Vol.1 pp. 14-22 International Journal of Scientific Research & Engineering Technology (IJSET)

Fuzzy Particle Swarm Optimization for Manufacturing Systems

M. Bechouat[#1], S. Kahla[#2]

[#]*Labget laboratory, Department of Electrical Engineering, University of Tébessa, Algeria*

[1]mohcene.oui@gmail.com

[2]samikahla40@yahoo.com

***Abstract*— Particle Swarm Optimization (PSO) is proposed in our research to generate Fuzzy Controller, a fuzzy logic control (FLC) is proposed to control manufacturing system presented by m-machine line as an m-order state-space. As results indicated, use particle swarm optimization (PSO) method for optimizing a fuzzy logic controller (FLC) for manufacturing system is better than that of fuzzy logic control (FLC) not optimized and applying fuzzy keeping the production demand.**

***Keywords*— Manufacturing, Control, Optimization, particle swarm, Fuzzy Logic.**

I. INTRODUCTION

The knowledge base of a Fuzzy Logic Controller (*FLC*) contains two components, namely, a fuzzy rule base and a data base, both being closely related to the concept of a linguistic variable. A rule-base, i.e., a collection of fuzzy IF–THEN rules, is used to describe a particular control strategy. When the use of fuzzy controller able to command the manufactring systems remains an inconvinent, this inconvinent is represented in the random choice of membership functions (MFs). We can say that the use of fuzzy logic is based on experience. This experience has a relationship with the system.

Particle Swarm Optimization (PSO) is one of the recent evolutionary optimization methods. This technique was originally developed by Kennedy & Eberhart in order to solve problems with continuous search space [1]. PSO is based on the metaphor of social interaction and communication, such as bird flocking and fish schooling. This algorithm can be easily implemented and it is computationally inexpensive, since its memory and CPU speed requirements are low.

A production system can be viewed as a network of machines and buffers. Items receive operations at each machine and wait for the next set of operations in a buffer with finite capacity [2], [3], [4].

The overall production control system is viewed as the surplus-based system [2], i.e., the decision is based on how far the cumulative production is ahead or behind the cumulative demand.

The method proposed in this paper is based on the flow control approach. This latter uses a continuous flow approximation to model the discrete flow of parts in manufacturing systems [3]. The idea to optimize the fuzzy applied for manufacturing system is proposed in [5], [7] and [8], where it used the genetic algorithms "*GAs*", and the supervising to amelioration the results. In our research we use the particle swarm optimisation to Generate the fuzzy and comparison with the using the fuzzy alone.

The control of the manufacturing system consist four production modules where used the fuzzy alone [4], in this paper the main contribution is using the particle swarm optimization for generate the fuzzy controller apply to manufactring system consist two production modules where we use the same parameters in [4], but the demand is "*d=1*".

II. PRODUCTION MODULE

Consider a serial line with *m* machines and *m-1* buffers as depicted in Fig.1. The line produces only one part type; a constant production demand rate "*d"* is given.

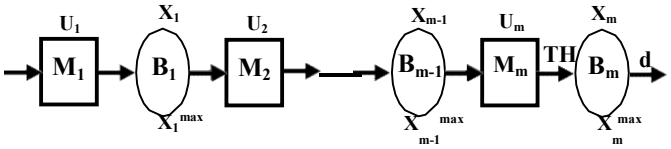

Fig.1 A serial line with *m* machines.

In our approach we are using a virtual buffer B_m at the end of the line. The level of this buffer is the cumulative difference between the actual production of machine *m* (which is the line throughput) and the target demand. Let the level of buffer *i* be the continuous state variable x_i and the production rate of machine *i* the control variable u_i (productionrate).The dynamics of the system are [3], [4]:

$$\begin{aligned} \dot{x}_i &= u_i - u_{i+1} \\ \dot{x}_m &= u_m - d \end{aligned} \quad \text{.....(1)}$$

The cumulate production is given by [4]:

$$y_i = \int_0^t u_i.dv \quad \text{.....(2)}$$

ISSN: 2356-5608

Where x_m is the level of the virtual buffer B_m. This variable can be either positive or negative, and is zero indicating the demand is fulfilled, while the level of the physical buffer, x_i, is bounded by 0 and a given buffer size x_i^{max}:

$$0 \le x_i \le x_i^{\max}$$
$$0 \le u_i \le u_i^{\max} \alpha_i \quad \text{..... (3)}$$

$$u_i^{\max} = \frac{1}{\tau_i} \quad \text{..... (4)}$$

Where:

τ_i : Maximum processing rate of M_i;

α_i: the machine state (0 or 1). In the duration of each machine state combination, the part flow is obstructed by the failed machine(s), and thus the line can be considered as the union of sub-lines, where the machines within a sub-line are all operational. Sub-lines are separated by the failed machine(s) and have no stochastic disturbances. The controllers in the optimal controller library are designed for these sub-lines.

