Supervisor Control of a Flexible Manufacturing System Used in Spinning Mills

Dan Ungureanu-Anghel*, Dan Lucian Mihaescu**

* Department of Automation and Applied Informatics, Faculty of Automation and Computers, “Politehnica” University of Timisoara, Bd. Vasile Parvan No. 2, 300223 Timisoara, Romania, e-mail: dan.ungureanu@aut.upt.ro

** Department of Automation and Applied Informatics, Faculty of Automation and Computers, “Politehnica” University of Timisoara, Bd. Vasile Parvan No. 2, 300223 Timisoara, Romania, e-mail: mihaescu@aut.upt.ro

Abstract: As part of the nonlinear dynamical systems, discrete events systems (DES) are a particular category which requires for analysis the own mathematical instruments, complete different from the differential equations used at the moment in the systems analysis. In this article the authors have followed the implementation of a supervisor accordingly to a flexible system of production used in spinning mills. In this way it was followed the modelation with the help of the sequential automates of the base equipment used (utilized machines), the transport system and the operating specifications after that by successive synthesis it was obtained the desired structure of the supervisor.

Keywords: Discrete event systems, supervisor, automate, language, flexible manufacturing system

1 Introduction

The theoretical principles connected to the supervised control of the discrete events systems were established by Ramadge-Wonham [6][7]. Thus, the discrete events systems are modelated with the help of sequential automates which generates formal languages accordingly to the modelated discrete events systems.
A DES represented by an automate is a quintet of form:

$$G = (Q, \sum, \delta, i, M) \tag{1}$$

where:

Q – the set of state, Σ – the set of events, δ – transition function, i – the initial state, M – the set of marked states.

The generated language by the automate G is noted through L(G) and it is named the closed behavior of G. The marked behavior of G is described through the marked language L_m(G). The defining mode of the generated and marked languages it is to be found in [1][7]. A DES G is said to be non-blocked if: $\overline{L_m(G)} = L(G)$, where $\overline{L_m(G)}$ is prefix closure of L_m(G)[1][7]. In [1][2][3] are defined the operations of composition which are used in the supervisors synthesis namely: the product, noted through x, and parallel composition, noted through $\|$.

Definition (Supervisor) A **supervisor** is an unit which supervises and guides the behavior of a controlled discrete events subsystem [4][1].

The supervisor [1][5][7] has the role of monitoring and supervise the events generated by the system and to deactivate a series of events accordingly to some imposed laws [5][7]. The supervisor is represented by an automate defined as it's follows [1][7]:

$$S = (X, \Sigma, \xi, x_0, X_m) \quad (2)$$

The behavior in closed loop of the system is described through the automate:

$$S / G = (X \times Q, \Sigma, \xi \times \delta, (x_0, i), X_m \times M) \quad (3)$$

2 The Flexible Manufacturing System for Spinning Mills (FMSSM)

2.1 Subsystems Definition

In Figure 1 it is presented the bloc diagram of the SFFF taken into consideration.

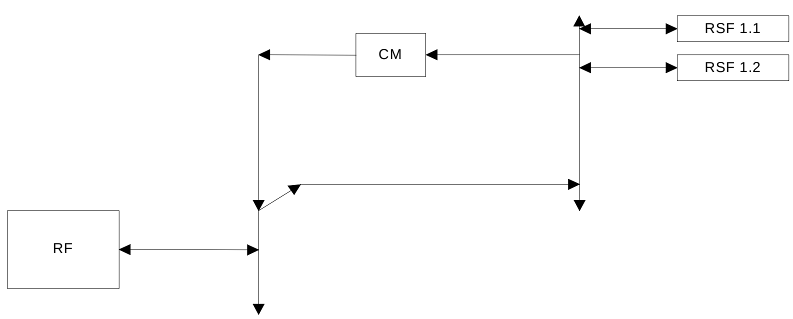

Figure 1
Bloc diagram of the considered FMSSM

The system has the following components:

- a RSF (ring spinning frame) machine;
- a CM (cleaning machine) machine;
- a RF (roving frame) machine;
- a transport system which assures the transfer between subsystems.

The normal functioning it is based on the production of coils with thick textile thread in the subsystem RF, the transfer of this coils to the RSF (through the transport system) where by processing results the thread with the desired size.

