Genetic Algorithm Tuning of Lyapunov-Based Controllers: An Application to a Single-Link Flexible Robot System

S. S. Ge, *Member, IEEE,* T. H. Lee, *Member, IEEE,* and G. Zhu

***Abstract*— In this paper, a systematic controller design approach is proposed to guarantee both closed-loop stability and desired performance of the overall system by effectively combining genetic algorithms (GA's) with Lyapunov's direct-controller design method. The effectiveness of the approach is shown by using a simple and efficient decimal GA optimization procedure to tune and optimize the performance of a Lyapunov-based robust controller for a single-link flexible robot. The feedback gains of the controller are tuned by the GA optimization process to achieve good results for tip motion control of the single-link flexible robot based on some suitable fitness functions. The paper includes results of simulation experiments demonstrating the effectiveness of the proposed genetic algorithm approach.**

I. Introduction

FOR control system design and applications, the property of closed-loop stability is a very basic requirement, but it is never the only one. Besides stability, the user is usually very interested in manipulating the performance of the system in terms of overshoot, oscillation and settling time. In many cases, controllers designed based on the Lyapunov's direct method are primarily concerned with closed-loop stability and robustness. For many such controllers, there are currently no systematic approaches to choose the controller parameters to obtain the desired performance. The controller parameters are usually determined by trial-and-error through simulations and experimental tests. In such cases, the paradigm of genetic algorithms (GA) appear to offer an effective way for automatically and efficiently searching for a set of controller parameters for better performance, as it is an increasingly acknowledged observation that GA is a global, robust, and data-independent search technique [5], [6]. It has also been shown that compared with other traditional heuristic optimization methods [10], [11], GA is likely to be more computationally efficient. This is especially so when the problem is large. While it is possible to try other optimization methods to find a solution to the problem posed, the advantage of using GA in this situation is that we do not have to *reformulate* the problem as GA allows us to carry out the optimization in its natural setting. This is definitely an important advantage in applications. We hereby proposed a systematic controller design approach for both closed-loop stability and desired performance by using GA's to tune the Lyapunov-based controllers. In this way, we can guarantee not only the closed-loop stability but also the performance of the system.

The effectiveness and usefulness of the approach is demonstrated by applying a simple and efficient decimal GA optimization procedure to tune and optimize the performance of a Lyapunov-based robust controller previously developed in [1] for single-link flexible robots. Specifically, the procedure seeks to tune the parameters of the controller to improve the tip motion performance of a single-link flexible robot system. The robust controller in [1] was developed based on Lyapunov's direct method, and stability of the closed-loop system can be assured by selecting a set of positive-valued controller parameters. However, in the development in [1], the issue of the choice of controller parameters to achieve "good" performance was not resolved. In this paper, the very promising GA paradigm is used to automatically and efficiently tune the set of controller parameters to achieve good results for tip motion control of the single-link flexible robot based on suitable fitness functions. The GA utilized is selected to be of the decimal real number type (instead of binary type) to achieve a simple and efficient computational process [7]. Four types of fitness functions are considered for optimization of the controller performance. Two of them are the normally used integral of squared errors ISE and integral time-multiplied absolute value of errors, $ITAE$ [2]. The other two, the modified ISE or $MISE$, and the modified $ITAE$ or $MITAE$, are similar to the above except that overshoots are penalized explicitly since overshoots and oscillations of the tip trajectory are quite undesirable in accurate tip position control of flexible robots. The paper includes results of simulation experiments demonstrating the effectiveness of the proposed genetic algorithm approach.

The paper is organized in the following manner. The dynamics of a single-link flexible robot is introduced in Section II, and the robust Lyapunov-based controller previously developed in [1] is briefly reviewed in Section III. The proposed approach to utilize a simple and efficient decimal GA optimization procedure to tune and optimize the performance

Manuscript received October 20, 1995; revised November 28, 1995.

The authors are with the Center for Intelligent Control, Department of Electrical Engineering, National University of Singapore, Singapore 119260.

Publisher Item Identifier S 0278-0046(96)03300-X.

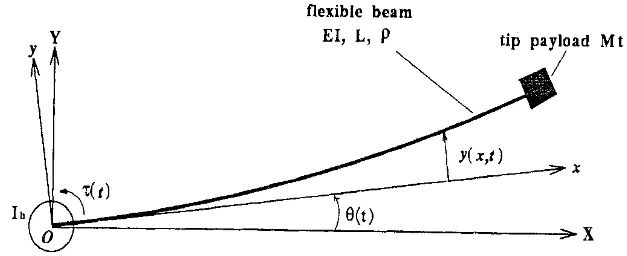


Fig. 1. Single-link flexible robot.

