INTERNATIONAL JOURNAL OF INTELLIGENT CONTROL AND SYSTEMS
VOL. 10, NO. 4, DECEMBER 2005, 269-276

Motion Control Design Optimization: Problem and Solutions

Qing ZHENG and Zhiqiang GAO

Abstract- **Given a particular motion control problem, the question investigated in this paper is how to select the most appropriate control law and its parameters. Here, the selections of both the controller and its parameters are formulated as a single optimization problem. A novel cost function is proposed to capture many practical design considerations. The choices of controllers include the conventional proportional-integral-derivative (PID) control and its variations, the parameterized loop-shaping method, as well as the more recent linear active disturbance rejection method. For the purpose of both performance optimization and ensuring the fairness of the controller comparison and selection, the parameters of each controller are optimized in the comparison study using genetic algorithm (GA). The proposed optimization method is demonstrated in a motion control design case study, where the results convincingly show that the controller, based on the active disturbance rejection concept, is clearly superior to others.**

Index terms-- **Motion Control, PID, Leadlag Compensator, Feedforward, Loop shaping, Active Disturbance Rejection, Cost Function, Genetic Algorithm, Optimization.**

1. INTRODUCTION

Motion control applications can be found in almost every sector of industry, from factory automation and robotics to high-tech computer hard-disk drives. They are used to regulate mechanical motions in terms of position, velocity, and acceleration, and/or to coordinate the motions of multiple axes or machine parts. Productivity and efficiency drive the advance of motion control technology. Significant advances have been made in the last several decades in digital control hardware, such as programmable logic controllers (PLC), microcontrollers, and digital signal processors (DSP), and more recently, in field programmable gate arrays (FPGA). It is anticipated that these advances will continue into the foreseeable future at a rapid rate. Such hardware advances present enormous opportunities for control researchers in investigating and implementing new control strategies that would have been otherwise impossible. The issue is made even more pressing by the present state of technology. Similar to other industrial control applications, the proportional-integral-derivative (PID) control is still the method of choice in most motion applications [1-5]. PID is simple to use and easy to understand. Over the years, practicing engineers have developed various variations of PID and tuning methods for better performance and the ease of application. But this method is still largely trial and error and its performance is severely limited by its simple structure, particular in motion control.

Industrial automation provides the unending need of high performance motion control systems. A typical cell may employ over a hundred feedback control loops. To make them faster, more accurate, and more tolerant of process uncertainties and disturbances are great challenges control engineers face on a daily basis. In addition to PID, there are of course many other control design methods that can be used in the context of motion control, including the pole-placement and loop-shaping methods found in most textbooks, and the more recent active disturbance rejection control (ADRC) [6,7]. Carefully designed and implemented, each can provide a workable solution in many applications, but the question remains: how does one go about selecting the best controller for a particular problem? In addition, how does one systematically select the best parameter setting for a controller? More importantly, before the above questions can be answered, what constitutes a “good” motion control system? Does it include high performance trajectory tracking, good disturbance rejection, robustness, keeping control signal smooth, and minimizing the wear and tear of actuators? If so, can they be captured in a cost function, which allows us to bring the powerful set of tools in optimization to bear? These are the questions to be addressed in this paper.

Using a class of motion control problems as the starting point, this paper demonstrates how the controller and parameter selections can be generalized as an optimization problem and solved using genetic algorithm (GA). A cost function that comprehensively represents the design considerations is first proposed. GA, an optimization algorithm developed based on the nature of evolution, fits well within the scope of motion control design. This is because in the settings of industrial motion control, the dynamics of the plant is generally nonlinear, time-varying and, to a large extent, unknown. Typically, control systems in industry operate in an environment that is full of noise and disturbances. Any meaningful optimization of control systems must take these issues into consideration and GA seems to fit the bill quite well.

GA is modeled from a biological process, rather than a physical one, such as simulated annealing. It is an adaptive method widely used in solving search and optimization problems and is evidenced as one of the effective alternative in operation research [8]. Some researchers have applied GA-based optimization to motion control design with limited success [9-10]. In particular, a robust 2 degree-of-freedom compensator for the motion control design using GA was presented in [9], but it merely considered the integral of the absolute error in the output response as the fitness, without considering disturbance rejection, the control signal and its smoothness. An auto-tuning method for the design of multi-loop PI controllers based on GA was presented in [10], but again, only the tracking performance was considered. In addition, the choices in control law selection were not fully explored, as both paper evaluated only one control method. In

Manuscript received October 2, 2005; revised November 20, 2006.
Q. Zheng and Z. Gao are with the Center for Advanced Control Technologies, Cleveland State University, Cleveland, OH 44115. (e-mail: q.zheng1@csuohio.edu, and Corresponding author: z.gao@ieee.org).

this paper, the design considerations, including trajectory tracking, keeping control signal smooth, and limiting the peak value of control signal etc., are captured in a single cost function. Furthermore the cost function is minimized with respect to not only the parameter settings of a controller, as seen in previous research, but also the choices in control law itself.

