Modeling of manufacturing systems with finite buffer sizes using PDEs

P.C.E. Goossens

SE 420523

Masters thesis

Supervisor: Prof. dr. ir. J.E. Rooda
Coach: Dr. ir. A.A.J. Lefeber

Technische Universiteit Eindhoven
Department of Mechanical Engineering
Systems Engineering Group

Eindhoven, November 2007

FINAL ASSIGNMENT

EINDHOVEN UNIVERSITY OF TECHNOLOGY
Department of Mechanical Engineering
Systems Engineering Group

September 2006

Student	P.C.E. Goossens
Supervisor	Prof.dr.ir J.E. Rooda
Advisor	Dr.ir. A.A.J. Lefeber
Start	September 2006
Finish	September 2007
Title	Modeling of manufacturing systems with finite buffers sizes using PDE's

Subject

The modeling and control of manufacturing systems becomes more and more complex. The current models used for this purpose all have their own shortcomings. These models can be divided into three groups: queuing theory, discrete event models and fluid models. Fluid models describe systems without any dynamics, focus only on throughput and are not concerned about flow time. Queuing theory takes the relation between throughput and flow time into account but focuses mainly on the steady state and is not suitable for control theory. The discrete events models can describe a simple manufacturing system but for larger systems it suffers from state explosion. To overcome these shortcomings a fourth group of models might be considered, PDE models. PDE models not only describe the dynamics of a system and gives a relation between the throughput and flow time but also control theory can be applied on these models.
Earlier research done on PDE models focuses on traffic flow models and how these models can be applied to manufacturing systems with infinite buffer sizes. Manufacturing systems with finite buffer sizes have not yet been considered.

Assignment

Study literature about traffic flow models in which traffic jams are modeled. Traffic jams show similar behavior as blocking in a manufacturing system due to a machine breakdown. Use the traffic flow models to propose a PDE which models a manufacturing system with finite buffer sizes. Validate this PDE model by comparing it to results obtained by discrete event simulation. Also queuing theory for systems with finite buffer sizes must be studied to determine the steady state. Finally, present the results in a report and provide recommendations for further research.

Prof.dr.ir J.E. Rooda

Dr.ir. A.A.J. Lefeber

Systems Engineering   Department of Mechanical Engineering

Summary

Modeling and control of manufacturing systems becomes more and more complex. Nowadays there are roughly three groups of models which can be used for modeling and control. These three groups are fluid models, discrete event models and queuing models. All of these models can be used for manufacturing systems but have their own shortcomings. Fluid models are not concerned about cycle time but are more focused on throughput. Discrete event models have a long calculation time for large manufacturing systems. Queuing models only determine the steady state of a manufacturing system and control theory can not be applied to these models.
A fourth group of models which may overcome these shortcomings are the partial differential equation (PDE) models. PDE models describe the flow of products as a continuous flow, the number of lots in the system is continuous as well as the position within the manufacturing system.

To determine if a PDE model might be used to model manufacturing systems with finite buffersizes the behavior of such a manufacturing system must be known. The determination of the behavior is the goal of this research: in order to know the behavior a PDE must describe, derive properties for the behavior of a tandem queue with finite buffersizes.

In this research the steady state and the dynamical behavior of a tandem queue is determined. The dynamical behavior is determined with the help of discrete event simulation and the steady state is determined with an approximation method based on queuing theory.

To determine the dynamical behavior a simulation experiment is done. In this experiment three kinds of dynamical behavior are simulated: the ramp up of a tandem queue, dynamical behavior after break down of the last machine in the tandem queue and the ramp down of the tandem queue. This experiment is done for a tandem queue of 10 workstations and a tandem queue of 100 workstations. For both tandem queues experiments are done for different parameter settings. In the results of all experiments similar behavior is seen.
During ramp up the mean number of products increases from 0 until steady state is reached. When the last machine breaks down, blocking is moving through the tandem queue from the last workstation to the first workstation with an almost constant velocity. During ramp down the mean number of products decreases from a maximal value

until it reaches steady state again. In the ramp down experiment all products which are stored in the buffer during blocking have to leave the system. When all these products are leaving the tandem queue the conditions in this tandem queue can be different from the conditions in the steady state for while. For these new conditions a new 'steady state' might be determined for some workstations. This new 'steady state' ends when the conditions of the real steady state are reached again.

To determine the steady state an approximation method based on queuing theory is used. To verify the results of this approximation method the steady state is compared to the steady state in the results of the discrete event simulations which are performed for the dynamical behavior.

After the behavior of a tandem queue is determined a PDE has to be determined which describes this behavior. A first step in determining a PDE can be to describe the dynamical behavior and the steady state with a finite volume method. Finite volume methods can describe the motion of waves and can approximate the solution of a PDE.

A finite volume method divides the domain into grid cells and in every grid cell the average density is known. This average density has to be updated every time step in every grid cell. The ingoing flux and the outgoing flux are needed for updating the density. These fluxes have to be described by a function which is only based on the average density in a grid cell. In this research the outgoing flux is determined with the results of the approximation method for the steady state.

Samenvatting

Het modelleren en regelen van fabricage systemen wordt steeds ingewikkelder. Tegenwoordig zijn er grofweg drie groepen modellen die voor deze doeleinden gebruikt worden. Dit zijn vloeistof modellen, discrete event modellen en wachtrij modellen. Deze modellen hebben allemaal tekortkomingen waardoor ze voor bepaalde doeleinden minder geschikt zijn. Vloeistof modellen modelleren voornamelijk doorzet en zijn minder gericht op het modelleren van doorlooptijd. Discrete event modellen hebben een lange rekentijd en zijn daardoor minder geschikt voor grote fabricage systemen. Wachtrij modellen kunnen vaak alleen toegepast worden op systemen in steady state.
Om deze tekortkomingen op te lossen zou een vierde groep van modellen beschouwd kunnen worden: partiële differentiaal vergelijkingen. Deze partiële differentiaal vergelijkingen beschrijven de stroom van producten in een fabricage system als een continue stroom van producten. Het aantal producten in een fabricage systeem wordt beschouwd als een continu aantal en ook de plaats in het systeem wordt beschouwd als een continue waarde.

Om te kunnen bepalen of partiële differentiaal vergelijkingen gebruikt kunnen worden voor fabricage systemen met eindige bufferinhouden zal eerst het gedrag van een fabricage systeem bekend moeten zijn. Dit onderzoek is dan ook gericht op het bepalen van dit gedrag: beschrijf het gedrag van een lijn werkstations met eindige bufferinhouden om te bepalen welk gedrag partiële differentiaal vergelijkingen moeten beschrijven.

In dit onderzoek is onderscheid gemaakt tussen steady state en dynamisch gedrag van een lijn werkstations. De steady state van een systeem is bepaald met een benaderingsmethode gebaseerd op wachtrij modellen en het dynamisch gedrag is bepaald met discrete event simulatie.

Om het dynamische gedrag van een lijn werkstations te bepalen is een simulatie experiment gedaan met behulp van een discrete event model. Het experiment is bedoeld om verschillende eigenschappen van dynamisch gedrag te modelleren. Het opstarten van een lijn werkstations is gesimuleerd, het gedrag dat ontstaat na het blokkeren van een machine wordt gesimuleerd en het leeglopen van een lijn met producten is gesimuleerd. Dit experiment is gedaan voor een lijn van 10 werkstations en een lijn van 100 werkstations met verschillende parameterwaarden voor bijvoorbeeld de maximale bufferinhoud en de variatiecoëfficiënt. In de resultaten van alle experimenten komt eenzelfde soort

gedrag naar voren.
Tijdens het opstarten van een lege lijn stijgt het aantal producten in elke werkstation van 0 totdat steady state is bereikt. In steady state blokkeert de laatste machine in de lijn en deze blokkade breidt zich uit in de lijn van het laatste werkstation tot het eerste werkstation. De snelheid waarmee de blokkade zich uitbreidt is ongeveer constant. Nadat de blokkade het eerste werkstation heeft bereikt zijn alle buffers en machines gevuld met producten. Leeglopen van de lijn is nu gesimuleerd. Het aantal producten in ieder werkstation daalt van een maximale waarde totdat de steady state weer is bereikt. Tijdens het leeglopen van de lijn moeten alle overtollige producten het systeem verlaten. In dit proces kunnen de omstandigheden in de lijn tijdelijk enigszins verschillen van de omstandigheden in de steady state. Voor deze nieuwe omstandigheden kan een nieuwe 'steady state' ingesteld worden die verdwijnt als de eigenlijke omstandigheden weer terug zijn en de echte steady state weer bereikt wordt.

Om de steady state te bepalen is gebruik gemaakt van een benaderingsmethode gebaseerd op wachtrij modellen. Om de resultaten van deze benaderingsmethode te controleren zijn ze vergeleken met een steady state verkregen uit de resultaten van de discrete event simulatie.

Nadat het gedrag van een lijn werkstations is beschreven kan gezocht worden naar partiële differentiaal vergelijkingen die dit gedrag kunnen beschrijven. Een eerste stap naar het zoeken van partiële differentiaal vergelijkingen kan zijn om het beschreven gedrag te modelleren met behulp van een eindige elementen methode.

Een eindige elementen methode verdeelt een domein in cellen en in iedere cel is de gemiddelde dichtheid bekend. Deze gemiddelde dichtheid wordt in elke tijdstap en in elke cel aangepast. Het aanpassen gebeurt met behulp van de ingaande flux en de uitgaande flux. Deze fluxes kunnen beschreven worden met behulp van een functie die alleen afhankelijk is van de gemiddelde dichtheid in een cel. In dit onderzoek is de uitgaande flux bepaald met de resultaten van de benaderingsmethode voor de steady state.

Contents

Chapter 1

Introduction

Modeling and control of manufacturing systems can be a complicated matter. Nowadays there are roughly three groups of models which can be used for modeling and control of manufacturing systems. These are fluid models, discrete event models and queuing models.
All of these groups of models have their shortcomings. Fluid models are more focused on throughput than cycle time, these models neglect the relation between throughput and cycle time. Discrete event models on the other hand do model a relation between the throughput and the cycle time, but calculation time is long for large systems. So discrete event models are most suitable for small manufacturing systems. The third group are queuing models. Queuing models include a relation between the throughput and cycle time, but a shortcoming can be found in the fact that these models only describe the steady state of a manufacturing system and they are not suitable for control theory.
To overcome these shortcomings a fourth group of models might be considered, the partial differential equation (PDE) models. A PDE model describes the flow of products through a system as a continuous flow. The number of lots is assumed to be continuous as well as the position of a lot in the manufacturing system. A PDE model describes the dynamics of a system, gives a relation between the throughput and flow time and it is also possible to apply control theory on these models.

1.1 Objective

For determining if a PDE model might be considered for a manufacturing system with finite buffersizes, the behavior of a manufacturing system must be known. The determination of this behavior is the goal of this research. In order to know the behavior a PDE model must describe, derive properties for the behavior of a tandem queue with finite buffersizes.

For the behavior of the tandem queue the steady state must be considered as well as the dynamical behavior of the system. The steady state might be determined with an approximation from queuing theory and dynamical behavior can be determined with discrete event simulation. With discrete event simulation an experiment has to be derived which describes some transient behavior of the tandem queue with finite buffersizes, e.g. ramp up and the effect of blocking a workstation.

1.2 Approach

As a start of this research traffic flow models are studied. PDE models are already used for modeling traffic flow and due to similarities between traffic flow and the flow of products in a manufacturing system these models might be used for manufacturing systems. An overview of these traffic flow models is given in Chapter 2.
After these models the behavior of a manufacturing system with finite buffers is examined. A discrete event model of a tandem queue is determined and simulation experiments are performed. In this experiment the dynamical behavior is examined, e.g. movement of blocking through the system after a machine in the tandem queue breaks down. The discrete event model, the experiment and the results are described in Chapter 3.
Not only the dynamical behavior is of importance also the steady state of a manufacturing system must be described by a PDE model. Steady state is determined with the help of an approximation method based on queuing theory. A description of the approximation method and the results are given in Chapter 4.
In Chapter 5 finite volume methods are explained. These models can be used to determine a solution of a PDE model and might be used as a first step in determining a PDE model.
At the end of this report a conclusion is given and recommendations for further research.

