Journal of Engineering and Applied Sciences 11 (2): 240-251, 2016
ISSN: 1816-949X

Optimized Control of Servo Motor Speed Applying Type-2 Fuzzy

[1]Mohahammad Javad Tajadini and [2]Majid Mohammadi
[1]Department of Electrical and Control,
Kerman Science and Research Branch, Islamic Azad University, Sirjan, Iran
[2]Department of Computer Engineering, Shahid Bahonar University, Kerman, Iran

Abstract: Because of the widespread use of motors, controlling their speed and position control are important. Different types of controllers for electric motors can confirm this statement. Many methods such as PID controllers and intelligent controllers have been proposed for motor control. This study aims to provide a new method for intelligent motor control using fuzzy controller type-2. After measuring motor parameters, a fuzzy controller with two inputs controls error and changes to minimize the errors in the shortest time possible. However, the extraction of rules and membership functions which is often based on trial and error, time-consuming and needs a specialist is a common problem. Evolutionary algorithms are a kind of search algorithm based on natural selection mechanism and a valid method for efficient and effective solution search process. In this method, Particle Swarm Algorithm (PSO) is used to determine type-2 fuzzy membership functions. The proposed method is simulated using data from a motor with MATLAB and in SIMULINK environment.

Key words: Fuzzy control, evolutionary algorithm (PSO), servo motors, fuzzy control type-2, PID controller

INTRODUCTION

DC motors are used for many applications. Due to theirconvenient control and fast performance, these motors can be set in a wide range of speeds. Speed controllers that are designed to perform a variety of tasks for DC motor speed control include two types: continuous (analog) and discrete (digital). Controllers can be of P (Proportional), PI (Proportional-Integral), PID (Proportional-Integral-Derivative), neural networks, fuzzy logic or a combination of these. According to the mentioned cases, using amethod which is fast, accurate and energy-efficient operation is very favorable. Overshooting and difficult settings of PID controllers are their problems. Therefore, achieving fast dynamics and high precision speed control has been one of the controlling goals of this project to gain which the fuzzy controller type-2 is used. The second controlling goal wasusing controllers in various motors and speeds and with different parameters and without redesigning of the controller which is estimated using the PSO algorithm.

In this study, servo motor has been implemented and reviewed at different speeds using fuzzy control type-2. Fuzzy control methods which have opened their place in the control of electrical machines in the recent decades have been proposed by Professor Lotfizadeh. There are three main objectives in designing and construction of the controllers:

- Enhancing reliability
- Reducing prices
- Integration

In addition to these goals, improving behavior (performance) of control systems and saving energy are also two main objectives in designing the driving. That is why in designing controllers in this project, simplicity of logic and ease of implementation have also been noted and this will result in the reduction of the price of the entire project.

In the following, evaluation of servo motors and comparing them with each other is done. Then, the mentioned controlling method is designed and reviewed and finally, the robustness of the method is shown.

DC MOTOR

In complex systems, a control loop is not usually sufficient for controlling requirements or even may be unable to guarantee the stability. Thus, several nested

Corresponding Author: Mohahammad Javad Tajadini, Department of Electrical and Control, Kerman Science and Research Branch, Islamic Azad University, Sirjan, Iran

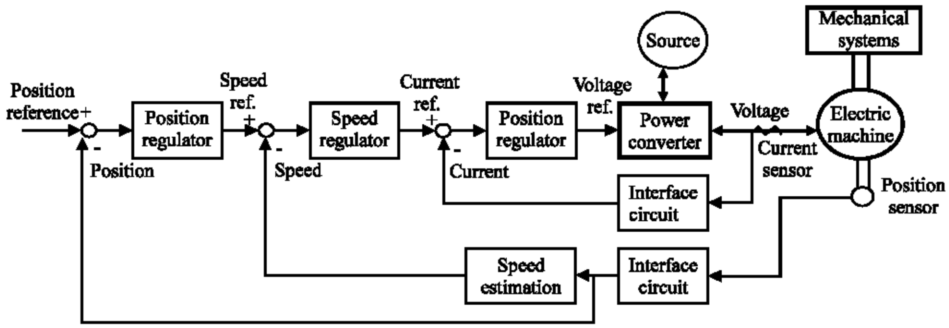

Fig. 1: The controlling block diagram of electric motors

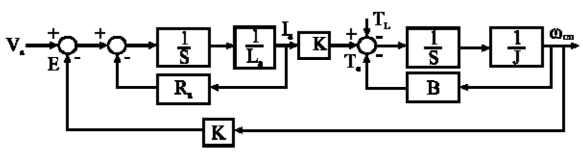

Fig. 2: DC motor model

control loops are used. For example, to control electric motors three loops are commonly used. The overall controlling block diagram of electric motors is demonstrated in Fig. 1.

The most sensitive and most important ring is the inner ring because if the speed of response of this ringis not appropriate, outer rings do not work in practice. In practice, the speed of the inner ring is considered 5-10 times faster than outer ring. The speed of response is the loop bandwidth. Band width ωbw is the sinusoidal signal frequency which causes response of the system to be $<1/\sqrt{2}$ of input value if it enters the system:

$$A\sin(\omega_{bw}t) \rightarrow \boxed{G(s)} \rightarrow \frac{1}{\sqrt{2}} A\sin(\omega_{bw}t)$$

As mentioned in the previous sections, DC motor model is like (Fig. 2). Armature resistance R_a is the damping component of the electric field and the coefficient of friction B is the damping component of mechanical part. These two components reduce the efficiency of the system but they can increase the stability and the speed of response. If these two passive damping components are deleted, system would be oscillating. To improve the control function, active damping components can be added as a feedback control system as shown in Fig. 3.

As is clear, by adding reactive resistance, system special values change. The larger reactive value from R_a, the control system will be more resistant to R_a changes. The same reasoning can be applied to the coefficient of friction (Zadeh, 1965; Zadeh, 1975).