This paper is organized as follows. Section II formulates the problem including the mathematical model, brief description of various control techniques, and proposed cost function. Section III addresses control law selection and parameter optimization. Simulation results with comparison are shown in Section IV. Finally, some concluding remarks are given in Section V.

2. PROBLEM FORMULATION

In this section, a class of motion control problems and the available control design methods, applicable in this study, are briefly described. A novel cost function is proposed and the optimization problem is formulated.

2.1 A Class of Motion Control Problems

In a typical application using motor as the power source, the equation of motion can be described as:

$$\ddot{y} = f(t, y, \dot{y}, w) + bu \tag{1}$$

where y is position, u is the motor current, b is the torque constant divided by inertia or mass, and w represents the external disturbance such as vibrations and torque disturbances. The friction, the effect of inertia change and various other nonlinearities in a motion system are all represented by the function $f(\cdot)$, which is mostly unknown and time-varying function. In most motion control literature, the linear time-invariant approximation is used:

$$\ddot{y} = -\frac{a}{J_t}\dot{y} + \frac{b}{J_t}u \tag{2}$$

where J_t is the total inertia of the motor and load, and b is the viscous friction coefficient. This linear approximation allows the use of the transfer function

$$G(s) = \frac{b}{s(J_t s + a)} \tag{3}$$

This problem as described in (1)-(3) is important because it represents almost all motion applications found in manufacturing industry. As will be seen in the later sections, some control design methods presuppose simple, known, plants in (3) while some can deal with the largely unknown plants in (1). Therefore, finding a suitable solution for each type of control problems is of great interest in practice.

Motion Profiling

An interesting characteristic of motion control is the use of motion profile. Using position control as an example, it is not good enough to just rotate an axis by the desired amount, say one revolution. How it gets there is also relevant. The velocity, acceleration and jerk (differentiation of the acceleration) all play important roles in a motion application and their desired trajectories are known as motion profiles. The motion profile is used as the command input in the closed-loop control, as opposed to the step command.

In this paper, a profile generator is used to produce a position profile, $v_1(t)$, and the velocity profile $v_2(t)$ from the desired set-point of one revolution. For simplicity, the industry standard trapezoid velocity profile, shown in Figure1, is used.

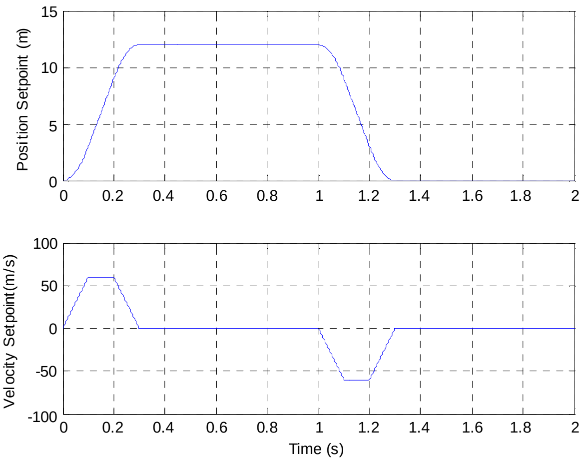


Fig. 1. Position and velocity profiles.

Advanced Motion Control Design Objectives

The necessary improvements of the current motion control techniques are listed below:

- High performance in terms of forcing the motion variables such as position and velocity to track the desired trajectory quickly and accurately;
- Smoother control signal and lower level of wear and tear of actuators;
- High degree of robustness: making the motion control system highly tolerant of dynamic variations such as inertia and friction changes;
- Better external disturbance rejection capability such as torque disturbance rejection;
- Simplification of controller design and tuning for the users.

2.2 Motion Control Design Options

Theoretically, there are many candidates for the motion control design problem described in (1) to (3). For the sake of practicality, three design methods are considered in this section: the empirical PID and its variations, the model-based loop-shaping method, and the largely model independent linear active disturbance rejection (LADRC) method. In order to evaluate these methods objectively based on practical design considerations, a novel cost function is proposed and shown in the next section.

