The self-tuning PID control in a slider–crank mechanism system by applying particle swarm optimization approach

Chih-Cheng Kao a, Chin-Wen Chuang b, Rong-Fong Fung a,∗

a Department of Mechanical and Automation Engineering, National Kaohsiung First University of Science and Technology, 1, University Road, Yenchau, Kaohsiung 824, Taiwan, ROC

b Department of Electrical Engineering, I-Shou University, 1, Section 1, Hsueh-Cheng Road, Ta-Hsu Hsiang 840, Kaohsiung, Taiwan, ROC

Received 14 March 2006

Abstract

By using the particle swarm optimization (PSO) algorithm, a novel design method for the self-tuning PID control in a slider–crank mechanism system is presented in this paper. This paper demonstrates, in detail, how to employ the PSO so as to search efficiently for the optimal PID controller parameters within a mechanism system. The proposed approach has superior features, including: easy implementation; stable convergence characteristics; and good computational efficiency. Fast tuning of optimum PID controller parameters yields high-quality solutions. By using the PSO approach, both the initial PID parameters under normal operating conditions and the optimal parameters of PID control under fully-loaded conditions can be determined. The proposed self-tuning PID controller will automatically tune its parameters within these ranges. Moreover, the PC-based controller is implemented to control the position of the motor-mechanism coupling system. In order to prove the performance of the proposed PSO self-tuning PID controller, the responses are compared with those by the real-coded genetic algorithm (RGA) PID controller and the fixed PID controller. The numerical simulations and experimental results will show the potential of the proposed controller.

Keywords: Slider–crank mechanism system; Self-tuning PID controller; Particle swarm optimization; Optimal control; Real-coded genetic algorithm PID controller

1. Introduction

During the past decades, process control techniques in the industry have made great advances. Numerous control methods such as: adaptive control; neural control; and fuzzy control have been studied [1–5]. Among these the best known is the proportional integral derivative (PID) controller [7–9], which has been widely used in the industry because of its simple structure and robust performance within a wide range of operating conditions. Unfortunately, it has been quite difficult to properly tune the gains of PID controllers because many industrial plants are often burdened with problems such as: high orders; time delays; and nonlinearities [1–6]. Over the years, several heuristic methods have been proposed for the tuning of PID controllers. The first method used the classical tuning rules proposed by Ziegler and Nichols [1–3]. In general, it is often hard to determine optimal or near optimal PID parameters with the Ziegler–Nichols formula in many industrial plants [1–3].

For this reason it is highly desirable to increase the capabilities of PID controllers by adding new features. Many artificial intelligence (AI) techniques have been employed to improve the controller performances for a wide range of plants whilst retaining their basic characteristics. AI techniques such as: neural networks; fuzzy systems; and neural-fuzzy logic have been widely applied to the proper tuning of PID controller parameters [1,2].

Particle swarm optimization (PSO), first introduced by Kennedy and Eberhart [10,11], is one of the modern
heuristic algorithms. The PSO technique can generate a high-quality solution within shorter calculation time and stable convergence characteristics than other stochastic methods [12–14]. Much research is still in progress for proving the potential of the PSO in solving complex power system operation problems. Because the PSO method is an excellent optimization methodology and a promising approach for solving the self-tuning PID controller parameters problem [15–17], this controller is called the PSO self-tuning PID controller.

However, the PID controller is not robust to wide parameter varying and large external disturbance. This serves especially for the highly coupling nonlinear system where the PID controller lacks adaptive capability. To achieve practical requirement, engineers have to adjust the parameters under different operating conditions. However, the robustness is limited to a small range. A rule to overcome this disadvantage is called the self-tuning rule. The parameter tuning at any instance is usually based on a structurally fixed mathematical model produced by an on-line identification procedure [18,19]. Unfortunately, recent plants find it difficult to obtain their fixed mathematical models. This paper proposes an intelligent self-tuning PID controller. Engineers should easily accept the straightforward design procedure. At the same time, the robustness will be expanded to a large range.

In this paper, a practical high-order mechanism system with a PID controller is adopted to test the performance of the proposed PSO self-tuning PID controller. This paper proposes a self-tuning PID control method to the position control of a slider–crank mechanism. As compared with the RGA approach [5,11,24,25], it is found that the nominal values and tuning ranges of the PID parameters by the PSO approach can be accurately determined. Numerical simulations and experimental results show the proposed controller is more robust than a fixed PID controller.

