Modeling and Control of a Manufacturing Flow Line Using Partial Differential Equations

Roel van den Berg, Erjen Lefeber, *Member, IEEE*, and Koos Rooda, *Member, IEEE*

***Abstract*—This brief deals with a control framework for manufacturing flow lines. For this framework, a continuous approximation model of the manufacturing system is required, which is computationally feasible and able to accurately describe the dynamics of the system (both throughput and flow time). Often used models, such as discrete-event models and flow models, fail to meet these specifications: the use of discrete-event models may lead to intractably large control problems, while flow models do not accurately describe the system dynamics. Therefore, we consider here a relatively new class of models for the description of manufacturing flow lines, namely partial differential equation (PDE)-models, which seems to meet the required specifications. However, for the few "manufacturing" PDE-models that have been introduced in literature so far, the accuracy has not been validated yet. In this brief, we, therefore, present a validation study on three of these PDE-models available from literature, which shows that there is a need for more accurate PDE-models. Furthermore, we propose to use one of these PDE-models for the design of a model predictive controller (MPC-controller), which is to be applied in closed loop with a discrete-event manufacturing flow line. For two considered tracking problems, the resulting MPC-controller is shown to outperform a classical push strategy.**

***Index Terms*—Control, discrete-event system, manufacturing flow line, model predictive control (MPC), partial differential equation (PDE), production control, production systems, tracking.**

I. Introduction

THE CONTROL of discrete-event manufacturing systems has been a field of interest for several decades. Starting from simple push and pull strategies (such as material requirements planning (MRP), enterprise resources planning (ERP), and just-in-time (JIT), see, e.g., [1]), more and more advanced control techniques have been introduced. The supervisory control theory was introduced in 1987 by Ramadge and Wonham [2], [3]. This theory, which is based on a discrete-event description of the manufacturing system, has been extended up to now by several authors, see, e.g., [4]–[8]. However, when it comes to the control of large manufacturing systems (or networks of such systems), supervisory control is not very suitable due to the high level of detail they deal with, which causes the corresponding control problem to grow intractably large. Control strategies for manufacturing systems using so-called flow models (see, e.g., [9]–[12]), on the other hand, do not suffer from this problem because they deal with a smaller level of detail, i.e., long-term time scales and average product flows. Although these flow models can be controlled in various ways, they do not give an accurate description of the dynamics of a manufacturing system. They are only throughput oriented, whereas flow time (also referred to as cycle time or throughput time) also plays an important role in the manufacturing industry. The relation between flow time and throughput is absent in the flow models, which causes unnatural behavior. For example, if lots are fed into an initially empty manufacturing system, according to flow models lots will immediately leave the system, whereas in practice at least some delay (production time) is present between the entering and leaving of lots.

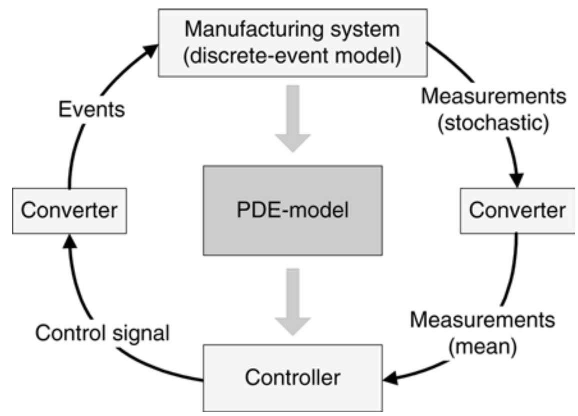

Fig. 1. Control framework for a discrete-event manufacturing system.

It is only recently that the class of partial differential equation (PDE)-models has been introduced for the description of manufacturing systems [13]–[15]. These models, which describe the average flow of products through a manufacturing line as a 1-D continuous flow, can provide flow time data in addition to throughput data, for both transient and steady state, which gives them an advantage over the earlier mentioned flow models. Up to now, however, the PDE-models have not been validated with discrete-event systems and, therefore, the accuracy of throughput and flow time data has not yet been proven.

The contribution of this brief is two-fold. In the first place, we present three currently available "manufacturing" PDE-models and validate them using a discrete-event model, in order to investigate the accuracy of this type of model for the description of manufacturing systems. In the second place, we use one of these PDE-models for the design of a controller that can be applied in closed loop with the discrete-event manufacturing system. Hereto, we make use of a control framework [16] (see Fig. 1), which is similar to the hierarchical control framework presented in [9]. In the framework, (see Fig. 1) the desired manufacturing system is represented by a discrete-event model. For the design of the controller for this discrete-event model, a continuous approximation model is used in the framework. In our case, this

Manuscript received June 10, 2005; revised May 25, 2006. Manuscript received in final form January 12, 2007. Recommended by Associate Editor G. Rosen.

The authors are with the Systems Engineering Group, Department of Mechanical Engineering, Technische Universiteit Eindhoven, 5600 MB, Eindhoven, The Netherlands (e-mail: r.a.v.d.berg@tue.nl; a.a.j.lefeber@tue.nl; j.e.rooda@tue.nl).

