*PERIODICA POLYTECHNICA SER. MECH. ENG. VOL. 45, NO. 2, PP. 175–200 (2001)*

# HYBRID DYNAMICAL APPROACH MAKES FMS SCHEDULING MORE EFFECTIVE

János SOMLÓ

Department of Manufacturing Engineering
Budapest University of Technology and Economics
H–1111 Budapest, Egry J., u. 1, Hungary
e-mail: somloj@eik.bme.hu

Received: March 23, 2001

## Abstract

In the present paper a general job-shop type scheduling problem for FMS is considered. About ten years ago by paper PERKINS, KUMAR [3] a new direction, the use of hybrid dynamical approach to the solution of manufacturing scheduling problems arose. Recently, MATVEEV, SAVKIN [5] developed a qualitative theory for the investigation of these and similar problems.

In the present paper an attempt is made to show the place and way of use of this approach in the solution of industrial problems. The formulation of the tasks involves complex problem obtained by CAPP and PPS subsystems interaction of CIM systems. The effect of the use of hybrid dynamical approach is in the opportunity of overlapping production, which may result in a significant increase of efficiency of equipment utilization. A proposal is given to determine the so called demand rates which are ideal inputs of servers (machines) realizing production according to the orders of higher hierarchical levels of PPS. The use of hybrid dynamical approach can be well investigated for single-machine (server) processing of multiple flows. But, at an ideal flow at the input of the machines (servers) the outputs differ from the ideal input for the next machines in order. This gives difficulties for job-shop type processing. A proposal is given in the paper to overcome this difficulty by the use of *controlled buffers* proposed.

As far as we know, this is the first publication in Hungary dealing with the use of hybrid dynamical approach to FMS scheduling.

*Keywords:* hybrid dynamical approach, FMS scheduling, demand rate, overlapping production, controlled buffers.

## 1. Introduction

The good solution of the scheduling is one of the key problems of the effectiveness of FMS (Flexible Manufacturing System) actions. FMS scheduling should be contained in the integrated software system consisting of CAPP (Computer Aided Process Planning), PPS (Production Planning System), CPC (Computer Process Control) subsystems. Using the Engineering Data Base, given by CAPP, the MRP subsystem of PPS (for example SAP) determines the production tasks for a given period of production which contains: from which part (or product), what quantity is required. Tasks are formulated as production orders. Then, scheduling of tasks follows. The classic theory of scheduling is extremely well covered by literature. This is due to the importance of the problem and to the wide field where it can be applied: computer and communication networks; computer software systems; surface, water and air transport; supply systems; armed force actions organization and many others are among the similar problems. From the enormous number of references discussing the manufacturing scheduling problems as an example we cite FRENCH [1]. In the present paper we deal with the general Job Shop type problem only. The problem itself can be formulated as a general nonlinear programming, and operational research approaches like dynamical programming, branch and bound methods, integer programming, etc. can be used for finding optimal solutions. The problem that causes all the difficulties lies in large dimensions and, much more, in the difficulty to find a proper optimum criterion which would reflect all the different aspects of FMS actions goodness. Cost would be a good criterion but it is very difficult to estimate (compute) it. In fact, the problem can be approached from the point of view of multi-criterion optimization and solution methods obtained.

In most of the practical problems no optimality criterions are explicitly formulated and only 'suitable' solutions are required. So, for example, in general, a schedule is good if it is 'dense' enough. For some parts (series) no lateness is allowed. The proper schedules can easily be generated using heuristic methods based in priority indices (like: FIFO, SLACK, STP, LPT, EDD, etc. (see e.g.: FRENCH [1])). For those, the optimum criterions can be estimated. Decision about the suitability of schedules can be made using these criterion values, or other aspects. Based on the heuristic methods expert systems can be developed, genetic algorithms and other artificial intelligence approaches can also be used. Discrete system simulation (Taylor, Simple ++, etc.) can be used with embedded scheduling. Multimedia may help to increase effectiveness. Even the interconnection of PPS and CAPP is possible by the feedback of the scheduling requirement to manufacturing data optimization (see the Secondary Optimization Method proposed by the author in SOMLÓ, NAGY [2]).

So, the classically formulated scheduling problems have modern and effective solutions. With the paper PERKINS, KUMAR [3], 'Stable Distributed, Real Time Scheduling of Flexible Manufacturing Assembly, Disassembly Systems' published in 1989 a new direction of manufacturing scheduling, the hybrid dynamical approach has started. In 1993 CHASE, SERRANO and RAMAGE [4] formulated the Switched Arrival and Switched Server Problems, showing that periodic trajectories are characteristic of the behavior of the second and chaos for the first.

In the book of MATVEEV, SAVKIN [5], 'Qualitative Theory of Hybrid Dynamical Systems' research results are presented for hybrid systems, that is, for systems where a network of digital and analog devices, or a digital device interact with continuous environment. The theory of differential automatons is used.

*The goal of present paper is to show that the hybrid dynamical approach allows to organize 'overlapping' production of parts, increasing effectiveness of FMS scheduling significantly.*

In the *2nd part* of the present paper, using the literature, some introduction to the hybrid dynamical approach for the solution of manufacturing scheduling problem is given. In the *3rd part* it is shown how the required inputs, that is the demand rates can be determined using the classic formulation of jobshop type scheduling tasks. It is also shown that the hybrid dynamical approach makes possible to realize overlapping production. This highly increases the efficiency of equipment utilization. In the *4th part* a proposal is outlined to use *controlled buffers for demand rate regulation*. The problem with the use of hybrid dynamical approach is that, though, at the first machine making some part types ideal, inputs can be applied, at the consecutive machines the real outputs of previous machines are the inputs. This highly differs from the ideal. This difficulty may be solved in the proposed way. That is buffers can be controlled in such a way that these could provide ideal inputs for the consecutive machines. As it was mentioned, the proposed solution is outlined in the 4th part. The *5th part* contains conclusions.

## 2. Hybrid Dynamical Approach

### *2.1. Production of High Numbers of Parts (HNP)*

Flexible Manufacturing Systems are dedicated to process one of a kind of parts or small series in automated way. Nevertheless, some FMS-s are suitable to manufacture high number parts (HNP). By high number we understand several hundreds, several thousands or more parts (sometime less). Of course, the character of manufacturing depends not only on the number but on 'size' of parts, too (the meaning of 'size' here is the volume, the complication, etc.).

### *2.2. The Task and Database*

The task of scheduling for an FMS is usually given by the *MRP subsystem of PPS*. (Formally, we use partly the notation of FRENCH [1]):
One has:
*n jobs* $(J_1, J_2, \ldots, J_n)$, to be processed on *m machines* $(M_1, M_2, \ldots, M_m)$. By *machines* we mean an equivalent group of machines, which can be in practice one machine, too. Sometimes instead of the word machine we use *server*. For all of the jobs the number of parts $n_i$, the due date $dd_i$, the release date $r_i$ ($r_i$ – is the time at which $J_i$ becomes available for processing) are given. $T_{\text{sch}}$ is the scheduling period. The *CAPP* subsystem determines for every job $J_i$ (a job $J_i$ is $n_i$ parts of type $i$) the operation sequence $o_{ij}$, where $j = 1, 2, \ldots, j_l$ ($j_l$ – is the number of operation sequences), $o_{ij}$ – is an integer showing in which machine the given operation is performed. Very important information are the *processing times* $q_{ij}$. The processing times are determined by the operation planning subsystem of CAPP including the manufacturing data determination section. We note that the engineering database for scheduling is generated from CAPP results. But, it needs proper modification. In engineering database for scheduling it is convenient to apply $\tau_{ij} = \dfrac{q_{ij}}{n_h}$, where $n_h$ is the number of machines in the equivalent group.

Sometimes it is convenient to use:

$$\tau_{ih} = \tau_{ij}$$

or

$$\tau_{ij}^{h} = \tau_{ij},$$

where $h$ is the index of the machine on which the $j$-th operation is realized for the $i$-th part type. In the following always the most convenient form will be used hoping that it will not cause misunderstanding. In PERKINS, KUMAR [3] (unlike FRENCH [1]) revisiting of the same machine is also considered. In this case $\tau_{ih}^{(k)} = \tau_{ih}$ is used at the $k$-th visit to the same machine.

