Genetic fuzzy self-tuning PID controllers for antilock braking systems

Abdel Badie Sharkawy
Mechanical Eng. Dept., Faculty of Eng., Assiut University, Assiut, Egypt
e-mail: ab.shark@aan.edu.eg

Since the emergence of PID controllers, control system engineers are in pursuit of more and more sophisticated versions of these controllers to achieve better performance, particularly in situations where providing a control action to even a minimal degree of satisfaction is a problem. This work is an attempt to contribute in this field. Variations in the values of weight, the friction coefficient of the road, road inclination and other nonlinear dynamics may highly affect the performance of Antilock Braking Systems (ABS). A self-tuning scheme seems necessary to overcome these effects. The addition of automatic tuning tool can track changes in system operation and compensate for drift due to aging and parameter uncertainties. The paper develops a self-tuning PID control scheme with application to ABS via combinations of fuzzy and Genetic Algorithms (GAs). Results are reported and discussed.

1. Introduction

Difficulties in designing an Antilock Braking System (ABS) may be classified as follow. First, the vehicle braking-dynamics are nonlinear. Second, there are many unknown environmental parameters, like road coefficient of friction which may be wetty, snowy or dry. Finally, parameter changes due to mechanical wear and aging. Recently, a great deal of research has been performed on the ABS [1-4]. Currently, most commercial ABS are based on look-up tabular approach [5]. These tables are calibrated through iterative laboratory experiments and engineering field tests.

The conventional PID controller for automated machines is widely accepted by industry. According to a survey reported in [6], more than 90% of control loops used in industry use PID. This is because PID controllers are easy to understand (has clear physical meanings i.e. present, past and predictive), easy to explain to others, and easy to implement. Unfortunately, many of the PID loops that are in operation are in continual need of monitoring and adjustment since they can easily become improperly tuned [7]. Generally speaking, in order to meet the demands of real time operation, self-tuning is necessary.
Motivated by the success of fuzzy controllers in controlling nonlinear, complex, time-varying dynamic process in real world, there has been steep increase in the research work on the theoretical aspects of Fuzzy Logic Controller (FLC). The main reason is that FLCs essentially incorporate human expertise in the control strategy, exploiting easier understanding of linguistic interpretation.

Among the different types of FLC structures, PI-type and PD-type FLCs are very common [7-8]. Other related works have employed different adaptation policies to improve one or more performance indices [9-11]. However, development of PID type FLCs was not that popular because they need the construction of three dimensional rule-base, which complicates the design. Moreover, to make PID type FLC adaptive in nature, the number of free adaptable parameters increases and their interaction and interdependence further complicates the situation [12].

The area of auto-tuning of PID controller using fuzzy systems has attracted many authors [11-19]. A fuzzy self-tuning incremental PID controller has been proposed by He et al. [13]. The controller implements a conventional PID structure, which starts its operation with values of proportional gain, integral time, and derivative time, obtained from the well known Ziegler-Nichols (Z-N) tuning formula. This scheme implements a supervisory fuzzy system which adaptively changes the control parameter after each sampling instant to improve the control performance. In another study, a survey for tuning PID controllers with fuzzy logic has been made by Visioli [14]. In his work, several PID control schemes have been simulated for linear systems. Although in most cases, the controllers showed superior performance for linear processes, they could only reduce the peak overshoot at the expense of increasing rise time, and the degradation becomes more and more significant with increasing time delay. This restricted the overall acceptance as a good controller mechanism [15].

Referring to aforementioned works, one may conclude that Z-N method has been widely accepted as a base for tuning PID controllers off- and on-line. This may be referred to the ability of the method to preserve good load disturbance attenuation. Most of these studies however, have transformed the tuning problem from using the Z-N tuning parameters to other author-defined parameters and use fuzzy logic as the tuning tool for the new parameters; [13-17, others]. This indicates the lack of a generalized approach which can be followed for linear and nonlinear plants. Furthermore, the suitability and application range of the Z-N method are very limited [19, 20]. For example, it is not suitable for plants with a delay to time-constant ratio smaller than 0.15 or larger than 0.6. This method also yields poor damping and high sensitivity and does not achieve robustness of the closed-loop when considerable parameter variations take place.

