

## A NOVEL HIERARCHICAL CONTROL ARCHITECTURE FOR A CLASS OF DISCRETE-EVENT SYSTEMS

Danjing Li<sup>1</sup>, Eckart Mayer<sup>1</sup>, Jörg Raisch<sup>1,2</sup>

1. *Systems and Control Theory Group, Max-Planck Institute, Dynamics of Complex Technical Systems, Sandtorstrasse 1, D-39106 Magdeburg, Germany*

2. *Lehrstuhl für Systemtheorie technischer Prozesse, Otto-von-Guericke Universität Magdeburg, Postfach 4120, D-39016 Magdeburg, Germany*

**Abstract:** This paper proposes a novel hierarchical control architecture for a class of discrete-event systems. Under the proposed control scheme, a max-plus algebra model is introduced on the upper level to provide an optimal online plan. On the lower (implementation) level, min-plus algebra is used to solve cooperation problems between sub-plants as well as problems caused by unexpected events. A simple rail traffic example is given to show the effectiveness of the idea. *Copyright © 2004 IFAC*

**Keywords:** Discrete-event systems, max-plus algebra, min-plus algebra, rail traffic.

### 1. INTRODUCTION

In complex discrete-event systems, such as automated transportation systems, challenging issues include coordination of sub-plants and adequate handling of unexpected or uncertain events such as blockings, mechanical failure, etc. Under such circumstances, it is important that the coordination plan and its implementation strategy should adapt to the dynamic environment. In most current transportation systems, the control plan is predefined and a human operator is needed to handle unexpected situations in order to adjust the plan when unforeseen events occur. Inspired by the simplicity and the effectiveness of max-plus algebra models for discrete-event systems, a new planning and control scheme is proposed in this paper. Fig. 1 illustrates the control architecture discussed in the sequel.

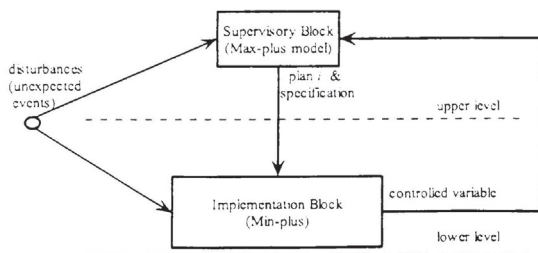


Fig. 1. Control structure

On the upper level, a max-plus algebra model is introduced to determine the time optimal plan. It also provides the event time specifications for the lower level. On the lower level, min-plus algebra not only coordinates the behaviour of sub-plants but also reduces energy consumption. In brief, the control system is a hierarchical structure: the upper level is a supervisory block, which produces the optimal plan for the lower level. The lower level is an implementation block based upon the specifications generated by the upper level. In this paper, a simple rail system is used to illustrate all concepts discussed. For this example, the plan generated by the upper control level specifies the sequence of trains and track segments where trains pass each other. The lower level generates velocity reference signals for the trains to implement the optimal plan.

This paper is structured as follows. Section 2 describes some basic ideas from max-plus algebra and min-plus algebra, which will be used in the sequel. Section 3 focuses on the supervisory block including methods for determining all feasible plans and, subsequently, for choosing the optimal plan in an online fashion. Section 4 explains how the specifications generated at the upper level can be implemented on the lower level with the help of min-plus algebra. In Section 5, a simple application example is given to illustrate the

effectiveness of the proposed approach, followed by some conclusions in Section 6.