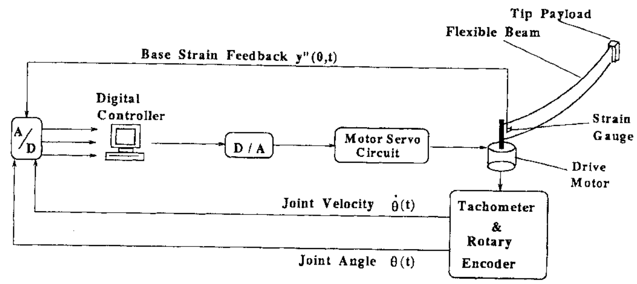


Fig. 2. Single-link flexible robot control system.

of the Lyapunov-based robust controller is presented in Section IV. Finally, Section V presents the results of simulation experiments demonstrating the effectiveness of the proposed genetic algorithm approach.

II. Dynamics of a Single-Link Flexible Robot

Consider a flexible beam which is clamped at its base on the rotor of a motor, and loaded by a point mass at its tip, as shown in Fig. 1. The flexible manipulator is rotating in the horizontal plane and thus, gravitational loading does not feature in this situation. Frame XOY is the fixed base frame and frame xOy is the local frame rotating with the hub. The relevant system parameters and variables are defined as follows.

- L length of the beam;
- EI uniform flexural rigidity of the beam;
- M_t point mass tip payload;
- I_h total moment of inertia with respect to the rotation joint axis of the beam;
- τ control torque;
- $\theta(t)$ joint angle;
- $y(x, t)$ elastic deflection measured from the undeformed beam.

Let $p(x, t) = x\theta(t) + y(x, t)$ represent the position of a point on the flexible beam, consistent with notation used elsewhere in the literature [3], [4]. The total kinetic energy E_k can be described by

$$E_k = \tfrac{1}{2} I_h\dot{\theta}^2 + \frac{\rho}{2}\int_0^L \dot{p}^2(x, t)\, dx + \tfrac{1}{2} M_t\dot{p}^2(L, t) \qquad (1)$$

and the total potential energy E_p is given by

$$E_p = \frac{EI}{2}\int_0^L [y''(x, t)]^2\, dx \qquad (2)$$

where, in the usual manner, the dots and primes denote the derivatives with respect to time and space, respectively. Using the extended Hamilton's principle:

$$\int_{t_o}^{t_f} \delta(E_k - E_p + \tau\theta)\, dt = 0 \qquad (3)$$

and substituting (1) and (2) into (3), we obtain the system dynamics described by the following partial differential equations (PDE):

$$(I_h + \tfrac{1}{3}\rho L^3)\ddot{\theta}(t) + \rho\int_0^L x\ddot{y}(x, t)\, dx$$
$$+ M_tL[L\ddot{\theta}(t) + \ddot{y}(L, t)] = \tau \qquad (4)$$
$$\rho[x\ddot{\theta}(t) + \ddot{y}(x, t)] = -EIy''''(x, t). \qquad (5)$$

The corresponding boundary conditions are given by

$$y(0, t) = 0 \qquad (6)$$
$$y'(0, t) = 0 \qquad (7)$$
$$y''(L, t) = 0 \qquad (8)$$
$$EIy'''(L, t) = M_t[L\ddot{\theta}(t) + \ddot{y}(L, t)]. \qquad (9)$$

Equations (6) and (7) hold because frame xOy is selected such that the axis Ox is tangent to the beam at the base. The third boundary condition (8) comes directly from the zero value of the bending moment at the tip, and the fourth one (9) is actually the motion equation of the tip payload M_t.

III. Controller Design

The objective of tip position control is to drive the tip of the flexible beam to a predefined position as fast as possible and with minimal overshoot and oscillation. If $p(L, t)$ is used to represent the tip position, and the final joint position is defined by θ_f, then we should have $p(L, t) = L\theta_f$ at the end of the control. One of the simplest tip position controllers developed in our previous work in [1] is given by

$$\tau = -k_p[\theta(t) - \theta_f] - k_v\dot{\theta}(t) - k_f y''(x_s, t)\int_0^t \dot{\theta}(s)y''(x_s, s)\, ds \qquad (10)$$

where $k_p, k_v, k_f > 0$. The closed-loop system has been shown to be stable by Lyapunov's direct method in [1]. From (10), one can see that only the joint angle θ, joint velocity $\dot{\theta}$, and strain feedback $y''(x_s, t)$ (at $x = x_s$) are needed to implement the controller. A suitable implementation structure is shown in Fig. 2 where the choice of the base strain feedback $y''(0, t)$ ($x_s = 0$) is utilized.

In [1], it has been shown that stability of the closed-loop system is attained as long as the controller gains, k_p, k_v, and k_f, are chosen as (any) positive scalars. Thus, subject to the simple constraint on the controller gains mentioned above, the property of stability of the closed loop system is independent of the flexible link system parameters; and the closed-loop system is thus robustly stable with respect to parametric variations in the flexible link system. However, while we have the property that any set of feedback gains $k_p, k_v, k_f > 0$ will stabilize the closed-loop system, it must be noted that such a set of arbitrarily selected feedback gains does not necessarily guarantee satisfactory tip motion performance as there exists a trade-off between the joint motion speed and the tip deflections encountered. Thus an effective procedure is needed to assist in arriving at the choice of controller gains to achieve the "best" tip motion performance possible for a given flexible link system. The same problem also exists in a large class of Lyapunov-based controllers, in which only stability but no system performance is considered. Hitherto, the usual way to attempt to pick the set of controller gains yielding good tip motion performance has been through running many simulations and then selecting a set that yields reasonable performance. The paradigm of GA optimization appears to provide an efficient and effective automatic tuning methodology for this problem and this is explored in the remaining sections.