Digital Object Identifier 10.1109/TCST.2007.903085

will be a PDE-model. Finally, the framework contains two conversion blocks for the connection between the controller and the discrete-event model. One conversion block translates the control signal into input events for the discrete-event system, the other conversion block translates output data from the system into suitable input for the controller. In order to validate the PDE-models and investigate the performance of the presented control framework, a simple manufacturing flow line has been studied in this research. Although various other types of manufacturing systems occur in practice, this simple flow line is an interesting starting point to gain insight into the applicability of PDE-models in the modeling and control of manufacturing systems.

The remainder of this brief is organized as follows. In Section II, we start with the presentation and validation of three currently available PDE-models used for the description of a manufacturing flow line. Then, in Section III, we use one of these PDE-models to design a controller. After testing this controller on the PDE-model, we describe the design of the signal conversions that are required to connect the controller to the discrete-event system. In Section IV, we consider the closed-loop controlled discrete-event system and evaluate its performance by means of simulation. Section V concludes this brief.

II. Modeling and Validation

In this section, first, three PDE-models available for describing a manufacturing flow line are presented. Then, the experiments used for validation of these PDE-models are discussed. Finally, the results of the validation study are presented and evaluated.

A. PDE-Models

The three considered PDE-models used for the description of a manufacturing flow line are all based on mass conservation. For a manufacturing line, this implies that every lot that enters the line should eventually come out. In order to describe the progress of lots through the line, the variable x is defined as the position (or degree of completion) of a lot, with $x = 0$ denoting the entrance of raw materials in the line and $x = 1$ the exit of finished goods. Now, density ρ (number of lots per unit of x), flux q (number of lots that pass by per unit of time), and velocity $v(\Delta x$ per unit of time) are defined as variables that vary with time t and position x, so that the mass conservation law can be written as

$$\frac{\partial \rho(x,t)}{\partial t} + \frac{\partial q(x,t)}{\partial x} = 0 \tag{1}$$

in which the flux q is given by

$$q(x,t) = \rho(x,t)v(x,t). \tag{2}$$

The total work-in-process (WIP) w in a manufacturing line is related to the density according to $w(t) = \int_0^1 \rho(x,t)dx$.

In addition to the mass conservation law, the models include an equation that describes the relation between the velocity of a lot and the density of lots in the line, which is different for each of the considered models. These relations, however, all have in common that they are based on the steady-state relations of an $M/M/1$ queuing system (see, e.g., [17]), which means that, for an exponentially distributed manufacturing flow line in steady state, the PDE-models give an exact description of the throughput and flow time. Therefore, during the validation study, we mainly focus on the transient phase. Before we come to this point, however, first the three PDE-models are presented.

Model 1 (Armbruster et al. [14]): This model consists of (1) and (2) and a relation defining the velocity as a function of the total WIP, the workstation's process rate μ and the number of identical workstation in series m

$$v(t) = \frac{\mu/m}{1 + \frac{1}{m}w(t)} = \frac{\mu}{m + w(t)}. \tag{3}$$

This relation is constructed as follows. In steady state, the WIP is homogeneously distributed over the line, i.e., in front of each workstation there are on average w/m lots. A new lot that arrives at a workstation thus is expected to leave this workstation after $(1 + w/m)/\mu$ units of time. For the line with m workstations the expected flow time is, therefore, $(1 + w/m)m/\mu$, and the velocity $\mu/(m + w)$. Note that for this model the velocity does not depend on x and thus is uniform over the whole manufacturing line for all t.

Model 2 (Armbruster et al. [14]): This model consists of (1) and (2), but here the development of the velocity in position and time is described by a second PDE, representing the momentum conservation law without diffusion

$$\frac{\partial v(x,t)}{\partial t} + \frac{1}{2}\frac{\partial v(x,t)^2}{\partial x} = 0. \tag{4}$$

Furthermore, the boundary condition for (4) is defined as

$$v(x = 0, t) = \frac{\mu/m}{1 + \frac{1}{m}w(t)} = \frac{\mu}{m + w(t)}. \tag{5}$$

This boundary condition is chosen in such a way that it satisfies the $M/M/1$ queuing relations if the system is in steady state.

Model 3 (Lefeber [15]): This model is composed of (1) and (2) and a velocity relation that depends on the density

$$v(x,t) = \frac{\mu}{m + \rho(x,t)}. \tag{6}$$

This velocity-density relation is similar to (3), only here the velocity of a lot depends on the *local* density of lots instead of the total WIP in the line.

B. Validation Experiments

For the presented PDE-models, we performed a validation study using a discrete-event model as the validation model. The goal of this validation study was to examine the accuracy of the throughput and flow time data (in transient and steady state) generated by the three considered PDE-models. In this subsection, we describe the various aspects of the validation process. The results of the validation study are presented in Section II-C.