In this study, a self-tuning PID controller is proposed for the ABS. Controlling the braking torque of ABS is necessary to avoid locking of the wheels so that the driver can keep control on the vehicle’s motion. Parameter variations and uncertainties imply the need for an auto-tuning operation to achieve the performance consistency. The article describes a generalized procedure for the development of a simple, model free fuzzy PID-type structure as an effective combination of three independent fuzzy systems. Each PID parameter is tuned via first order Takagi-Sugeno (T-S) fuzzy system, whose parameters are optimally determined off-line using a modified Genetic Algorithm (GA). The control goal is to keep the slipping ratio of the tires within desired range when braking is requested by the driver. With this approach, the PID controller can be automatically adjusted to meet the system uncertainties and achieve satisfactory response. Robustness against road conditions is examined via numerical tests and results are compared with previous works. This paper is organized as follows. Section 2 presents the mathematical model of ABS based on quarter car model. Section 3 introduces previous works in the area of tuning PID controllers. Section 4 demonstrates the proposed PID self-tuning scheme based on T-S fuzzy system. Learning of the fuzzy systems’ data base is achieved via optimal performance indices using a modified genetic algorithm. Simulation results and discussion are given in
section 5. Section 6 offers our concluding remarks.

2. Modeling of the antilock braking system

2.1. The quarter vehicle dynamic model

To verify the control performance, this Section demonstrates a simplified model for the quarter vehicle dynamic motion. It was derived by Assadin and Nouillant [1-2]; fig. 1. The nonlinear dynamics can be described as follows. The force balance in the longitudinal direction:

\[ m \alpha_x = \mu_r F_N. \]  

The slip ratio is defined by:

\[ \lambda = \frac{V_x - \omega R}{V_x}. \]  

Summing torques about the wheel center:

\[ J_\omega \alpha = -u + \mu_r R F_N, \]  

using (1, 2) and rearranging for \( \lambda \) yields:

\[ \lambda = \frac{\mu_r F_N}{V_x} \left( 1 - \frac{\lambda R^2}{J_\omega} \right) + \frac{R}{JV_x} u, \]  

where:

- \( m \) is the quarter vehicle mass, kg,
- \( a_x \) is the vehicle body acceleration, m/s²,
- \( V_x \) is the speed of the vehicle, m/s,
- \( F_N \) is the normal force, N,
- \( \alpha \) is the angular acceleration of the wheel, rad/s²,
- \( \omega \) is the angular velocity of the wheel, rad/s,
- \( \mu_r \) is the road coefficient of friction,
- \( R \) is the wheel radius, m
- \( u \) is the braking torque, N.m,
- \( \lambda \) is the slip ratio, and
- \( J_w \) is the wheel inertia, kg.m².

In eq. (2), the vehicle speed \( V_x \) and the wheel angular velocity are available signals through transducers mounted on suitable places. So that the slip ratio \( \lambda \) is available parameter for the ABS closed loop system.

2.2. Braking basics and problem definition

The ability of the ABS to maintain vehicle stability and steerability, and still produce shorter stopping distances than those from a locked wheel stop, comes from the shape of the adhesion coefficient \( \mu_r \) versus wheel slip ratio \( \lambda \). The friction coefficient can vary in a very wide range, depending on factors like:

![Fig. 1. The quarter vehicle dynamic model.](image)

![Fig. 2. Coefficient of road friction versus wheel slip ratio.](image)
a) Road surface conditions (dry or wet),
b) Tire side-slip angle,
c) Tire brand (summer tire, winter tire), and
d) The slip ratio between the tire and the road.

Dependence of the road friction coefficient on surface conditions and slip ratio is shown in fig. 2, [3]. The lateral force is essential to the steering of vehicle. It is obvious when slipping is equal to one, this force is equal to zero, which explains why the steering ability is lost during wheel lockup. The effective coefficient of friction between the tire and the road has the optimum performance when wheel slip ratio is at the range 0.18 → 0.22, and the worst performance occurs at $\lambda = 1$ (locked wheel). Most manufacturers use a set point for the slipping ratio $\lambda_d$ equal to 0.2 which is a good compromise for all road conditions [5]. So that, the control problem can be described as a set-point control system, that may implement PID controller; Fig. 3. Next Sections describes several PID strategies for this closed loop system.

3. PID tuning with fuzzy logic

In the early 1940’s, after extensive manual experimentation by the way of trial and error, Ziegler and Nichols invented the well-known Z-N formula for off-line tuning [21]. In time domain for example, the method yielded PID coefficients directly from the three important parameters of a stable plant to be controlled, namely the plant gain, time-constant and transport delay. These parameters can be easily obtained graphically from a step response of the open loop plant. The concept of fuzzy systems instead, may be used to emulate human expertise and on-line tune the control gains using Z-N formula. The coming Sub-sections demonstrate some of the previous works in this area. They are used in later Sections for comparison purposes.