## 2. MAX-PLUS ALGEBRA & MIN-PLUS ALGEBRA

### 2.1 Max-plus algebra

In max-plus algebra (e.g. Baccelli, *et al.*, 1992), addition  $\oplus$  is defined to be the maximum of two elements in conventional algebra, while multiplication  $\otimes$  is defined to be conventional addition on the set  $\mathbb{R}^* = \mathbb{R} \cup \{-\infty\}$ :

$$a \oplus b := \max(a, b), \quad (1)$$

$$a \otimes b := a + b. \quad (2)$$

$\varepsilon$  ( $\varepsilon = -\infty$ ) is the additive identity in the max-plus algebra.  $e$  ( $e = 0$ ) is the multiplicative identity. For a  $l \times m$  matrix  $A$  over  $\mathbb{R}^*$  and an  $m \times n$  matrix  $B$  over  $\mathbb{R}^*$ ,  $A \otimes B$  is defined by

$$(A \otimes B)_{ij} = \bigoplus_{k=1}^m A_{ik} \otimes B_{kj} = \max_k (A_{ik} + B_{kj}). \quad (3)$$

### 2.2 Max-plus models

Max-plus algebra is a convenient modelling and analysis tool not only for cyclic systems but also for systems with non-cyclic behaviour. In this contribution, we will focus on the latter.

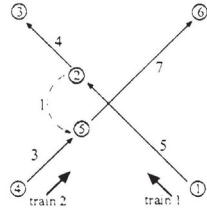


Fig. 2. A simple train-track network example

Consider a simple train-track network (Fig. 2), where two trains are travelling along their tracks. The movement of the trains is pictured as a graph with nodes and arcs. Nodes always correspond to specific trains crossing specific locations within the train-track system. For example, node 1 and node 4 represent the starting events of train 1 and train 2 respectively. Node 3 and node 6 represent the events of train 1 and train 2 arriving at their terminal stations. The arcs between the nodes represent the minimum time needed. For example, the event represented by node 2 may not happen earlier than 5 time units after the event time of node 1. The time associated with an arc is also called weight of the arc. The light grey part of Fig. 2 represents the crossing area, which only one train can occupy at any instant of time in order to ensure safe crossing. Node 5 represents the event of train 2 entering the crossing area. Node 2 represents the event of train 1 leaving the crossing area. A “control arc” (dashed arc from node 2 to node 5 in Fig. 2) is introduced. This control arc reflects a safety requirement: the earliest time at which train 2 is allowed to enter the crossing area is 1 time unit after train 1 has left it.

Let  $x_i$  denote the event time for node  $i$ . Then,  $\underline{x}_i$  is the earliest possible event time for node  $i$ , i.e. the lower bound of  $x_i$ . Train 1 and train 2 start their trips at time 0. The input to the system is a vector  $u \in \mathbb{R}^{*2}$  representing the earliest starting times:  $u = [0 \ 0]^T$ . The output of the system is defined by the earliest possible arrival times of the trains,  $Y = [\underline{x}_3 \ \underline{x}_6]^T$ . It can be conveniently computed by

$$\underline{X} = \begin{bmatrix} \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ 5 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & 4 & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & 3 & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & 7 & \varepsilon \end{bmatrix} \otimes \underline{X} \oplus \begin{bmatrix} \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & 1 & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \end{bmatrix} \otimes \underline{X} \oplus \begin{bmatrix} 0 & \varepsilon \\ \varepsilon & \varepsilon \\ \varepsilon & 0 \\ \varepsilon & \varepsilon \\ \varepsilon & \varepsilon \\ \varepsilon & \varepsilon \end{bmatrix} \otimes u, \quad (4)$$

$$Y = \begin{bmatrix} \varepsilon & \varepsilon & 0 & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 0 \end{bmatrix} \otimes \underline{X}, \quad (5)$$

where  $\underline{X} = [\underline{x}_1 \ \underline{x}_2 \ \underline{x}_3 \ \underline{x}_4 \ \underline{x}_5 \ \underline{x}_6]^T$  is the earliest possible event time vector.