IV. Genetic Algorithm Optimization

In this section, the methodology is used to optimally select feedback gains in the controller (10) under certain suitably specified performance fitness functions to achieve the desired tip motion performance of the flexible link system.

Compared with conventional optimization methods, GA possesses many advantages; for example, it is global, data-independent, and robust [5], [6]. Further, GA can be directly applied to solve an optimization problem with a certain fitness function without reformulating the problem into a suitable form. The GA optimization technique is based on the operations observed in natural selection and genetics. It works on a coded parameters set (population), which is initially generated at random, to optimize certain fitness function. Generally, the parameters are encoded into binary strings which are called chromosomes, and GA operates on the chromosomes instead of the parameters themselves. In this paper, a simple GA variant which works directly on real (decimal) parameters is used. Decimal-type GA's are equivalent to the traditionally used binary-type GA's in optimization [7]. While binary-type GA's, due to their advantage of simplicity, are widely used in actual hardware implementations, real-type GA's for computer-based numerical simulations lead to high computational efficiency, smaller computer memory requirements with no reduction of precision and greater freedom in selecting genetic operators [9].

To further simplify the genetic optimization process, the set of tuning parameters (k_p, k_v, k_f) is reduced to (ω_n, k_f), where ω_n is the natural frequency of the following 2nd-order system [1]:

$$I_{eq}\ddot{e}(t) + k_v\dot{e}(t) + k_p e = 0 \tag{11}$$

which is obtained under the assumption of rigid link robot motion and under the so called joint proportional-derivative (PD) controller by setting $k_f = 0$ in (10). Here, $I_{eq} = I_h + \frac{1}{3}\rho L^3 + M_t L^2$ (the equivalent inertia under the assumption of rigid link), and $e = \theta - \theta_f$ (the joint motion error). Since θ_f is constant, we have $\ddot{e} = \ddot{\theta}$ and $\dot{e} = \dot{\theta}$. For the 2nd-order system (11), if critical damping ($\xi = 1$) is assumed for fastest

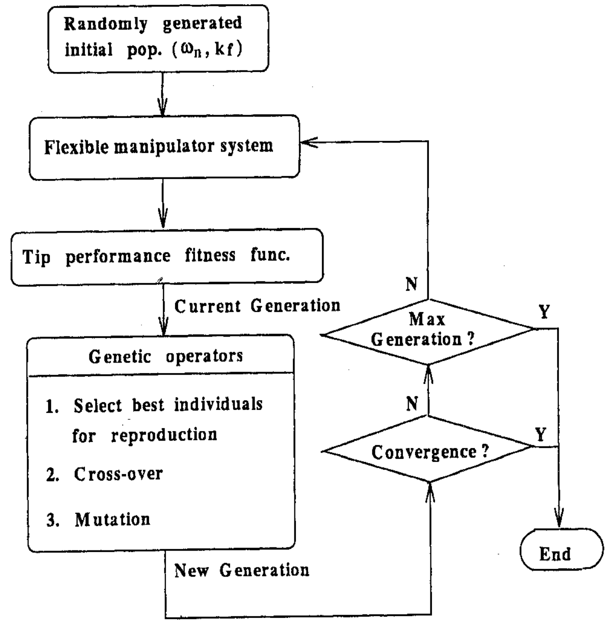


Fig. 3. GA optimization system.

overshoot free performance of θ, the feedback gains k_p and k_v are given by

$$\begin{cases} k_p = I_{eq} \cdot \omega_n^2 \\ k_v = 2I_{eq} \cdot \omega_n \end{cases}. \tag{12}$$

When I_{eq} is known, we shall consider only ω_n and k_f instead of k_p, k_v, and k_f to simplify the genetic optimization process. If I_{eq} is unknown, the parameters k_p, k_v, and k_f can then be simultaneously tuned in the same manner as described below. As a matter of fact, it is very desirable to have less tuning parameters for actual engineering implementation. Although (12) actually puts an extra constraint on the optimization, it is shown that comparatively good joint motion can be obtained while the computational cost is reduced. The overall GA optimization system for the single-link flexible robot is described by the flowchart in Fig. 3.

The initial population of size N is generated randomly to start the optimization process. The total population of each generation is evaluated based on the data from the single-link flexible robot simulation system using certain suitable chosen performance fitness functions. (In this case, these are the ISE, $ITAE$, $MISE$, and $MITAE$ fitness functions, which will be defined in detail later.) The next generation can be obtained through the genetic operators. The genetic operators are the core of the GA and are described in detail in the following.

A. Reproduction

By using the values of the performance fitness functions, select the best $N/2$ individuals of the current generation to be the parents for generating the next generation. This means that only genetically good individuals are selected to be parents.

