Modelling of manufacturing lines using higher order PDEs

S. Platschorre

SE 420413

Masters thesis

Supervisor: Prof. dr. ir. J.E. Rooda
Coach: Dr. ir. A.A.J. Lefeber
Ir. J.A.W.M. van Eekelen

Technische Universiteit Eindhoven
Department of Mechanical Engineering
Systems Engineering Group

Eindhoven, December 2004

Assignment

Preface

Writing this thesis is the final step in becoming a Master of Science in Mechanical Engineering. A process that took me almost seven and a half years. Graduating from the Technische Universiteit Eindhoven meant to me finding a balance between studying and being a student. Specially the last year, when I tried to combine graduating with my work, shacking up, organizing The Outdoor Challenge and winning some adventure races. This proved to be a hard task, but after fourteen months I succeeded in all.

My final year of studying was a year of researching higher order PDE models for manufacturing lines, part of a research in controlling manufacturing lines. I hope someday the captains of industry can make more profit using the final results. I also hope to be one of them by then. I could not have completed this research without the support of my coaches, Erjen Lefeber and Joost van Eekelen, thanks for your help, specially to Erjen who has cracked some of the difficult mathematical problems on a single A4.

Since sports and outdoor activities had an important role throughout my studies period I would like to thank my friends, Wolt, Herman, Monique, Jorrit and Dolf for all the great memories, my teammates Wolt, Raymond and Henri for our defeats and victories and finally Monique, Andor, Rosita, Jorrit, Annette and Dimitry for making The Outdoor Challenge the greatest race in the Netherlands. When naming my friends, I must not forget Erik, since he has some incriminating pictures of me.

I would like to thank my parents who gave me the opportunity to study, to discover the fun part of being a student and to become an adventure racer. Although they wanted me to study a bit more instead of exploring the outdoors they kept supporting me and therefore I owe them a lot, thanks!

And of course, special thanks to my girlfriend Rosita, for the support and all the fun, beyond and ahead.

To me finishing this project means entering the world of the working people, this will cost me a lot of taxes and BaCo, but I am excited to start this new challenge.

Sven Platschorre
Eindhoven, December 2004

Summary

Control of manufacturing systems is becoming more and more complex. Manufacturing systems become more complex itself and the requirements in the demand increase. In today's world meeting the customer demand in time is crucial for making a profitable business, so for companies it is important to be able to meet that demand. In companies with a complex production process with long flow times, high volumes, rework and reentrant flows it is hard to meet the customer demand. An example of such an industry is the semi-conductor industry. Companies need to know what has to be the input at this moment, given the current state, to meet a certain demand in time. This can be done by applying control theory. Therefore, a physical relevant model of the total supply chain of a company is needed. Currently three groups of models are available for manufacturing lines: discrete event models, queuing theory and fluid models. These models are suitable to model manufacturing systems, but from a control perspective they all have their own flaw. Discrete event models suffer from state explosion, as the problems grow. Queueing theory mainly deals with steady state analysis. Fluid models do not include any dynamics and are not concerned with flow time. Recently, partial differential equation (PDE) models have been proposed to overcome these flaws.

In this research the focus is on higher order PDE models. A model is proposed that consists of a mass conservation law and a momentum conservation law. In this research the momentum conservation law is constructed after examples from traffic flow models. Four terms from traffic flows models are introduced: relaxation, anticipation, memory and viscosity. Relaxation is a term that takes into account that the system needs time to go to a steady state speed and anticipation, memory and viscosity are look ahead terms that take into account what happens in front of products. The terms that come from traffic theory needed to get a physical meaning in manufacturing systems, this research partly succeeded in this.

All terms have been implemented in the model and one by one they have been validated. To perform the validation study a test case has been developed that is the ramp up of a simple manufacturing system. This simple manufacturing line has a transient state. From this test case a discrete event model has been built, that is assumed to be a good representation of the actual physical behavior of the line. To average discrete event simulation data as a function of time, three methods have been investigated. It has been concluded that 'vertical' averaging is the best method and that it is sufficient for

the purposes in this research.

To validate the model, PDE simulation results are compared to discrete event simulation results. To simulate the PDE a numerical solution method is used. A C++ code has been used to perform the numerical simulations. It appears in some experiments that the results of the PDE model are physically incorrect. The error influenced some of the conclusions about the model. Therefore, it is recommended to study the error and the solution method before further research using the code should be done.

This research shows that the proposed PDE model brings improvements compared to the previous proposed models based on PDEs. The term from the proposed model that is responsible for the improvements is the relaxation. This conclusion is not influenced by the error that occurs in the numerical solution. The terms anticipation and viscosity do not bring any improvement to the performance of the model. When looking at the physical meaning of the terms relaxation, anticipation and viscosity, these conclusions make sense. The term memory seems in some cases, with high utilizations, to minimize the present error. This conclusion has been highly influenced by the error and it is highly questioned if this term brings an improvement to the performance of the model when the error is solved. The fact that on physical arguments no improvement is expected underlines this. For future research on the model some suggestions have been made; a relaxation term for a conwip line is presented and a new parameter called entropy is introduced.

Samenvatting

Het regelen van fabricage systemen wordt steeds ingewikkelder. Fabricage systemen worden zelf steeds complexer en de eisen van de klant stijgen. In de huidige economie is het voldoen aan de eisen van de klant cruciaal om winst te maken, dus het is zeer belangrijk om aan deze eisen te voldoen. In bedrijven met een complex fabricage proces met grote procestijden, hoge volumes en recirculerende productenstromen is het moeilijk om aan de eisen van de klant te voldoen. Voorbeelden van dit soort bedrijven zijn te vinden in de semi-conductor industrie. Om aan de vraag van de klant te voldoen moet een bedrijf weten wat ze nu moeten doen, gegeven de huidige toestand. Een mogelijkheid om dit te bereiken is het toepassen van regeltechniek. Hiervoor is een fysisch relevant model nodig van een fabricage systeem. Momenteel zijn er drie groepen modellen beschikbaar, maar vanuit de regeltechniek voldoen deze niet. 'Discrete event' modellen kosten teveel rekentijd als het probleem te groot wordt, wachtrijtheorie behandelt vooral evenwichtssituaties en vloeistofmodellen kennen geen procestijden. Recentelijk zijn partiële differentiaal vergelijkingen voorgesteld als alternatief.

Dit onderzoek richt zich op hogere orde differentiaal vergelijkingen. Het voorgestelde model bestaat uit een massabalans en een impulsbalans. De impulsbalans is gebaseerd op verkeersmodellen. Vier termen uit de verkeertheorie zijn gentroduceerd: relaxatie, anticipatie, geheugen en viscositeit. Relaxatie modelleert het feit dat een systeem tijd nodig heeft om de evenwichtssnelheid te bereiken. Anticipatie, geheugen en viscositeit zijn termen die vooruit kijken. Ze nemen mee wat er voor het product gebeurt. Deze termen moeten een betekenis krijgen in fabricage systemen, dit onderzoek is hier gedeeltelijk in geslaagd.

Alle termen zijn één voor één geïmplementeerd en gevalideerd. Hiervoor is een test case ontworpen. De test case is het opstarten van een simpele fabricage lijn. Deze lijn heeft een overgangsfase tussen twee evenwichtstoestanden. Van de test case is een discrete event simulatie gebouwd waarvan is aangenomen dat deze de werkelijkheid voldoende dicht benadert. Om de discrete event resultaten te verwerken zijn er drie verschillende middelingmethodes voorgesteld. Hieruit blijkt dat verticaal middelen de beste methode is die geschikt is voor dit onderzoek.

Om het voorgestelde model te valideren is er van het model een numerieke oplossing nodig. De numerieke resultaten worden vergeleken met de discrete event resultaten.

Om de numerieke resultaten te verkrijgen is een C++ code gebruikt. Uit de experimenten blijkt dat enkele resultaten geen fysieke betekenis hebben. Er zit een fout in de numerieke oplossing en dit beïnvloed enkele conclusies over het model. Het is daarom ook aanbevolen om de numerieke oplossing verder te onderzoeken.

Dit onderzoek heeft aangetoond dat het voorgestelde model een verbetering is ten opzichte van eerder voorgestelde partiële differentiaal vergelijkingen. Relaxatie is de term die hiervoor verantwoordelijk is. Deze conclusie is niet beïnvloed door de eerder genoemde fout. De termen anticipatie en viscositeit zorgen niet voor een verbetering van het model. Als naar de fysieke betekenis wordt gekeken van deze termen zijn deze conclusies logisch. De term geheugen lijkt in enkele gevallen (met een hoge bezettingsgraad) de aanwezige fout te minimaliseren, of geheugen ook in een foutloze oplossing verbetering brengt is onduidelijk. Waarschijnlijk niet, ook omdat dit vanwege de fysieke betekenis niet te verwachten is. Voor verder onderzoek aan het model zijn enkele suggesties gemaakt; een relaxatie term voor een conwip lijn en een nieuwe parameter, entropie, is geïntroduceerd.

Contents

Assignment **i**

Preface **iii**

Summary **v**

Samenvatting **vii**

1 Introduction **1**

2 Higher order PDEs for manufacturing systems **3**

2.1 An approach for finding a model based on higher order PDEs 4

2.2 Basics of traffic flow models based on higher order PDEs 4

2.2.1 Mass conservation law 5

2.2.2 Momentum conservation law 6

2.3 Traffic flow models 7

2.3.1 Aw & Rascle 2000 7

2.3.2 Klar & Wegener 2000 8

2.3.3 Li 2002 8

2.3.4 Zhang 2003 9

2.4 Higher order PDE model for manufacturing systems 10

3 Validation set-up **13**

3.1 Description of the test case 13

3.2 Variables of the test case 14

3.3 Measuring the performance 16

4 Discrete event simulation **19**

4.1 Discrete event simulation of the test case 19

4.2 Averaging discrete event simulation results 20

4.2.1 Averaging method Horizontal 1 20

4.2.2 Averaging method Horizontal 2 22

4.2.3 Averaging method Vertical 23

4.3 Results of the averaging method 25

4.3.1 Conclusion of the averaging method 27

4.4 Results of the discrete event simulation 28

4.5 Conclusion of the discrete event simulation 28

5 A PDE model for manufacturing systems **31**

5.1 The PDE model 31

5.1.1 Relaxation 32

5.1.2 Anticipation 33

5.1.3 Memory 33

5.1.4 Viscosity 34

5.2 Boundary conditions and initial values 34

5.3 Simulation results from the PDE model 34

5.4 Results 35

5.5 Conclusions 36

6 Validation study **37**

6.1 No terms 37

6.2 Relaxation 39

6.3 Anticipation 43

6.4 Memory and Viscosity 45

6.5 Conclusion 48

7 Conclusions **51**

8 Recommendations **53**

Bibliography **55**

A Extensions to the PDE model **59**

A.1 State of a manufacturing system 59

A.2 Conwip vs. push line 60

A.3 Entropy 63

A.3.1 Calculating entropy 63

A.4 Conclusion 65

B Discrete event simulation **67**

B.1 χ code of the test case 67

B.2 Averaging method script 68

B.3 Processing discrete event simulation results 71

C PDE simulation **75**

C.1 C++ code of the PDE model 75

Chapter 1

Introduction

The control of manufacturing systems is becoming more and more complex. Manufacturing systems become more complex itself and the requirements in the demand increase. To be able to meet the demands in time with a minimal effort, control is necessary. In order to use techniques known from control theory, a physical relevant model of complex manufacturing systems is needed. Currently three groups of models are available for manufacturing lines: discrete event models, queuing theory and fluid models. These models are suitable to model manufacturing systems, but from a control perspective they all have their own flaw. Discrete event models suffer from state explosion, as the problems grow. Queueing theory mainly deals with steady state analysis. Fluid models do not include any dynamics and are not concerned with flow time. Recently, partial differential equation (PDE) models have been proposed to overcome these flaws. Current research on PDE modelling of manufacturing systems mainly consists of first order PDEs, based on first order traffic flow models. Validation studies have shown that the currently available first order models do not describe the transient behavior correctly. In this research the focus is on higher order PDE models.

Objective

The goal of this study is to propose a model for manufacturing systems based on higher order PDEs that describes both steady state and transient behavior correctly. The proposed model will be based on traffic flow models. It has to be investigated to what extent the properties of traffic flow models hold for manufacturing systems. This has to be done by a validation study, where PDE simulation experiments are compared to discrete event simulations. The test case used for the experiments must incorporate the most important properties of manufacturing systems. The validation study has to make clear if the model is good enough, where what is "good enough" has to be defined. Finally a conclusion will be made about the proposed model and suggestions for improvements in the model are done.

Approach

Current first order models are based on traffic flow models and to find higher order models, traffic theory again provides a good start. The benefit of taking existing models as a basis is that the models have been tested before. When looking for models in traffic theory it has to be kept in mind that a lot of models have flaws as has been pointed out by Daganzo [Dag95]. Therefore, traffic theory literature is studied that refer to Daganzo's article. In that way some flaws in the model can be avoided. The search for models and the proposed new model is presented in Chapter 2. In Chapter 2 the proposed model is discussed briefly, later this is done in more detail in Chapter 5. The proposed model has to be validated by means of a test case. This case is discussed in Chapter 3. Chapter 3 deals also with the question when the model is considered to be a good model. For the test case two types of simulations are made to perform the validation study in Chapter 6. A discrete event simulation and a PDE simulation are discussed in Chapter 4 and 5 respectively. In the validation study the results of the discrete event simulation and the PDE simulation are compared and conclusions are drawn about the validity of the PDE model. Finally this report ends with conclusions and recommendations for further research.

Chapter 2

Higher order PDEs for manufacturing systems

In today's world meeting the customer demand in time is crucial for making a profitable business, so for companies it is important to be able to meet that demand. In companies with a complex production process with long flow times, high volumes, rework and reentrant flows it is hard to meet the customer demand. An example of such an industry is the semi-conductor industry. To meet the customer demand companies need to know what has to be the input at this moment, given the current state, to meet a certain demand in time. This can be done by applying control theory. Therefore, a physical relevant model of the total supply chain of a company is needed. Currently some models are already available, but they do not meet the criteria needed to solve this problem. The model should be able to:

- incorporate throughput, flow time and work in process,
- be controlled using control theory,
- predict the steady state and transient behavior correctly,
- predict reentrant behavior and rework correctly.

Examples of currently available models are queueing theory, discrete event models (DEM) and models based on ordinary differential equations (ODEs). All these models have some drawbacks which makes them undesired in the control of complex factories. Queueing theory only deals with steady state behavior, DEMs take too much calculation time and ODEs do not incorporate flow time. This means that a need for a new model exists. The use of partial differential equations (PDEs) could be a solution. First order PDEs have been studied by several authors, [Ber04, Lef04, Arm04, Arm02]. The study of Van den Berg [Ber04] shows that the proposed first order models are unable to predict the transient behavior of manufacturing systems correctly. In this research

the possibility is studied to use higher order PDEs for the modelling of manufacturing systems. The next section explains a possible route to such a model.

2.1 An approach for finding a model based on higher order PDEs

The previous section makes the need for a new model for manufacturing systems based on higher order PDEs reasonable. This section explains how a new model can be found. The search for the model starts by studying traffic theory. Manufacturing systems and traffic flows look alike, cars become products and the road can be seen as a line of workstations. In traffic theory a lot of models based on PDEs are available that are able to describe the properties of a traffic flow correctly both in the steady and transient state. Since in traffic theory good results have been obtained using higher order PDEs the search for a model for manufacturing systems based on PDEs starts by studying traffic theory. In traffic theory PDEs have been used since 1955 when the Lighthill, Whitham and Richards (LWR) model was presented [Lig55, Ric56]. Since then, a lot of new models have been proposed. The LWR model is a first order PDE. Van den Berg used an LWR-like model in his study [Ber04] and showed that it is unable to predict the transient behavior of manufacturing systems correctly. Therefore, the focus in this study is on higher order models. When the traffic theory is searched for traffic flow models, it has to be taken into account that in 1995 Daganzo seriously criticized all available traffic flow models based on higher order PDEs so far, [Dag95]. The models before 1995 have flaws in them, which means the models show behavior that was not intended to be modeled. In the models with the flaws, cars are influenced by cars from behind and under certain conditions cars travel backwards. Therefore, the search for higher orders models is focussed on literature from 1995 until now. The most widely used traffic flow models are studied in Section 2.3 and the focus is on the structure of these models. The structure of the different traffic flow models shows great resemblance and the most general structure is taken as the basis for a new model for manufacturing systems in Section 2.4. Before the traffic flows are presented in Section 2.3 the basics of traffic flow models are explained in the next section.

2.2 Basics of traffic flow models based on higher order PDEs

To understand the traffic flow models presented in the next section it is necessary to explain the notation and give a definition of higher order PDEs. The basic equations of higher order traffic flow models are also explained in this section. These basic equations are partial differential equations (PDEs). PDEs are differential equations, with multiple

derivatives to different variables. An example of a PDE is the mass conservation law:

$$\frac{\partial \rho}{\partial t} + \frac{\partial q}{\partial x} = 0. \tag{2.1}$$

The mass conservation law will be explained later in this section. The notation used in this report is $\frac{\partial \rho}{\partial t} = \rho_t$, so the mass conservation law becomes: $\rho_t + q_x = 0$. In this research a PDE is called a higher order PDE if the PDE has at least one second order derivative (e.g. v_{xx} or $v_{\rho x}$). Every higher order PDE can be written to a system of first order PDEs and visa versa, therefore every coupled system of PDEs will be considered a higher order PDE model. All PDE models considered in this report are based on two equations (both PDEs), the mass conservation law and the momentum conservation law and are therefore higher order PDEs. The mass conservation law and the momentum conservation law are explained in this section by means of traffic flows.

2.2.1 Mass conservation law

All traffic flow models are based on the mass conservation law. The mass conservation law describes that cars on a certain piece of road, Δx, are conserved. This means that no cars disappear and no new cars are created. However, in time, the number of cars on Δx can change by cars arriving at $x = 0$ (inflow) or cars leaving at $x = \Delta x$ (outflow):

$$\frac{\partial}{\partial t} \int_{x=0}^{x=\Delta x} \rho(x,t)dx = q_{(x=0)} - q_{(x=\Delta x)}. \tag{2.2}$$

In this equation $\int_{x=0}^{x=\Delta x} \rho(x,t)\mathrm{d}x$ is the number of cars on a certain piece of road Δx and $q_{(x=0)} - q_{(x=\Delta x)}$ is the difference between inflow $q_{(x=0)}$ and outflow $q_{(x=\Delta x)}$. Note that a piece of road is considered without on- or off-ramps. In differential form (2.2) becomes (where $q = \rho v$):

$$\rho_t + q_x = 0. \tag{2.3}$$

This equation is called the mass conservation law. In the mass conservation law ρ_t is the term that describes the change in density over time and q_x is the term which describes the difference between the in- and outflow.