### *2.3. Hybrid Dynamical Approach to Scheduling for a Single Machine*

Following PERKINS, KUMAR [3] let us consider the processing of three high number of parts (HNP) jobs $(J_1, J_2, J_3)$, by a single machine. We consider a time period the beginning of which is later than the *release* time $r_i$ , the end of which is earlier than due date $dd_i$, and the end of scheduling period is long enough to be considered as a time section from $t = 0$ to $t = \infty$.

In traditional formulation of the tasks for scheduling, the number of parts to be produced $n_i$ during the production period is given. Hybrid dynamical approach uses the demand rate $d_i$ at the input of the machines. In this paper it is supposed that $d_i$ has a constant value. This is the number of parts to be produced during a unit time. Later, some aspects of its determination will be given.

Let be

$$u_i(t) := \langle \text{the total input (part demand) for } i\text{-th job in } [0, t] \rangle.$$

Then

$$u_i(t) = d_i t + u_i(0). \tag{1}$$

$u_i(t)$ has a real material meaning only at time instants when

$$t_k = \frac{k}{d_i} \qquad (k = 1, 2, \ldots).$$

Let us define also

$$y_i(t) := \langle \text{the total production of job } i \text{ in } t = [0, t] \rangle.$$

If

$$z_i(t) := \langle \text{the buffer level of part (job) } i \text{ at time } t \rangle,$$

then

$$z_i(t) = u_i(t) - y_i(t) = d_i t - y_i(t) + u_i(0). \tag{2}$$

$z_i(t)$ – corresponds to the error signal when using terms of classic control systems. Here the buffer is a suitable local storing device dedicated to the parts or to the machines. In fact, the task can be formulated in the following way: the rough parts are coming to the input buffer of the machine from the central buffer having infinite capacity with some given rate (corresponding to the demand rate). The input buffer content, at the same time, represents an order for processing. At a given time instant, due to the fulfilment of some conditions, the switching to produce a new part type is realized. Then, preparation for the production of the new part should be performed. This is the set-up of the equipment, transport and manipulation of parts. Some of these actions may be performed in parallel. In the following the time of the machine that cannot be used for real production, because of the above actions, will be termed as inactivity time.

The methods, based on which switching is determined are called the *switching policy*. After the *inactivity time* the production of another part type starts. The buffers can be imagined as individuals for every part type and every machine. But, in reality different part types (and even machines) can be served from the same 'physical' buffer. Let us consider a time 'window' $[t_1, t_2]$ of the process shown in *Fig. 1*. In this figure Clear the Largest Buffer Level Switching Policy is demonstrated.

A decision is made about switching wherever a buffer is emptied. At time $t_l$ parts belonging to $J_3$ are processed. Let us suppose initial conditions $y_1(t_1) = y_2(t_1) = y_3(t_1) = 0$, the $u_1(t_1), u_2(t_1), u_3(t_1)$ initial conditions are shown in *Fig. 1*. The processing rates are

$$p_{ih} = \frac{1}{\tau_{ih}}. \tag{3}$$

So, if a machine is active

$$y_i(t - t_1) = y_i(t_1) + p_{ih}(t - t_1). \tag{4}$$

Switching from $i = 3$ occurs if

$$z_3(t_{s1}) = u_3(t_{s1}) + y_3(t_{s1}) = u_3(t_1) = d_3(t_{s1} - t_1) - [y_3(t_1) + p_{3h}(t_{s1} - t_1)] = 0. \tag{5}$$

$t_{s1}$ – is the time of switching.
Then, a switch to processing another part type follows. In the given case we apply the so-called Clear the Largest Buffer Level Policy. Because $z_2(t_{s1}) > z_1(t_{s1})$, a switch to processing of part type $J_2$ is realized. Then, the manufacturing of parts of type $J_2$ is performed until the buffer content of $z_2$ becomes zero, etc.

As it was mentioned, after the switching time instant some inactivity period of the machine follows. Then instead of type $J_3$ type $J_2$ is produced. The $\delta_{32}$ – inactivity time is the maximum of different action times (set-up, transport, etc.) going on in parallel. It follows the above that at moment $t_{s2}$ a switch to processing part type $J_1$ occurs, etc. In *Fig. 1*, for the better demonstration, the proportion of the inactivity times is enlarged. In the Flexible Manufacturing Systems the proportion of inactivity time – activity time is very low. This is due to the high

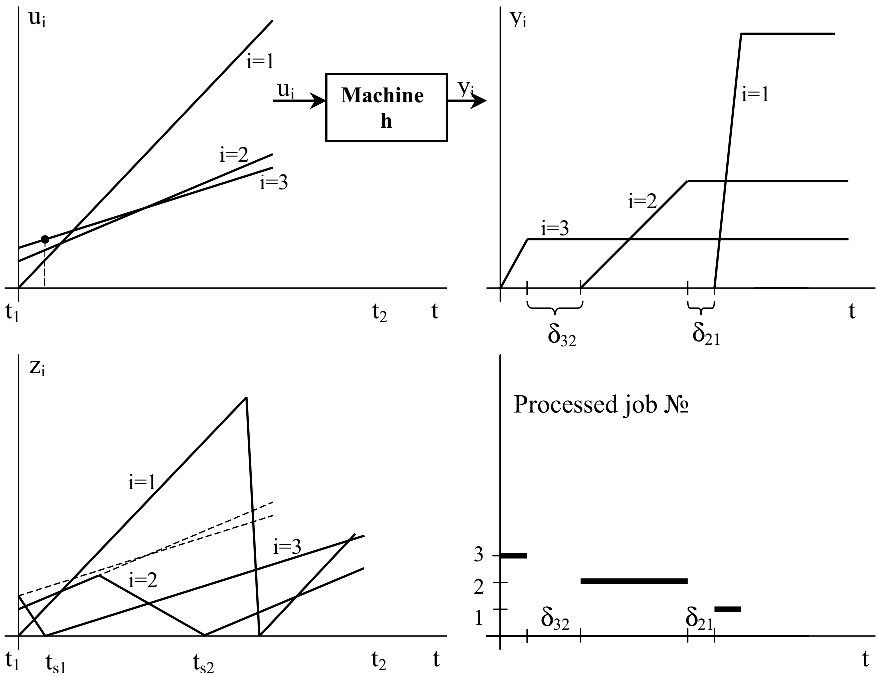


*Fig. 1.* Clear the Largest Buffer Level Switching Policy

level of automation not only of the part production but of the set-up, transport, and manipulation tasks, too.

In *Fig. 2* the picture of the change of the buffer levels, the sum of buffer levels and the input, output values are shown.