The general max-plus model in this contribution has the following form:

$$\underline{X} = A_{01} \otimes \underline{X} \oplus A_{02} \otimes \underline{X} \oplus B \otimes u, \quad (6)$$

$$Y = C \otimes \underline{X}, \quad (7)$$

where the elements of  $A_{01}$  correspond to “normal travelling arcs” representing train travelling times while the elements of  $A_{02}$  correspond to the control arcs. Of course, we can combine these two matrices into one matrix:

$$A_0 = A_{01} \oplus A_{02}. \quad (8)$$

Furthermore, an explicit max-plus model can be derived from (6), (7) and (8):

$$\underline{X} = A_0^* \otimes B \otimes u, \quad (9)$$

$$Y = C \otimes A_0^* \otimes B \otimes u, \quad (10)$$

$$A_0^* = I \oplus A_0 \oplus A_0^2 \oplus \dots \oplus A_0^n \oplus A_0^{n-1} \oplus \dots, \quad (11)$$

where  $I$  is identity matrix in max-plus algebra:

$$I = \begin{bmatrix} e & & \varepsilon \\ & \ddots & \\ \varepsilon & & e \end{bmatrix}. \quad (12)$$

A path is a sequence of arcs connecting a sequence of nodes. When the initial and the final node coincide, it is called a circuit (Baccelli, *et al.*, 1992). The length of a path or a circuit is equal to the number of arcs of which it is composed. For physically meaningful systems, the graph represented by  $A_0$  does not contain circuits. For such noncyclic  $n$  by  $n$  matrices  $A_0, A_0^k, k \geq n$ , does not contribute to the sum (11), see e.g. (Baccelli, *et al.*, 1992).

With the system model (9) and (10), we can easily “simulate” the system for a given input to compute all event times  $\underline{x}_i$ . Subsequently, it will be assumed that the system input  $u$  contains lower bounds for the state vector  $X$ , i.e.  $B=I$ . The simulation results of the max-plus algebra model will be used both for determining the optimal plan and for generating the reference signals for train velocities.



### 2.3 Min-plus algebra

Min-plus algebra (Baccelli, *et al.*, 1992; Cuninghame-Green, 1979; Cuninghame-Green, 1991) is the system  $\mathbb{R}_{\min} = (\mathbb{R} \cup \{+\infty\}, \oplus, \otimes, \varepsilon, e)$ . Here  $\oplus$  refers to minimisation, i.e.

$$a \oplus b := \min(a, b), \quad (13)$$

For scalars,  $\otimes$  coincides with  $\otimes$ , i.e.

$$a \otimes b = a \otimes b := a + b. \quad (14)$$

For matrices  $A, B$ , multiplication is defined by

$$(A \otimes B)_{ij} = \oplus_{k=1}^m A_{ik} \otimes B_{kj} = \min_k (A_{ik} + B_{kj}). \quad (15)$$

The operation  $\oplus$  has the neutral element  $\varepsilon = +\infty$ . The  $r$ -fold 'product' of a matrix  $X$  with itself will be written as:

$$X^{[r]} = \underbrace{X \otimes X \otimes \dots \otimes X}_{r \text{ times}}. \quad (16)$$

Similar to the  $*$  operation in max-plus algebra, in min-plus algebra, the  $*$ ' operation can be defined:

$$A^{*'} := I' \oplus A \oplus \dots \oplus A^{[n]} \oplus A^{[n-1]} \oplus \dots, \quad (17)$$

where  $I'$  is the identity matrix in min-plus algebra:

$$I' = \begin{bmatrix} e & & \varepsilon' \\ & \ddots & \\ \varepsilon' & & e \end{bmatrix}. \quad (18)$$

Note that the following relation holds for the  $*$  and  $*$ ' operations of max-plus algebra and min-plus algebra:

$$(-A^T)^{*'} = -(A^*)^T. \quad (19)$$

where "--" is the conventional minus.

## 3. SUPERVISORY STRATEGY

As indicated in Fig. 1, to generate the optimal plan for the sequence and timing of the trains, the supervisory block for rail traffic systems does not only need information on the track topology, but also real time information on the current status of the trains. The optimal plan then specifies the future sequence of all train movements and their passing places. While determining the optimal plan is an online task, generation of all feasible plans can be done offline. In this contribution, we assume that the network is not too large, so that the complete set of feasible plans can be established. Once all feasible plans are known, it is possible to select the optimal plan online and send it to the implement block.