Chapter 2

Partial Differential Equation models

As a start of our search for a PDE model for manufacturing systems with finite buffer sizes research is done on traffic flow models. Due to similarities between traffic flow and the flow of products in a manufacturing system these models might be used for manufacturing systems. The cars in these models can be seen as products and the road on which the cars drive can be seen as a manufacturing system.
A PDE model for traffic flow describes the flow of cars on a road as a continuous flow, when these models are used for manufacturing systems the flow of products is also described as a continuous flow. The number of lots and the position of a lot in the manufacturing system is assumed to be continuous. In the next paragraph a general explanation is given on PDE models and after that an overview is given of commonly used traffic flow models.

2.1 Basics of a PDE model

All PDE models are based on the mass conservation law and some other basic principles [Lef05]. These will be explained in this paragraph.
A PDE model describes the flow of cars through a system as a continuous flow. The number of cars on a part of the road and the position of the cars on a part of the road are continuous.
To describe this flow of cars on a part of the road three variables are used which can vary by time and place:

- the flow of cars, $u(x, t)$.
- the density, $\rho(x, t)$.
- the velocity, $v(x, t)$.

These variables are related to each other. The flow is the product of the density and the velocity:

$$u(x, t) = \rho(x, t)v(x, t).$$

Another basic relation is the mass conservation law:

$$\frac{\partial \rho}{\partial t}(x, t) + \frac{\partial u}{\partial x}(x, t) = 0.$$

Any PDE model must satisfy the mass conservation law. This conservation law means that on a part of the road cars can not be lost and no cars are created. The number of cars can only vary in time because of the inflow at the beginning of the road and the outflow at the end of the road.
In these basics the road can be replaced by a manufacturing system and the cars can be replaced by products. Subsequently these basics can be used for manufacturing systems. These two relations do not describe the complete model but are two basic relations that a PDE model must satisfy. At least one more relation is needed for a complete model. In the next section an overview is given of commonly used traffic flow models based on the previous basic equations.

2.2 Overview of traffic flow models

In the previous section two basic equation of traffic flow models are explained and in this section an overview is given of commonly used traffic flow models, each with a short explanation. All models are based on the basic equations in the previous section and most of these models can be found in [Hel01],[Hel96].
After the overview of traffic flow models a general density equation and velocity equation of traffic flow models is given. Most of the described models can be seen as a special form of these general equations.

2.2.1 Lighthill-Whitham

One of the first traffic flow models is proposed by Lighthill and Whitham [Lig55]. They used the mass conservation law and a static relation for the velocity:

$$V(x, t) = V_e[\rho(x, t)].$$

Combining this equation for the velocity and the mass conservation law a model is obtained which describes the propagation of kinematic waves:

$$\frac{\partial\rho}{\partial t} + \left[V_e + \rho\frac{\partial V_e}{\partial\rho}\right]\frac{\partial\rho}{\partial x} = 0.$$

As an equation for the velocity a linear relation can be chosen:

$$V_e(\rho) = V_{max}\left(1 - \frac{\rho}{\rho_{max}}\right).$$

In the model of Lighthill and Whitham shock fronts can be developed. To avoid the development of these shock fronts a diffusion term can be added to this model. Whitham added a diffusion term and obtained the following model:

$$\frac{\partial\rho}{\partial t} + V_e\frac{\partial\rho}{\partial x} = -\rho\frac{\partial V_e}{\partial\rho}\frac{\partial\rho}{\partial x} + D\frac{\partial^2\rho}{\partial x^2}.$$

2.2.2 Payne

Lighthill and Whitham assumed a static relation for the average velocity, therefore this model is not suitable for non-equilibrium situations like stop-and-go waves. Payne adjusted this assumption and used a dynamic equation for the average velocity:

$$\frac{\partial V}{\partial t} + V\frac{\partial V}{\partial x} = -\frac{D(\rho)}{\rho}\frac{\partial\rho}{\partial x} + \frac{1}{\tau}[V_e(\rho) - V]$$

with

$$D(\rho) = -\frac{1}{2\tau}\frac{\partial V_e}{\partial\rho} = \frac{1}{2\tau}\left|\frac{\partial V_e}{\partial\rho}\right|.$$

In this model the term $V\frac{\partial V}{\partial x}$ is the transport or convection term. The term $-[\frac{D(\rho)}{\rho}]\frac{\partial\rho}{\partial x}$ is the anticipation term, this term models the reaction of the drivers. The term $\frac{[V_e(\rho)-V]}{\Delta t}$ is the relaxation term, this describes the adjustment of the velocity to the equilibrium velocity [Pay71],[Pay79].

2.2.3 Phillips

Phillips replaced Payne's equation of $D(\rho)$ by $D(\rho) = \frac{\partial P}{\partial\rho}$:

$$\frac{\partial V}{\partial t} + V\frac{\partial V}{\partial x} = -\frac{1}{\rho}\frac{\partial P}{\partial x} + \frac{1}{\tau(\rho)}[V_e(\rho) - V].$$

The quantity P is the traffic pressure, $P(x,t) = \rho(x,t)\theta(x,t)$, in which $\theta(x,t)$ denotes the velocity variance of the vehicles. For this velocity variance Phillips assumed a variance-density relation such as $\theta(x,t) = \theta_0[1 - \rho(x,t)\rho_{jam}]$ [Phi79a], [Phi79b].

2.2.4 Kühne and Kerner-Konhäuser

Kühne and Kerner and Konhäuser used $\theta(x,t) = \theta_0$, but this leads to the development of shock waves like the model of Lighthill and Whitham. Therefore Kühne and Kerner and Konhäuser added a viscosity term. Kühne used $\nu\frac{\partial^2 V}{\partial x^2}$ as a viscosity term and Kerner and Konhäuser used $(\frac{\eta}{\rho})\frac{\partial^2 V}{\partial x^2}$ as a viscosity term [Nag02], [Ker93], [Ker94].

Kerner-Konhäuser:

$$\frac{\partial V}{\partial t} + V\frac{\partial V}{\partial x} = -\frac{\theta_0}{\rho}\frac{\partial \rho}{\partial x} + \frac{\eta}{\rho}\frac{\partial^2 V}{\partial x^2} + \frac{1}{\tau}(V_e(\rho) - V).$$

Kühne:

$$\frac{\partial V}{\partial t} + V\frac{\partial V}{\partial x} = -\frac{\theta_0}{\rho}\frac{\partial \rho}{\partial x} + \nu\frac{\partial^2 V}{\partial x^2} + \frac{1}{\tau}(V_e(\rho) - V).$$

2.2.5 Berg

Berg presented a model which is derived from an optimal velocity model and is equivalent to the model of Kerner and Konhäuser. The difference between these models are the coefficients, the coefficients of Berg's model are based on the parameters of the microscopic model while the coefficients of Kerner and Konhäuser's model are phenomenological [Nag02],[Ber00].

$$\frac{\partial v}{\partial t} + v\frac{\partial v}{\partial x} = \frac{1}{\tau}\left(V(\rho) - v\right) + \frac{V'(\rho)}{\tau}\left[\frac{1}{2\rho}\frac{\partial \rho}{\partial x} + \frac{1}{6\rho^2}\frac{\partial^2 \rho}{\partial x^2} - \frac{1}{2\rho^3}(\frac{\partial \rho}{\partial x})^2\right].$$

The preceding traffic flow models are found in literature and most of these models are closely related to each other. They can be written in a general form, this general form can be found in the next paragraph.

2.3 General form of traffic flow models

Previously some traffic flow models are described and a short explanation of these models is given. These models have some differences, but the models can all be written in a general form except the model of Berg. The models can be seen as a special form of the next two equations.

The density equation:

$$\frac{\partial \rho}{\partial t} + V\frac{\partial \rho}{\partial x} = -\rho\frac{\partial V}{\partial x} + D(\rho)\frac{\partial^2 \rho}{\partial x^2} + \xi_1(x,t).$$

The velocity equation:

$$\frac{\partial V}{\partial t} + V\frac{\partial V}{\partial x} = -\frac{1}{\rho}\frac{dP}{d\rho}\frac{\partial \rho}{\partial x} + \nu\frac{\partial^2 V}{\partial x^2} + \frac{1}{\tau}(V_e - V) + \xi_2(x,t).$$

The differences between the models are based on the different definitions of the diffusion $D(\rho)$, the fluctuations $\xi_1(x,t)$, $\xi_2(x,t)$, the traffic pressure $P(\rho)$, the viscosity-like quantity $\nu(\rho)$, the relaxation time $\tau(\rho)$ and the equilibrium velocity $V_e(\rho)$ [Hel01].

In this chapter the basics of PDE models are described and an overview of commonly used traffic flow models is given. Only before a PDE can be found which describes the behavior of a manufacturing system the behavior must be known. Even though discrete event simulation and queuing theory have their shortcomings the results of these models can be used to understand the behavior a PDE model must describe. For this reason in the next chapter the dynamical behavior is determined with discrete event simulation. After that the steady state is determined by means of queuing theory.

Chapter 3

Discrete event simulation

Discrete event simulation can be used to describe the dynamical behavior of a manufacturing system, e.g. the ramp up of a system. A PDE has to describe the dynamical behavior of a system and in this chapter simulation experiments are done with a discrete event model in order to show the dynamical behavior which has to be described by a PDE. These simulations are done for a tandem queue with identical machines for different parameter settings.
In the next paragraph a discrete event model for a tandem queue of identical machines is derived in Chi [Ver04],[Hof02]. This discrete event model is used to perform a simulation experiment which describes different kinds of dynamical behavior. After the description of the model this experiment is explained and results are shown.

3.1 Discrete event model

Discrete event simulations can be used to describe dynamical behavior of a manufacturing system with finite buffers. In this report a simulation experiment is done to determine the dynamical behavior of a tandem queue. For the experiment is chosen to do a discrete event simulation which shows the behavior of the tandem queue in three different situations. The first situation is at the start of the experiment, there are no products in the system and a ramp up of the system is simulated. The second situation occurs after several time-units when the system is in steady state and the last machine in the system breaks down. As a result, eventually all workstations will get blocked. The third situation is the behavior of the system after unblocking the last machine.
To perform this experiment a discrete event model of a tandem queue is needed in which the last machine breaks down and restarts again at a certain moment of time. This model is described here and a schematic view is given in Figure 3.1, showing the structure of the discrete event model.

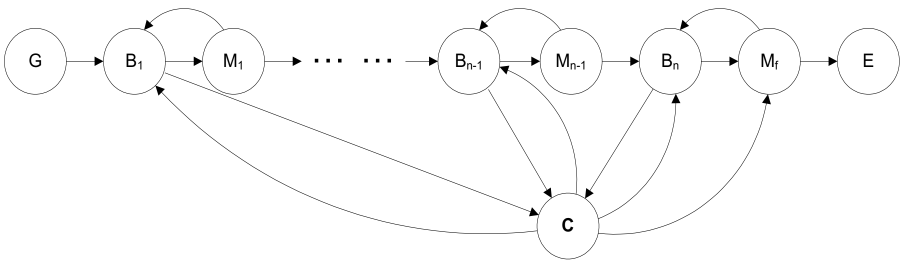

Figure 3.1: Schematic view of the discrete event model

The model contains a generator, an exit process, n buffers, n machines and a controller. A machine and the buffer in front of the machine together form a workstation. The value of n, number of workstations, can be set in the model before a simulation starts. The generator in the model is used to send products into the tandem queue and the exit receives the finished products of the last workstation in the queue. The buffer has a maximal number of products which can be stored.As long as that maximal value is not yet reached it receives products from the generator or the previous workstation. The buffer sends these products to the machine when the machine is empty. The machine processes the product and sends the product to the following machine or the exit. After sending the product away the machine sends a signal to the buffer in front of the machine to let know that the machine is empty. This is done because the buffer registers how many products are stored in the workstation.
The machines in the tandem queue are all identical except the last machine in the tandem queue. The last machine in the tandem queue has a small difference in the way it is modeled. In this machine blocking occurs at a certain moment of time and after several time-units the machine starts processing again. These moments of blocking and start processing again are send from the controller to the machine and to make this possible an extra channel is needed and the Chi-code of this process is somewhat different than the other machines. The process time and other parameters of the machine are the same as the other machines in the tandem queue.
The buffers in the line are all identical and in the model the maximal number of products in the buffers can be set. Only for the tandem queue with a buffersize of $b = 0$ the process of the buffers is modeled different from the case of $b > 0$. The process of the buffer must not store any products when $b = 0$.
The controller in this model is used to register the parameter of interest, the number of products in a workstation. Every time-unit the controller determines the number of products in every workstation and places this information in a tuple and puts the tuple into a list. After a simulation the list of tuples is written and every tuple in the list represents the number of products for the workstations at a certain time.
In the model several parameters must be set. The number of machines can be set in the model and when starting the model some variables must be provided. These variables include values for the distribution of the inter-arrival time at the generator, the

distribution of the process time for the machines and the maximal number of products in a buffer.
Other parameter values which have to be specified at the prompt are some moments of time. In the last machine blocking occurs at a certain moment of time and after the machine is down for some time-units it starts working again. These times must be set at the prompt and the last one is the time the simulation needs to be finished. The inter-arrival time, the process times and the maximal number of products can be set by an integer, the moments of times can be set by a list with a tuple for every event. In this tuple the time is set and the event which has to take place, events can be defined as start blocking or start processing.
The complete model can be found in Appendix B. In the next section the experiment done with the previous model is described and after that the results of the discrete event simulation are presented.