The mass conservation law is the basis for all traffic flow models. To solve the mass conservation law a relation is needed between ρ and v and boundary conditions. The boundary conditions are dealt with in Chapter 5. For the relation between ρ and v different equations are available. In the LWR model this is a static relation between ρ and v. As mentioned earlier Van den Berg [Ber04] showed that the currently available models are not appropriate to model manufacturing systems. Instead of a static relation

between ρ and v it is also possible to model this relation by a PDE. This is done in Payne-type models by means of the momentum conservation law.

2.2.2 Momentum conservation law

In 1971 Payne was the first to use a momentum conservation equation in traffic flow models, [Pay71]. This equation states that the internal change in momentum plus the change of momentum due to physical in- and outflow is equal to the internal and external forces acting on a control volume:

$$\frac{d}{dt}\iiint_{\mathcal{V}} \rho \underline{v} d\mathcal{V} + \iint_{\mathcal{F}} \rho \underline{v}(\underline{n} \cdot \underline{v}) d\mathcal{F} = \Sigma \text{ internal and external forces.} \tag{2.4}$$

The first term of (2.4) is the internal change of momentum within a control volume $\mathcal{V}$, where $\underline{v}$ is the speed vector. The second term is change of momentum due to in- and outflow through the surface $\mathcal{F}$, where $\mathcal{F}$ is the border of $\mathcal{V}$ and $\underline{n}$ is the normal vector. Equation 2.4 can be simplified following the derivation beneath, [Bir60]. The first term of (2.4) can be rewritten, because $\mathcal{V}$ is a fixed volume and the second term can be rewritten according to the theorem of Gauss and this gives:

$$\iiint_{\mathcal{V}} \frac{d\rho \underline{v}}{dt} d\mathcal{V} + \iiint_{\mathcal{V}} \nabla \rho \underline{v}\underline{v} d\mathcal{V} = \Sigma \text{ internal and external forces.} \tag{2.5}$$

This equation has to hold for any control volume $\mathcal{V}$, so the integral can be left out which gives the following equation:

$$\frac{d\rho \underline{v}}{dt} + \nabla \rho \underline{v}\underline{v} = \Sigma \text{ internal and external forces.} \tag{2.6}$$

In traffic flow models and manufacturing systems we study a 1D situation, so (2.6) becomes:

$$\frac{d\rho v}{dt} + \frac{d\rho v v}{dx} = \Sigma \text{ internal and external forces.} \tag{2.7}$$

Using the chain rule for differential equations gives:

$$\rho v_t + v\rho_t + (\rho v)v_x + v(\rho v)_x = \Sigma \text{ internal and external forces.} \tag{2.8}$$

This combined with the mass conservation law (2.3) gives:

$$v_t + vv_x = \frac{\Sigma \text{ internal and external forces}}{\rho}. \tag{2.9}$$

The left hand side of this equation is the total time derivative of v ($\dot{v}$), which describes the rate of velocity change experienced by a moving observer who observes the traffic flow while moving along with the stream at the same velocity, [Hoo01]. The total time derivative is composed of v_t, which is the local acceleration noticed by a standing observer and vv_x, which is the convection acceleration, describing the change in the mean velocity due to in- and outflowing cars with a different speed. The right hand side of (2.9) will be composed of the factors that influences the change of the velocity in time and place. A lot of different possibilities exist to model the in- and external forces in (2.9) (the right hand side). Payne himself also proposed a model. Only this model has some serious flaws (like backward travel) as Daganzo pointed out in 1995, [Dag95]. From 1995 until now a lot of new models are proposed which do not suffer from these flaws. The most important are mentioned in Section 2.3. Note that these models form a system of PDEs with the mass conservation equation. A third category of traffic flow models (besides LWR and Payne-type) are the Helbing-type models. These models have a third PDE describing the dynamics of the velocity variance. The velocity variance is a possible term in the in- and external forces of (2.9). This model is not studied in this research.

2.3 Traffic flow models

In this section some frequently used traffic flow models are presented. The considered models all consist of the mass conservation law and a momentum conservation law, in this research these models are called Payne type models. From the presented models a proposal for a new model for manufacturing systems based on higher order PDEs is made in the next section. In the previous section the basic equations have been explained, where in the momentum conservation equation the in- and external forces still have to be completed. Many proposals have been made for the in- and external forces since Payne's model, but as noted before many suffer some flaws. A search in literature resulted in the models shown below. These models do not suffer the flaws mentioned by Daganzo [Dag95] and are widely used and accepted in traffic theory. Note that the mass conservation law is the same for all models and therefore not shown here.

2.3.1 Aw & Rascle 2000

Aw and Rascle [Aw00, Ras02] presented one of the first models without the flaws noticed by Daganzo. They presented the model from a mathematical point of view, they took Payne models and corrected it on mathematical arguments. They concluded that a correct model should involve the "convective derivative" of the pressure $p(t)_t + vp(t)_x$

where $p(t)$ is a "pressure" function that can be filled in different ways. The momentum conservation law presented by Aw and Rascle looks as follows:

$$v_t + vv_x = \frac{v^e(\rho) - v(t)}{\tau} - p(\rho)_t - vp(\rho)_x. \tag{2.10}$$

Klar and Wegener [Kla00] took this model as basis and gave the pressure term a interpretation from traffic flow theory. The model of Klar and Wegener is presented below.

2.3.2 Klar & Wegener 2000

The model of Klar and Wegener [Kla00] was presented in 2000. The model gives an interpretation to the pressure term from the model of Aw and Rascle. The momentum conservation law presented by Klar and Wegener looks as follows:

$$v_t + vv_x = \frac{v^e(\rho) - v(t)}{\tau} - \frac{P_x^e(\rho)}{\rho} - \frac{b^e(\rho)\rho_x}{\rho} - \frac{c^e(\rho)v_x}{\rho}. \tag{2.11}$$

Klar and Wegener's model appears to be similar to Zhang's model. Since Zhang's model is discussed in detail below, no details are given on Klar and Wegener's model here.

2.3.3 Li 2002

Li presented in 2002 a higher order traffic flow model [Li02] based on Zhang's model of 1998 [Zha98]. This traffic flow model allows road situation changes such as obstacles. The model is derived based on the assumption that the time needed for a following vehicle to assume a certain speed is determined by leading vehicles while taking the road changes into account. Li's model is based on a microscopic law from which a macroscopic model is derived. Since in this research the final macroscopic model is of interest only the macroscopic model is presented here:

$$v_t + vv_x = \frac{v^e(\rho) - v(t)}{\tau} - \rho v_\rho^{e2}\rho_x - \rho v_x^e v_\rho^e + v^e v_x^e. \tag{2.12}$$

For the derivation is referred to [Li02]. As can be seen, this model has a great resemblance with Zhang's model. Only parameters differ and no term with v_{xx} exits, but a term with $v_x v_\rho$ is present. Some terms are dependent on the equilibrium speed (v^e), in stead of the current speed v. Note that the equilibrium speed is dependent on $\rho(x,t)$, so these terms are not constants.

2.3.4 Zhang 2003

Zhang presented in 2003 a traffic flow model [Zha03] (which was an update of an earlier presented model, [Zha98]) based on a microscopic car-following relation. This relation is developed to a macroscopic fluid-like model of a traffic flow. The microscopic car-following relation used by Zhang is:

$$v_n(t+dt) = F((1-\epsilon)(x_{n-1}(t)-x_n(t))+\epsilon(x_{n-2}(t)-x_{n-1}(t)))+G(v_{n-1}(t)-v_n(t)). \quad (2.13)$$

It takes into account that drivers respond to the relative speed of the car in front of them, $v_{n-1}(t)$, through $G(v_{n-1}(t) - v_n(t))$, and to the distance to the two cars in front of them, $x_{n-1}(t)$ and $x_{n-2}(t)$, through $F((1-\epsilon)(x_{n-1}-x_n)+\epsilon(x_{n-2}-x_{n-1}))$. Where G and F are increasing functions and ϵ is a parameter that accounts for the 'look further ahead' factor. In (2.13) n stands for the n^{th} car and $(n-1)$ is the car in front of the n^{th} car. Because the future speed of a car ($v_n(t+dt)$) depends on its current speed ($v_n(t)$), driver memory exists. Equation 2.13 can be used to derive (2.14), see [Zha03]. This results in a viscous macroscopic model:

$$v_t + vv_x = \frac{v^e(\rho) - v(t)}{\tau} - \frac{c^2}{\rho}\rho_x - 2\beta c(\rho)v_x + \nu(\rho)v_{xx}. \quad (2.14)$$

Where
- τ, relaxation time;
- c, traffic sound parameter;
- β, parameter;
- ν, viscosity coefficient.

In (2.14) the left hand side is the same as in (2.9). The first term of the righthand side is the relaxation term. It describes the influence of the respond time (τ) of drivers to changes in velocity. In the second righthand term a parameter appears called the traffic sound speed c, which is the upstream propagation speed of a small disturbance, c depends on the density and the equilibrium speed. In the third righthand term β is a parameter that describes the look ahead factor in the traffic stream. The last term is the viscosity term, where ν is the viscosity coefficient. Viscosity in traffic flow models is harder to interpret than in fluid models. It can be seen as the tendency of drivers to resist sharp changes in speed. Zhang's model is a model that does not suffer the flaws mentioned by Daganzo and removes some of the deficiencies of the first order models (like the LWR model).

The four models presented are frequently used in traffic flow theory and are considered physically relevant. Based on these models a new model for manufacturing systems will be proposed in the next section. As the basis for the new manufacturing model a complete model from traffic flow theory is taken. A complete model is taken because a lot of research has been done in these models and no flaws have been discovered.

So it is assumed that these models are correct from a mathematical point of view. Therefore, the focus can be on their physical relevance in manufacturing systems. Since the considered models look similar a choice is made for Zhang's model to come to a new model for manufacturing systems based on higher order PDEs. Zhang's model is chosen because this is the most general model of the four presented options.

2.4 Higher order PDE model for manufacturing systems

In the previous section the most used traffic flow models have been presented. It can be noticed that they all have the same structure and a lot of elements in the momentum conservation law are the same. From the presented models a new model for manufacturing systems is proposed by taking Zhang's model as a basis. This model consists of the mass conservation law and the momentum conservation law:

$$\rho_t + q_x = 0, \tag{2.15}$$

$$v_t + vv_x = \frac{v^e - v}{\tau} - \frac{c^2}{\rho}\rho_x - 2\beta c v_x + \nu v_{xx}. \tag{2.16}$$

In the momentum conservation law some terms need further explanation. Note that for the moment the parameters for the manufacturing systems model are copied from traffic flow theory, further research has to prove if this is the right way.

- $v_t + vv_x$: describes the change of the velocity in time and place.
- $\frac{v^e - v}{\tau}$: τ is a non-zero positive constant, so $v^e - v$ must be zero in steady state. v is the actual speed and v^e is the equilibrium speed. In equilibrium they must be equal. For the equilibrium velocity the speed which follows from queueing theory can be taken (see also Chapter 5).

  For v^e different possibilities exist. In this research two possibilities are taken from queueing theory:

  $$v^e(t) = \frac{\mu}{m + w(t)} \tag{2.17}$$

  where $w = \int_0^1 \rho(x,t)dx$ or

  $$v^e(x,t) = \frac{\mu}{m + \rho(x,t)}. \tag{2.18}$$

  The value of the relaxation time τ has to be found, this can be done by fitting the PDE on the results of a discrete event simulation (minimizing the norm between the two results). τ influences the time the PDE model needs to go to the equilibrium, important will be how τ influences the PDE.

- $-\frac{c^2}{\rho}\rho_x$: c is the product sound speed, $c = \rho v_\rho^e$. The product sound speed c is a negative parameter (v_ρ^e is negative). If ρ_x is positive (the WIP level in front of a product rises), then the speed drops and if ρ_x is negative (the WIP level in front of a products drops) then the speed rises. This is easy to understand, also if you realize that speed is related to the flow time of a product. The flow time rises when the WIP rises, and then the speed drops and the other way around.

- $-2\beta c v_x$: β is the second parameter that needs to be estimated, this can be done in a similar way as τ. Further the same reasoning as above can be done. If v_x is positive (the speed in front of a product rises) then speed of the product rises and the other way around.

- νv_{xx}: where ν is the viscosity of the flow, which can be calculated from $\nu = 2\beta\tau c^2$ [Zha03], so no new parameter is introduced. If v_{xx} is positive, products in front are accelerating, then the product speed rises and the other way around. In fluid dynamics v_{xx} is the term that describes viscosity.

The model proposed in this section, (2.3) and (2.16), needs to be validated. This will be done by defining a test case (Chapter 3) and for this test case a discrete event model (DEM) will be build to run simulations (Chapter 4). The results of the DEM will be compared (Chapter 6) with the simulation results of the PDE model (Chapter 5).

Chapter 3

Validation set-up

When a new model is proposed its physical relevance has to be validated. Validation means here making it reasonable that the model is physically relevant. This is done by testing it on a test case with a known result and checking if the new model is a good enough approach. The last means checking if the model is close enough to the reality. In this chapter is defined what is close enough. In the previous chapter a new model for manufacturing systems is proposed. To validate this model its performance is measured by applying the model to a test case. The test case and how the performance of the model is measured is described in this chapter. The test case is a ramp up of a simple manufacturing line. The input of the system is the arrival rate as a function of time and the results of interest are the work in process as a function of time and the cumulative outflux as a function of time. The performance of the model is determined by comparing the results of interest of the PDE model with the results of a discrete event model (DEM) of the test case.

This chapter first explains the test case and the second section describes the relevant variables and the in- and the output. The chapter concludes with an explanation of the performance and describes which experiments are performed. The following two chapters explain how the results of interest are measured from the DEM (Chapter 4) and the PDE model (Chapter 5).

3.1 Description of the test case

To measure the performance of the PDE model a test case is needed. This test case has to incorporate some important physical properties that the PDE model should be able to predict. The most important is transient behavior, since that is the property that first order PDE models could not predict correctly. To get transient behavior a ramp up of a simple manufacturing line is taken with a stochastic arrival rate and stochastic service rates. The manufacturing line consists of ten identical workstations. A graphical

representation of the test case can be seen in Figure 3.1. Since all workstations are identical this manufacturing line is a homogeneous line. In this research the following definition of a workstation is used.

Definition 3.1 (Workstation)**.** A workstation consists of a buffer with infinite capacity and a machine placed in series.

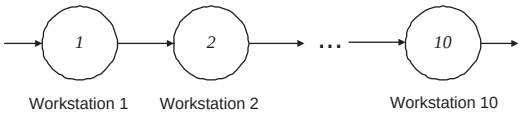

Figure 3.1: The test case, a manufacturing line with 10 identical workstations.

Unless stated otherwise a workstation has an exponential service rate of $\mu = 2$ [products/time unit]. The ramp up of the manufacturing line is the start-up of a line with zero inventory and where all machines are empty, so w is zero. At the start-up time ($t = 0$) the manufacturing line is fed with an exponentially distributed arrival rate λ. An assumption is made that the line has no transport times, no scrap and equipment that does not break down. This keeps the test case insightful. Eventually PDE models should be able to describe more complicated manufacturing lines, to be useful in the control of complex supply chains. For the moment the research focusses on relatively simple lines (without rework or reentrant behavior), if the results for the simple lines are satisfying the lines can be extended to more complex ones. In this research some suggestions will be done how to model more complex manufacturing lines using higher order PDEs, see Appendix A. Now that the test case is known, the relevant variables have to be defined. The next section learns that this is not as simple as it seems.

3.2 Variables of the test case

In the validation study the PDE models are compared to DEMs. When comparing PDEs with DEMs it has to be taken into account that a PDE has continuous variables and a DEM discrete variables. The first variable that has to be defined is the position x. In the PDE model the position x, is a continuous variable, in a manufacturing line position is not defined. To be able to model manufacturing lines with PDEs, position has to be defined in manufacturing lines. The beginning of the line is indicated with $x = 0$ and the end of the line is indicated with $x = 1$. Dividing the machines over the interval [0,1] can be done in several ways. This can be done by distributing position equally over the machines or by making the distribution dependent on the mean flow time or the service rate of the individual machines. This means that a machine with a longer process time gets a bigger amount of position. In our case the choice does not matter, since we assume that all service rates are the same and therefore the flow times of the individual workstations are the same. So the distribution is that every workstation gets

the same amount of position. Since we have 10 workstations every workstation has 0.1 x. More about distributing position over the machines can be found in [Ber04]. For every begin and end of a workstation the x is known, how to define position in between is difficult. A possibility is doing this by making use of the flow time. It is assumed that flow time is linearly related to position, this means that when half the flow time has passed the product travelled half the space interval of the workstation. Problem with this method is that flow time is not known until a product leaves the workstation, so the distribution can only be determined afterwards. For the moment it is not of interest how x is defined in between known places since the validation of the PDE model will only take place at the beginning and end of workstations. In DEMs these are the places where data can be gathered. From position the velocity v can be easily defined as the amount of position consumed by a product in a certain time period. The density ρ is defined as the amount of products on a certain position interval. This leads to the following definitions of x, v and ρ.

Definition 3.2 (Position)**.** Position x is defined on an interval [0,1], where 0 indicates the beginning of the line and 1 the end of the line. The position is distributed proportionally to the mean production time of the workstations, the beginning and end of the workstations are the known points in the position.

Definition 3.3 (Velocity)**.** Velocity v is defined as the amount of position consumed by a product per time unit.

Definition 3.4 (Density)**.** Density ρ is defined as the amount of products on a certain position interval.

Definition of the commonly used parameters as work in process w, throughput δ and the flow time (φ) can be found in [Hop01]. In manufacturing lines density is not a commonly used parameter, but w is. The w and ρ are related according to the following equation:

$$w(t) = \int_0^1 \rho(x,t)dx. \tag{3.1}$$

The w as a function of time is one of the results of interest, the other is the cumulative outflux ($Q(t)$), which can be calculated by taking the integral over time of the outflux (q):

$$Q(t) = \int_0^t q(t)dt. \tag{3.2}$$

The outflux is the throughput at $x = 1$. These two results are chosen as the results of interest since one of the underlying questions is to satisfy a certain demand in time which is related to $Q(t)$. It is possible to calculate the flow time of the system by

subtracting $Q(t)$ from the cumulative inflow $\int_0^t lambdadt$. All variables and the output of the test case are defined now. To perform the validation study the performance of the PDE model has to be defined and different experiments have to be performed. This is described in the next section.

3.3 Measuring the performance

In this research the question is, if the proposed new PDE model for manufacturing systems is a good enough approach and which values for the parameters are the best. By measuring the performance of the model it is possible to see if the model is good enough and it is possible to quantify which setting of the model gives the best result. The performance of the PDE model is measured by comparing its results to the results of a DEM. Simulations using DEM are assumed to be a good enough representation of the reality. So if the simulation results of the PDE model falls within the 95% confidence intervals of the DEM, then can be concluded that the PDE model is a good enough representation of reality. How the results and the 95% confidence interval are obtained from the DEM's is discussed in the next chapter. Obtaining results from the PDE model is discussed in Chapter 5.