3.1. Standard and nonlinear PID controllers

There are many types of PID controllers, e.g., PID plus gravity compensator, PID plus friction compensator, PID plus disturbance observer, etc [22]. Here, we shall consider the basic forms for PID controller which is placed in a unity feedback control system. A typical PID control law in its standard form is

$$u(t) = K_p \left[ e(t) + T_d \frac{de(t)}{dt} + \frac{1}{T_i} \int_0^t e(t)dt \right], \quad (5)$$

where $u(t)$ is the control variable, $e(t) = \lambda_d(t) - \lambda(t)$ is the system error (difference between the demand input $\lambda_d$ and the system output, $\lambda$), $K_p$ is the proportional gain, $T_d$ the derivative time constant and $T_i$ is the integral time constant. Eq. 5 can be rewritten as

$$u(t) = K_p e(t) + K_i \int e(t)dt + K_d \frac{de(t)}{dt}. \quad (6)$$

$K_d = K_p T_d$ the derivative gain and $K_i = K_p / T_i$ the integral gain.

In the design and tuning of a PID controller, the P, I and D actions need to be co-ordinated. This is not a trivial process, since coefficients of these three actions interact mutually and cannot be simply tuned individually in a de-coupled manner. An experiment carried out on the process using Z-N tuning method, can be stated as follows [6,21]. First, the process is controlled using the proportional gain $K_p$. The value of $K_p$ is slowly increased until continuous oscillations is happened. At the time of oscillation, the values of the gain $K_u$ and the oscillation period $t_u$ are noted. The method assumes that $K_p$ is 60% of the gain at time of oscillation. The integral time constant $T_i$ is 50% of the oscillation period and $t_u$ and the derivative time constant is 12.5% of the oscillation period. The Z-N method is devised for off-line tuning of continuous systems and can also be used on discrete cases for a fast sampling time.
Nonlinear PID controllers have been widely considered in literature. A large class can be generalized as follows [6, 12, 14]:

\[ u(t) = K_p f(e, \dot{e}) + K_d \frac{de(t)}{dt} + K_i \int_0^t e(\tau)d\tau , \]  

where \( f(e, \dot{e}) \) is a nonlinear function depends on the closed-loop error and the process delay [12]. The \( f(e, \dot{e}) \) may be represented by a fuzzy logic, in which the inputs are \( e(t) \) and \( \dot{e}(t) \); [18]. It comes out that the proportional gain depends on the current error and other parameters contained on \( f(e, \dot{e}) \). Effectiveness of this nonlinear PID has been discussed in [12] for linear systems.

### 3.2. Fuzzy Self tuning of a Single Parameter (SSP)

The scheme has been widely considered, which may be referred to its simplicity [13-15]. The method devised by He et al. [13], consists of parameterizing the Ziegler-Nichols formula by means of a single parameter \( \alpha \), then using an online fuzzy inference system to self-tune \( \alpha \). By this method, the three PID parameters can be expresses as:

\[ K_p = 1.2\alpha(t)K_u \]
\[ T_i = 0.75 \frac{1}{1+\alpha(t)}t_u \]
\[ T_d = 0.25T_i \]

where \( K_u \) and \( t_u \) are the ultimate gain and ultimate period, respectively, of the process. The value of \( \alpha(t) \) is determined recursively with the following equation:

\[ \alpha(t+1) = \begin{cases} 
\alpha(t) + \gamma h(t)(1-\alpha(t)) & \text{for } \alpha(t) > 0.5 \\
\alpha(t) + \gamma h(t)\alpha(t) & \text{for } \alpha(t) \leq 0.5 
\end{cases} , \]  

where \( \gamma \) is a positive constant that has to be chosen in the range \([0.2, 0.6]\) and \( h(t) \) is the output of the fuzzy inference system which has seven membership functions both for each of the two input \( (e, \dot{e}) \) and for the output.

The initial value of \( \alpha(t) \) is set equal to 0.5 which corresponds to the Ziegler-Nichols formula. With respect to the method however, the tuning of the scaling coefficient of the fuzzy modules and of the parameter \( \gamma \) is left to the user and no rule of thumb is given for this task.

### 3.3. Fuzzy Set-point Weighting (FSW)

This approach proposed by Visioli [16] consists of fuzzifying the set-point weight, leaving fixed the other three parameters (again, determined with the Ziegler-Nichols method). In this way, the control law can be written as

\[ u(t) = K_p [b(t)\lambda(t) - \lambda(t)] + K_d \frac{de(t)}{dt} + K_i \int_0^t e(\tau)d\tau \]

\[ b(t) = w + f(t) , \]  

where \( w \) is a positive constant less than or equal to 1, and \( f(t) \) is the output of the fuzzy inference system, which consists of five triangular membership functions for each of the two inputs \( e(t) \) and \( \dot{e}(t) \) and nine triangular membership functions for the output. The fuzzy rules are based on the Macvicar-Whelan matrix [17], as shown in Table 1, where the linguistic variables negative small NS, negative medium NM, negative big NB, negative very big NVB, zero Z, positive small PS, positive medium PM, positive big PB, and positive very big, PVB.