As is clear from Fig. 2, adding damping resistance is like a state feedback which can increase the control system dynamics. Armature current to voltage conversion function is as follows:

$$\frac{i_a}{V_a} = \frac{1}{sL_a + (R_a + R_{active})}$$

If reactive is considered much larger than sL_a in the band width ωbw of the control loop, the above equation changes to the following:

$$\frac{i_a}{V_a} \approx \frac{1}{(R_a + R_{active})}$$

If the relationship between torque and current and also armature voltage and reference torque are considered as follows:

$$T_e = Ki_a$$
$$V_a = (T_e^*/K)(R_a + R_{active})$$

Then:

$$\frac{T_e}{T_e^*} = \frac{Ki_a}{T_e^*} \approx \frac{(R_a + R_{active})}{(R_a + R_{active})} = 1$$

This equation shows that if the amount of reactive is considered large enough, torque can be controlled instantaneously. It should be noted that the amount of Reactive cannot increase too much because it causes excessive delays in the system and make the system unstable. In practice, the amount of damping component is implemented in terms of integral controller.

To design the current controller, supposing that the inertia of the motor and load is large enough, the amount of back EMF of motor can be considered stable and the motor in the form of a simple R-L Model as follows.

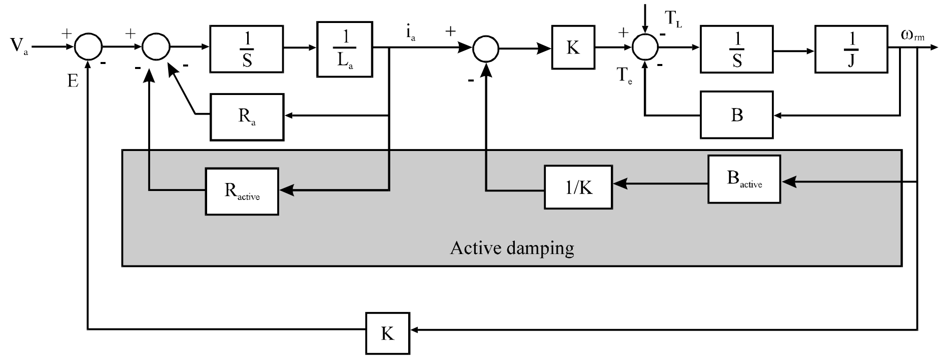

Fig. 3: Control feedback model of DC motor

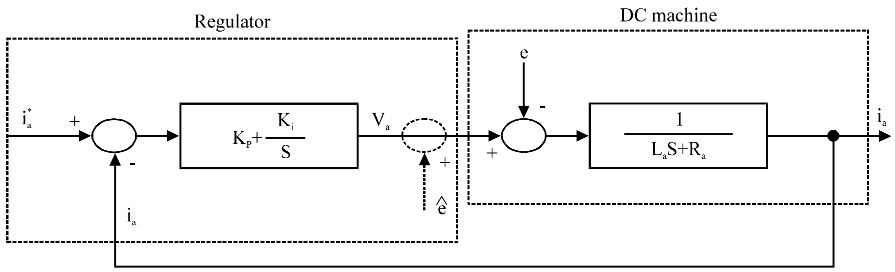

Fig. 4: DC motor current control model

If the value of the motor EMF voltage can be estimated based on the speed, then for removing confusion, the feed forward compensation can be used. This is shown in Fig. 4 by e^.

PI controller parameters of the current can be set to follow the closed-loop transfer function of a system as a low-pass filter. To do this, the values of controlling parameters should be set as follows:

$$K_p = L_a\omega_c$$
$$K_i = R_a\omega_c$$

where, ω_c is the same bandwidth of ωbw controller. By selecting these values, the closed-loop transfer function for the above shape becomes as follows:

$$\frac{i_a(S)}{i_a^*(S)} = \frac{\omega_c}{S+\omega_c} = \frac{1}{T_cS+1} \qquad T_c = \frac{1}{\omega_c}$$

After selecting the system bandwidth, the values of controller parameters are calculated according to the machine parameters. As is clear because the transfer function is no longer just a low-pass filter, there is no overshoot or continual error (Lughofer, 2011).

Motor modeling: For changing the motor speed, motorers face many problems for controlling the power output of

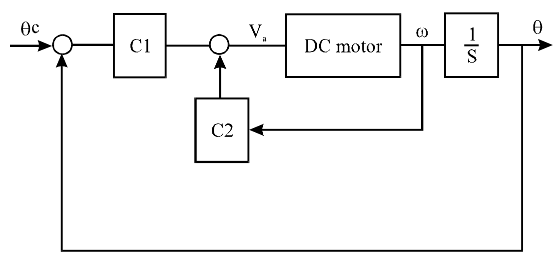

Fig. 5: DC motor controlling structure

the motor. These changes can have adverse effects such as changes in the proposed method have been simulated and implemented by using a DC machine data by MATLAB and different states have been investigated (Fig. 5).

Calculating mathematical model of the motor: As mentioned in previous studies, DC motor equation is as follows Fig. 6 and 7:

$$V_a = R_a i_a + L_a\frac{di_\alpha}{dt} + e$$
$$e = K_b\omega_m$$
$$T_e = K_m i_\alpha$$
$$T_e = J\frac{d\omega_m}{dt} + B\omega_m + T_L$$

Note that since we have considered the armature voltage system as the input and rotation speed as the output, the system transfer function is calculated as follows (Fig. 8):

$$\frac{\omega_{rm}}{V_a} = \frac{K_m}{(L_a s + R_a)(Js + B) + K_m K_b}$$

Designing PI controller: Global Proportional-Integral (PI) is a special case of classic controllers family which is known as the Proportional-Integral-Derivative (PID) controllers. This type of controllers has been the most common method for controlling industrial processes by feedback combination (structure). About >95% of all used controllers are PID. For electric motors applications, PI is the most commonly selected control structure. This controller is generally fed with an error (reference) signal minus the real answer. As shown in Fig. 9 control is performed in closed-loop control structure (Magdalena and Monasterio-Huelin, 1997).

