Robust PID Controller Design Using Particle Swarm Optimization

Liu-Hsu Lin, Fu-Cheng Wang, Member, IEEE, and Jia-Yush Yen Member, IEEE

Abstract—This paper proposes a novel method to synthesize robust proportional-integral-derivative (PID) controllers using particle swarm optimization (PSO). Robust control is well known for its ability in dealing with system uncertainties and disturbances. Standard robust control design, however, can result in controllers that are high-order and complicated and can be difficult to implement in practical applications. PID controllers are advantageous because of their simple structures and wide acceptance in engineering practice, but they lack profound theorems in dealing with system uncertainties and disturbances. Therefore, combining the advantages of these two control algorithms, robust PID-structure controllers are proposed to optimize system performance using PSO. Finally, an example is used to illustrate the design procedures. Simulation results show the proposed method to be effective.

I. INTRODUCTION

High-precision control systems, such as hard-disk drives and atomic force microscopes (AFMs), have stringent performance requirements despite system uncertainties and disturbances. Robust control has been frequently applied to these systems to satisfy performance criteria, because of its ability to cope with those system uncertainties and disturbances. Liu et al. [1] designed an $H_{\infty}$-based precision motion control system for a high-speed X-Y table for semiconductor wire-bonding so that consistent tracking performance could be achieved even in the presence of considerable resonance uncertainties and external disturbances. Itoh et al. [2] proposed a robust positioning method for a ball-screw driven table. Experimental results showed that the proposed controller provided the desired fast and precise positioning performance. Sebastian et al. [3] addressed the image problem of AFMs in the framework of modern robust control. Wang et al. [4, 5] applied robust control to a proton exchange membrane fuel cell (PEMFC) system to maintain steady voltage output and reduce hydrogen consumption by regulating the oxygen and hydrogen flow rate.

Experimental results demonstrated that robust controllers can achieve stability and reduce hydrogen consumption even with system perturbations.

Standard robust control algorithms result in controllers that can be very complicated and are not easily to implement in practical engineering applications. On the other hand, PID control has long been applied to industrial practice due to its simple structure and good performance. However, there is no profound theoretical background that can be used to tune the PID parameters for good performance when the system is complex and has uncertainties and external disturbances. Therefore, an effective design method which can combine the advantages of $H_{\infty}$ control and PID control design is needed.

There were several approaches proposed to synthesize PID controller for robust performance. Astrom et al. [6] presented a PI control method to reject load disturbance based on a non-convex optimization with constraints on sensitivity. In [7], LMI-based iterative optimization method was used to design robust controllers. The synthesis of $H_{\infty}$ PID controllers was presented in [8, 9], in which the robust performance problem was converted into simultaneous stabilization of a complex polynomial family based on the generalized Hermite-Biehler theorem. Then a linear programming design procedure was used to determining all admissible PID controllers. Besides, some researches were devoted to the robust PID controller design using the soft computing techniques such as the evolutionary algorithm and the genetic algorithm (GA) [10, 11].

In this paper, particle swarm optimization (PSO) is applied to design PID-structured robust controllers in which the controller parameters are tuned to achieve certain robust performance criteria. Furthermore, a measure of robust performance margin is proposed to evaluate system performance. Finally, an example is given to illustrate the design procedure. The performance of the proposed controller and a standard $H_{\infty}$ robust controller are compared in order to demonstrate the advantages of the proposed design. This paper is arranged as follows: in Section II the control problem is formulated. In Section III the design procedure of robust PID controllers using PSO algorithms is proposed. In Section IV an example is used to demonstrate the design and to verify robust performance of the system. Finally, some conclusions are drawn in Section V.
II. PROBLEM FORMULATION

In this section, the concepts of model uncertainty, robust stability, robust performance, and robust performance margin will be briefly introduced. Based on these concepts, the robust PID control design using PSO will be proposed in Section III.

