Proceedings of the
44th IEEE Conference on Decision and Control, and
the European Control Conference 2005
Seville, Spain, December 12-15, 2005

Control of manufacturing systems using state feedback and linear programming

J.A.W.M. van Eekelen, E. Lefeber, B.J.P. Roset, H. Nijmeijer and J.E. Rooda

Abstract— **Most studies on control of discrete event manufacturing systems focus on control in the event domain. However, in real-life production environments, events occur while time elapses. In this study we develop an explicit state feedback controller in a model predictive control (MPC) setting for the class of manufacturing systems where only communication and no choice occurs. The state of the manufacturing system is specified as a function of time instead of events. For larger systems, where explicit feedback control is too difficult or time consuming, we present an MPC control framework based on repeatedly solving linear programming problems.**

I. INTRODUCTION

Due to increasing industrial complexity and the costs involved, the need for manufacturing control strategies becomes stronger.

A certain class of manufacturing systems can be characterized by discrete event (DE) systems. In this study we consider manufacturing systems in which only communication and no choice occurs, so all product recipes, routes and orders are predetermined. However, this does not exclude variability (e.g. on process times). These systems can be described using max-plus algebra for example (see [1]).

Several proposals have been made to control discrete event manufacturing systems of the class described above. A rough division can be made between discrete event models and continuous approximation models (see [6], [11]). In this research, focus will be on the former. In Section III a short overview is given of DE models and DE control of manufacturing systems.

Goal of this study is to control systems from the considered class of manufacturing systems in a dynamical way: generate and update a manufacturing schedule continuously based on state measurements. We try to overcome some of the problems that arise when any of the currently available techniques is used. We first present an example of a manufacturing system, and explain the choice of state, inputs and control objective (Section II). A state feedback law is proposed which specifies the manufacturing schedule continuously over time based on state measurements. For larger problems, determining a state feedback law might become too difficult or too time consuming. The scheduling problem can then be stated as a linear programming problem in an intuitive way. This LP method can be used in a model predictive control approach to generate manufacturing schedules with a receding horizon strategy based on measurements.

All authors are with the Department of Mechanical Engineering, Technische Universiteit Eindhoven, The Netherlands,
[j.a.w.m.v.eekelen,a.a.j.lefeber,b.j.p.roset, h.nijmeijer,j.e.rooda]@tue.nl

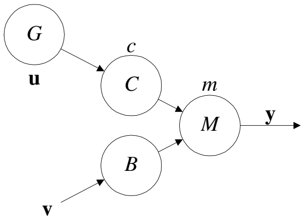

Fig. 1. Manufacturing system example.

II. MANUFACTURING SYSTEM EXAMPLE

Consider the manufacturing system of Figure 1 consisting of lot generator G, single lot machine M and external lot inflow $\mathbf{v}$. This external input can be regarded as third-party lot deliverance, with arrival times known in advance. Buffer B has infinite capacity and makes sure that an incoming products from $\mathbf{v}$ can be stored if needed and does not affect the dynamics. The machine assembles the two incoming products. The machine has constant process time m. The times of arrival of the external input are stored in vector $\mathbf{v}$. Single-place conveyor C transports lots between generator G and machine M with transport time c.

A. System dynamics

The machine starts assembling the two incoming parts when they are both available and the previous assembled lot has left the machine. Until both parts are present, they stay in the preceding buffer or conveyor. If the machine is busy and the two next parts are present, they also stay there.

In general, parameters that characterize the manufacturing system are the number of lots a machine can process simultaneously, process times, batch sizes and buffer capacities. In addition, product recipes, the number of different product types, routes and allowance for takeover specify the transport between single processes. The manufacturing system of Figure 1 can be described by the following rules:

- machine M can only start assembling when both components (from $\mathbf{u}$ and $\mathbf{v}$) are available;
- machine M can only process one assembly at a time;
- assembled products can always leave the system;
- lots can not pass other lots;
- conveyor C has space for one lot; buffer B has infinite capacity.