In the previous chapter it can be seen that the momentum conservation law is built up from different terms. These terms will be introduced step by step. The parameters in the terms need to be chosen. In order to determine which value or function for a parameter is the best and whether adding a term results in a better performance norms are used. With a norm the difference between two functions can be quantified. The results of interest (w and Q as a function of time) of the PDE are compared to the results of the DEM. This is done by calculating the 2-norm (which can easily be determined numerically) of the w, η_w, and the cumulative outflux, η_Q:

$$\eta_w = \| w_{PDE} - w_{DEM} \|_2 = \int_0^t (w(t)_{PDE} - w(t)_{DEM})^2 dt \tag{3.3}$$

and

$$\eta_Q = \| Q_{PDE} - Q_{DEM} \|_2 = \int_0^t (Q(t)_{PDE} - Q(t)_{DEM})^2 dt. \tag{3.4}$$

The performance of the PDE model will be checked for different utilization levels of the manufacturing line. The utilization is defined as $u = \lambda/\mu$ (μ is the same in all workstations) and the experiments will be done for u = 25, 50, 75 and 90 [%], which corresponds respectively to λ = 0.5, 1, 1.5 and 1.8 [products/time unit].

The test case is defined and the method of validation is known. Before the results are presented first is explained how the results from the DEM and the PDE model are obtained. This is done in Chapter 4 and 5 respectively.

Chapter 4

Discrete event simulation

Discrete event simulation is a well know simulation method for manufacturing systems. Although for large complex manufacturing systems discrete event simulation has relatively high computational cost, the simulation results with discrete event modeling in both transient and steady state situations are good. In this research it is assumed that discrete event simulation of the test case as described in Chapter 3 is a good enough approximation of the reality. The results of the DEM are needed to perform the validation study of the new model for manufacturing systems proposed in the first chapter.

This chapter deals with simulation of the DEM. First is discussed how the DEM is simulated and how the results of interest are subtracted from the simulation results. One of the problems that occur is that a lot of data has to be generated since the model is stochastic and data needs to be averaged. It appears that there are some traps when averaging data as a function of time. The process of averaging is discussed in section 4.2, this section concludes with a good method that will be used in this research. Finally this chapter is concluded with some of the results of the DEM simulation.

4.1 Discrete event simulation of the test case

In Chapter 2 a model for manufacturing lines using higher order PDEs has been proposed. To validate the PDE model, simulation results of the PDE are compared to the simulation results of a discrete event model. This section describes the DEM of the test case. For the presented test case a discrete event model is build and the model is simulated using χ-0.8. Discrete event simulation of a simple manufacturing line is straightforward using χ-0.8, [Hof02]. Here only the most important elements are discussed, the full χ code is presented in Appendix B. The line is built from a standard generator, buffer and machine, only some modifications are made to get the required results. The Buffer (`proc B`) is used to keep track of the state of each workstation. This information is send (on demand) to the Exit (`proc E`), the exit is used to gather

all the individual states of the workstations and send this information to an output file, where this information can be processed. The gathered information in the buffer is the work in process w (the buffer level and the machine status) and the cumulative outflux Q of the workstation (each time a product leaves the workstation, Q increases by 1). The information gathered by the Buffer is send to the exit process. This can be done in various ways and depends on how the collected data is processed. How the data is processed is discussed in the next section. As mentioned before the DEM includes stochastic behavior, so every simulation gives another output. For the validation study the w and the Q as a function of time are needed and since the DEM gives each simulation another realization of the experiment, the results needs to be averaged. To be able to do so, a lot of simulations are needed and a method for averaging the required data as a function of time. The next section deals with the averaging methods.

4.2 Averaging discrete event simulation results

In this research it is important to know how certain variables evolve in time. To get a smooth result from a discrete event simulation with stochastic behavior a lot of data is needed and that data needs to be averaged. The problem is that a lot of events happen that influence the results of interest. These events happen on different moments that differ for each simulation. This has to be taken into account when averaging the results. Averaging simulation data as a function of time can be done in various ways and the question is what method should be used in this research. Three methods are described in the sections below: horizontal averaging (two variants) and vertical averaging. All these methods are visualized by means of an example (averaging w levels as a function of time) and are applied to a test case (averaging w levels in the ramp up of a push line). Finally, a conclusion is made about which method is best to use for averaging variables as a function of time.

4.2.1 Averaging method Horizontal 1

The first method of averaging is a horizontal method since it takes the average of the event times (time is plotted horizontally). Taking the average of the event times results in a list of horizontal events, each horizontal event has a corresponding w (the average of the individual work in process level of the events). An event is defined as the time of an arrival of a product in, or a departure of a product from the manufacturing system. At an arrival the w increases by 1 and at a departure the w decreases by 1. Horizontal averaging determines the average event time, which means it takes from all experiments the n^{th} event and averages these times, this results in a list of n horizontal events. Note that the horizontal average has as many events (n) as the experiment with the least events. For each horizontal event the average of the corresponding w is determined, which means that it averages the corresponding w of every n^{th} event from the experiments. The method of horizontal averaging is visualized by using an arbitrary

data set. This data set contains the events and the corresponding w from two different experiments. In Table 4.1 this data set can be seen and in the last column the results of horizontal averaging is presented. These results are graphically represented in Figure 4.1. Figure 4.1 shows w as a function of time of the two experiments and the w as a function of time of the Horizontal 1 average for the two experiments.

	Experiment 1		Experiment 2		Horizontal 1	
Event number	time	w	time	w	time	w
1	1.0	1	0.5	1	0.75	1
2	2.5	2	2.0	2	2.25	2
3	4.0	3	4.5	3	4.25	3
4	4.75	4	8.0	2	6.375	3
5	7.0	3	11.5	1	9.25	2
6	9.0	2	14.0	2	11.5	2
7	12.5	1	16.0	1	14.25	1
8	15.0	0	18.0	0	16.5	0

Table 4.1: Horizontal 1 w average of two arbitrary experiment2

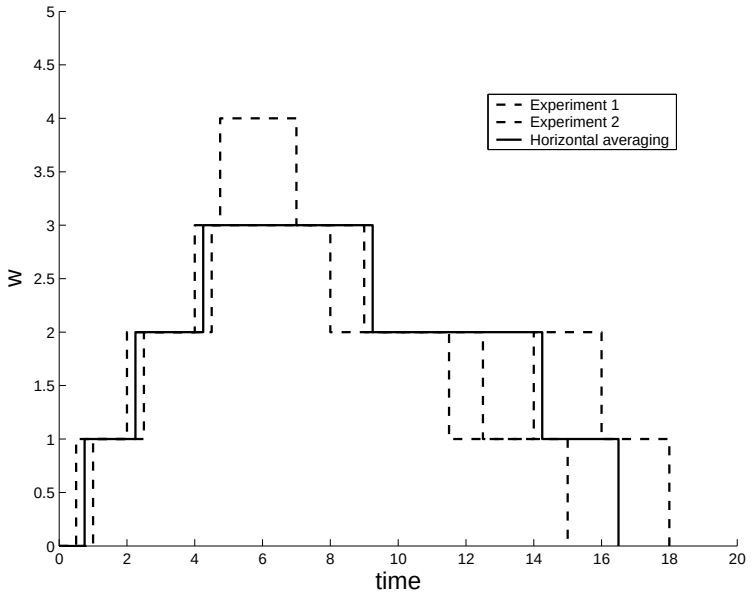

Figure 4.1: Horizontal w average of two arbitrary experiment

4.2.2 Averaging method Horizontal 2

The second method of averaging is also a horizontal method. The difference with Horizontal 1 is that here the events are separated in arrivals and departures. In Horizontal 2 averaging there are two types of events, arrivals of products in the manufacturing system (indicated with an n) and departures of products from the manufacturing system (indicated with an m). Horizontal 2 averaging calculates the average of the n^{th} and m^{th} events for all experiments. These results can be plotted in a time-event diagram. For the data set from Table 4.1 this has been done, as can be seen in Figure 4.2. In this figure the left side curves are the arrivals and the right side curves are the departures.

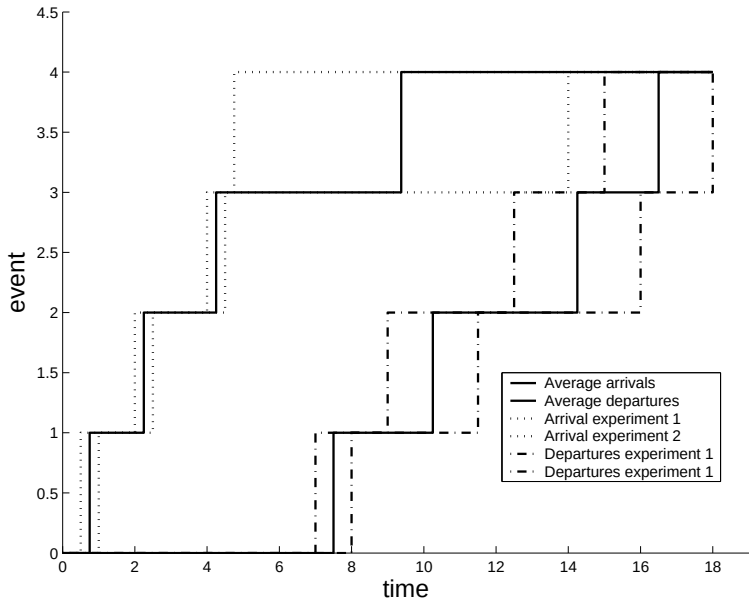

Figure 4.2: Time-event diagram

The w can be subtracted from this time-event diagram, by calculating the vertical difference between the n and m events. Note that w can only take integer numbers. There is no average taken over w. Table 4.2 shows the results for the Horizontal 2 w average of the data set and these results are graphically represented in Figure 4.3. Figure 4.3 shows the w as a function of time of the two experiments and the w as a function of time of the Horizontal 2 average for the two experiments.

Separating arrivals and departure, gives again arrivals and departures for the horizontal average, where at an arrival w increases by one and at a departure, w decreases by one. So w of the horizontal average can only take integer values and does not give a smooth w in time. Smooth results are desired since the behavior of the variables over time is required. Therefore, the Horizontal 2 method will not be considered as suitable solution.

	Experiment 1			Experiment 2			Horizontal 2		
Event number	time	w	Type	time	w	Type	time	w	Type
1	1.0	1	n	0.5	1	n	0.75	1	n
2	2.5	2	n	2.0	2	n	2.25	2	n
3	4.0	3	n	4.5	3	n	4.25	3	n
4	4.75	4	n	8.0	2	m	7.5	2	m
5	7.0	3	m	11.5	1	m	9.38	3	n
6	9.0	2	m	14.0	2	n	10.25	2	m
7	12.5	1	m	16.0	1	m	14.25	1	m
8	15.0	0	m	18.0	0	m	16.5	0	m

Table 4.2: Horizontal 2 w average of two arbitrary experiments

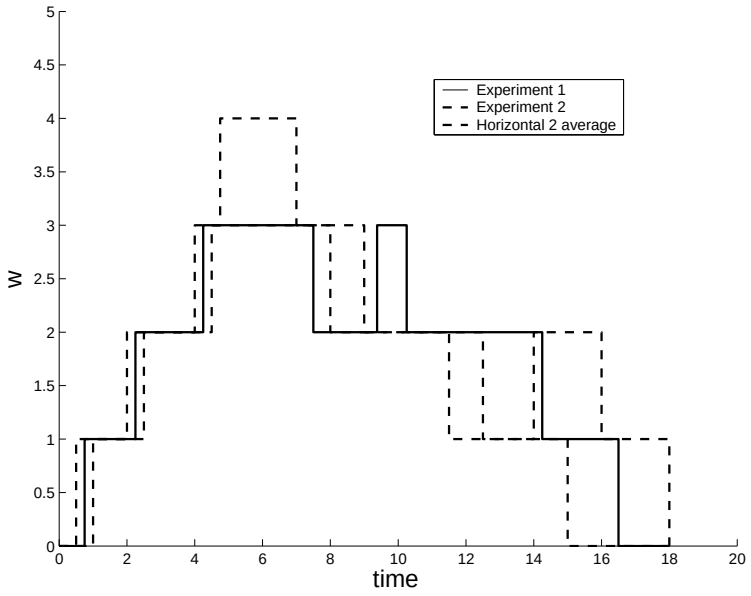

Figure 4.3: Horizontal 2 w average of two arbitrary experiment

4.2.3 Averaging method Vertical

The third method of averaging is a vertical method, which means no average in horizontal direction (time) is taken. Only w levels are averaged. The method of vertical averaging calculates for every event the average w at that moment over all experiments. A list of all events from the experiments is made and for every event in the list the average of the corresponding w in all experiments is calculated. Events are defined in the same way as with Horizontal 1 averaging (arrivals and departures). To show how vertical averaging works, the same data set as in Table 4.1 is used. Table 4.3 shows

the results for the vertical w average of the data set and these results are graphically represented in Figure 4.4.

time	w
0.5	0.5
1.0	1.0
2.0	1.5
2.5	2.0
4.0	2.5
4.5	3.0
4.75	3.5
7.0	3.0
8.0	2.5
9.0	2.0
11.5	1.5
12.5	1.0
14.0	1.5
15.0	1.0
16.0	0.5
18.0	0.0

Table 4.3: Vertical w average of two arbitrary experiments

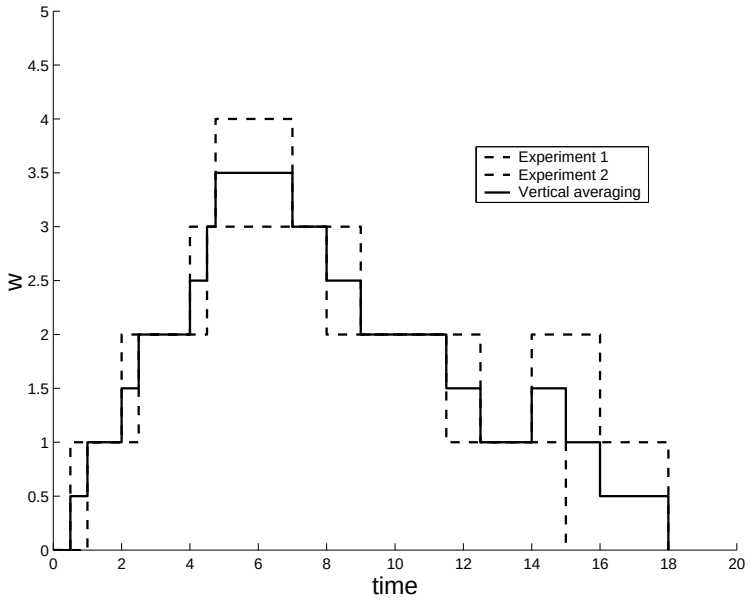

Figure 4.4: Vertical w average of two arbitrary experiment

4.3 Results of the averaging method

In the previous section three methods for averaging are presented. One method, Horizontal 2, has been rejected already. To study the other two methods, Horizontal 1 and vertical averaging, they are applied to a test case. Based on the results a choice for one of the two methods is made. The results are discussed in this section. The test case is the ramp up of a manufacturing line which consists of ten identical workstations. This is the same case as described in Chapter 3. A graphical representation of the test case can be seen in Figure 4.5.

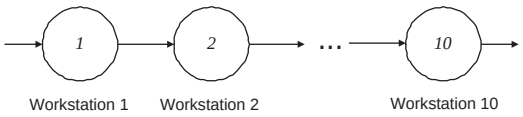

Figure 4.5: The test case, a manufacturing line with 10 identical workstations.

The products enter the system at an arrival rate of 1.0 [products per time unit], resulting in an utilization of the system of 50%. The systems is empty at the begin of an experiment, so w at $t = 0$ is zero. From this system the w as a function of time is desired. This can be done by averaging the results of a discrete event simulation of this system. The discrete event model used for these simulations can be seen in Appendix B, where the Exit is adapted to the method of averaging. In total 1000 experiments were done, on these experiments both methods for averaging (horizontal and vertical) were applied. The script containing both methods can be seen in Appendix B. The results for the average of 1000 experiments is shown in Figure 4.6.

Two things strike from this figure. One is that the steady state level of the horizontal w average is above the steady state level of the vertical w average. Queueing theory predicts a steady state w of 10, [Kle75]. The horizontal average is in steady state always above $w = 10$ and the vertical average is in steady state around $w = 10$. The second thing that strikes, is that vertical averaging has much smaller steps than horizontal averaging. Zooming in on an arbitrary part of the figure makes this clear, see Figure 4.7. This is logical, since the horizontal methods have as many steps as the number of events in an experiment and the vertical method has as many steps as the number of experiments times the number of events in a simulation. So, vertical averaging has much more steps and therefore smaller step sizes.

The fact that the steady state value of horizontal averaging lies significantly above 10 indicates that something is wrong with this method. To make sure that this cannot be a stochastic phenomenon a 2-sided 95% confidence interval is plotted around the horizontal averaging results. This shows that the lower boundary of the interval is also above the value 10, as can be seen in Figure 4.8. So it can be concluded that the horizontal method as described above has some flaws in it.

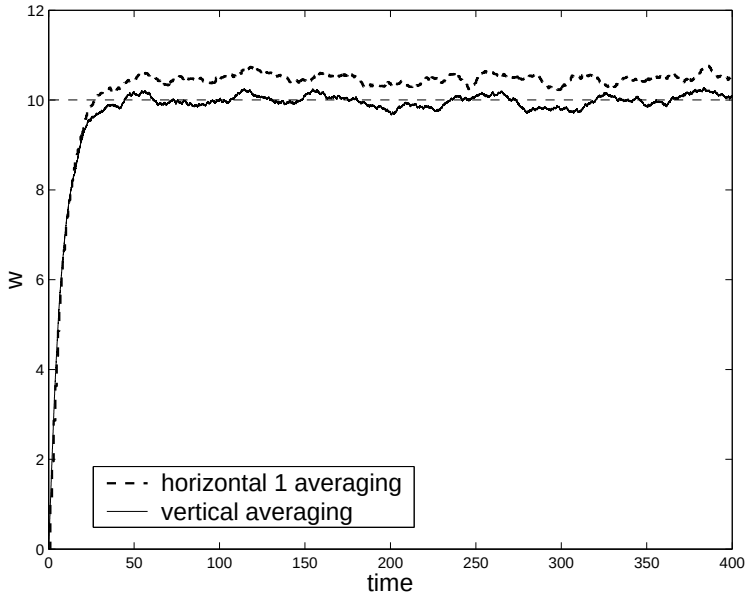

Figure 4.6: Vertical and horizontal w average of 1000 experiments.