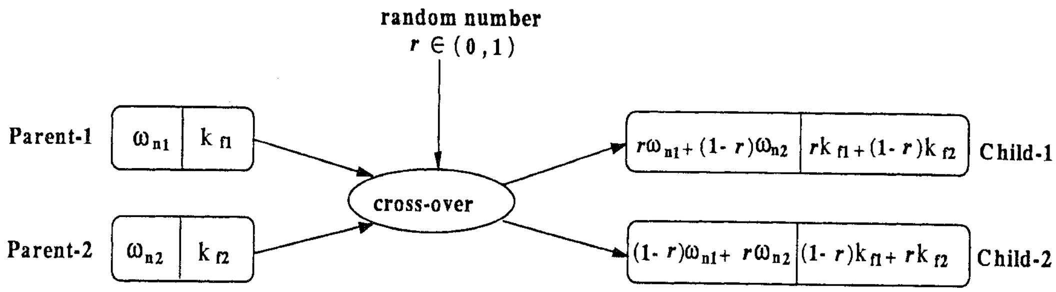


Fig. 4. Cross-over operator.

B. Cross-over

Randomly select two parents to exchange genetic information with each other and generate two new individuals. The cross-over operator is repeated until $N/2$ new individuals are generated so as to keep the population size at constant N. As shown in Fig. 4, the cross-over operator can be mathematically described as

if parents are (ω_{n1}, k_{f1}) and (ω_{n2}, k_{f2}), then

$$\text{Child-1:} \begin{cases} \omega_n = r * \omega_{n1} + (1-r) * \omega_{n2} \\ k_f = r * k_{f1} + (1-r) * k_{f2} \end{cases}$$

$$\text{Child-2:} \begin{cases} \omega_n = (1-r) * \omega_{n1} + r * \omega_{n2} \\ k_f = (1-r) * k_{f1} + r * k_{f2} \end{cases}$$

where $r \in (0, 1)$ is a random number. Such a cross-over operator for real (decimal) numbers is called *weighted average* operator [7]. One can see that the operator directly works with the real decimal numbers pairs (ω_{n1}, k_{f1}) and (ω_{n2}, k_{f2}) instead of any coded strings.

C. Mutation

Mutation takes place with a certain probability; thus the genetic information of the selected individual changes and subsequently new genetic information is introduced. In our simulations, the mutation rate is set to be 0.1 and the pair (ω_n, k_f) of the selected individual undergoes the following changes:

$$\begin{aligned} \omega_n &= \omega_n + (r_1 - 0.5) * 2 * \omega_{n-\max} \\ k_f &= k_f + (r_2 - 0.5) * 2 * k_{f-\max} \end{aligned} \tag{13}$$

where $r_1, r_2 \in (0, 1)$ are two random numbers, and $\omega_{n-\max}$ and $k_{f-\max}$ are maximum changes of ω_n and k_f under mutation.

Next, we shall introduce the fitness functions used in our simulations. Since the optimization of the fitness functions is the basic goal of GA, different fitness functions will produce, in general, different optimization results. The objective of the optimization here is to obtain better tip motion performance of the single-link flexible robot shown in Fig. 1, i.e., to find a set of feedback gains with which the controller (10) can drive the tip of the flexible beam to a predefined position as fast as possible and with minimal overshoot/oscillation.

If, as stated in Section II, the final position is given by $p(L, t) = L\theta_f$, then the following are two appropriate choices of fitness functions:

$$ISE = \int_0^T [L\theta_f - p(L, t)]^2 \, dt \tag{14}$$

$$ITAE = \int_0^T t|L\theta_f - p(L, t)| \, dt \tag{15}$$

where ISE means the integral of squared errors and $ITAE$ is the integral time-multiplied absolute value of errors [2]. Since overshoot of the tip trajectory is quite undesirable in tip position control of flexible robots, we will explicitly consider the overshoot and thus introduce the following two modified fitness functions:

$$MISE = \int_0^T K[L\theta_f - p(L, t)]^2 \, dt \tag{16}$$

$$MITAE = \int_0^T Kt|L\theta_f - p(L, t)| \, dt \tag{17}$$

where $MISE$ and $MITAE$ denote modified ISE and modified $ITAE$, respectively, and the penalty weight K is defined as

$$K = \begin{cases} 1.0, & \text{when } |p(L, t)| \le L\theta_f \text{ (no overshoot)} \\ 10.0, & \text{when } |p(L, t)| > L\theta_f \text{ (overshoot)} \end{cases} \tag{18}$$

i.e., a large weight has been put on the overshoot of the tip performance. For the four fitness functions (14)–(17), the integrals are taken within the evaluation time interval $[0, T]$. It should be noted that generally the evaluation time interval should be $[0, \infty]$. However, only the finite time interval $[0, T]$ will be considered for practicality and realizability. Obviously, the smaller the values of the four fitness functions, the better tip motion performances we can achieve.

V. Simulation Experiments

In this section, numerical simulations are carried out to verify the effectiveness of the GA optimization procedure described previously. The system parameters used in the simulations are given by Table I.