The probability of the machine sate between t and $t+dt$ is given by geometrically distribution [4]:

TABLE I
MACHINE STATE [4]

$\alpha_i(t)$	$\alpha_i(t+dt)$	Prob
0	0	$1-\mu_i$
0	1	μ_i
1	0	λ_i
1	1	$1-\lambda_i$

Where:

μ_i: Repair rate of M_i;

λ_i: Failure rate of M_i.

III. DISTRIBUTED FUZZY SHEDULING

In fuzzy logic controllers (FLC), the control policy is described by linguistic IF-THEN rules, which model the relationship between control inputs and outputs with appropriate mathematical representation. A rule antecedent (IF-part) describes conditions under which the rule is applicable and forms the composition of the inputs. The consequent (THEN-part) gives the response or conclusion that should be taken under these conditions [5].

A three-input (antecedent) rule of the Sugeno type has the form [4], [6]:

IF x_i is X_1 and x_l is X_2 and s_i is S then $r_i = R$

Where:

x_i, x_l are respectively the levels of the upstream and downstream buffers of Mi;

X_1, X2 and S are linguistic terms associated to the inputs variables (x_i, x_l and s_i);

s_i is the given by:

$$s_i = y_i - d \quad \text{..... (5)}$$

R is the crisp value associated to the output variable r_i.

The output gain generated by the fuzzy controller $O < r_i < 1$ gives the production rate according to the specified maximum capacity of Mi:

$$R = \begin{cases} 0 & if\ \alpha = 0 \\ \phi(x_i, x_l, s_i) & if\ \alpha = 1 \end{cases} \quad \text{..... (6)}$$

Where $\phi(x_i, x_l, s_i)$ is the value given by the sate of the buffer and the surplus [4].

The relationship between the production rate and fuzzy output "r_i" is:

$$u_i = \frac{r_i}{\tau_i} \quad \text{..... (7)}$$

Policy control by hedging point shows that for a production module "**MP**" taken in isolation and state machine operating ($\alpha_i=1$) there is a critical value "z_i",for which the control law given by the following expression [2], [4]:

$$r_i = \begin{cases} 0 & ,if\ z_i > 0 \\ d \bullet \tau_i & ,if\ z_i = 0 \\ 1 & ,if\ z_i < 0 \end{cases} \quad \text{..... (8)}$$

Fig.2 shows the structure of the local fuzzy controller for the single machine Mi. Next, this strategy is extended for controlling general manufacturing systems.

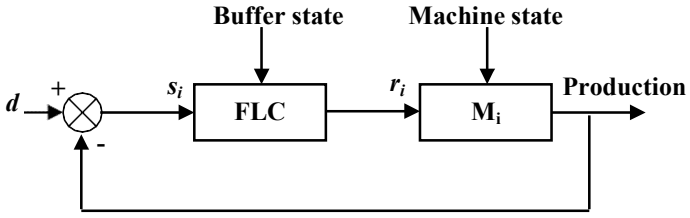

Fig.2 Local fuzzy controller for a single machine.

ISSN: 2356-5608

IV. EXEMLPE OF LINE TRANSFORMATION

Consider a serial line with *2* machines and *1* buffer as depicted in Fig.3. The line produces only one part type; constant productions demand rate "*d=1*".

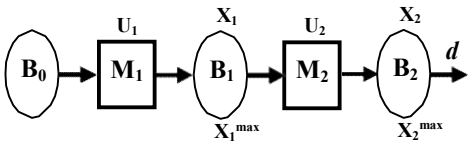

Fig.3 A serial line with *2* machines.