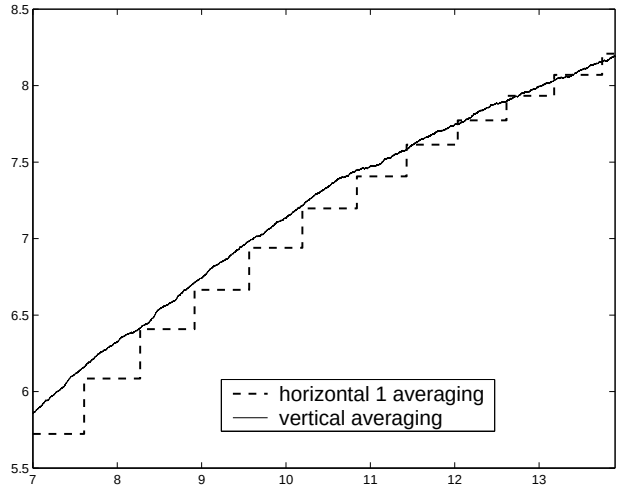

Figure 4.7: Zooming in on vertical and horizontal averaging.

The reason that horizontal averaging calculates the wrong average is that departures and arrivals are mixed up. This means that the horizontal average of the n^{th} event is a mix of departure and arrivals. The problem is that departures are not independent of arrivals, so the events are not independent. In vertical averaging this is not a problem

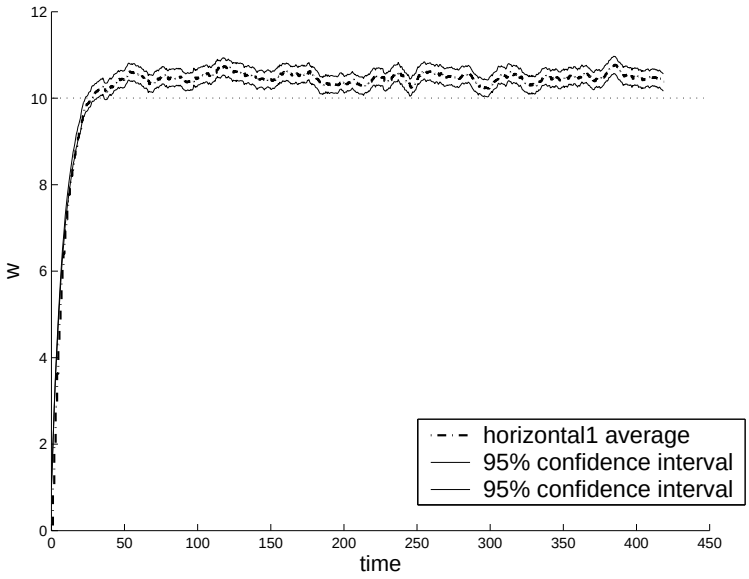

Figure 4.8: Horizontal w average of 1000 experiments with 95% confidence intervals.

because no average is calculated over the events. For every event the average w is calculated. Another problem with horizontal averaging is that it takes the average of the n^{th} event and calculates a corresponding w from events that took place at another time, it can happen that the horizontal w is above or under all w of the experiment, as can be seen in Figure 4.1 from time $t = 12.5$ until $t = 14.0$. This peculiarity also occurs with the Horizontal 2 method (Figure 4.3, t= 8.0 until t= 9.38).

4.3.1 Conclusion of the averaging method

For the validation study w as a function of time is needed. From this requirement the horizontal averaging method can be rejected, because it does not predict the average w correctly. The vertical averaging method predicts the w correctly and has a smaller step size, so the result is much more smooth than by horizontal averaging. The step-size can easily be decreased by carrying out more experiments. The result of the horizontal averaging keeps always the same step-size, because the step-size depends on the events and the expected number of events per time-unit is fixed. From these results vertical averaging is chosen to average the w as a function of time in this research. One thing needs to be mentioned when using the vertical method. The vertical method needs a lot of calculation time when a lot of experiments are done. This is since for every event the new w needs to be calculated. In the rest of this research data will not be gathered for every event that happens, but the measurements will take place on fixed time intervals (sampled measurement). Results for the test case are presented in the next section.

4.4 Results of the discrete event simulation

In the previous section a method for processing data from a discrete event simulation has been discussed. This method is applied on the test case as described in Chapter 3 and the results of interest, the w and Q as a function of time, are collected. In total 5000 experiments are done to determine the averages. The measurements are done on a sampled basis. The sample interval is chosen 0.5 [time units], since sampling reduces the calculation time and this interval is small enough to get a smooth result. In Appendix B the files used for the processing of the data are presented. Figure 4.9 shows the results of interest of the test case for a utilization level of 50% and $\lambda = 1.0$ [products/time unit].

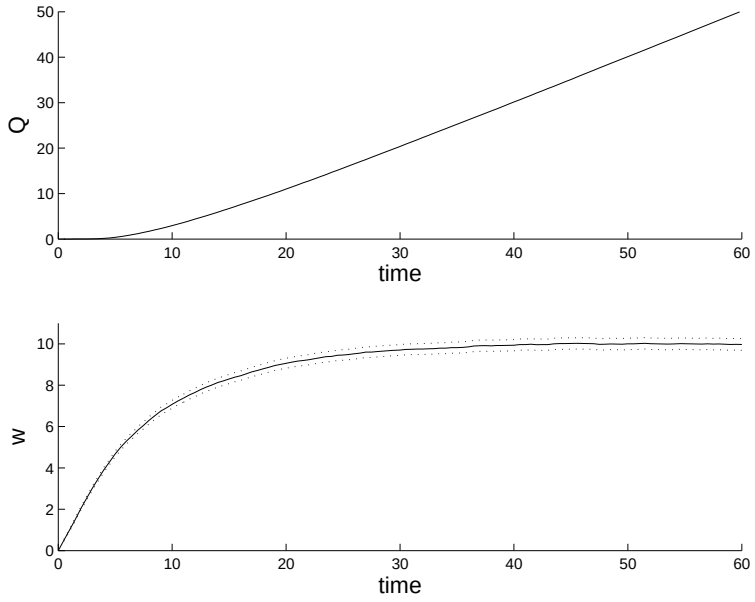

Figure 4.9: Test case results, w and Q with 95% confidence intervals.

The figure shows the w and the Q as a function of time together with the 95% confidence interval. Since a lot of experiments were done, the 95% confidence interval is small and hard to distinguish from the other lines.

4.5 Conclusion of the discrete event simulation

In this chapter a method to gather discrete event simulation results of the test is presented. First a discrete event simulation method using χ-0.8 was presented. A discussion on averaging the simulation data resulted in a good averaging method. In the last section a result of the discrete event simulation is presented (for u =50%). Now that a

test case has been presented, discrete event simulation results can both be gathered and processed, the first part of the validation study has been done. In the next chapter the second part of the validation study is discussed, the simulation of the PDE model. After that the validation of the PDE model for manufacturing can be performed. In Chapter 6 the results of this study can be seen.

Chapter 5

A PDE model for manufacturing systems

Partial differential equations (PDEs) are commonly used to model flows of fluids. Since in 1955 the LWR model [Lig55, Ric56] was introduced, PDEs are frequently used to model traffic flows. Commonly used traffic flow models based on higher order PDEs have been discussed in Chapter 2. In manufacturing systems PDEs are not frequently used to build models. Suggestions to model manufacturing systems have been made by several authors, [Lef04, Arm04]. In this research the proposed model for manufacturing systems is based on higher order traffic flow models. How the proposal is made has been illustrated clearly in Chapter 2. This chapter will discuss the model in more detail in Section 5.1. To simulate the PDE model initial and boundary conditions are needed as well as a method to solve higher order PDEs, this is discussed in Section 5.2 and Section 5.3 respectively. Finally some results of the simulation of the test case will be shown in Section 5.4. The validation study where the PDE results are compared to the DEM results is discussed in the next chapter.

5.1 The PDE model

The proposed model for manufacturing systems from Chapter 2 is discussed here in more detail, for clearness the model is printed here again:

$$\rho_t + q_x = 0, \tag{5.1}$$

$$v_t + v v_x = \frac{v^e(\rho) - v(x,t)}{\tau} - \frac{c^2}{\rho}\rho_x - 2\beta c v_x + \nu v_{xx}. \tag{5.2}$$

The mass conservation law, $\rho_t + q_x = 0$, and the left hand side of the momentum conservation law, $v_t + vv_x$, have already been discussed in detail in Chapter 2. Here the right hand side of momentum conservation, $\frac{v^e - v}{\tau} - \frac{c^2}{\rho}\rho_x - 2\beta c v_x + \nu v_{xx}$, is discussed term by term. The terms are given the following names: relaxation ($\frac{v^e - v}{\tau}$), anticipation ($-\frac{c^2}{\rho}\rho_x$), memory ($-2\beta c v_x$) and viscosity (νv_{xx}).

5.1.1 Relaxation

As mentioned before, queueing theory is not able to predict transient behavior, but it gives good steady state results. Therefore, queueing theory is used to determine the equilibrium speed v^e of a manufacturing system. The equilibrium speed is needed for the relaxation term, $\frac{v^e(\rho) - v(x,t)}{\tau}$. The relaxation reflects the time a system needs to reach an equilibrium. If the model is in steady state (for a homogeneous system), then the derivatives to time, place and density in (5.2) are zero. Substituting this in the momentum conservation law gives the following: $0 = \frac{v^e - v}{\tau} - 0 - 0 + 0$. This can only be true if v^e equals v. So then the speed of the system must be the equilibrium speed.

For v^e different possibilities exist. In this research two possibilities from queueing theory are taken. These possibilities can easily be derived from queueing theory, noticing that speed is the reciprocal of the flow time. The flow time φ can easily be determined by using Little's law [Lit61] $\varphi = \frac{w}{\delta}$, where δ is the throughput. It is assumed that the utilization $u < 1$, so $\delta = \lambda = u\mu$. Rewriting using $w = u + \frac{u^2}{1-u} = \frac{u}{1-u}$ gives the following relation for the flow time:

$$\varphi = \frac{1 + w}{\mu}. \tag{5.3}$$

The w in (5.3) is the local w in the workstation. This relation holds for a single workstation, but since for a manufacturing line with an exponential arrival and service rate all workstations work independent [Bor04, Kle75], this relation can be used to determine the equilibrium speed for the test case. The equilibrium speed is the reciprocal of the flow time. To calculate v^e two choices are possible. One possibility is to calculate v^e with w of the complete line, this results in:

$$v^e(t) = \frac{\mu}{m + w(t)}, \tag{5.4}$$

where m is the number of machines in the manufacturing line. In (5.4) v^e is dependant on time, but not on the position. The equilibrium speed is the same in the line despite varieties in local loads of the system. Equation 5.4 is used by Armbruster [Arm02]. The other possibility is to calculate v^e using the local w, which is in fact the density, this results in:

$$v^e(x,t) = \frac{\mu}{m + \rho(x,t)}. \tag{5.5}$$

In (5.5) v^e is dependant on time and position, so local variations in the load are taken into account. Equation 5.5 is used in [Lef04, Ber04]. Both (5.4) and (5.5) will be used in the experiments.

The relaxation time τ is a non-zero positive parameter. The value of τ has to be found. This can be done by fitting the PDE on the results of a discrete event simulation. This is discussed in Chapter 6.

5.1.2 Anticipation

Anticipation describes the effect that cars or products react on the density they notice in front of them. If ρ_x is positive (the density in front of a product rises), then the speed drops and if ρ_x is negative (the density in front of a product drops) then the speed rises. This is easy to understand, also if you realize that speed is related to the flow time of a product. The flow time rises when the w rises, and then the speed drops and the other way around. In Zhang's model the anticipation is described by the following term: $-\frac{c^2}{\rho}\rho_x$, where c is the product sound speed. Zhang proposes the following formula for the traffic sound speed:

$$c = \rho v^e_\rho \tag{5.6}$$

The traffic sound speed describes how fast a small disturbance propagates in an upstream direction (backwards). The derivative of v^e to ρ is negative, since v^e is inversely proportional to ρ, as can be seen in (5.4) and (5.5). Therefore c is a negative parameter.

5.1.3 Memory

Memory has just like anticipation to do with the effect that cars or products react to things that happen in front of them. Memory depends on the difference in the current speed ($v(x,t)$) and the speed of the cars or products in front ($v(x+dx),t)$). The term is called memory since the future speed ($v(x,t+dt)$) depends on the current speed. Zhang describes memory by $-2\beta c v_x$, where β is the second parameter that needs to be estimated (the other one is the relaxation time τ). The same reasoning can be done as with anticipation. If v_x is positive (the speed in front of a product rises) then speed of the product rises and the other way around.

5.1.4 Viscosity

In fluid dynamics v_{xx} is the term that describes viscosity. In traffic flows viscosity can be seen as the tendency of drivers to go along with the flow. The term of the proposed model (5.2) that describes viscosity is νv_{xx}. Zhang [Zha03] uses the following formula to calculate ν:

$$\nu = 2\tau c^2 \beta. \tag{5.7}$$

So ν is related to the relaxation, anticipation and memory, if one of them is zero, than no viscosity exits. If v_{xx} is positive, products in front are accelerating, then the product speed rises and the other way around (ν is a positive parameter).

5.2 Boundary conditions and initial values

Before the model can be simulated the borders of the system has to be defined. Two different kind of borders (conditions) are distinguished, initial conditions and boundary conditions. The initial conditions describe the state of the system at $t = 0$ for all x, which is the state of the system at the beginning of the experiments. The initial values are defined in the test case, the system is empty at $t = 0$, so the $\rho(x, t = 0) = 0$ for all x. If the system is empty at the start-up, the speed is maximal; $v(x, t = 0) = v_{\max}$. The maximum speed $v_{\max}$ is determined by the service rates of the machines. This can also be seen in (5.4) ($w(t) = 0$) and (5.5) ($\rho(x, t) = 0$).

The boundary conditions describe the state of the system at the borders, in this case this is the input of the system, so the boundary conditions give the state of the system at $x = 0$. Here the influx and the speed are taken as the left boundary condition. The influx is $q(x = 0, t) = \rho(x = 0, t) \cdot v(x = 0, t)$ and the speed is the equilibrium speed v^e (5.4) or (5.5). Notice that the equilibrium speed (in both cases) is dependant on the load of the system and not constant during an experiment. For v^e in the boundary conditions is chosen for (5.4), since this is the easiest to implement in the numerical solution method. In Chapter 6 it is shown that (5.4) and (5.5) give comparable results. Now that the borders of the system are defined the model can be simulated.

5.3 Simulation results from the PDE model

To get simulation results from the PDE a simulation tool has to be used. Here the C++ code developed by T.C. Jo, [Jo03] is used. More about the C++ code can be found in Appendix C. This code used a backward scheme to get a numerical solution [LeV98]. So the PDE model has to be rewritten according to this method. First we deal with the mass conservation law, applying the backward numerical method this becomes:

$$\frac{\rho(x,t+\Delta t)-\rho(x,t)}{\Delta t}+\frac{q(x)-q(x-\Delta x)}{\Delta x}=0. \tag{5.8}$$

Rewriting gives:

$$\rho(x,t+\Delta t)=\rho(x,t)-\frac{\Delta t}{\Delta x}(q(x)-q(x-\Delta x)). \tag{5.9}$$

Applying this method for the momentum conservation without any external forces $v_t + vv_x = 0$ this gives:

$$v(x,t+\Delta t)=v(x,t)-0.5\frac{\Delta t}{\Delta x}(v^2(x,t)-v^2(x-\Delta x,t)). \tag{5.10}$$

For the complete momentum conservation law this gives:

$$\begin{aligned} v(x,t+\Delta t) &= v(x,t)-0.5\frac{\Delta t}{\Delta x}(v^2(x,t)-v^2(x-\Delta x,t)) \\ &\quad +\Delta t\frac{v^e(\rho)-v(x,t)}{\tau} \\ &\quad -\frac{\Delta t}{\Delta x}\frac{c^2}{\rho(x,t)}(\rho(x,t)-\rho(x-\Delta x,t)) \\ &\quad -\frac{\Delta t}{\Delta x}2\beta c(v(x,t)-v(x-\Delta x,t)) \\ &\quad +\frac{\Delta t}{\Delta x}\nu(v(x,t)-2v(x-\Delta x,t)+v(x-2\Delta x,t)) \end{aligned} \tag{5.11}$$

These equations can be used in the C++ code to get a solution from the PDE-model for the test case. The initial and boundary conditions are described in Section 5.2. In (5.11) in the last line is a second order derivative of v, which results in a difference formula that looks two steps (∂x) back: $v(x-2\partial x)$, for the boundary condition here also the equilibrium speed is taken. Besides boundary conditions also the stability of the numerical solution is essential to get reliable solutions. The C++ code used here uses the Courant, Friedrichs and Lewy (CFL) condition [LeV98], which states that the speed of the solution ($\frac{\Delta x}{\Delta t}$) should be lower then the speed of the flow, v. Now all the ingredients are available to perform the experiments.

5.4 Results

In the previous sections the method to simulate PDE models is described. In Figure 5.1 the results of interest (w and Q as a function of time) are shown. The results shown

below are the simulation results for the test case as described in Chapter 3, the utilization $u = 50\%$ and $\lambda = 1.0$ [products/time unit]. The PDE model used is the mass conservation law (5.1) and the momentum conservation law (5.2) without any righthand side terms ($v_t + vv_x = 0$). This model is also used in [Ber04, Lef04].

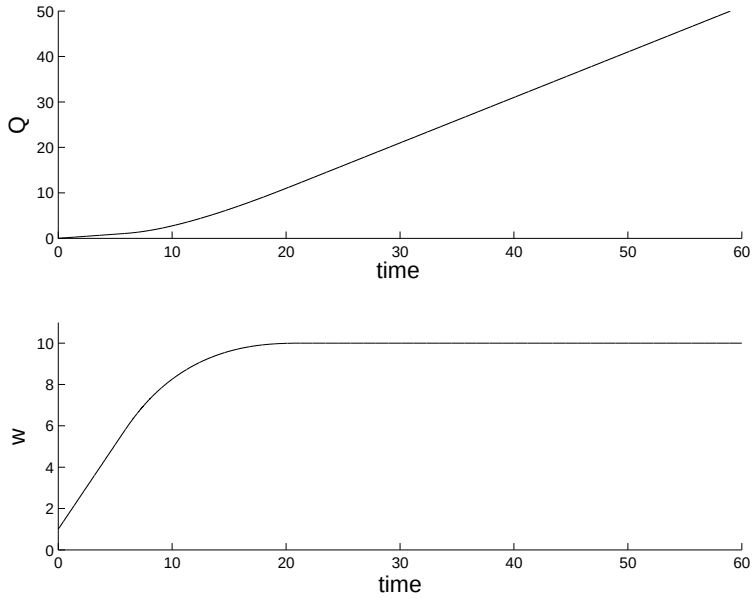

Figure 5.1: Test case result for the PDE model.