Before we are able to specify the dynamics of the system explicitly, we determine the system's state and specify the inputs and outputs. Depending on the control strategy a mathematical formulation of the system dynamics is chosen.

0-7803-9568-9/05/$20.00 ©2005 IEEE

B. State and inputs of the system

The state of a system is defined as all information needed together with input signal(s) to predict future output signal(s). For this manufacturing system we do not have an explicit state evolution description yet. The system can be described by max-linear equations (see [1]): all state elements are time values. It is not convenient to construct a meaningful and feasible initial condition for such a system given a physical description of the situation of the manufacturing system: where are lots (in buffers or being processed/transported); this is explained further in Section III). In max-linear system descriptions state x is written as a function of event counter k rather than time t.

To specify state x as a function of time t, our proposal is to define the state of the manufacturing system as a description of what actually can be seen at a certain time instance. This is a major difference with most discrete event descriptions: $x(t)$ instead of $x(k)$. A more elaborate study on specifying systems in both time domain and event domain with a mapping between them is presented in [8]. For the system of Figure 1 we distinguish four modes: machine M can be busy/idle and the conveyor contains a/no lot. These four combinations can be described by two variables: boolean variable x_1 indicating whether a lot is currently on the conveyor and boolean variable x_2 indicating whether a lot resides on the machine. For notational purposes, the boolean variables are represented as 'true' and 1, or 'false' and 0. In addition to these boolean variables, non-negative real variable x_3 indicates the remaining convey time and the non-negative real variable x_4 indicates the remaining process time. The booleans are necessary because a remaining process time of 0 does not rule out an empty machine. State element x_5 gives information about the number of lots that has already left the system. We need this information to index reference vector $\mathbf{r}$ and external input vector $\mathbf{v}$ (e.g. $\mathbf{v}(2)$ means the second element of $\mathbf{v}$). The state vector for the given manufacturing system thus is:

$$\mathbf{x} \in \{0,1\}^2 \times \mathbb{R}_+^2 \times \mathbb{N} \tag{1}$$

The input $\mathbf{u}$ is the manufacturing schedule containing release times of lots into the system and start times of machines. For each lot two time instances have to be scheduled: $u \in \mathbb{R}^m$ with $m = 2$. Component u_1 is the time instance the lot has to be generated. Component u_2 is the time instance the machine should start assembling the two incoming parts.

C. Control problem

The control goal is to meet a certain customer demand specified by due dates in vector $\mathbf{r}$. Within this control goal, a just-in-time policy is applied. The just in time policy assures that value is added as late as possible, which is cost effective.

III. LITERATURE REVIEW

Control of manufacturing systems has been a domain of interest during the last decades. As mentioned in the introduction, discrete event models and control methods exist for the class of manufacturing systems in this research. Two approaches are discussed in this section.

One way of controlling manufacturing systems is by using the max-plus algebra approach (see [1]). This algebra enables exact computation of event dates and works strictly in the event domain. Maia et al. [7] compute feedback controllers in this event domain based on the representation of max-plus models as timed event graphs (TEGs). A drawback of this method is that it is not possible to incorporate a physical initial condition of the manufacturing system in the computations without remodeling the system. And even if the system is remodeled to incorporate initial effects, only initial buffer levels can be incorporated rather quickly, but not products that are initially being processed (attempts to deal with this, like in [3] lead to additional states and inputs, which is mostly unwanted). Earlier, in 1998, Cofer and Garg [2] computed supervisory controllers for timed discrete event systems. Their general idea is to delay controllable events to modify system behavior to meet some other behavior.