The system is given by:

$$\begin{cases} \dot{x}(t) = \dot{x}_1(t) = [1 \; -1] \begin{bmatrix} u_1(t) \\ u(t) \end{bmatrix} = B \cdot u(t) \\ y(t) = \int_0^t u(v)dv \end{cases} \quad (9)$$

The proposed Fuzzy Logic Control has been modelled and simulated using **MATLAB/Simulink**. Figure 4 shows our developed ***Simulink*** model, FLC and manufacturing system, as depicted in Fig.4. The specifications of manufacturing system module used in this simulation are shown in TABLE II.

TABLE II
THE SPECIFICATION OF MANUFACTURING SYSTEM MODULE USED IN THE SIMULATION [4]

Production module	X^{maxi}	μ_i	λ_i	τ_i	Z_i
MP1	5	0.5	0.3	0.5	3.96
MP2	100	0.2	0.05	0.3	2.56

Fig.4 shows the model of the manufacturing system in MATLAB/Simulink.

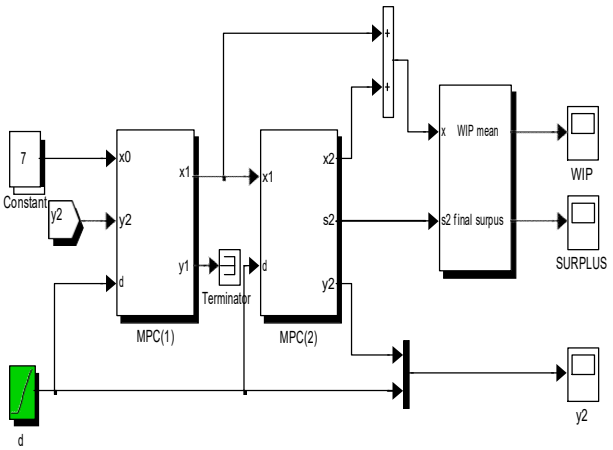

Fig.4 Model of the developed manufacturing System in MATLAB/Simulink.

The knowledge base defining the rules for the desired relationship is between the input and output variables in terms of the membership functions illustrated in TABLE III. The control rules are evaluated by an inference mechanism [4].

TABLE III
FLC RULES [4]

x_i	x_l	s	Φ
V	ANY	ANY	**0**
ANY	s	ANY	**0**
Non V	Non s	R	**1**
PV	Non s	E	**0**
N	Non s	E	**0**
Ps	V	E	**0.75**
Ps	PV	E	**0.5**
Ps	N	E	**0.25**
Ps	Ps	E	**0.25**
s	V	E	**1**
s	PV	E	**0.75**
s	N	E	**0.5**
s	Ps	E	**0.25**
PV	V	N	**0.5**
PV	PV	N	**0.25**
PV	N	N	**0.25**
PV	Ps	N	**0.25**
N	V	N	**1**
N	PV	N	**0.75**
N	N	N	**0.5**
N	Ps	N	**0.25**
Ps	V	N	**1**
Ps	PV	N	**0.75**
Ps	N	N	**0.5**
Ps	s	N	**0.25**
s	V	N	**1**
s	PV	N	**1**
s	N	N	**1**
s	Ps	N	**0.5**

Fig.5 illustrates the fuzzy set of the buffer inputs which Triangular memberships.

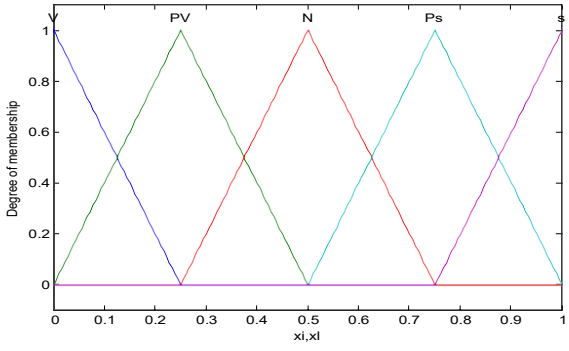

Fig.5 Membership functions of buffer inputs.

ISSN: 2356-5608

Fig.6 illustrates the fuzzy set of the surplus input which Triangular and trapezoidal memberships.