2. Slider–crank mechanism

2.1. PM synchronous motor

A model of a PM synchronous motor can be simplified and its block diagram is shown in Fig. 1. Usually, the PM synchronous motor is coupled with a gear speed reducer with a gear ratio of $n$. Hence, the applied torque can be described as [20]

$$\tau_e = \tau_m + B_m \omega_r + J_m \dot{\omega}_r,$$

where $\tau_m$ is the load torque, $B_m$ is the damping coefficient, $\omega_r$ is the rotor speed and $J_m$ is the moment of inertia.

2.2. Slider–crank mechanism

In this section, Hamilton’s principle and Lagrange multiplier are used to derive the differential equation for the slider–crank mechanism, which is shown in Fig. 2 [20].

The slider–crank mechanism consists of three parts: a crank, a rod and a slider. This section establishes a dynamic model of the slider–crank mechanism using the Lagrange method. Through reduction and incorporation, we can derive the dynamic equation as follows [21,22]:

$$M(\dot{\psi}) \ddot{\psi} + N(\psi, \dot{\psi}) - BU - D(\psi) + \Phi^T \lambda = 0,$$

where

$$\Phi = [r \cos \theta - l \cos \phi, \dot{\psi} \sin \theta], \quad B = [-nK_r, 0],$$

$$D(\psi) = \left[ \frac{F_{ser} \sin \theta}{F_{ser} \sin \phi} \right], \quad U = [u],$$

$$M(\dot{\psi}) = \begin{bmatrix} -\frac{l}{2} m_1 R_l^2 - (m_2 + m_3) r^2 \sin^2 \theta - n^2 J_m & -\frac{1}{2} m_2 \dot{r} l \sin \theta \sin \phi \\
-\frac{1}{2} m_3 \dot{r} l \sin \theta \sin \phi & -\frac{1}{2} m_3 l^2 - m_3 \dot{r}^2 \sin^2 \phi \end{bmatrix},$$

$$N(\psi, \dot{\psi}) = \begin{bmatrix} -(m_2 + m_3) r^2 \sin \theta \cos \theta - (m_2 + m_3) r \dot{\psi}^2 \sin \sin \phi - n^2 B_m \dot{\theta} \\
-(m_2 + m_3) r \dot{\psi}^2 \sin \sin \phi - m_3 \dot{r}^2 \phi^2 \sin \phi \cos \phi - \frac{1}{2} m_2 g \cos \phi \end{bmatrix}.$$

In which $m_1$, $m_2$, and $m_3$ are the masses of the crank, rod and slider, respectively; $r$ and $l$ are the lengths of the crank and rod; and $\theta$ and $\phi$ are angles of the crank and rod; $\lambda$ is the Lagrange multiplier, respectively.

The translation position $x_b$ can be obtained by transforming $\dot{\theta}$ as

$$x_b = r \cos \theta + l \cos \phi + \dot{l}$$

$$= r \cos \theta + (\dot{l}^2 - r^2 \sin^2 \theta) \dot{\theta} + \dot{l},$$

where $\dot{l}$ is half length of slider.

3. Particle swarm optimization

In 1995, Kennedy and Eberhart [10] first introduced the particle swarm optimization (PSO) method, which is derived from the social-psychological theory, and has been found to be robust in complex systems. Instead of using evolutionary operators to manipulate the individual particle each particle is treated as a valueless particle in $g$-dimensional search space, and keeps track of its coordinates in the problem space associated with the best solution (evaluating value) [11–14]. This value is called $pbest$. Another best value that is tracked by the global version of the particle swarm optimizer is the overall best value, and its location, obtained so far by any particle in the group, this is called $gbest$.