After the processing in RSF results „dirty" coils which are transported to CM where are cleaned and transferred to RF, the cycle ending here.

The functioning of the each subsystem (RSF, CM and RF) it is independently but straight connected with the transport system. In the case in which the transport system is not functional, the entire FMSSM is blocked making impossible the supply of the RSF with material which conducts to the complete block of the system.

The problems which have to be solved for the FMSSM taken into consideration are:

1 The assuring of train loading with full coils produced by RF;

2 The transfer of the train loaded in RF through a destination RSF;

3 Taking over the train by the proper RSF;

4 The transfer of the "dirty" trains from RSF to CM;

5 The cleaning of the "dirty" trains by the CM;

6 The transfer of the "clean" trains to the proper RF.

As part of this article it is not interesting the effective functioning mode of the considered machines but the way of interfacing with the transport system.

The restraints which are imposed to the considered FMSSM are:

1 Each RSF machine has two working lines which can process the same quality of the material;

2 In system can exists one or more RSF machines which works with the same quality;

3 A RF machine can produce at one time only one quality;

4 In system can exists one or more RF machines which producing the same quality.

5 A CM machine can clean many trains with different qualities but only a single train at one moment.

6 The transport system can transport one or more trains simultaneously.

Because considered system is simple in the mode of synthesis of the supervisor the quality is not taken into consideration.

Starting from those presented above, the supervisor which is desired to be elaborated has to assure: the managing of the traffic corresponding to the transport system for avoiding the collisions between trains;

2.2 The Automate of a RSF Line

In Figure 2 is presented the automate corresponding to a line from a RSF machine.

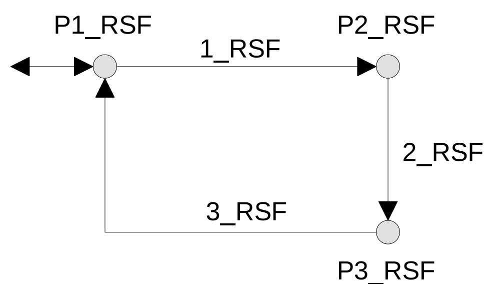

Figure 2
The automate corresponding to the working of a RSF line

Where: P1_RSF - the state of waiting a train, P2_RSF - arrived train and being in processing, P3_RSF - processed train and transported to exit, 1_RSF - the incoming train has entered completely in RSF line, 2_RSF - processed train ready to move to exit, 3_RSF - train exited from the RSF line

The automate of a RSF line is defined to be:

$RSF = (Q_{RSF}, \Sigma_{RSF}, \delta_{RSF}, i_{RSF}, M_{RSF})$ where:

1 Q_{RSF} *- The set of States:* $Q_{RSF} = \{P1\_RSF, P2\_RSF, P3\_RSF\}$

2 Σ_{RSF} *- The set of events:* $\Sigma_{RSF} = \{1\_RSF, 2\_RSF, 3\_RSF\}$

3 δ_{RSF} *- Transition function:*

$$\delta_{RSF}(P1\_RSF, 1\_RSF) = P2\_RSF$$

$$\delta_{RSF}(P2\_RSF, P2\_RSF) = P3\_RSF$$

$$\delta_{RSF}(P3\_RSF, 3\_RSF) = P1\_RSF$$

4 i_{RSF} – *Initial state:* $i_{RSF} = P1\_RSF$

5 M_{RSF} - *The set of marked states:*

$$M_{RSF} = \{P1\_RSF, P2\_RSF, P3\_RSF\}$$

2.3 The Automate of CM Machine

In Figure 3 is presented the automate corresponding to a CM machine.

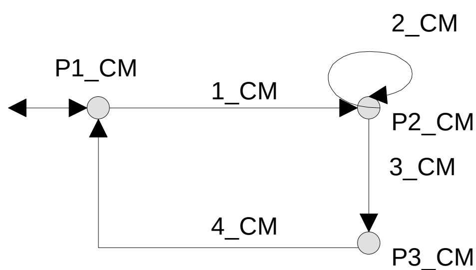

Figure 3
The automate corresponding to the working of a CM machine

Where: P1_CM - the state of waiting a train, P2_CM - arrived train and being in processing (cleaning), P3_CM - Processed(cleaned) train and transported to exit, 1_CM - the incoming train has entered completely in the cleaning machine, 2_CM- train being in the cleaning process, 3_CM - cleaned train, 4_CM - train exited completely from CM.