The complete validation study, with all the terms (relaxation, anticipation, memory and viscosity), is performed in the next chapter.

5.5 Conclusions

In this chapter the PDE model proposed in the first chapter is discussed in more detail. Also a method to simulate this model is presented, this includes a solution scheme, initial conditions and boundary conditions. In the next chapter the validation study for the PDE model is performed. This is now possible, since both the DEM and the PDE simulations are discussed. The expectations are that adding the extra terms (relaxation, anticipation, memory and viscosity) will results in a better performance of the PDE model. Also a better understanding of the influences of these terms on the PDE model can be expected.

Chapter 6

Validation study

In this research a new model for manufacturing systems is proposed based on higher order PDEs. When a new model is proposed the physical relevance has to be validated. In Chapter 3 a case is defined and this case is used to determine whether the model is physically relevant. The model is validated by comparing the simulation results of the PDE-model with the simulation results of the DEM. Chapter 3 discussed when a model is called physically relevant or in other words, when the model is considered a good enough approach of the reality. The criterium is that the results of the PDE model should be in the 95% confidence interval of the discrete event simulation results of 5000 experiments. In the validation study the model is tested in different set-ups. To compare the different setups, the performance of the model has to be defined. Performance is defined (as pointed out in Chapter 3) as the 2-norm of the results of interest, the work in process w and the cumulative outflux Q as a function of time, $\eta_w = \| w_{PDE} - w_{DEM} \|_2$ and $\eta_Q = \| Q_{PDE} - Q_{DEM} \|_2$. The lower η_w and η_Q are, the better the performance is.

This chapter is constructed as follows, first the experiments with the proposed model without the relaxation, anticipation, memory and viscosity term are discussed. This model is presented in [Arm02] and is used to see if adding additional terms yield in a better result. In Section 6.2 the relaxation term is added, in Section 6.3 the anticipation term, and finally in Section 6.4 memory and viscosity. The chapter ends with a conclusion about the performance of the model and the physical relevance.

6.1 No terms

In previous research [Arm02, Ber04, Lef04] on the use of PDEs in manufacturing systems a higher order PDE model has been studied already. This model is a combination of the mass conservation law and the momentum conservation law without any right hand side terms:

$$\rho_t + q_x = 0 \tag{6.1}$$

and

$$v_t + vv_x = 0. \tag{6.2}$$

This model is presented by Armburster [Arm02] and is validated in [Ber04, Lef04]. Van den Berg and Lefeber [Ber04, Lef04] conclude that this model describes the steady state accurately, but that the transient state differs too much from the reality (the DEM). In the validation study the results of (6.2) are used to compare the other results to. Comparing the results means checking if adding the extra terms to the model bring an improvement in the performance. In Table 6.1 the 2-norms of the results of interest are shown. Figure 6.1 shows the curves of the result of interest of the DEM and (6.2) for u = 50%.

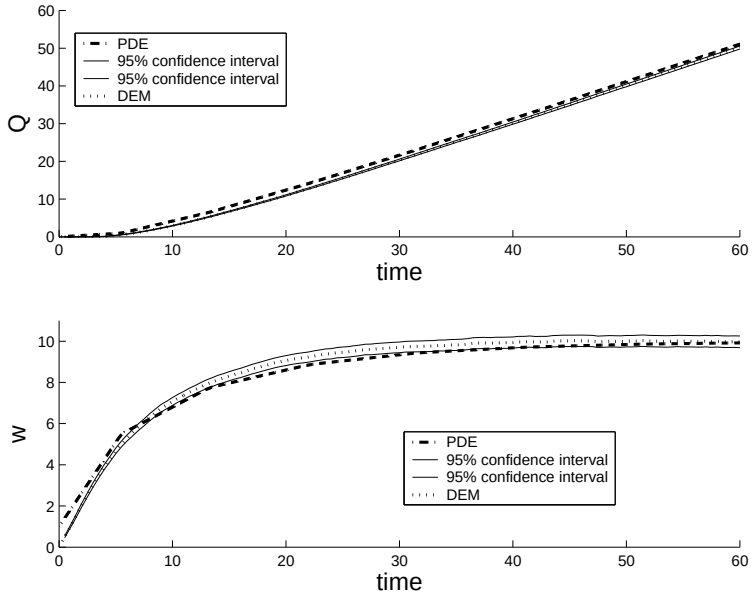

Figure 6.1: Results of interest from (6.2) for u= 50%

u=25%		u=50%		u=75%		u=90%	
η_w	η_Q	η_w	η_Q	η_w	η_Q	η_w	η_Q
3.80	189.1	10.13	133.46	436.7	1191	6762	8556

Table 6.1: Results of interest from (6.2) for u= 25, 50, 75 and 90%

In Figure 6.1 it is hard to distinguish the curves, therefore in Figure 6.2 the error of the PDE model is plotted. The error is the result of the PDE model subtracted by

the result of the DEM. In Figure 6.2 also the 95% confidence interval is plotted with respect to the time axis. What can be seen in Figure 6.2 is that in the transient state, up until t=60, the PDE model (6.2) is most of the time outside the 95% confidence intervals of the DEM. This is a confirmation of the conclusion of [Ber04, Lef04]. One of the remarks in [Ber04] is that in (6.2) terms can be added to the right hand term, that might improve the performance of the model. In the next sections this is done by adding the relaxation, anticipation, memory and viscosity.

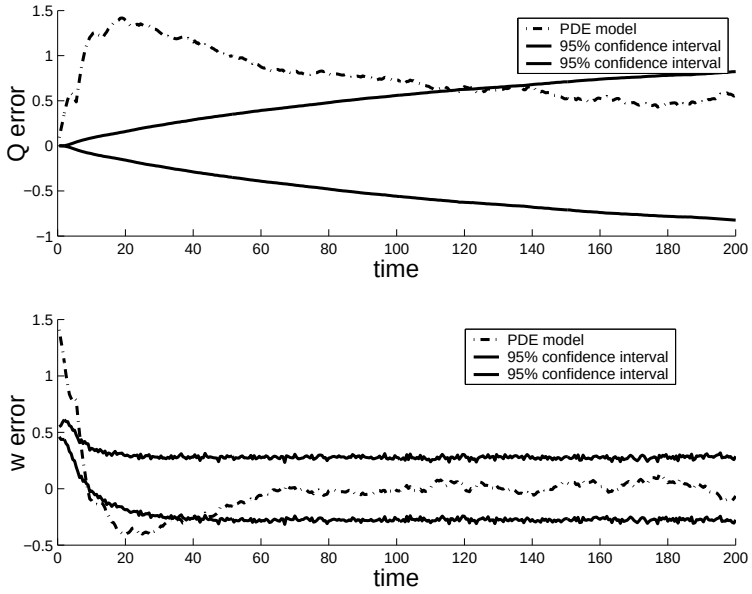

Figure 6.2: Error of the w and Q (6.2) for u= 50%

6.2 Relaxation

One of the drawbacks of (6.2) is that this equation (which describes the development of the velocity in time) has no dependance on $\rho(x,t)$. The speed in the line depends only on the load of the system through the left boundary condition. The left boundary condition is the influx λ and $v^e(x=0,t)$, where v^e depends on $w(t)$. Variations of ρ in the line are not taken into account. The relaxation term can solve this shortcoming, since v^e is dependent on $w(t)$ (option1) or $\rho(x,t)$ (option2). The relaxation terms also represents the fact that it takes time for a system to reach a steady state. This is modeled by τ, a parameter whose value has to be estimated.

The PDE model with relaxation is:

$$\rho_t + q_x = 0$$

and

$$v_t + vv_x = \frac{v^e(\rho) - v}{\tau}. \tag{6.3}$$

The variable v^e in (6.3) has two possibilities to be modeled, option 1:

$$v^e = \frac{\mu}{m + w(t)} \tag{6.4}$$

and option 2:

$$v^e = \frac{\mu}{m + \rho(x,t)}. \tag{6.5}$$

The results for (6.3) with different v^e are shown in Table 6.2 and Table 6.3. Table 6.2 shows option 1 and Table 6.3 shows option 2.

	u=25%		u=50%		u=75%		u=90%	
τ	η_w	η_Q	η_w	η_Q	η_w	η_Q	η_w	η_Q
0.25	-	-	16.2	86.4	200	60	-	-
0.50	-	-	14.6	88.0	162	58	-	-
1.0	4.2	186	12.2	91.4	102	66	2606	1641
2.0	-	-	-	-	37	117	1225	617
2.5	-	-	8.84	101	-	-	786	333
3.0	4.0	187	-	-	18	191	-	-
4.0	3.9	187	8.1	107	21	268	130	57
5.0	**3.9**	**187**	**7.9**	**111**	**35**	**340**	**36**	**160**
6.0	3.9	188	8.0	113.5	53	406	91	378
7.5	3.88	188	-	-	-	-	-	-
10	3.86	188	8.31	120	126	600	855	1577
15	3.84	188	8.71	123.7	-	-	-	-
25	3.83	188	9.16	127	-	-	-	-
50	3.81	189	-	-	-	-	-	-

Table 6.2: Results of interest from (6.3) and for v^e option 1

From Table 6.2 and Table 6.3 τ can be estimated. For both option 1 and option 2 in the case u=25% it can be noticed, that the bigger τ the better the performance for η_w, but the improvement is minimal. The performance of η_Q improves, when τ is lowered, but the improvements are small.

For u= 50, 75 and 90 % it is possible to indicate an optimal value of τ for the performance of η_w. For option 1 τ=5.0, 3.0 and 5.0 gives the best η_w performance for respectively u=50, 75, 90 %. For option 2 τ = 9.0–11.0, 9.0–10.0 and 12.0–13.0 gives the best results for η_w performance.

	u=25%		u=50%		u=75%		u=90%	
τ	η_w	η_Q	η_w	η_Q	η_w	η_Q	η_w	η_Q
0.25	4.0	185	-	-	-	-	-	-
1.0	4.4	185	43.6	89.6	7240	9200	-	-
4.0	-	-	9,65	96.5	-	-	-	-
5.0	-	-	8.76	100	-	-	9000	10000
6.0	-	-	8.26	103	-	-	-	-
8.0	-	-	7.90	108	58,4	288	-	-
9.0	3.9	188	7.85	110	35.3	303	1213	1410
10	**3.9**	**188**	**7.84**	**112**	**35.2**	**335**	**560**	**738**
11	-	-	7.84	112	43.9	373	204	395
12	-	-	7.90	114	-	-	187	467
12.5	-	-	-	-	-	-	176	500
13	-	-	-	-	-	-	184	541
14	-	-	-	-	-	-	254	673
15	3.9	188	8.0	117	86.16	499	342	837
20	-	-	-	-	-	-	924	1680
17.5	-	-	-	-	-	-	622	1263
25	3.8	188	-	-	-	-	1511	2440

Table 6.3: Results of interest from (6.3) and for v^e option 2

Table 6.2 show for u=50 and 75% that the smaller τ becomes, the better the performance of η_Q is. For u= 90%, a τ of 4.0 gives the best results. Table 6.3 shows for u=50% that the smaller τ becomes the better η_Q. For u=75 and 90% τ should be 8.0 and 11.0 respectively to get the smallest η_Q.

For both cases (option 1 and 2) it is desired to have a τ that gives good results for different utilizations. For $v^e = \frac{\mu}{m+w(t)}$ (option 1) this is τ=5 and for $v^e = \frac{\mu}{m+\rho(x,t)}$ (option 2) τ=10. It is remarkable to see that option 1 and 2 give comparable performances of the model, for example in case u=50%; η_w= 7.9 and 7.8 and η_Q=111 and 112 for option 1 and 2 respectively. This can be explained since the test case is a homogeneous line, so the speed throughout the line is very much the same, so dependence on x for v^e does not give an improvement. This can also be seen in Figure 6.3, which shows the error of the w and Q as a function of time for option 1 and option 2 for u=50%. Hardly any visual difference can be noticed between the two options.

Some conclusions have been made about the relaxation term, but not yet in comparison to the model of Section 6.1. Important is that adding the relaxation gives a relevant improvement of the performance of the system as can be seen in Table 6.4. The model can be considered almost good enough, only for $t < 10$ (6.3) is outside the 95% confidence interval.

The value for the performance in the case u=90% and v^e is option 2 attracts attention.

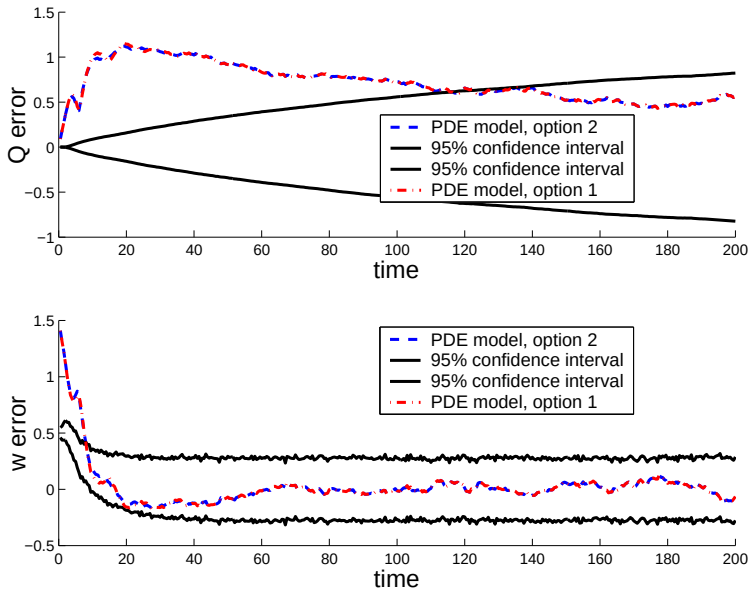

Figure 6.3: Error of the w and Q from (6.3), for option 1 and 2 and τ=5, for u= 50%

	u=25%		u=50%		u=75%		u=90%	
	η_w	η_Q	η_w	η_Q	η_w	η_Q	η_w	η_Q
no terms	3.80	189.1	10.13	133.46	436.7	1191	6762	8556
relaxation1	3.9	187	7.9	111	35	340	36	160
relaxation2	3.9	188	7.84	112	35.2	335	560	738

Table 6.4: Comparison between (6.2) and (6.3) (option 1 and 2)

The performance is significantly worse than for v^e is option 1. The first reason is that τ=10 is not the optimal value in this case. The second reason is an error, probably situated in the numerical solution method. In the next section this error occurs more strikingly and is discussed in more detail. The error is the reason that the results for the case u=90% and v^e is option 2 can be questioned. Therefore, no conclusions have been made on the performance for that case.

Since no difference in performance exits a choice for v^e has to be made on another argument. Expectations are that option 2 give better results in a broader range of cases, for instance a heterogeneous line, since for option 2, v^e is dependant on time and place. In the rest of the experiments option 2 is used to model v^e.

6.3 Anticipation

Adding the relaxation term gave a significant improvement in the performance of the system. The other terms that can be added are anticipation, memory and viscosity. All these terms are look ahead terms, they describe that products are aware of what happens in front of them and react on the conditions in front of them. In this section the anticipation term is added.

The model with the anticipation becomes as follows:

$$\rho_t + q_x = 0$$

and

$$v_t + vv_x = \frac{v^e - v}{\tau} - \frac{c^2}{\rho}\rho_x, \tag{6.6}$$

here $c = \rho v^e_\rho = \rho\frac{\partial v^e}{\partial \rho}$. The choice for option 2 for the equilibrium speed v^e has another benefit, v^e_ρ can now easily be determined, for option 1 this would have been more complicated. The derivative of the equilibrium speed with respect to the density now becomes:

$$v^e_\rho = \frac{-\mu}{(m+\rho)^2}. \tag{6.7}$$

The results for the experiments for different values of u are shown in Table 6.5.

u	η_w	η_Q	Steady State w
25%	11.30	121	3.54
50%	1796	1153	13.51
75%	Unknown error		
90%	Unknown error		

Table 6.5: Results for (6.6) for various utilizations

In Figure 6.4 and Figure 6.5 the results of interest are shown for respectively u=50% and 75%.

Figure 6.4 shows that the PDE reaches steady state, but that the steady value has not the expected w of 10 products, but higher w=13.5. For a utilization u=25% w is also overestimated as can been seen in Table 6.5. This could be explained if this happened in the transient state by the fact that speed drops due to the anticipation term, see Section 5.1. In steady state however ρ_x and $v_t + vv_x$ are zero, so v should equal v^e, see (6.6). For higher utilizations even a bigger errors occurs. In Table 6.5 is mentioned that for u= 75% and 90% an error occurs. This can also be seen in Figure 6.5 for u=75%. The drop in w (and the rise of Q) at t=180 cannot be explained. Such a drop also occurs

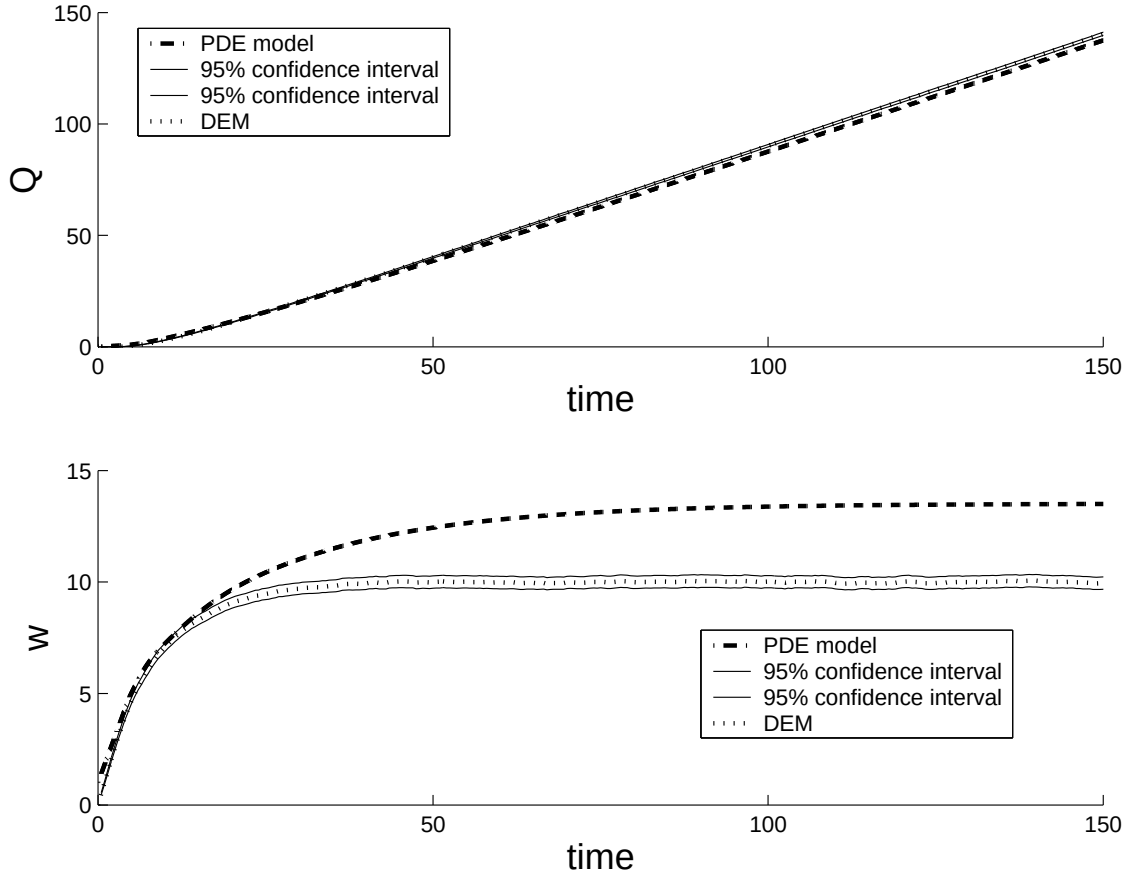

Figure 6.4: Results of interest from (6.6) for u= 50%

for u=90%. So it appears it happens for the higher utilization degrees. When noticing such an unknown effect for which no physical explanation exists the cause could be that the model is not implemented correctly. This however is not likely since the PDE model gives a physical relevant prediction until t=180 and also for the lower utilization degrees, the prediction remains physically relevant (although wrong). The problem might occur due to numerical solution problems. This is however not investigated in this research and stays an open question for further research.