The PSO concept consists of changing the velocity of each particle toward its $pbest$ and $gbest$ locations. For example [13,14], the $j$th particle is represented as $x_j = (x_{j,1}, x_{j,2}, \ldots, x_{j,g})$ in the $g$-dimensional space. The best previous position of the $j$th particle is recorded and repre-
sent as $pbest_j = (pbest_{j1}, pbest_{j2}, \ldots, pbest_{jg})$. The index of best particle among all particles in the group is represented by the $gbest_g$. The rate of the position change (velocity) for particle $j$ is represented as $v_{j} = (v_{j1}, v_{j2}, \ldots, v_{jg})$. The modified velocity and position of each particle can be calculated using the current velocity and distance from $pbest_{jg}$ to $gbest_g$ as shown in the following formulas [14,15]:

$$v_{j}^{(t+1)} = w \cdot v_{j}^{(t)} + c_1 \cdot \text{rand}(\cdot)(pbest_{j} - x_{j}^{(t)}) + c_2 \cdot \text{rand}(\cdot)(gbest_{g} - x_{j}^{(t)}),$$

$$x_{j}^{(t+1)} = x_{j}^{(t)} + v_{j}^{(t+1)},$$

where $n$ is the number of particles in a group; $m$ is the number of members in a particle; $t$ is the pointer of iterations (generations); $v_{j}^{(t)}$ is the velocity of the particle $j$ at iteration $t$, $v_{j}^{\min} \leq v_{j}^{(t)} \leq v_{j}^{\max}$; $w$ is the inertia weight factor; $c_1$, $c_2$ is the acceleration constant; Rand() is random number generator. $n$ is the number of members in a group; $gbest_{g}$ is the $g$ best of the group $g$.

In the above procedures, the parameter $V_{\text{max}}$ determined the resolution, or fitness, with which regions were searched between the present position and the target position. If $V_{\text{max}}$ is too high, particles might fly past good solutions. If $V_{\text{max}}$ is too low, particles may not explore sufficiently beyond local solutions.

The PSO algorithm [15,16] was mainly utilized to determine three optimal PID controller parameters: $k_p$, $k_i$, and $k_d$. We use the “individual” to replace the “particle” and the “population” to replace the “group” in this paper. We defined three controller parameters to compose an individual $K$ by $K := [k_p, k_i, k_d]$ with its dimension being $n \times 3$. A set of control parameters $k_p$, $k_i$, and $k_d$ can yield a good step response and result in minimization of performance criteria in the time domain. These performance criteria in the time domain include the settling time $t_s$, rising time $t_r$, overshoot $M_p$, and steady-state error $e_{ss}$. In the meantime, we define the evaluation function $f$, which is a reciprocal of the performance criterion, as [14]

$$f = \frac{1}{(1 - e^{-\beta}) \cdot (M_p + e_{ss}) + e^{-\beta} \cdot (t_s - t_r)},$$

where $\beta$ is the weighting factor.

The searching procedures are shown as follows [13–15]:

**Step 1:** Specify the lower and upper bounds of the three controller parameters and randomly initialize the individuals of the population including: searching points, velocities, $pbest$, and $gbests$.

**Step 2:** For each initial individual $K := [k_p, k_i, k_d]$ of the population, employ Routh–Hurwitz criterion to test the stability of the closed-loop system and calculate the values of the four performance criteria in the time domain, namely $M_p$, $e_{ss}$, $t_s$, and $t_r$.

**Step 3:** Calculate the evaluation value of each individual in the population using the evaluation function $f$ given by Eq. (5).

**Step 4:** Compare each individual evaluation value with its $pbest$. The best evaluation value among the $pbest$ is denoted as being the $gbest$.

**Step 5:** Modify the member velocity $v_{j}$ of each individual $K$ according to

$$v_{j}^{(t+1)} = w \cdot v_{j}^{(t)} + c_1 \cdot \text{rand}(\cdot)(pbest_{j} - x_{j}^{(t)}) + c_2 \cdot \text{Rand}(\cdot)(gbest_{g} - x_{j}^{(t)}),$$

$$j = 1, 2, \ldots, n; \quad g = 1, 2, 3.$$

When $g$ is 1, $v_{j1}$ represents the change in velocity of the $k_p$ parameter.

When $g$ is 2, $v_{j2}$ represents the change in velocity of the $k_i$ parameter.

When $g$ is 3, $v_{j3}$ represents the change in velocity of the $k_d$ parameter.

**Step 6:** If $v_{j}^{(t+1)} > V_{\text{max}}$, then $v_{j}^{(t+1)} = V_{\text{max}}$.

If $v_{j}^{(t+1)} < V_{\text{min}}$, then $v_{j}^{(t+1)} = V_{\text{min}}$.