In the paper PERKINS, KUMAR [3] the definition of the stability for a given system is proposed. According to that a switching policy is *stable* if

$$\sup_t [u_i(t) - y_i(t)] = \sup_t [z_i(t)] \leq M_i < +\infty, \tag{6}$$

for all part types $(J_1, J_2, J_3)$. That is, *it is required for the stability that all of the buffer levels should be bounded.*

## *Necessary Condition of Stability*

It is clear that for any switching policy the necessary condition of stability is

$$\rho := \sum_i \tau_{ih} d_i \leq 1. \tag{7}$$

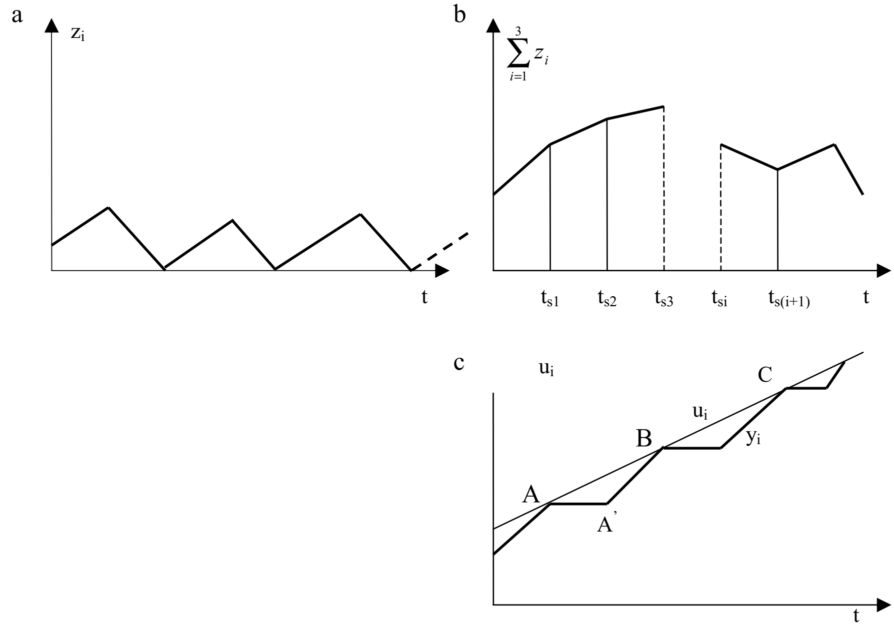


*Fig. 2.* The change of the buffer (a), summary buffer levels (b) and output values (c)

Here on the left side, for the sake of simplicity, it is not indicated that everything is related to the machine with index $h$. The necessary condition *for practical scheduling tasks* is always satisfied. Indeed, let on the machine $h$, during the $\tau_h$ time period, $n_1, n_2, \ldots, n_i, \ldots, n_n$ number of the parts be produced. Let us suppose that the production is equally distributed for the period of scheduling.
Then the following condition should be satisfied.

$$\sum_i \tau_{ih} n_i \leq \tau_h.$$

If

$$d_i = \frac{n_i}{\tau_h}.$$

One obtains the necessary condition (7). It is also clear that

$$d_i < \frac{1}{\tau_{ih}} \tag{8}$$

should also be always satisfied for any $i$.

The MRP subsystem of PPS checks the fulfilment of condition (7) and never delivers a task which does not satisfy it.

*In* PERKINS, KUMAR [3] *it is proved that the Clear the Largest Buffer Level Policy is stable.*

Let us analyze the main features of this policy. First let us deal with the switching time instants. According to PERKINS, KUMAR [3] the consecutive switching time instants are determined by

$$t_{s(j+1)} = t_{sj} + [1 - d_i \tau_{ih}]^{-1} [\delta_{ki} + z_i(t_{sj}) \tau_{ih}]. \tag{9}$$

Here: $i$ indicates to which job $J_i$ a switch is performed, $\delta_{ki}$ indicates the inactivity time necessary to change job type from $J_k$ to $J_i$. The $t_{sj}$, $t_{s(j+1)}$ are the consecutive switching time values.

Indeed, to empty $z_i(t_{sj})$ one needs time

$$t_{s(j+1)} = t_{sj} + \delta_{ki} + z_i(t_{sj}) \tau_{ih} + (t_{s(j+1)} - t_{sj}) d_i \tau_{ih}. \tag{10}$$

*Eq.* (9) follows from *Eq.* (10). An important quantity is

$$\rho = \sum_i \tau_{ih} d_i. \tag{11}$$

This quantity is called *machine load factor*. According to the necessary condition (7)

$$\rho \leq 1.$$

A component of $\rho$ is

$$\rho_i = \tau_{ih} d_i. \tag{12}$$

This is the particular machine load due to part type $i$.

It is easy to recognize that $\rho$ characterizes the part of the machining time, which is really spent to manufacturing. $(1 - \rho)$ characterizes inactivity and idle times. If $\rho$ is far less than 1 and the set-up (transport, etc.) times have low values the switching policies may result in a very good performance, when the output closely follows the input and low buffer levels are present.

In PERKINS, KUMAR [3] it is proved (as mentioned before) that the Clear the Largest Buffer Level Policy is stable. Also a formula for the upper bound of the summary buffer level was obtained. According to it, if the initial buffer levels satisfy

$$\sum_i z_i(0) \leq \frac{n\delta}{(1-\rho)} \max_i \left( \frac{\rho - \rho_i}{\tau_{ih}} \right),$$

then

$$\sum_i z_i(t) \leq \frac{\delta\rho}{\underline{\tau_{ih}}} + \frac{n\delta\overline{\tau_{ih}}}{\underline{\tau_{ih}}(1-\rho)} \max_i \left( \frac{\rho - \rho_i}{\tau_{ih}} \right), \tag{13}$$

for all $t \geq 0$. Here, $n$ is the number of part types. It is supposed that the inactivity time $\delta_{ki} = \delta$ is valid for all cases

$$\underline{\tau_{ih}} = \min_i \tau_{ih}; \qquad \overline{\tau_{ih}} = \max_i \tau_{ih}.$$

In PERKINS, KUMAR [3] a rather general case of switching policy, the Clear-a-Fraction (CAF) was analyzed. The Clear-the-Largest-Buffer-Level (CLB) Policy is a special case of CAF case Policy. In the above paper the Clear- the- Largest Work Policy was analyzed, too. Stability was proved and upper bounds were obtained.

In MATVEEV, SAVKIN [5] periodic switching policies and another version of the Clear-the-Largest-Buffer-Policy were considered. Deep analysis of the processes was performed.

Let us turn to the general aspects of the above analyzed problem. MATVEEV, SAVKIN [5] proved that this system could be described by a set of *logic-differential equations*. Indeed, let be $Q := \{q_1, q_2, q_3\}$ where $q_1, q_2, q_3$ are symbols. The discrete state $q_j$, where $j = 1, 2, 3$, corresponds to the case when the server (machine) removes (produces) work from the buffer $j$, and the discrete state variable $q(t) \in Q$ describes the state of the server (machine) at time $t$.

$x(t)$ – is the continuous state, that is the vector of the buffer contents. In the given case: $x(t) \in R^3$.

One can recognize that earlier (and later) we use for the same $z(t)$. Here, we changed to $x(t)$ to emphasize the hybrid dynamical aspect.

The logic differential equation applied to the above example, if the inactivity times can be neglected is

$$\text{If } q(t) = q_i \text{ then} \quad \begin{cases} \dot{x}_j = d_j - v_j \\ \dot{x}_i = d_i, \qquad i = 1, 2, 3;\ i \neq j. \end{cases} \tag{14}$$

Here

$$v_j = \frac{1}{\tau_{jh}}. \tag{15}$$

$q(t)$ is determined in the following way: If at any time $t_s$ one of the buffer contents becomes zero, $q(t_s) = q_k$ if $x_k(t_s) > x_i(t_s)$ for the other two buffers. The control law (14) is valid until the $j$-th buffer is emptied. Then a new section follows until the $k$-th buffer is emptied, etc.

If the inactivity period is taken into consideration a slight modification is needed.

For that case *Fig. 3*, taken from MATVEEV, SAVKIN [5], demonstrates the processes. Here, the cyclic switching policy is applied. That is, when $x_1$ becomes empty, a switch to $x_2$ occurs, etc. After the last buffer is emptied, a switch to the first comes again.

The general form of description of hybrid dynamical systems is

$$\begin{aligned} \dot{x}(t) &= f[x(t), q(t)], && x(t) \in R^n, \\ q(t_{+0}) &= \Phi[x(t), q(t)], && q(t) \in Q. \end{aligned} \tag{16}$$

Here $x(t)$ and $q(t)$ stand for the continuous and discrete states, respectively. Systems described by (16) are called multi-valued differential automatons (MDA). MATVEEV, SAVKIN [5] analyze the qualitative behavior of this kind of systems. In particular, those which describe manufacturing scheduling problems.