A PI controller is essentially a collection of proportional and integral operators. The performance of a proportional control is simply a closed-loop gain which improves the response of a system by amplifying the error signal. Integral operator pluralizes the error and therefore increases the response speed while the optimal output changes (e.g., in the form of a unique step function) and also the system error fits these changes. Integral function is applied when achieving an optimal response is important because it eliminates the steady-state error. Conversion function for the PI controller is as follows:

$$K_p = (1 + 1/T_i s)$$

Where:

K_p = The proportional gain
T_i = Integral gain
s = The Laplace operator frequency

Achieving proportional and integral gain in different systems is done through different approaches. In this

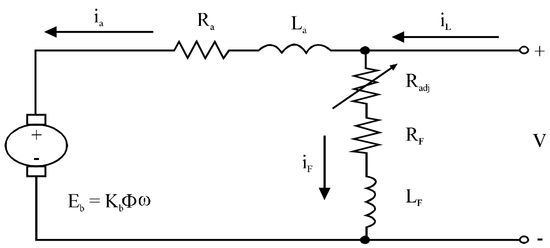

Fig. 6: Series DC motor equivalent circuit

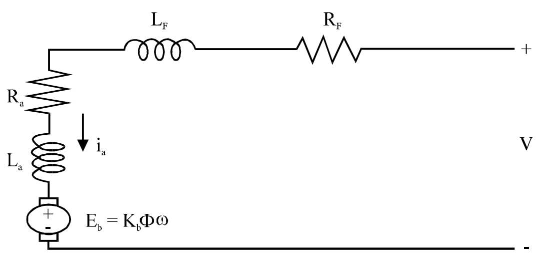

Fig. 7: DC motor shunt circuit

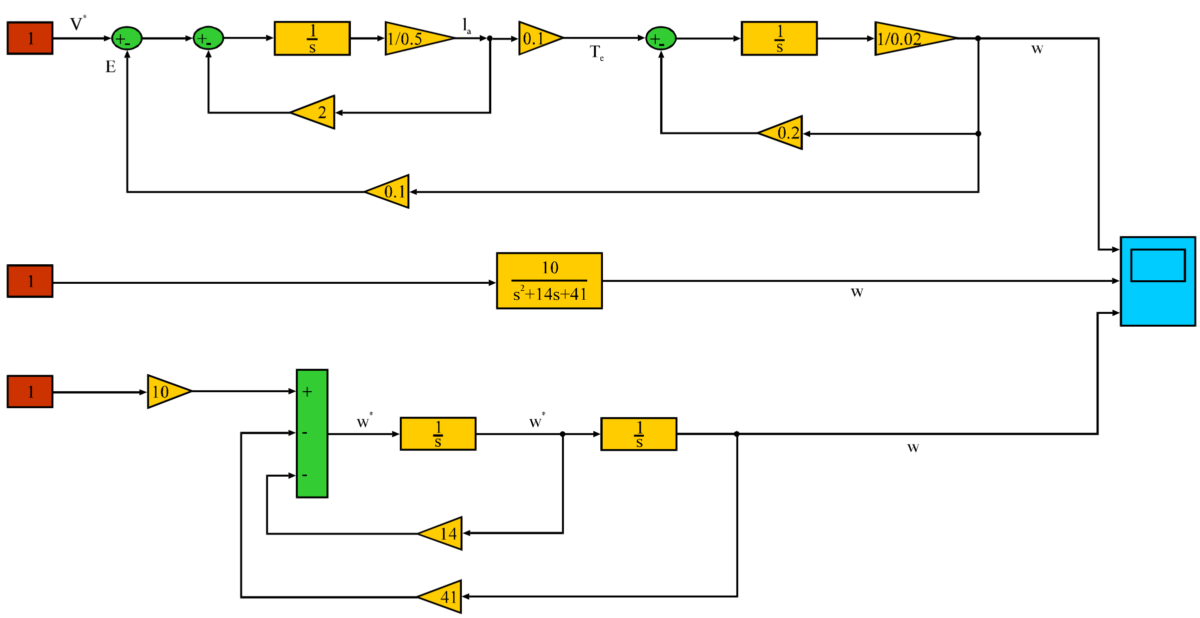

Fig. 8: Implementation of DC motor in MATLAB

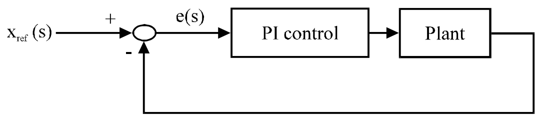

Fig. 9: Closed-loop feedback control for PI controller

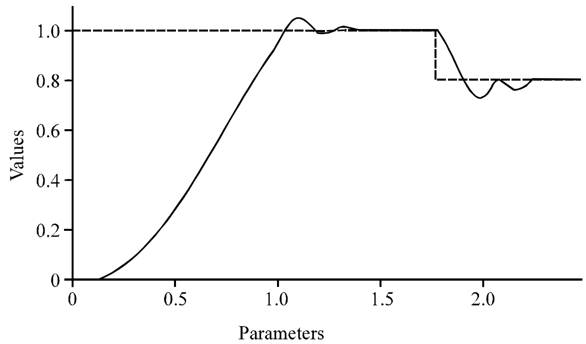

Fig. 10: Controlling motor speed with PI controller

study, we applied a standard Chi-squared error (RMSE) which is defined by the following equation (normalized and in terms of percentage) and used it as a function of cost and genetic algorithms to find the best values for the proportional and integral gain:

$$\text{Normalized RMSE (\%)} = \frac{1}{yr}\sqrt{\frac{1}{T}\int_0^T (y\text{-}yr)^2\,dt}\times 100$$

In the electric motor applications, the PI controlling error is as follows:

$$e(t) = P_{ed}(t)\text{-}P_e(t)$$

Where:

P_{ed} = The desired output or setpoint for motor in 1.5 MW

P_e = The real power delivery of the motor (Ng and Li, 1994)

The best achieved coefficients from the Particle Swarm algorithm for the proportional and integral gain are 5 and 18. The result of designing this controller is shown in Fig. 10. The controller is a PI. We then used the fuzzy controller design. The behavior of the controller was also evaluated.