Another approach using max-plus algebra in control of manufacturing systems is application of MPC to control such systems. De Schutter and Van den Boom first studied this method in [4]. Under conditions on the control objective and constraints, an MPC problem can be solved as a convex optimization problem. In further publications, based on the same idea of using MPC for control of max-plus described manufacturing systems, Van den Boom, De Schutter et al. deal with perturbations, uncertainties, adaptive model predictive control (see e.g. [10]). Within the max-algebraic notation of the system, it is again not convenient to find a feasible and meaningful initial condition for the system, since it involves backward iteration of the system, which basically comes down to repeatedly solving a set of max-algebraic equations. Another problem that may arise is a 'timing' problem: at a certain time instance the controller gives new input dates that have already passed in time or, analogously, information about future events is needed to compute optimal new input dates. This causality problem is also mentioned in [7], where Maia et al. turn this non-causal optimal solution into a suboptimal causal solution.

A second way of controlling the class of manufacturing systems under consideration is by means of sequencing and scheduling and especially rescheduling. For the class of manufacturing systems under study, scheduling problems are not NP-hard, since no combinatorial problems introduce MILP problems. In practice, all kinds of disturbances on the original production schedule occur. One could try to take all possible disruptions into account while generating the original production schedule. A possible effect can be the realization of conservative production schedules. Another option is to generate simple production schedules and repair/regenerate this schedule based on irregular events or new information. This procedure is called rescheduling. The world of rescheduling has been divided clearly in continents by Vieira et al. in [12]. They clearly state the differences between rescheduling environments, rescheduling strategies, rescheduling policies and rescheduling methods and give an extensive overview of papers in the different fields of rescheduling. However, the control theoretic approach of state feedback scheduling remains underexposed.

In this paper, we show that an intuitive approach in control of manufacturing systems leads to quite satisfying results. We deal with both causality and sensible use of initial conditions. First we consider the small example presented in the previous section. An explicit time-varying state feedback control law is presented as a proof of concept. Second, we use LP techniques for rescheduling in an MPC approach to deal with disruptions.

IV. CONTROL STRATEGY

In this research we use MPC to control manufacturing systems. Since the customer demand pattern is not known beforehand for a very long time and because the customer demand changes over time, we use the finite horizon MPC method. In the control problem, we optimize time instances at which events occur. Therefore, we introduce event horizon N_c. In [4] De Schutter and Van den Boom also use MPC with an event horizon. However, in this research we use state $\mathbf{x}(t)$ as initial condition rather than state $\mathbf{x}(k)$. The great advantage here is that we do not have to make backward iterations in the event domain to construct the initial condition based on what we see in the manufacturing system. In this study we translate current state $\mathbf{x}(t)$ at the moment of optimization directly into event based MPC input constraints.

The event horizon N_c is the number of products that has to be optimized. Each product has to be processed a number of manufacturing steps. At the moment of optimization, lots may have finished part of these process steps. For those lots, less time instances have to be optimized than for lots that still have to be fed into the system. In other words, a difference exists between the horizon (the number of products to be scheduled) and the number of events that has to be optimized (depending on number of process steps and partial completion of lots at the moment of optimization).

As explained in Section II-C we want to minimize the difference between actual output dates $\mathbf{y}$ and due dates $\mathbf{r}$. In the MPC setting, we minimize over the control horizon N_c. Within this control goal, events have to occur as late as possible (JIT). The former control goal is superior to the latter. The problem we consider is:

$$\min_{\mathbf{u},\mathbf{y}} J = \min_{\mathbf{u},\mathbf{y}} -\lambda^{\mathrm{T}}\mathbf{u} + \sum_{i=1}^{N_c} |y_i - r_i| \tag{2}$$

where vector $\mathbf{u}$ contains all time instances of events that have to be optimized and λ is a weighting vector containing small positive values assuring the priorities of the two control goals. The dimension of vector $\mathbf{u}$ may change over time, since some time instances at which events occur do not always have to be computed (partial completion of lots). This may have consequences for implementation of the MPC scheme, as will be shown in Sections V and VI.