The method however, is not a straight forward and large number of arithmetic operations is needed for on-line tuning.

<table>
<thead>
<tr>
<th>Table 1</th>
<th>Rule table of the FSW [17]</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>( f(t) )</th>
<th>( f(t) )</th>
<th>( f(t) )</th>
<th>( f(t) )</th>
<th>( f(t) )</th>
<th>( f(t) )</th>
<th>( f(t) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>NB</td>
<td>NB</td>
<td>NB</td>
<td>NB</td>
<td>NB</td>
<td>NB</td>
<td></td>
</tr>
<tr>
<td>NB</td>
<td>NB</td>
<td>NB</td>
<td>NB</td>
<td>NB</td>
<td>NB</td>
<td></td>
</tr>
<tr>
<td>NB</td>
<td>NB</td>
<td>NB</td>
<td>NB</td>
<td>NB</td>
<td>NB</td>
<td></td>
</tr>
<tr>
<td>NB</td>
<td>NB</td>
<td>NB</td>
<td>NB</td>
<td>NB</td>
<td>NB</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Error, ( e )</th>
<th>Z</th>
<th>NM</th>
<th>NS</th>
<th>Z</th>
<th>PS</th>
<th>PM</th>
</tr>
</thead>
<tbody>
<tr>
<td>PS</td>
<td>NS</td>
<td>Z</td>
<td>PS</td>
<td>PM</td>
<td>PB</td>
<td>PVB</td>
</tr>
</tbody>
</table>
3.4. Incremental Fuzzy Expert PID control (IFE)

Tzafestas and Papanikolopoulos [18], proposed to scale the values of the three control parameters (initially determined by the Ziegler-Nichols formula) during the transient response depending on the system error and its rate. In other words, the current values of the proportional, integral and derivative gains are increased or decreased by means of a fuzzy inference system according to the following relations:

\[
K_p = K_p + FS[c_1e(t),c_2\dot{e}(t)]\times k_1 \\
K_i = K_i + FS[c_1e(t),c_2\dot{e}(t)]\times k_2 \\
K_d = K_d + FS[c_1e(t),c_2\dot{e}(t)]\times k_3 ,
\]

where the basic tuning is the Ziegler-Nichols one, \(FS[c_1e(t),c_2\dot{e}(t)]\) is the output of fuzzy inference system, based on Macvicar-Whelan fuzzy rule matrix [17], which reflects the typical action of a human control. For example, the integral action has to be increased at the beginning of the transient response to decrease the rise time and then has to be decreased when the system error is negative to reduce the overshoot. The range of input membership functions is assumed to be normalized between -1 and 1. Finally, \(c_1, c_2\) are scaling factors and \(k_i, i = 1, 2, 3\) are constant parameters that determine the range of variation of each term. The whole fuzzy system involve 14 quantization levels for both \(e\) and \(\dot{e}\). Similar approach has been followed in [15], where the authors have to rely on trial and error procedure in order to identify parameters of the fuzzy systems.

However, tuning the three parameters \(k_i\) and the two scaling factors that multiply the inputs (\(e\) and \(\dot{e}\)) is left to the user, and it is not clear how these parameters influence the performance of the overall controller.

4. The proposed fuzzy self-tuning PID control scheme

In this Section, the proposed scheme is introduced. It does not rely on Z-N method or on a previous rule base especially designed for some plants. Instead, it uses first order T-S fuzzy systems as the tuning-tool for each of the PID control modules. A modified genetic algorithm is used to optimally select parameters of the fuzzy systems. Optimality is based on conventional performance indices; i.e. Integrated Absolute Error (IAE) and the Integrated Time multiplied by the Absolute Error (ITAE). In this way, the proposed scheme can be devised for any linear or nonlinear system in a straight forward manner. Furthermore, it uses simple rule base whose parameters can be learned off-line. Unlike the work of [23] which uses the control signal (the desired value is assumed known), the search of suitable (optimal) parameters in this work is based on the closed loop performance. For example, in IAE, the hatched area in Fig. 4 represents the performance index. A minimum area is achieved at the fastest physically possible response, no/small overshoot and close to zero steady state error. It means that decreasing overshoot will not result in an increase in the rise time or vice versa, as it is the case with Z-N method. The expected tuning result is simply the best physically possible response.