**Step 7:** Modify the member position of each individual $K$ according to

$$k_{j}^{(t+1)} = k_{j}^{(t)} + v_{j}^{(t+1)},$$

$$k_{j}^{\min} \leq k_{j}^{(t+1)} \leq k_{j}^{\max},$$

where $k_{j}^{\min}$ and $k_{j}^{\max}$ represent the lower and upper bounds of member $g$ of the individual $K$, respectively. For example, when $g$ is 1, the lower and upper bounds of the $k_p$ controller parameter are $k_{p}^{\min}$ and $k_{p}^{\max}$, respectively.

**Step 8:** If the number of iterations reaches the maximum, then go to Step 9. Otherwise, go back to Step 2.
Step 9: The individual that generates the latest \( g_{best} \) is an optimal controller parameter.

4. Fuzzy self-tuning PID controller

The structure of the proposed fuzzy PID controller is shown in Fig. 3. The detailed design procedures are described in the following sections.

4.1. Self-tuning PID

In most research, the parameters of self-tuning are based on known mathematical models. Even if the models are unknown, the on-line identification or parameter estimation is proposed to establish an estimated model. However, a modeling error usually causes unexpected conditions. Fortunately, for any complex systems, the PSO approach can easily obtain the most proper parameters under no-load and full-load conditions. Take the proportional controller for example. Under operating conditions between no-load and full-load, the proportional gain should be in the range of both full-load and no-load conditions. Let the proportional gains be \( P_{\text{min}} \) and \( P_{\text{max}} \) under no-load and full-load conditions, respectively. The proportional gain should be in the range of \( P_{\text{min}} \) to \( P_{\text{max}} \). According to common sense [23], if the proportional (\( P \)) gain increases then the rising time and steady state error will be reduced. A gain that is too large will cause a great overshoot and extreme oscillations. A gain that is too small will cause a steady-state error to exist. If the absolute values of error and error derivative are large, the proportional gain should be determined using the largest number to achieve fast rising time, that is \( P_{\text{max}} \). If the error and error derivative are small enough, the proportional gain should be \( P_{\text{min}} \) to maintain the minimum steady-state error. The relationship can be shown in Fig. 4. Hence, let the tuning rule be defined as

\[
K_P(t) = K^0_P - \Delta K_P(1 - \gamma(E(t), \dot{E}(t))),
\]

where \( K^0_P = P_{\text{min}}, \Delta K_P = P_{\text{max}} - P_{\text{min}} \) and \( \gamma \) is the adjusting factor. The adjusting factor \( \gamma \) will be decided on-line by fuzzy rules.

Via the same concept, the integral (I) gain should be increased along with the decreases of error and error derivative. The curve is shown in Fig. 5. Let the tuning rule be defined as

\[
K_I(t) = K^0_I + \Delta K_I(1 - \gamma(E(t), \dot{E}(t))),
\]

where \( K^0_I = I_{\text{min}}, \Delta K_I = I_{\text{max}} - I_{\text{min}}, I_{\text{max}} \) and \( I_{\text{min}} \) are the gains under no-load and full-load conditions, respectively.

Large differential controllers can increase the speed of response, and cause large steady-state errors, simultaneously. Therefore, the differential (\( D \)) gain should be decreased along with the error and error derivative decreases.

\[
K_D(t) = K^0_D - \Delta K_D(1 - \gamma(E(t), \dot{E}(t))),
\]
ing. The curve is shown in Fig. 6. The tuning rule can be shown as follows:

\[ K_D(t) = K_D^0 - \Delta K_D(1 - \gamma(E(t), \dot{E}(t))). \]  

(9)

Let \( K_D^0 = D_{\text{min}}, \Delta K_D = D_{\text{max}} - D_{\text{min}} \). \( D_{\text{max}} \) and \( D_{\text{min}} \) are the gains under full-load and no-load conditions, respectively.

The most significant of the proposed PID tuning method is that the internal parameters and tuning ranges of PID controllers are supported by the PSO approach. It will increase confidence in the controller for operators. After this, the tuning work will be completed by the fuzzy rule. In order to limit the \( k_p, k_i, \) and \( k_d \) within a reasonable range, Routh–Hurwitz criterion must be employed to test the stability of the closed-loop system. If the \( k_p, k_i, \) and \( k_d \) satisfy Routh–Hurwitz stability test, which is applied to the characteristic equation of the system, then the self-tuning PID controller is stable. The stability of the proposed self-tuning PID controller is proved in Appendix A. It is found that when the parameters \( k_p > 0, k_i > 0 \) and \( k_d > 0 \), the system is stable.