The manufacturing system we considered for the validation study consists of ten identical workstations in series, each workstation consisting of a buffer and a machine. The buffer has infinite capacity and uses a first-in first-out (FIFO) policy. In the machine, lots are processed one by one with a process time which is exponentially distributed with a mean of $1/\mu = 0.5$ h. Furthermore, lots are fed into the system according to a Poisson process with an arrival rate λ.

Using the described manufacturing line, we performed two types of experiments in order to validate the PDE-models: ramp up and ramp down experiments. During a ramp up experiment, lots are fed into the initially empty line using a fixed arrival rate λ. Once the line reaches steady state (i.e., when the mean flow time and throughput deviate less than 1% from the steady-state value) the experiment is ended. We performed four different ramp up experiments, using, respectively, the arrival rates: $\lambda = 0.5$, $\lambda = 1.0$, $\lambda = 1.5$, and $\lambda = 1.9$ lots/h. For the ramp down experiments, we assumed the system to initially be in steady state. Starting from this initial state, the arrival rate is instantaneously decreased to $\lambda = 0.5$. Once steady state is reached, the experiment is ended. We performed four different ramp down experiments, using different initial steady states, corresponding, respectively, to the arrival rates: $\lambda = 1.0$, $\lambda = 1.5$, $\lambda = 1.8$, and $\lambda = 1.9$ lots/h.

During the validation study, we used a discrete-event model of the manufacturing flow line as the validation model, since this is an accurate and well-accepted modeling technique in the analysis of manufacturing systems. The discrete-event model was constructed using χ, a specification language developed at the Eindhoven University of Technology [18].

The PDE-models have been solved numerically using the zeroth-order Godunov method (see, e.g., [19]). In order to assure stability of the solutions, a Courant number of ≤ 0.5 is used at all times. Furthermore, the initial conditions for the PDE-models are given by $\rho(x) = 0, v(x) = \mu/m$ for the ramp up experiments and $\rho(x) = m\lambda/(\mu - \lambda), v(x) = \mu/(m + \rho(x))$ for the ramp down experiments. Finally, for all PDE-models the left boundary condition $q(x = 0, t) = \lambda(t)$ and the right boundary is assumed to be free, i.e., once a lot reaches the end of the line $(x = 1)$, it immediately leaves the system.

Before the results of the PDE-models can be compared to those of the validation model, the outputs of both models need to be processed. Starting with the discrete-event model, this model generates throughput and flow time data, but these data are highly stochastic. In order to obtain a mean throughput and a mean flow time as functions of time, we performed multiple simulations with the discrete-event model for each instantiation of the experiments, and then averaged the throughput and flow time over these simulations. To guarantee a two-sided 95% confidence interval with a maximum relative width of 0.005 for all t on the mean throughput and mean flow time, we performed 1 000 000 independent simulations for each instantiation of the experiments. For the PDE-models, on the other hand, the available flux data should be transformed into throughput and flow time data: the mean throughput equals the outflux $q(x = 1, t)$ and the mean flow time is obtained by calculating the delay between the integrated influx $\int q(x = 0, t)dt$ and integrated outflux $\int q(x = 1, t)dt$.

C. Validation Results

Since the results of the validation experiments described in the previous subsection are too numerous to be presented completely, only one ramp up and one ramp down experiment are evaluated here. For the other results, which are similar to the ones presented here, the reader is referred to [20].

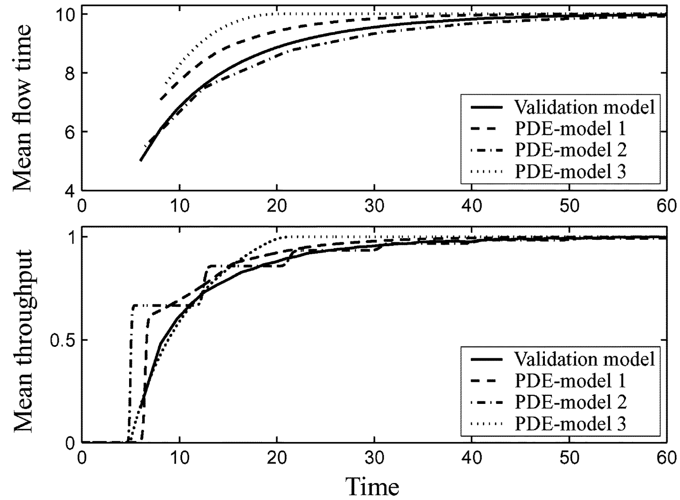

Fig. 2. Validation results for ramp up ($\lambda = 1.0$).