4.1. The control system architecture

Motivated by the work of [15, 18], three decoupled fuzzy systems constitute the proposed self-tuning system; each for one parameter of the PID controller, i.e. \(K_p, K_i\) and \(K_d\); eq. (6). The error \(e\) and change in error \(\dot{e}\) are used as behavior-recognizers of the closed loop performance. They are available signals in the closed loop system of the ABS and do not require extra hardware. The self-tuner can be expressed as:

\[
A = \int_0^T |e(t)|dt
\]

Fig. 4. A typical closed-loop error time-history.
where, a fuzzy P, fuzzy I and fuzzy D modules are connected in parallel to give the resultant controller signal. With this structure, independent control actions can be generated which should necessarily eliminate the problems associated with most practical two-terms or three-terms FLCs. The basic approach is summarized in fig. 5, where each fuzzy module is trying to recognize when the corresponding parameter is not properly tuned and then seeks to adjust it to obtain improved performance. In such a way, each fuzzy system can be looked at as gain scheduler (module). T-S type fuzzy systems are used to synthesize each module.

The T-S fuzzy system (also known as functional fuzzy system [7]) was proposed in an effort to develop a systematic approach to generating fuzzy rules from a given input-output data set; [24, 25]. A typical rule has the following form

\[
\text{IF } \text{x}_1 \text{ IS } A \text{ AND } \text{x}_2 \text{ IS } B \text{ THEN } z = f(\text{x}_1, \text{x}_2)
\]

where \( A \) and \( B \) are fuzzy sets in the antecedent, while \( z = f(\text{x}_1, \text{x}_2) \) is a crisp function in the consequent. With this form, the fuzzy system can be characterized as two input one output fuzzy systems.

Usually \( f(\text{x}_1, \text{x}_2) \) is a polynomial in the input variables \( \text{x}_1 \) and \( \text{x}_2 \), but it can be any function as long as it can appropriately describe the output of the model within the fuzzy region specified by the antecedent of the rule. Although there are no restrictions on the form of the input membership functions, Gaussians are used in the premise through out this work. A Gaussian function is specified by two parameters (\( c, \sigma \)):

\[
\mu_{A_i}(x_j) = \text{gaussian}(x_j; c, \sigma) = \exp\left(-\frac{1}{2} \left(\frac{x_j - c}{\sigma}\right)^2\right),
\]

where \( \mu \) is the membership grade, \( c \) represents the membership function’s center, \( \sigma \) determines its spread; \( A \) is the membership which represents a linguistic variable, \( i=1,2,...,n \) is the rule number, \( j = 1, 2 \) is subscript of the input variables.

In the proposed self-tuner, the inputs i.e. \( e \) and \( \dot{e} \) are normalized using three Gaussian membership functions; negative N, zero Z, and positive P. So that nine rules constitute the rule base for each module. For simplicity, the consequent part has been chosen to be a first order function of \( e \) and \( \dot{e} \). So that the rule bases have the following form:

\[
K_p = FS_1[e(t), \dot{e}(t)]
\]
\[
K_i = FS_2[e(t), \dot{e}(t)]
\]
\[
K_d = FS_3[e(t), \dot{e}(t)]
\]

(12)
Rule: IF $e$ IS $A$ and $\dot{e}$ is $B$ then

$$K^i = a_{i,j1}e + a_{i,j2}\dot{e} + a_{i,j3},$$  

(14)

where both $A$ and $B$ are positive (P), zero (Z) or negative (N), $K^i = f(e, \dot{e})$ is the gain to be tuned, i.e. $K_p$, $K_i$ or $K_d$, $a_{i,j1}$, $a_{i,j2}$ and $a_{i,j3}$ are the constants, and $j=1, 2, \ldots 9$ is the rule number.

Fig. 6 shows the fuzzy reasoning procedure for a first order T-S model. The fuzzy part is only in its antecedent. Since each rule has a crisp output, the overall output is obtained via weighted average, thus avoided the time-consuming process of defuzzification required in a Mamdani fuzzy model. The weighted sum has been used in [25] to further reduce computation.

With the above structure, the data-base of each module consists of 27 free parameters (coefficients of the first order polynomial) and 12 free parameters for the inputs membership functions. So that the total number of free parameters is 117. If identical input membership is chosen for the three modules, the total number of parameters is reduced to 93; i.e. parameters of the input membership functions $c_{N1}, \sigma_{N1}, c_{Z1}, \sigma_{Z1}, c_{P1}, \sigma_{P1}$ for $e$, $c_{N2}, \sigma_{N2}, c_{Z2}, \sigma_{Z2}, c_{P2}, \sigma_{P2}$ for $\dot{e}$, and $a_{jk}$ where $i = 1, 2, 3$ are the coefficients of the first order polynomial, $j = 1, 2, \ldots 9$ the rule number for each tuner and $k = 1, 2, 3$ the number of modules. Determination of optimal values for these parameters is the subject of next subsection.
4.2. Genetic algorithm-based parameter learning

GAs are optimization stochastic technique mimicking the natural selection, which consists of three operations, namely, reproduction, crossover, and mutation [25]. The most general considerations about GA can be stated as follows:

i) The searching procedure of the GA starts from multiple initial states simultaneously and proceeds in all of the parameter subspaces simultaneously.

ii) GA requires almost no prior knowledge of the concerned system, which enables it to deal with the completely unknown systems that other optimization methods may fail.

iii) GA can not evaluate the performance of a system properly at one step. For this reason, it can generally not be used as an on-line optimization strategy and is more suitable for fuzzy modeling.