3.2 Set up of simulation experiments

In the previous section a model is described to perform a simulation experiment. In this experiment three different kinds of dynamical behavior are described. The first one is a ramp up experiment. The second one is dynamical behavior after the last machine breaks down when the system is in steady state, blocking is then moving through the tandem queue. The third one is dynamical behavior after restarting the last machine.
The experiment is done for two tandem queues, a tandem queue of 100 workstations and a tandem queue of 10 workstations. For both tandem queues several simulations are done with different values for the mean inter-arrival time at the generator (t_a), the squared coefficient of variation (c_a^2) and the maximal number of products in the buffer (b). The mean process time for the machine (t_e) and the squared coefficient of variation (c_e^2) are equal in all simulations. These values are given in Table 3.1 and the experiment is done for all combinations of the values in the table.

To perform the experiment not only the mean inter-arrival time at the generator, the squared coefficients of variation and the maximal number of products in the buffer are needed but also some moments of time have to be defined. These moments are the moments of blocking the last machine, restarting the process again and the end of the simulation. Blocking the last machine occurs when the system is in steady state, restarting the process again happens when all workstations are blocked and the simulations end when the system is back in steady state again. The moments used in the simulations are shown in Table 3.2. In Table 3.2 also the number of simulations is given.

The simulation experiments are done with different values for the mean inter-arrival time at the generator, the squared coefficient of variation and the maximal number of products in the buffer, but in all simulations the same sort of behavior can be expected

Tandem queue			
number of machines	10	100	
maximal buffersize	0	1	5
Generator			
inter-arrival time	1.0	2.0	
squared coefficient of variation	0.1	1.0	5.0
Machines			
mean process time	1.0		
squared coefficient of variation	1.0		

Table 3.1: Simulation parameters with their values.

buffersize=0 or buffersize=1 and $t_a = 1$ or $t_a = 0$	
time	state of last machine
0-1500	up
1500-2000	down
2000-5000	up

buffersize=5 and $t_a = 1$	
time	state of last machine
0-3000	up
3000-4000	down
4000-11000	up

buffersize=5 and $t_a = 2$	
time	state of last machine
0-3000	up
3000-5000	down
5000-12000	up

number of simulations	2000

Table 3.2: Moments of break down and restart of the last machine and number of simulations.

as a result of blocking the last machine and restarting it. At the start of the experiment the manufacturing system is empty and a ramp up experiment is performed. The mean number of products for the workstations must increase to reach a steady state. After the steady state is reached for all workstations the last machine in the tandem queue breaks down. Expected is that the mean number of products in the last workstation increases

to a maximal value. The moment the last workstation reaches the maximal value the previous workstation can not send away finished products and this workstation is also blocked. The mean number of products in this workstation increases until it reaches the maximal value and the process repeats for other previous workstations until it reaches the first workstation. This movement of blocking should be seen in the results. The mean number of products increase to a maximal value for every workstation in an order from the last workstation to the first workstation. When all workstations are blocked and have reached their maximal values the last machine starts processing again. In the results the mean number of products in all workstations should decrease until it reaches steady state again.
Not only the mean number of products is of importance but the velocity of blocking moving through the tandem queue is another aspect a PDE has to describe. The velocity by which blocking moves through the tandem queue will be determined with the results of the discrete event simulation. From the results of a simulation experiment the number of products in a workstation is known every time-unit. From this information can be determined when the last workstation in the tandem queue reaches the maximal number of products in the workstation for the first time after the last machine is blocked. This moment of time is stored and for the previous workstation in the tandem queue is searched when the maximal number of products is reached for the first time. The search for the maximal number of products in this workstation starts at the moment of time found in previous search. This is repeated for every workstation and the result is a list of moments of time every workstations reaches the maximal number of products. This is done for 1000 simulations and the mean of these points of time is determined. In this determination the search for the maximal number of products in a workstation starts at the moment of time found in the previous search. This assumes that the workstations are filled with products in an order from the last workstations until the first workstation. But in the results of the discrete event simulation it might be possible that a workstation reaches the maximal number of products earlier than a workstation further in the tandem queue. This is neglected in this determination because this is not a consequence of blocking the last workstation.
The results of the simulation experiments are given in the next section.

3.3 Results of the discrete event simulation

In the previous sections the discrete event model is described and the experiment is explained. In this section the results of the discrete event simulation are shown. The purpose of the simulations is to determine dynamical behavior of a tandem queue in order to known the behavior which has to be described by a PDE. One of the aspects of the dynamical behavior that a PDE must describe is the mean number of products during different situations, for example during ramp up of a tandem queue. Besides the mean number of products another aspect of the dynamical behavior is the movement of blocking through the tandem queue.

These two aspect are discussed here, first the mean number of products and after that the movement of blocking in the tandem queue is examined. In the appendices the programming codes are given to determine the results.

Mean number of products In Figure 3.2 results are shown of a tandem queue of 10 machines with an inter-arrival time of $t_a = 1$ or $t_a = 2$, $c_a^2 = 1$,$t_e = 1$, $c_e^2 = 1$, $b = 0$ or $b = 1$ or $b = 5$ and in Figure 3.4 the results for the same values of a tandem queue of 100 machines are shown.
In Figure 3.2 the mean number of products are shown for three workstations in the system, workstation 1, workstation 5 and workstation 10, and in Figure 3.4 the mean number of products are shown for workstation 1, workstation 50 and workstation 100. The other workstations in the tandem queues are not shown but the mean number of products in the system develops in a similar way as workstation 5 or workstation 50.

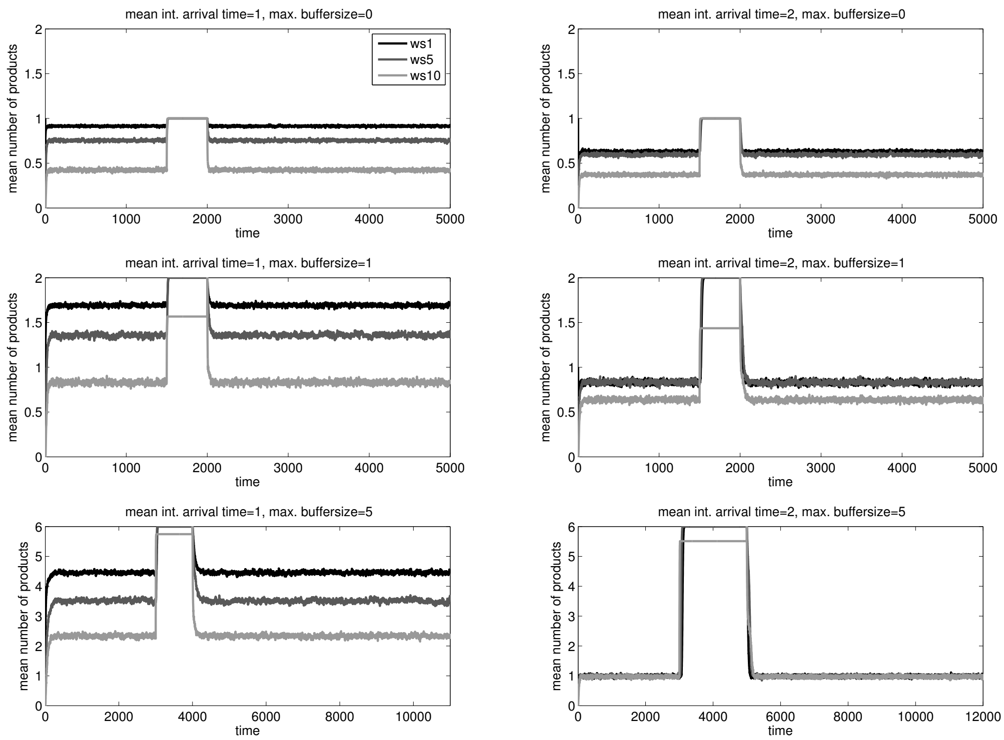

Figure 3.2: Mean number of products in a tandem queue of 10 workstations, $t_a = 1$ or $t_a = 2$, $c_a^2 = 1$,$t_e = 1$, $c_e^2 = 1$, $b = 0$ or $b = 1$ or $b = 5$.

Although in Figure 3.2 and Figure 3.4 results are shown for two different tandem queues with different inter-arrival times and maximal buffersizes the results show similar be-

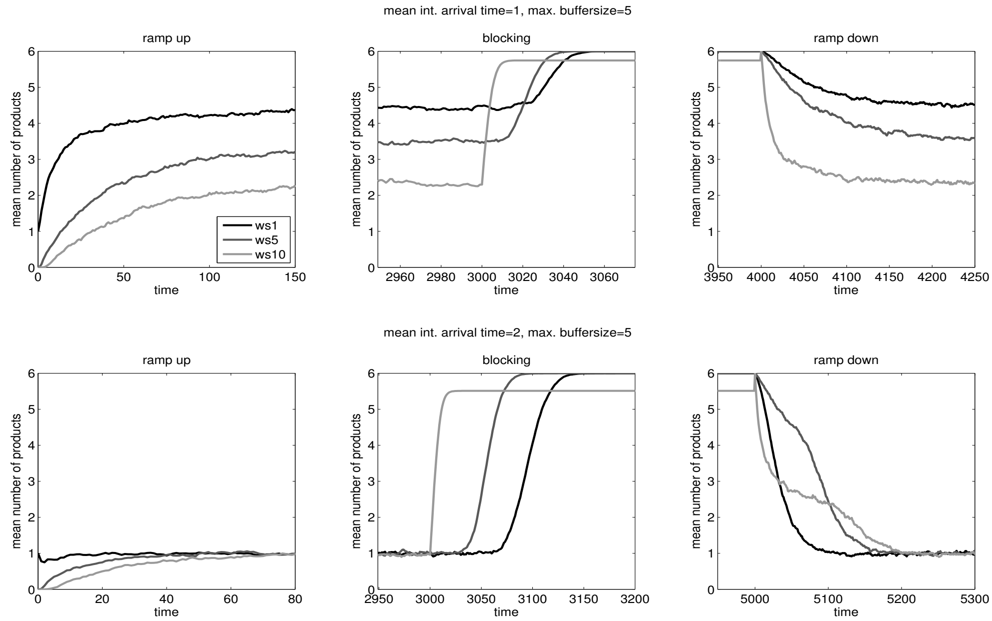

Figure 3.3: Mean number of products in a tandem queue of 10 workstations on a small time scale, $t_a = 1$ or $t_a = 2$, $c_a^2 = 1$, $t_e = 1$, $c_e^2 = 1$, $b = 5$.

havior. This behavior corresponds to the expected behavior described earlier in Section 3.2.

