Proceedings of the 17th World Congress
The International Federation of Automatic Control
Seoul, Korea, July 6-11, 2008

Robust Scheduling of cyclic Flow Shops based on stochastic Collision Functions

Wolfgang Meyer and Claudia Fiedler

Institute of Automation, Hamburg University of Technology, 21071 Hamburg, Germany (e-mail: w.meyer@tu-harburg.de , claudia. fiedler@tu-harburg.de)

Abstract: By the proposed methodology for robust controller design, we quantify the schedulability of a series of process plans (or jobs) to be produced at a robotic work cell with several machines and limited transport and processing capabilities. The uncertainties of the plant model as there are statistically distributed and event driven variations of transport and processing times as well as of job release intervals, are captured by a stochastic timed Petri net. Robustness of the schedules is measured in terms of a plan achievement function which plays the role of a fitness landscape in the multi dimensional search space of feasible and non feasible schedules. The definition of the plan achievement function goes back to a collision avoidance mechanism. The approach is exemplified for periodic schedules of cyclic flow shops.

1. PROBLEM STATEMENT

Consider the following production planning and control problem: In a robotic work cell, one robot serves several assembly stations which are arranged in a semi circle, in a fixed sequence of moves, repeatedly (Fig.1).

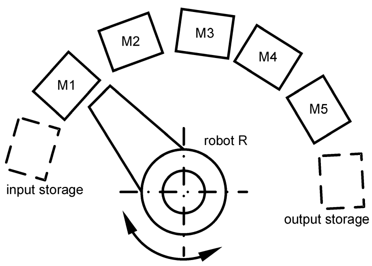

Fig.1. Physical layout of robotic work cell

All jobs (or parts, or processes, or products) entering the cell obey the same sequence of operations on the machines, with fixed processing times, and no buffering is allowed between operations. Such an environment is a *cyclic flow shop* (Timkovsky, 2004). The term cyclic refers to the common machine, the robot, which performs a handling or transport operation between each consecutive process operation in each job. Fig. 2 shows the process plan as a Gantt chart for this rank-1, multiplicity-6 problem (rank: number of cycle machines; multiplicity: number of loops). Three identical processes P1=P2=P3 are released to the cell from the input buffer, the first one at time zero (v_1=0), the following ones after release intervals v_2 and v_3.

The vector **v** = (0, v_2, v_3, …) is the schedule to be optimized for maximum plant productivity. For equal release intervals and identical jobs as in Fig.2, we obtain periodic schedules **v** = (0, v, v, …) both for machine and robot operations and arrive at stationary behaviour of the dynamic production system which means a fixed sequence of robot operations

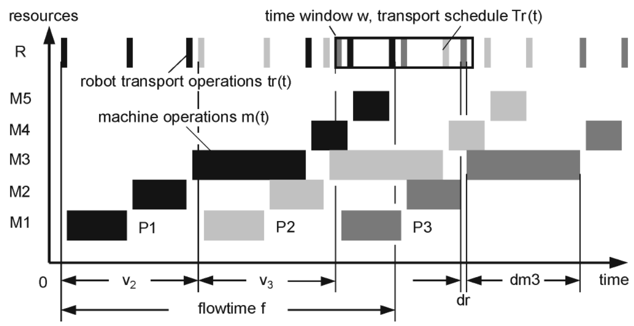

Fig.2. Resource Gantt chart for three identical processesP1, P2, P3. M machines, R robot. dm, dr durations of processing and transport operations

Tr(t) inside the time window w in Fig.2. Then, the *optimum* solution for the cyclic flow shop scheduling problem (CFSP)is the periodic sequence of robot operations (as there are: empty moves, loaded travel, and handling in general) with the smallest possible vector length $|\mathbf{v}|$ which fulfils the time requirements of the process plan. The minimum vector length (or shortest schedule **v** for a set of processes P_i) is limited by overlaps of machine and robot operations in the Gantt chart of Fig.2. Such overlap of bars is called a *collision*: of parallel requests by robot and machine operations for their transport and processing resources. A feasible schedule must be free of collisions, at least within a certain range of probability. In this paper, we try to quantify the *feasibility* (or *schedulability*) of a production plan endangered to transport collisions because of plant resource limitations and environmental distortions. We develop two measures of plan robustness. The first, discrete one measures the sensitivity of plan performance against variations of the transport routing (or the transport sequence Tr(t) in Fig.2). The second, continuous one captures the operational uncertainties of the plant by probabilistic measures for time parameters like transport and processing time intervals. By a combined robustness analysis, we determine whether the planned operations of the perturbed plant can still be maintained in case of parameter variations