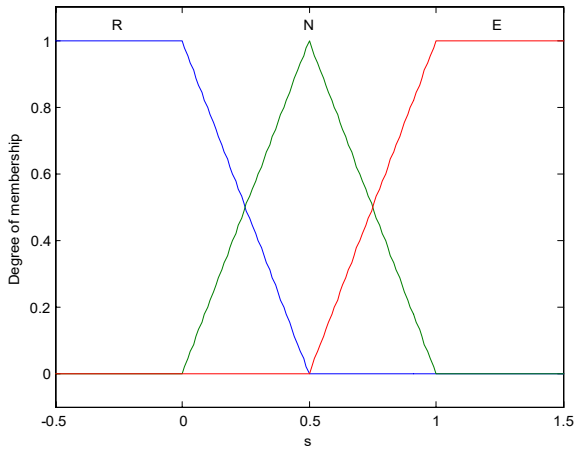

Fig.6 Membership functions of surplus input.

Fig.7 shows the surface of the base rules using in FLC for surplus =0.5.

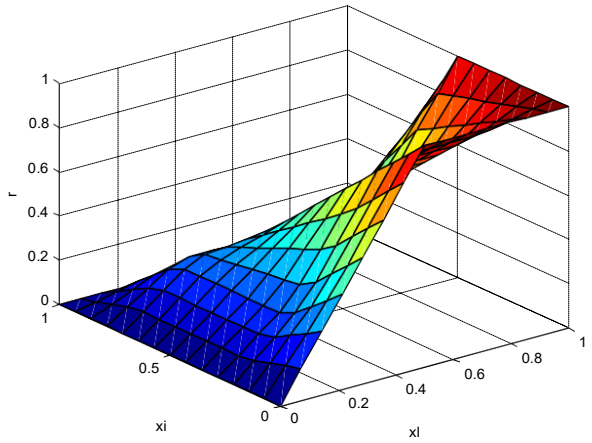

Fig.7 Rule surface of FLC.

V. RESULTS AND SIMULATION USING FUZZY

In the simulation, we simulate the last production and it medium production rate and it surplus.

The results demonstrates that the control policy keeping the demand *d=1*, but there is perturbation in some time (see Figure.8) that means the control policy is not robust.

Fig.8 shows the cumulate production of the last production module.

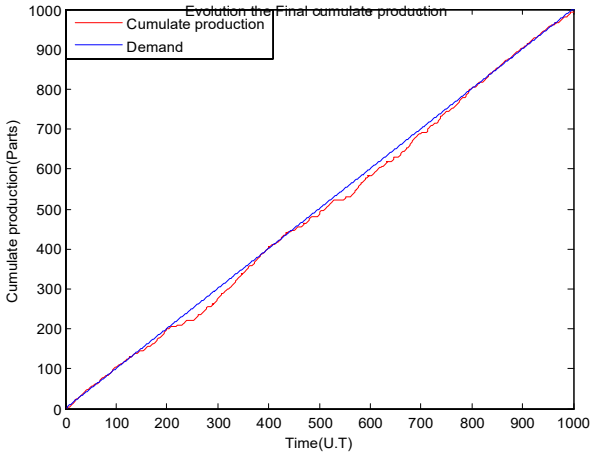

Fig.8 The cumulate production.

Fig.9 shows the medium production rate of the last production module.

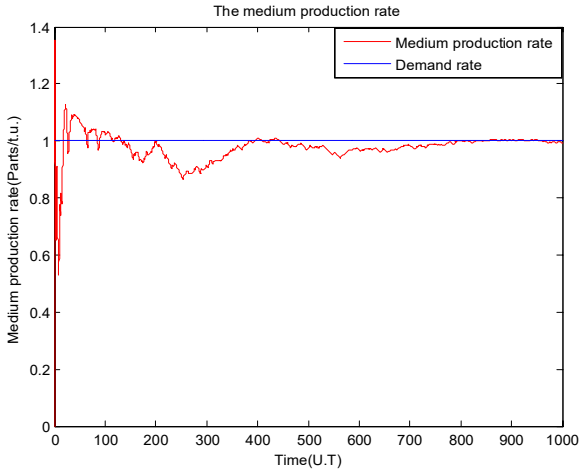

Fig.9 The medium production rate.

Fig.10 shows the surplus of the last production module.