PID and Its Variations

The PID controller is the most commonly used method in industry. It does not rely on the accurate mathematical model of (3). It is given as

$$u(t) = K_P e(t) + K_I \int e(t)dt + K_D \dot{e}(t) \tag{4}$$

Its transfer function is:

$$G_c(s) = \frac{(K_D s^2 + K_P s + K_I)}{s} \tag{5}$$

PID is widely used in practice because it can be easily understood and tuned. However, the very simplicity that makes it popular is also the cause that limits its performance and stability margins. The understanding of the nature of PID led to some variations, as shown below.

PID Control with Leadlag Compensation

This modification is intended to improve the transient response and the stability margin of the closed-loop system. The transfer function of the PID control with single stage leadlag compensation is:

$$G_c(s) = \left(\frac{K_{DII} s^2 + K_{PII} s + K_{III}}{s}\right)\left(\frac{s/z_0 + 1}{s/p_0 + 1}\right) \tag{6}$$

Compared to the original PID in (5), the additional term in (6) provides the means to change the frequency response of the open loop transfer function. By manipulating the zero and pole location of the leadlag compensator as well as the PID gains, the disturbance rejection, the overshoot, and the gain and phase margin can be changed to meet the design requirements. Note that this design method is still somewhat ad hoc and the amount of change in performance and stability margins is quite limited. To achieve further improvement, engineers also resort to the feedforward approach, as shown next.

PID with Velocity Feedforward Control

Velocity feedforward control is a common motion control strategy in industry. It is adopted to improve the system tracking performance and simplify design as well as tuning. Figure 2 shows the block diagram of the velocity feedforward control. Its transfer function is:

$$G_c(s) = G_{c2}(s)s + G_{c1}(s)G_{c2}(s) \tag{7}$$

where

$$G_{c1}(s) = \frac{(K_{Dpos} s^2 + K_{Ppos} s + K_{Ipos})}{s}$$

and

$$G_{c2}(s) = \frac{(K_{Dvel} s^2 + K_{Pvel} s + K_{Ivel})}{s}$$

respectively. This method takes advantage of the additional information available, which, in this case, is the differentiation of the setpoint, $\dot{y}_{sp}$. This cascade loop design enables the higher bandwidth inner velocity loop for better disturbance rejection and simplification of the design procedure by breaking it into smaller, easily manageable pieces. But the nature of the method is still trial and error.

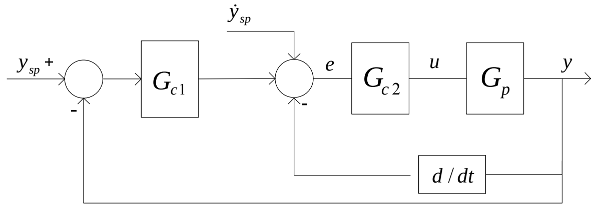


Fig. 2. PID with velocity feedforward control.

Parameterized Loop-Shaping Control

Assuming the linear approximation of the motion control plant in (3), a more systematic approach becomes feasible in dealing with various aspects of the design requirements, such as disturbance rejection, noise tolerance, and robust stability. Loop shaping is one such method, also known as frequency response based control design method. It is a method that comprehensively addresses multiple design concerns, such as transient response, disturbance rejection, stability margins, and noises. Loop shaping as a concept and design tool helps the practicing engineers greatly in improving the loop performance and stability margins. Unfortunately, manipulating loop-gain frequency responses and managing competing performance measures can be tedious and, in some cases, frustrating. A computer algorithm to automate this process was proposed in [11-14], which not only replaces the manual design but also leads to a self-tuning controller design and implementation. A parameterized loop-shaping design is proposed in [6] to further reduce the tuning complexity, which is briefly described below.

Consider the desired loop-gain frequency response in Figure 3, its transfer function can be characterized as:

$$L(s) = G_p(s)G_c(s) = \left(\frac{s+\omega_1}{s}\right)^m \frac{1}{\frac{s}{\omega_c}+1} \frac{1}{\left(\frac{s}{\omega_2}+1\right)^n} \tag{8}$$

where ω_c is the bandwidth, and

$$\omega_1 < \omega_c,\ \omega_2 > \omega_c,\ m \geq 0, \ and\ n \geq 0 \tag{9}$$

are selected to meet constraints shown in Figure 3. Both m and n are integers. Once the appropriate loop gain constraints are derived and the corresponding lowest order $L(s)$ in (8) is selected, the controller can be determined from

$$G_c(s) = \left(\frac{s+\omega_1}{s}\right)^m \frac{1}{\frac{s}{\omega_c}+1} \frac{1}{\left(\frac{s}{\omega_2}+1\right)^n} G_p^{-1}(s) \quad (10)$$

Note that the design is valid only if the plant is minimum-phase, as assumed. Furthermore, n should be selected such that $G_c(s)$ is proper.