Before we begin the simulations, the ranges of controller parameters, i.e., ω_n and k_f, must be specified. As noted previously, any choice of the feedback gains, with k_p, k_v, and k_f as positive scalars, will yield the property of uniform stability for the controller given by (10). Thus, for k_p and k_v given by (12), we can use any $\omega_n, k_f > 0$ to implement the controller. However, excessively large ω_n and k_f will

TABLE I
SYSTEM PARAMETERS

L	$1.0\ m$	EI	$2.0\ \ Nm^2$
ρ	$0.1\ \ Kg/m$	M_t	$0.05\ \ Kg$
I_h	$0.5\ \ Kgm^2$		

obviously lead to very large calculated control torques, and subsequently lead to actuator saturation in practice. To avoid this, we further impose the constraint that

$$|\tau| \le \tau_{\max} = 10\ \mathrm{N\cdot m}$$

where $\tau_{\max}$ is the maximum generating torque of the control motor. Some initial simulations of the flexible robot control system show that generally the controller in (10) reaches its maximum at the initial moment of the control process. Since we always set the initial position of the flexible beam at $\theta = 0$ for point-to-point control, the feedback gain k_p should thus satisfy

$$0 < k_p \le \frac{\tau_{\max}}{|\theta_f|}.$$

Accordingly, from (12), we obtain the range of the natural frequency ω_n as

$$0 < \omega_n \le \sqrt{\frac{\tau_{\max}}{|\theta_f| I_{eq}}}.$$

When the system parameters are set as in Table I, we obtain $I_{eq} = 0.5833$ Kgm^2 and consequently

$$0 < \omega_n \le 5.855. \tag{19}$$

The range of k_f is empirically obtained from the initial simulations of the system. With the system parameters in Table I, and if the base strain feedback is considered (i.e., $x_s = 0$, as shown in Fig. 2), a reasonable range of k_f is given by

$$100 < k_f < 2000. \tag{20}$$

Since inequalities (19) and (20) must be satisfied in the genetic optimization procedure, the mutation results of ω_n and k_f from (13) which exceed the above ranges will be ignored. It should be noted that such a problem only exists for the mutation operator. For the weighted average cross-over operator given in (13), the children's natural frequencies ω_n's and feedbacks gains k_f's will always be in the ranges

$$(\mathrm{Min}\,[\omega_{n1},\, \omega_{n2}], \quad \mathrm{Max}\,[\omega_{n1},\, \omega_{n2}])$$
$$(\mathrm{Min}\,[k_{f1},\, k_{f2}], \quad \mathrm{Max}\,[k_{f1},\, k_{f2}]).$$

Thus, the children's genetic information generated by the cross-over operator (13) will always be valid.

Different choices of ω_n and k_f will lead to very different tip motion performance characteristics. The effects of ω_n and k_f can be qualitatively described as follows.

1) A smaller ω_n results in a smoother tip motion (less elastic vibration) but comparatively longer time to reach the final position, while a larger ω_n (corresponding to larger control torque) will lead to a faster joint motion but with more serious elastic vibration of the flexible beam.
2) A smaller k_f has less effect on the tip motion performance, i.e., the elastic vibration of the flexible beam cannot be effectively suppressed, while a too-large k_f will cause overshoot of the tip trajectory $p(L, t)$ which is quite undesirable in tip motion control.

Therefore, in order to obtain better tip performance, it is obvious that some optimization procedures are necessary to optimally select ω_n and k_f. In this paper, the GA introduced in last section is used as an optimizer to tune the choices for ω_n and k_f and optimize the tip motion performance based on the four fitness functions (14)–(17).

In the simulations, the population size of each generation is set to be $N = 100$. The maximum mutation values of ω_n and k_f are selected to be $\omega_{n-\max} = 0.2$ and $k_{f-\max} = 100$. $k_{f-\max}$ is much larger than $\omega_{n-\max}$ because it was found that the tip motion performance of the flexible robot system is less sensitive to the change of k_f than to that of ω_n. The evaluation time interval is set to be $0 \le t \le 2$ s (i.e., $T = 2$ s) since we also found that generally, the tip of the flexible beam can reach, and settle down, in the neighborhood of the final position $p(L, t) = L\theta_f = 0.5$ m (i.e., $\theta_f = 0.5$ rad) in 2 s. Another problem we must take into consideration is the terminating condition of the GA optimization process. Generally speaking, the required maximum number of generations of GA optimization is problem-dependent. It may take too long to obtained the "best" result. From the engineering point of view, a convergence decision is needed to terminate the GA operations when the result is good enough. For this consideration, the following termination decisions are introduced: 1) if the considered fitness function does not change much (say less than $T_M = 0.005$) in a certain number of generations (say $G_N = 50$ generations), we assume that the optimization has converged and the process is terminated; and 2) otherwise, the process shall be terminated at the maximum number of generations (say $G_M = 500$). Obviously, the selection of the above numbers are not unique. Generally speaking, smaller T_M or larger G_N will lead to better optimization results but need longer optimization time. The introduction of the above convergence decision may cause the problem of local minimum, but it is more realistic in practice.