V. EXPLICIT STATE FEEDBACK CONTROL

Due to inevitable perturbations in a physical manufacturing system, changes in a computed manufacturing schedule might be necessary. As mentioned in Section III rescheduling is a very broad area of research, but the rescheduling methods have in common that (part of) the schedule has to be recomputed or even re-optimized. From a control point of view, this is not a desired way of handling perturbations. An explicit feedback law would immediately deal with perturbations. Based on the actual condition of the system (current state $\mathbf{x}(t)$) , the optimal schedule should be available continuously over time without re-optimization. In Example 1 an explicit time-varying state feedback law is presented.

Consider again the manufacturing system of Figure 1. An optimal schedule is computed with a horizon of two products. For this horizon and the cost function given in Section IV, a state feedback law is given in (3). This feedback law $\mathbf{u} = \mathbf{u}(\mathbf{x},t) = \begin{bmatrix} u_1 & u_2 & u_3 & u_4 \end{bmatrix}^{\mathrm{T}}$ gives the optimal production schedule for two products.

Proposition 1 (explicit state feedback law): Consider the manufacturing system of Figure 1 and the MPC objective function (2) with horizon $N_c = 2$ and $0 < \lambda \ll 1$. For an arbitrary reference vector (due dates) $\mathbf{r}$ and arbitrary external input signal $\mathbf{v}$ the times of generation of new lots (u_1 and u_2) and the start times of machine M (u_3 and u_4) are given in feedback law (3), depending on state $\mathbf{x}$ and current time t.

$$\begin{aligned}
u_1 &= \begin{cases} u_3 - c & \text{if } \neg(x_1 \wedge x_2) \\ \infty & \text{if } x_1 \wedge x_2 \end{cases} \\
u_2 &= \begin{cases} u_4 - c & \text{if } \neg(x_1 \vee x_2) \\ \infty & \text{if } x_1 \vee x_2 \end{cases} \\
u_3 &= \max \begin{pmatrix} t + (1 - x_1) \cdot c + x_3 \\ t + x_4 \\ \mathbf{v}(1 + x_2 + x_5) \\ \mathbf{r}(1 + x_2 + x_5) - m \end{pmatrix} \\
u_4 &= \begin{cases} \max \begin{pmatrix} u_3 + m \\ \mathbf{v}(2 + x_5) \\ \mathbf{r}(2 + x_5) - m \end{pmatrix} & \text{if } x_2 \\ \infty & \text{if } \neg x_2 . \end{cases}
\end{aligned} \tag{3}$$

In words: machines start assembling (u_3 and u_4) if the two incoming parts are available and not in transport anymore, the previous lot has left the machine and as close as possible to the due date r. New lots are generated (u_1 and u_2) exactly one conveyor period prior to these dates. If start times or generation times are not to be computed because lots are already in transport or on machines (as explained in Section IV), their corresponding values of control input $\mathbf{u}$ are set to infinity.

Example 1 (state feedback in disrupted simulation): For the manufacturing system of Section II a visualization of feedback law (3) in a simulation is given in Figure 2. To make this visualization, we started with an empty factory and the following reference and input vectors: $\mathbf{r} = \begin{bmatrix} 3 & 6 & 8 & 12 & 15 & 25 & \ldots \end{bmatrix}^{\mathrm{T}}$ and

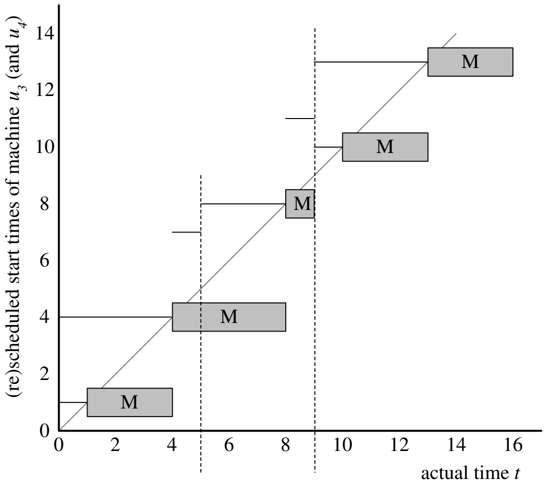

Fig. 2. Results of feedback law in simulation under disruptions.