In both figures the tandem queues are empty at the start of the experiment. From $time = 0$ the number of products in workstation 1 increases until it reaches steady state and the mean number of products stays at the same level. For the other workstations increasing of the mean number of products starts a little later because products have to go through the tandem queue before it reaches workstation 5 and workstation 10 or workstation 50 and workstation 100. Eventually all workstations are in steady state when the last workstation breaks down and the mean number of products in the last workstation increases to a maximal value. The moment the last workstation reaches the maximal value the previous workstation can not send away finished products and this workstation is also blocked. The mean number of products increases until it reaches the maximal value and the process repeats for other workstations until it reaches the first workstation. In this way the effect of blocking moves through the tandem queue until it reaches the first workstation. Because all workstations are blocked the mean number of products stays equal to the maximal value they have reached until the last machine starts working again. From that moment the mean number of products in the workstations starts decreasing immediately until it reaches steady state again. The mean number of products is now the same as before blocking the last machine. In the

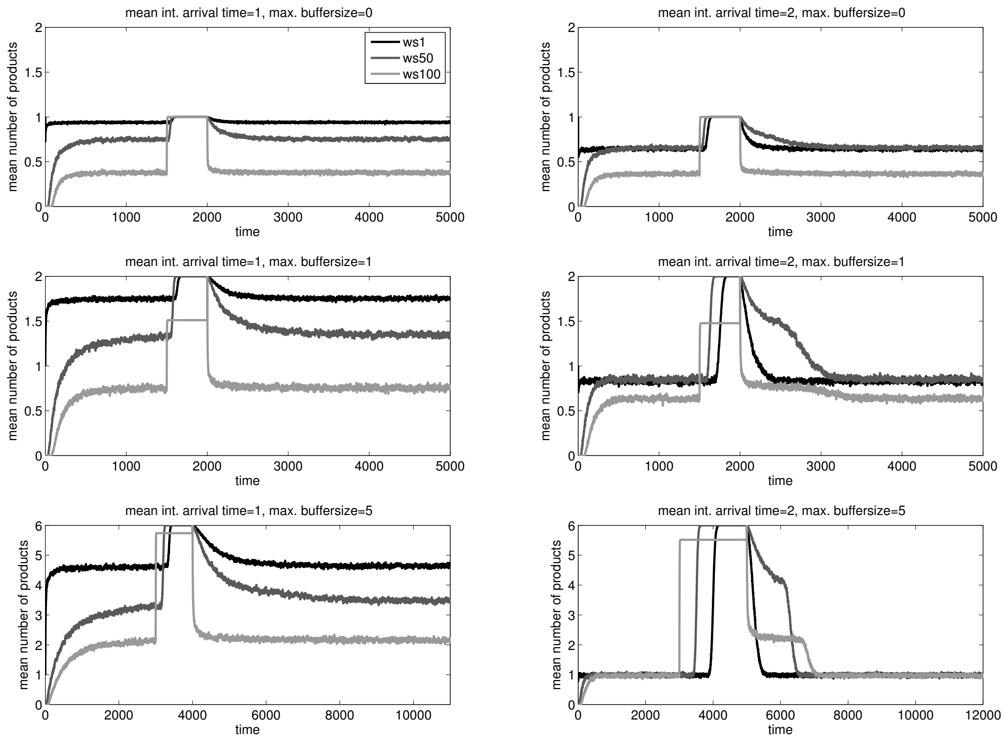

Figure 3.4: Mean number of products in a tandem queue of 100 workstations, $t_a = 1$ or $t_a = 2$, $c_a^2 = 1$,$t_e = 1$, $c_e^2 = 1$, $b = 0$ or $b = 1$ or $b = 5$.

results can be seen that the ramp down of the system takes more time than the ramp up of the system.

In Figure 3.4 it can be seen that first workstation 1 receives products in the ramp up experiment, then workstation 50 receives products and after that workstation 100. The order in which the workstations are blocked can also be seen, first workstation 100 reaches a maximal value, then workstation 50 and finally workstation 1. For the tandem queue of 10 workstations this effect is present in the results but in Figure 3.2 it is too small to determine. Therefore in Figure 3.3 parts of the results of a tandem queue of 10 workstations with a maximal buffersize of 5 are shown again on a smaller time scale. The behavior in this figure is similar to the behavior in Figure 3.4 for a tandem queue of 100 workstations with a maximal buffersize of 5. For the other graphs in Figure 3.2 the results are not shown on a smaller time scale, but the behavior is also similar to the behavior in Figure 3.4.

In Figure 3.4 and Figure 3.3 two points attract attention in these results. The first one is the fact that during the time that the workstations are blocked the mean number

of products for the last workstation is less than the mean number of products in the other workstations and when the machine is unblocked the mean number of products increases to the same value before decreasing to the steady state. The mean number of products in all workstations except the last one reaches a value of the maximal buffer size plus 1 product in the machine. For the last workstation this is different because on the moment of blocking the last machine can be empty or a product can be in the machine. If a product is in the machine the maximal number of products is equal to the maximal buffer size plus 1 product in the machine, like in all workstations. But if the machine is empty on the moment of break down the maximal number of products is equal to the maximal buffer size. So this is one product less and results in a lower mean number of products.
The moment the machine starts working again the mean number of products increases and reaches the same value as for the other workstations. During the moments of blocking the last machine is sometimes empty, but on the moment processing starts again the machine immediately receives a product from the buffer. An empty place in the buffer is immediately filled with the finished product of the previous machine, this is the reason for the increasing number of products at the time the machine starts processing again.
The previously described effect can only be seen in the results of tandem queues with a maximal buffersize of $b = 1$ and $b = 5$, for a maximal buffersize of $b = 0$ the process of the buffer is modeled a little different in the discrete event model. In the model of $b = 0$ the last workstation can, in some cases, receive one product after blocking the last machine. When the machine breaks down with no product in the machine, the workstation can receive one product before this workstation is full. Receiving a product in this case should not be possible. This happens only in the last workstation for tandem queues with $b = 0$, the effect on the results is small.
The second point of attention in Figure 3.4 and Figure 3.3 is the fact that there are differences in the way that the mean number of products decreases after unblocking the last machine. The tandem queues with $t_a = 2$, $c_a^2 = 1$, $t_e = 1$, $c_e^2 = 1$, $b = 1$ and $b = 5$ show some different behavior for the mean number of products of workstation 50, workstation 100 and workstation 5, workstation 10. The mean number of products decreases but there is a part for which it decreases slower. The reason is that during blocking workstations are completely filled, and all these extra products must leave the system after starting the process again. This means for the machines that every time a product is finished and send away to the next workstation it immediately receives a new product which has to be processed. For these machines the arrival rate is now temporarily higher than the arrival rate in the steady state before blocking. Another result of a full system is the fact that often when workstation 50 and workstation 5 are finished they can not send away the finished products because the following workstation is full. So after the last workstation starts working again blocking still occurs in the system. For these new conditions a new 'steady state' occurs until most of the extra products has left the system and the conditions are again equal to the conditions in the real steady state.
This reasoning can not only be applied to workstation 50, workstation100 and worksta-

tion 5, workstation 10 but for other workstations in the system the same can happen.

In Figure 3.5 and 3.6 results of a simulation of a tandem queue of 100 machines with a maximal buffersize of 5 are given in a different way than in Figure 3.4. In Figure 3.4 the mean number of products in a certain workstation is plotted against time, but in figures 3.5 and 3.6 the mean number of products in all workstations are shown for different moments of time.
In Figure 3.5 the results are shown of a tandem queue with $t_a = 1$, $c_a^2 = 1$, $t_e = 1$, $c_e^2 = 1$ and $b = 5$. The figure on the left shows the first 3000 time-units in the simulation experiment, this is the ramp up of the system. The figure in the middle shows the mean number of products between $time = 3000$ and $time = 4000$, in this period blocking moves through the tandem queue. The figure on the right gives a view from $time = 4000$ and further, this is a view on how the system returns to the steady state after the last workstation start working again. In Figure 3.6 the results are shown of a tandem queue with $t_a = 2$, $c_a^2 = 1$, $t_e = 1$, $c_e^2 = 1$ and $b = 5$. Similar behavior is shown, only the moment of restarting the last workstation is different than in Figure 3.5.

Blocking Until now the results of the discrete event simulation are given by means of the mean number of products in the system, but not only the mean number of products is important also the velocity of blocking can be of importance.
In the simulation experiment the last machine in the system breaks down when the system is in steady state. The last workstation can not process any products and puts all products in the buffer until this buffer is full. As a consequence the machine in front of this buffer can not send away the finished products and this machine is also blocked. This process keeps repeating until all workstations in the tandem queue are blocked.
After blocking the last machine a front is formed in the system, on one side the velocity of products is zero and on the other side the velocity is larger than zero. This front moves with a certain velocity from the last workstation of the tandem queue until it reaches the first workstation. This velocity is determined and it can be used as a validation parameter for PDE's.
The determination of this velocity is explained in Section 3.2. The determination of the velocity results in a list of moments of time. For every workstation the moment of time is determined on which the workstation is completely filled with products. This is the moment the previous machine gets blocked.
In Figure 3.7 and 3.8 these moments of time are plotted. This is done for the tandem queue of 10 workstations and 100 workstations. The values for the mean inter-arrival time, the squared coefficient of variation and the maximal number of products in the buffer are shown in the figure.

In Figure 3.7 and 3.8 can be seen that for both tandem queues with different values for the simulation parameters the same conclusion can be drawn. In the tandem queue blocking moves through the system with an almost constant velocity.

This chapter describes the dynamical behavior of a tandem queue in order to know the dynamical behavior a PDE must describe. Not only the dynamical behavior is of

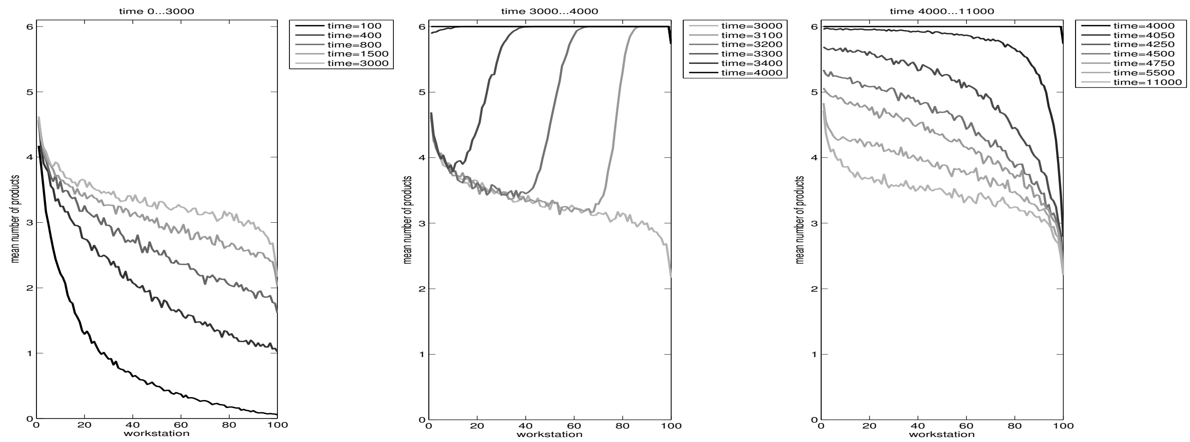

Figure 3.5: Mean number of products in workstation, $t_a = 1$, $c_a^2 = 1$, $t_e = 1$, $c_e^2 = 1$, $b = 5$.

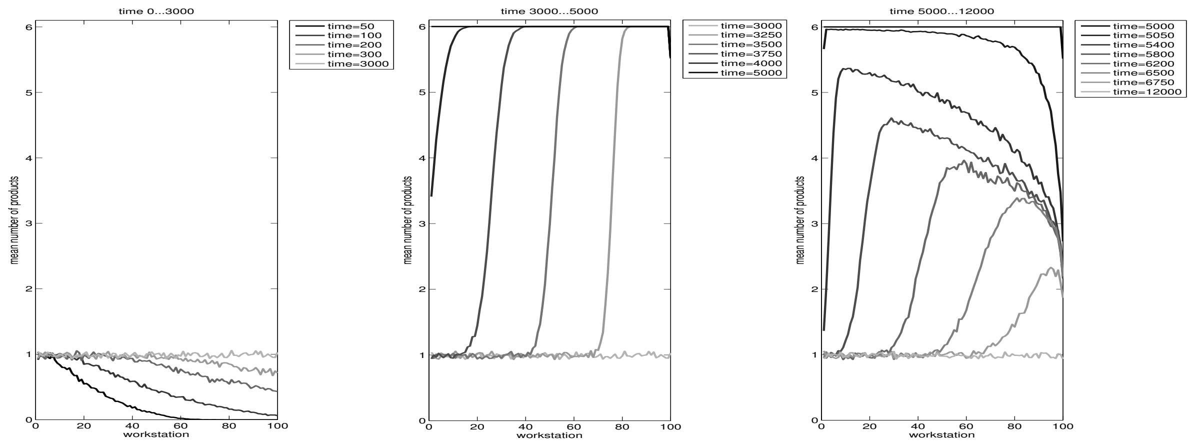

Figure 3.6: Mean number of products in workstation, $t_a = 2$, $c_a^2 = 1$, $t_e = 1$, $c_e^2 = 1$, $b = 5$.

importance also the steady state of a tandem queue has to be described by a PDE. Discrete event simulation can be used to determine the steady state, but it takes a long time. In the next chapter an approximation method based on queuing theory is used to determine the steady state in a faster way than discrete event simulation. The results of the discrete event simulations in this chapter are used to validate the steady state of the approximation method in the next chapter.