To facilitate a standard base for comparison, the four optimization experiments, corresponding to the four fitness functions (14)–(17), are carried out using the same set of initial population which is generated randomly. The terminating generations of the four fitness functions are 253 (ISE), 247 ($MISE$), 235 ($ITAE$), and 212 ($MITAE$). This implies that the GA optimization based on each of the four fitness functions can obtain the good enough result [convergence decision 1)] before reaching the maximum generations $G_M = 500$. For each optimization process, only ω_n, k_f, and the value of the corresponding fitness function of the best individual (with smallest fitness function value) in every generation are recorded. Fig. 5 shows the feedback gains k_p and k_v [obtained

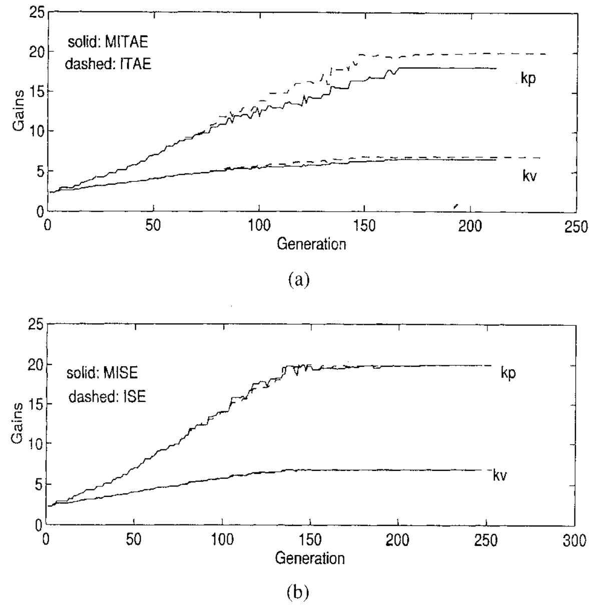


Fig. 5. Best feedback gains k_p and k_v w.r.t. (a) $ITAE$ and $MITAE$. (b) ISE and $MISE$.

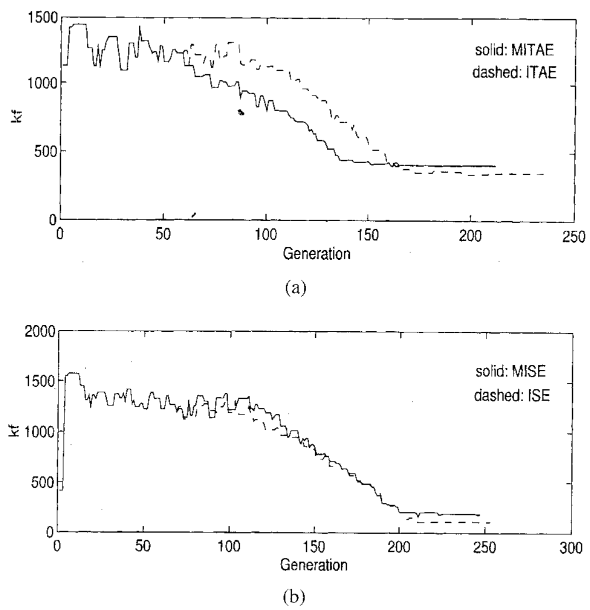


Fig. 6. Best feedback gain k_f w.r.t. (a) $ITAE$ and $MITAE$. (b) ISE and $MISE$.

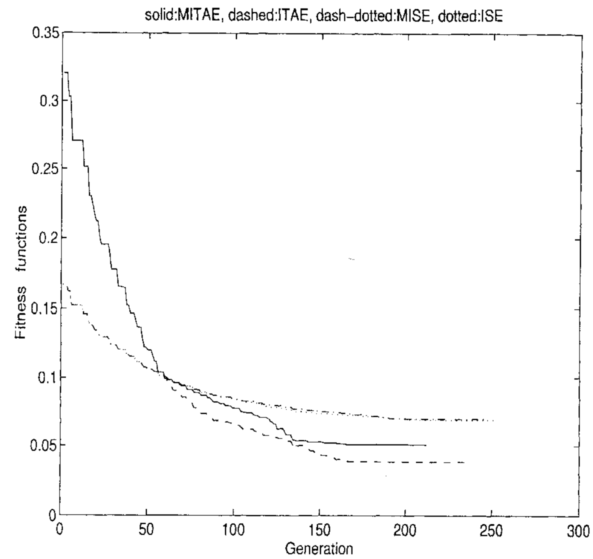


Fig. 7. Best (smallest) values of $ITAE$, $MITAE$, ISE, and $MISE$.