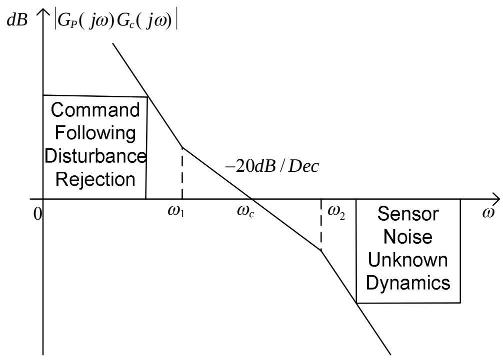


Fig. 3. Loop-shaping.

The downside of loop shaping is its reliance on the mathematical model of the plant. Can a high performance controller be obtained without such a model? An example is shown in the following.

Linear Active Disturbance Rejection Control

LADRC offers a new and inherently robust controller building block that requires very little information of the plant. Based on the extended state observer (ESO), this control algorithm actively estimates and compensates for, in real time, the *effects* of the unknown dynamics and disturbances, forcing an otherwise unknown plant to behave like a nominal one. Such strategy leads to the ditching of the prevailing model requirement in the current design. That is, instead of depending on the model of the plant, the controller draws the information needed to control the plant from the ESO. This is achieved by using an ESO to estimate y, $\dot{y}$ as well as $f(\cdot)$ and then reduce (1) to a double integral plant. More details of this novel control concept and associated algorithms can be found in [6-7, 15-17]. The idea of LADRC is briefly given below.

The parameterized linear extended state observer (LESO) for (1) is [6]

$$\begin{cases} \dot{z}(t) = Az(t) + Bu(t) + L(y(t) - \hat{y}(t)) \\ \hat{y}(t) = Cz(t) \end{cases} \quad (11)$$

with

$$A = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix},\ B = \begin{bmatrix} 0 \\ b_0 \\ 0 \end{bmatrix},\ C = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix},\ L = \begin{bmatrix} 3\omega_o \\ 3\omega_o^2 \\ \omega_o^3 \end{bmatrix}$$

where b_0 is an approximate value of b ; ω_o is the bandwidth of the observer; z_1, z_2, and z_3 are the estimates of y, $\dot{y}$, and f respectively. The control law is

$$u = \frac{\left[k_p(r - z_1) - k_d z_2\right] - z_3}{b_0} \quad (12)$$

where r is the setpoint. Here the gains can be selected as

$$k_d = 2\xi\omega_c,\ k_p = \omega_c^2 \quad (13)$$

where ω_c is the bandwidth of the controller and ξ is the damping ratio of the desired closed loop. Clearly the LADRC has two tuning parameters: ω_o and ω_c. The stability proof of LADRC is given in [6,17]. Practical applications have shown that this strategy is especially effective for nonlinear, time-varying, and uncertain plants, precisely as those found in industrial motion control.

2.3 The Cost Function and the Optimization Problem

For motion control, the main objectives include improving trajectory tracking performance, good disturbance rejection as well as robustness, and keeping control signal small and smooth to minimize the wear and tear of actuators. To this end, a new cost function, which comprehensively represents the above design considerations, is proposed. A "good" motion control solution can be found by minimizing the cost function. This brings us a useful tool to select the best controller and its parameter setting. This new cost function is defined as follows:

$$J_c = w_1 \frac{|e|_{\max}}{n_1} + w_2 \frac{ISE}{n_2} + w_3 \frac{|u|_{\max}}{n_3} + w_4 \frac{ISU}{n_4} + w_5 \frac{|\dot{u}|_{\max}}{n_5} + w_6 \frac{IS\dot{U}}{n_6} \quad (14)$$

with

$$ISE = \int_0^t e(\tau)^2\, d\tau,$$
$$ISU = \int_0^t u(\tau)^2\, d\tau,$$
$$IS\dot{U} = \int_0^t \dot{u}(\tau)^2\, d\tau.$$

where n_i, $i=1,2, \ldots, 6$ are normalizing factors, and w_i, $i=1,2, \ldots, 6$, are the weights. The higher the weight, the more important the corresponding cost function element becomes.