The automate of a CM machine is defined to be:

$$CM = (Q_{CM}, \Sigma_{CM}, \delta_{CM}, i_{CM}, M_{CM})$$

2.4 The Automate of RF Machine

In Figure 4 is presented the automate corresponding to a RF machine.

Where: P1_RF - the state of waiting a train, P2_RF - arrived train and being in processing (loading with full coils), P3_RF - processed train (loaded) an transported to exit, 1_RF - the incoming train has entered completely in the RF machine, 2_RF - train being in the processing (loading), 3_RF - processed train (loaded) ready to exit, 4_RF - train exited completely from RF.

The automate of a RF machine is: $RF = (Q_{RF}, \Sigma_{RF}, \delta_{RF}, i_{RF}, M_{RF})$.

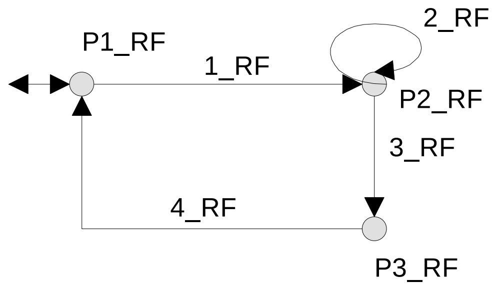

Figure 4
The automate corresponding to the working of a RF machine

2.5 The Automate of the Transport System TR

In Figure 5 is presented the automate corresponding to the transport system.

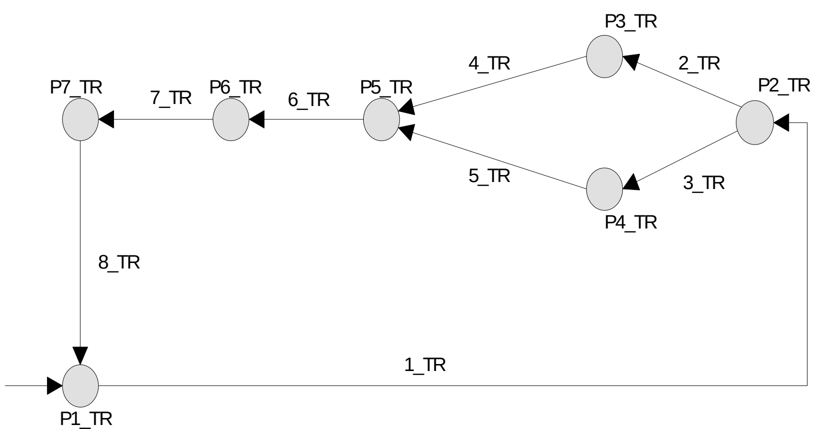

Figure 5
The automate corresponding to the working of the transport system

Where: P1_TR - The state of waiting a train from RF (the train is in RF), P2_TR - Train in transport to RSF1 or RSF2, P3_TR - The state of waiting a train from the first line RSF (the train is in RSF1), P4_TR - The state of waiting a train from the second line RSF (the train is in RSF1), P5_TR - train in transport to CM. Train arrived from RSF1 or RSF2, P6_TR - The state of waiting a train from CM (the train is in CM), P7_TR - Train in transport from CM to RF, 1_TR - The loading operation in RF is finished, 2_TR - Train arrived in line RSF1, 3_TR - Train arrived in line RSF2, 4_TR - The train left from RSF1 has arrived in CM, 5_TR - The train left from RSF2 has arrived in CM, 6_TR - Train arrived in CM, 7_TR - The cleaning operation in CM is finished, 8_TR - Train arrived in RF.

The automate of the transport system is: $TR = (Q_{TR}, \Sigma_{TR}, \delta_{TR}, i_{TR}, M_{TR})$.

2.6 FMSSM Specifications

In Figure 6 is presented the automate which synthesize the specifications imposed to the FMSSM.