In practice, training data can be obtained by experimentation or by the establishment of an ideal model. In this work, the ABS model in Section 2 is used to emulate the behavior of the ABS in order to collect training data. Fig. 7 shows the training process for each fuzzy module involved in the self-tuning system.

The following two closed-loop performance indices have been examined; the Integrated Absolute Error (IAE) and the integrated time multiplied by the absolute error (ITAE). They are defined as follows:

\[
J_{IAE} = IAE = \int_0^T |e(t)| dt = \sum_{k=0}^M |\lambda_d(k) - \lambda(k)| \Delta t. 
\]  
(15)

\[
J_{ITAE} = ITAE = \int_0^T |e(t)|^2 dt = \sum_{k=0}^M |\lambda_d(k) - \lambda(k)|^2 \Delta t^2, 
\]  
(16)

where \( e = \lambda_d - \lambda \) is the closed loop error, \( \lambda_d \) is the desired slip ratio, \( \lambda \) current slip ratio and \( \Delta t \) is the time step. \( M \) is the number of training samples. Because GA endeavors to maximize the fitness function, the fitness function of each gene is calculated as follows:

\[
F = \frac{1}{1 + J}, 
\]  
(17)

where \( J \) is the performance index and 1 is introduced at the denominator to prevent the fitness function from becoming infinitely large.

![Diagram of Genetic learning of the data-base of each module.](image)
To simplify the presentation, let us denote $F$ in eq. (17) as $F_{LAE}$ when $J = J_{LAE}$ and $F_{ITAE}$ when $J = J_{ITAE}$. With this notation, the controller is called PID-IAE when the fitness function is the performance index IAE, eq. (15) and PID-ITAE when the performance index is ITAE, eq. (16).

Coding of the parameters to be adjusted can be stated as follows:

$$c_{N1}, c_{N2}, ... \sigma_{N1}, \sigma_{N2}, ... a_{111}, a_{121}, ... a_{339},$$

where a certain number of each binary bits stand for each element. The combined string composes a gene (possible solution) in a population. Evaluation of each possible solution is performed via IAE/ITAE and genes of best solutions are allowed to reproduce with higher rates.

Although, genetic algorithms were developed a few decades ago, concrete theoretical analysis of the algorithm have not been provided until recent years [26, 27]. ref. [27] concludes that the canonical GA cannot always find the optimal solution within definite time. Furthermore, the paper points out that if the chromosome with the best performance in each generation is reserved for the next generation, the algorithm will globally converge. Inspired by these conclusions, the following two measures have been followed in this work:

i) In reproduction, we stochastically introduce a randomly generated gene at a probability of $R_h$ to replace one of the two parents selected for reproduction.

ii) Select the best performed genes in the current population at a rate of $R_e$ and place them directly in the next generation.

If the reproduction is carried out in the traditional way, the best gene will globally be lost and thus convergence can not be guaranteed. However, if we only adopt the second measure, the procedure of the evolution is no longer a Markov process and thus it does not satisfy the assumption of the convergence theory. Despite our successful application of these measures, mathematical analysis of them still lacks. The parameters $R_h$ and $R_e$ are adjusted such that at the beginning of the learning, $R_h$ is relatively large and $R_e$ is relatively small, and later, vise versa.

5. Numerical tests

5.1. Simulation data

Several numerical tests have been performed using the ABS example data presented in [5]. They are: $R = 0.33m$, $m=342kg$, $J = 1.13 Nm^2$, $g = 9.81 m/s^2$ and the desired slip ratio $d = 0.2$. Due to the fact that the wheel and vehicle velocity are nearly zero at the end of braking time, the magnitude of slip tends to infinity. Therefore, simulations are conducted up to the point when the vehicle is slowed to 0.5 m/s.

The following case study is considered. It deals with braking on a dry road ($\mu_r=0.8$), then after 1.5 second, the road changed to an icy one ($\mu_r=0.1$). The initial speed $V_x= 41.67m/sec$, i.e. 150km/hr. These conditions have been considered in order to examine robustness against road conditions.