### 3.1 Feasible plans

A complex train-track network usually contains track segments which are used by several trains. For safety reasons only one train is allowed to occupy such a resource at any instant of time. For each resource, there are several possible train sequences, usually causing different combinatoric possibilities. Correspondingly, for the whole system consisting of a number of trains, there are several feasible plans. For example, while the control arc in Fig. 2 realises a

special feasible plan, another feasible plan is to make train 2 pass the crossing area before train 1 enters it. Of course, to model this plan there is the need of adding nodes to the graph in Fig. 2. It is straightforward to find the time optimal plan within the set of feasible plans by simulating the max-plus model for each feasible plan. In the following, it is shown by use of an example, how the set of feasible plans and their max-plus models can be determined.

For a max-plus algebra model for train movements, all trains are assumed to move at full speed if possible. Such maximum speed assumption is only applied to find the time optimal plan. It is not necessary for the implementation block. Generally we know the distances between every two sequential nodes for each train on the corresponding tracks, under the maximum velocity assumption, we can determine the transition times for arcs connecting sequential nodes. In other words, for a given train track network, the travelling arcs and therefore the entries of matrix  $A_{01}$  are already determined.

On double-line track segments, forthcoming trains will not interfere. This is different for single-line track segments, which can only be occupied by one train at a time, so control arcs are necessary to ensure safe travelling for trains on these track segments. The structure of the control arcs carries the information about the sequence of trains. The weight of a control arc determines a possible safety time that should pass between the trains' movements. If two trains compete for the travel on one single-line track, there are two possibilities: either train 1 or train 2 can go first. If a train-track network includes several single-line track segments, the different combinations of those possibilities generate different plans. Of course, some of them could be infeasible, namely cause blocking.

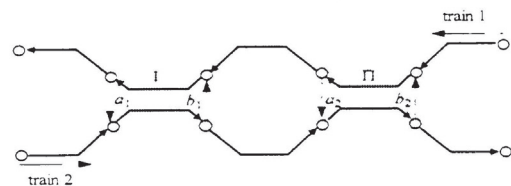


Fig. 3. A network with all possible control arcs

Fig. 3 shows the directed graph of a simple network with two single-line track segments and two trains, where train 1 moves from right to left and train 2 moves in opposite direction. For single-line segment I, the control arc labelled  $a_1$ , with  $a_1 \geq 0$  represents the fact that train 2 starts to occupy this segment  $a_1$  time units after train 1 has emptied it. The control arc labelled  $b_1$  states the reverse sequence. Accordingly, only one of those two arcs can exist in any one plan. Instead of erasing a non-existent arc from a graph, we can also label it by  $\varepsilon$ . Hence, by definition, non-existing arcs have weight  $-\infty$ . Similarly, in segment II, the arcs labelled  $a_2$  and  $b_2$  represent the two possibilities. Table 1 shows all four combinations of the selection of arcs. Among these combinations, the case  $a_1 \geq 0, b_2 \geq 0$  causes a conflict (blocking), thus this is an infeasible plan.

Table 1 Combinations of control arcs

combination	passing place of train 1 and train 2
$a_1, a_2 \geq 0$ $b_1 = b_2 = \varepsilon$	double-line track segment on the left of area I
$b_1, a_2 \geq 0$ $a_1 = b_2 = \varepsilon$	double-line track segment between area I and area II
$a_1, b_2 \geq 0$ $b_1 = a_2 = \varepsilon$	not feasible, see Fig. 4
$b_1, b_2 \geq 0$ $a_1 = a_2 = \varepsilon$	double-line track segment on the right of area II

For large train-track network systems, we need a simple criterion to determine whether a plan is infeasible. Fig. 4 shows that the conflict of combining  $a_1 \geq 0$  and  $b_2 \geq 0$  comes with a circuit in the directed graph. A circuit with a positive weight causes a contradiction, because the events on the circuit would have to “happen some time before they actually happen”. So the problem of determining infeasible plans can be solved by checking for the existence of circuits.