4.2. Adjusting factor \( \gamma \)

In order to intelligently and automatically tune the controller’s parameters, the adjusting factor \( \gamma \) is obtained by the fuzzy theory, in which the first work is to fuzzify the reference variables. The reference variables in the designing procedure of this paper are position error and its difference. There are seven linguistic variables used in this paper for each reference variables, they are: large negative (LN); medium negative (MN); small negative (SN); zero (ZO); small positive (SP); medium positive (MP); and large positive (LP). The membership functions representing these linguistic expressions are defined as triangular type functions, shown in Fig. 7. In order to process the reference variables more efficiently, the position error is to deal with a linear convert from \(-0.05 \text{ m} \) to \(+0.05 \text{ m} \) into \(0–1\). At the same time, the position error difference is converted from \(-100 \text{ m/s} \) to \(+100 \text{ m/s} \) to \(0–1\). The output factor is restricted in \(0–1\), shown in Fig. 7, too.

Secondly, the fuzzy relation and fuzzy rule are defined as 49 if-then rules for fuzzy inference. It is a common sense that if the absolute values of error and error difference are large, the adjusting factor needs to be large, and vice versa. The if-then rules can be established as follows:

\[ \text{Rule1: If } e(t) \text{ is LN and } \dot{e}(t) \text{ is LN, then } \gamma(t) \text{ is LP}, \]

\[ \text{Rule2: If } e(t) \text{ is MN and } \dot{e}(t) \text{ is MN, then } \gamma(t) \text{ is MP}, \]

\[ \text{Rule3: If } e(t) \text{ is SN and } \dot{e}(t) \text{ is SN, then } \gamma(t) \text{ is SP}, \]

\[ \vdots \]

(10)

The if-then rules can be tabulated as shown in Table 1. Finally, by the centroid defuzzification method, the adjusting factor can be calculated by

\[ \gamma(t) = \frac{\int_0^t x \max \{\mu_{e(t)}^L, \mu_{e(t)}^R\} dt}{\int_0^t \max \{\mu_{e(t)}^L, \mu_{e(t)}^R\} dt}. \]

(11)

where \( x \) indicates each central value of the linguistic variables of the adjusting factor. Finally, by applying the adjusting factor to Eqs. (7)–(9), it will form the intelligent self-tuning PID controllers.

Table 1

<table>
<thead>
<tr>
<th>( \mu(\gamma(t)) )</th>
<th>( e(t) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>LN</td>
<td>LP</td>
</tr>
<tr>
<td>MN</td>
<td>LP</td>
</tr>
<tr>
<td>SN</td>
<td>LP</td>
</tr>
<tr>
<td>ZO</td>
<td>LP</td>
</tr>
<tr>
<td>SP</td>
<td>LP</td>
</tr>
<tr>
<td>MP</td>
<td>LP</td>
</tr>
<tr>
<td>LP</td>
<td>LP</td>
</tr>
</tbody>
</table>

---

Fig. 6. Tuning rule of differential gain.

Fig. 7. Membership functions of error, error difference and adjusting factor. (a) Functions of errors, (b) functions of errors differences and (c) functions of adjusting factors.
5. Numerical results

In this section, numerical results are used to demonstrate the potential of the proposed control rule. In this paper, under the same conditions, we perform numerical simulations by using the PSO and RGA approaches and compare their convergence characteristics [5,11,24,25]. Fig. 8 shows their convergence properties. It is seen that the PSO approach has better evaluation value than that of the RGA approach. The results show that the PSO self-tuning PID controller could obtain a higher quality solution.

To demonstrate the performance, the self-tuning PID controller is compared with a fixed PID controller. The actual dimensions of the slider–crank mechanism as in Fig. 9 are \( m_1 = 0.05454 \text{ kg}, \ m_2 = 0.2795 \text{ kg}, \ m_3 = 0.16 \text{ kg}, \ r = 0.056 \text{ m}, \ R = 0.086 \text{ m}, \ l = 0.174 \text{ m}, \ l' = 0.055 \text{ m}, \ K_t = 0.6732 \text{ Nm/A}, \ J_m = 0.00062 \text{ Nms}^2 \text{ and } B_m = 0.000153 \text{ Nms/rad}. \) The objective is to regulate the desired translation position of slider from 0.2107 m to 0.1858 m. The desired specifications are: settling time \( t_s = 0.5 \text{ s}, \) rising time \( t_r = 0.25 \text{ s}, \) maximum overshoot \( M_p < 5\% \), and steady-state error \( e_{ss} < 1\% \).