To verify the full potentialities of the investigated PID controllers, it is assumed that no saturation levels are imposed to the control signal. After tuning and/or training of the examined PID controllers, a saturation level $u_s = \pm 4000 Nm$ has been imposed to test the controllers in the presence of typical process nonlinearity. Training has been performed using the parameter settings listed in table 2. The output surfaces of each module before and after learning are depicted in fig. 8. As a sample of the tuning results, membership functions of $e$ and $e \cdot \dot{e}$ before and after learning are shown in fig. 9, when training has been performed with the fitness function $F_{LAE}$ (i.e. the antecedent part of the PID-IAE controller).

For the other investigated methodologies (SSP and FSW described in section 3), with the exception of Z-N PID controller, i.e., parameters have been determined using the GA with fitness $F_{LAE}$, as the fitness function. Furthermore, the genetic algorithm with the parameters listed in table 2 has been used to retune the Z-N parameters. It is intended for the resulted PID to use the best possible fixed parameters defined by Z-N formula. This PID controller is denoted by G-PID.
5.2. Computational considerations

The initial fuzzy modules were determined after small number of trials. It is our point of view that the initially guessed modules should work properly for successful learning process. Dealing with genetic learning, the total number of iteration \(N\) should be chosen so as the system has enough chance to converge. A suitable choice ensures correct training and saves computation time since each gene is a possible solution which has to be evaluated according to eq. (17). In literature, two approaches are generally used for selecting the suitable number of population and generation for optimization problems similar to the problem considered in this work. The first relies on a relatively small number of population (e.g. 20 \(\rightarrow\) 30) and high number of generation (e.g. 700 \(\rightarrow\) 3000) [28-30]. The second uses high number of population (e.g. 100 \(\rightarrow\) 200) and low number of generations [31]. Adaptive change of the crossover probability helps more in speeding up the convergence. The coming results have been obtained using genetic parameters listed in table 2.

With a population of \(V = 100\) individuals for \(G =150\) generations, the fitness eq. (17) is evaluated 15,000 times. Indeed, this number \((N= V\times G)\) represents the number of evaluated points inside the search space, which may be used as a reference for similar optimization problems. The winning gene (optimal solution) is the best of stochastically competitive 15,000 genes. Referring to (18), with 32 bits for each variable, each possible solution (gene) has the length of 2976 bits. Performing the learning process using Matlab-7, M-file under Windows XP on a PC Pentium IV, 2800 Hz speed, requires around 1 and half hour. Due to the stochastic nature of GAs, at least two or three trails should be performed in order to be sure that convergence has taken place.

<table>
<thead>
<tr>
<th>Genetic parameter</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of generations, (G)</td>
<td>150</td>
</tr>
<tr>
<td>Population size, (V)</td>
<td>100</td>
</tr>
<tr>
<td>Crossover probability (R_c, R_e)</td>
<td>0.90, 0.60</td>
</tr>
<tr>
<td>Mutation probability</td>
<td>0.01</td>
</tr>
<tr>
<td>Bit number for each variable</td>
<td>32</td>
</tr>
</tbody>
</table>

5.3. Results and discussions

Fig. 10 shows the ABS response under 7 investigated control schemes. As it can be noticed PID-IAE gives the best response compared with other schemes. With respect to constant gain schemes, G-PID shows better results than Z-N which shows the highest overshoot and longest settling time; i.e. the worst performance. To provide a more detailed insight of the results, table 3 gives the values of the performance indices IAE, ITAE, and maximum attained overshoot for the examined PID controllers.

The braking torques have significantly changed their magnitude after 1.5 seconds to meet the new road condition (icy road); figs. 11 (a and b). Low braking torque is required for the vehicle to move on icy road in order to avoid locking. Fig. 12 shows very close stopping times have achieved by the investigated controllers. However, larger overshoot can be noticed by Z-N and SSP with respect to PID-IAE.

Time history of the PID-IAE modules is depicted in fig. 13. Unlike constant gain scheme (e.g. Z-N), their values are continuously changing during the braking period, which is imposed by the fuzzy modules in order to counter attack the error and change in error. After 1.5 seconds, smooth transition on their values has taken place in order to meet the new road demands. This behavior resulted in a relatively smoother torque transition and faster convergence.

Because the input saturation level is significantly high, it may be interesting for the study to investigate the response when no saturation is imposed. Fig. 14 shows the response of 4 PID schemes. It comes that fuzzy logic auto-tuning is more useful if saturations are significant in the process. In fact, it appears that the performance of the classic PID can be improved using time varying parameters [22]. Among the considered schemes, the PID-IAE exhibits the best performance compared with other fuzzy-tuning-based schemes. This appears from results shown in fig. 14 although the attained IAE and ITAE values are closer when there is no saturation, table 4. As a performance measure, the percent overshoot when input
saturation is and is not imposed is depicted in fig. 15. It is clear that saturation has little or no effect on the fixed gain schemes, i.e. Z-N and G-PID. On the other hand, PID-IAE and PID-Fuzzy have been greatly influenced.