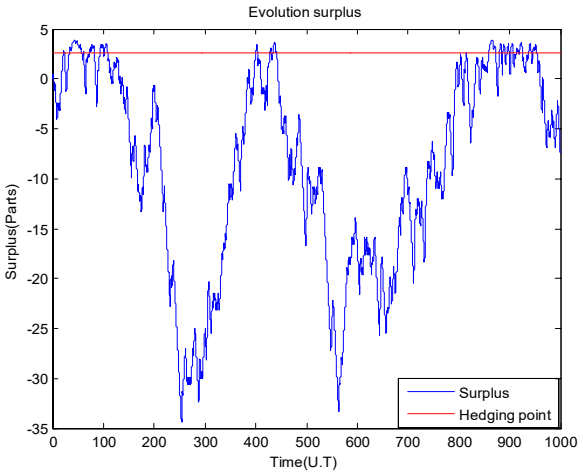

Fig.10 The surplus.

ISSN: 2356-5608

VI. PARTICULE SWARM OPTIMIZATION

The Particle swarm optimization (*PSO*) is first introduced by Kennedy and Eberhart in 1995 [1]. It can be obtained high quality solutions within shorter calculation time and stable convergence characteristics with PSO algorithm than other stochastic methods such as genetic algorithm.

Particle swarm optimization uses particles which represent potential solutions of the problem. Each particles fly in search space at a certain velocity which can be adjusted in light of proceeding flight experiences. The projected position of ith particle of the swarm x_i, and the velocity of this particle v_i at $(t+1)^{th}$ iteration are defined as the following two equations in this study [1]:

$$\begin{cases} v_{iD}^{t+1} = k \cdot v_{iD}^{t} + c_1 r_1 (P_{iD}^{t} - x_{iD}^{t}) + c_2 r_2 \left(g_i^{t} - x_{iD}^{t}\right) \\ x_{iD}^{t+1} = p_{iD}^{t} + x_{iD}^{t} \end{cases} \quad \ldots (10)$$

where, i = 1, ..., n and n is the size of the swarm, D is dimension of the problem space, k is the momentum or inertia, c_1 and c_2 are positive constants, r_1 and r_2 are random numbers which are uniformly distributed in [0, 1], t determines the iteration number, p_i represents the best previous position (the position giving the best fitness value) of the ith particle, and g represents the best particle among all the particles in the swarm. The algorithm of PSO can be depicted as follows [1]:

1. Initialize a population of particles with random positions and velocities on D-dimensions in the problem space,
2. Evaluate desired optimization fitness function in D variables for each particle,
3. Compare particle's fitness evaluation with its best previous position. If current value isbetter, then set best previous position equal to the current value, and p_i equals to the current location xi in D-dimensional space,
4. Identify the particle in the neighborhood with the best fitness so far, and assign its index to the variable g,
5. Change velocity and position of the particle according to Equation (10).
6. Loop to step 2 until a criterion is met or end of iterations.

At the end of the iterations, the best position of the swarm will be the solution of the problem. It is not possible to get an optimum result of the problem always, but the obtained solution will be an optimal one. It can not be able to an optimum result of the problem, but certainly it will be an optimal one.

VII. PSO LEARNING ALGORITHM

To accelerate the convergence of PSO, it was proposed to find a better solution in a minimum computation time and accuracy, we calculate the best solution, on minimizing a certain criterion (objective function) is the mean square error (MSE) calculated by the following equation:

$$MSE = \frac{1}{nT} \sum_{i=1}^{n} e(K)^2 \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (11)$$

Where: $e(k)$ is the total number of samples and T the sampling time, $e(k) =$ Surplus is the difference between the value of the cumulate production and the value of the demand.

Fig.11 illustrates the particle will consist of nine parameters which are the modal values of membership functions of each input, respecting the following constraint: $0 < a_i < a_{i+1} < a_{i+3} < 1$

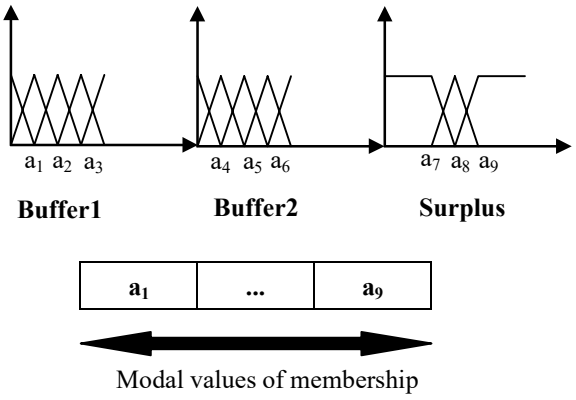

Fig.11 Particle structure of PSO.

