2010 8th IEEE International Conference on
Control and Automation
Xiamen, China, June 9-11, 2010

Modeling and Control of Flatness in Cold Rolling Mill Using Fuzzy Petri Nets

R. Dosthosseini, F. Sheikholeslam, *Member, IEEE*, J. Askari, and A. Z. Kouzani, *Member, IEEE*

Abstract— **Today, having a good flatness control in steel industry is essential to ensure an overall product quality, productivity and successful processing. Flatness error, given as difference between measured strip flatness and target curve, can be minimized by modifying roll gap with various control functions. In most practical systems, knowing the definition of the model in order to have an acceptable control is essential. In this paper, a fuzzy Petri net method for modeling and control of flatness in cold rolling mill is developed. The method combines the concepts of Petri net and fuzzy control theories. It focuses on the fuzzy decision making problems of the fuzzy rule tree structures. The method is able to detect and recover possible errors that can occur in the fuzzy rule of the knowledge-based system. The method is implemented and simulated. The results show that its error is less than that of a PI conventional controller.**

I. Introduction

To improve product quality and accuracy in steel industry, modeling and control of flatness has become important. Flatness systems have nonlinear time varying dynamics. Therefore, a proper model and also an improved control method would help decrease the error of the flatness system. To evaluate the quality of a steel strip, flatness control and the gauge accuracy are the most important parameters in cold rolling mill [1].

Measurement in flatness control systems is not identified in a quantitative form. Also, flatness control is not straightforward [1]. Therefore, measurement and manipulation of control parameters in cold rolling mill flatness control are complicated [2]. An intelligent control approach has a good potential to tackle these issues. In addition, Petri net can be employed as an alternative modeling and analysis formulation to make the system model simpler and more legible.

Conventional fuzzy control and coupled fuzzy-PID control algorithms are used to control flatness in hot strip mill [3]. It used the flatness prediction as the controlled objective in a back-propagation neural network model. The results showed that the coupled fuzzy-PID control algorithm reduced the flatness error significantly and achieved better stability at steady state.

A self-tuning PI control system was used for the flatness control of hot strip [4]. A flatness sensing system was employed to design a self-tuning PI control algorithm that improved the flatness of hot strip in finishing mill processes.

Dynamic effective matrix was used for flatness control in cold strip mills [5]. The influence of the change of parameters in rolling processes on the effective matrix was considered, and the approach was validated by industrial trials. Then, a fuzzy neural network effective matrix model was built, and then the network structure was optimized to solve the calculation problem of the dynamic effective matrix. The flatness control scheme for cold strip mills was proposed based on the dynamic effective matrix.

Fuzzy control method was employed for flatness control in cold rolling mill [6]. Strip flatness was described by an orthogonal polynomial regression based on measurement of output stress distribution. Two fuzzy logic controllers were developed: (i) skewing compensation controller to adjust the linear flatness error, and (ii) bending controller to eliminate parabolic flatness error.

A neural network-based method was realized for flatness control in cold rolling mill [1]. The ability to adapt and learn from environment, and the approximation of any non-linear function to a desired degree of accuracy are the important benefits of neural network approaches. The achieved results were compared against those of a conventional-error-decomposition function for flatness control.

In this paper, the fuzzy Petri net (FPN) method is chosen to model and control flatness in cold rolling mill. FPN has knowledge expression ability for designing dynamic knowledge expert system [7]. Generally, a FPN is based on fuzzy production rules, which have powerful modeling and analysis ability. FPN has a high ability to provide a basis for modeling and variant purposes such as knowledge representation [8], reasoning mechanisms [9], knowledge acquisition [10], etc.

Web based learning using FPN was introduced in [11], and a complete course generation platform in e-learning is developed. A fuzzy reasoning Petri net (FRPN) was developed [12] to represent decision making rules in a disassembly process. A formal reasoning algorithm based on FPRN was formed to perform fuzzy reasoning automatically allowing one to exploit maximum parallel reasoning potential embedded in the model.

Manuscript received September 28, 2009.

R. Dosthosseini is with the School of Engineering, Deakin University, Geelong, Victoria 3217, Australia (phone: +61352272183; fax: +61352272167; e-mail: rdost@deakin.edu.au & doosthoseyni@ec.iut.ac.ir), and the Electrical and Computer Engineering Department, Isfahan University of Technology, Isfahan, Iran, 84156.