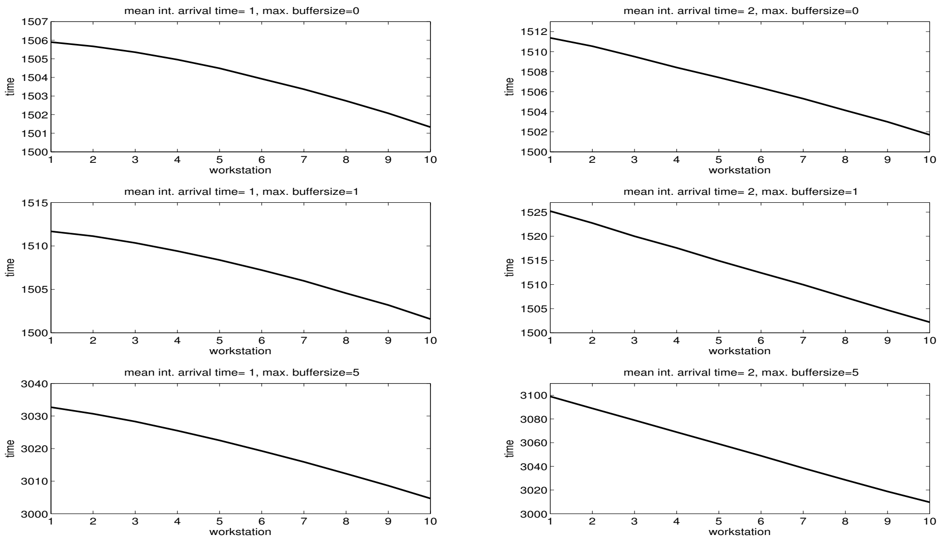

Figure 3.7: Moment of reaching the maximal number of products in a workstation in a tandem queue of 10 workstations,, $t_a = 1$ or $t_a = 2$, $c_a^2 = 1$, $t_e = 1$, $c_e^2 = 1$, $b = 0$ or $b = 1$ or $b = 5$.

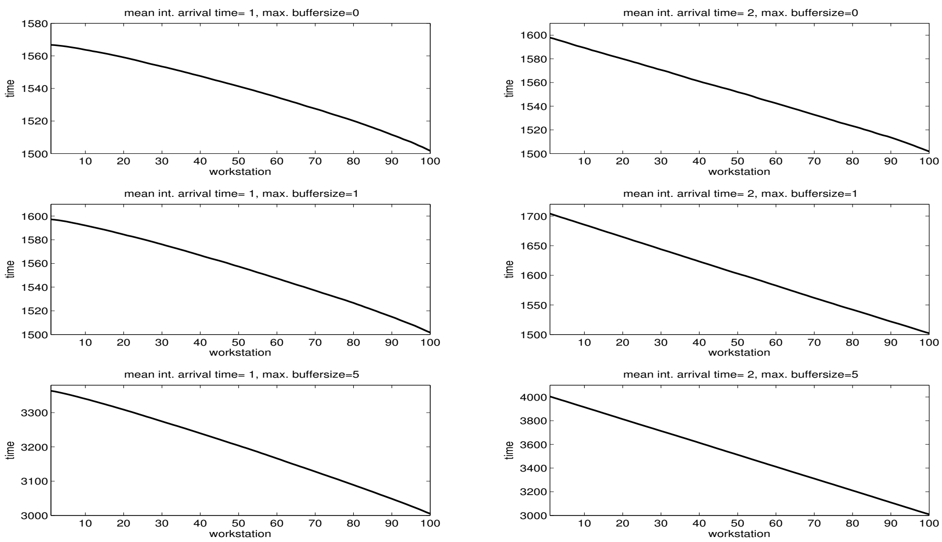

Figure 3.8: Moment of reaching the maximal number of products in a workstation in a tandem queue of 100 workstations,, $t_a = 1$ or $t_a = 2$, $c_a^2 = 1$, $t_e = 1$, $c_e^2 = 1$, $b = 0$ or $b = 1$ or $b = 5$.

Chapter 4

Steady state

In the previous chapter the dynamical behavior of a manufacturing system is described by means of discrete event simulation. The discrete event simulation describes the dynamical behavior in a tandem queue with 100 identical machines and a tandem queue of 10 identical machines. To validate results of a PDE not only the dynamical behavior is of importance but also the steady state of a manufacturing system has to be examined. This chapter is about the steady state of a tandem queue with finite buffers, more particularly the mean number of products in a workstation in steady state.

4.1 Determination of the steady state

The steady state is determined in two different ways. The first one is an approximation method based on queuing theory and for the second the results of the discrete event simulation in Chapter 3 are used. In the next two paragraphs an explanation is given on the approximation method and the discrete event simulation. After that the results are compared to each other.

4.1.1 Steady state with approximation algorithm

For most queuing systems with finite buffers it is not easy to determine performance characteristics precisely, for this reason an approximation method based on queuing theory is used in this paragraph. In this case the approximation method in [Vuu05] is used.
The approximation method determines the mean number of products, the throughput and the cycle time for a single-server tandem queue. Input parameters for this approximation method are mean process times, the squared coefficient of variation and the maximal buffersize.

The algorithm is based on decomposition of a single-server tandem queue and is explained in this section. In Figure 4.1 a single-server tandem queue is shown.

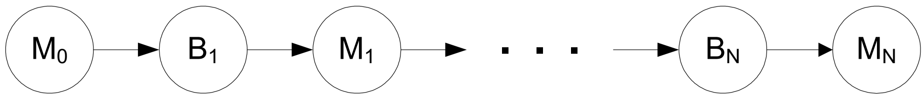

Figure 4.1: Single-server tandem queue.

The tandem queue exists of $N + 1$ machines and N buffers between the machines. The machines can only process one product at a time and the first machine in the queue always has a product to serve. This machine can be seen as the generator of this tandem queue. To determine the number of products in every workstation in steady state the tandem queue is decomposed into N subsystems. Every subsystem consists of a buffer, an arrival-server in front of the buffer and a departure-server after the buffer. The arrival-server represents not only the process time of the machine in front of the buffer but it includes possible starvation of the machine in front of the buffer before service. For the departure-server a similar description can be given, this represents the process time including possible blocking of this server.

The process time in the arrival-server is modeled according to the following description. In modeling the arrival-server not only the process time of the previous machine is taken into account but also the possibility of starvation of the previous machine. This is done in the following way. In a subsystem two situations can occur on the moment a product leaves the subsystem. The subsystem is empty or the subsystem is not empty. These situations are both possible with a certain probability. In the situation of the empty subsystem the system has to wait a residual process time of the previous machine and then the process time in the subsystem itself can be started. In the situation of the subsystem which is not empty a product already is in the buffer and the process time at the machine can start immediately. So with a certain probability the service time in an arrival server is a residual process time of the previous machine plus the process time and with a certain probability the service time is equal to the process time.

Modeling the departure server is based on a similar reasoning, only there is a difference between the service time of a product for which the previous product leaves behind an empty subsystem and the service time of a product for which the previous product does not leave behind an empty subsystem. This difference is made because when time passes between a product leaving the system and the moment a new product starts processing, it is less likely that the new product will get blocked.

In the situation of the empty subsystem the service time starts when a product arrives at the subsystem, at that moment there are again two possibilities. The following buffer is completely full or not. If the buffer is full the service time of the departure server is equal to the max of the process time for the current product and the residual service time of the departure server in the following subsystem. If the buffer is not completely full the product can move to the next subsystem after processing, so the service time is

equal to the process time. So in the situation that the previous product leaves behind a empty subsystem there are two possibilities and these occur with a certain probability. In the situation when the subsystem is not empty the service time starts immediately and at that moment there are three possibilities, which depend on the situation in the next subsystem.
The first possibility is that the next subsystem is full and blocks the current subsystem. When a product leaves the next subsystem it can receive a product from the current subsystem. In this case both subsystems start processing a product at the same time and the service time is than equal to the maximum of the processing time and the service time of departure server of the next subsystem. The second possibility is that the next subsystem has one place left for a product and when it receives a product the subsystem is full. The next subsystem has already started the processing and it takes a residual process time to finish that product. The service time of the departure server is in this case equal to the maximum of the residual service time of the departure server and the process time. The last possibility is when the other two possibilities do not occur, the service time of the departure server is than equal to the process time.

The distribution of service time in the arrival server, and the two distributions in the departure server wil be approximated by fitting the first two moments.

4.1.2 Steady state with discrete event simulation

In the previous paragraph the steady state of a single-server tandem queue is determined with the help of the approximation method based on queuing theory of [Vuu05]. To verify the results of the approximation method discrete event simulation is used. The discrete event model in Chapter 3 is used to determine the steady state of a single-server tandem queue.
The results of the simulations in Chapter 3 are used to determine the simulation. In the performed experiment the manufacturing system is empty at the start and after several time-units the system is in steady state. When the steady state is reached the last machine breaks down and blocking occurs. Just before the last machine breaks down the system is in steady state and the mean number of products at that moment is compared to the results of the approximation method. In the next paragraph the results of the approximation method and the discrete event simulation are shown and compared.

4.2 Results

In the previous section two different ways of determining the steady state are given. An approximation method based on queuing theory is used and discrete event simulation. Both ways determine the steady state of a tandem queue with identical machines and the results of these methods should correspond. In order to find out if the results of

both methods correspond the results are compared to each other in this section.
Figure 4.2 shows the steady state determined with the discrete event simulation and the approximation method for a tandem queue of 10 workstations and Figure 4.3 shows the same results for a tandem queue of 100 workstations. For both tandem queues the results are shown in six different cases. The mean inter-arrival time, the squared coefficients of variation and the maximal number of products in the buffer are different in these cases. The values are shown in the figures.

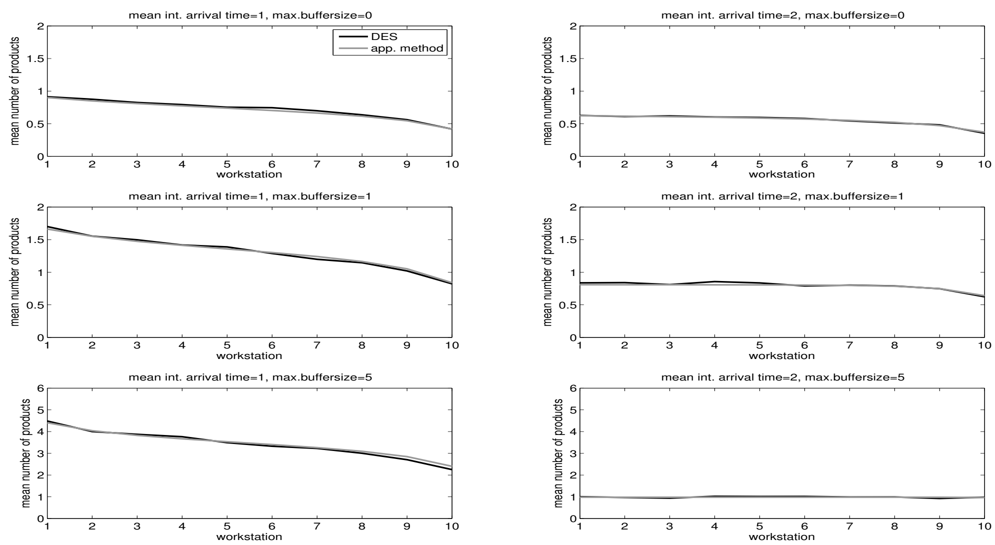

Figure 4.2: Mean number of products in a tandem queue of 10 workstations, $t_a = 1$ or $t_a = 2$, $c_a^2 = 1$, $t_e = 1$, $c_e^2 = 1$, $b = 0$ or $b = 1$ or $b = 5$.

From the results in figures 4.2 and 4.3 can be seen that there is a difference between the tandem queue of 100 workstations and the tandem queue of 10 workstations. For the tandem queue of 10 workstations the discrete event simulation determines the same steady state as the approximation method. Some small differences between the discrete event simulation and the approximation method can be caused by stochast behavior in the discrete event simulation. This might be solved by performing more simulations.
The results of a tandem queue of 100 workstations are different from the results of the tandem queue of 10 workstations. In the results for a tandem queue of 100 workstations the steady state determined with discrete event simulation and the approximation method do not correspond.