In reality anticipation (and other look ahead terms) can be seen as the control of the manufacturing line, control is able to adjust the speed of the products on basis of the load of the system in front. This can be done for example by lowering the service rate of a machine or it can be an operator which lowers his work speed, since he/she notices that his colleagues in front of him cannot catch up, or he/she even stops working and starts helping his colleagues. However such behavior is not implemented in the test case and therefore adding the anticipation does not bring an improvement in the

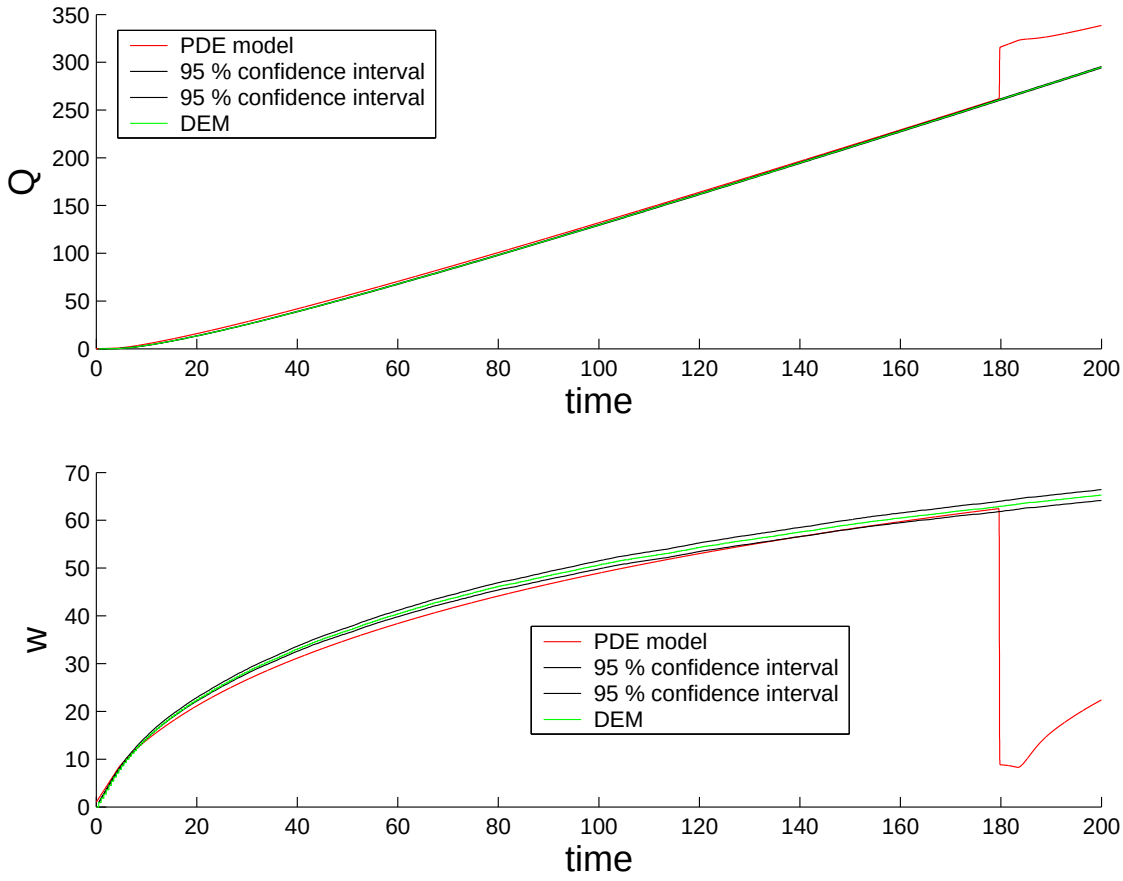

Figure 6.5: Results of interest from (6.6) for u= 75%

performance of the model. When investigating anticipation in manufacturing lines the product sound speed c has to be checked on its relevance, since this is a term that comes from traffic theory. The meaning for the product sound speed in manufacturing systems has to be found to answer this question. It is left for further research to investigate the anticipation. Note that anticipation is a look ahead factor and should not be confused with something like finite buffers, which also makes the speeds drop since buffers in front of a product can become full.

6.4 Memory and Viscosity

Memory and viscosity are like anticipation look ahead terms. Anticipation looks ahead to the density, memory and viscosity react to changes in velocity. For the same reason anticipation did not bring an improvement in the performance, it can be expected that

memory and viscosity also do not bring an improvement. The model with relaxation, memory and viscosity but without anticipation looks as follows:

$$\rho_t + q_x = 0$$

and

$$v_t + vv_x = \frac{v^e - v}{\tau} - 2\beta c v_x + \nu v_{xx}, \tag{6.8}$$

where v^e and c are the same as for (6.6) and β is a parameter whose value needs to be determined. Parameter ν is, as mentioned in Section 5.1, related to relaxation, anticipation and memory. If one of these is not present in the model, there is no viscosity. It is concluded that anticipation does not bring an improvement and is therefore left out of the model, so ν=0. Table 6.6 shows the results for different utilization degrees and different values of β for ν=0. In Section 5.1 is also mentioned that $\nu = 2\tau c^2\beta$, since τ is present in relaxation and c and β are present in memory this is also a possibility for ν. Table 6.7 shows the results for different utilization degrees and different values of β for $\nu = 2\tau c^2\beta$.

	u=25%		u=50%		u=75%		u=90%	
β	η_w	η_Q	η_w	η_Q	η_w	η_Q	η_w	η_Q
0	3.9	188	7.84	112	35.2	335	560	738
0.001	3.9	188	7.84	112	36.3	340	573	758
0.01	3.0	188	7.92	114	50.3	394	164	345
0.012	-	-	-	-	-	-	115	336
0.1	3.83	190.0	32.39	206.13	Error		Error	
1	Error		Error		Error		Error	

Table 6.6: Results for (6.8) for various utilizations and ν=0

	u=25%		u=50%		u=75%		u=90%	
β	η_w	η_Q	η_w	η_Q	η_w	η_Q	η_w	η_Q
0	3.9	188	7.84	112	35.2	335	560	738
0.0001	3.9	188	7.84	112	34.7	335	494	648
0.001	3.9	188	7.84	112	35.3	340	161	235
0.017	-	-	-	-	-	-	116	205
0.018	-	-	-	-	-	-	Error	
0.01	Error		Error		Error		Error	

Table 6.7: Results for (6.8) for various utilizations and $\nu = 2\tau c^2\beta$

Table 6.6 makes clear that for low utilizations (up to u=75%) memory does not bring an improvement in the performance, since for u= 25, 50 and 75% the performance of

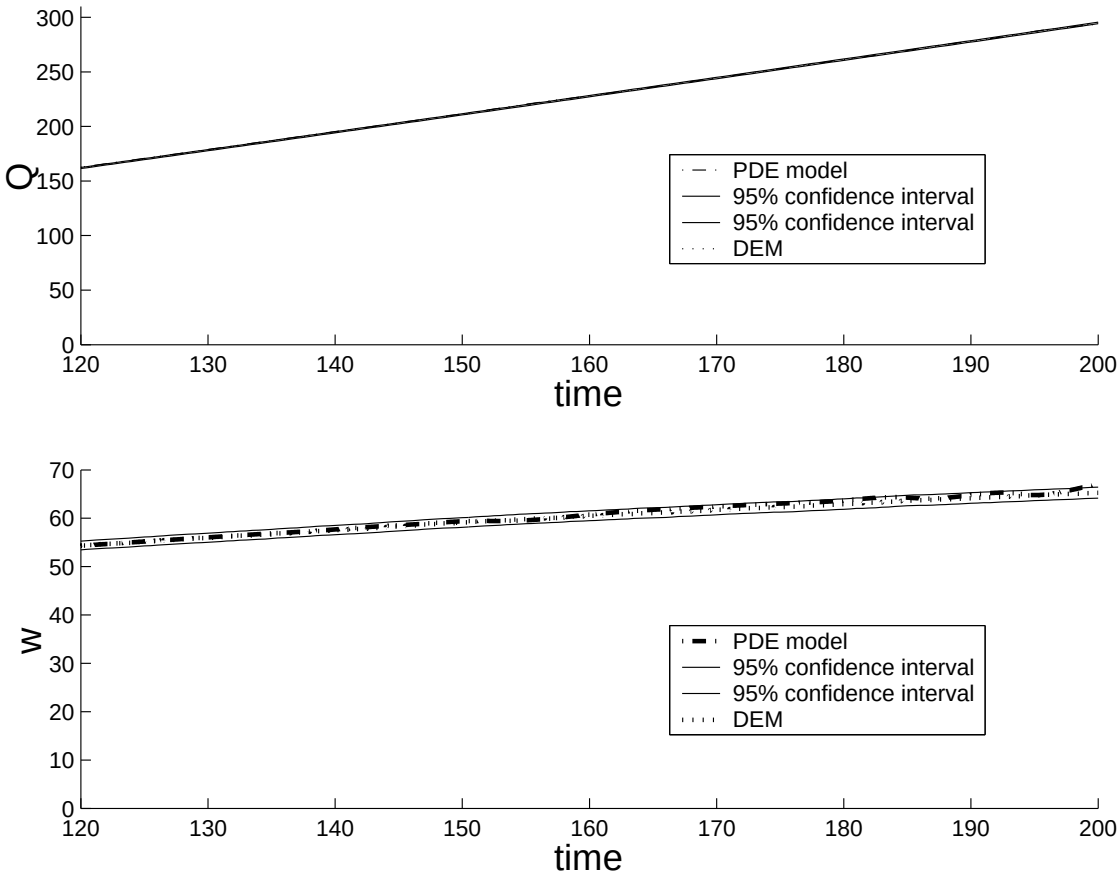

Figure 6.6: Zoom in of the results of interest from (6.8) for u= 90% ($\nu = 0$)

the model for $\beta > 0$ is worse or comparable to the performance for $\beta = 0$. For high utilizations $u = 90\%$, memory brings a significant improvement. This effect needs some further investigation. When taking a closer look at the results for u=90% for (6.3) one can see in Figure 6.7 that, at the end of the interval where the performance is measured, w and Q are out of the trust region of the DEM. This is the same error as discussed in the previous section about anticipation. When adding memory with $\beta = 0.012$ the error is almost completely gone as can be seen in Figure 6.6. Since the origin of the error is unknown it is hard to say why adding a memory term gives improvement.

For the viscosity can be concluded that this term does not bring an improvement in the performance. This can be seen in Table 6.7 where $\nu = 2\tau c^2\beta$ and this does not bring an improvement in the performance compared to Table 6.6 where ($\nu = 0$).

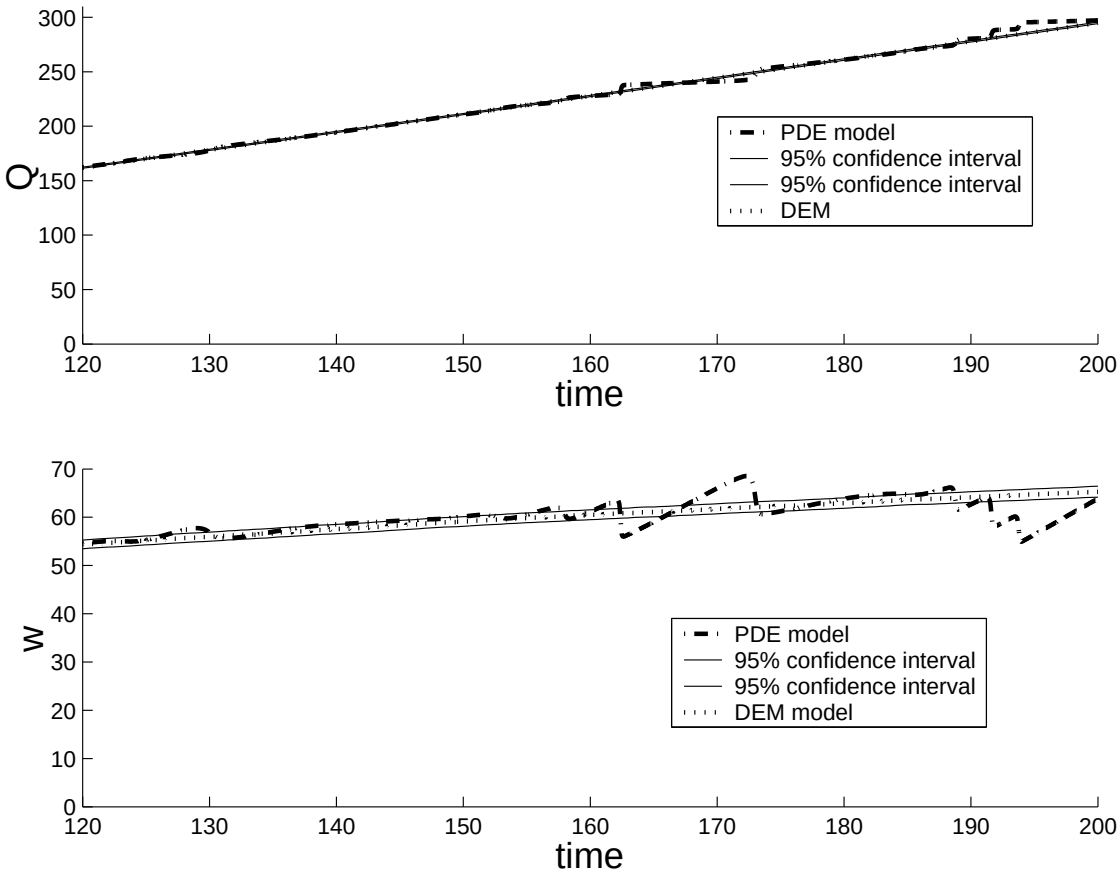

Figure 6.7: Zoom in of the results of interest from (6.3) for u= 90%

6.5 Conclusion

In this chapter a lot of results have been presented, which are summarized here.

The complete model, presented in Chapter 2 and discussed in detail in Chapter 5, has been validated term by term. First, the model without relaxation, anticipation, memory and viscosity (no terms) is studied. Then the model is expanded with the terms and the results are compared to the "no term" model. Below, the main results are presented term by term.

No terms Equation 6.2 gives good steady results, but the transient state is not described correctly.

Relaxation Adding relaxation to the momentum conservation law gives an improvement compared to the model with no terms. For the v^e in the relaxation term two

option are given, (6.4) and (6.5), that give comparable results. The relaxation parameter τ depends on which v^e is chosen; $\tau = 5$ for (6.4) and $\tau = 10$ for (6.5). To model v^e a choice is made for (6.5), since (6.5) depends on t and x and (6.4) depends only on t, so (6.5) is expected to hold for a broader range of situations.

Anticipation Anticipation in combination with relaxation does not bring an improvement to the model, so can be left out. It has to be remarked that in the experiments an unknown error occurred, probably situated in the numerical solution. Further research is necessary. However, it is not expected that solving the error will change the conclusion, since anticipation is a look forward term and in the test case no looking forward is present.

Memory Memory in combination with relaxation does not give an improvement in the performance for low utilizations (up to u=75%). This result was expected since memory is a look forward term just like anticipation. What strikes from the results is that memory reduces the error for high utilization degrees. Since the cause of the error is unknown no explanation has been found. This effect needs further research.

Viscosity Viscosity in combination with relaxation and memory does not give an improvement in the performance and the results do not differ significantly from memory.

The final conclusion is that model (6.3) performs the best and when simulating high utilizations adding a memory term can improve the performance. Important to realize is that an unknown error exists that influenced this conclusion. It is recommended to study the origin of this error. It is expected that this error is situated in the numerical solution method or the C++ code that solves the PDE model numerically. When the error is solved the expectation is that the conclusion holds that relaxation improves the performance. How memory will influence the model after solving the error is unpredictable. The proposed model does not meet the criterium for being good enough. The model does not fall in the chosen trust region of the DEM. When checking the model visually it seems a quit good representation of the reality only not at the start up. This needs further research.

The validation study has been performed and a conclusion about the PDE model has been made. It is clear that the research for PDE models for manufacturing systems does not end here. During this study some ideas arose about how certain behavior can be modeled, these ideas are discussed in Appendix A.

Chapter 7

Conclusions

This research aims to propose a new model for manufacturing systems. This model is composed of higher order PDEs and is based on traffic flow models. The model is needed for the control of complex manufacturing systems, since currently available models are not suited to address the control problem in a desired way. In this research a model is proposed that consists of a mass conservation law and a momentum conservation law. Such a model has been studied before, but that model was not able to predict the transient state correctly. In this research extra terms have been added to the momentum conservation law after examples from traffic flow models. The terms that come from traffic theory needed to get a physical meaning in manufacturing systems. The term with relaxation is translated to manufacturing physics, but the other terms need further research to be understood completely. All terms have been implemented in the model and one by one they have been validated. To perform the validation study a test case has been developed that is the ramp up of a simple manufacturing system. This is a simple manufacturing line with a transient state. From this test case a discrete event model has been built, that is assumed to be a good representation of the actual physical behavior of the line. When collecting the data of the discrete event simulation it appeared that processing the data is not as obvious as it seems.

Averaging of discrete event simulation data

In this research three methods have been investigated to average discrete event simulation data as a function of time. Two horizontal methods that averages the event times and one vertical method that averages the desired variable on certain time instances. It appears that Horizontal 1 predicts uncorrect average values. Horizontal 2 averaging has integer values. The vertical averaging method predicts the correct average and this average is continuous. The vertical method consumes however more calculating time than both other methods, but this can be solved easily by using sampling. It has been

concluded that vertical averaging is the best method out of the three presented methods and that it is sufficient for the purposes in this research.