The block diagram (see Fig.12) shows the strategy optimization of the controller:

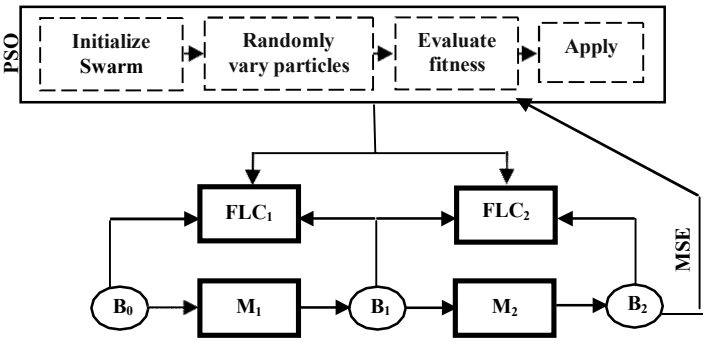

Fig.12 Distributed fuzzy PSO concept.

ISSN: 2356-5608

The pseudo code that describes the procedure is the following as [5], where used the genetic algorithms "**GAs**":

Initialization (creation of the initial swarm)
For i=1 to number of swarm„„iteration""
For j=1 to number of particles
Create the member ship function for the particle j
Run the simulation of the production
Evaluate finesse
End j
Rank the particles on their fitness function
Update new swarm by comparison the position of the particles
End i

Fig.13 illustrates the fuzzy set of the buffer1 input which Triangular memberships optimized.

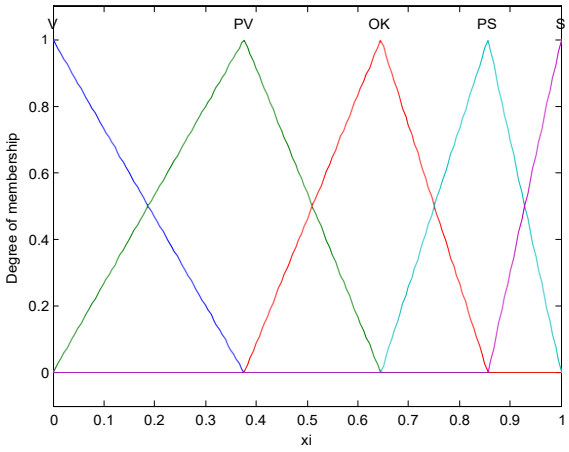

Fig.13 Membership functions of buffer1 input.

Fig.14 illustrates the fuzzy set of the buffer1 input which Triangular memberships optimized.

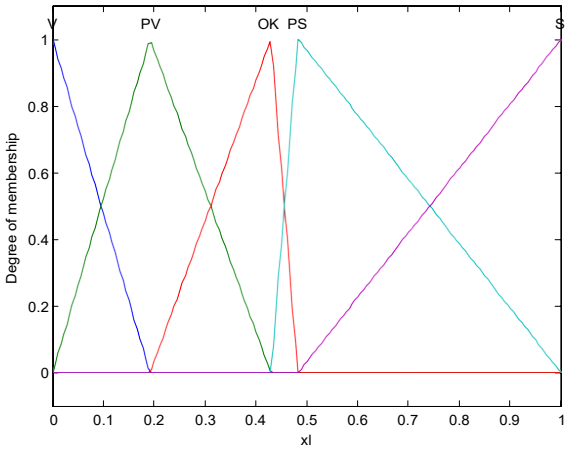

Fig.14 Membership functions of buffer2 input.

Fig.15 illustrates the fuzzy set of the surplus input which Triangular and trapezoidal memberships optimized.