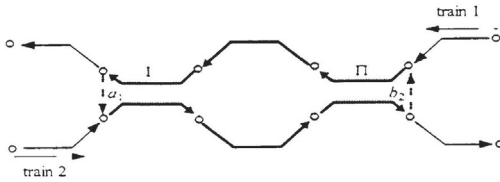


Fig. 4. Combination  $a_1 \geq 0, b_2 \geq 0$  and the circuit

Within the  $A_0$  matrix, the diagonal entry  $A_0(i, i)$  represents an arc from node  $i$  to node  $i$  itself. If  $A_0(i, i)$  is not  $\varepsilon$ , there is a circuit of length 1 containing node  $i$ . Suppose set  $\alpha(i)$  is the set of circuits of length 2 containing node  $i$ . Considering the matrix product definition under the max-plus scheme,  $A_0^2 = A_0 \otimes A_0$ ,  $A_0^2(i, i)$  represents the largest sum of weights for any circuit of length 2 containing node  $i$ . Hence, if  $A_0^2(i, i)$  is not  $\varepsilon$ , there is at least one circuit of length 2 containing node  $i$ . Similarly, if  $A_0^k(i, i)$  is not  $\varepsilon$ , there is at least one circuit of length  $k$  from node  $i$  to node  $i$ . In order to determine whether a plan is infeasible, we just need to calculate  $A_0^k, k \leq n$ , in the max-plus sense.

### 3.2 Online simulation

Based on the set of max-plus models for all feasible plans, the optimal plan with respect to a specific objective function can be easily determined by simulation. Max-plus simulation is based on the assumption that there is no unexpected event and that each train either waits for a synchronization condition to be met or otherwise moves at its maximum velocity. If any unexpected incident happens, it may affect some trains' travelling times either directly by blocking them or indirectly via synchronization with other trains. If this happens, the runtime event times do not match the event times calculated by a-priori model simulation any more. In order to do online plan optimisation, i.e. rescheduling at runtime, the state vector needs to be reinitialised based on the current status (time  $t_k$ ) of the trains for

online model simulation. The reinitialised state vector is denoted by  $\underline{X}_{in}$ . In general, it contains three different kinds of elements:

1. For an event  $j$  which already happened the reinitialised state value is the actual event time, i.e.

$$\underline{x}_{in_j}(t_k) = x_j. \quad (20)$$

2. For the next event of any train, the reinitialised state has to be calculated based on the current position of the respective train. Suppose at a time instant  $t_k$ , the distance a train has to move before it reaches the location associated with the next event ( $j$ ), is  $d_j(t_k)$ . Under the assumption of maximum velocity  $v_{max}$ , the next reinitialised state  $\underline{x}_{in_j}(t_k)$  will be:

$$\underline{x}_{in_j}(t_k) = \begin{cases} t_k + \frac{d_j(t_k)}{v_{max}} & \text{unblocked trains} \\ t_{UE\_release} + \frac{d_j(t_k)}{v_{max}} & \text{blocked trains} \end{cases} \quad (21)$$

where  $t_{UE\_release}$  is the estimated release time for the blocked train. If this time cannot be estimated beforehand, recalculation of the plan is based on the assumption of immediate release, i.e.

$$t_{UE\_release} = t_k. \quad (22)$$

3. For other events  $j$  in the future, the reinitialised states still remain undefined, i.e.

$$\underline{x}_{in_j}(t_k) = \varepsilon. \quad (23)$$

Therefore, the reinitialised state at time  $t_k$  is:

$$\underline{x}_{in_j}(t_k) = \begin{cases} x_j & \text{if } j \text{ is a past event} \\ \text{from (21)} & \text{if } j \text{ is a next event for any train} \\ \varepsilon & \text{otherwise} \end{cases} \quad (24)$$