F. Sheikholeslam, is with the Electrical and Computer Engineering Department, Isfahan University of Technology, Isfahan, Iran, 84156 (e-mail: sheikh@cc.iut.ac.ir).

J. Askari, is with the Electrical and Computer Engineering Department, Isfahan University of Technology, Isfahan, Iran, 84156 (e-mail: j-askari@cc.iut.ac.ir).

A. Z. Kouzani is with the School of Engineering, Deakin University, Geelong, Victoria 3217, Australia, (e-mail: kouzani@deakin.edu.au).

A fuzzy timed Petri net approach was introduced in [13] to show how the time factor can be added as an integral part of the model of transition and place. A FPN model was developed by adding and quantifying the concept of information that is affected by the aging factor. Also, a discrete-FTPN was considered as an algorithm to compute reachable states for discrete-FTPN models [14]. Properties of the continuous-FTPN model, which are used to describe the system's behavior, were presented.

In order to have a stable FPN model, necessary and sufficient conditions were introduced in [15]. Fuzzy control system modeling tools were employed and stability theorem of the fuzzy control system was developed based on the necessary and sufficient conditions under which the fuzzy control system was stable.

In this paper, a fuzzy Petri net method for modeling and control of flatness in cold rolling mill is developed. The method combines the concepts of Petri net and fuzzy control theories. It focuses on the fuzzy decision making problems of the fuzzy rule tree structures. The method is able to detect and recover possible errors that can occur in the fuzzy rule of the knowledge-based system.

This paper is organized as follows. Section II provides an overview of fuzzy control and strip shape pattern. Section III describes Petri nets. Section IV provides an overview of fuzzy Petri net. Section V explains FPN modeling and control of flatness in a cold rolling mill. Also, the results are presented and compared against those of a PI conventional control. Finally, conclusions are given in Section VI.

II. Fuzzy Control and Strip Shape Pattern

The block diagram description of the proposed control system for a cold rolling mill is shown Fig. 1.

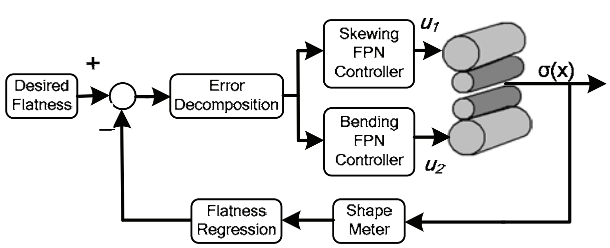

Fig. 1. Proposed system for flatness control in cold rolling mill

The flatness of the output strip can be presented by rolling mill output stress distribution across the strip width [6]. In the cold rolling mill, this stress distribution is measured by a shape meter in the width direction. The derivation equation of stress distribution using an orthogonal polynomial regression can be considered as follows [6]:

$$\Delta\sigma(x) = a_0\theta_0 + a_1\theta_1 + a_2\theta_2 \quad (1)$$

where

$$\theta_i.\theta_j = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases} \quad (2)$$

and

$$\theta_0 = b$$
$$\theta_1 = cx + d \quad (3)$$
$$\theta_2 = ex^2 + f$$

and

$$a_i = \sum_{i=0}^{2}(\Delta\sigma(x)\theta_i)/\sum_{i=0}^{2}(\theta_i.\theta_i). \quad (4)$$

Considering these equations, it is clear that the backup skewing can affect a_1, and the work roll bending can affect a_2. Also a_0 represents the stress coefficient. Let the desired output stress be:

$$\Delta\sigma^*(x) = a_0^*\theta_0 + a_1^*\theta_1 + a_2^*\theta_2 \quad (5)$$

The aim of flatness control using fuzzy control is to minimize the flatness error, $\Delta\sigma^*(x) - \Delta\sigma(x)$.