Fig. 10(a) shows the responses of the fixed PID controller. Using the PSO approach, we can get the parameters for optimal performance under no-load conditions, where \( K_{P}^{\text{no}} = 3.5695, \ K_{I}^{\text{no}} = 2.9267, \text{ and } K_{D}^{\text{no}} = 0.7727 \). The parameter variation is depicted by the slider mass \( m_3 \), changed from 0.16 kg to 0.8 kg. The external force is changed from 0 to 5 Nt. While the load and parameter variations do not exist, the performance can satisfy the requirement. However, compared with these curves the responses do not match the desired specification when varying loads or parameter variations exist. Obviously, the fixed PID controller obtained under no-load conditions cannot achieve the robustness with varying parameters and existing loads.

By the same way, the PID’s parameters for full-load conditions: the optimal parameters for PID controller with full-load are \( K_{P}^{\text{full}} = 4.1942, \ K_{I}^{\text{full}} = 1.2997, \) and \( K_{D}^{\text{full}} = 1.1313 \). Theoretically, the dynamic responses under full-load will match the requirements. It is also proven in Fig. 10(b). However, once the parameter varies and the external loads are removed, the responses will cause a large steady-state error.
Fig. 10. Numerical results with response of translation position. (a) Response of translation position with fixed PID control (parameters obtained under no-load condition), (b) response of translation position with fixed PID control (parameters obtained under full-load condition) and (c) response of translation position with self-tuning PID control.

and unexpected transient state as shown in Fig. 10(b). The large steady-state error and overshoot appeared to show the bad robustness of a fixed PID controller.

The proposed self-tuning PID controller is based on these two optimal parameters. Let the normal parameters be based on no-load conditions, that is \( K^0_P = K^0_P = \frac{3.5695}{0.16} \), \( K^0_I = K^0_I = 2.9267 \), and \( K^0_D = K^0_D = 0.7727 \). The tuning ranges are selected as \( \Delta K_P = K^\text{full}_P - K^\text{no}_P = 0.6247 \), \( \Delta K_I = K^\text{full}_I - K^\text{no}_I = 1.6270 \) and \( \Delta K_D = K^\text{full}_D - K^\text{no}_D = 0.3586 \). According to the position error and its difference, the fuzzy rule will automatically find the optimal adjusting factor. Applying Eqs. (7)–(9), the responses of the proposed control rule are shown in Fig. 10(c). Obviously, the proposed self-tuning PID controller has great robustness. Under different operating conditions, it still maintains the desired performances.

6. Experimental results

In order to demonstrate the proposed control rules, a PC-based experimental equipment is setup in this paper. The experimental instrument of the slider–crank mechanism is divided into three parts: the actuator; the slider crank; and the controller. The photographic representation is shown in Fig. 9. The first part consists of a PM
synchronous motor and a driver. The driver is worked on 3-phase, 220 V, and 60 Hz. The slider–crank mechanism is coupled with the PM motor. The translation position is measured by a photometer. The output of the photometer scalar is 20,000 pulses/m, and when mapped to real translation position, is 0–0.11 m. In order to carry out the parameter varying and external load an external mass (0.8 kg) is added onto the slider. The controller is based on a PC with Pentium-586 CPU. The PC plays the role of software development and data process. The data acquisition interface card (Advantech CO., PCL-1800) is installed in the ISA bus so as to handle the A/D and D/A processes. The graphical software of Simulink is used to implement the proposed control rule. At the same time, the linear converts between the physical scale and voltage from sensors are also determined by this software.

Based on the same requirements of simulations, the experimental results were acquired from the HP 54601B 100 MHz 4 channel oscilloscope. Firstly, the fixed PID setup under no-load is experimented. Under no-load conditions, the experimental results are shown in Fig. 11(a). However, the responses shown in Fig. 11(b) have a large

![Fig. 12. Response of translation position with fixed PID control (parameters obtained under full-load condition). (a) Experimental result with 0.8 kg external load and (b) experimental result with no-load.](image)

![Fig. 13. Response of translation position with RGA PID control. (a) Experimental result with no-load, (b) experimental result with 0.5 kg external load and (c) experimental result with 0.8 kg external load.](image)
overshoot when the load is added. It shows the bad robustness of the fixed PID controllers.