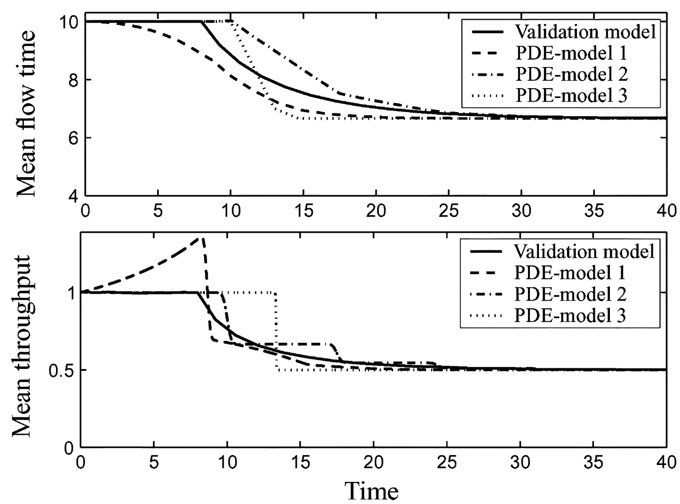

Fig. 3. Validation results for ramp down from $\lambda = 1.0$ to $\lambda = 0.5$.

In Figs. 2 and 3, the ramp up and ramp down developments of the flow time and throughput are plotted as a function of time for PDE-model 1 (dashed line), PDE-model 2 (dashed-dotted line), PDE-model 3 (dotted line), and the validation model (solid line). Figs. 2 and 3 show that once the steady state has been reached, the results of the PDE-models correspond to those of the validation model. This was to be expected since the PDE-models are based on exact steady-state relations from the queuing theory. In the transient phase, however, the PDE-models show significant deviation from the validation model.

For the ramp up experiment, all the PDE-models overestimate the flow time of the first lot (the first point of the curve), which implies that lots are moving too slowly through the line in the first part of the transient phase. Furthermore, the throughput predicted by the PDE-models increases too soon. Steady state (in flow time and throughput) is reached too soon by PDE-models 1 and 3, and too late by PDE-model 2.

For the ramp down experiment, the flow time and throughput of the system decrease too late for PDE-models 2 and 3, and

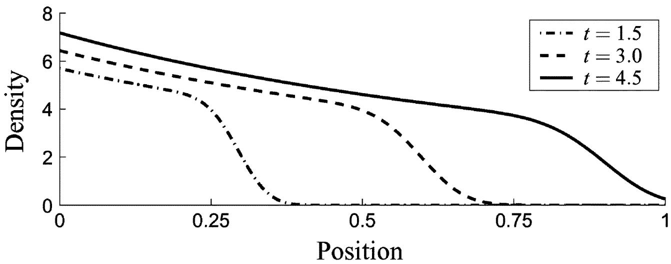

Fig. 4. PDE-model 2: ramp up evolution of density for $\lambda = 1.0$.

the flow time decreases too soon for PDE-model 1. Moreover, steady state is reached too soon for all PDE-models.

Another observation in the ramp down results is that for PDE-model 1 the throughput curve initially increases before it decreases to the steady-state value. This is caused by the model property which dictates that the velocity only depends on the total WIP in the system and, therefore, is uniform for all x. Because of this property, lots near the exit of the system can be influenced by lots which have just been fed into the system. This influence is unnatural and, therefore, undesired. The increase of throughput can be explained as follows: due to a decreased arrival rate, the total WIP in the system decreases, which causes the velocity to increase, see (3). However, the local density at the exit of the system still equals the steady-state density and, therefore, through (2), an increase in throughput results. For a ramp up experiment with PDE-model 1, the throughput would have decreased before increasing to the steady-state value, however, since the throughput is already zero this phenomenon cannot be seen in Fig. 2. For PDE-model 2, a similar incorrectness in the definition of the velocity can be identified: since $v(x = 0)$ only depends on the total WIP in the system, the velocity at the entrance of the system is incorrect whenever $\rho(x = 0)$ is not equal to the mean density in the system. If, for example, all lots in the system are located at the final two workstations, then the velocity of a new entering lot should be maximal since the first workstation is empty. However, since the total WIP in the system is larger than zero, $v(x = 0)$ is smaller than the maximum. PDE-model 3 does not incorporate such an incorrectness in the velocity definition since in this model the velocity is related to the local density.

Finally, we observe that the throughput curves of PDE-models 1 and especially 2 are not smooth. In order to explain this observation, consider PDE-model 2, for which the velocity at $x = 0$ depends on the total WIP in the system. Initially, the WIP increases linearly in time, since the influx is constant and the outflux is zero. As a result $v(x = 0)$ decreases nonlinearly in time, see (5). The lots move as a "wave" profile through the line, as visualized in Fig. 4. Once the front of this "wave" reaches $x = 1$, the outflux suddenly increases significantly, which causes the total WIP to increase much slower, and thus $v(x = 0)$ to decrease much slower. This results in a sudden change in the derivative of the flux around $x = 0$ which evolves through the line as time elapses and causes a new "wave" profile in the density distribution over x. At the moment this "wave" reaches $x = 1$, the outflux again increases significantly, etcetera. For PDE-model 1 the nonsmoothness is caused by the same effect. However, since for this model the velocity is uniform in the whole line, the sudden change in the derivative of the flux occurs in the whole line and not only around $x = 0$, which results in a smoother transition.