They are selected by users according to their demands on each element. Moreover, there are design constraints for a motion system as follows:

(1) The tracking error $e(t)$ has a given upper limit of k_1, i.e.,

$$|e(t)| < k_1 \tag{15}$$

(2) The motor current $u(t)$ has a given upper limit of k_2, i.e.,

$$|u(t)| < k_2 \tag{16}$$

(3) The rate of the change of the current $\dot{u}(t)$ is limited by the given supply voltage k_3, i.e.,

$$|\dot{u}(t)| < k_3 \tag{17}$$

(4) To reduce the wear and tear of the motor and prolong its life span, it is also desired that $u(t)$ is "smooth" with

$$\int_0^t \dot{u}(\tau)^2 d\tau < k_4 \tag{18}$$

where k_4 is a given constant.

The combined problem of selecting the best controller and finding appropriate controller parameters to minimize J_c with the constraints is an optimization problem. The parameter selection part can be expressed as $\min_{\mathbf{c}\in C} J_c$ subject to the design constraints (15)-(18). Here $\mathbf{c} = \{c_i\}$ are the settings of tuning parameters for each control algorithm, and $C = \{\mathbf{c}, c_{i\min} \le c_i \le c_{i\max}\}$ is the searching space of the tuning parameters for each control algorithm.

3. CONTROL LAW SELECTION AND PARAMETER OPTIMIZATION

The cost function (14) is highly nonlinear and discontinuous. Conventionally these control parameters were tuned manually or iteratively optimized [18]. Of course, manual tuning could be very time-consuming and sometimes frustrating. Iterative search is prone to entering local optimum, as shown in [18]. The parameter space C is essentially an uncountable set. It is impossible to exhaust the entire solution space to find the global optimal solution. Trade-off between the quality of solution and computational power must be made. GA renders us a decent method for solving this problem. In this section, we briefly describe how GA searches the optimal settings of tuning parameters for each controller and thus minimize the cost function.

GA is a direct search algorithm, which borrows the idea and the developing mechanism from genetic evolution and natural selection. It begins by randomly creating its population. Each individual of the population represents a search point in the space of potential solutions of the given optimization problem. Candidate solutions are combined by a crossover operator to produce offspring, which expands the current population of solutions. Thus the individuals in the population are evaluated via the fitness function. Meanwhile a mutation operator is performed at a certain probability level to increase variation in the search space. By favoring the mating of the more fit individuals, the more promising areas of the search space are explored. The process of evaluation, selection, crossover and mutation is repeated until a predetermined number of generations are reached or a satisfied solution has been found. The following sections describe each of the components of the GA method.

3.1 Initialization

For the controller parameters, we selected their lower and upper bound VLB and VUB as in Table.1 for different control algorithms. The initial populations of candidate solutions are generated randomly. Before the GA-based algorithm can be executed, a suitable encoding is necessary, which means that each possible solution in the search space can be encoded into a unique individual of binary numbers. In this study, the candidate solutions of the tuning parameters and the outputs of the tuning parameters are represented by binary values with different bits according to their ranges.

The population size and generation size are chosen differently for different control algorithms. They are shown in Table 1. If the corresponding output value of each parameter does not meet the requirement of constraints, the individual is discarded immediately after it is generated.

3.2 Evaluation

A fitness function must be devised for each problem to be solved as it provides the criterion for evaluating the achievement of the problem-specific objective. This procedure calculates the fitness values for all members of the population. Because our objective is to minimize the cost of J_c in (14), we define the fitness function F as follows:

$$F(\overrightarrow{X_i}) = -J_c \tag{19}$$

where $\overrightarrow{X_i}$ represents the ith individual in the population. The larger the fitness value is, the better the individual fits.

Therefore, the optimization task is to search the individual, which maximizes the fitness function $F(\overrightarrow{X_i})$.

3.3 Reproduction

The selection, crossover and mutation operation are the reproductive phase of GA. During this phase, individuals are selected according to their fitness value, and recombined to make up the new population of individuals. The selection uses the roulette wheel method. Those individuals, which have higher fitness value, have the higher opportunity to reproduce. Each new individual inherits some parts from either of the former individuals.

The purpose of selection is to provide more reproductive chances, on the whole, to those individuals better fit. The reproduction of an individual is based on its percentage probability that is calculated through dividing its fitness value by the total population fitness.

$$P(\overrightarrow{X_i}) = F(\overrightarrow{X_i}) / \sum_{j=1}^{N} F(\overrightarrow{X_j}) \quad (20)$$

where $P(\overrightarrow{X_i})$ and $F(\overrightarrow{X_i})$ represent the selection probability and fitness function value of the ith individual in the population, and N is the population size. From this probabilistic selection operator, we notice that the individuals with a higher fitness value have a higher probability of contributing one or more offspring in the following generation.