FUZZY SYSTEMS

Fuzzy logic, as a key element in the context of artificial intelligence, plays a key role in facing uncertainties and inaccurate information. In principle, the main motive behind the fuzzy logic provides a framework for describing human knowledge where imprecise knowledge of the information is a common feature. To

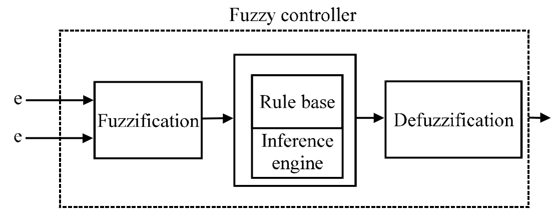

Fig. 11: The structure of a fuzzy logic controller

implement this logic, fuzzy logic must be able to model the variables in categories such as large, small which are often used by humans. If we look carefully, we see that these variables are also found in nature like. Therefore, to describe the variables in mathematics, concept of setson the one hand and fuzzy setsshould be used for this purpose on the other hand.

Type-1 fuzzy: Fuzzy sets were introduced for the first time in 1965 by Professor Lotfi (Atashpaz-Gargari and Lucas, 2007). These sets were the founders of a successful way to model the uncertainty and ambiguity. Since then, using fuzzy sets for computer systems, especially in control applications developed (Bonarini, 1996). Fuzzy systems, due to membership functions having exact belonged degrees have limited ability to reduce the uncertainty in fuzzy rules. In the real world, many sources of uncertainty exist in facing fuzzy systems including.

Describing parts of fuzzy rules are uncertain, i.e., the words used in the provision of the requisite and the result of the rules can have different meanings for different people.

The results of the expert group will often be different for the rule because experts are not necessarily in agreement with each other.

Noisy data can be used for adjusting the parameters of a fuzzy system. Measurements that activate a fuzzy system are noisy and therefore uncertain Hoffmann and (Jalalizadeh *et al*., 2011). Thus, the level of uncertainty is often dependent on the data (either this data is made of verbal information or numerical data). The fuzzy logic has been used successfully in various fields such as regression, system modeling and controlling and categorization of patterns. Among them, the control can be considered as one of the most relevant fields (Fig. 11).

Flc has three main components such as fuzzification, fuzzy inference motor and defuzzification. Flc block diagram is shown in and its result can be seen in Fig. 12.

Fuzzy system type-2: In 1975, Professor Zadeh introduced type-2 fuzzy sets as an extension of fuzzy sets (Lian *et al*., 2011). Since then, to distinguish between type-2 fuzzy sets and previous fuzzy sets, previous fuzzy sets are

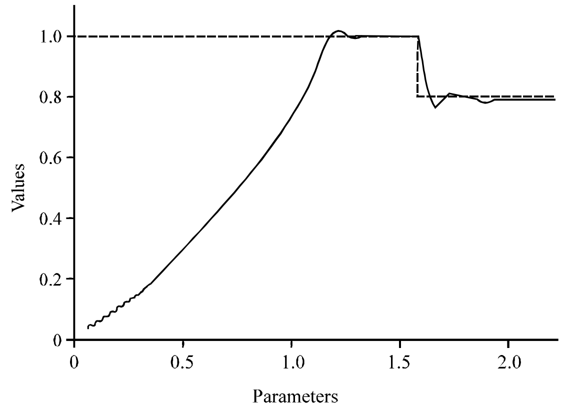

Fig. 12: Motor speed with fuzzy control type-1

typically called type-1 fuzzy sets. Fuzzy type-2 sets have fuzzy membership grades and therefore are called fuzzy-fuzzy sets which have the ability to reduce their effects and modeling to deal with uncertainty (Lughofer, 2011; Zadeh, 1965). In short, the reasons for the emergence of type-2 fuzzy logic can be expressed in five parts:

- Inability of mathematical descriptions of systems that generate data for time-varying mechanisms like mobile communications
- Inability of mathematical description such as signal-to-size (non-stationarity) non-stationary noise
- Inability of mathematical description to describe properties in pattern recognition which have non-stationary applications such as the classification of traffic images basic rule
- The knowledge gained from a number of experts to the questions contained words that are uncertain
- Using language phrases which have immeasurable range (Jalalizadeh *et al*., 2011; Zadeh, 1975)

Structure of a type-2 fuzzy system: The overall structure of a type-2 fuzzy system is shown in Fig. 12. A type-2 fuzzy system includes four parts of fuzzification, rules, inferences and the withdrawal. In fact, a fuzzy system is a mapping between an input and an output defuzzification. In a type-2 fuzzy system, output process consists of two stages. First, a type-2 fuzzy set is mapped to a type-1 fuzzy set. This is called the stage reduction or reduction of the order. Then, the reduced set is defuzzificated. Reduction methods in fuzzy type-2 are the same developed methods of defuzzification in type-1 fuzzy systems. Reduction includes methods of the center of gravity, the center ofthe set and height (Hoffmann and Pfister, 2013) (Fig. 13).

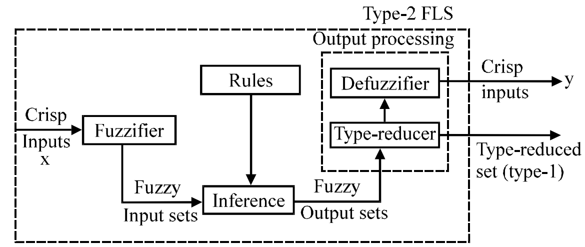

Fig. 13: Type-2 fuzzy system structure

Fuzzifier: Since the input of a fuzzy system is often unfuzzy values, a mechanism is needed to convert them into fuzzy values. This task is done by fuzzifier. So far, several mechanisms such as the singleton fuzzy, Gaussian type-1 and 2, triangular, etc., have been introduced. However, to reduce the volume of calculations, singleton fuzzifier is used.