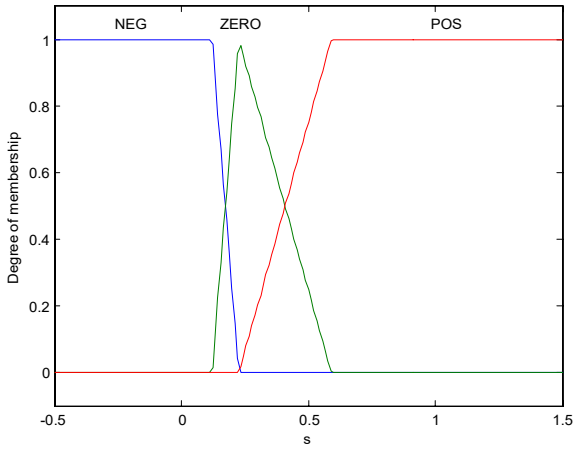

Fig.15 Membership functions of surplus input.

VIII. RESULTS AND SIMULATION USING FUZZY PSO

In the simulation, we simulate the last production and it medium production rate and it surplus. TABLE IV shown the following "PSO" parameters:

TABLE IV
PSO PARAMETRES

Size of the swarm	C_1	C_2	k
20	0.0001	0.0001	0.5

Fig.16 shows the cumulate production of the last production module using fuzzy optimized.

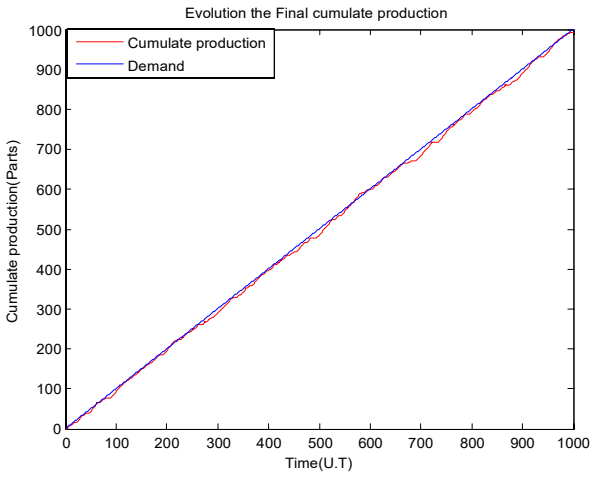

Fig.16 The cumulate production after optimization.

ISSN: 2356-5608

Fig.17 shows the medium production rate of the last production module using fuzzy optimized.

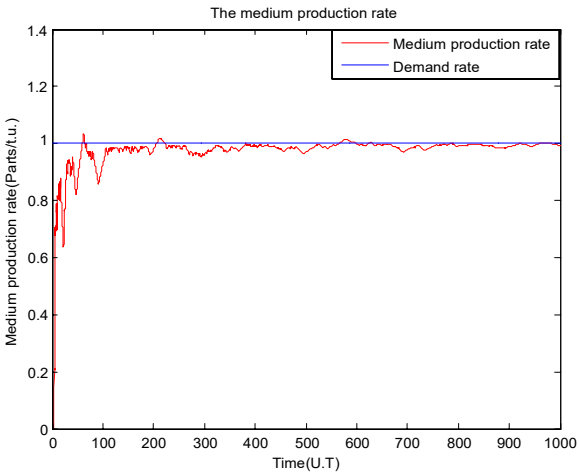

Fig.17 The medium production rate after optimization.

Fig.18 shows the surplus of the last production module using fuzzy optimized.

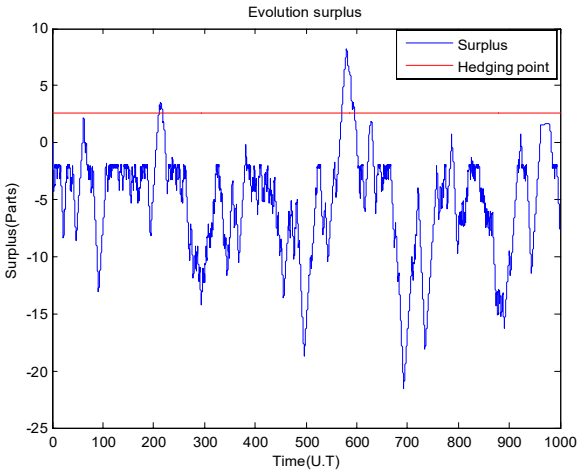

Fig.18 The surplus after optimization.

Conclusions Simulation results, have shown an important improvement of performance and production related costs, with the use of PSO.