978-1-1234-7890-2/08/$20.00 © 2008 IFAC 10.3182/20080706-5-KR-1001.0327

and up to which extent the original plan can still be achieved. Plan achievement here is measured in terms of time deviations of the implemented process plan as compared to the original one. The theory of robust scheduling is illustrated and evaluated by a stochastic Petri net model. For clarity of explanation, we focus on periodic schedules in this paper. However, the approach also covers non periodic schedules with varying jobs and sequence dependent set up or transport times.

2. ROBUST SCHEDULING: STATE OF THE ART

Robust scheduling methods aim to face uncertainties that usually arise during schedule execution, as there are: machine failures, urgent job arrivals and cancellations, due date changes etc. (Leung, 2004). These events may generate considerable differences between the predetermined schedule and its actual realization on the shop floor and sometimes may afford rescheduling. A great amount of work related to robust scheduling, therefore, has been devoted to rescheduling policies and methodologies; for a recent review and a rescheduling framework see (Vieira *et al.,* 2003). The frame work classifies rescheduling *environments* as static (deterministic and stochastic) or dynamic ones (cyclic, flow shop and job shop production); rescheduling *strategies* as there are dynamic (dispatching rules and control-theoretic approach) or predictive-reactive ones (periodic, event-driven and hybrid policies); and rescheduling *methods* like schedule generation or schedule repair (right-shift, partial and complete regeneration). Our application refers to the dynamic case under periodic or hybrid control policies.

In almost all applications, the robustness approaches fall into two broad categories: *worst case scenarios* which strictly keep to (hard) constraints but are conservative and time consuming, and *robust optimization* which treats some variables as (soft) constraints represented as fuzzy or stochastic variables and, therefore, may provide a probabilistic value for feasibility and schedulability of a process plan. Our approach mainly builds on the ideas of robust optimization and adaptive scheduling put forward in (Jensen, 2003) and (Mattfeld, 1996). We extend our previous investigations into deterministic timed Petri net schedulers (Fiedler *et al.,* 2005) (Fiedler, 2006) by stochastic elements and develop a methodology for robustness analysis on top of that representation.

3. PLANT MODEL AND COLLISION FUNCTIONS

The plant and the process plan of Figs. 1 and 2 are represented as p-timed Petri net (David *et al.*, 2005) and implemented as t-timed Petri net with the help of the PACE-tool (www.ibepace.com, 2007). The (p-timed) A-path (Zhou, 1993) of Fig.3 captures the sequence of processing operations m1, m2, … and their processing times dm1, dm2, … as well as the (loaded) transport and (unloaded) move operations of the robot, tr and mo, according to the process plan of Fig. 2. The availability of machine and robot resources (see Fig.1) is modelled by additional controller places M1, M2, … and R. Finally, tokens (black dots) represent processes (or jobs, or parts) being sent through the system. These principles can be extended to job shops with sequential and parallel machines, different process plans (or A-paths), and sequence dependent set up or move times, which is described elsewhere (Fiedler *et al.,* 2005) (Fiedler, 2006). In this paper, we restrict ourselves to the robotic cycle shop of Fig.3 without multi purpose machines or loops in the production line. The figure displays one part of the complete plant which consists of 5 such modules arranged in a sequence.

