribution which depends on the number of lots w in the queue just before lot i arrives. The arriving lot is placed on position $w - K$ in the queue, where position 0 corresponds with the head of the queue. The timer starts when either a lot arrives to an empty system, or a lot departs while leaving one or more lots behind. The duration of the "process time" is sampled from a distribution which depends on the number of lots w in the queue just after the timer starts, i.e., including a possibly newly arrived lot. We model the server as a timer to allow newly arriving lots to overtake *all* lots in the system while the timer is running. We need this to model the possibility that a lot which arrives second to a multi-machine workstation leaves first.

Example 4 Consider the Gantt chart in Fig. 11 which depicts what happened at a three machine workstation. At $t = 1$, the first lot arrives at the workstation, service at machine 1 is started, and service is completed at $t = 25$. At $t = 2$, the second lot

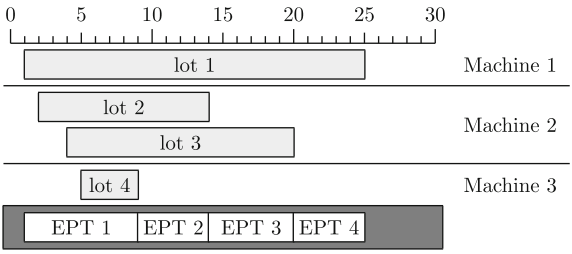


Fig. 11 Gantt chart of 4 lots at a three machine workstation, and the corresponding realization for the aggregate model

arrives at the workstation, service at machine 2 is started, and service is completed at $t = 14$. At $t = 4$, the third lot arrives at the workstation. For some reason it is not served at machine 3, but it waits to be served at machine 2. Its service at machine 2 (effectively) starts at $t = 14$ and is completed at $t = 20$. Finally, the fourth lot arrives at the workstation at $t = 5$, is served at machine 3, and leaves the system at $t = 9$.

In the aggregate model we model the resulting input-output behavior of this system differently. At $t = 1$, the first lot arrives and a timer is set, which expires at $t = 9$. Meanwhile, the second lot arrives at $t = 2$ and is inserted at the head of the queue. Next, the third lot arrives at $t = 4$, and is inserted in the middle of the queue, i.e., behind lot 2, but in front of lot 1. At $t = 5$, the fourth lot arrives which is inserted at the head of the queue, i.e., it overtakes the three lots already in the queue. When the timer expires at $t = 9$, the lot that is at the head of the queue leaves the system, i.e., lot 4 leaves the system. Then the timer is set again to expire at $t = 14$. Again, the head of the queue leaves the system, which is lot 2. The timer is set again to expire at $t = 20$, and lot 3 leaves the system. Next, the timer is set to ring at $t = 25$ and finally lot 1 leaves the system.

For more details about this aggregate model for integrated processing workstations, including implementation issues and algorithms for deriving distributions from real manufacturing data, the interested reader is referred to [19]. In that paper an extensive simulation study and an industry case study demonstrate that the aggregate model can accurately predict the cycle time distribution of integrated processing workstations in semiconductor manufacturing.

Most importantly, EPTs can be determined from real manufacturing data and yield relatively simple discrete event models of the manufacturing system under consideration. These relatively simple discrete event models serve as a starting point for controlling manufacturing systems.

3 Control Framework

In the previous section, the concept of Effective Process Times has been introduced as a means to arrive at relatively simple discrete event models for manufacturing systems, using measurements from the real manufacturing system under

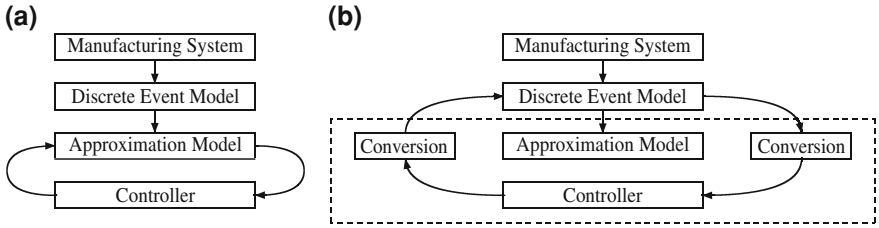


Fig. 12 **a** Control framework (I). **b** Control framework (II)

consideration. This is the first step in a control framework. The resulting discrete event models are large queueing networks which capture the dynamics reasonably well. These relatively simple discrete event models are not only a starting point for analyzing the dynamics of a manufacturing system, but can also be used as a starting point for controller design. If one is able to control the dynamics of the discrete event model of the manufacturing system, the resulting controller can also be used for controlling the real manufacturing system.