The fuzzy rule, that is used to control the flatness system, is considered as follows:

$$R = IF\ e_0\ is\ A^i\ AND\ e_1/e_2\ is\ B^i\ THEN\ u_1/u_2\ is\ C^i \quad (6)$$

where $e_0 = a_0^* - a_0$, the mean stress error signal, $e_1 = a_1^* - a_1$, the linear flatness error signal, and $e_2 = a_2^* - a_2$, the parabolic flatness error signal, are the inputs, A^i, B^i, and C^i are fuzzy sets, and u_1 and u_2 are the outputs signals for the fuzzy control system.

Input and output membership functions are shown in Fig. 2, and the associated symbols are described in Table I.

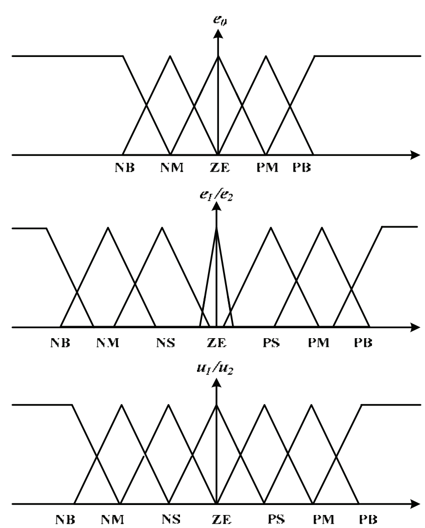

Fig. 2. Membership functions

All rules of the fuzzy skewing and bending controller are shown in Table II and III. The fuzzy rules of the decision tree structure of the flatness control for skewing controller is shown in Fig. 3. Also, the fuzzy rules of the decision tree structure of the flatness control for bending controller can be carried out in the same manner.

TABLE I
FUZZY DESCRIPTIONS SYMBOLS

Symbol	Description
NB	***Negative Big***
NM	***Negative Medium***
NS	***Negative Small***
ZE	***Zero***
PS	***Positive Small***
PM	***Positive Medium***
PB	***Positive Big***

TABLE II
RULES OF FUZZY SKEWING CONTROLLER

e_1 / e_0	NB	NM	NS	ZE	PS	PM	PB
NB	PM	PS	ZE	ZE	ZE	NS	NM
NM	PB	PM	PS	ZE	NS	NM	NB
ZE	PB	PM	PS	ZE	NS	NM	NB
PM	PB	PM	PS	ZE	NS	NM	NB
PB	PM	PS	ZE	ZE	ZE	NS	NM

TABLE III
RULES OF FUZZY BENDING CONTROLLER

e_2 / e_0	NB	NM	NS	ZE	PS	PM	PB
NB	PB	PM	ZE	ZE	ZE	NM	PB
NM	PB	PM	PS	ZE	NS	NM	PB
ZE	PM	PS	PS	ZE	NS	NS	PM
PM	PS	PS	PS	ZE	NS	NS	PM
PB	PS	PS	ZE	ZE	ZE	NS	PS

In this work, we use the basic concept of an actual fuzzy technique *Sugeno* fuzzy procedure [16]. This procedure was used in making decisions for fuzzy rules.

III. PETRI NETS

A Petri net is a mathematical modeling language for the description of discrete distributed systems [17]. A Petri net is a directed bipartite graph. It offers a graphical notation for stepwise processes that include choice, iteration, and concurrent execution. However, Petri net has an exact mathematical definition of their execution semantics, with mathematical theory for process analysis.

A Petri net is a graph that consists of *p*, *t*, *F*, *W*, *M*, and *INH* where

- $p_k \in p$ indicates the place in the net where p is all available places in the net. Inputs, outputs, and various states of the systems are defined as p_k.
- $t_j \in t$ is a transition of a system where t is a set of accepted transitions. It shows the events of the system. Each event includes some pre conditions which are represented with a place. Each transition is a set of input and output places. Places at the source of incoming arcs are called input places. On the other hand, places at the destination of outgoing arcs are called output places.