Obviously, more accurate PDE-models are required for the simulation of manufacturing systems. For now, we use PDE-model 3, aware of its shortcomings, in the remainder of this brief for the design of a controller. PDE-models 1 and 2 are not considered further here because of the properties as described before.

III. Controller Design

Now that we have selected PDE-model 3 for the continuous approximation model, the next step in the control framework is the design of a controller based on this model. In this section, first the studied control problem is presented. Then, model predictive control (MPC) is introduced to solve this problem. The MPC-controller is designed and subsequently tested on the PDE-model. Finally, we describe how the developed controller is connected to the discrete-event model of the manufacturing flow line.

A. Control Problem Description

The goal of the controller is to make the throughput of the manufacturing system follow a certain reference demand as accurately as possible. Thereby, the throughput should be corrected for possibly arising production shortages/surpluses, using the arrival rate as the only controllable input. In order to verify whether the controller is capable of reaching this goal, a test problem was sought. The simplest related control problem in the production management of a manufacturing system is to design the input as a function of time such that the output (outflux) of the system moves from one steady state throughput δ_1, to another steady state throughput δ_2, without creating a permanent backlog, i.e., once the new steady state throughput is reached, the backlog should be zero.

For the design of the controller, we used this control problem as a test problem. Here, we chose the reference demand to be

$$\delta(t) = \begin{cases} \delta_1 = 0 \text{ lots per hour}, & \text{for } t < 0 \\ \delta_2 = 1.0 \text{ lots per hour}, & \text{for } t \geq 0. \end{cases}$$

B. MPC

Due to the nonlinearity of the PDE-model and the time-delay that is present between a control input and its effect on the throughput, it is hard to find a suitable control method for the defined control problem. MPC (see, e.g., [21] and [22]), however, is able to deal with such problems, and is, therefore, used for the control of the PDE-model. Earlier applications of MPC in manufacturing systems can be found in [9] and references therein.

Before MPC can be applied to the PDE-model, first an internal prediction model and a goal function of MPC need to be defined. The internal model should be a good approximation of the PDE-model so that the predictions are accurate. The goal function, which is to be minimized, should represent the control

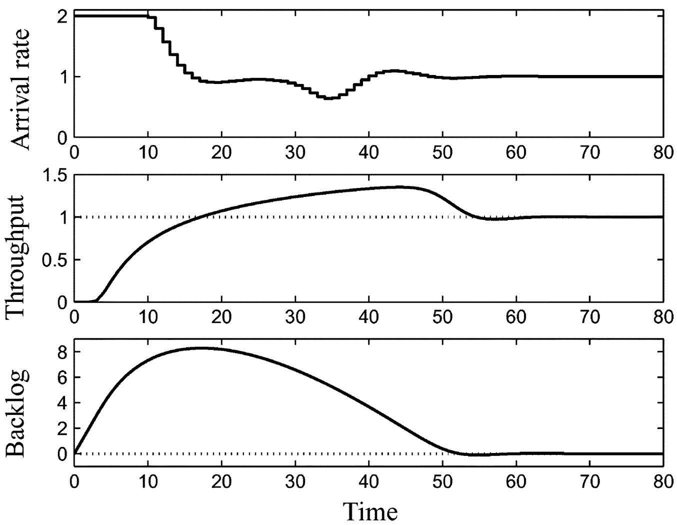

Fig. 5. Simulation results of MPC controller.

goal, i.e., tracking the defined reference demand as accurately as possible.

As internal prediction model, we used the PDE-model (1), (2), and (6), which was again solved using the zeroth-order Godunov method and a Courant number ≤ 0.5.

The goal function of MPC is defined as the normalized deviation between realized and reference output, which should be minimized. For the control problem defined in Section III-A, the goal function at sample k is

$$J_k(\mathbf{u}) = \sum_{i=1}^{p} b(k+i)^2 \tag{7}$$

in which $\mathbf{u}$ is the vector of current and future arrival rates. The length of this input vector equals the number of samples in the control horizon c. Parameter p represents the number of samples in the prediction horizon, k is the current sample, and $b(k+i)$ is the predicted backlog at sample $k+i$. Note that since the backlog is defined as

$$b(k+1) = b(k) + (\delta_2 - q(x=1,k))\Delta t^* \tag{8}$$

with Δt^* the sample time used in the goal function, there is no need to explicitly include the throughput and demand in the goal function: once the backlog remains zero, the throughput equals the reference demand. In order to reduce computation time, we here choose the sample time Δt^* as a multiple of the internal sampling rate Δt, i.e., the goal function (7) only takes into account a fraction of all sample points computed by the internal model.