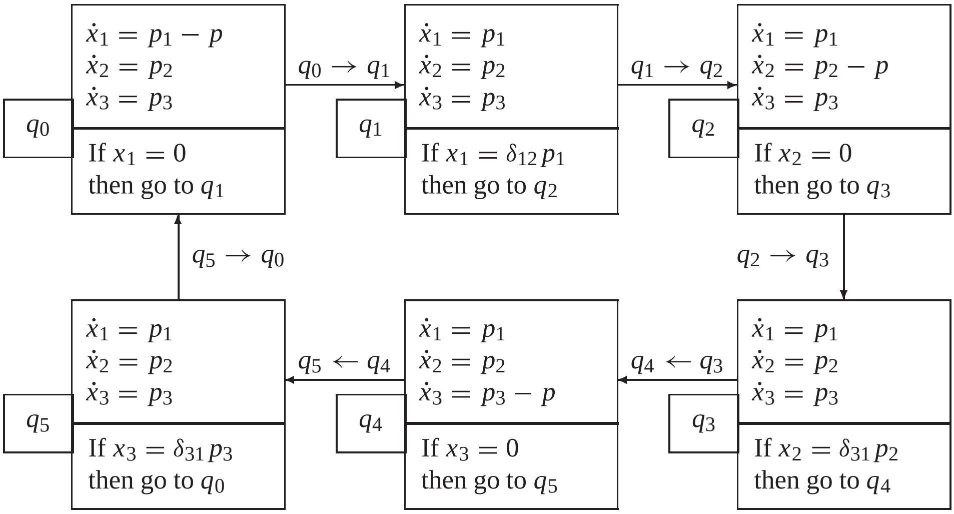


*Fig. 3.* Cyclic switching policy

### 2.4. *Hybrid Dynamical Approach to Scheduling for Job Shop Type Production*

Hybrid Dynamical Approach makes possible to schedule jobs in FMS. Usually, these problems are of job shop type. PERKINS, KUMAR [3] show that assembly and disassembly processes may also be considered.

At HNP processing the type number (the variety) of parts, usually, is not very high. If $\rho$ has a low value (e.g. $0.4 \div 0.5$) and the inactivity times allow it, the demand rates for the consecutive machines can be close to the ideal. At a closer look, it may be noticed that for any part type of any machine if the input is the ideal demand rate, the output, that is, the input for the next machine, may highly differ from the ideal. This input is of the kind as shown in *Fig.* 2*c*. In points A,B,C... the number of produced parts coincide with the required one. In section AA' no parts are delivered to the next machine. Starvation for parts may occur.

#### *Necessary Condition for Stability*

The definition of stability may be given in the same way as for Single-Machine (see: (7)). Clearly, the necessary condition for stability is

$$\rho_h := \sum_p d_p \left[\tau_{ph}^{(1)} + \tau_{ph}^{(2)} + \tau_{ph}^{(3)} + \ldots + \tau_{ph}^{(n_p,h)}\right] \leq 1, \qquad h = 1, 2, \ldots, m. \tag{17}$$

In (17), $\tau_{ph}^{(k)}$ is the processing time required by a part type with index $p$ on its $k$-th visit to the machine. $\tau_{ph}^{(n_p,h)}$ is the processing time at the last visit (let us notice that, unlike in FRENCH [1], several visits to the machines are allowed). In practical scheduling tasks (17) is always fulfilled, because the MRP subsystem of PPS never releases a task where this condition is not satisfied.

PERKINS, KUMAR [3] emphasized, that they could not find conditions providing stability when, at the first machine, the input is the ideal demand and at the consecutive machines the real output of the machine before is applied. On the contrary, the authors could construct a sample, when *dynamical instability occurred* (see also PERKINS, HUMES, KUMAR [6]). PERKINS, KUMAR [3] could show the stability of *CAF with back off switching policy*. There, the machines are considered as working in isolation (for these results for the Single-Machine case are valid). The difference is that in the case of the 'ideal buffer content', if there is starvation of part at the input of the machine, (see AA' in *Fig. 2c*) the machine is kept idle.

## 3. Hybrid Dynamical Approach to the Solution of FMS Scheduling Problems

### 3.1. Demand Rate for Single Machine Processing

Now, let us try to determine the demand rates $d_i$; $i = 1, 2, \ldots, n$, for single machine processing. It is clear that the upper limit value of demand rate is

$$\overline{\lim}\, d_i = \frac{1}{\tau_{ij}}, \qquad i = 1, 2, \ldots, n, \tag{18}$$

$j$ – is the machine group identification number.

This is due to the machining intensity constraint. As already mentioned, $\tau_{ij}$ is determined on CAPP subsystem level. In practice

$$d_i \le \frac{1}{\tau_{ij}} - (\Delta d_{up})_i \tag{19}$$

should be used, where $(\Delta d_{up})_i$ is the upper demand rate reserve value necessary for the realization. The lower limit of the demand rate is

$$\underline{\lim}\, d_i = \frac{n_i}{dd_i - r_i} \tag{20}$$

or

$$\underline{\lim}\, d_i = \frac{n_i}{T_{\text{sch}}}, \tag{21}$$

which represent the production task requirement. These relations give

$$d_i \ge \frac{n_i}{dd_i - r_i} + (\Delta d_{\text{lo}})_i \tag{22}$$

or

$$d_i \geq \frac{n_i}{T_{\mathrm{sch}}} + (\Delta d_{\mathrm{lo}})_i, \tag{23}$$

where $(\Delta d_{\mathrm{lo}})_i$ is the lower demand rate reserve value. In the following, for the sake of simplicity, only (21) and (23) will be considered. To generalize the results for the other case is a single task.

Another upper limit is given by the machine capacity constraint. It can be derived from the necessary condition of stability formulated above. Let us consider this relation in form

$$\sum_p n_p \tau_{pj} \leq T_{\mathrm{sch}}. \tag{24}$$

From that

$$\sum_p d_p \tau_{pj} \leq 1. \tag{25}$$

Now, let us determine the coefficient $\eta$ which makes (24) and (25) turn into equality.

$$\eta = \frac{T_{\mathrm{sch}}}{\sum_p n_p \tau_{pj}} = \frac{T_{\mathrm{sch}}}{t_{\Sigma j}}, \tag{26}$$

where

$$t_{\Sigma j} = \sum_p n_p \tau_{pj}$$

is the summary machining time requirement on the machine $j$.

From the above the other upper limit value is

$$\overline{\lim}\, d_i = \frac{n_i}{t_{\Sigma j}}. \tag{27}$$

Taking (18) into consideration the upper limit should be

$$\overline{\lim}\, d_i = \min\left\{\frac{1}{\tau_{ij}}, \frac{n_i}{t_{\Sigma j}}\right\}. \tag{28}$$

So, one gets the domain for demand rate

$$\frac{n_i}{T_{\mathrm{sch}}} + (\Delta d_{\mathrm{lo}})_i \leq d_i \leq \min\left\{\frac{1}{\tau_{ij}}, \frac{n_i}{t_{\Sigma j}}\right\} - (\Delta d_{\mathrm{up}})_i. \tag{29}$$

Note that the increase of $d_i$ is desirable. But, the necessary set-up, etc., times requirements constrain this increase. (This should be reflected in the value of $(\Delta d_{\mathrm{up}})_i$, too.) We think that the best aids for final decisions are the simulation studies.

Now, let us turn to the case of multi-machine job-shop type manufacturing.

### *3.2. Demand Rate for Multi-Machine Processing. Overlapping Production*

In the 1st part of this paper, formulation of the classic problems of scheduling was given. Let us suppose that for some given problems, for $(J_1, J_2, \ldots, J_n)$ part types, schedules were generated (e.g.: using a heuristic method based on FIFO, SLACK, or other priority indices). Let us suppose that we could construct and choose a 'suitable' schedule satisfying constraints and quality requirements. In *Fig.* 4 the GANTT diagrams for the $i$-th part type are given.