Numerical solution of the PDE model

To validate the model, PDE simulation results are compared to discrete event simulation results. To simulate the PDE a numerical solution method is used. A C++ code developed by T.C. Jo, [Jo03] has been used to perform the numerical simulations. It appears in some experiments that the results of the PDE model are physically incorrect. The error occurs in cases with a high utilization level. It is believed that these errors occur due to unknown behavior of the numerical solution method. The error influenced some of the conclusions about the model, but still it is believed that most of the conclusions will hold when the error is solved. It is however recommendable to study the error and the solution method before further research using the code should be done.

The PDE model

This research shows that the proposed PDE model brings improvements compared to the previous proposed model based on higher order PDEs. However, the proposed model does not satisfy the conditions set in this research for being good enough. The term from the proposed model that is responsible for the improvements is the relaxation. This conclusion is not influenced by the error that occurs in the numerical solution. The terms anticipation and viscosity do not bring any improvement to the performance of the model. When looking at the physical meaning of the terms relaxation, anticipation and viscosity, these conclusions make sense. The term memory seems in some cases, with high utilizations, to minimize the present error. This conclusion has been highly influenced by the error and it is highly questioned if this term brings an improvement to the performance of the model when the error is solved. The fact that on physical arguments no improvement is expected underlines this.

Chapter 8

Recommendations

The goal of this research was to propose a new physically relevant model for manufacturing systems based on higher order PDEs. During the research some unsolved problems occurred. In this chapter the main problems are mentioned and recommendations are done to overcome these problems. Also some general recommendations for further research are done.

Numerical solution of the PDE model

The main problem is the unknown error that occurs in cases where the utilization in the test case is high. The error is situated probably in the C++ code that contains the numerical solution method. Since the reason for and the exact location of the error is not known yet, it is recommended to study the solution method and the implementation first. Probably the best way to start, is looking for another numerical solution program and look if the error occurs again. This can indicate whether the error has to be searched in the used C++ code or in the used solution method. If the other program give results without the error, then there is probably an error in the C++ code and otherwise, if the other program gives also the error, the backward solution method can be wrong for this purpose.

The PDE model

The proposed model gives improvements in the performance. The relaxation term is mostly responsible for this improvement. The deviation at the start up needs further investigation, but it is believed that this deviation is caused by the numerical solution method or code. The influence of the memory is uncertain due to the error, this needs to be validated again. It is expected that memory will not bring any further improvements for this test case.

Another item that needs further research is the ramp down of a simple manufacturing line. Van den Berg [Ber04] showed that a difference exits between responses of ramp ups and ramp downs. A possible modification to the model might be introducing the velocity variance, as suggested in [Hoo00]. It should be studied if adding this extra PDE improves the model and what its physical meaning is.

The terms anticipation, memory and viscosity are look ahead terms. In the validation method a simple test case is used without look ahead physics. To see if the terms anticipation, memory and viscosity are physical relevant terms a test has to be designed with look ahead physics. This could be for instance in the form of a local controller. In Appendix A some suggestions have been done to add additional parameters to the model, this also needs further research. Finally also complex manufacturing systems with reentrant behavior and rework should be modeled. These items were not addressed in this research, but some work has already be done by Armbruster [Arm02, Arm04].

Bibliography

[Arm02] D. Armbruster, D. Marthaler, and C. Ringhofer. Efficient simulation of supply chains. *Proceedings of the 2002 Winter Simulation Conference*, pages 1345–1348, 2002.

[Arm04] D. Armbruster, D. Marthaler, and C. Ringhofer. Kinetic and fluid model hierarchies for supply chains. *SIAM Journal on Multiscale Modeling and Simulation*, 2(1):43–61, 2004.

[Aw00] A. Aw and M. Rascle. Resurrection of "second order" models of traffic flow. *SIAM Journal Applied Mathematics*, pages 916–938, 2000.

[Ber04] R.A. van den Berg. Modeling and controlling manufacturing systems. Master's thesis, Eindhoven University of Technology, Department of Mechanical Engineering, Systems Engineering, 2004. SE420379.

[Bir60] R.B. Bird, W.E. Steward, and E.M. Lightfoot. *Transport Phenomena*. Wiley, 2nd edition, 1960.

[Bor04] S.C. Borst. Lecture notes on queueing networks, 2004.

[Dag95] C.F. Daganzo. Requiem for second-order approximations of traffic flow. *Transportation Research B*, pages 277–286, 1995.

[Dem97] L. Demetrius. Directionality principles in thermodynamics and evolutions. *Proc. Natl Acad. Sci.*, 94:3491–3498, 1997.

[Hil98] C. Hillman. All entropys agree for an sft. Technical report, University of Washington, 1998. (Online: http://www.informatik.uni-trier.de/∼damm/Lehre/InfoCode/entropy.html).

[Hof02] A.T. Hofkamp and J.E. Rooda. χ *Reference Manual*. Systems Engineering Group, Eindhoven University of Technology, 2002. `http://se.wtb.tue.nl`.

[Hoo00] S.P. Hoogendoorn and P.H.L. Bovy. Continuum modeling of multiclass traffic flow. *Transportation Research Part B*, pages 123–146, 2000.

[Hoo01] S.P. Hoogendoorn and P.H.L. Bovy. State-of-the-art of vehicular traffic flow modelling. *Journal of Systems and Control Engineering*, 2001.

[Hop01] W.J. Hopp and M.L. Spearman. *Factory Physics: Foundations of Manufacturing Management*. Irwin/McGraw-Hill, 2nd edition, 2001.

[Jo03] T.-C. Jo. C^{++} code, 2003. Model for numerically solving PDE-models of manufacturing systems. Personal communication.

[Kla00] A. Klar and R. Wegener. Kinetic derivation of macroscopic anticipation models for vehicular traffic. *SIAM Journal on applied mathematics*, 60(5):1749–1766, 2000.

[Kle75] L. Kleinrock. *Queueing systems, volume 1: Theory*. Wiley-Interscience, 1975.

[Kle76] L. Kleinrock. *Queueing systems, volume 2: Computer applications*. Wiley-Interscience, 1976.

[Kul99] V.G. Kulkarni. *Modeling, Analysis, Design, and Control of Stochastic Systems*. Springer, 1999.

[Lef04] E. Lefeber, R.A. van den Berg, and J.E. Rooda. Modeling, validation and control of manufacturing systems. *Proceedings of the 2004 American Control Conference*, pages 4583–4588, 2004.

[LeV98] R.J. LeVeque. Finite difference methods for differential equations. Technical report, University of Washington, Department of Mathematics, 1998. (Online: http://www.amath.washington.edu/∼rjl/).

[Li02] T. Li. Well-posedness theory of an inhomogeneous traffic flow model. *Discrete and Continuous dynamical systems-series B*, 2(3):401–414, 2002.

[Lig55] M.J. Lighthill and G.B. Whitham. On kinematic waves: I. flow movement in long rivers. II. a theory of traffic flow on long crowded roads. *Proceedings of the Royal Society A*, pages 229:281–345, 1955.

[Lit61] J.D.C. Little. A proof of the queuing formula $l = \lambda w$. *Operations Research*, pages 9:383–387, 1961.

[Pay71] H.J. Payne. Models of freeway traffic and control. *Mathematical Models of Public Services*, pages 51–60, 1971. volume 1 of Simulations Counsils Proceedings Series.

[Ras02] M. Rascle. An improved macroscopic model of traffic flow: derivation and links with the lighthill-whitham model. *Mathematical and Computer Modelling*, pages 581–590, 2002.

[Ric56] P.I. Richards. Shock waves on the highway. *Operations Research*, pages 4:42–51, 1956.

[Ros97] S.M. Ross. *Introduction to probability models*. Academic Press, 6th edition, 1997.

[Zha98] H.M. Zhang. A theory of nonequilibrium traffic flow. *Transportation Research Part B*, pages 485–498, 1998.

[Zha03] H.M. Zhang. Driver memory, traffic viscosity and a viscous vehicular traffic flow model. *Transportation Research Part B*, pages 27–41, 2003.

Appendix A

Extensions to the PDE model

During this research some ideas arose that might be handy when modelling manufacturing lines. These ideas are discussed in this section. First some general information about the state of system is given and after that the equilibrium speed of a conwip line and the entropy of a manufacturing line is discussed.

A.1 State of a manufacturing system

In the next two sections, dealing with the conwip line and the entropy of a manufacturing line, the state of the manufacturing system plays an important role, therefore is first explained what the state of a manufacturing system is. The state of a system is defined as follows [Bor04].

Definition A.1 (State of a system)**.** The state of a system at time t is described by a vector $\bar{n}(t) = (n_1(t), n_2(t), ..., n_M(t))$ with n_m representing the number of products in the m^{th} workstation for $m = 1, 2, ..., M$. M indicates the total number of workstations in the system.

Note that the work in process is $w = \sum_{m=1}^{M} n_m$. An example of the state of manufacturing system is given below.

Example A.2 (States of simple line)**.** Consider a manufacturing line with 2 workstations in series and a constant work in process level of $w = 3$ [products]. In this system all the possible states are: [3,0] [2,1], [1,2] and [0,3].

When discussing the state of a system it is important to realize that systems can jump from one state to another. Assume that the current state of the manufacturing system is [1,2] (a line of 2 workstations). The state can change by an arrival of a product, or

the departure of a product from workstation 1 or 2. The new state then becomes: [2,2], [0,2] or [1,1] respectively.

When only exponential arrival and service rates are considered the transitions only depend on the current state and then the transitions are called a Markov chain. Two variants of the Markov chain exist; the Markov chain and the continuous time Markov chain(also known as Markov process). The definitions of Markov chains and Markov process are based on [Ros97, Kul99].

Definition A.3 (Markov chain)**.** A Markov chain is a stochastic process that takes on a finite or countable number of states. For every state i, P_{ij} indicates the probability that the process will go to state j on the next transition. This probability depends only on the current state. Transitions occur in discrete time intervals.

Definition A.4 (Markov process)**.** A Markov process is a stochastic process that takes on a finite or countable number of states. For every state i, P_{ij} indicates the probability that the process will go to state j on the next transition. This probability depends only on the current state. Transitions occur after the process has been in state i for an exponential distributed time $1/\upsilon$, where υ is the transition rate.

In a Markov chain time is no issue, it concerns only transitions and time between transitions is unknown or not taken into account. In a Markov process time between transitions is known (exponentially distributed), so a Markov process is the continuous time equivalent of a Markov chain.

For a Markov chain and a Markov process the limiting probability P_i denotes how often a state i is visited, [Ros97]. For a Markov process the stationary probability π_i denotes the fraction of time a system is in state i, [Ros97].

Definition A.5 (Limiting probability)**.** The limiting probability P_i is defined by $P_i = P_i\underline{\mathrm{P}}$ and $\sum_{i=0}^{\infty} P_i = 1$, where $\underline{\mathrm{P}}$ is a matrix containing the probabilities that a process makes a transition from state i to state j, p_{ij}.

Definition A.6 (Stationary probability)**.** The stationary probability π_i is defined by $\pi_i = \lim_{t\to\infty} P[i(t) = i]$ and $\sum_{i=0}^{\infty} \pi_i = 1$.

Now that the state of the manufacturing system and all additional terms are defined its is possible to deal with the conwip line and the entropy of a manufacturing line.

A.2 Conwip vs. push line

The test case presented in Chapter 3 is a push line. Instead of an arrival rate this line can also be controlled by maintaining a constant work in process level (conwip). The proposed PDE model for manufacturing systems should also be able to model such a conwip line, only by making some small modifications. The v^e in the relaxation term

changes, since the flow time of a conwip line differs from a push line. This section deals with the v^e of a conwip line.

For a conwip line the work in process w is known, if the throughput δ is known, the flow time can be calculated using Little's Law, [Lit61]. The throughput in a conwip line is constant and equals the production rate (μ) of a workstation minus the idle time of a workstation. This gives the following formula:

$$\delta = (1 - P[w_m = 0])\mu. \tag{A.1}$$

First an example is given how the v^e in a manufacturing line can be calculated.

Example A.7 (v^e of a conwip line)**.** A line with 2 workstations and $w = 2$. This system has three states: [2,0],[1,1] and [0,2]. The transition rates between the states are μ. The stationary probability can be easily calculated according to Definition A.6 and gives $\pi_i = 1/3$ for all three states. To determine the throughput at the end of the line, it can be seen that in one of the three states, [2,0], no output is generated. The throughput becomes: $\delta = (1-P[w_m = 0])\mu = 2/3\mu$. So the equilibrium speed becomes: $v^e = \frac{1}{\varphi} = \frac{\delta}{w} = \frac{2\mu}{3w} = 2/3$ [machine/time unit].

In (A.1) is $P[w_m = 0]$ the probability that a workstation m is idle ($w = 0$). In [Kle75], [Kle76] formulas can be found that describe a closed queueing network of workstations, which can be used to solve $P[w_m = 0]$.

The stationary probability of a state $\bar{n}$ is [Kle76]:

$$\pi(\bar{n}) = \frac{1}{G(w)} \prod_{m=1}^{M} x_m^{n_m}, \tag{A.2}$$

where x_m is defined by:

$$\mu_m x_m = \sum_{m=1}^{M} \mu_j x_j p_{mj}, \tag{A.3}$$

where p_{mj} is the probability that the system goes to state j and $G(w)$ is a normalization constant (to make sure that all the probabilities $p(\bar{n})$ add up to 1). Since $\sum_{j=1}^{N} p_{mj} = 1$ and $\mu_m = \mu$ it can be concluded from (A.3) that $x_m = x$ is a constant and therefore:

$$p(\bar{n}) = \frac{1}{G(w)} \prod_{m=1}^{M} \frac{x^{n_m}}{1} = \frac{x^w}{G(w)} = \text{constant} \tag{A.4}$$

Equation A.4 proves that the stationary probability is the same for all states. This can be explained by the fact that the states are not equally visited, but that states that

are less visited will not change that fast. For instance, a conwip line of 4 machines and $w = 5$ has the state [5,0,0,0]. This state will be visited less then state [2,1,1,1]. However, state [2,1,1,1] will transform much faster, since all workstations are full and the chance for a completion in one of the workstations is bigger than, when only one workstation is full. With the result that the stationary probability is the same for all states a general equation for the equilibrium speed in a homogeneous conwip line can be derived.

In [Kle76] the following formula for the number of possible states T can be found:

$$T = \binom{M + w - 1}{M - 1}, \tag{A.5}$$

where M is the number of machines in the line and w is the work in process. Combining (A.5) with (A.6) it follows that the following formula must hold:

$$\pi_n = \frac{1}{T} = \frac{1}{\binom{M+W-1}{M-1}}. \tag{A.6}$$

The probability that, for instance, the last machine is idle equals the number of states with the last machine empty divided by the total number of states. The number of states with the last machine empty can be calculated using (A.5) and using the fact that the products are now distributed over $M - 1$ machine, so M becomes $M - 1$, for the $P[w_m = 0]$ this results in:

$$P[w_m = 0] = \frac{\binom{M+w-2}{M-2}}{\binom{M+w-1}{W-1}} = \frac{M - 1}{M + w - 1}. \tag{A.7}$$

This results in the following formula for the equilibrium speed v^e in a conwip line:

$$v^e = \frac{\mu}{w}\left(1 - \frac{M - 1}{M + w - 1}\right) = \left(\frac{\mu}{M + w - 1}\right) \tag{A.8}$$

Checking (A.8) with Example A.7 gives the same $v^e = 2/3$ [machine/time unit]. Comparing (A.8) with (5.4), $v^e(t) = \frac{\mu}{m+w(t)}$, which is the v^e for a push line, gives that a conwip line has a higher equilibrium speed then a push line for the same w, the same M and exponential service rates.

It is left for further research to check if (A.8) can be used in the relaxation term and what results this gives. To do so (A.8) has to be scaled on the interval $x = [0, 1]$, this is left for further research.

A.3 Entropy

Entropy is a well known concept, specially known from chemistry and information theory. It appears however that entropy can be applied in a much wider scope. So is entropy be a parameter in the evolution of populations, [Dem97]. [Dem97] uses PDEs to model the dynamics of the evolution of populations. From here the idea arose to use entropy as a parameter in modelling manufacturing systems using PDEs. Entropy in manufacturing systems can be seen as a parameter that describes the possible number of states and the chance the system is in that state. It is a parameter that describes the predictability of the system, the more states a system has with a fair chance of being reached, the bigger the entropy of the system. A good example from manufacturing systems are the push line and the conwip line. Since the push line has more states that can be reached, the entropy of the push line will be bigger then the entropy of a comparable conwip line. In this section will be discussed how the entropy can be calculated in these two cases.

A.3.1 Calculating entropy

Different types of entropy, H, exist [Hil98], here the source entropy is taken since the source entropy is suited for Markov chains and processes. The source entropy can be calculated as follows:

$$-H = \sum_i \sum_j \pi_i P_{ij} \log P_{ij}. \tag{A.9}$$

The source entropy calculates for every state i ($\sum_i$) and all possible transitions P_{ij} ($\sum_j$) the $\pi_i P_{ij} log P_{ij}$ which can be seen as the likelihood that a certain transition P_{ij} occurs, taking into account the probability that the system is in the right state (π_i) for the transition. Calculating entropy in manufacturing lines starts with computing the stationary probability π_i. In [Bor04] the formulas for π_i for a heterogeneous push line, with multiple parallel machines in the workstation are given. From these formulas the following relation for the stationary probability for a homogeneous manufacturing line with only one server per workstation can be derived:

$$\pi_n = \frac{(1-u)^M u^{n_1+n_2+\cdots+n_M}}{u^M} \tag{A.10}$$

In (A.10) $n = \bar{n}(t)$ and M are used as defined in Definition A.1. With (A.10) it is possible to calculate the stationary probability for every possible state of a push line. For the conwip line the stationary probability is given in the previous section.

Now that the π_i is known, p_{ij} has to be found. This can be done by scaling the transition rates, his will be illustrated with a simple example for both a push and a conwip line.

Example A.8 (Transitions of a push line)**.** A simple homogeneous push line with 2 workstations. The state space with the transition rates look as follows:

$$\begin{array}{ccccccc}
[0,0] & \leftarrow \mu & [0,1] & \leftarrow \mu & [0,2] & \dots \\
\downarrow \lambda & \nearrow \mu & \downarrow \lambda & \nearrow \mu & \downarrow \lambda & \\
[1,0] & \leftarrow \mu & [1,1] & \leftarrow \mu & [1,2] & \dots \\
\downarrow \lambda & \nearrow \mu & \downarrow \lambda & \nearrow \mu & \downarrow \lambda & \\
[2,0] & \leftarrow \mu & [2,1] & \leftarrow \mu & [2,2] & \dots \\
\vdots & & \vdots & & \vdots & \ddots
\end{array} .$$

In the state space, μ and λ are the service and arrival rate of the push line, all the service rates are the same, since the line is homogeneous. The service and arrival rates are the transition rates from a state. For instance, state $[1,1]$, transforms to state $[0,2]$ and $[1,0]$ with rate μ and to state $[2,1]$ with rate λ. The p_{ij} here become respectively $p = \frac{\mu}{2\mu+\lambda} = \frac{1}{2+u}$ and $p = \frac{\lambda}{2\mu+\lambda} = \frac{u}{2+u}$, where $u = \frac{\lambda}{\mu}$.