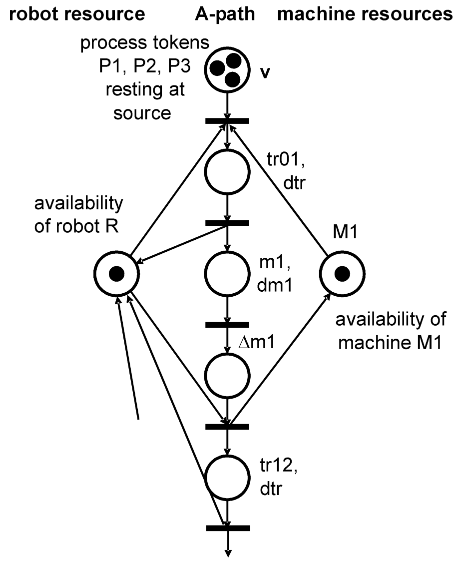

Fig.3. Petri net plant module

In Fig.3, two transport operations tr01 and tr12 are interlaced with one machine operation m1. Each operation (circles or *places*) begins and finishes with start/stop conditions (bars or *transitions*). The transport times dtr_i and processing times dm_i can either be deterministic (constant) or statistically distributed. Further, availability nodes for robot and machine resources monitor and control the flow of tokens (dots) through the net. This is done by synchronization of events: A-path transitions can only fire and thereby start an operation if control places contain tokens. The main idea here is that operations can only be performed if resources are available, otherwise they are delayed according to a simple FIFO rule. Tokens in resource places thus model the control flow and represent the actual scheduling algorithm. In case of more complex selection rules, the availability nodes are replaced by scheduling networks, see (Meyer *et al.*, 2006).

The source tokens (or processes) are released to the Petri net with varying rates. Depending on the release intervals v_i between succeeding tokens, queues may build up in front of machines if these are still busy with the preceding process or if the transport robot is not available. These delays or aberrations ΔP from pre specified processing times dP (see Table 1) are measured by additional places Δm1, Δm2, … for each operation and machine, see Fig.3. These additional places act as buffers for tokens which cannot be served immediately by the busy transport robot.

Table 1. Process plan for the running example

resources	M1	M2	M3	M4	M5	R	
operations	m1	m2	m3	m4	m5	tr	mo
processing times	200	180	380	120	120	20	10

The best schedule, then, is a sequence of release intervals 0, v_2 , v_3 , … for processes P1, P2, P3, … which leads to stable operation, has maximum throughput and sticks to the process plan, i.e. minimizes the sum of plan deviations $\Delta P = \Sigma \Delta m_i$. Fig.4 shows an example for the periodic though unfeasible schedule S480 with dynamic aberrations per machine Δm_i as elucidated by dashed framings. The Gantt chart was obtained from the deterministic plant model of Fig.3 where each place implements a constant processing time dm_i , dtr according to Table 1. Ten succeeding tokens were released to the Petri net at identical release intervals $v_2 = v_3 = \ldots v_i = \ldots v_{10} = 480$.

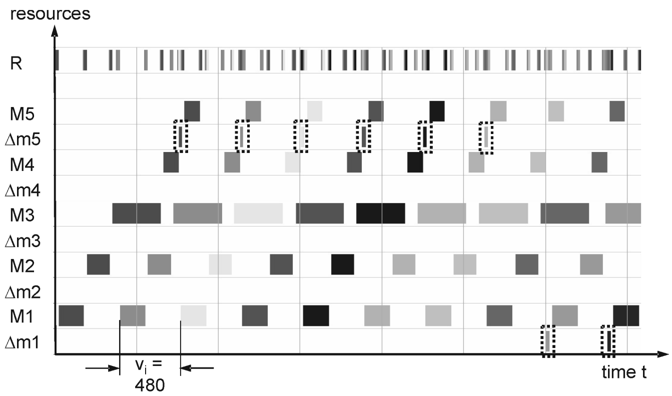

Fig.4. Deterministic Gantt chart for 10 deterministic processes from Table 1 and process release intervals v = 480

Finally, Fig.5a summarizes the outcome of about 2100 of such simulations for different release times v. This diagram of ΔP(v) quantifies the feasibility and quality of (periodic) schedules in terms of plan deviations ΔP for a series of overlapping processes from Table 1. ΔP(v) is termed *collision function*. For schedules **v** = (0, v, v, v, …) with ΔP(v) = 0, no collisions occur. We call them feasible periodic schedules.. The set of feasible release times v which make up a feasible schedule **v,** is $\mathbf{V} = \{\mathbf{v} \mid \Delta P(v)=0\}$ and indicated by thick lined intervals in Fig.5a. All remaining schedules are possible as well but lead to plan aberrations ΔP(v) ≠ 0.