Then, the updated vector of event times can be obtained for all feasible plans by simulation, i.e. by evaluating (9) with  $u = \underline{X}_{in}(t_k)$ . (9) and (10) are rewritten as

$$\underline{X}(t_k) = A_0^* \otimes B \otimes \underline{X}_{in}(t_k), \quad (25)$$

$$Y(t_k) = C \otimes A_0^* \otimes B \otimes \underline{X}_{in}(t_k). \quad (26)$$

### 3.3 Online plan optimisation

During runtime, the progress of the trains is observed by the supervisory block. At regular time instants (e.g. in an equidistant discrete time scheme), simulation is done for all feasible plans, comparing the values of the objective function. The objective function is evaluated from the output vector  $Y$ . An example for the objective function would be the last trains' arrival time, i.e. we want to minimise  $\max(Y_i)$ .

Note that the number of feasible plans reduces as the trains progress through the network. As described in section 3.1, building the set of feasible plans is based on the different possible combinations of control arcs at the single-line track segments. During runtime, once a train enters a single-line segment, there is no need to consider the competing control arcs on this track segment because the choice has already been made.



#### 4. IMPLEMENTATION

The output of the supervisory block generates the time optimal plan, i.e. the time specifications  $\underline{X}$  for the events. Due to the maximum speed assumption under max-plus,  $\underline{X}$  provides the earliest possible time for each event, and is therefore referred to as EPET (earliest possible event time) specifications. According to EPET, trains will always move at maximum speed, or otherwise stop to wait until the synchronization conditions are met. Clearly, this is undesirable from an energy saving point of view. An overall optimisation approach, minimising energy consumption for all trains in common, yields a large nonlinear problem and therefore is not realizable. Instead, a suboptimal more conservative approach is used here determining a velocity signal separately for each train. This is done as follows: first, as an upper bound for the event times, LNET (Latest necessary event time) is derived for each train. This is done in such a way that the trains' arrival time at the final destination according to EPET will be met. It is also ensured that LNET does not violate the EPET schedule of the other trains. In order to generate the LNET specifications, min-plus algebra, the dual system of max-plus algebra, is used.

Suppose  $Q$  is the event index set for all events corresponding to the arrival of trains at their final destinations and  $P_m$  is the index set for all events related to train  $m$ , then the LNET specifications  $\bar{X}_m$  for train  $m$  can be calculated as

$$\bar{X}_m(t_k) = (-A_0^T)^* \otimes \underline{X}R_m(t_k), \quad (27)$$

where

$$(\underline{X}R_m)_i(t_k) = \begin{cases} \underline{x}_i(t_k), & i \in Q \text{ or } i \in P_m \\ +\infty, & \text{otherwise} \end{cases} \quad (28)$$

The min-plus algebra equation (27) represents "backward simulation". Using (19), it can be rewritten as

$$\bar{X}_m(t_k) = -(A_0^*)^T \otimes \underline{X}R_m(t_k). \quad (29)$$

Now, for each train  $m$ , its EPET  $\underline{X}$  and LNET  $\bar{X}_m$  are available. Within this corridor, the velocity signal can be optimised locally for each train. If we neglect acceleration and braking effects, an energy optimal trajectory results from minimising

$$J_m = \int v_m^2 dt. \quad (30)$$

An example for an energy optimal trajectory is given in Fig. 5.

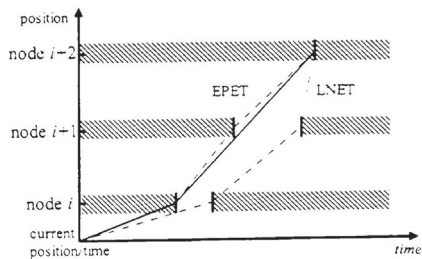


Fig. 5. Energy optimal trajectory (bold line)

#### 5. RAIL TRAFFIC CASE STUDY

Various studies concerning the modelling and control of rail traffic system have been done during the past decades. For example, recently Schutter *et al.* extended the model predictive control framework to transfer coordination in railway systems (2002).