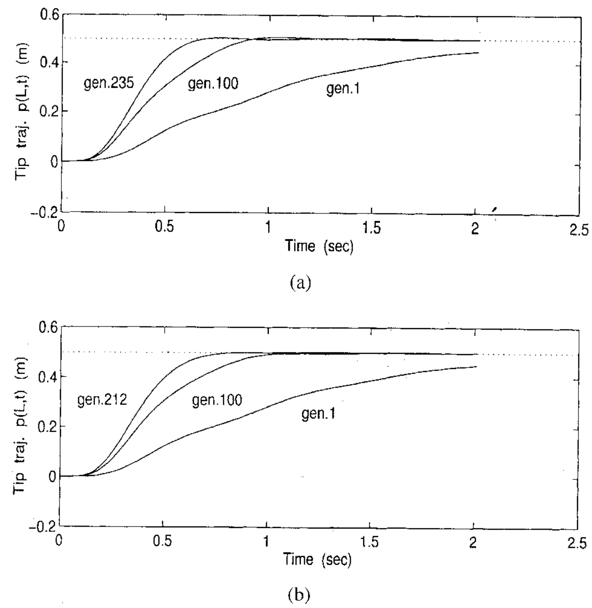


Fig. 8. Best tip performance of Generation 1, 100, and the final generation w.r.t. (a) $ITAE$. (b) $MITAE$.

from (12)] in accordance with the best individuals. It can be noted that k_p and k_v with respect to the fitness functions ISE and $MISE$ [Fig. 5(b)] are very close to each other. The corresponding k_f's are plotted in Fig. 6. The best values of the four fitness functions are shown in Fig. 7. From Fig. 7, one can observe that better performance (corresponding to smaller values of the fitness functions) will be obtained with the progression of the GA optimization processes.

In order to show clearly the effect of the genetic algorithm optimization, the tip motion trajectories, $p(L, t)$, of the best individuals of several selected generations are plotted in Figs. 8 and 9. It can be seen from these figures that tip motion performance is improved as the number of generations increases. Finally, the best tip performances (i.e., the results of the last generations) with respect to the four fitness functions are plotted together in Fig. 10. From the observations in Figs. 8–10, it is pertinent to make the following remarks.

A. *Remarks:*

1) The overshoot of the tip trajectories, for controllers tuned by the GA optimization for the fitness functions $ITAE$ and ISE, are a little larger than that for their modified types ($MITAE$ and $MISE$), respectively. This is as expected since larger weights have been put on the latter

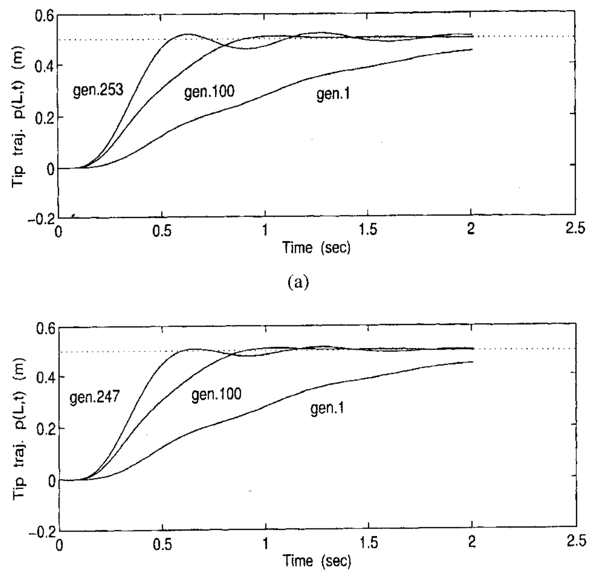


Fig. 9. Best tip performance of Generation 1, 100, and the final generation w.r.t. (a) ISE. (b) $MISE$.

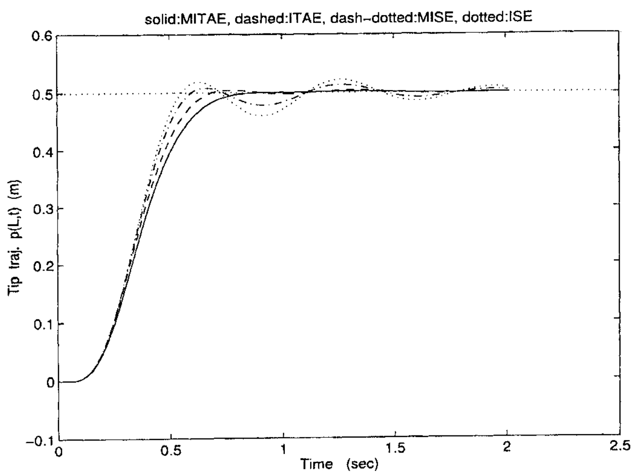


Fig. 10. Best tip performance w.r.t. $ITAE$, $MITAE$, ISE, and $MISE$.

two fitness functions when overshoot takes place. The price we pay for the smaller overshoot is that the tip motion is slowed down (though by not very much).

2) The drawback of ISE and $MISE$ as fitness functions for tip motion performance is that their minimization cannot effectively avoid oscillations of the tip trajectory. In contrast, the minimization of the fitness functions $ITAE$ and $MITAE$ yield better results in the sense that the oscillations are significantly smaller because the time t has been introduced to serve as an increasing weight when the tip of the flexible beam approaches the final position.