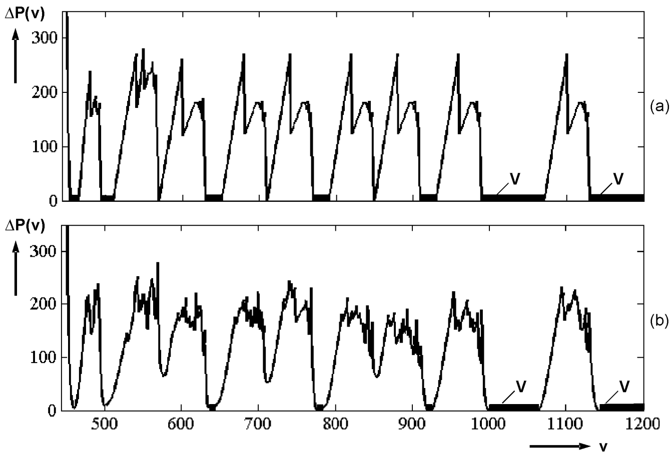

Fig.5. Accumulated plan aberrations ΔP = Σ Δmi as dependent on release intervals v, for ten processes and periodic schedules. (a) Deterministic processes from Table 1. **V** = {**v** | PA(v)=0} set of collision-free schedules. (b) Statistical variations of transport times added to (b): dtr ± Δtr = 20 ± 5.

Fig.5b is the respective stochastic collision function ΔP(v). It was obtained from simulation runs of the stochastic plant model where the timed Petri net of Fig.3 has been changed to a stochastic one by altering the 60 robot transport times (for 10 processes of Table 1) statistically in the course of the simulations (for details see Section 6). By comparison with Fig.5a we recognize that the stochastic set of feasible schedules **V** is considerably smaller than for the deterministic case. This finding supports the common sense expectation that the possibility space (here: the set of feasible schedules) gets smaller for more detailed models like the probabilistic ones (which capture modelling uncertainties of the transport system in addition, in this example).

4. ROBUST SCHEDULING APPROACH

We adopt here the idea of robust optimization for continuous problems and extend it to the discrete domain (Jensen, 2003) (Mattfeld, 1996). Central to problem solving by optimized search is the notation of the search space or fitness landscape. Fig.6a presents a one dimensional example. Per definition, robust optima are located on broad plateaus of the fitness curve whereas sensible optima are located on narrow peaks. A trade-off analysis then compromises between goal achievement and robustness of the solution. One example is the delicate balance between quality and robustness of a production plan, plan quality being measured in terms of plant productivity to be achieved by this plan (Briand *et al.* 2007).

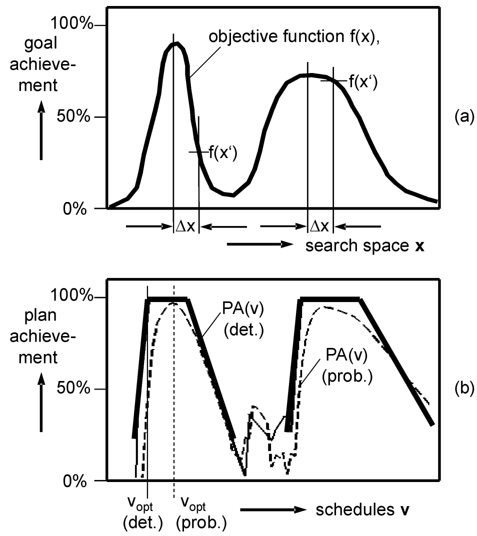

Fig.6. Search space configuration and fitness function. (a) General objective function f(x) and robustness measure Δx. (b) Plan achievement function PA(**v**) according to Eqn.1. Straight line: deterministic process from Fig.6a. Dashed line: stochastic process from Fig.6b. The cut out refers to the first two minima of Fig.5