In this chapter the steady state of a tandem queue is determined and in Chapter 3 the dynamical behavior is determined. The steady state and the dynamical behavior both have to be described by a PDE. A first step in determining a PDE is to

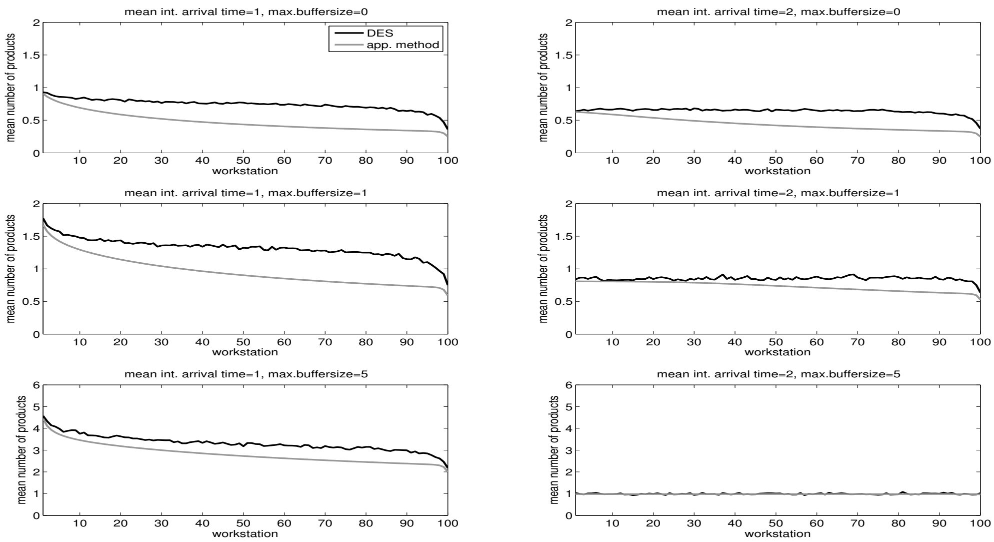

Figure 4.3: Mean number of products in a tandem queue of 100 workstations, $t_a = 1$ or $t_a = 2$, $c_a^2 = 1$, $t_e = 1$, $c_e^2 = 1$, $b = 0$ or $b = 1$ or $b = 5$.

take a look at finite volume methods. Finite volume methods are explained in the next chapter and the results of the approximation method are used. Because the results of a tandem queue of 100 workstations do not correspond to the results of the discrete event simulation only the results of the approximation method are used for the tandem queue of 10 workstations.

Chapter 5

Finite volume methods

In Chapter 3 and Chapter 4 the dynamical behavior and the steady state of a tandem queue are determined. In these chapters the discrete event system behavior is determined in order to know the behavior a PDE has to describe. A first step in determining a PDE is done in this chapter.
The solution of a PDE can be approximated with the help of finite volume methods. A finite volume method divides a domain into intervals and the average density in the intervals is updated every time-step. The update is done with the help of the density, the ingoing flux of the grid cell and the outgoing flux of the grid cell. The determination of these fluxes is of importance and this is done with the help of clearing functions.
In this chapter first the finite volume method is explained and after that clearing functions are explained.

5.1 Basics of finite volume methods

A first step in the determination of a PDE for manufacturing systems can be to describe the desired behavior first with a finite volume method. Finite volume methods approximate the solution of PDEs and can describe the same sort of behavior, like the movement of blocking. If the behavior of a tandem queue can be described by a finite volume method a PDE might be derived from the finite volume method.

Finite volume methods are based on subdividing the spatial domain into intervals, called grid cells [Lev03]. For every grid cell the initial average density is known and in every time step this average density is updated. This update is done with the help of the fluxes through the endpoints of the intervals.
In Figure 5.1 is shown how the domain is divided into grid cells in a finite volume method.

Figure 5.1 can be seen as a representation of the tandem queue from the earlier chapters. The start of the tandem queue is 0 in the domain and the end of the tandem queue

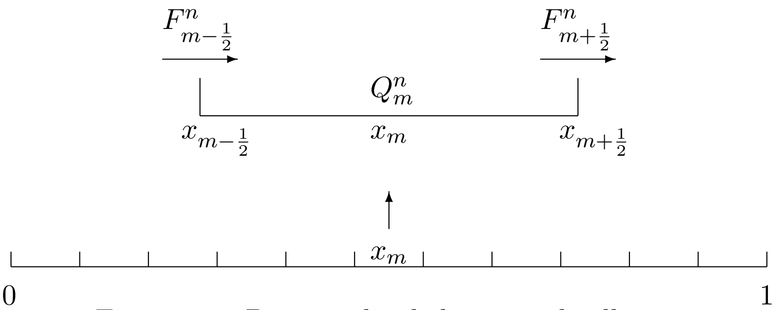

Figure 5.1: Domain divided into grid cells.

is 1. In between the domain is divided into grid cells and every grid cell represents a small group of workstations. Every grid cell has an average density. Q^n_m is the average density in the mth cell at time n.
When time passes products move through the tandem queue and the densities in the grid cells change. This is represented by the flux through the endpoints of the cells. $F^n_{m-\frac{1}{2}}$ and $F^n_{m+\frac{1}{2}}$ are the ingoing and the outgoing fluxes of the mth cell at time n.
In the finite volume method the average density is updated every time-step with the help of the ingoing flux and the outgoing flux. This update is explained below.

A finite volume method is based on subdividing the spatial domain into intervals as described earlier. Over each of these intervals an approximation of the integral of the density is determined and an average value for the density is approximated in every grid cell. This is done for several time steps, in each time step the average value is updated using approximations of the flux through the endpoints of the grid cells.
The value Q^n_m approximates the average value over the mth interval at time t_n,

$$Q^n_m \approx \frac{1}{\Delta x}\int_{x_{m-\frac{1}{2}}}^{x_{m+\frac{1}{2}}} \rho(x,t_n)dx. \tag{5.1}$$

In this formula $\Delta x = x_{m+\frac{1}{2}} - x_{m-\frac{1}{2}}$, this is the length of the cell.

The density in the grid cells will only change due to the fluxes through the endpoints of the grid cells. This leads to an integral form of the mass conservation law,

$$\frac{d}{dt}\int_{x_{m-\frac{1}{2}}}^{x_{m+\frac{1}{2}}} \rho(x,t)dx = f(q(x_{m-\frac{1}{2}},t)) - f(q(x_{m+\frac{1}{2}},t)). \tag{5.2}$$

After integrating and rearranging, this expression can be used to develop a numerical method of the form,

$$Q^{n+1}_m = Q^n_m - \frac{\Delta t}{\Delta x}(F^n_{m+\frac{1}{2}} - F^n_{m-\frac{1}{2}}). \tag{5.3}$$

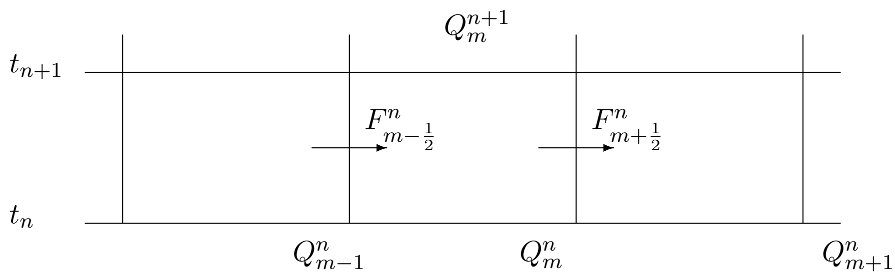

Figure 5.2: Illustration of a finite volume method for updating the cell average Q_m^n by fluxes at the cell edges.

Equation (5.3) can be used to update cell averages. Given Q_m^n, the average density in grid cell m at time t_n, an approximation is wanted of Q_m^{n+1}, the average density in the same grid cell at time t_{n+1}. To determine the average density in the next time step (5.3) uses $F^n_{m-\frac{1}{2}}$ and $F^n_{m+\frac{1}{2}}$. These are the ingoing flux and the outgoing flux in the grid cells.
In (5.3) $F^n_{m-\frac{1}{2}}$ is an approximation of the average flux along $x = x_{m-\frac{1}{2}}$,

$$F^n_{m-\frac{1}{2}} \approx \frac{1}{\Delta t}\int_{t_n}^{t_{n+1}} f(q(x_{m-\frac{1}{2}}, t))dt. \tag{5.4}$$

If the determination of the average flux is only based on the values of Q^n, a finite volume method can be used. In Figure 5.2 a schematic view is given on the working of finite volume methods.

On time t_n the cell averages Q_m^n are known and for one cell the $F_{m-\frac{1}{2}}$ is known. For this cell the outgoing flux, $F^n_{m+\frac{1}{2}}$, can be determined with the approximation based on the value of Q_m^n. With (5.3) the cell average in the cell can be updated and for this cell Q_m^{n+1} is determined. The outgoing flux from this cell is the ingoing flux for the next cell. So for the next cell the cell averages Q_m^n is known, the ingoing flux $F_{m-\frac{1}{2}}$ is known and the outgoing flux again can be determined. This process repeats for all cells until for all cells the average density is determined.
In the description here the updating from the flux and the average density is done in the direction downstream, from $x = 0$ to $x = 1$. Updating can also be done in the opposite direction, upstream. The working is the same only the direction is different.

In this section a finite volume method is explained. The method divides the domain in several grid cells and the average density in each cell has to be updated every time-step. For updating the average density fluxes are needed. Determining a flux function which describes the ingoing flux and the outgoing flux is difficult. This has to be a function which determines a flux only based on the density in the grid cell. In the next section more is explained about flux functions.

5.2 Determination of clearing functions

In the previous section a finite volume method is described. In this method the domain is divided into grid cells. In each grid cell the average density is determined with the help of the ingoing flux and the outgoing flux. The ingoing flux and the outgoing flux have to be described by a function which is only based on the density in the grid cell. For these flux functions a clearing function is used.

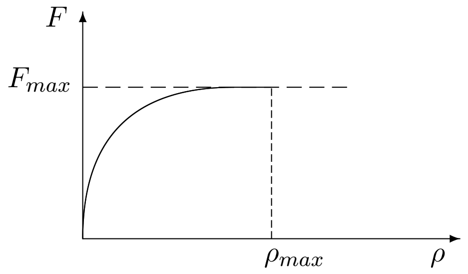

Figure 5.3: Fundamental diagram

In Figure 5.3 the shape of a clearing function is shown. On the moment that $\rho = 0$ there are no products in the grid cell and therefore no products can be send to the next grid cell, the flux is also zero. When the density increases the flux also increases. The density can increase until it reaches $\rho = \rho_{max}$. For $\rho = \rho_{max}$ the maximal number of products in a cell is reached and then the maximal value for the flux is also reached.

The clearing function can be determined with discrete event simulations, but the approximation method based on queuing theory used in Chapter 4 is faster. So to determine the clearing function the results of the approximation method are used. For a tandem queue of 100 workstations the results of the approximation method did not correspond to the results of the discrete event simulation. For this reason only the results of a tandem queue of 10 workstations are used.
The approximation method determines in steady state the mean number of products in each workstation in the tandem queue, the throughput and the cycle time. For the clearing function the flux and the density must be known. The mean number of products in each workstation can be seen as the density in a grid cell and the throughput can be seen as the outgoing flux of a grid cell.
In Figure 5.4 throughput is plotted against the mean number of products. This is done for tandem queues of 10 workstations and a maximal buffersize of 0, 1 or 5. For all tandem queues the squared coefficient of variation for the machines and the generator is 1 and the mean process time for the machines is also 1 . The inter arrival time of the generator varies to obtain different combinations of density and throughput.

In Figure 5.4 the clearing functions are shown of three tandem queues with different buffersizes. For each tandem queue the clearing function of workstation 1, workstation 5 and workstation 10 are shown.