Even though control theory exists for controlling discrete event systems, unfortunately none of it is appropriate for controlling discrete event models of real-life manufacturing systems. This is mainly due to the large number of states of a manufacturing system. Therefore, a different approach is needed.

If we concentrate on mass production, the distinction between lots is not really necessary and lots can be viewed in a more continuous way. Instead of the discrete event model we might consider an approximation model. This is the second step in the control framework. Next, we can use standard control theory for deriving a controller for the approximation model. These first three steps in the control framework are illustrated in Fig. 12a. We elaborate on this second and third step in the next two sections. For now it is sufficient to know that time is discretized into periods (e.g., shifts) and that the resulting controller provides production targets per shift for each machine. So for now we assume that the derived controller behaves as desired on the approximation model. As a fourth step this controller could be connected to the discrete event model. This cannot be done directly, since the derived controller is not a discrete event controller. The control actions still need to be transformed into events. It might very well be that the optimal control action is to produce 2.75 lots during the next shift. One still needs to decide how many lots to really start (2 or 3), and also when to start them. This is the left conversion block in Fig. 12b. From this figure, it can also be seen that a conversion is needed from discrete event model to controller. In the remainder of this chapter we assume to sample the discrete event model once every shift. Other strategies might be followed. For example, if at the beginning of a shift a machine breaks down it might not be such a good idea to wait until the end of the shift before setting new production targets. Designing proper conversion blocks is the fourth step in the control framework.

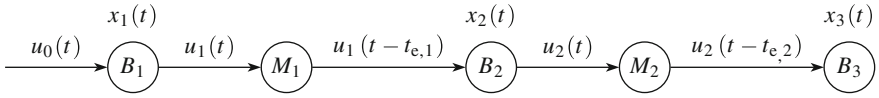


Fig. 13 A simple manufacturing system

After the fourth step, i.e., properly designing the two conversion blocks, a suitable discrete event controller for the discrete event model is obtained, as illustrated in Fig. 12b (dashed).

Eventually, as a fifth and final step, the designed controller can be disconnected from the discrete event model, and attached to the manufacturing system.

4 An Approximation Model

The analytical approximations models of Sect. 1.2 are only concerned with steady state, no dynamic behavior is included. This disadvantage is overcome by discrete event models as discussed in Sect. 2, where each lot is modeled separately and stochastically. In Sect. 2 we derived how less detailed discrete event models can be built by abstracting from all kinds of disturbances like machine failure, setups, operator behavior, etc. By aggregating all disturbances into one Effective Process Time, a complex manufacturing system can be modeled as a relatively simple queueing network. Furthermore, the data required for this model can easily be measured from manufacturing data.

Even though this approach considerably reduces the complexity of discrete event models for manufacturing systems, this aggregate model is still unsuitable for manufacturing planning and control. Therefore, in this section we introduce a next level of aggregation, by abstracting from events. Using the abstraction presented in Sect. 2 we can view a workstation as a node in a queueing network. In this section we assume that such a node processes a deterministic continuous stream of fluid. That is, we consider this queue as a so-called fluid queue.

For example, consider a simple manufacturing system consisting of two machines in series, as displayed in Fig. 13. Let $t_{e,i}$ denote the Effective Process Time of the ith machine for $i \in \{1, 2\}$. Furthermore, let $u_0(t)$ denote the rate at which lots arrive at the system at time t, $u_i(t)$ the rate at machine M_i starts lots at time t, $x_i(t)$ the number of lots in buffer B_i at time t ($i \in \{1, 2\}$) and $x_3(t)$ the cumulative number of lots produced by the manufacturing system at time t.

The rate of change of the buffer contents is given by the difference between the rates at which lots enter and leave the buffer, taking into account the time-delay due to processing:

$$\begin{aligned}\dot{x}_1(t) &= u_0(t) - u_1(t),\\ \dot{x}_2(t) &= u_1\left(t - t_{e,1}\right) - u_2(t),\\ \dot{x}_3(t) &= u_2\left(t - t_{e,2}\right).\end{aligned} \tag{6}$$

In practice, manufacturing systems are often controlled by means of setting production targets per shift. That is, time is divided into shifts for example, 8 or 12 h. For this period of 8 or 12 h it is determined how many lots should be started on each machine. The control problem then reduces to determining these production targets per shift.