In production planning, goal achievement is equivalent to zero deviation ΔP of the implemented plan as compared to the original one, e.g. the one of Table 1. For a convenient search space definition, we turn Fig.5 upside down by defining a plan achievement function PA(**v**) by

$$PA(\mathbf{v}) = 1 - \Delta P(\mathbf{v})/\Delta P_{max} \qquad (1).$$

We arrive at Fig.6b where PA(**v**) is interpreted as the objective function in the search space of possible schedules **v** = (0 , v_2 , v_3 … v_i). Every solution is positioned through its vector coordinates v_i and its plan achievement value PA(v_i) in the landscape of (1). For the deterministic case in Fig. 6b (solid line), the objective function is non-differentiable because of the mutual exclusion constraints posed by the limited resources of the plant. Therefore, the usual mathematical techniques for sensitivity analysis are not applicable here, as they are based on derivatives $\partial PA(\mathbf{v})/\partial \mathbf{v}$. The stochastic data of Fig.5b, however, smoothens the plan achievement function and allows for a gradual trade-off between feasible and non feasible schedules (dashed curve in Fig.6b).

The *schedule optimisation problem* involves at least two objectives: (a) Keep to the process plan by minimizing ΔP(**v**), and (b) Find the best schedule **v** without collisions for maximising productivity or minimising makespan. Both postulates (a) and (b) are formalized as follows:

$$\text{Minimize} \quad \sum_{i=2}^{n} v_i \qquad (2)$$

$$\text{subject to} \quad PA(\mathbf{v}) = 1 \qquad (3)$$

for periodic schedules: $\mathbf{v} = (0, v, v, \dots)$

for general schedules: $\mathbf{v} = (0, v_2, v_3, \dots, v_n)$

From Fig.6 it is clear that the optimum schedule is determined by the smallest feasible value v_{opt} , at the very left end of the search space. However, the question is not answered yet if this schedule is still valid in case of variations Δv of the release times Δv or of transport time variations Δtr or of processing time variations Δm. Δv on one hand, and Δtr and Δm on the other, are representatives for the two different kinds of uncertainties which may deteriorate the calculated schedules: input signal variations and model uncertainties.

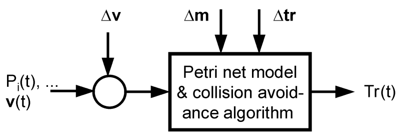

Fig.7. Signal flow diagram of scheduling algorithm including distortions. Input: chain of processes P_i released to the plant at schedule **v** ; output: transport schedule Tr ; distortions: of release intervals Δv, of processing times Δm, and of transport times Δtr

In Fig.7, both types of distortions have been separated. The block diagram summarises the *robust scheduling problem* as dealt with in this paper: For a given sequence of processes $P_i(t)$ and a schedule **v** of process release intervals v_i , find a schedule Tr(t) for the robot transport operations tr_i which is robust against variations in Δ**v**, Δtr_i or Δm_i (Fig. 2 sketches such an example for Tr(t)). In extension of (1), the evaluation functions PA for these three (deterministic or random) variables read as

$PA(\Delta\mathbf{v}) = 1 - \Delta P(\Delta\mathbf{v})/\Delta P_{max}$, **Δm** and **Δtr** are parameters (4)

$PA(\Delta\mathbf{m}) = 1 - \Delta P(\Delta\mathbf{m})/\Delta P_{max}$, **Δtr** and **Δv** are parameters (5)

$PA(\Delta\mathbf{tr}) = 1 - \Delta P(\Delta\mathbf{tr})/\Delta P_{max}$, **Δm** and **Δv** are parameters (6).

Eqns.(4) to (6) are vector equations. For instance, the vector **Δm** = (Δm1, Δm2, Δm3, Δm4, Δm5) stands for the processing time variation Δm_i which adds to dm_i at machine M_i . However, for easy explanation, we will demonstrate the principles of our robustness approach by scalar examples only, in the following.