The second experiment is under the fixed PID controllers whose parameters are obtained under full-load conditions. Fig. 12 (a) shows the experimental results. It shows that manually-chosen parameters can achieve desired requirements under full-load conditions. However, Fig. 12 (b) shows the same controller applied when the load is removed. Obviously, the fixed PID controller cannot overcome the large change in operating situations.

In the third experiments, we use the RGA method [11,24,25] to find the PID parameters to regulate the motor-mechanism coupling system. From the translation responses with no load shown in Fig. 13 (a), it PID parameters are \( k_p = 3.6021, \ k_i = 2.8483 \) and \( k_d = 0.78578 \), the changing loadings shown in Fig. 13 (b), it PID parameters are \( k_p = 3.8786, \ k_i = 1.9165 \) and \( k_d = 0.92406 \), and the translation responses with full load shown in Fig. 13 (c), it PID parameters are \( k_p = 4.1684, \ k_i = 1.3181 \) and \( k_d = 1.1292 \). One can achieve the same performance by use of the PSO self-tuning PID controller, but the time needed for off-line computation is longer than that by the PSO self-tuning PID on-line.

Finally, by applying the proposed self-tuning PID controller to the same experimental instrument, the responses of translation position with no-load conditions and are shown in Fig. 14 (a). It can achieve the same performance with the fixed PID controller under no-load conditions. The responses with full-load are shown in Fig. 14 (b). The dynamic responses are almost the same with those of no-load conditions. It also proves that the proposed self-tuning PID controller has great robustness in the presence of parameter variations and external loads.

7. Conclusion

This paper proposed a simple scheme of the self-tuning PSO PID method, which has the merits to avoid the shortcoming of premature convergence of the RGA method, obtain higher quality solution with better computation efficiency, and accurately determine the nominal values and tuning ranges of the PID parameters. From comparisons between numerical simulations and experimental results, the proposed method is found to have more robust stability and efficiency, and more easily and quickly solve the searching and tuning problems than the other two methods. Applying the intelligent fuzzy rule in proposed method, the adjusting factor can be tuned on-line and the time needed for off-line computation is shorter than that by the RGA PID. Moreover, numerical simulations and experimental results by a PC-based controller being implemented to the translation position control of a slider–crank mechanism show that the proposed controller is more robust than a fixed PID controller.

Acknowledgements

The authors are greatly indebted to the National Science Council R.O.C. for the support of the research through contract No. NSC93-2622-E-327-004-CC3.

Appendix A. Stability of the self-tuning PID controller

Using the PSO approach, we can obtain the parameters \( k_p, \ k_i, \) and \( k_d \) for optimal performance under the no-load and full-load conditions. In the no-load condition, we can determine the transfer function of the plant by MATLAB program as

\[
G(s) = \frac{21.7351}{s^2 + 0.0038s + 25.2555}. \tag{A.1}
\]

Also, the closed-loop system with a PID controller has the following characteristic equation

\[
s^3 + (0.0038 + 21.7351k_d)s^2 + (25.2555 + 21.7351k_p)s + 21.7351k_i = 0. \tag{A.2}
\]
The Routh array becomes as follows:

\[
\begin{array}{c|cccc}
& s^3 & s^2 & s^1 & s^0 \\
\hline
1 & 25.2555 + 21.7351k_p & 21.7351k_i \\
0.0038 + 21.7351k_d & 21.7351k_i \\
0.0038 + 21.7351k_d & 0 \\
0.0038 + 21.7351k_d & 21.7351k_i \\
\end{array}
\]

For the stability of the system, the coefficients in the first column must be positive for all values of \(k_p, k_i,\) and \(k_d,\) that is, it is found that as the parameters \(k_p > 0, k_i > 0\) and \(k_d > 0,\) the system is stable.

In the full-load condition, the transfer function of the plant is

\[
G(s) = \frac{19.4375}{s^2 + 0.0033s + 24.9531}
\]

and the characteristic equation of the closed-loop system with a PID controller is

\[
s^3 + (0.0033 + 19.4375k_d)s^2 + (24.9531 + 19.4375k_p)s + 19.4375k_i = 0.\]

Following the same principle, it is also found that as the parameters \(k_p > 0, k_i > 0\) and \(k_d > 0,\) the system is stable.

References