The purpose of crossover consists in the combination of useful individual segments from different individuals to form new and better performing offspring. In this paper, one point crossover technique is used. A crossover point $k \in \{1,2,...,n-1\}$ is generated randomly. Two parent individuals are cut from the crossover point to produce two "head" segments and two "tail" segments. The tail segments are swapped over to produce two new full individuals. This operator decides randomly whether each bit in the parent individuals is to be exchanged or not. Therefore it causes a strong mixing effect which is helpful to overcome the local optima.

Another method to avoid the local optimum is the mutation operation. Mutation randomly alters each gene in the individual with a small probability. A typical mutation is shown as below:

mutation point ↓
original offspring: 010011**1**110110
mutated offspring: 010011**0**110110.

By encoding, each point in the search space can be represented by a unique individual of binary numbers. Each gene in the individual shares the probability of being 0 or 1 by mutation. Therefore mutation helps insure that no point in the search space has a zero probability of being explored. A small mutation probability of 0.03 is used in this study. Note that if the mutation probability is too large, the process will lose its characteristics of inheritance and will become a pure random optimization process.

Table 1: Cost Function Value Comparison for Five Control Algorithms

Performance \ Algorithms	PID control	PID with leadlag compensation	PID with velocity feedforward control	Parameterized loop-shaping control	LADRC
J_{cmin}	5.3655	5.1287	4.8891	4.3592	2.7146
J_{cave}	5.3764	5.1737	4.9577	4.3648	2.7160
$Stdev$	0.011238	0.030697	0.091922	0.004909	0.002014
Best parameters	$K_p = 21.905$ $K_i = 59.048$ $K_d = 0.10429$	$K_{pll} = 27.778$ $K_{ill} = 61.429$ $K_{dll} = 0.38095$ $z_0 = 69.842$ $p_0 = 22.223$	$K_{pvel} = 188.89$ $K_{ivel} = 1030.2$ $K_{dvel} = 0.4119$ $K_{ppos} = 0.1064$ $K_{ipos} = 0.06984$ $K_{dpos} = 0$	$\omega_1 = 20$ $\omega_2 = 96.1905$ $\omega_c = 32.8571$	$b_0 = 180.1587$ $\omega_c = 110$ $\omega_o = 30$
Parameter searching space	$K_p \in [10,40]$ $K_i \in [40,120]$ $K_d \in [0.01,1]$	$K_{pll} \in [10,30]$ $K_{ill} \in [30,90]$ $K_{dll} \in [0,1]$ $z_0 \in [0.001,100]$ $p_0 \in [0.001,100]$	$K_{pvel} \in [100,300]$ $K_{ivel} \in [800,1300]$ $K_{dvel} \in [0.05,1]$ $K_{ppos} \in [0.1,0.5]$ $K_{ipos} \in [0,0.2]$ $K_{dpos} \in [0,0.2]$	$\omega_1 \in [10,40]$ $\omega_2 \in [90,220]$ $\omega_c \in [10,40]$ $m = 2$ $n = 3$	$b_0 \in [150,250]$ $\omega_c \in [10,150]$ $\omega_o \in [10,150]$
Population size p and Generation size g	$p = 300$ $g = 300$	$p = 400$ $g = 250$	$p = 450$ $g = 450$	$p = 400$ $g = 300$	$p = 250$ $g = 250$

4. A MOTION CONTROL CASE STUDY

In this study, the parameters in (3) are: $a = 3$, $b = 206$, and $J_t = 1$. To test the efficiency of the control design, a square wave torque disturbance with the amplitude of 10% max torque is added at t=0.5s; a square wave control signal disturbance with the amplitude of 1 is added at t=1.5s; 0.5% measurement white noise is introduced; and a resonant mode at $50Hz$ is added to the plant. The bounds of the design constraints in (15)-(18) are: $k_1 = 0.24$ and $k_2 = 8$; there are no specific limits for k_3 and k_4 in this motion system. The normalizing factors n_1 and n_3 in (14) are selected as the bounds of the design constraints: $n_1 = k_1 = 0.24$ and $n_3 = k_2 = 8$. The other normalizing factors n_2, n_4, n_5, and n_6 in (14) are derived from a well-tuned PID controller used as a benchmark: $n_2 = 0.0127$, $n_4 = 9.1632$, $n_5 = 3.3934e+3$, and $n_6 = 8.4513e+5$. The weights in (14) are selected such that each constituent in the cost function has the same priority: $w_1 = w_2 = w_3 = w_4 = w_5 = w_6 = 1$.