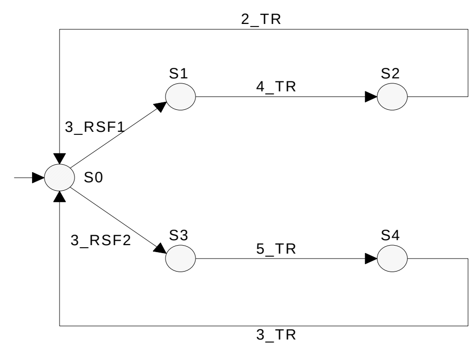

Figure 6
The automate corresponding to the synthesis of specification

Where: S0 - state of waiting a train from RSF1 or RSF2 , S1 - train in transport from RSF1. RSF2 is not able to take out a train, S2 - state of waiting a train from RF for RSF1, S3 - train in transport from RSF2. RSF1 is not able to take out a train, S4 - state of waiting a train from RF for RSF1.

The automate of the specifications is: $S = (Q_S, \Sigma_S, \delta_S, i_S, M_S)$.

3 The Synthesis of the Supervisor

The synthesis of the desired supervisor in the case of the FMSSM taken into consideration is based in the first step on parallel composition of all the automates defined above. Therefore:

$$G_{FMSSM} = RSF1 \| RSF2 \| CM \| RF \| TR \| S$$

after that the supervisor is obtained combining G_{FMSSM} with the specifications imposed to the system is validated the obtained solution.

For parallel composition it is used the associatively property of the operation. In this way the synthesis of the automates it is done in more steps.

Step 1 Parallel composition RSF1 || RSF2

In the case of parallel composition between RSF1 and RSF2 results the automate:

$$RSF1 || RSF2 = RSF12 = (Q_{RSF12}, \Sigma_{RSF12}, \delta_{RSF12}, i_{RSF12}, M_{RSF12})$$

Step 2 Parallel composition RSF1 || RSF2 || CM

The parallel composition *RSF1||RSF2||CM* is reduced in this case to the composition of the automate *RSF12 with CM*.

$$RSF12 || CM = RSF\_CM = \\ = (Q_{RSF\_CM}, \Sigma_{RSF\_CM}, \delta_{RSF\_CM}, i_{RSF\_CM}, M_{RSF\_CM})$$

Step 3 Parallel composition RSF1 || RSF2 || CM || RF

The parallel composition *RSF1||RSF2||CM||RF* is reduced in this case to the composition of the automate *RSF_CM with RF*.

$$RSF\_CM || RF = RSF\_CM\_RF = \\ = (Q_{RSF\_CM\_RF}, \Sigma_{RSF\_CM\_RF}, \delta_{RSF\_CM\_RF}, i_{RSF\_CM\_RF}, M_{RSF\_CM\_RF})$$

Step 4 Parallel composition RSF1 || RSF2 || CM || RF || TR

The parallel composition *RSF1||RSF2||CM||RF||TR* is reduced in this case to the composition of the automate *RSF_CM_RF* with *TR*.

$$RSF\_CM\_RF \parallel TR = RSF\_CM\_RF\_TR = \\ = (Q_{RSF\_CM\_RF\_TR}, \Sigma_{RSF\_CM\_RF\_TR}, \delta_{RSF\_CM\_RF\_TR}, \\ i_{RSF\_CM\_RF\_TR}, M_{RSF\_CM\_RF\_TR})$$

Step 5 Parallel composition RSF1 || RSF2 || CM || RF || TR || S

The parallel composition *RSF1||RSF2||CM||RF||TR||S* is reduced in this case to the composition of the automate *RSF_CM_RF_ TR with S*.

In the case of parallel composition between RSF_CM_RF_TR and S results the automate:

$$RSF\_CM\_RF\_TR \parallel S = G_{SFFF} =$$
$$= (Q_{RSF\_CM\_RF\_TR\|S}, \Sigma_{RSF\_CM\_RF\_TR\|S}, \delta_{RSF\_CM\_RF\_TR\|S}, i_{RSF\_CM\_RF\_TR}.i_S, M_{RSF\_CM\_RF\_TR\|S})$$

where:

1 $Q_{RSF\_CM\_RF\_TR\|S}$ - *The set of States*

$$Q_{RSF\_CM\_RF\_TR\|S} = \bigcup_{k=1,637} \{Pk_{RSF\_CM\_RF\|TR}.S0, Pk_{RSF\_CM\_RF\|TR}.S1, Pk_{RSF\_CM\_RF\|TR}.S2, Pk_{RSF\_CM\_RF\|TR}.S3, Pk_{RSF\_CM\_RF\|TR}.S4\}$$