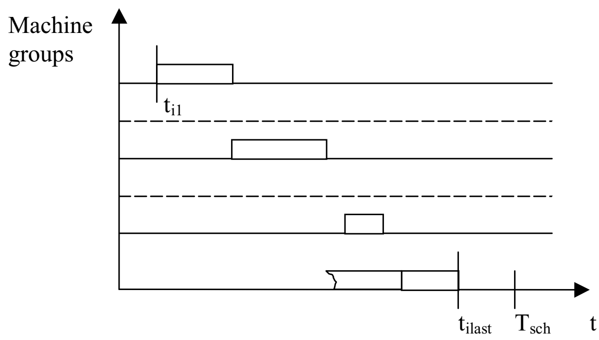


*Fig. 4.* GANTT diagrams for $i$-th part type

Let us outline the basic features of the solution. The first operation begins at $t_{i1} \geq r_i$. The end of the last operation is at $t_{il} \leq dd_i$. At the input and output of the machines, buffers of size $n_i$ are needed. In flexible manufacturing, overlapping production of parts is possible. This is due to the fact that buffers, transport and manipulation devices are automated. Set-up, transport, manipulation times are low. In *Fig.* 5 overlapping production is demonstrated for two consecutive machines. No set-up, etc. times are considered here. Clearly, the time for not overlapping production is

$$t_\Sigma = t_{j-1} + t_j = (\tau_{i(j-1)} + \tau_{ij})n_i.$$

Here and in the following, renumbering of the machine groups will be used. $\tau_{i1}, \tau_{i2}, \ldots, \tau_{il}$ are in the order of processing part-type $i$. Of course, the machining goes on the proper machine tool. In overlapping case

$$t_\Sigma = \tau_{i(j-1)} + \tau_{ij} n_i.$$

Here $\tau_{i(j-1)} < \tau_{ij}$ is supposed. The decrease of lead time is significant. The necessary buffer size decreases, too. Let us consider the general case. As it can be seen from *Fig.* 6, no set-up and/or transportation times are considered. The total time for the first 2 machines is

$$t^1_{\Sigma 2} = \tau_{i1} + \tau_{i2} n_i$$

if

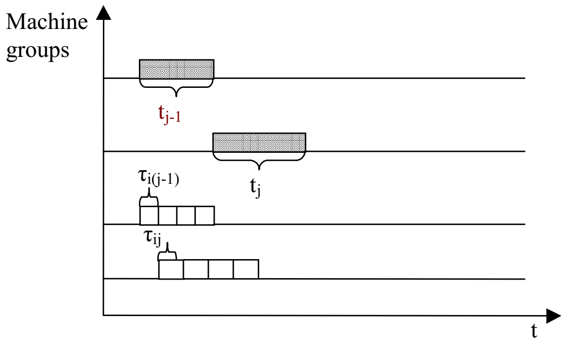


*Fig. 5.* Overlapping production

$$\tau_{i2} \geq \tau_{i1}, \tag{30}$$

$$t^2_{\Sigma 2} = \tau_{i1} n_i + \tau_{i2}$$

if

$$\tau_{i2} < \tau_{i1}.$$

From 2 to $l$ (where $l$ is the index of the last machine) for the $j$-th machine, one has

$$t^1_{\Sigma j} = t^a_{\Sigma j-1} + (\tau_{ij} - \tau_{i(j-1)}) n_i + \tau_{i(j-1)}$$

if

$$\tau_{ij} \geq \tau_{i(j-1)}, \tag{31}$$

$$t^2_{\Sigma j} = \tau^a_{\Sigma j-1} + \tau_{ij}$$

if

$$\tau_{ij} < \tau_{i(j-1)}.$$

$a$ is 1 or 2. The total time of production of parts of type $J_i$ is

$$(t_\Sigma)_i = t^a_{\Sigma l} \quad (a \text{ is 1 or 2}). \tag{32}$$

As we analyze the HNP production (and we are not interested in 'starting the line' and 'finishing production' problems) the 'cruising' regime of production can be imagined as a permanent flow. In this case the picture shown in *Fig.6* is extended to form a permanent flow.

Now, let us discuss the determination of the demand rates for multi-machine processing. It is clear that the demand rate should satisfy the following upper limit

$$(d_i)_j < \frac{1}{\tau_{ij}}, \quad \text{for all } j, \tag{33}$$

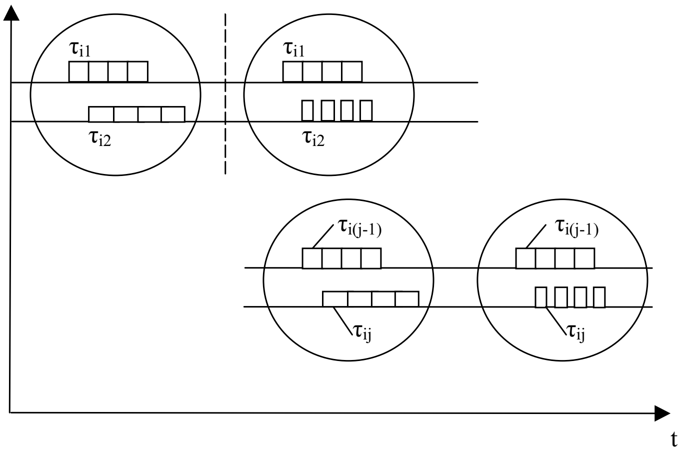


*Fig. 6.* Overlapping production in several machines

where $j$ indicates the machines part-type $i$ is processed on, and $(d_i)_j$ is the demand rate constrained by the given machine conditions.

It is supposed that the rate is constant for some part-types, irrespective of the machine it is processed on.

So, from (33) one gets

$$d_i < \frac{1}{\max\limits_j \tau_{ij}}. \tag{34}$$

Just as in the single-machine case, the lower limit of demand rate is determined by

$$\underline{\lim}\, d_i = \frac{n_i}{T_{\text{sch}}}. \tag{35}$$

Let us determine the upper limit value given by the total time necessary for the production of parts. The upper limit value of the demand rates can be determined simply based on (32), because no processing is possible for shorter time than $\tau^a_{\Sigma l}$

$$\overline{\lim}\, d_i = \frac{n_i}{\tau^a_{\Sigma l}}. \tag{36}$$

Let us simplify the above result. It is easy to recognize that

$$n_i \max_j \tau_{ij} < t^a_{\Sigma l} \leq (n_i - 1) \max_j \tau_{ij} + \sum_{j=1}^{l} \tau_{ij}\,. \tag{37}$$

Because $n_i$ is of high value it is clear that

$$t^a_{\Sigma l} \approx n_i \max_j \tau_{ij} \tag{38}$$

can be taken and so, for the demand rate one gets the upper limit given by relation (34). That is

$$d_i < \frac{1}{\max\limits_j \tau_{ij}}. \tag{39}$$

The important relation (39) reflects that the overall machining time for overlapping production gives the same upper limit value for demand rate as the machining intensity constraint.

Until now, the set-up, etc. times effects were neglected. For taking these into consideration, reserves should be introduced. For those, decrease of demand rates from the upper limit value is necessary. To produce $n_i$ parts of type $J_i$ MRP allows to use $(dd_i - r_i)$ or in analyzed case $T_{\text{sch}}$ time period. It is usually much higher than $t^a_{\Sigma l}$ in overlapped production. Of course, it should be kept in mind that in these cases several HNP series are processed in parallel.

For these cases the necessary condition for stability (17) should also be satisfied. To provide it let us determine the upper limit by

$$\overline{\lim}\, d_i = \frac{1}{\max\limits_j \tau_{ij}} \frac{t_\Sigma}{T_{\text{sch}}} \eta. \tag{40}$$

Here $\eta$ is a properly chosen coefficient. In relation (40) instead of $t^a_{\Sigma l}$ simply $t_\Sigma$ is used.