Using GA to search the tuning parameters for each control algorithm, Table 1 shows the simulation results over 30 runs, where $J_{c\min}$ is the minimum value of J_c, J_{cave} is the average value of J_c, *Stdev* is the standard deviation of J_c, and the best parameters are the parameter settings that generate the minimum values of J_c for each controller. The small standard deviations over 30 runs for each control method demonstrate the robustness of the GA. It is observed that: the cost function value of the worst case, PID control, is 1.98 times larger than that of LADRC. It shows that LADRC is much better than other control methods. It is not surprising because LADRC actively compensate for the unknown dynamics and disturbances. Figure 4 shows the system performance of PID control. Figure 5 shows the system performance of LADRC. It is immediately obvious that LADRC achieves much more satisfactory performance compared to other control techniques.

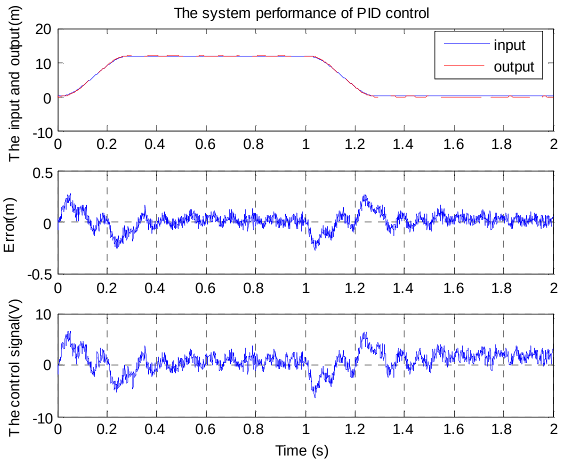


Fig. 4. The system performance of PID control.

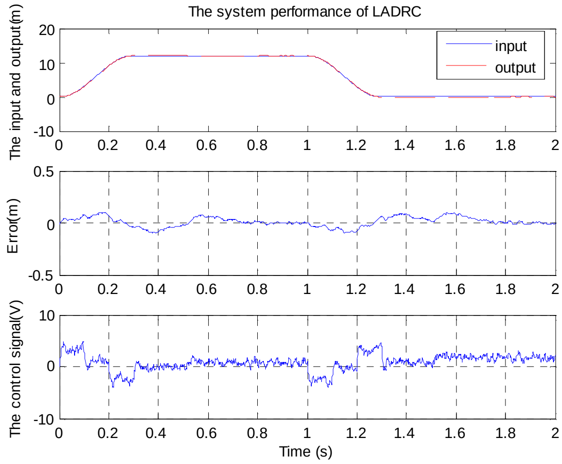


Fig. 5. The system performance of LADRC.

Together with Table 1, the case study indicates that LADRC is by far the best motion controller in terms of both the performance and the ease of design and tuning. It not only attains the lowest cost function values but also requires much less modeling information compared to the next best method, the loop shaping method. Given the plant in (1), LADRC only needs the approximate range of *b*, while the loop shaping method requires the transfer function model of the linear approximation of the plant. The limited performance of various PID controllers is as expected. The result here verifies why the cascaded control with velocity feedforward compensation is a favorite motion control strategy in practice. Among the PID solutions, it offers the best tracking performance. The PID control with leadlag compensator increases the stability margins of the control system but little else.

5. CONCLUSION

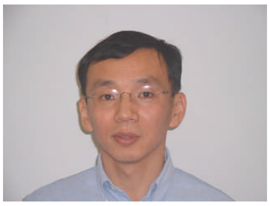
Efficient determination of the most suitable controller and its parameter setting for motion control is a very constructive work. In this paper, a conventional PID controller as well as its variations, and two alternative control algorithms are applied to address a class of motion control problems. A novel cost function is proposed and the research is formulated as an optimization problem. GA is employed to facilitate the search of global minimum and determine the optimal tuning parameters for each control technique. Simulation shows the robustness and effectiveness of the GA based tuning method. Five different control techniques are investigated and evaluated to show their respective advantages. Finally, the results in this paper go a long way in answering the important question of "what is the best control law for a given problem". For the case studied in this paper, the answer is clearly LADRC.