To that end, we sample the continuous time system (6) using a zero-order-hold sampling, cf. [2]. Assuming that the longest Effective Process Time is less than the duration of a shift, the resulting zero-order-hold sampling of the system in (6) becomes

$$\begin{bmatrix}\bar{x}_1(k+1)\\ \bar{x}_2(k+1)\\ \bar{x}_3(k+1)\\ \hline \bar{x}_4(k+1)\\ \bar{x}_5(k+1)\end{bmatrix} = \left[\begin{array}{ccc|cc} 1 & 0 & 0 & 0 & 0\\ 0 & 1 & 0 & \frac{t_{e,1}}{h} & 0\\ 0 & 0 & 1 & 0 & \frac{t_{e,2}}{h}\\ \hline 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0\end{array}\right] \begin{bmatrix}\bar{x}_1(k)\\ \bar{x}_2(k)\\ \bar{x}_3(k)\\ \hline \bar{x}_4(k)\\ \bar{x}_5(k)\end{bmatrix} + \left[\begin{array}{ccc} 1 & -1 & 0\\ 0 & \frac{h-t_{e,1}}{h} & -1\\ 0 & 0 & \frac{h-t_{e,2}}{h}\\ \hline 0 & 1 & 0\\ 0 & 0 & 1\end{array}\right] \begin{bmatrix}\bar{u}_0(k)\\ \bar{u}_1(k)\\ \bar{u}_2(k)\end{bmatrix} \tag{7}$$

where $\bar{u}_0(k)$ denotes the number of lots arriving at the system during shift k, $\bar{u}_i(k)$ the number of lots started at machine M_i during shift k, $\bar{x}_i(k)$ the number of lots in buffer B_i at the beginning of shift k ($i \in \{1, 2\}$), and $\bar{x}_3(k)$ the cumulative number of lots produced by the manufacturing system at the beginning of shift k. Furthermore, h denotes the sample period, e.g., 8 or 12 h. The auxiliary variables $\bar{x}_4(k)$ and $\bar{x}_5(k)$ are required to remember the starts during the previous shift, in order to incorporate the lots for which processing is started in shift k on machine M_1 and M_2 respectively but completed in shift $k+1$. If the longest Effective Process Time exceeds the duration of a shift, but not exceed the duration of two shifts, similarly auxiliary variables $\bar{x}_6(k)$, and $\bar{x}_7(k)$ are required.

The model (6) and its discrete time equivalent (7) are also subject to constraints. We present the constraints for the model (7). For the model (6), similar constraints hold.

The first constraint is a non-negativity constraint: buffer contents can never be negative. Also production targets cannot become negative. Expressed mathematically we have the following constraints:

$$\bar{x}_i(k) \geq 0 \qquad i \in \{1, 2, 3, 4, 5\} \qquad \forall k \tag{8a}$$

$$\bar{u}_j(k) \geq 0 \qquad j \in \{1, 2, 3\} \quad \forall k \tag{8b}$$

Furthermore, machines can produce at most at maximal capacity. That is, the total time spent on serving the required number of lots during a shift cannot exceed the duration of the shift:

$$t_{e,j} \cdot \bar{u}_j(k) \leq h \quad j \in \{1, 2, 3\} \quad \forall k \tag{8c}$$

where h again denotes the sample period or shift duration.

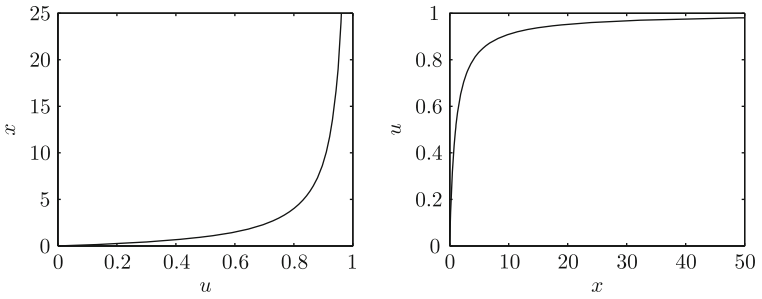


Fig. 14 Effective clearing function of (9) with $c_a = c_e = m = 1$

4.1 Clearing Functions

The model (7) with constraints (8) describes the dynamics of a manufacturing system well. By incorporating delays due to processing, the minimal flow time is also taken into account. Furthermore, steady-state corresponds with the mass conservation results presented in Sect. 1.2.1.