Rules: Human knowledge of experts will be classified here. This knowledge is usually expressed as if and when fuzzy rules, for instance, the ith rule for a p inputs and one output is as follows:

$$R^i\text{: If } x_1 \text{ is } \tilde{F}_1^i \text{ and } x_2 \text{ is } \tilde{F}_2^i \text{ and ... } x_p \text{ is } \tilde{F}_p^i$$
$$\text{then y is } \tilde{G}^i i = 1, 2, ..., M$$

Where:
$\tilde{F}_n^i$ = The antecedent fuzzy set
$\tilde{G}^i$ = The consequent fuzzy set

Fuzzy inference motor: Inference motor combines the rules and gives a mapping between input and output fuzzy sets. In type-2 fuzzy systems, an interval with respect to the upper and lower bounds of jth firing set input can be defined as follows:

$$F^j(\underline{x}) = [\underline{f}^j(\underline{x}), \bar{f}^j(\underline{x})]$$

$$\underline{f}^j(\underline{x}) = \underline{\mu}_{\tilde{F}_1^j} * \underline{\mu}_{\tilde{F}_2^j} * ... * \underline{\mu}_{\tilde{F}_n^j}$$

$$\bar{f}^j(\underline{x}) = \bar{\mu}_{\tilde{F}_1^j} * \bar{\mu}_{\tilde{F}_2^j} * ... * \bar{\mu}_{\tilde{F}_n^j}$$

In these relationships, $\underline{x} = [x_1, ..., x_n]^T$, $\underline{\mu}_{\tilde{F}^j}$ represents the lower bound and $\bar{\mu}_{\tilde{F}^j}$ represents the upper bound of the input membership functions. As it is seen, calculation of these relationships is very simple and direct.

Order reducer: Order reducer converts a type-2 fuzzy set into a type-1 fuzzy set. The membership amount of this fuzzy set is achieved through secondary membership function of type-2 fuzzy. Since, the secondary membership interval type-2 fuzzy sets havealways a value of 1, therefore reducing the membership function type-1 fuzzy output of order reducer will also be 1. Thus, to get this function, just the smallest amount (left end point) and the largest amount (right end point) are needed to be calculated. The left and right end points are also a measure of uncertainty, the more their difference, the more uncertain they are.

Defuzzifier: Since, the output of fuzzy systems is a fuzzy amount in order to be able to apply it to the controlled system, it should be converted to a certain amount by a defuzzifier. However, since the secondary membership function of type-2 fuzzy set is one, so the center of gravity can be easily defuzzified by averaging the left and right end points achieved through order reducer.

DESIGNINGTYPE-2FUZZY CONTROLLER

Rule-based fuzzy logic controllers are used when the dynamics of the system is not well understood or when the system shows major non-linear behavior. Fuzzy logic controllers apply the same logic to how humans make decisions and therefore controlling rules include expert knowledge of the system. As mentioned earlier, the great advantage of fuzzy control acts on the motor when the motor is needed neither to be described exactly nor to be linear. The process of designing fuzzy logic controller usually involves defining input, applying the rules and designing a method to convert the result of fuzzy rules to known as defuzzification.

In this part of the study, to adjust the motor speed, a type-2 fuzzy controller is used. The designer must select the appropriate input to be sure that controllers have adequate information to take appropriate decisions. To select the motor inputs of type-2 fuzzy control with consideration given at the modeling section, speed error and speed error changeshave been selected as the input and voltage as the output.

Selecting the input and output membership functions is the next step which is shown in Fig. 14-16. Here, the symmetrical triangular membership functions are used for input and output because they are more sensitive, especially when variables approach zero. The horizontal axis changes can be considered and setas fuzzy system parameters. As shown in Fig. 14-16, membership functions can be classified at 7 levels of Negative Big (NB),

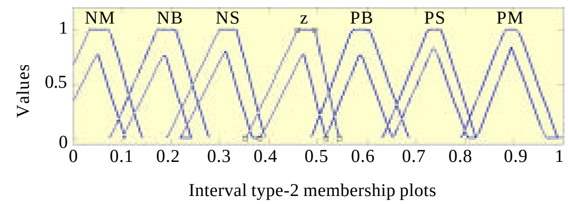

Fig. 14: Speed error input membership function

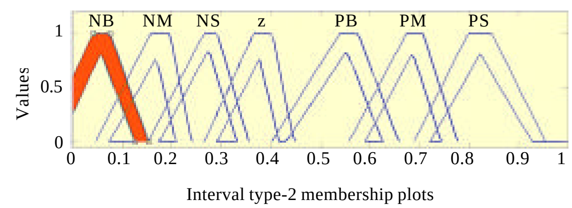

Fig. 15: Membership function of derivative of speed error changes

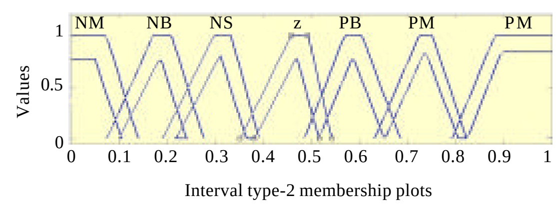

Fig. 16: Membership function of the type-2 fuzzy system output, motor speed

Negative Medium (NM), Negative Small (NS), Zero (ZE), Positive Small (PS), Positive Medium (PM), Positive Big (PB). As has been pointed out, the proposed fuzzy logic controller, based on the speed error and the speed error changes is calculated as follows:

$$e(k) = \omega ref\,(k)\text{-}\omega(k)$$

$$de(k) = e(k)\text{-}e(k\text{-}1)$$

In this relationship, ωref is the optimum speed and ω is the measurement speed of the generator. Simple and immediate understanding of the controllinglogic can be obtained with the following considerations.

If σΔe and Δe are large negative values, output voltage will be very large and its range increases and therefore the speed should decrease.

If Δe is big negative and σΔe is big positive, the output voltage is higher than the reference value but as the range is decreasing, speed changes must be small.

If Δe is small, the speed changes should be gentle because big changes may stimulate vibration modes of the system.

In this controller, the logic of “minimizing” and operator are used for combining conditions and Mamdani fuzzy system search motoris selected. Intended fuzzy rules are considered in Table 1. Due to the existence of 7 input membership functions, there are 49 rules in this controller.

In type-2 fuzzy systems there is a reducer which converts type-2 fuzzy systems to type-1 fuzzy systems. Then, using a defuzzifier the output center of gravity (centroid) is returned to permanent state. Finally, the fuzzy controller designed in MATLAB in the produced motor model is used in simulink.