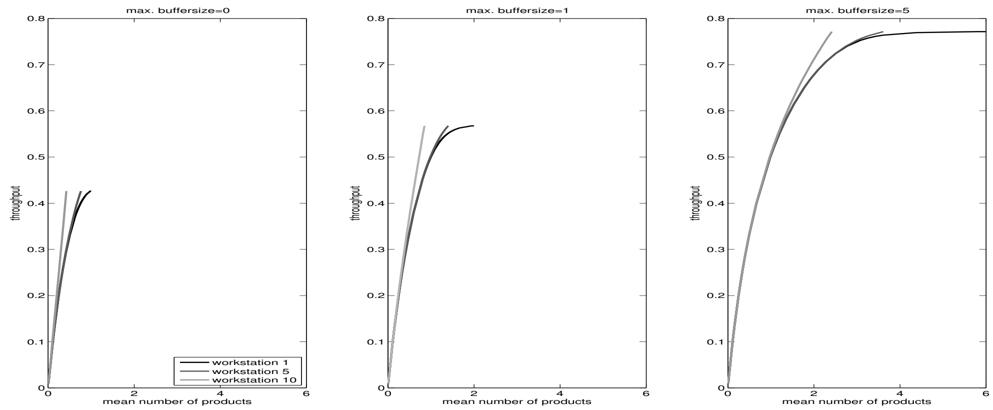

Figure 5.4: Clearing function in a tandem queue of 10 workstations with $c_a^2 = 1$, $t_e = 1$, $c_e^2 = 1$, $b = 0$ or $b = 1$ or $b = 5$.

In the graph of the clearing functions for a tandem queue with maximal buffersize of 0 the clearing function of workstation 1 have a similar form as the graph in Figure 5.3. For $\rho = 0$ the throughput is zero and for an increasing mean number of products the throughput increases. The density increases until it reaches a maximal density and at that point the maximal flux is also reached.
The clearing functions of workstation 5 and workstation 10 in the graph for a tandem queue with maximal buffersize 0 look like a straight line and do not reach the maximal density. Workstation 1 in the line tandem queue reaches the maximal density and by that the maximal throughput. With this maximal throughput the other workstations in the tandem queue have not yet reached the maximal density. For this reason the clearing functions of workstation 5 and workstation 10 do not show the same form as the clearing function in Figure 5.3. The same reasoning hold for the graphs of the tandem queue with maximal buffersize 1 and 5.
In Figure 5.4 can be seen that the clearing functions for workstations in a tandem queue are different for every workstation. So in a finite volume method different clearing functions are needed for workstations depending on the position within the tandem queue.

Chapter 6

Conclusions and recommendations

In this chapter conclusions of the previous research are summarized and recommendations for further research are given.

6.1 Conclusions

For modeling and control of manufacturing systems roughly three groups of models are used. These models are fluid models, queuing models and discrete event models. A fourth group of models might be partial differential equation models (PDE).
To determine if a PDE model might be considered for modeling a manufacturing system the behavior of a manufacturing system must be known. This is the goal of this research: in order to know the behavior a PDE must describe, derive properties for the behavior of a tandem queue with finite buffersizes.

The behavior of the tandem queue is divided into steady state behavior and dynamical behavior. Steady state is determined with an approximation method based on queuing theory and dynamical behavior is determined with discrete event simulation.

For the dynamical behavior a discrete event model is derived in Chi and with this model a simulation experiment is done. Ramp up of the system, blocking of a machine and ramp down of the system are simulated in an simulation experiment. The simulation experiment is done for a tandem queue of 10 workstations and a tandem queue of 100 workstations and for both queues the experiment is done for different parameter settings. The effects of the ramp up, blocking and starting the process are present in the tandem queue of 10 workstations and in the tandem queue of 100 workstation for the different parameter settings.

At the start of the simulation experiment the tandem queue is empty and during ramp up the mean number of products in all workstations increases from zero to a steady state. The mean number of products in workstation 1 immediately increases at the start of the experiment. The increase of the mean number of products in the other workstations start later because products have to go through the tandem queue first before reaching the workstations. A PDE should describe the increasing number of products and the order in which the workstations receive their first products.
When the system is in steady state the last machine breaks down and the products received by the workstation are placed in the buffer until this buffer is full. On that moment the machine in front of the full buffer is also blocked and the same process starts over again. Result is that blocking moves through the tandem queue from the last workstation to the first workstation with an almost constant velocity. A PDE should describe the order in which the workstations are blocked and the velocity of blocking.
When the complete tandem queue is blocked all workstations are full with products and a ramp down experiment is performed. The machine in the last workstation starts processing again and the number of products in each workstations decreases. All products have to leave the system and this can result in the fact that for a while the conditions for some workstations are different from the conditions in steady state before blocking. For these new conditions a new 'steady state' can occur. After this new 'steady state' the old conditions are reached again and the number of products decreases to the real steady state. The behavior in this ramp down experiment should be described by a PDE.
The ramp down experiment in the simulation takes more time than the ramp up experiment, this is also one of the aspects that a PDE has to describe.

The steady state of a system can be determined with discrete event simulation but an approximation method based on queuing theory is faster. To verify the results of the approximation method the results of the approximation method are compared to the results of discrete event simulation. For a tandem queue of 10 workstations the results correspond and the approximation method can be used. For the tandem queue of 100 workstations the results do not correspond.

A first step in determining a PDE is to model the previous behavior with the help of a finite volume method. A finite volume method divides a domain into grid cells and determines the average density in each grid cell every time unit. For the determination of the average density in each grid cell, the ingoing flux and the outgoing flux are needed. The determination of these fluxes are difficult and clearing functions are used for this purpose. These clearing functions determine the outgoing flux based only on the average density in a cell. The outgoing flux can be seen as the throughput in a system and the average density can be seen as the mean number of products.
The results of the approximation method based on queuing theory are used to determine the clearing functions. This is only done for the tandem queues of 10 workstations because for these tandem queues the results of the approximation method correspond to the results of the discrete event simulation. The throughput of a workstation is plotted against the mean number of products. From these results can be concluded

that the clearing function for workstation 1 is not the same as the clearing function for workstation 10. The clearing function depends on the position within the system.

6.2 Recommendations

In this report research is done on partial differential equation models for manufacturing systems with finite buffersizes. Steady state behavior of a tandem queue and dynamical behavior is described. This description of the behavior can be used in the search for a PDE and for the verification after a PDE is derived. Further research can be done on finding a PDE which describes the behavior found in this report and the results of such a PDE model can be compared to the results in this report.
In Chapter 3 a discrete event model is described and simulation experiments are done with this model. In the model for a tandem queue with a buffersize of 0 the last workstation in the tandem queue can, in some cases, receive a product after the last machine is blocked. When the last machine is blocked while there is no product in the machine, the workstation can receive one product. Receiving a product in this case should not be possible. The effect on the results is small but in further research this can be improved.
Another point for further research is the queuing theoretical approximation used in Chapter 4 [Vuu05]. For a large tandem queue the results of this approximation method do not correspond to the results of the discrete event simulations. The results of this approximation method can not be used in the determination of clearing functions or any other further research. Another approximation method based on queuing theory might give results not only for short tandem queues but also for a longer tandem queue. Recent results by Paul Frenken might be used as a starting point.
In Chapter 5 clearing functions for a tandem queue of 10 workstations are determined with the results of the approximation method of Chapter 4. From the results can be concluded that the clearing function depends on the position within the system. The clearing function for workstation 1 is not the same as the clearing function for workstation 10. In further research more have to be determined about clearing functions depending on the position in the system and how these clearing functions can be used to describe the behavior of a tandem queue with finite buffersizes.

Bibliography

[Ber00] P. Berg, A. Mason, and A. Woods. *Phys. Rev. E.*, 61:1056, 2000.

[Hel96] D. Helbing. Gas-kinetic derivation of Navier-Stokes-like traffic equations. *Physical review*, 53(3), March 1996.

[Hel01] D. Helbing. Traffic and related self-driven many-particle systems. *Reviews of modern physics*, 73, October 2001.

[Her98] M. Herrmann and B.S. Kerner. Local cluster effect in different traffic flow models. *Physica A-statistical mechanics and its applications*, June 1998.

[Hof02] A.T. Hofkamp and J.E. Rooda. χ reference manual, 2002. Technische Universiteit Einhoven, Department of mechanical engineering.

[Ker93] B.S. Kerner and P. Konhäuser. *Phys. Rev. E.*, 48:R2335, 1993.

[Ker94] B.S. Kerner and P. Konhauser. *Phys. Rev. E.*, 50(54), 1994.

[Lef05] A.A.J. Lefeber, R.A. van den Berg, and J.E. Rooda. Modelling manufacturing systems for control: A validation study. In *Networks of interacting machines: production organization in complex industrial systems and biological cells*. 2005.

[Lev03] R.J. Leveque. *Finite volume methods for hyperbolic problems*. Press syndicate of the university of Cambridge, 2003.

[Lig55] M.J. Lighthill and G.B. Whitham. On kinematic waves. II: a theory of traffic flow on long crowded roads. *Proc. R. Soc. London, Ser A.*, 229:317–345, 1955.

[Nag02] T. Nagatani. The physics of traffic jams. *Reports on progress in physics*, 65:1331–1386, August 2002.

[Nag03] K. Nagel, P. Wagner, and R. Woesler. Still flowing: Approaches to traffic flow and traffic jam modeling. *Operations research*, 51(5):681–710, January 2003.

[Pay71] H.J. Payne. *Mathematical Models of Public Systems*, pages 51–61, 1971. Edited by G.A. Bekey(Simulation Council, La Jolla, CA).

[Pay79] H.J. Payne. *Research directions in computer control of urban traffic systems*, page 251, 1979. Edited by W.S. Levine, E. Lieberman and J.J. Fearnsides(American Society of civil Engineers, New York).

[Phi79a] W.F. Phillips. *Transp. plan. Technol.*, 5, 1979.

[Phi79b] W.F. Phillips. *Proceedings of the 1978 IEEE conference on decision and control.*, page 1032, 1979.

[Ver04] J. Vervoort and J.E. Rooda. Learning χ, 2004. Technische Universiteit Einhoven, Department of mechanical engineering.

[Vuu05] M. Vuuren and I.J.B.F. Adan. Performance analysis of tandem queues with small buffers. *Proceedings of the Fifth International Conference on "Analysis of Manufacturing Systems-Production Management"*, 2005.

Appendix A

List of symbols

b	Maximal number of products in a buffer
c_a^2	Squared coefficient of variation on the inter-arrival time
c_e^2	Squared coefficient of variation on the mean processing time
$F_{m-1/2}^n$	Ingoing flux in grid cell m on time n
$F_{m+1/2}^n$	Outgoing flux in grid cell m on time n
n	Number of machines
t	Time
t_a	Mean inter-arrival time
t_e	Mean processing time
Q_m^n	Average density in grid cell m on time n
x_m	Grid cell m
Δx	Length of a grid cell

Appendix B

Discrete event model

In Chapter 3 dynamical behavior of a tandem queue is determined. To determine this behavior a discrete event model is derived in Chi. A short description can be found in Section 3.1, in this appendix the complete model is given with a short explanation on how to use the model.