$\mathbf{v} = \begin{bmatrix}0 & 4 & 5 & 6 & 12 & 14 & \ldots\end{bmatrix}^{\mathrm{T}}$. The process time m of the machine equals 3 time units and conveyor time c equals 1 time unit. The horizontal axis of the figure corresponds with the actual running time. The vertical axis corresponds with scheduled times. The horizontal lines in the figure are scheduled job starts u_3 and u_4 according to the explicit time-varying state feedback law (3). For simplicity, only machine jobs are shown. Transport of new lots entering the system has not been ignored, but is just not shown in the graph. In the figure, it can be seen that at $t = 0$ two jobs are scheduled ($N_c = 2$) and that when the machine is occupied, only one new machine start is scheduled ($u_4 = \infty$, i.e. not scheduled). At $t = 5.0$ a perturbation occurs (dashed line) while processing the lot on machine M resulting in immediate restarting of that lot on the machine. In Figure 2 we see that before $t = 5$, the next lot was scheduled to start on machine M at $t = 7$. The disruption at $t = 5$ immediately results in a revised schedule for the subsequent lot: from $t = 5$ this lot is scheduled to start on the machine at $t = 8$. A different situation occurs when lots finish earlier than planned. The feedback law will move up the start time of the next lot. At $t = 9$ a second disruption occurs: the third lot had been finished earlier than expected. Before $t = 9$ the subsequent lot was planned to start at $t = 11$, but after the disruption this lot can start at $t = 10$ (not immediately at $t = 9$ because transport/convey time $c = 1$ after lot generation is involved).

Another interesting property of feedback law (3) is shown in Figure 2. The horizontal lines (scheduled start times) never cross the diagonal line. This means that no timing problem occurs: jobs can never be started prior to the current time t.

For larger systems where it is not easy or possible to specify an explicit feedback law or where this feedback law becomes too large itself, it may be convenient to control the system by other means. In the next section, we develop a linear programming optimization procedure which calculates the optimal production schedule.

VI. FORMULATION OF LP PROBLEM

In cases where feedback controllers can not be computed or specified explicitly, we look for other options to deal with the control problem. In this section we use linear programming to compute the optimal production schedule. The LP formulation is event based: time instances at which events occur are optimized. The current state however is time-dependent and is included in the LP formulation by means of constraints. The current state also determines the number of design variables in the LP problem (lots that are already in the system may have finished some process steps already, which are not to be computed anymore). De Schutter and Van den Boom [4] also mention the possibility of solving their event based MPC control problem using LP problems, but they do not explicitly take into account the differing initial situations.

Although LP is a static optimization procedure, by repeating the optimization at events or after certain time periods and based on the current state, a dynamic system is created. The optimal schedule is not computed continuously over time, but on request.

The dynamics of the manufacturing system (the rules as described in Section II), have to be put into a standard LP form. The goal is to minimize objective J with respect to design variables $\mathbf{w}$. In the following subsections, different elements of the LP problem are treated. Example 2 illustrates the formulation of an LP problem.

A. Design variables

Design variables $\mathbf{w}$ are all generation/start times of lots at all processes within the horizon. The number of design variables can be augmented with some auxiliary variables $\mathbf{z}$, depending on the objective function form.

B. The objective function

The objective function assigns costs to earliness and tardiness (production/due date error). Moreover, the JIT policy is accounted for in the objective function. Due dates vector $\mathbf{r}$ has to be followed as closely as possible by the machine output dates vector $\mathbf{y}$. Linear costs are assigned to the difference between output dates $\mathbf{y}$ and reference $\mathbf{r}$. Both earliness and tardiness are penalized: $|y-r|$. This absolute value function in the objective can not be put into standard LP form directly. Auxiliary variables z (additional design variables) have to be introduced for each output y, together with some additional inequality constraints. Note that unequal penalties on earliness and tardiness introduce extra auxiliary variables. The absolute value of the objective function is then written as:

$$\begin{aligned} |y-r| &= z = \max(y-r,\ r-y) \\ \min|y-r| &= \min z \end{aligned} \tag{4}$$

with additional constraints:

$$z \geq y-r \text{ and } z \geq r-y. \tag{5}$$

In addition to costs on the production error, relatively small costs are assigned to all physical start times of lots, implementing the JIT policy. The problem now becomes:

$$\min_{\mathbf{w},\mathbf{z}} J = \min_{\mathbf{w},\mathbf{z}} -\lambda^{\mathrm{T}}\mathbf{w} + \sum \mathbf{z} \tag{6}$$

with λ a weighting vector. For just in time policy, the elements of λ can be relatively small positive numbers.

C. Constraints

The physical properties and limitations of the manufacturing system are translated into linear (in)equality constraints. The rules describing the dynamics of the example manufacturing system of Section II all translate into linear inequality constraints.

D. Incorporating the initial condition of the system

It is not likely that every optimization step starts with an empty manufacturing system. Therefore, a powerful way to incorporate the initial condition/state of the system in the LP problem is needed. This initial condition contains information about machine and conveyor statuses and buffer levels. The state elements as described in Section II-B must be translated into constraints.

Example 2 (formulation of LP problem): Consider again the manufacturing system of Section II with the same parameter and vector values as in Example 1. Suppose that initially (at time $t = 0$) machine M is busy with a remaining process time of 2 time units ($x_2 = 1$, $x_4 = 2$), conveyor C is empty ($x_1 = 0$, $x_3 = 0$) and no lot has finished so far ($x_5 = 0$). A horizon of three products is used and the control objective again is to meet the customer demand with equal penalties on earliness and tardiness under JIT policy.

One single formulation of the LP problem with a horizon of three products is then:

- Determine the number of design variables:
  Since the machine is occupied at the time of optimization (its output time can not be influenced anymore), only two lots need to be generated. (generation times w_1 and w_2). As a result, machine M also needs to start twice (w_3 and w_4) and 2 auxiliary variables (z_1 and z_2, see VI-B) have to be introduced. $\mathbf{w} = \begin{bmatrix} w_1 & w_2 & w_3 & w_4 & z_1 & z_2 \end{bmatrix}^{\mathrm{T}}$.
- Construct the constraints:
  All design variables (excluding auxiliary variables) must be larger than the current time: $w_i \geq t$ for $i \in \{1,2,3,4\}$. Incoming lots from generator G enter the system after each other: $w_1 \leq w_2$. Machine M processes lots if both incoming lots have arrived: $w_3 \geq w_1 + c$, $w_4 \geq w_2 + c$, $w_3 \geq \mathbf{v}(1)$ and $w_4 \geq \mathbf{v}(2)$. The machine processes the new lots after each other: $w_4 \geq w_3 + m$.
- Incorporating the initial condition: $w_3 \geq x_4$.
- The objective function. Since the first product output time can not be influenced (lot has already started) no costs will be assigned to this product. For the remaining two products, constraints have to be constructed for the two auxiliary variables z_1 and z_2 (see (5)).

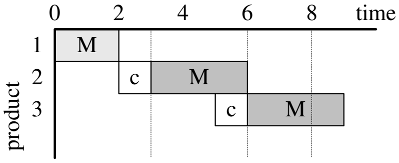

Fig. 3. Lot-time diagram example 1.

This optimization problem consists of 15 inequality constraints. Results of the optimization procedure are shown graphically in Figure 3. The due dates are indicated with the dotted lines. Product number 2 finishes exactly in time, product 3 is late. Equal penalties on earliness and tardiness combined with the JIT policy and the fact that only two products are actually optimized are an explanation for this.