Consider the single-input single-output (SISO) feedback control system of Figure 1, in which \( G(s) \) is the nominal plant, \( \Delta \) is the plant uncertainty with \( \| \Delta(s) \|_\infty < 1 \), \( d \) is the external disturbance, \( W_z(s) \) is the weighting function, such that the perturbed plant \( \hat{G}(s) \) with multiplicative uncertainty can be represented as follows:

\[
\hat{G}(s) = (1 + \Delta(s)W_z(s)) \cdot G(s).
\] (1)

![Figure 1. A feedback control system with multiplicative uncertainty.](image)

**Theorem 1** (Small Gain Theorem [12]).
Suppose \( M \in RH_\infty \) and let \( \gamma > 0 \). Then the interconnected system shown in Figure 2 is well-posed and internally stable for all \( \Delta \in RH_\infty \) with

(a) \( \| \Delta(s) \|_\infty \leq 1 / \gamma \) if and only if \( \| M \|_\infty < \gamma \);

(b) \( \| \Delta(s) \|_\infty < 1 / \gamma \) if and only if \( \| M \|_\infty \leq \gamma \),

where \( \| M \|_\infty \) is the \( \infty \) norm of system \( M \).

![Figure 2. Illustration of the Small Gain Theorem.](image)

When the Small Gain Theorem is applied, the closed-loop system in Figure 1 is internally stable for all \( \Delta(s) \) with \( \| \Delta(s) \|_\infty < 1 \) if and only if

\[
\| W_zT \|_\infty \leq 1,
\] (2)

where

\[
T(s) = G(s)C(s)/(1 + G(s)C(s))
\]

is the complementary sensitivity function. If the feedback system is internally stable, the nominal performance condition can be expressed as the following:

\[
\| W_zS \|_\infty \leq 1,
\] (3)

where \( W_z(s) \) is a weighting function and \( S(s) = 1/(1 + G(s)C(s)) \) is the sensitivity function which represents the transfer function \( T_{1\rightarrow r} \) or \( T_{d\rightarrow y} \).

The robust performance problem is that robust stability and disturbance rejection performance are achieved simultaneously. Therefore, the robust performance condition should be [13]

\[
\| W_zT \|_\infty \leq 1 \text{ and } \| W_zS \|_\infty \leq 1.
\] (4)

for all \( \Delta \) with \( \| \Delta(s) \|_\infty < 1 \). It is noted that (4) can be further simplified as in the following [13]:

\[
\| W_zS \| + \| W_zT \|_\infty \leq 1.\] (5)

The derivation of (5) was based on the robust stability criterion of (2) with uncertainty bound \( \| \Delta(s) \|_\infty < 1 \) and the nominal performance criterion of (3). Hence, the nominal performance level \( \alpha \) can be defined as:

\[
\alpha = \sup \{ \| \alpha W_zS \|_\infty < 1 \} = \frac{1}{\| W_zS \|}.
\] (6)

Similarly, the robust stability margin \( \beta \) can be defined as:

\[
\beta = \sup \{ \| \beta W_zT \|_\infty < 1 \} = \frac{1}{\| W_zT \|}.
\] (7)

Combined (6) and (7), the robust performance margin \( \phi \) can be defined as follows:

\[
\phi = \sup \{ \| \phi (W_zS + W_zT) \|_\infty < 1 \}
\]

\[
= \frac{1}{\| W_zS \| + \| W_zT \|_\infty}.
\] (8)

That is, \( \phi \) is a measure of the ability for the closed-loop system to achieve robust performance.

III. CONTROLLER DESIGN USING PARTICLE SWARM OPTIMIZATION

PSO is a population-based stochastic optimization technique developed in 1995 [14], from the simulation of social behavior of bird flocking or fish schooling. PSO has been found to be simple, effective and robust in solving problems with nonlinearity, non-differentiability, and multidimensional optimization [14-18]. Compared to other optimization algorithms, the advantages of PSO include its relative simplicity and stable convergence characteristic with good computational efficiency.