5. DETERMINISTIC ROBUSTNESS MEASURES

The question now is: For a given sequence of transport operations Tr(t) and a given process schedule **v**, how sensitive (or robust) is **v** against variations of machine processing times dm_i? With other words, how much does Δm_i degrade the plan performance PA as defined in (1)? The answer is formulated in terms of a robustness measure called *plan validity*: A transport schedule belongs to the set of *valid* schedules if the sequence of transport operations tr_i within the time window of width w = v is not altered (compare Fig.2). However, the actual location of tr_i along the time axis may change in order to cope with alterations of process times $dm_i \pm \Delta m_i$. As an example, Fig.8 shows the plan achievement function PA(Δm2) from (5) for Δm2 ranging from about -40 to +30, and for three schedules of v = 453, 455 and 457. For v = 455 and process durations of about 150 < dm2 < 190, the transport schedule Tr(t) remains unchanged. For dm2 > 190, first collisions occur. For dm2 > 220, a new transport routing is generated. In this example, the range of schedule *validity* is -25 ≤ Δm2 ≤ +8 for schedule *feasibility* ranging from 453 ≤ v ≤ 457.

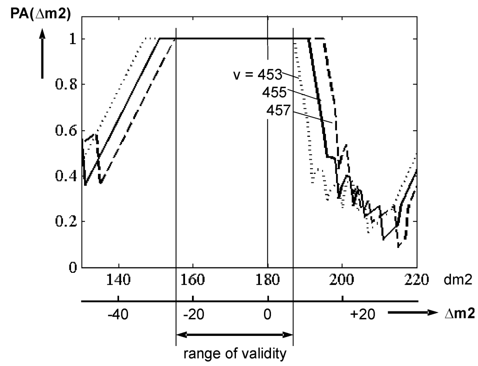

Fig.8. Robustness of plan achievement PA(Δm2) against deterministic variations of the machine processing time Δm2, for three schedules v = 453, v = 455 and v = 457, and Δtr = 0. Process plan from Table 1

6. PROBABILISTIC ROBUSTNESS MEASURES

In the above section, the variables of the robustness analysis, Δm_i , have been treated as deterministic numbers. As an example for an investigation based on probabilistic robustness measures, we now regard the train of transport operations tr_i as a discrete-time stochastic process **tr**(t) in the following sense which matches a lot of realistic situations at the factory floor:

(a) **tr**(t) is predictable (Papoulis, 1984). It consists of a family of deterministic pulse trains tr(t) and is completely specified in terms of the random variable Δtr as specified in Fig.10: **tr**(t) = tr(t) ± Δtr.

(b) **tr**(t) is stationary in the strict sense (SSS (Papoulis, 1984)). That means its first order probability density f(t, Δtr) = f(Δtr) is independent from time.

(c) f(Δtr) is equally distributed in the range $\pm\Delta_{max}$. As an example, for discrete values within $\Delta_{max} = \pm 5$, **tr**(t) and the probability density distribution f(Δtr) are sketched in Fig.11.

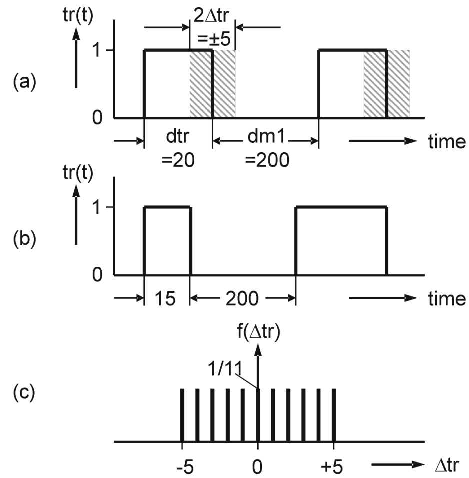

Fig.9. Statistics of stochastic transport process **tr**(t). (a) Chain of deterministic transport operations **tr**(t). Only two operations shown out of six. (b) One sample process of stochastic transport process **tr**(t). Variations Δtr = ±5 added to the deterministic process. (c) First order probability density f(Δtr) of stochastic variable Δtr.