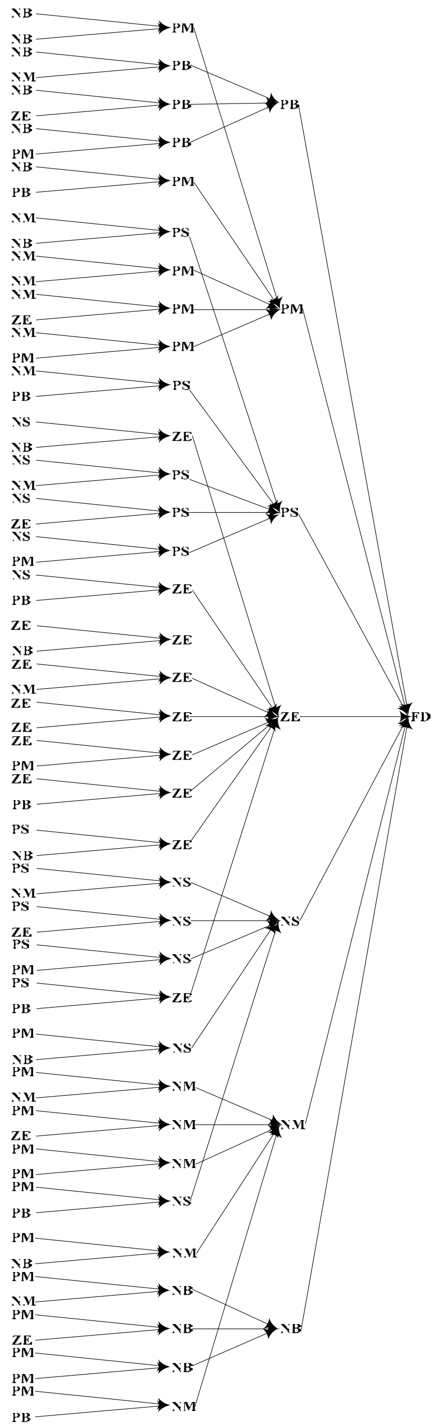

Fig. 3. Decision tree structure of skewing

- In addition, each place is a set of input and output transitions. Input transitions are located at the source of incoming arcs, and output transitions are those at the destination of outgoing arcs. A transition can be enabled if each input place of the transition is marked with a token [18]. An enabled transition fires by removing a token from each input place and adding a token to each output place.

A dead transition is one that never enabled. Also a transition without any input place is called a source transition, and a sink transition is a transition without any output place. A source transition is unconditionally enabled, and on the other hand, firing of a sink transition consumes tokens without producing any [19].

- F is the set of arcs where each arc connects a place and a transition. A weight function associated with each arc of the net is defined as $W: F \rightarrow N^+$, where N^+ is a set of non negative integers. If there is no digit on an arc, W is equal to 1. Also in a Petri net, the following conditions should be satisfied:
    1. $p \cap t = \emptyset$,
    2. $p \cup t \neq \emptyset$,
    3. $F \subseteq (p \times t) \cup (t \times p)$, the flow relation between the sets of places and transitions,
    4. $p \neq \emptyset$ and $t \neq \emptyset$, i.e. there should be at least one place and one transition.
- INH is an inhibition function, where $INH \subset (p \times t)$, represented by circle headed arcs connecting every place $p_k \in p$ contained $INH(t_j)$ to a transition $t_j \in t$. An inhibitor arc disables a transition $t_j \in t$ of a place $p_k \in p$ has W or more tokens. An inhibitor arc does not change the marking of a place $p_k \in p$ when the associated transition $t_j \in t$ fires.

When a token exists in a place, it shows the condition or the state indicating the place. A marking is an assignment of an integer to each place in the net that represents the number of tokens at that place [18]. Tokens and marking are used to record the state of a Petri net. M is a vector of order k, the number of places in the net, and m_i, ith member of M, denotes the number of tokens at place p_i.

A marking of a Petri net is reachable if there exists a series of transition firings that leads from M_0 to the marking. Therefore, a Petri net generates a graph whose nodes are reachable and whose edges represent transition firings using consecutive firing of enabled transitions.

To have a reasonable graph for a system using Petri net, terms of the system states and their changes and dynamic behavior of the system can be employed as a state or marking in a Petri net. Firing rule in a Petri net is described as follows [19]:

Firing rule: A transition t_j is said to be enabled if each input place p_k of t_j is marked with at least $W(p_k, t_j)$ tokens, where $W(p_k, t_j)$ is the weight value of the arc from p_k to t_j. On the other hand, depending on the event that actually takes place or not, an enabled transition t_j may or may not fire. After firing, $W(p_s, t_j)$ tokens from each input place p_s of t_j is removed, and then $W(t_j, p_s)$ tokens are added to each output place p_s. $W(t_j, p_s)$ is the weight value of the arc from t_j to p_s.