In PSO, each particle represents a candidate solution to the optimization problem. At the beginning, each particle spans randomly through the problem space and updates its velocity and position with two best values. The first best value,
called $P_{best}$, is the best solution achieved so far. Another best value, called $G_{best}$, is the global best solution obtained so far by any particle in the swarm. At each iteration, each particle moves to $P_{best}$ and $G_{best}$ locations. The cost function evaluates the performance of particles to determine whether the best solution is achieved. The concepts of PSO are illustrated in Figure 3, where $X_i$ is the current position, $X_i^{k+1}$ the new position, $V_i$ the current velocity and $V_i^{k+1}$ the new velocity. With new iteration, the velocity of each particle is calculated based on its current velocity, the distance from its previous best position and the distance from the global best position. Then the new velocity is used to calculate the next position of the particle in the search space as follows:

$$V_i^{k+1} = W \ast V_i^k + C_1 \ast \text{rand}_i \ast (P_{best} - X_i) + C_2 \ast \text{rand}_i \ast (G_{best} - X_i)$$

$$X_i^{k+1} = X_i^k + V_i^{k+1},$$

where $W$ is the inertia weight function. $C_1$ and $C_2$ are learning factors and represent the cognition and social components, respectively, which attract the particles to the local best and global best positions. $\text{rand}_i$ and $\text{rand}_i$ are random numbers in the range of $[0,1]$ and $P_{best_i}$ is $P_{best}$ of particle $i$.

![Figure 3. The movement of particles.](image)

Furthermore, the positions of particles are bounded by $X_{max}$ and $X_{min}$ and the velocities on each dimension are also bounded by the maximum velocity $V_{max}$. It is noted that $V_{max}$ can influence the convergence of the problem and needs to be carefully selected. For example, if $V_{max}$ is set too high, the particle may not sufficiently explore the search space [19], but if $V_{max}$ is set too low, the calculation may take too much time. The selection of inertia weight $W$ in (9) can be regarded as the compromise between global and local explorations. The following equation can be used to dynamically adjust $W$:

$$W = W_{max} - \frac{W_{max} - W_{min}}{N_{max}} \times N_{iter},$$

such that less iterations are required to find a sufficient optimal solution. In (11), $N_{max}$ is the maximum number of iteration and $N_{iter}$ is the current number of iteration. Note that the inertia weight $W$ decreases linearly from $W_{max}$ to $W_{min}$ during the iterations. At the beginning of the iterations, a larger $W$ setting can speed up the search and prevent the solution from falling into a local minimum.

Now suppose the controller $C(s)$ takes the PID structure as follows:

$$C(s) = K_p + \frac{K_i}{s} + K_d s.$$  

Besides, the following cost function is defined to compute the cost of each particle in swarm:

$$J(K_p, K_i, K_d) = \|W_1 S\| \|W_2 T\|_{\infty},$$

Furthermore, in order to ensure that the system is stable and the parameters $K_p$, $K_i$, and $K_d$ are within a reasonable range, the Routh-Hurwitz criterion is employed to test the stability of the closed-loop system. If the Routh-Hurwitz criterion fails with the resulting parameters, the particle can not be used as a candidate solution.

The design procedures of the PSO-based robust PID controller can be summarized as follows:

Step 1. Randomly initialize the positions and velocities of particles using uniform probability distribution.

Step 2. Return to Step 1 until the Routh-Hurwitz stability test is satisfied.

Step 3. Calculate the cost of each particle using the cost function of (13).

Step 4. Compare each particle’s current cost with its $P_{best}$. If it is better, set the current value as the new $P_{best}$.

Step 5. Compare the best cost among the entire particle with $G_{best}$. If the current value is better, set the current best value as $G_{best}$.

Step 6. Calculate the particle’s velocity by (9).

Step 7. Update the particle’s position by (10).

Step 8. Return to Step 2 until the maximum iterations or minimum error bound is attained.
IV. AN DESIGN EXAMPLE

In this section, an example is used to illustrate the robust PID control design using PSO. Refer to the control system shown in Figure 1 with the following nominal plant:

\[ G(s) = \frac{1}{(s + 0.01)^2} \]  

(14)

and weighting function:

\[ W_2(s) = \frac{0.1s}{0.05s + 1} \]  

(15)

Furthermore, the weighting function \( W_1(s) \) is selected to guarantee good tracking over the frequency range [0, 1] with the tracking error less than 10%:

\[ W_1(s) = \frac{10}{s^3 + 2s^2 + 2s + 1} \]  

(16)

Setting the PID controller as in (13) with the following bounds:

\[ K_p \in [0, 20] \], \( K_i \in [0, 20] \], and \( K_d \in [0, 20] \]. From our experiences, the parameters of the PSO are set as below:

- The number of particles = 50.
- Dimension of particles = 3.
- The maximum change in velocity:
  \[ V_{Kp}^{\text{max}} = K_p^{\text{max}} / 2 \],
  \[ V_{Ki}^{\text{max}} = K_i^{\text{max}} / 2 \], and \[ V_{Kd}^{\text{max}} = K_d^{\text{max}} / 2 \].
- Learning factor: \( C_1 = 2 \), \( C_2 = 2 \).
- Inertia weight: \( W_{\text{max}} = 0.4 \sim 0.9 \); \( W_{\text{min}} = 0.1 \sim 0.4 \).