2 $\Sigma_{RSF\_CM\_RF\_TR\|S}$ - *The set of events*

$$\Sigma_{RSF\_CM\_RF\_TR\|S} = \{1\_RSF1, 2\_RSF1, 3\_RSF1, 1\_RSF2, 2\_RSF, 3\_RSF3, 1\_CM, 2\_CM, 3\_CM, 4\_CM, 1\_RF, 2\_RF, 3\_RF, 4\_RF, 1\_TR, 2\_TR, 3\_TR, 4\_TR, 5\_TR, 6\_TR, 7\_TR\}$$

3 $\delta_{RSF\_CM\_RF\_TR\|S}$ - *Transition function*

It is considered: $s \in \Sigma_{RSF\_CM\_RF\_TR\|S}$ for which is defined the transition function accordingly to the states from RSF_CM_RF_TR in this way:

$$\forall s \in \Sigma_{RSF\_CM\_RF\_TR} \quad and \ \ i = 0,4 \quad we\ have:$$
$$\delta_{RSF\_CM\_RF\_TR\|S}(P_k\_RSF\_CM\_RF\_TR.Si\_TR, s) =$$
$$= \delta_{RSF\_CM\_RF\_TR}(P_k\_RSF\_CM\_RF, s).\delta_S(Si, s)$$

4 $i_{RSF\_CM\_RF\_TR\|S}$ - *The initial state:*

$$i_{RSF\_CM\_RF\_TR}.i_S = P1\_RSF\_CM\_RF\_TR.S0$$

5 $M_{RSF\_CM\_RF\_TR\|S}$ - *The set of marked states*

$$M_{RSF\_CM\_RF\_TR\|S} = \bigcup_{k=1,637} \{Pk_{RSF\_CM\_RF\|TR}.S0, Pk_{RSF\_CM\_RF\|TR}.S1, Pk_{RSF\_CM\_RF\|TR}.S2, Pk_{RSF\_CM\_RF\|TR}.S3, Pk_{RSF\_CM\_RF\|TR}.S4\}$$

The obtained result is even the structure of the desired supervisor.

Because of the fact that all the states are marked, the generated languages and marked of the supervisor are the same, fact which assures the non-blocking of the system as well as his controllability.

Conclusions

As part of this article it was followed the implementation of a supervisor for a FMSSM.

Therefore, were accomplished step by step the synthesis of the automates corresponding to the lines RSF1 and RSF2, synthesis of the automate RSF (resulted from the previous composition) with the automate CM, and step by step or integrated the rest of the automates RF and TR.

After obtaining the automate corresponding to the installation, it was realized the composition with the entered specifications resulting the wanted supervisor.

Although the chosen system is reduced (a RSF machine, a CM machine, a RF machine and a transport system) the way of synthesis of the supervisor it is relatively complicated because of the very large number of states.

References

[1] Christos G. Cassandras, Stephane Lafortune, Introduction to Discrete Event Systems, Kluwer Academic Publishers, 2001, Boston

[2] R. Leduc, M. Lawford, and W. Murray Wonham. Hierarchical interface-based supervisory control: AIP example. In Proc. of 39th Annual Allerton Conference on Comm.,Contr., and Comp., Oct 2001

[3] R. J. Leduc, B. A. Brandin, and W. Murray Wonham. Hierarchical interface-based nonblocking verification. In Proceedings of the Canadian Conference on Electrical and Computer Engineering, May 2000

[4] Tiberiu S. Letia, Adina M. Astilean , Sisteme cu Evenimente Discrete: modelare, analiza, sinteza si control, Editura microInformatica, 1998, Cluj

[5] Z. H. Zhang and W. Murray Wonham. STCT: an e±cient algorithm for supervisory control design. In Proc. of SCODES 2001, INRIA, Paris, July 2001

[6] W. Murray Wonham. Notes on Control of Discrete-Event Systems. Department of Electrical and Computer Engineering, University of Toronto, 2002, Notes and CTCT software can be downloaded at http://odin.control.toronto.edu/DES/

[7] P. J. G. Ramadge and W. M. Wonham, “Supervisory Control of a Class of Discrete Event Systems”, SIAM, Journal of Control and Optimization, Vol. 25, No. 1, pp. 206-230, January 1987