Tsourveloudis, N. & al defined the work-in-process inventory is measured by the number of unfinished parts in the buffers throughout the manufacturing system [5]. The important question in *WIP* management is: what is the minimum necessary *WIP*? The answer, which is not straightforward, is that *WIP* is highly associated with the fluctuations of demand. *WIP* is accumulated when the actual production rate is higher than the demand. However, when *WIP* is very low, unpredicted phenomena, such as machine failures, may lead the actual production behind the demand and thus delay deliveries and cause unsatisfied customers. Obviously, product demands of a constant level and pattern make the scheduling task easier than randomly changing demands, in our case the demand not changing, the *WIP* is given by:

$$WIP = \sum_{I=1}^{Numberof\ buffer} x_I(t) \quad \ldots\ldots\ldots\ldots\ldots (12)$$

Control policies that tend to keep *WIP* at low levels have drawn a great deal of attention from researchers and practitioners [5].

Fig.19 shows the Evolution of medium *WIP* in the production line with demand rate "*d=1*".

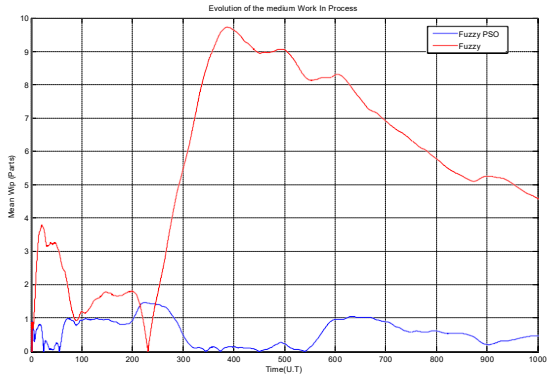

Fig.19 The medium *WIP*.

The fuzzy PSO achieve a substantial reduction of *WIP*.

IX. CONCLUSION

The fuzzy used to control the rate in each production stage so that satisfies the demand"*d=1*"for final products while reducing *WIP* within the system "presented by line of transformation". The PSO identifies the parameters for which the fuzzy controller performs optimal with respect to *WIP* and backlog minimization. The proposed strategy "fuzzy PSO" is compared to known heuristically tuned fuzzy control approaches. Simulation results show that the fuzzy PSO improves system"s performance.

Generally we can apply fuzzy PSO in more complex production systems such as multiple-part-type and/or re-entrant systems "flot-shop" and "job-shop" to get the best results.

ISSN: 2356-5608

REFERENCES

[1] J. Kennedy and R. Eberhart, 1995. "Particle swarm optimization", *in Proc.IEEE Int. Conf. Neural Networks (ICNN'95), vol. IV, Perth, Australia, p p. 1942–1948*.

[2] S.B. Gershwin, "Design and Operation of Manufacturing Systems: The control point policy," *IIE Transactions, vol. 2, pp. 891-906, October 2000*.

[3] Y.Hung Ma, Y.Koren, "Operation of Manufacturing Systems with Work-in-process Inventory andProduction Control," *NSF Engineering Research Center for Reconfigurable Manufacturing Systems, University of Michigan, Ann Arbor MI 48109-2125, USA*.

[4] Karim T, "Développement d"une méthodologie de pilotage intelligent par régulation de flux adaptée aux systèmes de production," *Thèse de doctorat, L'université de Savoie, France, 2008*.

[5] Tsourveloudis, N. C, Doitsidis, L, & Ioannidis, S. (2007), "Work-in-process schedulingby evolutionary tuned fuzzy controllers," *International Journal of AdvancedManufacturing Technology, 34(7–8), 748–761 Springer*.

[6] K. Tamani, R. Boukezzoula and G. Habchi, "Fuzzy Supervisory Based Capacity Allocation Control for Manufacturing Systems", *1-4244-1210-2/07/$25.00 C 2007 IEEE*.

[7] N. C. Tsourveloudis, L. Doitsidis, R. Boukezzoula and S. Ioannidis, " Optimized fuzzy scheduling of manufacturing"s systems", *ICINCO 2005 -intelligent control systems and optimization -*.

[8] S.Mahdi Homayouni , S.Hong Tang, N.Ismail, "Development of genetic fuzzy logic controllers for complex production systems," *S.M. Homayouni et al. / Computers & Industrial Engineering 57 (2009) 1247–1257 Elsevier*.

ISSN: 2356-5608