Using the aforementioned PSO techniques, the convergence of the cost function for the PSO search is shown in Figure 4 where the PSO method can prompt convergence and obtain good cost value after about twenty iterations. The best PID parameters are calculated as: \( K_p = 15.828 \), \( K_i = 1.439 \), and \( K_d = 6.546 \).

Furthermore, the nominal performance, robust stability, and robust performance measures of (2-3) and (5), respectively, are verified in Figure 5. It is clear that the designed robust PID controller does achieve these specifications.

Figure 4. The convergence of the cost function \( J \).

Figure 5. The nominal performance, robust stability, and robust performance of the system.

Figure 6 shows the step responses of the closed-loop system with the nominal plant \( G(s) \) and a perturbed plant \( \tilde{G}(s) \) with uncertainty \( \Delta(s) = 1 \), respectively. To illustrate the designed robust PID controller’s ability to cope with system disturbances, Figure 7 shows the step response of the system with a disturbance \( d(t) = 0.2 \sin(t) + 0.5 \delta(t - 3) \). These results clearly indicate that the designed robust PID controller can achieve good robust performance and rapidly cope with disturbances.

Figure 6. Step responses of the nominal plant and the perturbed plant with uncertainty \( \Delta(s) = 1 \).
In order to demonstrate the advantages of the proposed control, a conventional optimal $H_\infty$ controller is derived and compared with the designed robust PID controller. Using the same plant and weighting functions of (14-16), a MATLAB command, called `hinfsyn` [20], is applied to obtain the following optimal $H_\infty$ controller:

$$K(s) = \frac{(1.504 \times 10^8 s^5 + 3.582 \times 10^9 s^4 + 1.229 \times 10^{10} s^3 + 16.47 s^2 + 3.257 \times 10^5 s + 1.622 \times 10^6) / (s^5 + 2206 s^4 + 2.137 \times 10^6 s^4 + 6.494 \times 10^8 s^3 + 1.294 \times 10^9 s^2 + 1.292 \times 10^9 s + 6.451 \times 10^8)}{\times + \times + \times + \times + \times}.$$

(17)

$K(s)$ is a six-order function with large coefficients and, therefore, is difficult to implement for practical control application. Furthermore, Figure 8 illustrates the step responses of the nominal plant and the perturbed plant using the robust PID controller $C(s)$ and the traditional robust controller $K(s)$. Performance measures, including settling-time $T_s$, rise-time $T_r$, peak-time $T_p$, maximum overshoot $M.O.$, steady-state error $e_{ss}$, and robust performance margin $\phi$ are summarized in Table 1. The PSO-based robust PID controller achieves better performance in terms of $T_s$, $T_r$, and $e_{ss}$ shows a higher robust performance margin $\phi$.

Finally, in order to compare the ability of disturbance rejection in these two controllers, the same disturbance $d(t) = 0.2 \sin(t) + 0.5\delta(t - 3)$ is applied to the system with the traditional robust controller $K(s)$. The results in Figure 9 show that the traditional robust controller $K(s)$ has slightly larger disturbance responses in comparison to the PSO-based robust PID controller (see Figure 6).
V. CONCLUSION

In this paper, a novel robust PID control design method using PSO has been presented. The proposed method offers a simple and effective way to combine the advantages of sophisticated $H_{\infty}$ control algorithms and practical PID control structures. An example was used to illustrate the design procedures. From the simulation results, the robust PID controller achieves better disturbance attenuation than the traditional robust controller, as well as showing a higher robust performance margin. Therefore the proposed control algorithms are shown to be effective. In the future, this control method can be further extended and applied to multivariable systems.

REFERENCES