The results of this type-2 and 1 fuzzy controller and PID are shown in Fig. 13. In compare with type-1 fuzzy and PID, the mutation is reduced. As is obvious, type-2 fuzzy controller reacts faster than type-1 and PID.

In this case, we attempted to load. The behavior of the controller against the distortion of the wind speed is also evaluated and the result can be seen in Fig. 4-15. System reaction quality is worsened and the controller will probably need to be manipulated and re-arranged. The evolutionary algorithms are used to solve this problem.

Table 1: Fuzzy rules related to the designed controller

$\Delta e_i/e_i$	NB	NM	NS	Z	PS	PM	PB
NB	NB	NB	NB	NM	NS	NS	Z
NM	NB	NM	NM	NM	NS	Z	PS
NS	NB	NM	NS	NS	Z	PS	PM
Z	NB	NM	NS	Z	PS	PM	PB
PS	NM	NS	Z	PS	PS	PM	PB
PM	NS	Z	PS	PM	PM	PM	PB
PB	Z	PS	PS	PM	PB	PB	PB

EVOLUTIONARY ALGORITHMS

Eas are basically optimizers of functions-searching algorithms and can be applied to a wide range of issues, in which the function is given but decision-making variables which optimize the function are unknown. These issues include: optimization of motorering, scheduling, bioinformatics, developing hardware and even the arts. EAs, similar to many of the soft computing techniques, are suitable for complex and non-linear issues when other computational methods are not accurate to solve problems. As an optimal search algorithm, EAs are comparable with other soft computing techniques such as gradual annealing, taboo search or mathematical techniques such as gradient descent and simplex method.

EAs are of various species to be able to cover a wide range of issues with special characteristics. These features include: multi-modal issues, the dynamics, optimization problems with constraints and multi-objective optimization.

PSO optimization algorithm: PSO method is a global optimization method that can deal with problems the answer of which in an n-dimensional space is a point or level. In such an atmosphere, some assumptions are made and a preliminary speed is assigned to particles; communication channels between the particles are also considered. Then, the particles are moving in the space and the results are calculated after each period based on a criterion of merit. Over time, the particles are accelerated toward particles that have higher eligibility criteria and are in the same communication group. The main advantage of this method in compare with optimization strategies is that a large number of swarming particles cause a flexible

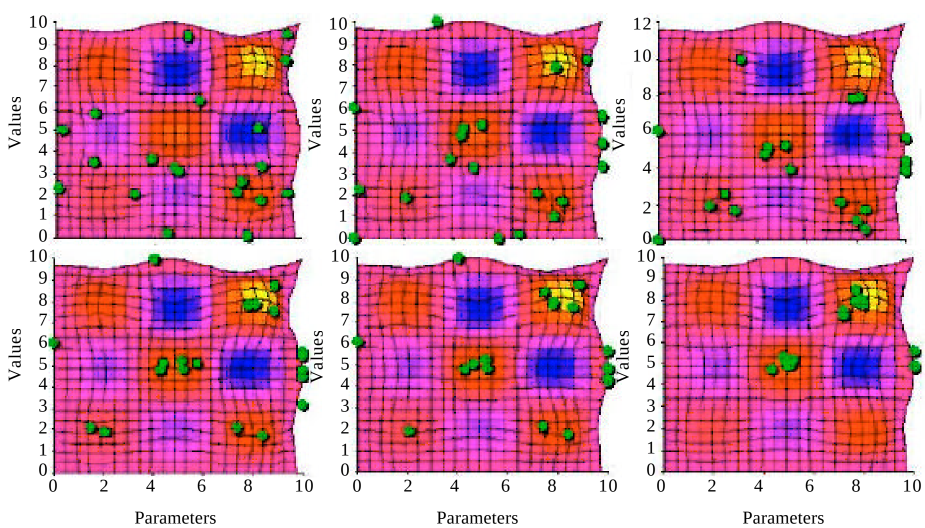

Fig. 17: Trend of particles in a group

picture, the initial position of particlesis shown in a two-dimensional searching space. These particles converge with the particles in the lower right corner with repetition of algorithm.

Each particle has a position that determines the coordinates of the particle in the multi-dimensional search space. Motion of a particle in the particle position will change over time. x_i^k specifies the position of particle i at time k. For movement in the space, every particle needs a speed. v_i^k specifies the speed of particle I at time k. By adding the speed to the position of each particle, a new position can be considered for the particle. The equation for updating the position of the particle is given in equation:

$$X_i^{k+1} = \text{Velocity}_i^{k+1} + X_i^k$$

Whether the position of a particle in the search space is a suitable position or not is evaluated by a fitness function. Particles are able to remember the best position they have ever had during their life. The best personal experience or best position met by a particle is called pbest. Particles can also be aware of the best position visited by the whole group. The position is simply called gbest.

What is called the velocity vector algorithm is the result of three factors, including the speed of the previous step, the best personal experience and the best collective experience which are updated and make the next speed. Next speed is obtained according to the following equation:

$$\text{Velocity}_i^{k+1} = \omega * \text{Velocity}_i^k + c_1 * \text{rand}(.) * (P_{best,i} - X_i^k) + c_2 * \text{rand}(.) * (G_{best} - X_i^k)$$

In the above equations, ω is the weight coefficient, c_1, c_2 are learning coefficients and rand (.) is a random number between 0 and 1. k is the current iteration. The inertia weight was for the first time introduced by (Karr, 1991). In fact, this weight involves a percentage of the previous speed of the particle in the calculation of the new speed. The more this amount, the more the general search increases, vice versa. Figure 18 shown the basis of PSO.