As an example, Fig. 4 shows the graph of a well known chemical reaction: $2H_2 + O_2 \rightarrow 2H_2O$ using Petri net.

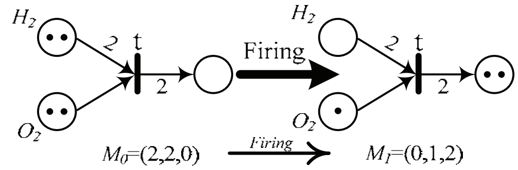

Fig. 4. Petri net model for H_2O

IV. Fuzzy Petri Nets

Generally, Petri nets cannot have sufficient power to represent and handle approximate and uncertain information [20,21]. To have fuzzy production rules, the basic concepts of fuzzy reasoning and propositional logic have been combined with the graphical representation of Petri nets. Here, fuzzy Petri net is used to model the fuzzy decision rule tree structure of practical fuzzy systems. The presented method structure consists of six parameters as follows [22]:

$$FPN = f(PN, pro, MF, FS, WFS, FM) \tag{7}$$

where *PN* is a Petri net as described in the previous section.

pro: Suppose $pro = \{pro_1, pro_2, \ldots, pro_n\}$ is a finite set of propositions that a proposition $pro_i, i = 1,2, \ldots, n$, is mapped on a place $p_k \in p$.

MF: is a membership function which describes the properties of the fuzzy set. Each membership function is described by a transition. For instance, $\mu_A(x)$ is a membership function for the proposition "$X\ is\ A$".

FS: is a firing strength function. Suppose $comp: MAX/MIN \rightarrow t$ is a fuzzy composite function that uses the MAX/MIN fuzzy operators to compose the membership grades of the required propositions. Therefore, $FS_k: comp \rightarrow t$ is a firing strength function of a rule R_k. It represents the strength of belief in a rule R_k. A larger value of FS_k increases the degree of belief for rule R_k. Let the rule R_k has a general form with two antecedent parameters and one consequent parameter such as:

$$IF\ X\ is\ A\ \ AND/OR\ \ Y\ is\ B\ THEN\ Z\ is\ C\ \ (SF_k) \tag{8}$$

where X and Y are premise parameters as input objects, Z is a consequent parameter as a decision output object, AND and OR are fuzzy operators, and A, B, and C are fuzzy sets. "$X\ is\ A$" or "Y $is\ B$" are fuzzy propositions and SF_k is the confidence value of the rule R_k. Based on the fuzzy operators OR/AND shown in the antecedent part of a rule R_k, SF_k that is associated with the conclusion of the rule R_k is measured as follows:

$$SF_k = MAX(\mu_A(x), \mu_B(y)) = \mu_A \vee \mu_B \tag{9}$$

or

$$SF_k = MIN(\mu_A(x), \mu_B(y)) = \mu_A \wedge \mu_B \tag{10}$$

WFS: is a winning rule. Suppose $R_1, R_2, \ldots, R_n$are the rules which constitute a fuzzy decision rule tree structure.
$lev_1, lev_2, \ldots, lev_j$ are the j level structures of the tree and $FS_1, FS_2, \ldots, FS_m$ are the firing strength of the m rules of lev_1. $WFS_k\colon MAX(FS_1, FS_2, \ldots, FS_m) \rightarrow t$, is the firing strength of the winning rule $R_k \rightarrow R_m$. WFS_k is used to select the winning rule R_k that has the highest confidence among all rules in a level.
FM: is a fuzzy marking of FPN that represents the distribution of tokens, fuzzy values, and over places. $FM\colon p \rightarrow N^+$ illustrates the degree of completion of the fuzzy event as a result of the processes of the fuzzy reasoning rules. In a FPN, a transition t_k is enabled at a fuzzy marking FM, if and only if $FM(p_k) \geq W(p_k, t_k)$, $INH(p_k, t_k) = \emptyset$, a token that represents the required input fuzzy variable or value must reach a place p_k to fire a transition t_k, and the fuzzy rule condition associated with each transition must be true.