VII. CASE STUDY

To illustrate the concepts of the previous section(s) further, we apply the LP-based MPC scheme to a bigger case study. In this case study we control a manufacturing system by means of repeated optimization of an LP problem as explained in Section VI. We use the same objective function as before. The manufacturing system contains stochastic behavior on the process times of machines. Within the optimization procedure, the process times are treated as if they were constant. As a result, the optimal schedule can not be implemented exactly due to the stochastic behavior. Therefore rescheduling is necessary. Whenever the actual process time of a job becomes known, the schedule is recomputed. This method results in feasible manufacturing schedules.

Consider the manufacturing system of Figure 4 consisting of generator G, infinite buffer B, machines M_1 and M_2, finite buffer $B(2)$ with capacity 2, batch machine F and exit E (stock). The mean process times of the machines are 3, 2 and 3 time units respectively (gamma distributed with variance 0.5) and the batch size equals 2. Products have recipes as indicated in the figure. The buffers B and $B(2)$ send to and receive from machines M_1 and M_2 alternately. The first lot has number 0 and goes via machine M_1. The horizon over which a schedule is optimized equals six lots. The reference vector (due dates) for single lots (not batches) is:

$$\mathbf{r} = \begin{bmatrix} 6 & 8 & 10 & 10 & 12 & 14 & 17 & 18 & 21 & 22 \end{bmatrix}^{\mathrm{T}}$$

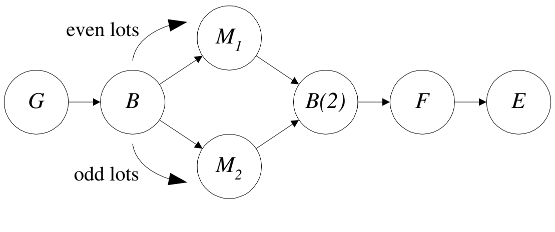

Fig. 4. Manufacturing system configuration.

The state of this system in the sense of Section II-B has the structure:

$$\mathbf{x} \in \{0,1\}^3 \times \mathbb{R}^3_+ \times \mathbb{N}^3 \tag{7}$$

representing 3 boolean variables indicating the presence of lots or batches on machines, 3 remaining process times and 3 natural variables (number of lots in B, $B(2)$ and the number of finished lots).

A discrete event model of the manufacturing system has been made using the specification language χ [5], [9] and has been coupled to MATLAB. In MATLAB the current state is translated into an LP problem. This LP problem uses constant process times (equal to the mean values). The LP problem is solved and gives the optimal schedule for horizon $N_c = 6$ and weighting vector elements $0 < \lambda \ll 1$. Whenever a job starts, its process time is drawn from the distribution. Immediately, the remaining schedule is recomputed. The first start event of a machine is then implemented in the discrete event simulator and the procedure starts over again by measuring the state and optimize in MATLAB. A simulation has been carried out. The results are shown in Figure 5. On the horizontal axis the actual simulation time is represented. On the vertical axis, the scheduled start times of the 3 machines is shown. The boxes are actual jobs with their process time in it. The scheduled job start can be shifted forward and backward, as a reaction to known actual process times. For example, the batch starting at $t = 7$ had been shifted forward at $t = 1$ because the very first lot had a process time bigger than 3. However, at $t = 4.11$ the start time of the batch starting at $t = 7$ could be shifted backward, since the first batch took shorter to process (2.29 instead of 3) than planned.

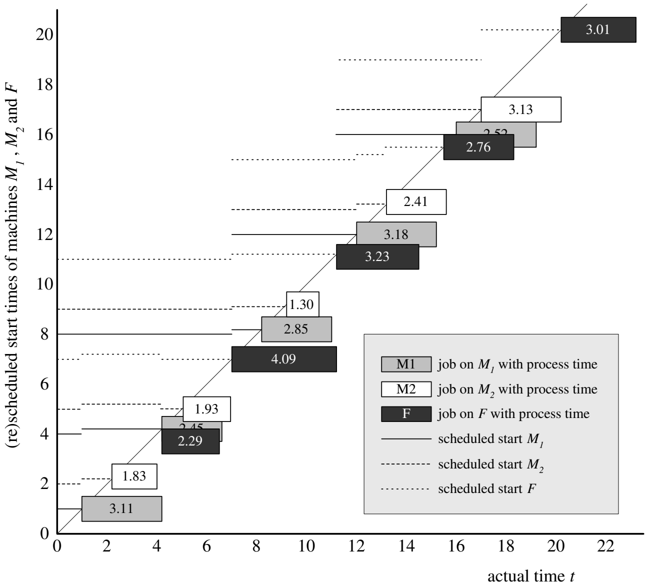

Fig. 5. Simulation with stochastic process times and repeated optimization.