The effectiveness of the hierarchical control architecture proposed in the previous Sections is illustrated by a small rail traffic example (Fig. 6) involving 3 trains and 3 tracks. Initially, train 1, 2 and 3 are located at the end points of 3 tracks, i.e. A, B and C, respectively. The trains move along the tracks in the directions shown in Fig. 6. In the middle of both, track AO and track CO, double-line track segments are available, which make it possible that two trains pass each other between points  $N_1$  and  $M_1$  or  $N_2$  and  $M_2$ , respectively. As in any real world application, unexpected events (e.g. blocking of the track, mechanic failure, etc.) may occur and have to be handled by the controller.

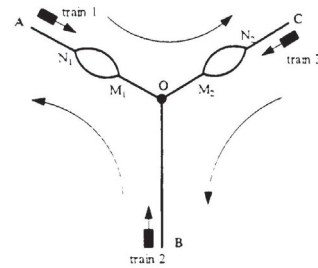


Fig. 6. A simple rail traffic network

##### 5.1. Feasible plans

First, the set of feasible plans is computed offline. Single-line track segment  $AN_1$  can either be occupied by train 1 or train 2. Nevertheless, the order is predetermined, as the track is already occupied by train 1 in the beginning. The same situation occurs for segments  $BO$  and  $CN_2$ . Therefore, although there are five single-line track segments, the order can only be chosen for segments  $OM_1$  and  $OM_2$ . The proposed approach yields a total set of 3 feasible plans pictured in Fig. 7-Fig. 9. In plan 1, train 1 occupies track segment  $OM_1$  just after train 2 empties it. Similarly, train 1 enters track segment  $OM_2$  after train 3 leaves it. In plan 2 the sequence of trains is reversed in both segments. Plan 3 is similar to plan 1 except of the trains' order on  $OM_2$ . Fig. 10 shows the directed graph for plan 1 including the control arcs needed to enforce this plan.

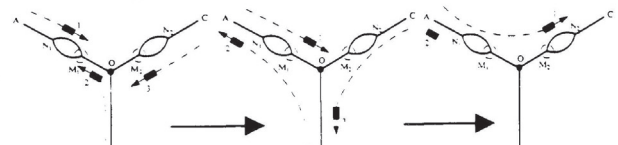


Fig. 7. Plan 1 ( $OM_1$ : train2 train1,  $OM_2$ : train3 train1)

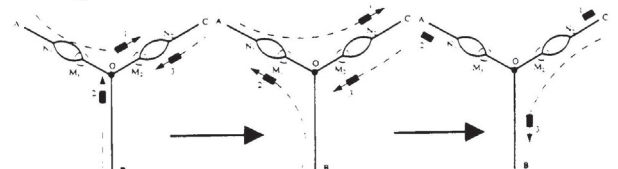


Fig. 8. Plan 2 ( $OM_1$ : train1 train2,  $OM_2$ : train1 train3)

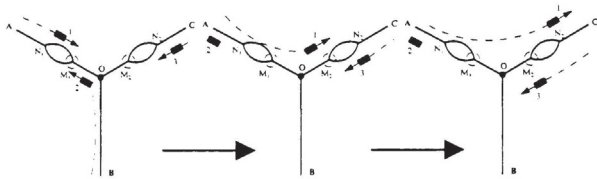


Fig. 9. Plan 3 (OM<sub>1</sub>: train2 train1, OM<sub>2</sub>: train1 train3)