According to the necessary condition of stability (see relation (17), if only one visit to machines is considered)

$$\rho_h = \sum_{i=1}^{n} d_i \tau_{ih} \le 1, \qquad h = 1, 2, \ldots, m. \tag{41}$$

As

$$t_\Sigma \approx n_i \max_j \tau_{ij} \tag{42}$$

(see (38))

$$d_i = \frac{n_i}{T_{\text{sch}}} \eta \tag{43}$$

can be taken, and from (41)

$$\sum_{i=1}^{n} \frac{n_i}{T_{\text{sch}}} \tau_{ih} \eta \le 1, \qquad h = 1, 2, \ldots, m \tag{44}$$

is obtained. In (44)

$$\sum_{i=1}^{n} n_i \tau_{ih} = t_h, \qquad h = 1, 2, \ldots, m \tag{45}$$

is the total working time to perform the production tasks on the machine with index $h$.

From relation (44) one gets the efficiency coefficient $\eta$, which turns (44) into equality for every machine group

$$(\eta)_h = \frac{T_{\text{sch}}}{t_h}, \qquad h = 1, 2, \ldots, m. \tag{46}$$

Accordingly,

$$\eta \geq T_{\text{sch}} \frac{1}{\max\limits_h t_h}, \qquad h = 1, 2, \ldots, m. \tag{47}$$

From (40), (42) and (47)

$$\overline{\lim}\, d_i = \frac{n_i}{\max\limits_h t_h}. \tag{48}$$

Having (34) and (48), the upper limit value of demand rate is

$$\overline{\lim}\, d_i = \min \left\{ \frac{1}{\max\limits_j \tau_{ij}}, \frac{n_i}{\max\limits_h t_h} \right\}. \tag{49}$$

The lower limit value is given by (35). So

$$\frac{n_i}{T_{\text{sch}}} + (\Delta d_{\text{lo}})_i \leq d_i \leq \min \left\{ \frac{1}{\max\limits_j \tau_{ij}}, \frac{n_i}{\max\limits_h t_h} \right\} + (\Delta d_{\text{up}})_i. \tag{50}$$

Now, let us consider the demand rates for the consecutive machines. The first machine gets a $d_i$ ramp-input signal. The consecutive machines get the same with $\tau_{ij}$ time delay as shown in *Fig. 7*.

### *3.3. Time Gain by Overlapped Manufacturing*

The lead time of overlapped production is much less than that of batch processing. It is clear that the straight comparison of different values does not give a full picture because of the complex nature of the problem. Nevertheless let us compare the overall time of production using these two approaches. As shown for overlapped

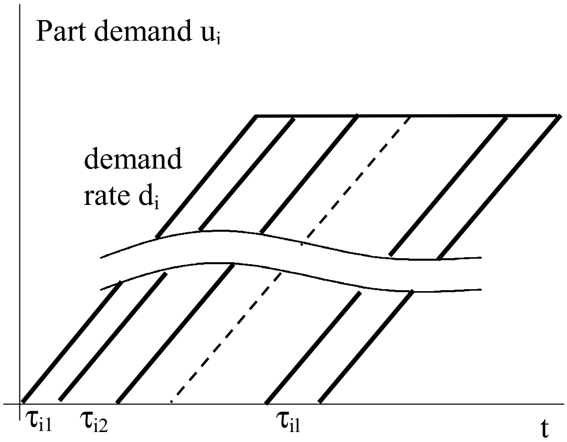


*Fig. 7.* Demand rates at consecutive machine groups

production the lead time for a type of parts can be well approximated by (see relation (38))

$$t_{\Sigma l} = n_i \max_j \tau_{ij} . \tag{51}$$

The same applies to batch processing (in the most favorable case).

$$(t_{\Sigma l})_b = \sum_{j=1}^{l} n_i \tau_{ij} , \tag{52}$$

where $l$ is the index of the last machine the part is processed on. The relation of the two values is

$$\varepsilon = \frac{(t_{\Sigma l})_b}{t_{\Sigma l}} = \frac{\sum\limits_{j=1}^{l} n_i \tau_{ij}}{n_i \max\limits_j \tau_{ij}} . \tag{53}$$

If $\max\limits_j \tau_{ij} = \tau_{ik}$, then

$$\varepsilon_i = 1 + \frac{\sum\limits_{\substack{j=1\\j\neq k}}^{l} n_i \tau_{ij}}{n_i \max\limits_j \tau_{ij}} = 1 + \frac{\sum\limits_{\substack{j=1\\j\neq k}}^{l} \tau_{ij}}{\tau_{ik}} . \tag{54}$$

The second term in equation (54) may have a significant value. So, the time gain by overlapped production may be very high.

### 3.4. Use of Secondary Optimization to Further Increase of Efficiency of FMS Scheduling

It is clear from relation (50) that in some cases the demand rates can be increased by the decrease of the value $\max_j \tau_{ij}$. The decrease of this value can be realized using the so-called Secondary Optimization Method (see: SOMLÓ, NAGY [2]; SOMLÓ, WATANABE [7]; WATANABE, SAKAMOTO [8]). It is shown in *Fig. 8* that in the manufacturing data determination of CAPP subsystem, it is possible to determine not only the optimal ($\tau_{ij\ \mathrm{opt}}$ giving minimum cost), but minimum time ($\tau_{ij\ \min}$) and minimum tool wear ($\tau_{ij\ \max}$) regimes, too. Also, any $\tau_{ijg}$ value (the index $g$ means given) can be realized in an optimal manner. How the Secondary Optimization may help to improve FMS scheduling? The following principle can be applied: $\max_j \tau_{ij}$ values can be decreased to $\tau_{ij\ \min}$ values. If $\tau_{ij\ \min}$ goes below any other $\tau_{ij}$ value this procedure can be repeated.

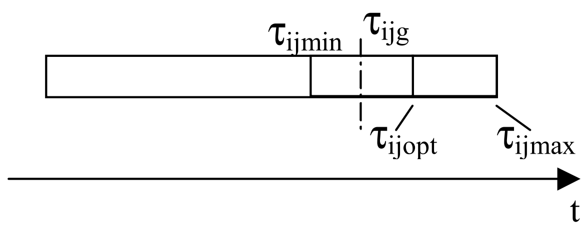


*Fig. 8*. Secondary optimization of manufacturing data

## 4. Active Buffer Policies

As mentioned, PERKINS, KUMAR [3] proposed to use distributed CAF (in particular Clear the Largest Buffer) policy with back off to provide stability of processes in general job shop type FMS scheduling. MATVEEV, SAVKIN [5] analyzed multiple server flow networks which also cover this field. They proposed to construct regularizable switched multiple server flow networks which exhibit a regular behavior.

PERKINS, HUMES, KUMAR [6] proposed to use the so called regulated buffer stabilizing techniques to reach stable performance for FMS scheduling problems.

Here we propose a different technique to be able *to use the results of hybrid dynamical approach for single machines in the case of multiple-machine, job shop type FMS scheduling.*

Let us first suppose that flexible buffers for all part types and machines of FMS are provided and all devices, including buffers, are computer controlled.

In *Fig. 9* the ideal inputs and the actual outputs for two consecutive machines are shown. Let us suppose that at time instant $T_1$ (point A) the input buffer storing type $i$ of parts on machine $(j - 1)$ is emptied. Then, a switch for serving some other parts follows. Parts of type $i$ are not produced, the input buffer $(j-1)$ grows by $z_{i(j-1)} = d_i(t - T_1)$.

Let us suppose that at time instant $T_{sw}$ the conditions to produce part type $i$ are fulfilled again. Then, after set up (etc.) period machining of part type $i$ begins until the input buffer empties (point B).

Now, let us consider the next ($j$-th) machine to produce part type $i$. At time instant $T_a$ the conditions to switch from type $r$ to type $i$ are fulfilled (Before, $T_a$ part type $r$ was produced). After $[\delta_{ri}]_j$ inactivity period at $T_{pr2}$ the machining of parts $i$ follows. But, as it can be recognized from *Fig. 9* in the period $[T_{pr2}, T_{pr1}]$ no part of type $i$ at the input buffer of the $j$-th machine is available. After $T_{pr1}$ the output of $(j-1)$ machine can cover the input of the $j$-th machine. Even in this respect there are some problems. Indeed, if $\tau_{ij} < \tau_{i(j-1)}$ the output of $(j-1)$ machine will not provide continuous production on $j$-th machine.