Nevertheless, one property of manufacturing systems is not yet taken into account in the model (7), (8). And that is the queueing relations (3).

In order not to lose the steady state queueing relation between throughput and queue length, we include this relation as a system constraint.

Consider a workstation that consists of m identical servers in parallel that all have a mean Effective Process Times t_e and coefficient of variation c_e. Furthermore, assume that the coefficient of variation of the interarrival times is c_a and that the utilization of this workstation is $\rho < 1$. Then we know from (3), (5) that in steady state the mean number of lots in this workstation is approximately given by

$$x = \frac{c_a^2 + c_e^2}{2} \cdot \frac{\rho^{\sqrt{2(m+1)}-1}}{m(1-\rho)} + \rho. \tag{9}$$

In Fig. 14 this relation has been depicted graphically. In the left-hand side of this figure one can clearly see that for an increasing utilization, the number of lots in this workstation increases nonlinearly. By swapping axes, this relation can be understood differently. Depending on the number of lots in the workstation, a certain utilization can be achieved, or a certain throughput. This has been depicted in the right-hand side of Fig. 14. This relation is also known as the *clearing function* as introduced by [7].

For the purpose of production planning, this effective clearing function provides an upper bound for the utilization of the workstation depending on the number of lots in this workstation. Therefore, for the model (7), in addition to the constraints (8) we also have (using $\rho = \bar{u} \cdot t_e/(h \cdot m)$ and $m = 1$):

$$\begin{aligned}
&\frac{c_{\mathrm{a},1}^2 + c_{\mathrm{e},1}^2}{2} \cdot \frac{u_1(k)^2}{\frac{h}{t_{\mathrm{e},1}}\left(\frac{h}{t_{\mathrm{e},1}} - u_1(k)\right)} + \frac{t_{\mathrm{e},1}}{h} u_1(k) \leq \bar{x}_1(k) \quad \forall k \\
&\frac{c_{\mathrm{a},2}^2 + c_{\mathrm{e},2}^2}{2} \cdot \frac{u_2(k)^2}{\frac{h}{t_{\mathrm{e},2}}\left(\frac{h}{t_{\mathrm{e},2}} - u_2(k)\right)} + \frac{t_{\mathrm{e},2}}{h} u_2(k) \leq \bar{x}_2(k) \quad \forall k.
\end{aligned} \tag{10}$$

The clearing function model for production planning then consists of the model (7) together with the constraints (8) and (10). When we want to use this clearing function model for production planning, we need the parameters c_{e} and c_{a}. In Sect. 2 we explained how Effective Process Times can be determined for each workstation, which provides us with the parameter c_{e} for each workstation. Additionally, for each workstation the interarrival times of lots can also be determined from arrival events, which provides us with the parameter c_{a} for each workstation. Therefore, both parameters can easily be determined from manufacturing data.

We conclude this section with some remarks about the additional constraints (10). The first remark is that these constraints are convex in the input u, so optimization problems become "simple" convex optimization problems. A second remark is that from a practical point of view, one can easily approximate each convex constraint by means of several linear constraints. A third remark is that the constraints (10) only hold for steady state, whereas our system is never in steady state. A more accurate planning result is obtained by conditioning the expected throughput on the current work in the buffer, resulting in so-called *transient clearing functions*. For the latter subject, the interested reader is referred to [15].

5 Controller Design

In the previous section we derived a fluid model as an approximation for the discrete event model derived earlier. The next step in the control framework presented in Sect. 3 is to control the approximation model using standard techniques from control theory.

Typically two control problems can be distinguished: the *trajectory generation problem* and the *reference tracking control problem*. The solution of the first problem serves as an input for the second problem.

To illustrate the difference between these two problems, consider the problem of automatically flying an airplane from A to B by means of an autopilot. Then also two problems are solved separately. The first problem is to determine a trajectory for the airplane to fly which brings it from A to B. The resulting flight plan is a solution to the trajectory generation problem. The second problem is the design of the autopilot itself. Given an arbitrary feasible reference trajectory for this airplane, how to make sure that it is tracked as well as possible, despite all kinds of disturbances. The latter is the reference tracking control problem. We follow a similar approach for the control of manufacturing systems.