Example A.9 (Transitions of a conwip line)**.** A simple homogeneous conwip line with 2 workstations. The state space with the transition rates look as follows:

$$\begin{array}{ccccc}
 & & & & [0,2] \\
 & & & \nearrow \mu & \\
 & & & \mu \swarrow & \\
 & & [1,1] & & \\
 & \nearrow \mu & & & \\
 & \mu \swarrow & & & \\
[2,0] & & & &
\end{array} ,$$

which is a diagonal from the state space of the push line. All the service rates μ are the same, since the line is homogeneous. The service rates are the transition rates from a state. For instance, state $[1,1]$, transforms to state $[0,2]$ and $[2,0]$. The p_{ij} here become twice $p = \frac{\mu}{2\mu} = \frac{1}{2}$.

Now that the stationary probabilities of the push and conwip line are known the entropy can be calculated using (A.9). A general solution is not found yet, but the entropy is here determined for two some simple examples.

Example A.10 (Calculating entropy)**.** Two simple examples with 2 workstations in line and exponential service rates of $\mu = 1$.
Example 1 is a push line with $\lambda = 1/2$ and this results in $w = 2$, $\delta = 1/2$.

Example 2 is a conwip line with $w = 2$ and this results in $\delta = 2/3$.

Entropy can be calculated using (A.9), the stationary probability π_i and the transition probabilities p_{ij}.

Entropy push line In a push line the state space is infinite. The entropy can be calculated using infinite summations. The derivation is not discussed here, but calculating the entropy for a push line of 2 machines give:

$$-H = \frac{2(1-u)}{u}\left(-\log(1+u) + \frac{u}{1+u}\log u\right) + (1-u)^2\log(1-u)\left(-\log(2+u) + \frac{u}{2+u}\log u\right) \quad \text{(A.11)}$$

This gives for example 1 an entropy of: $H = 0.52$.

Entropy conwip line The state space is here [2,0],[1,1] and [0,2] with $\pi_i = 0.25$, 0.50 and 0.25 respectively. Transition rates from [2,0] and [0,2] are both $p_{ij} = 1$ and the transition probabilities from [1,1] are $p_{ij} = 0.5$ and 0.5. This gives for the entropy: $H = -(2(0.25\ 1\ \log\ 1) + 2(0.5\ 0.5\ \log\ 0.5)) = 0.15$.

As expected the entropy the push line is higher then for the conwip line. Both lines have the same w, but the conwip line has a higher throughput. This higher throughput might be explained by the fact that the conwip line has less states. Entropy could be a useful parameter to quantify this difference.

A.4 Conclusion

More research is needed on entropy before it could be used as a parameter for manufacturing systems. The presented equations need to be validated and general formulas to calculate the entropy has to be found. However entropy could be a useful parameter and is known in many other research fields.

Appendix B

Discrete event simulation

In this appendix are presented the codes concerning the discrete event simulation. In the first section the χ code of the test case is given. The second section deals with the MATLAB® code used to gather the data of the discrete event simulation. The last two section contain MATLAB® code used to process the gathered data. All the code presented here is also available on the CD-ROM accompanying this thesis.

B.1 χ code of the test case

Of the test case a discrete event simulation is performed. This simulation is done in χ-0.8, the ascii-code of this simulation is presented below. The Exit gathers the states of the workstations each 0.5 [time unit].

```
// A manufacturing line with 10 workstations, exponential arrivals and process times is simulated.
// Homogenous processes are ramped up to a utilization of 50%.

type lot = real#nat

proc G(a: !lot, l: real) =
|[ i: nat, td: ->real
 | i:=1
 ; td:= negexp(l)
 ; *[delta sample td; a!<time,i>; i:=i+1]
]|

proc B(a: ?lot, b: !lot, c: !nat#nat, d: ?void, mach: ?void) =
|[ x: lot, xs: lot*, send: bool, m,i: nat
 | xs:= []; send:= false; m:= 0; i:=0
 ; *[ true          ; a?x               -> xs:= xs ++ [<time,x.1>]
    | len(xs) > 0   ; b!hd(xs)          -> m:= 1; xs:= tl(xs)
    | not send      ; d?                -> send:= true
    | send          ; c!<(len(xs)+m),i> -> send:= false
    | true          ; mach?             -> m:= 0; i:=i+1
    ]
]|
```

```
proc M(a: ?lot, b: !lot, mach: !void, l: real) =
|[ x: lot, td: ->real
 | td:= negexp(l)
 ; *[ a?x; delta sample td; b!x; mach!]
]|

proc ME(a: ?lot, mach: !void, l: real) =
|[ x: lot, td: ->real
 | td:= negexp(l)
 ; *[ a?x; delta sample td; mach!]
]|

proc E(c: (?nat#nat)^10, d: (!void)^10) =
|[i: nat, buf: nat*, outflux:nat*, info:nat#nat
 | i:=0; buf:=[]; outflux:=[]
 ; *[delta 0.5; !time, tab()
    ;*[i< 10 -> d.i!; c.i?info
                ; buf:= buf ++ [info.0]
                ; outflux:=outflux ++ [info.1]
                ; i:= i+1
      ]; i:= 0
    ;*[len(buf) > 0 -> !hd(buf), tab(); buf:=tl(buf)
      ]
    ;*[len(outflux) > 0  -> !hd(outflux),  tab(); outflux:= tl(outflux)
      ]
    ;!nl()
    ; buf:=[]; outflux:=[]
    ]
]|

syst S(l1,l2: real) =
|[ a: (-lot)^10, b: (-lot)^10, c: (-nat#nat)^10, d: (-void)^10, e: (-void)^10
 |  G(a.0,l1)
 || i: nat <- 0..10: B(a.i,b.i,c.i,d.i,e.i)
 || i: nat <- 0..9: M(b.i,a.(i+1),e.i,l2)
 || ME(b.9, e.9, l2)
 || E(c,d)
]|

xper = |[ S( 1.0 // interarrival time (hours/lot)
           , 1/2.0 // process time (hours/lot)
           ) ]|
```

B.2 Averaging method script

The MATLAB® script presented in this section is used to compare the averaging methods. Two methods are implemented: horizontal1 and vertical. This script is not used to process the final simulation results, note that a discrete event simulation is used that records every event, so no sampling is applied.

```
close all
clear all
hold on

%initialisatie
noe=[];
```

```
norm1=[];
norm2=[];
opp_vert=[];
opp_hor=[];
opp=[];

%-------------------------------------------------
%inladen van (een aantal van de) 10.000 experimenten
%-------------------------------------------------

for h = 1:10
    load (['wipmetingen',num2str(h)]);
    if h == 1
         wip1=wip;
         io1=io;
         time1=time;
    else
    [s1,s2] = size(wip);
    [t1,t2] = size(wip1);
      if t1 > s1
        io1 = [io1(1:s1,:) io];
        time1 = [time1(1:s1,:) time];
        wip1 = [wip1(1:s1,:) wip];
      else
        io1 = [io1 io(1:t1,:)];
        time1 = [time1 time(1:t1,:)];
        wip1 = [wip1 wip(1:t1,:)];
      end
    end
end

io=io1;
wip=wip1;
time=time1;
clear io1;
clear wip1;
clear tijd1;
%-------------------------------------------------


for q=100:100:1000   %voor een verschillend aantal experimenten

%plotten van de wip grafieken uit de dataset (maximaal 10 experimenten):

[n,m]=size(time);   % n=aantal meetpunten, m=aantal experimenten

%for p=1:min([q,10]) %aantal grafieken dat getekend gaat worden, maximaal 10.
 %    wip1=[];
 %    tijd1=[];
 %    for i=1:n-1
 %        tijd1=[tijd1;time(i,p);time(i+1,p)];
 %        wip1=[wip1;wip(i,p);wip(i,p)];
 %    end
 %    plot(tijd1,wip1,'b:')
 %end

%-------------------------------------------------------
%berekenen van de wip gemiddeldes (verticaal middelen):
%-------------------------------------------------------

%lijst maken met alle event tijdstippen:
event=[];
```

```
ion=[];
for p=1:q
    event=[event;time(:,p)];
    ion=[ion;io(:,p)];
end
a=sortrows([event,ion],[1]); %sorteren van de data
eventsvert=a(:,1);
ion=a(:,2);
clear a;
clear event;

%berekenen van de gemiddelde wip levels tussen de events
wiplevelsvert=[];
io1=ion./q;
totwip=0;
[k,l]=size(eventsvert);
for t=1:k %voor alle events
    totwip=totwip+io1(t,1);
    wiplevelsvert=[wiplevelsvert;totwip];
end

%95% betrouwbaarheidsinterval, om de rekentijd te beperken wordt slechts op een
%beperkt aantal momenten het betrouwbaarheid intervalberekend.
lowerboundv=[];
upperboundv=[];
stdev=[];

for u=1:1:max(eventsvert)
    wiplevelsvert1=wiplevelsvert(max(find(eventsvert<=u)));
    wip1=[];
    for v=1:1:m
        wip1=[wip1;wip((max(find(time(:,v)<=u))),v)];
    end
    stdev=std(wip1');
    lowerboundv=[lowerboundv;wiplevelsvert1 - ((1.96/sqrt(m))*stdev)];
    upperboundv=[upperboundv;wiplevelsvert1 + ((1.96/sqrt(m))*stdev)];
end

%plot(1:1:max(eventsvert),upperboundv,'k')
%plot(1:1:max(eventsvert),lowerboundv,'k')

%plotten van de resultaten
eventsvert=[0;eventsvert];
wiplevelsvert=[0;wiplevelsvert];

[n,o]=size(eventsvert);
wiplevels1=[];
events1=[];
for i=1:n-1
    events1=[events1;eventsvert(i,1);eventsvert(i+1,1)];
    wiplevels1=[wiplevels1;wiplevelsvert(i,1);wiplevelsvert(i,1)];
end
plot(events1,wiplevels1,'g')

%opslaan van de resultaten
%save('wipresults_vert.mat','events1','wiplevels1','lowerboundv','upperboundv')

%-------------------------------------------------------
%berekenen van de wip gemiddeldes (horzontaal middelen):
%-------------------------------------------------------

eventshor=mean(time(:,1:q)')';
```

```
wiplevelshor=cumsum(sum(io(:,1:q)./q,2));
wiplevelshortemp=(mean(wip'))'
stdev= (std(wip'))'; %is a column containing the standard deviation of each column

%95% reliability interval
upperboundh=wiplevelshor + ((1.96/sqrt(m))*stdev);
%plot(eventshor,upperboundh,'y')
lowerboundh=wiplevelshor - ((1.96/sqrt(m))*stdev);
%plot(eventshor,lowerboundh,'y')

%plotten van de average wip grafiek:
eventshor=[0;eventshor];
wiplevelshor=[0;wiplevelshor];

[n,o]=size(eventshor);
wiplevels2=[];
events2=[];
for i=1:n-1
    events2=[events2;eventshor(i,1);eventshor(i+1,1)];
    wiplevels2=[wiplevels2;wiplevelshor(i,1);wiplevelshor(i,1)];
end
%plot(events2,wiplevels2,'r')

wiplevelshortemp=[0;wiplevelshortemp];

[n,o]=size(eventshor);
wiplevels3=[];
events3=[];
for i=1:n-1
    events3=[events3;eventshor(i,1);eventshor(i+1,1)];
    wiplevels3=[wiplevels3;wiplevelshortemp(i,1);wiplevelshortemp(i,1)];
end
plot(events3,wiplevels3,'b')

%opslaan van de resultaten
%save('wipresults_hor.mat','events2','wiplevels2','lowerboundh','upperboundh')

end
```

B.3 Processing discrete event simulation results

The MATLAB® script presented in this section is used to process the final simulation results. The used averaging method is the vertical method on a sampled basis.

```
%------------------------------
%inladen van de experimenten
%------------------------------

load wip_CT_metingen ;
%------------------------------

%-----------------------------------------------------
%berekenen van de wip gemiddeldes (verticaal middelen):
%-----------------------------------------------------

%lijst met alle event tijdstippen:
events=time(:,1);
```

```
%berekenen van de gemiddelde wip levels tussen de events

wip=wip1+wip2+wip3+wip4+wip5+wip6+wip7+wip8+wip9+wip10;
wiplevels=mean(wip')';

%95% betrouwbaarheidsinterval
lowerboundw=[];
upperboundw=[];
stdevw=[];
[o,p]=size(wip);

stdevw=std((wip),0,2);
lowerboundw=[lowerboundw;wiplevels - ((1.96/sqrt(o))*stdevw)];
upperboundw=[upperboundw;wiplevels + ((1.96/sqrt(o))*stdevw)];

%plotten van de resultaten
subplot(2,1,1)
hold on
plot(events,upperboundw,'k')
plot(events,lowerboundw,'k')

events=[0;events];
wiplevels=[0;wiplevels];

[r,s]=size(events);
wiplevels1=[];
events1=[];
for i=1:r-1
    events1=[events1;events(i,1);events(i+1,1)];
    wiplevels1=[wiplevels1;wiplevels(i,1);wiplevels(i,1)];
end
plot(events1,wiplevels1,'g')
%-----------------------------------------------------

%-----------------------------------------------
%berekenen van de snelheid (verticaal middelen):
%-----------------------------------------------

%lijst met alle event tijdstippen:
events=time(:,1);

%berekenen van de gemiddelde wip levels tussen de events
S=(CT1+CT2+CT3+CT4+CT5+CT6+CT7+CT8+CT9+CT10).^-1;
speed=10*mean(S')';

%95% betrouwbaarheidsinterval
lowerboundv=[];
upperboundv=[];
stdev=[];
[o,p]=size(speed);

stdev=std((S),0,2);
lowerboundv=[lowerboundv;speed - ((1.96/sqrt(o))*stdev)];
upperboundv=[upperboundv;speed + ((1.96/sqrt(o))*stdev)];

%plotten van de resultaten
subplot(2,1,2)
hold on
plot(events,upperboundv,'k')
plot(events,lowerboundv,'k')
plot(events,speed,'g')
```

```
%-----------------------------------------------

%-------------------------
%opslaan van de resultaten:
%-------------------------
time_speed_DEM=events;
time_wip_DEM=events1;
speedDEM=speed;
wipDEM=wiplevels1;

save('wip.mat','time_speed_DEM','time_wip_DEM',
        'upperboundw','lowerboundw','upperboundv','lowerboundv','speedDEM','wipDEM')
```

Appendix C

PDE simulation

In this appendix are presented the codes concerning the PDE simulation. The full C++ code used for the numerical solution is available on CD-ROM. Here only the file containing the complete model is presented. More documentation about the C++ code can be found in [Ber04].

C.1 C++ code of the PDE model

Of the test case a discrete event simulation is performed. This simulation is done in χ-0.8, the ascii-code of this simulation is presented below.

```
#include<fstream.h>
#include<stdlib.h>
#include<math.h>
#include<string.h>
#include "basicmath.h"
#include "intel.h"

int li02(double cfl, double tstep, double & wip, double & meanv, double infx,
         int nx, int m, double * rho, double * vel, double * newrho,
         double * newvel,
     double alpha, double beta){

  double tempvel, Tau, ve, RO, mu, B;
  int i;

  tempvel=alpha/(1.0+beta*wip);
  Tau=10;
  ve=(alpha/(1.0+beta*wip)); //equilibrium velocity
  B=0.01;
  mu=2.0 * B * Tau;

  // --- calculate velocity and density ----

  // Impose Left Boundary Conditions
  newvel[0] = vel[0] - 0.5 * cfl * (sq(vel[0]) - sq(tempvel))
```

```
        //  + tstep * ((ve - vel[0]) / Tau)    // relaxation, option 1
            + tstep * (((alpha/(1.0+beta*rho[0])) - vel[0]) / Tau)
                                               // relaxation, option 2
        //  - (sq((alpha * beta)/sq(1 + beta * rho[0]))) * rho[0] * cfl * (rho[0] - infx)
                                               // anticipation
            + 2 * B * (alpha/sq(1 +beta * rho[0]))  * rho[0] * cfl * (vel[0] - tempvel)
                                               // memory
            + mu * sq((alpha * beta * rho[0])/sq(1 + beta * rho[0])) * cfl
                                               * (cfl/tstep) * (vel[0] - tempvel)
                                               // viscosity
       ;
  newrho[0] = rho[0] - cfl * (rho[0] * vel[0] - infx);

  newvel[1] = vel[1] - 0.5 * cfl * (sq(vel[1]) - sq(vel[0]))
         // + tstep * ((ve - vel[1]) / Tau)
            + tstep * (((alpha/(1.0+beta*rho[1])) - vel[1]) / Tau)
         // - (sq((alpha * beta)/sq(1 + beta * rho[1]))) * rho[1] * cfl * (rho[1] - rho[0])
            + 2 * B * (alpha/sq(1 +beta * rho[1]))  * rho[1] * cfl * (vel[1] - vel[0])
            + mu * sq((alpha * beta * rho[1])/sq(1 + beta * rho[1])) * cfl
                                               * (cfl/tstep) * (vel[1] - 2 * vel[0] + tempvel)
       ;
  newrho[1] = rho[1] - cfl*(rho[1]*vel[1]-rho[0]*vel[0]);

  // Calculate Interior Numerical Solution
  for (i=2;i<nx;i++) {
  newvel[i] = vel[i] - 0.5 * cfl * (sq(vel[i]) - sq(vel[i-1]))
         // + tstep * ((ve - vel[i]) / Tau)
            + tstep * (((alpha/(1.0+beta*rho[i])) - vel[i]) / Tau)
         // - (sq((alpha * beta)/sq(1 + beta * rho[i]))) * rho[i] * cfl * (rho[i] - rho[i-1])
            + 2 * B * (alpha/sq(1 +beta * rho[i])) * rho[i] * cfl * (vel[i] - vel[i-1])
            + mu * sq((alpha * beta * rho[i])/sq(1 + beta * rho[i])) * cfl
                                               * (cfl/tstep) * (vel[i] - 2 * vel[i-1] + vel[i-2])
       ;
  newrho[i] = rho[i] - cfl*(rho[i]*vel[i]-rho[i-1]*vel[i-1]);
  }

  wip=integrleft(newrho, 0, nx-1, 0, 1);
  meanv=integrleft(newvel, 0, nx-1, 0, 1);

  return 0;
}
```