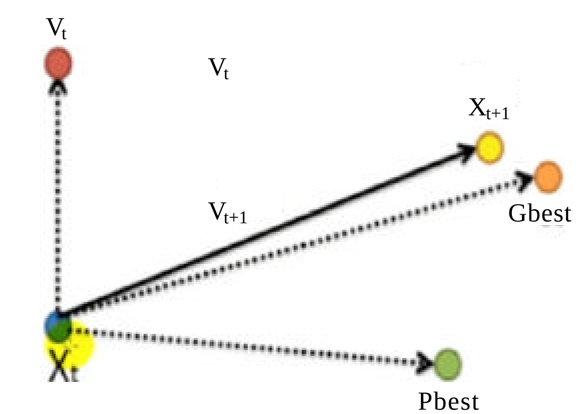

Fig. 18: Basis of PSO algorithm

DESIGNING TYPE-2 FUZZY CONTROLLER PSO

As noted above, the fuzzy controllers, because of the simplicity in the design and implementation of various applications are rapidly expanding. However, due to lack of systematic approach to designing and setting membership functions, trial and error is often used to regulate them. This is overwhelming especially when the number of membership functions is large or system's dynamics is complex and non-linear like in servo motors. Using Particle Swarm Algorithm (PSO), a method isprovided to set membership functions.

Here, without changing the fuzzy rules shown in the Table 1, periods shown in Fig. 19 are optimized using Particle Swarm algorithm and system error criterion RMSE. In a program written for this purpose, the best ratio which can be found in the range of [3, -3] multiplies the correction factors in the range of changes of error input and its derivative and the output of voltage and minimizes the RMSE's servo system. This has been done by using a Particle Swarm algorithm with repetition number of 50, the number of population explained in the previous chapter.

Figure 19 also illustrates the control level achieved by combining fuzzy rules and membership functions output for voltage variations. To explain how to set up the rules as shown in Fig. 20 and for a better resolution, input dimensions have been reduced to two dimensions of the software.

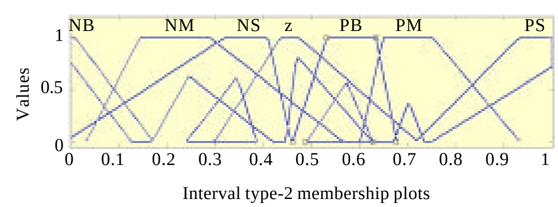

Fig. 19: Type-2 fuzzy membership function, optimized with PSO for voltage variations

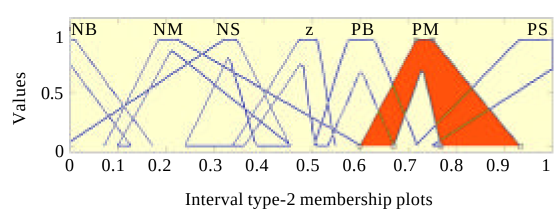

Fig. 20: Type-2 fuzzy membership function, optimized with PSO

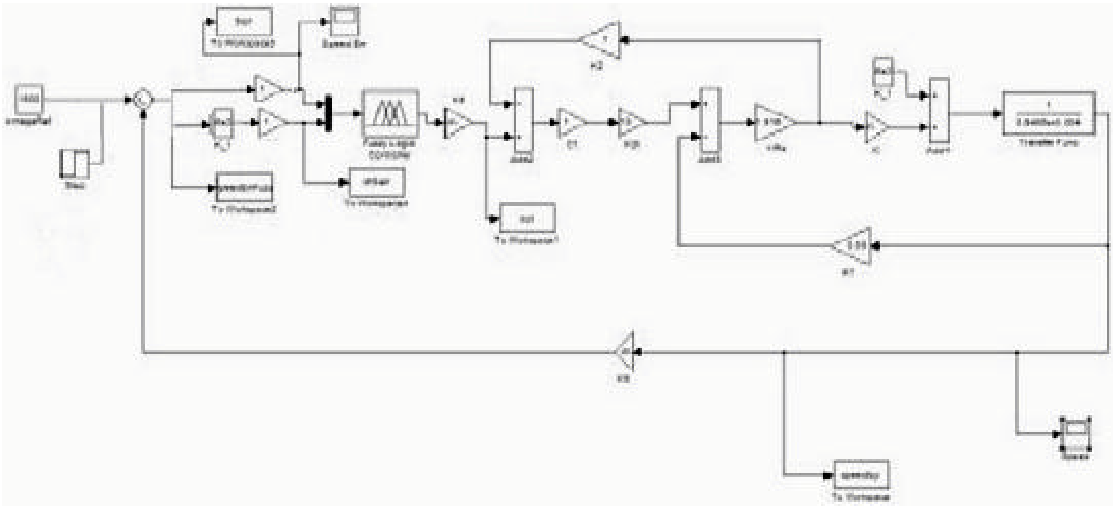

Fig. 21: Block of PSO fuzzy controller diagram in MATLB-SIMULINK

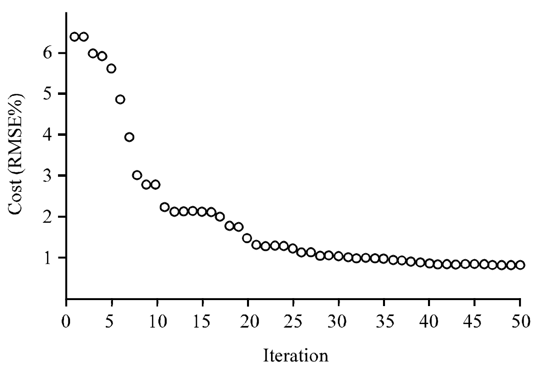

Fig. 22: Iterations

In the study, it was seen that with the error and its derivative at any moment and of course the motor speed deviation from nominal value, a type-2 fuzzy controller can be designed which itself may have advantages and disadvantages. The ability of Particle Swarm algorithm in discovering non-linear relationships between input data and nonlinear dynamics model of the data motivated scientists to make use of the ability of this algorithm. Each of type-2 fuzzy system and Particle Swarm algorithm has both advantages and disadvantages. Fuzzy systems are able to use human language and also facilitate the expression; they can use human experience and qualified personnel and experts while they are not able to learn. However, Particle Swarm algorithm using data collection has the ability of self-educating. Particle Swarm algorithm is not yet clear and will not be able to use human language. In this study, we aim to use type-2 fuzzy and Particle Swarm algorithm having the advantages of both methods as described in the previous chapter. Designing the appropriate controller for improving the quality of motor run is done and the results of this method are compared with the use of the PID controller and type-1 fuzzy.