5.1 Trajectory Generation Problem

The trajectory generation problem is the problem of finding a feasible reference trajectory for the system, also known as production planning. So for the example considered previously, the problem is to find a trajectory $\big(x_r(k), u_r(k)\big)$ which satisfies (7) as well as the constraints (8) and (10). Clearly, many trajectories exist that meet these requirements. Typically, "the best" trajectory is looked for. Therefore, the trajectory generation or production planning problem is often formulated as an optimization problem.

Example 5 Consider the system described by (7) together with the constraints (8) and (10). Assume that $c_{\mathrm{a,i}} = c_{\mathrm{e,i}} = 1$, $t_{\mathrm{e,i}} = 1$ $(i = 1, 2)$, $h = 2$, and that the cumulative demand is given by $x_{r,3}(k) = k$. If one would like to satisfy this cumulative demand while having a minimal number of jobs in the system, the trajectory generation problem can be formulated as the following optimization problem:

$$\begin{aligned} &\min_{u_r(k), x_r(k)} \sum_{k=1}^{N} x_1(k) + x_2(k) \\ &\text{subject to } x_{r,3} = k && k = 1, \ldots, N \\ &\qquad (7), (8), (10) && k = 1, \ldots, N \end{aligned}$$

The solution to this problem is given by

$$\begin{array}{llll} x_{r,1}(k) = 1 & u_{r,0}(k) = 1 & k = 1, \ldots, N \\ x_{r,2}(k) = 1 & u_{r,1}(k) = 1 & k = 1, \ldots, N \\ x_{r,3}(k) = k & u_{r,2}(k) = 1 & k = 1, \ldots, N \\ x_{r,4}(k) = 1 & & k = 1, \ldots, N \\ x_{r,5}(k) = 1 & & k = 1, \ldots, N. \end{array} \tag{11}$$

5.2 Reference Tracking: Model-Based Predictive Control (MPC)

For the reference tracking control problem, we assume that an *arbitrary* feasible reference trajectory is given. So for the example considered before we assume that a reference trajectory $\big(x_r(k), u_r(k)\big)$ is given which satisfies (7) together with the constraints (8) and (10). This could for example be the trajectory (11), but any other feasible reference trajectory can be used as a starting point as well. The goal in the reference tracking control problem is to find an input $u(k)$ which guarantees that the system tracks this reference input, while meeting the constraints (8) and (10).

In order to solve the reference tracking control problem, the tracking error dynamics is considered. For the remainder of this section we assume that the system dynamics is described by

$$x(k+1) = Ax(k) + Bu(k)$$

subject to the linear constraints

$$Ex(k) + Fu(k) \leq g.$$

Without loss of generality this can be extended to nonlinear dynamics with nonlinear constraints.

In addition, a feasible reference trajectory $\big(x_r(k), u_r(k)\big)$ is given, i.e., a trajectory which satisfies

$$x_r(k+1) = Ax_r(k) + Bu_r(k)$$

and

$$Ex_r(k) + Fu_r(k) \leq g.$$

Next, one can define the tracking error $\tilde{x}(k) = x(k) - x_r(k)$, and the input correction $\tilde{u}(k) = u(k) - u_r(k)$. Then the tracking error dynamics becomes

$$\tilde{x}(k+1) = A\tilde{x}(k) + B\tilde{u}(k) \tag{12a}$$

subject to the constraints

$$E\big(\tilde{x}(k) + x_r(k)\big) + F\big(\tilde{u}(k) + u_r(k)\big) \leq g$$

or

$$E\tilde{x}(k) + F\tilde{u}(k) \leq g - Ex_r(k) - Fu_r(k) \tag{12b}$$

Using these error coordinates, the reference tracking control problem can be formulated as to find an input correction $\tilde{u}(k)$ which steers the error dynamics (12a) toward 0, while satisfying the constraints (12b).

Since we have a system with constraints, the most suitable technique from standard control theory is Model-based Predictive Control (MPC).

The basic idea of MPC is to use the model of the system (12a) to predict the state evolution as a function of future inputs. Furthermore, a cost function is used which penalizes the predicted future deviations from the reference trajectory. This cost function is then minimized over the future inputs, subject to the constraints (12b). This optimization takes place over a so-called prediction horizon p, i.e., the first p inputs are determined in this optimization problem. The resulting control action then consists of the first of these inputs. One time period later, the entire procedure is repeated. Therefore, MPC is also called a receding horizon stategy. This is illustrated in Fig. 15.