References

1. G. Ellis, "Control System Design Guide", 2nd Edition, *Academic Press*, 2000.

2. "DC Motors, Speed Controls, Servo Systems", The Electro-Craft Engineering Handbook, 5th Edition, *Reliance Motion Control Inc.*.
3. J. Tal, "Motion Control Systems", The Control Handbook. Editor Williams S. Levine, pp. 1382-1386, *CRC Press and IEEE Press*, 1995.
4. D. Rapini, "New Directions In Motion Control", *Motion Control*, pp.37-38, Jan./Feb. 1999.
5. B. Armstrong and C. Canudas de Wit, "Friction Modeling and Compensation", The Control Handbook. Editor Williams S. Levine, pp. 1369-1382, *CRC Press and IEEE Press*, 1995.
6. Z. Gao, "Scaling and bandwidth-parameterization based controller tuning", *Proceedings of American Control Conference*, pp.4989-4996, 2003.
7. Z. Gao, S. Hu, and F. Jiang, "A novel motion control design approach based on active disturbance rejection", *Proceedings of IEEE Conference on Decision and Control*, pp.4877-4882, 2001.
8. D. E. Goldberg, "Genetic Algorithm in Search, Optimization, and Machine Learning", *Addison-wesley*, 1989.
9. K. Ito, M. Iwasaki, and N. Matsui, "GA-based practical compensator design for a motion control systems", *IEEE/ASME Transactions on Mechatronics*, vol 6, no 2, pp143-148, 2001.
10. C. Vlachos, J. T. Evans, and D. Williams, "PI controller tuning for multivariable process using genetic algorithms", *Proceedings of IEE Conference on Genetic Algorithms in Engineering Systems: Innovations and Applications*, no. 446, pp. 43-49, 1997.
11. R. J. Eucker and Z. Gao, "A novel self-tuning control design approach for industrial applications", *Proceedings of IEEE Industrial Application Society 2000 Annual Meeting and World Conference on Industrial Applications of Electrical Energy*, pp1306-1310, 2000.
12. B. Boulter and Z. Gao, "A novel approach for efficient self-tuning web tension regulation", *Real-Time Systems, The International Journal of Time-Critical Computing Systems*, vol. 11, no. 3, pp265-287, Nov. 1996.
13. Z. Gao, "An algorithm approach to loop shaping with applications to self-tuning control systems", *Journal of the Franklin Institute,* vol. 332B, no.6, pp. 643-656, 1995.
14. Z. Gao and P. Antsaklis, "New methods for control system design using matrix interpolation", *Proceedings of IEEE Conference on Decision and Control*, pp.2506-2511, 1996.
15. Z. Gao, Y. Huang, and J. Han, "An alternative paradigm for control system design", *Proceedings of IEEE conference on Decision and Control*, pp4578-4585, 2001.
16. Z. Gao, "From linear to nonlinear control means: A practical progression," *The Instrumentation, Systems, and Automation Society Transactions*, Vol. 41, pp. 177-189, 2002.
17. Z. Gao, "On active disturbance rejection control", to be presented at the *American Control Conference,* 2006.
18. F. J. Goforth "On motion control design and tuning techniques", *Proceedings of American Control Conference*, pp.716-721, 2004.

Qing Zheng received the B.Eng degree from North China University of Technology, Beijing, China in 1996 and the M.Eng degree from the National University of Singapore in 2003, both in electrical engineering. She was an Electronic Engineer with Beijing Institute of Control Devices, Beijing, China, from 1996-2001. Currently, she is pursuing the Ph.D. degree at Cleveland State University. Her present research interests include disturbance rejection control, robust and multivariable control and optimization with emphasis on their applications in process and chemical industries.



Zhiqiang Gao (M'89) received the Ph.D. degree in electrical engineering from the University of Notre Dame in 1990. He is currently an Associate Professor of Electrical Engineering Department and the Director of Center for Advanced Control Technologies at Cleveland State University. He and his group at CSU have collaborated extensively with industry and conducted multi-million dollar research and development projects. Employing an experimental science philosophy in research, the team of researchers actively experiments and explores effective control mechanisms in solving real world problems. This research philosophy has led to the discovery of a set of innovative solutions including wavelet based multi-resolution controller, nonlinear PID, active disturbance rejection, fuzzy logic, self-tuning, CAD software package, real time health monitoring and failure identification, etc. Applications of these techniques can be found in many industry sectors such as automotive, paper and pulp, manufacturing, medical instrumentation, and space electronics. Dr. Gao also lectures broadly in industry on basic and advanced control topics. More information can be found at http://cact.csuohio.edu/ .