The stochastic transport process **tr**(t) as defined above has been implemented into the Petri net model of Section 3. The outcome of 2100 simulation runs of the random model for $\Delta_{max} = \pm 5$ had been shown in Fig.5b already. These and further experiments with a wide range of Δ_{max}-values are summarised in Fig.10. The figure presents the experimental data in accordance with Eqns.(1) and (6): for PA(v) with Δtr as parameter, and for PA(Δtr) with v as parameter. Especially the latter type of diagram is well suited to separate robust schedules (e.g. S460) from less robust ones (e.g. S456) and even from unfeasible ones (e.g. S452, with PA(Δtr) < 1 for all Δtr). In fact, this diagram has the same meaning and importance as in regular continuous sensitivity analysis.

Similar figures have been obtained for PA(**Δm**) and statistical variations of **Δm** which are omitted here. We show a different process instead in Fig. 11, with doubled move times dmo as compared to Fig.10. Now, the range of feasible schedules is much smaller, leading to less robust schedules. In this example, only schedule S460 with v = 460 fulfils the requirement PA(Δtr) = 1: however, for zero distortions Δtr = 0 only.

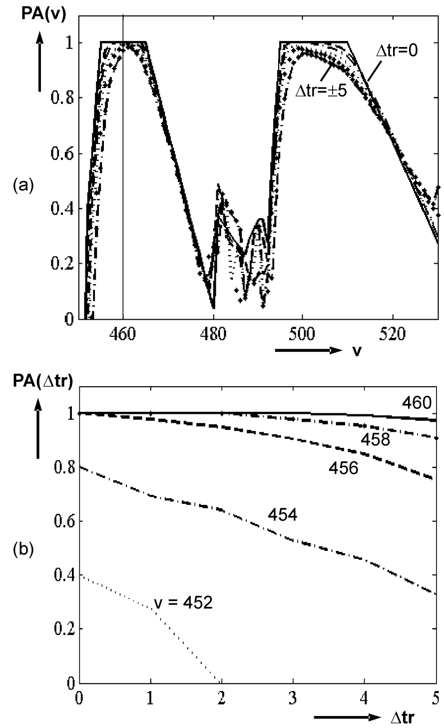

Fig.10. Plan achievement function PA for the running process from Table 1 (mo = 10). Stochastic variations of transport operations added. Periodic schedules only. (a) PA(v) with Δtr as parameter, Δm = 0. (b) PA(Δtr) with v as parameter, Δm = 0

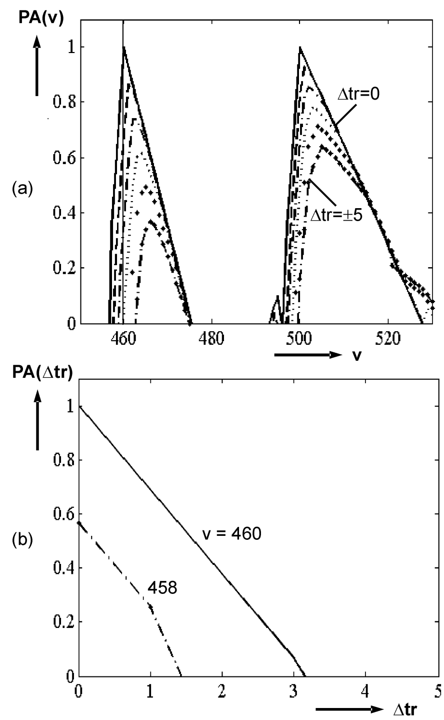

Fig.11. Plan achievement function PA for the process from Table 1 and extended move times dmo = 20. Stochastic variations of transport operations added. Periodic schedules only. (a) PA(v) with Δtr as parameter, Δm = 0. (b) PA(Δtr) with v as parameter, Δm = 0

7. WORST AND BEST CASE ANALYSIS

In Section 2, we mentioned worst case scenarios and robust optimisation as the two competing approaches for plan robustness. Fig.12 compares two deterministic fitness landscapes for the extreme transport aberrations of Δtr = -5 and Δtr = +5, with the stochastic case as treated in the preceding section. The stochastic search space (straight curve) is nicely located in between the deterministic boundaries (dashed and dotted curves) as expected. The figure clearly demonstrates the drawbacks of worst- and best-case scenarios as compared to a full probabilistic treatment. The two dashed curves do not support the human planner with useful decision aids whereas the maximum of the probabilistic one does.