Assume that at time k, the tracking error $\tilde{x}(k) = \tilde{x}(k|k)$ is measured. So we have the tracking error $\tilde{x}$ at time k given that we are currently at time k. Using a horizon of length p, we can define the input corrections for the times $k, k+1, \ldots, k+p-1$ given that we are currently at time k: $\tilde{u}(k|k), \tilde{u}(k+1|k), \ldots, \tilde{u}(k+p-1|k)$.

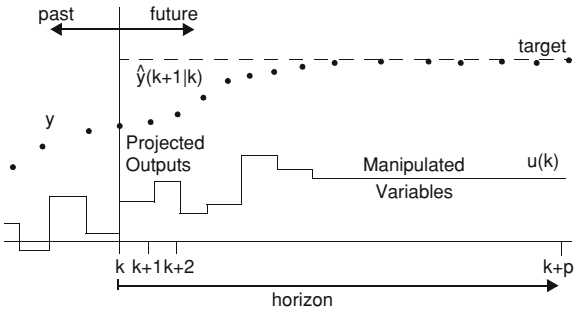


Fig. 15 The ingredients of MPC

By means of the model (12a) we are able to predict the resulting tracking errors as a function of these future input corrections:

$$\begin{bmatrix} \tilde{x}(k+1|k) \\ \tilde{x}(k+2|k) \\ \vdots \\ \tilde{x}(k+p|k) \end{bmatrix} = \begin{bmatrix} A \\ A^2 \\ \vdots \\ A^p \end{bmatrix} x(k|k) + \begin{bmatrix} B & 0 & \dots & 0 \\ AB & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ A^{p-1}B & \dots & AB & B \end{bmatrix} \begin{bmatrix} \tilde{u}(k|k) \\ \tilde{u}(k+1|k) \\ \vdots \\ \tilde{u}(k+p-1|k) \end{bmatrix} \tag{13}$$

Next we define a cost function for having a non-zero tracking error. One of the properties of our controlled system is that if we happen to be on the reference, we should stay on the reference. In particular this implies that the cost function should be such that costs are 0 if and only if the system stay in $(\tilde{x}, \tilde{u}) = (0, 0)$.

In control theory often a quadratic cost function is used:

$$\min_{u(k|k),\dots u(k+p-1|k)} \sum_{i=1}^{p} x(k+i|k)^T Q x(k+i|k) + u(k+i-1|k)^T R u(k+i-1|k) \tag{14}$$

with $Q = Q^T \geq 0$ and $R = R^T > 0$. But also other cost functions can be used, e.g., linear cost functions. What is most important is that costs are 0 if and only if the system stays in $(\tilde{x}, \tilde{u}) = (0, 0)$. Clearly the minimization should take place subject to the constraints (12b). Using a quadratic cost function as in (14) results in a QP (quadratic program) to be solved each time instant, whereas a linear cost function results in an LP (linear program), see e.g., [18].

The result from solving the above-mentioned optimization problem is a vector of future input corrections $\tilde{u}(k|k), \tilde{u}(k+1|k), \dots, \tilde{u}(k+p-1|k)$. At time k the input $\tilde{u}(k|k)$ is applied. Subsequently, at time $k+1$ the whole procedure starts all over again.

We conclude this section with some remarks. First, the stability of the MPC approach is not guaranteed. At least not in the way as presented here. In order to achieve guaranteed stability, one should take the horizon $p = \infty$. This is not desirable from a practical point of view. A second way of achieving stability is by adding the

terminal constraint that after the horizon, the system should be on the reference, i.e., one could add the constraint that $\tilde{x}(k+p) = 0$. Notice that in order to have a feasible optimization problem, again one should take p large enough.

For more information about MPC, the interested reader is referred to [5].

6 Concluding Remarks

In this chapter we provided a framework within which concepts from the field of systems and control can be used for controlling manufacturing systems. We presented the concept of Effective Process Times (EPTs) which can be used for modeling a manufacturing system as a large queuing network. Restricting ourselves to mass production enabled us to model manufacturing systems by means of a linear system subject to nonlinear constraints (clearing functions). These models then served as a starting point for designing controllers for these manufacturing systems using Model-based Predictive Control (MPC). Thoughout this chapter we provided examples to illustrate the most important ideas and concepts. We also provided additional references for the interested reader.