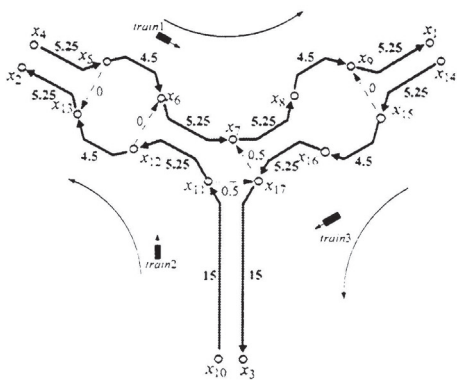


Fig. 10. Directed graph for plan 1

### 5.2. Closed loop simulation results

Plan 1 is the time optimal plan if there is no unexpected event. Fig. 11 shows the movements of the trains under the proposed hierarchical control scheme. In case of a disturbance (here train 2 is blocked on track BO), the supervisory level checks if it is necessary to change the plan. Such a situation is illustrated in Fig. 12. After train 2 has been blocked for some time, it is optimal to change from plan 1 to plan 2. This is determined in the supervisory block on the upper level and implemented by the implementation block on the lower level. This changes the operation sequence of trains: train 3 occupies track segment OM<sub>2</sub> after train 1 leaves it, train 2 enters track segment OM<sub>1</sub> after train 1 empties it. If the blocking is removed at time 12, this control policy will ensure that all trains arrive at final destinations at time 42.4. For comparison, Fig. 13 gives the result when there is no supervisory control. This implies that the original plan is maintained during the blocking and after it has been removed. Clearly it takes longer for the trains to finish their travel, as the last train arrives at 50.2. Note that in the case of blocking, the continuous adjustment of velocity signals leads to nonstraight trajectories in Fig. 12 and Fig. 13.

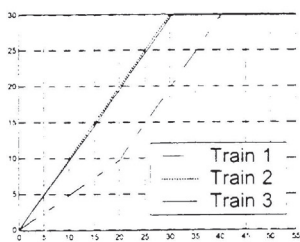


Fig. 11. Simulation result under normal situation

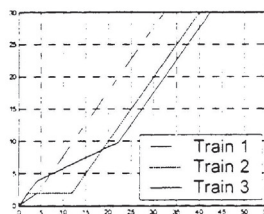


Fig. 12. Plan changing due to unexpected events

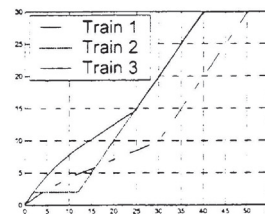


Fig. 13. Comparison study: No supervisory control

## 6. CONCLUSIONS

A novel hierarchical control architecture for a class of discrete-event systems has been proposed in this contribution. It has been applied to a rail traffic network. Based on a max-plus algebra model, the sequence of resource allocations is chosen such that minimum overall processing time is reached. An upper level supervisory block ensures the optimal sequence of train movements even under disruptive conditions. A lower level implementation block provides reference velocity signals for each train. By exploiting the remaining degrees of freedom, it reduces overall energy consumption. The implement policy is generated by use of the dual min-plus algebra model.

The method has been illustrated by means of a small rail traffic example. Simulation results show the effectiveness of the approach.

With respect to the extension of the method to large systems as well as for similar discrete-event systems in other fields, such as, flexible manufacturing or chemical process control, two aspects have not been treated in this contribution. The first is to find methods to cope with combinatoric explosion during the computation of the set of feasible plans for large systems; the second is the extension of the method to cyclically repeated processes where the advantages of the max-plus scheme can be further exploited.

## REFERENCES

- Baccelli, F., G.Cohen, G.J.Olsder, and J.P.Quadrat (1992). *Synchronization and Linearity*. John Wiley and Sons, New-York.
- Cuninghame-Green, R.A. (1991). Minimax algebra and applications. *Fuzzy Sets and Systems*, **41**, pp.251-267.
- Cuninghame-Green, R.A. (1979). *Minimax algebra*. Vol. 166 of Lecture Notes in Economics and Mathematical Systems, Springer-Verlag, Berlin.
- Schutter, B.De, T. Van den Boom, and A. Hegyi (2002). A model predictive control approach for recovery from delays in railway systems. *Transportation Research Record*, no.1793, pp.15-20.