As in type-2 fuzzy, its error and derivative are used to estimate the output control, fuzzy controllers can be considered as an advanced PD controller.

As it is clear in Fig. 21, the two inputs of the error and its derivative, using ki and kp which are adjustable parameters in the process of Particle Swarm algorithm, enter the first layer which is called the fuzzy membership function. Here, the triangular membership functions are used for type-2 fuzzy. The 7 separate membership functions are used for each input; there would be a total of 42 adjustable parameters.

According to the discussions in the previous chapter, a Particle Swarm algorithm is used which uses a 42-dimensional sample space exploration next 42 samples to determine the optimal parameters to minimize the RMS deviation from the ideal system. How the optimum Particle Swarm algorithm is achieved after 50 iterations can be seen in Fig. 22-24.

To clarify the issue on the one hand and a correct understanding of the designed type-2 fuzzy controller on the other hand,the obtained fuzzy membership functions obtained from Particle Swarm algorithm, present in type-2 fuzzy controller-PSO is displayed. Optimal triangular membership functions for error are shown in Fig. 6-8 shows the optimal triangular membership functions for error derivative. As it can be seen, due to being symmetric and preventing interference, membership functions are symmetric.

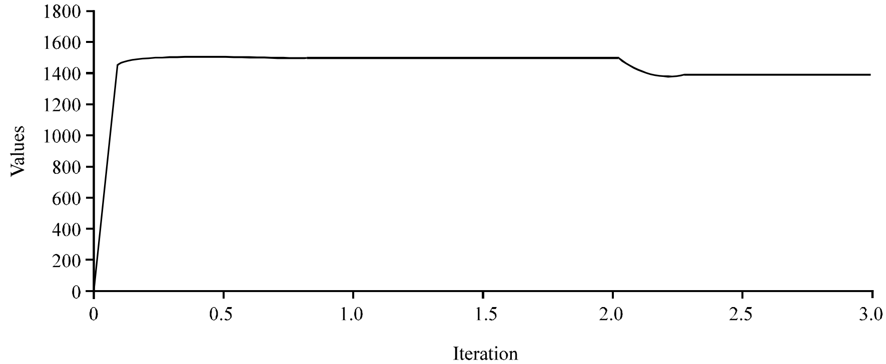

Fig. 23: The result of using type-2 fuzzy control and PSO

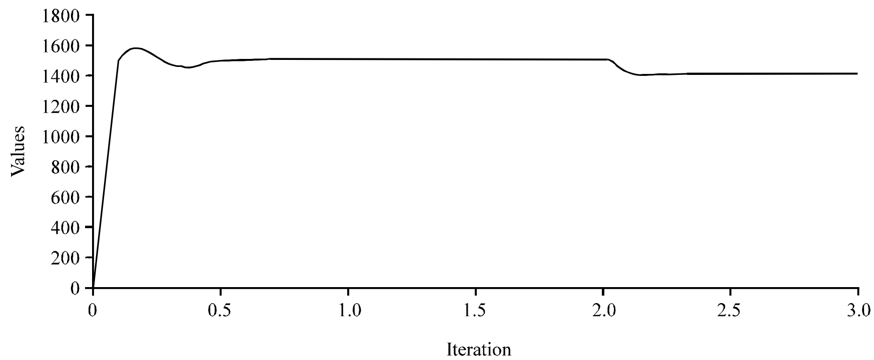

Fig. 24: The result of using typr-2 without PSO

COMPARING CONTROLLERS

As it is seen, in the previous study a type-2 fuzzy controller-PSO was designed based on input error and error derivative. This section compares this controller with classic PID controller and type-1 fuzzy controller PSO.

Motor parameters:

L_a = 18 mH, V = 125 V, R_F = 0/2 ohms, L_F = 44 mH

K_t = 3, K_b = 0/55, R_a= 0 /24 ohms, J = 0/5 kgm^2

REFERENCES

Atashpaz-Gargari, E. and C. Lucas, 2007. Imperialist competitive algorithm: An algorithm for optimization inspired by imperialistic competition. Proceedings of the IEEE Congress on Evolutionary Computation, September 25-28, 2007, Singapore, pp: 4661-4667.

Bonarini, A., 1996. Evolutionary Learning of Fuzzy Rules: Competition and Cooperation. In: Fuzzy Modelling. Witold, P. (Ed.). Springer US, USA., pp: 265-283.

Hoffmann, F. and G. Pfister, 2013. Evolutionary design of a fuzzy knowledge base for a mobile robot. Intl. J. Approx. Reason., 17: 447-469.

Jalalizadeh, S., S. Behzadpoor and M. Hashemi, 2011. PID design for AVR system by PSO and imperialist campetitive algorithms. Proceedings of the First International Conference on Computer Science, Engineering and Information Technology (CCSEIT-2011), September, 23-25, 2011, Tirunelveli, Tamil Nadu, India,-pp: 24.

Karr, C., 1991. Genetic algorithms for fuzzy controllers. J. AI Expert, 6: 26-33.

Lian, K., C. Zhang, L. Gao and X. Li, 2011. Integrated process planning and scheduling using an imperialist competitive algorithm. Intl. J. Prod. Res., 50: 4326-4343.

Lughofer, E., 2011. Evolving Fuzzy Systems-Methodologies, Advanced Concepts and Applications. Springer, Berlin, Heidelberg, ISBN: 978-3-642-18086-6.

Magdalena, L. and F. Monasterio-Huelin, 1997. A fuzzy logic controller with learning through the evolution of its knowledge base. Intl. J. Approx. Reason., 16: 335-358.

Ng, K.C. and Y. Li, 1994. Design of sophisticated fuzzy logic controllers using genetic algorithms in fuzzy systems 1994. Proceedings of the Third IEEE Conference on IEEE World Congress on Computational Intelligence, June 26-29, 1994, IEEE, Orlando, FL., pp: 1708-1712.

Zadeh, L.A., 1965. Fuzzy sets. Inform. Control, 8: 338-353.

Zadeh, L.A., 1975. The concept of a linguistic variable and its application to approximate reasoning-I. Inform. Sci., 8: 199-249.