VIII. CONCLUSIONS AND FUTURE WORKS

A. Conclusions

In this paper we have shown that discrete event manufacturing systems can be controlled in an MPC setting by means of an explicit state feedback law. The state of the system is a function of time t instead of the commonly used event counter k. For a small example (a perturbed manufacturing system), we showed the correct output of the feedback law. For larger manufacturing systems or larger horizons, the explicit state feedback law can be replaced with a linear programming optimization procedure, which is invoked whenever a disruption occurs or after passing of certain events. In a bigger case study, we have shown this control strategy implemented in a discrete event simulation model.

B. Future works

Future work on this subject focuses on more structured ways of deriving explicit state feedback control laws and investigation of stability of the controllers. In addition to this, generic formulation of LP problems based on manufacturing layouts and product recipes is another topic.

REFERENCES

[1] F. Bacelli, G. Cohen, G.J. Olsder, and J.-P. Quadrat. *Synchronization and linearity: an algebra for discrete event systems*. Wiley, 1992.

[2] D.D. Cofer and V.K. Garg. Supervisory control of real-timediscrete event systems using lattice theory. *Proc. 33rd Conference on Decision and Control*, pages 978–983, December 1994.

[3] G. Cohen, P. Moller, J.-P. Quadrat, and M. Viot. Algebraic tools for the performance evaluation of discrete event systems. In *IEEE Proceedings*, volume 77, pages 39–58, 1989.

[4] B. De Schutter and T. van den Boom. Model predictive control for max-plus-linear discrete event systems. *Automatica*, 37(7):1049–1056, July 2001.

[5] A.T. Hofkamp and J.E. Rooda. χ reference manual. Technical report, Technische Universiteit Eindhoven, http://se.wtb.tue.nl/documentation, 2002.

[6] J. Kimemia and S.B. Gershwin. An algorithm for the computer control of a flexible manufacturing system. *IIE Transactions*, 15(4):353–362, December 1983.

[7] C.A. Maia, L. Hardouin, R. Santos-Mendes, and B. Cottenceau. Optimal closed-loop control of timed event graphs in dioid. *IEEE Transactions on Automatic Control*, 48(12):2284–2287, Dec. 2003.

[8] B.J.P. Roset, H. Nijmeijer, J.A.W.M. van Eekelen, E. Lefeber, and J.E. Rooda. Event driven manufacturing systems as time domain control systems. *44th IEEE Conference on Decision and Control and European Control Conference, Sevilla*, 2005.

[9] D.A. van Beek and J.E. Rooda. Languages and applications in hybrid modelling and simulation: Positioning of Chi. *Control Engineering Practice*, 8(1):81–91, January 2000.

[10] T.J.J. van den Boom, B. De Schutter, G. Schullerus, and V. Krebs. Adaptive model predictive control for max-plus-linear discrete event input-output systems. *IEE Proceedings – Control Theory and Applications*, 151(3):339–346, May 2004.

[11] F.D. Vargas-Villamil, D.E. Rivera, and K.G. Kempf. A hierarchical approach to production control of reentrant semiconductor manufacturing lines. *IEEE Transactions on Control Systems Technology*, 11(4):578–587, July 2003.

[12] G.E. Vieira, J.W. Herrmann, and E. Lin. Rescheduling manufacturing systems: a framework of strategies, policies, and methods. *Journal of Scheduling*, 6(1):39–62, 2003.