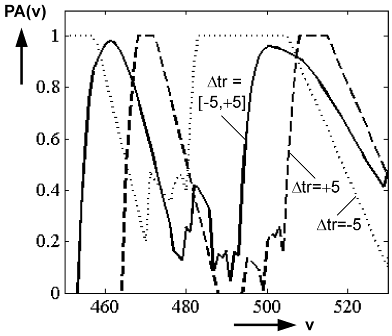

Fig.12 Plan achievement function PA(v) for the running process from Table 1 and periodic schedules. Comparison of deterministic transport processes Δtr = -5 and Δtr = +5 (dotted and dashed curves) and stochastic transport processes Δtr = ±5

8. CONCLUSIONS

In this paper, we tried to quantify the schedulability of a series of process plans (or jobs) to be produced at robotic work cells with limited transport and processing capabilities. Feasibility, validity and finally robustness of the schedules are measured in terms of a plan achievement function PA which plays the role of a fitness landscape in the multi dimensional space of schedules. The constraints and limitations of the plant are captured by a timed Petri net model and a collision avoidance algorithm which acts as a filter on possible solutions. The uncertainties of the plant model as there are statistically distributed or just event driven variations of transport and processing times as well as of order intake, are captured by stochastic places and event driven control elements of the Petri net. By variation of the model parameters, PA is calculated for the complete range of schedules and a wide range of model and signal uncertainties. Based on this set of plan achievement functions, quantitative measures for schedule robustness are derived. They help to solve the fundamental flexibility –productivity conflict which is inherent to all planning decisions at the factory floor. The methodology is exemplified for periodic schedules of cyclic flow shops with constant move times but can be extended to job shops with varying jobs and sequence dependent set up or move times. Different classes of probability distributions can be investigated as well.

ACKNOWLEDGEMENT

We thank Matthias Hüsig for performing simulations and for processing numerical data for this paper.

REFERENCES

Briand, C., H. Trung La and J. Erschler (2007). A robust approach for single machine scheduling problem. *Journal of Scheduling,* **10-3**, pp. 209-221.

David, R. and H. Alla (2005). *Discrete, continuous and Hybrid Petri Nets.* Springer-Verlag, Heidelberg.

Fiedler, C., W. Meyer and A. v. Drathen (2005). Designing Transitory Routing Controllers for Hybrid Plants. *Proc. IEEE Conf. Control Applications,* Toronto, pp. 934-939.

Fiedler, C. (2006). Event-driven Generation of Periodic Hoist Schedules. *Proc. IEEE Conf. Systems, Man, and Cybernetics,* Taipei.

Jensen, M.T. (2003). Generating robust and flexible Job Shop Schedules using genetic Algorithms. *IEEE Transactions on Evolutionary Computing,* **7-3**, pp. 275 – 288.

Leung, J.Y-T. (Ed.) (2004). *Handbook of Scheduling,* Boca Raton: Chapman & Hall.

Mattfeld, D.C. (1996). *Evolutionary Search and the Job Shop.* Physica – Verlag, Heidelberg.

Meyer, W. and C. Fiedler (2006). Mission Control by Coordinating Shared Resources. *Proc. IEEE Conf. Systems of Systems Engineering,* Los Angeles, pp. 71-76.

Papoulis, A. (1984). *Probability, Random Variables, and Stochastic Processes.* McGraw-Hill Book Comp., London.

Timkovsky, V.G. (2004). Cycle Shop Scheduling. In: *Handbook of Scheduling*. pp. 7-1 – 7-22. Boca Raton: Chapman & Hall.

Vieira, G.E., J.W. Herrmann and E. Lin (2003). Rescheduling of Manufacturing Systems: A Framework of Strategies, Policies, and Methods. *Journal of Scheduling,* **6**, pp. 39-62.

Zhou, M. and F. DiCesare (1993). *Petri Net Synthesis for Discrete Event Control of Manufacturing Systems.* Kluwer Academic Publishers, London.