Using the defined internal model and goal function, MPC has been applied to the PDE-model to solve the test problem described in Section III-A. For the implementation of MPC, we used $\Delta t = 0.25$ h, $\Delta t^* = 1.0$ h, $p = 100$ samples, and $c = 8$ samples. Furthermore, we constrained the influx to the interval $0 \leq \lambda \leq 2$. In Fig. 5, the simulation results of the MPC-controlled PDE-model are visualized. As these results show, stabilization at the new steady state throughput using the input determined by MPC takes a little longer than it does for a constant arrival rate (see Fig. 2, PDE-model 3). However, whereas application of the constant arrival rate results in a positive backlog, application of the input determined by MPC eventually results in a backlog of zero. We, therefore, conclude that the applied implementation of MPC is suitable for the control of the PDE-model.

C. Connecting the Controller to the Discrete-Event Model

The proposed controller is connected to the discrete-event model of the manufacturing flow line using two signal conversion blocks (see Fig. 1). The discrete time control signal has to be converted into events for the discrete-event model. Furthermore, the output of the discrete-event model has to be converted into a suitable feedback signal for the controller.

The control signal conversion is kept simple so that it affects the control signal as little as possible: using the computed arrival rate of the current sample, the amount of lots that should be started in this sample is determined. This amount (rounded to an integer) is inserted all at once in the buffer of the first workstation in the line. This procedure is repeated for the following samples. A good property of this conversion is that the number of started lots per sample is close to the intended amount. A disadvantage, however, is that the arrival of lots in the manufacturing system no longer follows a Poisson process, whereas this was assumed in the design of the internal prediction model of the MPC controller. More research is required to find a more suitable conversion.

For the conversion of the output of the discrete-event model into a suitable feedback signal for the controller, we used a first-order observer

$$d(k+1) = d(k) + K_d(y_{\text{meas}}(k) - y_{\text{mod}} - d(k))$$

in which d is a disturbance parameter, y_{meas} is the measured output, y_{mod} is the output according to the internal model, and K_{d} is the observer gain. The disturbance parameter is used as a correction to the internal model. The predictions of the corrected model y_{corr}

$$y_{\text{corr}}(k) = y_{\text{mod}}(k) + d(k)$$

are used for the computation of the goal function value. Here, the output y represents the total amount of finished lots.

Note that the disturbance parameter d corrects for deviations between predictions and measurements independently of the cause of this deviation. Therefore, this parameter can be used for both the filtering of stochastic noise in the measurement and the correction of errors in the internal model. This means that the MPC controller should work equally well with an imperfect internal model of the manufacturing line, such as PDE-model 3. However, since the observer is first-order, it can only correct the output of the internal model for constant (or slowly varying) deviations with the discrete-event model.

IV. Performance Evaluation

With the MPC controller and conversions blocks described in Section III, the control framework (see Fig. 1) is complete. For the resulting closed-loop controlled manufacturing flow line the performance was evaluated by means of simulation: for two tracking problems the performance of the MPC-controlled flow line was compared to the performance of the flow line controlled by a classical push strategy. In this section, first, the two tracking problems are presented. Then, the implementation of the MPC controller and the push strategy is described. Finally, the simulation results are presented and discussed.

A. Two Tracking Problems

The first tracking problem deals with a slowly varying demand which requires a low utilization of the system, whereas the second tracking problem deals with a fast varying demand which requires high utilization of the system. The throughput demand for the first problem follows a sine-function with a mean of 1.0 lots/h, an amplitude of 0.5 lots/h, and a frequency of 1/400 cycles/h, so that the cumulated reference production as a function of time is given by

$$y_{\mathrm{ref}}(t) = 1.0t + \frac{100}{\pi}\left(1 - \cos\left(\frac{\pi}{200}t\right)\right).$$

For the second problem, the throughput demand follows a sine-function with a mean of 1.5 lots/h, an amplitude of 0.4 lots/h, and a frequency of 1/80 cycles/h, which results in a cumulated reference production of

$$y_{\mathrm{ref}}(t) = 1.5t + \frac{16}{\pi}\left(1 - \cos\left(\frac{\pi}{40}t\right)\right).$$

Here, the cumulated production is used as the tracking parameter instead of the throughput and backlog which were used in Section III-A. The cumulated production is preferred because this quantity is easier to measure in the discrete-event model and because it is less sensitive to variability.

B. Implementation

The defined tracking objectives are implemented in the MPC controller (7) by choosing $b(k+i) = y_{\mathrm{corr}}(k+i) - y_{\mathrm{ref}}(k+i)$. For the first tracking problem, we used $c = 5$ samples, $p = 15$ samples, and $\Delta t^* = 10$ h. The second problem requires a more "aggressive" control since the demand fluctuates faster; therefore, we chose $c = 3$ samples, $p = 5$ samples, and $\Delta t^* = 10$ h. For both the tracking problems, we used an observer gain $K_d = 0.0001$ and a sample time of the observer of 0.1 h. The other settings of the MPC controller equal those described in Section III-B.