V. FPN Design for Flatness Control

In FPN, rules become active when their inputs receive new values. Membership functions of the antecedent propositions of each rule are calculated to determine the confidence of each of them. Each rule uses the fuzzy operator AND to combine its antecedent membership grades. These combination processes give the firing strength value for each fuzzy rule. The firing strengths of all rules are combined by a MAX composition function to determine the highest one. A highest firing strength rule describes the winning rule from the whole rules.

The following steps are employed to design a fuzzy Petri model to control the flatness system in cold rolling mill [22]. In this work, we describe only a FPN model to minimize the linear flatness error signal, e_1. The FPN model to minimize the parabolic flatness error signal e_2 can be devised in the same manner.

Step1. Submit the input signals of the desired fuzzy rules. p_{IO1} and p_{IO2} are input places, respectively, for e_0 and e_1, and t_{IOD1} and t_{IOD2} are the input transitions. In the model shown in Fig. 5, the transitions t_{IOD1} and t_{IOD2} are used to distribute the input objects e_0 and e_1 to activate the construction step of the propositions of the first and second antecedent parts of the rules.

Step 2. Construct the antecedent propositions and calculate the membership grade for each of them. In this problem, $p_{apro1}, \ldots, p_{apro12}$ are antecedent propositions, where p_{aproi} is used to model the ith common antecedent proposition of the rules. Also, $t_{aMF1}, \ldots, t_{aMF12}$ are antecedent membership function transitions, where the transition t_{aMFi} uses the membership function of the ith proposition to compute the degree of truth of this proposition. $p_{aMG1}, \ldots, p_{aMG12}$ are antecedent membership grade places, where the token that could be shown in the place p_{aMGi} represents the value of the membership grade of the ith antecedent proposition.

Note that the number of tokens in a place p_{aMGi}, is proportional to the number of the common propositions of the first or second antecedent part of the rules.

Step 3. Calculate the firing strength for each rule. $t_{FS1}, \ldots, t_{FS35}$ are firing strength transitions, where t_{FSi} uses the fuzzy operator AND of a rule R_i to perform the MIN composition operation on the antecedent propositions of this rule. The result of this calculation represents the firing strength of the rule R_i. Also, $p_{FS1}, \ldots, p_{FS35}$ are firing strength places. The token that could mark the place p_{FSi}, represents the firing strength value of the rule R_i. As shown in Fig. 5, firing a transition t_{FSi} represents the construction of the antecedent part of the rule R_i. Since a transition t_{FSi} must fire one time, an inhibitor arc from the place p_{FSi} to transition $t_{FSi}$is attached.

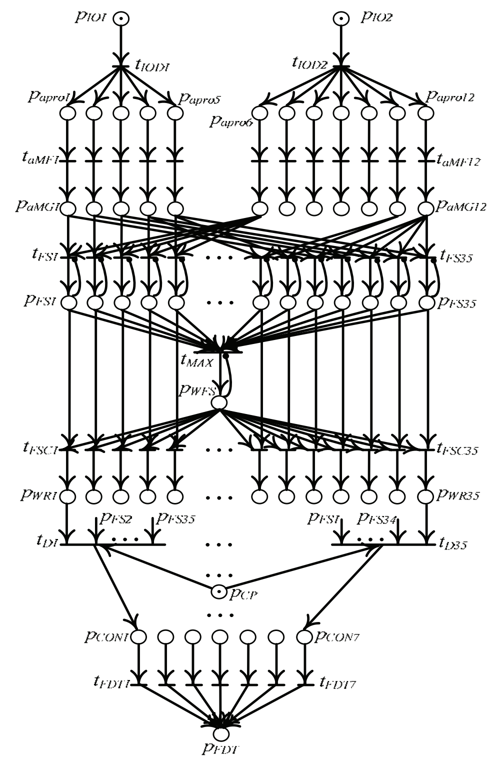

Fig. 5. FPN model of skewing controller

Step 4. Perform a MAX composition operation on the firing strengths of the activated rules to select the winning rule among the whole activated rules. In order to have the winning rule, $t_{MAX} = MAX(p_{FS1}, \ldots, p_{FS35})$ is used as a MAX composition transition. Also p_{WFS} is the winning firing strength place of the winning rule among all activated rules.