For the tip position control of the single-link flexible robot, there is a tradeoff between the fast motion speed and the small oscillations/overshoots. From the figures, one can see that the oscillations/overshoots of $ITAE$ and $MITAE$ are significantly smaller than that of ISE and $MISE$, while the corresponding tip motions are only slightly slowed down. Further, it has been shown that $MITAE$ and $ITAE$ achieve the satisfactory results at 212 and 235 generations, respectively, which are faster than $MISE$ and ISE. Therefore, we conclude that the fitness functions $MITAE$ and $ITAE$ are more suitable than $MISE$ and ISE in the sense of achieving much smaller oscillations and negligible overshoots of the tip motion performance of the single-link robot.

VI. Conclusion

In this paper, the genetic algorithm optimization procedure has been used successfully to achieve optimal tip motion performance for a Lyapunov-based controller for a single-link flexible robot. While the controller can assure the stability of the closed-loop system simply by a choice of a set of positive-valued feedback gains, the GA procedure greatly improves its efficacy by providing a facility to tune the feedback gains to optimize certain desired fitness functions. The paper has also included results of simulation experiments demonstrating the effectiveness of the proposed Genetic Algorithm approach. The approach of combining the Lyapunov-based controller design with the GA optimization technique presented in this paper can be generalized and applied to a wide range of systems including adaptive controllers design based on Lyapunov's direct method, by which not only the closed-loop stability is guaranteed, but also the system performance can be optimized.

References

[1] S. S. Ge, T. H. Lee, and G. Zhu, "Improving joint PD control of single-link flexible robots by strain/tip feedback," in *Proc. Conf. Control, Applications,* MI, 1996.

[2] W. C. Schultz and V. C. Rideout, "Control system performance measures: Past, present and future, " *IEEE Trans. Automat. Contr. AC-6,* vol. 22, pp. 22–35, 1961.

[3] H. Krishnan, "Bounded input discrete-time control of a single-link flexible beam, " Master thesis, Dept. Elec. Eng., Univ. Waterloo, 1988.

[4] Y. Sakawa, F. Matsuno, and S. Fukushima, "Modeling and feedback control of a flexible arm, " *J. Robotic Syst.,* vol. 2, no. 4, pp. 453–472, 1985.

[5] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning, Reading.* Reading MA: Addison-Wesley, 1989.

[6] L. Davis, *Handbook of Genetic Algorithms.* New York: Thomson, 1991.

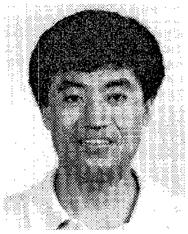
[7] S. Obavashi, "Genetic algorithm for aerodynamic inverse optimization problems," in *Proc. 1st IEE/IEEE Int. Conf. GA's in Engineering Systems: Innovations and Applications, GALESIA '95,* 1995, pp. 7–12.

[8] V. Salami and G. Cain, "An adaptive PID controller based on genetic algorithm processor," in *Proc. 1st IEE/IEEE Int. Conf. GA's in Engineering Systems: Innovations and Applications, GALESIA '95,* 1995, pp. 88–93.

[9] A. H. Wright, "Genetic algorithms for real parameter optimization," in *Foundations of Genetic Algorithms,* J. E. Rawlins, Ed. San Mateo, CA: Morgan Kaufmann, 1991, pp. 205–218.

[10] J. Zhao, S. McKenzie, and A. J. Soper, "A genetic algorithm for the design of multipoint connections in a local access network," in *Proc. 1st IEE/IEEE Int. Conf. GA's in Engineering Systems: Innovations and Applications, GALESIA '95,* 1995, pp. 30–33.

[11] J. F. Miller and P. Thomson, "Combinational and sequential logic optimization using genetic algorithms, " in *Proc. 1st IEE/IEEE Int. Conf. GA's in Engineering Systems: Innovations and Applications, GALESIA '95,* 1995, pp. 34–38.



S. S. Ge (S'90–M'92) received the B.Sc. degree in control engineering in 1986 from Beijing University of Aeronautics and Astronautics, Beijing, P.R. China, and the Ph.D. and the D.I.C. degrees in mechanical/electrical engineering in January 1993 from the Imperial College of Science, Technology and Medicine, University of London, U.K.

From May 1992 to June 1993, he was a post-doctoral research associate with the Department of Engineering, University of Leicester, U.K. Since July 1993, he has been with the Department of Electrical Engineering, National University of Singapore, as a lecturer. His current research interests are robust adaptive control, adaptive control of robots, neural network and fuzzy logic control, GA optimization, and real-time control systems.

T. H. Lee (M'88), for a photograph and biography, see p. 105 of the February 1996 issue of this TRANSACTIONS.

G. Zhu received the B.Eng. degree in automatic control in 1993 from Xi'an Jiao-tong University, Xi'an, Shaan'xi, P.R. China. He is currently working toward the Ph.D. degree in the Department of Electrical Engineering, National University of Singapore. His present research interests are modeling and control of flexible manipulators, robust control, adaptive control, and intelligent control.