We presented MPC as a possible approach from control theory for controlling manufacturing systems. But many more suitable approaches can be used, ranging from classical control theory using z-transforms and transfer functions, dynamic programming and optimal control, to robust control and approximate dynamic programming. A good overview of these kinds of approaches for the dynamic modeling and control of supply chains has been provided in the review paper [17].

But also the approximation model presented in Sect. 4 is only one of the possible choices for modeling manufacturing systems. An overview on aggregate models for manufacturing systems has been given in [13]. In the model presented here a fluid approximation has been presented where the number of jobs was modeled continuously, but the position in the factory was modeled discretely. Using a less detailed model, we can even abstract from workstations and model manufacturing flow as a real fluid using continuum models [1, 3, 6]. Optimal control of PDE models for manufacturing systems has been presented in [12].

From the above it is clear that the modeling and control of manufacturing systems has been, and still is, an open and active research area. In this chapter we provided some of the basic models and standard control approaches, illustrated by examples so that they can be applied straighforwardly.

Acknowledgements Erjen Lefeber is supported by the Netherlands Organization for Scientific Research (NWO-VIDI grant 639.072.072).

References

1. Armbruster D, Marthaler DE, Ringhofer C, Kempf K, Jo TC (2006) A continuum model for a re-entrant factory. Operations Research 54(5):933–950
2. Åström KJ, Wittenmark B (1990) Computer-controlled systems: theory and design, 2nd edn. Prentice-Hall, Englewood Cliffs
3. Daganzo CF (2003) A Theory of Supply Chains. Springer, New York
4. Fischbein S, Yellig E (2011) Why is it so hard to build and validate discrete event simulation models of manufacturing facilities. In: Kempf KG, Uzsoy R, Keskinocak P (eds) Planning production and inventories in the extended enterprise: a state of the art handbook, volume 2, Springer international series in operations research and management science, vol 152, chap 12. Springer, New York pp 271–288
5. Garcia CE, Prett DM, Morari M (1989) Model predictive control: theory and practice—a survey. Automatica 25(3):335–348
6. Göttlich S, Herty M, Klar A (2005) Network models for supply chains. Commun Math Sci 3(4):545–559
7. Graves SC (1986) A tactical planning model for a job shop. Oper Res 34(4):522–533
8. Hopp WJ, Spearman ML (2000) Fact Physics, 2nd edn. McGraw-Hill, New York
9. Jacobs JH, Etman LFP, Campen EJJv, Rooda JE (2003) Characterization of the operational time variability using effective processing times. IEEE Trans Semicond Manuf 16(3):511–520
10. Kingman JFC (1961) The single server queue in heavy traffic. Proc Camb Philos Soc 57:902–904
11. Kuehn PJ (1979) Approximate analysis of general queueing networks by decomposition. IEEE Trans Commun 27:113–126
12. La Marca M, Armbruster D, Herty M, Ringhofer C (2010) Control of continuum models of production systems. IEEE Trans Autom Control 55(11):2511–2526
13. Lefeber E, Armbruster D (2011) Aggregate modeling of manufacturing systems. In: Kempf KG, Uzsoy R, Keskinocak P (eds) Planning production and inventories in the extended enterprise: a state of the art handbook, volume 1, Springer international series in operations research and management science, vol 151, chap 17, pp 509–536 Springer, New York.
14. Little JDC (1961) A proof of the queueing formula $l = \lambda w$. Oper Res 9:383–387
15. Missbauer H (2009) Models of the transient behaviour of production units to optimize the aggregate material flow. Int J Prod Econ 118:387–397
16. Sakasegawa H (1977) An approximation formula $L_q \approx \alpha\rho^{\beta}/(1-\rho)$. Ann Inst Stat Mech 29:67–75
17. Sarimveis H, Patrinos P, Tarantilis CD, Kiranoudis CT (2008) Dynamic modeling and control of supply chain systems: A review. Comput Oper Res 35(11):3530–3561
18. Vargas-Villamil FD, Rivera DE, Kempf (2003) A hierarchical approach to production control of reentrant semiconductor manufacturing lines. IEEE Trans Control Syst Technol 11(4):578–587
19. Veeger CPL, Etman LFP, Lefeber E, Adan IJBF, Herk Jv, Rooda JE (2011) Predicting cycle time distributions for integrated processing workstations: an aggregate modeling approach. IEEE Trans Semicond Manuf 24(2):223–236