Fig. 8. The output surfaces of the fuzzy modules.

Fig. 9. Membership functions of the antecedent part before (----) and after (—) training using $F_{IAE}$. 

Table 3
Performance of PID tuning methods

<table>
<thead>
<tr>
<th>Controller type</th>
<th>Performance measure</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>IAE</td>
</tr>
<tr>
<td>Z-N</td>
<td>0.1170</td>
</tr>
<tr>
<td>G-PID</td>
<td>0.0262</td>
</tr>
<tr>
<td>SSP</td>
<td>0.064</td>
</tr>
<tr>
<td>FSW</td>
<td>0.0382</td>
</tr>
<tr>
<td>PID-fuzzy</td>
<td>0.045</td>
</tr>
<tr>
<td>PID-ITAE</td>
<td>0.0432</td>
</tr>
<tr>
<td>PID-IAE</td>
<td>0.0422</td>
</tr>
</tbody>
</table>

Fig. 10. Slip ratio under different control laws.

Fig. 11. The input torques.
Fig. 12. Performance of ABS under three PID strategies.

Fig. 13. Time history of the PID-IAE gains (before imposing saturation level on \( u \)).

Fig. 14. Slip ratio under different control schemes when saturation on the input is not imposed.

Table 4

<table>
<thead>
<tr>
<th>Controller type</th>
<th>Performance measure</th>
<th>Maximum O.S%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Z-N</td>
<td>0.1035</td>
<td>59.47</td>
</tr>
<tr>
<td>G-PID</td>
<td>0.0262</td>
<td>22.7</td>
</tr>
<tr>
<td>SSP</td>
<td>0.041</td>
<td>21.45</td>
</tr>
<tr>
<td>FSW</td>
<td>0.0226</td>
<td>15.95</td>
</tr>
<tr>
<td>PID-fuzzy</td>
<td>0.0572</td>
<td>88.78</td>
</tr>
<tr>
<td>PID-ITAE</td>
<td>0.02</td>
<td>20.76</td>
</tr>
<tr>
<td>PID-IAE</td>
<td>0.059</td>
<td>73.21</td>
</tr>
</tbody>
</table>

Performances of PID tuning methods when saturation is not imposed.
Finally, one should consider the case of tuning the controllers i.e. of the fuzzy module’s parameters, since in practical industrial applications; the use of genetically trained controllers is possible if a good process model is available. It emerges that, for the fuzzy-logic-based tuning methods for which a technique for the selection of the parameters of the fuzzy logic has not been considered, it might be difficult to perform this task effectively. For example, for the IFE method (sub-section 3.4), the choice of parameter $k_3$ is critical since its value has to be kept very low, otherwise the overall control system can be unstable [16]. Also, the SSP control structure is very difficult to set, as it depends on results obtained from GA, which is not always easy to improve the Z-N response and, when it succeeds the selected rules is very difficult to interpret because other tuning parameters interfere ($\gamma$ and $\alpha$). Manual tuning of the proposed scheme is straightforward since it depends on three decoupled modules and setting of the fuzzy modules can be done in a manner similar to that of the Z-N, which is advantageous over most of the investigated schemes.

6. Conclusions

In this article, a fuzzy self-tuning scheme has been proposed for PID controllers. The proposed scheme utilizes three decoupled modules, each for one of the PID parameters. Each module is two-input one-output T-S type fuzzy system. Optimal selection of the fuzzy modules has been obtained using a modified genetic algorithm. The performance has been verified using the automobiles’ ABS. In the presented case study, robustness against road friction characteristics has been considered. The control goal is to keep the slip ratio to an assigned value (0.2) despite road friction characteristics. Comparison with previous works shows the competitiveness of the proposed scheme.

The salient features can be summarized as follows:
1. The proposed control scheme presents a generalized procedure which can be followed for linear and nonlinear systems.
2. Improved control action has been obtained by replacing the Z-N tuned PID controllers with fuzzy self-tuning systems, as it is the case in SSP and FSW proposed by earlier investigators, and can be further enhanced using the proposed control scheme.
3. The proposed PID controller resulted in relatively fast response with low overshoot. As a consequence, there is a remarkable enhancement with regard to other examined PID controllers.
4. An advantage of the proposed scheme is the ability to switch on/off any of the control actions. The basic decoupled nature facilitates such possibility. This procedure is needed for practical implementation because of the drift which usually exists between theoretical establishment and actual experimentation.
5. A probable area of future work is to achieve adaptivity of each/some of the three control actions, while maintaining its simplicity.

References

S. Abdel Badie / Genetic fuzzy self-tuning


Received September 3, 2006
Accepted November 9, 2006