B.1 χ model

In this section a χ model is given for a line of 10 workstations with a maximal buffersize of $b > 0$. For $b = 0$ the χ model is different for the proces of the buffer. Both processes can be found in the model.

```
//-----------------------------------------------------------------------------
// Imports from chi libraries
//-----------------------------------------------------------------------------
from random import*
from std import*
from fileio import*

//-----------------------------------------------------------------------------
// Define number of machines
//-----------------------------------------------------------------------------
const m:nat= 10                  // number of machines
    , m1:nat= 11                 // number of machines plus 1
    , mmin1:nat=9                // number of machines minus 1
```

```
//------------------------------------------------------------------------------------
// Processes
//------------------------------------------------------------------------------------
proc G(a:!void, p1,p2:real)=
|[ t:->real
 | t:=gamma(p1,p2)
 ;*[ true; a! -> delta sample(t)]
]|

proc B(a:?void,b:!void,c:?void,d:?void,e:!nat,n:nat)=
|[ xs, atM:nat
 | xs:=0; atM:=0
 ;*[ xs<n;a? -> xs:=xs+1
   | xs>0;b! -> xs:=xs-1;atM:=atM+1
   | true;c? -> atM:=atM-1
   | true;d? -> e!xs+atM
   ]
]|

//------------------------------------------------------------------------------------
// Buffer when max. number of products in the buffer is zero
//------------------------------------------------------------------------------------
//proc B(a:?void,b:!void,c:?void,d:?void,e:!nat)=
//|[ xs, atM:nat
// | xs:=0; atM:=0
// ;*[ xs+atM=0;a? -> xs:=xs+1
//   | xs+atM=1;b! -> xs:=xs-1;atM:=atM+1
//   | true;c? -> atM:=atM-1
//   | true;d? -> e!xs+atM
//   ]
//]|

proc M(a:?void,b:!void,c:!void, p1,p2:real)=
|[ u:->real
 | u:=gamma(p1,p2)
 ;*[true -> a?; delta sample u; b!; c!]
]|

proc Mf(a:?void,b:!void,c:!void, d:?bool, p1,p2:real)=
|[ u:->real, idle,up:bool, tfinish,tremain:real
 | u:=gamma(p1,p2)
 ; idle:=true; up:=true
 ;*[ up and idle; a? -> tfinish:=time+sample u; idle:=false
   | up and not idle; delta tfinish-time -> b!;c!;idle:=true
   | true; d?up ->[ not up -> tremain:=tfinish-time
                  | up     -> tfinish:=time+tremain
                  ]
   ]
]|
```

```
proc E(a:?void)=
|[ *[true -> a? ]
]|

proc C(a:(!void)^m,b:(?nat)^m,c:!bool,todolist:(nat#bool)*)=
|[ wip:nat^m,wiplist:(nat^m)*,n,i:nat
 | n:=0; wiplist:=[]
 ;*[ len(todolist)>0
    -> *[ n<hd(todolist).0
         -> i:=0; *[i<m ->a.i!;b.i?wip.i;i:=i+1]
          ; wiplist:=wiplist++[wip]
          ; n:=n+1
          ; delta 1.0
        ]
     ; c!hd(todolist).1
     ; todolist:=tl(todolist)
   ]
 ; !"wiplist=", wiplist,"\n"
]|

//-------------------------------------------------------------------------------------
// Clusters
//-------------------------------------------------------------------------------------
clus WS(gm2b:?void, m2b:!void, c2b:?void, b2c:!nat, n:nat, p1, p2:real)=
|[ b2m:-void, mfin:-void
 | B(gm2b, b2m, mfin, c2b, b2c, n)
|| M(b2m, m2b, mfin, p1, p2)
]|

clus WSf(gm2b:?void, m2e:!void, c2b:?void, b2c:!nat, mdown:?bool, n:nat, p1, p2:real)=
|[ b2m:-void, mfin:-void
 | B(gm2b, b2m, mfin, c2b, b2c, n)
|| Mf(b2m, m2e, mfin, mdown, p1, p2)
]|

clus PRODLINE(pg1, pg2, p1, p2:real,maxwip:nat,todolist:(nat#bool)*)=
|[ ws:(-void)^m1, askwip:(-void)^m, sendwip:(-nat)^m,breakdown:-bool
 | G(ws.0, pg1, pg2)
|| i: nat <-0..mmin1: WS(ws.i, ws.(i+1), askwip.i, sendwip.i, maxwip, p1, p2)
|| WSf(ws.mmin1,ws.m,askwip.mmin1,sendwip.mmin1, breakdown ,maxwip, p1, p2)
|| E(ws.m)
|| C(askwip,sendwip,breakdown,todolist)
]|

xper(pg1, pg2, p1, p2:real,maxwip:nat,todolist:(nat#bool)*)=
 |[ PRODLINE(pg1, pg2, p1, p2, maxwip, todolist)]|
```

B.2 Explanation on χ model

To perform the simulation several parameters have to defined. The number of machines have to be set in the model and other parameters have to be defined at the prompt. After defining the number of machines the model has to be compiled with:

```
>> chic filename.chi
```

Then the model can be started with the following Python code:

```
#!/usr/bin/env python

import os
for m in range(0, 100):

    os.system("./filename pg1 pg2 p1 p2 b '[ <block 0> <start 1> <end 0> ]' >> output1.txt")
    os.system("./filename pg1 pg2 p1 p2 b '[ <block 0> <start 1> <end 0> ]' >> output2.txt")
    os.system("./filename pg1 pg2 p1 p2 b '[ <block 0> <start 1> <end 0> ]' >> output3.txt")
    os.system("./filename pg1 pg2 p1 p2 b '[ <block 0> <start 1> <end 0> ]' >> output4.txt")
    os.system("./filename pg1 pg2 p1 p2 b '[ <block 0> <start 1> <end 0> ]' >> output5.txt")
```

The parameters can be found in Table B.1.

Before running the Python script has to be made executable with:

```
>> chmod u+x filename.py
```

Then the file can be started with:

```
>> ./filename.py
```

Running this Python script gives 5 files and each file contains the results of 100 simulations.

parameters	type	description
pg1,pg2	real	parameters needed in gamma distribution of the inter-arrival time mean inter-arrival time= $pg1 * pg2$, variance= $pg1 * pg2^2$
p1,p2	real	parameters needed in gamma distribution of the process time mean process time= $p1 * p2$, variance= $p1 * p2^2$
b	nat	maximal buffersize
block	real	moment of blocking
start	real	moment of starting the process again
end	real	endtime of simulation

Table B.1: Simulation parameters

Appendix C

Programming codes to determine simulation results

In Appendix B the discrete event model is given and an explanation on how to use the model for simulations. The results of these simulations have to be processed to determine a mean number of products and to determine the velocity of blocking. The mean number of products and the velocity after processing the simulations results are shown in Chapter 3 and in this appendix a description is given on how these mean results and the velocity are determined.

C.1 Determination of mean number of products

Simulating the χ model of Appendix B results in files with lists of tuples. Every list represents one simulation and contains one tuple for every time-step. These tuples represent the number of products in each workstation at a certain time.
With the following Python code the mean number of products in every workstation can be determined from several lists.

Before using the Python code, the Python code first has to be made executable with:

```
>> chmod u+x filename.py
```

Then the file can be started with:

```
>> ./filename.py
```

The Python file:

```
#!/usr/bin/env python
import time, sys, os, string, math

from mlabwrap import mlab
from Numeric import *
from math import log
from math import pow

#--------------------------------------------------------------------------------------
# Define path
#--------------------------------------------------------------------------------------

path="/.../.../"

dir=os.listdir(path)

#--------------------------------------------------------------------------------------
# Matrices used to store mean nr of prod and the std
# Results determined for a line of 10 workstations and
# the simulation takes 5000 time units. Values have to be changed
# when other simulation parameters are used.
#--------------------------------------------------------------------------------------

totmean = []
totstd=[]
totrun =0

updatestd=[[0]*10]*5001
updatemean=[[0]*10]*5001

#--------------------------------------------------------------------------------------
# Determine the mean number of products in a workstation and the std.
#--------------------------------------------------------------------------------------

for file in dir:

#----- open file with results
    if (file[0:8]=="filename"):
       inputfile=open(file,'r')

#----- read a list with results
       for line in inputfile:
          if (line[0:8]=="wiplist="):
                 matrix = (line.replace(' ',',').replace('[,<|,','[[')
                               .replace(',|>,]',']]').replace(',|>,<|,',',],[')
                               .replace('wiplist=',''))
                 exp = eval(matrix)
```

```
#----- the first list
                if totrun<=0:
                      totmean=exp
                      totstd=updatestd
                      totrun=1

#----- not the first list and the mean and std have to be updated every time
                elif totrun>0:
                      n=totrun+1
                      nrsample=len(exp)
                      nrmach=len(exp[0])
                      k=0
                      j=0

                      while k<nrsample:
                             while j<nrmach:
                                   updatestd[k][j]=((n-2.)/(n-1)*totstd[k][j]+
                                                    1./n*(exp[k][j]-totmean[k][j])**2)
                                   updatemean[k][j]= ((n-1.)/n* totmean[k][j]+
                                                      (1./n)* exp[k][j])
                                   j=j+1
                             k=k+1
                             j=0

                      totrun=n
                      totstd=updatestd
                      totmean=updatemean

#--------------------------------------------------------------------------------------
# saving the mean number of products as a matrix in Matlab
#--------------------------------------------------------------------------------------

saveresults = mlab.results(totmean, totstd, totrun)
```

In this python file the results are saved in Matlab. The following Matlab file is needed.

```
function [saveresults]=results(totmean,totstd,totrun)
saveresults=1
save mean.mat totmean totstd totrun
```

Combine result of two simulation runs With the previous Python file the results of several simulations can be combined to determine a mean number of products. This mean number of products is saved in a mat-file in Matlab. To combine the results of two simulations runs two mat-files have to be combined. The next code can be used in Matlab for this purpose.

```
clc
clear all
close all
%-------------------------------------------------------------------------------------
% open files with mean results
%-------------------------------------------------------------------------------------

load file1.mat
totmean1=totmean;
totstd1=totstd;
n1=totrun;

load file2.mat
totmean2=totmean;
totstd2=totstd;
n2=totrun;

%-------------------------------------------------------------------------------------
% combine files to determine the mean
%-------------------------------------------------------------------------------------

[nrsample, nrmach]=size(totmean1);

k=1;
j=1;
while k<=nrsample;
  while j<=nrmach
        updatestd(k,j)= 1/(n1+n2-1)*((n1-1)*totstdn1(k,j)+(n2-1)*totstdn2(k,j)+...
                        (n1*n2)/(n1+n2)*(totmean1(k,j)-totmeann2(k,j))^2);
        updatemean(k,j)= n1/(n1+n2)*totmean1(k,j)+n2/(n1+n2)*totmeann2(k,j);
        j=j+1;
  end
        j=1;
        k=k+1;
end
totrun=n1+n2;
totstd=updatestd;
totmean=updatemean;

%-------------------------------------------------------------------------------------
% save the results in a mat-file
%-------------------------------------------------------------------------------------

save file3.mat totrun totstd totmean
```

C.2 Determination of velocity of blocking

In Chapter 3 the velocity of blocking moving through the line of workstations is determined. This is done with the results of discrete event simulation.
From the results of a simulation experiment the number of products in a workstation is known every time-unit. From this information can be determined for every workstation when the maximal number of products is reached and the result is a list of moments of time every workstations reaches the maximal number of products. These lists are put in a matrix and with Matlab the mean of these moments of time can be determined.

```
#!/usr/bin/env python

import time, sys, os, string, math
from mlabwrap import mlab
from Numeric import *
from math import log
from math import pow

#-------------------------------------------------------------------------------------
# define path, initialize and open file to write
#-------------------------------------------------------------------------------------

path="/.../.../"
dir=os.listdir(path)

outputfile=open('file1.txt', 'w')

newtotvalues=[]
outputfile.write('[')

#------------------------------------------------------------------------------------
# Determine the moments the maximal  number of
# products is reached in a workstation
#------------------------------------------------------------------------------------

for file in dir:

#----- open file with results
    if (file[0:5]=="file2"):
       inputfile=open(file,'r')

#-----read results of 1 simulation
       for line in inputfile:
           if (line[0:8]=="wiplist="):
                 matrix = (line.replace(' ',',').replace('[,<|,','[[')
                              .replace(',|>,]',']]').replace(',|>,<|,','],[')
                              .replace('wiplist=',''))
                 values = eval(matrix)
```

```
#-----the values in this file have to be adapted to the simulation results

                 m=9                   #-----number of machines minus 1.
                 t=3000                #-----moment the last machine is blocked.
                 max=values[3999][m]   #-----time just before last machine
                                       # starts processing.

                 pointoftime=[0]*(m+1)

#------------------------------------------------------------------------------------
# searching the max. number of products for each workstation.
# Starting from the last machine in the line from time=t.
#------------------------------------------------------------------------------------
                 i=m
                 while i>=0:
                       while values[t][i]<max:
                             t=t+1;

                       max=values[3999][m]
                       pointoftime[i]=t
                       i=i-1;
           outputfile.write(str(pointoftime))
           outputfile.write(';')


outputfile.write('];'  )

outputfile.close
```

Appendix D

Approximation method based on queuing theory

In this Appendix an explanation is given on how to use the approximation method based on queueing theory from Chapter 4.

The steady state of a line of workstations is determined with an approximation method based on queuing theory [Vuu05].
The approximation method models a line of workstations with finite buffersizes. The first workstation in the line is never starved and can be seen as the generator of the line.

```
>> [t,L,S] = SolveTandemQueue([te],[ce^ 2],[b])
```

The input information of the approximation method exist of three vectors, `[te], [ce^ 2] and [b]`. The first vector `[te]` contains the mean process time of the machine, the second vector `[ce^ 2]` contains the squared coefficient of variation of the machines and the third vector `[b]` contains the maximal buffersize. The first workstation in the line does not have a buffer, so the first two workstations contain one value more than the third vector.

The output of the approximation method is a vector `L` which contains the mean number of products in each workstations and a workstation is the buffer plus the machine behind the buffer. Other output values are `t`, the throughput of the line of workstations and `S` is the flow time of the line of workstations.