Using the push strategy, the number of lots $x(k)$ that is released into the flow line at sample k is determined by

$$x(k) = \max\left[y_{\mathrm{ref}}(k \cdot \Delta t^* + \hat{\varphi}(k)) - y(k) - w(k), 0\right].$$

This equation can be explained as follows: lots that are released at sample k will be finished (approximately) at time $t = k \cdot \Delta t^* + \hat{\varphi}(k)$, with $\hat{\varphi}(k)$ the expected flow time at sample k. In order to reach the cumulated reference production $y_{\mathrm{ref}}(k \cdot \Delta t^* + \hat{\varphi}(k))$, this amount minus what was already finished (y) and what is still in the system (w) should be released into the system. Since a negative number of lots cannot be fed into the system, $x(k)$ must be nonnegative. The computed amount of lots $x(k)$ (rounded to an integer) is inserted all at once in the buffer of the first workstation.

For the two tracking problems, we used a sample time Δt^* of 10 h. The expected flow time $\hat{\varphi}$ was determined by taking the average of the flow times of the ten most recently finished lots. For $y(k) \leq 10$, the average was determined over all the finished lots. For $y(k) = 0$, the expected flow time equaled the sum of all process times, i.e., $m \cdot 1/\mu = 5$ h.

TABLE I
SIMULATION RESULTS: OVERALL ERROR E

	Problem 1		Problem 2	
	MPC	Push	MPC	Push
Mean E (lots)	$1.6 \cdot 10^3$	$6.7 \cdot 10^3$	$1.3 \cdot 10^4$	$2.8 \cdot 10^4$
Std on E (lots)	$6.0 \cdot 10^2$	$2.0 \cdot 10^3$	$3.9 \cdot 10^3$	$8.1 \cdot 10^3$

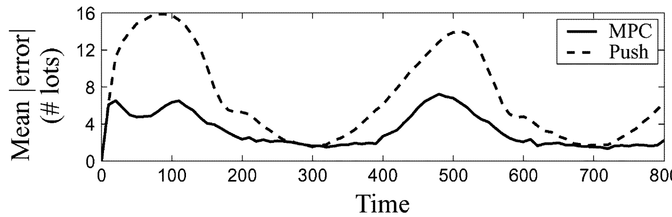

Fig. 6. Simulation results of the first tracking problem.

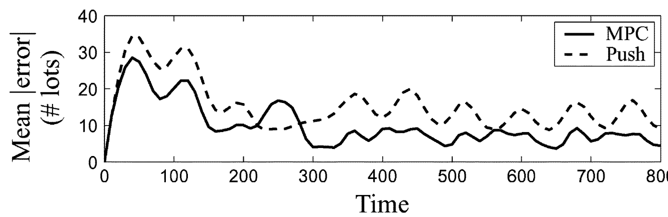

Fig. 7. Simulation results of the second tracking problem.

C. Simulation Results

For the two tracking problems simulations were performed with a run length of 800 h. In order to quantify the performance of the controllers, we defined the overall error E

$$E = \sum_t e(t)^2, \qquad \text{for } t = 0, 10, 20, \ldots, 800 \tag{9}$$

in which $e(t) = y(t) - y_{\mathrm{ref}}(t)$ is the momentary error at time t and $y(t)$ is the cumulated production at time t as realized by the system. To guarantee a 95% two-sided confidence interval on the mean E with a relative width of less than 0.05, we performed 300 independent simulation runs with the MPC-controlled system and with the push system for both the tracking problems. The mean E and standard deviation on E resulting from these simulations are listed in Table I. Furthermore, in Figs. 6 and 7 the mean absolute error $|e(t)|$ is plotted for both tracking problems. From these results, it can be concluded that the MPC-controlled system not only follows the reference signal closer than the push system for both tracking problems, but it also provides a more reliable output since the standard deviation on E is smaller. Clearly, the MPC controller with internal PDE-model has, despite the shortcomings of the PDE-model, better insight in the dynamics of the discrete-event system and, therefore, can more accurately follow the reference trajectory and reduce the output variability caused by the stochastic nature of the system.

V. Conclusion

In this brief, we discussed a relatively new class of models, namely PDE-models, used for simulation and control of a manufacturing flow line. In contrast to other simulation models, PDE-models are both computationally tractable and capable of describing a manufacturing line's behavior (throughput and flow time) in both transient and steady state, which makes them particularly suitable for manufacturing control purposes. The validation study we performed on three currently available PDE-models, however, shows that more accurate PDE-models are required.

Subsequently, we used one of the studied PDE-models for the design of an MPC-controller for the discrete-event manufacturing system. The resulting MPC-controller was successfully connected to the discrete-event system using a control signal conversion and an output observer.

Finally, numerical experiments were performed to evaluate the performance of the MPC-controlled manufacturing flow line in comparison with the performance of the flow line controlled by a classical push strategy. Results indicate that, for the defined tracking problems, the MPC-controlled flow line not only follows the reference signal more accurately than the push system, but it also has a more reliable output since the output variability is smaller.