Step 5. Determine the winning rule that has the highest confidence among the activated rules. $t_{FSC1}, \ldots, t_{FSC35}$ are firing strength comparison transitions, where the transition t_{FSCi} is used to compare the firing strength FS_i of the rule R_i with the winning firing strength WFS_i. $p_{WR1}, \ldots, p_{WR35}$ are winning rile places, where the token that could be marked a place p_{WRi} denotes that the rule R_i is selected to fire.

Step 6. To determine the conclusion of the winning rule, $t_{D1}, \dots, t_{D35}$, decision transitions that are used to specify the decision of the winning rules, $R_1, \dots, R_{35}$, p_{CP}, a place to model a common consequent parameter of the rules, and $p_{CON1}, \dots, p_{CON7}$, conclusion places to describe the various decisions of the rules, are used. In addition, only one of the $p_{CON1}, \dots, p_{CON7}$ places will contain a token.

Step 7. Determine the final decision for the desired rule tree. $t_{FDT1}, \dots, t_{FDT7}$ are final decision tree transitions where the conclusion places $p_{CON1}, \dots, p_{CON7}$ use these transitions to transfer the token that represents the final decision of the tree to the final decision tree place p_{FDT} to model the final result for the entire decision tree.

Fig. 5 shows the FPN model to control the flatness system in cold rolling mill. The simulation using this approach is based on the estimated model of a sample cold rolling mill in Esfahan's Mobarakeh Steel Company Enterprise, Esfahan, Iran. The models of skewing and bending controller are estimated and used in the simulation. To have a tuned gain for controllers, the simulation is performed when actuators are held constant. Therefore, both FPN and PI conventional controllers are tuned in this way [1]. The results are shown in Fig.6.

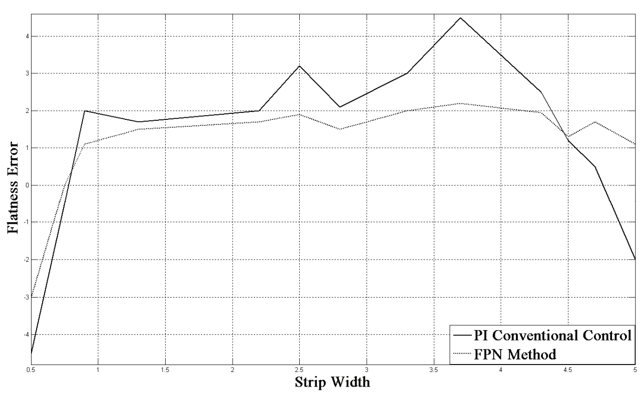

Fig. 6. Simulation results

According to Fig. 6, the results show that the proposed FPN method is effective for control and modeling of flatness control of cold rolling mill, and the error using the present method is less than PI conventional control.

VI. Conclusion

In this paper, the fuzzy Petri net approach was used for flatness control and modeling of a cold rolling mill. The method is based on fuzzy production rules, which has powerful modeling and analysis ability. FPN has a high ability to provide a basis for modeling and control. Simulation results show that the error using FPN was lower that of a PI conventional control.

Acknowledgment

The authors would like to thank Esfahan's Mobarakeh Steel Company Enterprise for their assistance with this work.

References

[1] M. S. Shim, K. Y. Lee, and D. S. Lee, "Intelligent controller design for the flatness control in a cold rolling process," *Proc. 40th IEEE Conference on Decision and Control*, Orlando, Felorida USA, 2001, pp. 2720–2725.

[2] G. T. Lee, and M. S. Shim, "Development of hardware-in-the-loop simulator for a shape control system in TCM," *Technical Research Report of POSCO Research Lab,* Korea, 1999.

[3] H. T. Zhu, Z. Y. Jiang, A. K. Tieu, and G. D. Wang, "A fuzzy algorithm for flatness control in hot strip mill," *Journal of Materials Processing Technology*, vol. 140, 2003, pp. 123–128.

[4] J. J. Choi, W. K. Hong, and J. S. Kim, "Self-tuning PI control system design for the flatness of hot strip in finishing mill processes," *KSME International Journal*, vol. 18, no. 3, 2004, pp. 379–387.