To avoid this kind of problems the following method, called *controlled buffer technique is proposed here*. Let us introduce the so called (virtual) auxiliary buffer as shown in *Fig. 10*.

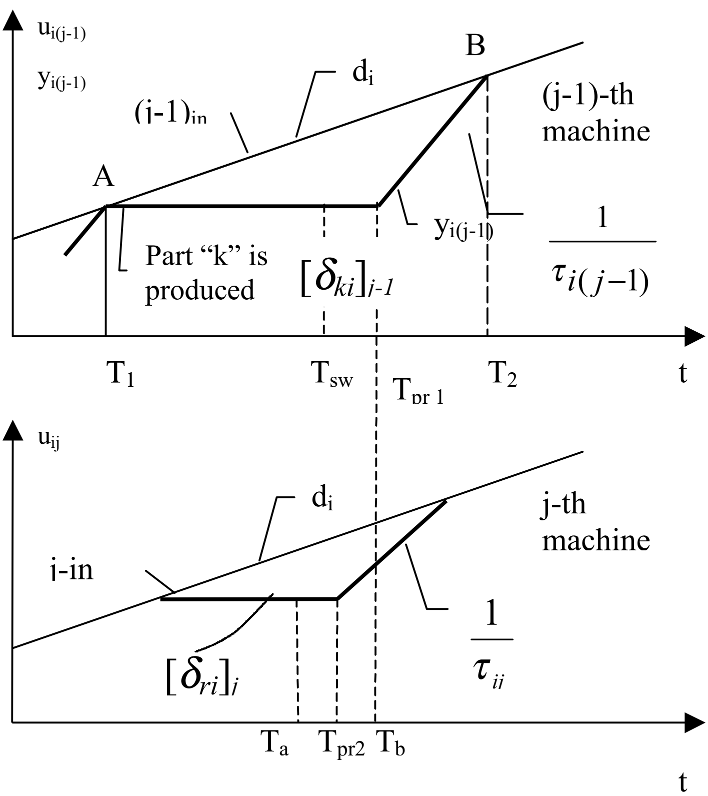


*Fig. 9.* Ideal inputs and actual outputs of machine

*The controlled buffers act as follows*. At the input buffer of $j$-th machine the ideal input ($d_i$ demand rate) and the corresponding number of parts already machined on $(j-1)$ machine should always be introduced. If the output buffer of machine $(j-1)$ is unable to cover demands, then the auxiliary buffer comes to its place and delivers the proper number of parts (not finished production). The proper content of the auxiliary buffer is provided before the regular system actions. (It is also possible to include service sections during regular actions to fulfill this task.) There is always surplus which goes to fill back the auxiliary buffer. When the part type $i$ production on machine $(j-1)$ begins, the $d_i$ demand rate on machine $j$ is covered by this production. It can be recognized that the same number of parts that was taken out of the auxiliary buffer, later, is returned from the output of the $(j-1)$-th machine. Indeed, as in an 'activity' period the $(j-1)$ machine produces exactly the number of parts between two empty buffer positions (A and B in *Fig.* 9) the number of parts will return which were taken from the auxiliary buffer.

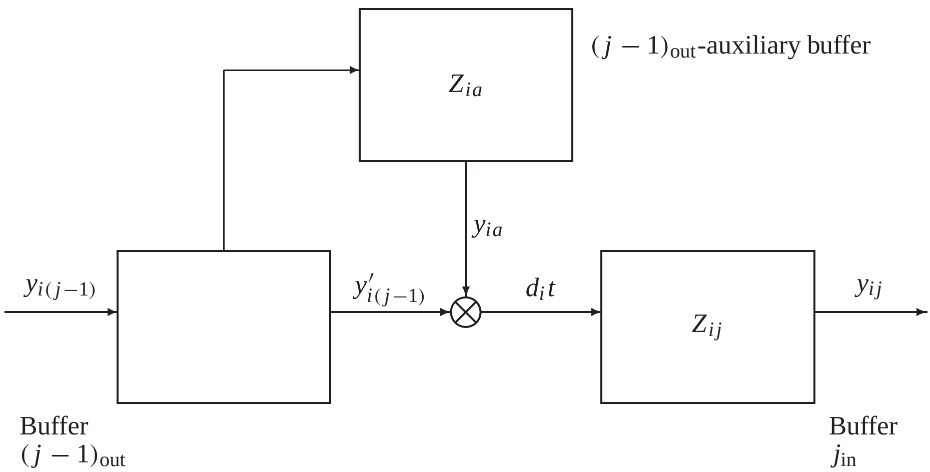


*Fig. 10*. Controlled buffer technique

We think that using the results of PERKINS, KUMAR [3]; PERKINS, HUMES, KUMAR [6] and MATVEEV, SAVKIN [5] the stability conditions of the use of the controlled buffer technique and parameter estimations (buffer sizes, etc.) for this may be obtained. Namely, if the initial auxiliary buffer content is higher than the boundary value for the input buffer content, the proposed system will work properly. The periodic motion of $(j-1)$ output, $(j-1)$ auxiliary buffer flows stay local. This does not affect the system performance where all of the $d_i$ required demand rates could be realized.

*Stopping Conditions*

Clearly, the stopping condition for delivering rough parts to the system is

$$\int_0^{T_{\text{stop}}} d_i \, dt = n_i. \tag{55}$$

Here $d_i$ is the demand rate not only for the 'regular' part of work, but also for the filling up regime of the auxiliary buffers. At the end of production of some part types clearing of the auxiliary buffers regimes is also necessary. Realization of these tasks is very simple.

*Virtual Buffers*

It is easy to recognize that the above mentioned buffers have virtual character. It means that during the physical realization on any machine there can be a unique buffer for part type $J_i$, or for several part types. The output, auxiliary and input buffers are formed virtually on the control computer without any physical motion.

*Real-Time, Distributed Control of FMS Actions*

It has been shown in the present paper that the demand rates for manufacturing HNP can be determined in a unique manner. It is well known (see: the literature) that for single-machine processing suitable switching policies based on the input buffer content can be developed which have stable performance characteristics. In the present paper, proposal was outlined to provide the given demand rate for any machine. This was based on the use of controlled buffers. Using this, the single machine approach can be extended to job shop type FMS scheduling.

In the following, the real-time, distributed control strategy will be outlined. This strategy was originally proposed in PERKINS, KUMAR [3].

*Time 0*: Set $T_0 := 0$ and $z_{ij} = z_{ij}(T_0)$ for all $i$ *and* $j$ (for simplicity, no return to the same machine is considered here).

Determine according to the switching policy for all $i$ and $j$ which part type will be served. Let it be $(J_l)_j$ for the $j$-th machine.

Now, let us consider any time when a switch of the type of parts produced on machine $j$ occurs. The time when a switch from $l$-th to $k$-th type is realized is $T_n$ (for simplicity no other indices are applied). Then after time

$$T_n + \delta_{lk},$$

part type $k$ is produced until

$$T_{n+1} := T_n + \frac{z_{kj}(T_n)\tau_{kj} + \delta_{lk}}{1 - d_k \tau_{kj}}. \tag{56}$$

(See *Eq.* (9)).

At $T_{n+1}$ a switch to the production of a new part type is realized, which part is determined by the switching policy.

When using auxiliary buffers parts of type $k$, and the other types can always be delivered to the machine at the required rate. The buffer contents can be updated. Formally,

$$\begin{aligned} &z_{kj}(T_{n+1}) \equiv 0, \\ &z_{sj}(T_{n+1}) := z_{sj}(T_n) + (T_{n+1} - T_n)d_s; \quad \text{for } s = 1, 2, \ldots, n; \text{ and } s \neq k. \end{aligned} \tag{57}$$

The switching policy is based on the buffer content only. So, the system control actions have real-time, distributed character.

### *Fluid Analogs of FMS Scheduling Problems*

In the literature (see: Chase, Serano, Ramadge [4], Matveev, Savkin [5]) fluid, tank and server analogs of work flow, buffer and machining processes in FMS are considered. This visualizes the HNP manufacturing very well. (At the same time, one should be careful because the manufacturing problems are discrete in nature, which is reflected when processes on micro level are considered).