Although these first findings are promising, more research is required to reveal the true value of PDE-models and the presented control framework in the field of manufacturing. For example, in this research, we only studied an $M/M/1$ flow line, whereas other manufacturing systems, such as $G/G/m/N$ systems, or systems with reentrant behavior also require investigation.

References

[1] W. Hopp and M. Spearman, *Factory Physics: Foundations of Manufacturing Management*, 2nd ed. Singapore: McGraw-Hill, 2000.

[2] P. Ramadge and W. Wonham, "Supervisory control of a class of discrete-event systems," *SIAM J. Control Opt.*, vol. 25, pp. 206–230, 1987.

[3] P. Ramadge and W. Wonham, "The control of discrete event systems," *IEEE Proc.*, vol. 77, no. 1, pp. 81–98, 1989.

[4] S. Bogdan, F. Lewis, Z. Kovacic, A. Gurel, and M. Stajdohar, "An implementation of the matrix-based supervisory controller of flexible manufacturing systems," *IEEE Trans. Control Syst. Technol.*, vol. 10, no. 5, pp. 709–716, Sep. 2002.

[5] M. Heymann, F. Lin, and G. Meyer, "Multiuser discrete-event control with active events," *IEEE Trans. Autom. Control*, vol. 47, no. 2, pp. 314–318, Feb. 2002.

[6] F. Charbonnier, H. Alla, and R. David, "The supervised control of discrete-event dynamic systems," *IEEE Trans. Control Syst. Technol.*, vol. 7, no. 2, pp. 175–187, Mar. 1999.

[7] B. Brandin and W. Wonham, "Supervisory control of timed discrete-event systems," *IEEE Trans. Autom. Control*, vol. 39, no. 2, pp. 329–342, Feb. 1994.

[8] T. Ushio, "A necessary and sufficient condition for the existence of finite state supervisors in discrete-event systems," *IEEE Trans. Autom. Control*, vol. 38, no. 1, pp. 135–138, Jan. 1993.

[9] F. Vargas-Villamil, D. Rivera, and K. Kempf, "A hierarchical approach to production control of reentrant semiconductor manufacturing lines," *IEEE Trans. Control Syst. Technol.*, vol. 11, no. 4, pp. 578–587, Jul. 2003.

[10] C. Cassandras, Y. Wardi, B. Melamed, G. Sun, and C. Panayiotou, "Perturbation analysis for on-line control and optimization of stochastic fluid models," *IEEE Trans. Autom. Control*, vol. 47, no. 8, pp. 1234–1248, Aug. 2002.

[11] J. Harrison, *Brownian Motion and Stochastic Flow Systems*. New York: Wiley, 1995.

[12] J. Kimemia and S. Gershwin, "An algorithm for the computer control of a flexible manufacturing system," *IIE Trans*, vol. 15, no. 4, pp. 353–362, 1983.

[13] D. Armbruster, D. Marthaler, C. Ringhofer, K. Kempf, and T.-C. Jo, "A continuum model for a re-entrant factory," *Oper. Res.*, vol. 54, no. 5, pp. 933–950, 2006.

[14] D. Armbruster, D. Marthaler, and C. Ringhofer, "Kinetic and fluid model hierarchies for supply chains," *SIAM J. Multiscale Model. Simulation*, vol. 2, no. 1, pp. 43–61, 2004.

[15] E. Lefeber, "Nonlinear models for control of manufacturing systems," in *Nonlinear Dynamics of Production Systems*, G. Radons and R. Neugebauer, Eds. Berlin, Germany: Wiley, 2004, ch. 5, pp. 69–81.

[16] E. Lefeber, R. van den Berg, and J. Rooda, "Modelling manufacturing systems for control: A validation study," in *Networks of Interacting Machines: Production Organization in Complex Industrial Systems and Biological Cells*, D. Armbruster, K. Kaneko, and A. Mikhailov, Eds. Singapore: World Scientific, 2005, ch. 4, pp. 101–126, .

[17] D. Gross and C. Harris, *Fundamentals of Queueing Theory*. New York: Wiley, 1985.

[18] D. van Beek, K. Man, M. Reniers, J. Rooda, and R. Schiffelers, "Syntax and consistent equation semantics of hybrid chi," *J. Logic Algebraic Program.*, vol. 68, no. 1-2, pp. 129–210, 2006.

[19] R. Leveque, *Finite Volume Methods for Hyperbolic Problems*. Cambridge, U.K.: Cambridge Univ. Press, 2002.

[20] R. van den Berg, "Partial differential equations in modelling and control of manufacturing systems," M.S. thesis, Eindhoven Univ. Technol., Eindhoven, The Netherlands, 2004 [Online]. Available: http://se.wtb.tue.nl/~lefeber/MscTheses/vandenBerg.pdf.

[21] F. Allgöwer and A. Zheng, *Nonlinear Model Predictive Control, Progress in Systems and Control Theory*. Basel, Switzerland: Birkhäuser, 2000.

[22] J. Maciejowski, *Predictive Control with Constraints*. Englewood Cliffs, NJ: Prentice-Hall, 2002.