[5] L. Hongmin, H. Haitao, S. Xiuying, and J. Guangbiao, "Flatness Control Based on Dynamic Effective Matrix for Cold Strip MillsFlatness Control Based on Dynamic Effective Matrix for Cold Strip Mills," Chinese Journal of Mechanical Engineering, vol. 22, no. 2, 2009, pp. 1–10.

[6] X. J. Pan, and T. Y. Chai, "Application of Fuzzy Logic on Flatness Control of Cold Rolling Mill," *IFAC Automation in the Steel Industry*, Kyongju, Korea, 1997, pp. 39–42.

[7] Z. Zhang, S. Wang, and L. Xue, "Application of Fuzzy Petri Net Expert System in Insulator Running State," *Proceedings of the IEEE International Conference on Automation and Logistics*, Qingdao, China, 2008, pp. 2649–2654.

[8] S. M. Chen, J. S. Ke, and J. F. Chang, "Knowledge representation using fuzzy petri nets", *IEEE Trans. Knowledge. Data Engineering*, vol. 2, 1990, pp. 311–319.

[9] J. Lee, K. F. R. Liu, and W. Chiang. "A Fuzzy Petri Net-Based Expert System and Its Application to Damage Assessment of Bridges", *IEEE Transactions on Systems, Man, and Cybernetics –Part B: Cybernetics*, vol. 29, no. 3, 1999, pp.350–369.

[10] H. Scarpelli, F. Gomide, and R. R. Yager. "A Reasoning Algorithm for High-level Fuzzy Petri Nets", IEEE Trans. Fuzzy System, 4(3), pp. 282–293, 1996

[11] Y. M. Huang, J. N. Chen, T. C. Huang, Y. L. Jeng, and Y. H. Kuo, "Standardized course generation process using Dynamic Fuzzy Petri Nets," *Journal of Expert Systems with Applications*, vol. 34, 2008, pp. 72–86.

[12] M. Gao, M. C. Zhou, and Y. Tang, "Intelligent decision making in disassembly process based on fuzzy reasoning petri nets," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 34, no. 5, 2004, pp. 2029–2034.

[13] W. Pedrycz, and H. Camargo, "Fuzzy timed petri nets," *Journal of Fuzzy Sets and Systems*, vol. 140, 2003, pp. 301–330.

[14] Z. Ding, H. Bunke, O. Kipersztok, M. Schneider, and A. Kandel, "Fuzzy timed petri nets – analysis and implementation," *Journal of Mathematical and Computer Modelling*, vol. 43, 2006, pp. 385–400.

[15] R. Zhu, C. Shi, and X. Yang, "A new petri net model and stability analysis of fuzzy control system," *Proceedings of the 2009 IEEE International Conference on Networking, Sensing and Control*, Japan, 2009, pp. 113–117.

[16] J. S. R. Jang, C. T. Sun, and E. Mizutani, "Neuro fuzzy and soft computing," *Prentice Hall International*, London.

[17] C. A. Petri, and W. Reisig, "Petri net," *Scholarpedia*, vol. 3, no. 4, 2008, 6477.

[18] M. B. Dwyer, and L. A. Clarke, "A compact petri net representation and its implications for analysis," *IEEE Transaction on Software Engineering*, vol. 22, no. 11, 1993, pp. 794–811.

[19] T. Murata, "Petri nets: Properties, analysis, and applications," *Proceedings of the IEEE*, vol. 77, no. 4, 1989, pp. 541–580.

[20] T. Murata, "Temporal uncertainty and fuzzy-timing high-level Petri nets," *Proceeding 17th International Conference of Application and Theory of Petri Nets*, Osaka, Japan, 1996, pp. 11–28.

[21] T. Murata, T. Suzuki, S. M. Shatz, "Fuzzy-timing high level petri net model of a real-time network protocol," *Proceeding ITC-CSCC 96*, Seoul, Korea, 1996, pp. 1170–1173.

[22] S. M. Koriem, "A fuzzy Petri net tool for modeling and verification of knowledge-based systems," The Computer Journal, vol. 43, no. 3, 2000, pp. 206–223.