In *Fig. 11* the scheme of fluid analogous to the use of controlled buffer technique is given. The Figure does not need too many comments. The fluids come to the tanks at given rates $(d_1, d_2, \ldots, d_n)$. The servers (the valves) remove fluid from the tanks at given rates according to some switching policies. After the switches some inactivity period occurs. The constant input flows are realized by the locally acting input, output, auxiliary buffer control from the auxiliary buffer.

When there is no output at the $y_{i(j-1)}$ flow line, the $V1$ intelligent valve provides the $d_{ij}$ flow from the auxiliary buffer. When the output $(j-1)$ is 'active', it partly covers $d_{ij}$ for the next tank and partly fills up the auxiliary tank. This action is due to intelligent valves $V1$ and $V2$. In this way it is provided that $d_{b(j-1)} = d_{bj} = d_b$; $d_{i(j-1)} = d_{ij} = d_i$; $d_{l(j-1)} = d_{lj} = d_l$, all the time.

### *Working Principle Summary*

Now, let us summarize the method proposed for on-line, real-time, distributed closed-loop control of job-shop type FMS.

Let us suppose to have $J_1, J_2, \ldots, J_n$ jobs which are served according to the conditions and requirements. The serving machines (servers) are equipped with computer controlled buffers. These can serve all inputs and outputs and form auxiliary buffers, too. The rough parts are coming to the first machine at the given demand rate determined according to the method proposed in the 3rd part of this paper. The parts to the first machine processing some parts type are coming from

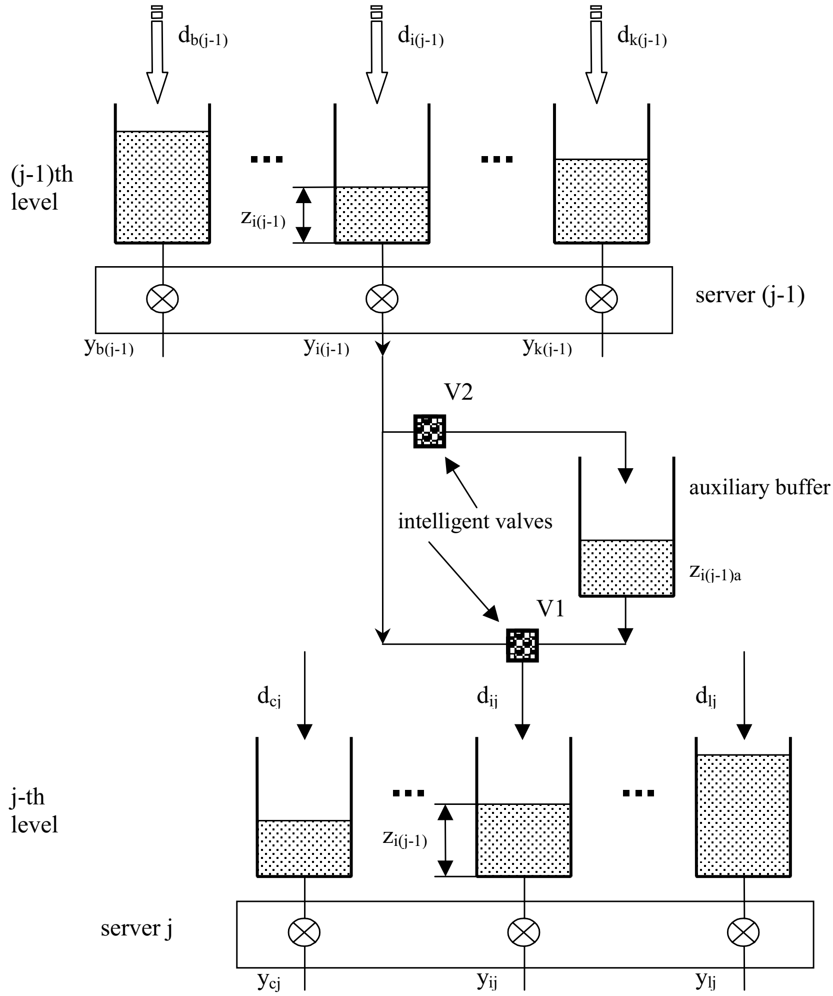


*Fig. 11.* Fluid analog of processes in FMS

the central buffer, having infinite capacity. The input buffers contents, at the same time, represent the production order, too. The same demand rate is introduced to the consecutive machines with a time delay having the value of one part production time on the machine before. These demand rates are not only introduced but realized, too, as input flows to the input buffers. This opportunity is due to the use of controlled buffers applying the auxiliary buffer local control schemes.

Then, all the machines are controlled in real-time, distributed manner according to the given switching policies.

## 5. Conclusions

The idea of hybrid dynamical approach to the solution of FMS scheduling problem gives an opportunity to significantly increase the effectiveness of utilization of the equipment of these high value systems. So, the economic effect of the use of this may be very high.

The theoretical results obtained recently concerning hybrid dynamical systems give strong basis for the solution of problems. PERKINS, KUMAR [3]; MATVEEV, SAVKIN [5] can be considered as basic works.

In the present paper an attempt is made to clear the relation of classic scheduling problems formulation and hybrid dynamical approach. It has been shown that the essence of the effect of the latter is the opportunity of overlapping production. The use of hybrid dynamical approach to a single-machine problem is well established with nice results. Not so clear is the situation when examining job-shop type multi-machine problems in FMSs (with possible assembling and disassembling processes included). In the present paper the buffer control is proposed as a solution to reduce the problems to the use of the single-machine case.

The modern simulation methods (Taylor, Simple ++, etc. as well as the continuous system simulation languages Matlab, Matrix-x, PSIM, etc.) give excellent opportunity to analyze the application problems details concerning the use of hybrid dynamical approach. These studies can lead to new results and better understanding of the problems.

## Acknowledgement

The author expresses thanks to Ph.D. students V. Lukanyin and A. Anufriev for the help in research and formulation of the results.

Special thanks are due for the advice and exceptionally high help to Dr. F. Erdélyi.

The author express thanks to the Australian Research Counsil to OTKA of the Hungarian Academy of Sciences (OTKA T 026407 and T 029072) for the support of the research.

## References

[1] FRENCH, S., *Sequencing and Scheduling: An Introduction to the Mathematics of Job-Shop*, Wiley. 1982, pp. 245.

[2] SOMLÓ, J. – NAGY, J., A New Approach to Cutting Data Optimization, *PROLAMAT'76* Stirling, Scotland (See also: North Holland 1976).

[3] PERKINS, J. R. – KUMAR, P. R., Stable, Distributed, Real-Time Scheduling of Flexible Manufacturing (Assembly) Disassembly Systems, *IEEE Transactions on Automatic Control*. **34**, No. 2. February 1989, pp. 139–148.

[4] CHASE, C. – SERANO, J. – RAMADGE, P. J., Periodicity and Chaos from Switched Flow Systems: Contrasting Examples of Discretely Controlled Continuous Systems. *IEEE Transactions on Automatic Control*, **38**, No. 1, January 1993, pp. 70–83.

[5] MATVEEV, A. S. – SAVKIN, A. V., *Qualitative Theory of Hybrid Dynamical Systems*, Birkhauser 2000, pp. 348.

[6] PERKINS, J. R. – HUMES, C. – KUMAR, P. R., Distributed Scheduling of Flexible Manufacturing Systems: Stability and Performance, *IEEE Transactions on Robotics and Automation*, **10**, No. 2, April 1994.

[7] SOMLÓ, J. – WATANABE, T., Intelligent Scheduling of FMS: The Optimization Approach, *Japan/USA Symp. On Flexible Automation*, **1**, pp. 447–454, San Francisco 1992.

[8] WATANABE, T. – SAKAMOTO, M., On-line Scheduling for Adaptive Control Machine Tools in FMS, *IFAC 9th World Congress*, Budapest 1984.